



**AVSI System Architecture  
Virtual Integration (SAVI)  
Proof Of Concept  
Demonstration**

Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

Peter H Feiler  
April 2009



# Software Architecture Virtual Integration (SAVI)

## ■ Current

- ◆ *Active – Boeing, Airbus, Lockheed Martin, BAE Systems, DoD (Army, Navy), FAA, GE Aviation, Rockwell Collins, SEI/Carnegie Mellon*
- ◆ *Joining – Dassault-Aviation, Honeywell, JPL/NASA*

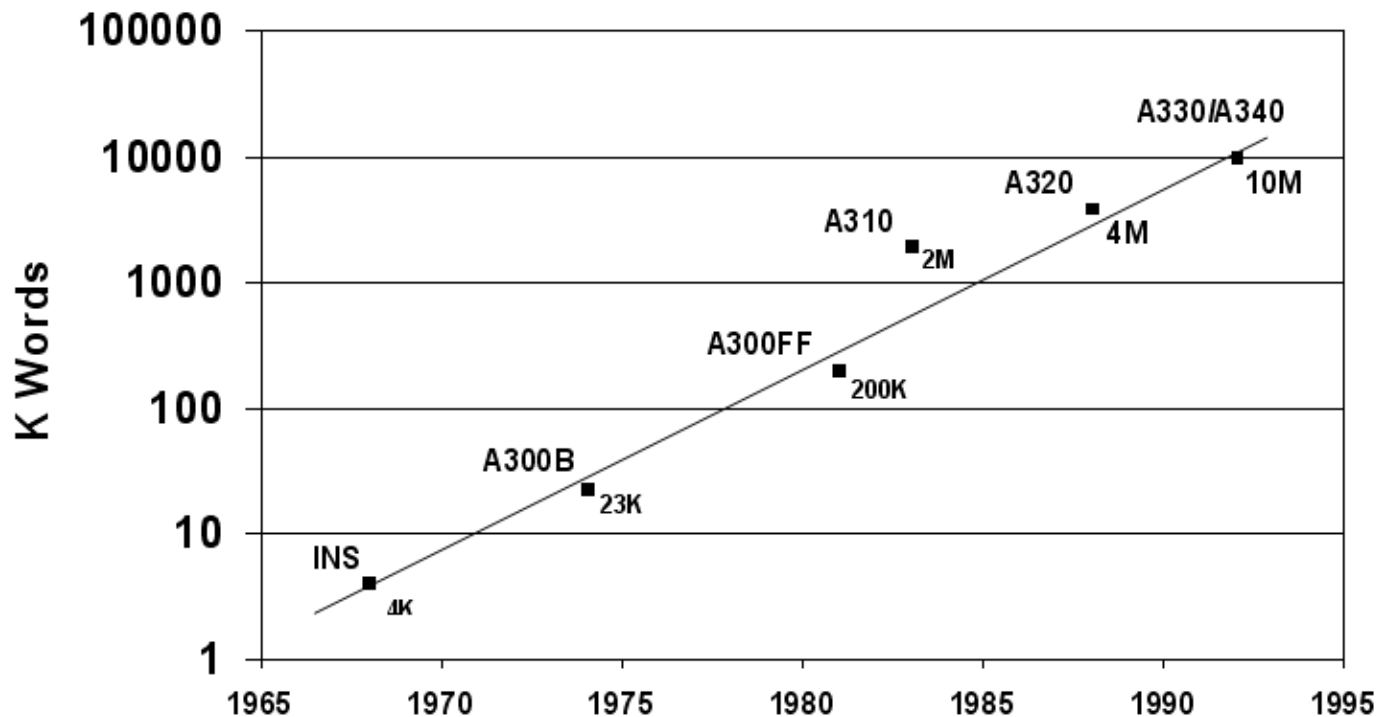
## ■ Potential

- ◆ *Current AVSI members – DoD (Air Force), Goodrich, Hamilton Sundstrand (UTC) {Sikorsky, P&W}*
- ◆ *Potential new members – General Dynamics, Meggitt, Northrup Grumman, Raytheon, Thales, Woodward*



# What's Coming? More Complexity!

**Airbus Code Growth. Boeing Numbers Similar.**



*J.P. Potocki De Montalk, Computer Software in Civil Aircraft, Sixth Annual Conference on Computer Assurance (COMPASS '91), Gaithersburg, MD, June 24-27, 1991.*



**Aerospace Vehicle  
Systems Institute**

# Why AVSI?

- **Rapid technological advancement and obsolescence combined with increasingly complex hardware and software evolution present integration problems affecting all of us**
  - ◆ ***It's not going to get better, it's only going to get worse***
    - ◆ Boeing and Airbus have published data showing doubling of size and complexity every two years
  - ◆ ***We can't afford to solve it alone***
  - ◆ ***We can't afford to solve it multiple times***
  - ◆ ***We can't afford not to solve it***



# ***SAVI Project Objective***

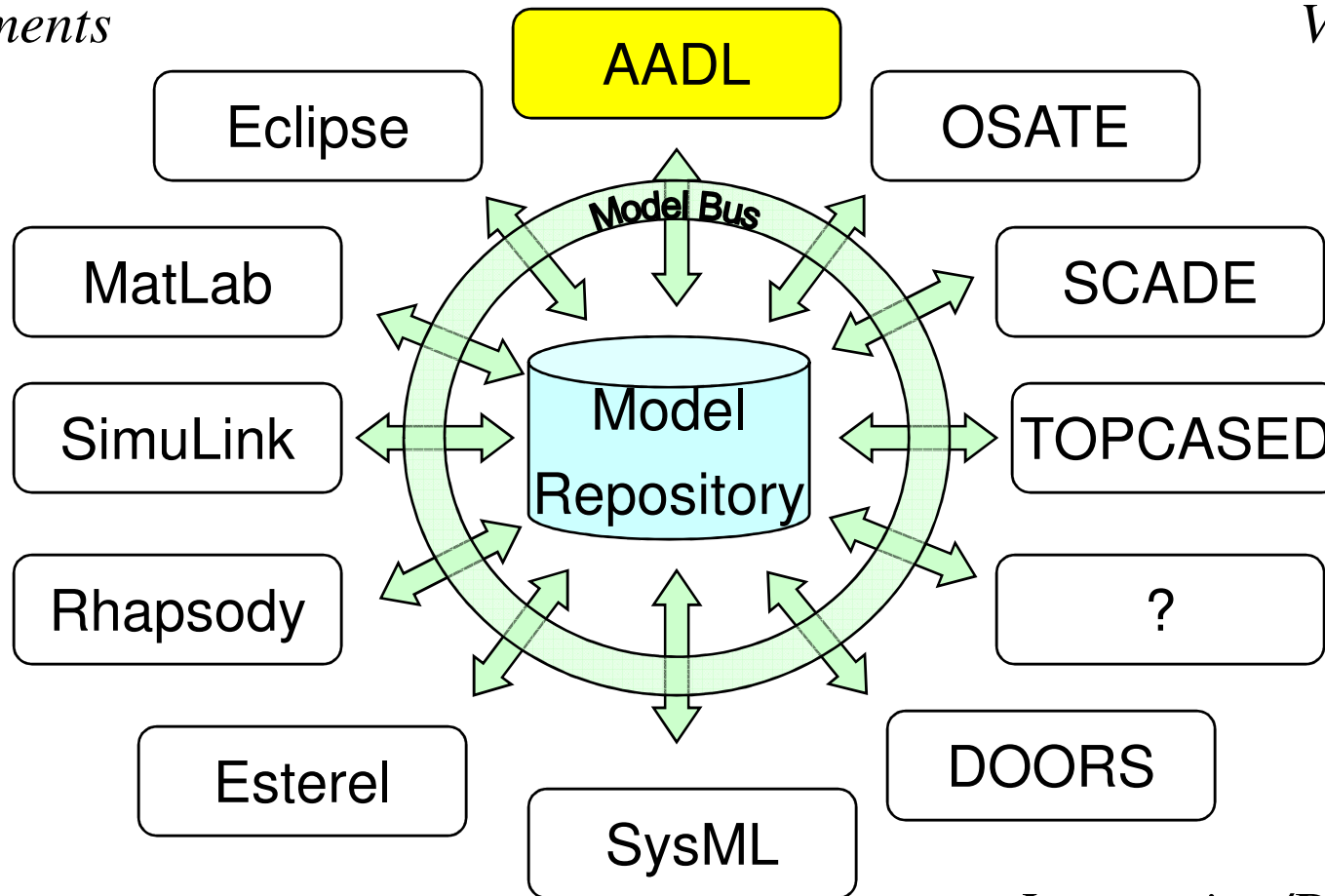
- **Overall Concept of Operations**
  - *Design and production based on early and continuous integration (virtual => physical)*
  - *Integrate, then build*
- **Objective**
  - *Shift architecting, design, and production activities to explicitly address integration issues early, reducing program execution risks, cycle time and cost*
- **Approach**
  - *Adopt/develop “virtual integration-based” software and system development processes with emphasis on integrating component-based, model-based and proof-based development*



# Single Multi-Aspect Model Repository & Model Bus

*Requirements*

*Verification*



*Design*

*Integration/Deployment*



Aerospace Vehicle  
Systems Institute

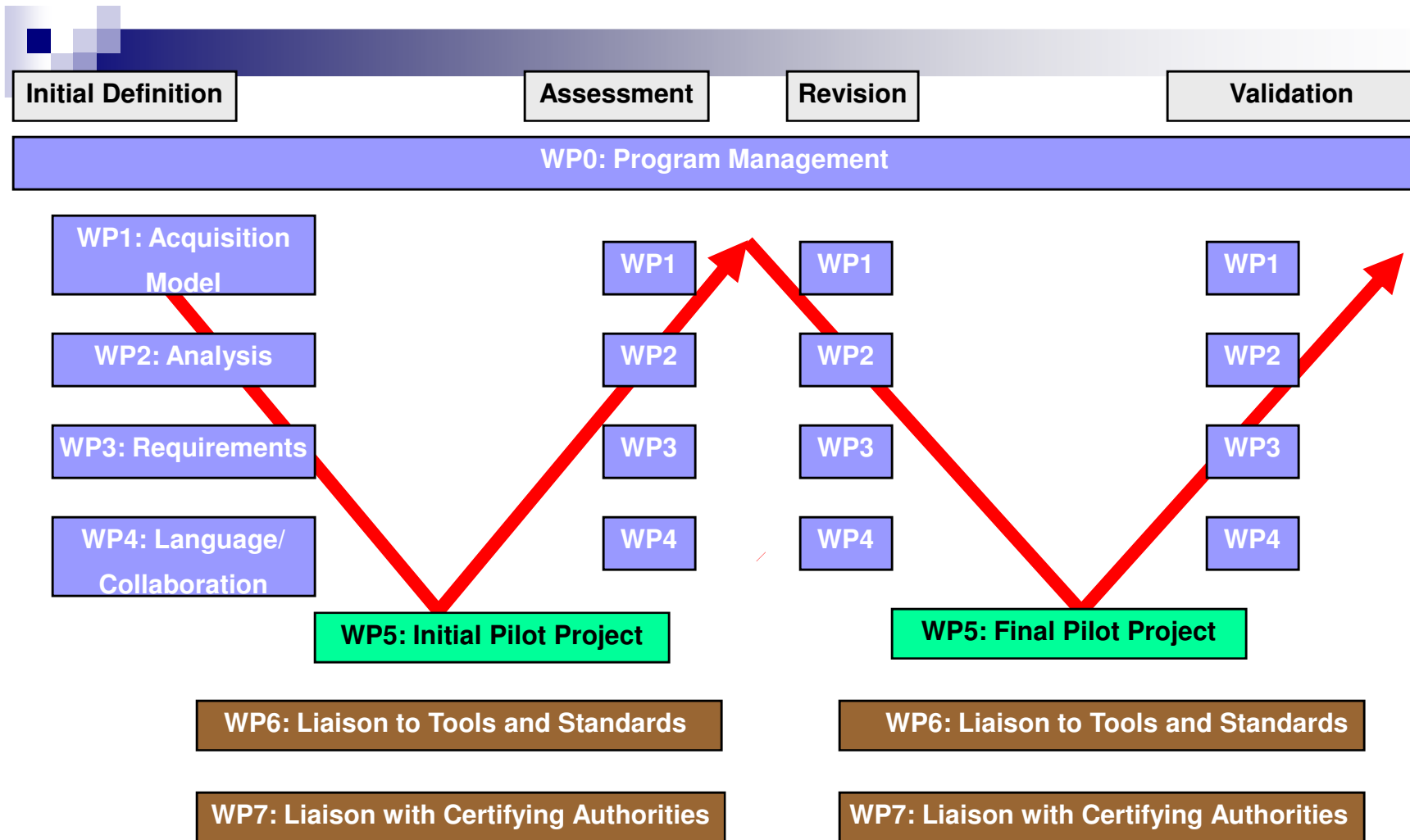
Slide 6

# Multi Phase Project

- **Proof of Concept (POC) Project**
  - *12 month project*
  - *As-is & To-Be process, ROI model, AADL-based demo*
  - *Initial POC demo within 2 months*
  - *Green light from Executive Board for pilot project*
- **Pilot Project**
  - *3-4 years, \$30+M*
  - *Establish practice infrastructure*
  - *Two phases: apply to production system, apply by product group*



# Pilot Project Work Packages (Notional)

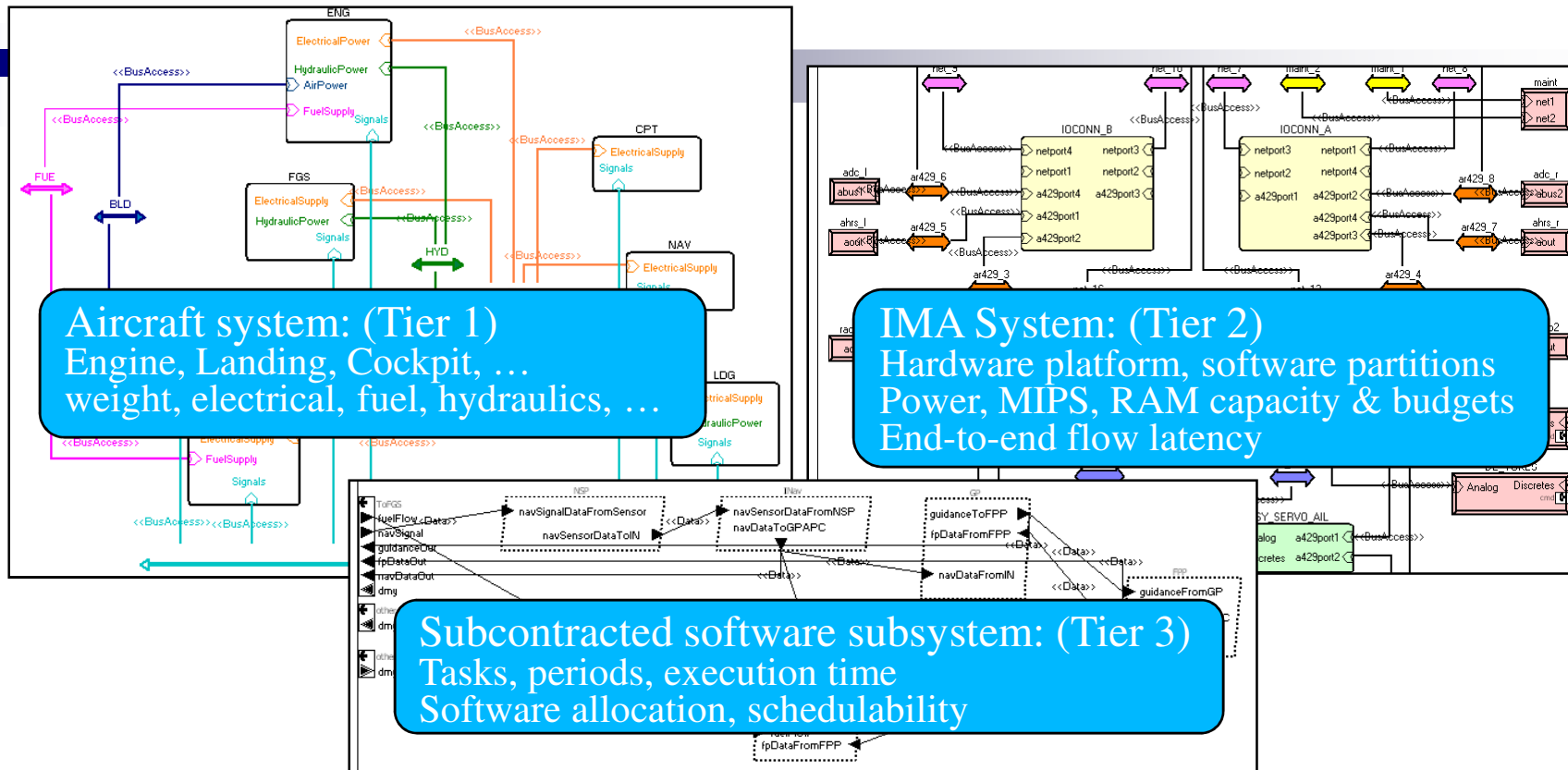


# PoC Prioritized Requirements

#	Requirement	Category
1	Establish Model Bus infrastructure	Process
2	Establish Model Repository Infrastructure	Process
3	Inform RoI estimates through AFE58 performance & results	Process
4	Analyses be conducted across the system	Analysis
5	Two or more analyses must be conducted	Analysis
6	Analyses be conducted at multiple levels of abstraction	Analysis
7	Analyses must validate system model consistency at multiple levels of abstraction	Analysis
8	Analyses must be conducted at the highest system level abstraction	Analysis
9	Model infrastructure must contain multiple model representations	Model
10	Model infrastructure must contain multiple communicating components	Model



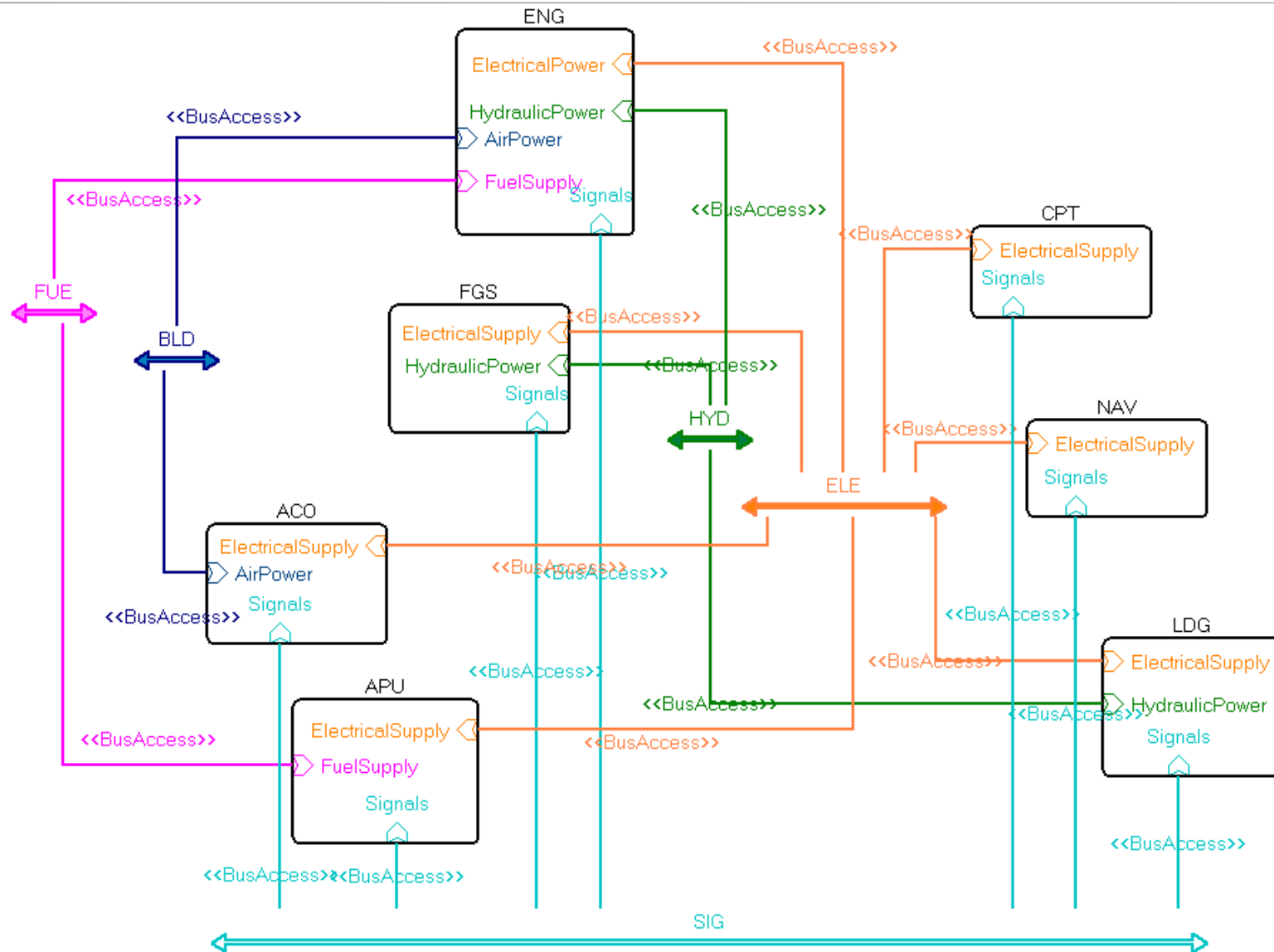
# Proof of Concept Demo



- Multi-tier system & software system
- Integrator & subcontractor virtual integration



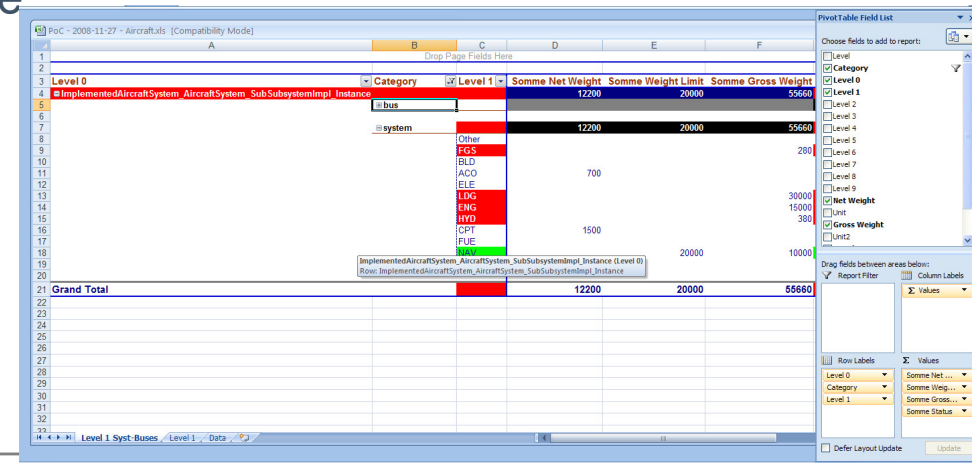
# Aircraft System Architecture



# Tier 1 Analyses

Based on Top-level Aircraft Model

- Weight analysis
  - Weight limit, gross weight, net weight
  - Problem: Engine weight limit exceeded by gross weight
  - Correction: increase Engine weight limit within Aircraft weight limit
  - Analysis by plug-in
  - Analysis by MS Excel via CSV file



Level 0	Category	Level 1	Somme Net Weight	Somme Weight Limit	Somme Gross Weight
ImplementedAircraftSystem_AircraftSystem_SubSubsystemImpl_Instance	bus		12200	20000	55660
	system		12200	20000	55660
	Other				2800
	ECS				
	ELD				
	ACO		700		
	ELE				
	LDG				30000
	ENG				15000
	HYD				380
	ICPT		1500		
	FUE			20000	10000
Grand Total			12200	20000	55660



# Tier 1 Analyses - 2

---

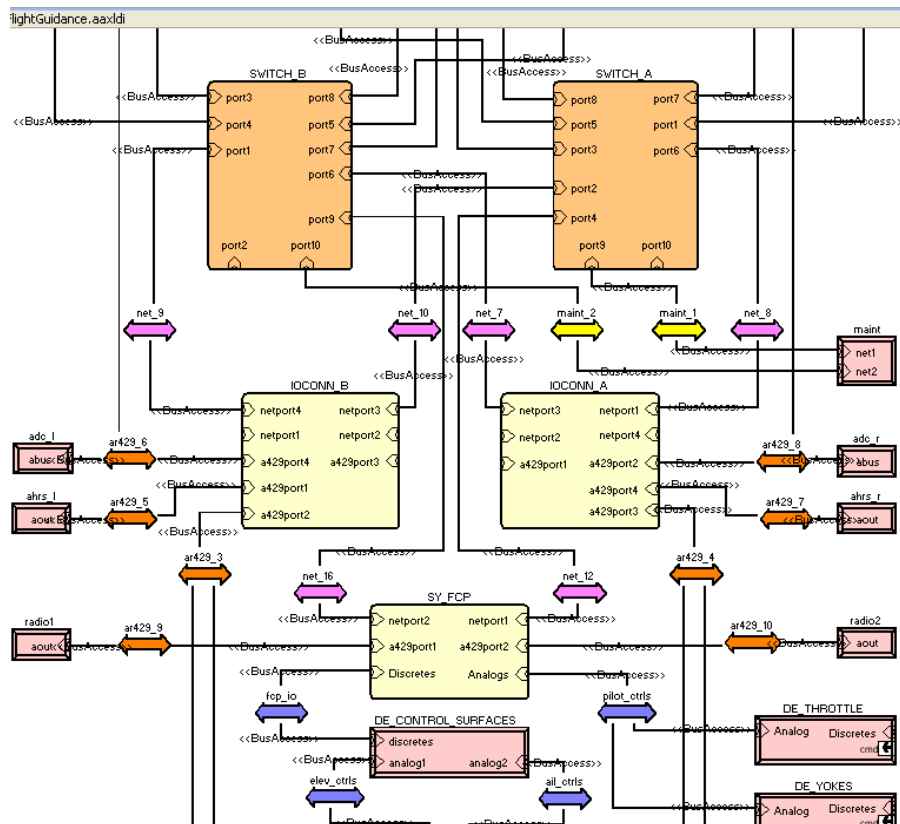
## -Power analysis

- Power capacity of bus
- Power supply by Engine and Aux Power Unit
- Power budget by power consumers
- Problem: total power supply over capacity
- Correction: reduce supply by APU
- Resulting problem: budgets exceed available power
- Options:
  - Reduce power consumption
  - Replace with higher capacity power bus => increased weight

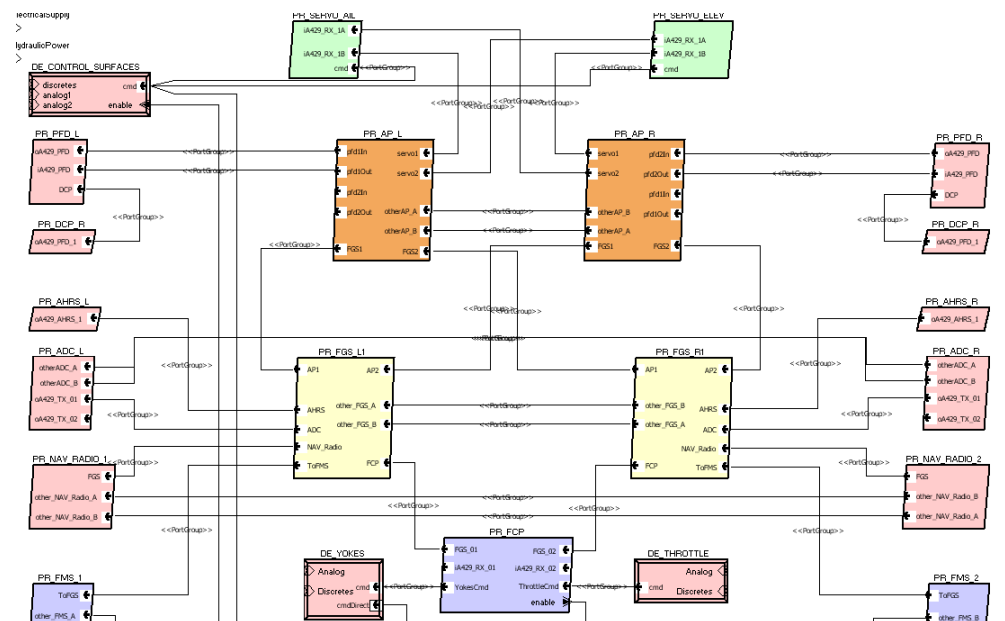


# Tier 2: Flight Guidance IMA Architecture

## Physical view



## Logical view



## Tier 2 Analyses: Subsystem partitions

---

### Flight Guidance IMA elaborated

- Subsystems to be contracted out
- Weight analysis revisited
- Power analysis revisited
- MIPS, RAM, ROM Resource Analysis
  - MIPS capacity & budget, RAM/ROM capacity & budget
  - Initial result ok, but incomplete budget assignments
- End to end latency analysis
  - Direct mode & IMA mode for stick to surface



# Tier 2 Flow Latency Analysis Results

## Two end-to-end flows

- Direct stick to surface
- Stick to surface via IMA

Analysis utilizes partition rate of subsystems & any flow space latency on application components

- Lower bound of worst case end-to-end latency
- Can be extended to determine latency jitter

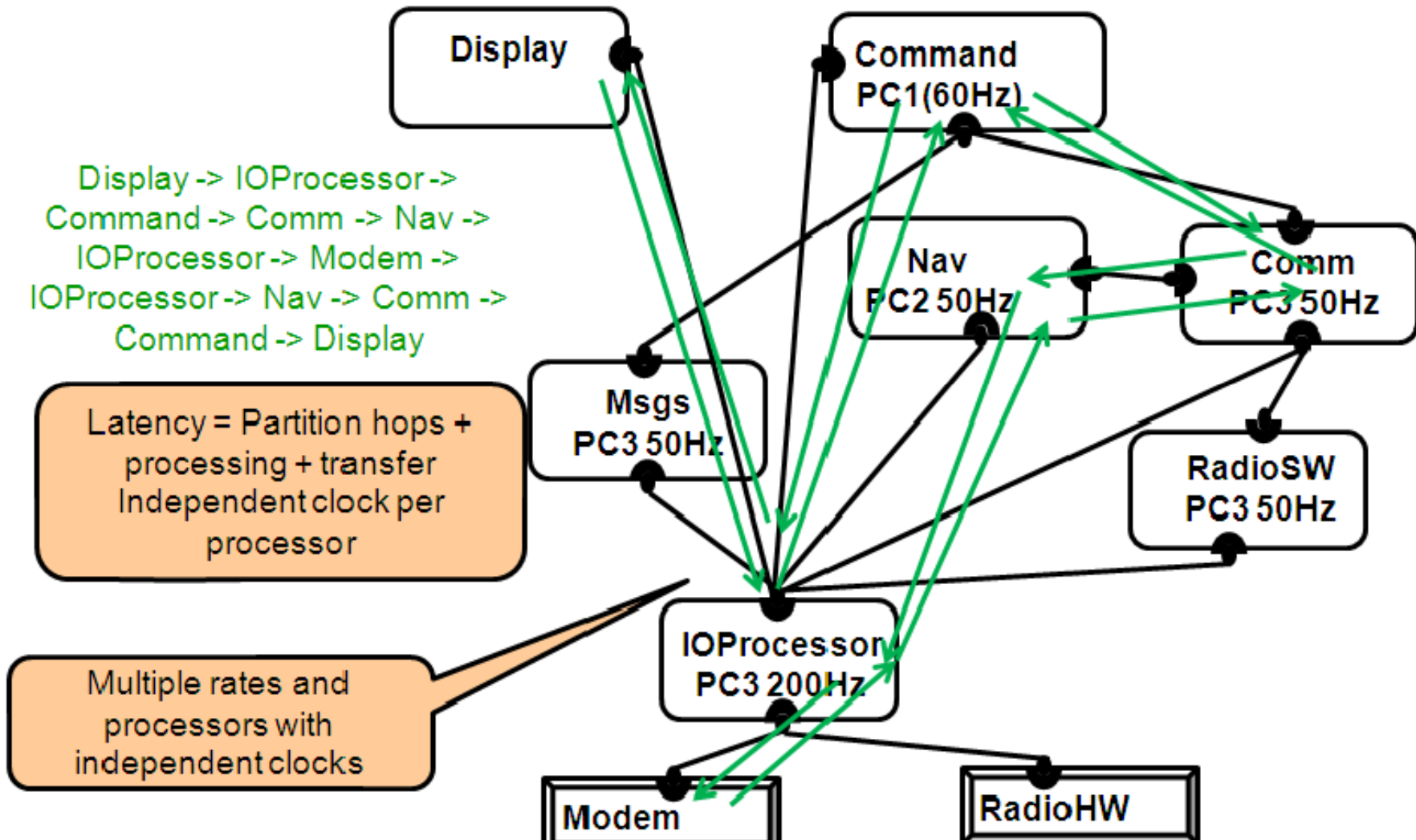
## IMA path exceeds requirement

- For synchronous system (all processors on same clock)
- For asynchronous systems (all processor

Flow Latency Analysis Marker (4 items)	
✘	End-to-end flow fStickToSurface_NormalModes calculated latency (ASynchronous)46.6 ms exceeds expected latency 25.0 ms
✘	End-to-end flow fStickToSurface_NormalModes calculated latency (Synchronous)43.6 ms exceeds expected latency 25.0 ms
i	End-to-end flow fStickToSurface_DirectMode calculated latency (ASynchronous)121.0 us is less than expected latency 150.0 us
i	End-to-end flow fStickToSurface_DirectMode calculated latency (Synchronous)121.0 us is less than expected latency 150.0 us



# Flow Use Scenario through Subsystem Architecture



# SAVI Model Repository Requirements

---

Support development process & system structure

Support integrator & subcontractor

Support versioning, configurations, development branches of systems and models

Support development teams within integrator & subcontractor

Support different views & representations

AADL Packages, refinement via  
extends, multiple variants

Versioning of AADL models &  
fragments



# Integrator – Subcontractor Negotiations

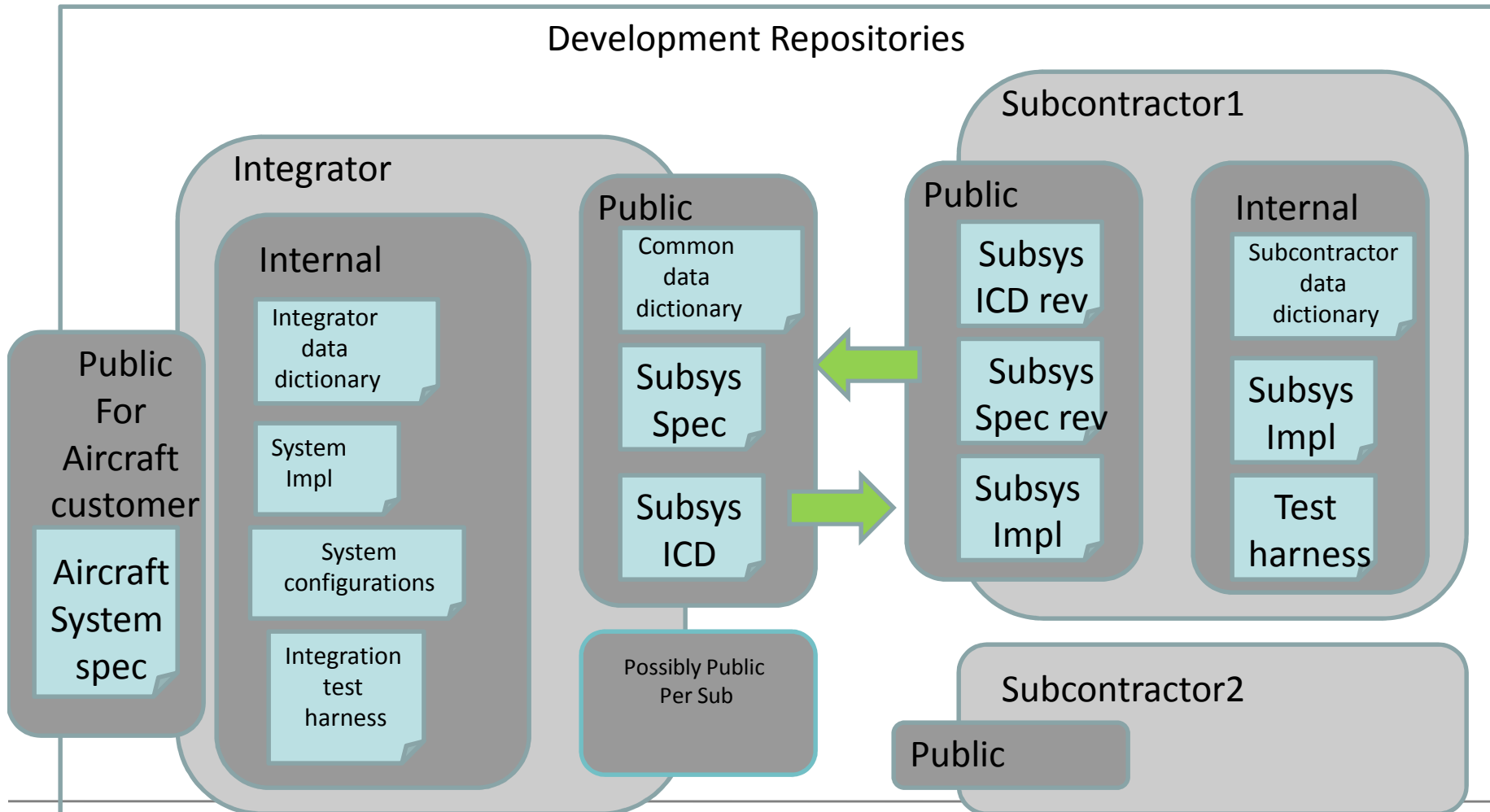
---

## Three phase process

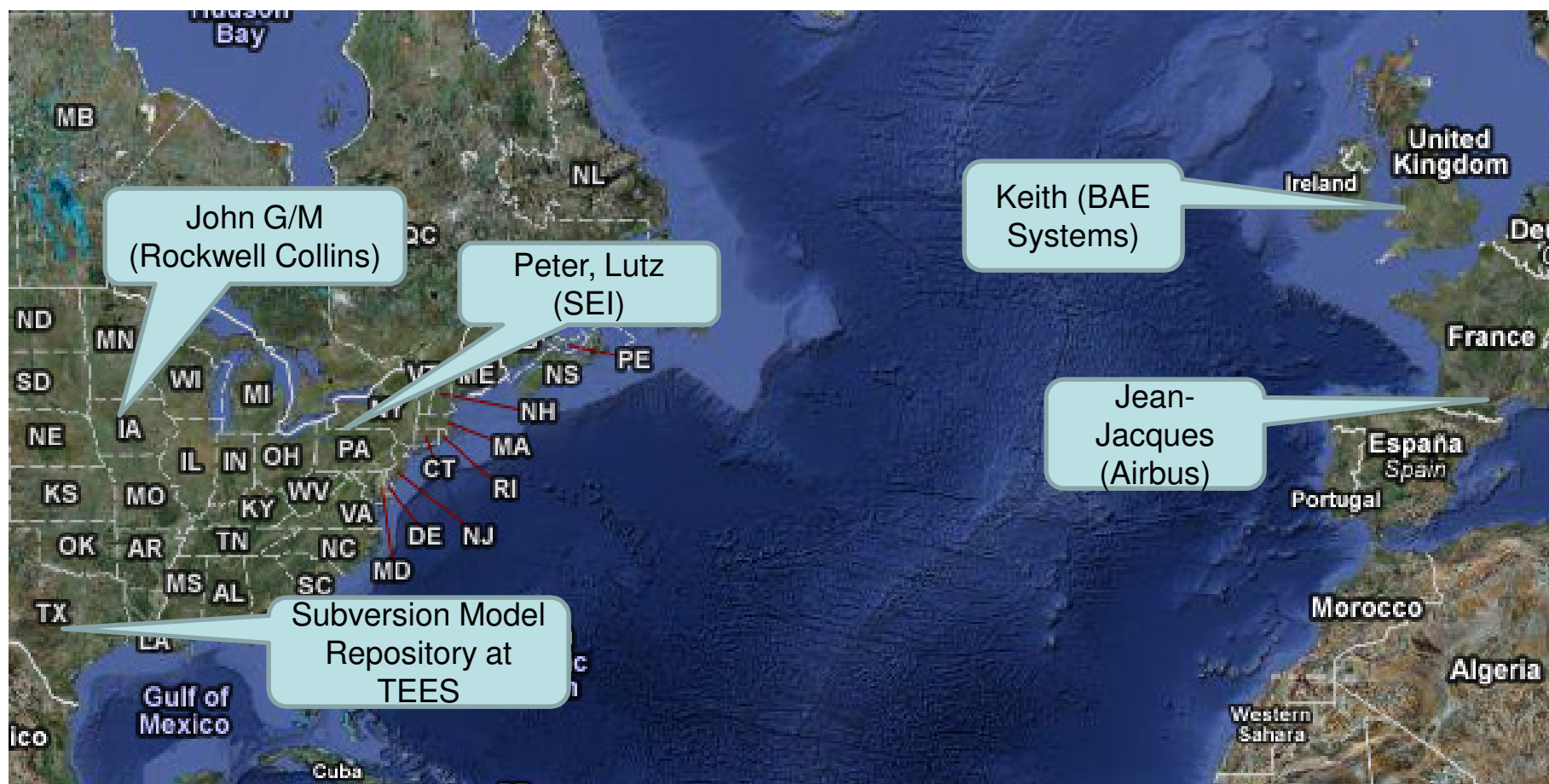
- Phase 1: System architecture specification & RFP
  - Integrator defines top-level system architecture & subsystem specification
- Phase 2: Joint subsystem specification refinement
  - Subcontractor evaluates subsystem specification & proposes subsystem specification refinements
  - Integrator evaluates impact of proposed changes
- Phase 3: Subsystem development & delivery
  - Subcontractor develops subsystem architecture
  - Subcontractor repeatedly delivers intermediate models for iterative virtual integration testing by integrator



# Integrator – Subcontractor Repositories



# Distributed POC Model Development



# Phase 2 Proposal Evaluation

## Subcontractor delivers

- Refined ICD (port group types)
- Port specifications
- Data types with base type, size, rate & measurement units
- Proposed mapping to ARINC 429 protocol in some cases

## Integrator evaluates proposals

- Integrates subcontractor subsystem specs into flight guidance & aircraft model
- Integrator runs functional integration analysis
- Integrator resolves conflicts between subcontractor specs

[-] [i] ARINC429 Connection Consistency Marker (6 items)
[X] Source number bits 3 and destination number bits 4 differ
[X] Source number bits 3 and destination number bits 4 differ
[X] Source number bits 4 and destination number bits 3 differ
[X] Source number bits 4 and destination number bits 3 differ
[X] Source Word ID 13 and Word ID 12 differ
[X] Source Word ID 13 and Word ID 12 differ
[+] [i] Instantiation Marker (20 items)
[-] [i] Port Connection Consistency Marker (14 items)
[X] Source base type uint3 and destination base type uint4 differ
[X] Source base type uint3 and destination base type uint4 differ



# Tier 3: Subcontractor Analyses

---

## Subcontractor models analyzed stand-alone

- Two IMA subsystem models down to thread level
- One subcontractor model of black box subsystem
  - Own hardware & software elaborated to executable Ada code

## Process & thread allocation analysis

- Allocation constraints
  - Not co-located
- Scheduling analysis: RMS vs. EDF, others
- Recording of allocation decisions

## Use of alternate scheduling analysis tool

- Explicitly recorded allocations
- Same results



# Virtual Integration Analyses - 1

## Subcontractor models integrated with Aircraft model

- Weight analysis revisited
- Power analysis revisited
  - Reduced power consumption
  - Subsystem below power budget
  - IMA power supply subsystem analysis with rollup
- MIPS, RAM, ROM Resource Analysis revisited
  - Budget rollup based on thread data

```
process ImplementedAircraftSystem_AircraftSystem_FlowSubSubsystemImpl_Instance.FGS.PR_FGS_L1 total 33.3 MIPS below budget 50.0 MIPS (33.3 % slack)
process ImplementedAircraftSystem_AircraftSystem_FlowSubSubsystemImpl_Instance.FGS.PR_FGS_R1 total 33.3 MIPS below budget 50.0 MIPS (33.3 % slack)
process ImplementedAircraftSystem_AircraftSystem_FlowSubSubsystemImpl_Instance.FGS.PR_FMS_1 total 111.3 MIPS below budget 120.0 MIPS (7.2 % slack)
process ImplementedAircraftSystem_AircraftSystem_FlowSubSubsystemImpl_Instance.FGS.PR_FMS_2 total 111.3 MIPS below budget 120.0 MIPS (7.2 % slack)
```

— Reduced total based on more accurate data

**i** MIPS capacity 800.000 MIPS : MIPS budget 544.333 MIPS

**i** RAM capacity 2.048 GB : RAM budget 140.028 MB

**i** ROM capacity 4.096 GB : ROM budget 632.280 MB



# Virtual Integration Analyses - 2

## Subcontractor models integrated with Aircraft model

- Latency analysis revisited
  - Latency has increased
  - Configure in one subsystem at a time to determine contributing subsystem task model

Flow Latency Analysis Marker (4 items)

✘	End-to-end flow fStickToSurface_NormalModes calculated latency (ASynchronous)196.6 ms exceeds expected latency 25.0 ms
✘	End-to-end flow fStickToSurface_NormalModes calculated latency (Synchronous)185.6 ms exceeds expected latency 25.0 ms
i	End-to-end flow fStickToSurface_DirectMode calculated latency (ASynchronous)121.0 us is less than expected latency 150.0 us
i	End-to-end flow fStickToSurface_DirectMode calculated latency (Synchronous)121.0 us is less than expected latency 150.0 us

- Examine latency computation trace as “Report” CSV file

A	B	C	D	E	F	G	H	I	J	K
owner	flow	model element name	deadline or c	sampling delay	partition delay	flow spec	additional	total (ms)	expected	
Inst. End2End	fStickToSurface_Norm	Subcomponent	DE_YOKES	0.0 us	0.0 us	0.0 us	500.0 us	500.0 us	500.0 us	25.0 ms
Inst. End2End	fStickToSurface_Norm	Connection	ImplementedAircraftSystem	0.0 us	0.0 us	0.0 us	0.0 us	500.0 us	500.0 us	25.0 ms
Inst. End2End	fStickToSurface_Norm	Subcomponent	THR_FCP:fYokesToFGS1	3000.0 us	5.0 ms	0.0 us	3000.0 us	3000.0 us	8.5 ms	25.0 ms
Inst. End2End	fStickToSurface_Norm	Connection	ImplementedAircraftSystem	0.0 us	0.0 us	0.0 us	0.0 us	3000.0 us	8.5 ms	25.0 ms
Inst. End2End	fStickToSurface_Norm	Subcomponent	THR_FGSMMain:PFCLAWS1	50.0 ms	100.0 ms	0.0 us	50.0 ms	50.0 ms	155.5 ms	25.0 ms
Inst. End2End	fStickToSurface_Norm	Connection	ImplementedAircraftSystem	0.0 us	0.0 us	0.0 us	0.0 us	50.0 ms	155.5 ms	25.0 ms
Inst. End2End	fStickToSurface_Norm	Subcomponent	THR_AP:fAPCLAWS1	3000.0 us	5.0 ms	0.0 us	3000.0 us	3000.0 us	158.5 ms	25.0 ms
Inst. End2End	fStickToSurface_Norm	Connection	ImplementedAircraftSystem	0.0 us	0.0 us	0.0 us	0.0 us	3000.0 us	158.5 ms	25.0 ms



# Virtual Integration Analyses - 3

## Integrator performs analyses on Flight Guidance IMA

- Scheduling analysis

Scheduling Analysis Marker (3 items)

- i** FlightGuidanceIMAConfiguration\_FlightGuidance\_a4\_fullybound\_Instance.CPU\_1.cpu is schedulable with 55.7 % utilization
- i** FlightGuidanceIMAConfiguration\_FlightGuidance\_a4\_fullybound\_Instance.CPU\_2.cpu is schedulable with 75.0 % utilization
- i** FlightGuidanceIMAConfiguration\_FlightGuidance\_a4\_fullybound\_Instance.CPU\_3.cpu is schedulable with 54.0 % utilization

- Network bandwidth analysis

- i** Total Bus bandwidth budget 22.0 Kbps of bound tasks with loopback within bandwidth capacity 10000.0 Kbps of net\_16
- i** Total Bus bandwidth budget 22.0 Kbps of bound tasks within bandwidth capacity 10000.0 Kbps of net\_12
- i** Total Bus bandwidth budget 22.0 Kbps of bound tasks within bandwidth capacity 10000.0 Kbps of net\_16
- i** Total Bus bandwidth budget 3.0 Kbps of bound tasks with loopback within bandwidth capacity 13.3 Kbps of ar429\_10
- i** Total Bus bandwidth budget 3.0 Kbps of bound tasks with loopback within bandwidth capacity 13.3 Kbps of ar429\_9
- i** Total Bus bandwidth budget 3.0 Kbps of bound tasks within bandwidth capacity 13.3 Kbps of ar429\_10
- i** Total Bus bandwidth budget 3.0 Kbps of bound tasks within bandwidth capacity 13.3 Kbps of ar429\_9
- i** Total Bus bandwidth budget 30.0 Kbps of bound tasks with loopback within bandwidth capacity 100.0 Kbps of ar429\_3
- i** Total Bus bandwidth budget 30.0 Kbps of bound tasks with loopback within bandwidth capacity 100.0 Kbps of ar429\_4



# Possible Other Demos

---

## Architecture consistency checks

- Application data types, measurement units, base types
- Required connections & hardware connectivity
- System Tier consistency
- Consistency between logical and physical connections

## Security analysis

## Safety criticality analysis

## Fault propagation & isolation analysis

## Model checking of redundancy mode logic

## Auto generation of runtime system



# Recent SEI Customer Projects

---

## AVSI SAVI

- Proof of Concept pilot of multi-tier modeling and analysis of aircraft architecture including integrator/subcontractor support

## NASA IV&V

- Case study of JPL Mission Data System reference architecture and Validation & Verification Framework

## US Army AMRDEC

- Comparative modeling of six CAAS IMA helicopter architectures

## PEO Aviation

- Modeling and analysis of Apache helicopter IMA architecture in context of SEI ATAM

## Automotive supplier

- Multi-dimensional variation of embedded software subsystem



# Towards Architecture Centric Engineering

---

Build on architecture tradeoff analysis (e.g., SEI ATAM)

- Provides focused evaluation method
- MBE/AADL provides quantitative analysis & starter models to build on

Project reviews & root cause analysis

- Identify systemic risks in problem systems & in technology migration
- AADL provides semantic framework to identify issues and potential mitigation strategies

Architecture documentation of existing systems

- Leverage existing design data bases
- Challenge: abstract away from design details (“what” instead of “how”)

System and software assurance

- Provides structured approach to safety/dependability assurance
- MBE/AADL provides evidence based on validated models



# AADL & Other Standards

---

## AADL & OMG MARTE

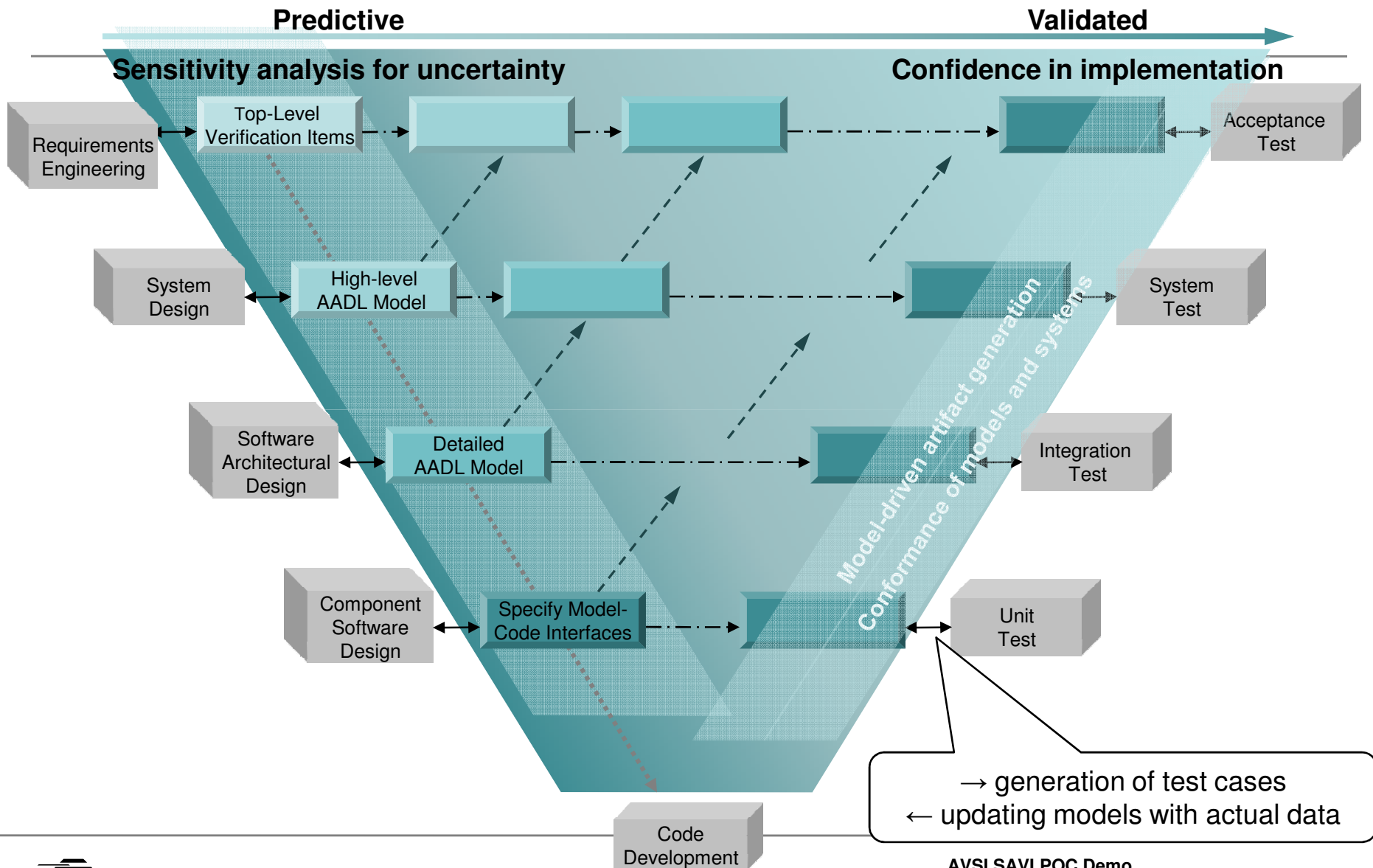
- MARTE met with SAE AADL during RFP in 2004
- Joint AADL UML profile effort
- AADL sub-profile appendix in MARTE Document (in ballot 2009)

## Embedded systems & System engineering

- Meeting of minds: technical leads of AADL & SysML (Dec 2008)
- Coordination: AADL, MARTE, SysML (April 2009)
- Collaboration: AADL, MARTE, SysML, INCOSE cross membership & joint meetings



# Increased Confidence through Virtual Integration



# Predictability through Virtual Integration

---

## Reduce the risks

- Analyze system early and throughout life cycle
- Understand system wide impact
- Validate assumptions across system

## Increase the confidence

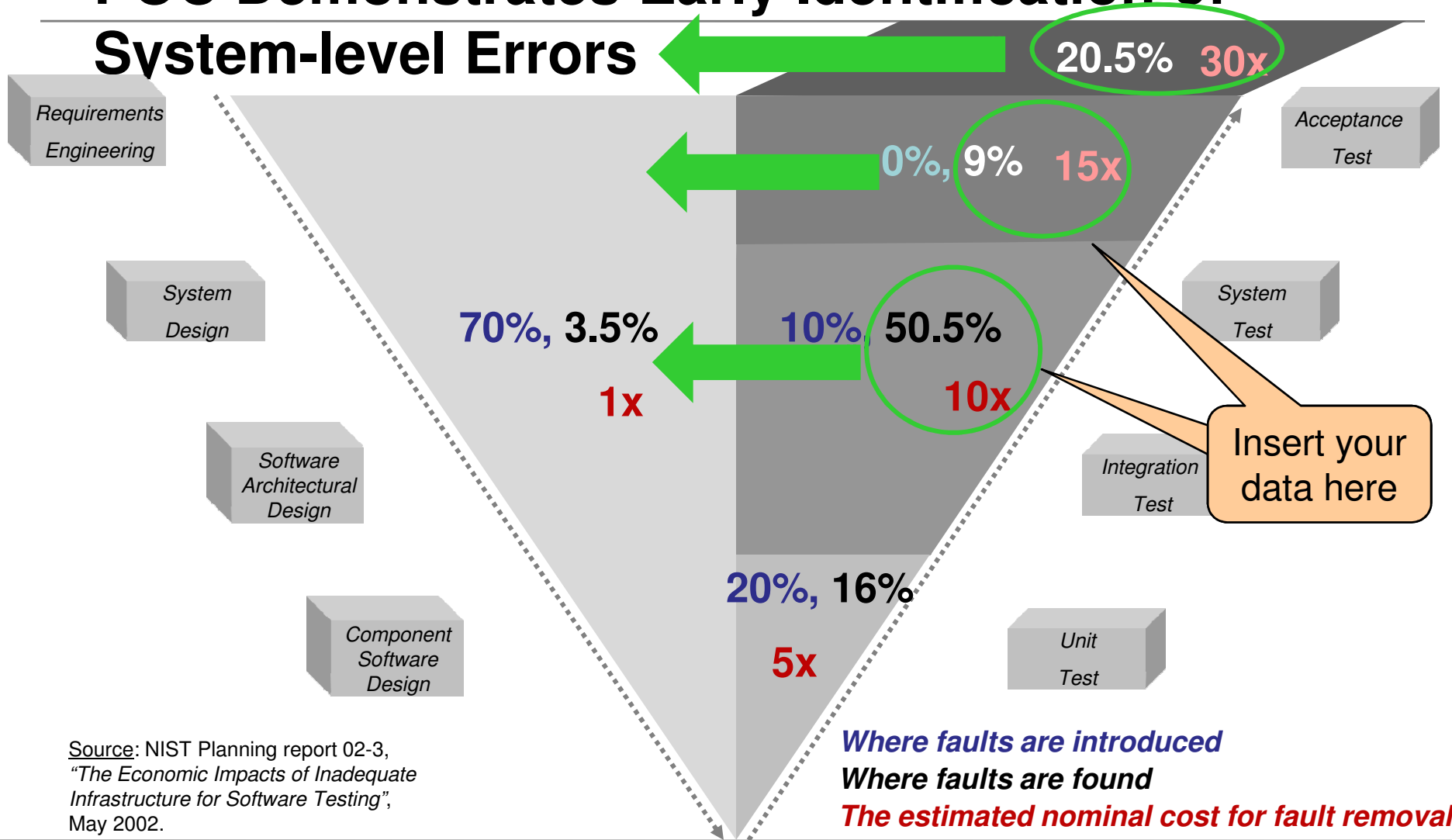
- Validate models to complement integration testing
- Validate model assumptions in operational system
- Evolve system models in increasing fidelity

## Reduce the cost

- Fewer system integration problems
- Fewer validation steps through use of validated generators



# POC Demonstrates Early Identification of System-level Errors





**Software Engineering Institute**

**Carnegie Mellon**

Peter H Feiler

[phf@sei.cmu.edu](mailto:phf@sei.cmu.edu)

## NO WARRANTY

---

**THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.**

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

