MATLAB AUTOMOTIVE 2022:

AUTOMOTIVE DEVOPS FOR MODEL-BASED DESIGN

Curt Hillier, Technical Director,
Automotive Systems Engineering

APRIL 5, 2022

SECURE CONNECTIONS
FOR A SMARTER WORLD

PUBLIC

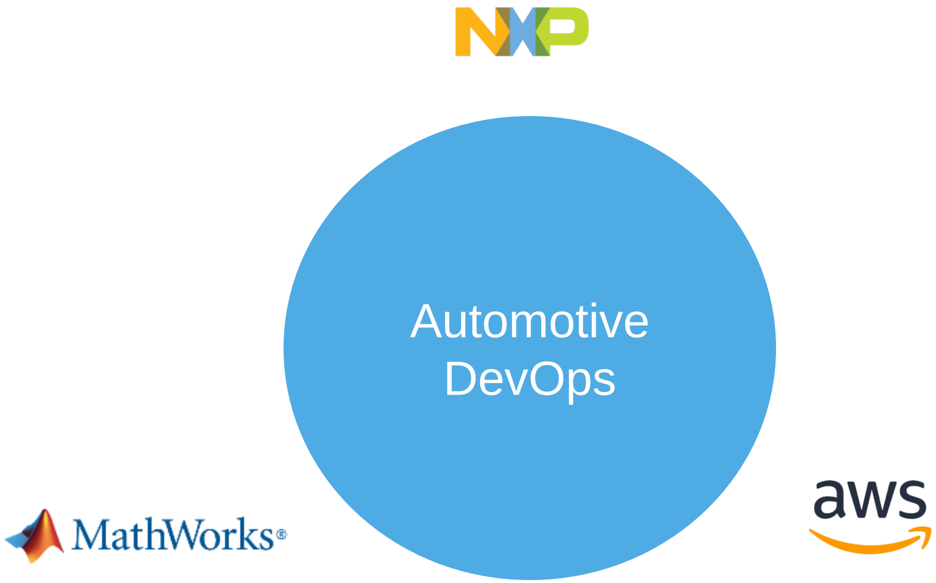
Vehicles are undergoing a transformation from mostly mechanically defined features and capabilities to software defined.

Mechanical → Software

The key automotive industry trends are:

- transition to Agile development
- increased size of software development teams
- migration of tools and workflows to the cloud
- continued adoption of model-based design engineering

NXP, the MathWorks, and AWS have collaborated to build an example Automotive DevOps solution which can enable the future of Model-based design



Let coders do more coding!

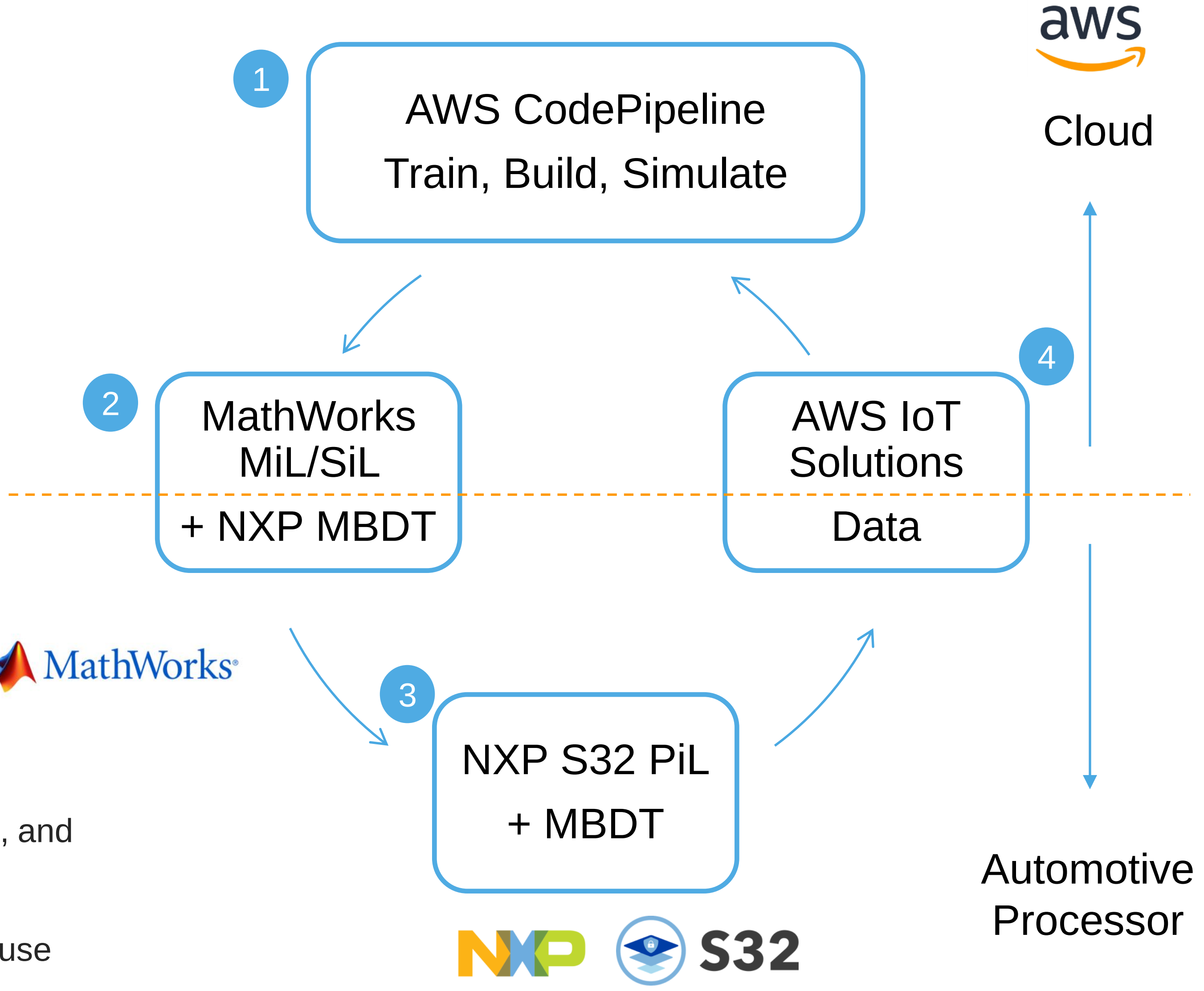
OVERVIEW

- The Automotive DevOps model-based design solution incorporates:
  - AWS CodeSuite services
  - MathWorks® model-based design tools
  - Advanced vehicle control algorithms executing on NXP Automotive processors

- The solution allows users to develop and simulate in the cloud, and then easily deploy to Automotive silicon for algorithm validation.

Major components supporting the solution include:

1. **AWS CodePipeline:** Build and simulate models in the cloud
2. **MathWorks with NXP MBDT:** tools for designing, simulating, and implementing automotive software and system models
3. **NXP GoldBox:** execute algorithm on Automotive processor, use profiler to measure execution time
4. **AWS IoT Solutions:** publish data to the cloud



MiL = Model in the Loop
SiL = Software in the Loop
PiL = Processor in the Loop
MBDT = Model-based Design Toolbox Add On

Solution Products

SECURE CONNECTIONS
FOR A SMARTER WORLD

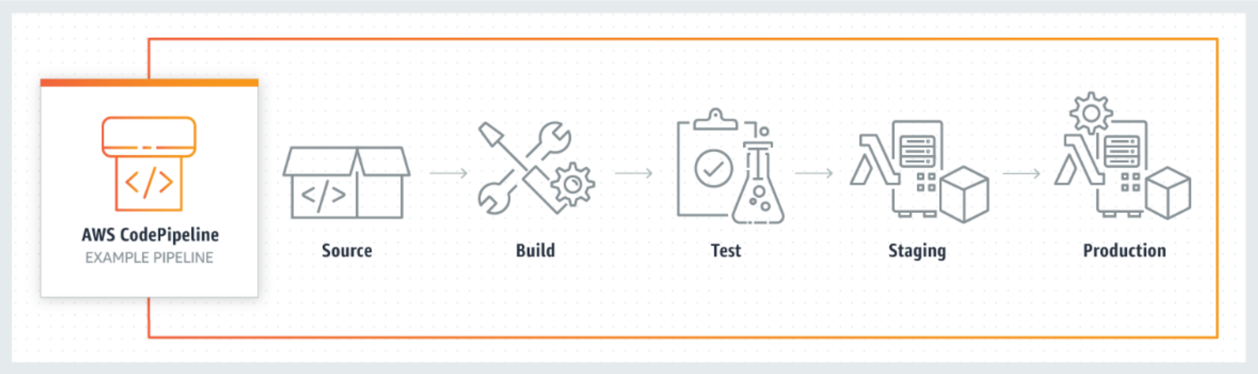
S32

AWS PRODUCTS: CODEPIPELINE

Why did NXP choose to work with AWS?

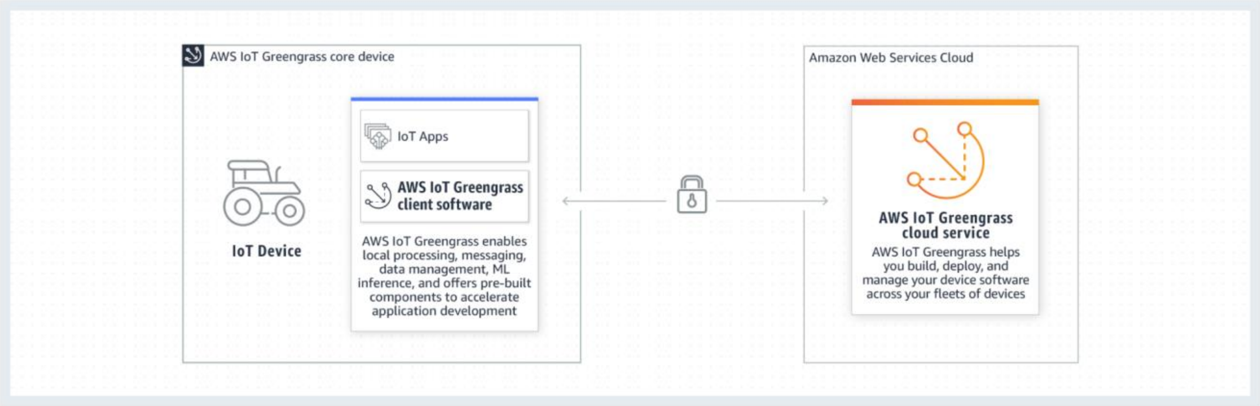
- Numerous solutions for connected vehicles
- Strong collaboration
- Developing leading edge solutions
- Easy to use

How it works



AWS CodePipeline:

- fully managed continuous delivery service that helps you automate your release pipelines for fast and reliable application and infrastructure updates.



AWS Greengrass & IoT Solutions:

- open-source edge runtime and cloud service for building, deploying, and managing device software.

MATHWORKS PRODUCTS

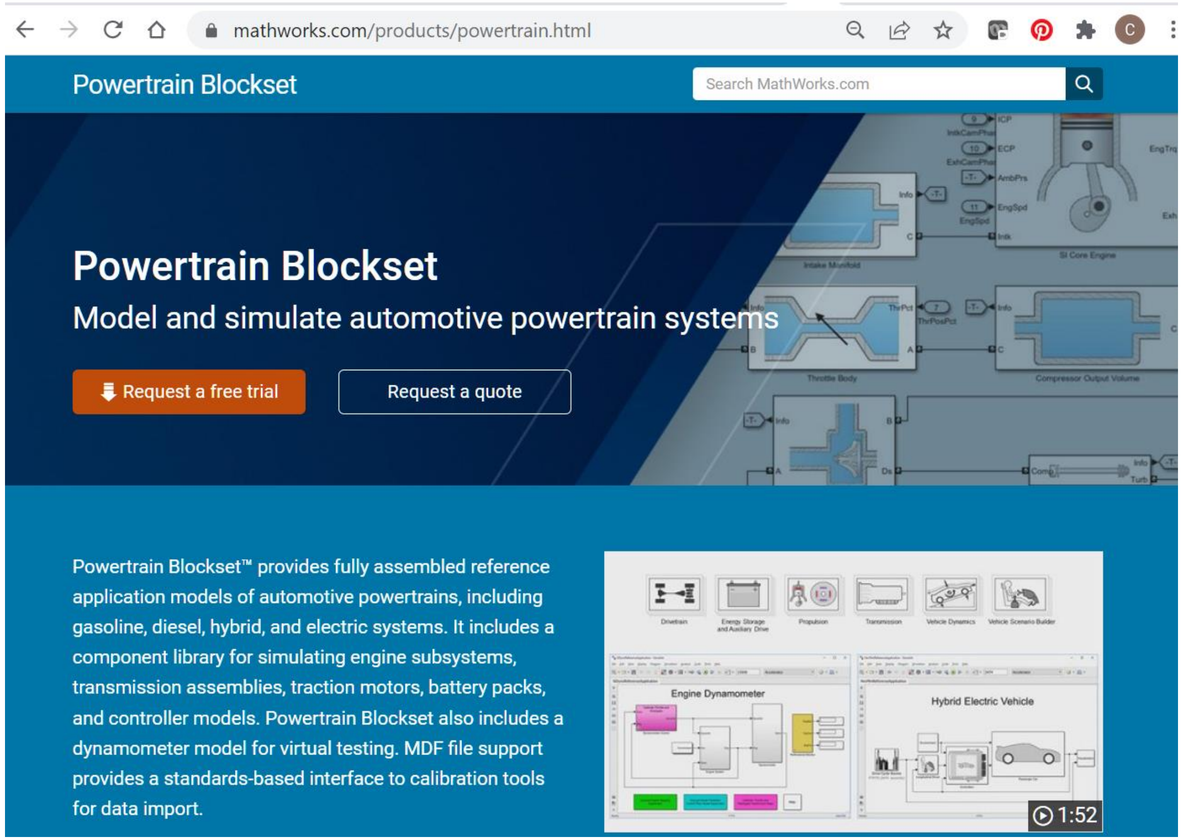
Why did we choose to work with the MathWorks?

- Domain Expertise – across numerous domains
- toolchain for software and simulation development
- Quick to get started: pre-built examples, quality, documentation.

The HEV Model Predictive Control application uses these key MathWorks products:

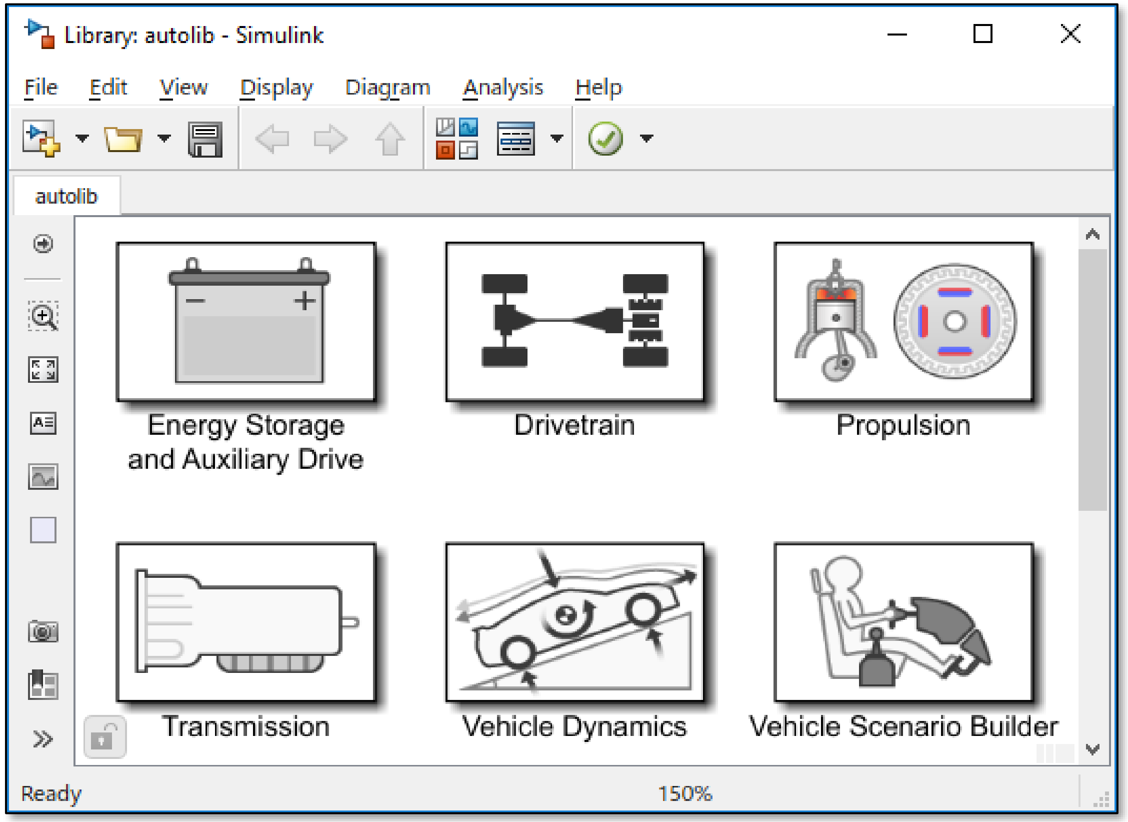
- Simulink
- Powertrain Blockset
- Vehicle Dynamics Blockset
- Embedded Coder

Allows us to build a car (use a pre-built model), generate C code, and run simulation

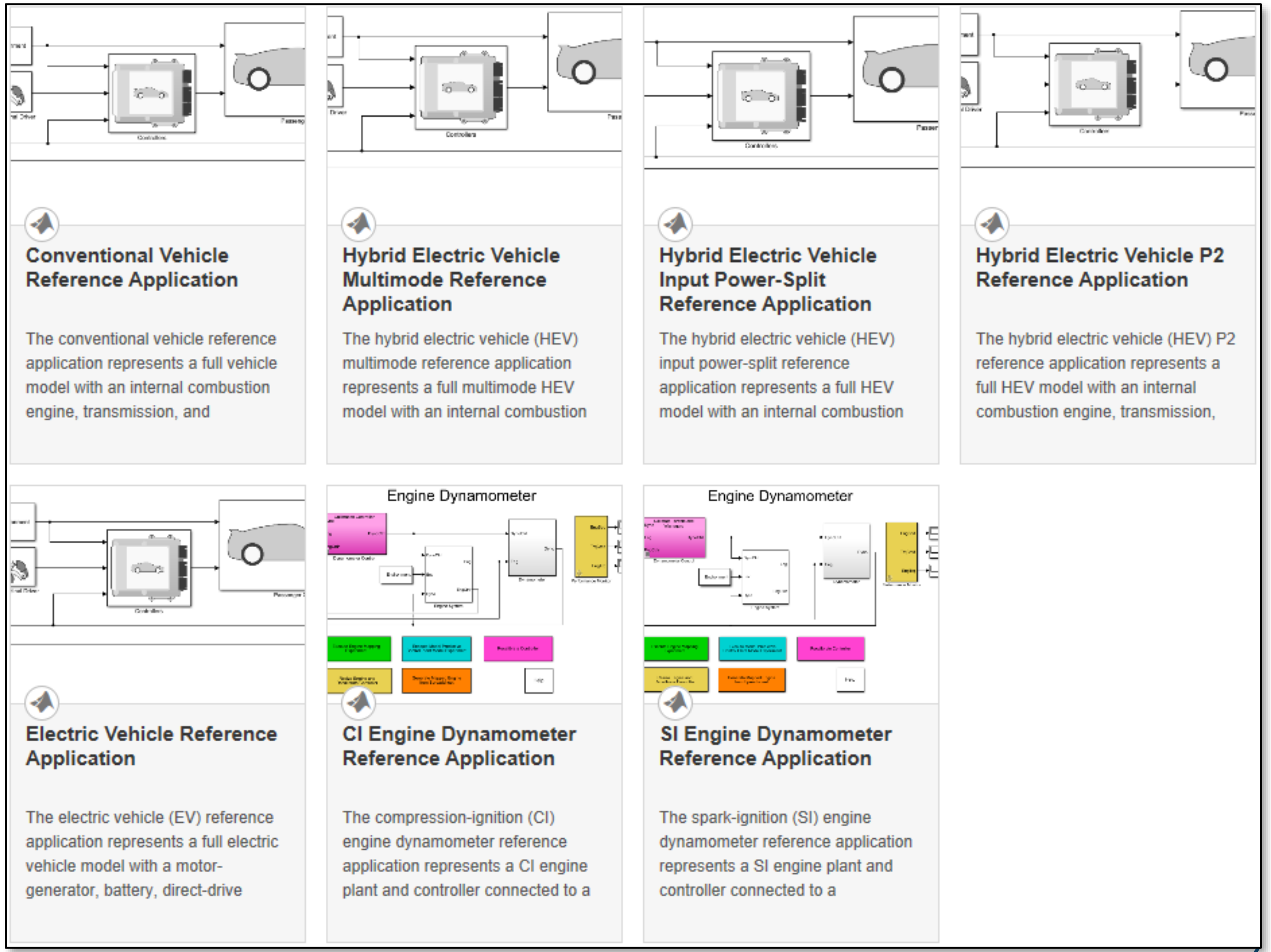


Powertrain Blockset ™

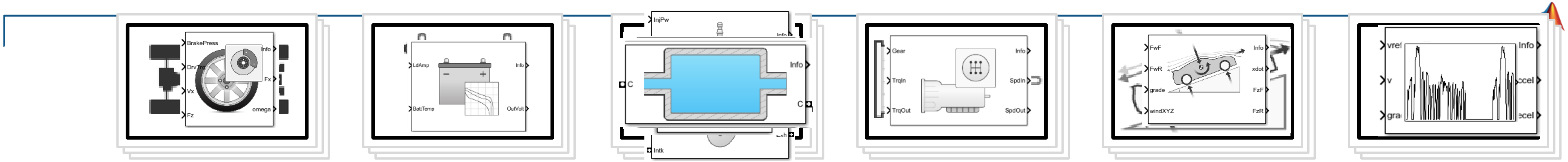
Library of blocks



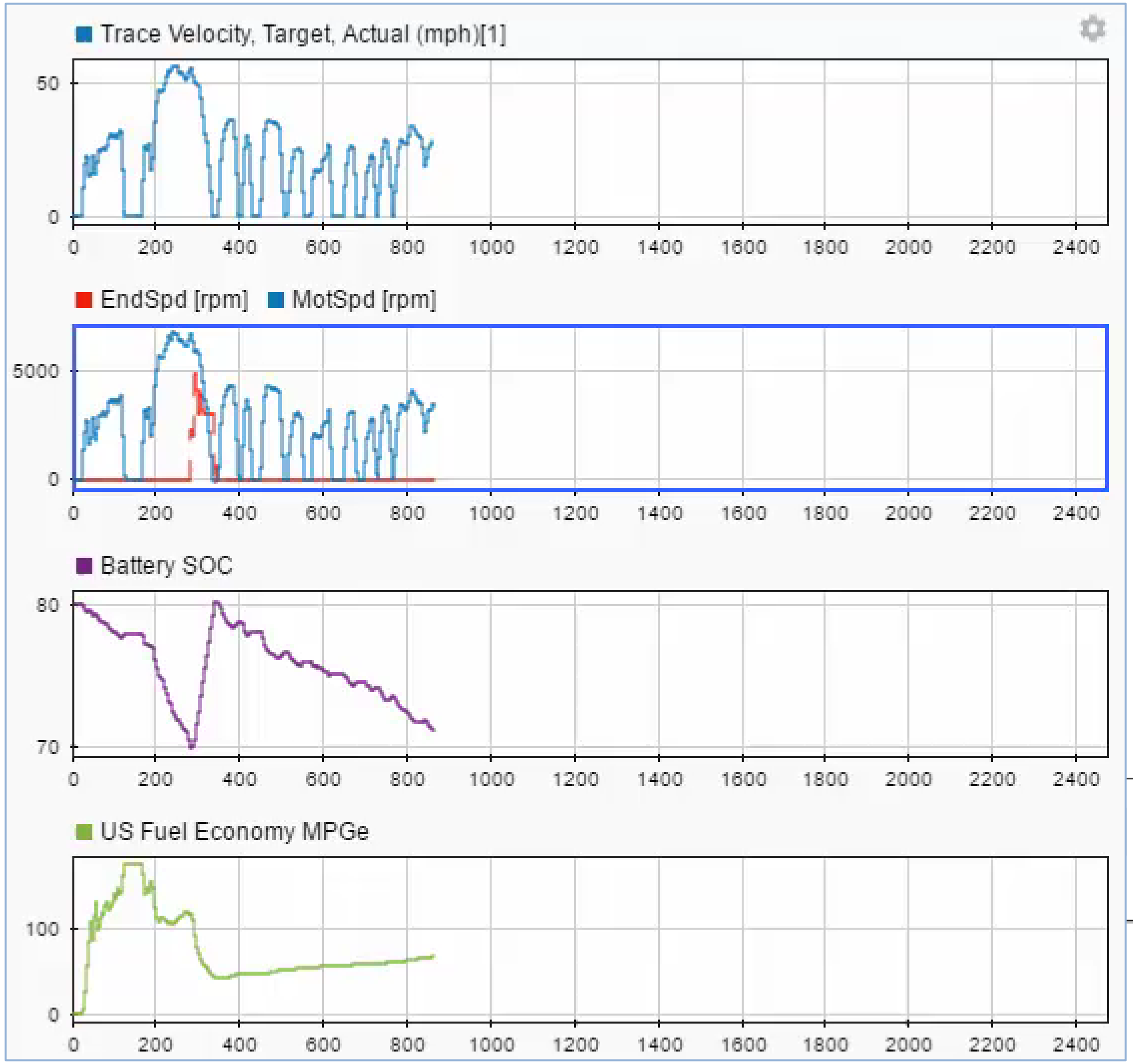
Pre-built reference applications

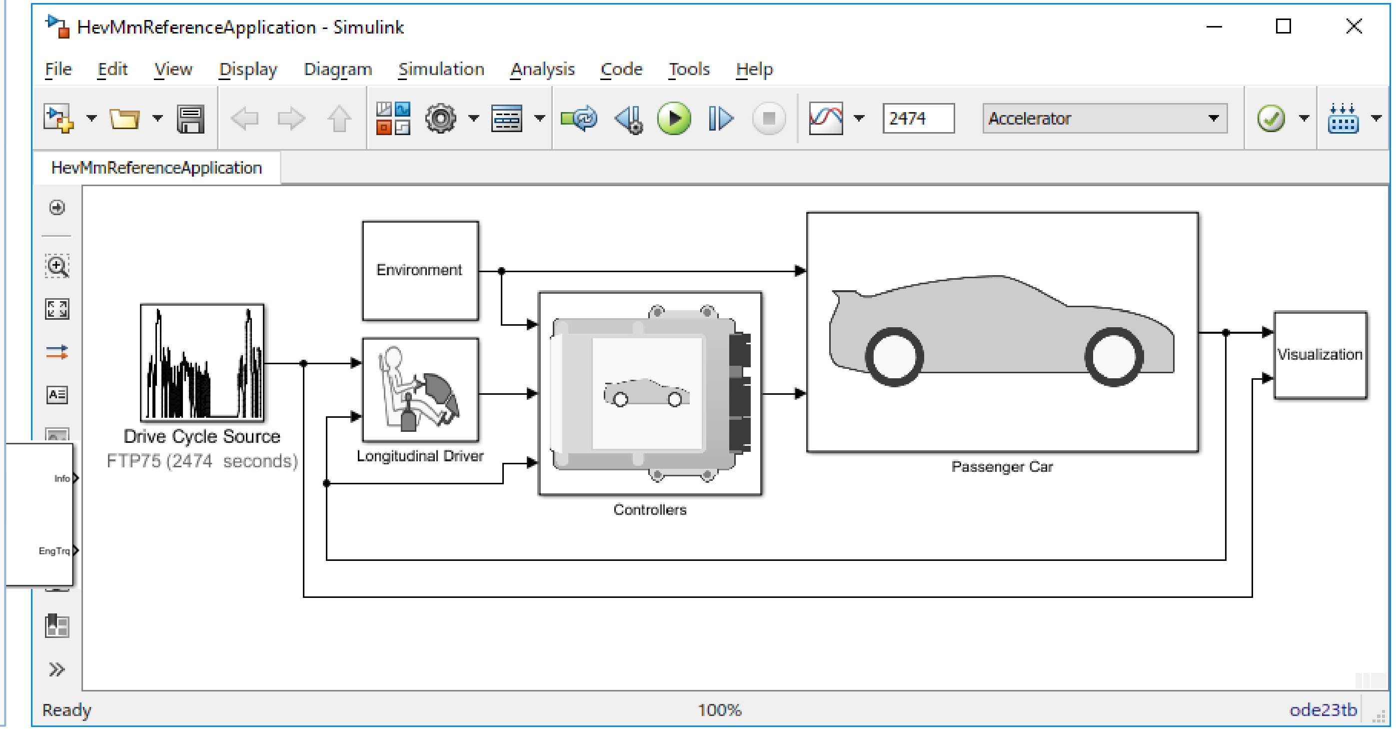


MathWorks



Drivetrain | Energy Storage and Auxiliary Drive | Propulsion | Transmission | Vehicle Dynamics | Vehicle Scenario Builder

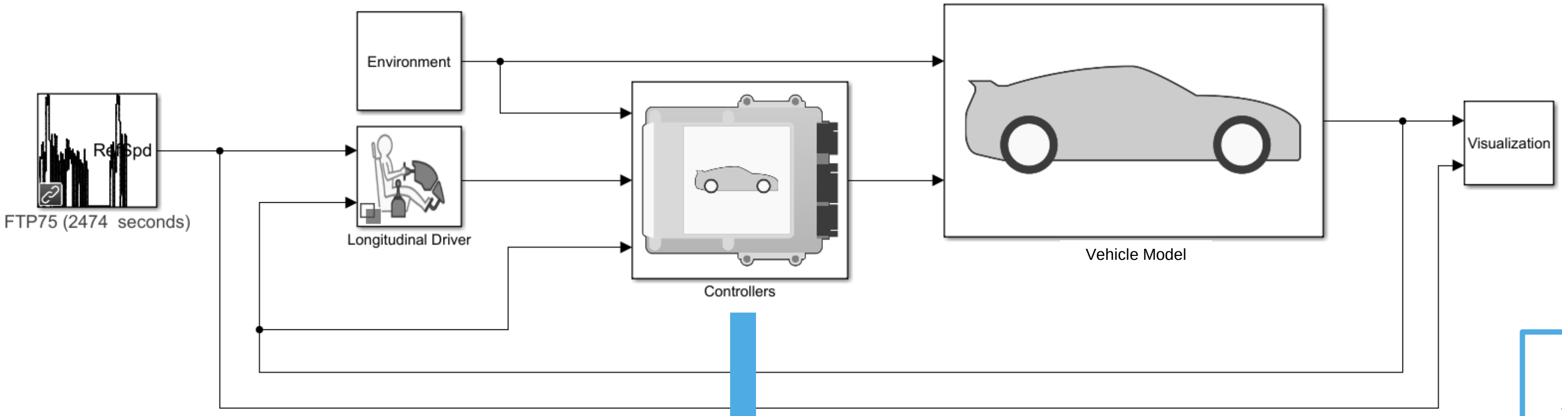


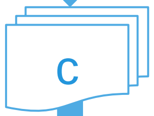


MBDT OVERVIEW: PROCESSOR IN THE LOOP (PIL)

S32

MathWorks®





NXP MBDT "HCP"

NXP MBDT:

- brings the power of Model-based design to NXP Automotive real-time processors and vehicle network processors
- Excellent for early prototyping and performance measurements

NXP **S32G2** GoldBox
Vehicle Network Processor

NXP **S32S** GreenBox 2
Real-Time Controller

Service Oriented Gateway
- in-vehicle secure networking
- edge processing
- cloud interface

xEV Propulsion Controller
- Battery SoX algorithms
- Energy Management
- Motor control

NXP

Automotive DevOps Demonstration

SECURE CONNECTIONS FOR A SMARTER WORLD

PUBLIC

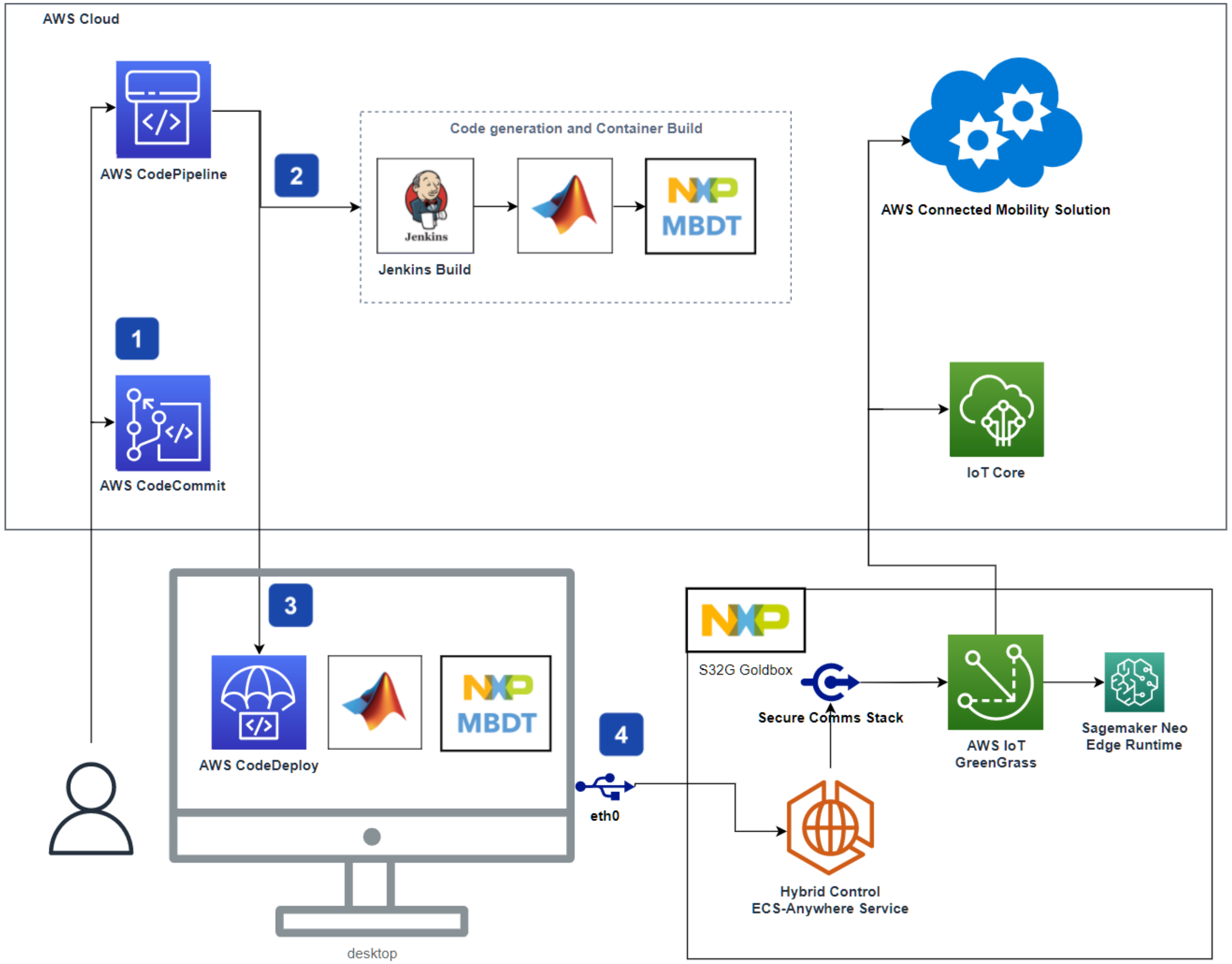
LET'S WALK THROUGH THE DEMONSTRATION: "LOCAL DESKTOP"

1 Code Commit

2 Build

3 Deploy to target

4 PiL Simulation with GUIs / metrics and performance profiling



MathWorks® aws

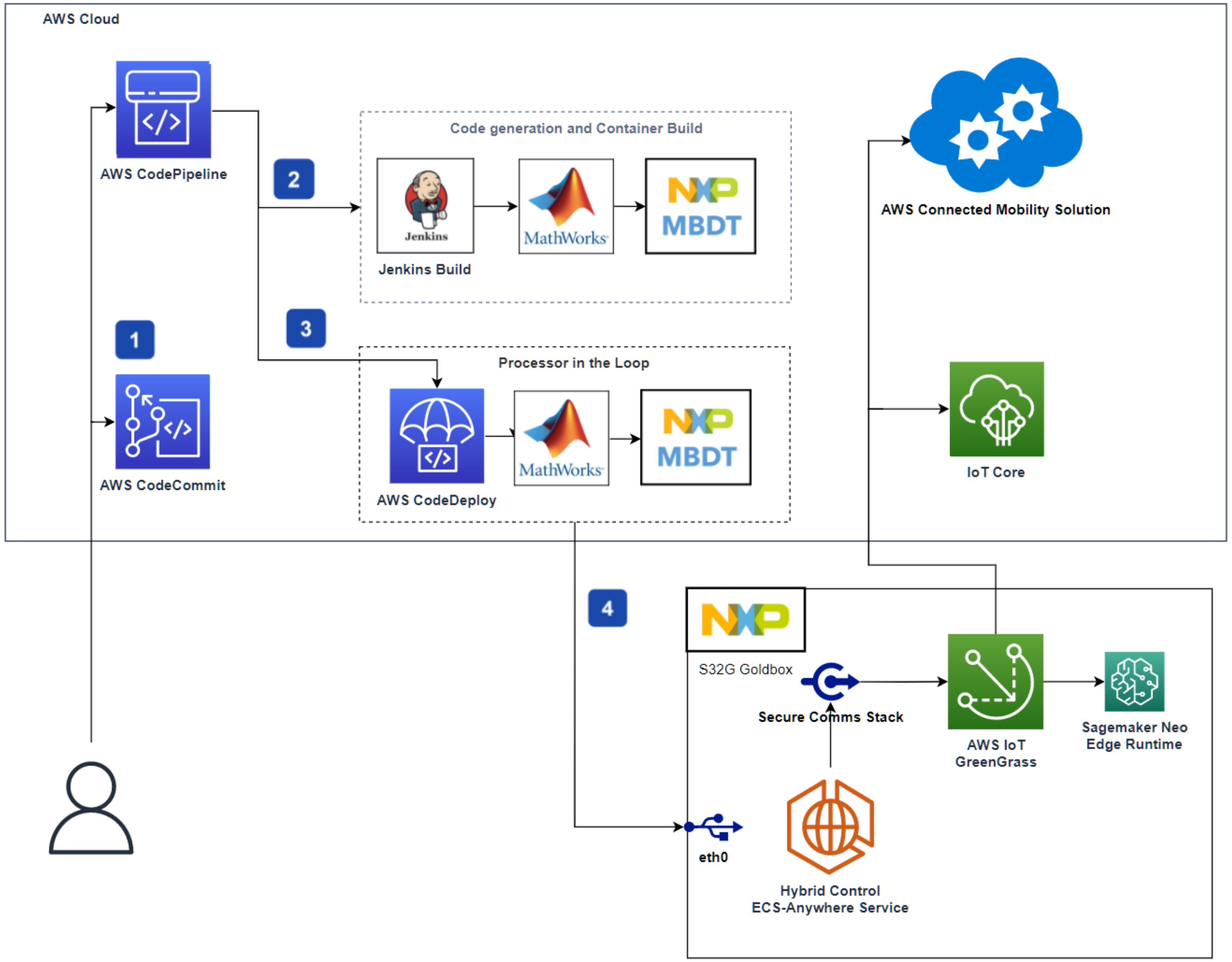
LET'S WALK THROUGH THE DEMONSTRATION: "ALL CLOUD TOOLCHAIN"

1 Code Commit

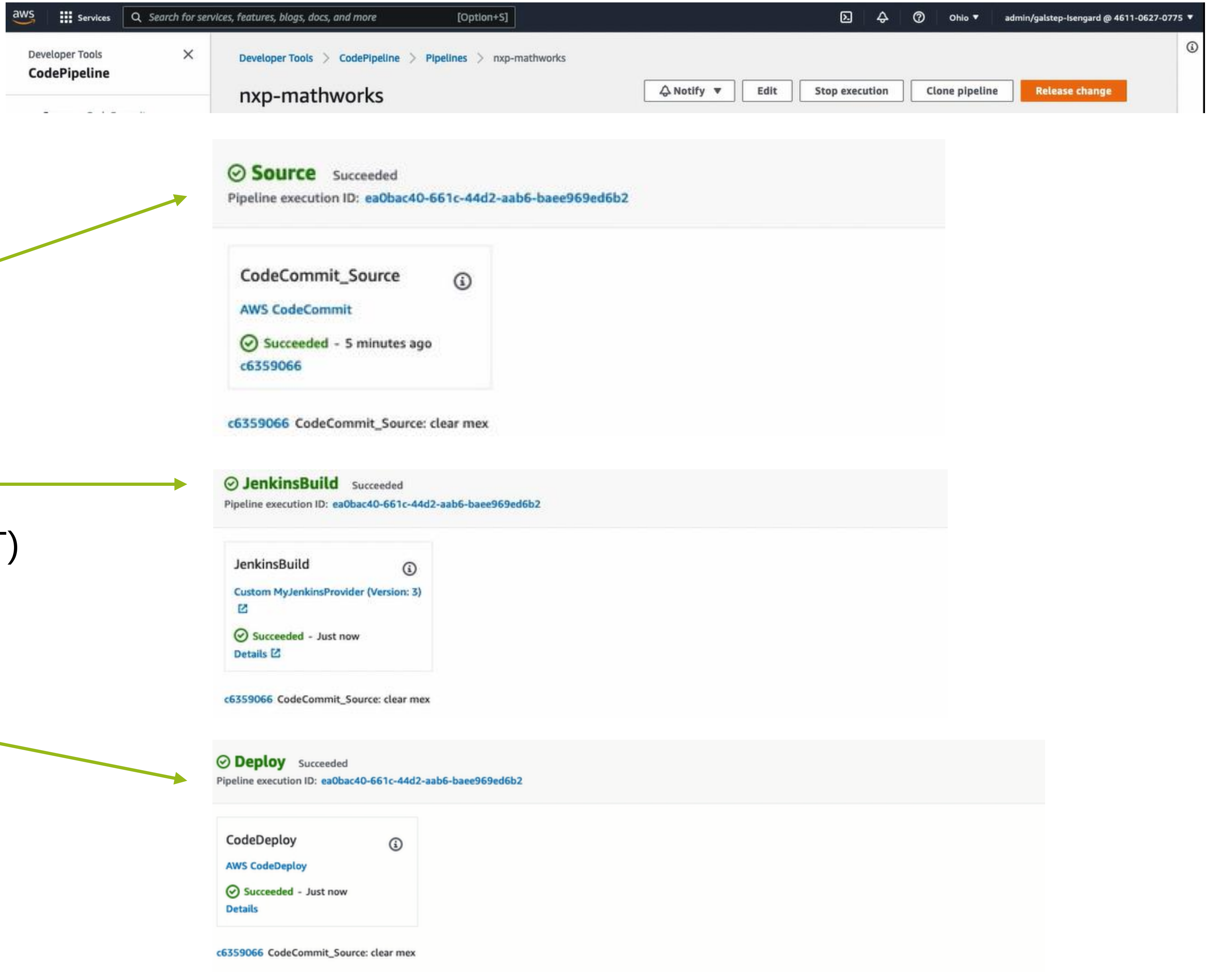
2 Build

3 Deploy to target

4 PiL Simulation with GUIs / metrics and performance profiling



AWS CODEPIPELINE EXAMPLE



Code commit

Code build

(MATLAB codegen and NXP MBDT)

Code deploy

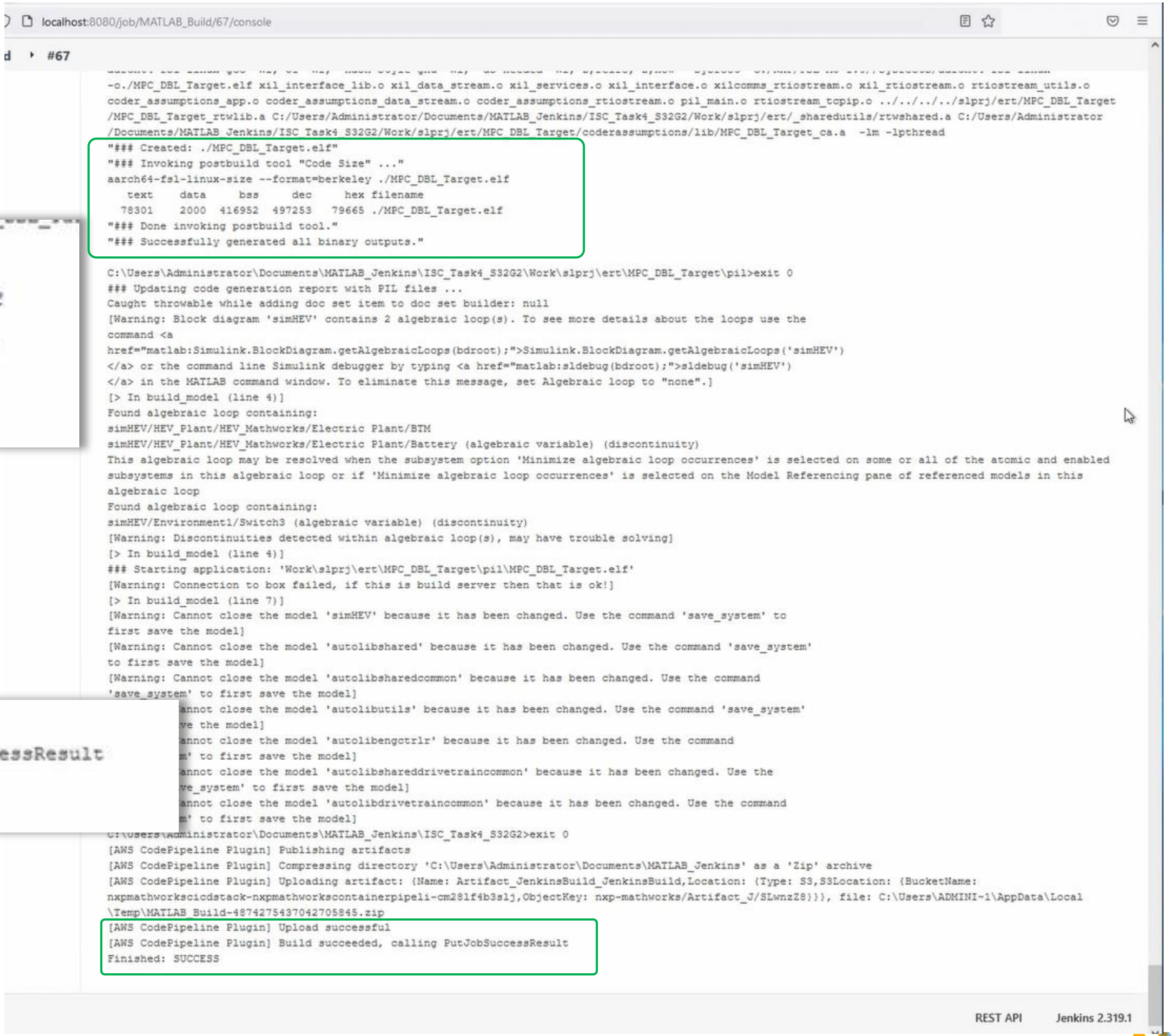
JENKINS BUILD OUTPUT

- Create elf file for target execution on S32G

```
"### Created: ./MPC_DBL_Target.elf"
"### Invoking postbuild tool "Code Size" ..."
aarch64-fsl-linux-size --format=berkeley ./MPC_DBL_Target.elf
   text    data     bss     dec     hex filename
  78301    2000  416952  497253   79665 ./MPC_DBL_Target.elf
"### Done invoking postbuild tool."
"### Successfully generated all binary outputs."
```

- AWS CodePipeline Build successful

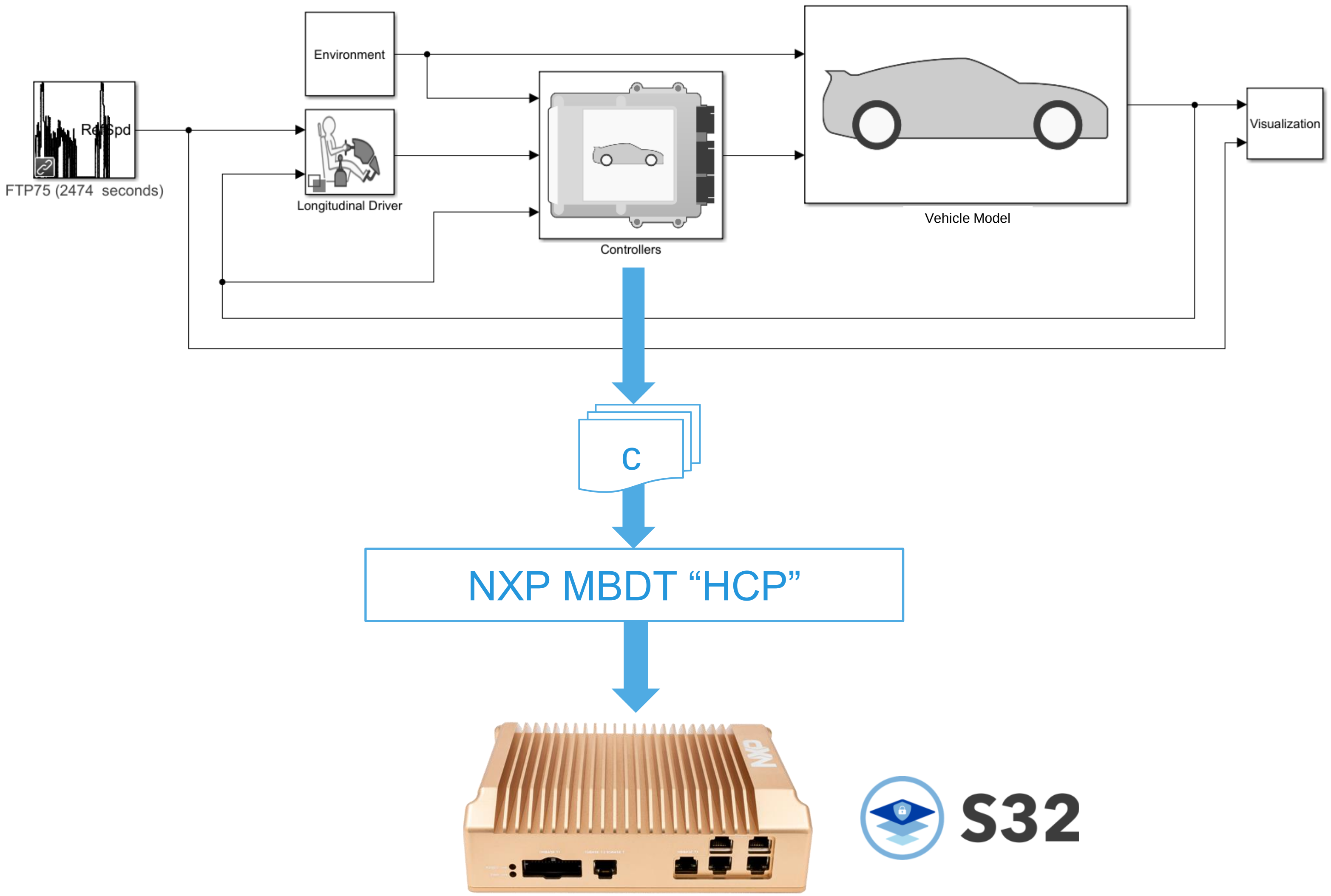
```
[AWS CodePipeline Plugin] Upload successful
[AWS CodePipeline Plugin] Build succeeded, calling PutJobSuccessResult
Finished: SUCCESS
```



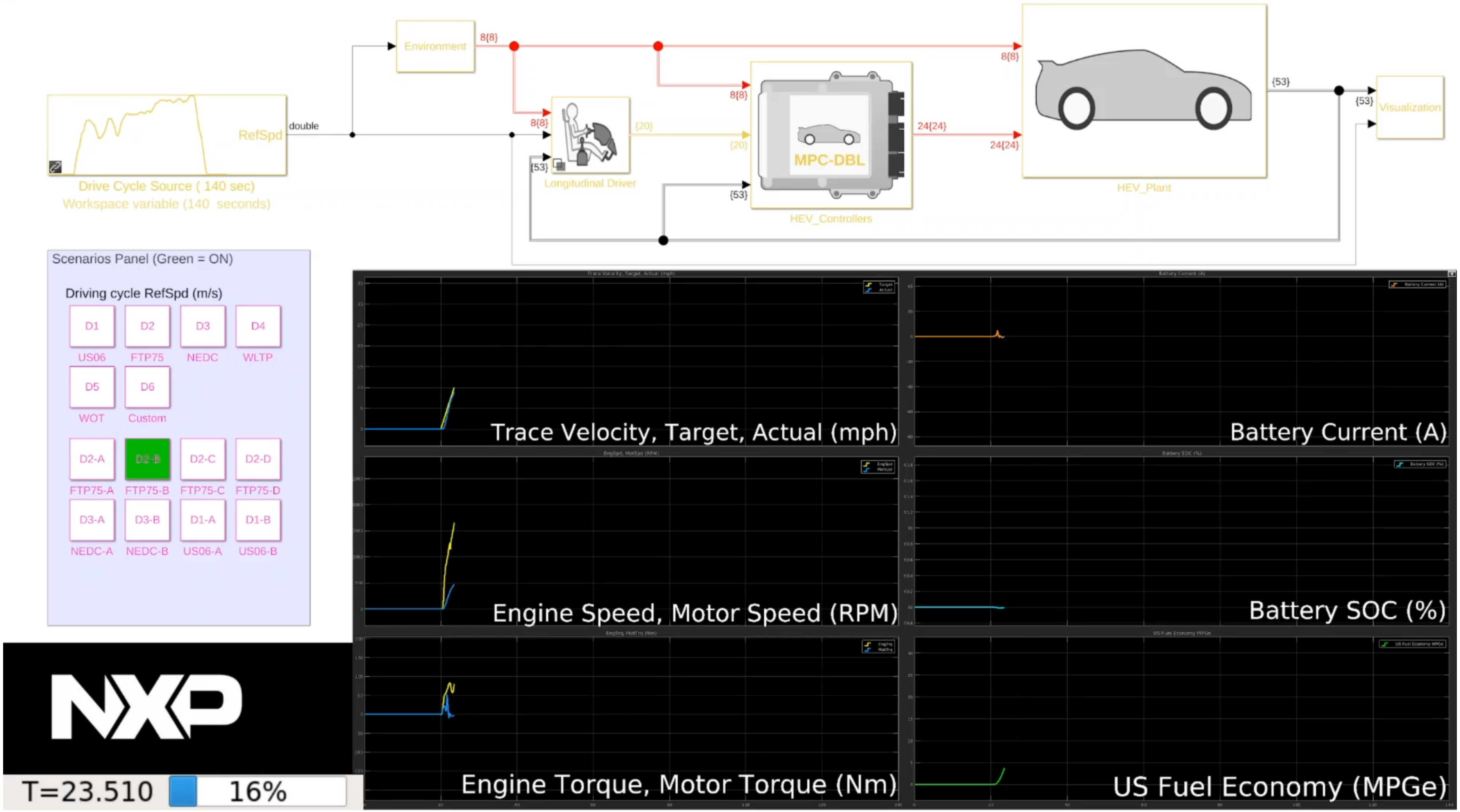
PIL SIMULATION

MathWorks®

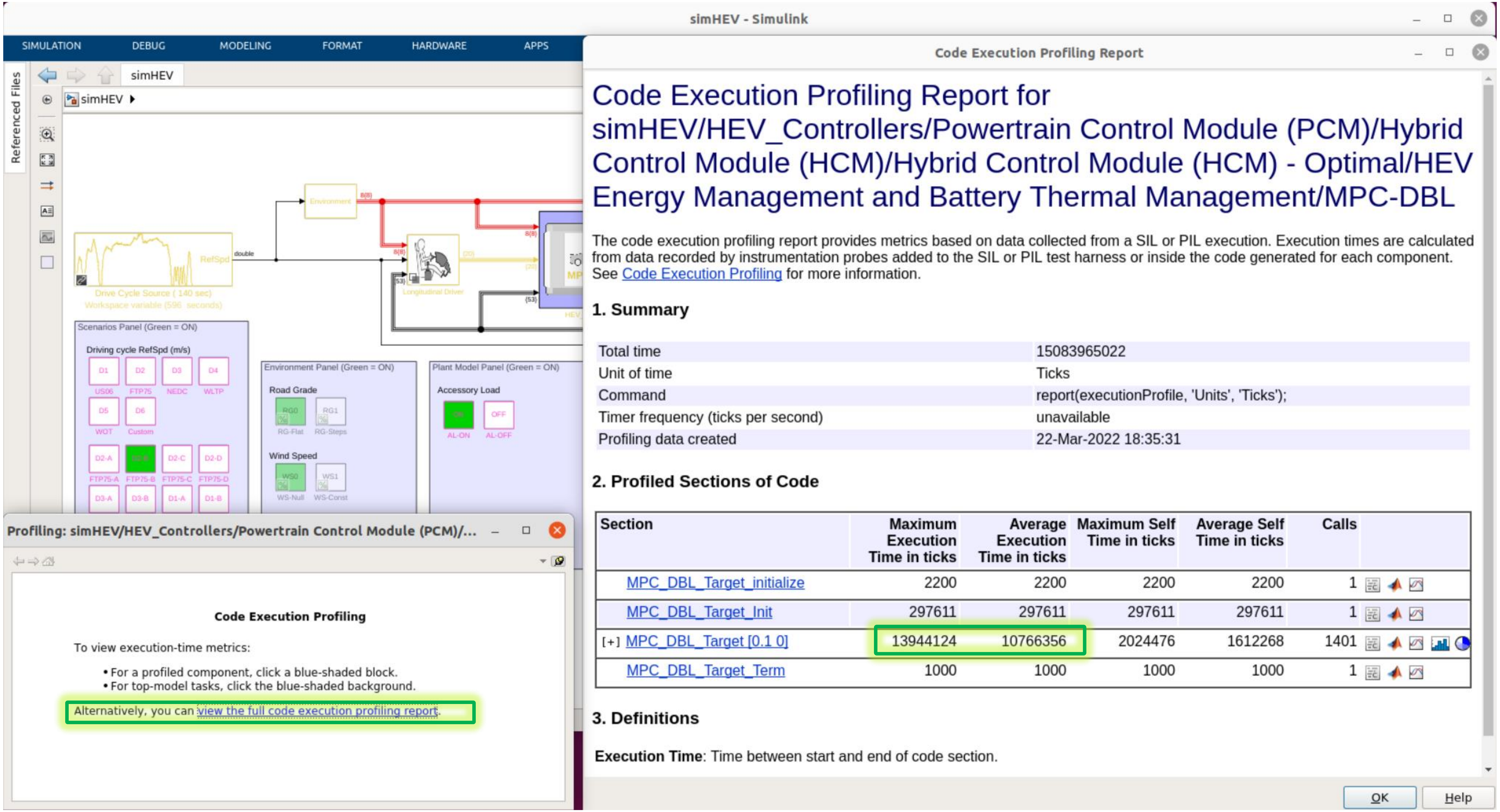
- ✓ Processor in the Loop
- ✓ Simulate with standard drive cycles (e.g. FTP-75, US06, WLTP III)
- ✓ S32G target profiling supported by NXP MBDT



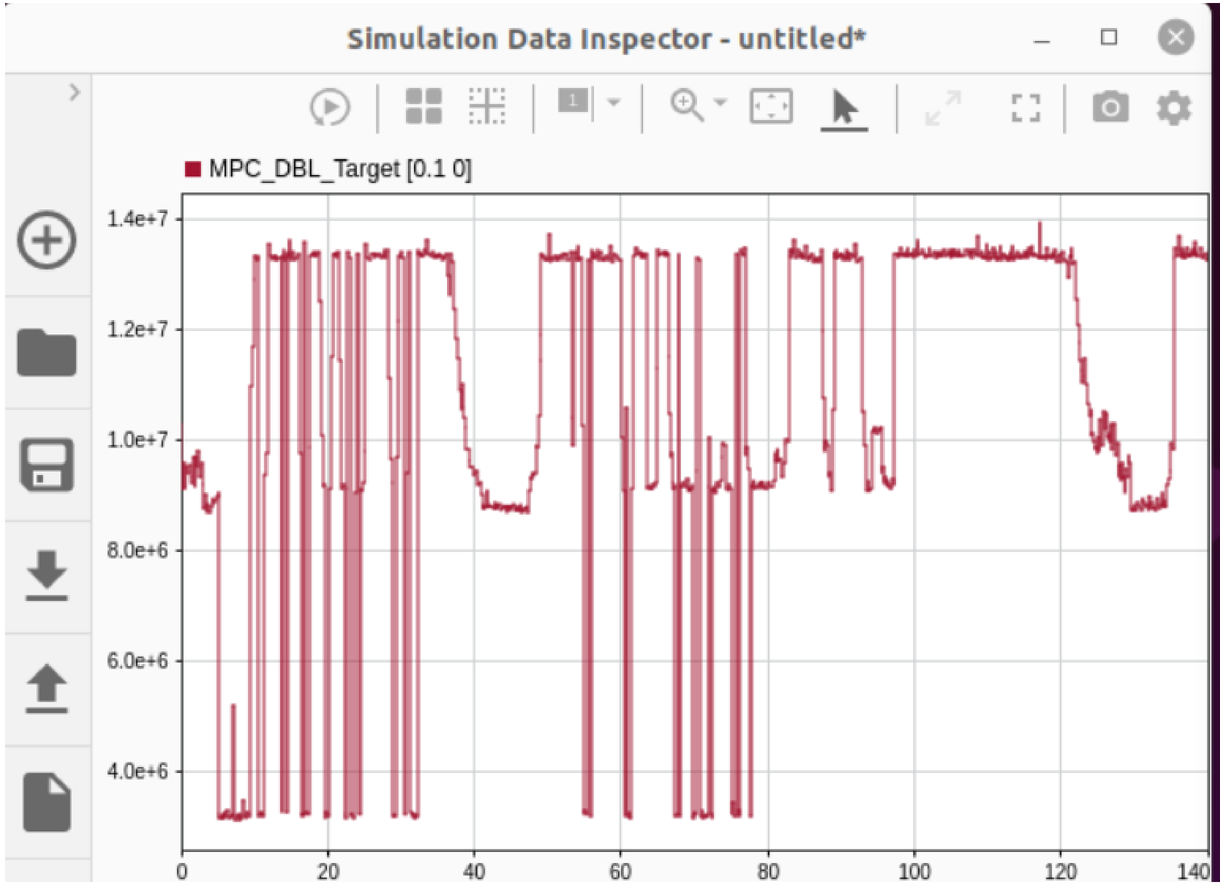
NXP S32G2 GoldBox
Arm Cortex-A53


Environment
8{8}
Drive Cycle Source (140 sec)
Workspace variable (140 seconds)
RefSpd
double
Longitudinal Driver
{20}
{53}
MPC-DBL
HEV_Controllers
24{24}
HEV_Plant
Visualization
Scenarios Panel (Green = ON)
Driving cycle RefSpd (m/s)
D1
US06
D2
FTP75
D3
NEDC
D4
WLTP
D5
WOT
D6
Custom
D2-A
FTP75-A
D2-B
FTP75-B
D2-C
FTP75-C
D2-D
FTP75-D
D3-A
NEDC-A
D3-B
NEDC-B
D1-A
US06-A
D1-B
US06-B
NXP
T=23.510
16%
Trace Velocity, Target, Actual (mph)
Battery Current (A)
Engine Speed, Motor Speed (RPM)
Battery SOC (%)
Engine Torque, Motor Torque (Nm)
US Fuel Economy (MPGe)

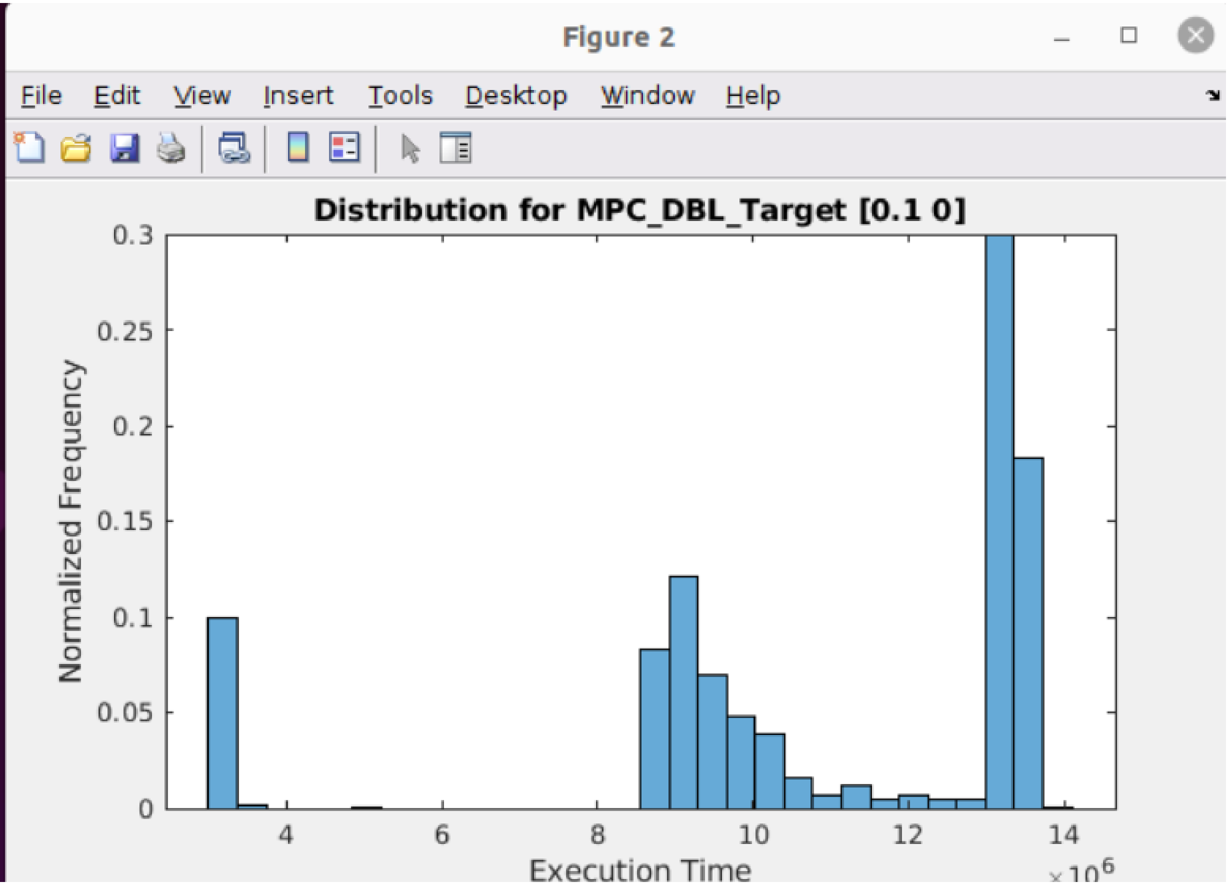
PROFILING


simHEV - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
HARDWARE
APPS
simHEV
Referenced Files
Code Execution Profiling Report
Code Execution Profiling Report for simHEV/HEV_Controllers/Powertrain Control Module (PCM)/Hybrid Control Module (HCM)/Hybrid Control Module (HCM) - Optimal/HEV Energy Management and Battery Thermal Management/MPC-DBL
The code execution profiling report provides metrics based on data collected from a SIL or PIL execution. Execution times are calculated from data recorded by instrumentation probes added to the SIL or PIL test harness or inside the code generated for each component.
See Code Execution Profiling for more information.
1. Summary
Total time 15083965022
Unit of time Ticks
Command report(executionProfile, 'Units', 'Ticks');
Timer frequency (ticks per second) unavailable
Profiling data created 22-Mar-2022 18:35:31
2. Profiled Sections of Code
Section | Maximum Execution Time in ticks | Average Execution Time in ticks | Maximum Self Time in ticks | Average Self Time in ticks | Calls
MPC_DBL_Target_initialize 2200 2200 2200 2200 1
MPC_DBL_Target_Init 297611 297611 297611 297611 1
[+] MPC_DBL_Target [0.1 0] 13944124 10766356 2024476 1612268 1401
MPC_DBL_Target_Term 1000 1000 1000 1000 1
3. Definitions
Execution Time: Time between start and end of code section.
OK
Help
Profiling: simHEV/HEV_Controllers/Powertrain Control Module (PCM)/...
Code Execution Profiling
To view execution-time metrics:
• For a profiled component, click a blue-shaded block.
• For top-model tasks, click the blue-shaded background.
Alternatively, you can view the full code execution profiling report.
Scenarios Panel (Green = ON)
Driving cycle RefSpd (m/s)
Environment Panel (Green = ON)
Road Grade
Wind Speed
Plant Model Panel (Green = ON)
Accessory Load

PROFILING RESULTS: TIME SERIES AND HISTOGRAM



Time Series



Histogram

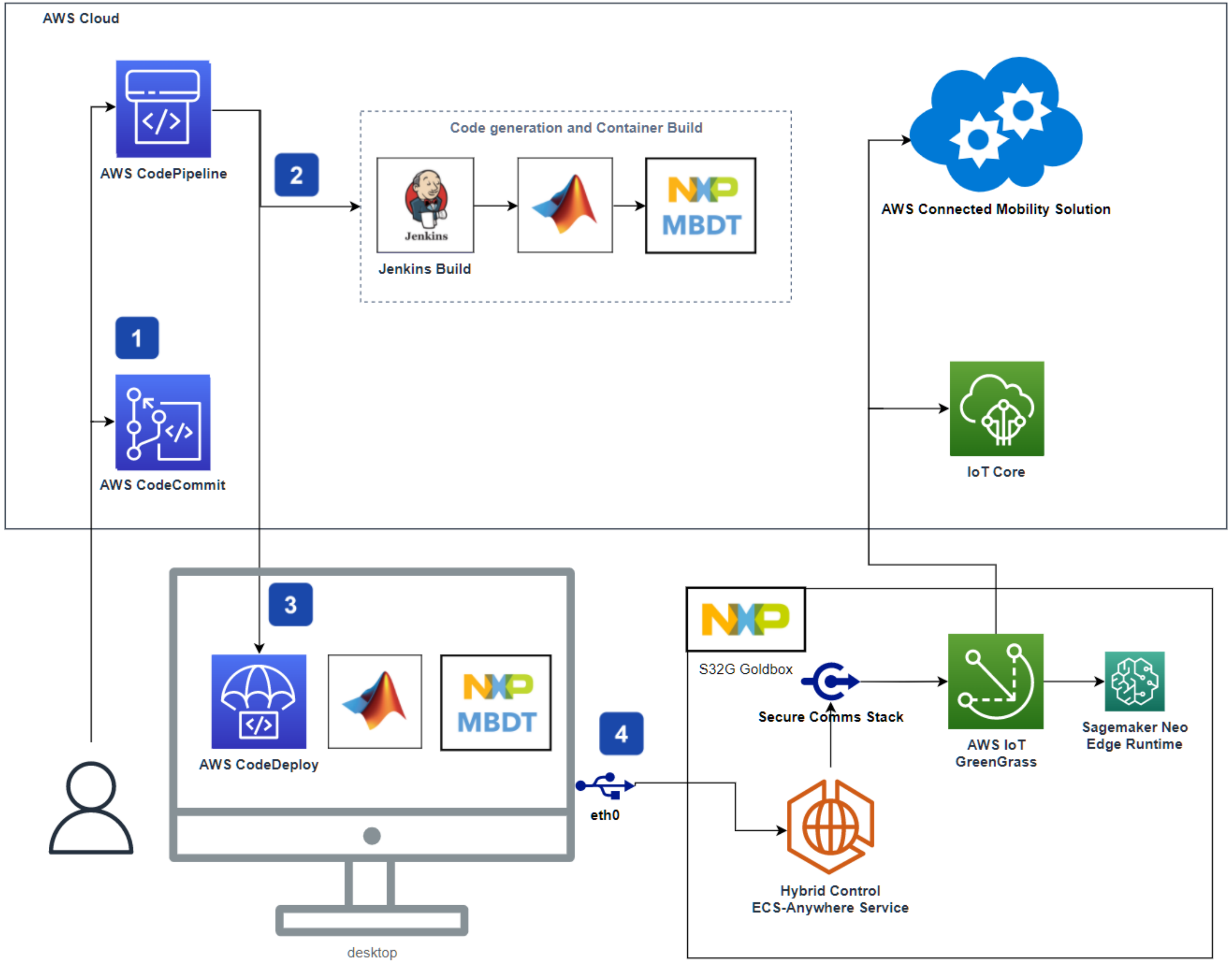
RECAP: AUTOMOTIVE DEVOPS MODEL-BASED DESIGN EXAMPLE

1 **Code Commit**: track changes for global user base

2 **Code Build**: leverage installed tools for multiple users; integrated with MathWorks and NXP model-based design tools

3 **Deploy**: push compiled code to HIL / PIL systems for testing

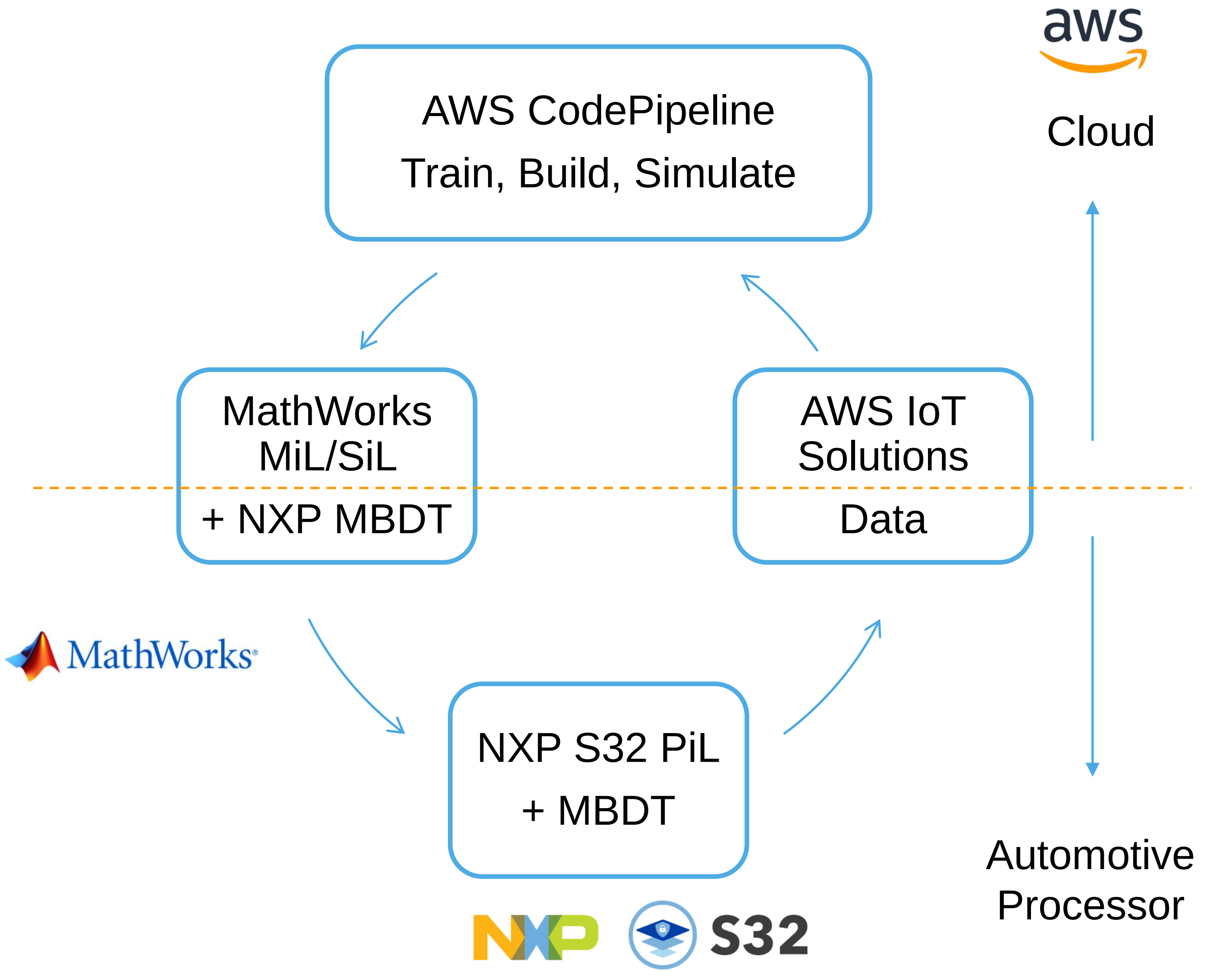
4 **Simulate**: PIL execution with real-time profiling on Automotive processors



Design, build and simulate in the cloud. Engineers use model-based systems engineering (MBSE) to manage system complexity, improve communication, and produce optimized systems

Deploy to the Automotive Edge. NXP's S32G Vehicle Network Processors interface with all the vehicle functional domains and provide secure processing (AI/ML) and network acceleration for vehicle edge services.

Integration with AWS CodePipeline and AWS IoT Greengrass enables a DevOps workflow built on AWS.



Conclusion

SECURE CONNECTIONS
FOR A SMARTER WORLD

S32

SUMMARY AND SOLUTION BENEFITS

- Develop in the cloud and deploy to edge

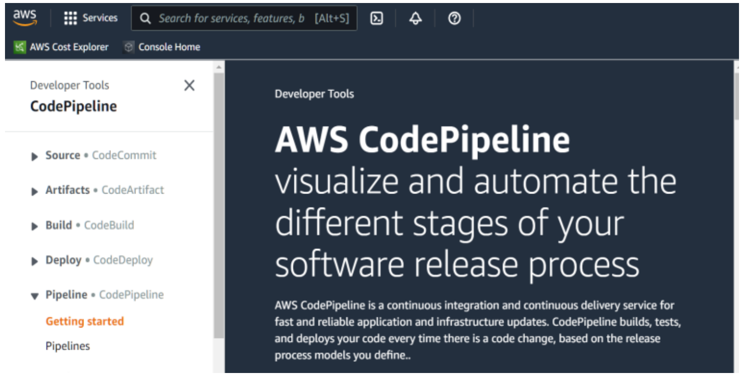
- To meet the automotive trends, we highlighted:
  - transition to a Continuous Integration / Continuous Deployment workflow
  → AWS CodePipeline

  - increased size of software development teams
  → cloud migration using AWS solutions

  - migration of tools and workflows to the cloud
  → AWS hosting MathWorks and NXP MBDT

  - continued adoption of model-based design engineering
  → supported by MathWorks toolchains and NXP MBDT

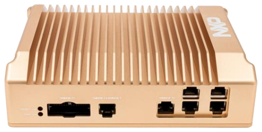
- Edge deployment using NXP MBDT and Automotive Real-Time Processors and Vehicle Network Processors:
  → real world benchmarking on Automotive targets



S32

FOR MORE INFORMATION

- **NXP:**

GoldBox	GreenBox	MBDT
		
www.nxp.com/GoldBox	www.nxp.com/GreenBox	www.nxp.com/MBDT

  - Connected EV Management Demo

- **AWS:**
  - AWS CodePipeline \| Continuous Integration & Continuous Delivery (amazon.com)
  - Intelligence at the IoT Edge — AWS IoT Greengrass — Amazon Web Services

- **MathWorks:**
  - https://www.mathworks.com/solutions/automotive/virtual-vehicle.html

- For follow up: curt.hillier@nxp.com

SECURE CONNECTIONS
FOR A SMARTER WORLD