Capella

by Artal

kinéis

# Optimization of Kinéis (nano-satellites constellation) system tests using MBSE approach

Jonathan Lasalle - ARTAL Technologies

# The actors: CNES / Kinéis / Artal

cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES

- Designs / manages complex space systems
- First satisfying Capella experience some months ago
- MBSE spreading in progress

# The actors: CNES / Kinéis / Artal

CENTRE NATIONAL D'ÉTUDES SPATIALES

- Designs / manages complex space systems
- First satisfying Capella experience some months ago
- MBSE spreading in progress

- Strongly linked to the CNES
- Currently designs a new complex space system

# The actors: CNES / Kinéis / Artal

CNES
CENTRE NATIONAL D'ÉTUDES SPATIALES

- Designs / manages complex space systems
- First satisfying Capella experience some months ago
- MBSE spreading in progress

kinéis

- Strongly linked to the CNES
- Currently designs a new complex space system

artal technologies

- Capella expertise: training / coaching
- Capella customization

# History: introducing Capella inside CNES SVOM project

- CNES/Artal: first MBSE collaboration (2018->2021)
  - SVOM: Space system dedicated to gamma ray detection
    - Reverse engineering: historical processes vs MBSE principles comparison
    - Operational capture of system test
      Development of Capella extensions dedicated to V&V

⇒ Promising results : to be validated on other cases

# History: introducing Capella inside CNES SVOM project

- CNES/Artal: first MBSE collaboration (2018->2021)
  - SVOM: Space system dedicated to gamma ray detection
    - Reverse engineering: historical processes vs MBSE principles comparison
    - Operational capture of system test  
      Development of Capella extensions dedicated to V&V

  ⇒ Promising results : to be validated on other cases

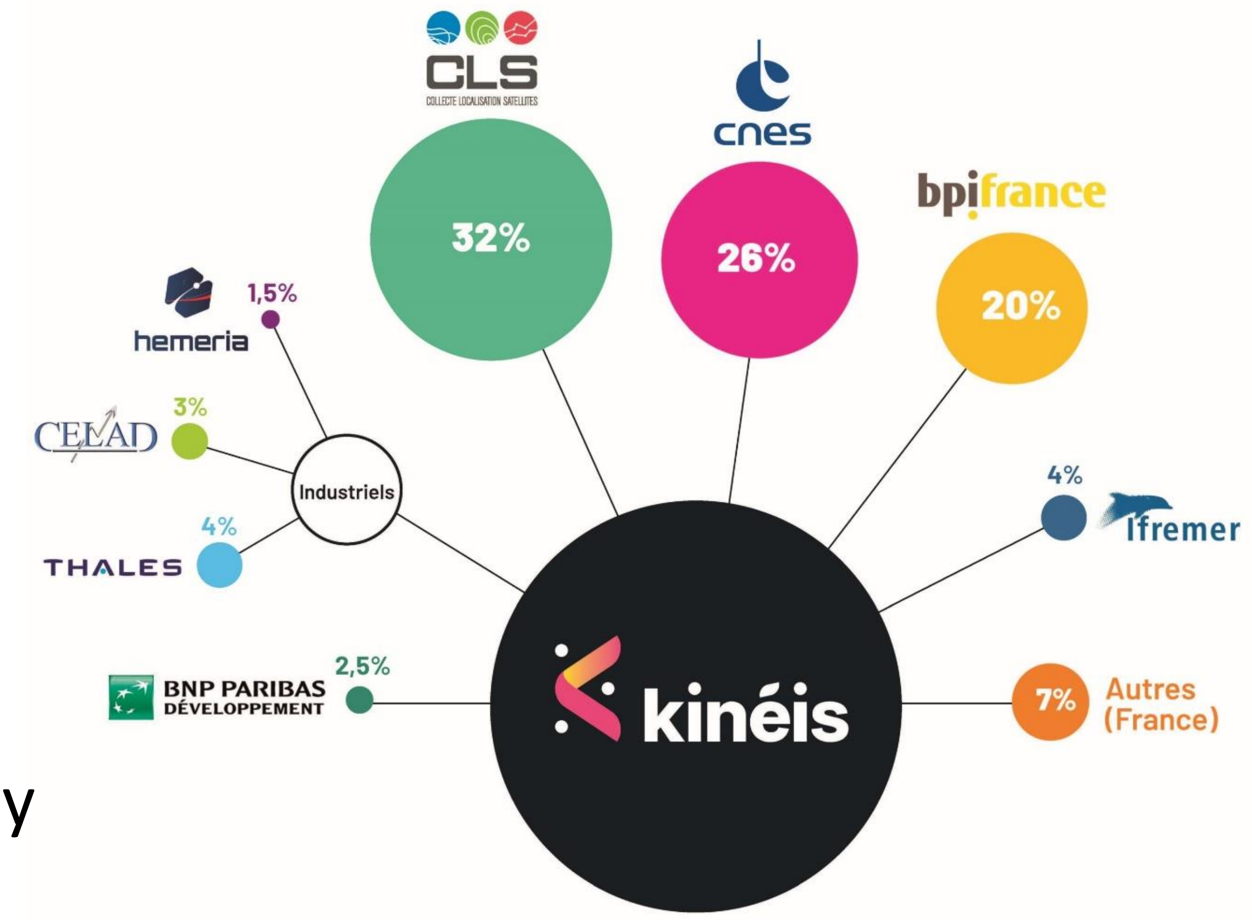
- Set-up of the Kinéis/Artal collaboration

⇒ **1st CNES POC on an operational project under development**

  - Capture Kineis system using Capella in order to identify the dedicated functional chains
  - Definition of validation objectives and associated test scenarios
  - Capella model being considered as single source of truth

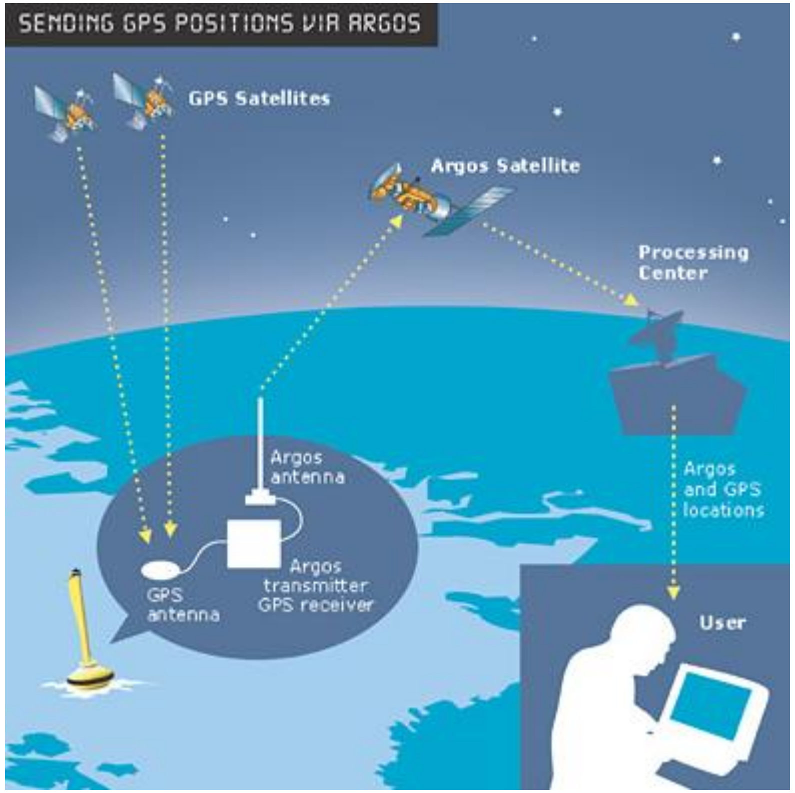
# Kinéis company

- **Kinéis**
  - Created in 2018
  - Where NewSpace meets the IoT (Internet of Things)
  - ~ 50 employees (+150% in 18 months)
  - Initiated by:
    - the CNES (French Space Agency)
    - CLS (*Collecte, Localisation, Satellites*)
- **Goal**
  - Democratization of Argos technology
  - Extend it to the entire IoT market

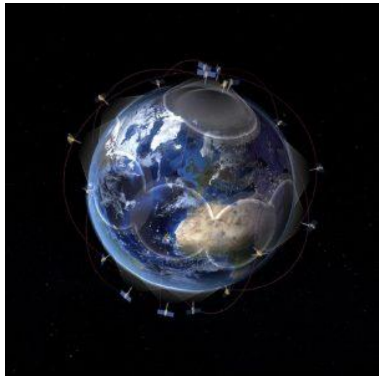


# ARGOS system

- Initiated in 1978
- Dedicated to studying, monitoring and protecting our planet's environment
- Structure
  - Space segment: GPS and Argos satellites
  - Ground segment: transmitters and process centers
- Data collection from various devices around the world
  - 7 Argos Satellites
  - 22 000 active transmitters (8 000 dedicated to animal tracking)
  - Over 100 countries
  - 3 Argos generation coexist: ARGOS 2, ARGOS 3 and ARGOS 4
  - Revisit time: 1H30 / 2H (satellite connection period)



# Kinéis main goal



- Kinéis main purpose:
  - Extension of the ARGOS system to handle IoT principles
  - Development, production and launch into orbit: 25 new nanosatellites
  - Installation of 20 ground stations around the globe
  - Revisit time => 5 min / 15 min (instead of 1H30 / 2H)
  - Upgrade of the IT infrastructures
- Challenges:
  - Design the new system
  - Validate and launch the constellation
  - Become a key actor for the IoT market

- MBSE process
  - Use the Capella model as the (single) source of truth for system tests

⇒Support the system tests need using Capella models

# Reminder: main benefits of MBSE/Capella approach

Why using models instead of regular documents to describe the system ?

# Reminder: main benefits of MBSE/Capella approach

Why using models instead of regular documents to describe the system ?

Communicate: specify and clarify each system components perimeter and expectations using a formal language to avoid ambiguities

# Reminder: main benefits of MBSE/Capella approach

Why using models instead of regular documents to describe the system ?

Communicate: specify and clarify each system components perimeter and expectations using a formal language to avoid ambiguities

Secure : being able to ensure the consistency and the completeness of the system test definition by using traceability mechanism

# Reminder: main benefits of MBSE/Capella approach

Why using models instead of regular documents to describe the system ?

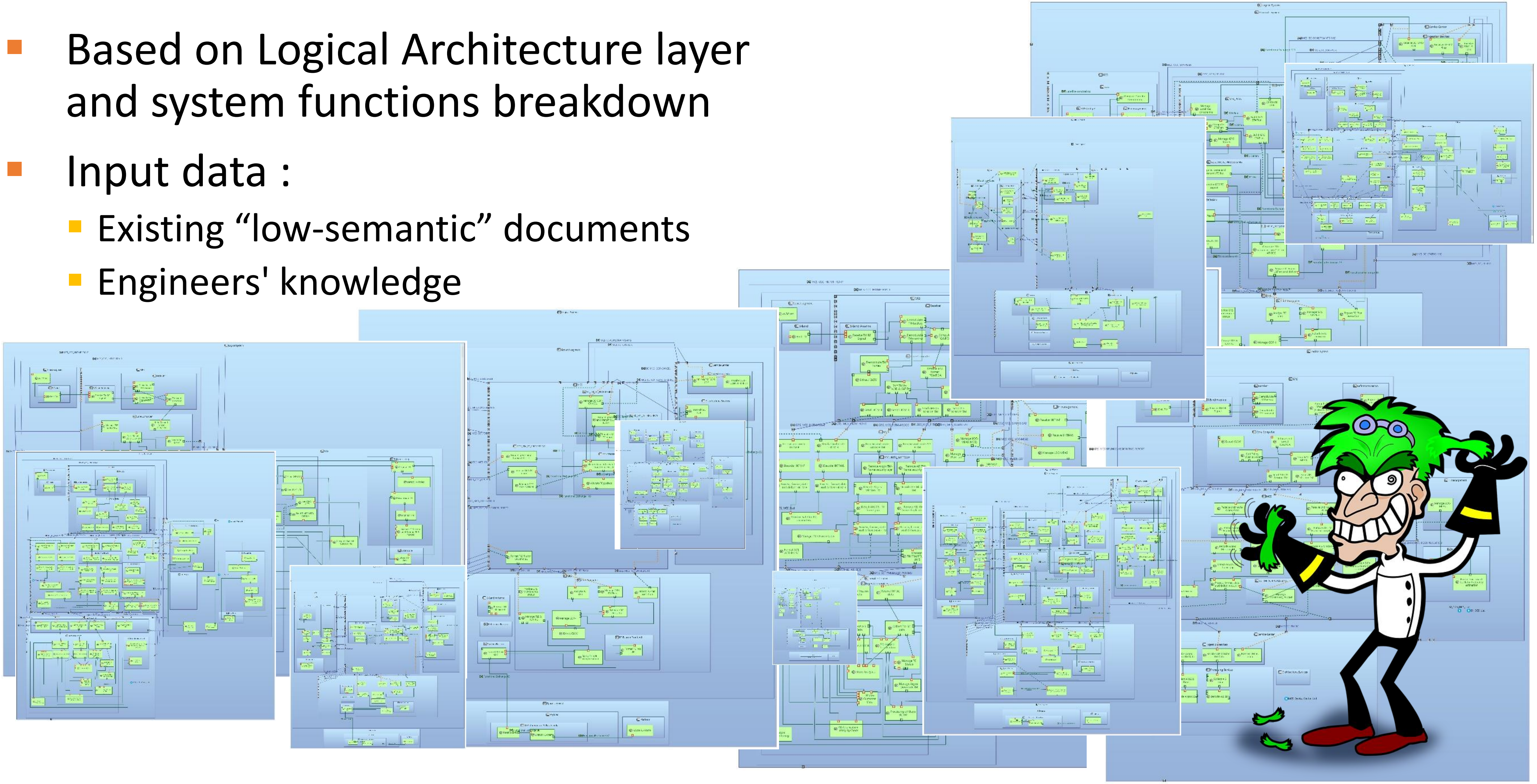
Communicate: specify and clarify each system components perimeter and expectations using a formal language to avoid ambiguities

Secure : being able to ensure the consistency and the completeness of the system test definition by using traceability mechanism

Generate: take advantage of the formal description to generate assets required for system test manual execution and archiving

# Step 1: model initialization (overall view)

- Based on Logical Architecture layer and system functions breakdown
- Input data :
  - Existing “low-semantic” documents
  - Engineers' knowledge

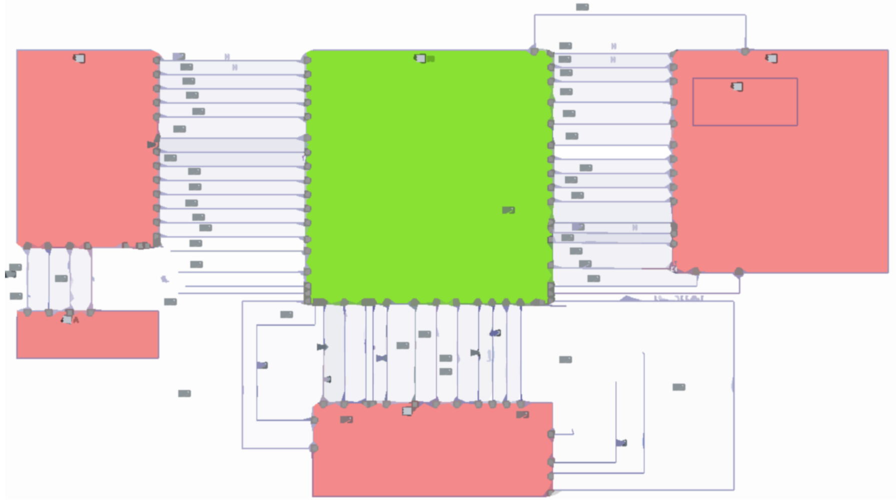


# Step 2: model reconciliation

- Two models were built in parallel:

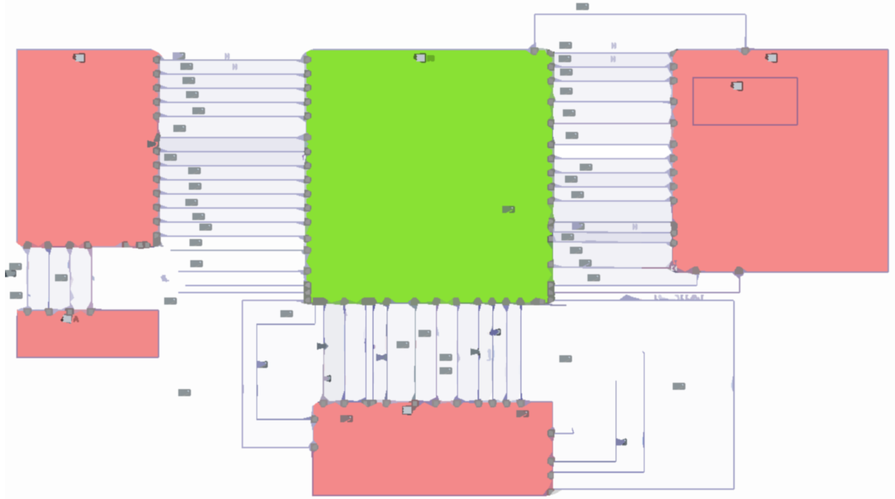
# Step 2: model reconciliation

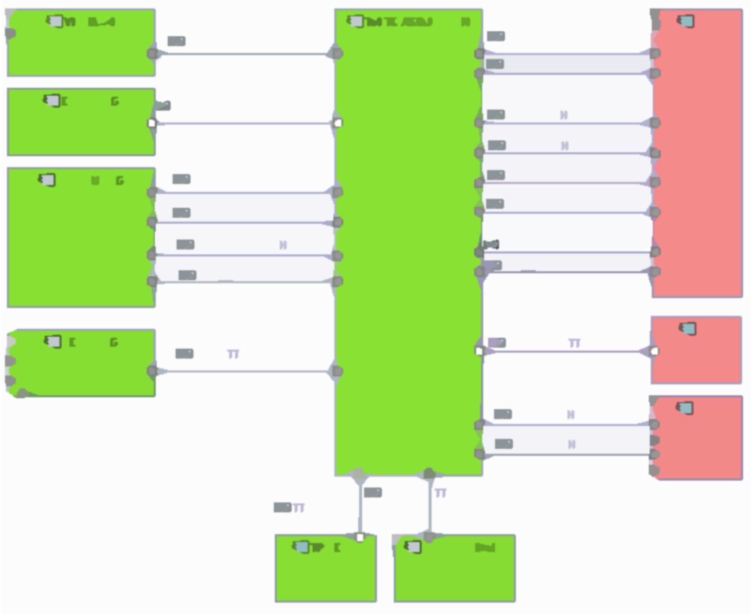
- Two models were built in parallel:
  - The main integration model (by Kineis)



# Step 2: model reconciliation

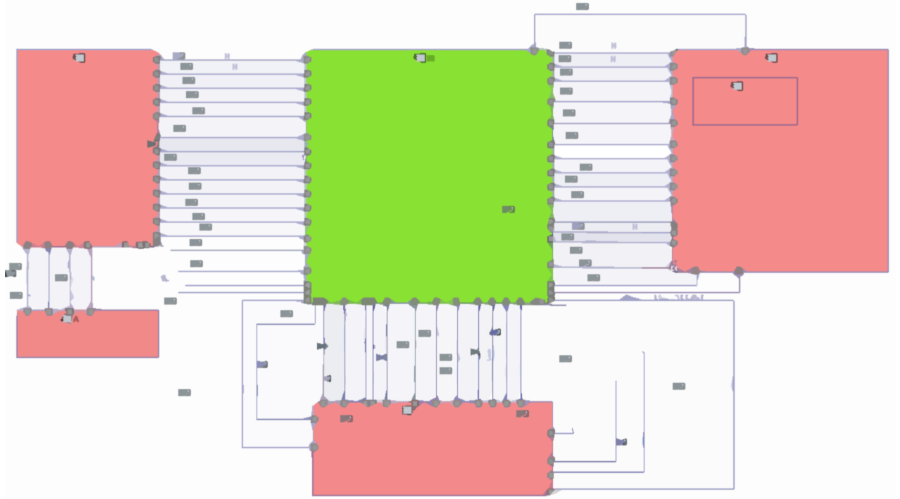
- Two models were built in parallel:
  - The main integration model (by Kineis)
  - The model of one of the sub-component (by Thales Alenia Space)

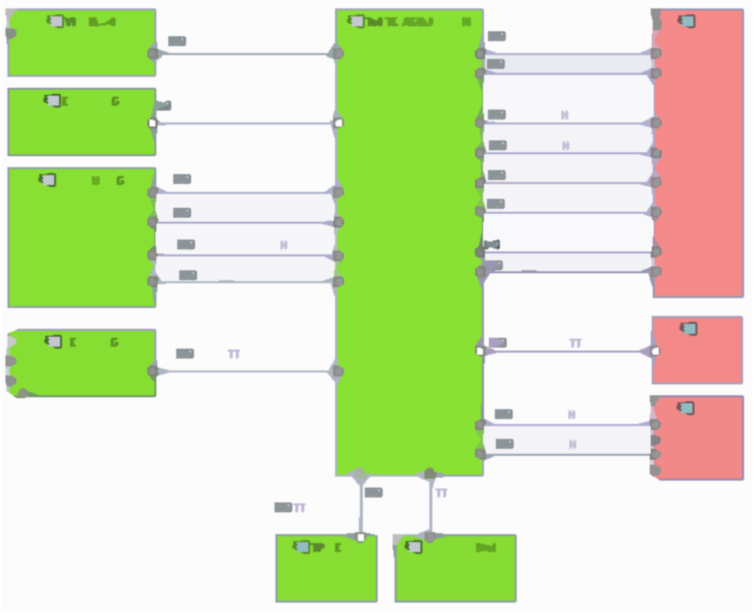




# Step 2: model reconciliation

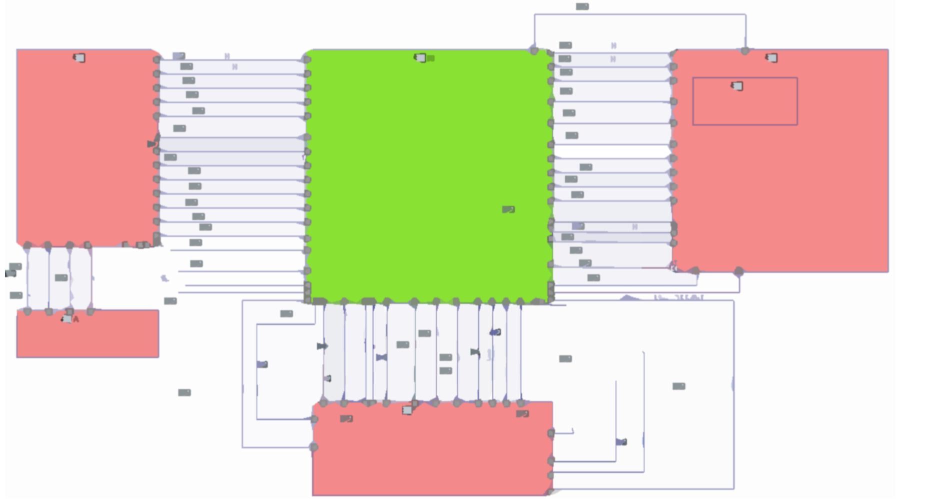
- Two models were built in parallel:
  - The main integration model (by Kineis)
  - The model of one of the sub-component (by Thales Alenia Space)
- Goals:
  - The main model must integrate sub-model “public interface” updates
  - The main model must partially integrates sub-items (to have a complete overall view)

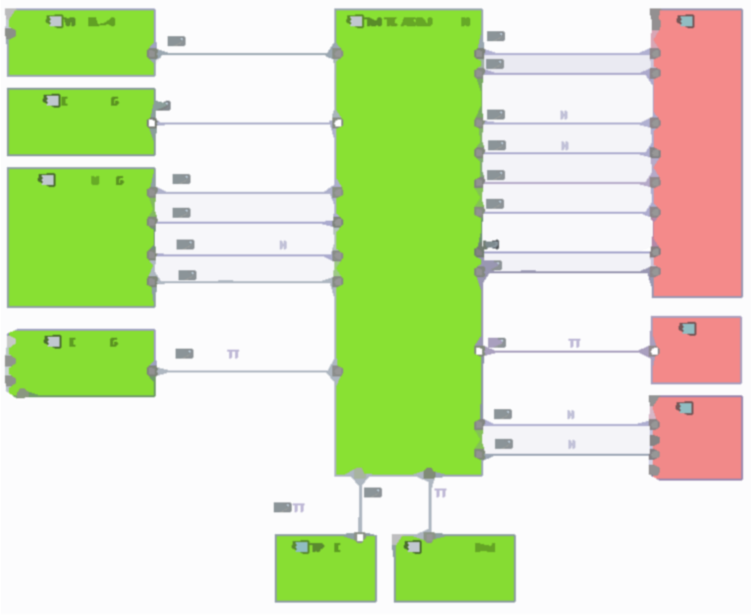




# Step 2: model reconciliation

- Two models were built in parallel:
  - The main integration model (by Kineis)
  - The model of one of the sub-component (by Thales Alenia Space)
- Goals:
  - The main model must integrate sub-model “public interface” updates
  - The main model must partially integrates sub-items (to have a complete overall view)
- Challenges:
  - How to link the two models ?
  - How to “automatically” inject modifications of the sub-model into the main one ? (Iteratively ?)





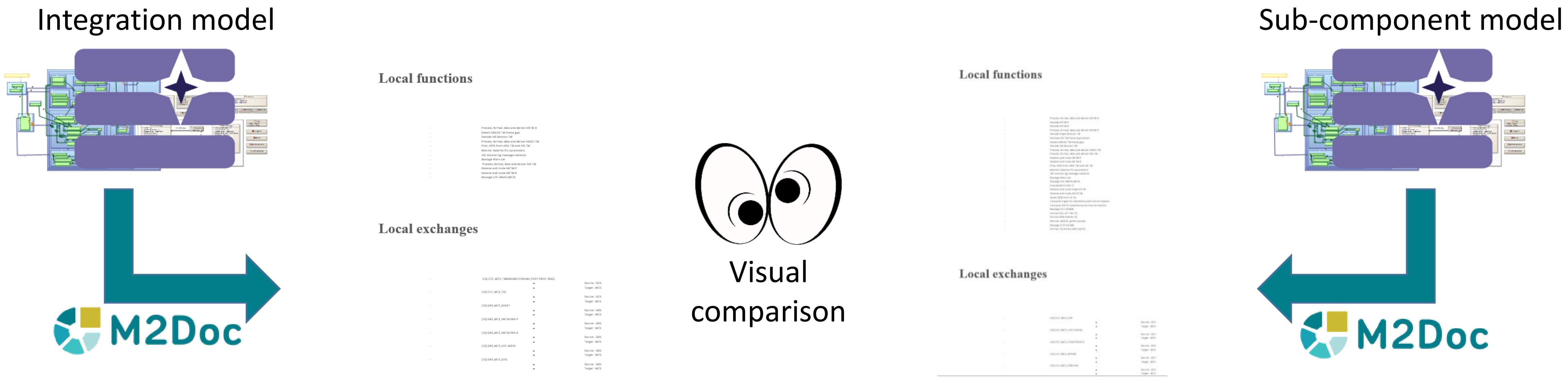
# Step 2: model reconciliation

- Solutions:
  - Updates the models to reconnect them using a Capella dedicated feature (REC/RPL)
  - Set traceability link “manually” between the two models
  - Implement a specific “home-made” synchronization algorithm
  - Implement Word document generation to easily compare them “manually”

# Step 2: model reconciliation

- Solutions:
  - Updates the models to reconnect them using a Capella dedicated feature (REC/RPL)
  - Set traceability link “manually” between the two models
  - Implement a specific “home-made” synchronization algorithm
  - Implement Word document generation to easily compare them “manually”

- Winner:



Integration model

Sub-component model

Local functions

Local exchanges

Local functions

Local exchanges

Visual comparison

# Step 2: model reconciliation

- Solutions:
  - Updates the models to reconnect them using a Capella dedi
  - Set traceability link “manually” between the two models
  - Implement a specific “home-made” synchronization algorithm
  - Implement Word document generation to easily compare them “manu

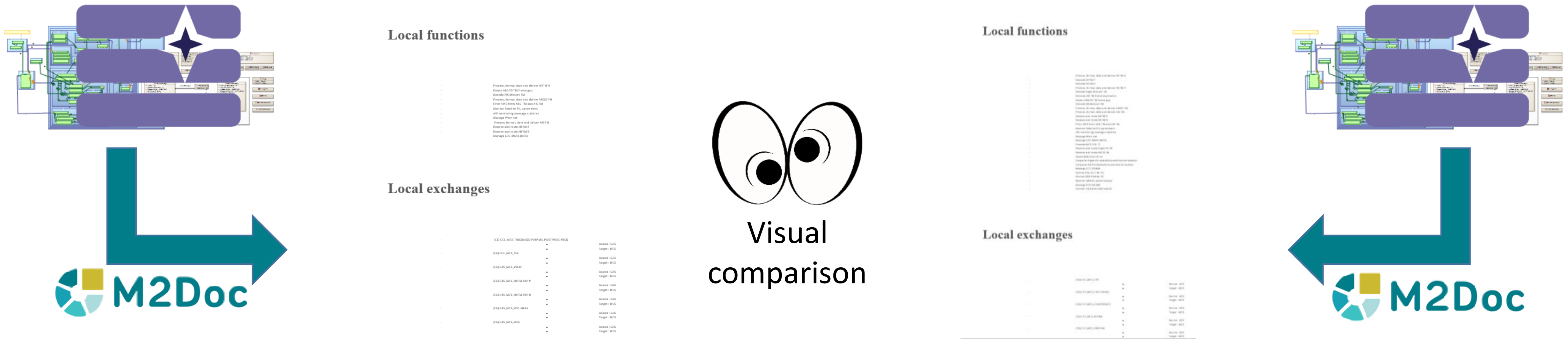
KINEIS feedback

Define the system MBSE model at the beginning of the program in order to work on the same MBSE project with the sub-contractor

- Winner:

Integration model

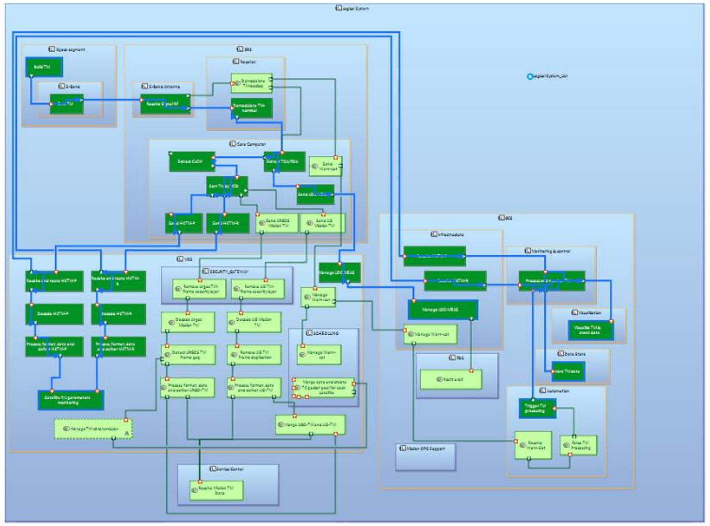
Sub-component model



Visual comparison

# Step 3: description of V&V objectives

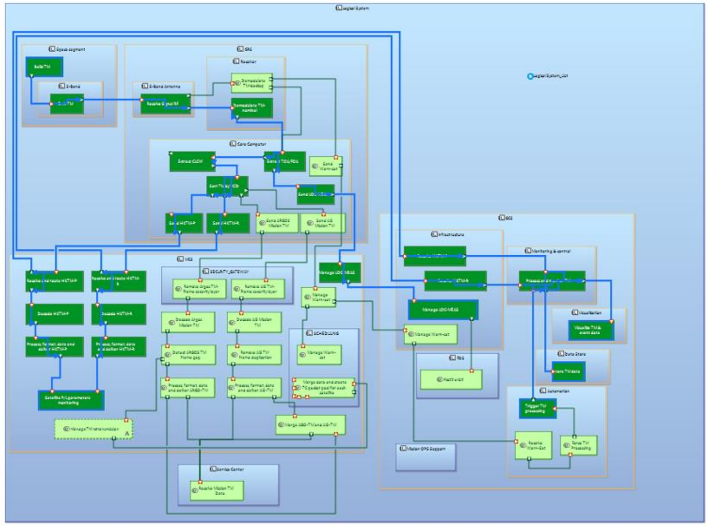
- Capture functional chains:
  - In order to highlight composite behaviors
  - In order to identify validation objectives
    - Entry point of the proposed process defined during the CNES/Artal collaboration



# Step 3: description of V&V objectives

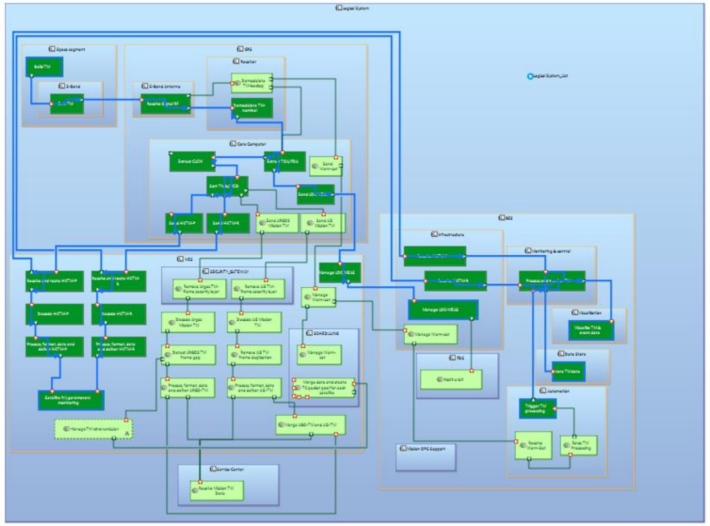
- Capture functional chains:
  - In order to highlight composite behaviors
  - In order to identify validation objectives
    - Entry point of the proposed process defined during the CNES/Artal collaboration

- Limitation: hard to read/understand for non-initiated people



# Step 3: description of V&V objectives

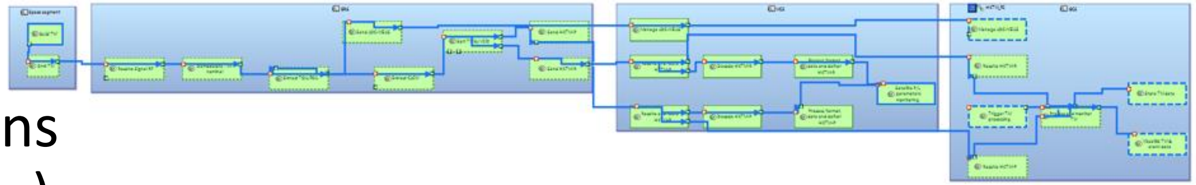
- Capture functional chains:
  - In order to highlight composite behaviors
  - In order to identify validation objectives
    - Entry point of the proposed process defined during the CNES/Artal collaboration



- Limitation: hard to read/understand for non-initiated people
  - Creation of dedicated representations (Focusing on a given functional chain)

    ⇒ To clarify inputs and functions to be validated for each sub-system

    ⇒ Assisted by a new dedicated tool that simplify their generation



# The system test process

- Application of the “system test” process defined during the CNES/Artal collaboration
- Main steps:
  - Capture the system architecture ✓

# The system test process

- Application of the “system test” process defined during the CNES/Artal collaboration
- Main steps:
  - Capture the system architecture ✓
  - Capture validation needs ✓
    - Identification of the functional chains to validate

# The system test process

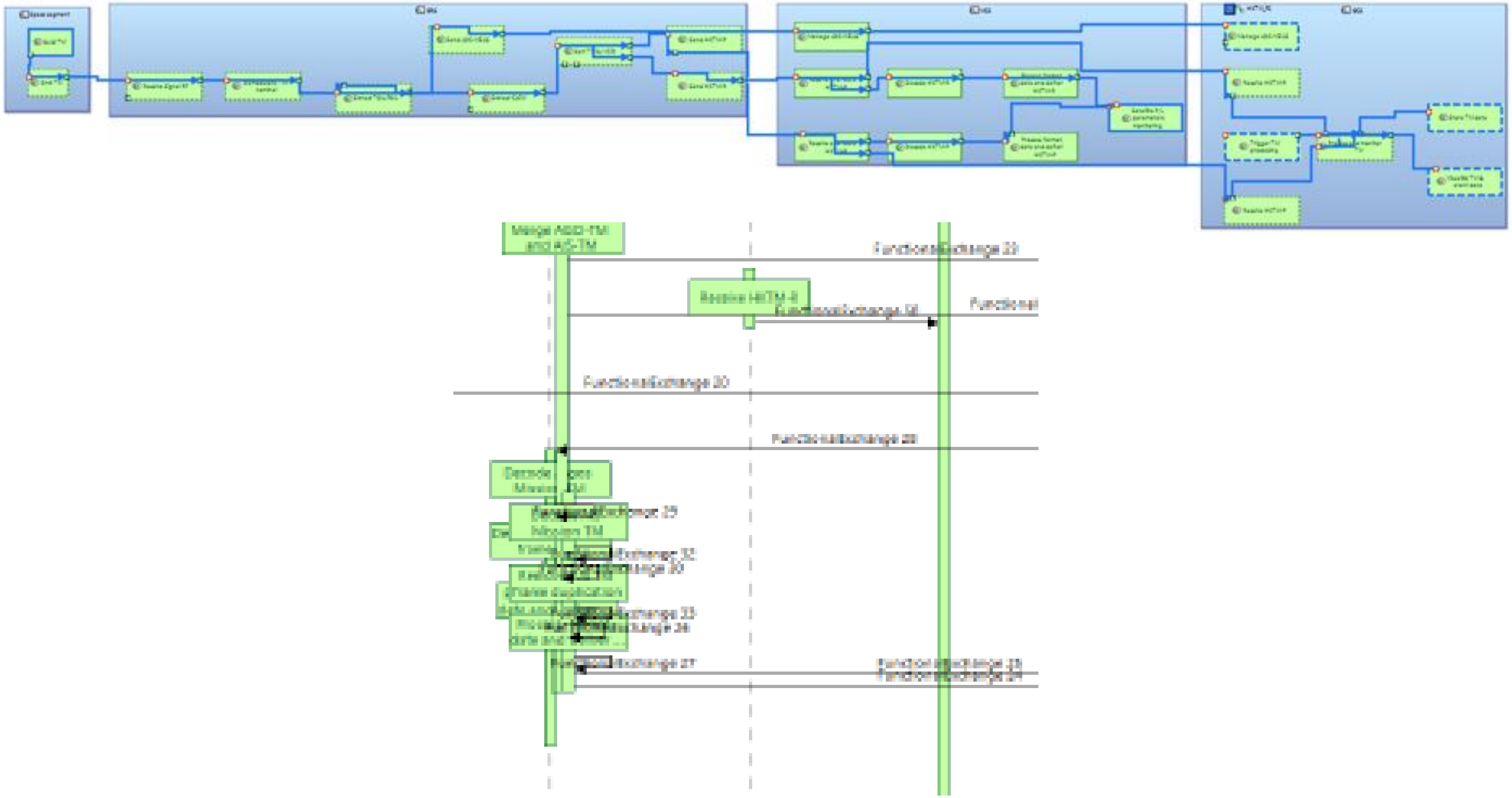
- Application of the “system test” process defined during the CNES/Artal collaboration
- Main steps:
  - Capture the system architecture ✓
  - Capture validation needs ✓
    - Identification of the functional chains to validate
  - Specification of corresponding tests (using exchange scenarios)
    - Scenarios can be generated by a dedicated “home made” tool

# Step 4: generation of test scenarios

- The CNES/SVOM process/tools not fully “compliant” with the Kinéis case

# Step 4: generation of test scenarios

- The CNES/SVOM process/tools not fully “compliant” with the Kinéis case

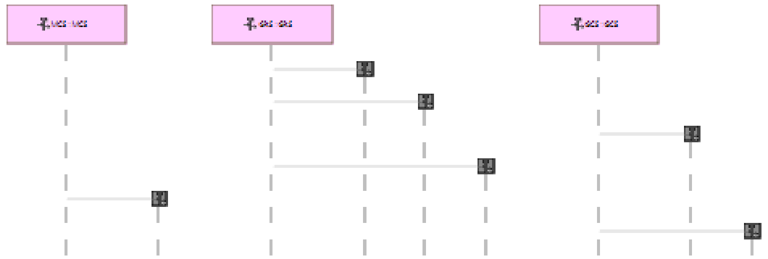


# Step 4: generation of “readable” test scenarios

- Cutomization of Capella behaviors in order to obtain usable scenarios

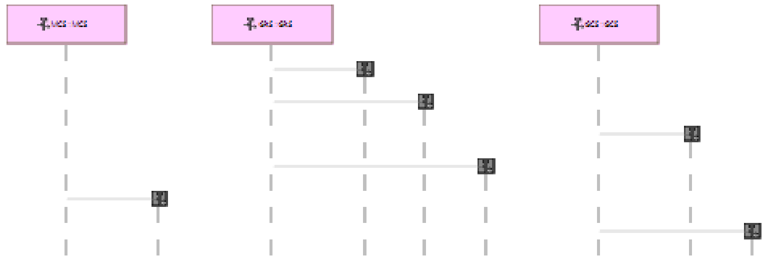
# Step 4: generation of “readable” test scenarios

- Cutomization of Capella behaviors in order to obtain usable scenarios
  - New “sub-lifeline” generation feature
    - Allows functions “superposition”
      Clear/determinist reading

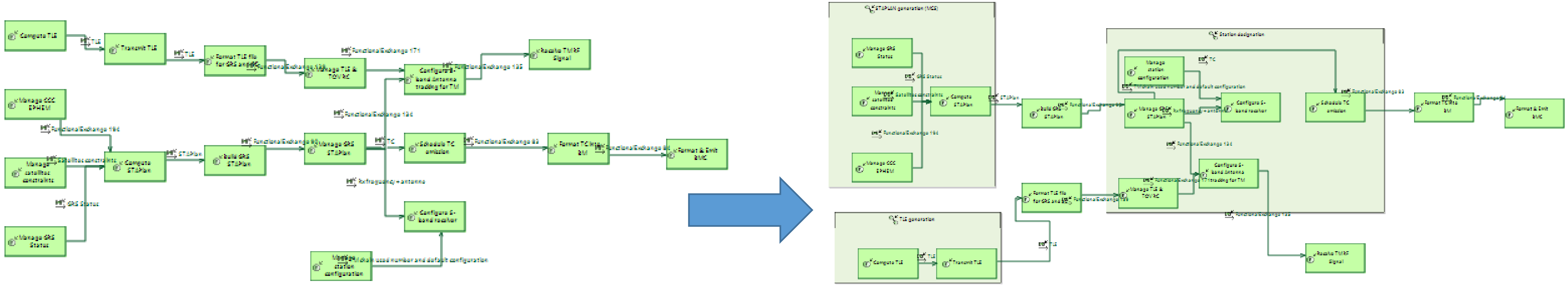


# Step 4: generation of “readable” test scenarios

- Cutomization of Capella behaviors in order to obtain usable scenarios
  - New “sub-lifeline” generation feature
    - Allows functions “superposition”
      Clear/determinist reading

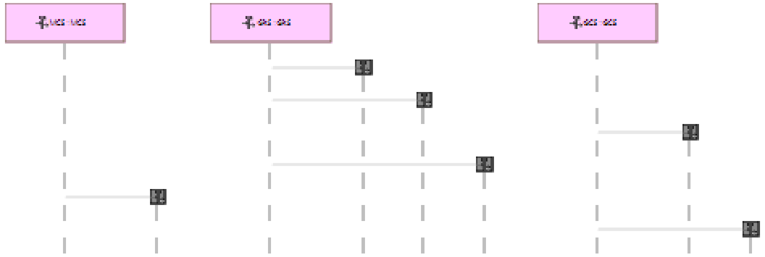


  - Injection of “sub-functional chain” support to compact scenarios

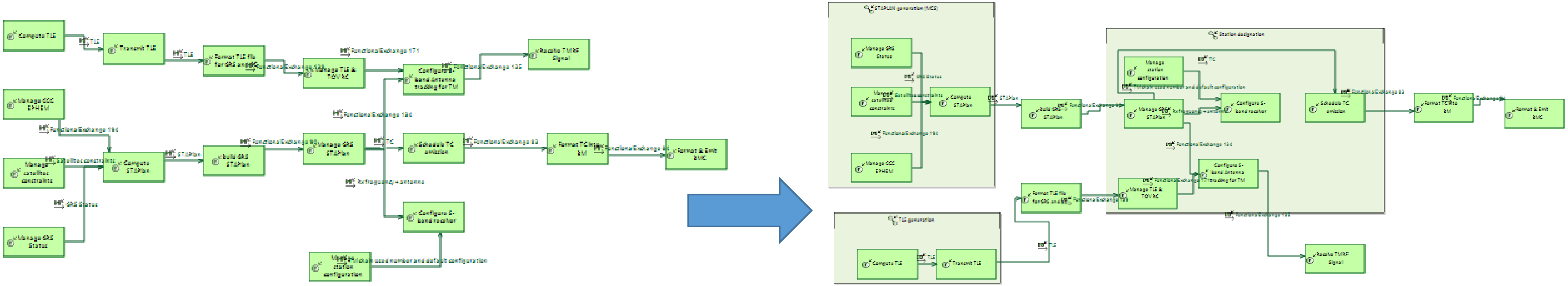


# Step 4: generation of “readable” test scenarios

- Cutomization of Capella behaviors in order to obtain usable scenarios
  - New “sub-lifeline” generation feature
    - Allows functions “superposition”
      Clear/determinist reading

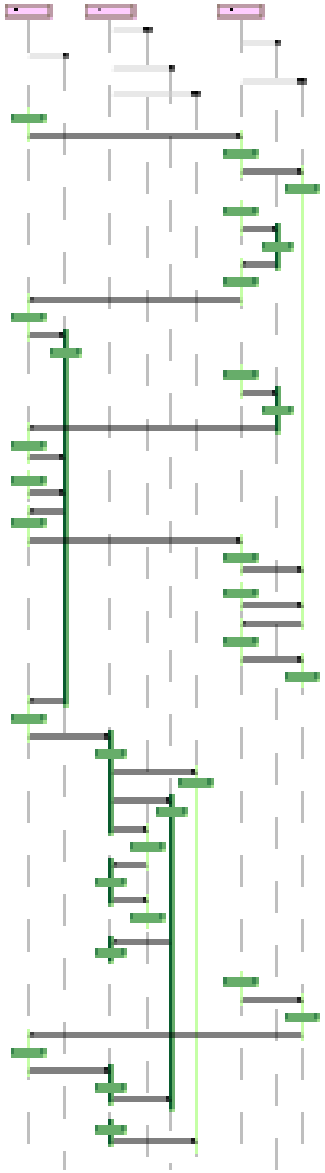


  - Injection of “sub-functional chain” support to compact scenarios

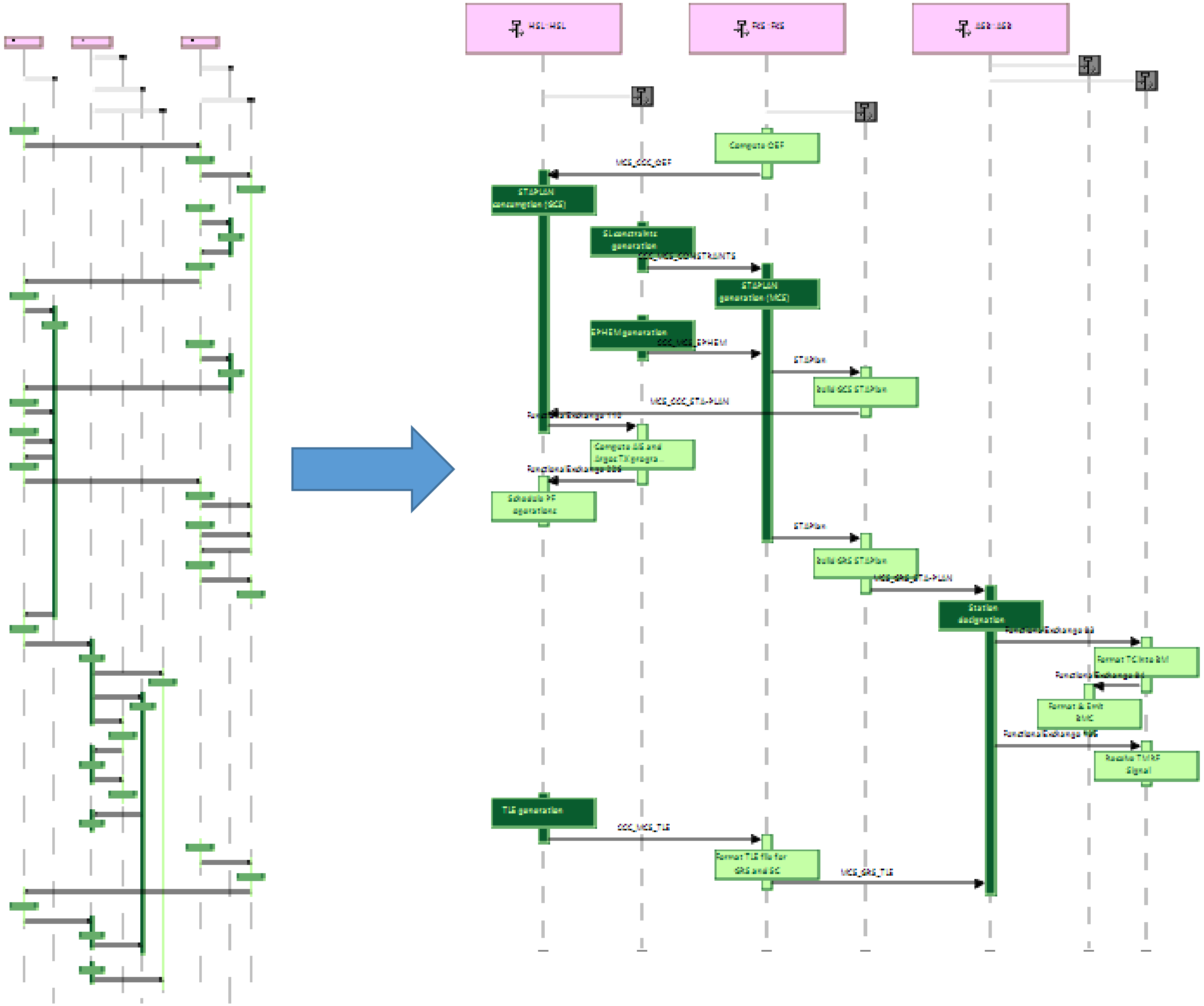


  - Injection of “super-compression” feature for extreme cases

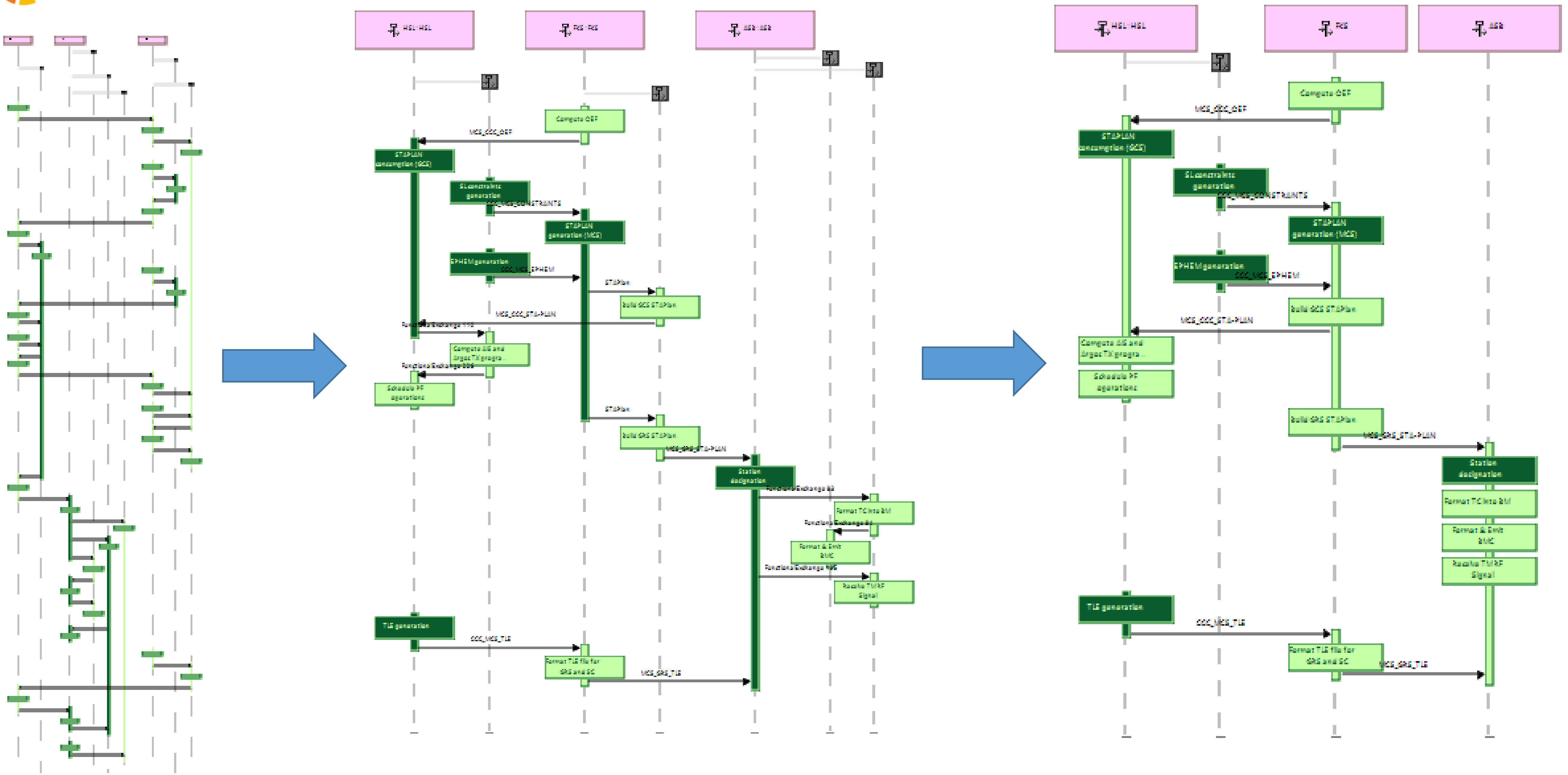
# Step 4: generation of “readable” test scenarios



# Step 4: generation of “readable” test scenarios



# Step 4: generation of “readable” test scenarios



# The system test process

- Application of the “system test” process defined during the CNES/Artal collaboration
- Main steps:
  - Capture the system architecture ✓
  - Capture validation needs ✓
    - Identification of the functional chains to validate
  - Specification of corresponding tests (using exchange scenarios)
    - Scenarios can be generated by a dedicated “home made” tool

# The system test process

- Application of the “system test” process defined during the CNES/Artal collaboration
- Main steps:
  - Capture the system architecture ✓
  - Capture validation needs ✓
    - Identification of the functional chains to validate
  - Specification of corresponding tests (using exchange scenarios) ✓
    - Scenarios can be generated by a dedicated “home made” tool

Incl. ADVANCED features

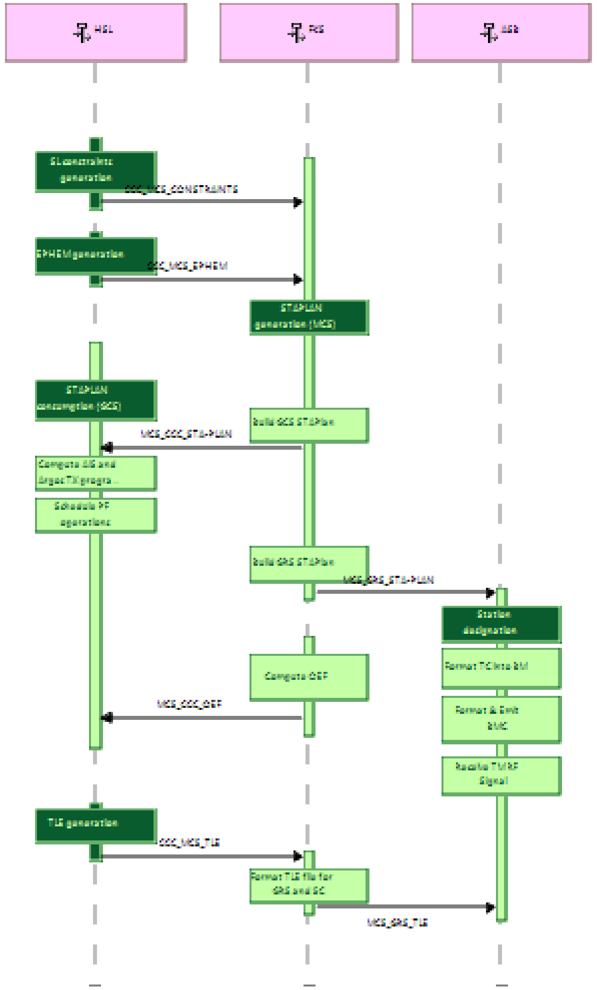
# The system test process

- Application of the “system test” process defined during the CNES/Artal collaboration
- Main steps:
  - Capture the system architecture ✓
  - Capture validation needs ✓
    - Identification of the functional chains to validate
  - Specification of corresponding tests (using exchange scenarios) ✓
    - Scenarios can be generated by a dedicated “home made” tool
  - Identification of user interactions and success criteria

Incl. ADVANCED features

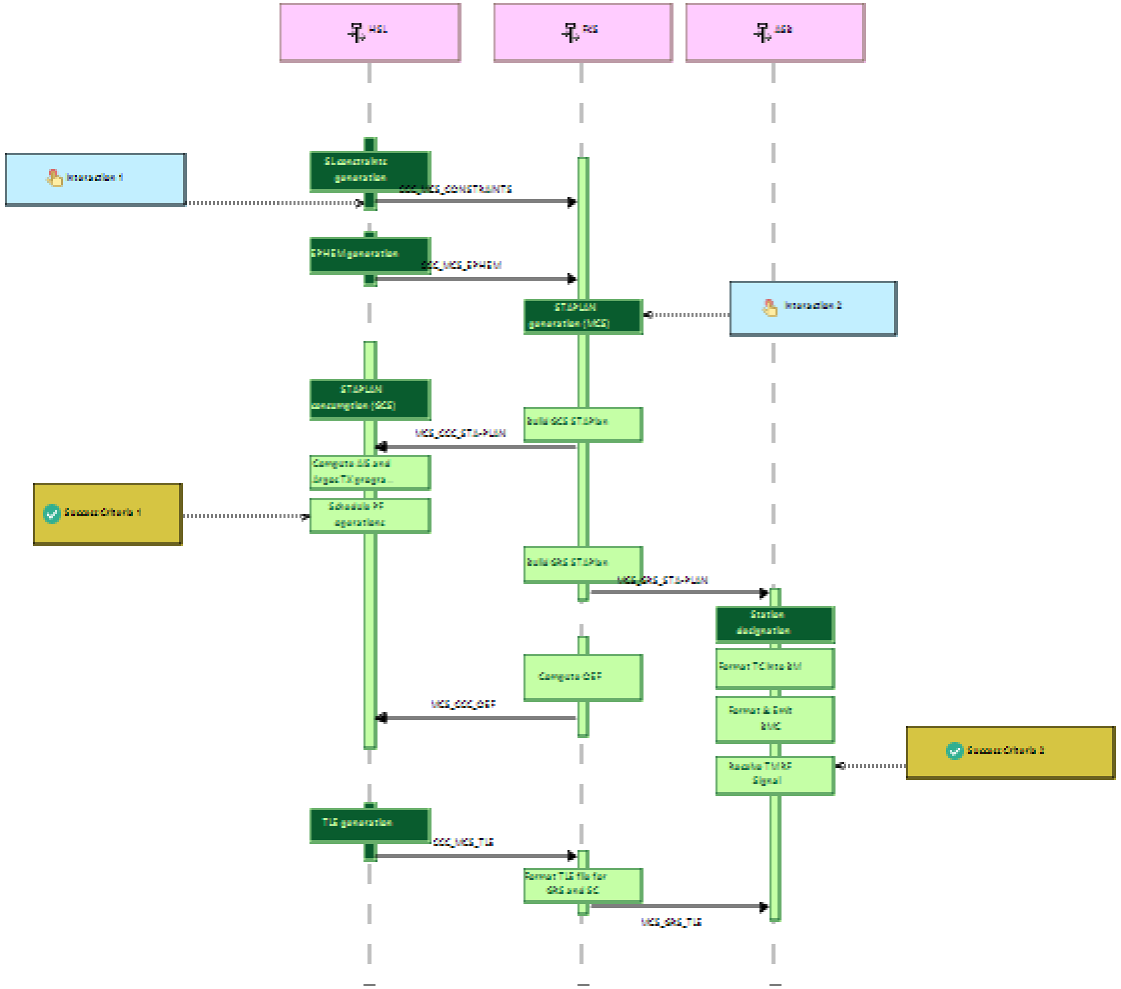
# Step 5: user interaction and success criteria

- Inject test scenarios execution instructions



# Step 5: user interaction and success criteria

- Inject test scenarios execution instructions

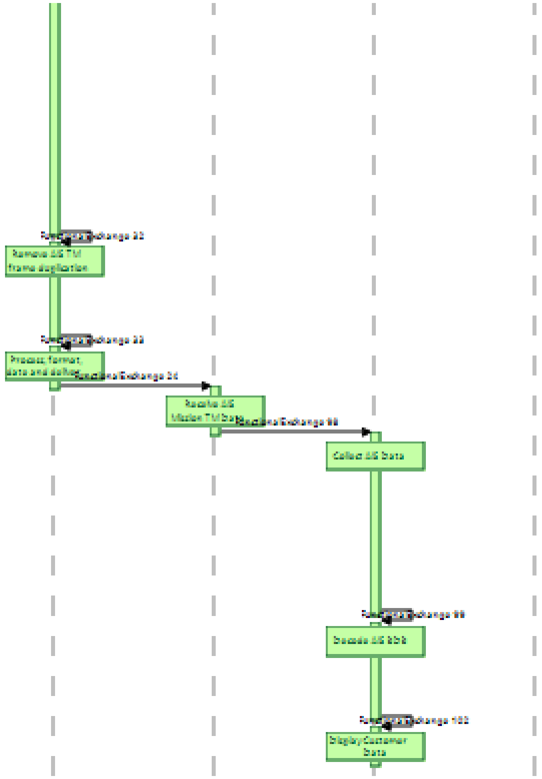


# Bonus step: make scenarios more editable

- The edition of complex Capella scenarios can become laborious
  - Several “graphical dependencies”
  - Lot of impacts of moving an item

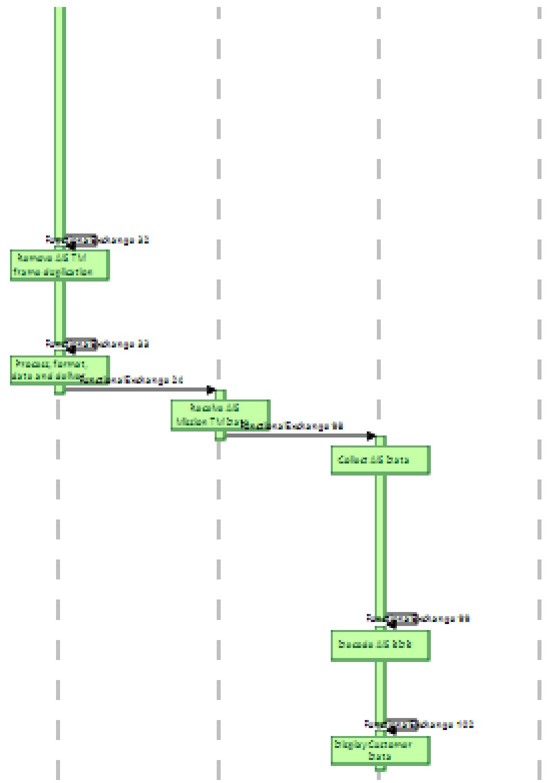
# Bonus step: make scenarios more editable

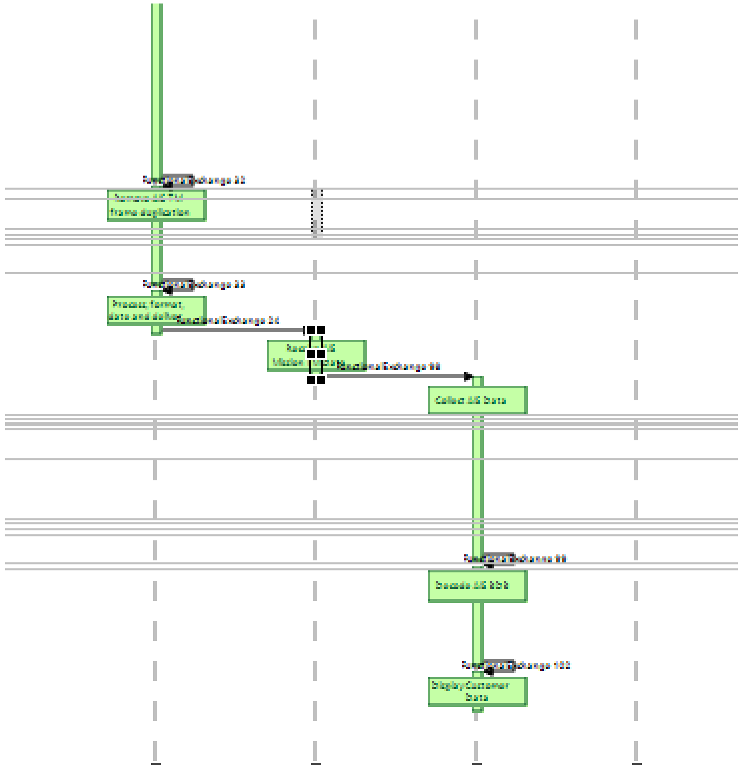
- The edition of complex Capella scenarios can become laborious
  - Several “graphical dependencies”
  - Lot of impacts of moving an item


Remove AIS TM frame duplication
Process, format, data and deliver
Receive AIS Mission TM Data
Collect AIS Data
Display Customer Data

# Bonus step: make scenarios more editable

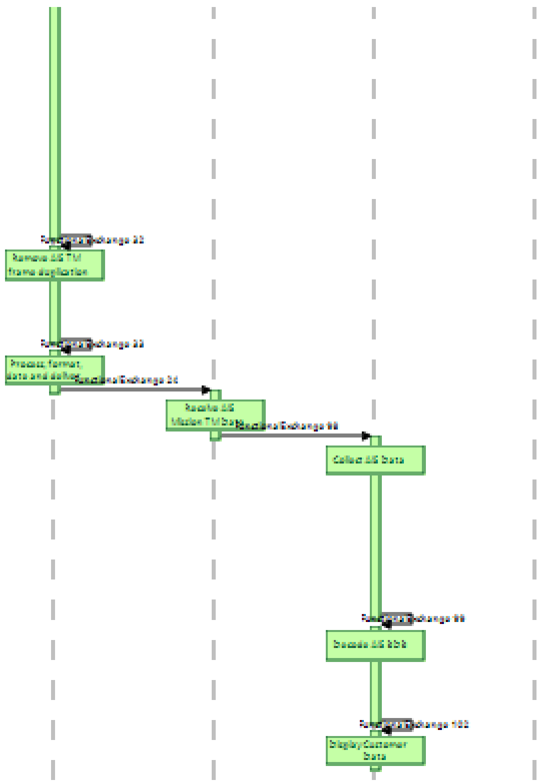
- The edition of complex Capella scenarios can become laborious
  - Several “graphical dependencies”
  - Lot of impacts of moving an item

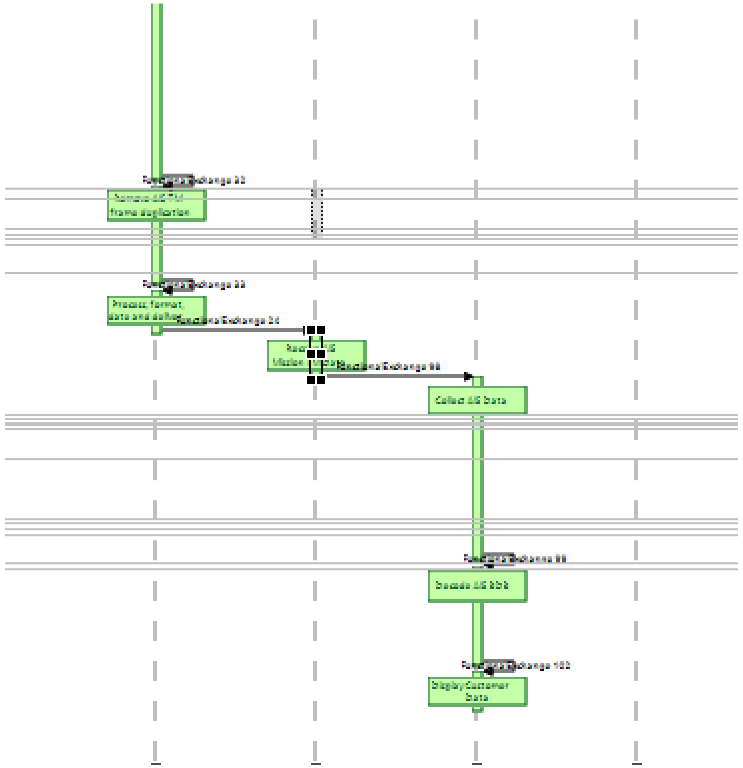


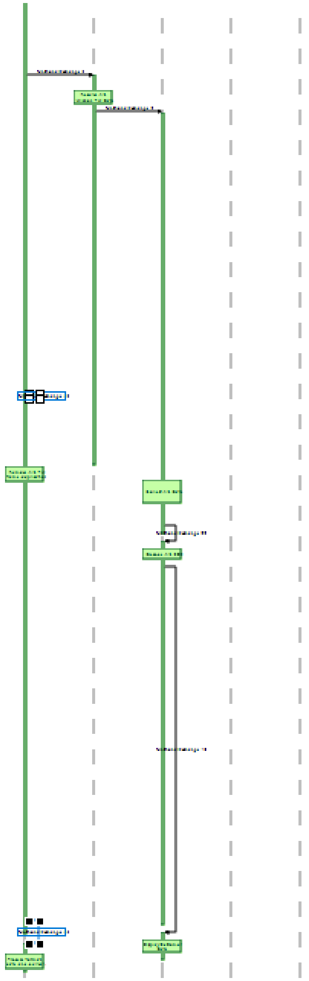


# Bonus step: make scenarios more editable

- The edition of complex Capella scenarios can become laborious
  - Several “graphical dependencies”
  - Lot of impacts of moving an item


Remove AIS TM frame duplication
Process, format, data and deliver
Receive AIS Mission TM Data
Collect AIS Data
Display Customer Data


Remove AIS TM frame duplication
Process, format, data and deliver
Receive AIS Mission TM Data
Collect AIS Data
Display Customer Data



# Bonus step: make scenarios more editable

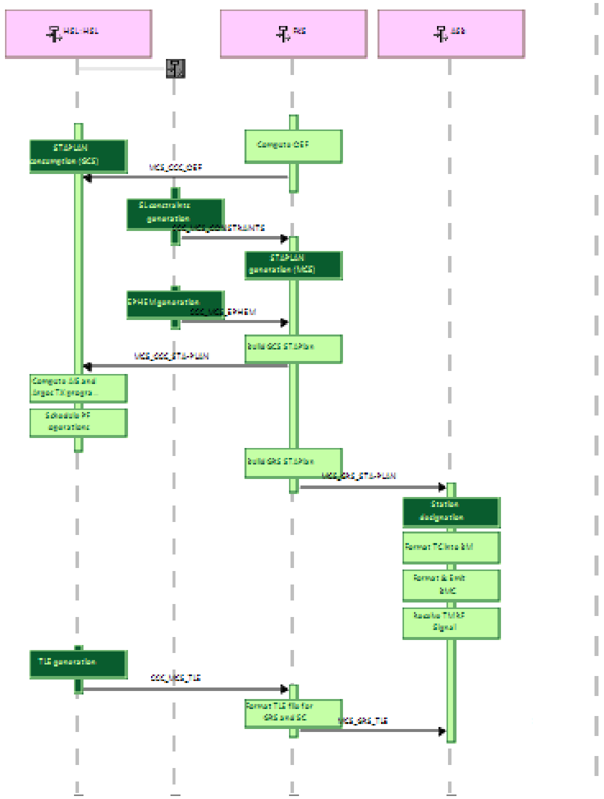
- The edition of complex Capella scenarios can become laborious
  - Several “graphical dependencies”
  - Lot of impacts of moving an item

⇒ Deletion of graphical constraints in generated scenarios

# Bonus step: make scenarios more editable

- The edition of complex Capella scenarios can become laborious
  - Several “graphical dependencies”
  - Lot of impacts of moving an item

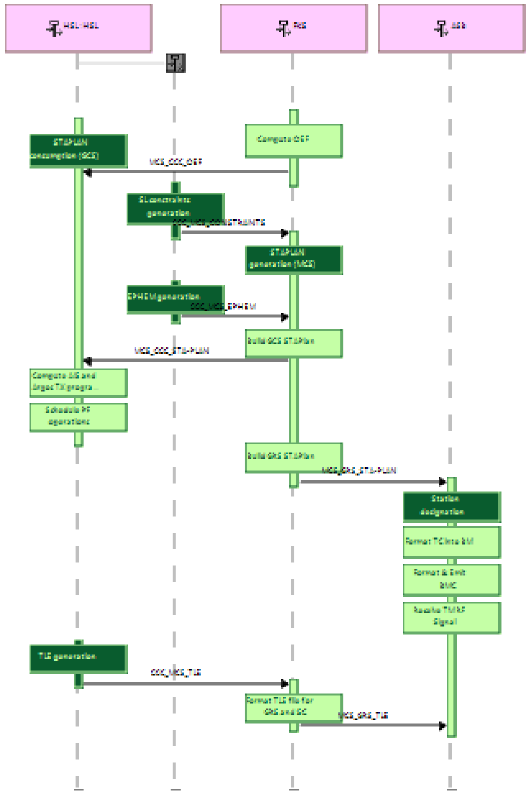
⇒ Deletion of graphical constraints in generated scenarios

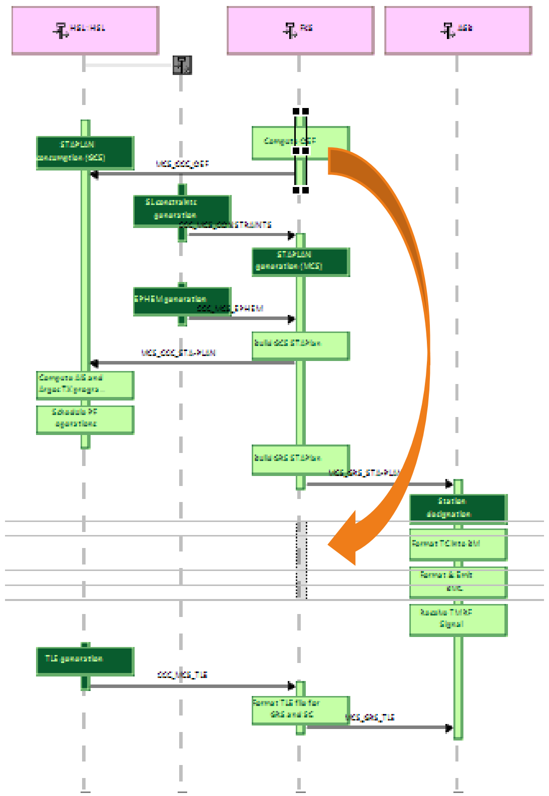


# Bonus step: make scenarios more editable

- The edition of complex Capella scenarios can become laborious
  - Several “graphical dependencies”
  - Lot of impacts of moving an item

⇒ Deletion of graphical constraints in generated scenarios

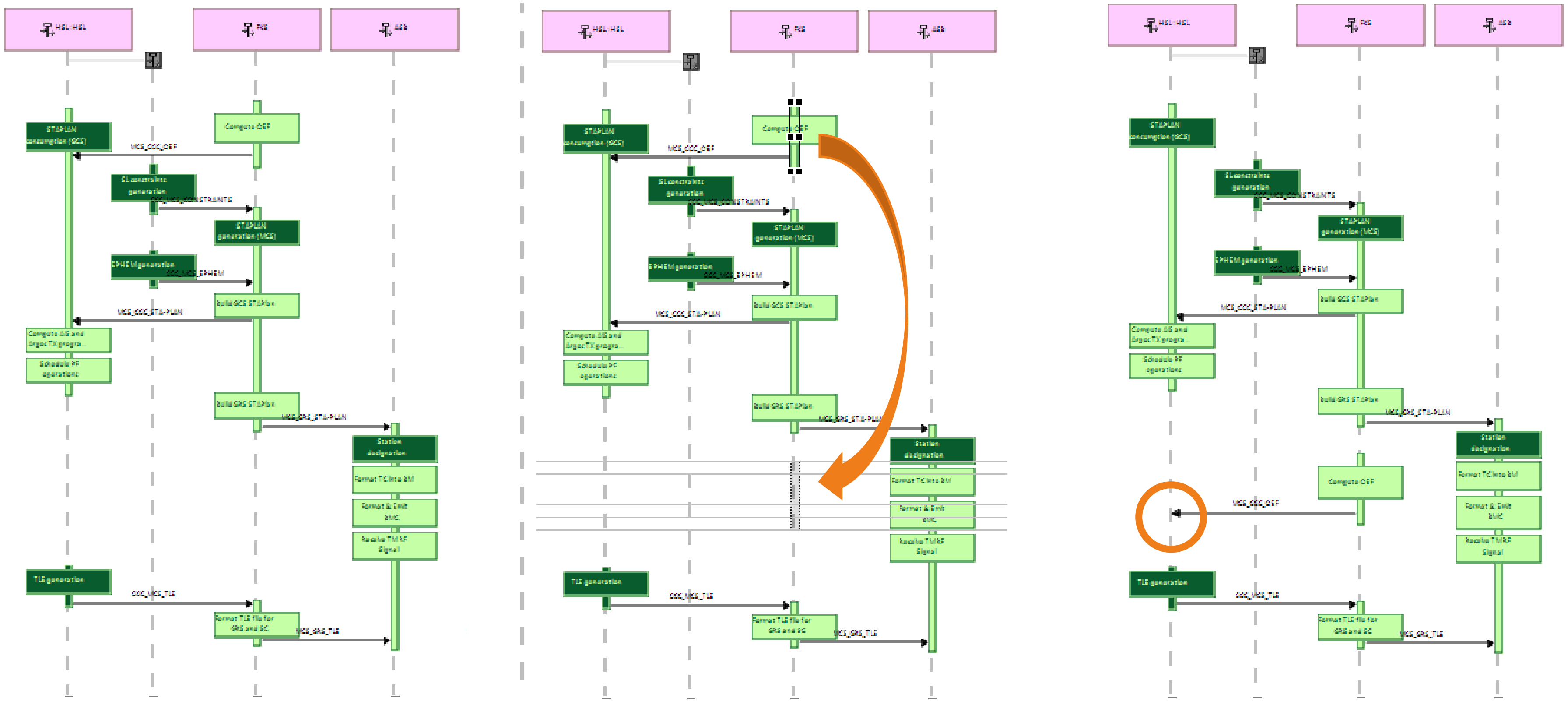




# Bonus step: make scenarios more editable

- The edition of complex Capella scenarios can become laborious
  - Several “graphical dependencies”
  - Lot of impacts of moving an item

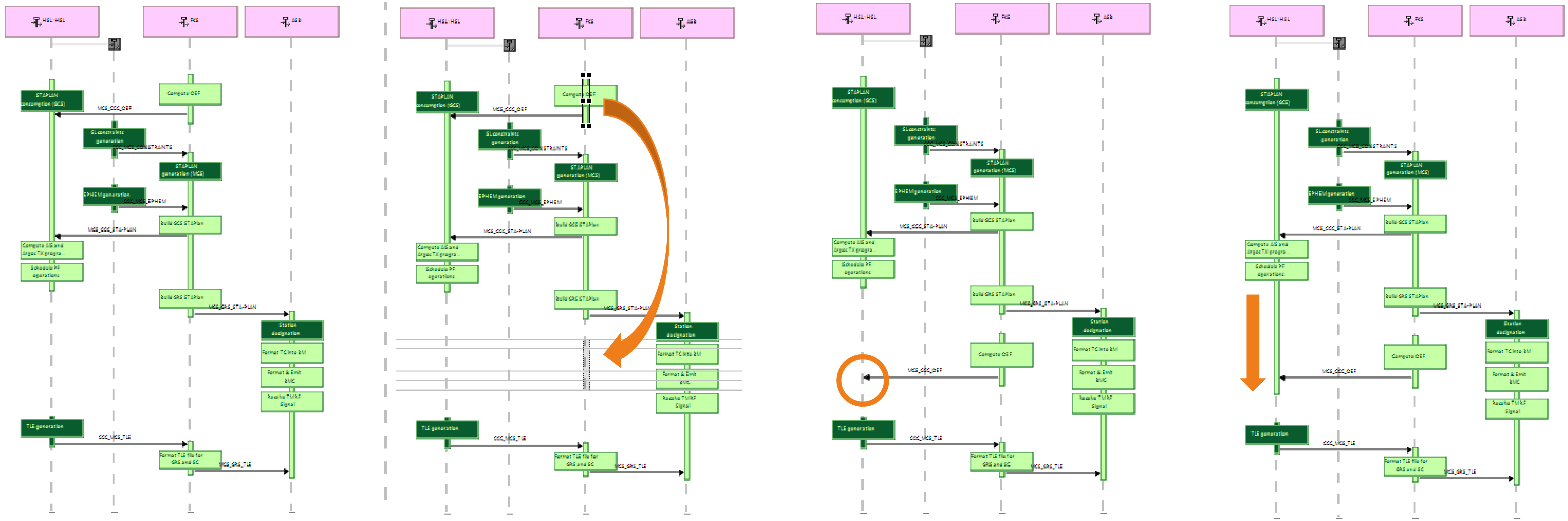
⇒ Deletion of graphical constraints in generated scenarios



# Bonus step: make scenarios more editable

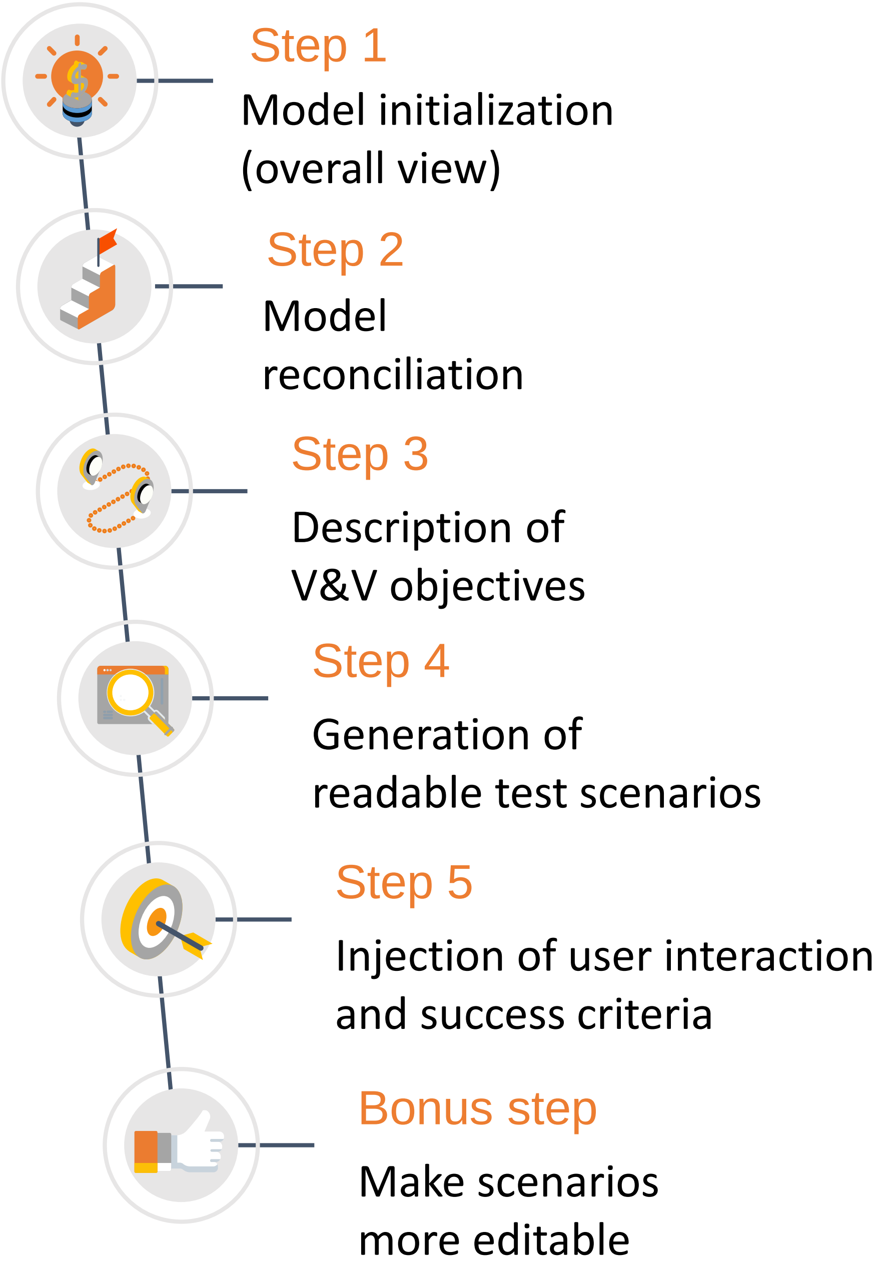
- The edition of complex Capella scenarios can become laborious
  - Several “graphical dependencies”
  - Lot of impacts of moving an item

⇒ Deletion of graphical constraints in generated scenarios



# Conclusion

- Review:
  - 1 year Kinéis / Artal collaboration
  - Remaining 9 months before launch
  - Kinéis have gained a new skill: MBSE/Capella
  - System tests capture in progress
    - Based on Capella models (growing)
      - 33 components / 112 component ports
      - 200 functions / 500 function ports
      - 33 functional chains
    - Using a dedicated Capella extension



Step 1
Model initialization (overall view)

Step 2
Model reconciliation

Step 3
Description of V&V objectives

Step 4
Generation of readable test scenarios

Step 5
Injection of user interaction and success criteria

Bonus step
Make scenarios more editable

# Conclusion: Kinéis feedback

- ⊕ Improvement of communication between teams
- ⊕ Optimization of component development plans according to the functional needs of the system
- ⊕ Automatic update of V&V objectives and scenarios (in case of system architecture modification)
- ⊕ Better visibility for system test definition
- ⊕ Automatic initialization of test cases scenario

- ⊖ Capella not easy to pick-up, assistance (and time) required
- ⊖ Difficult responsiveness in real time (during meetings)
- ⊖ Model reconciliations are laborious

  => Essential to respect scheduling: main view prior to sub-components in this case

More information on

http://capella.artal-group.com