Policy and Resource Orchestration in Software defined Networks

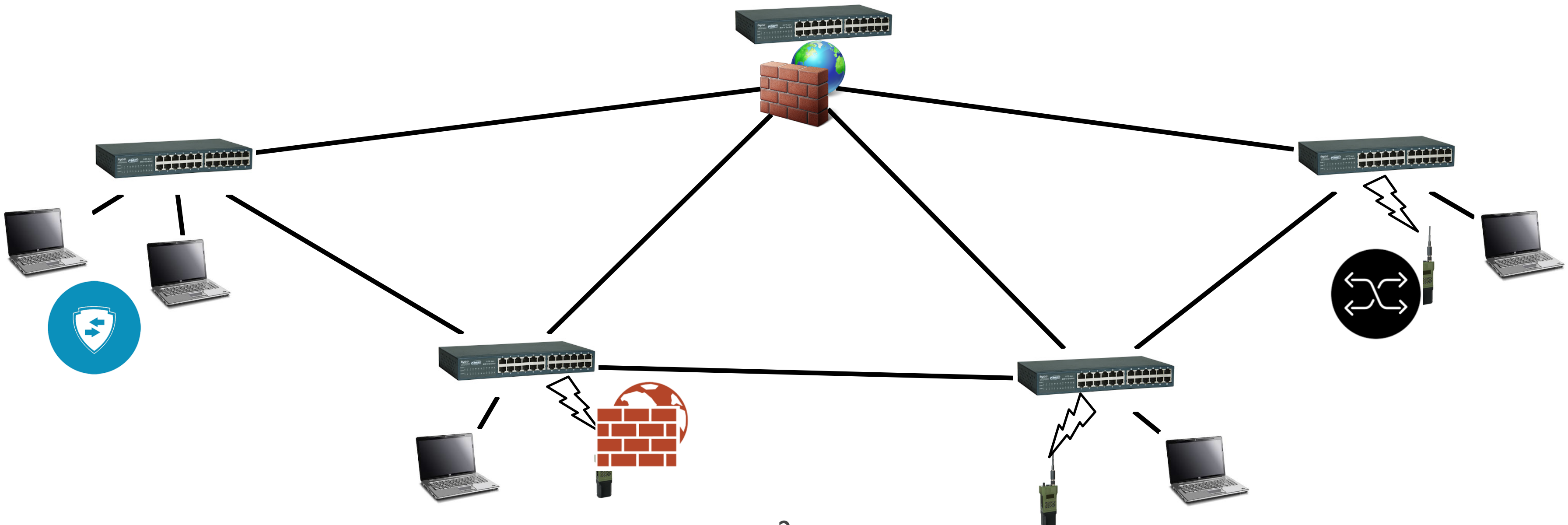
Anduo Wang
adw@temple.edu

Jie Wu
jiewu@temple.edu

Temple University

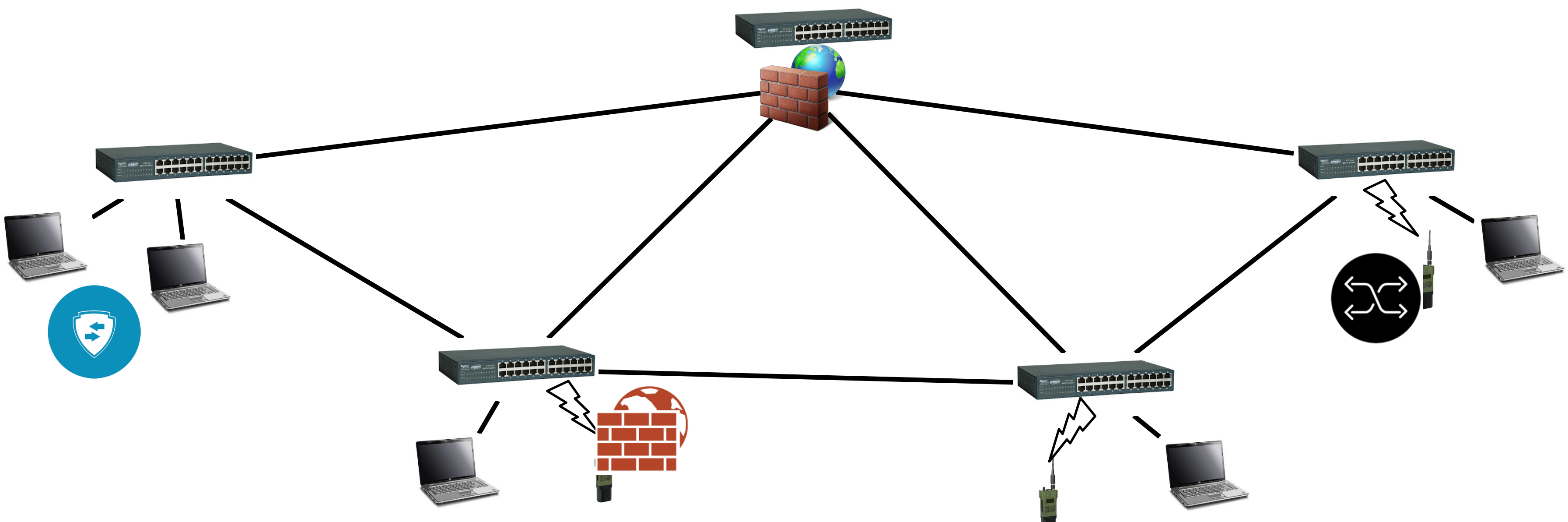
October 19, 2018 CIC'18

traditional network management

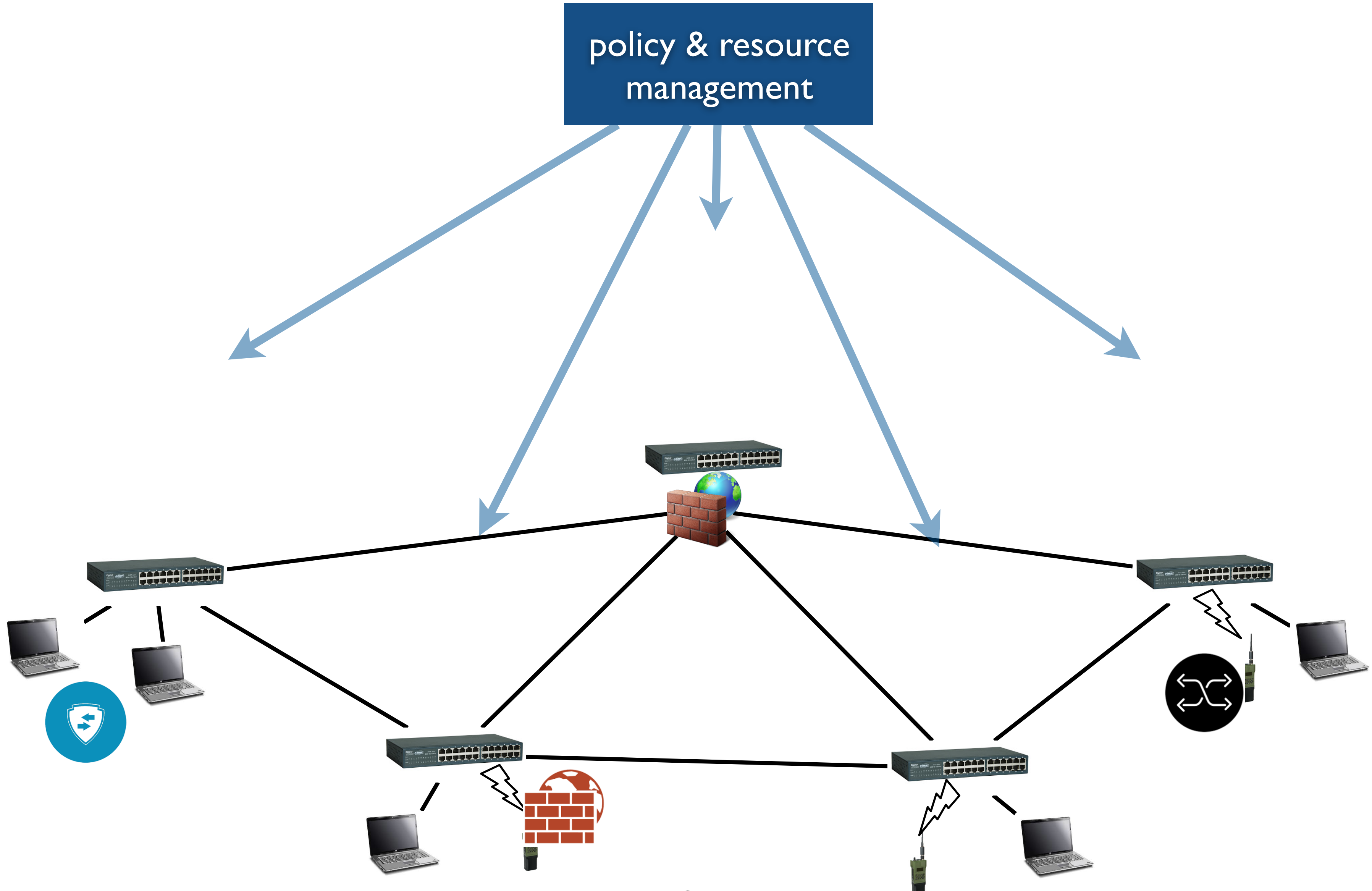


traditional network management

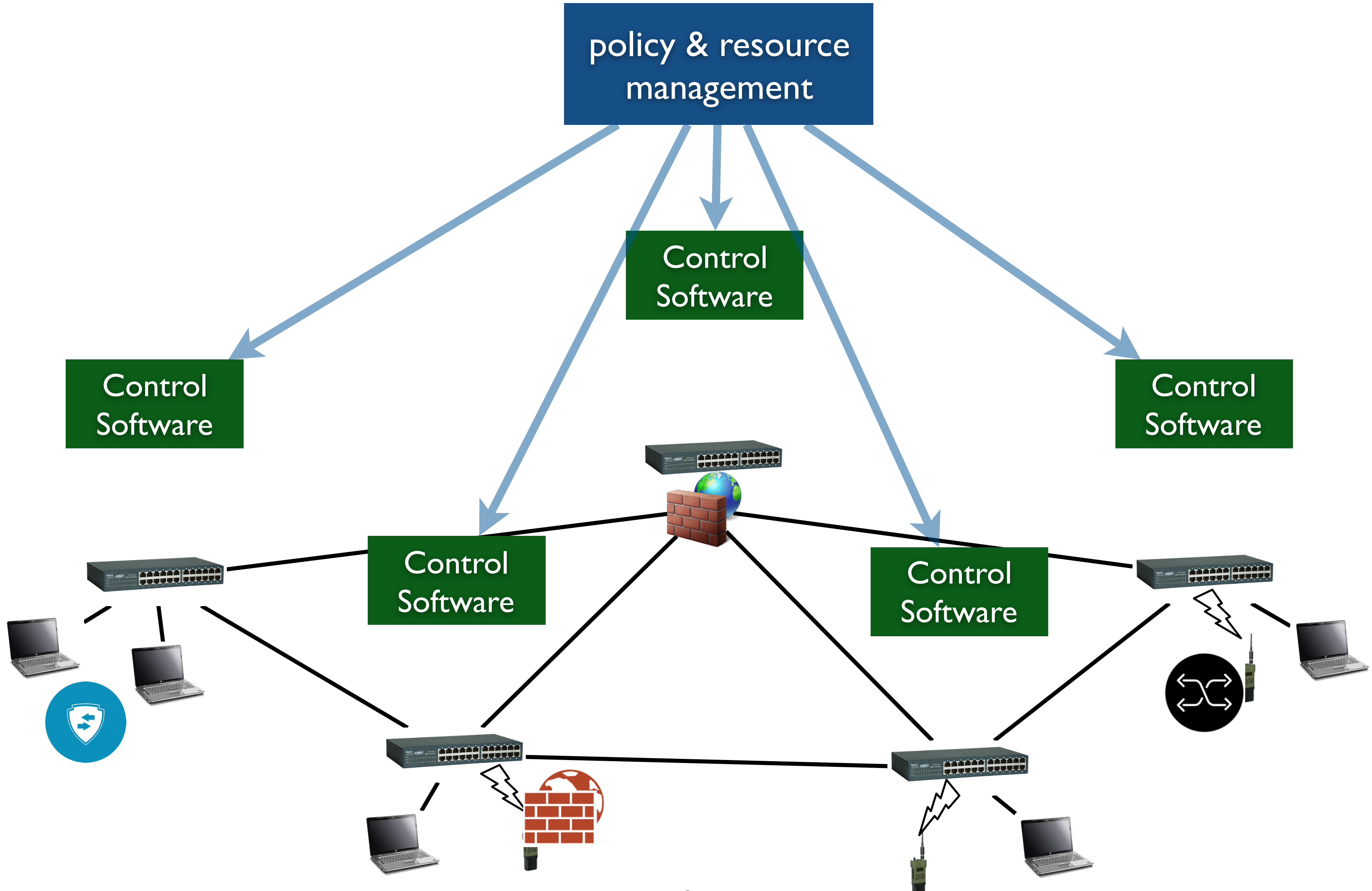
policy & resource management



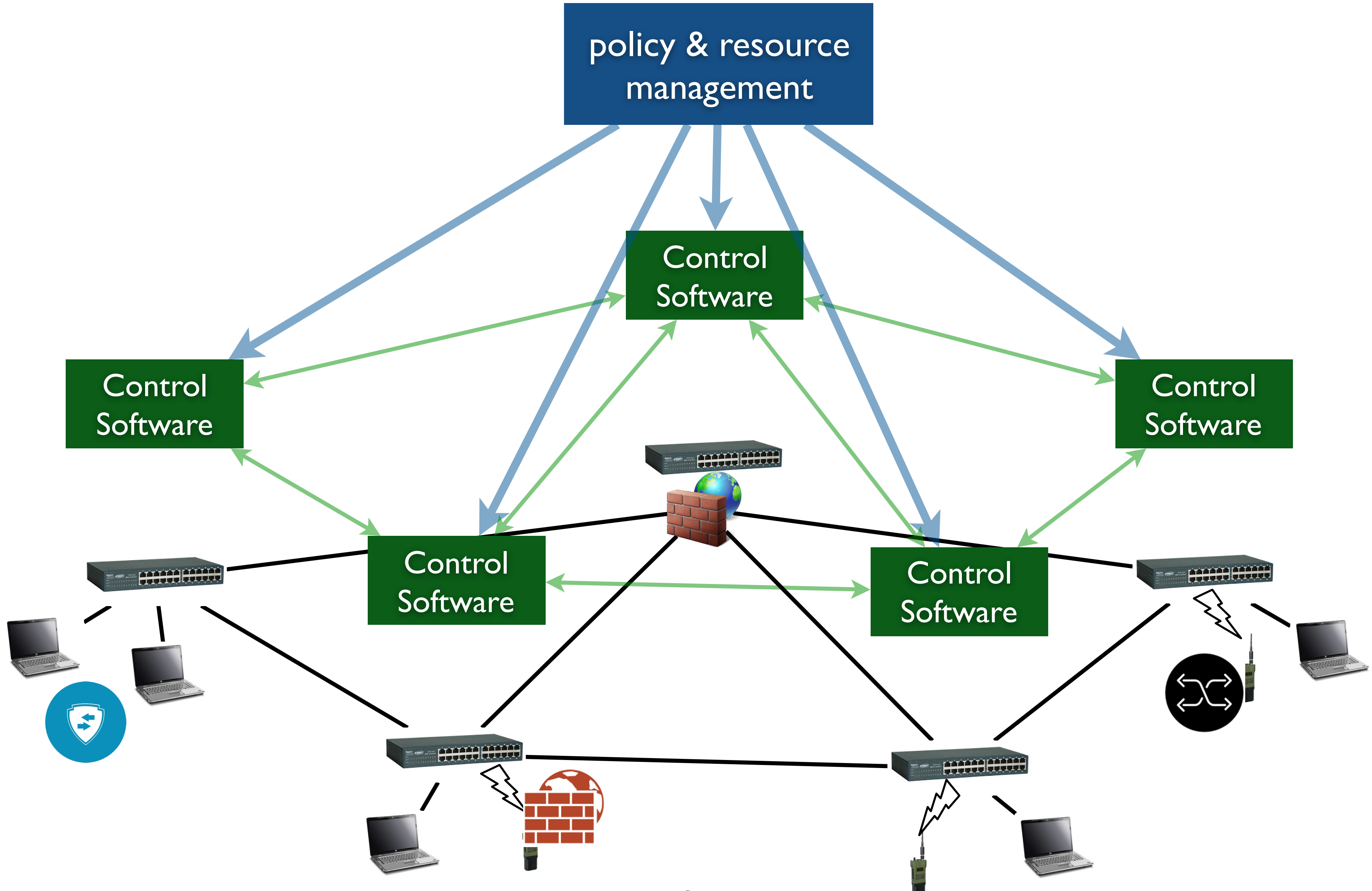
traditional network management



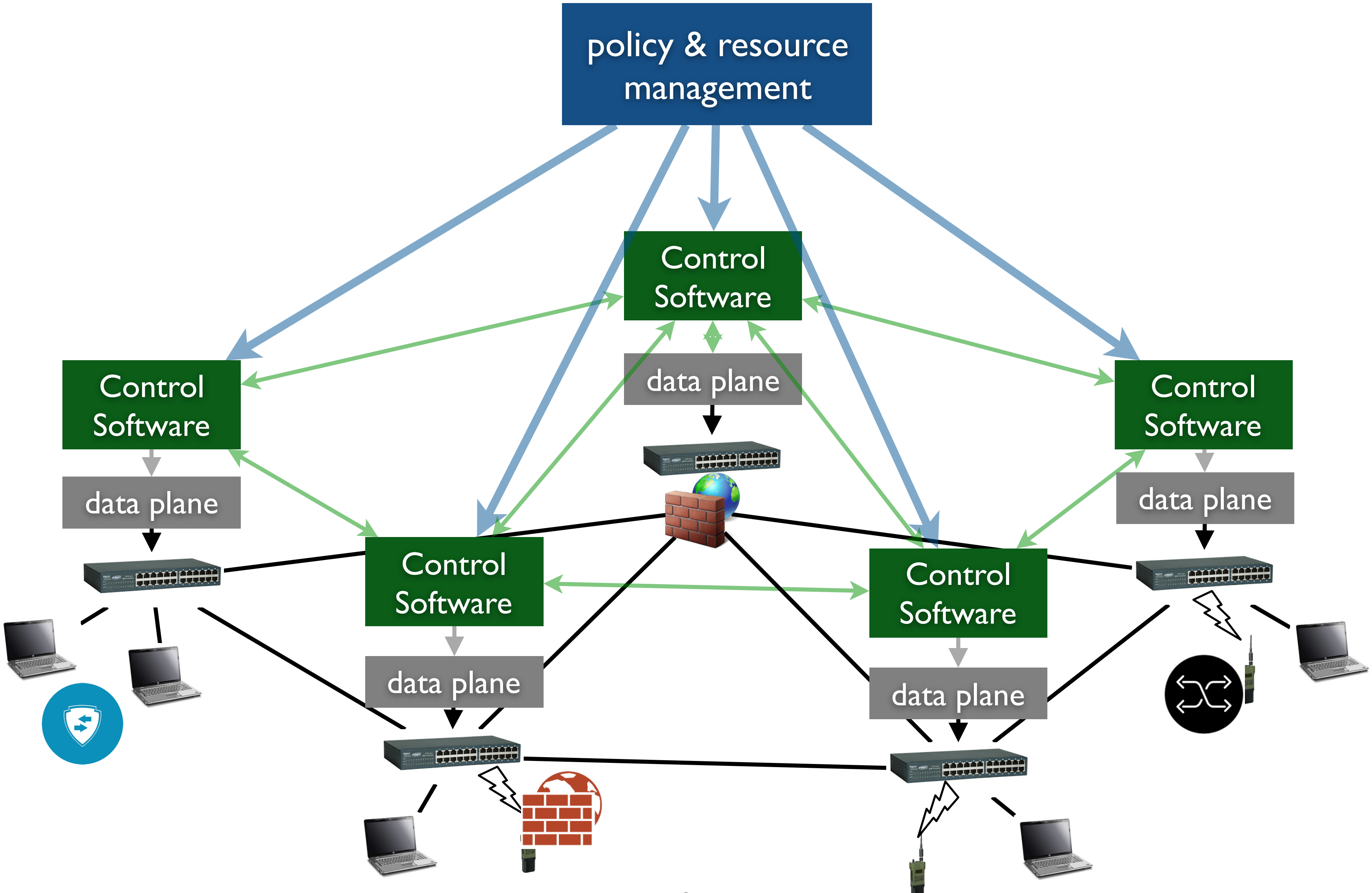
traditional network management



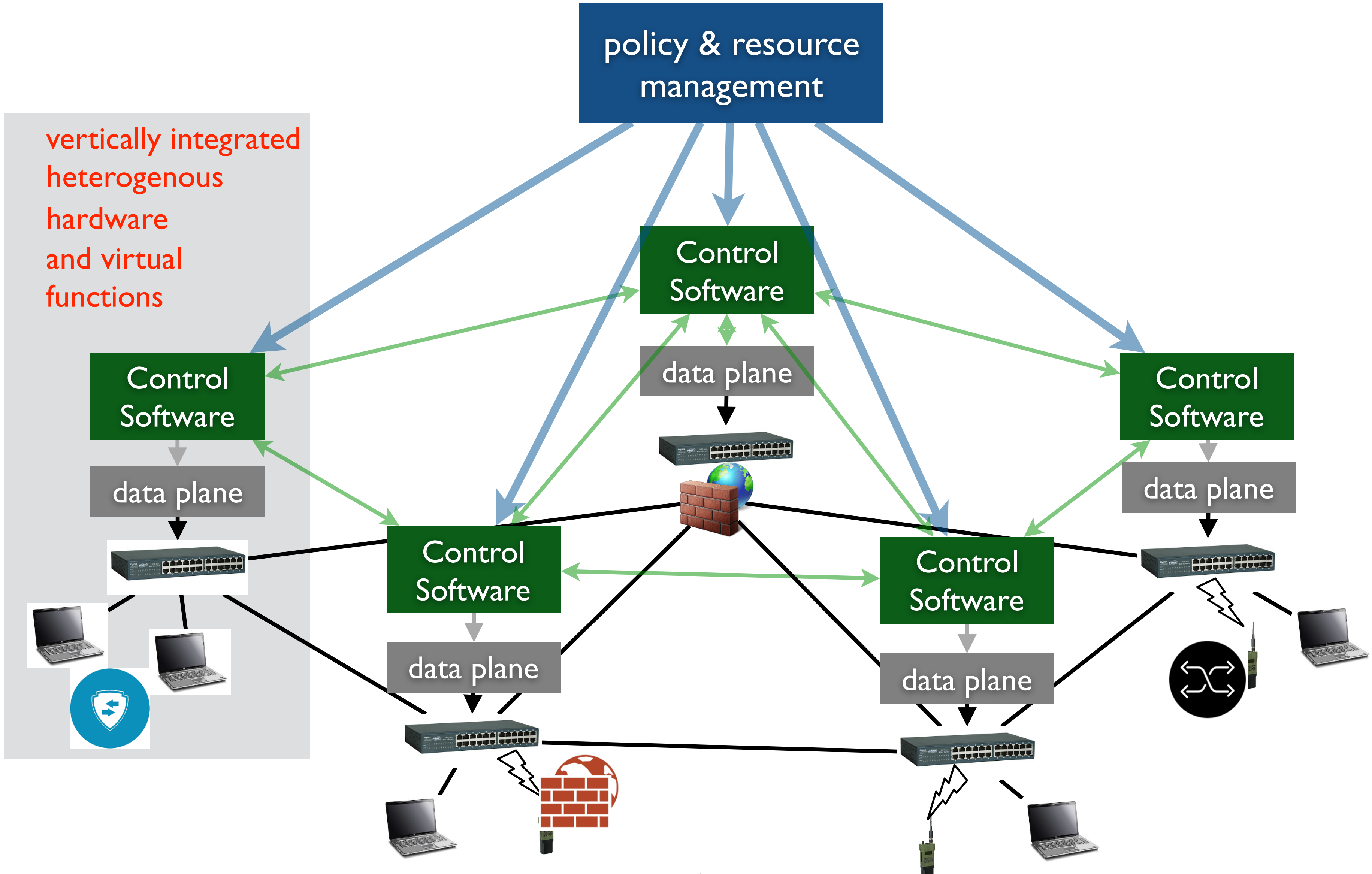
traditional network management



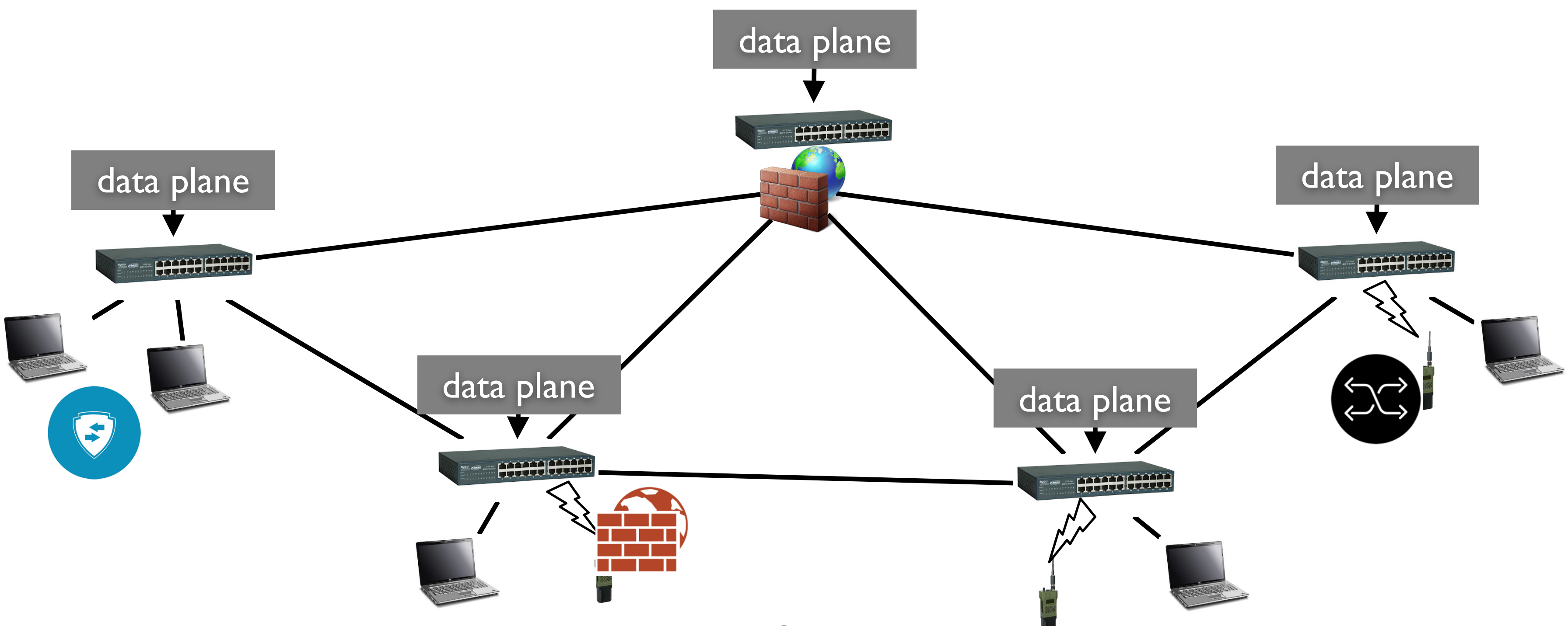
traditional network management



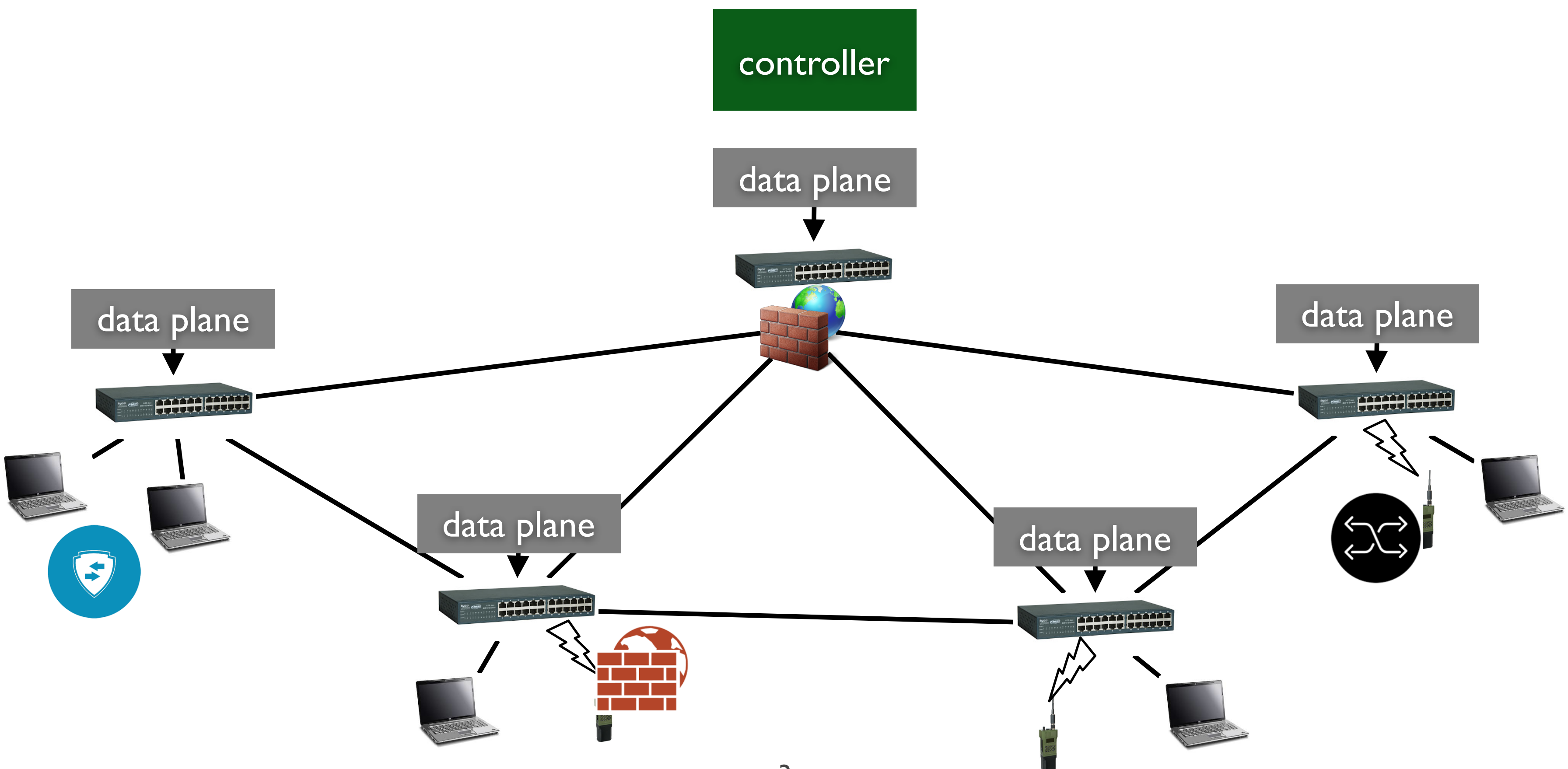
traditional network management



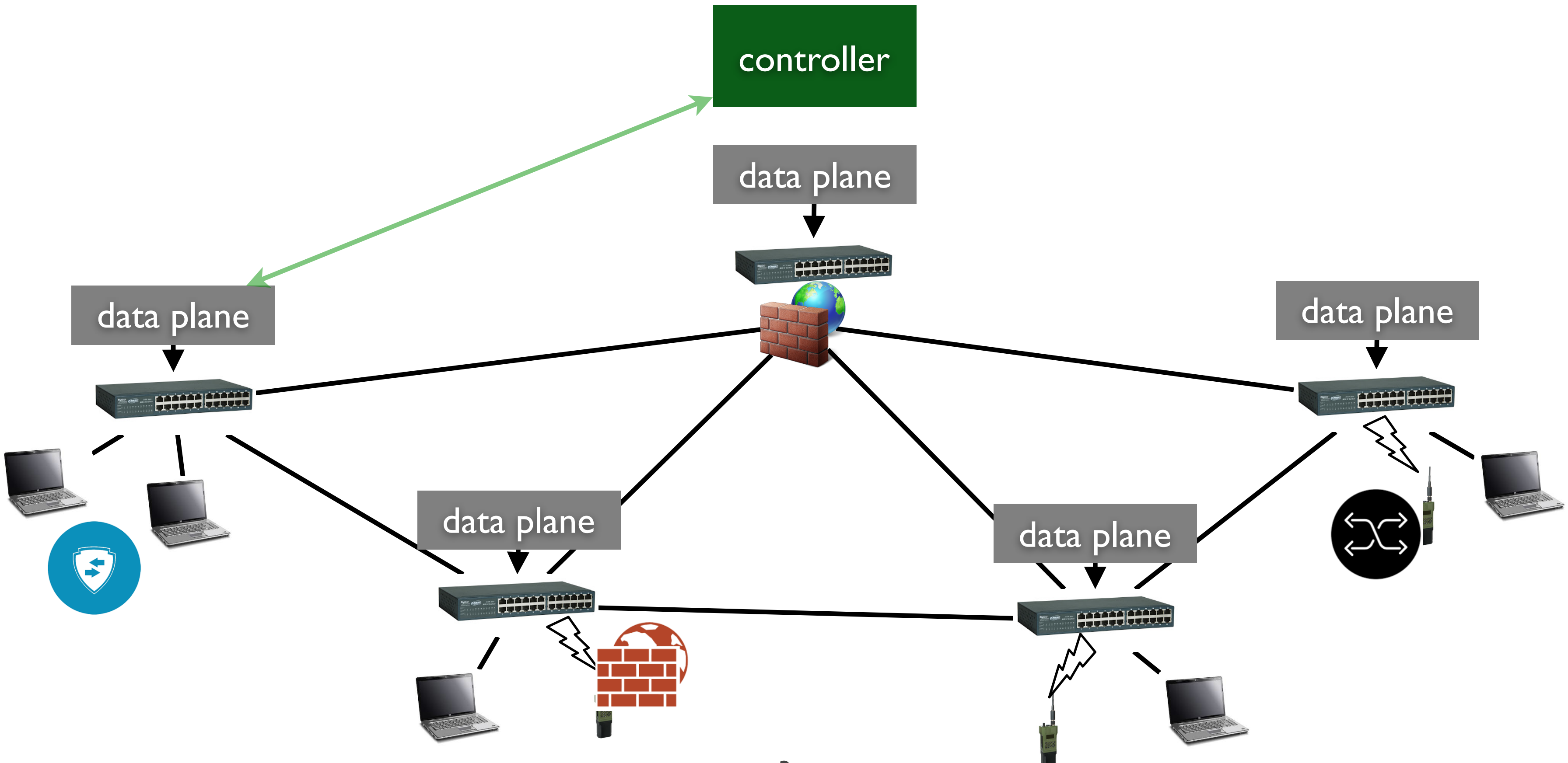
software-defined networking (SDN)



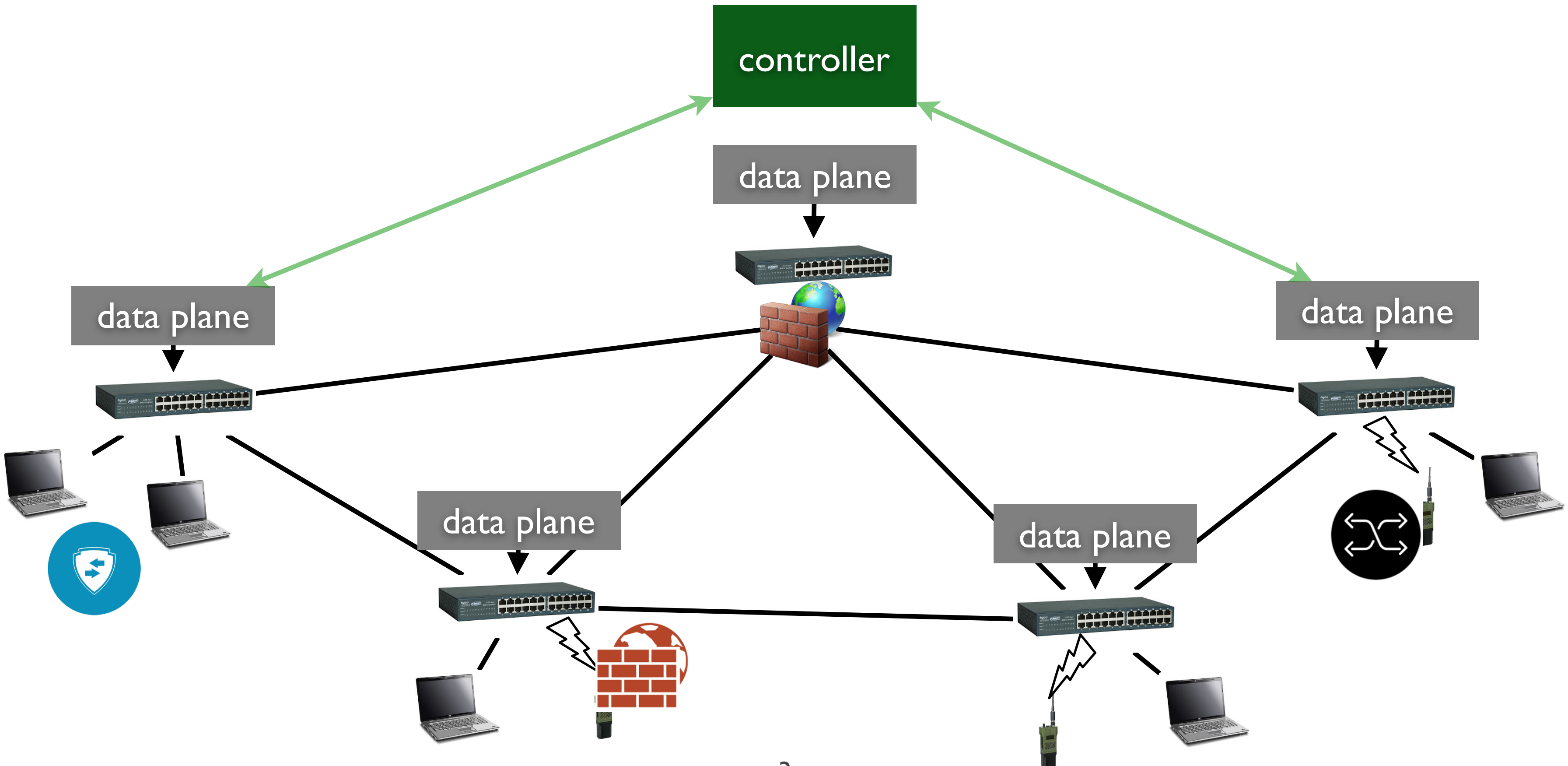
software-defined networking (SDN)



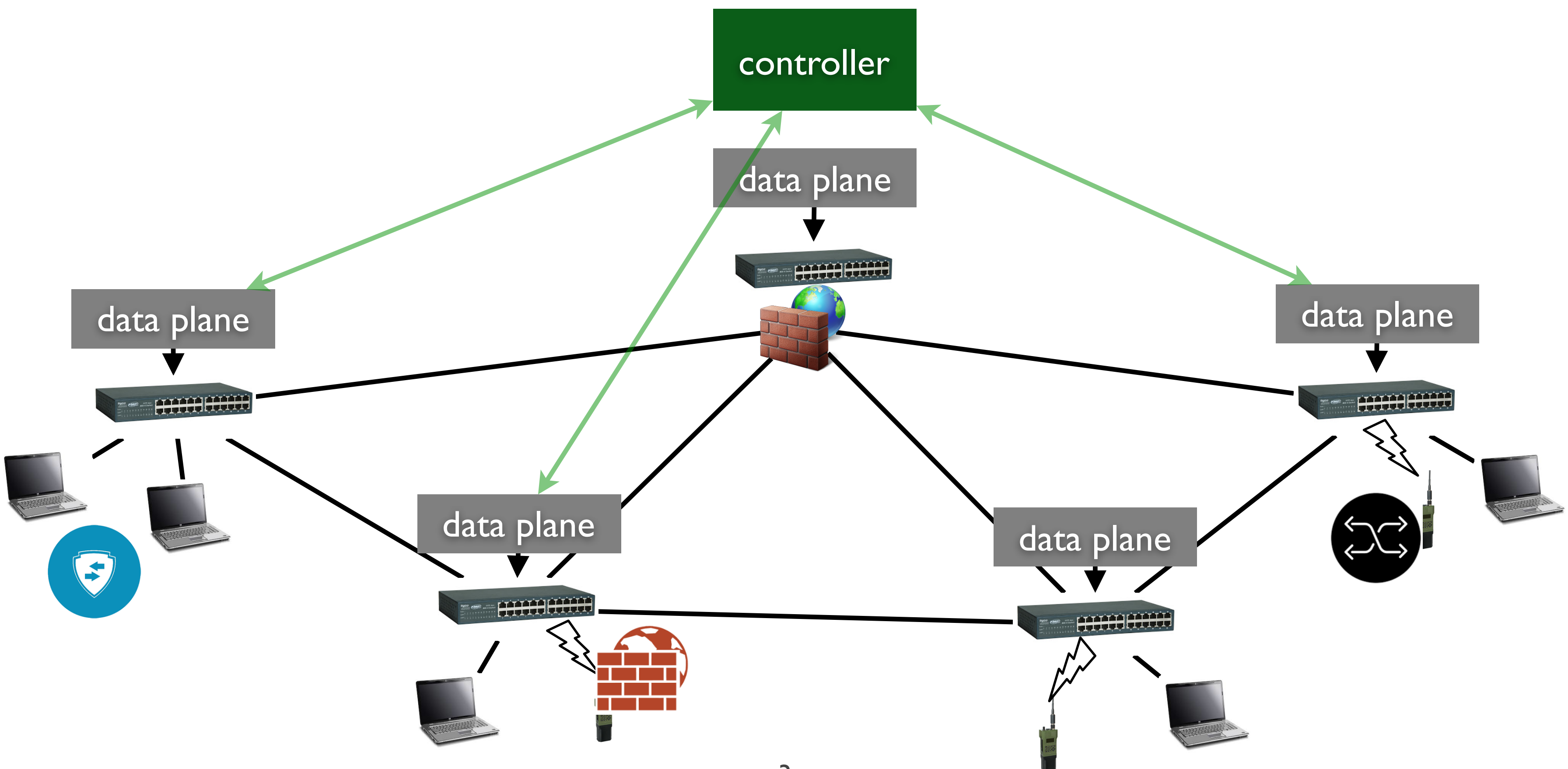
software-defined networking (SDN)



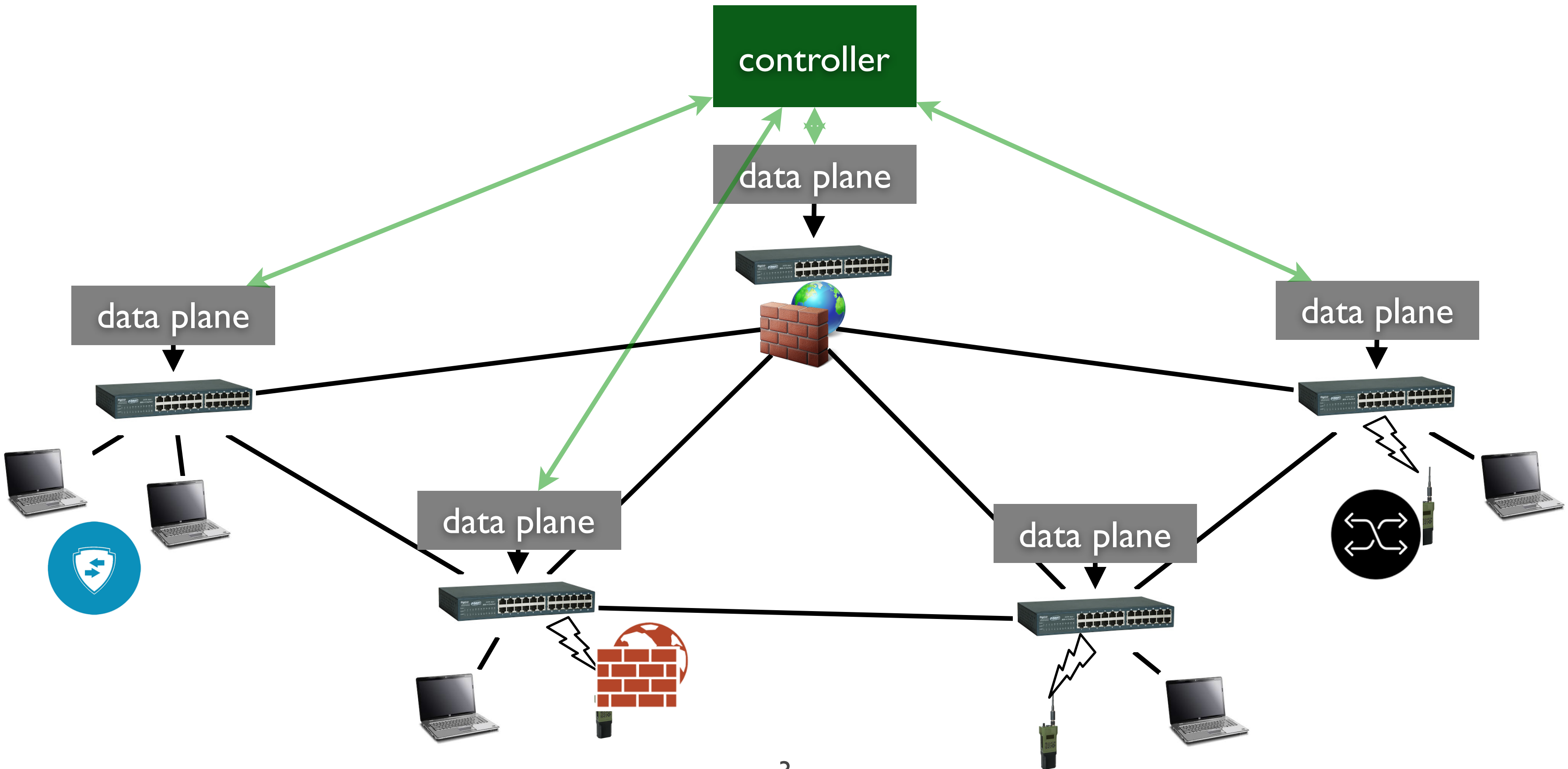
software-defined networking (SDN)



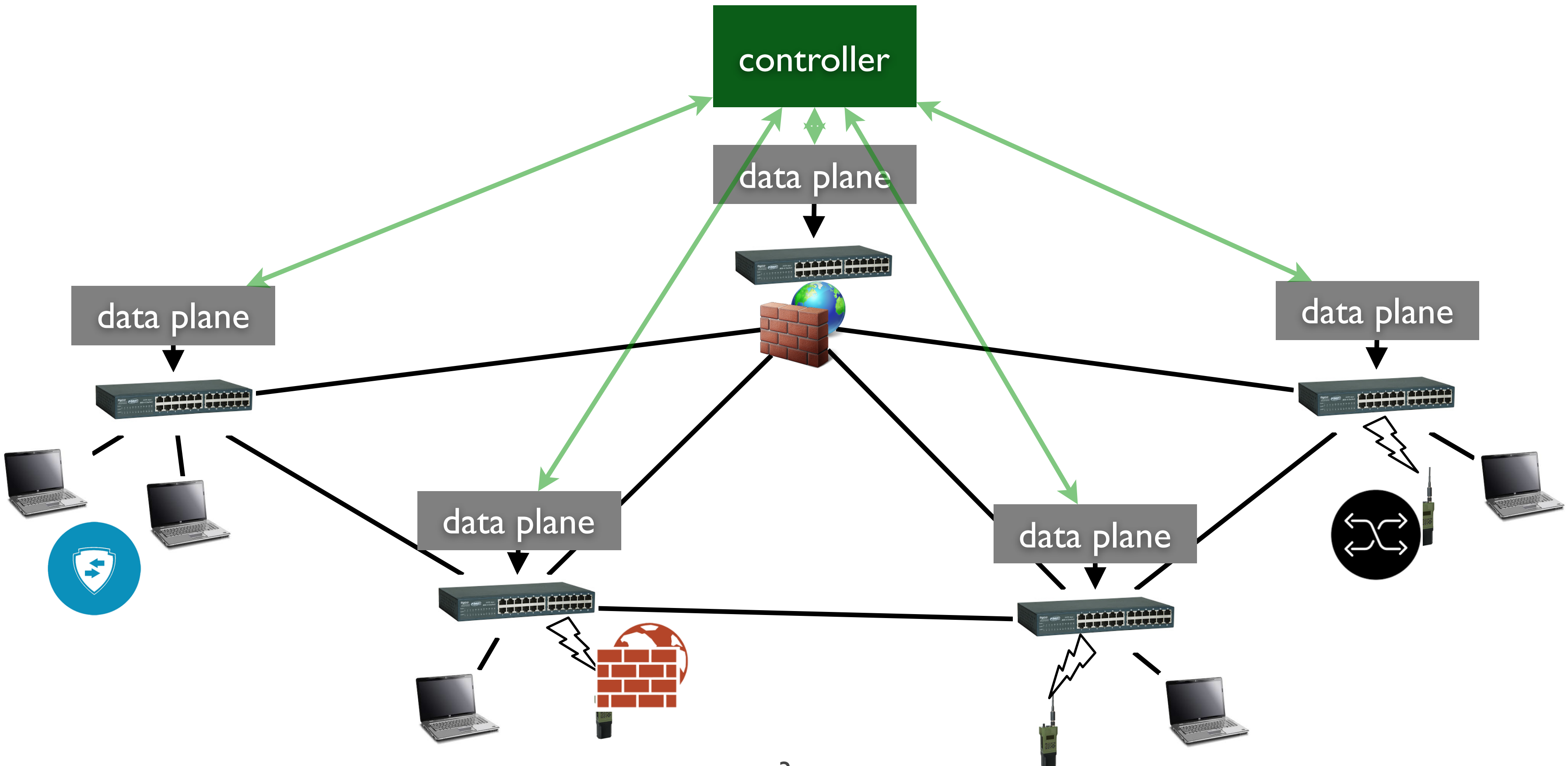
software-defined networking (SDN)



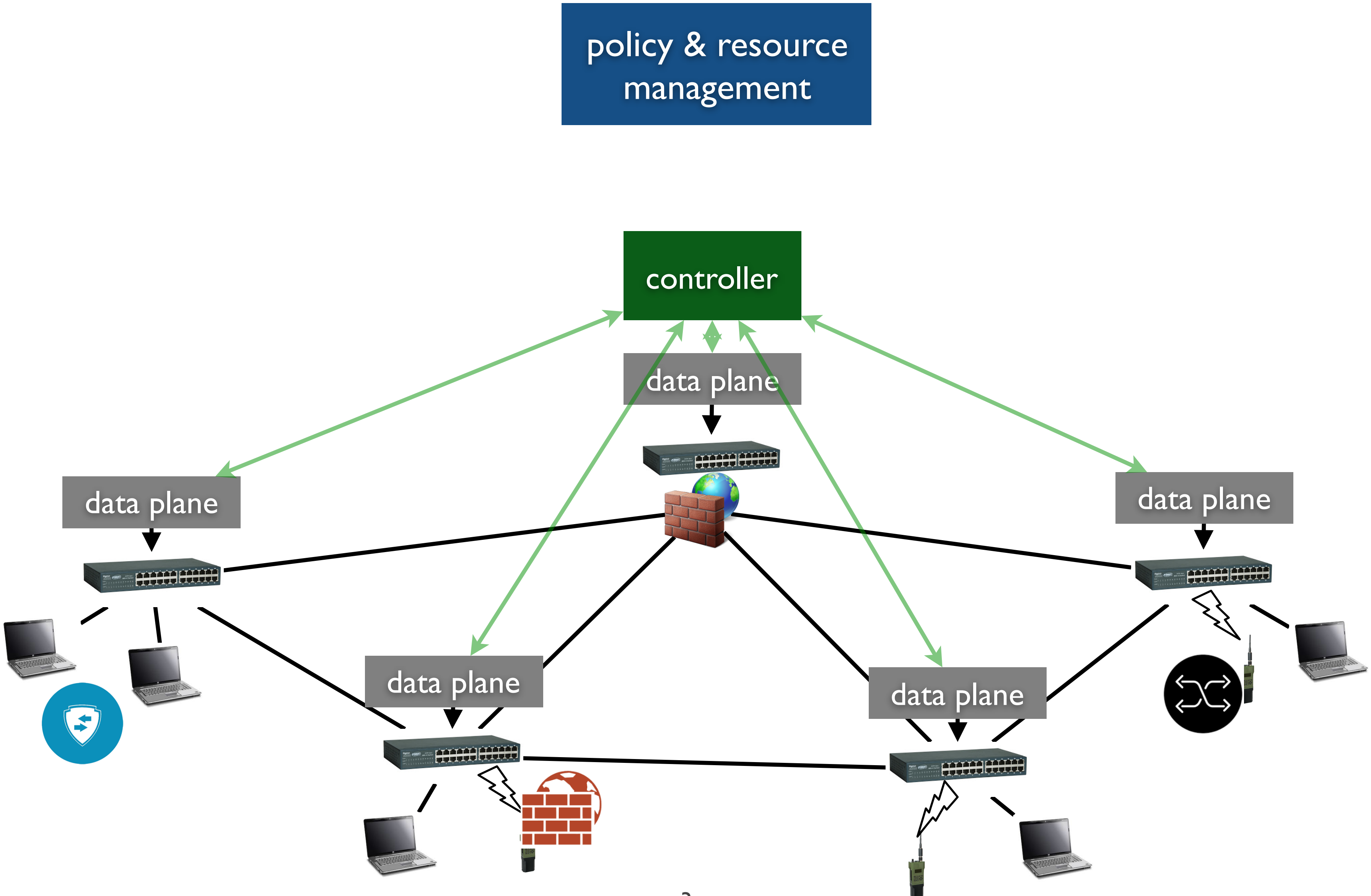
software-defined networking (SDN)



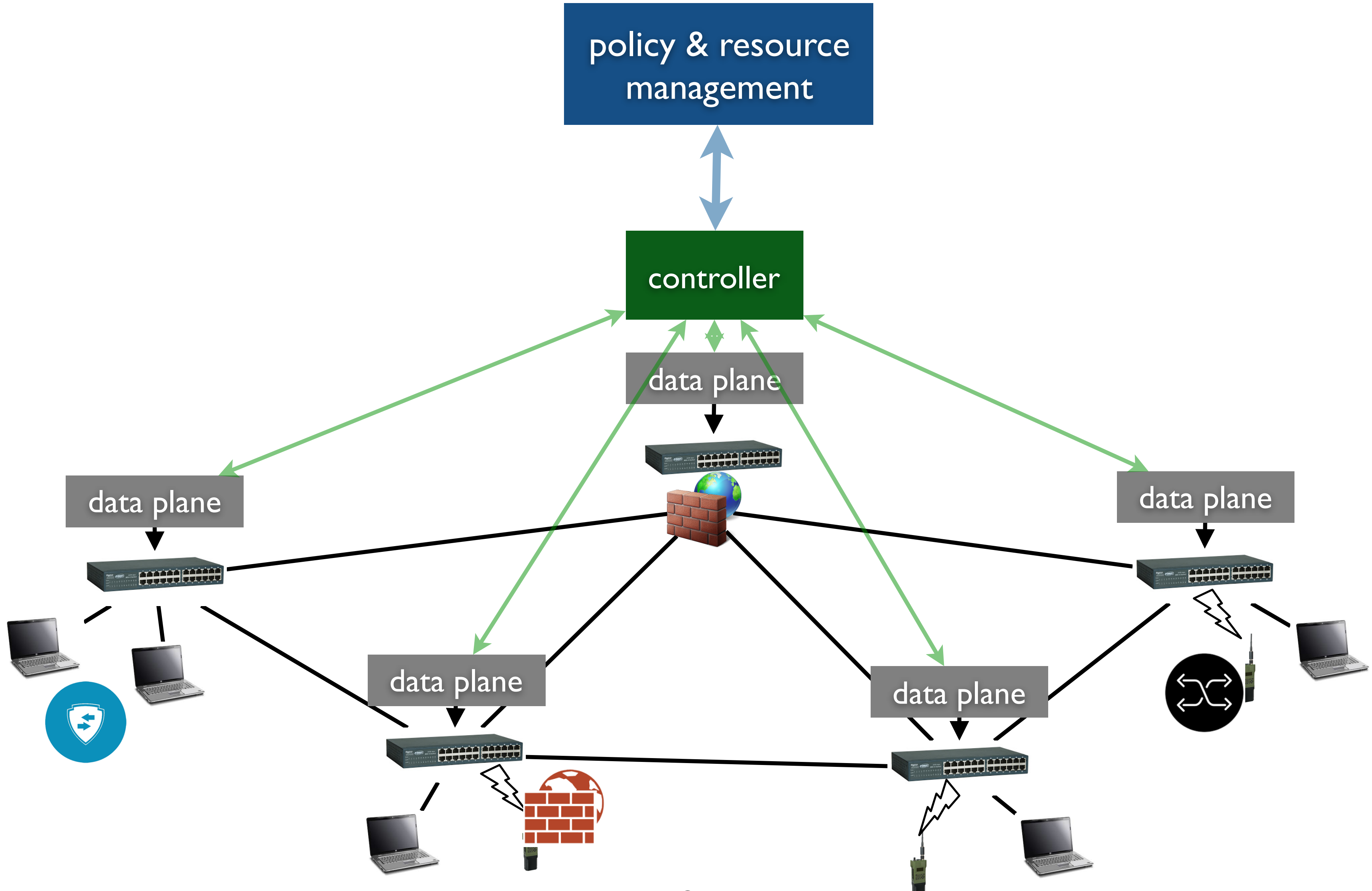
software-defined networking (SDN)



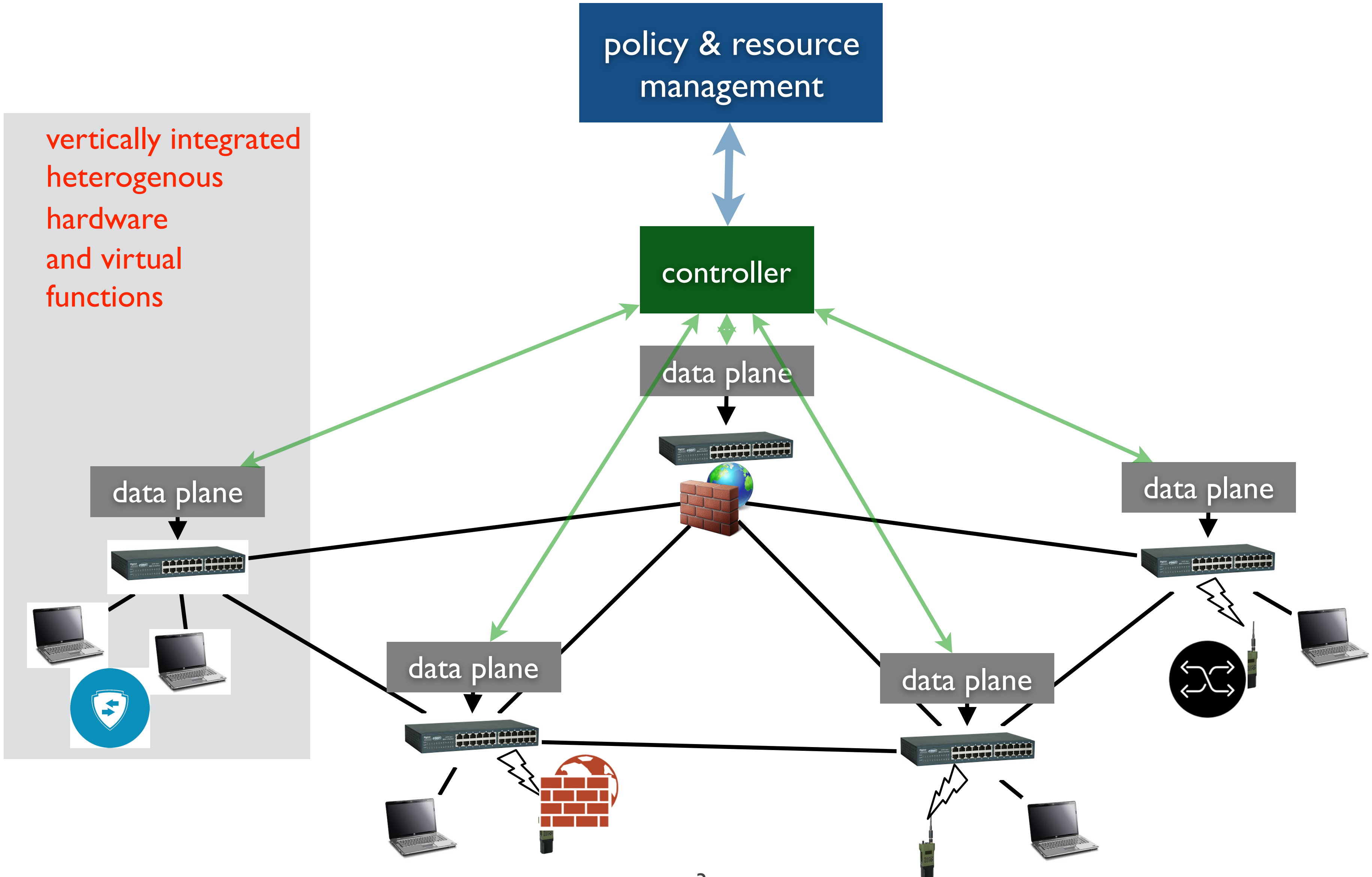
software-defined networking (SDN)



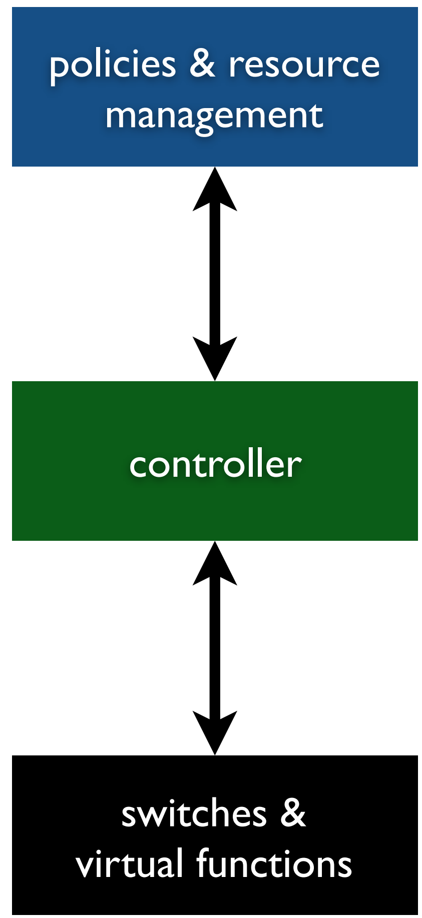
software-defined networking (SDN)



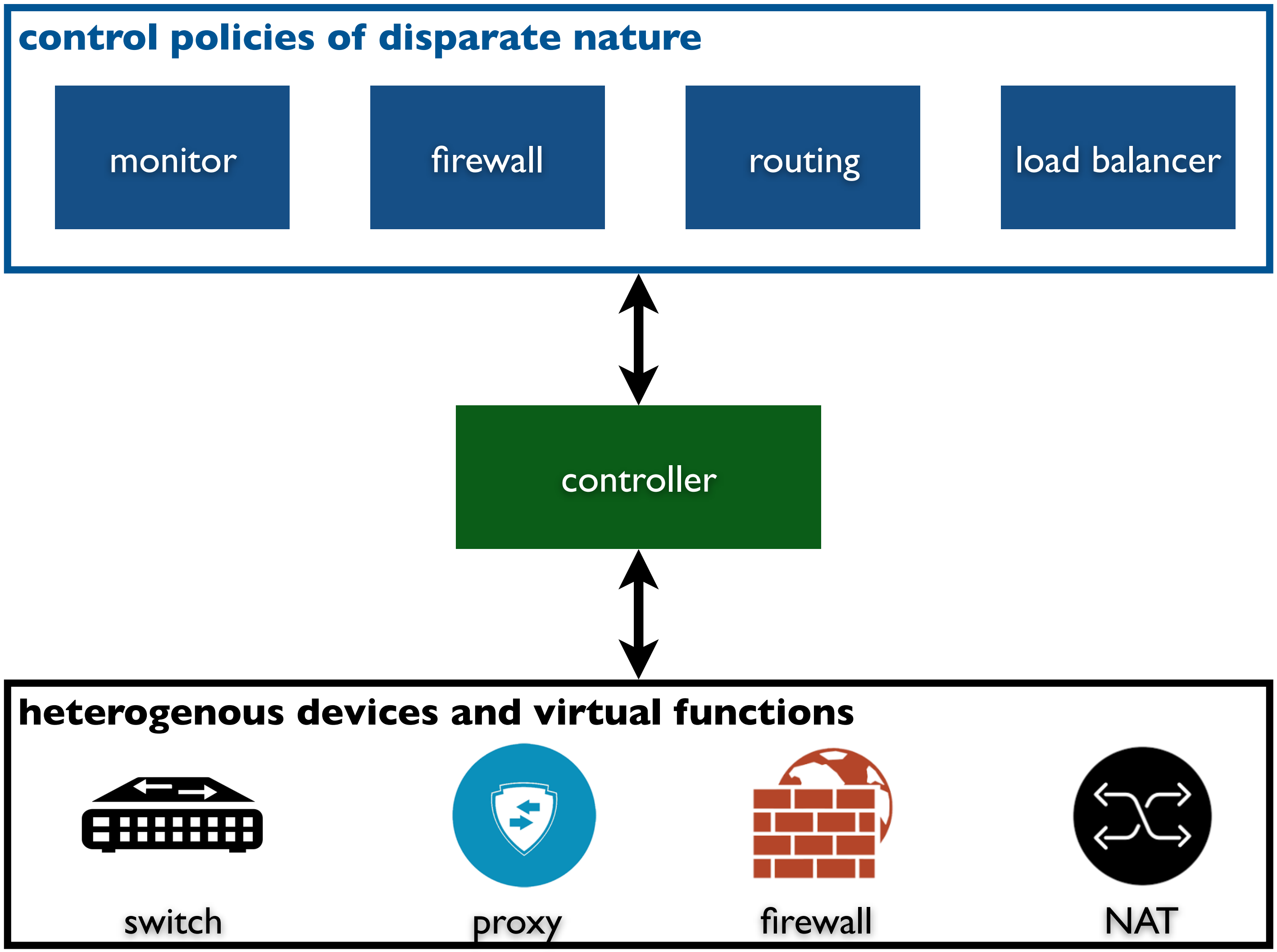
software-defined networking (SDN)



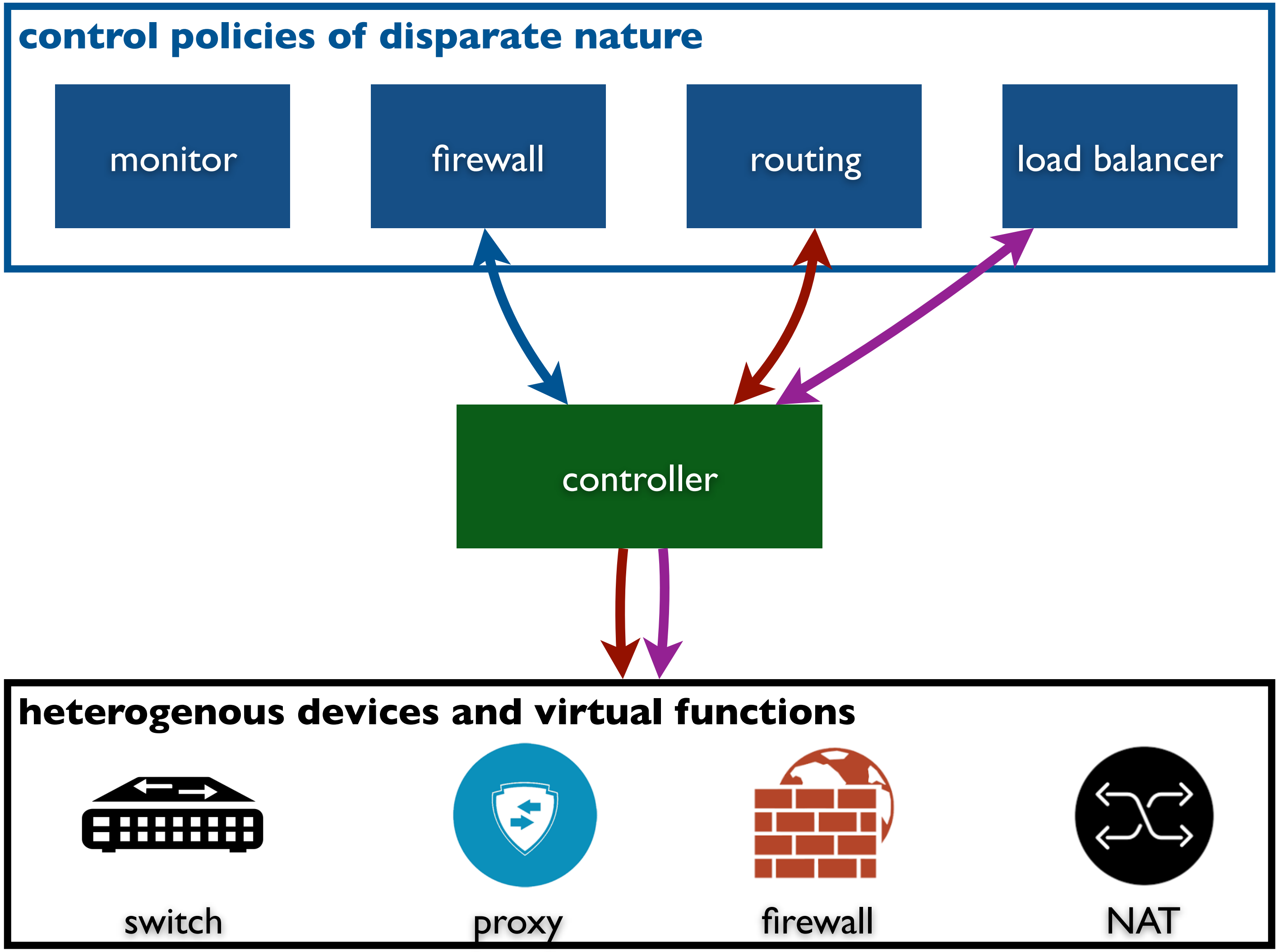
orchestrating policies & resources in SDN



orchestrating policies & resources in SDN



policy orchestration



policy orchestration

today, the onus of coordinating SDN policies falls on the admin to write modular control application

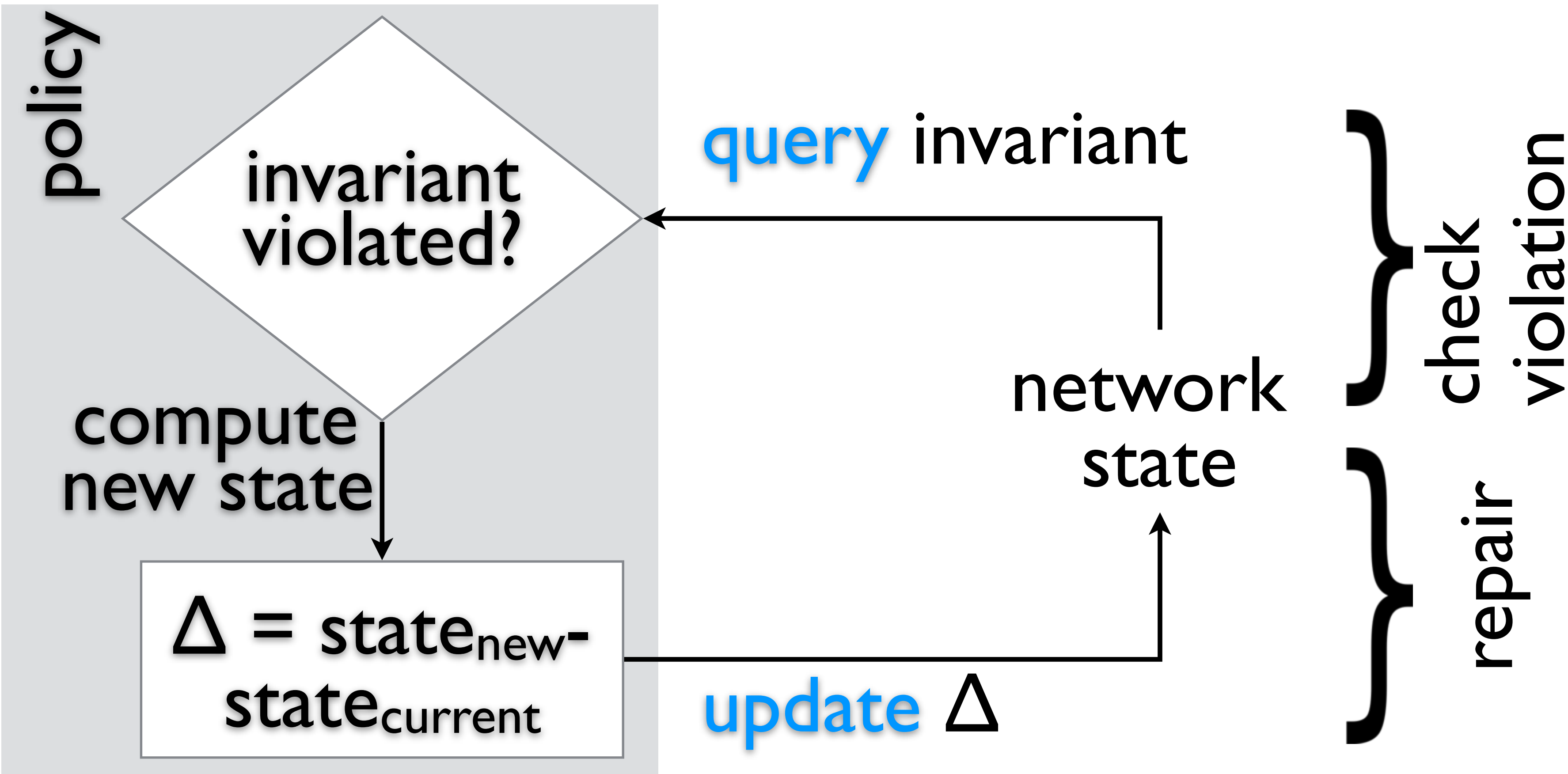
- policy prefixed in specific controller program — syntax varies from one domain specific language to another
- manual composition of controller programs relies on the internalized knowledge of experienced admin

our approach

- orchestration as a controller primitive
- policy as semantic units that maintain properties
- automating policy coordination by logical reasoning about network properties

a semantic model

model SDN policies as data query/update



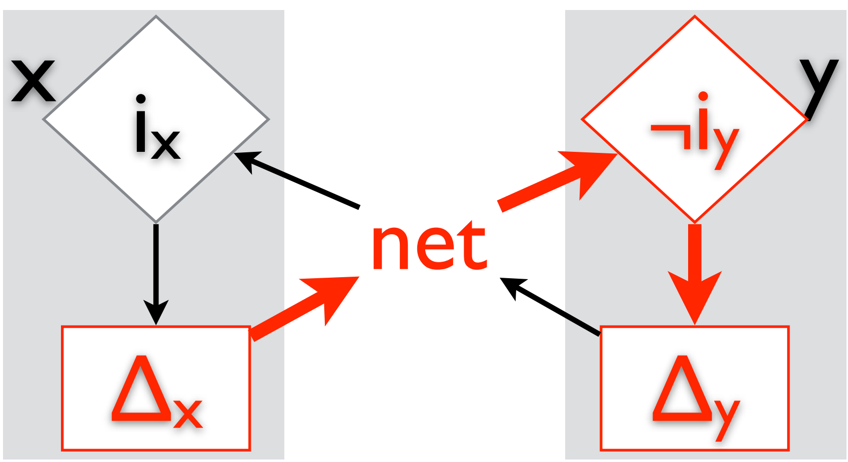
semantic dependency

policy x depends on y (denoted by x→y) if

semantic dependency

policy x depends on y (denoted by x➙y) if

x can violate y invariant and
trigger y action

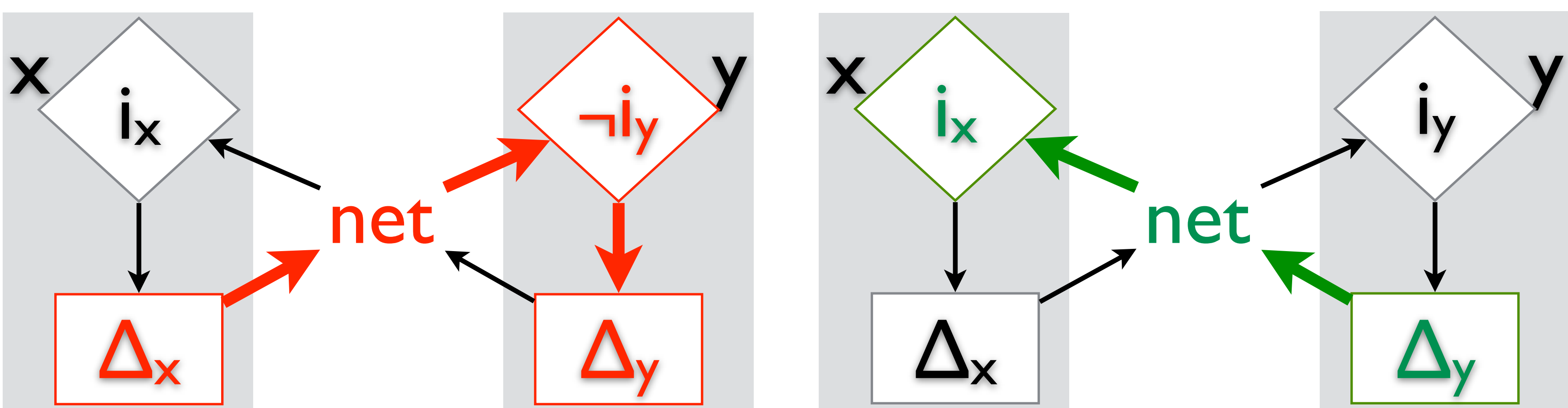


semantic dependency

policy x depends on y (denoted by x→y) if

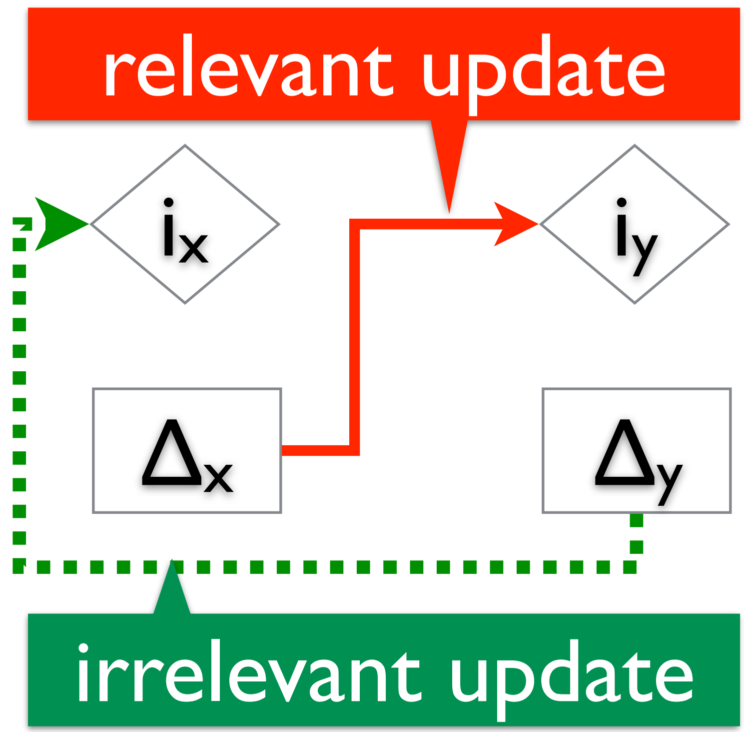
x can violate y invariant and trigger y action

but y will never affect x

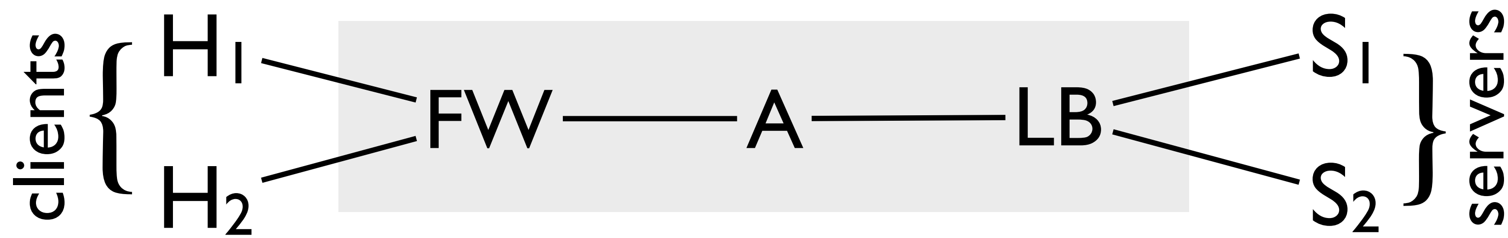


data (ir)relevance reasoning

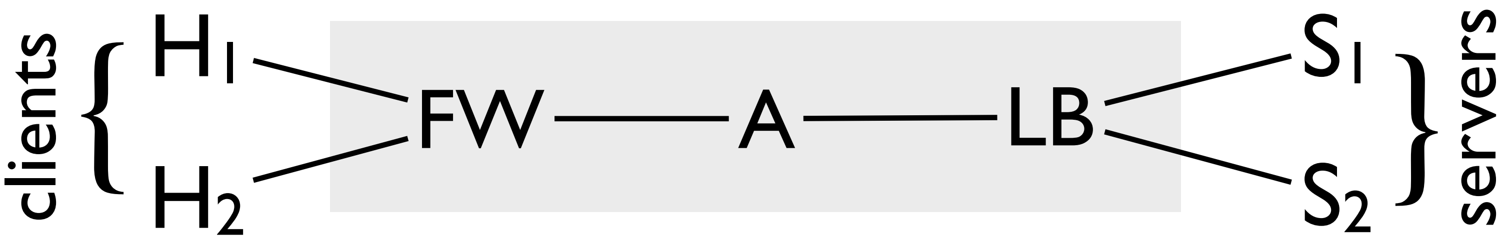
- update Δ_x is relevant to query i_y if $\Delta_x \wedge i_y$ is SAT
- Δ_y is irrelevant to i_x if $\Delta_y \wedge i_x$ is UNSAT



running example: SDN policies

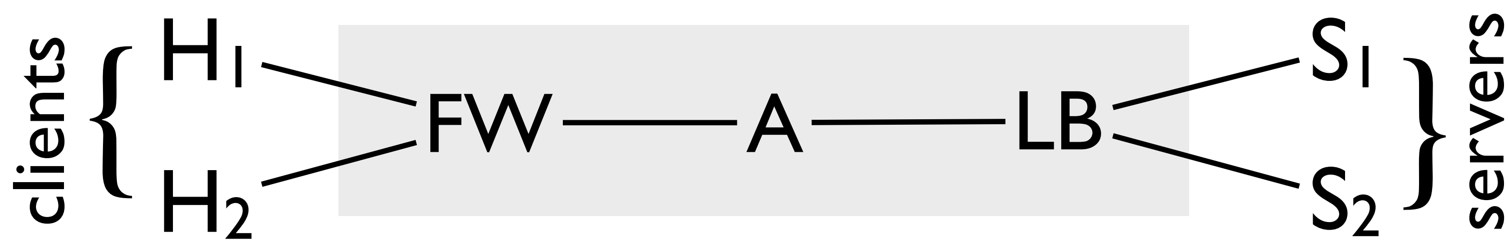


running example: SDN policies



`fw`, *firewall* blocks traffic from/to H_2

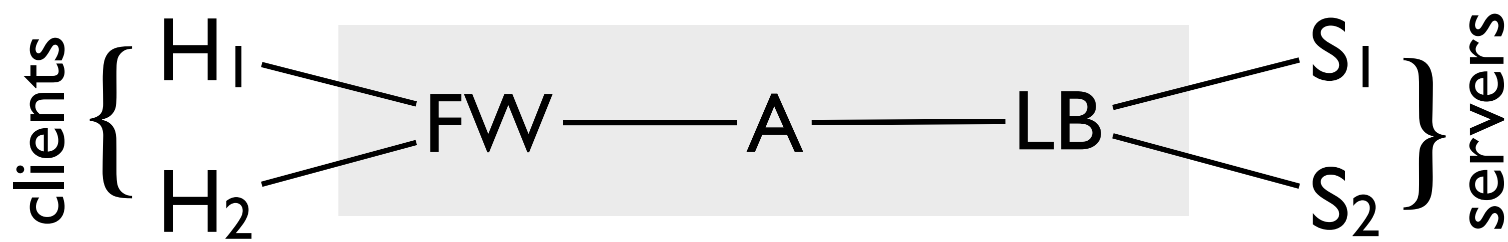
running example: SDN policies



`fw`, *firewall* blocks traffic from/to H_2

`lb`, *load balancer* directs H_1 traffic from/to S

running example: SDN policies

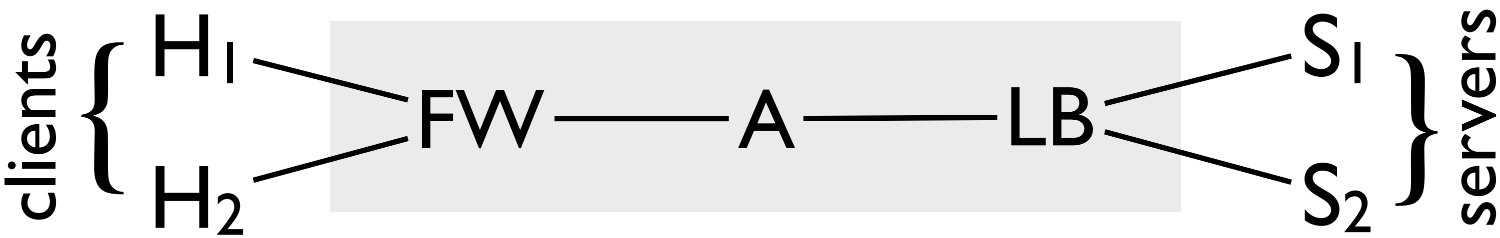


`fw`, *firewall* blocks traffic from/to H_2

`lb`, *load balancer* directs H_1 traffic from/to S

```
lb≜if_(client traffic?, lb₁, lb₂) where
    lb₁ ≜ pick a server from S₁,S₂
    lb₂ ≜ restore public server address
```

running example: SDN policies



`fw`, *firewall* blocks traffic from/to H_2

`lb`, *load balancer* directs H_1 traffic from/to S

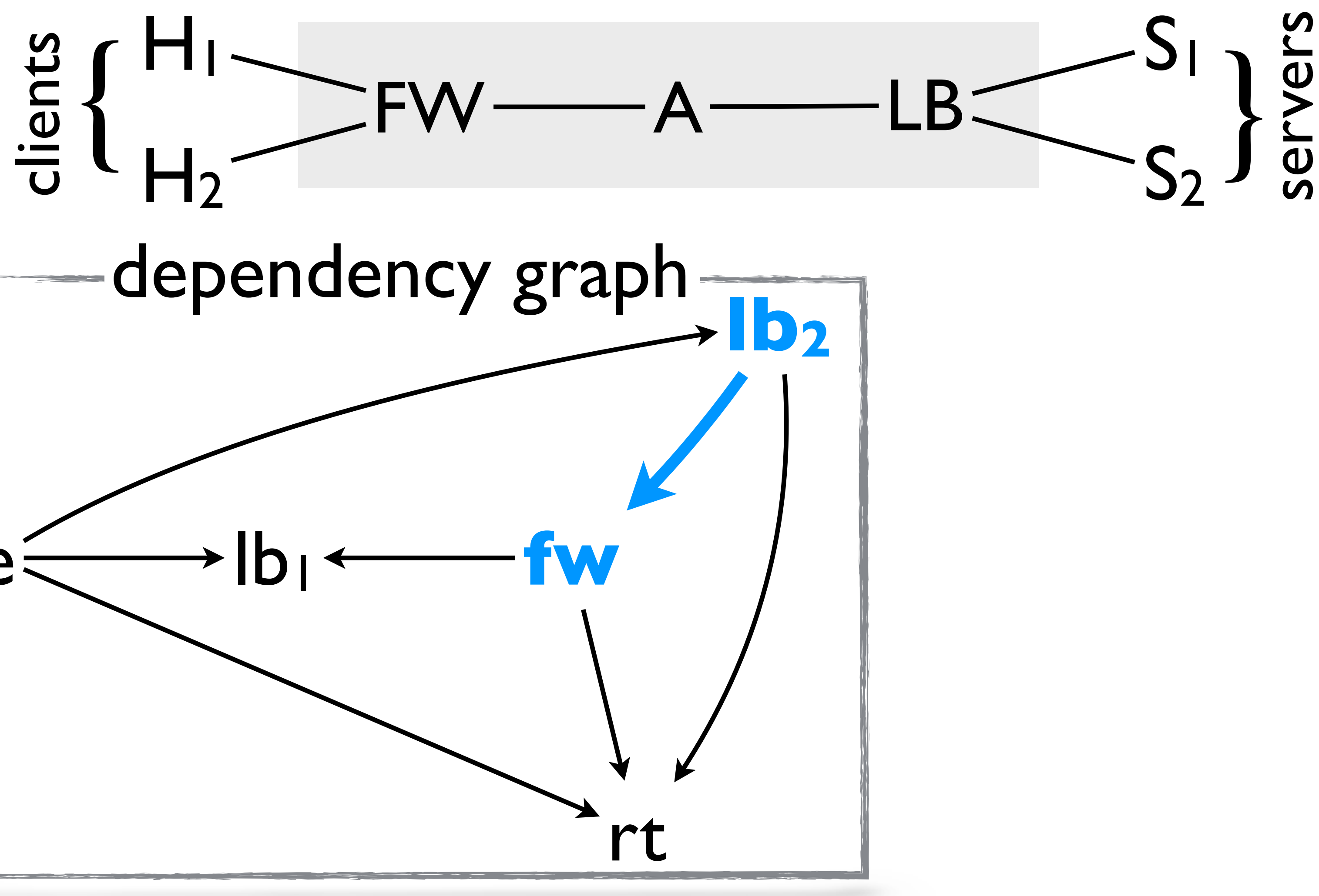
`lb≜if_(client traffic?, lb₁, lb₂) where`

`lb₁ ≜ pick a server from S₁,S₂`

`lb₂ ≜ restore public server address`

`rt`, *routing* between $H_{1,2}$ and S

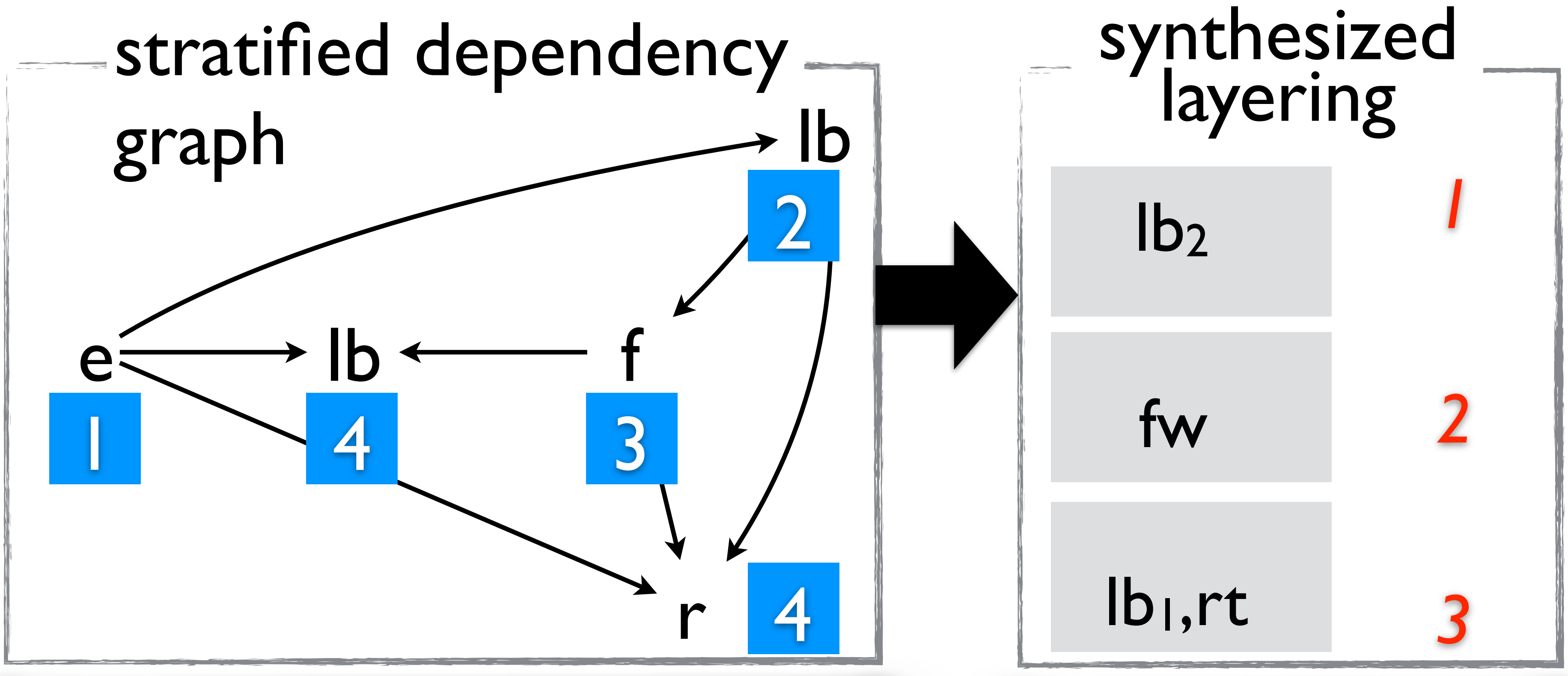
dependency graph



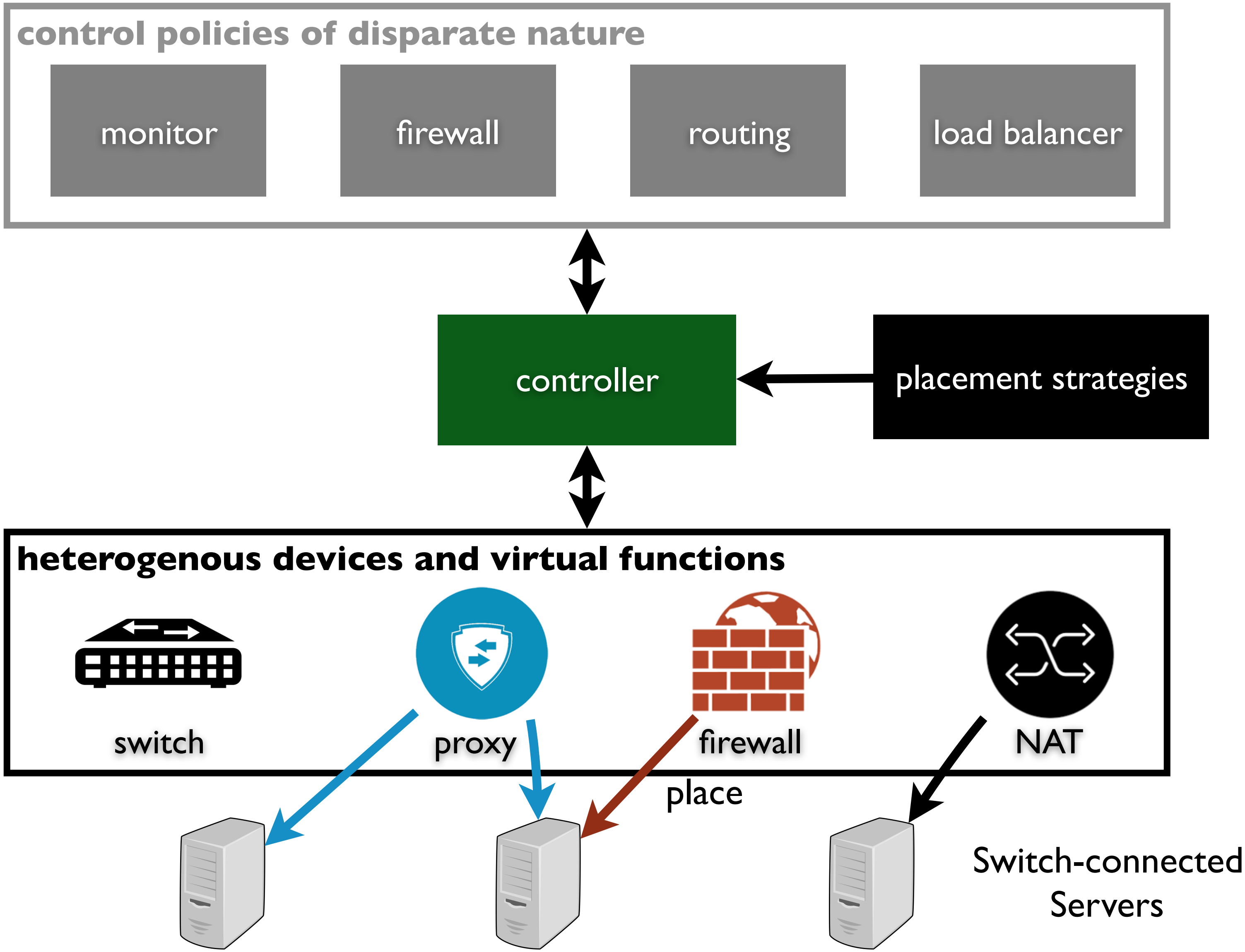
semantic layering

construct layering with stratification number

- correctness guarantee: the semantics of every policy will be preserved



resource orchestration



Middlebox

- Network Function Virtualization (NFV)
  - Technology of virtualizing network functions into software building blocks
- Middlebox: software implementation of network services
  - Improve the network performance:
    - Web proxy and video transcoder, load balancer, …
  - Enhance the security:
    - Firewall, IDS/IPS, passive network monitor, …
- Examples

Web Proxy

Firewall

NAT

Flows-to-Middlebox Requirement

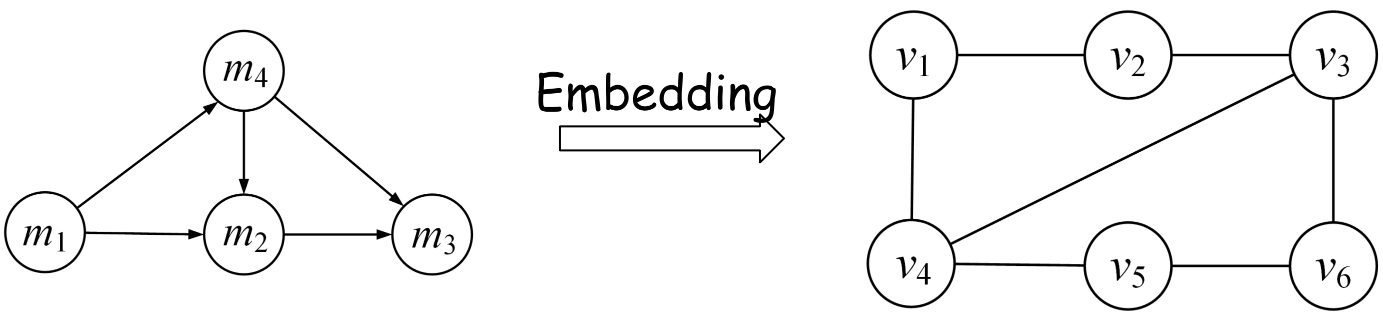
- Multiple middleboxes may/may not have a serving order
  - Examples
    - Firewall usually before Proxy
    - Virus scanner either before or after NAT gateway
- Categories
  - Non-ordered middlebox set (i.e., independent)
  - Totally-ordered middlebox set (service chain)

  - Partially-ordered middlebox set

[1] Dynamic Service Function Chaining in SDN-Enabled Networks with Middleboxes (ICNP '16)

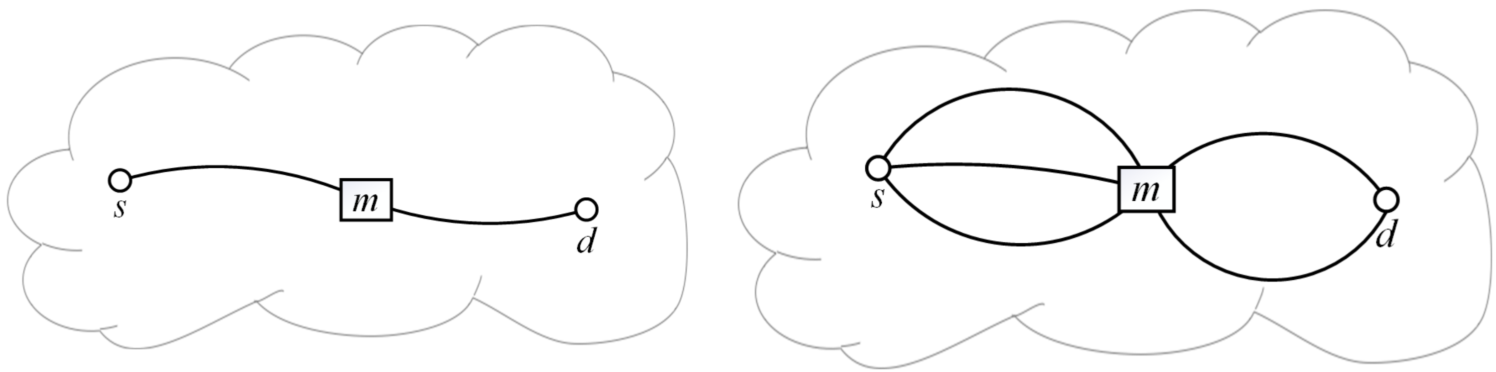
Middlebox Placement Problems

- **Graph embedding**
  - Middlebox graph, G_m, of multiple service chains that needs to be embedded in a give network graph, G_n.



Middlebox Placement Problems

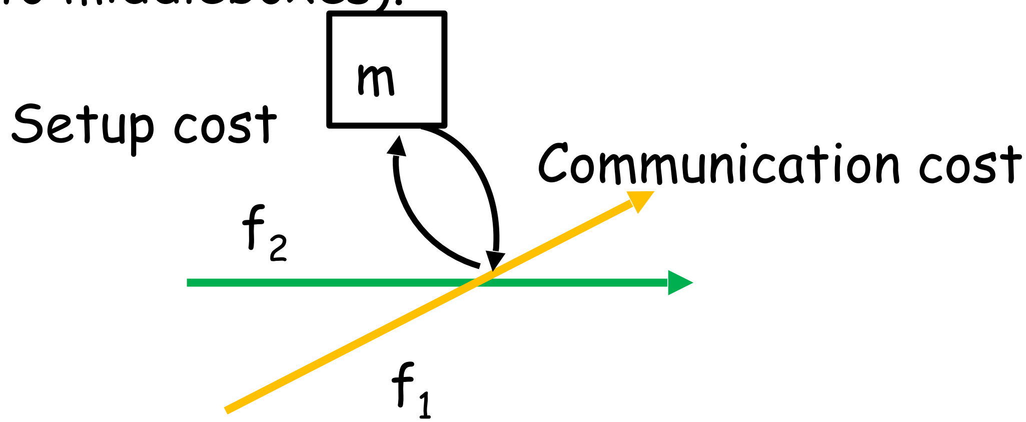
- **Graph flow routing**
  - Shortest path or maximum flow between a given source and destination that have to go through a given middlebox in G_n.



[2] Provably Efficient Algorithms for Joint Placement and Allocation of Virtual Network Functions(INFOCOM '17)

Middlebox Placement Problem

- **Facility allocation**
  - Optimal placement of facilities (i.e., middlebox) to minimize transportation costs (i.e., traffic, including detour traffic from flows to middleboxes).
- Cost

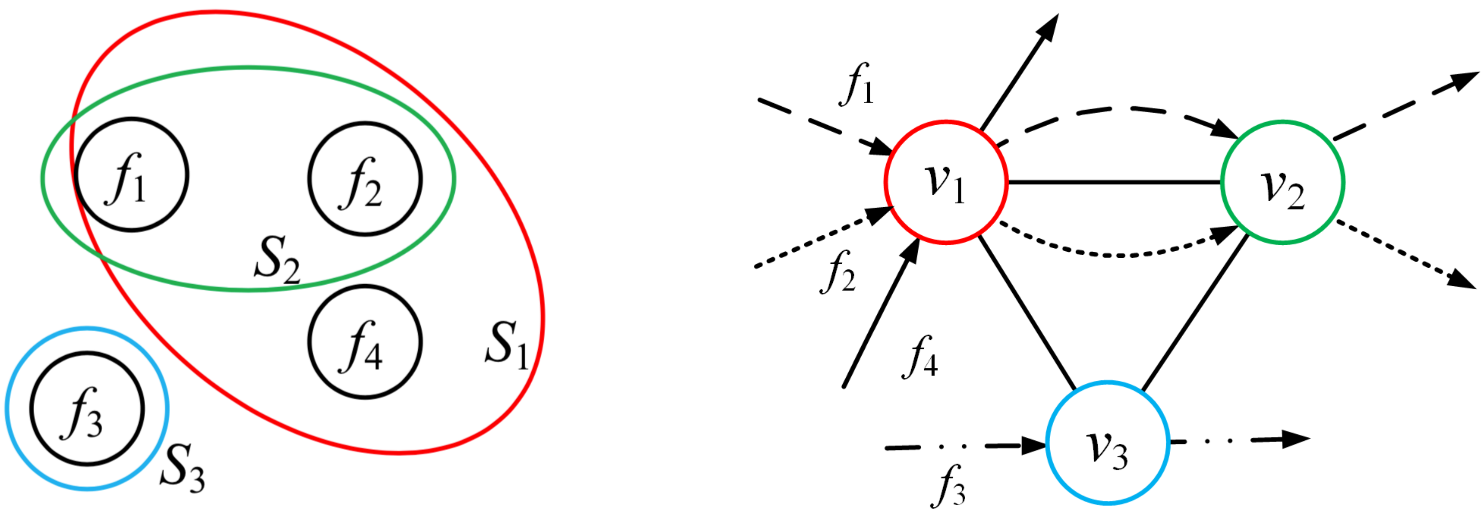


- Objective
  - Minimizing sum of middlebox setup cost and communication cost

[3] Near Optimal Placement of Virtual Network Functions (INFOCOM '15)

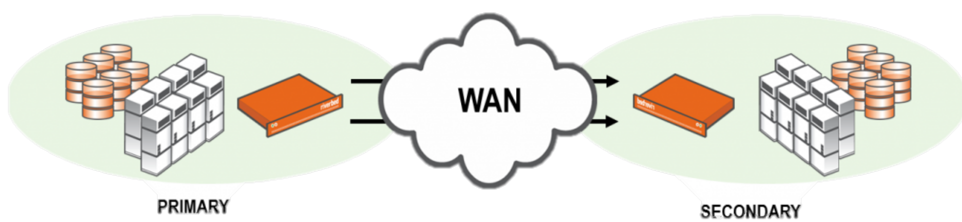
Middlebox Placement Problems

- **Set covering**
  - Minimize the number of middleboxes used to cover all flows.



Middlebox Traffic Changing Effects [4]

- Middleboxes may change flow rates in different ways
  - Citrix CloudBridge WAN accelerator: 20% (diminishing)



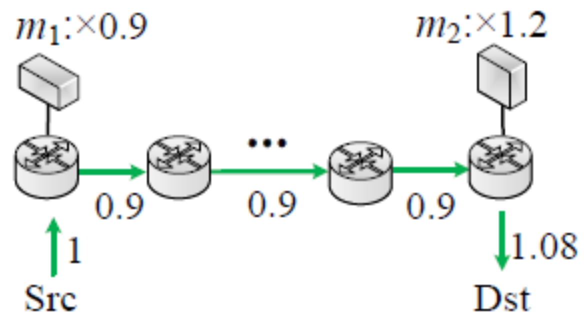
  - BCH(63,48) encoder: 130% (expanding)

- Objetive: minimizing total traffic

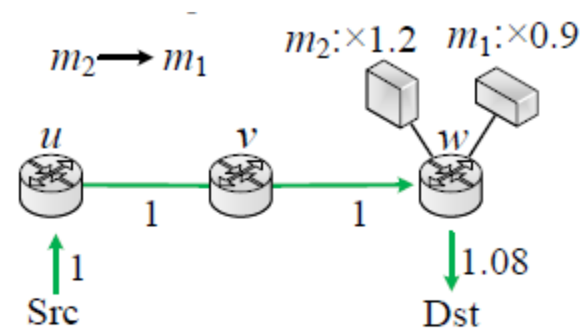
[4] Traffic Aware Placement of Interdependent NFV Middleboxes (INFOCOM '17)

Middlebox Placement Examples

- Independent middleboxes

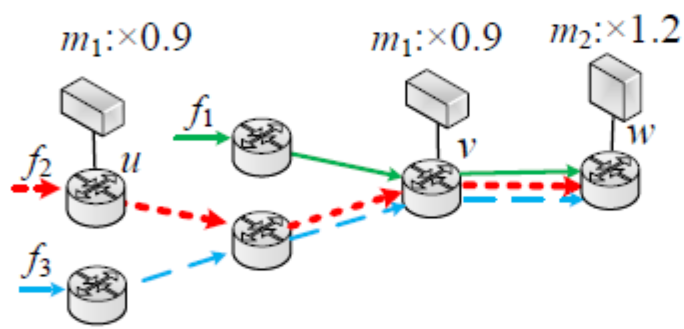


- Dependent middleboxes (m_2 before m_1)



Flow Placement Examples (cont'd)

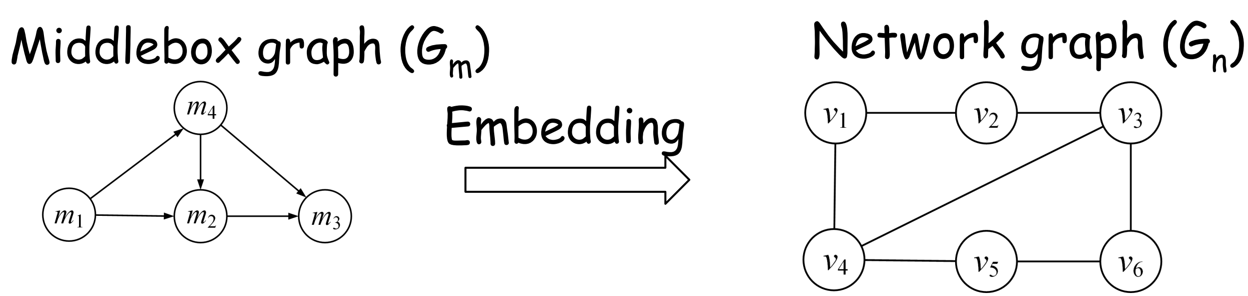
- A flow covered by multiple middleboxes



[5] NFV Middlebox Placement with Balanced Set-up Cost and Bandwidth Consumption(ICPP '18)

Challenges: NP-completeness

Node capacity	✓	*	✓	*	*
Edge capacity	✓	✓	*	*	*
Node placement constraint	*	✓	*	✓	✓
Edge routing constraint	*	*	✓	✓	*
Latency constraint	*	*	*	*	✓
NP-completeness and inapproximability under any objective[6]					



[6] Charting the Complexity Landscape of Virtual Network Embeddings (IFIP '18)

Other Challenges

- Special network graphs
  - Such as trees to make embedding tractable
- Other flow-to-middlebox policy
  - Forbidden to pass through certain middleboxes
- Other scheduling problems
  - Such as classic flow shop