Model-Based Engineering with AADL

SSD/ACPS/MBE Team

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Document Markings

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM21-0152

Outline

The Problem: Safety-Critical Embedded Software Systems

- Part of the Solution: AADL
- Zooming Out: A Holistic View of Research Using AADL

DoD Impacts: Army AADL Success Story

Alignment with Digital Engineering Strategy

ModDevOps and Digital Twins

Model-based Engineering Runtime Verification of AADL Models

The Problem: Safety-Critical Embedded Software Systems

Model-Based Engineering with AADL:
Transitioning Research to Practice

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

The Safety-Critical Embedded Software System Challenge

Problem:

- Software increasingly dominates safety and mission-critical system development
- Issues discovered long after they are created

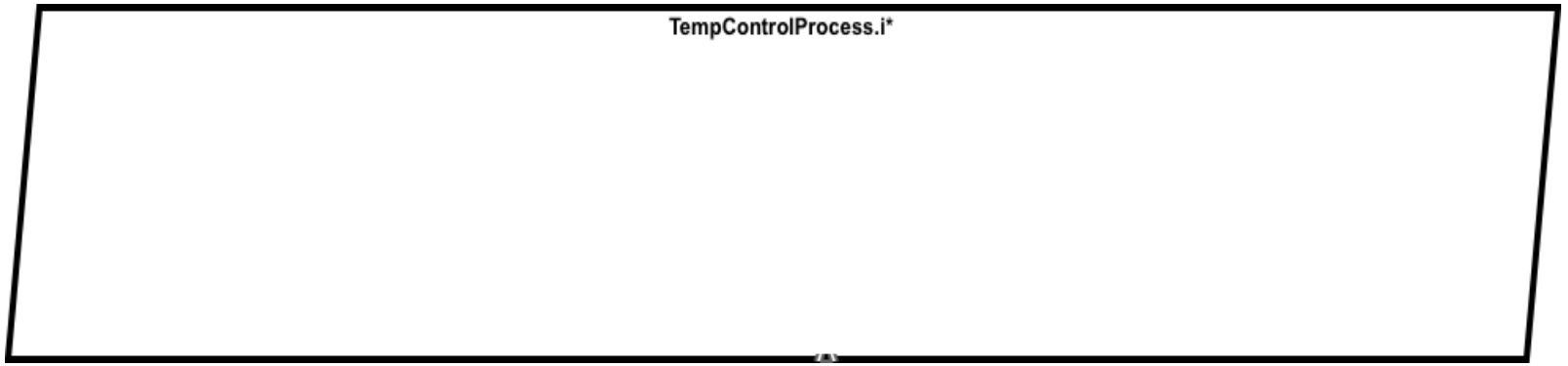
Goal:

Early discovery of system-level issues through virtual integration and incremental analytical assurance

Solution:

- **Language** standardized via SAE International & matured into practice through pilot projects & industry initiatives
- **Tooling** available under open source license continually enhances analysis, verification, and generation capabilities
- Direct alignment with DoD Digital Engineering Strategy

AADL Overview

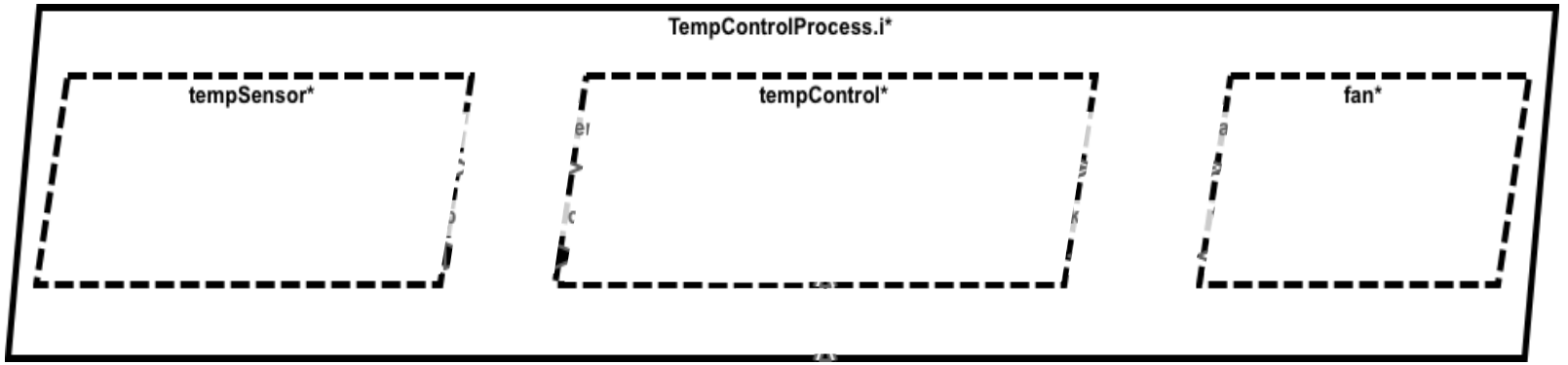


Like a lot of models that engineers draw every day on their whiteboards, AADL consists of boxes and lines

The difference between AADL and a whiteboard is that AADL has precise *semantics*

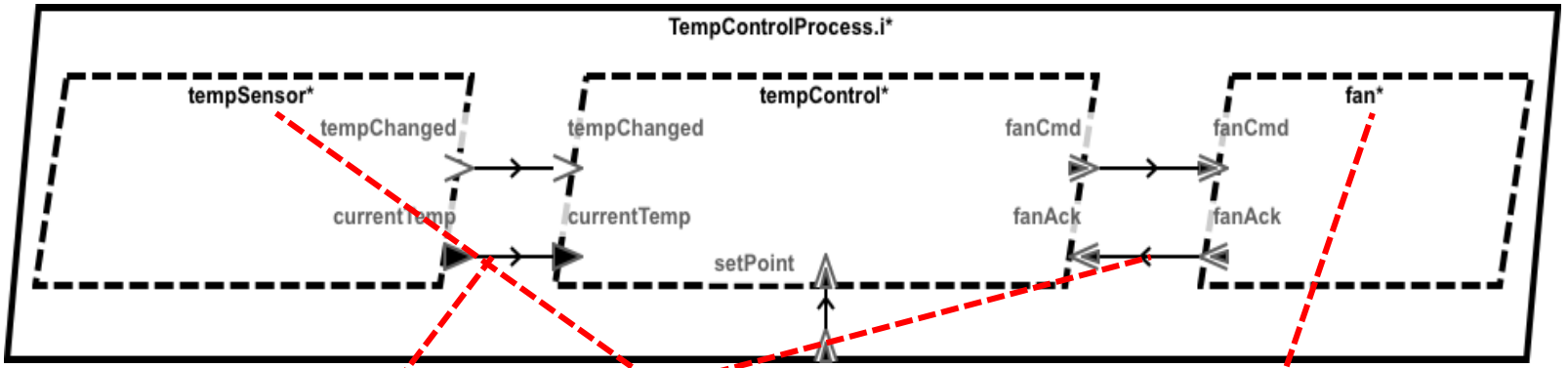
This box represents a computer process – a protected region of memory and a space where we can allocate individual threads

AADL Overview



Those threads are also boxes – but they have very precise meanings.

AADL Overview



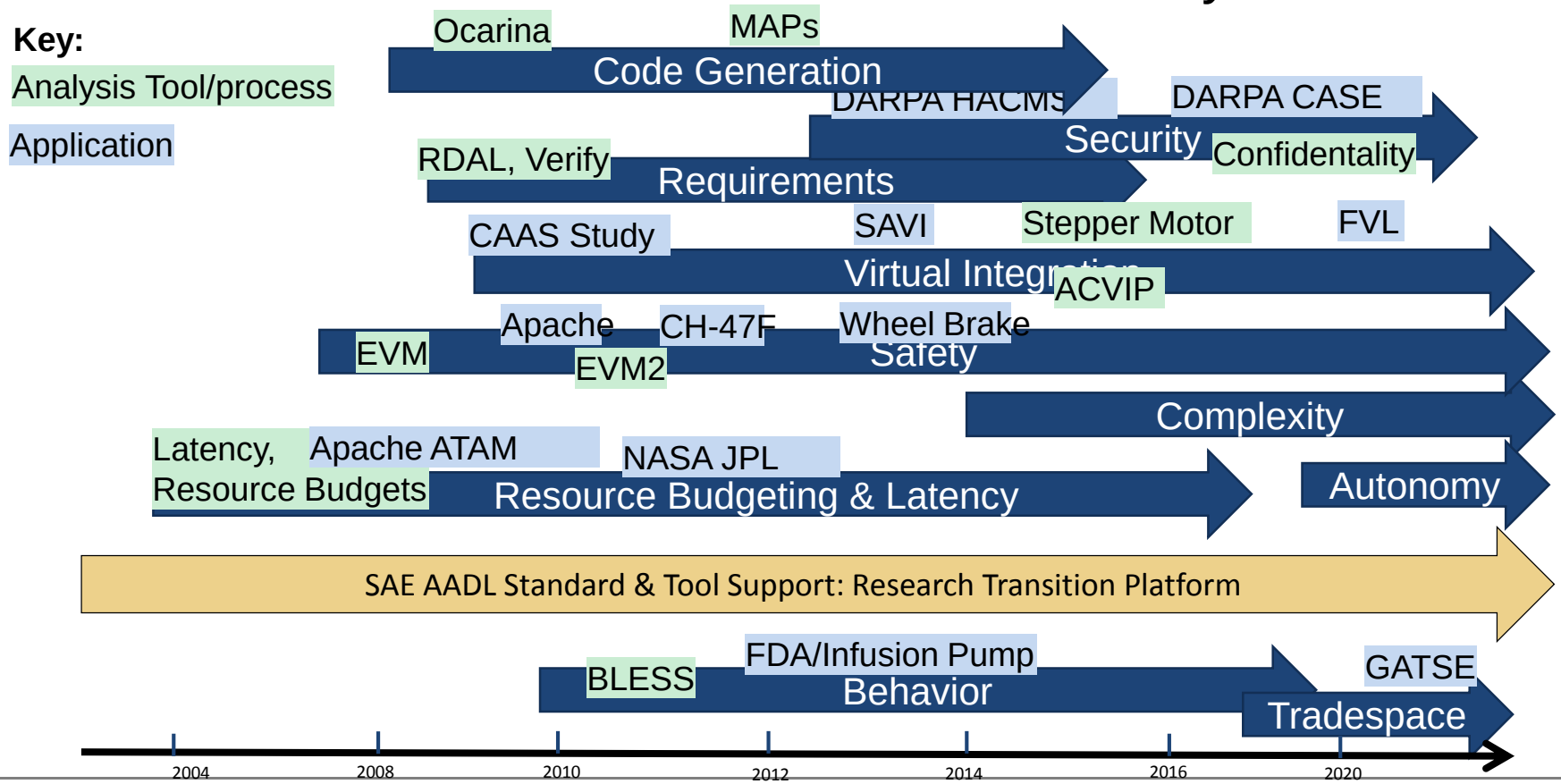
We can connect the threads together using lines to represent different types of intra-process communication

We add more semantics via *properties* – they are useful for both system analyses and to guide code generation

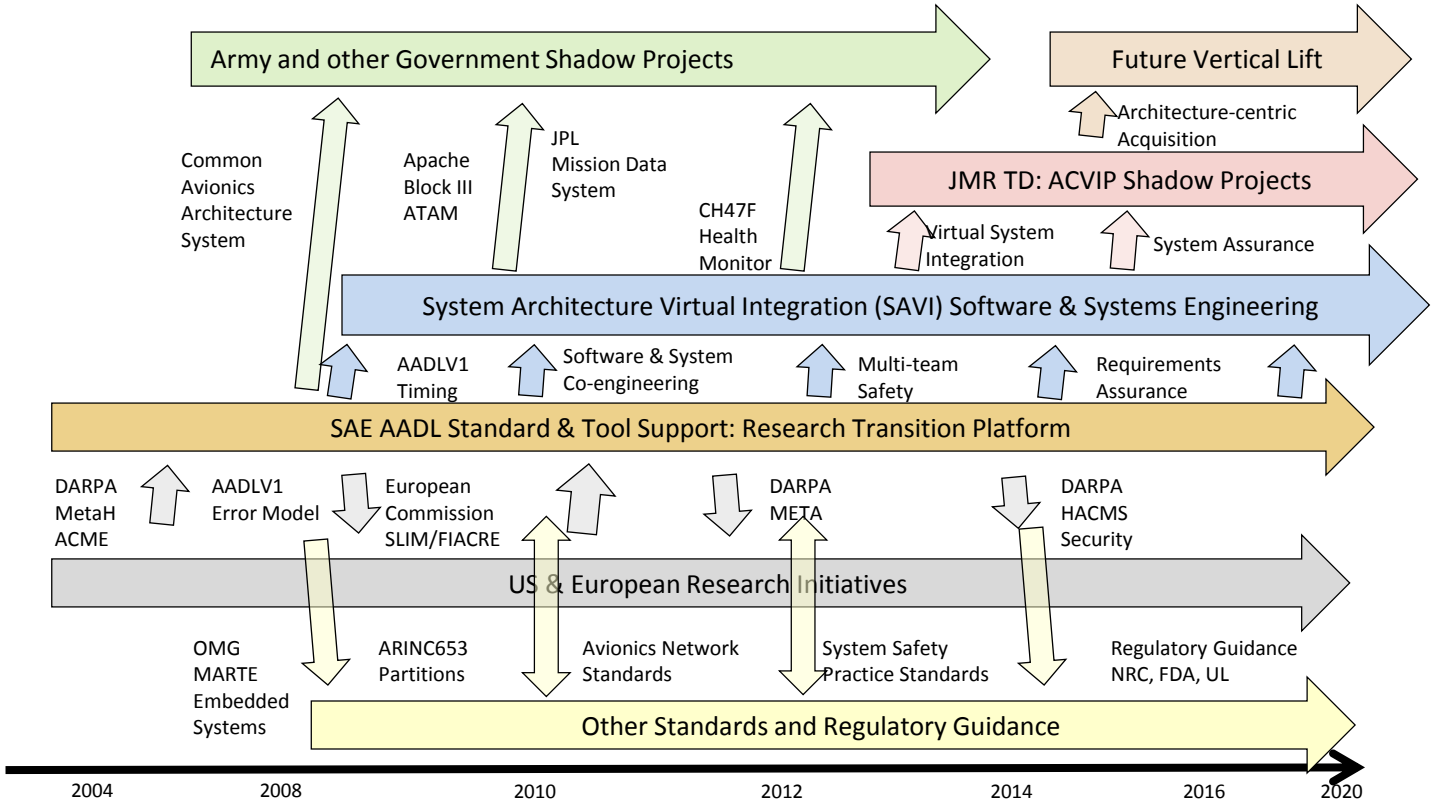
This box shows a periodic thread – it is dispatched regularly according to some clock

And this thread is sporadic – it is dispatched whenever a message arrives at a specified port

A Holistic View of Lines of Research Enabled by AADL

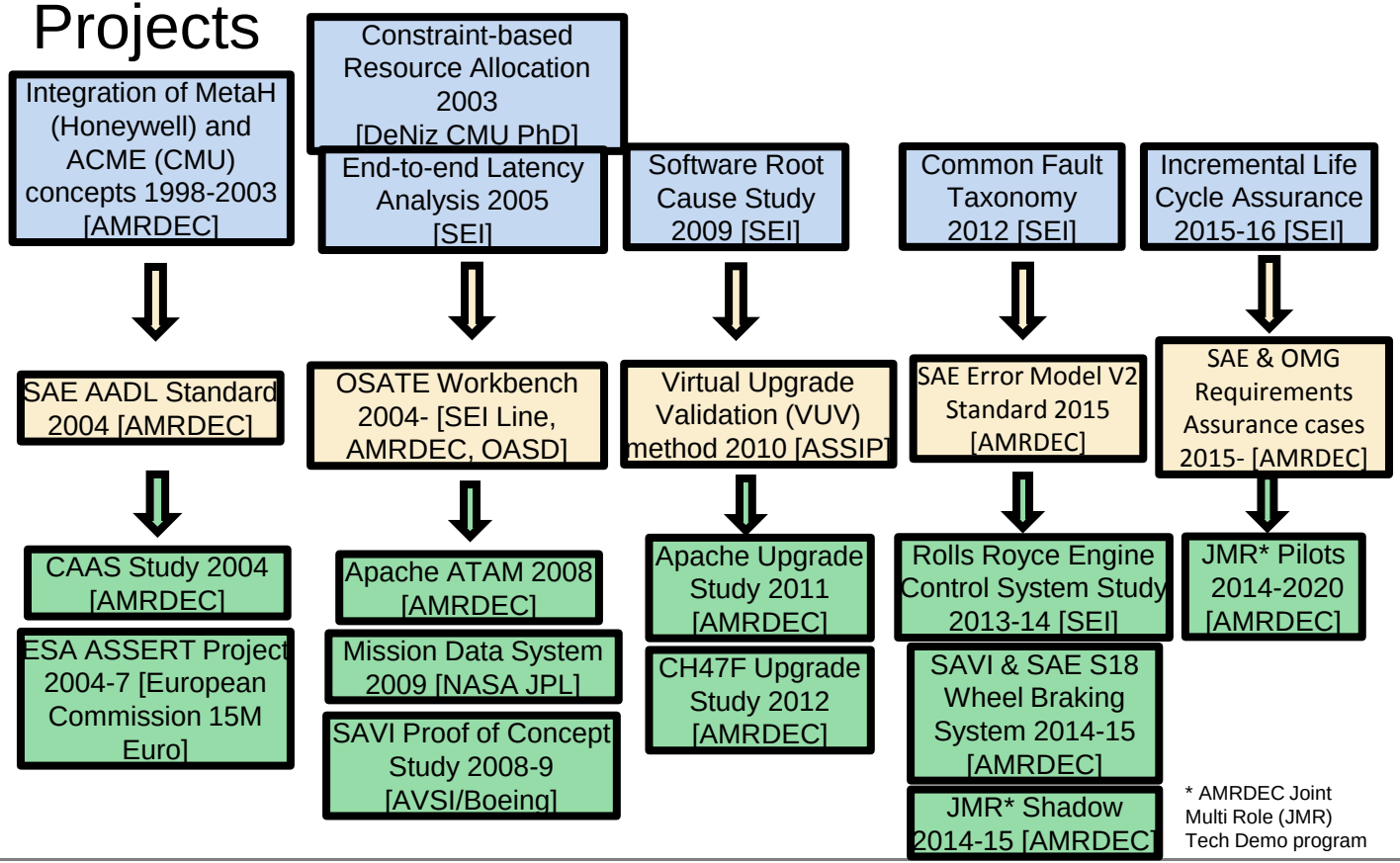


AADL-based Virtual System Integration Technology Approach



Evolution, Maturation and Transition

SEI Research to AADL Standard to Pilot Projects



SAE International AADL Standard Suite (AS-5506 series)

Core AADL language standard [V1 2004, V2 2012, V2.2 2017]

- Focused on embedded software system modeling, analysis, and generation
- Strongly typed language with well-defined semantics for execution of threads, processes on partitions and processor, sampled/queued communication, modes, end to end flows
- Textual and graphical notation
- Revision V3 in progress: interface composition, system configuration, binding, type system unification

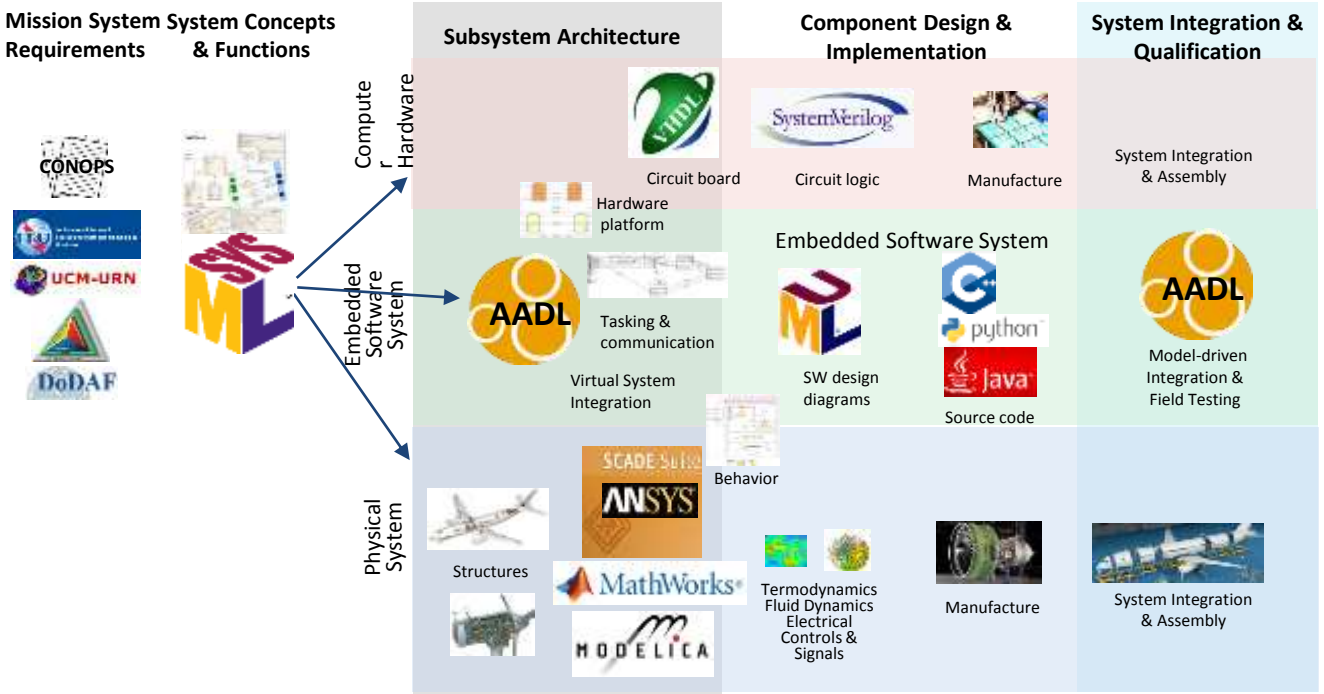
Standardized AADL Annex Extensions

- Error Model language for safety, reliability, security analysis [2006, 2015]
- ARINC653 extension for partitioned architectures [2011, 2015]
- Behavior Specification Language for modes and interaction behavior [2011, 2017]
- Data Modeling extension for interfacing with data models (UML, ASN.1, …) [2011]
- AADL Runtime System & Code Generation [2006, 2015]
- FACE Annex [2019]

AADL Annexes in Progress

- Network Specification Annex
- Cyber Security Annex
- Requirements Definition and Assurance Annex
- Synchronous System Specification Annex

Multiple Languages and Tools to Meet Users Needs



Filling the Modeling and Analysis Gap for Embedded Software System

DoD Impacts: Army AADL Success Story

Model-Based Engineering with AADL:
Transitioning Research to Practice

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Helping to Revolutionize Army Aviation

Over many years, the SEI has had an outstanding partnership with the U.S. Army, which is at the vanguard of applying AADL and ACVIP to the Army's future vertical lift challenge.

Image source: army.mil

Benefits of AADL & ACVIP (via Alex Boydston)

- Decreased fielding time by finding problems early
- Early risk reduction by discovering performance issues early
- Increased cybersecurity by using AADL/ACVIP to improve system security
- Decreased development costs and support for MOSA and certification by transforming procurement supporting MBE and ACVIP

Image source: army.mil

Virtual integration of software, hardware, and system supports verification, airworthiness, safety, and cybersecurity certification

Impact

Finding Problems Early (AMRDEC/SEI)

Summary: 6-week virtual integration of health monitoring system on CH47 using AADL

Result: Identified 20 major integration issues early

Benefit: Avoided 12-month delay on 24-month program

CH47 Chinook

High Assurance Cyber Military Systems (HACMS)

Unmanned Quadcopter

TARDEC Autonomous Truck

Improving System Security (DARPA/AFRL)

Summary: AADL applied to unmanned aerial vehicles & autonomous truck

Result: AADL models enforced security policies and were used to auto-build the system

Benefit: Combined with formal methods verification, prevented security intrusion by a red team

Transforming Procurement (Joint Multi-Role)

Summary: Industry/DoD process demonstration

Result: Pre-integration fault identification

Benefit: 10X reduction integration test cost

Makes complex capabilities possible through Agile analytic and virtual integration of real-time safety- and security-critical cyber-physical embedded systems

How AADL and SEI Research Enable Transition

Research

- SEI
- K-State
- Telecom Paris
- UMinn
- GTRI
- Adventium

“Valley of Death”

DoD & Industry

- Army
- ANSYS[1]
- Dassault
- Ellidiss
- Physical Optics Corp
- Innovative Defense Technologies

1 https://www.ansys.com/blog/create-models-architecture-analysis-design-language-aadl

Alignment with Digital Engineering Strategy

Model-Based Engineering with AADL:
Transitioning Research to Practice

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Alignment with Digital Engineering Strategy

June 2018 – Office of the Deputy Assistant Secretary of Defense for Systems Engineering

Michael Griffin (former USD(R&E)): “Those implementing the practices must develop the “how” – the implementation steps necessary to apply digital engineering in each enterprise.”

https://fas.org/man/eprint/digeng-2018.pdf

Boydston et al.: “ACVIP plays a key role in addressing issues in cyber-physical systems (CPS) and can be a key contributor to the U.S. Department of Defense (DoD) Digital Engineering Strategy.”

Architecture-Centric Virtual Integration Process (ACVIP): A Key Component of the DoD Digital Engineering Strategy

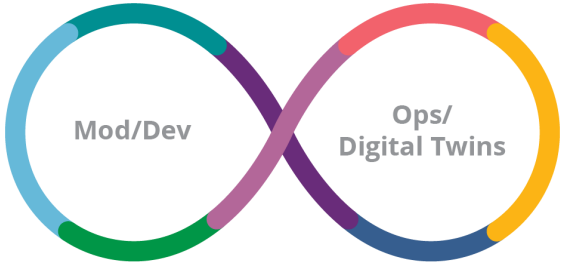
Image source: *DoD Digital Engineering Strategy*, June 2018

ModDevOps and Digital Twins

Model-Based Engineering with AADL:
Transitioning Research to Practice

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

From DevOps to ModDevOps



DevOps delivers software faster with increased quality:

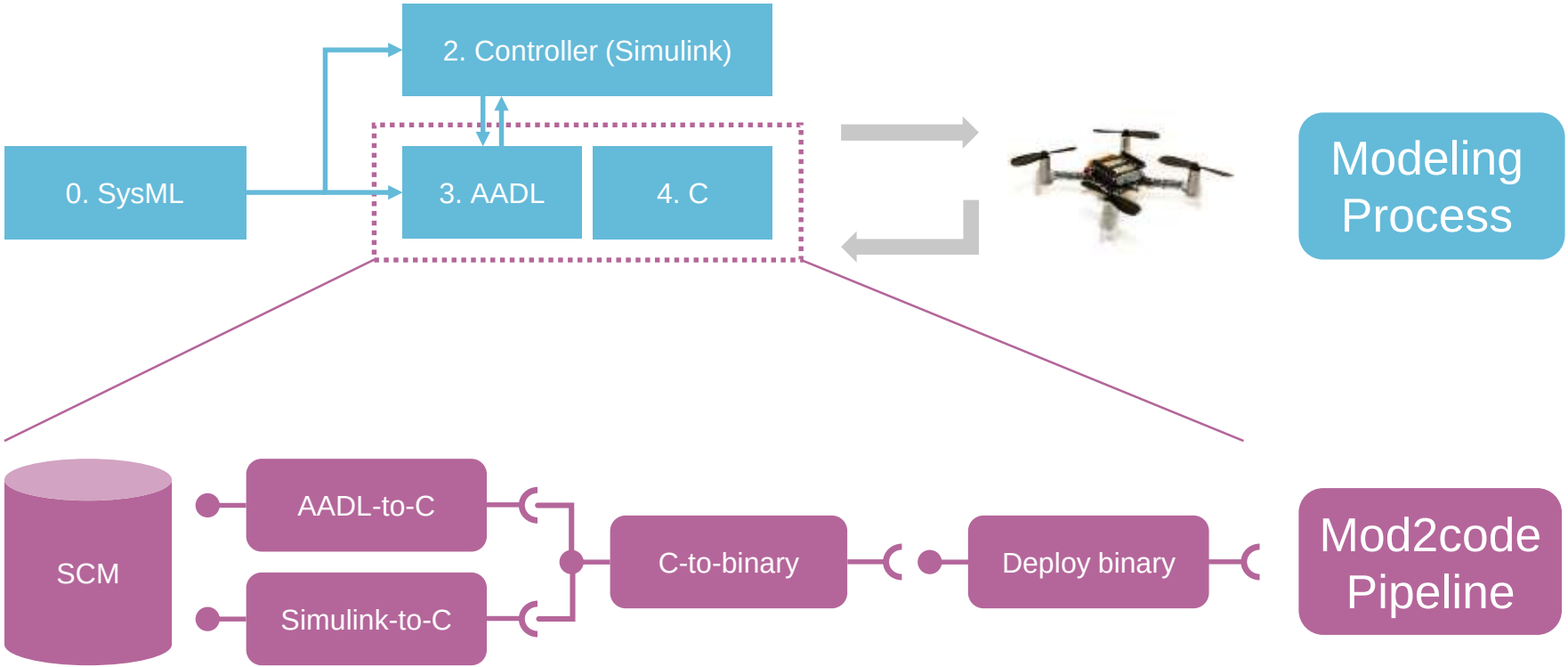
- Continuous integration/deployment
- Containerized systems

DevOps is a software process, to be adapted to systems.

"ModDevOps is a systems/software co-engineering culture and practice that aims at unifying systems engineering (Mod), software development (Dev) and software operation (Ops). The main characteristic of the ModDevOps is to strongly advocate abstraction, automation, and monitoring at all steps of system construction."

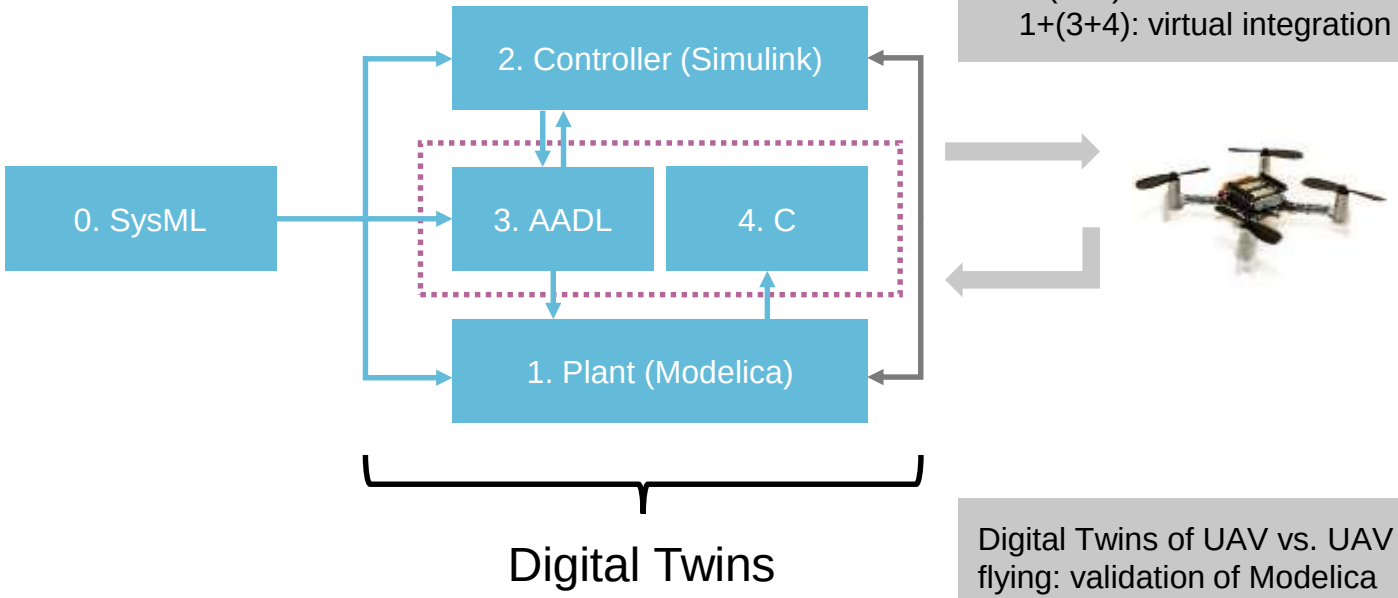
(adapted from https://software.af.mil/training/devops/)

ModDevOps in Action – Modeling Process



From ModDevOps to TwinOps

1-2-3-4: “mega-modeling” V&V
- 1-2: HLR validation
- 2-(3+4): validation of LLR
  1+(3+4): virtual integration



Digital Twins of UAV vs. UAV flying: validation of Modelica model, efficiency of the controller (overshoot verification) and timing verification of software.

From ModDevOps to TwinOps

SCM

Mod2Code

Cloud Deployment

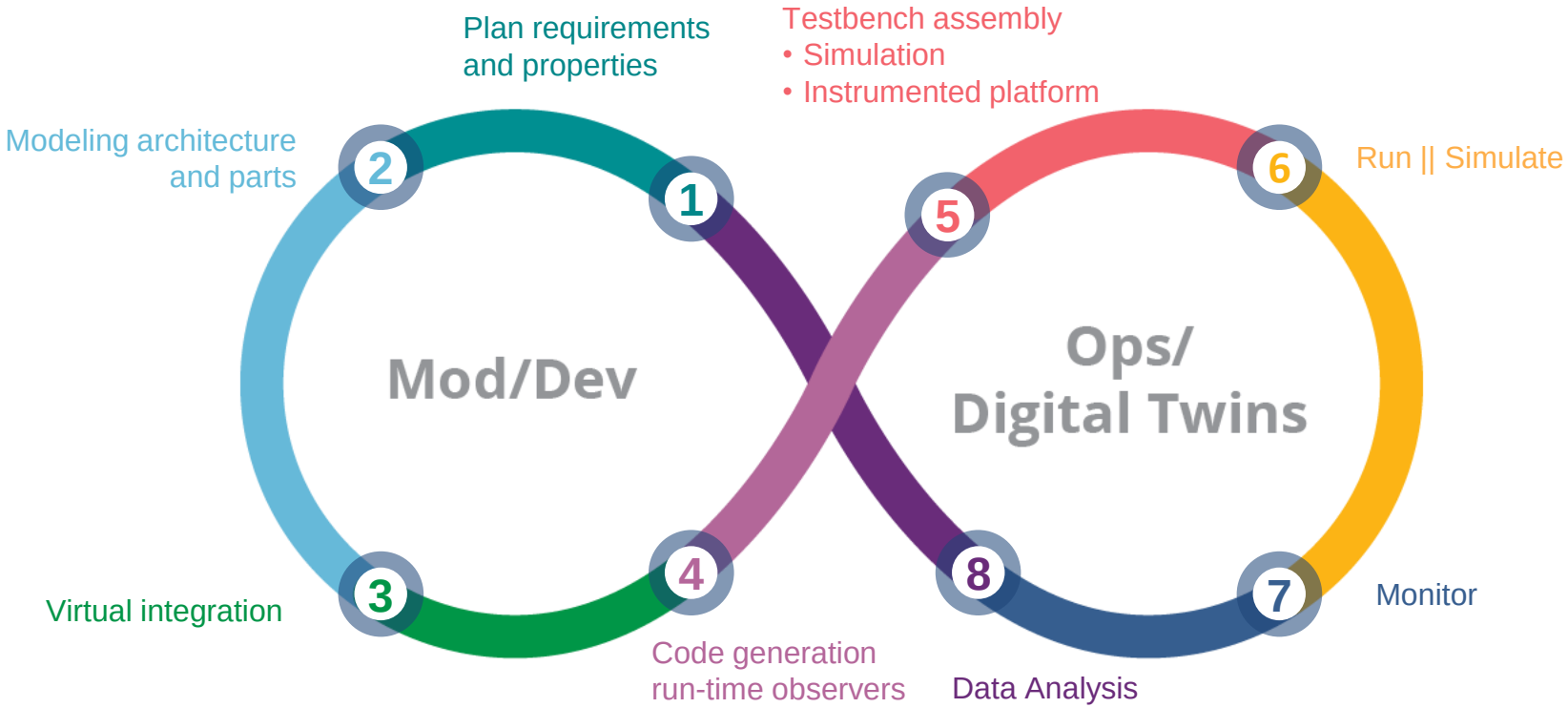
SCM

Mod2Simu

Digital Twins

Modeling Process

TwinOps: Continuous System Improvement through ModDevOps and Digital Twins



Model-based Engineering

Runtime Verification of AADL Models

Model-Based Engineering with AADL:
Transitioning Research to Practice

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Applied Research and Transition

Challenge: Development of approach to help ensure that properties specified and analyzed in an architecture are preserved in the runtime system.

Participating with an organization via SBIR award to develop and transition an approach, Currently in Phase 2

Project Goal: ***Develop a software tool that will check instrumentation data collected from an integrated mission system to see if the observed system behaviors of an integrated mission system conform to required and allowed behaviors defined in an Architectural Analysis and Design Language (AADL) model of the integrated aviation software and hardware mission system.”***

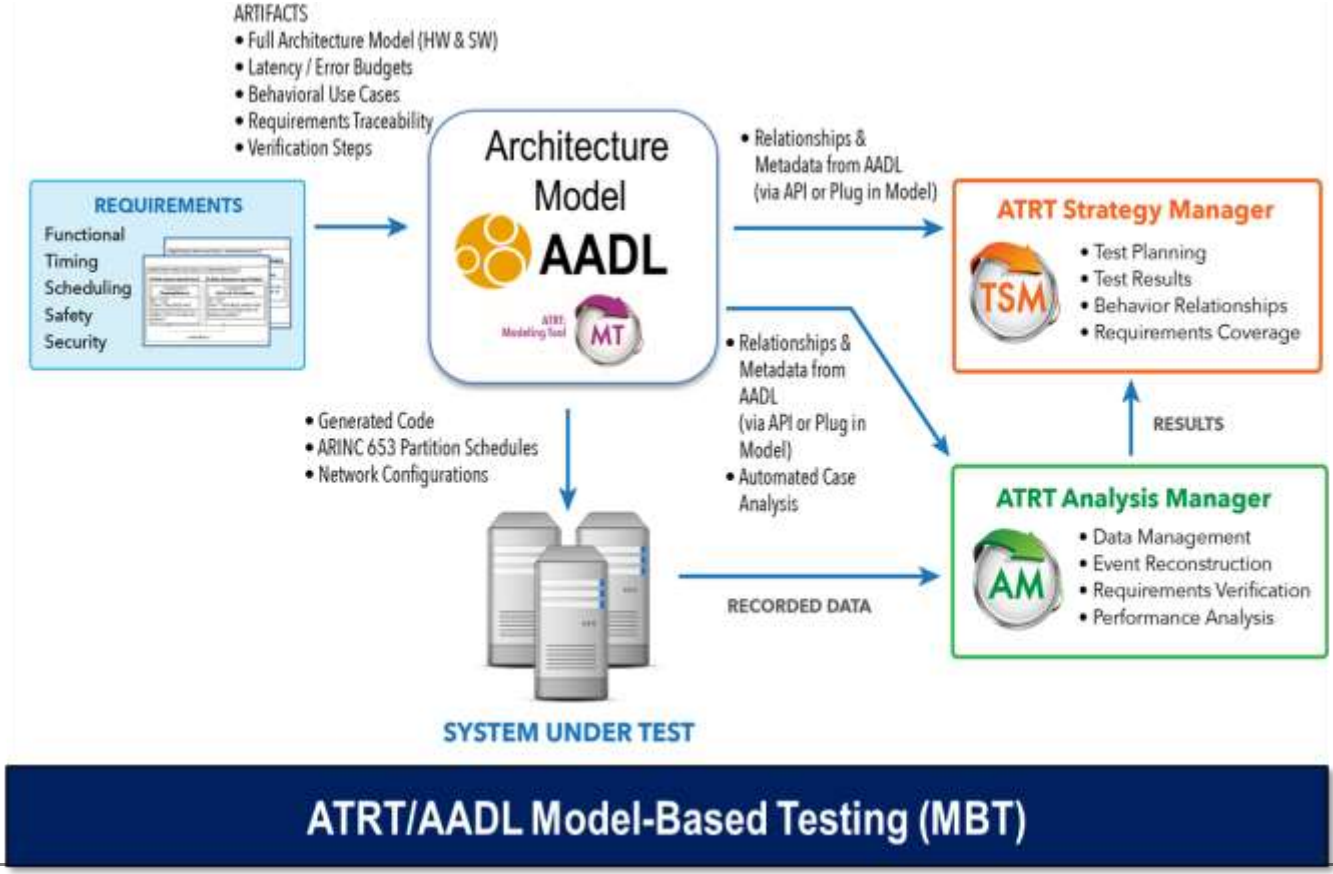
.

Technical Approach

Partner with an organization that produces a industrial tool (ATRT) that performs a similar verification task (SysML - centric):

- Identify the concepts and properties in AADL to verify from an executing system
- Map semantics of AADL models into a form that conforms to ATRT, adapt ATRT where necessary – produce analysis plugin in OSATE
- Ensure that ATRT state reconstruction from runtime data is correct
- Post evaluation of data from test scenarios: correlate runtime state and data with those specified in AADL. Continuous verification of property values (in and out of bounds) during the entire test period.
- Produce verification reports, test conducts, trace to system level requirements
- Set up a test environment analogous to avionics mission platform and system
- Produce a tool that is TRL 6
- Identify and engage transition partners

Technical Approach-2

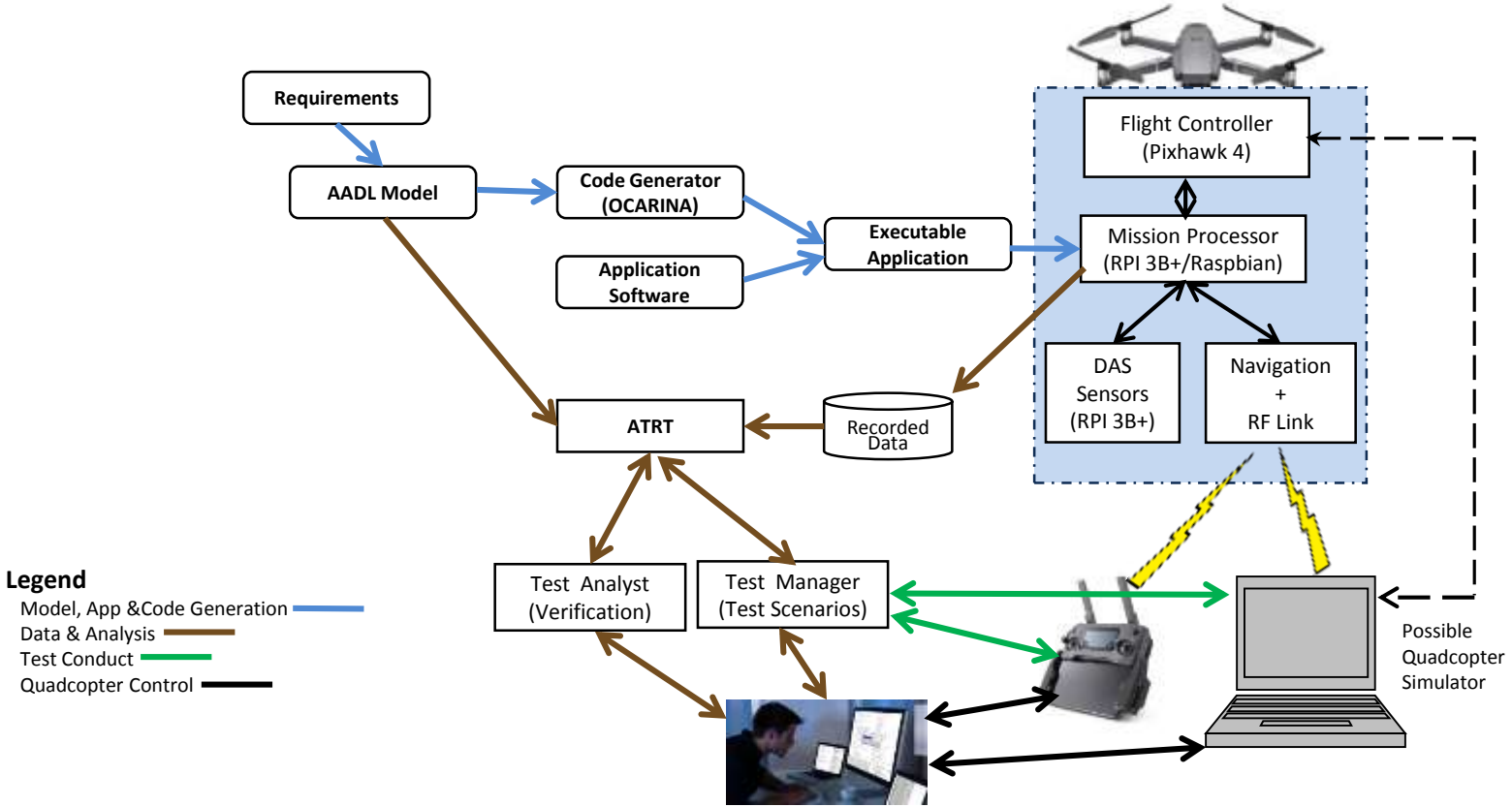


Architectural Properties

AADL model properties that can be verified include:

- End to end flow of data, events, event-data
- Latency (between/through logical components, execution of threads Ensure that ATRT state reconstruction from runtime data is correct
- Modal operation of threads
- Communication bus bandwidth (worst case loads, scheduled loads)
- Power bus capacity (power)
- Resource utilization of bound loads (memory, CPU)
- Error flow (ensure error types are handled/mitigated)
- Functional hazard analysis, fault tree analysis
- Security properties – Information disclosure as in STRIDE security framework: Data confidentiality, trust boundaries

Model-based Testing – Platform/Testbed Flows



For More Information

Contact Us

Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479

Web: www.sei.cmu.edu

Email: info@sei.cmu.edu