Agentic Model-Based Design with FMI

- A Two-Axis Interoperability Architecture

- Presentation at the Asian Modelica & FMI Conference, Hangzhou, 09/21/2026

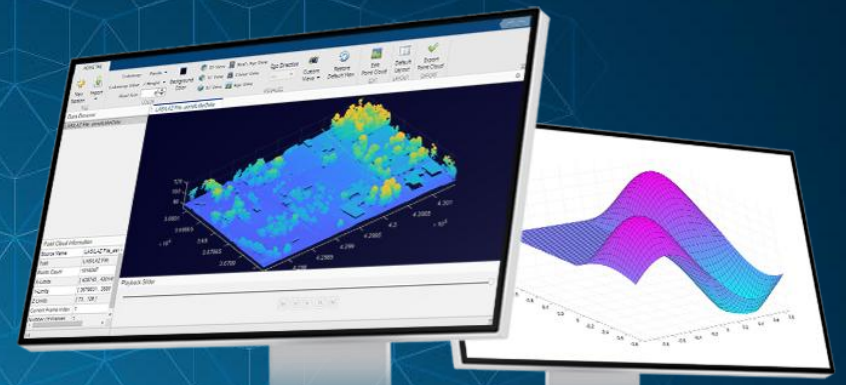


Weiwu Li



Haihua Feng

MATLAB® & SIMULINK®



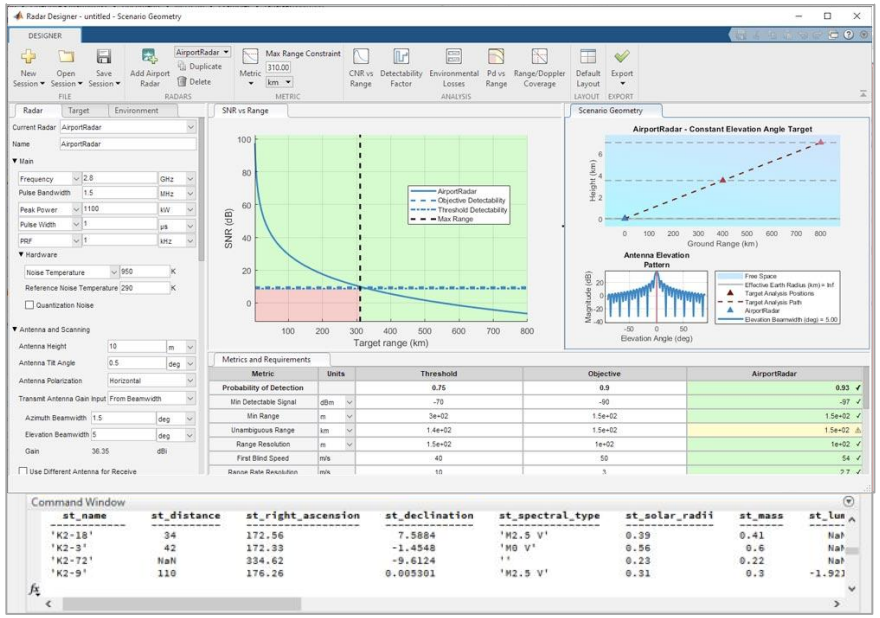
MathWorks Products

MATLAB – Programming environment for algorithm development, data analysis, visualization, and numeric computation.

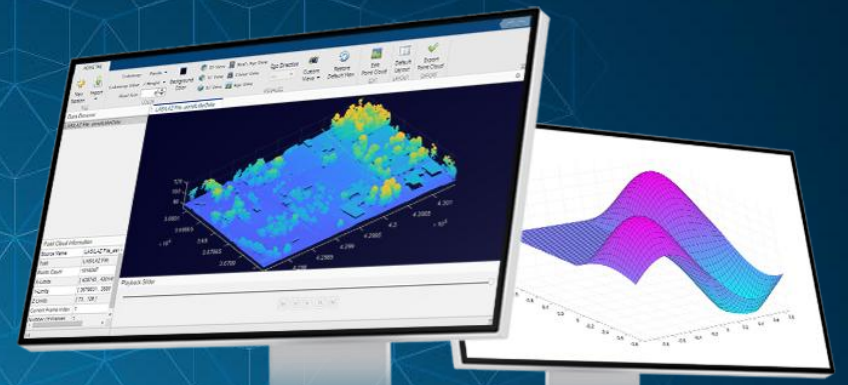
Simulink – Block diagram environment for simulation and Model-Based Design of multidomain and engineered systems.

130+ add-on products for specialized tasks.

Radar Toolbox



MATLAB® & SIMULINK®



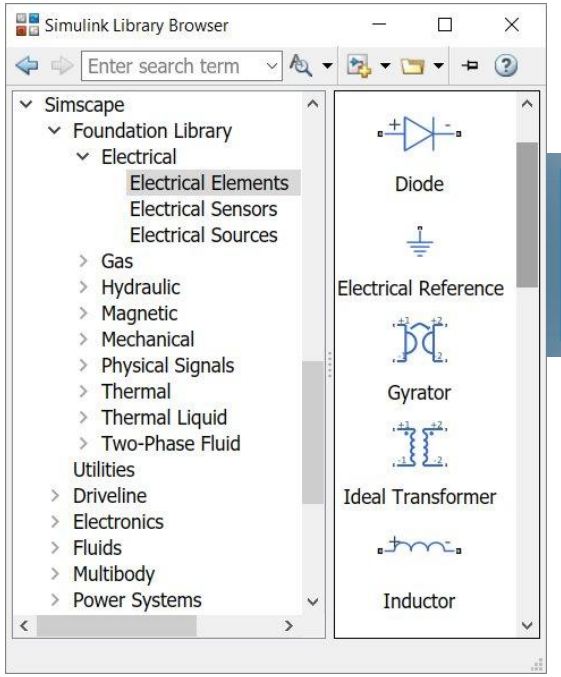
MathWorks Products

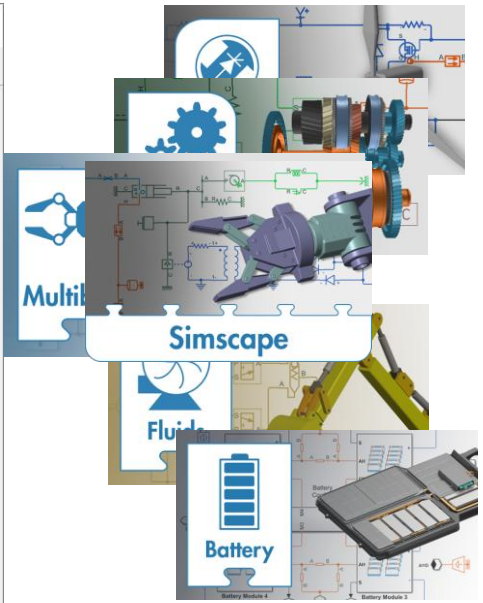
MATLAB – Programming environment for algorithm development, data analysis, visualization, and numeric computation.

Simulink – Block diagram environment for simulation and Model-Based Design of multidomain and engineered systems.

130+ add-on products for specialized tasks.

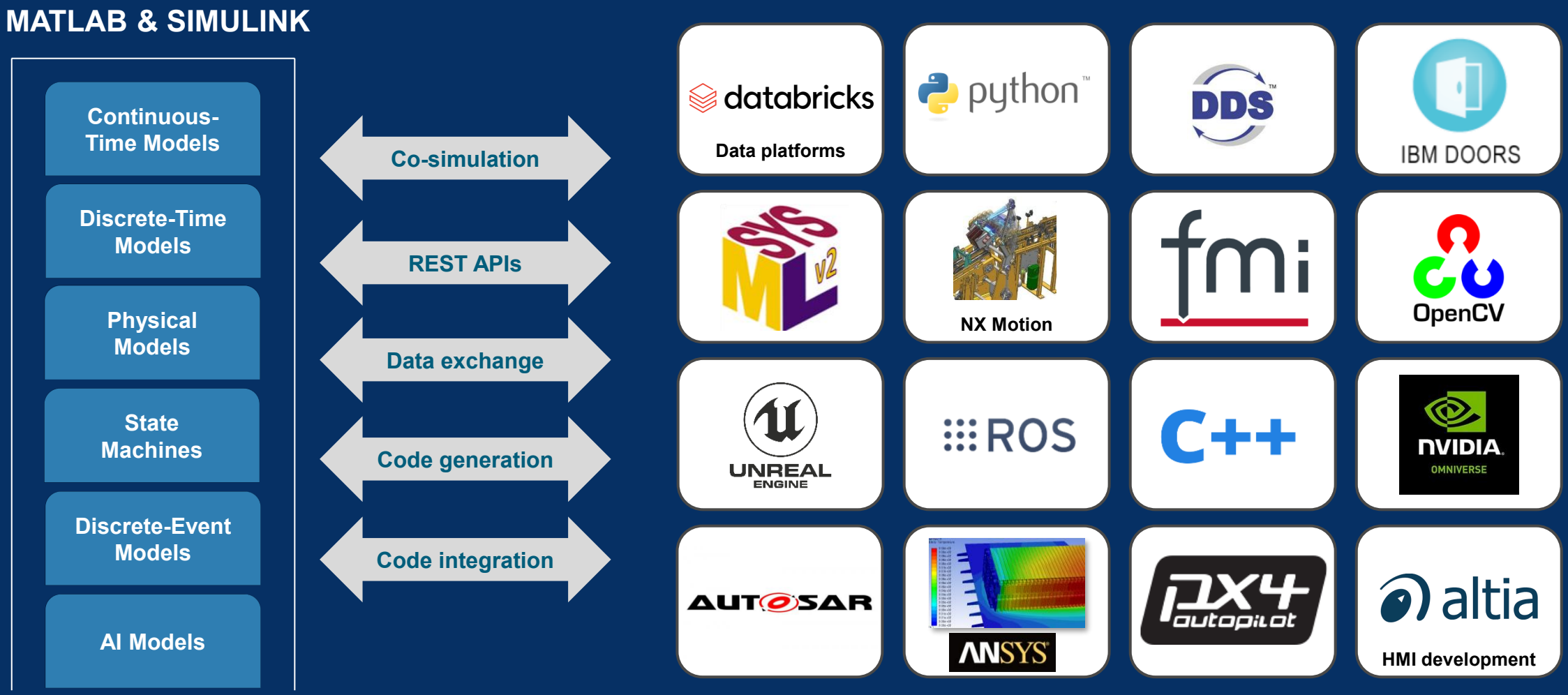
Simscape





Interoperability matters

Connecting with 100+ third-party tools, standards, and languages



Example list of supported 3rd party tools, standards, and languages

MathWorks

Agenda

- Model-Based Design with agentic AI
- A two axis-architecture for interoperability
- Simulink Agentic Toolkit: An intro
- Case study – battery fault protection
- Summary

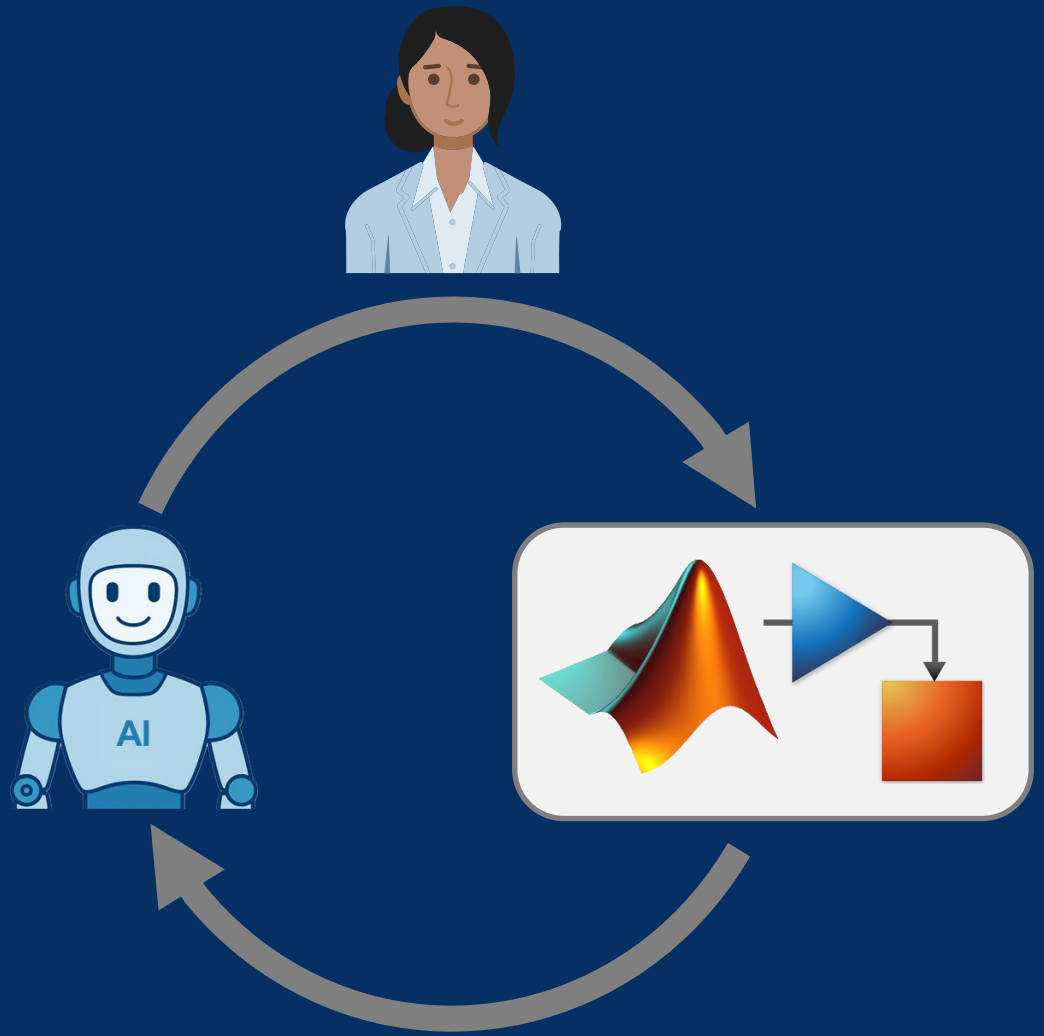
MathWorks®

Agenda

- **Model-Based Design with agentic AI**
- A two axis-architecture for interoperability
- Simulink Agentic Toolkit: An intro
- Case study – battery fault protection
- Summary

MathWorks®

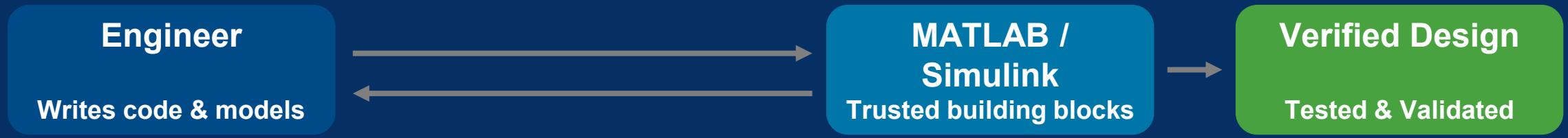
Agentic AI is transforming engineering



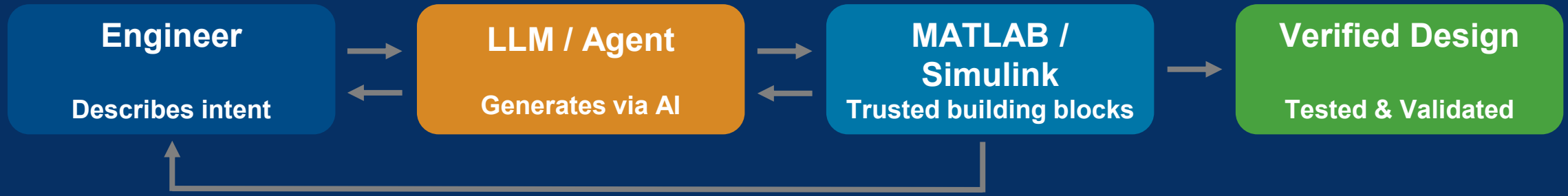
MathWorks®

Agentic AI changes how you practice Model-Based Design

Traditional



With AI



What doesn't change

The engineer owns the design. Engineering evidence verifies it.

MathWorks®

Agenda

- Model-Based Design with agentic AI
- **A two axis-architecture for interoperability**
- Simulink Agentic Toolkit: An intro
- Case study – battery fault protection
- Summary

MathWorks®

The two-axis interoperability with agentic Model-Based Design

Need 1

Agent ↔ Tool

How does the agent actually do the work?

- Modeling, simulation, and verification happen inside engineering tools
- Each tool has its own API and conventions
- Without a common interface, every tool needs a bespoke adapter

Need 2

Model ↔ Model

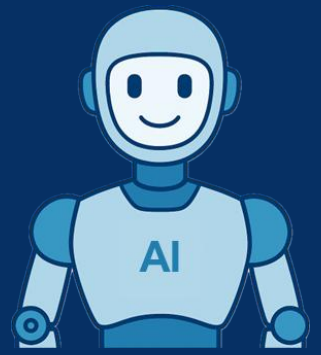
Whose models is it working with?

- System models mix components from suppliers, teams, and other environments
- Supplier models must stay encapsulated to protect IP
- Without a common model interface, every boundary is translated by hand

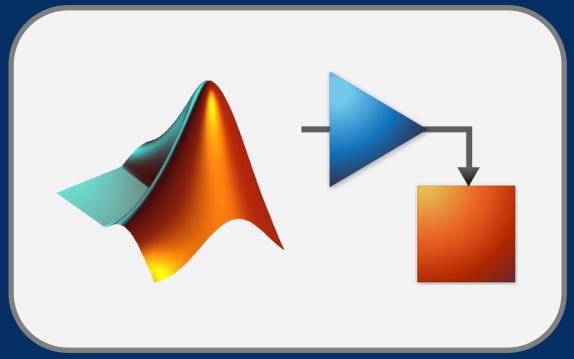
These two axes are related, but distinct. Strong agent access to one tool does not solve model integration; a well-formed multi-vendor model assembly still needs procedural orchestration to build, configure, simulate, test, and package.

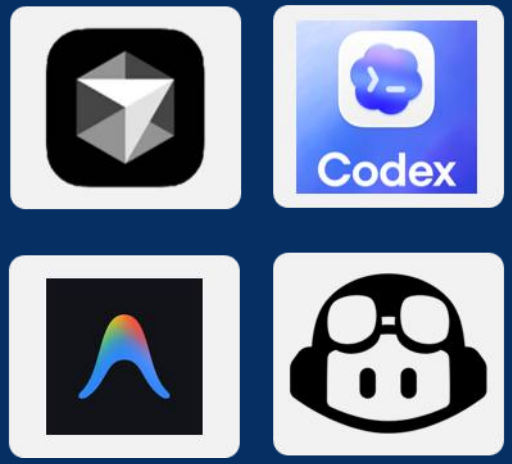
MathWorks®

Model Context Protocol (MCP) connects engineering tools with AI Agent





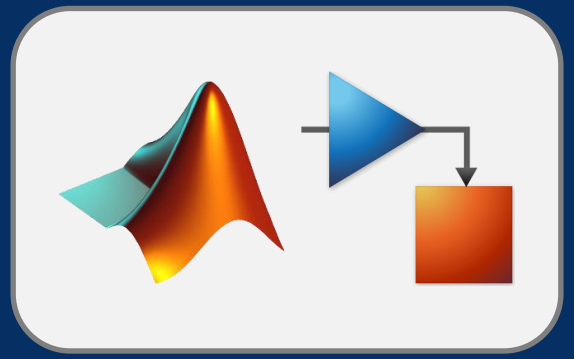




Model-Based Design Tools

MathWorks

FMI Standard connects toolchains across the Model-Based Design workflow



Functional Mockup Interface (FMI)



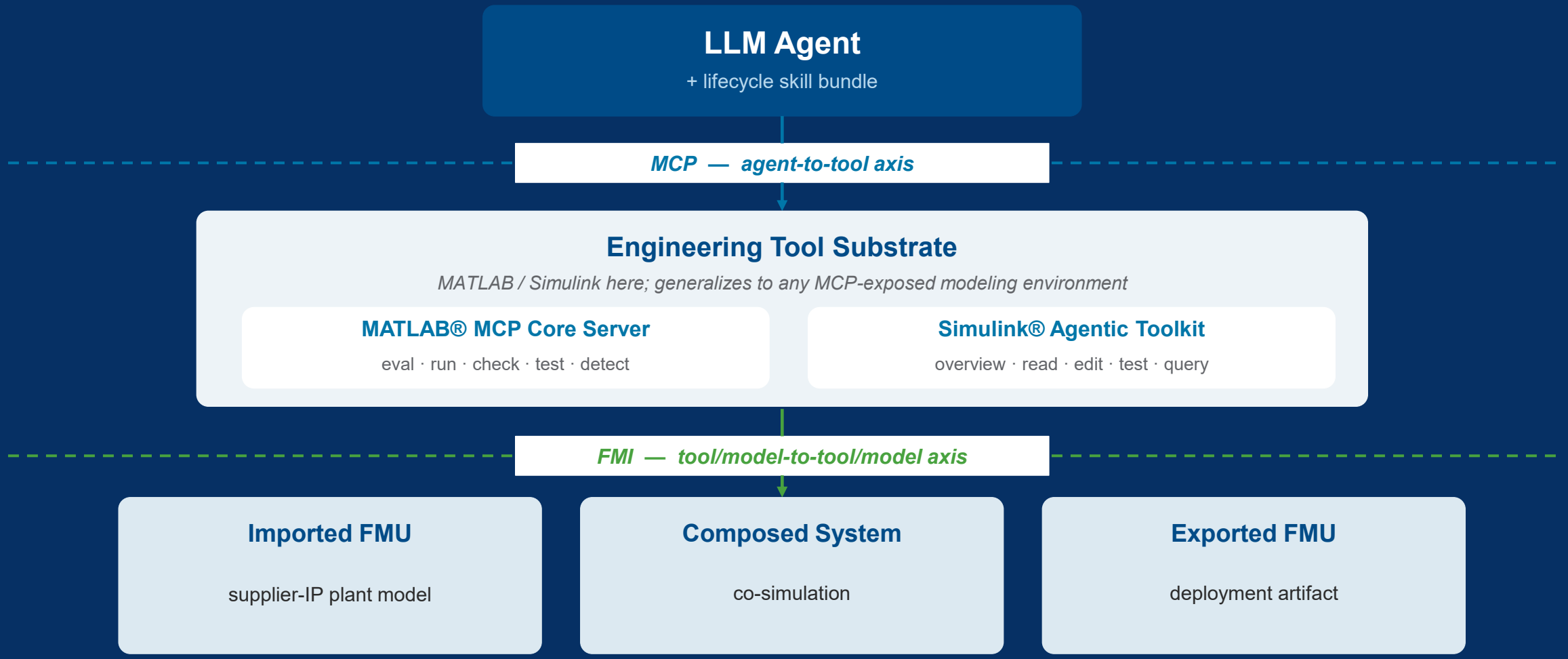


Model-based Design: Tool A

Model-based Design: Tool B

MathWorks®

Two axes, Two standards



MathWorks

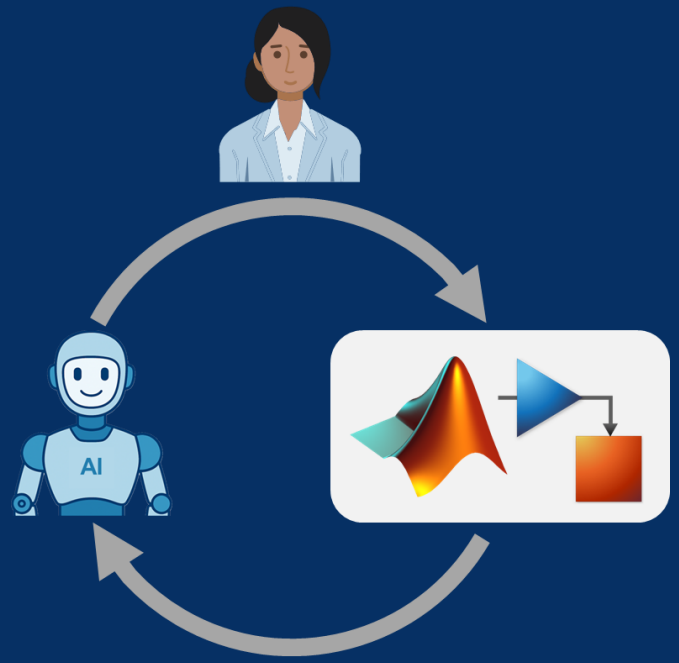
Agenda

- Model-Based Design with agentic AI
- A two axis-architecture for interoperability
- **Simulink Agentic Toolkit: An intro**
- Case study – battery fault protection
- Summary

MathWorks®

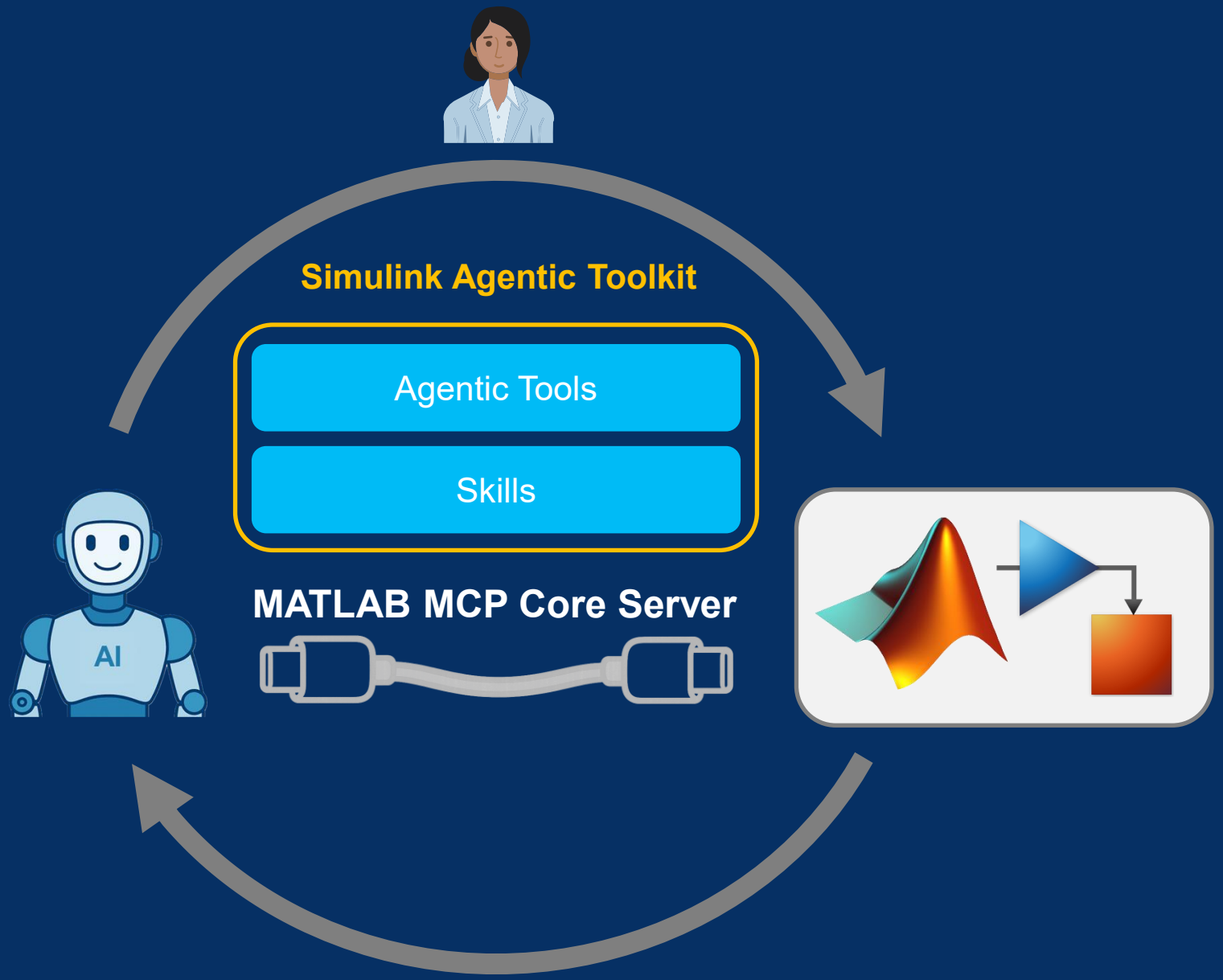
Simulink Agentic Toolkit

Enable your AI agents to work effectively with Simulink and Model-Based Design



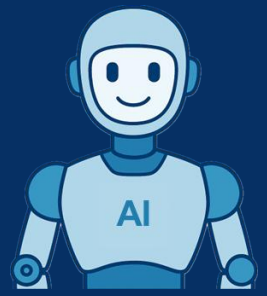
MathWorks®

Connect AI agents with specific tools and knowledge



MathWorks®

Simulink Agentic Toolkit equips agents to engineer models, not just code



AI Agent

MCP Tools

Agent Capability	Tool Name	Purpose
Understand	`model_overview`	Explore model architecture
	`model_read`	Understand model behavior
	`model_query_params`	Inspect any model parameter
	`model_resolve_params`	Resolve workspace variables
Change	`model_edit`	Build and modify models
Verify	`model_test*`	Verify models by creating & running tests



Simulink

SKILLS

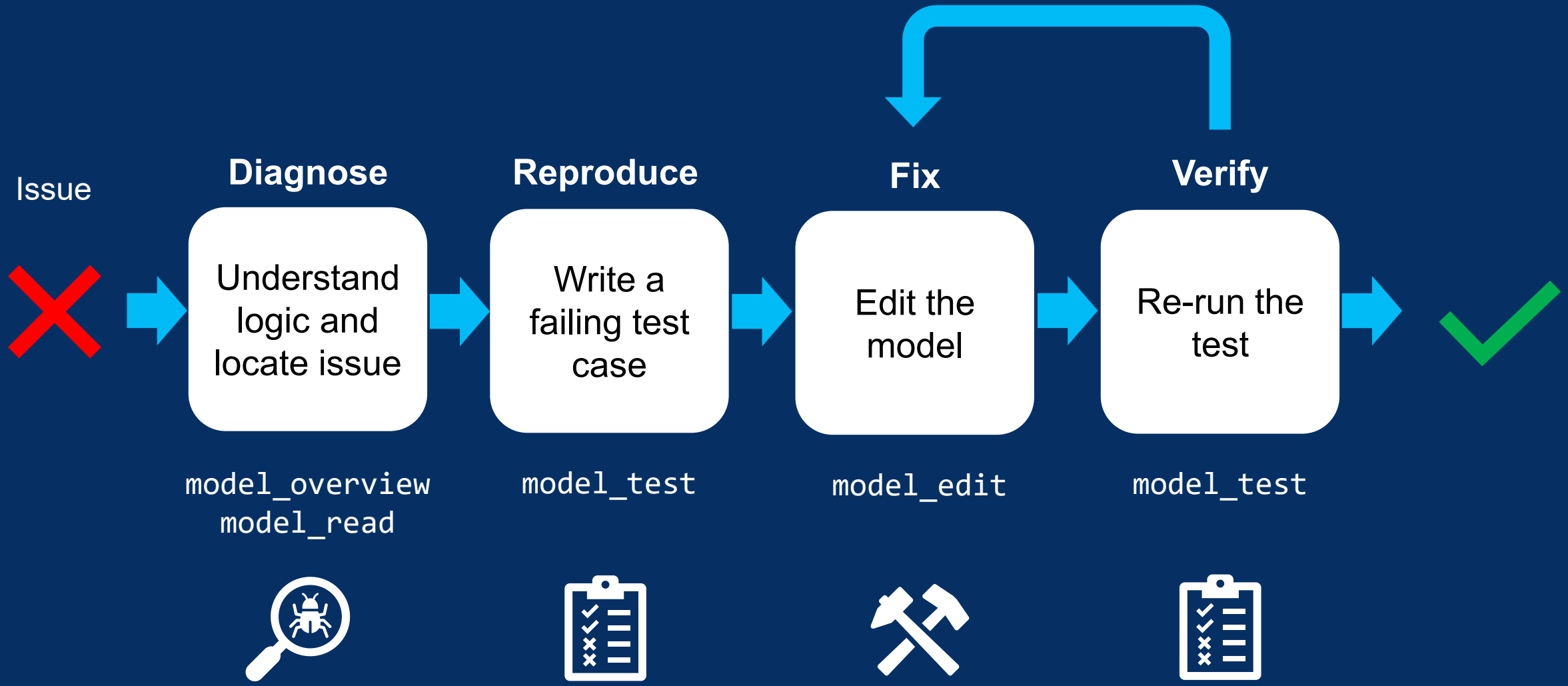
MATLAB MCP Core Server

*requires Simulink Test

MathWorks®

Agentic Toolkit enables automation of complex workflows

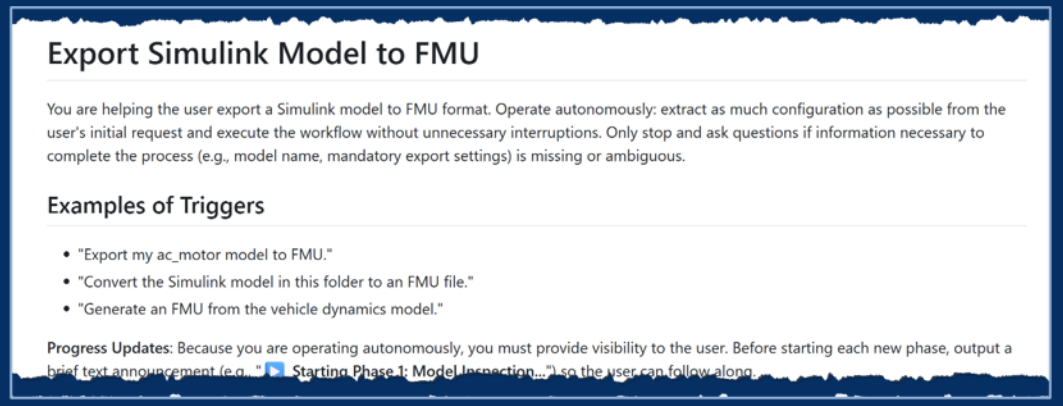
Troubleshoot – Fix – Verify



MathWorks®

Skill for Simulink FMU creation

- Accelerate your FMU building process with Agentic skill
- A MathWorks export-fmu skill is now available on GitHub:
  - https://github.com/simulink/skills/blob/main/simulink-fmu-export
- Supports a multi-phase workflow:
  - Model inspection
  - Configuration
  - FMU generation
  - Error analysis
  - Post-export validation




```
$ gemini "Export my active_suspension model to an FMU and make sure the tuning variables are exposed."

 Gemini: I will activate the 'export-fmu' skill to guide us through the export process.

[Phase 1: Inspection]
 Analyzing model structure, solver settings, and workspace variables...
 Model solver is compatible (Fixed-step).
 Identified tunable workspace variables: 'base_damping_coefficient', 'pid_proportional_gain'...

[Phase 2: Configuration]
The initial request was slightly vague. I need to clarify some specific export settings.

? Which FMI version do you want to target?
  FMI 3.0
❯ FMI 2.0

? What FMU type should be generated?
❯ Co-Simulation (CS)
  Model Exchange (ME)

? Which tuning parameters should be explicitly exposed in the FMU?
❯ base_damping_coefficient, pid_proportional_gain

? Do you want to cross-compile the FMU to include Linux binaries? (y/N)
❯ y

⚙ Applying configurations to active_suspension.slx...
 FMI version set to 2.0.
 FMU type set to Co-Simulation.
 Configured requested variables as tunable ModelArguments.

[Phase 3: Generation]
 Generating C code and building the FMU (this may take a few moments)...
 Build successful. FMU created at: active_suspension.fmu
```

Agenda

- Model-Based Design with agentic AI
- A two axis-architecture for interoperability
- Simulink Agentic Toolkit: An intro
- **Case study – battery fault protection**
- Summary

MathWorks®

BMS fault protection around a supplier battery pack

The problem

Lithium-ion packs carry real safety risk under over-charge, over-discharge, and temperature extremes. The BMS must detect faults, open the relay, derate current, and manage coolant.



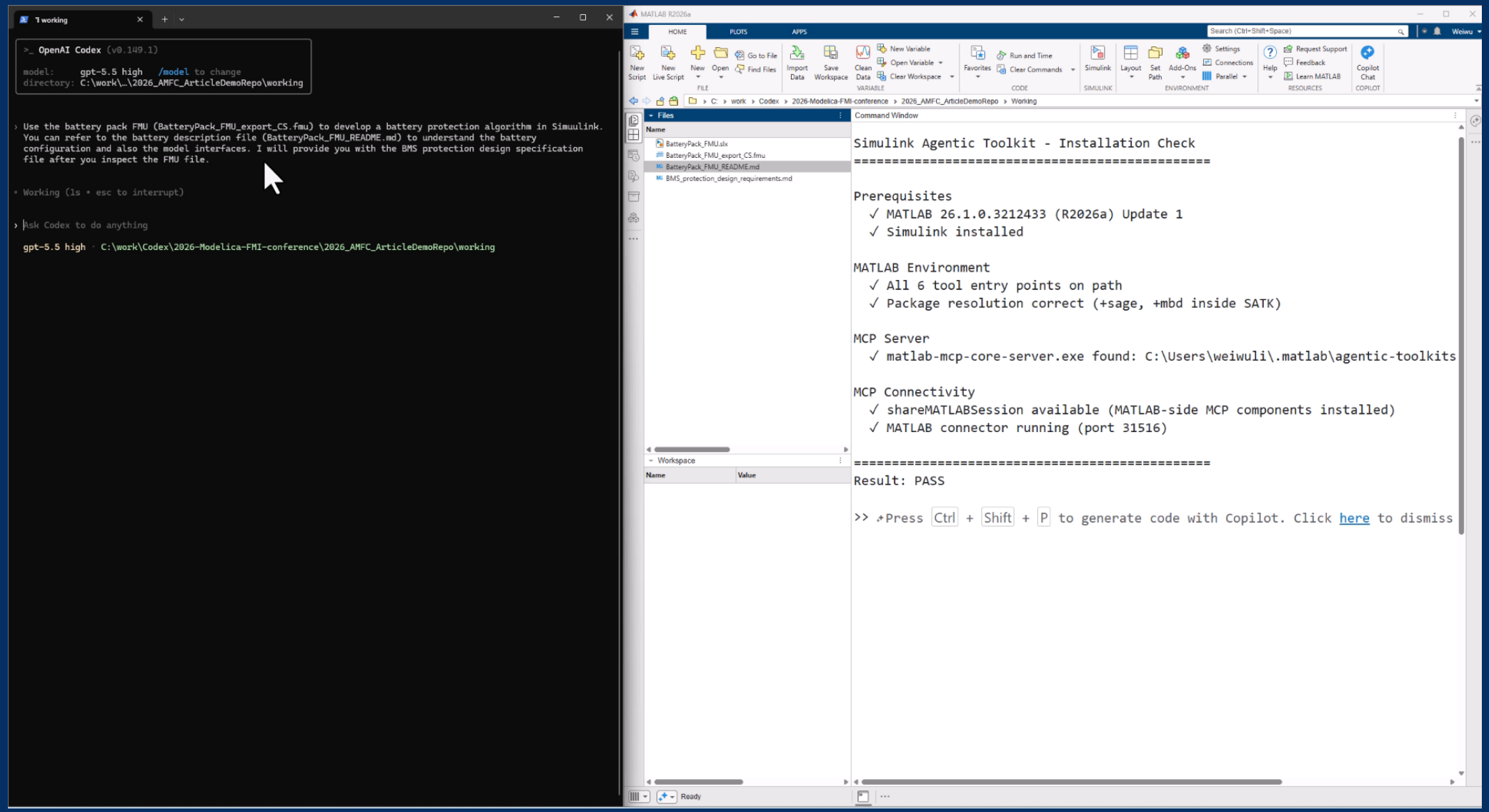
The setup

Plant: FMI 3.0 co-simulation FMU, treated as supplier IP — pack current and coolant mass-flow inputs, module-resolved outputs, tunable initial SOC and cell temperature.

Controller: built from scratch by the agent, with OpenAI Codex over the MATLAB MCP Core Server.

MathWorks®

BMS fault protection around a supplier battery pack



MathWorks

BMS fault protection around a supplier battery pack

1. **FMU import**

   Inspect modelDescription.xml; identify inputs, outputs, tunable parameters

2. **Harness construction**

   Build the Simulink harness from a human-supplied architecture sketch; condition module-resolved outputs to pack level

3. **BMS design**

   Input conditioning, current limiting, protection logic, SOC handling, thermal management

4. **Scenario verification**

   Repeatable scenario suite: nominal charge/discharge, overvoltage, thermal, current-limit clamping, fault reset

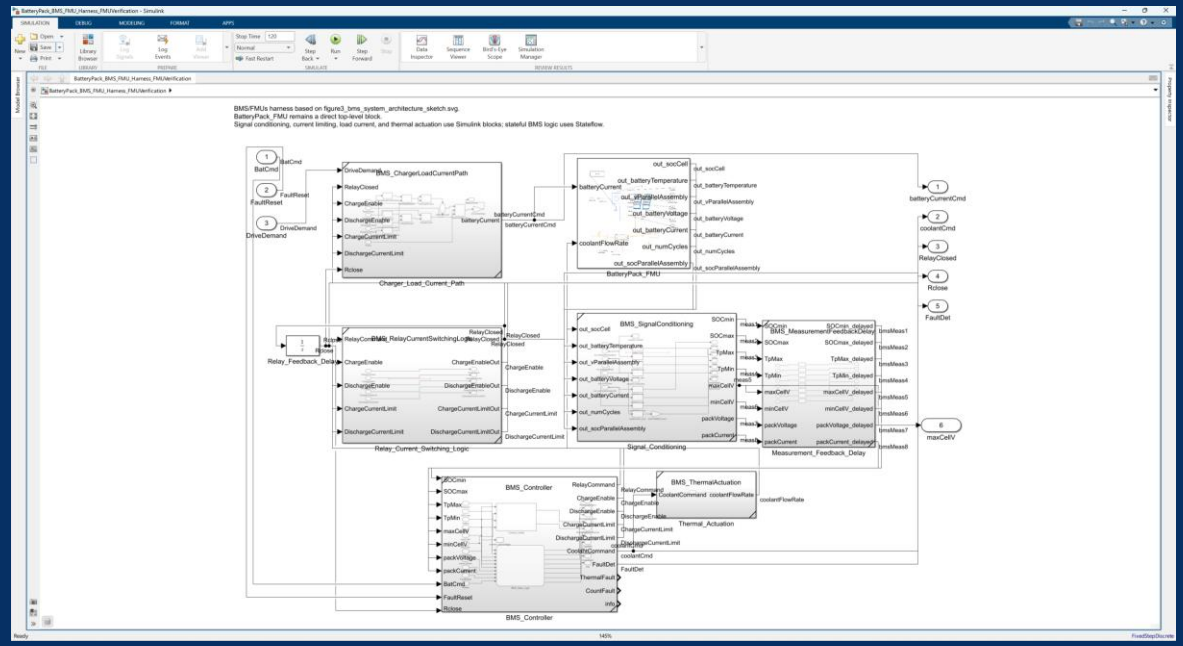
5. **FMU export and verification**

   Package the integrated system as one FMI 3.0 co-simulation FMU with a curated interface, verify outside Simulink in Python

Agentic AI could accelerate this design work by

10X

- MCP gave the agent a consistent way to reach the tool
- FMI gave the model a consistent way to cross tool and organizational boundaries



Agent-generated Simulink integration harness: supplier battery FMU, charger/load actuation, relay and current-switching logic, signal conditioning, and the BMS controller subsystem.

MathWorks

Future work

- **SSP** so the agent can compose multi-FMU systems drawing on several suppliers and toolchains — not a single plant FMU in one harness.

- **Broader deployment paths** beyond FMU-in-Python, including browser-based deployment.

MathWorks®

Agenda

- Model-Based Design with agentic AI
- A two axis-architecture for interoperability
- Simulink Agentic Toolkit: An intro
- Case study – battery fault protection
- **Summary**

MathWorks®

Key takeaway

- Agentic AI is quicky transforming Model-Based Design
- MCP and FMI standard play important roles in interoperability workflows
- Simulink Agentic Toolkit and relevant skills supports your cross-tool chain agentic design

MathWorks®