MathWorks
AUTOMOTIVE CONFERENCE 2022
India

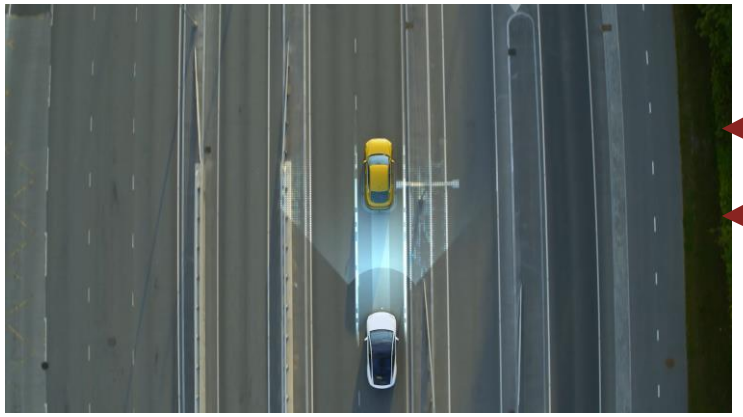
Software-Defined Vehicles: Workflows for In-Car and Cloud Applications

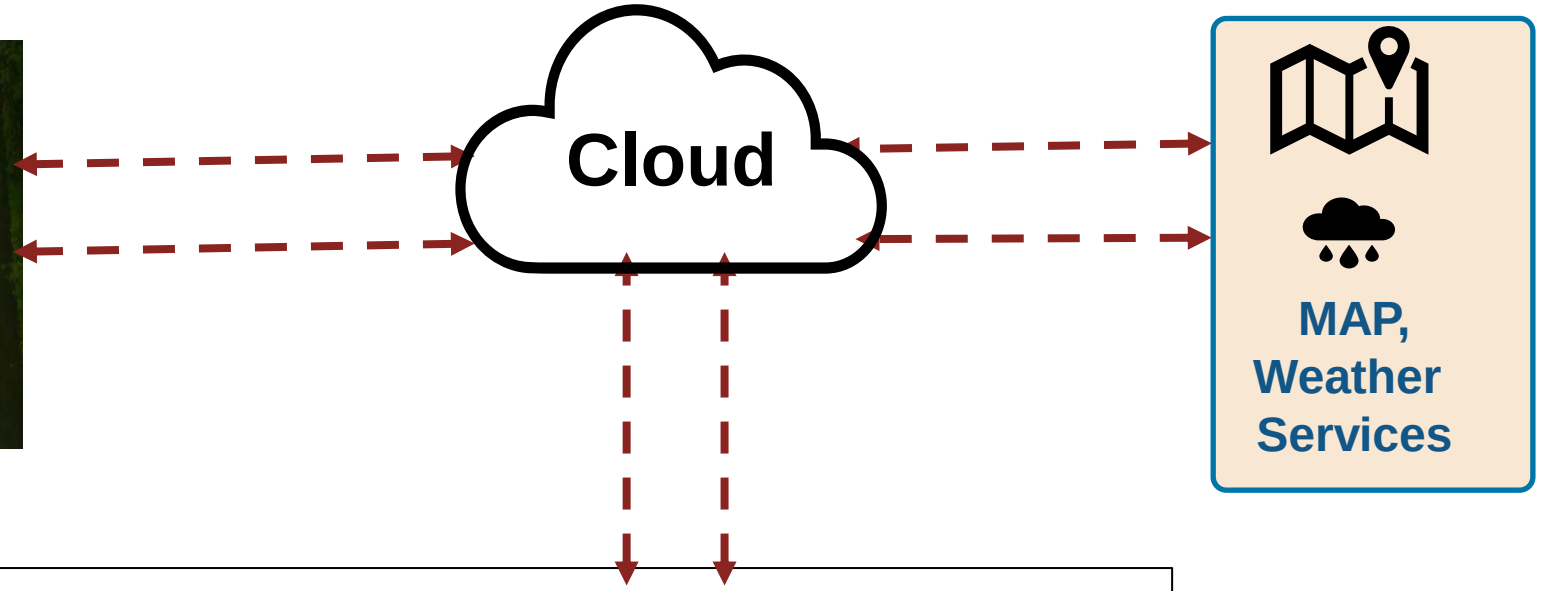
Prasanna Deshpande, MathWorks

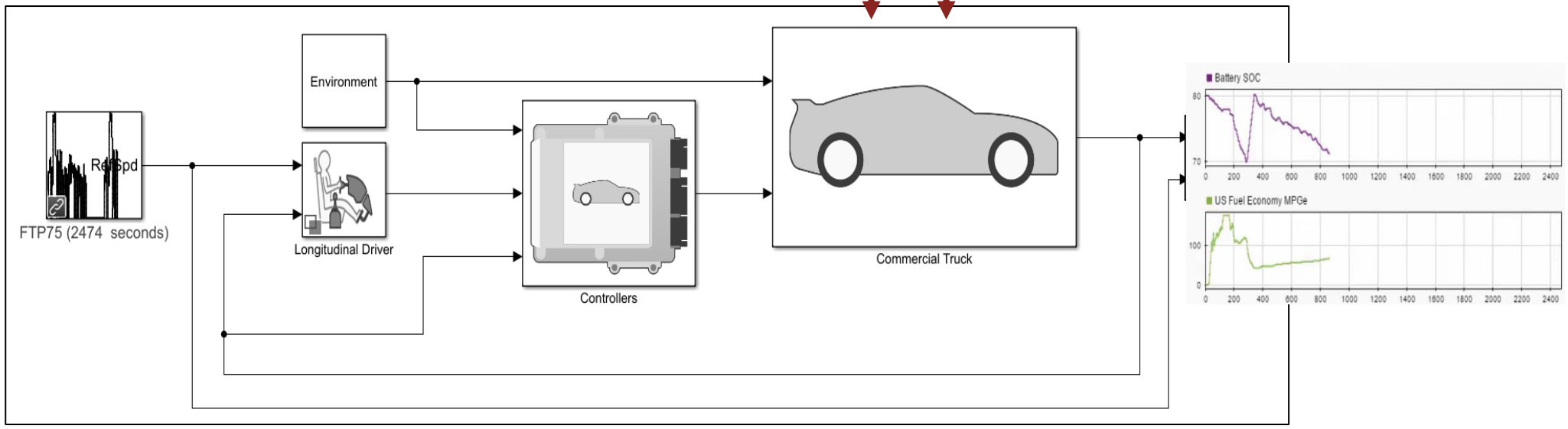
Nukul Sehgal, MathWorks

MathWorks

Application design for Autonomous Electric Vehicle







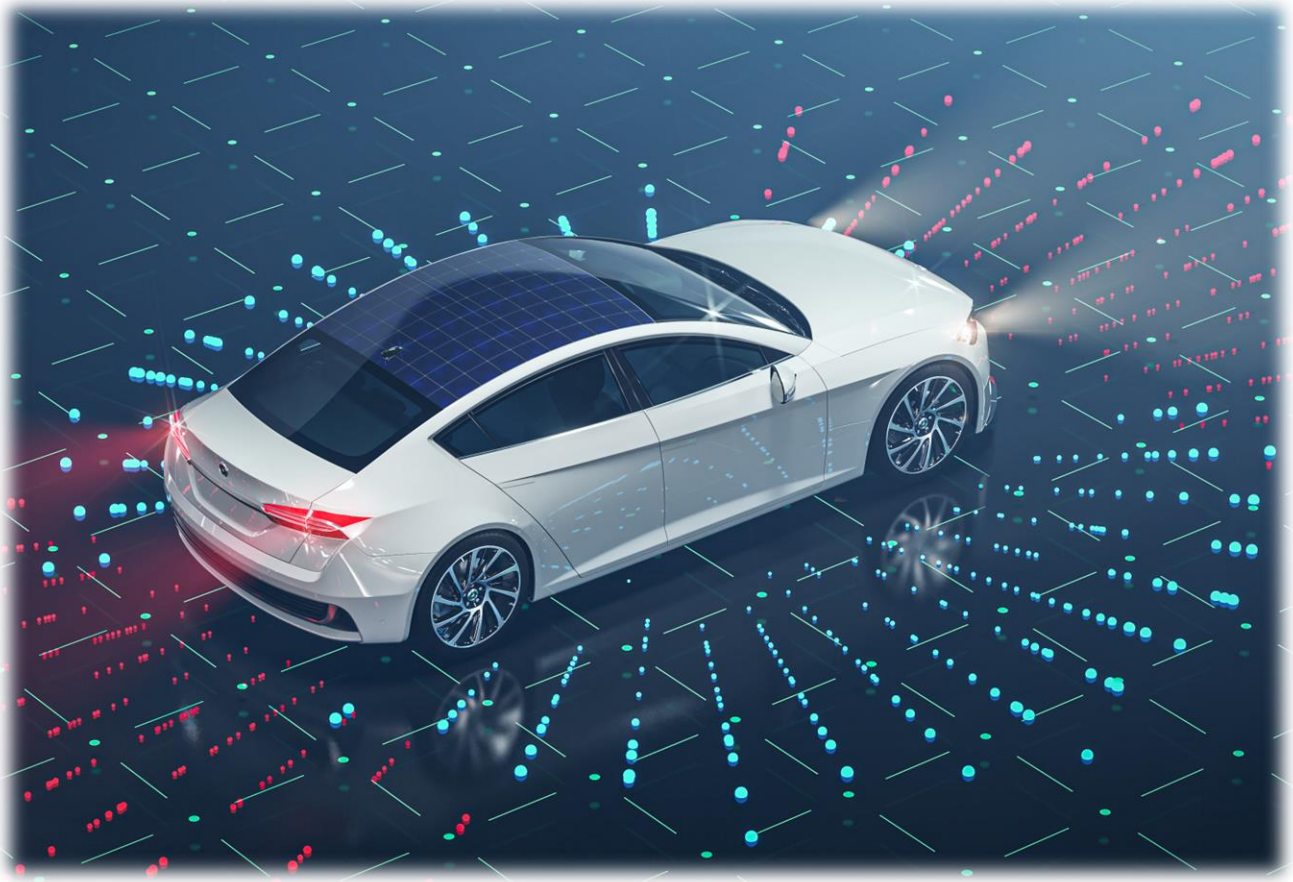
MathWorks
AUTOMOTIVE CONFERENCE 2022
India

Software-Defined Vehicles: Workflows for In-Car and Cloud Applications

Prasanna Deshpande, MathWorks

Nukul Sehgal, MathWorks

MathWorks

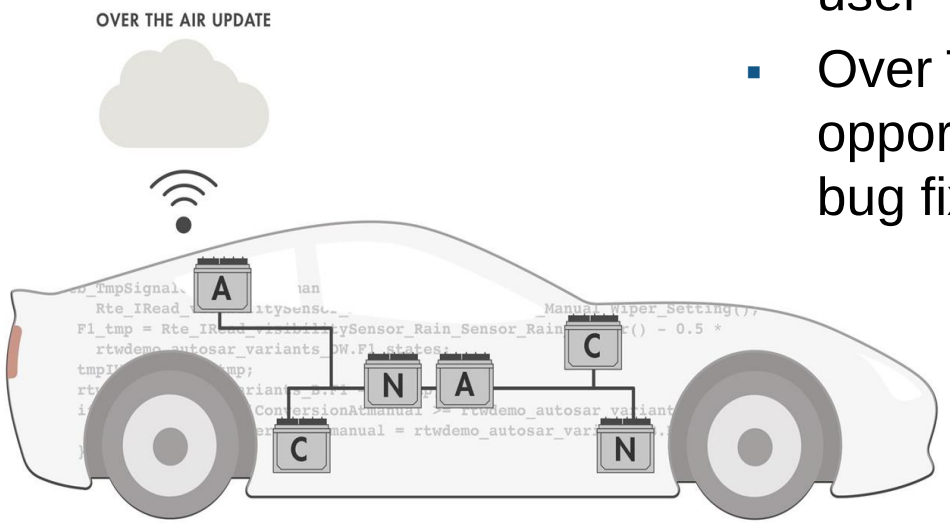


Brand-distinctive features and main value for the vehicle will come from Software

Value of Software-Defined Vehicle

For the user

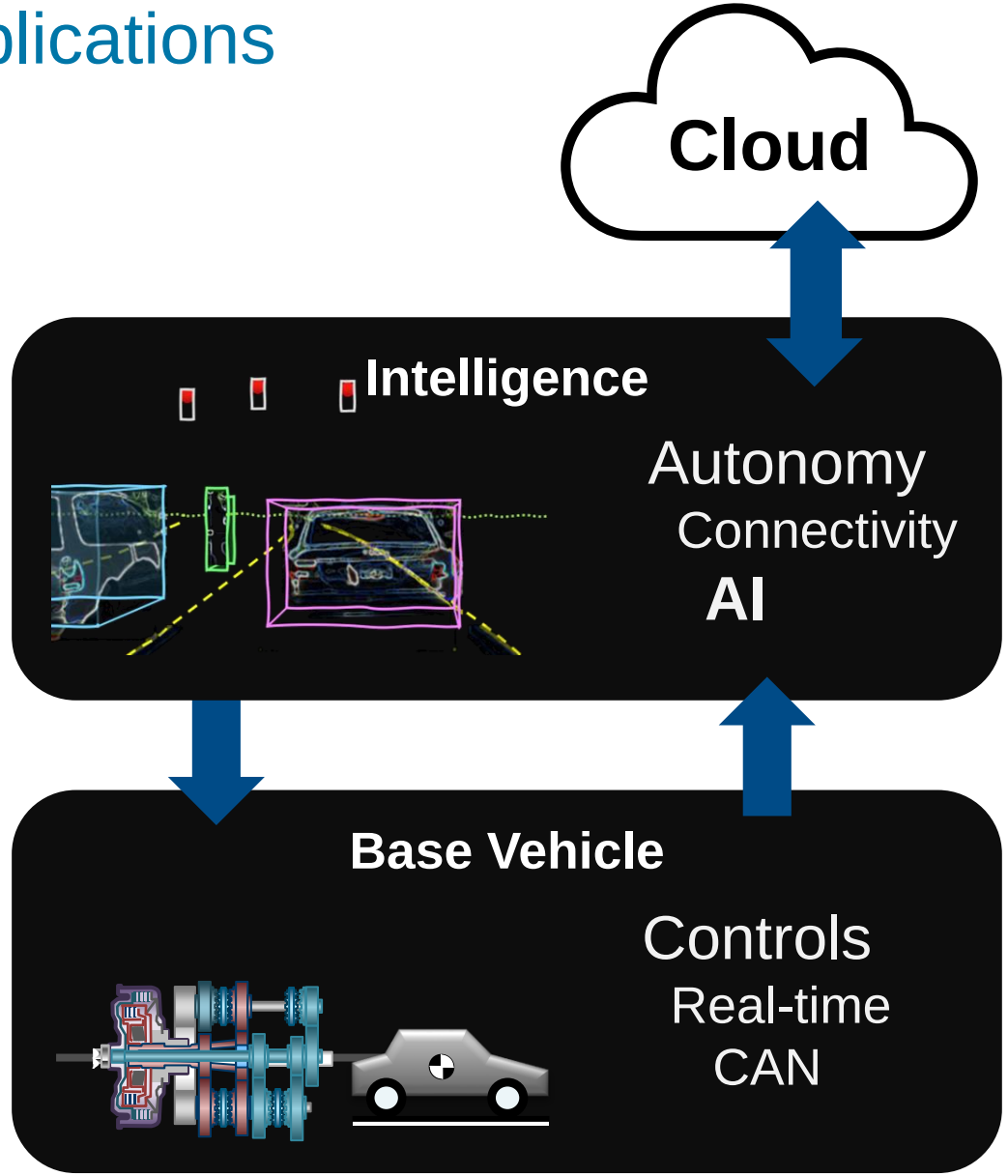
- A vehicle that can have features added to and removed from it, throughout its life
- My vehicle is tailored for my needs
- Smartphone on wheels
- Better and better for me as it collects my data, I get features that add value to me



For the maker

- Build a modular and scalable software platform
- Use the vehicle captured data creatively to re-fine the vehicle design
- Extract data from the vehicle and monetize it for adding extra value to the user
- Over The Air updates of the software – opportunity to provide new features and bug fixes without re-calls

Software-Defined Vehicles: Workflows for In-Car and Cloud Applications



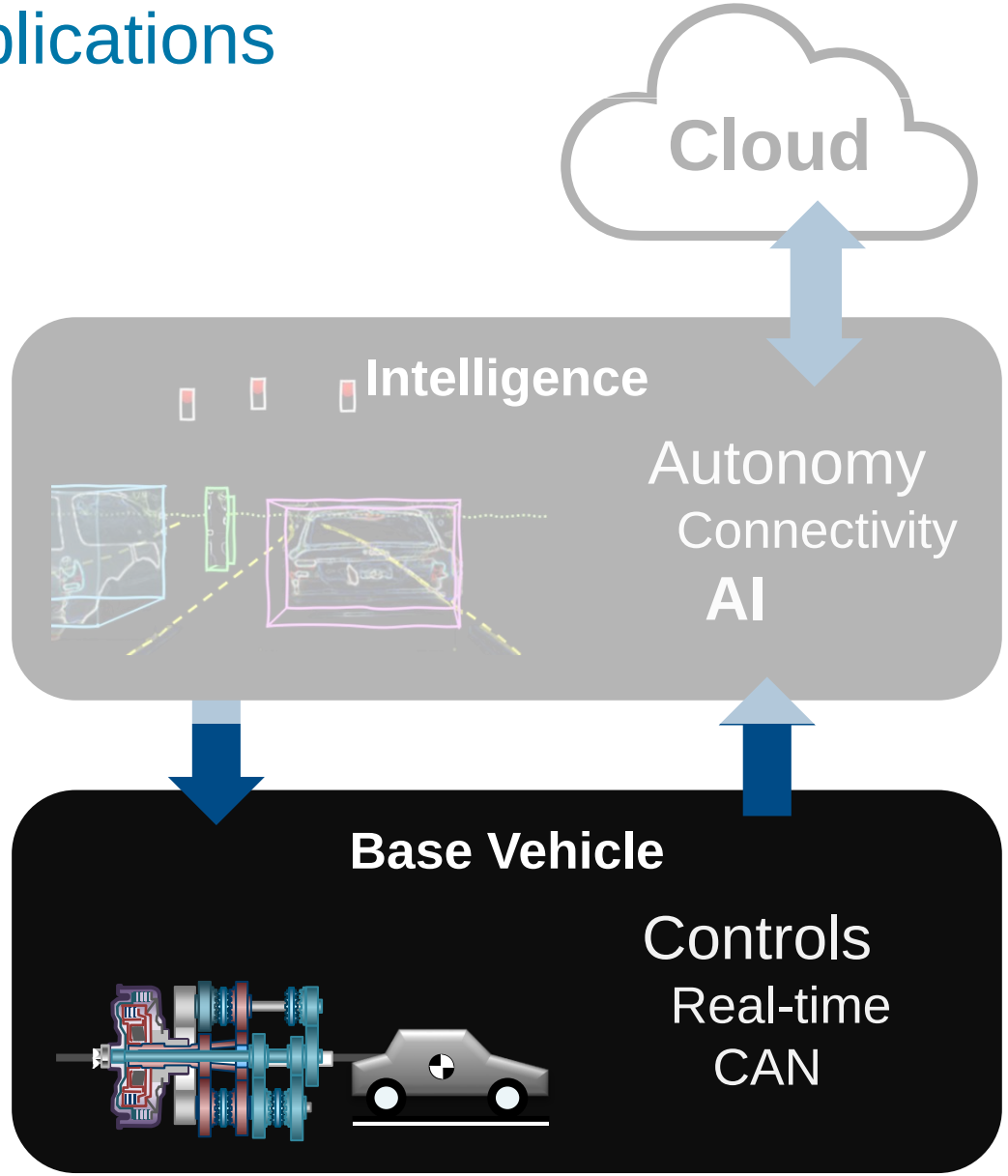
- Business Applications
- Data Handling and Monitoring

- OTA Updates

- Intelligence using Machine Learning and Deep Learning

- Service oriented Software Architecture
- Evolving hardware architectures

Software-Defined Vehicles: Workflows for In-Car and Cloud Applications



- Business Applications
- Data Handling and Monitoring

- OTA Updates

- Intelligence using Machine Learning and Deep Learning

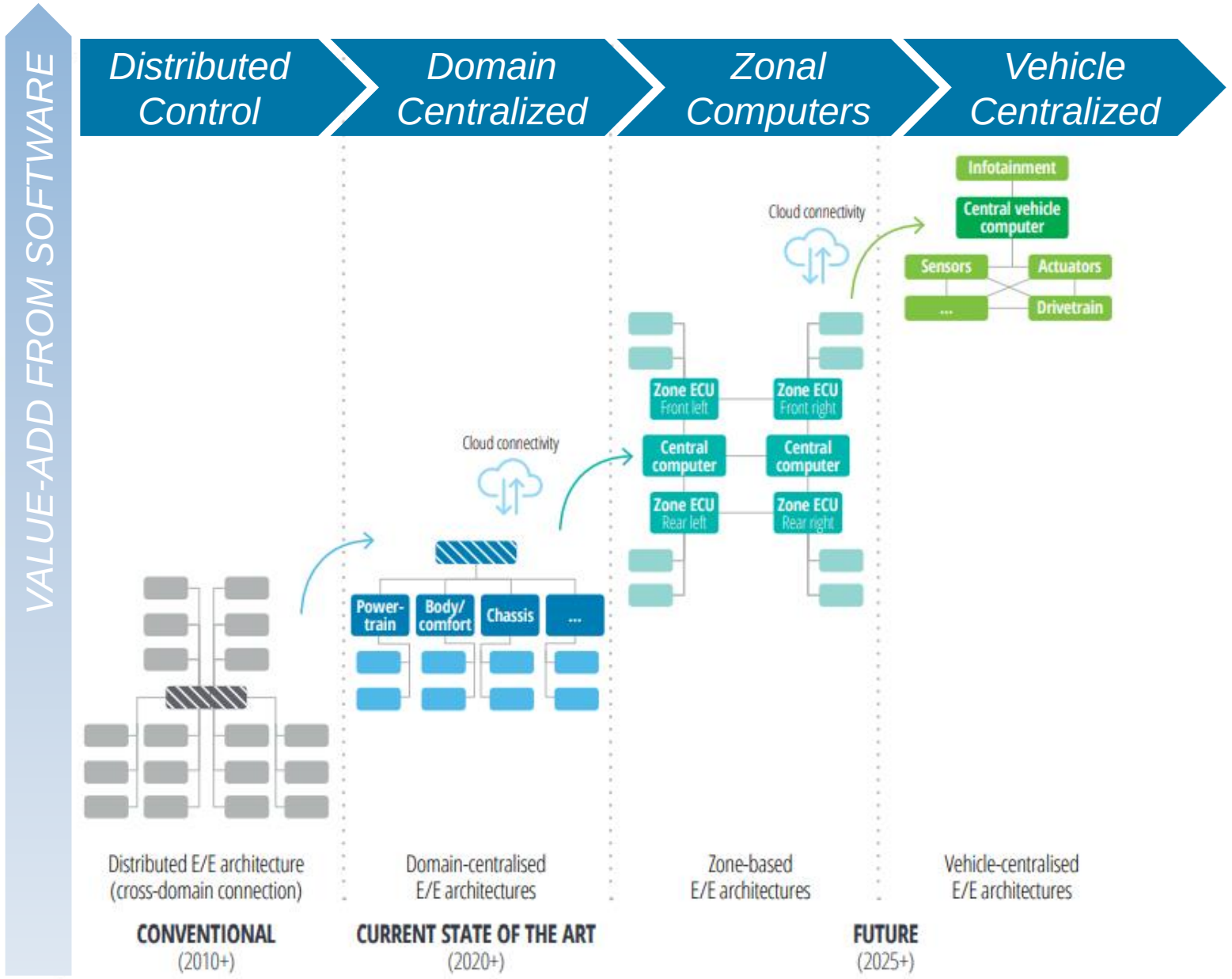
- Service oriented Software Architecture
- Evolving hardware architectures

Evolving Architectures

Need to consolidate E&E architectures
with rising software complexity

Endgame: One central or few Zone Oriented *domain Independent* controllers

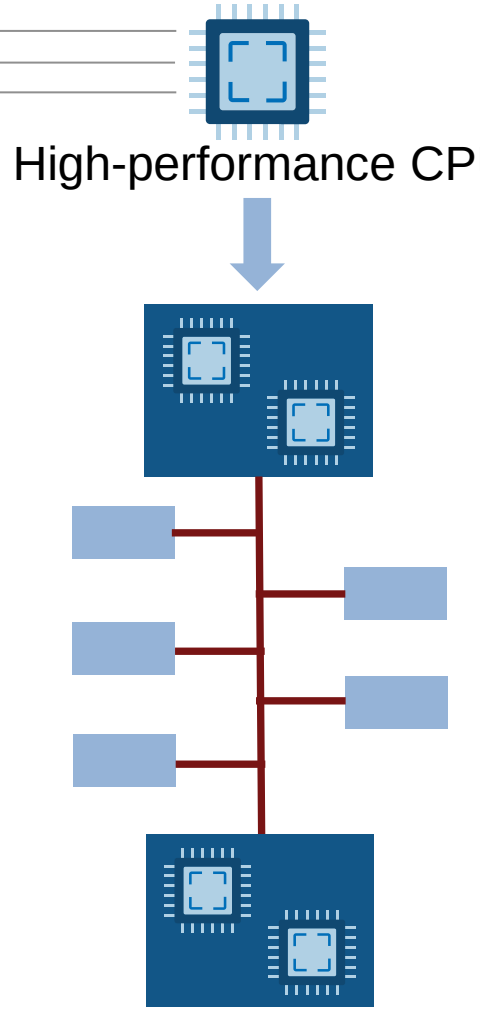
Model, simulate and deploy service-oriented applications



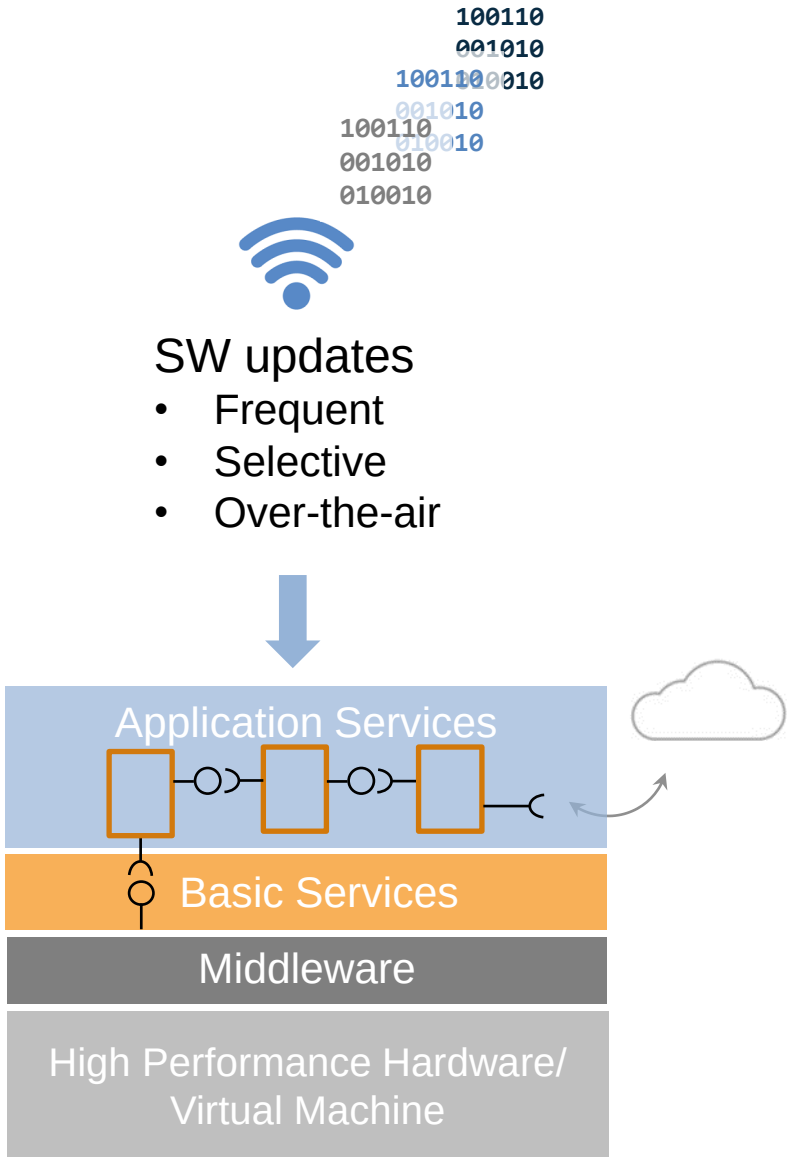
Source: *Deloitte*

Abstraction and new architectures

Exponential growth of SW features

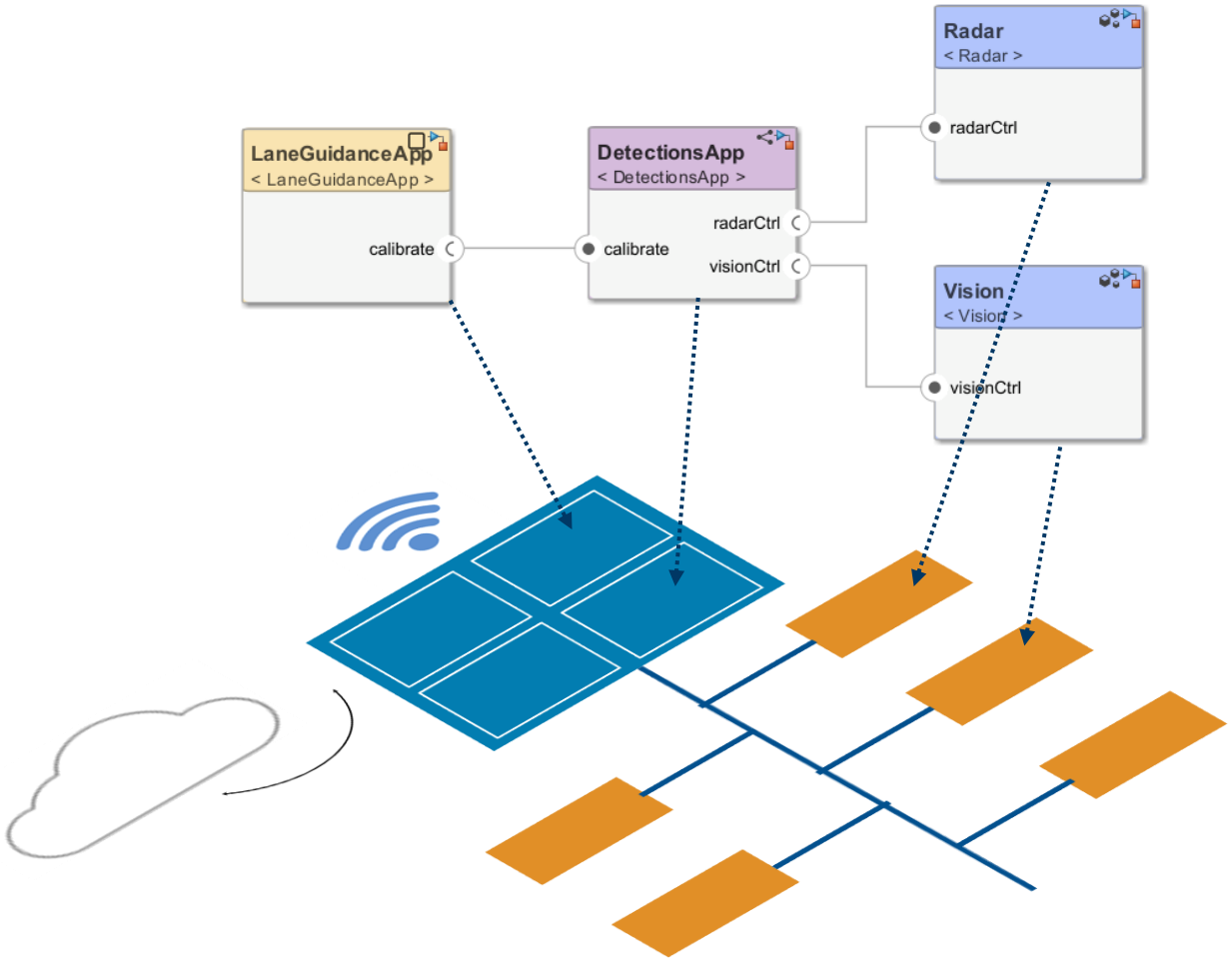


New E/E architectures:
Vehicle and Zone Controllers



Higher HW abstraction:
Service-oriented architectures

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	LaneGuidanceApp_main
2	DetectionsApp.calibrate.Calibrate
3	Radar.radarCtrl.Adjust
4	Vision.visionCtrl.Adjust
5	DetectionsApp.radarResponse
6	DetectionsApp.visionResponse
7	LaneGuidanceApp.detectionsResponse

Service-oriented architectures (SOA) with Model-Based Design

Simulink is evolving to address the changing architectures

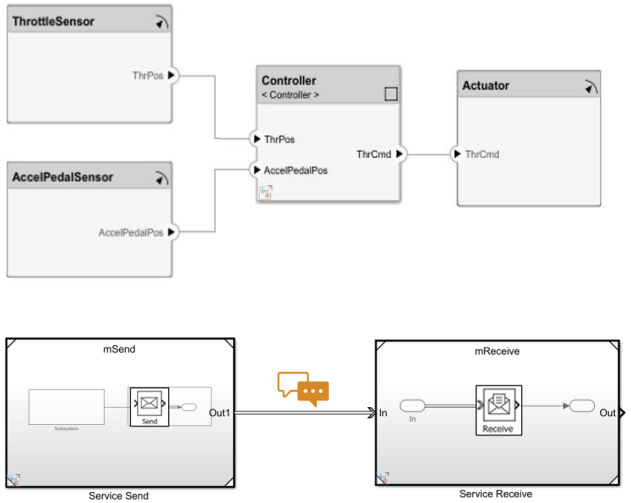
Modeling

Simulation

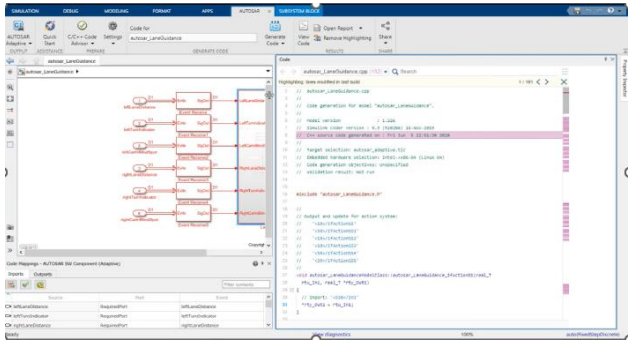
+

Automation

Coding | Verification



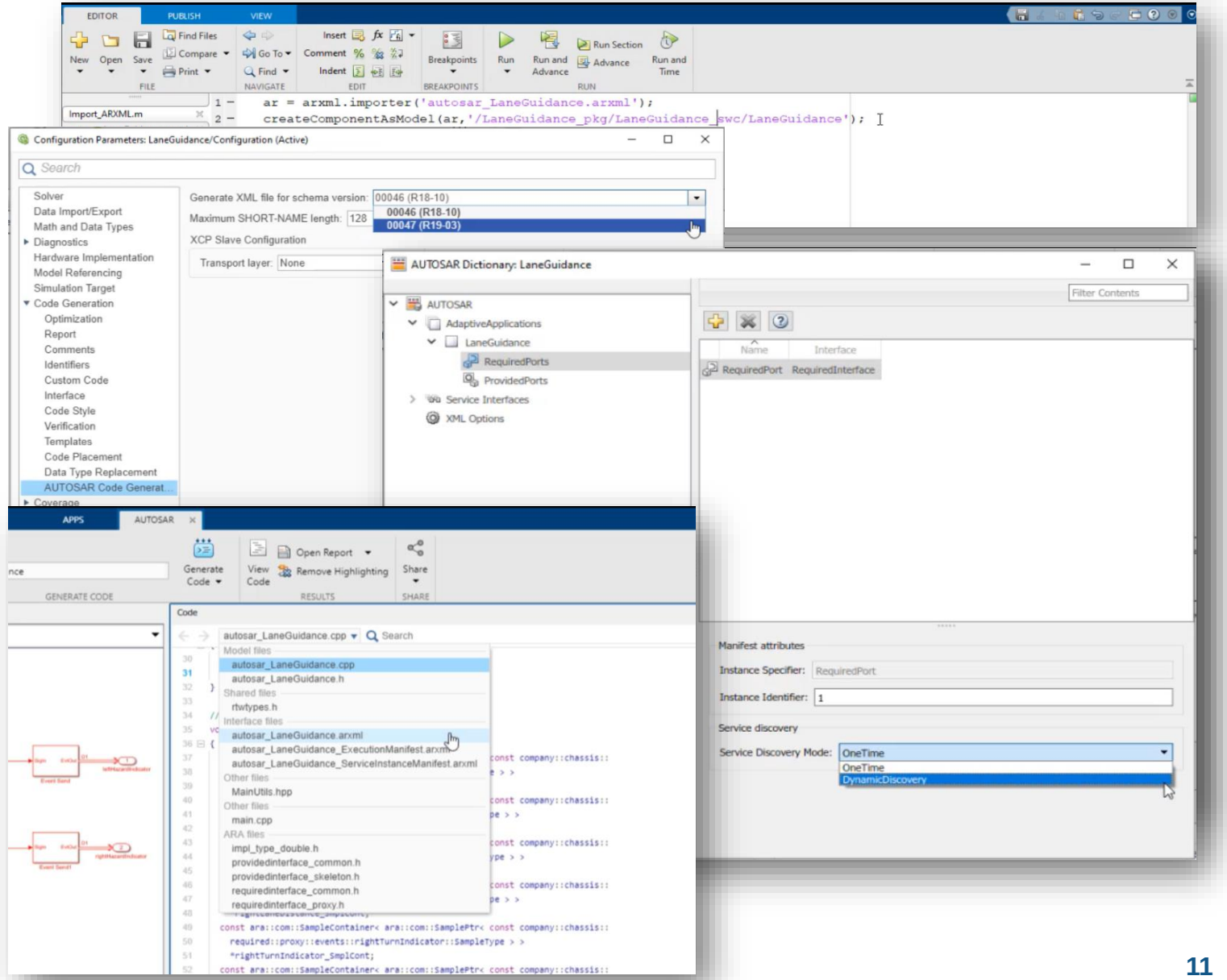
- **Fast design iterations** of service-oriented architectures and applications
- **Maximize reuse** of existing skills and assets
- **Ensure traceability** across all the stages



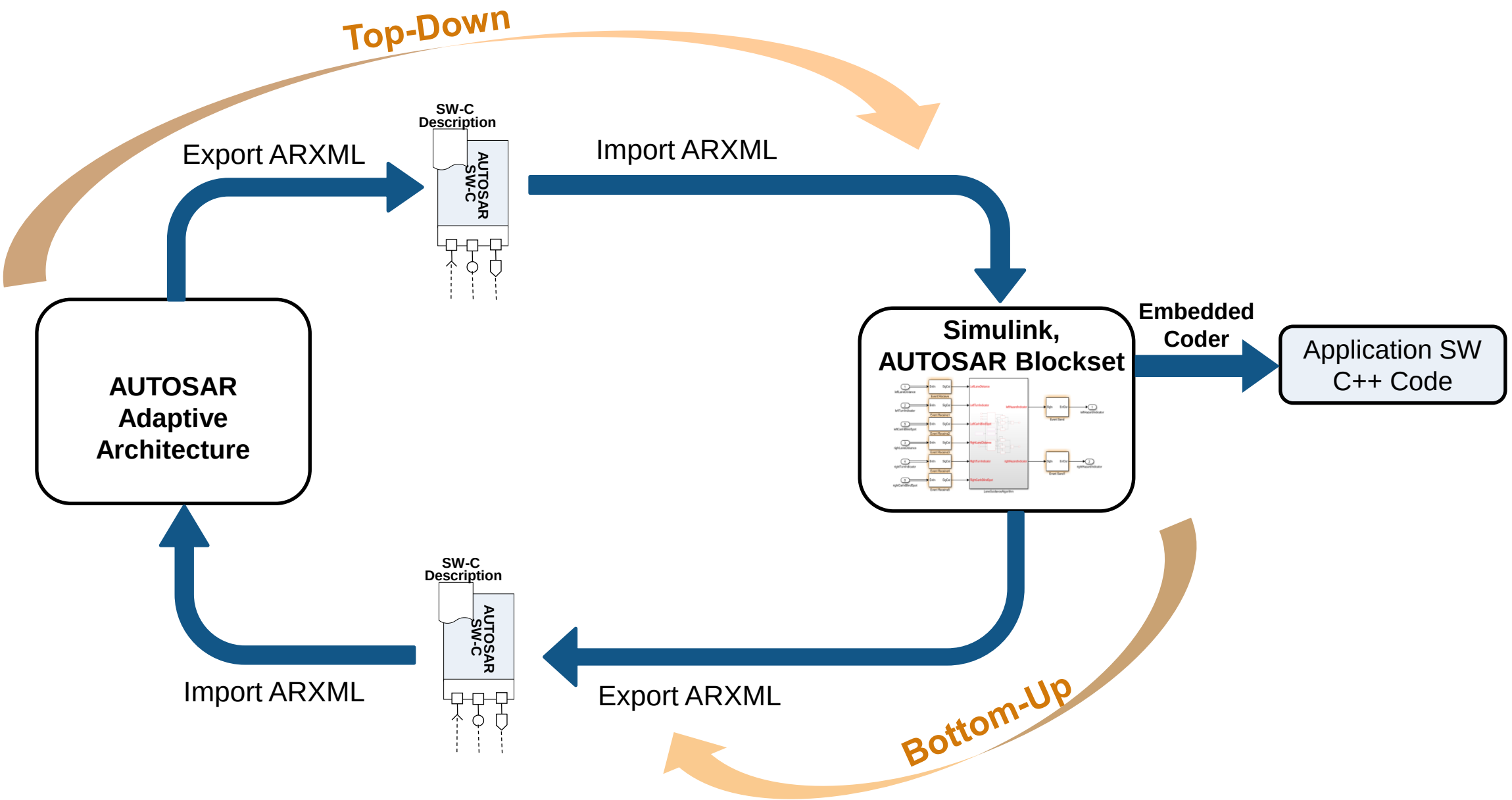
- **Generate code** compliant to automotive standards
- Deploy to **multiple targets**
- Enable **continuous integration**

AUTOSAR Adaptive in action

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



AUTOSAR Adaptive workflows



DDS Blockset

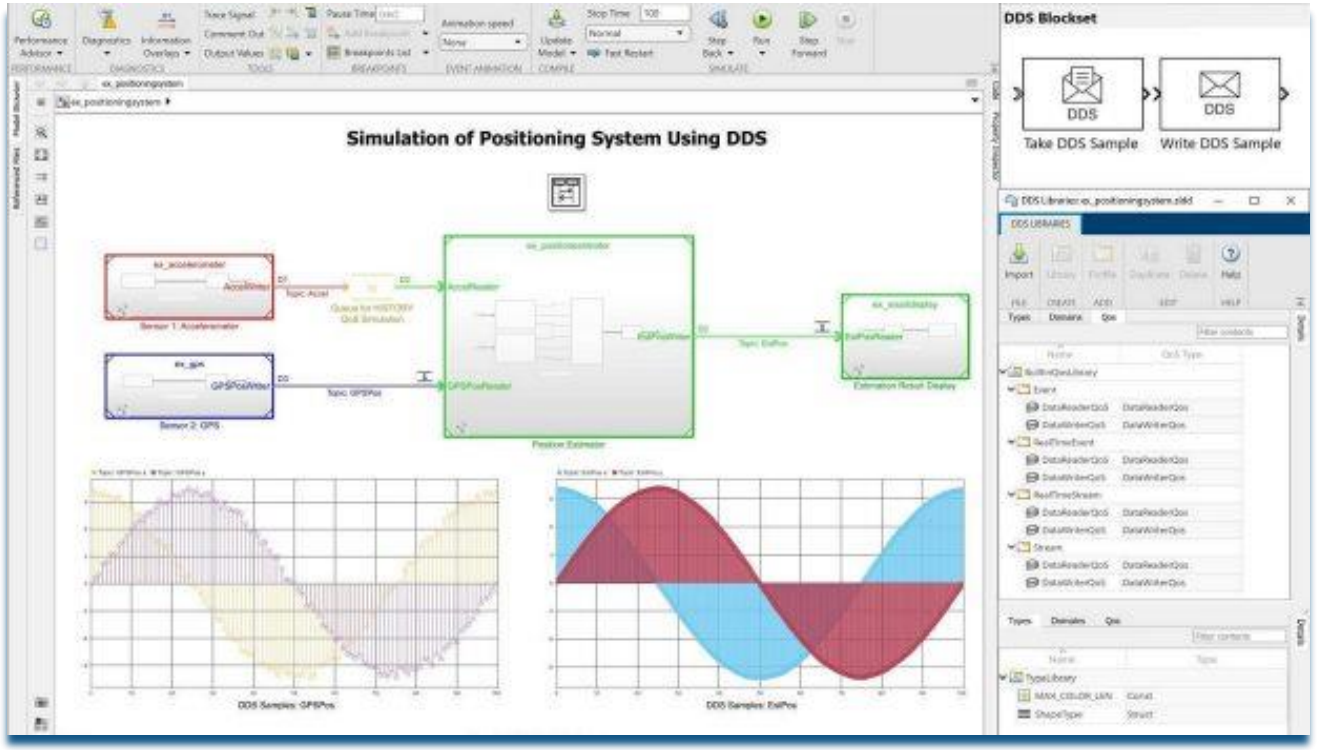
Design and simulate DDS applications

Data Distribution Services (DDS) is a middleware standard uses SOA methodology

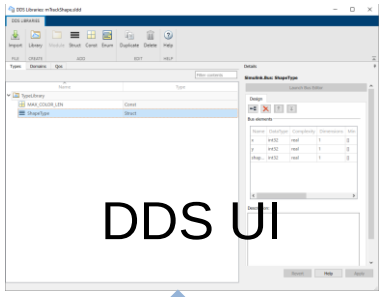
Develop software for DDS based embedded systems in Simulink

DDS Blockset provides	• Apps and blocks to model and simulate DDS software applications that publish or subscribe to DDS and their QoS • DDS dictionary to manage DDS definitions • API's to Import and Export DDS libraries • C++ production code generation with DDS APIs (with Embedded Coder)

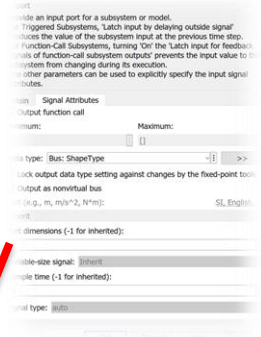
DDS Blockset fully integrates with third-party DDS stacks including RTI Connext and eProsima Fast DDS



DDS Workflow - User Workflow with UI Steps

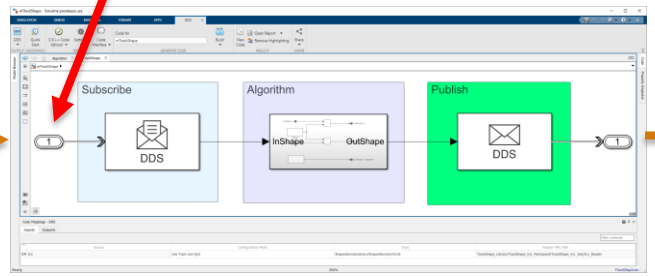


DDS UI

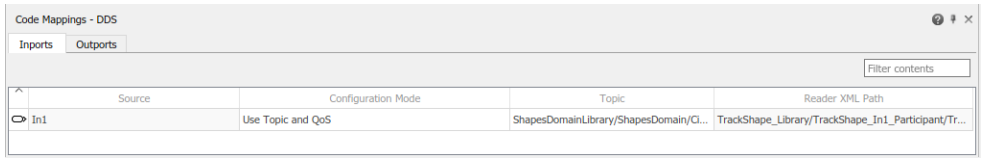


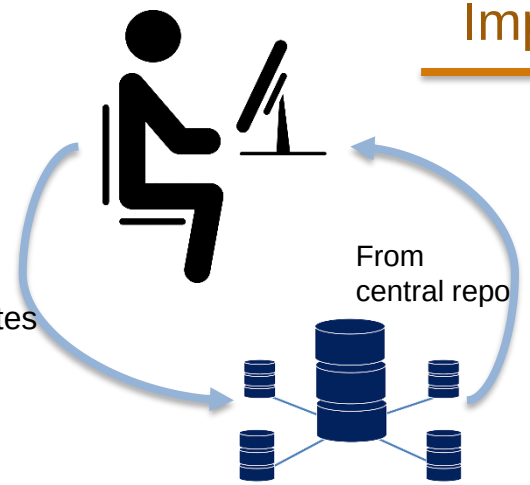
Import

Simulink Data Dictionary



Generate





Creates

From central repo

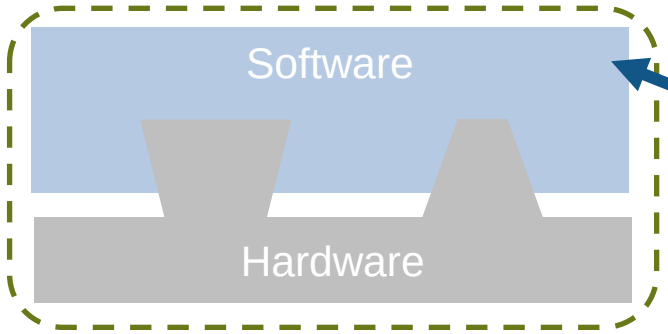
Interface definitions
QoS definitions
(XML)

Deploy

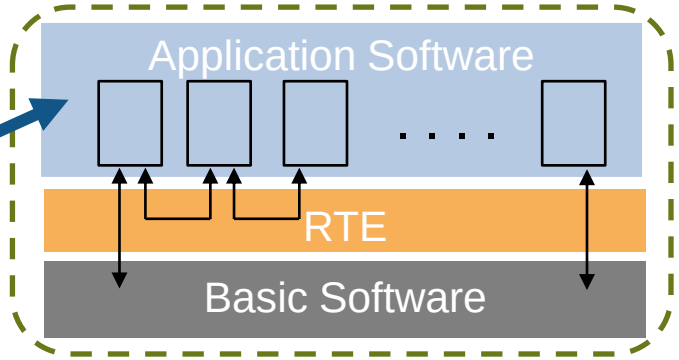
IDL/XML
+
Vendor support

DDS Databus

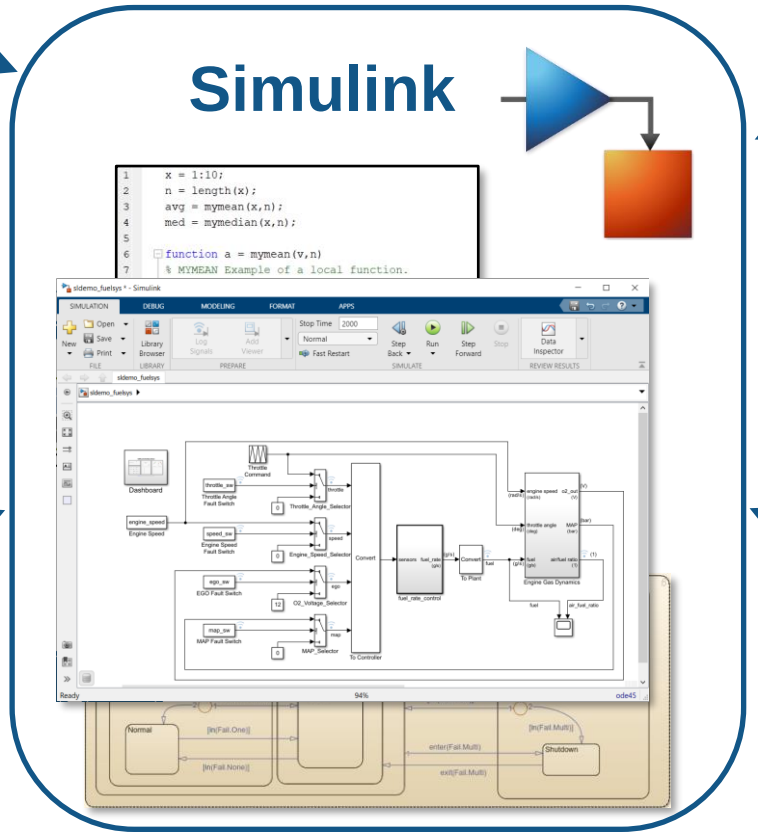
Simulink : design software once, deploy to many targets

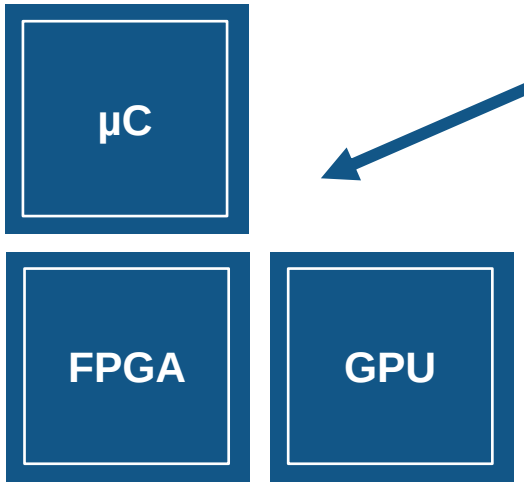


Legacy ECU (C)

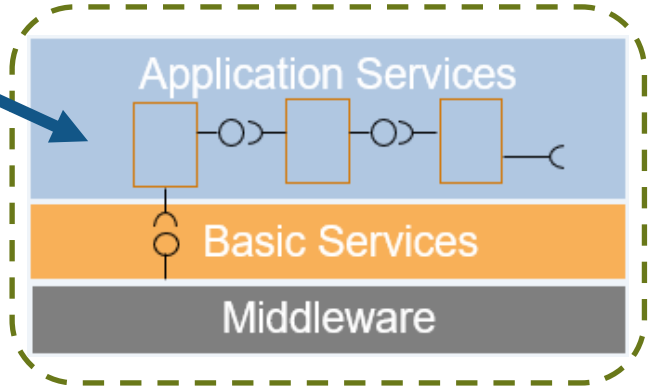


AUTOSAR Classic (C)



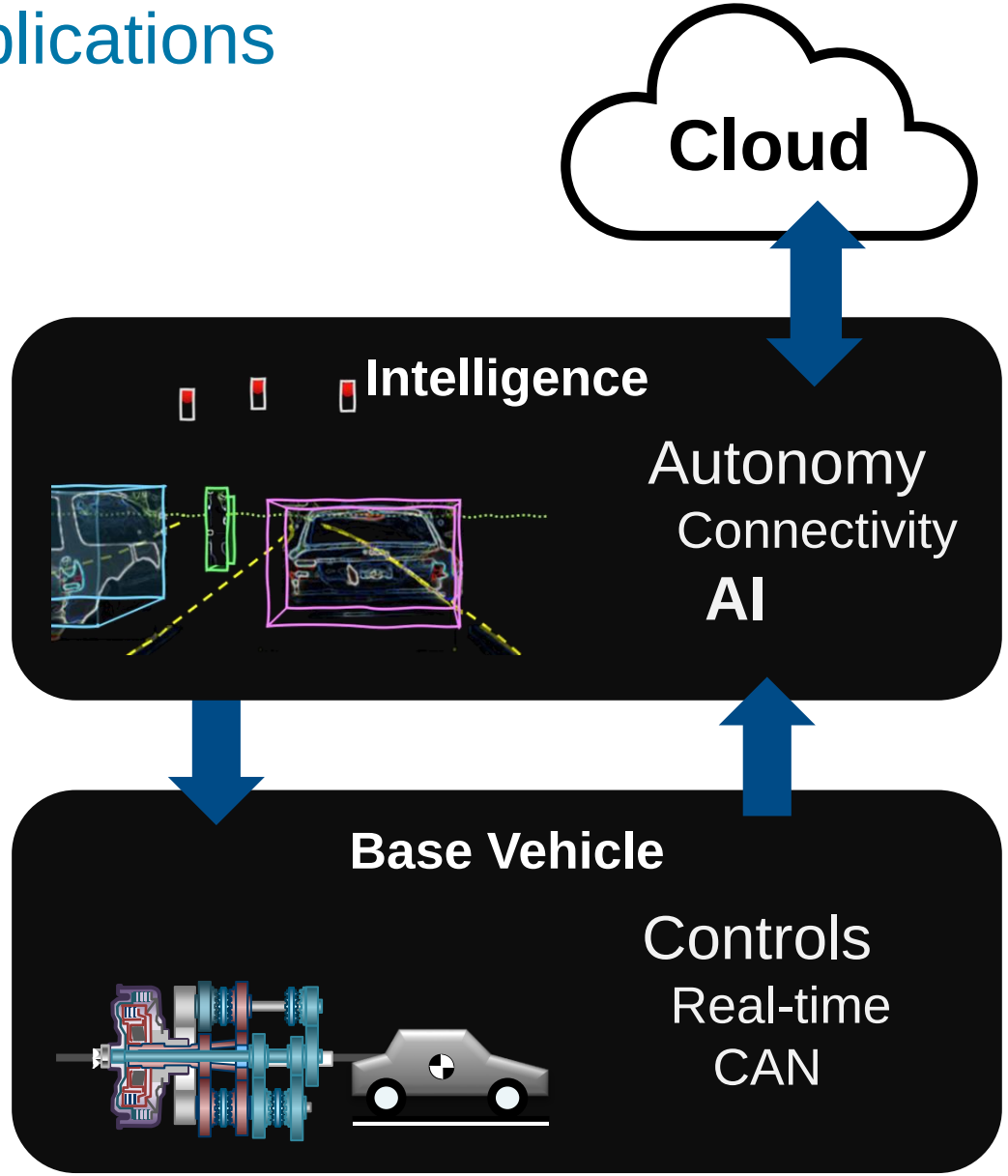


High Performance Processors (C/C++, cuda, HDL)



AUTOSAR Adaptive / ROS / DDS (C++)

Software-Defined Vehicles: Workflows for In-Car and Cloud Applications



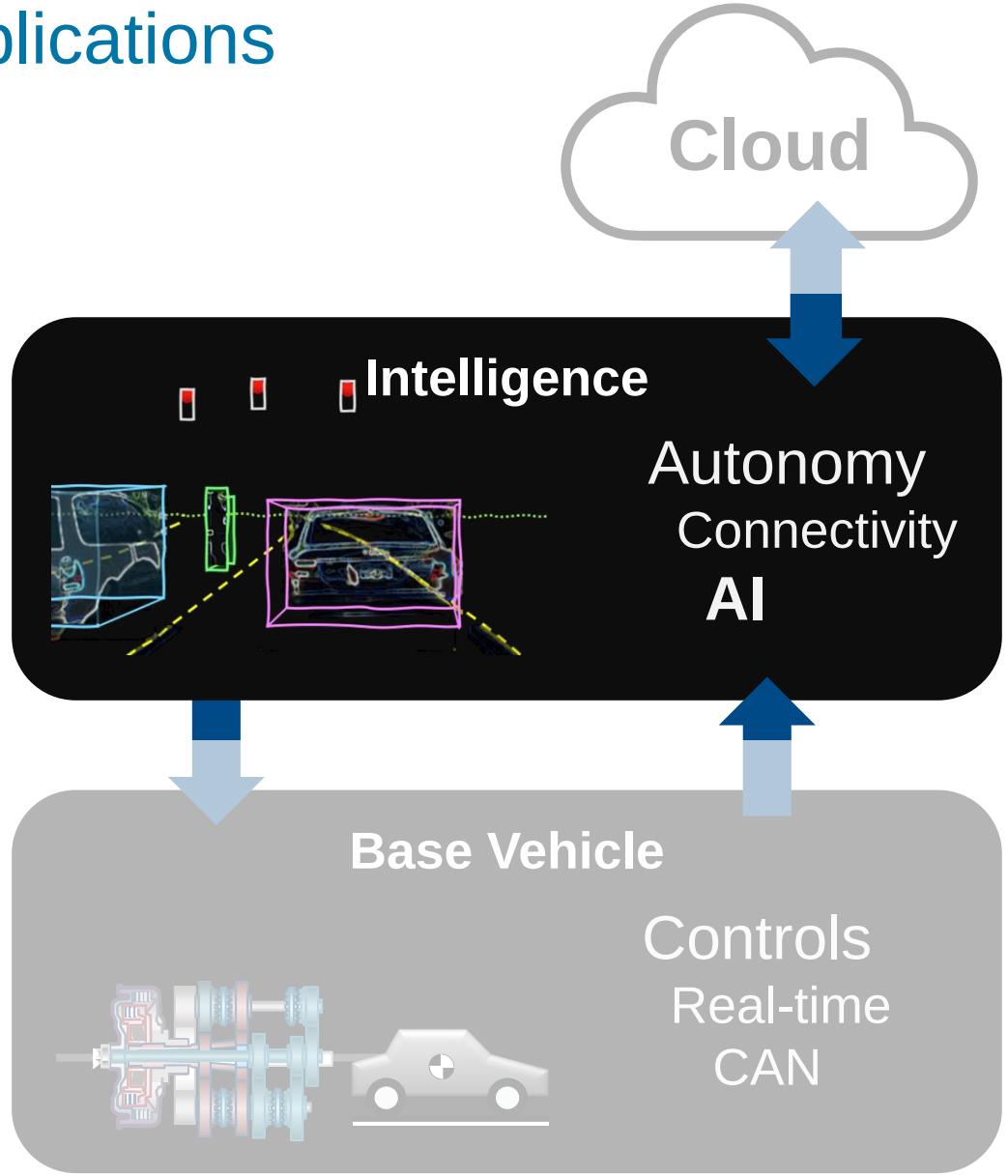
- Business Applications
- Data Handling and Monitoring

- OTA Updates

- Intelligence using Machine Learning and Deep Learning

- Service oriented Software Architecture
- Evolving hardware architectures

Software-Defined Vehicles: Workflows for In-Car and Cloud Applications



- Business Applications
- Data Handling and Monitoring

- OTA Updates

- Intelligence using Machine Learning and Deep Learning

- Service oriented Software Architecture
- Evolving hardware architectures

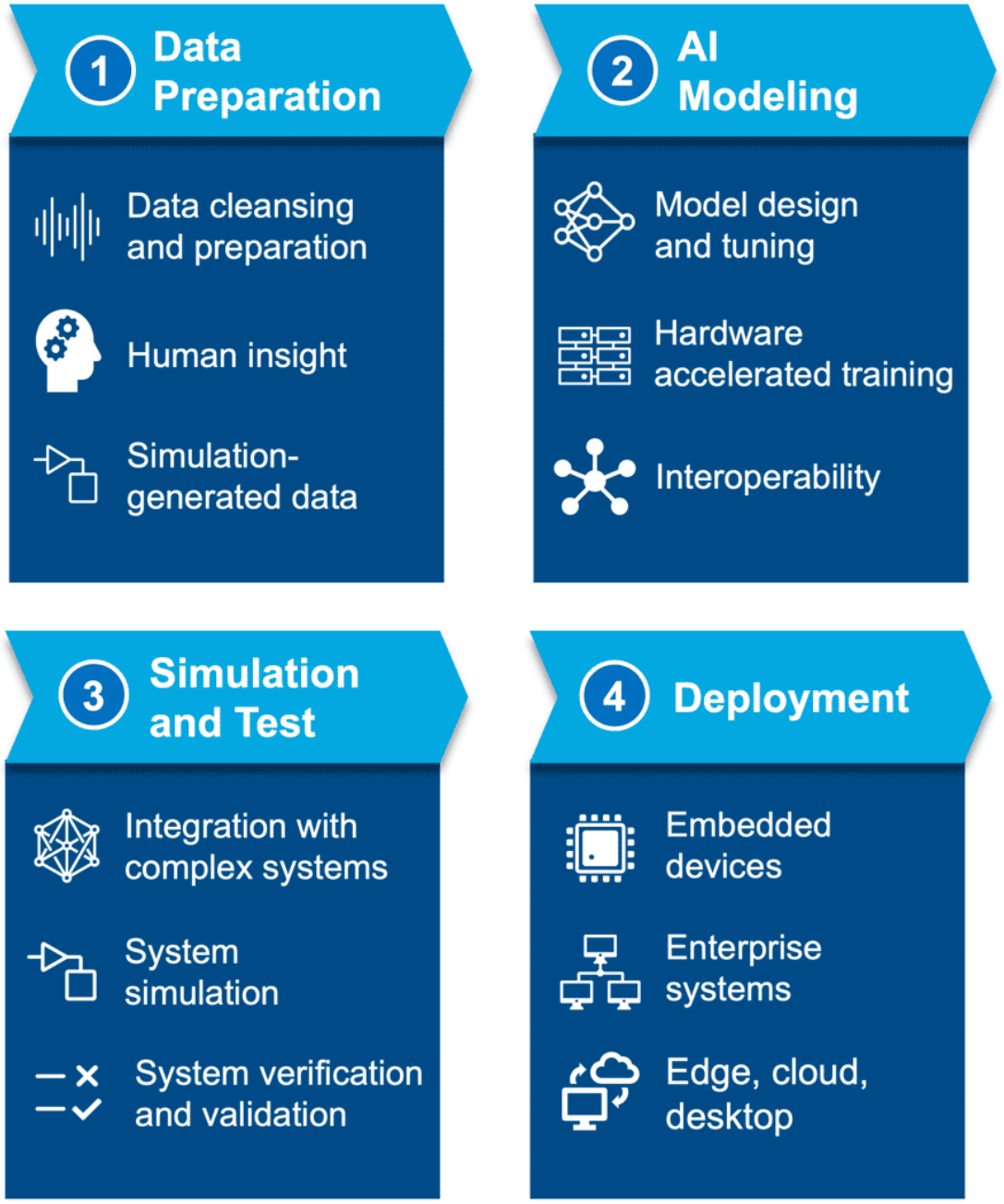
Adding intelligence using Machine Learning and Deep Learning

Software defined vehicles will generate huge amounts of data

Entire workflow for AI

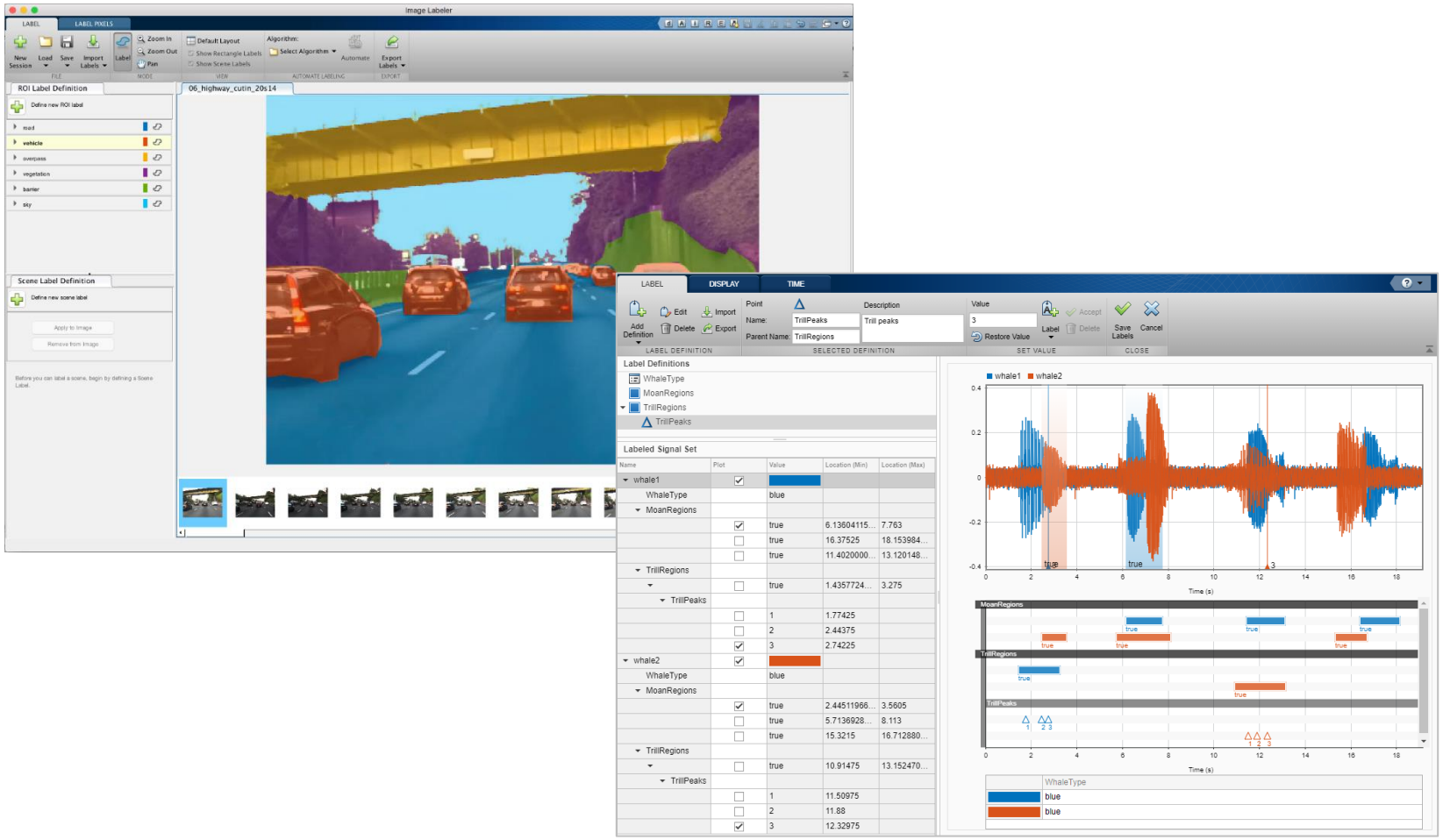
Interoperability with open source

Deployment on various platforms



Automated labeling Apps save you weeks to months

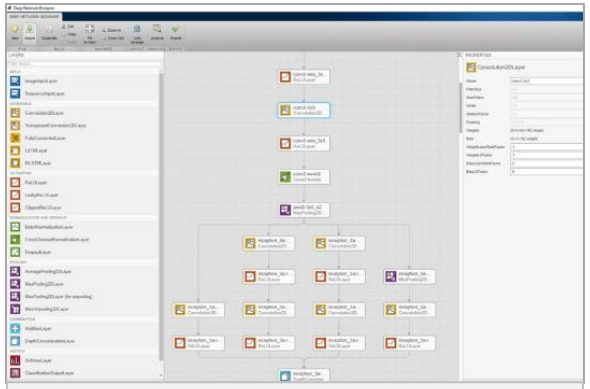
1. Data Preparation
2. AI Modeling
3. Simulation and Test
4. Deployment



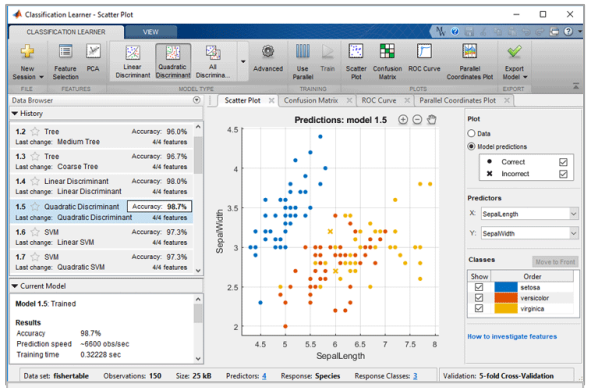
MathWorks

AI modeling Apps automate training, tuning, visualization...

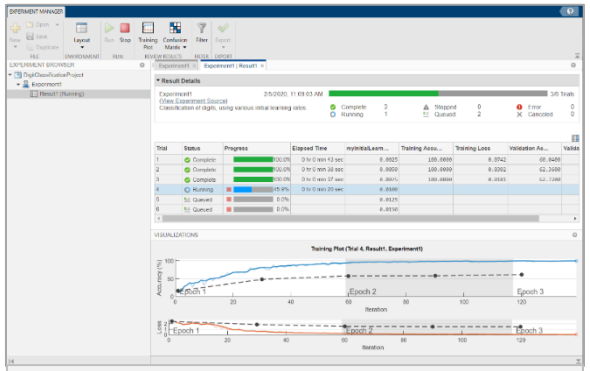
1. Data Preparation
2. AI Modeling
3. Simulation and Test
4. Deployment



Deep Network Designer app to build, visualize, and edit deep learning networks.

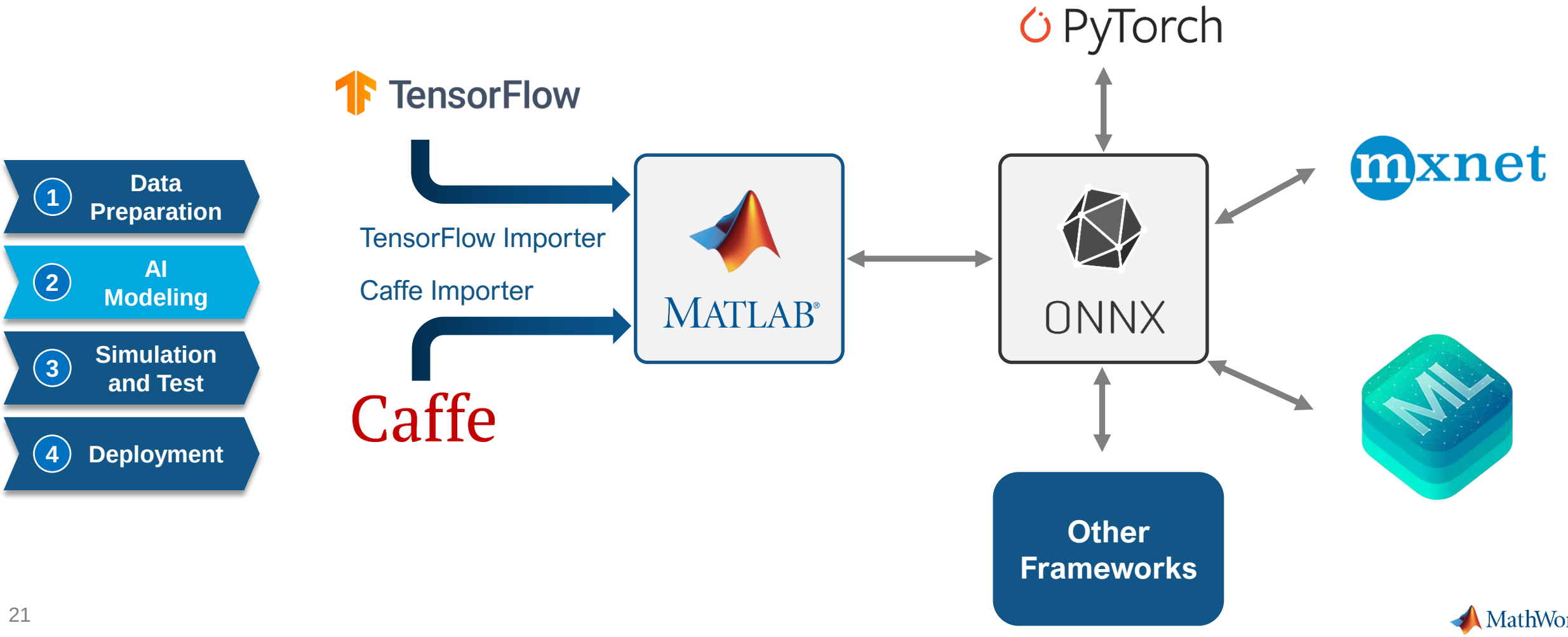


Classification Learner app to try different classifiers and find the best fit for data sets.



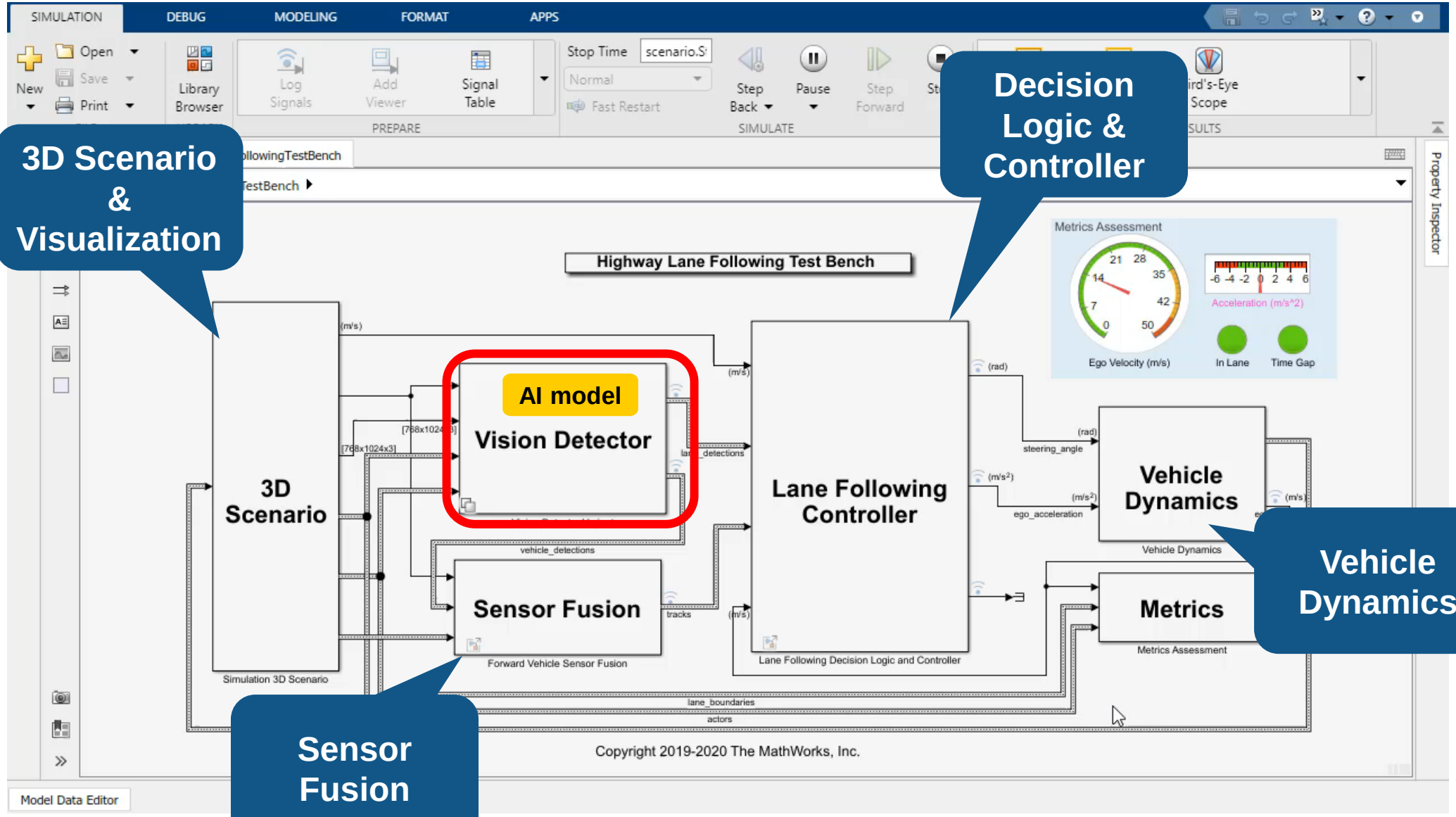
Experiment Manager app to run deep learning experiments to train networks and compare results.

Access AI models from the broader AI community



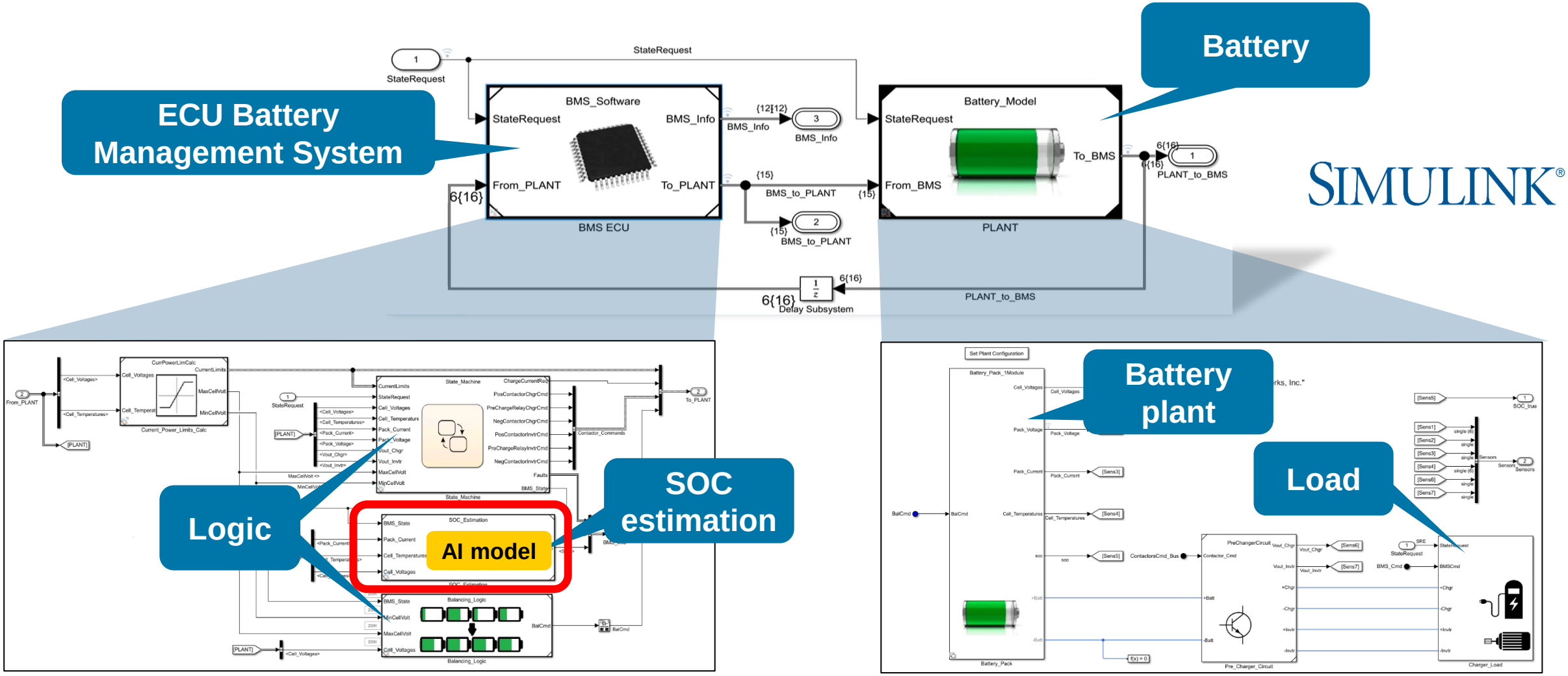
AI for Vision Detection

1. Data Preparation
2. AI Modeling
3. Simulation and Test
4. Deployment



MathWorks

AI for Battery State of Charge (SOC) Estimation

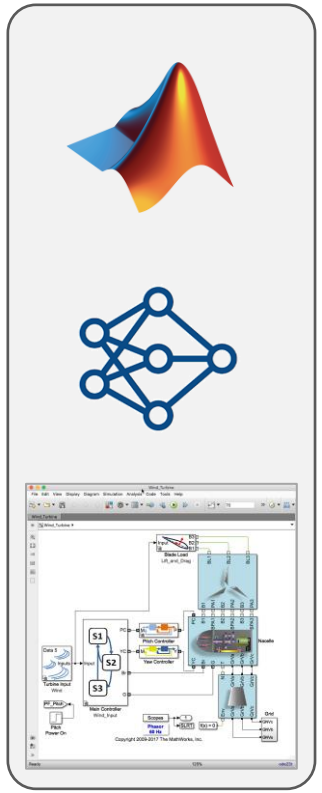


Closed-loop system

23

Deploy to any processor with zero coding errors

1. Data Preparation
2. AI Modeling
3. Simulation and Test
4. Deployment

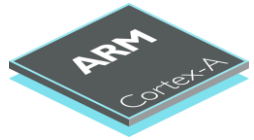


Code Generation

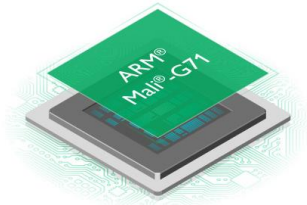
CPU

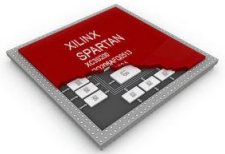
GPU

FPGA



NVIDIA

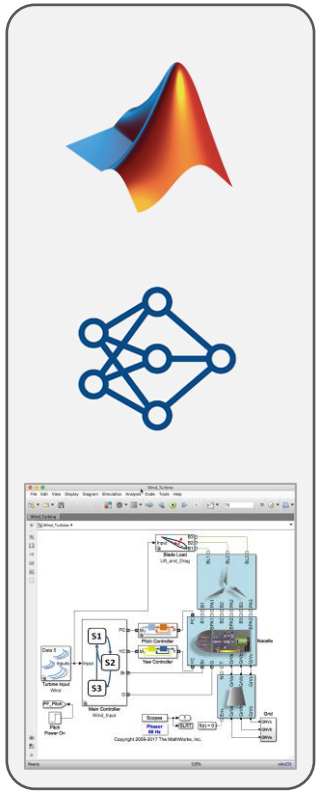




MathWorks

Deploy to any enterprise IT infrastructure

1. Data Preparation
2. AI Modeling
3. Simulation and Test
4. Deployment



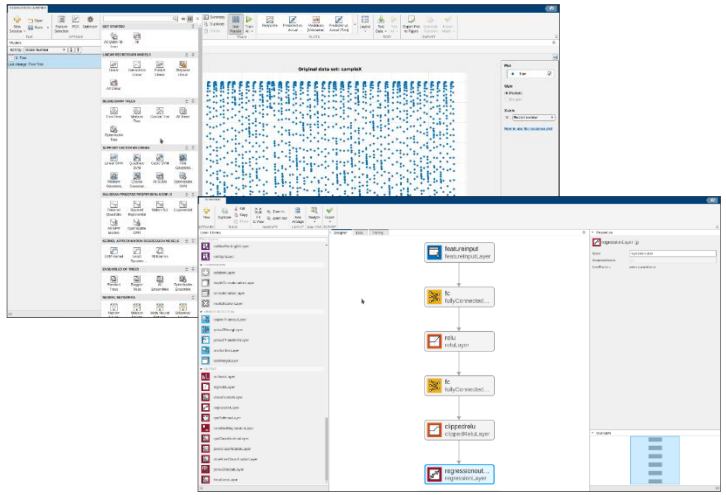
MATLAB Production Server

Request Broker

- Databases
- Streaming
- Cloud Storage
- Dashboards
- Containers
- Custom Tools

Cloud & Datacenter Infrastructure

AI using Machine Learning and Deep Learning



Low-Code Tools to Get Started

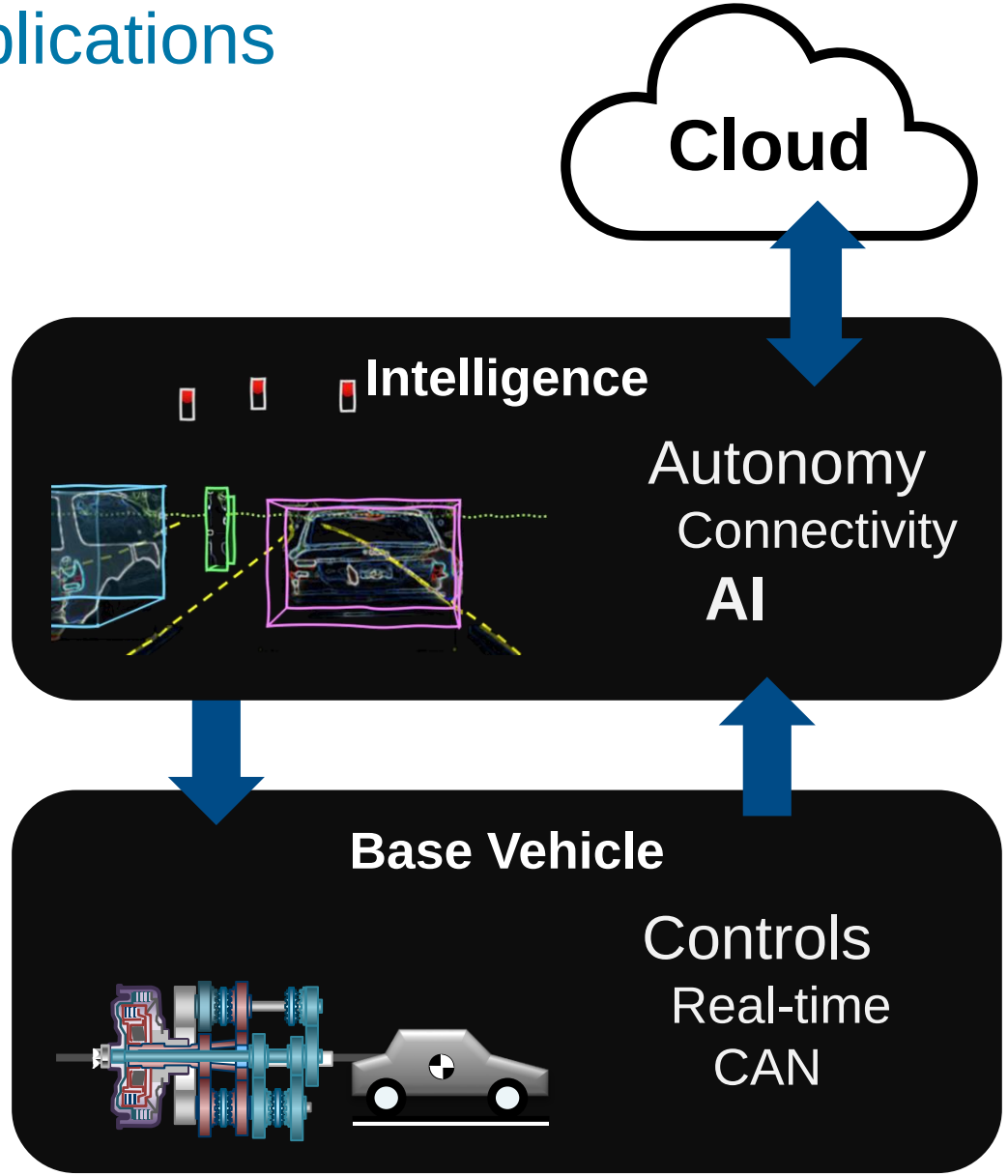
Data Preparation	AI Modeling	Simulation & Test	Deployment
Data cleansing and preparation	Model design and tuning	Integration with complex systems	Embedded devices
Human insight	Hardware accelerated training	System simulation	Enterprise systems
Simulation-generated data	Interoperability	System verification and validation	Edge, cloud, desktop

Complete Workflow from Data to Deployment

MATLAB + TensorFlow PyTorch

Exchange Models with Other Frameworks

Software-Defined Vehicles: Workflows for In-Car and Cloud Applications



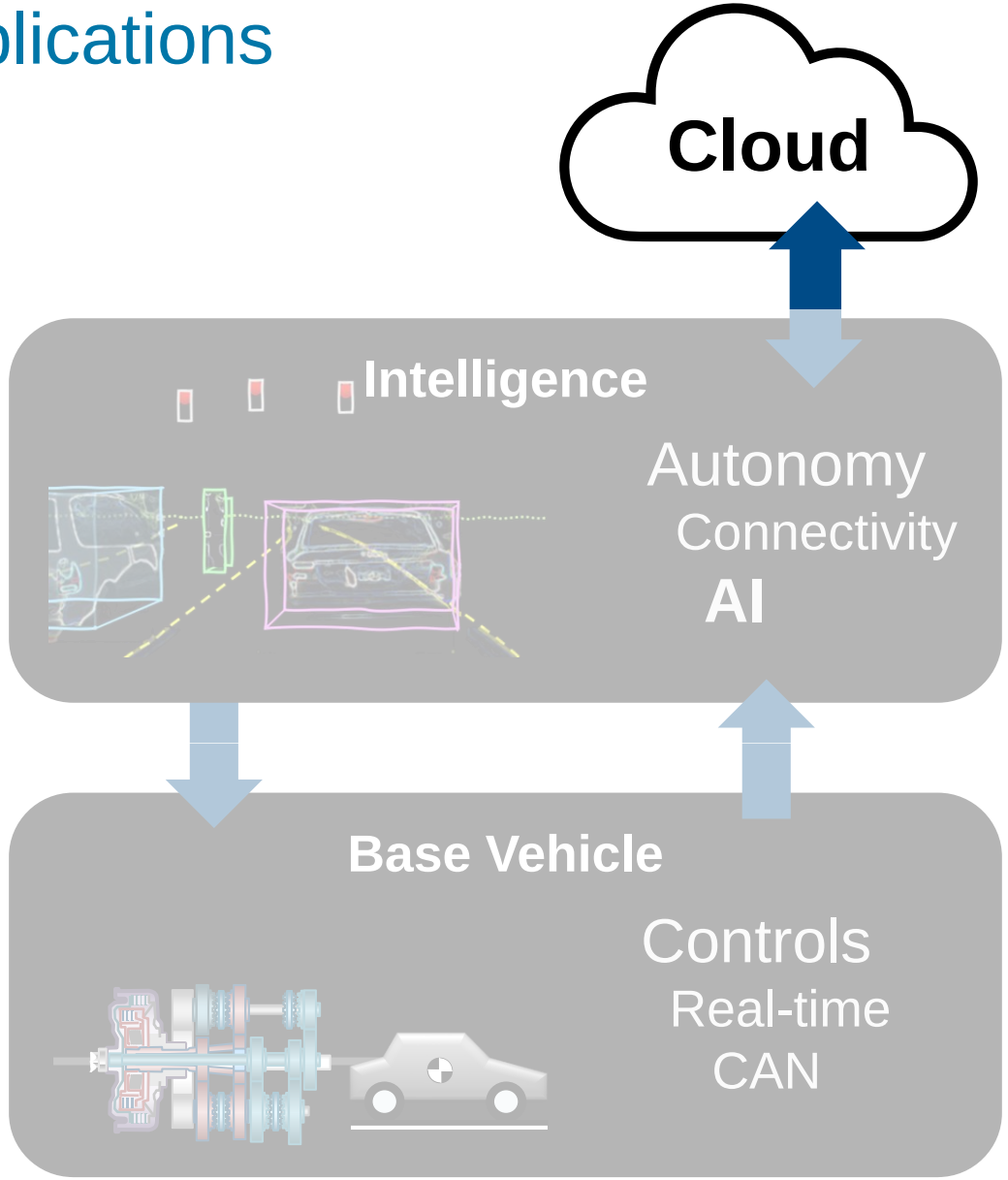
- Business Applications
- Data Handling and Monitoring

- OTA Updates

- Intelligence using Machine Learning and Deep Learning

- Service oriented Software Architecture
- Evolving hardware architectures

Software-Defined Vehicles: Workflows for In-Car and Cloud Applications



- Business Applications
- Data Handling and Monitoring

- **OTA Updates**

- **Intelligence using Machine Learning and Deep Learning**

- Service oriented Software Architecture
- Evolving hardware architectures

Cloud Workflows

Develop
using online tools

Make deployment of software tools simple

Give engineers access to their tools from anywhere

Connect to data & simulate
using elastic resources

Access to more data and compute resources

Make design robust with limitless real-world scenarios

Integrate & test
using cloud-based CI

Use latest tooling and compute resources

Allow whole engineering team to join early integration

Deploy
software and simulation

Extend deployment to a new platform

Create new businesses with live update and digital twin

Cloud Native Connectors

Coding Tools & Services

Jenkins, GitLab, circleci

Data Access

hadoop, Google BigQuery

Coding & Data Science Platforms

jupyter, databricks, DOMINO

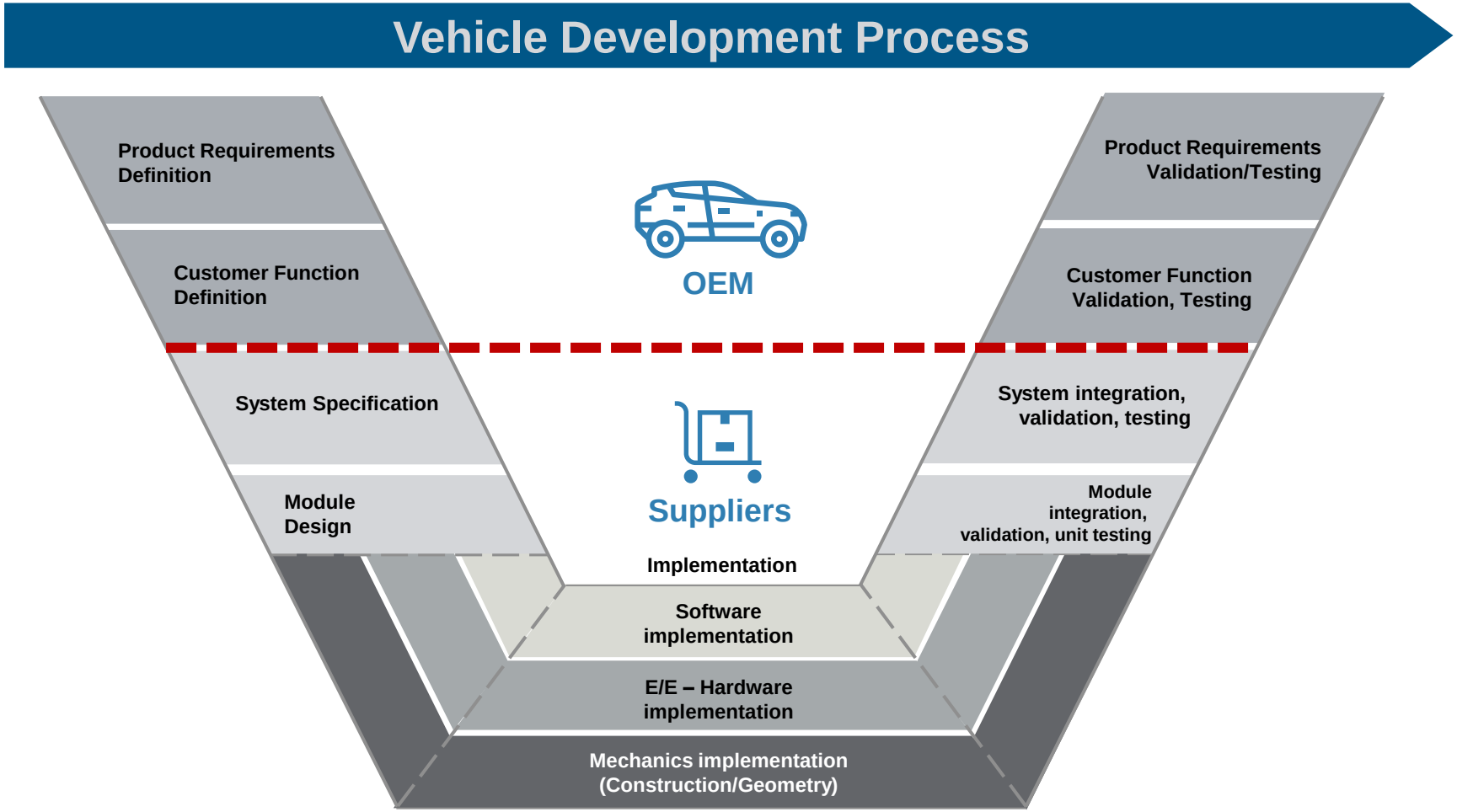
Containers

docker, NVIDIA GPU CLOUD, kubernetes

Public Clouds

Azure, aws

What can we learn from the ‘V-Model’ for the future?



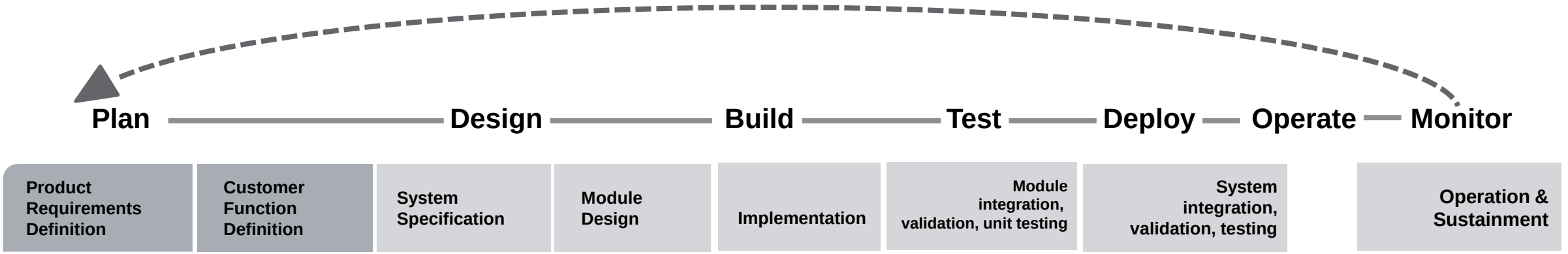
For years the ‘V’ model driven Vehicle Development Process has served the industry well.

This strong V-model oriented approach has resulted in issues for **software development** process.

Some impacts:

1. Missing a milestone meant that the feature will not make it into the series vehicle.
2. Big bang integration was expected to just work
3. There was no room for failure and the cost of failure has been very high (‘zero-defect’ quality principle)
4. Missing understanding of full vehicle system in module teams, which leads to integration issues and calls backs.

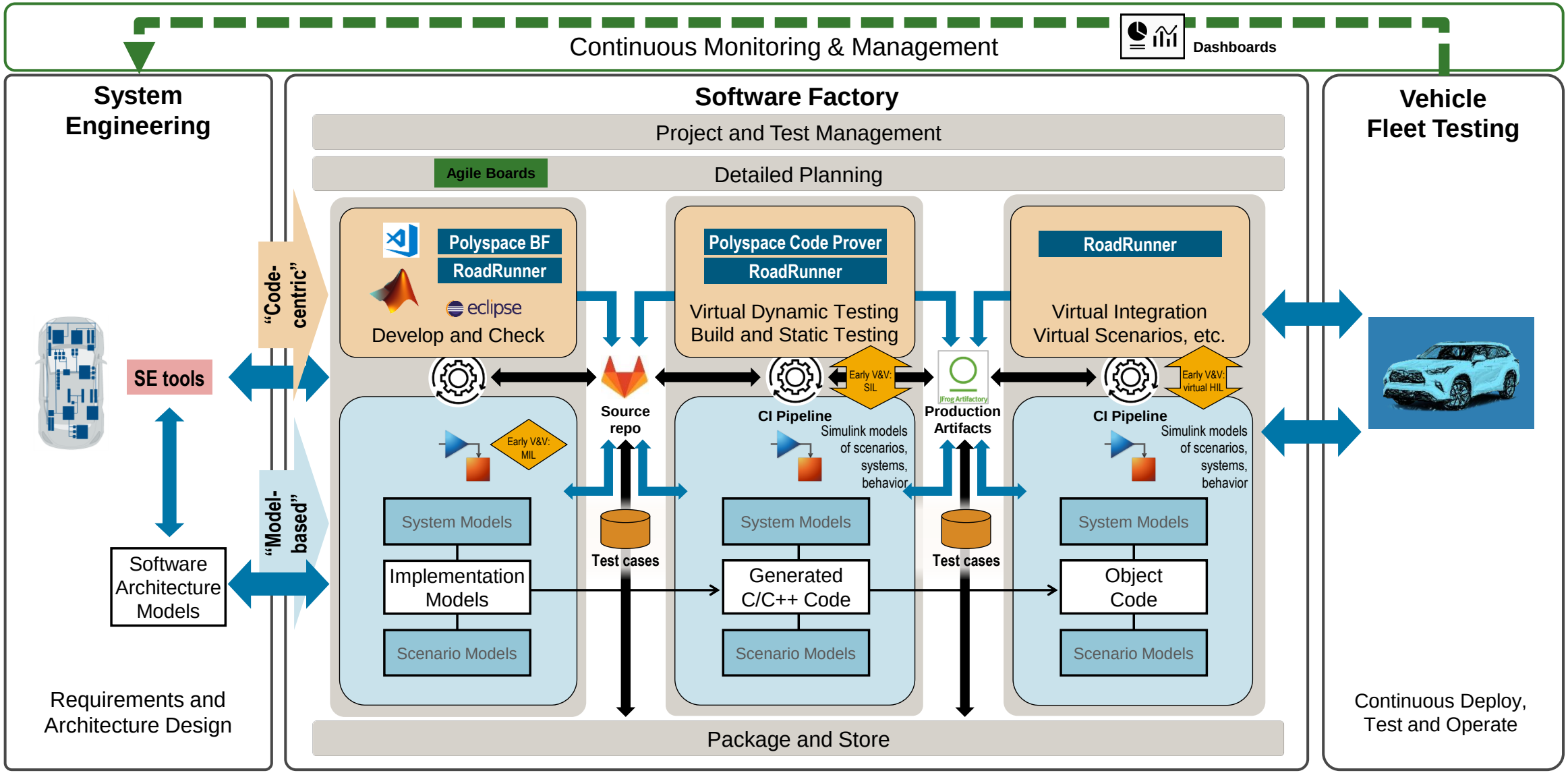
An Automotive View of the DevOps Lifecycle



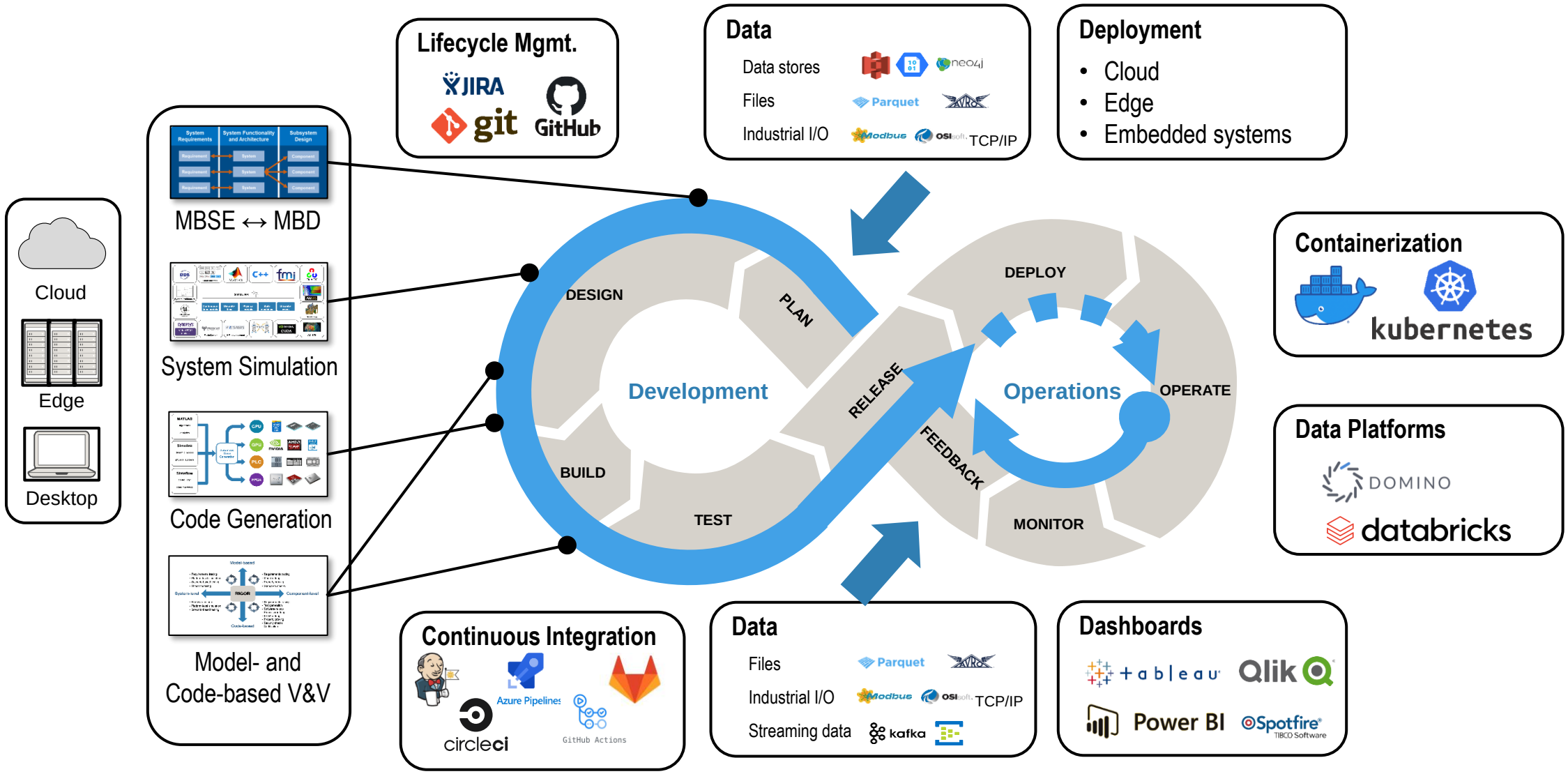
Perpetually Upgradeable Vehicles

- Faster Development and Release Cycles
- Better Leverage Data from System Operation
- Enable New Types of Algorithms and Functionality in the Field

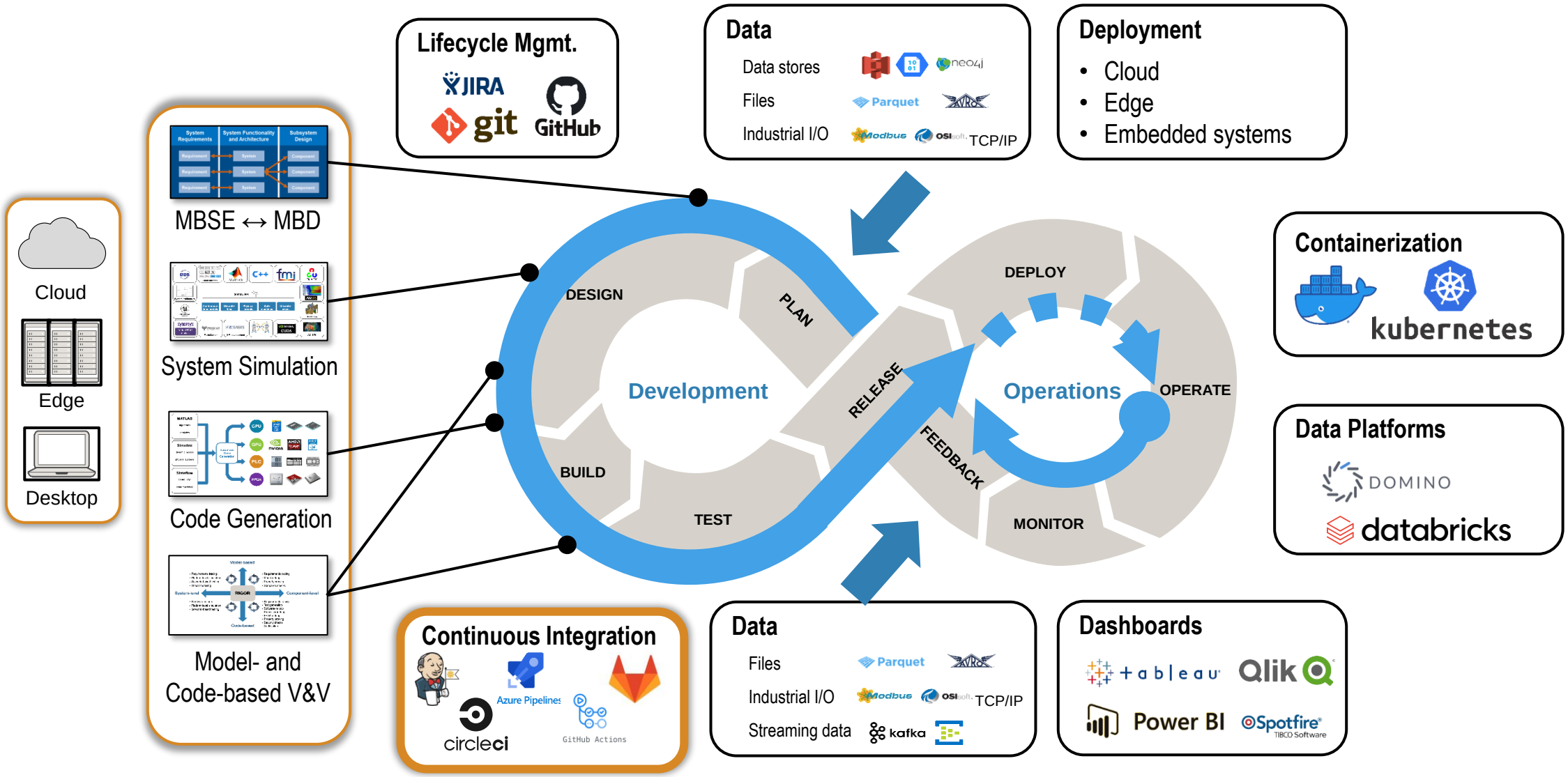
Integrating Model-Based Design in Software Factory



Example for DevOps building blocks for embedded production SW



Continuous Integration for embedded production SW



Continuous Integration Workflow with MATLAB and Simulink

Source Control Server

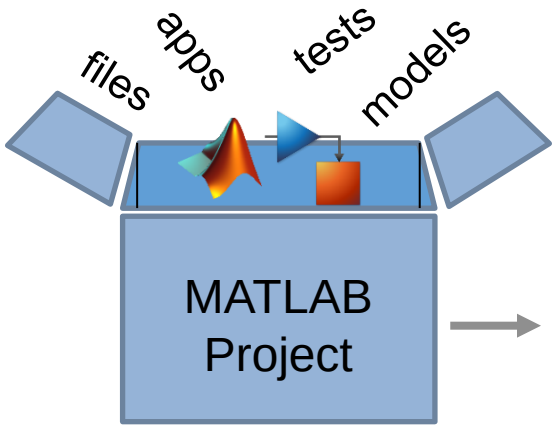
CI Server

Develop | **Test** | **Build** | **Notify and Deploy**

- *Run tests:*
  - ✓ *MATLAB Unit Tests*
  - ✓ *Simulink Test*

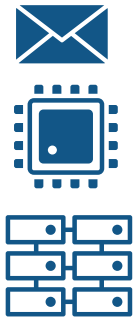
- *Compile MEX*
- *Generate Code*
- *Package (Toolboxes, Apps)*

- *Publish reports*
- *Email Notification*
- *Publish to Server*
- *Hardware*



Tests pass?

Java
MATLAB
C/C++
Python
.NET
Excel
dll
COM



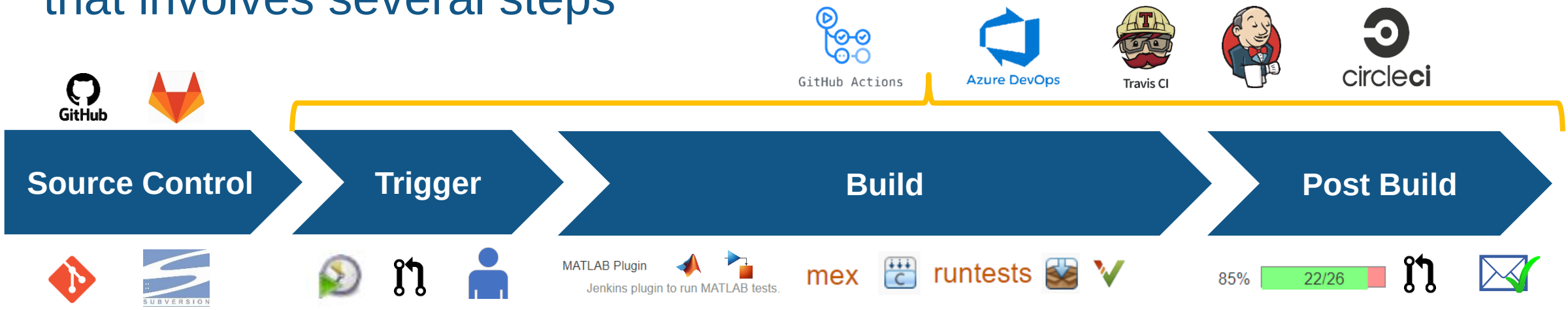
Commit and push changes to Git

GitLab triggers Jenkins

Jenkins run tests

Build, generate code and package

Continuous Integration (CI) is a software development practice that involves several steps



- Git
- GitHub
- Subversion
- GitLab

⋮

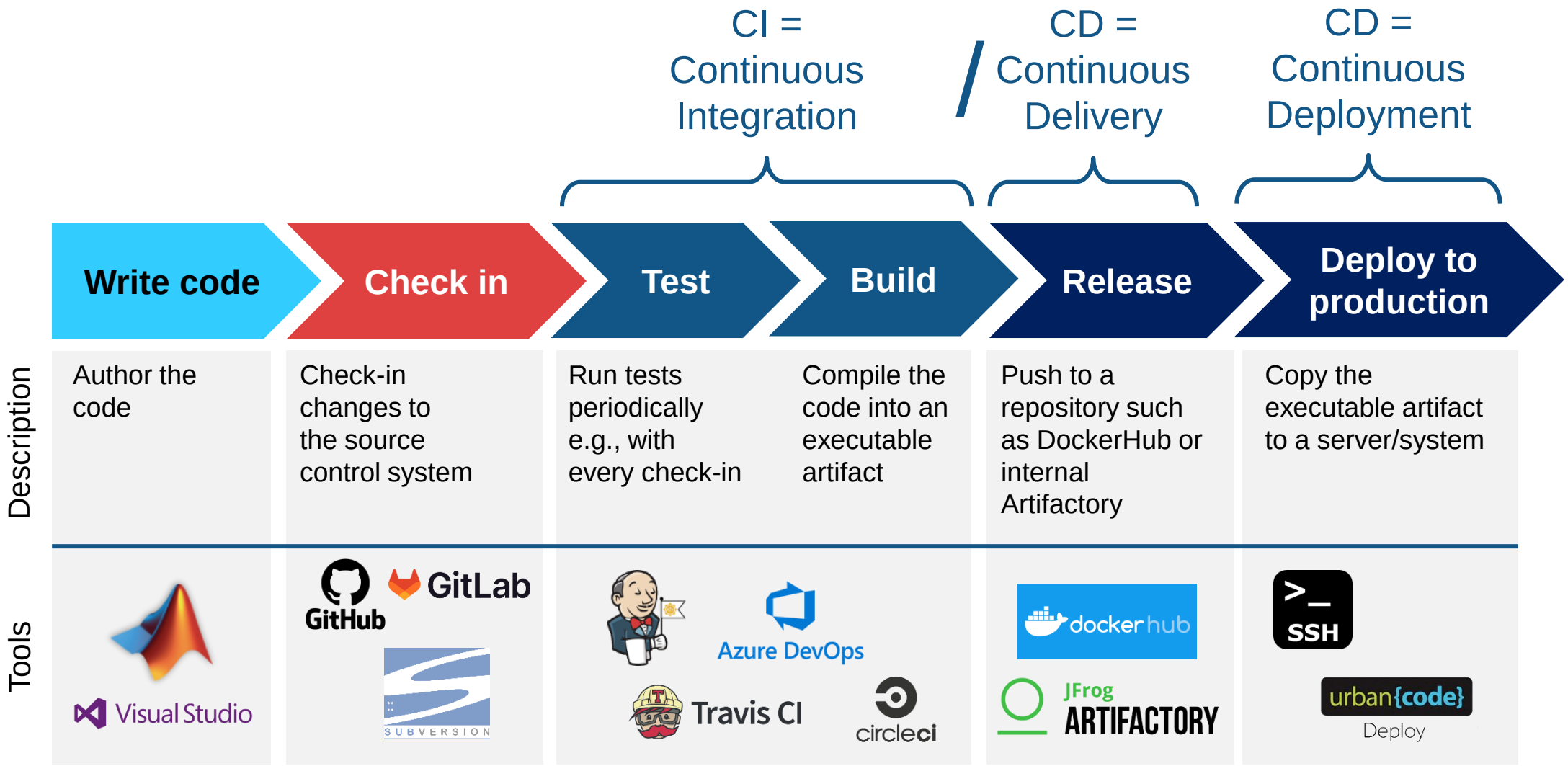
- Push
- Merge Request
- Pull Request
- Check In
- Periodic
- Manual

- Run MATLAB / Simulink Tests
- Run Performance Tests
- Compile MEX Files
- Generate Code†
- Package Toolboxes
- Build Components with MATLAB Compiler Stack†
- Integrate with other Software Technologies

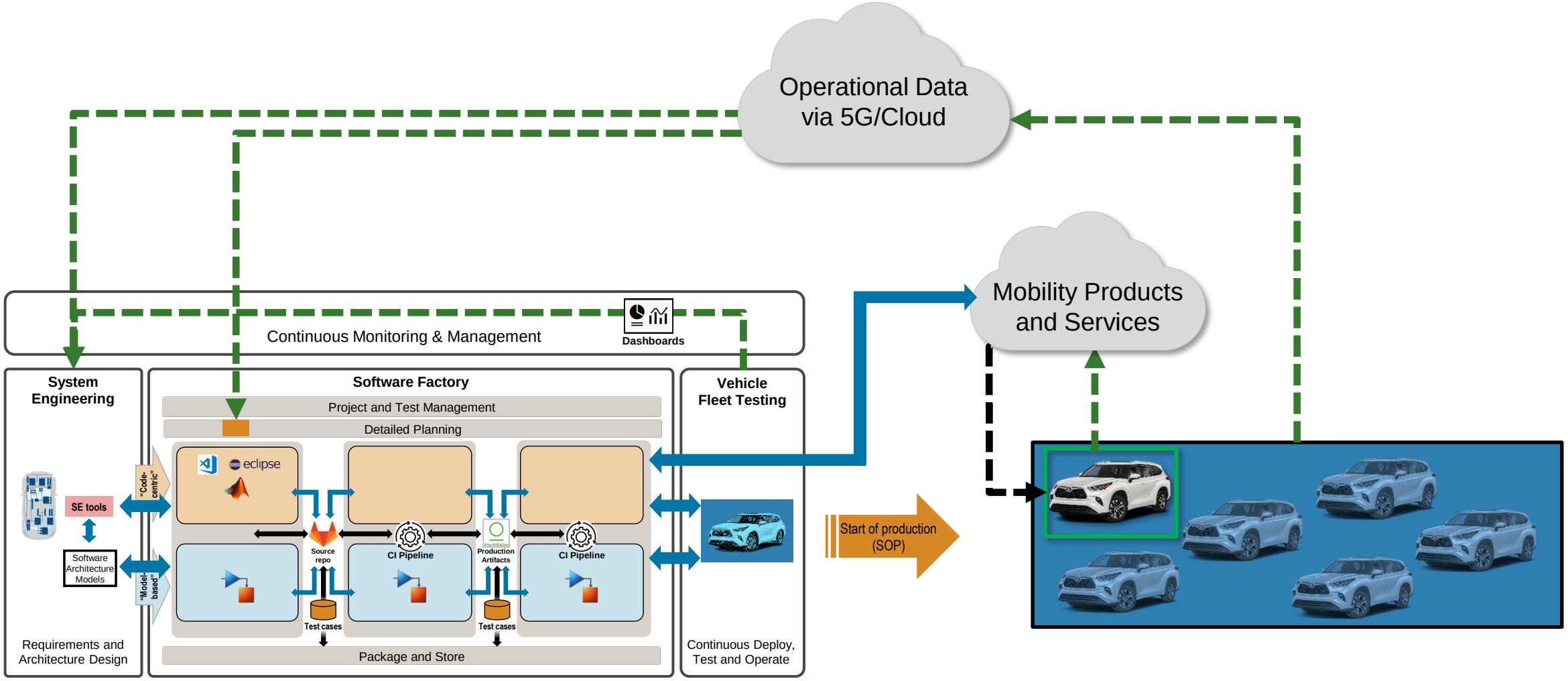
- Publish:
  - Test Results
  - Coverage Results
  - Performance Results
- Accept Merge Request
- Email Notification
- Build Another Project

† Transformation products (e.g., MathWorks Coder and Compiler products) may require Client Access Licensing (CAL)

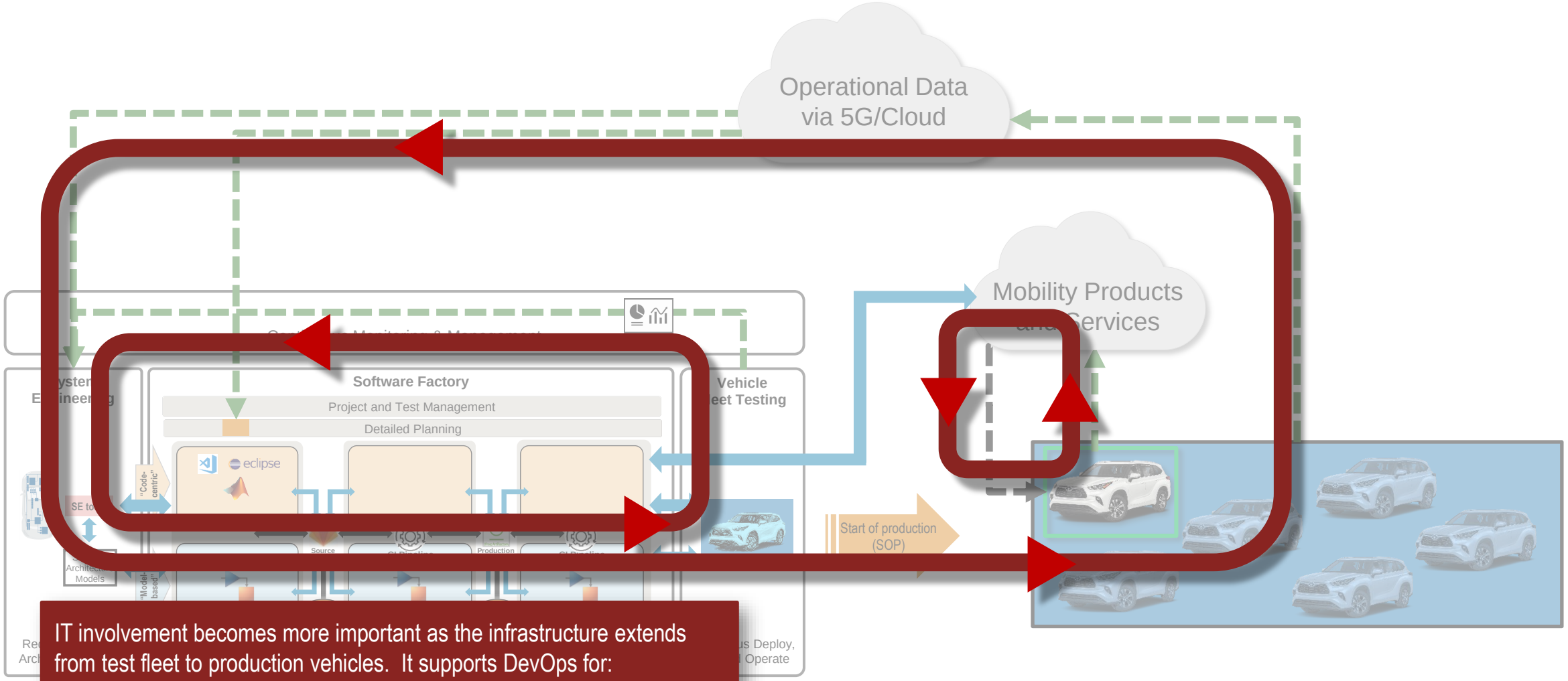
Continuous Delivery and Deployment (CD)



These changes are steps to a broader DevOps view



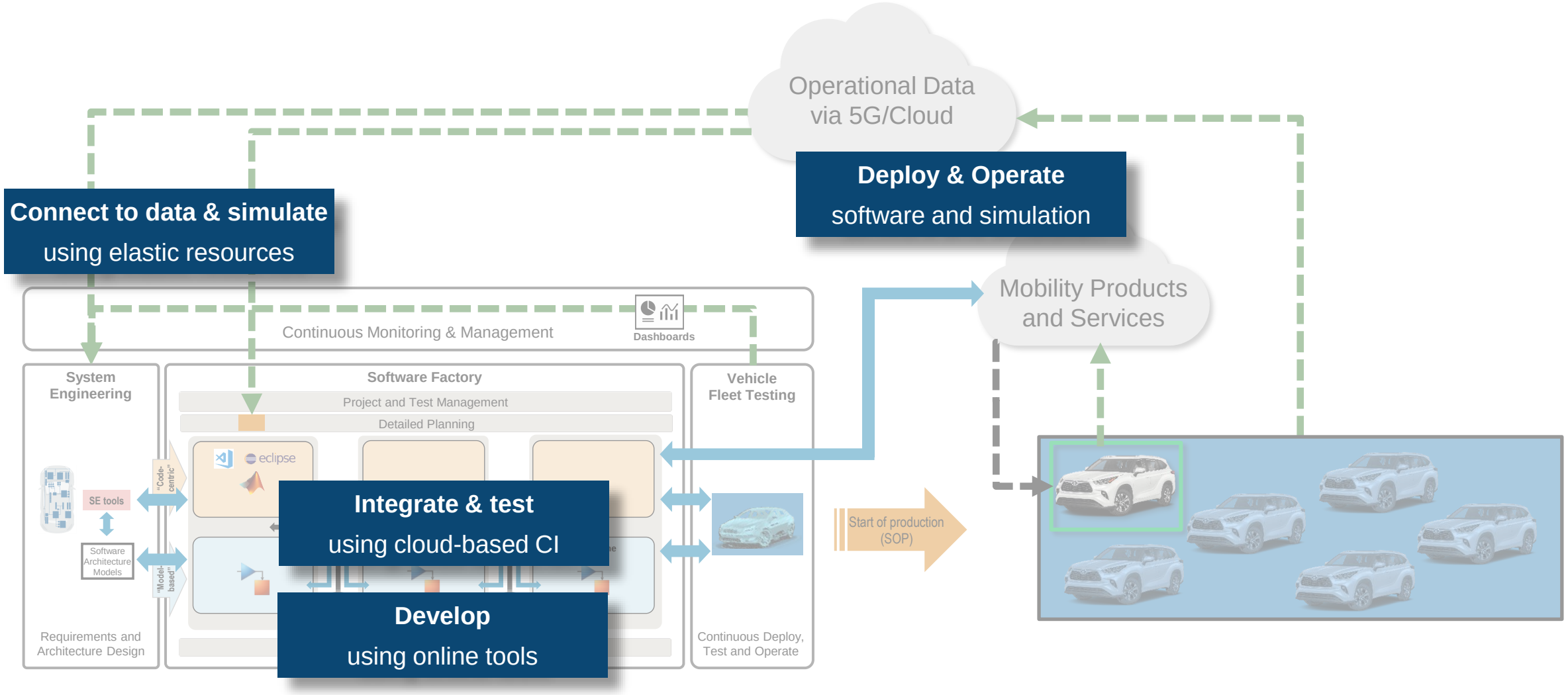
These changes are steps to a broader DevOps view



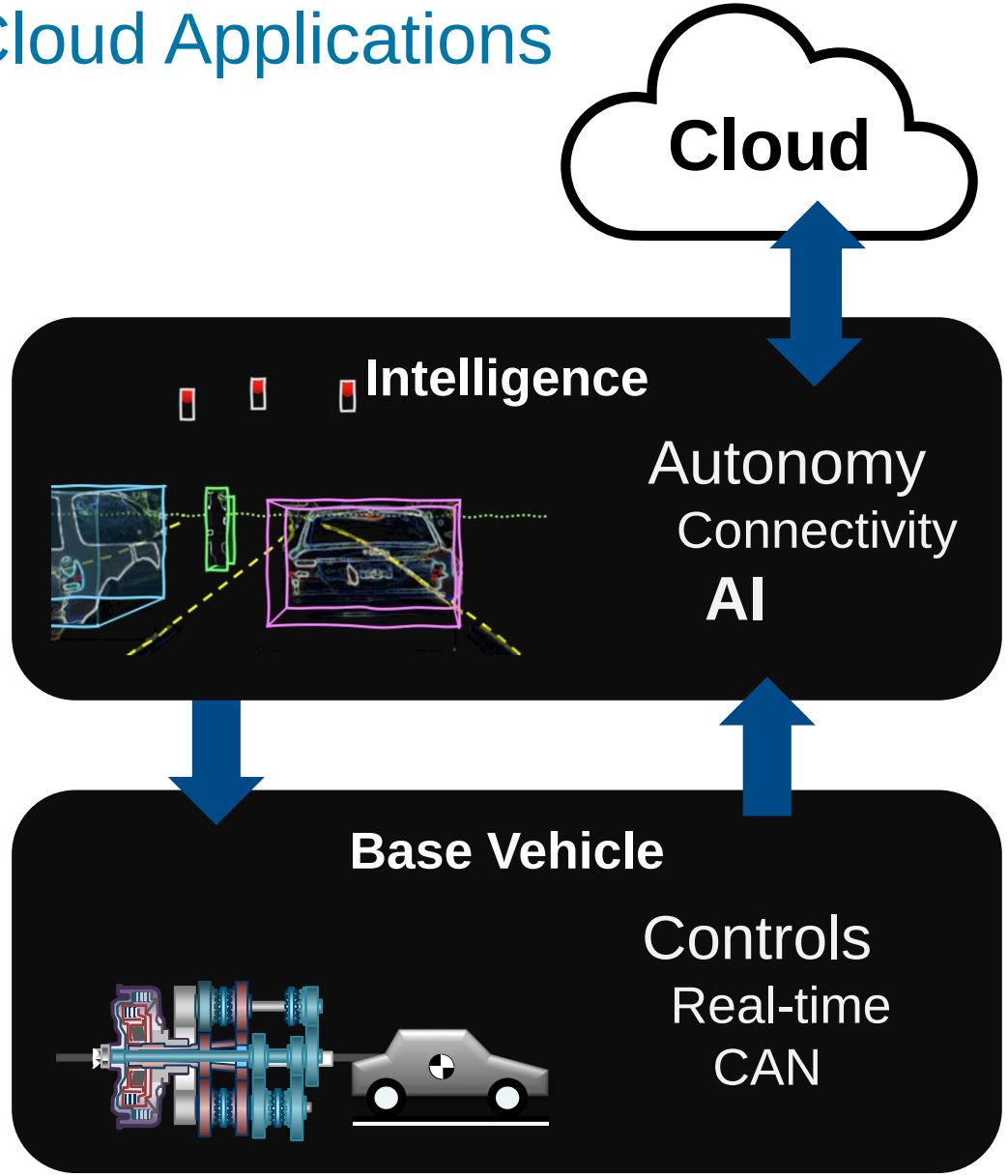
IT involvement becomes more important as the infrastructure extends from test fleet to production vehicles. It supports DevOps for:

1. more effective fleet testing
2. system development and field updates, and
3. support for value-added offerings.

Cloud use case enables a broader DevOps ecosystem



Summary: Software-Defined Vehicles: Workflows for In-Car and Cloud Applications



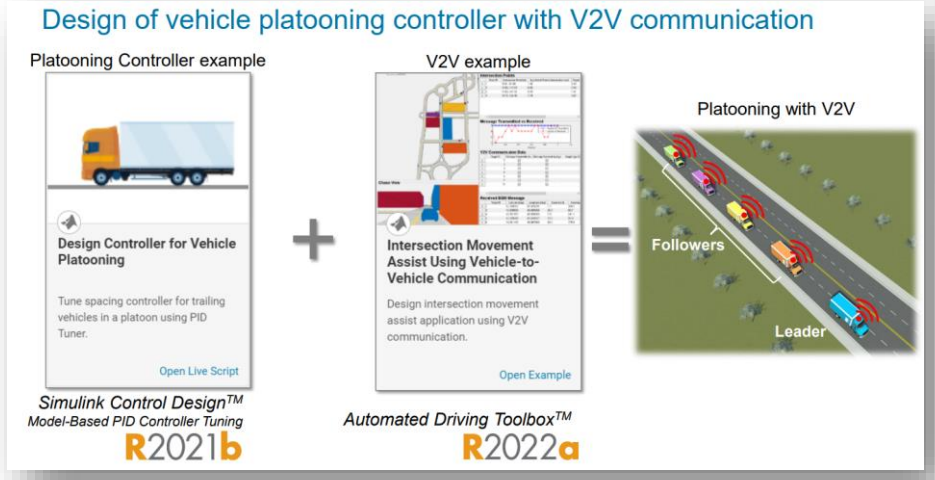
- Business Applications
- Data Handling and Monitoring

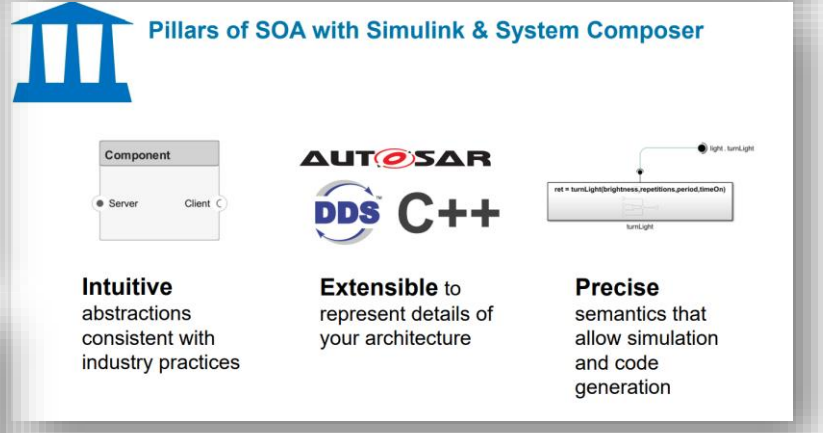
- OTA Updates

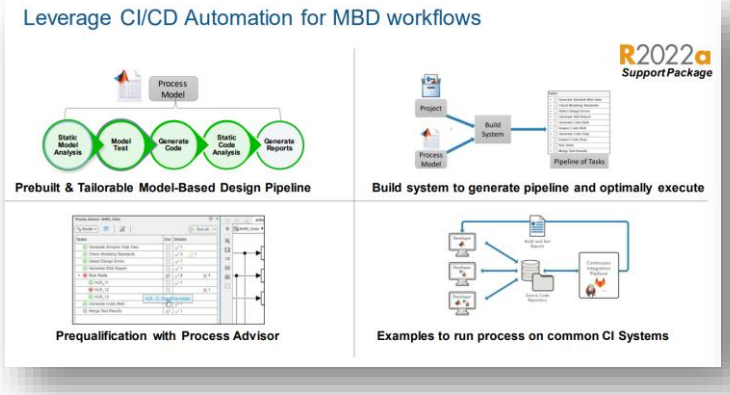
- Intelligence using Machine Learning and Deep Learning

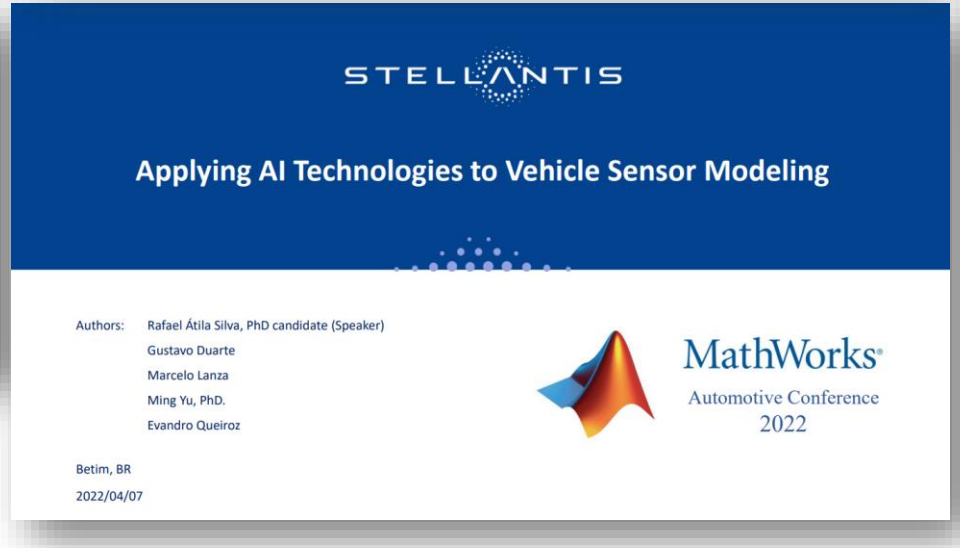
- Service oriented Software Architecture
- Evolving hardware architectures

More Resources: SDV Vision of leading automotive companies









Call to Action

- Visit us at our demo booth, outside the seminar hall
- Various leading customers have presented their vision (on the previous slide).. Watch the resources and let us know your thoughts
- MathWorks would be happy to collaborate with you for developing your SDV solutions