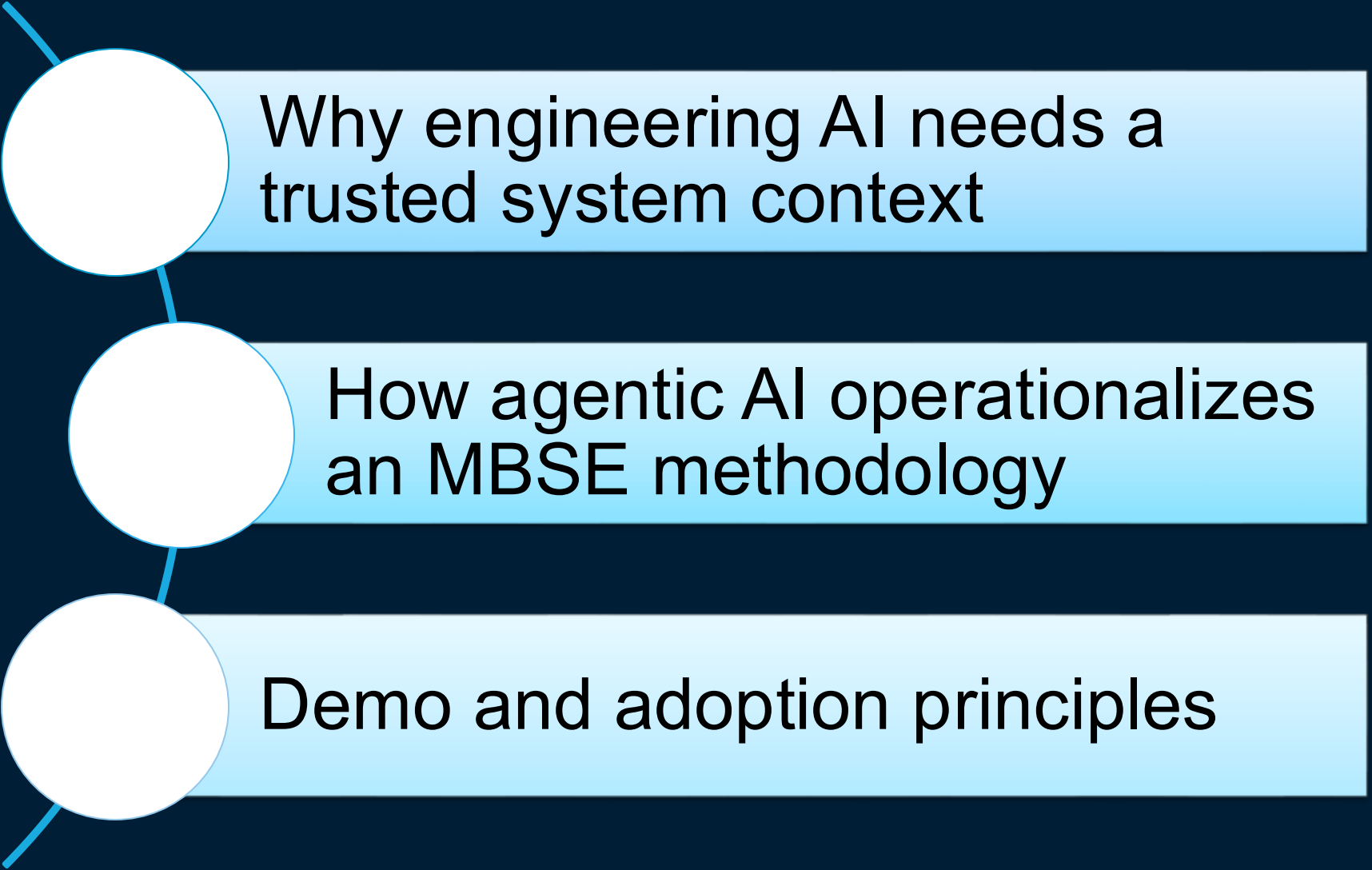


Operationalizing Agentic AI in Trusted Engineering Workflows

Massimiliano Campagnoli
Senior Application Engineer



What we will explore

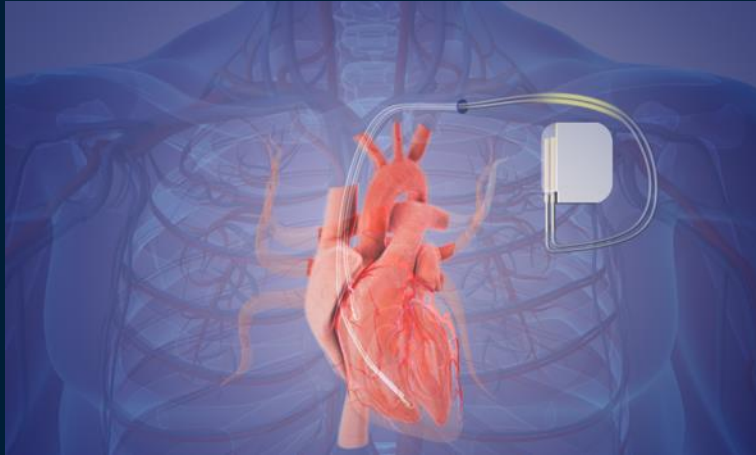


Why engineering AI needs a trusted system context

How agentic AI operationalizes an MBSE methodology

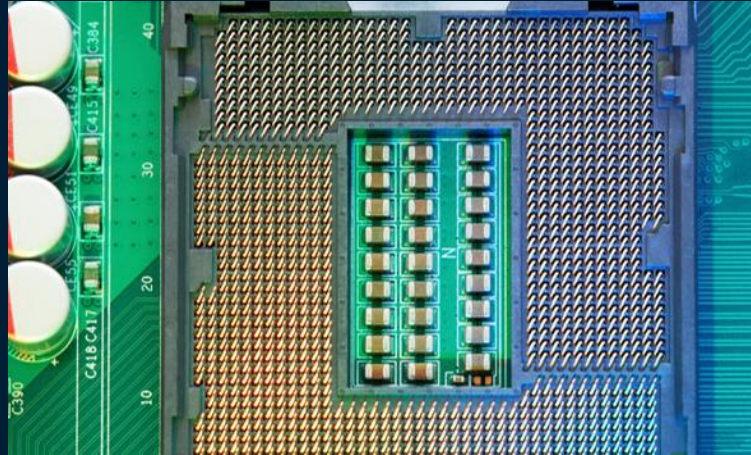
Demo and adoption principles

AI in engineering is not new. What is changing is its role.



Medtronic

Detecting heart disorders in implantable cardiac monitors



Princeton University

Creating wireless chip designs



Eaton

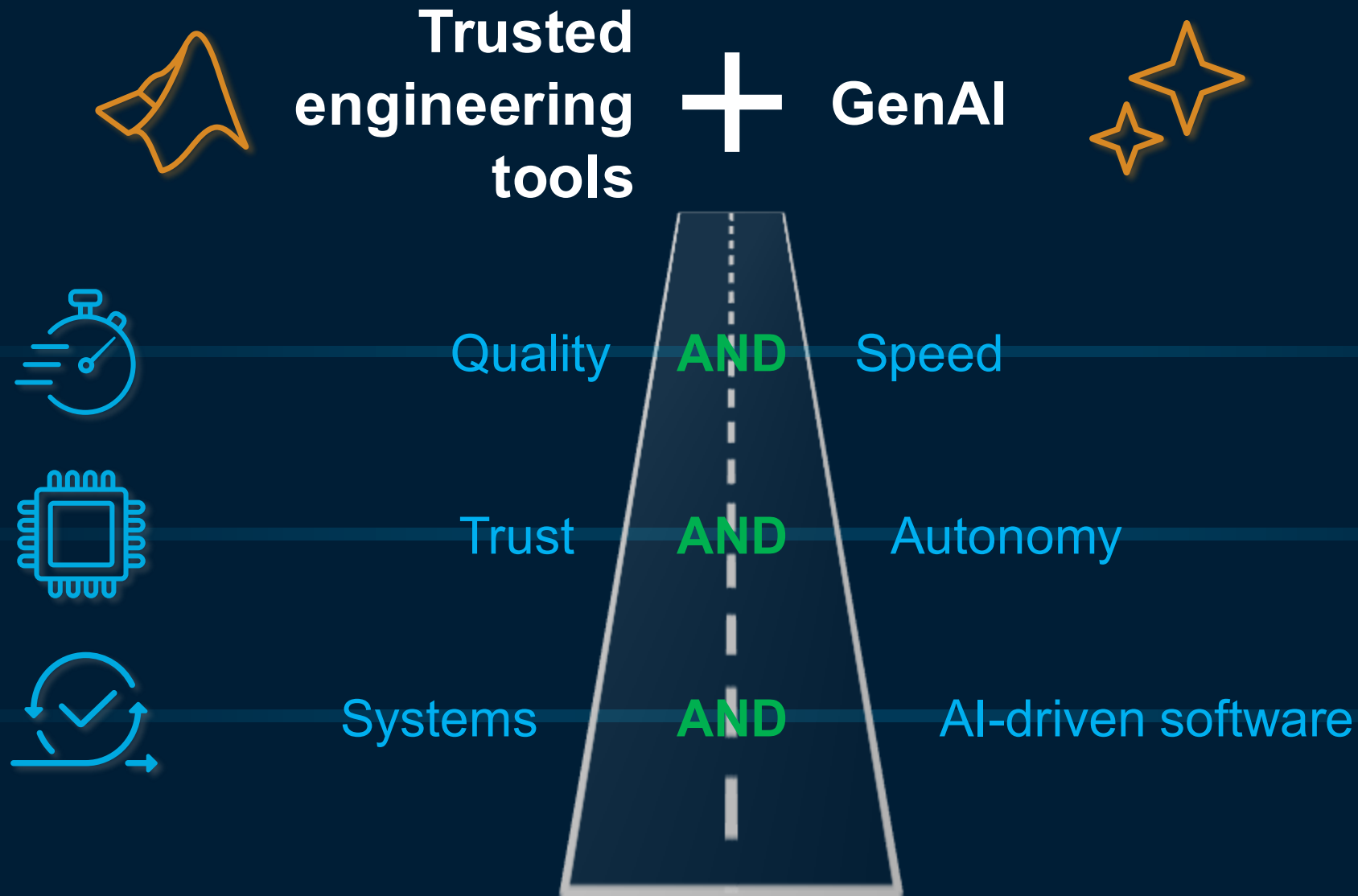
Advancing solar power systems performance

Real-world systems need verifiable engineering

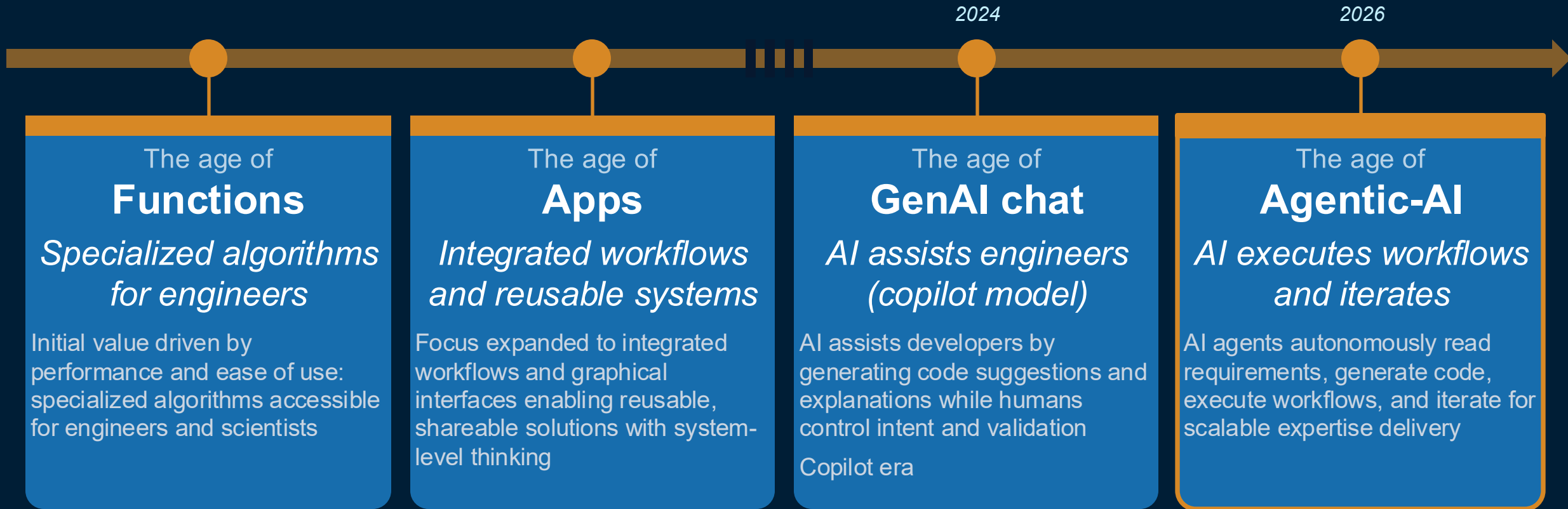


In engineering, useful AI must produce outcomes that can be inspected, tested, and traced

AI value in engineering requires a trusted foundation

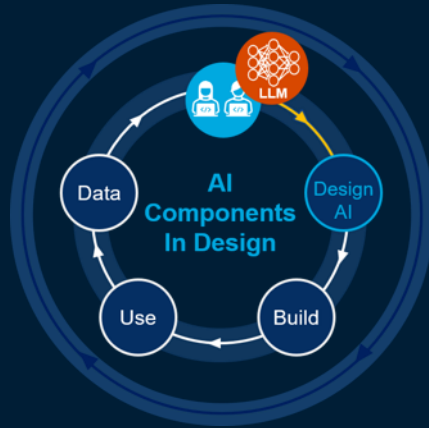


The evolution of computing eras



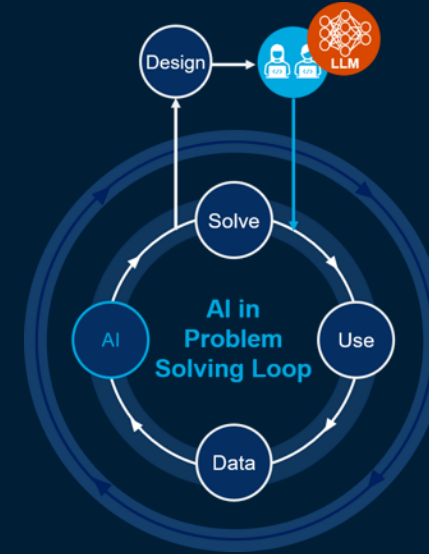
GenAI meets where you are

Two pathways to AI-augmented Model-Based Design



Integrated AI

- Embedded in the engineering environment
- Grounded assistance
- Minimal setup



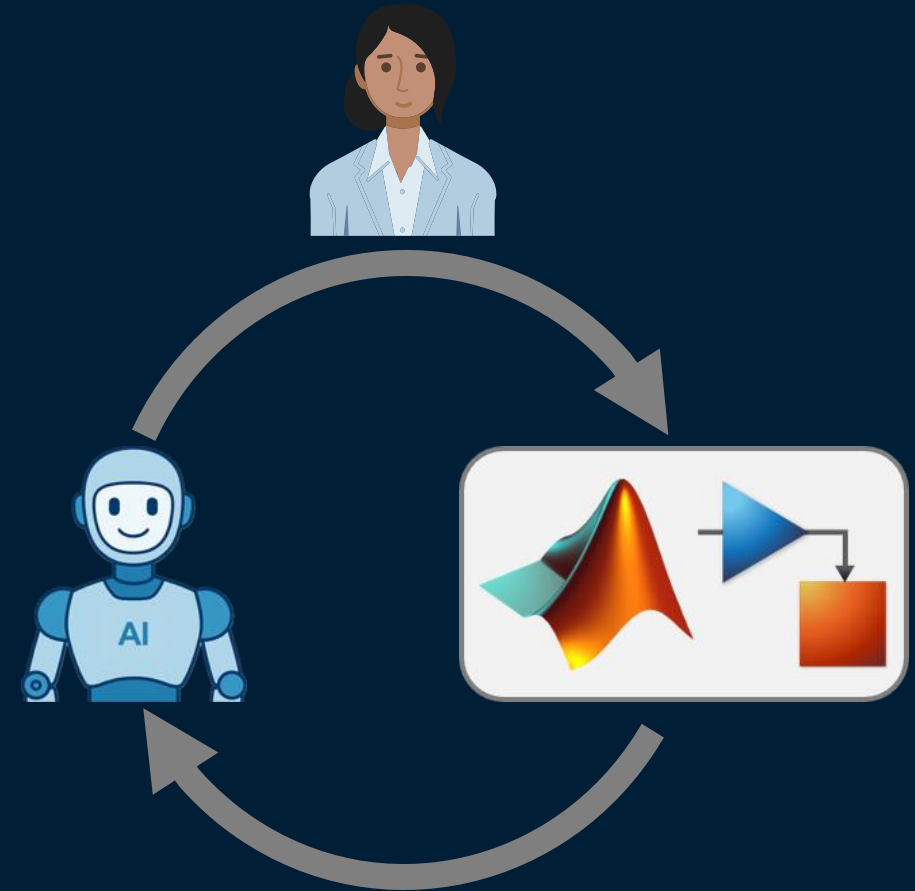
Agentic AI

- Connects external agents to engineering workflows
- Executes through tools and skills
- Flexible agent and LLM infrastructure

Simulink Agentic Toolkit

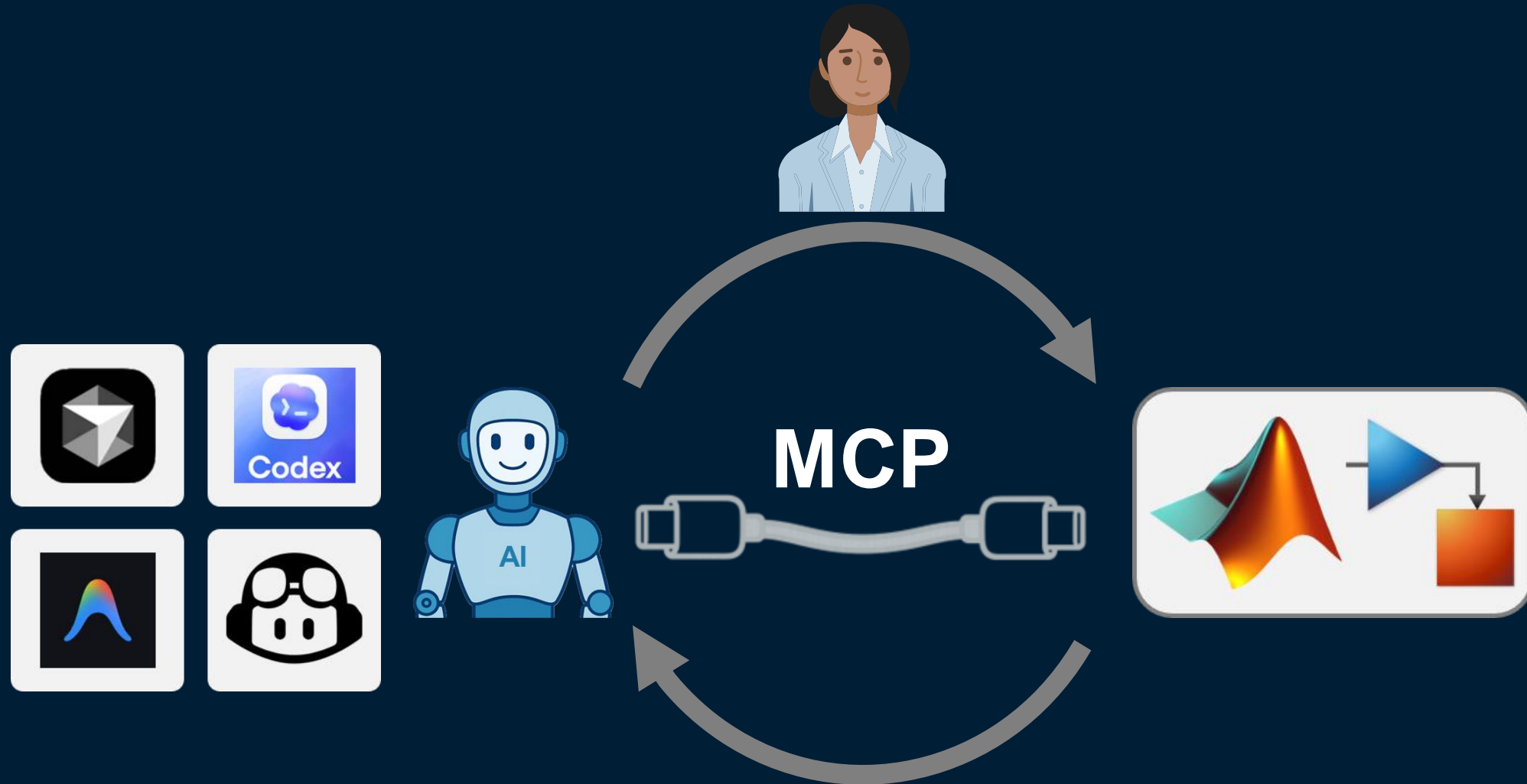
Enable your AI agents to work effectively with Simulink and Model-Based Design

- **Execution**
AI executes engineering tasks, not just suggestions
- **Orchestration**
Workflows become structured, repeatable, and scalable
- **Iteration**
Systems evolve through continuous iteration

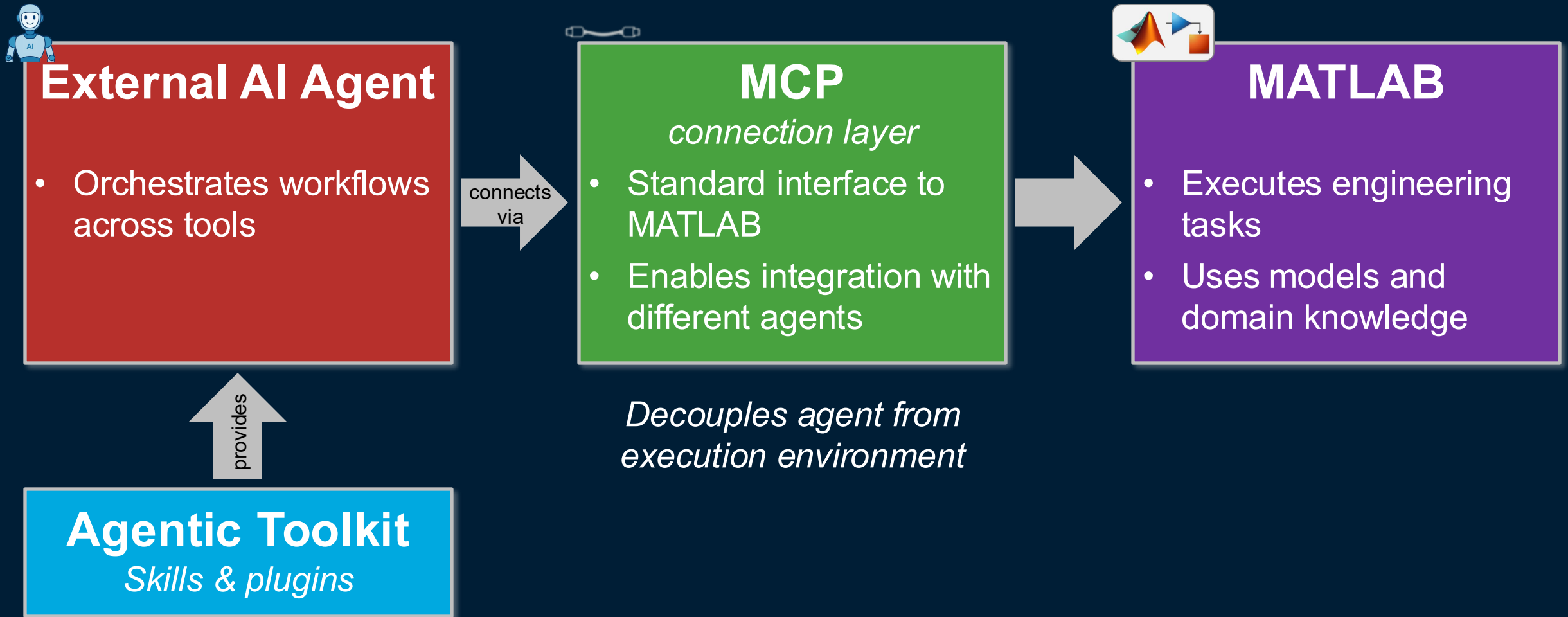


AI becomes part of the engineering loop

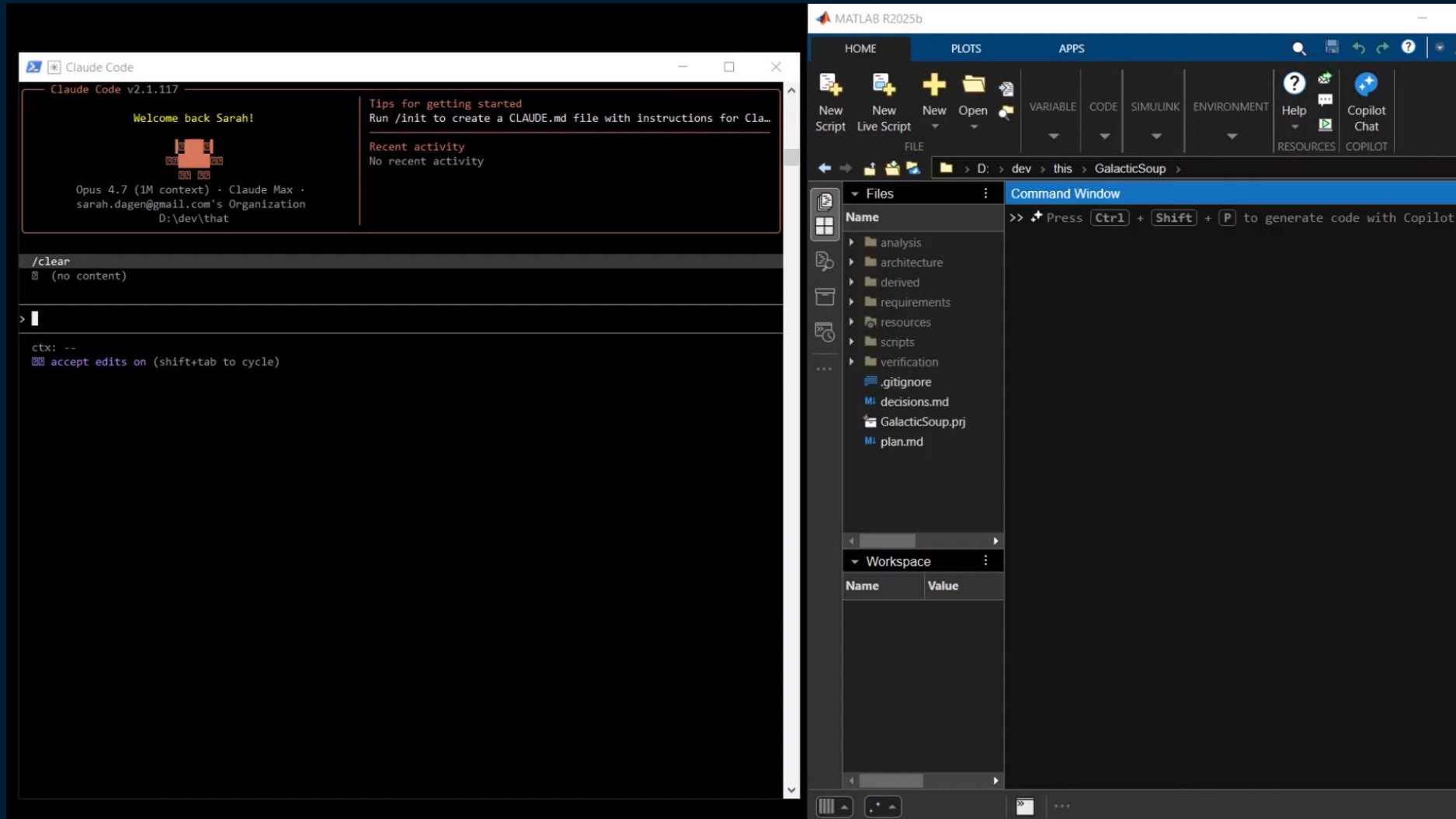
MCP enables agentic workflows



Connecting external AI to MATLAB



Demo: from stakeholder needs to traceable architecture



Model-Based Systems Engineering

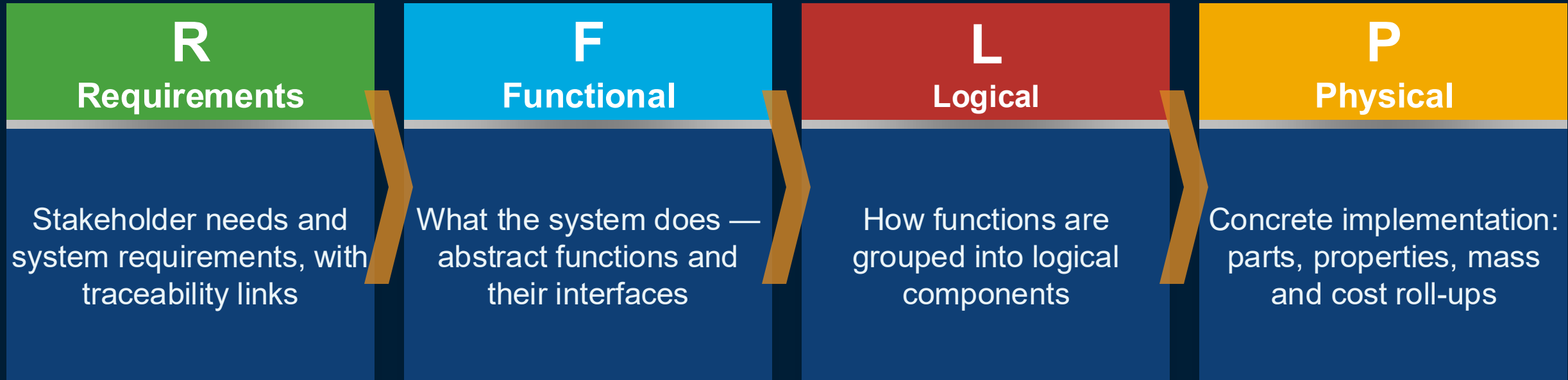
Managing system complexity with MBSE

MBSE replaces or augments document-centered workflows with models, connecting what you already do in Simulink into a system-level picture

See the whole system	Single source of truth	Traceability across the lifecycle	Earlier system-level validation
• Capture architecture, interfaces, and interactions beyond individual Simulink models • Avoid integration surprises later	• Requirements, architecture, and design all connected • Reduces inconsistencies and misalignment	• Requirement → architecture → behavior → test • Faster impact analysis and change management	• Simulate system behavior in context (not just component behavior) • Catch issues when they're cheap to fix

Methodology-based workflow orchestration

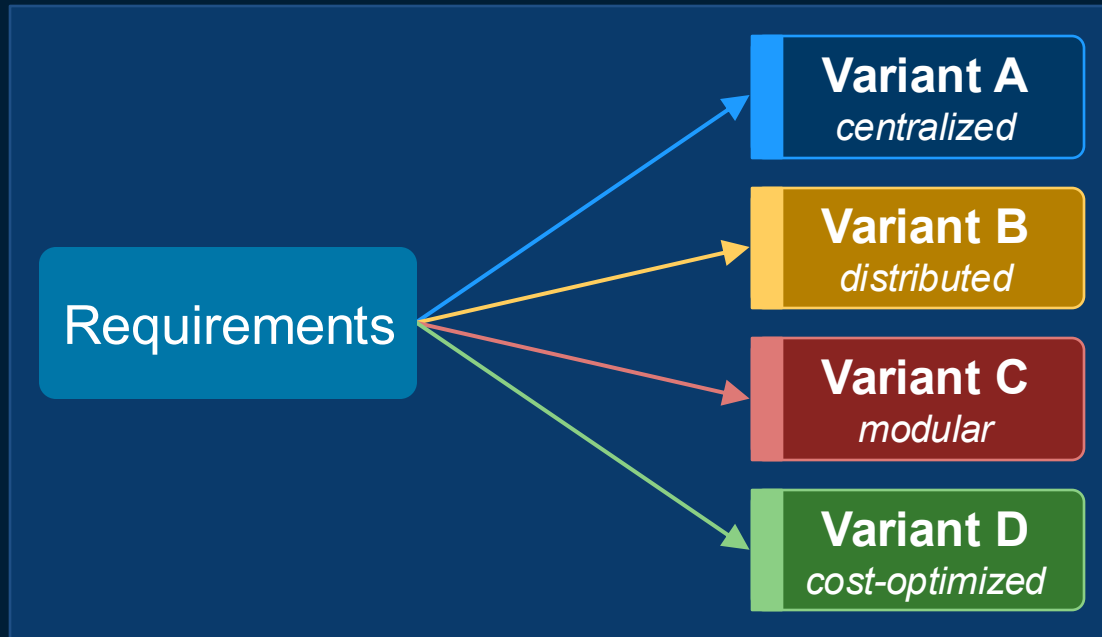
A staged approach that turns intent into traceable architecture



The value is not faster artifact generation. It is faster, traceable exploration before commitment

Why this matters

Where agentic AI creates early value



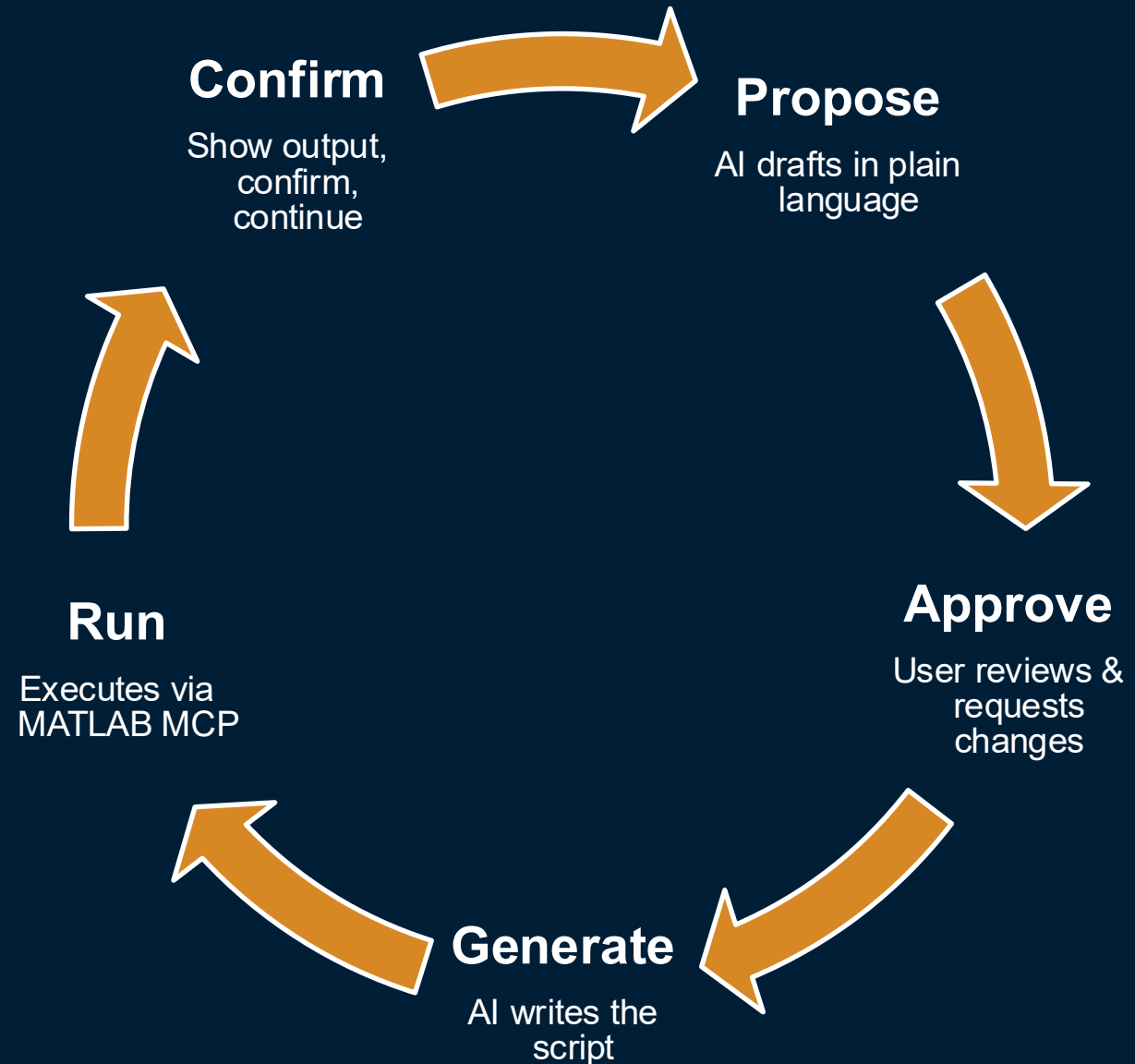
- **Explore architectural alternatives**
One requirements set → multiple design options
- **Expose requirement ambiguities earlier**
Questions emerge before design reviews
- **Reduce the cost of decisions**
Evaluate alternatives before committing

How it's built

Five skills, one repeating loop

SKILLS

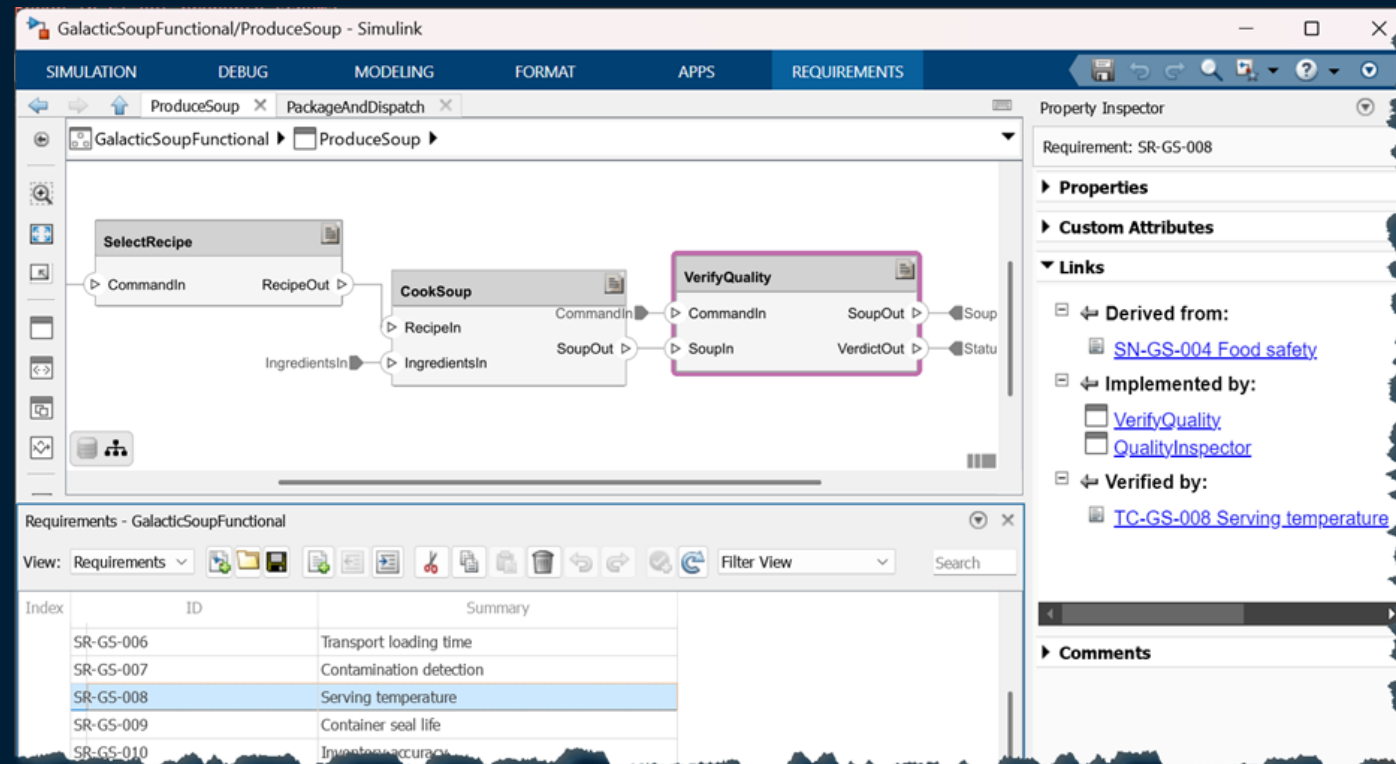
mbse-workflow	Guided end-to-end RFLP workflow
matlab-project	Project structure, APIs, integrity checks
system-composer	System Composer API reference
mbse-architecture	F/L/P, interfaces, stereotypes, allocations
simulink-requirements	Requirements Toolbox API reference



Traceability

Stakeholder needs in, traceable system architectures out

	C	D
1	Description	Rationale
2	System shall cook at least 8 distinct recipes selectable at runtime.	Eight recipes gives menu variety while staying within ingredient
3	System shall sustain total throughput of at least 200 bowls/hour.	200 bowls/hour matches expected peak order rate aggregated
4	System shall achieve average automation level of at least 0.8 across components.	0.8 automation lets the 5-operator crew cover a 24/7 facility with
5	System shall require at most 5 concurrent operators at peak load.	Facility berthing and life support are sized for a 5-person crew
6	System shall generate a shipping manifest per batch including destination.	Manifest is required for customs, transport routing, and recipe
7	System shall load packaged soup onto transport within 10 minutes of packaging.	10-minute window keeps loading pipeline aligned with the 200



Artifacts

Analytical outputs

ALLOCATIONS

Allocation Set Browser

LogicalToPhysical

GalacticSoupFunctionalToLogical

FunctionalToLogical

GalacticSoupLogicalToPhysical

LogicalToPhysical

alacticSoupPhysical

ControlCenter

ComputeCluster

OperatorConsole

CookingFacility

CookingStations

QualitySensors

IngredientStorage

AmbientStorage

ColdStorage

MaterialTransport

PackagingLine

FillingMachines

ControlAndSupe

OperatorInter

ProductionCc

CookingSystem

CookingUnit

QualityInspe

RecipeContr

IngredientHandl

InventoryMar

MaterialHanc

StorageSubs

PackagingAndLc

LoadingHanc

Allocations

GalacticSoupPhysical - Architecture Views Gallery

VIEWS

View Browser

Search

Views

CostDrivers

HighPowerConsum

MassDrivers

Chassis

ColdStorage

CookingStations

DeliveryRockets

DockingBays

Sequence Diagrams

Component Properties

Name

Value

Main

Stereotype

Add...

PhysicalProper

Change

Mass_kg

2200

Power_kW

75

Cost_credits

160000

GravityToleran

12

Parameters

Analysis Viewer

HOME

Instances

GalacticSoupPhysical

ControlCenter

ComputeCluster

OperatorCon

CookingFacility

CookingStati

QualitySensors

IngredientStorage

AmbientStorage

ColdStorage

MaterialTransport

PackagingLine

FillingMachines

LoadingCranes

SealingStations

RocketTerminal

DeliveryRockets

DockingBays

Roll-Ups

	Cost_credits	GravityTolerance_g	Mass_kg	Power_kW
GalacticSoupPhysical				
ControlCenter	130000	12	280	32
ComputeCluster	90000	12	180	28
OperatorCon		12	100	4
CookingFacility		12	1950	143
CookingStati		12	1800	140
QualitySensors	80000	12	150	3
IngredientStorage	300000	12	3900	115
AmbientStorage	40000	12	800	5
ColdStorage	160000	12	2200	75
MaterialTransport	100000	12	900	35
PackagingLine	290000	12	2250	88
FillingMachines	85000	12	500	18
LoadingCranes	140000	12	1300	55
SealingStations	65000	12	450	15
RocketTerminal	770000	15	4100	80
DeliveryRockets	450000	15	1800	40
DockingBays	220000	15	1600	25
Roll-Ups	100000	15	700	15

Views

Roll-Ups

Bootstrap with scripts – Maintain in place



Phase 1

Bootstrap

Script-driven generation

Agent writes & runs scripts that create the project, requirements, architectures, and allocations from scratch.

Fast, but every run is a full rebuild.



Transition

Steady state

Project takes shape

Architectures are reviewed, refined, and committed.

Re-running the bootstrap workflow now would clobber real work and pollute the git history.



Phase 2

Maintenance

In-situ edits via skills

Use the skills' in-place editing APIs: add a requirement, link it to this component, regenerate just the allocation matrix.

Surgical changes, clean diffs.

Where this pays off

Faster initial artifacts · Controlled evolution · Methodology compliance by design

Key takeaways



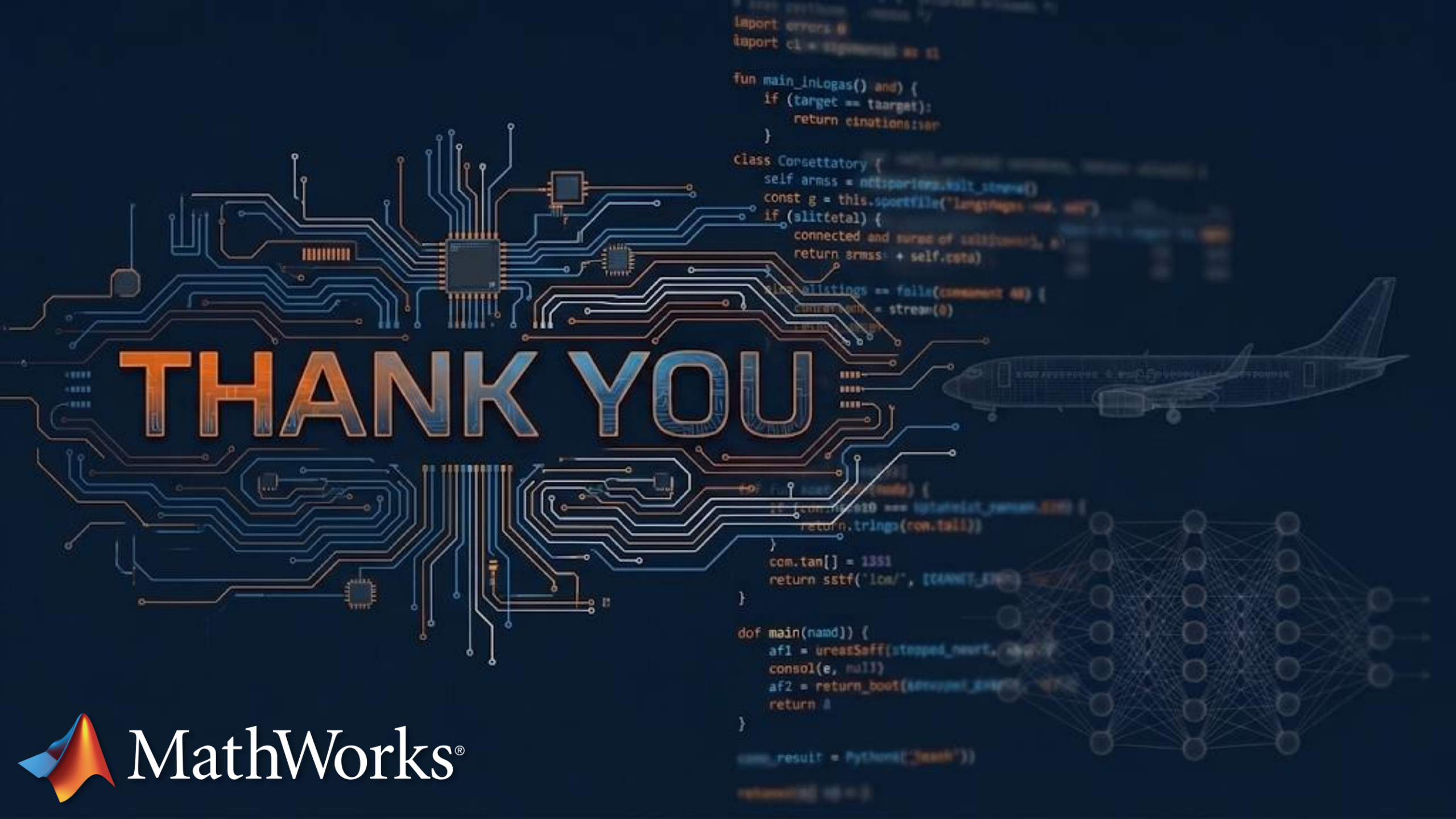
Agentic AI earns trust when it:

- Operates inside a defined engineering method
- Uses proven tools and domain-specific skills
- Keeps humans at approval and decision points
- Produces testable, traceable, reviewable artifacts
- Accelerates exploration without replacing engineering judgment

Automate the workflow. Preserve engineering authority.

Continue exploring: agentic AI for MBSE

- [Model-Based Systems Engineering and Agentic AI](#)
 - Augments systems engineers by automating modeling tasks, analyzing system architectures, and supporting informed decisions
- [Trade Studies with Agentic-AI-powered MBSE](#)
 - Automatically explores design alternatives, performs trade-off analysis, and supports data-driven engineering decisions
- [MBSE with Agentic AI on GitHub](#)
 - The most up-to-date MBSE skills



THANK YOU

```
fun main_inLogas() and) {  
    if (target == taarget):  
        return cinations:ser  
    }  
}
```

```
class Consettatory {  
    self armss = ndisporions.kalt_stream4)  
    const g = this.sportfile("longthaps -nd. 44")  
    if (alittetal) {  
        connected and surer of isstthower), a  
        return armss + self.ceta)  
    }
```

```
    else allistings == fails(comment 44) {  
        console.log = stream(0)  
        console.log = stream(0)  
    }
```

```
    if (com.tan[0] == 1351) {  
        return trings(row.tall))  
    }
```

```
    com.tan[] = 1351  
    return sstf("low/", [CANNOT_KING])  
}
```

```
dof main(namd)) {  
    af1 = ureastSaff(stopped_hwert, 44)  
    consol(e, null)  
    af2 = return_boot(sstuffed_sstuffed, 44)  
    return 4  
}
```

```
com_result = Python("3.4.4")
```

```
return 44 + 44
```

