

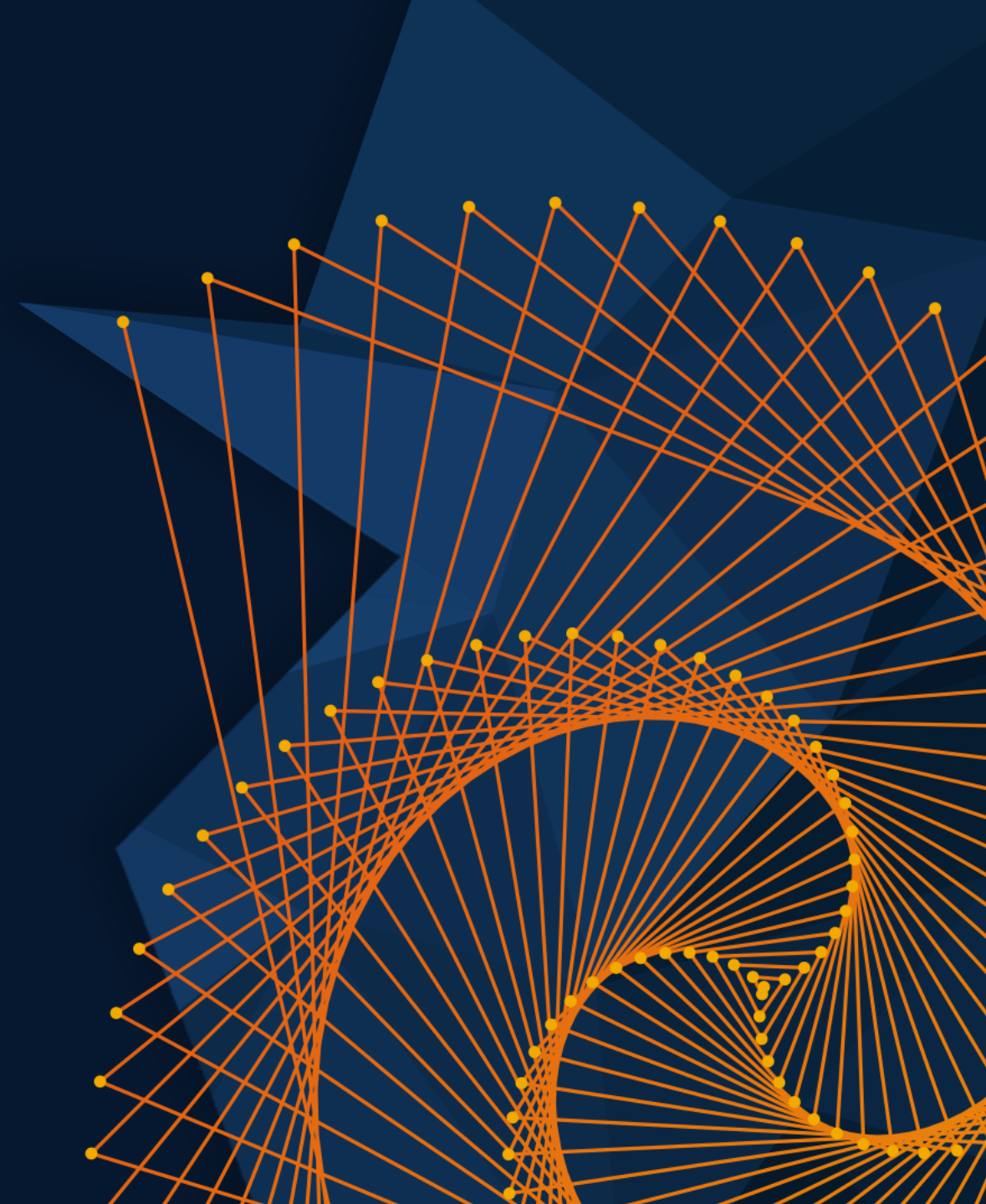
# MATLAB EXPO

June 11, 2024 | Korea

---

## Model-Based Design for Digital Engineering: Impact and Directions

*Arun Mulpur*



## Digital Engineering:

A holistic approach to designing complex engineered systems

Use **models** instead of documents

Integrate **data** across models

Evolve design team **culture**

# Model-Based Design for Digital Engineering



# Application Trends



**Autonomous**



**Connectivity**



**Electrification**

# Application Trends



**Autonomous**

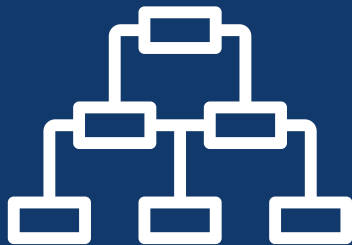


**Connectivity**



**Electrification**

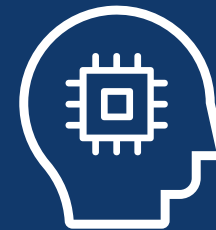
# Workflow Trends



**Systems Engineering  
& Design**



**Modern  
Software Practices**



**AI for  
System Development**

## Application Trends



Autonomous

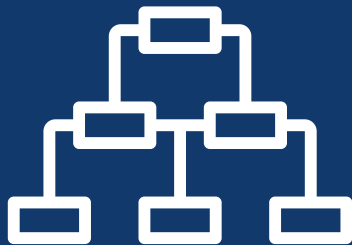


Connectivity



Electrification

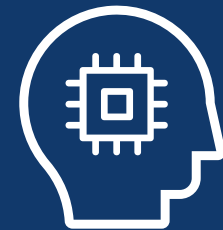
## Workflow Trends



Systems Engineering  
& Design

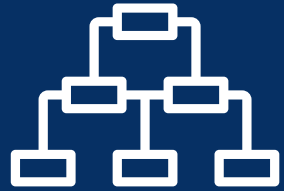


Modern  
Software Practices



AI for  
System Development

# Workflow Trends



1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early

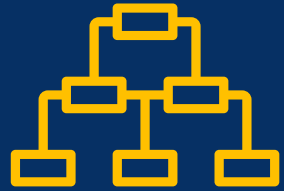


5. Apply standard software workflows
6. Design and simulate in the cloud



7. Design your system with AI

# Workflow Trends



1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early



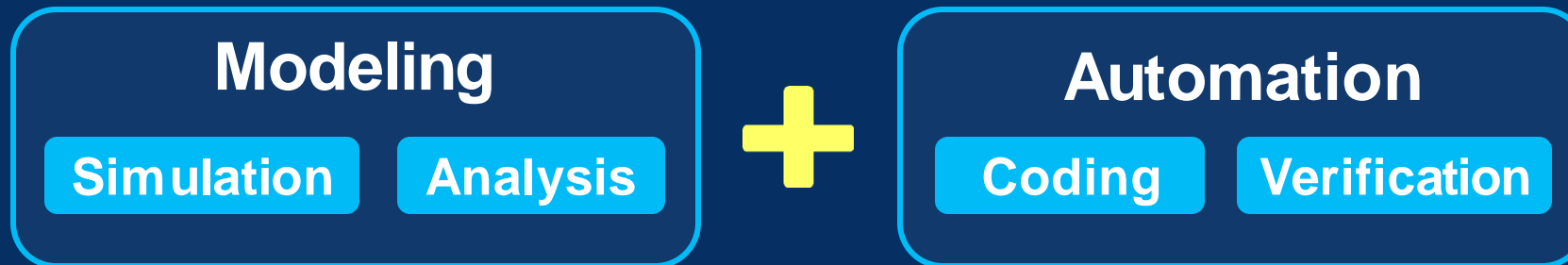
5. Apply standard software workflows
6. Design and simulate in the cloud



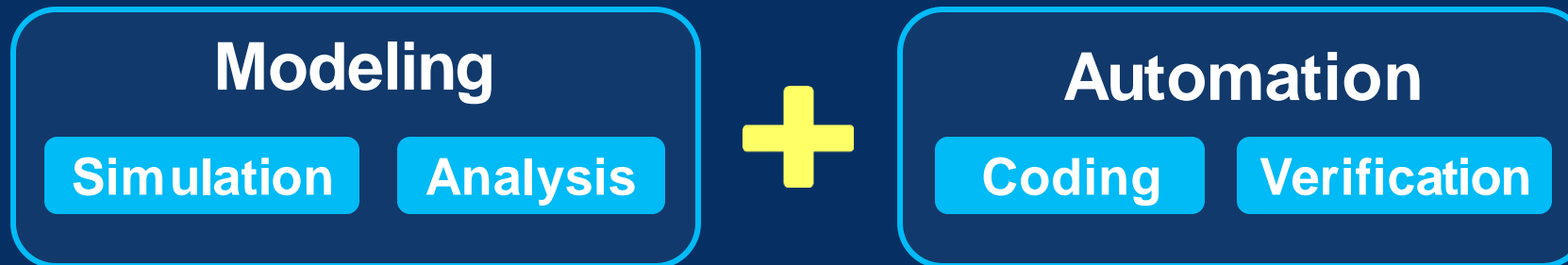
7. Design your system with AI



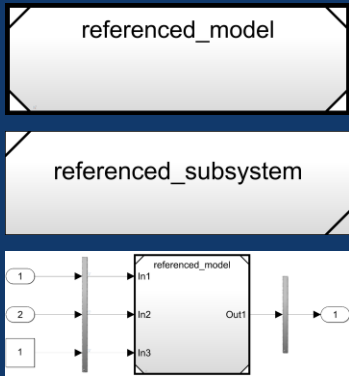
# ① Automate everything



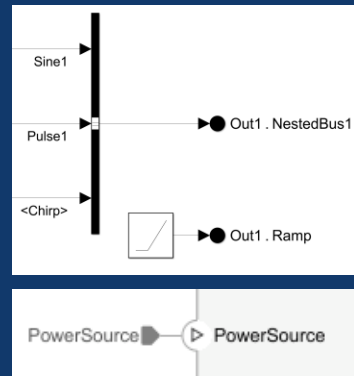
# ① Automate everything



# 2 Scale to complex systems



Components



Buses, Ports,  
and Connectors

VariantManagement	
Controller	
Linear Controller	Ctrl == Controller
Nonlinear Controller	Ctrl == Controller
Smart Controller	Ctrl == Controller
Sensor Modifier	
AI Sensor (...)	SmartSensorMod
FL Sensor (...)	SmartSensorMod
Plant	
External (Model file: s...)	PlantLoc == Plant
Internal	PlantLoc == Plant
Experimental	SimType==Intern

Variant Manager

Block Parameters: vehic\_AOB

vehic\_AOB [Model Exchange, v1.0]

FMU Block

FMU with bus signals and structured parameters

Open FMU Documentation File

Parameters Simulation Input Output

Model Exchange settings

Enable FMU tolerance Relative t

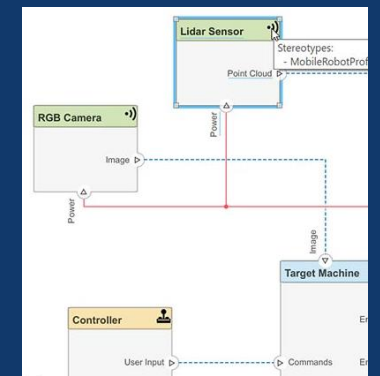
Block sample time (-1 for inherited): 0

Debugging

[Open FMU Working Directory...](#) Open

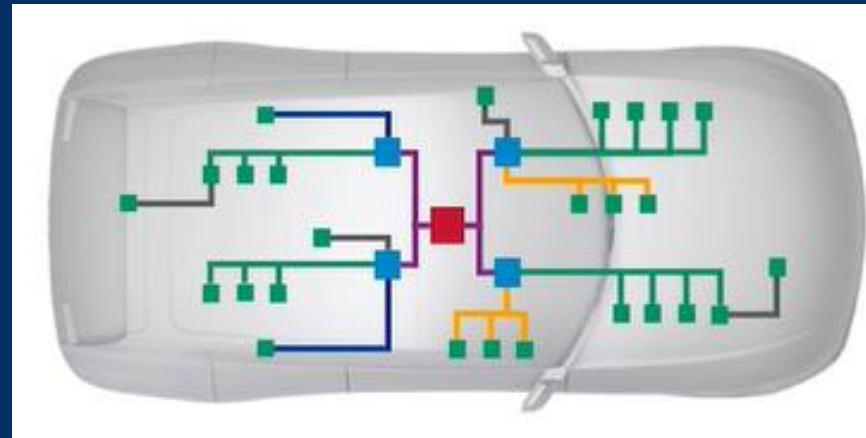
Enable FMU debug logging Redir

Third-Party Tool  
Integration

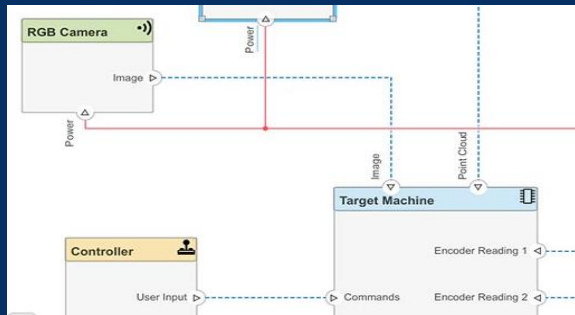


Architecture

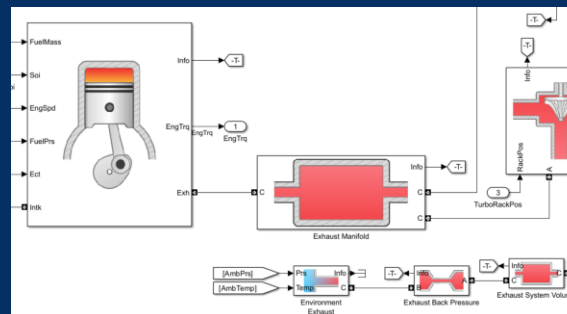
# 2 Scale to complex systems



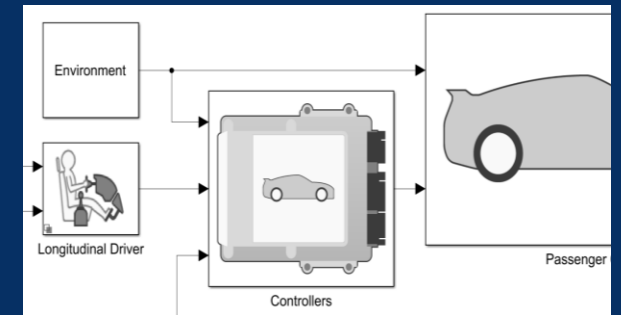
## Service-Oriented Architecture



System Composer



Components



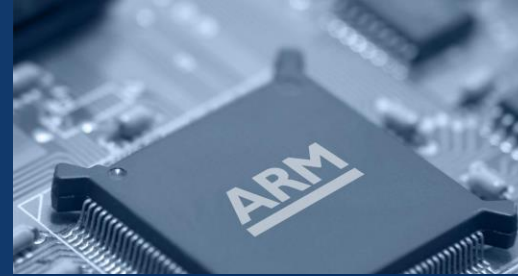
Full System

# ③ Use automatic code generation

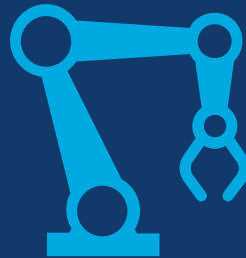


**3700**

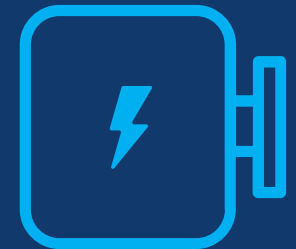
Organizations use  
automatic code generation



CPU



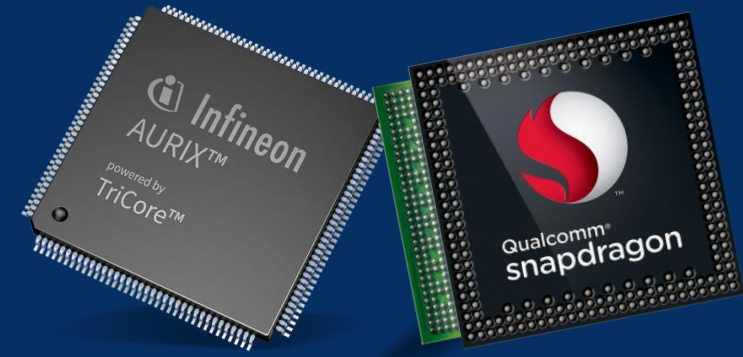
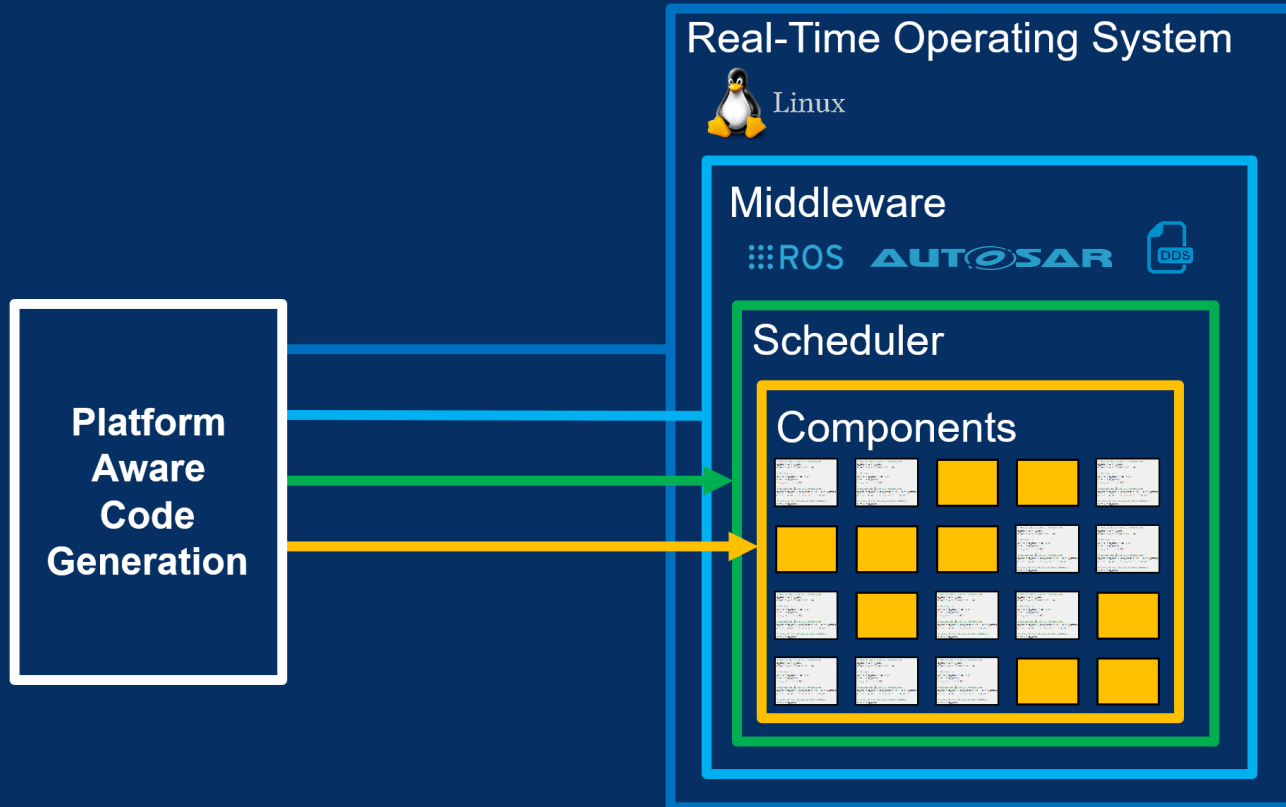
GPU



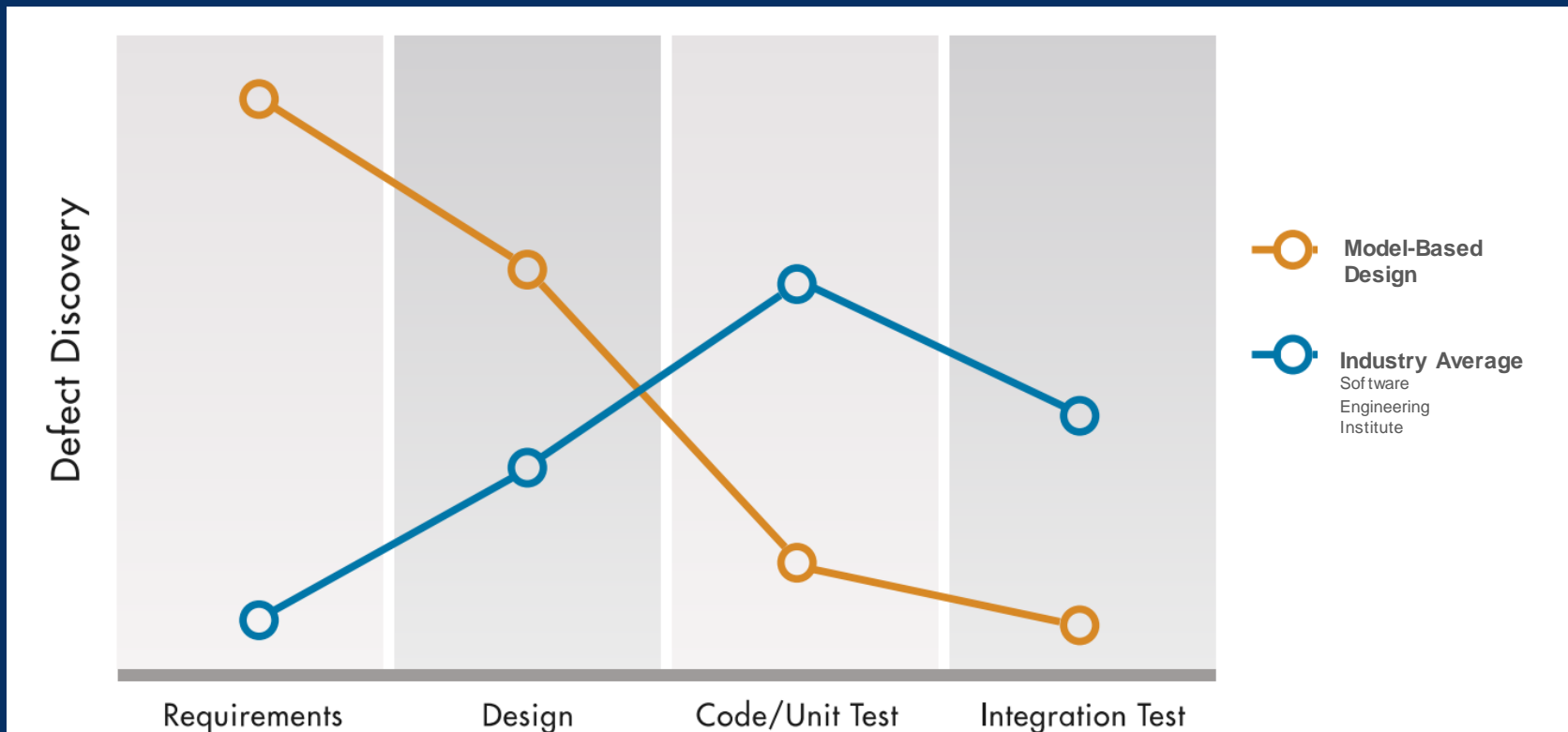
FPGA, ASIC, PLC



# 3 Use automatic code generation



# 4 Prevent defects early



# 4 Prevent defects early



**Find Defects Sooner**

<b>Design</b>	<b>Test</b>	<b>Code</b>	<b>Certify</b>
Simulink Design Verifier	Simulink Test	Polyspace Bug Finder	DO Qualification Kit
Simulink Check	Simulink Coverage	Polyspace Code Prover	IEC Certification Kit
HDL Verifier	MATLAB Test	Polyspace Access	Simulink Code Inspector
Simulink Fault Analyzer	Polyspace Test		

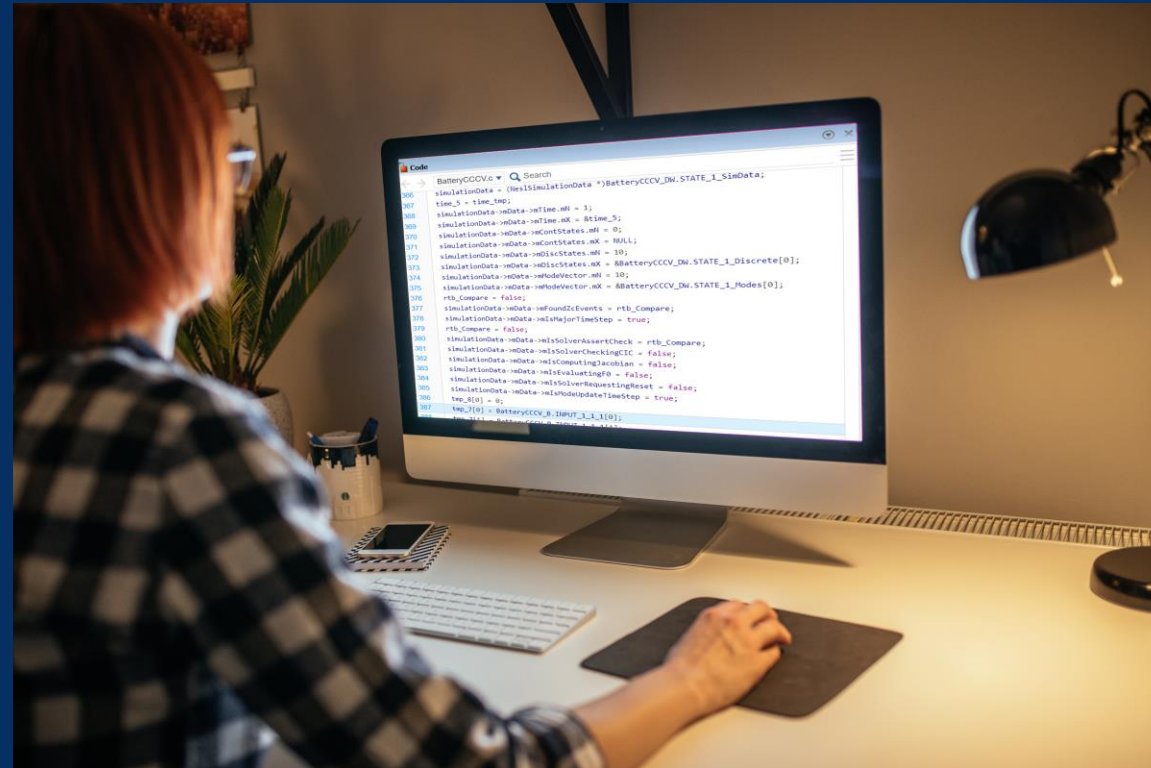




# UL Certification of Battery Management System Software with Model-Based Design



The Saft Flex'ion Gen2

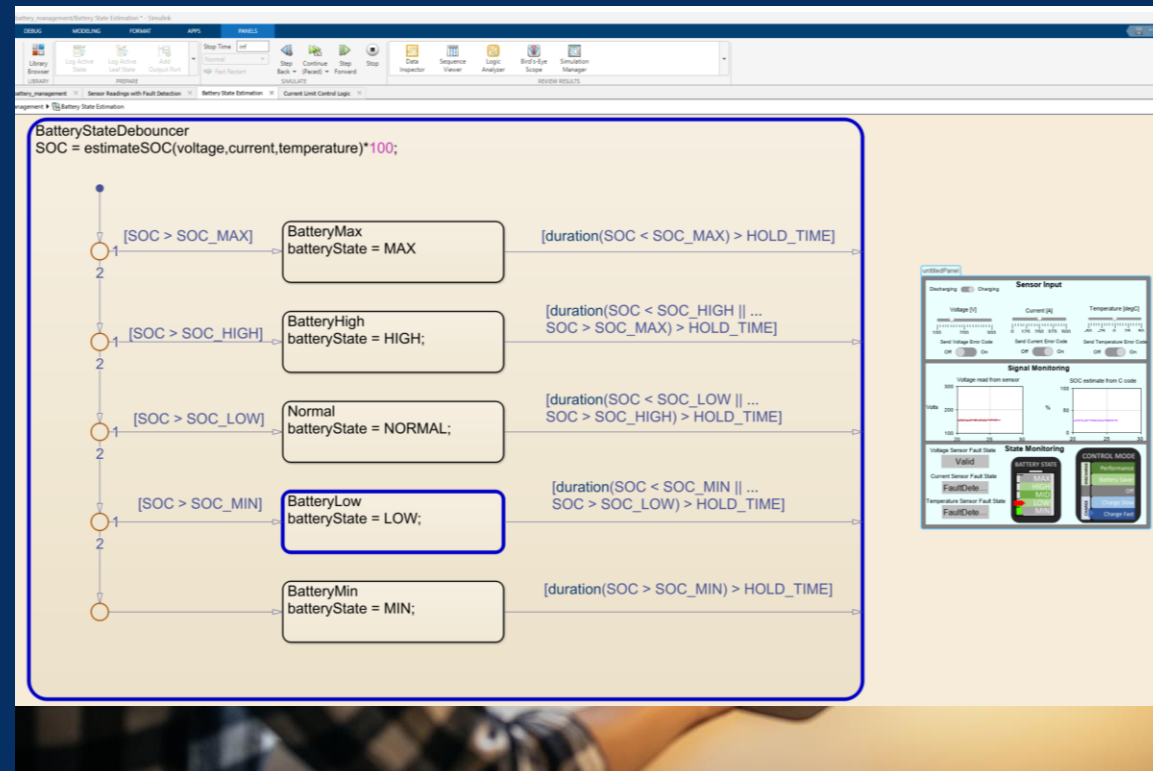




# UL Certification of Battery Management System Software with Model-Based Design



The Saft Flex'ion Gen2





# UL Certification of Battery Management System Software with Model-Based Design



The Saft Flex'ion Gen2



# Application Trends



Autonomous

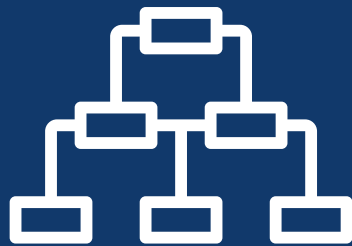


Connectivity



Electrification

# Workflow Trends



Systems Engineering  
& Design

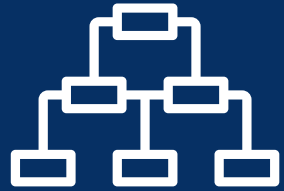


Modern  
Software Practices



AI for  
System Development

# Workflow Trends



1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early



5. Apply standard software workflows
6. Design and simulate in the cloud



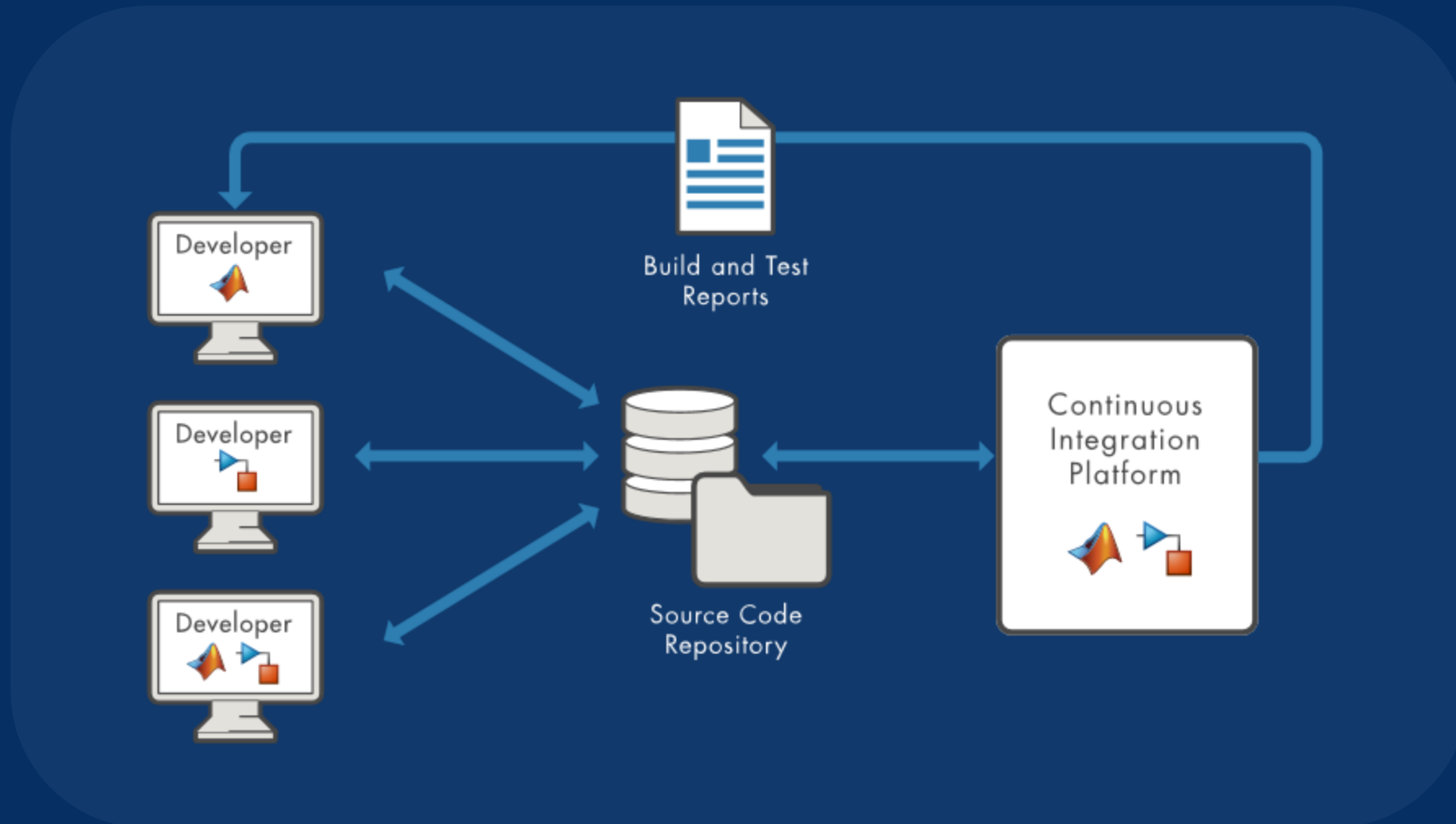
7. Design your system with AI

# 5 Apply standard software workflows



“Software is the language of automation.”  
- *Jensen Huang, co-founder and CEO of NVIDIA*

# 5 Apply standard software workflows



# Model Based Design 확대



## SW 개발 시간 단축

✓ 시뮬레이션 가능한 모델 활용

## Seamless SW 개발 환경

✓ SW 개발 난이도 감소 및 품질 향상

## 모델 기반 Variant 관리

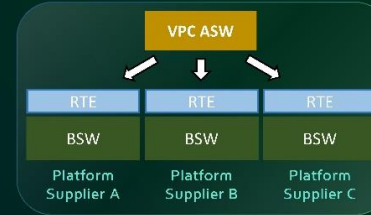
✓ 모델 기반으로 다양한 사양 대응



## ✓ 기능안전/A-SPICE 강화



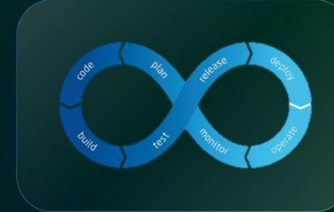
## ✓ AUTOSAR 확대



## ✓ 가상화 & 자동화 검증 확대



## ✓ DevOps 도입



**HYUNDAI**  
MOTOR GROUP

- Vehicle Platform Control (VPC) – 1<sup>st</sup> domain-based controller
- Utilized MBD to develop two SoCs
- Common MBD development platform with Tier 1s





MATLAB EXPO

New system ... How to ?

**Legacy**

Verified, Validated, Stable, ... High inertia

Easy starting Limit

Easy starting Limit

**New system**

Large, Complex, Long lifetime

**MBD Process**

Fast prototype, Efficient Co-working

New setup Acceleration

Fast Prototype Acceleration

16



한상설 팀장

국방과학연구소



MATLAB EXPO  
KOREA



국방과학연구소  
Agency for Defense Development

- ADA's process shift from Legacy Workflows to MBD leads to accelerated development and improved reliability
- MBD impact seen in technology-intensive Korea space sector

# 5 Apply standard software workflows



## Technical Articles and Newsletters

Overview | Search Technical Articles | Newsletters ▾ | Cleve's Corner Collection

### Workflow Steps

The workflow consists of the following steps (Figure 4):

1. **Trigger** a pipeline in GitLab and observe that the Verify and Build stages
2. **Detect** a test-case failure in GitLab CI pipeline and create an Issue to tra
3. **Reproduce** the issue on our desktop MATLAB.
4. **Fix the issue** in the model.
5. **Test locally** to ensure the test case passes.
6. **Review** the changes on the testing branch.
7. **Commit** the change to Git and trigger the CI pipeline in GitLab.

## Step-by-Step Tutorials

The screenshot shows the MATLAB R2023a interface with the Process Advisor tool open. The Process Advisor window displays a list of tasks for the 'Flight\_Control' model. The tasks and their status are as follows:

Tasks	Out	Details
Generate Simulink Web View	✓	1
Check Modeling Standards	✓	3 △ 1
Detect Design Errors	✓	1
Generate SDD Report	✓	1
Generate Code (Top)	✓	1

The Process Advisor window also shows a 'Run All' button and a 'Model' dropdown menu. The background shows the MATLAB interface with the 'MODELING' tab selected and the 'Process Advisor' icon highlighted in a red box. The right side of the interface shows a block diagram with blocks labeled 'PilotPitchCmd' and 'PilotRollCmd'.

## Process Advisor



**Jenkins**



**git**



**MATLAB® & SIMULINK®**

**AUTOSAR**



“Bring everything into MATLAB.”

- *Martin Römpert, Continental Automotive Technologies GmbH*



**Process Advisor**

PROCESS ADVISOR

Edit Run All Incremental Build

PROJECT BUILD SETTINGS

Tasks	Out	Details
<ul style="list-style-type: none"> <li>Check Modeling Standards                             <ul style="list-style-type: none"> <li>db_ControlMode.slx (99+ ✓, 9 △)</li> <li>db_Controller.slx (30 ✓, 2 △)</li> <li>db_DriverSwRequest.slx (29 ✓, 3 △)</li> <li>db_TargetSpeedThrottle.slx (30 ✓, 2 △)</li> </ul> </li> <li>Generate Code (Top)                             <ul style="list-style-type: none"> <li>db_Controller.slx (1 ✓, 16 △)</li> </ul> </li> <li>Generate Code (Ref)                             <ul style="list-style-type: none"> <li>db_TargetSpeedThrottle.slx (1 ✓)</li> <li>db_ControlMode.slx (1 ✓)</li> <li>db_DriverSwRequest.slx (1 ✓)</li> </ul> </li> <li>Run Tests (6 ✗)                             <ul style="list-style-type: none"> <li>db_ControlMode.slx (1 ✓)</li> <li>ControlModeDisabled (1 ✓)</li> <li>db_DriverSwRequest.slx (6 ✗)                                     <ul style="list-style-type: none"> <li>Cancel button</li> <li>Decrement button hold</li> <li>Decrement button short (1 ✗)</li> <li>Enable button (1 ✗)</li> <li>Increment button hold (1 ✗)</li> <li>Increment button short (1 ✗)</li> <li>Resume button (1 ✗)</li> <li>Set button (1 ✗)</li> </ul> </li> </ul> </li> <li>Merge Test Results                             <ul style="list-style-type: none"> <li>db_ControlMode.slx (1 ✓)</li> <li>db_DriverSwRequest.slx (1 ✓)</li> </ul> </li> </ul>		

Details

Project Path Startup Shutdown Environment

Git Refresh Commit Fetch Push Pull Remote Branches Stashes Submodules

including requirements based unit tests

All Project (67) Modified (0)

Name	Status	Git	Review Status	Owner	Classification
.github	✓	·			
CI	✓	·			
data	✓	·			
generated_reports	✓	·			
models	✓	·			
requirements	✓	·			
scripts	✓	·			
.gitattributes	✓	·			
.gitignore	✓	·			
.gitlab-ci.yml	✓	·			
azure-pipelines.yml	✓	·			
azure-pipelines-docker...	✓	·			
Jenkinsfile	✓	·			
README.md	✓	·			
runCI.m	✓	·			



Tasks	Out	Details
Generate Simulink Web View		✓ 1
Check Modeling Standards		✓ 3  1
Detect Design Errors		✓ 1
Generate SDD Report		✓ 1
Generate Code (Top)		✓ 1

Process Advisor

- Utilize the digital thread to run only what you need
- Identify stale tests
- Interact with the model

2X

# 6 Design and simulate in the cloud



A screenshot of a web browser displaying the MATLAB Online login page. The browser's address bar shows 'matlab.mathworks.com'. The page features the MathWorks logo and 'MATLAB Online' text. A large 'MATLAB Online' heading is centered. Below it is the MathWorks logo, the word 'Email', and a text input field containing 'mcarone@mathworks.com'. There are links for 'No account? Create one!' and 'By signing in you agree to our privacy policy.'. A blue 'Next' button is positioned to the right of the input field, with a mouse cursor hovering over it. At the bottom, there are links for 'Learn about MATLAB Online' and 'Use MATLAB Drive™ to synchronize your MATLAB files'.

# 6 Design and simulate in the cloud



```
for i = 1:10000
    in(i) = Simulink.SimulationInput(my_model)
    in(i) = setVariable(my_var, i);
end
out = parsim(in);
```

## Massive simulations



Parallel Computing Toolbox



MATLAB Parallel Server





# ⑥ Design and simulate in the cloud



Example: Integrating four widely used toolchains



# 6 Design and simulate in the cloud



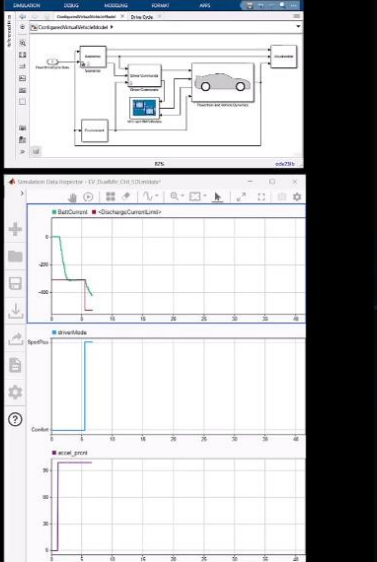
## Automotive Software Development in the Cloud

1

powered by **aws** FOR AUTOMOTIVE

**MathWorks**

Virtual Vehicle Simulation in MATLAB & Simulink




Application Code from Model

**Elektrobit**

Road-ready Automotive Software

Infotainment (Android Automotive OS)



Vehicle Control Unit (Adaptive AUTOSAR)


[285.18] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[286.18] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[287.19] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[288.20] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[289.20] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[210.21] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[211.21] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[212.22] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[213.23] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[214.23] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[215.24] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[216.24] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[217.25] Adaptive AUTOSAR HPC - received discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75

Battery Management System (Classic AUTOSAR)

[281.02] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[281.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[282.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[283.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[284.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[285.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[286.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[287.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[288.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[289.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[290.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[291.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75
[292.62] Classic AUTOSAR ECU - sending discharge current limit: -310.00	charge current limit: 102.30	charge: 0.75

**synopsys**

Virtual ECUs in Synopsys Silver



ECU Simulation

## Application Trends



Autonomous

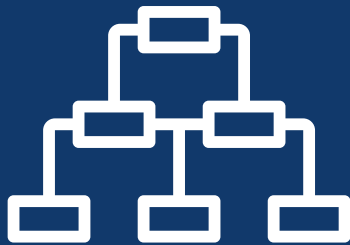


Connectivity



Electrification

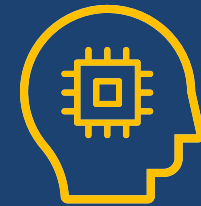
## Workflow Trends



Systems Engineering  
& Design

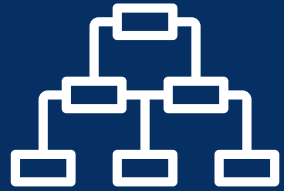


Modern  
Software Practices



AI for  
System Development

# Workflow Trends



1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early



5. Apply standard software workflows
6. Design and simulate in the cloud

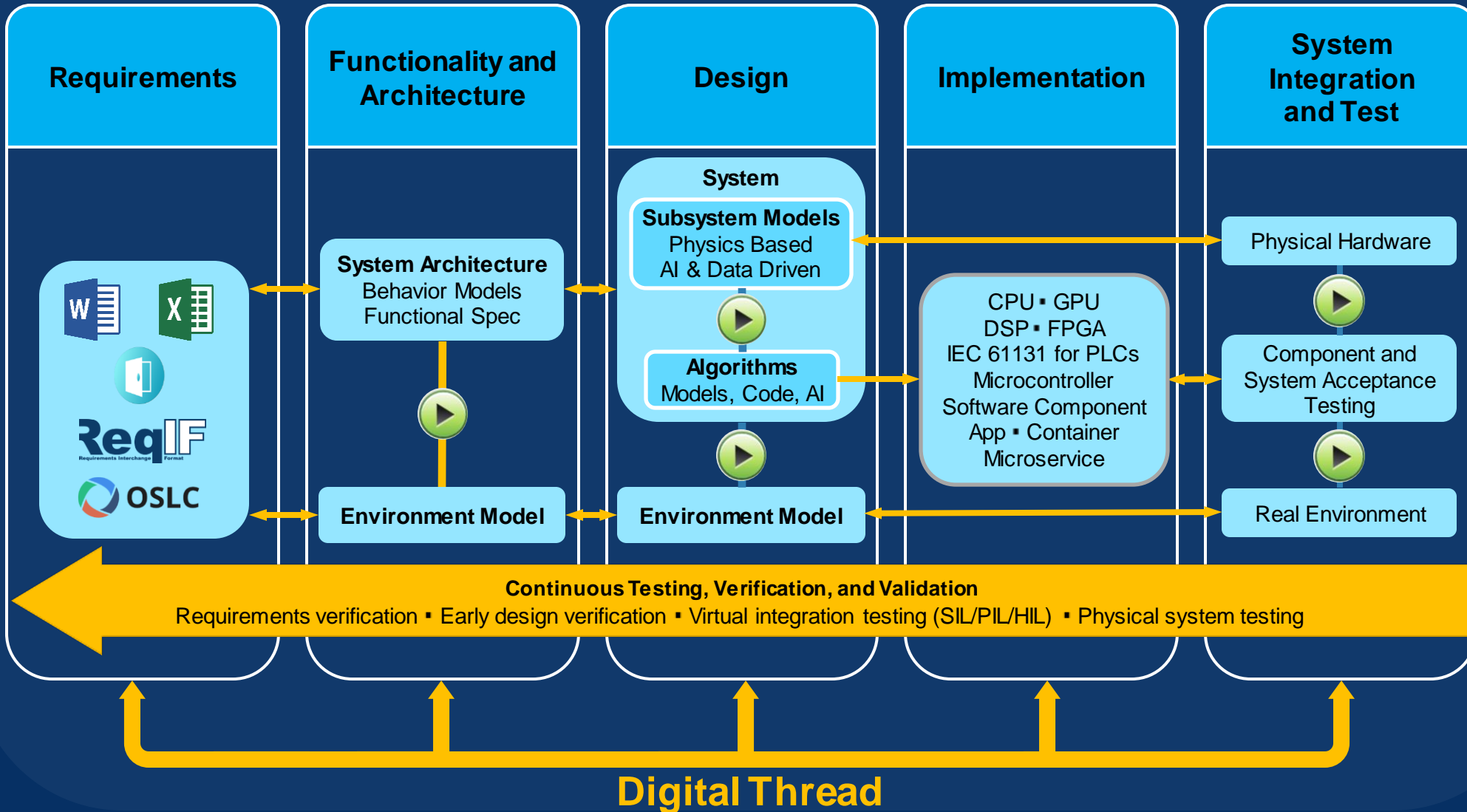


7. Design your system with AI

# 7 Design your system with AI



## Integrating AI into Model-Based Design



# 7 Design your system with AI



AI

**Find out more:**

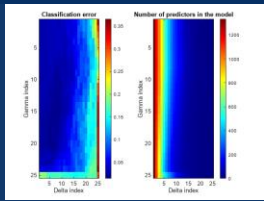
**AI Playground in the Demo Showcase**

See the latest advances in AI and MATLAB.

# 7 Design your system with AI



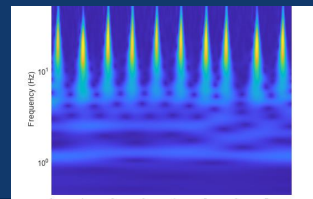
## AI Reference Examples



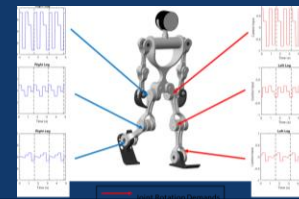
Predictive Maintenance



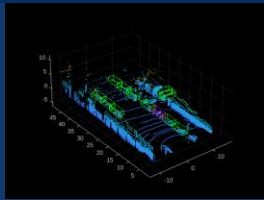
Hyperspectral Imaging



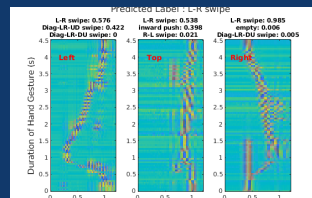
Signal Processing



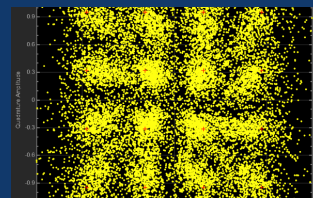
Robotic Control



Lidar Processing



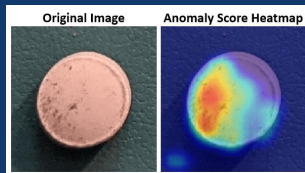
Radar Processing



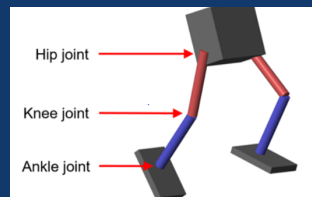
Wireless Communications



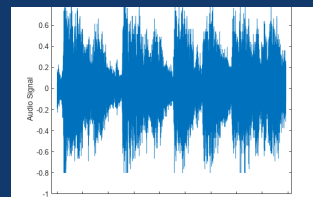
Automated Driving



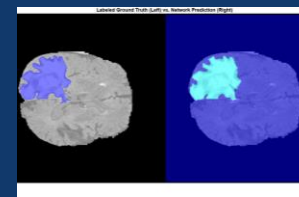
Visual Inspection



Reinforcement Learning



Audio

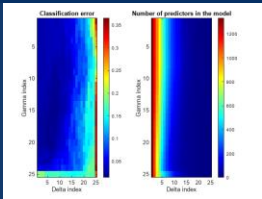


Medical Imaging

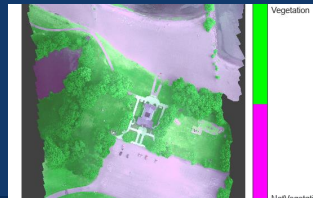
# 7 Design your system with AI



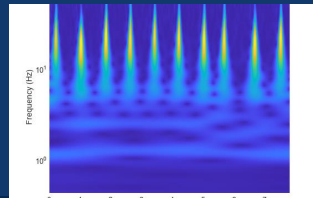
## AI Reference Examples



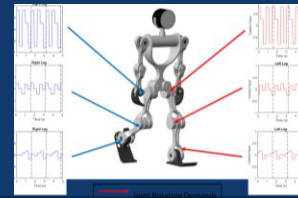
Predictive Maintenance



Hyperspectral Imaging



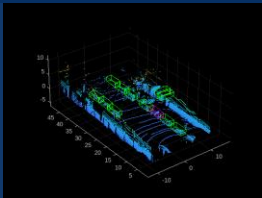
Signal Processing



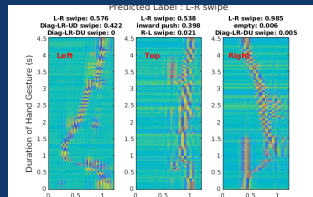
Robotic Control



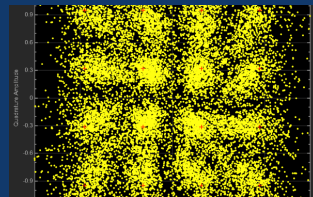
CPU



Lidar Processing



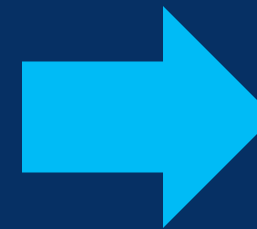
Radar Processing



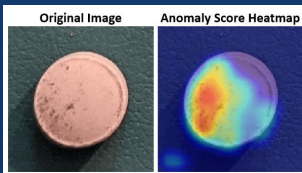
Wireless Communications



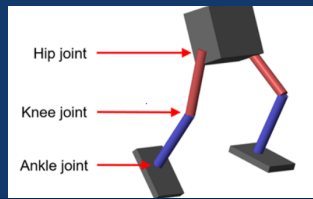
Automated Driving



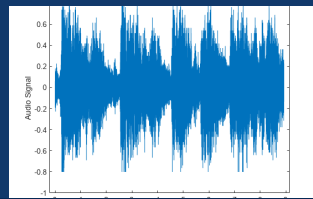
GPU



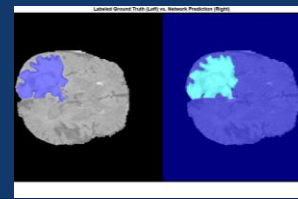
Visual Inspection



Reinforcement Learning



Audio



Medical Imaging



FPGA, ASIC, PLC





Mercedes-Benz

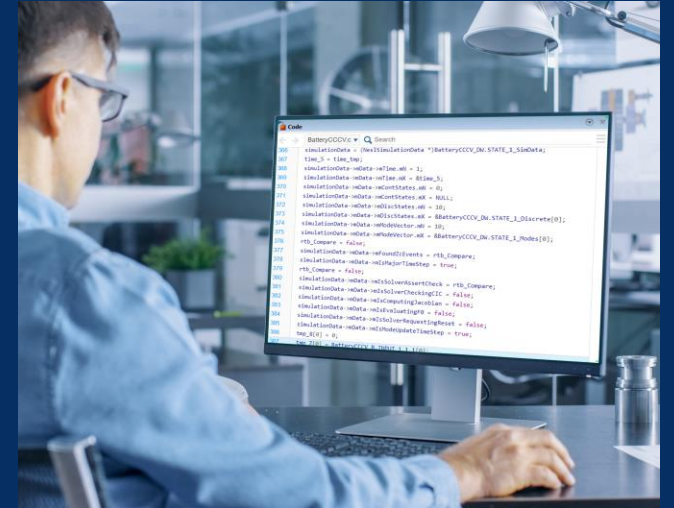
# Simulates Hardware Sensors with Deep Neural Networks





Mercedes-Benz

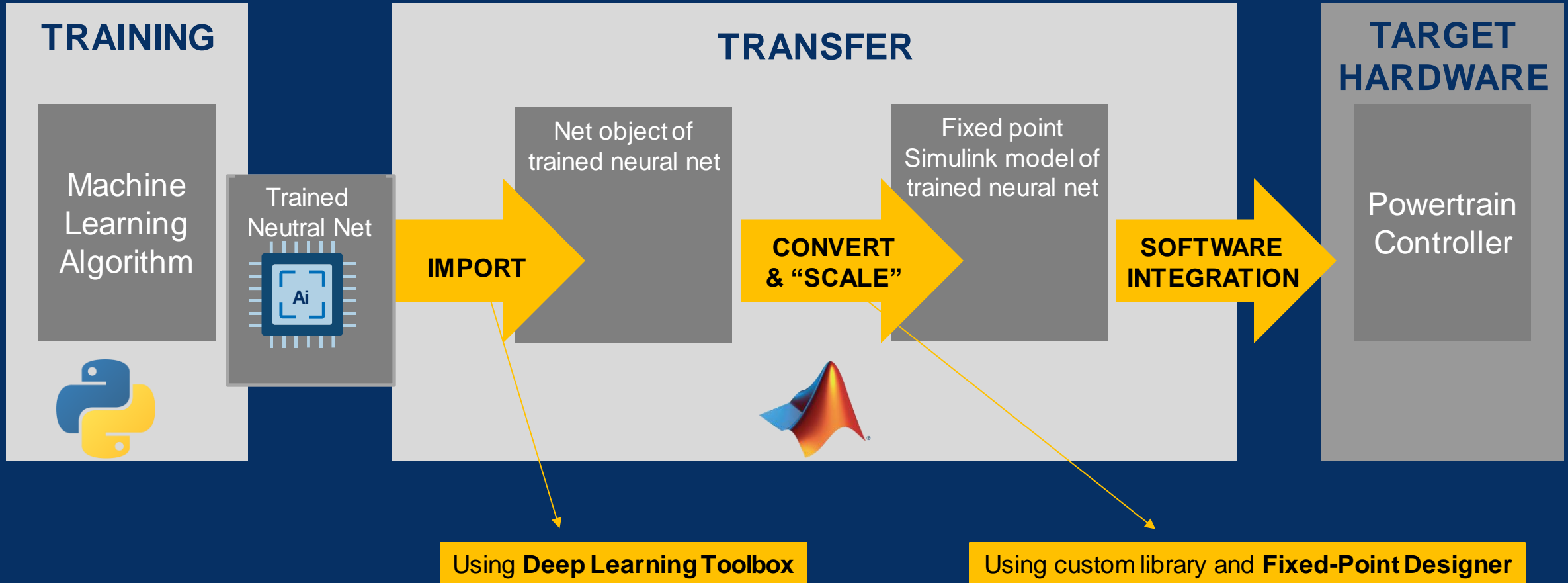
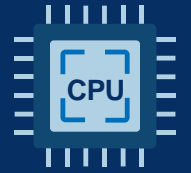
# Simulates Hardware Sensors with Deep Neural Networks





Mercedes-Benz

# Simulates Hardware Sensors with Deep Neural Networks





Mercedes-Benz

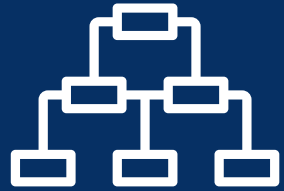
## Simulates Hardware Sensors with Deep Neural Networks



*“We are already using the **automated workflow** we created with MATLAB and Simulink for other use cases ... small adaptations to support deployment on two different powertrain controllers, and the workflow is also applicable to **other types of deep learning models** such as gated recurrent units and fully connected neural networks ... we **committed fewer errors** in creating the model and the code.”*

*- Katja Deuschl, AI Developer, Mercedes-Benz*

# Workflow Trends



1. Automate everything
2. Scale to complex systems
3. Use automatic code generation
4. Prevent defects early



5. Apply standard software workflows
6. Design and simulate in the cloud



7. Design your system with AI

# Application Trends



**Autonomous**



**Connectivity**



**Electrification**

## Workflow Trends



**Systems Engineering  
& Design**



**Modern  
Software Practices**

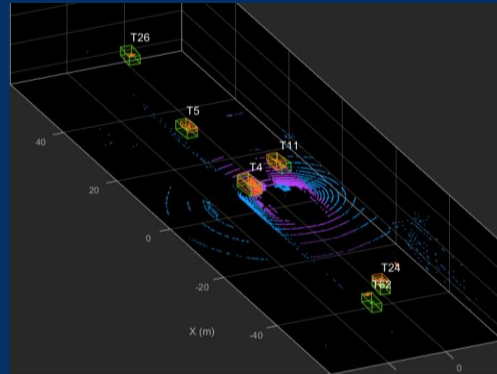


**AI for  
System Development**

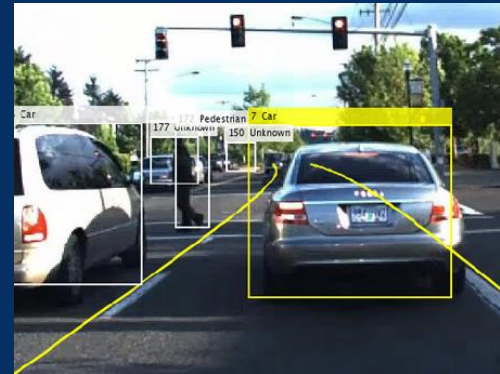
# Deliver autonomous systems



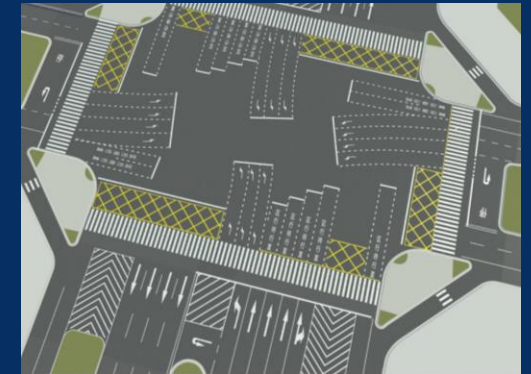
**Braking and Steering**



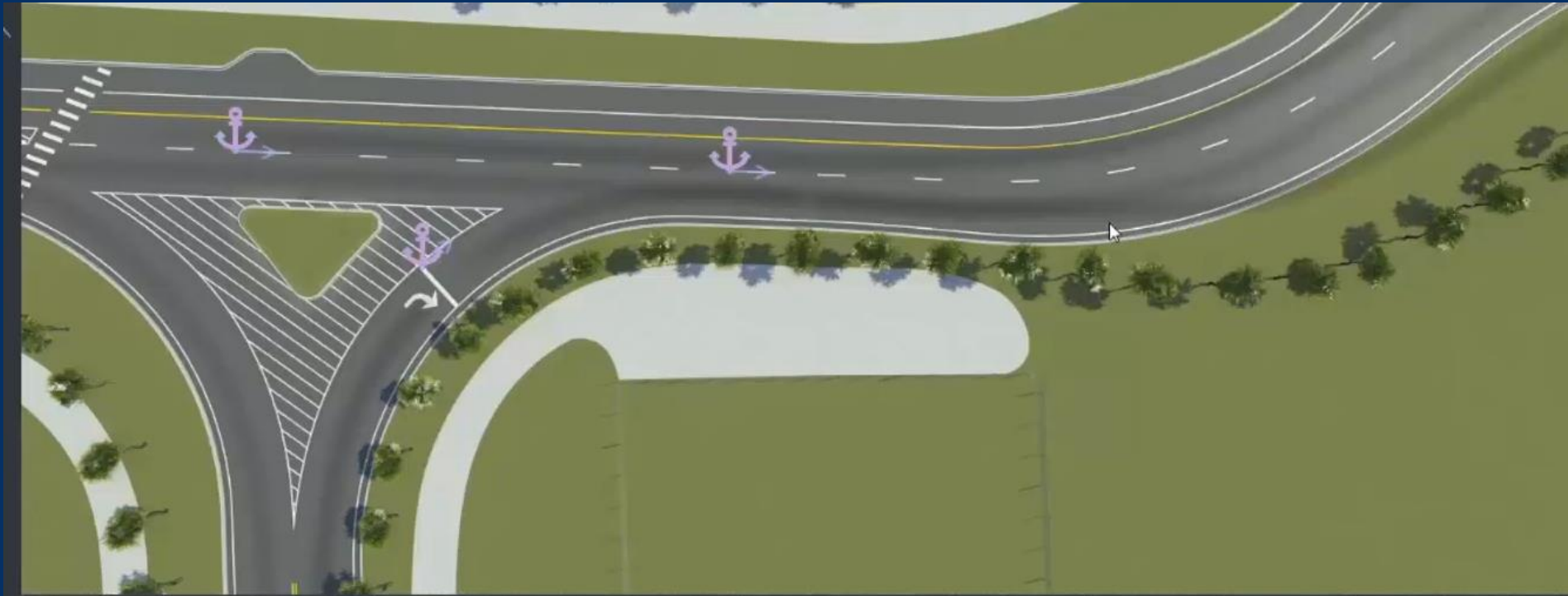
**Sensor Fusion and Tracking**



**Computer Vision  
Radar, Lidar**



**Road Network Design**







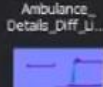



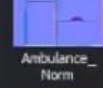







2D Editor | Logic

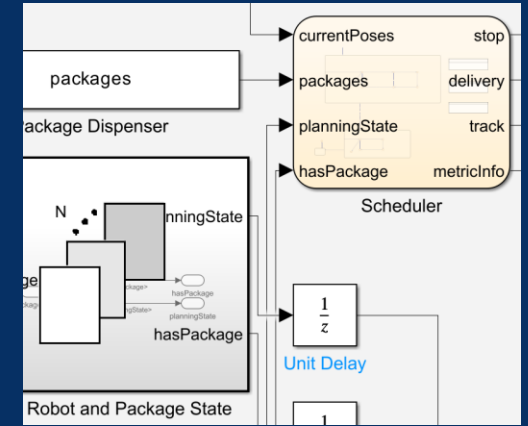
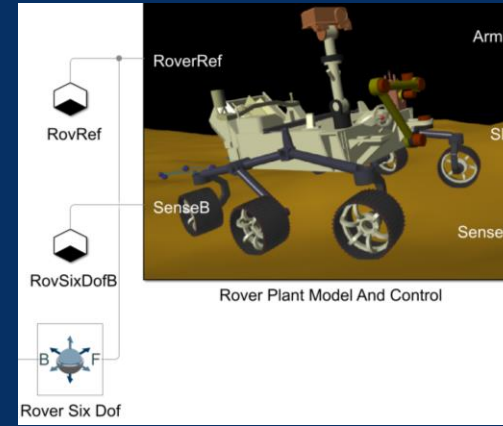
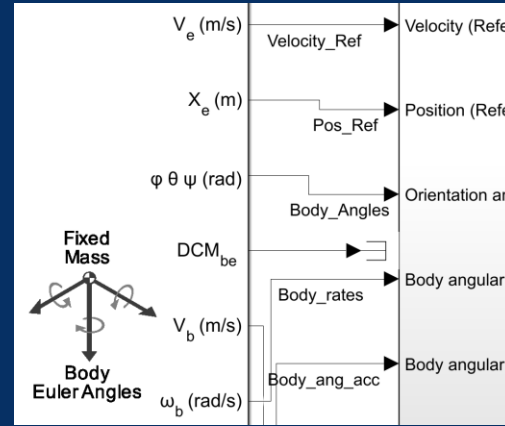
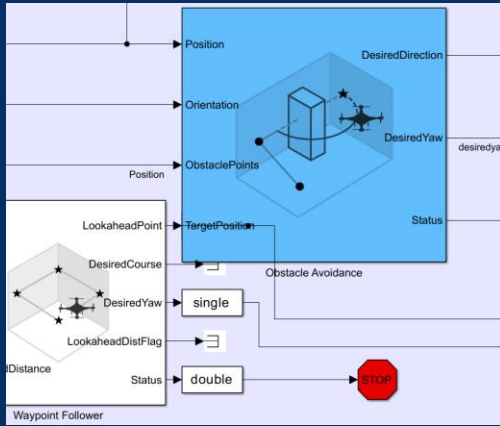
Library Browser

- Assets
  - Assemblies
  - Behaviors
  - Buildings
  - Damage
  - Extrusions
  - Markings
  - Materials
  - Posts
  - Props
  - Rail
  - RoadStyles
  - Signs
  - Stencils
  - Test
  - Vehicles

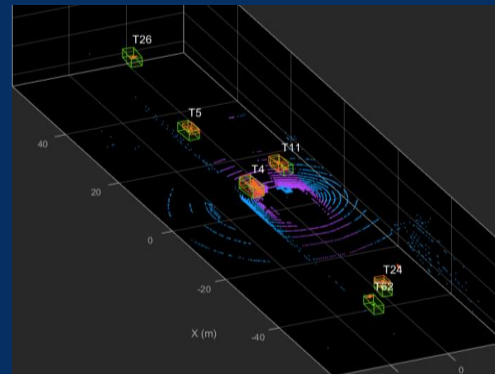
A logic editor interface with a purple rounded rectangle and a blue clock icon connected by lines.

 Ambulance	 Ambulance_Details_Diff	 Ambulance_Details_Diff_U...	 Ambulance_Details_Norm
 Ambulance_Details_Spec	 Ambulance_Diff	 Ambulance_Norm	 Ambulance_Spec
 CementTruck	 CementTruck_Diff	 CementTruck_Norm	 CementTruck_Spec
 CementTruck	 CementTruck	 CementTruck	 CementTruck





**Unmanned Aerial Vehicle**



**Autonomous Underwater Vehicle**



**Ground Robot**



**Industrial Robot**

# Application Trends



**Autonomous**



**Connectivity**



**Electrification**

## Workflow Trends



**Systems Engineering  
& Design**



**Modern  
Software Practices**



**AI for  
System Development**

# Application Trends



**Autonomous**



**Connectivity**



**Electrification**

# Workflow Trends



**Systems Engineering  
& Design**



**Modern  
Software Practices**



**AI for  
System Development**

# 5G Standard



ects  
atasets  
d that  
neering,

# 6G Technology Implications



**RCR Wireless News**  
INTELLIGENCE ON ALL THINGS WIRELESS

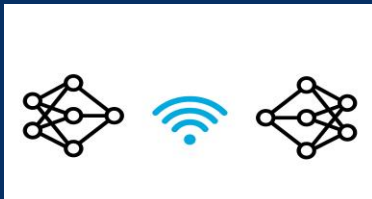
| China aims for 6G commercialization by 2030: Report

By **Juan Pedro Tomás** December 13, 2023

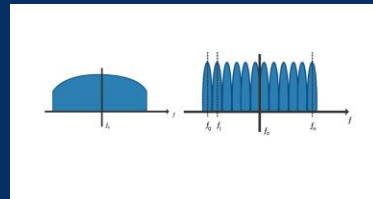
6G Standards

# 6G being designed now

## Key Technologies



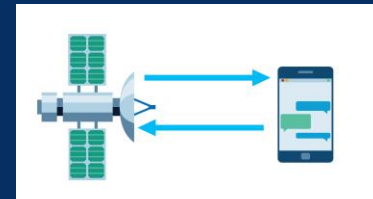
Artificial intelligence  
and machine learning



Spectrum bands



Network that can  
sense



Extreme connectivity



New network  
architectures

## MathWorks Products

Communications,  
5G, WLAN Toolboxes,  
w/ Deep Learning Toolbox

RF Blockset,  
Antenna Toolbox

WLAN Toolbox,  
5G Toolbox,  
Radar Toolbox

Satellite Communications  
Toolbox







6G Exploration Library (R2024a)



# Wireless Trends – AI in Wireless






## Wireless challenges

-  Hard-to-model problems
-  Computational infeasibility of optimal solution
-  Efficient modem parameter optimization
-  Dealing with non-linearity



AI-enhanced  
wireless communications

## AI strengths

-  Determining appropriate representations for hard-to-model problems
-  Finding near-ideal and computationally realizable solutions
-  Modeling non-linear functions

## Applying AI to solve difficult wireless challenges

Deep wireless domain knowledge is required to optimally use AI capabilities

# Application Trends



**Autonomous**



**Connectivity**



**Electrification**

# Workflow Trends



**Systems Engineering  
& Design**



**Modern  
Software Practices**



**AI for  
System Development**



# Application Trends



**Autonomous**



**Connectivity**



**Electrification**

# Workflow Trends



**Systems Engineering  
& Design**

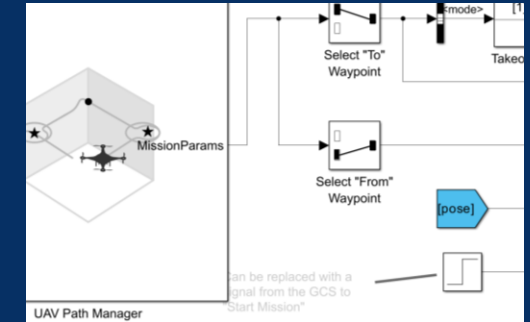
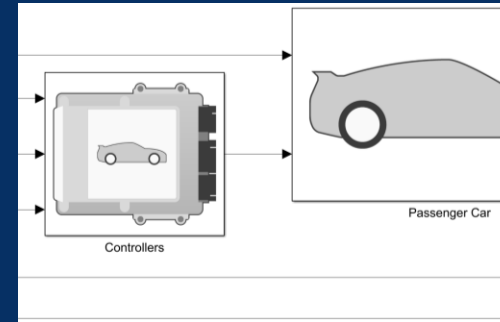
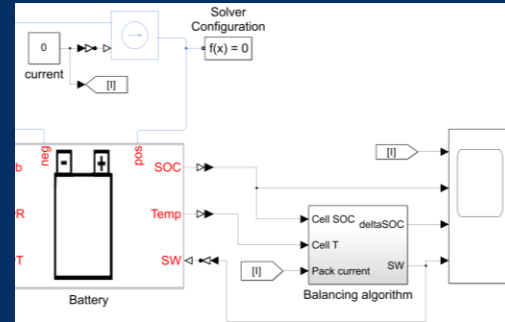
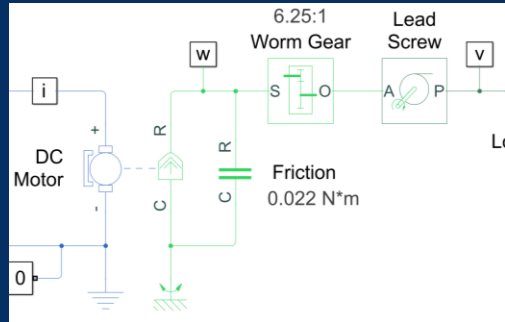


**Modern  
Software Practices**



**AI for  
System Development**

# Electric Vehicles



**Electric Motors**



**Battery Packs**

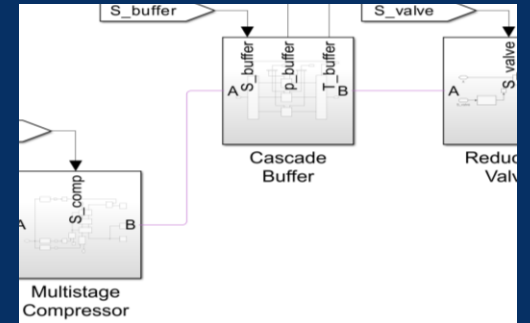
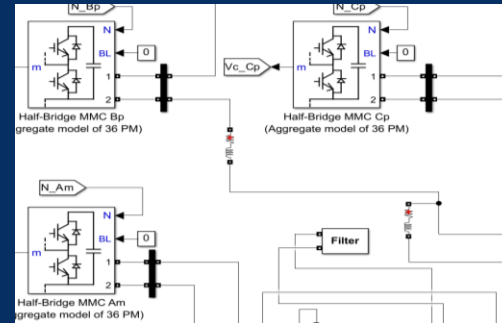
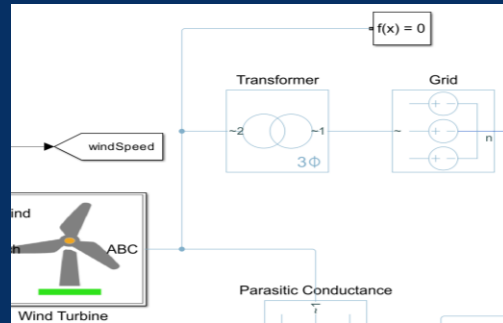
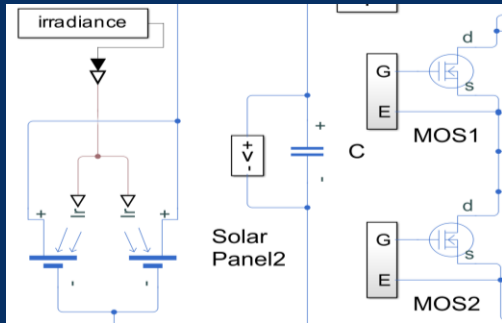


**Full Vehicle Models**



**Aerial Vehicles**

# Green Energy



Solar



Wind

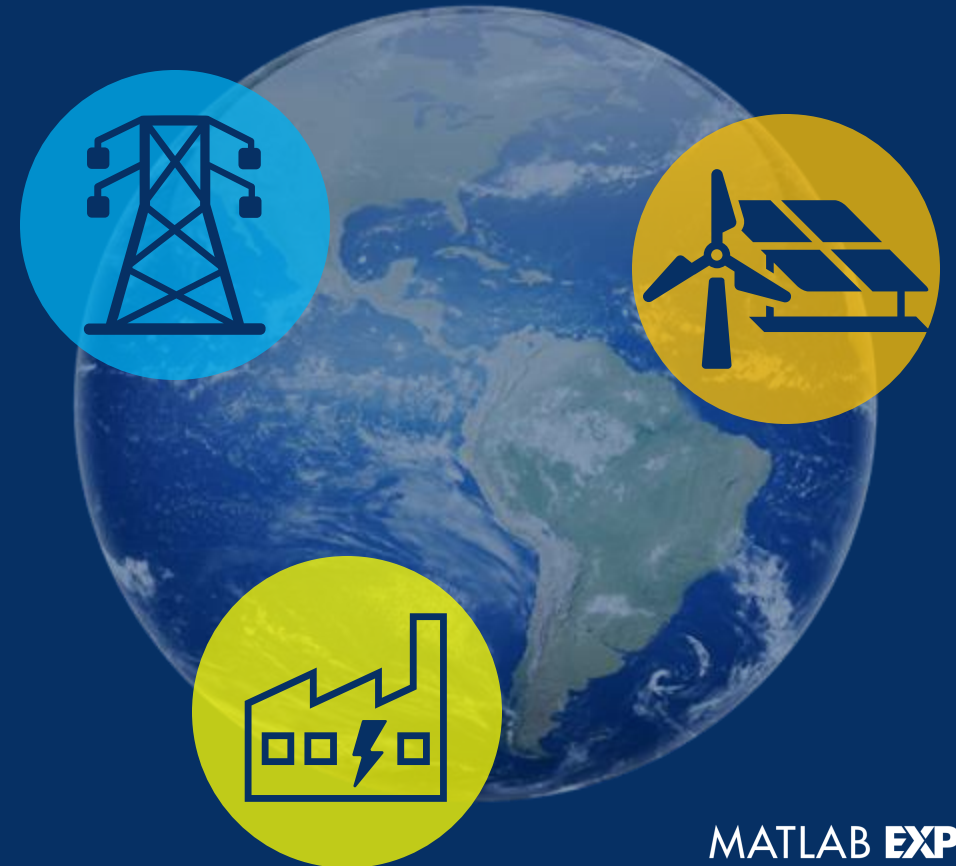


Hydroelectric



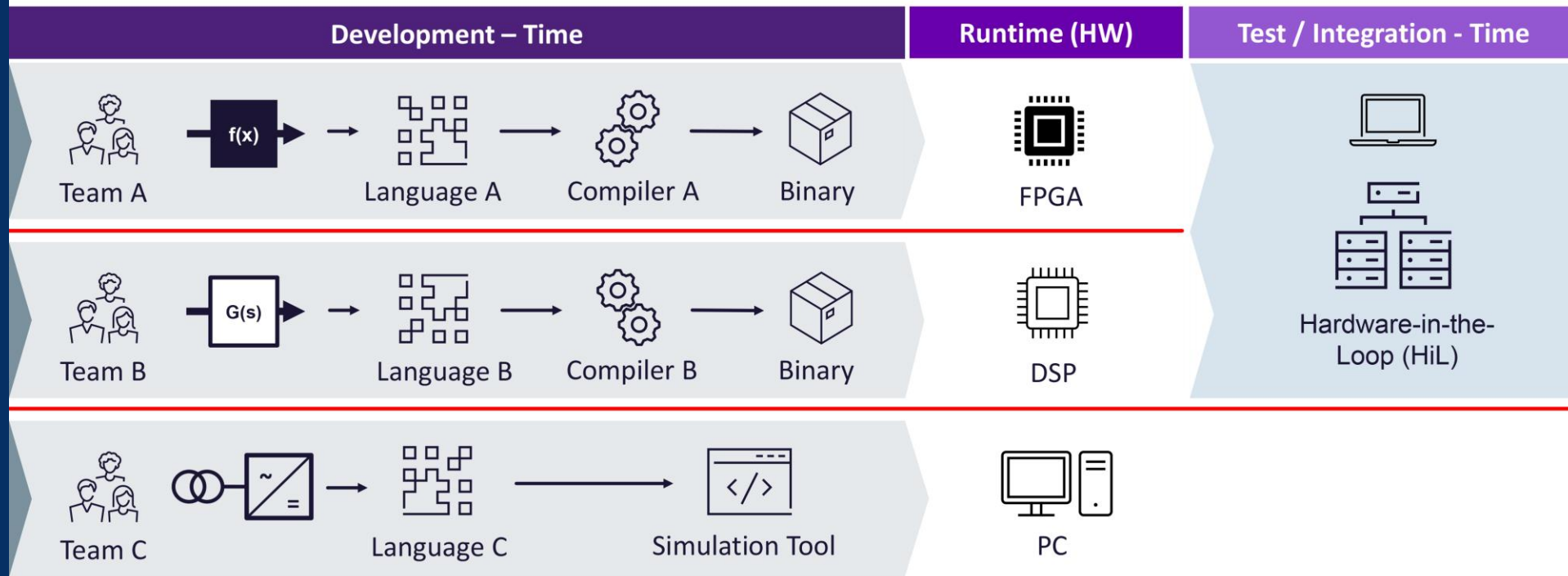
Green Hydrogen

1. Engineer solutions in solar, biomass, hydrogen, wind
2. Retrofit or upgrade infrastructure
3. Strengthen electrical grid



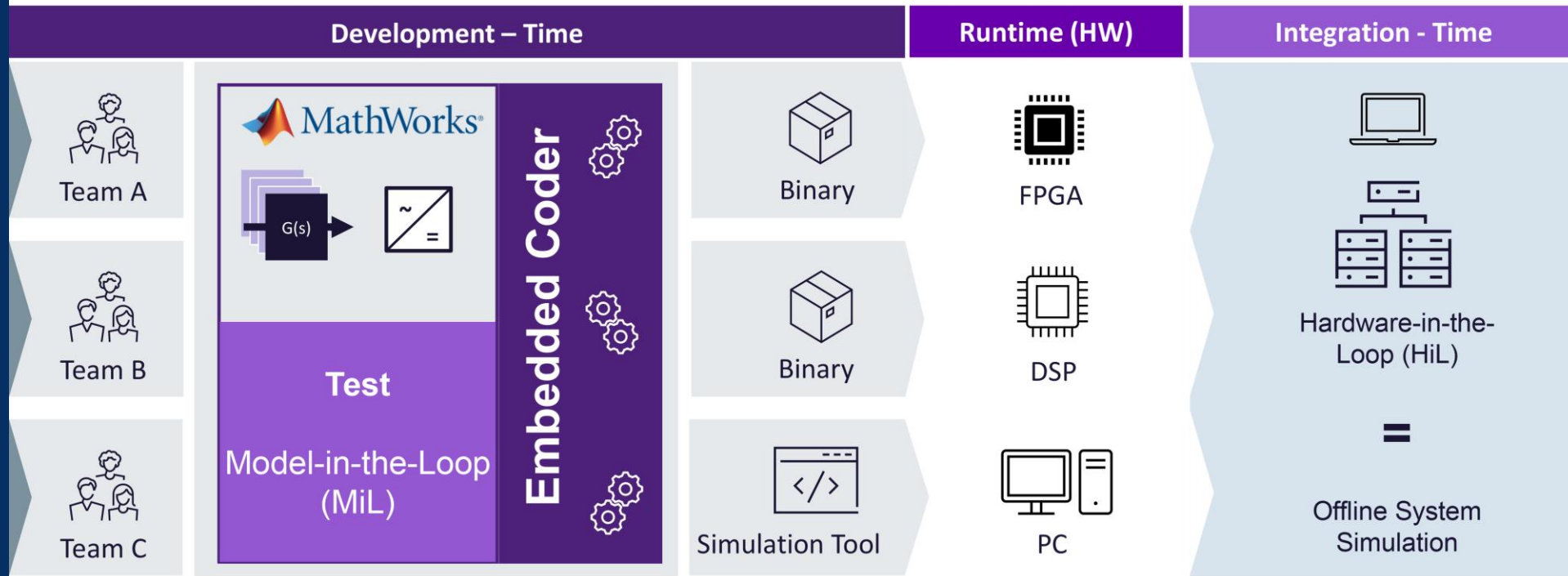
### Swimlane Engineering

When the organization shapes development

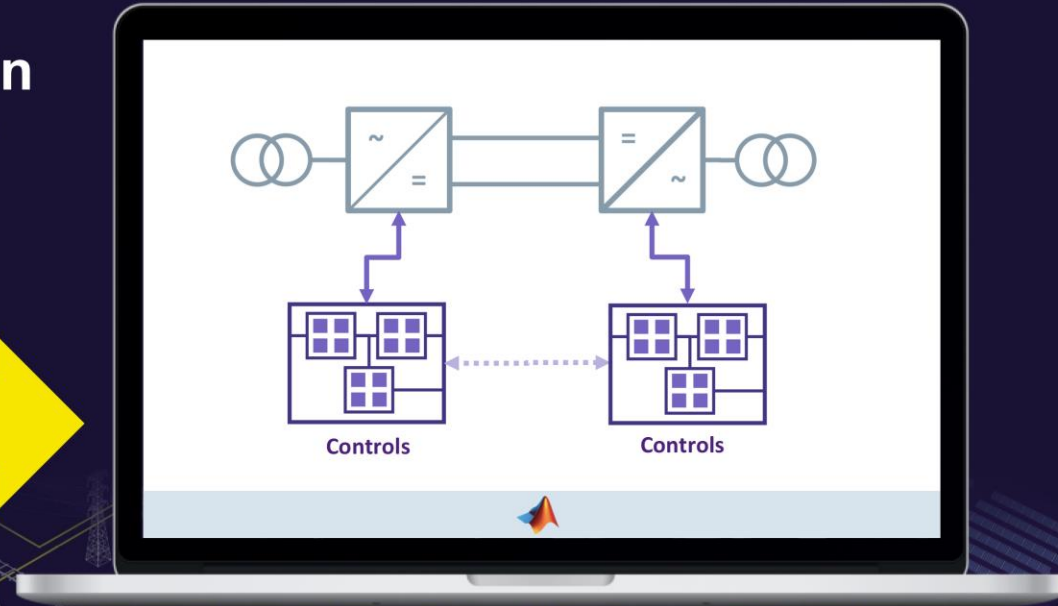


### Centralized Engineering Ecosystem

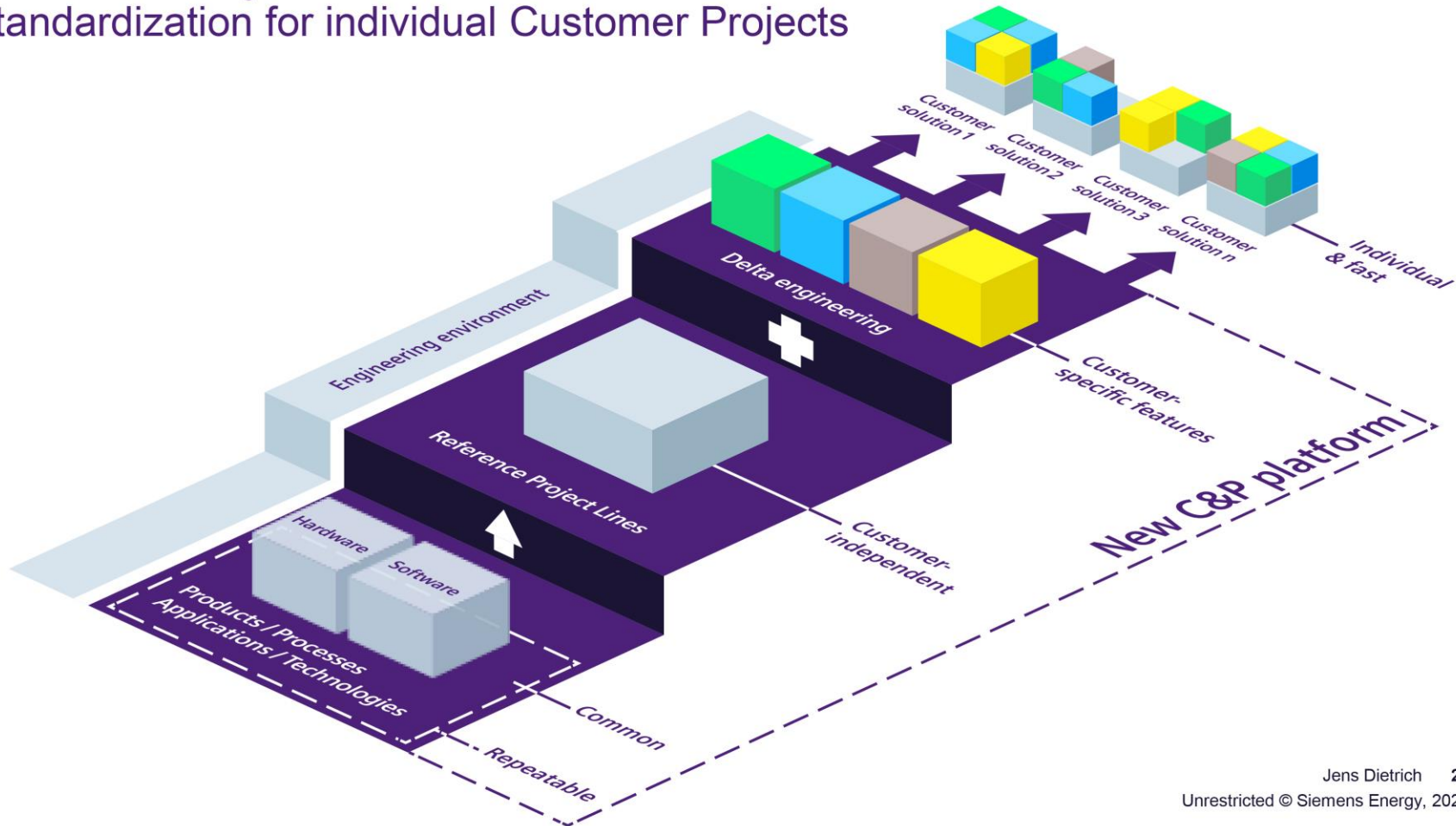
When development extends across the organization



Putting everything together  
A Simulink based digital twin  
lets us analyze and test our  
system early on



### Reference Project Lines Standardization for individual Customer Projects





# Application Trends



**Autonomous**



**Connectivity**



**Electrification**

# Workflow Trends



**Systems Engineering  
& Design**



**Modern  
Software Practices**



**AI for  
System Development**

# MATLAB EXPO

## Thank you



© 2024 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.

