

Process step descriptions

- [ARCH.900 Determine the operational needs](#)
 - [ARCH.901 Capture existing and initial operational understanding](#)
 - [ARCH.001 Define boundary of wider system of interest \(operating entity of SysABB\)](#)
 - [ARCH.004 Analyse trade space factors](#)
 - [ARCH.005 Define set of system of interest lifecycle variants](#)
 - [ARCH.002 Create initial set of operational capabilities](#)
 - [ARCH.144 Define enterprise goals](#)
 - [ARCH.003 Create initial set of operational actors & operational entities](#)
 - [ARCH.189 Create templates and static text for CONOPS, CONUSE and CONEMP](#)
 - [ARCH.006 Create initial set of operational states for operational entities/actors](#)
 - [ARCH.007 Create set of initial operational activities](#)
 - [ARCH.008 Define abstract concepts](#)
 - [ARCH.009 Define measures of effectiveness](#)
 - [ARCH.902 Analyse the operational capabilities to determine detailed operational needs](#)
 - [ARCH.177 Complete the definition of the operational capability of interest](#)
 - [ARCH.078 Create bare business operational scenario](#)
 - [ARCH.079 Create bare business operational process](#)
 - [ARCH.080 Model operational activities and interactions](#)
 - [ARCH.153 Model operational data](#)
 - [ARCH.159 Model operational states](#)
 - [ARCH.161 Map operational activities to operational states](#)
 - [ARCH.903 Assess & mitigate the operational risks](#)
 - [ARCH.171 Prepare business risk analysis](#)
 - [ARCH.012 Determine measure of effectiveness target values](#)
 - [ARCH.013 Identify risks to business effectiveness \(business loss-risk mapping\)](#)
 - [ARCH.172 Prepare security risk analysis](#)
 - [ARCH.018 Identify internal & external issues affecting successful security](#)
 - [ARCH.019 Determine need for information availability, confidentiality, integrity](#)
 - [ARCH.170 Identify security losses and threats](#)
 - [ARCH.173 Prepare safety risk analysis](#)
 - [ARCH.023 Identify relevant safety legislation and regulation](#)
 - [ARCH.025 Create foundation for safety risk model](#)
 - [Draft foundation accidents](#)
 - [Draft foundation hazard](#)
 - [Draft foundation control measure](#)
 - [ARCH.026 Define the state model of accidents, hazardous and safe state](#)
 - [ARCH.020 Identify & classify operational deviations](#)
 - [ARCH.011 Assess operational performance risks of scenario](#)
 - [ARCH.014 Evaluate risks to business effectiveness](#)
 - [ARCH.015 Define business risk control measures](#)
 - [ARCH.017 Assess security risks of scenario](#)
 - [ARCH.021 Evaluate security risks](#)
 - [ARCH.022 Define operational security measures](#)
 - [ARCH.024 Assess operational safety risks of scenario](#)
 - [ARCH.027 Evaluate operational safety risks](#)
 - [ARCH.028 Define operational safety measures](#)
 - [ARCH.029 Allocate risk control responsibilities \(op. activity to op. entity/actor\)](#)
 - [ARCH.175 Produce consolidated operational deviation analysis report](#)
 - [ARCH.904 Incorporate risk control measures in the operational processes](#)
 - [ARCH.031 Create operational scenario with risk control measures](#)
 - [ARCH.032 Create operational process with risk control measures](#)
 - [ARCH.033 Update operational activities and interactions](#)
 - [ARCH.147 Update operational data](#)
 - [ARCH.160 Update operational states](#)
 - [ARCH.162 Update operational activity to state mapping](#)
 - [ARCH.178 Update dependencies and pre/post conditions](#)

- [ARCH.905 Consolidate the operational needs](#)
 - [ARCH.121 Consolidate operational activities and interactions](#)
 - [ARCH.034 Produce concept of operations \(CONOPS\)](#)
 - [ARCH.140 Consolidate the allocation of operational activities](#)
 - [ARCH.183 Consolidate operational data](#)
- [ARCH.910 Determine the system requirements](#)
 - [ARCH.911 Identify the system's contribution to the operational needs](#)
 - [ARCH.035 Identify constraints on the system solution](#)
 - [ARCH.043 Define set of potential system actors](#)
 - [ARCH.049 Define set of potential system capabilities/missions](#)
 - [ARCH.039 Define set of potential system and actor functions](#)
 - [ARCH.036 Identify all alternative system boundaries](#)
 - [ARCH.042 Select system boundary \(or set of variants\)](#)
 - [ARCH.912 Finalise the system context and constraints](#)
 - [ARCH.048 Identify system implementation constraints](#)
 - [ARCH.045 Define complete set of system-level actors](#)
 - [ARCH.119 Define consolidated set of system capabilities/missions](#)
 - [ARCH.041 Transfer or trace upper level model elements to system level](#)
 - [ARCH.044 Define requirements from non-actor stakeholders](#)
 - [ARCH.051 Align the system boundary, capabilities and actors with collaboration projects](#)
 - [ARCH.913 Analyse the system capabilities to determine detailed system needs](#)
 - [ARCH.179 Complete the definition of the system capability of interest](#)
 - [ARCH.088 Define system functions and functional exchanges](#)
 - [ARCH.052 Create initial system exchange scenarios](#)
 - [ARCH.053 Create initial system functional chains](#)
 - [ARCH.054 Model data flowing between system functions](#)
 - [ARCH.055 Model states on system level](#)
 - [ARCH.056 Map system functionality to states](#)
 - [ARCH.158 Model external interface layers](#)
 - [ARCH.057 Model non-payload data on external interfaces](#)
 - [ARCH.058 Define measures of performance](#)
 - [ARCH.915 Assess & mitigate the risks of system failure](#)
 - [ARCH.065 Derive safety target to system functions and define additional risk control measure needed](#)
 - [ARCH.066 Identify system level deviations](#)
 - [ARCH.067 Populate the fault tree](#)
 - [ARCH.068 Assess deviation probability \(external constraints\)](#)
 - [ARCH.069 Add system level risk measures](#)
 - [ARCH.070 Calculate system deviation probability](#)
 - [ARCH.916 Incorporate risk control measures in system scenarios/functional chains](#)
 - [ARCH.141 Update system functions with risk control measures](#)
 - [ARCH.142 Create system exchange scenarios with risk control measures](#)
 - [ARCH.143 Create system functional chains with risk control measures](#)
 - [ARCH.149 Update system data with risk control measures](#)
 - [ARCH.150 Update system interface model with risk control measures](#)
 - [ARCH.163 Update system states](#)
 - [ARCH.164 Update function mapping to system states](#)
 - [ARCH.914 Produce the unregulated system documents](#)
 - [ARCH.062 Produce the concept of use \(CONUSE\)](#)
 - [ARCH.061 Produce the concept of employment \(CONEMP\)](#)
 - [ARCH.917 Finalise the system requirements](#)
 - [ARCH.152 Consolidate system functionality](#)
 - [ARCH.157 Consolidate system data](#)
 - [ARCH.081 Update constraints on system solution](#)
 - [ARCH.109 Consolidate traceability between model elements at system level and model elements at operational level](#)
 - [ARCH.082 Trade off system requirements and constraints](#)
 - [ARCH.083 Agree the system requirements with stakeholders](#)
- [ARCH.920 Define the logical architecture](#)
 - [ARCH.921 Decompose system functionality](#)
 - [ARCH.084 Execute automatic transition of system elements to logical level](#)
 - [ARCH.182 Split the system functions](#)

- [ARCH.085 Create or update functional chains at logical level](#)
 - [ARCH.087 Update logical data model](#)
 - [ARCH.191 Refine logical capability realisations](#)
- [ARCH.922 Define logical components](#)
 - [ARCH.180 Define logical component candidates](#)
 - [ARCH.187 Adopt logical component definitions from collaboration projects](#)
 - [ARCH.181 Reconcile candidate logical components](#)
- [ARCH.923 Apportion the non-functional requirements](#)
 - [ARCH.107 Apportion non-functional requirements to logical functions](#)
 - [ARCH.108 Define acceptance criteria for non-functional requirements](#)
- [ARCH.924 Finalise requirements at logical level](#)
 - [ARCH.154 Consolidate logical functional flow](#)
 - [ARCH.096 Allocate logical functions to logical components \(including alternative allocations\)](#)
 - [ARCH.086 Create or update exchange scenarios at logical level](#)
 - [ARCH.110 Consolidate traceability between model elements at logical level and model elements at system level](#)
 - [ARCH.190 Consolidate logical data](#)
 - [ARCH.099 Define behaviour for logical functions](#)
- [ARCH.930 Define the physical architecture](#)
 - [ARCH.931 Transfer logical level requirements to physical level](#)
 - [ARCH.111 Execute automatic transition of logical elements to physical level](#)
 - [ARCH.932 Define the subsystem boundaries](#)
 - [ARCH.090 Identify alternative subsystem boundary options](#)
 - [ARCH.097 Evaluate subsystem boundary options against the architectural tradeoff criteria](#)
 - [ARCH.098 Deploy logical components to subsystems](#)
 - [ARCH.116 Define the subsystems](#)
 - [ARCH.117 Adopt subsystem definition from collaborative project](#)
 - [ARCH.933 Define the subsystem interfaces](#)
 - [ARCH.156 Define single inter-subsystem interface](#)
 - [ARCH.151 Refine system interface to subsystem-actor interface](#)
 - [ARCH.114 Define single interface layer](#)
 - [ARCH.128 Define functions of single interface layer](#)
 - [ARCH.129 Define data models for interface layers](#)
 - [ARCH.130 Define interface layer scenarios](#)
 - [ARCH.122 Define behaviour for interface layer functions](#)
 - [ARCH.186 Consolidate single subsystem interface](#)
 - [ARCH.934 Define supporting physical architecture](#)
 - [ARCH.125 Define structure of communication assets](#)
 - [ARCH.126 Define structure of computation assets](#)
 - [ARCH.127 Define location kinds](#)
 - [ARCH.935 Assess risks associated with physical architecture](#)
 - [ARCH.101 Carry out failure modes, effects and criticality analysis \(FMECA\) on physical functional chains](#)
 - [ARCH.103 Identify new physical functions needed for robustness to failures](#)
 - [ARCH.104 Identify new non-functional requirements needed for robustness to failures](#)
 - [ARCH.105 Identify new non-functional requirements needed to mitigate hazards introduced by physical architecture choice](#)
 - [ARCH.106 Identify new physical functions needed for mitigating hazards introduced by physical architecture choice](#)
 - [ARCH.135 Carry out HAZID on tenderable elements](#)
 - [ARCH.936 Allocation/Derivation of NFRs](#)
 - [ARCH.131 Derive & allocate NFRs to location kinds](#)
 - [ARCH.132 Derive & allocate NFRs to communication assets](#)
 - [ARCH.133 Derive & allocate NFRs to computation assets](#)
 - [ARCH.137 Allocate interface NFRs to interface layers/interface functions](#)
 - [ARCH.155 Allocate hazard mitigation NFRs to subsystems](#)
 - [ARCH.938 Consolidate the overall architecture](#)
 - [ARCH.192 Refine physical capability realisations](#)
 - [ARCH.176 Create subsystem exchange scenarios](#)
 - [ARCH.136 Push back chosen physical architecture from child models to parent model](#)
 - [ARCH.169 Consolidate traceability between model elements at physical level and model elements at logical level](#)
 - [ARCH.168 Consolidate the overall generic physical architecture](#)
 - [ARCH.100 Obtain stakeholder acceptance of preferred architecture](#)
- [ARCH.925 Transition from parent model to child models](#)

- [ARCH.112 Execute transition from logical architecture of parent model to child model system architecture](#)
-

SETT Status:

- [AMOD-093 Subsystems definition](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
- [ARCH.001 Define boundary of wider system of interest \(operating entity of SysABB\)](#)
 - [arch_process](#)
 - [arch_safe_enough_to_try](#)
- [ARCH.002 Create initial set of operational capabilities](#)
 - [arch_process](#)
 - [arch_safe_enough_to_try](#)
- [ARCH.003 Create initial set of operational actors & operational entities](#)
 - [arch_safe_enough_to_try](#)
- [ARCH.004 Analyse trade space factors](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [operational-layer](#)
- [ARCH.005 Define set of system of interest lifecycle variants](#)
 - [arch_process](#)
 - [arch_safe_enough_to_try](#)
- [ARCH.006 Create initial set of operational states for operational entities/actors](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
- [ARCH.007 Create set of initial operational activities](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
- [ARCH.008 Define abstract concepts](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
- [ARCH.009 Define measures of effectiveness](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
- [ARCH.012 Determine measure of effectiveness target values](#)
 - [arch_process](#)
 - [arch_safe_enough_to_try](#)
 - [arch_risk_step](#)
 - [operational-layer](#)
- [ARCH.023 Identify relevant safety legislation and regulation](#)
 - [arch_process](#)
 - [arch_safe_enough_to_try](#)
 - [arch_risk_step](#)
 - [operational-layer](#)
-

ARCH.028 Define operational safety measures

- [per_operational_capability](#)
- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [arch_risk_step](#)
- [operational-layer](#)

•

ARCH.031 Create operational scenario with risk control measures

- [per_operational_capability](#)
- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [arch_risk_step](#)
- [operational-layer](#)

•

ARCH.032 Create operational process with risk control measures

- [per_operational_capability](#)
- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [arch_risk_step](#)
- [operational-layer](#)

•

ARCH.033 Update operational activities and interactions

- [per_operational_capability](#)
- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [arch_risk_step](#)
- [operational-layer](#)

•

ARCH.034 Produce concept of operations (CONOPS)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [operational-layer](#)

•

ARCH.035 Identify constraints on the system solution

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [system-layer](#)

•

ARCH.036 Identify all alternative system boundaries

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [system-layer](#)

•

ARCH.039 Define set of potential system and actor functions

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [system-layer](#)
- [rca_pro](#)

•

ARCH.040 yyTransfer OA to system level function, allocate entirely to actor

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [system-layer](#)

•

ARCH.041 Transfer or trace upper level model elements to system level

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [system-layer](#)

- [ARCH.042 Select system boundary \(or set of variants\)](#)
 - [arch_process](#)
 - [arch_safe_enough_to_try](#)
 - [rca_pro](#)
 - [system-layer](#)
- [ARCH.043 Define set of potential system actors](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [system-layer](#)
 - [rca_pro](#)
- [ARCH.044 Define requirements from non-actor stakeholders](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)
- [ARCH.045 Define complete set of system-level actors](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)
- [ARCH.048 Identify system implementation constraints](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)
- [ARCH.049 Define set of potential system capabilities/missions](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [system-layer](#)
 - [rca_pro](#)
- [ARCH.051 Align the system boundary, capabilities and actors with collaboration projects](#)
 - [arch_safe_enough_to_try](#)
 - [rca_pro](#)
 - [system-layer](#)
 - [arch_process](#)
- [ARCH.052 Create initial system exchange scenarios](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)
- [ARCH.053 Create initial system functional chains](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)
- [ARCH.054 Model data flowing between system functions](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)

- - [ARCH.055 Model states on system level](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)
- - [ARCH.056 Map system functionality to states](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)
- - [ARCH.057 Model non-payload data on external interfaces](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [arch_safe_enough_to_try](#)
 - [optional](#)
 - [system-layer](#)
- - [ARCH.058 Define measures of performance](#)
 - [arch_process](#)
 - [arch_safe_enough_to_try](#)
 - [rca_pro](#)
 - [system-layer](#)
- - [ARCH.078 Create bare business operational scenario](#)
 - [per_operational_capability](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
- - [ARCH.079 Create bare business operational process](#)
 - [per_operational_capability](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
- - [ARCH.080 Model operational activities and interactions](#)
 - [per_operational_capability](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
- - [ARCH.081 Update constraints on system solution](#)
 - [arch_process](#)
 - [arch_safe_enough_to_try](#)
 - [rca_pro](#)
 - [system-layer](#)
- - [ARCH.082 Trade off system requirements and constraints](#)
 - [arch_process](#)
 - [arch_safe_enough_to_try](#)
 - [rca_pro](#)
 - [system-layer](#)
- - [ARCH.083 Agree the system requirements with stakeholders](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)
 - [arch_safe_enough_to_try](#)
- - [ARCH.084 Execute automatic transition of system elements to logical level](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.085 Create or update functional chains at logical level](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.086 Create or update exchange scenarios at logical level](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.087 Update logical data model](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.088 Define system functions and functional exchanges](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [system-layer](#)

• [ARCH.090 Identify alternative subsystem boundary options](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [physical-layer](#)

• [ARCH.096 Allocate logical functions to logical components \(including alternative allocations\)](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.097 Evaluate subsystem boundary options against the architectural tradeoff criteria](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [physical-layer](#)

• [ARCH.098 Deploy logical components to subsystems](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [physical-layer](#)

• [ARCH.099 Define behaviour for logical functions](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.109 Consolidate traceability between model elements at system level and model elements at operational level](#)

- [arch_process](#)
- [system-layer](#)
- [arch_safe_enough_to_try](#)

• [ARCH.111 Execute automatic transition of logical elements to physical level](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [physical-layer](#)

• [ARCH.114 Define single interface layer](#)

- [arch_process](#)
- [rca_pro](#)
- [physical-layer](#)
- [arch_safe_enough_to_try](#)

• [ARCH.116 Define the subsystems](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [physical-layer](#)

• [ARCH.117 Adopt subsystem definition from collaborative project](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [physical-layer](#)

• [ARCH.119 Define consolidated set of system capabilities/missions](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [system-layer](#)

• [ARCH.121 Consolidate operational activities and interactions](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [operational-layer](#)

• [ARCH.140 Consolidate the allocation of operational activities](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)

• [ARCH.141 Update system functions with risk control measures](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [system-layer](#)
- [arch_risk_step](#)

• [ARCH.142 Create system exchange scenarios with risk control measures](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [system-layer](#)
- [arch_risk_step](#)

• [ARCH.143 Create system functional chains with risk control measures](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [system-layer](#)
- [arch_risk_step](#)

- - [ARCH.144 Define enterprise goals](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
- - [ARCH.147 Update operational data](#)
 - [per_operational_capability](#)
 - [arch_process](#)
 - [arch_safe_enough_to_try](#)
 - [arch_risk_step](#)
 - [operational-layer](#)
- - [ARCH.149 Update system data with risk control measures](#)
 - [rca](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [system-layer](#)
 - [arch_risk_step](#)
- - [ARCH.150 Update system interface model with risk control measures](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [system-layer](#)
 - [arch_risk_step](#)
- - [ARCH.151 Refine system interface to subsystem-actor interface](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [physical-layer](#)
- - [ARCH.152 Consolidate system functionality](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)
- - [ARCH.153 Model operational data](#)
 - [per_operational_capability](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
- - [ARCH.154 Consolidate logical functional flow](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [logical-layer](#)
- - [ARCH.156 Define single inter-subsystem interface](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [physical-layer](#)
 - [arch_safe_enough_to_try](#)
- - [ARCH.157 Consolidate system data](#)
 - [arch_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [system-layer](#)
-

[ARCH.158 Model external interface layers](#)

- [arch safe enough to try](#)
- [arch_process](#)
- [rca_pro](#)
- [optional](#)
- [system-layer](#)

•

[ARCH.159 Model operational states](#)

- [per_operational_capability](#)
- [arch safe enough to try](#)
- [arch_process](#)

•

[ARCH.160 Update operational states](#)

- [per_operational_capability](#)
- [arch_process](#)
- [arch safe enough to try](#)
- [arch_risk_step](#)
- [operational-layer](#)

•

[ARCH.161 Map operational activities to operational states](#)

- [per_operational_capability](#)
- [arch safe enough to try](#)
- [arch_process](#)

•

[ARCH.162 Update operational activity to state mapping](#)

- [per_operational_capability](#)
- [arch_process](#)
- [arch safe enough to try](#)
- [arch_risk_step](#)
- [operational-layer](#)

•

[ARCH.163 Update system states](#)

- [arch_process](#)
- [arch safe enough to try](#)
- [system-layer](#)
- [optional](#)
- [arch_risk_step](#)

•

[ARCH.164 Update function mapping to system states](#)

- [arch safe enough to try](#)
- [arch_process](#)
- [system-layer](#)
- [optional](#)
- [arch_risk_step](#)

•

[ARCH.169 Consolidate traceability between model elements at physical level and model elements at logical level](#)

- [arch_process](#)
- [rca_pro](#)
- [physical-layer](#)
- [arch safe enough to try](#)

•

[ARCH.177 Complete the definition of the operational capability of interest](#)

- [per_operational_capability](#)
- [arch safe enough to try](#)
- [arch_process](#)

•

[ARCH.178 Update dependencies and pre/post conditions](#)

- [per_operational_capability](#)
- [arch safe enough to try](#)
- [arch_process](#)

- [arch_risk_step](#)
- [operational-layer](#)

• [ARCH.179 Complete the definition of the system capability of interest](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [system-layer](#)

• [ARCH.180 Define logical component candidates](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.181 Reconcile candidate logical components](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.182 Split the system functions](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.183 Consolidate operational data](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)

• [ARCH.186 Consolidate single subsystem interface](#)

- [arch_process](#)
- [rca_pro](#)
- [physical-layer](#)
- [arch_safe_enough_to_try](#)

• [ARCH.187 Adopt logical component definitions from collaboration projects](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [logical-layer](#)
- [optional](#)

• [ARCH.190 Consolidate logical data](#)

- [arch_process](#)
- [rca_pro](#)
- [arch_safe_enough_to_try](#)
- [logical-layer](#)

• [ARCH.191 Refine logical capability realisations](#)

- [arch_safe_enough_to_try](#)
- [arch_process](#)
- [optional](#)
- [logical-layer](#)
- [rca_pro](#)

• [ARCH.192 Refine physical capability realisations](#)

- [arch_process](#)
- [arch_safe_enough_to_try](#)
- [rca_pro](#)

- [optional](#)
-

Not SETT Status:

- [AMOD-004 Fault tree \(per operational deviation\)](#)
 - [arch_process](#)
 - [arch_not_safe_enough_to_try](#)
- [AMOD-146 System needs and constraints tradeoff decision record](#)
 - [rca](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
- [ARCH.013 Identify risks to business effectiveness \(business loss-risk mapping\)](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [arch_risk_step](#)
 - [operational-layer](#)
- [ARCH.014 Evaluate risks to business effectiveness](#)
 - [per_operational_capability](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
- [ARCH.015 Define business risk control measures](#)
 - [per_operational_capability](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
- [ARCH.018 Identify internal & external issues affecting successful security](#)
 - [arch_process](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_risk_step](#)
 - [operational-layer](#)
- [ARCH.019 Determine need for information availability, confidentiality, integrity](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [operational-layer](#)
 - [arch_risk_step](#)
- [ARCH.020 Identify & classify operational deviations](#)
 - [per_operational_capability](#)
 - [arch_process](#)
 - [operational-layer](#)
 - [arch_risk_step](#)
 - [arch_not_safe_enough_to_try](#)
- [ARCH.021 Evaluate security risks](#)
 - [per_operational_capability](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
- [ARCH.022 Define operational security measures](#)
 - [per_operational_capability](#)
 - [arch_process](#)
 - [arch_not_safe_enough_to_try](#)
-

[ARCH.025 Create foundation for safety risk model](#)

- [arch_process](#)
- [arch_not_safe_enough_to_try](#)
- [arch_risk_step](#)
- [operational-layer](#)

• [ARCH.026 Define the state model of accidents, hazardous and safe state](#)

- [arch_process](#)
- [arch_risk_step](#)
- [operational-layer](#)
- [arch_not_safe_enough_to_try](#)

• [ARCH.027 Evaluate operational safety risks](#)

- [per_operational_capability](#)
- [arch_process](#)
- [arch_risk_step](#)
- [operational-layer](#)
- [arch_not_safe_enough_to_try](#)

• [ARCH.029 Allocate risk control responsibilities \(op. activity to op. entity/actor\)](#)

- [per_operational_capability](#)
- [arch_process](#)
- [arch_not_safe_enough_to_try](#)
- [operational-layer](#)
- [arch_risk_step](#)

• [ARCH.061 Produce the concept of employment \(CONEMP\)](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [system-layer](#)

• [ARCH.062 Produce the concept of use \(CONUSE\)](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [system-layer](#)

• [ARCH.066 Identify system level deviations](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [system-layer](#)
- [arch_risk_step](#)

• [ARCH.067 Populate the fault tree](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [system-layer](#)
- [arch_risk_step](#)

• [ARCH.068 Assess deviation probability \(external constraints\)](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [system-layer](#)
- [arch_risk_step](#)

• [ARCH.069 Add system level risk measures](#)

- [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [system-layer](#)
 - [arch_risk_step](#)
-

[ARCH.070 Calculate system deviation probability](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [system-layer](#)

• [ARCH.071 Identify security risks introduced by chosen system boundary](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)

• [ARCH.072 Evaluate security risks introduced by chosen system boundary](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)

• [ARCH.073 Define system security measures](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)

• [ARCH.074 Allocate system security measures](#)

- [arch_process](#)
- [arch_not_safe_enough_to_try](#)

• [ARCH.076 Identify risks to business effectiveness introduced by chosen system boundary](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)

• [ARCH.077 Evaluate risks to business effectiveness introduced by chosen system boundary](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)

• [ARCH.089 Define architecture tradeoff criteria and weighting](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)

• [ARCH.100 Obtain stakeholder acceptance of preferred architecture](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [physical-layer](#)
- [rca_pro](#)

• [ARCH.101 Carry out failure modes, effects and criticality analysis \(FMECA\) on physical functional chains](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [arch_risk_step](#)
- [physical-layer](#)

• [ARCH.103 Identify new physical functions needed for robustness to failures](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [arch_risk_step](#)
- [physical-layer](#)

• [ARCH.104 Identify new non-functional requirements needed for robustness to failures](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [arch_risk_step](#)
- [physical-layer](#)

• [ARCH.105 Identify new non-functional requirements needed to mitigate hazards introduced by physical architecture choice](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)

- [arch_risk_step](#)
- [physical-layer](#)

• [ARCH.106 Identify new physical functions needed for mitigating hazards introduced by physical architecture choice](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [arch_risk_step](#)
- [physical-layer](#)

• [ARCH.107 Apportion non-functional requirements to logical functions](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.108 Define acceptance criteria for non-functional requirements](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.110 Consolidate traceability between model elements at logical level and model elements at system level](#)

- [arch_process](#)
- [arch_not_safe_enough_to_try](#)
- [rca_pro](#)
- [logical-layer](#)

• [ARCH.113 Transfer logical data to physical level](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)

• [ARCH.122 Define behaviour for interface layer functions](#)

- [arch_process](#)
- [arch_not_safe_enough_to_try](#)
- [rca_pro](#)
- [physical-layer](#)

• [ARCH.125 Define structure of communication assets](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [physical-layer](#)
- [rca_pro](#)

• [ARCH.126 Define structure of computation assets](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [physical-layer](#)

• [ARCH.127 Define location kinds](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [physical-layer](#)

• [ARCH.128 Define functions of single interface layer](#)

- [arch_not_safe_enough_to_try](#)
- [arch_process](#)
- [rca_pro](#)
- [physical-layer](#)

- - [ARCH.129 Define data models for interface layers](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [physical-layer](#)
- - [ARCH.130 Define interface layer scenarios](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [physical-layer](#)
- - [ARCH.131 Derive & allocate NFRs to location kinds](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [physical-layer](#)
- - [ARCH.132 Derive & allocate NFRs to communication assets](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
- - [ARCH.133 Derive & allocate NFRs to computation assets](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [physical-layer](#)
- - [ARCH.135 Carry out HAZID on tenderable elements](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [physical-layer](#)
 - [arch_risk_step](#)
- - [ARCH.137 Allocate interface NFRs to interface layers/interface functions](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [physical-layer](#)
 - [rca_pro](#)
- - [ARCH.138 Define system business risk control measures](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
- - [ARCH.139 Allocate system business risk control measures](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
- - [ARCH.155 Allocate hazard mitigation NFRs to subsystems](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [physical-layer](#)
- - [ARCH.168 Consolidate the overall generic physical architecture](#)
 - [rca_pro](#)
 - [arch_process](#)
 - [arch_not_safe_enough_to_try](#)
 - [physical-layer](#)

- - [ARCH.170 Identify security losses and threats](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
 - [arch_risk_step](#)
 - [operational-layer](#)
- - [ARCH.175 Produce consolidated operational deviation analysis report](#)
 - [arch_process](#)
 - [arch_not_safe_enough_to_try](#)
 - [operational-layer](#)
 - [arch_risk_step](#)
- - [ARCH.176 Create subsystem exchange scenarios](#)
 - [arch_process](#)
 - [rca_pro](#)
 - [physical-layer](#)
 - [arch_not_safe_enough_to_try](#)
- - [ARCH.189 Create templates and static text for CONOPS, CONUSE and CONEMP](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)
- - [ARCH.914 Produce the unregulated system documents](#)
 - [arch_not_safe_enough_to_try](#)
 - [arch_process](#)

ARCH.900 Determine the operational needs

- [ARCH.901 Capture existing and initial operational understanding](#)
 - [ARCH.001 Define boundary of wider system of interest \(operating entity of SysABB\)](#)
 - [ARCH.004 Analyse trade space factors](#)
 - [ARCH.005 Define set of system of interest lifecycle variants](#)
 - [ARCH.002 Create initial set of operational capabilities](#)
 - [ARCH.144 Define enterprise goals](#)
 - [ARCH.003 Create initial set of operational actors & operational entities](#)
 - [ARCH.189 Create templates and static text for CONOPS, CONUSE and CONEMP](#)
 - [ARCH.006 Create initial set of operational states for operational entities/actors](#)
 - [ARCH.007 Create set of initial operational activities](#)
 - [ARCH.008 Define abstract concepts](#)
 - [ARCH.009 Define measures of effectiveness](#)
- [ARCH.902 Analyse the operational capabilities to determine detailed operational needs](#)
 - [ARCH.177 Complete the definition of the operational capability of interest](#)
 - [ARCH.078 Create bare business operational scenario](#)
 - [ARCH.079 Create bare business operational process](#)
 - [ARCH.080 Model operational activities and interactions](#)
 - [ARCH.153 Model operational data](#)
 - [ARCH.159 Model operational states](#)
 - [ARCH.161 Map operational activities to operational states](#)
- [ARCH.903 Assess & mitigate the operational risks](#)
 - [ARCH.171 Prepare business risk analysis](#)
 - [ARCH.012 Determine measure of effectiveness target values](#)
 - [ARCH.013 Identify risks to business effectiveness \(business loss-risk mapping\)](#)
 - [ARCH.172 Prepare security risk analysis](#)
 - [ARCH.018 Identify internal & external issues affecting successful security](#)
 - [ARCH.019 Determine need for information availability, confidentiality, integrity](#)
 - [ARCH.170 Identify security losses and threats](#)
 - [ARCH.173 Prepare safety risk analysis](#)
 - [ARCH.023 Identify relevant safety legislation and regulation](#)
 - [ARCH.025 Create foundation for safety risk model](#)
 - [Draft foundation accidents](#)
 - [Draft foundation hazard](#)
 - [Draft foundation control measure](#)
 - [ARCH.026 Define the state model of accidents, hazardous and safe state](#)
 - [ARCH.020 Identify & classify operational deviations](#)
 - [ARCH.011 Assess operational performance risks of scenario](#)
 - [ARCH.014 Evaluate risks to business effectiveness](#)
 - [ARCH.015 Define business risk control measures](#)
 - [ARCH.017 Assess security risks of scenario](#)
 - [ARCH.021 Evaluate security risks](#)
 - [ARCH.022 Define operational security measures](#)
 - [ARCH.024 Assess operational safety risks of scenario](#)
 - [ARCH.027 Evaluate operational safety risks](#)
 - [ARCH.028 Define operational safety measures](#)
 - [ARCH.029 Allocate risk control responsibilities \(op. activity to op. entity/actor\)](#)
 - [ARCH.175 Produce consolidated operational deviation analysis report](#)
- [ARCH.904 Incorporate risk control measures in the operational processes](#)
 - [ARCH.031 Create operational scenario with risk control measures](#)
 - [ARCH.032 Create operational process with risk control measures](#)
 - [ARCH.033 Update operational activities and interactions](#)
 - [ARCH.147 Update operational data](#)
 - [ARCH.160 Update operational states](#)
 - [ARCH.162 Update operational activity to state mapping](#)
 - [ARCH.178 Update dependencies and pre/post conditions](#)
- [ARCH.905 Consolidate the operational needs](#)

- [ARCH.121 Consolidate operational activities and interactions](#)
- [ARCH.034 Produce concept of operations \(CONOPS\)](#)
- [ARCH.140 Consolidate the allocation of operational activities](#)
- [ARCH.183 Consolidate operational data](#)

ARCH.901 Capture existing and initial operational understanding

- [ARCH.001 Define boundary of wider system of interest \(operating entity of SysABB\)](#)
- [ARCH.004 Analyse trade space factors](#)
- [ARCH.005 Define set of system of interest lifecycle variants](#)
- [ARCH.002 Create initial set of operational capabilities](#)
- [ARCH.144 Define enterprise goals](#)
- [ARCH.003 Create initial set of operational actors & operational entities](#)
- [ARCH.189 Create templates and static text for CONOPS, CONUSE and CONEMP](#)
- [ARCH.006 Create initial set of operational states for operational entities/actors](#)
- [ARCH.007 Create set of initial operational activities](#)
- [ARCH.008 Define abstract concepts](#)
- [ARCH.009 Define measures of effectiveness](#)

ARCH.001 Define boundary of wider system of interest (operating entity of SysABB)

Goal	Identify the scope of the enterprise (the operating organisation of SysABB) Identify the wider environment, environment and wider system of interest.
Requirements met by this process step	CSM-SMS Guidance 1.1a) EN 50126-1 7.2.2 a) CSM-SMS Guidance 1.1f) ISO 27001 4.3
Inputs	Knowledge of workshop participants Baseline 6 actor definitions
Outputs	Initial version of AMOD-022 Operational entity & actor definition
Methodology	Any brainstorming technique to identify, initially, the scope of the enterprise and the levels of stakeholders outside the enterprise. Discussion of the precise scope of the enterprise in terms of what value is required outside the enterprise - which business organisation is the right level to define as WSOI in order to provide all the value to externals that will ultimately be needed from SysABB.
Tools and non-human resources	General brainstorming materials; Team for Capella
Cardinality	One-off, with possible revisions later.
Completion criteria	The definition of enterprise and actors is safe enough to try. All review feedback has been addressed.
Design review	ARCH.R.1
Step done by (Responsible)	Operational concept architect
Provides input to/assists (Contributes)	Functional architects Operations domain experts SEA Product manager
Uses outputs (Informed)	None directly

ARCH.004 Analyse trade space factors

Goal	Identify and characterise the factors that will dominate tradeoff decisions on operational and system architecture.
Requirements met by this process step	EN 50126-1 7.2.2 b) ISO 15288 6.4.1.3 b 1)
Inputs	Consultation with a small set of key stakeholders Existing strategic documentation Knowledge of the person doing the work
Outputs	AMOD-010 Trade space assessment
Methodology	<p>This task is mostly a case of the person responsible using their own knowledge and some research or stakeholder consultation to explore a standard range of factors to determine which of them is particularly dominant in this current project.</p> <p>Mnemonics such as TEPIDOIL or PESTEL are used in defence systems engineering to prompt the engineer to consider relevant factors. Tutorial material on these mnemonics can be found by searching them on the Internet or consulting a systems engineering textbook.</p> <p>For each heading in the mnemonic, find out</p> <ul style="list-style-type: none"> ▪ if this factor is at all relevant to DBB ▪ in what ways this factor manifests for DBB - identify specific factors under this theme <p>Then attempt to rank the factors according to their relative dominance in architectural decision-making for DBB/SysABB.</p> <p>Document the findings of the analysis in plain English.</p>
Tools and non-human resources	<p>Systems Engineering literature</p> <p>Confluence (note: an area for this, and a ticket SM-2557 - Carry out and document PESTEL constraint analysis for section on operational constraints FINISHED, have already been set up)</p>
Cardinality	One-off, with possibility for later revisions.
Completion criteria	<p>The analysis results are judged to be reasonable, accurate and relevant to DBB.</p> <p>All review feedback has been addressed.</p>
Design review	
Step done by (Responsible)	Operational concept architect

Provides input to /assists (Contributes)	Product Owner
Uses outputs (Informed)	None directly

ARCH.005 Define set of system of interest lifecycle variants

Goal	Achieve and document a shared understanding of the system's possible lifecycles.
Requirements met by this process step	ISO 15288 6.4.1.3 c 1) EN 50126-1 7.3.2 a)
Inputs	Briefing material from around the business Knowledge of contributors
Outputs	AMOD-012 System lifecycle model
Methodology	<p>Two aspects are to be examined:</p> <ul style="list-style-type: none"> • the basic shape of the lifecycle of any system of interest (e.g. SysDBS) instance, regardless of its individual capabilities - each phase of the lifecycle may expose specific requirements for the system's design; • the progression of possible migration states - there is a finite set of migration paths and these need to be identified, because each may expose specific requirements for the system's design. <p>In theory these aspects can be modelled together, but it may be simpler to separate them if possible.</p> <p>From the business briefing materials, a proposed states model should be created.</p> <p>This should then be reviewed with contributors and adjusted according to their knowledge.</p> <p>The results should be drawn up into one or more Capella state diagrams.</p>
Tools and non-human resources	Team for Capella
Cardinality	One-off, with possible revisions later
Completion criteria	Contributors & reviewers agree the lifecycle model(s) are safe enough to try. All review feedback has been addressed.
Design review	ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	Operational concept architect
Provides input to /assists (Contributes)	External stakeholders concerned with system lifecycle activities
Uses outputs (Informed)	<p>All activities where system capabilities or functions are defined, i.e.</p> <p>ARCH.091 Transfer selected operational capabilities to system level as system capabilities</p> <p>ARCH.092 Transfer selected operational activities to system level as system capabilities</p> <p>ARCH.093 Transfer selected operational activities to system level as system functions</p> <p>ARCH.094 XXCreate missions at system level</p> <p>ARCH.913 Analyse the system capabilities to determine detailed system needs</p>

ARCH.002 Create initial set of operational capabilities

Goal	Identify enough operational capabilities to provide a starting point for understanding the problem space
Requirements met by this process step	CSM-SMS guidance 1.1a) EN 50126-1 7.2.2 a) ISO 15288 6.4.1.3 b 2)
Inputs	Business briefing material Knowledge of contributors AMOD-022 Enterprise & environment definition
Outputs	AMOD-021 Operational capabilities definition
Methodology	Brainstorming using any practical method to identify candidates, then rationalise by agreeing a reasonable level and size of item. Model the capabilities in Capella. Multiple diagrams can be created to implement view AMOD-021 , to keep diagram size acceptable. Diagrams could group capabilities on a similar theme. Capabilities may appear on multiple diagrams if they cross multiple themes. Note: it is not necessary to identify all dependencies/relationships between operational capabilities, nor to define very precise and synchronised start/end conditions at this stage. However, there should be some idea of these properties so that they can be used as a starting point for operational scenario analysis. They will be updated later in the process as more is learned about the capability set and the operational needs.
Tools	Any practical brainstorming tools + appropriate environment Team for Capella
Cardinality	One-off at the beginning of the project. Revisions allowed to the whole set based on what is learned during analysis of the individual capabilities.
Completion criteria	The participants in brainstorming agree that a reasonable level of coverage has been achieved; The output model views comply with their model specification and modelling rules; The accountable role judges that a reasonable level of coverage has been achieved.
Design review	ARCH.R.1 Operational capability review ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	Operational concept architect
Provides input to /assists (Contributes)	Lead system architect SEA product manager Functional architects Systems engineers Operations domain experts

**Uses outputs
(Informed)**

None directly

ARCH.144 Define enterprise goals

[SM-3118](#) - Create Confluence page for activity definition ARCH.144 **FINISHED**

Goal	Identify the goals that the WSOI is attempting to achieve Provide business-level basis for the need for the SOI
Requirements met by this process step	None identified
Inputs	DBB promotional material (initial slides) Starke Schiene promotional material
Outputs	AMOD-106 Enterprise goals
Methodology	Review the initial business promotional material for DSD and extract concrete targets and their related metrics. For example, "DSD will increase capacity by 35%" results in: Enterprise: Railway operation in Germany Enterprise Goal: increase capacity by 35% Metric: capacity Use the relationships defined in the viewpoint definition to link up these elements. Model goals at the level of the German transport enterprise, the German railway enterprise, and the rail infrastructure enterprise.
Tools and non-human resources	Any drawing tool
Cardinality	One-off with revisions permitted
Completion criteria	The output view conforms to the viewpoint specification The content of the output view is agreed by top management of DSD as being an accurate representation of the current known enterprise goals
Design review	
Step done by (Responsible)	Operational concept architect
Provides input to/assists (Contributes)	Product owner Top management of DSD
Uses outputs (Informed)	None directly

ARCH.003 Create initial set of operational actors & operational entities

Goal	Identify and define the actors and entities at operational level who need value from the organisation operating SysABB.
Requirements met by this process step	CSM-SMS Guidance 1.1a) EN 50126-1 7.2.2 a) ISO 27001 4.2
Inputs	AMOD-022 Operational entity & actor definition Knowledge of workshop contributors
Outputs	AMOD-022 Operational entity & actor definition (updated)
Methodology	General brainstorming techniques to identify, evaluate and refine actors and entities. Classify each actor/entity according to the onion-model level to which it belongs
Tools and non-human resources	General brainstorming materials; Team for Capella
Cardinality	One-off, with possible revisions later.
Completion criteria	The definition and levelling (wider environment, environment, wider system of interest) of actors and entities is safe enough to try. All review feedback has been addressed.
Design review	
Step done by (Responsible)	Operational concept architect
Provides input to/assists (Contributes)	SEA product manager Lead system architect Functional architects Operational domain experts
Uses outputs (Informed)	None directly

ARCH.189 Create templates and static text for CONOPS, CONUSE and CONEMP

Goal	Communicate operational and system needs effectively to stakeholders at a high, non-normative level
Requirements met by this process step	Contributes to ISO 15288 6.4.2.3 b) 1)
Inputs	<p>relevant standards for the format and content of the documents - suggestions include</p> <p>ISO 29148</p> <p>NASA systems engineering handbook</p> <p>Transport for New South Wales standard T MU AM 06008 ST</p> <p>Examples of CONOPS, CONUSE and CONEMP documents from industry, including</p> <ul style="list-style-type: none"> ▪ deep tube railway generic operations and maintenance concept 2020.pdf ▪ PPI-005606-4-Example-SSRC-CONUSE-141231.pdf
Outputs	<p>Templates, as Confluence pages, for</p> <ul style="list-style-type: none"> ▪ AMOD-037 DBS concept of operations ▪ AMOD-065 Concept of use ▪ AMOD-066 Concept of employment
Methodology	<p>Using the reference sources listed in the Inputs section, plus any other relevant sources, create Confluence heading structures for the three deliverables.</p> <p>Identify the need for any boilerplate (static, non-model-generated) text and populate these sections. For example, write the introduction and the instructions for reading.</p> <p>Identify the model queries needed to extract information from the model into sections that contain model content. Get help from the SET circle where needed.</p> <p>Review the boilerplate with appropriate stakeholders.</p> <p>Test the template with model data to ensure correct extraction.</p>
Tools and non-human resources	<p>Confluence</p> <p>Teams for Capella</p> <p>Jupyter and other plugins as specified by the SET circle</p>
Cardinality	One-off with revisions permitted.
Completion criteria	<p>Boilerplate sections are complete, reviewed and considered SETT</p> <p>Section structure of document is sufficient to meet the needs of the document's users</p> <p>Model queries have been set up and successfully tested</p>
Design review	ARCH.R.2 Operational review - consolidated

Step done by (Responsible)	Operational concepts architect
Provides input to /assists (Contributes)	Cross-cutting engineer
Uses outputs (Informed)	Equipment supplier Operational stakeholder

ARCH.006 Create initial set of operational states for operational entities/actors

Goal	Capture understanding of state-based differences in behaviour of operational entities/actors
Requirements met by this process step	CSM-RA Guidance 3.11 CSM-SMS Guidance 1.1 a) EN 50126-1 7.2.2 a)
Inputs	Knowledge of workshop participants AMOD-022 Operational entity & actor definition
Outputs	AMOD-023 Operational entity/actor states [WIP]
Methodology	<p>This activity does not need to be done for every operational entity and actor.</p> <p>It is only needed when</p> <ul style="list-style-type: none"> ■ there is a clear and relevant difference in the behaviour of an entity/actor, depending on its state, or ■ the behaviour of the wider system of interest is defined in terms of changing the state of an entity/actor. <p>In either case it should be fairly clear what the states are, based on knowledge of the modeller; if necessary, a brainstorming workshop can be held to identify candidates and come to an agreement.</p> <p>The results should be modelled as a states diagram in accordance with modelling rules.</p> <p>Where there are existing state definitions, but only a subset of them are of interest, then it is permissible to merge some states into a larger one. There is no real need to precisely follow an external definition if it is not helpful to do so.</p>
Tools and non-human resources	Standards and definitions belonging to particular entities
Cardinality	Once per operational entity, but only where necessary.
Completion criteria	There exists a model view (AMOD-023 Operational entity/actor states) for the entity/actor of interest, that is compliant with its definition and rules.
Design review	ARCH.R.1 Operational capability review ARCH.R.2 Operational review - consolidated

Step done by (Responsible)	Operational concept architect
Provides input to /assists (Contributes)	Domain experts
Uses outputs (Informed)	ARCH.010 Carry out initial operational scenario analysis for each operational capability

ARCH.007 Create set of initial operational activities

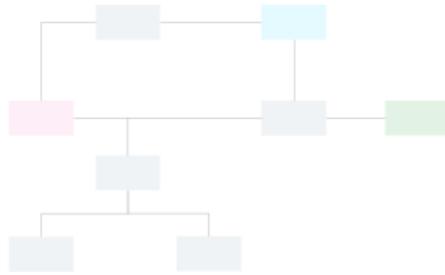
Goal	Capture existing content that could be candidate operational activities, to avoid unnecessary effort inventing new ones, and maintain consistency with business processes where appropriate
Requirements met by this process step	CSM-SMS Guidance 1.1 a), f) ISO 27001 4.3 RiL 114.0210 05 (6) EN 50126-1 7.2.2 a)
Inputs	DB Netz future E2E process map Knowledge of contributors AMOD-021 Operational capabilities definition
Outputs	AMOD-024 Operational activity definition and allocation
Methodology	Brainstorming using any practical method to identify candidates, then rationalise by agreeing a reasonable level and size of item. Allocate the activities to the operational entity or actor that is believed to carry them out. This activity should be a follow-up to ARCH.002 , to sift out candidates from that activity that fit better as operational activities, but have been initially defined as capability. The intention is not to spend a great deal of effort on an exhaustive analysis - we have further opportunities to capture operational activities - but rather to collect together any obvious candidate operational activities that we are already aware of, in one form or other, so that we are not forced to invent new items where we already had something.
Tools and non-human resources	Basic brainstorming materials Team for Capella
Cardinality	One-off (no revisions needed)
Completion criteria	Output view conforms to its modelling rules. The obvious sources of candidate operational activities have been examined and a reasonable set of candidates captured.
Step done by (Responsible)	Operational concept architect
Design review	

Provides input to /assists (Contributes)	Lead system architect SEA product manager Functional architects Systems engineers Operations domain experts
Uses outputs (Informed)	None directly

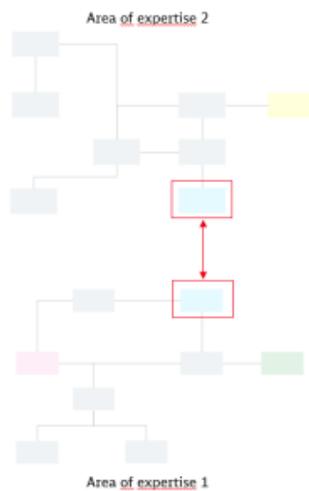
ARCH.008 Define abstract concepts

Goal	Identify the most important items of data that are known at the beginning of the process to be necessary for the achievement of the project
Requirements met by this process step	None.
Inputs	RCA domain knowledge model Knowledge of participants
Outputs	AMOD-025 Abstract concepts (limited scope)
Methodology	<p>Abstract concepts must be aligned with any international collaborations. Therefore, this task may be done jointly with other railway organisations if it makes sense to do the work once jointly and then adopt the outcome for ourselves.</p> <p>Using expert knowledge or existing models, capture model elements to represent abstract concepts.</p> <p>Abstract concepts are the basics of the way the railway works;</p> <p>Operationally, these objects exist even if they are not currently /typically transmitted by electronic communication.</p> <p>Two kind of abstract concept class can be defined:</p> <ul style="list-style-type: none">• Generic abstract concepts classes should be not implementation-specific, pass the "steam-train" test of technology independence and abstract enough that it complies to all architecture layers.• Solution abstract concept classes should be implementation-specific and implement at least one generic abstract concept class (e.g. in the RCA project a "Drive Protection Section" is an implementation of the generic abstract concept "Point" and "Level Crossing". <p>See as an addition the 5. Modelling Rules for Data.</p> <p><i>Guidance:</i></p> <ul style="list-style-type: none">• Check if an abstract concept which you would like to create already exist, if so please involve all persons of interest in your discussion.• After the relationships of the abstract concept are defined, please share the results with the group. <p><i>Way of working</i></p>

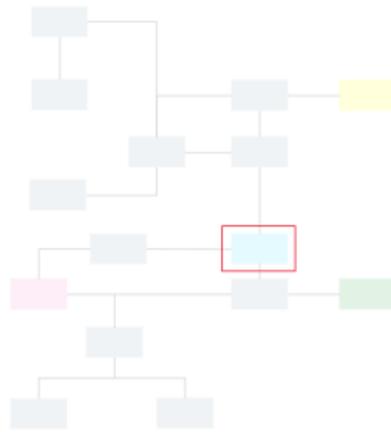
1. Each system architect per area of expertise shall elaborate their abstract concepts with identification of potential shared concepts with other area(s) of expertise



2. For each shared concept, discussion has to be made with the system architect of the other area of expertise. (Note: an abstract concept can also be shared with more than 2 area of expertise). If there is different definitions for that shared concept, then record them through rationale or comments in your modelling support. The rationales would help the Operational Concept Architect to understand the decision of having different definition and to avoid further discussion with those system architects



3. After discussion with the other area of expertise, a consolidation has to be done between the 2 ontologies of abstract concepts. This is a process that has to be done incrementally with the other areas of expertise (green, pink and yellow on the example)



Area of expertise 1 and 2 combined

4. Consolidation/review of all ontologies on a single project done by the Operational Concept Architect and the Cross cutting engineer to improve abstract concept definition. The shared concept can be grouped and definition could be refined. (See also [ARCH.183](#))



Pro tips:

- 2 representatives, system architect/system engineer or system architect/subsystem architect, of each area of expertise is enough for discussion
- Meeting of 2h is where you have the best results

Tools and non-human resources

Team for Capella

Cardinality

Once per area of expertise (limited scope) (this could be the area of a technical circle, or of an operational aspect; not necessary to capture everything exhaustively, just capture what is known at the start)

Completion criteria	Output model views conform to their modelling rules. All known initial sources of abstract concepts have been captured to a SETT level.
Design review	ARCH.R.1 Operational capability review ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	Operational concept architect
Provides input to /assists (Contributes)	Domain experts System architect
Uses outputs (Informed)	None directly

ARCH.009 Define measures of effectiveness

Goal	Achieve shared understanding of the quantitative measures that will be used to determine if the system is a success
Requirements met by this process step	ISO 29148 (2018) para 5.2.3 ISO 15288 (2015) para 6.4.1.3 a)
Inputs	AMOD-106 Enterprise goals AMOD-026 Service reliability/ availability targets Business performance indicators for DB business units
Outputs	AMOD-106 Enterprise goals (modified)
Methodology	<p>The measures of effectiveness indicate how well the railway infrastructure operations unit (WSOI) is delivering the business value (the effect) that it is required to. This is not the same as SysABB system performance. We define that later in a different task.</p> <p>Proposal: split the task in two stages.</p> <p>First, create a proposal within the DSD team.</p> <p>Second, take this proposal to business stakeholders (not restricted to DB, could also include stakeholders from external RUs for example), present and discuss, refine with feedback and agree.</p> <p>The proposal can be created by brainstorming a suitable set of performance indicators given the targets that are set for DBB to achieve. Reference should be made to DB's quality management system, particularly any regulations (Richtlinien) that define performance indicators, or define how performance indicators should be created.</p> <p>It may be possible to use an entirely existing set of effectiveness measures, but this should not be assumed.</p> <p>Further guidance</p> <p>The NASA Systems Engineering Handbook defines a measure of effectiveness as:</p> <p><i>A measure by which a stakeholder's expectations are judged in assessing satisfaction with products or systems produced and delivered in accordance with the associated technical effort. The MOE is deemed to be critical to not only the acceptability of the product by the stakeholder but also critical to operational/mission usage. A MOE is typically qualitative in nature or not able to be used directly as a design-to requirement.</i></p> <p>These measures must have meaning at the level of stakeholders outside the wider system of interest. This means they must be expressed in terms that are relevant to the stakeholders and their needs - and not in terms of performance of activities internal to the wider system of interest (since stakeholders neither know nor care about how the WSOI delivers, only that it does).</p>

i Example: passengers care about **journey time**, but they do not care how many seconds it takes to set a route for a train.

There already exists a set of measures that are used for quantifying the effectiveness of the railway as it currently operates. These should be elicited from knowledgeable stakeholders and assessed for their suitability as indicators in the future.

Further measures may be identified. New measures may be desirable to quantify effectiveness in areas that have traditionally not been formally monitored. However, the team must be mindful that adding large numbers of new additional measures could indicate that they are thinking on the level of system performance (see [ARCH.058 Define measures of performance](#)), rather than operational effectiveness, since the railway infrastructure has broadly the same operational capabilities now as it will in future.

Existing measures may be replaced with new ones where the new ones are considered better suited to future operation.

Tools and non-human resources	DB regulations DB quality management system, indicators currently in use
Cardinality	One-off with revisions permitted
Completion criteria	The set of measures of effectiveness is agreed between DSD and the wider business stakeholders. The set of measures of effectiveness is accepted by DSD top management. Measures of effectiveness are defined sufficiently to assure that they are observable and controllable by DBB.
Step done by (Responsible)	Operational concept architect
Design review	
Provides input to /assists (Contributes)	Lead system architect System architect Railway operations expert
Uses outputs (Informed)	None directly

ARCH.902 Analyse the operational capabilities to determine detailed operational needs

- [ARCH.177 Complete the definition of the operational capability of interest](#)
- [ARCH.078 Create bare business operational scenario](#)
- [ARCH.079 Create bare business operational process](#)
- [ARCH.080 Model operational activities and interactions](#)
- [ARCH.153 Model operational data](#)
- [ARCH.159 Model operational states](#)
- [ARCH.161 Map operational activities to operational states](#)

ARCH.177 Complete the definition of the operational capability of interest

[SM-4933](#) - Update confluence page for activity definition: ARCH.177 FINISHED

Goal	Visualise the E2E relationships between the OCs from the perspective of 1 OC, the Operational Capability of Interest Elaborate generic pre and post conditions
Requirements met by this process step	CSM-SMS guidance 1.1a) EN 50126-1 7.2.2 a) ISO 15288 6.4.1.3 b 2)
Inputs	AMOD-021 Operational capabilities definition
Outputs	AMOD-137 Single operational capability context
Methodology	<p>When AMOD-021 is done, the context of operational capability (COC) diagram can be created and Capella automatically gathers and displays all the relationships for the operational capability of interest. If there are entities /actors connected to the operational capability of interest, you can decide to remove those if this makes the diagram clearer.</p> <p>If you feel there is a relationship with another operational capability missing, add it in the diagram, choosing generalisation, <<extend>> or <<include>> according to the modelling rules, but don't overthink at this level: there will be a process step to update it anyway.</p> <p>By having dependencies between your operational capability of interest and the other operational capabilities, you can elaborate generic pre and post condition for your operational capability of interest. These can be further elaborated later when more is known (ARCH.178 Update dependencies and pre/post conditions).</p> <p>At this point, there is no need to align with other operational capability owners yet.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per Operational Capability
Completion criteria	The output views conform to their modelling rules.
Design review	ARCH.R.1 Operational feature review ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	System Architect
Provides input to /assists (Contributes)	Systems Engineer

**Uses outputs
(Informed)**

None

ARCH.078 Create bare business operational scenario

Goal	Visualise the initial understanding of business interactions between operational entities and actors in this scenario
Requirements met by this process step	CSM-SMS Guidance 1.1a) EN 50126-1 7.2.2 a) ISO 15288 6.4.2.3 c 1)
Inputs	AMOD-021 Operational capabilities definition AMOD-022 Operational entity & actor definition AMOD-023 Operational entity/actor states (where applicable) AMOD-024 Operational activity definition and allocation
Outputs	AMOD-027 Operational bare business scenario
Methodology	<p>If the person creating the output view knows enough about how the actors and entities work together for this capability, they can create the sequence diagram directly from their own knowledge and must only develop a view that complies with all the modelling rules.</p> <p>If, on the other hand, it is necessary to gather information from others, a brainstorming workshop should be held.</p> <p>If relevant scenarios are already defined at an international level (e.g. RCA), these should be referred to and reused if appropriate.</p> <p>Participants should be sufficient to represent knowledge/concerns of all the involved actors and activity areas of the wider system of interest..</p> <p>The brainstorming goal is to identify sufficient interactions to get from the starting conditions to the ending conditions.</p> <p>Where new operational activities or interactions are identified, these should be first created on the corresponding AMOD-028 Operational activities and interaction definitions (single operational capability).</p> <p>The operational entities and actors should be chosen with care. The entities "wider environment", "environment", and "wider system of interest/WSOI" should not normally be used - these entities exist to provide some categories for other entities and actors, not to be treated as a real interacting thing.</p> <p>Use the most specific possible operational entities given the known responsibilities. Where an operational entity represents an organisation, and there are some operational actors inside it, but it is not clear which of them has a particular involvement in a scenario, then it is appropriate to use the operational entity itself (this is analogous to saying "we know this organisational unit has a responsibility, but we don't know precisely which role within that unit is actually going to do it").</p>
Tools and non-human resources	Team for Capella (optional) General brainstorming equipment
Cardinality	Once per operational capability
Completion criteria	<p>The output view complies with its modelling rules.</p> <p>The content of the output view represents the team's current understanding of the way this operational capability is intended to be delivered.</p>

Step done by (Responsible)	System architect
Design review	ARCH.R.1 Operational capability review
Provides input to /assists (Contributes)	<p>Inclusive OR:</p> <ul style="list-style-type: none"> • Subsystem architect • Expert for GoA4 railway operation • Cross-cutting engineer • Engineer
Uses outputs (Informed)	<p>RAMS manager</p> <p>RAMS engineer</p> <p>Verification & validation manager</p>

ARCH.079 Create bare business operational process

Goal	Visualise the initial understanding of decision logic and activity flow for this operational capability.
Requirements met by this process step	CSM-SMS Guidance 1.1a) EN 50126-1 7.2.2 a) ISO 15288 6.4.2.3 c 1)
Inputs	AMOD-021 Initial operational capabilities AMOD-022 Operational entity & actor definition AMOD-024 Initial operational activity definition and allocation
Outputs	AMOD-029 Operational bare business process
Methodology	<p>If the person creating the output view knows enough about how the process flow works for this capability, they can create the operational process diagram directly from their own knowledge and must only develop a view that complies with all the modelling rules.</p> <p>If, on the other hand, it is necessary to gather information from others, a brainstorming workshop should be held.</p> <p>If relevant parts of operational processes are already defined at an international level (e.g. RCA), these should be referred to and reused if appropriate.</p> <p>Participants should be sufficient to represent knowledge/concerns of all the involved actors and activity areas of the wider system of interest.</p> <p>The brainstorming goal is to identify sufficient operational activity involvements, interactions and sequence links to get from the starting conditions to the ending conditions.</p> <p>Where new operational activities or interactions are identified, these should be first created on the corresponding AMOD-028 Operational activities and interaction definitions (single operational capability).</p>
Tools and non-human resources	Team for Capella

Cardinality	Once per operational capability
Completion criteria	<p>The output view complies with its modelling rules.</p> <p>The content of the output view represents the team's current understanding of the way this operational capability is intended to be delivered.</p>
Design review	ARCH.R.1 Operational capability review
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	<p>Inclusive OR:</p> <ul style="list-style-type: none"> • Subsystem architect • Expert for GoA4 railway operation • Cross-cutting engineer • Engineer
Uses outputs (Informed)	None directly

ARCH.080 Model operational activities and interactions

Goal	Visualise all the operational activities and interactions needed for one operational capability Capture new operational activities and interactions identified during scenario/process analysis
Requirements met by this process step	CSM-SMS Guidance 1.1a) EN 50126-1 7.2.2 a) ISO 15288 6.4.2.3 c 1)
Inputs	AMOD-024 Operational activity definition and allocation Pattern for defining operational activities
Outputs	AMOD-028 Operational activities and interaction definitions (single operational capability)
Methodology	<p>After an initial set of operational activities had been defined and allocated (as per AMOD-024), per capability a thorough analysis of operational activities (this activity), operational scenarios and operational processes (ARCH.078 and ARCH.079) is done. The recommended workflow is as follows:</p> <ul style="list-style-type: none"> • Define viewpoint instance: Per operational capability define one instance of AMOD-028 and populate the related operational activities that belong to the scope of that capability. • Define operational process: Inside the instance of AMOD-028 define the operational process and continue with the ARCH.079 and ARCH.078. • Iterate through these three activities as needed (ARCH.078, ARCH.079, ARCH.080). <p>Note: During the analysis of operational scenarios and processes, it may be necessary to create new operational activities and/or interactions. These should be created on AMOD-028.</p> <p>Note: Pattern for defining operational activities</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability.
Completion criteria	The output view conforms to its modelling rules. The set of operational activities and interactions is safe enough to try.
Design review	ARCH.R.1 Operational capability review ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Inclusive OR: <ul style="list-style-type: none"> • Subsystem architect • Expert for GoA4 railway operation • Cross-cutting engineer • Engineer
Uses outputs (Informed)	None directly

ARCH.153 Model operational data

[RCAMT-369](#) - Refine data model rules so that data can be reused on lower levels and not need to be automatic transferred **FINISHED**

Goal	Define and capture the specific information content needed for exchange between operational entities/actors
Requirements met by this process step	RiL 114.0210 05 (6) (partial) [ISO 27001]
Inputs	AMOD-027 Operational bare business scenario AMOD-028 Operational activities and interaction definitions (single operational capability) AMOD-029 Operational bare business process AMOD-025 Abstract concepts (limited scope)
Outputs	AMOD-025 Abstract concepts (limited scope updated) AMOD-105 Operational data objects [O.CDB] AMOD-110 Operational exchange items [O.CDB]

<p>Methodology</p>	<p>As an operational capability is built up as a process and as an exchange scenario, it is important to define the information content carried by interactions between operational entities and actors.</p> <p>Create the output diagrams as needed, following the 5. Modelling Rules for Data.</p> <p>In case of doubt, share tensions at the weekly modelling forum.</p> <p>Exchange items</p> <p>Check all instances of AMOD-110 Operational exchange items [O.CDB] for existing reusable exchange items, before creating new ones. It is not allowed to have identical data model elements more than once.</p> <p><i>Note:</i> New data objects and types should be modelled initially on AMOD-105 so that other modellers can look for them and reuse them where appropriate.</p> <p>Each interaction should have allocated at least one exchange item.</p> <p>The data information between two operational activities is permanently present at the operational level, therefore FLO should be used as communication mechanism for exchange items in all cases.</p> <p>Define for each exchange item the types of data, via the Exchange Item Elements. The information content of the exchange item should be defined as far as possible using an appropriate combination of data objects (simple types or structured types).</p> <p>Where the information content is unknown or unclear, write a text description on the exchange item, to act as a placeholder/STUB until the information content is resolved (communication mechanism of exchange item = UNSET).</p> <p>Data objects</p> <p>Check all instances of AMOD-105 Operational data objects [O.CDB] for existing reusable data objects and types, before creating new ones. It is not allowed to have identical data model elements more than once.</p> <p><i>Note:</i> New data objects and types should be modelled initially on AMOD-105 Operational data objects [O.CDB] so that other modellers can look for them and reuse them where appropriate.</p> <p>Instances of AMOD-105 should be structured in line with the structure used for the abstract concepts (so that there is generally one instance of AMOD-105 per instance of AMOD-025 - unless it is necessary to split the views up for readability).</p> <p>Abstract concepts</p> <p>Abstract concepts are defined on operational level in AMOD-025 Abstract concepts. These concepts may be updated, and/or new concepts might be identified during system analysis. In such a case, the person responsible for the abstract concept (e.g. operational concept architect) must be involved in the change process.</p> <p>If a new abstract concept is to be defined, follow the process step ARCH.008 Define abstract concepts.</p>
<p>Tools and non-human resources</p>	<p>Team for Capella</p>
<p>Cardinality</p>	<p>Once per operational capability, revisions allowed.</p>
<p>Completion criteria</p>	<p>The output views conform to their modelling rules.</p>
<p>Design review</p>	<p>ARCH.R.1 Operational capability review</p> <p>ARCH.R.2 Operational review - consolidated</p>

Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Systems engineer
Uses outputs (Informed)	None directly.

ARCH.159 Model operational states

[SM-3521](#) - Create Confluence page for activity definition ARCH.159 FINISHED

Goal	Refine understanding of state-based differences in behaviour of operational entities/actors
Requirements met by this process step	None identified (TBD)
Inputs	AMOD-022 Enterprise & environment definition AMOD-023 Operational entity/actor states
Outputs	AMOD-023 Operational entity/actor states (updated/new)
Methodology	<p>During analysis of operational activities, an operational process and an operational scenario, new states may be identified for operational entities, actors or abstract concepts, that were not known during ARCH.006.</p> <p>Refer to ARCH.006 for basic principles on how to approach the modelling of operational states</p> <p>Either update existing instances of view AMOD-023 or, where no statefulness was previously identified for an operational actor /entity, create a new one.</p> <p>In both cases, follow the modelling rules for AMOD-023.</p> <p>This activity should be done in parallel with the initial analysis of an operational capability.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	<p>(The output views conform to their modelling rules</p> <p>AND</p> <p>Statefulness that was identified during the analysis of the operational capability has been captured in the model)</p> <p>OR</p> <p>No further statefulness was identified during the analysis of this operational capability</p>
Design review	
Step done by (Responsible)	System architect

Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None identified

ARCH.161 Map operational activities to operational states

[SM-3522](#) - Create Confluence page for activity definition ARCH.161 FINISHED

Goal	Refine understanding of state-based differences in behaviour of operational entities/actors
Requirements met by this process step	None identified (TBD)
Inputs	AMOD-023 Operational entity/actor states AMOD-027 Operational bare business scenario AMOD-028 Operational activities and interaction definitions (single operational capability)
Outputs	Output is not directly visible on a view, it is a set of relationships in the model viewable via the "Available in states" property of the activities or the "Operational Activities / Functions" property of a state.
Methodology	<p>For each operational entity/actor involved in the scenario</p> <p style="padding-left: 40px;">For each state that the entity/actor can exhibit</p> <p style="padding-left: 80px;">Define the set of activities, allocated to this entity/actor, that are available while the entity/actor is in this state</p> <p style="padding-left: 40px;">end for</p> <p>end for</p> <p>If it is not clear whether an activity should or should not be available, set it to be available.</p> <p>Example from current (staffed) train operation: the train conductor (Zugchef) might have two states, "Routine operation" and "Emergency operation". In each state they have different duties. During routine operation, the Zugchef might be involved in checking the tickets of passengers. But they would not bother doing this during an emergency, rather they might be performing their duties to protect the train and evacuate passengers. Hence "check tickets" as an operational activity would be made available only in the "routine operation" state and not in the "emergency operation" state.</p> <p>Note: Mapping of operational activities to states can be done by setting "Available in states" property of the activities or the "Operational Activities / Functions" property of a state. Whereas the properties "entry", "exit" and "do activity" of a state shall NOT be used.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	For all the involved entities for this capability, the operational activities associated with each state have been correctly assigned.
Design review	
Step done by (Responsible)	System architect

Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.903 Assess & mitigate the operational risks

- [ARCH.171 Prepare business risk analysis](#)
- [ARCH.172 Prepare security risk analysis](#)
- [ARCH.173 Prepare safety risk analysis](#)
- [ARCH.020 Identify & classify operational deviations](#)
- [ARCH.011 Assess operational performance risks of scenario](#)
- [ARCH.017 Assess security risks of scenario](#)
- [ARCH.024 Assess operational safety risks of scenario](#)
- [ARCH.029 Allocate risk control responsibilities \(op. activity to op. entity/actor\)](#)
- [ARCH.175 Produce consolidated operational deviation analysis report](#)

Note the following look-up table. The approach for business, security and safety risks is generally the same, using a state model to identify potential undesired transitions.

We have a general set of terms for the state model, and a specific term for each of the three main areas of risk being assessed.

	Loss state (something bad has happened)	Not-allowed state (something bad could happen)	Transition from allowed to not-allowed state	Allowed state (nothing bad can happen)
Safety	Accident	Hazardous state	Hazard occurrence	Safe state
Security	Loss	Threat state	Vulnerability occurrence	Non-threat state
Business	Loss	Not-allowed state	Business risk occurrence	Allowed state

ARCH.171 Prepare business risk analysis

- [ARCH.012 Determine measure of effectiveness target values](#)
- [ARCH.013 Identify risks to business effectiveness \(business loss-risk mapping\)](#)

ARCH.012 Determine measure of effectiveness target values

[SM-2748](#) - Populate Confluence page for activity definition: ARCH.012 Determine measure of effectiveness target values **FINISHED**

Goal	Establish clear benchmarks to drive the performance of SysABB to a level that makes it suitable for widespread deployment.
Requirements met by this process step	ISO 29148 (2018) para 5.2.3 ISO 15288 (2015) para 6.4.1.3 a)
Inputs	AMOD-106 Enterprise goals
Outputs	CONOPS boilerplate section

Methodology	<p>In ARCH.144 the enterprise and its goals are defined.</p> <p>In ARCH.009 a set of metrics/measures is defined to quantify the railway infrastructure's operational effectiveness (that is, the degree to which the goals are achieved).</p> <p>In this activity, target values are to be defined for these measures of effectiveness.</p> <p>It will probably not be possible to arrive at a single target value for every possible application of SysABB. Measures of effectiveness can be imagined at national level but also at local levels and they may vary.</p> <p>For example, while overall railway capacity may increase by 35% nationally, it does not follow that every railway that deploys SysABB needs to achieve a 35% capacity increase. Some will achieve higher, and some lower.</p> <p>The intention is to establish targets that will drive the performance targets of SysABB to a level where SysABB is sufficient to be used widely on DB. In this activity, a strategic decision needs to be made whether that means setting the MoE targets to reflect the most demanding digital railway operations, or to a less strict value, whereby we would accept that SysABB would not be the right solution for some worst-case scenarios.</p> <p>These strategic decisions should be driven by senior DB management but it is for our team to organise the activity and obtain answers to all open points so that the development of SysABB can be guided by a vision of the level of effectiveness the digital railway is expected to achieve.</p> <p>General approach should be:</p> <ul style="list-style-type: none"> ■ identify the stakeholders who can give an authoritative decision on what the measures of effectiveness should be (for any MoEs where the target is not already fixed by strategies such as Starke Schiene) ■ consult the stakeholders to seek their opinions on the correct targets ■ obtain consensus agreement/compromises such that all stakeholders are satisfied ■ document the MoE targets in the CONOPS
Tools and non-human resources	<p>No specific tools identified</p>
Cardinality	<p>One-off with revisions permitted (under exceptional circumstances, that is, large-scale business change)</p>
Completion criteria	<p>The MoE target values were agreed with stakeholders</p> <p>The MoE target values were documented</p>

Design review	ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Railway industry stakeholders
Uses outputs (Informed)	SysABB suppliers All readers of the CONOPS

ARCH.013 Identify risks to business effectiveness (business loss-risk mapping)

[SM-2749](#) - Populate Confluence page for activity definition: ARCH.013 Identify risks to business effectiveness (feared event-undesired state mapping) **CREATED**

Goal	Provide a consistent starting point for business risk analysis
Requirements met by this process step	RiL 0120 series (DB quality management system)
Inputs	Selected parts of the RiL 0120 series relevant to this activity AMOD-106 Enterprise goals including measures of effectiveness
Outputs	AMOD-130 Business loss and risk state model
Methodology	<p>The main intended pattern is <allowed state> makes transition to <not-allowed state> which can lead to <loss>.</p> <p>Complementary patterns are:</p> <ul style="list-style-type: none"> • <not-allowed state>recovers to <allowed state> (for situations where there is a successful mitigation, or the problem clears up by itself) • <allowed state> persists in <allowed state> (a transition that allows the modelling of the probability of successful mitigations - and hence completes the state diagram as a hidden Markov model where the sum of all outgoing transitions from a state has probability 1) • <not-allowed state> persists in <not-allowed state> (where the not-allowed state is still active but a loss has not occurred) <p>The viewpoint definition for AMOD-130 indicates the detail of how this is captured in the analysis model.</p> <p>Using the relevant source material and regulations (with reference to existing material in the DB quality management system), define a standard set of business losses.</p> <p>In this activity, there is only the following to do:</p> <ul style="list-style-type: none"> • For each business loss, define a standard set of not-allowed states that can lead to the loss • For each not-allowed state, define a corresponding allowed state which is the exact opposite of the not-allowed state • Define the possible transitions from not-allowed states to losses • Assess the severity of losses • Derive, from the severity of a loss, the maximum tolerable occurrence rate of a not-allowed state <p>Not-allowed states must be expressed in terms of general business/operational conditions that could lead to a loss, excluding any notion of a system failure. They can include reference to abstract concepts that are not risk-related, for example an operational plan.</p> <p>Note: the intention here is not to create a working state model of the entire business, rather to reduce down to an abstract set the critical business losses and the mechanisms by which they happen, so that sufficient mitigations can be built into the operational concept. The focus is on creating a good experience for our customers by avoiding delays, making the railway user-friendly, and anticipating deviations that we can mitigate to maintain service quality.</p>
Tools and non-human resources	Team for Capella

Cardinality	One-off with possibility of revisions
Completion criteria	<p>The output view is complete i.e. sufficient losses and not-allowed states have been covered</p> <p>The output view includes elements that are required by DB regulations, where applicable</p> <p>The output view conforms to its modelling rules</p>
Design review	ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	RAMS architect
Provides input to /assists (Contributes)	RAMS engineer
Uses outputs (Informed)	None directly

ARCH.172 Prepare security risk analysis

- [ARCH.018 Identify internal & external issues affecting successful security](#)
- [ARCH.019 Determine need for information availability, confidentiality, integrity](#)
- [ARCH.170 Identify security losses and threats](#)

ARCH.018 Identify internal & external issues affecting successful security

[SM-2754](#) - Populate Confluence page for activity definition: ARCH.018 Identify internal & external issues affecting successful security

CREATED

Goal	Gain an understanding of the complete set of regulations that must be complied with i.e. all the sources of regulatory requirements.
Requirements met by this process step	ISO 27001 4.1 RiL 114.0210 05 (5)
Inputs	Expert knowledge Standards DB regulatory framework
Outputs	Consolidated list of security requirements including rationales.
Methodology	<p>There are multiple possible sources of definitions for security losses or threats that may results in unsafe situations.</p> <p>This task is to arbitrate between any conflicting regulatory requirement and arrive at a single decision for each type of requirement.</p> <p>For example, if security threats are defined differently in several places, this task includes the decision on a preferred source for the securty threat.</p>
Tools and non-human resources	Any preferred documentation tool (suggestion: document first in a Confluence page)
Cardinality	One-off with revisions allowed
Completion criteria	<p>All relevant legislation and regulations have been identified</p> <p>In case of contradicting or overlapping security requirements consolidated requirements have been derived and rationalized.</p>
Design review	ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	System RAMS manager

Completion criteria evaluated by (Accountable)	TBD.
Provides input to /assists (Contributes)	None identified
Reviews (Contributes)	TBD.
Uses outputs (Informed)	None identified

ARCH.019 Determine need for information availability, confidentiality, integrity

[SM-2755](#) - Populate Confluence page for activity definition: ARCH.019 Determine need for information availability, confidentiality, integrity **CREATED**

Goal	Set of information objects that are to be protected against loss, corruption or loss of integrity to ensure safe railw: operation.
Requirements met by this process step	IEC 62443 RiL 114.0210 05 (6)
Inputs	AMOD-110 Operational exchange items [O.CDB]
Outputs	Viewpoint that identifies all information objects whose vulnerability results in a loss (with respect to safety or business).
Methodology	<p>For each information object provided in instances of AMOD-110 it shall be determined whether any of the following characteristics of that information potentially results in a loss (with respect to safety of business):</p> <ul style="list-style-type: none"> ■ The defined information object is not transmitted ■ The defined information object is corrupted (i.e. not readable or unexpected information is exchanged) ■ The defined information object is having expected but manipulated values (e.g. a journey profile is exchanged but destination values are modified) <p>Note: The list above is not an exhaustive list of characteristics and has to be consolidated with security experts.</p>
Tools and non-human resources	Team for Capella
Cardinality	One-off with possibility of revisions
Completion criteria	<p>The The output view is complete i.e. sufficient losses and threat states have been covered</p> <p>The output view includes elements that are required by DB regulations, where applicable</p> <p>The output view conforms to its modelling rules</p>
Design review	ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	Security expert
Provides input to /assists (Contributes)	None identified
Reviews (Contributes)	Security engineer
Uses outputs (Informed)	ARCH.170 Identify security losses & threats

ARCH.170 Identify security losses and threats

[SM-3940](#) - Populate Confluence page for activity definition: ARCH.170 FINISHED

Goal	Provide a consistent starting point for security risk analysis
Requirements met by this process step	RiL 114.0210 (and thereby ISO 27001)
Inputs	Selected parts of the RiL 114 series relevant to this activity AMOD-106 Enterprise goals including measures of effectiveness
Outputs	AMOD-131 Security loss and threat state model
Methodology	<p>The main intended pattern is <non-threat state> makes transition to <threat state> which can lead to <loss state>. This transition is called a vulnerability occurrence.</p> <p>Complementary patterns are:</p> <ul style="list-style-type: none"> • <threat state>recovers to <non-threat state> (for situations where there is a successful mitigation, or the problem clears up by itself) • <non-threat state> persists in <non-threat state> (a transition that allows the modelling of the probability of successful mitigations - and hence completes the state diagram as a hidden Markov model where the sum of all outgoing transitions from a state has probability 1) • <threat state> persists in <threat state> (where the threat state is still active but a loss has not occurred) <p>The viewpoint definition for AMOD-131 indicates the detail of how this is captured in the analysis model.</p> <p>Using the relevant source material and regulations (with reference to existing material in the DB security regulations), define a standard set of security losses.</p> <p>In this activity, there is only the following to do:</p> <ul style="list-style-type: none"> • For each security loss, define a standard set of threat states that can lead to the loss • For each threat state, define a corresponding non-threat state which is the exact opposite of the threat state • Define the possible transitions from threat states to losses • Assess the severity of losses • Derive, from the severity of a loss, the maximum tolerable occurrence rate of a threat state <p>Threat states must be expressed in terms of general security conditions that could lead to a loss, excluding any notion of a system failure. They can include reference to abstract concepts that are not risk-related, for example an operational plan.</p> <p>Note: the intention here is not to create a working state model of the entire business, rather to reduce down to an abstract set the critical business losses caused by threat/vulnerability combinations and the mechanisms by which they happen, so that sufficient mitigations can be built into the operational concept. The focus is on identifying our abstract fundamental security needs, which will be apportioned further to the system and its technical components in later stages of the ARCH process.</p>
Tools and non-human resources	Team for Capella
Cardinality	One-off with possibility of revisions

Completion criteria	<p>The output view is complete i.e. sufficient losses and threat states have been covered</p> <p>The output view includes elements that are required by DB regulations, where applicable</p> <p>The output view conforms to its modelling rules</p>
Design review	<p>ARCH.R.2 Operational review - consolidated</p>
Step done by (Responsible)	<p>Security expert</p>
Provides input to /assists (Contributes)	<p>None identified</p>
Uses outputs (Informed)	<p>None directly</p>

ARCH.173 Prepare safety risk analysis

- [ARCH.023 Identify relevant safety legislation and regulation](#)
- [ARCH.025 Create foundation for safety risk model](#)
- [ARCH.026 Define the state model of accidents, hazardous and safe state](#)

ARCH.023 Identify relevant safety legislation and regulation

[SM-2761](#) - Populate Confluence page for activity definition: ARCH.023 Identify safety-related legal requirements from interested parties

FINISHED

Goal	Gain an understanding of the complete set of legislation and regulations that must be complied with i.e. all the sources of legislative and regulatory requirements.
Requirements met by this process step	CSM-SMS guidance 1.1 d)
Inputs	<p>Expert knowledge</p> <p>European railway legislation as maintained by the European Railway Agency</p> <p>German railway legislation</p> <p>DB regulatory framework</p> <p>A list of input documents, including the version numbers (to be developed outside ARCH and maintained when changes occur)</p>
Outputs	AMOD-128 Safety compliance strategy
Methodology	<p>There are multiple possible sources of definitions for safety-related accidents, safety targets, hazards, and other safety-related items needed for a consistent analysis.</p> <p>This task is to arbitrate between any conflicting rules and arrive at a single decision for each type of safety rule and source of information, ensuring that the single source of truth is sufficient to meet all requirements from all relevant original sources.</p> <p>For example, if accidents are defined differently in several places, this task includes the decision on a preferred source for the accident set, or concludes that a new definition is needed that does not match any existing standards (in this case, an explanation should be noted, in case of queries by approval bodies).</p> <p>The following sources, as a minimum, must be examined:</p> <ul style="list-style-type: none"> ■ European legislation as hosted by the European Railway Agency ■ National legislation including AEG and EIGV ■ Guidance for homologation as defined by the Federal Office for the Railway (Eisenbahnbundesamt) ■ Railway safety standards EN 50126 and related standards from the same suite
Tools and non-human resources	<p>Any preferred documentation tool</p> <p>(suggestion: document first in a Confluence page)</p>
Cardinality	One-off with revisions allowed
Completion criteria	<p>All relevant legislation and regulations have been identified</p> <p>The safety compliance strategy for DBB specifies one consistent approach to safety that satisfies all relevant legislation and regulations</p>
Design review	Special check by Design Authority

Step done by (Responsible)	System RAMS manager
Provides input to /assists (Contributes)	Homologation manager DB legal department
Uses outputs (Informed)	None identified

ARCH.025 Create foundation for safety risk model

[SM-5635](#) - Populate Confluence page for activity definition: ARCH.025 Create foundation for safety risk state model READY FOR REVIEW

Goal	<i>Identify a stable set of "Accidents", "Hazards" and "Safe Control Measures" each with an with an empty cutset as a starting point for Risk Control Management for the execution of risk analyses on Operational Layer</i>
Requirements met by this process step	<p>CSM-SMS guidance 1.1 b), 3.1.1.1 a),</p> <p>CSM-RA (402/2013) §2.5,</p> <p>EN 50126-1:2017 §7.4</p> <p>EN 50126-2:2017 §5</p>
Inputs	<p>AMOD-128 Safety compliance strategy</p> <p>AMOD-129 Relevant safety legislation/regulations</p> <p>(for Accidents:)</p> <p>[EU] DIRECTIVE (EU) 2016/798 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL (of 11 May 2016 on railway safety)</p> <p>[EU2] DIRECTIVE 2004/49/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL (of 29 April 2004 on railway safety)</p> <p>[UIC] UIC SAFETY DB (https://safetydb.uic.org/)</p> <p>[BEU] Bundesstelle für Eisenbahnunfalluntersuchung - BEU (https://www.eisenbahn-unfalluntersuchung.de)</p> <p>[RiL] Ril 423.0101A01 Definition der gefährlichen Ereignisse im Bahnbetrieb 4230101A01</p> <p>(for Hazards)</p> <p>EN62267_BB1.pdf (Bahnanwendungen - Automatischer städtischer schienengebundener Personennahverkehr)</p> <p>Railway applications, IEC 62290-1: Urban Guided Transport Management and Command/Control Systems, Part 1: System Principles and Fundamental Concepts, Date: 2004, IEC 62290-1_2004.pdf</p> <p>Railway applications, IEC 62290-2: Urban Guided Transport Management and Command/Control Systems, Part 2: Functional Requirements Specification, Date: 13.01.2009, IEC 62290-2_2009.pdf</p> <p>IEEE 1474.1 Standard for Communications- Based Train Control (CBTC), Performance and Functional Requirements, Date: 25.02.2005, IEEE 1474.1-2004.pdf</p> <p>IEEE 1474.2 Standard for Communications- Based Train Control (CBTC), Performance and Functional Requirements, Date: 12.12.2003, IEEE 1474.2-2003.pdf</p> <p>IEC 61508-1: Railway applications, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements, Ed. 2.0, 2010-04, IEC_61508-1_2010.pdf</p> <p>MODSafe Project: MODSafe Homepage - Documentlist</p> <p>Especially, D4.2 "Analysis of Common Safety Requirements Allocation for MODSafe continuous Safety Measures and Functions" and D2.1 "First List of Hazards, Preliminary Hazard Analysis (PHA)"</p> <p>NeuPro.71_Risikoanalyse_V2.0.pdf</p> <p>SafetyFunctions_VDE V 0831-103.PDF</p>

	<p>UNISIG SPecification: Subset-088, SUBSET-088 v3.5.4.zip, and Subset-091 (baseline v3.6.0R02), SUBSET-091_v360.pdf</p> <p>TeSIP; s. SIRF/TeSIP (EBA, Sicherheitsregelung Fahrzeuge)</p> <p>TSI's (e.g. TSI Loc&Pas, §4.2.5.5.8): see List of TSI's (ERA Homepage)</p> <p>TSI-1300-2014, update from 2019, Persons with Disabilities and with Reduced Mobility TSI, https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02014R1300-20190616&from=EN</p> <p>TSI-1302-2014, update from 10.03.2020, Rolling Stock - Locomotives and Passengers TSI https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02014R1302-20200311&from=EN</p> <p>TSI-0321-2013, update from 11.03.2020, Rolling Stock - Freight Wagons TSI http://data.europa.eu/eli/reg/2013/321/2020-03-11</p> <p>TSI, Guide for the application of the TSI OPE (Operations), 28/06/2019, guide_on_application_of_ope_tsi_2019_en.pdf</p> <p>TSI, Operation and Traffic Management, IMPLEMENTING REGULATION (EU) 2019/773, 16 May 2019; s. TSI Operation and Traffic Management</p> <p>Other inputs, e.g. projects like Shift2Rail, X2Rail</p>
Outputs	<p><i>(for Accidents)</i></p> <p><i>stable set of "Accidents" with an empty cutset</i></p> <p><i>(for hazards)</i></p> <p><i>stable set of generic "Hazards" related to the set of identified "Accidents"</i></p>
Methodology	<p><i>state the methodology as a list of instructions; this information forms the Description field of a ticket for this process to be done, so it must be sufficient for someone to follow the process, even if they have not done it before.</i></p>
Tools and non-human resources	<p><i>list of all tools (physical and software) needed for this process, plus any non-obvious resources that need to be planned for (such as access to particular non-DBB individuals, operational areas, site visits...)</i></p>
Cardinality	<p><i>state how many times this process step is expected to be carried out - e.g.</i></p> <ul style="list-style-type: none"> • <i>one-off</i> • <i>once per <model element></i> • <i>once per baseline</i> <p><i>etc.</i></p>
Completion criteria	<p><i>state the conditions that must be TRUE before this process step can be considered complete.</i></p> <p><i>These should be inserted in the Acceptance Criteria of a Jira ticket for this process step to be done.</i></p>
Design review	<p><i>Link to the corresponding design review where the completion of this activity is evaluated.</i></p>
Step done by (Responsible)	<p><i>Identify the role that is responsible for organising this process step and producing the output;</i></p> <p><i>This is the Assignee in Jira.</i></p>
Provides input to /assists (Contributes)	<p><i>Identify the roles that assist in the work or provide input information</i></p>

**Uses outputs
(Informed)**

*Identify the roles and/or process areas that make use of this information **outside** of the ARCH process area;*

Draft foundation accidents

Goal	Identification and consolidation of the high-level accidents list for operational level
Requirements met by this process step	CSM-RA (402/2013) §2.5 EN 50126-1:2017 §7.4 EN 50126-2:2017 §5
Inputs	Sources/References: <ul style="list-style-type: none">• [EU] DIRECTIVE (EU) 2016/798 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL (of 11 May 2016 on railway safety)• [EU2] DIRECTIVE 2004/49/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL (of 29 April 2004 on railway safety)• [UIC] UIC SAFETY DB (https://safetydb.uic.org/)• [BEU] Bundesstelle für Eisenbahnunfalluntersuchung - BEU (https://www.eisenbahn-unfalluntersuchung.de)• [RiL] Ril 423.0101A01 Definition der gefährlichen Ereignisse im Bahnbetrieb 4230101A01
Outputs	Final and complete set of accidents within transportation systems in EXCEL and Mindmap format: DBS Accidents List

Methodology

Definition/Scope

Accident is considered as "An unintended event or series of events that results in potential loss of life, loss of health or loss of environmental integrity. "

Accident are understood as the following safety-related losses:

- Loss of life to people
- Loss of health to people
- Loss of integrity of environment

The "loss of a system or service" considered in CENELEC and CSM-RA is excluded from the analysis, this case will be considered separately as "incident".

Specific harm/accident resulting from terrorism (e.g. bomb explosion), arson, vandalism are not considered.

The consequences related to passenger security matter are also not considered.

Although suicide is not differentiated in the accident list, the related consequence is covered by the accident list (considering a single person is affected).

Methodology

The hereafter steps have been followed for the definition of the accidents:

1. Consideration of the accident explicitly defined in standard and regulation texts, add classical high level accidents based on the own experience;
2. Adaptation/supplementation/Distinction of the accident considering different way of harming people (different extent for a same harm is not considered as a different accident);
3. Verification the different type of area of railway system i.e. platform, train, track, infrastructure (bridge, tunnel) are reflected;
4. Verification the harm related to the hazard captured (from standard and regulation texts) when carrying out the "[Collection of Hazards](#)" step are covered by the accident list;
5. Verification the consistency/relevance of the accident when carrying out the "[Generic Hazard Identification](#)" step.

When performing the three last steps, some accidents have been renamed/refined/supplemented (leave of the accident tree) if needed and sometimes the tree structure has been changed.

Remark: DB (EDS - Safety-Datenbank) database for accidents and disturbances/malfunctions does implement the categories of BEU

Three different consideration levels are considered for accident definition:

1. accidents affecting train - called "Rolling Stock Accident" i.e. , abbreviated in RSA
e.g. collision of train, derailment, train entering a life hostile environment..
2. Explosion/Blast, Fire, Pollution (the accident can be assigned neither to Rolling Stock Accident nor to Accident of Person(s) strictly)
3. accidents affecting persons directly (called "Accident of Person(s)", abbreviated in AP, e.g. asphyxia, crush, slip/trip, burn, deafening)

All identified accident have been assigned with the unique ID and structured in the hierarchical order.

Tools and non-human resources

- MS EXCEL
- Freeplane: https://www.freeplane.org/wiki/index.php/Portable_Freeplane

Cardinality	<ul style="list-style-type: none"> • Created and provided as a first stable draft or baseline incl. reviews • Accidents will be further implemented and maintained in Capella (Risk-State-Model) • In case of significant changes, set of accidents will be changed and confirmed by review (ensured by implemented quality management system and related processes)
Completion criteria	Finite and complete list of consolidated applicable independent accidents in transportation with clear description. The cutset of accidents provided in the list shall be empty.
Design review	<i>Link to the corresponding design review where the completion of this activity is evaluated.</i>
Step done by (Responsible)	RAMS Expert, Risk Manager and/or System Architect
Provides input to /assists (Contributes)	RU, IM, any competent expert
Uses outputs (Informed)	System Architect, RAMS Expert, Risk Manager

Draft foundation hazard

Goal	Identification and consolidation of the high-level hazards for operational level [and assignment to the identified accidents]
Requirements met by this process step	CSM-RA (402/2013) §2.5 EN 50126-1:2017 §7.4 EN 50126-2:2017 §5
Inputs	AMOD-128 Safety compliance strategy AMOD-129 Relevant safety legislation/regulations The list of the analyzed norms, standards and projects (->Collection of Hazards):

- Standards/Norms: [EN62267_BB1.pdf](#) (Bahnanwendungen - Automatischer städtischer schienengebundener Personennahverkehr)
So maybe start with this task. It has already been initiated by me. The related page can be found in my Sandbox at [EN 62267](#).
You can do me a favour, if you could check my translation, I already did for the hazards.
Within this standard (EN 62267:BB1) and with respect to Table 1 in this standard, I got to "8 Sicherstellen des Erkennens und die Bewältigung von Notfallsituationen" at page 23, where you can proceed.
- Standards/Norms: Railway applications, IEC 62290-1: Urban Guided Transport Management and Command /Control Systems, Part 1: System Principles and Fundamental Concepts, Date: 2004, [IEC 62290-1 2004.pdf](#)
- Standards/Norms: Railway applications, IEC 62290-2: Urban Guided Transport Management and Command /Control Systems, Part 2: Functional Requirements Specification, Date: 13.01.2009, [IEC 62290-2 2009.pdf](#)
- Standards/Norms: IEEE 1474.1 Standard for Communications- Based Train Control (CBTC), Performance and Functional Requirements, Date: 25.02.2005, [IEEE 1474.1-2004.pdf](#)
- Standards/Norms: IEEE 1474.2 Standard for Communications- Based Train Control (CBTC), Performance and Functional Requirements, Date: 12.12.2003, [IEEE 1474.2-2003.pdf](#)
- Standards/Norms: IEC 61508-1: Railway applications, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 1: General requirements, Ed. 2.0, 2010-04, [IEC 61508-1 2010.pdf](#)
- MODSafe Project: [MODSafe Homepage - Documentlist](#)
Especially, D4.2 "Analysis of Common Safety Requirements Allocation for MODSafe continuous Safety Measures and Functions" (starting from Chapter 10) shall be taken into account. You can also take a look in D2.1 "First List of Hazards, Preliminary Hazard Analysis (PHA)", if you need more information.
- Neupro Risk Analysis: [NeuPro.71_Risikoanalyse_V2.0.pdf](#) bzw. Vornorm [SafetyFunctions_VDE V 0831-103.PDF](#)
Neupro Risk Analysis is based on the VDE standard. So they shall be more or less equal respectively similar. Just use both of them to make a check for completeness.
For the source Neupro, there are several informations distributed with respect to hazards (Gefährdung (Maßgebende Ausfallart)), system safety functions etc. in chapters 5 and 6. Accidents are somehow mentioned as "Ereignisart" - please align them with the accident list given in confluence (see references in the template).
For the VDE standard, take a look in Annex B. With "Ausfallarten der Funktion", the hazard is described (see also compared to Risk Analysis Neupro). The safety function is stated as headline. "Rahmenbedingungen" are external conditions, and "Auswirkungen" should be related to accidents.
I would start with Neupro, and complete information about hazards (if there are any additions not mentioned within related Risk Analysis) by using the VDE standard. You can do me a favour, if you can add THR (tolerable hazard rate) mentioned in NeuPro Risk Analysis within the corresponding column comment.
- UNISIG Specification: Subset-088, [SUBSET-088 v3.5.4.zip](#), and Subset-091 (baseline v3.6.0R02), [SUBSET-091 v360.pdf](#)
I think there should be only two mentioned: Something like overspeed and something like passing a danger point. Please have a look.
- EBA - SIRF/TeSIP [TeSIP](#); s. [SIRF/TeSIP \(EBA, Sicherheitsregelung Fahrzeuge\)](#)
There is an excel-sheet and the related information can be found in column "D" (function), "I" & "K" (hazard).
- TSI's (e.g. TSI Loc&Pas, §4.2.5.5.8): see [List of TSI's \(ERA Homepage\)](#)
 - TSI-1300-2014, update from 2019, Persons with Disabilities and with Reduced Mobility TSI, <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02014R1300-20190616&from=EN>
 - TSI-1302-2014, update from 10.03.2020, Rolling Stock - Locomotives and Passengers TSI <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02014R1302-20200311&from=EN>
 - TSI-0321-2013, update from 11.03.2020, Rolling Stock - Freight Wagons TSI <http://data.europa.eu/eli/reg/2013/321/2020-03-11>
 - TSI, Guide for the application of the TSI OPE (Operations), 28/06/2019, [guide on application of ope tsi 2019 en.pdf](#)
 - TSI, Operation and Traffic Management, IMPLEMENTING REGULATION (EU) 2019/773, 16 May 2019; s. [TSI Operation and Traffic Management](#)
- Other projects: e.g. Shift2Rail, X2Rail ...

The **consolidated list of the relevant accidents**: [DBS Accidents List](#)

Expert knowledge

Outputs	<p>Finite set of identified hazards including Accident-Hazard-Relation and relevant conditions and functions (if given) as Excel table:</p> <p>https://rmt.jaas.service.deutschebahn.com/confluence/display/SSI/Collection+of+Hazards</p> <p>Set of hazards and accident-hazard-relations (as Mindmaps):</p> <p>Generic Hazard Identification with sub-websites</p>
Methodology	<p>In order to establish a complete list of typical hazards in railway operation different railway norms have been analyzed for the hazards incl. causes and related accidents and risk control measures including following steps:</p> <ol style="list-style-type: none"> 1. Collection of hazards from relevant documents (see list of the analyzed norms, standards and projects listed above in Inputs-Section) 2. Consolidation of the hazards identified by relating them to the finite set of accidents identified (see consolidated list of the relevant accidents listed above in Inputs-Section) <h2>Collection of hazards from relevant norm documents</h2> <p>The purpose of this task is to establish a complete list of typical hazards in railway operation. The analysis of the norm documents shall</p> <ul style="list-style-type: none"> • present the completeness of the hazard list and • establish a generic description of hazards. <p>The memory location of the analysed norm documents in the storage system DB Confluence is:</p> <p>https://rmt.jaas.service.deutschebahn.com/confluence/display/SSI/Collection+of+Hazards</p> <p>Every document is considered in sub-sides in DB Confluence.</p> <h2>Hazard assignment and consolidation</h2> <p>The consolidation of hazards was done by 1:n mapping of hazards to the accidents (hazard are considered as the conditions that may lead to an accident, n hazards could be mapped to 1 accident). There is no constraint that a hazard is solely allocated / mapped to one accident.</p> <p>For the consolidation of the hazards from reference norm documents the program “Freeplane” was used, which is carried out in a mind map. During consolidation it has been checked for every accident if full set of related hazard already exists in database, if necessary new hazard have been defined and added to database.</p> <p>The author, the state of the mind map and the name of the person, who makes the review, are listed on the Confluence-site given in the section Output.</p> <p>Every mind map is considered in sub sides in DB Confluence. The links to these sub sides are listed in the table. On these sub-pages the review comments can be entered. The review comments can be also inserted into the mind map for each accident category.</p>
Tools and non-human resources	<ul style="list-style-type: none"> • MS EXCEL • <i>Freeplane</i>: https://www.freeplane.org/wiki/index.php/Portable_Freeplane
Cardinality	<ul style="list-style-type: none"> • Created and provided as a first stable draft or baseline incl. reviews • Hazards will be further implemented and maintained in Capella (Risk-State-Model) • In case of significant changes, set of hazards will be changed and confirmed by review (ensured by implemented quality management system and related processes)

Completion criteria	List of consolidated applicable hazards with clear description [assigned to previously identified accidents] satisfying all working group members
Design review	<p><i>Link to the corresponding design review where the completion of this activity is evaluated.</i></p> <p>Remark: Internal review is documented in list in Generic Hazard Identification (in general still no quality management system implemented in DBS)</p>
Step done by (Responsible)	RAMS Expert or Risk Manager
Provides input to /assists (Contributes)	other RAMS Expert or Risk Manager
Uses outputs (Informed)	<i>Identify the roles and/or process areas that make use of this information outside of the ARCH process area;</i>

Draft foundation control measure

Goal	Identify risk control measures to avoid hazards / to bring the system from hazardous to the non-hazardous state
Requirements met by this process step	<p>CSM-RA (402/2013) §4 (Risikomanagementverfahren)</p> <p>EN 50126-1:2017 §7.4, §7.5</p> <p>EN 50126-2:2017 §5</p>
Inputs	<p>https://rmt.jaas.service.deutschebahn.com/confluence/display/SSI/Safety+functions</p> <p>Analyzed standards:</p> <ul style="list-style-type: none"> • VDE 0831-103 (2020-09) - Elektrische Bahn-Signalanlagen, Teil 103: Ermittlung von Sicherheitsanforderungen an technische Funktionen in der Eisenbahnsignaltechnik; Ausgabedatum: 2020-09 • SIRF TESIP - Sicherheitsregelung Fahrzeug - Überarbeitete Fassung 2019; Methode zum Festlegen und Nachweisen sicherheitsbezogener Anforderungen und Bewertung der Risiken im Rahmen der Umsetzung der CSM-RA und der EN 50126; Datum: 01.01.2020 • IEC 62290-2/Ed.1 2CD © IEC - 2009/01/13 – 11 – 9/1229/CD; RAILWAY APPLICATIONS – URBAN GUIDED TRANSPORT MANAGEMENT AND COMMAND/CONTROL SYSTEMS; Part 2: Functional requirements specification • IEC 62290-1 Railway applications - Urban guided transport management and command/control systems - Part 1: System principles and fundamental concepts • 1474.1 IEEE Standard for Communications-Based Train Control (CBTC); Performance and Functional Requirements; 2005 • 1474.2 IEEE Standard for User Interface Requirements in Communications-Based Train Control (CBTC) Systems; 2003 • EN62267 (2010) - Railway applications – Automated urban guided transport (AUGT) – Safety requirements (IEC 62267:2009) • ETCS Subset-088 v354 ETCS Application Level 1 - Safety Analysis, Part 1 - Functional Fault Tree • MODSafe Project: MODSafe Homepage - Documentlist; Especially, D4.2 "Analysis of Common Safety Requirements Allocation for MODSafe continuous Safety Measures and Functions", D2.1 "First List of Hazards, Preliminary Hazard Analysis (PHA)"
Outputs	<p>Finite set of already known and used Control Measure to cope with hazards (as mindmaps):</p> <p>https://rmt.jaas.service.deutschebahn.com/confluence/display/SSI/Safety+functions</p> <p>See subpages sorted by accident.</p>
Methodology	<p>Control measure (CM) are used on operational level. The control measure shall avoid a hazard or mitigate a hazard. Following steps have been applied in order to create a set of suitable control measures:</p> <p>1) Identify already existing control measures resp. system (safety) functions</p> <p>The different railway standards have been analyzed (see Input) in order to identify safety functions which can mitigate identified hazards or their failure modes.</p> <p>2) Define control measures to mitigate hazards/to negate their conditions</p> <p>For every hazard/its condition or failure mode a minimum one control measure has been assigned in the mindmaps created during hazard-accident assignment process.</p> <p>3) Consolidate the set of identified control measures and system (safety) functions</p> <p>The list of identified control measures/safety functions have been mapped to the control measures assigned to the identified hazards and hazard failure modes. Where necessary new control measures have been defined in order to suit to the failure mode/hazard, where possible the established safety functions or their adaptations have been used.</p>

Tools and non-human resources	<ul style="list-style-type: none"> • MS EXCEL • Freeplane: https://www.freeplane.org/wiki/index.php/Portable_Freeplane
Cardinality	<ul style="list-style-type: none"> • Created and provided as a first stable draft or baseline incl. reviews • Control Measures will be further implemented and maintained in Capella, if applicable (Risk-State-Model) • In case of significant changes, set of Control Measures will be changed and confirmed by review (ensured by implemented quality management system and related processes)
Completion criteria	List of consolidated control measures applied to all before identified hazards and their failure modes sorted by accident; no further Control Measures needed for Risk Management Procedure
Design review	<p><i>Link to the corresponding design review where the completion of this activity is evaluated.</i></p> <p>Remark: Internal review is documented in list in Safety functions (still no quality management process in DBS).</p>
Step done by (Responsible)	RAMS Expert, Risk Manager or System Architect
Provides input to /assists (Contributes)	other RAMS Expert, Risk Manager or System Architect
Uses outputs (Informed)	<p><i>Identify the roles and/or process areas that make use of this information outside of the ARCH process area;</i></p> <p>The consolidation of the list of control measures is a basis for the implementation of state model in Capella.</p>

ARCH.026 Define the state model of accidents, hazardous and safe state

[SM-2762](#) - Populate Confluence page for activity definition: ARCH.026 Identify operational safety risks (hazard-accident mapping)

FINISHED

[SM-2480](#) - Update modelling rules and ARCH.026 / ARCH.020 for operational level safety elements

IN IMPLEMENTATION

[SM-5059](#) - Update ARCH.026 after feedback from RAMS manager

FINISHED

Goal	Provide a consistent starting point for safety risk analysis
Requirements met by this process step	CSM-SMS guidance 1.1 b), 3.1.1.1 a) (further to be filled as recommended by system RAMS manager)
Inputs	AMOD-128 Safety compliance strategy AMOD-129 Relevant safety legislation/regulations List of accidents and assigned hazards (following a tree structure) ARCH.020 Identify & classify operational deviations
Outputs	AMOD-030 Accident & hazard state model
Methodology	<p>This process step consist in preparing the risk model basis which will be enriched iteratively with steps ARCH.020 Identify & classify operational deviations, ARCH.027 Evaluate operational safety risks and ARCH.028 Define operational safety measures</p> <p>The main intended pattern of the state model is the representation of a <safe state> that makes transition to a <hazardous state> which can then lead to an <accident state>. as shown by the upper part of the below diagram</p> <p>Complementary patterns of the model (shown above) are:</p>

- <accident state> persists in <accident state> (it is considered once the accident state is reached, that it cannot be left thus this transition exists only for calculation reason: probability = 1)
- <hazardous state> recovers to <safe state> (for situations where there is a successful safety reaction, or the hazard clears up by itself)
- <safe state> persists in <safe state> (a transition that allows the modelling of the probability of successful mitigations - and hence completes the state diagram as a hidden Markov model where the sum of all outgoing transitions from a state has probability 1)
- <hazardous state> persists in <hazardous state> (where the hazardous state is still active but an accident has not occurred). Since it is considered that hazardous state cannot be kept thus this transition exists only for calculation reason (probability = 0)

See [Risk Assessment \(EN 50126\)](#) for background and further explanation of the buildup of the model.

The viewpoint definition for [AMOD-030 Accident & hazard state model](#) indicates the detail of how this is captured in the analysis model.

A set of hazards assigned to the respective accident which are identified from the relevant legislation, regulations, standards are used for defining the accident states and related hazardous state(s) in the model as starting point. This initial set will be consolidated and potentially updated/supplemented when carrying out the operational deviation analysis of [ARCH.020](#).

A Safety Risk Model is to be created for each hazardous state identified in relation to an accident state.

Different risk models may be needed for one hazardous state if for the same hazardous state, there exist:

- Different accident states with different severities;
- Different conditions (including different involved safety-related functions) for state transitions or same conditions with drastically different probabilities;

Remarks:

- When several successive accidents are related to a hazard, only the accident occurring first is considered (e.g. for a collision with a static obstacle, a subsequent derailment is not considered), if the subsequent accident is of importance, this shall be addressed in a separate risk model.
- Before starting creating a new risk model, the coverage and the existence of potential variant models shall be investigated. By variant models the difference in terms of conditions/events involved to lead to the accident or related probability (not the severity extent) is meant. As far as possible only one risk model per hazard shall be defined, if the conditions cannot be combined together or the probability cannot be covered by a worst case assumption then a variant risk model is needed. In this case the different variant and the respective severity when different can be reflected on each model in instantiating the accident description (e.g. by mentioning "big static obstacle").

The set of the risk state models defined as starting basis will be then completed by potentially additional hazardous states or new conditions for transitions between states found out after performing the 1st part of ARCH.020.

The worst case consequence of a particular accident related to a hazard in question is considered in the given severity level.

When creating a risk model, the following steps have to be carried out:

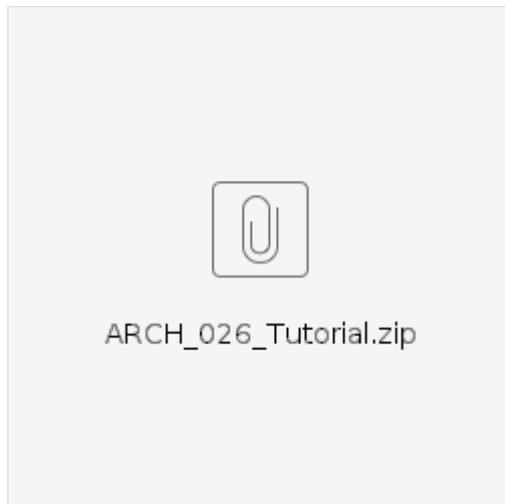
- Define the accident state;
- Define the initial severity category of the accident state (worst case), the accident rate target is automatically derived;
- Define the hazardous state;
- Define the safe state (which is the exact opposite of the hazardous state);
- For each created state: Select predefined risk state type property - accident state, hazardous state or safe state.

The hazardous state must be expressed in terms of physical conditions that could lead to an accident, **excluding** any notion of a system or safety function failure. They can include reference to abstract concepts that are not safety-related, for example an operational plan.

Then the following is to be done just formally without any content consideration

- For each accident state, define one decision nodes (pseudo states) to represent the "Accident state entry gate"
- For each hazardous state, define one decision nodes (pseudo states) to represent the "Hazardous state entry gate"
- For each safe state, define two decision nodes (pseudo states):
 1. One to represent the "Hazardous state recovery gate"
 2. One to represent the "Safe state persistence gate"
- Define the following unnamed transitions:
 1. From hazardous state to hazardous state (reflexive, probability set to 0, see reason above)
 2. From "Hazardous state entry gate" state to hazardous state
 3. From "Hazardous state recovery gate" to safe state
 4. From "Safe state persistence gate" to safe state
 5. From "Accident state entry gate" to accident state
 6. From accident state to accident state (reflexive, probability set to 1, see reason above)

Guidance:



Tools and non-human resources	Team for Capella
Cardinality	One-off with possibility of revisions - refinement in several model if the conditions transitions cannot be put together
Completion criteria	<p>The accident states and hazard states set is complete</p> <p>The accident states and hazard states set covers the needs of the relevant safety legislation/regulations</p> <p>The output view conforms to its modelling rules</p>
Design review	ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	RAMS Architect
Provides input to /assists (Contributes)	NA
Uses outputs (Informed)	None directly

ARCH.020 Identify & classify operational deviations

[SM-2756](#) - Populate Confluence page for activity definition: ARCH.020 Identify & classify operational deviations **IN IMPLEMENTATION**

[SM-2480](#) - Update modelling rules and ARCH.026 / ARCH.020 for operational level safety elements **IN IMPLEMENTATION**

Goal	Gain confidence that all the deviations having a potential impact on safety, business risk and security are identified.
Requirements met by this process step	CSM-SMS guidance 1.1 b), 3.1.1.1 a)
Inputs	AMOD-027 Operational bare business scenario AMOD-028 Operational activities and interaction definitions (single operational capability) AMOD-029 Operational bare business process AMOD-030 Accident and hazard state model AMOD-130 Business loss and risk state model AMOD-131 Security loss and threat state model
Outputs	AMOD-030 Accident and hazard state model (updated) AMOD-130 Business loss and risk state model (updated) AMOD-131 Security loss and threat state model (updated) AMOD-136 Single operational capability deviation analysis report
Methodology	<p>The ARCH.026 step is made of the two following sub-steps:</p> <p>1 - Build the condition into the relevant risk state model</p> <p><u>Safety risks</u></p> <p>The risk state models relevant to the capability (thinkable safety consequence of failure) in question are identified. The conditions identified in the safety trees are integrated inside the following transitions of the risk state model</p> <ul style="list-style-type: none"> • transition from safe state to hazardous state • transition from hazardous state to accident state <p><u>Business risks</u></p> <p>TBD</p> <p><u>Security risks</u></p> <p>TBD</p> <p>2 - Perform the deviation analysis (focus on effects) and consolidate the risk models needed (supplement them if needed)</p> <p>The primary focus of this task is to identify what deviations in operational interactions can occur during the execution of an operational capability. Therefore the primary input is AMOD-029 but AMOD-027 and AMOD-028 may be useful cross-references.</p> <p>This task aims for verifying the exhaustivity of the conditions and to build a relationship between deviations of bare business models (the safety ones at least).</p>

! The activity should be done jointly between the RAMS architect and Security Architect and the system architect. The system architect is responsible for organising the work but should not attempt to do this alone.

For every interaction in the bare business process

Determine which of the following deviations could occur on the interaction:

It is important to capture the object in consideration

1. Not providing the interaction leads to a hazard.
=>Guideword: "**1-Not providing**" # state/info change intended to be used
2. Providing the interaction leads to a hazard
 - a. Consider contexts in which the (control) interaction may never be safe => the interaction may never be unintended
=> Guideword "**2.1-unintended**" # state/info change not intended to be used
 - b. Consider contexts in which the (control) interaction has an incorrect parameter (e.g. setting an incorrect emergency)
=> Guideword "**2.2-incorrect**" # parameter/way of performing the function (e.g. passenger)
 - c. Consider contexts in which an insufficient or excessive (control) interaction may be unsafe (e.g. providing insufficient)
=> Guideword "**2.3a-insufficient**", # parameter
=> Guideword "**2.3b-excessive**" # parameter
 - d. Consider contexts in which the direction of the (control) interaction may be unsafe (e.g. providing turn left instead of right)
=> Guideword "**2.4-wrong direction**" # info
 - e. Consider contexts in which the (control) interaction has already been provided (e.g. repetitive or oscillatory control)
=> Guideword "**2.5-already**" # state/info already used
3. Providing a potentially safe (control) interaction but too early, too late, or in the wrong order
=> Guideword "**3a-too early**", # state/info intended to be used
=> Guideword "**3b-too late**", # state/info intended to be used
=> Guideword "**3c-wrong order**" # state/info sequence
4. The control action lasts too long or is stopped too soon (for continuous control actions, not discrete ones)
=> Guideword "**4a-too long**", # state/info continuous command
=> Guideword "**4b-too short**" # state/info continuous command

The rationale for deciding a deviation will **not** occur as a string property value on the interaction with name "Rationale" be recorded

Safety risk

When having/contributing to a safety consequence,

- The related accident shall be mentioned in the deviation. TBD A "safety risk" attribute shall be then allocated to the deviation.
- The existence of the related risk model (coping hazard and accident) shall be checked, if missing the additional one shall be created. In this case the Step ARCH.026 shall be revisited to complete the initial safety risk models with missing hazards identification analysis.

~~the deviations identified as related/contributing to a safety consequence shall be linked to Safe State to Accident risk model.~~

~~Remark: the Step ARCH.026 shall be revisited to complete the initial safety risk models with missing hazards identification analysis. (integrated above)~~

Business risk

TBD ...

Security risk

TBD ...

2 - Integrate the deviation into the relevant risk model

Safety Risks:

- The hazardous state description shall be checked then consolidated/updated considering all the relevant deviation
- potential hazardous state in addition to the one defined during the performance of ARCH.026 at first time
- Once the hazardous description is frozen the deviation judged as having an impact on safety (linked to a potential Safe state to hazardous state or to the hazardous state to accident state transition
- For Safe state to hazardous state transition then to the hazardous state to accident state transition, the combination materialized by a joint and diamond) of the identified deviations shall be defined (by adding joint or choice connection
- The additional condition needed (not covered by a deviation) shall be added if any (e.g. passenger behaviour) → to be
- It should be ensured that for one transition all deviations/conditions are independent from one another. If two deviations then they should be combined into one.

~~Remark: The Step ARCH.026 shall be revisited to complete the safety conditions identified as a result of the interaction~~

The defined combination of deviation shall be checked also against the tree for hazard investigation. If a condition is missing, the architect who will decide the potentially needed additional interaction or operation activity

Business risks

...

Security risk

...

For each deviation that **could** occur

Define the deviation by re-using and filling the predefined constraint template (Deviation Type, DeviationName, DeviationMaxPermitted HourlyRate, DeviationDescription)

For each individual way that the deviation could occur

Identify the state model corresponding to the consequences of the deviation, for security losses and time and business losses and risks; determine if the current deviation could cause a transition to a not-allowed state

i It is possible that new not-allowed states could be identified during this process. Any new states should be fed into [ARCH.026](#), [ARCH.170](#) or [ARCH.013](#).

*Note: a single deviation **could** cause transitions to multiple not-allowed states, including not-allowed states. A single deviation could cause security, safety **and** business risks all at once)*

If the current deviation **can** cause a transition to a not-allowed state

On the corresponding state diagram, create a new transition from the non-hazardous state to the not-allowed state

If this is the second transition going from the non-hazardous state to the decision node

Create a join node and adapt already existing transitions for this state change such that a transition leads to the join node

```

    Add a transition from the join node to the "total hazard occurrence" join node
  end if

  Set the trigger of the new transition to be the interaction currently being examined

  Set the guard of the new transition to the defined constraint, i.e. the previously defined deviatio

end if

end for

end for

end for

```

Now populate the complementary transitions, where they can occur naturally:

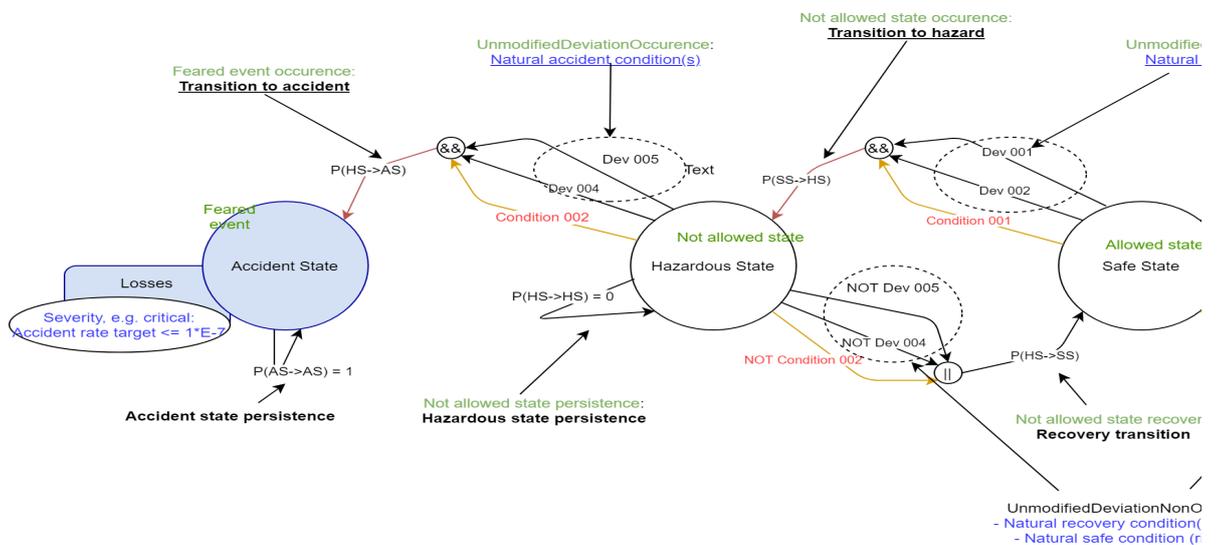
- <not allowed state> recovers to <allowed state> (for situations where the not allowed state clears up by itself)
- <allowed state> persists in <allowed state> (a transition that allows the modelling of the probability of successful n state diagram as a hidden Markov model where the sum of all outgoing transitions from a state has probability 1)
- <not allowed state> persists in <not allowed state> (where the not allowed state is still active but a feared event ha

! To Dos

[Markus Reimann](#) the following is still to be added:

- It has been agreed to model the transition from hazardous state to accident (as per the figure below Dev005 to be included in the description.

ARCH.020 wrt. Safety risks



Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	<p>All the needed safety risk model which have been identified during the deviation analysis are available</p> <p>All the needed business risk model which have been identified during the deviation analysis are available</p> <p>The output views conform to their modelling rules</p> <p>All plausible deviations have been identified and classified in terms of the transitions they could cause</p>
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	<p>RAMS architect</p> <p>Security architect</p>
Uses outputs (Informed)	External stakeholders e.g. independent safety assessor

ARCH.011 Assess operational performance risks of scenario

[SM-2747](#) - Populate Confluence page for activity definition: ARCH.011 Assess operational performance risks of scenario FINISHED

Goal	Specify how operational performance risks will be managed down to an acceptable level
Requirements met by this process step	DB Regulation 0451.0100
Inputs	See sub-activity definitions
Outputs	See sub-activity definitions
Methodology	See sub-activity definitions ARCH.014 Evaluate risks to business effectiveness ARCH.015 Define business risk control measures ARCH.016 Allocate business risk control responsibilities (op. activity to op. entity /actor)
Tools and non-human resources	See sub-activity definitions
Cardinality	Once per operational capability
Completion criteria	See sub-activity definitions
Design review	ARCH.R.1 Operational capability review ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	See sub-activity definitions
Completion criteria evaluated by (Accountable)	See sub-activity definitions
Provides input to /assists (Contributes)	See sub-activity definitions
Reviews (Contributes)	See sub-activity definitions
Uses outputs (Informed)	See sub-activity definitions

ARCH.014 Evaluate risks to business effectiveness

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.015 Define business risk control measures

[SM-2751](#) - Populate Confluence page for activity definition: ARCH.015 Define business risk control measures **CREATED**

Goal	Identify operational requirements needed for business risk mitigation
Requirements met by this process step	None defined
Inputs	AMOD-130 Business loss and risk state model AMOD-031 Unified risk model
Outputs	AMOD-031 Unified risk model (updated)
Methodology	<div data-bbox="304 846 970 965" style="border: 1px solid gray; padding: 5px;"><p> This step must be done jointly between system and RAMS architect.</p></div> <p>For every possible service-affecting deviation</p> <ul style="list-style-type: none">If the risk level of the deviation is not negligible<ul style="list-style-type: none">Decide how the risk level will be reduced to a tolerable levelDocument the rationale for the decisionend if <p>end for</p> <p>Where in the risk model to insert a risk reduction measure</p> <p>There are multiple points in the risk model where it is possible to reduce risk, and there is an order of preference for these. They are listed below in order of highest to lowest preference:</p> <ul style="list-style-type: none">• Reduce the probability of a business risk occurring (i.e., reduce the probability of the transition from allowed state to not-allowed state) (this is the same as increasing the probability that the allowed state persists);• Reduce the probability of a not-allowed state persisting (this is the same as increasing the probability of recovery from the not-allowed state to the allowed state);• Reduce the severity of the loss state(s) that could be caused by the not-allowed states reached from the deviation (by increasing the probability of a transition to a less severe sub-state within the loss state).

It is permissible to define more than one risk reduction measure for any one possible deviation. When doing so, the ["Swiss cheese" model](#) should be considered: that each measure has weaknesses, and so any set of measures should when taken together not have any common weaknesses.

Defining risk reduction measures in the operational activities model

Risk reduction measures in the operational layer of the model are always operational activities. Because the risk reduction measures have been determined in terms of probabilities, there is always a maximum allowable failure rate associated with any operational activity that is a risk reduction measure.

Where a suitable operational activity exists in the model, it is sufficient simply to add the maximum allowable failure rate to the corresponding outgoing interaction of that activity, following the modelling rules for that type of non-functional need.

Where no suitable operational activity exists in the model, one should be created, and the maximum allowable service-affecting failure rate should be added to the corresponding outgoing interaction.

Refer to [ARCH.033 Update operational activities and interactions](#) for instructions on updating or defining new operational activities.

If there exists more than one type of service-affecting failure for any activity, each rate can be added to one outgoing interaction of the operational activity.

 Instructions for updating the model to reflect the risk reduction measures can be found under the process steps of group [ARCH.904](#) - it is recommended to run these process steps in parallel with the risk assessment and mitigation process steps.

Tools and non-human resources	Team for Capella
Cardinality	Once per operational activity
Completion criteria	There exists a set of risk reduction measures in the model for each possible service-affecting deviation to the current operational capability. Rationales have been recorded for each risk reduction.
Design review	ARCH.R.1 Operational feature review ARCH.R.2 Operational review - consolidated

Step done by (Responsible)	System architect RAMS architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None identified

ARCH.017 Assess security risks of scenario

[SM-2753](#) - Populate Confluence page for activity definition: ARCH.017 Assess information security risks of scenario

FINISHED

Goal	Specify how information and physical security risks will be managed down to an acceptable level
Requirements met by this process step	DB Regulation 0451.0100 DB Regulation 0114.0210 (and thereby ISO 27001)
Inputs	See sub-activity definitions
Outputs	See sub-activity definitions
Methodology	See sub-activity definitions ARCH.021 Evaluate security risks ARCH.022 Define operational security measures ARCH.023 Allocate security responsibilities (op. activity to op. entity/actor)
Tools and non-human resources	See sub-activity definitions
Cardinality	Once per operational capability
Completion criteria	See sub-activity definitions
Design review	ARCH.R.1 Operational capability review ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	See sub-activity definitions
Provides input to /assists (Contributes)	See sub-activity definitions
Uses outputs (Informed)	See sub-activity definitions

ARCH.021 Evaluate security risks

Goal	
Requirements met by this process step	RiL 114.0210 05 (7)
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.022 Define operational security measures

[SM-2758](#) - Populate Confluence page for activity definition: ARCH.022 Define operational security measures **CREATED**

Goal	Identify operational requirements needed for security risk mitigation
Requirements met by this process step	RiL 114.0210 05 (15)
Inputs	AMOD-131 Security loss and threat state model AMOD-031 Unified risk model
Outputs	AMOD-031 Unified risk model (updated)

Methodology

 This step must be done jointly between system architect and security expert.

For every possible security-affecting deviation

 If the risk level of the deviation is not negligible

 Decide how the risk level will be reduced to a tolerable level

 Document the rationale for the decision

 end if

end for

Where in the risk model to insert a risk reduction measure

There are multiple points in the risk model where it is possible to reduce risk, and there is an order of preference for these. They are listed below in order of highest to lowest preference:

- Reduce the probability of a threat state occurring (i.e., reduce the probability of the transition from allowed state to threat state) (this is the same as **increasing** the probability that the allowed state persists);
- Reduce the probability of a threat state **persisting** (this is the same as **increasing** the probability of recovery from the threat state to the allowed state);
- Reduce the severity of the loss state(s) that could be caused by the threat states reached from the deviation (by increasing the probability of a transition to a less severe sub-state within the loss state).

It is permissible to define more than one risk reduction measure for any one possible deviation. When doing so, the "[Swiss cheese](#)" model should be considered: that each measure has weaknesses, and so any set of measures should when taken together not have any common weaknesses.

Defining risk reduction measures in the operational activities model

Risk reduction measures in the operational layer of the model are always operational activities. Because the risk reduction measures have been determined in terms of probabilities, there is always a maximum allowable failure rate associated with any operational activity that is a risk reduction measure.

Where a suitable operational activity exists in the model, it is sufficient simply to add the maximum allowable failure rate to the corresponding outgoing interaction that activity, following the modelling rules for that type of non-functional need.

Where no suitable operational activity exists in the model, one should be created, and the maximum allowable security-affecting failure rate should be added to the corresponding outgoing interaction.

Refer to [ARCH.033 Update operational activities and interactions](#) for instructions on updating or defining new operational activities.

If there exists more than one type of security-affecting failure for any activity, each rate can be added to one outgoing interaction of the operational activity.

 Instructions for updating the model to reflect the risk reduction measures can be found under the process steps of group [ARCH.904](#) - it is recommended to run these process steps in parallel with the risk assessment and mitigation process steps.

Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	There exists a set of risk reduction measures in the model for each possible security-affecting deviation to the current operational capability. Rationales have been recorded for each risk reduction.
Design review	ARCH.R.1 Operational capability review ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	System architect Security expert
Provides input to /assists (Contributes)	None identified.
Uses outputs (Informed)	None identified.

ARCH.024 Assess operational safety risks of scenario

- [ARCH.027 Evaluate operational safety risks](#)
- [ARCH.028 Define operational safety measures](#)

ARCH.027 Evaluate operational safety risks

[SM-2763](#) - Populate Confluence page for activity definition: ARCH.027 Evaluate operational safety risks

IN IMPLEMENTATION

[SM-3977](#) - Define the hazardous state likelihood categories

FINISHED

Goal	
Requirements met by this process step	DB Regulation 0451.0100 CSM-SMS guidance 1.1 b), 3.1.1.1 a) CSM-RA guidance 4
Inputs	AMOD-030 Accident and hazard state model AMOD-027 Operational bare business scenario AMOD-029 Operational bare business process
Outputs	AMOD-030 Accident and hazard state model (updated)

Methodology	<p>Once a safety risk state model (including variants) is completed for each hazard-accident scenario related to an Operational Capability then the natural probability shall be defined for each deviation/condition contributing to the transition from the safe state to the hazardous state and from the hazardous state to the accident state.</p> <p>The probability of a deviation/condition occurrence shall be estimated with the support of Railway Experts if needed.</p> <p>If the probability can be estimated precisely, then the probability value shall be inserted manually. Otherwise a frequency category (for transitions between safe state to hazardous state) or a conditional category probability (for transitions between hazardous state and accident state) shall be chosen. In the latter case the probability for the selected category will be automatically derived from the upper boundary (greater value) of the bandwidth defined the "DBS (Safety) Frequency Categories" and the "DBS Conditional Probability Categories" tables in the DBS System Development Safety Plan (in preparation). When neither particular probability is given nor frequency category is selected, 1 shall be considered as default probability.</p> <p>The probability of the transition to Hazard $P(SS>HS)$ and the probability of the transition to accident $P(HS>AS)$ can then be calculated, allowing the resulting accident state probability to be derived as the product of these 2 transition probabilities.</p> <p>The achievement of the accident state probability target is then to be verified. If the natural accident state probability does not meet the target, the needed accident probability reduction factor shall be derived based on the ratio between the accident state target and the natural accident state probability.</p> <p>This risk reduction factor will then have to be allocated on the operational risk control measures to be added to the risk state model and/or on the deviations/conditions of risk state model that need to be constrained (see ARCH.028).</p>
Tools and non-human resources	<p>Team for Capella</p>
Cardinality	<p>Once per operational capability, for each related risk state model.</p> <p>To be reassessed for each project/prototype, as natural probabilities may then differ depending on particular context.</p>
Completion criteria	<p>There exists an estimate of the probability of each deviation /condition of the current operational capability that could lead to a hazardous state occurrence or an accident state occurrence.</p> <p>The output view conforms to its modelling rules</p>
Design review	<p>ARCH.R.1 Operational capability review</p>

Step done by (Responsible)	RAMS architect
Provides input to /assists (Contributes)	<ul style="list-style-type: none">• System architect• Railway expert (for natural probability setting)
Uses outputs (Informed)	System architect

ARCH.028 Define operational safety measures

[SM-2764](#) - Populate Confluence page for activity definition: ARCH.028 Define operational safety measures

IN IMPLEMENTATION

REVIEW

Goal	Identify operational requirements needed for safety risk mitigation
Requirements met by this process step	CSM-SMS guidance 3.1.1.1 c) ISO 15288 (2015) para 6.4.3.3 b) 3
Inputs	AMOD-030 Accident and hazard state model AMOD-031 Unified risk model
Outputs	AMOD-031 Unified risk model (updated) AMOD-028 Operational activities and interaction definitions (single operational capability) (updated)
Methodology	<div data-bbox="306 999 973 1120" style="border: 1px solid gray; padding: 5px;"><p> This step must be done jointly between system and RAMS architect.</p></div> <p>For every possible safety-affecting deviation</p> <ul style="list-style-type: none">If the risk level of the deviation is not negligible<ul style="list-style-type: none">Decide how the risk level will be reduced to a tolerable levelDocument the rationale for the decisionend if <p>end for</p> <p>Note: The above reasoning is no longer valid. the achievement of the accident probability reduction factor/target is now driving the definition of RCM on deviations</p> <p>Note: OMO the "Defining risk reduction measures in the operational activities model" section below may not be needed, some elements inside shall be checked and if needed integrated above and the "Where in the risk model to insert a risk control measure" title below may be also not needed</p> <p>Where in the risk model to insert a risk control measure</p>

There are multiple points in the risk model where it is possible to reduce risk **accident probability target**. These points are always related to the deviations (safety condition) considered in the safety-risk model, and there is an order of preference for these.

They are listed below in order of highest to lowest preference:

- Reduce the probability of a hazardous state occurring (i.e., reduce the probability of the transition from safe state to hazardous state) (this is the same as **increasing** the probability that the safe state persists); - "Risk Prevention Measure".
- Reduce the probability of a accident state occurring (i.e., reduce the probability of the transition from hazardous state to accident state) (this is the same as **increasing** the probability of hazardous state recovery to the safe state). - "Risk Mitigation Measure"
- ~~Reduce the probability of a hazardous state **persisting** (this is the same as **increasing** the probability of recovery from the hazardous state to the safe state); - no longer exists.~~
- Reduce the severity of the accident state(s), **this results in reducing the accident probability target**. This option - implying there is a relevant barrier allowing to alleviate the severity level (e.g. crashworthiness for collision at a moderate speed) - is to be disregarded as far as possible since it generates conditions to be exported and fulfilled, **constraining/restricting the applicability of the risk model and causing the definition and the management of "variant" risk model**. ~~that could be caused by the hazardous states reached from the deviation (by increasing the probability of a transition to a less severe sub-state within the accident state).~~

There are two possible ways for reducing the probability with respect to the two first points above : either by constraining the occurrence of a deviation identified at ARCH. with a lower occurrence probability target **is to the deviation in questions** or by adding an additional mitigation measure (as a new operational activity) incl. probability occurrence target (for its failure) independent to the deviation.

It is permissible to define more than one risk reduction measure for any one possible deviation. When doing so, the "[Swiss cheese model](#)" should be considered: that each measure has weaknesses, and so any set of measures should when taken together not have any common weaknesses.

Defining risk reduction measures in the operational activities model

Risk reduction measures in the operational layer of the model are always operational activities. Because the risk reduction measures have been determined in terms of probabilities, there is always a maximum allowable wrong-side failure rate associated with any operational activity that is a risk reduction measure.

Note: The above reasoning seems no longer valid.

Where a suitable operational activity exists in the model, it is sufficient simply to add the maximum allowable wrong-side failure rate to the corresponding outgoing interaction of that activity, following the modelling rules for that type of non-functional need.

Note: OMO "Where a suitable operational activity exists in the model," shall be integrated above, the rest of the sentence is covered by above changed text "following the modelling rules for that type of non-functional need." is to be added too

Where no suitable operational activity exists in the model, one should be created, and the maximum allowable wrong-side failure rate should be added to the corresponding outgoing interaction.

Note: OMO "Where no suitable operational activity exists in the model, one should be created" shall be integrated above, the rest of the sentence is covered by above changed text

Refer to [ARCH.033 Update operational activities and interactions](#) for instructions on updating or defining new operational activities.

If there exists more than one type of wrong-side failure for any activity, each wrong-side failure rate is added to one outgoing interaction of that activity.

Note: The above reasoning seems no longer valid.



Instructions for updating the model to reflect the risk reduction measures can be found under the process steps of group [ARCH.904](#) - it is recommended to run these process steps in parallel with the risk assessment and mitigation process steps.

Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	There exists a set of risk reduction measures in the model for each possible safety-affecting deviation to the current operational capability. Rationales have been recorded for each risk reduction.
Design review	ARCH.R.1 Operational capability review ARCH.R.2 Operational review - consolidated

Step done by (Responsible)	System architect RAMS architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None identified

ARCH.029 Allocate risk control responsibilities (op. activity to op. entity/actor)

[SM-2765](#) - Populate Confluence page for activity definition: ARCH.029 Define and allocate safety responsibilities (op. activity to op. entity /actor) **IN IMPLEMENTATION**

Goal	Identify the operational entities or actors responsible for risk control measures.
Requirements met by this process step	CSM-SMS guidance 1.1 c), 2.3.1, 2.3.4, 3.1.1.1 e) (for safety) RiL 114.0201 (for security)
Inputs	Definitions of operational risk control measures (as produced by ARCH.028 , ARCH.015 and ARCH.022) and modelled in the various views of ARCH.904 AMOD-022 Enterprise & environment definition
Outputs	AMOD-024 Operational activity definition and allocation
Methodology	<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 10px;"> <p> This step should be done jointly by the system architect, security architect and RAMS architect.</p> </div> <p>For each operational activity that is defined as a risk reduction measure</p> <p style="padding-left: 40px;">allocate the operational activity to the appropriate operational entity or actor</p> <p>end for</p> <p>This activity should be reasonably straightforward. The core objective is to distinguish risk control measures that are completely external to the wider system of interest (in which case, they are less likely to be supported by SysABB) from measures that the wider system of interest is responsible for - in which case, they could become wholly or partly requirements on SysABB.</p> <p>If the activity could be done by more than one operational entity, then it should be replicated and a replica allocated to each actor.</p> <p>If the activity is too big to be allocated to one operational entity then it should be split and the parts allocated to the appropriate operational entity (although it is acceptable to leave it as a duty of an operational entity even if they cannot do it all by themselves - because this kind of operational activity can be pushed down by ARCH.049 as a system capability and the internal functions distributed between actors and the system).</p> <p>In general, the preference should be to allocate responsibility to existing actors/entities rather than creating new ones.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	All the operational activities identified as risk control measures for this operational capability have been allocated to an operational actor or entity.

Design review	ARCH.R.1 Operational capability review ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	System architect RAMS architect Security architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None identified

ARCH.175 Produce consolidated operational deviation analysis report

[SM-5395](#) - Populate Confluence page for process step ARCH.175 Produce consolidated operational deviation analysis report CREATED

Goal	<i>high-level statement of the engineering objective being addressed, including reference to standards/legislation if that is relevant</i>
Requirements met by this process step	<i>cite requirements from standards and legislation (by standard/document number and paragraph number only, no complete copy and paste)</i>
Inputs	<i>list of all input information needed for this process step to be carried out; for model views, add a hyperlink to the model view description; documents, published info, websites, knowledge from particular roles are all admissible.</i>
Outputs	<i>list of all output artefacts this process should produce; for model views, add a hyperlink to the model view description; for documents, add a link to the document template or to the document publication area if it is directly generated from Confluence;</i>
Methodology	<i>state the methodology as a list of instructions; this information forms the Description field of a ticket for this process to be done, so it must be sufficient for someone to follow the process, even if they have not done it before.</i>
Tools and non-human resources	<i>list of all tools (physical and software) needed for this process, plus any non-obvious resources that need to be planned (such as access to particular non-DBB individuals, operational areas, site visits...)</i>
Cardinality	<p><i>state how many times this process step is expected to be carried out - e.g.</i></p> <ul style="list-style-type: none"> • <i>one-off</i> • <i>once per <model element></i> • <i>once per baseline</i> <p><i>etc.</i></p>
Completion criteria	<p><i>state the conditions that must be TRUE before this process step can be considered complete.</i></p> <p><i>These should be inserted in the Acceptance Criteria of a Jira ticket for this process step to be done.</i></p>
Design review	<i>Link to the corresponding design review where the completion of this activity is evaluated.</i>
Step done by (Responsible)	<p><i>Identify the role that is responsible for organising this process step and producing the output;</i></p> <p><i>This is the Assignee in Jira.</i></p>
Provides input to /assists (Contributes)	<i>Identify the roles that assist in the work or provide input information</i>
Uses outputs (Informed)	<i>Identify the roles and/or process areas that make use of this information outside of the ARCH process area;</i>

ARCH.904 Incorporate risk control measures in the operational processes

- [ARCH.031 Create operational scenario with risk control measures](#)
- [ARCH.032 Create operational process with risk control measures](#)
- [ARCH.033 Update operational activities and interactions](#)
- [ARCH.147 Update operational data](#)
- [ARCH.160 Update operational states](#)
- [ARCH.162 Update operational activity to state mapping](#)
- [ARCH.178 Update dependencies and pre/post conditions](#)

ARCH.031 Create operational scenario with risk control measures

[SM-4350](#) - Update process steps ARCH.031 and ARCH.032 to emphasise need to catch all non-success logical paths FINISHED

Goal	Define how an operational capability will be fulfilled whilst managing identified risks to an acceptable level.
Requirements met by this process step	None defined
Inputs	<p>AMOD-028 Operational activities and interaction definitions (single operational capability)</p> <p>AMOD-027 Operational bare business scenario</p> <p>(reference only):</p> <p>AMOD-107 Safety hazard log</p> <p>AMOD-108 Security issue log</p> <p>AMOD-109 Business risk log</p>
Outputs	AMOD-033 Operational business scenario with risk control measures
Methodology	<p>Begin by setting out the interactions of the scenario for the bare business scenario.</p> <p>Then introduce any new interactions identified through the risk assessment, at the correct point in the scenario.</p> <p>Ensure that all the non-success paths identified in the deviation analysis are now represented on scenario.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	<p>The output view conforms to its modelling rules</p> <p>All relevant risk mitigations have now been incorporated in the scenario as described in the output view.</p>
Design review	ARCH.R.1 Operational capability review
Step done by (Responsible)	System architect
Provides input to/assists (Contributes)	None identified
Uses outputs (Informed)	RAMS manager

ARCH.032 Create operational process with risk control measures

Goal	Define how an operational capability will be fulfilled whilst managing identified risks to an acceptable level.
Requirements met by this process step	None identified (although this is part of the process for allocating risk control responsibility)
Inputs	<p>AMOD-028 Operational activities and interaction definitions (single operational capability)</p> <p>AMOD-029 Operational bare business process</p> <p>(reference only):</p> <p>AMOD-107 Safety hazard, security issue, business risk log</p> <p>AMOD-108 Security issue log</p> <p>AMOD-109 Business risk log</p>
Outputs	AMOD-035 Operational business process with risk control measures
Methodology	<p>Begin by setting out the activities and interactions of the bare business process, without linking them together with sequences.</p> <p>Then introduce any new activities and interactions identified through the risk assessment.</p> <p>Ensure that all non-success paths identified during the deviation analysis are now represented on the diagram.</p> <p>When all activities are placed sensibly in order, complete the sequence links.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	<p>The output view conforms to its modelling rules</p> <p>All relevant risk mitigations have now been incorporated in the process as described in the output view.</p>
Design review	ARCH.R.1 Operational capability review
Step done by (Responsible)	System architect
Provides input to/assists (Contributes)	None identified
Uses outputs (Informed)	RAMS manager (potentially)

ARCH.033 Update operational activities and interactions

Goal	Define new operational activities and interactions needed for risk mitigation
Requirements met by this process step	None defined
Inputs	AMOD-028 Operational activities and interaction definitions (single operational capability) AMOD-107 Safety hazard, security issue, business risk log AMOD-108 Security issue log AMOD-109 Business risk log
Outputs	AMOD-028 Operational activities and interaction definitions (single operational capability) (updated)
Methodology	<p>For every entry in the risk logs,</p> <ul style="list-style-type: none"> where the risk mitigation specifies a new operational activity, <ul style="list-style-type: none"> create a new operational activity; create appropriate interactions between the new operational activity and existing activities where the risk mitigation specifies a non-functional requirement against an operational activity on the current diagram <ul style="list-style-type: none"> create a new non-functional requirement with content matching the risk mitigation; link the non-functional requirement to the activity specified in the risk mitigation.
Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	<p>The output view conforms to its modelling rules.</p> <p>All relevant mitigations from all risk logs have been captured in the output view.</p>
Design review	
Step done by (Responsible)	System architect
Provides input to/assists (Contributes)	None specified
Uses outputs (Informed)	None directly

ARCH.147 Update operational data

Goal	Update operational data that are required to be involved in risk mitigation.
Requirements met by this process step	ISO 15288 6.3.4 d) 2
Inputs	AMOD-110 Operational exchange items [O.CDB] AMOD-105 Operational data objects [O.CDB]
Outputs	AMOD-110 Operational exchange items [O.CDB] AMOD-105 Operational data objects [O.CDB] Updated allocation of operational exchange items to interactions between operational activities.
Methodology	<p>For every entry in the risk logs,</p> <ul style="list-style-type: none"> where the risk mitigation specifies the need for additional operational data <ul style="list-style-type: none"> create new elements (exchange items, data object as necessary) allocate new elements to appropriate interactions between operational activities where the risk mitigation specifies a constraint on operational data <ul style="list-style-type: none"> update the exchange items, data objects as necessary review allocation of updated elements to ensure that elements allocating elements ist still supported. examples are among others constraints on data separation, redundancy. <p>As guidance see methodology in ARCH.153 Model operational data.</p>
eTools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	Output model views are updated in accordance with modelling rules. All relevant mitigations from all risk logs have been captured in the output view.
Design review	ARCH.R.1 Operational capability review ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	System architect
Provides input to/assists (Contributes)	None specified
Uses outputs (Informed)	None directly

ARCH.160 Update operational states

[SM-3523](#) - Create Confluence page for activity definition ARCH.160 **FINISHED**

Goal	Refine understanding of state-based differences in behaviour of operational entities/actors
Requirements met by this process step	ISO 15288 6.3.4 d) 2
Inputs	AMOD-022 Enterprise & environment definition AMOD-023 Operational entity/actor states
Outputs	AMOD-023 Operational entity/actor states (updated/new)
Methodology	<p>During risk assessment for an operational scenario, new states may be identified for operational entities or actors, that were not known during ARCH.006 or ARCH.159.</p> <p>Refer to ARCH.006 for basic principles on how to approach the modelling of operational states.</p> <p>Either update existing instances of view AMOD-023 or, where no statefulness was previously identified for an operational actor/entity, create a new one.</p> <p>In both cases, follow the modelling rules for AMOD-023.</p> <p>This activity should be done in parallel with the initial analysis of an operational capability.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability

Completion criteria	<p>(The output views conform to their modelling rules</p> <p>AND</p> <p>Statefulness that was identified during the analysis of the operational capability has been captured in the model)</p> <p>OR</p> <p>No further statefulness was identified during the analysis of this operational capability</p>
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None identified

ARCH.162 Update operational activity to state mapping

[SM-3524](#) - Create Confluence page for activity definition ARCH.162 FINISHED

Goal	Refine understanding of state-based differences in behaviour of operational entities/actors
Requirements met by this process step	ISO 15288 6.3.4 d) 2
Inputs	AMOD-023 Operational entity/actor states AMOD-033 Operational business scenario with risk control measures AMOD-028 Operational activities and interaction definitions (single operational capability)
Outputs	Output is not directly visible on a view, it is a set of relationships in the model viewable via the "Available states" property of the activities.
Methodology	Where the risk assessment: <ul style="list-style-type: none">identified new operational activities necessary for this operational capabilityoridentified new states exhibited by entities/actors involved in this capability repeat the steps from ARCH.161 with the new activities and/or states.
Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	For all the involved entities for this capability, the operational activities associated with each state have been correctly assigned.
Design review	
Step done by (Responsible)	System architect
Provides input to/assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.178 Update dependencies and pre/post conditions

Goal	<p>Define new dependencies between OCOI and other OCS</p> <p>Refine the pre and post condition with further description</p>
Requirements met by this process step	<p>CSM-SMS guidance 1.1a)</p> <p>EN 50126-1 7.2.2 a)</p> <p>ISO 15288 6.4.1.3 b 2)</p>
Inputs	<p>AMOD-028 Operational activities and interaction definitions (single operational capability)</p> <p>AMOD-035 Operational process (business process with risk control measures)</p>
Outputs	<p>AMOD-137 Single operational capability context</p>
Methodology	<p>By now, OAIB and OPD with risk control measures have been elaborated and you have a clear vision about the boundary of your operational capability of interest (OCOI).</p> <p>The COC can then be updated in terms of dependencies along with the pre and post condition. This time, it is necessary to align with the owner of the other OCs connected to the OCOI.</p> <p>The end result should be that an OC's start conditions correspond to the end conditions of other OCs (and vice versa). Overlaps should be avoided. Gaps should be checked to ensure that nothing important has been missed.</p> <p>Depending of the number of dependencies, this process step can require a lot of time to align with the other OCs</p> <p>Remark: If there are Entities/Actors connected to the OCOI, you can decide to remove those if this makes the diagram clearer.</p>
Tools and non-human resources	<p>Team for Capella</p>
Cardinality	<p>Once per Operational Capability</p>
Completion criteria	<p>The output views conform to their modelling rules.</p>
Design review	<p>ARCH.R.1 Operational capability review</p> <p>ARCH.R.2 Operational review - consolidated</p>
Step done by (Responsible)	<p>System Architect</p>
Provides input to /assists (Contributes)	<p>Systems Engineer</p>
Uses outputs (Informed)	<p>None</p>

ARCH.905 Consolidate the operational needs

- [ARCH.121 Consolidate operational activities and interactions](#)
- [ARCH.034 Produce concept of operations \(CONOPS\)](#)
- [ARCH.140 Consolidate the allocation of operational activities](#)
- [ARCH.183 Consolidate operational data](#)

ARCH.121 Consolidate operational activities and interactions

[SM-3041](#) - Populate Confluence page for activity definition: ARCH.121 **FINISHED**

Goal	Achieve an elegant and optimal operational activity architecture
Requirements met by this process step	None listed
Inputs	AMOD-028 Operational activities and interaction definitions (single operational capability) AMOD-035 Operational business process with risk control measures AMOD-033 Operational business scenario with risk control measures
Outputs	AMOD-101 Consolidated operational activities & interactions
Methodology	<p>The results of scenario-based analysis of operational activities will be mixed and may have gaps. There will be a large number of operational activities but there will be little structure to assist in understanding the set of activities as a whole. The set of operational activities and interactions needs to be consolidated. There are 3 main points to consider:</p> <ul style="list-style-type: none">• Grouping and structure The activities should be grouped by common themes and consideration should be given to creating a hierarchy of activities in order to nest the leaf-level activities. More material on clustering of activities is given in section 5 of the FAT material that was used in 2019.• Gaps and duplicates <p>Any hierarchy should be as flat as possible. Layering for the sake of layering should be avoided. It is not necessary for every leaf level activity to be on the same layer of the hierarchy. Use just enough layering to help readers navigate the model, no more.</p>

When looking at a group of activities, it may become obvious that there is one missing from the set. This may mean that an operational process has a missing part, or that a capability is missing. If it appears this is the case, go back and amend the capability/process. Do not simply add the activity without it appearing in at least one process. This would mean there is no known business need in the model for the activity.

Since operational activities are created by each group working on operational capabilities, it is possible that one group may have missed the fact that the other group has already created an activity. There may be duplicates. Where duplicates exist, they should be eliminated. Ensure that a duplicate is replaced by the master item on all views where it appears, BEFORE deleting the duplicate. On deletion, the duplicate should no longer be related to any other model element.

- Flow optimisation

There should be an overall pattern in the flow through the operational activities. This may only be visible when looking from a high level down onto the diagram. It should be possible to differentiate between feed-forward and feedback flows. Where possible, the diagram should be arranged so that feedback interactions flow in the opposite direction to feed-forward interactions.

Tools and non-human resources	Team for Capella
Cardinality	One-off, with revisions permitted
Completion criteria	<p>The produced view is compliant with its modelling rules.</p> <p>The complete set of operational activities and interactions is safe enough to try.</p>
Design review	ARCH.R.2 Operational review - consolidated

Step done by (Responsible)	Operational concept architect
Provides input to /assists (Contributes)	Any contributor to previous operational modelling steps, in the case of specific questions.
Uses outputs (Informed)	None directly

ARCH.034 Produce concept of operations (CONOPS)

Goal	Communicate the concept of DBB operations to stakeholders inside and outside the DBB team.
Requirements met by this process step	None defined
Inputs	AMOD-010 Trade space assessment AMOD-012 System lifecycle model AMOD-021 Operational capabilities definition AMOD-022 Enterprise & environment definition AMOD-023 Operational entity/actor states AMOD-025 Abstract concepts
Outputs	AMOD-037 DBB concept of operations
Methodology	<p>The CONOPS documentation will be generated automatically from a combination of already-existing model content and hand-written "boilerplate" standard text.</p> <p>Some of the sections of boilerplate are generated by other activities in the process.</p> <p>Missing sections of boilerplate defined in the template for AMOD-037 should be generated in this activity.</p>
Tools and non-human resources	<p>Team for Capella</p> <p>CONOPS generation scripts and templates</p>
Cardinality	One-off with revisions
Completion criteria	<p>The CONOPS output documentation contains all required content as defined in the template.</p> <p>The CONOPS output documentation is free of automatic generation errors.</p>
Design review	
Step done by (Responsible)	<p>Operational concept architect</p> <p>Engine room (generation)</p>
Provides input to /assists (Contributes)	None identified

**Uses outputs
(Informed)**

All DBB colleagues

Suppliers of DBB equipment

Safety assessors

Stakeholders within wider DB

Railway undertakings

ARCH.140 Consolidate the allocation of operational activities

[SM-3060](#) - Populate Confluence page for activity definition: ARCH.140

FINISHED

Goal	Assure that all operational activities are allocated to the right entities
Requirements met by this process step	None defined
Inputs	<p>AMOD-101 Consolidated operational activities and interactions</p> <p>AMOD-024 Operational activity definition and allocation</p> <p>AMOD-035 Operational business process with risk control measures</p> <p>AMOD-033 Operational business scenario with risk control measures</p>
Outputs	AMOD-024 Operational activity definition and allocation (updated)
Methodology	<p>If not already complete, populate the view with all the operational entities and actors.</p> <p>Add the operational activities currently allocated to the operational entities and actors.</p> <p>NB: most of these should already have been allocated when they were included in operational entity scenarios (viewpoint AMOD-033). There should be few operational activities remaining that are not allocated, except for those that are clustered at a higher level from the leaf-level activities that were actually used in the OESs and the OPDs.</p> <p>We are not concerned with allocating any activity that is not leaf-level, as long as all leaf-level activities are allocated to one entity/actor.</p> <p>For each leaf-level activity that is not yet allocated</p> <ul style="list-style-type: none"> • Determine if this activity is ever used in an OES and/or an OPD. • If it is used, determine (with the owner of the corresponding capability) which actor or entity should carry out this activity; • else if not, make it a candidate for deletion <p>Arrange the output view in good order. This will be a large view and may not be easily printable.</p>

Tools and non-human resources	Team for Capella
Cardinality	One-off with revisions permitted.
Completion criteria	The output view conforms to its modelling rules. The allocation of operational activities is an accurate representation of the operational vision.
Design review	ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	Operational concept architect
Provides input to /assists (Contributes)	System architect owning a specific capability that uses an operational activity
Uses outputs (Informed)	All engineering roles (could be published as a poster for all to gain awareness)

ARCH.183 Consolidate operational data

[RCAMT-369](#) - Refine data model rules so that data can be reused on lower levels and not need to be automatic transferred FINISHED

Goal	Achieve elegant and optimal operational data diagrams.
Requirements met by this process step	None.
Inputs	AMOD-101 Consolidated operational activities & interactions AMOD-105 Operational data objects [O.CDB] AMOD-110 Operational exchange items [O.CDB]
Outputs	AMOD-025 Abstract concepts (new full scope viewpoint) AMOD-105 Operational data objects [O.CDB] AMOD-110 Operational exchange items [O.CDB]
Methodology	<p>At this point, operational feature review have been concluded and consolidation of operational activities and interaction is being done (ARCH.121) which includes to optimise flows and this includes to optimise the operational data as well. This is why this process is required.</p> <p>The consolidation should be done only for abstract concepts and operational data and will be the end of the activity started in the process step ARCH.008 Define abstract concepts and based on what has been done on each OC data elaboration.</p> <p>This process step includes to create a new viewpoint AMOD-025 but focusing on full-scope. This means a consolidation of the AMOD-025 (limited scope) done in previous process steps should be elaborated.</p> <p>The operational data will be consolidated but it will be done on the previous viewpoint (limited scope).</p> <p>Consolidation should aim to achieve:</p> <ul style="list-style-type: none"> ■ efficient model of abstract concepts and data ■ elimination of duplicates ■ realisation of multiple needs with reused items, where appropriate ■ elimination of gaps in definitions and coverage <p>For more guidance see methodology on ARCH.153 Model operational data.</p>
Tools and non-human resources	Team for Capella
Cardinality	One-off, with revisions permitted
Completion criteria	<p>The produced view is compliant with its modelling rules.</p> <p>The complete set of operational data is safe enough to try.</p>
Design review	ARCH.R.2 Operational review - consolidated
Step done by (Responsible)	Operational concept architect

Provides input to/assists (Contributes)	Any contributor to previous operational modelling steps, in the case of specific questions.
Uses outputs (Informed)	None directly

ARCH.910 Determine the system requirements

- [ARCH.911 Identify the system's contribution to the operational needs](#)
 - [ARCH.035 Identify constraints on the system solution](#)
 - [ARCH.043 Define set of potential system actors](#)
 - [ARCH.049 Define set of potential system capabilities/missions](#)
 - [ARCH.039 Define set of potential system and actor functions](#)
 - [ARCH.036 Identify all alternative system boundaries](#)
 - [ARCH.042 Select system boundary \(or set of variants\)](#)
- [ARCH.912 Finalise the system context and constraints](#)
 - [ARCH.048 Identify system implementation constraints](#)
 - [ARCH.045 Define complete set of system-level actors](#)
 - [ARCH.119 Define consolidated set of system capabilities/missions](#)
 - [ARCH.041 Transfer or trace upper level model elements to system level](#)
 - [ARCH.044 Define requirements from non-actor stakeholders](#)
 - [ARCH.051 Align the system boundary, capabilities and actors with collaboration projects](#)
- [ARCH.913 Analyse the system capabilities to determine detailed system needs](#)
 - [ARCH.179 Complete the definition of the system capability of interest](#)
 - [ARCH.088 Define system functions and functional exchanges](#)
 - [ARCH.052 Create initial system exchange scenarios](#)
 - [ARCH.053 Create initial system functional chains](#)
 - [ARCH.054 Model data flowing between system functions](#)
 - [ARCH.055 Model states on system level](#)
 - [ARCH.056 Map system functionality to states](#)
 - [ARCH.158 Model external interface layers](#)
 - [ARCH.057 Model non-payload data on external interfaces](#)
 - [ARCH.058 Define measures of performance](#)
- [ARCH.915 Assess & mitigate the risks of system failure](#)
 - [ARCH.065 Derive safety target to system functions and define additional risk control measure needed](#)
 - [ARCH.066 Identify system level deviations](#)
 - [ARCH.067 Populate the fault tree](#)
 - [ARCH.068 Assess deviation probability \(external constraints\)](#)
 - [ARCH.069 Add system level risk measures](#)
 - [ARCH.070 Calculate system deviation probability](#)
- [ARCH.916 Incorporate risk control measures in system scenarios/functional chains](#)
 - [ARCH.141 Update system functions with risk control measures](#)
 - [ARCH.142 Create system exchange scenarios with risk control measures](#)
 - [ARCH.143 Create system functional chains with risk control measures](#)
 - [ARCH.149 Update system data with risk control measures](#)
 - [ARCH.150 Update system interface model with risk control measures](#)
 - [ARCH.163 Update system states](#)
 - [ARCH.164 Update function mapping to system states](#)
- [ARCH.914 Produce the unregulated system documents](#)
 - [ARCH.062 Produce the concept of use \(CONUSE\)](#)
 - [ARCH.061 Produce the concept of employment \(CONEMP\)](#)
- [ARCH.917 Finalise the system requirements](#)
 - [ARCH.152 Consolidate system functionality](#)
 - [ARCH.157 Consolidate system data](#)
 - [ARCH.081 Update constraints on system solution](#)
 - [ARCH.109 Consolidate traceability between model elements at system level and model elements at operational level](#)
 - [ARCH.082 Trade off system requirements and constraints](#)
 - [ARCH.083 Agree the system requirements with stakeholders](#)

ARCH.911 Identify the system's contribution to the operational needs

- [ARCH.035 Identify constraints on the system solution](#)
- [ARCH.043 Define set of potential system actors](#)
- [ARCH.049 Define set of potential system capabilities/missions](#)
- [ARCH.039 Define set of potential system and actor functions](#)
- [ARCH.036 Identify all alternative system boundaries](#)
- [ARCH.042 Select system boundary \(or set of variants\)](#)

ARCH.035 Identify constraints on the system solution

[SM-2771](#) - Populate Confluence page for activity definition: ARCH.035 Identify constraints on the system solution FINISHED

Goal	Ensure the system designed is fit for its purpose and its environment
Requirements met by this process step	ISO 15288 6.4.2.3 d 1) EN 50126-1 7.3.2 b), 7.3.7 c)
Inputs	<p>AMOD-010 Trade space assessment</p> <p>AMOD-024 Operational activity definition and allocation</p> <p>Knowledge of responsible and contributing roles</p> <p>Railway domain source material from openly available sources</p> <p>Contemporary CCS project material within DB (DSTW/ETCS etc.)</p> <p>Knowledge of system borders of collaborative projects (optional)</p>
Outputs	AMOD-116 System implementation constraints
Methodology	<p>The question to be answered is "how do the trade space considerations practically constrain our choice of system boundary and our implementation of that system?".</p> <p>This task must be bounded carefully to avoid spending too much time in one go. It is not possible to capture every possible consideration just by brainstorming. Something like the "80:20" rule should be applied when determining the depth to which the constraints should be analysed.</p> <p>The input information will be quite high-level and may not be focused on potential system boundaries. Where content from collaborative projects (e.g. RCA) is planned to be implemented in the model under consideration, the system border of the respective project will be a constraint for this step.</p> <p>This activity must convert this input information into a set of practical considerations and constraints for setting the system boundary and implementing the system physically on the infrastructure.</p> <p>The considerations identified in the trade space assessment should all have some meaningful constraint on the physical system implementation.</p> <p>Brainstorming could be used, with domain experts, to identify this meaning from the trade space assessment.</p> <p>Material on "operational" (usage) target pictures from projects such as ETCS/DSTW might be useful as comparators.</p>

Tools and non-human resources	General brainstorming materials Team for Capella (to be decided, depending on format of AMOD-116)
Cardinality	One-off, with possible revisions.
Completion criteria	The set of identified system implementation constraints is safe enough to try.
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Railway operations expert
Uses outputs (Informed)	None directly

ARCH.043 Define set of potential system actors

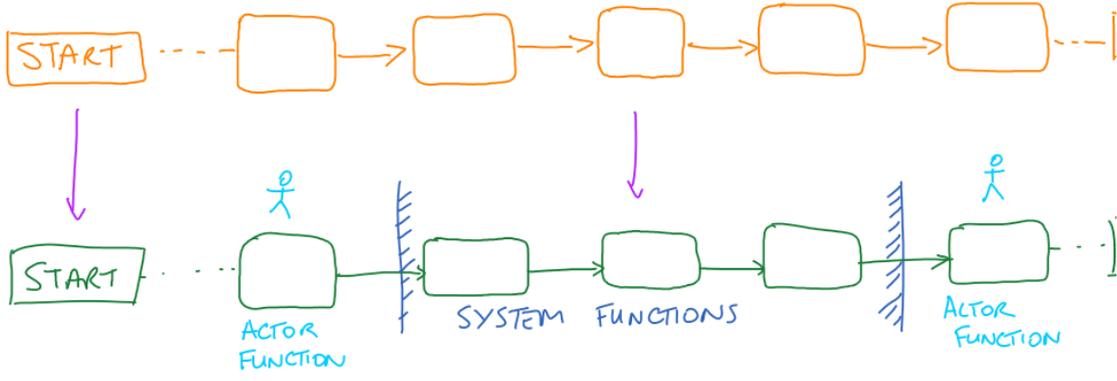
[SM-2779](#) - Populate Confluence page for activity definition: ARCH.043 Select operational actors for transition/tracing to system level

FINISHED

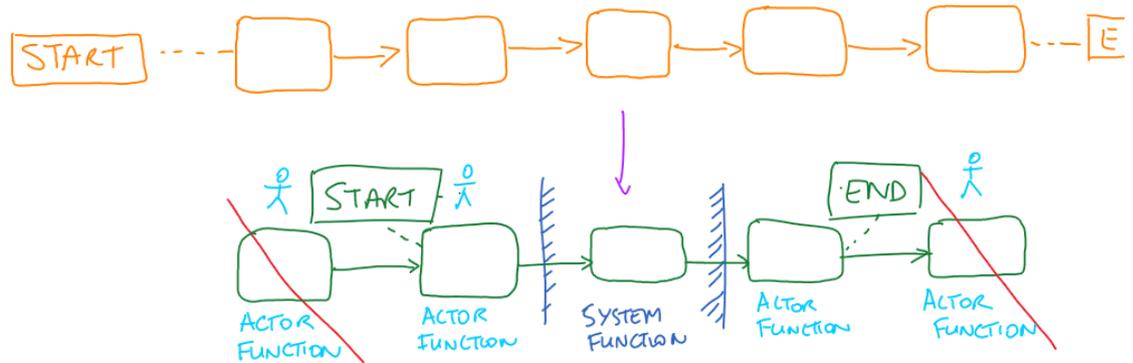
Goal	Identify and model the operational entities that actually interact with the system
Requirements met by this process step	ISO 15288 6.4.2.3 a 1)
Inputs	<p>AMOD-024 Operational activity definition and allocation</p> <p>System-level actors from collaborative projects (optional)</p>
Outputs	<p>System actor objects in the model</p> <p>(visualised when AMOD-044 Actor functions (system level) is created during ARCH.040 Transfer OA to system level function, allocate entirely to actor)</p> <p>AND</p> <p>AMOD-122 Operational entity/actor to system level transfer rationales</p>
Methodology	<p>FOR each operational entity or operational actor</p> <p> IF (at least one of its allocated operational activities is not to be automated AND that same activity is expected to interact with system functionality (that is, there is an interaction with an operational activity that has been selected to become a system function or system capability))</p> <p> IF a system actor already exists that is suitable to represent this operational entity/actor</p> <p> create a traceability relationship from the system actor to the operational entity/actor via the system actor's "realised operational entities"/"realised operational actors" properties</p> <p> ELSE</p> <p> execute the "Create system actor" automatic transition on the operational entity or operational actor.</p> <p> END IF</p> <p> END IF</p> <p>END FOR</p> <p>FOR each operational entity or operational actor that is transferred to system level as actor</p> <p> a rationale for the decision how to transfer it shall be documented in AMOD-122. E.g. put a rationale why one system actor is traced to one or more operational entity / operational actor or why an operational actor became a system actor.</p> <p>END FOR</p> <p>If relevant system actors are defined in a collaborative project (e.g. RCA), they may be transferred and utilised, if appropriate.</p>

Tools and non-human resources	Team for Capella
Cardinality	One-off with revisions permitted
Completion criteria	For each operational entity that interacts with the system an actor at system level has been created and this actor is traced to the original operational entity.
Design review	ARCH.R.4 System review (consolidated)
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.049 Define set of potential system capabilities/missions

Goal	Set the scope of the system's capabilities
Requirements met by this process step	None defined
Inputs	AMOD-021 Operational capabilities definition System functional chains from collaborative projects (e.g. RCA)
Outputs	AMOD-120 Operational capability to system capability transfer rationales
Methodology	<p>For each operational capability in the model, decide whether they should be represented as a system capability, a system mission or have no representation.</p> <p>Criteria:</p> <p>Choose transition to system capability if:</p> <ul style="list-style-type: none"> The operational capability will be achieved fully or in part by the system; <p>AND</p> <ul style="list-style-type: none"> it makes logical sense for the complete story (the chain of events between the starting and ending conditions of the operational capability) to remain in one piece at system level <p>(in other words, looking at the interactions at the very start and end of the operational process, and at the decisive system boundary, if all those interactions will become functional exchanges between a system actor and the system)</p>  <p>In the event that a system capability from a collaborative project (e.g. RCA) corresponds to an operational capability within the system under consideration (e.g. DBS), it can be decided not to perform the transfer of the operational capability to the system level. Instead, the system functional chains from the collaborative projects will be transferred and assigned to appropriate system capabilities. In such a case traceability must be established from the system capability to which a transferred functional chain has been assigned to its corresponding operational capability, rather than from the operational level to the system level.</p> <p>Else, choose transition to system mission if</p> <ul style="list-style-type: none"> The operational capability will be achieved in part by the system; <p>AND</p> <ul style="list-style-type: none"> Given the system boundary, the starting and ending conditions of the operational capability will not be the same as the starting and ending conditions of a system capability

(in other words, looking at the interactions at the very start and end of the operational process, if **not all** those int will become functional exchanges between a system actor and the system).



Else, choose no transition if the operational capability is achieved **entirely without** use of the system (this should be rare, not impossible, depending on the decisions made on the system boundary).

Document the decision and rationale for each operational capability.

Tools and non-human resources	Team for Capella
Cardinality	Once per operational capability
Completion criteria	The output view contains a valid decision and rationale for each operational capability in the model.
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Lead system architect
Uses outputs (Informed)	None directly

ARCH.039 Define set of potential system and actor functions

Goal	Refine the functional scope of the system
Requirements met by this process step	CSM-SMS Guidance 3.1.1.1 c), 3.1.5
Inputs	AMOD-101 Consolidated operational activities and interactions System-level scenarios from collaborative models (optional)
Outputs	AMOD-121 Operational activity to system level transfer rationales
Methodology	<p>For each operational activity, decide whether the activity should be represented at system level as a system capability, a system function or an actor function.</p> <p>Select system capability if</p> <ul style="list-style-type: none"> it is expected that the activity will be broken out into a set of system and actor functions AND the activity is expected to be triggered by a system actor AND the activity is expected to yield an observable valuable result to one or more system actors AND the activity is not already contained within the process of an operational capability that has been directly traced down as a system capability <p>Else, select system function if</p> <ul style="list-style-type: none"> it is expected that the activity will be entirely done by the system AND the activity is not triggered by a system actor XOR the activity does not yield an observable valuable result to one or more system actors <p>Else, select actor function if it is expected that the activity will be done entirely by one and only one actor.</p> <p>Else, if system-level scenarios, corresponding to operational activities in the model of the system under consideration, have been defined in collaborative projects (e.g. RCA), choose to transfer them to the system level of the system under consideration.</p> <p>Document the rationale for the system-level representation of each operational activity as AMOD-121.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per operational activity

Completion criteria	The output view contains a valid decision and rationale for each operational activity in the model.
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Lead system architect
Uses outputs (Informed)	None directly

ARCH.036 Identify all alternative system boundaries

[SM-2772](#) - Populate Confluence page for activity definition: ARCH.036 Identify all alternative system boundaries FINISHED

Goal	Enable decisions to be made about whether functionality is to be included inside the system, or left to external actors.
Requirements met by this process step	ISO 15288 6.4.1.3 c 2)
Inputs	<p>AMOD-024 Operational activity definition and allocation</p> <p>Contemporary CCS project material relevant to responsible enterprise (specifically about system capabilities and activities, system target picture etc.)</p>
Outputs	AMOD-041 System boundary options
Methodology	<p>The operational activities produced at the operational layer are all necessary for the operational capabilities to be achieved. Therefore, they must be implemented in some way at the system layer of the model - either as system capabilities (with the exact functionality and border to be determined later), system functions, or actor functions.</p> <p>For each operational activity, determine if there is any question around whether or not this should be performed by the system under consideration.</p> <p>In case system activities are to be adapted from collaborative projects; all activities allocated to the system within the collaborative project are to be assigned to the system under consideration of the enterprise project. Those system activities allocated to external actors within the collaborative project must be assigned either to the system under consideration of the enterprise project, or to actors external to said system.</p> <p>Where there is doubt on the allocation of the activity between the system or actors, sketch the options, showing each alternative with the candidate functionality of the system and the candidate functionality of actors. Produce one sketch per doubted operational activity, or per themed group. Do not attempt to sketch all permutations of all alternative function allocations together. Each specific decision should be handled individually.</p>
Tools and non-human resources	Any drawing tool
Cardinality	One-off, with revisions permitted
Completion criteria	<p>The produced views comply with their modelling rules</p> <p>For each individual allocation question, the set of alternatives produced is complete and reasonable</p>
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Railway operations expert
Uses outputs (Informed)	None directly

ARCH.042 Select system boundary (or set of variants)

[SM-2778](#) - Populate Confluence page for activity definition: ARCH.042 Select system boundary (or set of variants) **FINISHED**

[SM-3430](#) - Add DAR artefacts to tradeoff tasks **FINISHED**

Goal	Determine the functional scope of the system
Requirements met by this process step	ISO 15288 6.4.3.3 a 1), 6.4.1.3 d), 6.4.4.3 c) 1 EN 50126-1 7.3.2.1 b)
Inputs	AMOD-041 System boundary options AMOD-010 Trade space assessment AMOD-116 System implementation constraints
Outputs	AMOD-046 Selected system boundary tradeoff record AMOD-142 Single subsystem definition
Methodology	<p>Trade-off methods should be used to determine the optimal alternative for each of the choices of system scope that exist.</p> <p>A range of trade-off methods is documented here: Techniques, methodology and guidance</p> <p>The DAR method is recommended and the template for AMOD-142 is constructed on this basis.</p> <p>The criteria for the assessment of the options should be derived from the trade space assessment and/or the system implementation constraints (both of which can be interpreted as quality factors that the system boundary options will achieve to varying degrees).</p> <p>It should be possible to reuse criteria for multiple decisions.</p>
Tools and non-human resources	Technique-dependent
Cardinality	Once per individual system boundary decision
Completion criteria	The selected system boundary definitions are safe enough to try.
Design review	ARCH.R.4 System review - consolidated
Step done by (Responsible)	Lead system architect

Provides input to /assists (Contributes)	System architect Any other role considered helpful by the responsible role
Uses outputs (Informed)	System architect

ARCH.912 Finalise the system context and constraints

- [ARCH.048 Identify system implementation constraints](#)
- [ARCH.045 Define complete set of system-level actors](#)
- [ARCH.119 Define consolidated set of system capabilities/missions](#)
- [ARCH.041 Transfer or trace upper level model elements to system level](#)
- [ARCH.044 Define requirements from non-actor stakeholders](#)
- [ARCH.051 Align the system boundary, capabilities and actors with collaboration projects](#)

ARCH.048 Identify system implementation constraints

[SM-2782](#) - Populate Confluence page for activity definition: ARCH.048 Identify system implementation constraints FINISHED

Goal	Produce a list of candidate non-functional requirements on the system, derived from implementation constraints
Requirements met by this process step	ISO 15288 6.4.2.3 b 2)-4), d 2)-3), e 2)-4) EN 50126-1 7.2.2 c)-e)
Inputs	<p>AMOD-010 Trade space assessment</p> <p>AMOD-012 System lifecycle model</p> <p>IM standards applying to any point of the engineering lifecycle</p> <p>Expert knowledge of railway engineering delivery</p>
Outputs	AMOD-049 Non-functional requirements implementation
Methodology	<p>The trade space assessment is a starting point in understanding tradeoff factors for the system under development. At this point in the process, a more detailed knowledge of the specific constraints on the system's implementation should be introduced.</p> <p>Constraints may be based on non-tradeable limitations (like the laws of physics, for example) or on tradeable limitations where a certain choice has been made and we don't propose to change it (for example, the need for earthing and bonding of trackside equipment is there because the line is electrified with OLE and whilst we could say "remove the OLE for our system" we choose not to).</p> <p>This requires understanding of the lifecycle and of the tradeoff factors.</p> <p>Implementation constraints will have been communicated to suppliers many times in previous specifications. They may also be found in laws, international standards, and IM standards.</p> <p>We must decide if the system is to comply with these existing requirements, or if the IM is to lobby for changes to these requirements. Either outcome is possible but we must take account of the time and effort needed to persuade government to change legislation.</p> <p>We will probably need to consult with our neighbouring organisational units who are more experienced with implementation of equipment on the track, in order to pick up the needs and practical considerations that we should include.</p> <p>At all points we should (internally) question everything we are told and satisfy ourselves that these constraints are either non-tradeable or firm decisions before accepting them.</p>

Tools and non-human resources	Brainstorming resources (optional) KRWD Perinorm
Cardinality	One-off with possibility to revise
Completion criteria	The set of identified constraints is complete enough with respect to the stage of maturity of the system (this means that it is not necessary to identify 100% of constraints the first time this activity is done; but there will be a point where it is necessary to gain assurance that we have covered enough constraints to make the set safe enough to fly. This will come at the point where there are no further opportunities to iterate before delivering the first system implementation into service.)
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Implementation experts including from neighbouring business units
Uses outputs (Informed)	None directly

ARCH.045 Define complete set of system-level actors

[SM-2781](#) - Populate Confluence page for activity definition: ARCH.045 Define complete set of system-level actors FINISHED

Goal	Consolidate allocation of system level functions to either system actors or the system.
Requirements met by this process step	None defined
Inputs	AMOD-121 Operational activity to system level transfer rationales AMOD-046 Selected system boundary sketch AMOD-045 System capabilities
Outputs	AMOD-119 System context definition [SAB] AMOD-045 System capabilities (updated)
Methodology	<p>The following activities transferred operational entities and operational activities to system level:</p> <ul style="list-style-type: none">• ARCH.043 Select and transfer operational actors to system level (incl. traceability)• ARCH.040 Transfer OA to system level function, allocate entirely to actor• ARCH.041 Transfer or trace OA to system capability• ARCH.050 Map missions to system capabilities• ARCH.093 Transfer selected operational activities to system level as system functions <p>Based on the these activities the AMOD-046 shall be used to generate AMOD-119 System context definition [SAB], which defines all system actors and the system functions being either allocated to the system or the system actors. This is intentionally a large, working viewpoint; it is not intended for publication, but rather as a place to see the whole picture all at once.</p> <p>Basically this activity consolidates and captures AMOD-046, hence when doing this activity AMOD-046 can be already complete maybe nothing needs to be changed, but this activity has the purpose of consolidating exactly that.</p>

Tools and non-human resources	Team for Capella
Cardinality	Once per system capability, revisit to ensure it is still valid Once per consolidated package of system capabilities
Completion criteria	All system level functions are allocated to the system or a system level actor. The output view conforms to all its modelling rules.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System engineer
Provides input to /assists (Contributes)	System architect
Uses outputs (Informed)	None directly.

ARCH.119 Define consolidated set of system capabilities/missions

[SM-5053](#) - Populate Confluence page for process step ARCH.119 **FINISHED**

Goal	Structure and consolidate all system capabilities
Requirements met by this process step	None identified.
Inputs	AMOD-045 System capabilities AMOD-120 Operational capability to system capability transfer rationales
Outputs	AMOD-045 System capabilities
Methodology	<p>Due to the previous process steps a large number of system capabilities is available. For consolidation of the system capabilities it is important to get an overview of ALL system capabilities.</p> <ul style="list-style-type: none">• Grouping: Organizing the system capabilities into groups will bring structure to the diagram and facilitate understanding. The groups may be of functional nature and/or depend on stakeholder allocation. Set relations (include, extend, generalize) between the system capabilities if required.• Duplicates: Where duplicate system capabilities exist, they should be eliminated. Ensure that a duplicate is replaced by the master item on all views where it appears, BEFORE deleting the duplicate. On deletion, the duplicate should no longer be related to any other model element.• If a deletion is not reasonable (rationales from AMOD-120 should be taken into account) an inclusion, extension or generalization relation between the capabilities should be considered. If the system capabilities just "sound" similar but have different meanings, the names should be changed.• Gaps: When there are obvious gaps identified the missing system capability should be added. Nevertheless, the root cause of the lack should be identified to make sure this capability is required and was not omitted on purpose.• Naming: The names of the system capabilities must be consistent. They should use similar language and terms.
Tools and non-human resources	Team for Capella
Cardinality	Once per package of capabilities to be delivered, with iterations allowed.
Completion criteria	The produced view is compliant with the modeling rules.
Design review	ARCH.R.4 System review (consolidated)
Step done by (Responsible)	Lead system architect
Provides input to /assists (Contributes)	System architect Cross-cutting engineer
Uses outputs (Informed)	None identified.

ARCH.041 Transfer or trace upper level model elements to system level

[SM-2777](#) - Populate Confluence page for activity definition: ARCH.041 Transfer or trace OA to system capability FINISHED

Goal	Transfer operational activities to system capabilities and establish traceability.
Requirements met by this process step	None defined
Inputs	<p>AMOD-101 Consolidated operational activities & interactions</p> <p>AMOD-121 Operational activity to system level transfer rationales</p> <p>System capabilities from collaborative projects (optional)</p>
Outputs	AMOD-045 System capabilities
Methodology	<p>Each operational activity that shall be transferred as system capability to system level may be transitioned automatically from operational level to system level in Capella.</p> <p>Ensure that each system capability has a traceability link to its original operational activity (it is called "Outgoing Generic Traces" in Capella).</p> <p>If system capabilities corresponding to the operational activity are already defined within a collaborative project (e. g. RCA), they may be referred to and utilised, and a traceability link from that system capability to the corresponding operational activity implemented.</p> <p>Where it is desired that a system capability could be realised in several partial steps (for example, beginning with one train on plain line, then adding points, other trains, signals etc) then several system capabilities may be created and related to the more general capability via Generalisation.</p> <p>This allows for partial capabilities to be developed separately and for a progression to be created leading to the full implementation of a generic capability.</p>
Tools and non-human resources	Team for Capella
Cardinality	One-off with revisions for managing changes to the set of operational activities.
Completion criteria	The transfer of all operational activities that shall be transferred as system capabilities to system level (in accordance with AMOD-121) is completed and traceability link from system capability to operational activity is implemented.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System engineer
Provides input to /assists (Contributes)	System engineer

**Uses outputs
(Informed)**

None directly

ARCH.044 Define requirements from non-actor stakeholders

[SM-2780](#) - Populate Confluence page for activity definition: ARCH.044 Define requirements from non-actor stakeholders FINISHED

Goal	Ensure all the requirements are captured early
Requirements met by this process step	ISO 15288 6.4.2.3 b 2)-4), d 2)-3), e 2)-4) EN 50126-1 7.2.2 c)-e)
Inputs	Stakeholder engagement
Outputs	AMOD-050 Stakeholder non-functional requirements
Methodology	<p>Non-actor stakeholder candidates are identified in AMOD-022 Enterprise & environment definition. Operational entities the "wider environment" and possibly also in the "environment" boundaries could be non-actor stakeholders with need that could influence the development of the system.</p> <p>Two main stages to this step:</p> <ul style="list-style-type: none">gather the raw needs from the stakeholders, by brainstorming workshop or by reading material released by particu stakeholders;resolve conflicts between the needs of different stakeholders, and convert the raw information into a set of standardised high-level requirements.
Tools and non-human resources	Team for Capella
Cardinality	Theoretically one-off with revisions allowed; practically, may need to be done in several batches, divided by stakeholde stakeholder group.
Completion criteria	<p>The set of non-actor stakeholders was identified.</p> <p>The set of non-actor stakeholders that was consulted was sufficient to give confidence that risk of undiscovered requirements is as low as practicable.</p> <p>The capture of requirements from each stakeholder set is sufficiently thorough.</p> <p>Conflicts in the requirements arising from different stakeholders have been resolved.</p>
Design review	
Step done by (Responsible)	new role Stakeholder manager/Business development manager System architect
Provides input to /assists (Contributes)	System architect Workshop facilitator
Uses outputs (Informed)	None directly

ARCH.051 Align the system boundary, capabilities and actors with collaboration projects

Draft

[RCAMT-480](#) - Populate Confluence page for activity definition: ARCH.051 Align the system boundary, capabilities and actors with collaboration projects **FINISHED**

Goal	Align defined system context and capabilities with collaboration projects.
Requirements met by this process step	None
Inputs	AMOD-045 System capabilities AMOD-119 System context definition AMOD-120 Record of system capabilities and missions AMOD-121 Record of system and actor functions AMOD-122 Record of system actors AMOD-049 Non-functional requirements implementation AMOD-050 Stakeholder non-functional requirements
Outputs	AMOD-045 System capabilities (Updated) AMOD-119 System context definition (Updated) AMOD-120 Record of system capabilities and missions (Updated) AMOD-121 Record of system and actor functions (Updated) AMOD-122 Record of system actors (Updated) AMOD-049 Non-functional requirements implementation (Updated) AMOD-050 Stakeholder non-functional requirements (Updated)
Methodology	Analyse and compare the provided content from the collaboration project with the own project's content. Outcome of the analysis may result in: <ul style="list-style-type: none">• Going back to the collaborative project and discussing potential need of modification of the system context or model elements (e.g. system capabilities) coming from that project.• Updating the defined AMOD-045 System capabilities and AMOD-119 System context definition to adapt to content from collaborative project.
Tools and non-human resources	Export of model content in Confluence or PDF. Teams for Capella
Cardinality	Once per collaboration project, with permitted revisions
Completion criteria	System context and capabilities are aligned in the model. The output view conforms to its modelling rules.

Design review	Not part of
Step done by (Responsible)	Lead system Architect
Provides input to /assists (Contributes)	System Architect System Engineer
Uses outputs (Informed)	None.

ARCH.913 Analyse the system capabilities to determine detailed system needs

- [ARCH.179 Complete the definition of the system capability of interest](#)
- [ARCH.088 Define system functions and functional exchanges](#)
- [ARCH.052 Create initial system exchange scenarios](#)
- [ARCH.053 Create initial system functional chains](#)
- [ARCH.054 Model data flowing between system functions](#)
- [ARCH.055 Model states on system level](#)
- [ARCH.056 Map system functionality to states](#)
- [ARCH.158 Model external interface layers](#)
- [ARCH.057 Model non-payload data on external interfaces](#)
- [ARCH.058 Define measures of performance](#)

ARCH.179 Complete the definition of the system capability of interest

Goal	<p>Visualise the end-to-end relationships between System Capabilities from the perspective of one System Capability, the System Capability of Interest (SCOI)</p> <p>Elaborate generic pre and post-conditions</p>
Requirements met by this process step	None identified.
Inputs	AMOD-045 System capabilities
Outputs	AMOD-138 Single system capability context
Methodology	<p>Once a system capability has been defined according to AMOD-045 System capabilities, a Contextual Capability (CC) diagram can be created for it. This diagram is automatically populated with the system capability of interest as well as all currently existing relationships of the system capability of interest.</p> <p>If there is a relationship with another system capability missing from the diagram, it can be manually added. This is done by introducing the relevant system capability into the diagram and creating either a capability generalisation, extends or includes relationship between it and the system capability of interest, according to the modelling rules.</p> <p>Dependencies between the system capability of interest and other system capabilities allow for the elaboration of generic pre and post-conditions for the system capability of interest. This elaboration can be carried out as further information related to the system capability of interest is gained.</p> <p>At this point, there is no need to align with other system capability owners yet.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per System Capability
Completion criteria	The output view conforms to its modelling rules.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System Architect
Provides input to /assists (Contributes)	Systems Engineer
Uses outputs (Informed)	None

ARCH.088 Define system functions and functional exchanges

[RCAMT-390](#) - Add mandatory rationales to ARCH.088, 182 and other steps that require functions to justify their inputs. FINISHED

Goal	Identify and capture functions and exchanges that are needed for a particular capability
Requirements met by this process step	See ARCH.913 Analyse the system capabilities to determine detailed system needs
Inputs	<p>Individual needs identified during ARCH.052 Create initial system exchange scenarios and ARCH.053 Create initial system functional chains.</p> <p>(this activity should run in parallel with those two activities)</p> <p>System functional chains from collaborative projects (optional)</p> <p>System exchange scenarios from collaborative projects (optional)</p>
Outputs	AMOD-056 System functions and exchanges (single system capability)
Methodology	<p>As a exchange scenario or functional chain is built, functions are arranged in a sequence.</p> <p>Functions/functional exchanges can be reused if appropriate, but if no existing function/exchange meets the need at that point in the scenario/process, then a new function needs to be created. That is what this activity is for.</p> <p>As a part of this step, the rules and the guidelines for the definition of system functions should be followed defined within 2. Modelling Rules for System Analysis#ModRules_VE_SysFunction. Additional information can be found here Pattern for defining system functions and Pattern for defining Configuration and Calibration data. Each system function associated with operational states of the system under consideration (“core function”) will then be assigned to one of five categories representing the main functions of a control system (control, actuate, indicate, sense, observe) they are supported by configuration and calibration data. The types of resulting functional exchanges between these functions shall also be limited to those defined in the 2. Modelling Rules for System Analysis#ModRules_VE_SysFuncExchange.</p> <p>Document the justification for a particular design decision with a rationale.</p> <p>If functions already created previously before the ARCH process has been completely introduced, this functions should be always be reused/renamed, in preference to creating new ones, if it is appropriate.</p> <p>If relevant system functions and functional exchanges are already defined at an international level, these should be referred to and reused if appropriate.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per system capability.
Completion criteria	<p>The output diagram conforms to its modelling rules.</p> <p>The set of functions and functional exchanges identified is safe enough to try.</p>
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System architect

Provides input to /assists (Contributes)	Inclusive OR: <ul style="list-style-type: none">• Subsystem architect• Expert for GoA4 railway operation• Cross-cutting engineer• Engineer
Uses outputs (Informed)	RAMS manager RAMS engineer Verification & validation manager

ARCH.052 Create initial system exchange scenarios

Goal	Visualise the sequence of functional exchanges between the system and the actors needed to achieve the outcome of a system capability
Requirements met by this process step	None defined (see ARCH.913 Analyse the system capabilities to determine detailed system needs)
Inputs	<p>AMOD-045 System capabilities</p> <p>Knowledge of workshop participants (optional)</p> <p>System functions from collaborative projects (optional)</p>
Outputs	AMOD-058 Initial system exchange scenario
Methodology	<p>If the person creating the output view knows enough about how the system works with the actors for this capability, they can create the sequence diagram directly from their own knowledge and must only develop a view that complies with all the modelling rules.</p> <p>If, on the other hand, it is necessary to gather information from others, a brainstorming workshop should be held.</p> <p>If relevant scenarios are already defined at an international level , these should be referred to and reused if appropriate.</p> <p>Participants should be sufficient to represent knowledge/concerns of all the involved actors and functional areas of the system.</p> <p>The brainstorming goal is to identify sufficient external functional exchanges to get from the starting conditions to the ending conditions.</p> <p>The number of internal exchanges (that is, functional exchanges between pairs of functions that are both allocated to System of Interest) should be kept to a minimum.</p> <p>Where new functions or functional exchanges are identified, these should be first created on the corresponding AMOD-056 System functions and exchanges (single system capability).</p> <p>Where behaviour is expected of an actor in order to reach the end conditions of a system capability, actor functions should be modelled, but these can be captured at a more abstract/placeholder level than the system functions.</p>
Tools and non-human resources	<p>Team for Capella</p> <p>(optional) General brainstorming equipment</p>
Cardinality	Once per system capability
Completion criteria	<p>The output view complies with its modelling rules.</p> <p>The content of the output view represents the team's current understanding of the way this system capability is intended to be delivered.</p>
Design review	
Step done by (Responsible)	System architect

Provides input to/assists (Contributes)	Inclusive OR: <ul style="list-style-type: none">• Subsystem architect• Expert for GoA4 railway operation• Cross-cutting engineer• Engineer
Uses outputs (Informed)	RAMS manager RAMS engineer Verification & validation manager

ARCH.053 Create initial system functional chains

Goal	Define the sequence and decision logic of system functionality needed to achieve a system capability in a format that is usable for failure mode analysis.
Requirements met by this process step	ISO 15288 6.4.3.3 b 1) EN 50126-1 7.3.2 a), d), e), 7.5.2
Inputs	AMOD-045 System capabilities Knowledge of workshop participants (optional) Functional chains from collaborative projects (optional)
Outputs	AMOD-059 Initial system functional chain description
Methodology	<p>If the person creating the output view knows enough about how the system works with the actors for this capability, they can create the functional chain description diagram directly from their own knowledge and must only develop a view that complies with all the modelling rules.</p> <p>If, on the other hand, it is necessary to gather information from others, a brainstorming workshop should be held.</p> <p>If relevant functional chains are already defined at an international level (e.g. RCA), these should be referred to and reused if appropriate.</p> <p>Participants should be sufficient to represent knowledge/concerns of all the involved actors and functional areas of the system.</p> <p>The brainstorming session should identify sufficient functions and functional exchanges to get from the starting conditions to the ending conditions, and identify the decision and sequence logic that is needed (parallelism, optional branches, iterations).</p> <p>The number of internal exchanges (that is, functional exchanges between pairs of functions that are both allocated to System of interest) should be kept to a minimum.</p> <p>Where new functions or functional exchanges are identified, these should be first created on the corresponding AMOD-056 System functions and exchanges (single system capability).</p> <p>Where behaviour is expected of an actor in order to reach the end conditions of a system capability, actor functions should be modelled, but these can be captured at a more abstract/placeholder level than the system functions.</p>
Tools and non-human resources	Team for Capella Optional: basic brainstorming tools
Cardinality	Once per system capability
Completion criteria	The output view complies with its modelling rules. The content of the output view represents the team's current understanding of the way this system capability is intended to be delivered.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System architect

Provides input to /assists (Contributes)	Inclusive OR: <ul style="list-style-type: none">• Subsystem architect• Expert for GoA4 railway operation• Cross-cutting engineer• Engineer
Uses outputs (Informed)	RAMS manager RAMS engineer Verification & validation manager

ARCH.054 Model data flowing between system functions

[RCAMT-369](#) - Refine data model rules so that data can be reused on lower levels and not need to be automatic transferred **FINISHED**

Goal	Capture the requirements for information/data flow between system functions
Requirements met by this process step	ISO 15288 6.4.2.3 b 1), c 2) EN 50126-1 7.2.2 a), 7.3.2 a)
Inputs	There may be existing views to which the data for the current system capability can be added, but if not, then create new views. AMOD-056 System functions and exchanges (single system capability) AMOD-025 Abstract concepts AMOD-110 Operational exchange items [O.CDB] AMOD-105 Operational data objects [O.CDB]
Outputs	AMOD-025 Abstract concepts (updated) AMOD-112 System data objects [S.CDB] (updated) AMOD-113 System exchange items [S.CDB] (updated)

Methodology

During the modelling of system scenarios ([ARCH.052](#) and [ARCH.053](#)), new functional exchanges will be created. The data or information that flows along these exchanges must be defined.

If relevant data are already defined at an international level (e.g. RCA), these should be referred to and reused if appropriate.

Exchange items

Check all instances of [AMOD-110 Operational exchange items \[O.CDB\]](#) and [AMOD-113 System exchange items \[S.CDB\]](#) for existing reusable exchange items, before creating new ones. It is not allowed to have identical data model elements on multiple levels. If there is a need to refine or change an exchange item from system level, please create a new exchange item at the logical level and link it via a realisation to the corresponding data object at the higher level.

Note: New data objects and types should be modelled initially on [AMOD-113](#) so that other modellers can look for them and reuse them where appropriate.

Each functional exchange should have allocated at least one exchange item.

The data information between two system functions is permanently present at the logical level, therefore FLOW should be used as communication mechanism for exchange items in all cases.

In some exceptional cases, it is allowed to use an EVENT, but only for functional exchanges between an actor function and a system function if the interface is constrained by an external standard.

Define for each exchange item the types of data, via the Exchange Item Elements. The information content of the exchange item should be defined as far as possible using an appropriate combination of **data objects** (simple types or structured types).

Where the information content is unknown or unclear, write a text description on the exchange item, to act as a placeholder/STUB until the information content is resolved (communication mechanism of exchange item = UNSET).

Data objects

Check all instances of [AMOD-105 Operational data objects \[O.CDB\]](#) and [AMOD-112 System data objects \[S.CDB\]](#) for existing reusable data objects and types, before creating new ones. It is not allowed to have identical data model elements on multiple levels. If there is a need to refine or change a data object from operational level, please create a new data object at the logical level and link it via a realisation to the corresponding data object at the higher level.

When a new data object is created on system level the rationale for their creation must be documented.

Note: New data objects and types should be modelled initially on [AMOD-112](#) so that other modellers can look for them and reuse them where appropriate.

Instances of [AMOD-112](#) should be structured in line with the structure used for the abstract concepts (so that there is generally one instance of [AMOD-112](#) per instance of [AMOD-025](#) - unless it is necessary to split the views up for readability).

Abstract concepts

Abstract concepts are defined on operational level in [AMOD-025 Abstract concepts](#). These concepts may be updated, and/or new concepts might be identified during definition of the logical architecture. In such a case, the person responsible for the abstract concept (e.g. operational concept architect) must be involved in the change process.

If a new abstract concept is to be defined, follow the process step [ARCH.008 Define abstract concepts](#).

Note: due to the feature based approach it is theoretically possible that data objects might be created on a lower layer first, e.g. when feature 1 work group reached logical layer. Later, when feature 2 work group starts to work on system layer, the need for the same data object might be identified. If so the data object should be moved from logical level to system level.

Tools and non-human resources	Team for Capella
Cardinality	Once per system capability
Completion criteria	The output views conform to their modelling rules. The exchange item definitions represent the best current knowledge of the information content required for exchange at system level.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.055 Model states on system level

[SM-2797](#) - Populate Confluence page for activity definition: ARCH.055 Model system states **FINISHED**

Goal	<p>Capture understanding of how the system's behaviour changes based on states;</p> <p>Refine understanding of state-based differences in behaviour of system actors</p>
Requirements met by this process step	None defined
Inputs	<p>AMOD-119 System context definition</p> <p>AMOD-023 Operational entity/actor states (as reference)</p> <p>In particular if there exists an instance of AMOD-023 for the wider system of interest, this is a reference for the creation of states on the system's state model.</p> <p>Actor states from collaborative projects (optional)</p>
Outputs	<p>AMOD-060 System/actor states</p>
Methodology	<p>During analysis of a system capability, new states may be identified for the system, for actors, or for abstract concepts.</p> <p>New states diagrams should be created when statefulness is identified on an actor, abstract concept or the system for the first time.</p> <p>In case state diagrams corresponding to system capability scenarios from collaborative projects exist, they should be transferred, in which case this process step will not be applicable.</p> <p>Thereafter, the existing instance of AMOD-060 should be updated unless it is necessary to create a separate view for size reasons. Independent dimensions of statefulness should be modelled as regions on the same state machine, rather than as a separate state machine.</p> <p>This activity should be done in parallel with the initial analysis of a system capability.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per system capability
Completion criteria	<p>The output view conforms to its modelling rules.</p> <p>All known/identified states associated with the system, or an actor, for the current capability, have been captured in the output view.</p>
Design review	<p>ARCH.R.3 System capability review</p> <p>ARCH.R.4 System review - consolidated</p>
Step done by (Responsible)	System architect

Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.056 Map system functionality to states

[SM-2798](#) - Populate Confluence page for activity definition: ARCH.056 Map system functionality to states/modes FINISHED

Goal	Gain understanding of state-based changes to the system's behaviour
Requirements met by this process step	None defined
Inputs	AMOD-060 System/actor states System/actor states from collaborative projects (optional)
Outputs	AMOD-055 System functions to state allocation map [Matrix] Output is a set of new relationships in the model according to the architectural tuple below: system function is available in system state
Methodology	<p>System functions may not be available all the time. Their availability may be dependent on a system state or mode.</p> <p>If relevant states are already defined within a collaborative project, these should be referred to and utilised for this activity, where appropriate.</p> <p>This activity should be carried out in parallel to the modelling of system functionality for a single system capability. The scope of the activity is all system functions involved in the capability.</p> <p>For each state dimension (that is, each individual state machine or state machine region on the system):</p> <ul style="list-style-type: none"> If a system function is only available in certain system states <ul style="list-style-type: none"> Allocate the function to those states where it is available; Else if a system function is available independently of this state dimension: <ul style="list-style-type: none"> Allocate the function to all states in this dimension <p>end for</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per system capability
Completion criteria	All system functions involved in this capability are allocated to the correct states
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Cross-cutting engineer
Uses outputs (Informed)	None directly

ARCH.158 Model external interface layers

[SM-3429](#) - Create Confluence page for activity definition ARCH.158 FINISHED

Goal	Identify the constraints on system design arising from external interfaces
Requirements met by this process step	EN 50126-1 7.3.2.1 b)
Inputs	AMOD-056 System functions and exchanges (single system capability) AMOD-058 Initial system exchange scenario Interface standards/specifications from elsewhere
Outputs	AMOD-114 System interface definition AMOD-115 External interface behaviour [S.ES]

<p>Methodology</p>	<p>Since an external interface can impose constraints on how the system should be designed, every external interface should be defined to the fullest extent possible, even at the system level of the model. This is in contrast to the definition of internal interactions, which are kept at the application layer until the physical level of the model.</p> <p>The responsible person should familiarise themselves with the Open Systems Interconnect layer model of interfaces. This is used throughout our processes as a basis for identifying layers of interfaces concerned with different tasks. The reference model for OSI is available here.</p> <p>For the purpose of DBS/RCA, the following interpretations of OSI are made:</p> <ul style="list-style-type: none"> • application layer - information content that needs to be transferred in order to achieve the system capabilities • presentation layer - data formatting and type conversion, packet structuring, bit field formats • session layer - message sequencing & handshaking • transport layer - connection setup, monitoring and shutdown • network layer - addressing and routing • data link layer - formats for physical transmission, byte/bit order, parities, checksums... • physical layer - physical configurations, voltages/energy levels, polarities, frequencies, pinouts etc. <p>These layers are a guideline and can be adapted and tailored as necessary for a specific interface. Sometimes not all layers are needed (for example, a discrete digital relay interface could be adequately represented by application and physical layers only), where other interfaces define extra layers not explicitly included in OSI, most commonly a "safety" layer for error-detection and correction to prevent incorrect data being used in safety critical applications.</p> <p>On the system level only those layers that are externally constrained and/or relevant to the interface shall be modelled.</p> <p>How to do this task</p> <ul style="list-style-type: none"> • Decide on an appropriate set of interface layers for the interface you are modelling; • The application layer of the interface should already be represented by the functions and functional exchanges that interact across the system boundary in the existing scenarios used as input to this activity; • Create exchange scenarios for any interface-specific behaviours that need to be modelled (for example any sequences associated with link setup/closure, link management, error detection/correction; segregate these scenarios so they only model behaviour on one layer at a time; • Create a SAB view to represent only the interfacing actor and the system <ul style="list-style-type: none"> • Create component exchanges between the system and the actor for all layers of the interface between application and data link inclusive; • Create a physical link to represent the physical layer of the interface. <p>Where it is not yet known how an interface layer will be realised, the component exchange or physical link should still be created. This indicates that a gap is known and can be managed.</p> <p>Where an external standard/specification dictates the nature of the interface, the interface to the system of interest shall exactly reflect the implementation of that standard.</p>
<p>Tools and non-human resources</p>	<p>Team for Capella</p>
<p>Cardinality</p>	<p>Once per external interface</p>
<p>Completion criteria</p>	<p>The output views conform to their modelling rules</p> <p>The interface model content is a correct match for the known constraints applying to that interface</p>
<p>Design review</p>	<p>ARCH.R.3 System capability review</p>

Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.057 Model non-payload data on external interfaces

Goal	Complete the characterisation of external interfaces by modelling data for layers below application
Requirements met by this process step	ISO 15288 6.4.4.3 c 1) CSM-RA guidance 1.2
Inputs	AMOD-114 System interface definition AMOD-115 External interface behaviour [S.ES] Non-payload data defined in collaborative projects (optional)
Outputs	AMOD-112 System data objects [S.CDB] AMOD-113 System exchange items [S.CDB]
Methodology	Based on the interface layers identified in ARCH.158 Model external interface layers , capture the necessary exchange item models and new data objects in their own independent instances of AMOD-112 and AMOD-113 as appropriate. This includes packet structure, header definitions etc. for each layer of an interface. If relevant data are already defined at an international level, these should be referred to and reused if appropriate. For more guidance see methodology on ARCH.054 Model data flowing between system functions .
Tools and non-human resources	Team for Capella
Cardinality	Once per external interface
Completion criteria	The output views conform to their modelling rules. All known layers of the external interface have been captured.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.058 Define measures of performance

[SM-2800](#) - Populate Confluence page for activity definition: ARCH.058 Define measures of performance **FINISHED**

Goal	Gain understanding of the system characteristics that are considered necessary to achieve the measures of effectiveness
Requirements met by this process step	ISO 15288 6.4.3.2 c) / 6.4.3.3 c) 2)
Inputs	AMOD-106 Enterprise goals (modified after ARCH.009 Define measures of effectiveness) Any existing version of AMOD-057 Measures of performance
Outputs	AMOD-057 Measures of performance (updated, if it already existed)

Methodology	<p>For the capability of interest:</p> <p>Consider the context of the system capability and what operational capabilities are supported by it;</p> <p>Given the system functionality, data and statefulness being modelled in parallel, identify what should be measured in order to support a claim that the measures of effectiveness have been achieved;</p> <p>Re-use existing measures of performance where appropriate, rather than creating new ones; target values can be made more stringent if necessary;</p> <p>If existing measures of performance are suitable, but their target values are more stringent than is needed for the capability of interest</p> <p style="padding-left: 40px;">Leave the target value as it stands (since, if reached, the needs of the capability of interest are met)</p> <p style="padding-left: 40px;">Note that this measure of performance is also needed for the capability of interest (ensuring that it will not be removed later or reduced in stringency without examining its impact on the capability of interest)</p> <p>Measures of performance are distinct from the measures of effectiveness already created, because they are focused on the system and not on the wider stakeholder environment. In other words, they are formulated from different viewpoints: MoEs from the stakeholders' viewpoint, MoPs from the viewpoint of the "supplier" of the system of interest.</p> <p>The intent of defining MoPs is to answer the question "how well must the system perform in order that the system's stakeholders can achieve their mission?".</p> <p>Each MoP contributes to the evaluation of one or more MoEs.</p> <p>Measures of performance should be stated in technical, measurable terms but they are distinct from non-functional parameters in that they do not necessarily need to be specified directly on the outputs of functions or the quality of individual types of data; they should be a little higher-level. The individual non-functional parameters will be defined in the risk assessment stage ARCH.915 Assess & mitigate the risks of system failure and will, in turn, contribute to the achievement of the MoPs.</p> <p>They can be used as a basis for verification criteria.</p> <p>MoPs may emerge whenever a new capability is assessed, but the set of MoPs should be engineered alongside the set of capabilities, with strong consultation with other system architects and the lead system architect so that a coherent approach is taken.</p> <p>Example</p> <p>Measure of effectiveness: the capacity of the network is x trains per hour in each direction.</p> <p>Possible measures of system performance:</p> <ul style="list-style-type: none"> ■ the number of trains that the system can support simultaneously is greater than or equal to y ■ the processing time for the system to provide a safe authority for a train unit to follow a preceding train unit is no greater than z
Tools and non-human resources	<p>Team for Capella</p>
Cardinality	<p>Once per system capability</p>
Completion criteria	<p>Sufficient MoPs have been defined for the capability of interest.</p>

Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Lead system architect Operational concept architect
Uses outputs (Informed)	Not defined

ARCH.915 Assess & mitigate the risks of system failure

- [ARCH.065 Derive safety target to system functions and define additional risk control measure needed](#)
- [ARCH.070 Calculate system deviation probability](#)

ARCH.065 Derive safety target to system functions and define additional risk control measure needed

- [ARCH.066 Identify system level deviations](#)
- [ARCH.067 Populate the fault tree](#)
- [ARCH.068 Assess deviation probability \(external constraints\)](#)
- [ARCH.069 Add system level risk measures](#)

ARCH.066 Identify system level deviations

[SM-2808](#) - Populate Confluence page for activity definition: ARCH.066 Identify system level deviations

READY FOR REVIEW

Goal	Manage the risks (safety, security, business) that are caused by the system or the actors deviating from designed behaviour
Requirements met by this process step	EN 50126 7.4.2.1 1) and 2) ISO 15288 6.4.3.3 b) 3
Inputs	AMOD-059 Initial system functional chain description AMOD-115 External interface behaviour [S.ES] For reference: AMOD-058 Initial system exchange scenario AMOD-056 System functions and exchanges (single system capability) AMOD-030 Accident and hazard state model AMOD-130 Business loss and risk state model AMOD-131 Security loss and threat state model
Outputs	AMOD-071 System failure modes and effects analysis AMOD-004 Fault tree (per operational deviation)
Methodology	<p>The primary focus of this task is to identify what deviations in functional exchanges can occur during the execution of the functional chain for a specific system capability.</p> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"><p> The activity should be done jointly between the RAMS architect and Security Architect and the system architect. The system architect is responsible for organising the work but should not attempt to do this alone.</p></div>



Exclusion

Some system-level functional exchanges are placeholders for real-world physics and effects (generally these are the exchanges emerging from the "plant" placeholder functions as defined by the [pattern for system function definition](#)). These should not be considered for deviation analysis at this level, because they should already have been identified and assessed at operational level in most cases. It may be necessary to return to operational level if a gap is discovered.

Deviation analysis at system level focuses on functional exchanges between the other functions in the control loop pattern (or other designed-in system functions, where the pattern is not used). These are all considered to be "control actions" within the meaning defined by texts on system-theoretic process analysis (STPA).

That encompasses:

- reference values incoming from actors, whether human or machine
- human-perceptible states
- required states
- estimated states
- sensed states
- (possibly in a few cases) actual external input states

but generally not external output states.

For every functional exchange in the system functional chain (but see exclusion):

Determine which of the following deviations could occur on the functional exchanges:

1. Not providing the functional exchange could lead to a risk.
=> Guideword: "**1-Not providing**" # state/info change intended to be used
2. Providing the functional exchange could lead to a risk.
 - a. Consider contexts in which the functional exchange may never be safe, implying that the exchange should never be made unintentionally:
=> Guideword "**2.1-unintended**" # state/info change in consideration but not intended to be used
 - b. Consider contexts in which the functional exchange has an incorrect parameter (e.g. setting an incorrect emergency frequency on a radio)
=> Guideword "**2.2-incorrect**" # parameter/function performance
 - c. Consider contexts in which an insufficient or excessive value of a parameter may be unsafe (e.g. providing insufficient or excessive braking commands)
=> Guideword "**2.3a-insufficient**", # parameter
=> Guideword "**2.3b-excessive**" # parameter
 - d. Consider contexts in which the direction of the information on the exchange may be unsafe (e.g. providing turn left instead of turn right commands)
=> Guideword "**2.4-wrong direction**" # info
 - e. Consider contexts in which the functional exchange has already been provided (e.g. repetitive or oscillatory control actions)
=> Guideword "**2.5-already**" # state/info already used
3. Providing a potentially safe functional exchange but too early, too late, or in the wrong order
=> Guideword "**3a-too early**", # state/info intended to be used
=> Guideword "**3b-too late**", # state/info intended to be used
=> Guideword "**3c-wrong order**" # state/info sequence
4. The control action lasts too long or is stopped too soon (for continuous control actions, not discrete ones)
=> Guideword "**4a-too long**", # state/info continuous command
=> Guideword "**4b-too short**" # state/info continuous command

The rationale for deciding a deviation will **not** occur as a string property value on the interaction with name "Rationale for discounting deviations" shall be recorded

For each deviation identified

Identify any deviations at operational level that could be caused by, or contributed to by, this deviation
(Note that it is possible for system deviations to be combined logically in order to cause a deviation, e.g.

(system deviation A OR system deviation B) implies operational deviation X;

(system deviation A AND system deviation B) implies operational deviation X.

Model the appropriate causal link and combinatorial logic in the fault tree.

OR combination (OR-gate): If one or more of the system level deviations are sufficient for the occurrence of the deviation at operational level.

Exclusive OR combination (XOR-gate): If only one and not more of the system level deviations are sufficient for the occurrence of the deviation at operational level.

AND combination (AND-gate): If all system level deviations that are combined by this AND-gate have to occur for the occurrence of the deviation at operational level.

Note

It is possible that, through analysing system failure modes, new routes are identified towards accidents that were not recognised at operational level. In this case, the operational risk assessment needs to be revisited in order to set up the top-level accident-hazardous state-safe state model, set an acceptable deviation probability and assign responsibility. Where an appropriate operational activity does not already exist, a very generic operational activity should be added to capture the need for protection against a certain risk. The rationale for this operational activity should clearly state that the need for it arose from system-level analysis and not directly from an operational need.

Do not attempt to model protections against accidents directly at system level; this will undermine the layering structure of the safety case.

This advice is equally valid for the equivalents for security and business risk.

Tools and non-human resources	Team for Capella [Link] for modelling fault trees - ticketed SET-183 - Jira issue doesn't exist or you don't have permission to view it.)
Cardinality	Once per system capability
Completion criteria	The functional chain has been completely checked for possible deviations that lead to operational deviations; The fault tree has been updated with causal links between system deviations and operational deviations
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System architect

Provides input to /assists (Contributes)	RAMS architect Security architect
Uses outputs (Informed)	Independent safety assessor

ARCH.067 Populate the fault tree

[SM-2809](#) - Populate Confluence page for activity definition: ARCH.067 Populate the fault tree

IN PEER REVIEW

Goal	Visualize all logical combinations of system level deviations that cause one operational deviation in a fault tree and define initial probabilities for the system level deviations.
Requirements met by this process step	ISO 15288 6.4.3.3 c)
Inputs	None identified
Outputs	AMOD-004 Fault tree (per operational deviation)
Methodology	<p>Following prerequisite activities have been completed in ARCH.066:</p> <ul style="list-style-type: none">• Deviations of functional exchanges on system level (system level deviations) have been identified• Causal links between a system level deviation that cause an operational deviations have been defined• Combinatorial logic of all system level deviations that cause one operational deviations has been defined <p>In this activity the fault tree shall be generated as per AMOD-004, such that the logical combination of system level deviations that results into an operational deviation are shown.</p> <p>Further, the probabilities for the occurrence of an operational deviation shall be broken down to the causing system level deviations. At this moment of the process a meaningful breakdown of probabilities will be very limited, because the physical manifestation of the system of interest is needed to make this more meaningful, which is not existent yet. Because of that as first approach only the equal distribution of probabilities is suggested depending on the logical combinations applied:</p>

- OR-gate: Distribute the probability of the operational deviation to the system level deviations such that the sum of the probabilities of the system level deviations does not exceed the probability of the operational deviation.
- XOR-gate: Distribute the probability of the operational deviation to the system level deviations such that each individual probability of a system level deviation does not exceed the probability of the operational deviation.
- AND-gate: Distribute the probability of the operational deviation to the system level deviations such that the product of the probabilities of the system level deviations does not exceed the probability of the operational deviation.



Note

The distribution of the probabilities among the system level deviation will become important in one of the following process activities, when external constraints that are implied on the system are considered. E.g. when an input is used from an actor/system of low safety integrity in a system of that requires high safety integrity. Then safety measures have to be incorporated to compensate for the input of low safety integrity.

Tools and non-human resources	Team for Capella (tbd - possibly a further tool or ' ' of modelling fault trees - ticketed SET-183)
Cardinality	Once per system capability per operational deviation, with allowed revisions after changes in the operational level risk model or the system function definitions
Completion criteria	All system level deviations are included included in at least one fault tree for an operational deviation.
Design review	ARCH.R.3 System capability review ARCH.R.4 System review - consolidated
Step done by (Responsible)	System architect

Provides input to /assists (Contributes)	RAMS architect Security architect
Uses outputs (Informed)	Independent safety assessor

ARCH.068 Assess deviation probability (external constraints)

[SM-2810](#) - Populate Confluence page for activity definition: ARCH.068 Assess the deviation probability IN PEER REVIEW

Goal	Define a probability value for deviations of functional exchanges added on system level
Requirements met by this process step	EN 50126-1 6.4.3.1, 6.4.3.2 ISO 15288 6.4.3.3 c)
Inputs	AMOD-004 Fault tree (per operational deviation)
Outputs	AMOD-004 Fault tree (per operational deviation) (updated)
Methodology	<p>Following prerequisite activities have been completed in ARCH.067:</p> <ul style="list-style-type: none"> Distribution of operational deviation probability to system level deviations <p>In this activity all system level deviations of functional exchanges that are coming from actors or systems external to the system of interest shall be assessed in terms of their safety integrity. This means to set the probability of the deviation in accordance to that safety integrity.</p> <p>Example: Deviations on incoming functional exchanges, where the provider of the functional exchange might be a human being such that the probability of a deviation type "incorrect" might be 1E-04/h. In case the information conveyed by this functional exchange is used in SIL4 functions additional safety measures are needed to compensate for the low integrity input.</p> <p>The fault tree shall be updated to reflect the assessment of the aforementioned system level deviations. In the fault tree it shall be checked, if the related operational deviations are still not exceeded when taking into account the assessed system level deviations.</p> <p>If the result is that the operational deviation probability is not exceeded then this process can be followed by ARCH.070.</p> <p>If the result is that the operational deviation probability is exceeded then this process shall be followed by ARCH.069.</p> <div style="border: 1px solid gray; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>Functional exchanges that are related to railway processes that require high safety integrity (e.g. SIL4) and these functional exchanges are outgoing of the system of interest (i.e. consumed by actors or other systems) non functional requirements will be defined on these functional exchanges to define the limits of accountability of the system of interest.</p> </div>
Tools and non-human resources	Team for Capella (tbd - possibly a further tool or plugin for modelling fault trees - ticketed SET-183)
Cardinality	Once, with allowed revisions after changes in the operational level risk model or the system function definitions

Completion criteria	The deviations of all functional exchanges provided by actors or other systems have been assessed and are reflected in the fault tree.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	RAMS architect Security architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None identified

ARCH.069 Add system level risk measures

[SM-2811](#) - Populate Confluence page for activity definition: ARCH.069 Add system level risk measure

IN PEER REVIEW

Goal	Add system functionality to lower the probability of a related operational level deviation occurrence.
Requirements met by this process step	EN 50126-1 6.4.3.1, 6.4.3.2 ISO 15288 6.4.3.3 d)
Inputs	AMOD-004 Fault tree (per operational deviation) AMOD-059 Initial system functional chain description AMOD-115 External interface behaviour [S.ES] AMOD-056 System functions and exchanges (single system capability)
Outputs	AMOD-056 System functions and exchanges (single system capability) (updated) AMOD-004 Fault tree (per operational deviation) (updated)
Methodology	<p>Following prerequisite activities have been completed in ARCH.066 and ARCH.067:</p> <ul style="list-style-type: none"> ■ Fault tree has been populated to show system level deviation probabilities and resulting operational deviation probability ■ System level deviations of external incoming functional exchanges have been assessed and are reflected in the fault tree. <p>After the assessment of system level deviations of incoming functional exchanges have been reflected in the fault tree and the result is that the related operational deviation probability is exceeded this activity is carried out to add additional safety measure, i.e. to add system functions and functional exchanges to compensate where incoming functional exchanges are of safety integrity lower than needed:</p> <ul style="list-style-type: none"> • Define new system functions and functional exchanges in relation to the functional exchanges of insufficient safety integrity, e.g. error checking, verification, backup functionality as redundancy or arbiter functionality that allows validation of incoming functional exchanges. <p style="padding-left: 40px;">Example: Functionality for visual object recognition in a train unit is of insufficient safety integrity, functionality for visual object recognition in another train unit can be used to produce a validated object recognition result of higher safety integrity.</p> <ul style="list-style-type: none"> • Assess the possible deviations of the new functional exchange (Control action deviation category, probability) • Update the fault tree by adding the newly defined system level deviations and by defining its logical combination with other system level deviations that cause the same operational deviation.
Tools and non-human resources	Team for Capella (tbd - possibly a further tool or plugin for modelling fault trees - ticketed SET-183)
Cardinality	Once, with allowed revisions after changes in the operational level risk model or the system function definitions
Completion criteria	All needs for additional safety measures (as per ARCH.068) have been addressed, their deviations have been assessed and documented in the fault tree and these measures can be proven as effective via means like the fault tree.

Design review	ARCH.R.3 System capability review ARCH.R.4 System review - consolidated
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	RAMS architect Security architect
Uses outputs (Informed)	None identified.

ARCH.070 Calculate system deviation probability

[SM-2812](#) - Populate Confluence page for activity definition: ARCH.070 Calculate system deviation probability

READY FOR REVIEW

Goal	Calculate system deviation probabilities taking into account identified system level deviations and safety measures.
Requirements met by this process step	EN 50126-1 6.3.3.1, 6.3.3.2, 6.3.3.3 ISO 15288 6.4.3.3 e)
Inputs	AMOD-004 Fault tree (per operational deviation)
Outputs	AMOD-004 Fault tree (per operational deviation) (updated)
Methodology	<p>Following prerequisite activities have been completed in ARCH.067, ARCH.068 and ARCH.069:</p> <ul style="list-style-type: none"> Initial fault tree has been populated per operational capability System level deviations of external incoming functional exchanges have been assessed and are reflected in the fault tree Additional safety measures have been added to compensate for deviations of incoming functional exchanges that are of low safety integrity <p>In this activity the overall system level deviation probabilities shall be calculated, taking into account the aforementioned activities (initial fault tree, assessed deviations of external functional exchanges, additional safety measures).</p> <p>The result shall be compared to the probability of the caused operational deviation:</p> <ul style="list-style-type: none"> If the probability of the caused operational level deviation is not exceeded by the calculated system level probability this process activity is completed. If the probability of the cause operational level deviation is exceeded by the calculated system level probability this predecessor process activities shall be repeated, starting at ARCH.068.
Tools and non-human resources	Team for Capella (tbd - possibly a further tool or plugin for modelling fault trees - ticketed SET-183)
Cardinality	Once, with allowed revisions after changes in the operational level risk model or the system function definitions or after additional safety measure.
Completion criteria	Overall system level deviation probability does not exceed the caused operational level deviation probability.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	RAMS architect Security architect
Provides input to /assists (Contributes)	System architect

**Uses outputs
(Informed)**

None identified.

ARCH.916 Incorporate risk control measures in system scenarios /functional chains

- [ARCH.141 Update system functions with risk control measures](#)
- [ARCH.142 Create system exchange scenarios with risk control measures](#)
- [ARCH.143 Create system functional chains with risk control measures](#)
- [ARCH.149 Update system data with risk control measures](#)
- [ARCH.150 Update system interface model with risk control measures](#)
- [ARCH.163 Update system states](#)
- [ARCH.164 Update function mapping to system states](#)

ARCH.141 Update system functions with risk control measures

[SM-3061](#) - Populate Confluence page for activity definition: ARCH.141 **FINISHED**

[RCAMT-275](#) - Define modelling rules in ARCH for expressing maximum allowed wrong-side failure rates **CREATED**

Goal	Capture new functional requirements arising from risk mitigation
Requirements met by this process step	ISO 15288 6.4.3.3 b) 3)
Inputs	(to be confirmed) AMOD-032 Allocated risk control measures AMOD-056 System functions and exchanges (single system capability)
Outputs	AMOD-056 System functions and exchanges (single system capability) (updated)
Methodology	<p>For every entry in the allocated risk control measures,</p> <p>where the risk mitigation specifies a new function,</p> <p>create a new function, optionally based on the control loop principles as defined in the 2. Modelling Rules for System Analysis;</p> <p>create appropriate functional exchanges between the new function and existing functions, in accordance with the control loop principle as defined in the modelling rules for functional exchanges, if appropriate;</p> <p>where the risk mitigation specifies a non-functional attribute against a function on the current diagram</p> <p>capture the attribute(s) according to methods to be defined in</p> <div data-bbox="427 1626 845 1742"><p>SM-3927 - ARCH Define the methodology for modelling non-functional attributes IN PROGRESS</p></div> <div data-bbox="427 1749 845 1933"><p>RCAMT-160 - RCA-METH Derive basic modeling patterns for NFRs / ARCH Define the methodology for modelling non-functional attributes IN IMPLEMENTATION</p></div>

Tools and non-human resources	Team for Capella
Cardinality	Once per system capability
Completion criteria	The output view conforms to its modelling rules.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.142 Create system exchange scenarios with risk control measures

[SM-3062](#) - Populate Confluence page for activity definition: ARCH.142 **FINISHED**

Goal	Capture the definitive set of interactions between the system and the actors needed to achieve a system capability
Requirements met by this process step	ISO 15288 6.4.3.3 b) 3)
Inputs	AMOD-056 System functions and exchanges (single system capability) AMOD-058 Initial system exchange scenario
Outputs	AMOD-117 System exchange scenarios with risk control measures [S.ES]
Methodology	Taking the initial exchange scenario as a base, create a new exchange scenario, now incorporating any new functions required to mitigate risks.
Tools and non-human resources	Team for Capella
Cardinality	Once per system capability
Completion criteria	The output view conforms to the modelling rules The risk control measures included are appropriate and adequate for mitigating the risks associated with the system capability.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.143 Create system functional chains with risk control measures

[SM-3063](#) - Populate Confluence page for activity definition: ARCH.143 **FINISHED**

Goal	Capture the definitive set of functional flows needed to achieve a system capability
Requirements met by this process step	ISO 15288 6.4.3.3 b) 3)
Inputs	AMOD-056 System functions and exchanges (single system capability) AMOD-059 Initial system functional chain description
Outputs	AMOD-118 System functional chain descriptions with risk control measures [SFCD]
Methodology	Taking the initial functional chain description as a base, create a new functional chain description, now incorporating any new functions required to mitigate risks.
Tools and non-human resources	Team for Capella
Cardinality	Once per system capability
Completion criteria	The output view conforms to its modelling rules The risk control measures included are appropriate and adequate for mitigating the risks associated with the system capability.
Design review	ARCH.R.3 System capability review
Step done by (Responsible)	System architect
Provides input to/assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.149 Update system data with risk control measures

[RCAMT-369](#) - Refine data model rules so that data can be reused on lower levels and not need to be automatic transferred **FINISHED**

Goal	Capture the new requirements for information/data flow between system functions that are needed for risk control
Requirements met by this process step	ISO 15288 6.4.3.3 b) 3
Inputs	AMOD-112 System common data objects [S.CDB] AMOD-113 System exchange items [S.CDB]
Outputs	AMOD-112 System common data objects [S.CDB] AMOD-113 System exchange items [S.CDB]
Methodology	<p>For every entry in the risk logs,</p> <ul style="list-style-type: none"> where the risk mitigation specifies the need for additional system data items <ul style="list-style-type: none"> create new elements (exchange items, data objects as necessary) allocate new elements to appropriate exchanges between system functions where the risk mitigation specifies a non-functional attribute on system data <ul style="list-style-type: none"> capture the attribute(s) according to methods to be defined in <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;"> SM-3927 - ARCH Define the methodology for modelling non-functional attributes IN PROGRESS </div> <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;"> RCAMT-160 - RCA-METH Derive basic modeling patterns for NFRs / ARCH Define the methodology for modelling non-functional attributes IN IMPLEMENTATION </div> among others, examples are constraints on data separation, redundancy <p>See steps ARCH.054 and ARCH.057 for more details on data modelling methodology.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per system capability

Completion criteria	The output view conforms to its modelling rules. All new functional exchanges identified during risk assessment for this system capability are supported with an appropriate set of data items.
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.150 Update system interface model with risk control measures

Goal	Capture new constraints on external interfaces arising from risk control measures
Requirements met by this process step	None defined
Inputs	AMOD-114 System interface definition AMOD-115 External interface behaviour [S.ES]
Outputs	AMOD-114 System interface definition AMOD-115 External interface behaviour [S.ES]
Methodology	<p>For every entry in the allocated risk control measures,</p> <p>where the risk mitigation specifies new interface layers and/or behaviours during the delivery of the current system capability,</p> <p>these should be added to the relevant view (or, if necessary, new views should be created);</p> <p>where the risk mitigation specifies a non-functional attribute against an interface layer and/or behaviour on the current diagram</p> <p>capture the attribute(s) according to methods to be defined in</p> <div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <p>SM-3927 - ARCH Define the methodology for modelling non-functional attributes IN PROGRESS</p> </div> <div style="border: 1px solid #ccc; padding: 5px;"> <p>RCAMT-160 - RCA-METH Derive basic modeling patterns for NFRs / ARCH Define the methodology for modelling non-functional attributes IN IMPLEMENTATION</p> </div> <p>See step ARCH.158 for more details on interface layer modelling methodology.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per system capability
Completion criteria	<p>The output view conforms to its modelling rules.</p> <p>All new interface layers identified during risk assessment for this system capability are accurately modelled for their structure and behaviour.</p>
Design review	

Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.163 Update system states

[SM-3526](#) - Create Confluence page for activity definition ARCH.163 FINISHED

Goal	Capture changes to the state-based behaviour of the system and/or actors arising from risk control measures
Requirements met by this process step	ISO 15288 6.4.3.3 b) 3
Inputs	AMOD-060 System/actor states
Outputs	AMOD-060 System/actor states (updated)
Methodology	<p>During risk assessment for a system exchange scenario, new states may be identified for the mitigation of a risk during the delivery of the current system capability, that were not known during ARCH.055.</p> <p>Either update existing instances of view AMOD-060 or, where no statefulness was previously identified for an op /entity, create a new one.</p> <p>In both cases, follow the modelling rules for AMOD-060.</p> <p>See steps ARCH.055 for more details on state modelling methodology.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per system capability
Completion criteria	<p>The output view conforms to its modelling rules.</p> <p>All new states identified during risk assessment for this system capability are captured on the appropriate view.</p>
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.164 Update function mapping to system states

[SM-3527](#) - Create Confluence page for activity definition ARCH.164 **FINISHED**

Goal	Capture changes to the state-based behaviour of the system and/or actors arising from risk control measures
Requirements met by this process step	ISO 15288 6.4.3.3 b) 3
Inputs	AMOD-055 System functions to state allocation map [Matrix]
Outputs	AMOD-055 System functions to state allocation map [Matrix] (updated)
Methodology	Where the risk assessment: identified new system functions necessary for this system capability or identified new states exhibited by entities/actors involved in this capability repeat the steps from ARCH.056 with the new system functions and/or states.
Tools and non-human resources	Team for Capella
Cardinality	Once per system capability
Completion criteria	The output view conforms to its modelling rules. The state to function mapping is complete for all functions involved with the current system capability.
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	None directly

ARCH.914 Produce the unregulated system documents

- [ARCH.062 Produce the concept of use \(CONUSE\)](#)
- [ARCH.061 Produce the concept of employment \(CONEMP\)](#)

ARCH.062 Produce the concept of use (CONUSE)

[SM-2804](#) - Populate Confluence page for activity definition: ARCH.062 Produce the concept of use (CONUSE) **FINISHED**

Goal	Inform stakeholders about the current IM concepts for use of the system
Requirements met by this process step	None defined
Inputs	AMOD-045 System capabilities AMOD-060 System/actor states AMOD-119 System context definition [SAB]
Outputs	AMOD-065 Concept of use
Methodology	<p>The CONUSE will be automatically generated from a combination of pre-defined boilerplate text and mo</p> <p>Dependent on enabler</p> <p>SM-3365 - SEA-METH::ARCH Tooling of unregulated doc generation CONOPS, CONUSE, CONEMP PAUS</p>
Tools and non-human resources	<p>Team for Capella</p> <p>CONUSE generation script</p> <p>Confluence</p>
Cardinality	Once for the whole system; iterate as required by artefact's audience (release on demand)
Completion criteria	<p>The CONUSE document has been successfully generated and complies with its rules;</p> <p>The sets of capabilities associated with each non-operation lifecycle state are complete given the curren understanding of how the system will be employed</p>
Design review	
Step done by (Responsible)	Engine room
Provides input to/assists (Contributes)	System architect
Uses outputs (Informed)	<p>Subsystem manufacturers</p> <p>System implementing teams</p> <p>System users</p>

ARCH.061 Produce the concept of employment (CONEMP)

[SM-2803](#) - Populate Confluence page for activity definition: ARCH.061 Produce the concept of employment (CONEMP) **FINISHED**

Goal	Inform stakeholders about the current DBB concepts for employment of SysABB
Requirements met by this process step	None defined
Inputs	<p>tbd</p> <p>tbd System capabilities associated with non-operation lifecycle state</p>
Outputs	AMOD-066 Concept of employment
Methodology	<p>The CONEMP will be automatically generated from a combination of pre-defined boilerplate text and model content.</p> <p>Dependent on enabler</p> <p>SM-3365 - SEA-METH::ARCH Tooling of unregulated doc generation CONOPS, CONUSE, CONEMP PAUSED</p>
Tools and non-human resources	<p>Team for Capella</p> <p>CONEMP generation script</p> <p>Confluence</p>
Cardinality	Once for the whole system; iterate as required by artefact's audience (release on demand)
Completion criteria	<p>The CONEMP document has been successfully generated and complies with its rules;</p> <p>The sets of capabilities associated with each non-operation lifecycle state are complete given the current understanding of how SysABB will be employed</p>
Design review	ARCH.R.4 System review - consolidated
Step done by (Responsible)	Engine room
Provides input to /assists (Contributes)	System architect
Uses outputs (Informed)	<p>SysABB subsystem manufacturers</p> <p>SysABB implementing teams</p>

ARCH.917 Finalise the system requirements

- [ARCH.152 Consolidate system functionality](#)
- [ARCH.157 Consolidate system data](#)
- [ARCH.081 Update constraints on system solution](#)
- [ARCH.109 Consolidate traceability between model elements at system level and model elements at operational level](#)
- [ARCH.082 Trade off system requirements and constraints](#)
- [ARCH.083 Agree the system requirements with stakeholders](#)

ARCH.152 Consolidate system functionality

[RCAMT-377](#) - Merge ARCH.152 and ARCH.174 into one process step **FINISHED**

Goal	Ensure consistency of system functions, their exchanged functional exchanges and that all system functions are allocated correctly to either the system or the system actors.
Requirements met by this process step	None.
Inputs	AMOD-103 Consolidated system functions All system functions All Actors
Outputs	AMOD-103 Consolidated system functions AMOD-133 Consolidated system function allocation It may be necessary to propagate any modifications to AMOD-055 System functions to state allocation map [Matrix] AMOD-056 System functions and exchanges (single system capability) AMOD-058 Initial system exchange scenario AMOD-059 Initial system functional chain description AMOD-115 External interface behaviour [S.ES] AMOD-117 System exchange scenarios with risk control measures [S.ES] AMOD-118 System functional chain descriptions with risk control measures [SFCD]

<p>Methodology</p>	<p>The system functions, their functional exchanges and the allocated exchange items have been developed using the input transferred from operational level (ARCH.093) or they have been derived per system capability (ARCH.088) and they have been enriched with risk measures in (ARCH.141).</p> <p>This activity shall ensure that the system functions for the system as a whole do fit together. As exemplary guidance the following aspects should be considered:</p> <ul style="list-style-type: none"> • Avoid capability silos: From the conglomerate of system functions that have been derived from one system capability some exchanges should be there to system functions derived somehow else (e.g. transferred from operational level, derived from risk measures). E.g. to get input needed from a supplier function or to send a result to a consumer function. • Commonality: Functions of similar scope might have been developed when derived from capabilities. Where commonality has been identified, it should be decided to combine functions. • Completeness: AMOD-103 should ideally contain all system functions defined, i.e. this view should provide the required overview to ensure that all functions are connected and no un-used elements are left (e.g. functions or ports). Also all functional exchanges should have exchange items allocated to (in support of ARCH.157). • On system level at the system level there should only be leaf-level functions • if possible, all system functions should be assigned to one and only one of the function categories defined within the modelling rules. • Consolidate the allocation of system functions : Add the functions currently allocated to the system and the system actors. <p>NB: most of these should already have been allocated when AMOD-114 or AMOD-117 have been done. All function on system level should be allocated to an actor or to the system.</p> <p>For each leaf-level function that is not yet allocated</p> <p>Determine if this function is ever used in an SES and/or an SFCD.</p> <p>If it is used</p> <p>determine (with the owner of the corresponding capability), if it can be allocated to a system or to which actor it should be allocated to;</p> <p>else if not</p> <p>make it a candidate for deletion;</p> <p>Verify that all functional exchanges that cross the system boundary are allocated to a component exchange.</p> <p>Arrange the output view in good order. This will be a large view and may not be easily printable.</p> <div data-bbox="306 1464 1484 1648" style="border: 1px solid black; padding: 10px; margin-top: 20px;"> <p> Caution!</p> <p>Where this process step results in changes to the existing system functions and/or exchanges, they must be updated on all views where they are used (including the viewpoints for the single system capabilities). the single-capability views must not be allowed to lose consistency with the consolidated views.</p> </div>
<p>Tools and non-human resources</p>	<p>Team for Capella</p>
<p>Cardinality</p>	<p>At least once per capability package prior to review ARCH.R.4.</p>

Completion criteria	<p>The output view conforms to its modelling rules.</p> <p>The whole set of system functions is safe enough to try.</p> <p>The allocation of system functions is an accurate representation of the system scope.</p>
Design review	<p>ARCH.R.4 System review - consolidated</p>
Step done by (Responsible)	<p>Lead system architect</p>
Provides input to /assists (Contributes)	<p>System engineer</p> <p>System architect owning a specific capability that uses a system function</p>
Uses outputs (Informed)	<p>None directly.</p>

ARCH.157 Consolidate system data

[RCAMT-369](#) - Refine data model rules so that data can be reused on lower levels and not need to be automatic transferred

FINISHED

Goal	Avoid duplications, ensure re-usability and consistency of modelled system data.
Requirements met by this process step	None defined
Inputs	AMOD-112 System data objects [S.CDB] AMOD-113 System exchange items [S.CDB]
Outputs	AMOD-112 System data objects [S.CDB] AMOD-113 System exchange items [S.CDB]
Methodology	<p>System data modelling has different drivers and is using various inputs (i.e. operational data, functional exchanges and the implementation of risk measures) as described in ARCH.151, ARCH.054 and ARCH.149.</p> <p>Hence during this activity the existing viewpoints AMOD-112 and AMOD-113 shall be reviewed and updated, where required with focus on the following aspects:</p> <ul style="list-style-type: none">• Re-usability and consistency: Are all common data objects been re-used as appropriate or are there potentially two or more specific data objects defined, which are very similar in scope? In this case it should be considered to create a common data object.• Completeness:<ul style="list-style-type: none">• System exchange items: Are all system exchange items actually linked to a system data items?• Functional exchange allocation: Are all system exchange items allocated to functional exchanges?• Viewpoints: Is there at least one instance of AMOD-112 per instance of AMOD-025?• Readability:<ul style="list-style-type: none">• Do the instances of AMOD-112 provide good readability to convey the full definition of the respective system data.• Is there also an overview of all system data available as working diagram available. This helps to ensure re-usability, consistency and completeness

Tools and non-human resources	Team for Capella
Cardinality	At least once per capability package prior to review ARCH.R.4.
Completion criteria	<p>The output views conform to their modelling rules.</p> <p>The exchange item definition and system data definition have been judged to be consistently defined (with respect to completeness, re-usability).</p> <p>All functional exchanges on system level have at least one exchange item allocated.</p>
Design review	
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	System engineer
Uses outputs (Informed)	None directly.

ARCH.081 Update constraints on system solution

[SM-2821](#) - Populate Confluence page for activity definition: ARCH.081 Collect all constraints in one set FINISHED

Goal	Enable the informed tradeoff of system needs against implementation constraints
Requirements met by this process step	EN 50126-1 7.5.2 ISO 15288 6.4.8.3 a) 5, 6.4.11.3 a) 5, 6.4.10.3 a) 4
Inputs	AMOD-010 Trade space assessment AMOD-012 System lifecycle model Reference specification V&V strategy (from Ref-IVV)
Outputs	AMOD-116 System implementation constraints Potential updates to: AMOD-050 Stakeholder non-functional requirements AMOD-044 Actor functions overview AMOD-056 System functions and exchanges (single system capability) AMOD-058 Initial system exchange scenario AMOD-059 Initial system functional chain description AMOD-114 System interface definition AMOD-117 System exchange scenarios with risk control measures [S.ES] AMOD-118 System functional chain descriptions with risk control measures [SFCD]
Methodology	For each specified input (and any other input that may be relevant): <ul style="list-style-type: none"> • assess each constraint in the contents; • identify if the constraint has a material effect on the implementation or solution space of the system; <ul style="list-style-type: none"> • if the constraint has no effect, note the rationale for the decision, and discard the constraint; • identify the best way to represent the constraint at system level (e.g. as a non-functional aspect, a new function, a limit or constraint on an existing function etc.); • model the constraint.
Tools and non-human resources	Team for Capella
Cardinality	Once per package of system capabilities
Completion criteria	All the constraints and concerns in the input materials have been either converted to system-relevant constraints in the model, or discarded as irrelevant, with justification.
Design review	ARCH.R.4 System review - consolidated
Step done by (Responsible)	Lead system architect
Provides input to /assists (Contributes)	System architect, operational concepts architect, systems engineer

**Uses outputs
(Informed)**

None directly

ARCH.109 Consolidate traceability between model elements at system level and model elements at operational level

Goal	
Requirements met by this process step	none defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.082 Trade off system requirements and constraints

[SM-2823](#) - Populate Confluence page for activity definition: ARCH.082 Trade off system requirements and constraints FINISHED

Goal	Decide on the optimal scope for the system
Requirements met by this process step	ISO 15288 6.4.3.3 c) 1
Inputs	AMOD-116 System implementation constraints AMOD-073 Behavioural specification models (TBD)
Outputs	AMOD-146 System needs and constraints tradeoff decision record
Methodology	<p>There exists a large number of viable and popular methods for carrying out architectural tradeoffs.</p> <p>To refine the definition of the goal of this activity: the intent is to determine the set of requirements that can be committed to while achieving the greatest possible value to stakeholders.</p> <p>For minor decisions, it will be sufficient for the lead system architect to make the decision, and justify it with a Rationale element in the model.</p> <p>For more complex decisions, a more formal method is needed to provide a robust basis for the decision.</p> <p>The preferred methodology for complex tradeoffs is Decision Analysis and Resolution, as defined by the Capability Maturity Model Integration programme.</p> <p>Briefly summarised, the DAR methodology requires:</p> <ul style="list-style-type: none"> • Statement of the problem (i.e. the decision being made) • Definition of the alternatives between which the decision is being made • Definition of the evaluation criteria • Collection of evaluation data • Evaluation of data, including weighting of data • Selection of preferred solution • Agreement of preferred solution with stakeholders (this step to be carried out separately in ARCH.083 Agree the system requirements with stakeholders) <p>A template for AMOD-146 System needs and constraints tradeoff decision record is provided here for applying the DAR methodology: Template for tradeoff records (DAR technique)</p> <p>Other, comparable architectural tradeoff methods such as ATAM, CESAM's tradeoff method, House of Quality/QFD may be used as long as they define the information above in a clear way.</p> <p>NB: it is permissible for a solution to be selected that is not shown by the analysis to be the optimal one, if there is an overbearing influence that was not considered in the evaluation criteria. However, in this case, this should be recorded on the template so that the history of the decision is traceable.</p>
Tools and non-human resources	<p>Tradeoff templates/tools as selected by the decision-maker</p> <p>Team for Capella</p>
Cardinality	Once per tradeoff needed between system needs and external constraints
Completion criteria	The preferred solution has been selected and its selection justified.

Design review	ARCH.R.4 System review - consolidated
Step done by (Responsible)	Lead system architect
Provides input to /assists (Contributes)	System architect, system engineer
Uses outputs (Informed)	Stakeholder

ARCH.083 Agree the system requirements with stakeholders

[RCAMT-395](#) - Populate Confluence page for activity definition: ARCH.083 Agree the system requirements with stakeholders **FINISHED**

Goal	
Requirements met by this process step	ISO 15288 6.4.3.3 d) 1
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.920 Define the logical architecture

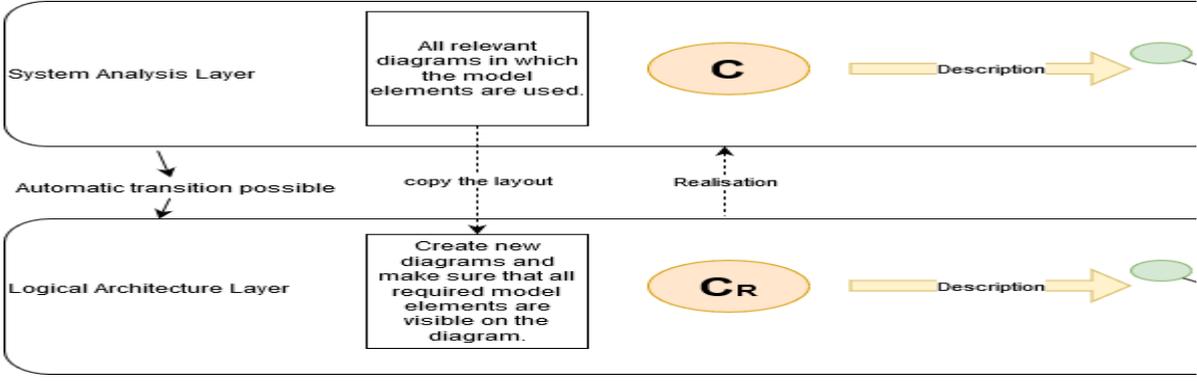
- [ARCH.921 Decompose system functionality](#)
 - [ARCH.084 Execute automatic transition of system elements to logical level](#)
 - [ARCH.182 Split the system functions](#)
 - [ARCH.085 Create or update functional chains at logical level](#)
 - [ARCH.087 Update logical data model](#)
 - [ARCH.191 Refine logical capability realisations](#)
- [ARCH.922 Define logical components](#)
 - [ARCH.180 Define logical component candidates](#)
 - [ARCH.187 Adopt logical component definitions from collaboration projects](#)
 - [ARCH.181 Reconcile candidate logical components](#)
- [ARCH.923 Apportion the non-functional requirements](#)
 - [ARCH.107 Apportion non-functional requirements to logical functions](#)
 - [ARCH.108 Define acceptance criteria for non-functional requirements](#)
- [ARCH.924 Finalise requirements at logical level](#)
 - [ARCH.154 Consolidate logical functional flow](#)
 - [ARCH.096 Allocate logical functions to logical components \(including alternative allocations\)](#)
 - [ARCH.086 Create or update exchange scenarios at logical level](#)
 - [ARCH.110 Consolidate traceability between model elements at logical level and model elements at system level](#)
 - [ARCH.190 Consolidate logical data](#)
 - [ARCH.099 Define behaviour for logical functions](#)

ARCH.921 Decompose system functionality

- [ARCH.084 Execute automatic transition of system elements to logical level](#)
- [ARCH.182 Split the system functions](#)
- [ARCH.085 Create or update functional chains at logical level](#)
- [ARCH.087 Update logical data model](#)
- [ARCH.191 Refine logical capability realisations](#)

ARCH.084 Execute automatic transition of system elements to logical level

Goal	Provide the top-down inputs needed to develop the logical architecture model in alignment with the system needs.
Requirements met by this process step	ISO 15288 6.4.4.3 a)
Inputs	All needed model elements from system layer of the analysis model. AMOD-056 System functions and exchanges (single system capability) AMOD-118 System functional chain descriptions with risk control measures [SFCD]
Outputs	AMOD-081 Logical functional flow definition (single system capability realisation) AMOD-082 Logical functional chain definition

<p>Methodology</p>	<p>The capella tool uses model layers to separate concerns so that the engineering process can be carried out in stages. Here elements from one layer of the model in any other layer.</p> <p>The 'system needs' have been modelled in the system layer of Capella. To work on these needs and break them down for later activities. During consolidation, the copies will be removed, and direct traceability will be established between de</p> <p><i>Note:</i> Not all model elements have to be decomposed. Some elements might not need to be further decomposed. System</p>  <p>The transitions needed are as follows:</p> <ul style="list-style-type: none"> ■ System capabilities to be automatically transferred to logical level. ■ System actors and allocated functions to be automatically transferred to logical level. ■ Component Exchanges to be automatically transferred to logical level. ■ System functions to be automatically transferred to logical level. ■ Functional Exchanges to be automatically transferred to logical level. ■ Functional chains (only system functional chain description with risk control measures) to be automatically transferred <p>Important note: Where elements already exist at logical level, it may be more appropriate to create a trace from the logical level that there is nothing appropriate at logical level that can be directly traced (via the "Realised <element type>" property) is valid (necessary and sufficient). Wherever this is done, ensure that any associated property values of a higher-level element</p> <p>Following initial viewpoints shall be created in this step:</p> <ul style="list-style-type: none"> • AMOD-081 Logical functional flow definition (single system capability realisation) • AMOD-082 Logical functional chain definition <p>Make sure that all the required model elements appear on the diagram.</p> <p>Please copy the layout from following system level viewpoints:</p> <ul style="list-style-type: none"> • AMOD-056 System functions and exchanges (single system capability) • AMOD-118 System functional chain descriptions with risk control measures [SFCD] <p><i>Note:</i> It is possible that initially other viewpoints must be created in this process step, these can be added here later.</p>
<p>Tools and non-human resources</p>	<p>Team for Capella</p>
<p>Cardinality</p>	<p>Carried out as required to maintain synchronisation between system and logical levels of the model; as a minimum once per system capability, at the point when the responsible team passes review ARCH.R.4 and moves i</p>

Completion criteria	<p>For the system capability, or package of capability in the scope of the transfer:</p> <ul style="list-style-type: none"> all system-level needs have been successfully transferred to new logical elements or traced from existing logical elements <p>Initial viewpoints are created with the transferred content and the layout is the same as the layout on the system level.</p>
Design review	<p>No design review checks this since it is an intermediate step, but the work should be coordinated and checked by the creator.</p>
Step done by (Responsible)	<ul style="list-style-type: none"> Engineer
Provides input to /assists (Contributes)	<p>Inclusive OR:</p> <ul style="list-style-type: none"> Engine room System architect Cross-cutting engineer
Uses outputs (Informed)	<p>None.</p>

ARCH.182 Split the system functions

[RCAMT-390](#) - Add mandatory rationales to ARCH.088, 182 and other steps that require functions to justify their inputs.

FINISHED

Goal	Each transferred system function of interest has been split according to defined splitting principles and met a required level of detail into the logical architecture level.
Requirements met by this process step	ISO 15288 6.4.4.3 c)
Inputs	AMOD-081 Logical functional flow definition (single system capability realisation) AMOD-082 Logical functional chain definition AMOD-092 Logical exchange items (LCDB) FC's Architecture.pdf (Constraint) Example: Patterns for decomposing system functions into logical functions
Outputs	AMOD-081 Logical functional flow definition (single system capability realisation)

Methodology

The logical level of the architecture is concerned with abstract partitioning of functionality. Therefore, when we split functions at logical level, we can only do this on the basis of some **logical** or abstract splitting principle.

- [Define the relevant splitting principle](#)
- [Preparation of the tool](#)
- [Split one function of interest](#)

Define the relevant splitting principle

1. Identify if system level functions are already categorised into specific **function categories** as defined in the relevant function pattern
 - a. E.g. Railway operation functions (control, actuate, indicate, sense, observe or the placeholder for plant (actor) functions)
 - b. E.g. Map data (provide, maintain)
 - c. E.g. Data maintenance or software update functions (pattern yet to be defined)
2. Identify the **operated object types** (Operated object means one of two things: concrete railway assets or abstract concepts). The "object" here is the object of the phrase that defines the function, for example "sense the rotational speed of one axle" has operated object "axle".
3. Identify for each operated object the **types of state** that can change over time. This could be:
 - a. Discrete state (e.g. state type "point position" has possible values "left", "moving/unlocked", "right")
 - b. Continuous state in the sense of state-space modelling (scalar or vector quantities such as position, speed, acceleration).
4. Identify **abstraction layers** for the logical architecture which can be used to decompose functions according to different level of abstraction:
E.g. for functions the abstraction layers are defined as following for the project:

One area of control (yellow): functions that affect states of multiple objects in an area of control (e.g. functions dealing with operational plans or movement permissions)

Trackside object of one area of control (red): functions that affect states of only trackside objects of one area of control (e.g. functions for point machines)

One train unit (green): functions that affect only states of objects of one train unit (e.g. functions for movement supervision on one train unit)

Abstraction layer name	Description and examples
Plan implementation in one area of control	Functions transforming <ul style="list-style-type: none">• between an abstract plan and one state of one CCS abstract object (e.g. operational plan to required drive protection section position state)• between two types of abstract plans (e.g. operational plan to journey profile)
Safety control in one area of control	Functions authorising one safe state of one abstract CCS object (e.g. position state of one drive protection section or speed profile "state" of one movement permission)
Object abstraction in one area of control	Functions transforming one state between different abstract CCS objects (e.g. TVPS state from/to moveable object state)
Device abstraction in one area of control	Functions transforming <ul style="list-style-type: none">• one state between one abstract CCS object and one concrete trackside object (e.g. point state from/to drive protection section state)• one state between one abstract CCS object and one abstract train unit object (e.g. movement permission to movement authority)
Device control of one trackside object in one area of control	Functions controlling/observing/actuating/sensing one state of one concrete trackside object (e.g. one point machine)
Plan implementation on one train unit	Functions transforming between an abstract plan and one state of an abstract train unit object (e.g. journey profile to required target speed profile)
Safety control on one train unit	Functions controlling safety one safe state of one train unit object (e.g. emergency brake activation, safe door release)
Object abstraction on one train unit	Functions transforming one state between different abstract train unit objects or one state between one abstract and one concrete train unit object
Object control or sense on one train unit	Functions controlling/observing/actuating/sensing one specific state of one train unit (e.g. will mostly be inside TCMS actor)

5. Identify whether functions can be divided according to **safety criticality** (safety critical or not safety critical).
Note: Even when this is not otherwise a logical split because the functionality is the same in both cases, just with a different value of safety performance required, there could be situation where a logical function is duplicated both inside and outside the system border, with one instance being safety-critical and one being non-safety critical, just at a lower level of safety criticality. There are 2 possible values: safety-critical and non-safety-critical.

Preparation of the tool

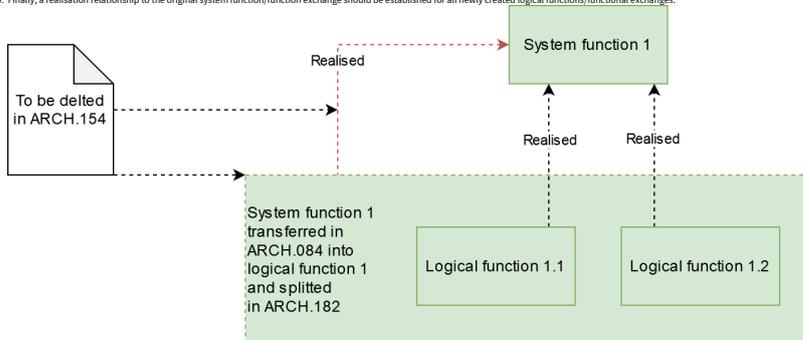
- As a starting point choose a realised capability and the defined functional chain for this capability. The functional chain description [AMOP_082](#) was initially created in [ARCH_084](#).
- This is the functional chain of interest within this chain are the core functions of interest, those that are linked with sequence links. These functions should be now split, if needed.
- Functions which only provide information to a function of interest should be decomposed as part of other realised capabilities (if they are already decomposed, the LDFB needs to be updated/refreshed so that the last leaf level function is shown on the diagram).
- Open the [AMOP_082](#) diagram for the chosen capability (initially created in [ARCH_084](#)). The diagram should show all the functions of interest for the functional chain which describes the capability.
- It can help to use text boxes on the LDFB diagram to show the relevant logical or abstract splitting principles behind the functions. (note: this is helpful to add later the mandatory splitting rationale against the function)

Split one function of interest

Note: If there exist already logical functions in the model, this functions shall be reconciled with the transferred system functions or with the new leaf level logical functions as part of this process step. So that at the end only one set of consistent logical functions exists which are traced to system functions.

Decide now for each function of interest which splitting principles to apply. Perform the following steps:

1. Check if there are any mistakes or misunderstandings regarding the pattern for **function categories** and check if a further split is needed based on the functional category. If not, decompose this logical function of interest, so that the function can fit into the defined functional categories.
2. Check which individual **operated object type** should be handled by the function. Confirm that the function operates only one individual object into the logical level, taking into consideration that one function that controls one individual operated object type can be instantiated many times. If the splitting principle seems correctly applied, split this logical function of interest so that the function can operate the individual operated object type as required.
3. Check if the function needs to be split according to **state type**. This will depend on how much detail it is considered useful to show. For example, when controlling the state of one point machine, it is only necessary to show that we are controlling the left/right drives, not that we are controlling the individual states of each wire on the interface; on the other hand, when controlling the state(s) of one train unit, we are interested separately in the state of the emergency brake, the state of the service braking and traction, the state of the air-tightness, the state of the brake types enabled... so it makes sense to split the functions down in this instance.
4. Check if this function needs to be split along the defined **abstraction layers**.
 - Split this logical function of interest so that the function can provide the required functionality on each abstraction layer.
 - Split a function only for a specific layer when the input state and the output state of the function have been modified. Functions should only be created within a layer if an additional new functionality is really added or an operated state can assume a different state by the new function.
5. Check if this function needs to be split into **safety critical and not safety critical** function. Decompose this logical function of interest, so that there is one safety-critical function and one without safety criticality.
6. Split/Create the **functional exchanges** between the split functions and connect the existing functional exchanges from the parent function to the new leaf function.
7. Define a rationale to justify **why** the system function was split and the splitting principles which were taken as the basis for the split.
8. Define a description for each newly created function.
9. Finally, a realisation relationship to the original system function/function exchange should be established for all newly created logical functions/functional exchanges.



Note: during consolidation in [ARCH_154](#), the function that has been transferred to logical layer and split into logical functions will be deleted.
10. Start with the next function of interest or finish by cleaning up the diagrams/views.

Tools and non-human resources

Team for Capella

Cardinality

Once per functional chain of interest.

Completion criteria	The output view complies with its modelling rules. The content of the output view represents the team's current understanding of the way these system functions should be decomposed in the logical architecture layer.
Design review	ARCH.B.5 Logical capability review
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Inclusive OR: <ul style="list-style-type: none"> • Subsystem architect • Engineer • Expert for GoM railway operation • Cross-cutting engineer
Uses outputs (Informed)	RAMS manager RAMS engineer Verification & validation manager

ARCH.085 Create or update functional chains at logical level

[SM-4715](#) - Modify methodology of ARCH.085 **FINISHED**

Goal	Refine the functional chains with decomposed logical function definitions
Requirements met by this process step	ISO 15288 6.4.4.3 c) 4
Inputs	AMOD-082 Logical functional chain definition AMOD-081 Logical functional flow definition (single system capability realisation)
Outputs	AMOD-082 Logical functional chain definition

<p>Methodology</p>	<p>Top-down:</p> <p>The viewpoint AMOD-082 Logical functional chain definition was initially created in step ARCH.084 after the functional chain was transferred from system level.</p> <p>Within step ARCH.182 the functions are further decomposed so the initially created logical functional chains must be updated, so that the newly created functions of interest and functions which provide information are now respresented in AMOD-082 Logical functional chain definition.</p> <p>Do the following modifications on AMOD-082:</p> <p style="padding-left: 40px;">If involved functions were decomposed during ARCH.182</p> <p style="padding-left: 80px;">Replace the function involvements on the functional chain such that only leaf-level functions are involved</p> <p style="padding-left: 40px;">end if</p> <p style="padding-left: 40px;">If the logic of the functional chain has changed because of alterations to arising from decomposition in ARCH.182</p> <p style="padding-left: 80px;">update the logic of the functional chain so that it is consistent with the rest of the model</p> <p style="padding-left: 40px;">end if</p> <p>Bottom-up:</p> <p>If there are already logical functions that exist in the model from previous activities that happened before the ARCH process was fully introduced, the sequential order of these functions can be described in a function chain. The functional chain should be located underneath an assumed corresponding capability realisation. However, it is important that the functions used in this chain are merged with the top-down functions in process step ARCH.182. The same should happen with the functional chain in this step. So that at the end of this process step, the top down and bottom up functions produce a common result in a combined functional chain.</p>
<p>Tools and non-human resources</p>	<p>Team for Capella</p>
<p>Cardinality</p>	<p>Once per capability realisation</p>
<p>Completion criteria</p>	<p>The output views conform to their modelling rules</p>
<p>Design review</p>	<p>ARCH.R.5 Logical capability review</p>

Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	Inclusive OR: <ul style="list-style-type: none">• Subsystem architect• Engineer• Cross-cutting engineer
Uses outputs (Informed)	None identified

ARCH.087 Update logical data model

[RCAMT-369](#) - Refine data model rules so that data can be reused on lower levels and not need to be automatic transferred

FINISHED

Goal	Capture the requirements for information/data flow between logical functions
Requirements met by this process step	None.
Inputs	AMOD-081 Logical functional flow definition (single system capability realisation) AMOD-025 Abstract concepts AMOD-112 System data objects [S.CDB] AMOD-113 System exchange items [S.CDB]
Outputs	AMOD-025 Abstract concepts (Updated) AMOD-091 Logical data objects [L.CDB] AMOD-092 Logical exchange items [L.CDB]

<p>Methodology</p>	<p>During the ARCH.182 new functional exchanges will be created. In case additional data or information must be defined that flows along these the following methodology shall be used:</p> <p>Exchange items</p> <p>Check all instances of AMOD-110 Operational exchange items [O.CDB] and AMOD-113 System exchange items [S.CDB] and AMOD-092 Logical exchange items [L.CDB] for existing reusable exchange items, before creating new ones. It is not allowed to have identical data model elements on multiple levels. If there is a need to refine or change an exchange item from system level, please create a new exchange item at the logical level and link it via a realisation to the corresponding data object at the higher level.</p> <p><i>Note:</i> New data objects and types should be modelled initially on AMOD-092 so that other modellers can look for them and reuse them where appropriate.</p> <p>Each functional exchange should have allocated at least one exchange item.</p> <p>The data information between two logical functions is permanently present at the logical level, therefore FLOW should be used as communication mechanism for exchange items in all cases.</p> <p>In some exceptional cases, it is allowed to use an EVENT, but only for functional exchanges between an actor function and a logical function allocated to a logical component if the interface is constrained by an external standard.</p> <p>Define for each exchange item the types of data, via the Exchange Item Elements. The information content of the exchange item should be defined as far as possible using an appropriate combination of data objects (simple types or structured types).</p> <p>Where the information content is unknown or unclear, write a text description on the exchange item, to act as a placeholder/STUB until the information content is resolved (communication mechanism of exchange item = UNSET).</p> <p>Data objects</p> <p>Check all instances of AMOD-105 Operational data objects [O.CDB] and AMOD-112 System data objects [S.CDB] and AMOD-091 Logical data objects [L.CDB] for existing reusable data objects and types, before creating new ones. It is not allowed to have identical data model elements on multiple levels. If there is a need to refine or change a data object from system level, please create a new data object at the logical level and link it via a realisation to the corresponding data object at the higher level.</p> <p>When a new data object is created on logical level the rationale for their creation must be documented.</p> <p><i>Note:</i> New data objects and types should be modelled initially on AMOD-091 so that other modellers can look for them and reuse them where appropriate.</p> <p>Instances of AMOD-091 should be structured in line with the structure used for the abstract concepts (so that there is generally one instance of AMOD-091 per instance of AMOD-025 - unless it is necessary to split the views up for readability).</p> <p>Abstract concepts</p> <p>Abstract concepts are defined on operational level in AMOD-025 Abstract concepts. These concepts may be updated, and/or new concepts might be identified during definition of the logical architecture. In such a case, the person responsible for the abstract concept (e.g. operational concept architect) must be involved in the change process.</p> <p>If a new abstract concept is to be defined, follow the process step ARCH.008 Define abstract concepts.</p>
<p>Tools and non-human resources</p>	<p>Team for Capella</p>
<p>Cardinality</p>	<p>Once per system capability</p>

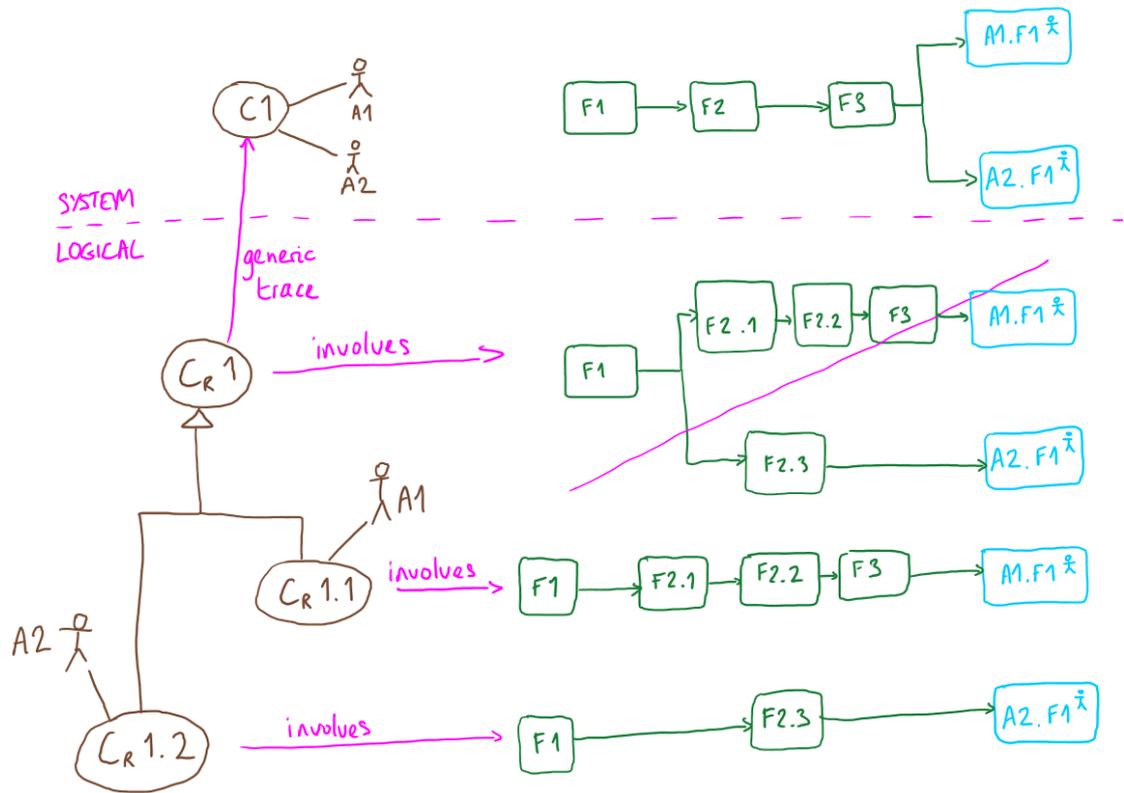
Completion criteria	<p>The output views conform to their modelling rules.</p> <p>The exchange item definitions represent the best current knowledge of the information content required for exchange at logical level</p>
Design review	ARCH.R.5 Logical capability review
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	<ul style="list-style-type: none"> • System Engineer • Subsystem Architect
Uses outputs (Informed)	None directly

ARCH.191 Refine logical capability realisations

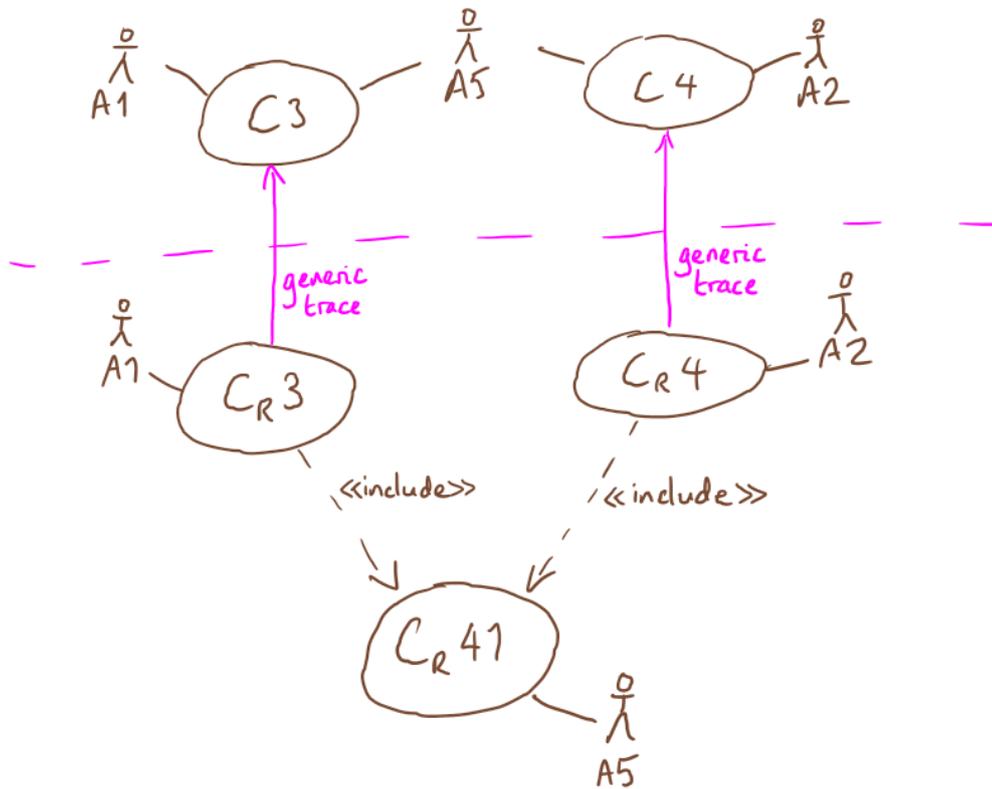
Draft

[SM-5412](#) - Create Confluence page for process step ARCH.191 Refine logical capability realisations **FINISHED**

Goal	Capture any refinements/adjustments to system capabilities after splits are done at logical level
Requirements met by this process step	ISO 15288 6.4.3.3 d) 2) EN 50126 7.6.2
Inputs	AMOD-045 System capabilities
Outputs	AMOD-147 Logical capability realisation
Methodology	<p>For the most part, the system capabilities are not expected to change when transferred down to logical level.</p> <p>However, the "logical capability realisations" still need to be shown on a dedicated viewpoint for at least two reasons:</p> <ol style="list-style-type: none">1. this viewpoint is a statement of the capabilities that will be tested in the simulation of the logical architecture and therefore is an input to SIM and IVV activities.2. during decomposition of logical functions it may arise that a system capability must be split or specialised because the functional splitting results in variations in the functional chain. <p>Some example conditions (non-exhaustive) under which it may be appropriate to refine the system capabilities at logical level:</p> <p>a) after functional decomposition, the capability's value can be independently delivered to two or more actors.</p> <p>That means that the end conditions of the capability can now be expressed in two or more independent groups.</p> <p>The capability realisation should be refined by capability generalisation - that is, there are now two independent types of the original capability.</p>



b) after functional decomposition, a functional chain portion is defined that can be separated off and reused as a capability in its own right; in this case, the new capability realisation should be **included** in the capability realisations it is reused within.



**Important**

The amount of refinement of capability realisations at logical level should be kept to the minimum necessary, because each new capability realisation creates new work to do in verification and validation.

Further refinement will be allowed at subsystem level - see [ARCH.192 Refine physical capability realisations](#)

Tools and non-human resources	Team for Capella
Cardinality	Once per package of system capabilities being delivered
Completion criteria	The output view conforms to its modelling rules.
Design review	ARCH.R.5 Logical capability review ARCH.R.6 Logical review - consolidated
Step done by (Responsible)	System architect
Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	IVV role TBD RCAMT-400 - Updates to ARCH to integrate artefact usage and role involvement with IVV CREATED

ARCH.922 Define logical components

- [ARCH.180 Define logical component candidates](#)
- [ARCH.187 Adopt logical component definitions from collaboration projects](#)
- [ARCH.181 Reconcile candidate logical components](#)

ARCH.180 Define logical component candidates

[RCAMT-390](#) - Add mandatory rationales to ARCH.088, 182 and other steps that require functions to justify their inputs.

FINISHED

Goal	Define logical component candidates based on splitting principles.
Requirements met by this process step	ISO15288 6.4.4.3 c)
Inputs	Concepts and other informations from railway and RAMS experts.
Outputs	AMOD-125 Logical architecture definition

<p>Methodology</p>	<p>The logical level of the architecture is concerned with abstract partitioning of functionality - not with practical considerations such as the geographical location of a component, network bandwidth, power supply, packaging of hardware, etc.</p> <p>Therefore, when we split functions at logical level, we can only do this on the basis of some logical or abstract splitting principle.</p> <p>Functions can then be grouped into logical components when they have the same combination of splitting principle values.</p> <p>This process step should be carried out together with ARCH.182 Split the system functions.</p> <p>After all possible splitting principles have been identified in ARCH.182, the set of candidate logical components can be thought of as an n-dimensional vector space where each dimension corresponds to one logical splitting principle.</p> <p>Based on the maximum set of splitting principles, these principles are grouped in a meaningful way, such that a manoeuvrable number of logical components are defined. These then produced several version of AMOD-125 Logical architecture definition, i.e. the candidates of logical architecture.</p> <p>Potentially then there are hundreds of candidate logical components.</p> <p>It is important to recognise that</p> <ul style="list-style-type: none"> not every single one of these candidates will actually be needed (Final check in ARCH.181 Reconcile candidate logical components) there is not a 1:1 mapping between these logical components and subsystems; rather, one subsystem will host many logical components; candidate logical components should only be created in the model where there exists at least one logical function with the corresponding values of splitting principle. Define a rationale for each logical component to justify why the logical component is needed. <hr/> <p>Example 1 for a matrix to define the candidates:</p> <p>We have defined two splitting principles:</p> <p>Splitting principle 1: safety criticality (possible options <i>safety-critical</i> or <i>non-safety-critical</i>)</p> <p>Splitting principle 2: system function category (possible options <i>control</i>, <i>actuate</i>, <i>sense</i>, <i>observe</i>, <i>plant</i>)</p> <p>Then the set of candidate logical components is a 2-dimensional vector space containing 10 possible combinations of values (that is, a 2 x 5 matrix):</p> <table border="1" data-bbox="236 1043 788 1301"> <thead> <tr> <th>Function category</th> <th>Safety-critical control</th> <th>Non-safety-critical control</th> </tr> </thead> <tbody> <tr> <td>Control</td> <td>Safety-critical control</td> <td>Non-safety-critical control</td> </tr> <tr> <td>Actuate</td> <td>Safety-critical actuate</td> <td>Non-safety-critical actuate</td> </tr> <tr> <td>Sense</td> <td>Safety-critical sense</td> <td>Non-safety-critical sense</td> </tr> <tr> <td>Indicate</td> <td>Safety-critical indicate</td> <td>Non-safety-critical indicate</td> </tr> <tr> <td>Observe</td> <td>Safety-critical observe</td> <td>Non-safety-critical observe</td> </tr> <tr> <td>Plant</td> <td>Safety-critical plant</td> <td>Non-safety-critical plant</td> </tr> </tbody> </table> <p>Example 2 for a matrix to define the candidates:</p> <p>We have defined 4 splitting principles:</p> <p>Splitting principle 1: safety criticality (possible options <i>safety-critical</i> or <i>non-safety-critical</i>)</p> <p>Splitting principle 2: system function category (possible options <i>control</i>, <i>actuate</i>, <i>indicate</i>, <i>sense</i>, <i>observe</i>, <i>plant</i>)</p> <p>Splitting principle 3: operated object / abstract concepts (<i>Point machine</i>, <i>Usage Restriction Area</i>, <i>Movement Permission</i>)</p> <p>Splitting principle 4: RCA logical abstraction layers (<i>Plan Implementation Layer</i>; <i>Safety Control Layer</i>; <i>Object Aggregation Layer</i>; <i>Device Abstraction Layer</i>; <i>Device Control Layer</i>)</p> <p>In total there are 2 x 6 x n x 5 (that is, 60n) possible combinations of values, where n is the number of operated objects defined.</p> <p>Further example model views and explanations are provided in Pattern for decomposing system functions into logical functions.</p>	Function category	Safety-critical control	Non-safety-critical control	Control	Safety-critical control	Non-safety-critical control	Actuate	Safety-critical actuate	Non-safety-critical actuate	Sense	Safety-critical sense	Non-safety-critical sense	Indicate	Safety-critical indicate	Non-safety-critical indicate	Observe	Safety-critical observe	Non-safety-critical observe	Plant	Safety-critical plant	Non-safety-critical plant
Function category	Safety-critical control	Non-safety-critical control																				
Control	Safety-critical control	Non-safety-critical control																				
Actuate	Safety-critical actuate	Non-safety-critical actuate																				
Sense	Safety-critical sense	Non-safety-critical sense																				
Indicate	Safety-critical indicate	Non-safety-critical indicate																				
Observe	Safety-critical observe	Non-safety-critical observe																				
Plant	Safety-critical plant	Non-safety-critical plant																				
<p>Tools and non-human resources</p>	<p>Team for Capella</p>																					
<p>Cardinality</p>	<p>One off after all possible logical splitting principles have been identified. But with allowed iterations after different candidates emerge from review steps.</p>																					
<p>Completion criteria</p>	<p>The output view complies with its modelling rules.</p> <p>Maximal set of logical component candidates defined and implemented in a manoeuvrable number of logical components.</p>																					
<p>Design review</p>	<p>ARCH.R.5 Logical capability review</p> <p>ARCH.R.6 Logical review - consolidated</p>																					

Step done by (Responsible)	<ul style="list-style-type: none"> • System architect
Provides input to /assists (Contributes)	<p>Inclusive OR:</p> <ul style="list-style-type: none"> • Engineer • Subsystem architect • Expert for GoA4 railway operation • Cross-cutting engineer
Uses outputs (Informed)	<p>RAMS manager</p> <p>RAMS engineer</p> <p>Verification & validation manager</p>

ARCH.187 Adopt logical component definitions from collaboration projects

Goal	Identification and adoption of relevant logical components from collaboration projects.
Requirements met by this process step	None
Inputs	Defined logical components in collaboration projects (e.g. EULYNX, S2R or RCA)
Outputs	AMOD-125 Logical architecture definition
Methodology	<p>Analyse and compare the provided content from the collaboration project with the project`s logical componer candidates defined in ARCH.180 Define logical component candidates.</p> <p>Outcome of the analysis may result in:</p> <ul style="list-style-type: none"> ▪ Going back to the collaborative project and discuss potential need to modifying the logical components coming from that project. ▪ Updating the defined logical component candidates in AMOD-125 to adapt to logical components from collaborative project. This shall be reconciled accordingly as part of ARCH.181.
Tools and non-human resources	Capella for Teams
Cardinality	Once per collaboration project, with permitted revisions
Completion criteria	<p>All logical components candidates are identified and adopted and implemented into the model.</p> <p>The output view conforms to its modelling rules.</p>
Design review	Completion criteria agreed between lead system architect, system and subsystem architects.
Step done by (Responsible)	Lead System Architect
Provides input to /assists (Contributes)	<ul style="list-style-type: none"> • System architect • Subsystem architect • Engineer
Uses outputs (Informed)	Responsible person of the collaboration project

ARCH.181 Reconcile candidate logical components

Goal	To ensure that there are no gaps or duplicates between logical components created locally and those inherited from international collaborations.
Requirements met by this process step	ISO 15288 6.4.4.3 c) 4 ISO/IEC/IEEE 42010 5.8
Inputs	AMOD-125 Logical architecture definition AMOD-081 Logical functional flow definition (single system capability realisation)
Outputs	AMOD-125 Logical architecture definition
Methodology	<p>For each logical component that supports the capability realisation of interest:</p> <ol style="list-style-type: none"> 1. Consolidate the description for each logical component to ensure its purpose is clearly described, i.e. that splitting principles that are covered by this component are documented and rationales are provided where multiple principles are grouped. Additionally considerations about alternative logical components may be given. 2. Consolidate names of logical components to ensure that its purpose can be derived from that name. 3. Compare all created logical components from ARCH.180 and ARCH.187 with each other based on name and description <ol style="list-style-type: none"> a. Identify duplicated logical components. If there are any, choose one logical component and delete the other o b. Identify missing logical components if there are any, go back to ARCH.180 and ARCH.187 to validate if this is re the case. <p>end for</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per capability realisation. Iterations are allowed.
Completion criteria	<p>Duplicate logical components have been reconciled.</p> <p>Gaps in logical component coverage have been closed (where it is anticipated or known that a corresponding logical function exists).</p>
Design review	ARCH.R.5 Logical feature review
Step done by (Responsible)	<ul style="list-style-type: none"> • System architect
Provides input to /assists (Contributes)	<ul style="list-style-type: none"> • Lead system architect • Subsystem architect • Engineer • Cross-cutting engineer
Uses outputs (Informed)	<p>RAMS manager</p> <p>RAMS engineer</p> <p>Verification & validation manager</p>

ARCH.923 Apportion the non-functional requirements

- [ARCH.107 Apportion non-functional requirements to logical functions](#)
- [ARCH.108 Define acceptance criteria for non-functional requirements](#)

ARCH.107 Apportion non-functional requirements to logical functions

[RCAMT-265](#) - Populate Confluence page for activity definition: ARCH.107 Apportion non-functional requirements to logical functions

CREATED

Goal	<i>high-level statement of the engineering objective being addressed, including reference to standards/legislation if the relevant</i>
Requirements met by this process step	EN 50126-1 7.6.1
Inputs	AMOD-082 Logical functional chain definition AMOD-081 Logical functional flow definition (single system capability realisation) AMOD-091 Logical data objects [L.CDB] AMOD-092 Logical exchange items [L.CDB] AMOD-083 State-based behaviour definition (logical function)
Outputs	<i>None.</i>
Methodology	<i>state the methodology as a list of instructions; this information forms the Description field of a ticket for this process to be done, so it must be sufficient for someone to follow the process, even if they have not done it before.</i>
Tools and non-human resources	Team for Capella
Cardinality	<i>state how many times this process step is expected to be carried out - e.g.</i> <ul style="list-style-type: none"> • one-off • once per <model element> • once per baseline <i>etc.</i>
Completion criteria	<i>state the conditions that must be TRUE before this process step can be considered complete.</i> <i>These should be inserted in the Acceptance Criteria of a Jira ticket for this process step to be done.</i>
Design review	<i>Link to the corresponding design review where the completion of this activity is evaluated.</i>
Step done by (Responsible)	<i>Identify the role that is responsible for organising this process step and producing the output;</i> <i>This is the Assignee in Jira.</i>
Provides input to /assists (Contributes)	<i>Identify the roles that assist in the work or provide input information</i>
Uses outputs (Informed)	<i>Identify the roles and/or process areas that make use of this information outside of the ARCH process area;</i>

ARCH.108 Define acceptance criteria for non-functional requirements

[RCAMT-266](#) - Populate Confluence page for activity definition: ARCH.108 Define acceptance criteria for non-functional requirements

CREATED

Goal	<i>high-level statement of the engineering objective being addressed, including reference to standards/legislation if that is relevant</i>
Requirements met by this process step	EN 50126-1 7.6.1
Inputs	<i>list of all input information needed for this process step to be carried out; for model views, add a hyperlink to the model view description; documents, published info, websites, knowledge from particular roles are all admissible.</i>
Outputs	<i>list of all output artefacts this process should produce; for model views, add a hyperlink to the model view description; for documents, add a link to the document template or to the document publication area if it is directly generated from Confluence;</i>

Methodology	<p>Only a summary of an initial investigation:</p> <ul style="list-style-type: none"> • Acceptance criteria are used to comply with DIN EN 50126 phase 5 (7.6.1.d, Abnahmekriterien). These criteria are defined in order to proof RAMS requirements are satisfied. • Acceptance criteria are conditions the product must satisfy in order to be accepted by the customer. • Acceptance criteria help to avoid unexpected development results. <ul style="list-style-type: none"> • Acceptance criteria can be provided in different formats: scenario oriented or rule oriented. <ul style="list-style-type: none"> • Scenario oriented Acceptance criteria are written in the Given /When/Then format. Under Given conditions, when specific action is performed, then the required outcome of the scenario is provided. • Rule oriented Acceptance criteria are simply a list of verifiable criteria that are required to satisfy a specific user story • Rules for acceptance criteria: <ul style="list-style-type: none"> • Document them before development starts • Do not add technical details • Acceptance criteria must be achievable • Acceptance criteria must be testable • Acceptance criteria must be understandable for everyone • Acceptance criteria must be written from User perspective
Tools and non-human resources	<p>Capella for Teams</p>
Cardinality	<p><i>state how many times this process step is expected to be carried out - e.g.</i></p> <ul style="list-style-type: none"> • <i>one-off</i> • <i>once per <model element></i> • <i>once per baseline</i> <p><i>etc.</i></p>

Completion criteria	<p><i>state the conditions that must be TRUE before this process step can be considered complete.</i></p> <p><i>These should be inserted in the Acceptance Criteria of a Jira ticket for this process step to be done.</i></p>
Design review	<p><i>Link to the corresponding design review where the completion of this activity is evaluated.</i></p>
Step done by (Responsible)	<p><i>Identify the role that is responsible for organising this process step and producing the output;</i></p> <p><i>This is the Assignee in Jira.</i></p>
Provides input to /assists (Contributes)	<p><i>Identify the roles that assist in the work or provide input information</i></p>
Uses outputs (Informed)	<p><i>Identify the roles and/or process areas that make use of this information outside of the ARCH process area;</i></p>

ARCH.924 Finalise requirements at logical level

- [ARCH.154 Consolidate logical functional flow](#)
- [ARCH.096 Allocate logical functions to logical components \(including alternative allocations\)](#)
- [ARCH.086 Create or update exchange scenarios at logical level](#)
- [ARCH.110 Consolidate traceability between model elements at logical level and model elements at system level](#)
- [ARCH.190 Consolidate logical data](#)
- [ARCH.099 Define behaviour for logical functions](#)

ARCH.154 Consolidate logical functional flow

Goal	Ensure consistency of logical functions and their exchanged functional exchanges for each logical component.
Requirements met by this process step	None.
Inputs	AMOD-081 Logical functional flow definition (single system capability realisation)
Outputs	AMOD-088 Consolidated logical functional flow definition AMOD-081 Logical functional flow definition (single system capability realisation) (updated)

<p>Methodology</p>	<p>The logical functions, their functional exchanges and the allocated exchange items have been developed using the input transferred from system level (ARCH.084) or they have been derived per capability realisation (ARCH.182/ARCH.085/ARCH.086).</p> <p>This activity shall ensure that the logical functions for each logical component as a whole do fit together. As exemplary guidance the following aspects should be considered:</p> <ul style="list-style-type: none"> • Avoid capability silos: From the conglomerate of logical functions that have been derived from one capability realisation some exchanges should be there to logical functions derived somehow else (e.g. transferred from system level, derived from risk measures). E.g. to get input needed from a supplier function or to send a result to a consumer function. In other words, maximise the re-use of logical functions in more than one capability realisation, as opposed to having new and separate functions for each capability realisation. • Commonality: Functions of similar scope might have been developed when derived from capabilities. Where commonality has been identified, it should be decided to combine functions. • Removal of duplicates: Where duplication or scope overlaps are identified, resolve these with common functions where possible, re-link/re-involve functions that have changed (including updating functional chains and scenarios) and delete the duplicates only once they have been fully disconnected from the rest of the model. • Removal of transferred system function which were splitted into logical functions in ARCH.182. Before deleting a function please check if other diagrams are affected, if this is the case correct the layout of this diagrams by removing the function out of this diagram. Also check if all realised links from the logical functions to the corresponding system function are set correctly. • Completeness: AMOD-088 should ideally contain all logical functions defined, i.e. this view should provide the required overview to ensure that all functions are connected and no un-used elements are left (e.g. functions or ports). Also all functional exchanges should have exchange items allocated to.
<p>Tools and non-human resources</p>	<p>Team for Capella</p>
<p>Cardinality</p>	<p>At least once per capability package prior to review ARCH.R.6.</p>
<p>Completion criteria</p>	<p>The output view conforms to its modelling rules.</p> <p>The whole set of system functions is safe enough to try.</p>
<p>Design review</p>	<p>ARCH.R.6 Logical review - consolidated</p>
<p>Step done by (Responsible)</p>	<p>System architect</p> <p>Subsystem architect</p>

Provides input to /assists (Contributes)	System engineer
Uses outputs (Informed)	None directly.

ARCH.096 Allocate logical functions to logical components (including alternative allocations)

[RCAMT-390](#) - Add mandatory rationales to ARCH.088, 182 and other steps that require functions to justify their inputs.

FINISHED

Goal	Define the responsibilities (functional scope) of logical components
Requirements met by this process step	ISO 15288 6.4.4.3 c) 3, d) EN 50126-1 7.6.1
Inputs	AMOD-125 Logical architecture definition AMOD-081 Logical functional flow definition (single system capability realisation)
Outputs	AMOD-125 Logical architecture definition (per realised capability) AMOD-125 Logical architecture definition (logical component consolidated)
Methodology	<div data-bbox="304 938 1139 1160" style="border: 1px solid #ccc; padding: 10px;"><p>i Note: This activity needs to be carried out after or in parallel with other activities under ARCH.921 and ARCH.922.</p><p>This methodology combines top-down and bottom-up approaches, since it is recognised that ARCH is being introduced onto a project where work has already been done at logical level.</p></div> <p>Top-down</p> <ul style="list-style-type: none">• After completion of the process steps in ARCH.921 Decompose system functionality there exist a set of leaf-level logical functions. This logical functions can be allocated to different logical components.• After completion of the process steps in ARCH.922 Define logical components there exist a set of defined logical components, which do not have allocated logical functions yet.

1. Allocate logical functions to logical components

- Get a logical component and allocate all leaf-level logical functions which are relevant for this logical component.
- At the end of this activity, all leaf-level functions concerned with the achievement of the current realised capability should be allocated to one logical component.
- Where there exists a design choice as to which logical component or actor should do a particular function, two leaf-level functions should be created (go back to the steps in [ARCH.921](#)), and allocate it to the respective component/actor. The design choice is then made in [ARCH.097](#).
- Define a rationale to justify **why** a function is allocated to a logical component (only if it is otherwise not clear how you came to that decision).

2. Identify superfluous logical components

- Logical components without any allocated logical function shall obtain rationale why logical functions cannot be allocated.
- Another iteration of [ARCH.921](#) / [ARCH.922](#) is required in order to delete the superfluous logical component or to create missing logical functions for allocation

3. Identify missing logical components

- Logical functions that cannot be allocated to a logical component shall obtain a rationale that describes the logical function as an orphan function
- Another iteration of [ARCH.921](#) / [ARCH.922](#) is required in order to remove the superfluous logical function or to create a new logical component for allocation

4. Identify duplicate logical functions

- When there are different logical functions that describe the same functionality a merge of similar logical functions should be considered

5. Logical component is consolidated when:

- At least one logical function is allocated to it, which provides an unique functionality.
- The allocated logical function realises at least one system function.
- The logical component has a comprehensive description, which includes rationales for each decision made, further provides information and reasoning about dismissed but possible alternative logical components.

- Where there exists the possibility for a function to be carried out by more than one logical component/actor depending on the configuration of the system or a state, the function should be defined once as a replicable element catalogue (REC) without being allocated, then replicas (RPL) of the function should be created and allocated to the respective logical component/actor. These replicas should then have their availability in states/modes set to the correct configuration or other state.

Rationale: depending on grades of automation, either an actor or a logical component will carry out a particular function. It shouldn't be possible that two logical components do the same function, though - that is ruled out by the splitting rules for logical components.

Note: it is possible that this step changed in the future but currently we need to keep the REC/RPL (or at least some other way) for alternative function allocation; this should begin at system level and go consistently down until we refactor the functions at physical.

Bottom-up

If there exist already logical functions allocated to logical components. These allocations should be validated during the parallel activities in [ARCH.921](#) and then visualised on the output view ([AMOD-125](#)).

Tools and non-human resources	Team for Capella
Cardinality	Once per realised system capability, with revisions allowed
Completion criteria	The output view conforms to its modelling rules. All logical functions associated with the current capability have been allocated to the appropriate actor or logical component
Design review	ARCH.R.5 Logical feature review ARCH.R.6 Logical review - consolidated
Step done by (Responsible)	<ul style="list-style-type: none"> • System architect
Provides input to /assists (Contributes)	<ul style="list-style-type: none"> • Subsystem architect • Engineer • Cross-cutting engineer
Uses outputs (Informed)	RAMS manager RAMS engineer Verification & validation manager

ARCH.086 Create or update exchange scenarios at logical level

Goal	Visualise interactions between and within logical components to reveal potential weaknesses on the logical architecture.
Requirements met by this process step	ISO 15288 6.4.4.3 c) 4
Inputs	<p>AMOD-125 Logical architecture definition</p> <p>AMOD-084 Logical exchange scenario definition (where one already exists for this capability, or is defined externally for a portion of the capability by an external group)</p> <p>AMOD-082 Logical functional chain definition</p>
Outputs	AMOD-084 Logical exchange scenario definition
Methodology	<p>This process step is one of the last steps on logical level architecture. The task is to generate an Exchange Scenario diagram. The Exchange Scenario diagram illustrates the complete set of functional exchanges that are required to achieve the end conditions of a realised system capability.</p> <p>In this step no model elements shall be created or added. The Exchange Scenario diagram only displays model elements that already exist. The process of creating the Exchange Scenario diagram is intended to reveal missing or superfluous functional exchanges or weaknesses in decomposition. When during this step any issues are discovered, another iteration of the previous process steps (ARCH.921 , ARCH.922, ARCH.096) might be required.</p> <p>Rules for the Exchange Scenario diagram see modelling rules.</p> <ul style="list-style-type: none"> • The logical components/actors serve as lifelines in the Exchange Scenario diagram. As a consequence, the functional exchanges in the Exchange Scenario diagram are displayed between logical components/actors. • The number of internal exchanges between function pairs shall be kept to a minimum. Internal exchanges are functional exchanges between functions that are both allocated to the same logical component or actor. <p>It is possible that functional exchanges have different start/end components based on their configurations. An example would be the Grade of Automation (GoA). In such a case the system capability may have to be modeled as a set of variants. The variants then have different scope definitions for the system functionality. If the set of variants is not an option, then such functional exchanges should be modeled within the usual logical fragments (ALT, PAR etc.).</p>
Tools and non-human resources	Team for Capella
Cardinality	Zero to n times per realised system capability (as many different sequences should be created as are judged necessary to gain confidence in the design)

Completion criteria	<p>The output view conforms to its modelling rules</p> <p>The content of the output view represents the team's current understanding of the way this realised system capability is intended to be delivered.</p>
Design review	<p>ARCH.R.5 Logical capability review</p> <p>ARCH.R.6 Logical review - consolidated</p>
Step done by (Responsible)	<ul style="list-style-type: none"> • System architect
Provides input to /assists (Contributes)	<p>Inclusive OR:</p> <ul style="list-style-type: none"> • Subsystem architect • Engineer
Uses outputs (Informed)	<p>None identified</p>

ARCH.110 Consolidate traceability between model elements at logical level and model elements at system level

[RCAMT-267](#) - Populate Confluence page for activity definition: ARCH.110 Consolidate traceability between model elements at logical level and model elements at system level **FINISHED**

Goal	Traceability between logical NFRs and system NFRs established.
Requirements met by this process step	None defined
Inputs	None.
Outputs	AMOD-126 Logical-system traceability report
Methodology	<p>Identify a NFR into the logical architecture layer:</p> <ol style="list-style-type: none"> 1. Check if there is already a generic trace relationship to a system level NFR. If that is the case check if traceability is reasonable and correct. 2. To establish a new traceability between NFRs proceed as follows: <ol style="list-style-type: none"> a. Right click on the model element into the capella explorer (property value on logical level) --> wizards --> Trace manager b. add now a new outgoing generic trace relationship from the logical level NFR to the system level NFR 3. Repeat the above two steps until all logical level NFRs have an outgoing generic trace relationship to at least one system level NFR.
Tools and non-human resources	Capella 4 Teams
Cardinality	
Completion criteria	Each logical level model element has a trace relationship to at least one system level model element.
Design review	ARCH.R.6 Logical review - consolidated
Step done by (Responsible)	System Architect
Provides input to/assists (Contributes)	<ul style="list-style-type: none"> • Engineer • Cross-cutting engineer • Subsystem Architect
Uses outputs (Informed)	None.

ARCH.190 Consolidate logical data

[RCAMT-383](#) - Populate Confluence page for process step ARCH.190 Consolidate logical data FINISHED

Goal	Avoid duplications, ensure re-usability and consistency of modelled logical data.
Requirements met by this process step	None defined
Inputs	AMOD-091 Logical data objects [L.CDB] AMOD-092 Logical exchange items [L.CDB]
Outputs	AMOD-091 Logical data objects [L.CDB] AMOD-092 Logical exchange items [L.CDB]
Methodology	<p>This process step involves the review and, if required, the update of the existing viewpoints AMOD-091 and AMOD-092 with focus on the following aspects:</p> <ul style="list-style-type: none">• Re-usability and consistency: Are all common data objects re-used as appropriate or are there potentially two or more specific data objects defined, which are very similar in scope? In this case it should be considered to create common data object.• Completeness:<ul style="list-style-type: none">• Logical exchange items: Are all logical exchange items actually linked to a logical data item?• Functional exchange allocation: Are all logical exchange items allocated to functional exchanges?• Viewpoints: Is there at least one instance of AMOD-091 per instance of AMOD-025?• Readability:<ul style="list-style-type: none">• Do the instances of AMOD-091 provide good readability to convey the full definition of the respective logical data?• Is there also an overview of all logical data available as working diagram? This helps to ensure re-usability, consistency and completeness
Tools and non-human resources	Team for Capella
Cardinality	At least once per capability package prior to review ARCH.R.5.
Completion criteria	<p>The output views conform to their modelling rules.</p> <p>The exchange item definition and logical data definition have been judged to be consistently defined (with respect to completeness, re-usability).</p> <p>All functional exchanges on logical level have at least one exchange item allocated.</p>
Design review	ARCH.R.5 Logical capability review
Step done by (Responsible)	System architect
Provides input to/assists (Contributes)	System engineer

**Uses outputs
(Informed)**

None directly.

ARCH.099 Define behaviour for logical functions

[RCAMT-418](#) - Populate Confluence page for process step ARCH.099 Define behaviour for logical functions FINISHED

Goal	Define the behaviour of a logical function
Requirements met by this process step	none
Inputs	AMOD-081 Logical functional flow definition (single system capability realisation) AMOD-082 Logical functional chain definition
Outputs	AMOD-083 State-based behaviour definition (logical function) AMOD-081 Logical functional flow definition (single system capability realisation) (updated)
Methodology	<p>Introduction</p> <p>The scope of this process step is to create a behavioural definition for a logical function. This means to define how a function transforms its inputs to its outputs. It consists of a precise definition how data objects of exchange items (inputs and outputs through functional exchanges) are produced and consumed via the function ports.</p> <hr/> <p>Preparation step</p> <p><i>Before performing the three main steps of this process step try to describe what the logical function actually should do in an informal way. This can be done for example with natural language, drawings, tables or simple sketches. The goal here is to focus once more on content rather than formalising it directly. After that when there is an overall consistent picture of the logical function existing move on then with the first step.</i></p> <p>First step</p> <p>Consider the logical function and characterise its possible behaviour.</p> <p>The goal is to find out which behavioural model fits the best or is most suitable for the logical function. The following aspects shall be taken into account:</p> <ul style="list-style-type: none">• incoming exchange items and their data objects• outgoing exchange items and their data objects• purpose• informal description (from the preparation step)• relation to other surrounding functions <p>Decide then which kind of behavioural definition shall be created for the considered logical function. The behavioural definition is based on a behavioural model to be chosen. The following (non-exhaustive) options are proposed consisting of different mathematical and abstract computational models:</p>

recommended options for behavioural models

- deterministic finite state machine
- mathematical function (e.g. transfer function)
- constraint (i.e. single boolean expression)
- lookup table (with deterministic conditions)
- truth table

Note: Where it is not possible or practical to model the behaviour with the aforementioned options, another deterministic behavioural model shall be identified and selected.

Second step

Consider the chosen behavioural model of the logical function.

The goal is to find out which notation fits the best or is most suitable for the behavioural model.

Decide which kind of notation (i.e. modelling language) shall be used for the behavioural model. The following options are proposed consisting of different notations to cover different behavioural models:

recommended options for modelling languages

- Capella 5 state machine diagram
 - applicable and usable for state machines including extended concepts like hierarchical states
 - extended by an additional modelling language with textual notation to express precisely the conditions of change events, guard conditions and actions in states and transitions
- SysML v2 textual notation
 - applicable and usable for all behaviours
 - e.g. actions, states and constraints may be used (with action language capabilities build-in natively)
- Action language
 - applicable and usable for behaviours expressed more abstract with a focus on execution of these
 - textual notation of model elements are on a more abstract level than programming languages - means focussing more on what and less on how
 - e.g. Action Language for Foundational UML (ALF) related to UML and SysML
- Declarative language
 - applicable and usable to express constraints on model elements (e.g. computational logic) rather than providing abstract implementations of these constraints
 - textual notation of model elements are on a more abstract level than programming languages - means focussing more on what and less on how
 - e.g. Object Constraint Language (OCL) related to UML

Note: Where it is not possible or practical to model the behaviour with the aforementioned options, another notation shall be identified and selected.

Third step

Create the content of behaviour definition based on the chosen behavioural model and notation.

Consider each functional exchange which provides incoming exchange items via the function's input ports.

1. For each exchange item identify each contained data object (exchange item element).
2. For each data object identify the possible values it can have, including the regular values and values which represent absent or invalid inputs.

Consider each functional exchange which provides outgoing exchange items via function's output ports.

1. For each exchange item identify each contained data object (exchange item element).
2. For each data object identify the possible values it can have, including the regular values and values which represent absent or invalid outputs.

Define the behaviour.

1. Express under which circumstances values of incoming data objects lead to values of outgoing data objects.
2. Express this based on the chosen behavioural model and notation.

Final step

Depending on the chosen behavioural model and notation finalise or update the corresponding viewpoints.

- for behaviour definitions of kind deterministic finite state machines create [AMOD-083 State-based behaviour definition \(logical function\)](#)
- for every other kind of behaviour definition update [AMOD-081 Logical functional flow definition \(single system capability realisation\)](#) with attaching the behaviour definition to the logical function (e.g. with a constraint element containing then the notated behaviour)

Guidance

Please consider the following pages:

- [Technical note on definition of functions](#)
- [Concepts for behaviour definitions on logical level](#)

The goal of behaviour definitions is to focus on a formal definitions of *what* should be done rather than *how* it should be done. This means in general that with every behavioural model and modelling language chosen, try to express more a declarative behaviour definition rather than an abstract or prototypical implementation of it.

A simple example:

- A behaviour definition should not include some concrete or arbitrary sorting algorithm, but instead a formal definition of a constraint that the outgoing list of items is sorted from small to big.

In those cases the function's behaviour is defined with rules that logically relate the outputs to the inputs. Instead of giving an implementation or strategy which would look like code for computing the outputs based on the inputs, it provides rules for checking that the function provides correct outputs based on the inputs. A prototypical implementation would be done by e. g. simulation following the behaviour definition.

Tools and non-human resources	<ul style="list-style-type: none">• Teams for Capella• or an additional external tool in which the viewpoint is created and then transferred to Teams for Capella
Cardinality	once for each logical function
Completion criteria	Behaviour is fully defined including the consideration of all exchange items and their exchange item elements.
Design review	ARCH.R.6 Logical review - consolidated

Step done by (Responsible)	System architect
Completion criteria evaluated by (Accountable)	Behaviour of the function is completely specified.
Provides input to /assists (Contributes)	Inclusive OR: <ul style="list-style-type: none"> • Engineer • Subsystem architect
Reviews (Contributes)	ARCH.R.6 Logical review - consolidated
Uses outputs (Informed)	none

ARCH.930 Define the physical architecture

- [ARCH.931 Transfer logical level requirements to physical level](#)
 - [ARCH.111 Execute automatic transition of logical elements to physical level](#)
- [ARCH.932 Define the subsystem boundaries](#)
 - [ARCH.090 Identify alternative subsystem boundary options](#)
 - [ARCH.097 Evaluate subsystem boundary options against the architectural tradeoff criteria](#)
 - [ARCH.098 Deploy logical components to subsystems](#)
 - [ARCH.116 Define the subsystems](#)
 - [ARCH.117 Adopt subsystem definition from collaborative project](#)
- [ARCH.933 Define the subsystem interfaces](#)
 - [ARCH.156 Define single inter-subsystem interface](#)
 - [ARCH.151 Refine system interface to subsystem-actor interface](#)
 - [ARCH.114 Define single interface layer](#)
 - [ARCH.128 Define functions of single interface layer](#)
 - [ARCH.129 Define data models for interface layers](#)
 - [ARCH.130 Define interface layer scenarios](#)
 - [ARCH.122 Define behaviour for interface layer functions](#)
 - [ARCH.186 Consolidate single subsystem interface](#)
- [ARCH.934 Define supporting physical architecture](#)
 - [ARCH.125 Define structure of communication assets](#)
 - [ARCH.126 Define structure of computation assets](#)
 - [ARCH.127 Define location kinds](#)
- [ARCH.935 Assess risks associated with physical architecture](#)
 - [ARCH.101 Carry out failure modes, effects and criticality analysis \(FMECA\) on physical functional chains](#)
 - [ARCH.103 Identify new physical functions needed for robustness to failures](#)
 - [ARCH.104 Identify new non-functional requirements needed for robustness to failures](#)
 - [ARCH.105 Identify new non-functional requirements needed to mitigate hazards introduced by physical architecture choice](#)
 - [ARCH.106 Identify new physical functions needed for mitigating hazards introduced by physical architecture choice](#)
 - [ARCH.135 Carry out HAZID on tenderable elements](#)
- [ARCH.936 Allocation/Derivation of NFRs](#)
 - [ARCH.131 Derive & allocate NFRs to location kinds](#)
 - [ARCH.132 Derive & allocate NFRs to communication assets](#)
 - [ARCH.133 Derive & allocate NFRs to computation assets](#)
 - [ARCH.137 Allocate interface NFRs to interface layers/interface functions](#)
 - [ARCH.155 Allocate hazard mitigation NFRs to subsystems](#)
- [ARCH.938 Consolidate the overall architecture](#)
 - [ARCH.192 Refine physical capability realisations](#)
 - [ARCH.176 Create subsystem exchange scenarios](#)
 - [ARCH.136 Push back chosen physical architecture from child models to parent model](#)
 - [ARCH.169 Consolidate traceability between model elements at physical level and model elements at logical level](#)
 - [ARCH.168 Consolidate the overall generic physical architecture](#)
 - [ARCH.100 Obtain stakeholder acceptance of preferred architecture](#)

ARCH.931 Transfer logical level requirements to physical level

- [ARCH.111 Execute automatic transition of logical elements to physical level](#)

ARCH.111 Execute automatic transition of logical elements to physical level

Goal	Provide the top-down inputs needed to develop the physical architecture model in alignment with the logical level.
Requirements met by this process step	None defined
Inputs	Model elements on logical level
Outputs	Model elements on physical level
Methodology	<p>For the set of capabilities currently being considered, initiate the automatic transition of the logical capability realisation.</p> <p>The capella tool uses model layers to separate concerns so that the engineering process can be carried out in stages. Here is a diagram illustrating the automatic transition process:</p> <p>The transitions needed are as follows:</p> <ul style="list-style-type: none"> • Logical capability realisations to be automatically transferred to physical level as physical capability realisations. • Logical actors to be automatically transferred to physical level as physical actors. • Logical actor functions to be automatically transferred to physical level as physical actor functions allocated to the • Logical components, component exchanges, logical functions and functional exchange to be automatically transferred • Functional chains to be automatically transferred to physical level. <p>Important note: Where elements already exist at physical level, it may be more appropriate to create a trace from the logical level that there is nothing appropriate at physical level that can be directly traced (via the "Realised <element type>" property). Wherever this is done, ensure that any associated property values of a higher-level</p>
Tools and non-human resources	Team for Capella
Cardinality	Carried out as required to maintain synchronisation between logical and physical level of the model; as a minimum once per realised capability, at the point when the responsible team passes review ARCH.R.4 and moves
Completion criteria	The set of physical capability realisations in the model includes all capabilities to be delivered within the set under consideration
Design review	No design review checks this since it is an intermediate step, but the work should be coordinated and checked by the responsible team
Step done by (Responsible)	Engineer

Provides input to /assists (Contributes)	Inclusive OR: <ul style="list-style-type: none">• Engine room• System architect• Cross-cutting engineer
Uses outputs (Informed)	None.

ARCH.932 Define the subsystem boundaries

- [ARCH.090 Identify alternative subsystem boundary options](#)
- [ARCH.097 Evaluate subsystem boundary options against the architectural tradeoff criteria](#)
- [ARCH.098 Deploy logical components to subsystems](#)
- [ARCH.116 Define the subsystems](#)
- [ARCH.117 Adopt subsystem definition from collaborative project](#)

ARCH.090 Identify alternative subsystem boundary options

Goal	Definition of subsystem boundary options on logical components patterns to allow later selection of substantiated subsystem boundaries.
Requirements met by this process step	ISO 15288 6.4.4.3 c) 2 ISO 15288 6.4.4.3 c) 4 EN 50126-1 7.6.1
Inputs	Logical components and allocated logical functions transferred to physical level AMOD-125 Logical architecture definition
Outputs	AMOD-080 Subsystem boundary option sketch
Methodology	<p>The logical components produced at logical level (in accordance with unique combination of splitting principles) must be hosted by subsystems to ensure that its allocated functionality will be realized on physical level.</p> <p>Hence in this activity candidate subsystems shall be sketched that allow the hosting of defined logical components.</p> <p>In case subsystems are to be adopted from collaborative projects, this is done in ARCH.117. These adopted subsystem shall also be defined among the candidate subsystem to allow subsequent selection of subsystem boundaries and, if needed, re-iterate the definition those adopted subsystem within the collaborative project.</p> <p>In case of uncertainties regarding the definition of candidate subsystems, sketch the options showing each alternative of the candidate subsystems.</p> <p>This step only needs to be done for subsystems/logical components that are not included in subsystems defined by collaborative projects.</p>
Tools and non-human resources	Any drawing tool (or simply pen and paper)
Cardinality	Once with iterations permitted.
Completion criteria	<p>Candidate subsystem boundaries have been sketched.</p> <p>Alternative sets of subsystem boundary options have been defined, where reasoning for having alternative candidate subsystems is documented.</p>
Design review	Review that completion criteria are satisfied.
Step done by (Responsible)	System architect

Provides input to /assists (Contributes)	Inclusive OR: <ul style="list-style-type: none">• Systems engineer• Subsystems architect
Uses outputs (Informed)	ARCH.097 Evaluate subsystem boundary options against the architectural tradeoff criteria

ARCH.097 Evaluate subsystem boundary options against the architectural tradeoff criteria

[RCAMT-364](#) - Populate Confluence page for activity definition: ARCH.097 Evaluate logical architecture options against the architectural tradeoff criteria **FINISHED**

[RCAMT-328](#) - Move methodology content of ARCH.089 into ARCH.097 and other tradeoff steps **FINISHED**

Goal	Determine, in a traceable and reproducible way, the optimal alternative for each of the choices of subsystem scope that exist.
Requirements met by this process step	ISO 15288 6.4.4.3 e)
Inputs	<p>AMOD-010 Trade space assessment</p> <p>AMOD-116 System implementation constraints</p> <p>AMOD-080 Subsystem boundary option sketch</p>
Outputs	AMOD-124 Physical architecture tradeoff record
Methodology	<p>The outputs of already performed activities on operational and system level shall be analysed to derive the key impacts on architecture candidates. This includes results from market explorations, organisational factors (e.g. strategies, regulatory and legal constraints), business factors (e.g. business concepts of operations) and any other factor that may impact the system suitability at any stage in its lifecycle. These analysis results are found in AMOD-010 and AMOD-116 in general terms.</p> <p>Within this activity the following steps are suggested to be performed concurrently to ARCH.090 Identify alternative subsystem boundary options to ensure that determined factors are applicable to the architecture candidates at hand:</p> <ol style="list-style-type: none"> 1. From input materials define the criteria for assessing the architectural options, and the score weighting (if using) 2. Evaluate each option against each criterion 3. Make a decision on which option to adopt <p>The DAR (Decision Analysis and Resolution) method is recommended and the template for AMOD-124 is constructed on this basis. However, there are several trade off techniques in literature. E.g. you could refer to the CESAM guide (section 7.2) or ATAM. DBS has compiled some basic guidance here: Tradeoff analysis techniques overview.</p> <p>It is not mandatory to select the option that scores highest in the evaluation, if there is a good reason to favour another option; in this case, the rationale must be very clearly stated and it may result in the need to adjust the weighting of the criteria, or introduce new criteria representing the decisive factor.</p> <p>The decision may be made individually by the Lead System Architect or in a more collective group with other involved stakeholders; in this second case, voting systems may be needed in order to score each option in a way that represents the collective opinion of the group.</p>
Tools and non-human resources	Dependent on the tradeoff technique used
Cardinality	Once per physical architecture decision.

Completion criteria	The Lead System Architect is satisfied with the outcome of the decision.
Design review	ARCH.R.7 Physical review - system capability realisation ARCH.R.8 Physical review - consolidated
Step done by (Responsible)	Lead system architect
Provides input to /assists (Contributes)	System architect Subsystem architect Other technical experts as needed
Uses outputs (Informed)	None outside ARCH.

ARCH.098 Deploy logical components to subsystems

[RCAMT-367](#) - Modifications to ARCH.932 regarding subsystem definitions **FINISHED**

[RCAMT-414](#) - Populate Confluence page for activity definition: ARCH.098 Deploy logical components to subsystems **FINISHED**

[RCAMT-390](#) - Add mandatory rationales to ARCH.088, 182 and other steps that require functions to justify their inputs. **FINISHED**

Goal	Deploy logical components derived from logical level to defined subsystems.
Requirements met by this process step	None defined
Inputs	AMOD-093 Subsystems definition AMOD-080 Subsystem boundary option sketch AMOD-124 Physical architecture tradeoff record
Outputs	AMOD-085 Subsystems involved in single realised capability AMOD-093 Subsystems definition (updated)

Methodology

First step

Create the viewpoint [AMOD-085 Subsystems involved in single realised capability](#) for the realised capability of interest. A selection of subsystems shall be modeled there including the deployed logical components, their logical exchanges to the involved physical actors and the allocated functions.

1. Select a subset of the defined subsystems which are involved in a single realised capability of interest.
2. Select the physical actors which are involved in this single system capability.
3. Arrange the subset of the subsystems and the involved physical actors on the diagram.
4. Deploy on [AMOD-080](#) and [AMOD-124](#).
5. Define a rationale to justify the deployment of a logical component to a subsystem (only if it is otherwise not clear how you came to that decision).
6. Arrange the logical component exchanges of the logical components between two subsystems and between a subsystem and a physical actor.
7. Show all functions which are allocated to a deployed logical component of a subsystem.
8. Show all functions which are allocated to a physical actor and which have a functional exchange connected to a function allocated to at least one of the before deployed logical components.
9. Arrange the functions and functional exchanges.

Note: it is possible to copy the layout (functionality which is provided by Capella) from the logical level [AMOD-125](#) because this viewpoint only added the subsystem. The rest of the content should be the same which is shown on [AMOD-125](#) for a single realised capability.

Second step

Update [AMOD-093 Subsystems definition](#). All subsystems shall be modelled there concretely in an overall picture including the deployed logical components and their logical exchanges to physical actors.

1. Show all the deployed logical components to the subsystems.
2. Show all physical actors which will be linked to a subsystem.
3. Show all logical component exchanges of logical components between two subsystems and between a subsystem and a physical actor.

Note: Functions are not shown in [AMOD-093 Subsystems definition](#).

Guidance: Physical links between two subsystems may not be modelled always. A physical link doesn't always exist between two subsystems directly but rather could go into for example a network port of the communications platform. This will be added in later steps of the process.

Tools and non-human resources	Team for Capella
Cardinality	Once per realised capability.
Completion criteria	All logical components which are of interest for a single capability are deployed to the correct subsystem.
Design review	ARCH.R.7 Physical review - system capability realisation
Step done by (Responsible)	Systems architect
Provides input to /assists (Contributes)	Inclusive OR: <ul style="list-style-type: none"> • Engineer • Subsystems architect
Uses outputs (Informed)	None.

ARCH.116 Define the subsystems

[RCAMT-416](#) - Update ARCH.116 and AMOD-093: Subsystems definition **FINISHED**

[RCAMT-390](#) - Add mandatory rationales to ARCH.088, 182 and other steps that require functions to justify their inputs. **FINISHED**

Goal	Definition of all subsystems with their physical boundaries in an overall picture.
Requirements met by this process step	ISO 15288 6.4.4.3 c) ISO 15288 6.4.4.3 d) EN 50126-1 6.5.3
Inputs	AMOD-080 Subsystem boundary option sketch AMOD-124 Physical architecture tradeoff record
Outputs	AMOD-093 Subsystems definition
Methodology	<p>Main steps</p> <ol style="list-style-type: none"> 1. Define and model each subsystem based on the inputs of this process step. 2. Combine these new defined subsystems with those created by ARCH.117 Adopt subsystem definition from collaborative project on the new created viewpoint. 3. Justify each subsystem with a rationale. <p>The outcome of this is the initial creation of AMOD-093 Subsystems definition. All subsystems together shall be depicted there concretely in an overall picture.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once to model subsystems with selected subsystem boundaries.
Completion criteria	All intended subsystems are defined.
Design review	ARCH.R.8 Physical review - consolidated
Step done by (Responsible)	Systems architect
Provides input to/assists (Contributes)	Inclusive OR: <ul style="list-style-type: none"> • Engineer • Subsystems architect
Uses outputs (Informed)	None.

ARCH.117 Adopt subsystem definition from collaborative project

[SM-2851](#) - Populate Confluence page for activity definition: ARCH.117 Adopt subsystem definition from collaborative project FINISHED

Goal	Identification and adoption of relevant Subsystems from collaboration projects.
Requirements met by this process step	None
Inputs	Defined Subsystems in collaboration projects (e.g. EULYNX, S2R or RCA)
Outputs	AMOD-093 Subsystems definition
Methodology	Analyse and compare the provided content from the collaboration project with the project`s subsystem candidates defined in ARCH.116 Define the subsystems . Outcome of the analysis may result in: <ul style="list-style-type: none">• Going back to the collaborative project and discuss potential need to modifying the subsystems comin from that project.• Updating the defined subsystems in AMOD-125 to adapt to subsystems from collaborative project.
Tools and non-human resources	Capella for Teams
Cardinality	Once per collaboration project, with permitted revisions
Completion criteria	All subsystems are identified and adopted and implemented into the model. The output view conforms to its modelling rules.
Design review	Completion criteria agreed between lead system architect, system and subsystem architects.
Step done by (Responsible)	Lead System Architect
Provides input to/assists (Contributes)	<ul style="list-style-type: none">• System architect• Subsystem architect• Engineer
Uses outputs (Informed)	Responsible person of the collaboration project

ARCH.933 Define the subsystem interfaces

- [ARCH.156 Define single inter-subsystem interface](#)
- [ARCH.151 Refine system interface to subsystem-actor interface](#)
- [ARCH.114 Define single interface layer](#)
- [ARCH.128 Define functions of single interface layer](#)
- [ARCH.129 Define data models for interface layers](#)
- [ARCH.130 Define interface layer scenarios](#)
- [ARCH.122 Define behaviour for interface layer functions](#)
- [ARCH.186 Consolidate single subsystem interface](#)

ARCH.156 Define single inter-subsystem interface

ARCH.151 Refine system interface to subsystem-actor interface

[RCAMT-517](#) - Populate Confluence page for activity definition: ARCH.151 Refine system interface to subsystem-actor interface

READY FOR REVIEW

Goal	Refine a system interface to a subsystem-actor interface
Requirements met by this process step	None defined
Inputs	AMOD-093 Subsystems definition AMOD-085 Subsystems involved in single realised capability AMOD-100 Interface layer definition
Outputs	AMOD-111 Subsystem-actor interface definition

Methodology	<p>Introduction</p> <p>The subsystem interfaces are the actual interfaces which can be identified and exist between a subsystem and an external system / actor. This process step covers the refinement of those interfaces defined in previous process areas (e.g. ARCH.910 Determine the system requirements and ARCH.920 Define the logical architecture). Subsystem interfaces consist of additional elements inside and between a subsystem and an external system / actor to enable technology-dependent interaction. This means that subsystem interfaces are highly influenced and impacted by the used technology.</p> <hr/> <p>First step: analyse the subsystem interface</p> <p>Analyse a pair of a subsystem and external system / actor provided by AMOD-085 and AMOD-093. A candidate for an subsystem interface is indicated from the situation that are existing logical exchanges between logical components deployed to the subsystem and the external system / actor crossing those boundary of those systems.</p> <p>Decide then which kind of interface is intended and needed. This shall take into account design decisions made in ARCH.934 for the physical architecture as well as the existing content provided by AMOD-085 and AMOD-093 which can constrain the possibilities.</p> <div data-bbox="256 573 636 833" style="border: 1px solid black; padding: 5px;"> <p>possible categories of subsystem interfaces</p> <ul style="list-style-type: none"> • network communication interface <ul style="list-style-type: none"> • e.g. protocol stacks • human-machine interface <ul style="list-style-type: none"> • e.g. user interfaces • hardware interface <ul style="list-style-type: none"> • e.g. 4-wire point machine interface • e.g. power supply interface • software interface <ul style="list-style-type: none"> • e.g. application programming interface </div> <p><i>Note: Where it is not possible or practical to model the interface with the aforementioned categories, another one shall be identified and selected.</i></p> <p>Second step: define the subsystem interface</p> <p>This process step as well as the corresponding process steps ARCH.114 Define single interface layer in parallel. This is required to define the subsystem interface in the whole showing the interconnections of all interface layers as well as to focus on details of each interface layer itself.</p> <table border="1" data-bbox="256 1039 1206 1312"> <thead> <tr> <th>process step</th> <th>actions</th> </tr> </thead> <tbody> <tr> <td>this</td> <td> create AMOD-111 Subsystem-actor interface definition for the subsystem interface as a whole <ul style="list-style-type: none"> • define intra-subsystem exchanges between the interface layer components of different interface layers • define intra-subsystem exchanges between the interface layer components and the existing logical components </td> </tr> <tr> <td>ARCH.114 Define single interface layer</td> <td> create AMOD-100 Interface layer definition for each interface layer <ul style="list-style-type: none"> • define interface layer components of the interface layer for each subsystem </td> </tr> </tbody> </table> <p>Guidance</p> <ul style="list-style-type: none"> • Pattern for defining interfaces <p>The content, architectural design decisions and technological choices can lead to different kinds of interface layers of the considered inter-subsystem interface. This includes possible investigation of existing interface layers specifications (e.g. TCP/IP, RaSTA, UDP for network communication). A trade-off analysis could be performed here independently from the ARCH process to work out different options.</p>	process step	actions	this	create AMOD-111 Subsystem-actor interface definition for the subsystem interface as a whole <ul style="list-style-type: none"> • define intra-subsystem exchanges between the interface layer components of different interface layers • define intra-subsystem exchanges between the interface layer components and the existing logical components 	ARCH.114 Define single interface layer	create AMOD-100 Interface layer definition for each interface layer <ul style="list-style-type: none"> • define interface layer components of the interface layer for each subsystem
process step	actions						
this	create AMOD-111 Subsystem-actor interface definition for the subsystem interface as a whole <ul style="list-style-type: none"> • define intra-subsystem exchanges between the interface layer components of different interface layers • define intra-subsystem exchanges between the interface layer components and the existing logical components 						
ARCH.114 Define single interface layer	create AMOD-100 Interface layer definition for each interface layer <ul style="list-style-type: none"> • define interface layer components of the interface layer for each subsystem 						
Tools and non-human resources	Team for Capella						
Cardinality	Once per every identified pair of a subsystem and an external system / actor.						
Completion criteria	The output views conform to their modelling rules.						
Design review	ARCH.R.7 Physical review - system capability realisation ARCH.R.8 Physical review - consolidated						
Step done by (Responsible)	<ul style="list-style-type: none"> • subsystem architect 						
Provides input to /assists (Contributes)	<ul style="list-style-type: none"> • subsystem architect for communication/computation • system architect • engineer 						

**Uses outputs
(Informed)**

none

ARCH.114 Define single interface layer

[RCAMT-335](#) - Populate Confluence page for activity definition: ARCH.114 Define reusable interface layers

FINISHED

Goal	Define reusable interface layers and their relationships independent of the usage in a specific physical interface.
Requirements met by this process step	None defined / TBD
Inputs	AMOD-081 Logical functional flow definition (single system capability realisation) AMOD-082 Logical functional chain definition AMOD-125 Logical architecture definition AMOD-093 Subsystems definition
Outputs	AMOD-100 Interface layer definition
Methodology	<p>General information</p> <p>This process step describes the necessary tasks and actions to create <u>reusable</u> interface layers for the physical interfaces. These reusable interface layers are probably imposed on the project. They are complementing the one-off interface layers which are identified by ARCH.122 Define state-based behaviour specification for interface layer functions and further developed in ARCH.115 Define one-off interface layers. The definition of the reusable interface layers is done independent of their usage in a subsystem or actor. Allocation of the layers will be done later in ARCH.186 Finalise single inter-subsystem interface.</p> <p>Background information</p> <p>Physical interfaces are the actual interfaces which can be identified and exist between physical entities. Before these physical interfaces can be specified in full interface layers are identified, defined and specified. From an architectural and methodological perspective interface layers are very useful to manage complexity applying structuring and modularisation. Interfaces can be composed of one or more interface layers handling a specific concern which includes for example exchanged information and behaviour. Physical interfaces and their interface layers are highly influenced and impacted by the used technology.</p> <p>Underlying methodology</p> <p>The methodology behind this is the interface-centric approach, which provides distinct views on an interface between two physical entities. It is based content-wise on the port-centric refactoring of the logical components (and their behaviours) delivered by AMOD-093 Subsystems definition. The approach considers interfaces and interface layers always consisting of two sides interfacing with each other. This means an interface and also an interface layers consist of two interface layer components linked together. Information is exchanged between them over a link.</p> <p>How to do this task</p>

- do every step for ARCH.115 first
- then do necessary actions in the model (technically) and... to classify every reusable interface layer
- Decide on an appropriate set of interface layers for the interface you are modelling;
- The application layer of the interface should already be represented by the functions and functional exchanges that interact across the system boundary in the existing scenarios used as input to this activity

Task regarding new content

The following steps are intended to ...

1. Decide which technology to be used.
 - a. Depending on the content, architectural design decisions and technological choices there can be different kinds of physical interfaces and their interface layers.
 - computer network communication with protocol stacks and protocol layers, where the protocol layers correspond to interface layers (the reference model for OSI is available [here](#))
 - human-machine communication with an user interface
 - electrical interface
 - mechanical interface
 - hardware interface
 - software interface
 - (further types)
 - b. The responsible person should familiarise themselves with the technology-dependent layer model of interfaces. This is used throughout our processes as a basis for identifying layers of interfaces concerned with different tasks.
 - c. T
2. Based on patterns, constrains, existing specifications, models of the used technology create new technology-dependent interface layers complementing the one(s) which were already provided by AMOD-093.
 - a. Create interface layer component for one side
 - b. Create interface layer component for the other side.
 - c. Create a component exchange between the two new created interface layer component.
 - d. Repeat a,b,c until all required interface layers are created.
 - e. Create component exchange between two layers on the each side.

For the purpose of DBS/RCA, the following interpretations of OSI are made:

- application layer - information content that needs to be transferred in order to achieve the system capabilities
- presentation layer - data formatting and type conversion, packet structuring, bit field formats
- session layer - message sequencing & handshaking
- transport layer - connection setup, monitoring and shutdown
- network layer - addressing and routing
- data link layer - formats for physical transmission, byte/bit order, parities, checksums...
- physical layer - physical configurations, voltages/energy levels, polarities, frequencies, pinouts etc.

	<p>These layers are a guideline and can be adapted and tailored as necessary for a specific interface. Sometimes not all layers are needed (for example, a discrete digital relay interface could be adequately represented by application and physical layers only), where other interfaces define extra layers not explicitly included in OSI, most commonly a "safety" layer for error-detection and correction to prevent incorrect data being used in safety critical applications.</p> <p>Where an external standard/specification dictates the nature of the interface, the interface to the system of interest shall exactly reflect the implementation of that standard.</p>
Tools and non-human resources	Team for Capella
Cardinality	Once per every identified pair of two adjacent subsystems.
Completion criteria	The output views conform to their modelling rules.
Design review	ARCH.R.x (TBD)
Step done by (Responsible)	<ul style="list-style-type: none"> • System architect • Subsystem architect • Engineer
Completion criteria evaluated by (Accountable)	

ARCH.128 Define functions of single interface layer

[RCAMT-332](#) - Populate Confluence page for activity definition: ARCH.128 Define functions of single interface layer

READY FOR REVIEW

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	

ARCH.129 Define data models for interface layers

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	

ARCH.130 Define interface layer scenarios

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	

ARCH.122 Define behaviour for interface layer functions

[RCAMT-420](#) - Update and define ARCH.122 Define behaviour for interface layer functions

READY FOR REVIEW

Goal	Deployment of components and functions from logical level to subsystems. Re-factoring of these components and functions that provide objects/things to other subsystems such that these components and functions are mirrored in the consuming subsystems. Definition behaviour at each end of a subsystem interface via state machines.
Requirements met by this process step	None defined
Inputs	AMOD-093 Subsystems definition other...
Outputs	AMOD-144 State-based behaviour definition (interface layer function)
Methodology	<p>Summary of methodology</p> <p>To ensure compliance with modelling standards (which propose an interface centric approach) the first part of this activity comprises functional and structural refactoring, i.e.</p> <ul style="list-style-type: none">• the logical components (including their allocated functions) that have been transferred from logical level shall be deployed to selected subsystems;• the functions shall be decomposed into master and mimic functionality and• new logical components shall be created, such that all master and mimic functions are allocated to logical components. <p>The second part of this activity comprises the modelling of state machines for each logical component to define the behaviour of each logical component.</p> <p>Table of content:</p> <h2>1. Deployment of logical components to subsystems:</h2> <p>The main goal of this activity is to deploy the logical components and their allocated functions that have been transferred from logical level to physical level. Having in mind the subsequent decomposition pattern of those logical components into a mimic and a master component, the existing logical components coming from logical level will be re-used as components that have the master functionality allocated. In other words the logical components (including allocated functions) coming from logical level will be re-used as "master" components by adhering to the following process steps:</p> <h3>1.1. Process steps</h3>

- Deploy each logical component to one subsystem. The deployed logical components shall result in a meaningful distribution of functionality among subsystems.
- Rename the logical component to relate it to the hosting subsystem and to its operated object that it handles (the latter should be already captured by the name used on logical level).
 - E.g. DPS plan executor in PE, if the logical component DPS plan executor is deployed to the Subsystem PE}
- Rename the functions allocated to the logical component to add "(master)" and the end of the function name.

1.2. Guidance for process steps

Consider the following guidance material to help in the deployment of logical components, but at the same time keep in mind that there is no boiler plate that suits all kind deployments:

Guidance 1: Allocate each state type control function to a separate subsystem

- State type function "Control the drivability state of one DPS" and the related logical component may be deployed to one subsystem called "Safety Logic".
- State type function "Control the position state of one point" and the related logical component may be deployed to the subsystem called "Fixed Object Transactor"
- State type function "Control the drive state of one point machine" and the related logical component may be deployed to the subsystem called "Object Controller Point"

Guidance 2: Allocate each state type observe/estimate function to the subsystem where the more specific state type is controlled (This aids the subsequent application of the control loop pattern):

- State type function "Estimate the drivability state of one DPS" and the related logical component may be allocated to the subsystem that executes the control function "Control the position state of one point"
- State type function "Estimate the position state of one point" and the related logical component may be allocated to the same subsystem that executes the control function "Control the drive state of one point machine"

Guidance 3: Exceptions

- Exchange between subsystems and actors: Example the function "Control the state of the railway" hosted by the actor "TMS-PAS" provides the operational plan and requires the function "Control the execution state of one operational plan" inside the subsystem. This is an exception of guidance 2, where state type observe/estimate operations should only be hosted by the subsystems that control the next more specific state type. In this specific case the control function and the observe/estimate functions of the same state type are hosted by the same subsystem ("Control the **execution state of one operational plan**" and "Estimate the **execution state of one operational Plan**"

1.3. Capella example

Examples: In the following you find an example about a possible version of [AMOD-093 Subsystems definition](#) after logical components from logical level have been deployed to subsystems. For the example Capella has been used.

2. Refactor subsystem structure into master and mimic structure

The main goal of this activity is to create the mimic functionality and the related logical component that synchronises a state change of the related master functionality (i.e.: the functionality that has been deployed in the previous step via its logical component):

This is required to realize the interface-centric approach from EULYNX ModStan. This means that each subsystem hosts not only those logical components that are core to its subsystem-owned functionality, as we saw at logical level, but additionally hosts a mimic of any controllers or observers that provide information used by those core logical components. As result, the functionality of subsystems is clustered around the ports (interface ends/sides) as much as possible with the application of the master/mimic pattern. The functionality which cannot be clustered around the interface is then considered as the subsystem-owned functionality. This is then only applicable and unique to a specific subsystem and cannot be re(used) in another subsystem.

This "master" controller (observer) and a "mimic" controller (observer) is very much like a synchronisation relationship. This allows a state change on a subsystem's output-facing logical component to be triggered by state changes or functions allocated to the same subsystem's input-facing logical components.

2.1. Process-steps

To achieve the goal discussed previously, the following process steps should be done for refactoring the logical components and functions:

1. Mimics definition: Where functional exchanges cross boundaries of subsystems the decomposition pattern of controller/observers into master and mimic should be applied (see below). I.e. where a required/observed state type is input to a subsystem:
 - a. A mimic of the providing function and its corresponding component shall be created as part of the subsystem that consumes that input
 - b. The functional exchange that conveys that required/observed state type to the subsystem shall be connecting as input to the new mimic function.
 - c. New functional exchange that conveys the same required/observed state shall be connected between new mimic function and original consumer of that information.
 - i. Please note that these subsystem-internal functional exchanges are placeholder of the primary behavioural component of the subsystem, whose creation may be added as process step to this ARCH at a later moment in time.
 - d. Component exchanges shall be defined such that each logical component that have exchanging functions allocated are connected via a component exchange.
2. The logical components that have the mimic functionality allocated shall be named such that its name refers to the operating state type and the its hosting subsystem.
 - a. E.g. "DPS Plan executor in SL" in case the operating state type is the DPS plan and the hosting subsystem is the subsystem SL.
3. Traceability: When an original logical component and its allocated function have been refactored into mimic and master, it must be ensured that both the master and the mimic function as well as their related components are tracing to the same component and function on logical level.

2.2. Capella example

After the definition of mimics and master the previous Capella example should look as follows:

2.3. Pattern for decomposing controller and observers into masters and mimics

The **applied pattern** for decomposing controllers and observers into masters and mimics is defined as follows.

2.4. Definitions for the decomposition pattern

- **Controllers:** Actuators can only be controlled by a master controller.
 - A master controller of state type A can pass a required state to a master controller of state type B if both master controllers are deployed to the same subsystem.
 - A mimic controller of state type A can pass a required state to a master controller of state type B if both controllers are deployed to the same subsystem
 - A mimic controller of state type A is always deployed to a different subsystem from the subsystem that hosts the master controller of state type A.
 - A mimic controller cannot be connected to another mimic controller.
- **Observers:** Sensors are always connected to a master observer.
 - A master observer of state type A can pass an observed state to a master observer of state type B if both master observers are deployed to the same subsystem.
 - A mimic observer of state type A is always deployed to a different subsystem from the subsystem that hosts the master observer of state type A.
 - A mimic observer can provide an observed state type A to a master observer of state type B, if both these observers are deployed to the same subsystem.
 - A mimic observer cannot be connected to another mimic observer.
- **Operated object:** Defined as kind of state that can change over time. Can be a discrete state (on-off) or a continuous state in the sense of state-space modelling (scalar or vector quantities such as position, speed, acceleration). The "object" here is the object of the phrase that defines the function, for example "sense the rotational speed of one axle" has operated object "axle". Concretely it can mean things:
 - concrete railway assets
 - abstract concepts

2.5. State machines for interface-specific behaviour

After deployment and re-factoring each subsystem hosts at least one logical component per interface that it has with another subsystem/actor. For each logical component a state machine is to be created that defines the behaviour of that component as a sum its functions.

If the master component already came with state machine from logical level, keep it. Create then for mimic component a in first simpler step a 1:1 clone of it. Then change both state machine in a way that they can synchronise their state through chosen synchronisation techniques. This can be for example simple request/response exchanges or a three-way handshake mechanism. The goal of the state machines is to synchronize the states between two subsystems consistently. Especially that the mimic state machine mirrors the state of the master.

If the master component does not already have a state machine, it shall be created, along with its mimic, having in mind the following guidance:-

	<ul style="list-style-type: none"> • Finite-state: Some of the controlled objects can be described using a finite state machine; that means that there is a finite number of possible states that the object can occupy. Where it is possible and meaningful to model the state of an object using a finite state machine, there should exist sufficient functions to control every possible transition defined in the finite state machine. • Continuous (infinite-state): Some controlled objects cannot be directly described using a finite state machine because their state is a continuous variable. For example, the position of a train unit's front end is infinitely variable, and even if we restrict it to a certain level of precision (for example, to the nearest metre), there exist so many possible states that it is impractical to model this using a finite-state machine. In this case we could define a finite-state machine on particular regions or value sets of the continuous variable, perhaps in terms of the variable's relationship to other parameters. E.g. the continuous train unit motion could be modelled via the finite state of value set of "not yet at next scheduled stopping point" and "at next scheduled stopping point". Where it is not possible or practical to model such states with a finite-state machine, another deterministic behavioural specification method shall be defined on the logical component: <ul style="list-style-type: none"> • Mathematical function ($y = f(x)$, transfer function in s or z domain); • Truth table; • Lookup table with deterministic conditions. <p>More detailed guidance on how to model non-finite state machine behaviour will be provided in a later version of this process step.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> RCAMT-329 - ARCH.122: Add guidance on alternative behavioural specification (besides STMs) FINISHED </div>
Tools and non-human resources	Team for Capella
Cardinality	One off with allowed revisions.
Completion criteria	<p>All control and observe functions that have been transferred from logical level to physical level have been re-purposed such there exist a mimic and a master version of those functions.</p> <p>Each mimic and master function is allocated to a logical component that represents a specific behaviour, which in turn is modelled as state machine or other appropriate behavioural specification.</p>
Design review	ARCH.R.7 ARCH.R.8
Step done by (Responsible)	Subsystem architect
Provides input to /assists (Contributes)	Inclusive OR: <ul style="list-style-type: none"> • System architect • Cross-cutting engineer • Engineer
Uses outputs (Informed)	None directly

ARCH.186 Consolidate single subsystem interface

ARCH.934 Define supporting physical architecture

- [ARCH.125 Define structure of communication assets](#)
- [ARCH.126 Define structure of computation assets](#)
- [ARCH.127 Define location kinds](#)

ARCH.125 Define structure of communication assets

Goal	Define the communication elements and their interfaces
Requirements met by this process step	DCC R:7A Track side requirement (7)
Inputs	
Outputs	AMOD-096 Communication asset structure
Methodology	<p>ARCH.116 defines the subsystems, which are a kind of tenderable element. Among those tenderable elements, there is also the communication elements that would be part of our design.</p> <p>This process step aims at defining those communication elements. Here are the main step:</p> <ol style="list-style-type: none"> 1. Define the communication elements 2. Define the interfaces of those communication elements to the computation hardware (through component commsystem connection) 3. Define the interfaces between communication elements and the subsystems (through subsystem commsystem connection)
Tools and non-human resources	Team for Capella
Cardinality	
Completion criteria	All intended structure of communication assets are defined
Design review	
Step done by (Responsible)	Systems architect
Completion criteria evaluated by (Accountable)	

ARCH.126 Define structure of computation assets

Goal	Define the computation elements
Requirements met by this process step	DCC R:7A Track side requirement (7)
Inputs	
Outputs	AMOD-097 Computation asset structure
Methodology	<p>ARCH.116 defines the subsystems, which are a kind of tenderable element. Among those tenderable elements, there is also the computation elements that would be part of our design.</p> <p>This process step aims at defining those computation elements. Here are the main steps:</p> <ol style="list-style-type: none"> 1. Define the computation elements (software & hardware) 2. Define the interfaces of those hardware computation elements to the communication elements 3. Define the hosting interfaces between computation elements and the software-bounded subsystems (through subsystem commsystem connection)
Tools and non-human resources	Team for Capella
Cardinality	
Completion criteria	All intended structure of computation assets are defined
Design review	
Step done by (Responsible)	Systems architect
Completion criteria evaluated by (Accountable)	

ARCH.127 Define location kinds

Goal	Define the environmental conditions for a kind of location where a subsystem might be deployed
Requirements met by this process step	Environment-specific requirements
Inputs	
Outputs	AMOD-095 Location kind definitions
Methodology	<p>General information</p> <p>Main steps</p>
Tools and non-human resources	Team for Capella
Cardinality	
Completion criteria	All intended location kinds are defined
Design review	
Step done by (Responsible)	Systems architect
Completion criteria evaluated by (Accountable)	

ARCH.935 Assess risks associated with physical architecture

- [ARCH.101 Carry out failure modes, effects and criticality analysis \(FMECA\) on physical functional chains](#)
- [ARCH.103 Identify new physical functions needed for robustness to failures](#)
- [ARCH.104 Identify new non-functional requirements needed for robustness to failures](#)
- [ARCH.105 Identify new non-functional requirements needed to mitigate hazards introduced by physical architecture choice](#)
- [ARCH.106 Identify new physical functions needed for mitigating hazards introduced by physical architecture choice](#)
- [ARCH.135 Carry out HAZID on tenderable elements](#)

ARCH.101 Carry out failure modes, effects and criticality analysis (FMECA) on physical functional chains

Goal	
Requirements met by this process step	EN 50126-1 7.6.1
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.103 Identify new physical functions needed for robustness to failures

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.104 Identify new non-functional requirements needed for robustness to failures

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.105 Identify new non-functional requirements needed to mitigate hazards introduced by physical architecture choice

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.106 Identify new physical functions needed for mitigating hazards introduced by physical architecture choice

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.135 Carry out HAZID on tenderable elements

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	

ARCH.936 Allocation/Derivation of NFRs

- [ARCH.131 Derive & allocate NFRs to location kinds](#)
- [ARCH.132 Derive & allocate NFRs to communication assets](#)
- [ARCH.133 Derive & allocate NFRs to computation assets](#)
- [ARCH.137 Allocate interface NFRs to interface layers/interface functions](#)
- [ARCH.155 Allocate hazard mitigation NFRs to subsystems](#)

ARCH.131 Derive & allocate NFRs to location kinds

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.132 Derive & allocate NFRs to communication assets

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.133 Derive & allocate NFRs to computation assets

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.137 Allocate interface NFRs to interface layers/interface functions

Goal	
Requirements met by this process step	None defined
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	

ARCH.155 Allocate hazard mitigation NFRs to subsystems

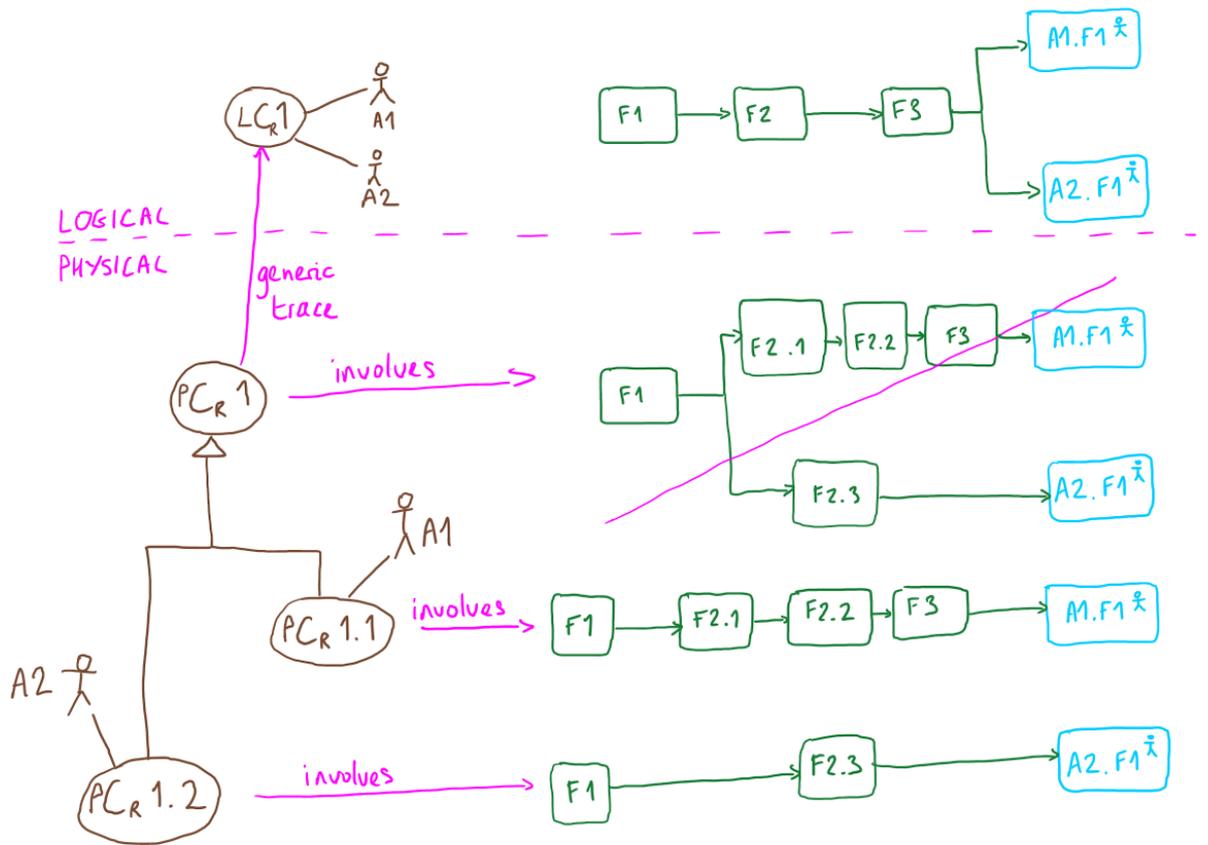
ARCH.938 Consolidate the overall architecture

- [ARCH.192 Refine physical capability realisations](#)
- [ARCH.176 Create subsystem exchange scenarios](#)
- [ARCH.136 Push back chosen physical architecture from child models to parent model](#)
- [ARCH.169 Consolidate traceability between model elements at physical level and model elements at logical level](#)
- [ARCH.168 Consolidate the overall generic physical architecture](#)
- [ARCH.100 Obtain stakeholder acceptance of preferred architecture](#)

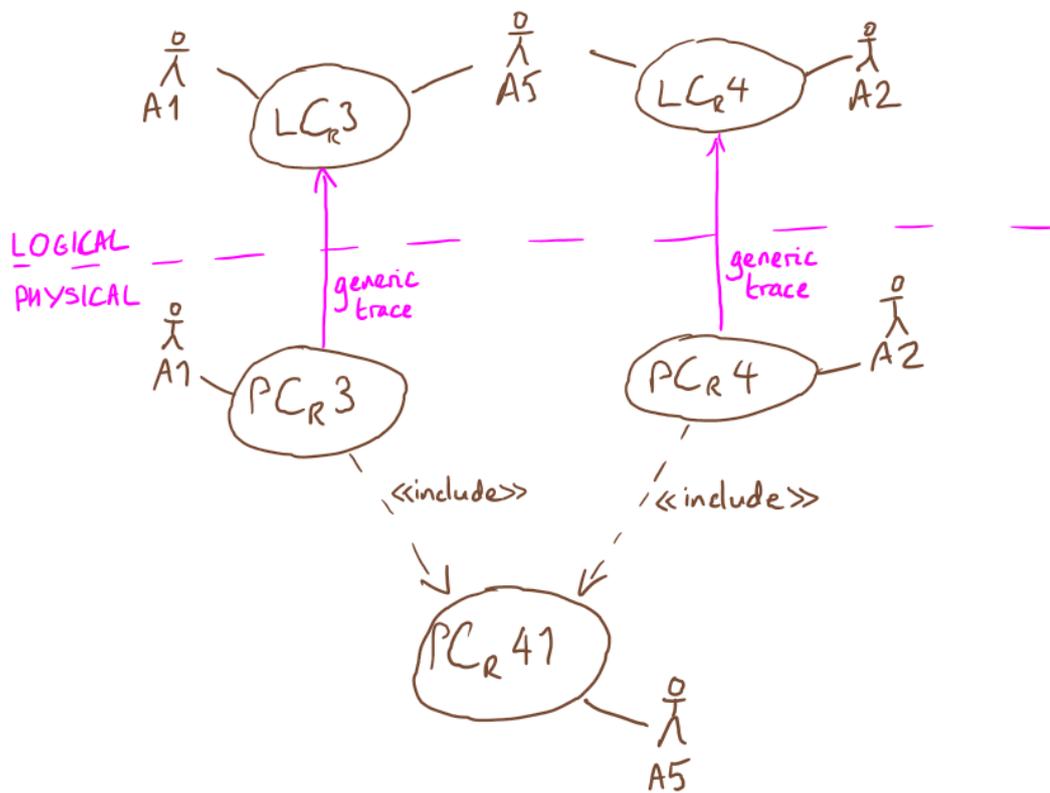
ARCH.192 Refine physical capability realisations

[SM-5415](#) - Create Confluence page for process step ARCH.192 Refine physical capability realisations FINISHED

Goal	Capture any refinements/adjustments to system capabilities after splits are done at physical level
Requirements met by this process step	ISO 15288 6.4.3.3 d) 2) EN 50126-1 7.6.2
Inputs	AMOD-147 Logical capability realisation
Outputs	AMOD-148 Physical capability realisation
Methodology	<p>In many cases, the system capabilities/logical capability realisations can stay at the same level of granularity when transferred down to physical level.</p> <p>However, the "physical capability realisations" still need to be shown on a dedicated viewpoint for at least two reasons</p> <ol style="list-style-type: none">1. this viewpoint is a statement of the capabilities that will be tested in the simulation of the physical architecture and therefore is an input to SIM and IVV activities.2. during decomposition of physical functions it may arise that a capability realisation must be split or specialised because the functional splitting results in variations in the functional chain.3. some logical components may need to be instantiated several times within different subsystem boundaries (for example, diagnostics, configuration data), resulting in several independent variant functional chains (example: logical capability realisation "update configuration data of topology" splits to "update configuration data of topology on S subsystem", "update configuration data of topology on PE subsystem", and so on, because these can be independently done at different times). <p>Some example conditions (non-exhaustive) under which it may be appropriate to refine the system capabilities at logic level:</p> <p>a) after functional decomposition, the capability's value can be independently delivered to two or more actors.</p> <p>That means that the end conditions of the capability can now be expressed in two or more independent groups.</p> <p>The capability realisation should be refined by capability generalisation - that is, there are now two independent types of the original capability.</p>

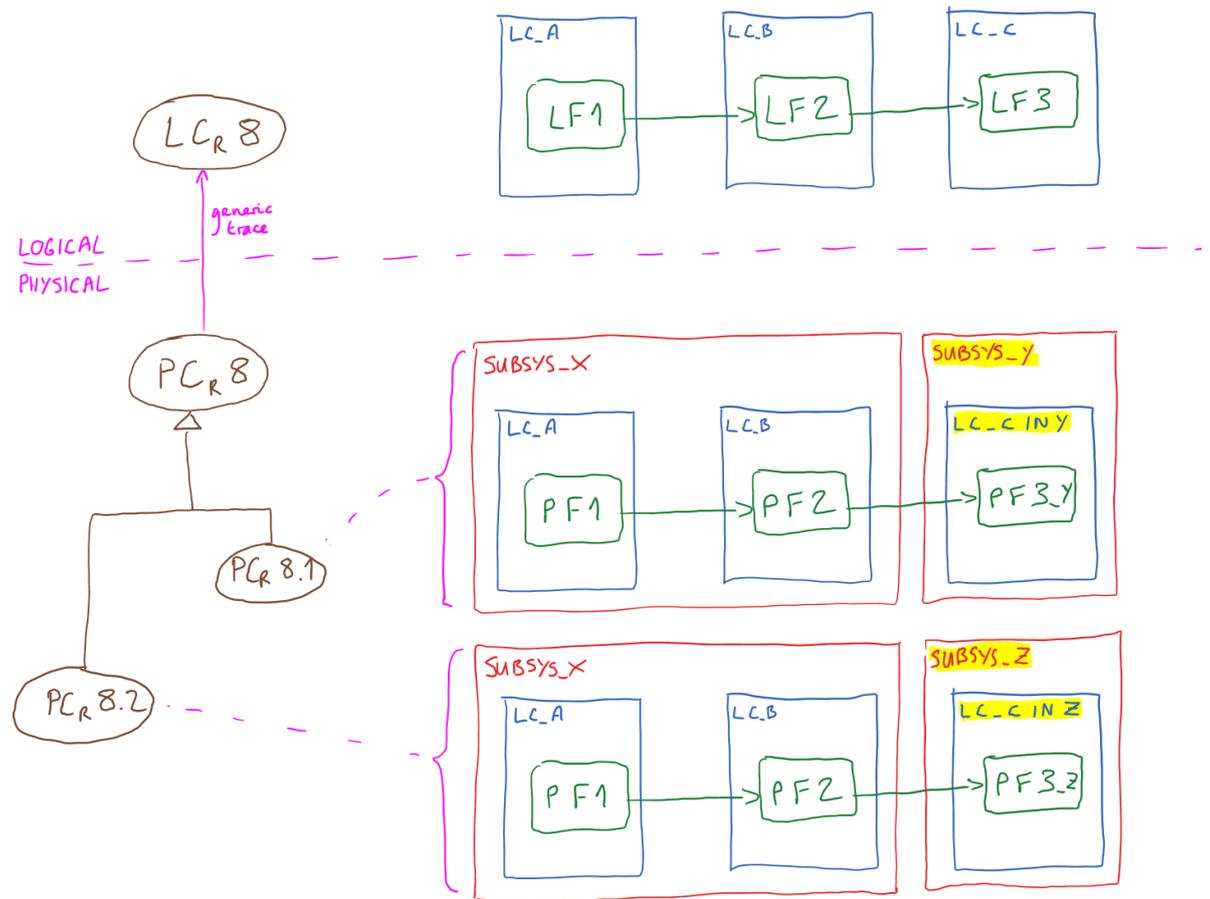


b) after functional decomposition, a functional chain portion is defined that can be separated off and reused as a capat in its own right; in this case, the new capability realisation should be **included** in the capability realisations it is reused within.



c) where a logical component is reused at physical level more than once, but hosted inside separate subsystems, then it becomes necessary to split the capability realisation so that it can be carried out independently for each hosting subsystem. This is particularly important for capabilities such as configuration data updates and diagnostics; at logical level these functionalities were defined once for the whole system; now, they will be deployed to separate subsystems so that they work independently.

The capability realisation should be refined by **capability generalisation** - that is, there are now two independent types of the original capability.



Tools and non-human resources	Team for Capella
Cardinality	Once per package of system capabilities being delivered
Completion criteria	The output view conforms to its modelling rules.
Design review	ARCH.R.7 Physical review - system capability realisation ARCH.R.8 Physical review - consolidated
Step done by (Responsible)	System architect

Provides input to /assists (Contributes)	None identified
Uses outputs (Informed)	IVV role TBD RCAMT-400 - Updates to ARCH to integrate artefact usage and role involvement with IVV CREATED

ARCH.176 Create subsystem exchange scenarios

[RCAMT-322](#) - Create Confluence page for process step ARCH.176 CREATED

Goal	<i>high-level statement of the engineering objective being addressed, including reference to standards/legislation if that is relevant</i>
Requirements met by this process step	<i>cite requirements from standards and legislation (by standard/document number and paragraph number only, no complete copy and paste)</i>
Inputs	<i>list of all input information needed for this process step to be carried out; for model views, add a hyperlink to the model view description; documents, published info, websites, knowledge from particular roles are all admissible.</i>
Outputs	<i>list of all output artefacts this process should produce; for model views, add a hyperlink to the model view description; for documents, add a link to the document template or to the document publication area if it is directly generated from Confluence;</i>
Methodology	<i>state the methodology as a list of instructions; this information forms the Description field of a ticket for this process to be done, so it must be sufficient for someone to follow the process, even if they have not done it before.</i>
Tools and non-human resources	<i>list of all tools (physical and software) needed for this process, plus any non-obvious resources that need to be planned (such as access to particular non-DBB individuals, operational areas, site visits...)</i>
Cardinality	<p><i>state how many times this process step is expected to be carried out - e.g.</i></p> <ul style="list-style-type: none"> • <i>one-off</i> • <i>once per <model element></i> • <i>once per baseline</i> <p><i>etc.</i></p>
Completion criteria	<p><i>state the conditions that must be TRUE before this process step can be considered complete.</i></p> <p><i>These should be inserted in the Acceptance Criteria of a Jira ticket for this process step to be done.</i></p>
Design review	<i>Link to the corresponding design review where the completion of this activity is evaluated.</i>
Step done by (Responsible)	<p><i>Identify the role that is responsible for organising this process step and producing the output;</i></p> <p><i>This is the Assignee in Jira.</i></p>
Provides input to /assists (Contributes)	<i>Identify the roles that assist in the work or provide input information</i>
Uses outputs (Informed)	<i>Identify the roles and/or process areas that make use of this information outside of the ARCH process area;</i>

ARCH.136 Push back chosen physical architecture from child models to parent model

 Page in draft, need to be reviewed and approved, feedback welcomed 😊

<p>Requirements met by this process step</p>	<p><input type="checkbox"/> Elsa Lamblin related to rcamt-160 : need to highlight dependency/impacted steps and deliverables about this push back and its inherent manual activities</p> <p>RCAMT-160 - RCA-METH Derive basic modeling patterns for NFRs / ARCH Define the methodology for modelling non-functional attributes IN IMPLEMENTATION</p>
<p>Inputs</p>	<p>for each submodels</p> <ul style="list-style-type: none"> • AMOD-xx1 RCA Submodel • AMOD-xx2 non-TRCA submodel <p>Inputs required are the following :</p> <ul style="list-style-type: none"> • AMOD-093 • AMOD-100 • AMOD-143 • AMOD-085 • AMOD-140 • AMOD-095 (tbc) • AMOD-096 (tbc)
<p>Outputs</p>	<p>AMOD-xxx DBS global Physical Architecture (tbc)</p>
<p>Methodology</p>	<p>Execute the scoped horizontal extraction from both submodel (RCA and non-RCA) to the over-model DBS.</p> <p>It will consist mainly in a copy-paste of all physical elements from submodels to the root model.</p> <p>⚠ Caution : The traceability from DBS logical architecture to those physical element will not be done at this step.</p> <p>It would be done :</p> <ol style="list-style-type: none"> 1) manually in the ARCH.169 and ARCH.168 steps, for short term 2) semi-automatically in middle/long term thanks to script developped later. <p>But manual check steps will be always executed after this phase.</p>
<p>Tools and non-human resources</p>	<p>Team for Capella</p> <p>System-to-System Capella plugin</p>
<p>Cardinality</p>	<p>One submodel for RCA project, one submodel for non-RCA scope, for one root DBS model</p>
<p>Completion criteria</p>	<p>All the physical architecture of both submodels (RCA and non-RCA) has been imported in root DBS model.</p>

Design review	<p><i>The Engineer should check if the scope is well imported to the root model from each submodel, which means :</i></p> <ul style="list-style-type: none"> • <i>Physical actors, nodes physical components, physical links and physical ports</i> • <i>Deployed behavior physical components, components exchanges, flow ports</i> • <i>Physical functions, functional exchanges</i> • <i>Physical functional chains</i> • <i>Physical external actors, physical external actors exchanges</i>
Step done by (Responsible)	<i>Engineer</i>
Provides input to/assists (Contributes)	<i>Engineers responsible for each submodels</i>
Uses outputs (Informed)	

ARCH.169 Consolidate traceability between model elements at physical level and model elements at logical level

[RCAMT-330](#) - Create Confluence page for process step ARCH.169 FINISHED

Goal	Traceability between model elements at logical level and model elements at physical level established.
Requirements met by this process step	None defined
Inputs	All relevant model elements on logical level and physical level
Outputs	AMOD-127 Physical level - logical level traceability report
Methodology	<p>Identify a model element to be considered in the physical architecture layer:</p> <p>Check if the model element already possesses a relationship to a corresponding model element at logical level. If that is the case, check if that relationship is reasonable and correct.</p> <p>All elements that are present on the physical layer as a result of a Capella transition are automatically provided with <i>Realisation Links</i> between themselves and their source logical-level elements. This is the preferred solution for assuring correct traceability in the model. If no <i>Realisation Links</i> can be automatically generated for a given model element, establish a new <i>Generic Trace</i> relationship between them.</p> <p>Repeat the above steps until all model elements at physical level possess either <i>Realisation Links</i> or an outgoing <i>Generic Trace</i> relationship to at least one corresponding model element at logical level. Afterwards, check once again whether all of those relationships are correct.</p> <p>Guidance on creating Generic Traces:</p> <ul style="list-style-type: none"> • Right click on the system-level model element in the Capella Explorer → Wizards → Trace Manager • Add a new outgoing generic trace relationship from the physical-level model element to the corresponding logical-level model element
Tools and non-human resources	Team for Capella
Cardinality	At least once prior to ARCH.R.8 Physical review - consolidated
Completion criteria	Each physical-level model element possesses either a Realisation Link or a Generic Trace relationship to at least one logical-level model element.
Design review	ARCH.R.8 Physical review - consolidated
Step done by (Responsible)	Cross-cutting engineer
Provides input to /assists (Contributes)	<ul style="list-style-type: none"> • Engineer • Cross-cutting engineer • Subsystem Architect

**Uses outputs
(Informed)**

None

ARCH.168 Consolidate the overall generic physical architecture

[RCAMT-358](#) - Create Confluence page for activity definition ARCH.168 CREATED

Goal	<i>high-level statement of the engineering objective being addressed, including reference to standards/legislation if that is relevant</i>
Requirements met by this process step	<i>cite requirements from standards and legislation (by standard/document number and paragraph number only, no complete copy and paste)</i>
Inputs	AMOD-095 Location kind definitions AMOD-097 Computation asset structure AMOD-093 Subsystems definition AMOD-085 Subsystems involved in single realised capability AMOD-143 Inter-subsystem interface definition
Outputs	AMOD-098 Consolidated tenderable element structure AMOD-142 Single subsystem definition
Methodology	<i>state the methodology as a list of instructions; this information forms the Description field of a ticket for this process to be done, so it must be sufficient for someone to follow the process, even if they have not done it before.</i>
Tools and non-human resources	Team for Capella
Cardinality	At least once prior to ARCH.R.8 Physical review - consolidated
Completion criteria	<i>state the conditions that must be TRUE before this process step can be considered complete.</i> <i>These should be inserted in the Acceptance Criteria of a Jira ticket for this process step to be done.</i>
Design review	ARCH.R.8 Physical review - consolidated
Step done by (Responsible)	Lead system architect

Provides input to /assists (Contributes)	<ul style="list-style-type: none">• Engineer• Cross-cutting engineer• Subsystem Architect
Uses outputs (Informed)	None

ARCH.100 Obtain stakeholder acceptance of preferred architecture

Goal	
Requirements met by this process step	ISO 15288 6.4.4.3 f)
Inputs	
Outputs	
Methodology	
Tools and non-human resources	
Cardinality	
Completion criteria	
Design review	
Step done by (Responsible)	
Completion criteria evaluated by (Accountable)	
Provides input to/assists (Contributes)	
Reviews (Contributes)	
Uses outputs (Informed)	

ARCH.925 Transition from parent model to child models

 First draft, still in progress

- [ARCH.112 Execute transition from logical architecture of parent model to child model system architecture](#)

ARCH.112 Execute transition from logical architecture of parent model to child model system architecture

 Page in draft, need to be reviewed and approved, feedback welcomed 😊

Goal	Provide the submodels in a system-of-system consistent way.
Requirements met by this process step	Not defined yet
Inputs	Parent Model elements on logical level
Outputs	for each single system model elements in the logical layer in system of system level : creation of a separated model starting at system layer
Methodology	<p>For the set of logical components currently being considered, initiate the manual transition of the logical components to a separated subsystems model, with all the inter-relationships with other logical components which will be then, external to the new submodel, and with all the design choices inherited from the DBS over-model.</p> <p>Execute vertical transition(see 3.1.2) for the all main logical components representing single systems into different submodel (which should represent the RCA submodel, according to the limit decided in accordance with the multi-team requirements and RCA project, and the non-RCA projects, which should contain the other components not contained in RCA project)</p> <p>The transition should transform the LA subpart as follow:</p> <ul style="list-style-type: none"> • Logical capability realisations to be automatically transferred to System level of the submodel as system capability. • Logical actors to be automatically transferred to System level of the submodel as external actors. • Logical actor functions to be automatically transferred to System level of the submodel as external actor functions allocated to their respective actors. • Logical components, component exchanges, logical functions and functional exchanges to System level of the submodel as systems, systems component exchanges, system functions and system functional exchanges. • Functional chains to be automatically transferred to System level of the submodel
Tools and non-human resources	<p>Team for Capella</p> <p>System-to-System Capella plugin</p>
Cardinality	once per logical main components in the SoS model, could be scalable in the future in case of decoupling subsystems and referential architecture
Completion criteria	The main logical component(s) selected for submodel is/are well transitioned, with all the other external systems /logical components interacting with him/them seen as external actors at system level in the new submodel,
Design review	The Engineer should check if the scope is well derived for each submodel
Step done by (Responsible)	Engineer

Provides input to /assists (Contributes)	<i>Inclusive OR:</i> <ul style="list-style-type: none">• <i>Engine room</i>• <i>System architect</i>• <i>Cross-cutting engineer</i>
Uses outputs (Informed)	<i>Engineers responsible to the child model design and management</i>