

Model Service-Oriented Architectures (SOA) in Simulink

Luigi Milia – Automotive Industry Manager

Shwetha Bhadravathi Patil – AUTOSAR Product Manager

Agenda

- SOA: a new paradigm for automotive software
- Simulink for Service-Oriented Applications
- Key take-aways

Agenda

- SOA: a new paradigm for automotive software
- Simulink for Service-Oriented Applications
- Key take-aways

Your next car will have a lot more software

- **How to** add more software

- Add more functions to ECUs?
- Add more ECUs to E/E network?



- system integration and testing too complex
- hardware too costly

- Incremental approach **doesn't scale...**



To ensure **safety** in increasing degrees of **autonomy**, software **quality** and **complexity** are a key challenge for the automotive industry, requiring a **rethink** today's vehicle **software and E/E architectures**.*



* SDV Trends, Challenges, and Implications for OEMs - McKinsey presentation at Gasgoo – July 2020

Industry is investing to transform software development

- **Consolidation** of people and electronics
- Development of **new software platforms**

The complexity of modern electronics/electrical (E/E) architectures is making it more challenging to actually assemble components like wiring harnesses into vehicles so suppliers like Bosch and competitors Continental and Aptiv APTV +1.7% are following the path set out by Tesla TSLA +0.2% with more powerful computers that consolidate the capabilities of many smaller discrete electronic control units. This in turn can help lead to simplified wiring.



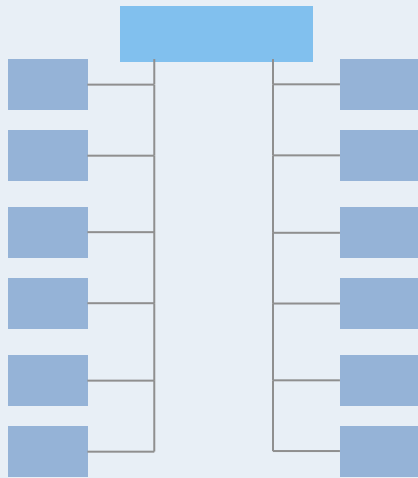
Bosch Consolidates All Automotive Software And Electronics Into New Division



OEMs developing their own **vehicle operating systems**: VW.OS from *Car.Software Org*

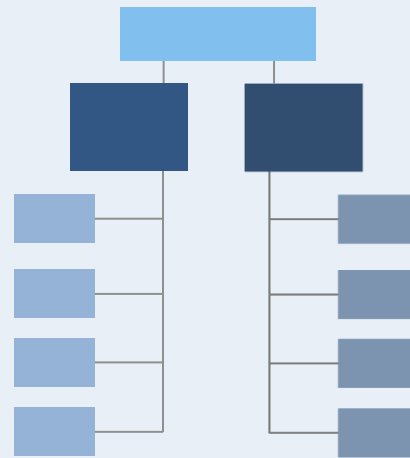
Evolution of E/E architectures

yesterday



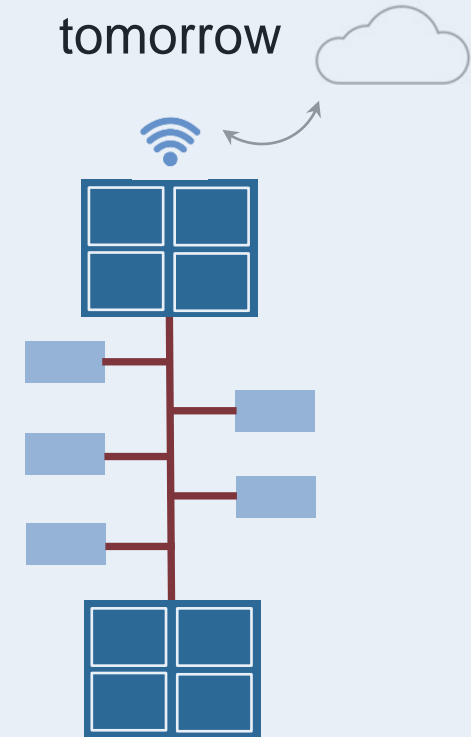
- Centralized gateway
- ~100 ECUs
- One function per ECU
- Heavy wiring

today



- Domain controllers (powertrain, chassis, body, ...)
- Multi-core ECUs
- Consolidation of functions

tomorrow



- Vehicle/zone controllers (front, rear, ...)
- High-performance CPUs (Multi-core, GPU, FPGA)
- High-speed ethernet

Towards Service-Oriented Architectures

yesterday

- High SW-HW coupling
- No standard APIs
- No/minimal SW reuse

today

- Static SW component allocation (design-time)
- Signal-based communication
- HW abstraction
- Monolithic update (full image flashing)

tomorrow

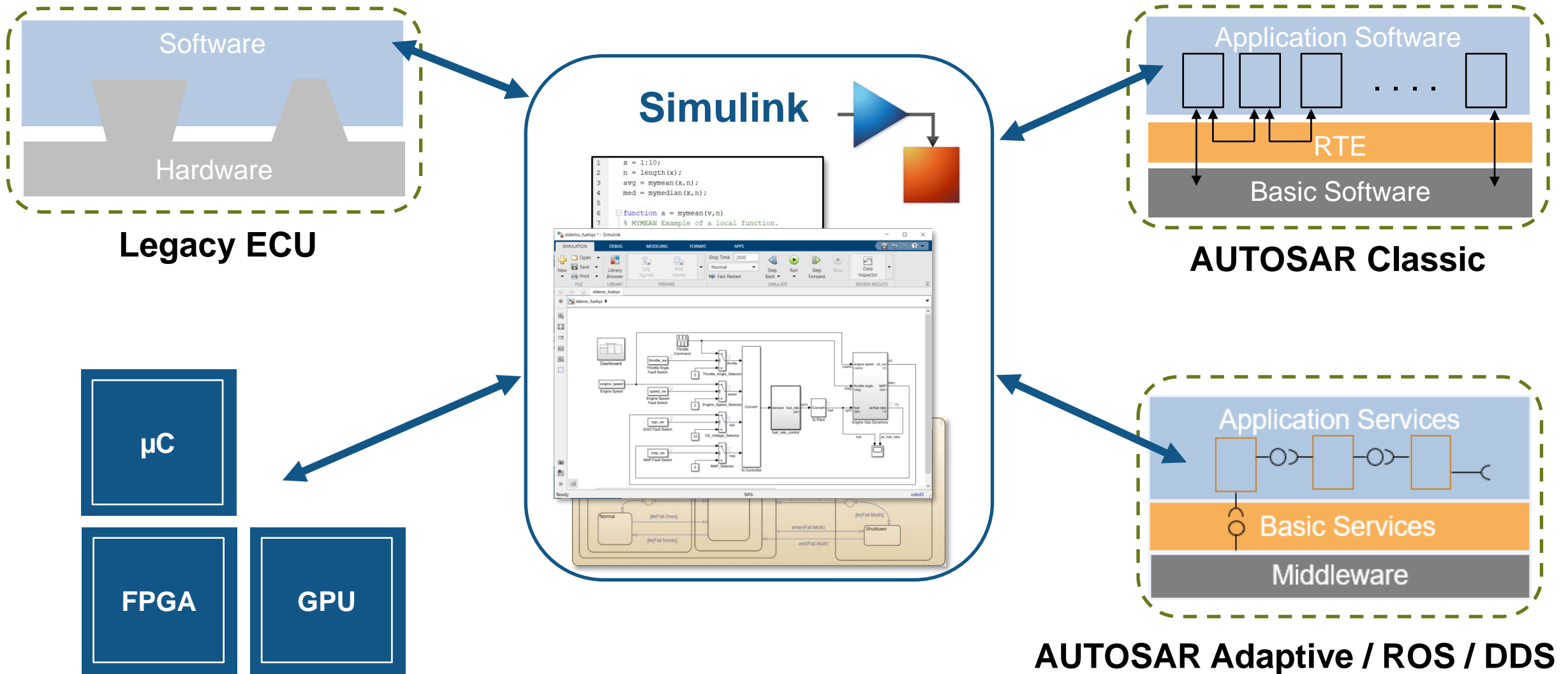
- Dynamic Service discovery (run-time)
- Service-oriented communication
- Higher HW abstraction
- Selective updates (OTA)

Legacy

Components-based

Service-oriented

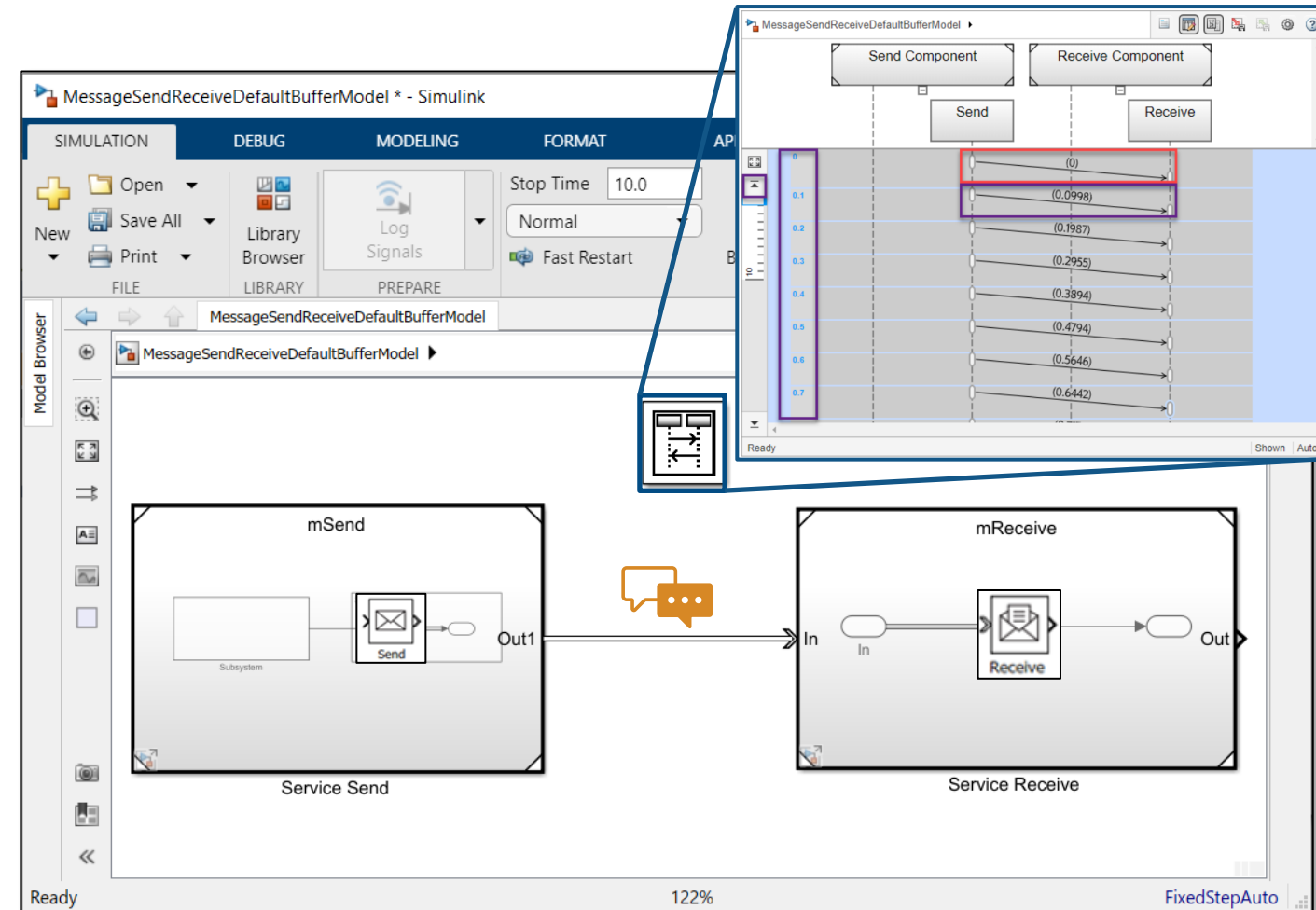
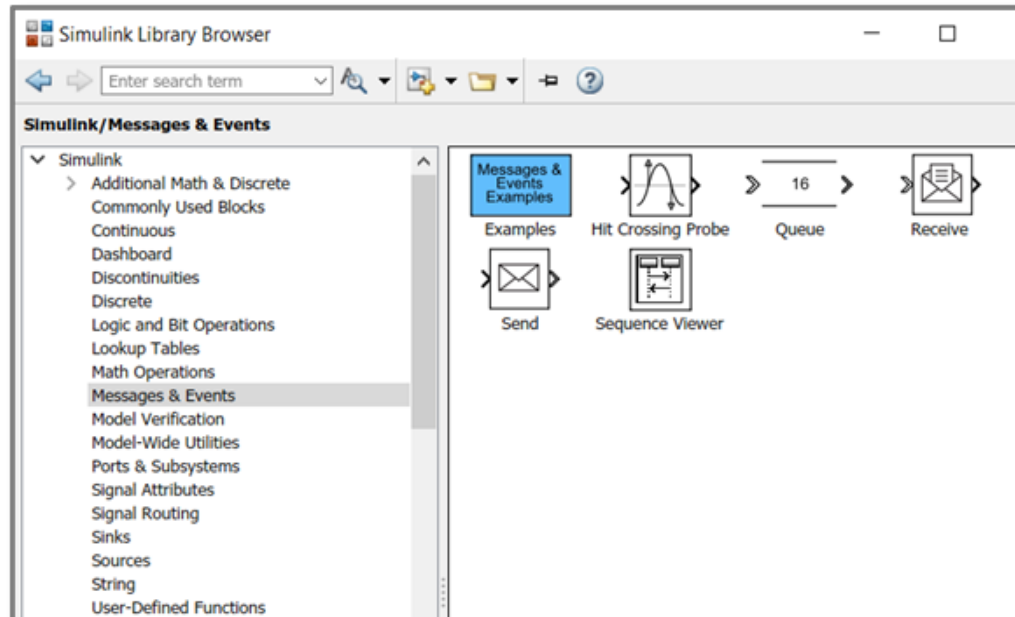
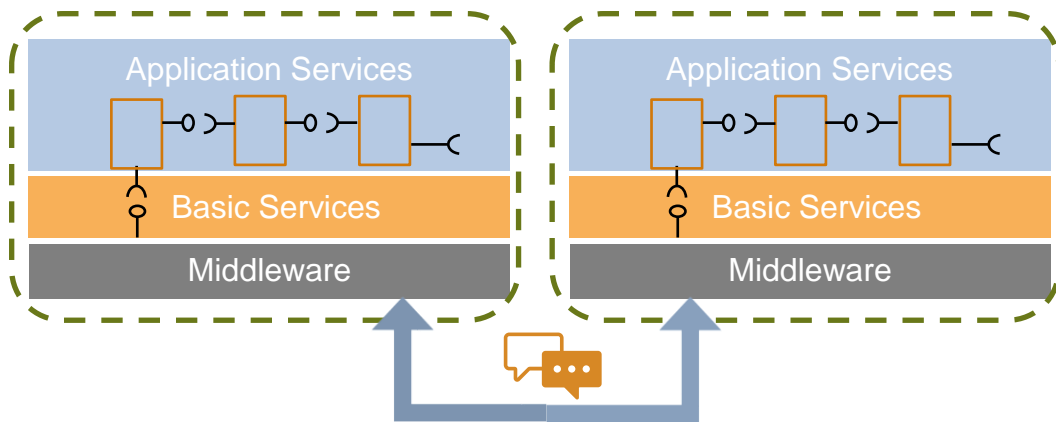
Simulink : design software once, deploy to many targets



Agenda

- SOA: a new paradigm for automotive software
- Simulink for Service-Oriented Applications
- Key take-aways

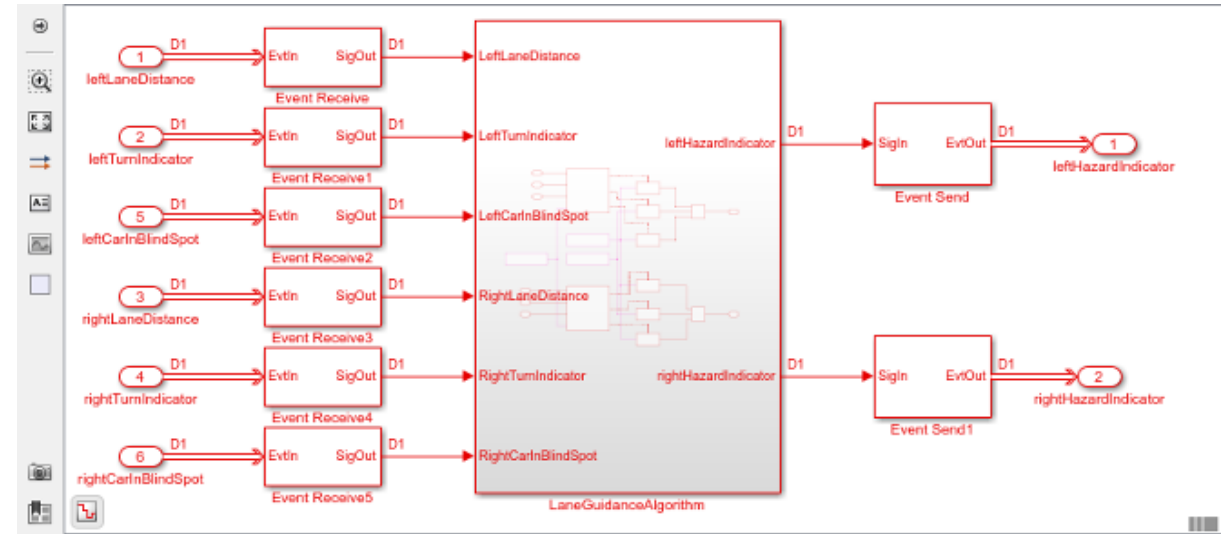
Simulink Messages for Service-oriented communication



You can model service-oriented communication using messages (Send/Receive).

Simulink support for AUTOSAR Adaptive

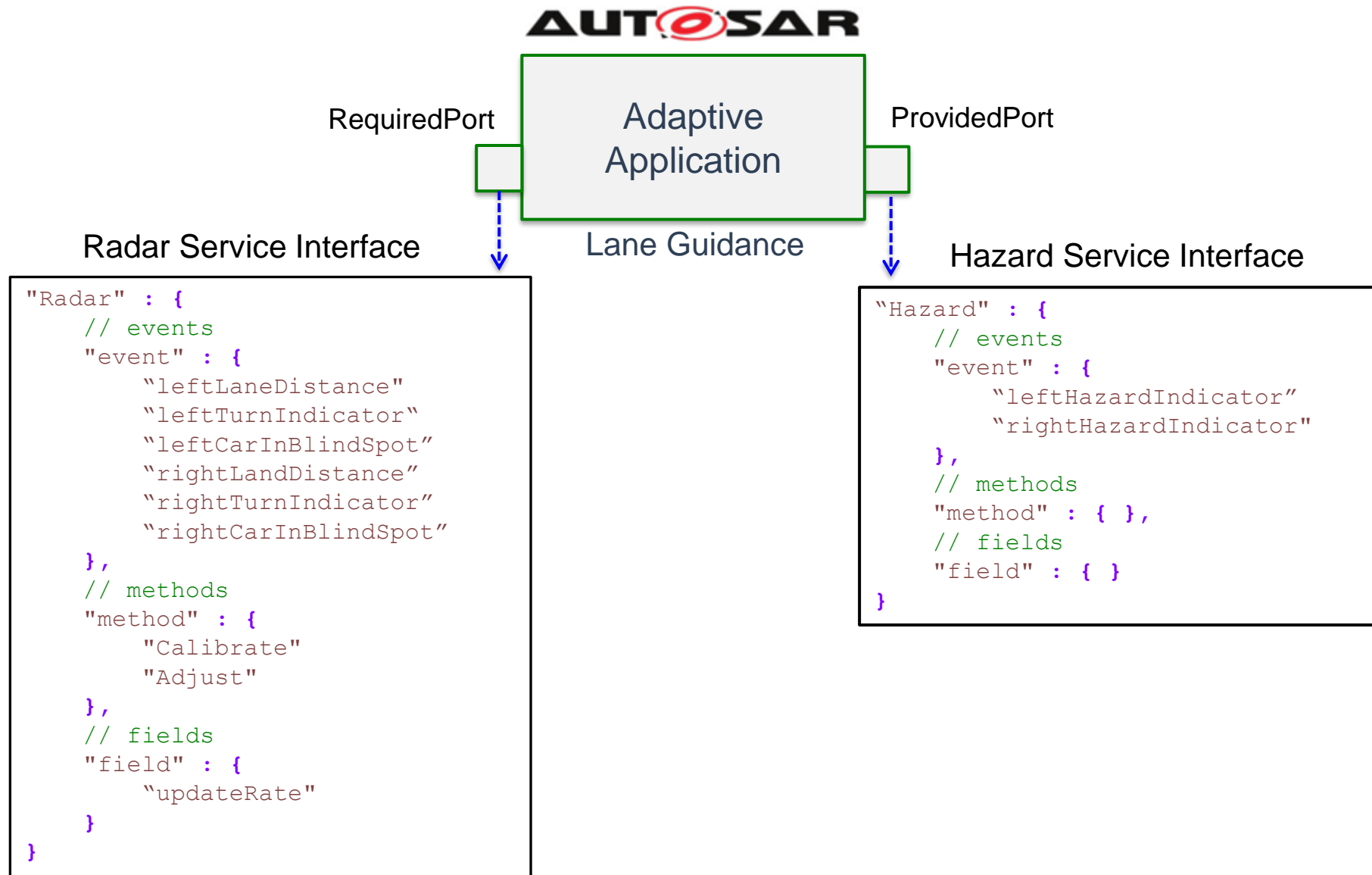
- In AUTOSAR Adaptive, services implement **communication** through:
 - Events
 - Methods
 - Fields
- In Simulink, **Events** can be modeled as **Messages** and then configured for code generation **using AUTOSAR Blockset**.



Name	SwCalibrationAccess	DisplayFormat
leftHazardIndicator	ReadOnly	
rightHazardIndicator	ReadOnly	

AUTOSAR Adaptive C++ compliant code is generated by Embedded Coder.

Adaptive SW architecture concepts



Modelling an AUTOSAR Adaptive application in Simulink

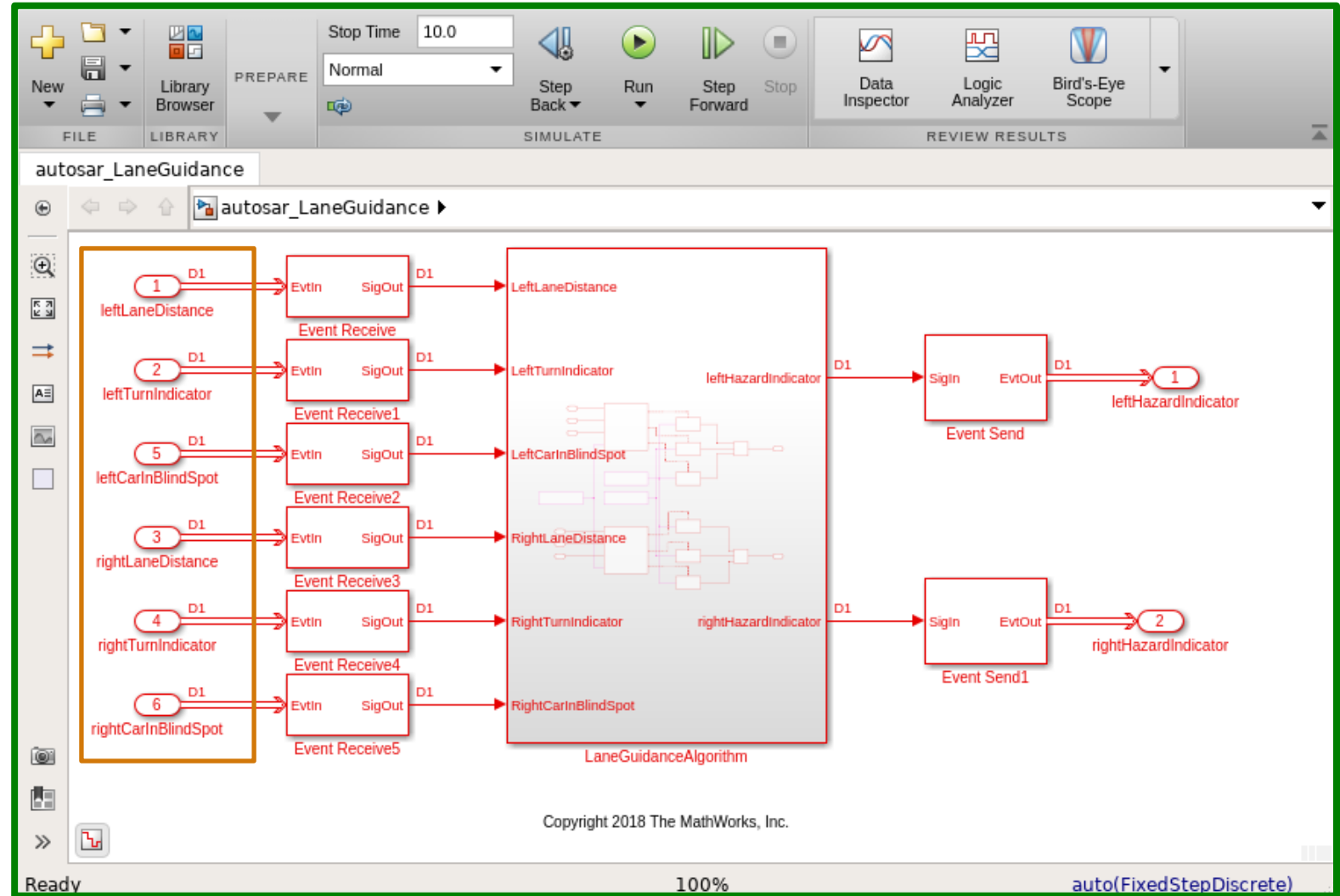


Adaptive Application

RequiredPort

```

"Radar" : {
  // events
  "event" : {
    "leftLaneDistance"
    "leftTurnIndicator"
    "leftCarInBlindSpot"
    "rightLandDistance"
    "rightTurnIndicator"
    "rightCarInBlindSpot"
  },
  // methods
  "method" : {
    "Calibrate"
    "Adjust"
  },
  // fields
  "field" : {
    "updateRate"
  }
}
  
```



Modelling an AUTOSAR Adaptive application in Simulink

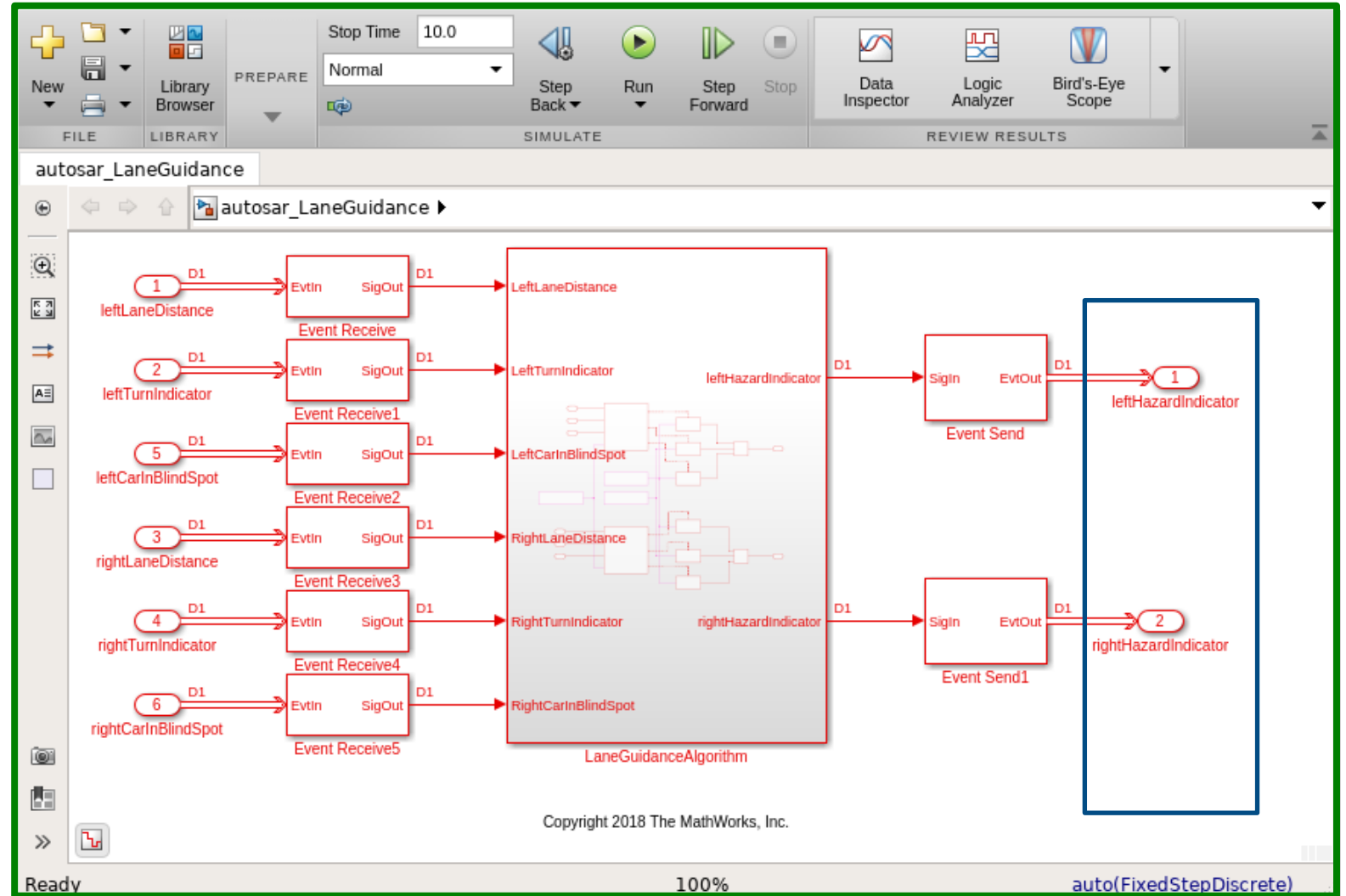


Adaptive Application

ProvidedPort

```

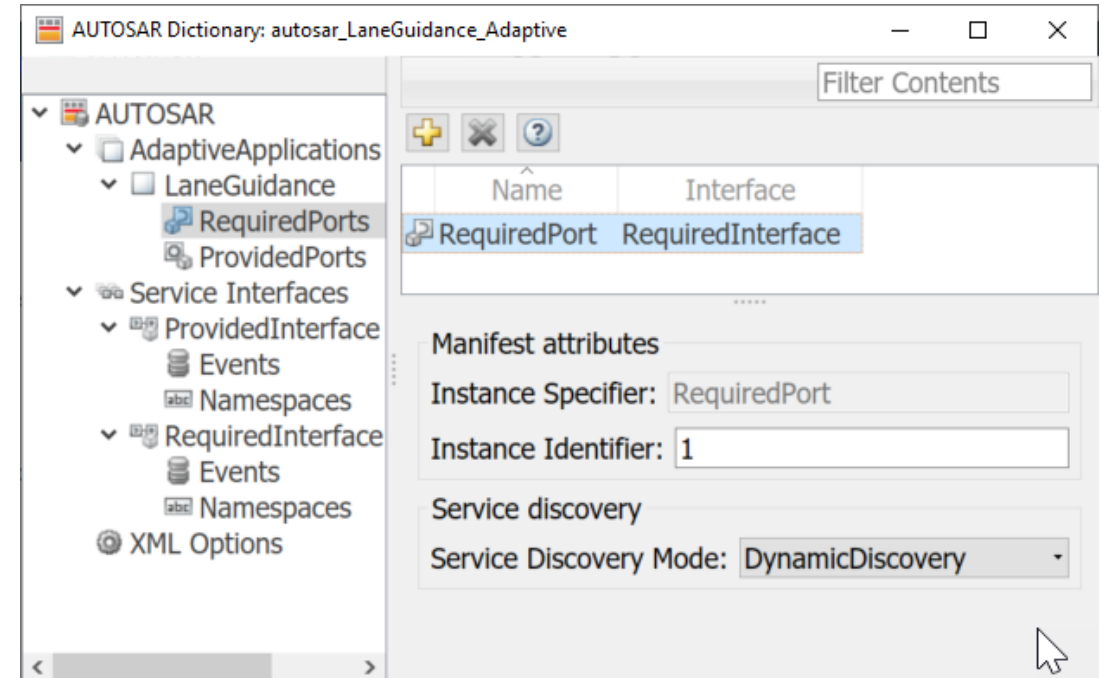
"Hazard" : {
  // events
  "event" : {
    "leftHazardIndicator"
    "rightHazardIndicator"
  },
  // methods
  "method" : { },
  // fields
  "field" : { }
}
    
```



Dynamic Service Discovery

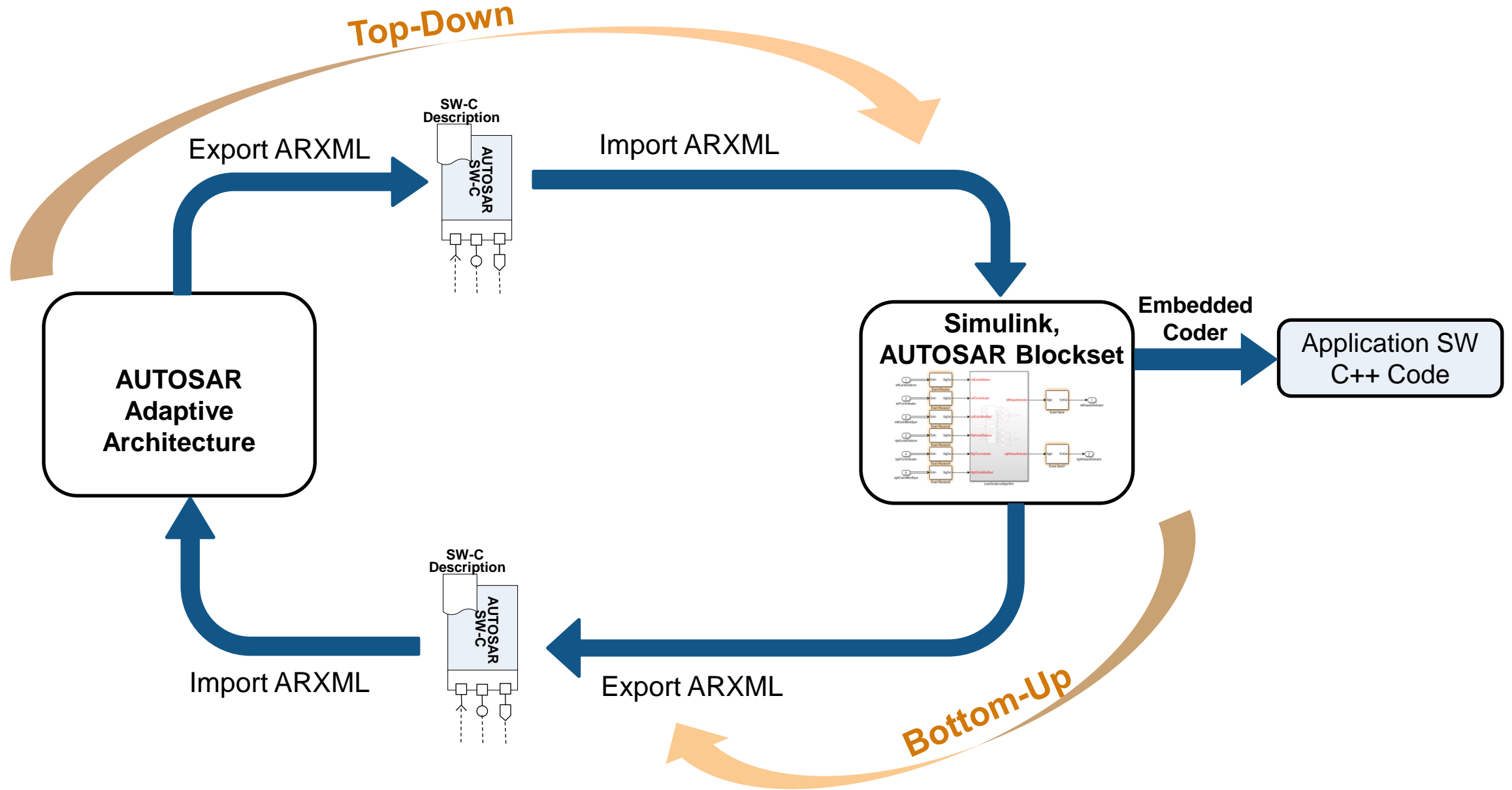
Find adaptive services by using dynamic discovery

- Configure AUTOSAR adaptive applications to **discover and subscribe to adaptive services** as they become available
- You can also configure service port programmatically as OneTime or DynamicDiscovery



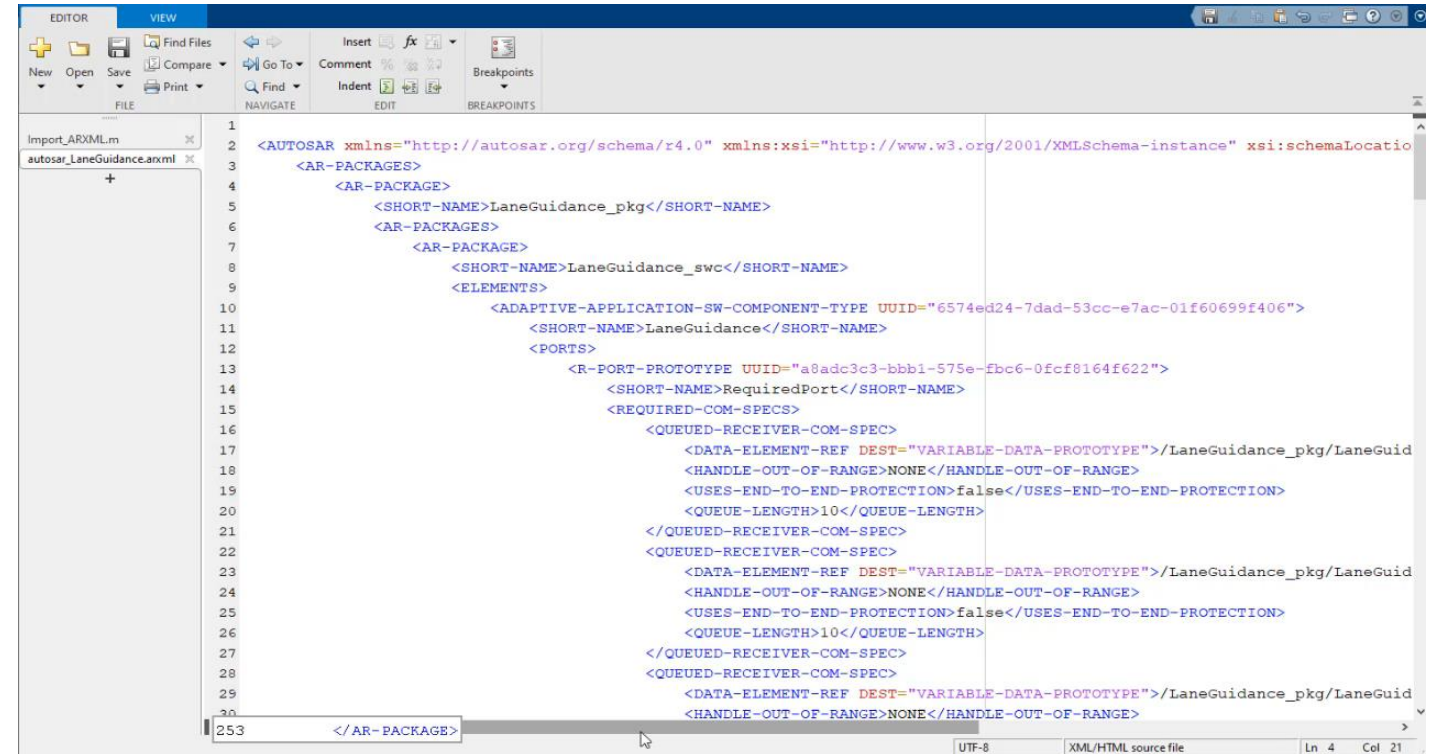
```
apiObj = autosar.api.getAUTOSARProperties("autosar_LaneGuidance");  
apiObj.set("/LaneGuidance_pkg/LaneGuidance_swc/LaneGuidance/RequiredPort/",  
"ServiceDiscoveryMode", "DynamicDiscovery")
```

AUTOSAR Adaptive workflows



AUTOSAR Adaptive in action

- Create model from ARXML
- Verify AUTOSAR properties
- Configure Service Discovery
- Generate code



```

1
2 <AUTOSAR xmlns="http://autosar.org/schema/r4.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocatio
3
4   <AR-PACKAGES>
5     <AR-PACKAGE>
6       <SHORT-NAME>LaneGuidance_pkg</SHORT-NAME>
7       <AR-PACKAGES>
8         <AR-PACKAGE>
9           <SHORT-NAME>LaneGuidance_sw</SHORT-NAME>
10          <ELEMENTS>
11            <ADAPTIVE-APPLICATION-SW-COMPONENT-TYPE UUID="6574ed24-7dad-53cc-e7ac-01f60699f406">
12              <SHORT-NAME>LaneGuidance</SHORT-NAME>
13              <PORTS>
14                <R-PORT-PROTOTYPE UUID="a8adc3c3-bbb1-575e-fbc6-0fcf8164f622">
15                  <SHORT-NAME>RequiredPort</SHORT-NAME>
16                  <REQUIRED-COM-SPECS>
17                    <QUEUED-RECEIVER-COM-SPEC>
18                      <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE"/>LaneGuidance_pkg/LaneGuid
19                      <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
20                      <USES-END-TO-END-PROTECTION>false</USES-END-TO-END-PROTECTION>
21                      <QUEUE-LENGTH>10</QUEUE-LENGTH>
22                    </QUEUED-RECEIVER-COM-SPEC>
23                    <QUEUED-RECEIVER-COM-SPEC>
24                      <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE"/>LaneGuidance_pkg/LaneGuid
25                      <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
26                      <USES-END-TO-END-PROTECTION>false</USES-END-TO-END-PROTECTION>
27                      <QUEUE-LENGTH>10</QUEUE-LENGTH>
28                    </QUEUED-RECEIVER-COM-SPEC>
29                    <QUEUED-RECEIVER-COM-SPEC>
30                      <DATA-ELEMENT-REF DEST="VARIABLE-DATA-PROTOTYPE"/>LaneGuidance_pkg/LaneGuid
31                      <HANDLE-OUT-OF-RANGE>NONE</HANDLE-OUT-OF-RANGE>
32                    </QUEUED-RECEIVER-COM-SPEC>
33                  </REQUIRED-COM-SPECS>
34                </R-PORT-PROTOTYPE>
35              </PORTS>
36            </ADAPTIVE-APPLICATION-SW-COMPONENT-TYPE>
37          </ELEMENTS>
38        </AR-PACKAGE>
39      </AR-PACKAGES>
40    </AR-PACKAGE>
41  </AR-PACKAGES>
  
```

Agenda

- SOA: a new paradigm for automotive software
- Simulink for Service-Oriented Applications
- Key take-aways

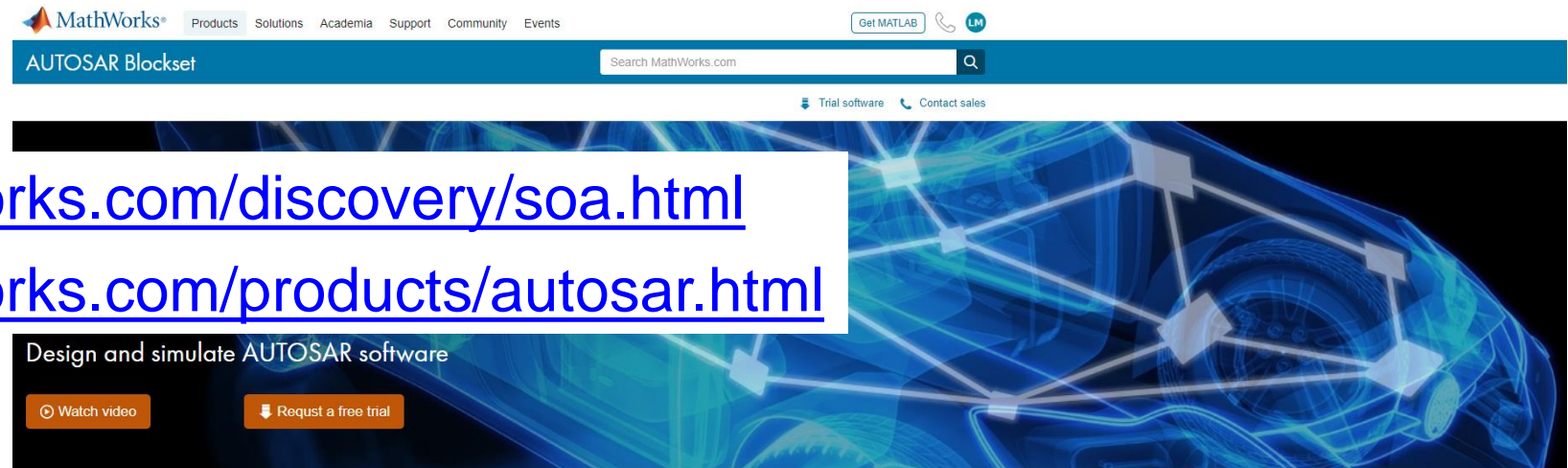
Key take-aways

- Automotive E/E and SW architectures are changing dramatically
- Service-oriented architectures are foundation for dynamic SW configuration, updates and event-driven communication
- **Leverage the power of Model-Based Design to model, simulate and deploy SOA applications compatible with AUTOSAR Adaptive, ROS and DDS.**

Additional resources

- Learn more :

- <https://www.mathworks.com/discovery/soa.html>
- <https://www.mathworks.com/products/autosar.html>

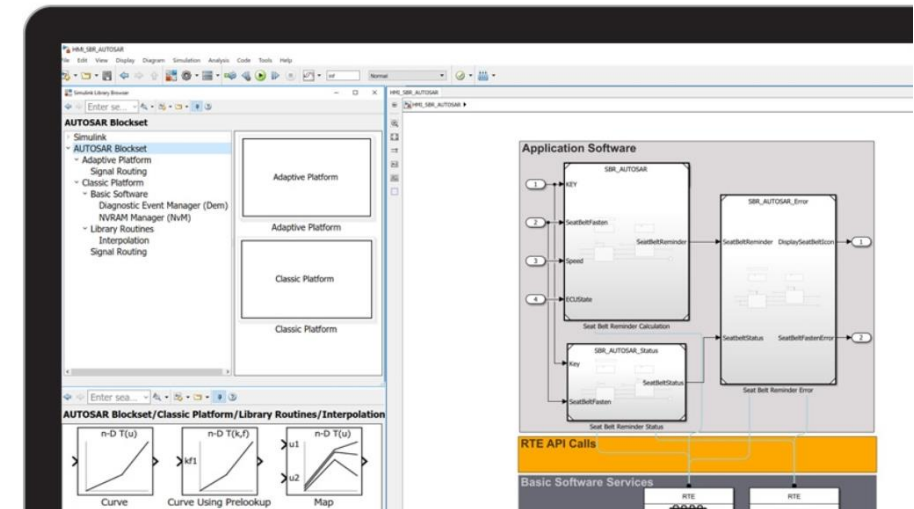


Examples and How To

- Message-Based Communication Between Software Components - Example
- Run-Time Software Modeling (5 Videos) - Video Series
- Model AUTOSAR Adaptive Software Components - Example
- Configure AUTOSAR Adaptive Service Communication - Example

(requires System Composer™). In the AUTOSAR architecture model, you can author software compositions, components, and interfaces. You can add simulation behavior including Basic Software service components. Alternatively, you can round-trip (import and export) software descriptions via ARXML files.

AUTOSAR Blockset supports C and C++ production code generation (with Embedded Coder®). It is qualified for use with the ISO 26262 standard (with IEC Certification Kit).



Presenter contact info and poll questions

Please contact us with questions



Luigi Milia
Automotive Industry Manager
lmilia@mathworks.com



Shwetha Bhadravathi Patil
AUTOSAR Product Manager
shwethap@mathworks.com

- Are you already working on SOA based applications?
- Do you plan to work on SOA based applications in the near future?
- Do you plan to use MBD for SOA?
- Are you interested in a follow-up conversation with MathWorks?