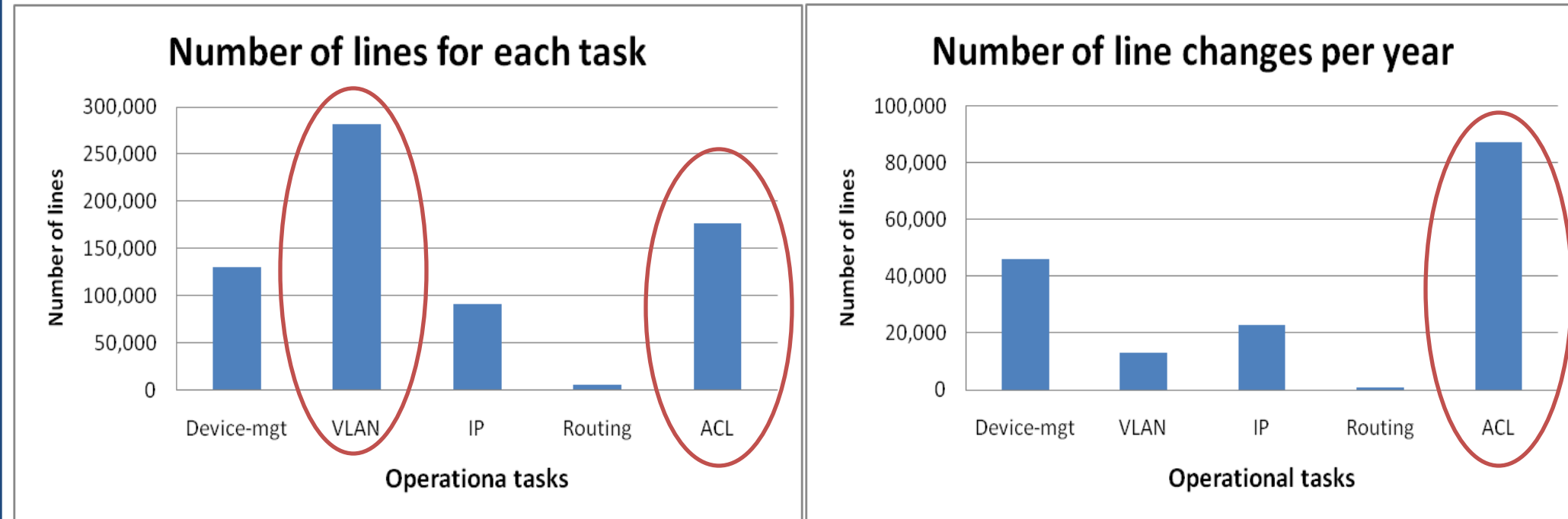
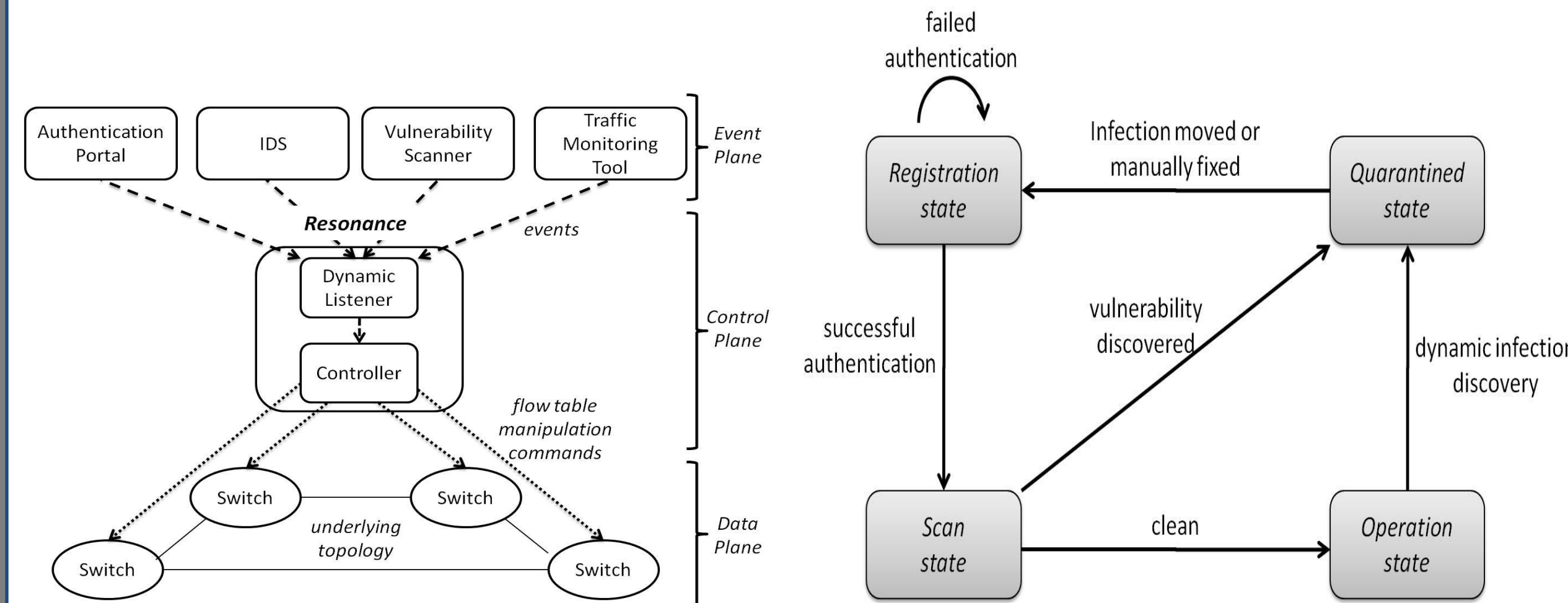


Network Management & Configuration Today



- **A network is built with:**
 - Thousands of interconnected devices
 - Each device configured individually
 - Low-level CLI commands
 - Extensive usage of VLANs
 - **A network changes:**
 - On a daily basis
 - With lots of changes in policy
- ➔ *Policy is spread out and enforced by multiple devices, which themselves change on a daily basis.*

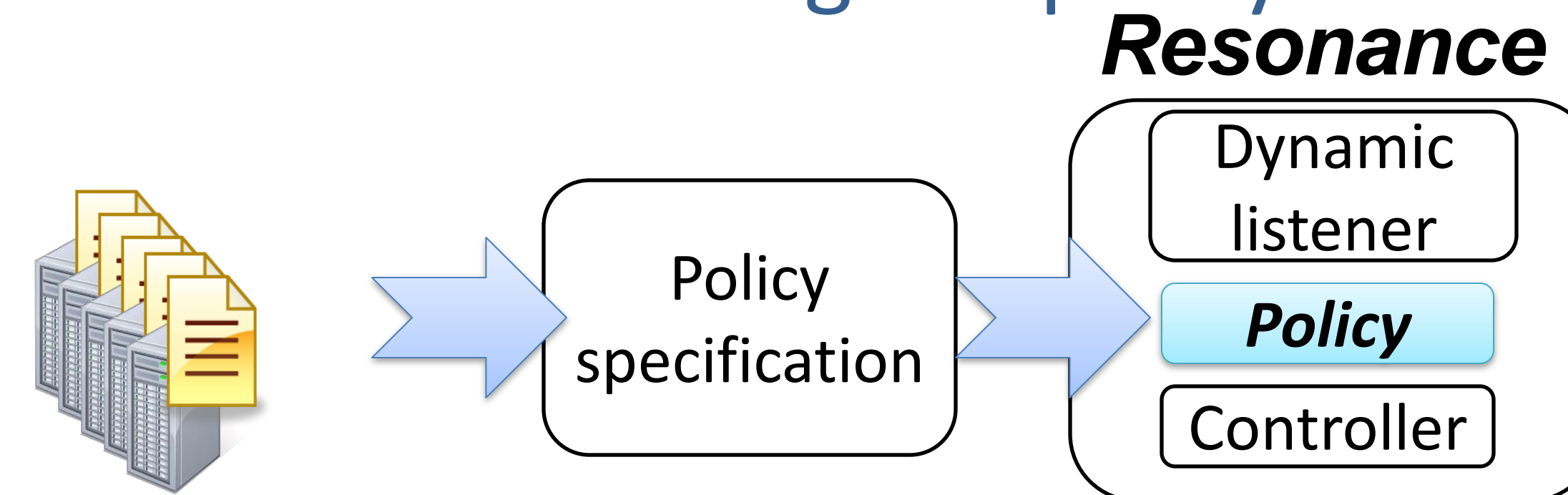
Resonance OpenFlow-based Network Management



- Central controller for traffic management
- Dynamic listener to react to events
- State machine model and security classes for role-based access control and traffic manipulation

Policy Language

Rewriting the policy

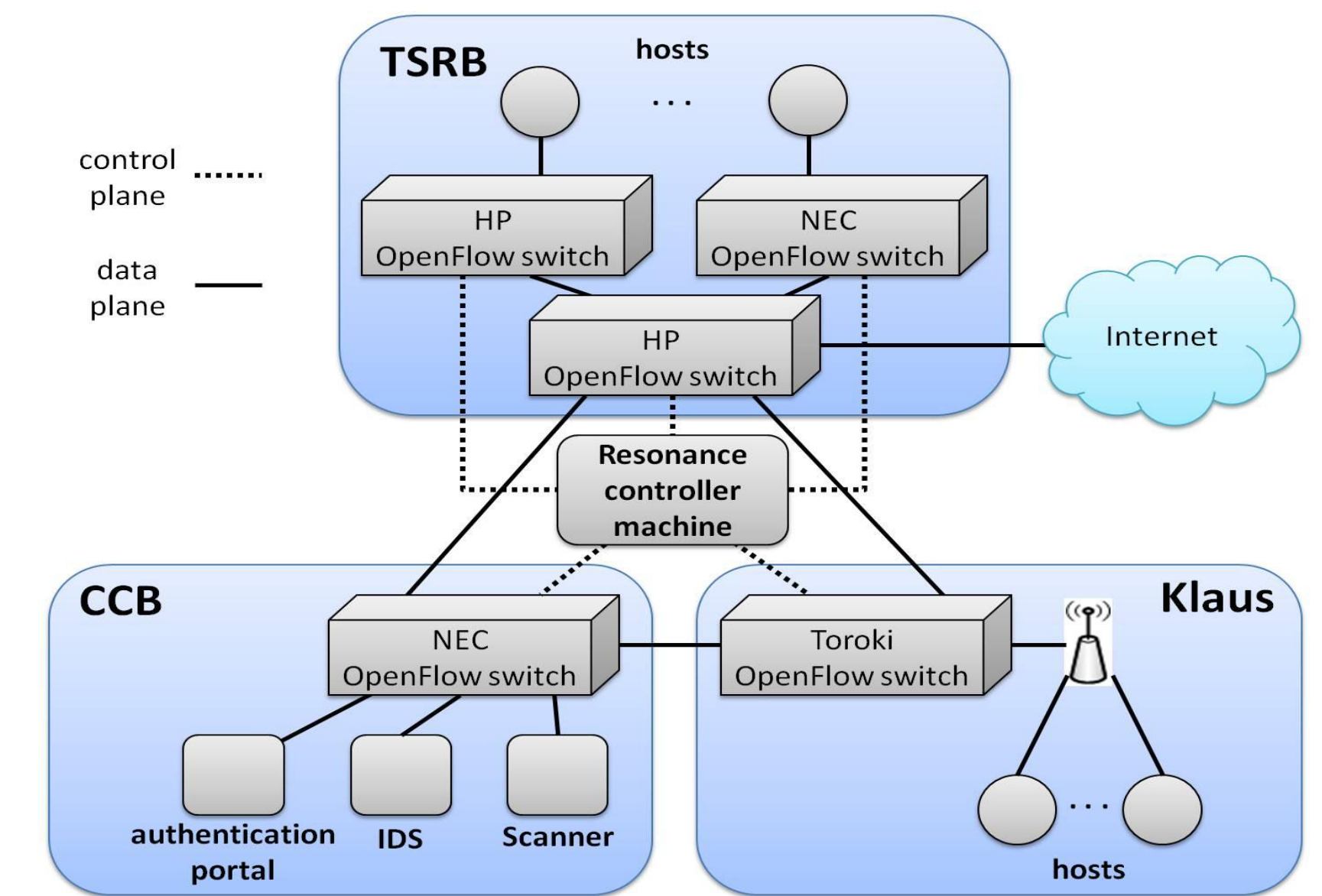


- Simple
- Flexible
- Queryable
- Centralized
- Dynamic

```

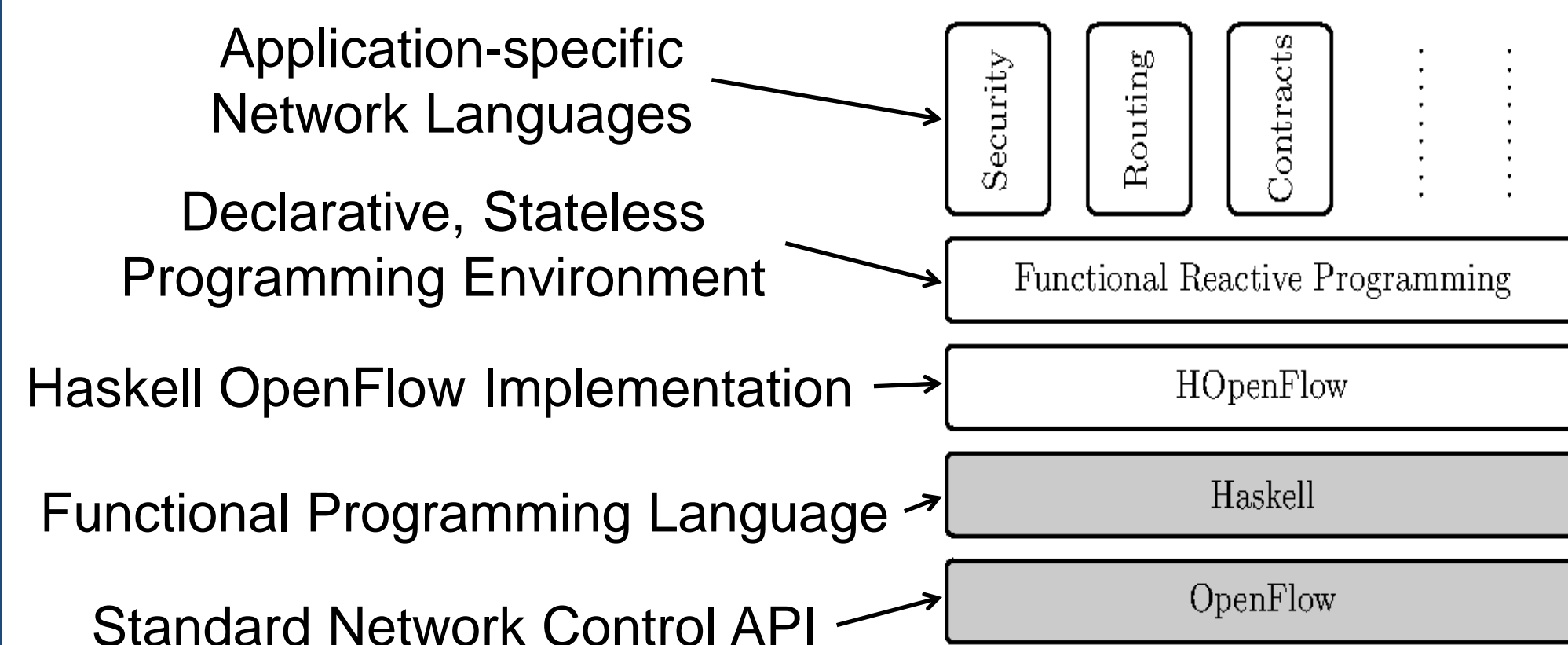
policy1 = []
policy2 = [[]]
policy3 = [[InPortIs 18]]
policy4 = [[InPortIs 17, SenderMacIs (ethAddress 1 2 3 4 5 6)]]
policy5 = [[InPortIs 17, SenderIPIn (ipAddress 1 1 0 0 //17)]]
policy6 = [[InPortIs 17, SenderIPIn (ipAddress 1 1 0 0 //8)]]
policy7 = [[InPortIs 17, SenderIPIn (ipAddress 1 1 0 0 //8), SenderAuthenticated]]
policy8 = [[InPortIs 17, SenderIPIn (ipAddress 1 1 0 0 //8), SenderAuthenticated, SenderRole admin]]
policy9 = [[SenderAuthenticated, SenderRole guest, TimeBetween 9 17]]
policy10 = [[InPortIs 17, SenderMacIs (ethAddress 1 2 3 4 5 6), SenderAuthenticated, SenderRole guest, TimeBetween 9 17]]
policy11 = [[SenderRole guest], [SenderRole admin]]
policy12 = [[TimeBetween 8 9], [TimeBetween 9 10]]
    
```

Campus Deployment



- **Future Expansion**
 - Testbed spanning three campus buildings
 - Push for production network
 - Support multiple projects
- **Current Resources**
 - 17 OpenFlow-enabled switches
 - 2 ProtoGENI nodes
 - 4 PlanetLab nodes

Nettle: Domain-Specific Languages for Network Control



Future Plans

- Reimplement & enforce policy for an deployed subnet
- Test on multi-campus scenario
- Improve scalability
- Policy language for routing
- Improve fault tolerance