

Smart Models on Smart Cars

October 20, 2022 | Stuttgart

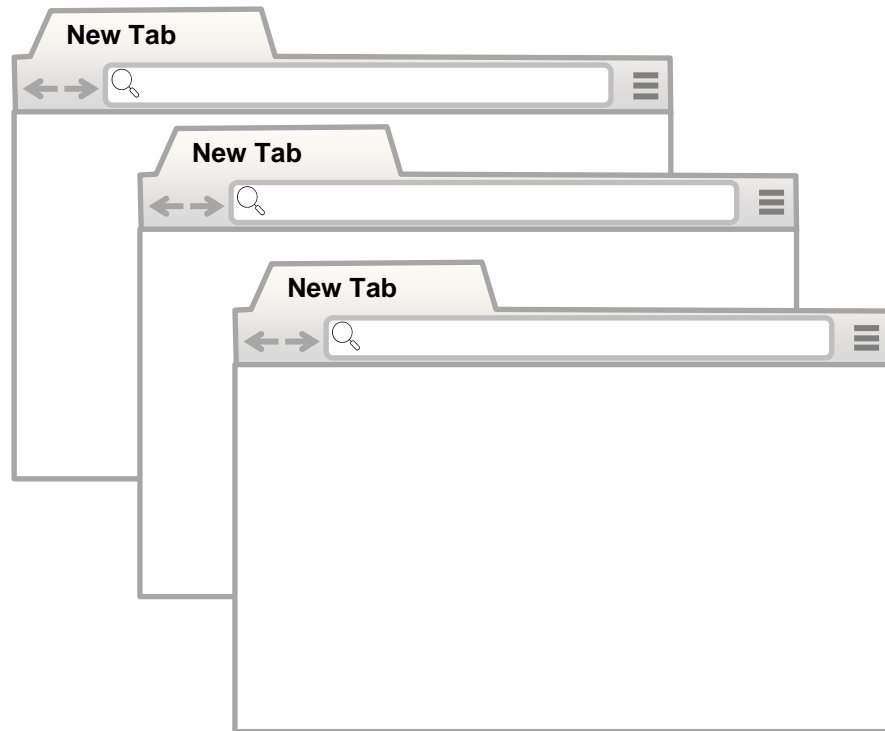
Tunc Simsek



Software is changing how we build cars..

Cars are changing how we write software!

The transition to multi-process architecture changed how people write code for the web



- Enabled web pages to be self-contained applications
- Gave birth to multiple frameworks such as JavaScript, React, ..
- Expect similarly great impact as automotive architectures evolve

Each tab is a different process.

Previously was a monolithic architecture.

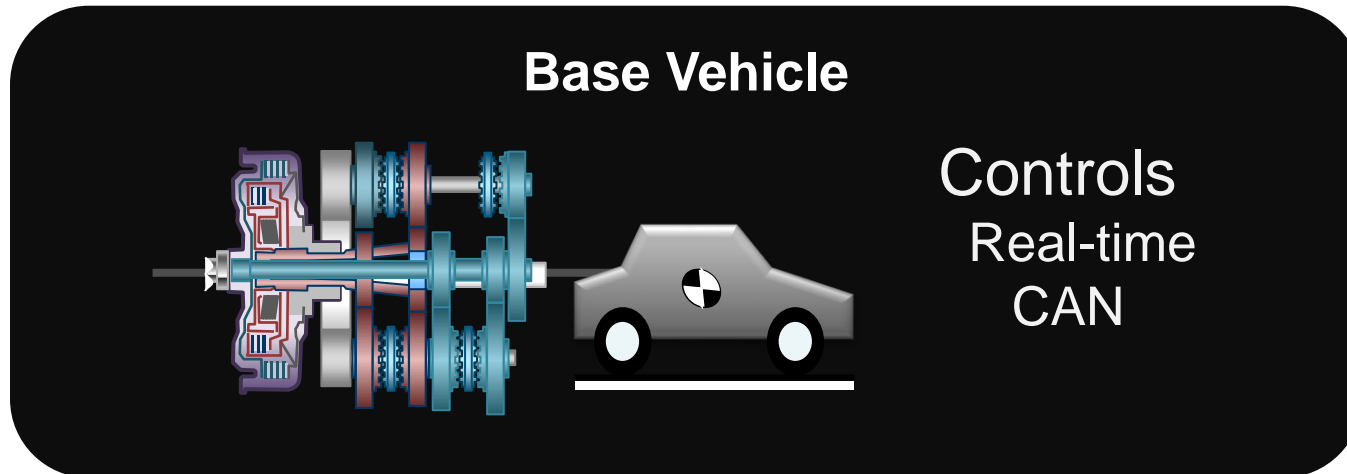
Automotive architectures shape how we create software



Steering,
Braking



Speed,
Velocity

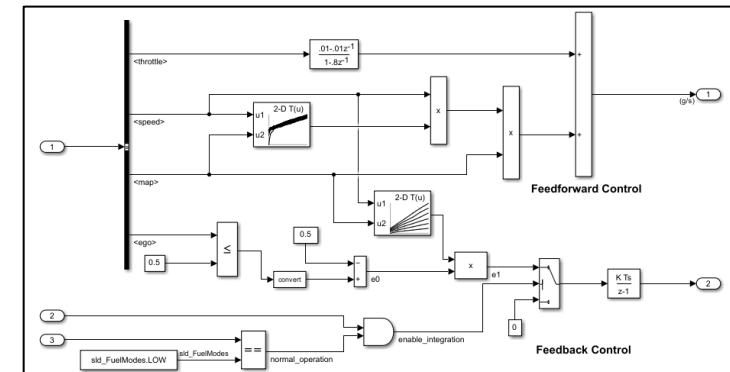


Code-centric Design

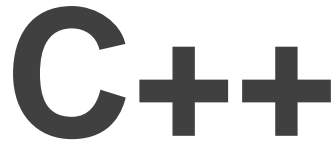
C++



Model-Based Design



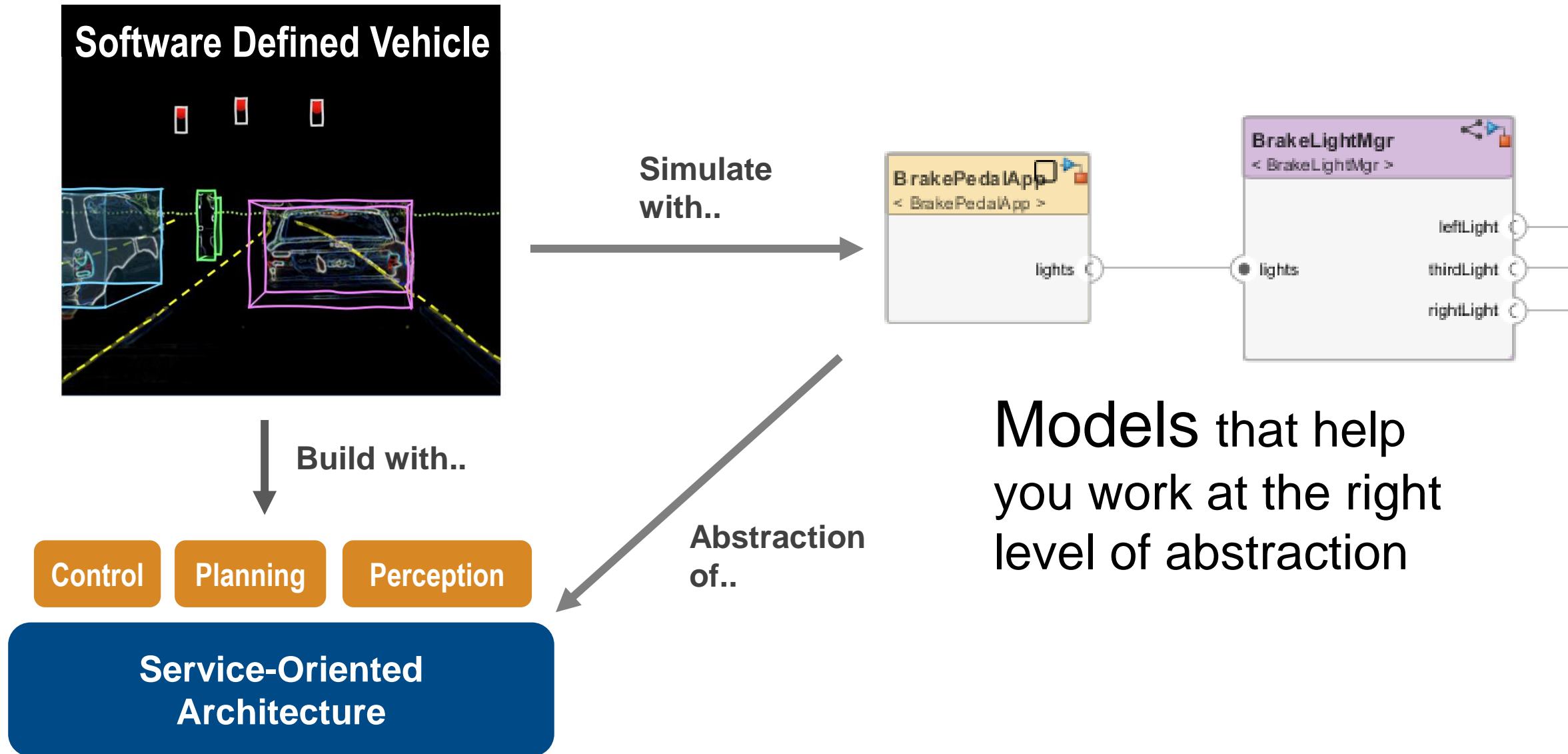
Emerging frameworks are trending to service-oriented architectures

The logo for C++ programming language, consisting of a large 'C' followed by two '+' signs.The logo for ROS (Robot Operating System), featuring three vertical bars of varying heights to the left of the text 'ROS'.The logo for AUTOSAR Adaptive Platform, with 'AUTOSAR' in a bold, black font and 'Adaptive Platform' in a smaller font below it.

Trend towards service-oriented architectures (SOA)

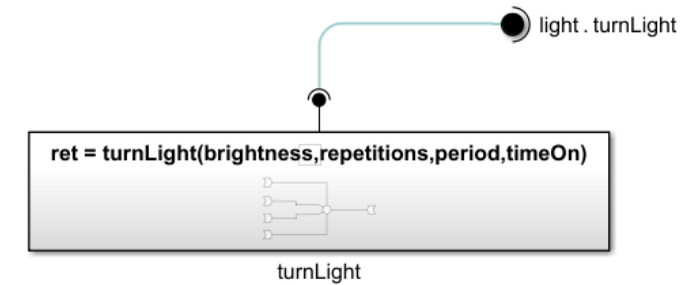
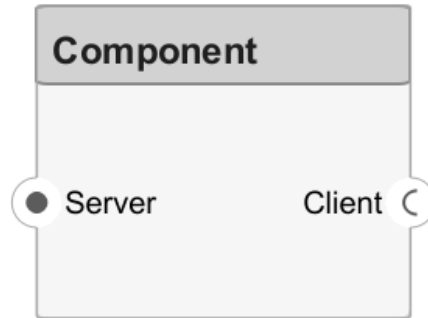
- SOA helps to implement value add capabilities of Software Defined Vehicles
- Access to vehicle data such as camera and sensors to strengthen AI models
- Connectivity & Cloud

Smart CARS call for smart ways to write code





Pillars of SOA with Simulink & System Composer



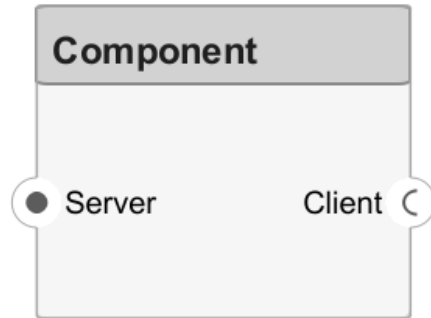
Intuitive
 abstractions
 consistent with
 industry practices

Extensible to
 represent details of
 your architecture

Precise
 semantics that
 allow simulation
 and code
 generation



Pillars of SOA with Simulink & System Composer



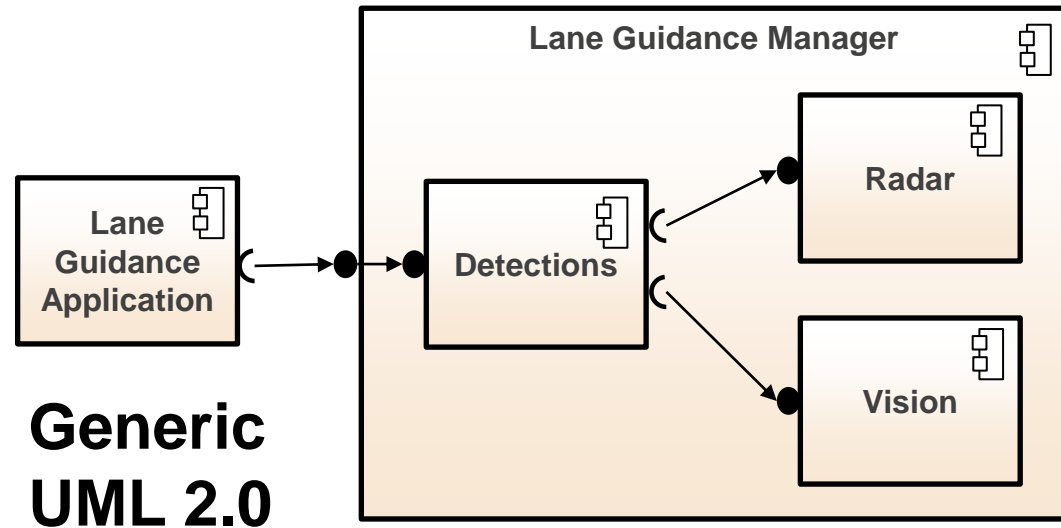
Intuitive

abstractions

consistent with

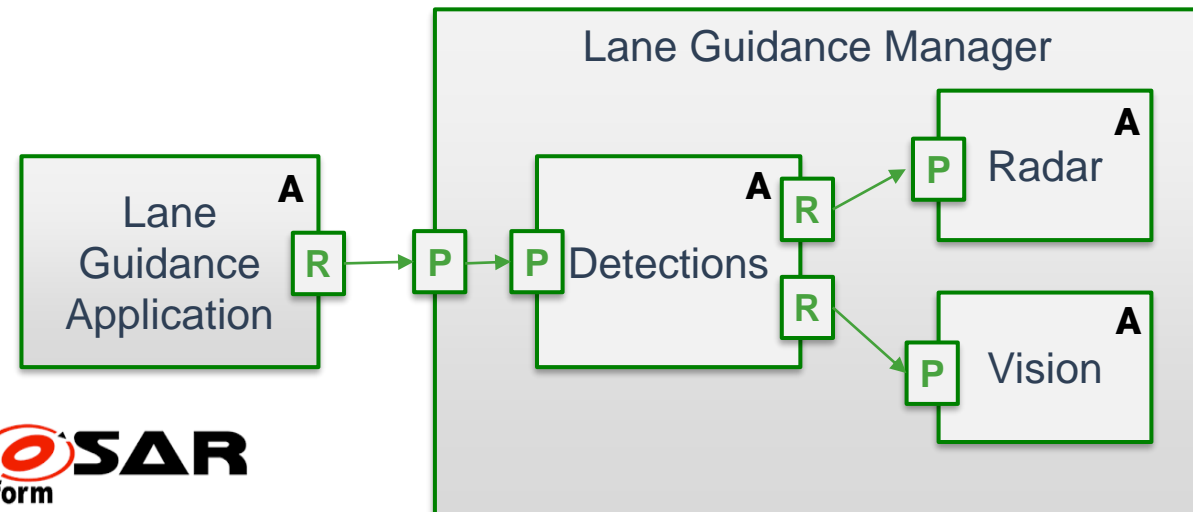
industry practices

Component diagrams are good at modeling service-oriented architectures



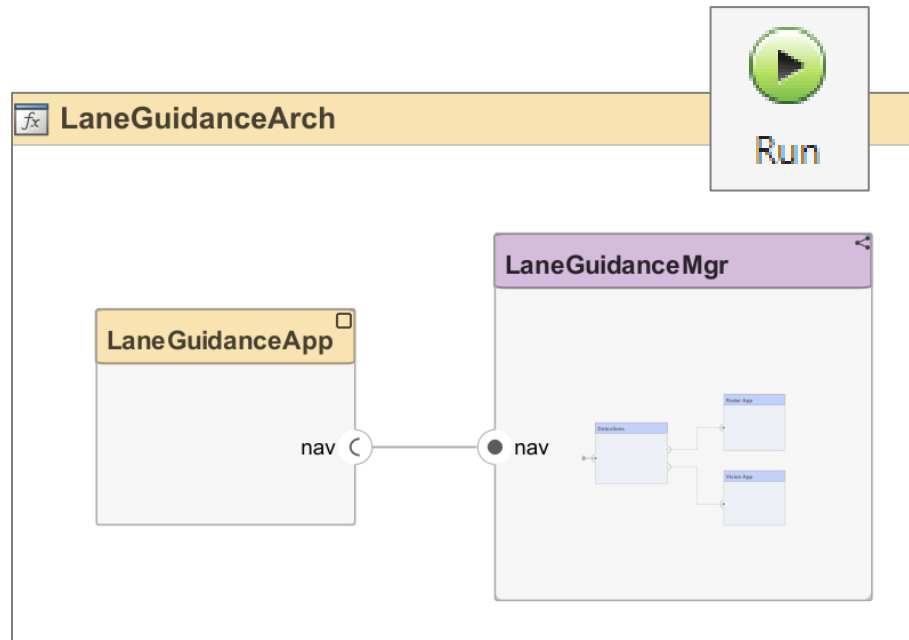
**Generic
UML 2.0**

- Helps to understand system
- Generate middleware code and artifacts (e.g. ARXML, IDL)
- Generic tools don't simulate ☹️

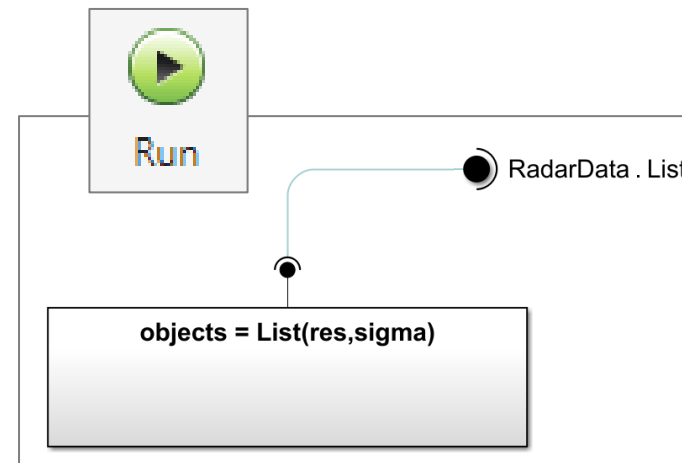


AUTOSAR
Adaptive Platform

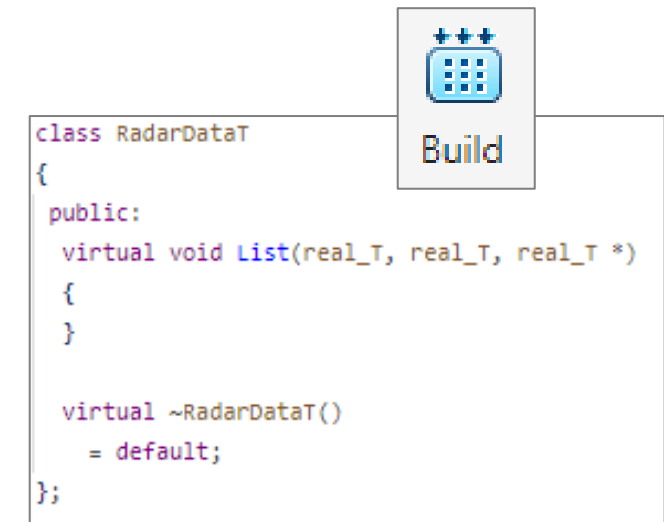
We are bridging the gap between industry standards, simulation and deployment



Describe SOA with
System Composer



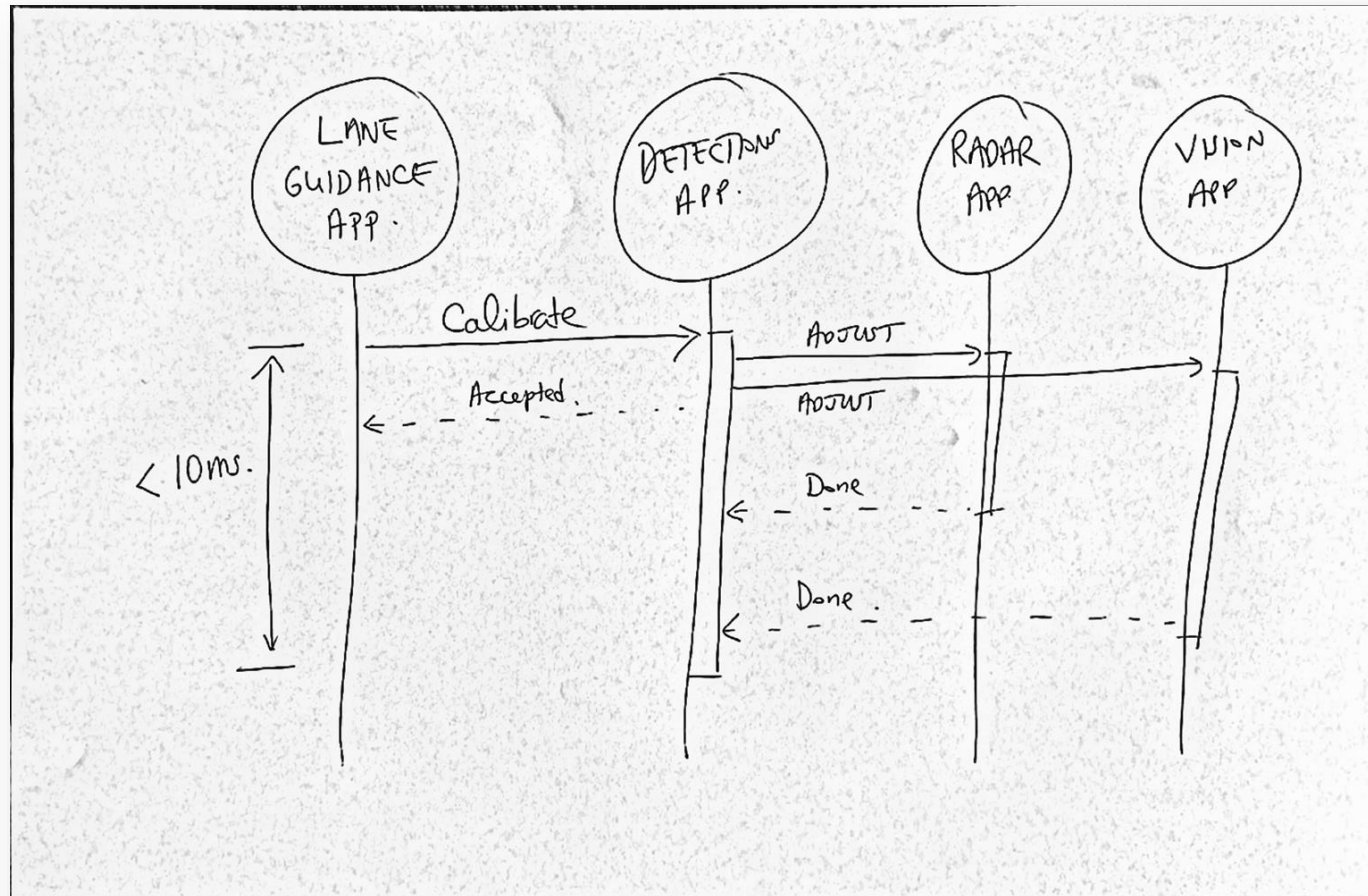
Implement detailed
components with
Simulink



Generate C++ with
Embedded Coder

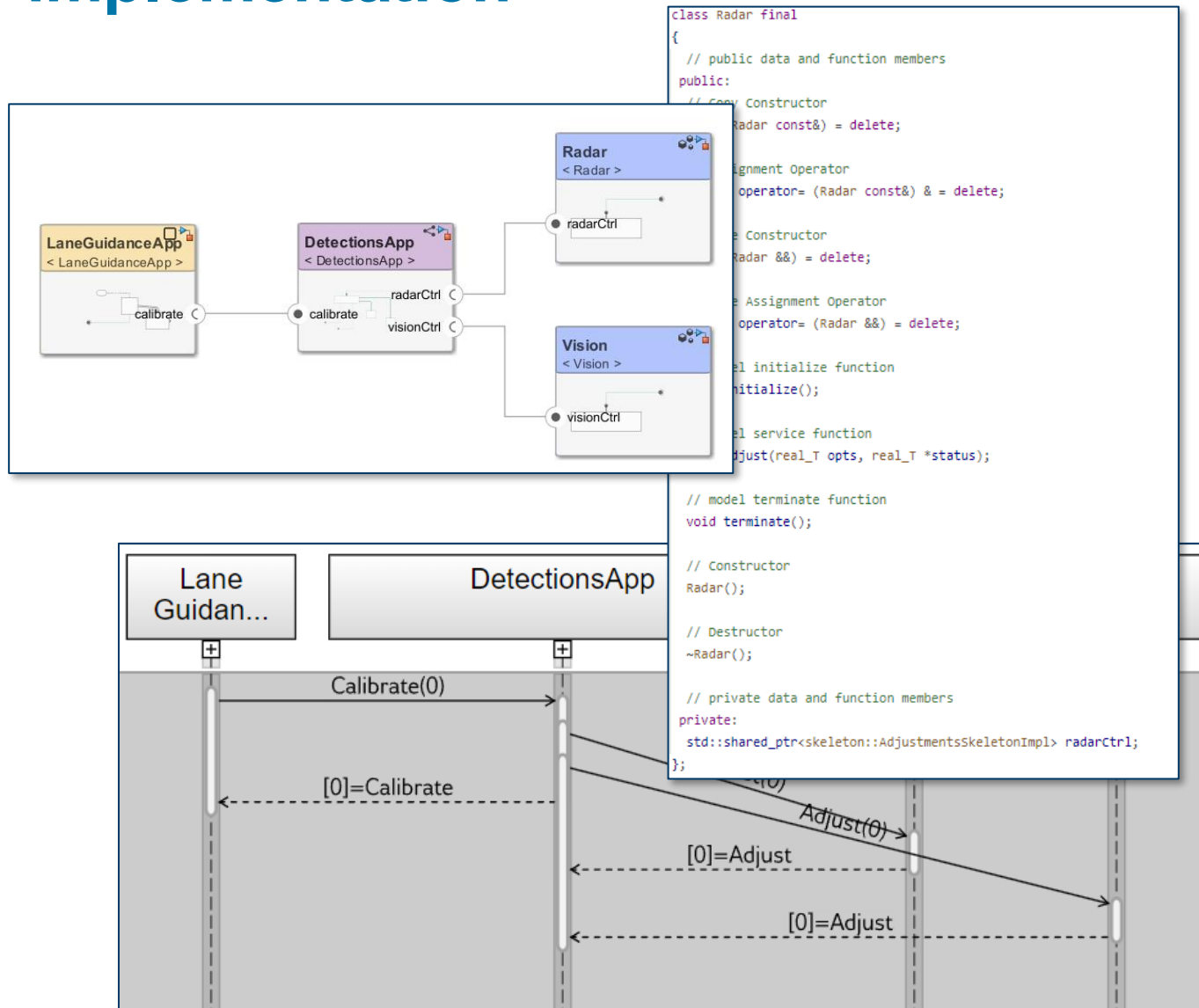
With intuitive abstractions, you can do things quickly

From sketch-to-code in minutes



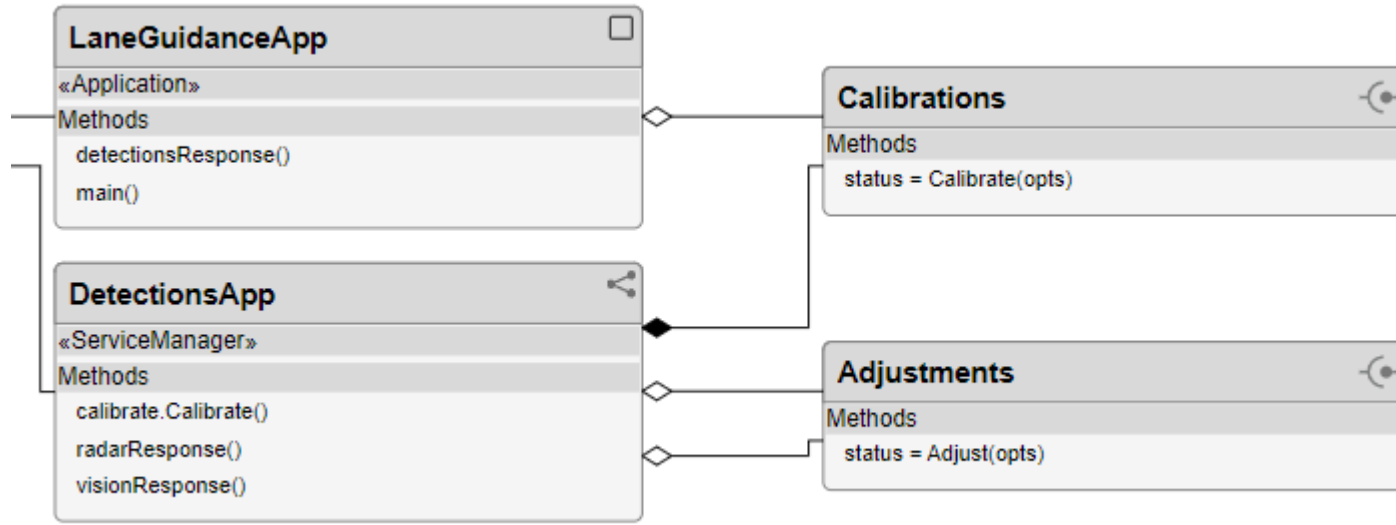
- Start from a sketch of a new service function called Calibrate
- Go from sketch to working code in minutes

Models and code can be handed off to designers for detailed implementation



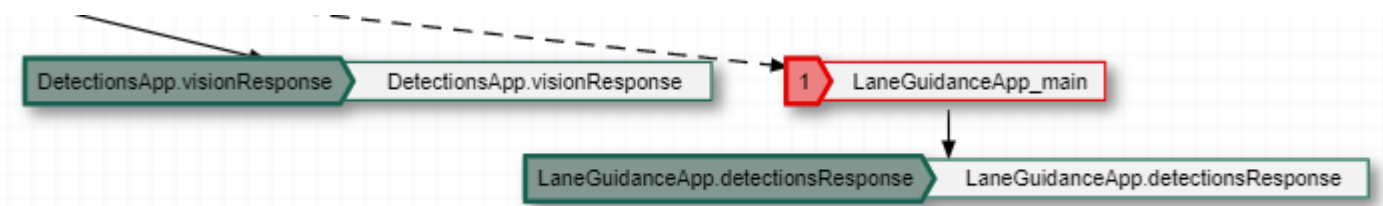
- Component and service descriptions
- Sequence Diagrams
- Skeleton C++ code

Automatically created views can help communicate design intent to Model-based and Code-centric designers



Class Diagrams can communicate the **structure** of your designs

Schedule editor can communicate required, **rates, events** and **dependencies**





Pillars of SOA with Simulink & System Composer

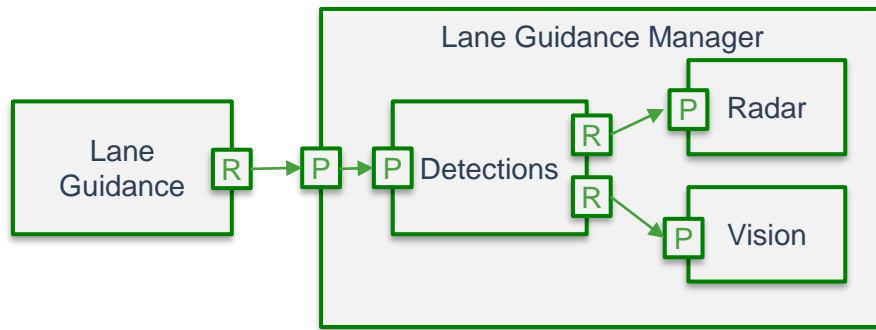
AUTOSAR

DDS **C++**

Extensible to
represent details of
your architecture

Truthfully represent capabilities of your service-oriented architecture

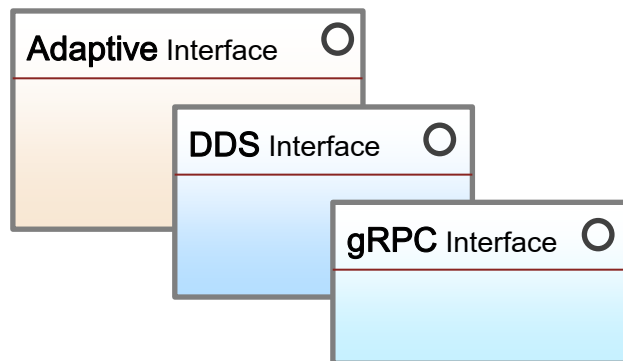
Syntax



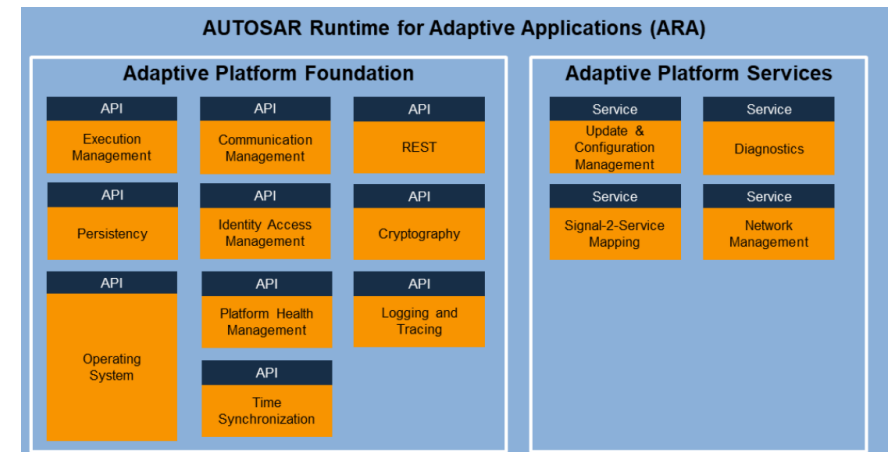
Quality-of-Service



Interfaces



Platform Services



SIMULATION DEBUG MODELING FORMAT APPS

Classic Platform Model Advisor Compare Environment Functions Editor Interface Editor Schedule Editor Profile Editor Apply Stereotypes Classic Component Composition Adapter Sequence Diagram Architecture Views Analysis Model Allocation Editor Model Settings Update Model Stop Time 10.0 Normal Run Stop Export Fast Restart

PLATFORM EVALUATE & MANAGE DESIGN PROFILES COMPONENT DIAGRAMS VIEWS SETUP COMPILE SIMULATE EXPORT

Model Browser
Referenced Files

Composition

tpc_composition > Composition

Composition

Property Inspector

Component

Architecture Info

NAME	VALUE
- Main	
Name	Composition
Kind	Composition
Stereotype	Add..
Parameters	Select

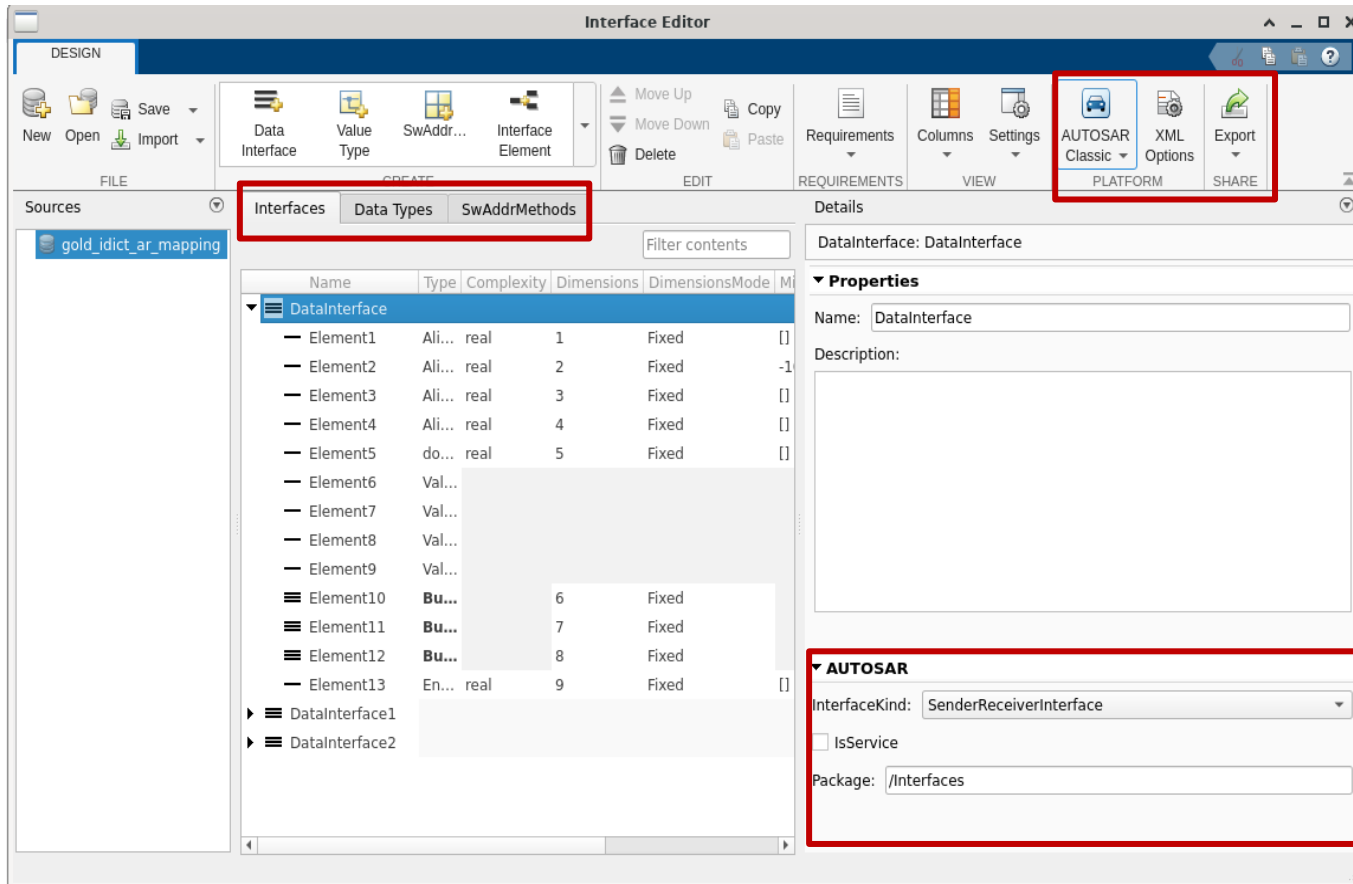
Interfaces

Edit Dictionary Dictionary View

	Type	Dimensions	Units	Complexity	Minimum	Maximum
interfaces.sidd						
APP_HwIO						
Value (hwio_t)	hwio_t	1		real	0	
APP_Percen						
TPS_HwIO						

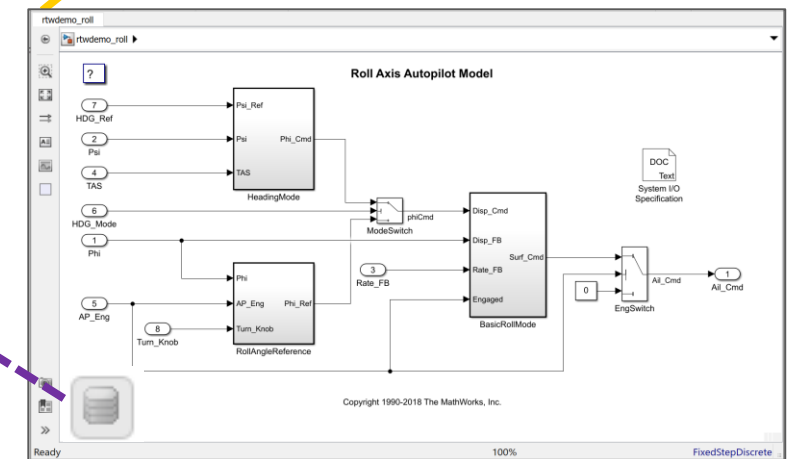
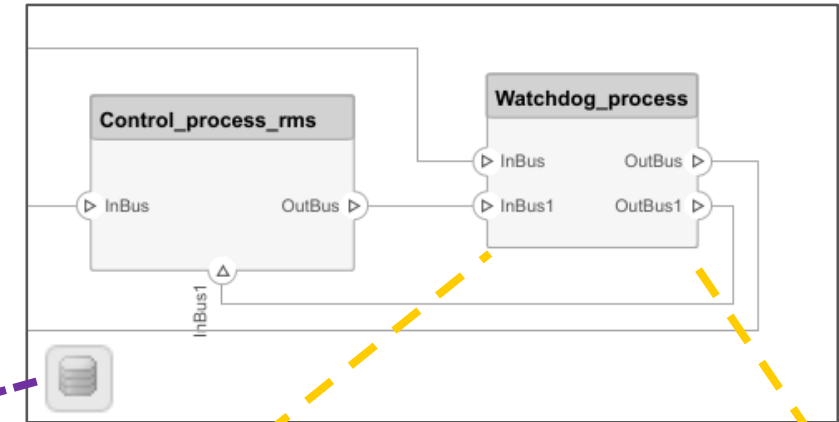
Create an AUTOSAR Composition with AUTOSAR Blockset

Dictionary allows you to manage AUTOSAR interfaces across your architecture and algorithm models



Interface Dictionary Editor

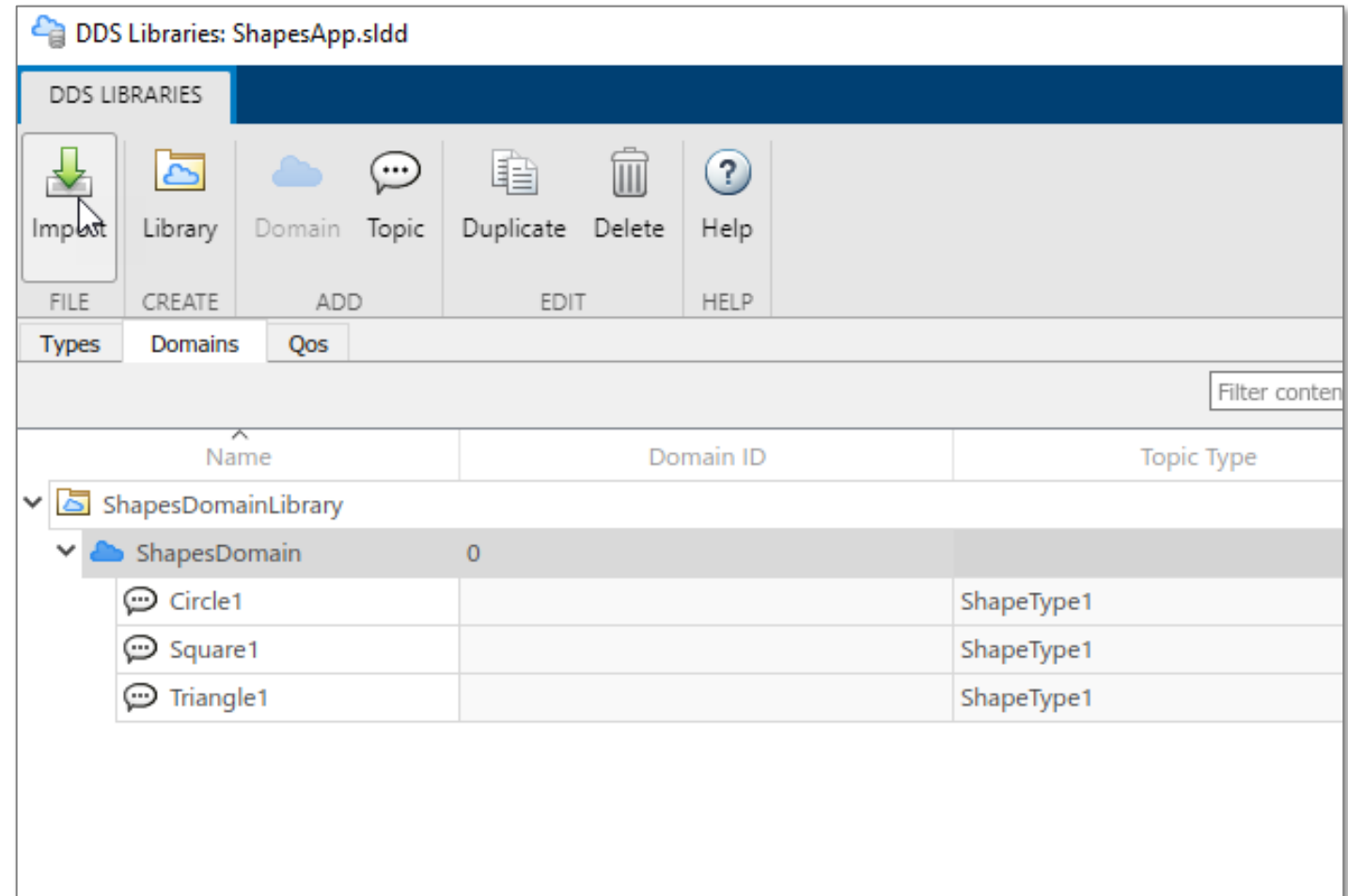
Architecture Model



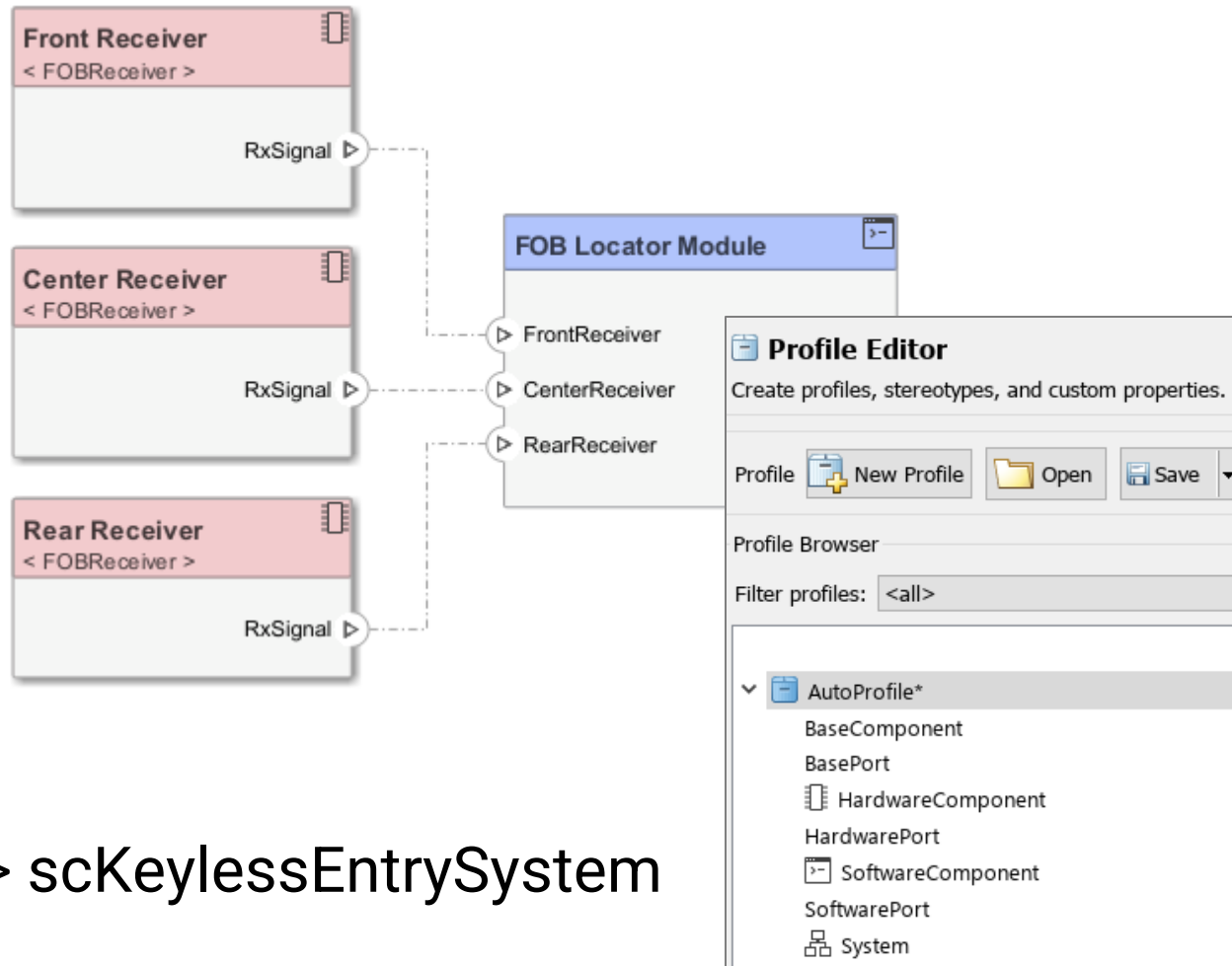
Algorithm Model

DDS Dictionary extends the central interface management paradigm for publish/subscribe architectures

- Interfaces expressed with DDS concepts:
 - Topic Types
 - Domains
 - QoS
- Shared between architecture and design models



You can capture additional details of your architecture with Profiles & Stereotypes

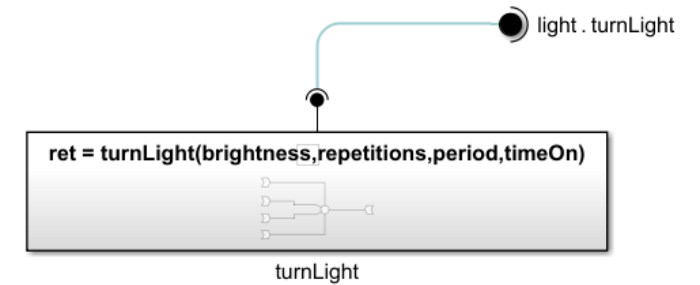


>> scKeylessEntrySystem

- Consistent with industry practices
- Classify elements to capture their meaning
- Graphically expressive with color themes, icons and line styles

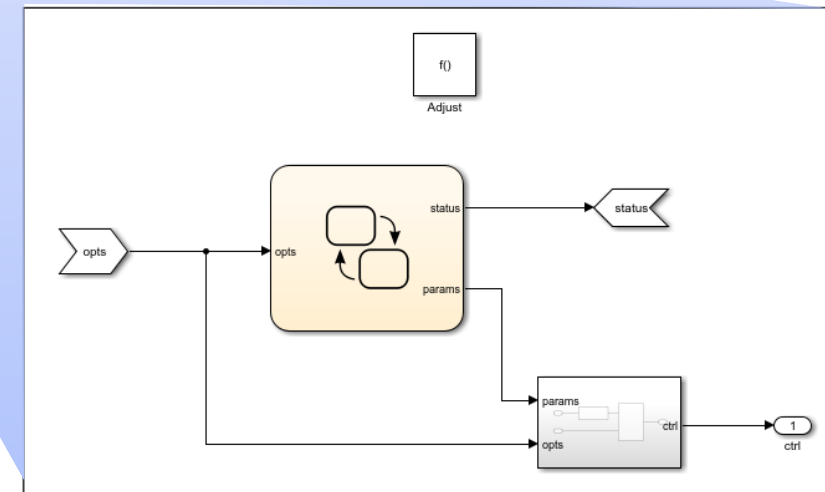
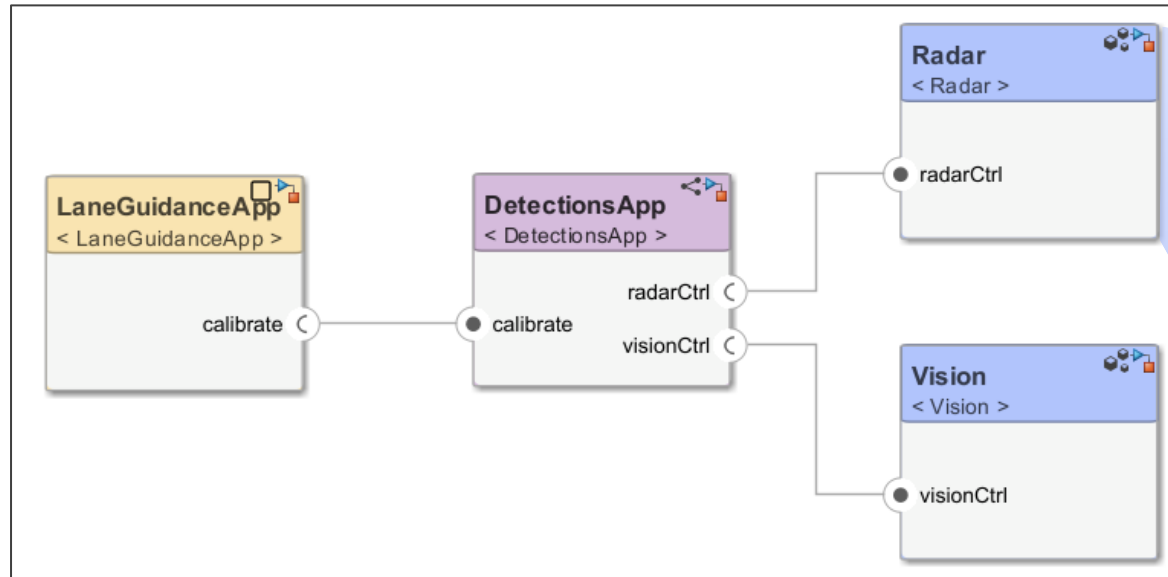


Pillars of SOA with Simulink & System Composer



Precise semantics
that allow simulation
and code generation

What we simulate and generate code



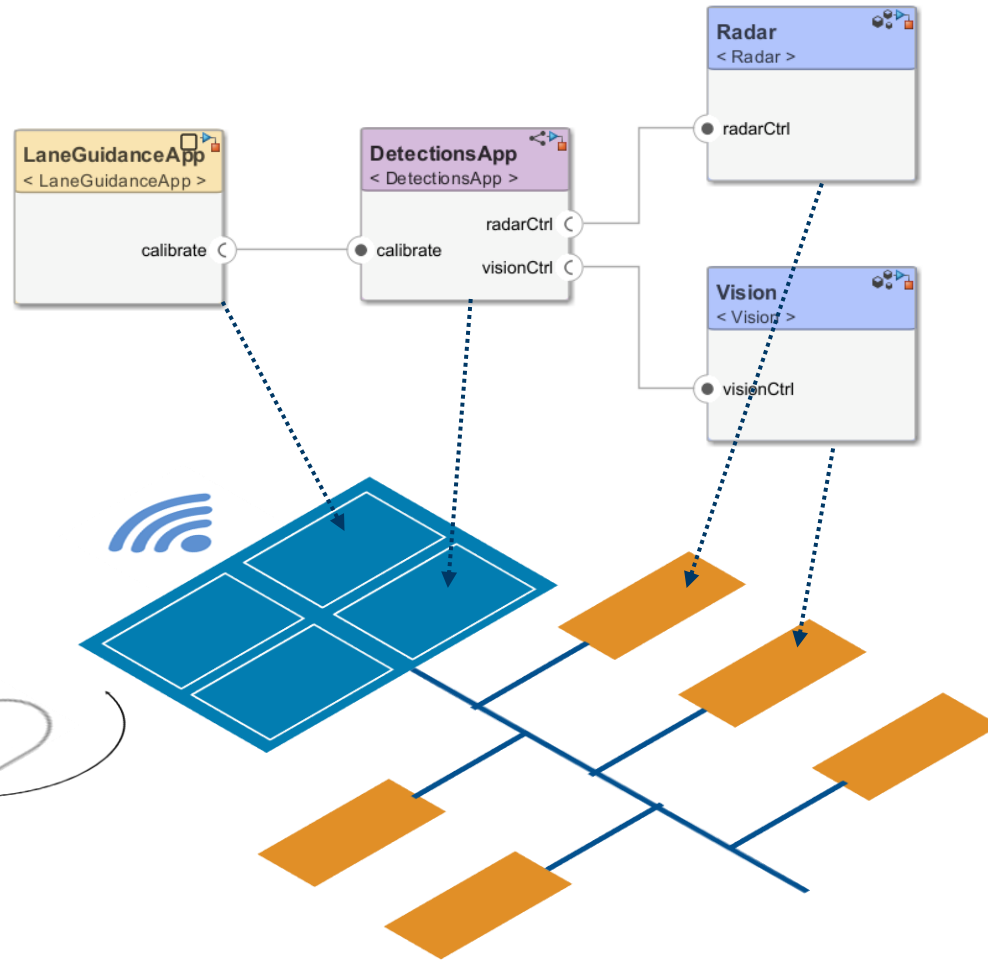
Architectures

- Service interfaces
- Scheduling
- Event sequences

Components

- Signal interfaces
- Algorithms

Capture execution requirements in a distributed service architecture

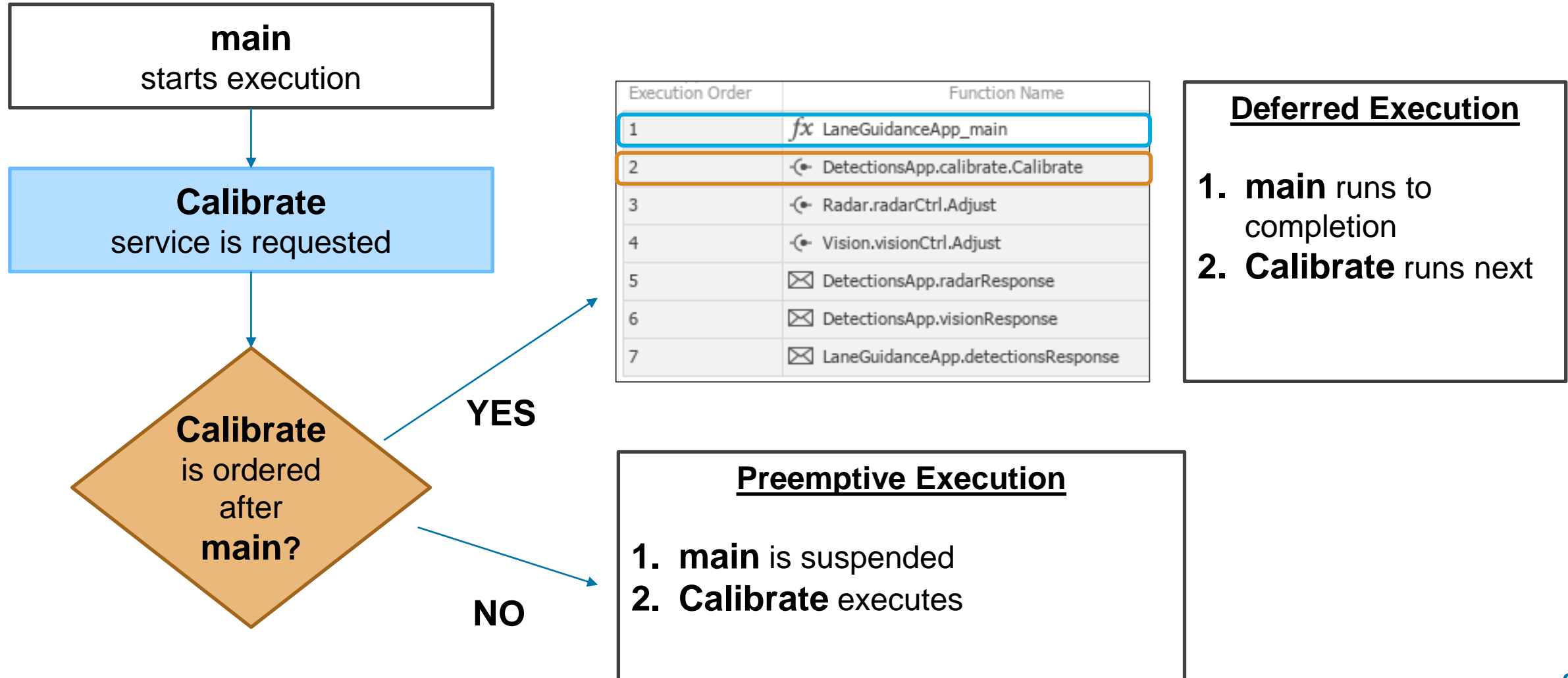


What can I model & simulate in a distributed architecture?

- SOA architectures are event-based
- Describe **periodic, service** and **message** events
- Simulate relative ordering and queueing

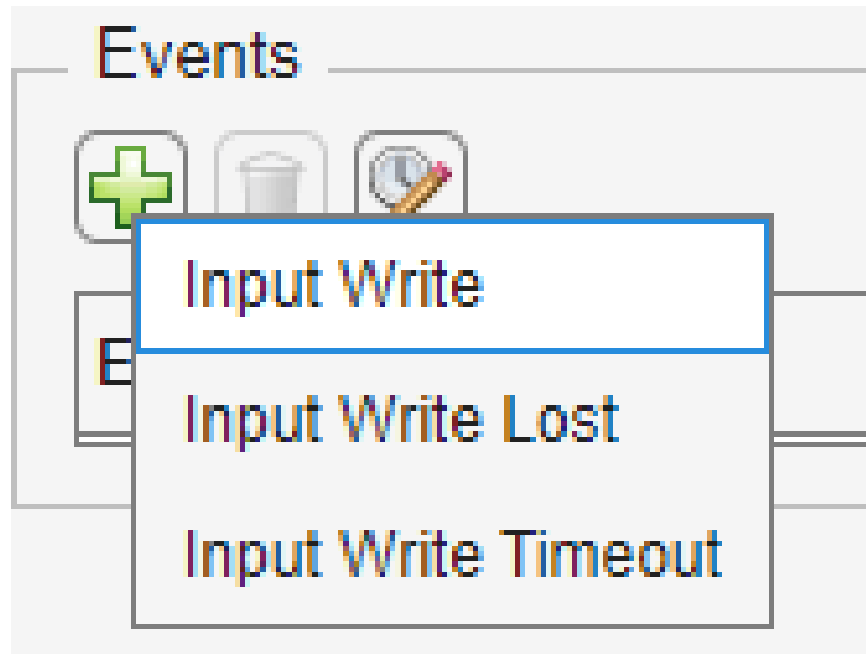
Execution Order	Function Name
1	<i>fx</i> LaneGuidanceApp_main
2	☛ DetectionsApp.calibrate.Calibrate
3	☛ Radar.radarCtrl.Adjust
4	☛ Vision.visionCtrl.Adjust
5	☒ DetectionsApp.radarResponse
6	☒ DetectionsApp.visionResponse
7	☒ LaneGuidanceApp.detectionsResponse

Ordering allows simulation of pre-emptive and deferred execution

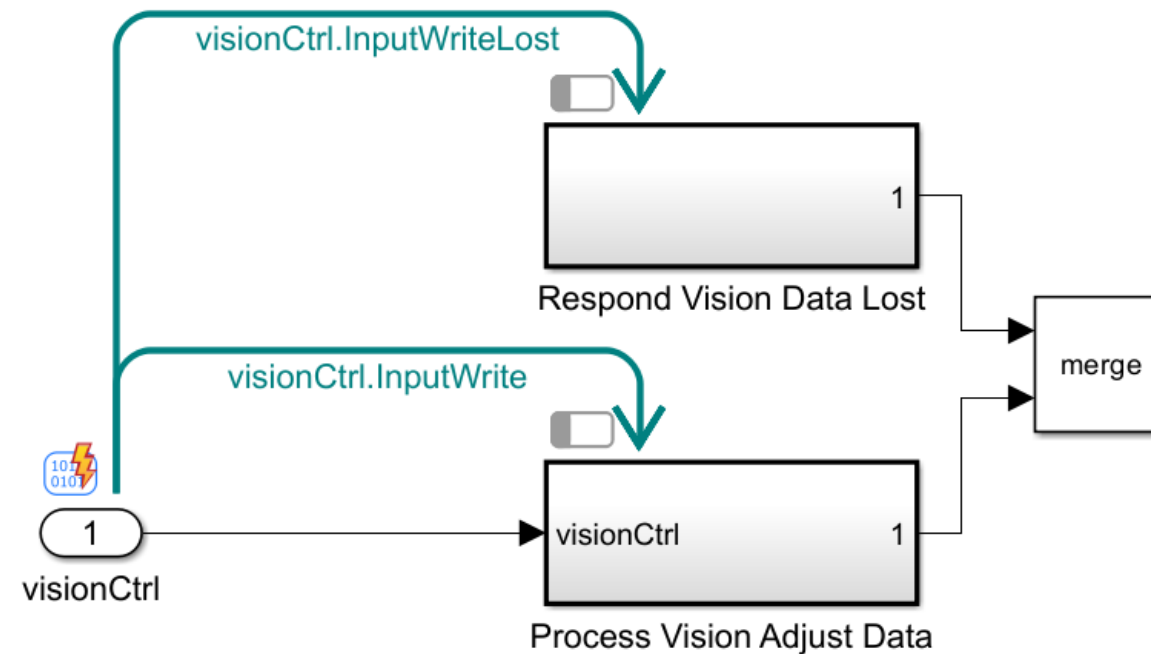


In addition to application events, DDS Blockset allows you to define responses to communication events

Define events that map to DDS communication events



Simulate response functions triggered by events

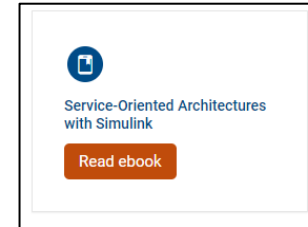


Key takeaways

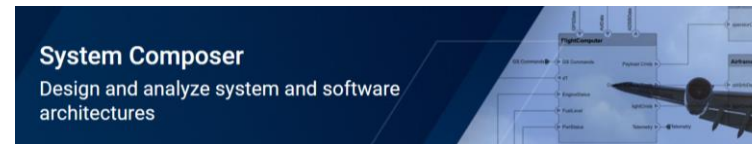
- Automotive software architectures are evolving, pushed by needs of **Software Defined Vehicles**
- New, **service-oriented architectures** are required to master complexity and enable frequent updates
- You can design, simulate and generate code to deploy service-oriented applications in **Simulink and System Composer**, reusing your existing expertise and models

Call to action

- [SOA Webpage](#)



- [System Composer](#)



- [AUTOSAR Blockset](#)



- [DDS Blockset](#)



- Technical Paper from Embedded World 2022 - [Develop and Integrate AUTOSAR Classic and Adaptive Applications Based on SOME/IP](#)

- Presentation from AUTOSAR Conference 2022 - [Designing and deploying interoperable AUTOSAR and non-AUTOSAR applications for heterogeneous automated driving platforms](#)