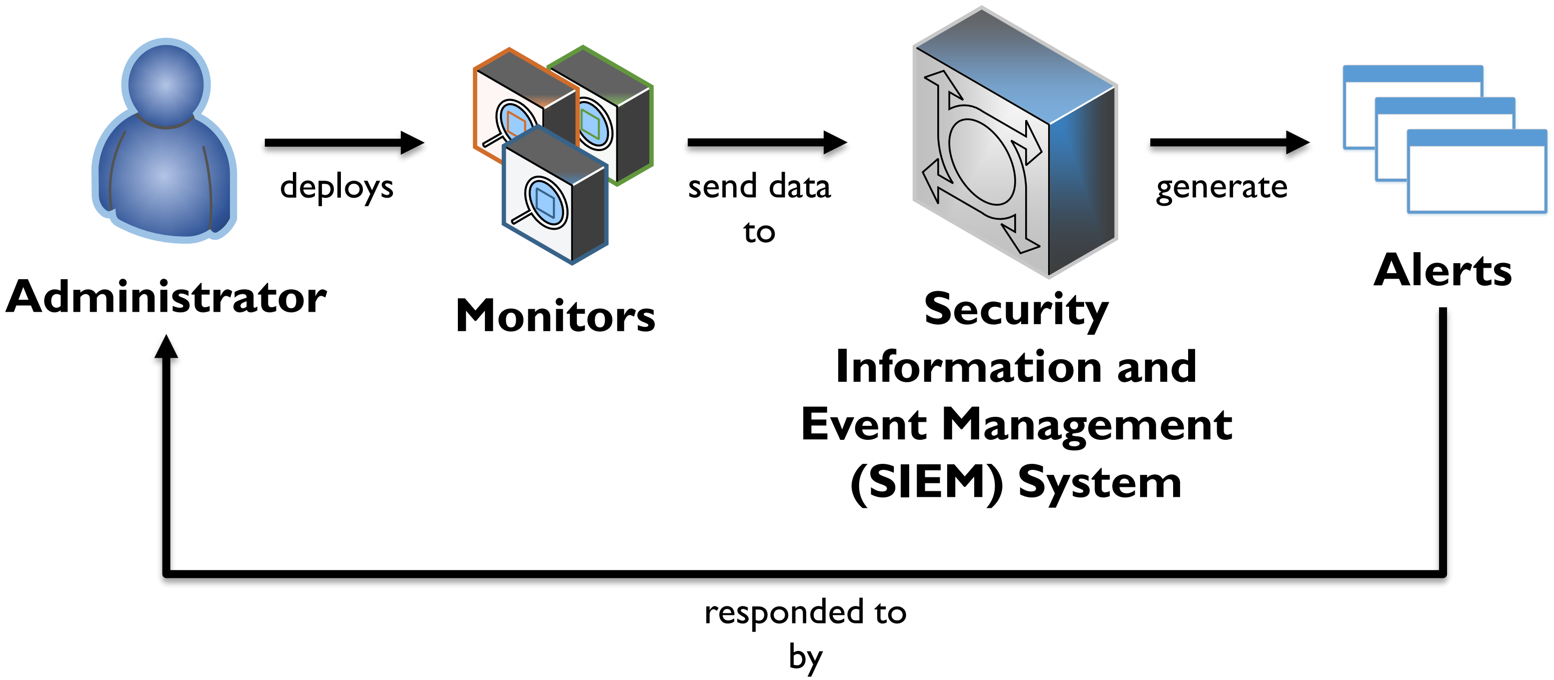
# A Quantitative Methodology for Security Monitor Deployment

Uttam Thakore, Gabriel A. Weaver, William H. Sanders
University of Illinois at Urbana-Champaign

# Security monitoring today

Administrators must decide what data to collect and how it should be analyzed



# Problem

Difficult to determine which monitors are **necessary to meet intrusion detection requirements**

Risks:

– Overprovisioned monitors – large volumes of poorly actionable logs

– Underprovisioned monitors – insufficient ability to detect or investigate security incidents

We help administrators determine exactly where they stand

– Can expose weaknesses in monitoring

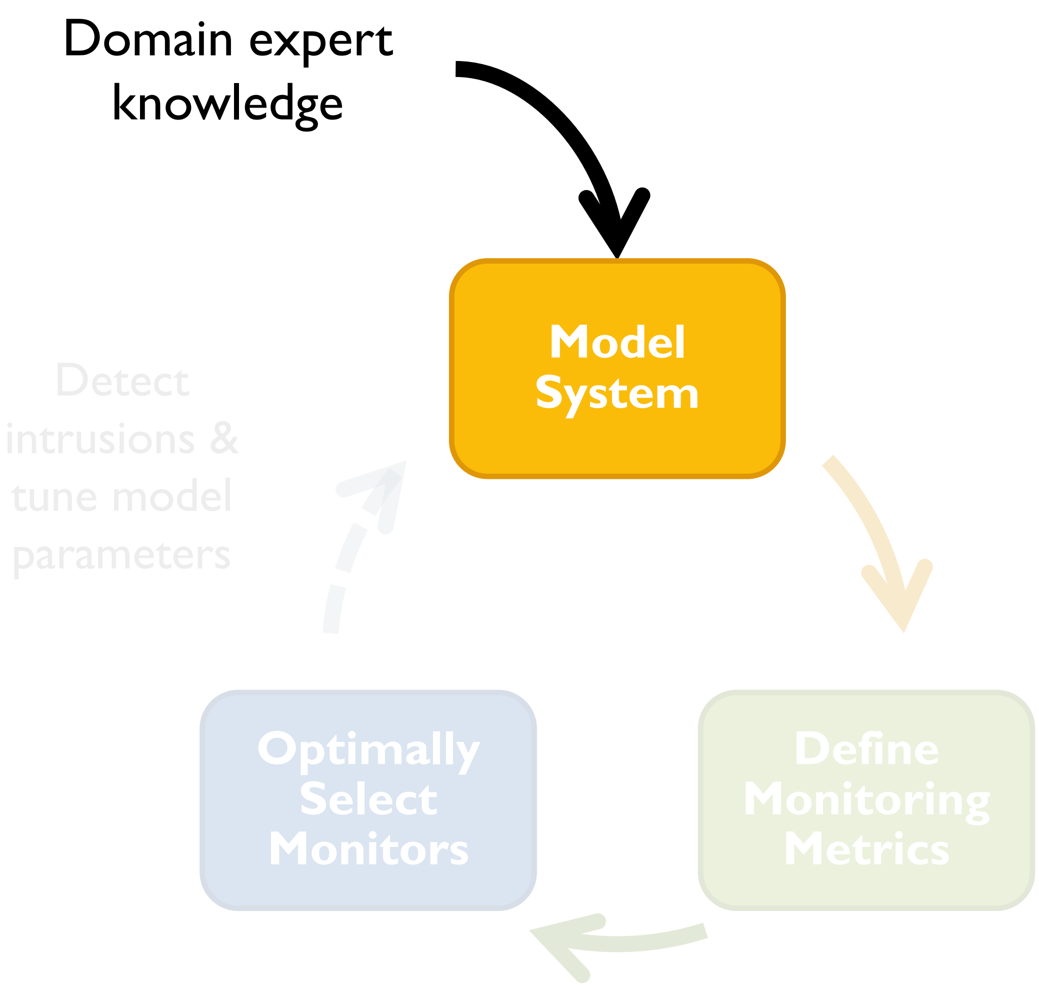
# Our contribution

We have developed a **quantitative**, **cost-sensitive** methodology for monitor selection that **meets intrusion detection requirements**

# Guiding principles

- Monitors and computing assets can be **compromised**
  - Monitor compromise can affect ability to detect intrusions
- **Redundant monitoring** can mitigate the effect of compromise or unavailability

# Outline


Domain expert knowledge
Model System
Define Monitoring Metrics
Optimally Select Monitors
Detect intrusions & tune model parameters

# Model: Data model

**Monitors**

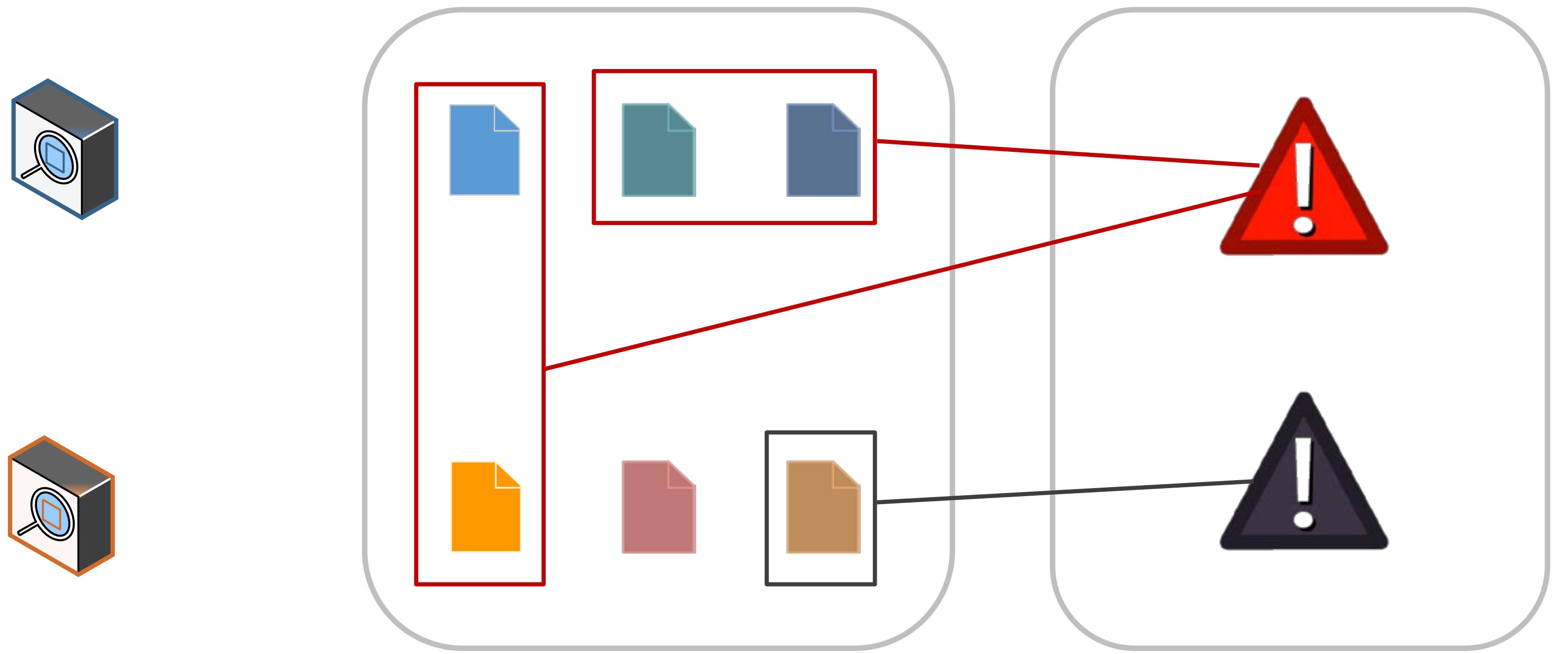
Sensors that collect information about the system

**Indicators**

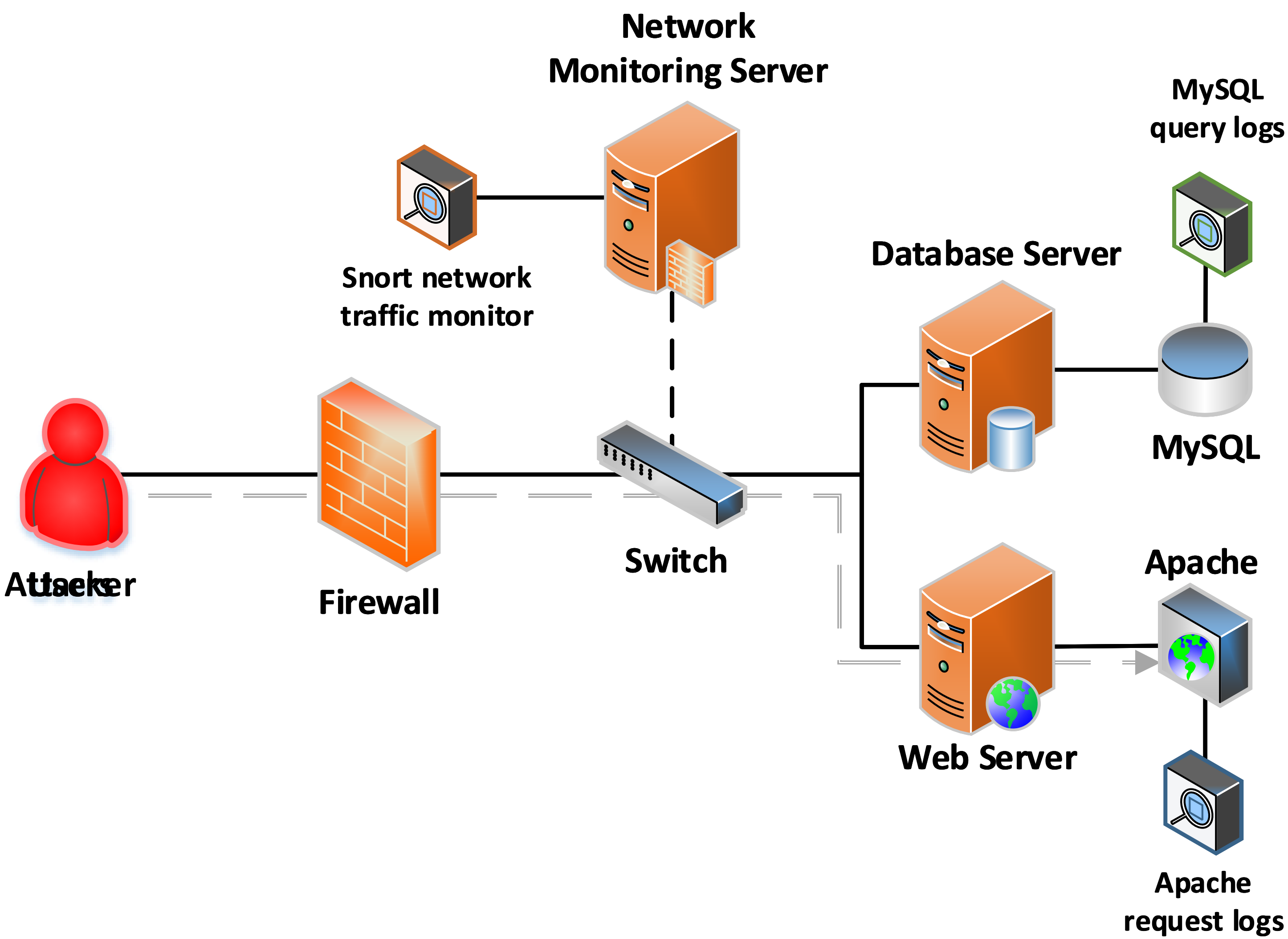
Primitives representing information provided by monitors about events

**Events**

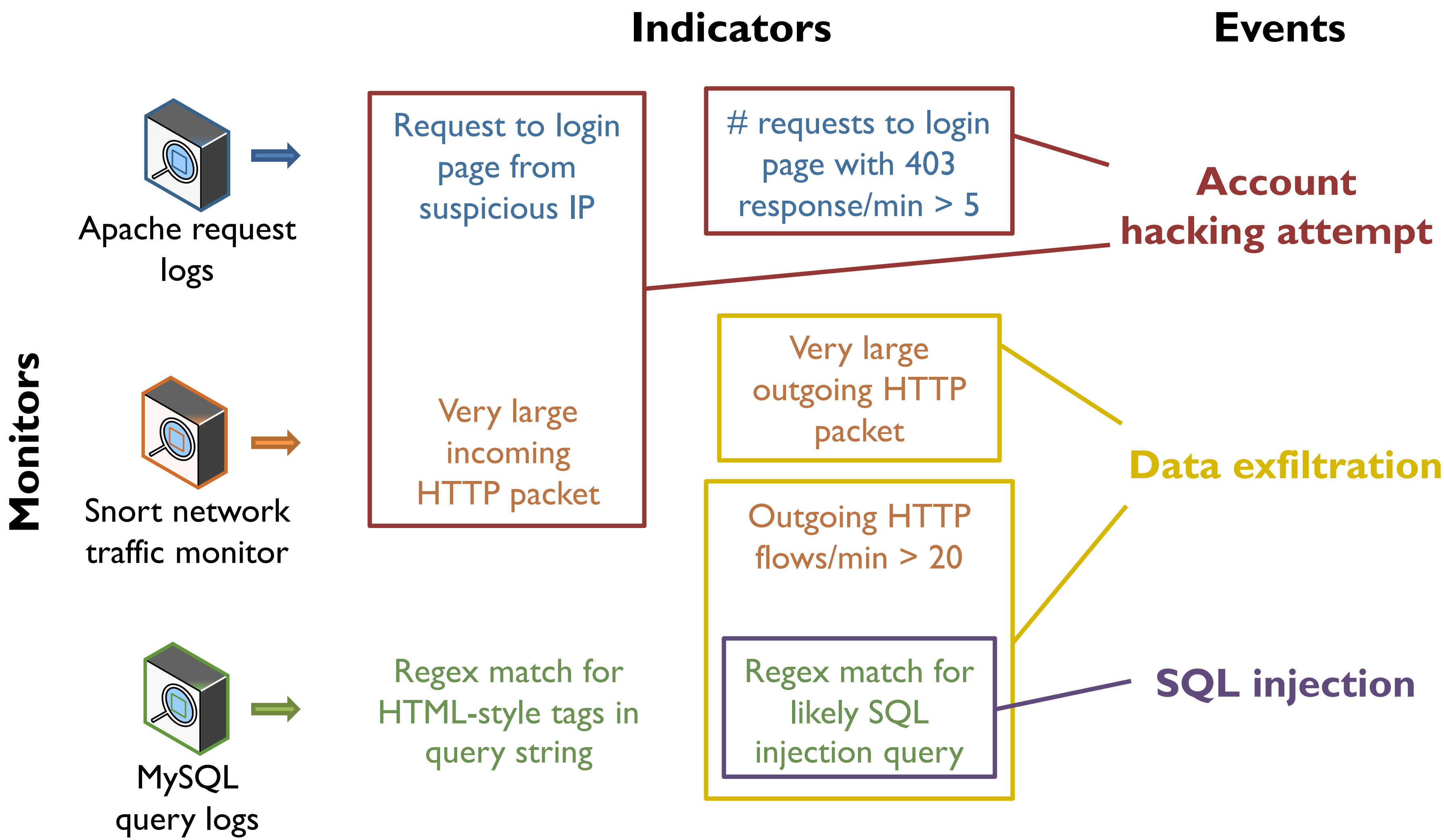
Intrusions or actions symptomatic of attacks



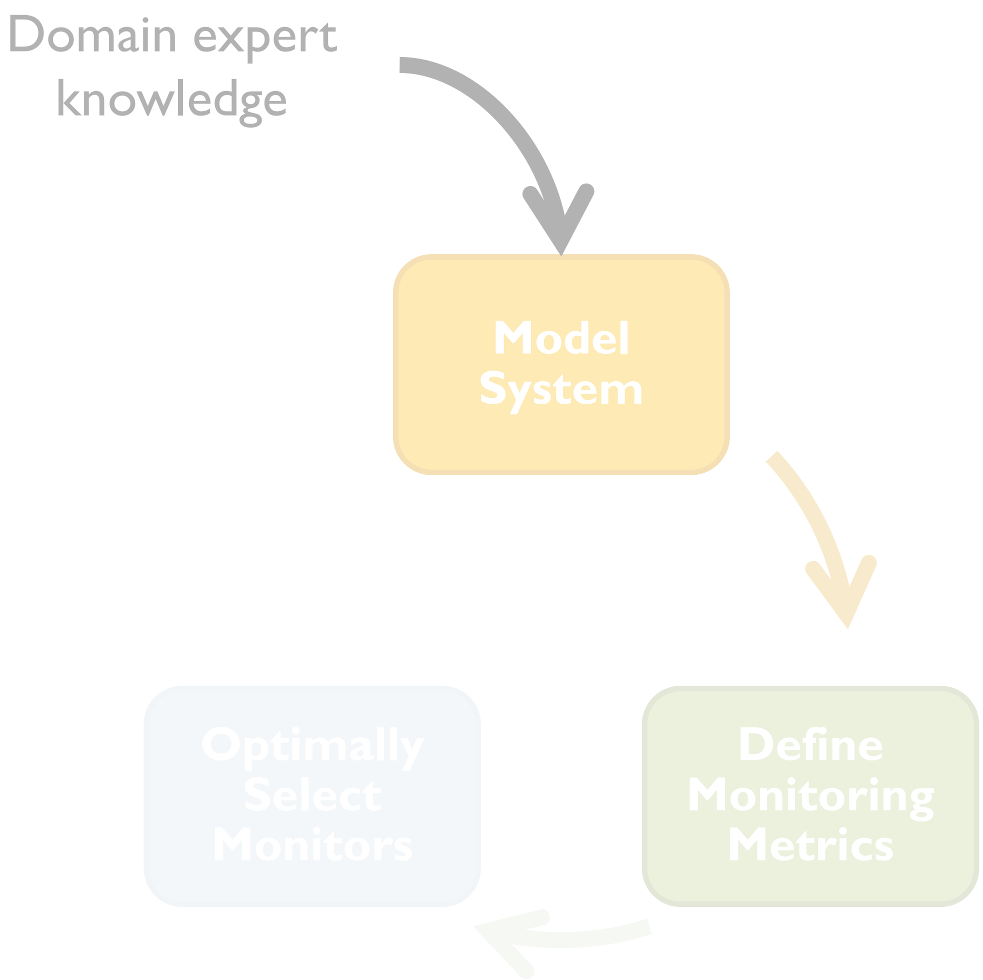
# Case study: E-commerce web service



# Model: Case study data model



# Outline

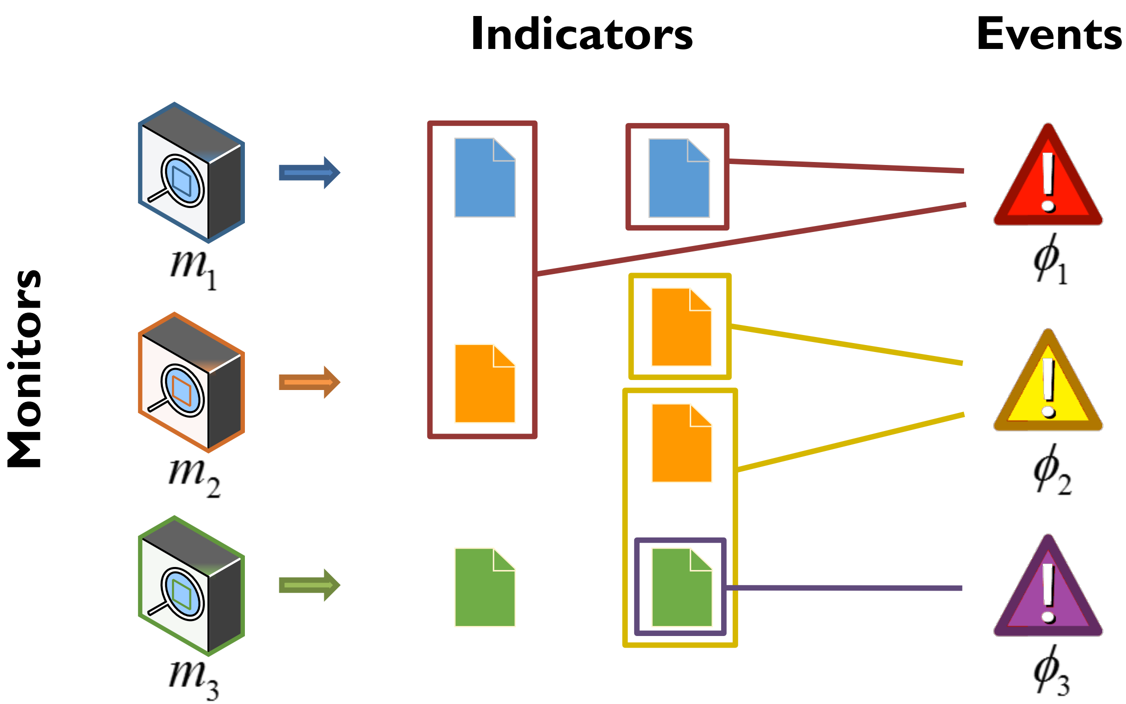

Domain expert
knowledge
Model
System
Define
Monitoring
Metrics
Optimally
Select
Monitors

# Monitoring metrics

- **Goal of metrics**: *quantify* utility and cost of monitors in supporting intrusion detection
- Three monitor utility metrics:
  - Coverage
  - Redundancy
  - Confidence
- One cost metric:
  - Monitor cost

# Metrics: Coverage

**Definition:** overall fraction of select events that are detectable given a set of monitors

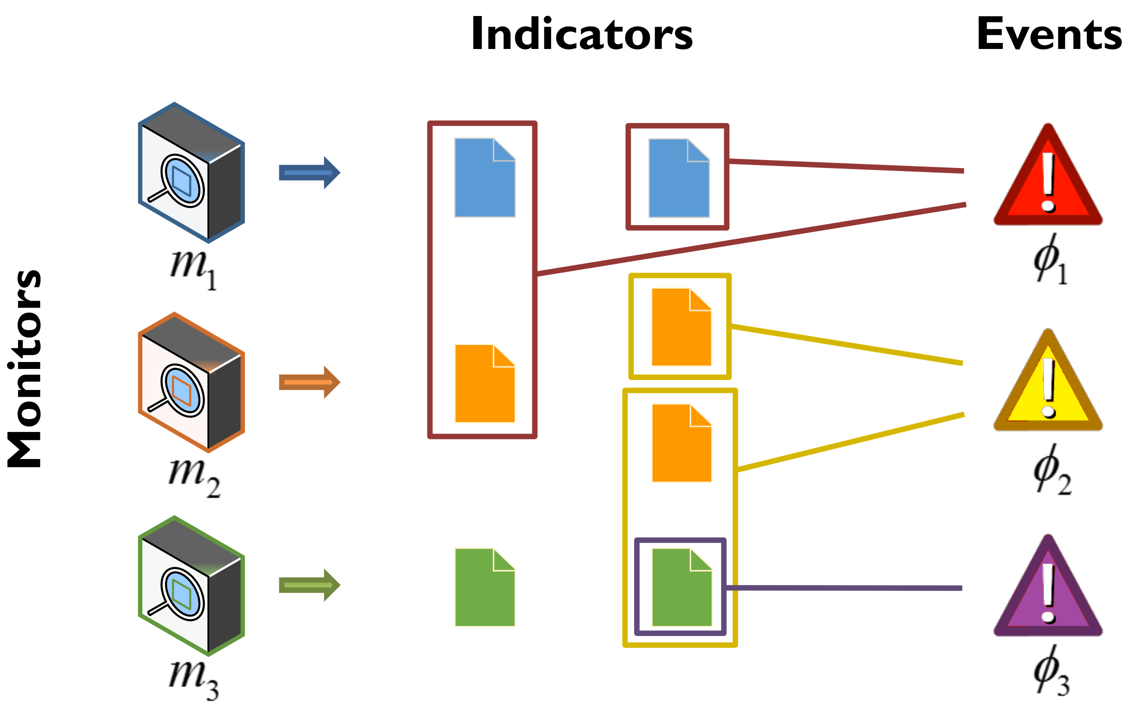



$$\mathbf{Cov}\left(\{\phi_1,\ \phi_2,\ \phi_3\},\ \{m_2, m_3\}\right) = 67\%$$

$$\mathbf{Cov}\left(\Phi, M_d\right) = \frac{\left|\left\{\phi \mid \delta\left(\phi, M_d\right) \wedge \phi \in \Phi\right\}\right|}{|\Phi|}$$

# Metrics: Redundancy

**Definition:** the number of ways an event can be detected given a set of monitors




$$\mathbf{Red}\left(\phi_1, \{m_1, m_2\}\right) = 2$$

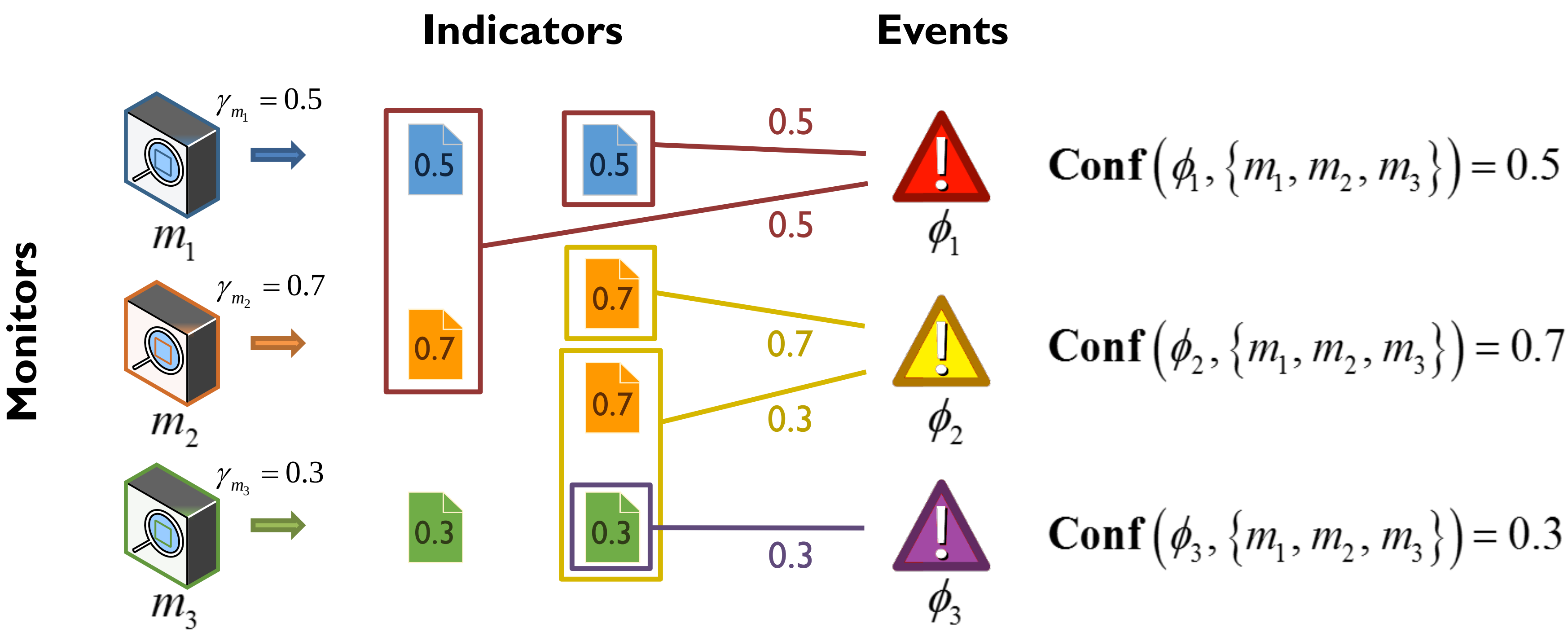
$$\mathbf{Red}\left(\phi_2, \{m_1, m_2\}\right) = 1$$

$$\mathbf{Red}\left(\phi_3, \{m_1, m_2\}\right) = 0$$

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

# Metrics: Confidence

**Definition:** belief in the ability to detect events accurately, even when monitors are compromised




$$\mathbf{Conf}\left(\phi, M_d\right) = \max_{\sigma \in \beta(\phi)} \min_{\iota \in \sigma} \gamma_I\left(\iota, M_d\right)$$

# Metrics: Cost model

- Resource utilization cost
  - CPU utilization
  - Memory utilization
  - Disk storage
  - Network communication
- Amortized purchase price and recurring maintenance cost

# Outline


Domain expert knowlege
Model System
Define Monitoring Metrics
Optimally Select Monitors

# Optimal selection methodology

**Goal**: to be able to use methodology to answer a variety of monitor selection questions

- Minimum set of monitors that can detect a given attack/set of attacks (assuming no compromise)?
- Under cost constraints, what set of monitors will maximize ability to detect a high-priority attack?

# Capturing intrusion detection requirements

Represent detection requirements as **weights on metric values** and **minimum metric value constraints**

$\phi_1$ Account hacking attempt

$\phi_2$ Data exfiltration

$\phi_3$ SQL injection

**Requirements:**

- Must detect exfiltration
- Exfiltration, then SQL injection, are high priority (decreasing priority)
- Best effort for all others

$$\mathbf{min}_{\mathrm{Red}_{\phi_2}} = 1 \qquad \mathbf{w}_{\mathrm{Red}_{\phi_2}} = 2 \qquad \mathbf{w}_{\mathrm{Cov}} = 1$$

$$\mathbf{w}_{\mathrm{Red}_{\phi_3}} = 1$$

# Optimal selection methodology:
## Constrained-cost monitor selection

$$\underset{M_d}{\arg\max} \quad \begin{array}{l} \mathbf{w}_{\mathrm{Cov}}\mathbf{Cov}(\Phi, M_d) + \\ \quad \sum_{\phi \in \Phi} \mathbf{w}_{\mathrm{Red}_\phi}\mathbf{Red}(\phi, M_d) + \mathbf{w}_{\mathrm{Conf}_\phi}\mathbf{Conf}(\phi, M_d) \end{array}$$

$$\text{s.t.} \quad \begin{array}{ll} \mathbf{Cost}(M_d) \le \mathbf{maxCost} & \\ \mathbf{Cov}(\Phi, M_d) \ge \mathbf{min}_{\mathrm{Cov}} & \\ \mathbf{Red}(\phi, M_d) \ge \mathbf{min}_{\mathrm{Red}_\phi}, & \forall \phi \in \Phi \\ \mathbf{Conf}(\phi, M_d) \ge \mathbf{min}_{\mathrm{Conf}_\phi}, & \forall \phi \in \Phi \\ M_d \in \{0,1\}^{|M|} & \end{array}$$

Objective function: monitoring utility, defined as weighted sum of metric values
- Parameterized by user-specified weight parameters

Constraints:
- Cost function to minimize
- User-specified minimum detection metric requirements

0-1 integer nonlinear programming problem, with monitors as input variables

# Solving for optimal monitor selection

- Branch-and-bound algorithm
  - Searches over space of possible selections, pruning suboptimal sets of monitor selections
- Greedy heuristic algorithm
  - Maximizes effective utility increase by incrementally adding monitors until constraints are met

# EVALUATION

# Experiment Setup

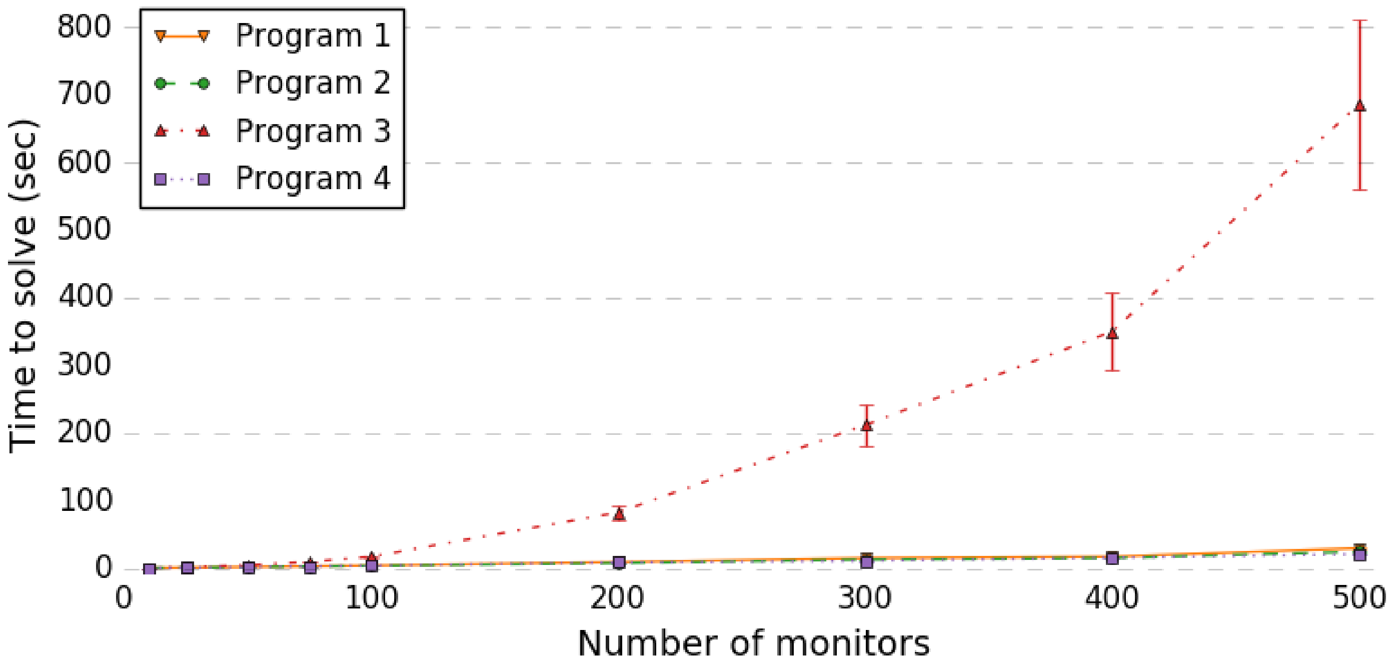
- Parameters: number of *monitors* and *events*
- Randomly generated 100 models for each set of parameters
- Created 4 sets of intrusion detection requirements – *optimal deployment programs*

**Goal:** Observe scalability and accuracy of greedy solution

# Evaluation: Greedy algorithm

Runtime complexity: $O\left(|I|\left(|M|^3 + |B||M|^2\right)\right)$

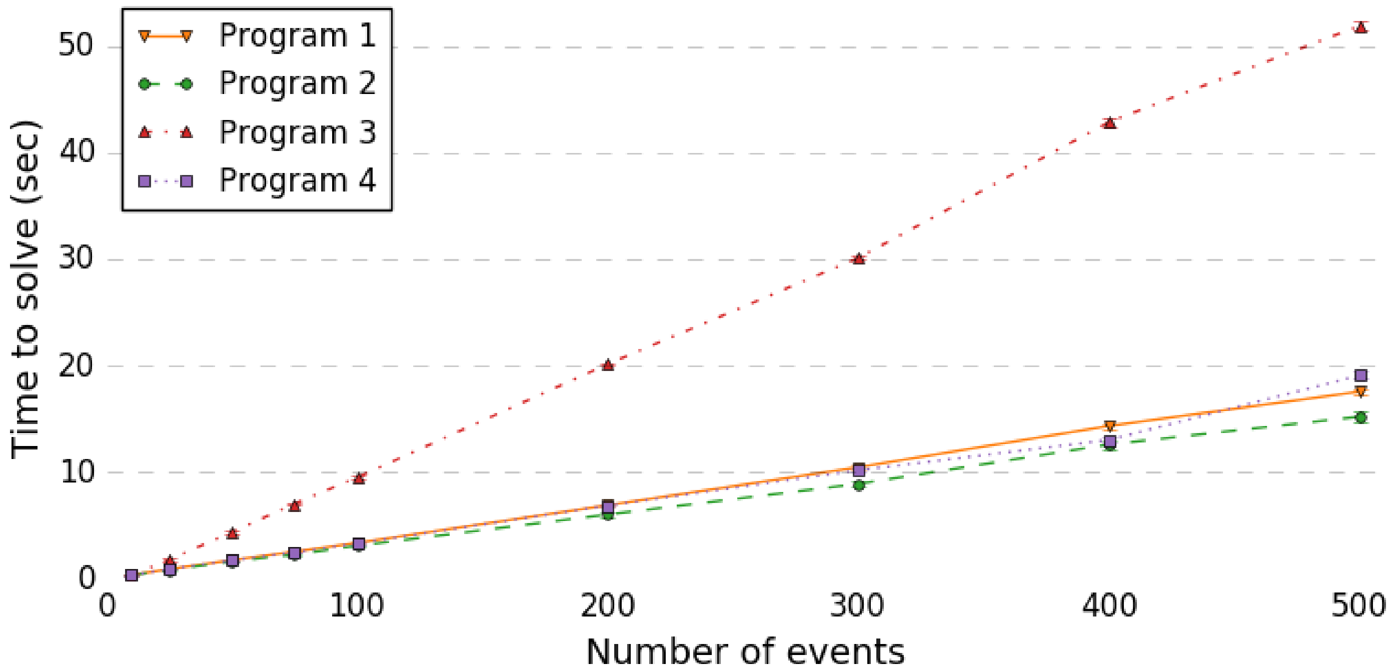
– Polynomial in the number of monitors $(|M|)$



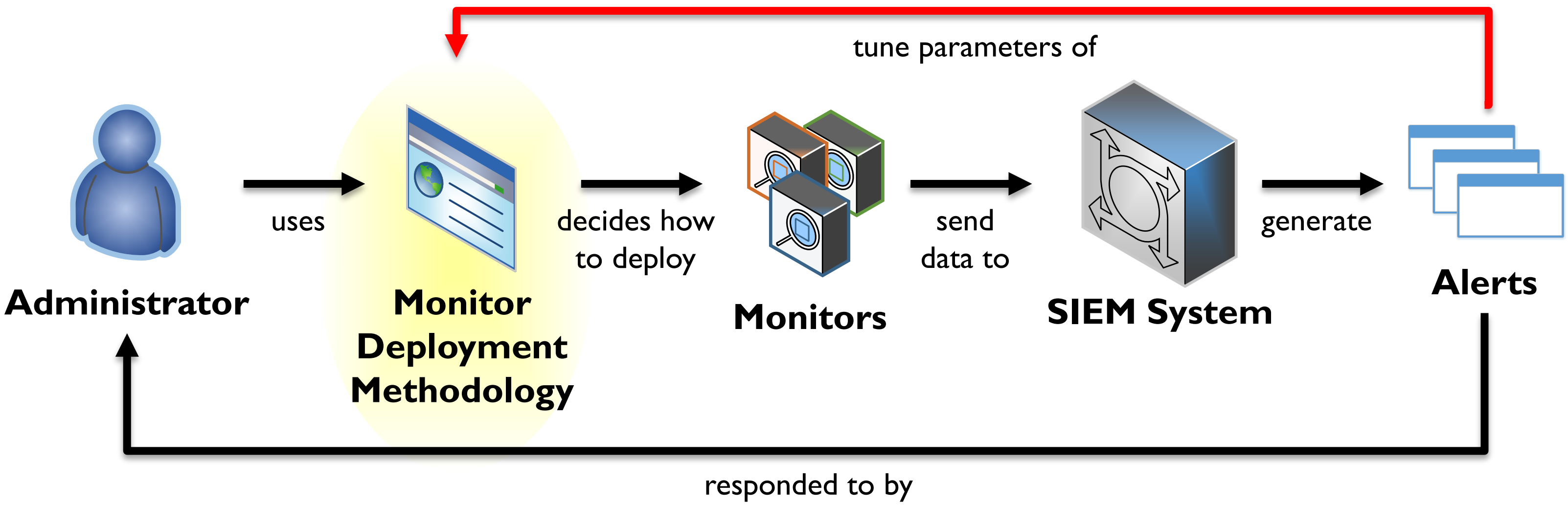
# Evaluation: Greedy algorithm

Runtime complexity: $O\left(|I|\left(|M|^3+|B||M|^2\right)\right)$

– Linear in the number of minimal indicator sets $\left(|B|\right)$ and indicators $\left(|I|\right)$



# Conclusions




- We help administrators make model-driven monitor placement decisions
- Administrators can more easily evaluate deployments
- Our methodology is expressive and scalable

**Future work:**

- Preemptive monitoring