Georgia Institute of Technology

Dr. Nick Feamster
Associate Professor

Software Defined Networking

In this course, you will learn about software defined networking and how it is changing the way communications networks are managed, maintained, and secured.

School of Computer Science

Module 6.5: Programming SDNs

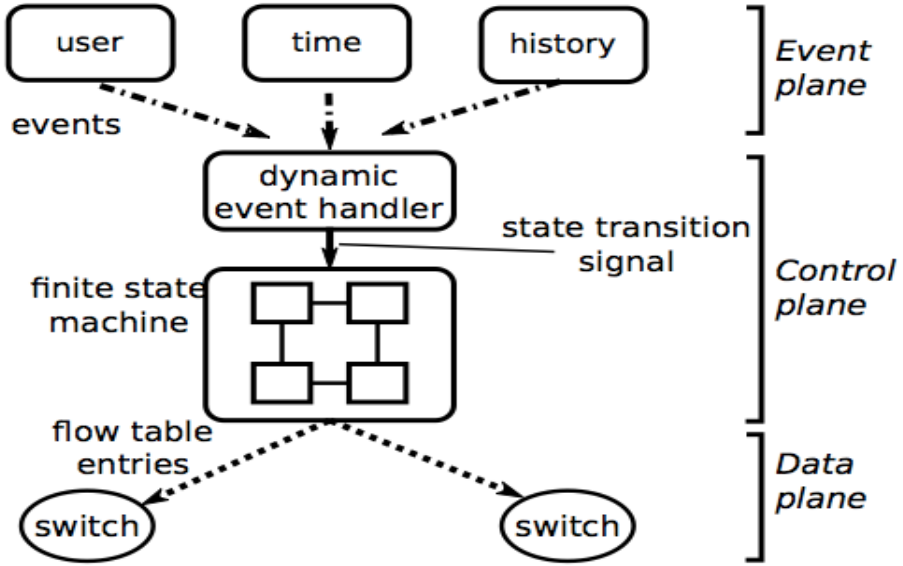
- Four Lessons
  - Motivation for Programming SDNs
  - Programming Languages for SDNs
  - Composing SDN Control
    - Pyretic
  - **Event-Driven SDN**
- Programming Assignment
- Quiz

Much of Network Configuration is Really Just Event Processing!

- Rate limit all Bittorrent traffic between the hours of 9 a.m. and 5 p.m.
- Do not use more than 100 GB of my monthly allocation for Netflix traffic
- If a host becomes infected, re-direct it to a captive portal with software patches
- …

Resonance: Event-Based Network Control

Main Idea: Express network policies as event-based programs.



Resonance: Inference-Based Access Control for Enterprise Networks. Nayak, Reimers, Feamster, Clark. *ACM SIGCOMM Workshop on Enterprise Networks*. August 2009.

Event-Driven Control Domains

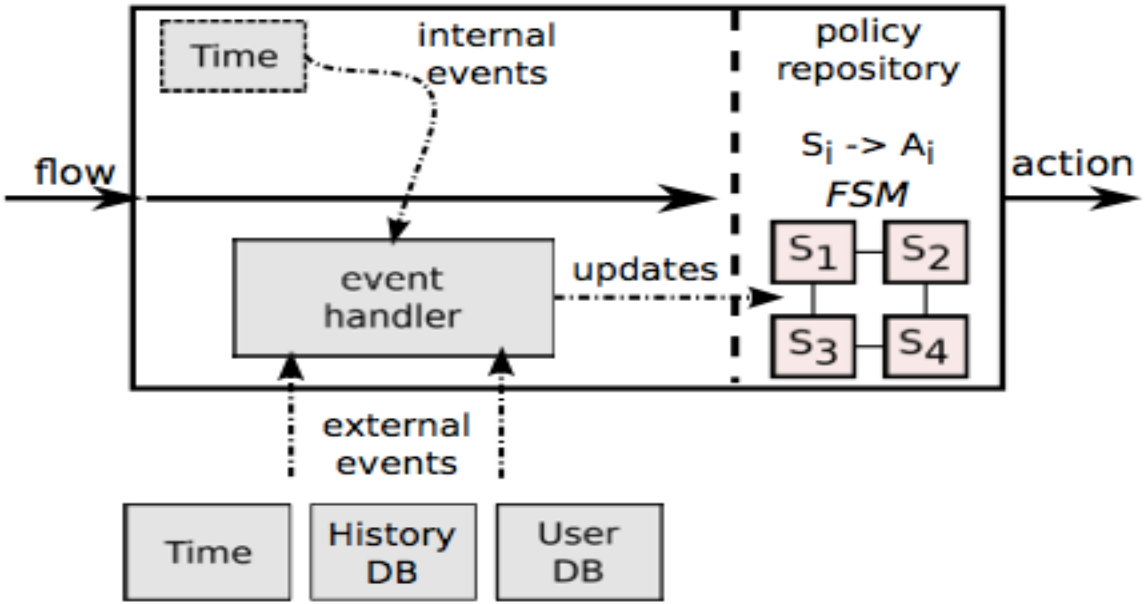
domains	*Examples*
Time	peak traffic hours, academic semester start date
History	amount of data usage, traffic rate, traffic delay, loss rate
User	identity of the user, assignment to distinct policy group
Flow	ingress port, ether src, ether dst, ether type, vlan id, vlan priority, IP src, IP dst, IP dst, IP ToS bits, src port, dst port

Resonance: Finite State Machine

- **State:** A set of domain values represents a state. Representation of network state.

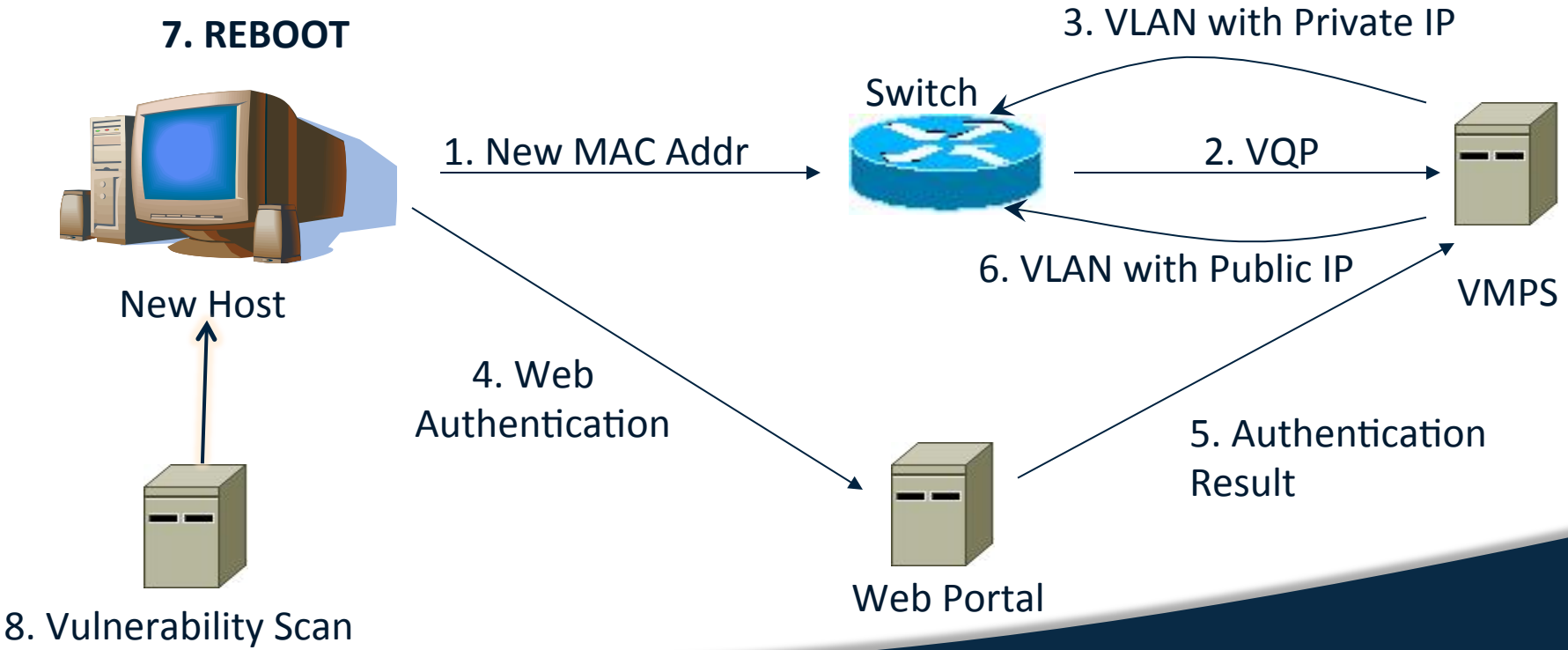
- **Events:** Event-driven control domains invoke events, which trigger state transitions in the controller's finite state machine.
  - Intrusions
  - Traffic fluctuations
  - Arrival/departure of hosts

Resonance: Dynamic Event Handler



- Reacts to domain events
- Determines event source
- Updates state based on event type
- Can process both internal and external events

Example from Campus Network: Access Control

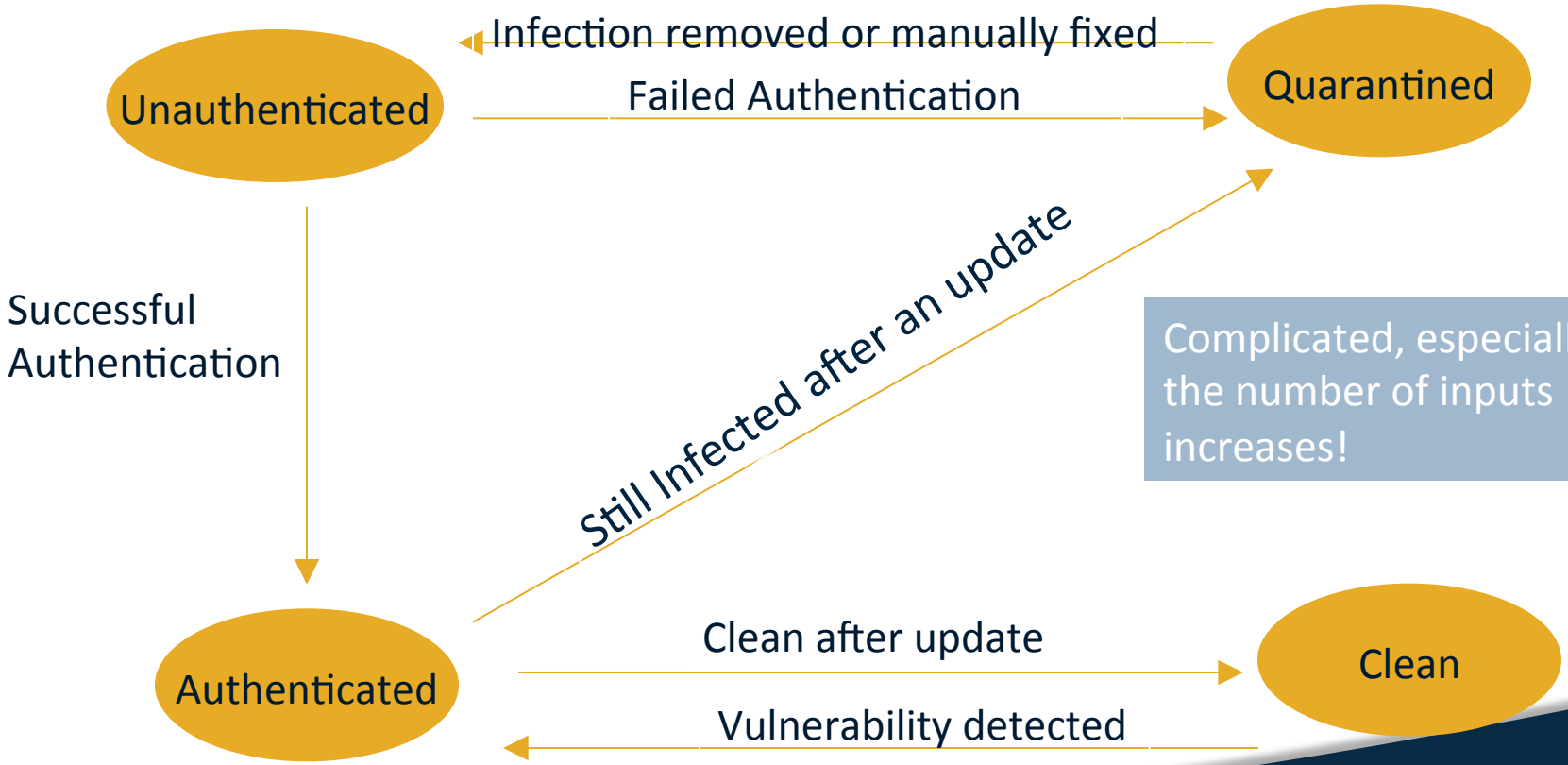


Problems with Current Approach

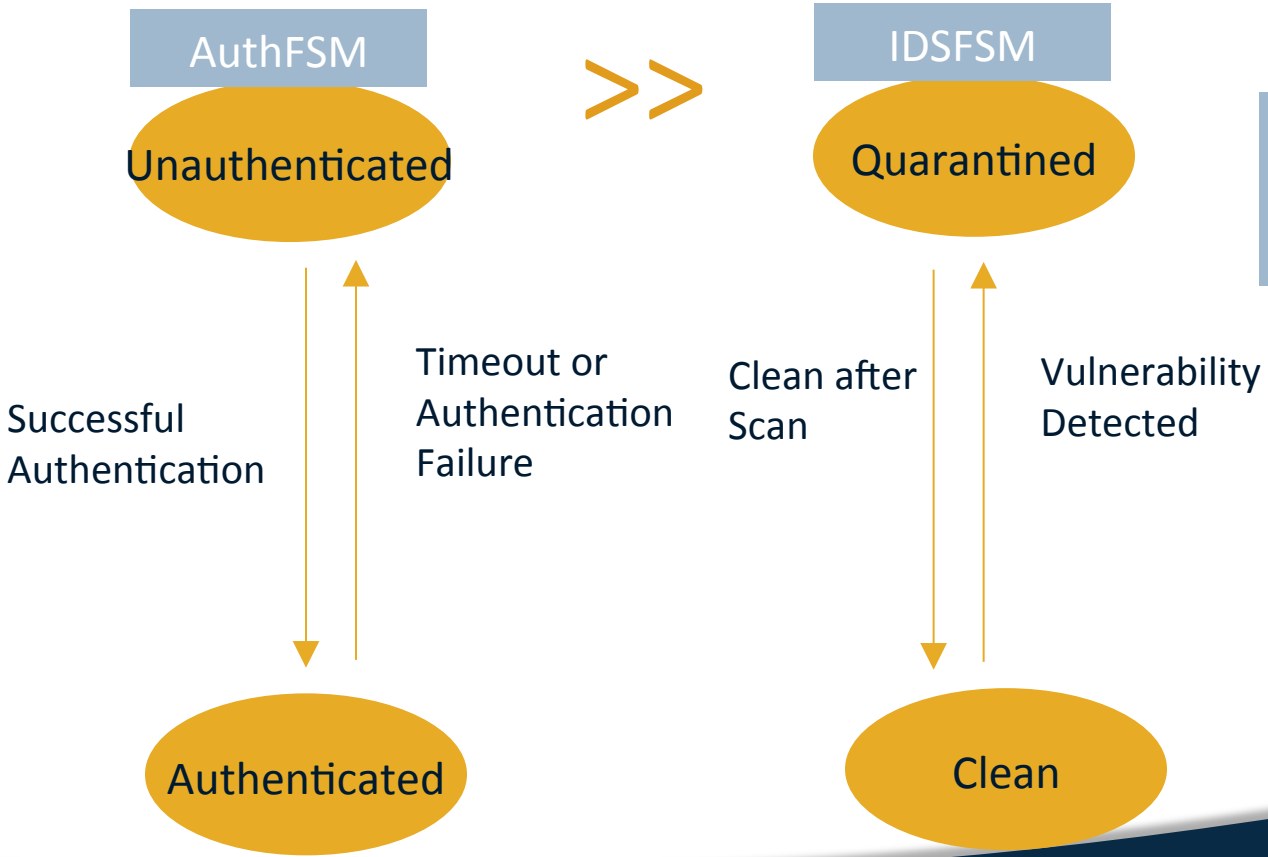
- Access control is **too coarse-grained**
  - Static, inflexible and prone to misconfigurations
  - Need to rely on VLANs to isolate infected machines
- **Cannot dynamically remap** hosts to different portions of the network
  - Needs a DHCP request which for a windows user would mean a reboot
- **Monitoring is not continuous**

Express policies that incorporate network dynamics.

Policy: State Machine, OpenFlow Rules

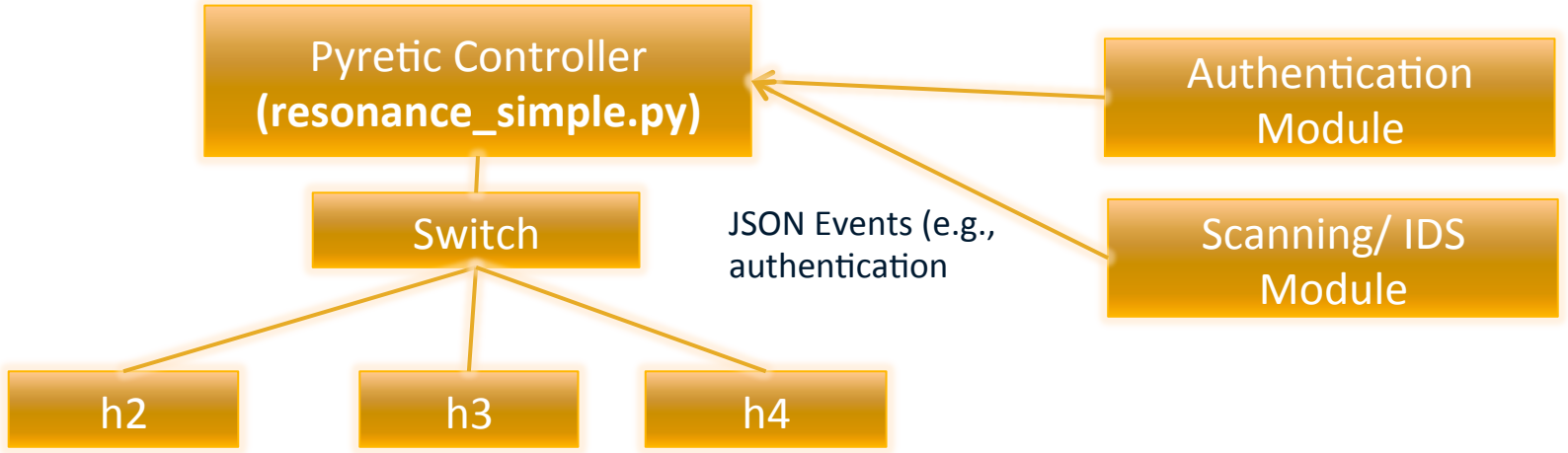


Simpler: Sequential Composition



Simpler: Use Pyretic to sequentially compose FSMs!

Resonance with Sequential Composition



- `$ sudo mn --topo single,3 --mac -arp`
- Default: drop from unauthenticated MAC addresses or vulnerable hosts
- Policy changes once host is authenticated

Summary

- Network configuration and policies must often express what should happen
  - In response to events (security, traffic, etc.)
  - At different times of day
  - For different groups of users
- State machines can help determine what rules are appropriate to install
- Composition keeps FSMs simple