A High-Assurance Partitioned Development Environment

David Greve and Matthew Wilding

Rockwell Collins Advanced Technology Center
Cedar Rapids, IA
{dagreve, mmwildin}@rockwellcollins.com

John Launchbury and Peter White

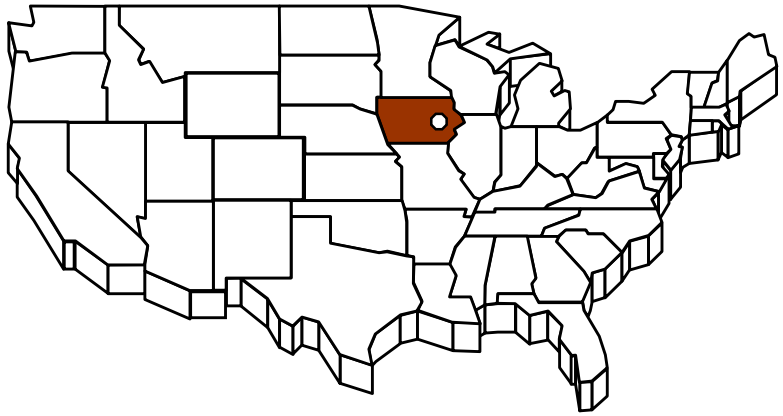
Galois Connections, Inc.

HCSS 03
April 2003

Rockwell Collins

- **Advanced Communication and Aviation Equipment**
  - Air Transport, Business, Regional, and Military Markets
  - $2.5 Billion in Sales

- **Headquartered in Cedar Rapids, IA**
  - 17,000 Employees Worldwide
  - Advanced Technology Center
    - Advanced Computing Systems



Advanced Technology Center

Air Transport
Business and Regional
Displays
SATCOM
Flight Guidance Systems
Data Management Systems
Passenger Systems

Military Joint Strike
JTRS
KC-135
GPS / Navigation

Commercial Systems

Government Systems

Advanced Technology Center

- The **Advanced Technology Center** (**ATC**) identifies, acquires, develops and transitions value-driven technologies to support the continued growth of Rockwell Collins.
- The **Advanced Computing Systems** department addresses emerging technologies for high assurance computing systems with particular emphasis on embedded systems.
- The **Formal Methods Center of Excellence** applies mathematical tools and reasoning to the problem of producing high assurance systems.

Outline

- Integrated Modular Avionics
- Intrinsic Partitioning
- Partitioning for Security
- Formal Verification
- AAMP7 Development Environment

"Security is about separation
Comp;uters are about sharing"

-Brian Snow, Dept. of Defense
April 1, 2003

Federated Architecture

- One Computer System For Each Unique Function
  - Autopilot
  - Flight Management
  - Displays
- Limited Dependencies Between Functions
  - Exchange of Sensor and Control Data
  - Provides Strong Functional Isolation
- System Certification
  - All Components Considered Together
  - Verification of Components Acting Together
  - “You don’t certify a single application, you certify an entire system”

Firewalls
Key Management
Encryption

Integrated Modular Avionics (IMA)

- One Computer System For Many Distinct Functions
  - Leverage Improved Computing Capability
  - Reduce Hardware Related Costs
- Incremental Certification
  - Functions verified ONCE, INDEPENDENTLY, and only to the LEVEL APPROPRIATE to their criticality
  - Composition of functions retains individual certification
  - Crucial for IMA

MILS

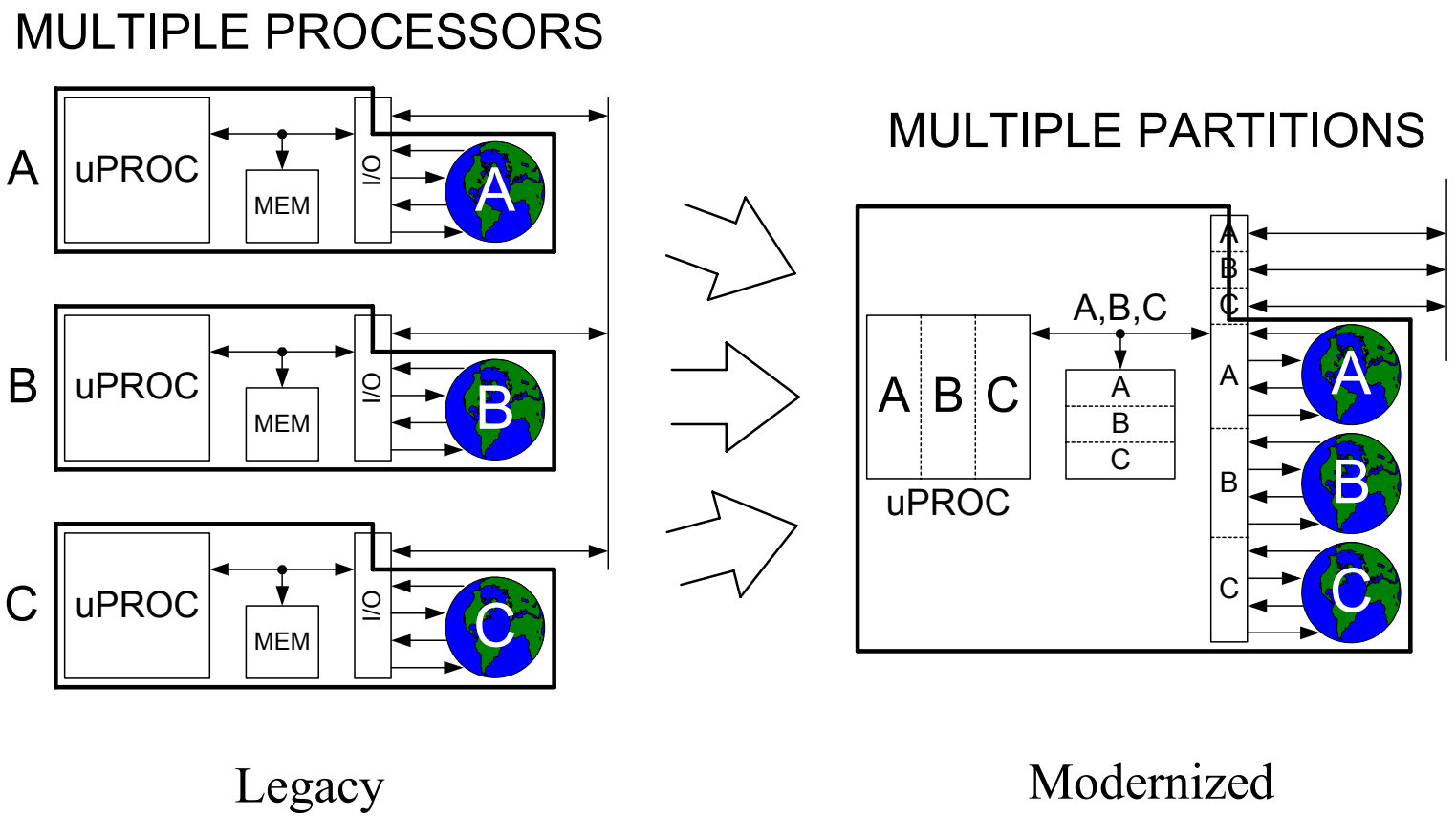
- What About Functional Interaction?
  - No longer physically isolated
  - Without isolation, must consider interaction
  - PARTITIONING provides necessary isolation

Partitioning

- **Partitioning**
  - Isolating, both in space and in time, two or more functions executing concurrently on the same computer system
  - Enables composition of two or more previously distinct functions onto a single computer system
- **Isolation**
  - **Spatial**
    - **Memory management unit**
    - **Provides Read/Write protection between partitions**
  - **Temporal**
    - **Periodic Partition switching**
    - **Watchdog Timer**

If You Can Keep Them Separate (Partitioning)
Then You Can Bring Them Together (Composition)

Conceptual System Composition



Real-Time Partitioning Considerations

- **Partition Latency**
  - Time Between Successive Executions of a Given Partition
  - Can Be Minimized by Increasing Partition Switch Rate
- **Partition Switch Overhead**
  - Processor Activity Associated with Partition Context Switching
  - Limits Maximum Partition Switch Rate
- **Interrupts**
  - Interrupts Cannot Change Partition Time Allocations
  - Interrupts Must Be Partitioned, Too.

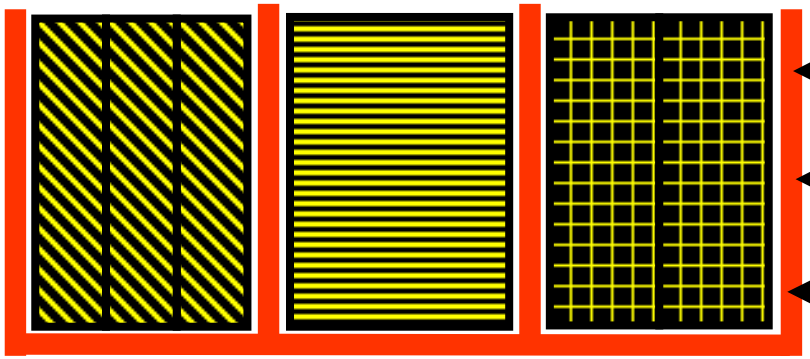
Outline

- Integrated Modular Avionics
- Intrinsic Partitioning
- Partitioning for Security
- Formal Verification
- Development Environment

Intrinsic Partitioning

- **Intrinsic Partitioning**
  - **Computing Platform Enforces Data Isolation**
  - **Technique Pioneered by Rockwell Collins, ATC**
  - **Provides Real-Time Performance**
  - **Addresses IMA Concerns**

Multi-Tasking OS



Functions

Variables

Call Stack

Heap

Operating System

BIOS

Process State

Scheduling Data Structures

Peripherals

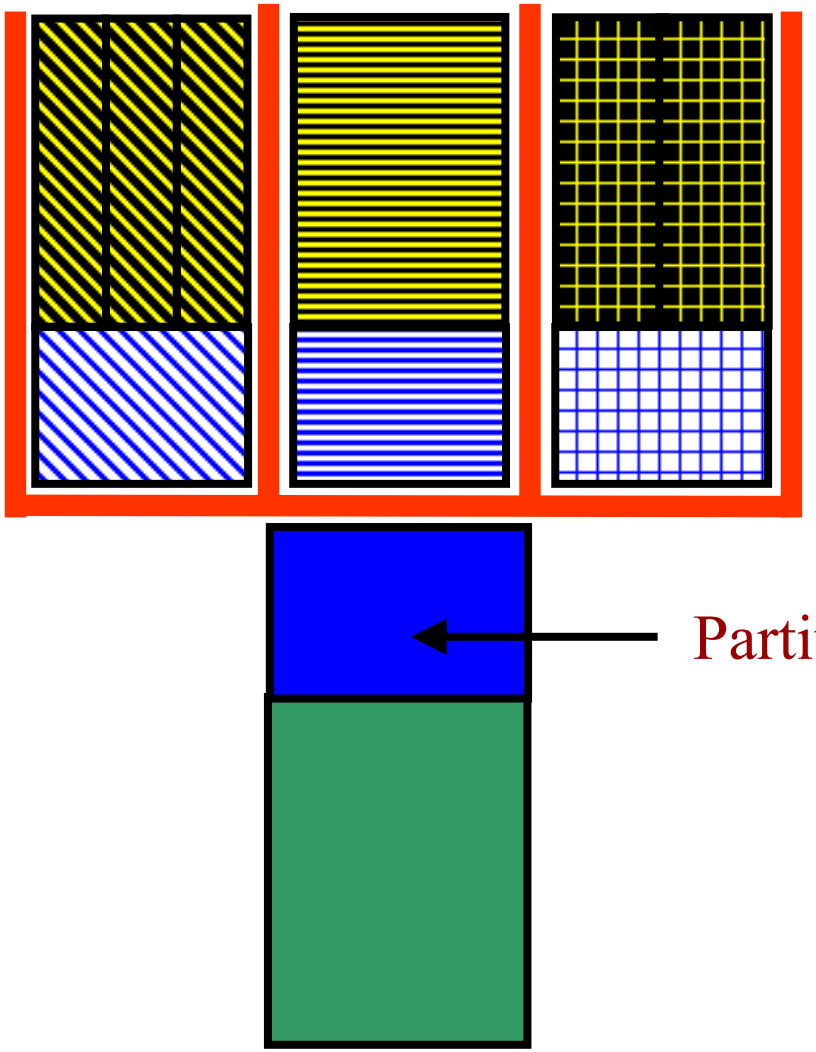
Hardware

System Data Structures

System Configuration

Processor Configuration

Micro Kernel Partitioning

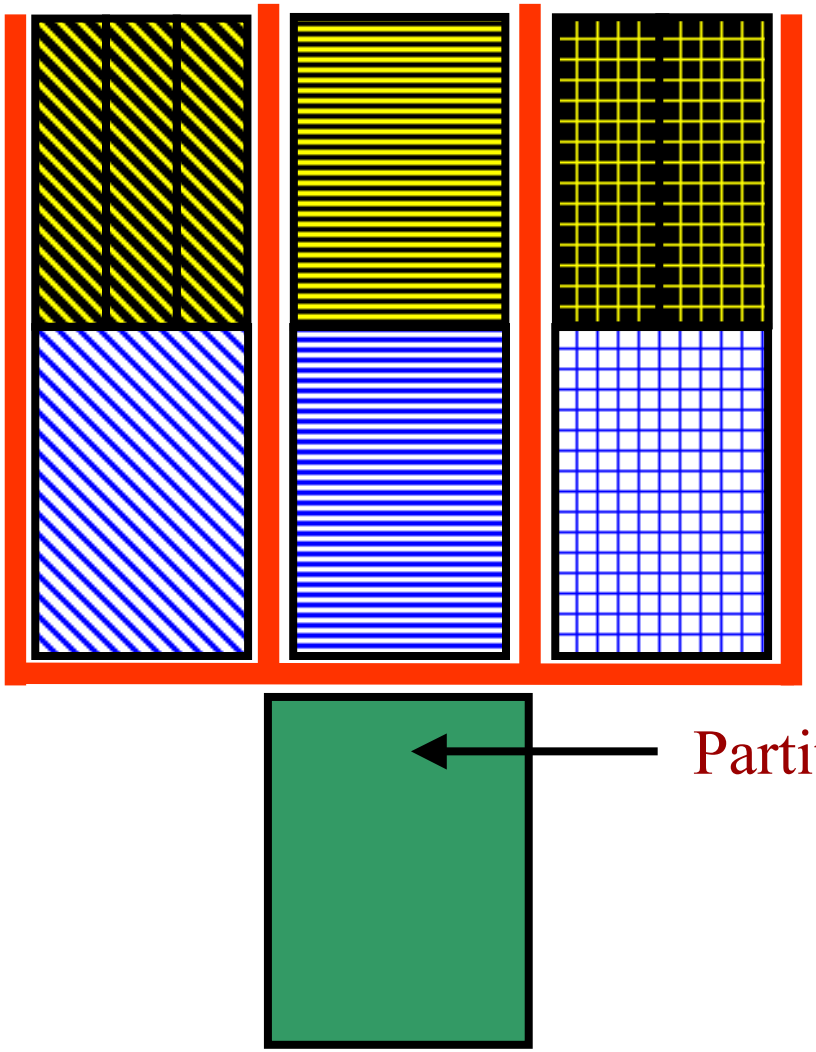


Intrinsic Partitioning

Functions

OS

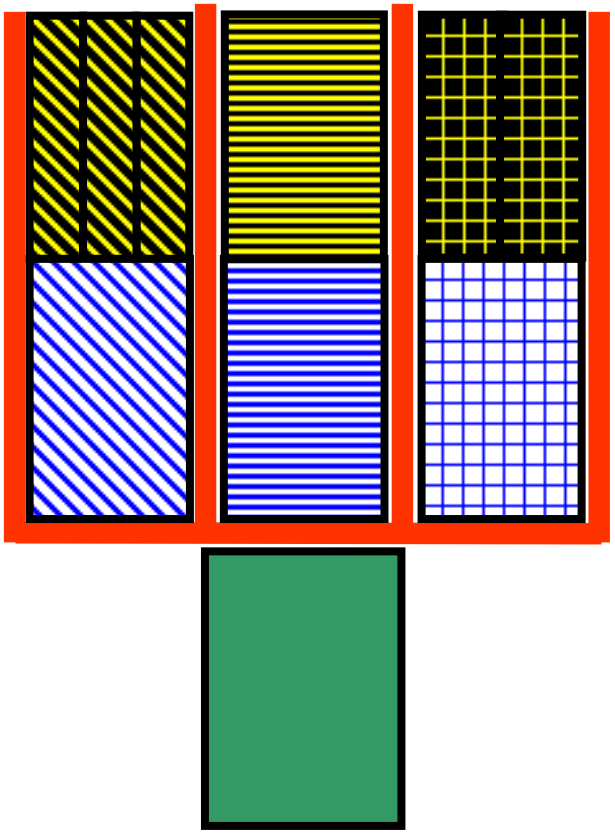
Hardware



Partitioning Data Structures

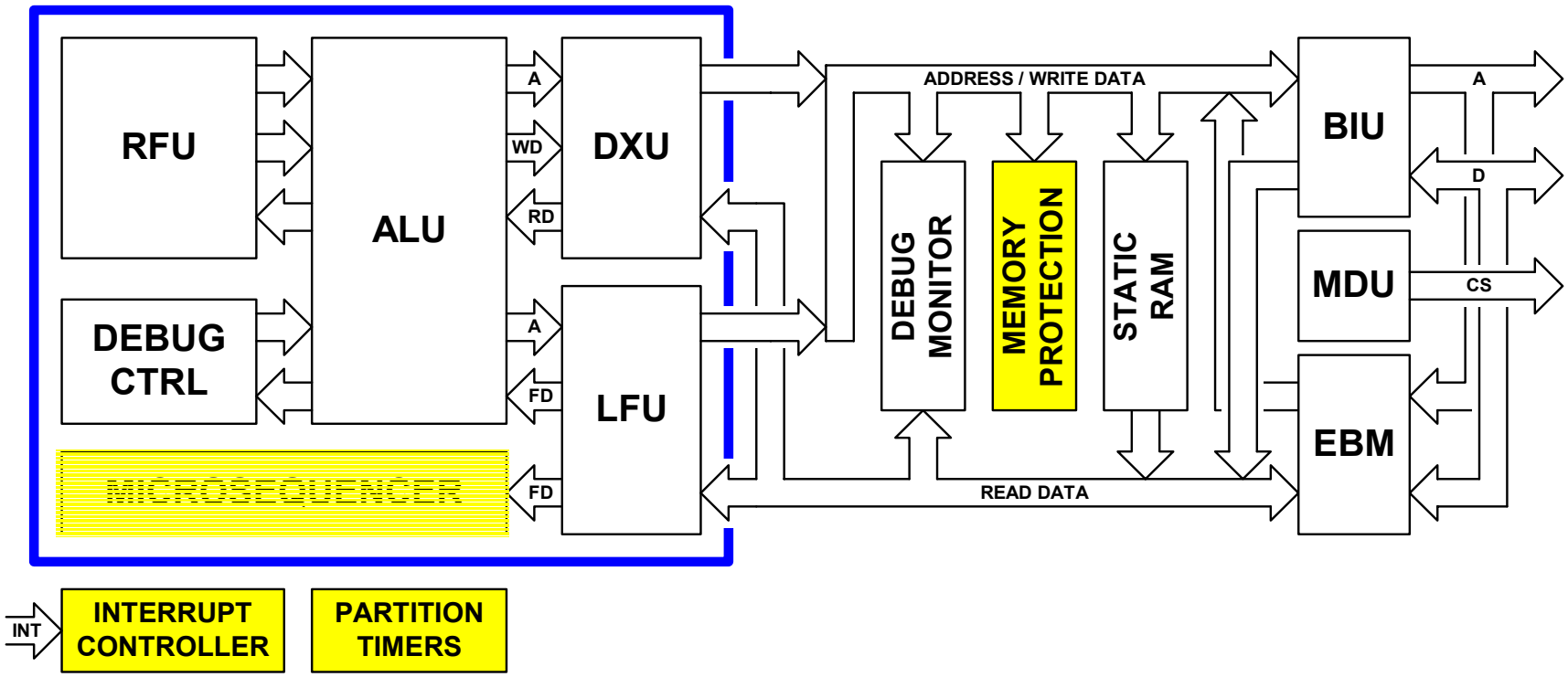
Intrinsic Partitioning

- **Micro-Coded Partitioning Kernel**
  - Minimal Code, Functionality, and State
  - Analyzable, Fast, and Efficient
- **Simple Data Structures**
  - Supports "Virtual Machine" Partitioning
    - Each Partition Has Its Own Operating System
  - Hierarchical Scheduling
- **Dedicated Interrupts**
  - Partition Switch Interrupt
  - Power Down Warning Interrupt
  - Access Violation Interrupt
  - ABORT (Mild Reset)
  - Partition-Aware Interrupts



- **Supports High Assurance, Evaluatable Architectures**

Partition Management Unit Architecture



- **Intrinsic Partitioning Implemented In JEM1**
  - functionality enforced with off-chip Partition Management Unit (PMU)
- **PMU Designed into AAMP7 microprocessor**

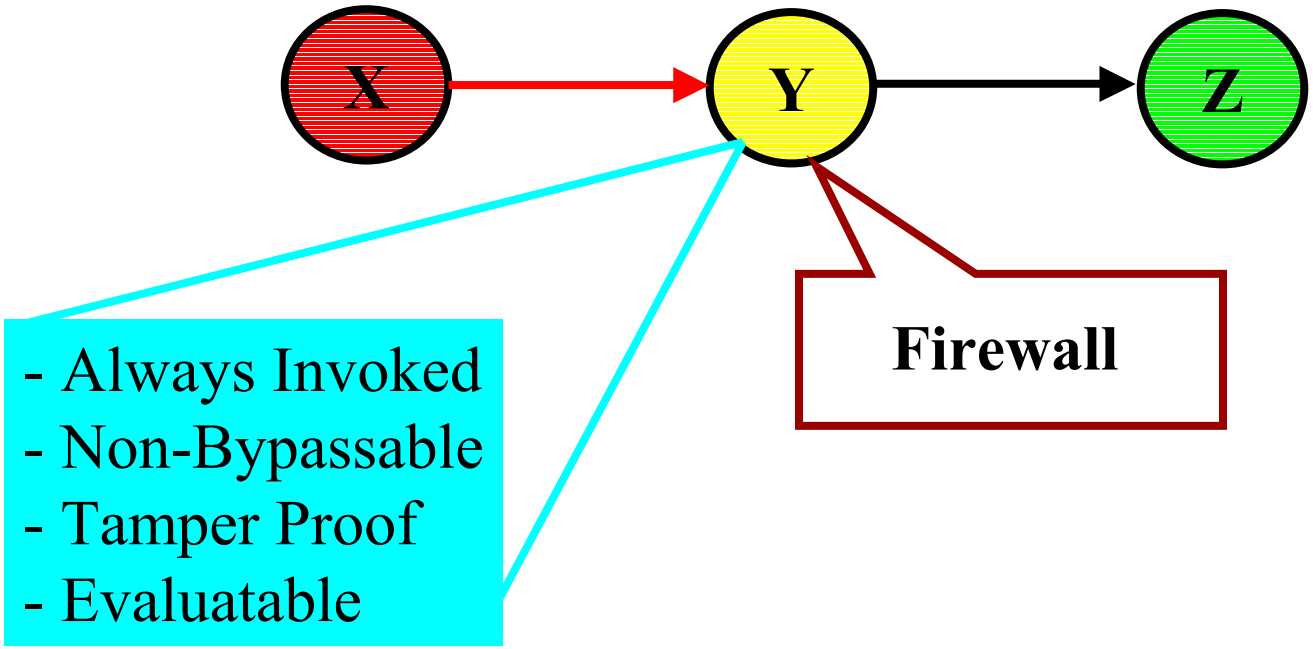
Outline

- Integrated Modular Avionics
- Intrinsic Partitioning
- Partitioning for Security
- Formal Verification
- Development Environment

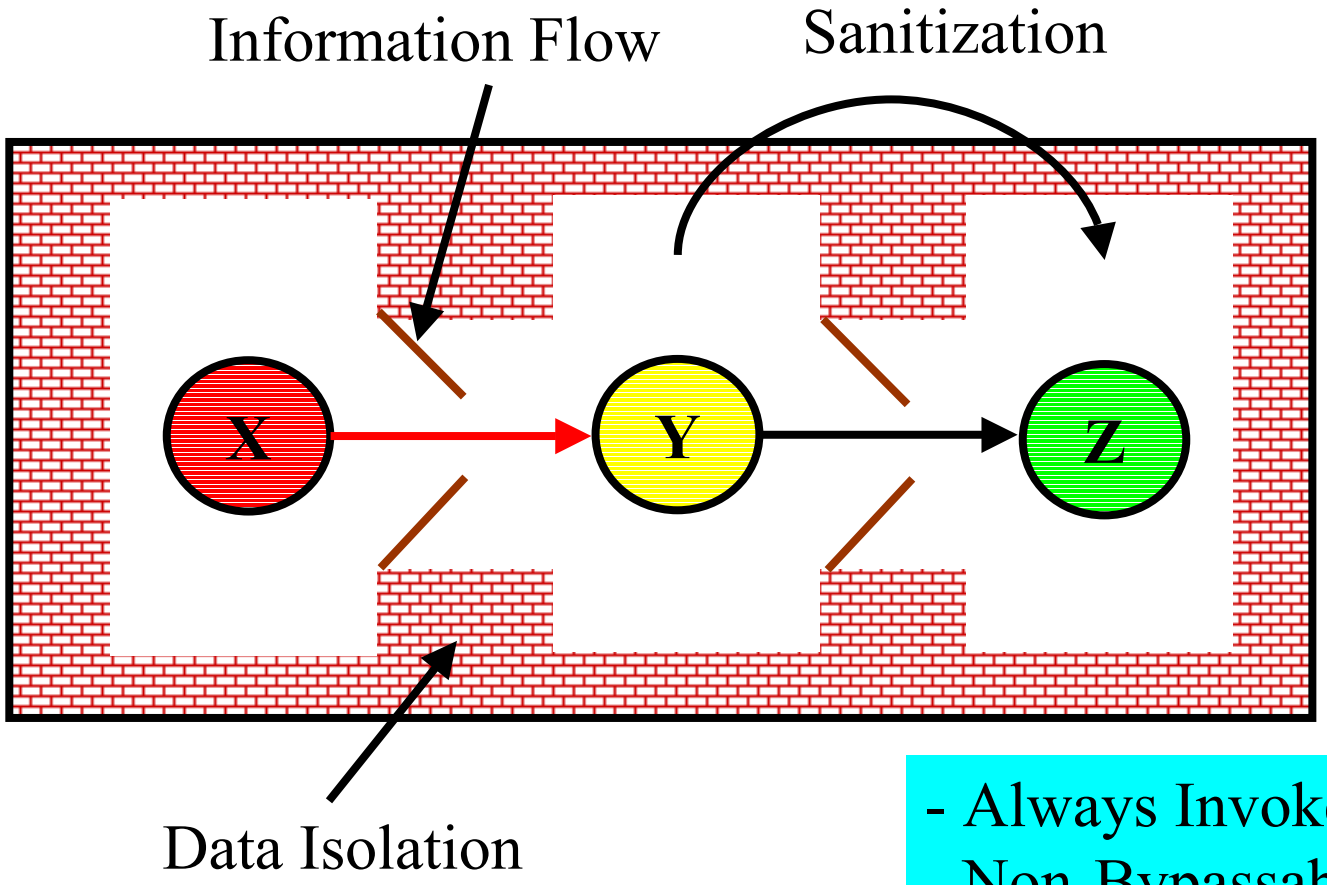
Separation Kernel

- **Concept First Published in 1980's**
  - Building Block for Secure Systems
  - Decomposes Challenge of Building Secure System
    - Allows Applications to Enforce and Manage Own Security Policy
  - Provides High Assurance Separation
- **Effective Security Policies Must Be**
  - Always Invoked
  - Non-Bypassable
  - Tamper Proof
  - Evaluatable
- **Separation Kernels Support Security Policies with**
  - Information Flow Control
  - Data Isolaton
  - Sanitization (Periods Processing)

Application Level Security Policy



Security Kernel Services



- Always Invoked
- Non-Bypassable
- Tamper Proof
- Evaluatable

Intrinsic Partitioning for Security

- **IMA very similar to MILS**
  - Originally Relied on Physical Separation, Now on Partitioning
  - Isolation of Concerns: Incremental Certification
- **Intrinsic Partitioning is a "Separation Kernel" designed into the processing platform**
  - Separation as a System Design Philosophy
- **Formal Analysis**
  - Mandated for Highest Security Certifications
  - Intrinsic Partitioning Designed with Formal Verification in Mind
    - Limited Functionality, Limited Problem Scope
  - Lowest Level Implementation
    - Independent of Software
    - Simplest Level to Implement/Verify Separation

Outline

- Integrated Modular Avionics
- Intrinsic Partitioning
- Partitioning for Security
- Formal Verification
- Development Environment

Formal Process

- **Formal Process**
  - Process Adheres to Conventional or Accepted Methods or Standards
  - Specific Steps are Taken, Specific Documentation is Produced
- **Rigorous Process**
  - Forces Attention to Easily Overlooked Details
- **Not "Formal Methods"**
  - Complementary Concepts

Formal Methods

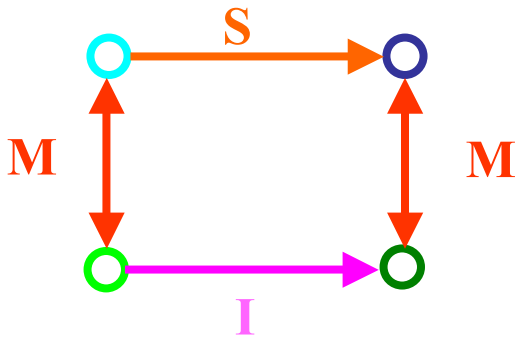
- **Formal Methods**
  - Discipline in which Mathematical Reasoning is Applied to the Development or Verification of Computer Systems
  - Formal Languages
    - Rigorously Defined Syntax and Semantics (Meaning)
  - Formal Tools
    - Computer Programs that Manipulate Formal Languages
    - Employ Logic and Rules of Inference
- **Rigorous Specification**
  - Forces Attention to Easily Overlooked Details
- **Part of Formal Process**
  - DO-178B
    - Alternative Means
  - Common Criteria
    - Required Part of Certification Process

$$X < X + 1$$

$$(P \ \& \ Q) \Rightarrow P$$

Formal Techniques

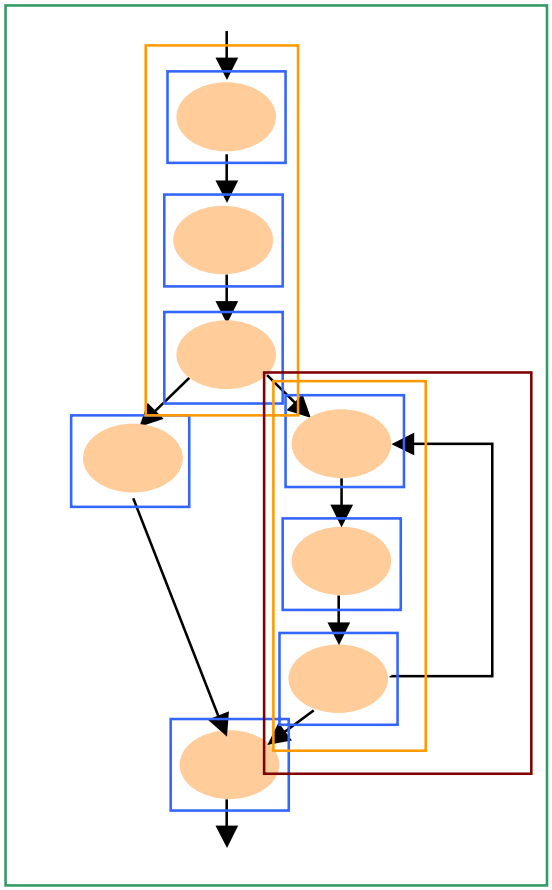
- **Formal Specification**
  - Rigorous Mathematical Description of System
  - Many Formal Languages/Tools
    - Manipulated by Computational Means
- **Formal Validation**
  - Consistency and Completeness
- **Formal Synthesis**
  - Derivation of Implementation from Specification
    - Kestrel, Derivation Reasoning Systems
- **Formal Verification**
  - Proof of Correspondence Between Implementation and Specification
  - Mechanical Proof Systems
    - Model Checkers, Equivalence Checkers
    - Theorem Provers (PVS, HOL, ACL2, etc.)

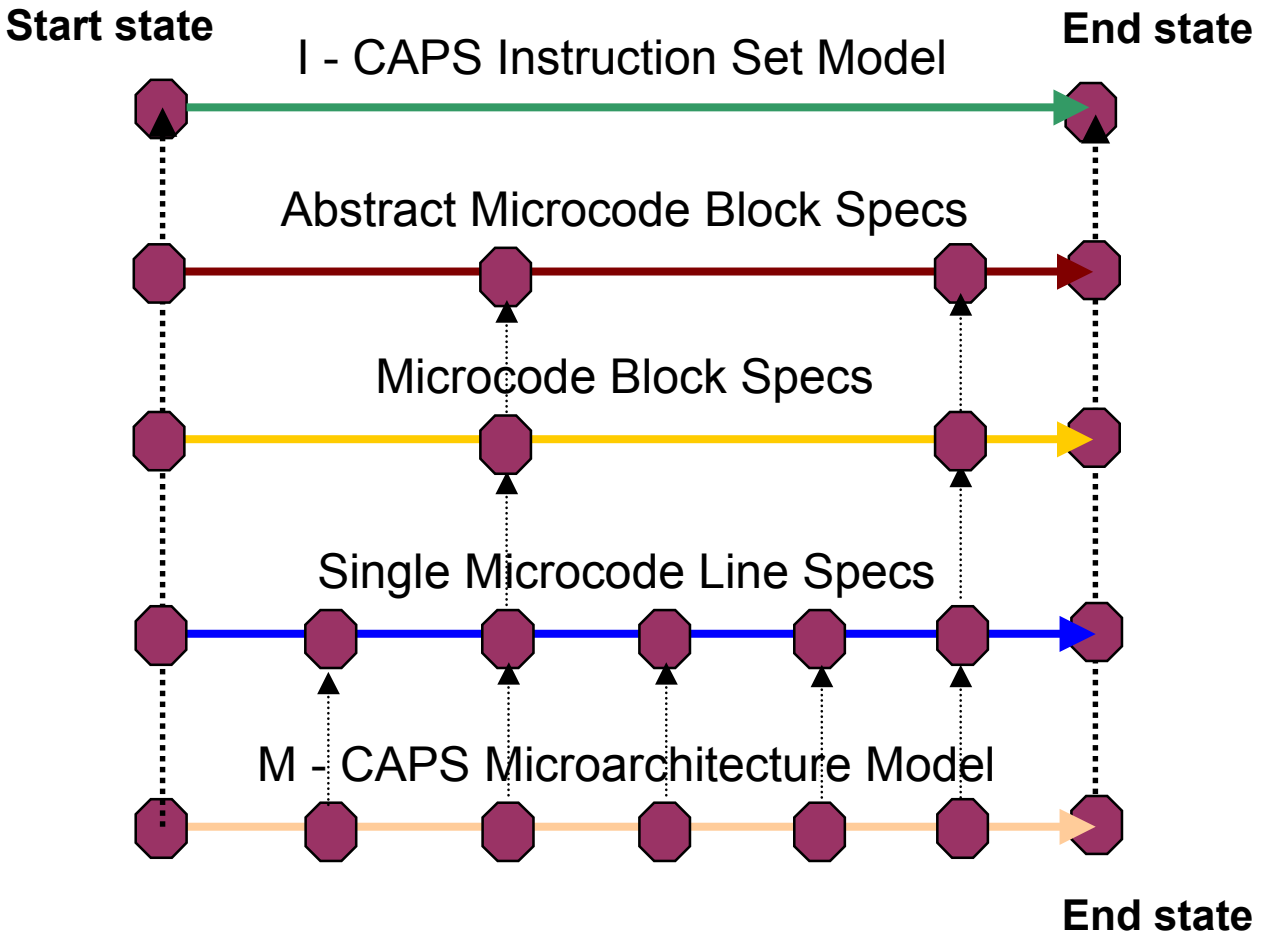


RC Formal Methods History

- **Rockwell Collins Formal Methods History**
  - **FY94: Microcode correctness for AAMP5 (*NASA Langley*)**
  - **FY96: Microcode correctness for AAMP-FV (*NASA Langley*)**
  - **FY97-99 Avionics Application Partitioning (*DARPA*)**
  - **FY98: High-Speed Executable Formal Model of the JEM1 (IR&D)**
  - **FY99: Autopilot Mode Confusion (NASA Langley)**
  - **FY99-01: CAPS Analysis (IR&D)**
  - **FY02-FY03: AAMP7 partitioning analysis (IR&D)**

CAPS Analysis
(microcode correctness proofs)

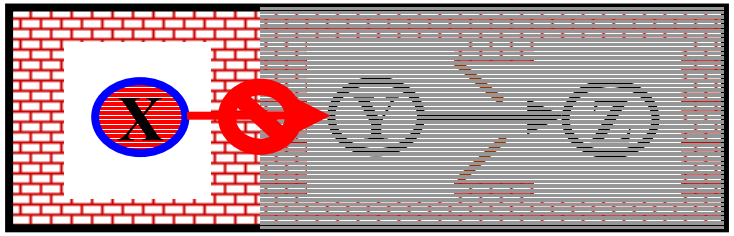


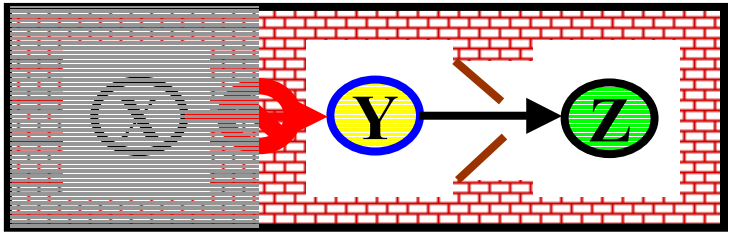


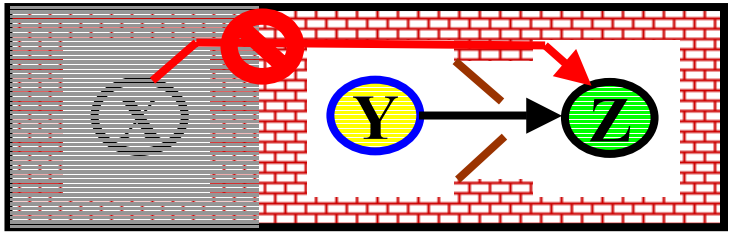
Rockwell Collins' microcode verification work presented Tuesday.

Formalized Separation Kernel Security Policy

- **Informal Security Policy**
  - Information Flow Control
  - Data Isolation
  - Sanitization
- **Need for Formalize**
  - Precise Mathematical Description
  - Suitable for Formal Analysis
- **Formal Security Policy**
  - Infiltration
  - Exfiltration
  - Mediation

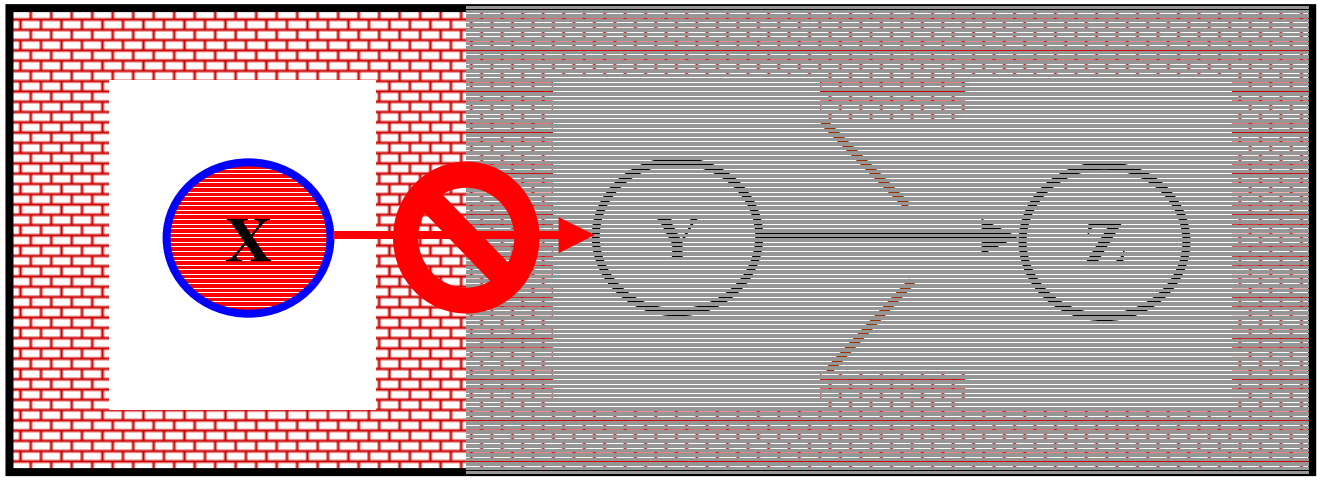






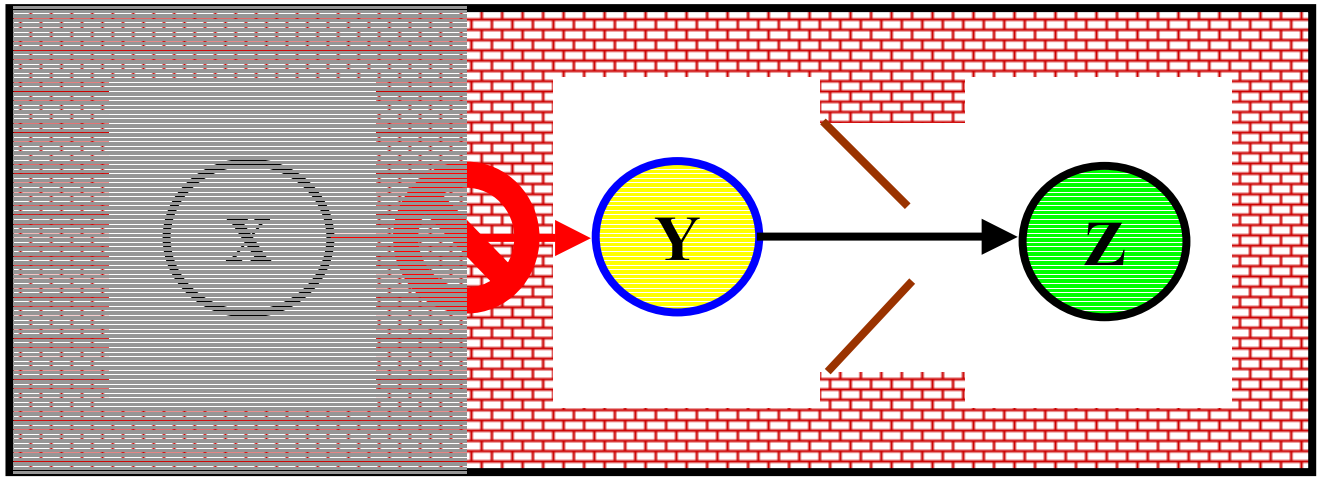
(No) Exfiltration

```
(defthm Exfiltration
  (implies
     (not (Direct-Interaction-Allowed (Current-Partition st) y))
     (equal (Accessible-Information y (Step-System st))
            (Accessible-Information y st))))
```



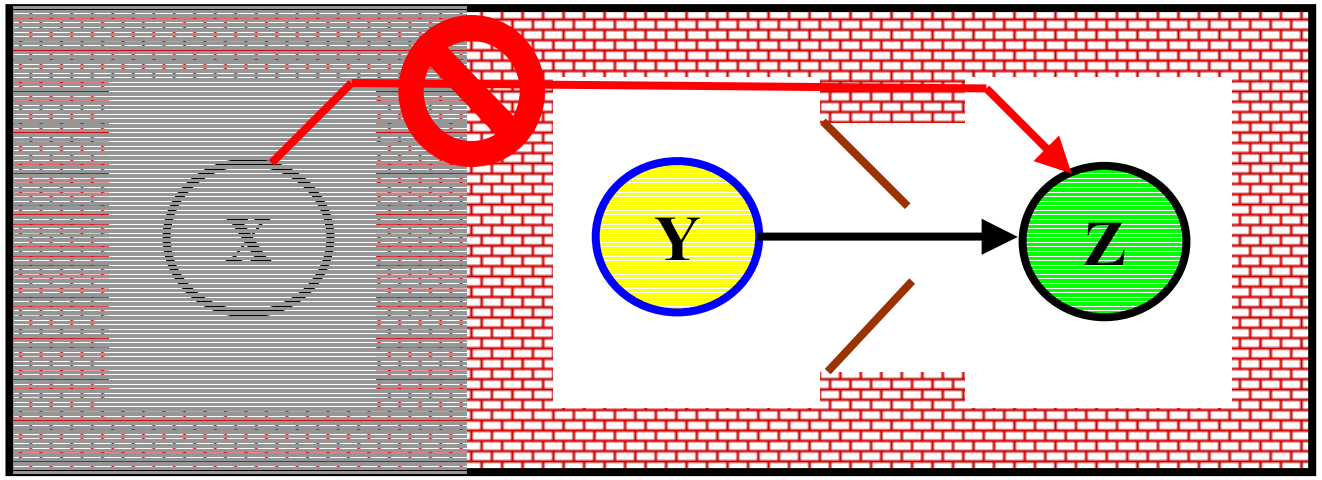
(No) Infiltration

```
(defthm Infiltration
  (implies
    (and (equal (Kernel-State st1) (Kernel-State st2))
         (equal y (Current-Partition st1))
         (equal (Accessible-Information y st1) (Accessible-Information y st2)))
    (equal (Accessible-Information y (Step-System st1))
           (Accessible-Information y (Step-System st2)))))
```



(No) Mediation

```
(defthm Mediation
  (implies
    (and (Direct-Interaction-Allowed (Current-Partition st1) z)
         (equal (Kernel-State st1) (Kernel-State st2))
         (equal (Accessible-Information (Current-Partition st1) st1)
                (Accessible-Information (Current-Partition st1) st2)
         (equal (Accessible-Information z st1) (Accessible-Information z st2)))
    (equal (Accessible-Information z (Step-System st1))
           (Accessible-Information z (Step-System st2)))))
```



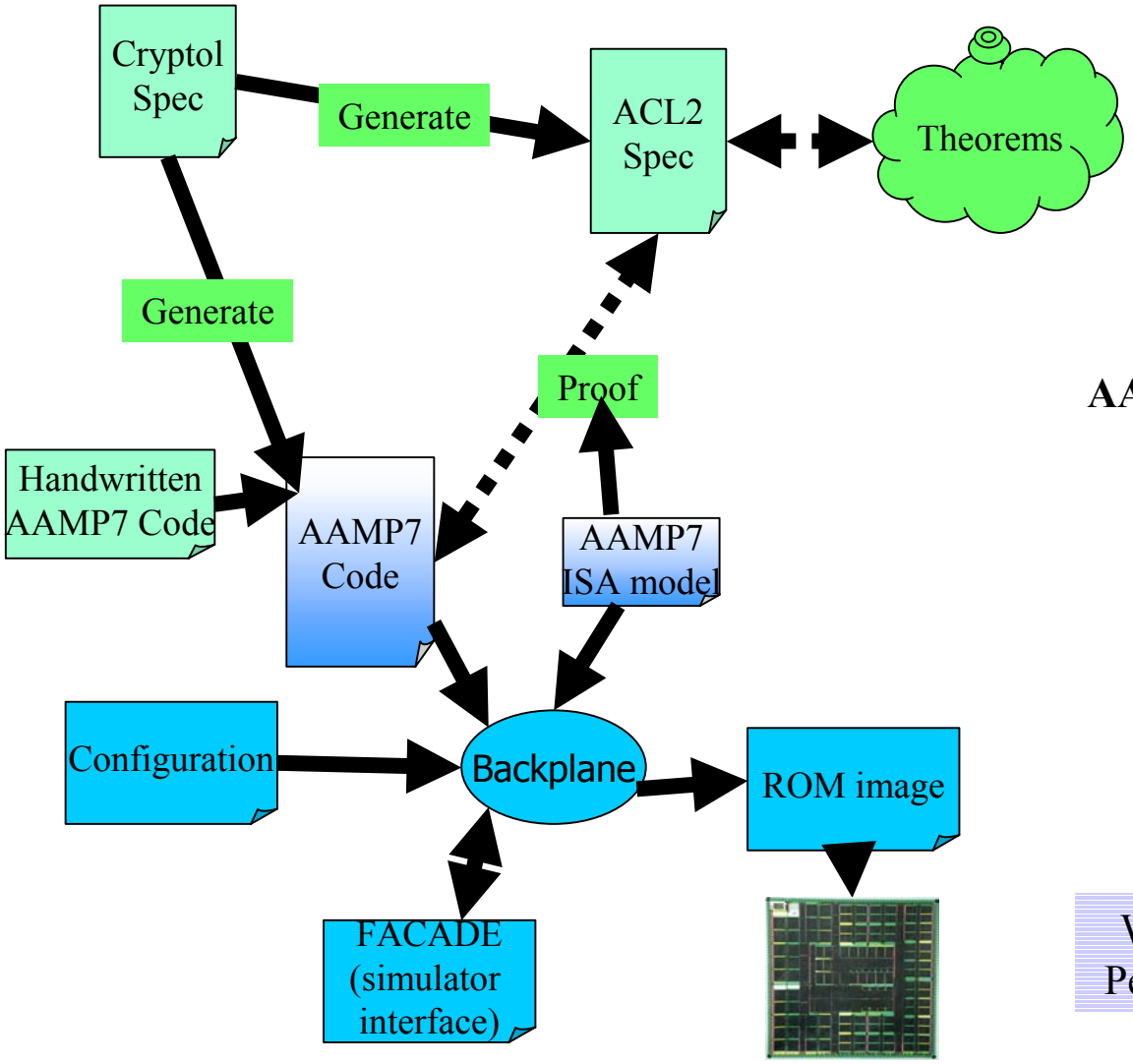
Effort

- **ACL2-checked Proofs**
  - **Currently connecting Implementation Model to Security Policy using the ACL2 theorem proving system**
  - **Prior Rockwell Collins FM Work Crucial**
    - **Schedule**
    - **Capability**

Outline

- Integrated Modular Avionics
- Intrinsic Partitioning
- Partitioning for Security
- Formal Verification
- Application Development Environment

Development Environment Project Overview



AAMP7 Development Environment

- **Cryptol**
- **Instruction-level code proofs**
- **Partitioning support**

Work with John Launchbury and Peter White of Galois Connections

Summary

- Integrated Modular Avionics
  - ***Safety-Critical Avionics Integration Concept***
- Intrinsic Partitioning
  - ***"Separation Kernel" in a MILS Computing Platform***
- Partitioning for Security
  - ***Application-Level Firewalls Supported***
- Formal Verification
  - ***Provides High Assurance Intrinsic Partitioning***
- AAMP7 development environment
  - ***Supports high-assurance application development exploiting intrinsic partitioning***