Collaborative model-based engineering and large systems development

Catherine Morlet, Alberto Gonzalez Fernandez

ESA ESTEC

14/11/2023

Outline of the presentation

❑ Introduction: Galileo system and its complexity

❑ Collaborative end-to-end system design including security levels

❑ Collaboration among stakeholders

- ➢ sub-systems
- ➢ linking design and specifications
- ➢ reviewers/readers

❑ Concluding remarks

Outline of the presentation

- ❑ **Introduction: Galileo and its complexity**
- ❑ Collaborative end-to-end system design including security levels
- ❑ Collaboration among stakeholders
  - ➢ sub-systems
  - ➢ linking design and specifications
  - ➢ reviewers/readers
- ❑ Concluding remarks

Galileo – program level view

Galileo program

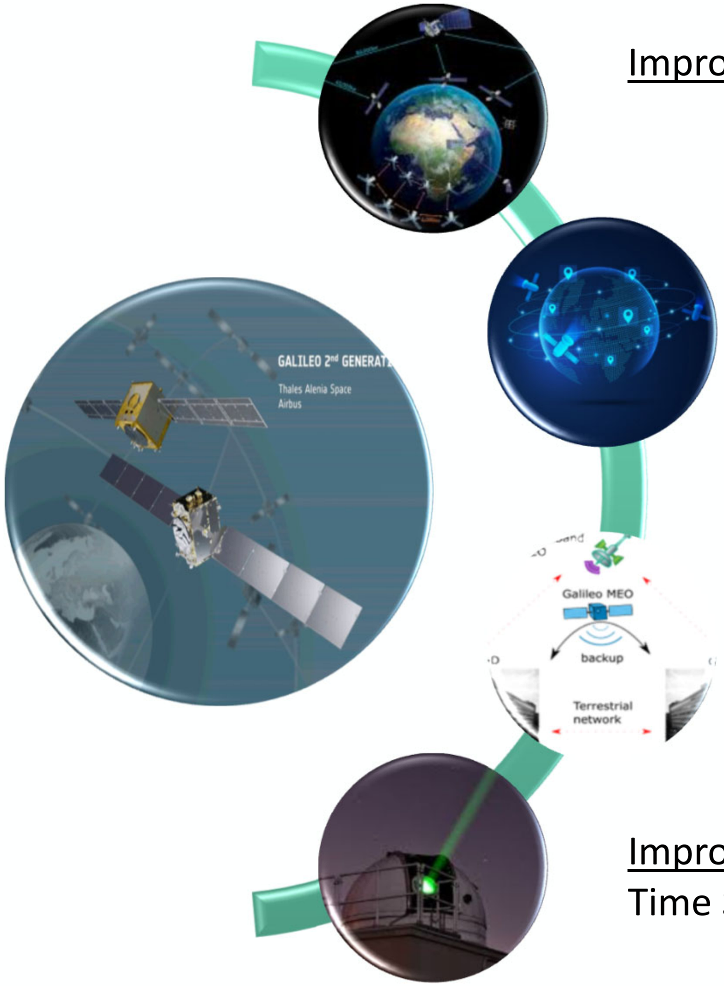
- Europe's initiative started in the 1990's for a state-of-the-art global satellite navigation system
- First satellite launched in 2005 and operational since end 2016
- Core system composed of currently 28 satellites in orbit (24 providing service worldwide), 2 ground centres, 15 remote sites worldwide complemented by a set of service facilities
- More than 4bn users around the world (meaning navigation receivers with Galileo embedded)
- … and best-in-class navigation system today

Galileo is permanently evolving with system-level enhancements along the deployment of the system to improve performance and ensure the highest user adoption.

The second generation of Galileo (G2G) is the instantiation of the mission objectives once the full second generation of the constellation (and associated ground system) is deployed and in operation [timeframe ~10+ years from now]

Galileo 2nd Generation – program level view

G2 Full Operational Capability
(Space and Ground)



Improve connectivity and observability

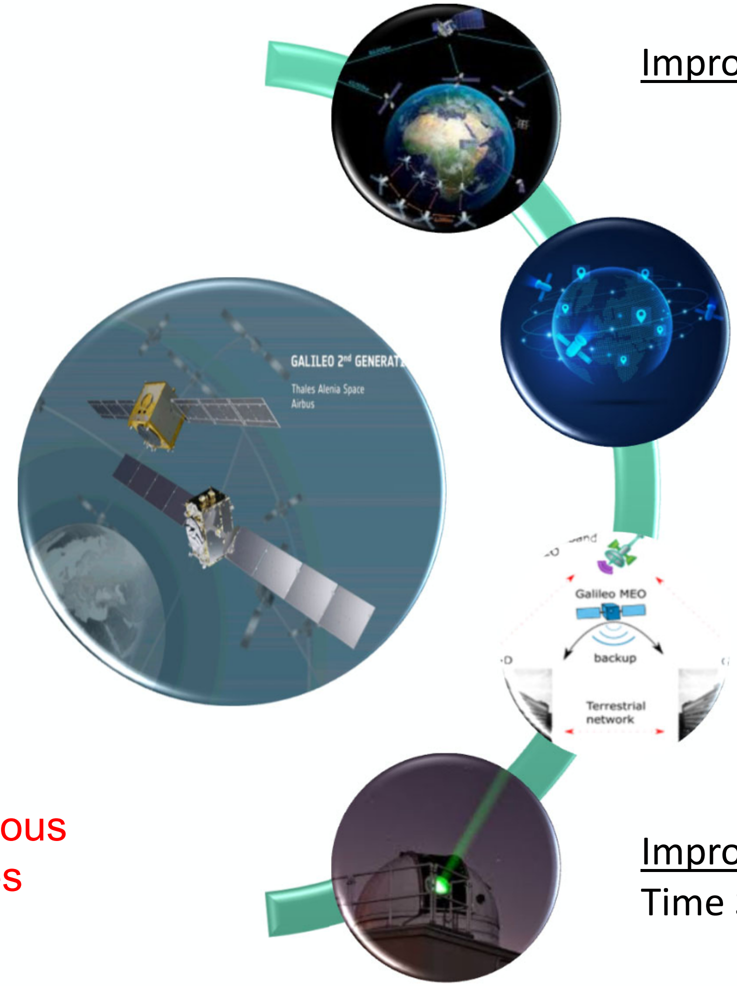
Increase constellation for performance and service resilience

Include more services e.g. two-way services

Improve Orbit Determination and Time Synchronisation

Galileo 2nd Generation – program level view

G2 Full Operational Capability (Space and Ground)



<u>Improve</u> connectivity and observability

<u>Increase</u> constellation for performance and service resilience

<u>Include</u> more services e.g. two-way services

<u>Improve</u> Orbit Determination and Time Synchronisation

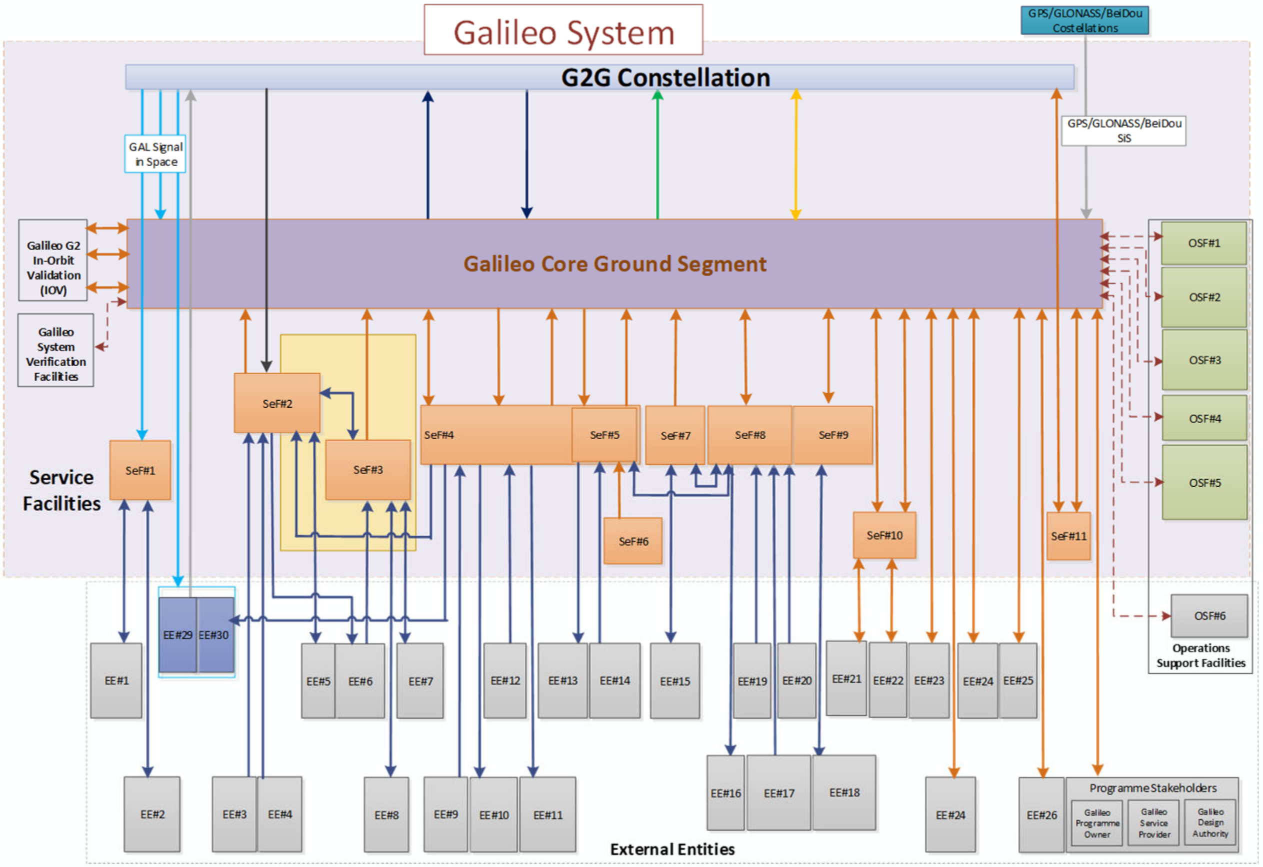
Not a new system but a continuous evolution with additional features and better performance !

Galileo – design complexity view

In terms of architecture of the system:

- A satellite constellation
- Distributed core ground segment with sites worldwide to collect data that contribute to the generation of navigation products (closed loop system), to perform the monitoring and control (including security aspects)
- Several services facilities on ground (in Europe)
- And Interfaces with many external entities (worldwide)

Note: for each segment and facility, a set of one or more security levels is foreseen



Outline of the presentation

- ❑ Introduction: Galileo and its complexity
- ❑ **Collaborative end-to-end system design including security levels**
- ❑ Collaboration among stakeholders
  - ➢ sub-systems
  - ➢ linking design and specifications
  - ➢ reviewers/readers
- ❑ Concluding remarks

Collaborative end-to-end system design including security levels – a bit of history

How we started with MBSE

The system of interest is complex: many sub-systems and interfaces to maintain coherently through any evolution.

Collaborative end-to-end system design including security levels – a bit of history

How we started with MBSE

The system of interest is complex: many sub-systems and interfaces to maintain coherently through any evolution.

Choice of a tool: In 2018 we chose Capella with Team4Capella as the most promising and quicker to use for non-MBSE experts (T4C installed on VM with dedicated server remotely accessible) – co-design of a single model by engineers from different companies and not all collocated

Collaborative end-to-end system design including security levels – a bit of history

How we started with MBSE

The system of interest is complex: many sub-systems and interfaces to maintain coherently through any evolution.

Choice of a tool: In 2018 we chose Capella with Team4Capella as the most promising and quicker to use for non-MBSE experts (T4C installed on VM with dedicated server remotely accessible) – co-design of a single model by engineers from different companies and not all collocated

Training of the team and set-up of first projects: set of engineers to become the developers trained as a group and who developed first attempts of the system model together sharing their experience and their methodology in weekly progress meetings

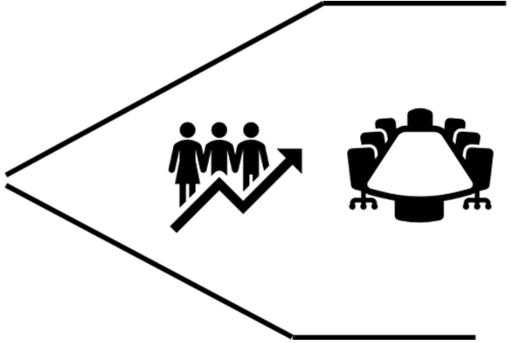
Collaborative end-to-end system design including security levels – a bit of history

How we started with MBSE

The system of interest is complex: many sub-systems and interfaces to maintain coherently through any evolution.

Choice of a tool: In 2018 we chose Capella with Team4Capella as the most promising and quicker to use for non-MBSE experts (T4C installed on VM with dedicated server remotely accessible) – co-design of a single model by engineers from different companies and not all collocated

Training of the team and set-up of first projects: set of engineers to become the developers trained as a group and who developed first attempts of the system model together sharing their experience and their methodology in weekly progress meetings



Collaborative end-to-end system design including security levels – a bit of history

Verification by an MBSE expert of our initial project (in Capella at system analysis level before transition to the logical analysis): consistency of the approach was checked, readiness to go to the next level confirmed but we were going sometimes already too deep in our system analysis.

Collaborative end-to-end system design including security levels – a bit of history

Verification by an MBSE expert of our initial project (in Capella at system analysis level before transition to the logical analysis): consistency of the approach was checked, readiness to go to the next level confirmed but we were going sometimes already too deep in our system analysis.

Production of a first (unclassified) model for our SRR and some additional separated models for the classified aspects

Collaborative end-to-end system design including security levels – a bit of history

Verification by an MBSE expert of our initial project (in Capella at system analysis level before transition to the logical analysis): consistency of the approach was checked, readiness to go to the next level confirmed but we were going sometimes already too deep in our system analysis.

Production of a first (unclassified) model for our SRR and some additional separated models for the classified aspects

Decision to rebuild for next phase:

- To simplify the system analysis and move the details to the logical analysis
- To extend our system perimeter and define new interfaces based on SRR decisions
- To construct a fully coordinated and synchronised view of the end-to-end system with several security classification branches (security branches are developed on dedicated IT infrastructure)

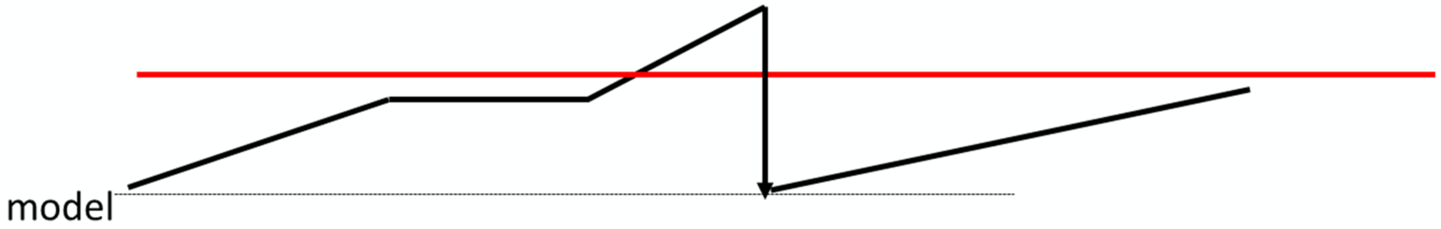
Collaborative end-to-end system design including security levels – a bit of history

Verification by an MBSE expert of our initial project (in Capella at system analysis level before transition to the logical analysis): consistency of the approach was checked, readiness to go to the next level confirmed but we were going sometimes already too deep in our system analysis.

Production of a first (unclassified) model for our SRR and some additional separated models for the classified aspects

Decision to rebuild for next phase:

- To simplify the system analysis and move the details to the logical analysis
- To extend our system perimeter and define new interfaces based on SRR decisions
- To construct a fully coordinated and synchronised view of the end-to-end system with several security classification branches (security branches are developed on dedicated IT infrastructure)



Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Galileo is a project where different security classification levels and need-to-know co-exist. Need to have a methodology that allows for parallel modelling that are consistent and address the same one system of interest.

Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Galileo is a project where different security classification levels and need-to-know co-exist. Need to have a methodology that allows for parallel modelling that are consistent and address the same one system of interest.

Result today is a 5-branches model fully synchronised to the first branch that represents our unclassified model, the 4 other branches being the 4 classified modules.

Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Galileo is a project where different security classification levels and need-to-know co-exist. Need to have a methodology that allows for parallel modelling that are consistent and address the same one system of interest.

Result today is a 5-branches model fully synchronised to the first branch that represents our unclassified model, the 4 other branches being the 4 classified modules.

The security levels are such that the more restrictive the less project members have the right to access information

Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Galileo is a project where different security classification levels and need-to-know co-exist. Need to have a methodology that allows for parallel modelling that are consistent and address the same one system of interest.

Result today is a 5-branches model fully synchronised to the first branch that represents our unclassified model, the 4 other branches being the 4 classified modules.

The security levels are such that the more restrictive the less project members have the right to access information

A full project overview would consist in the aggregation of all security levels branches

Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Galileo is a project where different security classification levels and need-to-know co-exist. Need to have a methodology that allows for parallel modelling that are consistent and address the same one system of interest.

Result today is a 5-branches model fully synchronised to the first branch that represents our unclassified model, the 4 other branches being the 4 classified modules.

The security levels are such that the more restrictive the less project members have the right to access information

A full project overview would consist in the aggregation of all security levels branches

Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Reference branch: the unclassified part of the design (with all system functions included as well as all logical components);

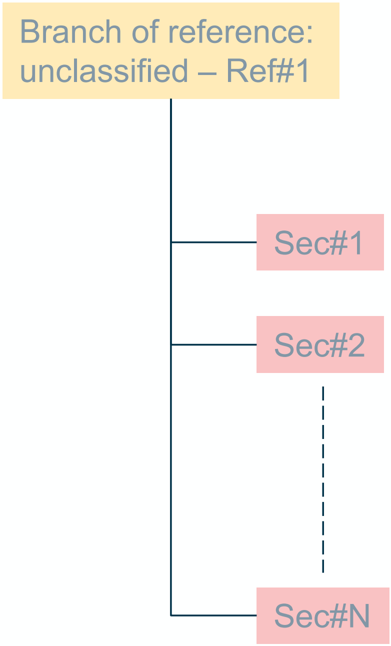
Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Reference branch: the unclassified part of the design (with all system functions included as well as all logical components);

Branch of reference: unclassified – Ref#1

Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Reference branch: the unclassified part of the design (with all system functions included as well as all logical components);



Branch of reference: unclassified – Ref#1

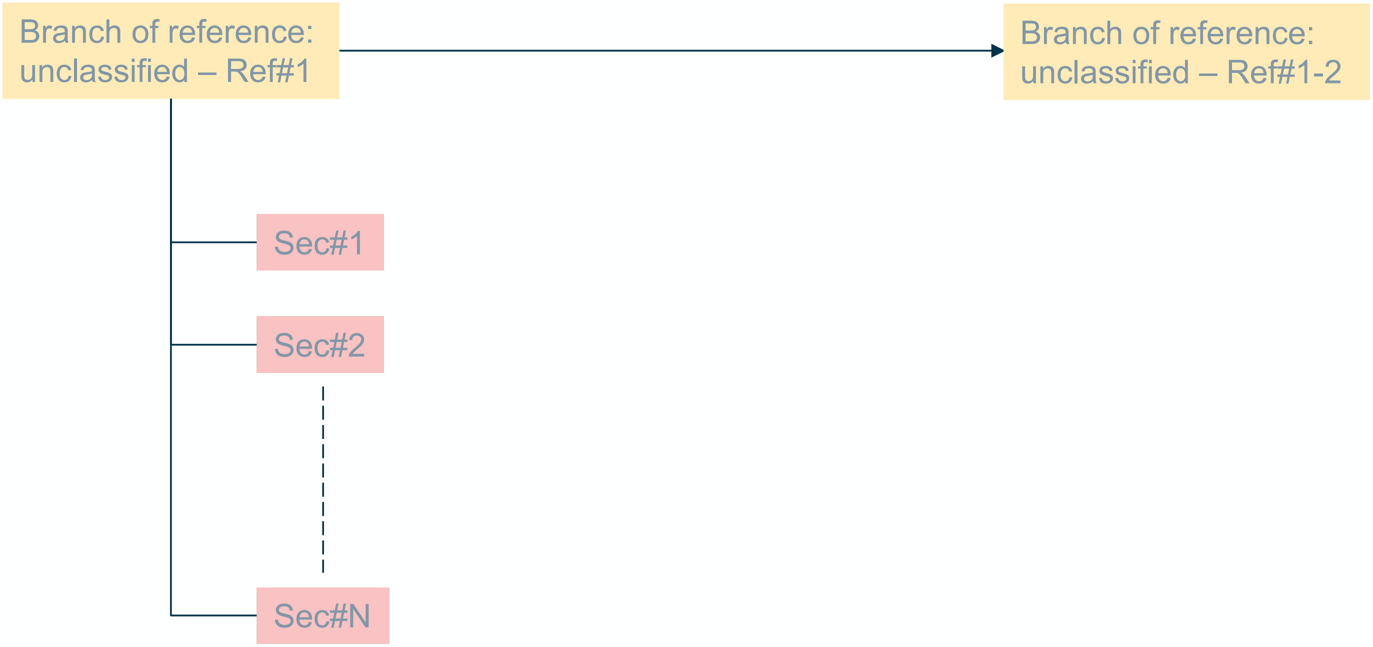
Sec#1

Sec#2

Sec#N

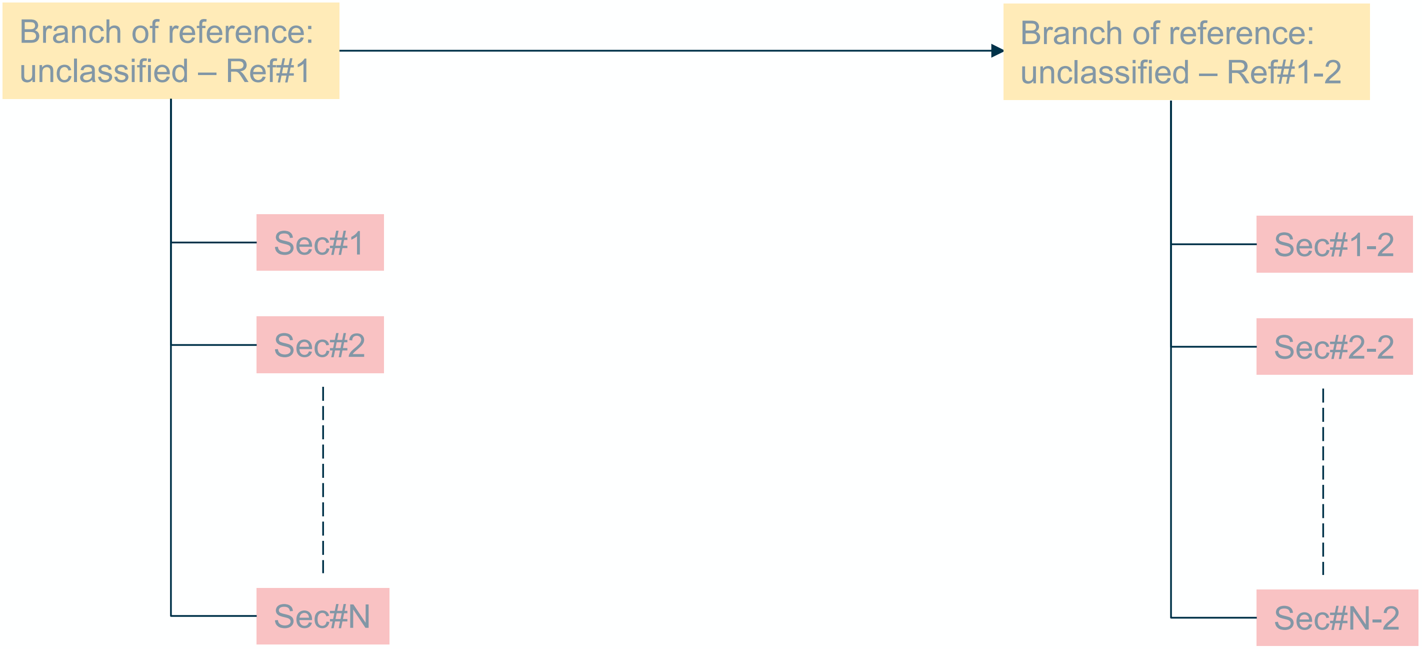
Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Reference branch: the unclassified part of the design (with all system functions included as well as all logical components);


Branch of reference: unclassified – Ref#1
Branch of reference: unclassified – Ref#1-2
Sec#1
Sec#2
Sec#N

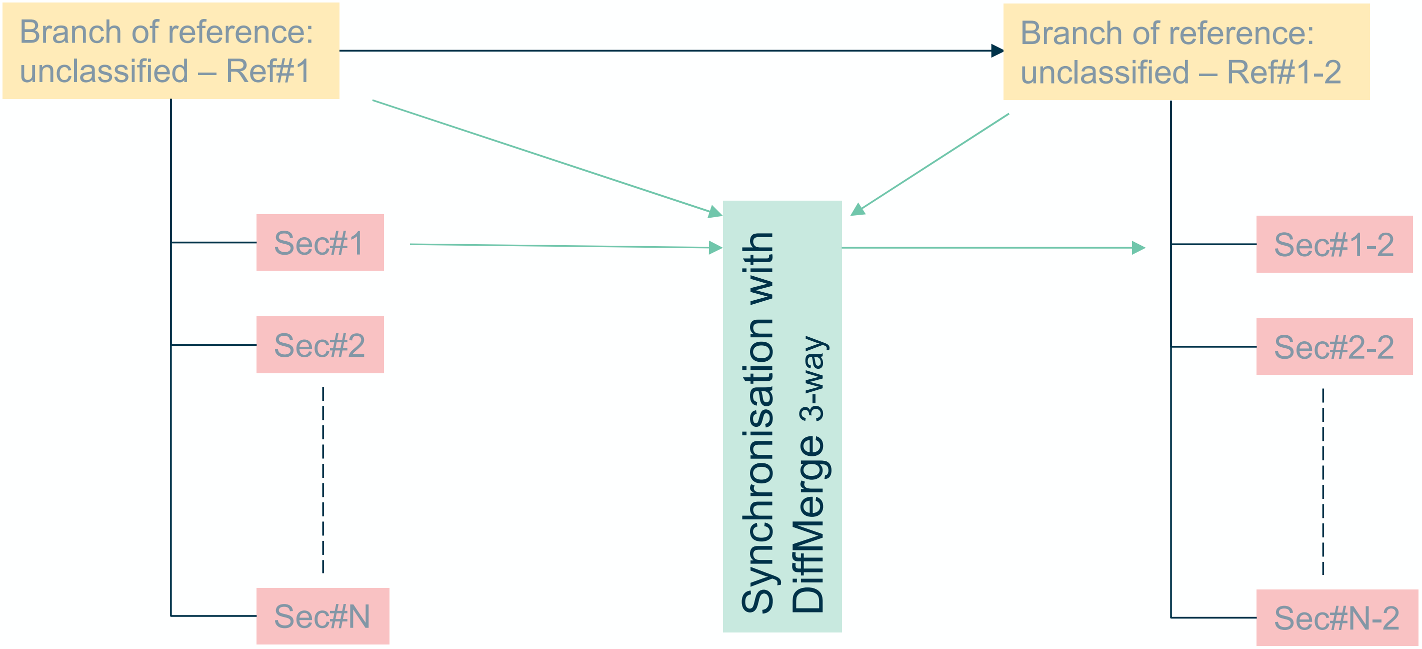
Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Reference branch: the unclassified part of the design (with all system functions included as well as all logical components);


Branch of reference: unclassified – Ref#1
Sec#1
Sec#2
Sec#N
Branch of reference: unclassified – Ref#1-2
Sec#1-2
Sec#2-2
Sec#N-2

Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Reference branch: the unclassified part of the design (with all system functions included as well as all logical components);


Branch of reference: unclassified – Ref#1

Branch of reference: unclassified – Ref#1-2

Sec#1

Sec#2

Sec#N

Synchronisation with DiffMerge 3-way

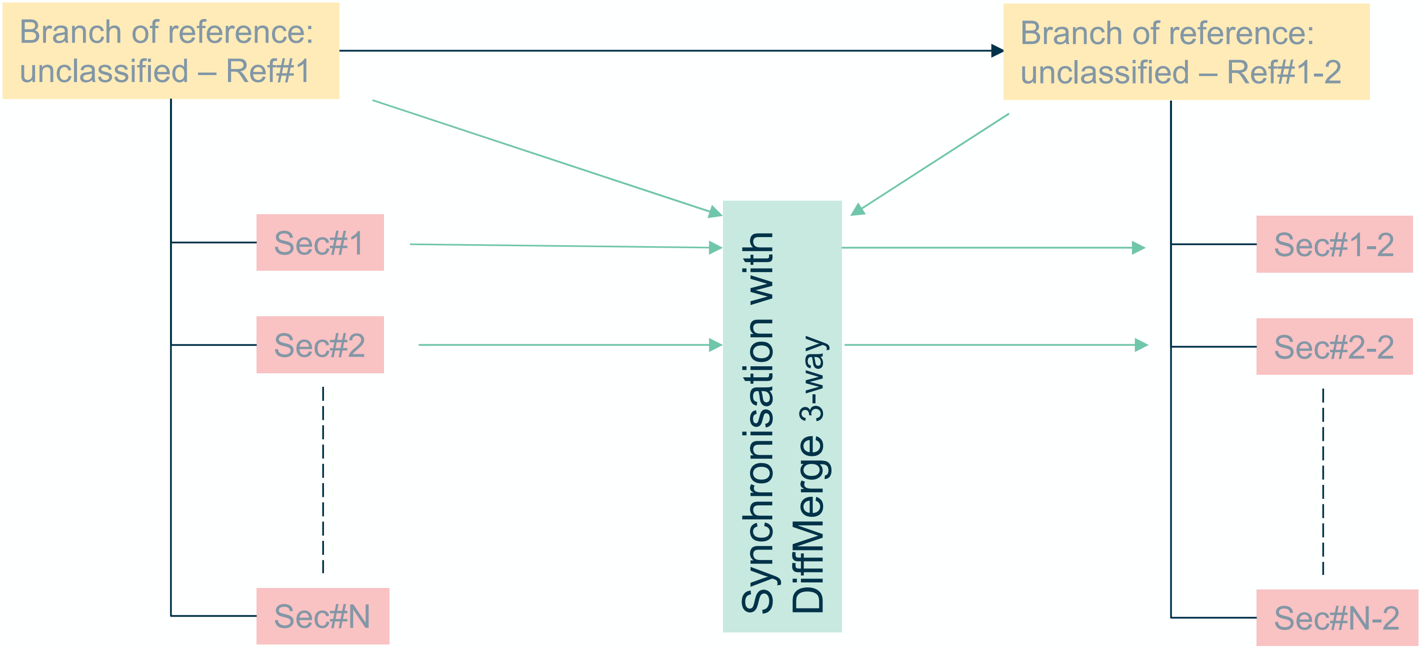
Sec#1-2

Sec#2-2

Sec#N-2

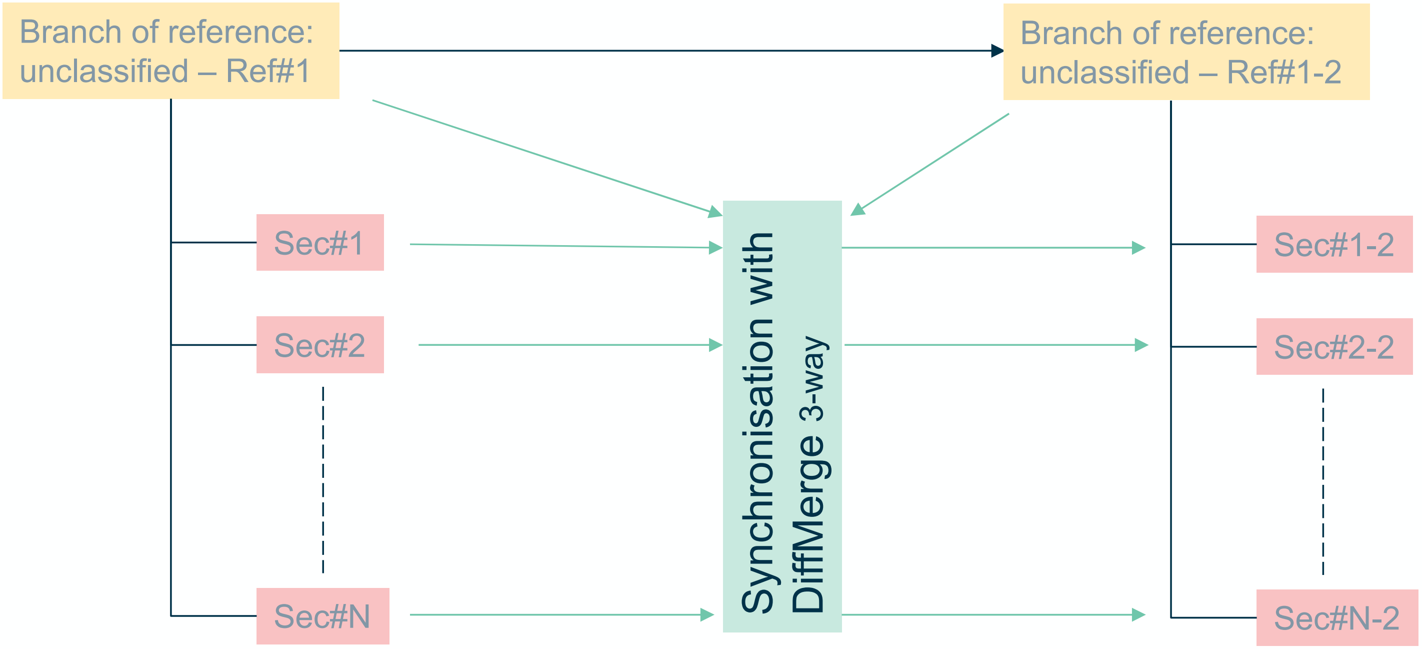
Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Reference branch: the unclassified part of the design (with all system functions included as well as all logical components);


Branch of reference: unclassified – Ref#1
Branch of reference: unclassified – Ref#1-2
Sec#1
Sec#2
Sec#N
Synchronisation with DiffMerge 3-way
Sec#1-2
Sec#2-2
Sec#N-2

Collaborative end-to-end system design including security levels – branching and DiffMerge usage

Reference branch: the unclassified part of the design (with all system functions included as well as all logical components);



Branch of reference: unclassified – Ref#1

Branch of reference: unclassified – Ref#1-2

Sec#1

Sec#2

Sec#N

Synchronisation with DiffMerge 3-way

Sec#1-2

Sec#2-2

Sec#N-2

Collaborative end-to-end system design including security levels – methodology needs

Along the way, it appeared that having a framework/methodology for our project was key to:

- Facilitate any synchronisation (with DiffMerge)
- Build a full end-to-end model (concatenation in a given order of all branches with DiffMerge)
- Document the model (some diagram view to be built by each designer, where to write descriptive text, etc.) for other engineers to find the information they need for their own part of the model development
- Use of a common colour coding (PVMT/DS usage)

Collaborative end-to-end system design including security levels – methodology needs

Along the way, it appeared that having a framework/methodology for our project was key to:

- Facilitate any synchronisation (with DiffMerge)
- Build a full end-to-end model (concatenation in a given order of all branches with DiffMerge)
- Document the model (some diagram view to be built by each designer, where to write descriptive text, etc.) for other engineers to find the information they need for their own part of the model development
- Use of a common colour coding (PVMT/DS usage)

Additionally, usage of M2Doc to provide export in form of structured documents (not yet everyone is familiar with MBSE and people are used to read documents!)

Benefits observed by the system engineers developing the model:

- Harmonisation of key elements developed in the model (e.g. figures LFBD, SDFB, LDFB, LAB, FS)
- Text description of what we are representing in the different figures
- Detailed description of each exchange between functions (see example table)
- Ease the co-development thanks to the descriptive text
- Partial exports

Benefits for the readers: systematic structure and set of information

Outline of the presentation

❑ Introduction: Galileo and its complexity

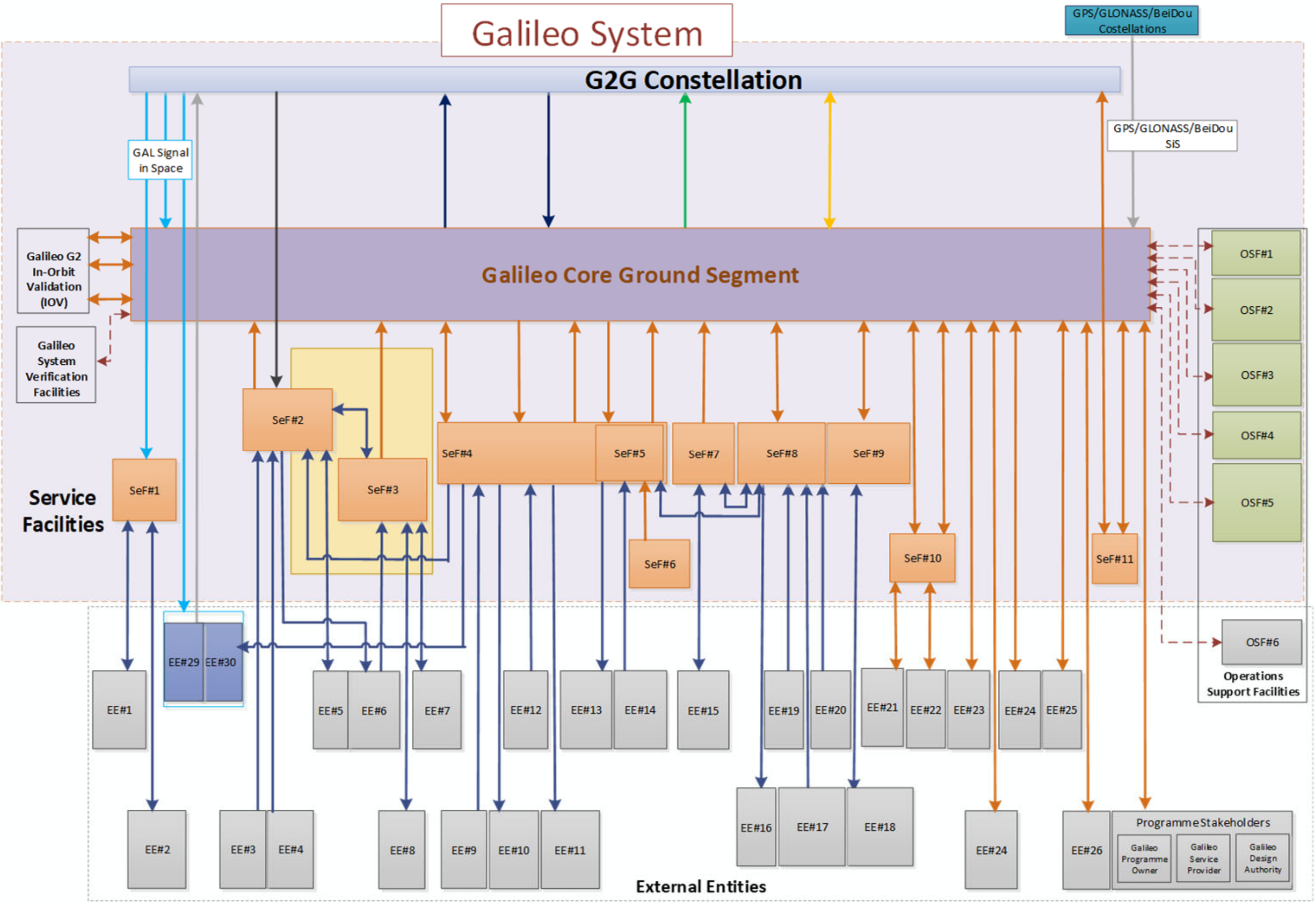
❑ Collaborative end-to-end system design including security levels

❑ **Collaboration among stakeholders**

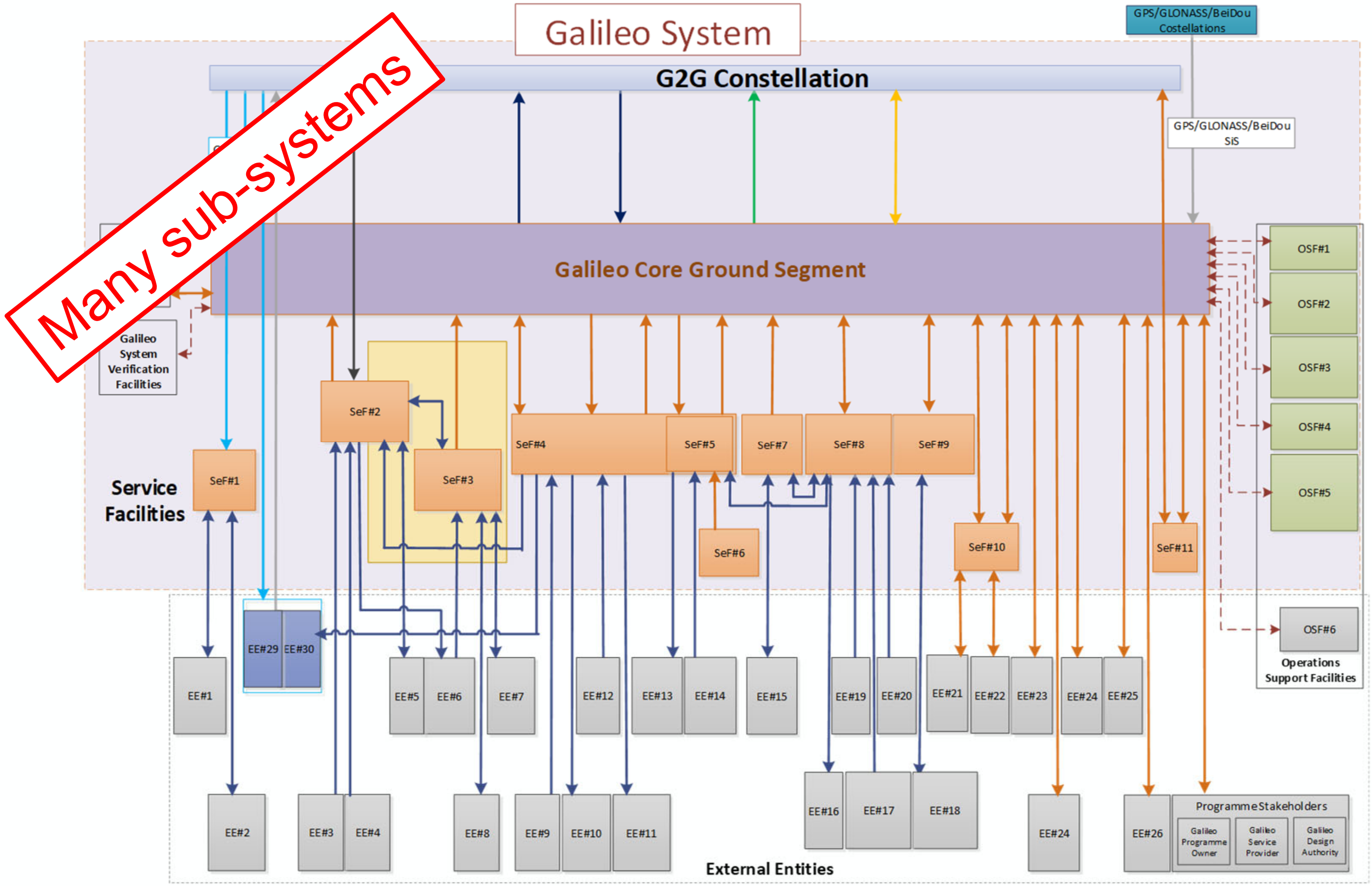
- **sub-systems**
- **linking design and specifications**
- **reviewers/readers**

❑ Concluding remarks

Collaboration among stakeholders

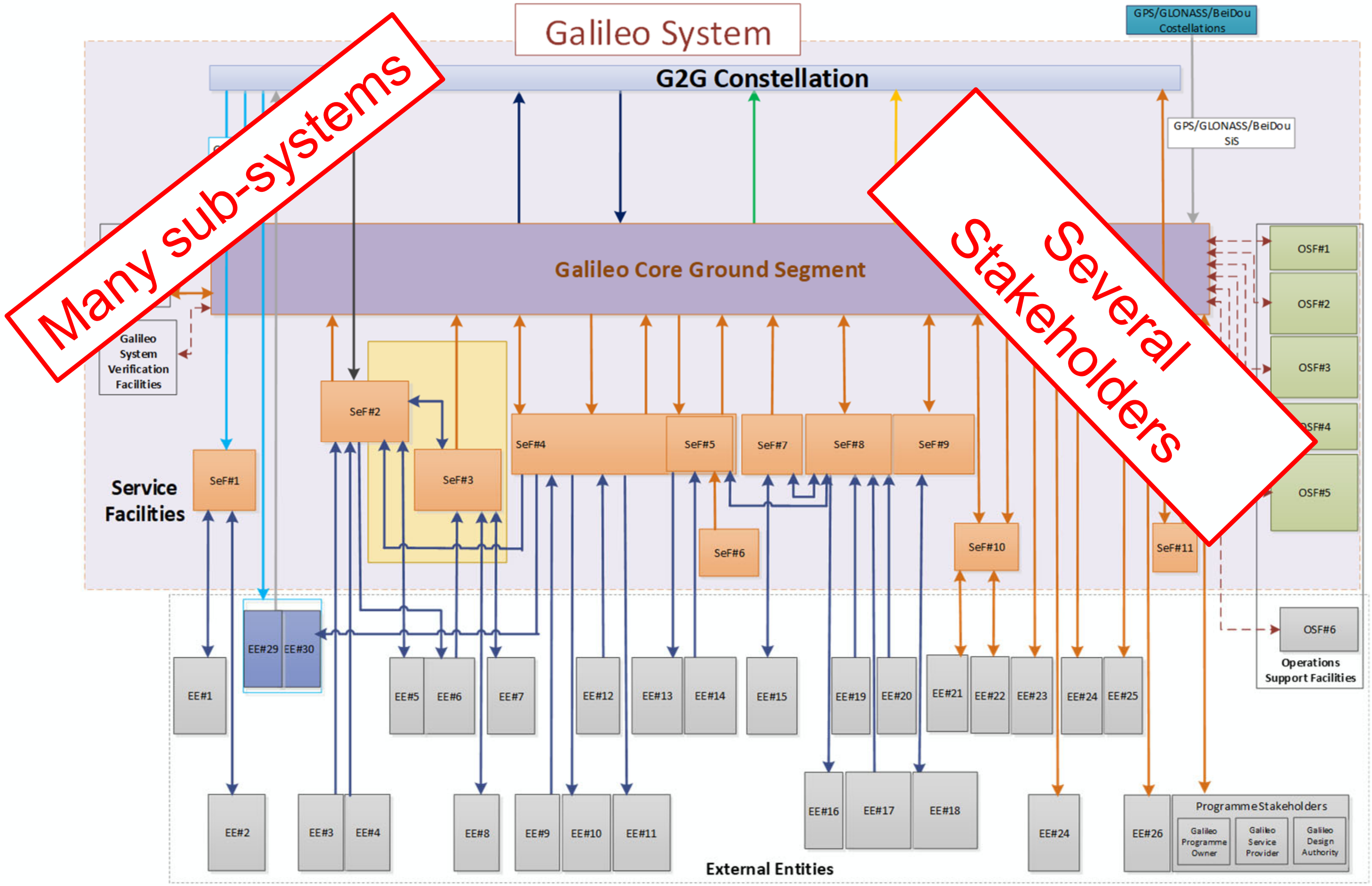


Collaboration among stakeholders



Many sub-systems

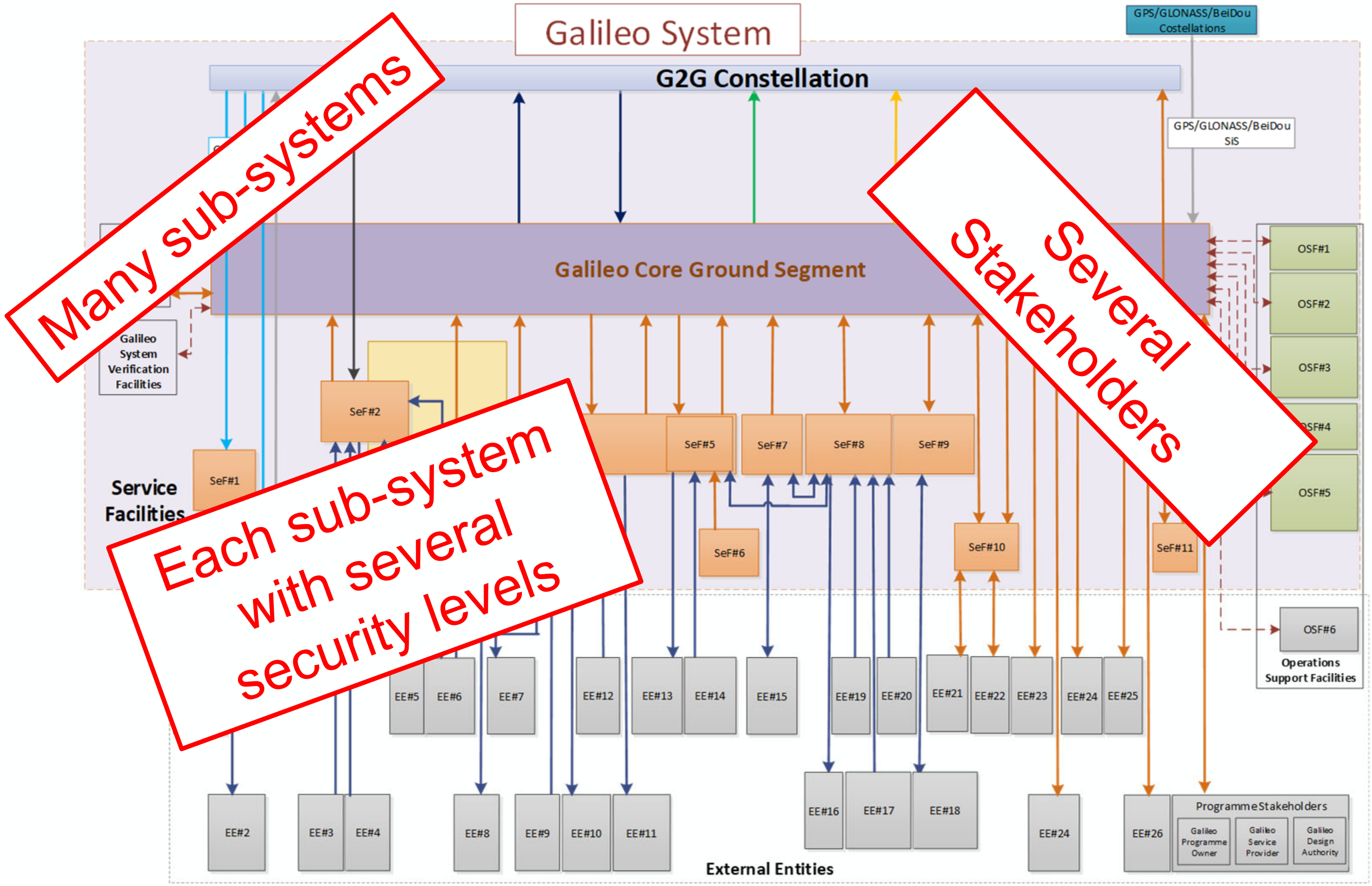
Collaboration among stakeholders



Many sub-systems

Several Stakeholders

Collaboration among stakeholders

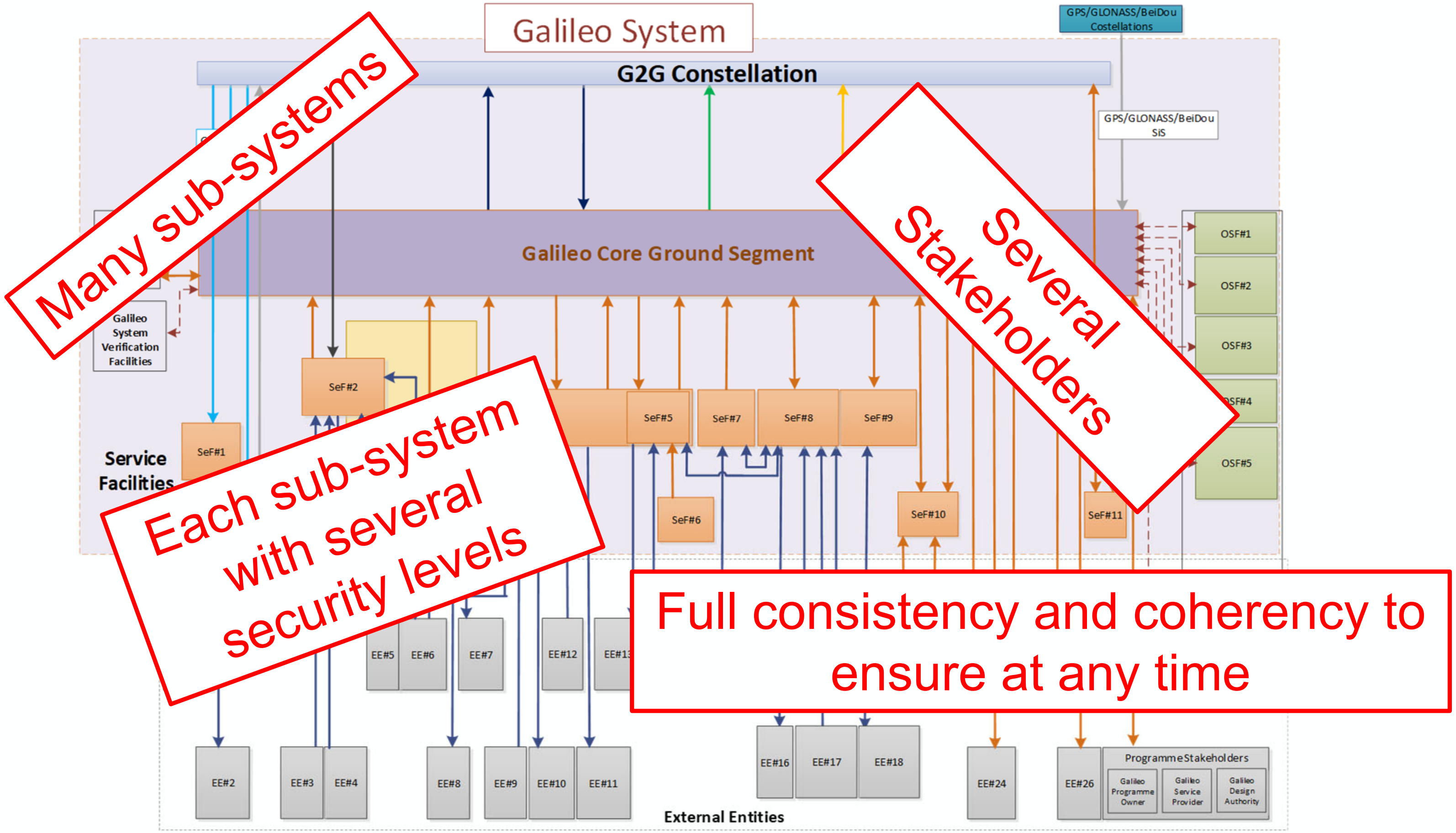


Many sub-systems

Several Stakeholders

Each sub-system with several security levels

Collaboration among stakeholders



Many sub-systems

Several Stakeholders

Each sub-system with several security levels

Full consistency and coherency to ensure at any time

Collaboration among stakeholders: from system to subsystem and back

The system-to-subsystem transition is a great asset for exchanging with suppliers. A model centred around a segment (logical component) can be provided to the segment supplier, who can send it back to the customer (ESA) with the proposed changes/requests for deviation (RFDs).

Image source: https://www.eclipse.org/community/eclipse_newsletter/2018/july/images/phases.png

Collaboration among stakeholders: from system to subsystem and back

The system-to-subsystem transition is a great asset for exchanging with suppliers. A model centred around a segment (logical component) can be provided to the segment supplier, who can send it back to the customer (ESA) with the proposed changes/requests for deviation (RFDs).

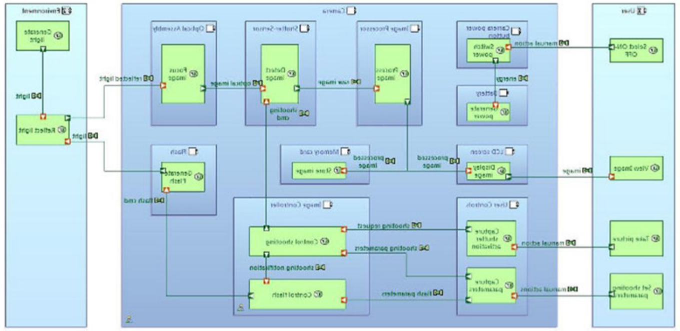


Image source: https://www.eclipse.org/community/eclipse_newsletter/2018/july/images/phases.png

Collaboration among stakeholders: from system to subsystem and back

The system-to-subsystem transition is a great asset for exchanging with suppliers. A model centred around a segment (logical component) can be provided to the segment supplier, who can send it back to the customer (ESA) with the proposed changes/requests for deviation (RFDs).

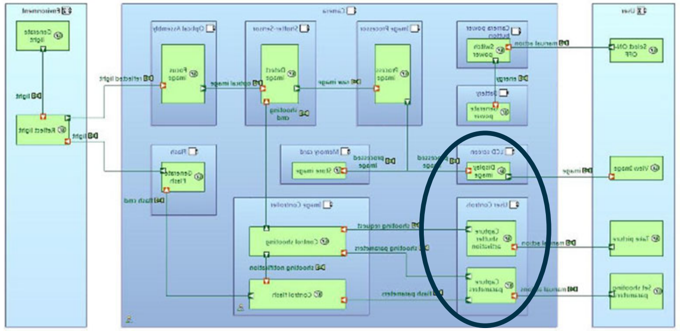
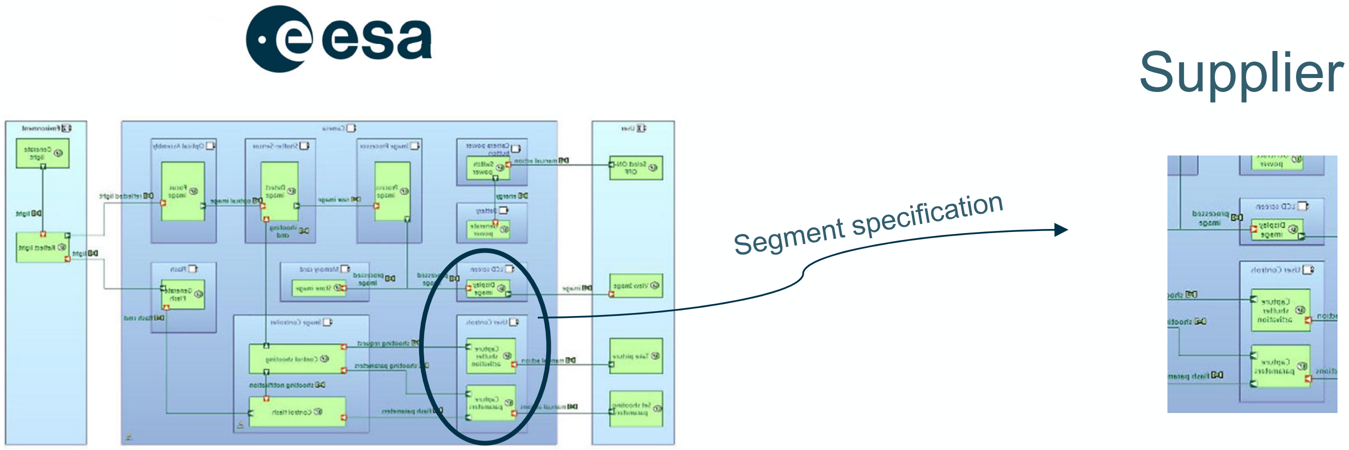


Image source: https://www.eclipse.org/community/eclipse_newsletter/2018/july/images/phases.png

Collaboration among stakeholders: from system to subsystem and back

The system-to-subsystem transition is a great asset for exchanging with suppliers. A model centred around a segment (logical component) can be provided to the segment supplier, who can send it back to the customer (ESA) with the proposed changes/requests for deviation (RFDs).



esa

Segment specification

Supplier

Image source: https://www.eclipse.org/community/eclipse_newsletter/2018/july/images/phases.png

Collaboration among stakeholders: from system to subsystem and back

The system-to-subsystem transition is a great asset for exchanging with suppliers. A model centred around a segment (logical component) can be provided to the segment supplier, who can send it back to the customer (ESA) with the proposed changes/requests for deviation (RFDs).

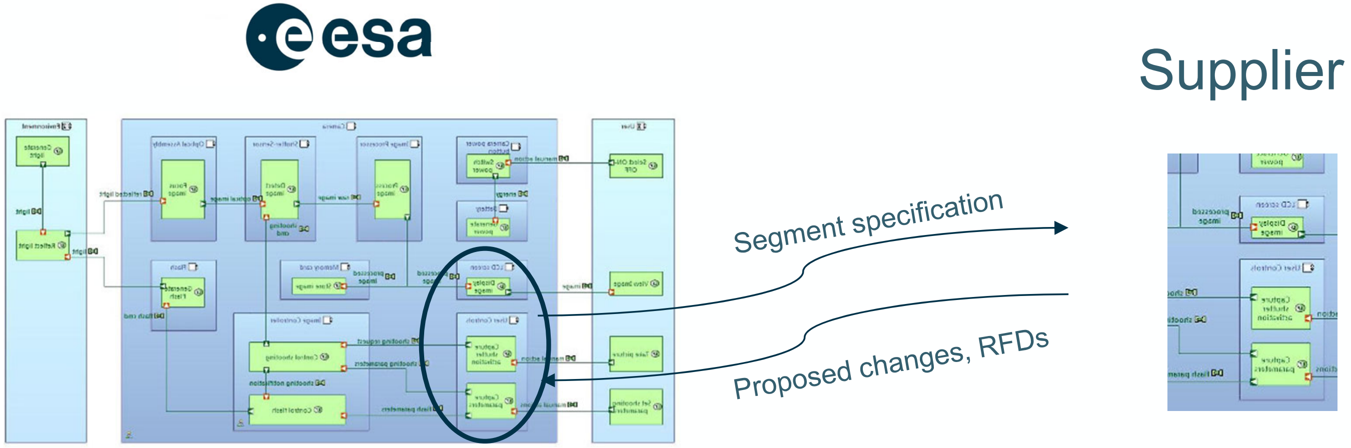
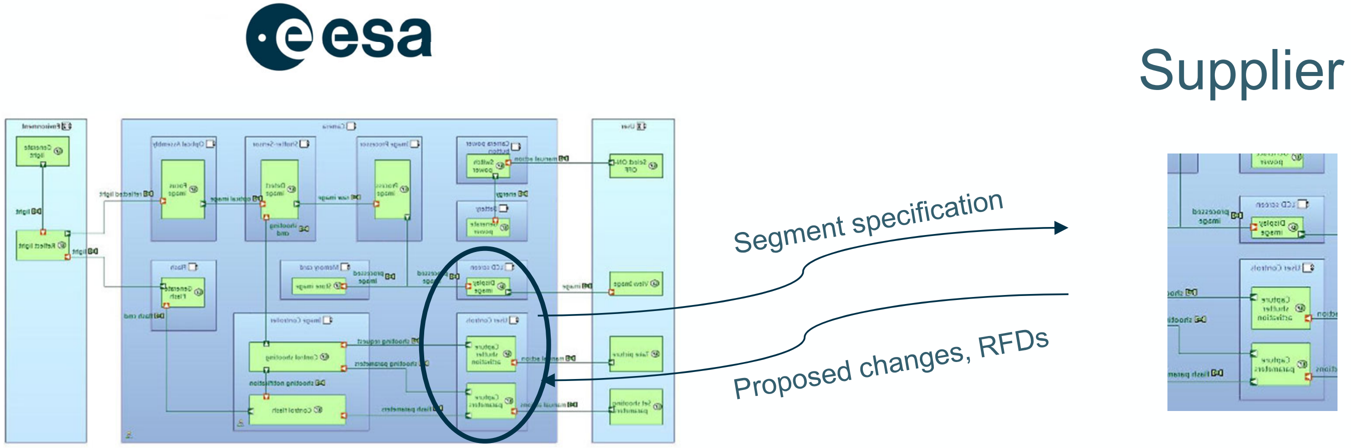


Image source: https://www.eclipse.org/community/eclipse_newsletter/2018/july/images/phases.png

Collaboration among stakeholders: from system to subsystem and back

The system-to-subsystem transition is a great asset for exchanging with suppliers. A model centred around a segment (logical component) can be provided to the segment supplier, who can send it back to the customer (ESA) with the proposed changes/requests for deviation (RFDs).



Supplier

Segment specification

Proposed changes, RFDs

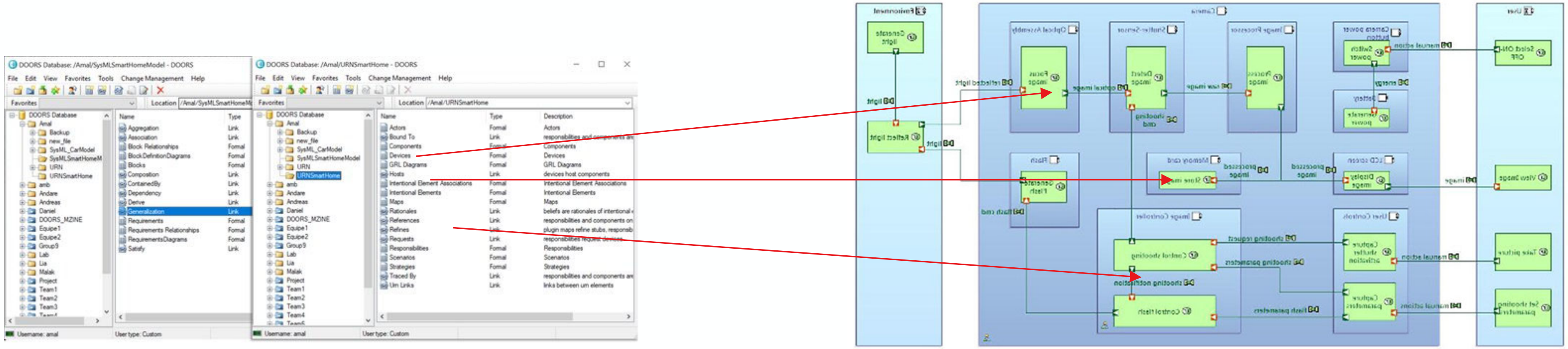
2 types of transition: vertical and horizontal. Vertical implies a transformation of the model, which is not compatible with the back-and-forth need → Choice of horizontal transition

Image source: https://www.eclipse.org/community/eclipse_newsletter/2018/july/images/phases.png

Collaboration among stakeholders: linking system design and requirements

(Textual) requirements to model traceability is key to check completeness, consistency, correctness

Possible tool solutions: requirements viewpoint, Reqtify, Publication for Capella



DOORS Database: /Amal/SysMLSmartHomeModel - DOORS

Location /Amal/SysMLSmartHomeM

Name	Type
Aggregation	Link
Association	Link
Block Relationships	Formal
BlockDefinitionDiagrams	Formal
Blocks	Formal
Composition	Link
ContainedBy	Link
Dependency	Link
Derive	Link
Generalization	Link
Requirements	Formal
Requirements Relationships	Formal
RequirementsDiagrams	Formal
Satisfy	Link

DOORS Database: /Amal/URNSmartHome - DOORS

Location /Amal/URNSmartHome

Name	Type	Description
Actors	Formal	Actors
Bound To	Link	responsibilities and components ar
Components	Formal	Components
Devices	Formal	Devices
GRL Diagrams	Formal	GRL Diagrams
Hosts	Link	devices host components
Intentional Element Associations	Formal	Intentional Element Associations
Intentional Elements	Formal	Intentional Elements
Maps	Formal	Maps
Rationales	Link	beliefs are rationales of intentional
References	Link	responsibilities and components on
Refines	Link	plugin maps refine stubs, responsib
Requests	Link	responsibilities request devices
Responsibilities	Formal	Responsibilities
Scenarios	Formal	Scenarios
Strategies	Formal	Strategies
Traced By	Link	responsibilities and components are
Urn Links	Link	links between urn elements

Username: amal

User type: Custom

Image source: https://www.eclipse.org/community/eclipse_newsletter/2018/july/images/phases.png
"Traceability Management of GRL and SysML Models", A. Anda and D. Amyot, SAM '20: 12th System Analysis and Modelling Conference

Collaboration among stakeholders: from engineers to customers

Communication with stakeholders such as management, customers, etc. is still an MBSE pain point.

The main need is to be able to understand and review the model contents without being MBSE/tool users. Then, provide feedback to engineers that can be integrated into the model.

Possible solutions: M2Doc exports, HTML exports, Python4Capella, Publication for Capella

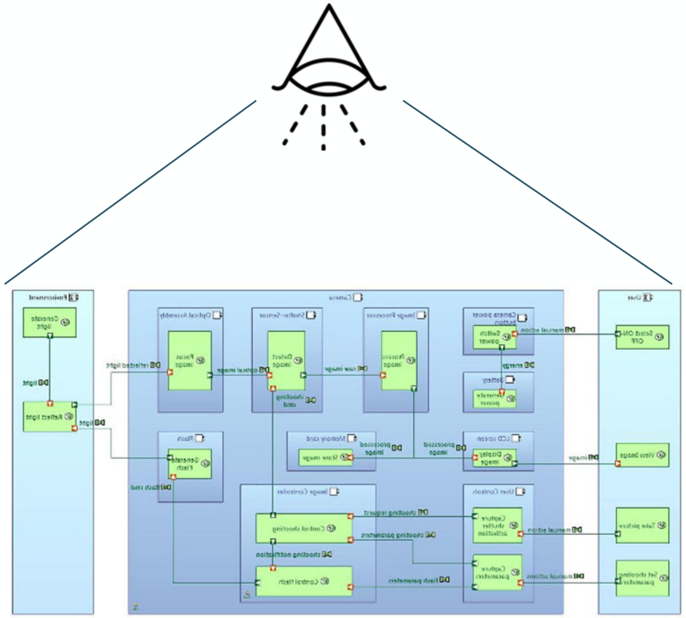


Image source: https://www.eclipse.org/community/eclipse_newsletter/2018/july/images/phases.png
Uniconlabs, https://www.flaticon.com/free-icon/eye_2838178

Outline of the presentation

- ❑ Introduction: Galileo and its complexity
- ❑ Collaborative end-to-end system design including security levels
- ❑ Collaboration among stakeholders
  - ➢ **sub-systems**
  - ➢ **linking design and specifications**
  - ➢ **reviewers/readers**
- ❑ **Concluding remarks**

Concluding remarks (1/2)

Collaboration is essential, at different levels of the project development and with different stakeholders

By using Capella and T4C we managed to obtain

- A model composed of several branches synchronised
- A consistent / coherent design developed by several system engineers concurrently
- Fast generation of documentation
- Partial export (html or word docs)

Concluding remarks (1/2)

Collaboration is essential, at different levels of the project development and with different stakeholders

By using Capella and T4C we managed to obtain

- A model composed of several branches synchronised
- A consistent / coherent design developed by several system engineers concurrently
- Fast generation of documentation
- Partial export (html or word docs)

The systems engineering team needs to define its project-specific methodology

Team training and working sessions are essential

==> Take time to build the methodology and team

Concluding remarks (2/2)

The synchronisation of model branches has been significantly used so far with good results

==> Maybe resolution of conflicts could be more explicit / easy

Concluding remarks (2/2)

The synchronisation of model branches has been significantly used so far with good results

==> Maybe resolution of conflicts could be more explicit / easy

Interactions between system and sub-systems are essential in large projects ==> the right level of transition of the model elements needs to be available to the designers

Concluding remarks (2/2)

The synchronisation of model branches has been significantly used so far with good results

==> Maybe resolution of conflicts could be more explicit / easy

Interactions between system and sub-systems are essential in large projects ==> the right level of transition of the model elements needs to be available to the designers

To apply MBSE all along the project life cycle, need to have good mechanisms for

- Dynamic traceability with requirements
- Reading and commenting the model for reviewers

Thank you !

Any questions?

ESA additional contact point:

Eric Bouton (ESA/ESTEC – NAV Directorate)