

THALES

Building a future we can all trust



Introducing MBSE in an organization successfully

Capella days, November 14th, 2023

www.thalesgroup.com



Introduction

■ Key question addressed:

What was needed to successfully introduce MBSE in our way of working?

- What lessons did we learn from our first attempt to introduce MBSE
- How we turned those lessons into a new MBSE way of working



Erik Visser

erik.visser@nl.thalesgroup.com

About our organization

Member of the Thales Group

- Over 77.000 employees in 68 countries

Thales SIX, Huizen, The Netherlands

- Secure Information and Communication Systems (SIX)
- Sales, Engineering and Manufacturing of communication products
- Working on the SOTAS product family



THALES
Building a future we can all trust

SOTAS capabilities overview

Voice communication

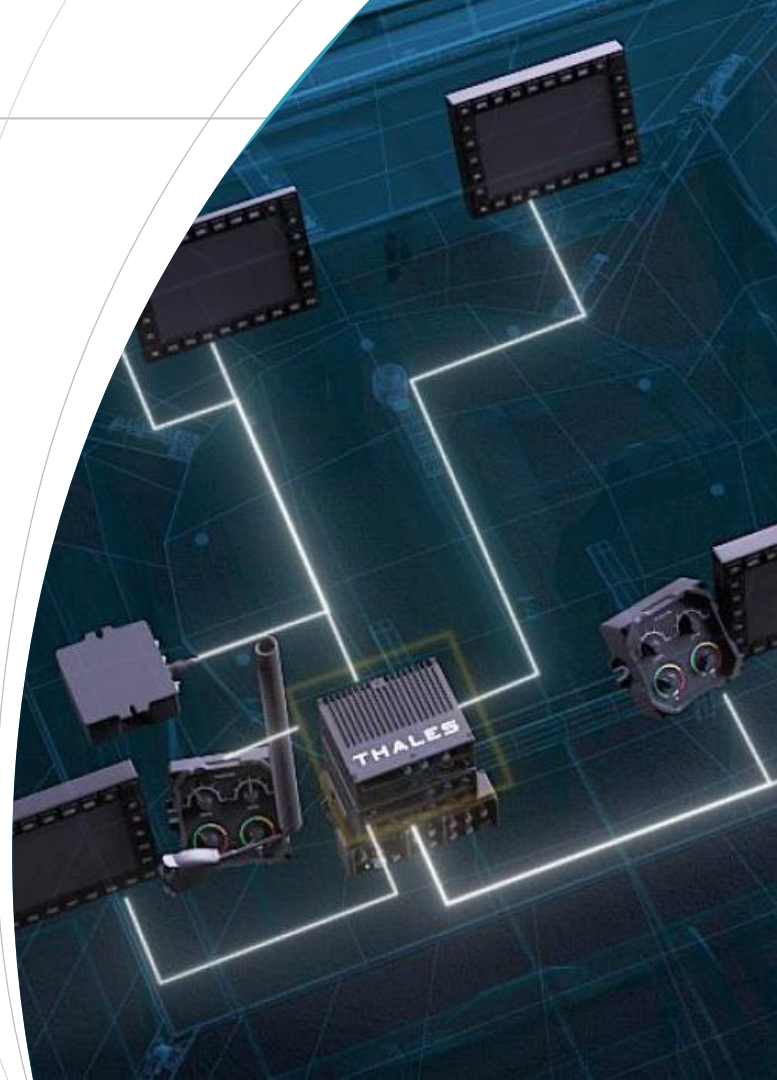
- Local communication (intra-system)
- Connectivity to VoIP infrastructure (inter-system)

Networking

- LAN/Ethernet networking
- IP-routing and services

Application Hosting

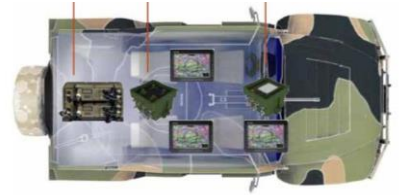
- Provide processing environment
- Includes (remote desktop) application control



SOTAS product family

Family of modular building blocks for building a solution

- Can be configured and scaled for all types of use cases and platforms
- Deployed in armoured vehicles, but also used in civil environments
- Designed to withstands harsh and challenging conditions (military grade)
- Embedded hardware and software development



Our engineering department

About 60 FTEs, divided into:

> System Engineering

System requirements engineers and system architects

> Hardware Engineering

Hardware architects and developers

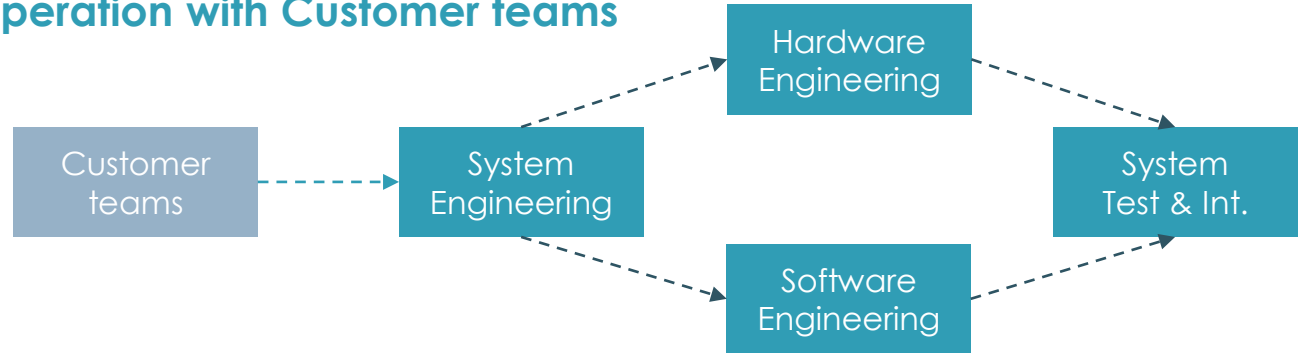
> Software Engineering

Software architects and developers

> System Test & Integration

IVVQ manager and system testers

Working in close cooperation with Customer teams





Our first attempt at introducing MBSE

And the lessons we learned

Our first attempt in introducing MBSE

12 years ago, we were launching a new revision of the SOTAS product family

- As successor of an earlier SOTAS product family, with updated technology
- Requiring significant development

Learned about Model Based System Engineering and Arcadia

- Expected this magic of modelling to help us be more efficient and effective

So, we jumped right in!

- The system engineers and lead architects took the Arcadia/Capella training
- And started using models to help do our requirements engineering jobs

But it didn't help us to be more efficient or effective

Time passed...

- Two years ago, another major development was about to start
- Time to look back and contemplate: Why didn't MBSE help us earlier?
 - What lessons have we learned?



In hindsight – pitfalls of our first attempt



Individuals working on models on their own

And just have the stakeholders review them



Not knowing when to stop modelling

Spending a lot of time modelling lots of detail



Using separate models for different aspects

But forgetting about the commonality and relationship between them



Skipping the Operational Analysis

“We know what the customer needs because we know the existing products”



Using a waterfall engineering approach

Every model had to be complete and correct at first iteration



No clear vision on how to use what we modeled

We continued writing traditional text-based requirements documents

In hindsight – pitfalls of our first attempt



Individuals working on models on their own

And just have the stakeholders review them



Not knowing when to stop modeling

Spending a lot of time modeling



Using separate models

But for each model, we didn't have a clear communication between them



Skipping modeling

"We knew we were going to build a new product because we know the existing products"



Using an old modeling approach

Every model was not complete and correct at first iteration



No clear vision on how to use what we modeled

We continued writing traditional text-based requirements documents

We didn't have a plan

But above all and foremost – it's about the people!

Realized it isn't MBSE that

- defines *the right product* for the customer
- provides *good architectures*
- makes sure we develop *the right product*
- makes sure we develop the product *in the right way*

We do all of that!



And we don't do it on our own – we do it together!

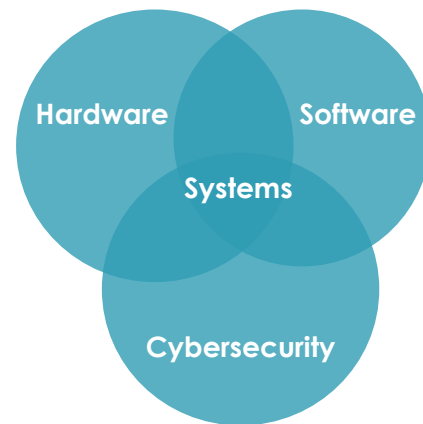
Big products



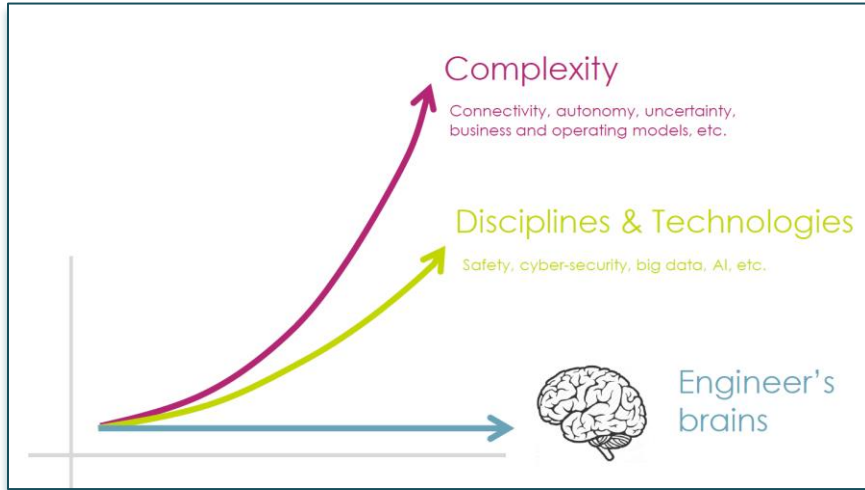
Multiple teams



Multiple disciplines



Increasing complexity and disciplines



How to get those brains to work together effectively and efficiently?

- By creating a common language to define a common understanding
- And use MBSE to help achieve that

Most important lesson learned

■ Modelling is not about the model

It's about getting
a common understanding!



Modelling is a team job

- By working on the model together, we get a better understanding
- Ensure that working on our models is integrated into the daily work-process



Modelling is a continuous job

■ The insights we gain today, help us improve the model tomorrow

■ Accepting that it is a continuous job helps to get today's job done

- It doesn't have to be complete and perfect today – it just has to serve today's goal
- We don't have to worry (too much) about tomorrow



Share the output of modelling

Our modelling gives us a common understanding

- That understanding is to be shared in the organization

Define how the result of modelling is shared...

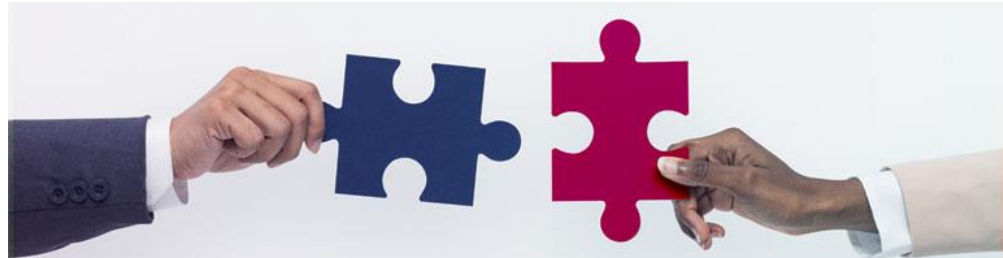
- E.g. by using diagrams in our documents, by exporting the models themselves

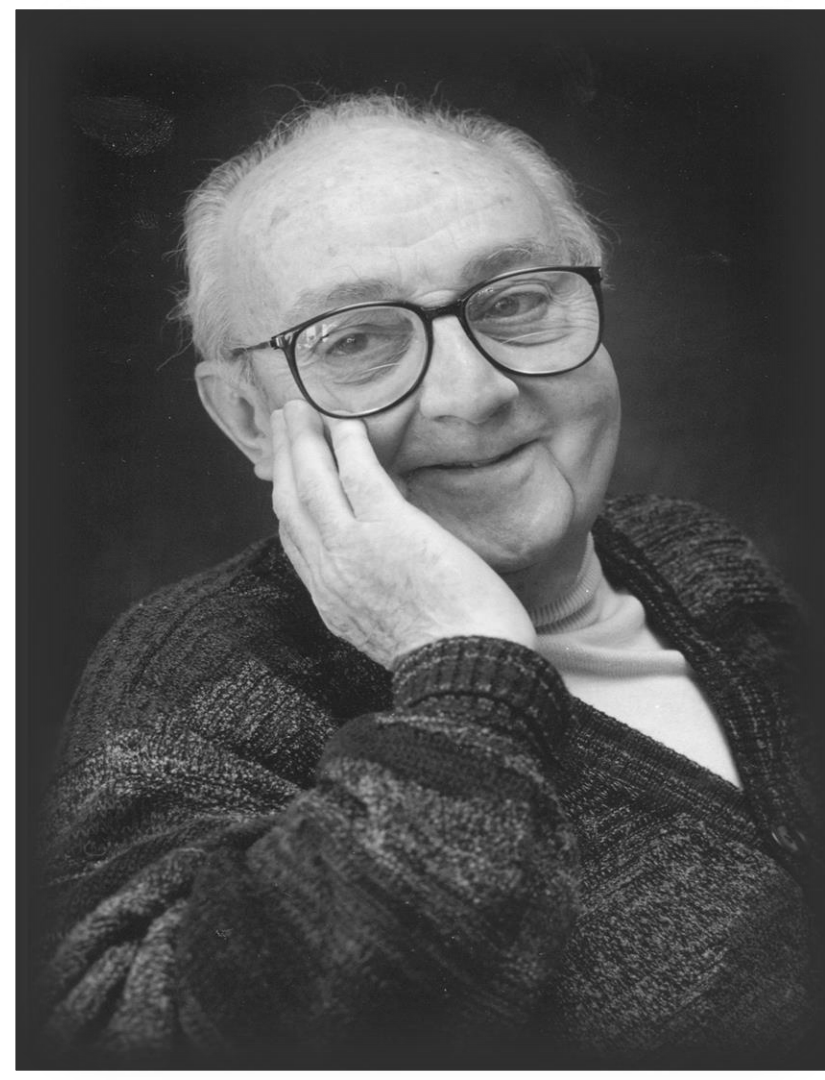
... which part is shared ...

- E.g. which diagrams are used? Which descriptions are used?

... and how

- E.g. in which documents will those diagrams or descriptions be included? And for what purpose?





All models are wrong
but some are useful

George E.P. Box

A fresh start !

Based on these lessons,
we reconsidered our way of working

➤ How can we be effective and efficient in our
product line engineering?

And we started by defining our vision...





SOTAS Engineering Vision

We share a **common language**
that we use to create a **common understanding** of
what SOTAS provides
why it does so, and
how that is realized

This common understanding is
continuously updated,
easily accessible and
used throughout the organization

And set the objectives

Objectives for our new engineering process:

- **Create a common understanding of the SOTAS product family**
- **Efficiently address the variability and commonality between the products**
 - Multiple variants of a product must comply to the same or similar requirements
 - Multiple variants of a product will be built using common components
- **Support the jobs of the stakeholders involved:**
 - Customer team, System Engineer team, Hardware and Software teams, System Test & Integration team

And use modelling to reach those objectives

Because modelling helps us:

- To express ourselves in more than just words
- To be consistent in the way we present our information
- To help identify what we overlooked
- To share knowledge and transfer work



And use Arcadia to structure our approach

OA: What our customers are trying to achieve

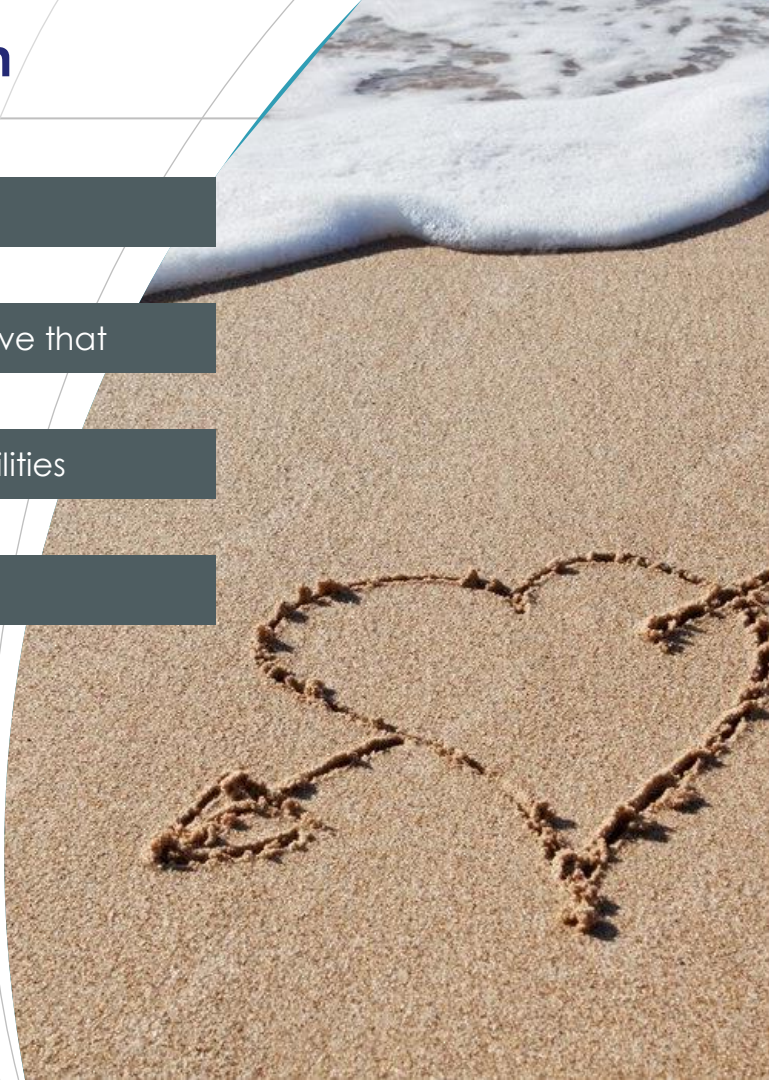
SA: What capabilities our products provide to help them achieve that

LA: What our products need to do to facilitate those capabilities

PA: How that behavior is realized

Arcadia clearly separates these various layers of concern

➤ Which facilitates addressing them separately



Navigating over the layers of Arcadia



■ Navigating over the layers, using What and Why questions:

- Facilitates assessing the consistency and completeness



Putting MBSE into practice!

The SOTAS Engineering Framework – hierarchy of models

Solution Model

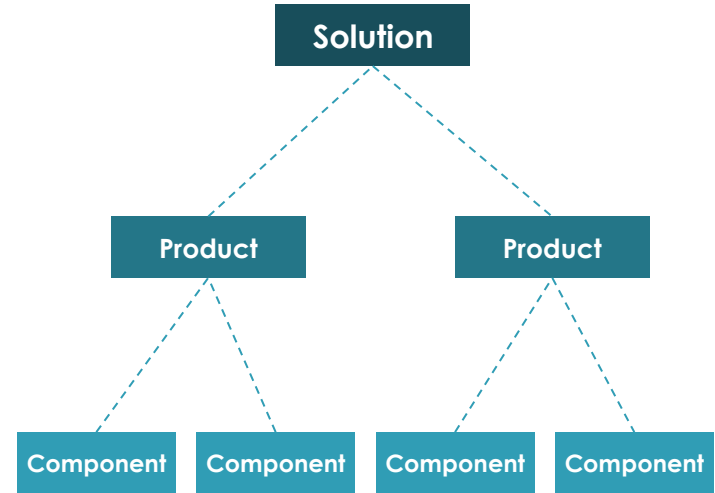
- Defines common language
- Defines the product line reference solution architecture

Product architecture models

- Decomposition of a product into HW and/or SW components
- Emphasis on role and responsibility of those components

Component model

- Decomposition of a SW component into SW modules
- Emphasis on role and responsibility of the SW modules



The Solution model

Overall objective:

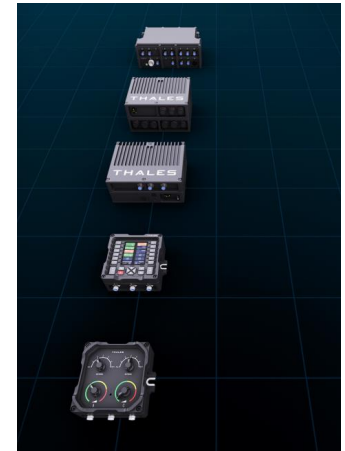
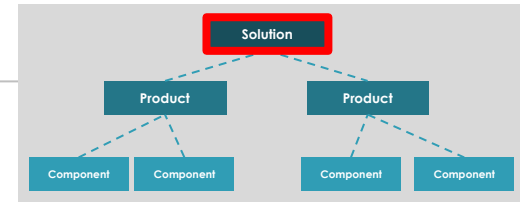
- Define our common understanding/common language
- Support defining family of products
 - Efficiently define and reuse product requirements

Operational Analysis:

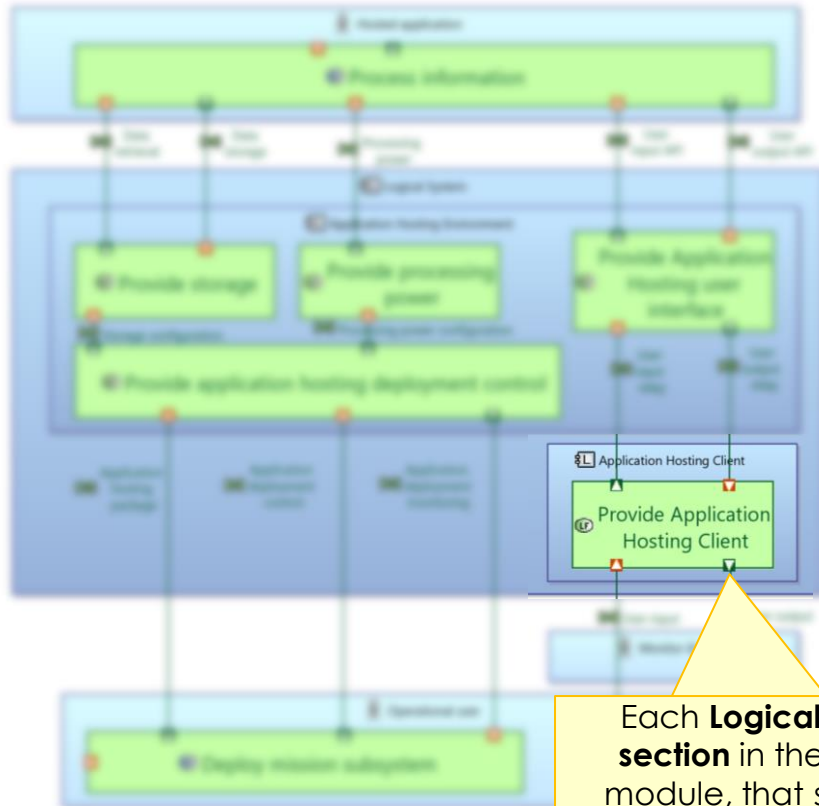
- Define the operational need of the customer:
 - What are the problems they face?
 - What do they try to accomplish?

System Analysis

- What does the SOTAS solution provide to help the customer in accomplishing their goals
 - Setting the scope of the SOTAS solution
 - Defining the interaction of the SOTAS solution with its actors

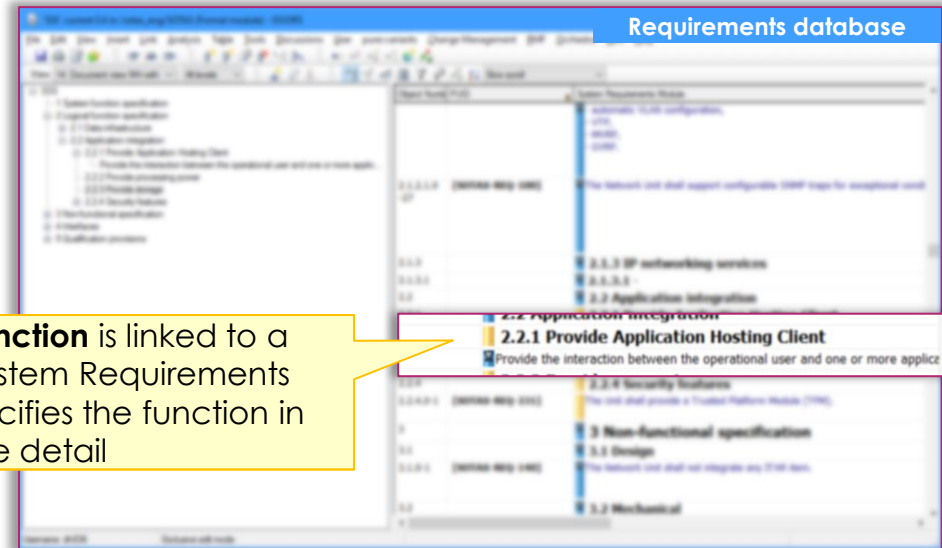


Logical Architecture: What does the SOTAS solution need to do?



The System is decomposed into **Logical components** that each provide one or more **Logical functions**

A **Requirements database** is used to capture the superset of product requirements



Each **Logical Function** is linked to a **section** in the System Requirements module, that specifies the function in more detail

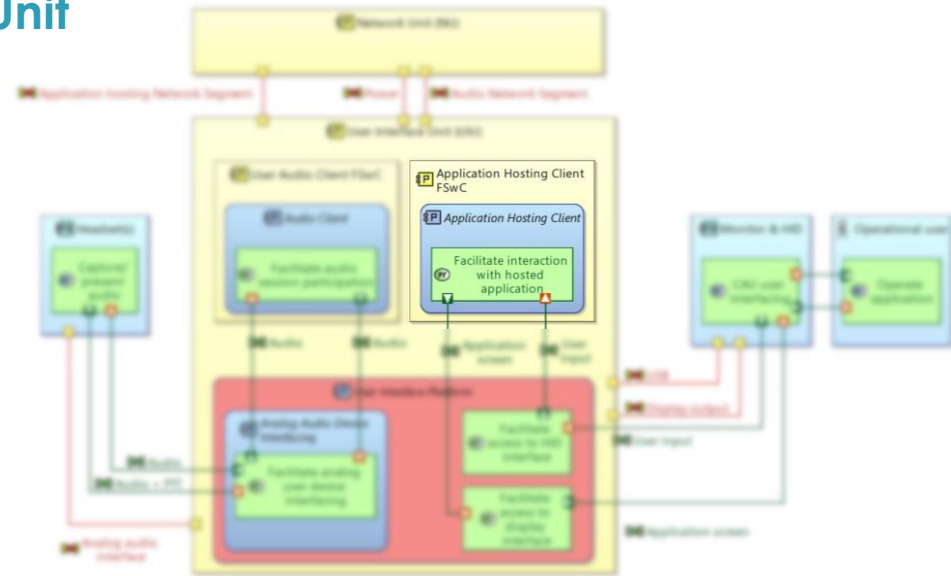
Physical Architecture: Allocation to conceptual products

Allocates components to abstract product types

- E.g. User Interface Unit
 - Hosts **Analog Audio Device Interfacing** component
 - Hosts **Application Hosting Client** component

Blueprint for multiple variants of the Unit

- With/Without Application Hosting
- With variable number of audio interfaces



Feature Model: Product variant definition

A Feature Model based on the Components and Functions from Capella

- A Component from the Solution Model corresponds with a feature
- For each product variant, the applicable features are enabled
 - Enabling a feature implies that the related requirements apply

Feature model

- Application integration
- Application Hosting Client
- Security features

Requirements database

- 2.2 Application integration
- 2.2.1 Provide Application Hosting Client
- Provide the interaction between the operational user and one or more applica...

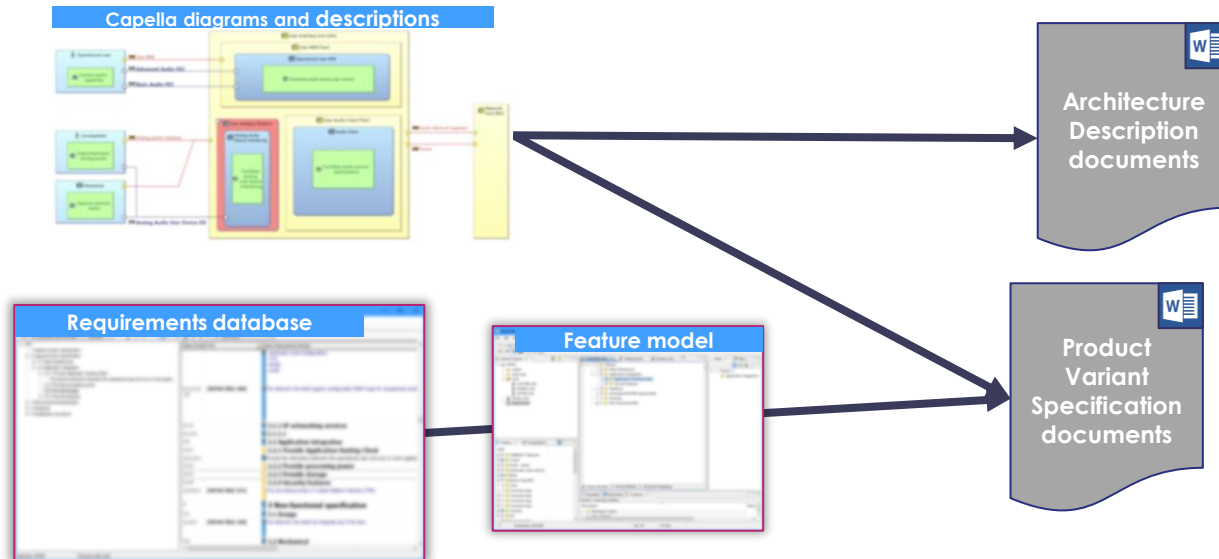
A **feature** in the Feature Model corresponds with a **Component-section** in the **Requirements database**

Enabling a feature in the Feature Model implies that the related **requirements apply** to that product variant

Sharing information: Generating documents at Solution level

Documents as means to share information

- Easily accessible for all stakeholders (w/o Capella tooling experience)
- Guide the reader through the story
- Use multiple sources (Capella diagrams, Requirements DB, Feature Model)



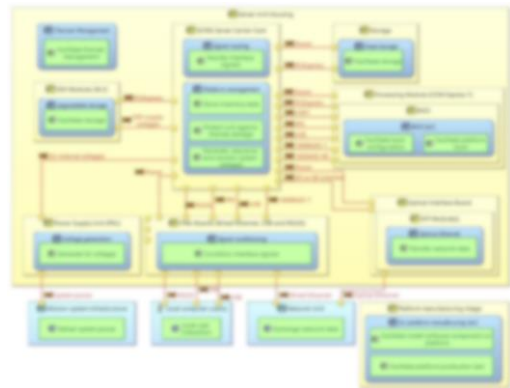
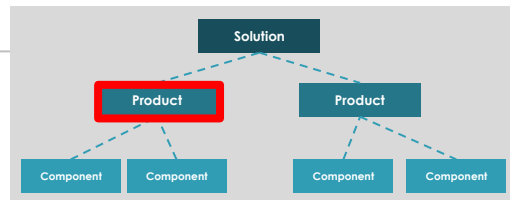
The Product models

Objectives:

- **Decompose a product into HW and/or SW components**
 - e.g. hardware boards, Platform SW, SW application package
 - HW and SW components may be specific to the product or can be generic (used in multiple products)
- **Define role and responsibility of each HW/SW component**
- **Identify interfaces between components**

Used to generate Product Design document

- **Specific design document for each product**
- **Using diagrams and descriptions from the Capella model**



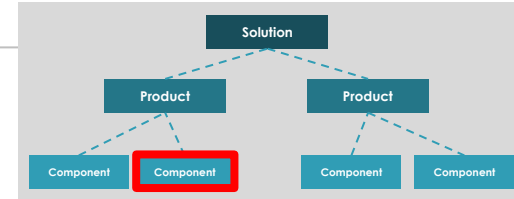
The Software Component models

Objectives:

- **Decompose a Software Component into Software Modules**
 - Where Software Modules (e.g. libraries, executables) may be specific to the Component or can be generic (used in multiple Components)
- **Define role and responsibility of each Software module**
- **Identify interfaces between Software modules**

Used to generate SW Component Design document

- Using diagrams and descriptions from the Capella model



The SOTAS Engineering Framework revisited

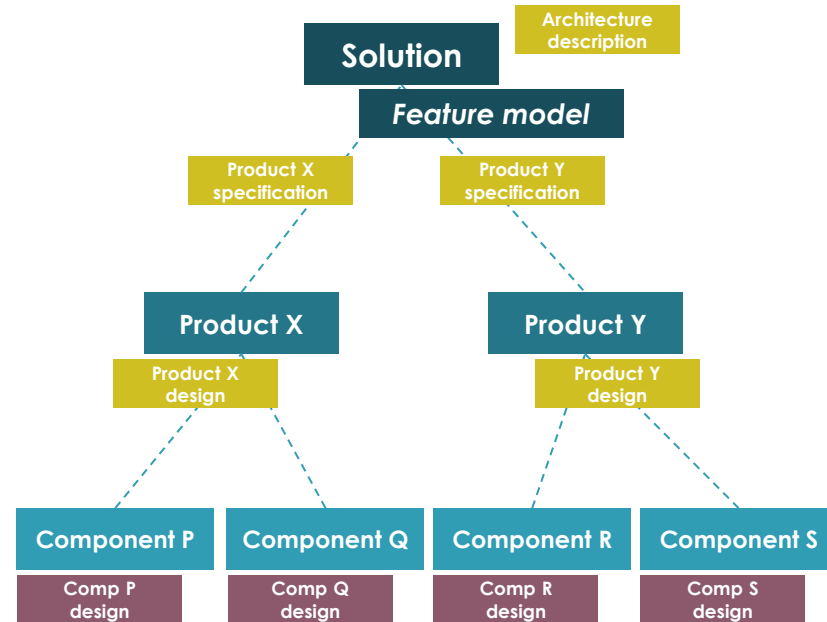
Showing Feature Model and generated documents

Document generation is automated:

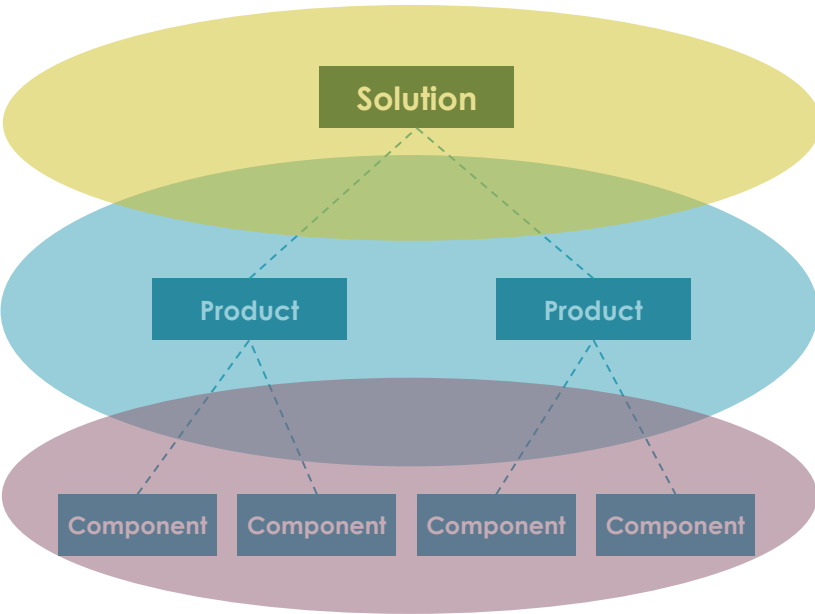
- Eases deployment of changes in the model (which may impact multiple documents)

Nightly generation of documents

- Changes submitted today result in a new “current” revision of the documents tomorrow
- Documents are available at a central location, for use throughout the organization



Teams working on the models together



Solution Engineering team

- Solution/Product Line Architects working on Solution model
- Together with Customer Technical Lead
- In cooperation with Product architects

Product Engineering teams

- Product Architects working on Product model
- Together with Solution Engineering team members
- In cooperation with HW/SW component architects

Hardware and Software Architecture teams

- HW/SW Architects working on Component models
- Together with Product Architects
- In cooperation with HW/SW engineers

■ The hierarchy of models aligns with the structure of the organization

And work on them continuously

Working on the models is integrated in our daily work-process

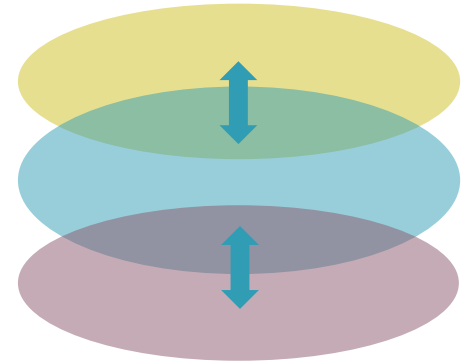
- Teams use weekly work-sessions to discuss and refine (changes to) the models

Transfer of knowledge between the teams is achieved through cooperatively working on models

- Not by passing along documents

WEEKLY PLANNER

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	SUNDAY



In summary: What we did to introduce MBSE successfully



Plan our MBSE

- Define what we want to achieve with modelling
- Define the models, how they relate and their scope



Organize working together

- Build the models together, resulting in a common understanding
- Use and improve our models in joint working sessions
- Have the relevant stakeholders participate



Embed sharing the modelling artefacts

- Use the model to generate specification and design documentation
- Have the up-to-date documentation easily available

Thank you!