Assured Cloud Computing:

The Odessa Monitoring System

Roy Campbell

Assuring the Cloud -Summer Workshop

Griffiss Institute, July 11th 2011 Rome, NY

“Fly, fight, and win in air, space, and cyberspace”

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
11 JUL 2011

2. REPORT TYPE

3. DATES COVERED
00-00-2011 to 00-00-2011

4. TITLE AND SUBTITLE
Assured Cloud Computing: The Odessa Monitoring System

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Illinois at Urbana?Champaign ,Information Trust Institute,Urbana,IL,61801

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR’S ACRONYM(S)

11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**74**	

United States Air Force and Cloud Computing

- United States Air Force Vision
  - Global vigilance, reach and power
- Net centric military superiority
  - Rapid technological advance
  - Computer-based weapons systems
- Problems
  - Overseas commitments and operations
  - Global networking requirements
  - Government and commercial off-the-shelf technology
  - Secure computing over blue and gray networks
  - Agility and mobility

Background

- Federal Cloud Computing Strategy. Vivek Kundra, US Chief Information Officer, Feb 8th 2011, The White House:

The cloud computing model can significantly help agencies grappling with the need to provide highly reliable, innovative services quickly despite resource constraints

http://www.cio.gov/documents/Federal-Cloud-Computing-Strategy.pdf

- Appendix 1: Potential Federal Spending on Cloud. DOD 2 billion plus.
- Appendix 2: Agency Resources for Cloud Computing

Recent Problems in the Cloud

April 21st 2011 Amazon Elastic Block Store (EBS) went offline, leaving the many Web and database servers depending on that storage broken. Not until Easter Sunday (April 24) was service restored to all users.

June 19th 2011 Dropbox, one of the most popular ways to share and sync files online, says the accounts became unlocked at 1:54pm Pacific time Sunday when a programming change introduced a bug. The company closed the hole a little less than 4 hours later.

June 22nd 2011 Microsoft's BPOS (Business Productivity Online Suite) cloud-hosted communication and collaboration suite suffered an outage on Wednesday for more than three hours and involved a networking hardware problem that affected customers in North America.

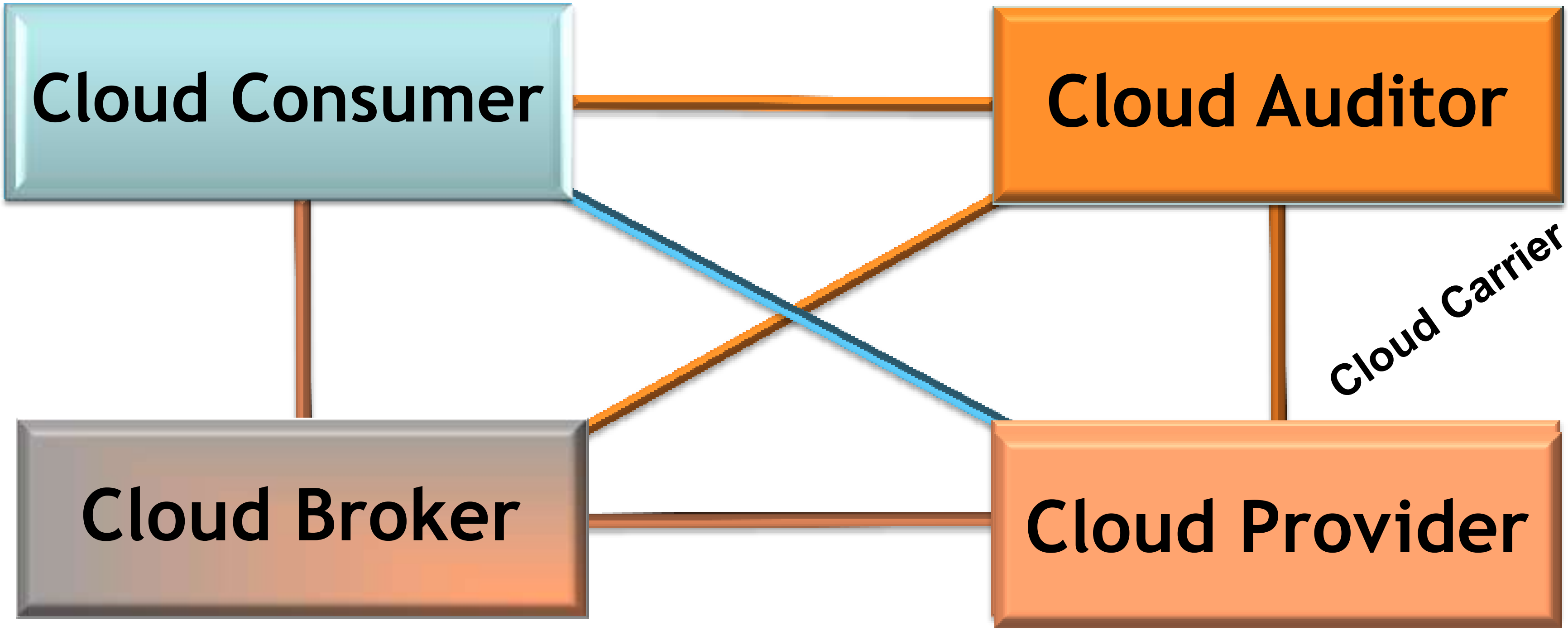
NIST Cloud Computing Standards Roadmap

July 5, 2011: http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/StandardsRoadmap/NIST_CCSRWG_092_NIST_SP_500-291_Jul5.pdf

The NIST Definition of Cloud Computing identified cloud computing as:

a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Interactions between Actors in Cloud Computing



Cloud Consumer

Cloud Auditor

Cloud Carrier

Cloud Broker

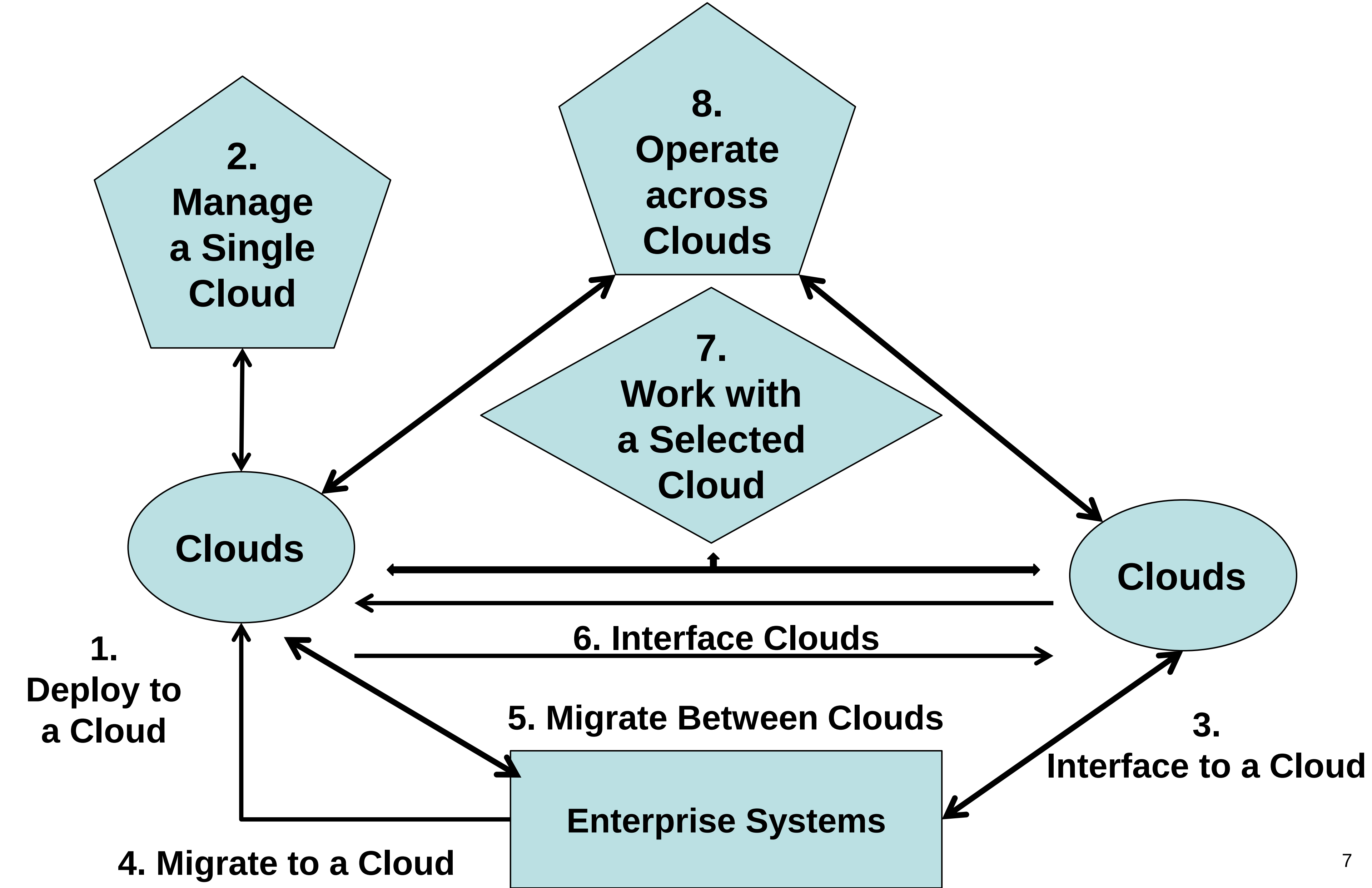
Cloud Provider

🟦 The communication path between a cloud provider & a cloud consumer

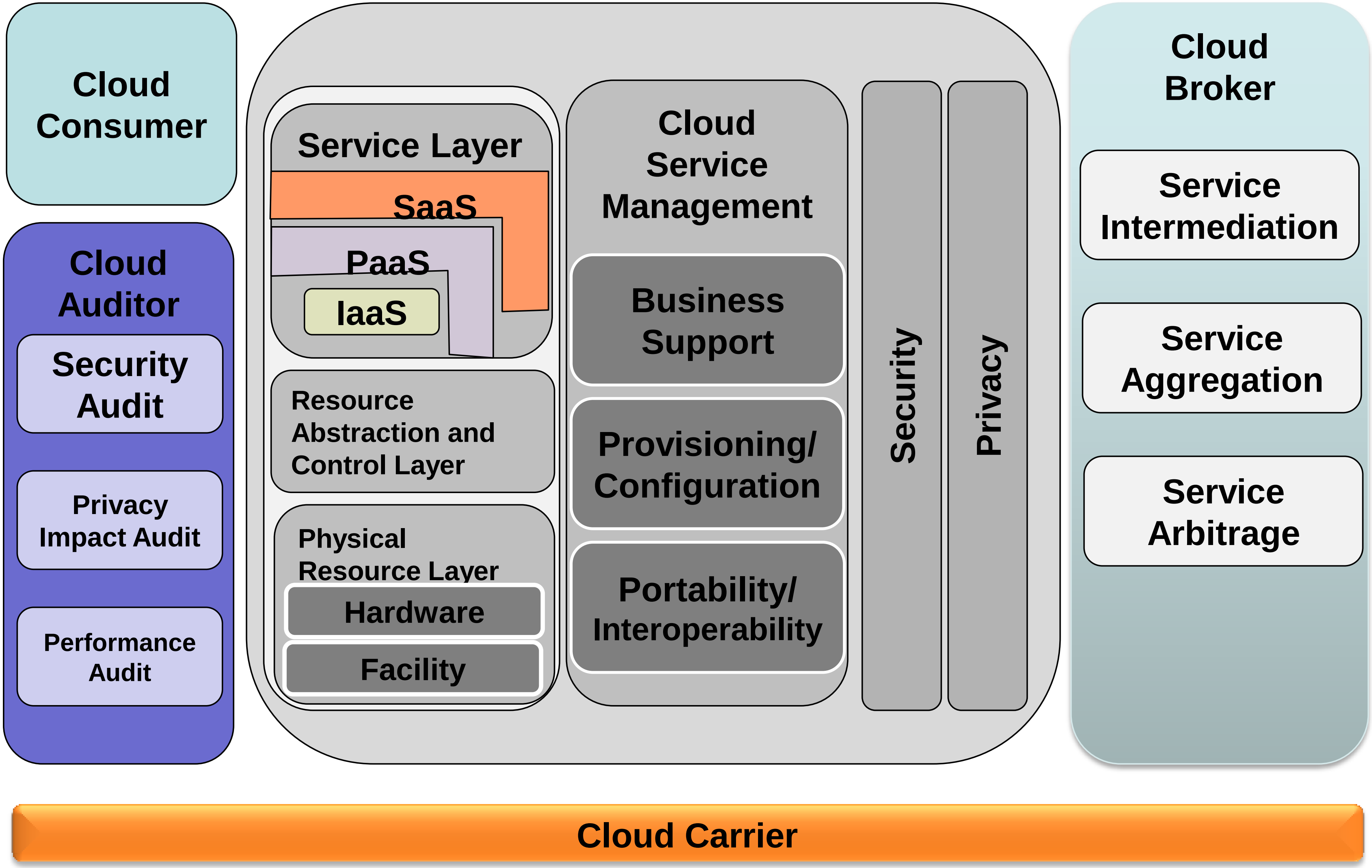
🟧 The communication paths for a cloud auditor to collect auditing information

🟧 The communication paths for a cloud broker to provide service to a cloud consumer

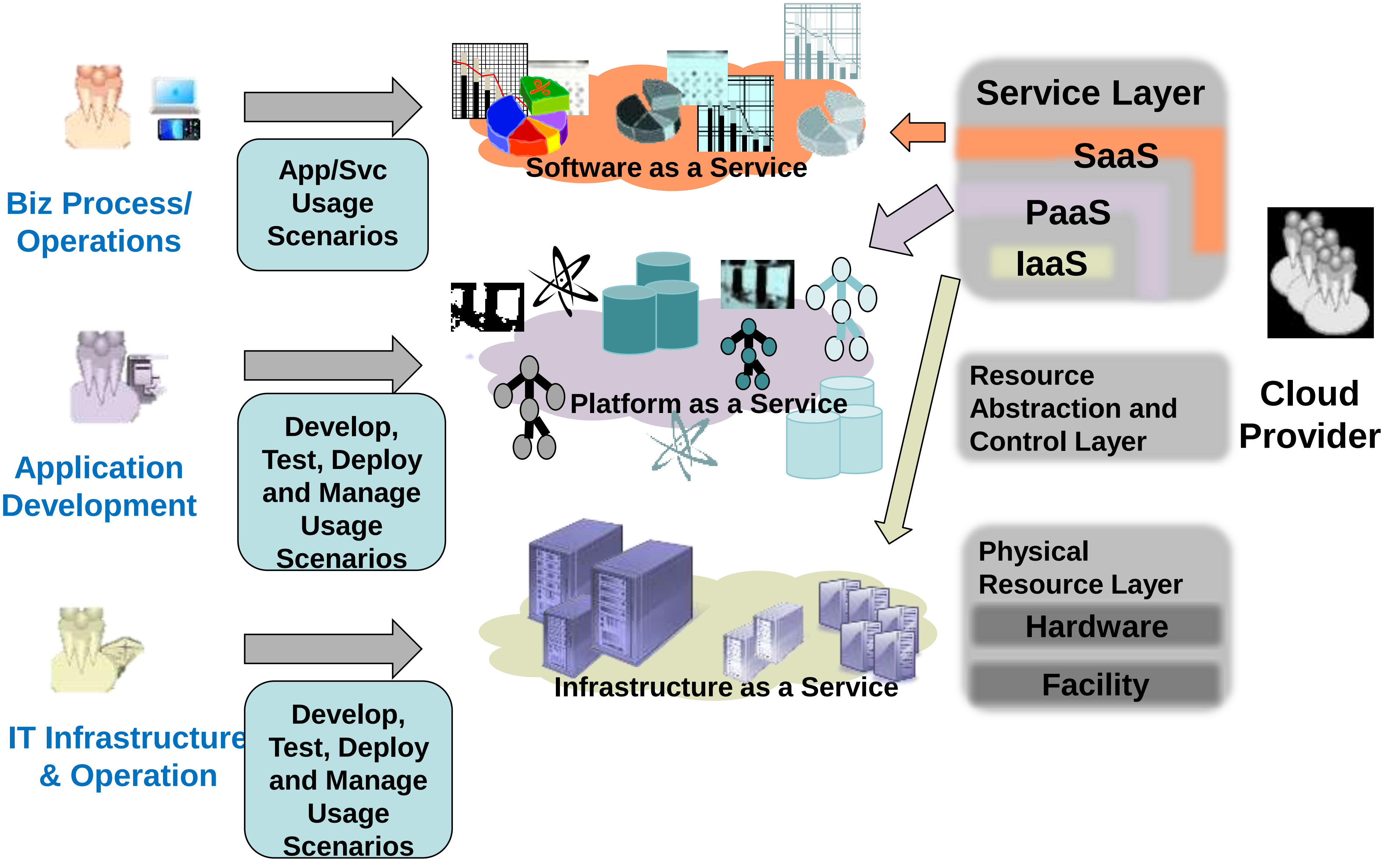
Deployment Generic Scenario Perspective



The Combined Conceptual Reference Diagram



Cloud Provider: Service Orchestration



Biz Process/ Operations

App/Svc Usage Scenarios

Software as a Service

Service Layer

SaaS

PaaS

IaaS

Application Development

Develop, Test, Deploy and Manage Usage Scenarios

Platform as a Service

Resource Abstraction and Control Layer

Cloud Provider

IT Infrastructure & Operation

Develop, Test, Deploy and Manage Usage Scenarios

Infrastructure as a Service

Physical Resource Layer

Hardware

Facility

Cloud Security Standards

- A very new topic
- Multiple bodies are trying to standardize
  - Cloud Security Alliance
    - Security Guidance for Critical Areas of Focus in Cloud Computing
    - Top Threats to Cloud Computing
    - Cloud Audit (A6→Automated Audit,Assertion,Assessment,and Assurance API)
  - NIST Cloud Security Initiative
    - Guidelines on Security and Privacy in Public Cloud Computing
  - Military → IASE standards from DISA-CSD
  - Federal Government
    - FedRAMP(2011)
    - Evolved from NIST 800-053, from 2009
    - Assessment procedures
  - OASIS Identity in the cloud
    - Open standards for identity deployment, provisioning and management

Assured Cloud Computing Center: Requirements and Challenges

- Mission Oriented
- Interoperability (across blue and gray networks)
- A plethora of evolving standards
- End-to-end, Cross-layered
  - Security
  - Dependability
  - Timeliness

Architecture + Design + Testing + Formal Verification

- Use of formal methods to:
  - Analyze, reason, prototype and evaluate architectures
  - Design and optimize the performance of secure, timely, fault-tolerant, mission-oriented cloud computing.
- Evaluation of a wide range of necessary Assured Cloud Computing components
- Along with engaging AFRL in technological exchange, we plan to integrate AFRL personnel into our research agenda, as well as provide focused education delivery

A Survivable and Distributed Cloud-Computing-based Infrastructure

- Configuration and management of:
  - Dynamic systems-of-systems
  - Trusted and partially trusted resources and,
  - Services sourced from multiple organizations

- Assured mission-critical computations and workflows with configurations that do not violate any security or reliability requirements

- Models of the trustworthiness of a workflow or computation's completion for a given configuration in order to specify the right configuration for high assurances

Research Agenda

1) Flexible and dynamic distributed cloud-computing-based architectures that are survivable
2) Novel security primitives, protocols, and mechanisms to secure and support assured computations
3) Algorithms and techniques to enhance end-to-end timeliness of computations
4) Algorithms that detect security policy or reliability requirement violations in a given configuration
5) Algorithms that dynamically configure resources for a given workflow based on security policy and reliability requirements and
6) Algorithms, models, and tools to estimate the probability of completion of a workflow for a given configuration

Deliverables

a) Groundbreaking research in new algorithms and techniques

b) Development and experimental evaluation of prototypes

c) Education and technical exchange

ITI Capabilities

- Since 2004 ITI has supported a multidisciplinary “Research Network” of 100+ research faculty to complete a cumulative of almost $60M in sponsored research into trustworthy systems

- College of Engineering, of which ITI is a part, is ranked sixth in the nation

- Both Departments of Electrical and Computer Engineering and Computer Science ranked in the top five of the nation

Approach to Assurance

- Assurance is the key factor:
  - Model the trustworthiness of a workflow
  - Model configuration of dynamic systems-of-systems
  - Check Configurations do not violate security or reliability requirements

- Requires algorithms, models and tools that:
  1. Model a cloud configuration
  2. Detect security or reliability violations
  3. Dynamically configure resources for a given workflow
  4. Estimate the probability of completion of a workflow for a given configuration

ACC-UCoE@UIUC

- Undertake core research and development to address these challenges for new and modified architectures, algorithms, and techniques:
  - Design, formally analyze, run-time configuration, experimental evaluation

- Will deliver:
  - Research: new algorithms and techniques
  - Engineering: development and experimental evaluation of prototypes
  - A focused workforce development that includes education, and technology exchange

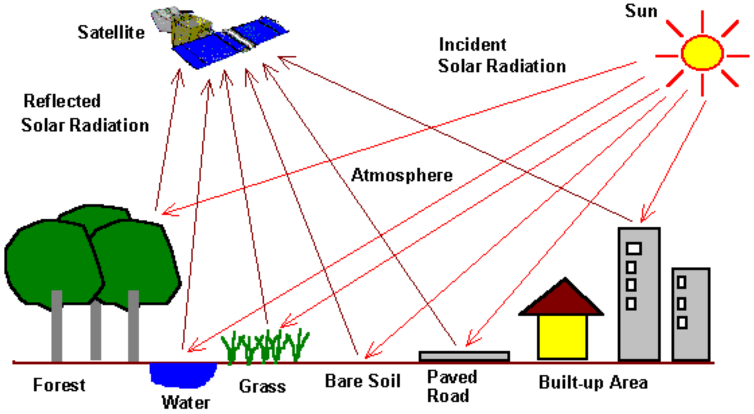
Air Force Mission: Disaster Relief in Hostile Territory

Locate and identify stranded civilians and damaged infrastructures

- Available resources
- Blue/Gray networks
- GIS & Cloud Computing
- Communication
- Imaging
- Search
- Confidentiality
- Authorization

Use relay stations to remote monitor remote sensors

- Wireless Networks
- Real-time
- Sensor fusion
- Security & Authentication
- Reliability and Availability
- Search & Analysis
- Satellite Coverage



- Enable secure and reliable services/computations with available resources from multiple organizations in real-time

Risk Analysis

- Research may not yield desired results and we could discover technological limitations
- Granularity of localization that can be achieved for a given amount of computing /communication overhead
- Leverage mature technologies and proven tools and technologies
- Architectural strategies (e.g. leveraging multiple paths, complement integrity protection)

Organizational structure

Rakesh Bobba

Indy Gupta

Gul Agha

Principal Investigator
Roy Campbell

Advisory Committee

- Design
  Indranil Gupta
  - Distributed Architectures
  - Security Protocols
  - Real-time Assuredness
- Formal Methods
  Gul Agha
  - Safety Properties
  - Real-time Properties
  - Performance Properties
- Run-time
  Campbell
  - Policy Detection
  - Dynamic Mapping
  - Trustworthiness Estimation
  - Security State Monitoring
- Test-bed
  Kalbarczyk
  - Incident Replay Engine (IRE)
  - Test-bed
- Education
  Bashir
  - Technical Exchange & Speakers
  - Workshops
  - Visiting Scholars (summers)
  - STEM Internship Programs

Roy Campbell

Zbigniew Kalbarczyk

Masood Bashir

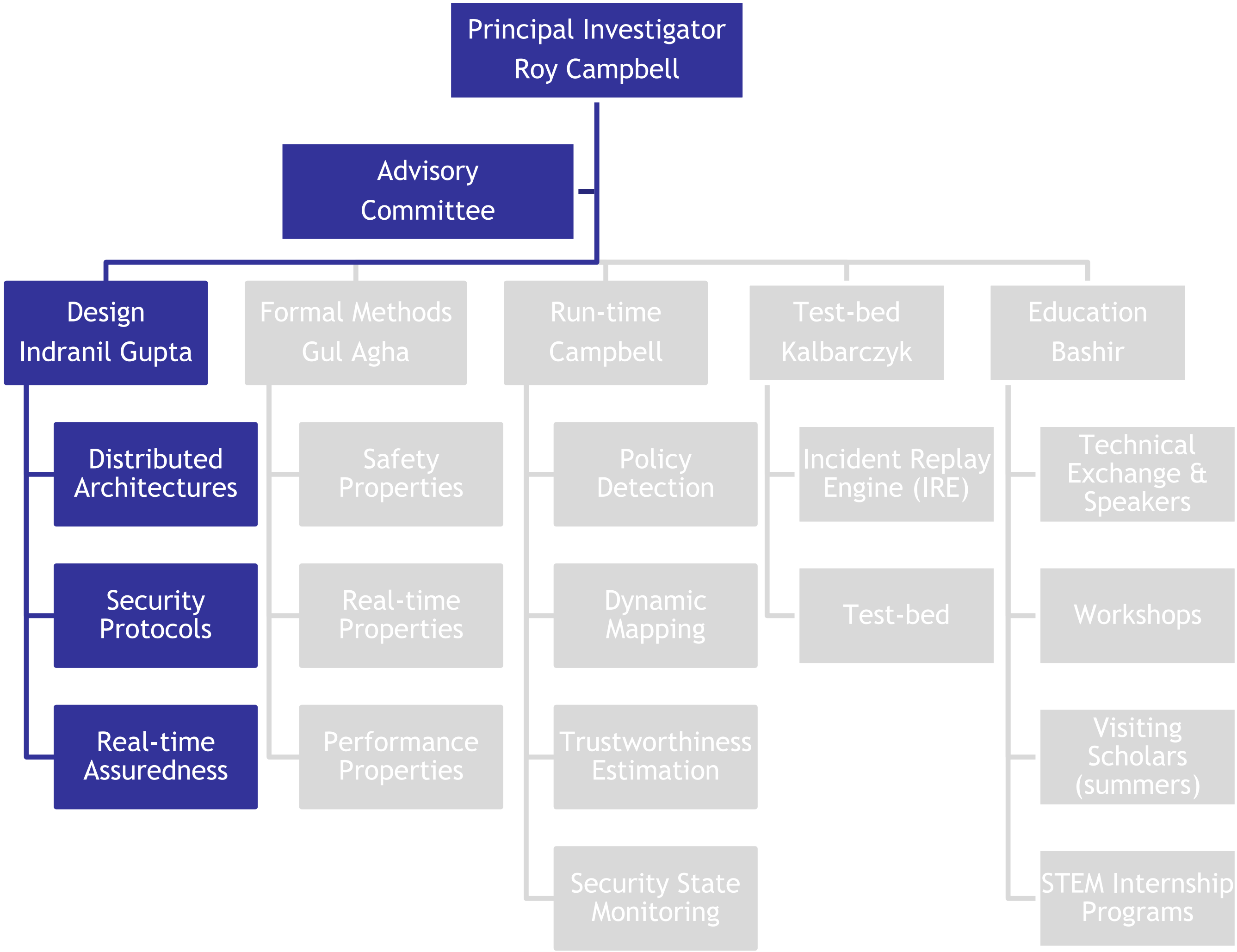
Jose Meseguer

Ravi Iyer

David Nicol

Bill Sanders

Design


Principal Investigator
Roy Campbell
Advisory Committee
Design
Indranil Gupta
Formal Methods
Gul Agha
Run-time
Campbell
Test-bed
Kalbarczyk
Education
Bashir
Distributed Architectures
Security Protocols
Real-time Assuredness
Safety Properties
Real-time Properties
Performance Properties
Policy Detection
Dynamic Mapping
Trustworthiness Estimation
Security State Monitoring
Incident Replay Engine (IRE)
Test-bed
Technical Exchange & Speakers
Workshops
Visiting Scholars (summers)
STEM Internship Programs

Design Challenges for Assured Mission-Critical Computations in Cloud-Based Infrastructure

- Design of Algorithms and Techniques for Real-time Assuredness in Cloud Computing

  Indranil Gupta (and student Brian Cho)

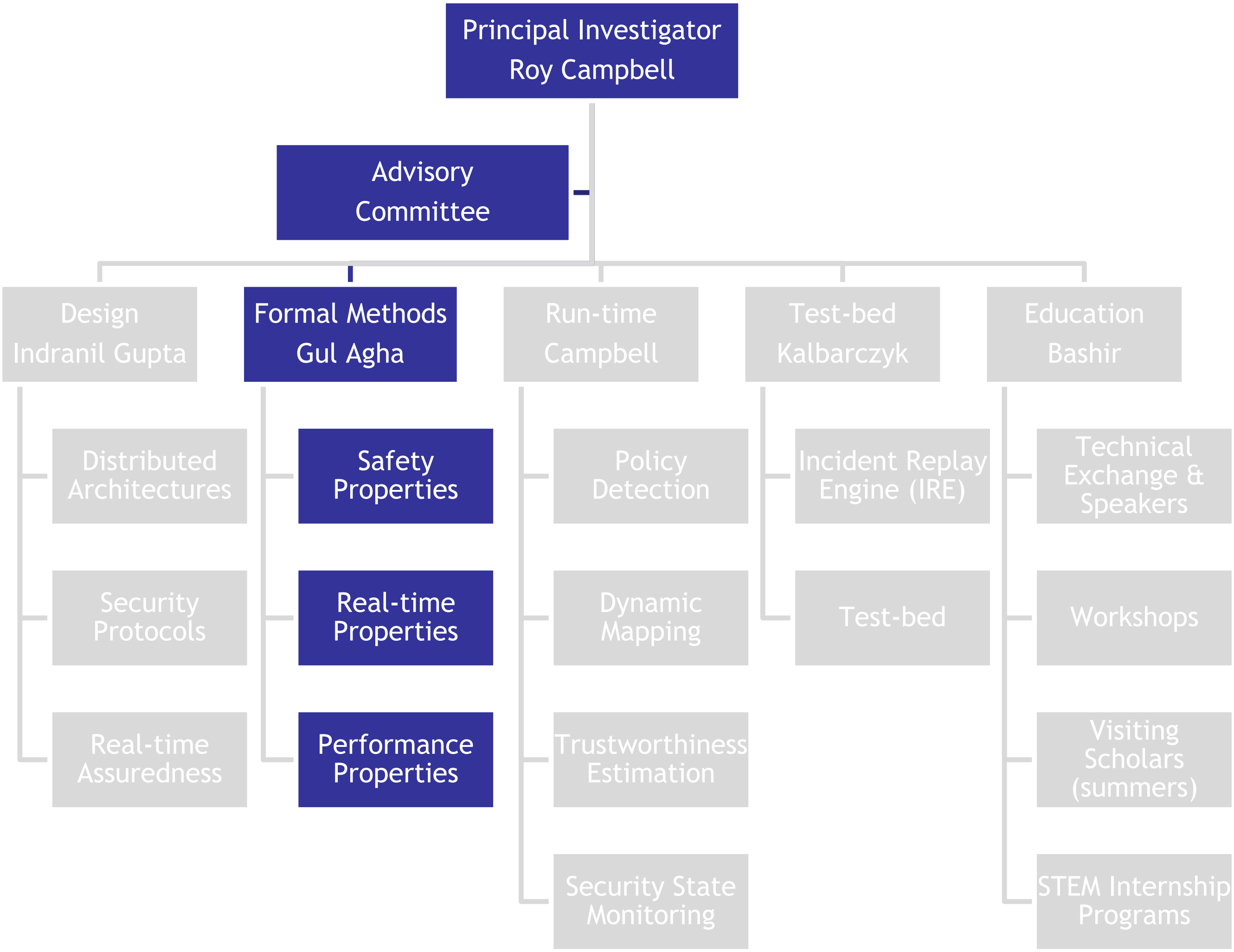
- Design of Novel Security Primitives, Protocols, and Mechanisms

  Rakesh Bobba

- Formal Design of Distributed Cloud-Computing-Based Architecture

  Gul Agha + Jose Meseguer

Formal Methods



Principal Investigator
Roy Campbell

Advisory
Committee

Design
Indranil Gupta

- Distributed Architectures
- Security Protocols
- Real-time Assuredness

Formal Methods
Gul Agha

- Safety Properties
- Real-time Properties
- Performance Properties

Run-time
Campbell

- Policy Detection
- Dynamic Mapping
- Trustworthiness Estimation
- Security State Monitoring

Test-bed
Kalbarczyk

- Incident Replay Engine (IRE)
- Test-bed

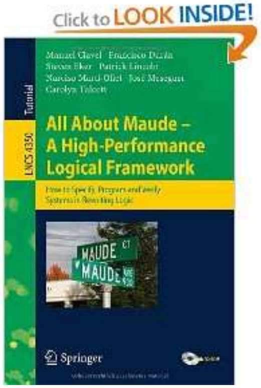
Education
Bashir

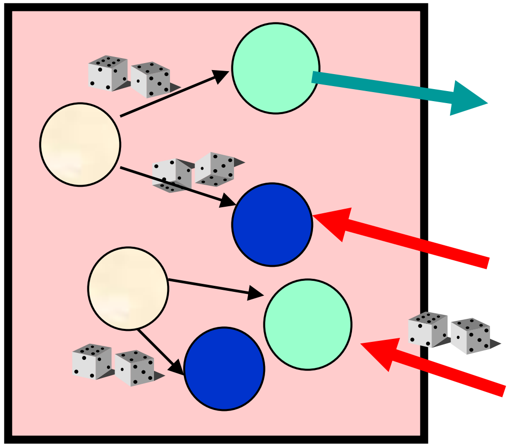
- Technical Exchange & Speakers
- Workshops
- Visiting Scholars (summers)
- STEM Internship Programs

Formal Methods Team

- *Rewriting Rules* as Executable specifications
- Maude provides methods for proving properties of programs
  - ➢ Safety (security), liveness
- *Examples:*
  - *actors using term rewriting*
  - *Two level actor semantics for middleware*
  - *pMaude: Probabilities on tactics of rule application*
    - ➢ Statistical metrics.
    - ➢ Quantify robustness stability, timeliness

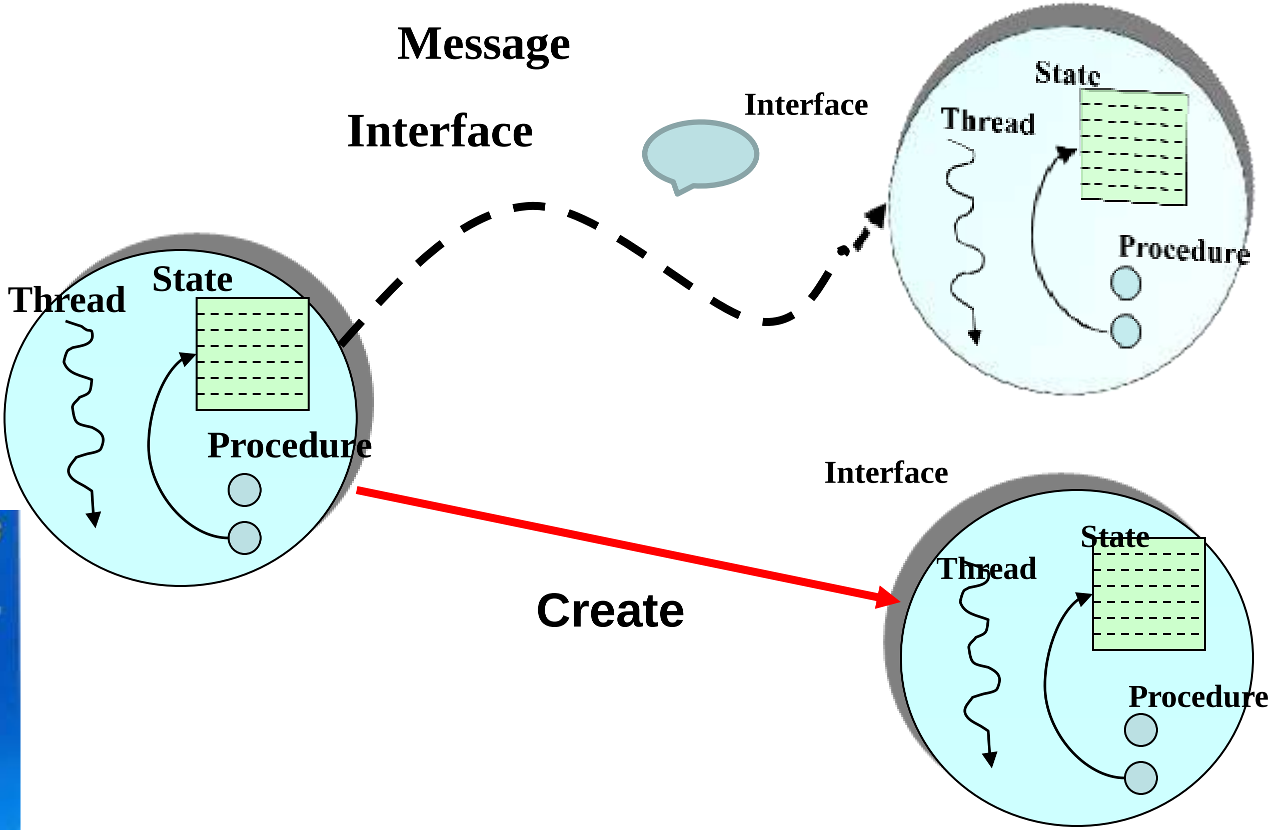
Jose Meseguer

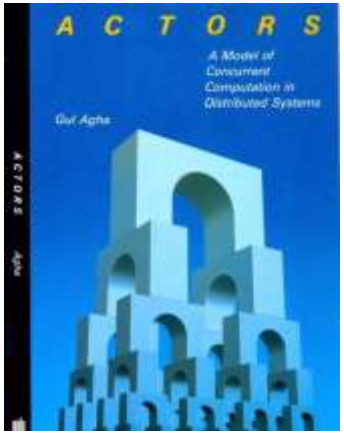

Click to LOOK INSIDE!
Manuel Clavel · Francisco Durán · Steven Eker · Patrick Lincoln · Narciso Martí-Oliet · José Meseguer · Carolyn Talcott
Tutorial
LNCS 4350
All About Maude – A High-Performance Logical Framework
How to Specify, Program and Verify Systems in Rewriting Logic
MAUDE
Springer



Formal Methods Team

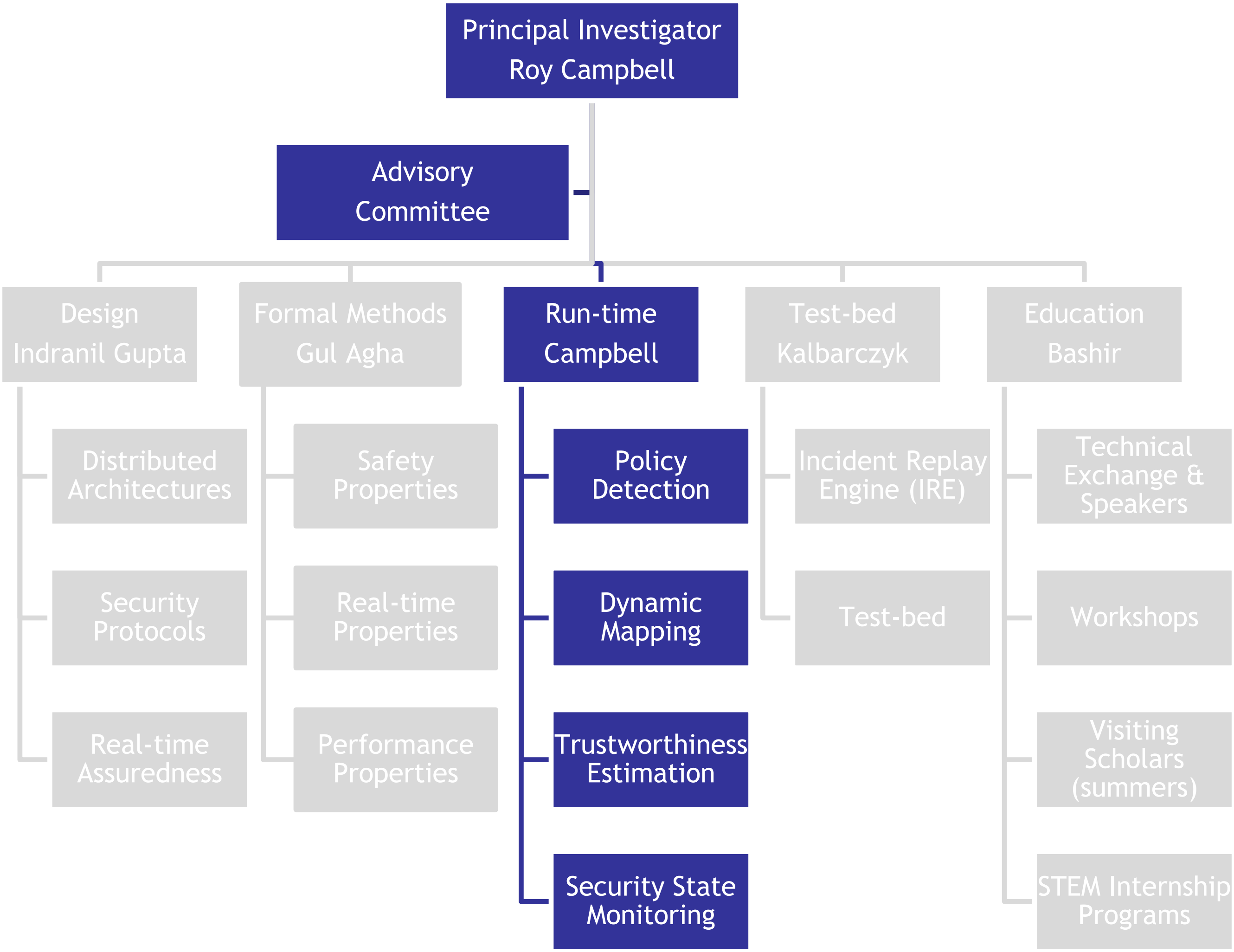
Gul Agha





- New actors have their own address
- Addresses may be communicated in messages

Run-time



Principal Investigator
Roy Campbell

Advisory
Committee

Design
Indranil Gupta

Distributed Architectures

Security Protocols

Real-time Assuredness

Formal Methods
Gul Agha

Safety Properties

Real-time Properties

Performance Properties

Run-time
Campbell

Policy Detection

Dynamic Mapping

Trustworthiness Estimation

Security State Monitoring

Test-bed
Kalbarczyk

Incident Replay Engine (IRE)

Test-bed

Education
Bashir

Technical Exchange & Speakers

Workshops

Visiting Scholars (summers)

STEM Internship Programs

Run-Time Configuration, Workflow Scheduling, and Security Monitoring Consideration

Security in cloud requires situational awareness and dynamic response to events

Fast and Scalable Detection of Policy Violations in Dynamic Assured Cloud Computing

Policy-based Dynamic Mapping of Services and Workflows

Trustworthiness Estimation for Workflow Completion

Security State Monitoring and Attack Response

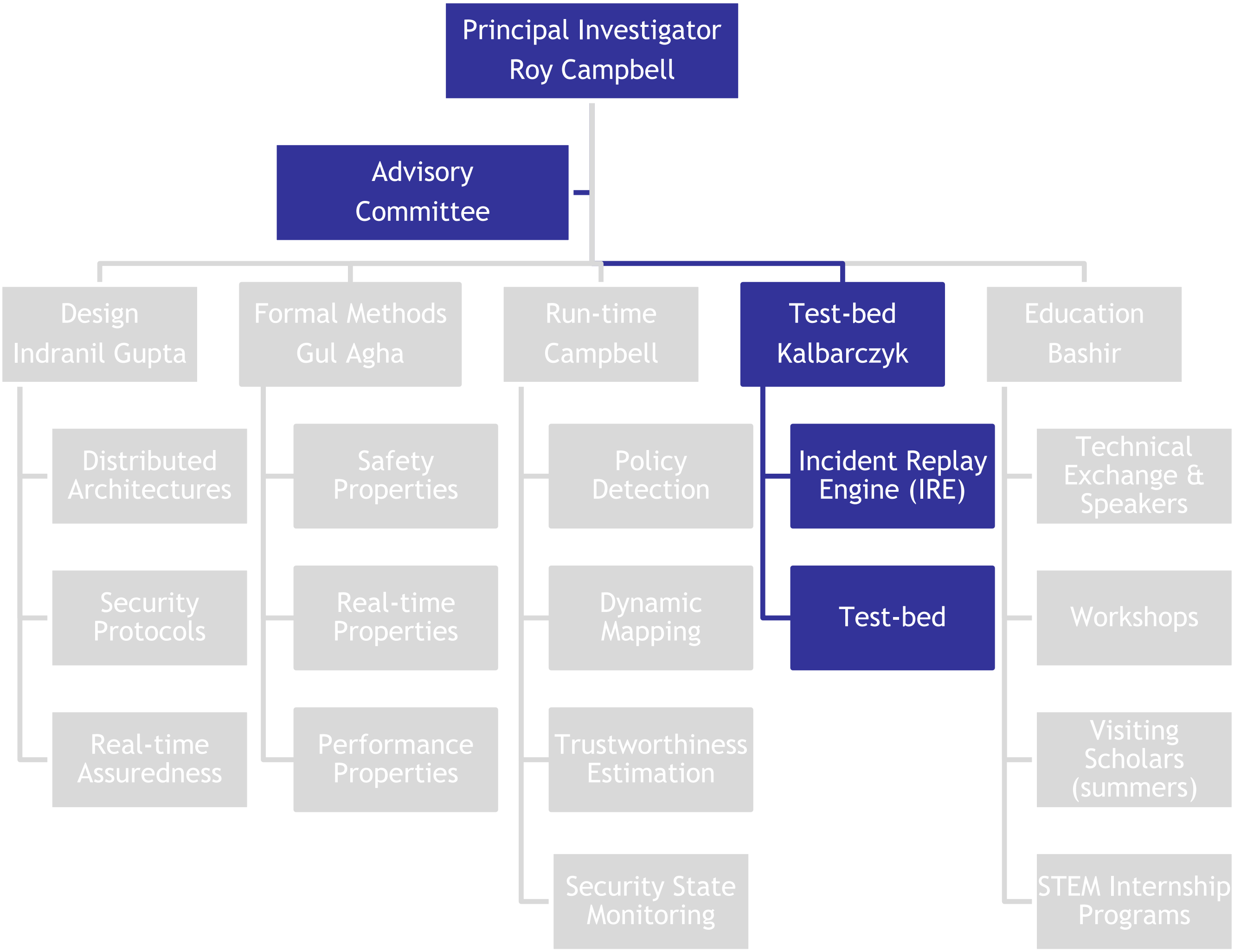
Roy Campbell

David Nicol

Bill Sanders

Rakesh Bobba

Test-bed



Principal Investigator
Roy Campbell

Advisory
Committee

Design
Indranil Gupta

Distributed Architectures

Security Protocols

Real-time Assuredness

Formal Methods
Gul Agha

Safety Properties

Real-time Properties

Performance Properties

Run-time
Campbell

Policy Detection

Dynamic Mapping

Trustworthiness Estimation

Security State Monitoring

Test-bed
Kalbarczyk

Incident Replay Engine (IRE)

Test-bed

Education
Bashir

Technical Exchange & Speakers

Workshops

Visiting Scholars (summers)

STEM Internship Programs

Outline

- Objectives for creating the test-bed
- Capabilities of the test-bed for experimental evaluation
- Validation tools
  - Example: characterization of error resiliency of virtualization environment in Cloud Infrastructure
- Reliability/security protection techniques
  - Example: application checkpointing through OS/Hypervisor-level techniques
- Analysis of security incidents
  - Example: incidents at NCSA
- Current facilities

Goals and People

- Create a distributed networked test-bed to:
  - provide an open platform to prototype and test new system configurations and applications
  - experimentally verify the effectiveness of algorithms and techniques for security and reliability monitoring
  - demonstrate the effectiveness of the developed architectures, algorithms, and protocols in presence of accidental failures and malicious attacks
- Complement formal analysis and verification of safety, real-time-, and performance-related properties of developed architectures, protocols, and algorithms

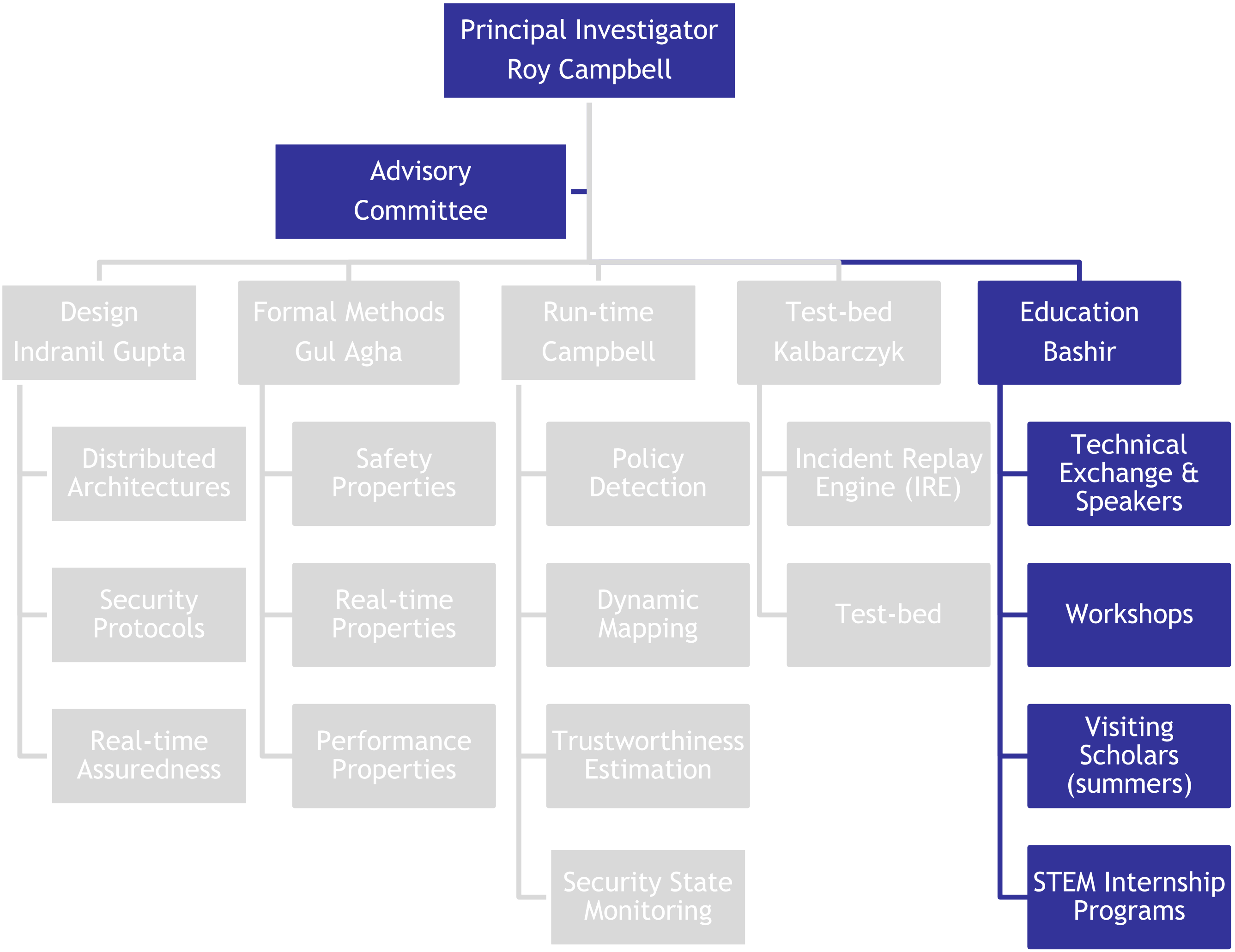
Zbigniew Kalbarczyk

Ravishankar Iyer

Example Capabilities of the Test-bed

- Validation tools
  - Validation of Virtualization Environment in Cloud Infrastructure using fault/error injection
- Rapid prototyping of designs
  - Application check pointing through OS/Hypervisor-level techniques, e.g., Xen, KVM
- Data-driven modeling of security incidents
  - Use knowledge on attack patterns learnt from the analysis of real security incidents to create security test-bed

Education



Principal Investigator
Roy Campbell

Advisory
Committee

Design
Indranil Gupta

Distributed Architectures

Security Protocols

Real-time Assuredness

Formal Methods
Gul Agha

Safety Properties

Real-time Properties

Performance Properties

Run-time
Campbell

Policy Detection

Dynamic Mapping

Trustworthiness Estimation

Security State Monitoring

Test-bed
Kalbarczyk

Incident Replay Engine (IRE)

Test-bed

Education
Bashir

Technical Exchange & Speakers

Workshops

Visiting Scholars (summers)

STEM Internship Programs

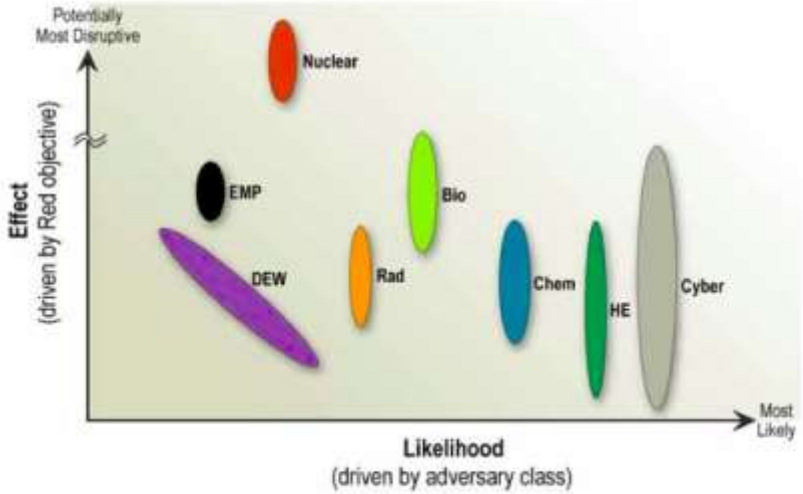
ITI Educational Initiatives

PROGRAMS

- NSA Center for Information Assurance Education and Research
- National Center of Academic Excellence in Information Assurance Education (CAEIAE)
- Graduate Degrees (MS, PhD)
- NSF-SFS scholarship
- Information Trust and Security Summer Internship
- Trust Curricular Roadmaps
- Trust related Short Courses
- Courses meet National Security Systems (CNSS) Training Standards
- Trust & Security Seminar Series
- Distinguished Lecture Series

Masooda Bashir

Cloud Security - Summary



From *“Challenges to Military Operations in Support of U.S. Interests”*

- U.S. forces depend
  - C4ISR
  - precision navigation/targeting
  - Communications (Above Figure)
- The barriers to entry even for high-end cyber warfare capabilities are low

Distributed Security Policy Conformance

Mirko Montanari, Ellick Chan, Kevin Larson,
Wucherl Yoo, Roy H. Campbell
{mmontan2, emchan, klarson5, wyoo5, rhc}@illinois.edu

Department of Computer Science
University of Illinois at Urbana-Champaign

June 8th, 2011

U.S. AIR FORCE

Policy Compliance in Large Distributed Systems

- Infrastructure security policies used by organization to manage their systems and provide a basic level of security

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

NERC
NORTH AMERICAN ELECTRIC
RELIABILITY CORPORATION

- Challenges:
  - How do we make compliance monitoring scale to large systems?
    - Large enterprise networks, Power grid, data centers
  - How do we make the monitoring system secure?

Proposed Approach

Distributed security assessment, *delegation*, *detection and response* leveraging shared configuration information and global policies

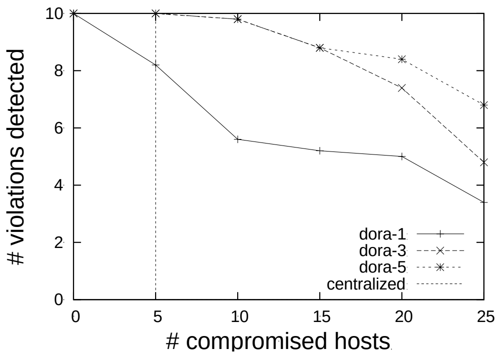
Goal -- Scalable and resilient system of systems that do not depend on static or hierarchical infrastructure

References

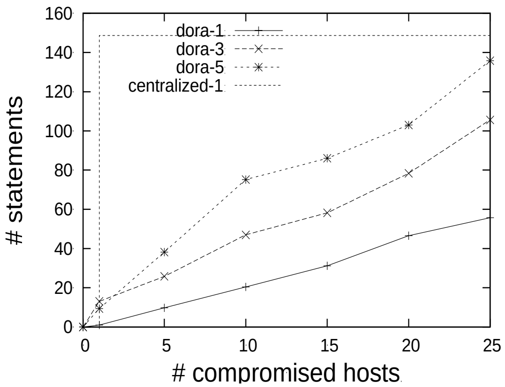
- Mirko Montanari and Roy Campbell, *Attack-resilient Compliance Monitoring for Large Distributed Infrastructure Systems*, 5th International Conference on Network and System Security (NSS 2011).
- Mirko Montanari, Ellick Chan, Kevin Larson, Wucherl Yoo, Roy H. Campbell, *Distributed Security Policy Conformance,* IFIP SEC 2011.

Approach

- State of system represented as logic statements using ontologies
  - Security and reliability requirements expressed as policies
  - Interactions between elements as workflows
- Distributed compliance monitoring avoids central bottlenecks and targets for attacks; disperses information and improves reactivity
  - Distributed reasoning algorithms for detection of states that violate policies
- Detection of violation of policies allows auditing, enforcement, and enables dynamic mapping of workflow operations.



Initial Monitoring Integrity Results



Initial Monitoring Confidentiality Results

Policy Compliance

- Rules that specify the desirable configuration and state of the infrastructure

Security Policies

- *All computer systems connected to the internal network must run an authorized anti-virus software*
- *Critical systems should be protected from multi-step attacks exploiting known vulnerabilities*

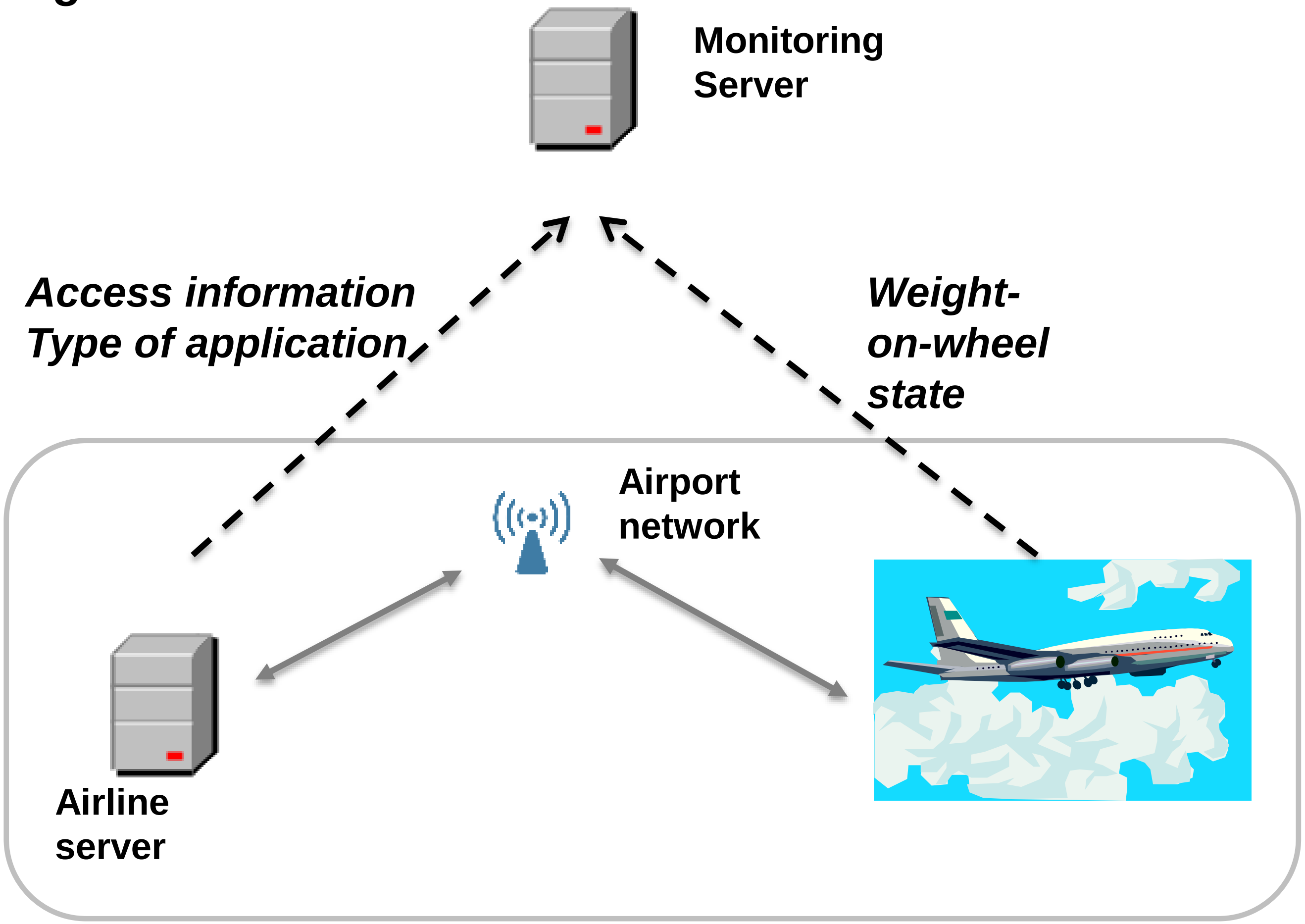
Infrastructure Policies (Airports, Power grid …)

- *Aircrafts are required to connect to the airport infrastructure when they touch ground*
- *Airline applications can be accessed only when the aircraft is parked at the gate*

Information Integration – General Architecture

Server for integrating information

Software running on each device that monitors the state of the system



We need to monitor for changes and evaluate their impact on the overall system

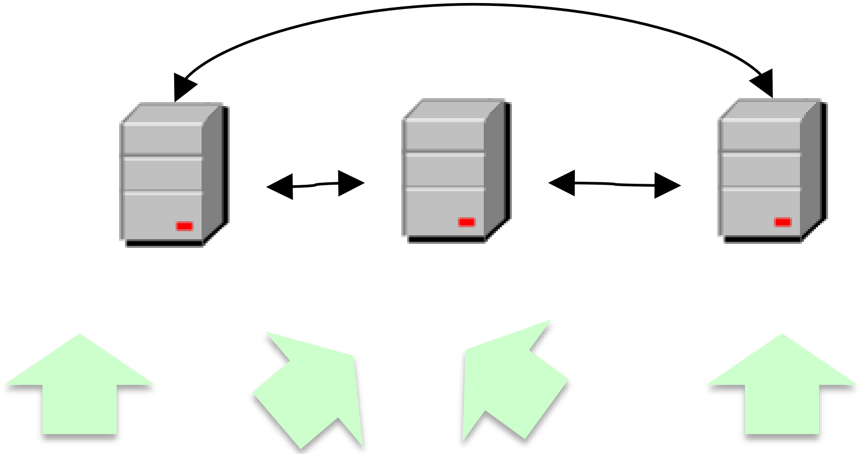
Security - Byzantine Replication

Verifiers: information is integrated redundantly in multiple servers.

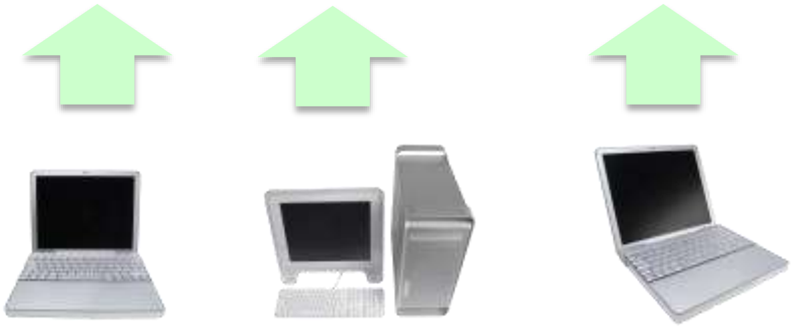
Verifiers can be managed by different departments in the organization

Violations are detected by using byzantine agreement

Agents: devices run software to monitor the state (e.g., forensic analysis, VM introspection)



Problem: each verifier needs to verify liveness and receive updates from all machines in the system



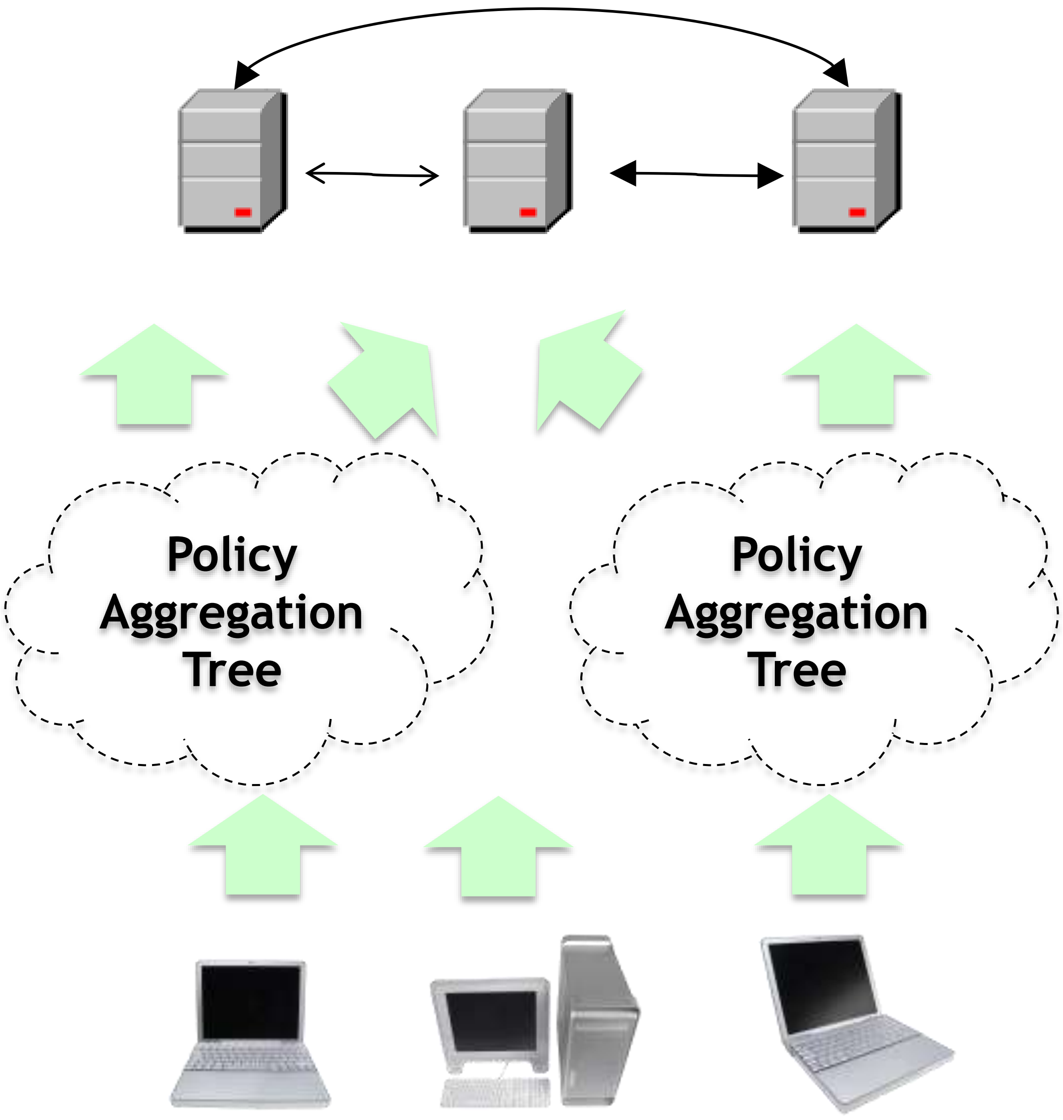
Odessa Architecture

We use *delegation* for making the solution scale

3) Scalable pub/sub architecture for managing failures of verifiers

2) Detection of liveness is distributed across multiple machines

1) Policy validation (partially) pushed to agents



Configuration Management - Policies

- **State and configurations are represented using RDF (Datalog)**
- Policies are specified using Datalog rules

Airport Network Infrastructure

aircrafts must connect with the airport wireless network after landing for updating software

(A type Aircraft), (A weight-on-wheels TRUE), NOT (A connectedTo N), (N partof P), (P type Airport) → FAIL

Enterprise Networks

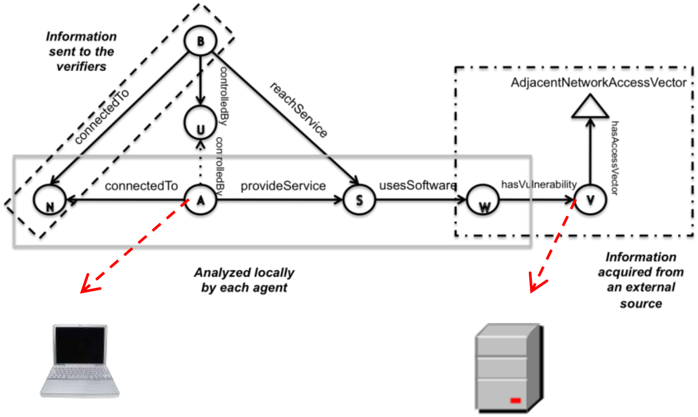
Malicious users must not be able to compromise ***critical systems using sequences of known vulnerabilities***

(H type CriticalHost), (U type MaliciousUser), (U canCompromise H) → FAIL

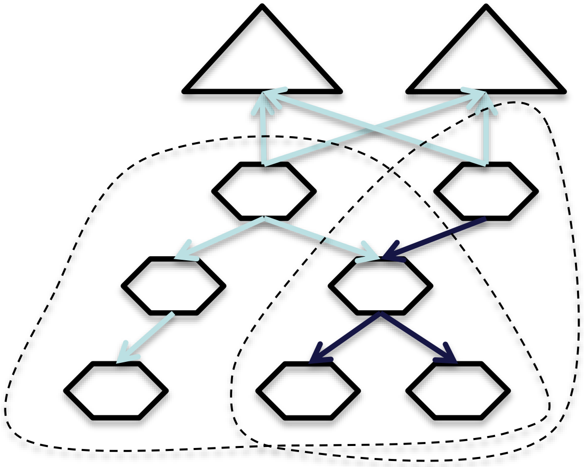
(U canCompromise H_1), (H_1 canCommunicate S), (S type Service), (S providedBy H_2), (S hasVulnerability V) → (A canCompromise H_2)

Scalability Mechanisms

1. Distribution of policy validation
   - Policies are split into a portion of the rule that can be validated locally on each machine


Information sent to the verifiers
B
connectedTo
controlledBy
U
controlledBy
reachService
AdjacentNetworkAccessVector
hasAccessVector
N
connectedTo
A
provideService
S
usesSoftware
W
hasVulnerability
V
Analyzed locally by each agent
Information acquired from an external source

2. **DHT-based mechanisms for introducing new verifiers upon failures and for detecting failures**
   - **Pub/Sub for disseminating information about new verifiers and for detecting failures**



1) Distribution of policy validation

- Rule analysis algorithm matches parts of rules with sources of information
- Partial validation of policies can be performed by agents
  - Reduce the information to share globally for efficiency and privacy

Rule graph generation

Each rule is transformed in a graph and meta-information from the annotation are integrated in the representation

Determination of local statements

Each agent determines the statements that can be found only locally

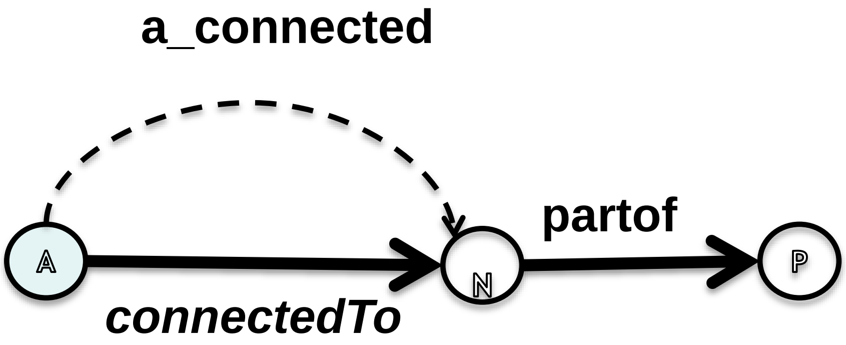
Rule execution

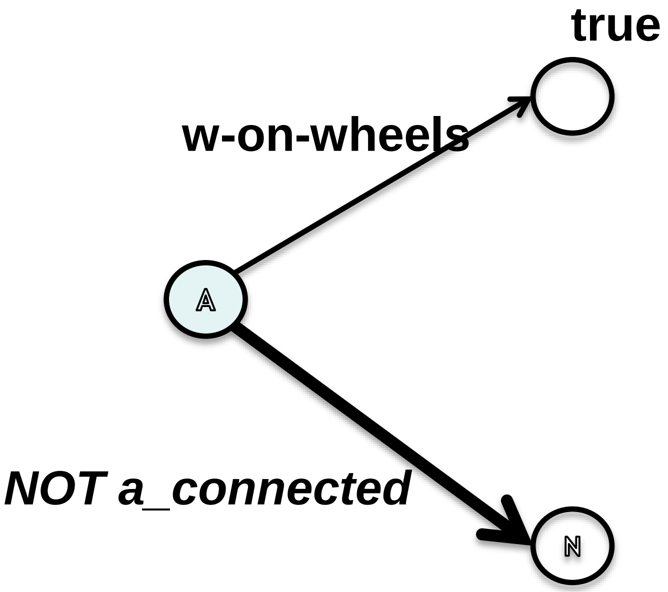
Statements are exchanged between agents to complete the evaluation

A) Rule Graph Generation

(A connectedTo N), (N partof P)
→ (A a_connected N)

(A w-on-wheels true),
NOT (A a_connected N) → fail

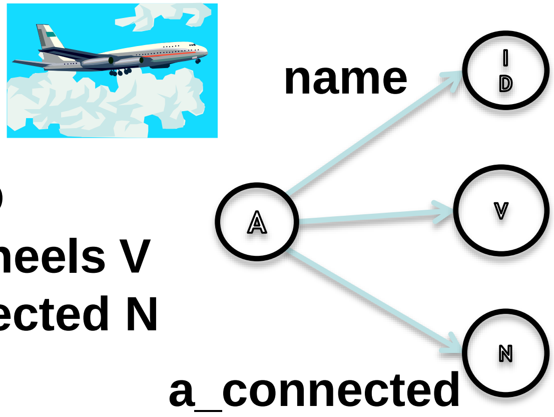




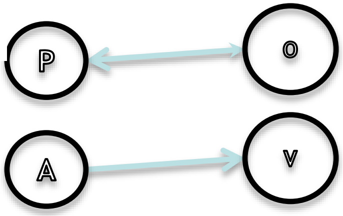
Information Sources

- **Each agent provides specific information about the system**
- **The statements that are generated only locally are represented in a “local graph”**

Aircraft$_1$ A
aircraft$_1$ name ID
aircraft$_1$ w-on-wheels V
aircraft$_1$ a_connected N



Airport P:
O operatingAt P
A w-on-wheels V



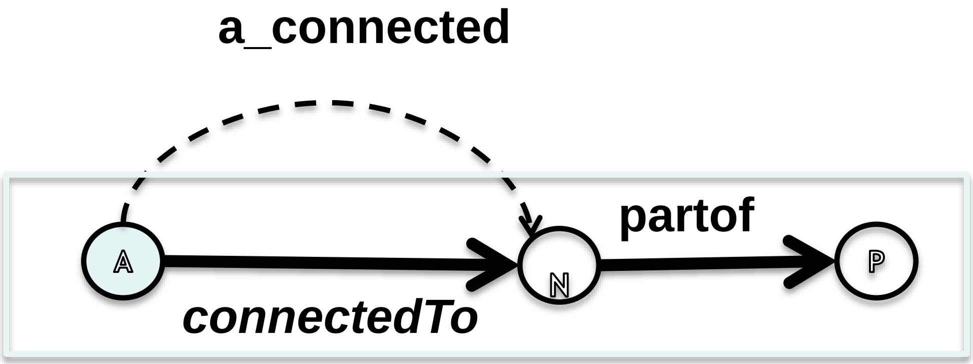
- Given this information **we know that certain predicate can be generated only by a specific device**

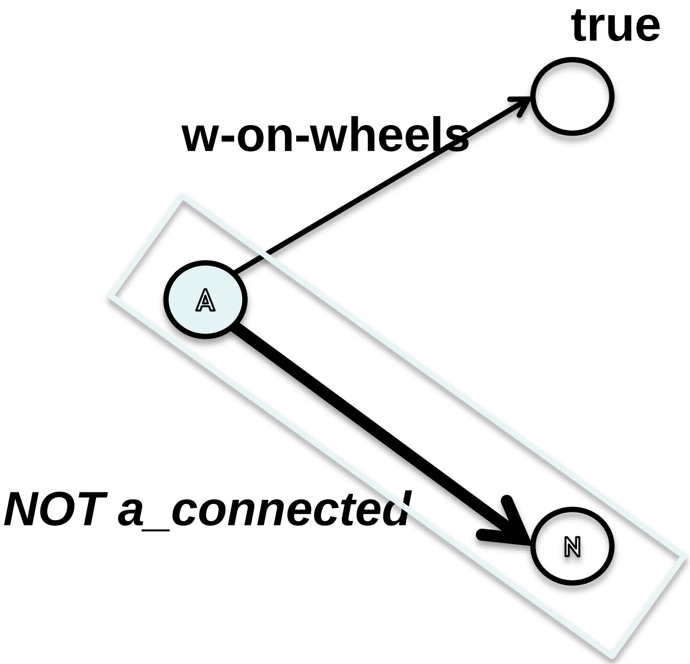
given a specific airplane A, its ID is provided only by A
given a specific airplane A, the list of networks is generated only by A
given a specific airport P, the list of operating airlines is provided by P

B) Source Propagation

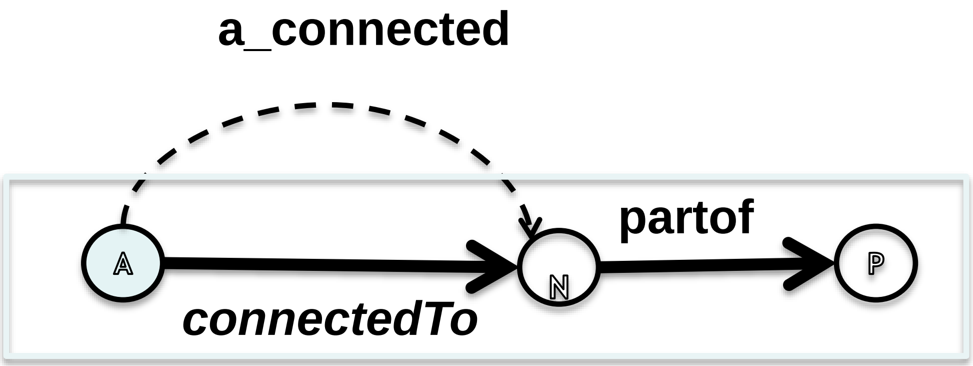
(A connectedTo N), (N partof P)
→ (A a_connected N)

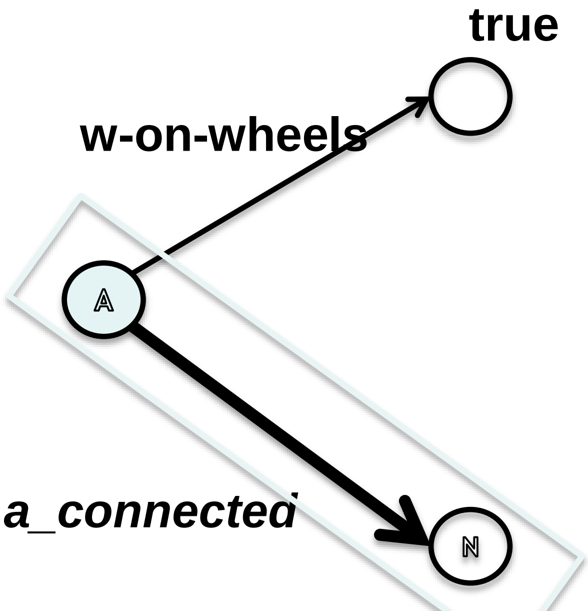
(A w-on-wheels true),
NOT (A a_connected N) → fail





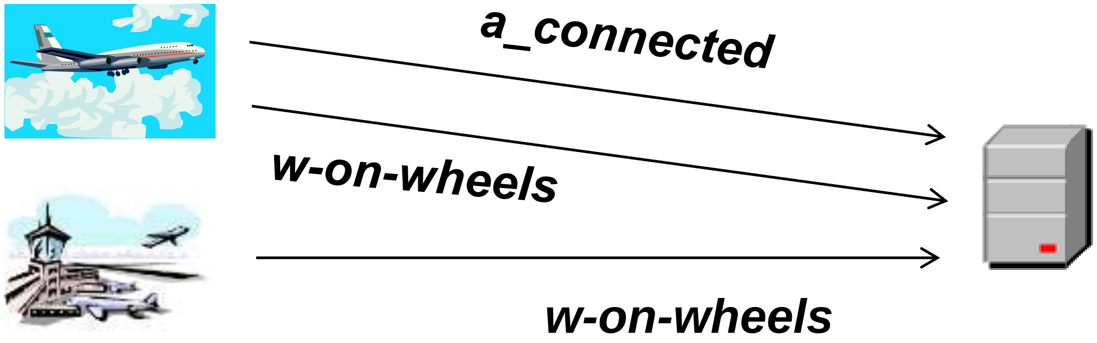
C) Execution





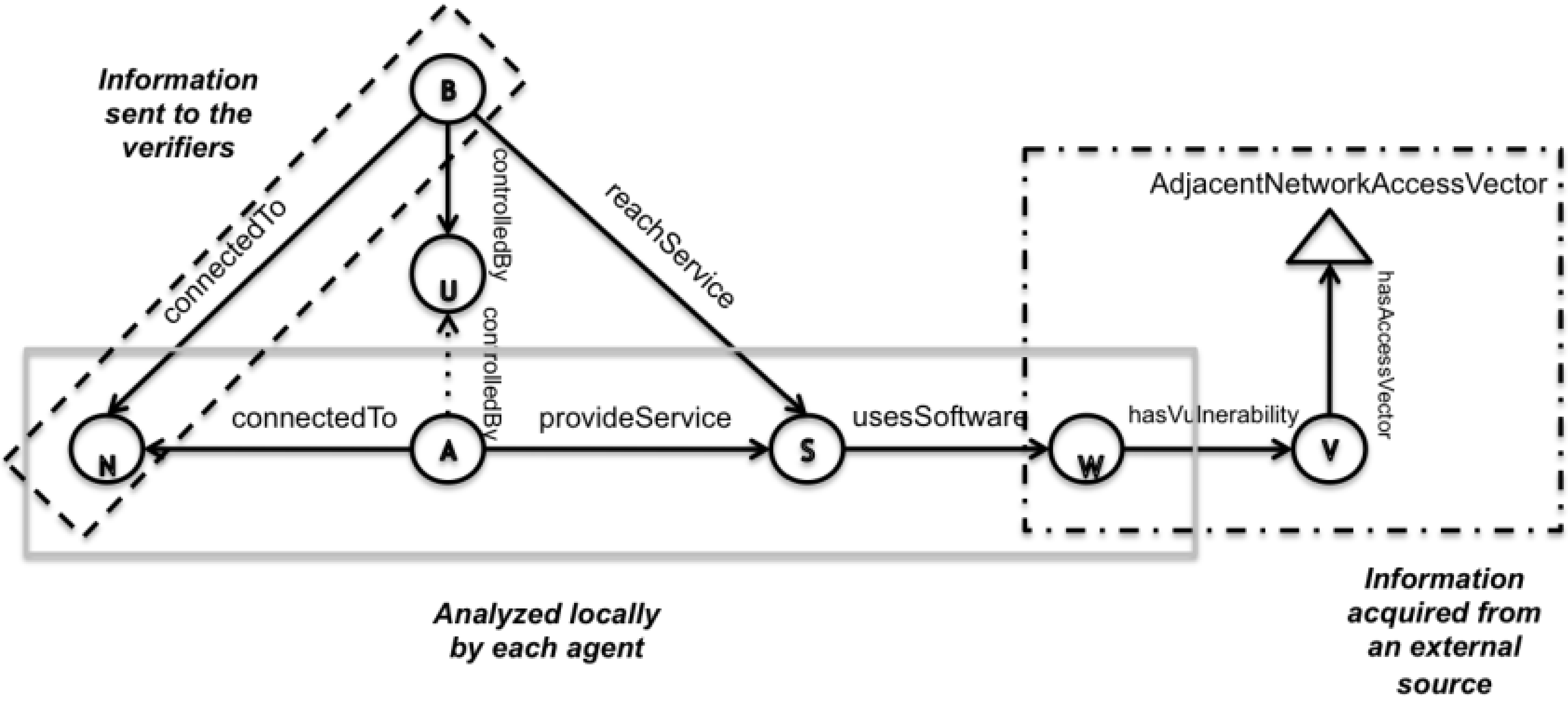
Rule processed locally:
A connectedTo N AND N partof P
→ A a_connected N

Rule processed at the verifier
A a_connectedTo N AND A w-on-wheels true → FAIL



Partial validation of complex rules

- Complex rules can be partitioned in multiple parts.
- Some parts can be validated locally, others are validated in the verifiers


Information sent to the verifiers
B
U
N
A
S
W
V
connectedTo
controlledBy
reachService
controlledBy
connectedTo
provideService
usesSoftware
hasVulnerability
hasAccessVector
AdjacentNetworkAccessVector
Analyzed locally by each agent
Information acquired from an external source

2) Pub Sub mechanism

- Each verifier needs to acquire information about all hosts that provide specific types of statements
  - A *a_connectedTo* N, A *w-on-wheels* true → FAIL
  - All agents generating statements about "*a_connectedTo*" and "*w-on-wheels*" need to send information to verifiers

- We generate a DHT SCRIBE topic *H(P)* for each predicate *P*
  1. All agents subscribe to the topics of the predicates they potentially provide
  2. New verifier notify agents by publishing a message in all topics relevant to the rules
  3. Agents maintain information about verifiers and send information directly to them

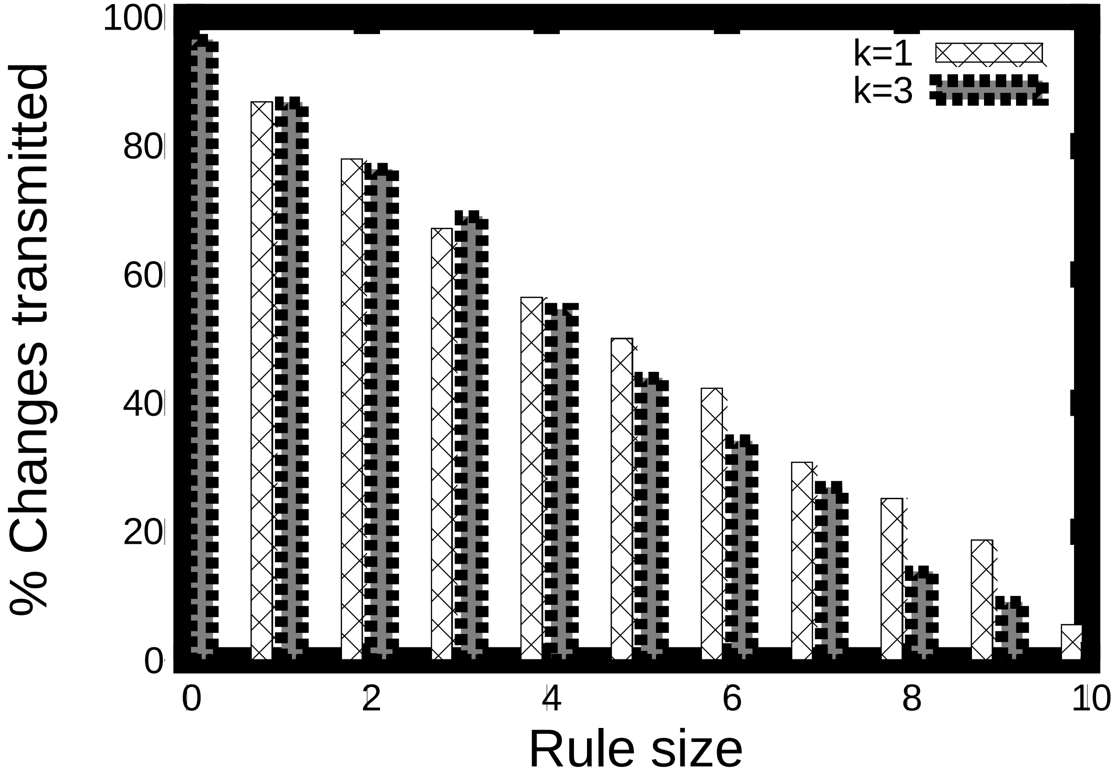
Experimental Results

- Odessa implemented in Java and C
  - Communication built on top of Freepastry
  - To increase the trustworthiness of agents, we run them in Dom0 when possible.

Mechanism	Configuration obtained
Dom0 (XenAccess, file system)	Running processes, network connections, configuration files
Host VM (Linux kernel module)	Fast detection of new network communications

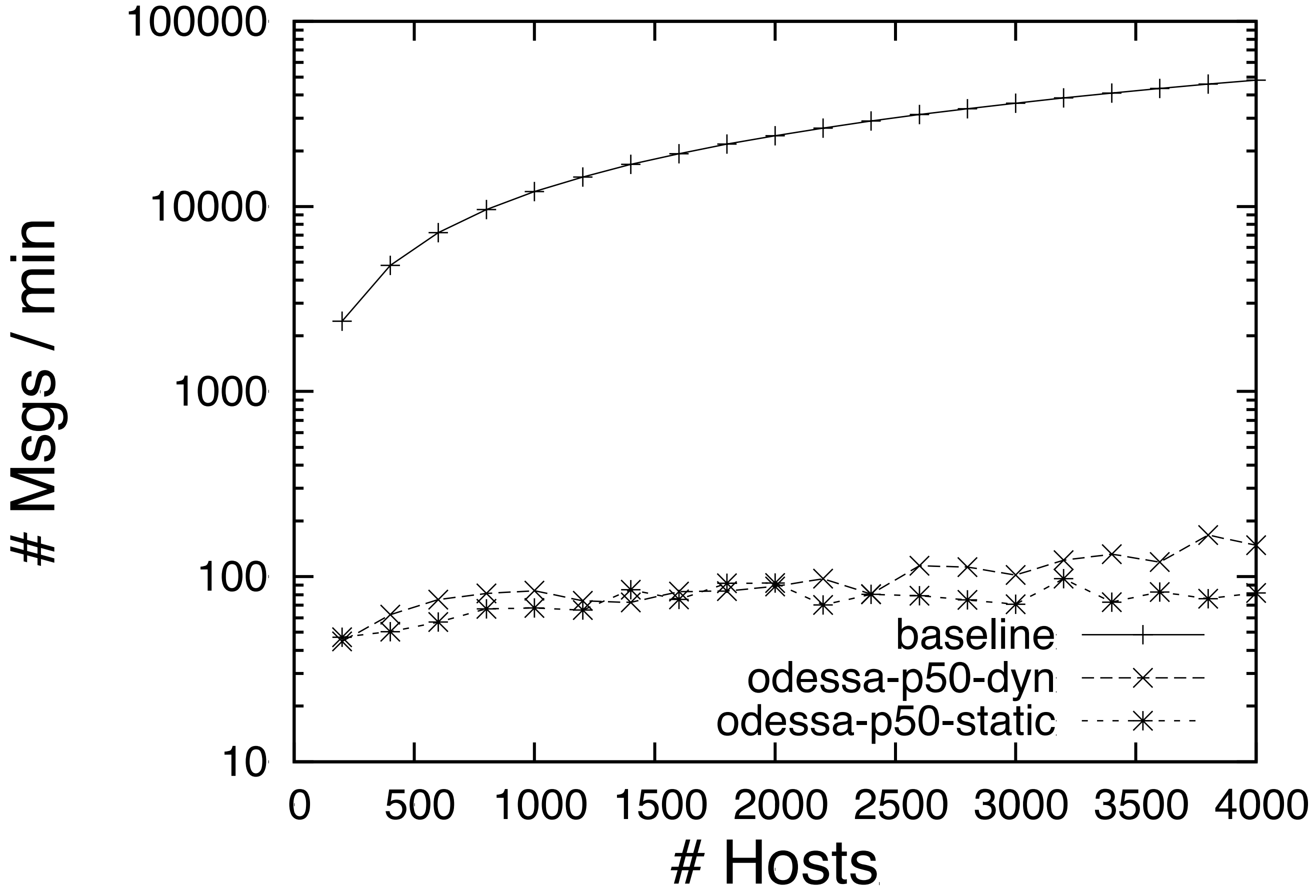
- **Using such information, we Implemented policies for validating:**
  - **Presence of specific programs**
  - **NFS authorizations across networks**
  - **Attack graph generation**

Delegation Experiments: Reduced Load



When large portions of the rules are processed locally, the amount of information transmitted to verifiers because of configuration changes is reduced

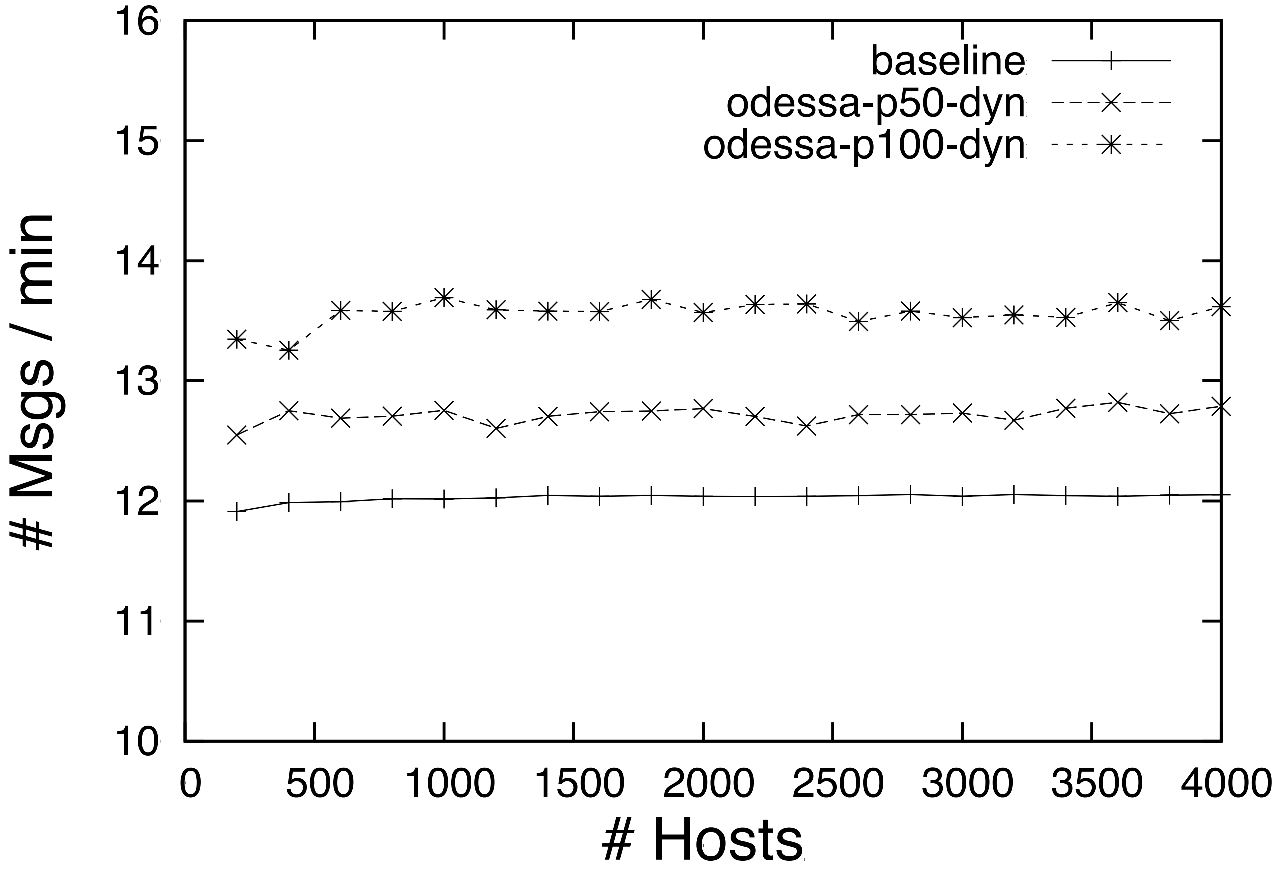
Scalability Experiments: Maximum Load



Maximum number of messages sent by nodes (log-scale)

Odessa reduces of orders of magnitude the load on any central monitoring host.

Scalability Experiments: Average Load



Average number of messages sent and received by each node for monitoring

Odessa does not significantly increase monitoring overhead compared to a centralized solution

Summary of Security Characteristics

Compromised verifiers

- Policies are validated redundantly on several verifiers
  - Byzantine agreement between verifiers

Compromised agents

- Multiple agents acquires independently the same information about the state

Hardening of the agents

- Agents are separated from the device they monitor
  - Forensic information
  - Virtual machine introspection

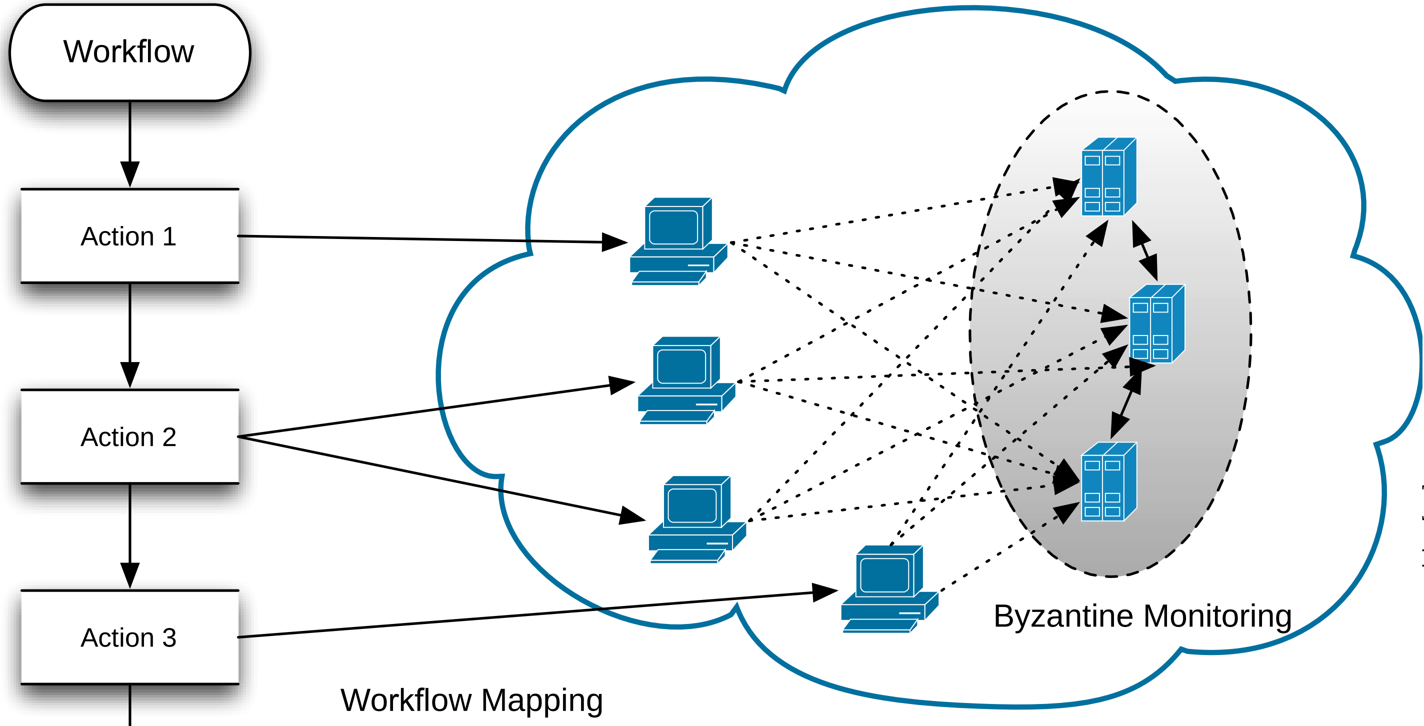
Related Work

- SNMP, WBEM
  - Good protocols for communicating with agents and acquiring information. Their implementations often rely on a centralized architecture
- TVA [Jajodia '03], MulVal [Ou '06]
  - Scanning is slow in detecting policy violations. Multiple scannings for redundancy increase network load.
- Top Down management architecture [Narain '08]
  - Completely rely on centralized control. If the central point is compromised, the architecture is insecure

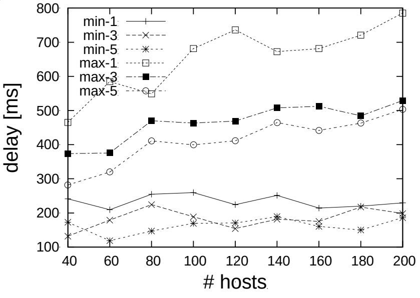
Policy-based Dynamic Mapping of Services and Workflows

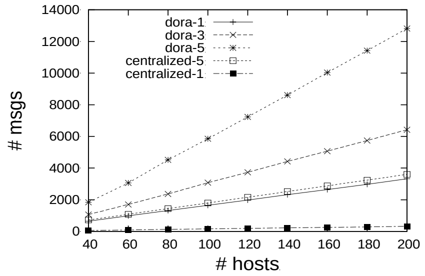
- Dynamic mapping of services require by workflows to systems that implement them
- Guiding organizational policies that support change in response to dynamic changes
- Choice of services for workflow respects security policies
- Detection of policy violations
- Optimized algorithms to perform dynamic and distributed mapping between workflows and services

Mapping and Monitoring

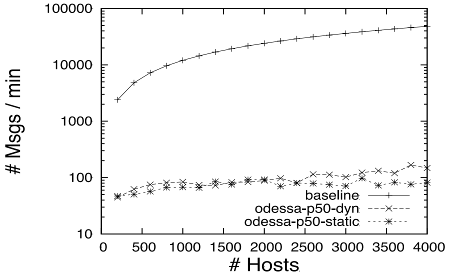


Delay





Compliance Monitoring Load

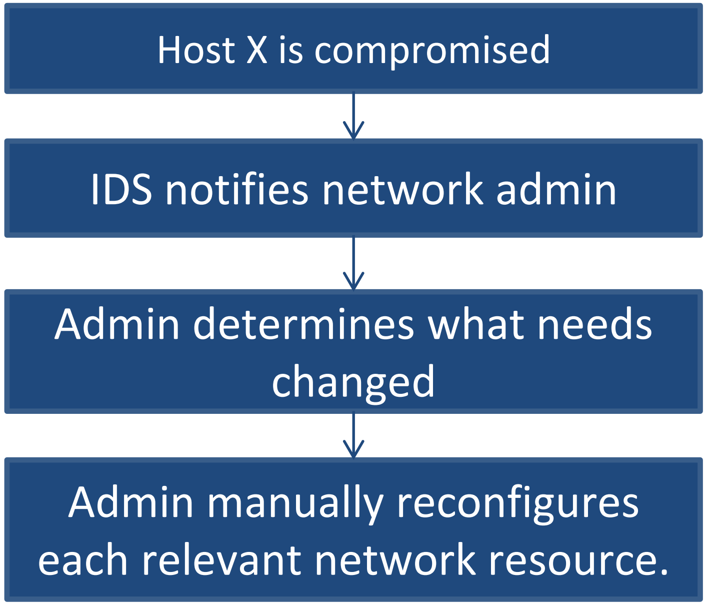


Liveness Monitoring Load

Network Policy Management Extension

Manual Policy Enforcement

- Network administrators configure hosts, switches and middleware manually.
- This process is slow and error-prone.
- Cloud networks are far too dynamic to be managed with manual configuration.


Host X is compromised

IDS notifies network admin

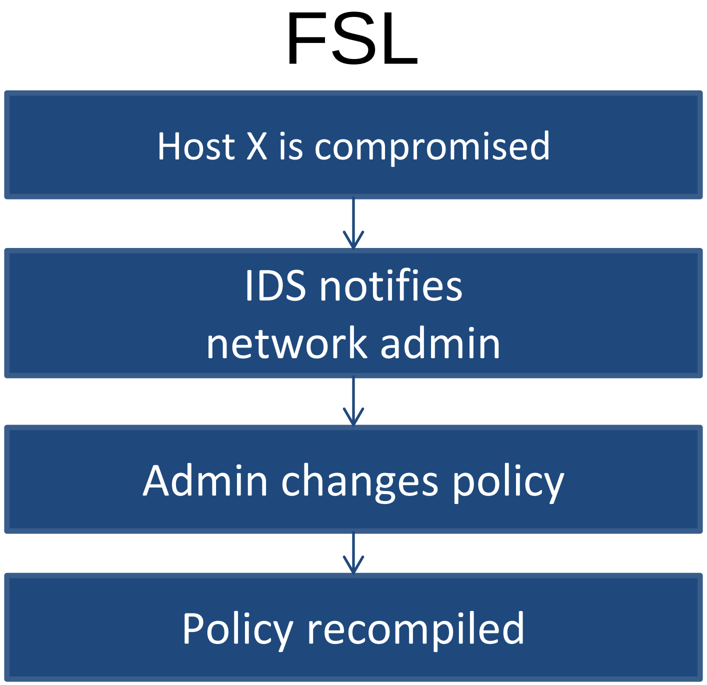
Admin determines what needs changed

Admin manually reconfigures each relevant network resource.

Current Static Network Policy Management

Static Policy (e.g. FSL)

- Policy-based network configuration consolidates configuration data.
- Administrators write policies to define network operation.
- However, policies apply to individual hosts.
- Changing policies requires recompiling.


FSL

Host X is compromised

IDS notifies network admin

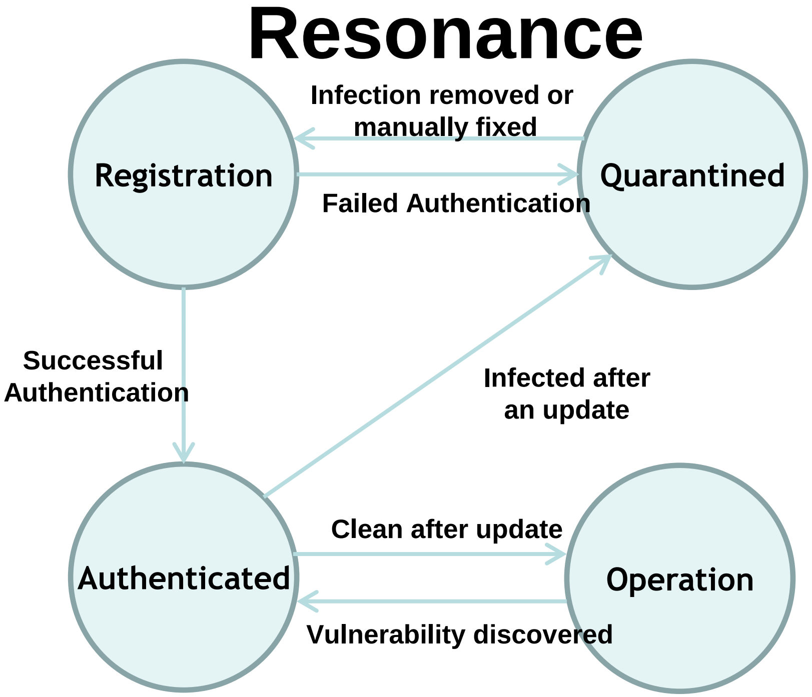
Admin changes policy

Policy recompiled

State-based Static Policy Management

State-based Policy (e.g. Resonance)

- Resonance provides limited dynamic policy enforcement with finite-state machine
- Not all systems can be modeled with a reasonable number of states
- Forces policies into a rigid paradigm

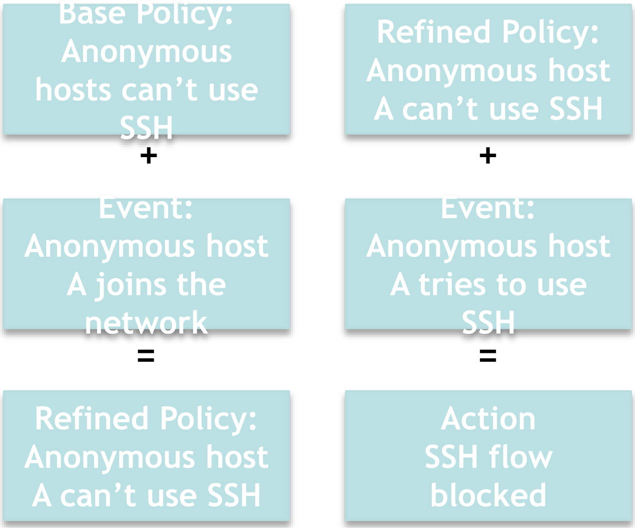


Proposed Solution: Dynamic Policy

Using inference, dynamic policy system checks network events (Observed Data) against a set of given conditions (Base Policy). When a given condition is satisfied, the inference engine produces:

- Actions – Changes to the network necessitated to enforce policy
- Refined Policy – New conditions amended to Base Policy

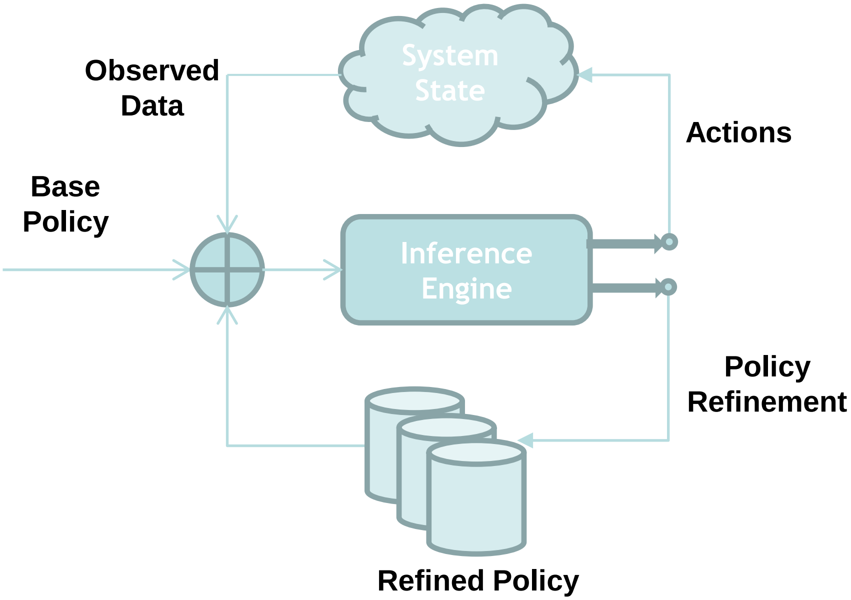
Example



Base Policy: Anonymous hosts can't use SSH
+
Event: Anonymous host A joins the network
=
Refined Policy: Anonymous host A can't use SSH

Refined Policy: Anonymous host A can't use SSH
+
Event: Anonymous host A tries to use SSH
=
Action SSH flow blocked

Dynamic Policy Information Flow



Proposed Solution: Dynamic Policy

Static Policy	Dynamic Policy
Administrators define policies for individual hosts.	Administrators define general base policies for classes of hosts.
Violations must corrected manually.	Violations can be automatically corrected upon detection.
Every rule must be manually defined by the administrator.	Refined policies can be logically inferred from existing policies and data.
Incurs little computational overhead.	Can be resource intensive.

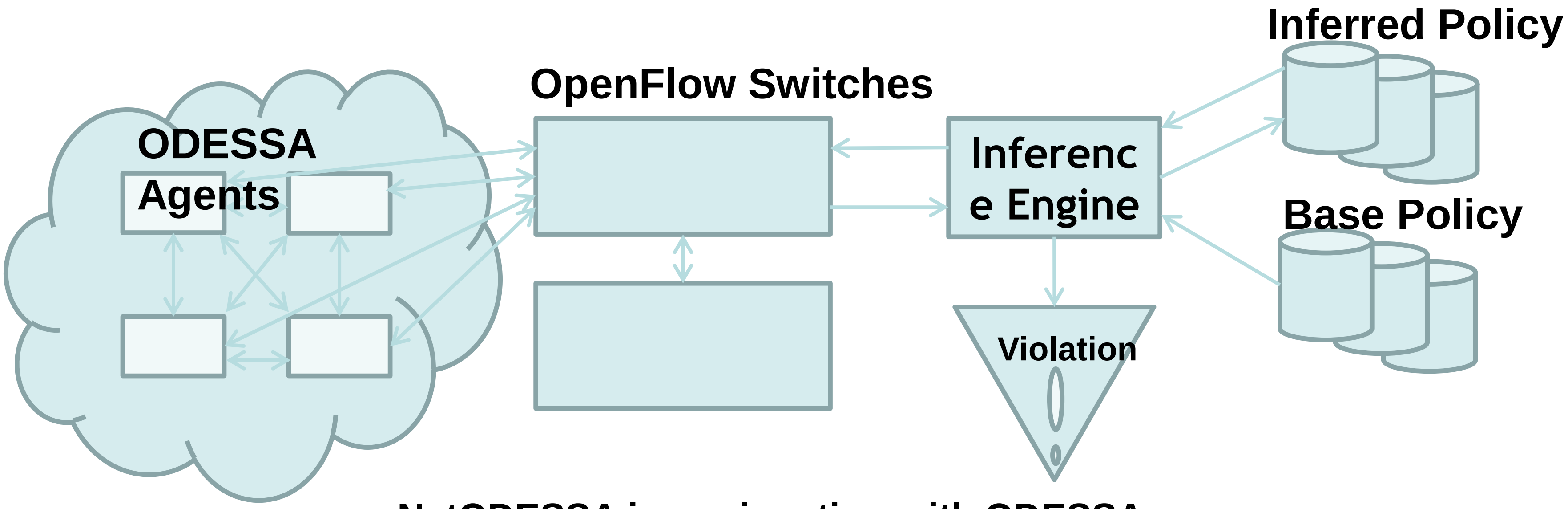
Dynamic Policy in the Network

Dynamic Policy is implemented in the network architecture using programmable switches. This enables policy to be context aware, adapting itself to the state of the network at runtime.

This design offers additional advantages:

- Cannot be directly altered by end hosts, malicious code, etc.
- Policy can be automatically enforced
- Required for some policies, e.g. path specification
- Can improve network efficiency, not just security

Our Design: NetODESSA

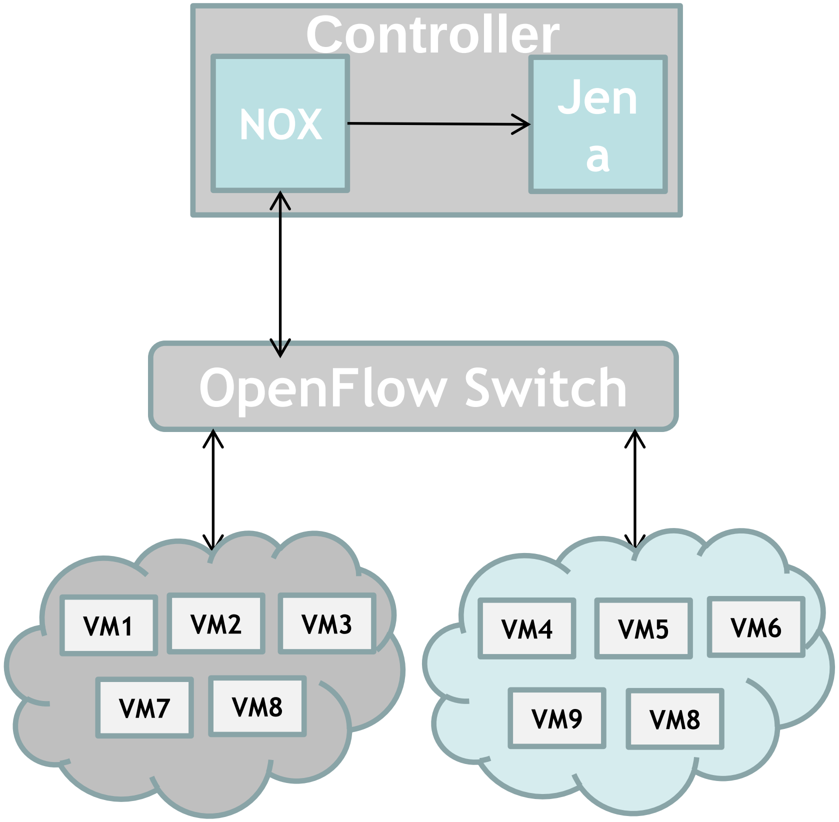


NetODESSA in conjunction with ODESSA

Experiment: Inference Benchmarking

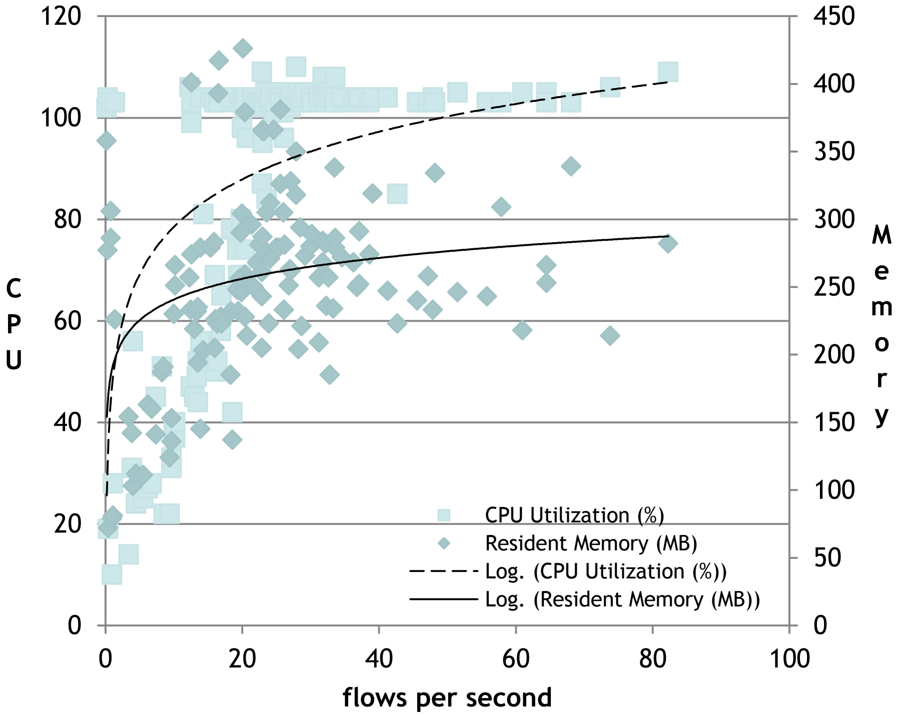
In this experiment, we simulated monitoring between two networks connected with an OpenFlow switch, using a NOX controller.

Our goal was to implement basic policy monitoring and to measure the resource utilization for performing policy inference.

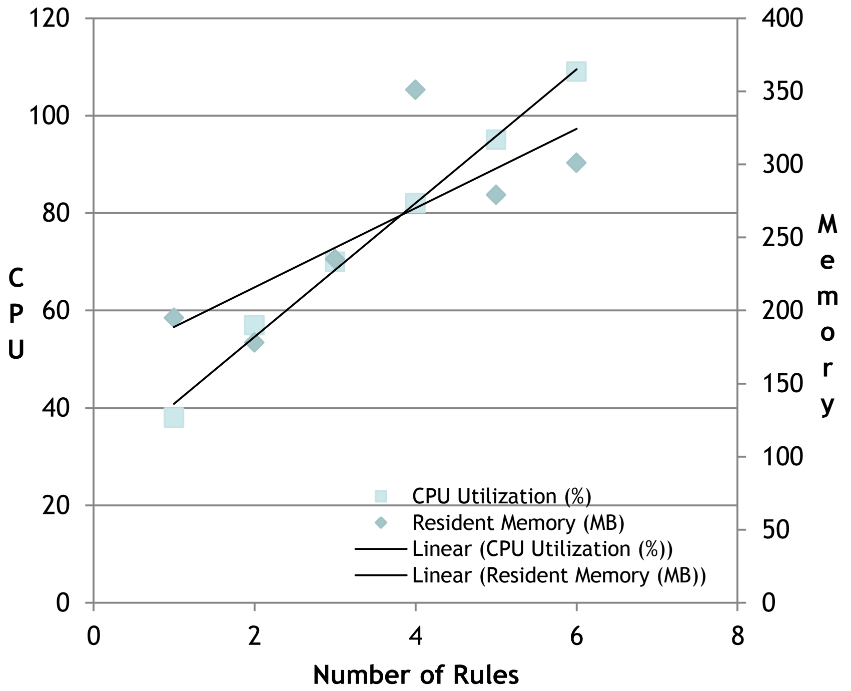


Experiment: Inference Benchmarking

From our results, we conclude that dynamic policy monitoring will be bound by the limitations of physical resources. However, our current inference engine uses the OWL reasoner, which is not suited as well for our purposes as others. Previous work has indicated that a more sophisticated implementation with a specialized reasoner will be more scalable.



This graph shows resource utilization relative to the amount of network traffic being observed.



Here we see how resource utilization trends with respect to the number of rules being checked.

Conclusions

- Policy compliance is an important component of the security posture of large organizations

- Policy compliance monitoring system need to be scalable and secure
  - Our architecture increases the security by introducing replication of monitoring
  - Delegation is used decrease the load and make our solution scale to large networks

- Future Work
  - Automatic reconfiguration of the agents to recover from violations
  - Consistency for detecting correctly short-lived violations

Acknowledgements

Information Trust INSTITUTE

BOEING®

TCIPG

Assured Cloud Computing University Center of Excellence

ILLINOIS