Giorgio Ciacchella

Adopting Model-Based Practices with Capella and TASTE for Student-Developed CubeSat Systems

How MBSE is Helping GU Orbit to Design an Autonomous Nanosatellite

Contents

5’ # GU Orbit

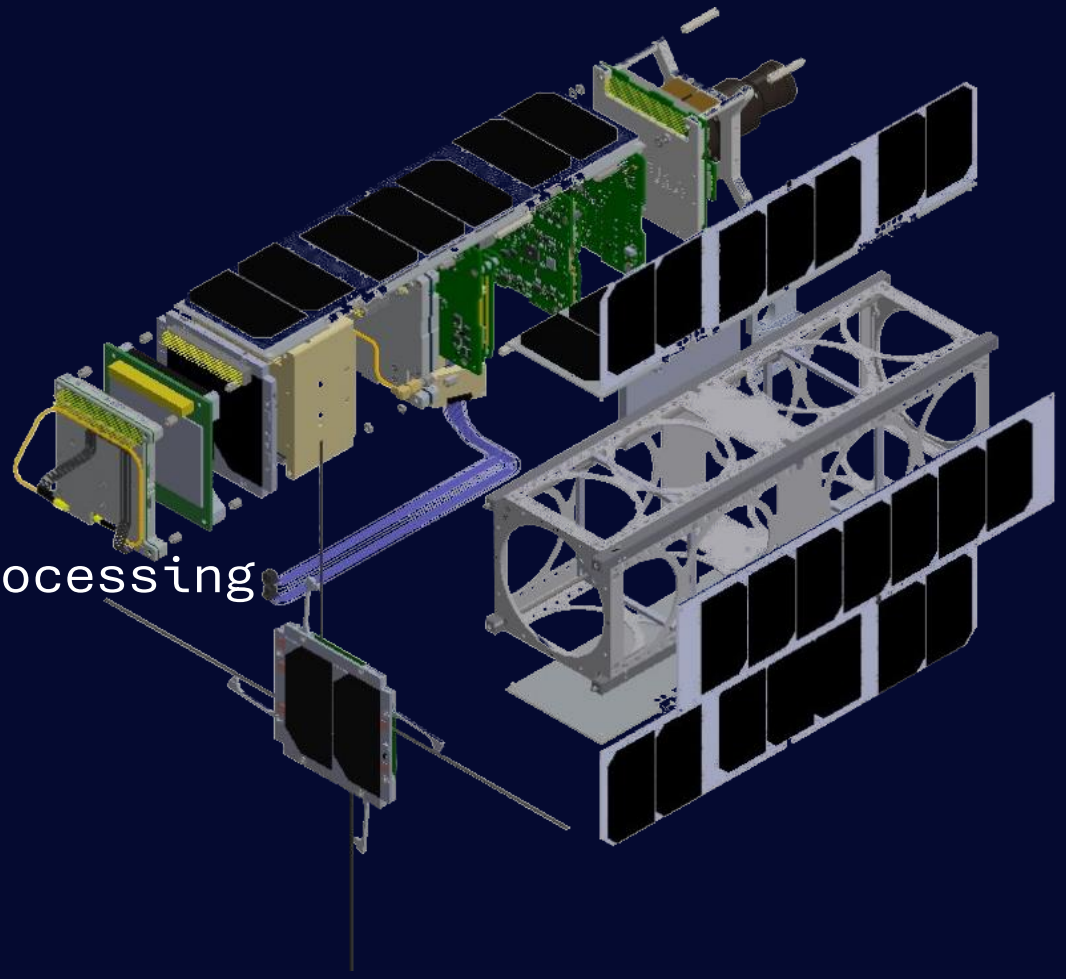
- Who we are
- What we do
- How we do it

Who we are

- University of Glasgow, Scotland
- Student-run society
  - Founded 2019
  - Mostly Bachelors / Masters students
  - Diverse team
    - 80+ students
    - 20+ countries
    - 10+ disciplines

What we do

- Nanosatellite development
  - Three pillars:
    - Technology Demonstration
    - Data Collection
    - In-house Subsystem Development
  - Technologies:
    - Machine Learning & On-Board Processing
    - Deployable Drag-Sail
  - Mission:
    - Earth Observation
    - Wildfire Risk Assessment



What we do

- Nanosatellite development
  - Started the **OirthirSAT** spin-off mission
    - Similar technologies
    - Coastline Monitoring mission
    - Off-the-shelf subsystems
    - Won £600K from UK Government!
    - Currently in Phase D (Assembly, Integration, Testing)

What we do

- Research & Development
  - SSEA-22
    - *Mechanical design and deployment of a quasi-rhombic pyramid drag sail for safe de-orbit of a 3U CubeSat*
  - IAC-23
    - *An Open-Source Method for Model-Based Development of Embedded Systems: Experience Report from a CubeSat Student Project [1, 2]*
    - *Calculation of the Death Index of the most catastrophic wildfires*
  - IAC-24 (upcoming!)
    - *Astraeus-01 Mission Proposal: a Student CubeSat for Autonomous Wildfire Risk Assessment Enhanced by Technology Demonstration*
    - *AI-Based Wildfire Risk Assessment from Low-Cost Multispectral Data: Collection, Processing, and Analysis for Sub-6U CubeSat Missions*
    - *Financial Feasibility Analysis of the Launch of Miniature Satellites for Student Teams, Based on Mission Design*

How we do it

- Model-Based approaches!
  - Systems Engineering
    - Functional paradigm (Capella)
    - Numerical paradigm (beyond scope)
  - Software Engineering
    - Embedded systems (TASTE)

How we do it

- Agile lifecycles
  - How
    - ~2 year “sprint”
    - Sounding balloon missions
  - Why
    - Graduations = fast & forced turnover
    - Consistent full lifecycle experience
    - Cyclical full lifecycle validation
  - What: CloudView
    - 3 subsystems
    - 3 hours
    - 35 km

15’

MBSE Adoption

- Motivations
- Tools
- Processes
- Lessons Learnt

Motivations

- Graduations = fast & forced turnover
- Document-based SE
  - “Student-grade”
  - Voluntary compliance
  - Teams regularly starting over!
- Model-based SE
  - Still “student-grade”
  - Consistency enforced
  - Teams consistently progressing!

Tools

- Capella (Systems Engineering)
  - Developed by Thales & Obeo
  - Free and Open-Source
  - Implements ARCADIA method
- TASTE (Embedded Software Development)
  - Developed by the European Space Agency & industry
  - Free and Open-Source
  - Targets heterogenous systems
    - Multiple code blocks
    - Multiple languages
    - Multiple hardware nodes
  - Support for formal verification!

Processes

- Knowledge acquisition
  - Online tutorials & docs
  - Expert training courses
- Knowledge transfer
  - Pair/Mob sessions
    - 1+ Navigator
      - experienced
      - can only talk - no actions
    - 1 Driver
      - inexperienced
      - physically acts
  - Applied to:
    - Modelling (Systems)
    - Programming (Software)

Processes

- Model Development
  - Concurrent engineering: interdisciplinary collaboration
- Software Development
  - "Bridge method" (next section)

Lessons Learnt

- MBSE helpful in learning SE
  - Method + Tool integration
- Mob modelling + Concurrent engineering =
  - Method knowledge transfer (SE, MBSE, ARCADIA)
  - Tool knowledge transfer (Capella)
  - System knowledge exchange!

25’

Embedded Systems Development

“Bridging the Gap”

- The Gap
- The Precedent
- Tools and Workflows
- The Bridge

The Gap

- Systems Engineering → actual systems
- Output of SE
  - Traditionally: spec for manual implementation
  - Better: the actual system itself!
- Tools exist to bridge this gap
  - Proprietary
  - Outside Capella ecosystem

The Precedent

- Automatic bridge plugin
- Developed:
  - In 2020
  - By N7 Space
  - Under ESA contract
- No further contracts = no further maintenance
- Divergence of both tools’ internals
- Stuck in 2020 ☹

From https://mbse-capella.org/addons:

Open-Source Add-ons

Capella-TASTE-Plugin

Contact: N7 Space - License: EPL - EXPORT BRIDGE

The plugin allows to export Capella data and physical architecture models into ASN.1 and AADL models compatible with the TASTE toolchain maintained by the European Space Agency, making it possible to further concretize the model and generate executable code that can be compiled and deployed onto an embedded target platform. Data packages, classes, enumerations, collections, numeric types and similar are transformed into the corresponding ASN.1 constructs. Architecture models are transformed into TASTE Interface and Deployment Views. Additionally, on a Linux platform (e.g. within TASTE Virtual Machine), the plugin provides several convenience actions enabling to partially replace TASTE GUI and facilitate model refinement and compilation directly from Capella.

The plugin was developed under a programme of, and funded by, the European Space Agency. Any views expressed within the plugin or its documentation can in no way be taken to reflect the official opinion of the European Space Agency.

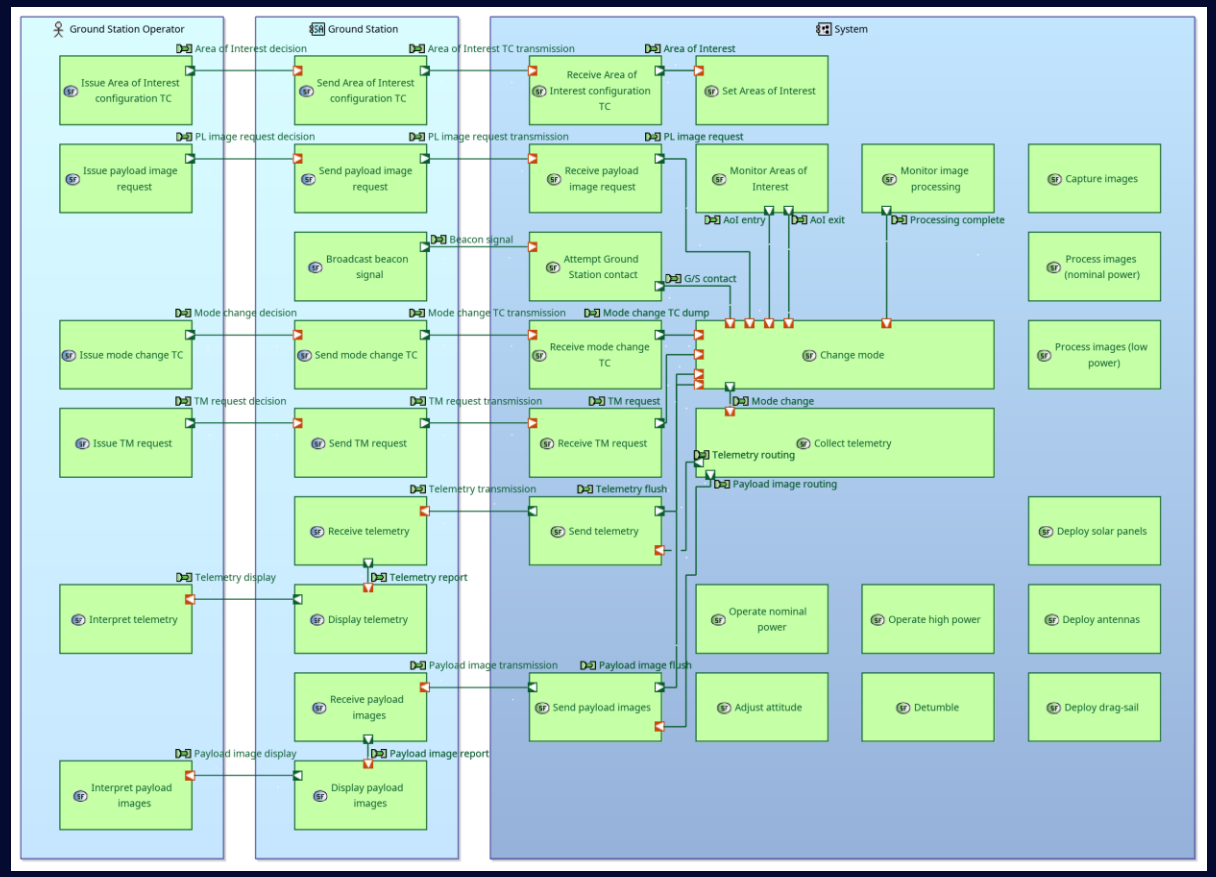
Installation procedure

Tools and Workflows

- Capella
  - ARCADIA workflow
  - 4 main levels
    - Operational
      - context
    - System
      - system
    - Logical
      - subsystems
    - Physical
      - components
  - Transverse models
    - Finite State Machine
    - Class Diagram

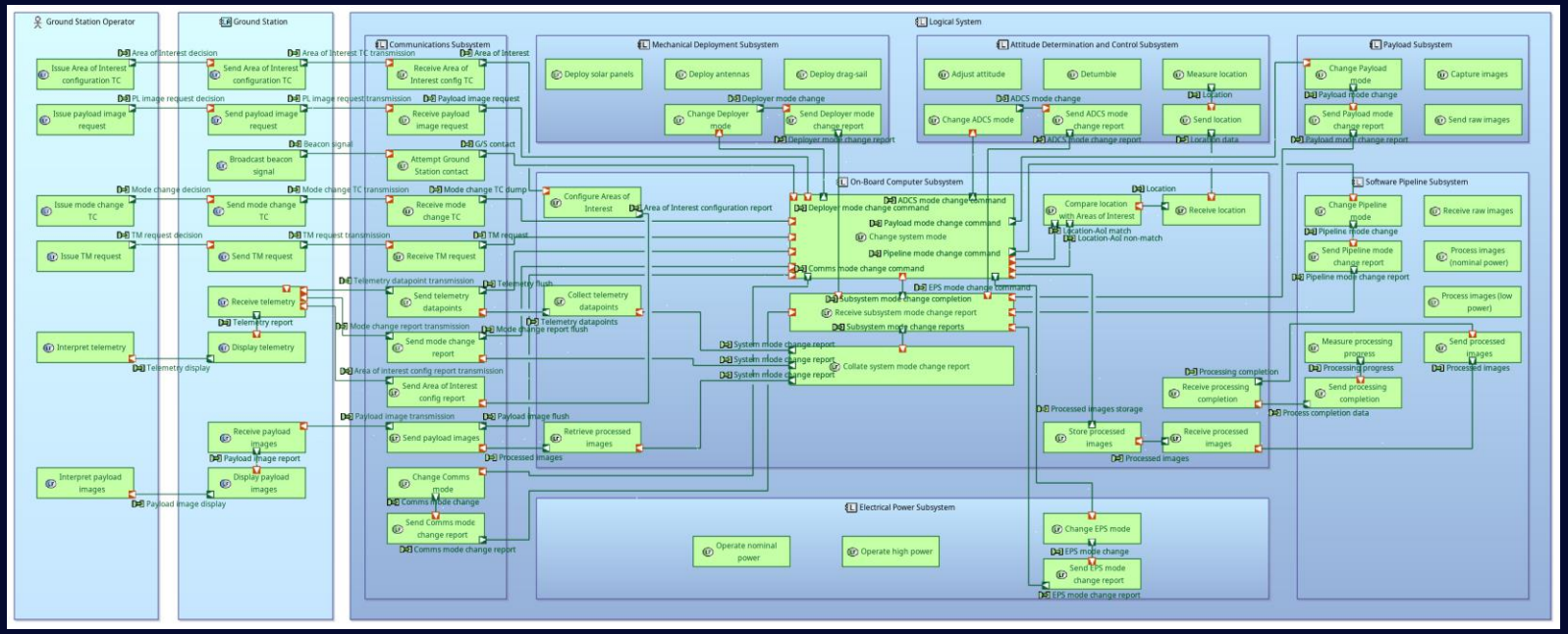
Tools and Workflows

- Capella
  - ARCADIA workflow
  - 4 main levels
    - Operational
      - context
    - System
      - system
    - Logical
      - subsystems
    - Physical
      - components
  - Transverse models
    - Finite State Machine
    - Class Diagram



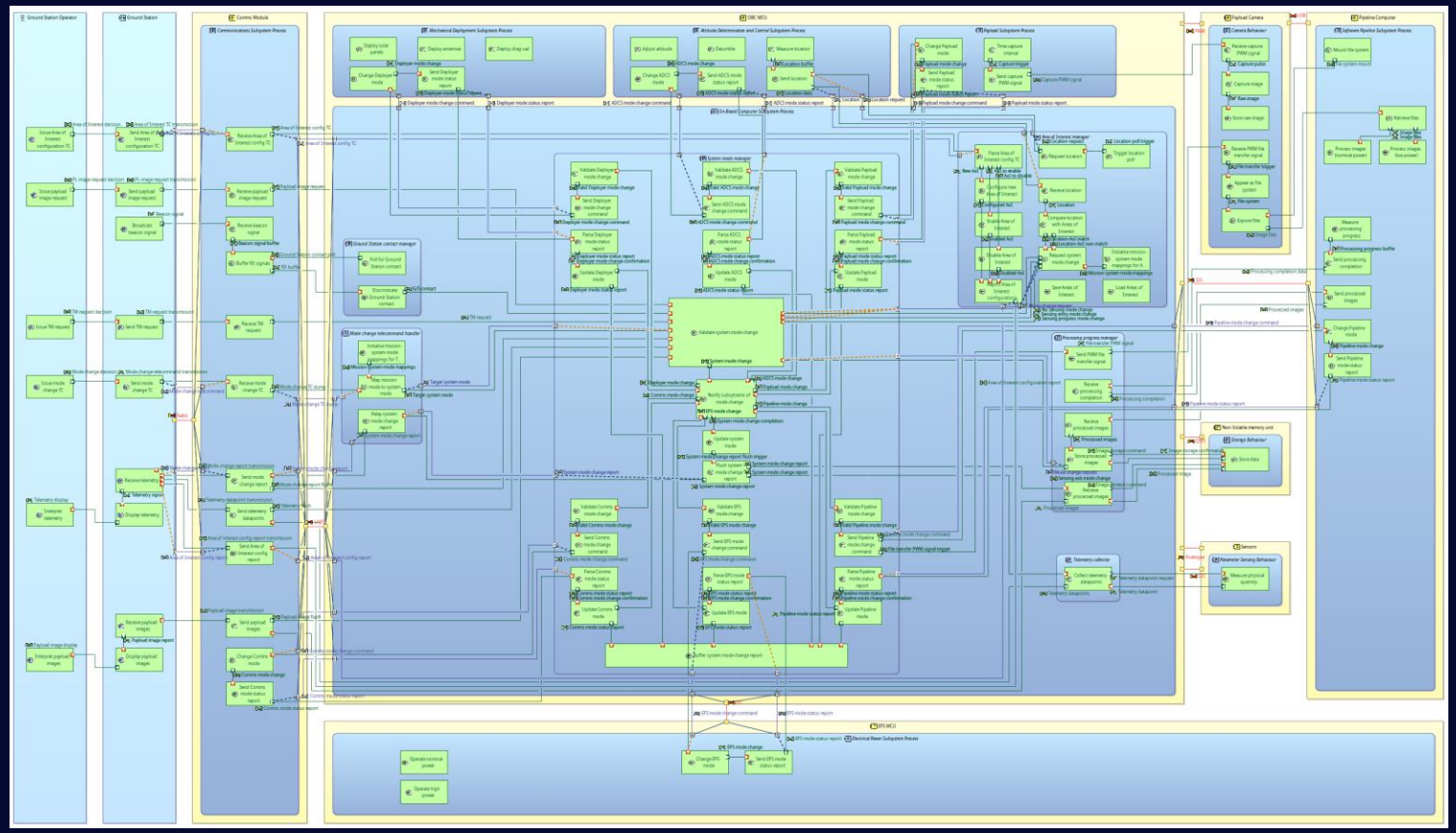
Tools and Workflows

- Capella
  - ARCADIA workflow
  - 4 main levels
    - Operational
      - context
    - System
      - system
    - Logical
      - subsystems
    - Physical
      - components
  - Transverse models
    - Finite State Machine
    - Class Diagram



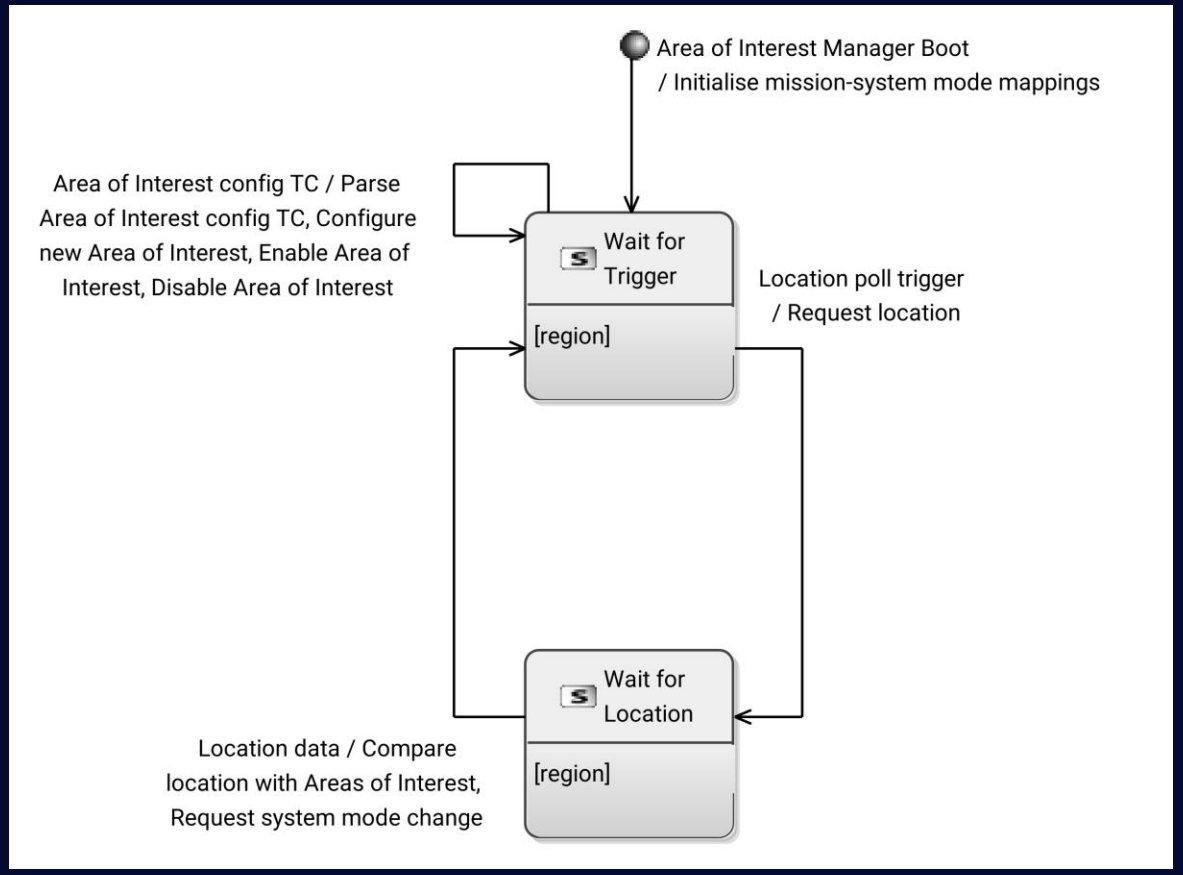
Tools and Workflows

- Capella
  - ARCADIA workflow
  - 4 main levels
    - Operational
      - context
    - System
      - system
    - Logical
      - subsystems
    - Physical
      - components
  - Transverse models
    - Finite State Machine
    - Class Diagram



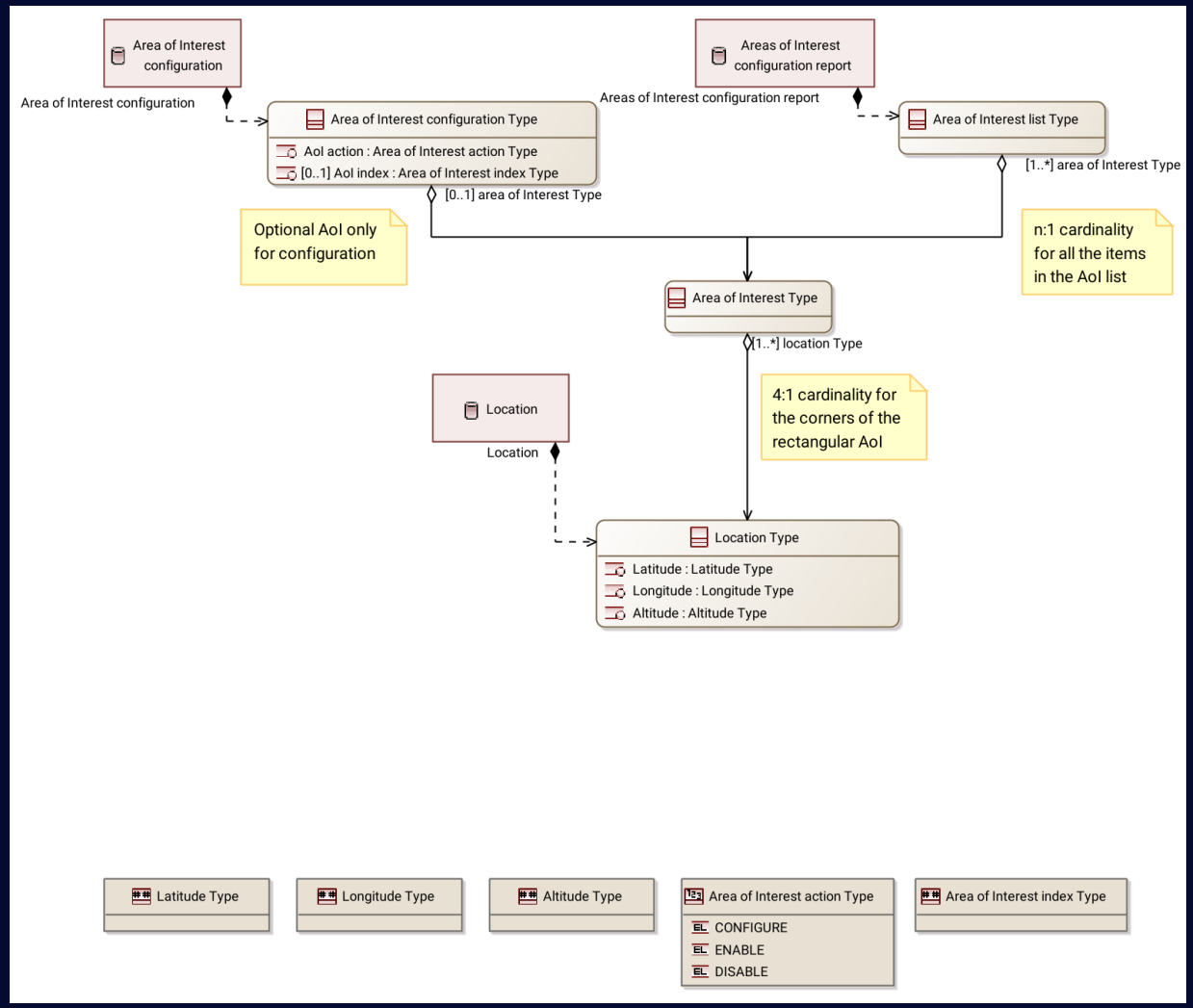
Tools and Workflows

- Capella
  - ARCADIA workflow
  - 4 main levels
    - Operational
      - context
    - System
      - system
    - Logical
      - subsystems
    - Physical
      - components
  - Transverse models
    - Finite State Machine
    - Class Diagram



Tools and Workflows

- Capella
  - ARCADIA workflow
  - 4 main levels
    - Operational
      - context
    - System
      - system
    - Logical
      - subsystems
    - Physical
      - components
  - Transverse models
    - Finite State Machine
    - Class Diagram



Tools and Workflows

- TASTE
  - Own workflow
  - 3 views
    - Data
      - data types
    - Interface
      - components
      - interfaces
    - Deployment
      - SW → HW allocations

Tools and Workflows

- TASTE
  - Own workflow
  - 3 views
    - Data
      - data types
    - Interface
      - components
      - interfaces
    - Deployment
      - SW → HW allocations

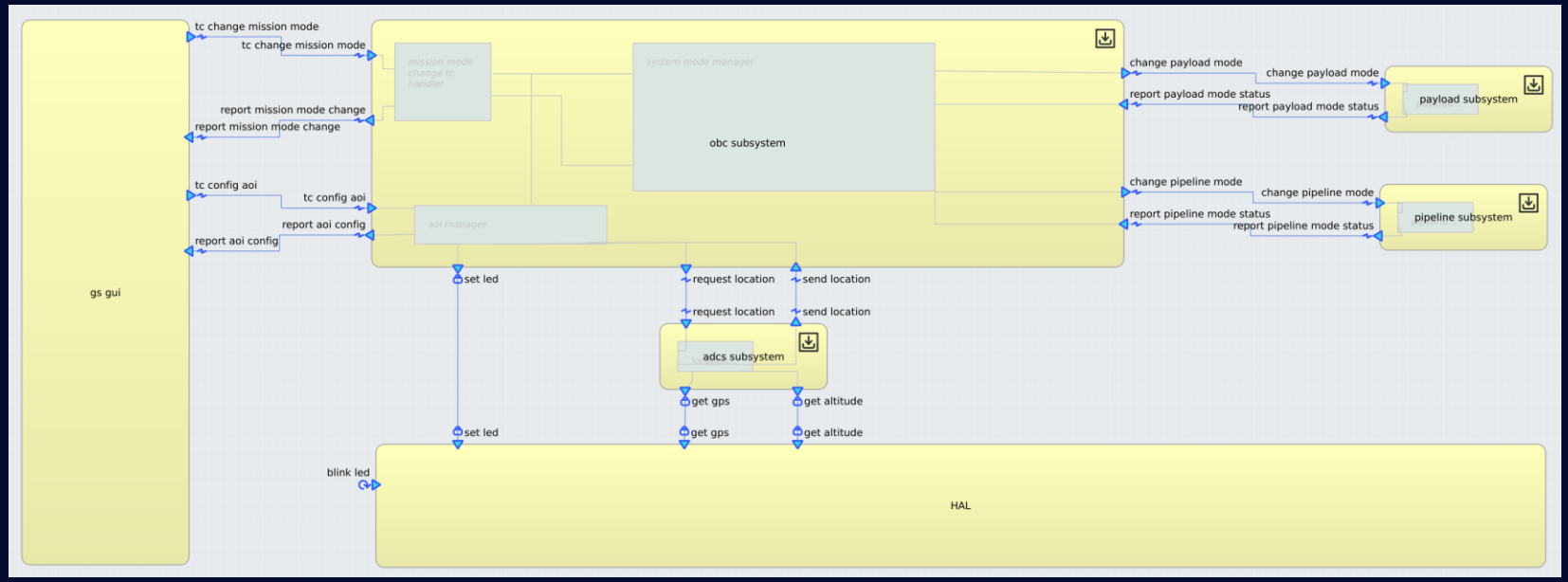
```
Latitude-WGS84          ::= REAL (-90.0 .. 90.0)
Longitude-WGS84         ::= REAL (-180.0 .. 180.0)
Altitude-m              ::= REAL (0.0 .. 100000.0)
Location                ::= SEQUENCE {
    lat                 Latitude-WGS84,
    lon                 Longitude-WGS84,
    alt                 Altitude-m
}

Area-of-Interest        ::= SEQUENCE {
    loc-nw              Location,
    loc-ne              Location,
    loc-se              Location,
    loc-sw              Location,
    enabled             BOOLEAN
}

AoI-List                ::= SEQUENCE (SIZE (4)) OF
Area-of-Interest
AoI-List-Index          ::= INTEGER (0 .. 3)
```

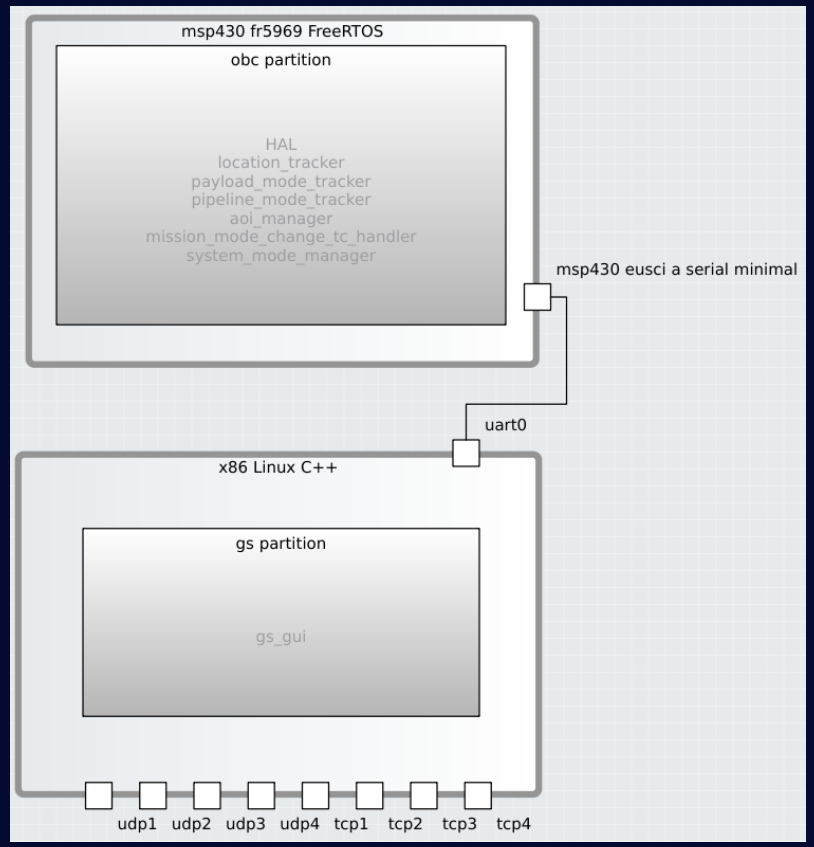
Tools and Workflows

- TASTE
  - Own workflow
  - 3 views
    - Data
      - data types
    - Interface
      - components
      - interfaces
    - Deployment
      - SW → HW allocations



Tools and Workflows

- TASTE
  - Own workflow
  - 3 views
    - Data
      - data types
    - Interface
      - components
      - interfaces
    - Deployment
      - SW → HW allocations



The Bridge: Goals

- Make the most of the similarities between Capella and TASTE
- Future-proof
  - Theoretical foundation
    - Mappings
    - Activities
  - Follow-up work
    - Automation
    - Formalisation

The Bridge: Entity/Relationship Mappings

Capella	TASTE
Node PC: Root / Fork / Leaf	(HW Board) / (HW Module) / Device
Behaviour PC: Root / Fork / Leaf	Partition / Function Group / Function
Function	Function implementation step
State of Leaf *Behaviour PC*	SDL State
Transition of Leaf *Behaviour PC*	SDL Transition
Physical Port	Device Port
Physical Link	Device Link
Flow Port: In / Out	Interface endpoint: Provided / Required
Interface	Interface
Exchange Item	Interface parameter
Class	ASN.1 `SEQUENCE` type
Data Type	Primitive ASN.1 type

Table 1: Mapping of entities between Capella (Physical level) and TASTE.

Capella	TASTE
Root *Behaviour PC* to *Node PC*	Deployment of Partition to Device
Fork/Leaf *Behaviour PC* to Root *Behaviour PC*	Deployment of Function / Function Group to Partition
Fork/Leaf *Behaviour PC* to Fork *Behaviour PC*	Function nesting
Flow Port to *Physical Port*	Deployment of Interface to Device Link
Function to Leaf *Behaviour PC*	Function implementation steps (informal)
Function to FSM *Transition*	SDL Function implementation steps (formal)
Exchange Item to *Interface*	Assignment of Parameters to Interface
Class or *Data Type* to *Exchange Item*	Parameter typing
Child *Class* to Parent *Class*	ASN.1 `SEQUENCE` nesting
Data Type to *Class*	ASN.1 `SEQUENCE` field membership

Table 2: Mapping of relationships between Capella (Physical level) and TASTE.

The Bridge: Diagram Mappings

Capella	TASTE
Physical [CDB]	Data View
Physical [CII]	Interface View
[PAB] PCs View	Deployment View
Physical [MSM]	SDL Implementation

Table 3: Mapping of implementation diagrams between Capella and TASTE.

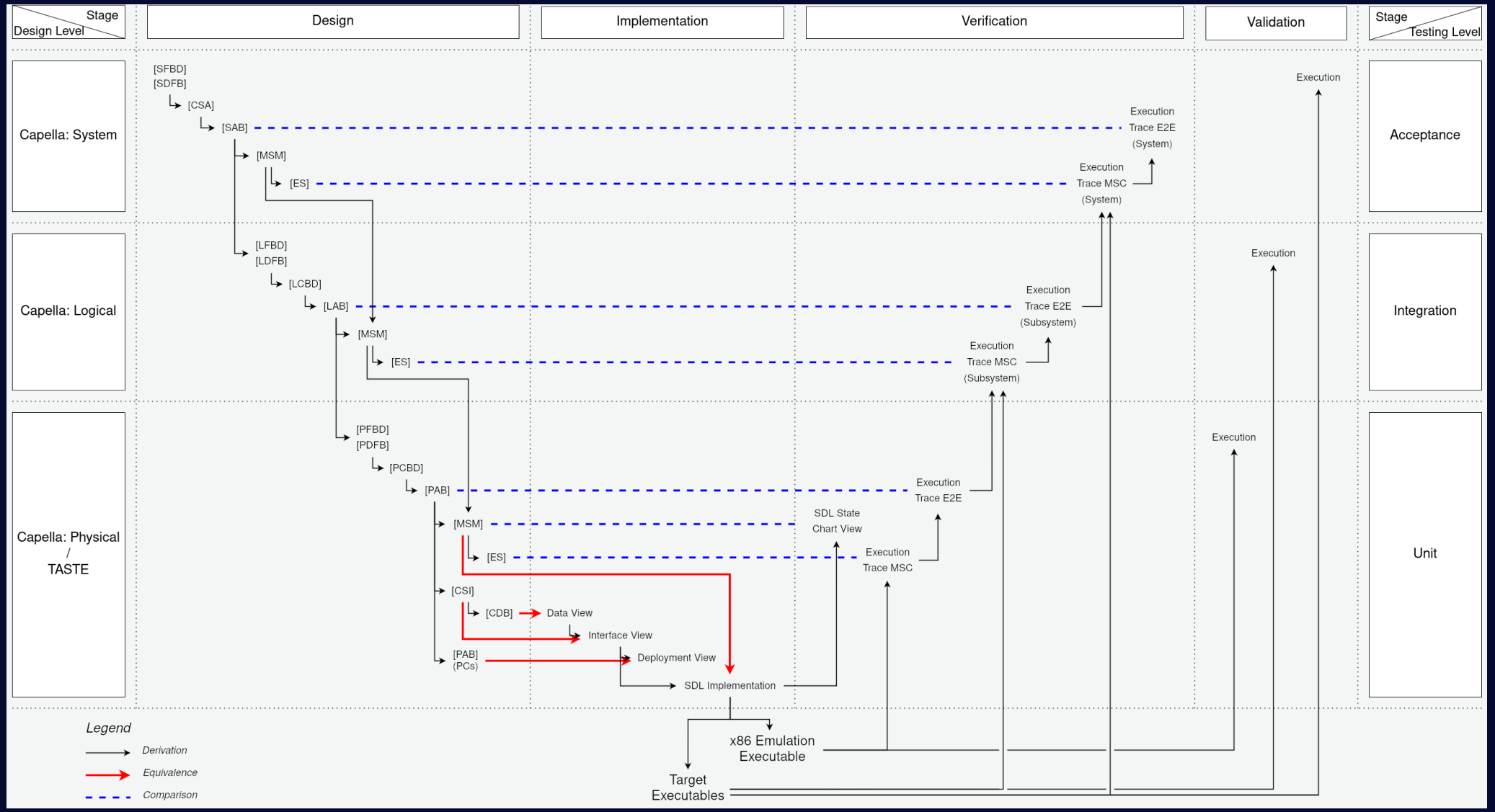
Capella	TASTE
Physical [MSM]	SDL State Chart View
Physical [ES]	Simulation MSC
[PAB] *FC* View	Simulation E2E
Logical [ES]	Execution Trace MSC
[LAB] *FC* View	Execution Trace E2E
System [ES]	Live Execution MSC
[SAB] *FC* View	Live Execution E2E

Table 4: Mapping of verification and validation diagrams between Capella and TASTE.

(red arrows in the next slide)

(blue dashes in the next slide)

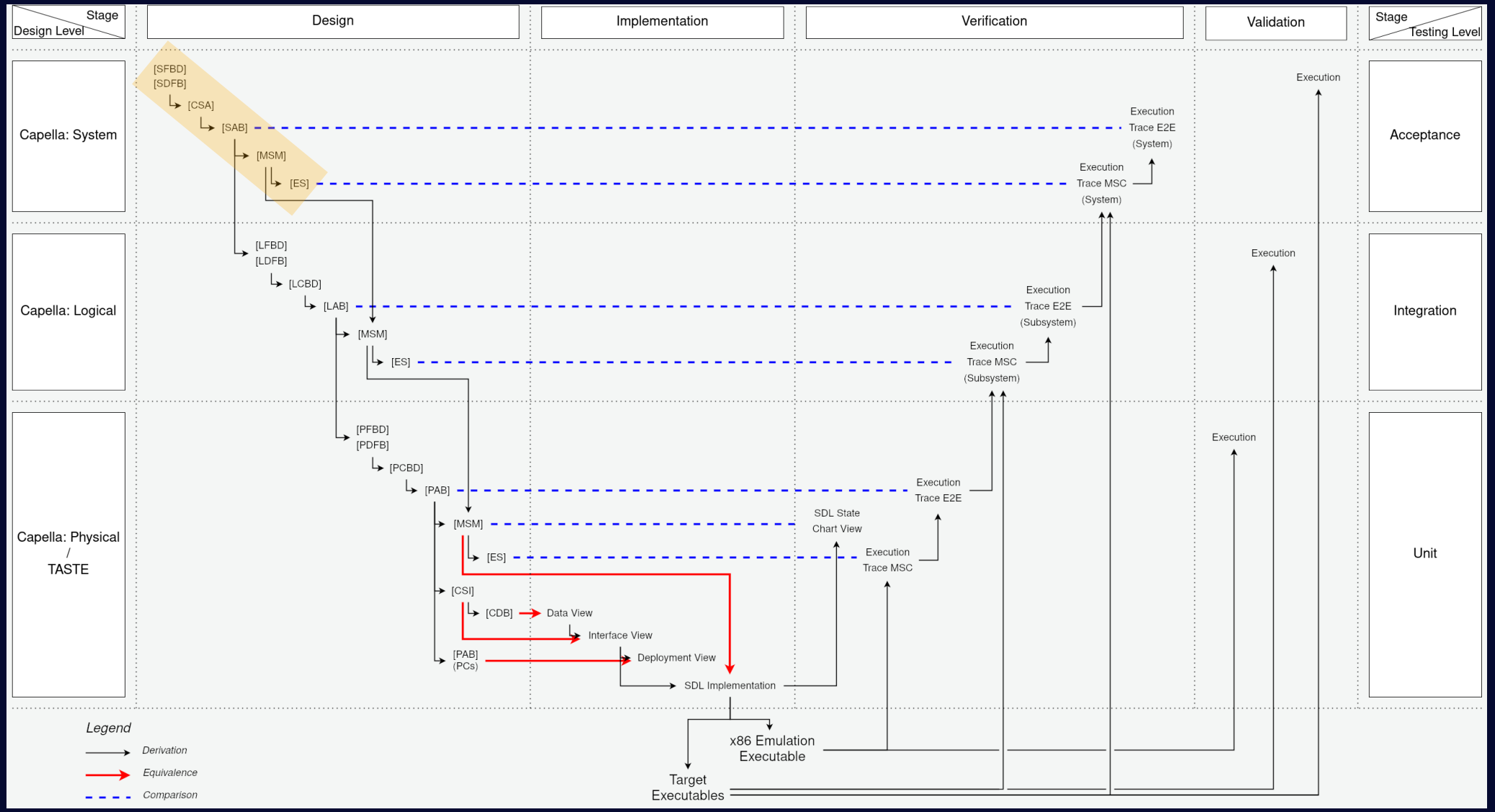
The Bridge: Workflow



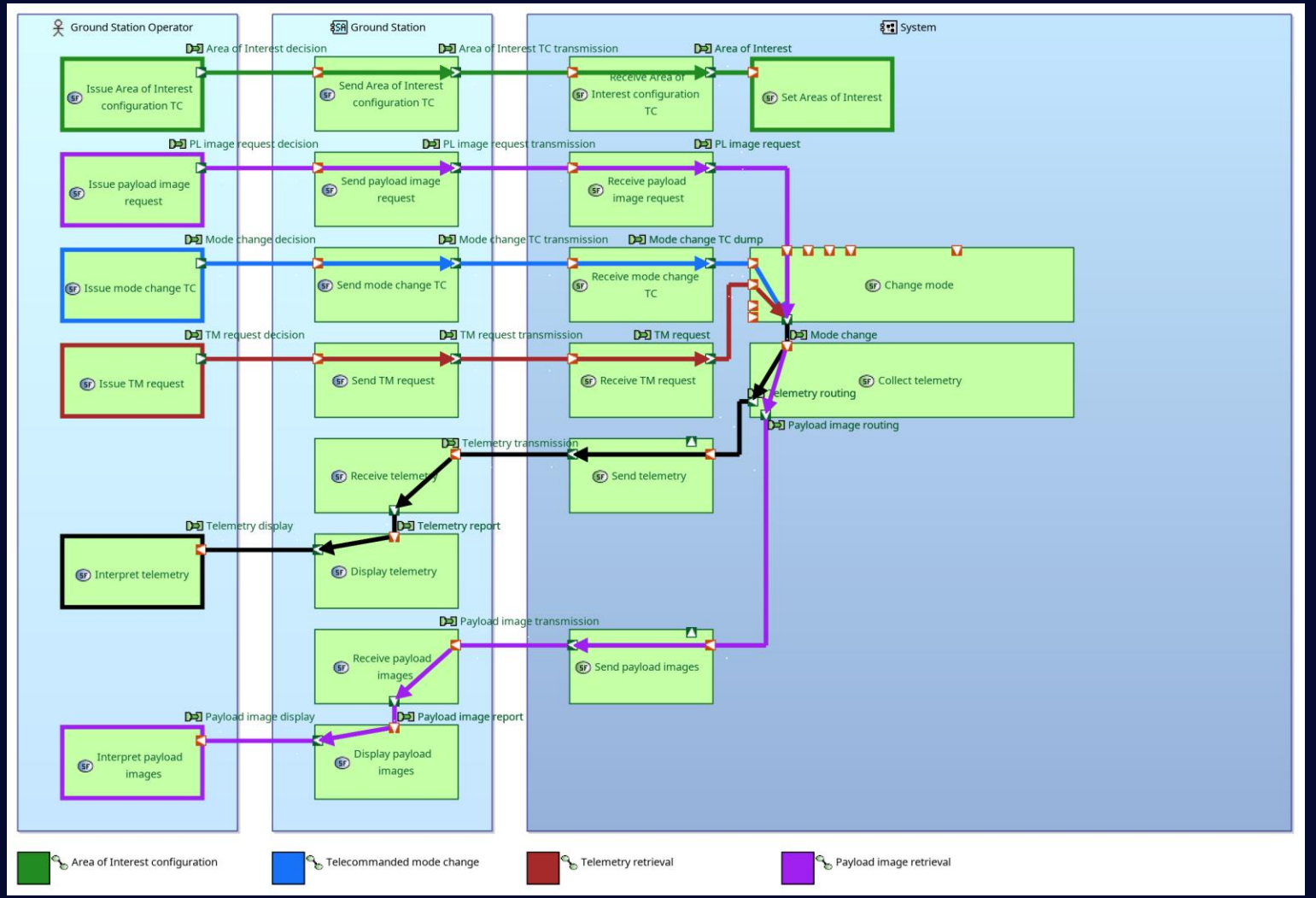
The Bridge: Walkthrough Case Study

- Area of Interest (AoI) management feature
  - Functions:
    - Configure AoIs
      - add/overwrite
      - enable
      - disable
    - Trigger imaging mode when inside AoI
    - Trigger non-imaging mode when outside AoI

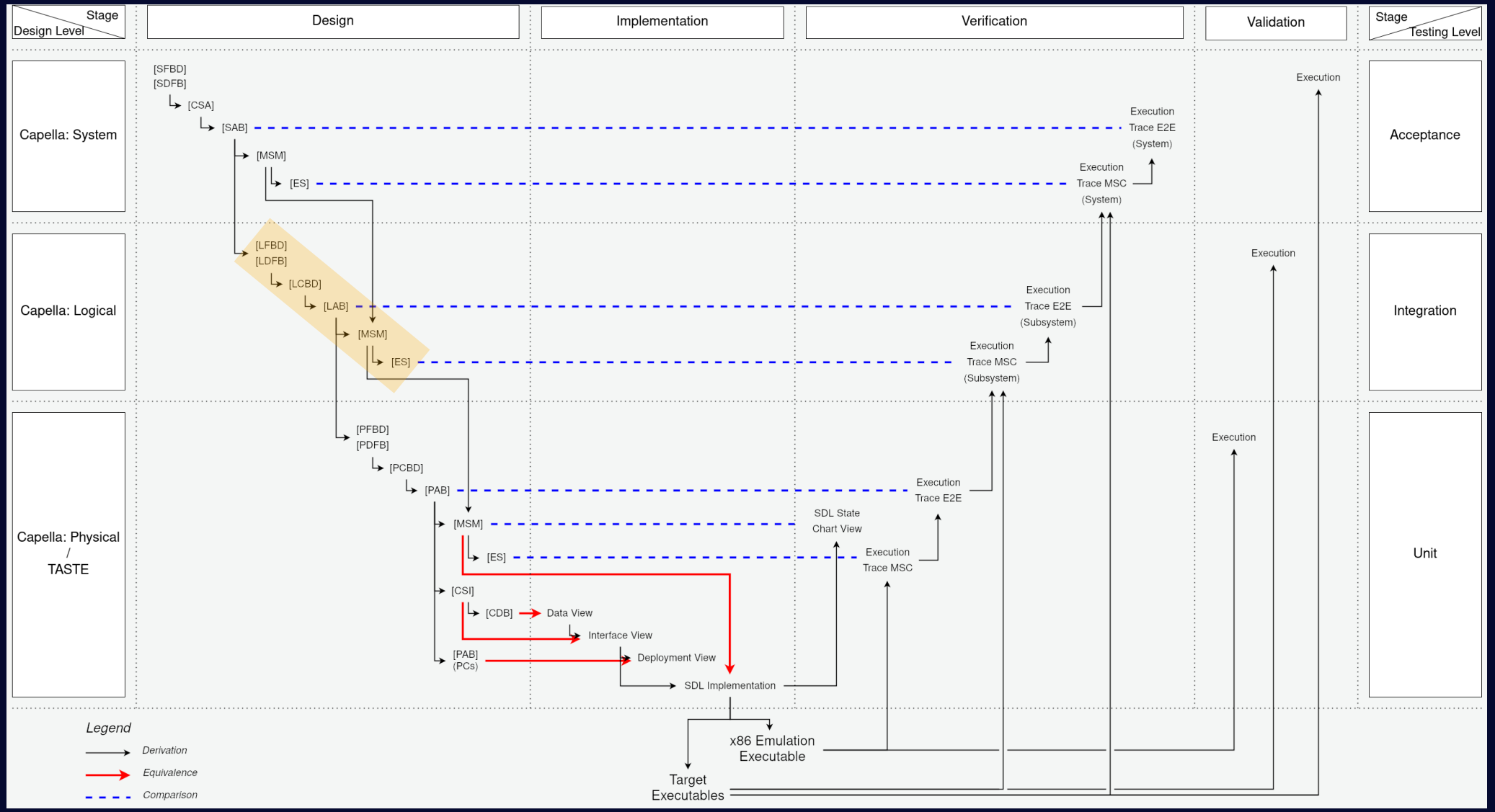
The Bridge: System Modelling



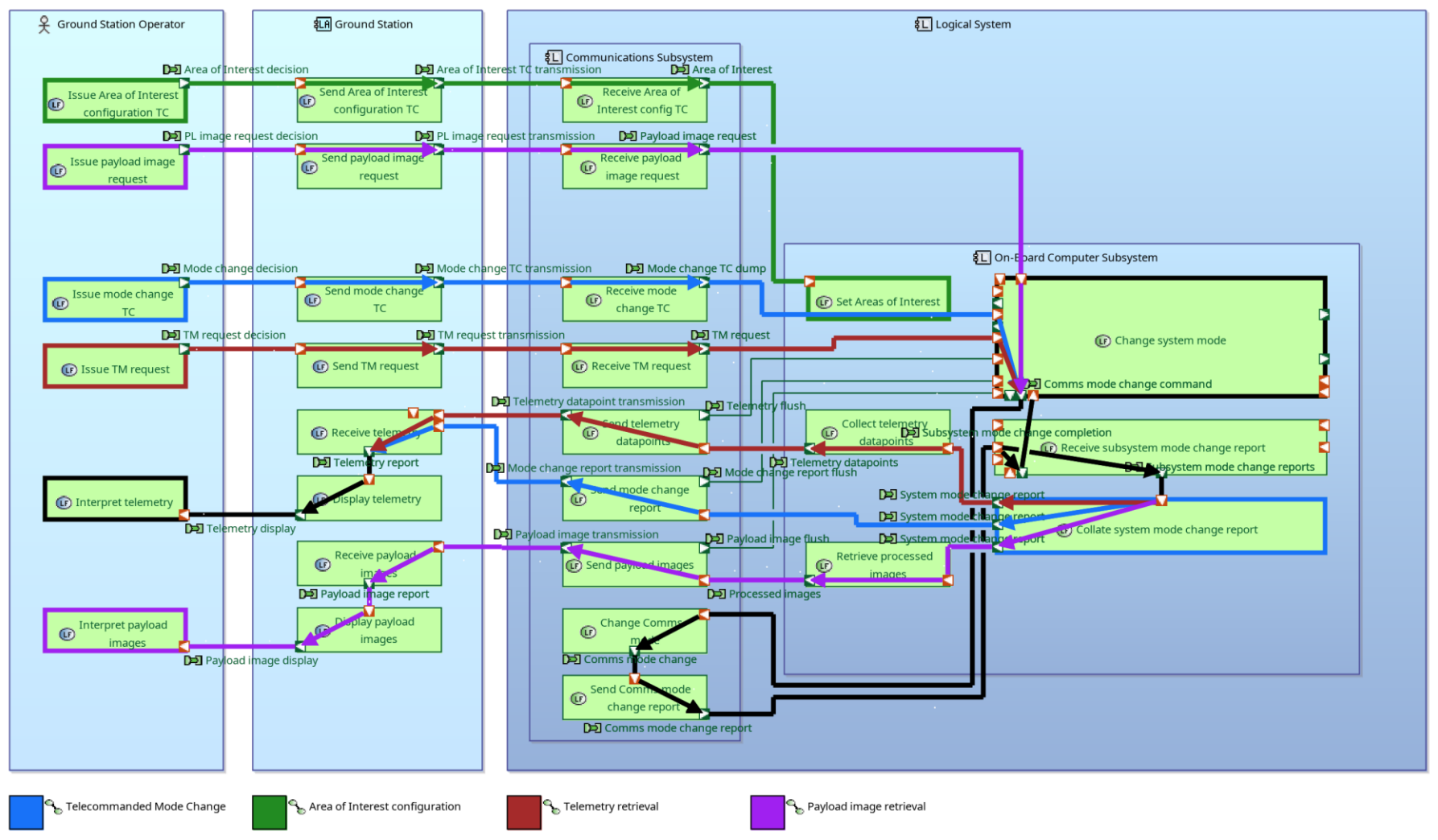
The Bridge: System Modelling



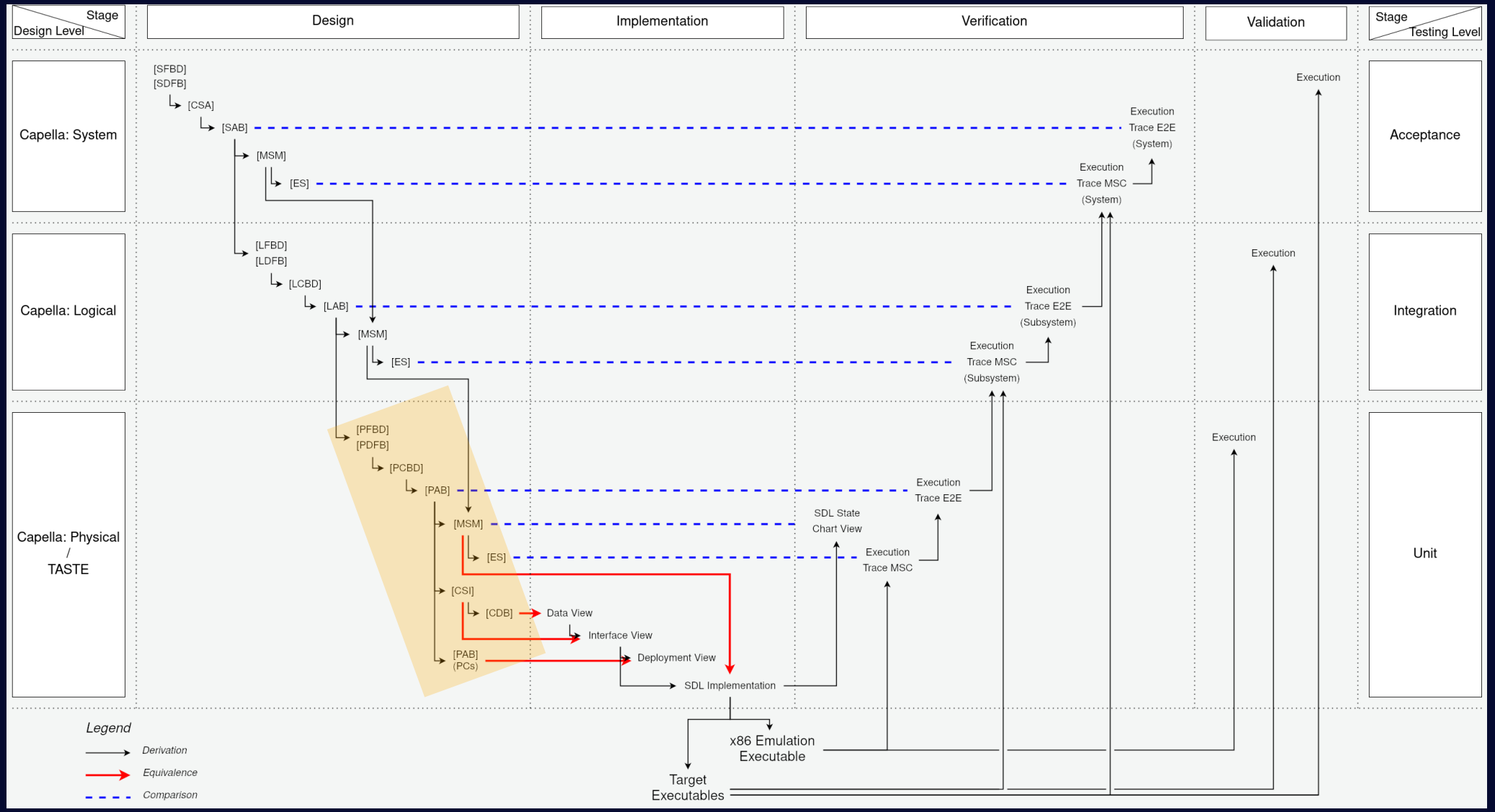
The Bridge: Logical Modelling



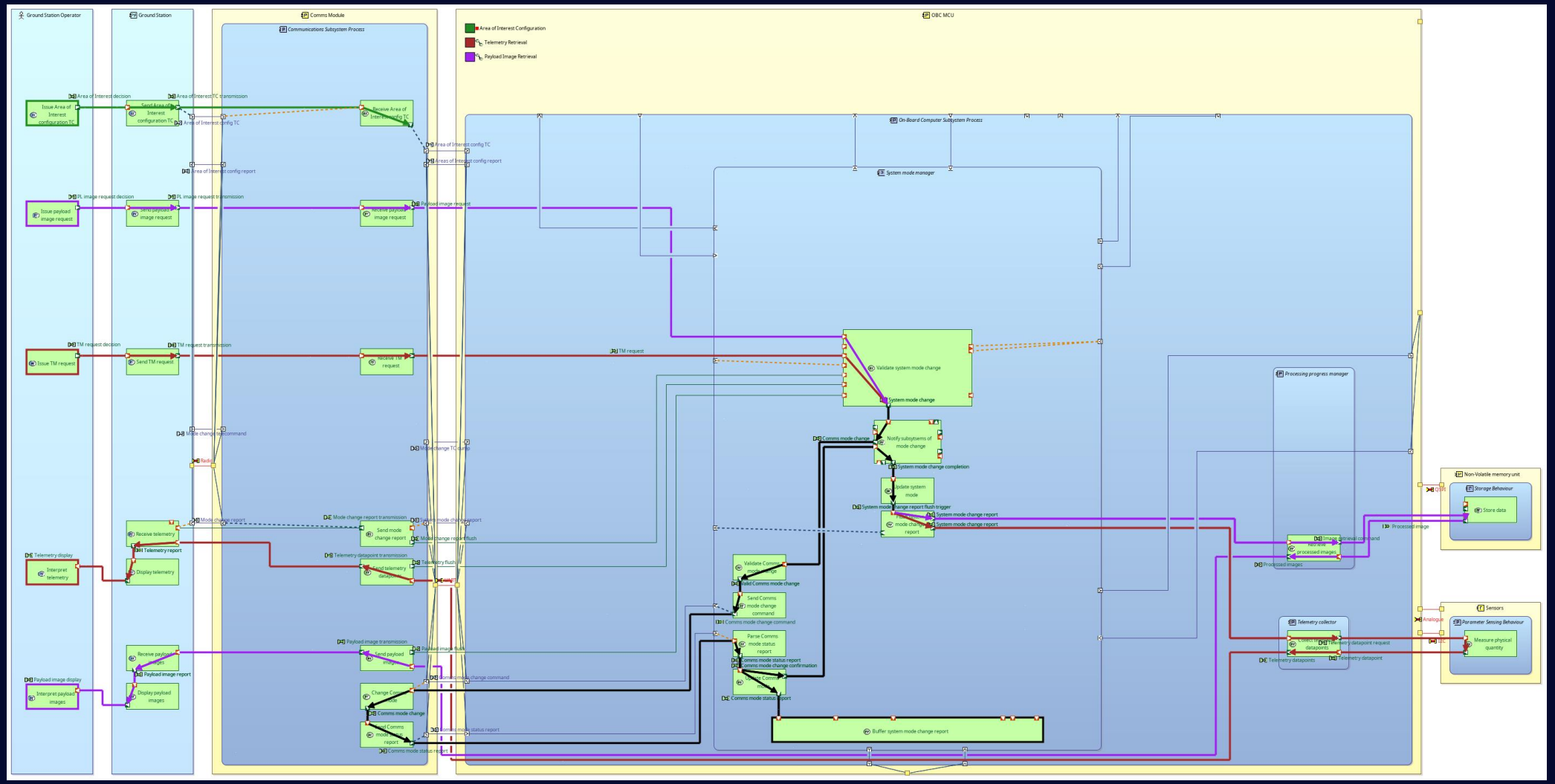
The Bridge: Logical Modelling



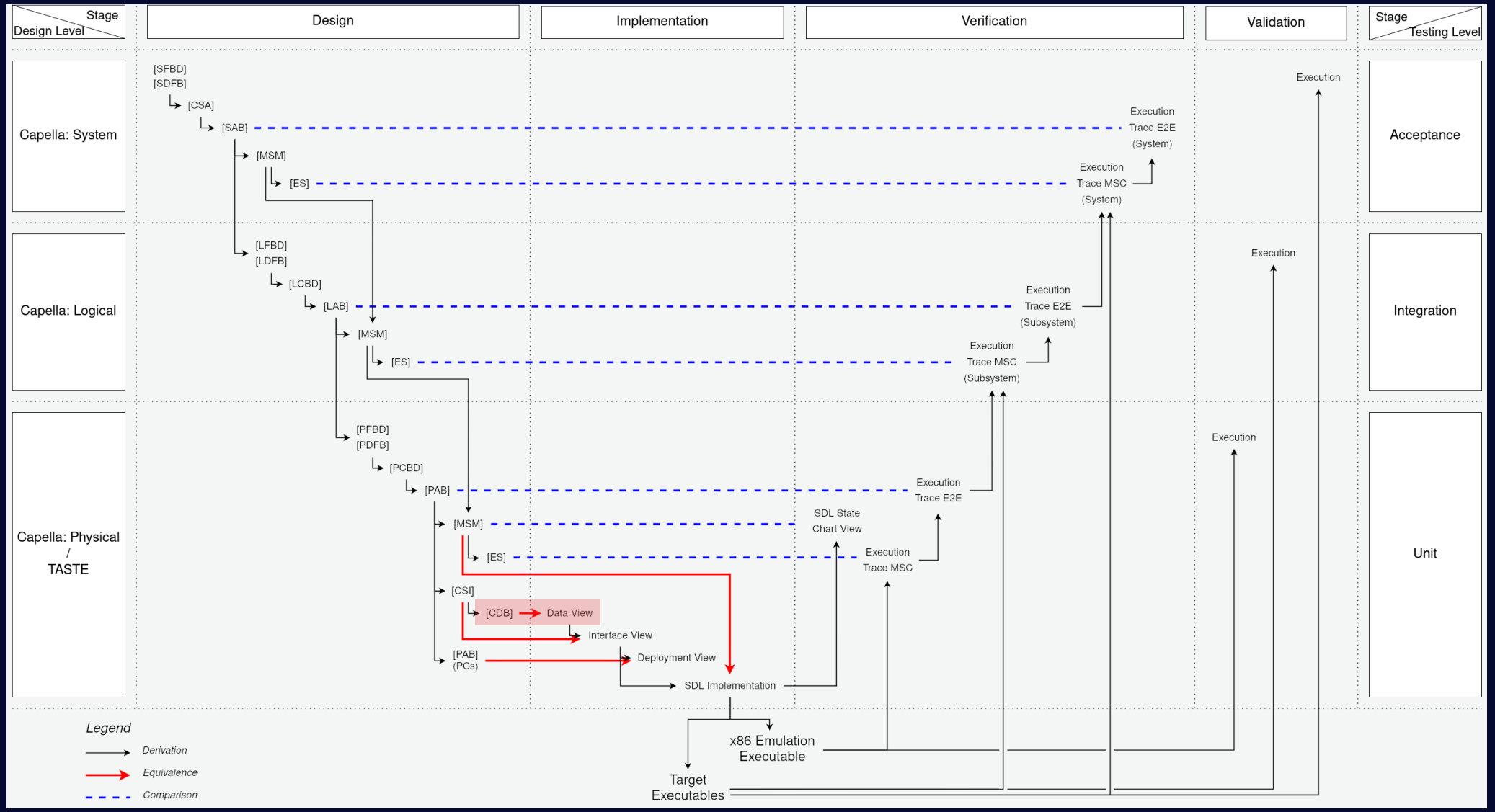
The Bridge: Physical Modelling



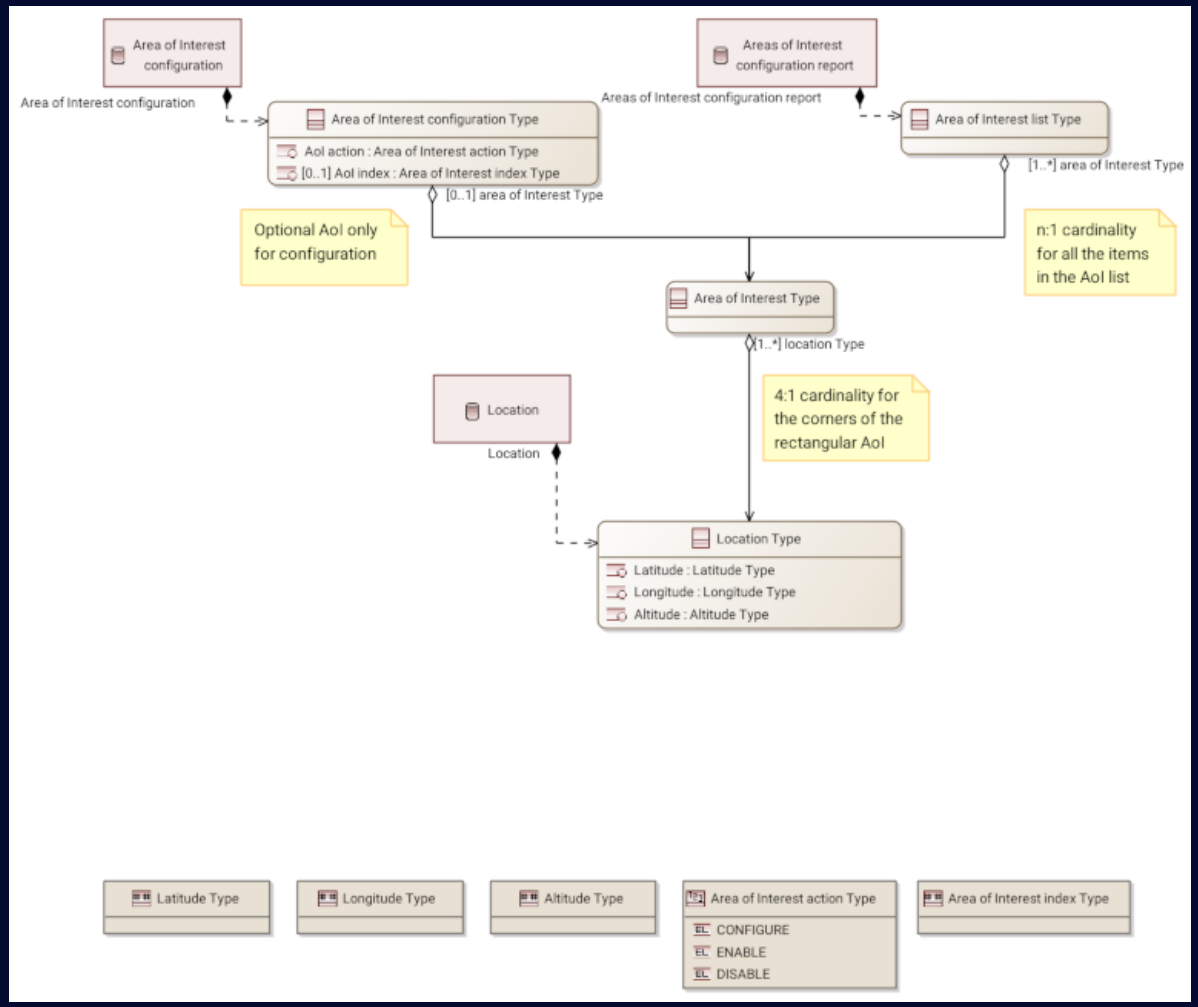
The Bridge: Physical Modelling



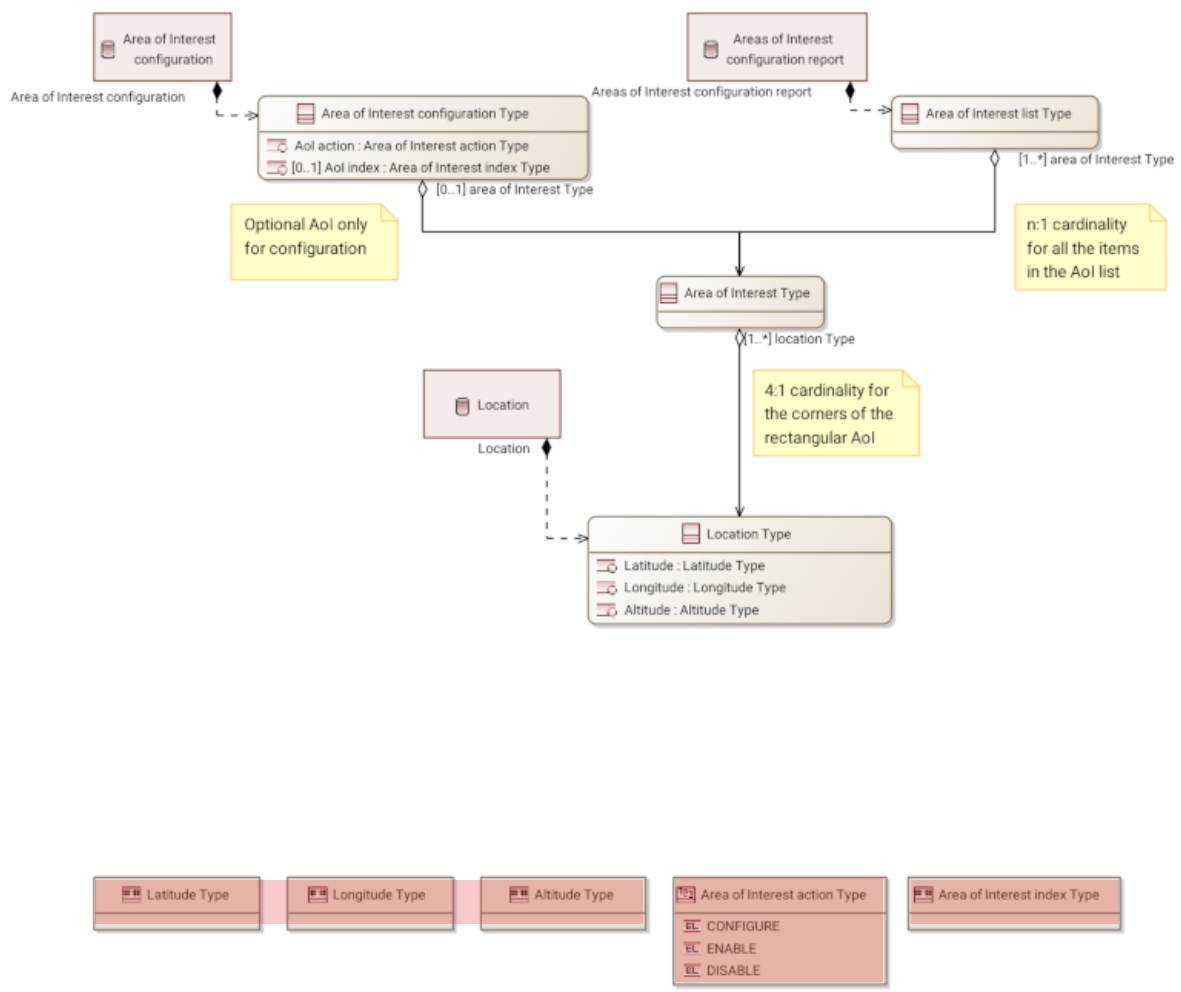
The Bridge: Data Implementation



The Bridge: Data Implementation

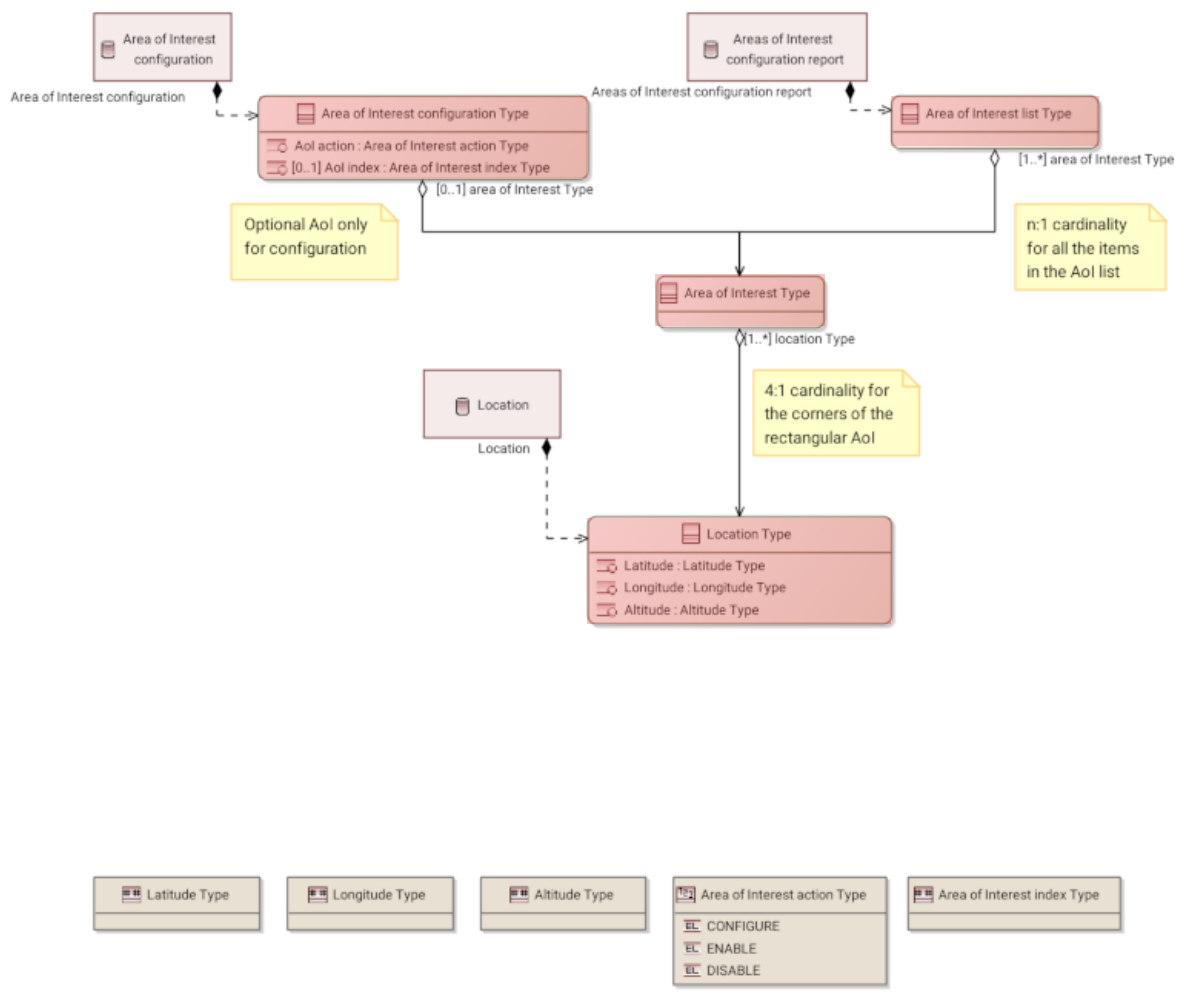


The Bridge: Data Implementation



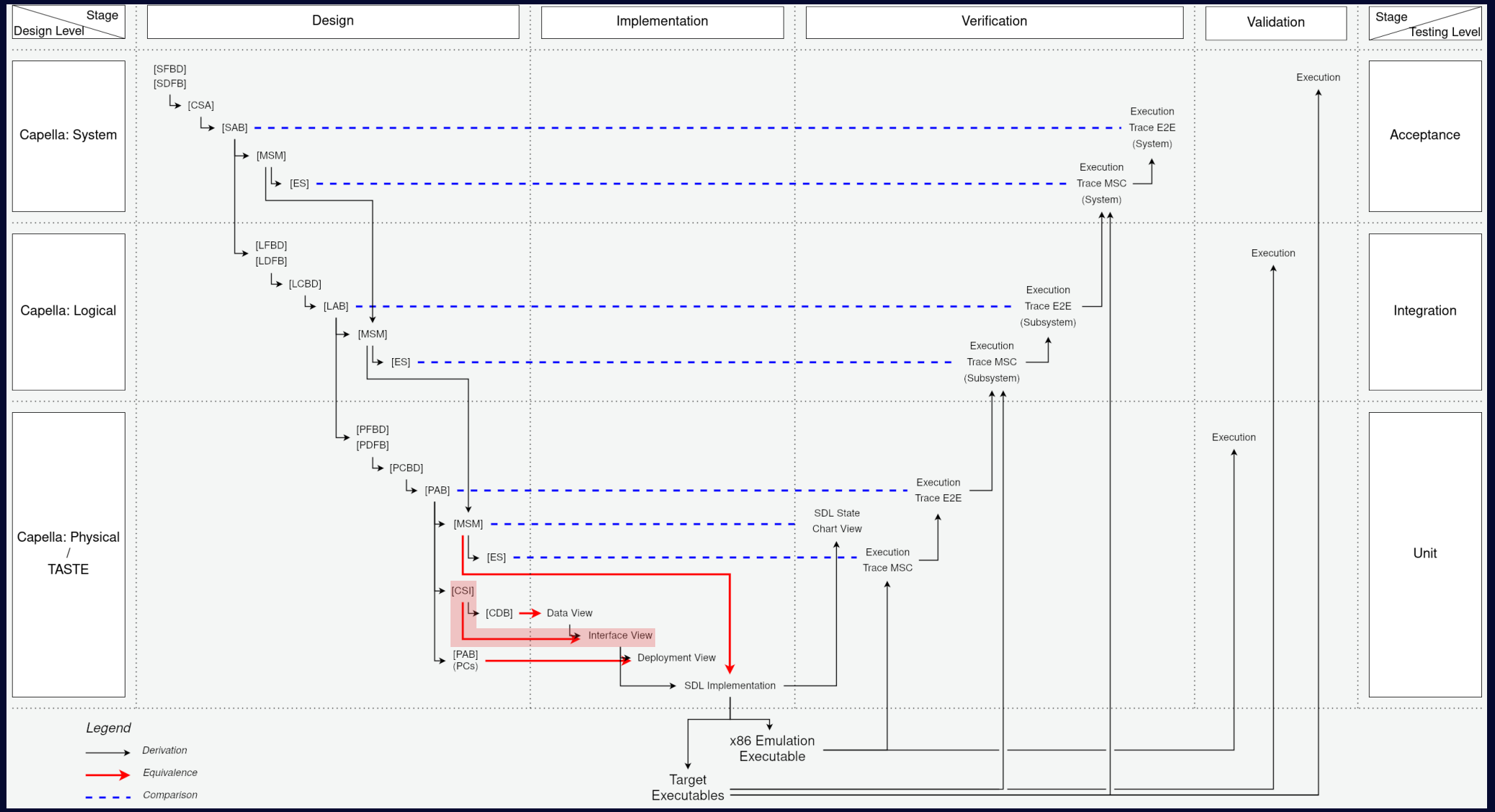

```
48  Latitude-WGS84          ::= REAL (-90.0 .. 90.0)
49  Longitude-WGS84         ::= REAL (-180.0 .. 180.0)
50  Altitude-m              ::= REAL (0.0 .. 100000.0)
51  Location                ::= SEQUENCE {
52      lat                 Latitude-WGS84,
53      lon                 Longitude-WGS84,
54      alt                 Altitude-m
55  }
56
57  Area-of-Interest        ::= SEQUENCE {
58      loc-nw              Location,
59      loc-ne              Location,
60      loc-se              Location,
61      loc-sw              Location,
62      enabled             BOOLEAN
63  }
64
65  AoI-List                ::= SEQUENCE (SIZE (4)) OF Area-of-Interest
66  AoI-List-Index          ::= INTEGER (0 .. 3)
67
68  AoI-Config-TC           ::= CHOICE {
69      configure           SEQUENCE {
70          index           AoI-List-Index,
71          aoi             Area-of-Interest
72      },
73      enable              SEQUENCE {
74          index           AoI-List-Index
75      },
76      disable             SEQUENCE {
77          index           AoI-List-Index
78      }
79  }
80
81  AoI-Config-Report       ::= SEQUENCE {
82      aoi-list-upd        AoI-List
83  }
```

The Bridge: Data Implementation

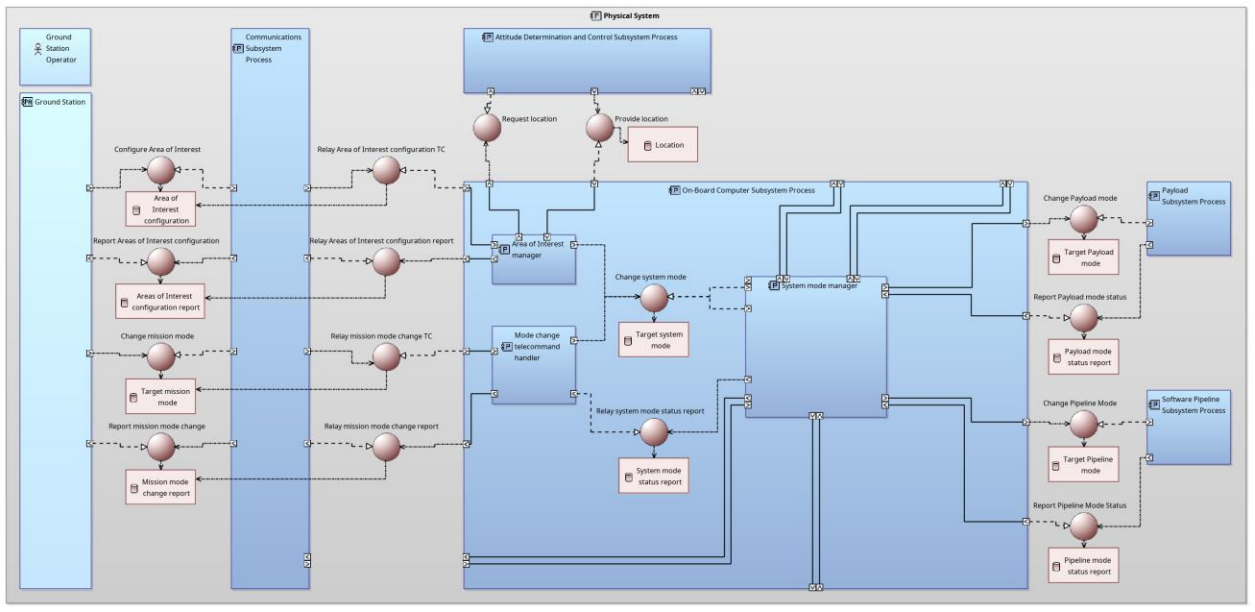



```
48  Latitude-WGS84          ::= REAL (-90.0 .. 90.0)
49  Longitude-WGS84         ::= REAL (-180.0 .. 180.0)
50  Altitude-m              ::= REAL (0.0 .. 100000.0)
51  Location                ::= SEQUENCE {
52      lat                 Latitude-WGS84,
53      lon                 Longitude-WGS84,
54      alt                 Altitude-m
55  }
56
57  Area-of-Interest        ::= SEQUENCE {
58      loc-nw              Location,
59      loc-ne              Location,
60      loc-se              Location,
61      loc-sw              Location,
62      enabled             BOOLEAN
63  }
64
65  AoI-List                ::= SEQUENCE (SIZE (4)) OF Area-of-Interest
66  AoI-List-Index          ::= INTEGER (0 .. 3)
67
68  AoI-Config-TC           ::= CHOICE {
69      configure           SEQUENCE {
70          index           AoI-List-Index,
71          aoi             Area-of-Interest
72      },
73      enable              SEQUENCE {
74          index           AoI-List-Index
75      },
76      disable             SEQUENCE {
77          index           AoI-List-Index
78      }
79  }
80
81  AoI-Config-Report       ::= SEQUENCE {
82      aoi-list-upd        AoI-List
83  }
```

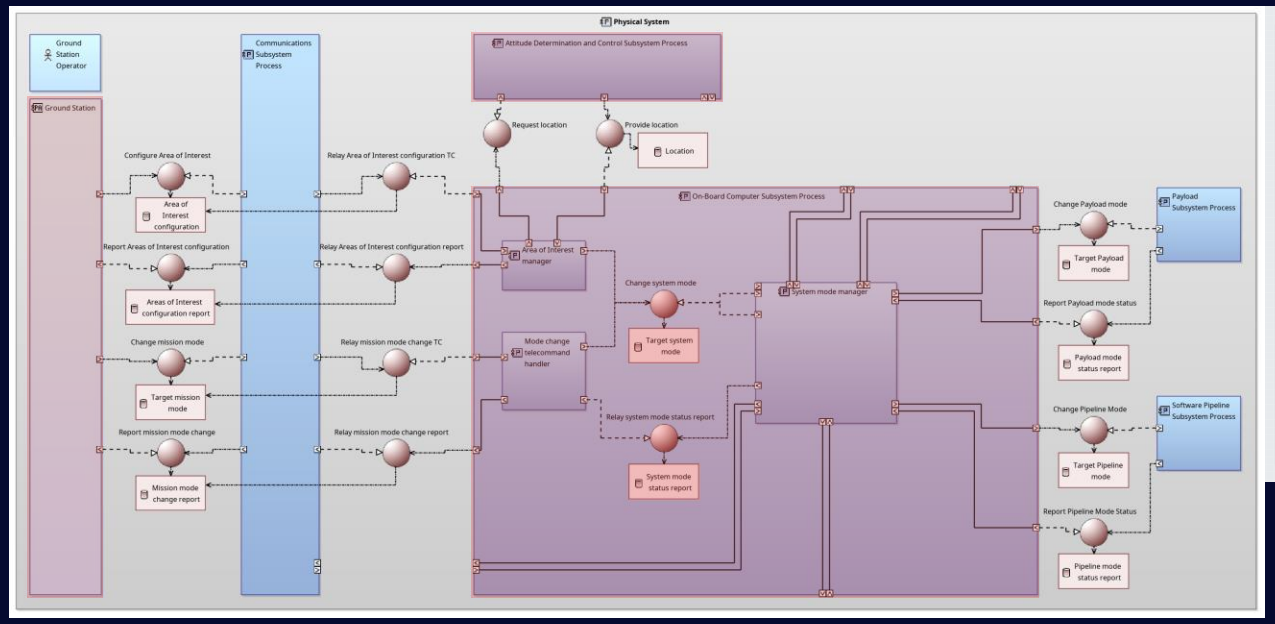
The Bridge: Interface Implementation

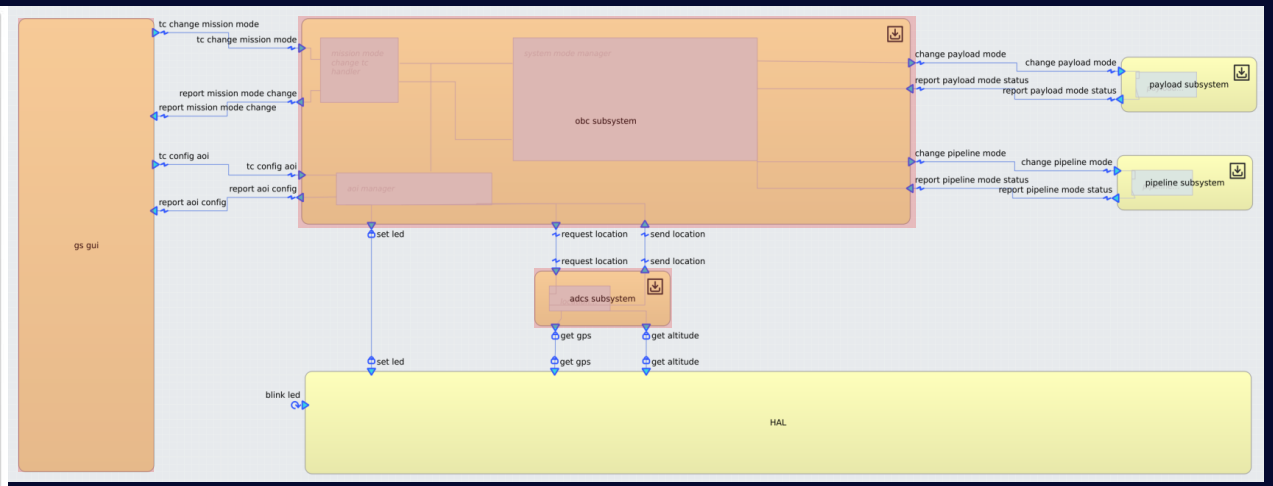


The Bridge: Interface Implementation

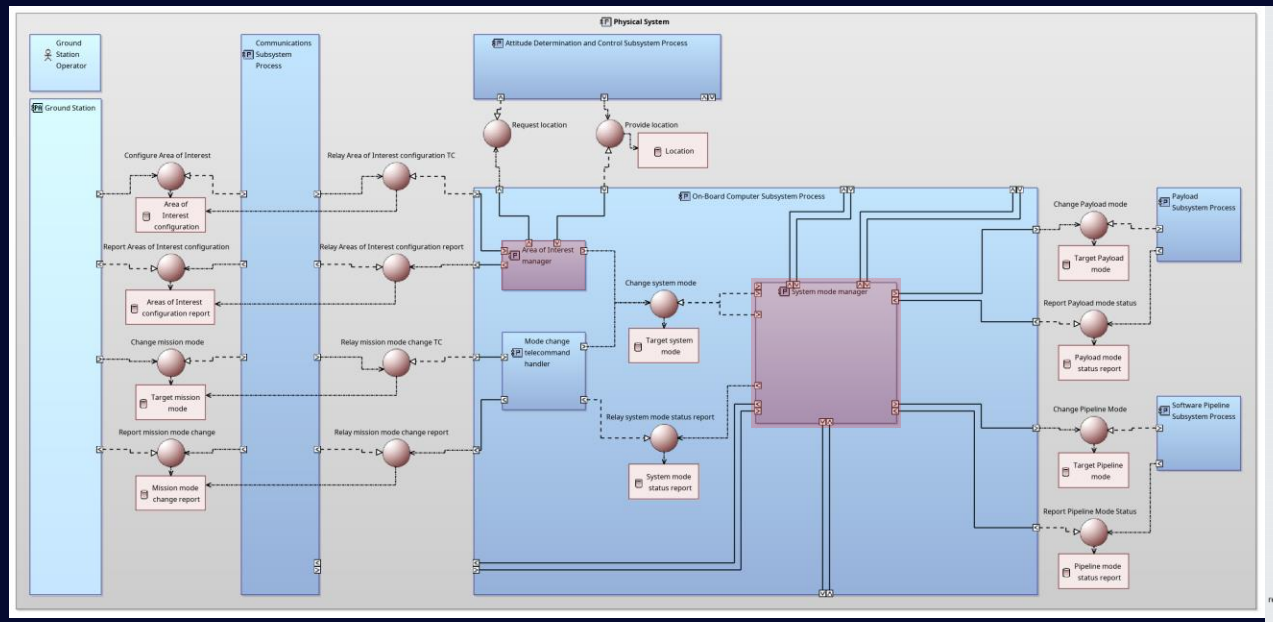


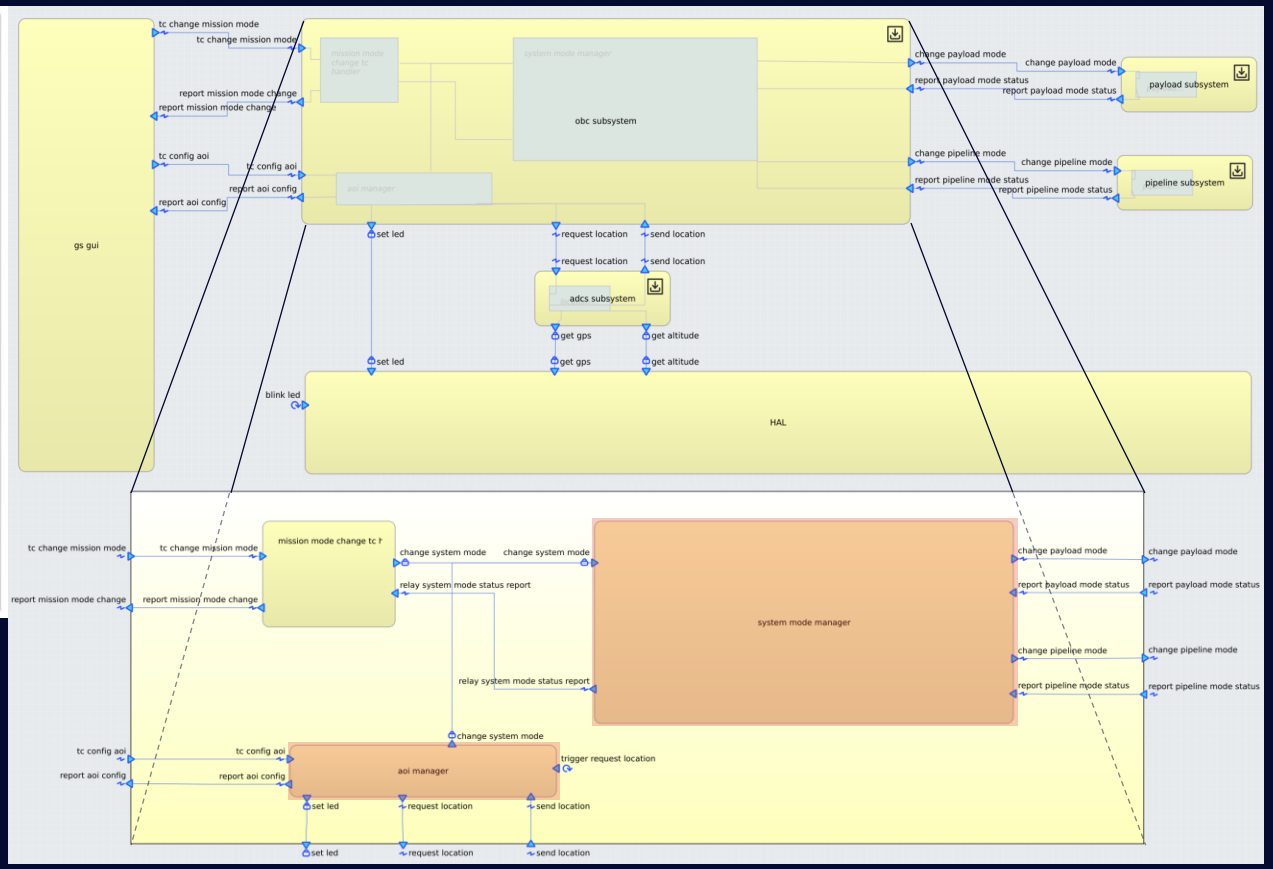
The Bridge: Interface Implementation



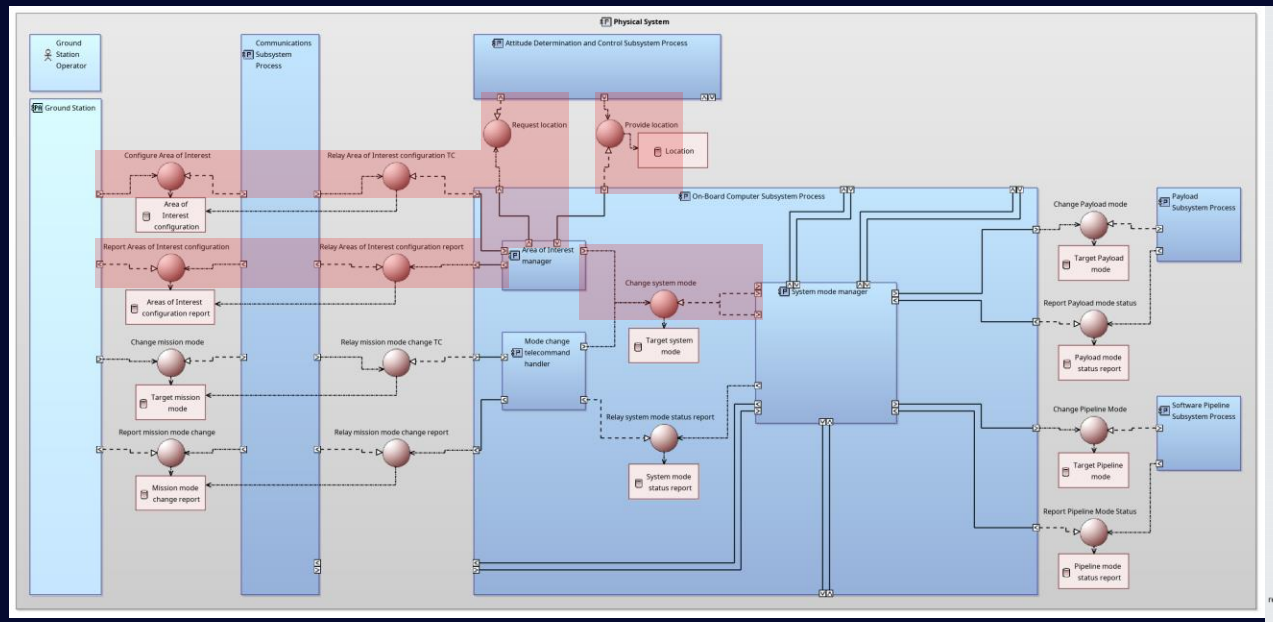


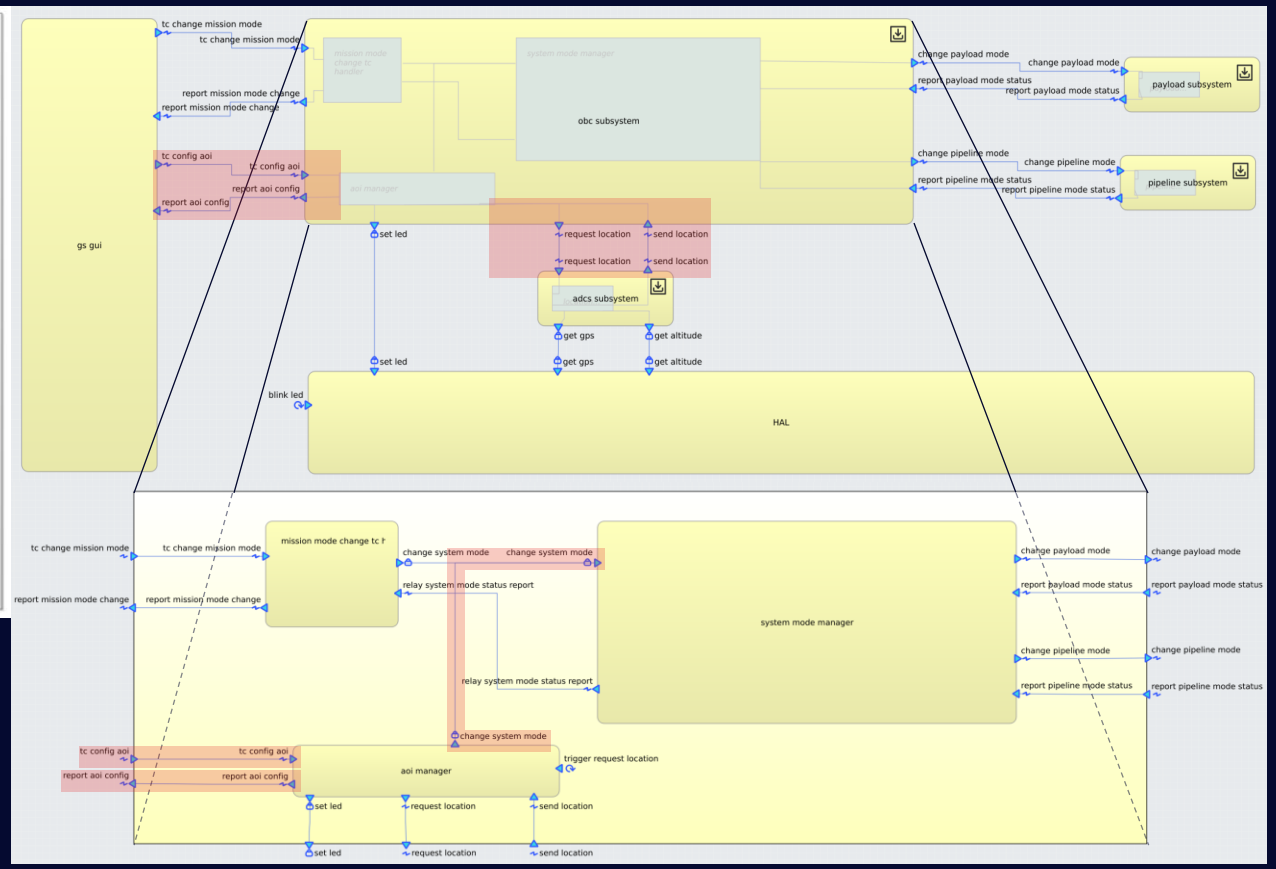
The Bridge: Interface Implementation



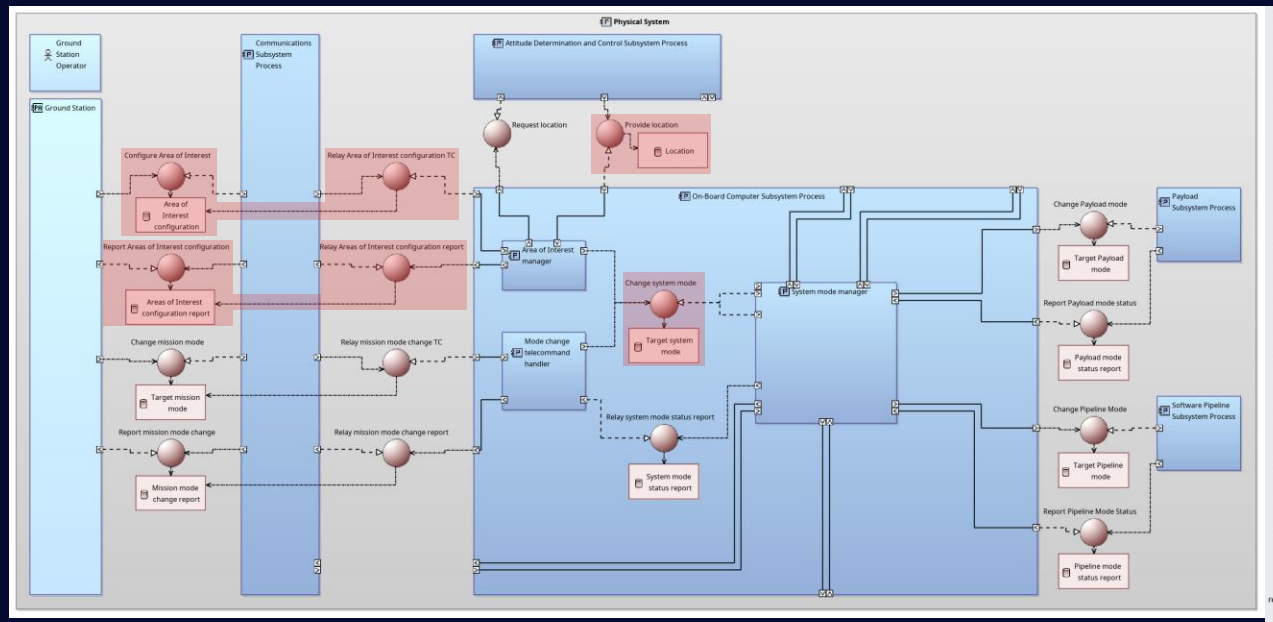


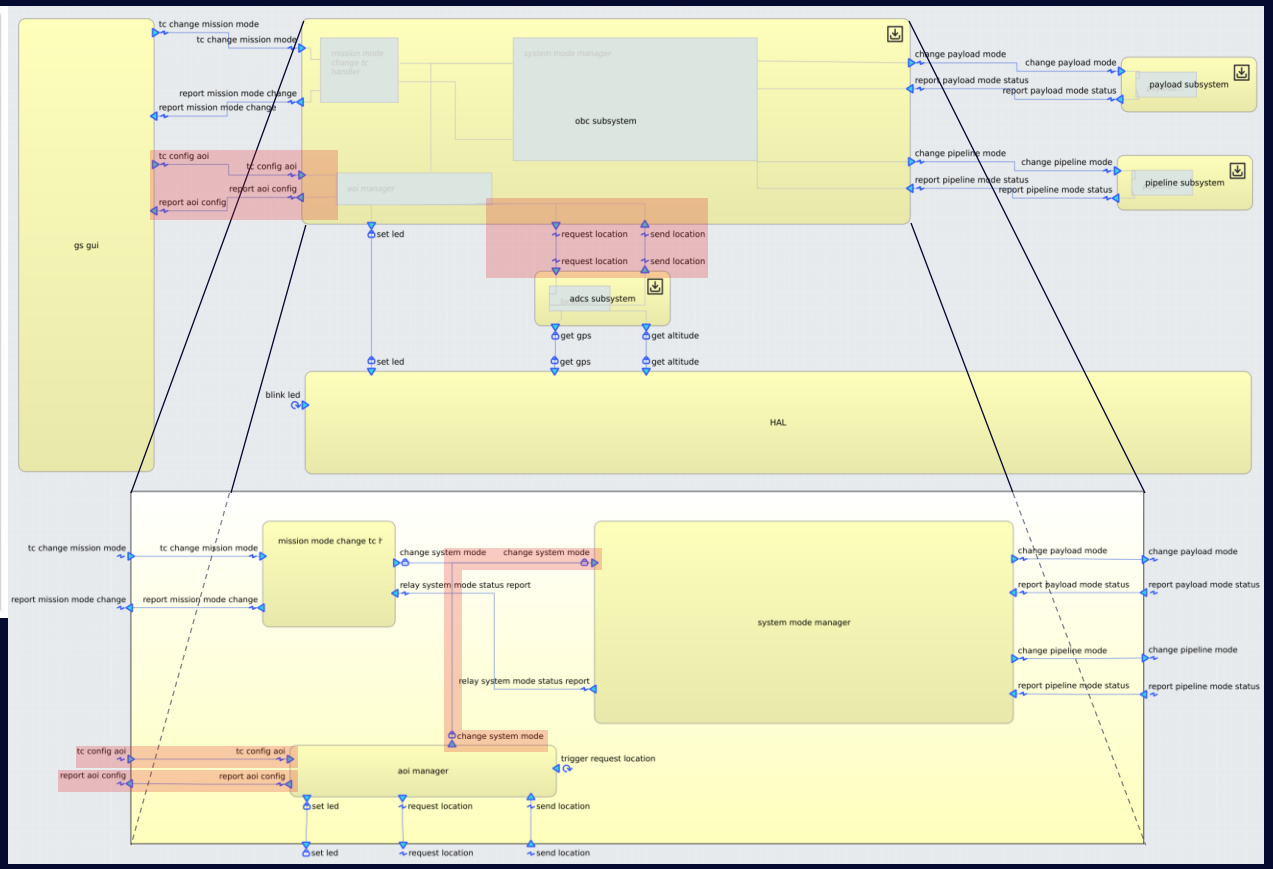
The Bridge: Interface Implementation



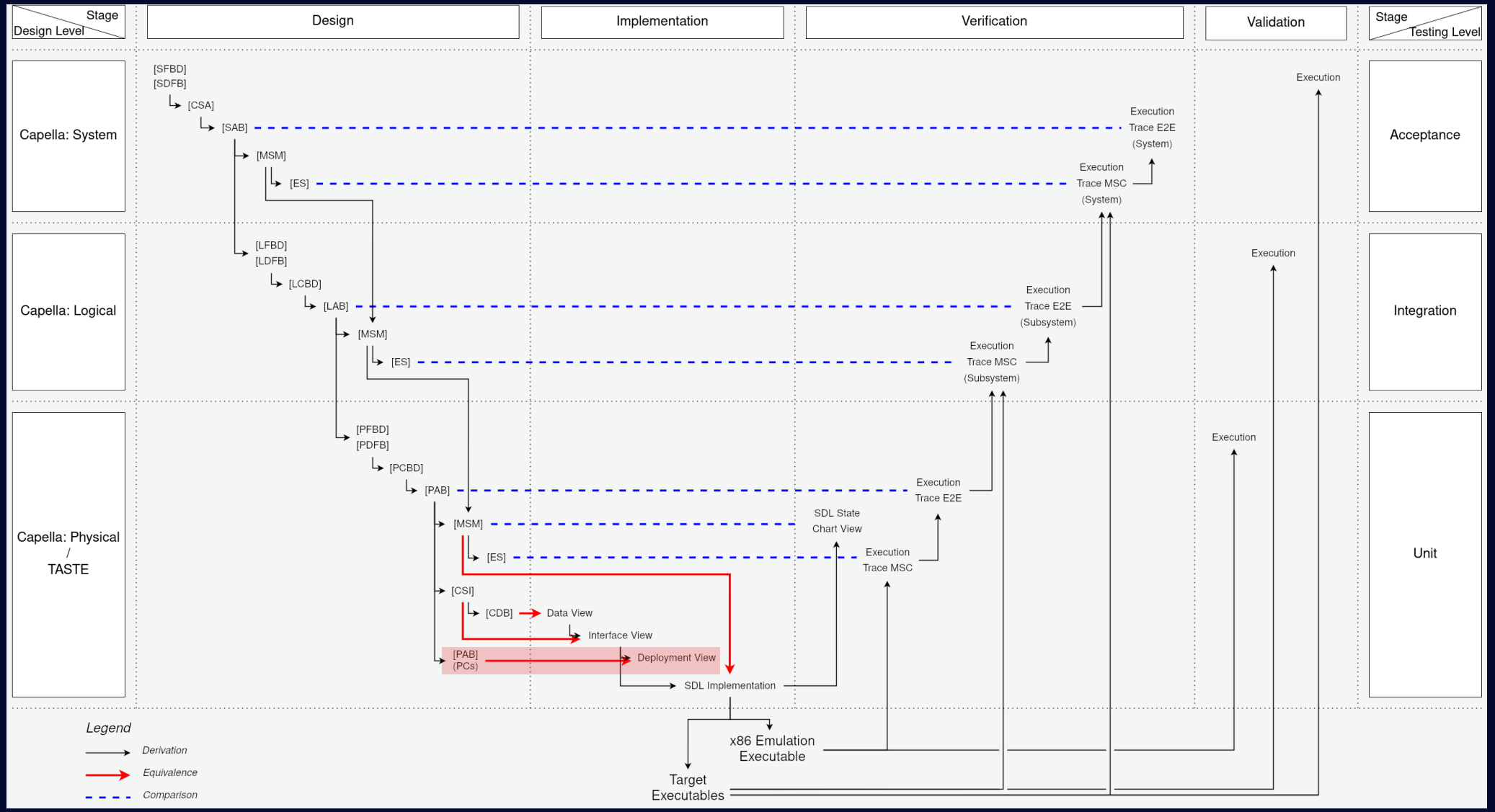


The Bridge: Interface Implementation

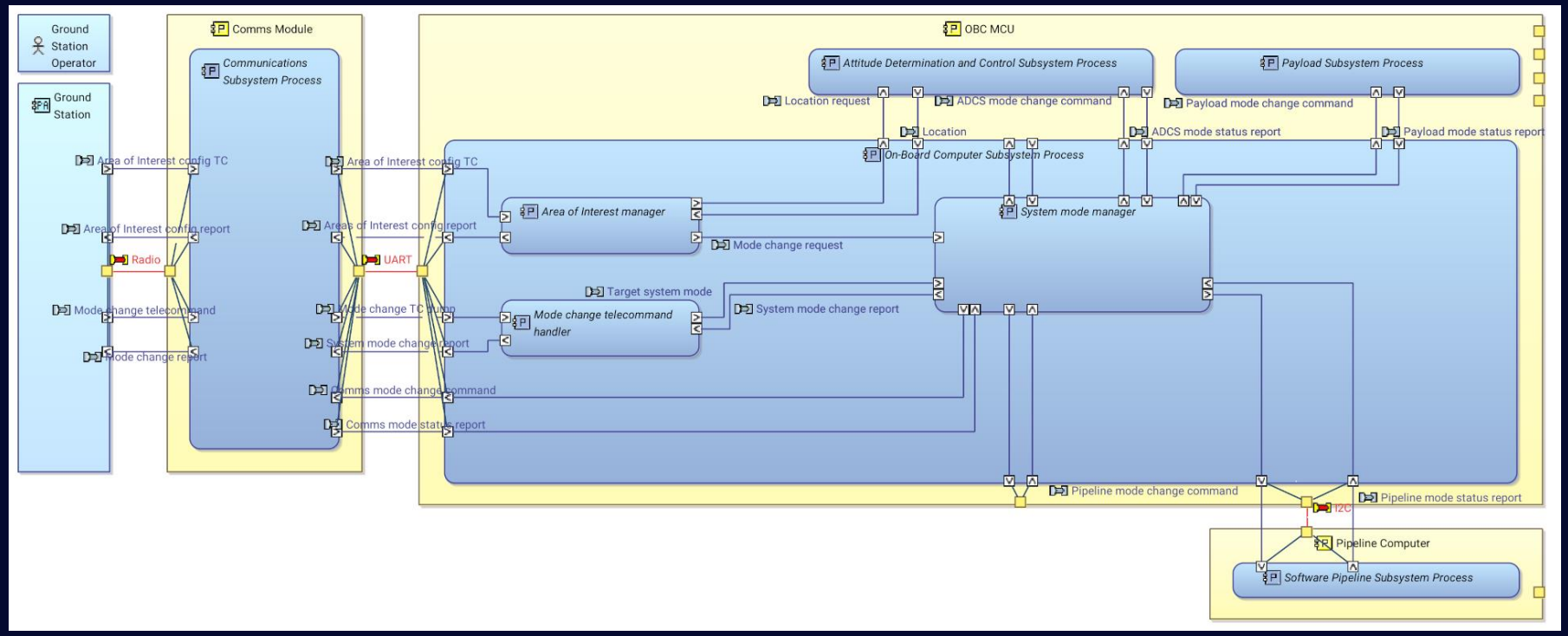




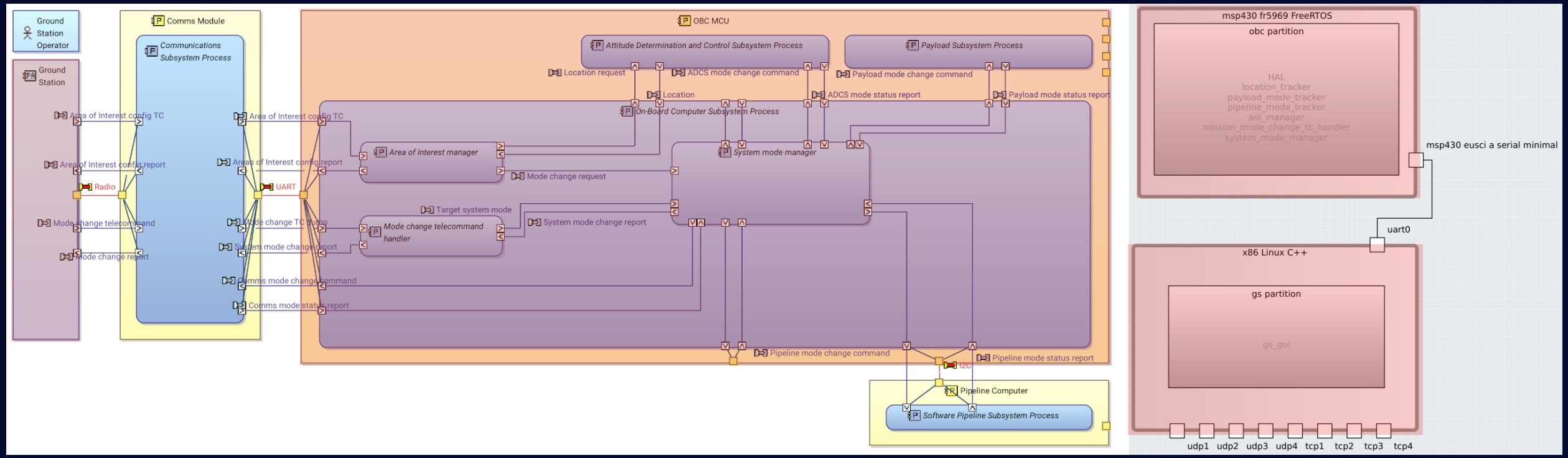
The Bridge: Deployment Implementation



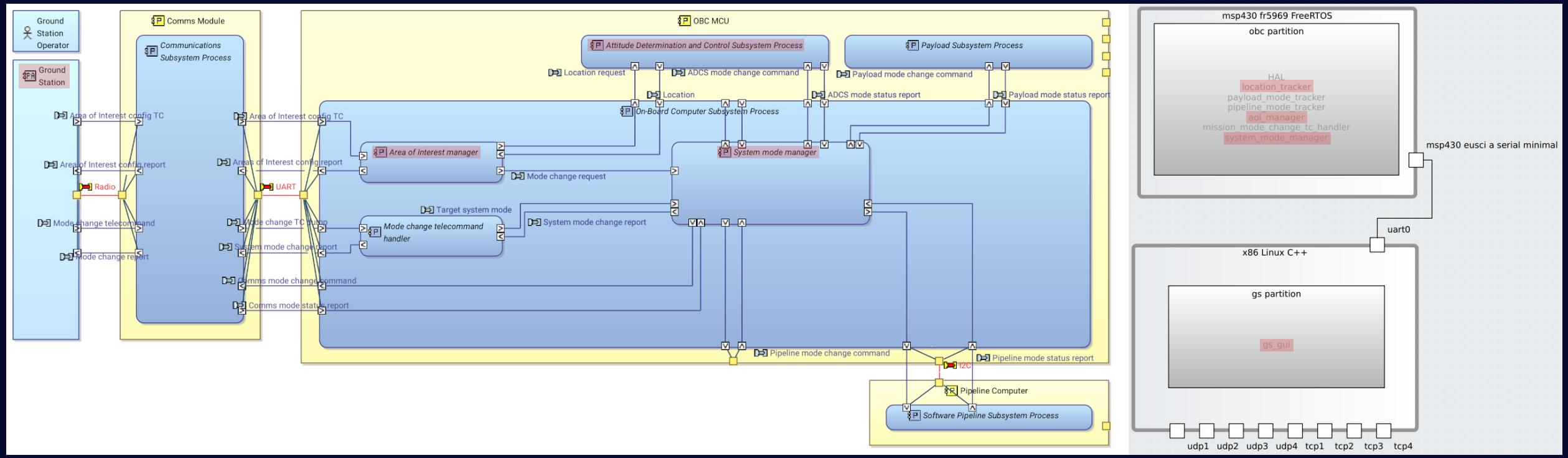
The Bridge: Deployment Implementation



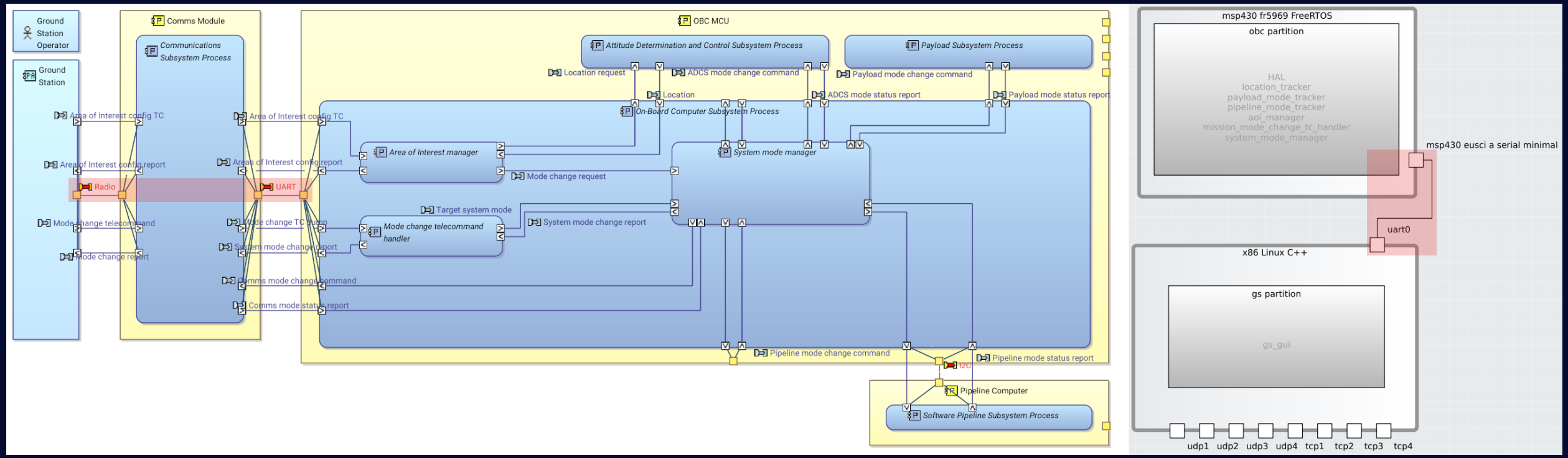
The Bridge: Deployment Implementation



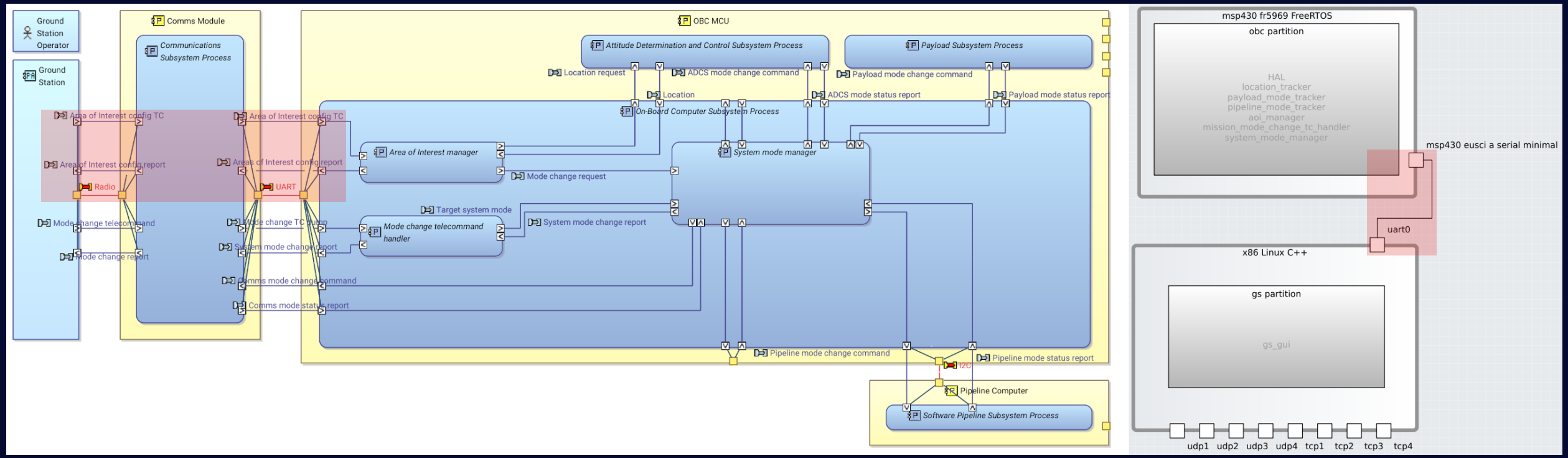
The Bridge: Deployment Implementation



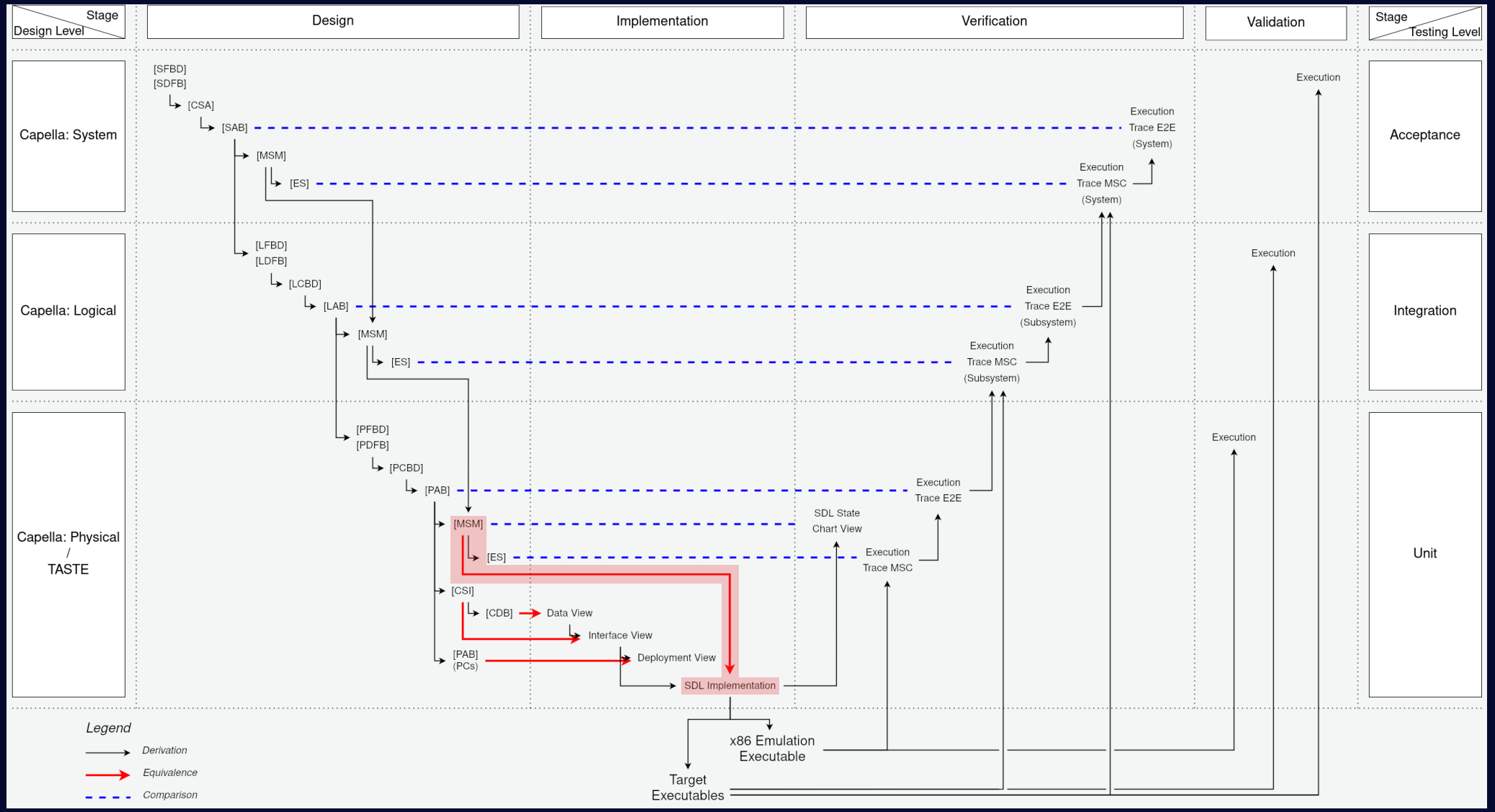
The Bridge: Deployment Implementation



The Bridge: Deployment Implementation

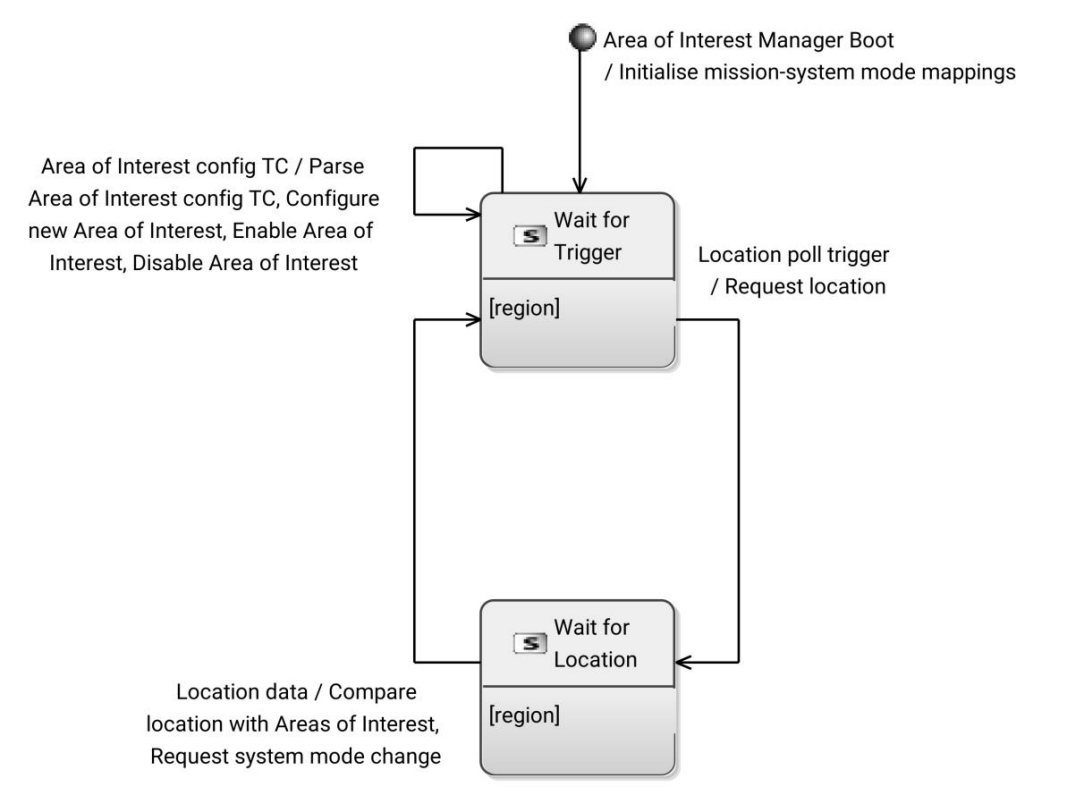


The Bridge: Behaviour Implementations



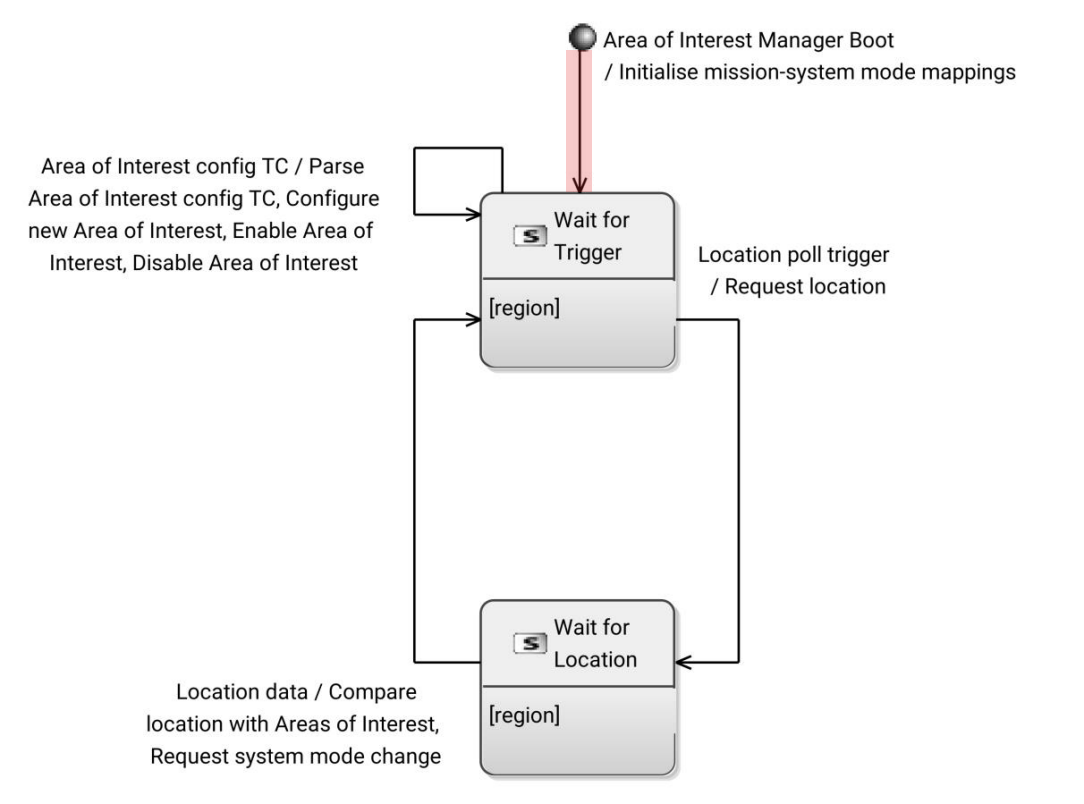
The Bridge: Behaviour Implementations

Capella:
Physical [MSM]



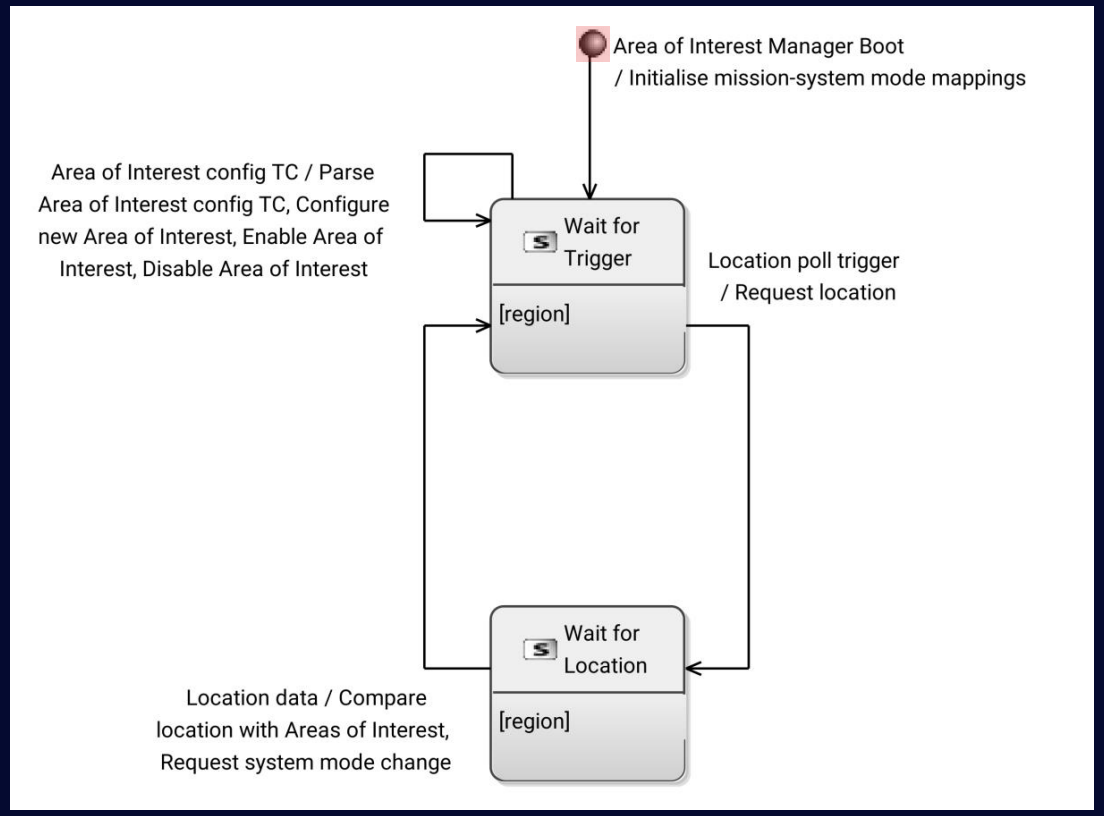
The Bridge: Behaviour Implementations

Capella:
Physical [MSM]

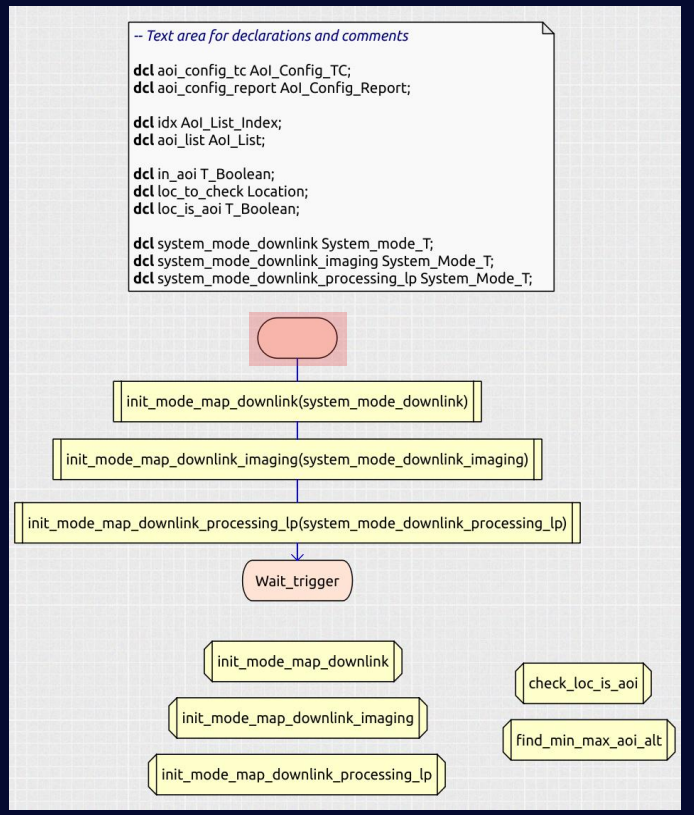


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

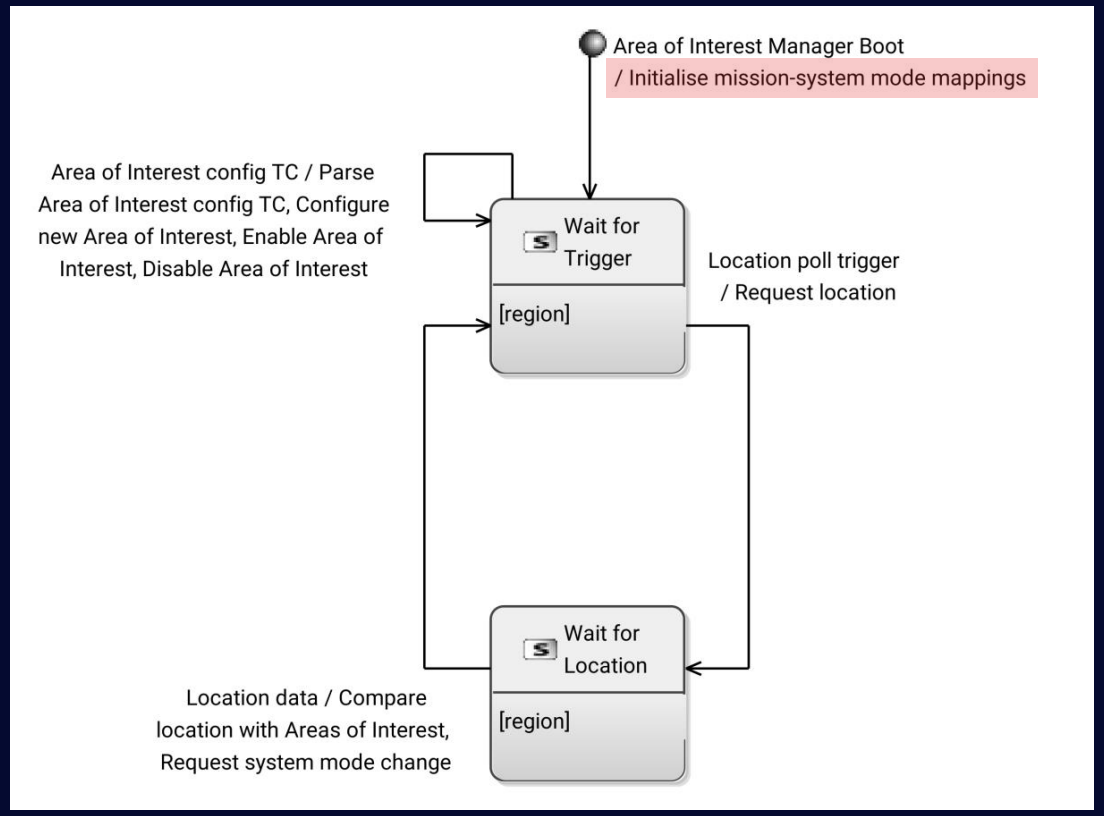


TASTE: SDL implementation

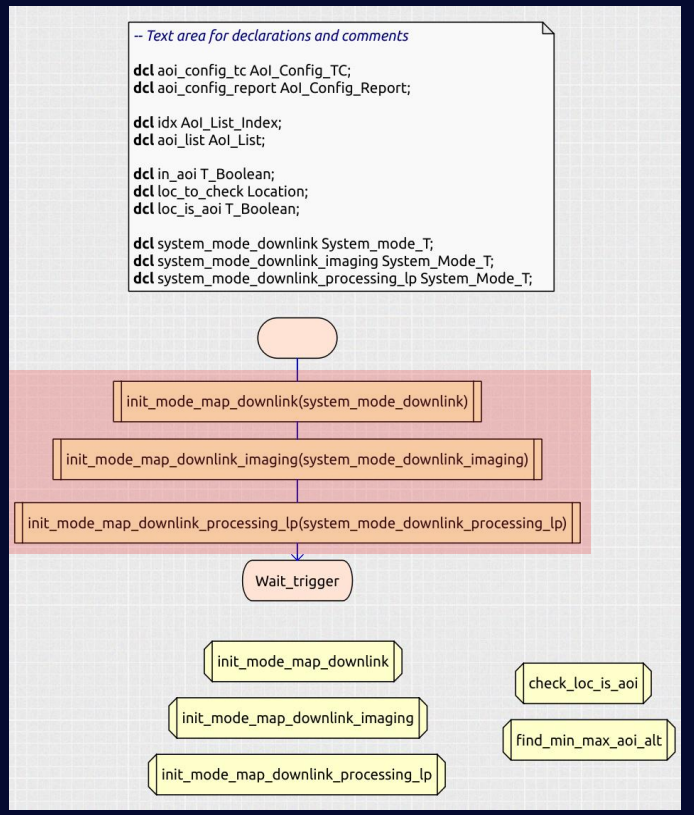


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

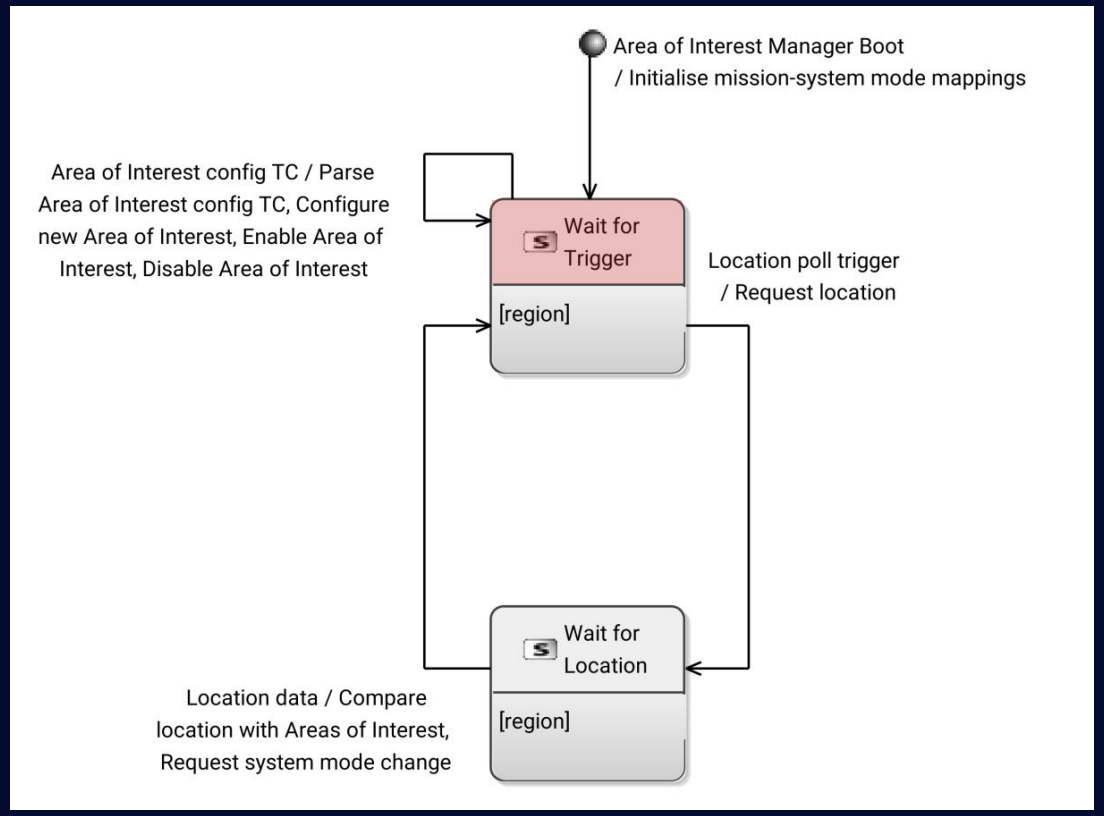


TASTE: SDL implementation

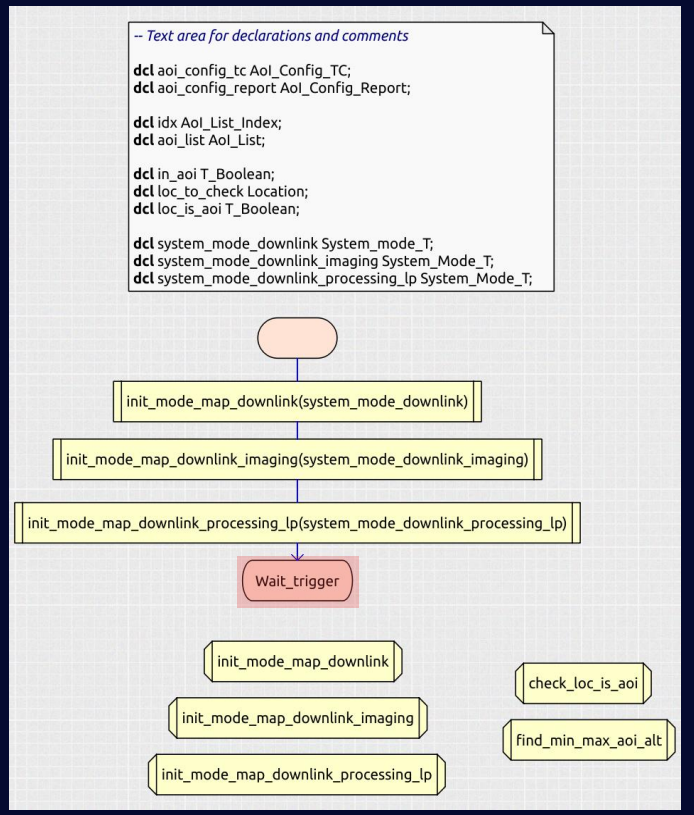


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

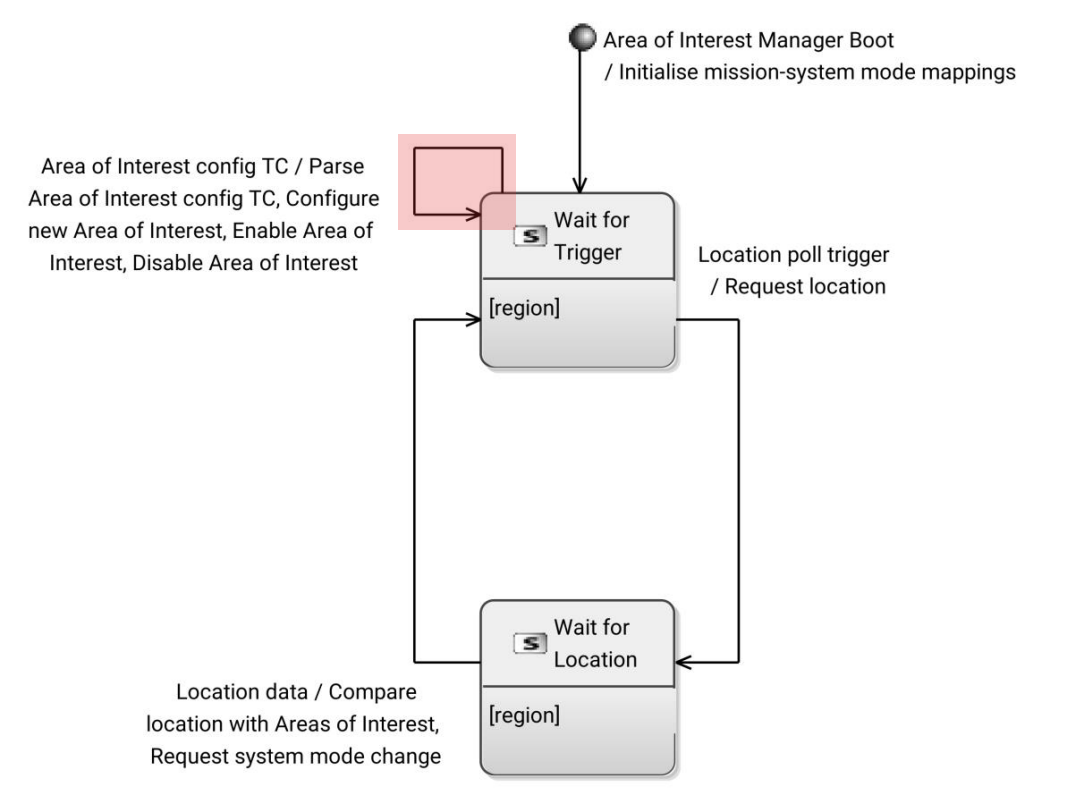


TASTE: SDL implementation



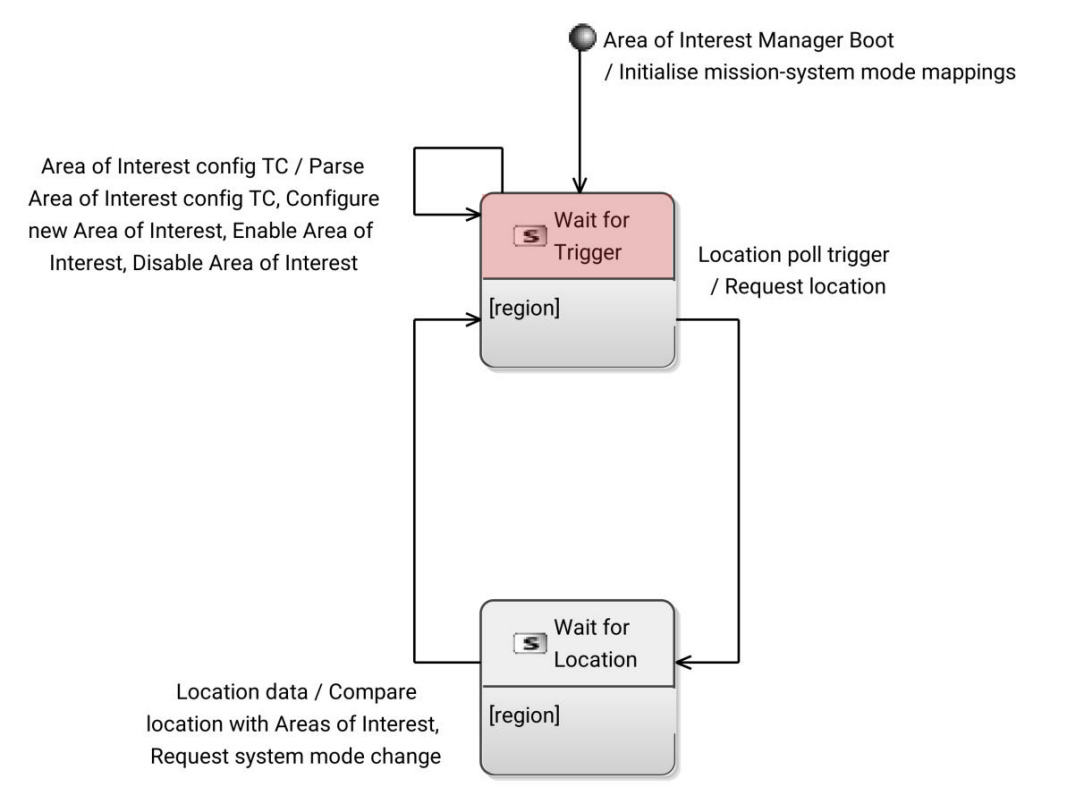
The Bridge: Behaviour Implementations

Capella:
Physical [MSM]

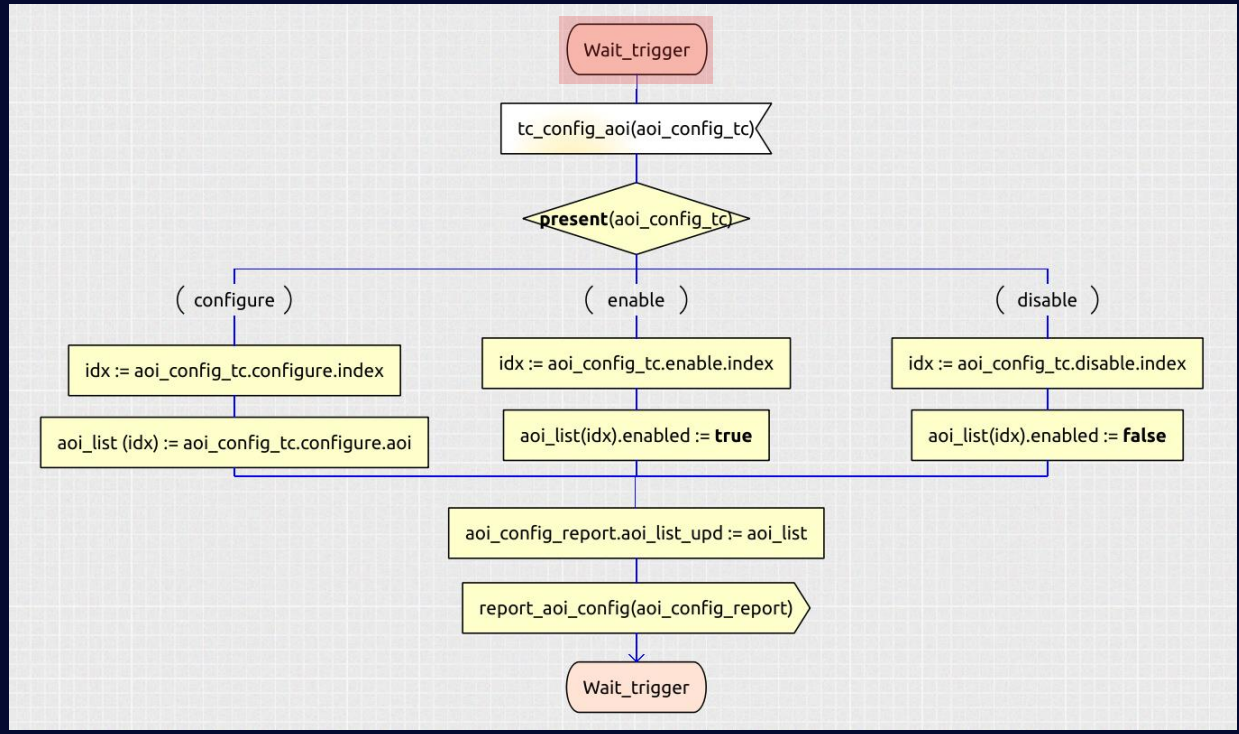


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

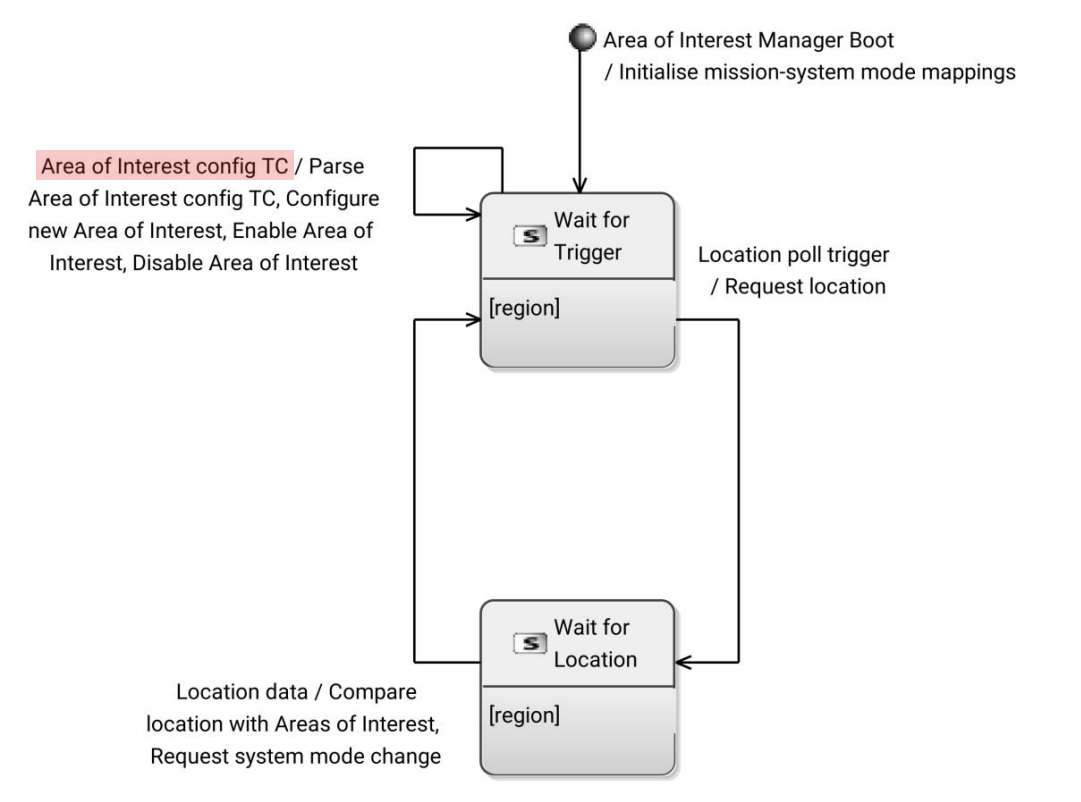


TASTE: SDL implementation

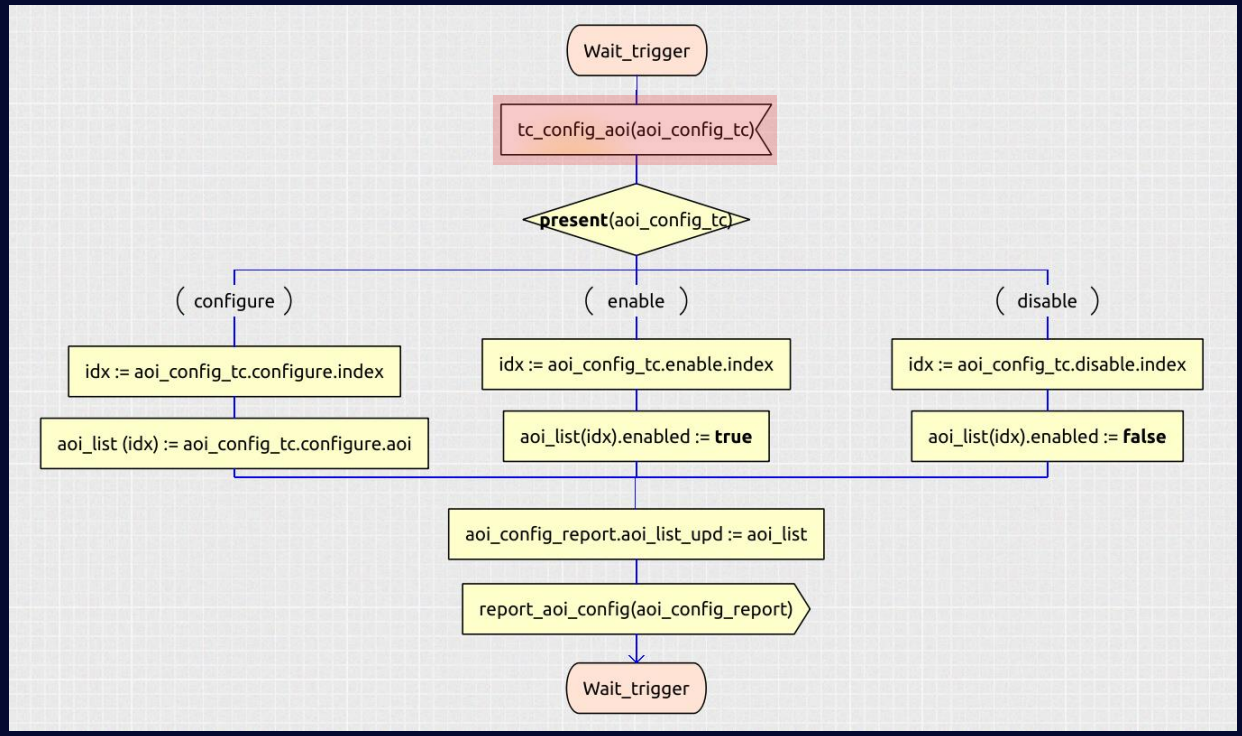


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

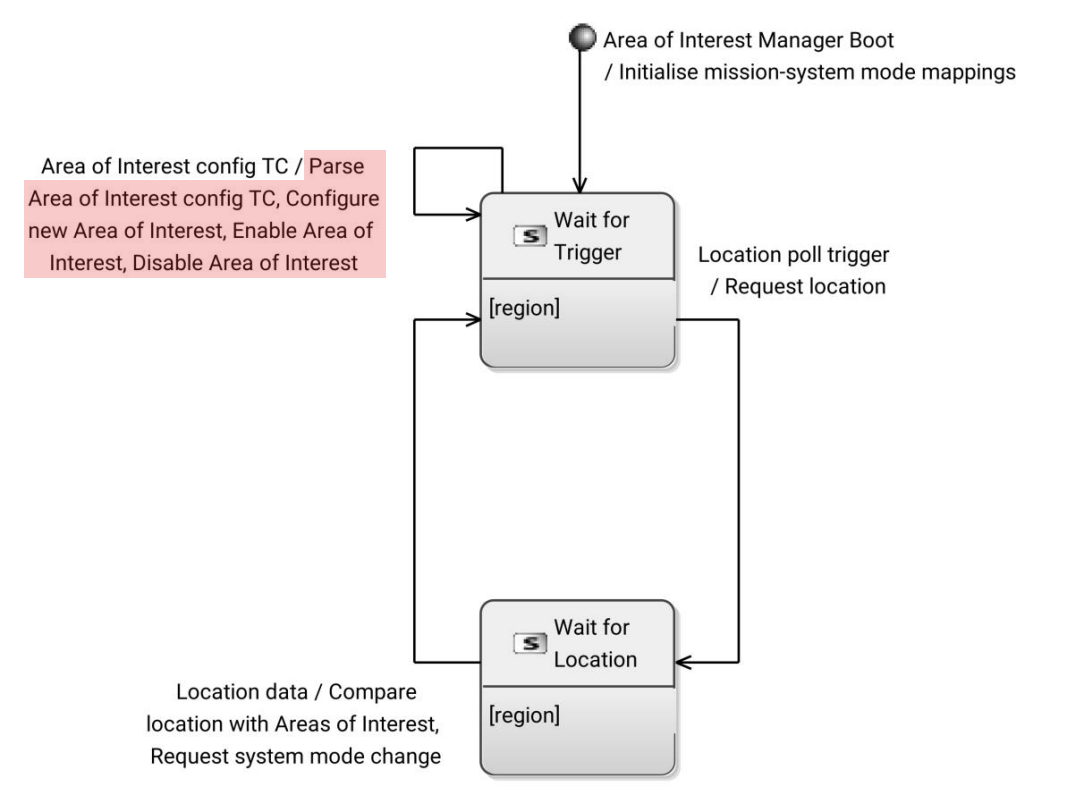


TASTE: SDL implementation

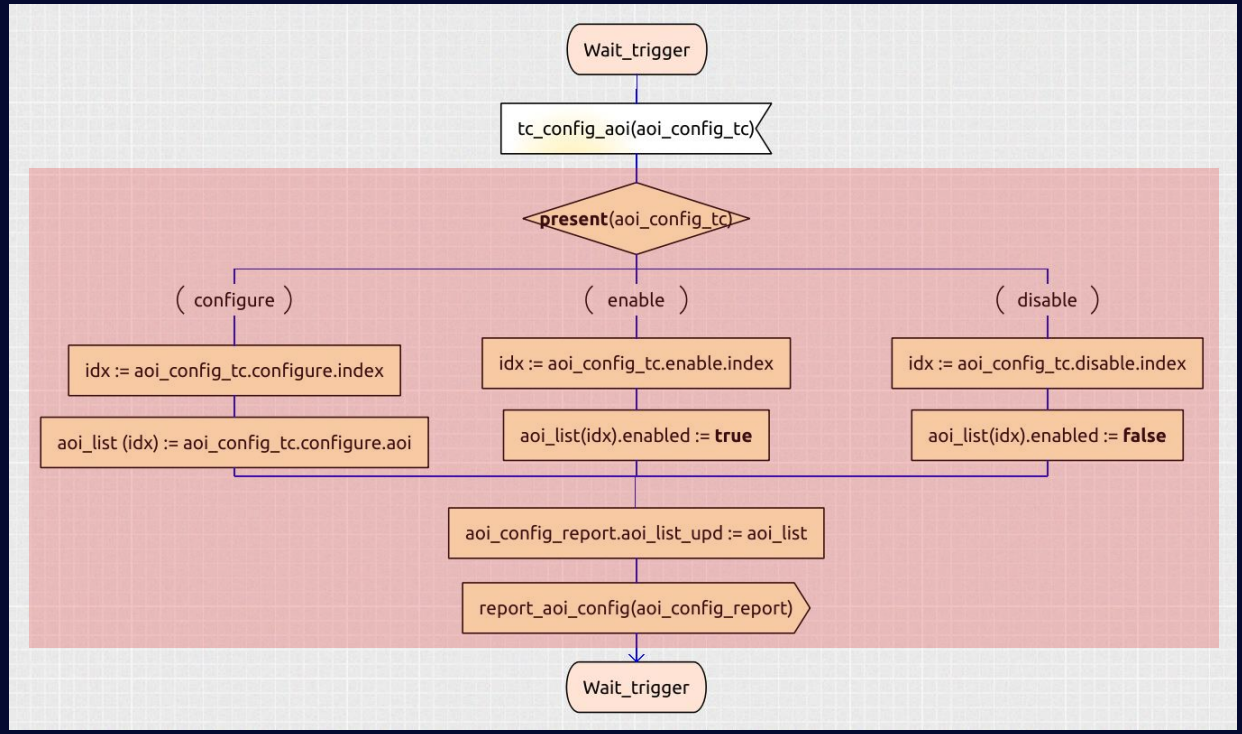


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

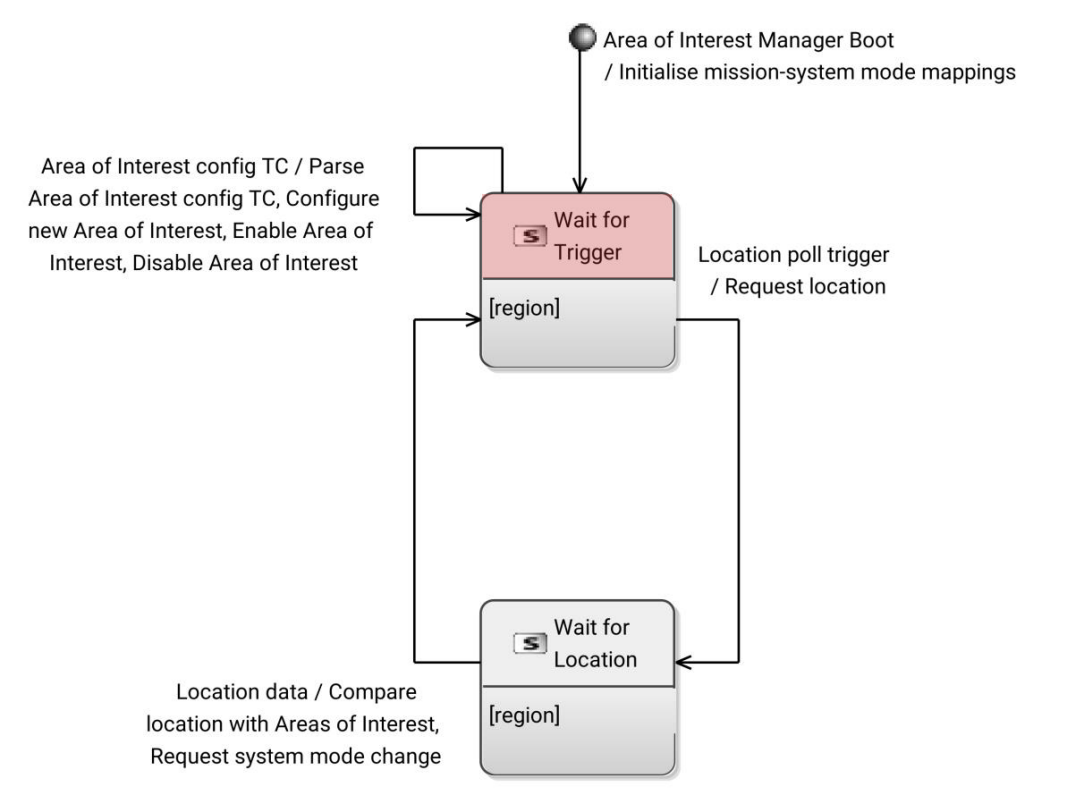


TASTE: SDL implementation

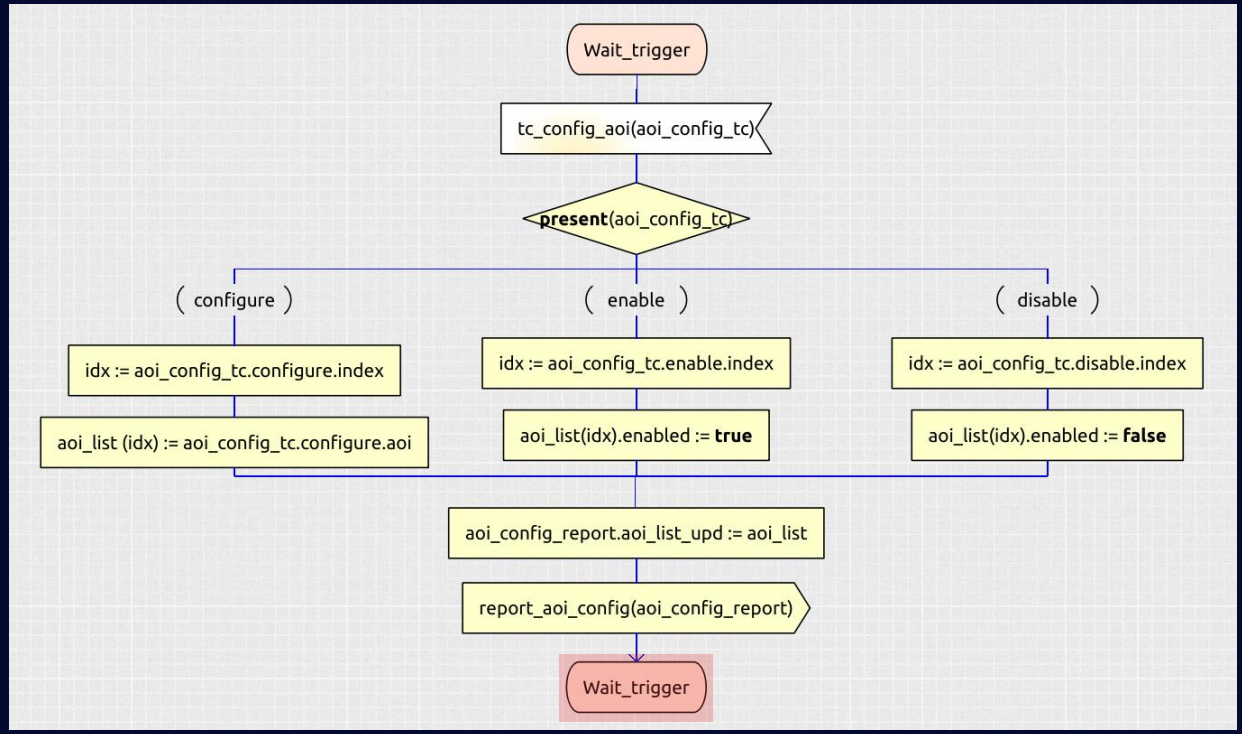


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

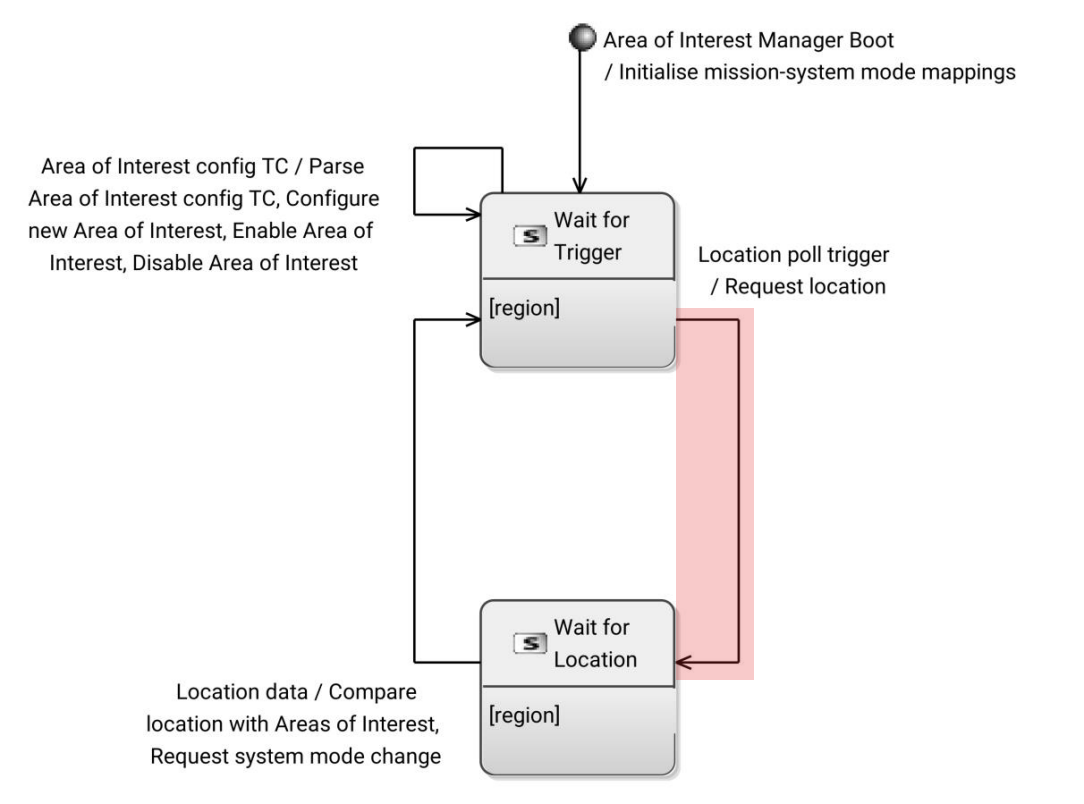


TASTE: SDL implementation



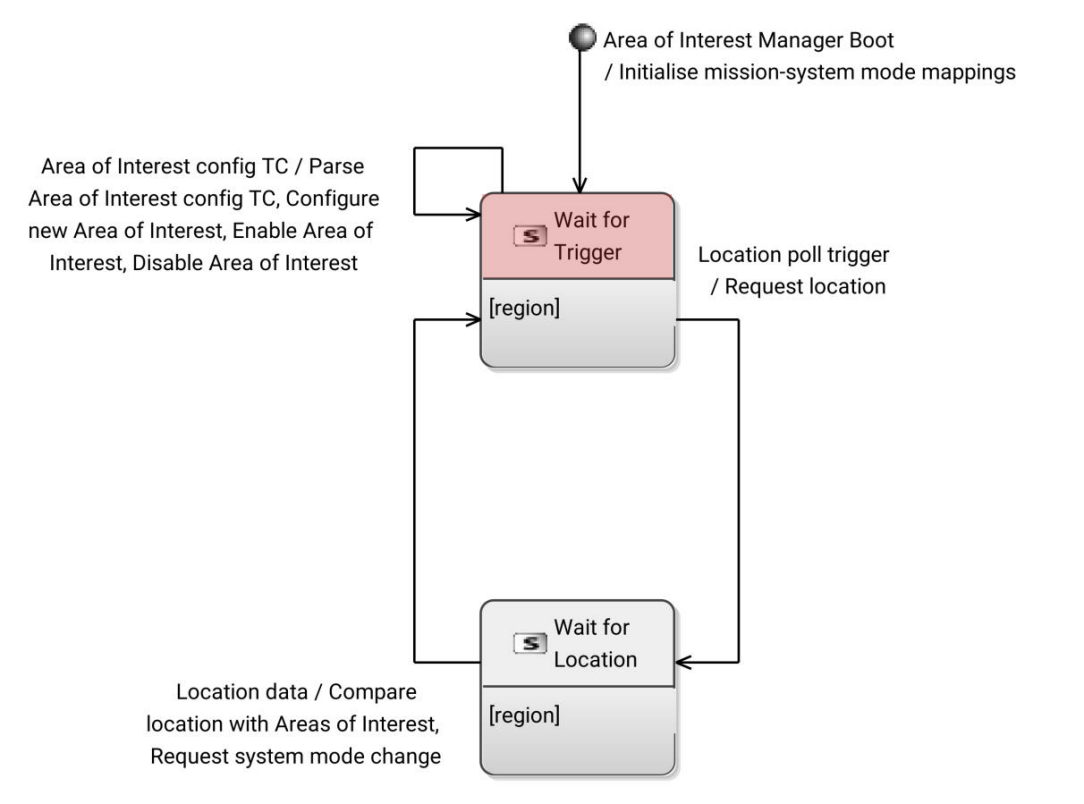
The Bridge: Behaviour Implementations

Capella:
Physical [MSM]

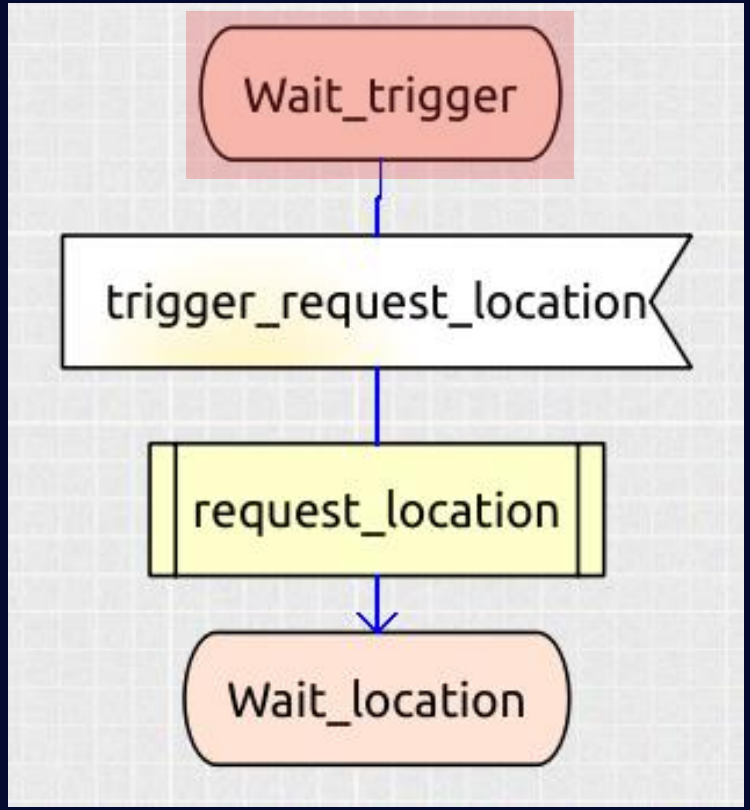


The Bridge: Behaviour Implementations

Capella:
Physical [MSM]

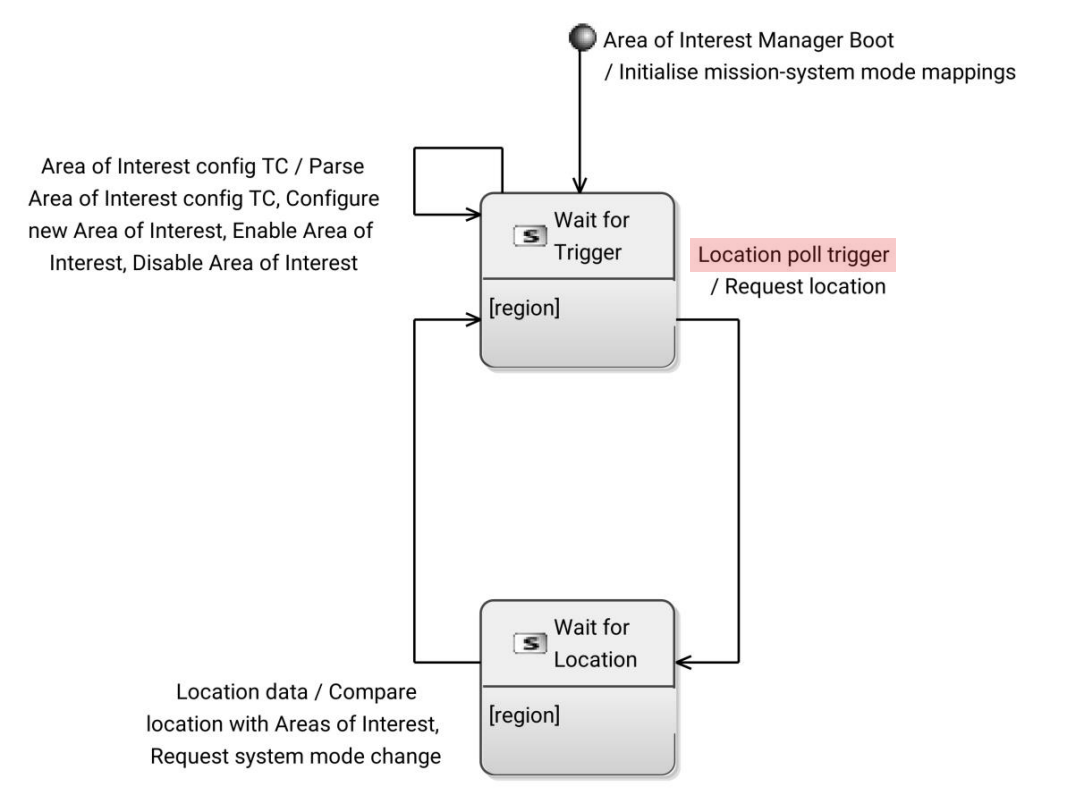


TASTE:
SDL implementation

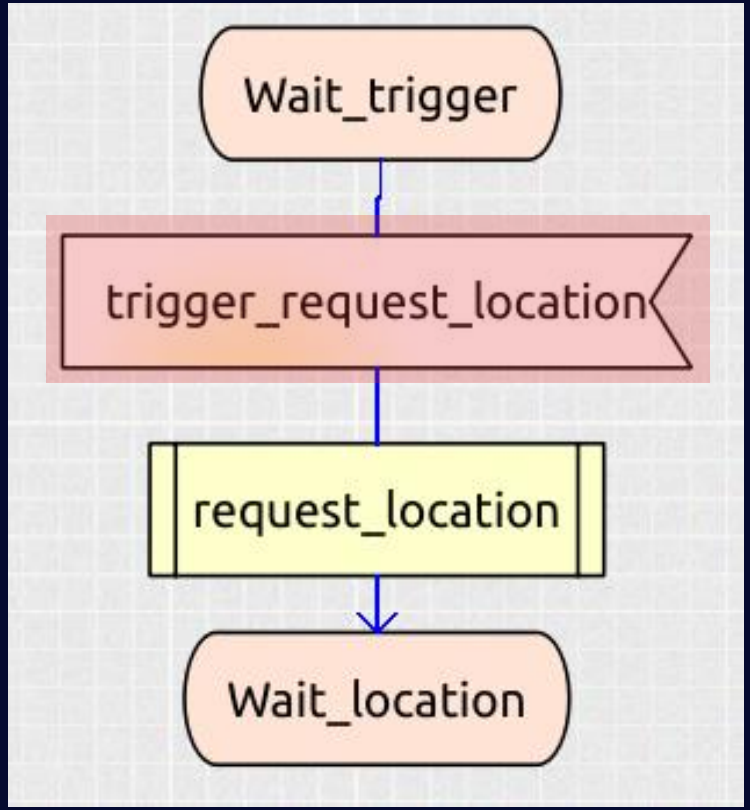


The Bridge: Behaviour Implementations

Capella:
Physical [MSM]

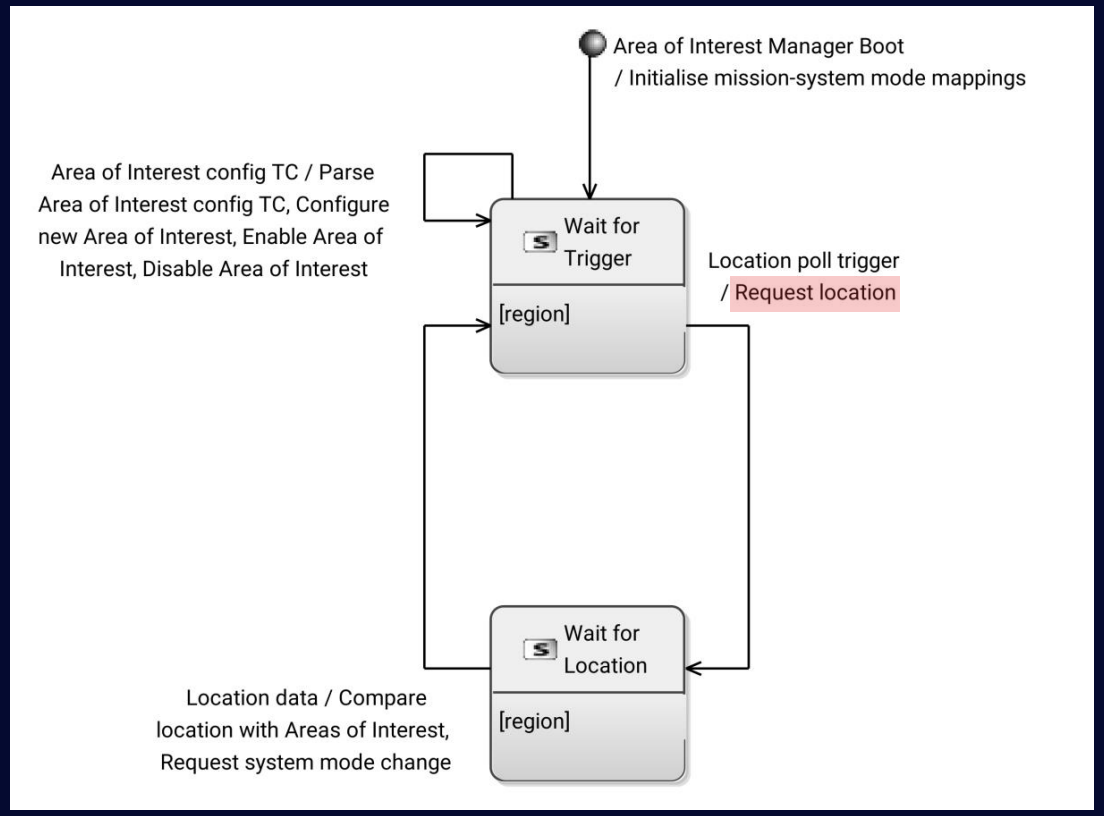


TASTE:
SDL implementation

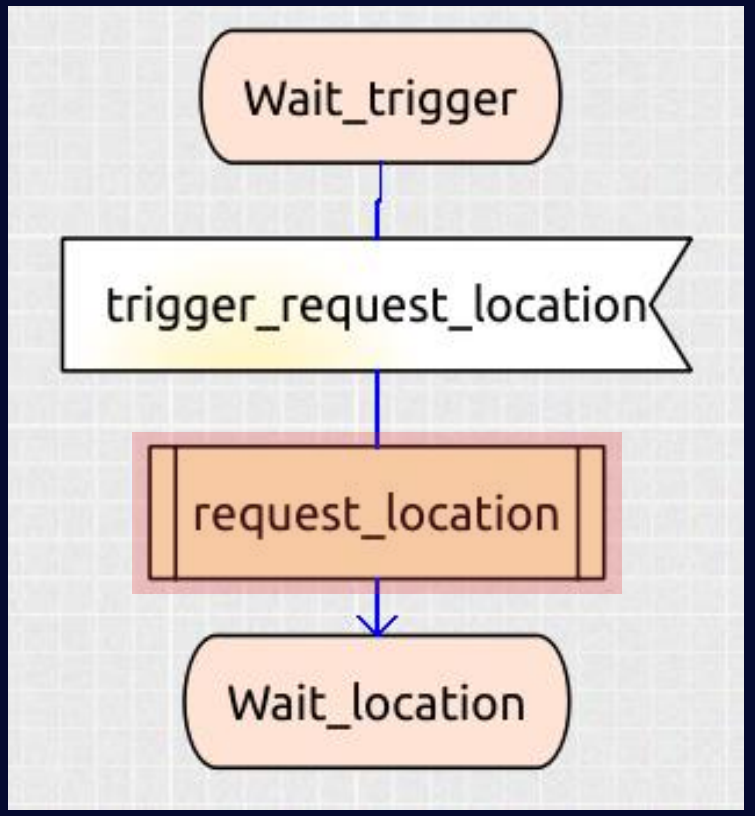


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

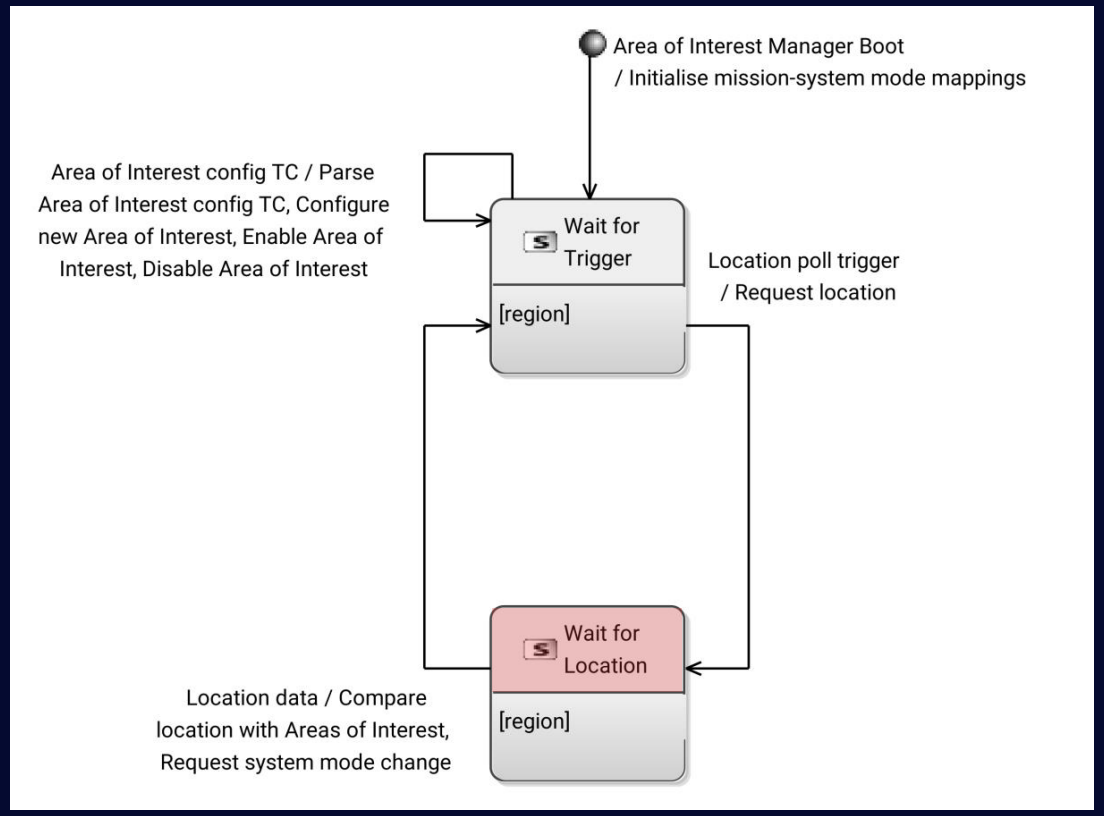


TASTE: SDL implementation

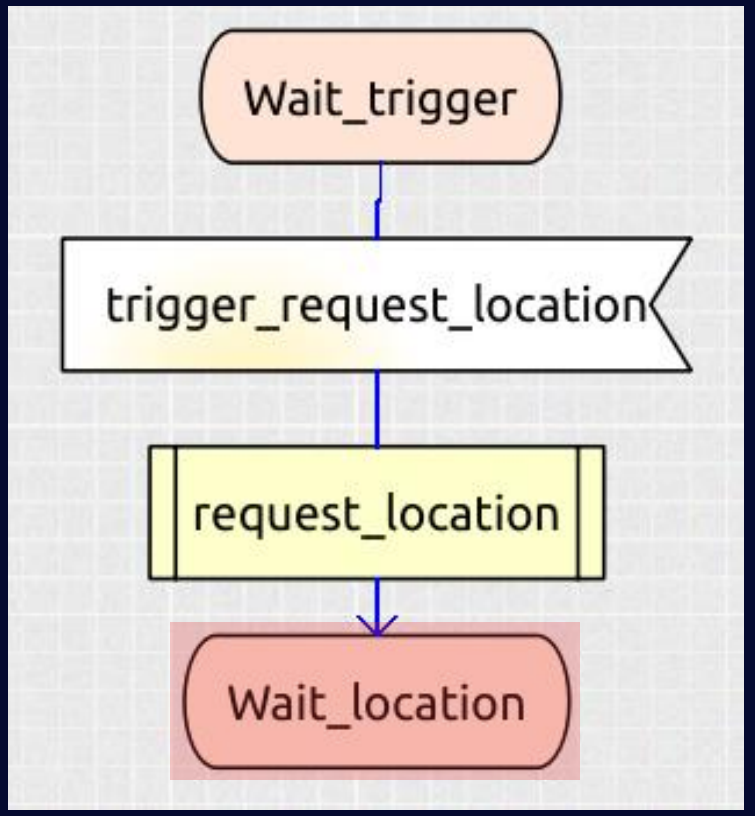


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

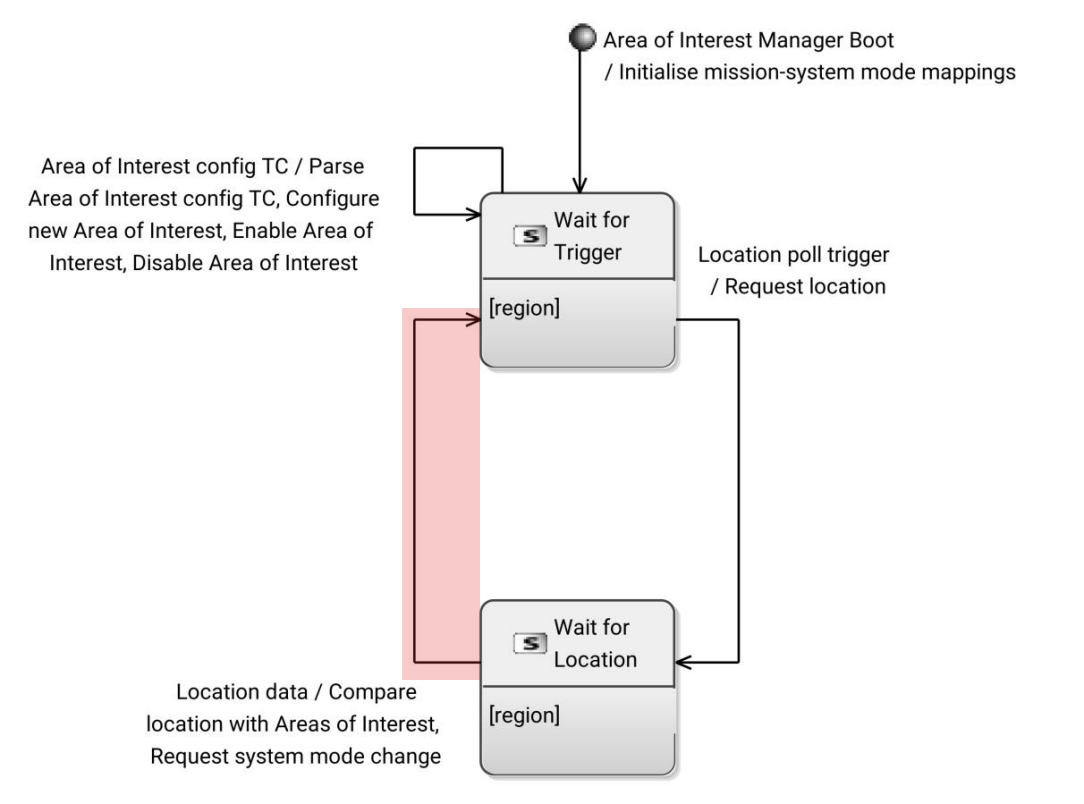


TASTE: SDL implementation



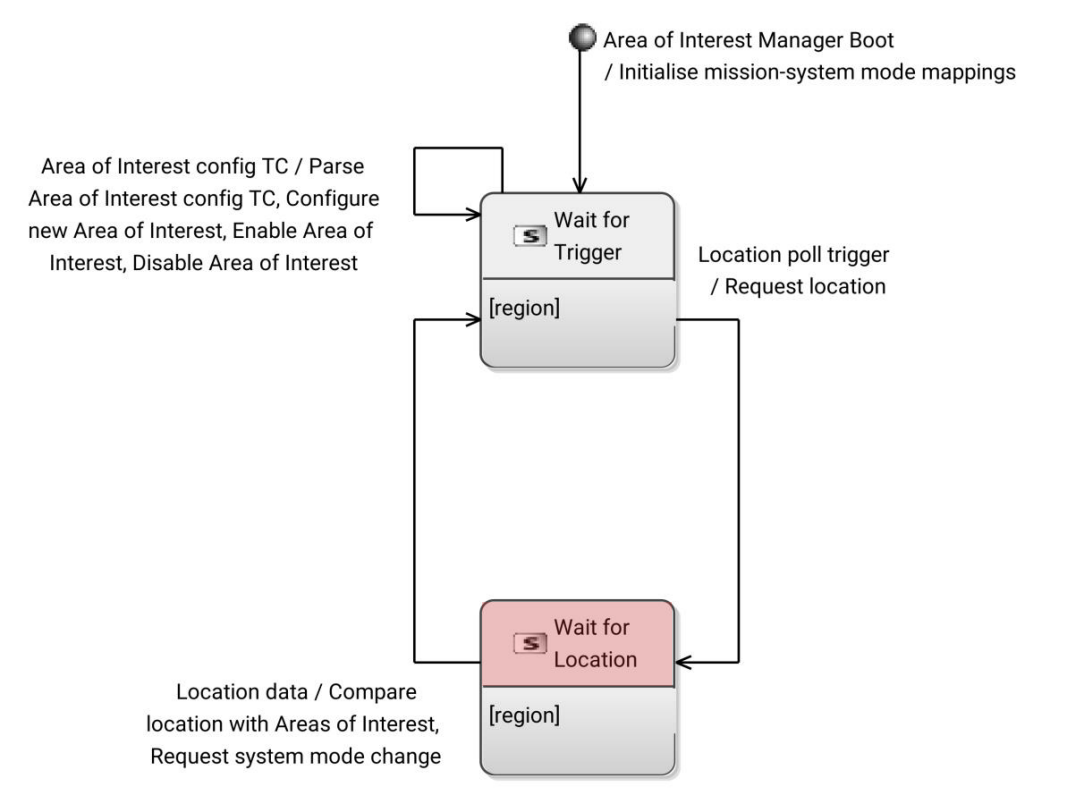
The Bridge: Behaviour Implementations

Capella:
Physical [MSM]

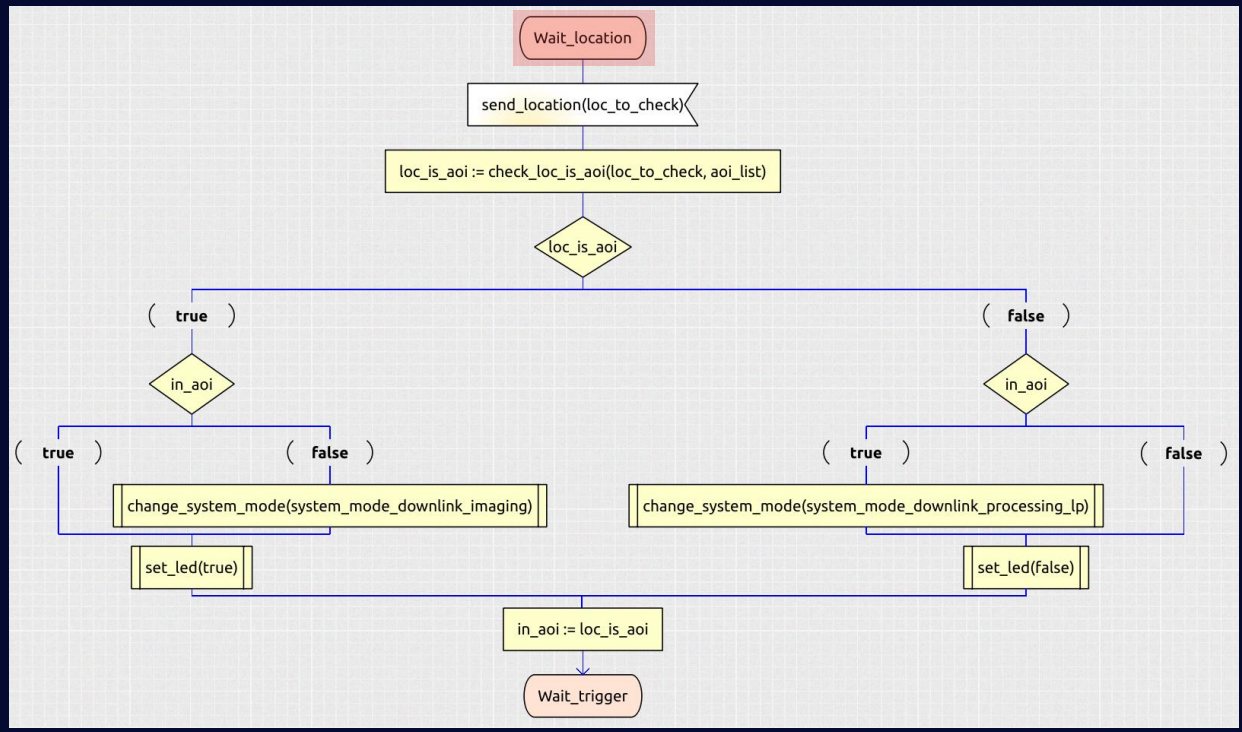


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

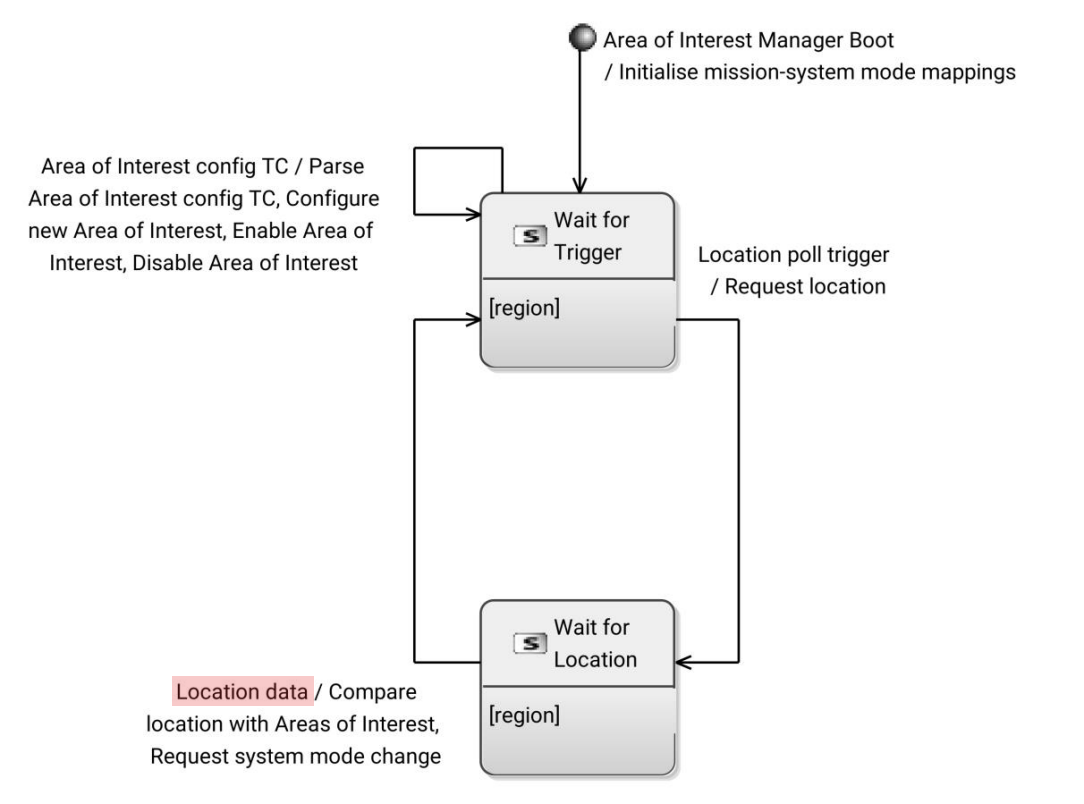


TASTE: SDL implementation

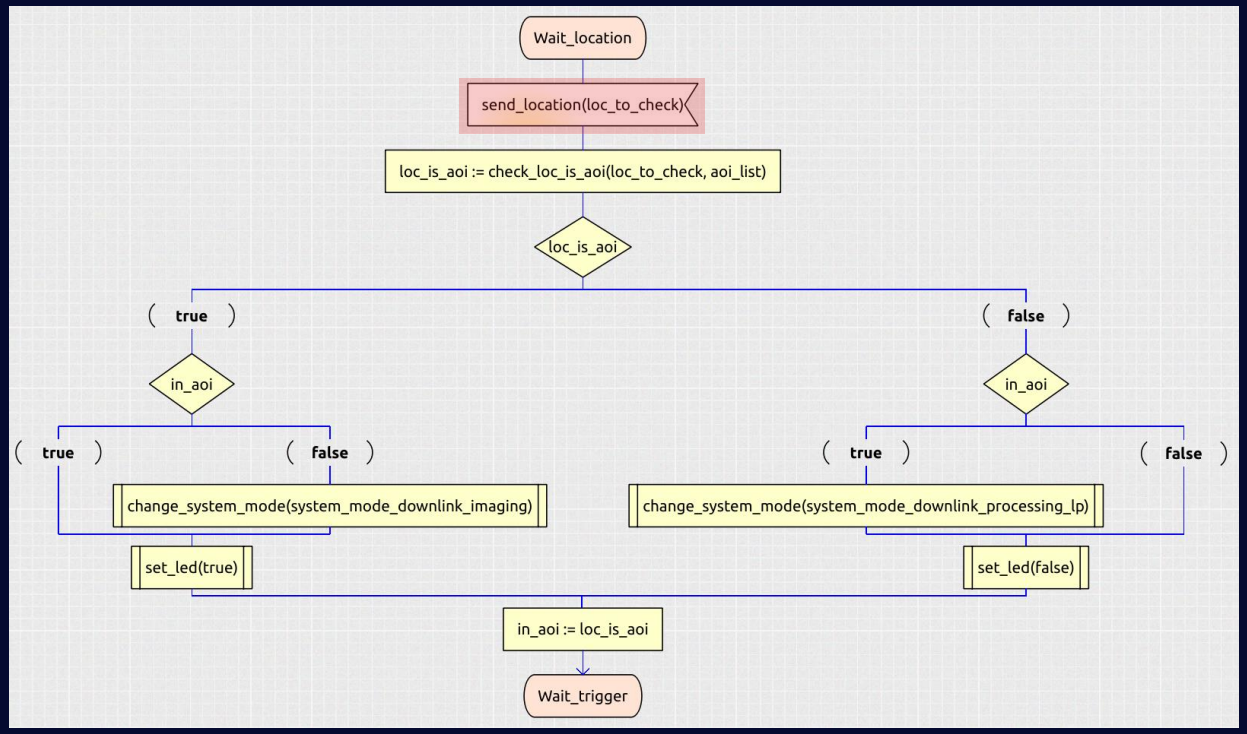


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

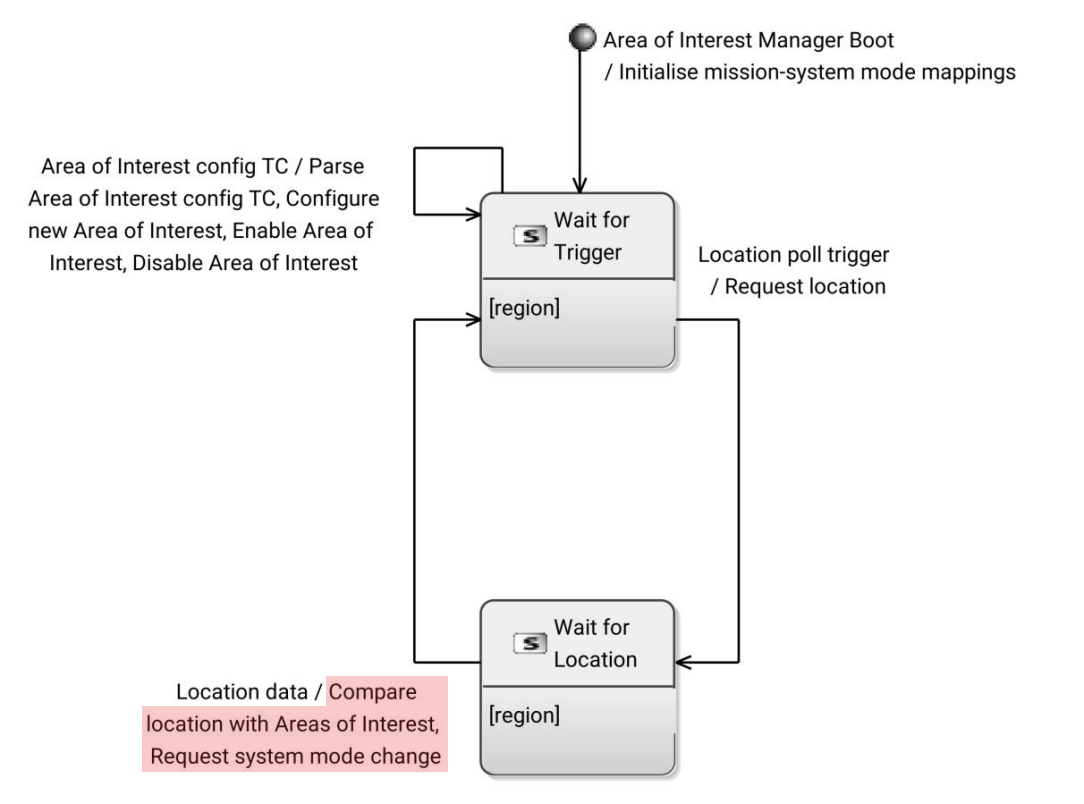


TASTE: SDL implementation

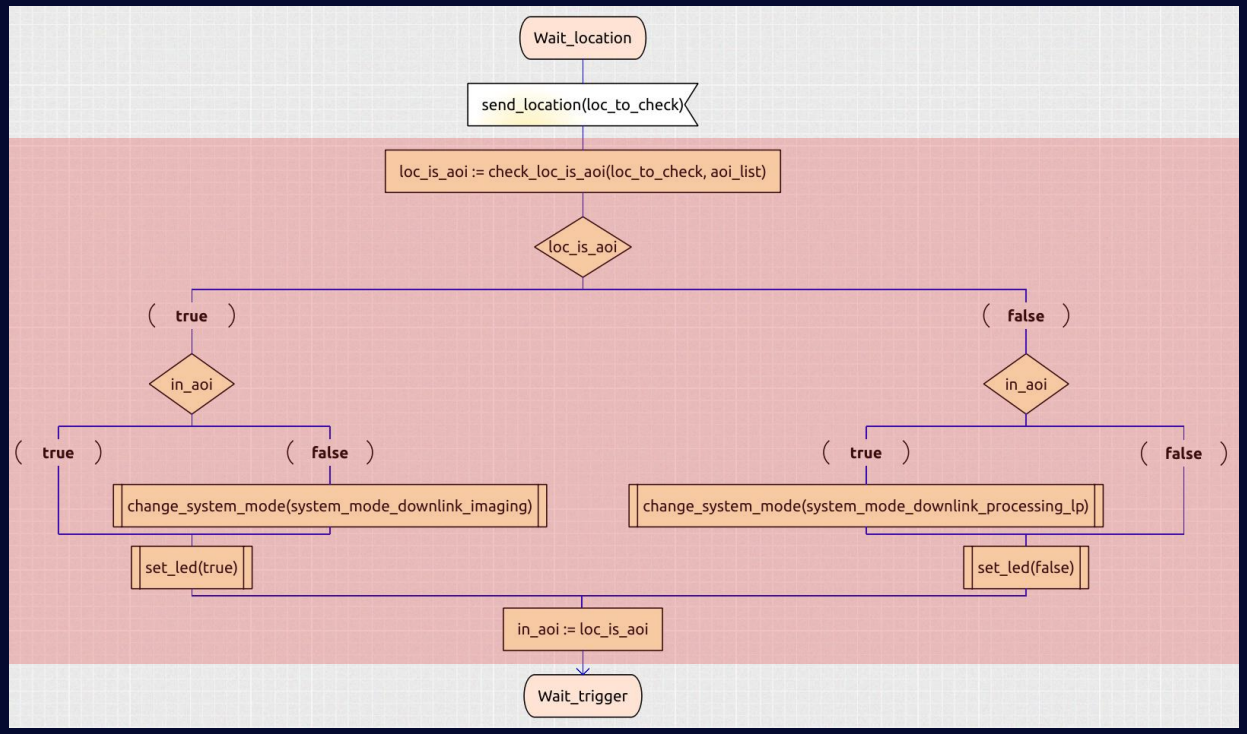


The Bridge: Behaviour Implementations

Capella: Physical [MSM]

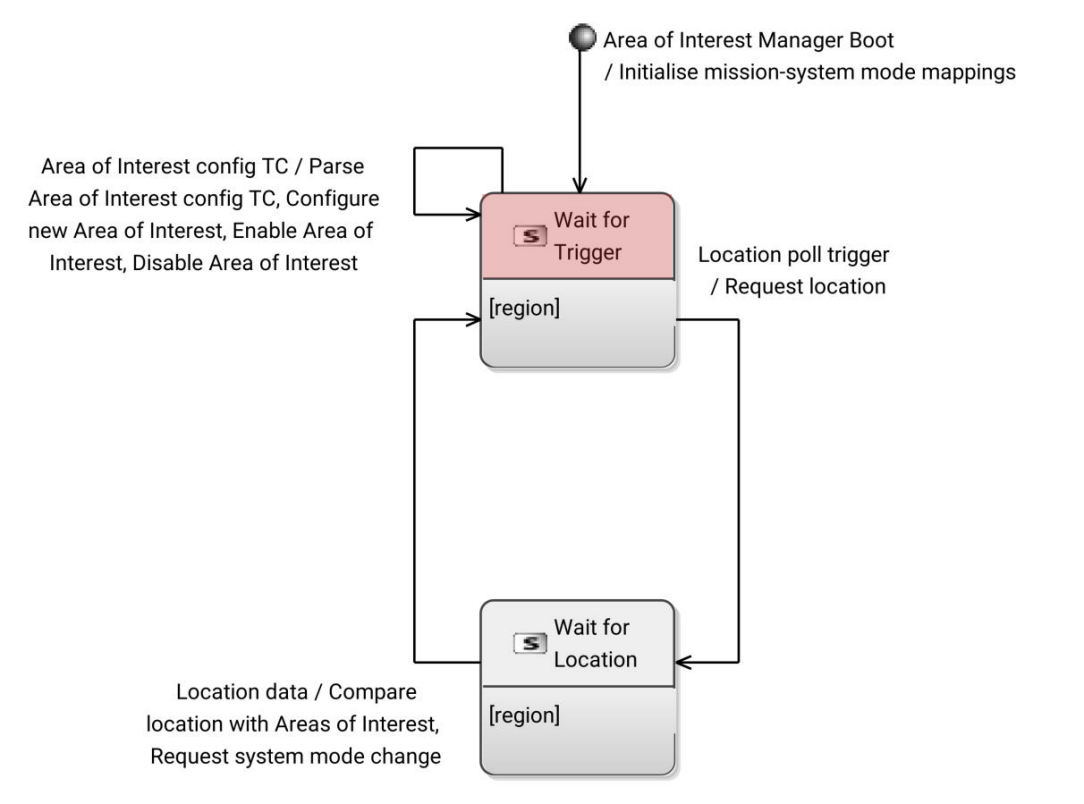


TASTE: SDL implementation

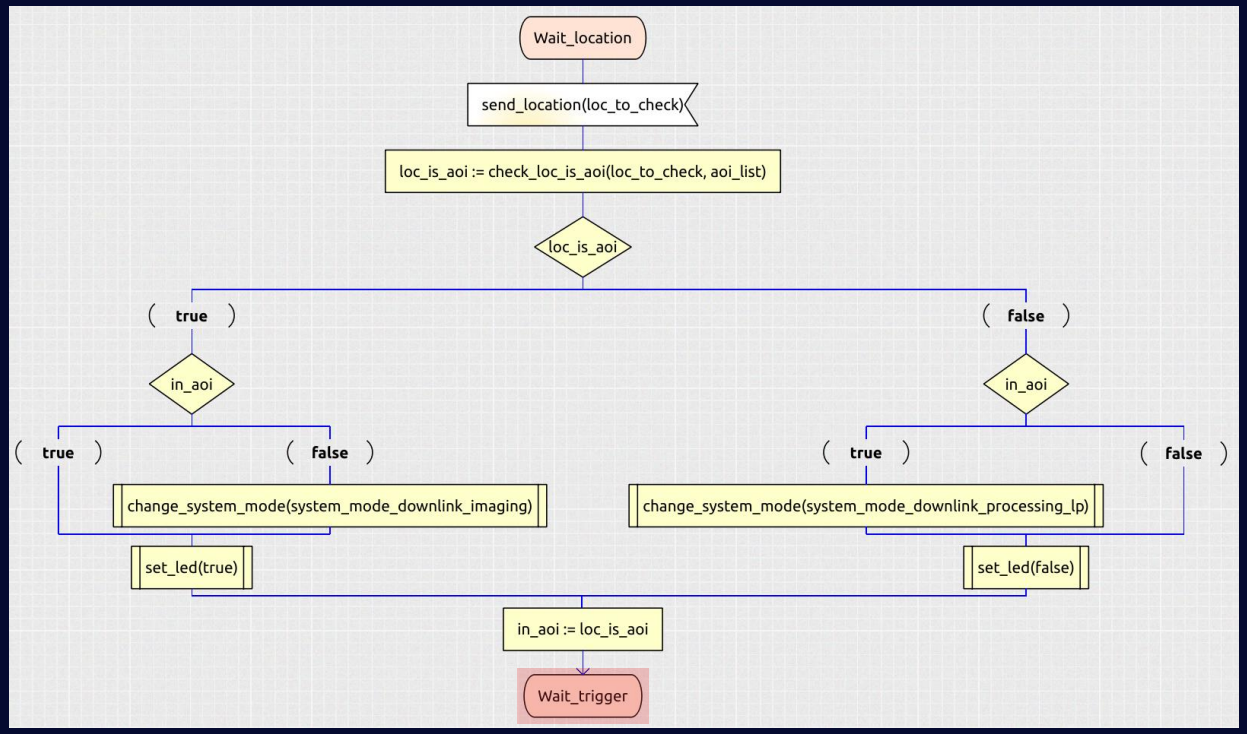


The Bridge: Behaviour Implementations

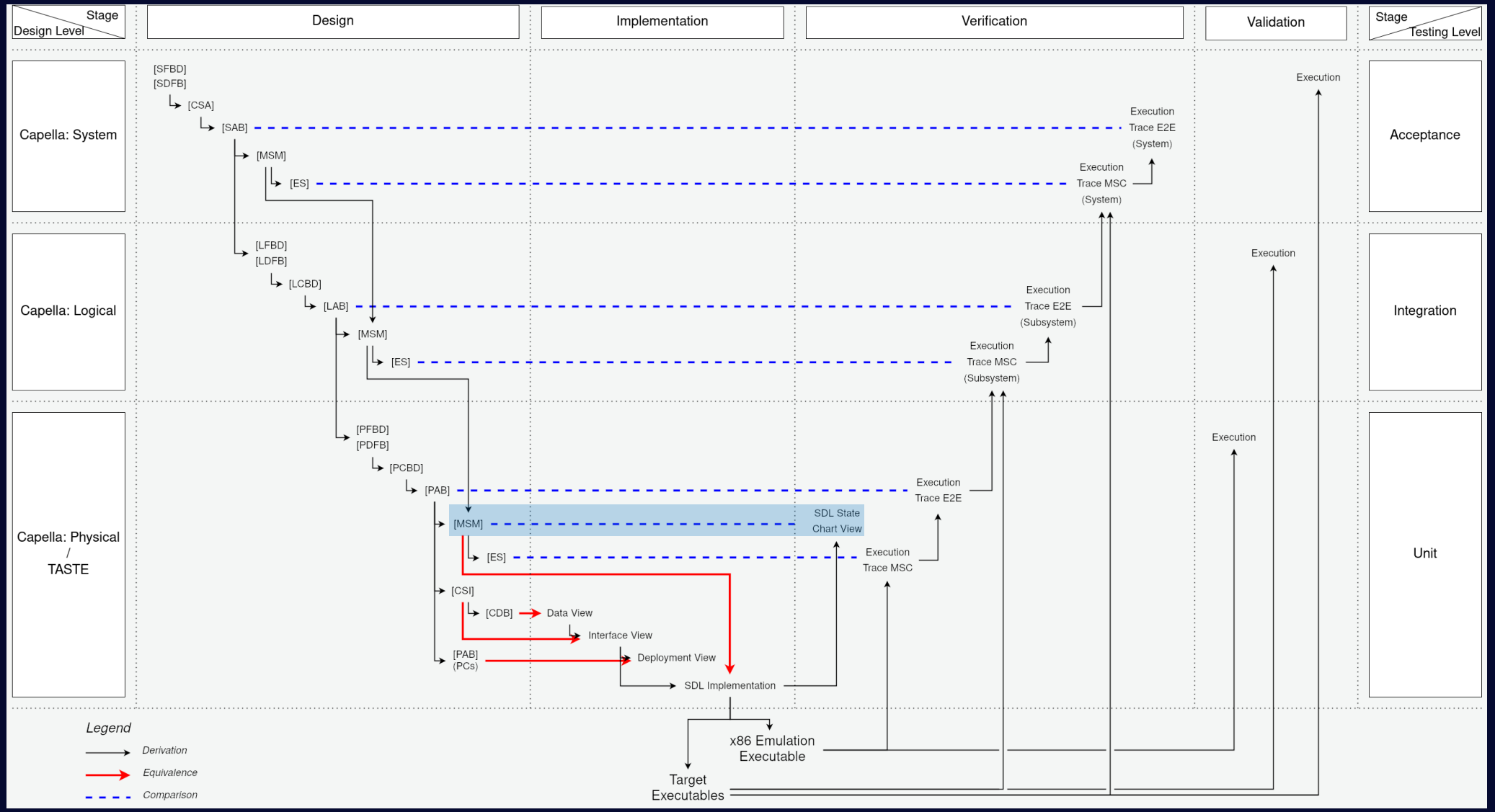
Capella: Physical [MSM]



TASTE: SDL implementation

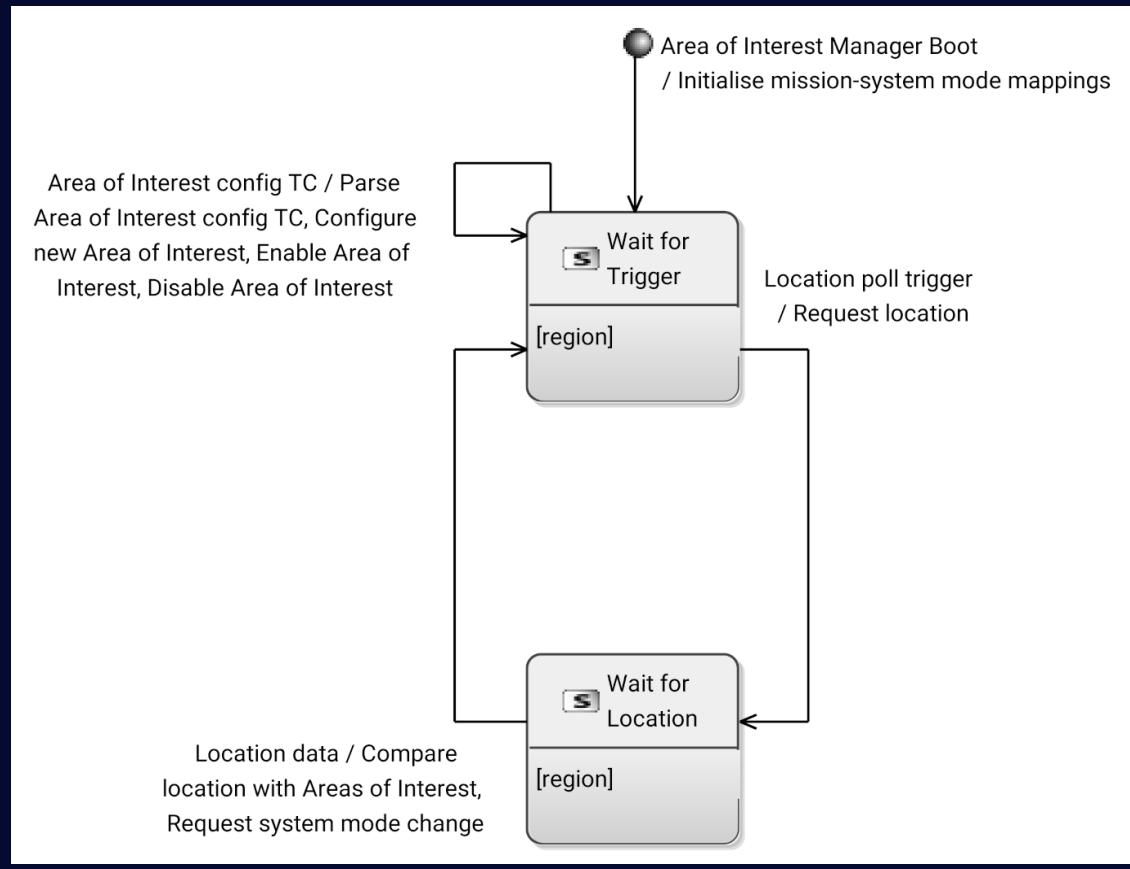


The Bridge: Behaviour Verifications

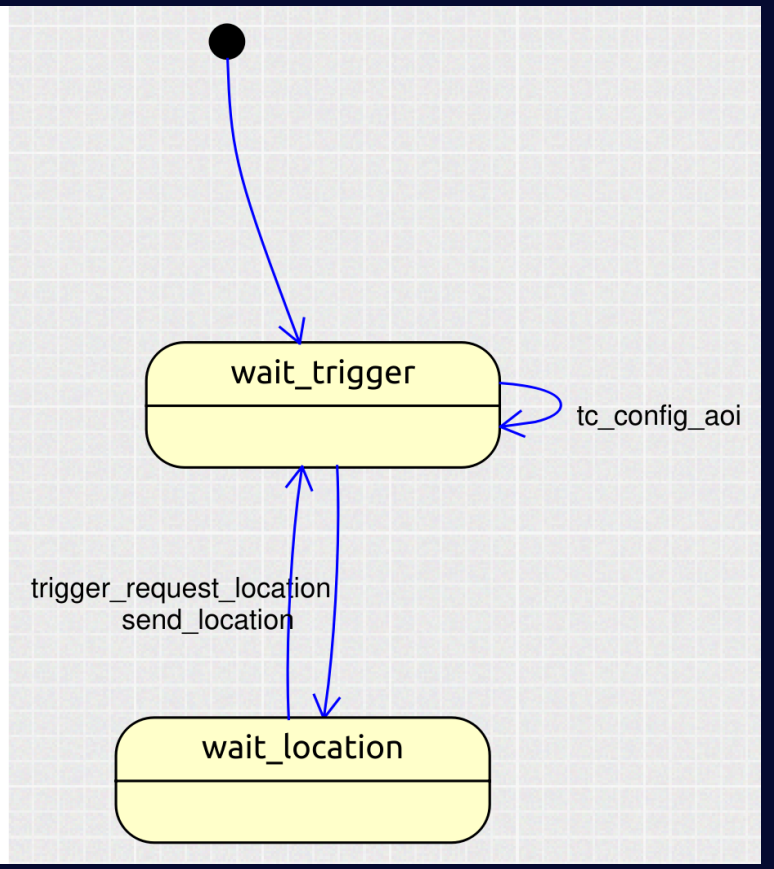


The Bridge: Behaviour Verifications

Capella:
Physical [MSM]

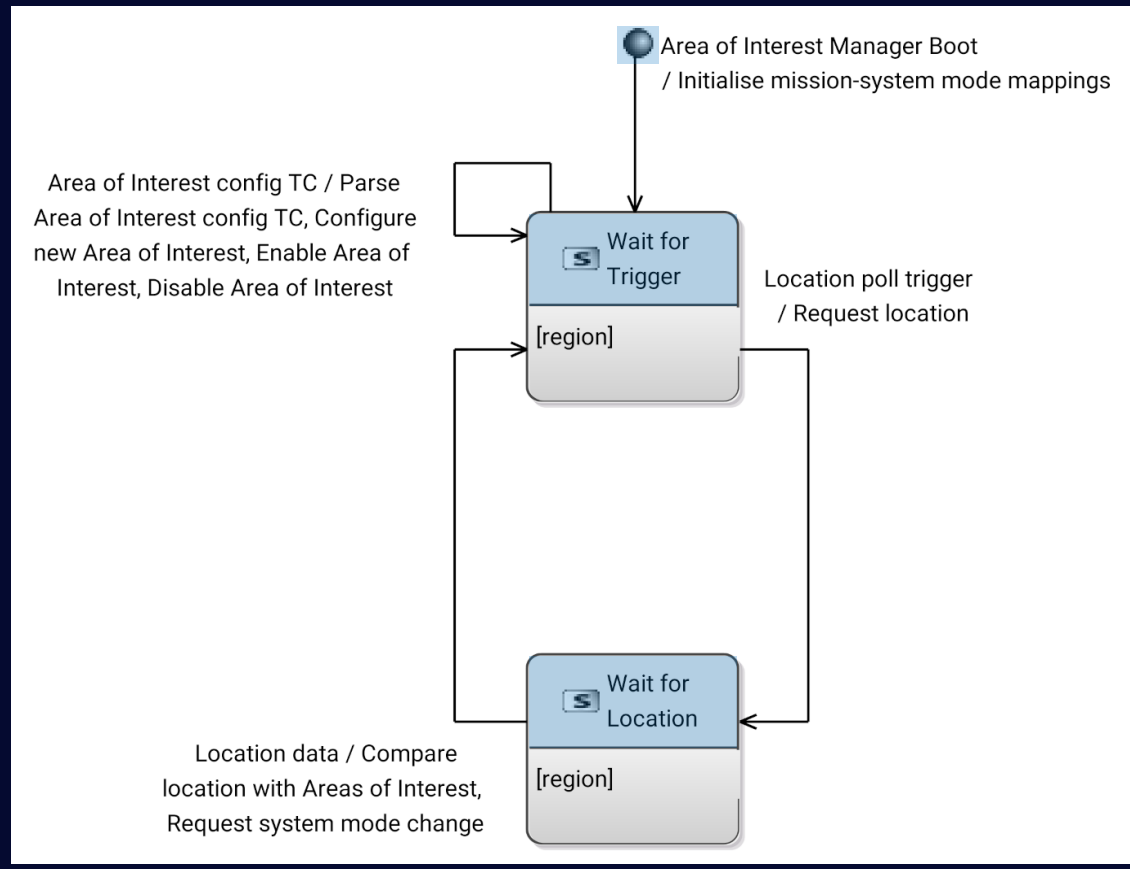


TASTE:
SDL State Chart View

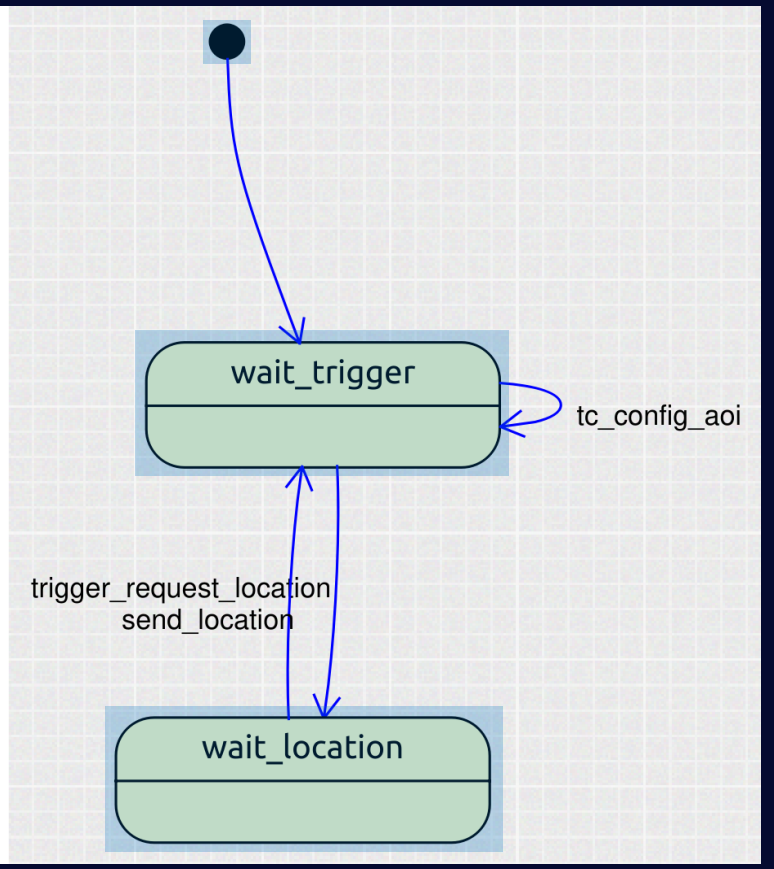


The Bridge: Behaviour Verifications

Capella:
Physical [MSM]

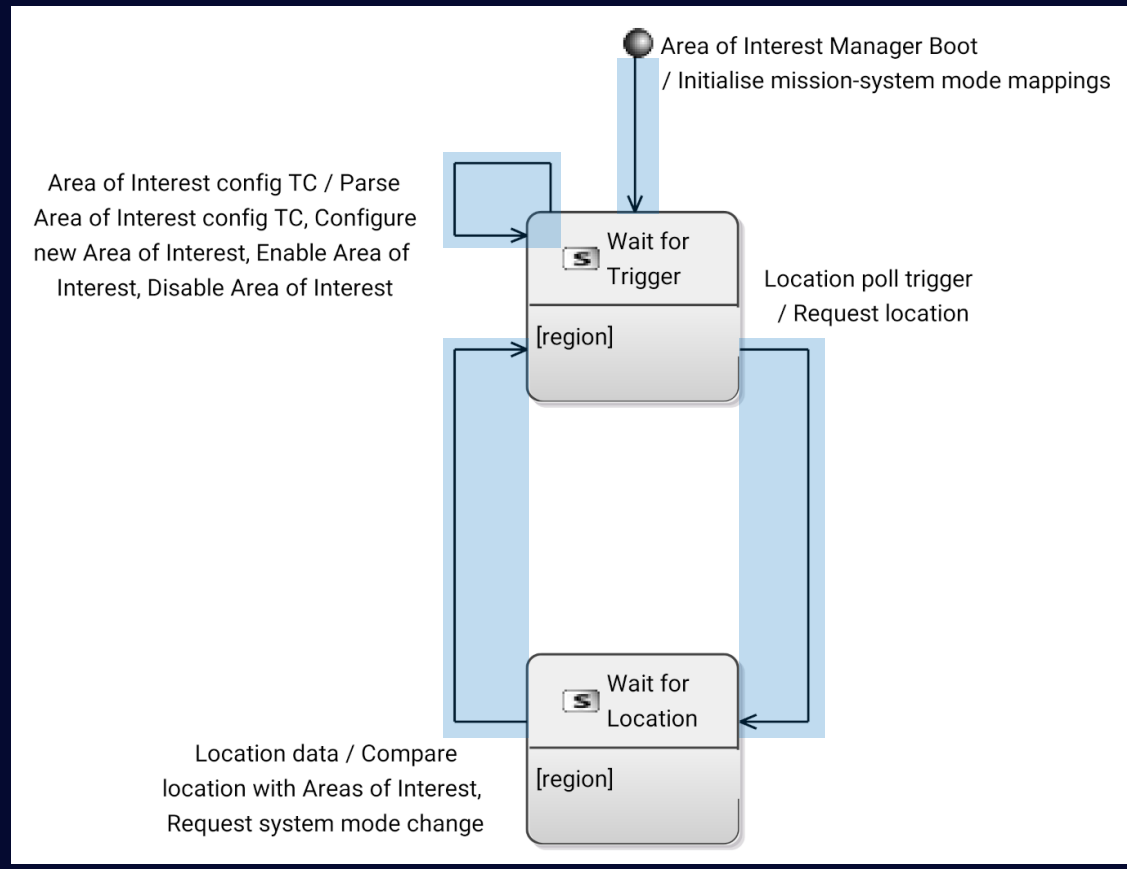


TASTE:
SDL State Chart View

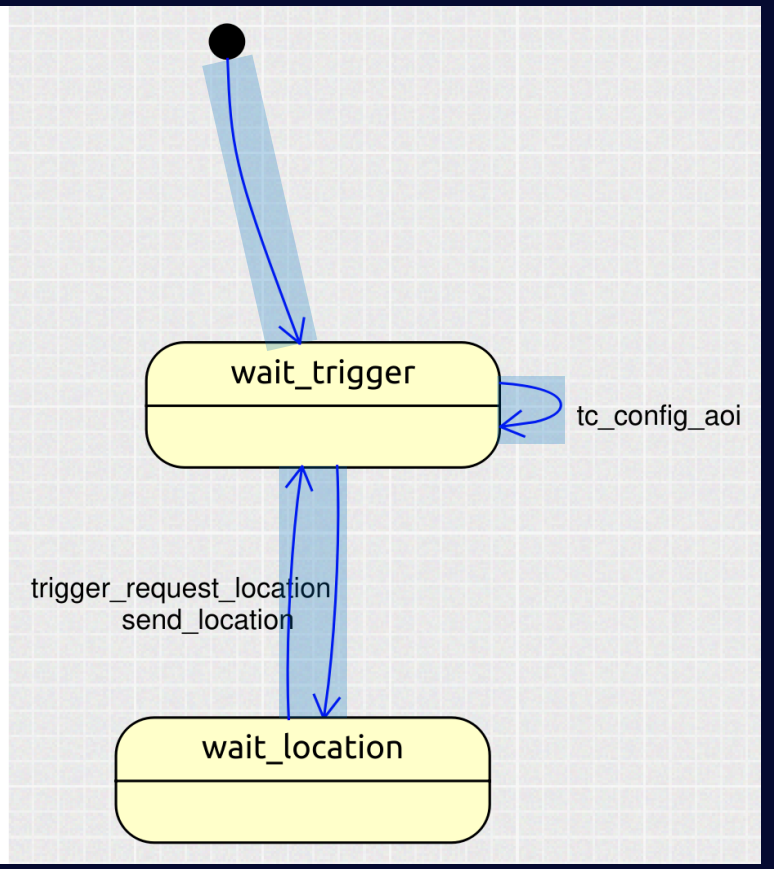


The Bridge: Behaviour Verifications

Capella:
Physical [MSM]

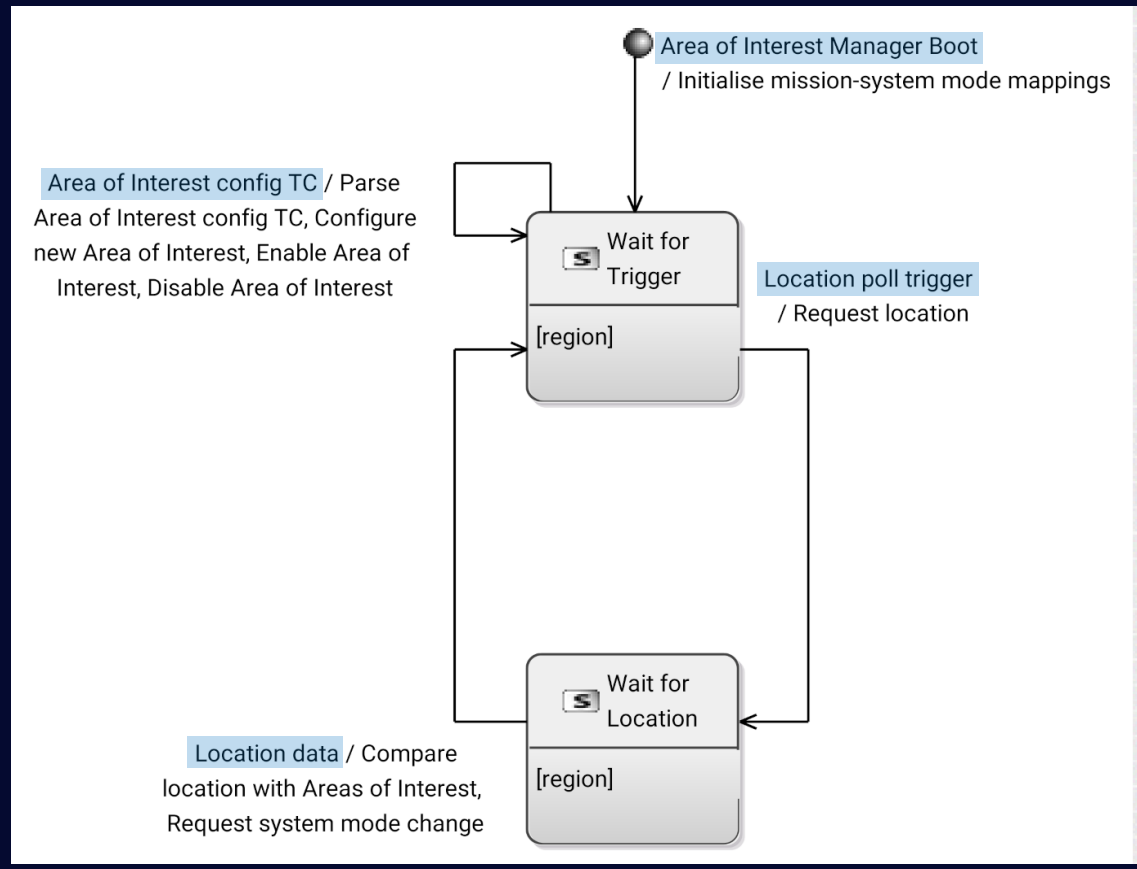


TASTE:
SDL State Chart View

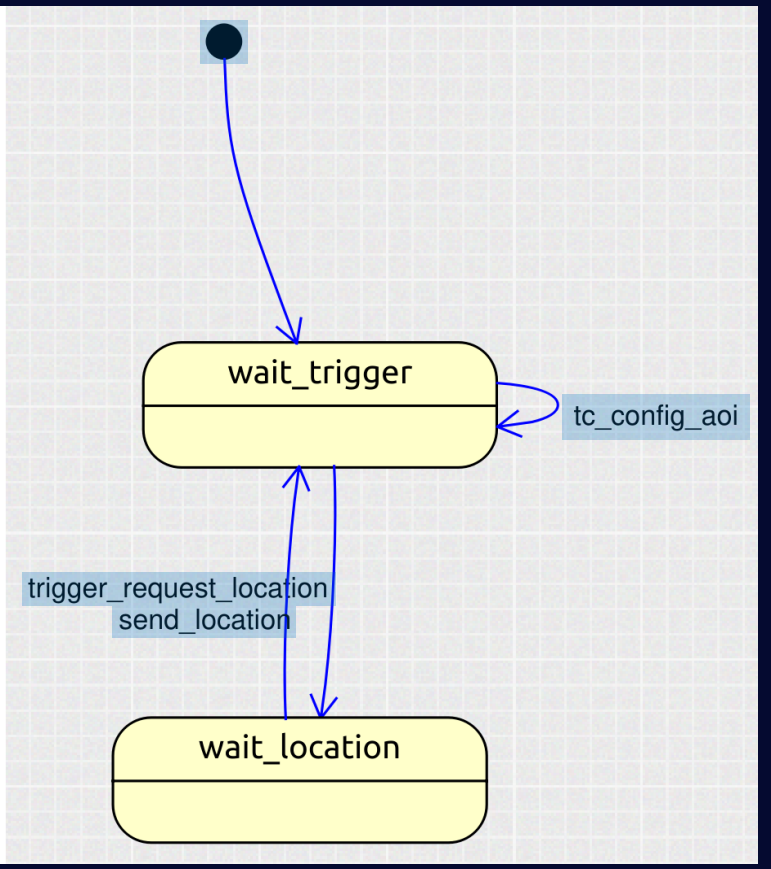


The Bridge: Behaviour Verifications

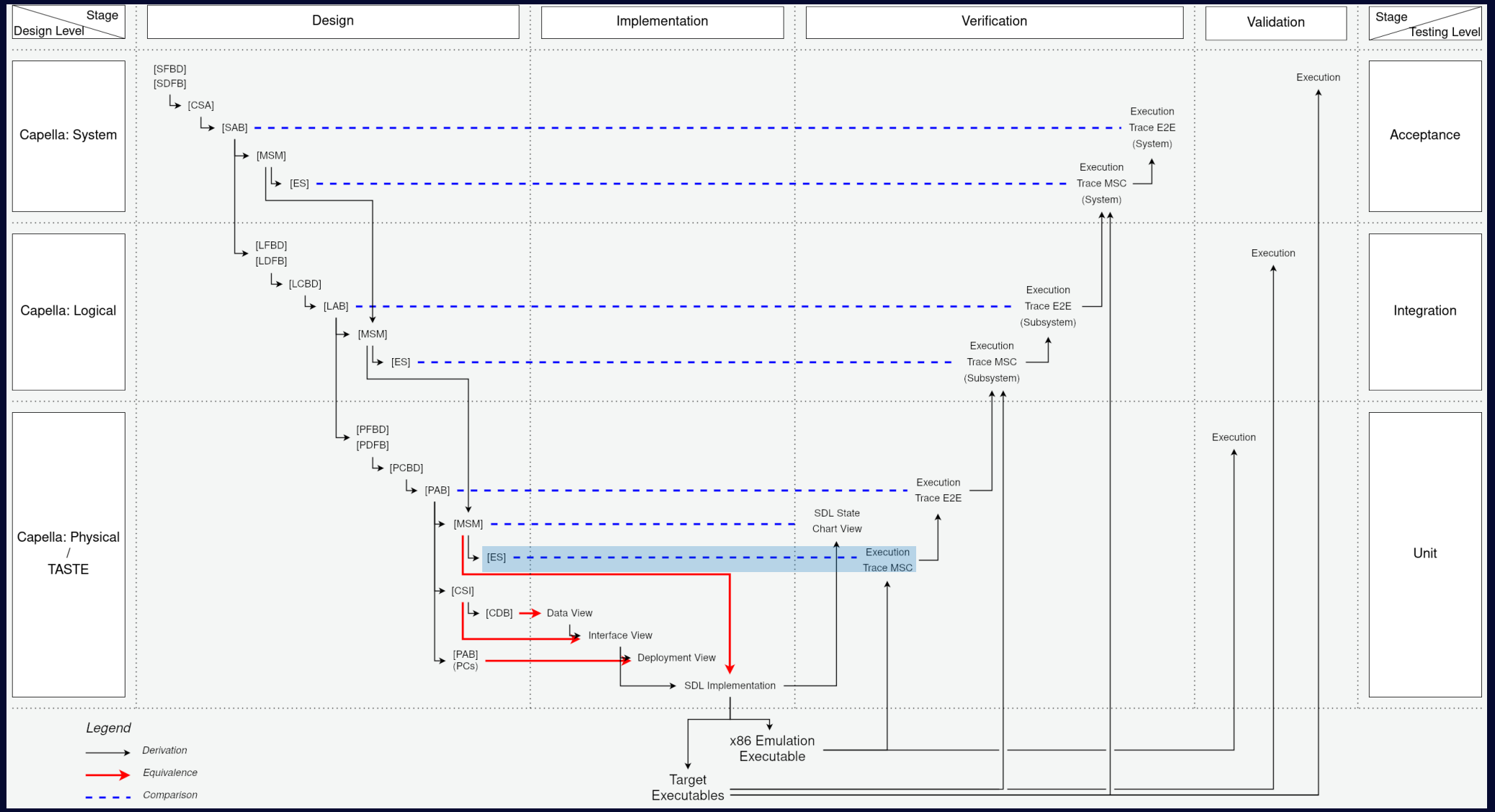
Capella:
Physical [MSM]



TASTE:
SDL State Chart View

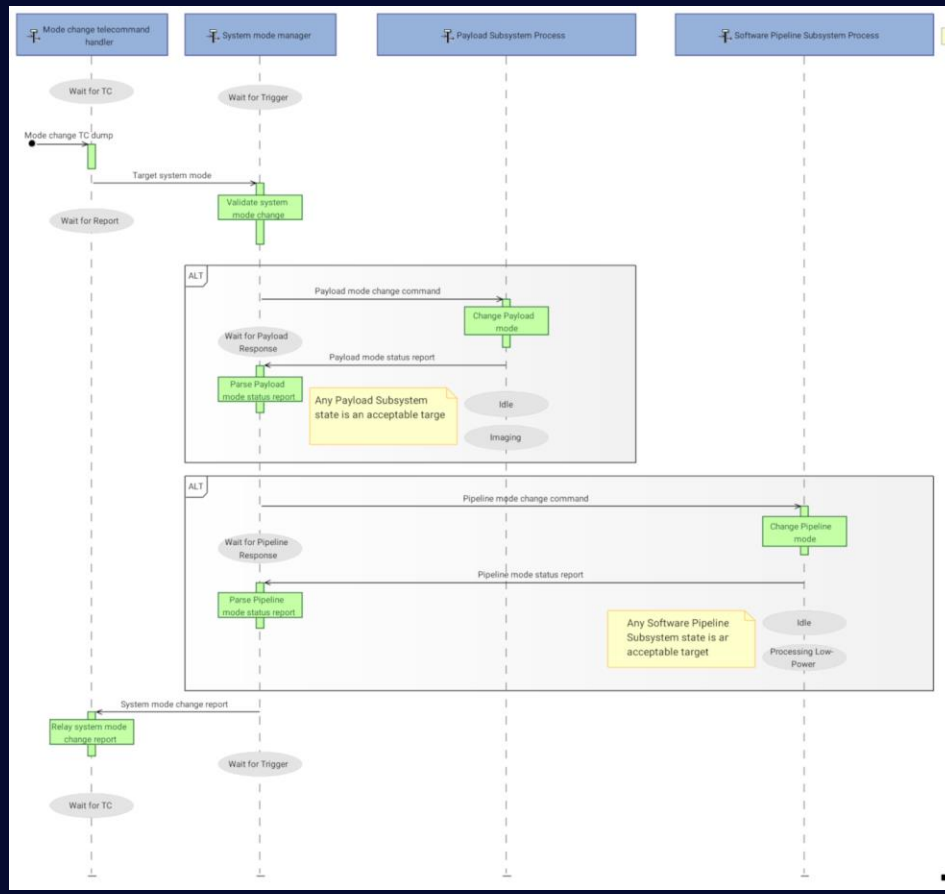


The Bridge: Exchange Verifications

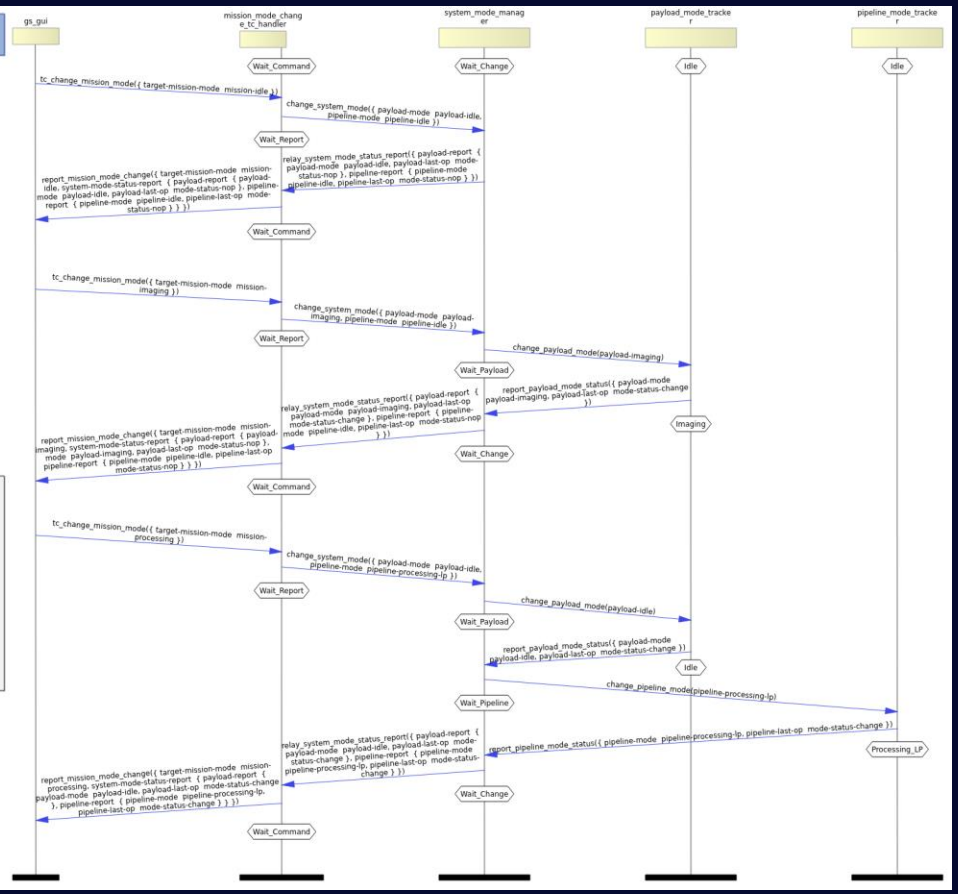


The Bridge: Exchange Verifications

Capella:
Physical [ES]

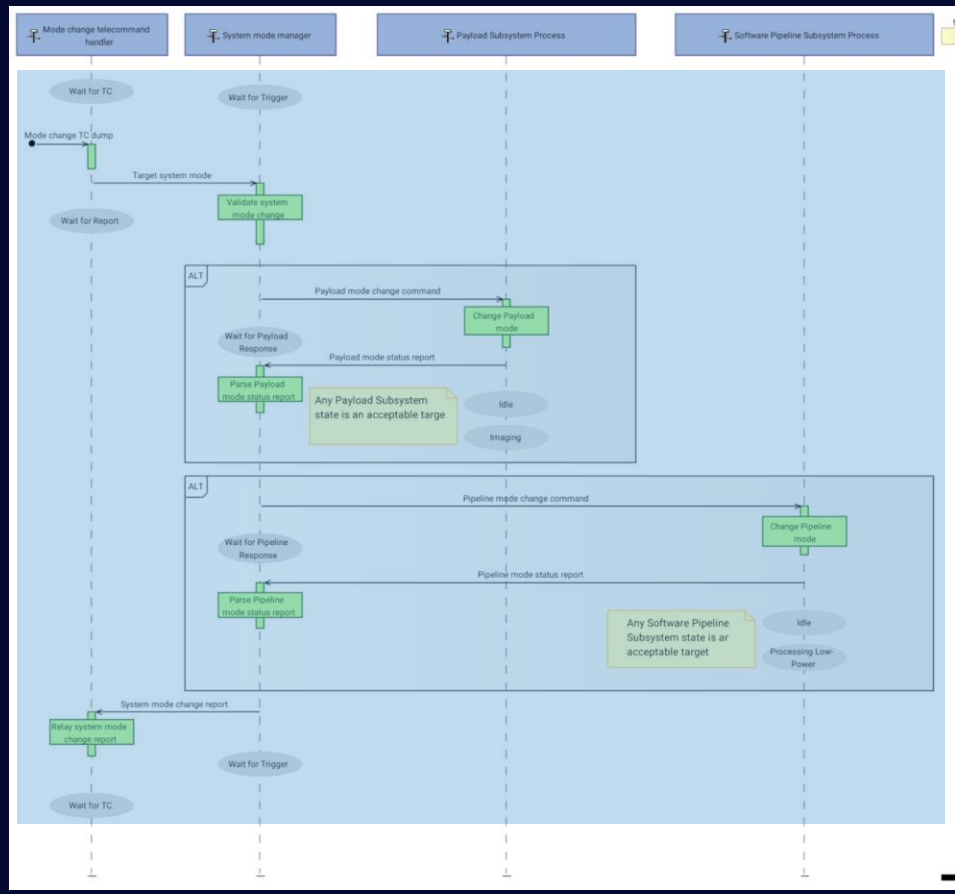


TASTE:
Message Sequence Chart

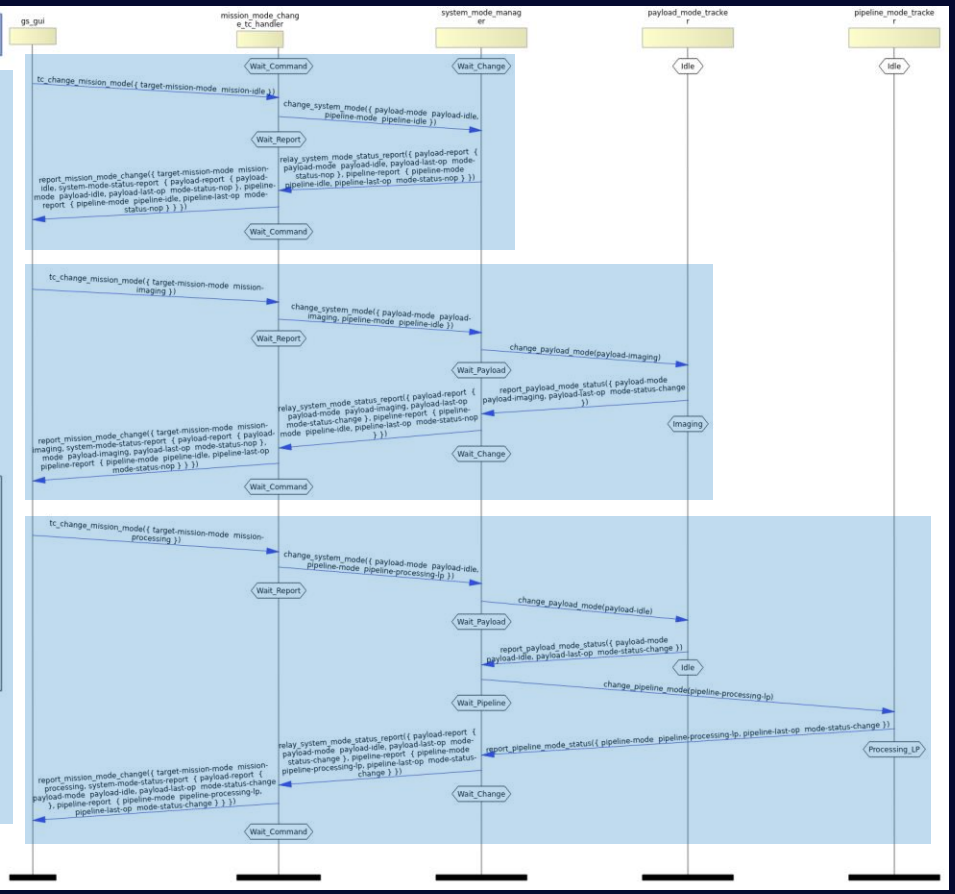


The Bridge: Exchange Verifications

Capella:
Physical [ES]

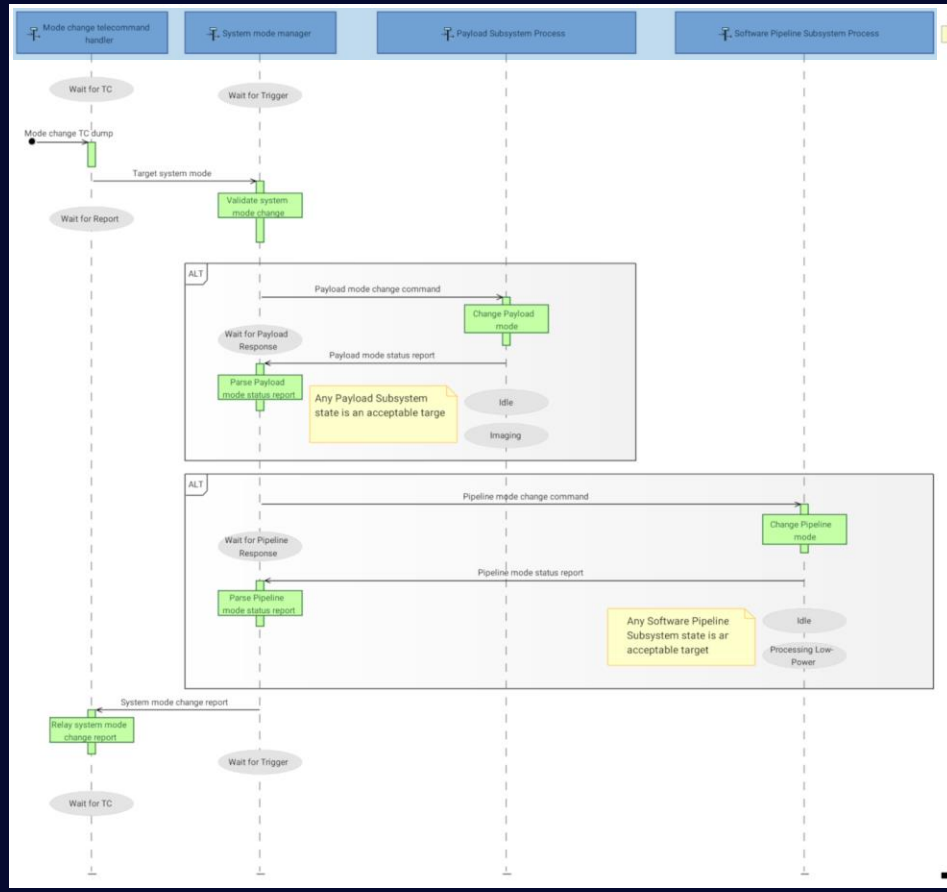


TASTE:
Message Sequence Chart

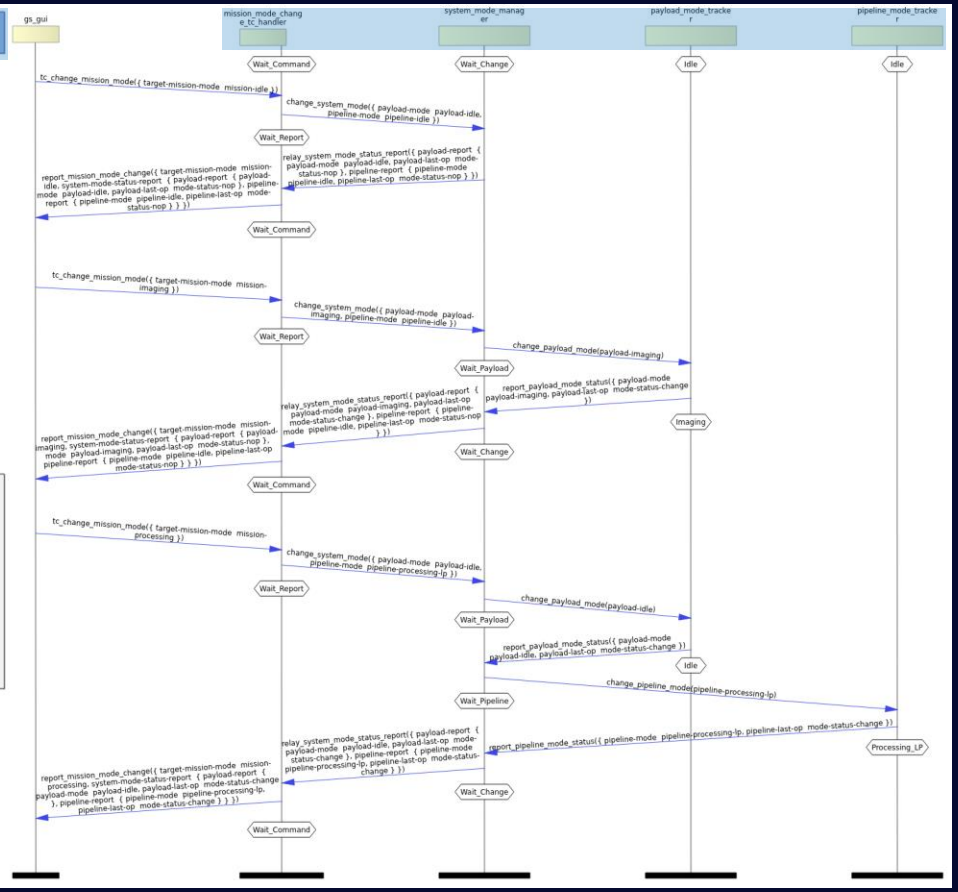


The Bridge: Exchange Verifications

Capella:
Physical [ES]

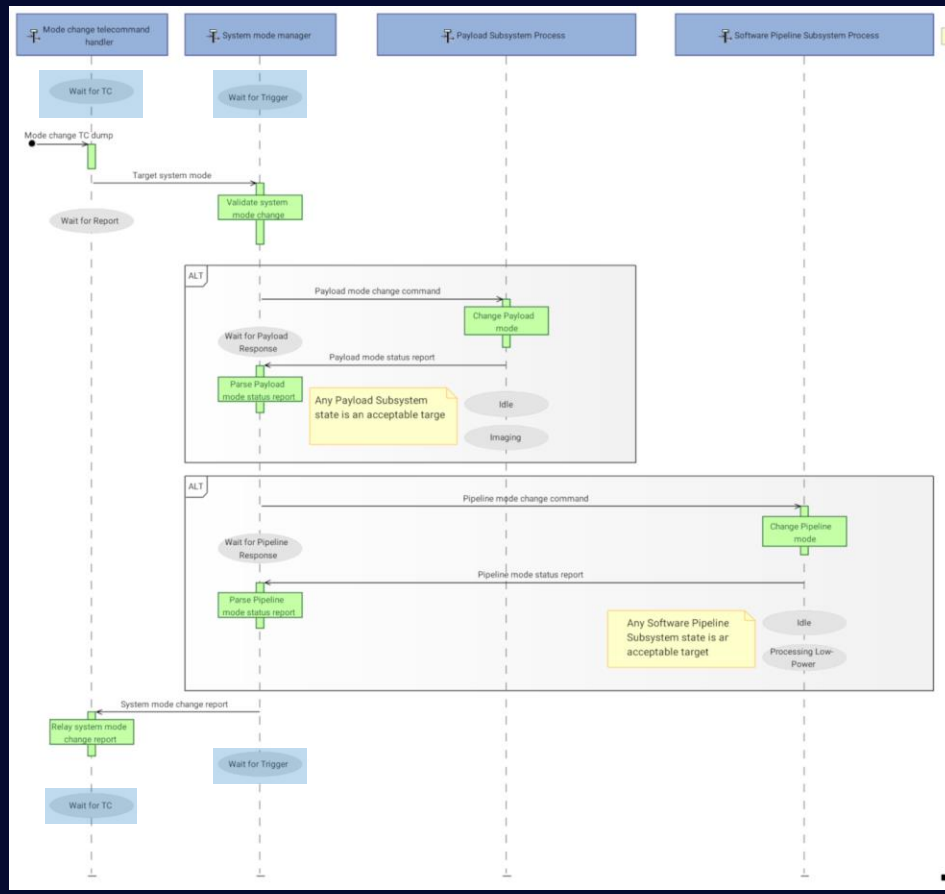


TASTE:
Message Sequence Chart

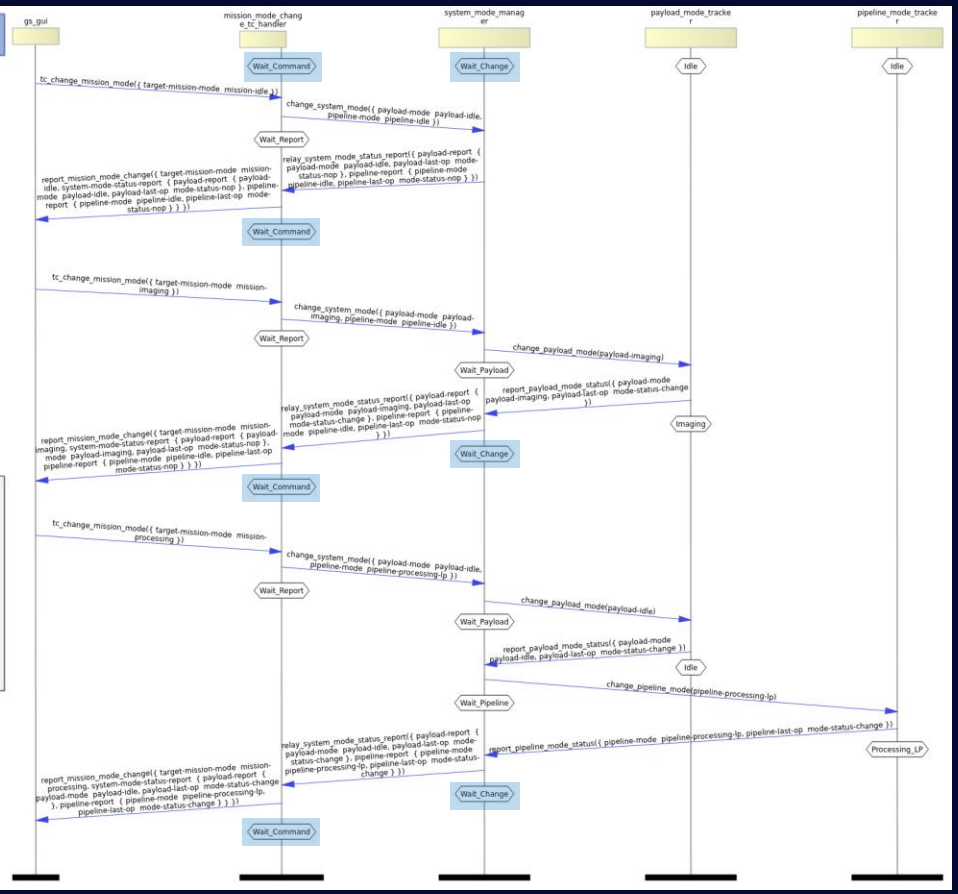


The Bridge: Exchange Verifications

Capella:
Physical [ES]

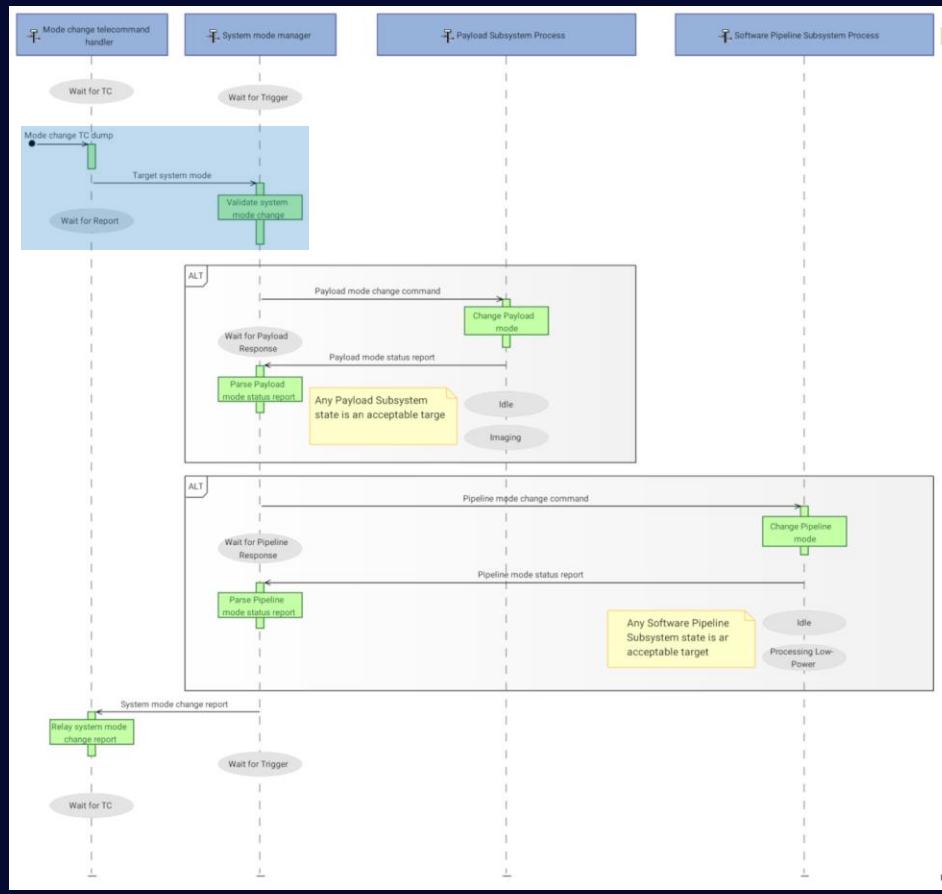


TASTE:
Message Sequence Chart

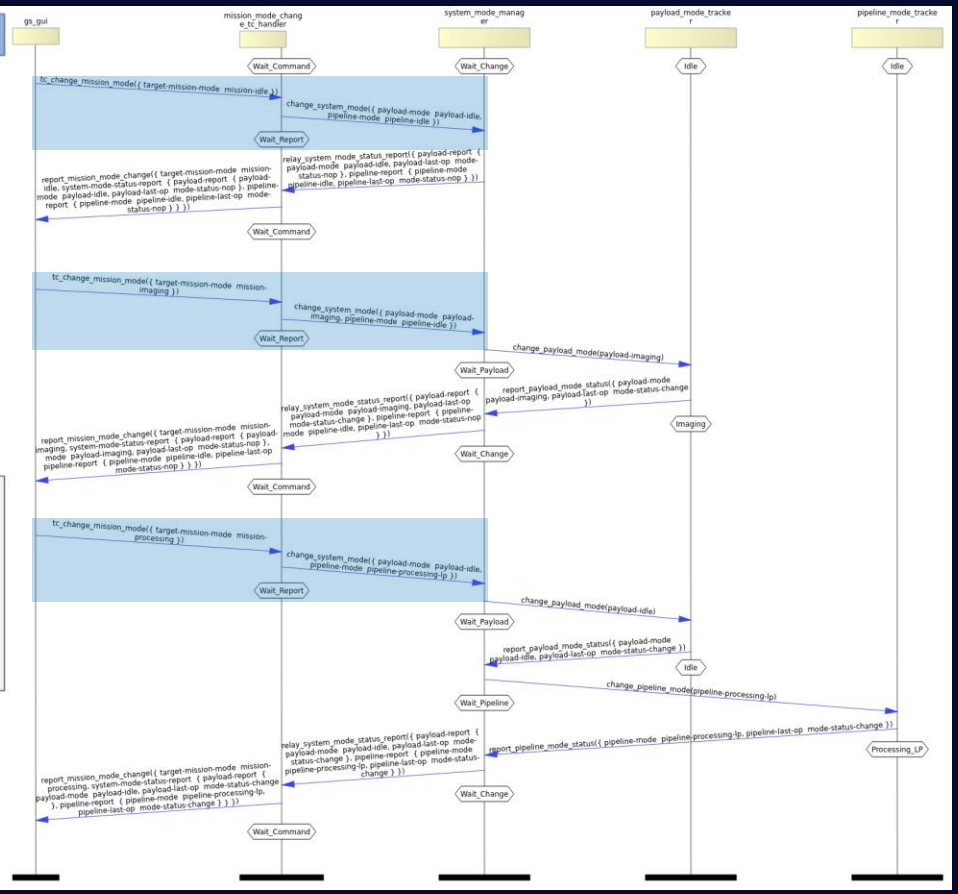


The Bridge: Exchange Verifications

Capella:
Physical [ES]

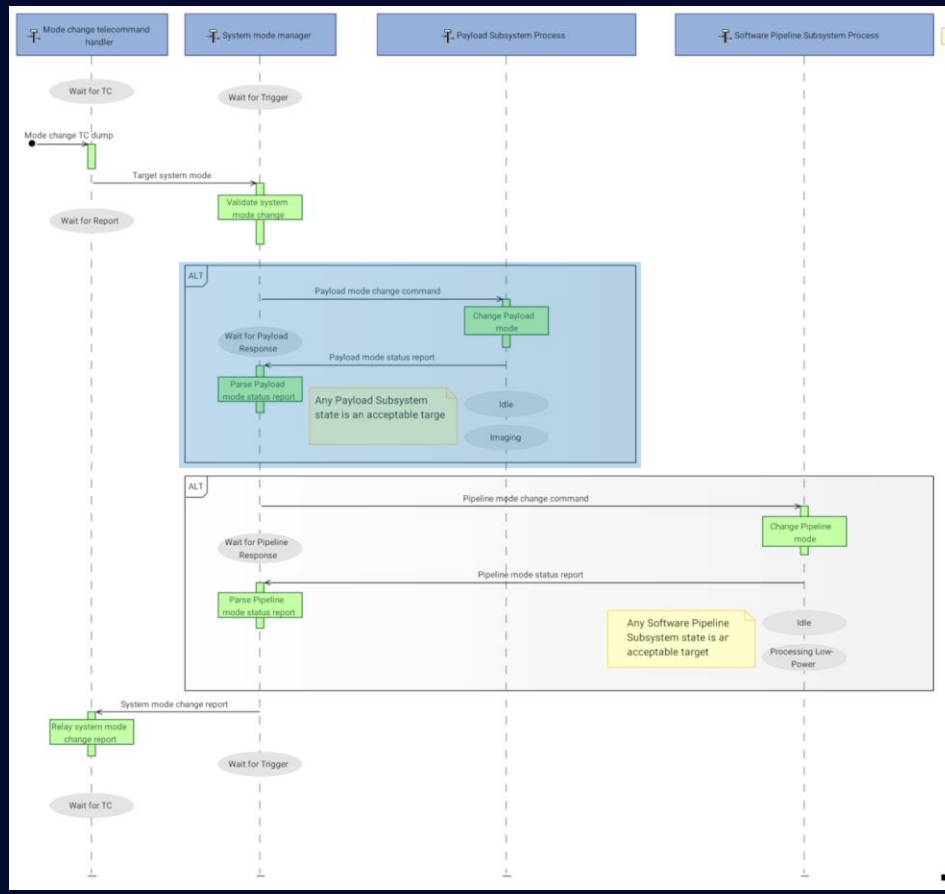


TASTE:
Message Sequence Chart

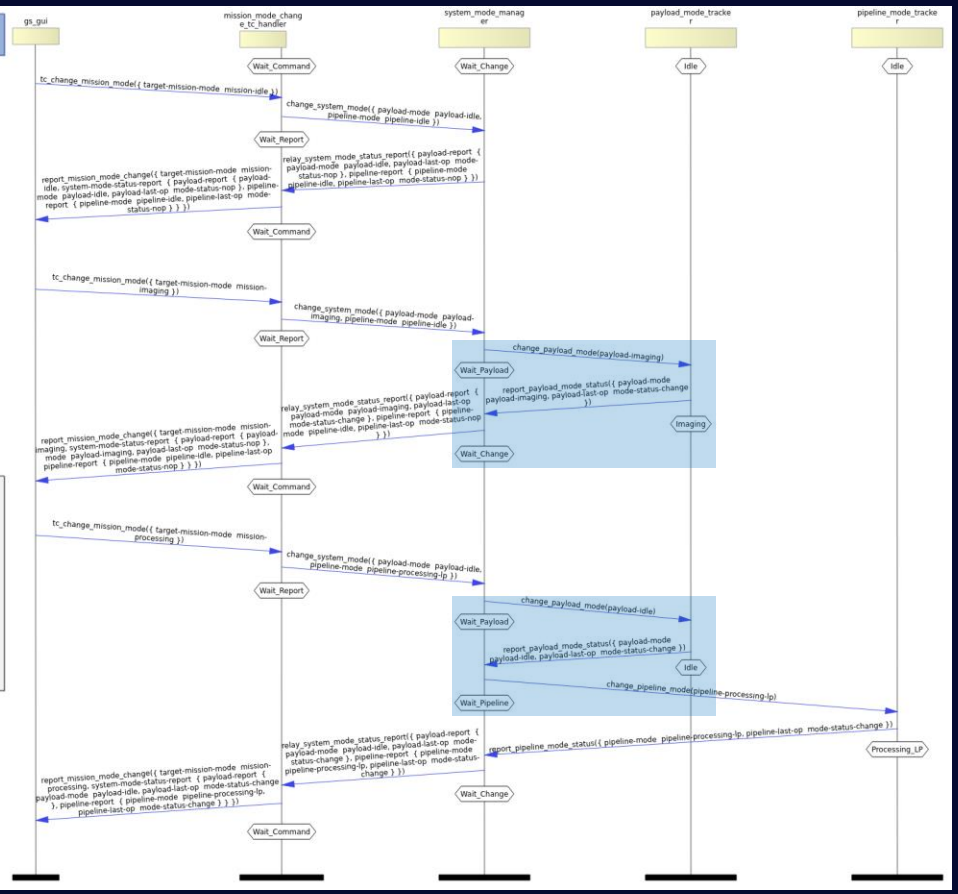


The Bridge: Exchange Verifications

Capella:
Physical [ES]

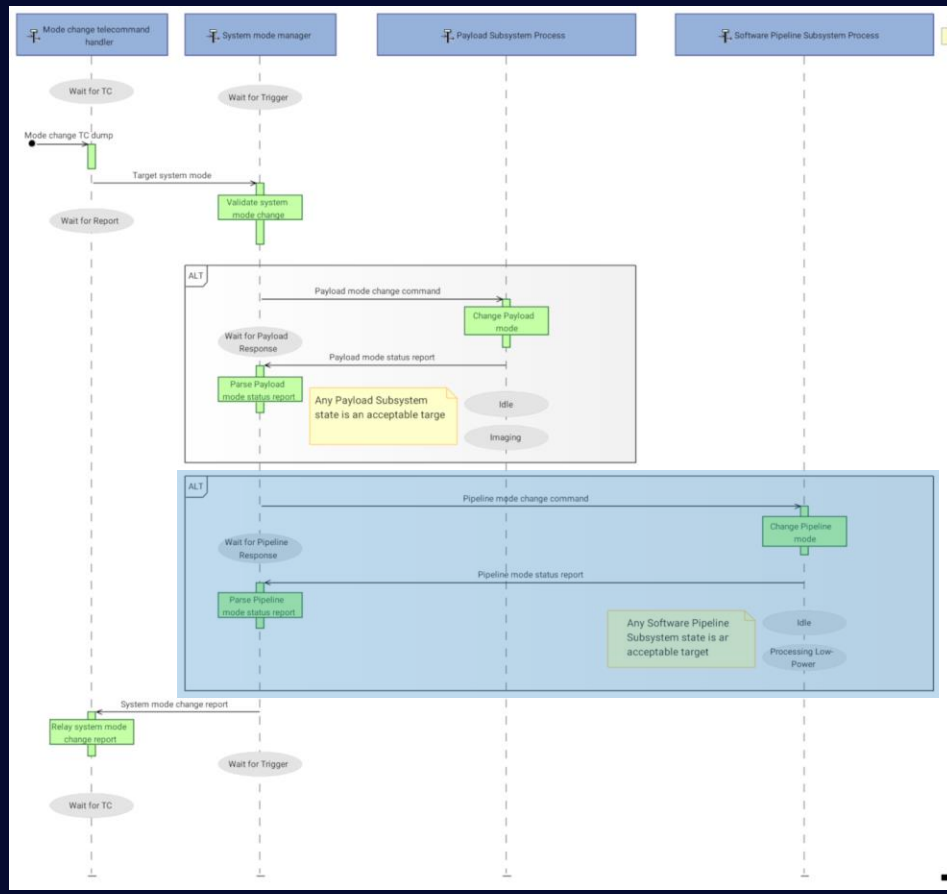


TASTE:
Message Sequence Chart

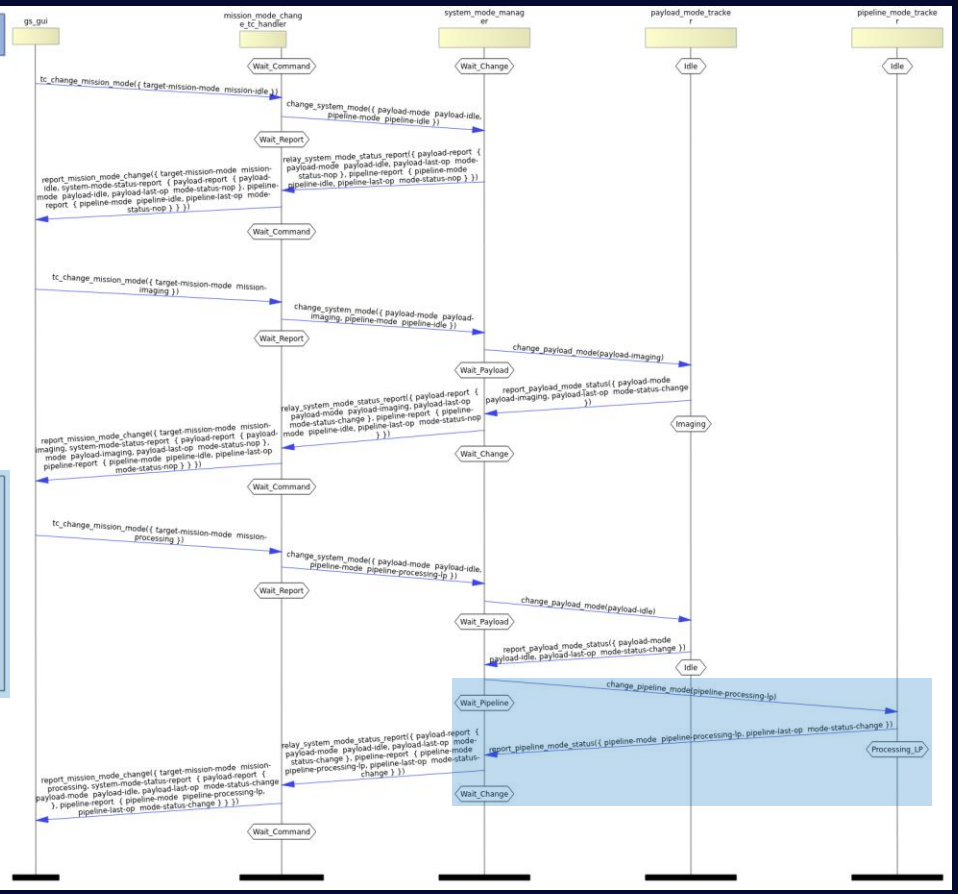


The Bridge: Exchange Verifications

Capella:
Physical [ES]

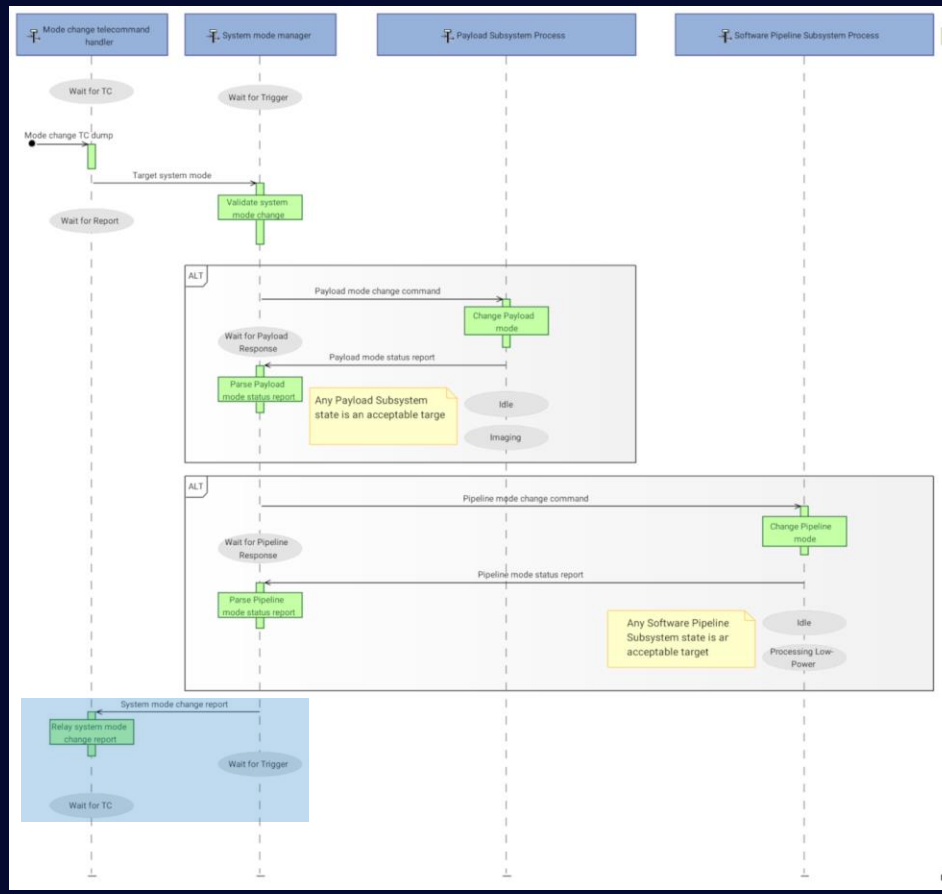


TASTE:
Message Sequence Chart

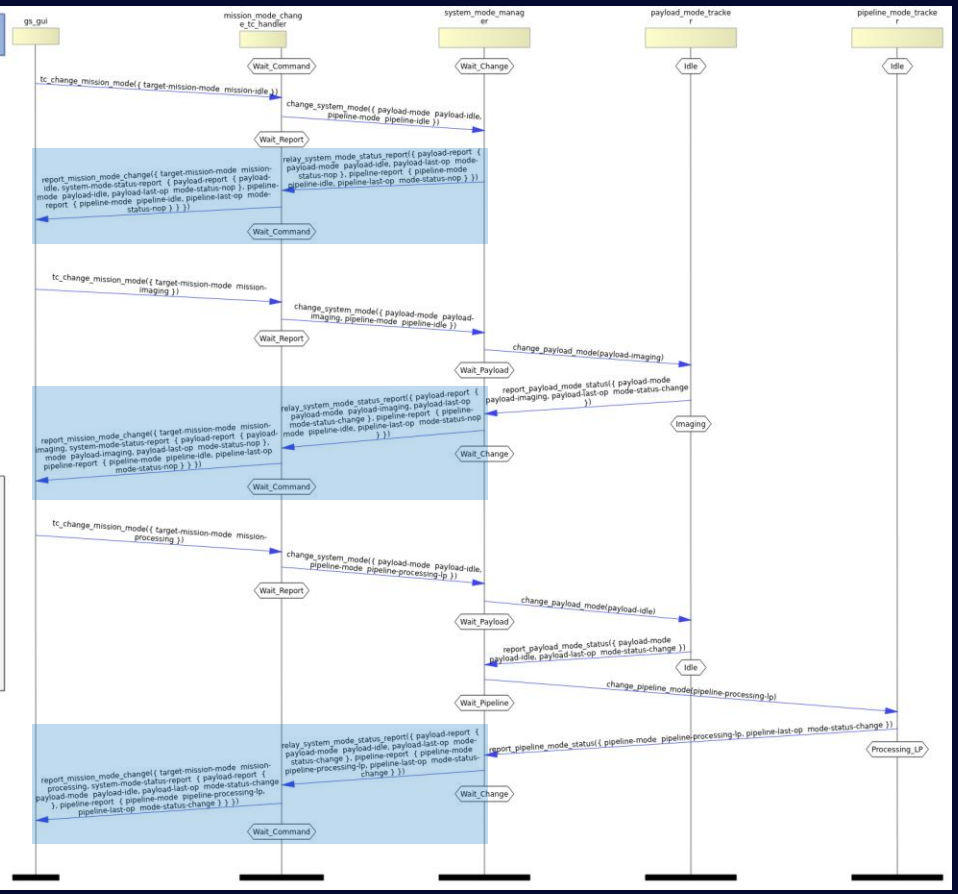


The Bridge: Exchange Verifications

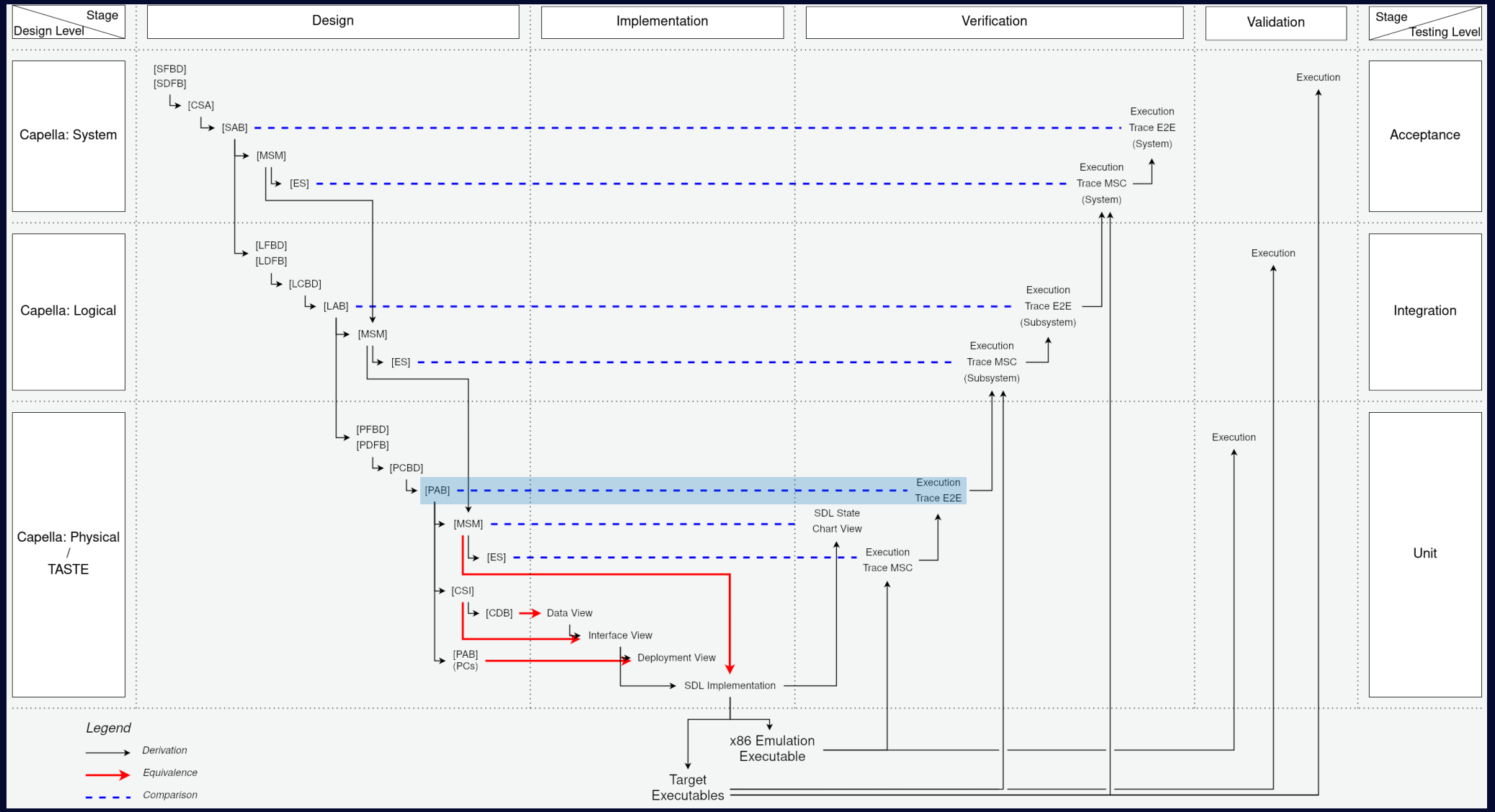
Capella:
Physical [ES]



TASTE:
Message Sequence Chart

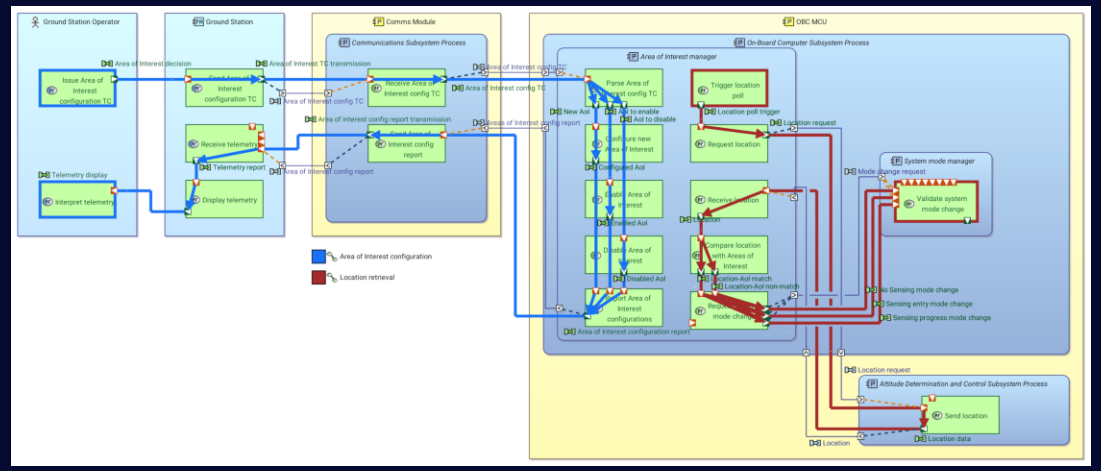


The Bridge: Impact Verifications

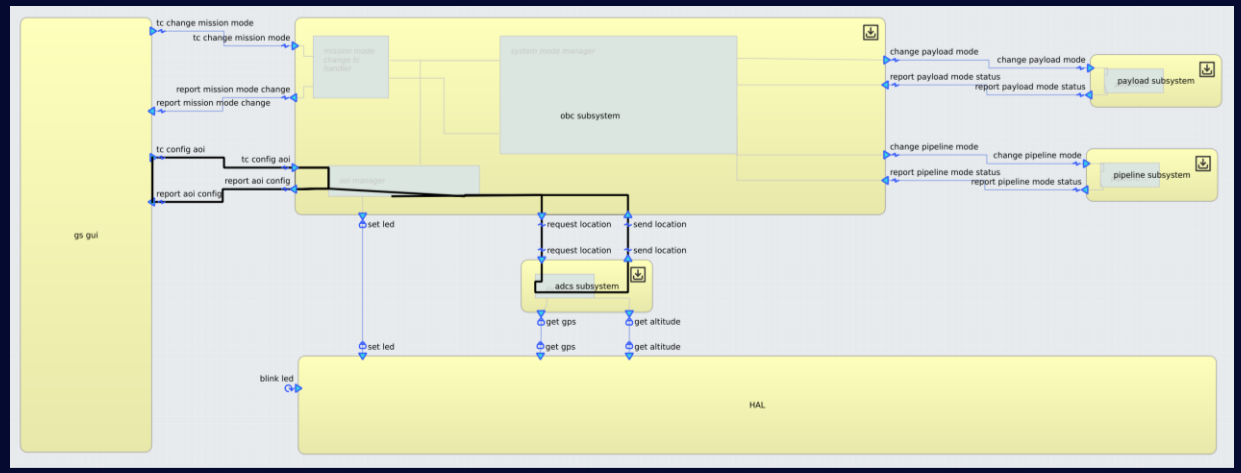


The Bridge: Impact Verifications

Capella:
[PAB] (highlighting FCs)

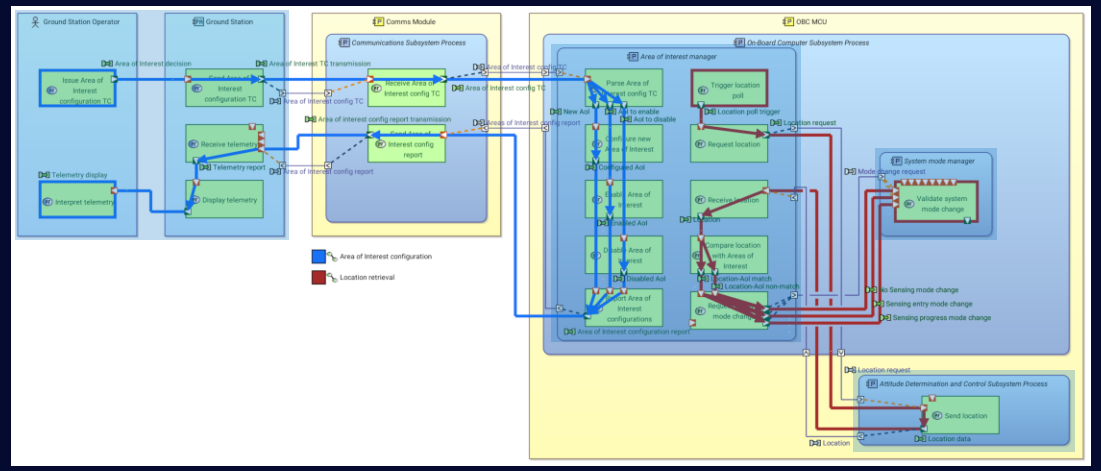


TASTE:
End-to-End Data View

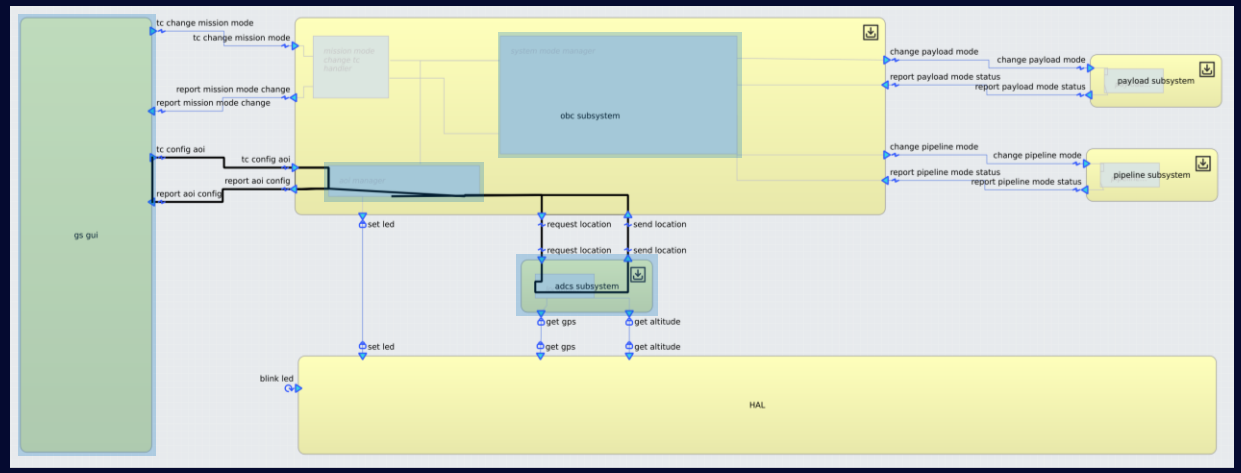


The Bridge: Impact Verifications

Capella:
[PAB] (highlighting FCs)

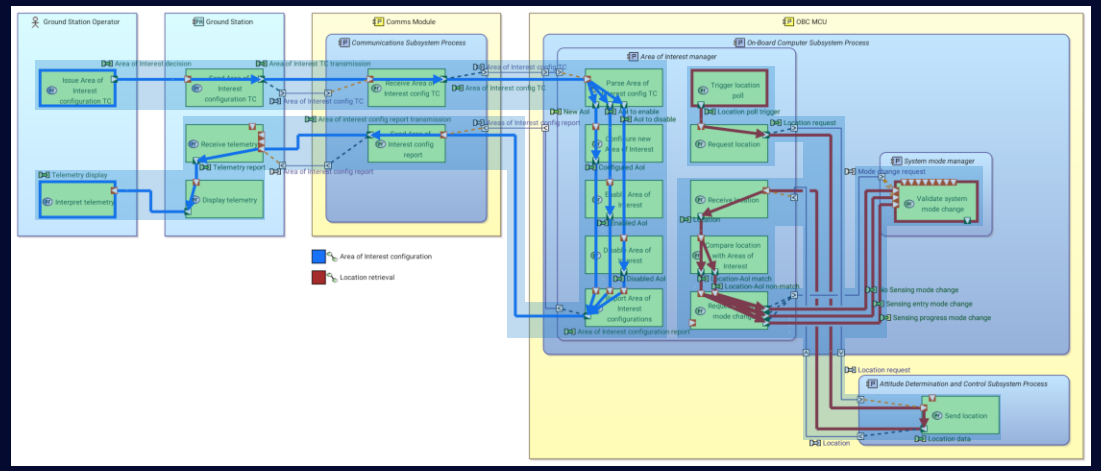


TASTE:
End-to-End Data View

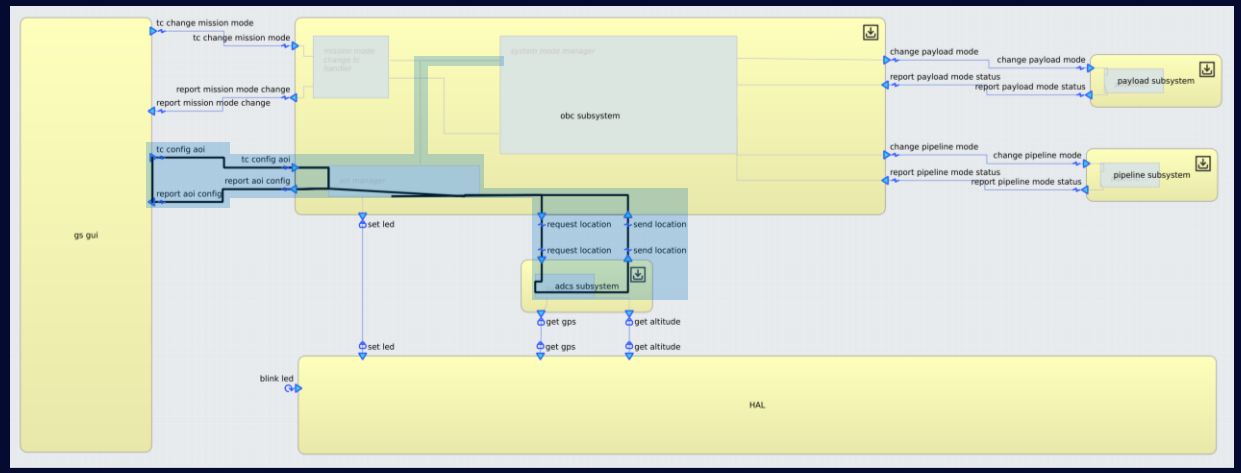


The Bridge: Impact Verifications

Capella:
[PAB] (highlighting FCs)

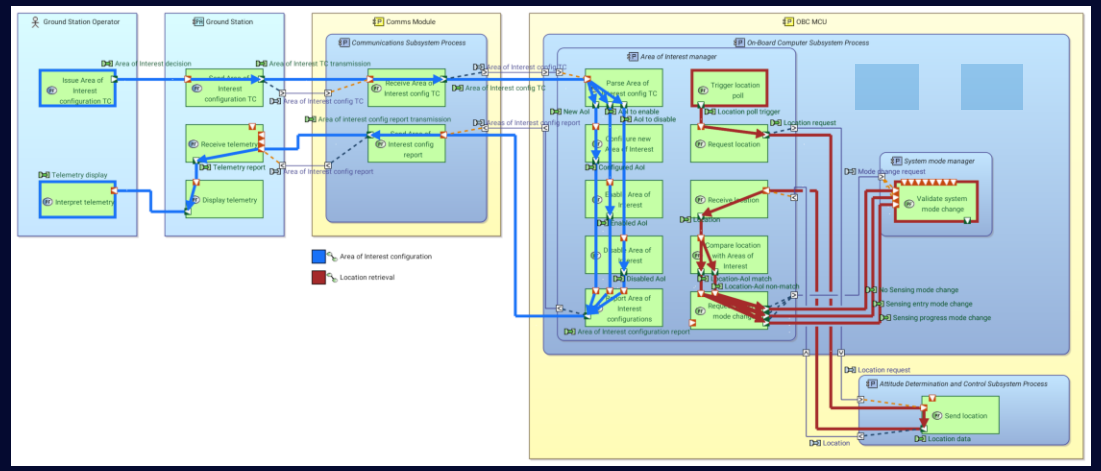


TASTE:
End-to-End Data View

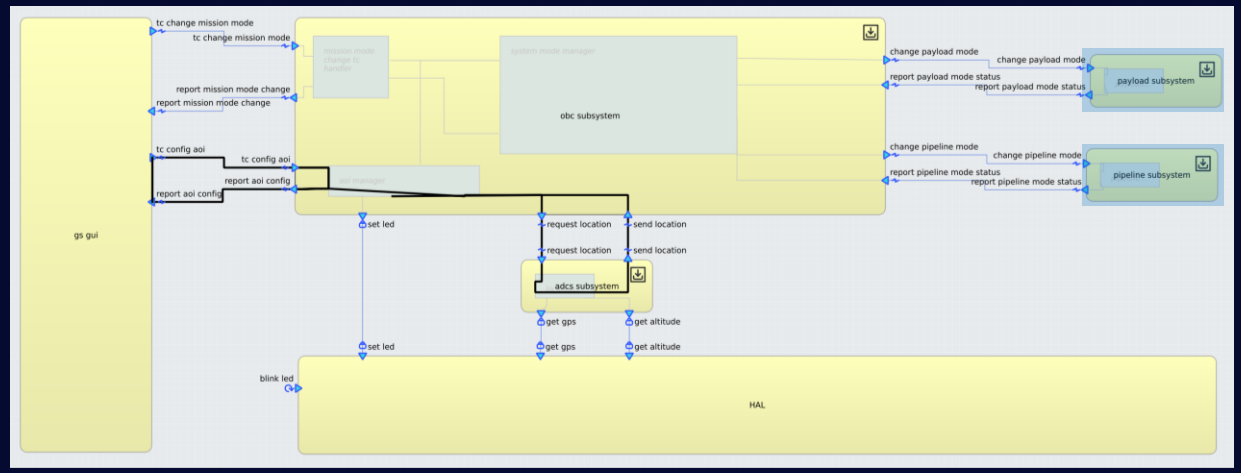


The Bridge: Impact Verifications

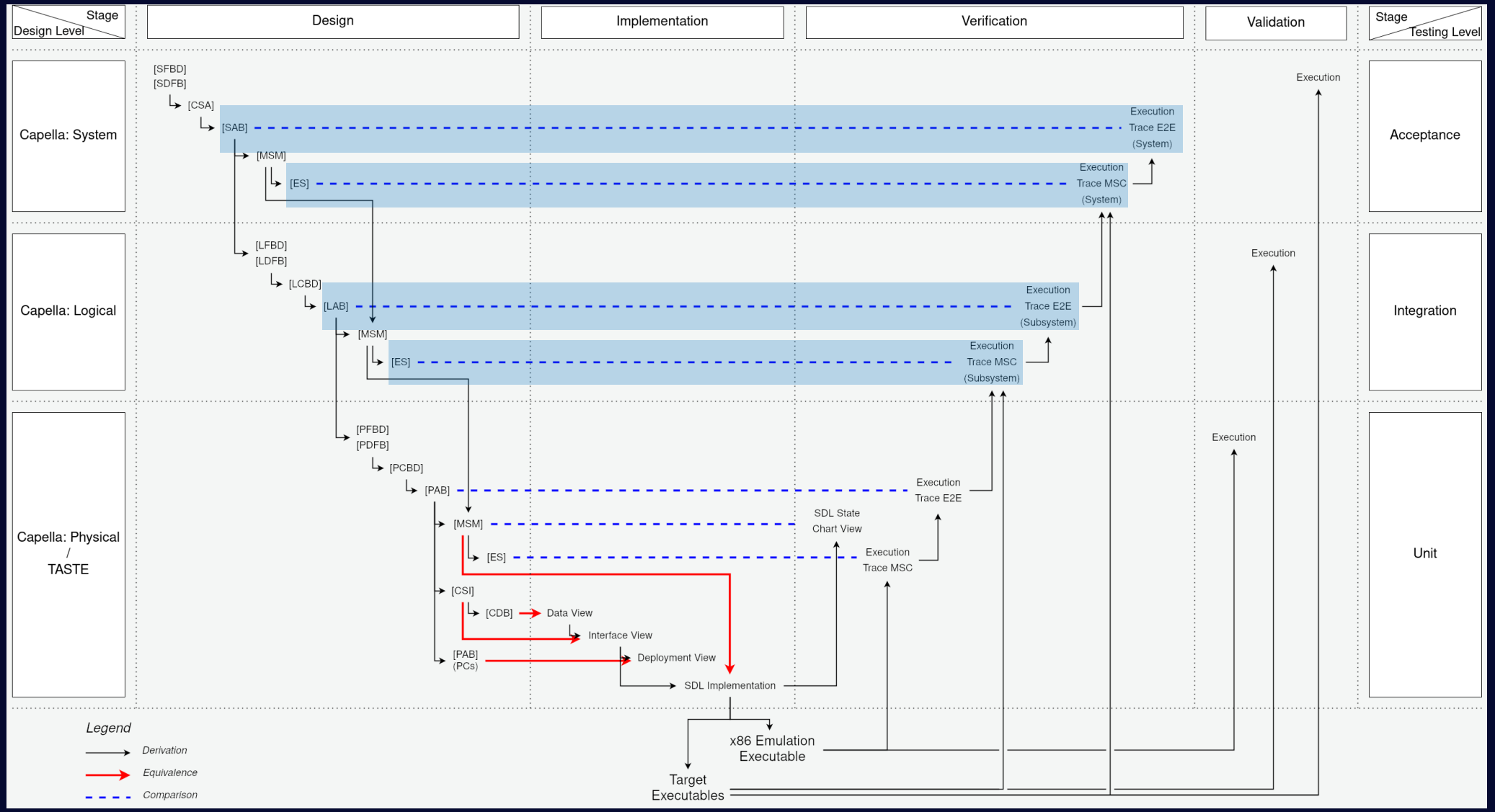
Capella:
[PAB] (highlighting FCs)



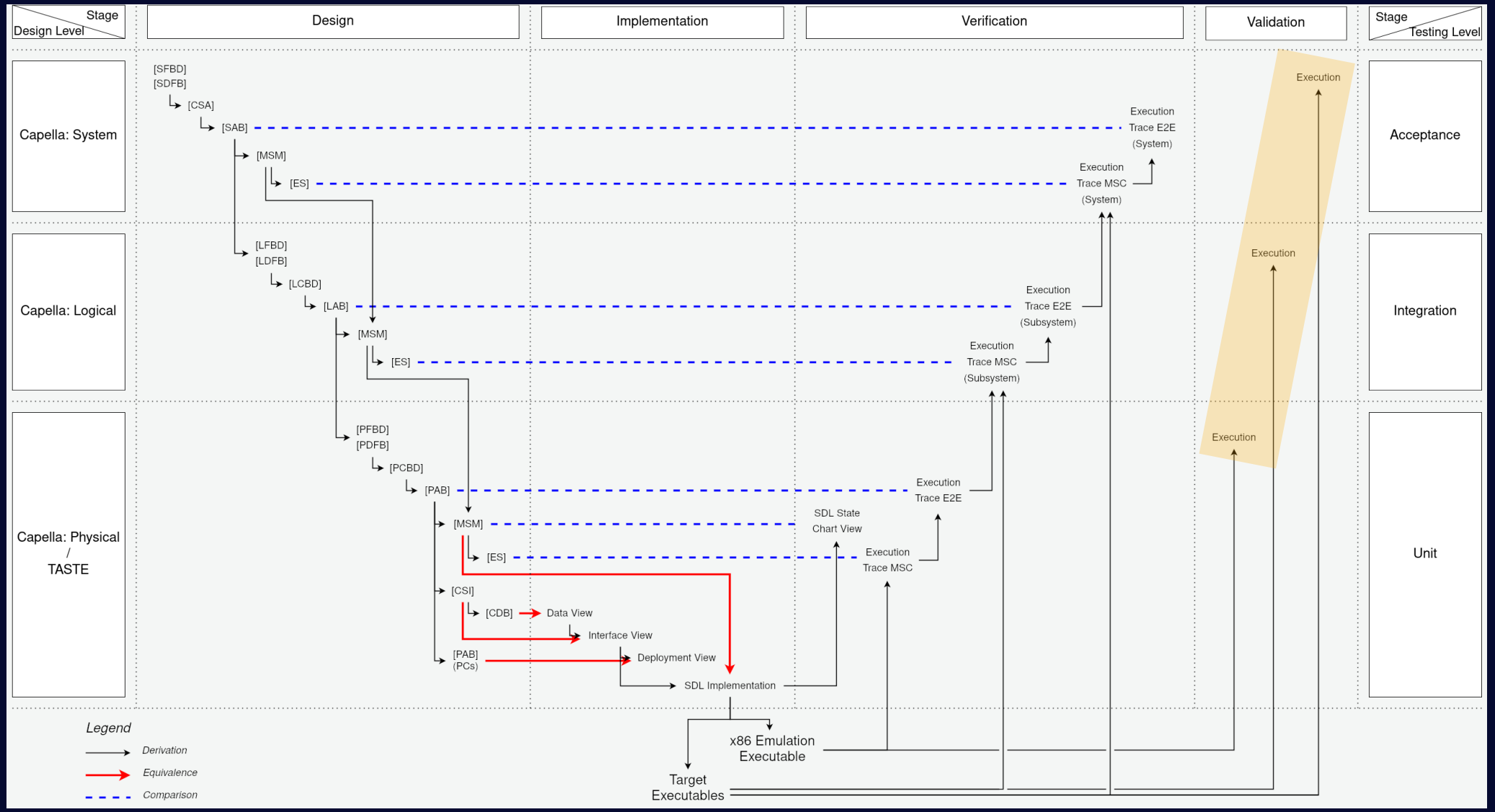
TASTE:
End-to-End Data View



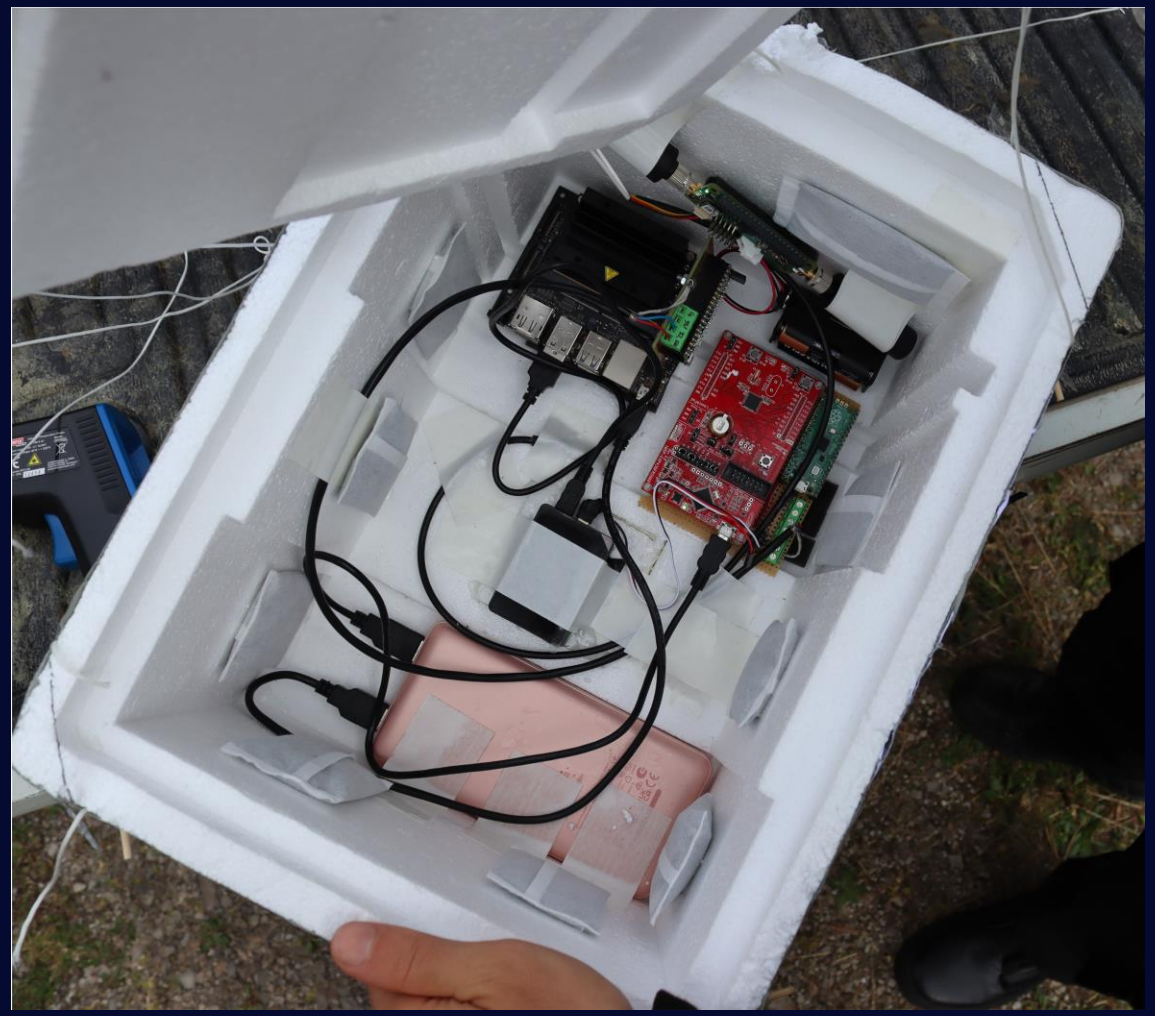
The Bridge: Higher-Level Verifications



The Bridge: Validations



The Bridge: Validations



The Bridge: Validation Results

The Bridge: Advantages

- Upstream shift:
  implementation activities → design model
- Thorough system analysis
- Easier debugging
  - Function implementations
  - Architectural flaws (including async!)
- It works!

The Bridge: Limitations and Further Works

- Manual labour
  - Time-consuming
  - Entrypoint for bugs
  - (by design)
- Automation
  - Implementations
  - Verifications
- Behaviour verification exclusive to component level
- Extension
  - Formal verification
  - Higher levels
    - Subsystem
    - System

Giorgio Ciacchella

Thank you for your attention!

10’ for questions ☺

Adopting Model-Based Practices with Capella and TASTE for Student-Developed CubeSat Systems

How MBSE is Helping GU Orbit to Design an Autonomous Nanosatellite

Resources

- Capella-TASTE Method
  - [1] IAC-23 paper
  - [2] IAC-23 iPoster
- Capella model
  - GitHub repo
  - Browsable HTML export
- TASTE model
  - GitHub repo