

# CAPELLA DAYS 2021 WARM-UP session

## Introduction to CAPELLA/ARCADIA and NASA Systems Engineering handbook: Modeling overview with the HUBBLE Space Telescope

*DROUIN Remy*



- **Avionic systems engineer in the French Air Force**
- **Team Leader in automotive industry designing connected solutions**
- **Program Manager in elevator industry designing connected solutions**
- **Head of system department in defense industry designing High Energy Laser Systems**
- **Lecturer for French universities introducing Systems Engineering and Model-Based Systems Engineering**

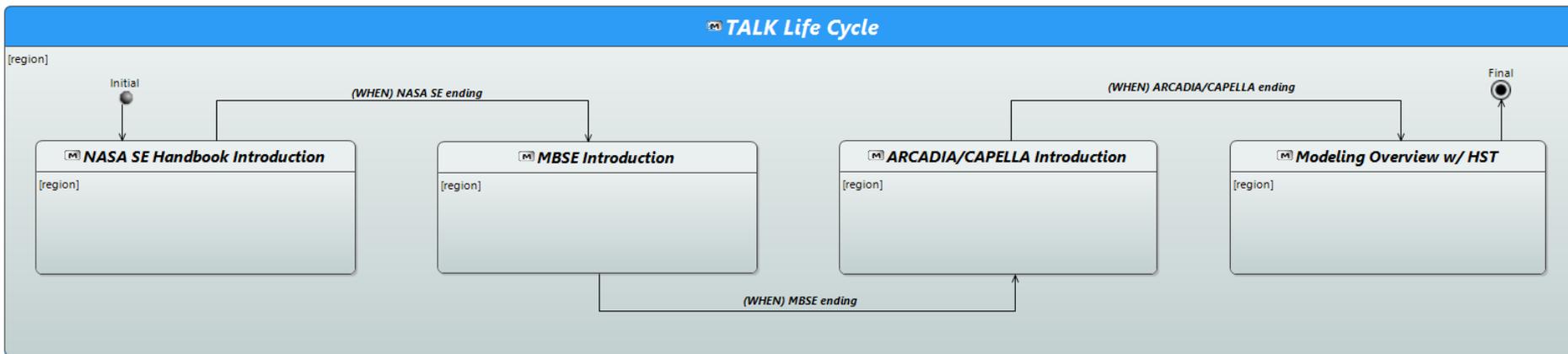
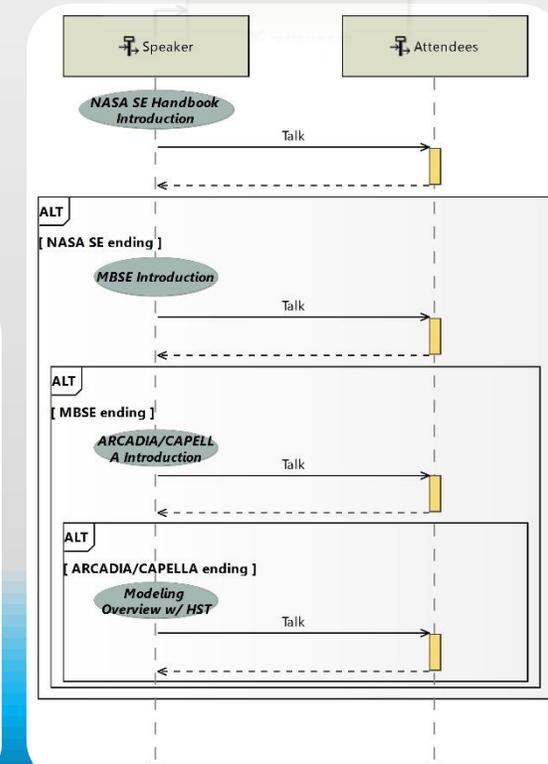
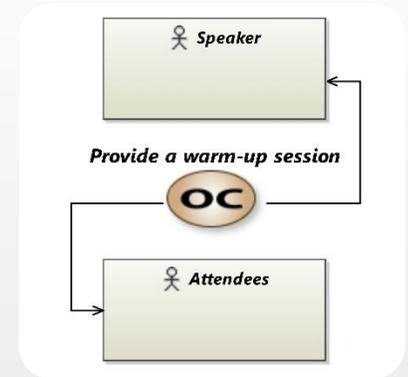
*“The NASA System Engineering (SE) handbook aims to provide general guidance and information on systems engineering, as it should be applied throughout NASA. The handbook introduces 3 common technical processes. One of these, is the System Design Process, describing the stakeholders expectations, requirements definition, logical decomposition and design solution definition. The 4 activities can be supported by a Model-Based Systems Engineering (MBSE) approach. To do so, an appropriate method and tool is necessary as the one provided by the ARCHitecture Analysis & Design Integrated Approach. ARCADIA, with its modeler CAPELLA, is a MBSE solution supporting system modeling activities. Based on 4 architectural layers, which are Operational Analysis, System Analysis, Logical and Physical Architecture, it is a structured architecture engineering method for defining and validating multi-domain systems. This talk will present an educational overview of the ARCADIA methodology and System Design Process from the NASA SE, by introducing MBSE artefacts for space system. The HUBBLE Space Telescope (HST) is a Cassegrain reflector telescope. Orbiting above the earth, HST elaborates a clear view of the universe free from the blurring and absorbing effects of the atmosphere. In order to illustrate the journey throughout CAPELLA, the HST will be introduced, as example, based on public information available.”*

➤ NASA Systems Engineering Handbook

➤ MBSE

➤ CAPELLA/ARCADIA

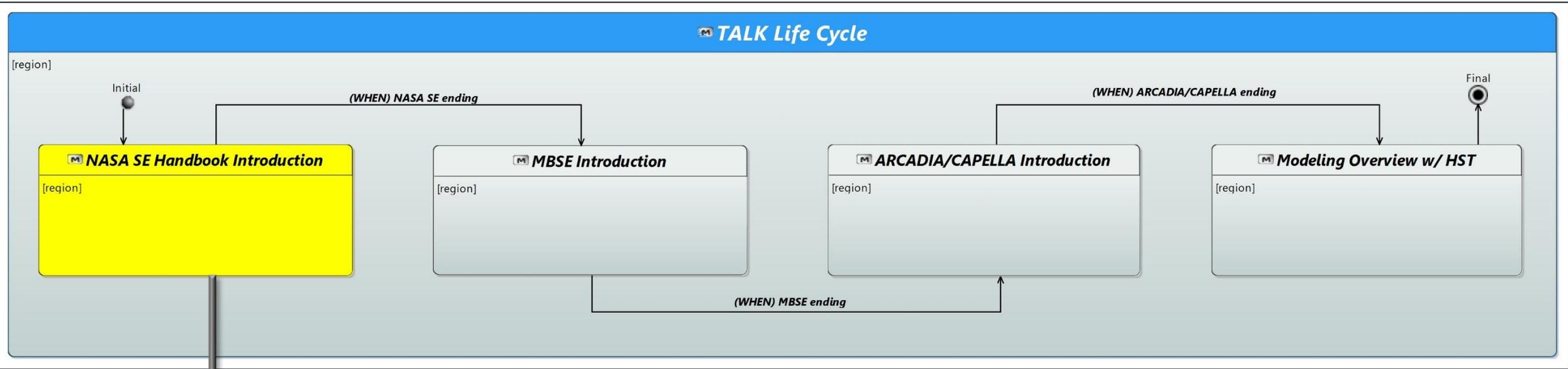
➤ Modeling overview with HUBBLE Space Telescope



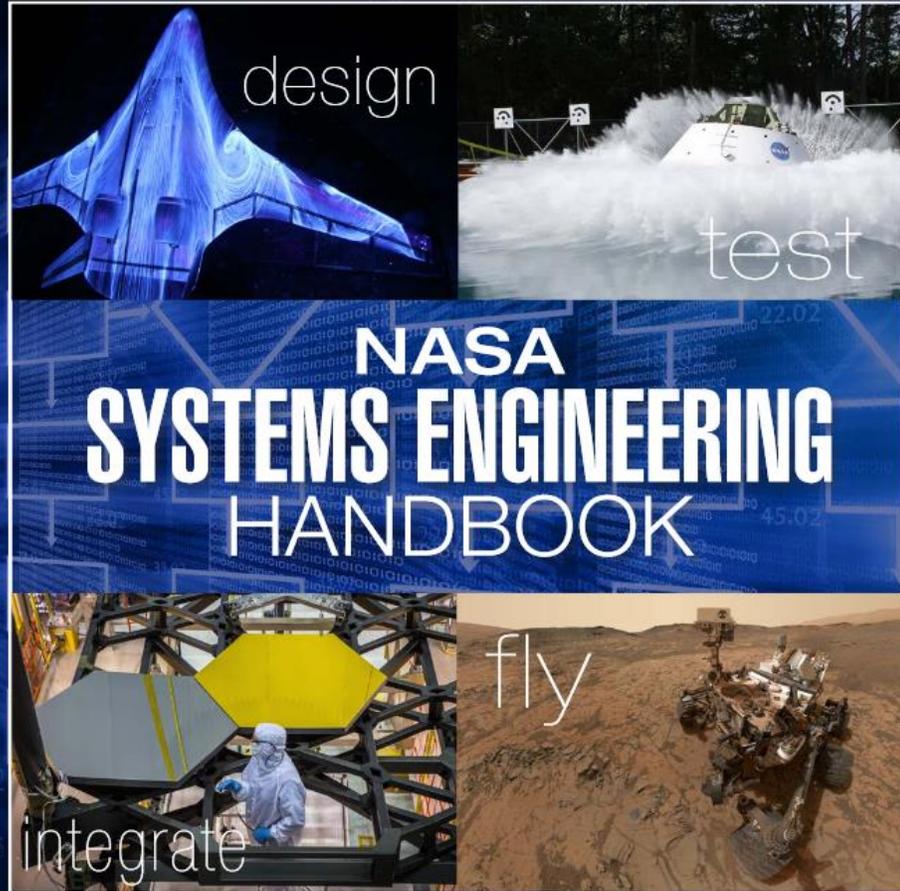
**« All models are wrong, but some are useful »**

George E.P.Box (British statistician)





This block contains a collage of images related to NASA Systems Engineering. On the left is the cover of the 'NASA Systems Engineering Handbook'. To its right is a slide titled 'Expanded Guidance for NASA Systems Engineering' with a diagram showing the relationship between MBSE, CAPPELLA, and HST. Below these are several diagrams, including a Functional Flow Block Diagram (FFBD) and a diagram of a system with external systems and interfaces.

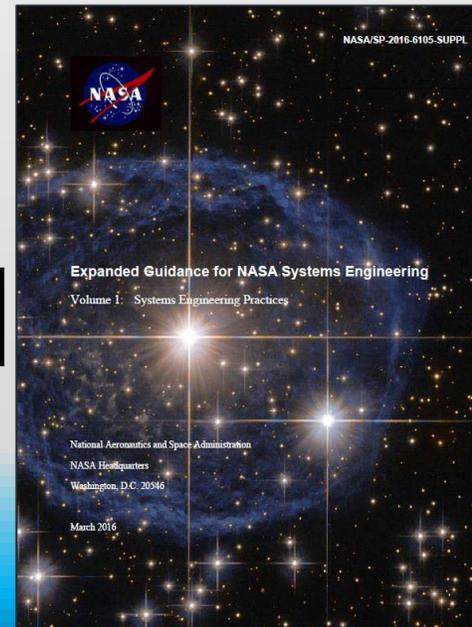


www.nasa.gov

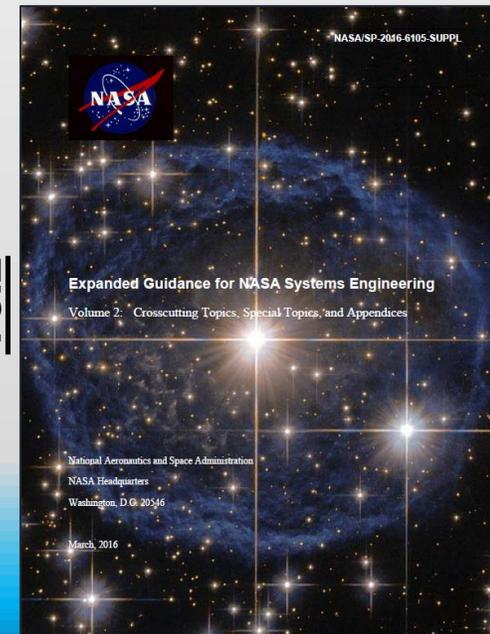
NASA Systems Engineering Handbook is intended to provide general guidance and information on systems engineering that will be useful to the NASA community. It provides a generic description of Systems Engineering (SE) as it should be applied throughout NASA. This handbook describes systems engineering best practices that should be incorporated in the development and implementation of large and small NASA programs and projects.

## Expanded Guidance for NASA Systems Engineering

Vol1



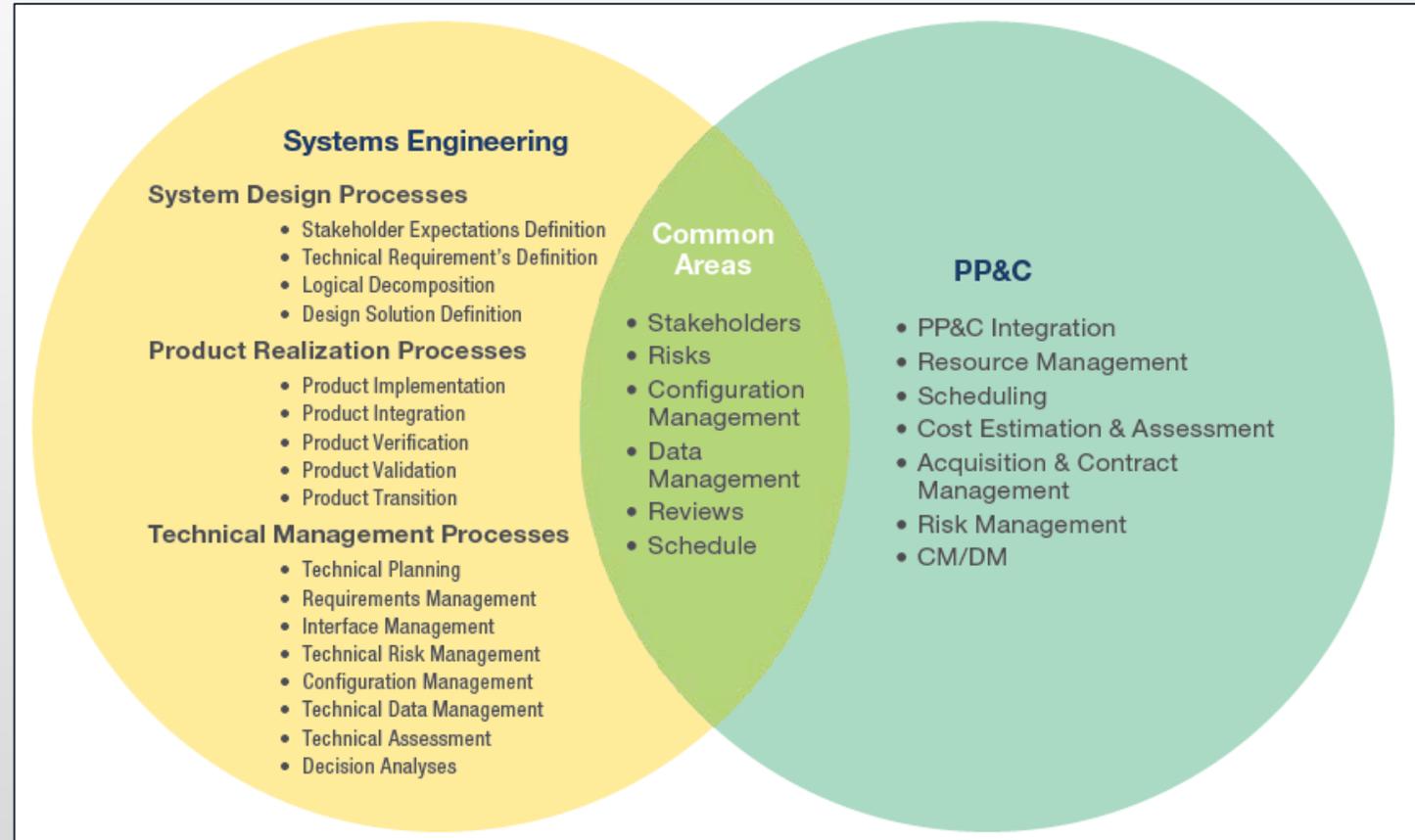
Vol2



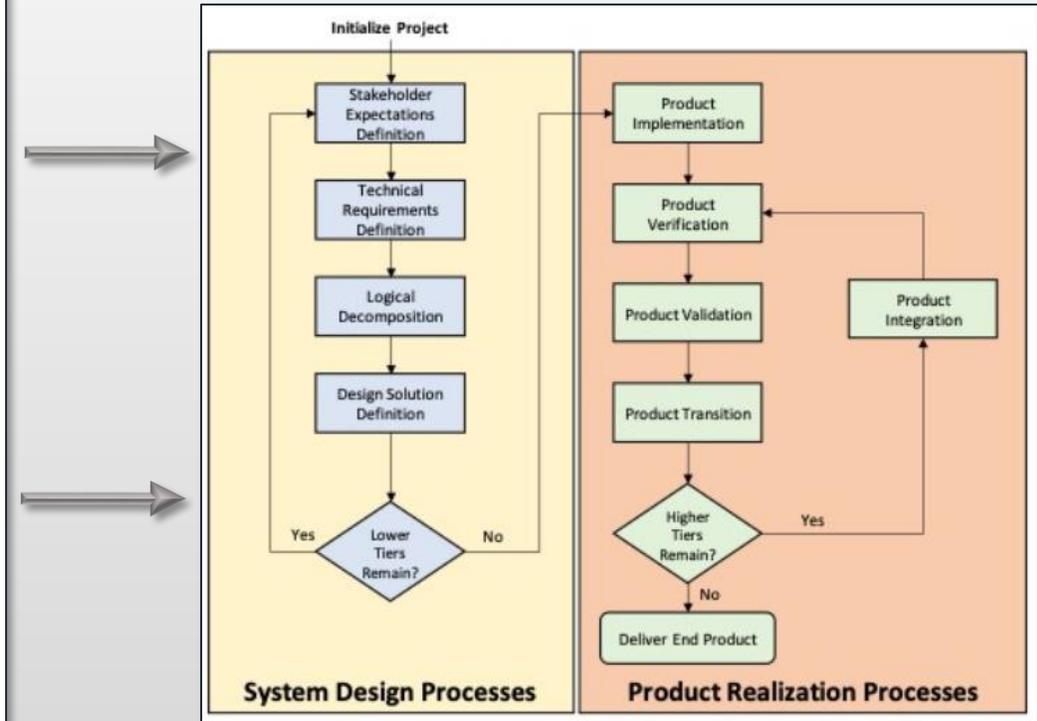
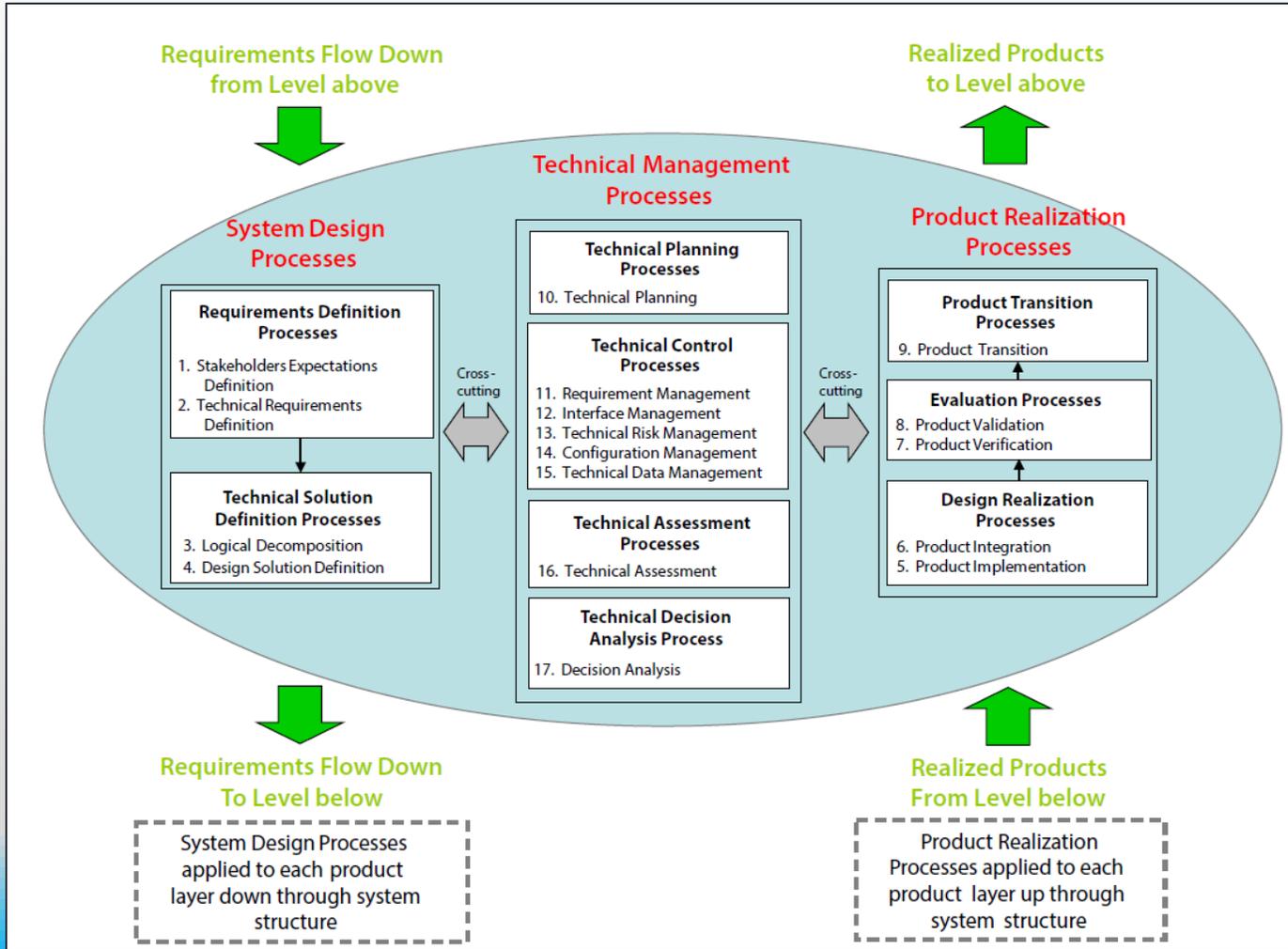
“Systems engineering” is defined as a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system.

The systems engineer usually plays the key role in leading:

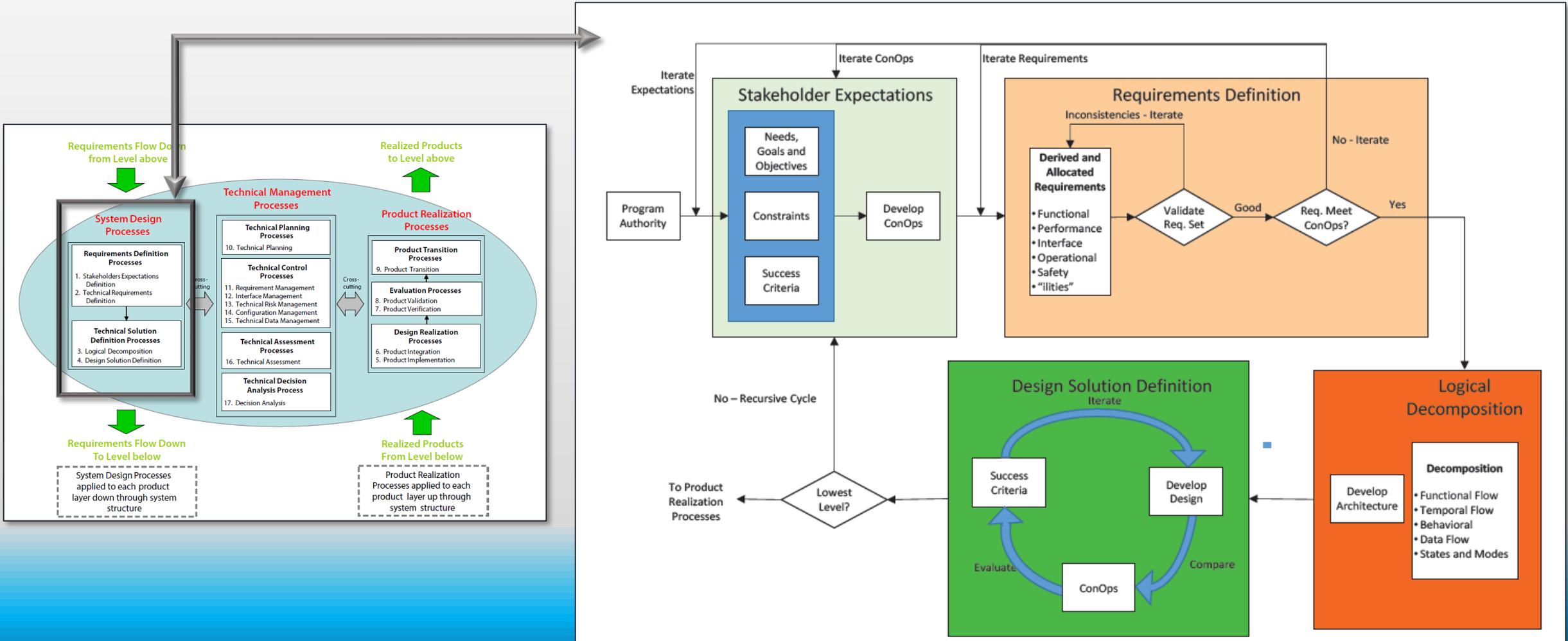
- the development of the concept of operations and resulting system architecture
- defining boundaries
- defining and allocating requirements
- evaluating design tradeoffs
- balancing technical risk between systems
- defining and assessing interfaces, providing oversight of verification and validation activities



There are 3 sets of common technical processes: system design, product realization, and technical management.

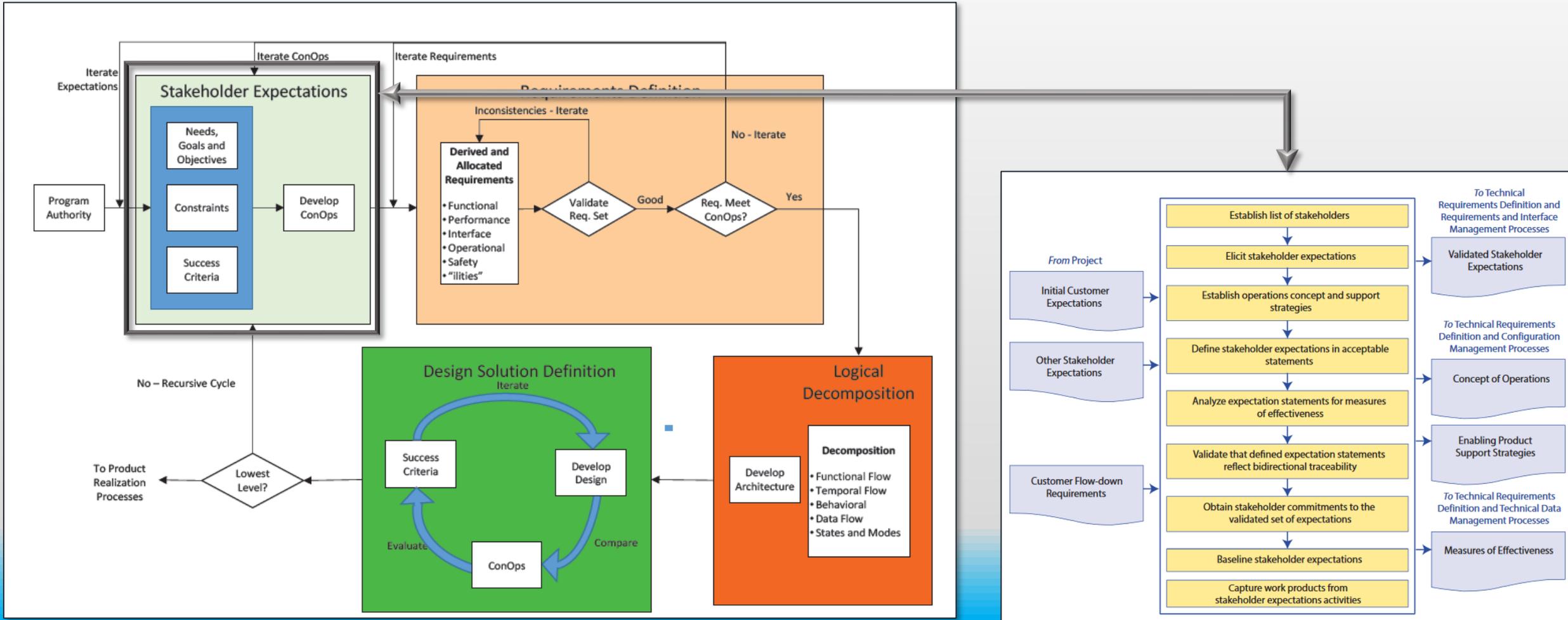


The four system design processes are used to define and baseline stakeholder expectations, generate and baseline technical requirements, decompose the requirements into logical and behavioral models, and convert the technical requirements into a design solution that will satisfy the baselined stakeholder expectations.



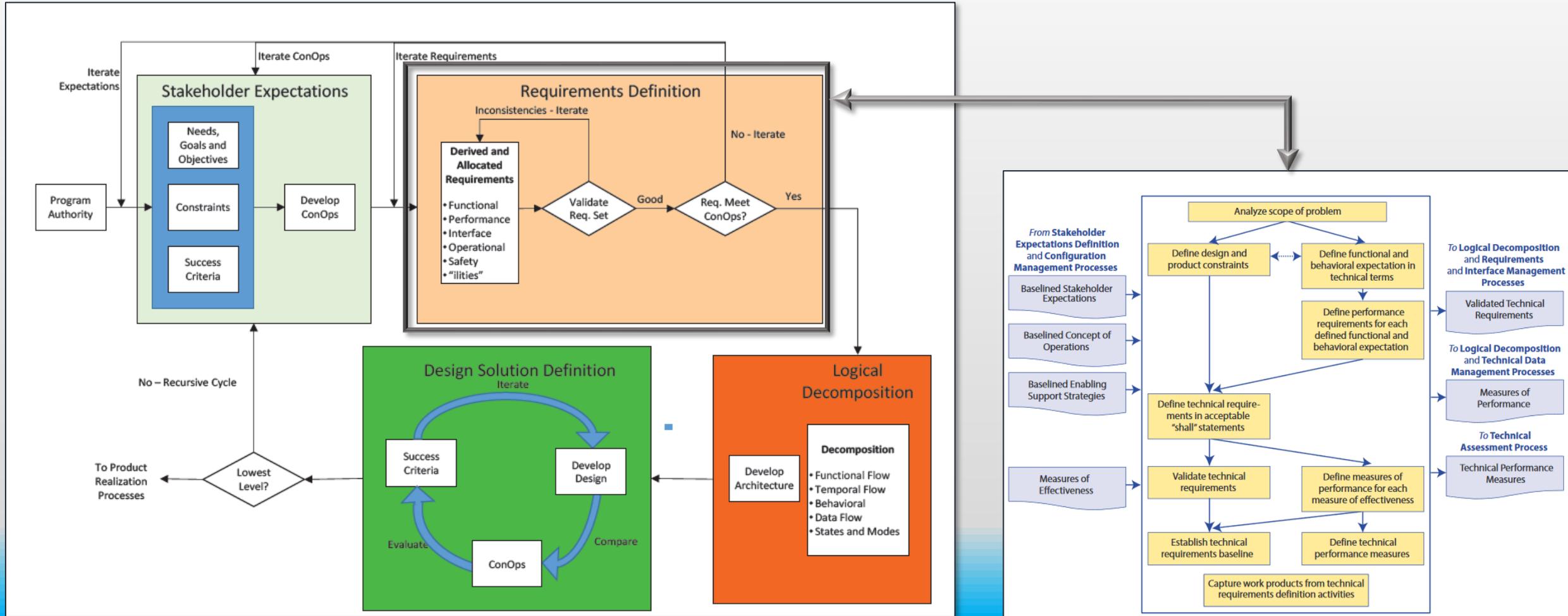
## Stakeholder expectations

The main purpose of this process is to identify who the stakeholders are and how they intend to use the product.



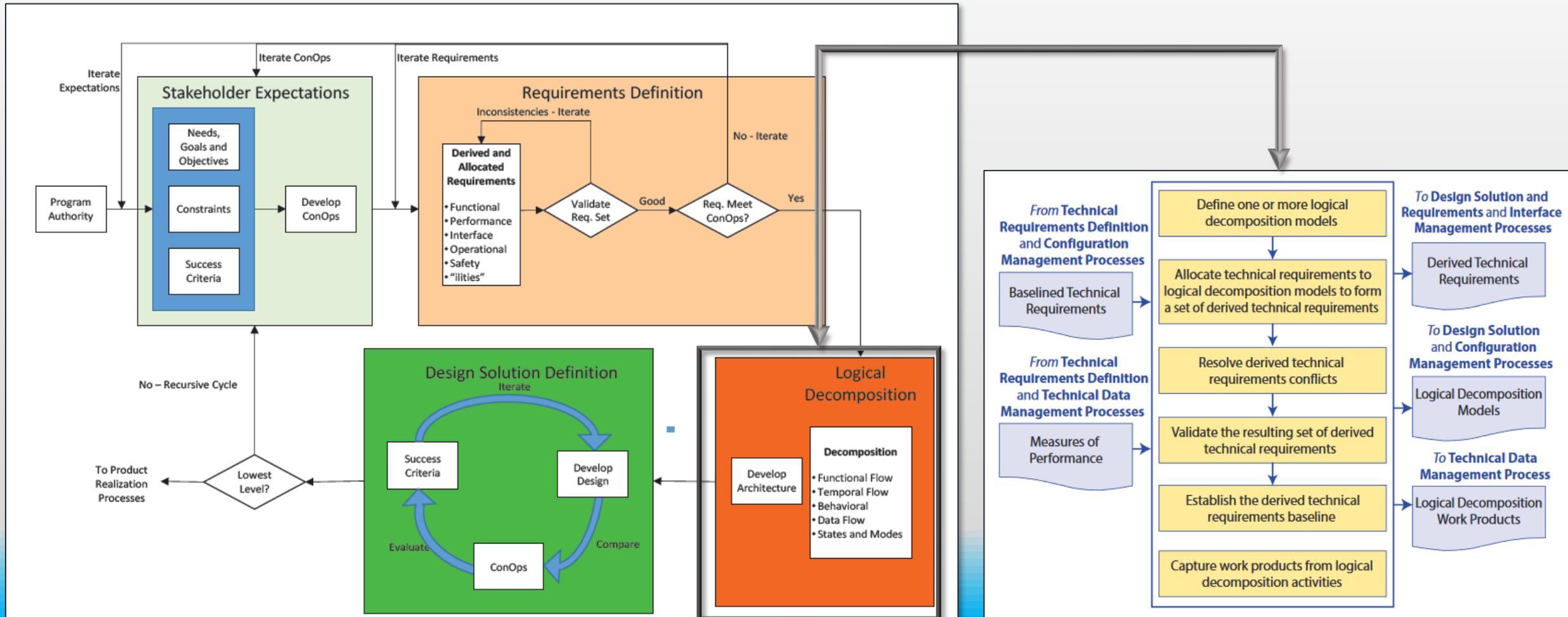
## Requirements definition

The Requirements Definition process transforms the stakeholder expectations into a definition of the problem.



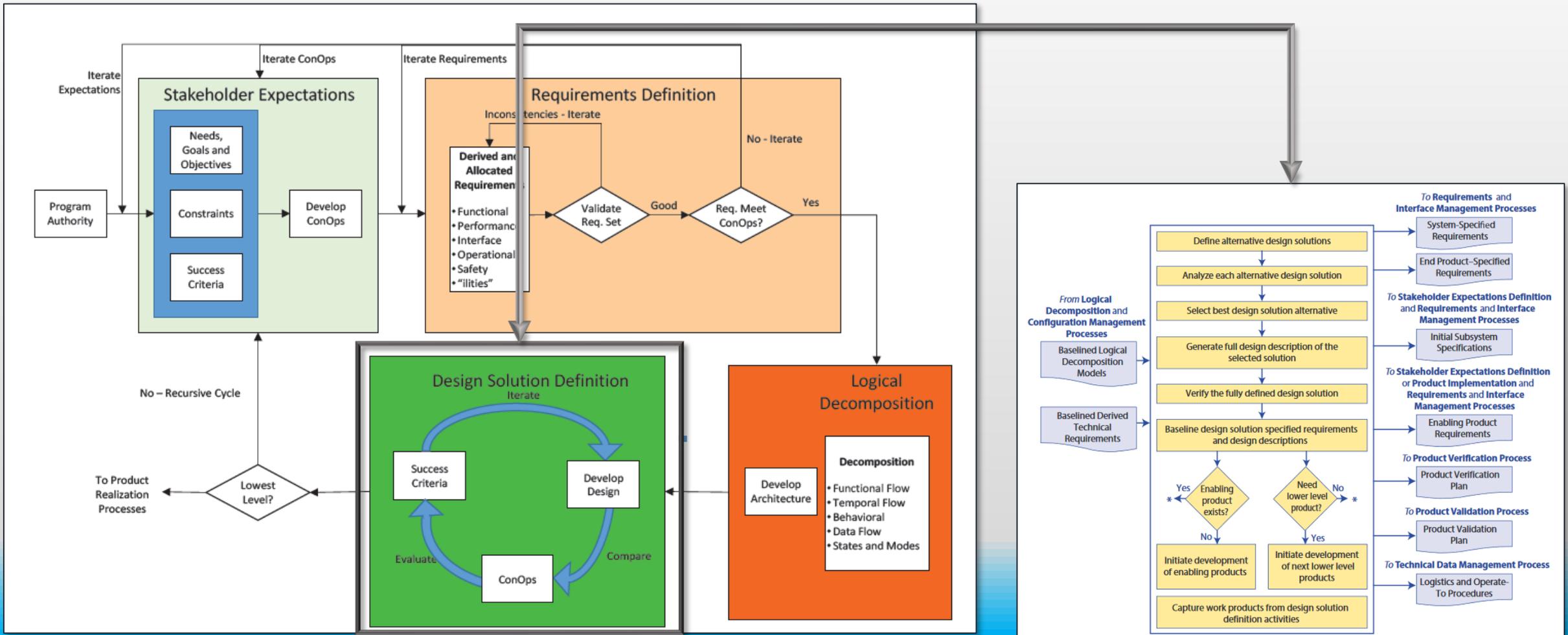
## Logical decomposition

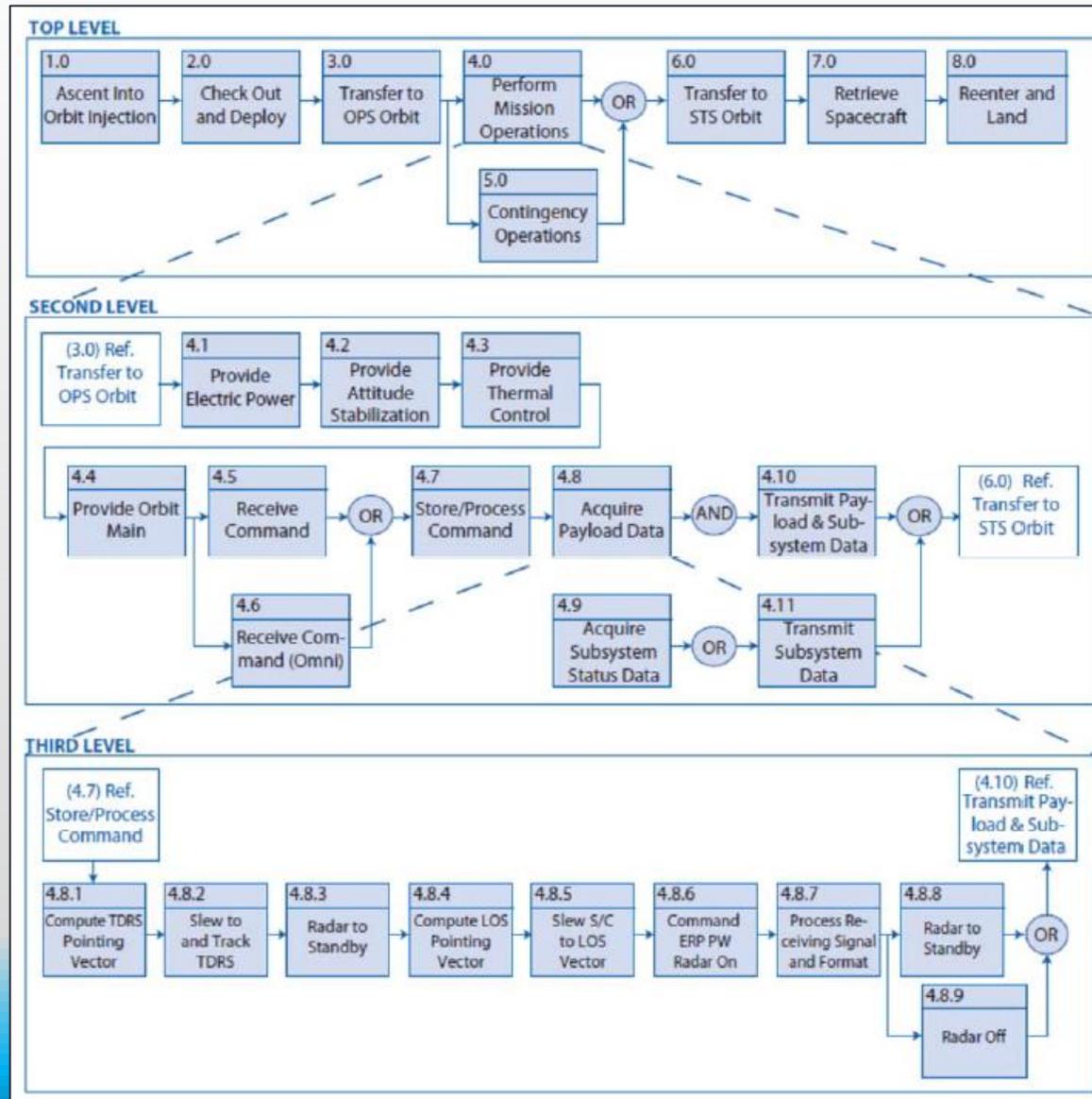
Logical decomposition utilizes functional analysis to create a system architecture and to decompose top-level requirements and allocate them down to the lowest desired levels.



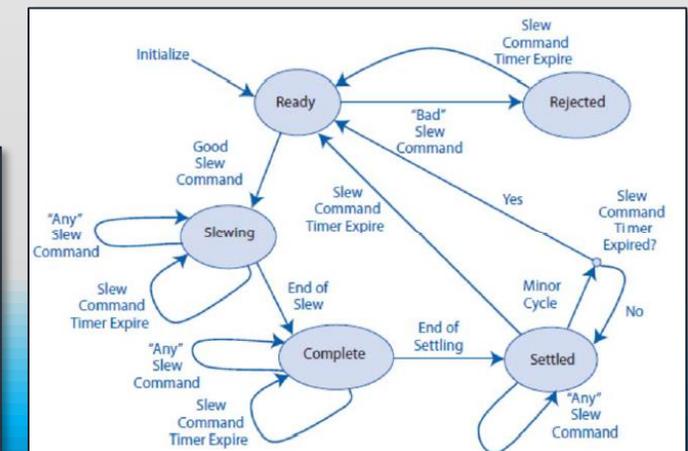
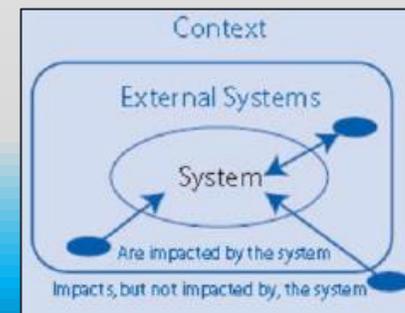
## Design solution definition

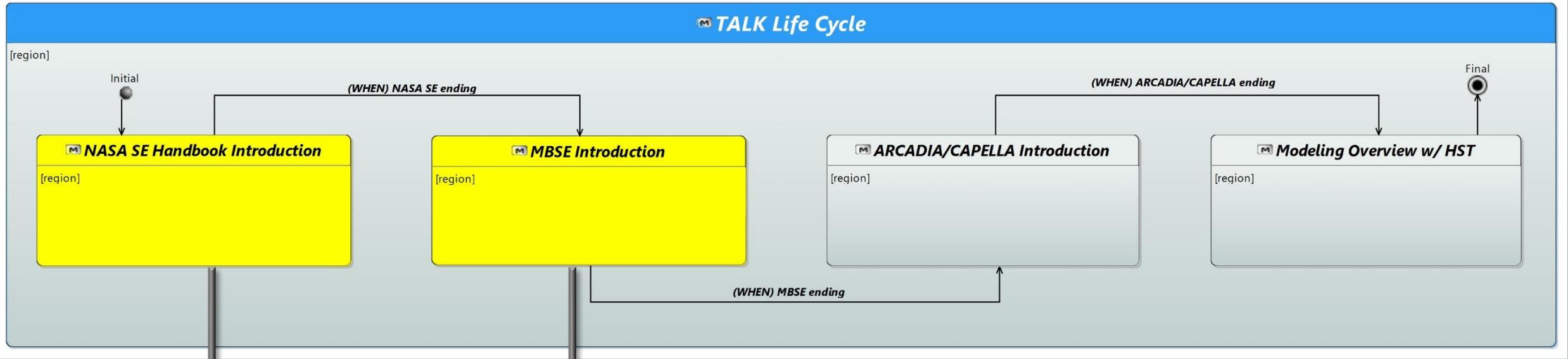
The Design Solution Definition process is used to translate the requirements derived from the stakeholder expectations and the outputs of the Logical Decomposition process into a design solution.





- Functional analysis can be performed by Functional Flow Block Diagrams (FFBDs). They are functionally oriented and not solution oriented
- It is made of functional block representing an action to be accomplished
- The functional architecture is developed using a decomposition and interactions between each functions (functional flows)
- Functional analysis looks across all life cycle processes



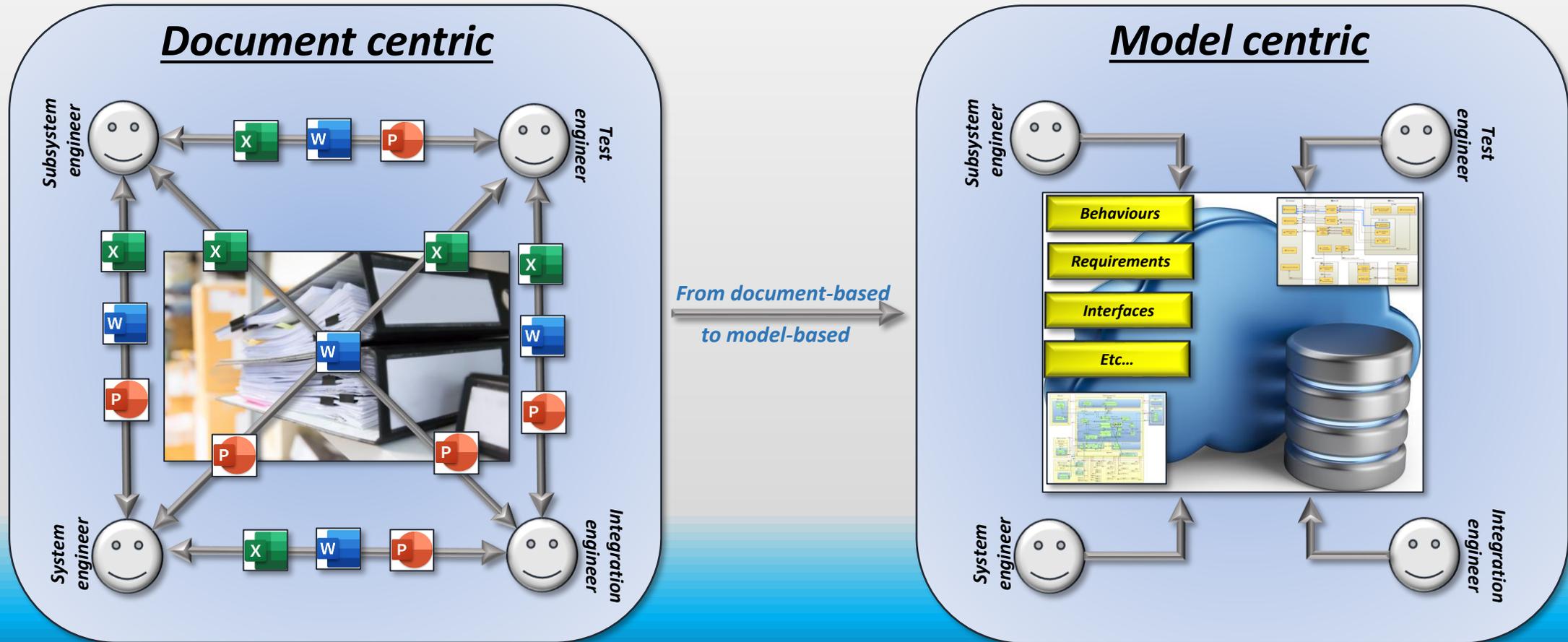


This block contains a collage of slides from the 'NASA Systems Engineering Handbook' presentation. It includes the handbook cover, a slide titled 'Expanded Guidance for NASA Systems Engineering' with a diagram, and several slides showing functional decomposition diagrams and text explaining the process of developing functional architecture using decomposition and interactions between functions.

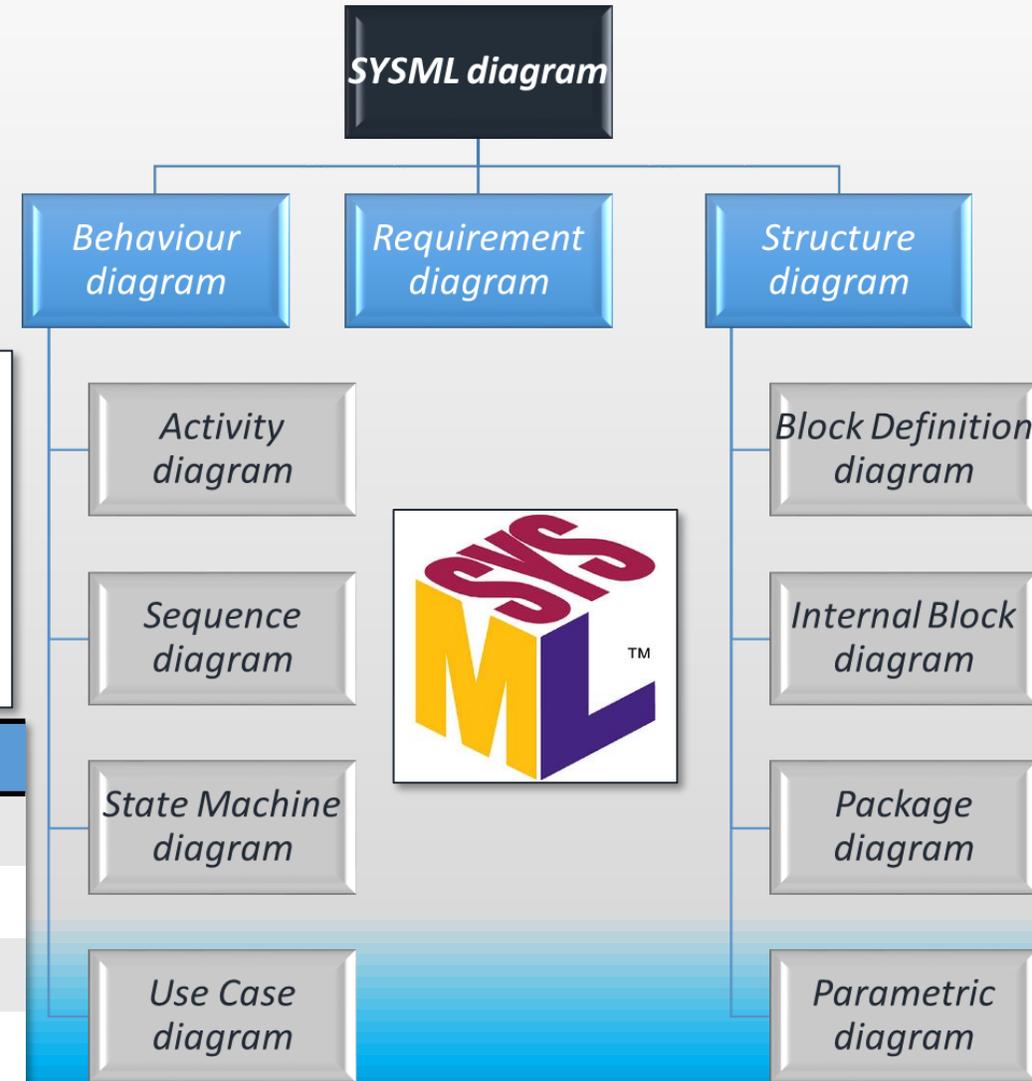
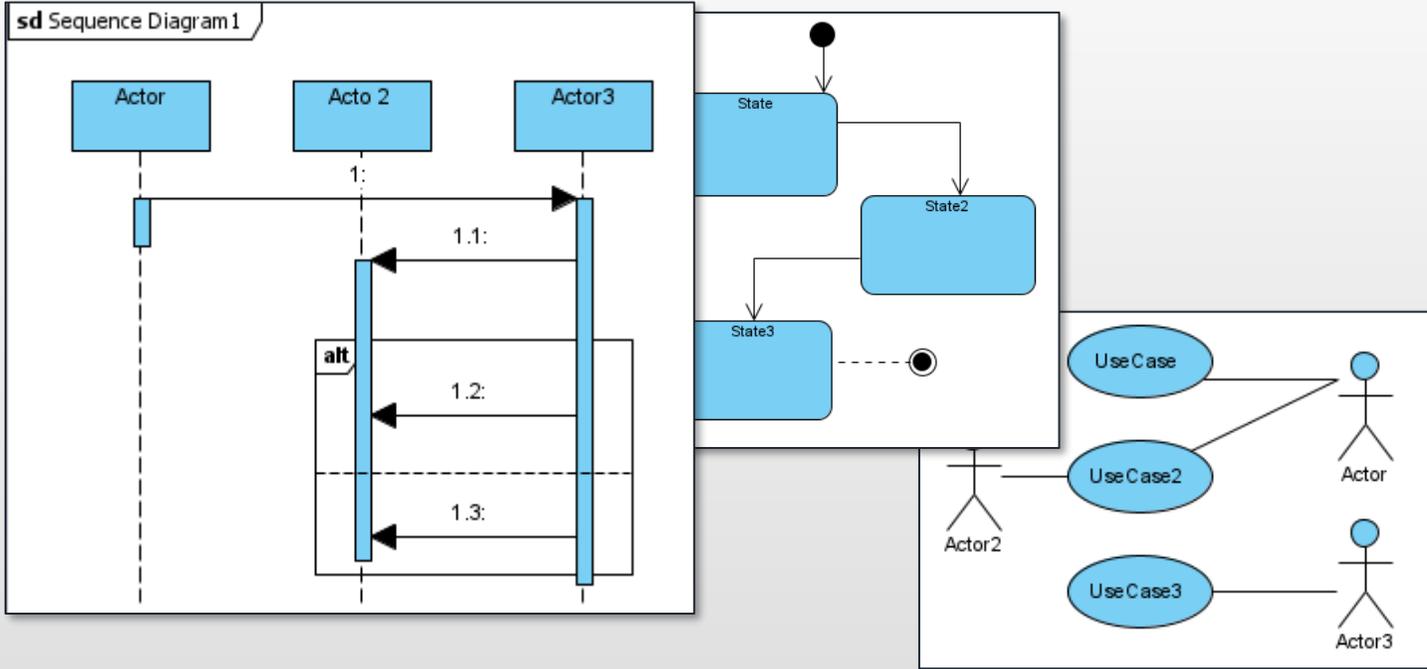
This block contains a collage of slides from the 'MBSE Introduction' presentation. It features a comparison between 'Document centric' and 'Model centric' approaches with diagrams. Below this is a table of modeling objectives:

Modeling objectives (examples)
Characterize an existing system
Specify and design a new or modified system
Evaluate a system
Train users on how to operate or maintain a system

**Model-based systems engineering (MBSE)** is a systems engineering methodology that focuses on creating and exploiting domain models as the primary means of information exchange between engineers, rather than on document-based information exchange. MBSE is a formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.

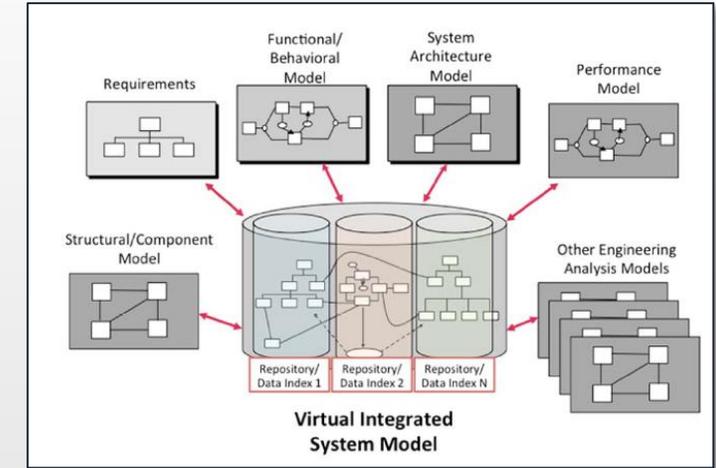
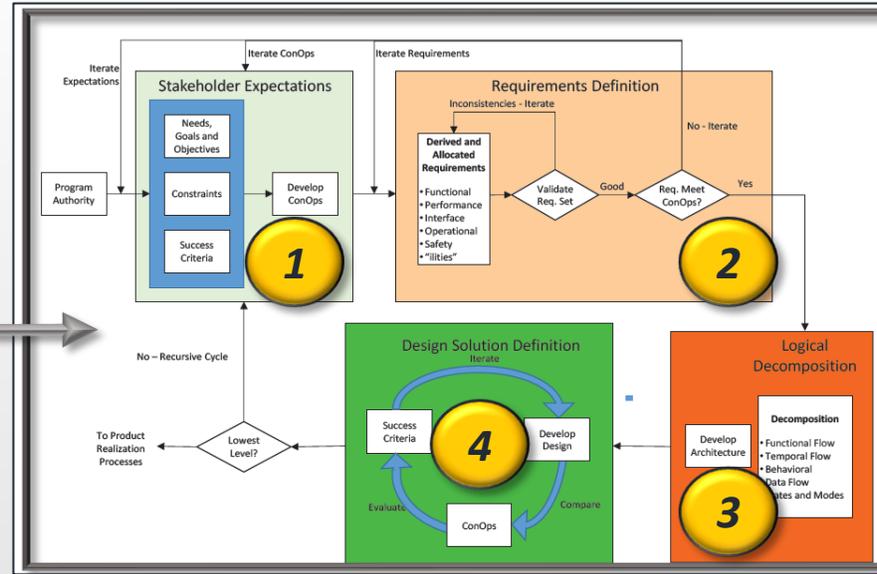
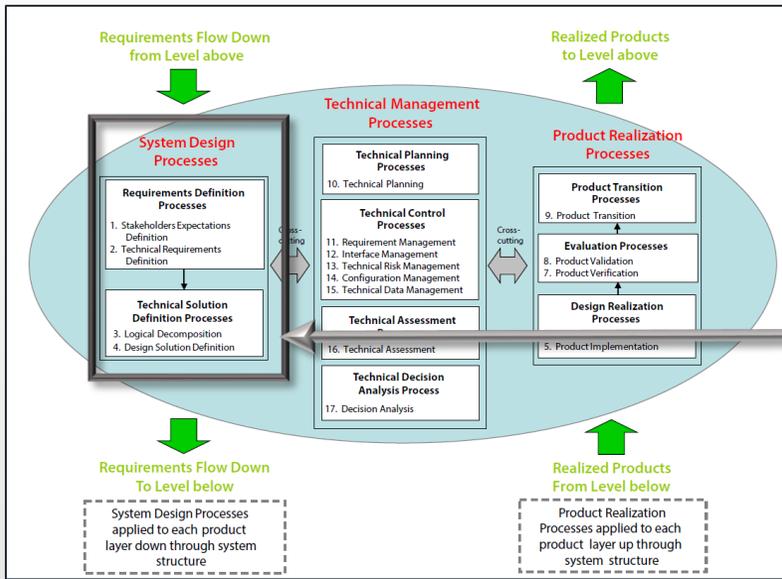


A model consists of elements that represent requirements , design element, and their relationships. SYSML is a graphical modeling language that supports modeling activities.



### Modeling objectives (sample)

- Characterize an existing system
- Specify and design a new or modified system
- Evaluate a system
- Train users on how to operate or maintain a system



## System Design Processes MBSE Contributions (NASA)

<b>1</b>	Stakeholders expectations definition	Need, goals and objectives are kept within the model and form the top tier of eventual requirements flowdown
<b>2</b>	Technical requirements	Requirements are kept within the models
<b>3</b>	Logical decomposition	Requirements can be categorized into functional, behavioral, performance etc... These can be used to develop functional block and behavior diagrams
<b>4</b>	Design Solution Definition	Allows integration of information and designs from different engineering domains supporting the single source of truth

Model-based systems engineering does not affect process but will enable the opportunity for overall better quality, lower cost, and lower risk.

## Overall MBSE benefits (sample):

Enhance communication

Reduce development risk

Encourage collaboration

Manage complexity

Automatic document generation

Reuse of existing models in several projects

Better requirements traceability

More stakeholder involvement

Digitalization

Single source of truth

## NASA MBSE benefits (sample)

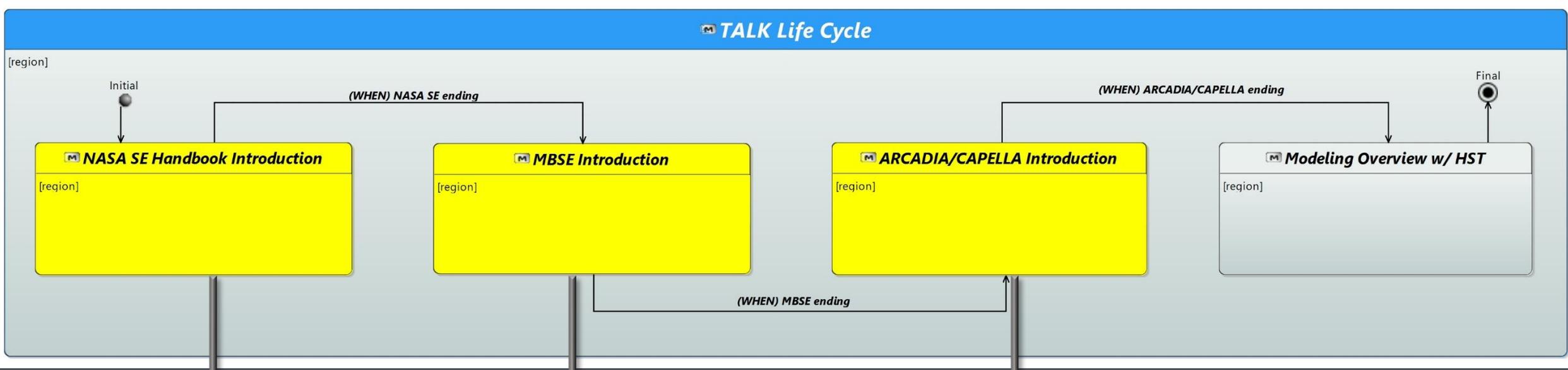
Greater consistency of all products because any single piece of design information can be expressed authoritatively in a single place that can later be referred to by others for decisions, derivations, or formation of artifacts

Better visibility into the salient characteristics of a system because multiple views can be created that succinctly address specific stakeholder concerns

Model-based artifacts can be generated automatically, lowering the effort to keep them up to date with the result that artifacts can always match the best available information

Navigation, traceability, and interrogation of information are facilitated in the model-based approach

Can be less investment lost in erroneous design because sometimes the model reveals a flaw as soon as it is created, enabling correction before downstream work is done, work that would be invalid if the upstream mistake were not corrected immediately



**NASA Systems Engineering Handbook** Speaker: DROUIN Remy

NASA Systems Engineering Handbook is intended to provide general guidance and information on systems engineering that will be useful to the NASA community. It provides a generic description of Systems Engineering (SE) as it should be applied throughout NASA. This handbook describes systems engineering best practices that should be incorporated in the development and implementation of large and small NASA programs and projects.

**Expanded Guidance for NASA Systems Engineering**

Speaker: DROUIN Remy

intend to use the product.

Speaker: DROUIN Remy

in be performed by Functional Analysis (FFBA). They are not solution oriented.

Speaker: DROUIN Remy

block representing an action

Then the functional architecture is developed using a decomposition and interactions between each functions (functional flows).

Functional analysis looks across all life cycle processes.

**MBSE Introduction** Speaker: DROUIN Remy

**Model-based systems engineering (MBSE)** is a systems engineering methodology that focuses on creating and exploiting domain models as the primary means of information exchange between engineers, rather than on document-based information exchange. MBSE is a formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.

**Document centric** vs **Model centric**

Speaker: DROUIN Remy

if eventual requirements

Speaker: DROUIN Remy

Modeling objectives (examples)

- Characterize an existing system
- Specify and design a new or modified system
- Evaluate a system
- Train users on how to operate or maintain a system

**ARCADIA/CAPELLA - Introduction** Speaker: DROUIN Remy

Architecture Analysis & Design Integrated Approach (ARCADIA) is a structured architecture engineering method for defining and validating multi-domain systems, based on architecture-centric and model-driven engineering activities. ARCADIA is a method based on functional analysis and focuses on developing the system by starting from needs analysis and solution development up to integrated verification and validation.

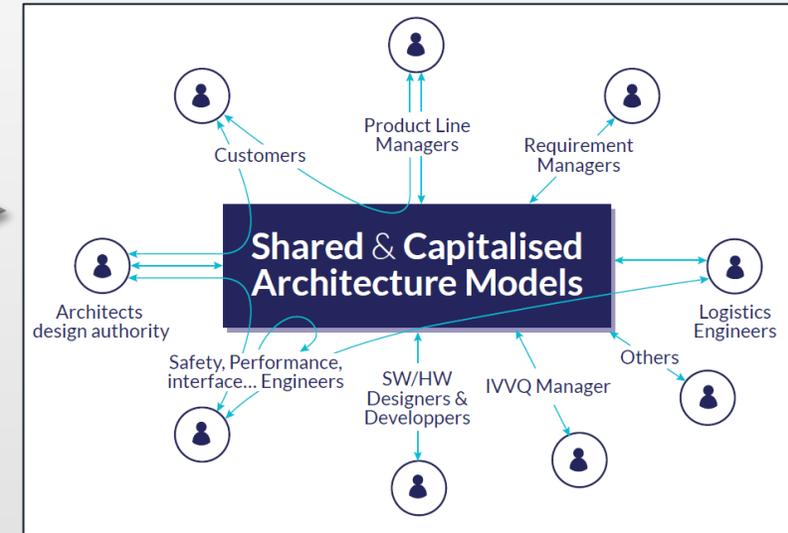
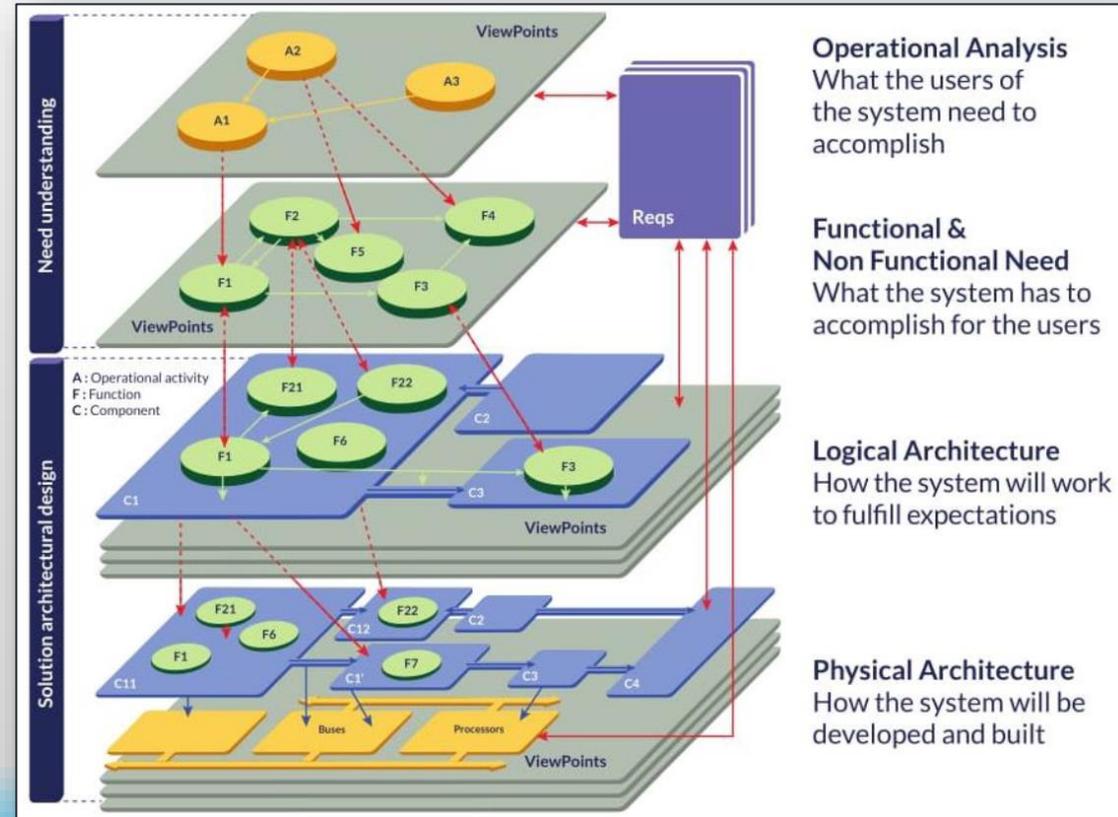
- Operational Analysis (Operational Architecture)
- Functional & Behavioral Analysis (Functional Architecture)
- Physical Architecture (Physical Architecture)
- Logical Architecture (Logical Architecture)
- Shared - Capitalized Architecture Matrix

- Ensure engineering-wide collaboration by sharing the same reference architecture
- Master the complexity of systems and architectures
- Define the best optimal architectures through trade-off analysis
- Master different engineering levels and traceability with automated transition and information refinement

Speaker: DROUIN Remy

Speaker: DROUIN Remy

Architecture Analysis & Design Integrated Approach (ARCADIA) is a structured architecture engineering method for defining and validating multi-domain systems, based on architecture-centric and model-driven engineering activities. ARCADIA is a method based on functional analysis and focuses on developing the system by starting from needs analysis and solution development up to integrated verification and validation.



- Ensure engineering-wide collaboration by sharing the same reference architecture
- Master the complexity of systems and architectures
- Define the best optimal architectures through trade-off analysis
- Master different engineering levels and traceability with automated transition and information refinement



Designing complex and critical systems, and more generally architectures that are subject to multiple functional and non-functional constraints, is an activity which requires a level of rigor that can only be provided by formalized and toolled modeling approaches like the ones based on Arcadia/Capella and SysML tools.

## Method

The Arcadia method enforces an approach structured on different engineering perspectives establishing a clear separation between system context and need modeling (operational need analysis and system need analysis) and solution modeling (logical and physical architectures).

## Language

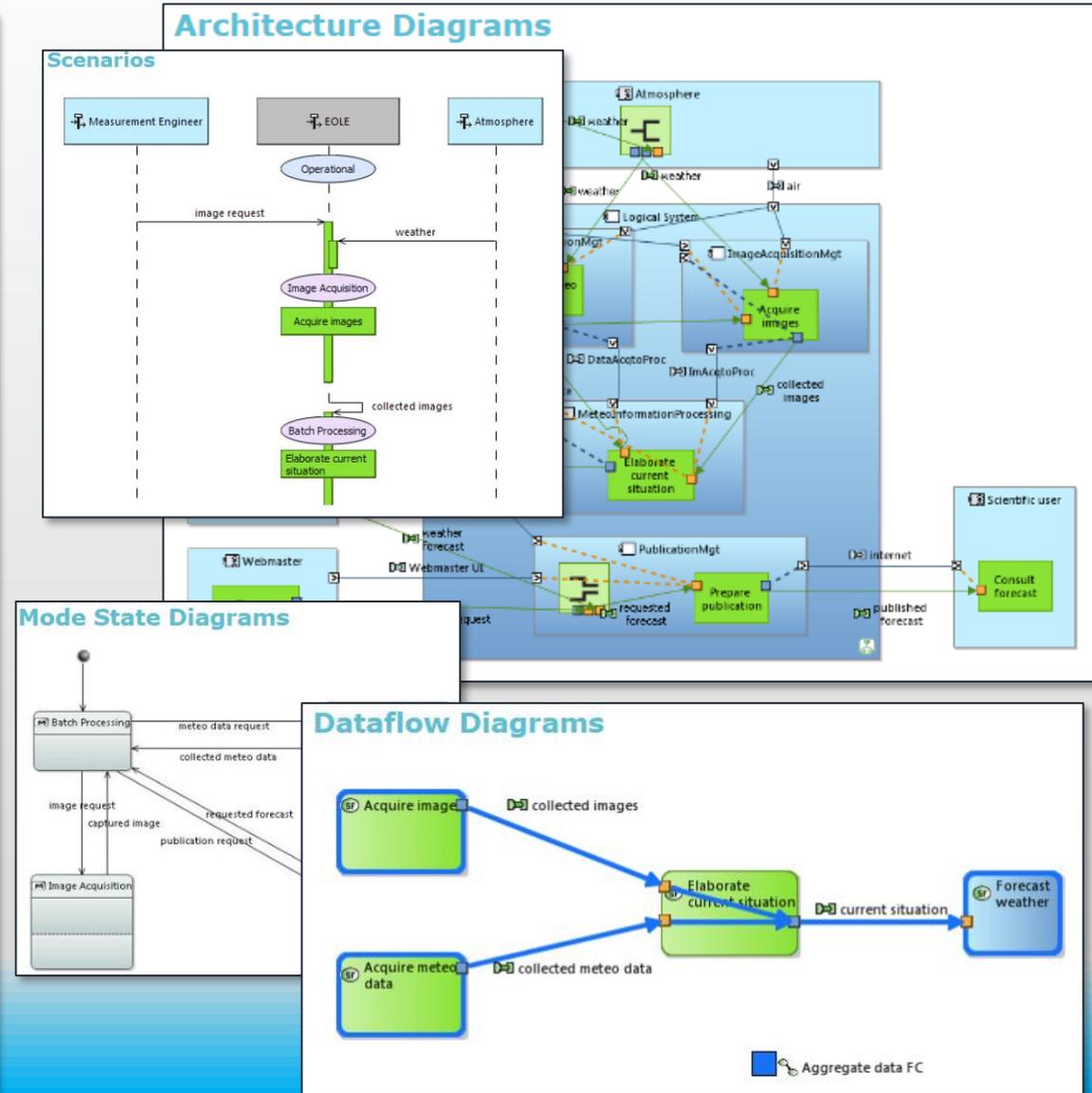
The Arcadia concepts are mostly similar to the UML/SysML standard and the NATO Architecture Framework (NAF) standard. Because of the focus on architectural design, some of the SysML concepts have been simplified or specialized in order to better match the concepts system engineering practitioners already use in their engineering documents and assets.

## Diagrams

Arcadia method is supported by various kinds of diagrams largely inspired by UML and SysML:

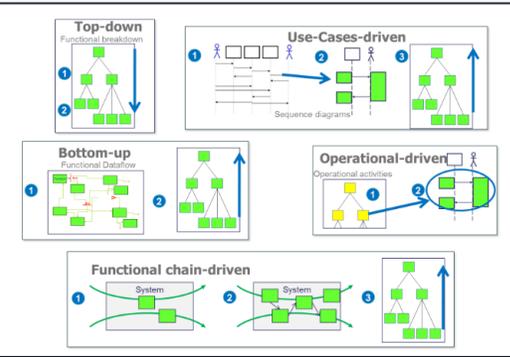
- Architecture diagrams;
- Dataflows diagrams;
- Functional chains diagrams;
- Sequence diagrams;
- Tree diagrams;
- Mode and States diagrams;
- Classes and Interfaces diagrams.

Diagram	Description
Breakdown diagram	Stakeholders/Functions/Components decomposition through graphical tree
Capability diagram	Equivalent to a use-case diagram, used to organize the functional analysis
Dataflow diagram	Provide informations exchanged between functions
Architecture diagram	Described the assembly of components or functions and interfaces
Scenario	Provides dynamic behavior between functions
Mode&State	Provide the working type of function or actor or system.
Class diagram	Often, data-class diagram compresses exchange items or data parameters utilized in a system

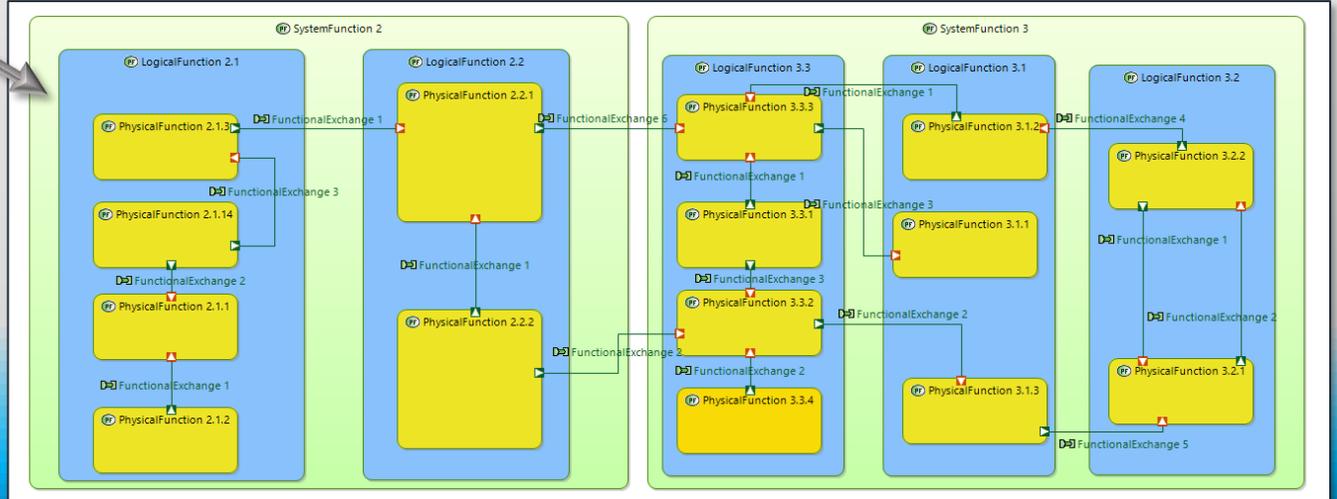
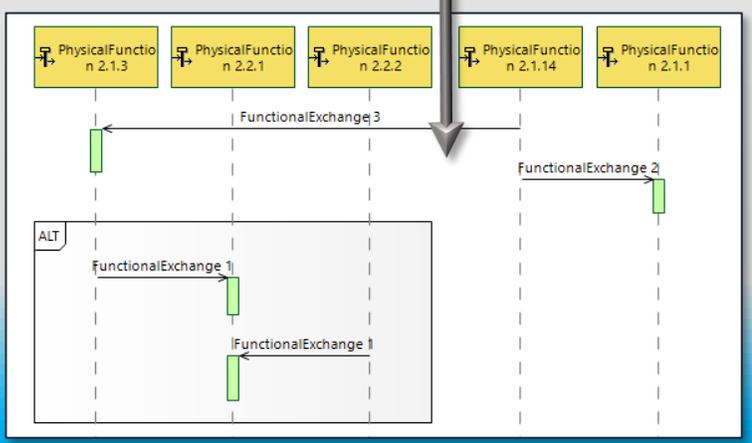
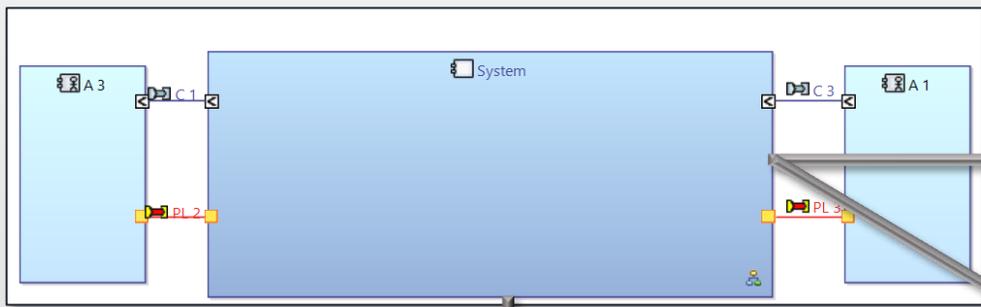
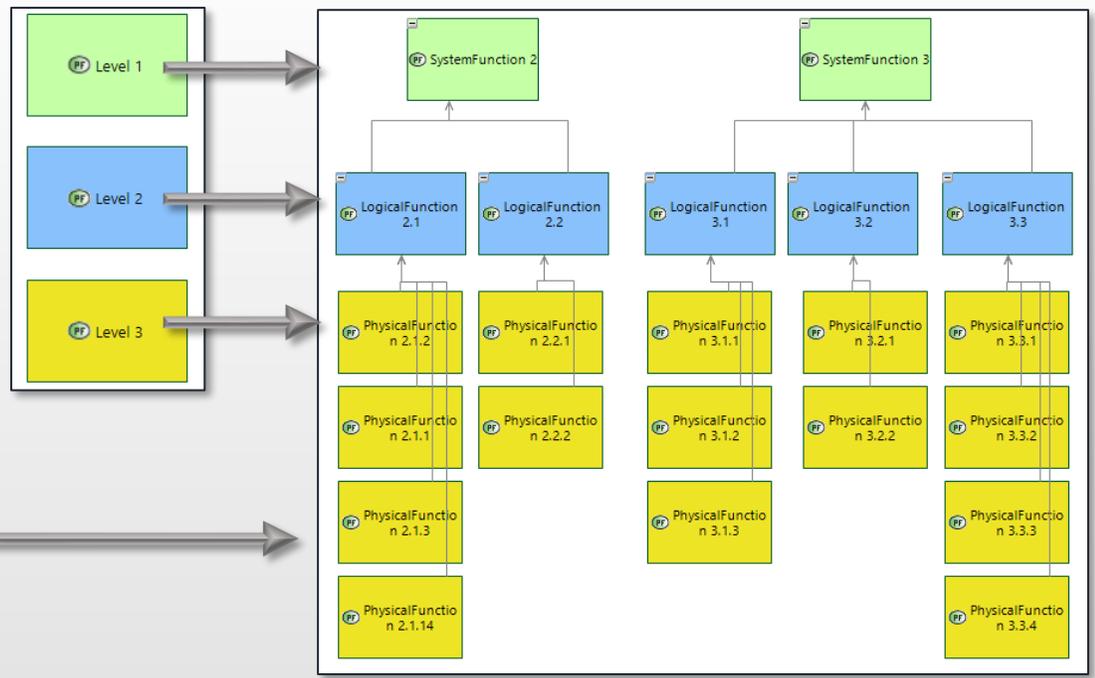


# ARCADIA/CAPELLA – Functional analysis

Speaker: DROUIN Rémy



- Define the system
- Define the functions
- Define the tree diagram
- Define the functional dependencies

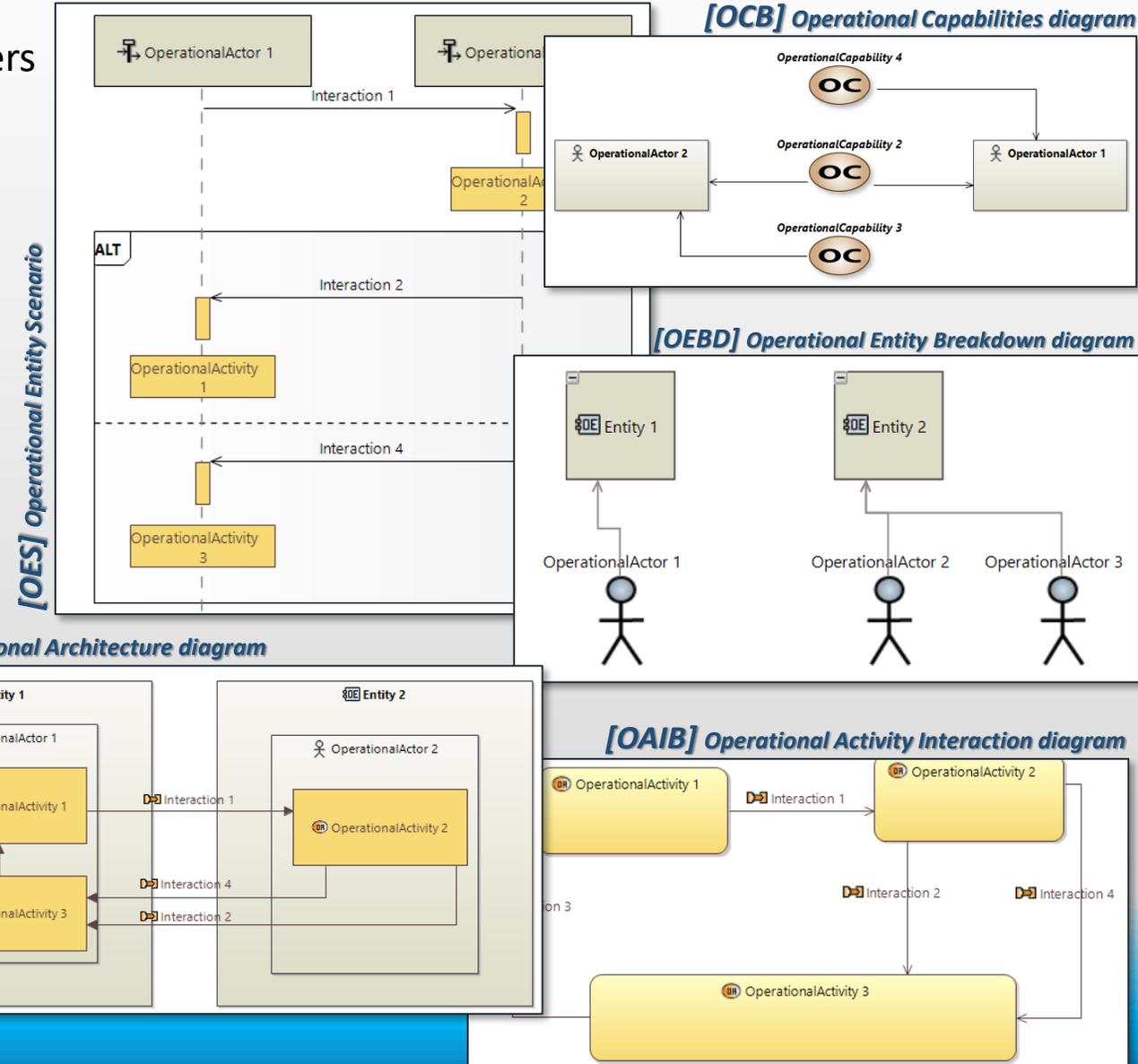
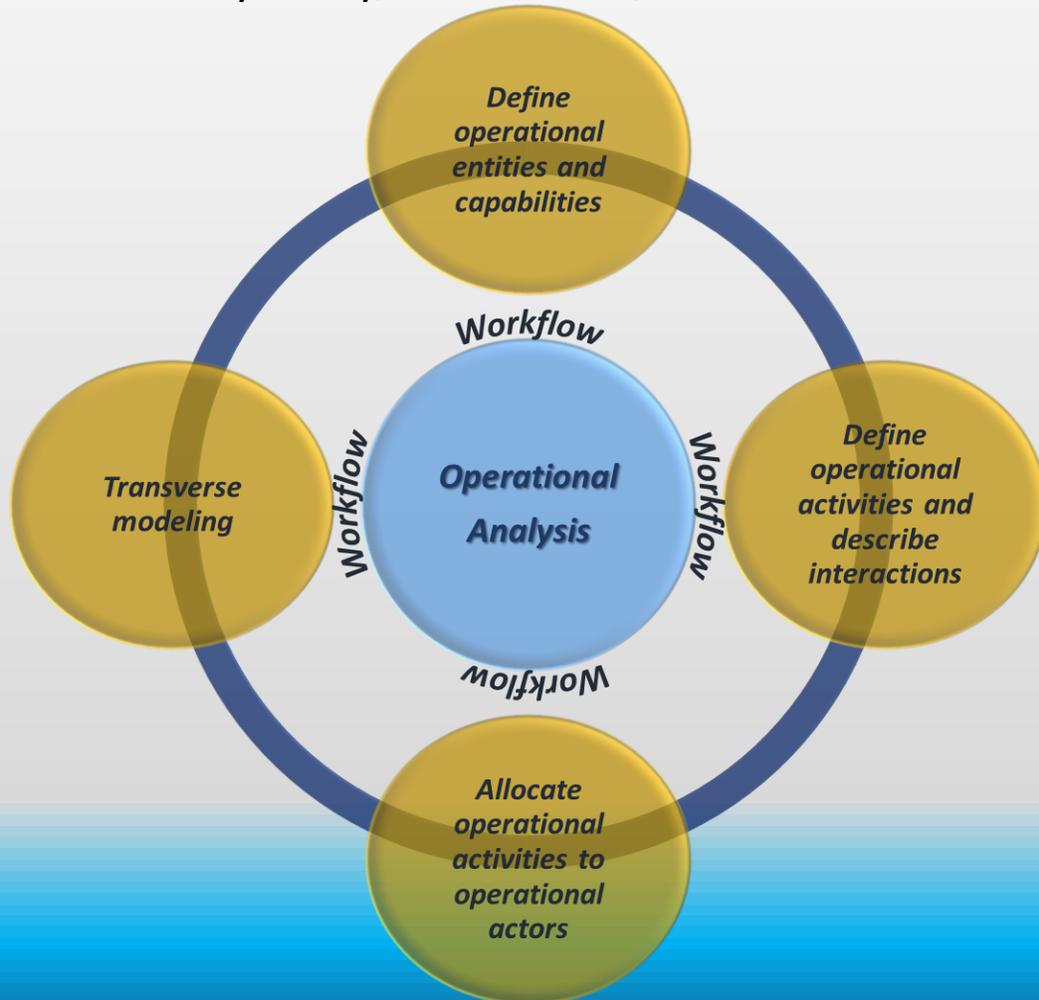


# ARCADIA/CAPELLA - Operational Analysis

Speaker: DROUIN Remy

## Define Stakeholder needs and environment

- Capture and consolidate operational needs from stakeholders
- Define what the users have to accomplish
- Identify entity, actors, roles, activities and concepts

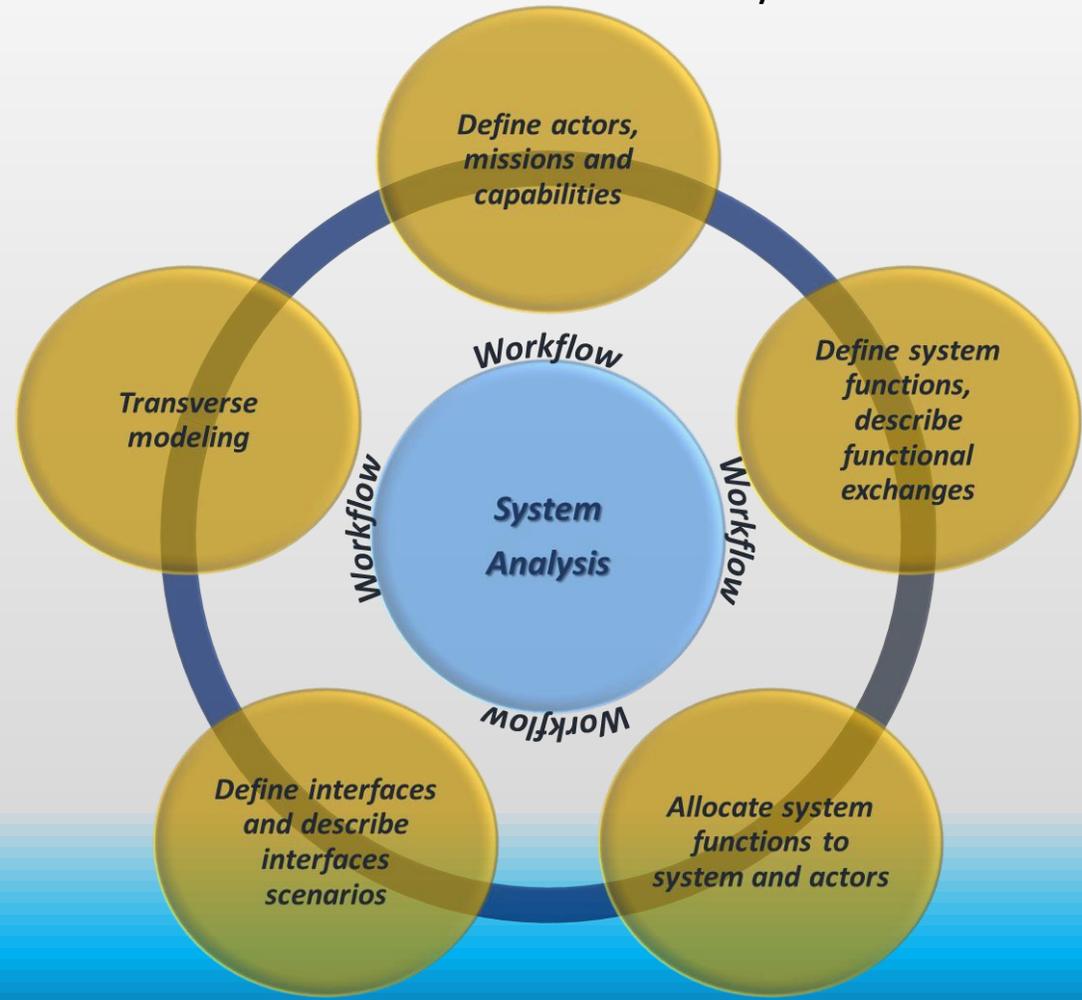


# ARCADIA/CAPELLA - System Analysis

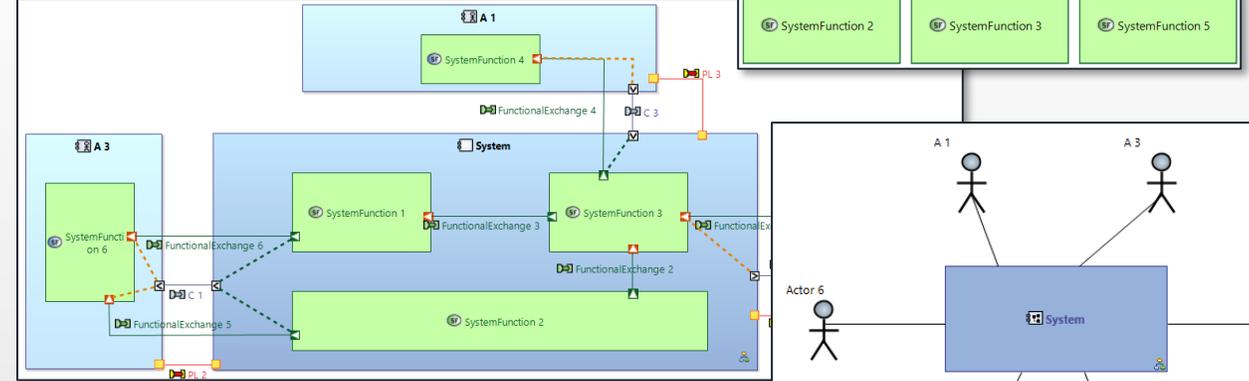
Speaker: DROUIN Remy

## Formalize system requirements

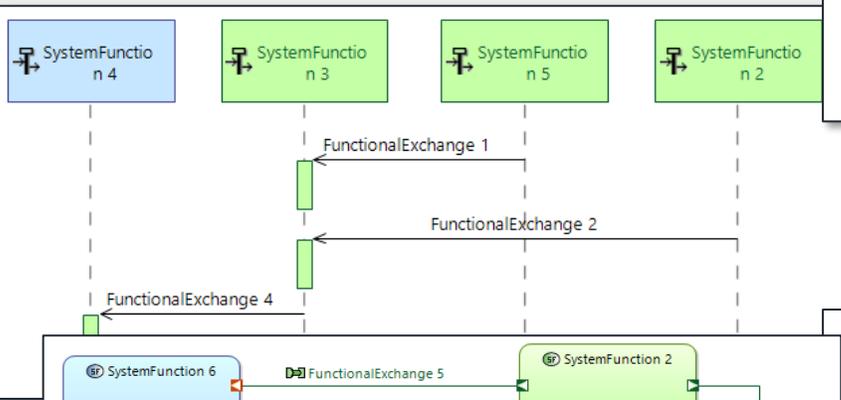
- Identify the boundary of the system
- Define what the system has to accomplish for the users
- Model functional dataflows and dynamic behaviour



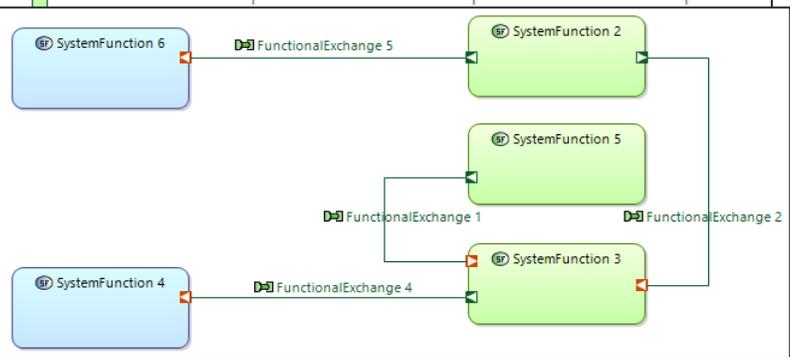
[SAB] System Architecture diagram



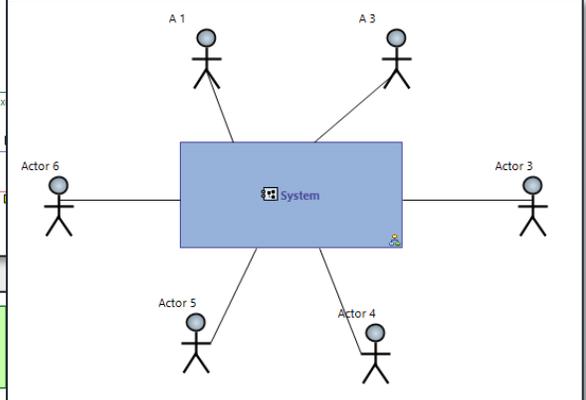
[FS] Functional Scenario



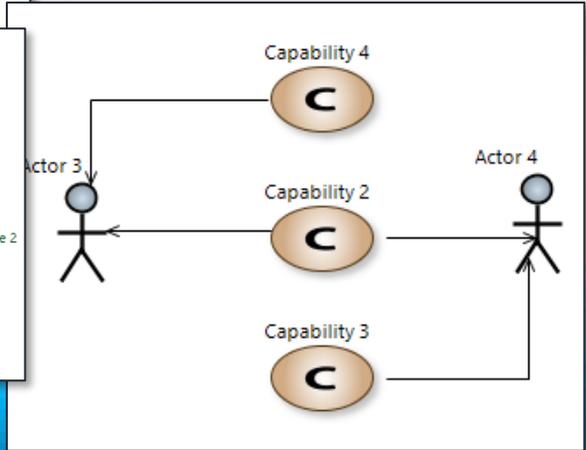
[SDFB] Functional Dataflow Blank diagram



[CSA] Contextual System Actors diagram



[MCB] Mission Capabilities Blank diagram

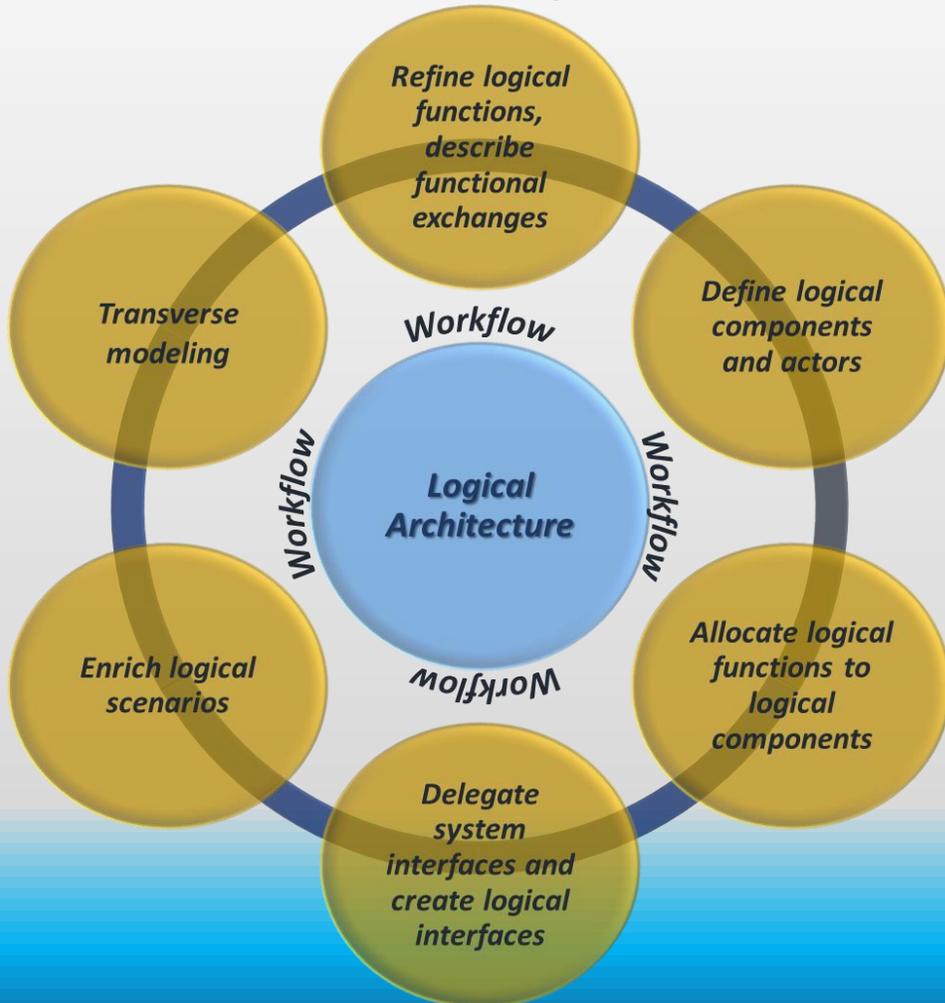


# ARCADIA/CAPELLA - Logical Architecture

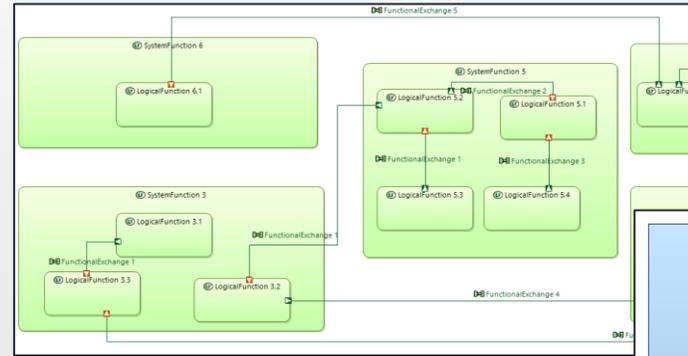
Speaker: DROUIN Remy

## Developp system logical architecture

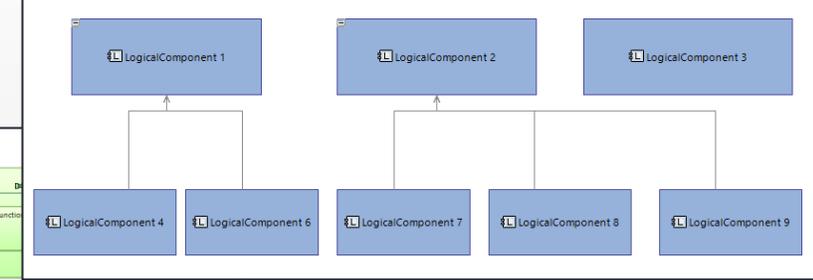
- See the system as a white box define how the system will work so as to fulfill expectations
- Perform a trade-off analysis



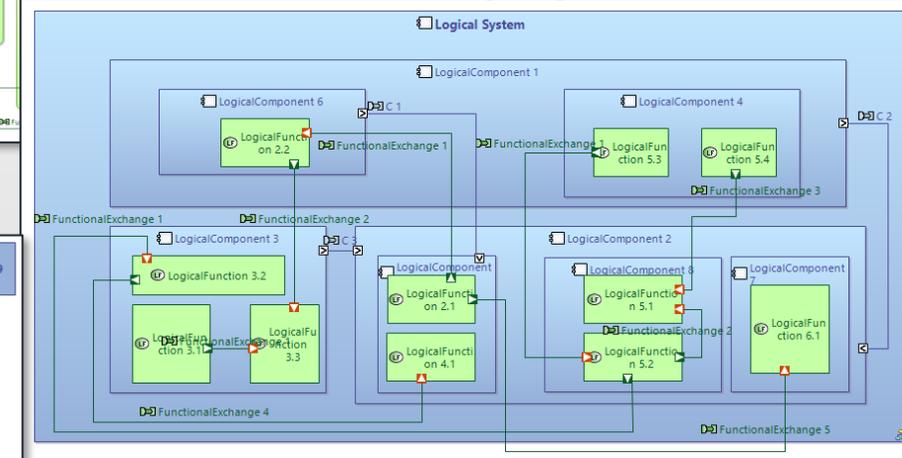
[LDFB] Functional Dataflow Blank diagram



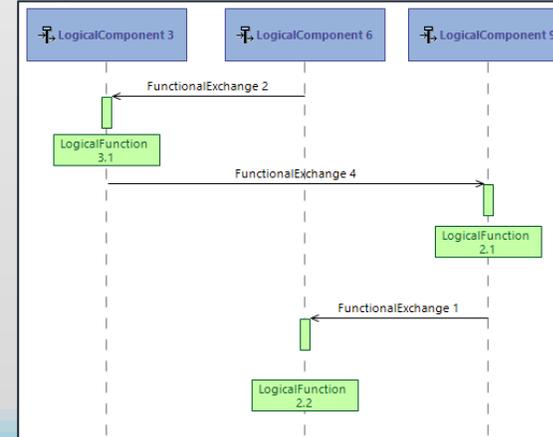
[LCBD] Logical Component Breakdown diagram



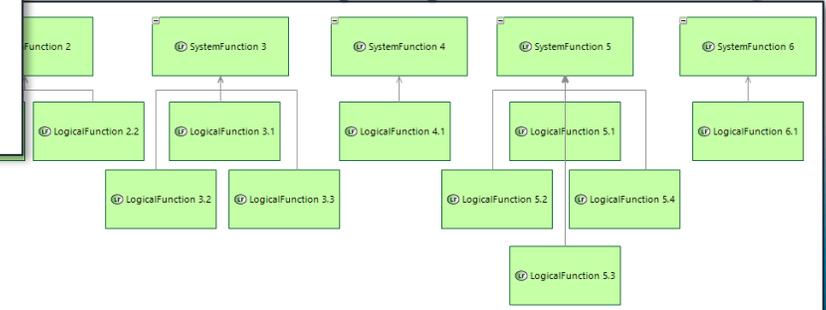
[LAB] Logical Architecture diagram



[ES] Exchange Scenario

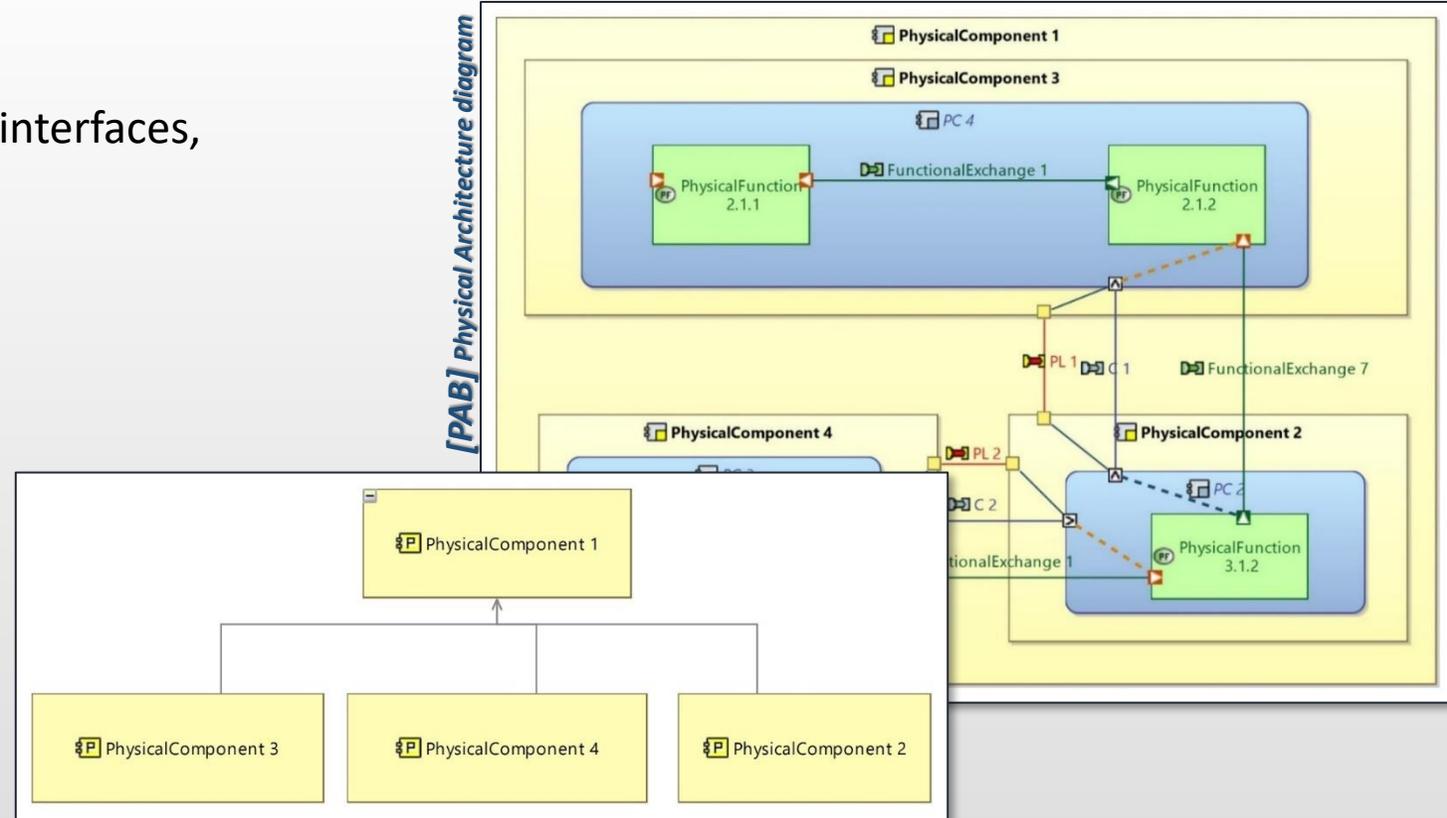
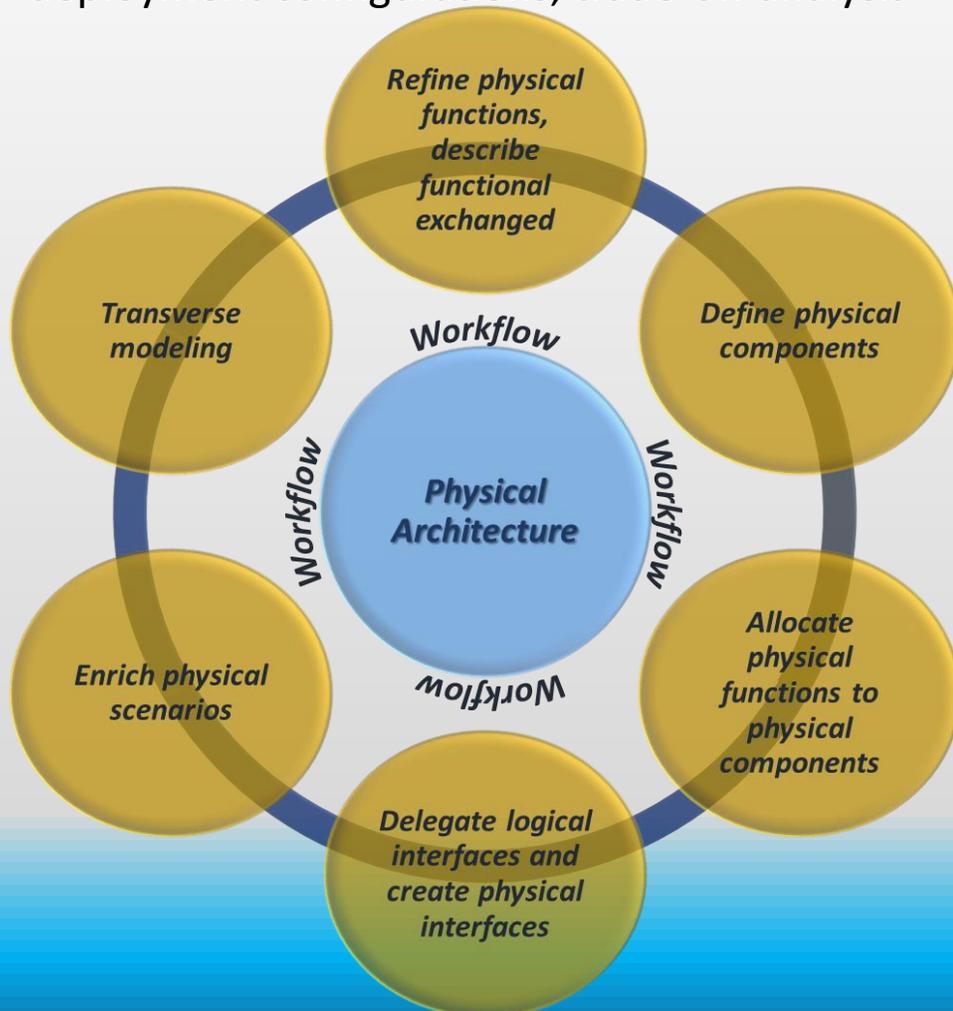


[LFBBD] Functional Breakdown diagram



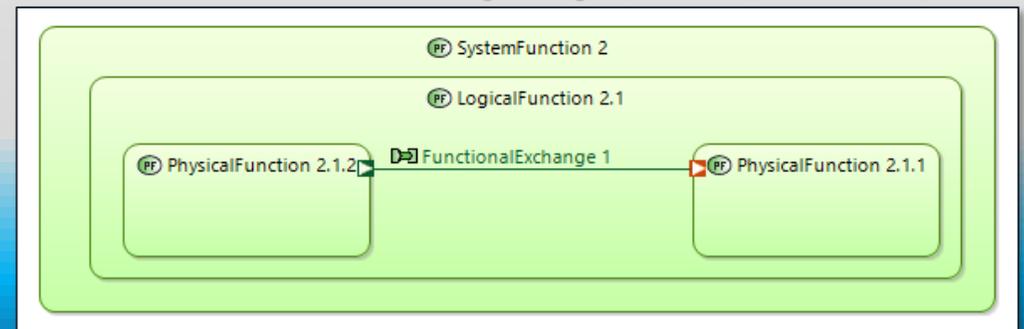
## Develop system Physical architecture

- How the system will be developed and built
- Software vs. hardware allocation, specification of interfaces,
- deployment configurations, trade-off analysis



[PCBD] Physical Component Breakdown diagram

[PDFB] Physical Dataflow Blank diagram



# ARCADIA/CAPELLA - Add ons sample

Speaker: DROUIN Remy

Several add ons are available in order to unleash the power of MBSE workbench

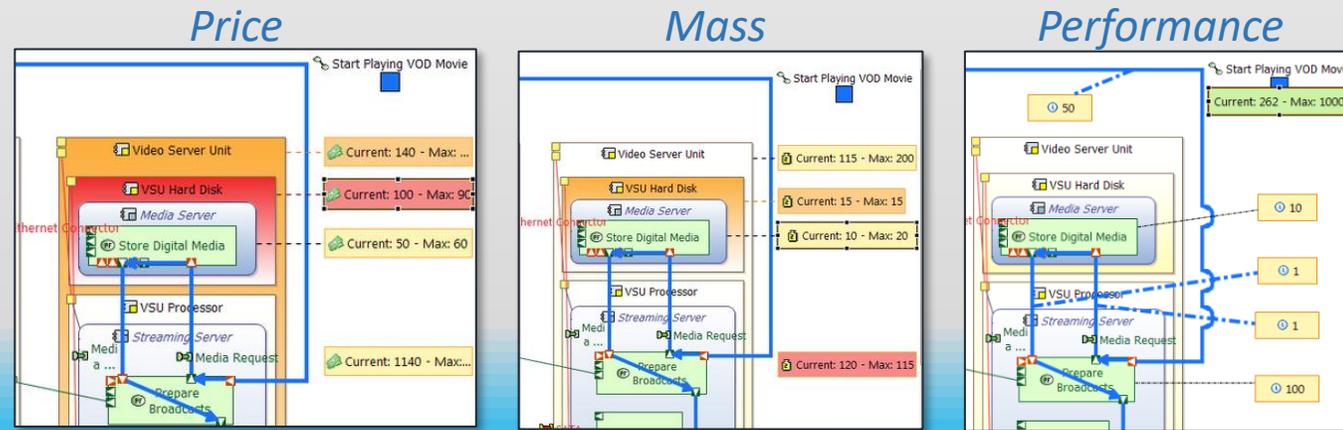
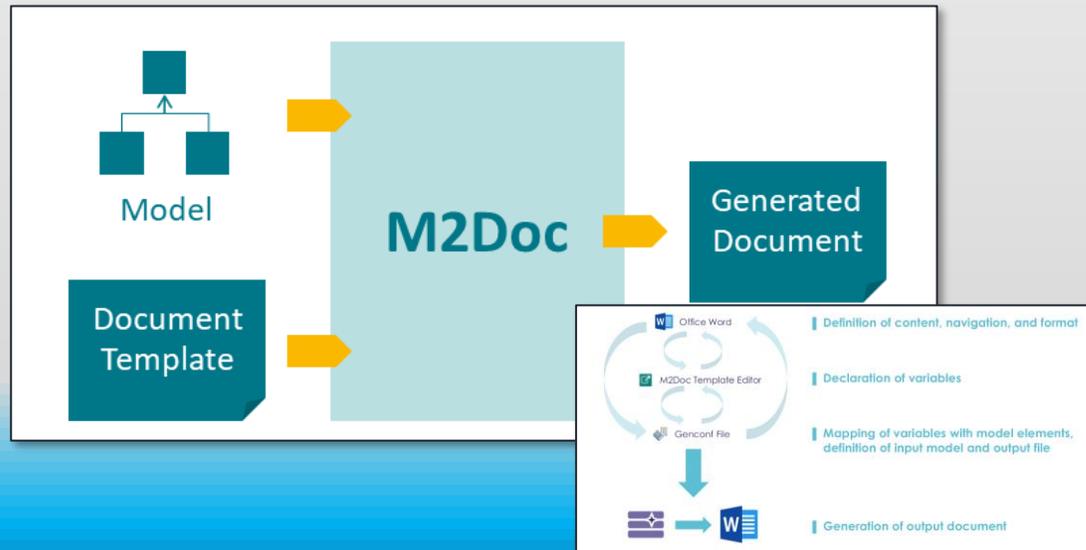
## Open-Source Add ons

Capella	Subsystem-Transition	Requirements	Filtering	Basic-Viewpoints	xhtml-docgen	M2Doc
5.0.x						
1.4.x						
1.3.x	(releases)	(releases)	(releases)	(releases)	(releases)	(releases)
1.2.x						
1.1.x						
1.0.x						
0.8.x						

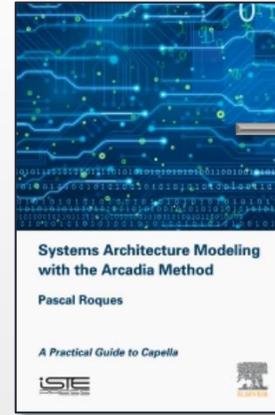
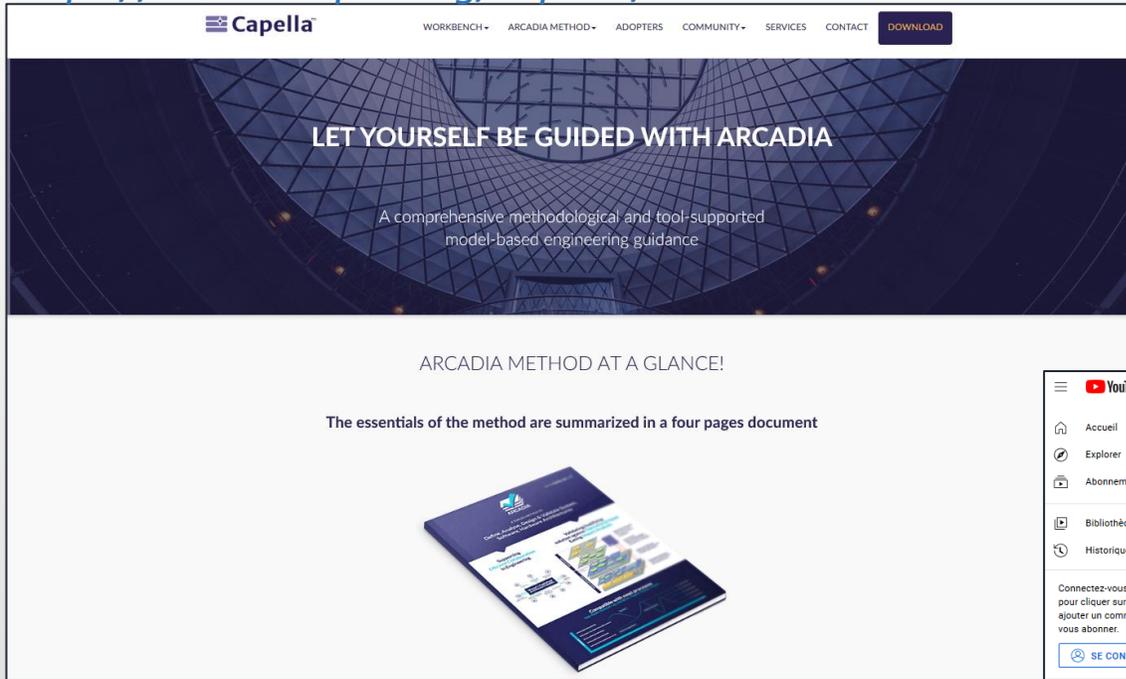
The **Mass viewpoint** enables to simply describe the non-functional aspect of mass in Capella.  
 The **Price viewpoint** enables to simply describe the non-functional aspect of price in Capella.  
 The **Performance viewpoint** enables to simply describe the non-functional aspect of performance in Capella.

**Xhtml docgen** addon enables the end-user to generate an HTML website from a Capella project.  
 Sharing models with all stakeholders is essential in model-based systems engineering. Publishing and sharing HTML versions of models helps make models The reference of all engineering activities.

**M2Doc** generates MS-Word documents from Capella models. M2Doc uses customizable Word templates to extract data and diagrams from your models and display them in a docx file.

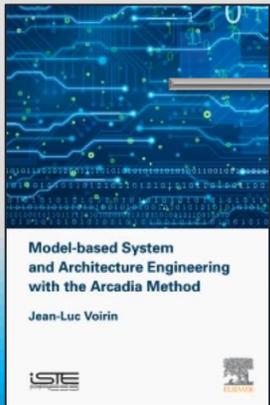


<https://www.eclipse.org/capella/arcadia.html>

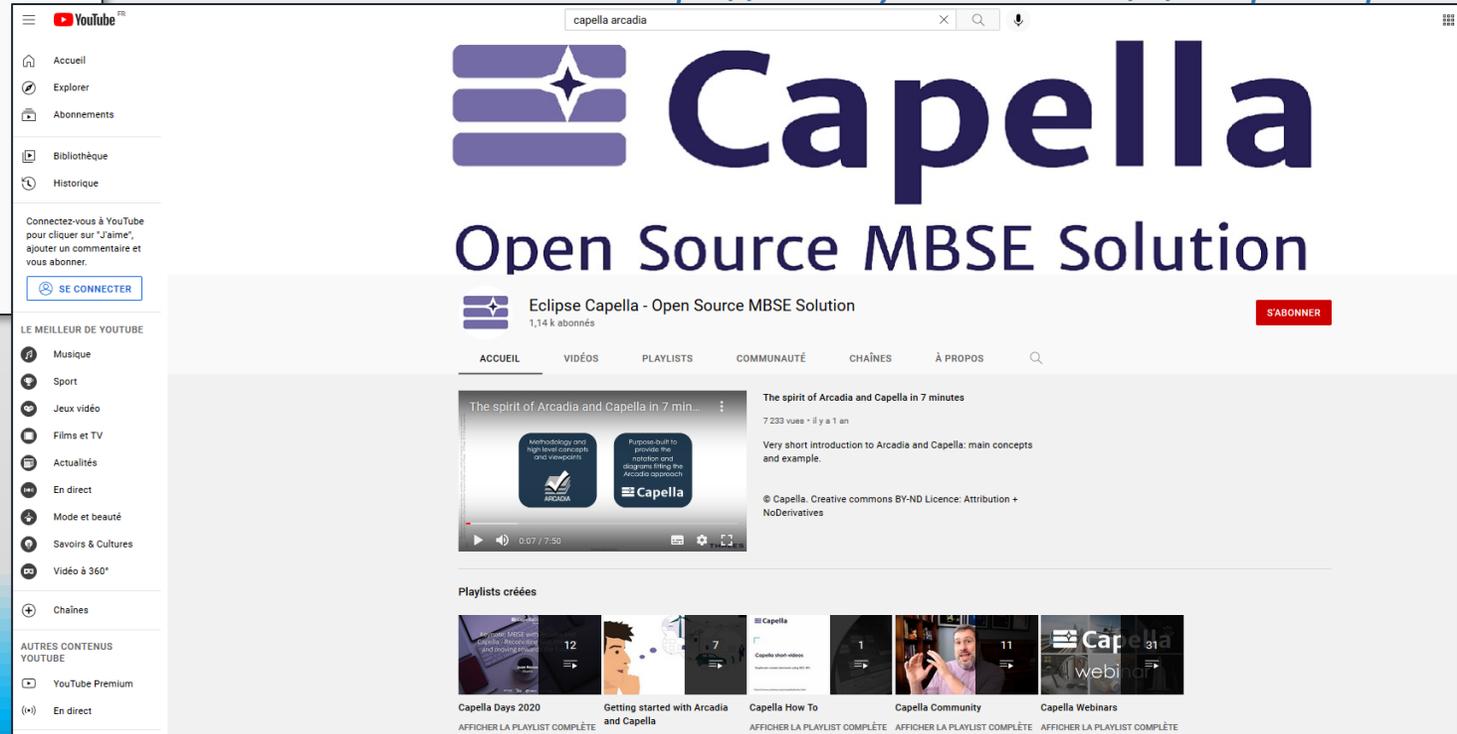


“Systems Architecture Modeling with the ARCADIA Method”  
@Pascal Roques

<https://www.youtube.com/c/EclipseCapella>



“Model-Based Systems and Architecture Engineering with the ARCADIA Method”  
@Jean Luc Voirin



## @HELP Contents [CAPELLA modeler]

Help - Capella

Search:  Go [Scope](#): All topics

**Contents**

- Workbench User Guide
- Java development user guide
- Platform Plug-in Developer Guide
- JDT Plug-in Developer Guide
- Plug-in Development Environment Guide
- Acceleo Query Language Documentation
- Capella Guide
  - Release Note
  - User Manual
  - Developer Manual
  - Diagrams**
    - Common Diagram Tools
    - Activity Interaction Scenario
    - Capability Realization Blank
    - Capability Realization Refinement
    - Capella Architecture
    - Class Diagram Blank
    - Component Exchanges Scenario
    - Component Interfaces Scenario
    - Configuration Items Breakdown
    - Contextual Capability
    - Contextual Capability Realization Involvement
    - Contextual Component Detailed Interfaces
    - Contextual Component External Interfaces
    - Contextual Component Internal Interfaces
    - Contextual Mission
    - Contextual Operational Capability
    - Contextual System Actors
    - EPBS Architecture Blank
    - Functional Chain Description
    - Functional Scenario
    - Interface Diagram
    - Interfaces Diagram Blank
    - Logical Architecture Blank
    - Logical Component Breakdown
    - Logical Data Flow Blank
    - Logical Function Breakdown
    - Missions Blank
    - Missions Capabilities Blank
    - Modes & States [DEPRECATED]
    - Mode State Machine
    - Operational Activity Breakdown
    - Operational Activity Interaction Blank
    - Operational Capabilities Blank
    - Operational Entity Blank
    - Operational Entity Breakdown
    - Operational Interaction Scenario
    - Operational Process Description

**Diagrams**

Contents

- [Common Diagram Tools](#)
- [Activity Interaction Scenario](#)
- [Capability Realization Blank](#)
- [Capability Realization Refinement](#)
- [Capella Architecture](#)
- [Class Diagram Blank](#)
- [Component Exchanges Scenario](#)
- [Component Interfaces Scenario](#)
- [Configuration Items Breakdown](#)
- [Contextual Capability](#)
- [Contextual Capability Realization Involvement](#)
- [Contextual Component Detailed Interfaces](#)
- [Contextual Component External Interfaces](#)
- [Contextual Component Internal Interfaces](#)
- [Contextual Mission](#)
- [Contextual Operational Capability](#)
- [Contextual System Actors](#)
- [EPBS Architecture Blank](#)
- [Functional Chain Description](#)
- [Functional Scenario](#)
- [Interface Diagram](#)
- [Interfaces Diagram Blank](#)
- [Logical Architecture Blank](#)
- [Logical Component Breakdown](#)
- [Logical Data Flow Blank](#)
- [Logical Function Breakdown](#)
- [Missions Blank](#)
- [Missions Capabilities Blank](#)
- [Modes & States \[DEPRECATED\]](#)
- [Mode State Machine](#)
- [Operational Activity Breakdown](#)
- [Operational Activity Interaction Blank](#)
- [Operational Capabilities Blank](#)
- [Operational Entity Blank](#)
- [Operational Entity Breakdown](#)
- [Operational Interaction Scenario](#)
- [Operational Process Description](#)

<https://forum.mbse-capella.org/>

**Capella** [Sign Up](#) [Log In](#)

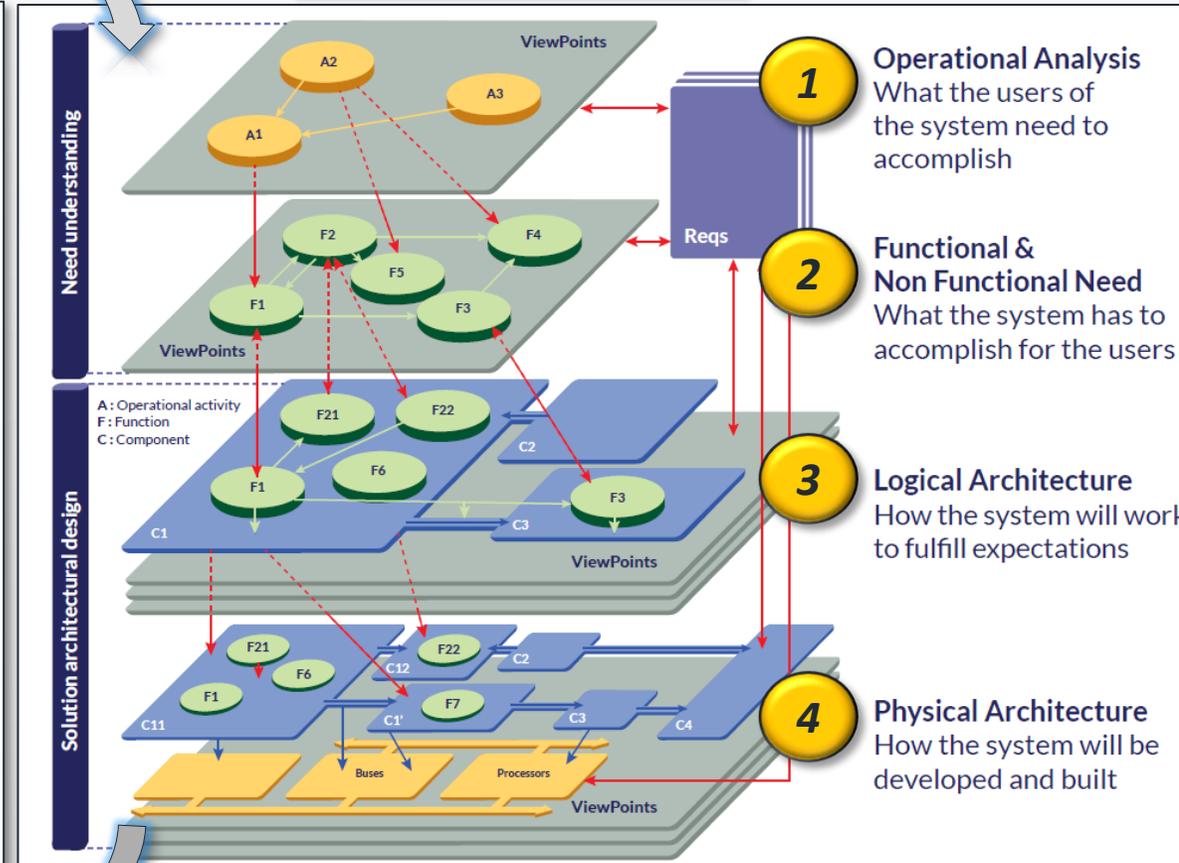
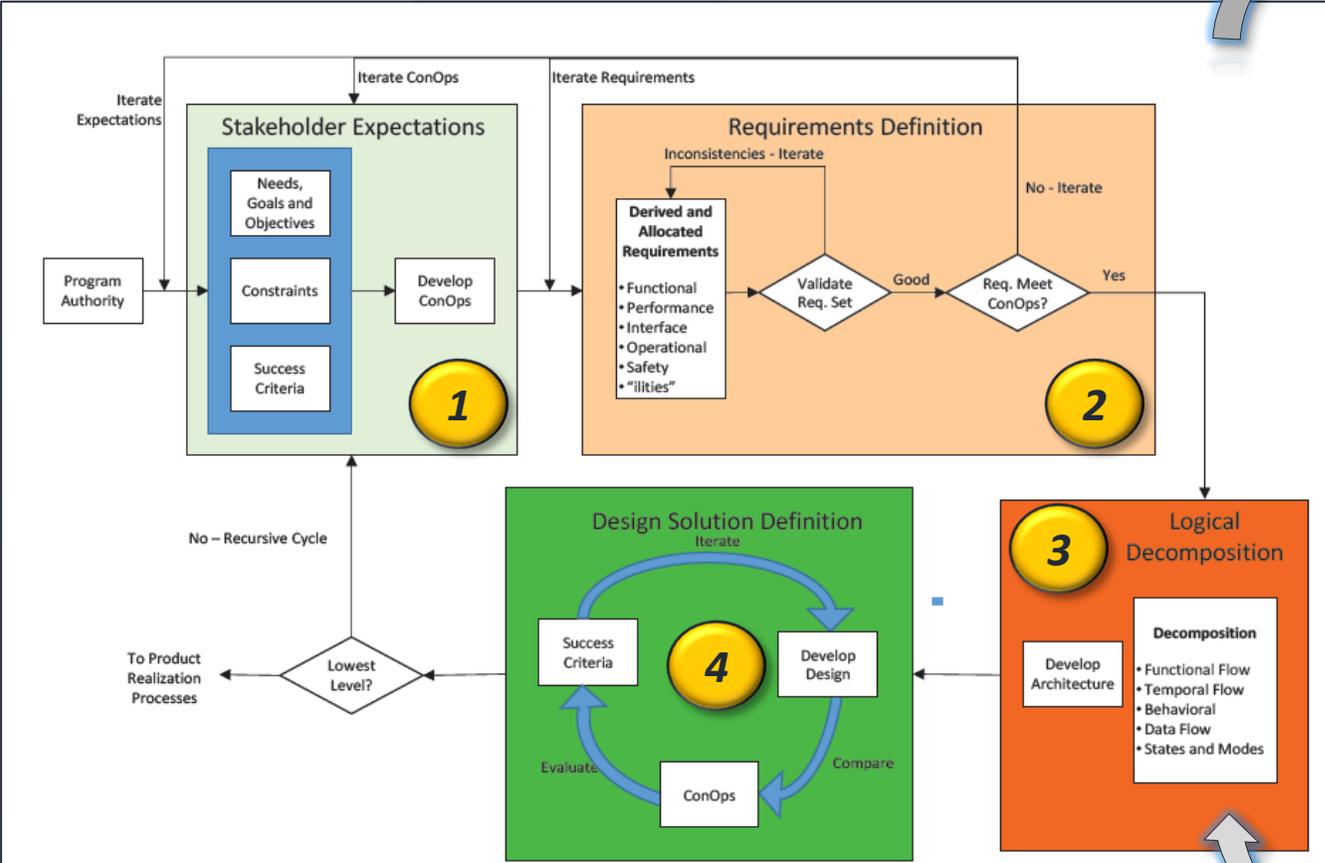
Welcome to the [Eclipse Capella](#) community forum.

[all categories](#) **Categories** [Latest](#) [Top](#)

Category	Topics	Latest
<b>News</b> Latest news about Capella	159	 REC / RPL and parts naming ■ Capella 0 1h
<b>Arcadia</b> The Arcadia model-based engineering method	3 / month	 Functional Chain EDIT ■ Capella 5 17h
<b>Capella</b> Any tool-related question, suggestions, problem, etc	15 / month	 Requirement VP: Extract Requirement Allocation? ■ Add-ons 1 19h
<b>Capella Studio &amp; Kitalpha</b>	167	 Capella 5 Issue with behavior PC ■ Capella 3 3d
<b>Document Generation</b> Doc generation with M2Doc or other Add-ons.	1 / month	 EMF Diff/Merge : same match keys ■ Capella 2 4d
<b>Add-ons</b> Any question related to other Capella Add-ons	4 / month	 Open Capella internal link from external browser? ■ Add-ons 0 4d

# ARCADIA Versus NASA System Engineering Handbook

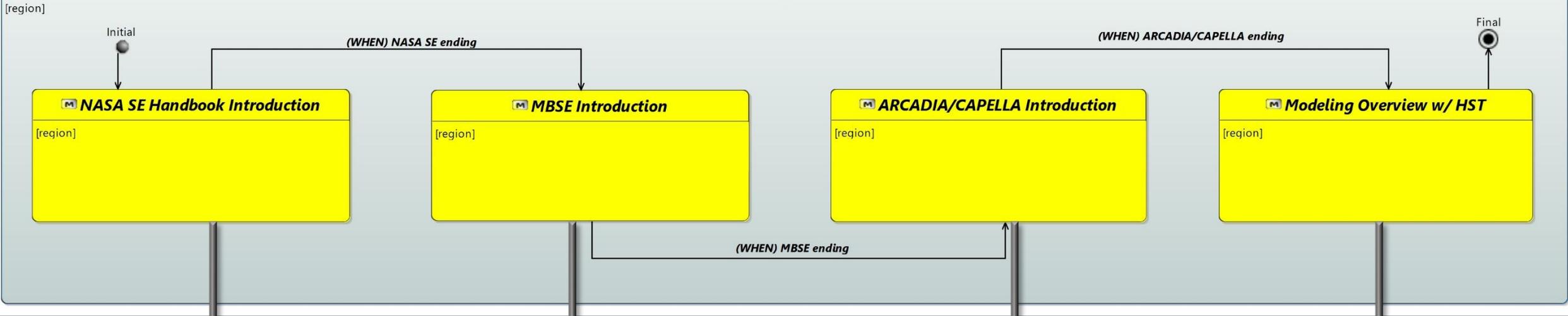
Speaker: DROUIN Remy



Requirements definition process  
Technical solution definition process

Need understanding  
Solution architectural design

## TALK Life Cycle



**NASA Systems Engineering Handbook** Speaker: DROUIN Remy

NASA Systems Engineering Handbook is intended to provide general guidance and information on systems engineering that will be useful to the NASA community. It provides a generic description of Systems Engineering (SE) as it should be applied throughout NASA. This handbook describes systems engineering best practices that should be incorporated in the development and implementation of large and small NASA programs and projects.

**Expanded Guidance for NASA Systems Engineering** Speaker: DROUIN Remy

...end to use the product.

...med by Functional IDs. They are ion oriented

...representing an action

...is developed using its between each

> Functional analysis looks across all life cycle processes

**MBSE Introduction** Speaker: DROUIN Remy

**Model-based systems engineering (MBSE)** is a systems engineering methodology that focuses on creating and exploiting domain models as the primary means of information exchange between engineers, rather than on document-based information exchange. MBSE is a formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.

**Document centric** vs **Model centric**

**Life Cycle activity**

Life Cycle activity	MBSE Contribution
1 Stakeholders expectations definition	Need, goals and objectives are kept within the model and form the top tier of eventual requirements
2 Technical requirements	Requirements are kept within the models
3 Logical decomposition	Requirements can be categorized into functional, behavioral, performance etc... These can be used to develop functional block and behavior diagrams
4 Design Solution Definition	Allows integration of information and designs from different engineering domains supporting the high source of truth

**Modeling objectives (example)**

- Characterize an existing system
- Specify and design a new or modified system
- Evaluate a system
- Train users on how to operate or maintain a system

**ARCADIA/CAPELLA - Introduction** Speaker: DROUIN Remy

**Architecture Analysis & Design Integrated Approach (ARCADIA)** is a structured architecture engineering method for defining and validating multi-domain systems, based on architecture-centric and model-driven engineering activities. ARCADIA is a method based on functional analysis and focuses on developing the system by starting from needs analysis and solution development up to integrated verification and validation.

- Ensure engineering-wide collaboration by sharing the same reference architecture
- Master the complexity of systems and architectures
- Define the best optimal architectures through trade-off analysis
- Master different engineering levels and traceability with automated transition and information refinement

**Capella**

Logical Architecture

Physical Architecture

Behavioral Architecture

Functional Architecture

**HUBBLE Space Telescope - Introduction** Speaker: DROUIN Remy

**The goal is not to modelize the whole system, but just to apply few diagrams showing the application of CAPELLA**

Hubble is a Cassegrain reflector telescope. Light from celestial objects travels down a tube, is collected by a bowl-like, inwardly curved primary mirror and reflected toward a smaller, dome-shaped, outwardly curved secondary mirror. The secondary mirror bounces the light back to the primary mirror and through a hole in its center. The light is focused on a small area called the focal plane, where it is picked up by its various science instruments.

Orbiting high above the Earth, the Hubble Space Telescope has a clear view of the universe free from the blurring and absorbing effects of the atmosphere. In addition to observing visible and near-infrared light, Hubble detects ultraviolet light, which is absorbed by the atmosphere and visible only from space. The telescope has beamed hundreds of thousands of celestial images back to Earth during its time in space.

**The goal is not to apply the MBSE approach to the entire system, but just to apply few diagrams showing the application of ARCADIA/CAPELLA**

« All models are wrong, but some are useful » George E.P.Box (British statistician)

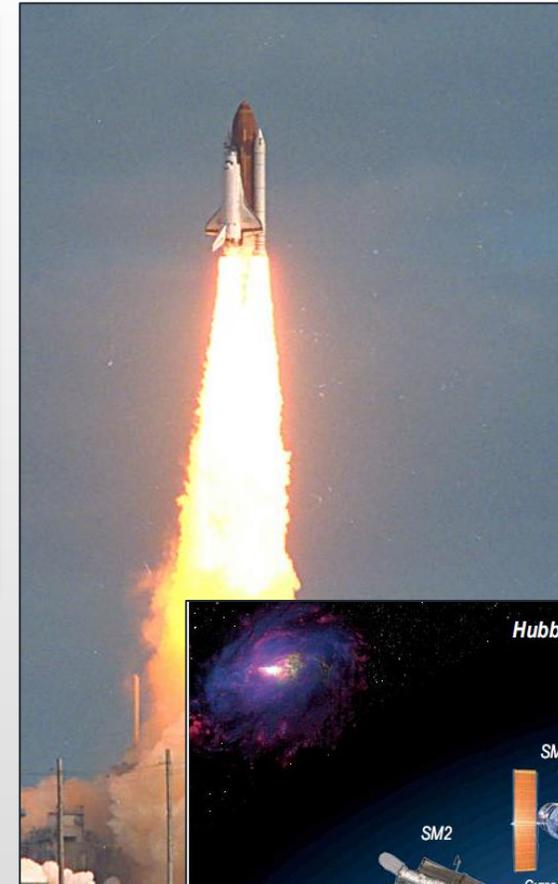
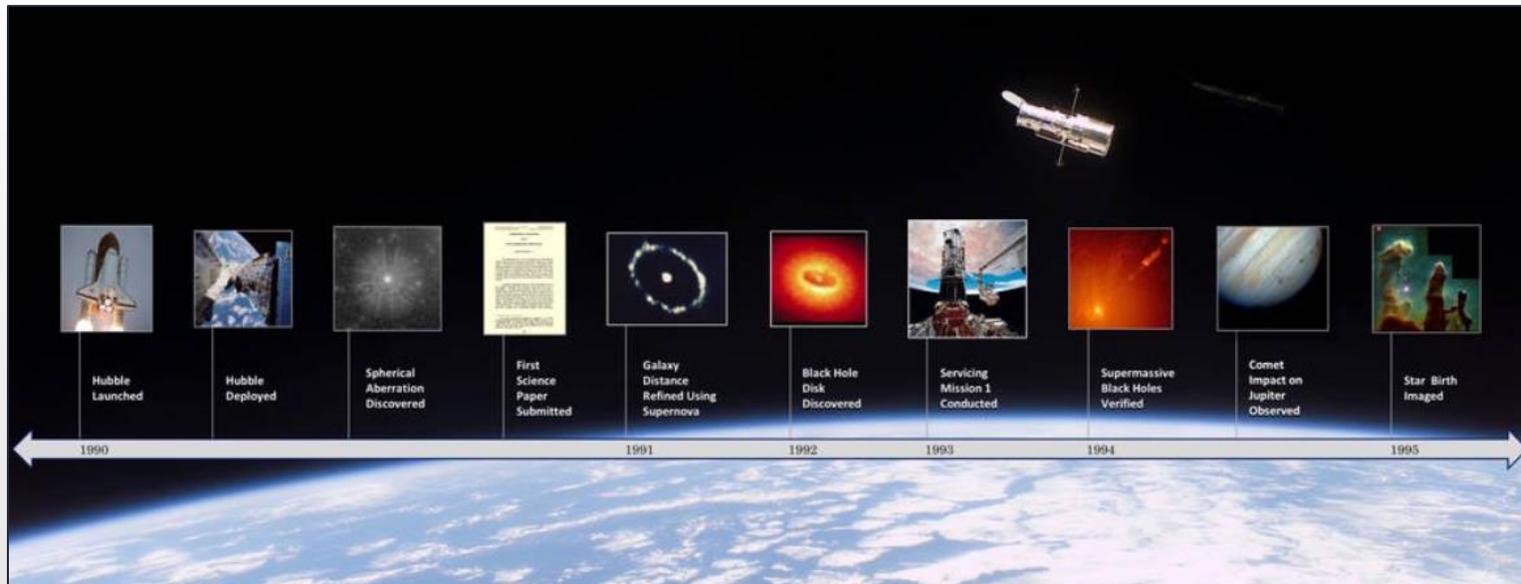
Hubble is a Cassegrain reflector telescope. Light from celestial objects travels down a tube, is collected by a bowl-like, inwardly curved primary mirror and reflected toward a smaller, dome-shaped, outwardly curved secondary mirror. The secondary mirror bounces the light back to the primary mirror and through a hole in its center. The light is focused on a small area called the focal plane, where it is picked up by its various science instruments.



Orbiting high above the Earth, the Hubble Space Telescope has a clear view of the universe free from the blurring and absorbing effects of the atmosphere. In addition to observing visible and near-infrared light, Hubble detects ultraviolet light, which is absorbed by the atmosphere and visible only from space. The telescope has beamed hundreds of thousands of celestial images back to Earth during its time in space.

# HUBBLE Space Telescope facts

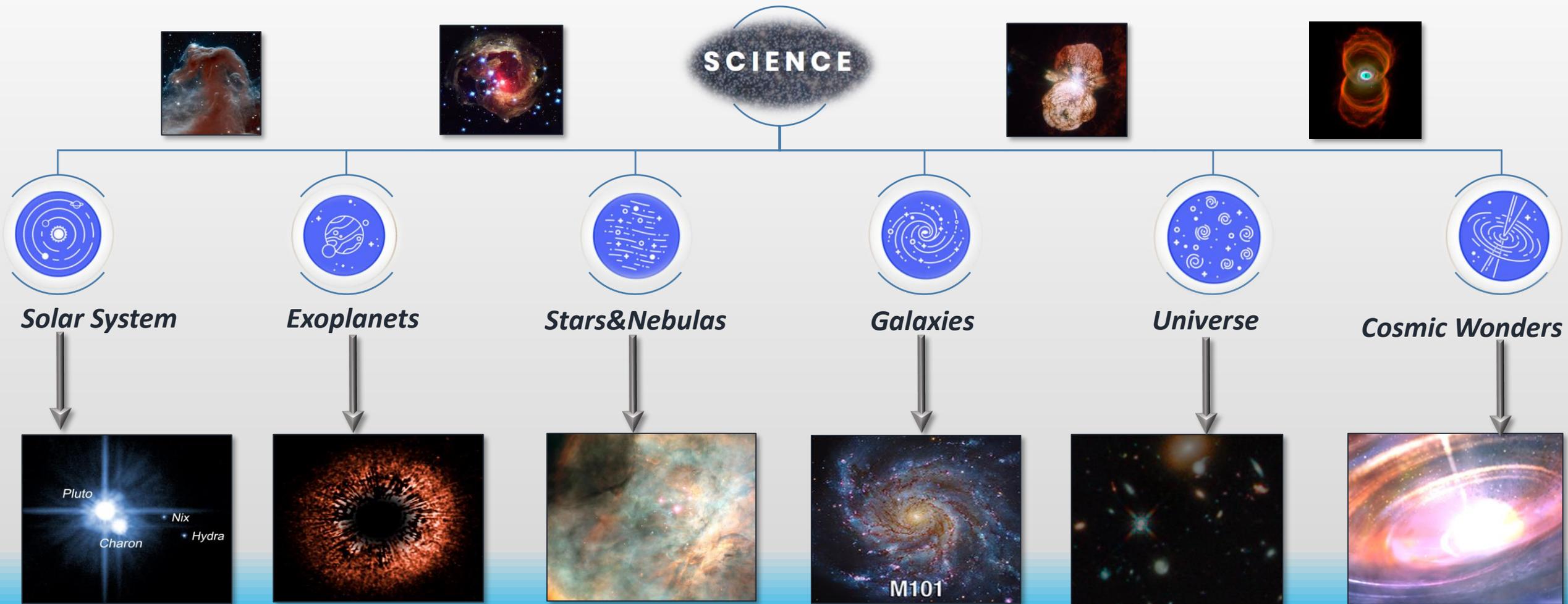
Speaker: DROUIN Remy



- **Launched:** April 24, 1990
- **Deployed:** April 25, 1990. First Image – May 20, 1990
- **Servicing missions:** 4 (SM1 – SM4)
- **Launch vehicle:** Space Shuttle Discovery (STS-31)
- **Launch site:** Kennedy Space Center, Florida
- **Location:** Orbiting 340 miles (540 kilometers) above the Earth
- **Orbital Period:** Approximately 95 minutes to complete one orbit around Earth
- **Speed:** About 17,000 mph (27,300 kph)
- **Wavelength coverage:** Sensitivity to light: Ultraviolet through Infrared (115-1700nm)



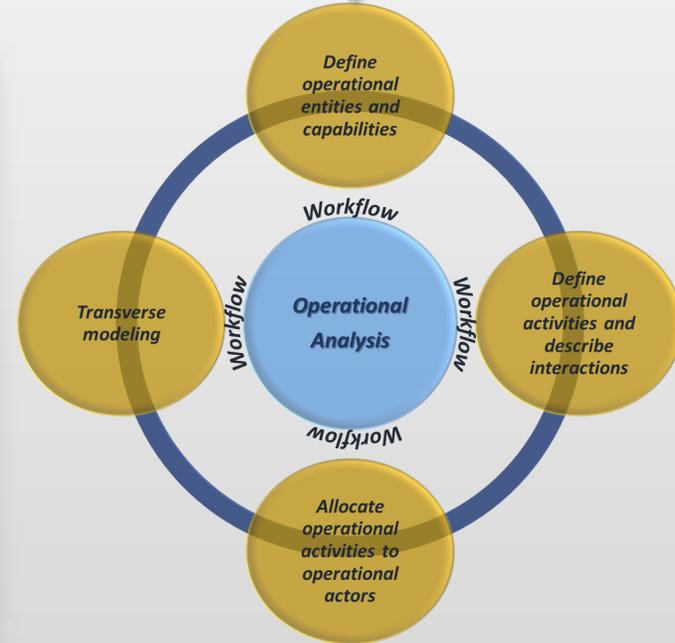
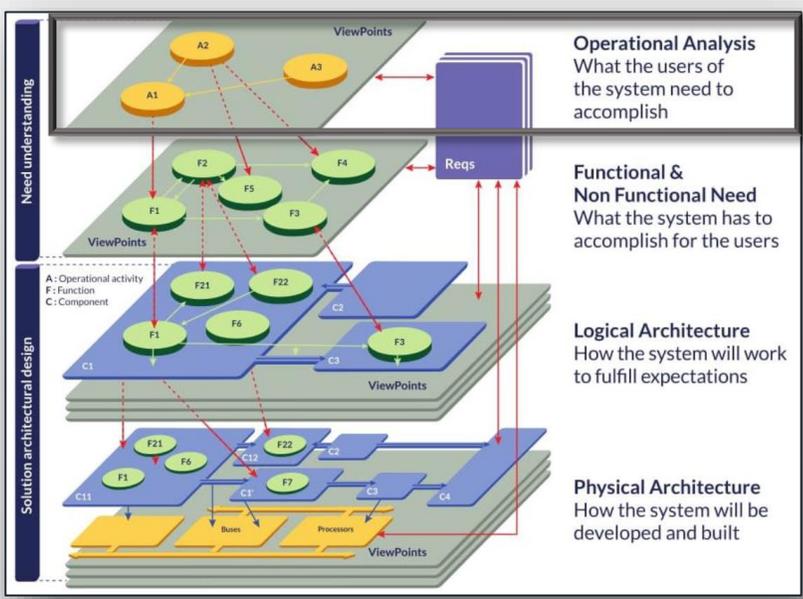
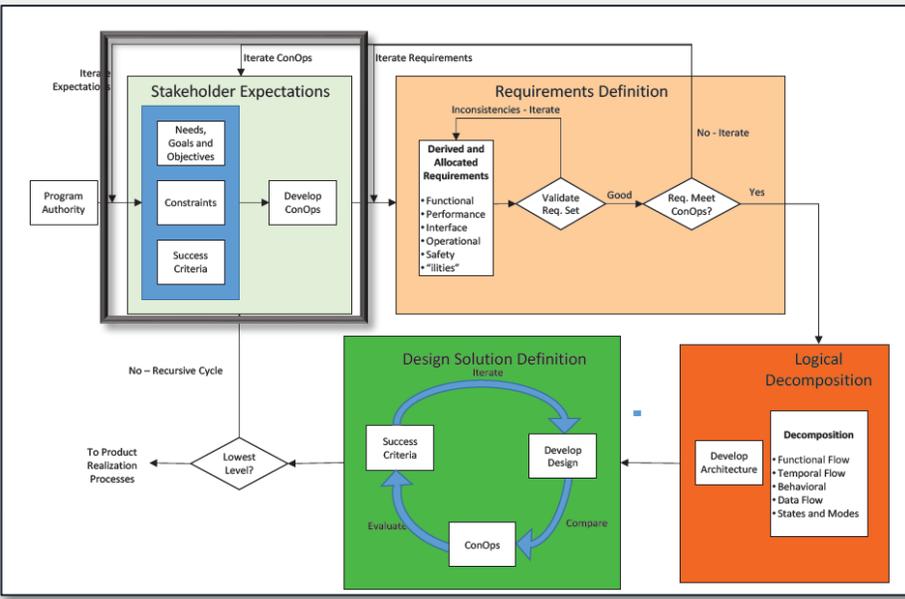
Answer some of the most compelling astronomical questions of our time, and uncovered mysteries we never knew existed



# Operational Analysis

**Requirements definition process**  
**Technical solution definition process**

**Need understanding**  
**Solution architectural design**



Activity explorer

Activity explorer

Diagram editor

Workflow of Hubble Space Telescope

- Operational Analysis** - Define Stakeholder Needs and Environment: Capture and consolidate operational needs from stakeholders. Define what the users of the system have to accomplish. Identify entities, actors, roles, activities, concepts.
- System Analysis** - Formalize System Requirements: Identify the boundary of the system, consolidate requirements. Define what the system has to accomplish for the users. Model functional dataflows and dynamic behaviour.
- Logical Architecture** - Develop System Logical Architecture: See the system as a whole box: define how the system will work so as to fulfill expectations. Perform a first trade-off analysis.
- Physical Architecture** - Develop System Physical Architecture: How the system will be developed and built. Software vs hardware allocation, specification of interfaces, deployment, configurations, trade-off analysis.
- EPBS** - Formalize Component Requirements: Manage industrial criteria and integration strategy: what is expected from each designer/sub-contractor. Specify requirements and interfaces of all configuration items.

Operational Analysis

- Operational Analysis
- System Analysis
- Define Operational Entities and Capabilities
  - DOED Create a new Operational Entity Breakdown diagram
  - DOCE Create a new Operational Capabilities diagram
- Define Operational Activities and describe Interactions
  - DOAD Create a new Operational Activity Breakdown diagram
  - DOAI Create a new Operational Activity Interaction diagram
  - DOAS Create a new Operational Activity Scenario
- Allocate Operational Activities to Operational Actors, Entities or Roles
  - DOAR Create a new Operational Architecture diagram
  - DOAR Create a new Operational Role diagram
  - DOES Create a new Operational Entity Scenario
- Transverse Modelling
  - DCBI Create a new Class diagram

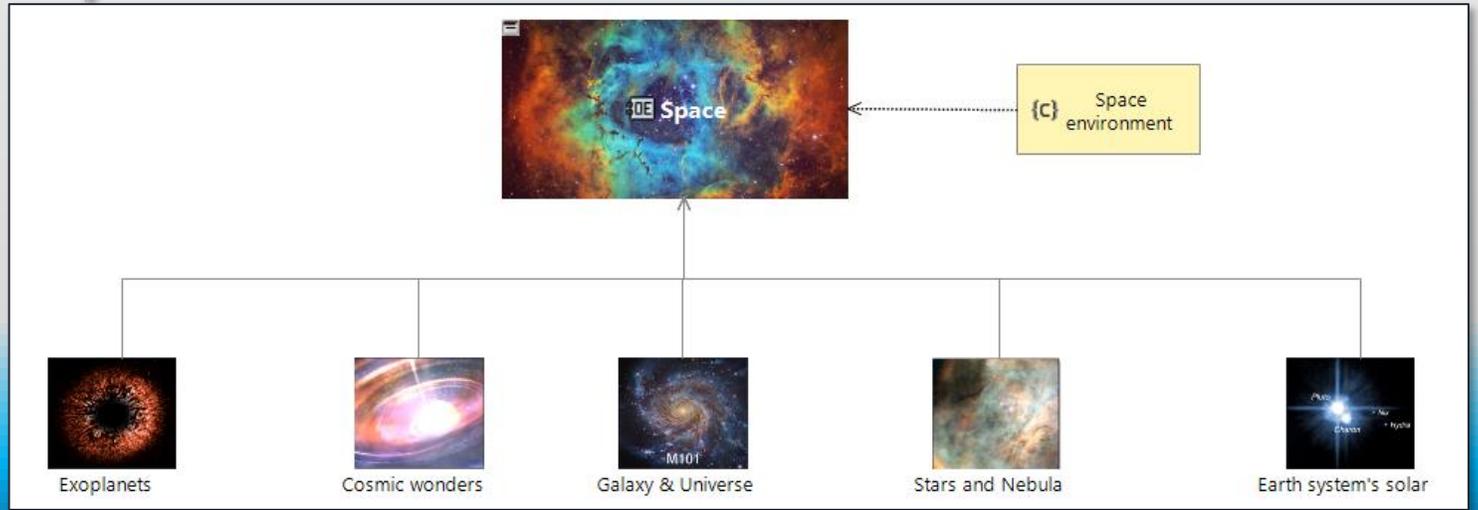
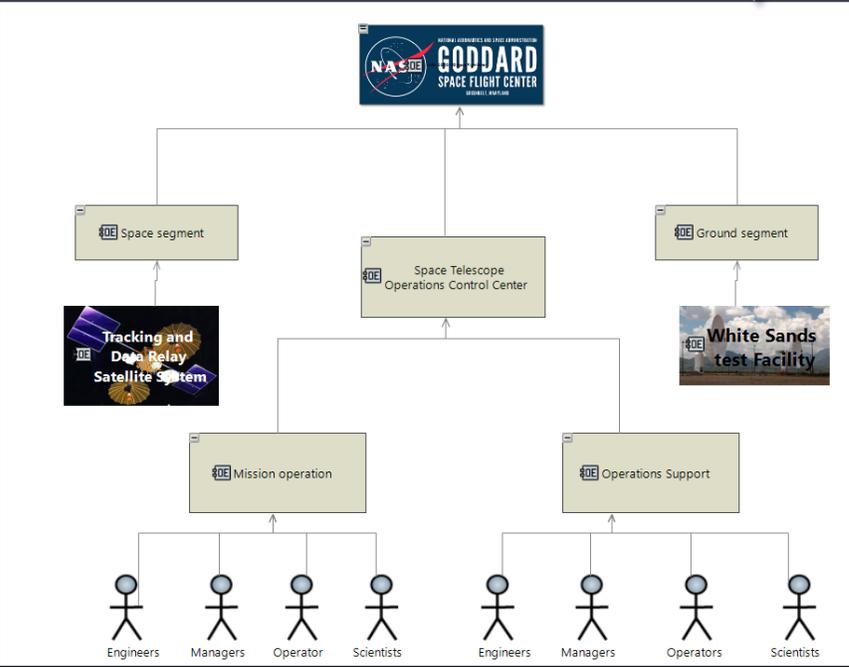
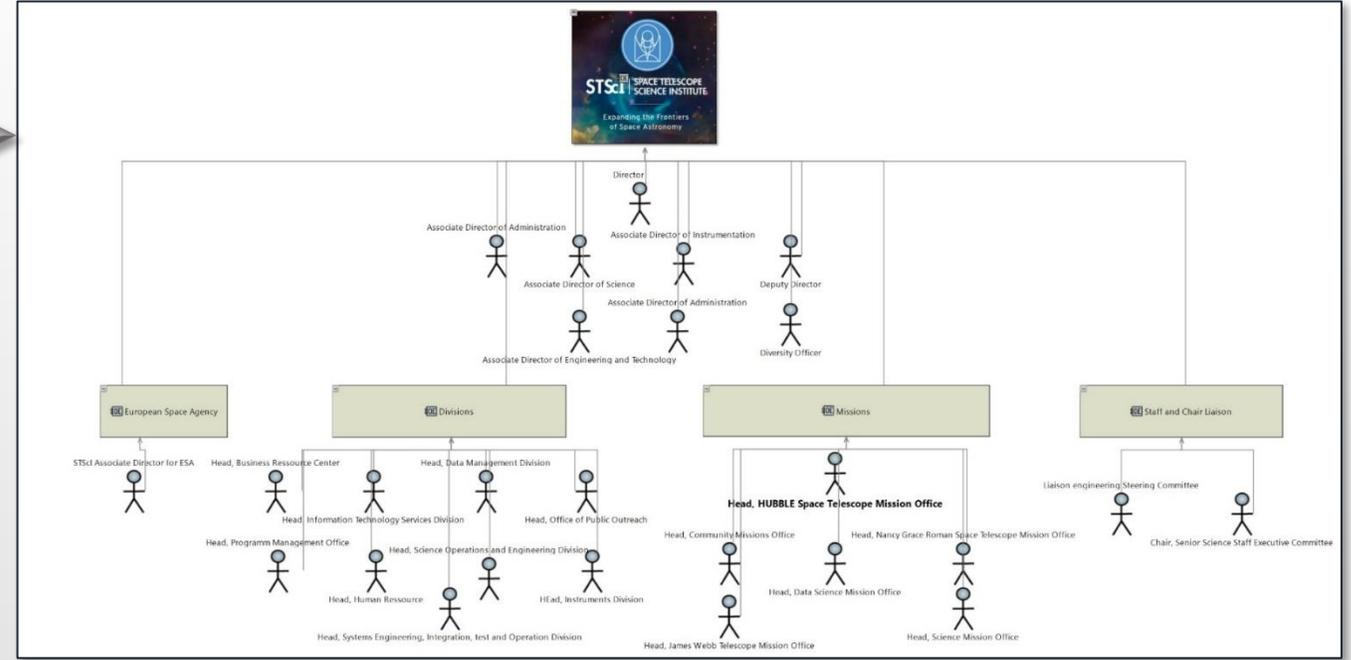
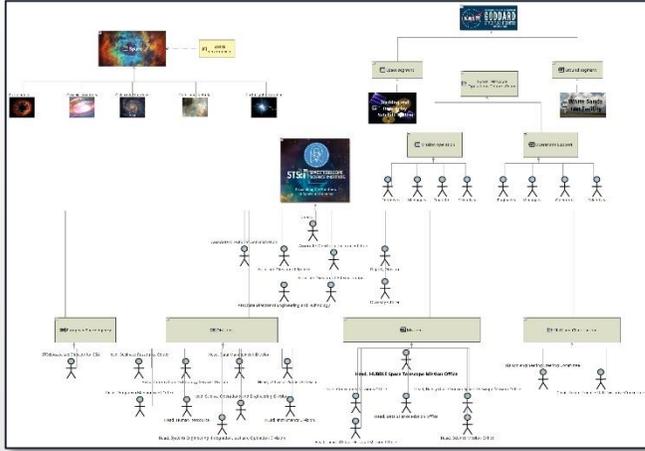
Diagram Editor

The diagram illustrates the operational analysis of the Hubble Space Telescope, showing the interaction between various entities and activities. Key components include:

- Entities:** NASA/GSFC/AD Space Flight Center, Space Telescope Science Institute, Space Telescope Operations Control Center, and White Sands test facility.
- Activities:** Mission operation, Operations support, Mission planning, and various control and monitoring tasks.
- Interactions:** Data flows and control signals between the entities and activities.

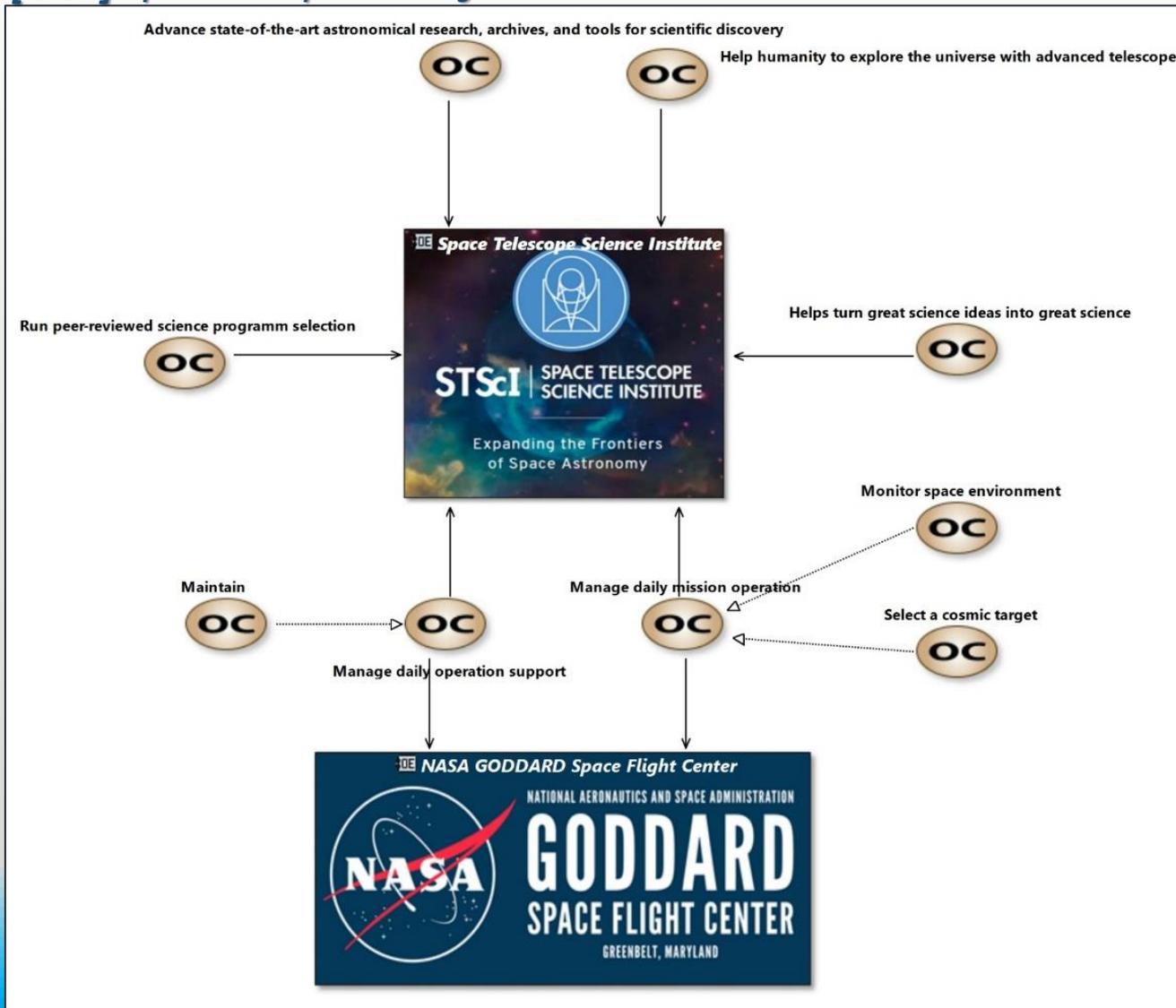


## [OEBD] Operational Entity Breakdown diagram





## [OCB] Operational Capabilities diagram



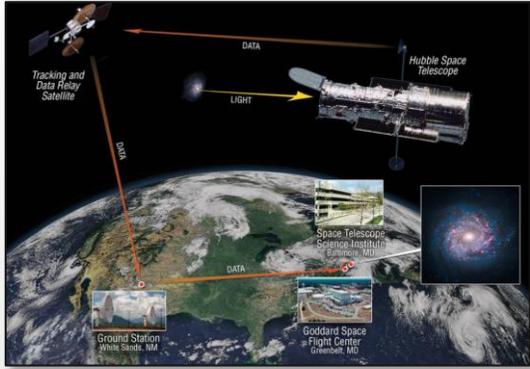
Behind Hubble's captivating images and groundbreaking science is a team of people who control the telescope, ensure its health and safety, and innovate ways to keep it at top performance more than three decades after its launch. This group of engineers, scientists, and operators at NASA's Goddard Space Flight Center work together to monitor Hubble as it travels around Earth, point the telescope at cosmic targets, and solve any problems that arise. They perform their work in specialized facilities that provide the tools and equipment needed to operate this great observatory and continue its legacy of success.



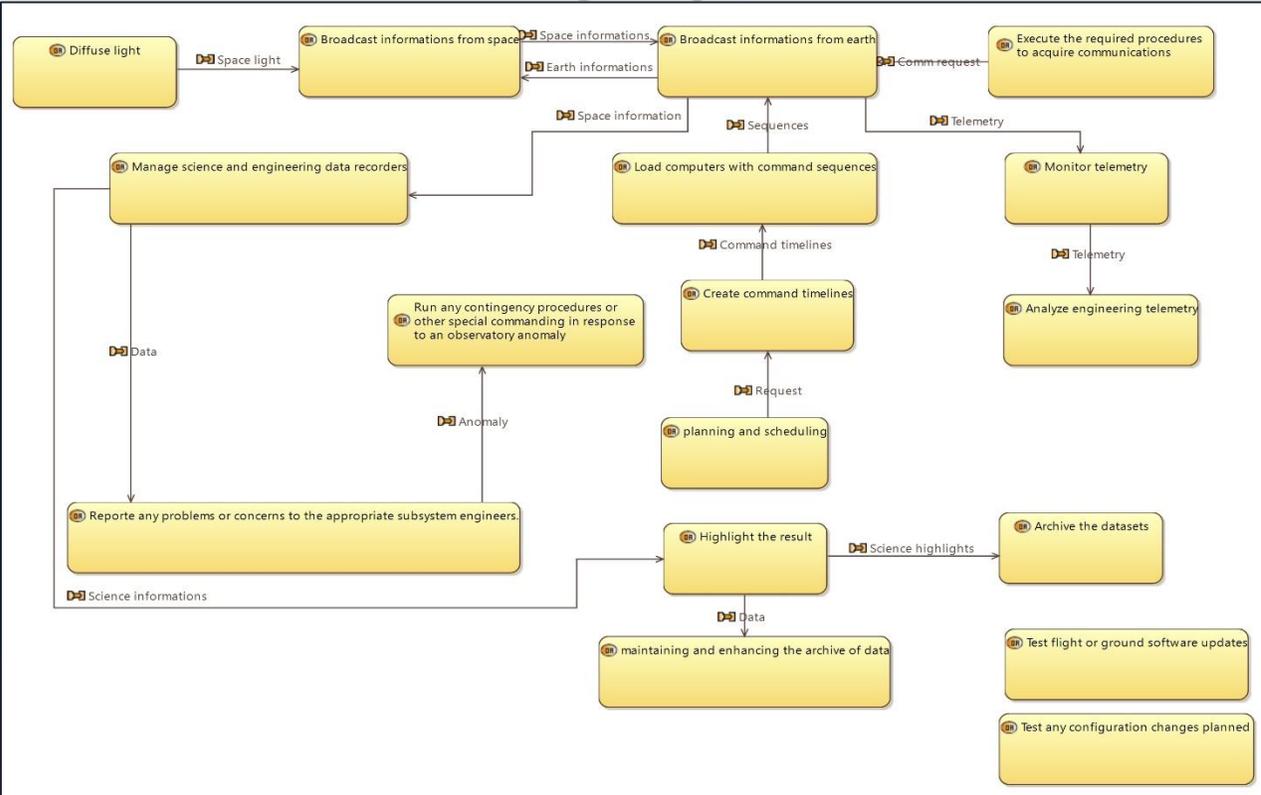
**HUBBLE SPACE TELESCOPE (HST)**  
 LAUNCHED APRIL 24, 1990

We are the science operations center for the Hubble Space Telescope. Our work covers the lifecycle of a scientific proposal for Hubble observations, which we have managed since its launch in 1990. We help turn great science ideas into great science, highlight the results, and distribute the data acquired for others to use. Our work includes running the peer-reviewed science program selection, planning and scheduling of the telescope, characterizing the performance of the instruments, maintaining and enhancing the archive of data, and making the data freely available to the world.

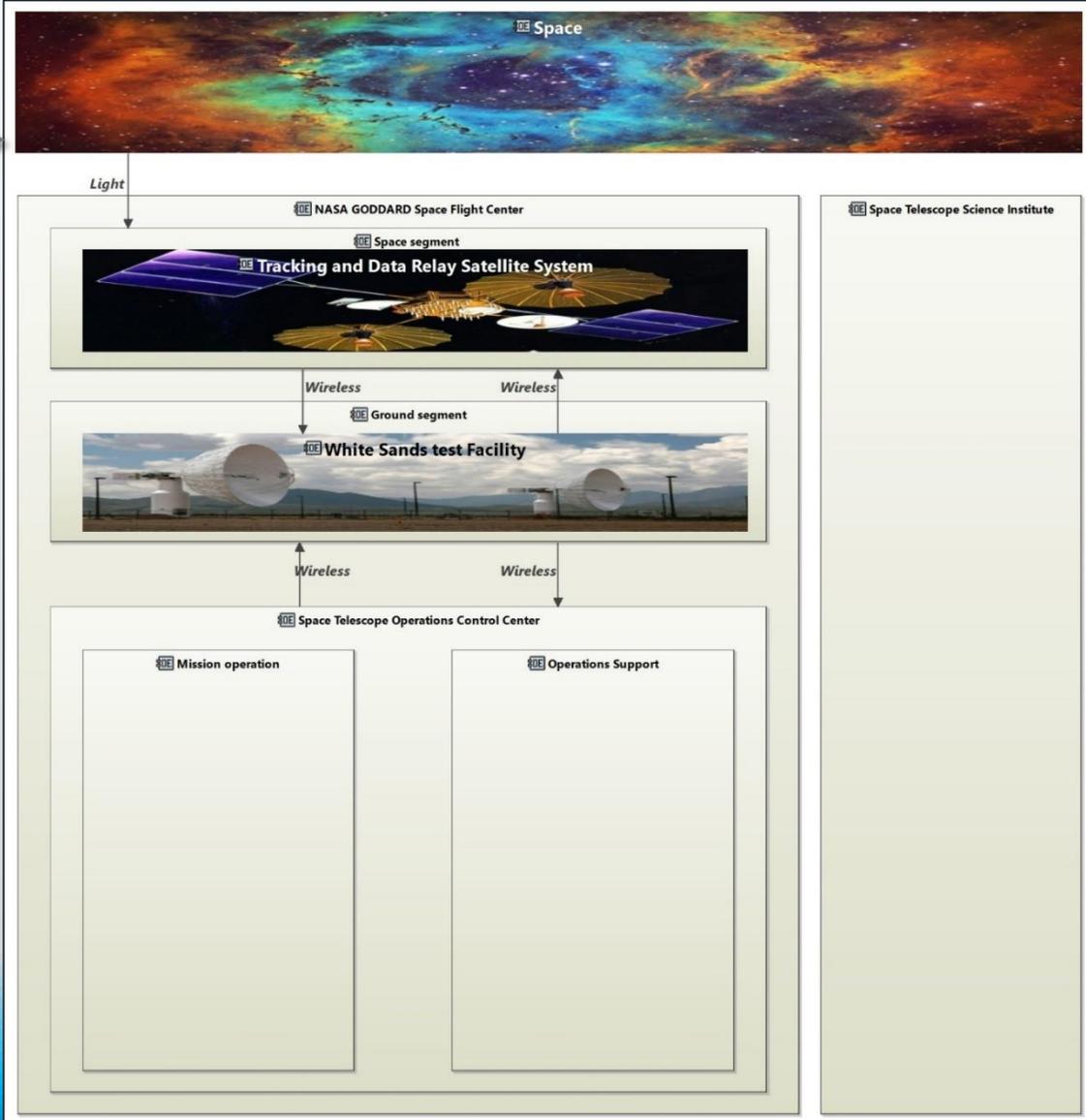




[OAI] Operational Activity Interaction diagram



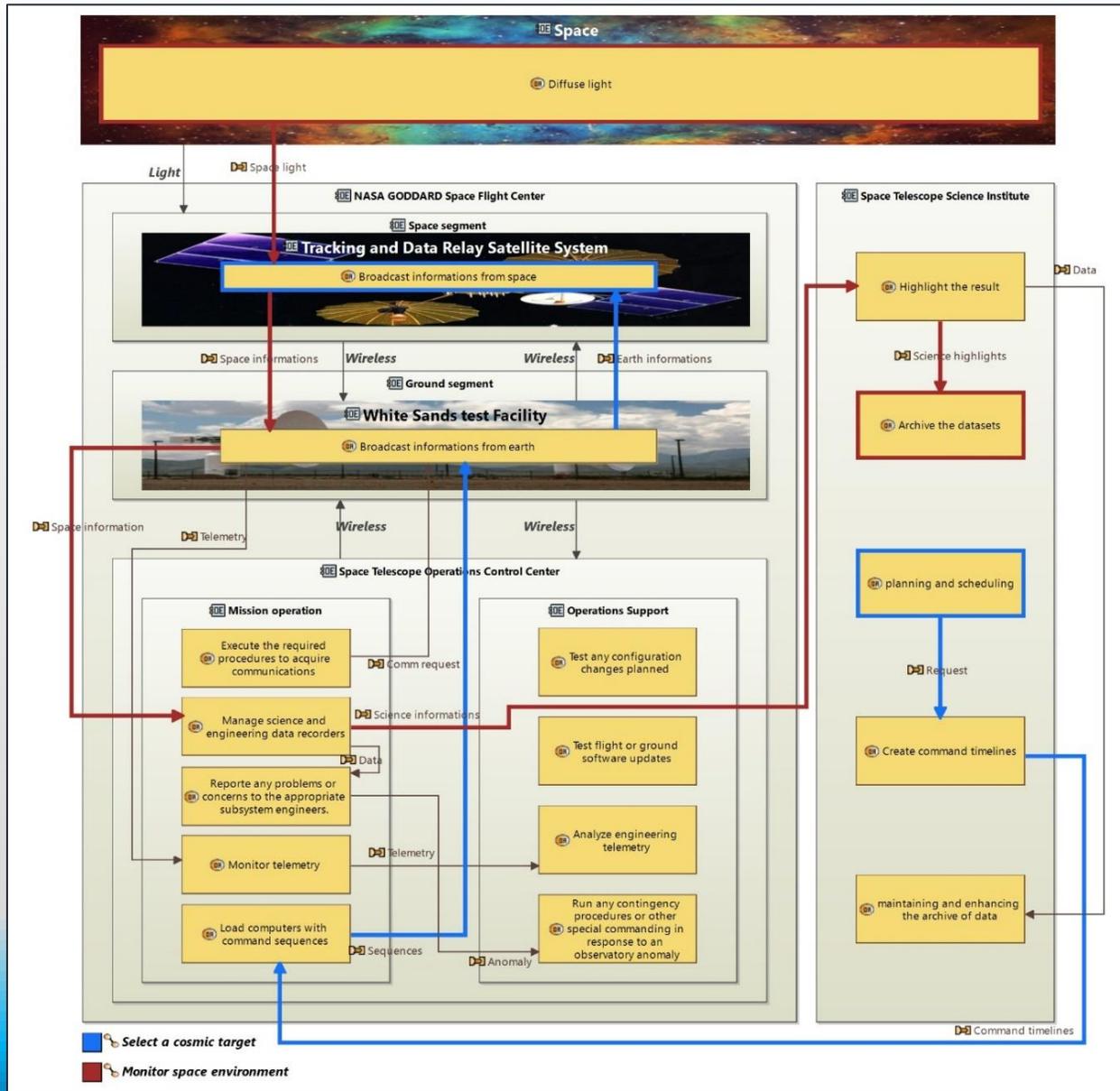
[OAB] Operational Architecture diagram



# Operational Analysis

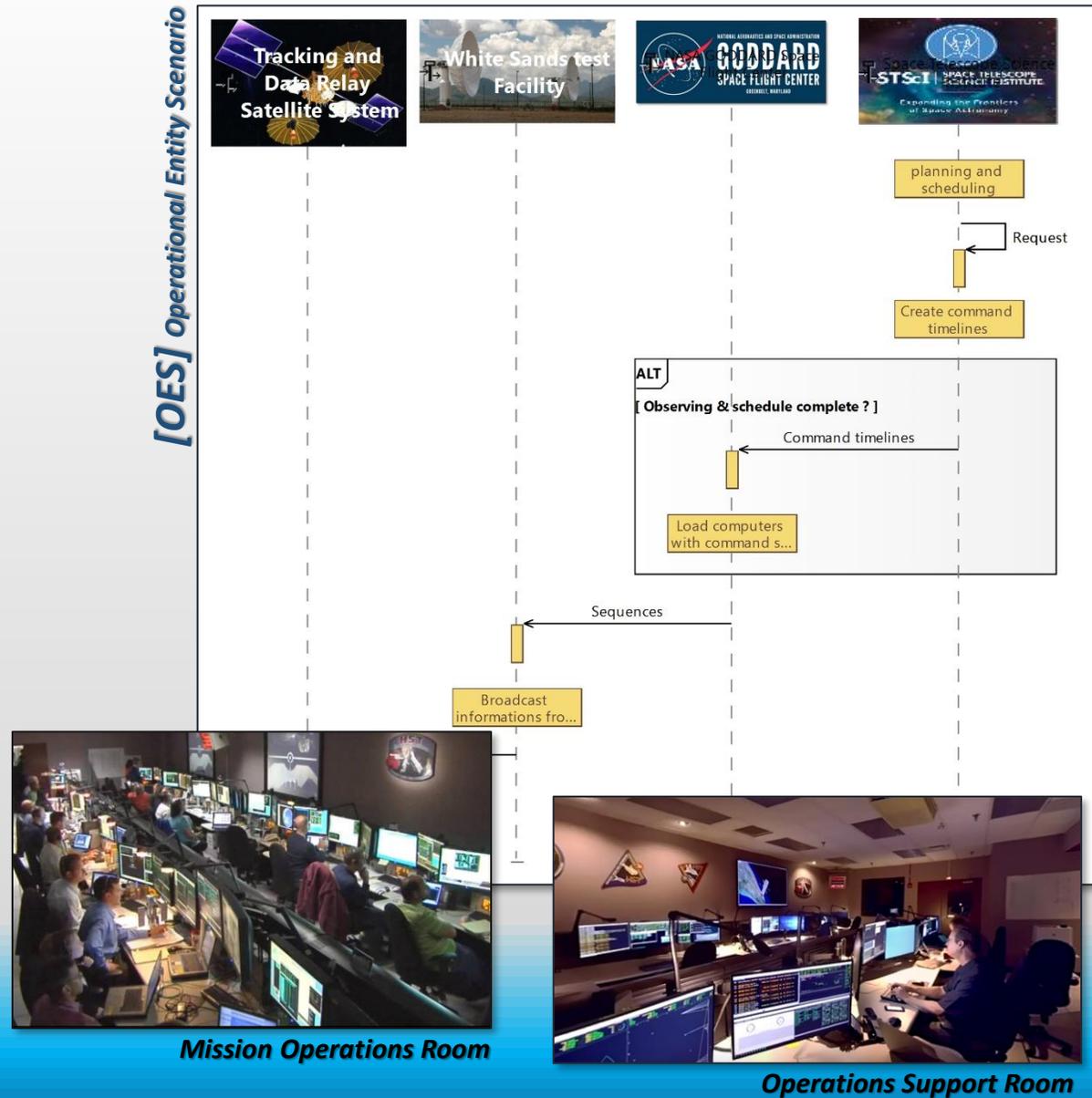


Speaker: DROUIN Remy



[OAB] Operational Architecture diagram

[OES] Operational Entity Scenario



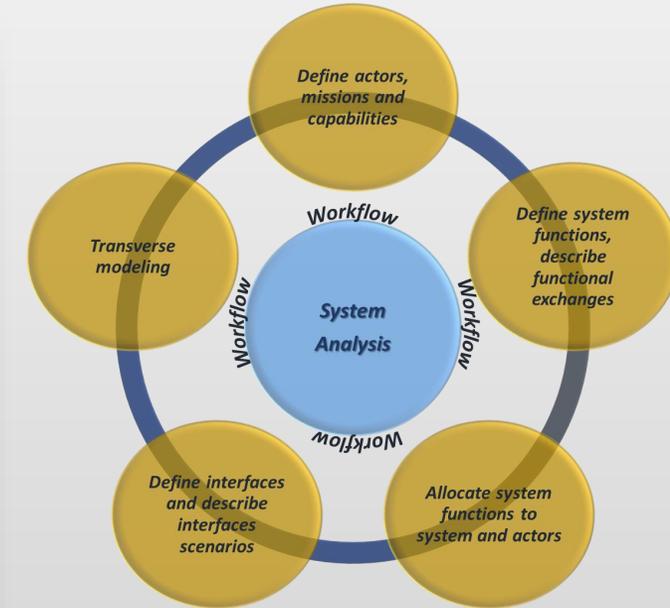
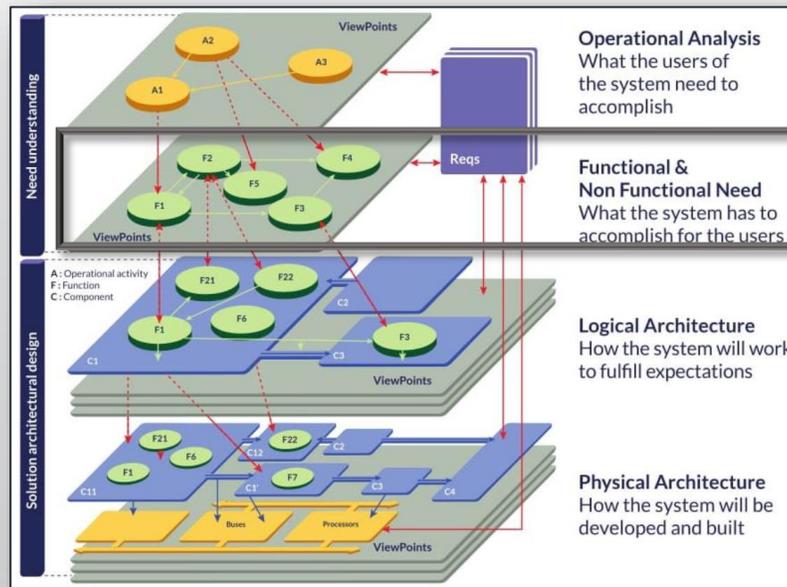
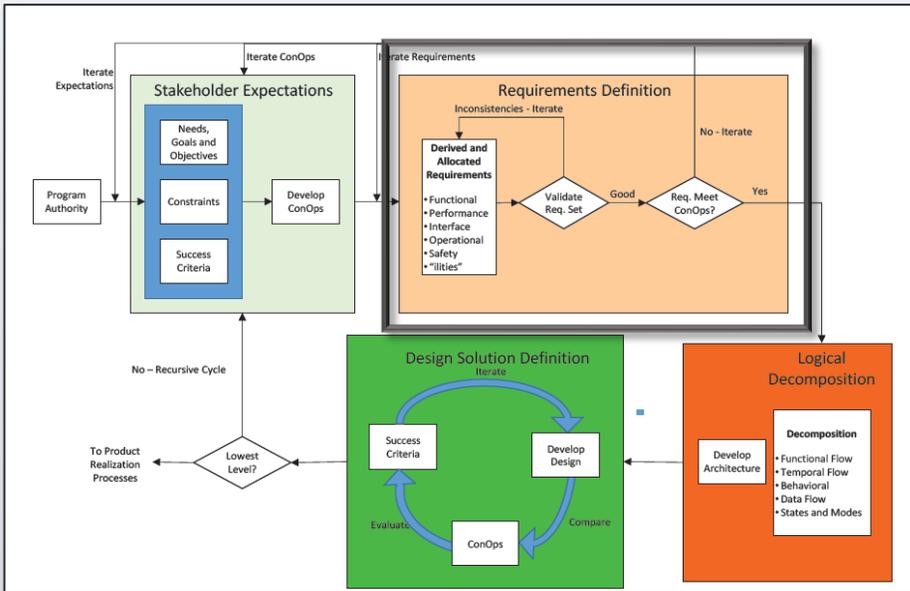
Mission Operations Room

Operations Support Room

# System Analysis

**Requirements definition process**  
**Technical solution definition process**

**Need understanding**  
**Solution architectural design**



## Activity explorer

## Activity explorer

## Diagram editor

The image displays a series of screenshots from the Capella software interface, illustrating the workflow for the Hubble Space Telescope system analysis. The workflow is organized into several stages:

- Operational Analysis:** Define Stakeholder Needs and Environment. Capture and consolidate operational needs from stakeholders. Define what the users of the system have to accomplish. Identify entities, actors, roles, activities, concepts.
- System Analysis:** Formalize System Requirements. Identify the boundary of the system, consolidate requirements. Define what the system has to accomplish for the users. Model functional dataflows and dynamic behaviour.
- Logical Architecture:** Develop System Logical Architecture. See the system as a white box: define how the system will work so as to fulfill expectations. Perform a first trade-off analysis.
- Physical Architecture:** Develop System Physical Architecture. How the system will be developed and built. Software vs. hardware allocation, specification of interfaces, deployment configurations, trade-off analysis.
- EPBS:** Formalize Component Requirements. Manage industrial criteria and integration strategy: what is expected from each designer/sub-contractor. Specify requirements and interfaces of all configuration items.

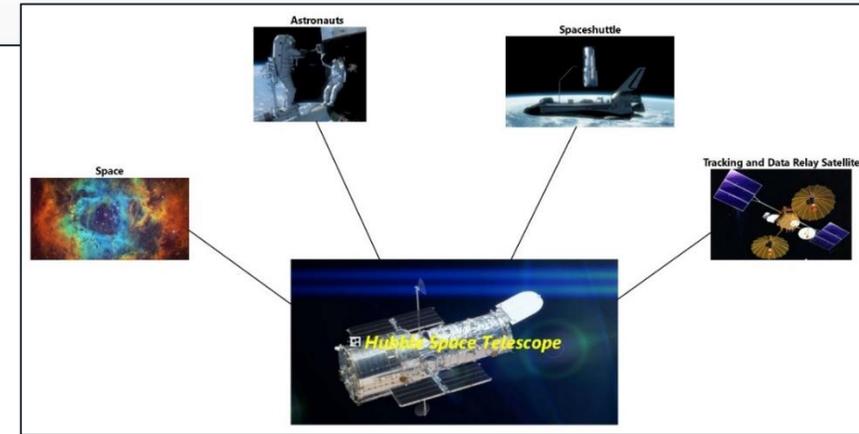
The 'System Analysis' section is further detailed with tasks such as:

- Transition From Operational Activities
  - Perform an automated transition of Operational Activities
  - Create a System Functions / Operational Activities Traceability Matrix
- Define Actors, Missions and Capabilities
  - Perform an automated transition of System Analysis Capabilities
  - Continuously create new System Actors from Operational Entities / Actors
  - Continuously create new System Capability or Mission from Operational Capability
  - (CSA) Create a new Contentual System Actors diagram
  - Create a new Mission and / or Capability Blank diagram
  - Create a System Actors / Operational Entities Traceability Matrix
- Refine System Functions, describe Functional Exchanges
  - (SFFD) Create a new Functional Breakdown diagram
  - (SDFD) Create a new Functional Dataflow Blank diagram
  - (FS) Create a new Functional Scenario
- Allocate System Functions to System and Actors

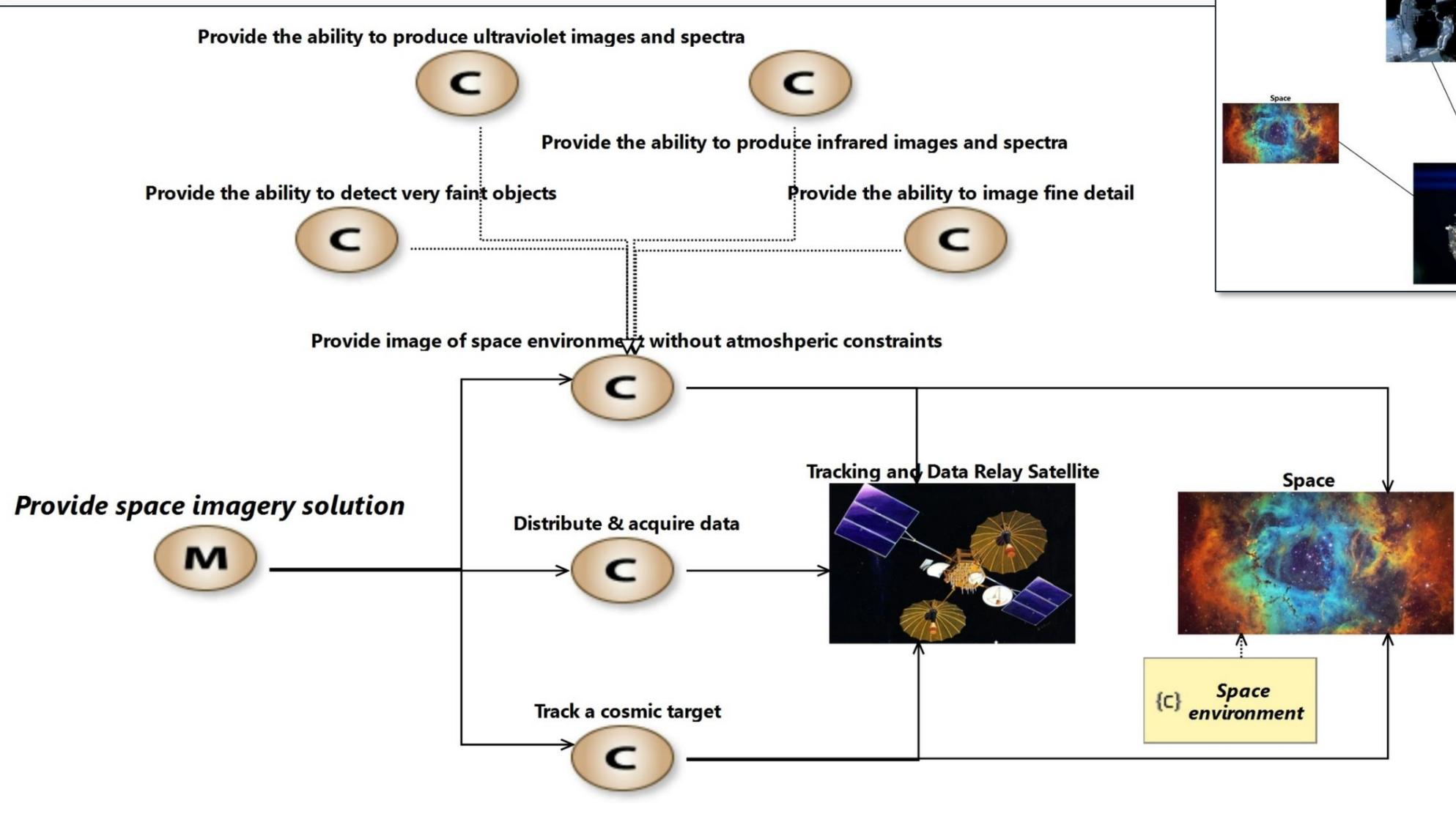
The diagram editor window shows a complex functional architecture diagram with various components and their interactions.

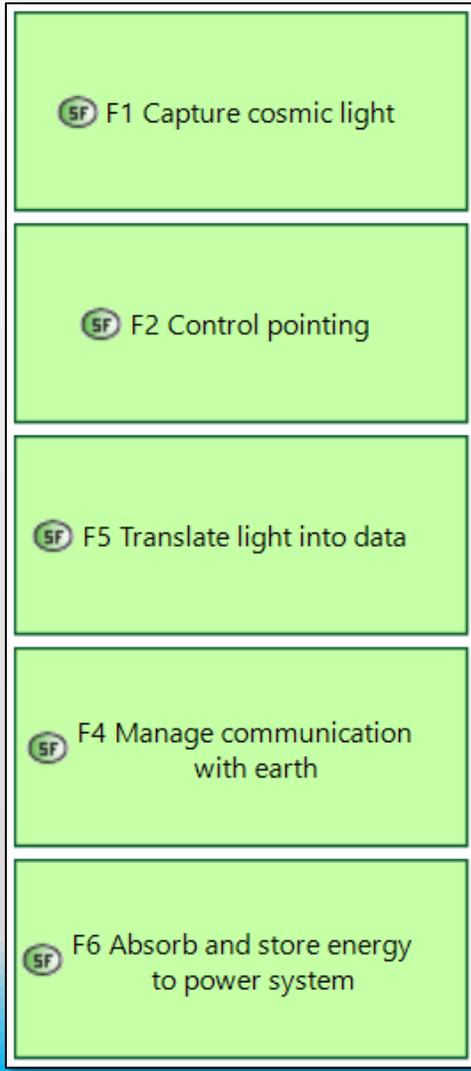


[CSA] Contextual System Actors diagram

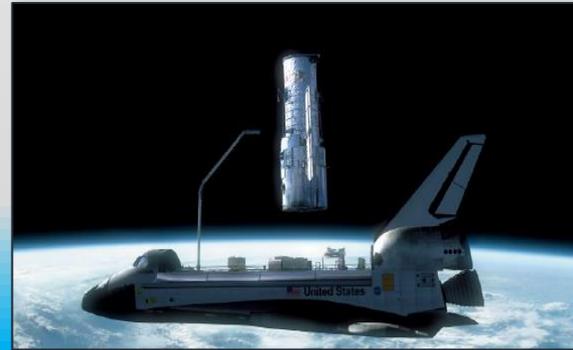
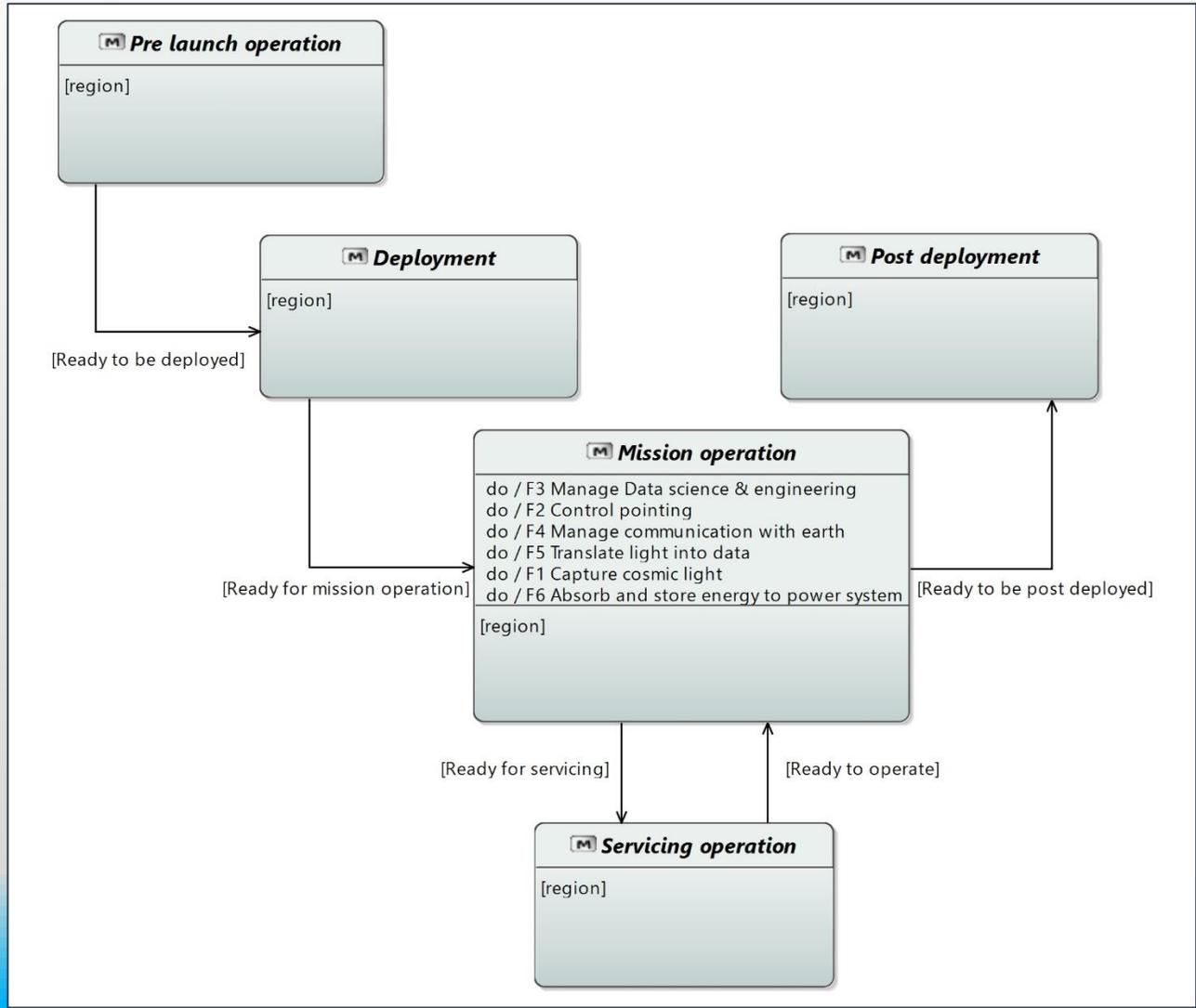


[MCB] Mission Capability Blank diagram



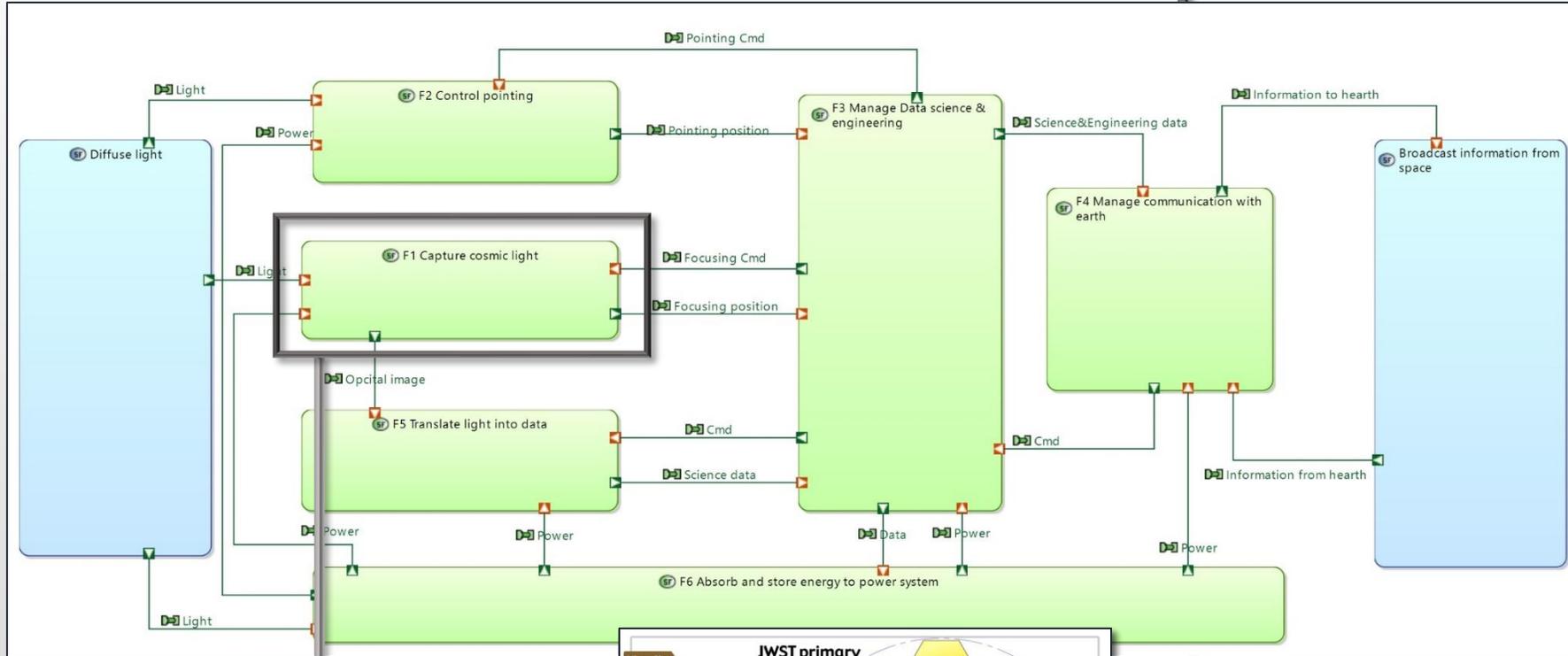


## [M&S] Mode&State diagram

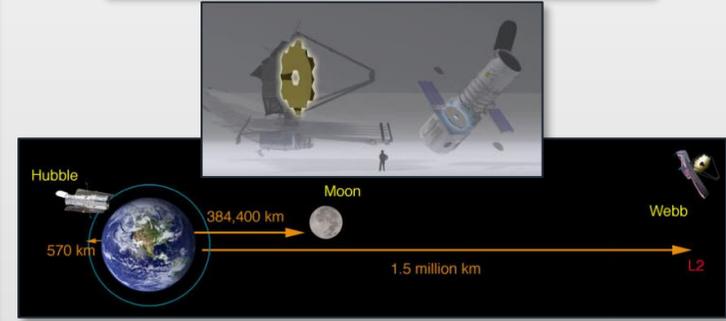
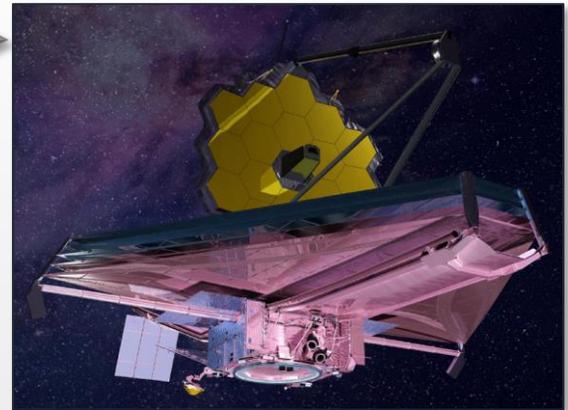




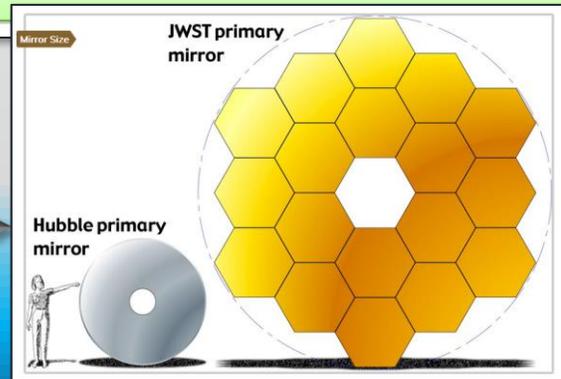
## [SDFB] Functional Dataflow Blank diagram



James Webb Telescope



F1 Capture cosmic light

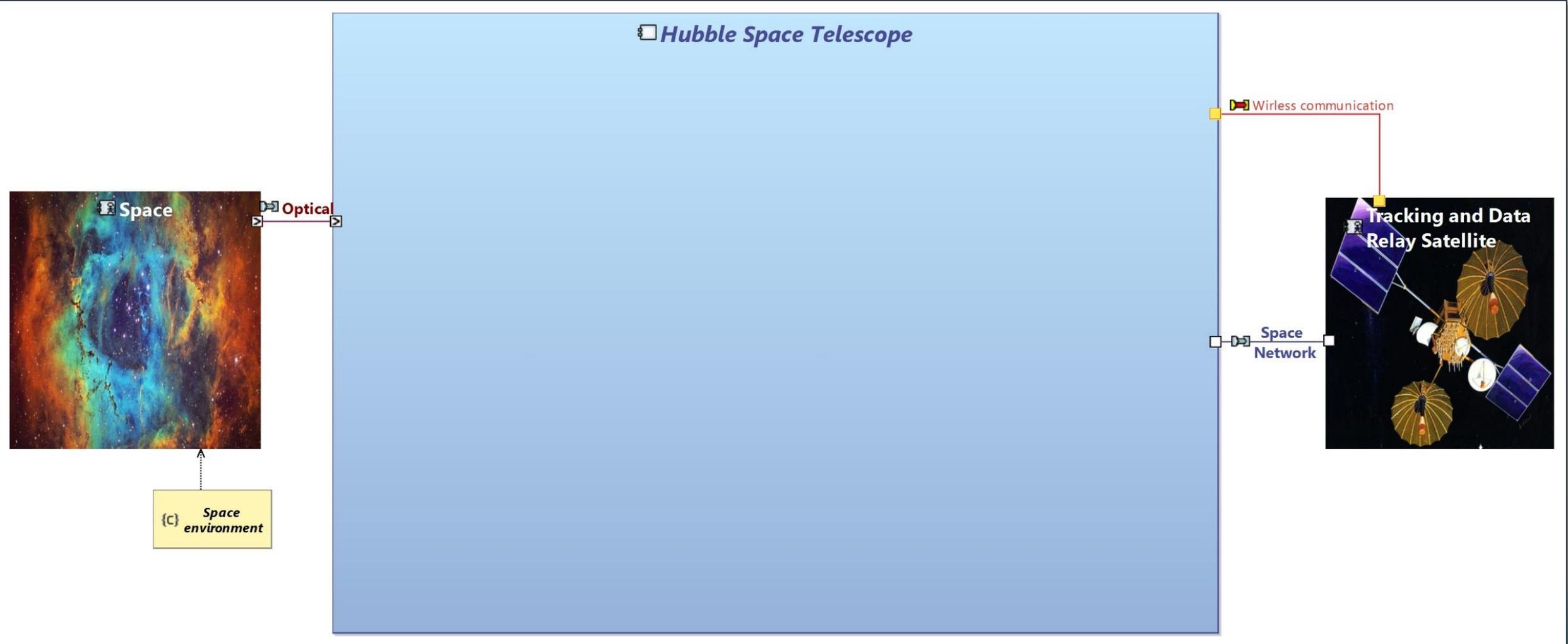


HUBBLE Telescope



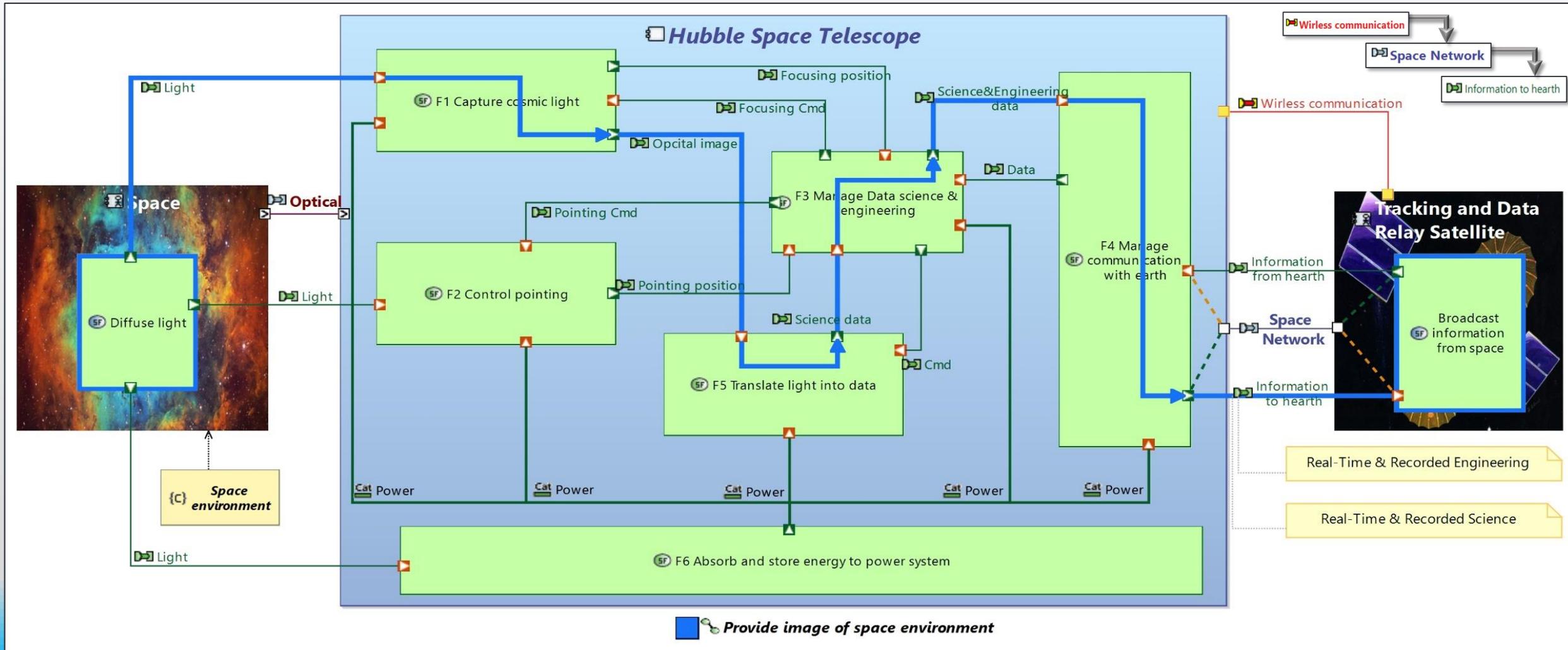


[SAB] System Architecture diagram



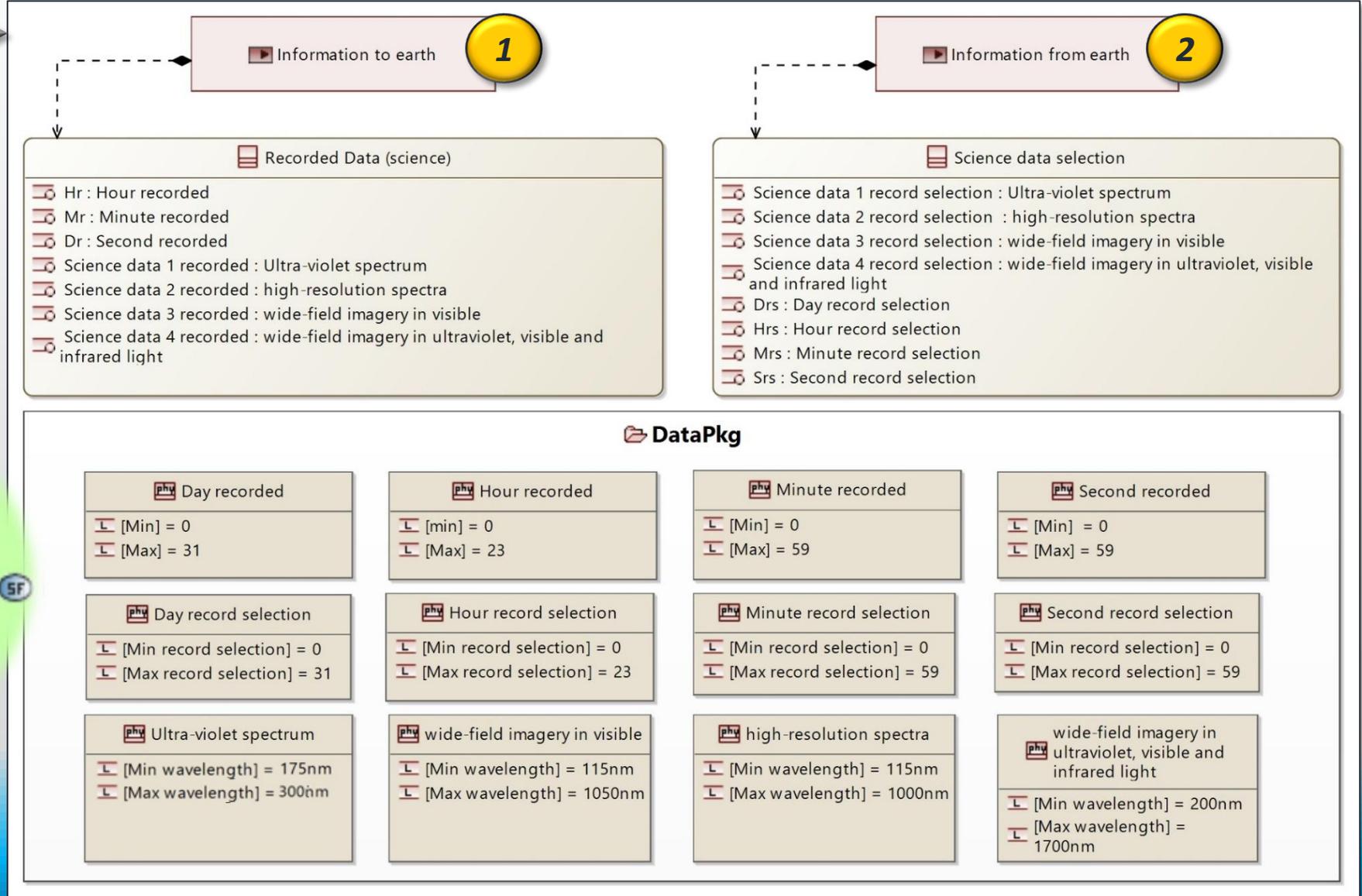
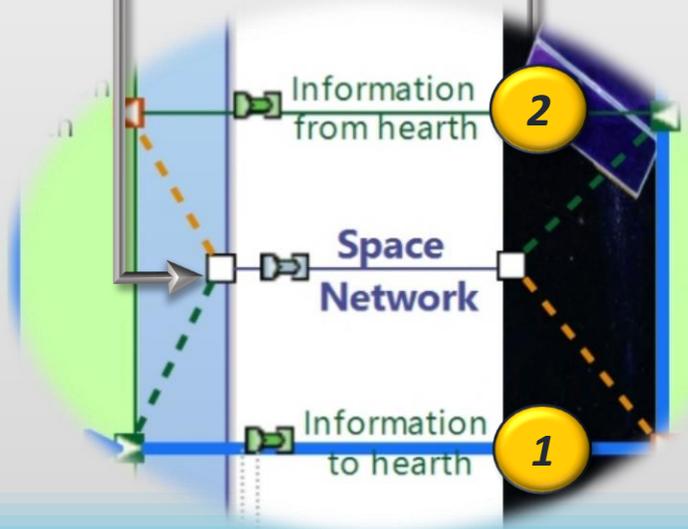
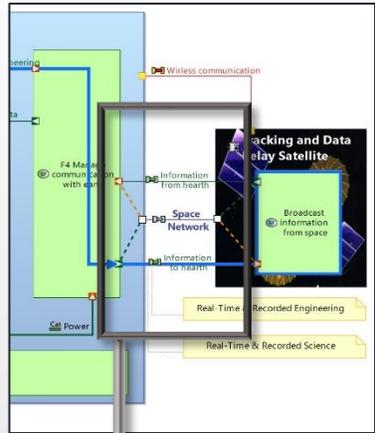


## [SAB] System Architecture diagram



Real-Time & Recorded Engineering

Real-Time & Recorded Science

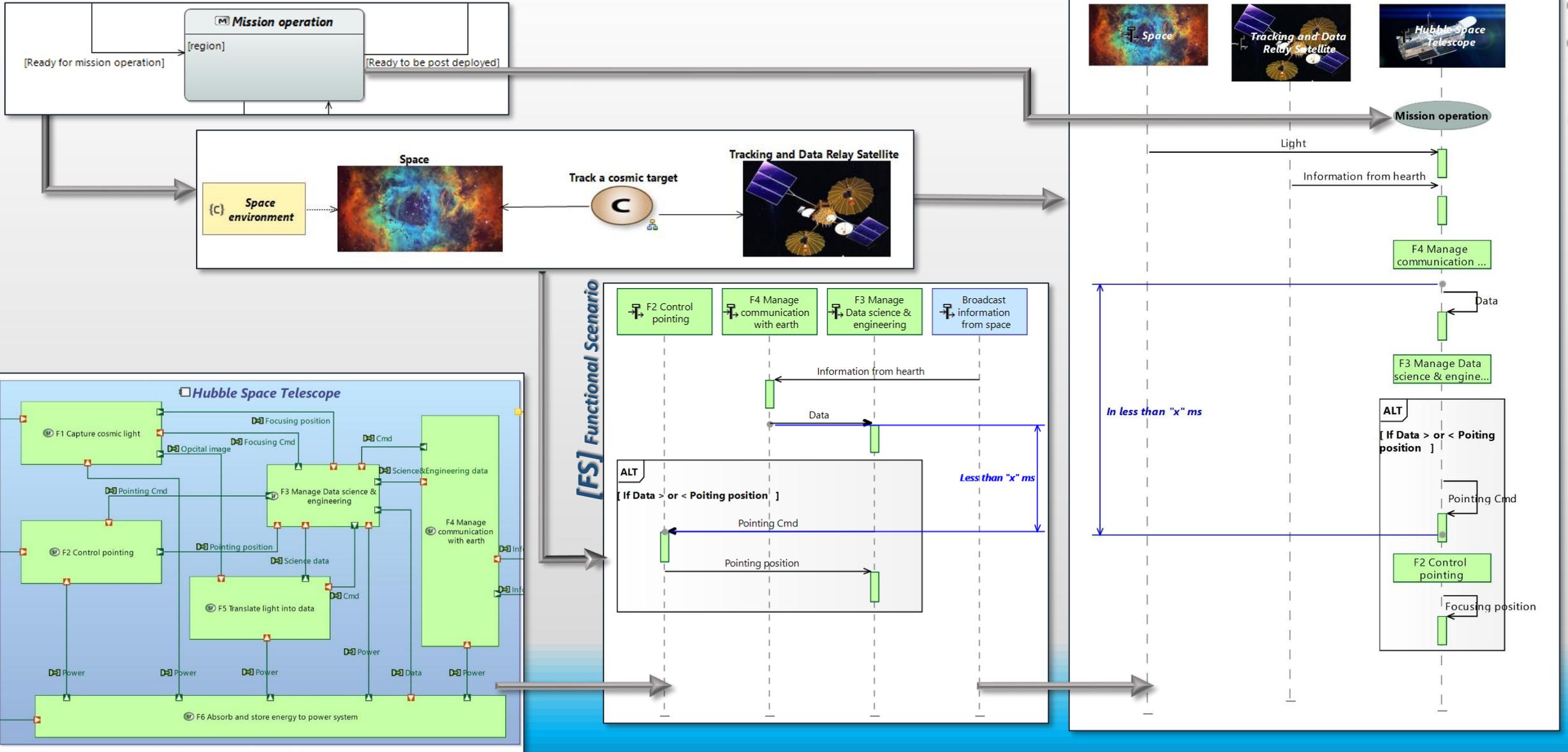


# System Analysis



Speaker: DROUIN Remy

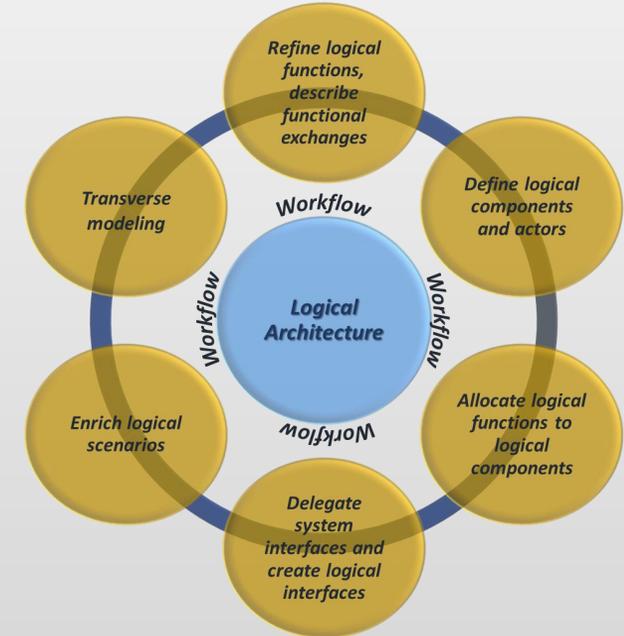
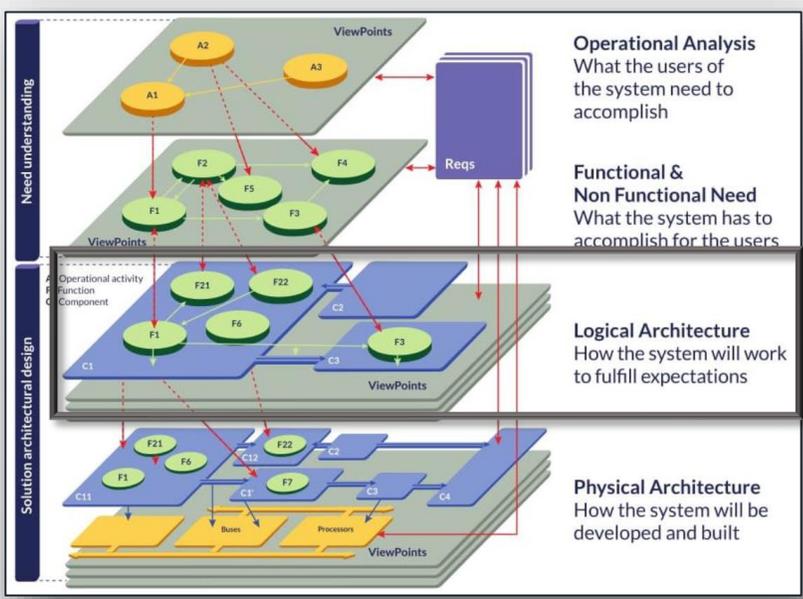
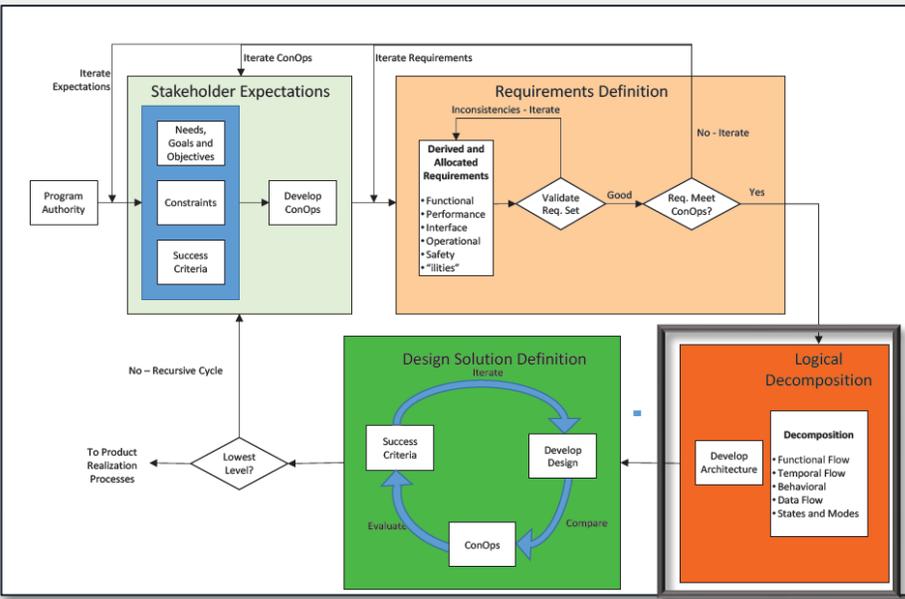
[ES] Exchange Scenario



# Logical Architecture

**Requirements definition process**  
**Technical solution definition process**

**Need understanding**  
**Solution architectural design**



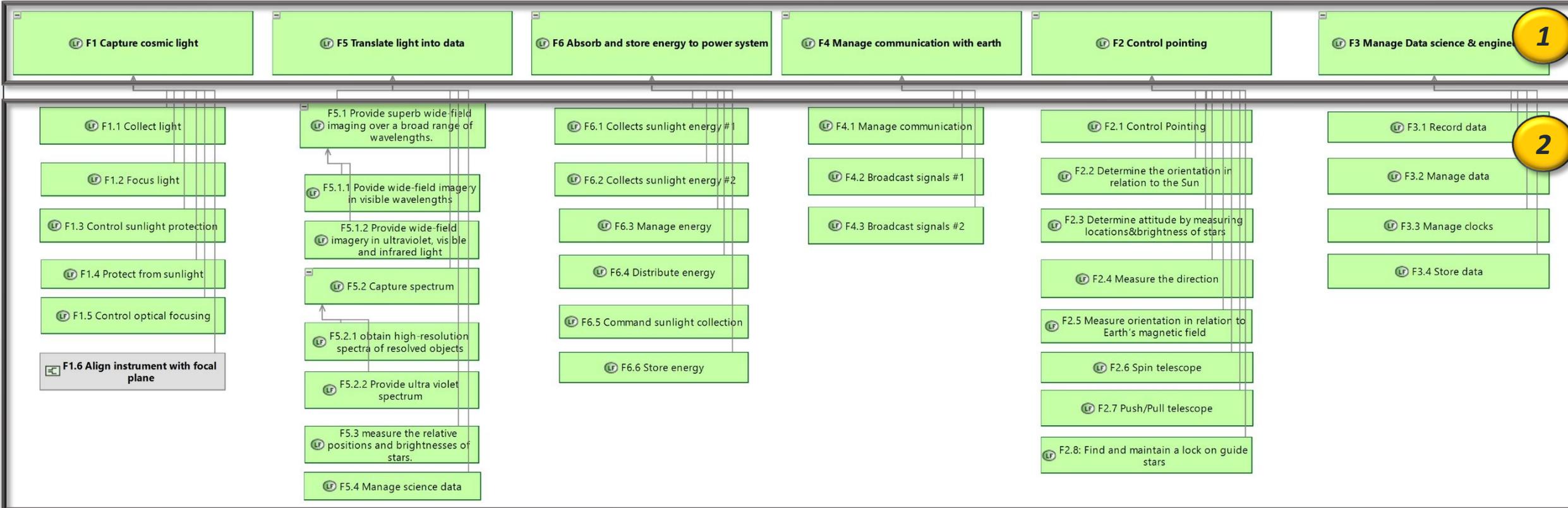
## Activity explorer

## Activity explorer

## Diagram editor

The image displays a collage of screenshots from the Capella software interface, illustrating the Logical Architecture process for the Hubble Space Telescope. The screenshots show the 'Activity explorer' (Workflow of Hubble Space Telescope), the 'Logical Architecture' activity details, the 'Diagrams Viewer', and the 'Diagram editor' showing a complex functional decomposition diagram. The diagram editor shows a detailed functional decomposition of the Hubble Space Telescope, including subsystems like Painting control, Sensors, Gyroscope, Magnetic Sensing System, Reaction Wheel, Magnetic torques, Power supply management, Solar Array #1, Solar Array #2, Batteries, and Store energy. The diagram editor also shows a sidebar with various model elements like Components, Logical Component, Logical Actor, Component Exchange, In Flow Port, Component Exchanges, Functions, Functional Exchange, Port Allocation, Manage Function Allocation, Allocated Functions, Functional Chains, Functional Exchanges, Accelerators, Functions from Mode / State, and Elements from Scenario.

## [LFBFD] Functional Breakdown diagram

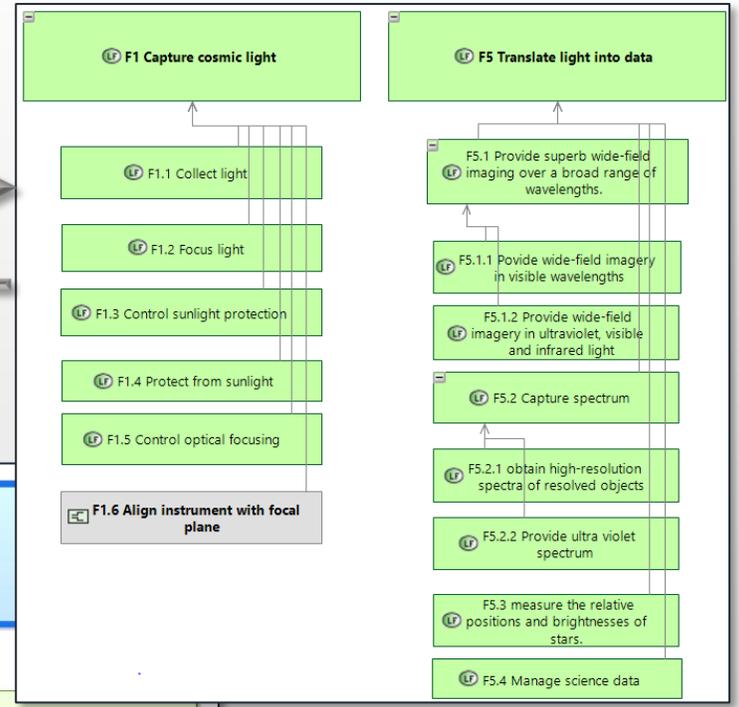
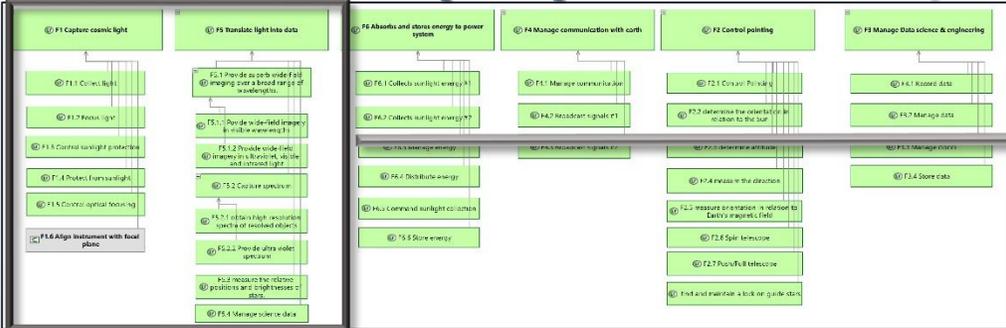


**1** From System Analysis

**2** From Logical Architecture

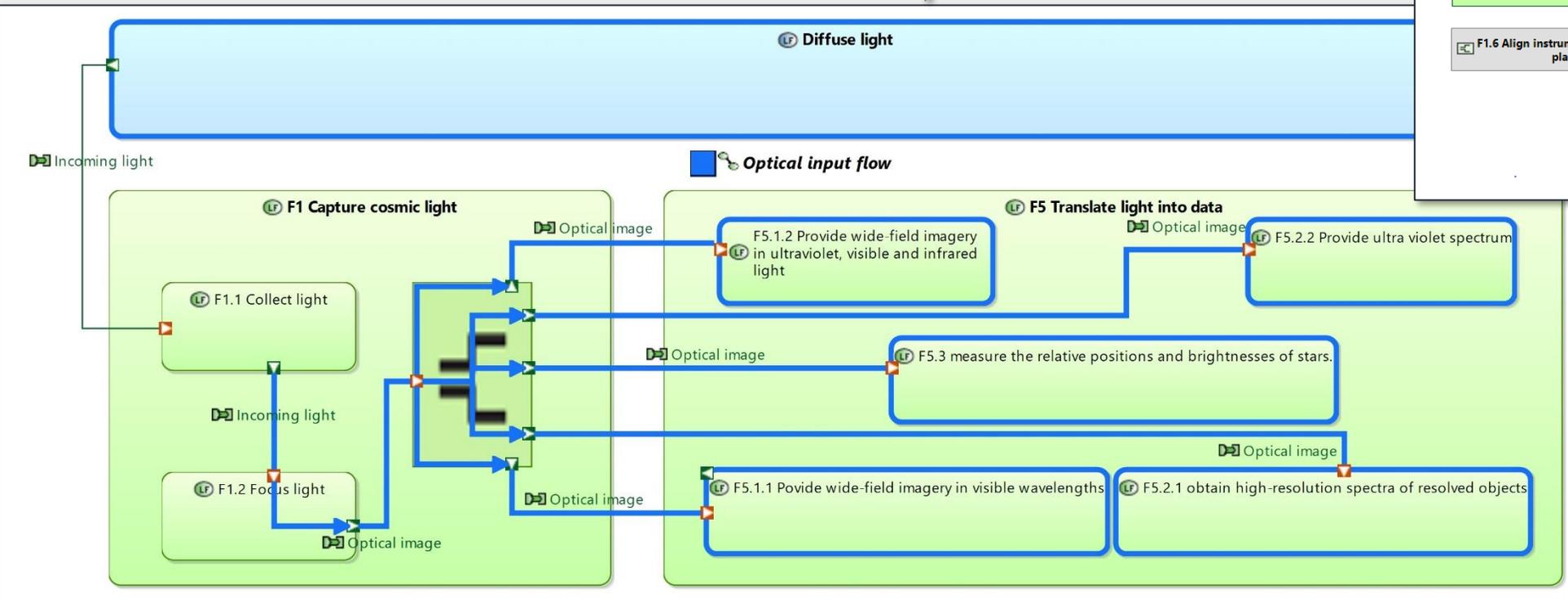


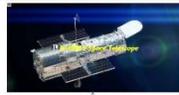
[LFB] Functional Breakdown diagram



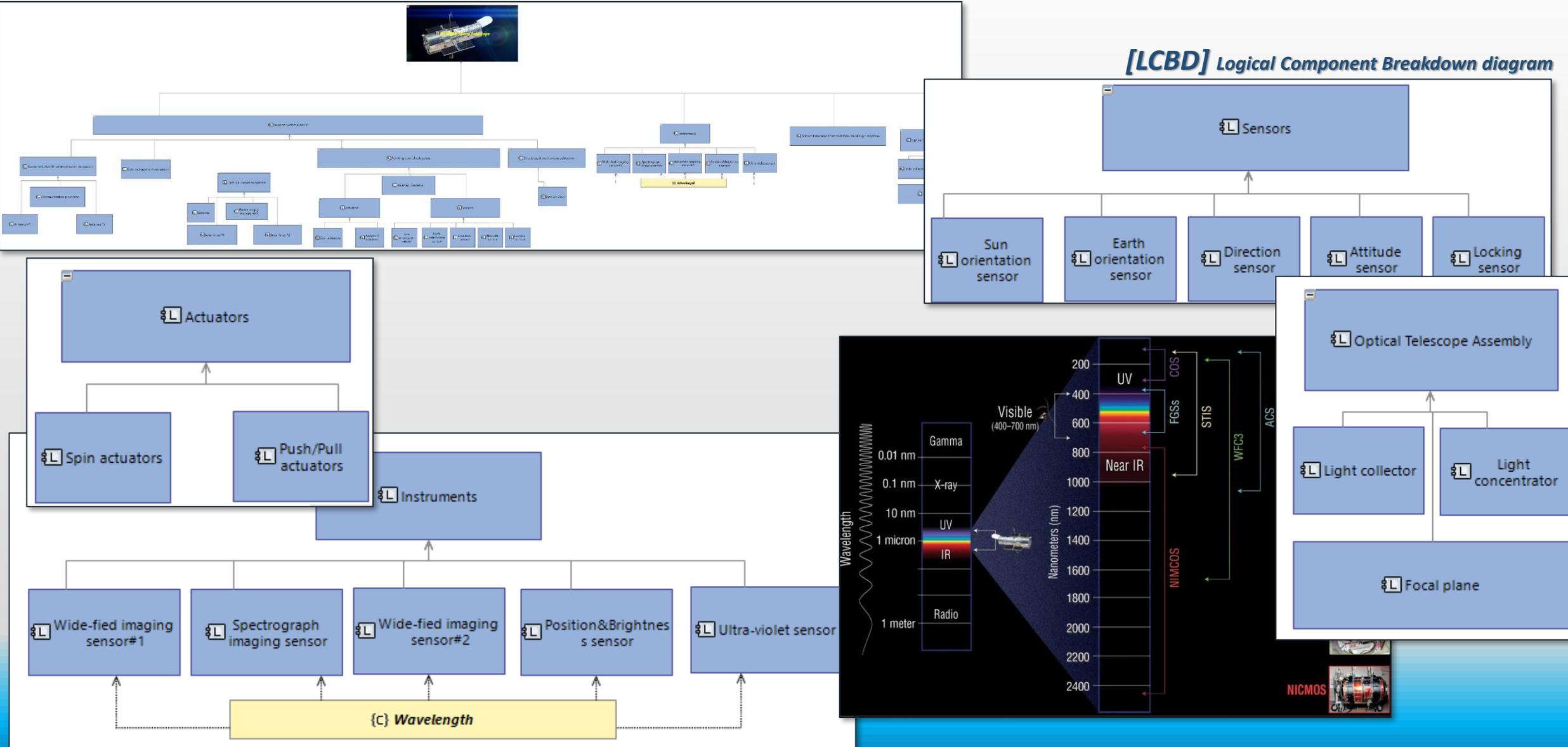
[LFB] Functional Breakdown diagram

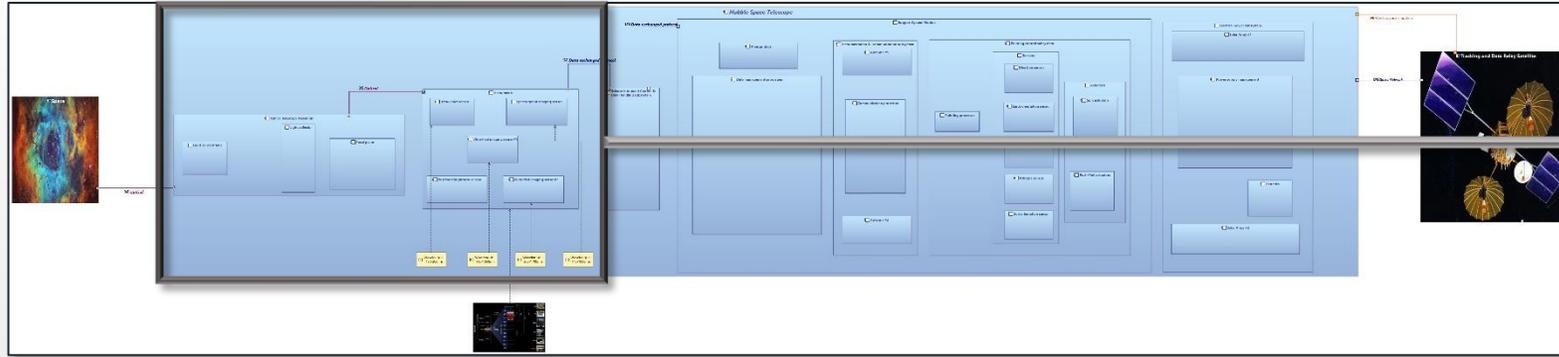
[LDFB] Functional Dataflow Blank diagram



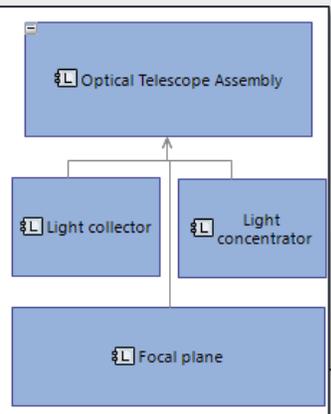


[LCBD] Logical Component Breakdown diagram

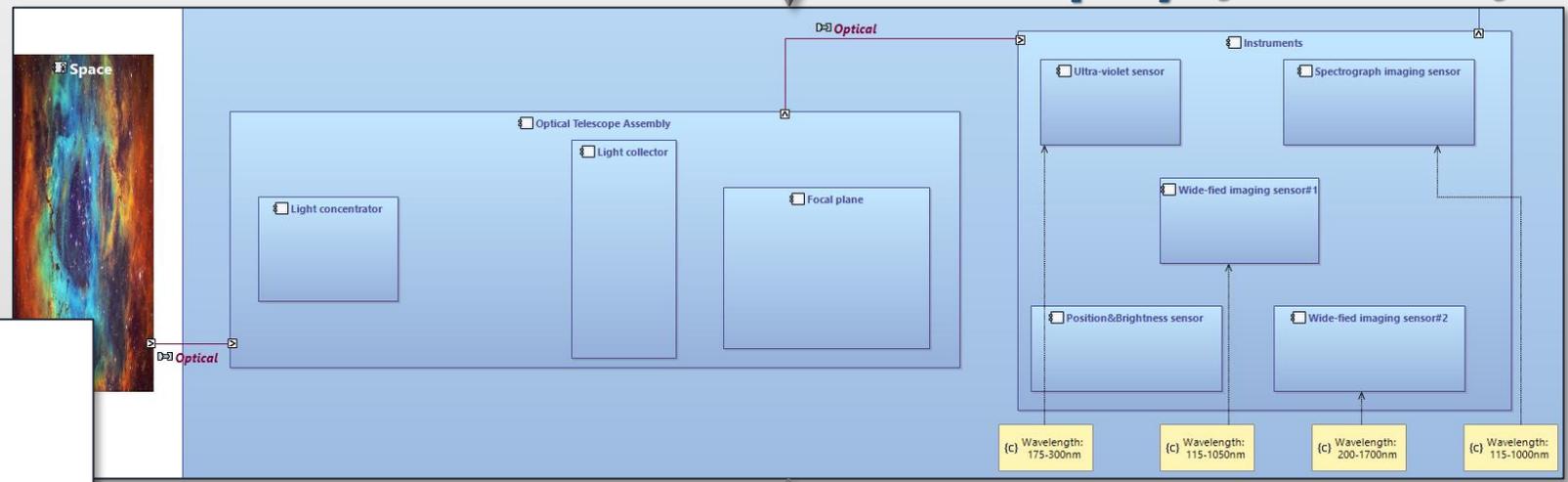
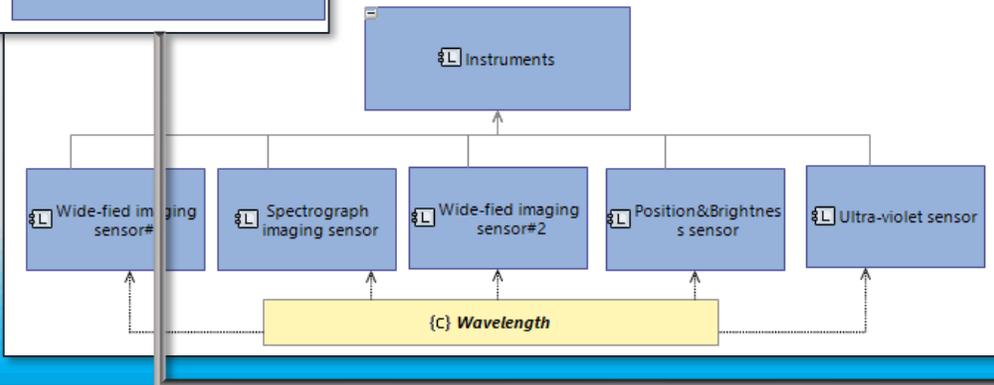


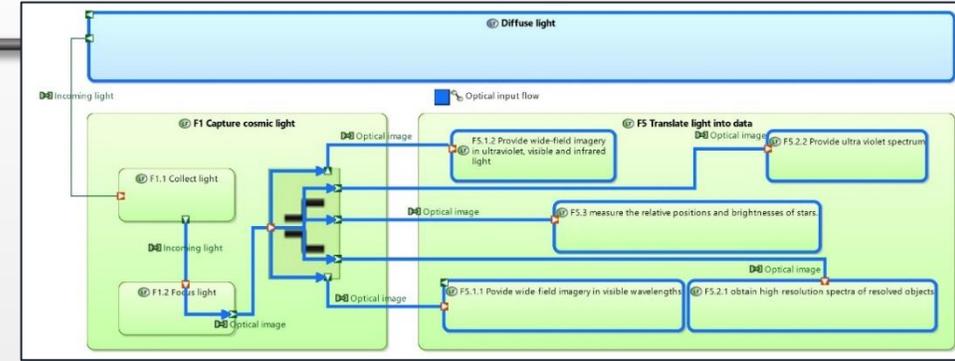
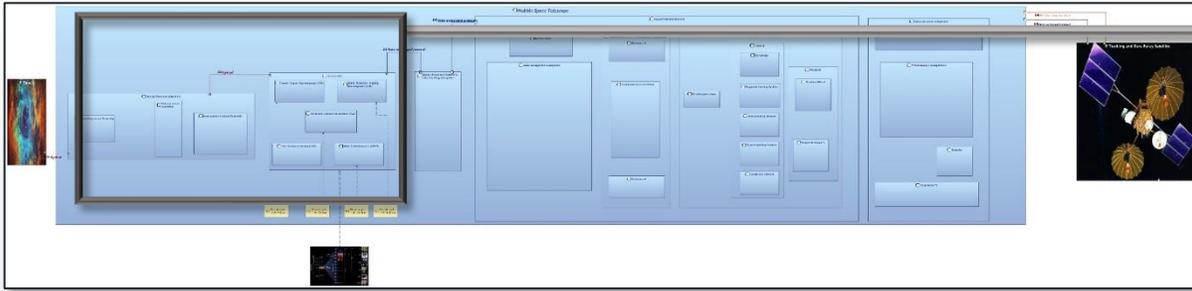


**[LAB] Logical Architecture diagram**

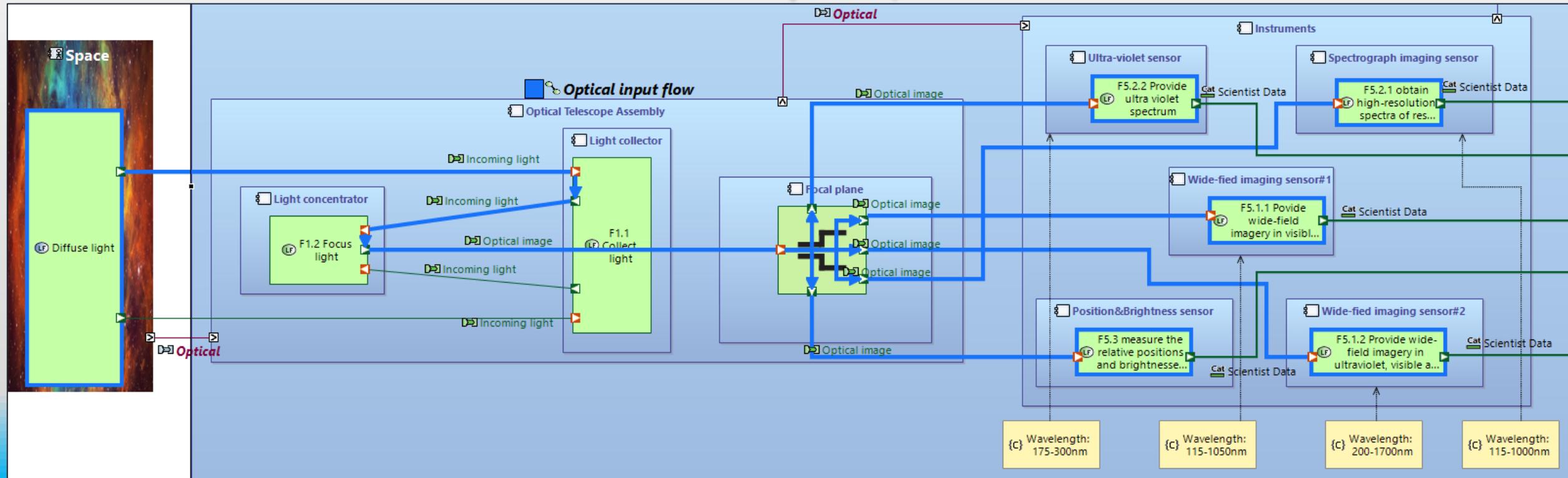


**[LCBD] Logical Component Breakdown diagram**





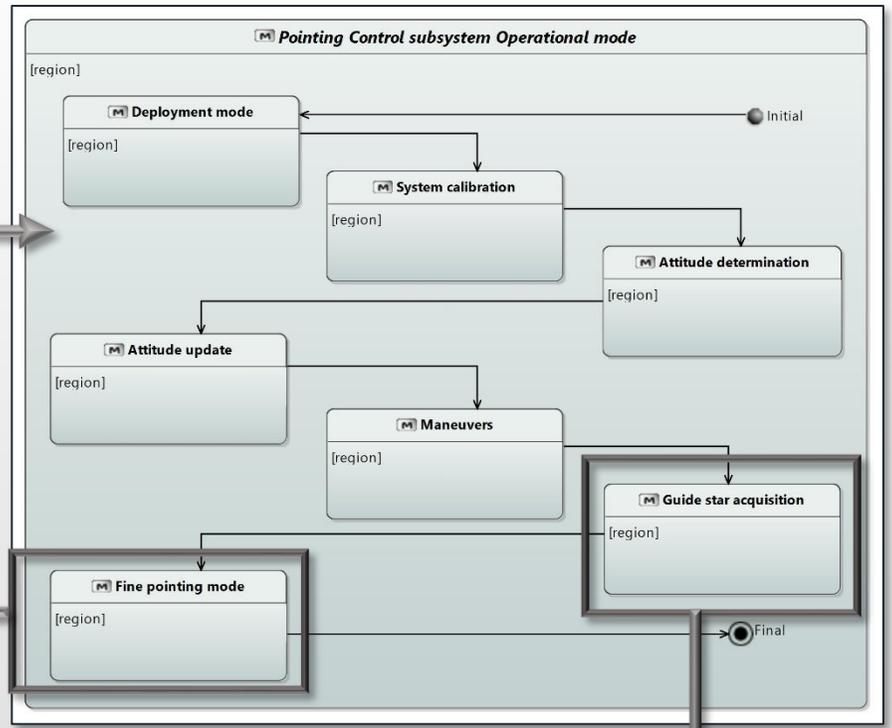
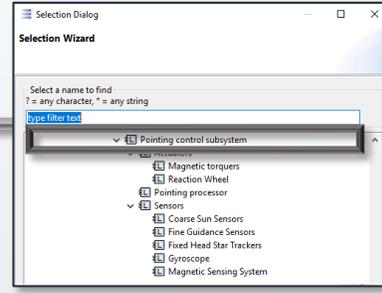
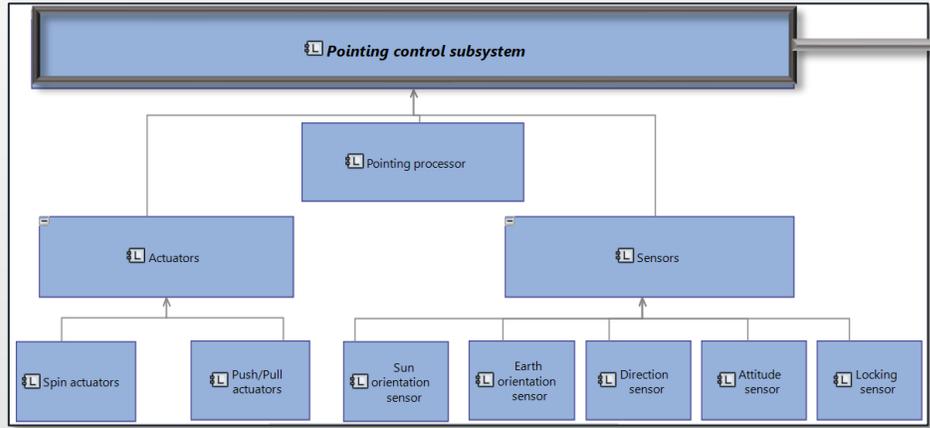
[LAB] Logical Architecture Diagram





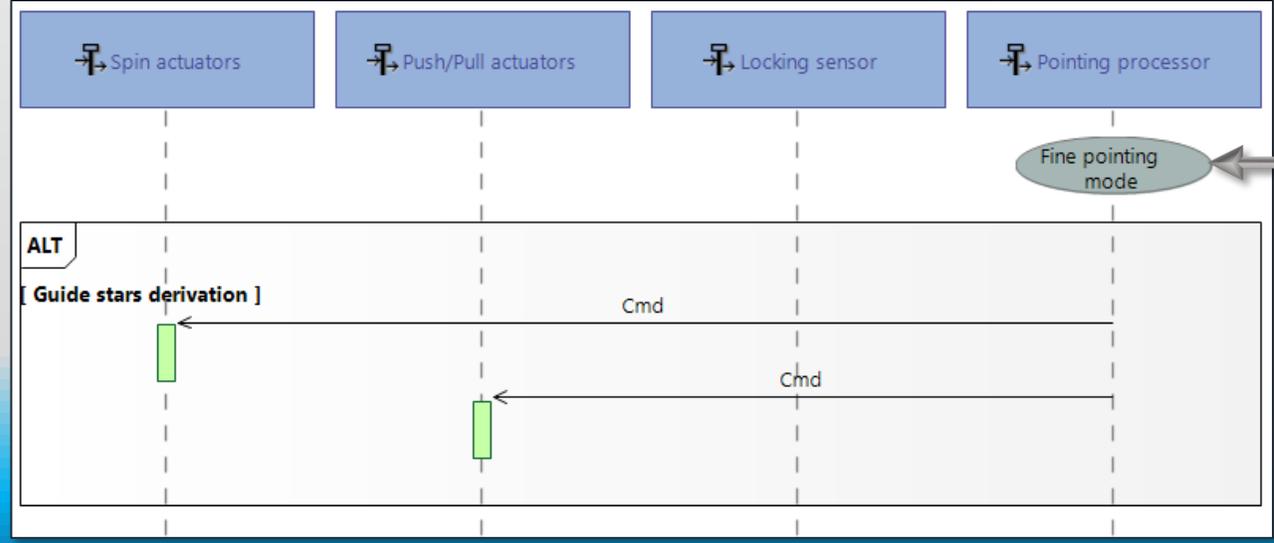
## Pointing control subsystem analysis Breakdown/Mode&State/Sequence diagram

### [LCBD] Logical Component Breakdown diagram

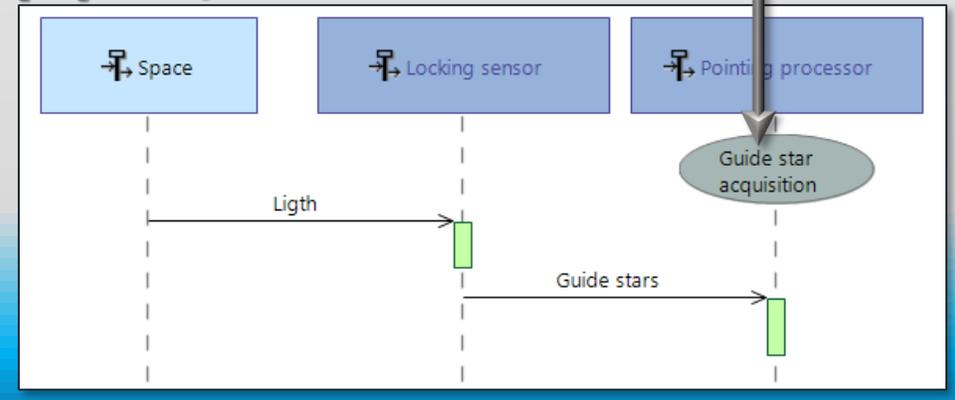


[M&S] Mode&State diagram

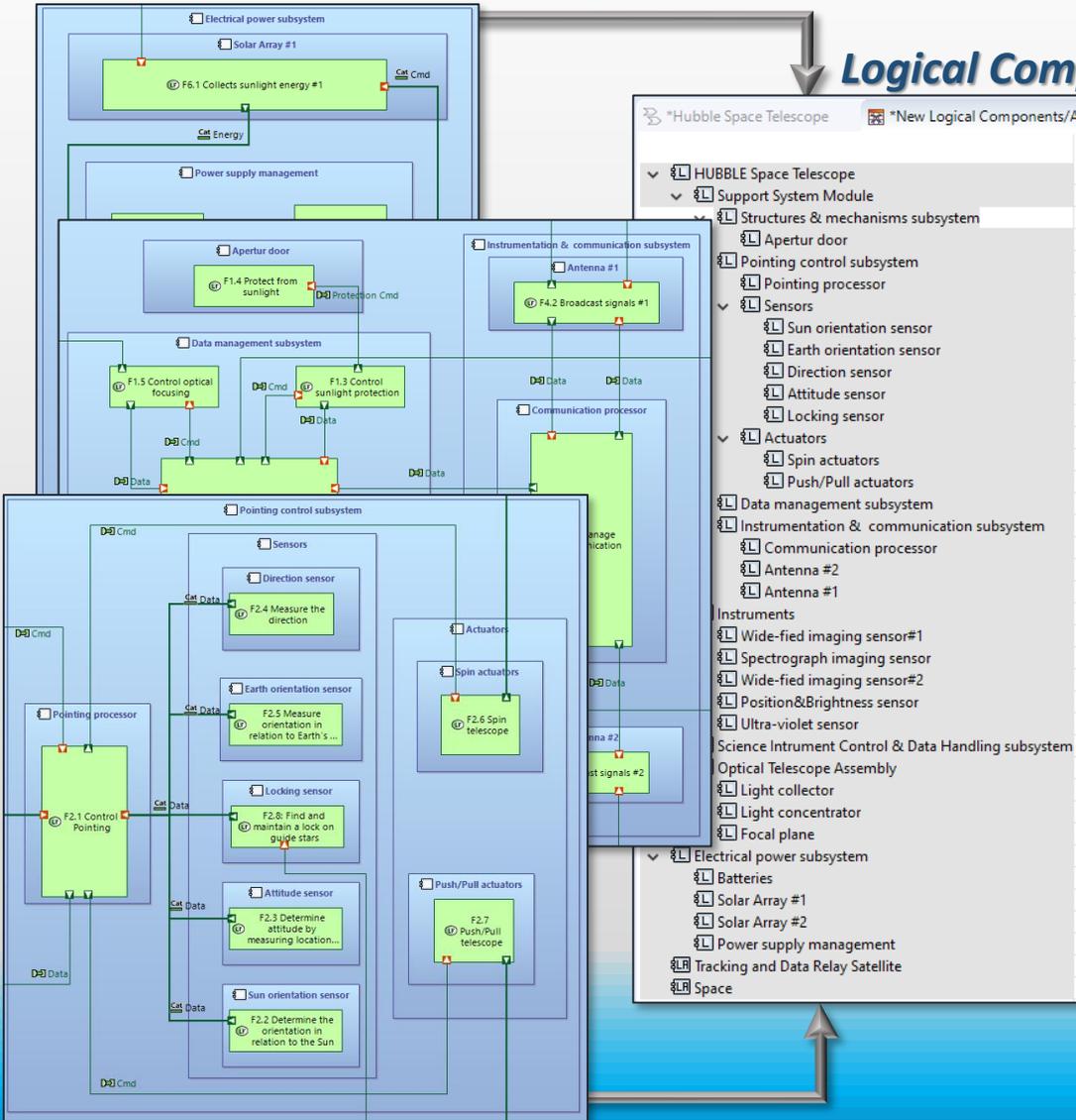
### [ES] Exchange Scenario



### [ES] Exchange Scenario



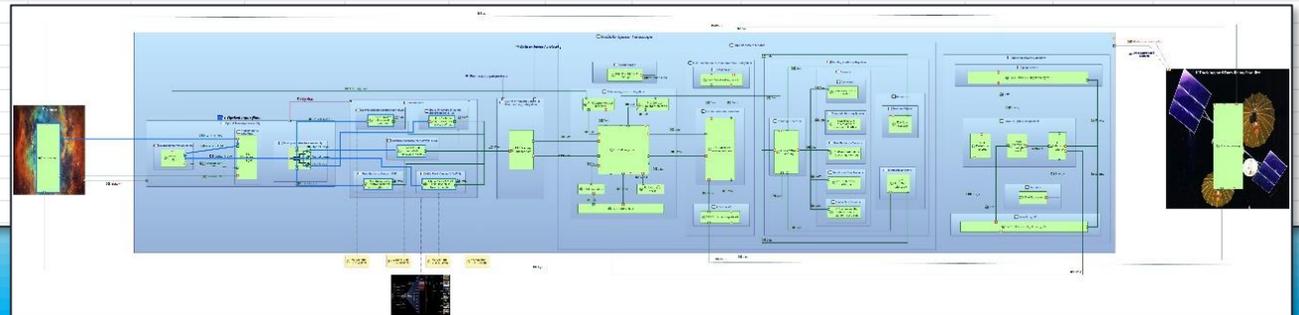
## [LAB] Logical Architecture Diagram



## Logical Components/Logical functions allocation matrix

	LP F1.1 C...	LP F1.2 F...	LP F1.4 Pro...	LP F1.3 C...	LP F1.5 C...	LP F1.6 ...	LP F5.1...	LP F5.1.1 ...	LP F5.2.1 ...	LP
*Hubble Space Telescope										
Support System Module										
Structures & mechanisms subsystem										
Apertur door										
Pointing control subsystem										
Pointing processor										
Sensors										
Sun orientation sensor										
Earth orientation sensor										
Direction sensor										
Attitude sensor										
Locking sensor										
Actuators										
Spin actuators										
Push/Pull actuators										
Data management subsystem						X				
Instrumentation & communication subsystem										
Communication processor										
Antenna #2										
Antenna #1										
Instruments										
Wide-fied imaging sensor#1								X		
Spectrograph imaging sensor									X	
Wide-fied imaging sensor#2										X
Position&Brightness sensor										
Ultra-violet sensor										
Science Inrument Control & Data Handling subsystem										X
Optical Telescope Assembly										
Light collector										
Light concentrator										
Focal plane										
Electrical power subsystem										
Batteries										
Solar Array #1										
Solar Array #2										
Power supply management										
Tracking and Data Relay Satellite										
Space										

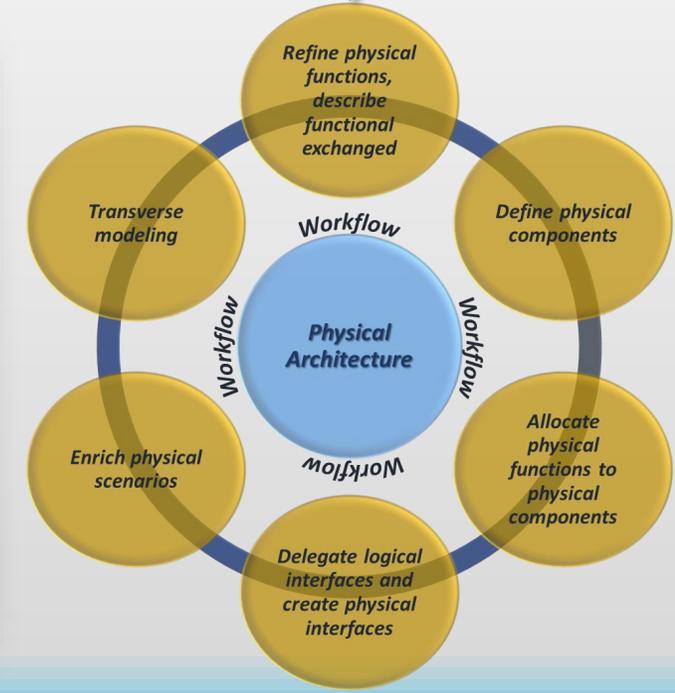
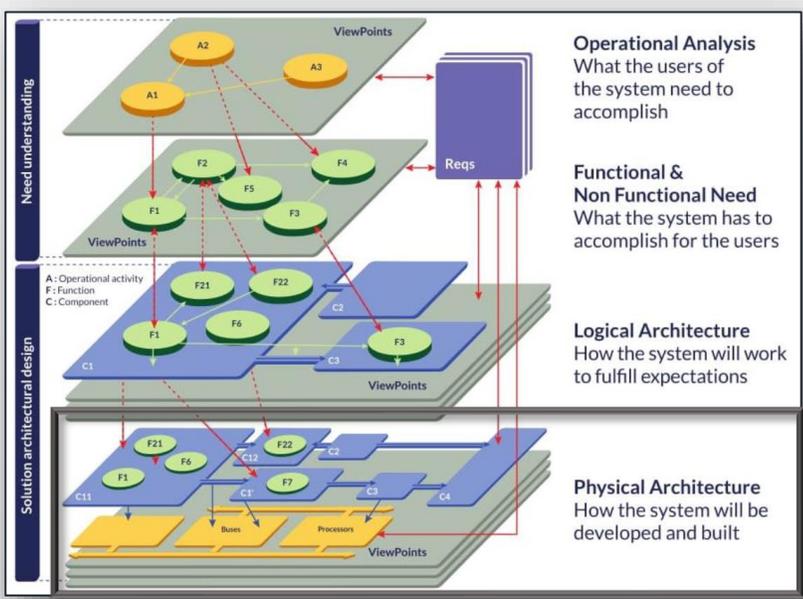
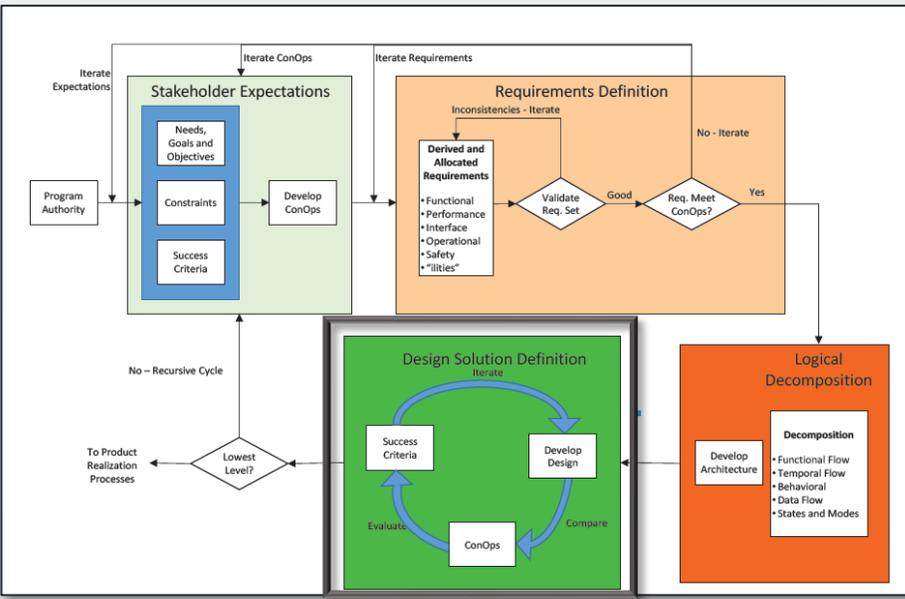
- State Machine and Capability Function Matrix
- Logical Functions - Requirements
- Logical Components - Requirements
- Logical Functions - System Functions
- Logical Components/Actors - Logical Functions
- Logical Architecture Requirement Refinements
- Logical Interfaces - System Interfaces
- Logical Actors - System Actors
- Interfaces - Capabilities
- Interfaces - Capabilities and Scenarios



# Physical Architecture

**Requirements definition process**  
**Technical solution definition process**

**Need understanding**  
**Solution architectural design**



## Activity explorer

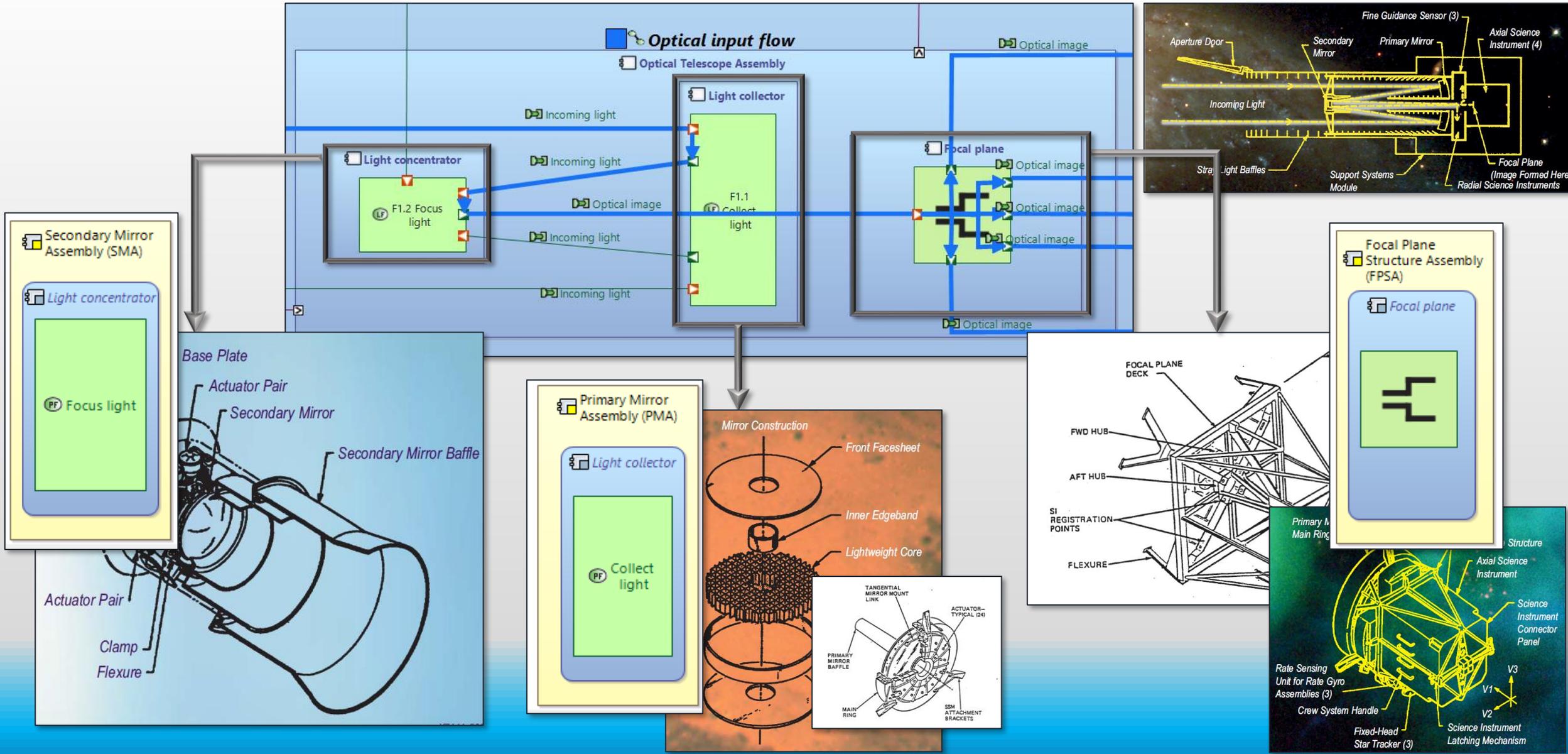
The screenshot shows the Capella Project Explorer interface. On the left, a tree view lists project files. The main area displays the 'Workflow of Hubble Space Telescope' with a list of activities: Operational Analysis, System Analysis, Logical Architecture, Physical Architecture (highlighted with a red box), and EPBS. Each activity has a brief description of its purpose.

## Activity explorer

This screenshot shows a detailed view of the Physical Architecture activity explorer. It lists several tasks under the heading 'Transition from Logical Functions', including 'Perform an automated transition of Logical Functions', 'Create Traceability Matrix', and 'Refine Physical Functions, describe Functional Exchanges'. There are also tasks for 'Define Physical Components and Actors, Manage deployments' and 'Allocate Physical Functions to Physical Components'.

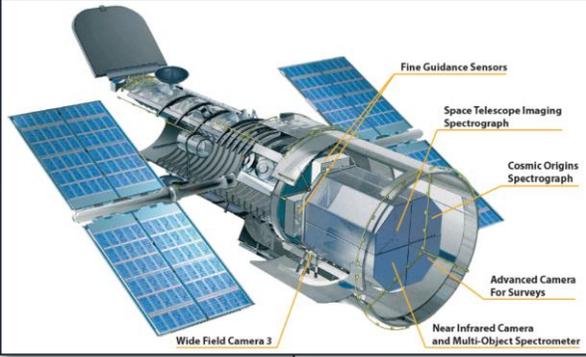
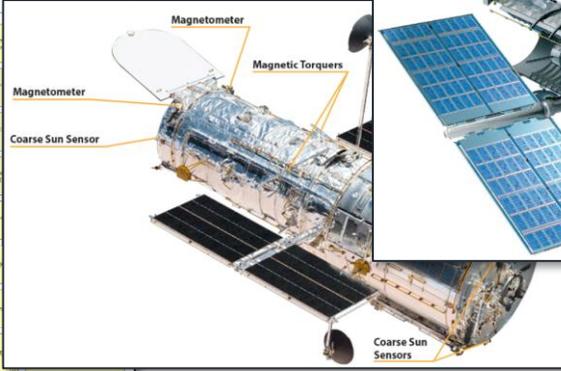
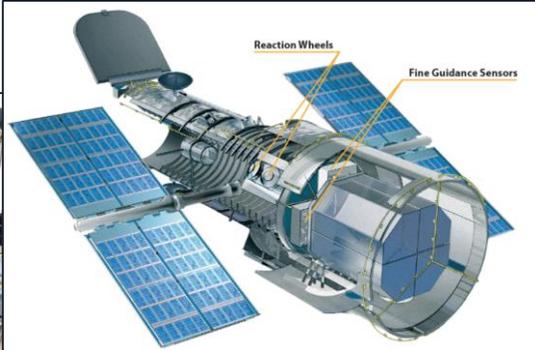
## Diagram editor

The screenshot displays the Diagram editor for the Hubble Space Telescope's physical architecture. It features a grid of component boxes, each with a name and a small icon. The components are organized into a structured layout. A right-hand pane shows a 'Components' list with a tree view of the hierarchy, including 'Node PC', 'Contained in', and 'Applied Property Value Groups'.





## [PCBD] Physical Component Breakdown diagram



### Actuators

- Magnetic troquers
  - Push/Pull actuators
    - F2.7: Push/Pull telescope
- Reaction Wheels Assembly (RWA)
  - Spin actuators
    - F2.6: Spin telescope

### Sensors

- Fixed Head Star Tracker (FHST)
  - Attitude sensor
    - F2.3: Determine attitude by measuring locations & brightness of stars
- Fine Guidance Sensor (FGS)
  - Locking sensor
    - F2.8: find and maintain a lock on guide stars
- Coarse Sun Sensor (CSS)
  - Sun orientation sensor
    - F2.2: Determine the orientation in relation to the Sun
- Magnetic Sensing System (MSS)
  - Earth orientation sensor
    - F2.5: Measure orientation in relation to Earth's magnetic field
- Gyroscope
  - Rotation sensor
    - F2.4: Measure the rotation

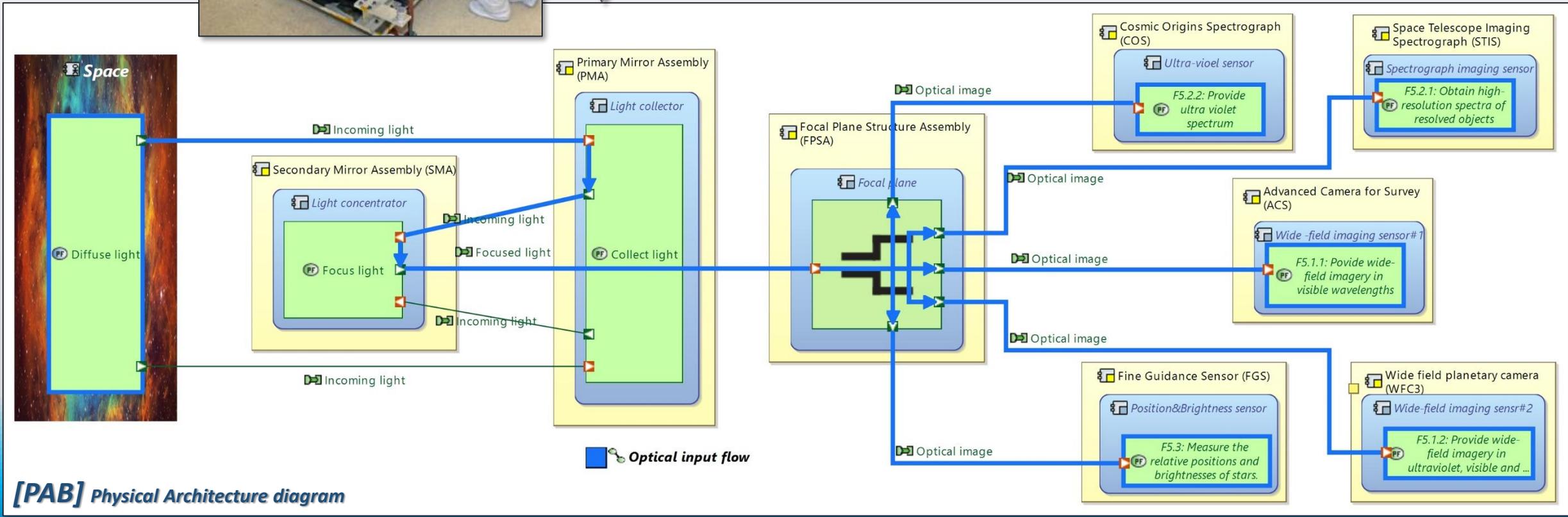
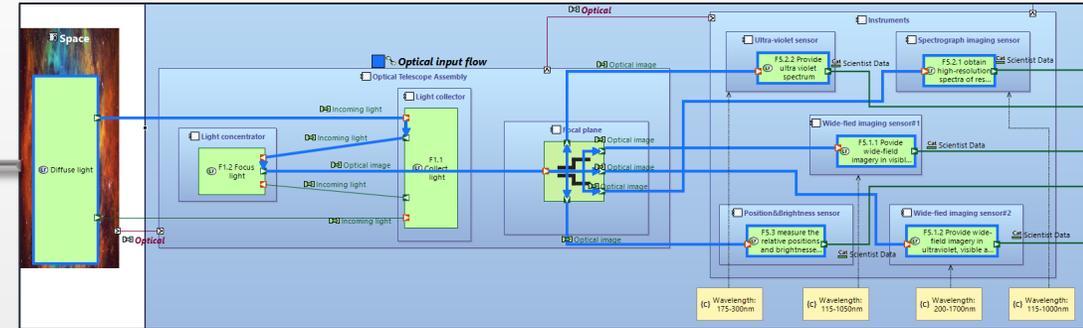
### Instruments

- Space Telescope Imaging Spectrograph (STIS)
  - Spectrograph imaging sensor
    - F5.2.1: Obtain high-resolution spectra of resolved objects
- Advanced Camera for Survey (ACS)
  - Wide-field imaging sensor#1
    - F5.1.1: Provide wide-field imagery in visible wavelengths
- Wide field planetary camera (WFC3)
  - Wide-field imaging sensr#2
    - F5.1.2: Provide wide-field imagery in ultraviolet, visible and infrared light
- Cosmic Origins Spectrograph (COS)
  - Ultra-violet sensor
    - F5.2.2: Provide ultra violet spectrum
- Fine Guidance Sensor (FGS)
  - Position & Brightness sensor
    - F5.3: measure the relative positions and brightnesses of stars.



From [LAB]  
To [PAB]

## [LAB] Logical Architecture Diagram

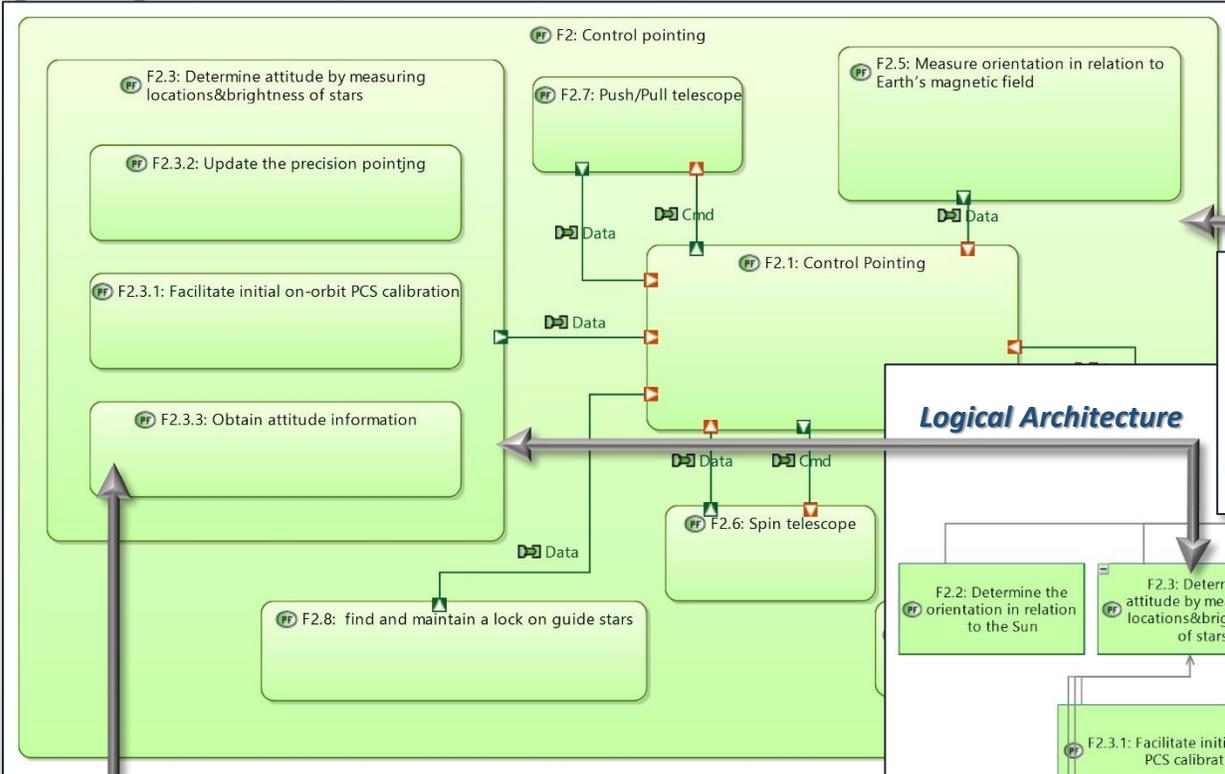


## [PAB] Physical Architecture diagram

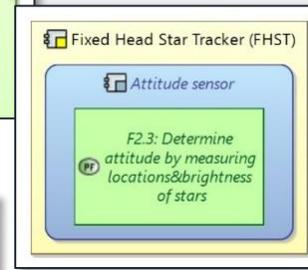


The Fixed Head Star Tracker (“Determine attitude by measuring location&brightness of stars”) function) is a sensitive, electro-optical detector which has the capability of locating and tracking a target star within its Field Of View

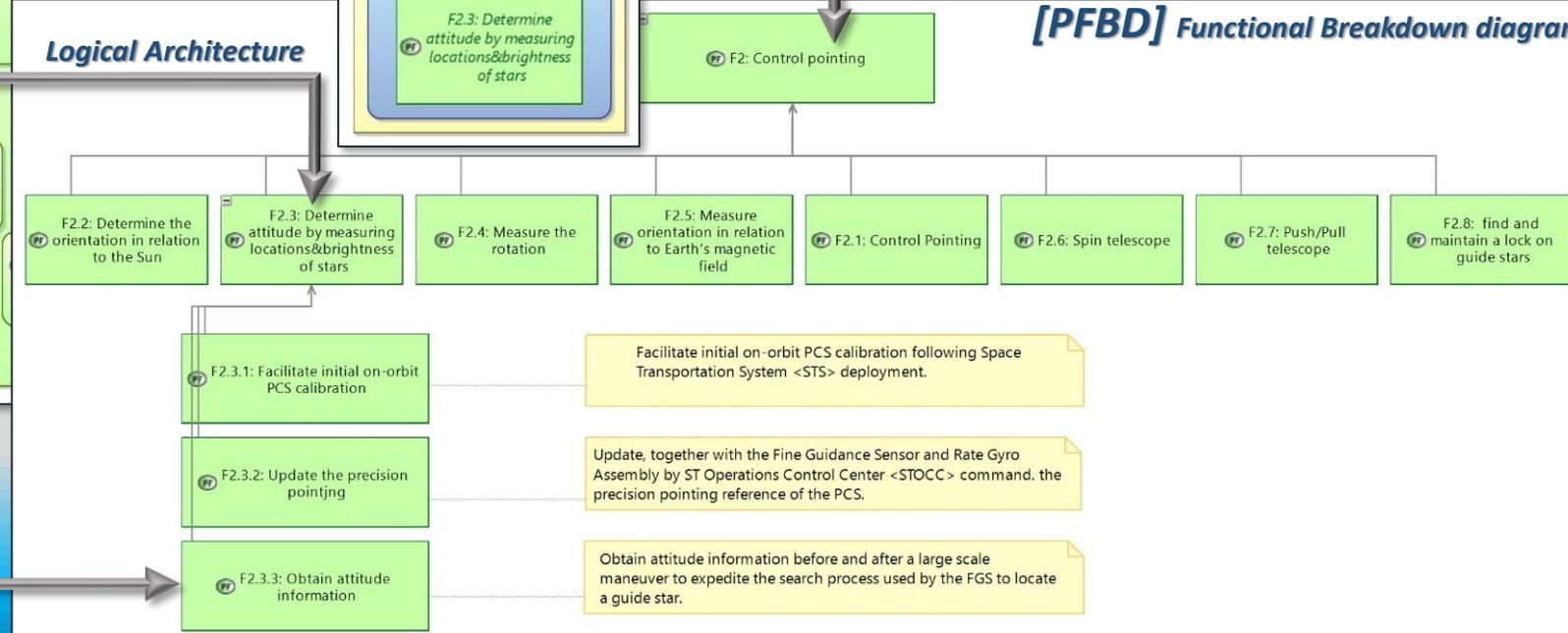
[PDFB] Physical Dataflow Blank diagram



System Analysis



[PFBD] Functional Breakdown diagram



Physical Architecture

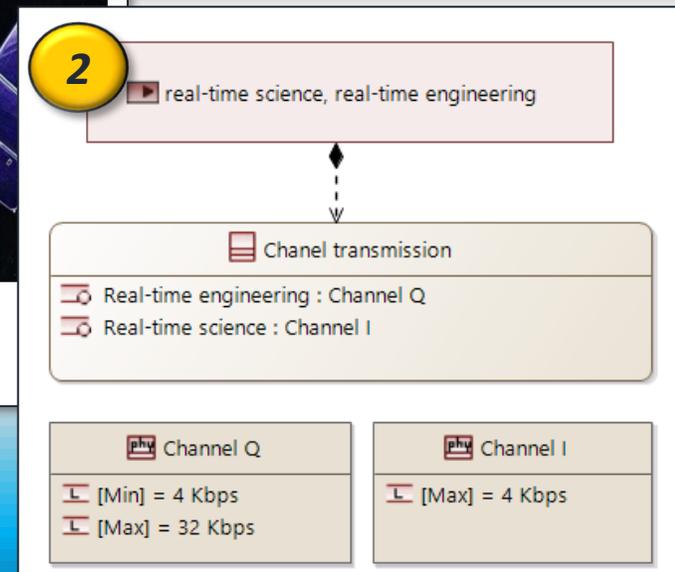
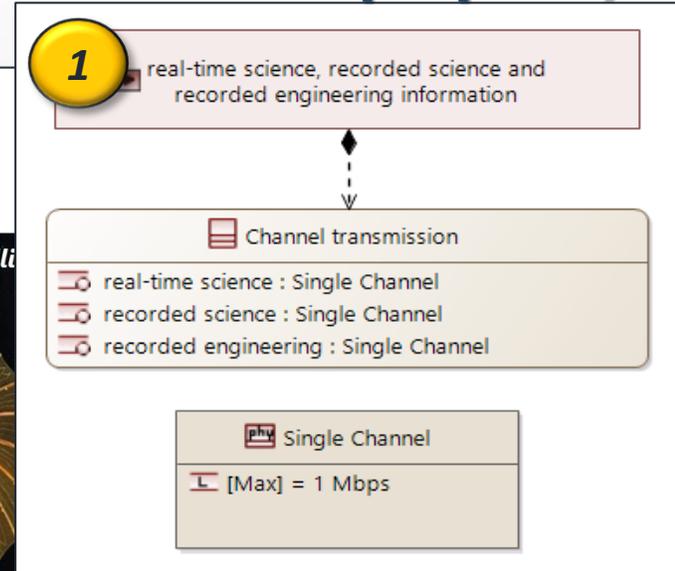
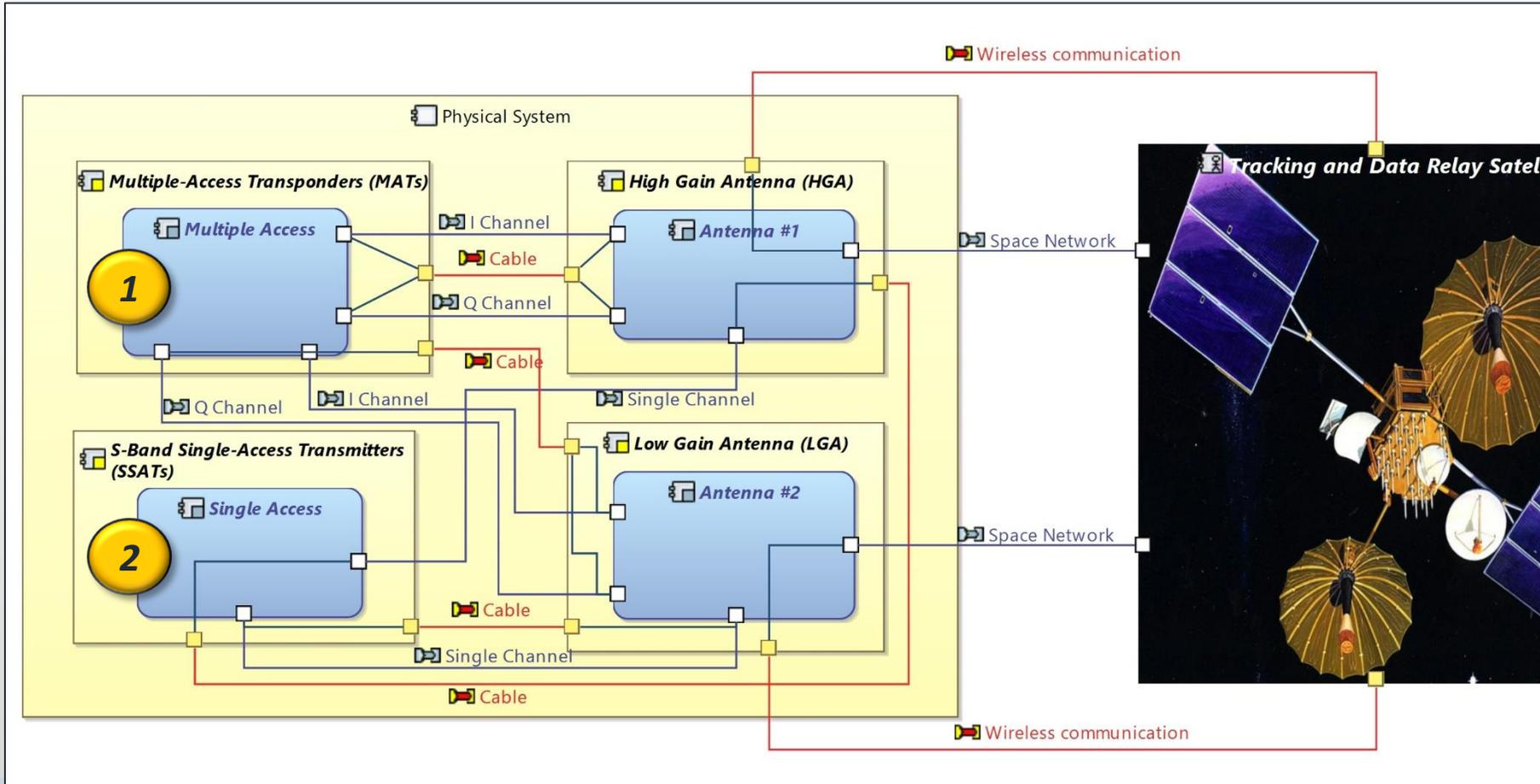
Logical Architecture



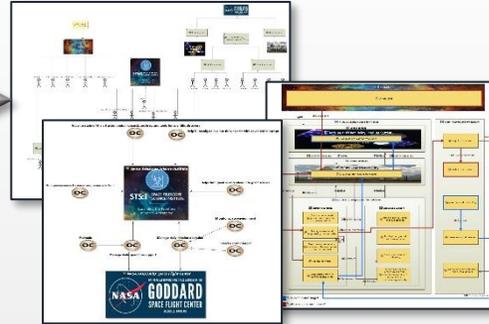


[CDB] Class diagram

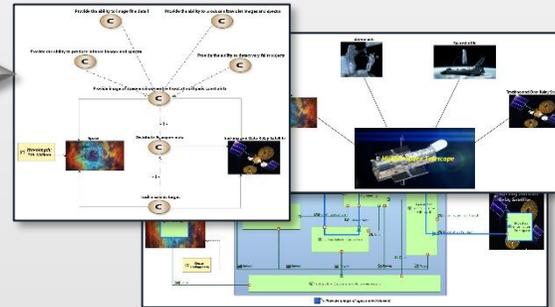
[PAB] Physical Architecture diagram



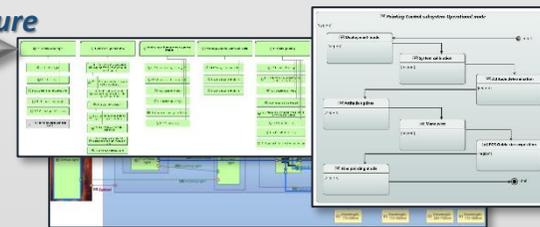
## Operational Analysis



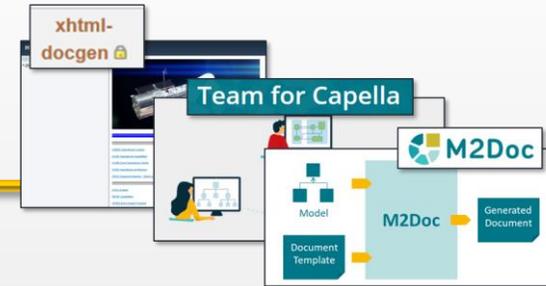
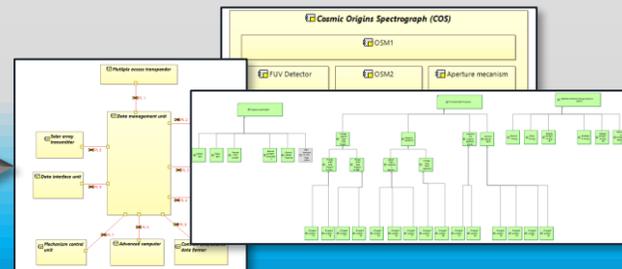
## System Analysis



## Logical Architecture



## Physical Architecture



Subsystem engineer

System engineer

Test engineer

Integration engineer

