Beyond Static Models: Verification-Based Digital Engineering

The key to employing Digital Engineering is achieving **a measure of *authoritative* virtualization that replaces, automates, or truncates formerly real-world activities***

* https://software.af.mil/wp-content/uploads/2021/05/Digital-Building-Code-and-Scorecard-Memo-v15.pdf

Agenda

- What is Digital Engineering?
- A Common Digital Transformation Starting Point
- A Common Digital Transformation
- A Verification-Based Digital Transformation with MathWorks

Agenda

- What is Digital Engineering?
- A Common Digital Transformation Starting Point
- A Common Digital Transformation
- A Verification-Based Digital Transformation with MathWorks

Agenda

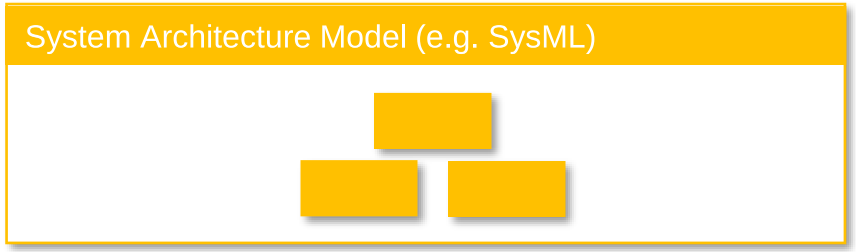
- What is Digital Engineering?
- A Common Digital Transformation Starting Point
- A Common Digital Transformation
- A Verification-Based Digital Transformation with MathWorks

Top-Down Digital Engineering

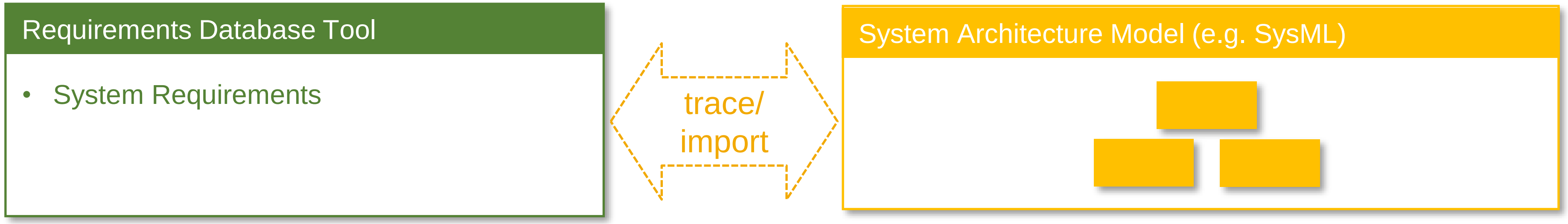
- Your customers are more often requiring delivery and interchange based on pre-defined Open System Architectures

- These architectures are often SysML models
  - Legacy architectures (paper, Visio, etc) are typically MANUALLY translated into a SysML architecture model

- These architectures serve as a top-level starting point



Requirements

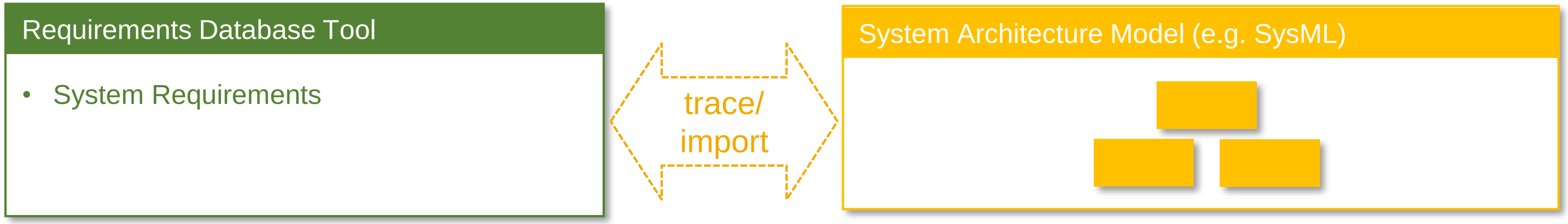

Requirements Database Tool
• System Requirements
trace/
import
System Architecture Model (e.g. SysML)

- Textual requirements continue to define desired system behavior

- Your customers often deliver a high-level requirements set (CONOPS/Mission Definition)
  - You must verify that the system satisfies these requirements!

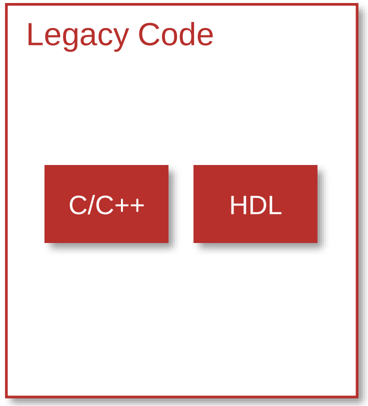
- The Digital Thread requires that these requirements be allocated to the System Architecture

Bottom Up?



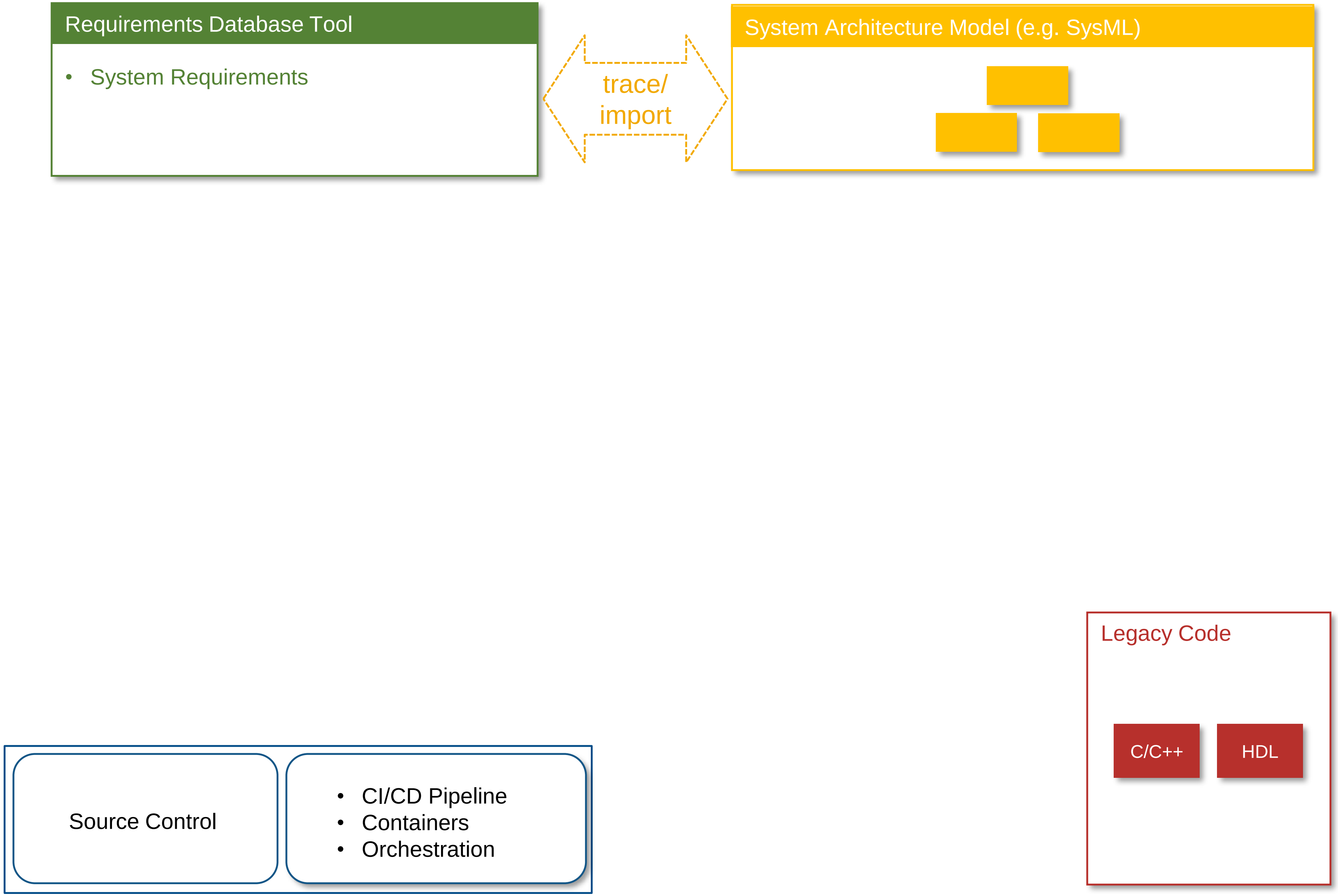
Requirements Database Tool
• System Requirements
trace/
import
System Architecture Model (e.g. SysML)

- Do you have legacy code you want to reuse?



DevSecOps?

- Configuration Management tools help manage collaboration and results
- Continuous Integration tools help automate verification and artifact generation
- Containers with the CI workflow extend verification capabilities



Requirements Database Tool

- System Requirements

trace/ import

System Architecture Model (e.g. SysML)

Legacy Code

C/C++

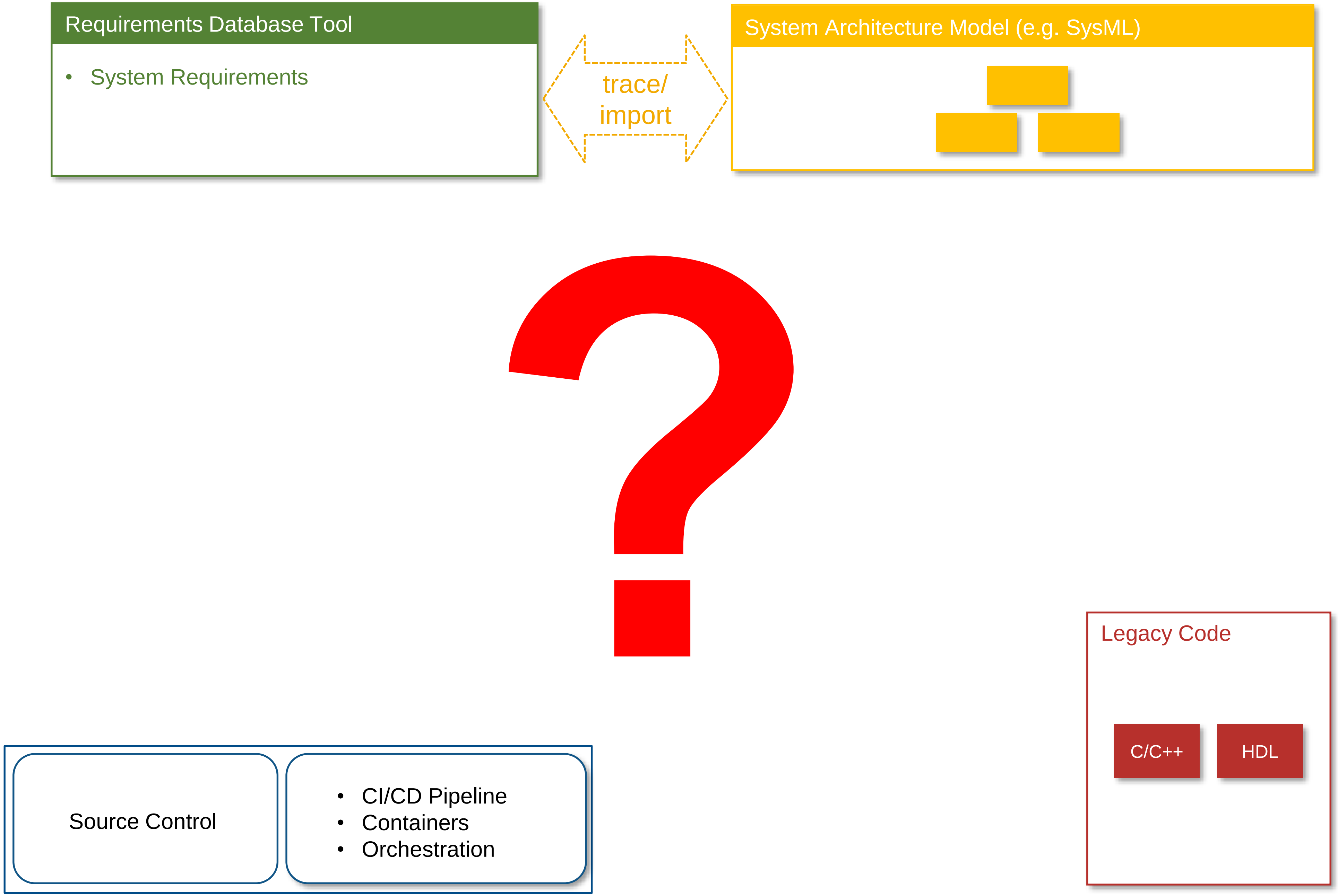
HDL

Source Control

- CI/CD Pipeline
- Containers
- Orchestration

What now?

- Digital Engineering requires that all of these tools are somehow connected to create:
  - A complete engineering process
  - Tools to enable that process
  - Artifacts to show that the process was completed successfully



Requirements Database Tool

- System Requirements

trace/ import

System Architecture Model (e.g. SysML)

Legacy Code

C/C++

HDL

Source Control

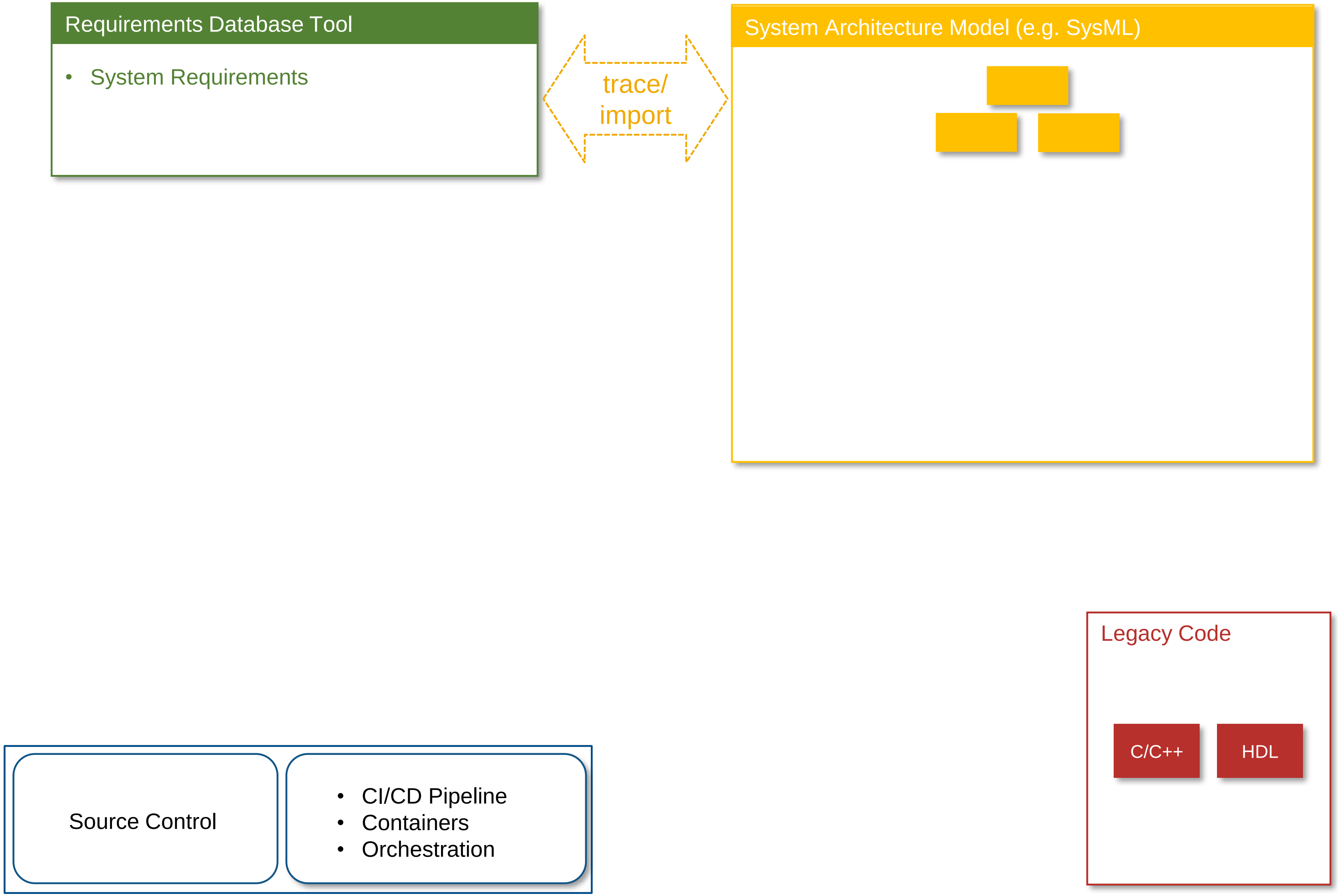
- CI/CD Pipeline
- Containers
- Orchestration

Agenda

- What is Digital Engineering?
- A Common Digital Transformation Starting Point
- A Common Digital Transformation
- A Verification-Based Digital Transformation with MathWorks

Systems & Software

- The system architecture must be elaborated until it contains component(s) allocated to software



Requirements Database Tool

- System Requirements

trace/ import

System Architecture Model (e.g. SysML)

Legacy Code

C/C++

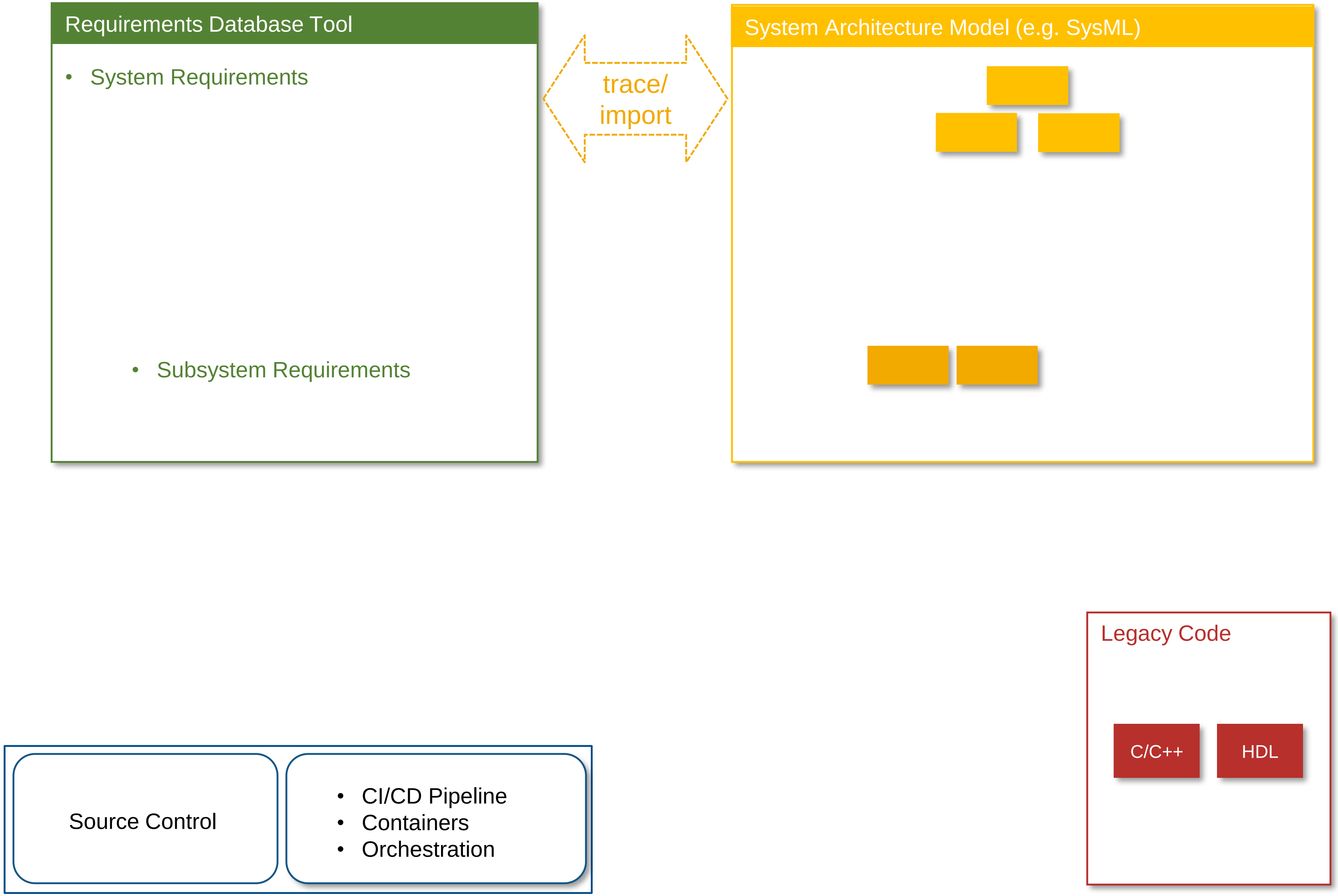
HDL

Source Control

- CI/CD Pipeline
- Containers
- Orchestration

Systems & Software

- In parallel, the Requirements must be derived, allocated, and traced to the lower-level components of the elaborated architecture



Requirements Database Tool

- System Requirements
  - Subsystem Requirements

trace/ import

System Architecture Model (e.g. SysML)

Legacy Code

C/C++

HDL

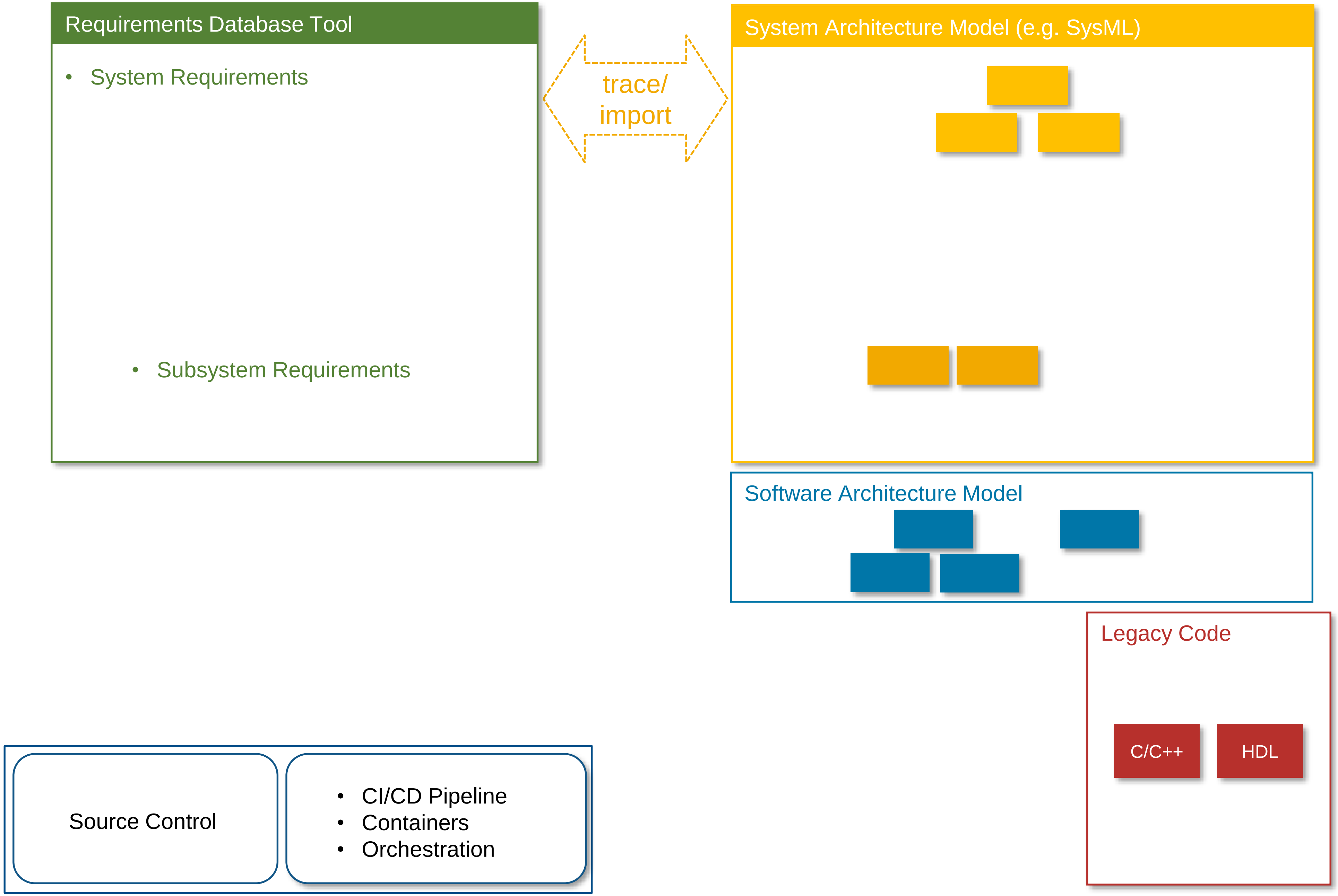
Source Control

- CI/CD Pipeline
- Containers
- Orchestration

Systems & Software

- A Software Architecture must be defined for the system component(s) allocated to software

- Some or all of this Software Architecture MUST represent the existing legacy code



Requirements Database Tool

- System Requirements
- Subsystem Requirements

trace/ import

System Architecture Model (e.g. SysML)

Software Architecture Model

Legacy Code

C/C++

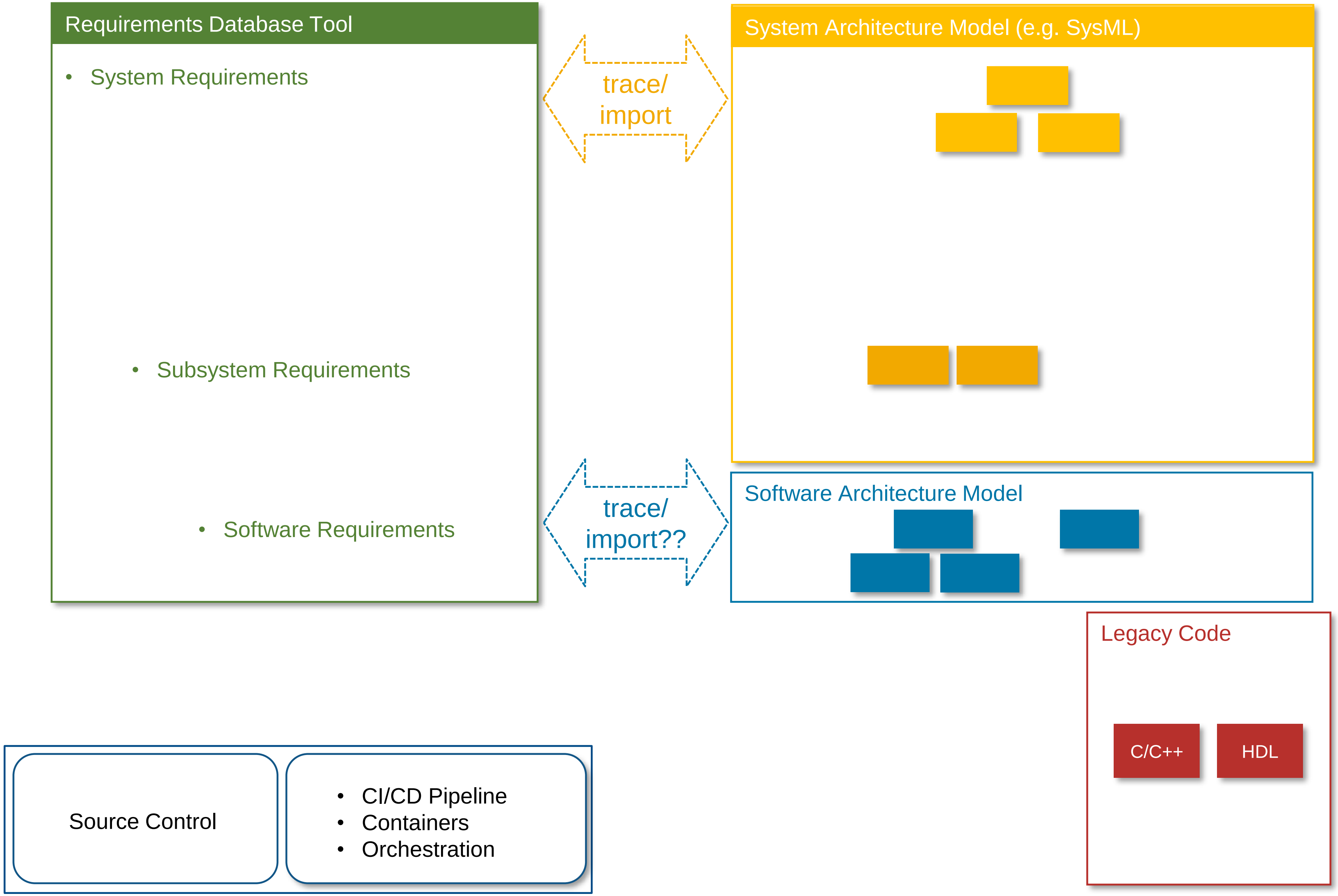
HDL

Source Control

- CI/CD Pipeline
- Containers
- Orchestration

Systems & Software

- In parallel, Software Requirements must be derived, allocated, and traced to the Software Architecture



Requirements Database Tool

- System Requirements
    - Subsystem Requirements
        - Software Requirements

trace/ import

System Architecture Model (e.g. SysML)

trace/ import??

Software Architecture Model

Legacy Code

C/C++

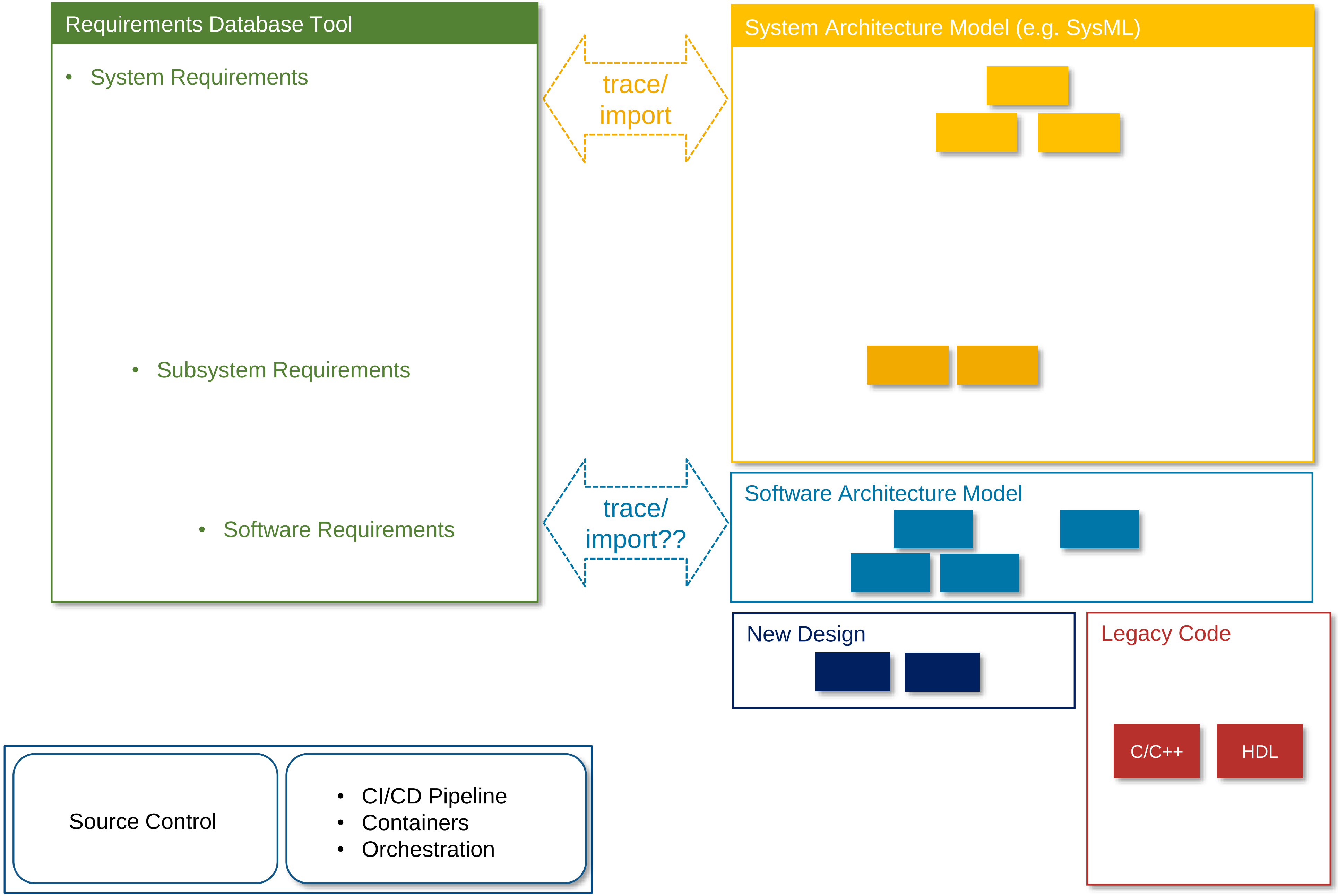
HDL

Source Control

- CI/CD Pipeline
- Containers
- Orchestration

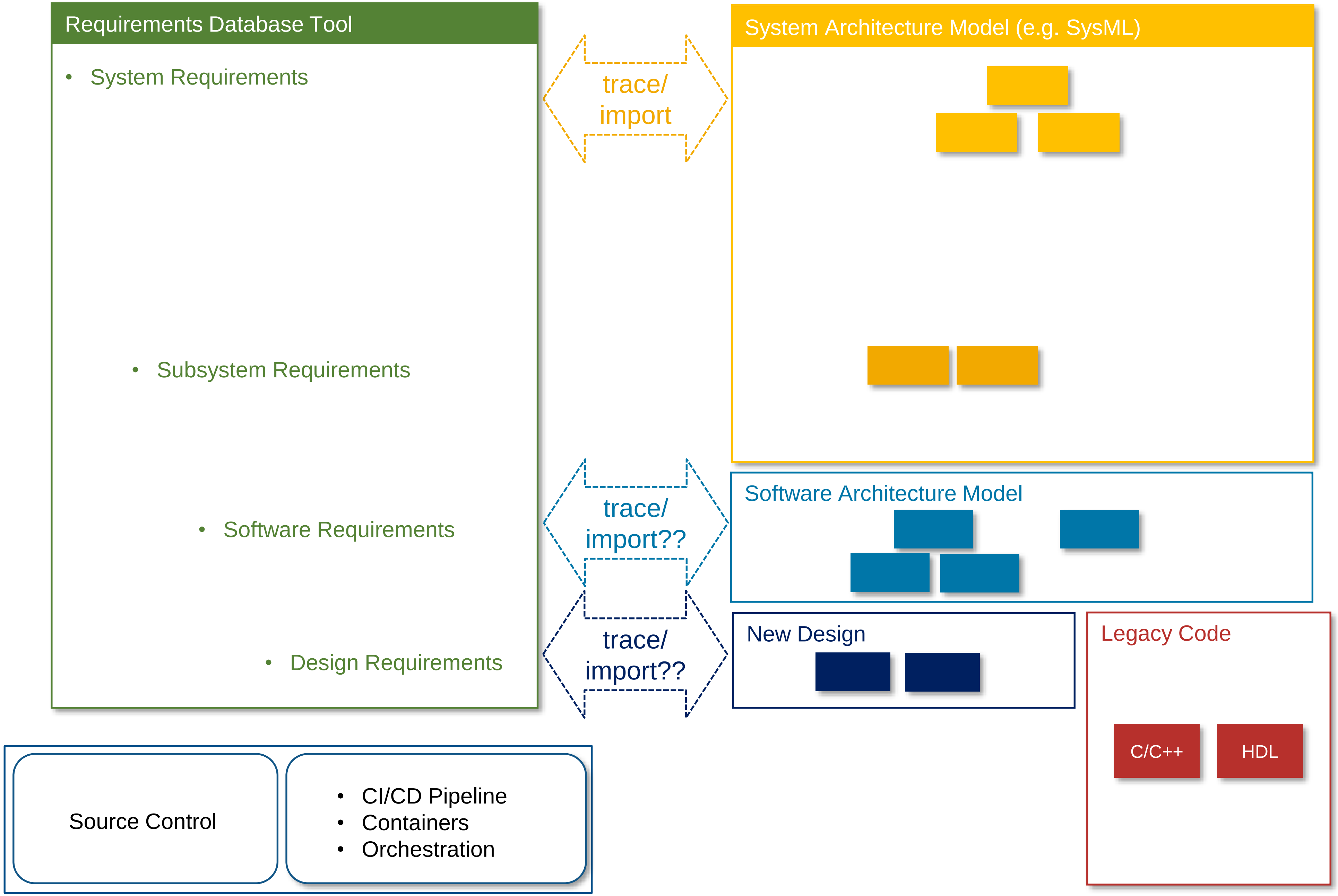
Systems & Software

- New designs must then be built around or replacing the existing legacy code



Systems & Software

- In parallel, design requirements must be derived, allocated, and traced to the new and existing code



Requirements Database Tool

- System Requirements
- Subsystem Requirements
- Software Requirements
- Design Requirements

System Architecture Model (e.g. SysML)

Software Architecture Model

New Design

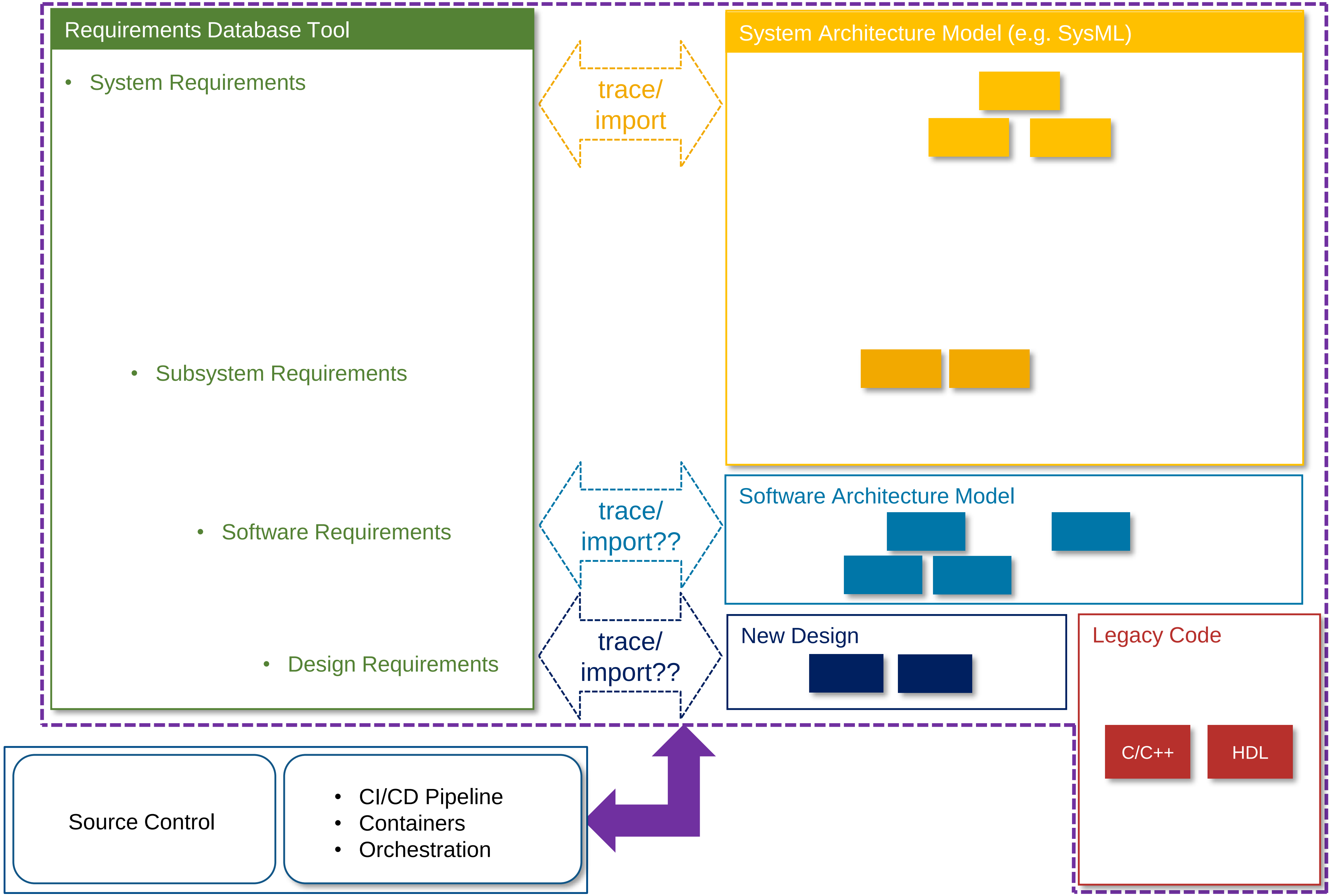
Legacy Code

Source Control

- CI/CD Pipeline
- Containers
- Orchestration

Systems & Software

- All of these must then be integrated with the CM/CI tools
- The legacy code, new design, software architecture, and system architecture must all then be VERIFIED AGAINST THEIR REQUIREMENTS!
  - Where are the test cases authored?
  - How do you verify an architecture model?

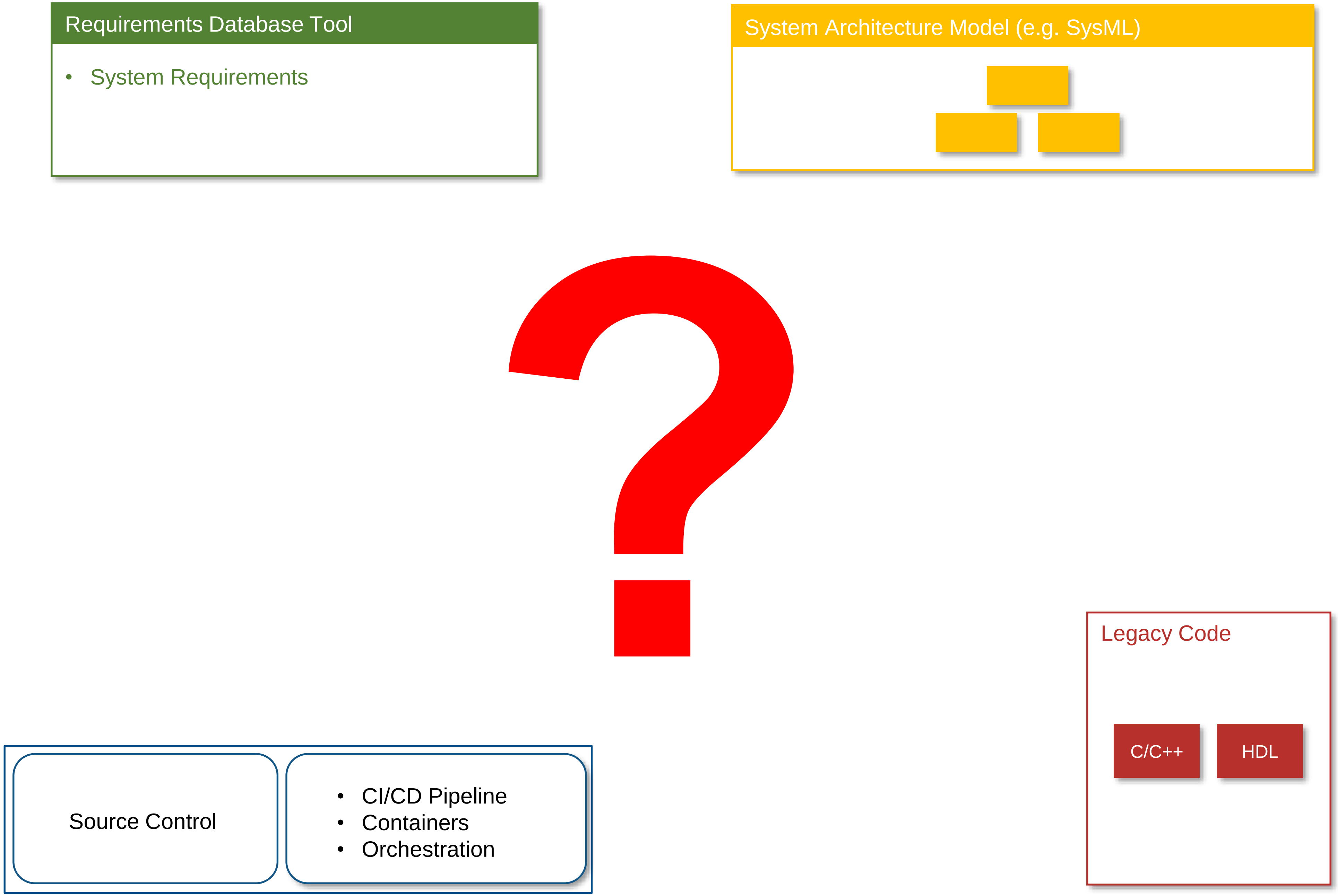


Agenda

- What is Digital Engineering?
- A Common Digital Transformation Starting Point
- A Common Digital Transformation
- A Verification-Based Digital Transformation with MathWorks

A better way?

- Is there a better way to enable the digital thread, collaboration between the engineering teams, and automated verification?



Requirements Database Tool

- System Requirements

System Architecture Model (e.g. SysML)

Legacy Code

Source Control

- CI/CD Pipeline
- Containers
- Orchestration

A better way?

- Is there a better way to enable the digital thread, collaboration between the engineering teams, and automated verification?

Requirements Database Tool

- System Requirements

System Architecture Model (e.g. SysML)

MATLAB® & SIMULINK®

Legacy Code

C/C++

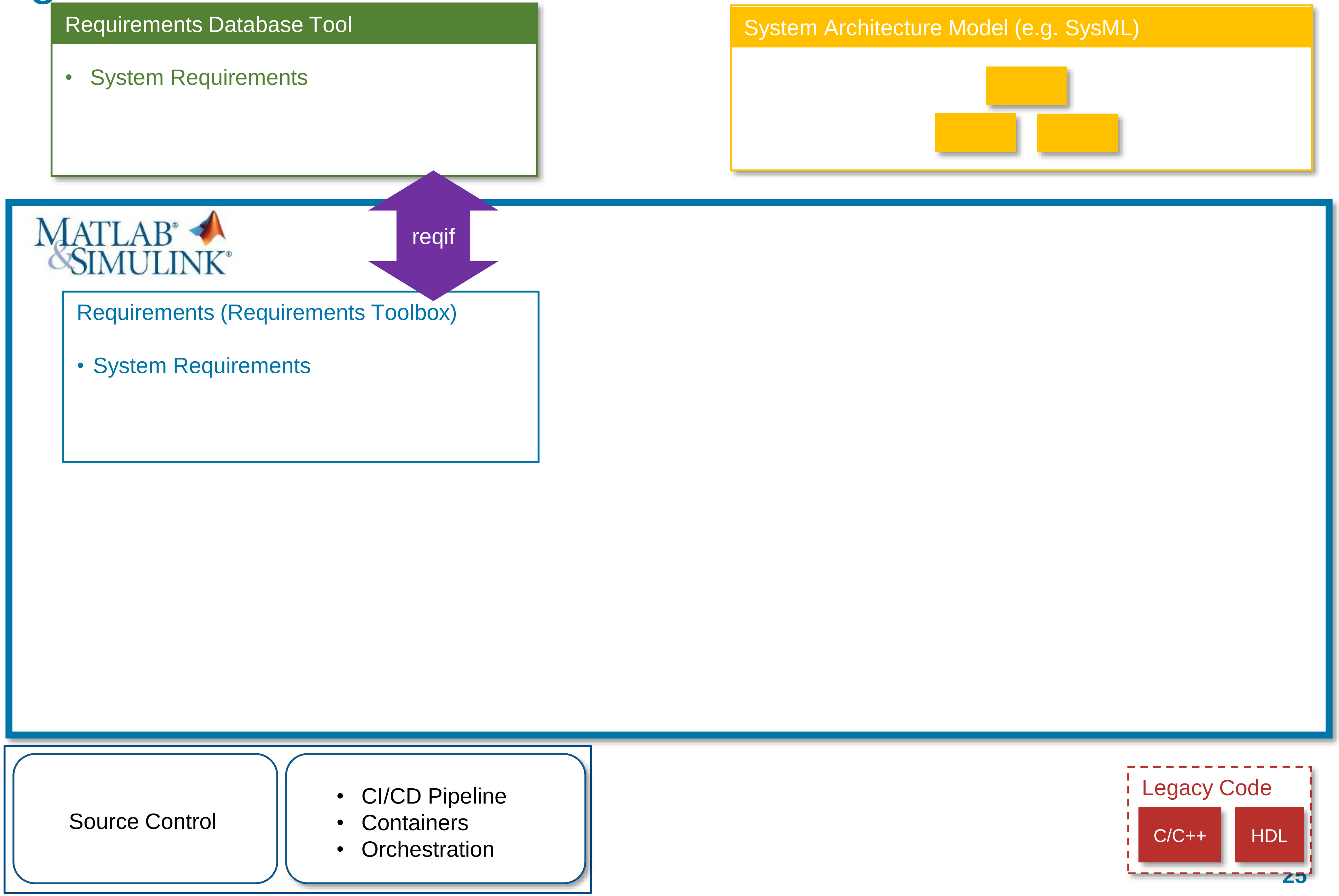
HDL

Source Control

- CI/CD Pipeline
- Containers
- Orchestration

Digital Engineering with MathWorks

- Requirements are imported into Requirements Toolbox via .reqif or direct interchange



Requirements Database Tool

- System Requirements

System Architecture Model (e.g. SysML)

reqif

MATLAB & SIMULINK

Requirements (Requirements Toolbox)

- System Requirements

Source Control

- CI/CD Pipeline
- Containers
- Orchestration

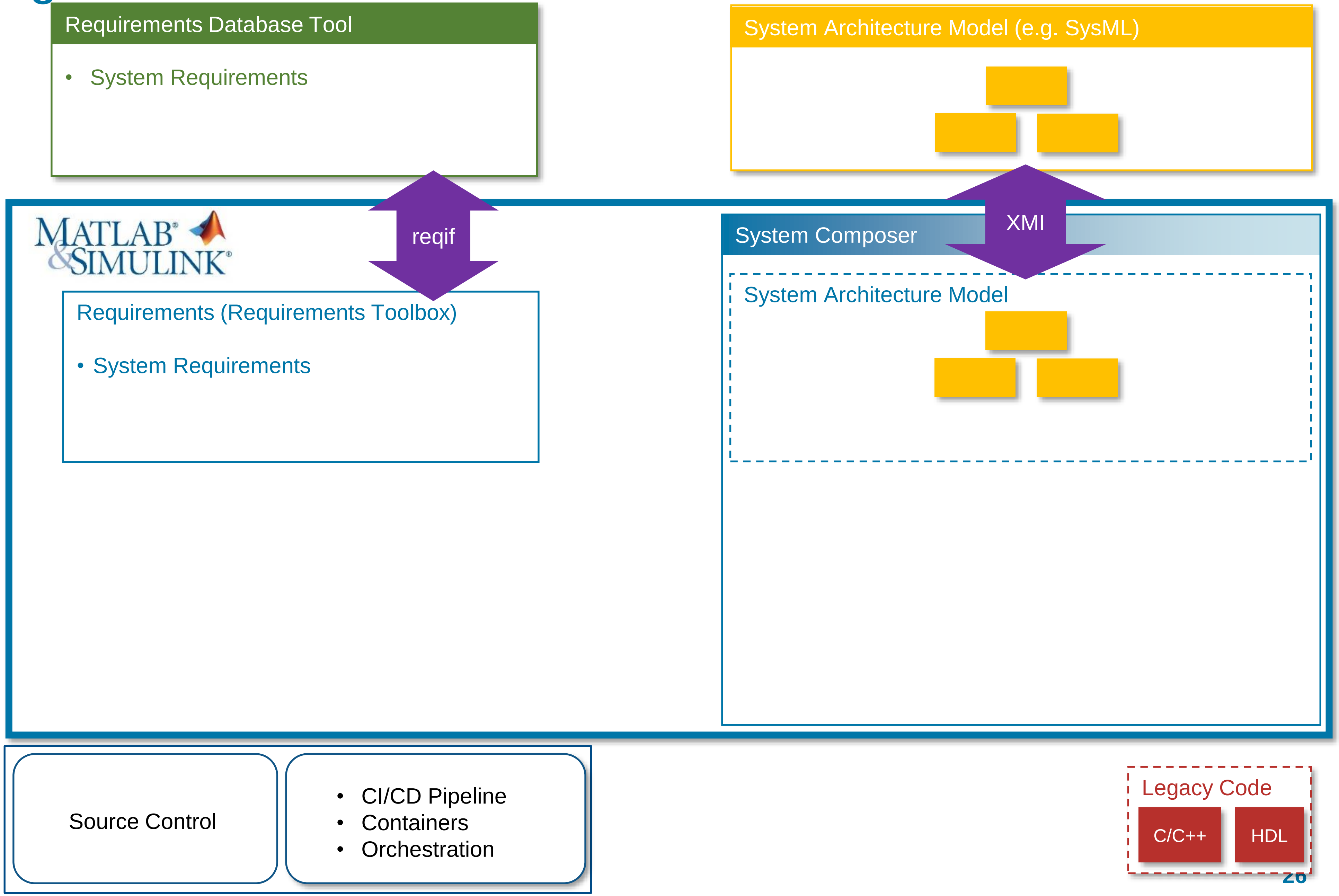
Legacy Code

C/C++

HDL

Digital Engineering with MathWorks

- SysML models are automatically imported into System Composer via the XMI interchange



Requirements Database Tool

- System Requirements

System Architecture Model (e.g. SysML)

Requirements (Requirements Toolbox)

- System Requirements

System Composer

System Architecture Model

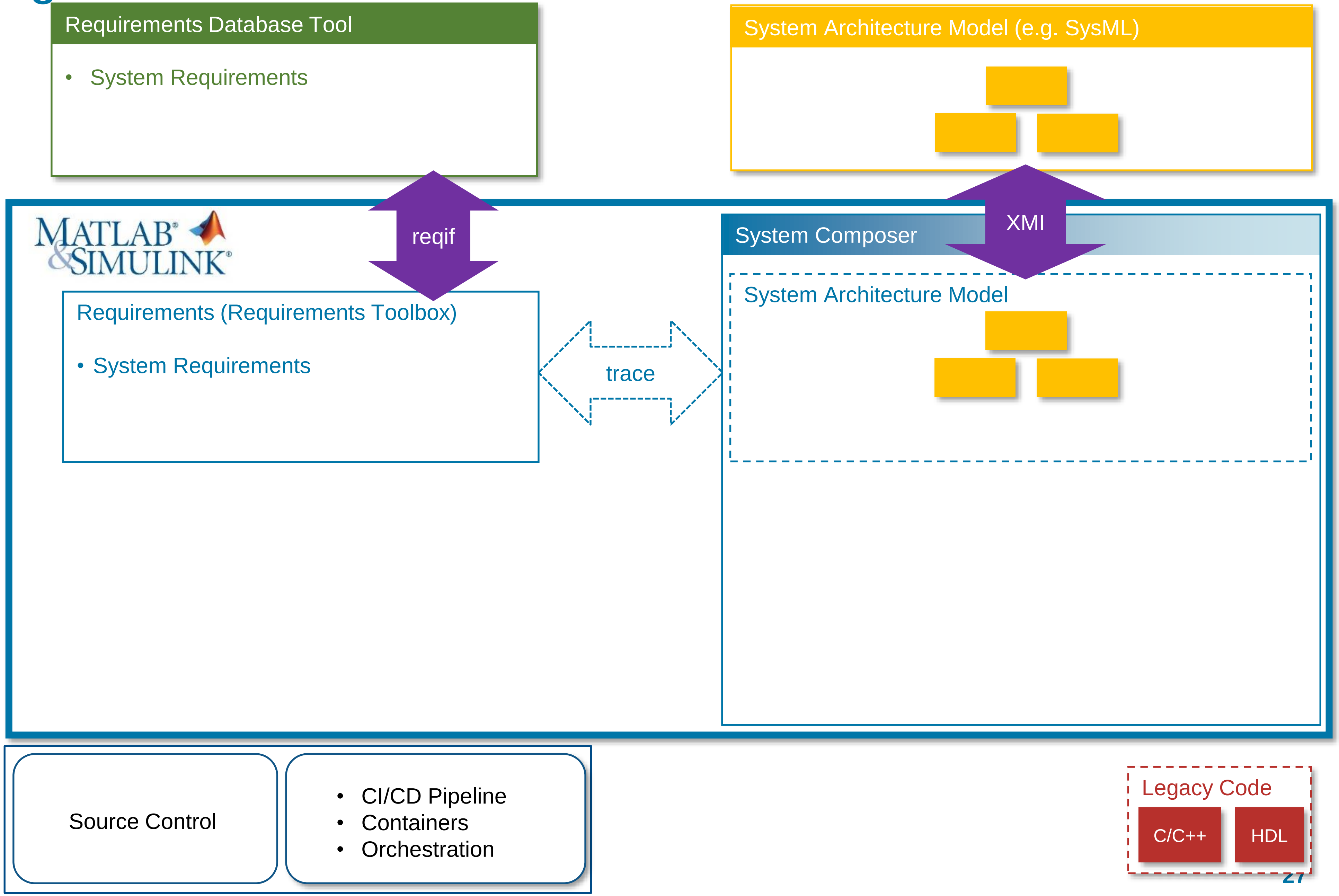
Source Control

- CI/CD Pipeline
- Containers
- Orchestration

Legacy Code

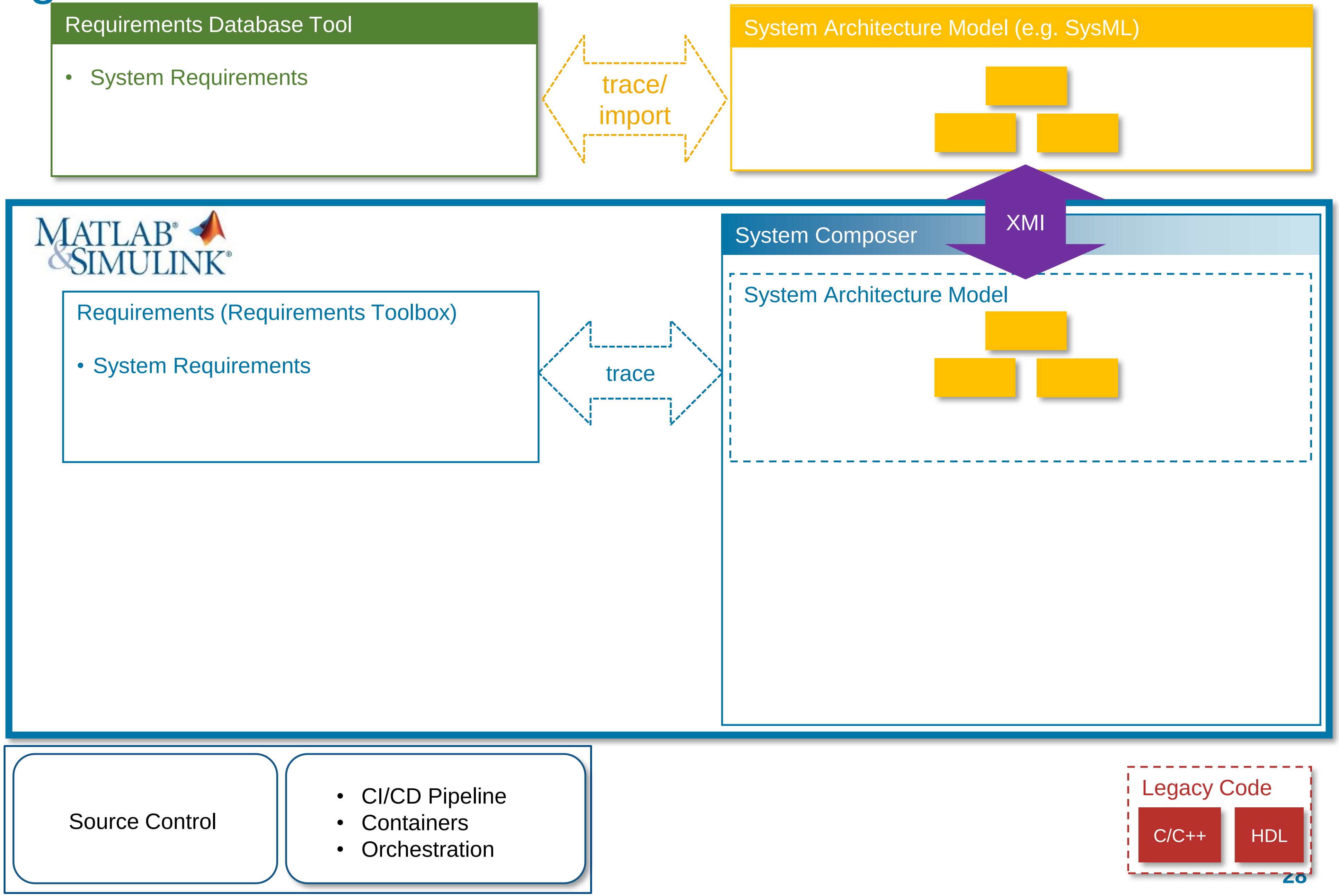
Digital Engineering with MathWorks

- Requirements are then be traced to the architecture model



Digital Engineering with MathWorks

- If the SysML model was linked to requirements, the XMI import can also create the Requirements Toolbox instantiation and traceability, no .reqif needed



Requirements Database Tool

- System Requirements

trace/ import

System Architecture Model (e.g. SysML)

XMI

MATLAB & SIMULINK

Requirements (Requirements Toolbox)

- System Requirements

trace

System Composer

System Architecture Model

Source Control

- CI/CD Pipeline
- Containers
- Orchestration

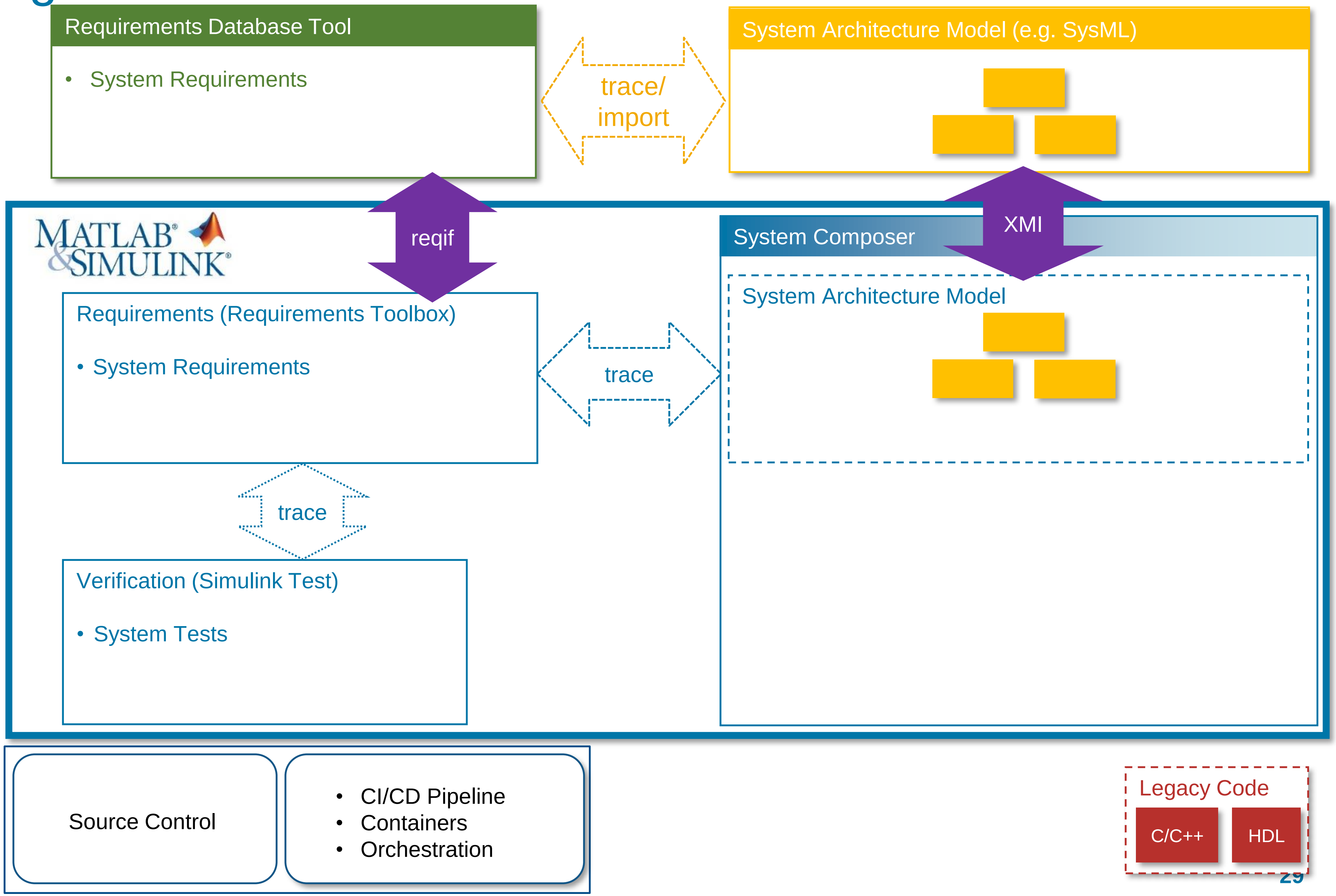
Legacy Code

C/C++

HDL

Digital Engineering with MathWorks

- Test cases can be authored for system verification



Requirements Database Tool

- System Requirements

trace/ import

System Architecture Model (e.g. SysML)

reqif

XMI

MATLAB & SIMULINK

Requirements (Requirements Toolbox)

- System Requirements

trace

System Composer

System Architecture Model

trace

Verification (Simulink Test)

- System Tests

Source Control

- CI/CD Pipeline
- Containers
- Orchestration

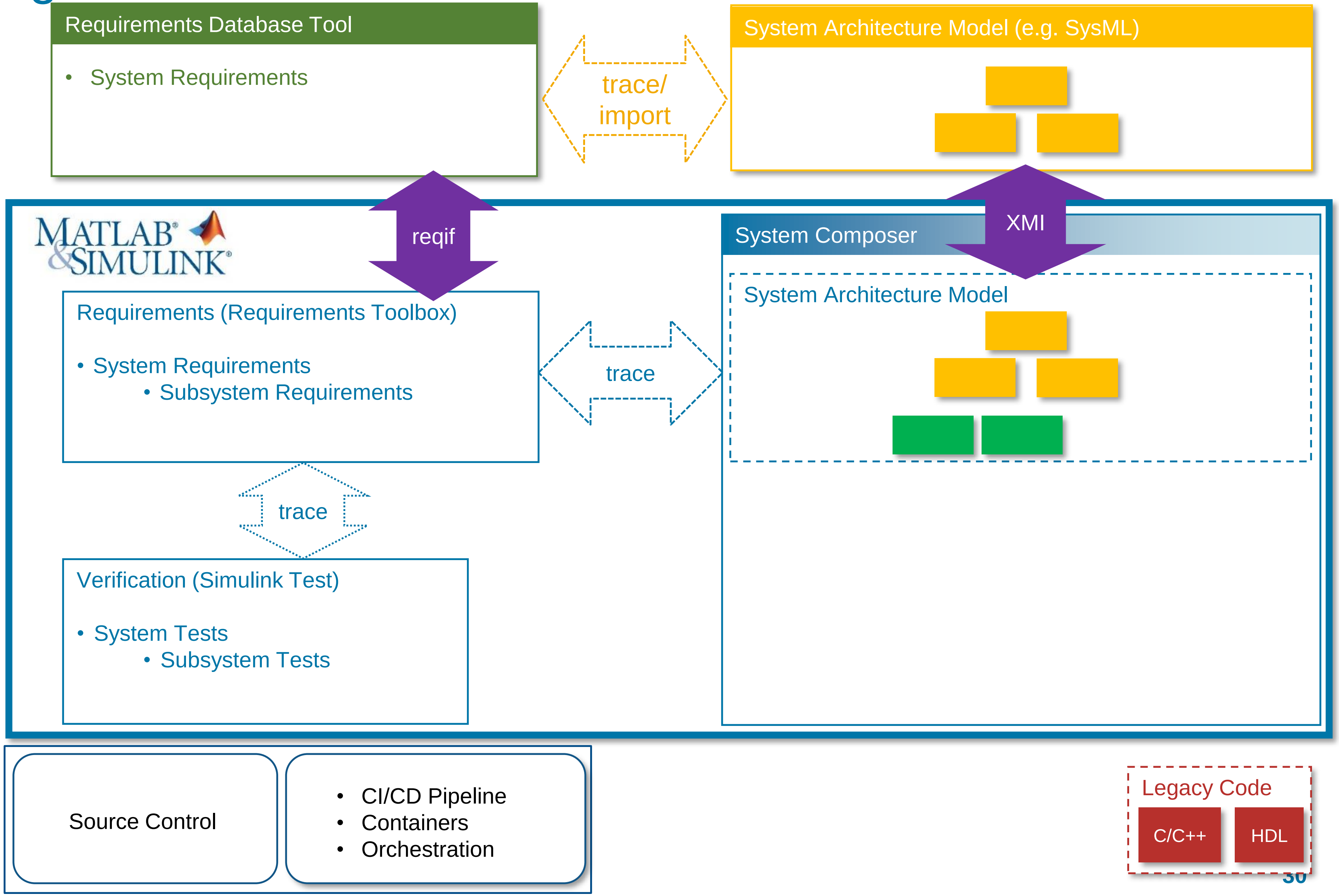
Legacy Code

C/C++

HDL

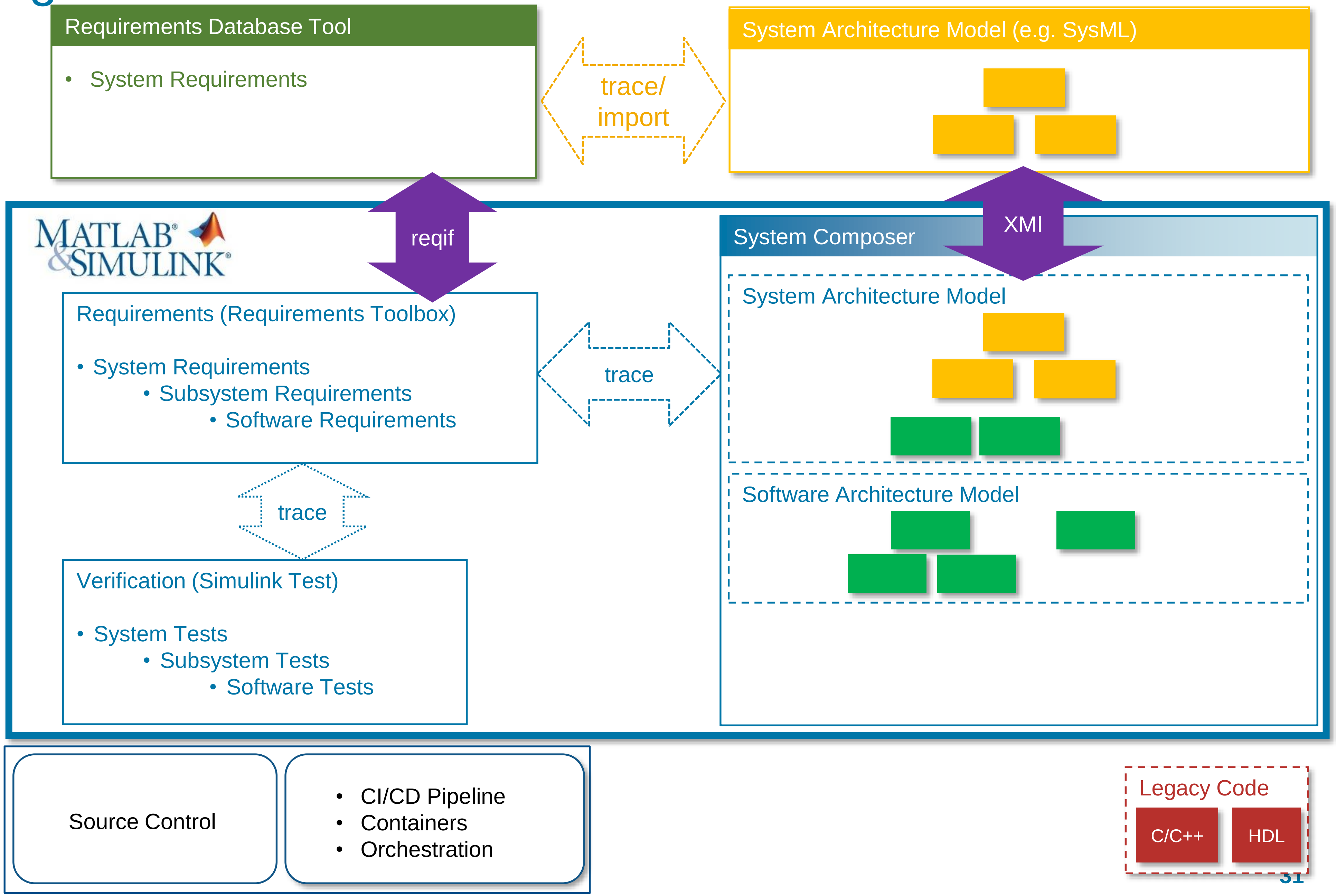
Digital Engineering with MathWorks

- The system architecture model is further elaborated inside System Composer
- The requirements derivation, allocation, and trace is done in parallel in Simulink Requirements
- Test cases for the derived requirements are developed in parallel in Simulink Test



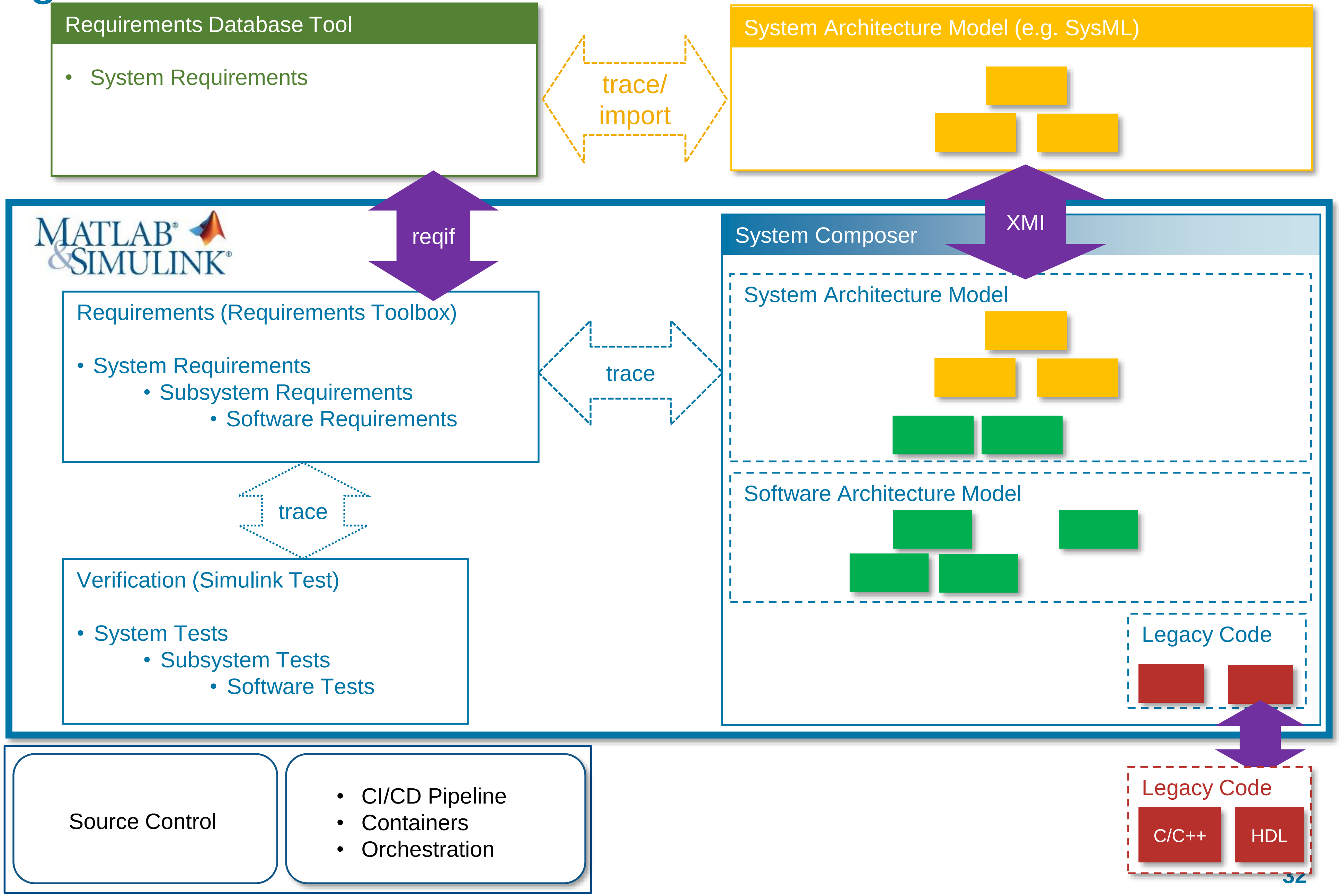
Digital Engineering with MathWorks

- A software architecture model is elaborated in the SAME MODEL AS THE SYSTEM ARCHITECTURE
- Software requirements are derived, allocated, and traced to the software architecture
- Test cases for the derived software requirements are developed in parallel in Simulink Test



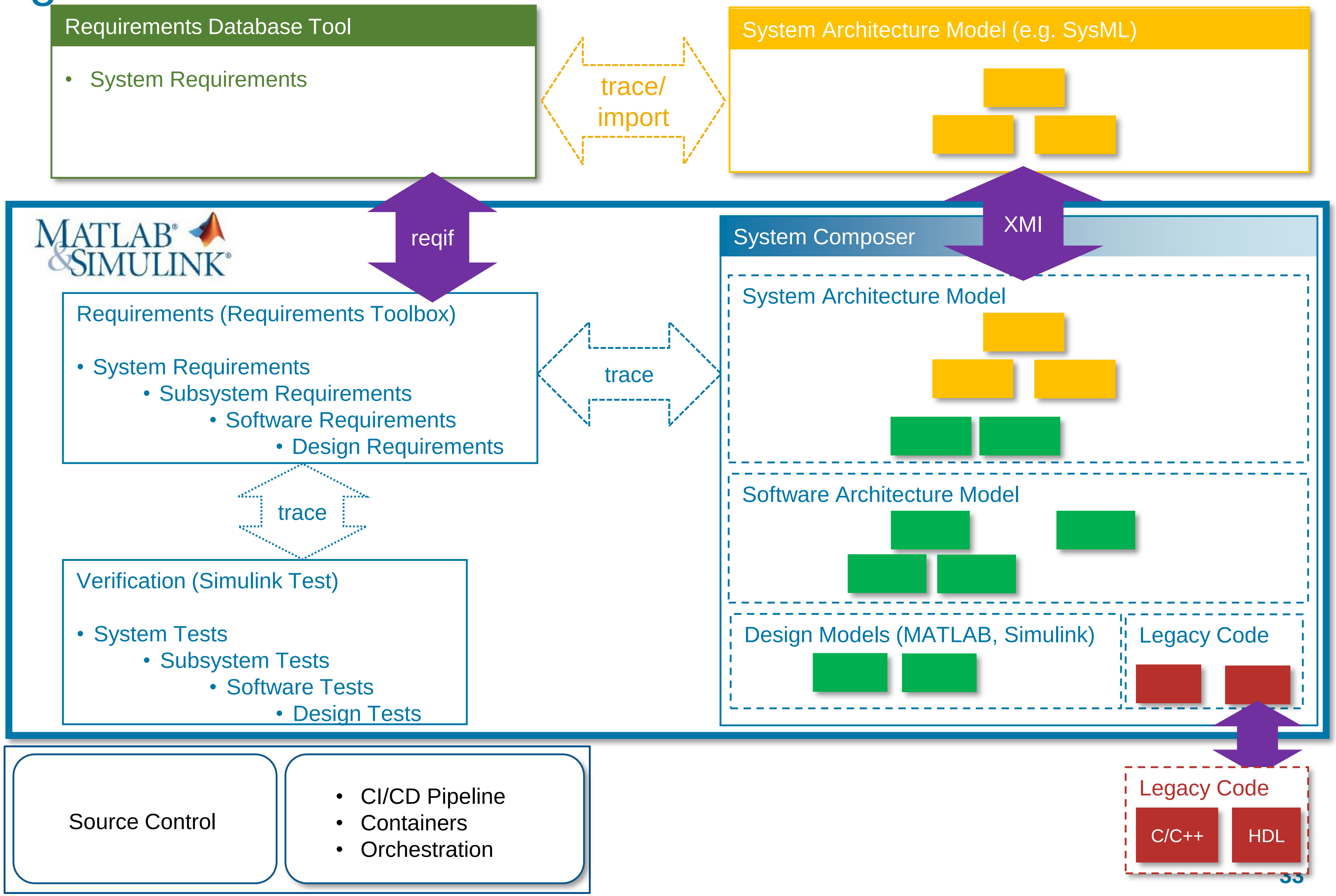
Digital Engineering with MathWorks

- The legacy code can be imported into THE SAME MODEL to populate the components of the software architecture



Digital Engineering with MathWorks

- New designs are developed with Model-Based Design around the existing legacy code
- Design requirements are derived, allocated, and traced to new design models and legacy code model components
- Test cases for the derived design requirements are developed in parallel in Simulink Test



Requirements Database Tool

- System Requirements

trace/ import

System Architecture Model (e.g. SysML)

reqif

XMI

MATLAB & SIMULINK

Requirements (Requirements Toolbox)

- System Requirements
  - Subsystem Requirements
    - Software Requirements
      - Design Requirements

trace

trace

Verification (Simulink Test)

- System Tests
  - Subsystem Tests
    - Software Tests
      - Design Tests

System Composer

System Architecture Model

Software Architecture Model

Design Models (MATLAB, Simulink)

Legacy Code

Source Control

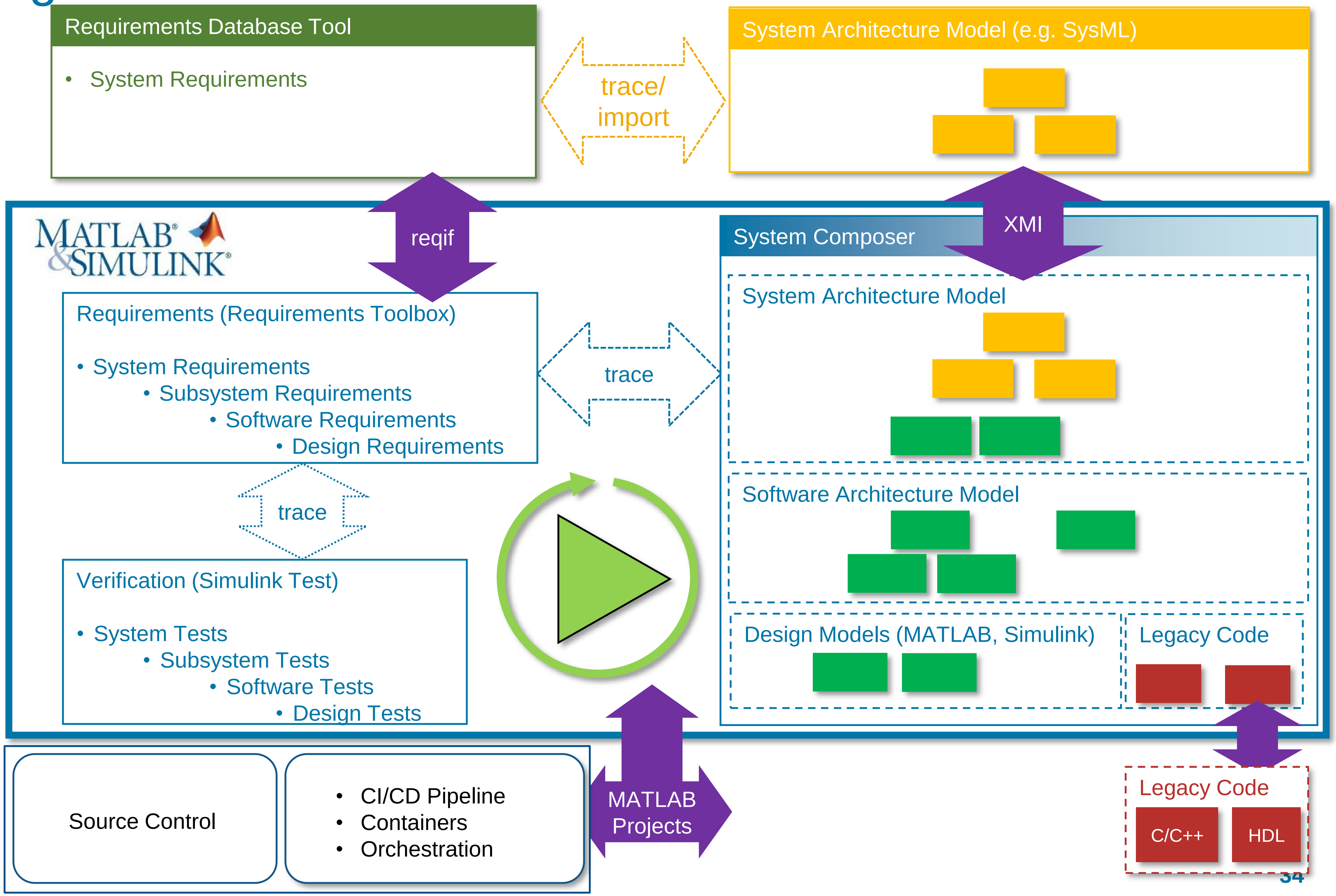
- CI/CD Pipeline
- Containers
- Orchestration

Legacy Code

C/C++ HDL

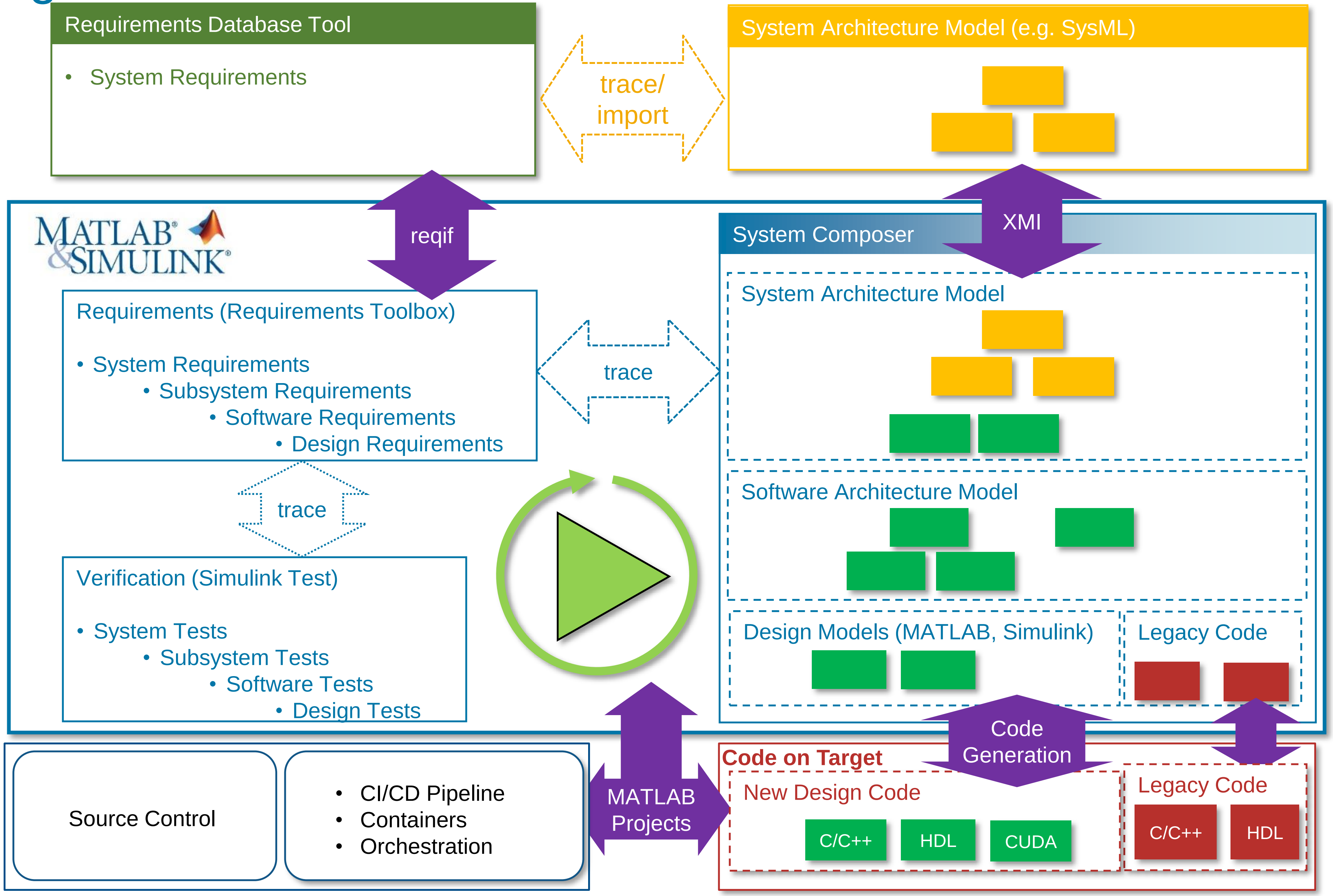
Digital Engineering with MathWorks

- Models can be verified using simulation!
- Verification can be automated both locally, and as part of the larger CM/CI/CD pipeline



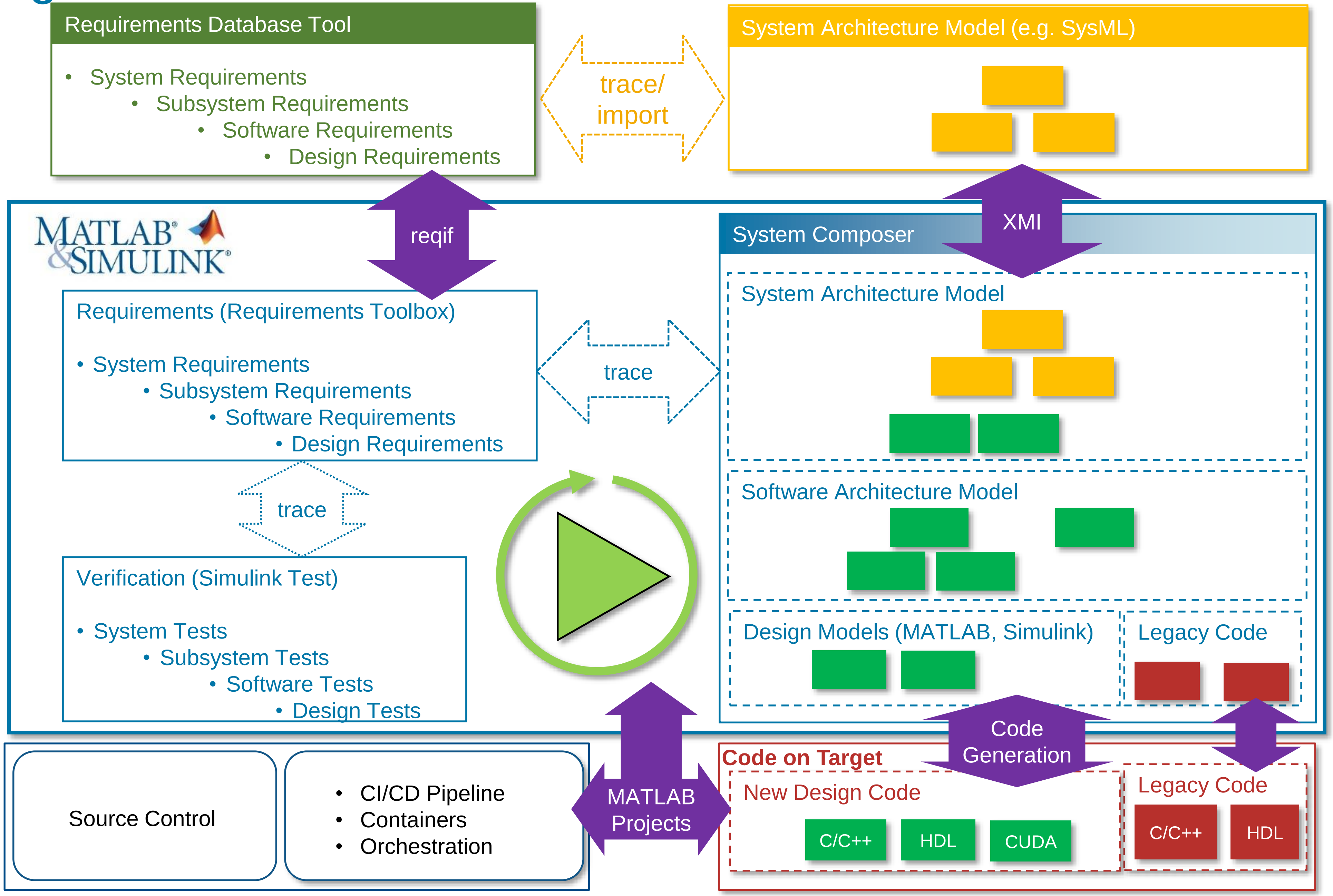
Digital Engineering with MathWorks

- Final implementation for new designs can be automatically generated from design models
- The Digital Thread is automatically extended into the generated code
- The generated code can be verified by the same tests used to verify the models



Digital Engineering with MathWorks

- All work from MathWorks tools can be exported via .reqif and/or XMI if your customer requires artifact delivery in a specific tool/format



We’re here to help!

Digital Engineering and DevSecOps are complicated areas. MathWorks has significant experience working with customers to find the optimal blend for how to integrate our tools into these workflows. Please reach out and we’ll be happy to follow up with you on these or other topics.

Mike Anthony: manthony@mathworks.com

Kirsten McCane: kmccane@mathworks.com

Jon Friedman: jfriedma@mathworks.com

www.mathworks.com