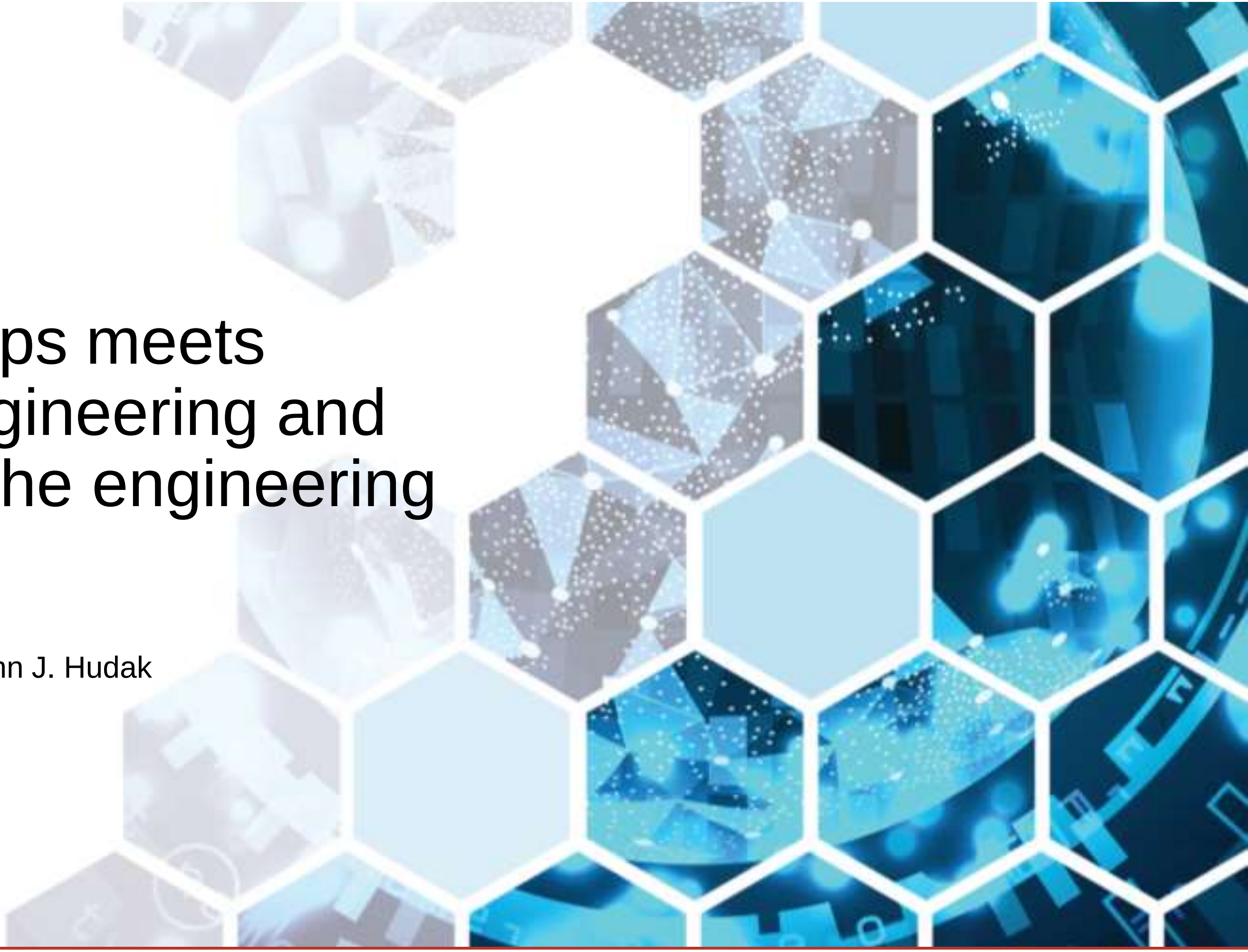


TwinOps – DevOps meets Model-Based Engineering and Digital Twins for the engineering of CPS

Jerome Hugues, Anton Hristosov, John J. Hudak and Joe Yankel

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Carnegie Mellon University
Software Engineering Institute

Model-Based Engineering for Cyber-Physical Systems@SEI

Create the best design that holds up over time as the system evolves. + Test the design without having to write any code. = Build a single model to assess hardware and embedded software before the system is built.

SAE AADL / ACVIP

- Standardized language and process for the engineering safety-critical systems.

OSATE

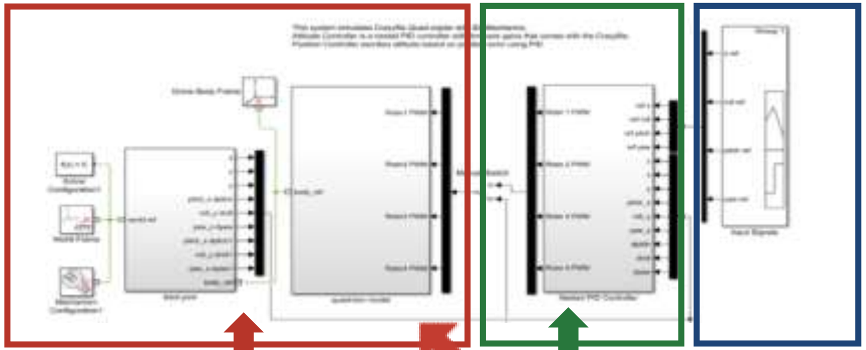
- Open Source AADL toolset for performing verification and validation (V&V).

Pilot Projects

- Maturity increased through case studies and feedback from practitionners

TwinOps problem space: CPS Integration and Testing

High-Level Architecture

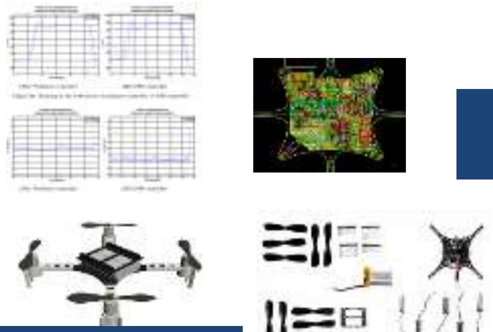


TwinOps: leverage other source of truth (e.g., CAD, Physics) to improve SW V&V

⇒ Use precise models instead of (naïve) abstractions for improved SW V&V

⇒ Combine domains, including SysEng

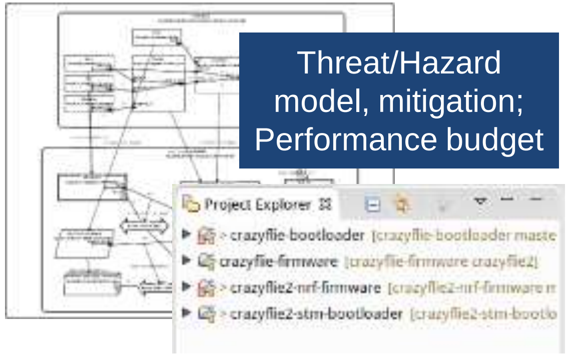
Implementation Space



Actual sources

?

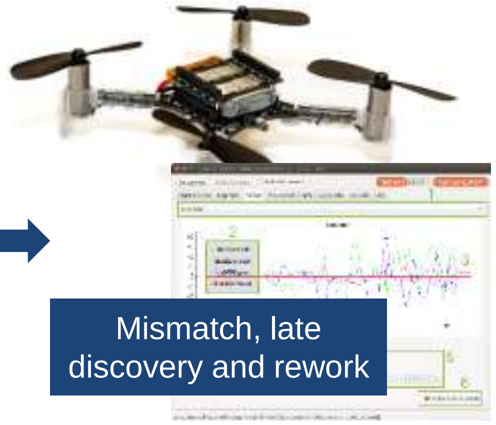
SW/HW Architecture



Threat/Hazard model, mitigation; Performance budget

Project Explorer

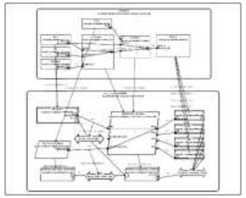
- crazyflie-bootloader [crazyflie-bootloader maste
- crazyflie-firmware [crazyflie-firmware crazyflie2]
- crazyflie2-nrf-firmware [crazyflie2-nrf-firmware n
- crazyflie2-stm-bootloader [crazyflie2-stm-bootlo



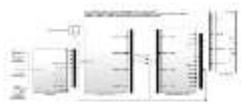
Mismatch, late discovery and rework

Technology Focus: Models and Code Generation

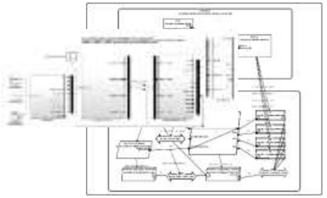
One can ***generate code*** from models ready to be embedded in the system (e.g., AADL to C) and get insights from the system to refine the model metrics.



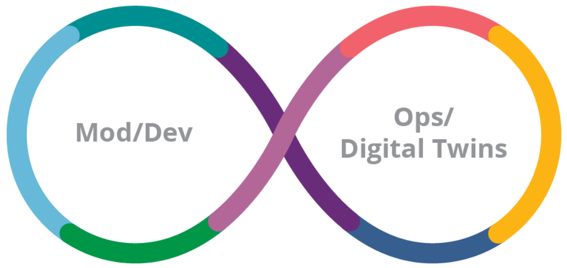
One can ***simulate models*** and generate simulation code as a mock-up of some system parts.



One can build ***Digital Twins***, that compare actual system and its digital simulated doppelganger.



From DevOps to ModDevOps



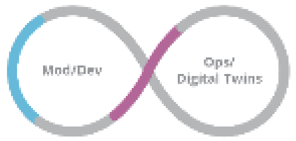
DevOps delivers software faster with increased quality:

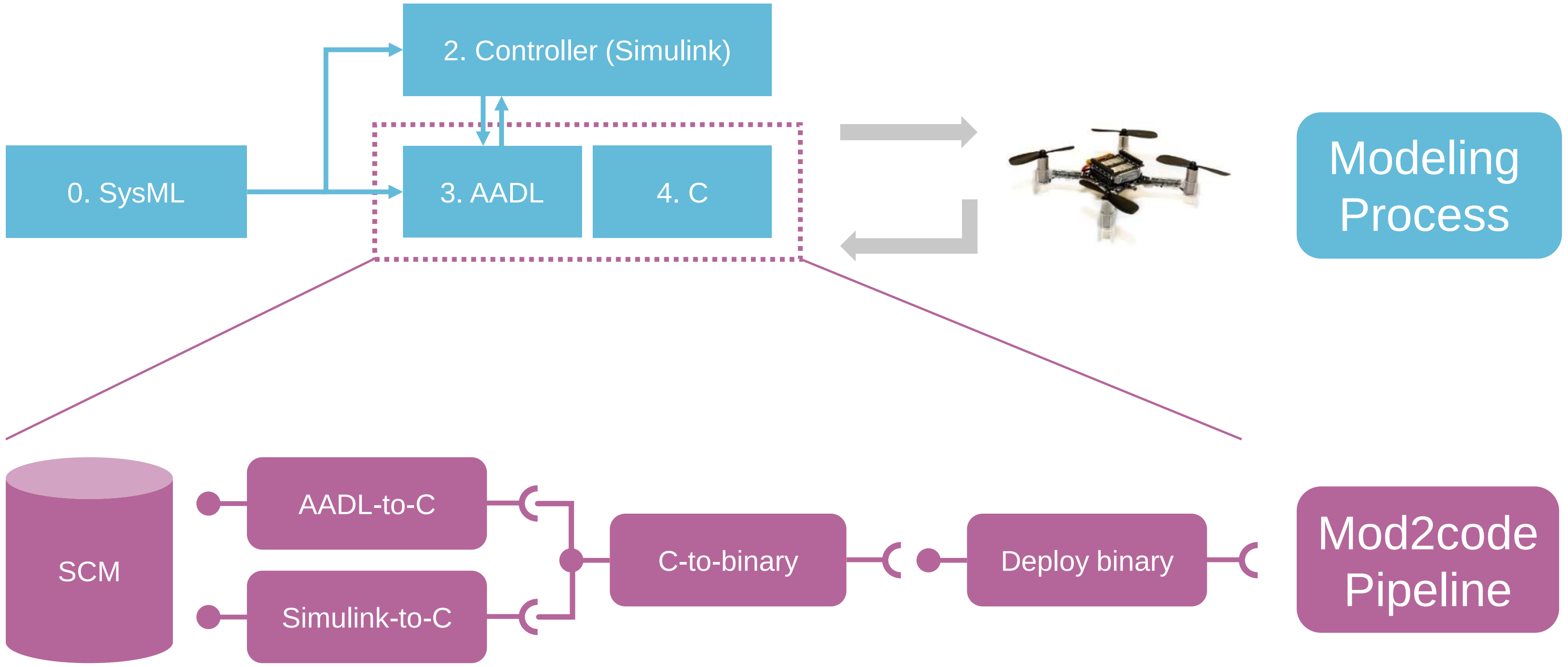
- Continuous integration/deployment
- Containerized systems

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

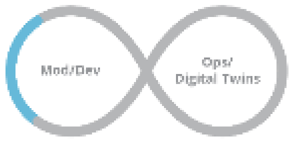
***ModDevOps** is a ~~software~~systems/software co-engineering culture and practice that aims at unifying systems engineering (Mod), software development (Dev) and software operation (Ops). The main characteristic of ModDevOps is to strongly advocate abstraction, automation and monitoring at all steps of system construction, from integration, testing, releasing to deployment and infrastructure management.* (adapted from https://software.af.mil/training/devops/)

ModDevOps in Action – Modeling Process



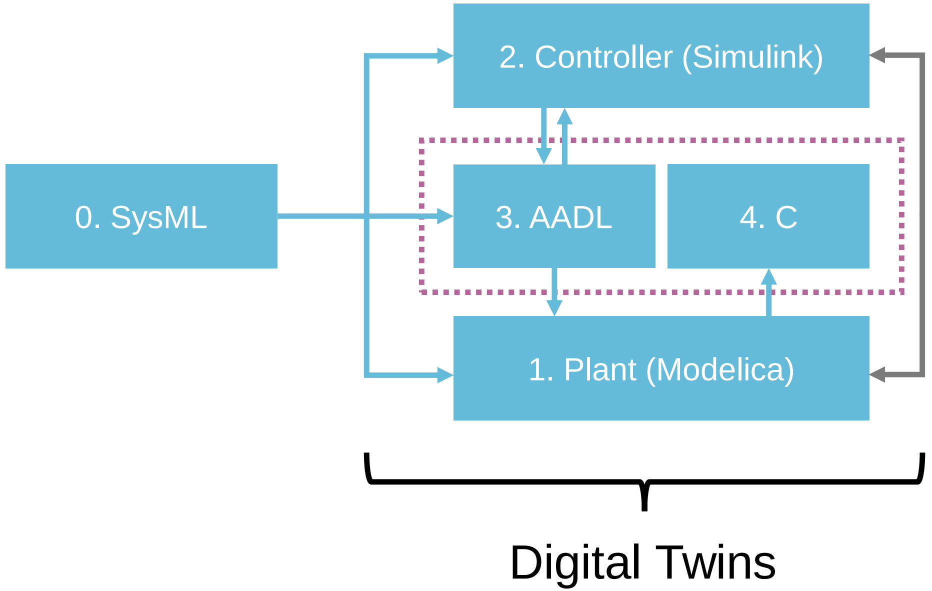


From ModDevOps to TwinOps



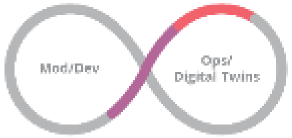
1-2-3-4: “mega-modeling” V&V

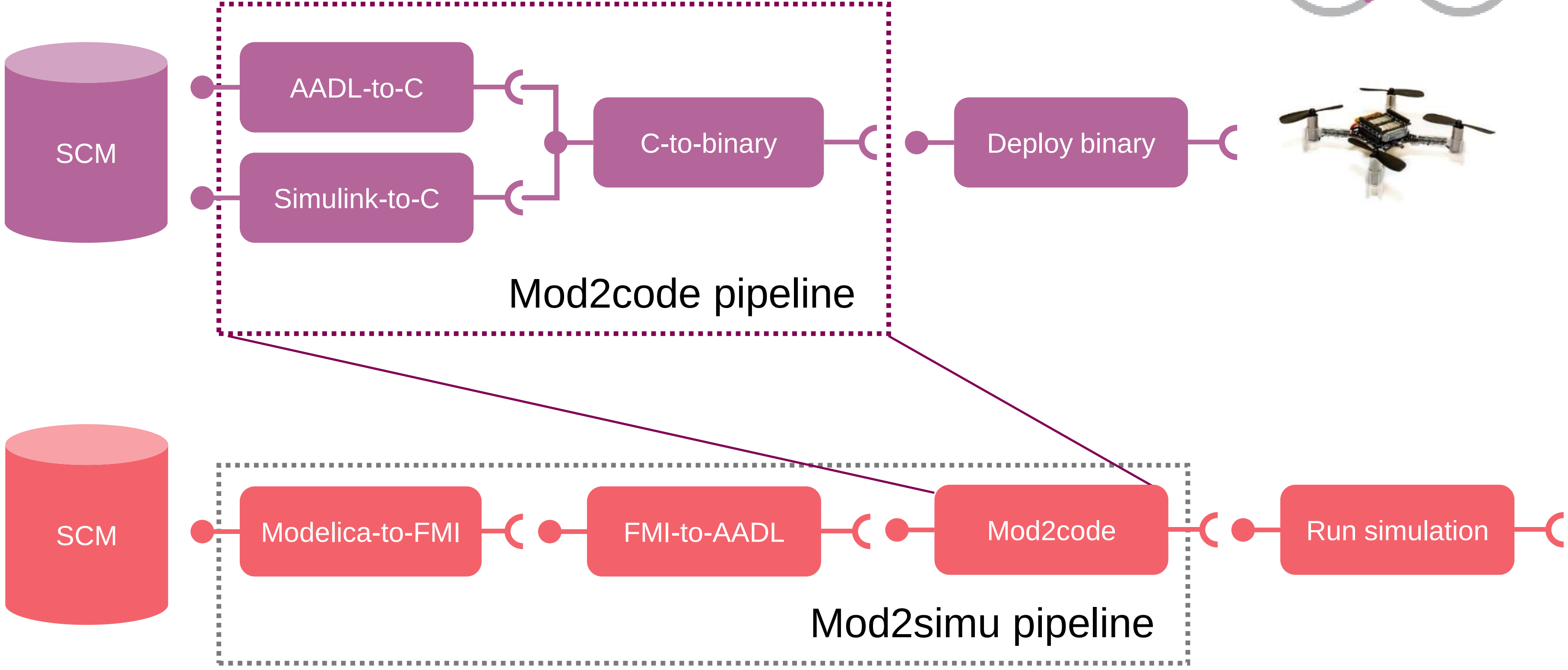
- 1-2: HLR validation
- 2-(3+4): validation of LLR
  1+(3+4): virtual integration



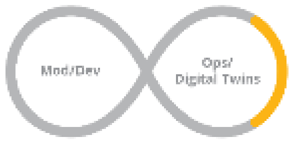
Digital Twins of UAV vs. UAV flying: validation of Modelica model, efficiency of the controller (overshoot verification) and timing verification of software.

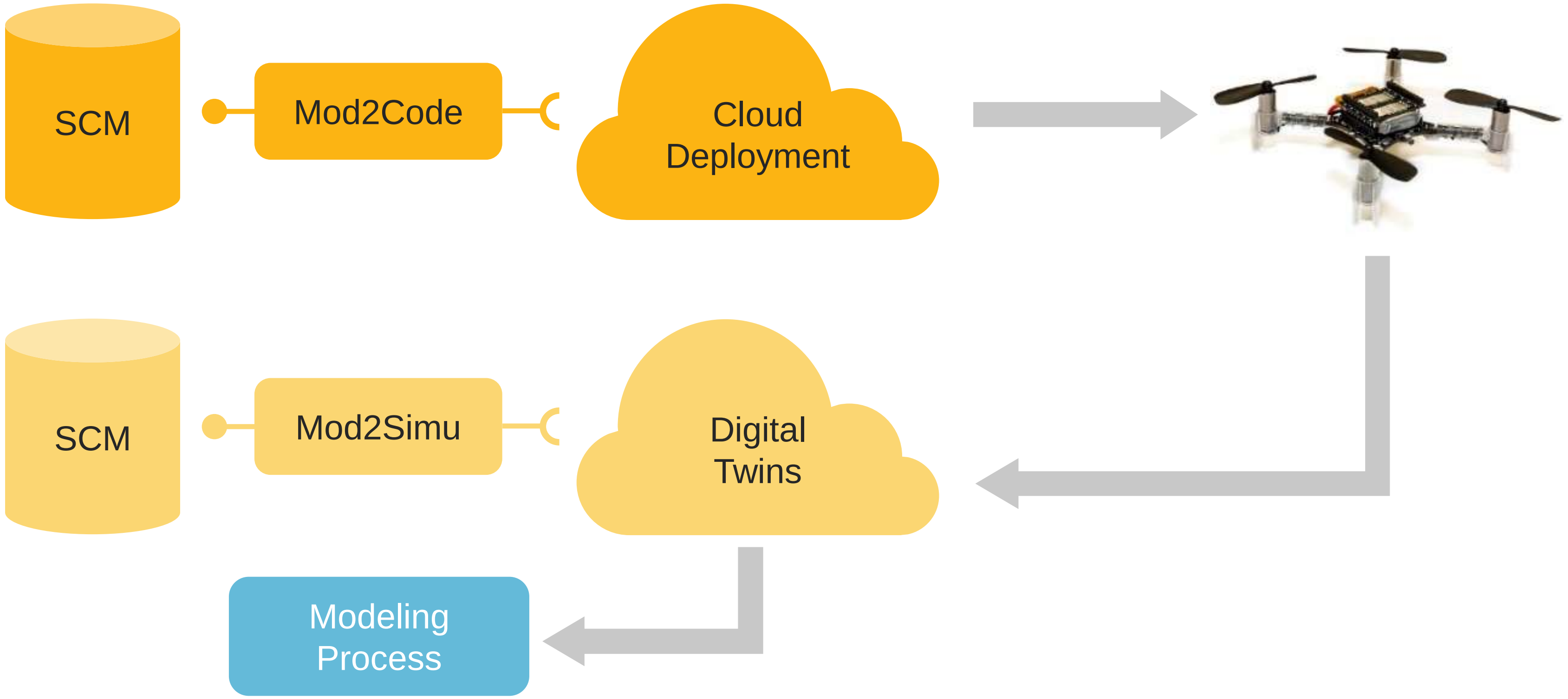
ModDevOps in Action – ModDevOps Pipeline #2



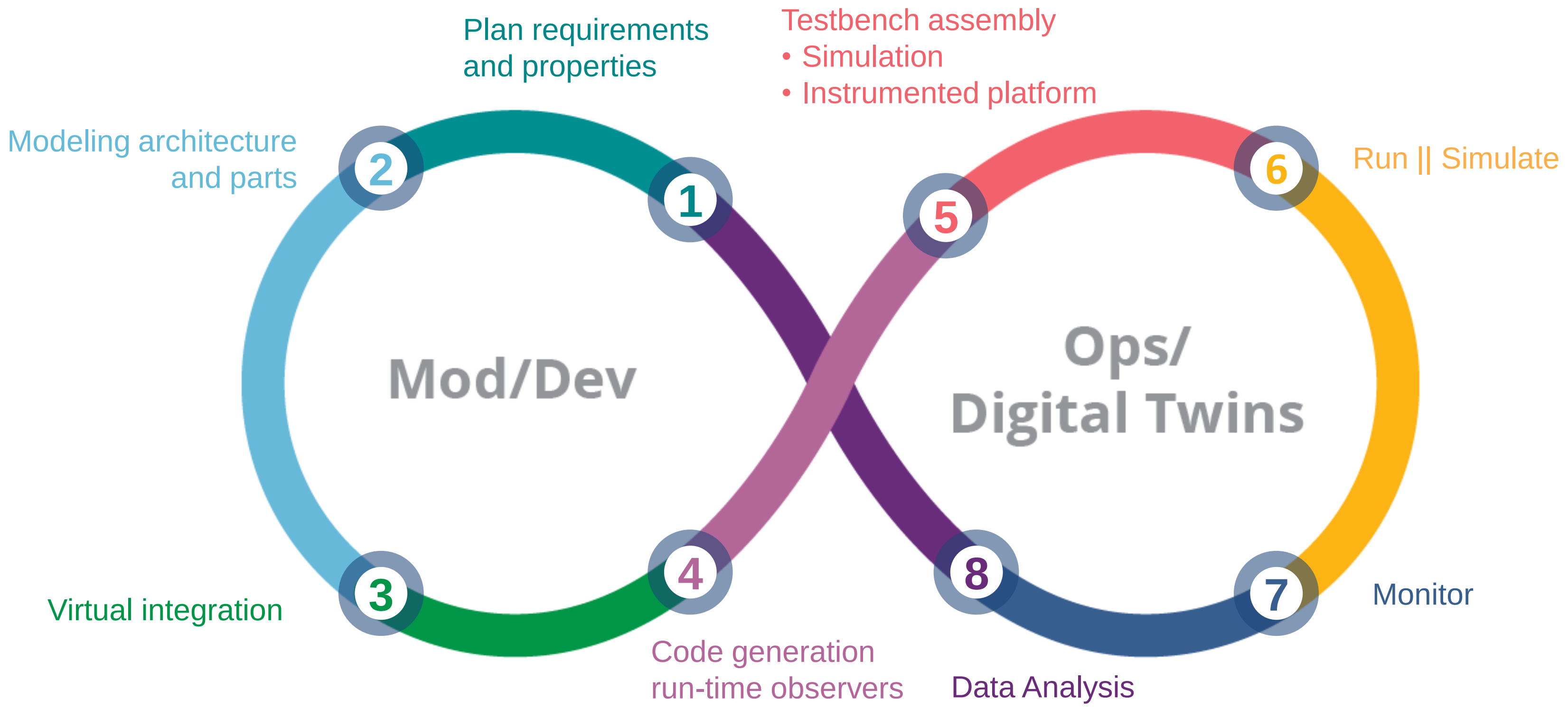


From ModDevOps to TwinOps





TwinOps: Continuous System Improvement through ModDevOps and Digital Twins


Plan requirements and properties
Testbench assembly
• Simulation
• Instrumented platform
Modeling architecture and parts
Run || Simulate
1
2
3
4
5
6
7
8
Mod/Dev
Ops/ Digital Twins
Virtual integration
Monitor
Code generation run-time observers
Data Analysis