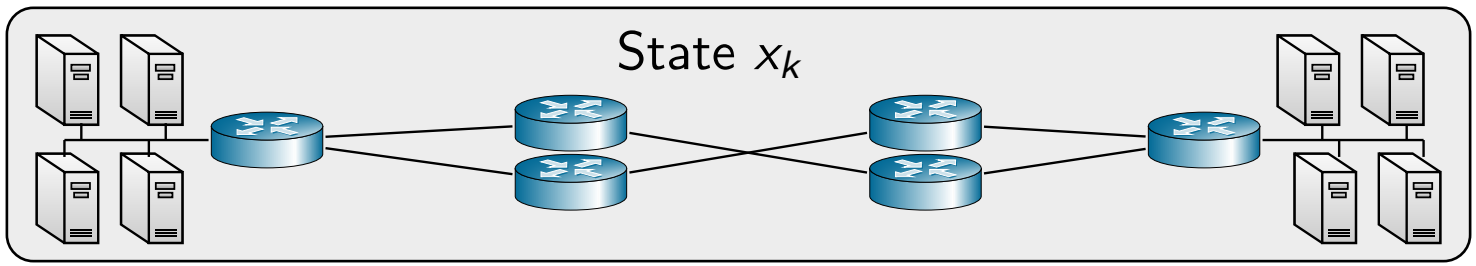
CSLE: A Reinforcement Learning Platform for Autonomous Security Management

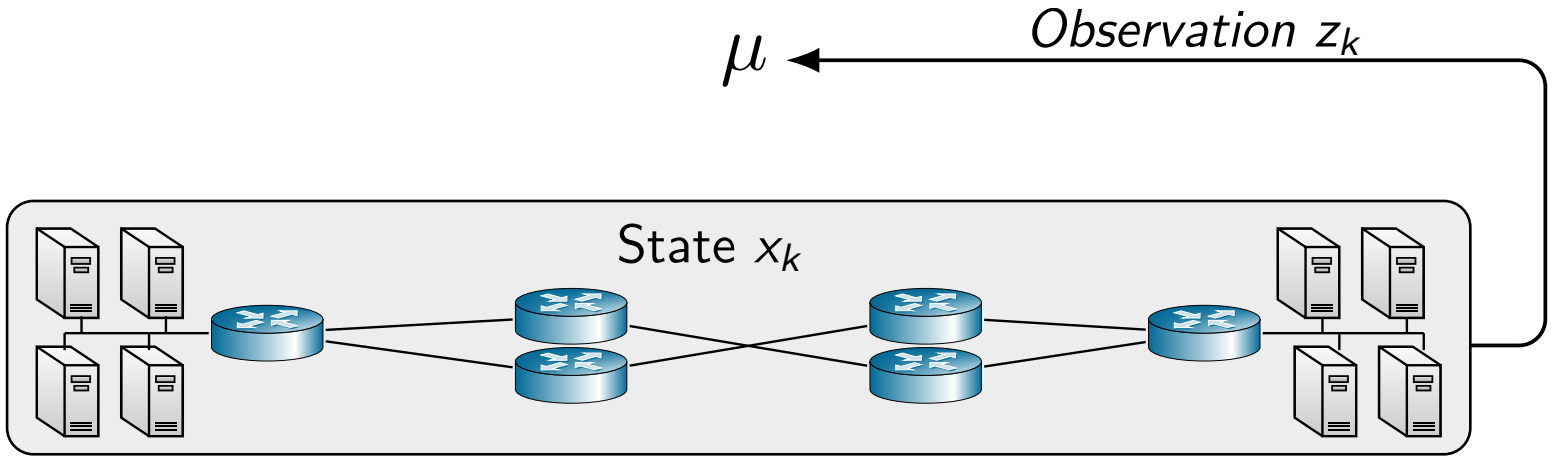
Ninth Annual Conference on Machine Learning and Systems (MLSys)
Bellevue, WA, USA, *May 19, 2026*

Kim Hammar
kimham@kth.se

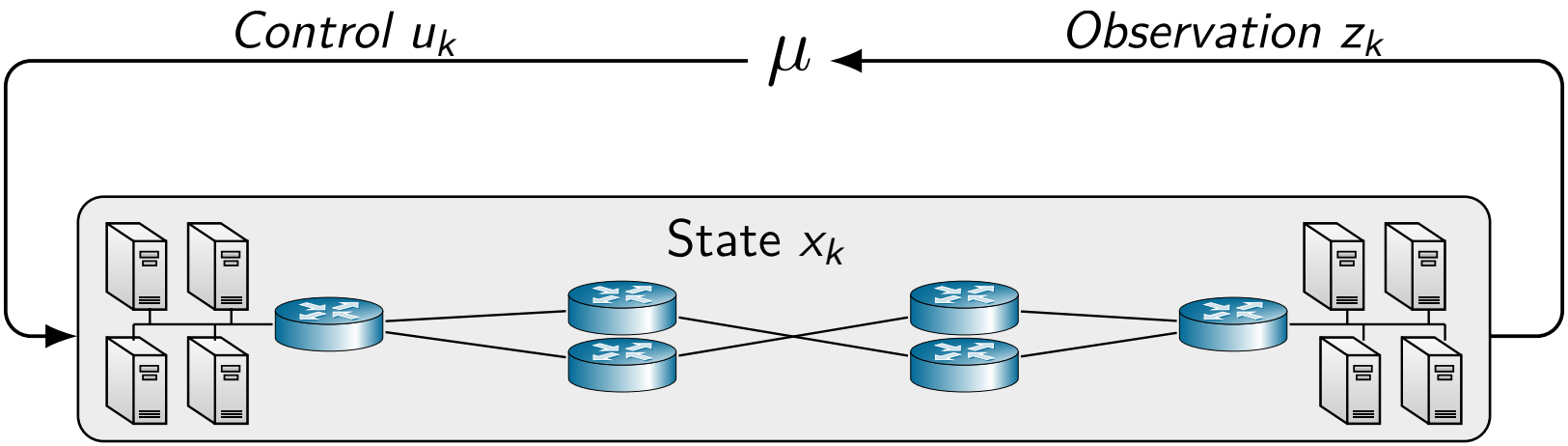
Autonomous Security Management as a Control Problem



Autonomous Security Management as a Control Problem

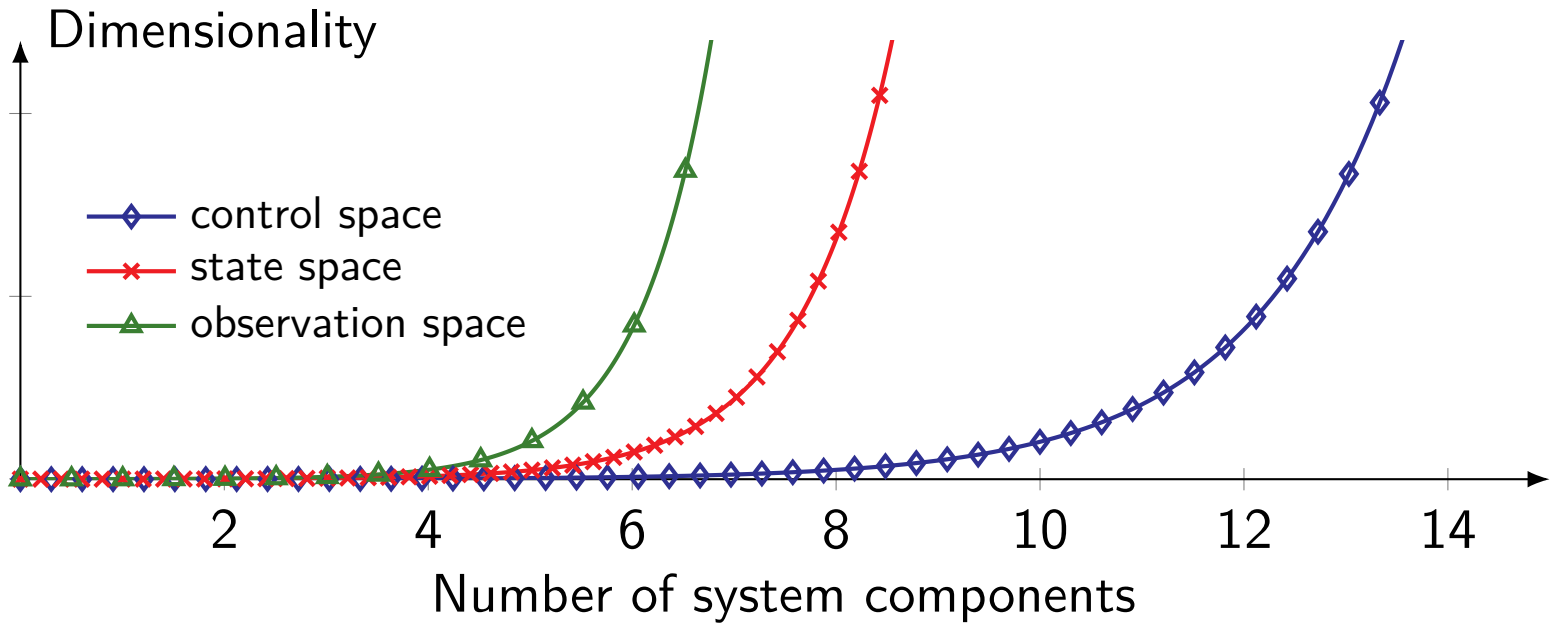


Autonomous Security Management as a Control Problem



- State x_k (e.g., security status and system configuration).
- Observation z_k (e.g., log files and security alerts).
- Control u_k (e.g., network segmentation and access control).
- **Goal**: find a strategy μ that meets security objectives.

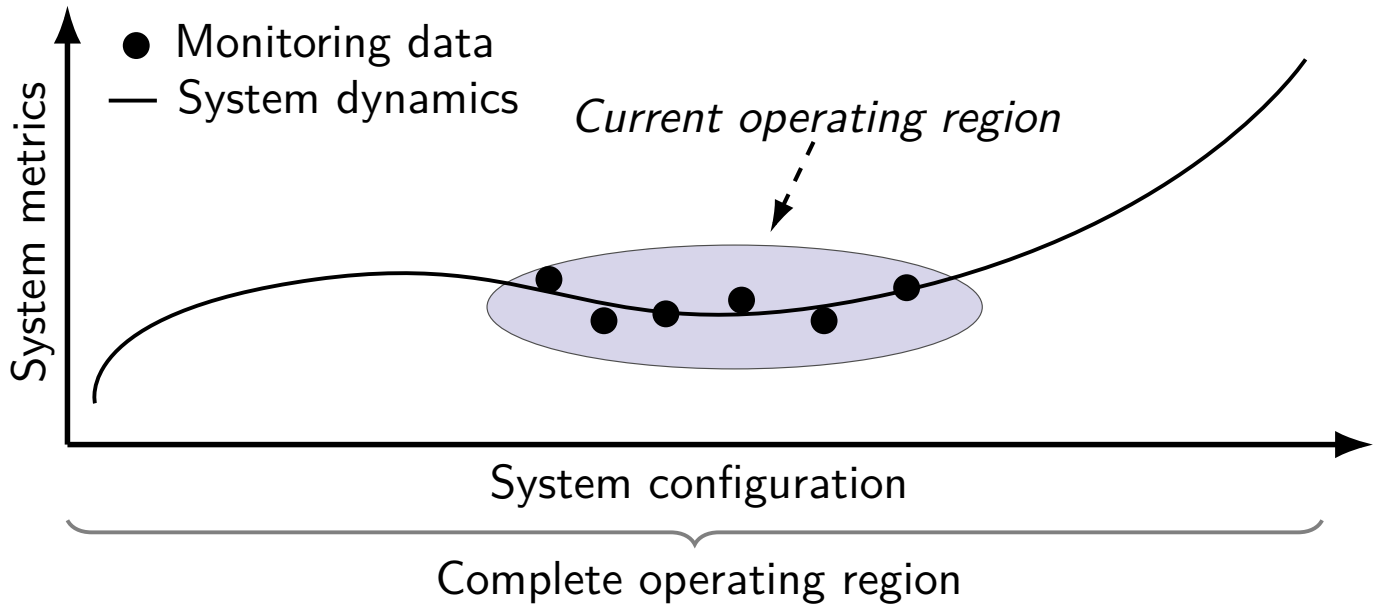
The Scalability Challenge



Curse of dimensionality

Networked system contain thousands of interconnected system variables $\Longrightarrow$ combinatorial explosion.

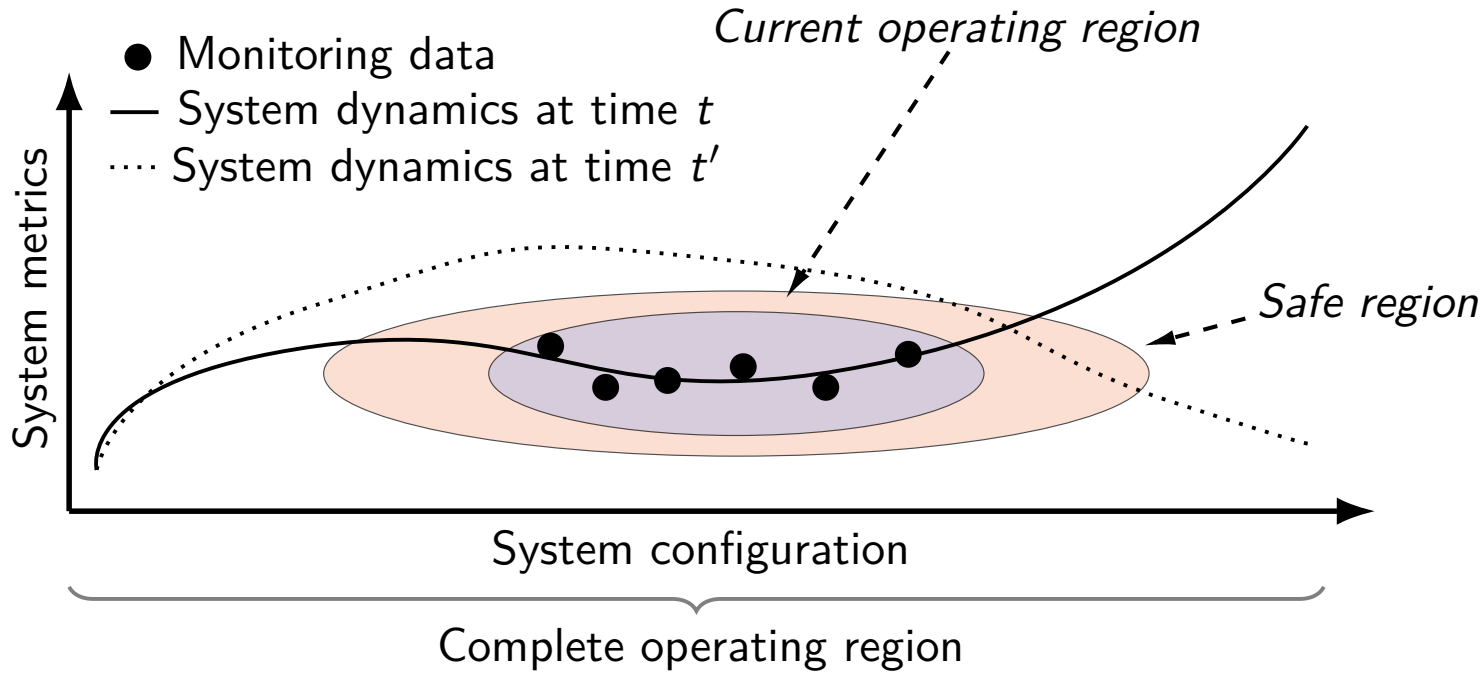
The Identification Challenge



System identification and modeling

There are no equations that describe how a networked system operates. The dynamics must be learned from data.

The Operation Challenge



Operational constraints and events

Networked systems are subject to operational constraints (e.g., performance and availability requirements) and unexpected events (e.g., non-stationarity and cyberattacks).

Summary Of Challenges

Curse of dimensionality

Networked system contain thousands of interconnected system variables $\Longrightarrow$ combinatorial explosion.

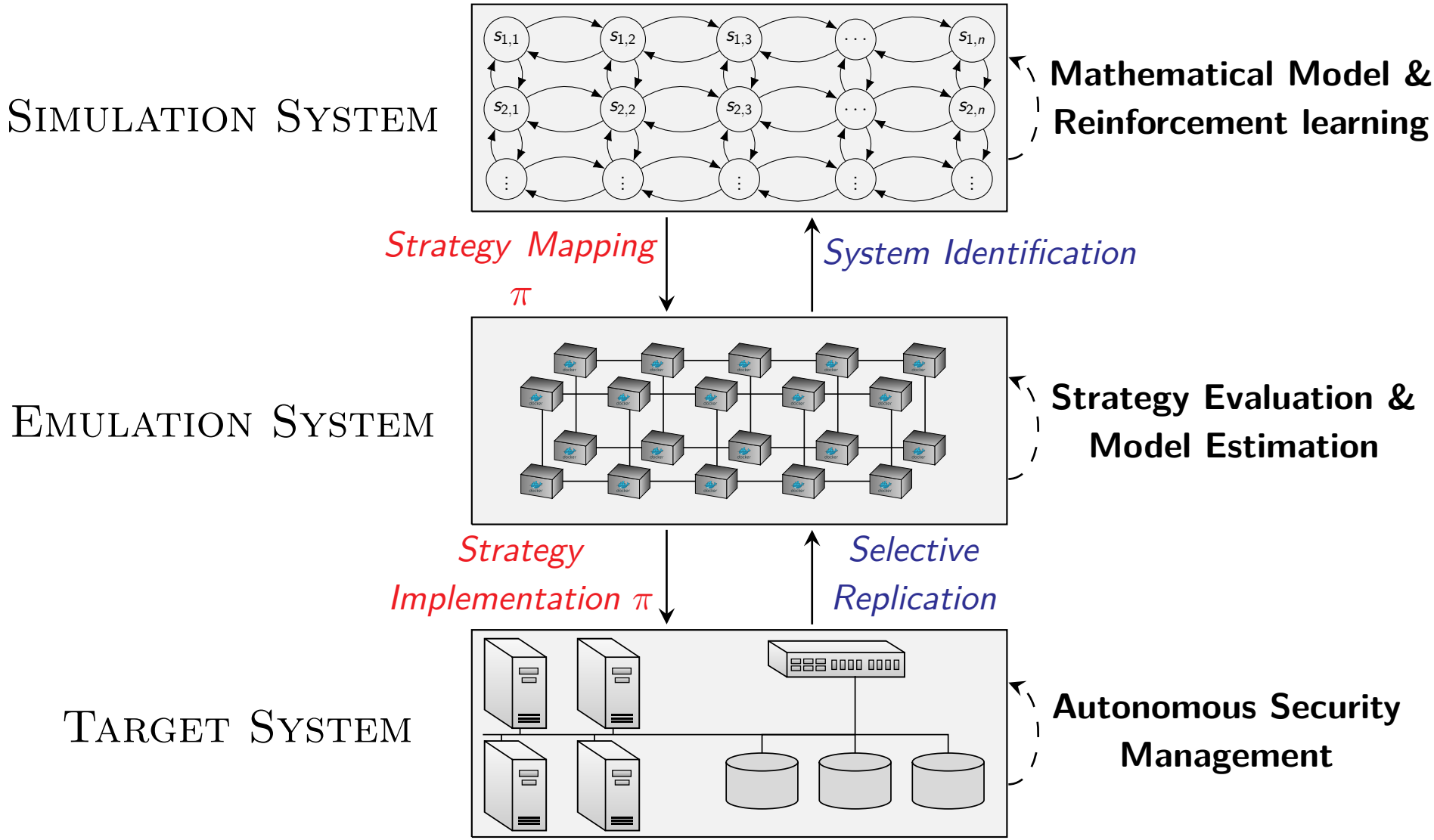
System identification and modeling

There are no equations that describe how a networked system operates. The dynamics must be learned from data.

Operational constraints and events

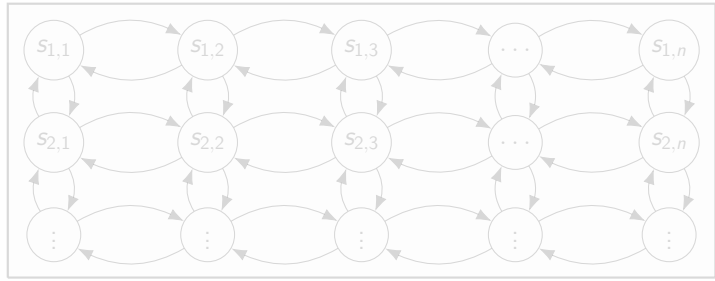
Networked systems are subject to operational constraints (e.g., performance and availability requirements) and unexpected events (e.g., non-stationarity and cyberattacks).

Methodology for Building Autonomous Security Systems



Methodology for Building Autonomous Security Systems

Simulation System

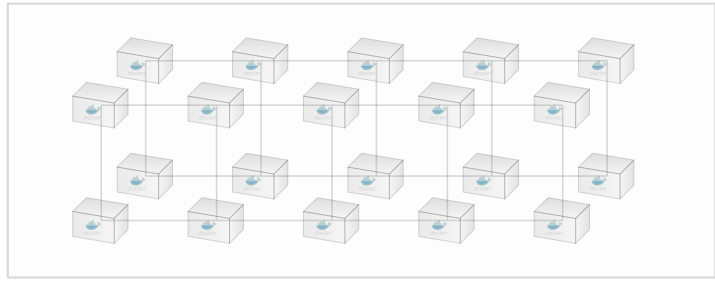


Mathematical Model & Reinforcement learning

Strategy Mapping π

System Identification

Emulation System

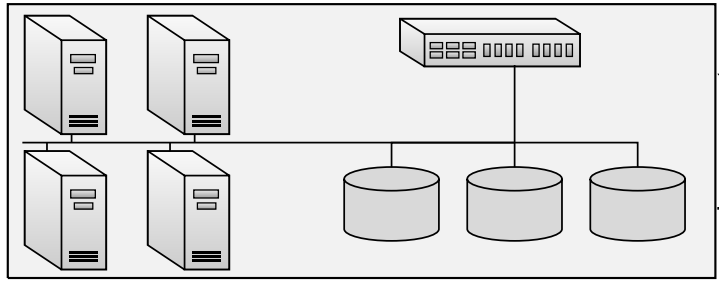


Strategy Evaluation & Model Estimation

Strategy Implementation π

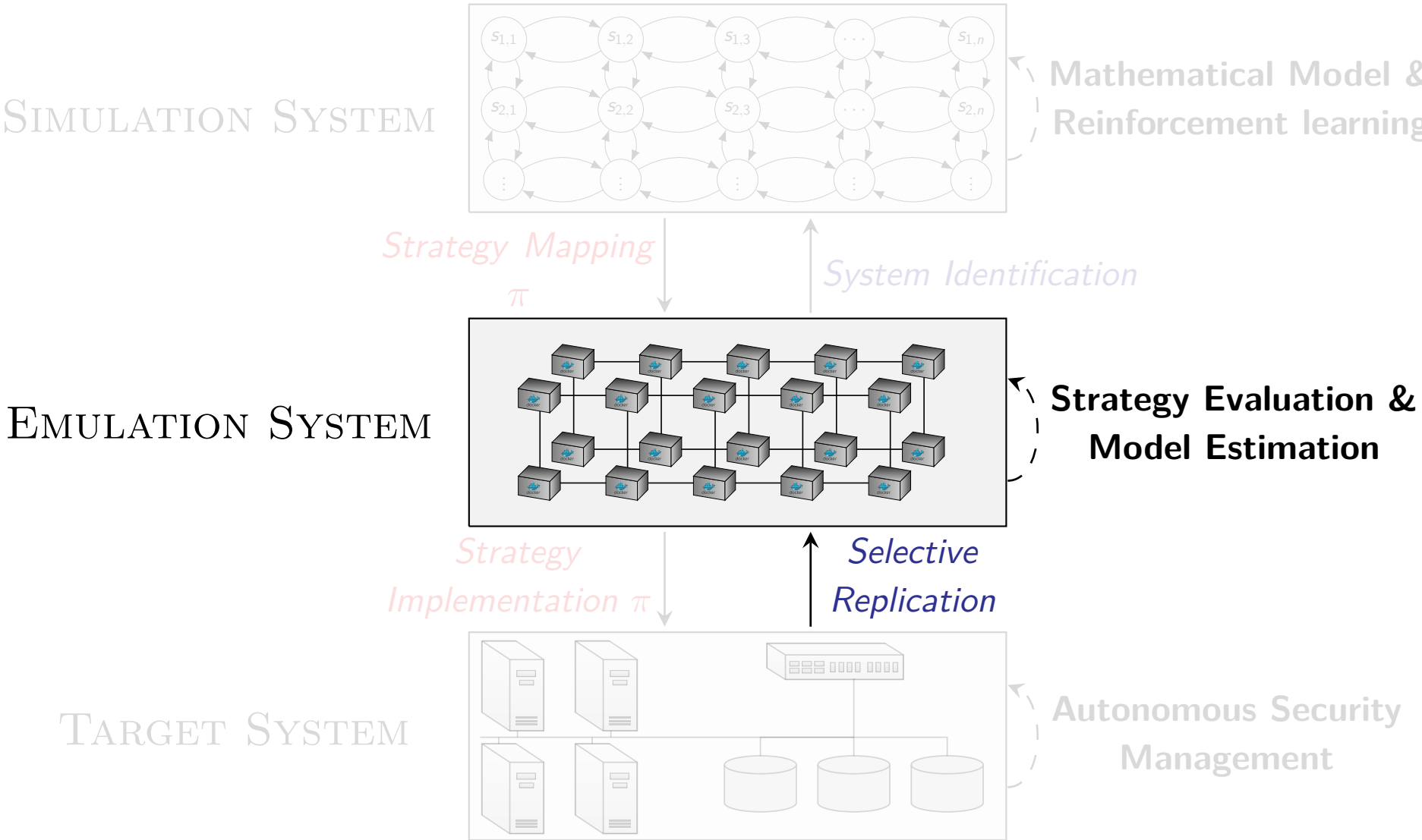
Selective Replication

Target System

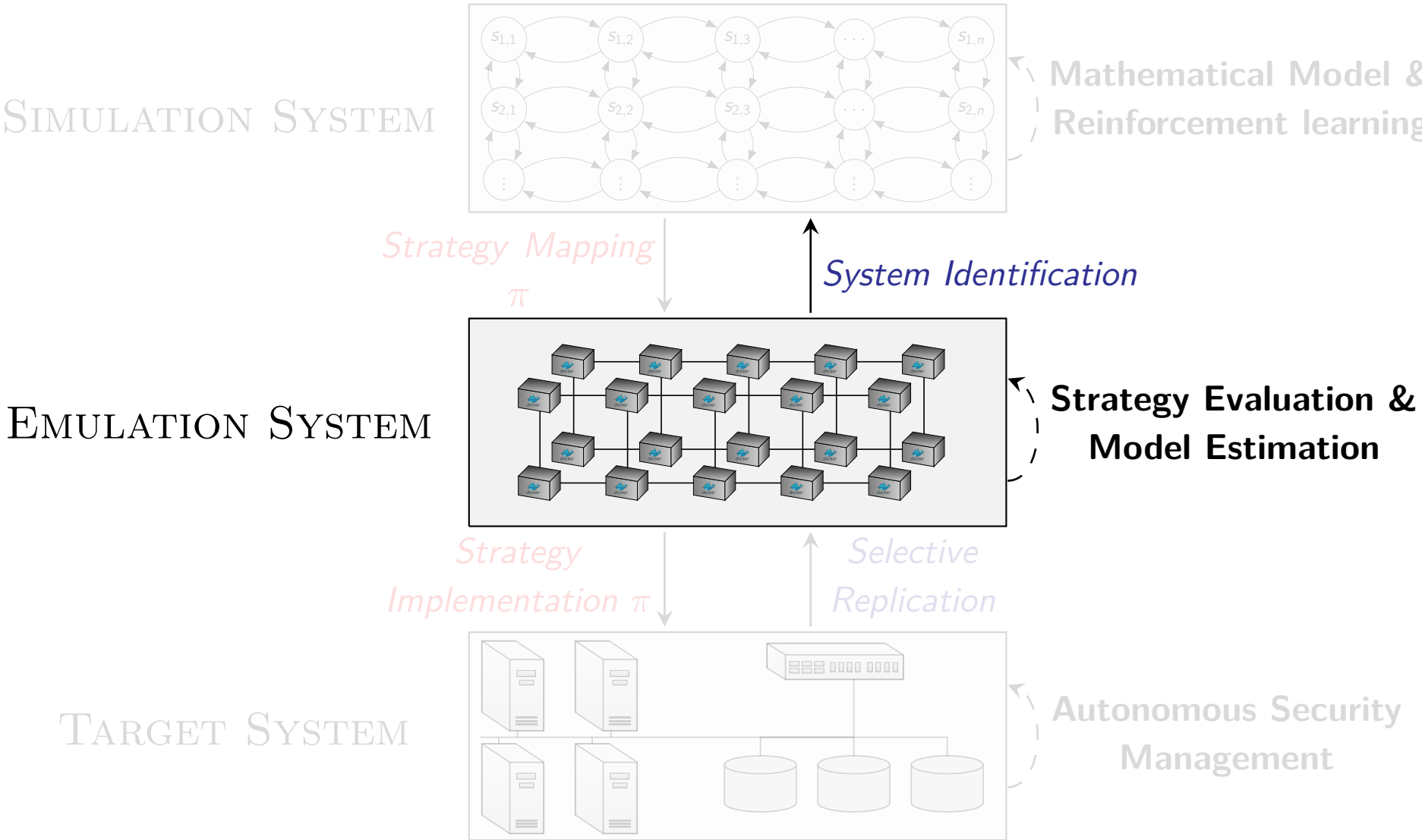


Autonomous Security Management

Methodology for Building Autonomous Security Systems

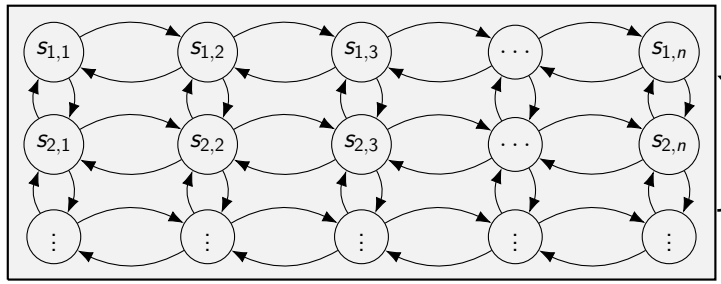


Methodology for Building Autonomous Security Systems



Methodology for Building Autonomous Security Systems

Simulation System

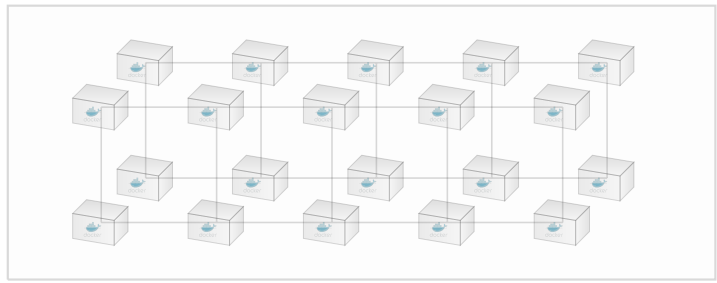


Mathematical Model & Reinforcement learning

Strategy Mapping π

System Identification

Emulation System



Strategy Evaluation & Model Estimation

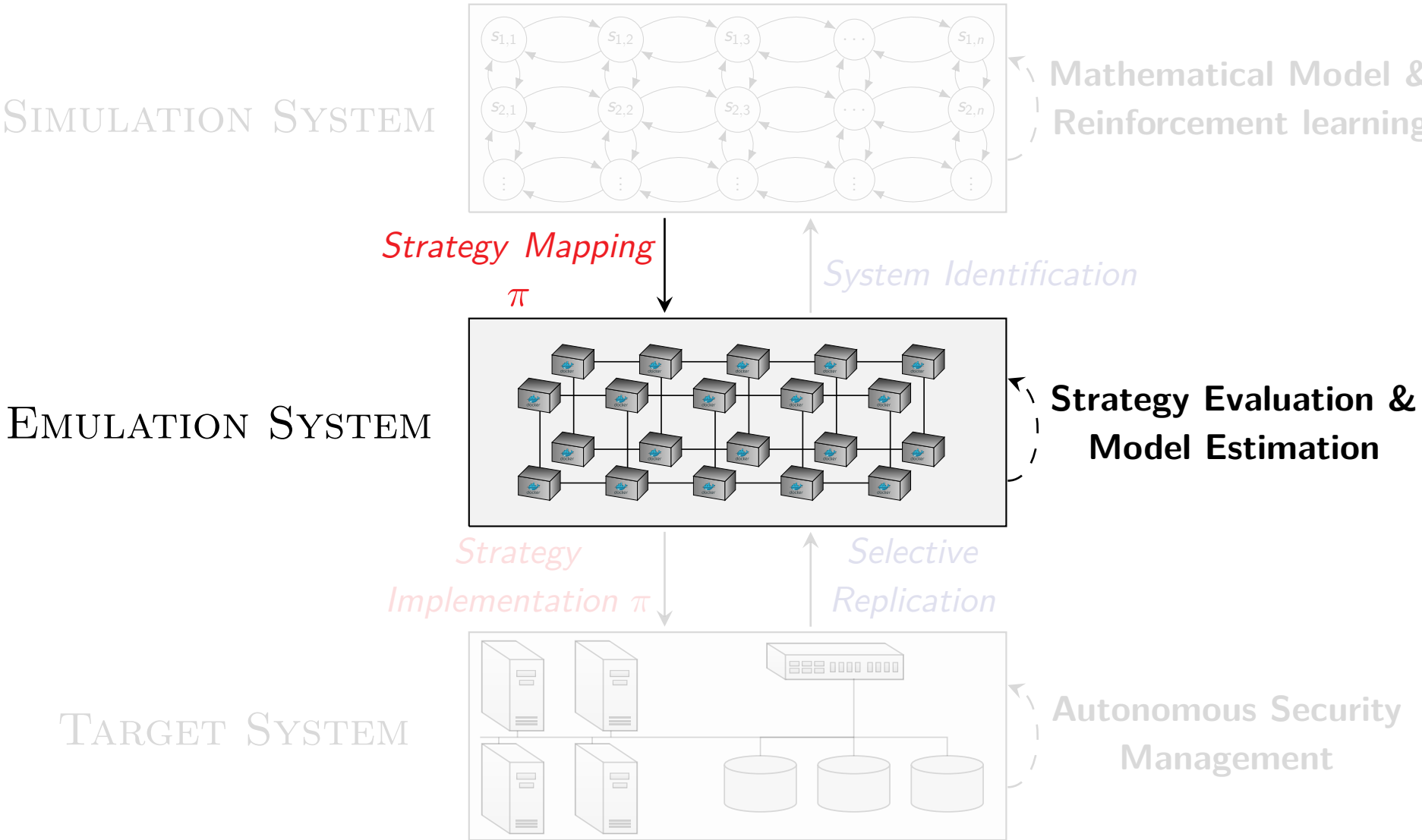
Strategy Implementation π

Selective Replication

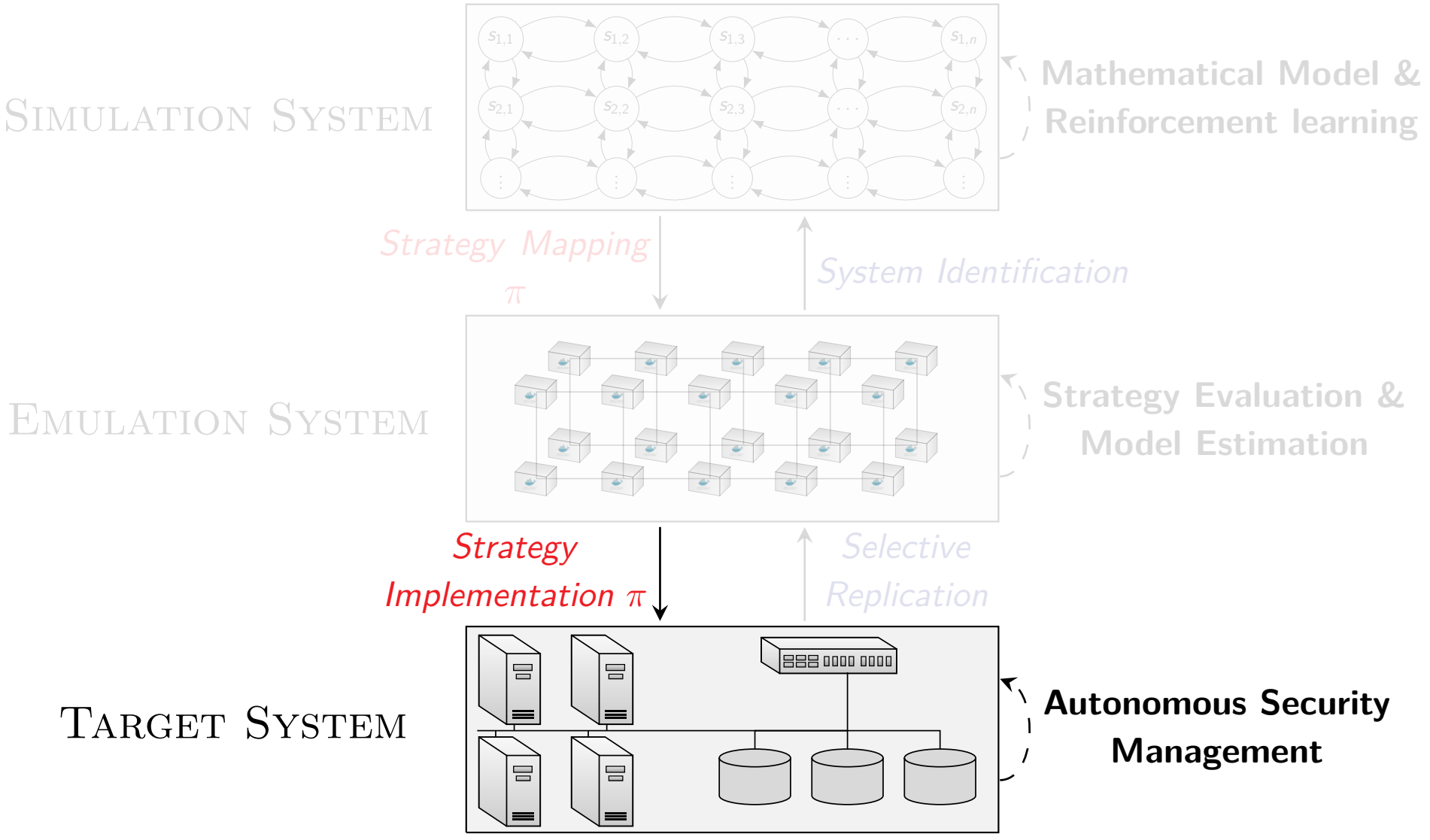
Target System

Autonomous Security Management

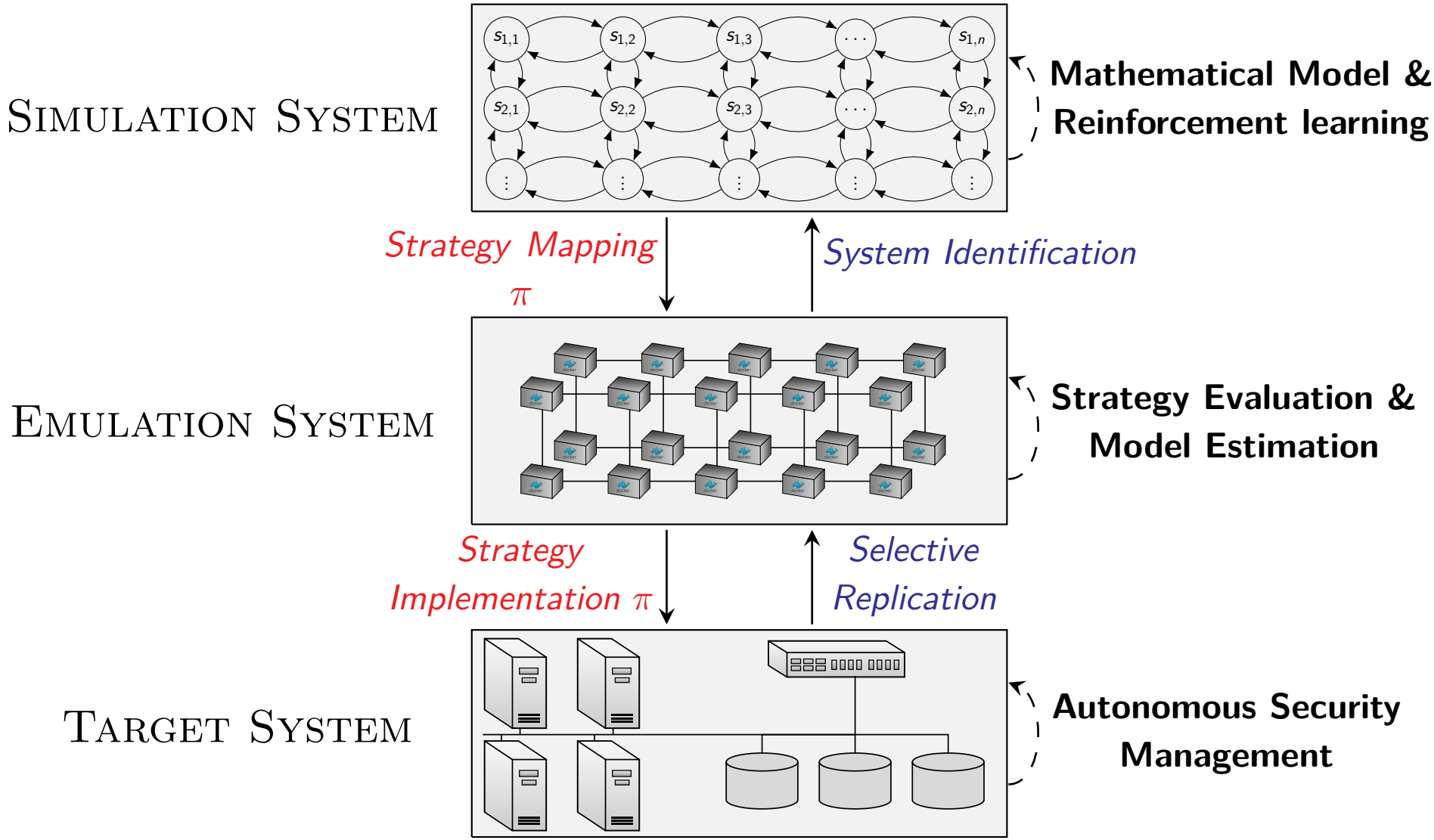
Methodology for Building Autonomous Security Systems



Methodology for Building Autonomous Security Systems



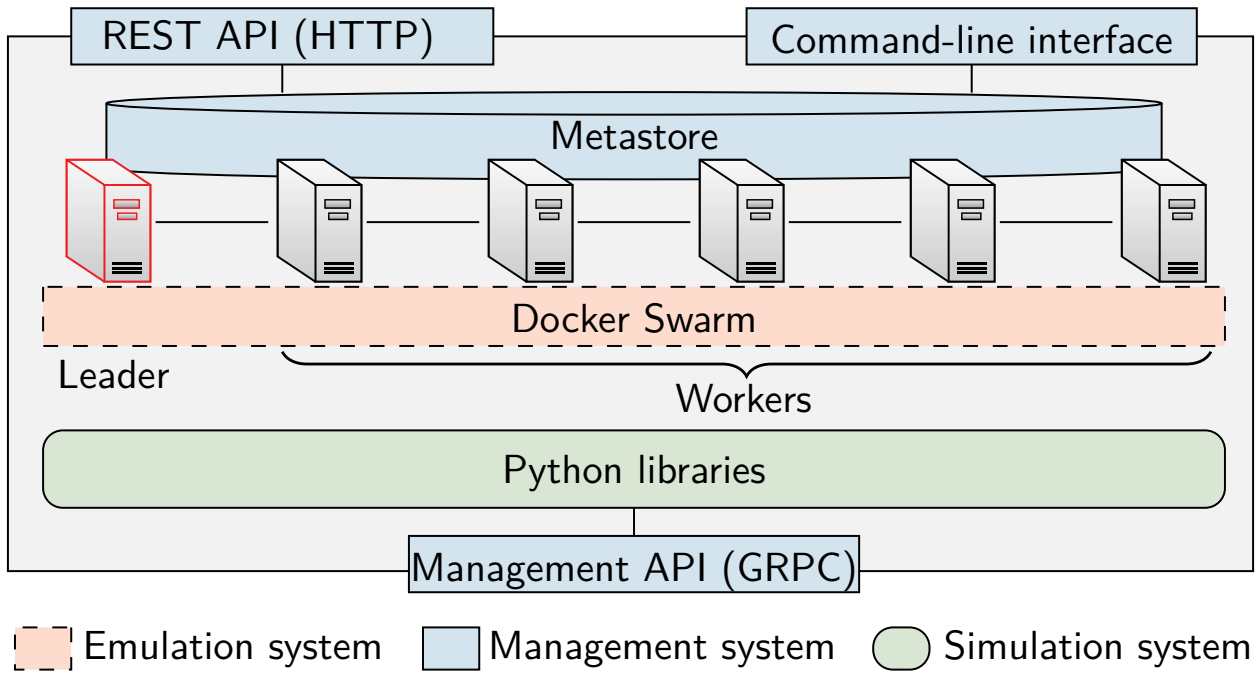
Methodology for Building Autonomous Security Systems



CSLE: A Platform that Supports the Methodology

- The Cyber Security Learning Environment (CSLE) enables experimentation with reinforcement learning for autonomous security management under realistic conditions.

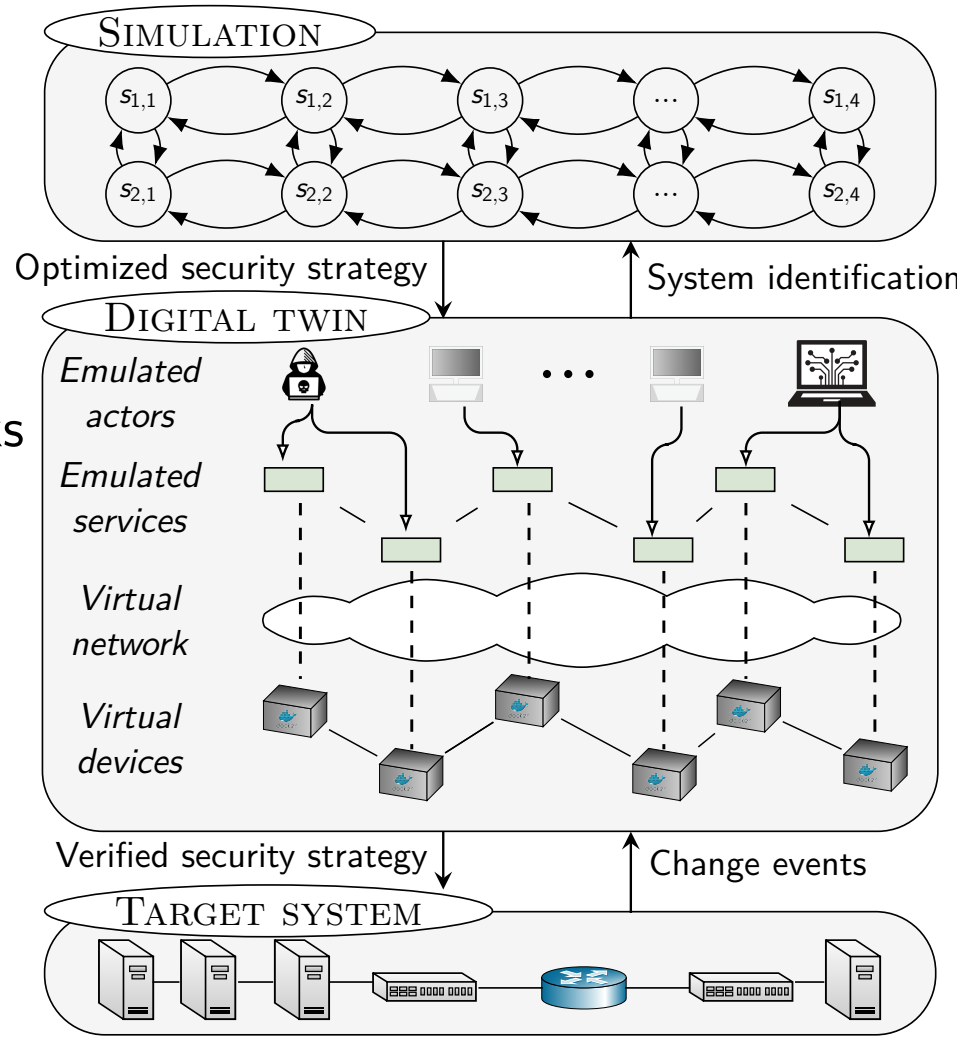
- The implementation of CSLE consists of three systems:
  - An emulation system for creating digital twins.
  - A simulation system for reinforcement learning.
  - A management system for orchestration.



The Emulation System

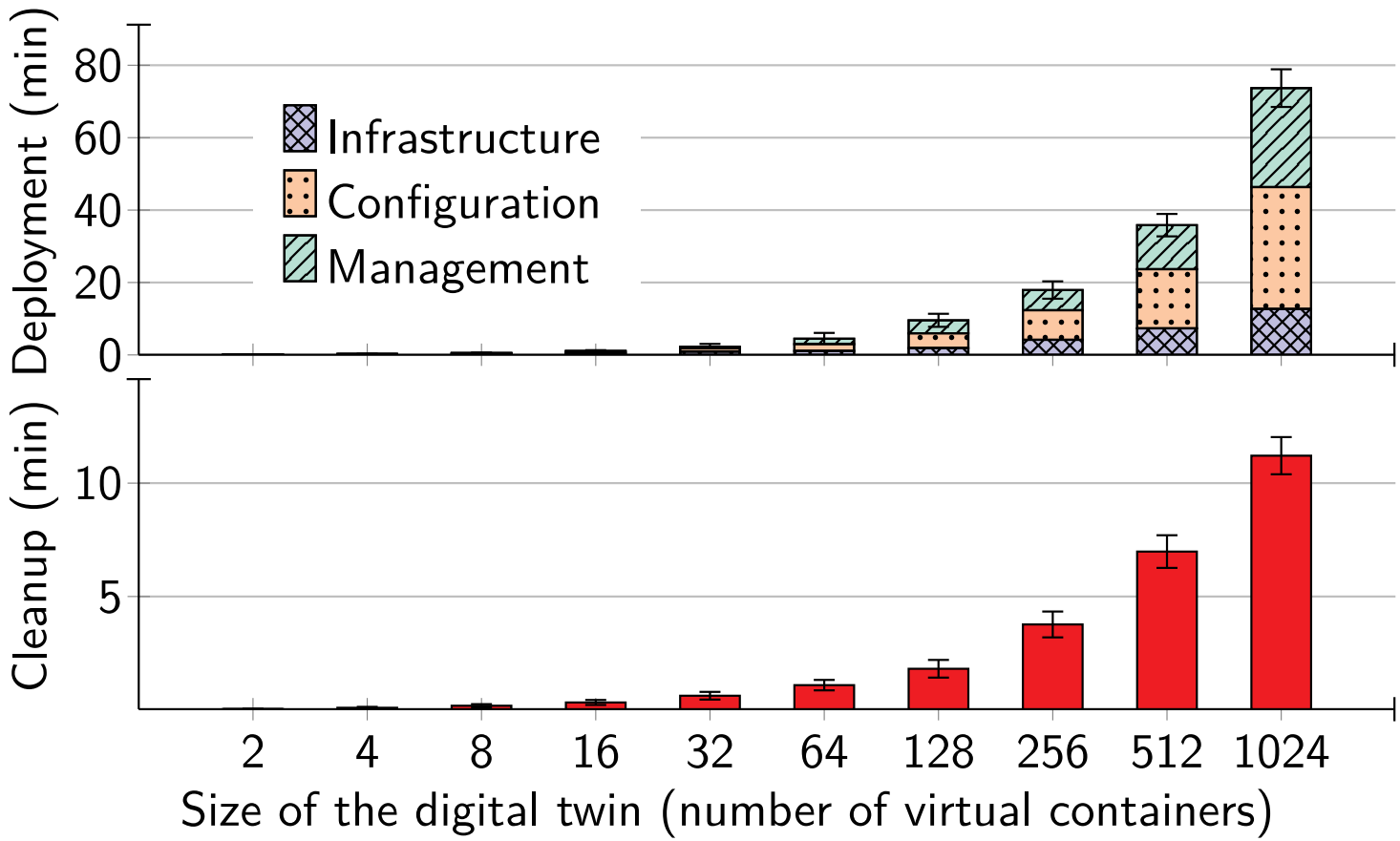
- **We emulate**
  - System components using Docker.
  - Actors using automated scripts.
  - Network links using virtual networks netem.

- A digital twin virtual replica of a networked system.



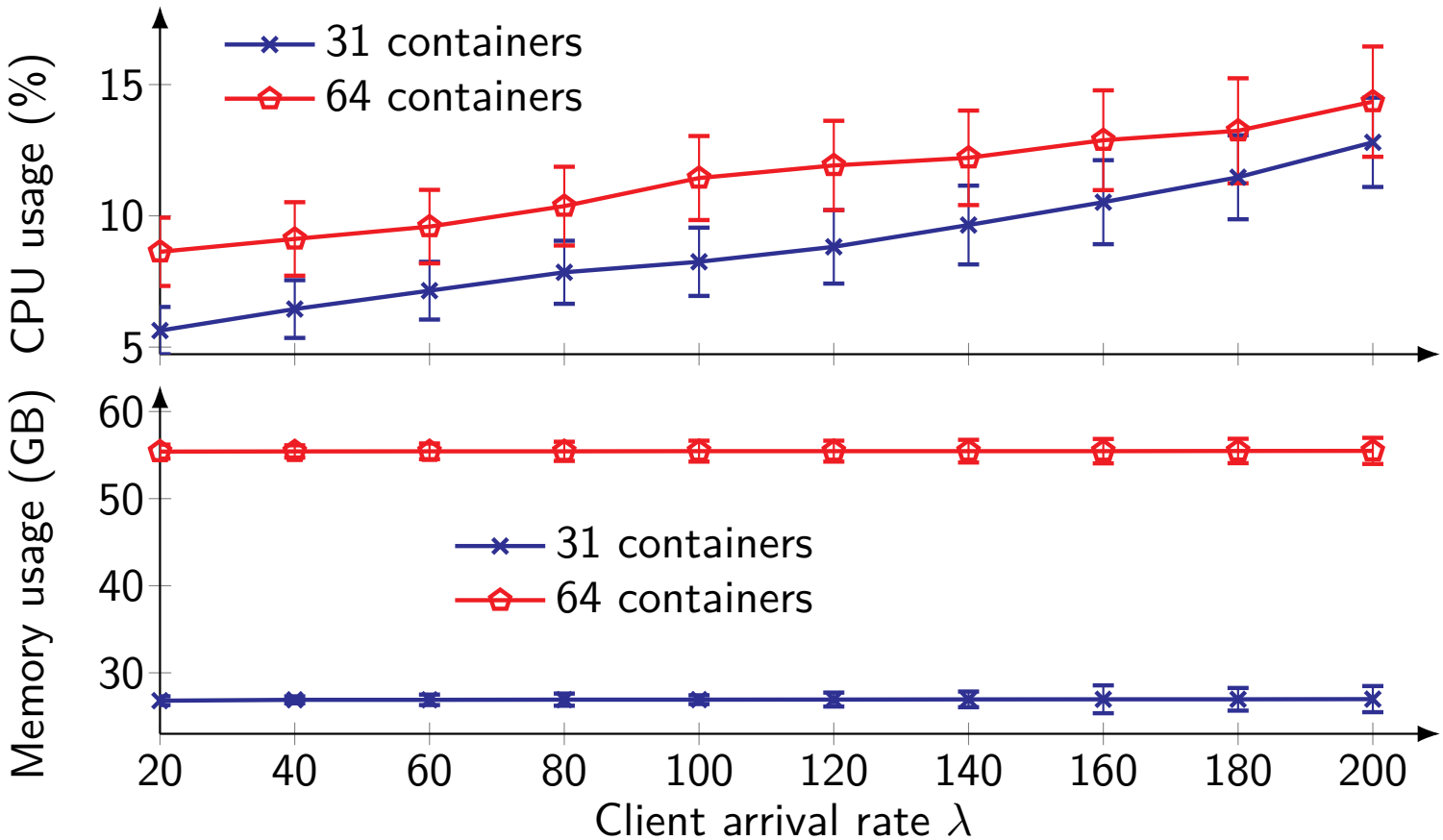
Scalability of the Emulation System (1/2)

- Virtualization lets us quickly instantiate large digital twins.



Scalability of the Emulation System (2/2)

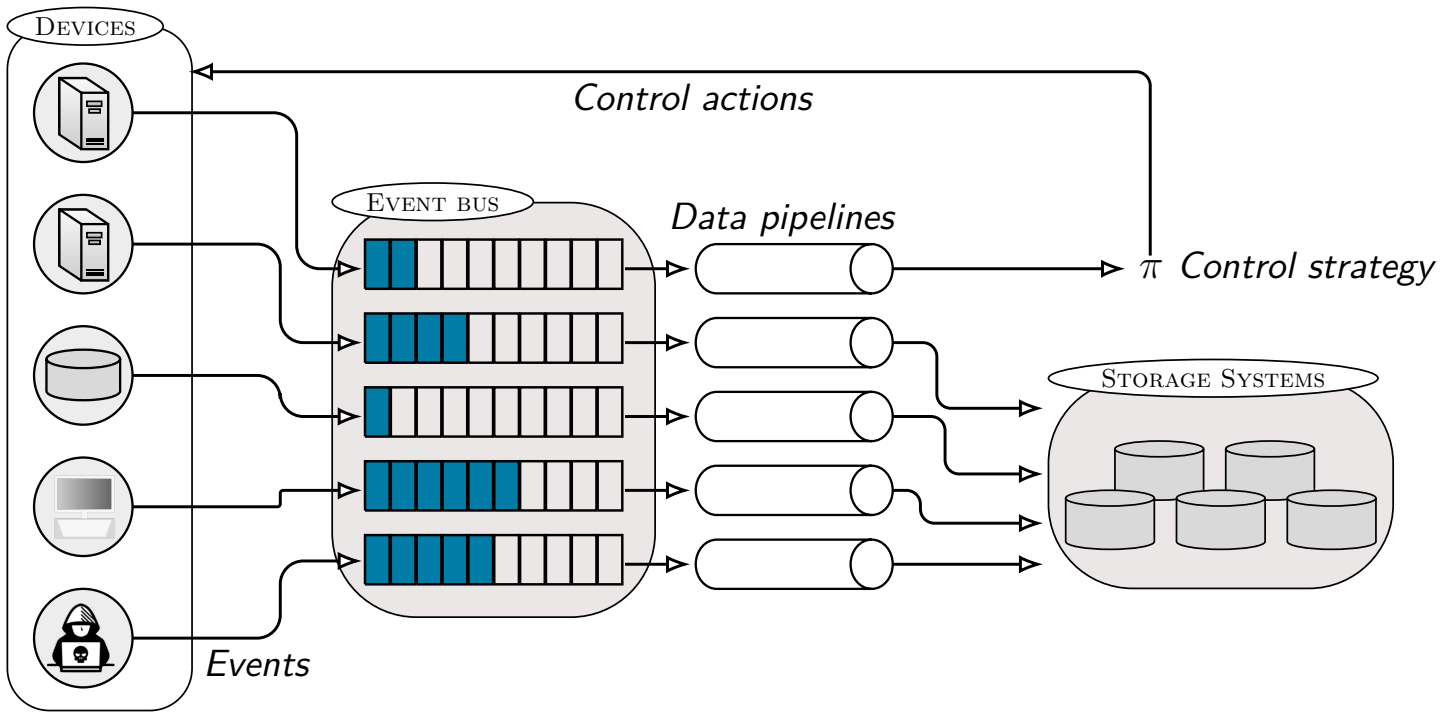
- Lightweight containers allow us to run large digital twins on commodity hardware.



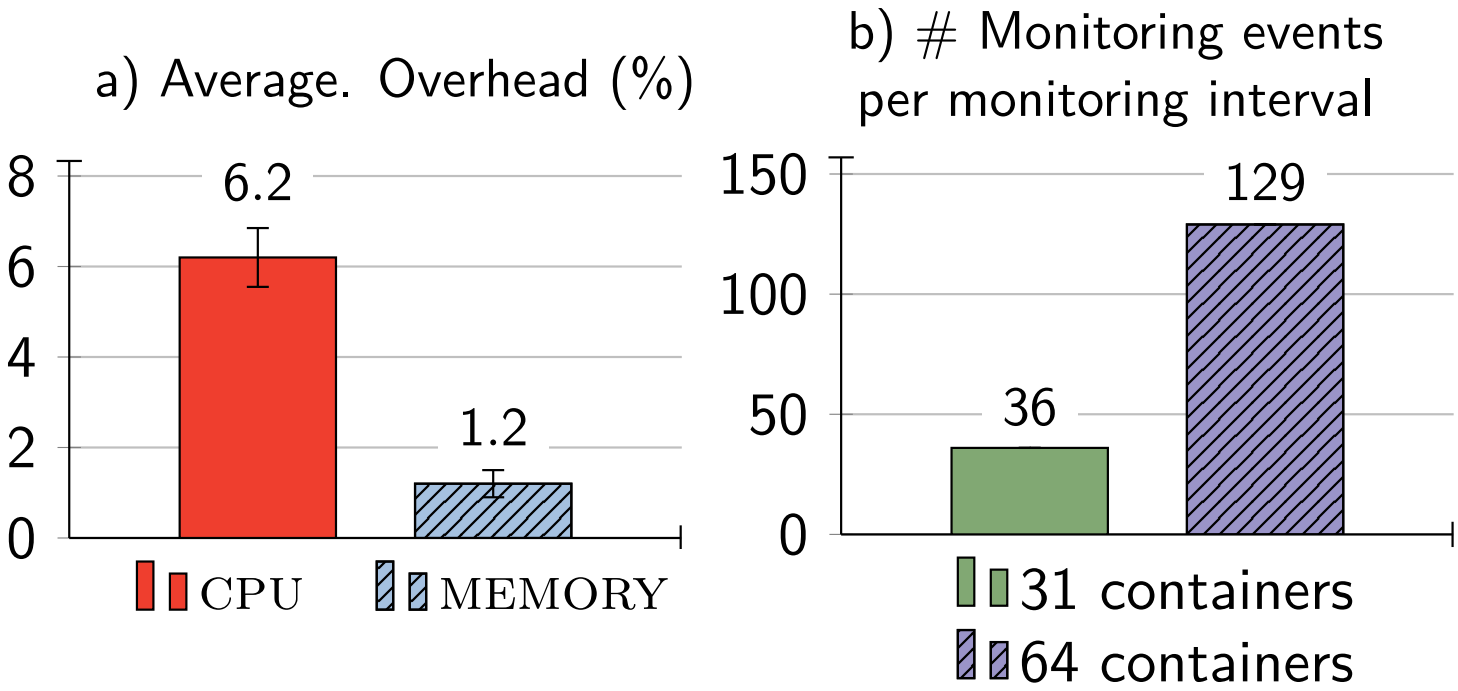
The Management System

- The management system provides APIs for real-time monitoring and control of digital twins.

- Each host in a digital twin runs a **management agent** that
  - Allows to execute control actions.
  - Reads local metrics and pushes them to an event bus for real-time monitoring.

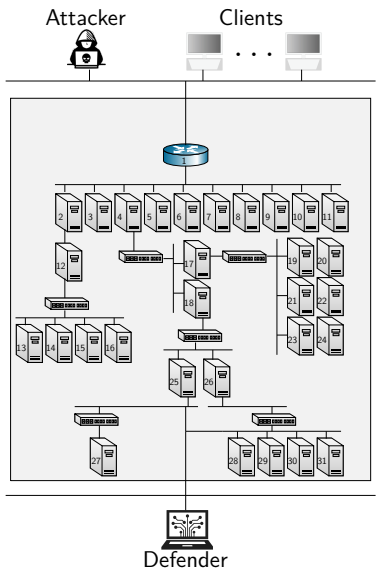


Computational Overhead of the Management System

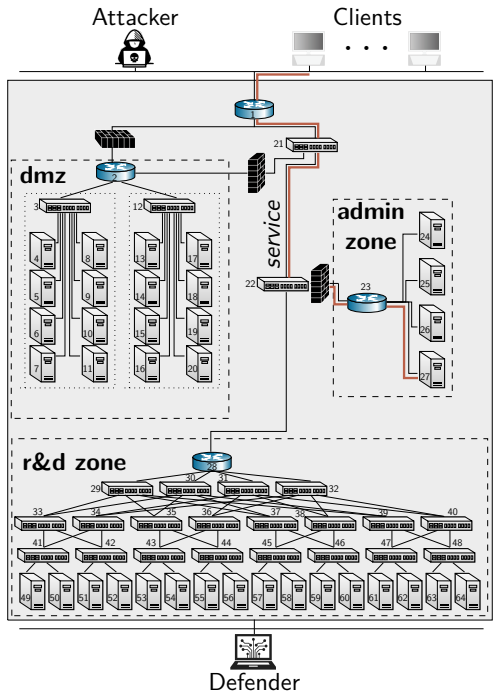


Example Use Cases for Experimental Evaluation

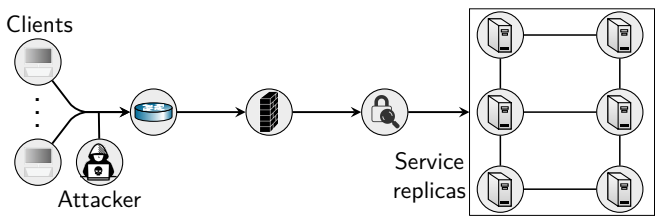
- We demonstrate CSLE by applying it to four use cases.
  - **Flow control**: block network flows to mitigate intrusions.
  - **Segmentation control**: direct network flows or create network zones to mitigate intrusions.
  - **Recovery control**: Decide when to recover components in a replicated system to maintain service availability.
  - **Replication control**: Select the number of replicas.



a) Target system for the flow control use case.



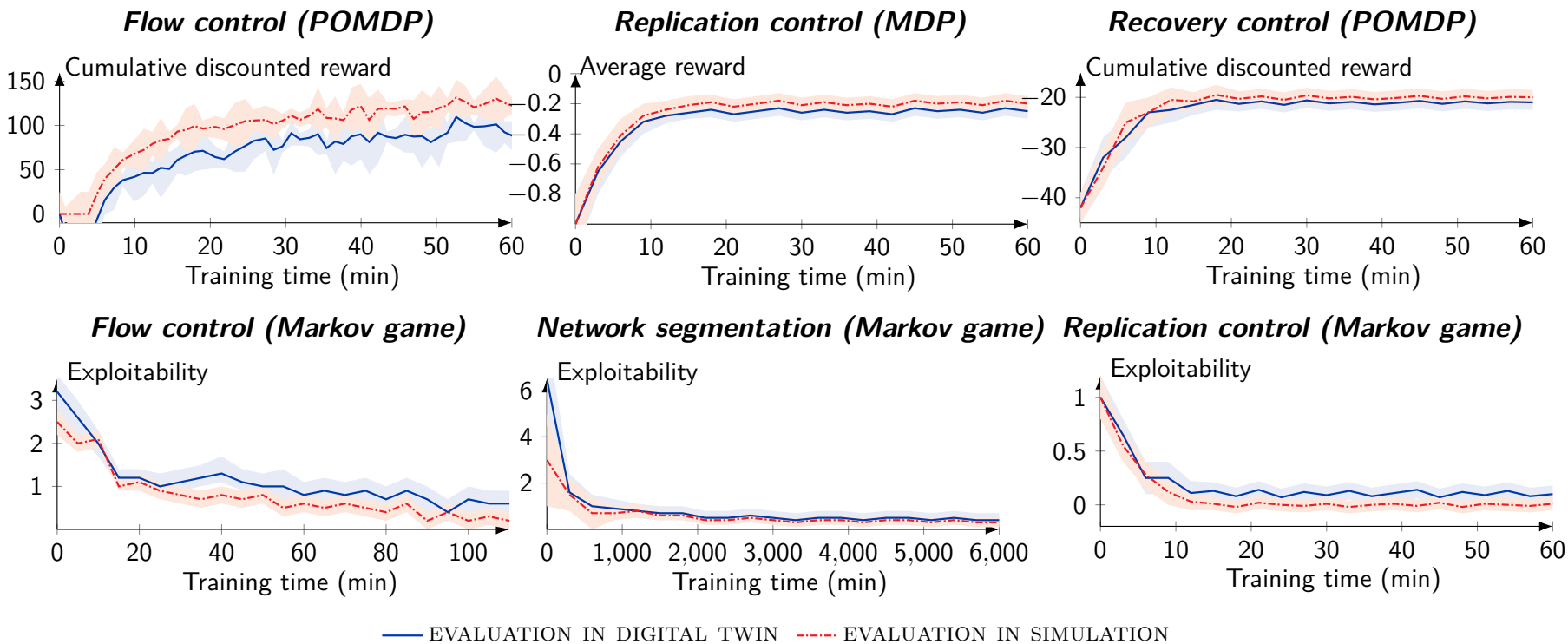
b) Target system for the segmentation use case.



c) Target system for the replication and recovery use cases.

Experimental Evaluation

- **Evaluation metrics:**
  - Reward for decision-theoretic models (↑ better).
  - Exploitability for game-theoretic models. (↓ better).



Performance on the simulator transfers to the digital twin.

Conclusion

- CSLE is a platform for autonomous security management.
  - Enables experimentation in realistic operating conditions.
  - We use CSLE to demonstrate simulation-to-emulation transfer.
  - Open source: https://github.com/Kim-Hammar/csle.