

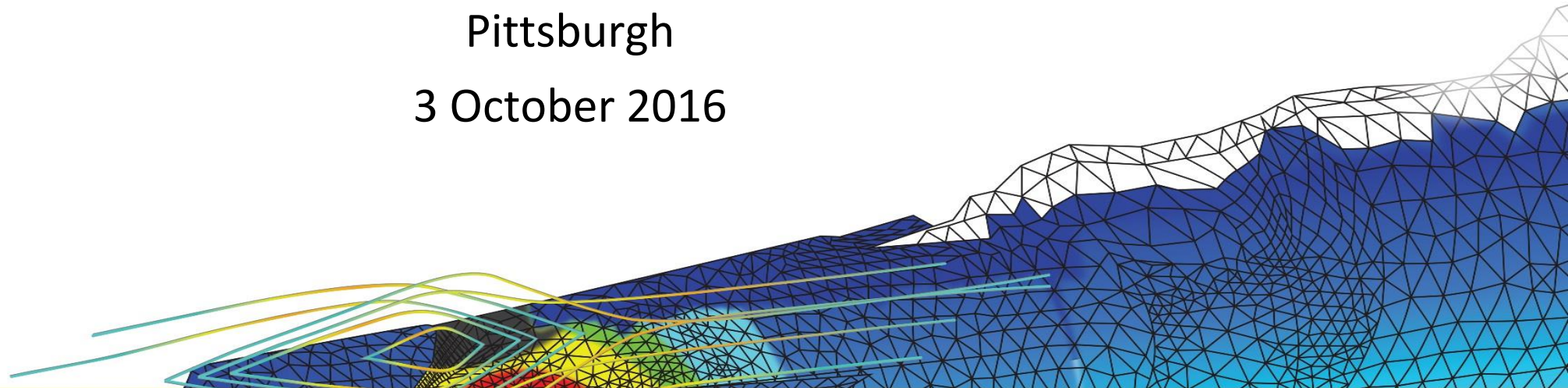


# SCADE AADL Capabilities

AADL standards meeting

Pittsburgh

3 October 2016

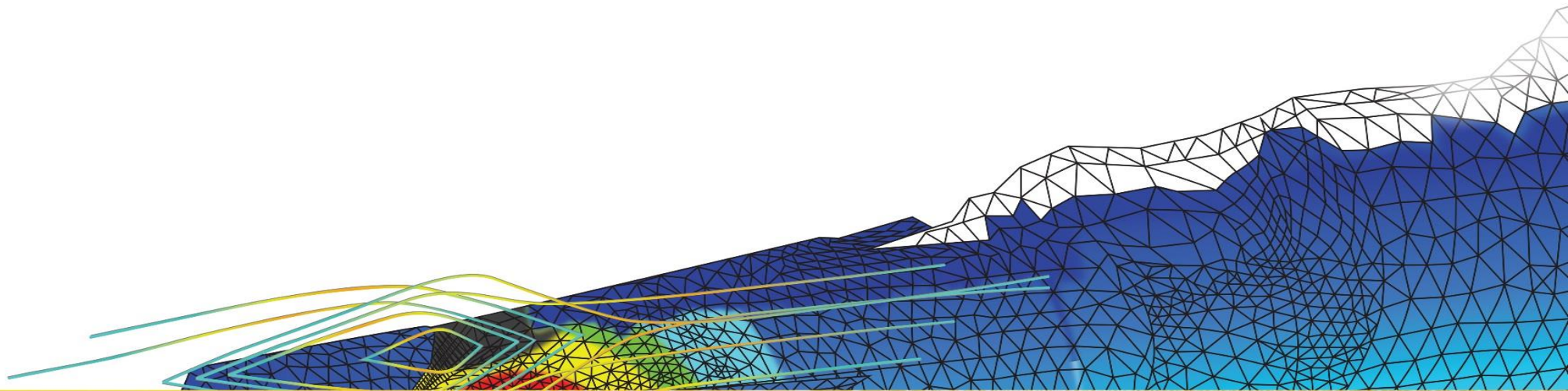


# Agenda

- **ANSYS MBSE Quick Overview**
- **SCADE AADL Integration and Demo**
- **Q&A**

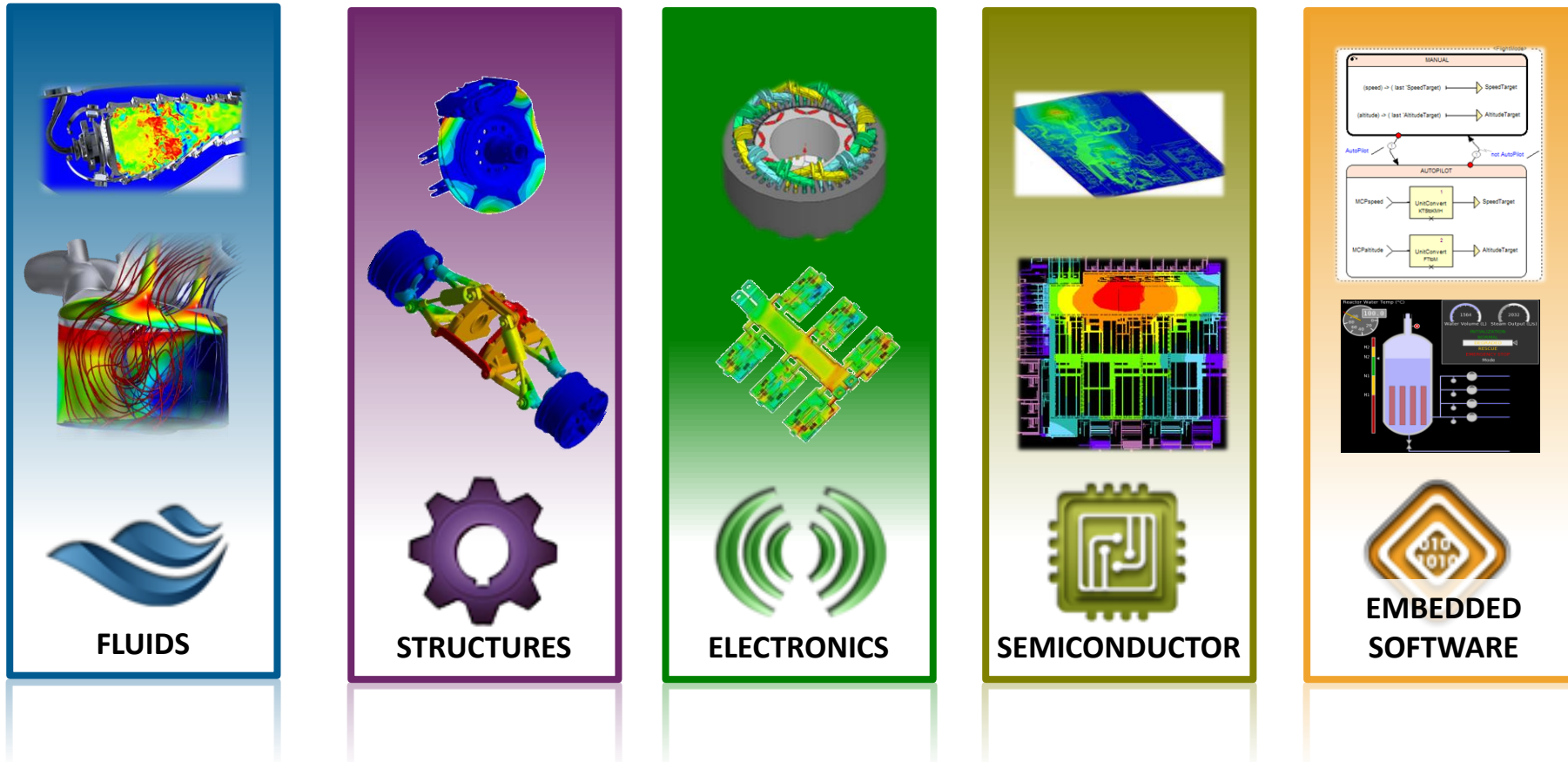


# ANSYS MBSE Quick Overview



# ANSYS Enables Systems

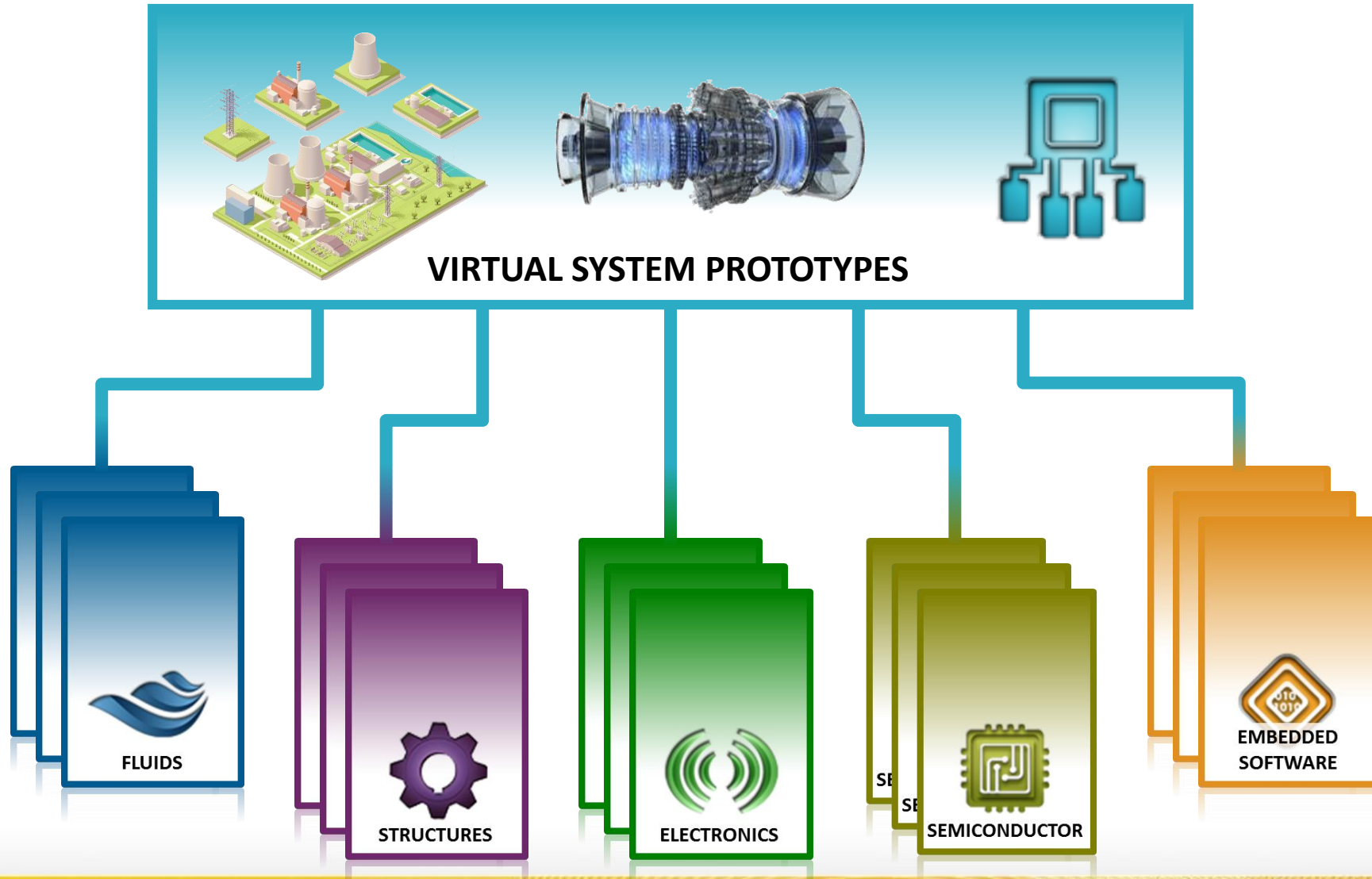
*From Comprehensive Component-Level Design & Simulation ...*





# ANSYS Enables Systems

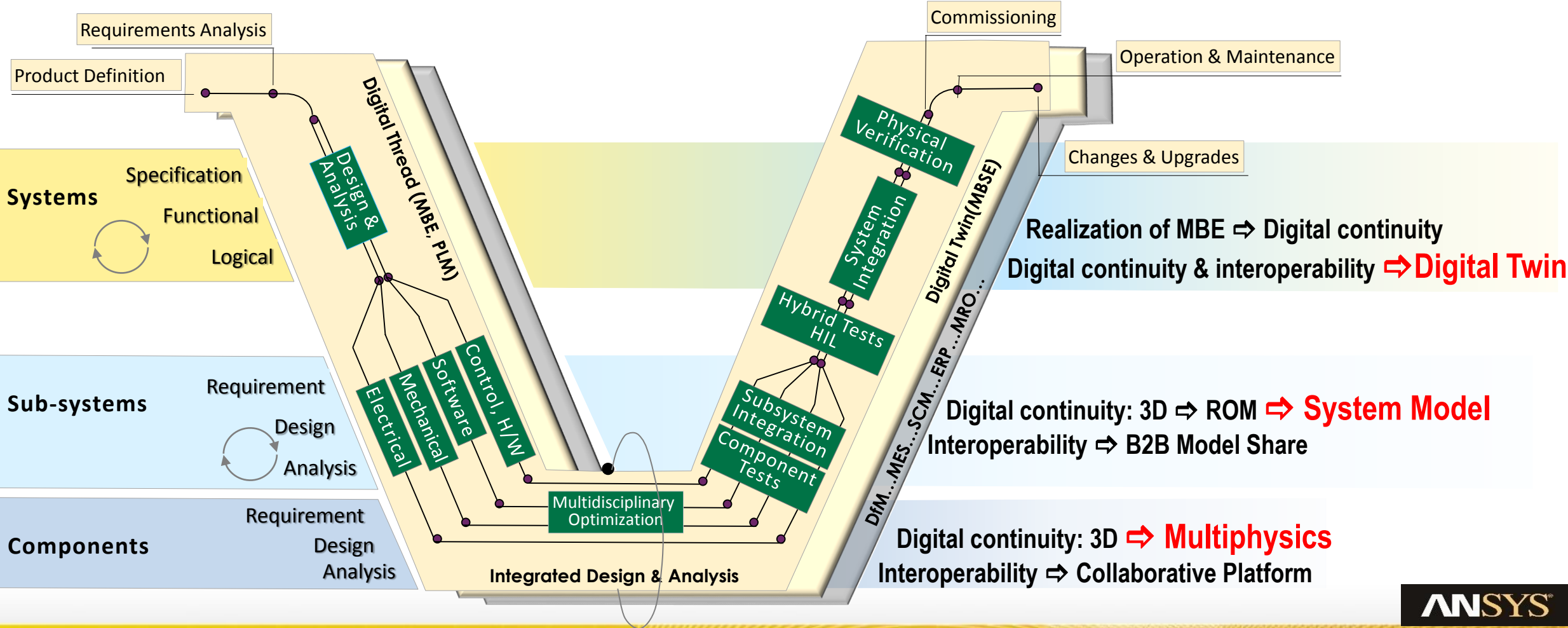
*... To Complete Systems Simulation*



# Model-Based System Engineering: Product Lifecycle

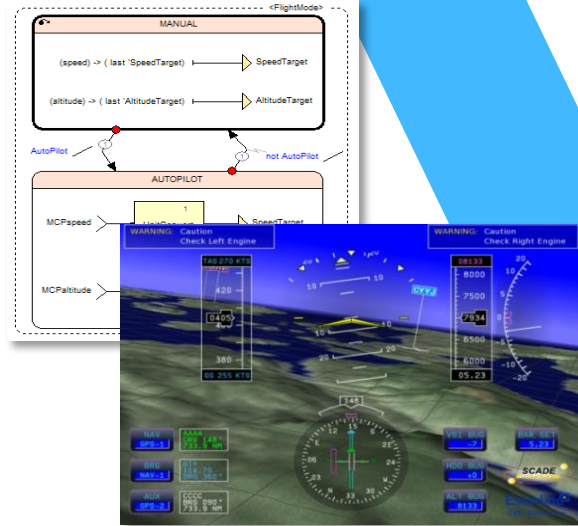
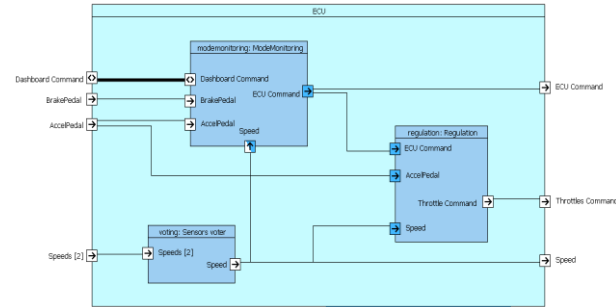
## Evolution of Model Based Engineering

Single Source of Truth Between Design Models, Manufacturing and Operations



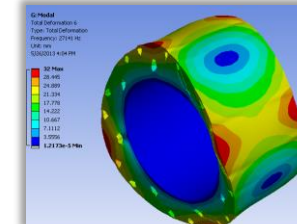
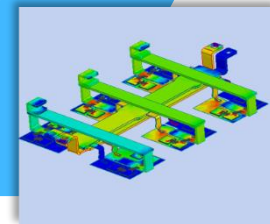
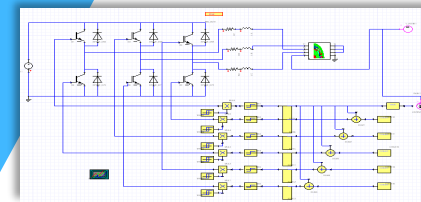
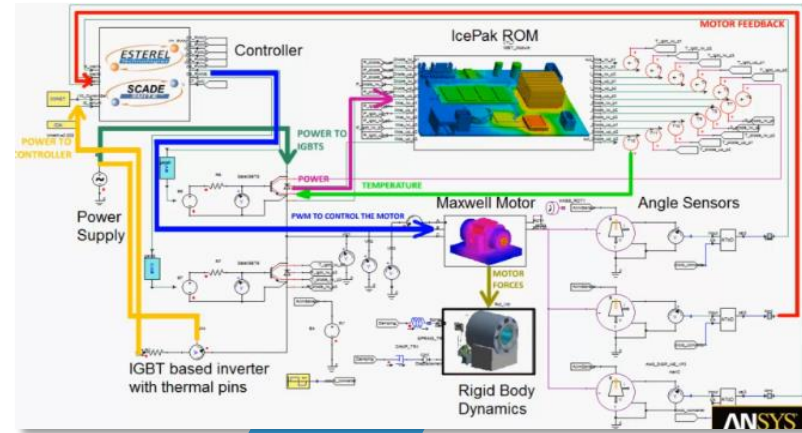
# ANSYS Model-Based Engineering Solutions

## Model-Based Systems Engineering



## Model-Based Software Engineering

## Multi-Physics & System Simulation



## 3D Physical Simulation

# SCADE: A Proven Embedded Software Development Toolset



“Airbus never experienced any bug in flight in our Flight Control System software produced automatically.”

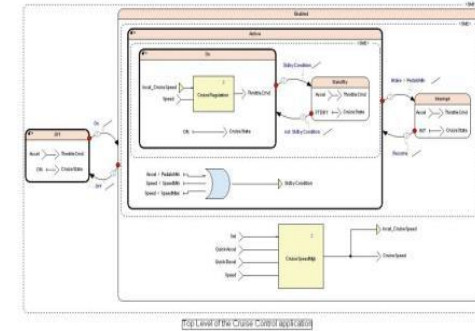
Jean-Charles DALBIN,  
Tools and Methods Director  
Airbus



# SCADE Suite for Embedded Controls Software Design

## Embedded Control Software Design

*Efficient modeling of controls, logic and algorithm designs  
within a single environment*



## Integrated Suite for Prototyping, Modeling, Simulation, Verification, and Optimization

*Efficient debugging and optimization of software models  
and code size, speed and performance*

```
[...]
void Button_ABC_N(inC_Button_ABC_N *inC,
outC_Button_ABC_N *outC)
{
  /* ABC_N::Button::SSM::SSM_SMI_dispatch_sel */
  SSM_Button_SMI_ST SSM_SMI_dispatch_sel;

  if (outC->init)
  {
    outC->init = kcg_false;
    SSM_SMI_dispatch_sel =
      SSM_SMI_Unselected__ABC_N;
  }
  else
  {
    SSM_SMI_dispatch_sel = outC->M_pre_;
  }
  switch (SSM_SMI_dispatch_sel) {
  case SSM_SMI_Locked__ABC_N:
    outC->foreground = white_ABC_N;
    outC->background = green_ABC_N;
    if (inC->Unlock)
    {
      outC->M_pre_ = SSM_SMI_Pselected__ABC_N;
    }
    else
    {
      outC->M_pre_ = SSM_SMI_Locked__ABC_N;
    }
    break;
  case SSM_SMI_WaitUnlock__ABC_N:
    outC->foreground = black_ABC_N;
    outC->background = grey_ABC_N;
    if (inC->Unlock)
    {
      outC->M_pre_ = SSM_SMI_Unselected__ABC_N;
    }
    else
    {
      outC->M_pre_ = SSM_SMI_WaitUnlock__ABC_N;
    }
    break;
  }
  [...]
}
```

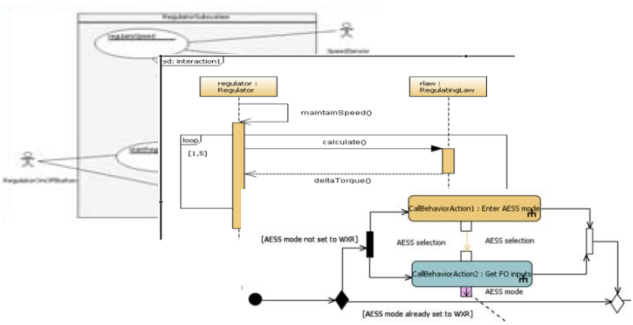
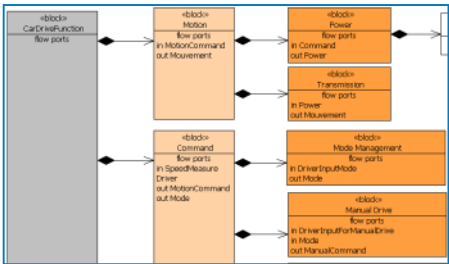
## Certified Code Generation

*Automatic C certified code generator  
(DO-178C, ISO26262, IEC 61508, IEC 61508, EN 50128)  
provides 50% embedded code production and testing cost reduction*

# SCADE System allows for the definition of a proper and safe architecture

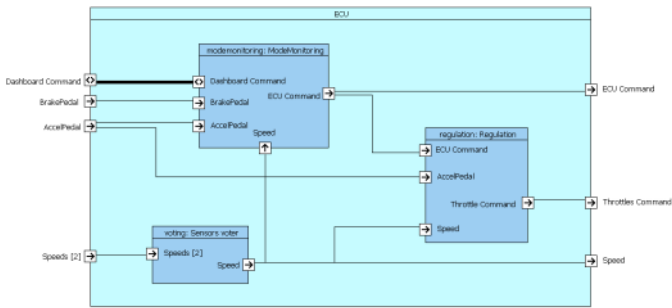
**Model-Based Embedded Systems Engineering**  
*Complete Systems Engineering environment  
for Electronics Systems and Software*

*(SysML-standard based)*



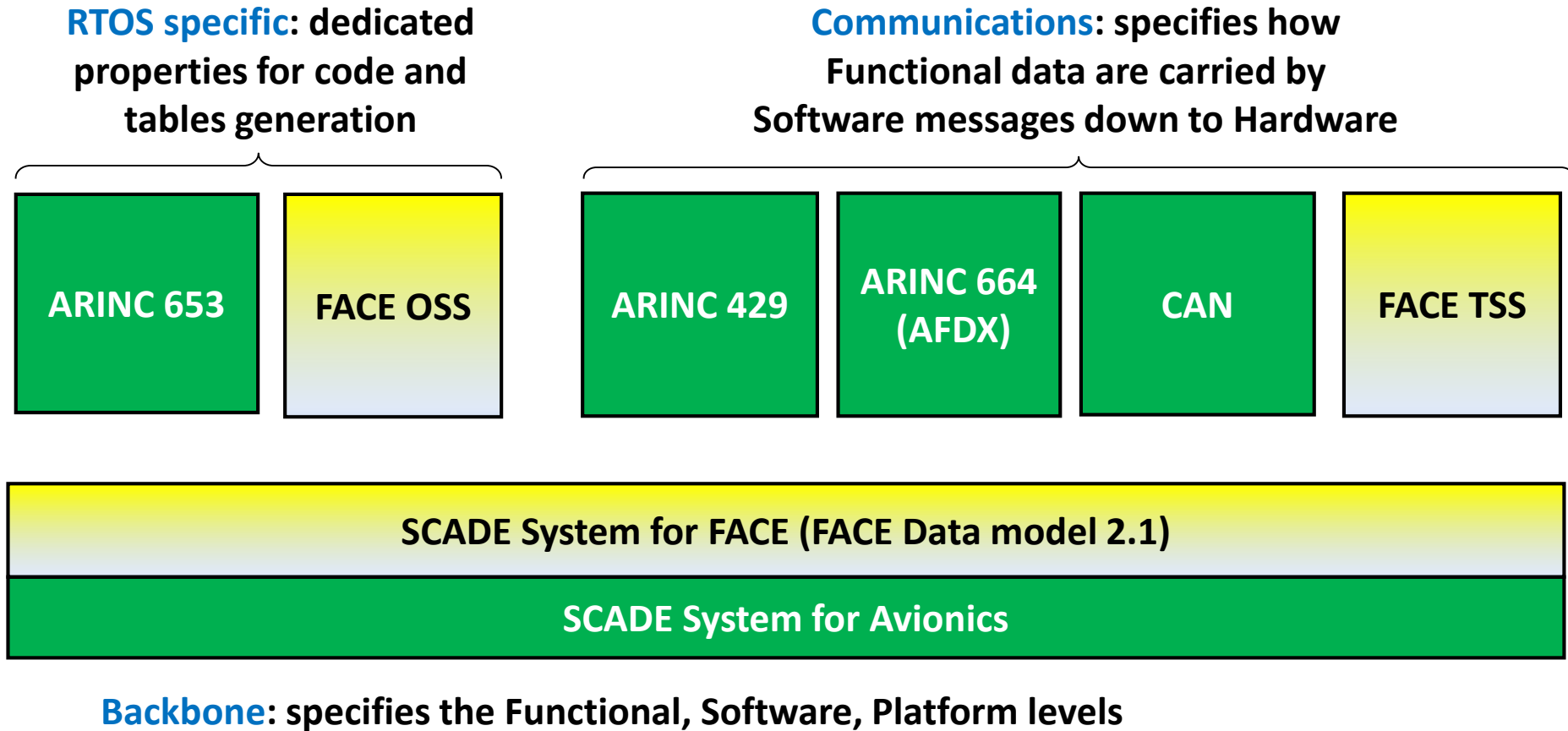
**Requirements Analysis, Functional and Architectural Design**  
*Supports industry and customer specific systems engineering  
configurability (such as FACE and AADL)*

**Embedded System/Software Synchronization**  
*Enables modular System design and verification, Automatic I/O definition  
synchronization, Interface Control Documents production  
and Team Collaboration support*





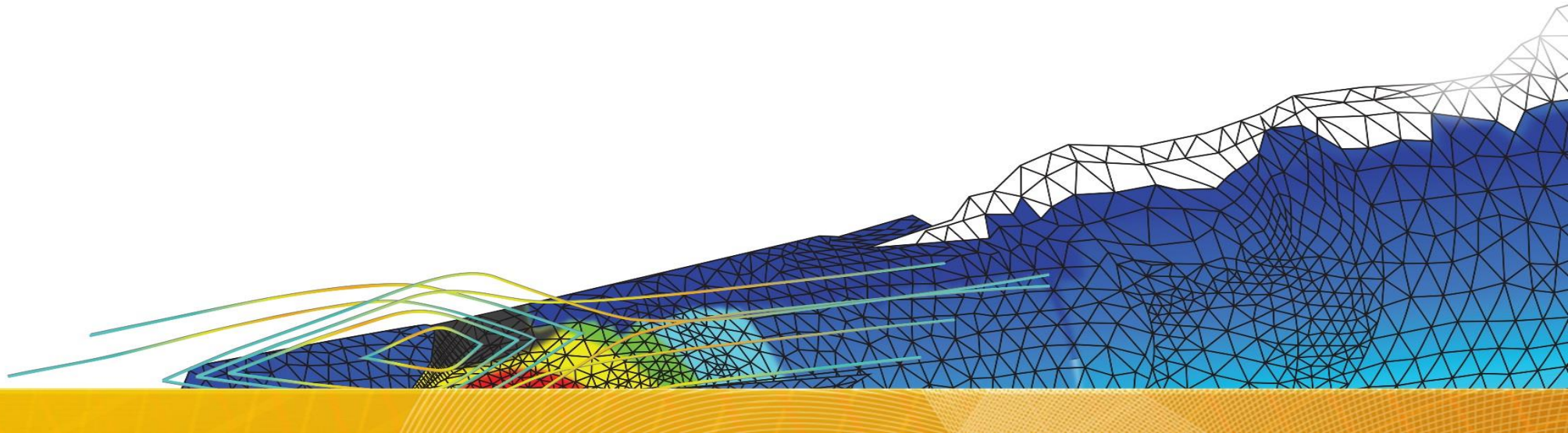
# SCADE System Avionics Packaging with FACE



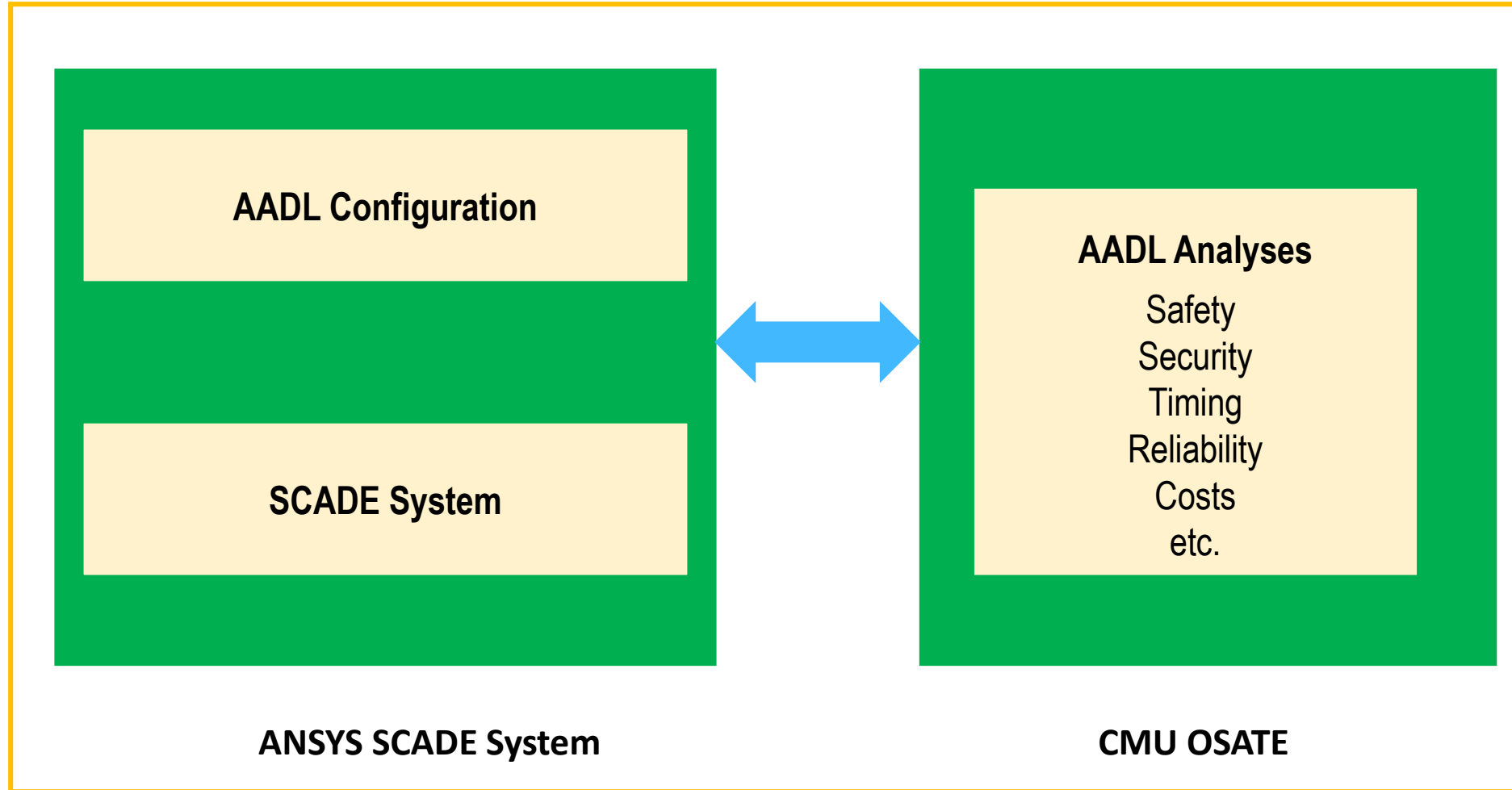




# SCADE AADL Integration and Demo



# SCADE System AADL Configuration

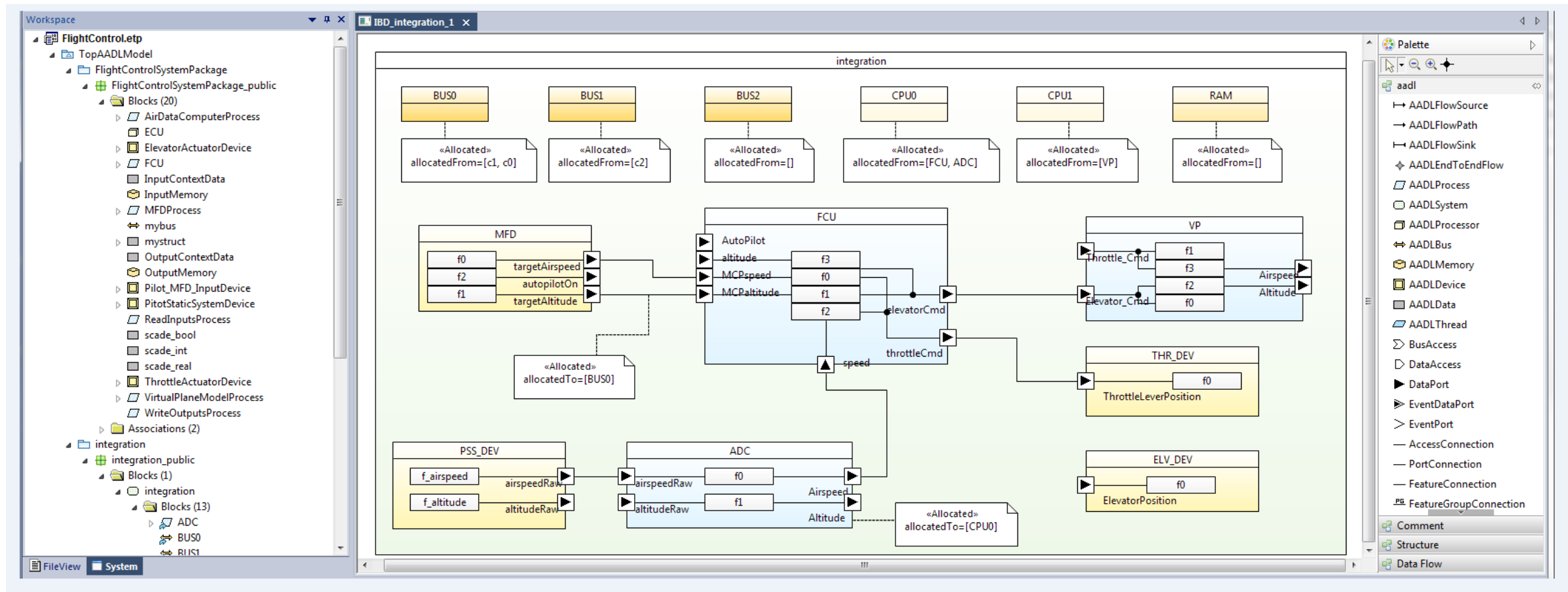


Within the SCADE Studio (Eclipse) platform

# SCADE AADL flow

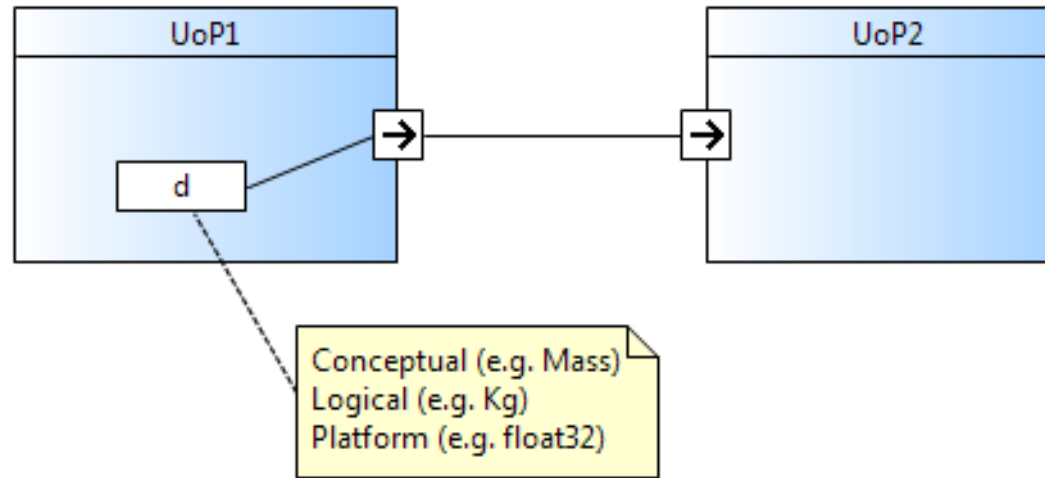
- Using the AADL Configuration, **SCADE System becomes an AADL modeler**
- The SCADE System user can define the architecture of his application in terms of processes, threads, processors, buses, etc.
- The SCADE System user can implement the software in SCADE Suite or Display and calculate metrics such as WCET of functions
- The AADL model, with these pieces of information (the metrics), can be read into OSATE which performs the usual analyses (schedulability, security, safety, etc.)

# SCADE and OSATE AADL Demo



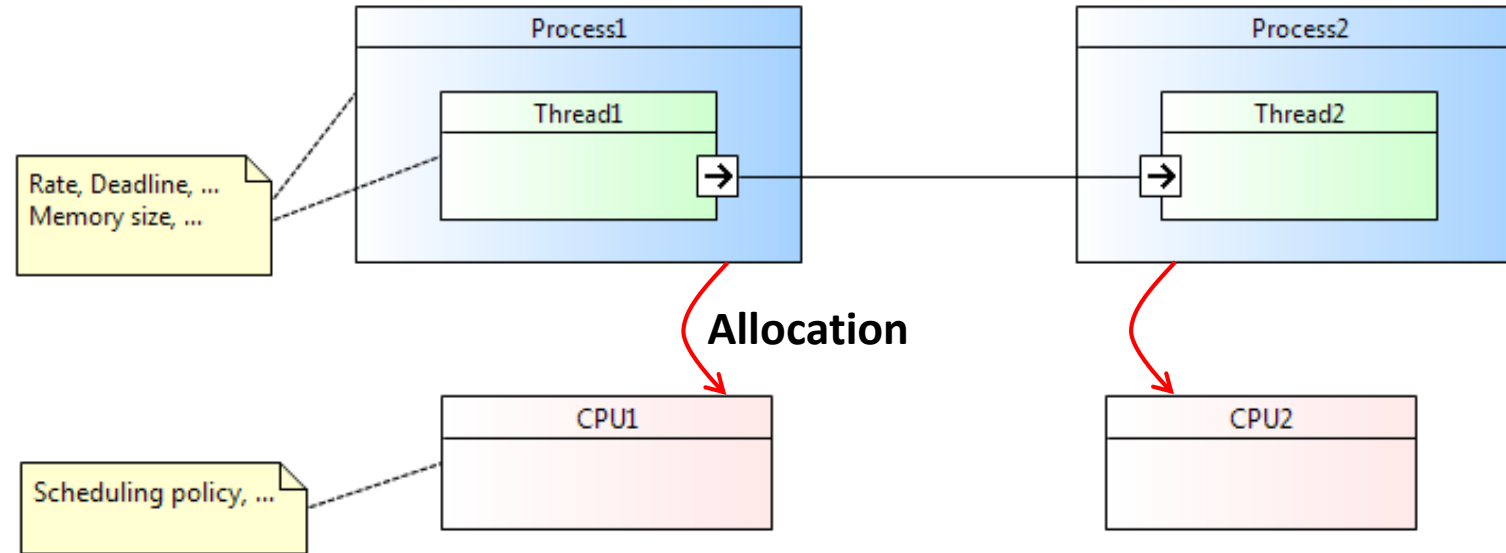


# FACE



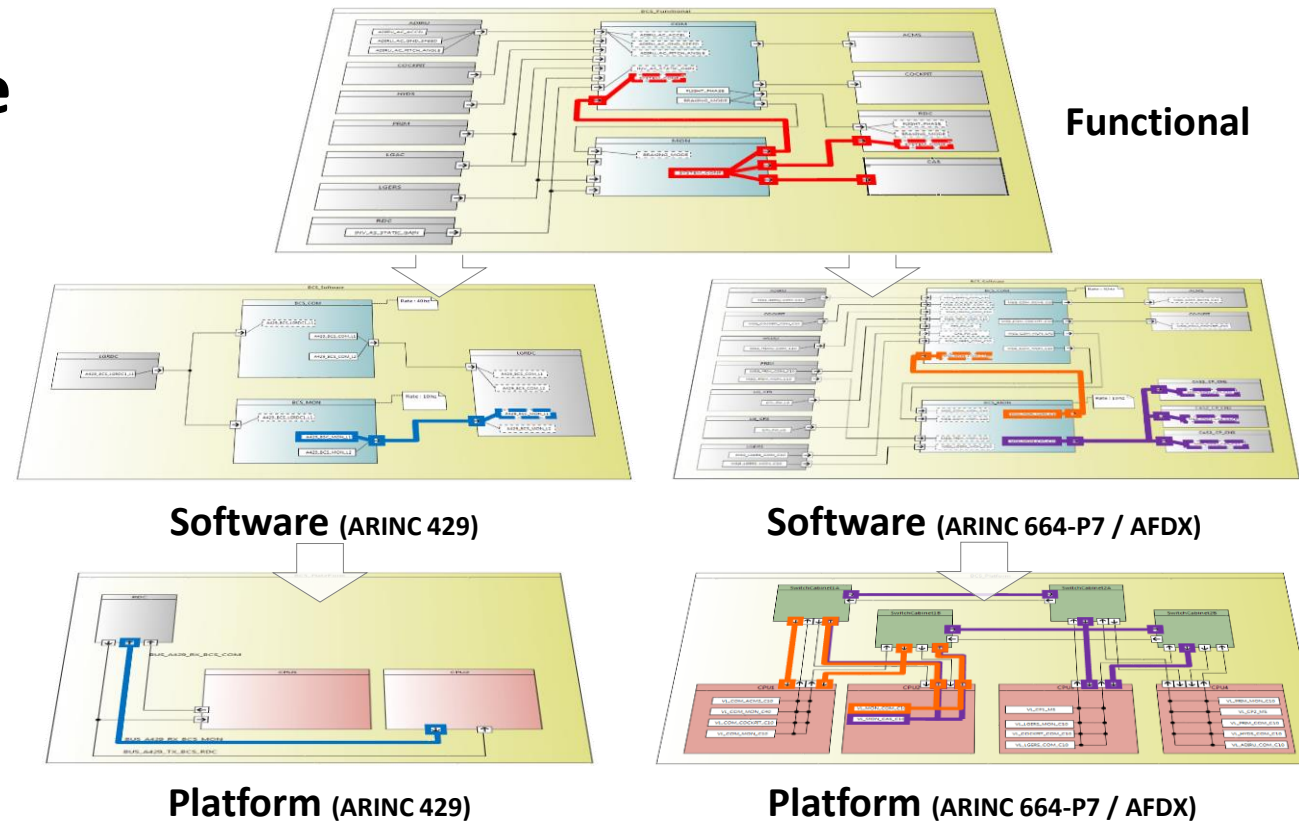
- Platform Independent
  - Best detailed description of exchanged data
  - ➔ Clean “Unit Of Portability” definition
- No platform dependent details on UoP (rate, memory size, ...)
  - No platform dependent details on data exchanges means (protocol messages, ...)

# AADL



- **Clean definition of SW/HW with allocations**
- **Details on timing and process scheduling**
- ➔ **Allows for static analysis e.g. processes deadline satisfaction**
- **No detail on the data exchanged by the threads beside implementation type**
- **No platform dependent details on data exchanges means (protocol messages, ...)**

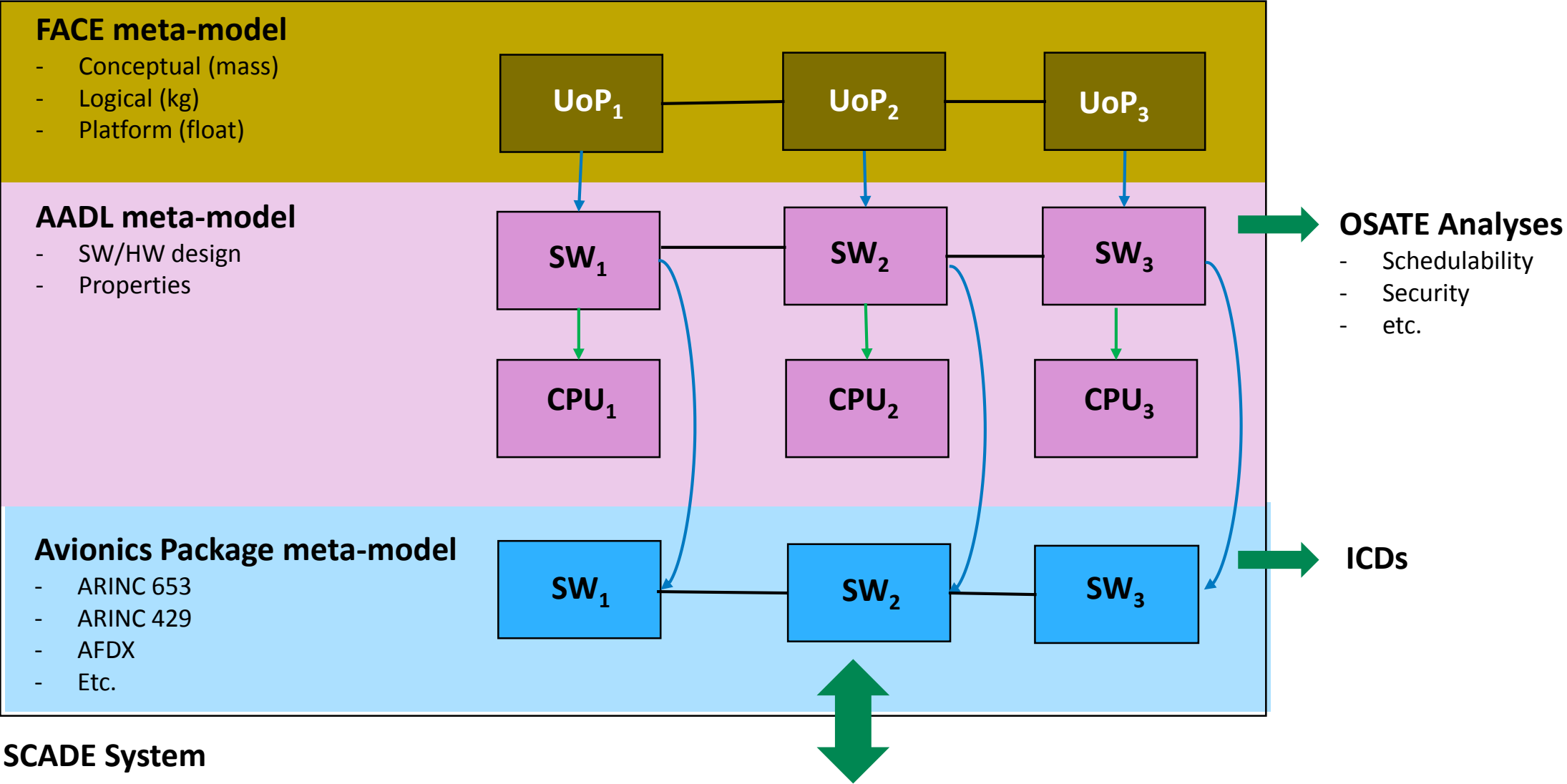
# SCADE Avionics Package



- Detailed description on the data exchanges means  
➔ Allows for ICD generation
- Extensibility through refinement at any level

- No detail on the functional data exchanged beside implementation type
- No details on the SW process control (events, ...) beside ARINC 653 rates

# Reconciliation: FACE, AADL and the Avionics Package 1/2





# Benefits of FACE, AADL and Avionics Package Integration

- **AADL analysis a priori done on an “abstraction” of the real system, focusing on the processes (100’s)**
- **FACE and SCADE Avionics Package deals with the refined system design, focusing on the real ICD (1000’s of messages)**
- **Refinement of the DataMessages structure & information can be done through simple metamodel inheritance**
  - **ARINC 429, AFDX and CAN are provided with model source in the SCADE Avionics Package distribution**
  - **Customizations and introduction of additional protocols (e.g. TTEthernet) only require SCADE System Configurator for their design**



# Q&A

