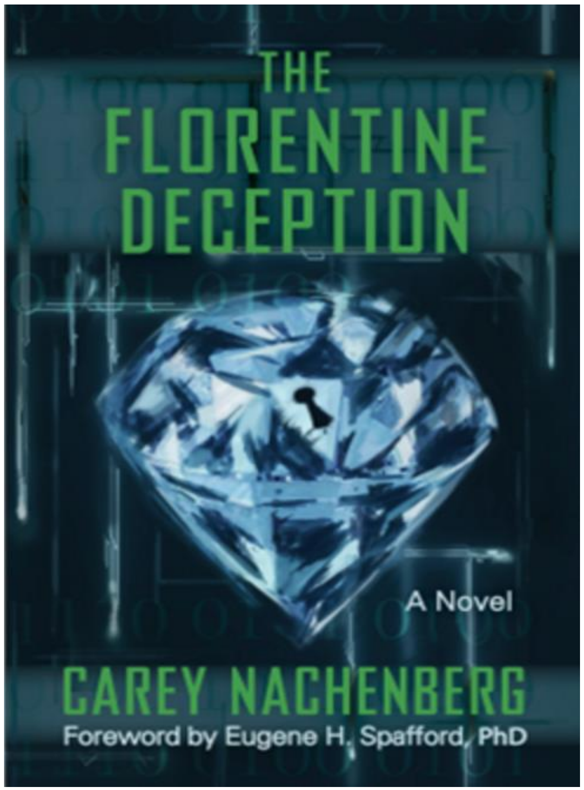
Engineering in Cyber Resiliency: A Pragmatic* Approach

Bill Sanders
Donald Biggar Willett Professor of Engineering
Head, Department of Electrical and Computer Engineering

* but not perfect

Being Pragmatic:
Separating Science Fantasy from Science Fact



*"If there's one disadvantage to spending more than a quarter of a century in security, it's that you become hypersensitized to **mangled terminology and fantasy passed off as current science**"*

David Harley, Senior Research Fellow, ESET

Said when speaking about *The Florentine Deception* by Carey Nachenberg.

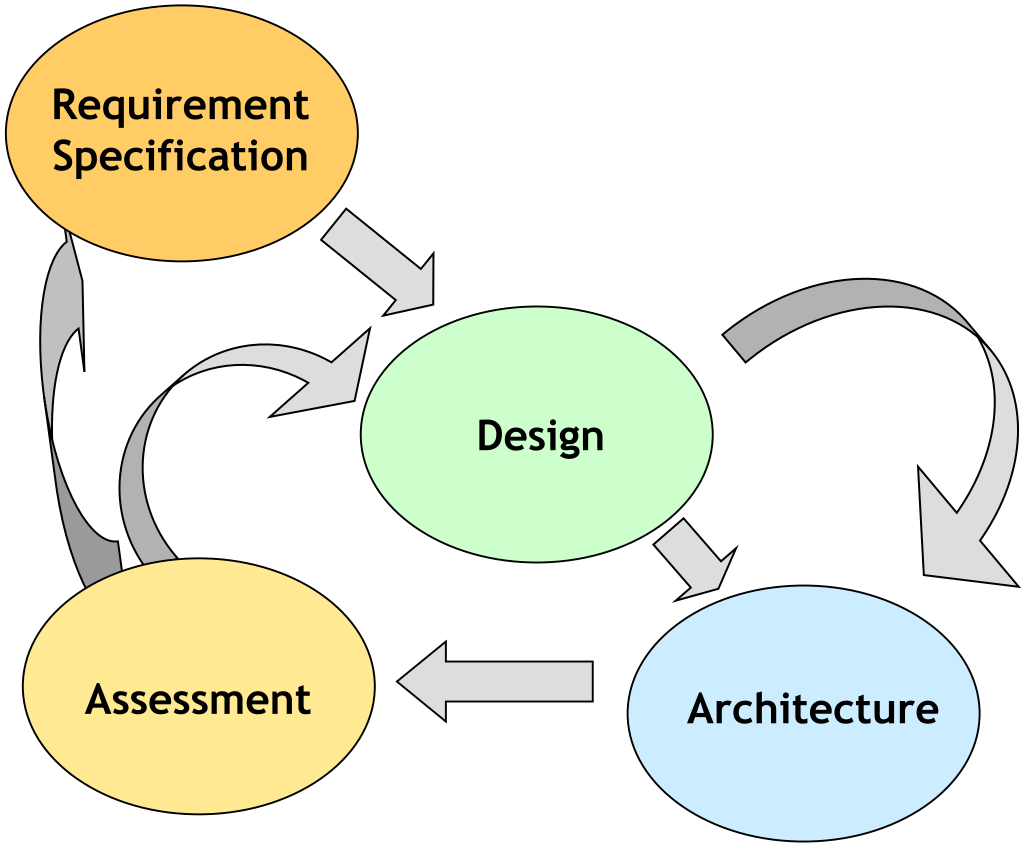
Cyber Security Facts (according to Sanders)

- Cyber systems are **complex**, and their complexity will only continue to increase.
- **Absolute cyber security is unattainable**.
- Cyber systems intended to be trustworthy must **operate through attacks.**
- Protect the best you can, but realize that **perfect protection is impossible, so resiliency can only be achieved through tolerating attacks through online detection and response**.
- **Assessment** of the "amount" of security that a particular resiliency approach provides **is essential**.
- **Perfect cyber security is science fantasy, and perfection is the enemy of good.**

THE CRITICAL NEED: Provide Assured Trustworthy System Operation in Hostile Environments

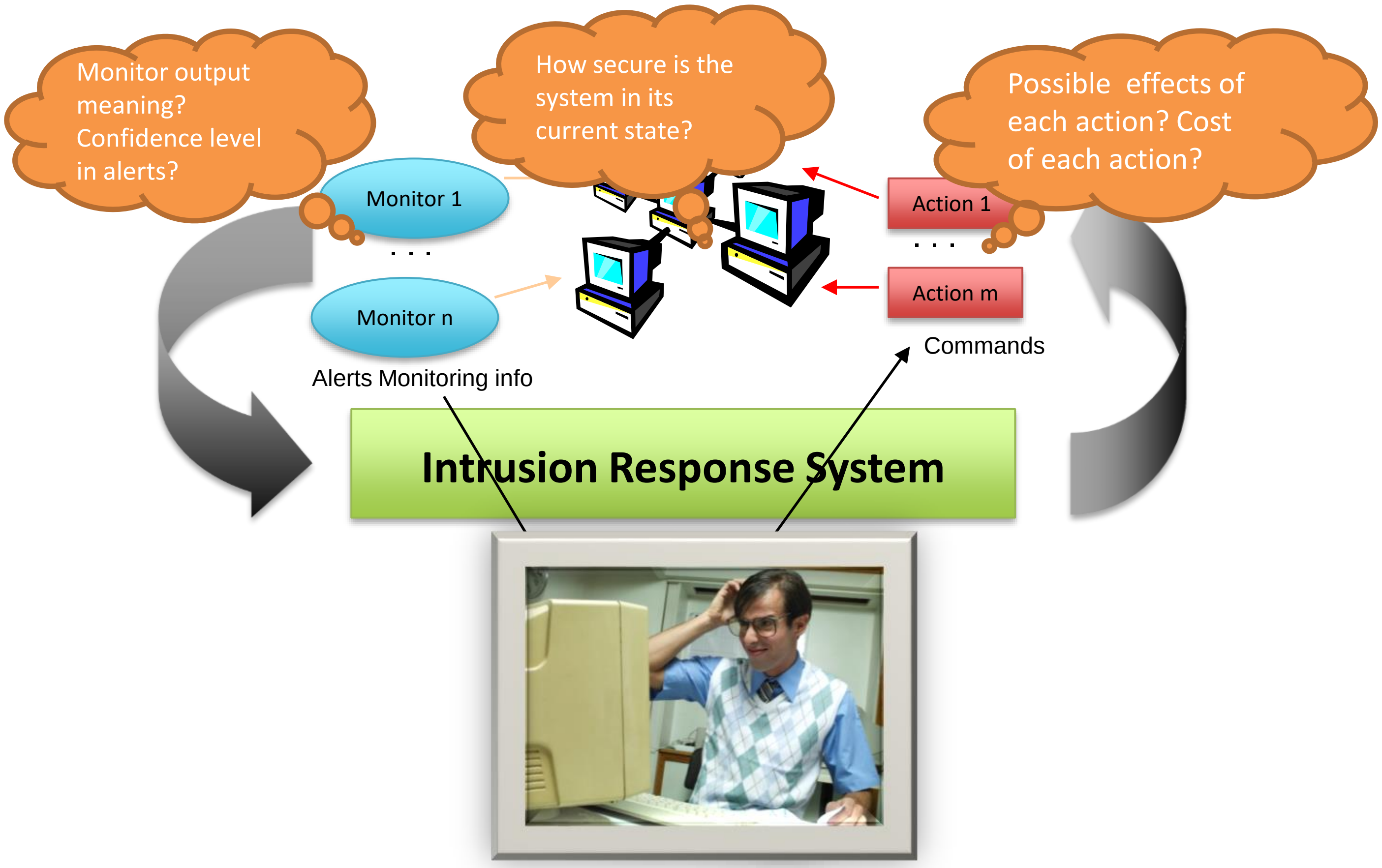
- **Be Trustworthy**
  - A system which does what is supposed to do, and nothing else
  - Availability, Security, Safety, …
- **Tolerate a Hostile Environment**
  - Accidental Failures, Design Flaws, and Malicious Attacks
- **Consider the cyber, physical, and social system aspects**
- **Provide Assurance through Assessment**
  - Provide justification that the system will operated as expected
  - Choose among design alternatives to achieve greater trustworthiness.

Engineering in Resiliency: Trust Specification, Design, Implementation, and Validation

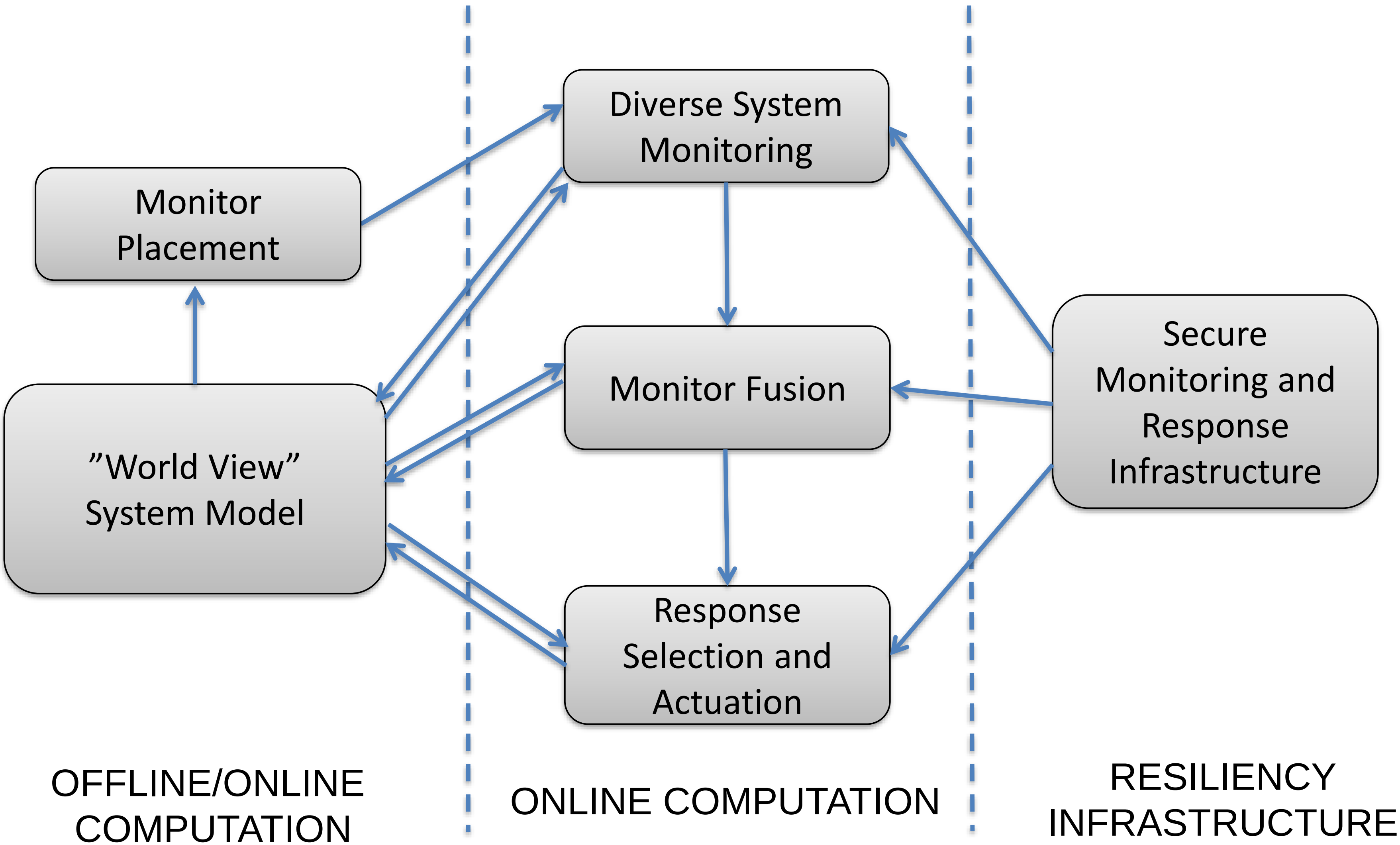


Engineering in Resiliency: Design and Architecture

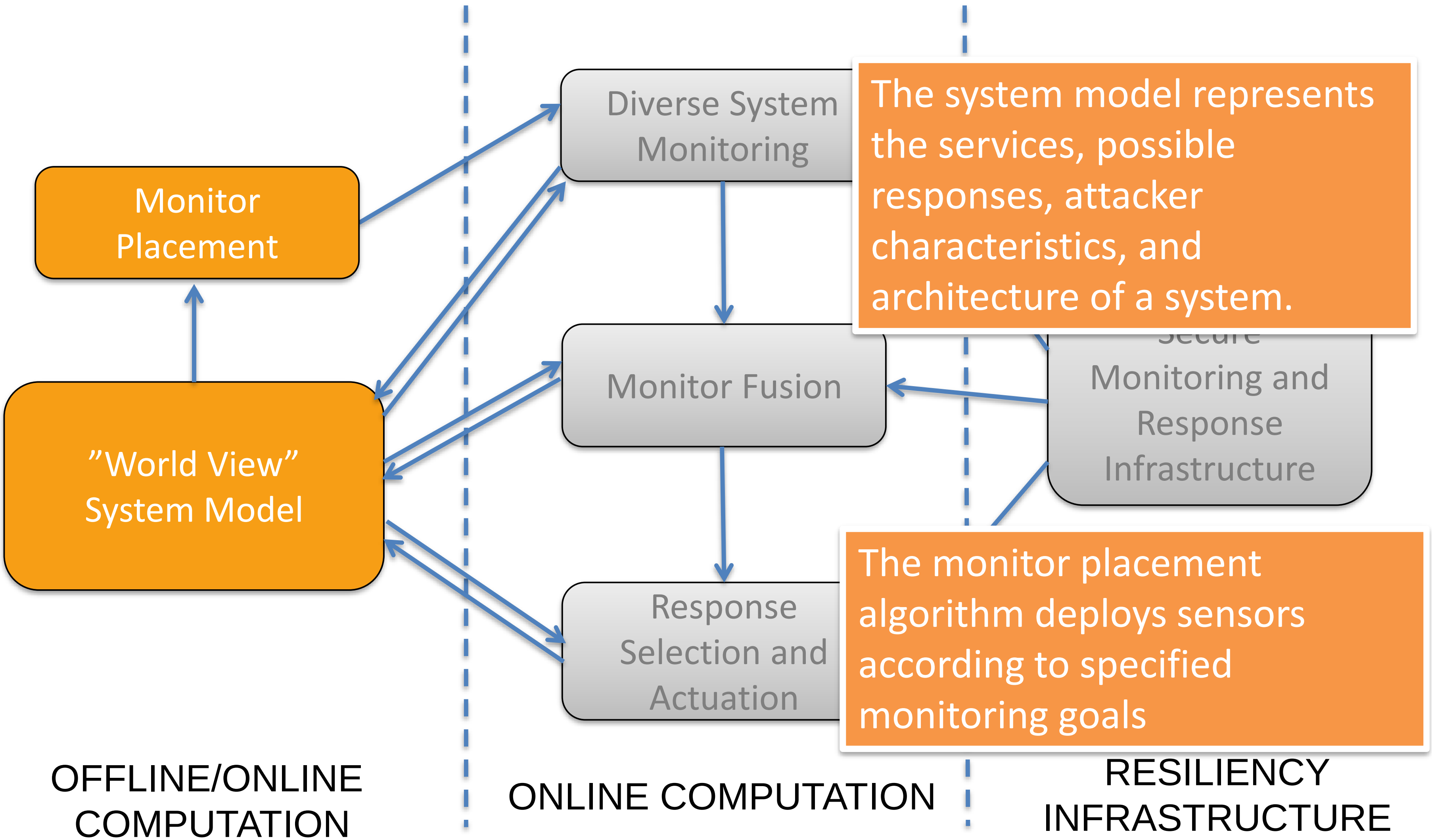
Resiliency Design Challenges



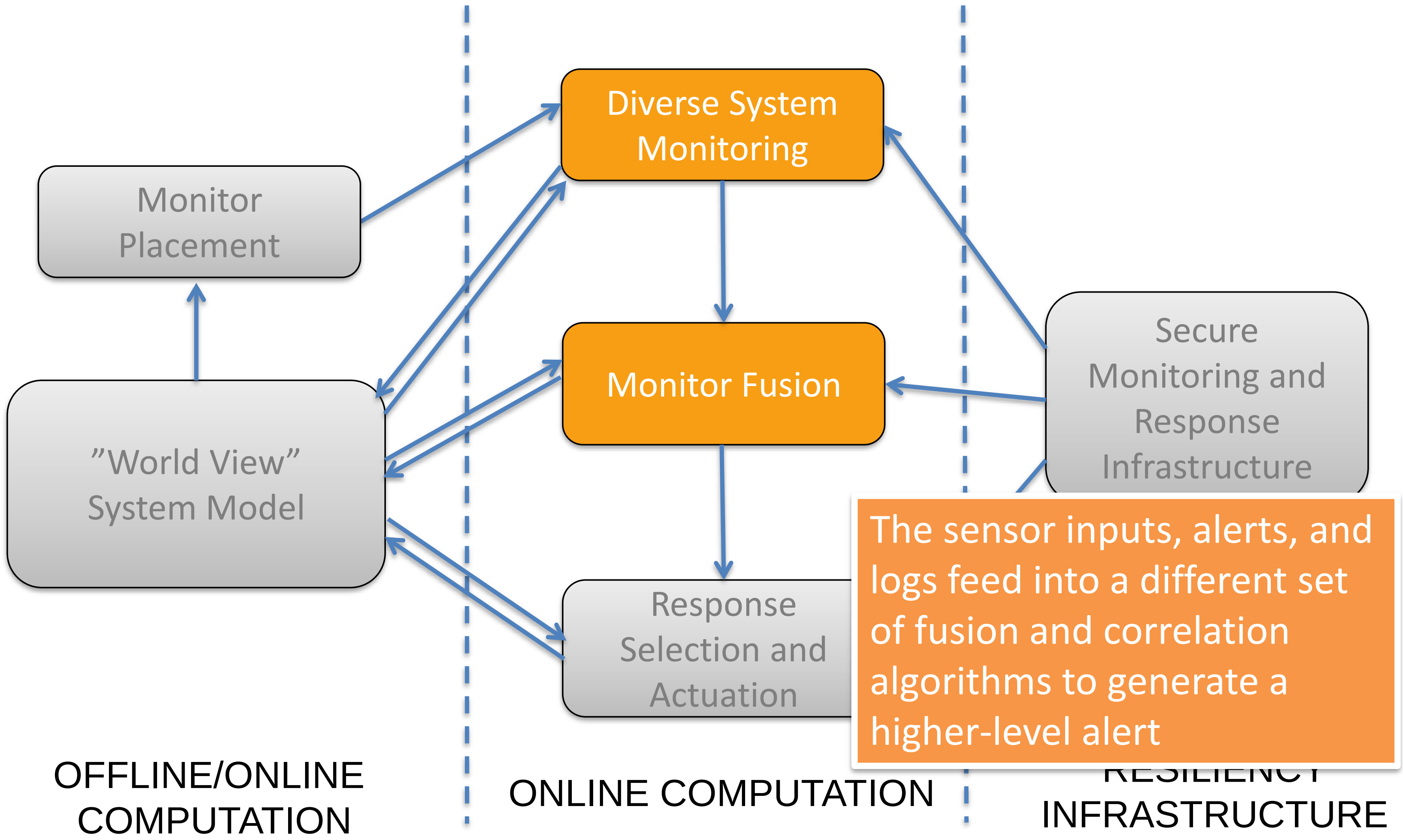
Notional Architecture for Resiliency



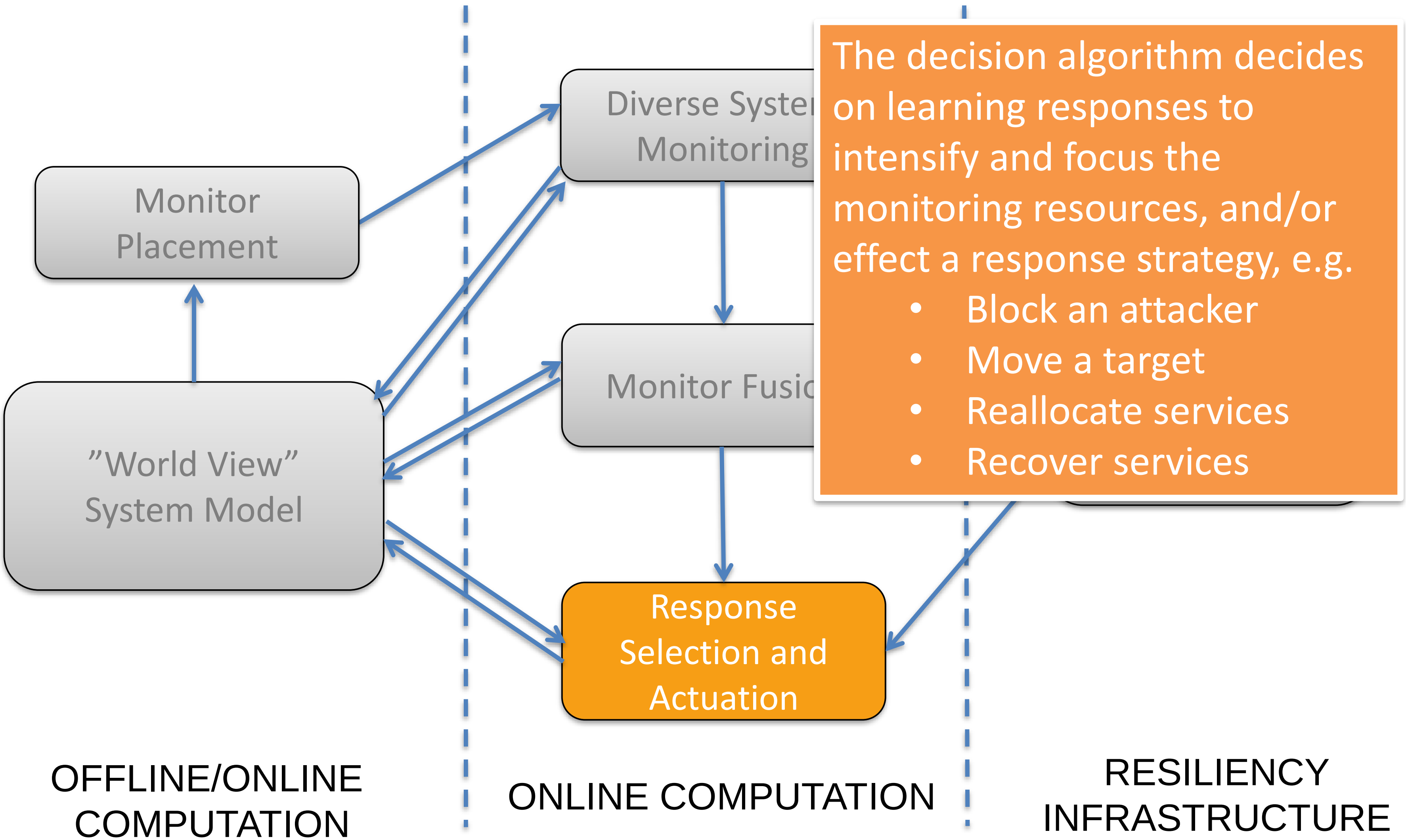
Notional Architecture for Resiliency



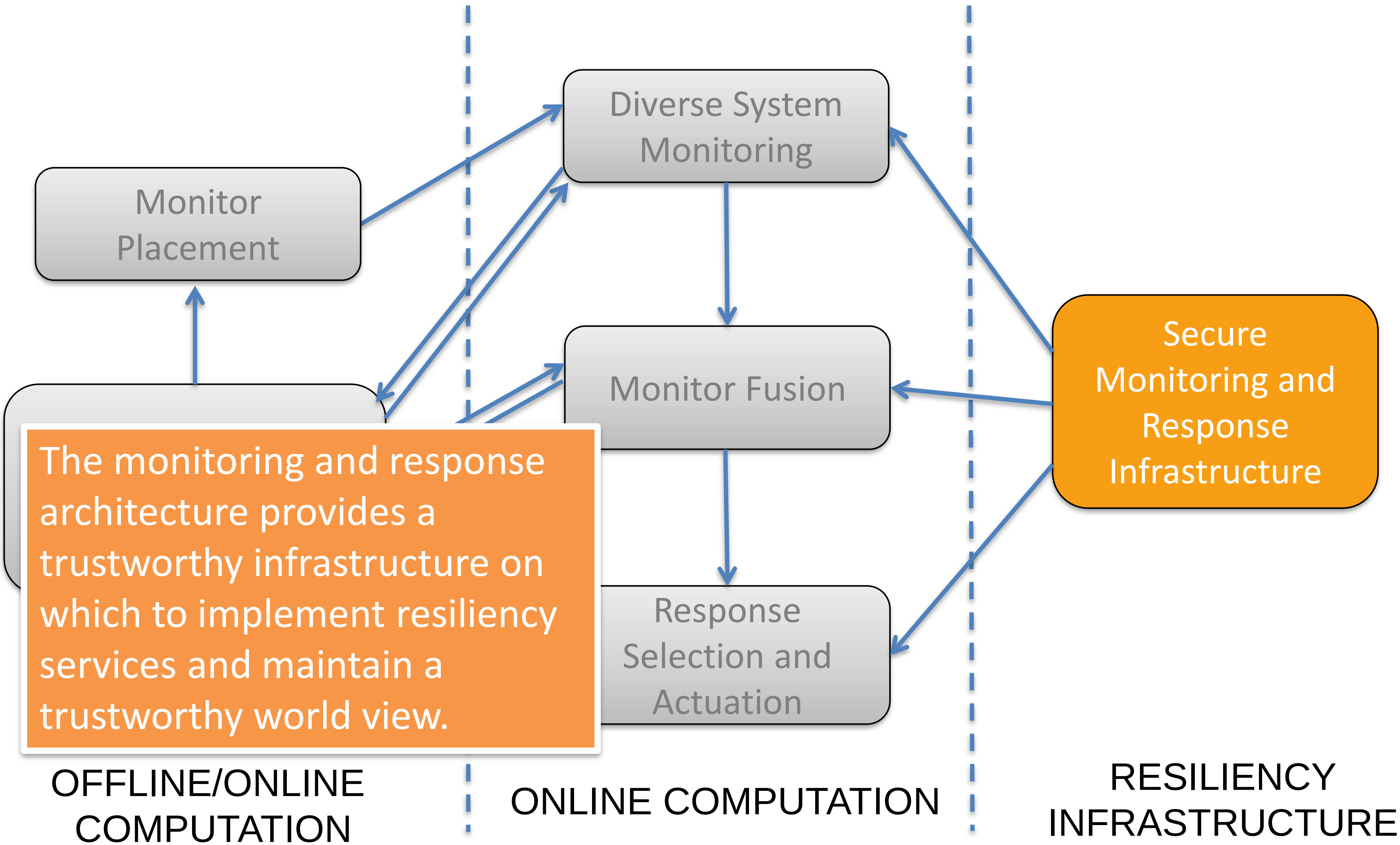
Notional Architecture for Resiliency



Notional Architecture for Resiliency



Notional Architecture for Resiliency

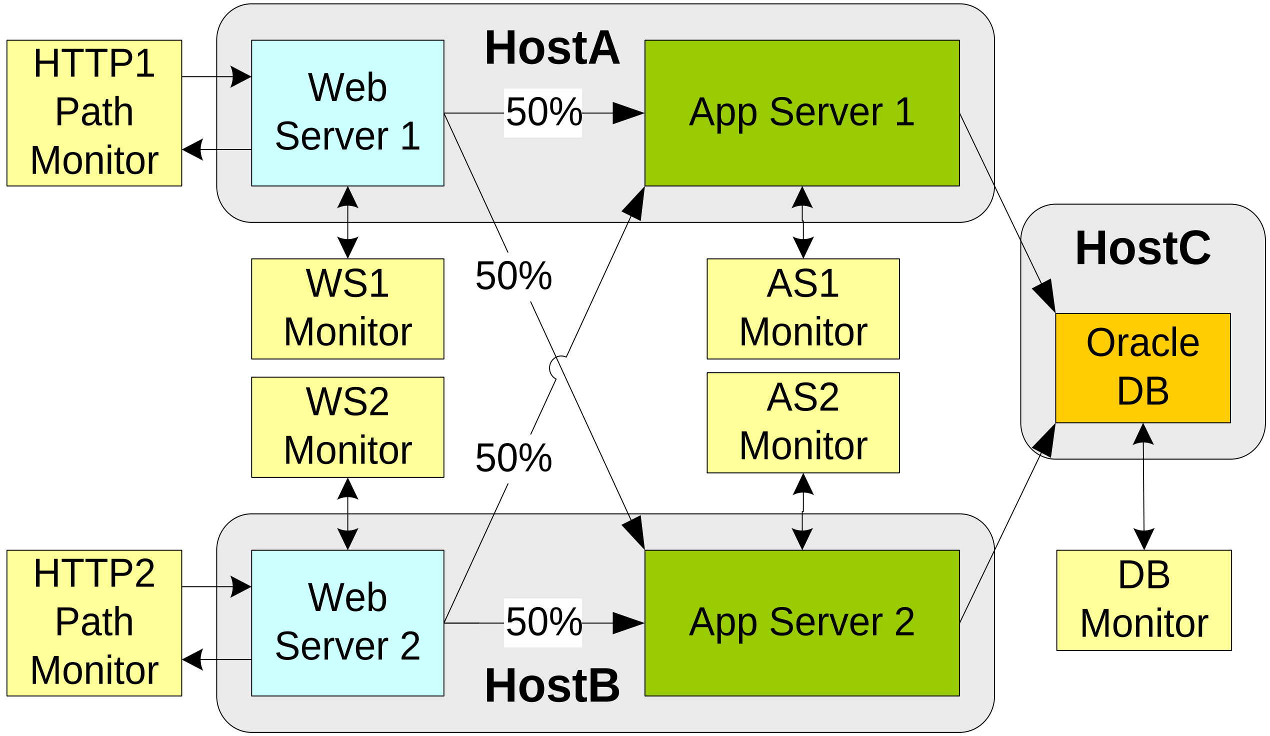


The monitoring and response architecture provides a trustworthy infrastructure on which to implement resiliency services and maintain a trustworthy world view.

Challenges in Providing Cyber Resiliency

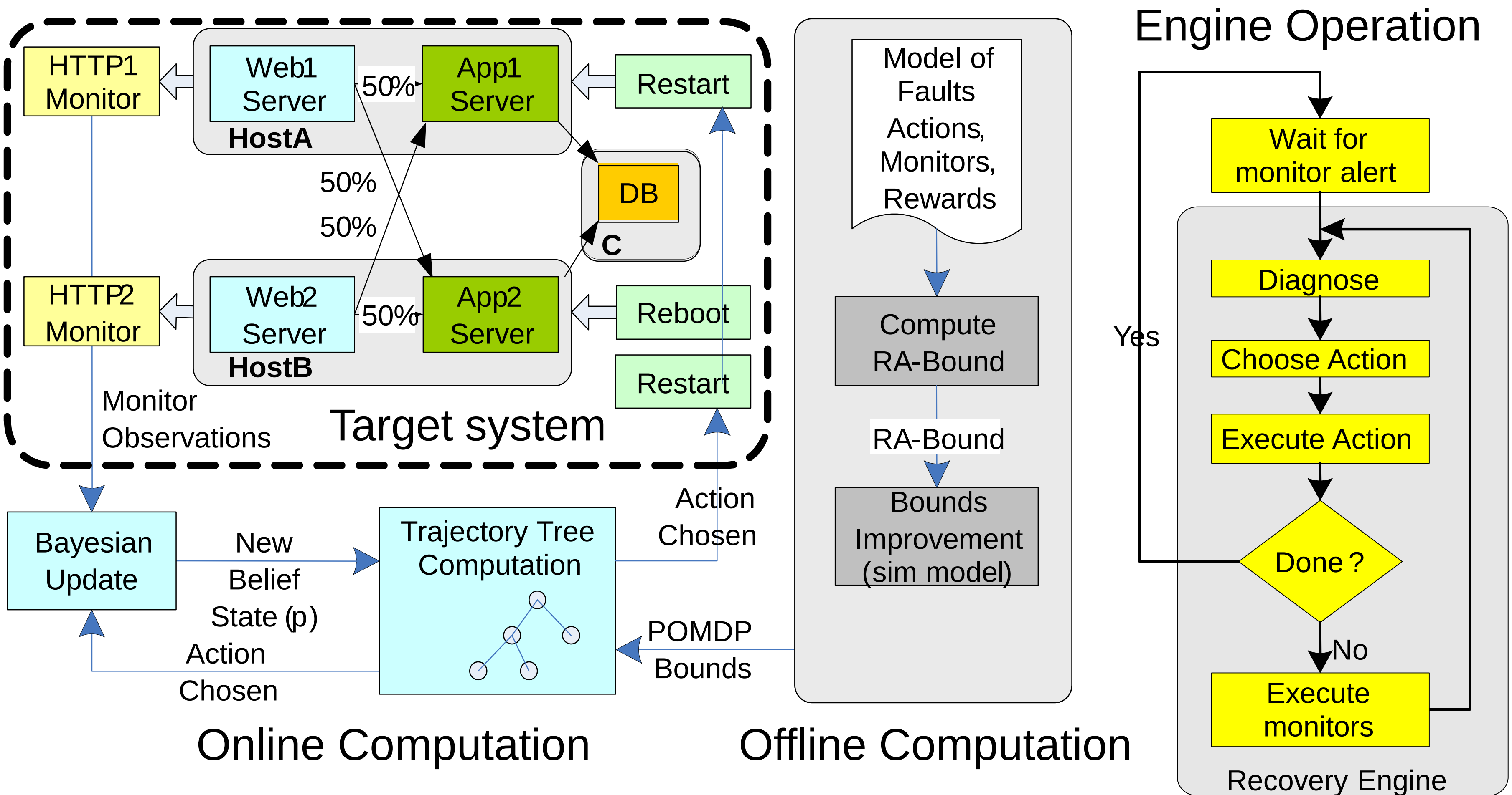
- Adaptation inherently increases the attack surface of a system
- Monitoring is increasingly possible, but creates a data deluge that makes difficult to identify relevant attack indicators
- Monitors are corruptible, which makes knowledge about the cyber state of the system only partially trustworthy
- A world model is needed to reason about indicators, but this reasoning is fallible if an attacker can work outside the model
- Catastrophic failures are (hopefully) rare, but can have a huge impact. Predictions based on historical data are ly bad at coping with rare events.

Example 1: E-commerce System with Accidental Failures (SRDS '05, DSN '06, IEEE Trans Dep/Sec '11 with AT&T Research)



- Fault models: fail-silent (crash), non fail-silent (zombie) faults
- Recovery Actions: restart component, reboot host.
- Individual component monitors: only detect crashes
- End-to-end path monitors: detect crashes and zombies but poor localization
- Recovery Cost: fraction of "lost" requests (i.e. user-perceived availability)

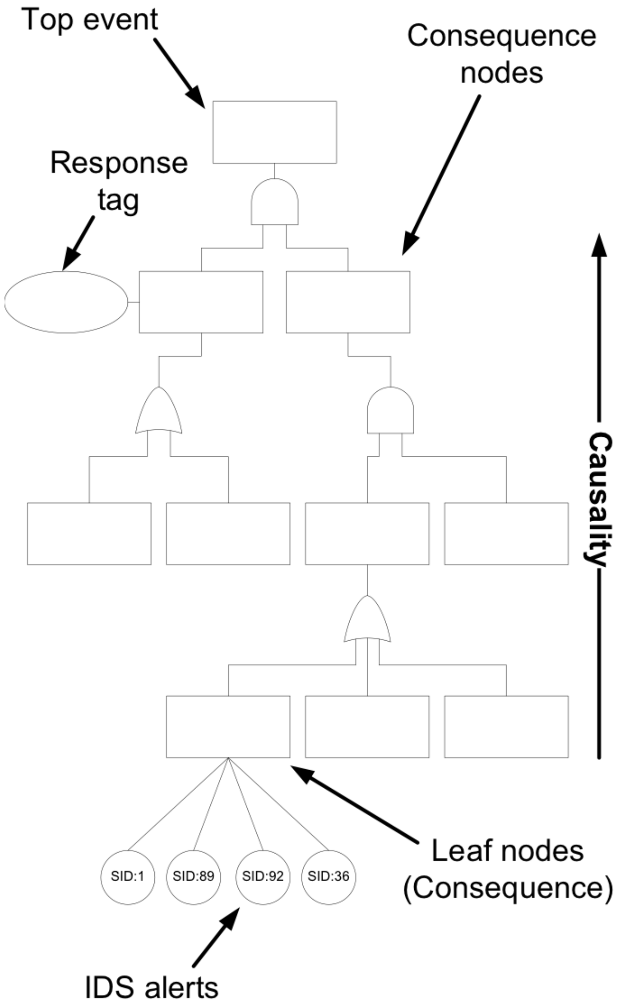
Recovery Engine Architecture



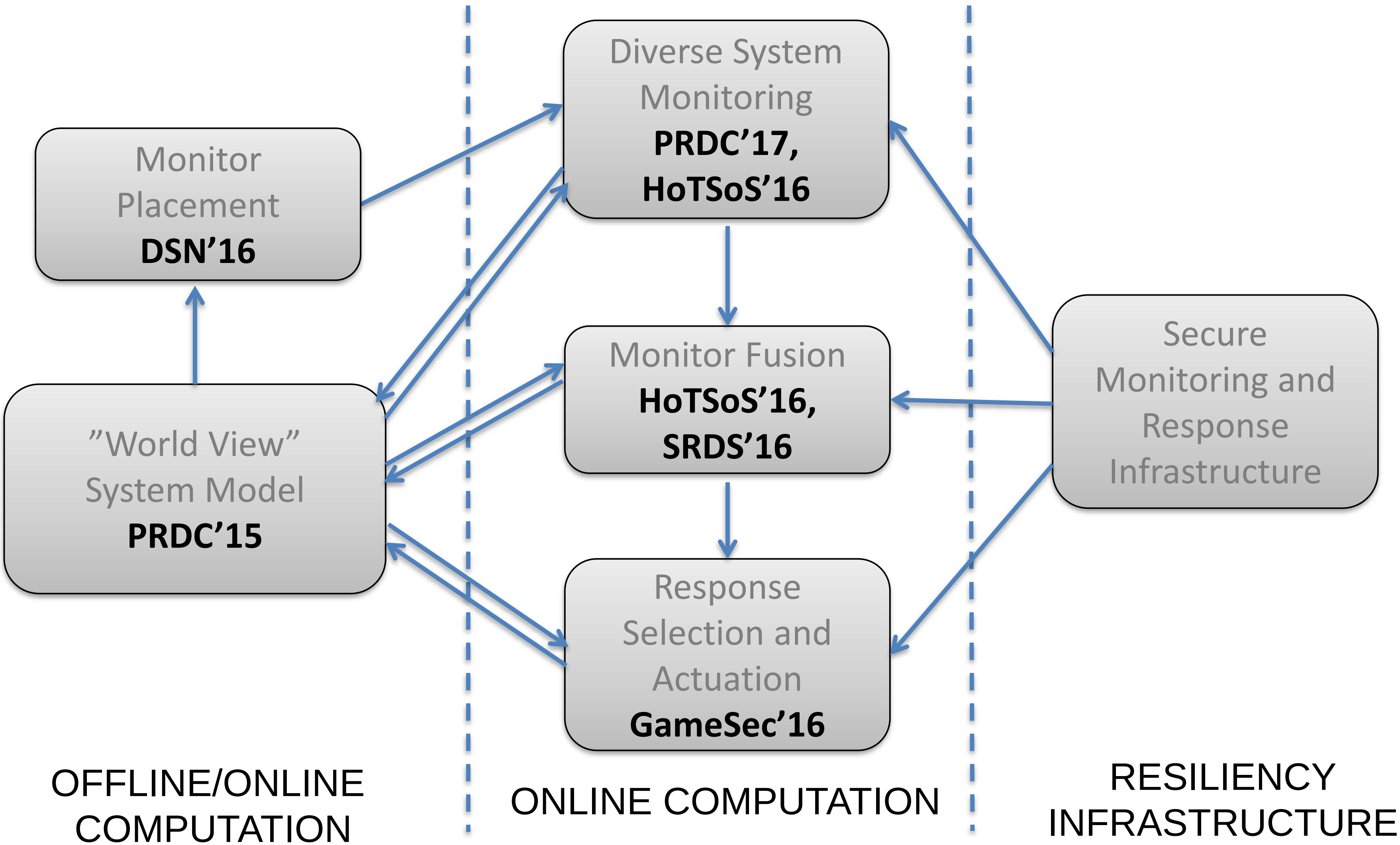
- Action that maximizes value function tree is chosen at each step
- What to use for remaining cost at the leaves of the tree?
  - Zero cost, heuristic cost, bound?

Example 2: Recovery and Response Approach for Malicious Attacks (DSN'09, IEEE Trans. Par. & Dist. Sys 2014)

- **RRE: a real-time automatic, scalable, adaptive and cost-sensitive intrusion response system**
  - Accounts for planned adversarial behavior
  - Accounts for uncertainties in IDS alerts
- Models adversary behavior and responses using **Attack-Response Tree** (ART)
- Employs a game-theoretic response strategy against adversaries in a two-player Stackelberg game
- Developed distributed and hierarchical prototype implementation



Current Work Guided is by Notional Architecture



World View Model Construction using CPTL (PRDC'15)

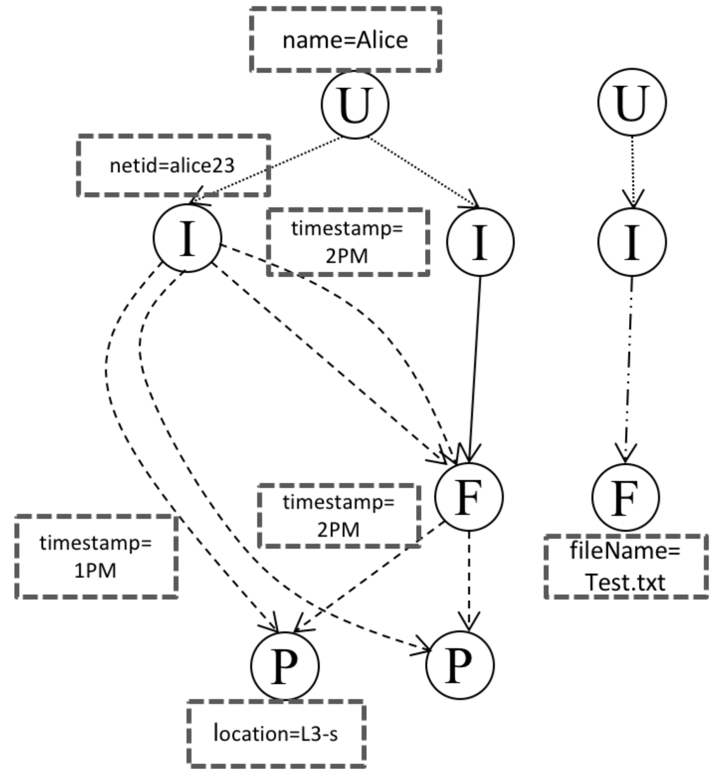
"World View" System Model

- Situational awareness is needed for resiliency
- CPTL models cyber, physical, and human system aspects
- Is a system state model that is:
  - represents heterogeneous types of data and the relations among them,
  - is updated at runtime
- CPTL can be used to:
  - exchange data among resiliency providing mechanisms
  - calculate metrics on system state

Concept Name	Icon
User	U
Identity	I
File	F
Printer	P

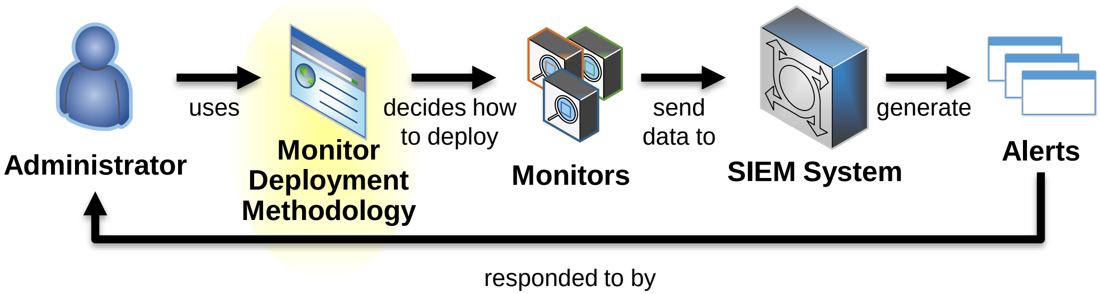
Role Name	Icon
create	- · - ->
write	⟶
prints	- - - ->
hasIdentity	·········>

Feature Name	Icon
timestamp, fileName, location, netid, name	Feature name=feature value

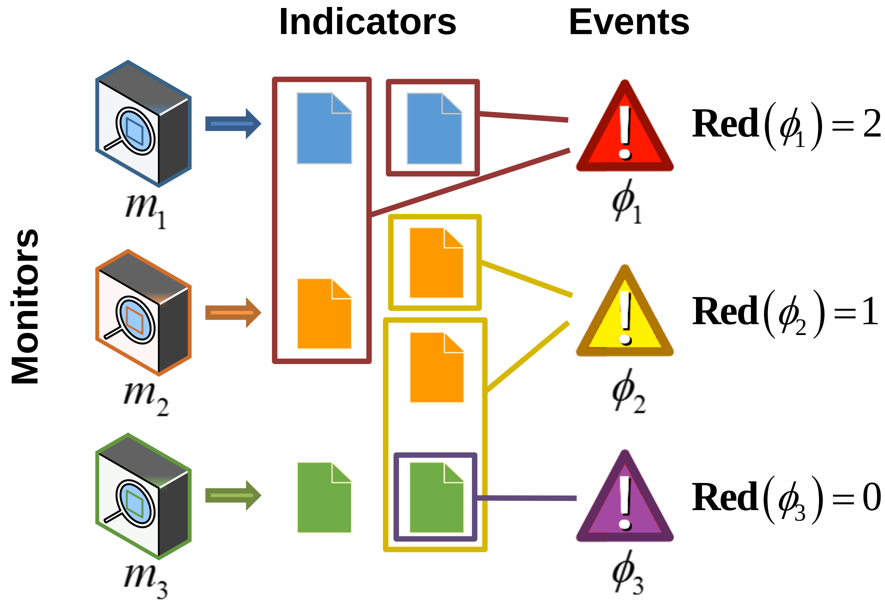


Monitor Placement Methodology (DSN'16)

Monitor Placement



- Methodology for monitor deployment to **meet intrusion detection goals** and **minimize monitoring cost**
  - Uses *quantitative metrics* to capture monitor utility and cost
  - Uses *integer programming* to determine optimal monitor deployment based on intrusion detection goals and cost requirements



$$\mathbf{Red}(\phi, M_d) = \sum_{\sigma \in \varsigma(\phi, M_d)} \min_{\iota \in \sigma} \left| \{ m \mid m \in M_d,\ \iota \in \alpha(m) \} \right|$$

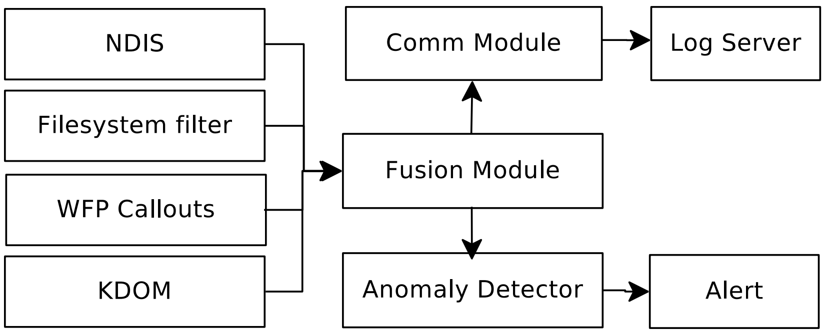
Diverse System Monitoring (PRDC'17)

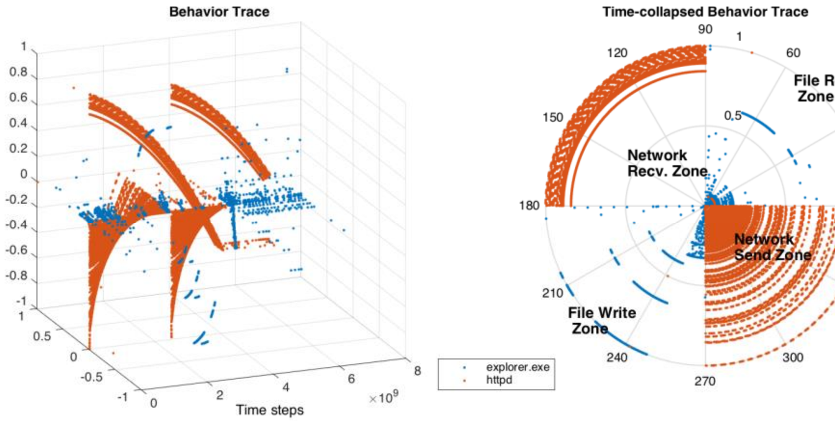
Diverse System Monitoring

- Kobra is a kernel level monitor
  - Collects process behavior traces using kernel modules
  - Network operations, file operations, process communication
- The traces are fused by generating a complex-valued time signal
- The normal behavior profile is generated by learning a space using sparse representation dictionary learning

$$\mathbf{D}^* = \arg\min_{\mathbf{D}} \sum_{i=1}^{N} \min \{ ||\mathbf{D}x_i - y_i||^2 + \lambda ||x_i||_1 \}$$

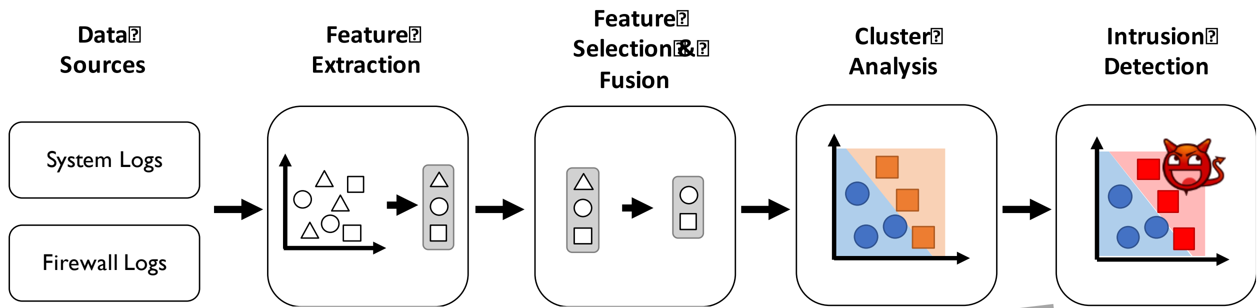
- Anomaly detection uses the learned profile to detect actions that lie outside the space of known actions

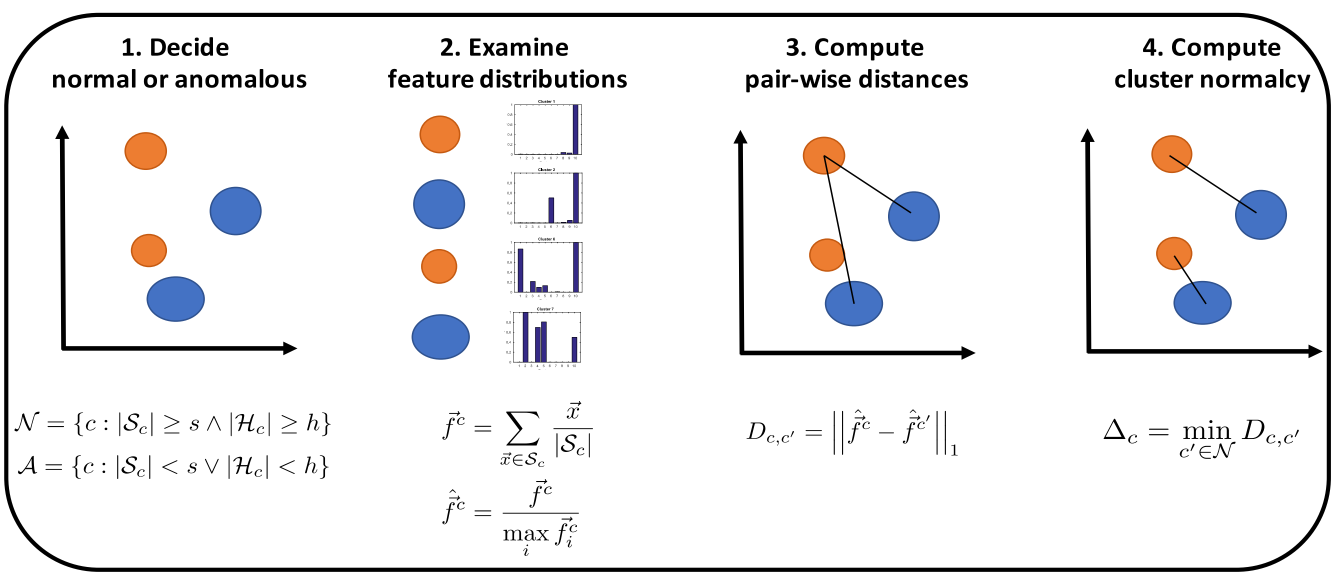




Monitor Fusion Algorithms (HoTSoS'16)

- Combine host-level authentication logs and network-level firewall logs
- Perform unsupervised cluster analysis
- Able to detect more intrusions than otherwise detected by each of the monitors individually
- Provide concise representation as a prioritized list of clusters

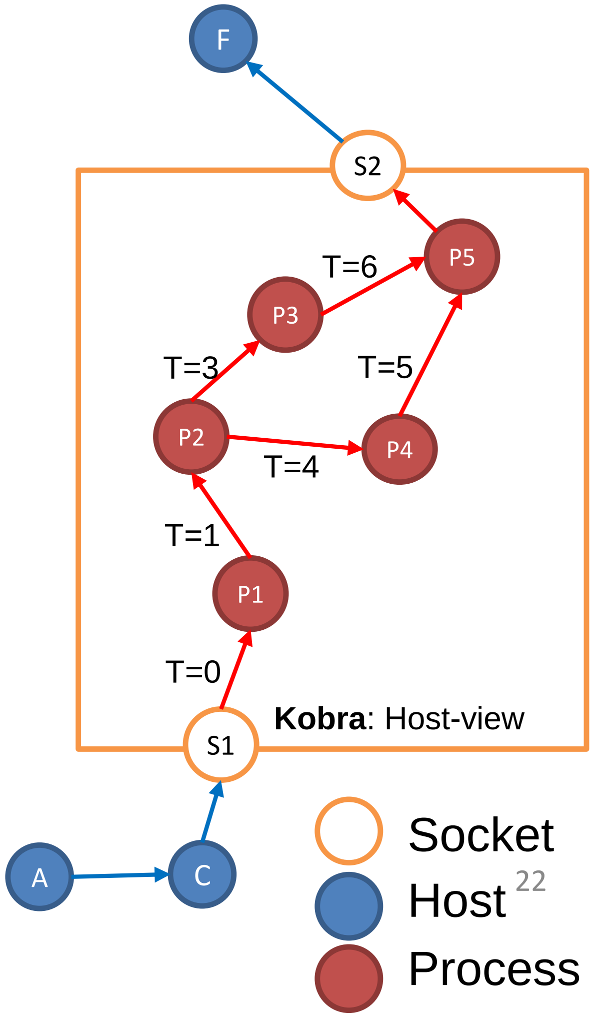




Monitor Fusion Algorithms (SRDS'16)

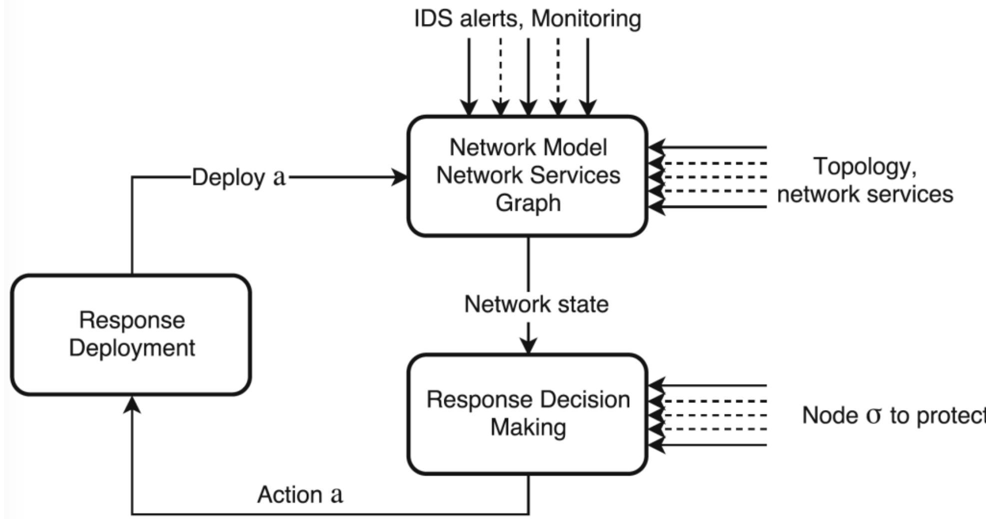
Monitor Fusion

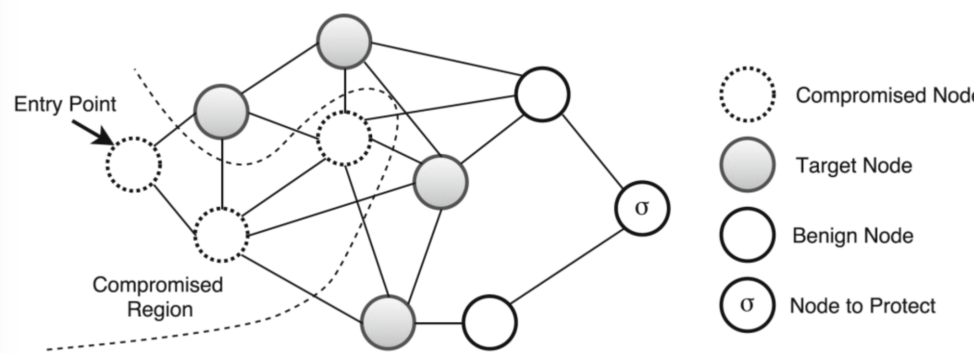
- Lateral Movement detection by fusing host-level process communication with network flow information
- Process communication used to infer network flow causation
  - *Kobra* collects the communication events and builds a process communication graph
  - Avoids the use of heuristics or signatures
- Hierarchical fusion of events results in a causation chain that describes lateral movement in the system
- Local inference of causation events allows for fusion without the need for a global clock



Response Selection and Actuation (GameSec'16)

- Goal is to design an autonomous incident response engine
  - Uses game theory for decision making
  - Uses real data-sets (when available)
  - Can scale to large systems
- Account for the effects of response actions
- Account for the system evolution
- Account for the defender's observations and actions
- Make online decisions
- Hierarchical design for scalability





Engineering in Resiliency: Assessment

Trust Assessment Challenge

- Systems operate in adversarial environments
  - Adversaries seek to degrade system operation by affecting the confidentiality, integrity, and/or availability of the system information and services
  - "Secure" systems must be able to meet their operational objectives despite attack attempts by adversaries
- System security is not absolute
  - No real system is perfectly secure
  - Some systems are more secure than others
  - *But how much more secure are they?*

Quantifying Resiliency

- At design time
  - System architects make trade-off decisions to best meet all design criteria
  - Other design criteria can be quantified: performance, reliability, operating and maintenance costs, etc.
  - *How can we quantify the security of different system designs?*
- During system operation and maintenance
  - Modifying the system architecture can improve or worsen system security
  - *How can we compare the security of different possible system configurations?*

Model-based system-level resiliency evaluation

Practical Applications of Security Metrics

Organizational-level Metrics

Questions the CIO cannot answer:

- How much risk am I carrying?
- Am I better off now than I was this time last year?
- Am I spending the right amount of money on the right things?
- How do I compare to my peers?
- What risk transfer options do I have?

(From CRA, Four Grand Challenges in Trustworthy Computing, 2003)

Technical Metrics

Questions the design engineer cannot answer:

- Is design A or B more secure (confidentially, integrity, availability, privacy)?
- Have I made the appropriate design trade off between timeliness, security, and cost?
- How will the system, as implemented, respond to a specific attack scenario?
- What is the most critical part of the system to test, from a security point of view?

A Question neither can answer:

- How do the technical metrics impact the organizational-level security metrics?

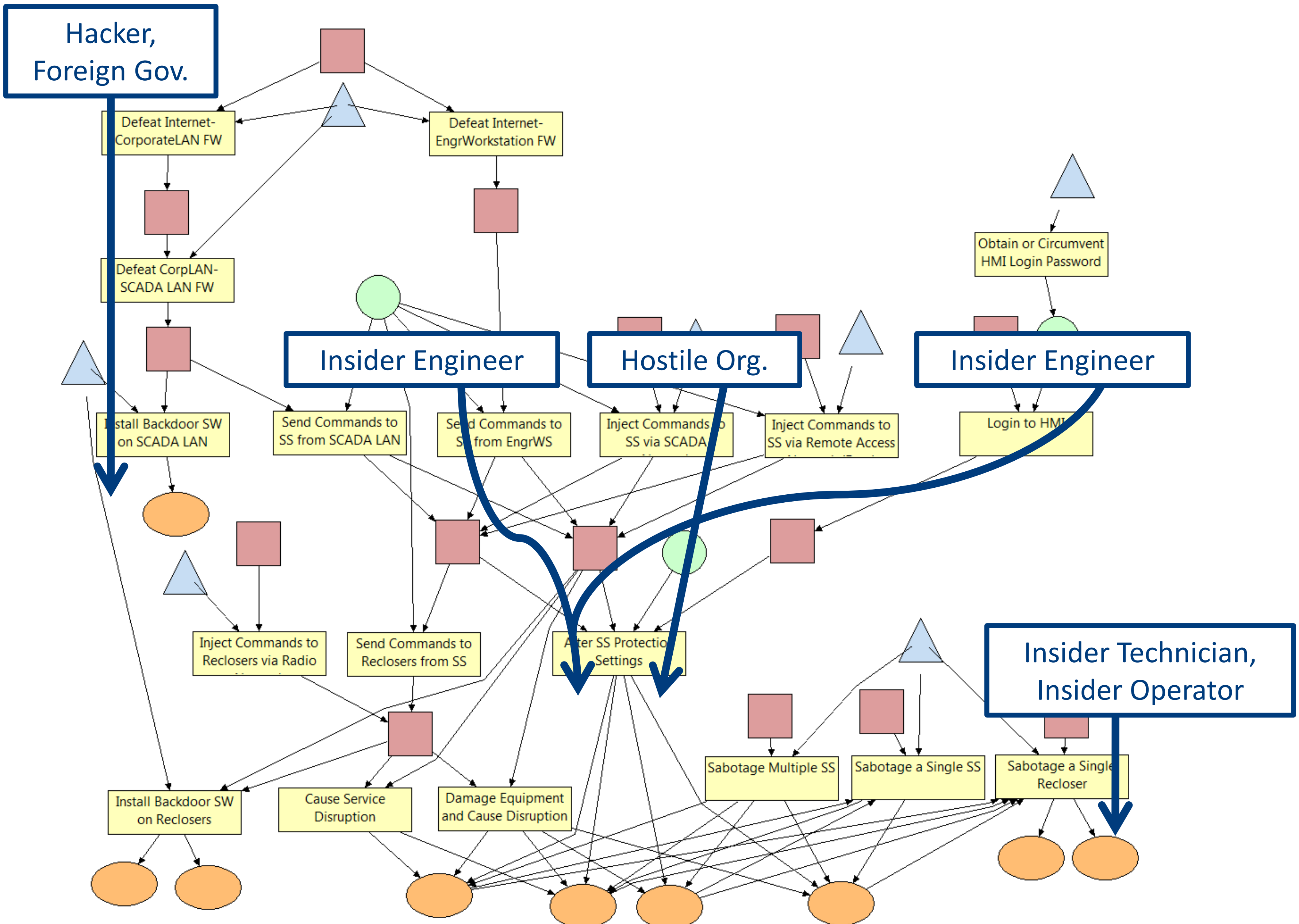
Contrasting Approaches

Typical Situation Today:

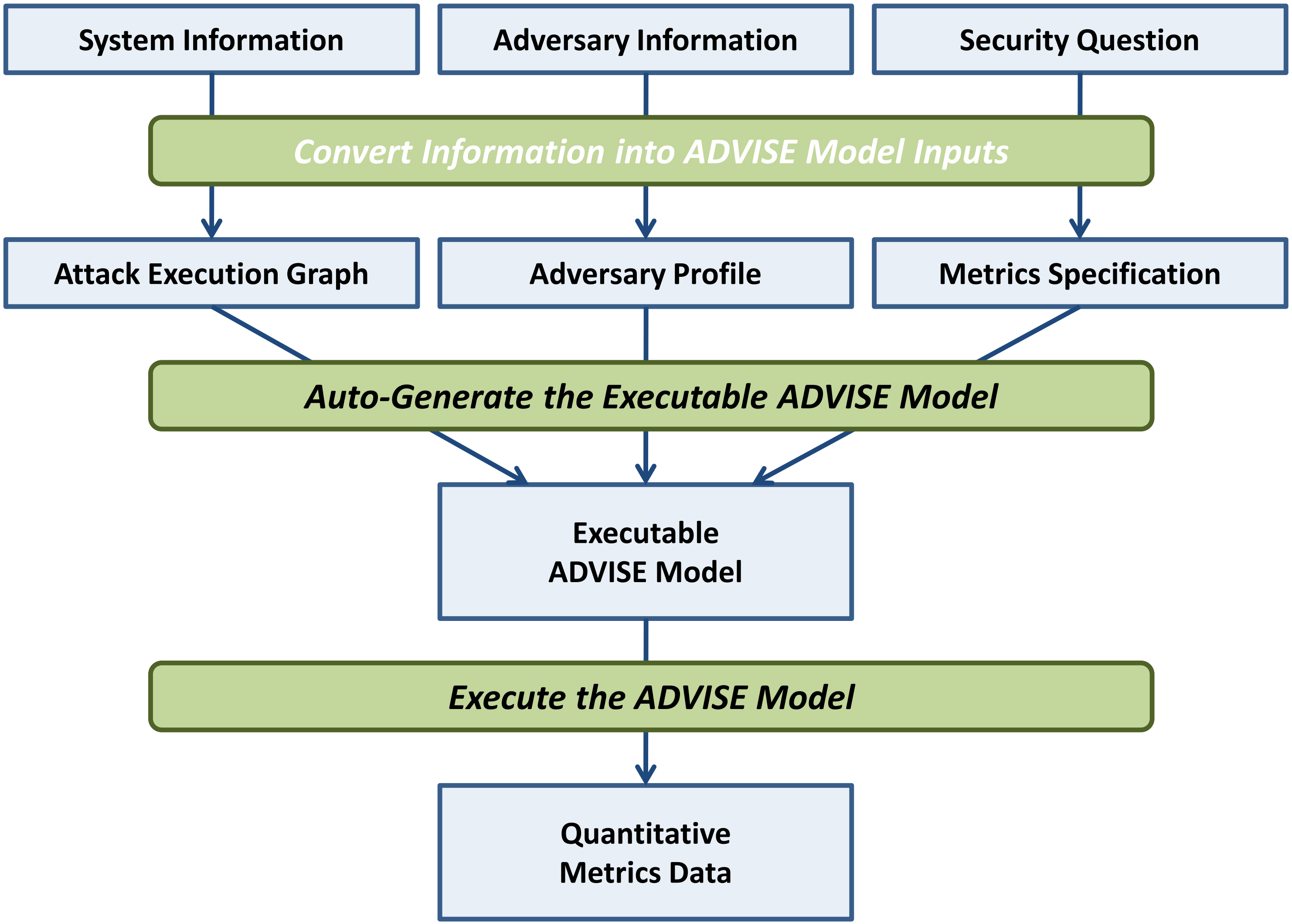
- Process:
  - Rely on a trusted analyst (wizard?) that examines situation, and gives advice based on experience, or
  - Form decision in a collective manner based on informal discussions among stakeholder experts
- *Limitations:*
  - *No* way to audit decision process
  - *No* quantifiable ranking of alternative options

Goal For Tomorrow:

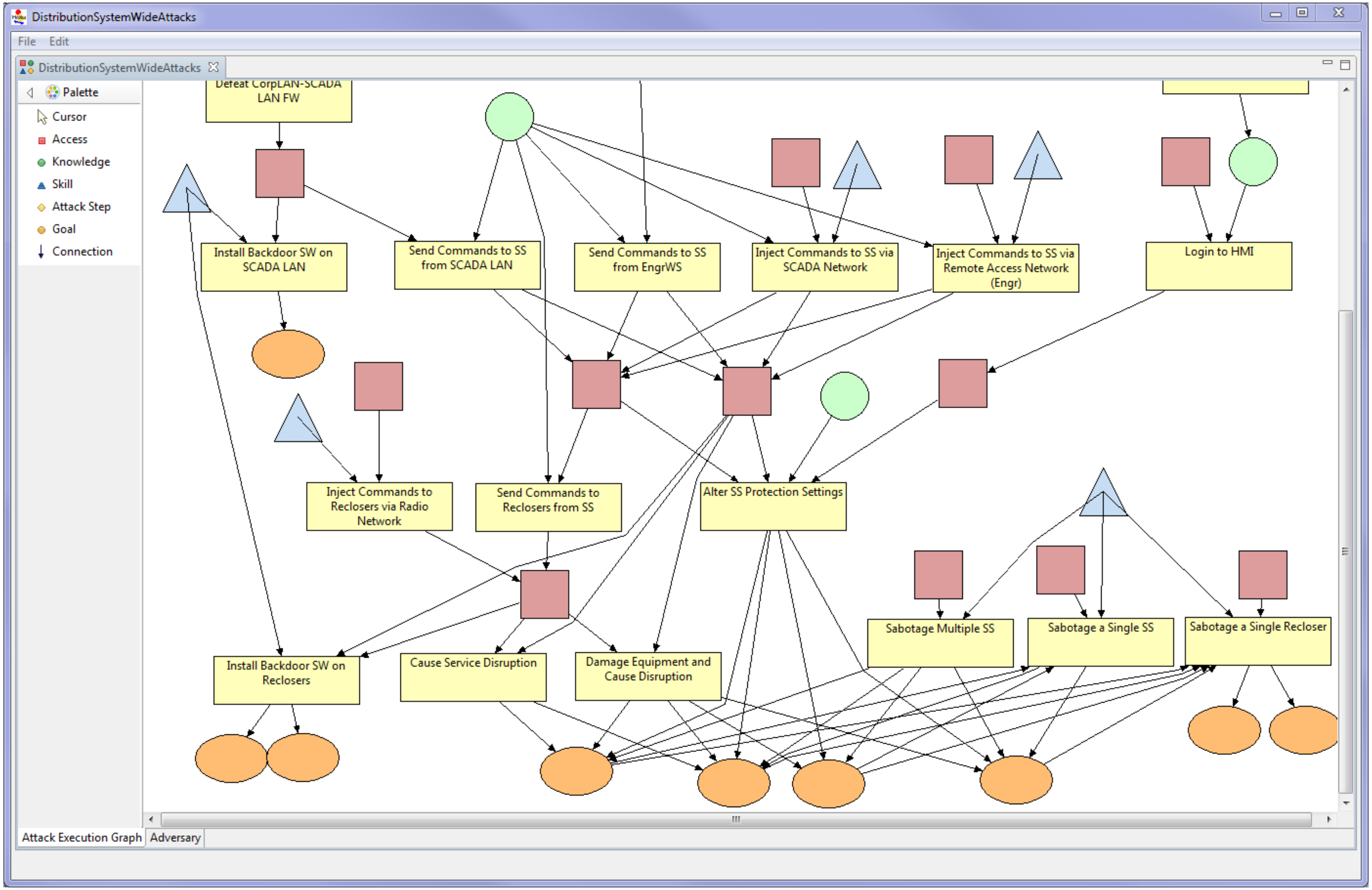
- Usable tool set that enables diverse stakeholders to express
  - Multi-faceted aspects of model
  - Multiple objectives
- Way for diverse stake holders to express concerns and objectives in common terminology
- Quantifiable ranking of alternate security policies and architectures
- Auditable decision process


Hacker, Foreign Gov.
Defeat Internet-CorporateLAN FW
Defeat Internet-EngrWorkstation FW
Defeat CorpLAN-SCADA LAN FW
Obtain or Circumvent HMI Login Password
Insider Engineer
Hostile Org.
Insider Engineer
Install Backdoor SW on SCADA LAN
Send Commands to SS from SCADA LAN
Send Commands to SS from EngrWS
Inject Commands to SS via SCADA
Inject Commands to SS via Remote Access
Login to HMI
Inject Commands to Reclosers via Radio
Send Commands to Reclosers from SS
Alter SS Protection Settings
Insider Technician, Insider Operator
Sabotage Multiple SS
Sabotage a Single SS
Sabotage a Single Recloser
Install Backdoor SW on Reclosers
Cause Service Disruption
Damage Equipment and Cause Disruption

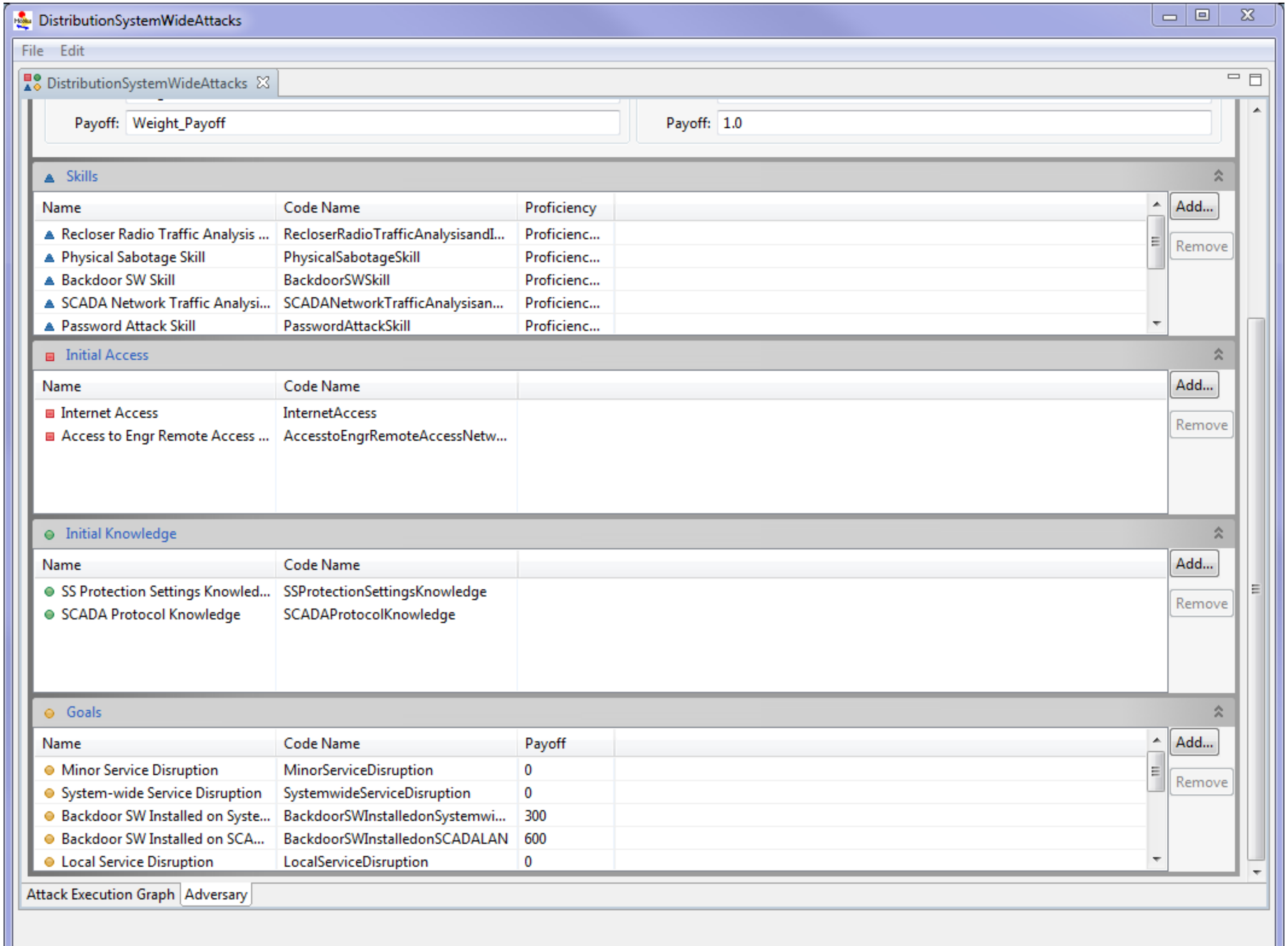
ADVISE Method Overview (DSN’10, MetriSec’10, QEST’11)



Attack Execution Graph Editor

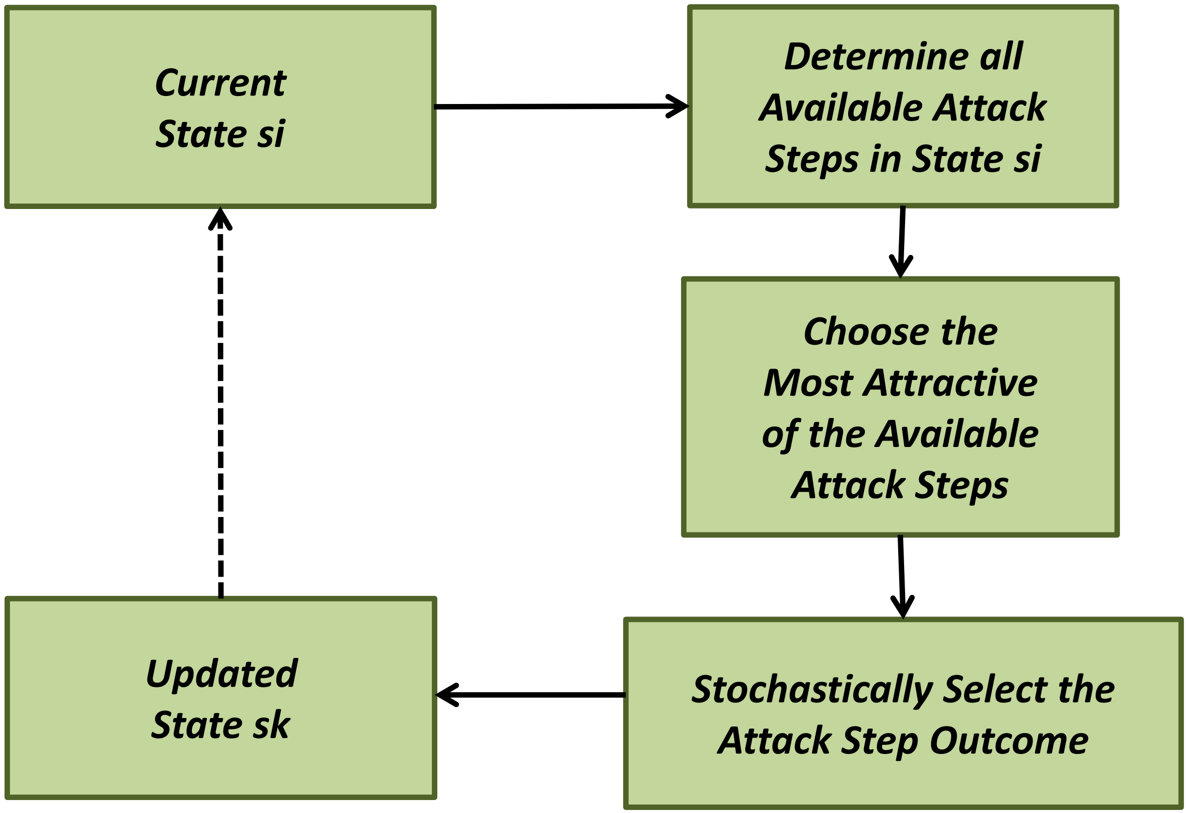


Adversary Editor

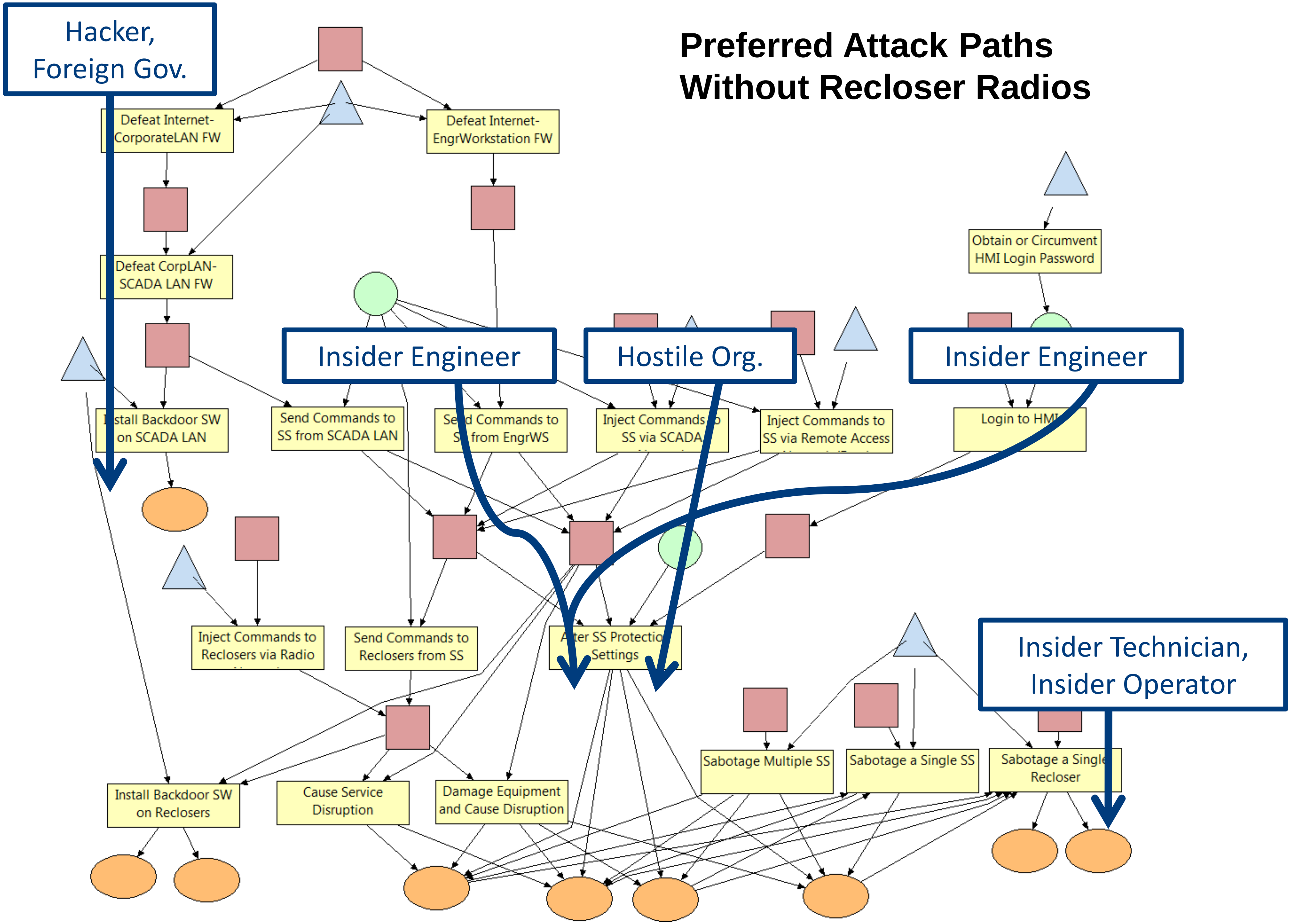


Model Execution: the Attack Decision Cycle

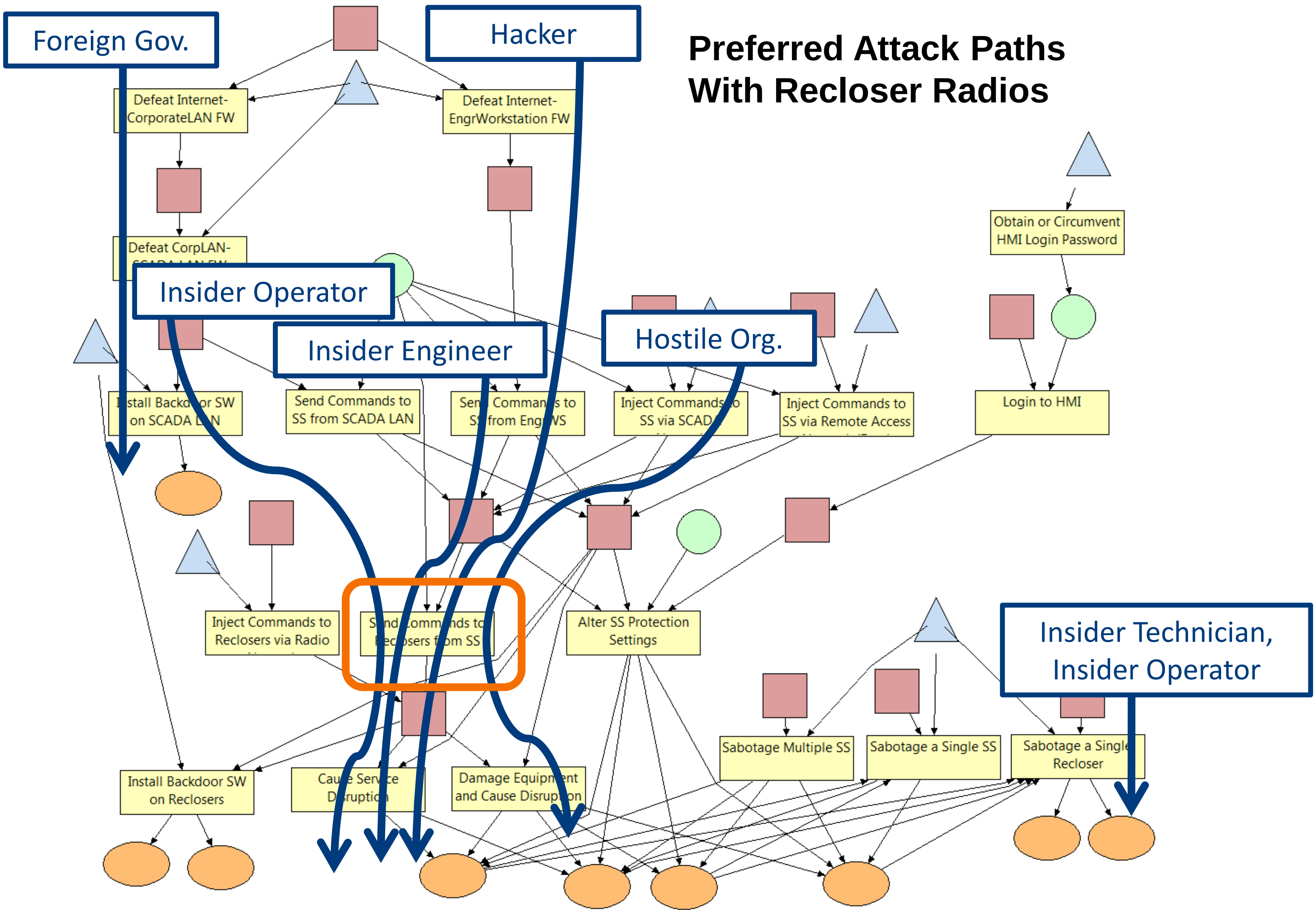
- The adversary selects the most attractive available attack step based on his attack preferences.
- State transitions are determined by the outcome of the attack step chosen by the adversary.

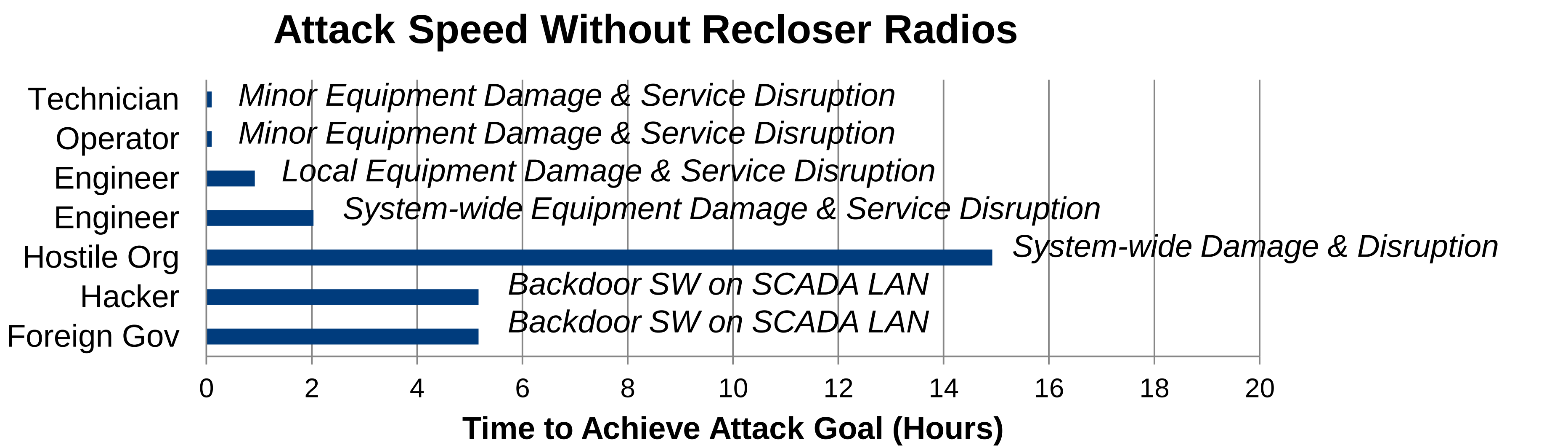


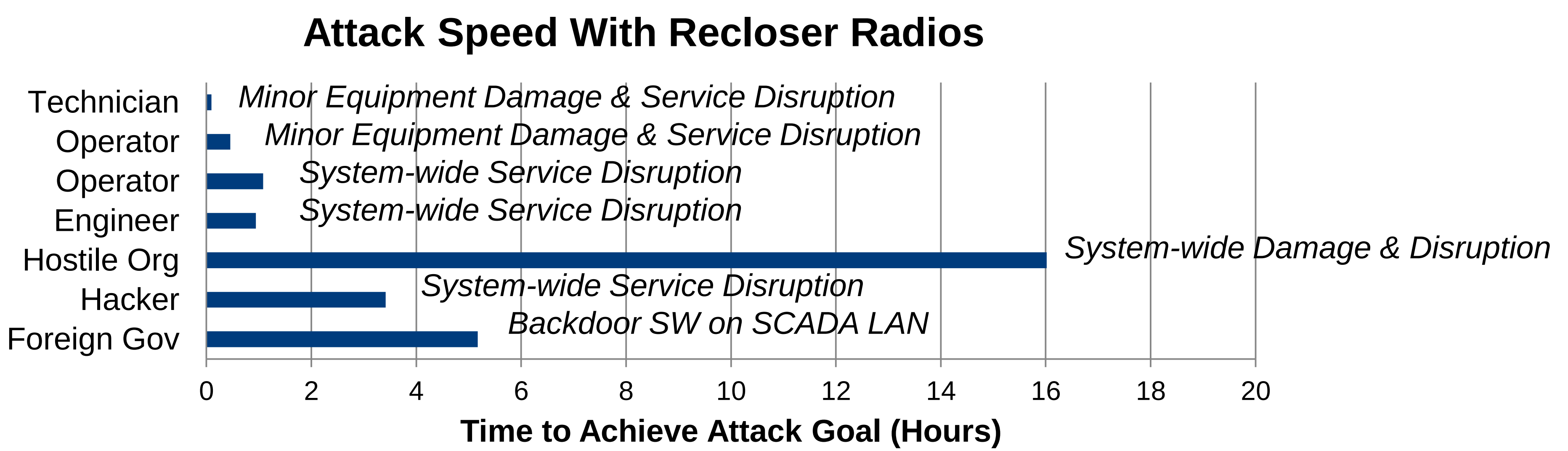
Preferred Attack Paths Without Recloser Radios



Preferred Attack Paths With Recloser Radios




Attack Speed Without Recloser Radios
Technician
Operator
Engineer
Engineer
Hostile Org
Hacker
Foreign Gov
Minor Equipment Damage & Service Disruption
Minor Equipment Damage & Service Disruption
Local Equipment Damage & Service Disruption
System-wide Equipment Damage & Service Disruption
System-wide Damage & Disruption
Backdoor SW on SCADA LAN
Backdoor SW on SCADA LAN
0
2
4
6
8
10
12
14
16
18
20
Time to Achieve Attack Goal (Hours)


Attack Speed With Recloser Radios
Technician
Operator
Operator
Engineer
Hostile Org
Hacker
Foreign Gov
Minor Equipment Damage & Service Disruption
Minor Equipment Damage & Service Disruption
System-wide Service Disruption
System-wide Service Disruption
System-wide Damage & Disruption
System-wide Service Disruption
Backdoor SW on SCADA LAN
0
2
4
6
8
10
12
14
16
18
20
Time to Achieve Attack Goal (Hours)

The Path Forward

The Path Forward

- Perfect security is science fantasy, and perfection is the enemy of good
- Resiliency mechanisms are needed to tolerate attacks, responding to provide a specified service despite partially successful attacks
- Assessment tools are needed at design time to choose between alternative resiliency mechanisms
- For the good of society, pragmatic approaches are needed to engineer resiliency into cyber systems for use in critical applications
- We're just at the beginning of the journey, and much work remains to be done