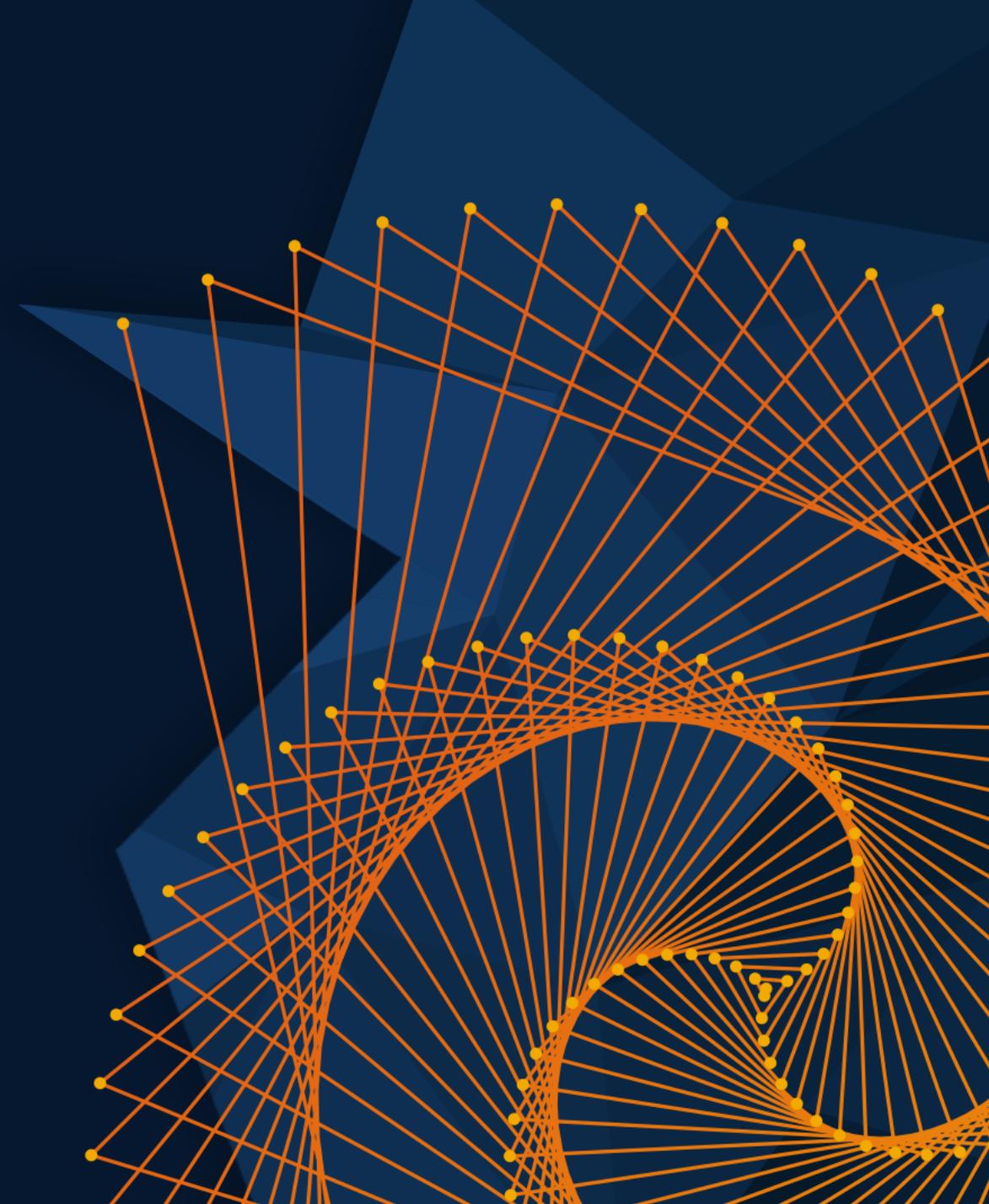


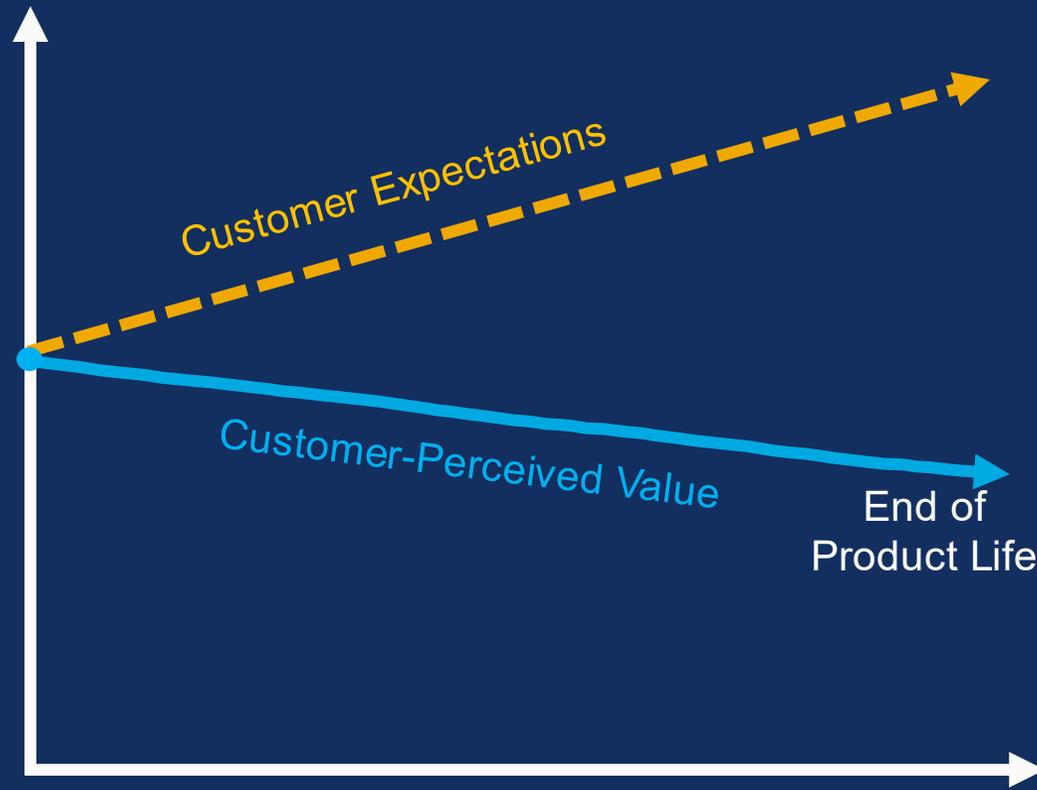
From Embedded to Empowered: The Rise of AI in Software-Defined Products

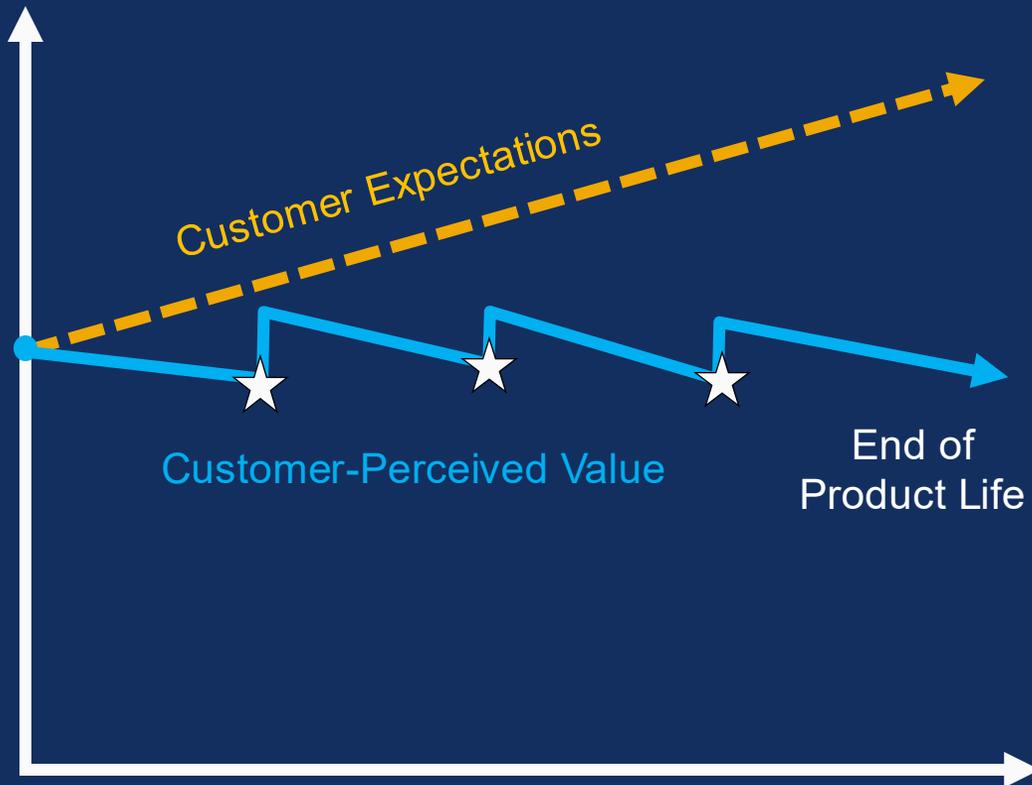
Tianyi Zhu

Senior Product Manager, Simulink & AI









Apps

AI

OS





Software-Defined Products



Software-Defined **Systems**



Software-Defined Products



Software-Defined **Systems**



Reliability



Functional Safety



Physical Components

Software-Defined Products



AI + Software-Defined Systems



Intelligence



Reliability



Functional Safety



Physical Components

Software-Defined Vehicle

Brand-Distinctive Features and Main Customer Value Will Come From Software

Customer expectations

- Sustainable and safe mobility
- Digital life continuity



Technology and innovation

- Artificial Intelligence (AI)
- Electrification
- Autonomy
- Connectivity

Software-Defined Vehicle

Brand-Distinctive Features and Main Customer Value Will Come From Software

Customer expectations

- Sustainable and safe mobility
- Digital life continuity

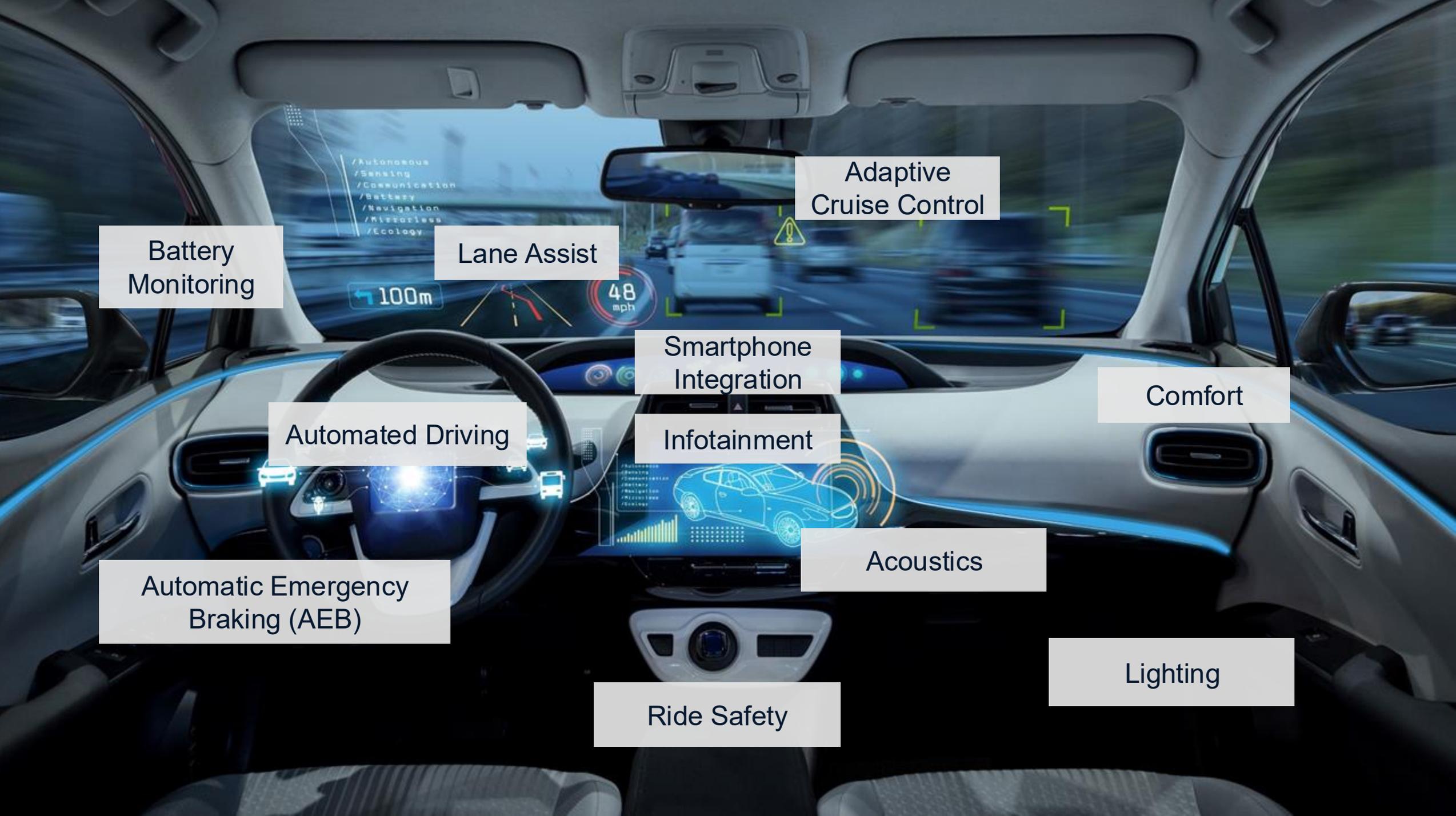


Technology and innovation

- Artificial Intelligence (AI)
- Electrification
- Autonomy
- Connectivity

Business opportunity

- App stores, software features on demand
- Subscription plans for software services



Battery Monitoring

Lane Assist

Adaptive Cruise Control

Automated Driving

Smartphone Integration

Comfort

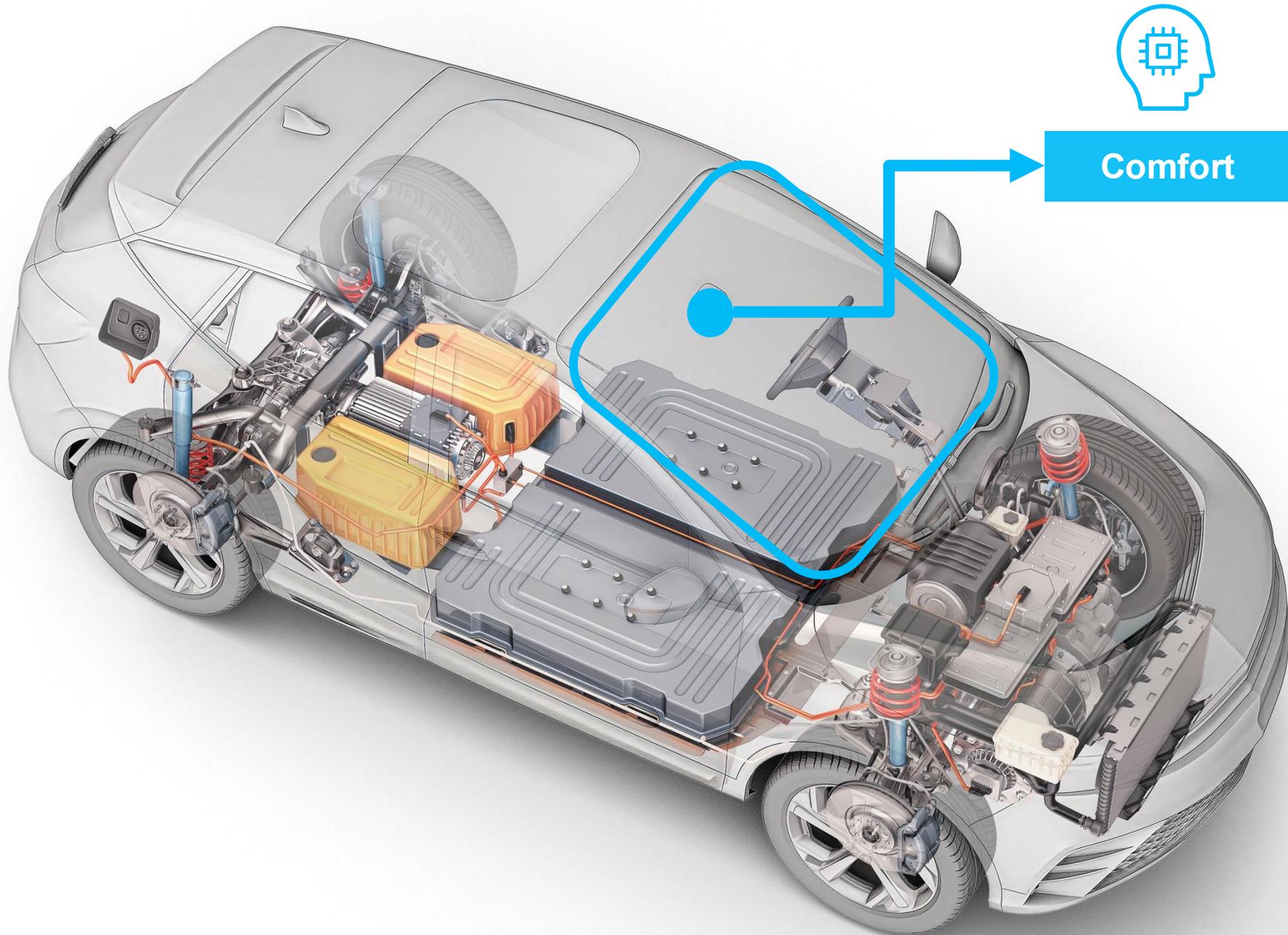
Infotainment

Automatic Emergency Braking (AEB)

Acoustics

Ride Safety

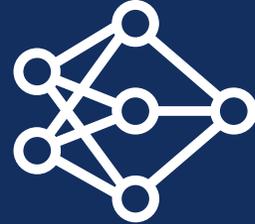
Lighting



Comfort



Mercedes Benz Enhance Cabin Comfort with AI-Powered Air Mass Flow Estimation



Blower speed
Climate control flaps
Stepper positions

...



Mercedes Benz Enhance Cabin Comfort with AI-Powered Air Mass Flow Estimation



Data Preparation

Data visualization & pre-processing

MATLAB

AI Modeling

Low-code AI modeling & training

Statistics & Machine Learning Toolbox

Deep Learning Toolbox

Compression

Model size reduction with pruning & quantization

Simulation & Test

System integration & testing

Simulink

Deployment

Fixed-point conversion & code generation for embedded HW

Fixed-Point Designer

Embedded Coder

Mercedes Benz

Enhance Cabin Comfort with AI-Powered Air Mass Flow Estimation

Prediction Accuracy

Increased by **50%**

Memory & Performance

Requirements **met**

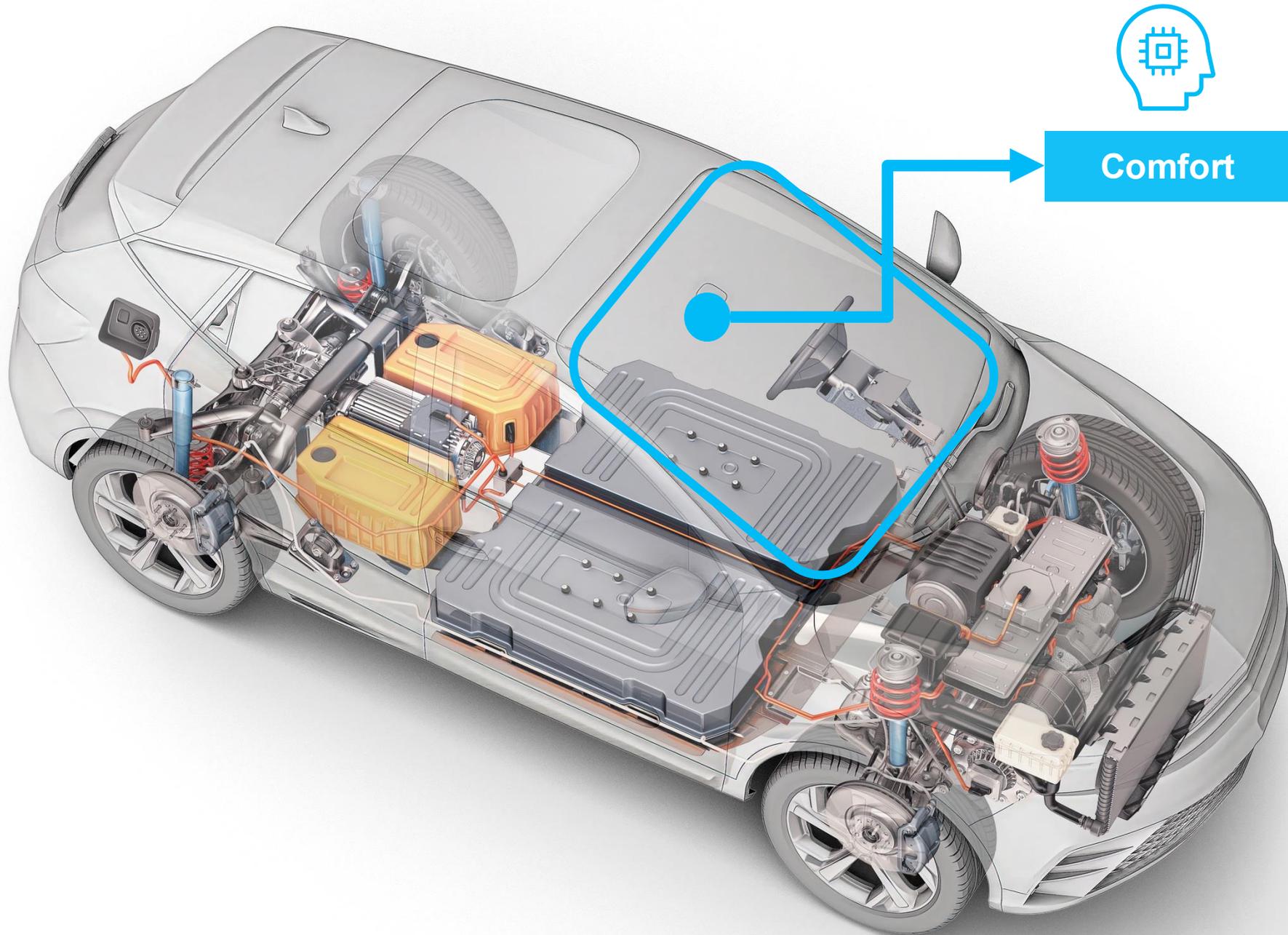
Development Time

Significantly **reduced**



*“[We] started off something from the concept phase to final target deployment, [and] the overall duration is **significantly reduced** with the tool capabilities.*

- Athulya Thazha, Technical Manager, Mercedes-Benz

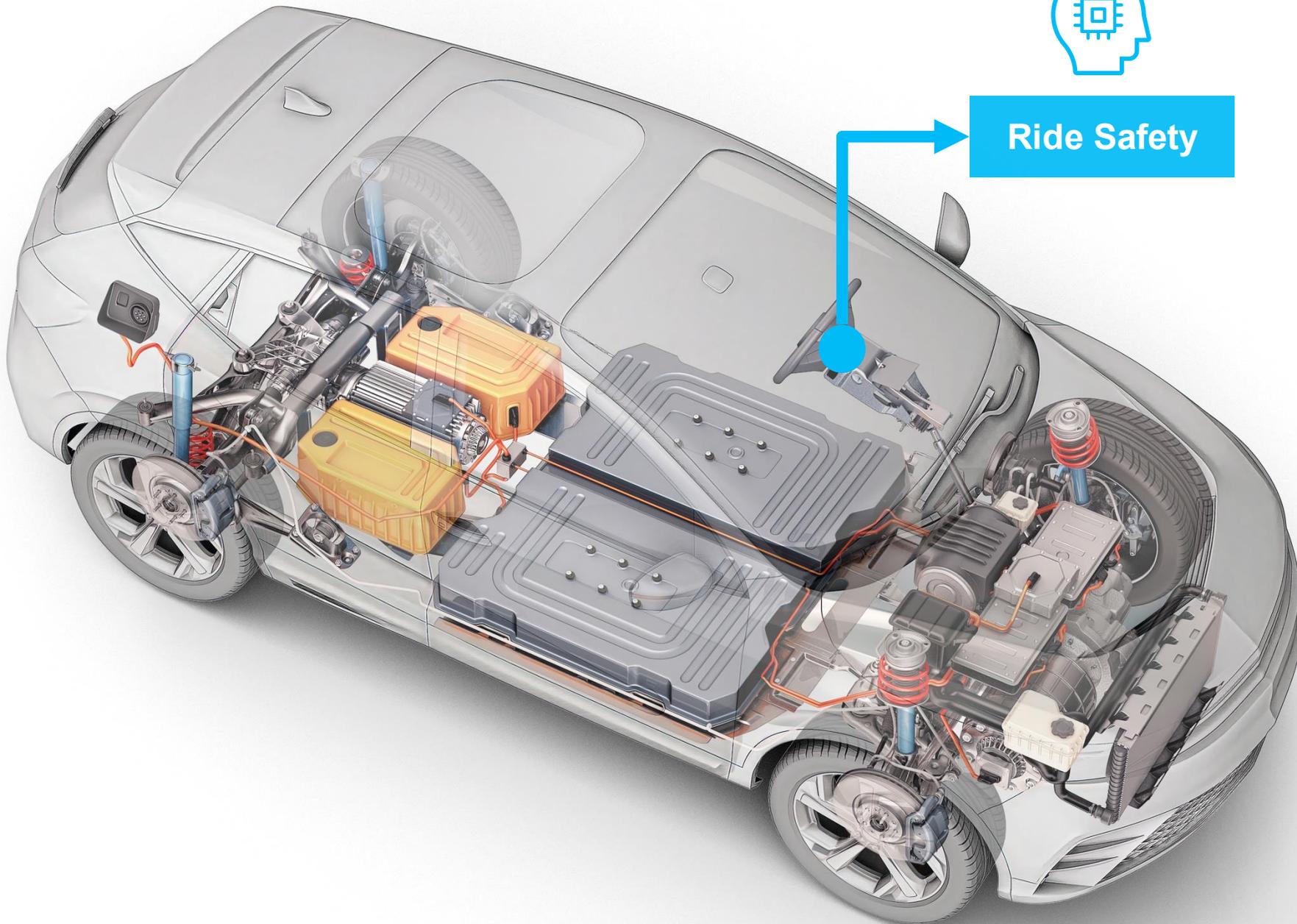


Comfort

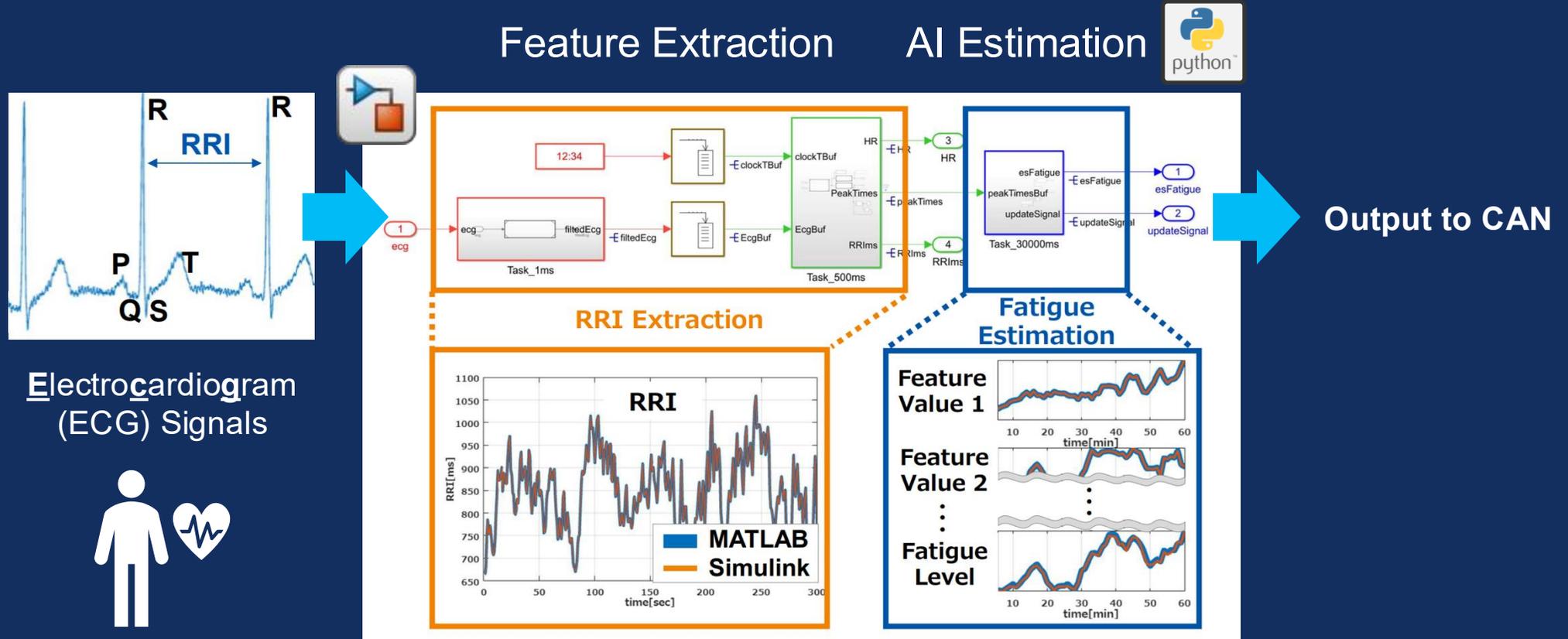




Ride Safety



Enhance Ride Safety with AI-Powered Driver Fatigue Detection



AI Resources - from specific **domains** to broader **communities**

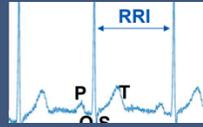
AI reference examples



Predictive Maintenance



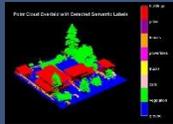
Hyperspectral Imaging



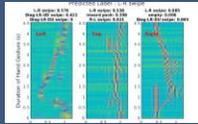
Signal Processing



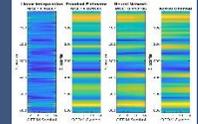
Robotic Control



Lidar Processing



Radar Processing



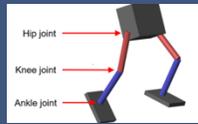
Wireless Communications



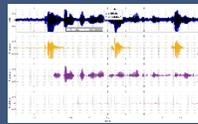
Automated Driving



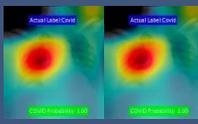
Visual Inspection



Reinforcement Learning



Audio



Medical Imaging

Interoperability with AI ecosystem

TensorFlow

TensorFlow
Importer

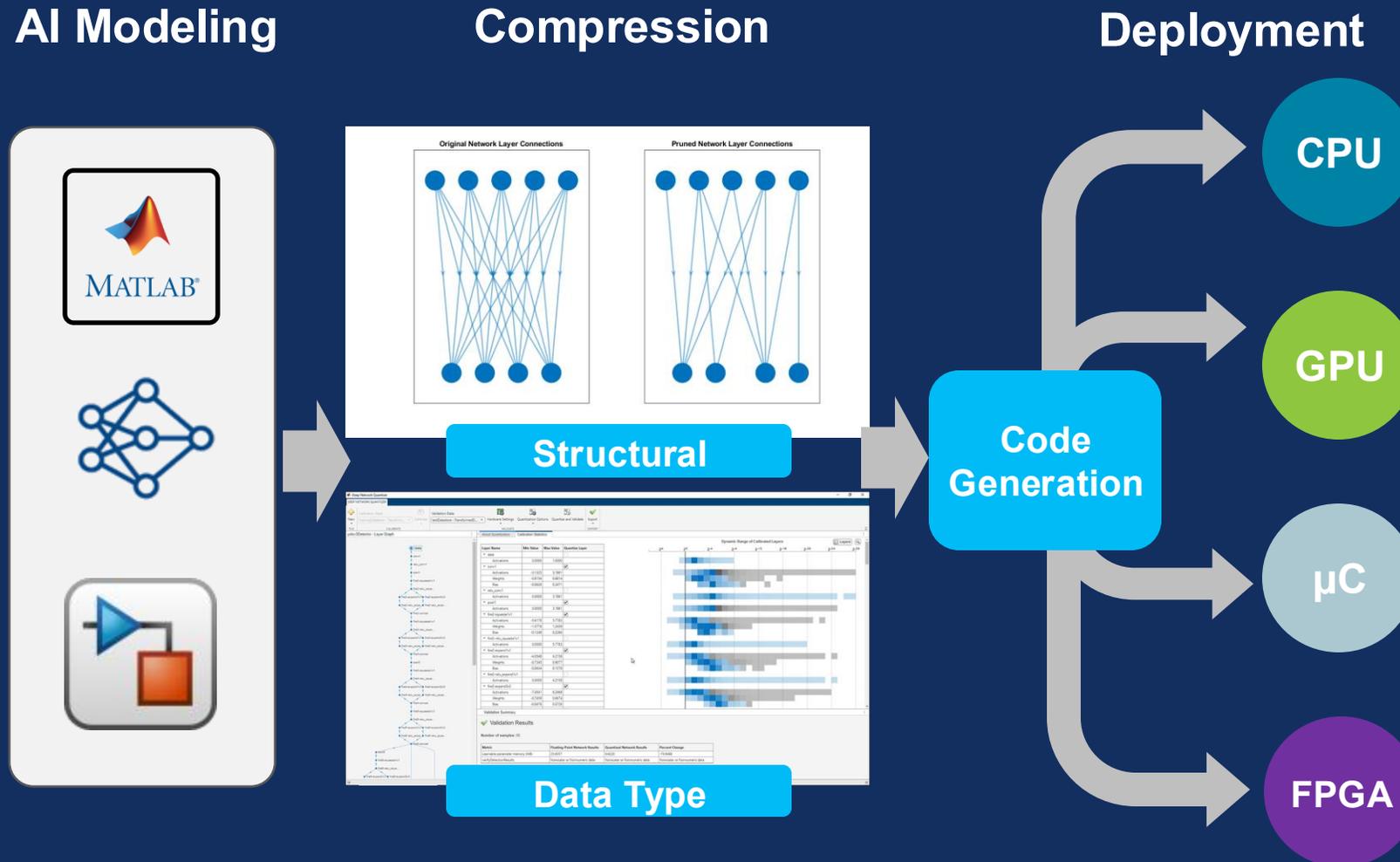


PyTorch

PyTorch
Importer

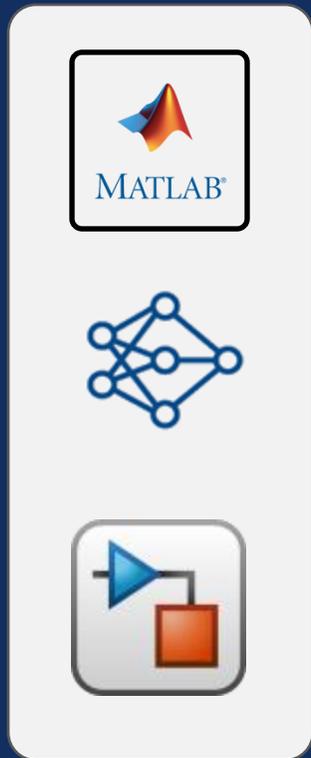


AI Optimization – from **compression** to **deployment**

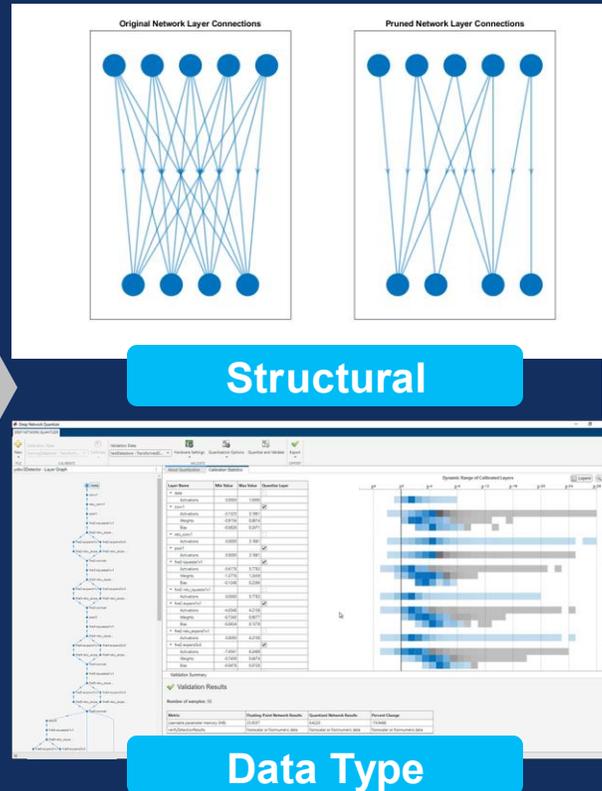


AI Optimization – from **compression** to **deployment**

AI Modeling



Compression



Deployment

Code Generation

CPU

GPU

μC

FPGA



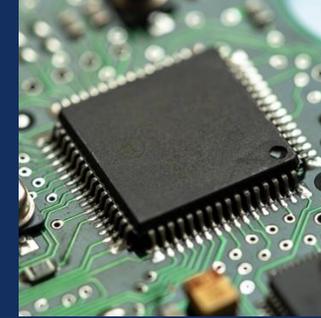
Infineon
PPU - AURIX™ microcontroller



Qualcomm®
Hexagon™ NPU - Snapdragon

TS TECH

Enhance Ride Safety with AI-Powered Driver Fatigue Detection

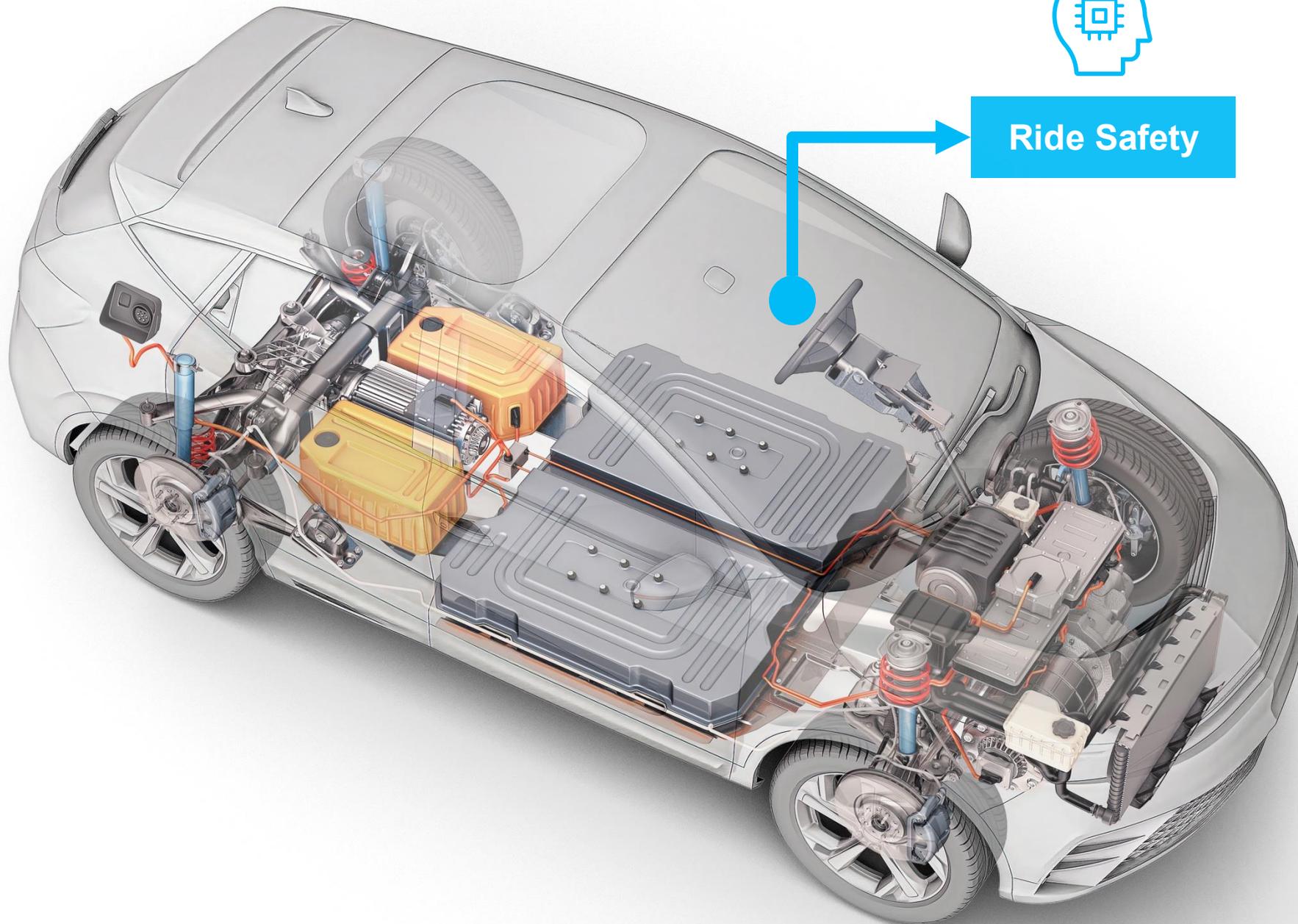


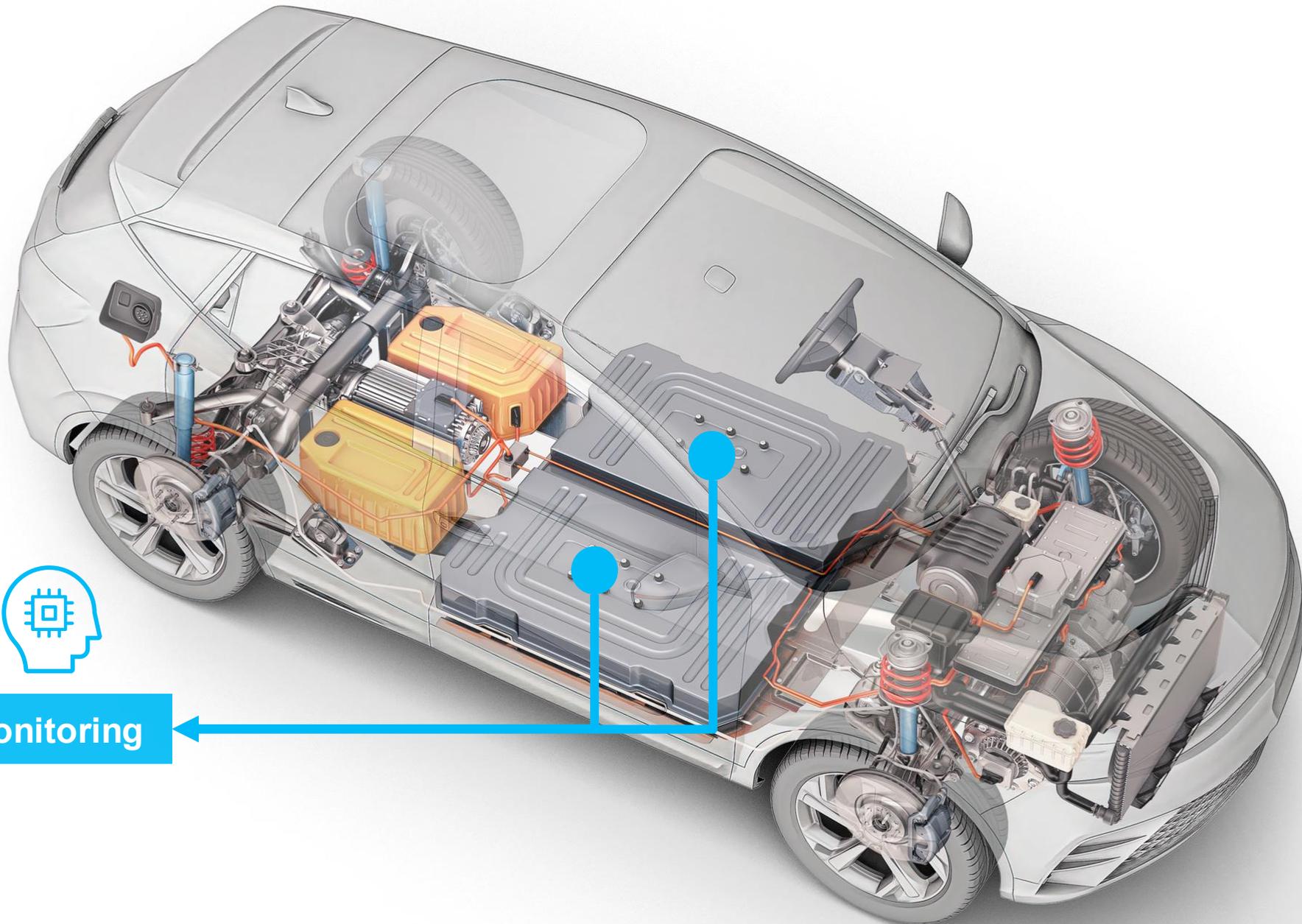
“We were able to realize a service that **combines machine learning algorithms built in Python with other functions in real time** using Simulink.”

- Ryotaro Abe, TS TECH



Ride Safety

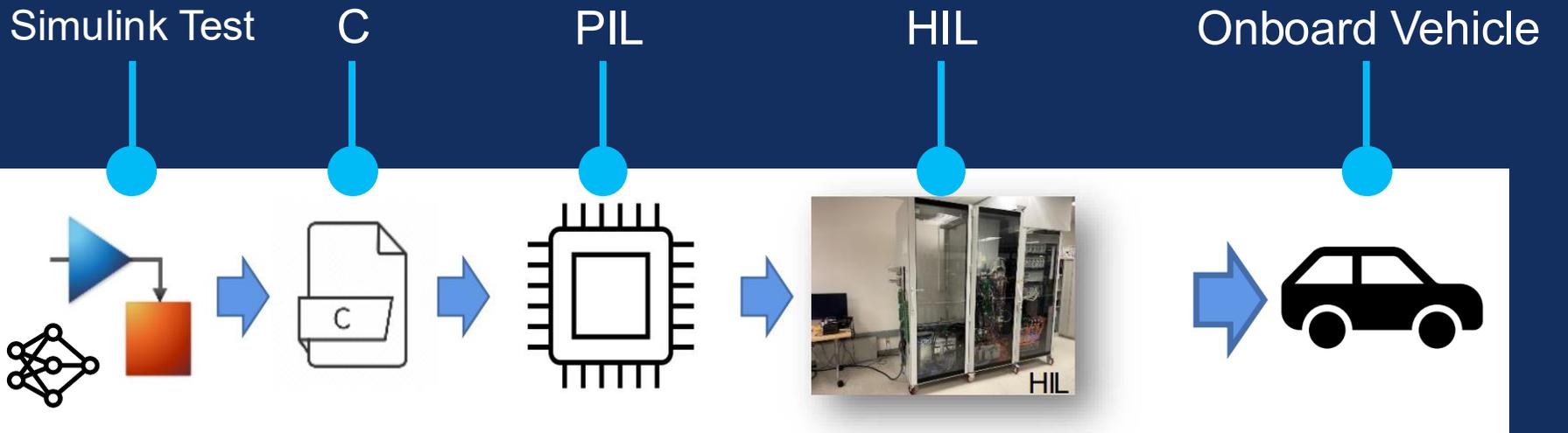
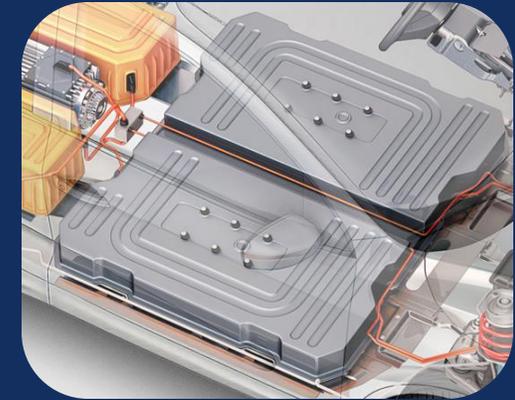
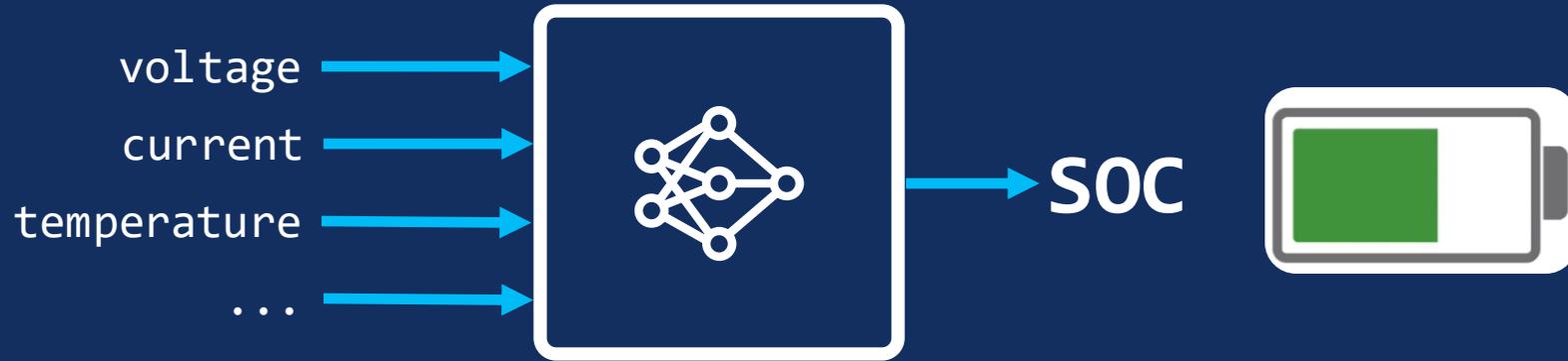




Monitoring

Gotion

Monitor Battery SOC with AI-Powered Estimations



Gotion

Monitor Battery SOC with AI-Powered Estimations

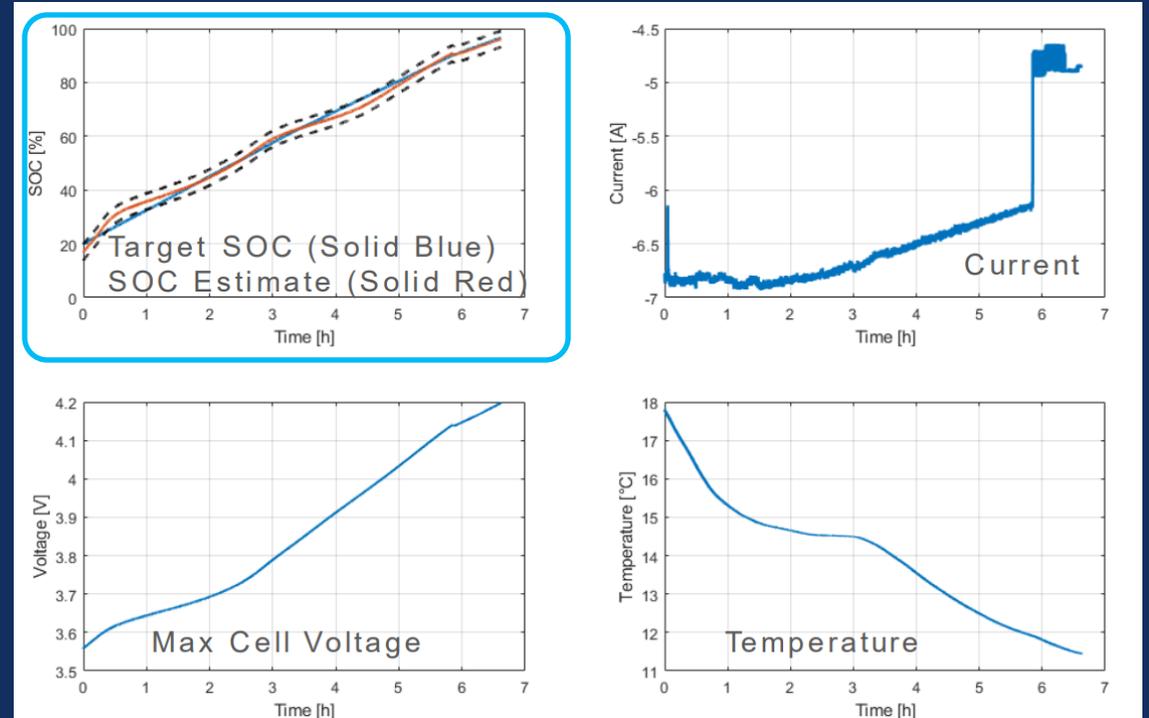
Performance Category	Metrics
Memory Usage	< 2KB ROM < 100B RAM
Execution Time	~ 50 μ s/call
Onboard Prediction Accuracy	> 97%

Size (Memory)

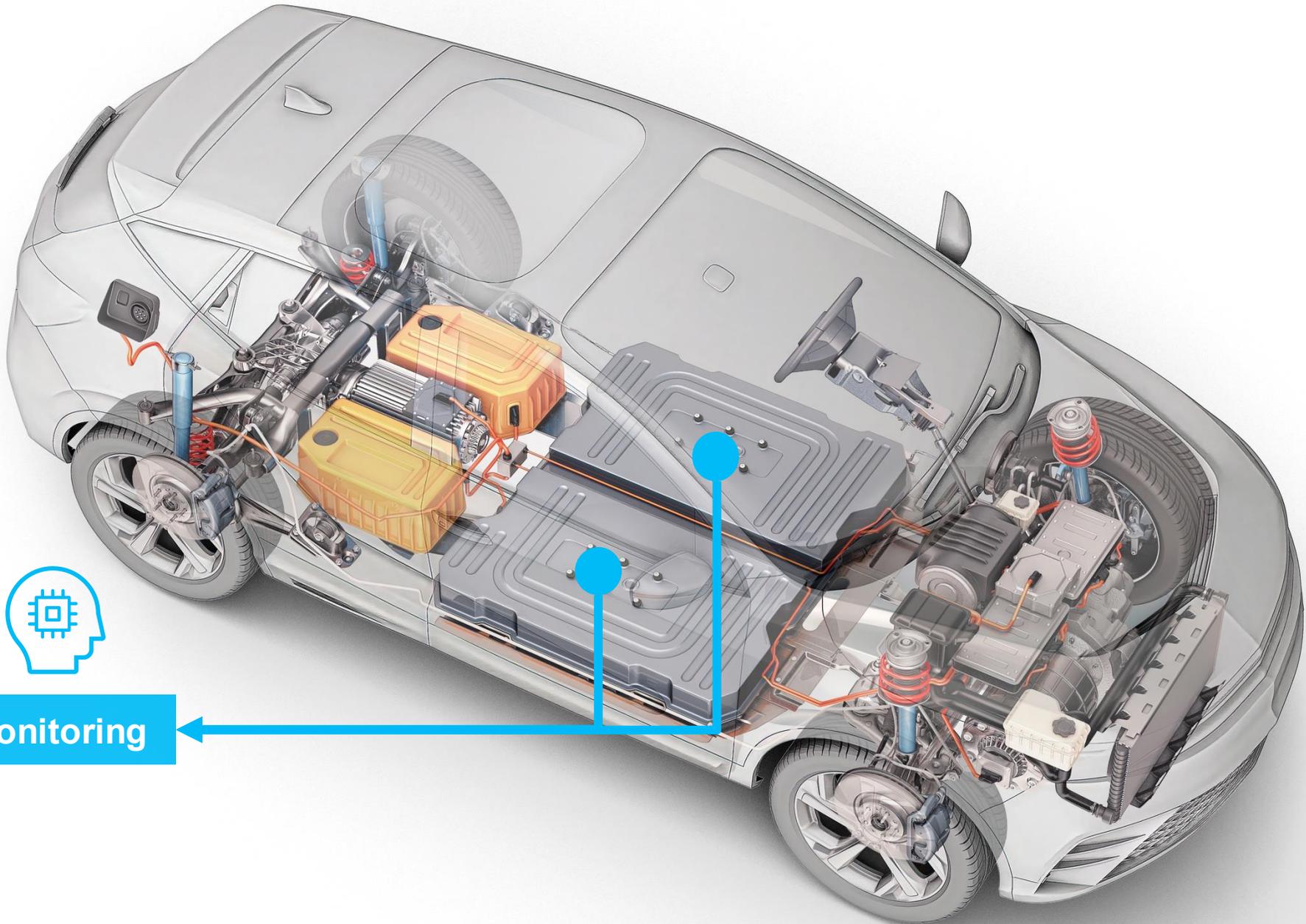


Speed

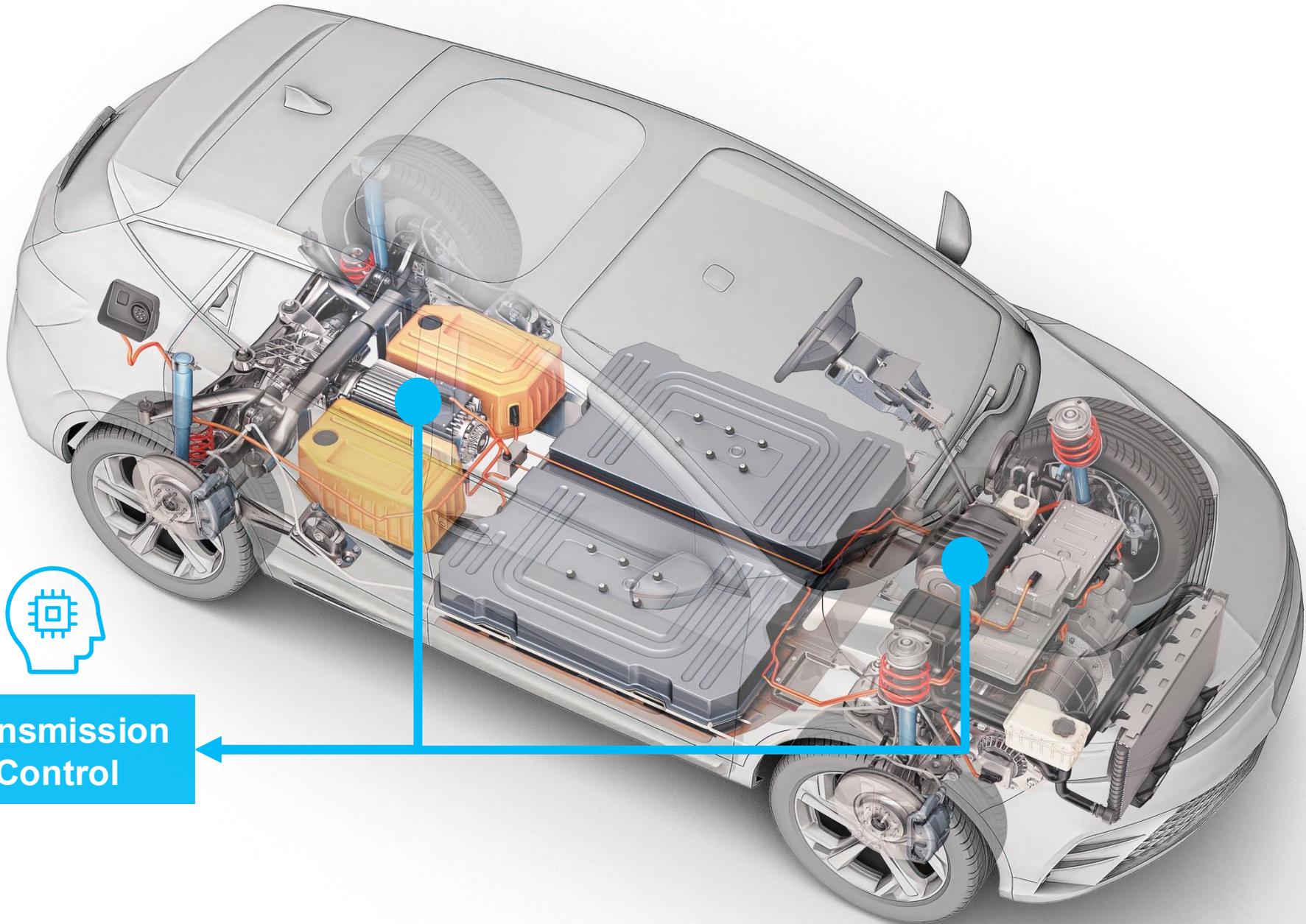
Accuracy



Results from Onboard Vehicle Tests



Monitoring

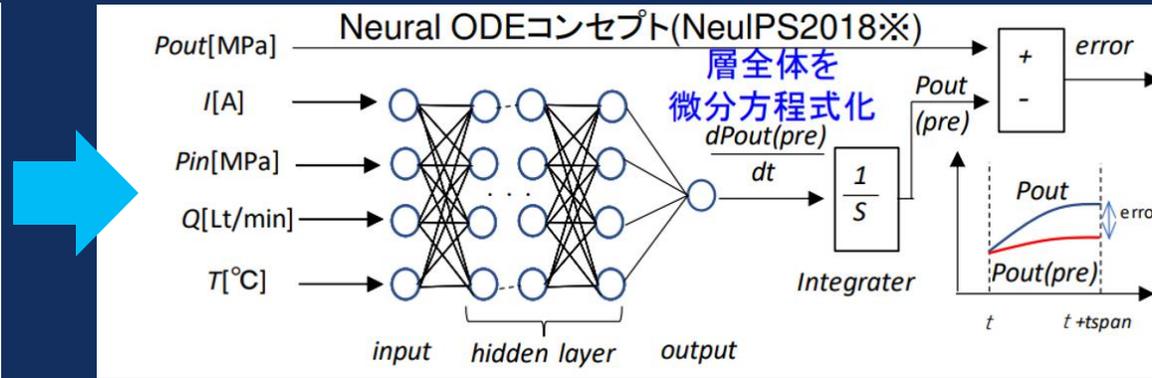


**Transmission
Control**

Subaru

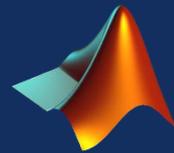
Reduce Analysis Time of Transmission Control System with AI-Powered Reduced Order Models

Hydraulic System



99%

3rd-Party Physical System Model



Neural ODE Model
Trained in MATLAB
Simulated in Simulink



Reduction in simulation time
while maintaining simulation accuracy

AI in Software-Defined Systems

Software-Defined Vehicles



AI-Powered Oversteering Detection
for Stability Control

BMW

Space Exploration



AI-Powered Anomaly Detection
for Satellites

Airbus Defense & Space

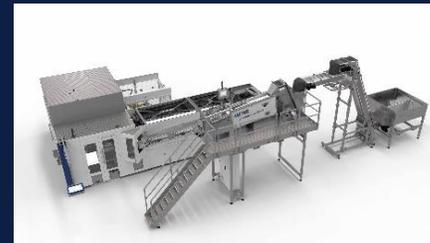
Renewable Energy Systems



AI-Powered Predictive Maintenance
for Offshore Wind Power

Korea Institute of Energy Research

Industrial Systems



AI-Powered Process Control
for Manufacturing PET Bottles

Krones AG

Embedded



Empowered

Predictive AI + Systems

Generative AI

- Smarter systems
- Optimized performance
- Lower component costs

- More productive
- Faster to market
- Lower development costs

Generative AI MATLAB Copilot

The screenshot displays the MATLAB R2025a software interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The Copilot menu is open, showing options: Ask Copilot (Generate code inline), Generate Test (Generate test for the current file), Explain Code (Explain selected code in chat), Generate Comments (Generate code comments for the selected code in editor), and Copilot Settings (Configure MATLAB Copilot). The code editor shows MATLAB code for designing a Butterworth notch filter. The Command Window at the bottom displays the instruction: >> Press `Ctrl+P` to generate code with Copilot. The status bar at the bottom indicates Editor: 100% UTF-8 LF Script Ln 18 Col 1.

```
16 grid on;  
17  
18 % Design a Butterworth notch filter to re  
19 d = designfilt('bandstopiir', ...  
20 'FilterOrder', 2, ...  
21 'HalfPowerFrequency1', 59, ...  
22 'HalfPowerFrequency2', 61, ...  
23 'DesignMethod', 'butter', ...  
24 'SampleRate', Fs);  
25  
26 % Plot the frequency response of the filt  
27 figure;  
28 freqz(d, [], Fs);  
29 title('Frequency Response of the Notch Filter');  
30  
31 % Filter the signal using filtfilt to avoid phase dis  
32 butterLoop = filtfilt(d, openLoop);  
33
```

MATLAB Copilot
AI assistant optimized for MATLAB

- Chat and Learn
- Code Smarter
- Understand and Improve

Chat & Learn

movmedian() vs. hampel()

Copilot Chat

SD Seth

What is the difference between movmedian and hampel?

1. movmedian:

- o This function computes the moving median of a data vector over a specified window size. It is primarily used for smoothing data by replacing each data point with the median of neighboring points within the window.
- o It is effective for reducing noise while preserving the overall trend of the data.

```
% Example of using movmedian
data = randn(1, 100); % Sample data
smoothedData = movmedian(data, 5); % 5-point moving median
plot(data);
hold on;
plot(smoothedData, 'r');
legend('Original Data', 'Smoothed Data');
```

2. hampel:

- o The hampel function applies the Hampel filter, which is designed to detect and replace outliers in a dataset. It uses a sliding window to compute local medians and identifies outliers based on the median absolute deviation (MAD).
- o This method is robust against outliers, making it suitable for datasets with significant noise or extreme values.

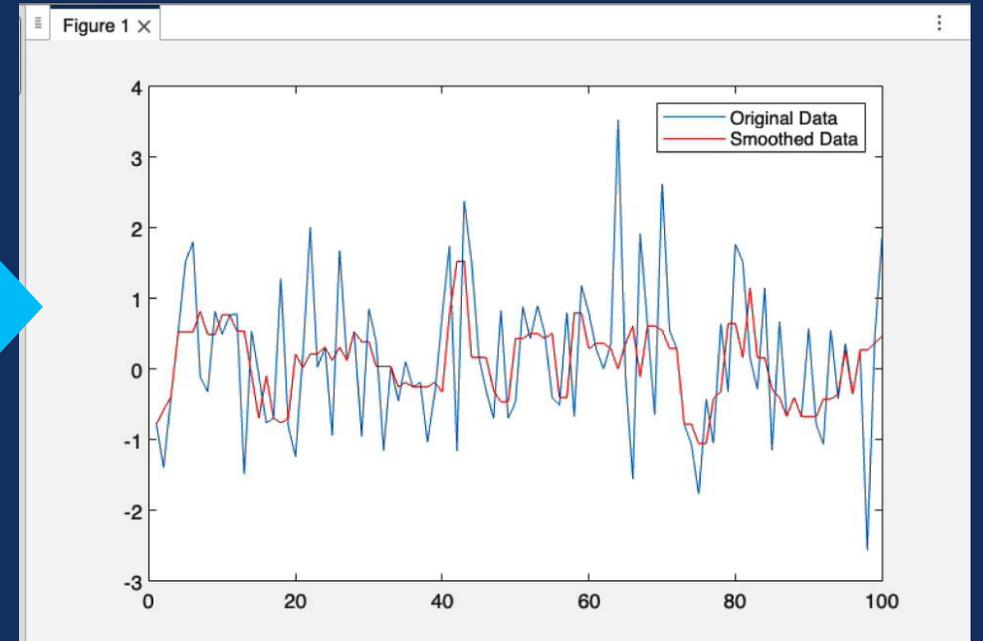
```
% Example of using hampel
noisyData = randn(1, 100); % Sample noisy data
filteredData = hampel(noisyData, 5); % 5-point window for outlier detection
plot(noisyData);
hold on;
plot(filteredData, 'r');
legend('Noisy Data', 'Filtered Data');
```

In summary, use `movmedian`

Stop Generating

Working on your request...

Validate generated output before use.



Generate, Modify, Explain

The screenshot shows the MATLAB IDE interface. The top menu bar includes HOME, PLOTS, APPS, EDITOR, PUBLISH, and VIEW. The main workspace contains a code editor for 'PlotCurrentTorque.m'. A Copilot suggestion box is overlaid on the code, containing the text: 'Put both lines on the same chart, with torque on the left and current on the right'. Below this text are buttons for 'Accept', 'Discard', and 'View Diff'. The code in the editor is as follows:

```
1 % Create a new figure
2 figure;
3
4 % Create axes for the current and torque
5 yyaxis left
6 plot(time, current, 'LineWidth', 1);
7 ylabel('Current (A)');
8 grid on
9 title('Motor Current and Load Torque');
10
11 yyaxis right
12 plot(time, -torque, 'LineWidth', 1);
13 ylabel('Torque (N*m)');
14 ylim([-3e-4 5e-5]);
15
16 % Add x-axis label
17 xlabel('Time (s)');
18
```

At the bottom of the IDE, the Command Window is visible, showing the text: 'Command Window'.

The screenshot shows the MATLAB Command Window. It contains the following text:

```
>> mysine
>> mysine
Unrecognized function or variable 'sine'.

Error in mysine (line 3)
y = sine(x);
~~~~~
```

Below the error message, there is a button labeled 'Explain Error'.

The screenshot shows the Copilot explanation for the error. It includes the following text:

Probable Explanation: This error indicates that MATLAB cannot find a function or variable named 'sine'. It usually occurs when the function is not defined or misspelled.

Troubleshooting Recommendation: Check if you have defined the function 'sine' in your code or if it is a built-in function. If it's a custom function, ensure it's in the current folder or on the MATLAB path. If you meant to use the sine function, use `sin(x)` instead.

use `sin(x)` instead.

Autocomplete

The screenshot shows the MATLAB IDE interface with the following components:

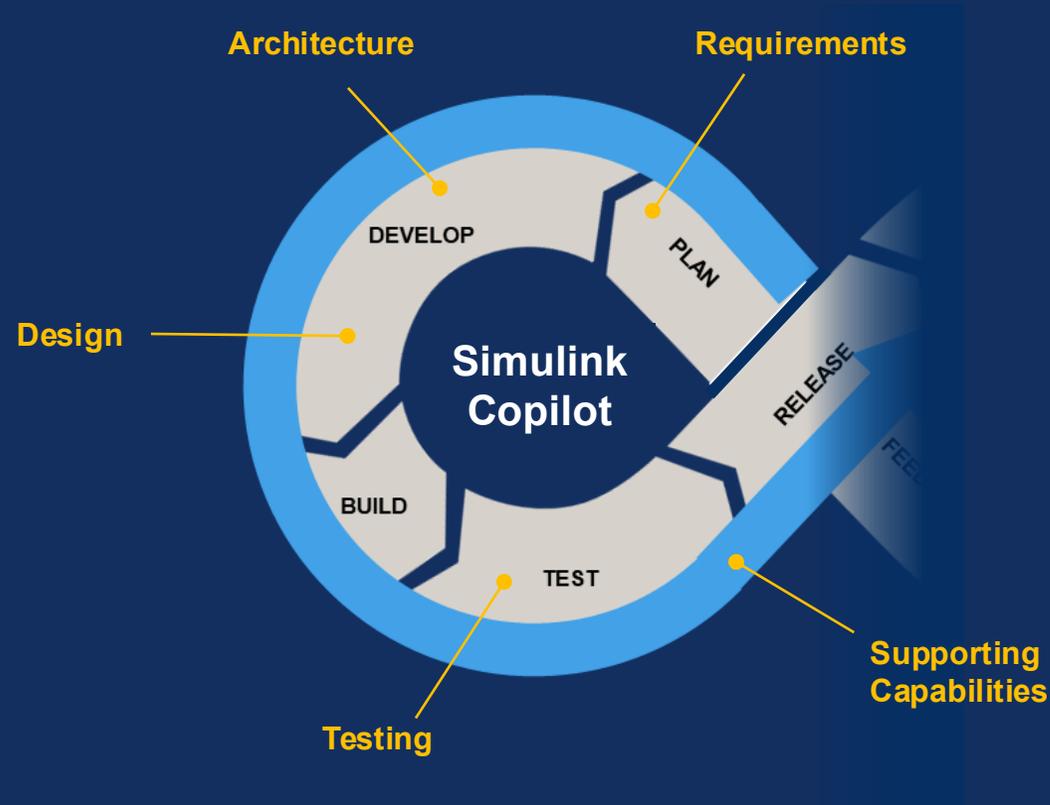
- Top Tabs:** HOME, PLOTS, APPS, EDITOR (selected), PUBLISH, VIEW.
- Toolbars:** FILE (New, Open, Save, Print, Compare), NAVIGATE (Go To, Find, Bookmark), CODE (Copilot, Refactor), and ANALYZE (Profiler, Code Issues, Debugger).
- Path:** / > Users > sdeland > Documents > MATLAB >
- Editor:** A file named 'mysphere.m' is open. The code is as follows:

```
1 [x,y,z] = sphere;  
2 r = 2;  
3 surf(x*r,y*r,z*r)  
4 axis equal  
5  
6 % Calculate the surface area and volume  
7 % Surface area of the sphere  
8 surfaceArea = 4 * pi * r^2;  
  
% Volume of the sphere  
volume = (4/3) * pi * r^3;  
  
% Display the results  
fprintf('Surface Area: %.2f\n', surfaceArea);  
fprintf('Volume: %.2f\n', volume);
```

Annotations in the image:

- A yellow box highlights the code from line 1 to line 6, labeled "Typed Code".
- A blue box highlights the code from line 7 to line 8, labeled "Suggested Code".

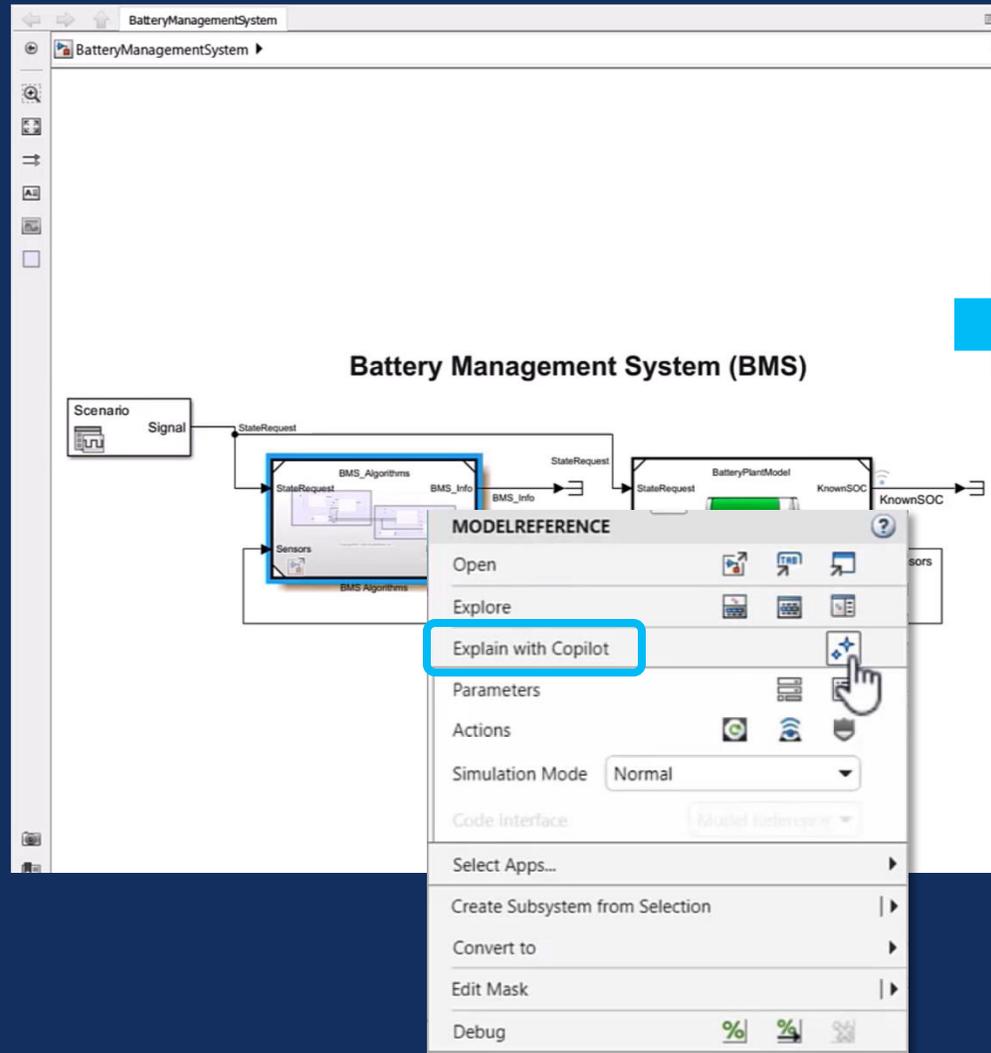
Generative AI Simulink Copilot



A GenAI assistant designed to improve your productivity **throughout** your development lifecycle

Copilot is built to improve engineers' productivity

Gain insights
Get advice
Suggest fixes
Create models



Tianyi

Can you walk me through this model?

which encompasses specific functionalities, promoting reliability and ease of maintenance.

Critical Parameters

- **Current Limits:** Parameters that define maximum charge and discharge currents based on temperature and voltage conditions are critical for protecting the battery.
- **State Variables:** Variables such as SOC and temperature are essential for monitoring the battery's health and performance.

Key Components

- **Signal Editor:** This component allows for the display, creation, editing, and switching of interchangeable scenarios, facilitating testing and validation of different operating conditions.
- **BMS Algorithms:** This subsystem contains the core algorithms for battery management, including state management and control logic.
- **Plant Model:** This subsystem models the physical characteristics of the battery, simulating its behavior under various conditions.

The BMS model is a comprehensive system that integrates various functionalities to ensure the safe and efficient operation of battery systems, making it a critical component in applications such as electric vehicles and renewable energy storage systems.

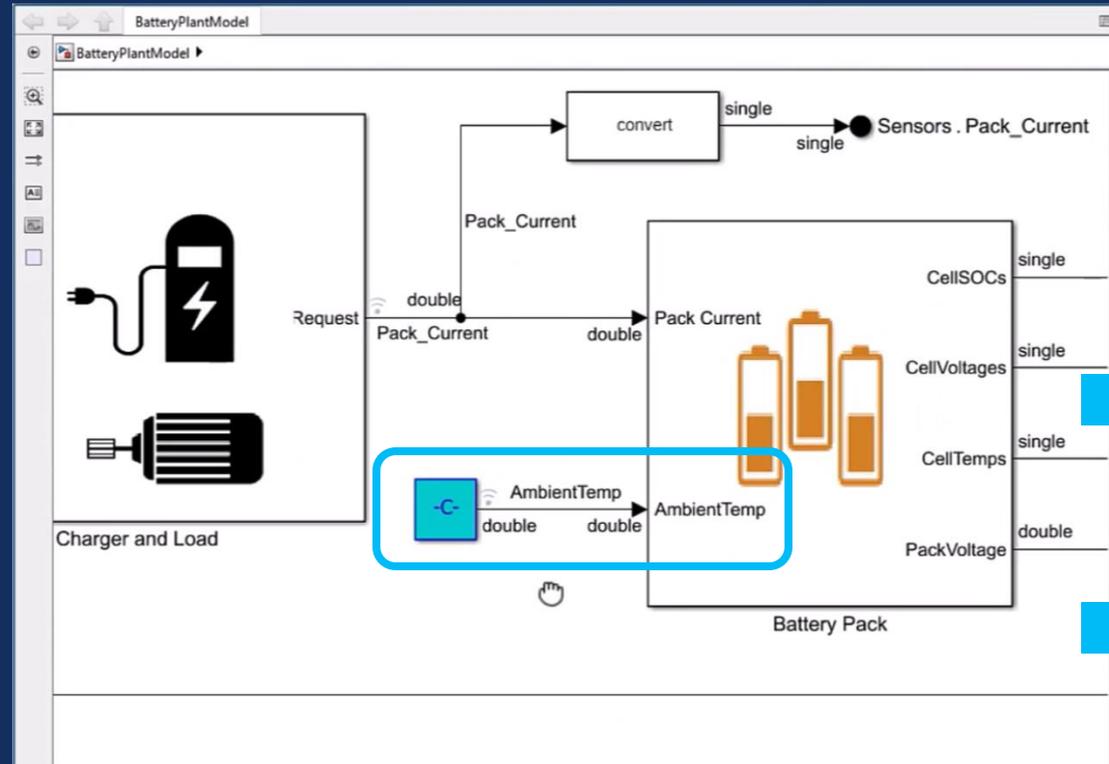
If you have any specific areas of the model you'd like to explore further or any questions, feel free to ask!

👍 🗨️ 📄

Ask Copilot

Copilot is built to improve engineers' productivity

Gain insights
Get advice
Suggest fixes
Create models



Copilot Chat (Beta)

How to implement a spike in ambient temp? Point me to the place in the model for changing this.

Copilot
Reasoned for 6 sec >

Retrieved the location in the BatteryPlantModel where ambient temperature is controlled.

To implement a spike in ambient temperature in your BatteryPlantModel, you can modify the output of the [Constant](#) block, which currently defines the ambient temperature as a constant value (Battery.Templnit).

This block's output is connected to the input port labeled In2 of the [Battery Pack](#) subsystem, where it serves as the ambient temperature input for thermal calculations.

To create a spike in ambient temperature:

1. Replace the [Constant](#) block with a dynamic signal source, such as a Step block or Signal Builder, to define the spike behavior (e.g., magnitude, timing, and duration).
2. Configure the signal source to introduce the desired spike in the ambient temperature.
3. Simulate the model to observe the effects of the spike on the battery system.

Let me know if you need further assistance in configuring the signal source or simulating the model!

👍 🗨️ 📄

Copilot is built to improve engineers' productivity

Gain insights
Get advice
Suggest fixes
Create models

The screenshot displays the MATLAB/Simulink environment. The main workspace shows a Simulink model titled "Battery Plant Model". Below the workspace is the "Diagnostic Viewer" window, which contains the following error message:

```
Data type mismatch. 'Output Port 1' of 'BatteryPlantModel/Charger and Load' is a signal of data type 'single'. However, it is driving a signal of data type 'double'.
```

Underneath the error message, there are "Suggested Actions":

- Explain this with copilot

A blue arrow points from the "Explain this with copilot" action to the "Copilot Chat (Beta)" window on the right. The Copilot Chat window shows the following content:

Copilot Reasoned for 6 sec >

Analyzed the data type mismatch error involving the 'Output Port 1' of 'BatteryPlantModel/Charger and Load'

The error occurs due to a "data type mismatch" in your model. Specifically, the output port of the [Charger and Load](#) block produces a signal of type 'single', while it connects to another block expecting a signal of type 'double'. This situation arises when different blocks in your Simulink model are configured to use conflicting data types.

Possible Solution:

- Option 1: Change the Output Data Type**
 - Access the parameters of the [Charger and Load](#) block.
 - Modify its output data type to 'double' to match the downstream block's requirements.
- Option 2: Add a Data Type Conversion Block**
 - If you prefer to retain the 'single' output, insert a **Data Type Conversion** block between the [Charger and Load](#) block and the downstream block.
 - Configure the Data Type Conversion block to convert the signal from 'single' to 'double'.

Both options will resolve the data type mismatch, and the choice depends on whether you need to keep the signal as 'single' for other parts of the model or are comfortable changing it to 'double' at the source.

Let me know if you'd like additional guidance!

At the bottom of the Copilot Chat window, there are icons for thumbs up, thumbs down, and a copy icon.

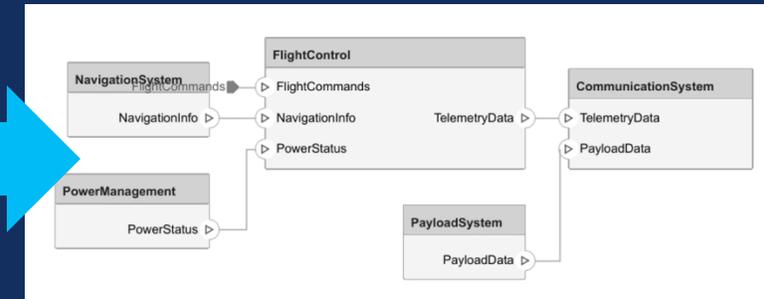
Error

Copilot is built to improve engineers' productivity

Gain insights
Get advice
Suggest fixes
Create models

Requirements

- Aircraft Capabilities
 - Airworthiness
 - 1.1.1 Range: The aircraft shall be controllable for all distances within line-of-sight
 - 1.1.2 Rain Conditions: All aircraft components shall be able to operate under rain for at least 10 minutes
 - 1.1.3 Power: Aircraft shall provide power to operate all onboard systems for the complete duration of the flight
 - 1.1.4 Emergency Power: In case of power generation failure, aircraft shall provide emergency power for up to 20 minutes of flight for all flight-critical systems.
 - 1.1.5 Control Surface Fault-Tolerance: Aircraft shall be tolerant to single-servo failures
 - 1.1.6 Fuel: Aircraft propulsion shall be capable of operating with 89 Octane Gasoline
 - 1.1.7 No Payload Flights: Aircraft shall be able to safely get airborne without a payload
 - 1.1.8 Flight Data Recorder: Aircraft shall provide the means to record all flight data from every component on board for the duration of the flight
 - 1.1.9 Flight Identification: Aircraft shall provide an ADSB transponder on board for proper identification
 - 1.1.10 Lights: Aircraft shall provide aeronautical standard navigation and strobe lights
 - 1.1.11 Flight Computer: Aircraft shall provide a flight computer to autonomously conduct safe flight operations from launch to recovery
 - 1.1.12 Endurance: The aircraft shall be able to sustain uninterrupted flights for up to three hours
 - Flight Control: Aircraft shall provide a dedicated and permanent communications link for command and control for all the range of operations
 - 2.2.2 Payload: Aircraft shall provide a dedicated and permanent payload data link for all the range of operations
- 1.3 Payload Capabilities
 - 1.3.1 Carrying Capacity: Aircraft shall be able to carry up to 2.2 Kg of payload
 - 1.3.2 Payload Bay Capacity: Aircraft shall provide a payload bay with at least 300 cm^3
 - 1.3.3 Default Payload: Aircraft shall provide a pre-installed payload in the form of a gimbal-stabilized visual camera providing Full-Motion video.
 - 1.3.4 Payload Protection: Aircraft shall be provided a universal payload-retraction system to protect the payload during takeoff and landing
- 1.4 Construction
 - 1.4.1 Modularity: Aircraft shall be constructed in modular components for easy spare part procurement and replacement



System Composer

1. Requirements Specification

1.1 Functional Requirements

1.1.1. Gear Shift Determination

The Automatic Transmission Gear System shall determine the appropriate gear based on current gear, throttle input, and vehicle speed.

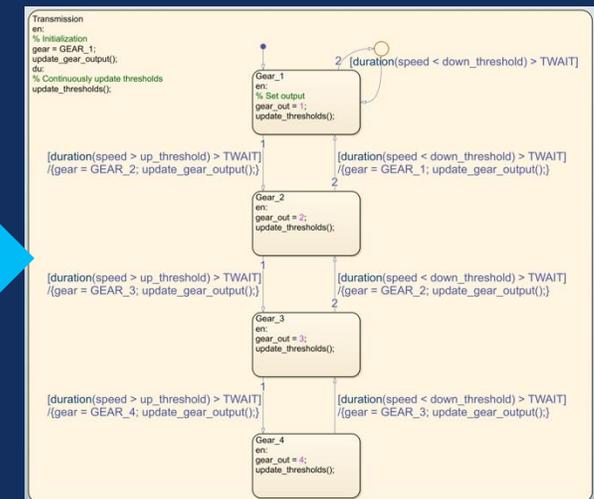
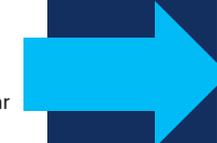
1.1.2. Threshold Calculation

The system shall calculate up_threshold and down_threshold values using the current gear and throttle input.

1.1.3. Upshift Condition

If the vehicle speed remains above the upshift threshold for a duration longer than the calibration time period (TWAIT), the System shall initiate an upshift to the next higher gear.

1.1.4. Downshift Condition



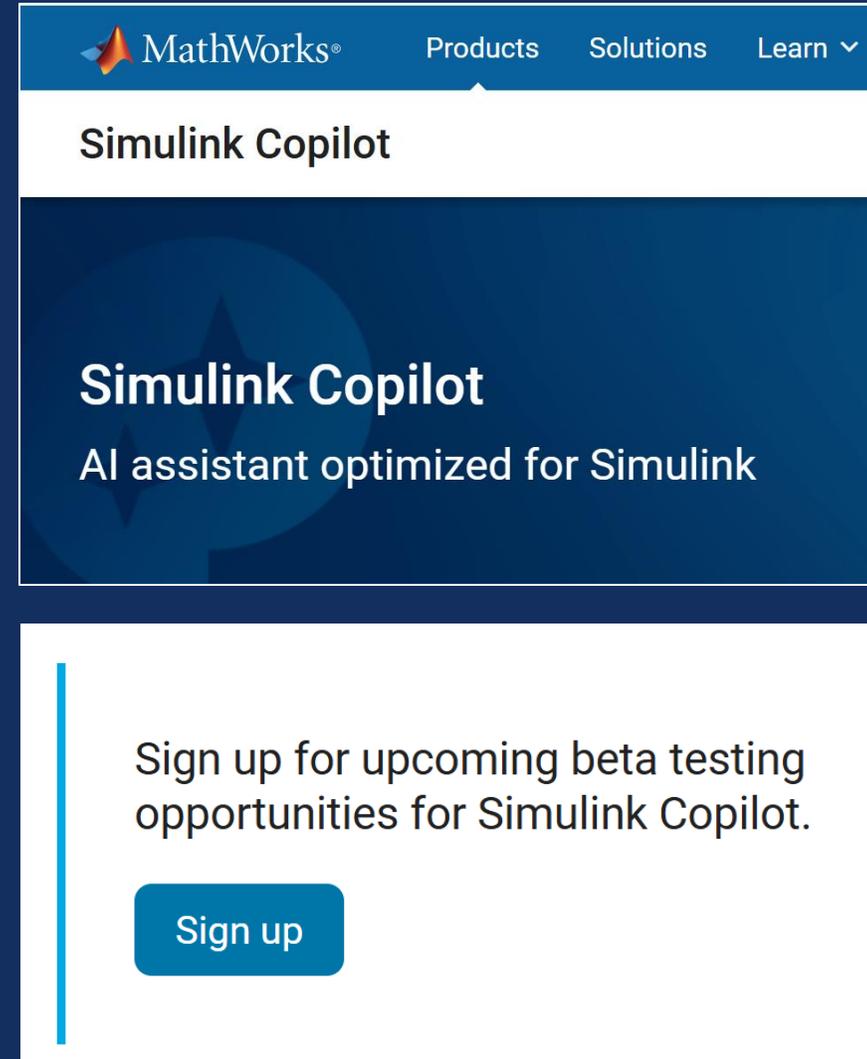
Simulink / Stateflow

Simulink Copilot will be available soon...

Connect with me

Participate in a beta by registering your interest

Discuss tasks you would like to accelerate with Generative AI



The screenshot shows the MathWorks website header with the logo and navigation links: Products, Solutions, and Learn. Below the header, the page title is "Simulink Copilot". The main content area features a dark blue background with a large speech bubble icon and the text "Simulink Copilot" and "AI assistant optimized for Simulink". At the bottom, there is a white section with a blue vertical line on the left, containing the text "Sign up for upcoming beta testing opportunities for Simulink Copilot." and a blue "Sign up" button.

Embedded



Empowered

Generative AI

- More productive
- Faster to market
- Lower development costs

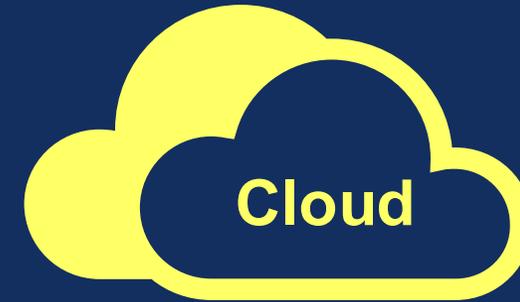
Embedded



Empowered

Generative AI

- More productive
- Faster to market
- Lower development costs



- Increased virtualization
- Higher scalability
- More collaboration

In my **Software-Defined System**, can we release a major feature with no hardware changes, leveraging the **cloud**? ☁️

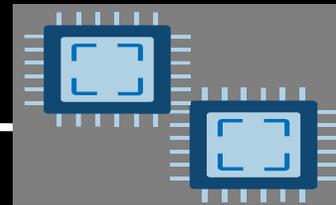
“Sport+” Mode

▲ Reduce 0-60mph time

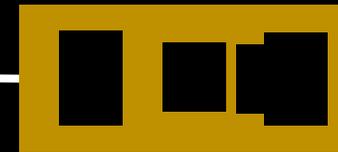
↔️ Minimum Range Change



Infotainment System (IVI)
QM



Vehicle Control Unit (HPC)
ASIL-B



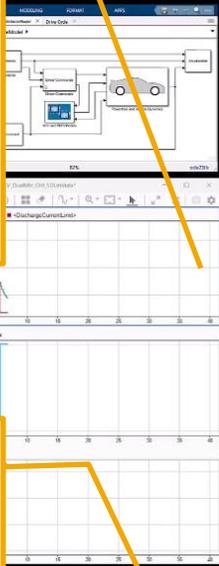
Battery Management
(Embedded Edge)
ASIL-D





- Run parallel simulations
- Automate model checks, simulations, report generation in CI pipelines

MathWorks
Simulation in MATLAB & Simulink



- Automatically generate code, interactively or in CI pipeline

Elektrobit

- Invoke function from instrument cluster

SYNOPSIS

- Integrate application code with production middleware

- Integrate with detailed virtual ECU

Automotive Software Development in the Cloud

powered by 

Interactive Design and Development
with MATLAB & Simulink



Simulation & Design Exploration at Scale



Integrate Automated Testing and CI
with MATLAB & Simulink



Elektrobit
Road-ready Automotive Software

Infotainment (Android Automotive OS)

Deploy/Operationalize
MATLAB Code and Simulink Models

Battery Management System (Classic AUTOSAR)

Time	Class	ECU	Event	Value	Unit	Charge	Current	Level	State
1281.82	Classic	AUTOSAR	ECU	sending discharge current	1282	-	318.88		charge
1282.82	Classic	AUTOSAR	ECU	receiving discharge current	1282	-	318.88		charge
1283.82	Classic	AUTOSAR	ECU	receiving discharge current	1283	-	318.88		charge
1284.82	Classic	AUTOSAR	ECU	receiving discharge current	1284	-	318.88		charge
1285.82	Classic	AUTOSAR	ECU	receiving discharge current	1285	-	318.88		charge
1286.82	Classic	AUTOSAR	ECU	receiving discharge current	1286	-	318.88		charge
1287.82	Classic	AUTOSAR	ECU	receiving discharge current	1287	-	318.88		charge
1288.82	Classic	AUTOSAR	ECU	receiving discharge current	1288	-	318.88		charge
1289.82	Classic	AUTOSAR	ECU	receiving discharge current	1289	-	318.88		charge
1290.82	Classic	AUTOSAR	ECU	receiving discharge current	1290	-	318.88		charge
1291.82	Classic	AUTOSAR	ECU	receiving discharge current	1291	-	318.88		charge
1292.82	Classic	AUTOSAR	ECU	receiving discharge current	1292	-	318.88		charge
1293.82	Classic	AUTOSAR	ECU	receiving discharge current	1293	-	318.88		charge
1294.82	Classic	AUTOSAR	ECU	receiving discharge current	1294	-	318.88		charge
1295.82	Classic	AUTOSAR	ECU	receiving discharge current	1295	-	318.88		charge
1296.82	Classic	AUTOSAR	ECU	receiving discharge current	1296	-	318.88		charge
1297.82	Classic	AUTOSAR	ECU	receiving discharge current	1297	-	318.88		charge
1298.82	Classic	AUTOSAR	ECU	receiving discharge current	1298	-	318.88		charge
1299.82	Classic	AUTOSAR	ECU	receiving discharge current	1299	-	318.88		charge
1300.82	Classic	AUTOSAR	ECU	receiving discharge current	1300	-	318.88		charge

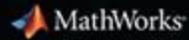


ECU Simulation

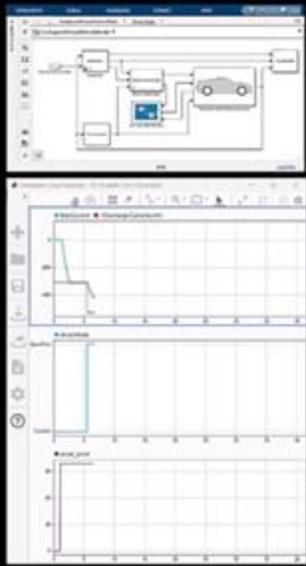
Automotive Software Development in the Cloud

1

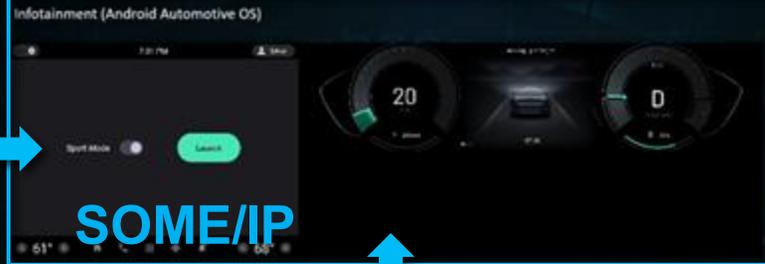
powered by 



Virtual Vehicle Simulation in MATLAB & Simulink



Road-ready Automotive Software



Vehicle Control Unit (Adaptive AUTOSAR)

```
[209_18] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[209_18] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[209_19] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[209_20] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[210_21] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[211_22] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[212_23] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[213_24] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[214_25] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[215_26] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[216_27] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[217_28] Adaptive AUTOSAR HPC - received discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
```

Battery Management System (Classic AUTOSAR)

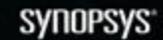
```
[281_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[281_03] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[282_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[283_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[284_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[285_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[286_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[287_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[288_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[289_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[290_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[291_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
[292_02] Classic AUTOSAR ECU - sending discharge current limit: -310.00 charge current limit: 182.30 charge: 0.75
```



Application Code from Model



ECU Simulation



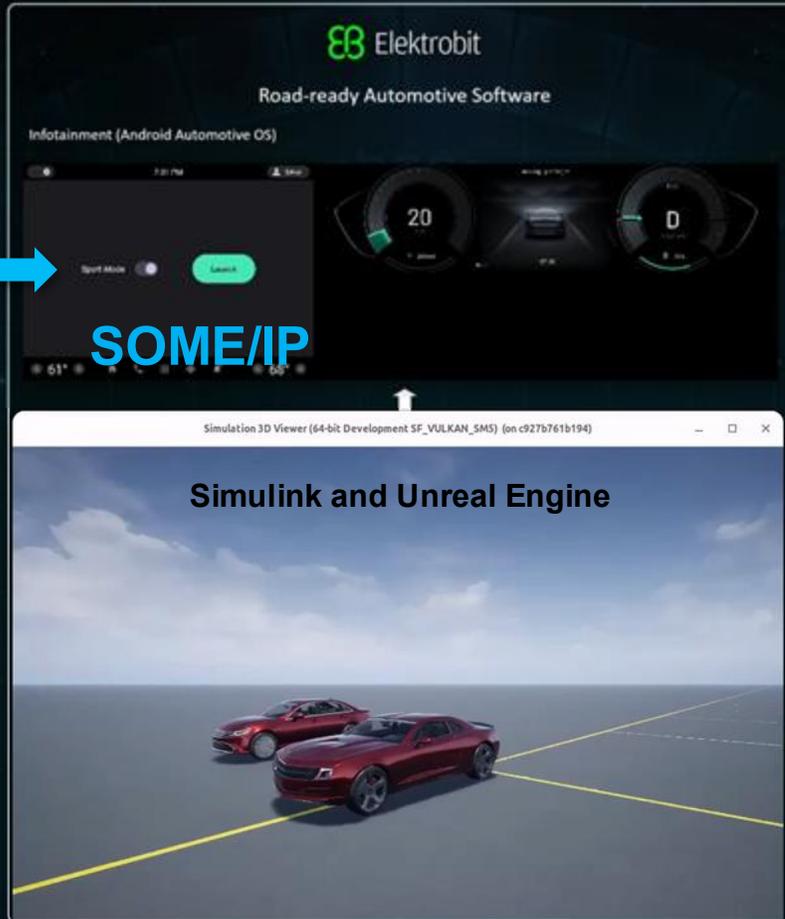
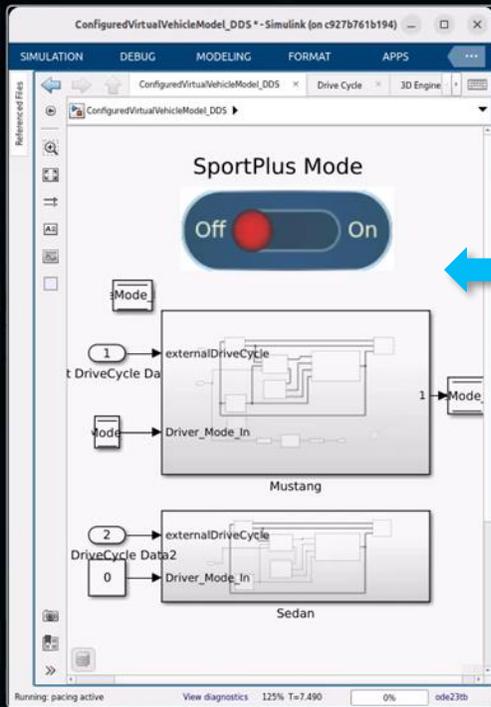
Virtual ECUs in Synopsys Silver



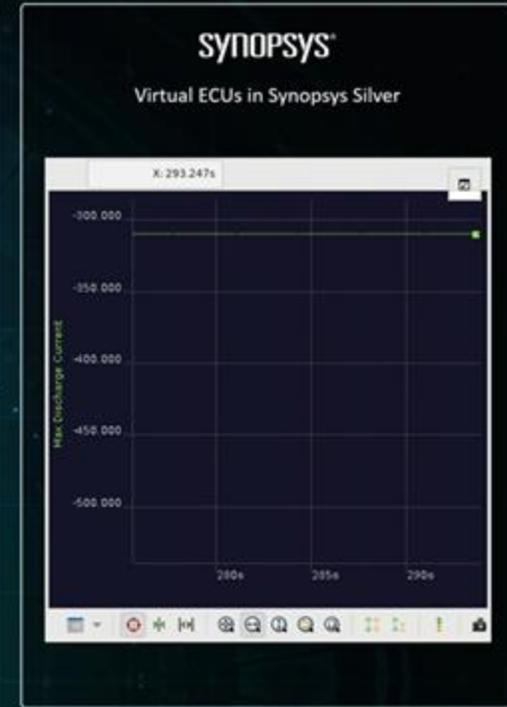
Automotive Software Development in the Cloud

1

powered by 



The Elektrobit software interface is divided into two main sections. The top section, titled 'Road-ready Automotive Software', displays a virtual infotainment system (Android Automotive OS) with a speedometer showing 20, a gear indicator showing 'D', and a 'Sport Mode' toggle. A blue arrow points from the Simulink 'SportPlus Mode' toggle to this 'Sport Mode' toggle, with the text 'SOME/IP' overlaid in blue. The bottom section, titled 'Simulation 3D Viewer (64-bit Development SF_VULKAN_SMS)', shows a 3D rendering of two red cars on a flat surface under a blue sky.

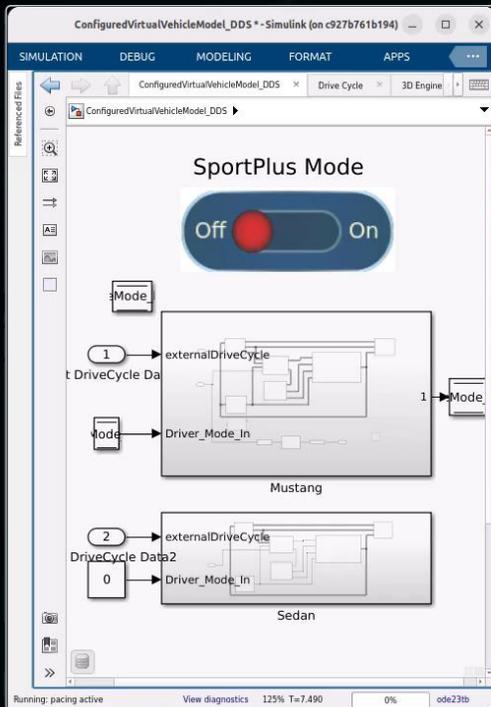



Application Code from Model

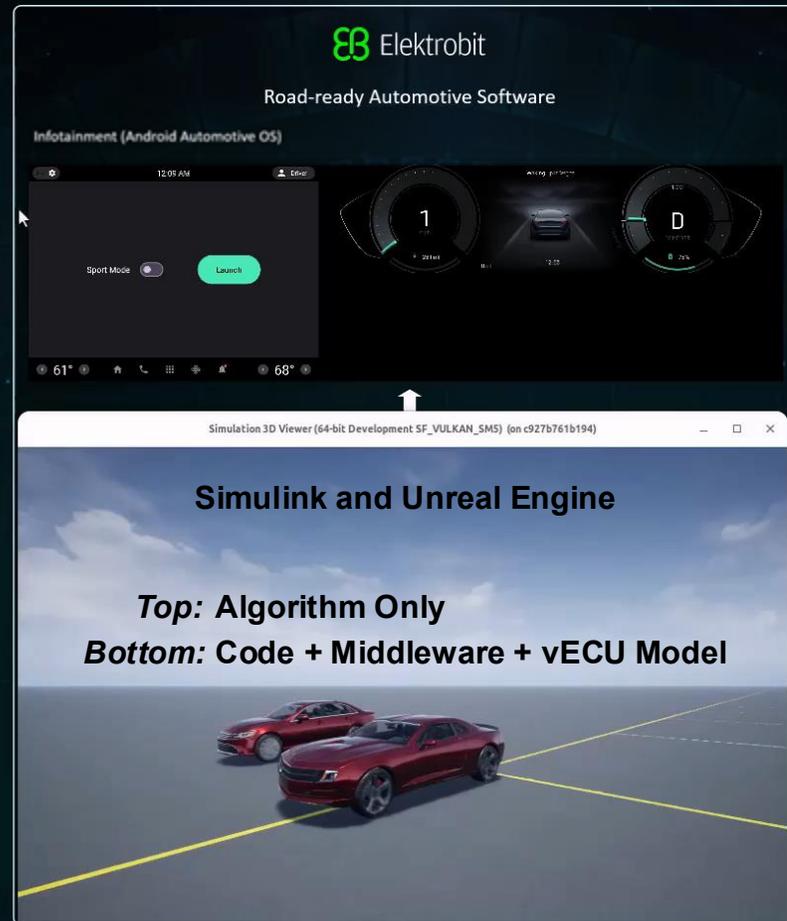

ECU Simulation

Automotive Software Development in the Cloud

powered by 




Application Code from Model



Elektrobit
Road-ready Automotive Software

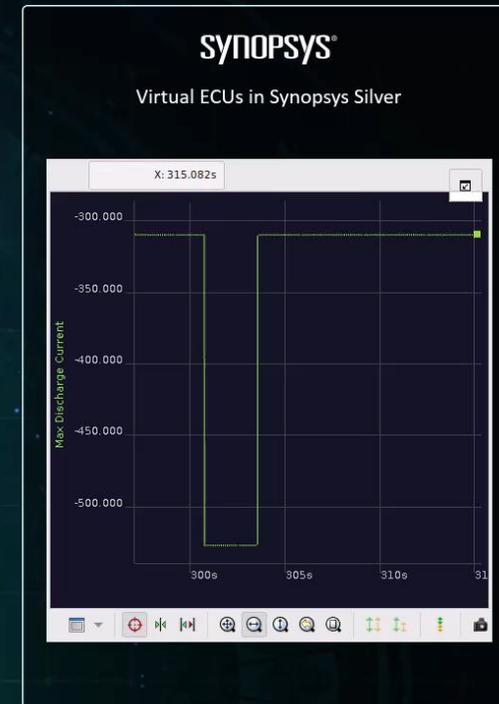
Infotainment (Android Automotive OS)

Sport Mode Launch

Simulation 3D Viewer (64-bit Development SF_VULKAN_SM5)

Simulink and Unreal Engine

Top: Algorithm Only
Bottom: Code + Middleware + vECU Model




ECU Simulation

Software-Defined Products



AI + Software-Defined Systems



Intelligence



Reliability



Functional Safety



Physical Components



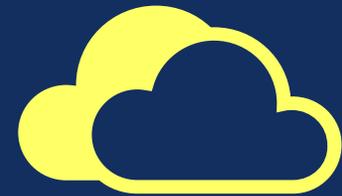
Embedded



Empowered



Building AI in Systems



Enabling AI and Cloud for Development

Thank You



© 2025 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

