GUUG Frühjahrsfachgespräch 2015

FIREWALLMANAGEMENT OHNE KOPFSCHMERZEN

Anders Henke

Topics

- (My) Situation
- Rock-Solid Technology
- Approach
- Data Model

(My) Situation

- Internet Service Provider (Hosting, Access, …)
  - Hundreds of public services
  - Thousands of non-public services
  - Service-Oriented Architecture (SOA)
  - No constraints on technology
  - ~ 1200 developers, ~ 200 sysadmins, ~20 network engineers

- There's more than one way to do it!
  - Traditional Software development, Systems Operations, Applications Operations
  - DevOps(ish)
  - OpsDev(ish)
  - … and more.

- Multi-layered defense in Depth
  - Using proven, rock-solid technology

Rock-Solid Technology

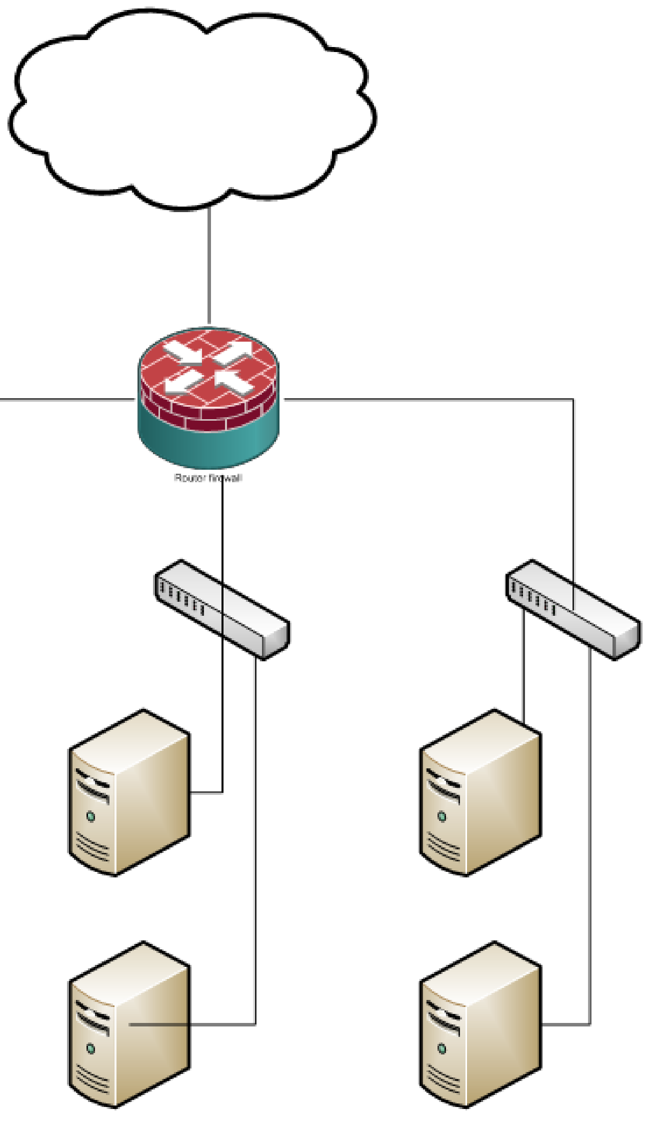
“Roadblock in Palestine” by Harry Pockets, CC-BY-SA 3.0

Firewall Technology

- Stateless Packetfilters
  - Simple, robust technology
- Avoid Stateful Packetfilters
  - Easier to break than stateless packetfilters
  - Cool for egress traffic, not for ingress traffic
  - →little/no benefit to us and avoided, if possible
- No “Next-Generation Firewalls”
  - Adds features we have no use for
  - Adds caveats we don’t want to have
  - →no benefits to us

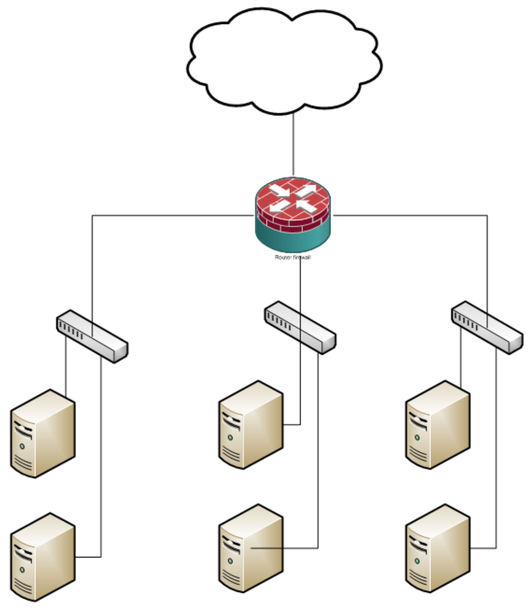
“Roadblock in Palestine” by Harry Pockets, CC-BY-SA 3.0

Traditional Firewall



Traditional Firewall

- Roadblock strictly separating any networks
  - Internet from internal networks
  - Internal networks from each other
  - Internal networks from Internet



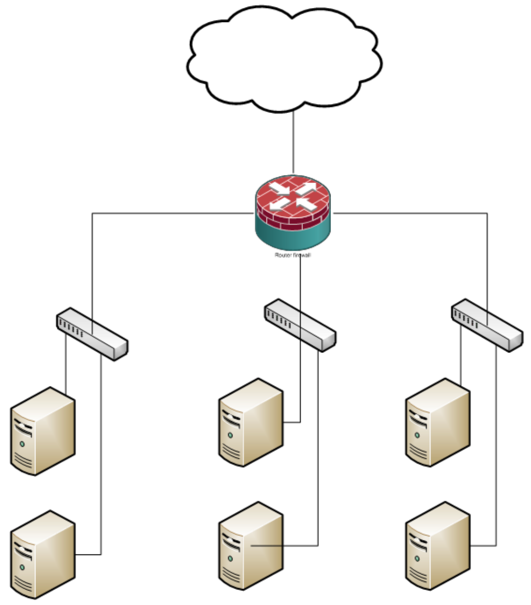
Traditional Firewall Management

- Lost in Translation
  - Dev gives connection requirements to Ops,
    Ops gives translates requirements to sysadmin,
    sysadmin translates to network-speak,
    firewall engineer translates to management software,
    firewall software translates to Cisco/Juniper/…
- Consequences
  - Many Firewall tickets require “rework”
  - Getting firewall rules done right may take weeks.

Scalability

- High demand of fine-granular internal firewall rules
  - More services, more hosts, more firewall rules
  - “We’re really secure, we do have tons of firewall rules”
  - Adding a single node to a cluster: up to 100 extra rules
  - Adding “infrastructure” nodes: extra rules for every network

- Exponential growth of fine-granular rules
  - Exceeding growth of customers
  - Exceeding growth of applications
  - Exceeding Moore’s law

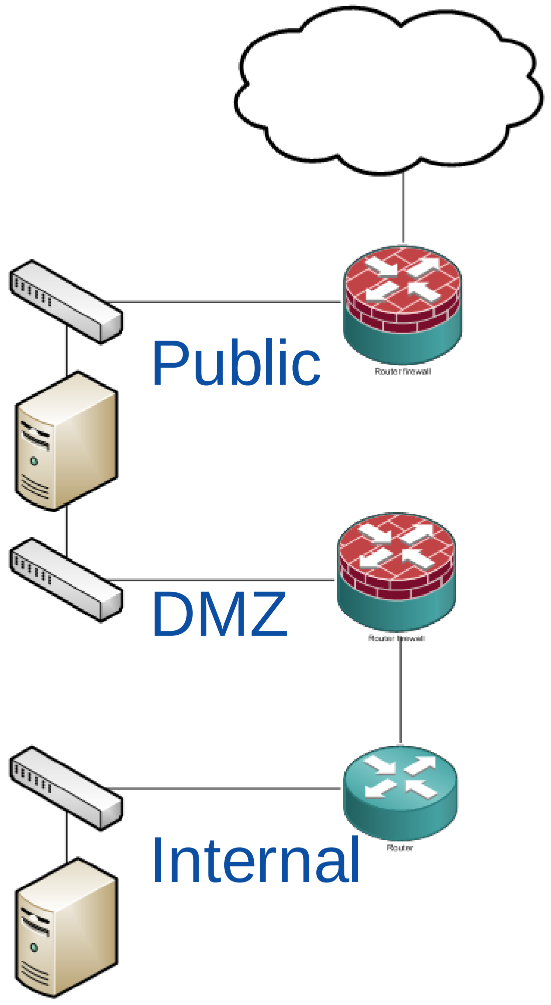


Restart

- Restart

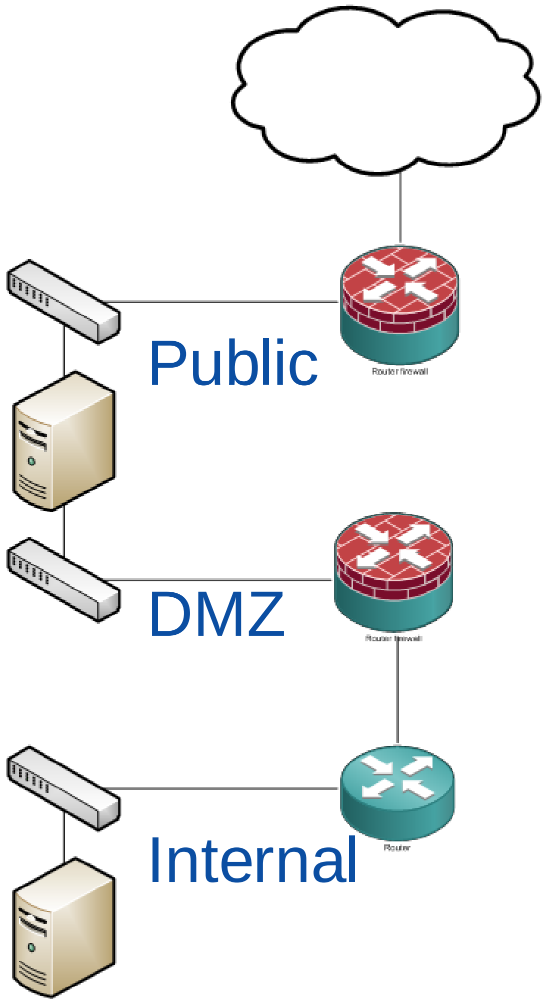
New Network Firewalling Concept

- Network Firewalling Zones
  - Fine-granular ACLs for public services
  - Subnet-granular ACLs from DMZ to internal networks
  - No network ACLs between internal networks
  - No network ACLs from internal networks to DMZ



New Firewalling Concept

- Network Firewalling Zones
  - Fine-granular ACLs for public services
  - Subnet-granular ACLs from DMZ to internal networks
  - No network ACLs between internal networks
  - No network ACLs from internal networks to DMZ
- Host Firewall
  - Fine-granular netfilter ACLs on every host
  - Also secures access within the same subnet
  - Every host only has its own set of ACLs: O(1)
- Application
  - Enforce Authentication / Authorization



How to manage netfilter?

- Custom scripting
  - Shell script, macro processor
  - Works well for very specific environments
  - You need to know what you’re doing.
- Off-the-shelf software
  - Typically: one admin to rule them all
  - Does not fit our environment
- Custom Management Software
  - Multi-user-aware
  - Tight integration with corporate Intranet, processes and tooling

Firewall Rule Management Software

- Approach
  - Changes do typically affect a fraction of hosts
  - Diff is centrally calculated, a change notification broadcasted to affected hosts
  - On notification or agent restart, agents do poll new netfilter configuration
  - Netfilter configuration is cached locally and applied on agent restart.

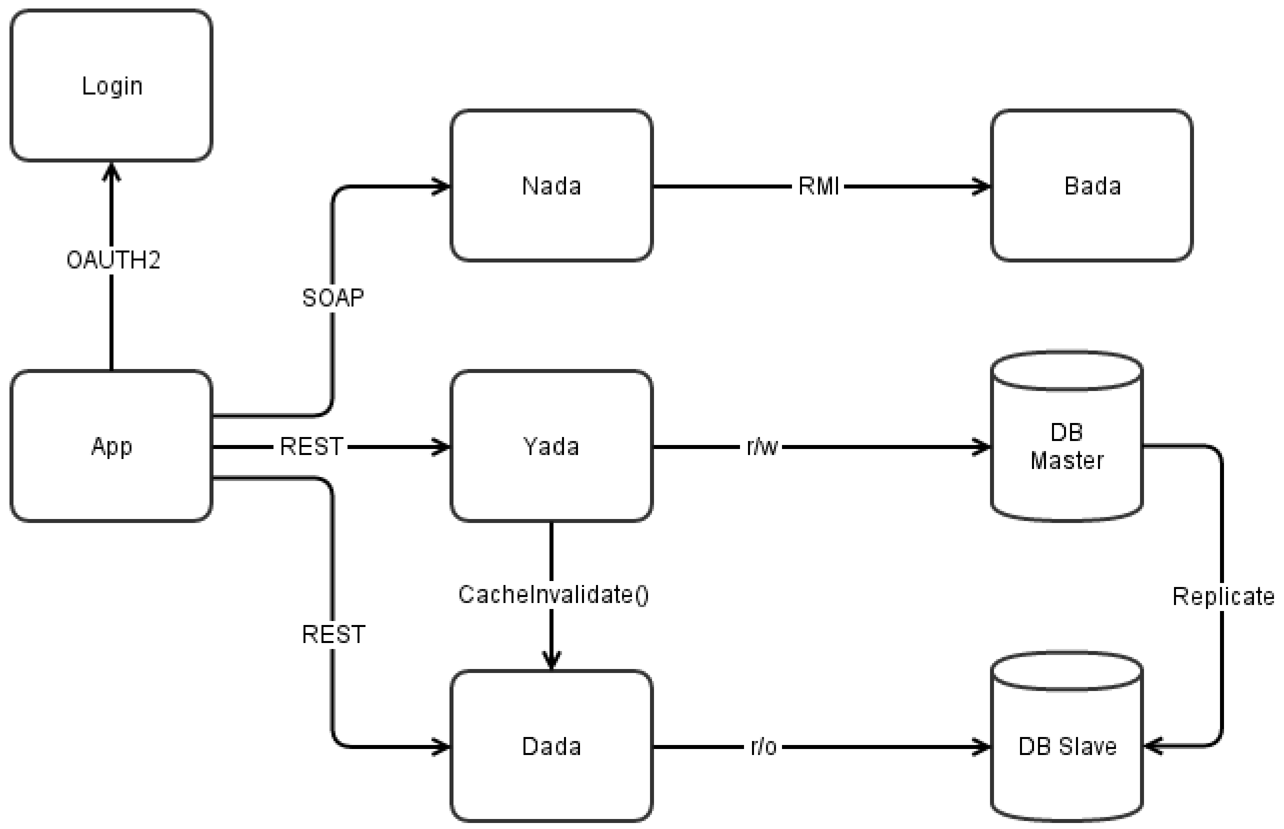
Firewall Rule Management Software

- No spreadsheet, no scripting language,
  no macro processor, no security groups

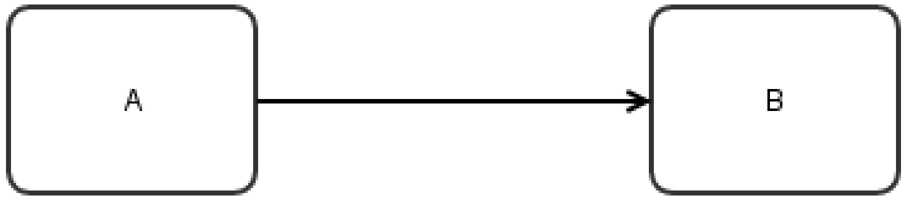
- Sysadmins do know about IP addresses, ports and protocols
  - … Application Owners, Developers and Management don’t necessarily do.

- Use a data model everyone understands!
  - Everyone should be able to read firewall rules!
  - Everyone should be able to write correct firewall rules!
  - Everyone should be able to approve/deploy their firewall rules!

Systems Architecture



Systems Architecture, Excerpt

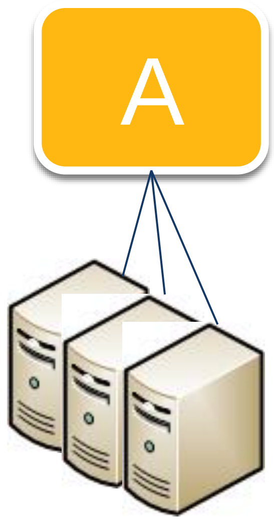


Application,
Consumer,
Client
…

Service,
Provider,
Server
…

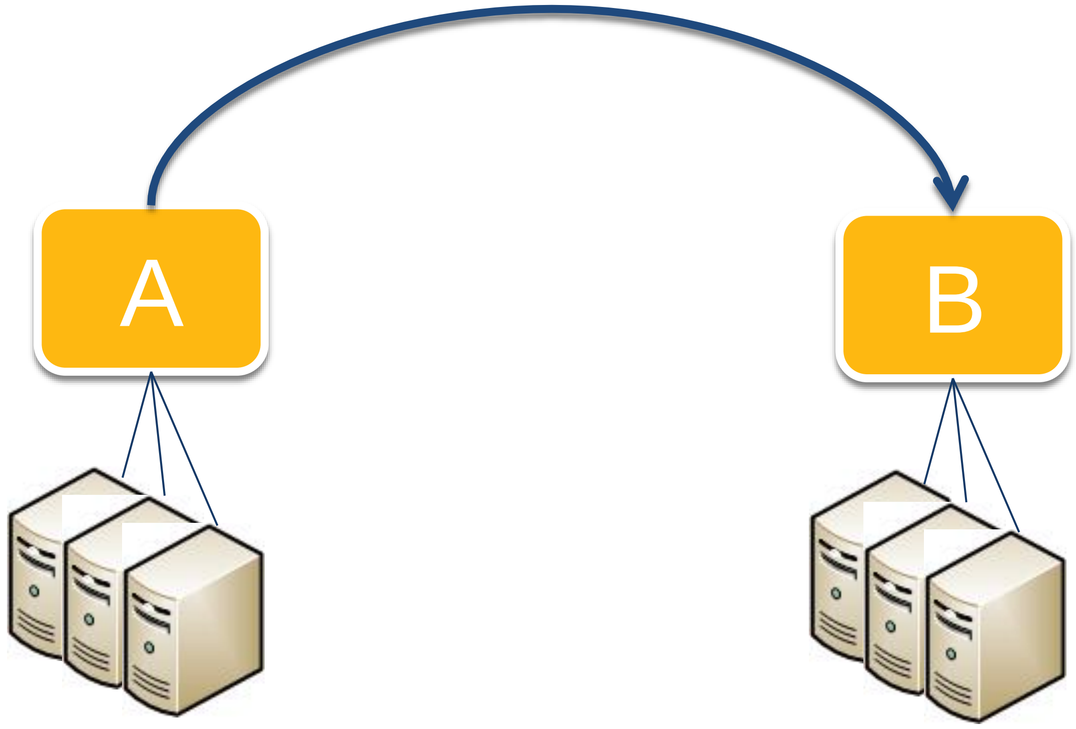
Firewall Management Data Model

- Remember your average Monitoring system?
  - A cluster of hosts is running service “A”
  - Every object does have a distinct name and responsible team
  - Apply the same principles to firewalling.



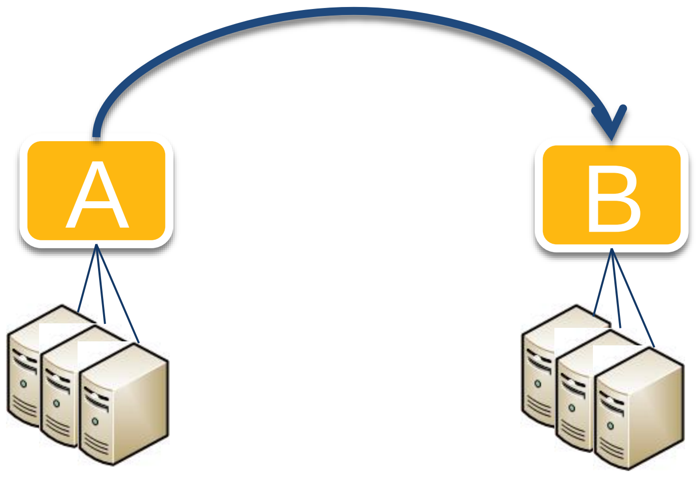
Firewall Management Data Model

- Service A does access Service B.



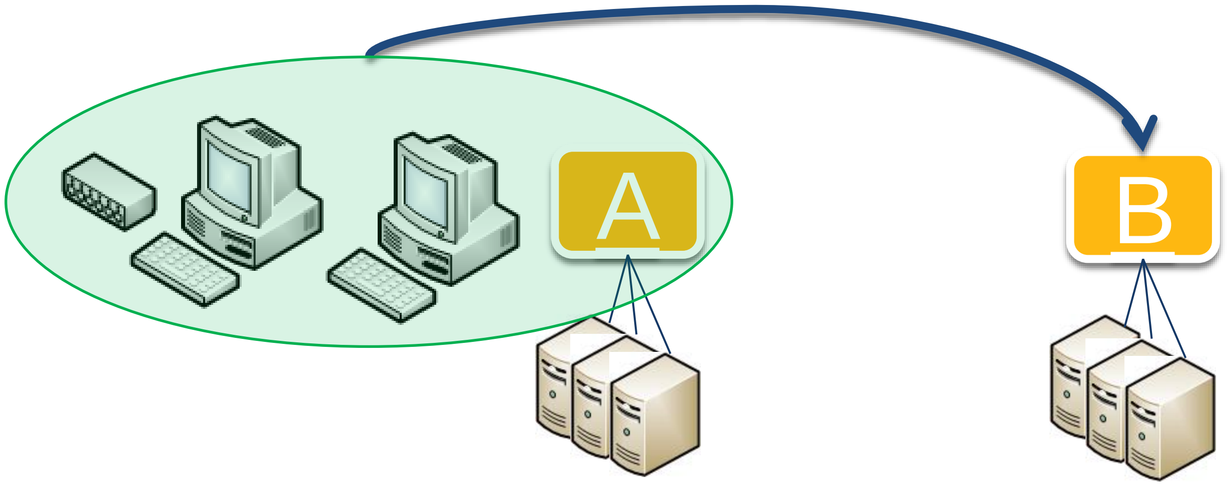
Treat everything as a graph of linked nodes

- Follow from service A (source) to its hosts
  - Destination = Definition of B
  - Consult source hosts for “`ip route get $destination`”
    - (Cache result as a fallback)
- Deploy rule on hosts running service B
  - Create/Replace custom chain for service B
  - Fill in pre-calculated, ordered rules
  - Link custom chain to INPUT chain
  - Done!
- Optionally: deploy on hosts running service A as an egress filter

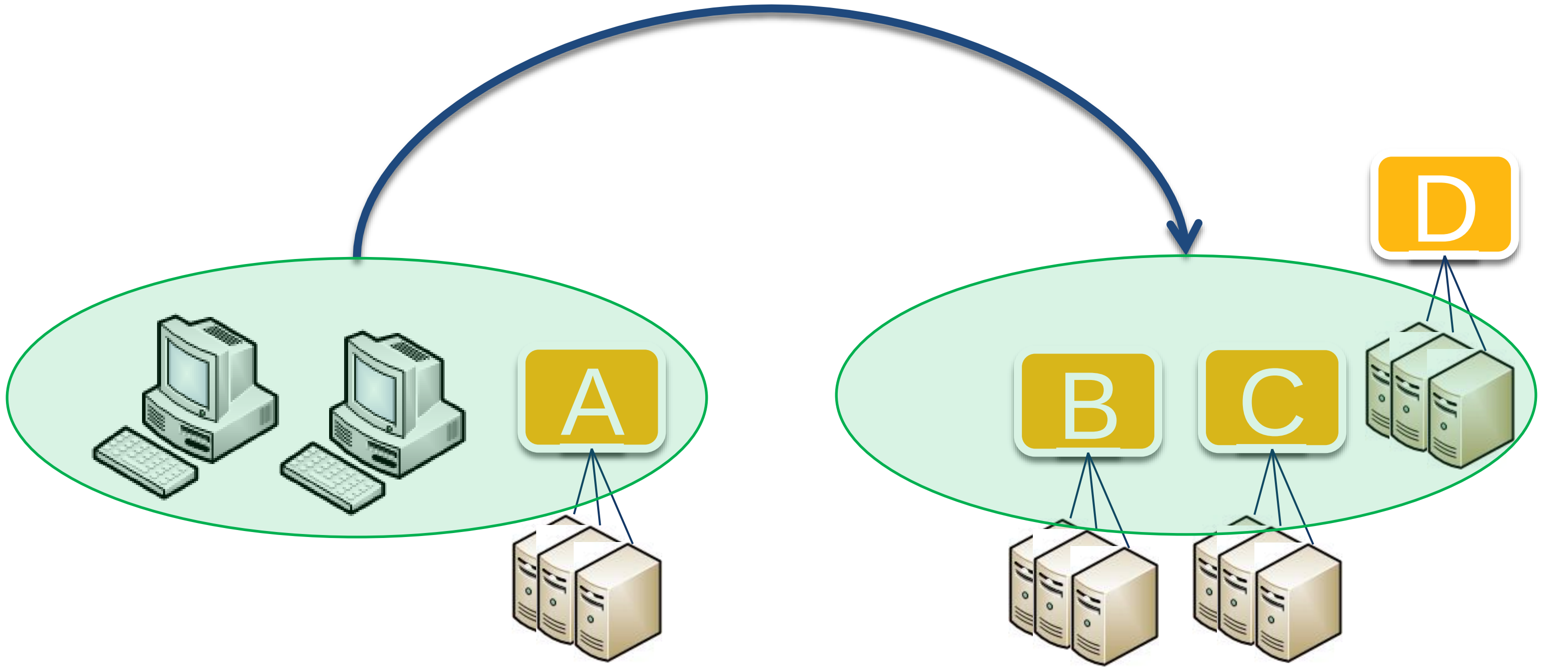


Legacy and Management

- Networks
  - CIDR-style networks do exist to integrate legacy sources
- Groups
  - One can create and manage groups
  - Groups can contain any other objects



Spot the error!

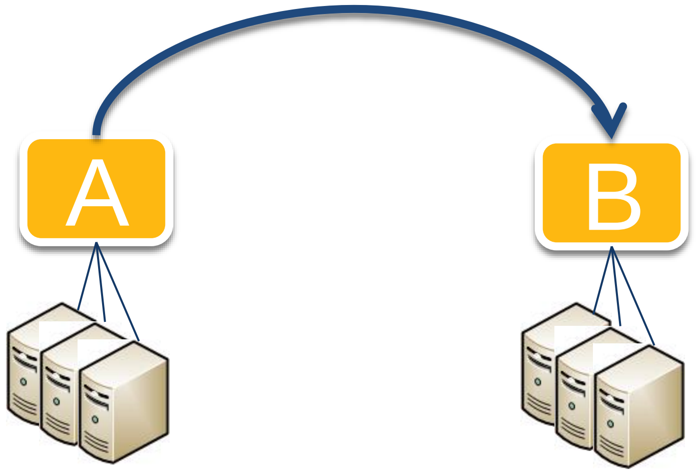


Default Policies

- Every service does have its own default policy
  - Restricted Service: default policy “reject”
  - Unrestricted Service: default policy “accept”

- Host default policy = catch-all for “unknown”
  - If you’d like to access all services, create a rule involving all services.
  - If you need “access all areas”, create a rule involving all services AND the host.

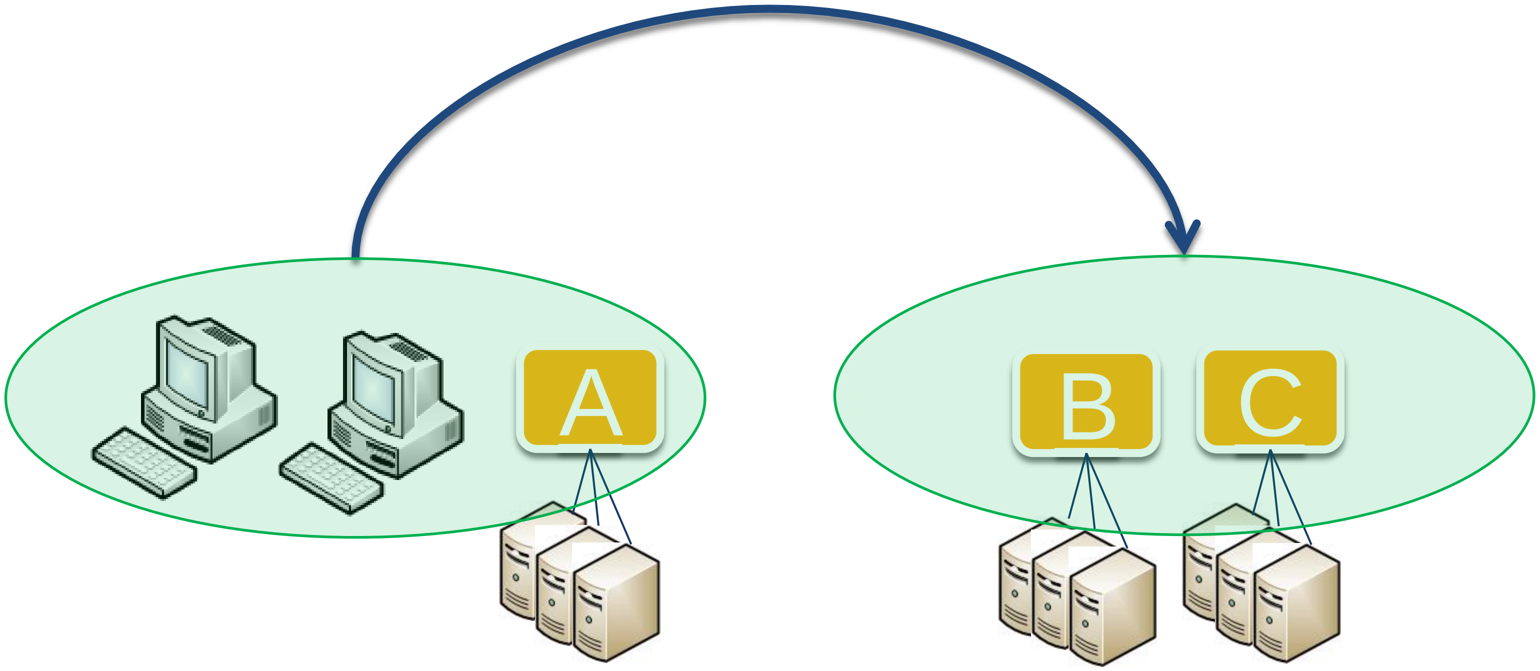
Responsibilities



- Responsibilities
  - Services running on a host are owned by the team running the host.
  - The team running the host may assign approval permissions for specific services to other teams.
- Approvals
  - The requesting team has to be responsible for source or destination (host/service).
  - If the requesting team is responsible for destination, we'll assume their approval; otherwise, we'll request their approval.
- Changes
  - Changing anything which results in re-deployment requires re-approval.
  - Until re-approval, the last approved stated is deployed.

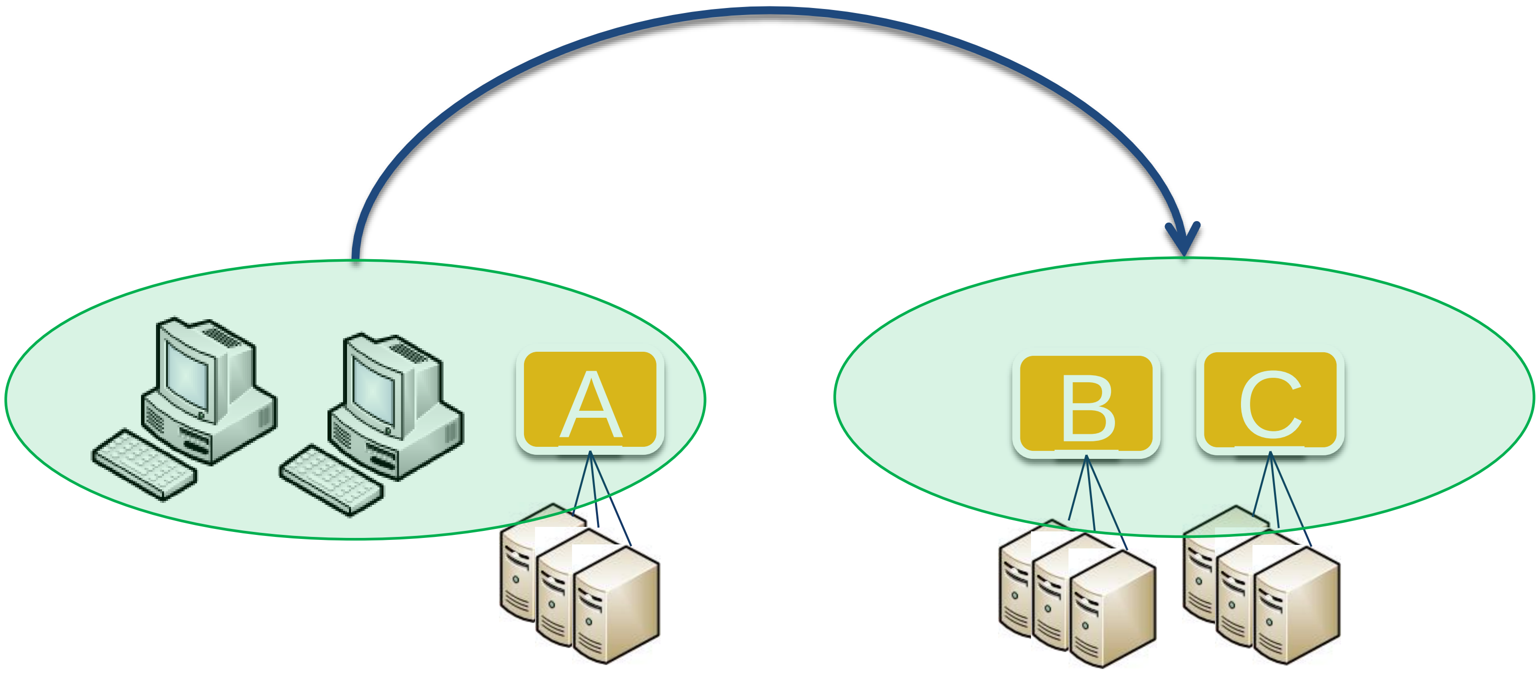
Stories solved

- Distributed authoring, approval and deployment of firewall rules
- Simple data model, simple rules, simple auditing
  - “Workstations of Team Y and Jenkins may access Tomcat Admin Consoles of Team Y”
- “Do what I mean” for IPv4/IPv6, incl. Default Address Selection



Stories solved (2)

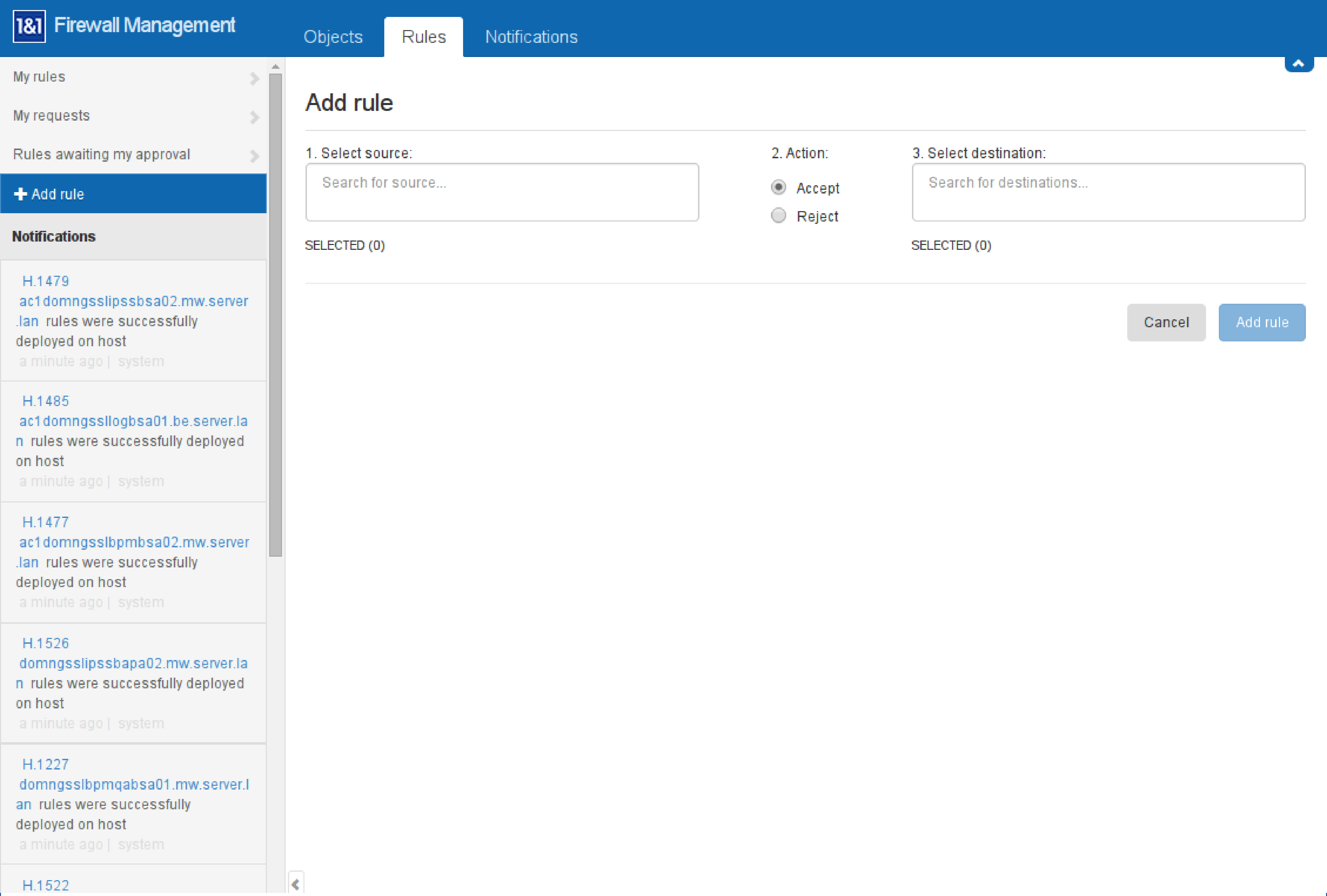
- Rules may become more static
  - Clients do have less (recognizable) rules: more auditable
    - At best, every client only has a single rule with all destination services
  - When endpoints change: change the rule, don't create additional rules



Images

- “Roadblock in Palestine” by Harry Pockets – Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons https://commons.wikimedia.org/wiki/File:Roadblock_in_Palestine.jpg

BACKUP



1&1 Firewall Management

Objects | Rules | Notifications

My rules

My requests

Rules awaiting my approval

+ Add rule

Notifications

H.1479 ac1domngsslipssbsa02.mw.server.lan rules were successfully deployed on host
a minute ago | system

H.1485 ac1domngssllogbsa01.be.server.lan rules were successfully deployed on host
a minute ago | system

H.1477 ac1domngsslbpmbsa02.mw.server.lan rules were successfully deployed on host
a minute ago | system

H.1526 domngsslipssbapa02.mw.server.lan rules were successfully deployed on host
a minute ago | system

H.1227 domngsslbpmqabsa01.mw.server.lan rules were successfully deployed on host
a minute ago | system

H.1522

Add rule

1. Select source: Search for source...

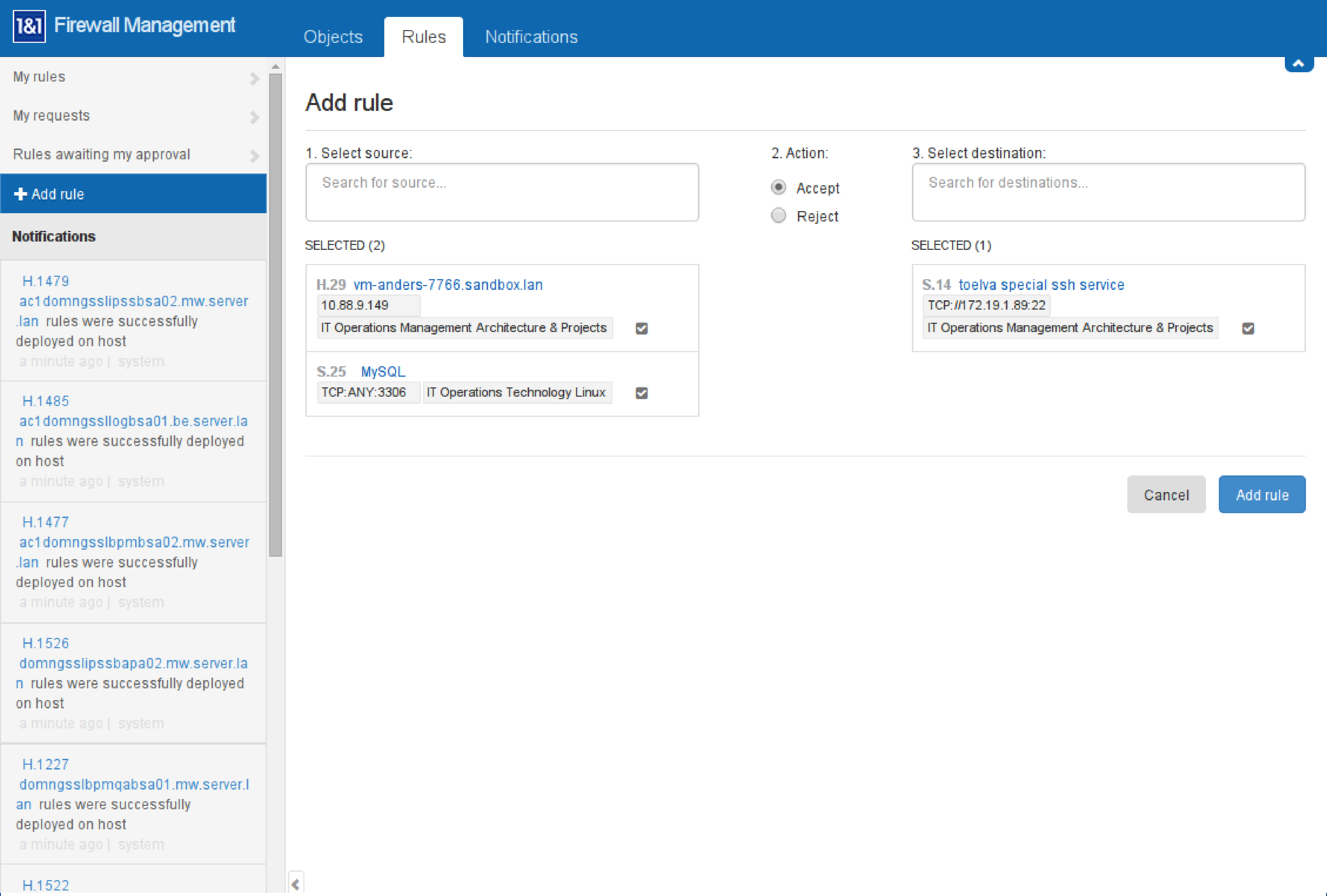
SELECTED (0)

2. Action: [✓ Accept] Reject

3. Select destination: Search for destinations...

SELECTED (0)

Cancel | Add rule



H.1479 ac1domngsslipssbsa02.mw.server.lan rules were successfully deployed on host
a minute ago | system

H.1485 ac1domngssllogbsa01.be.server.lan rules were successfully deployed on host
a minute ago | system

H.1477 ac1domngsslbpmbsa02.mw.server.lan rules were successfully deployed on host
a minute ago | system

H.1526 domngsslipssbapa02.mw.server.lan rules were successfully deployed on host
a minute ago | system

H.1227 domngsslbpmqabsa01.mw.server.lan rules were successfully deployed on host
a minute ago | system

H.1522

Add rule

1. Select source: Search for source...

SELECTED (2)

H.29 vm-anders-7766.sandbox.lan
10.88.9.149
IT Operations Management Architecture & Projects ✓

S.25 MySQL
TCP:ANY:3306 | IT Operations Technology Linux ✓

2. Action: [✓ Accept] Reject

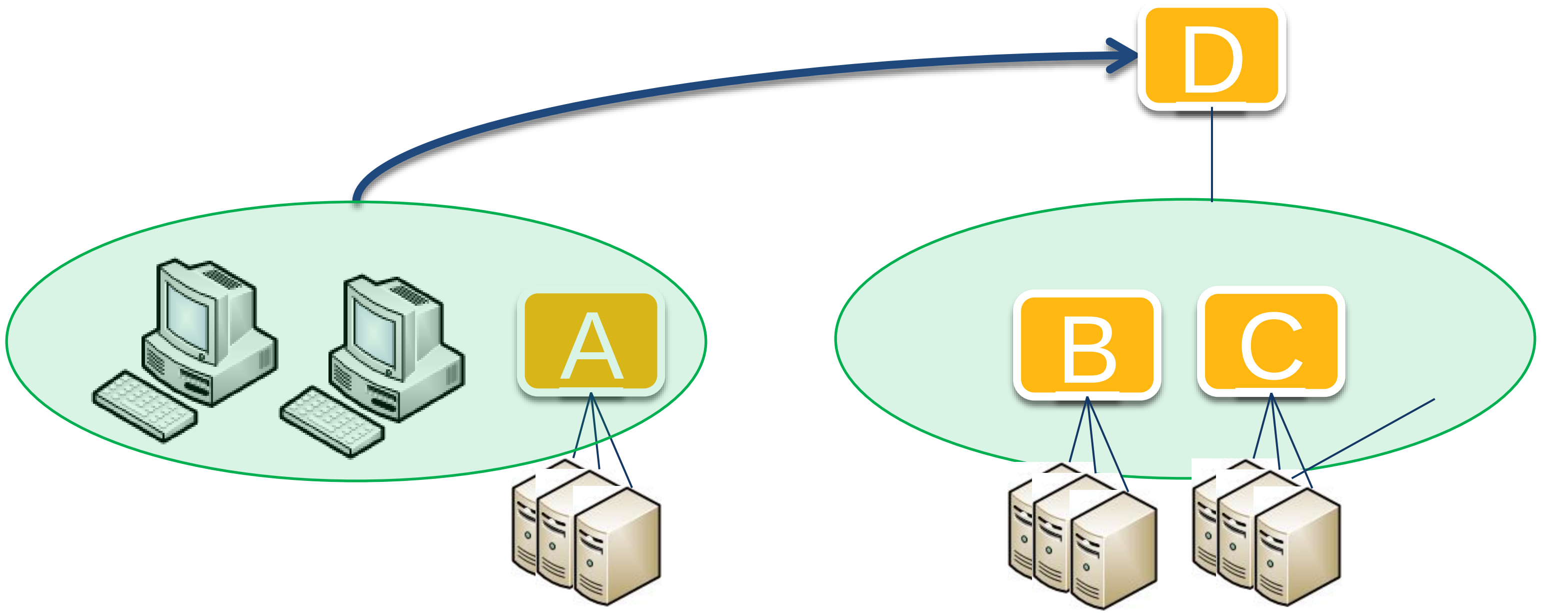
3. Select destination: Search for destinations...

SELECTED (1)

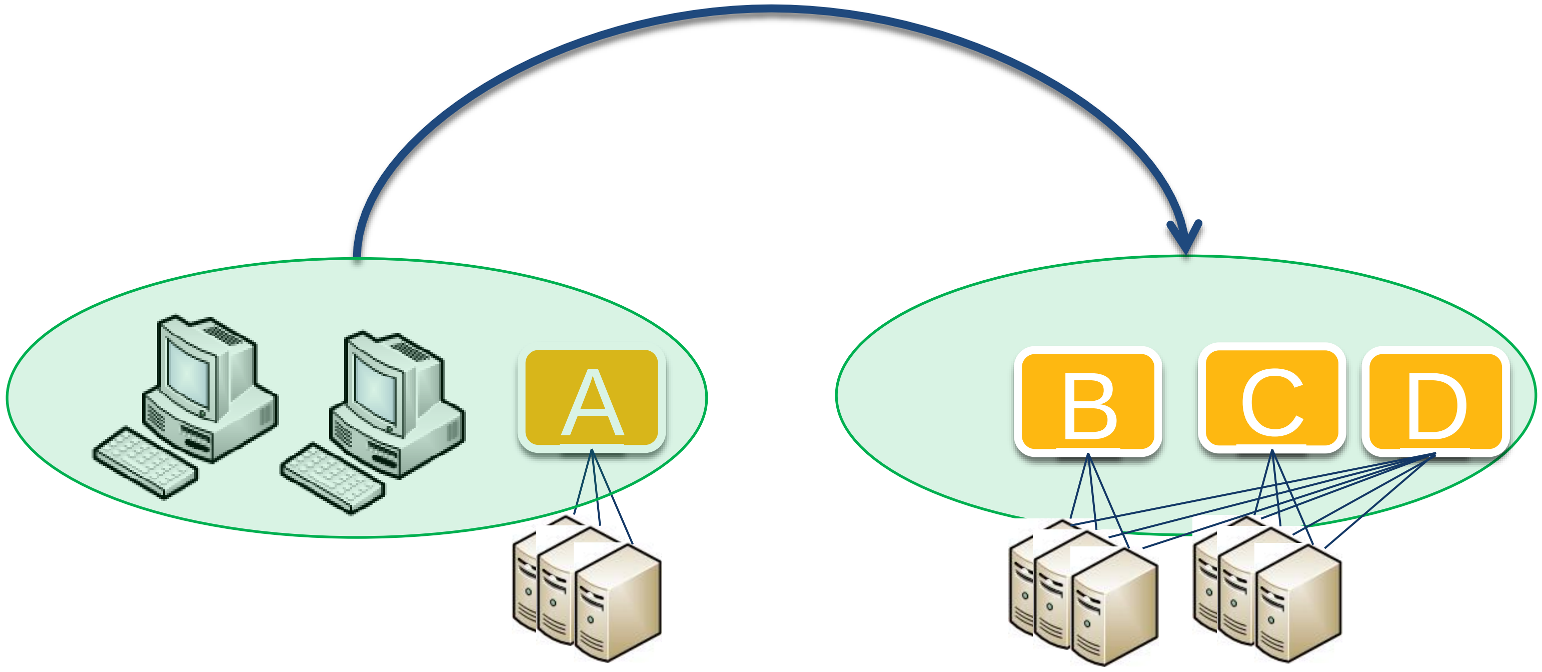
S.14 toelva special ssh service
TCP://172.19.1.89:22
IT Operations Management Architecture & Projects ✓

Cancel | Add rule

Services may be hosted on groups



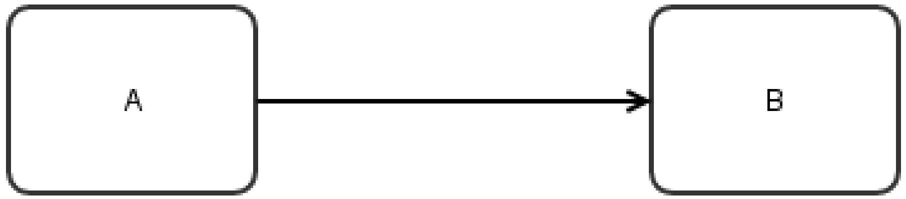
Services are not limited to a single cluster.



Not yet implemented…

- Calculate & Request Network-ACLs
- Graphical Audit

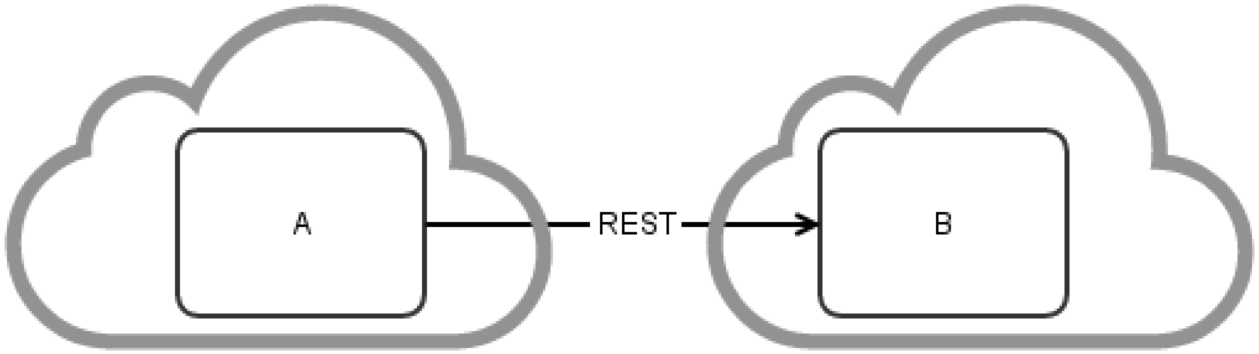
Transformation of System Architecture to Firewall Rules



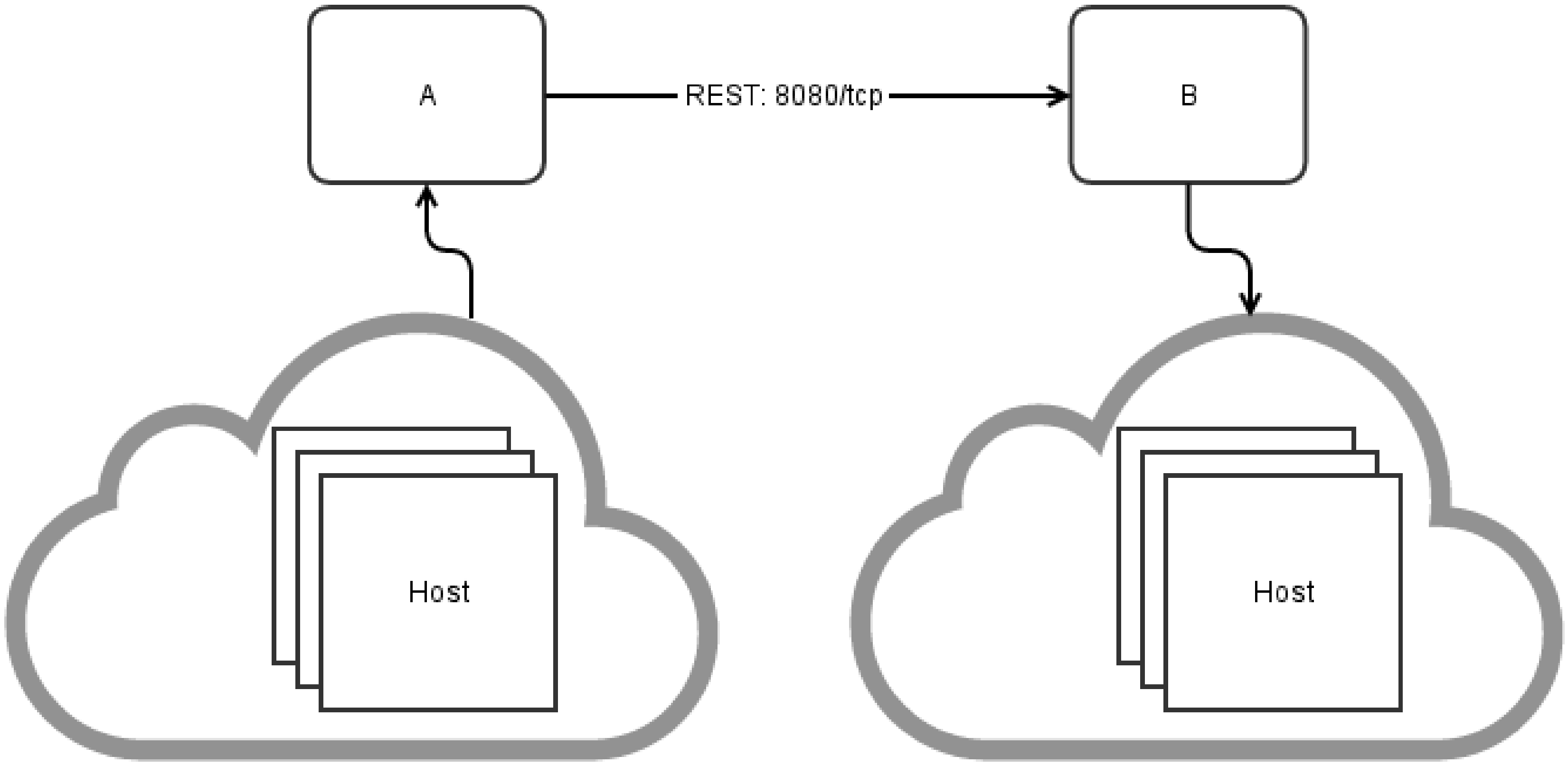
Application

Service

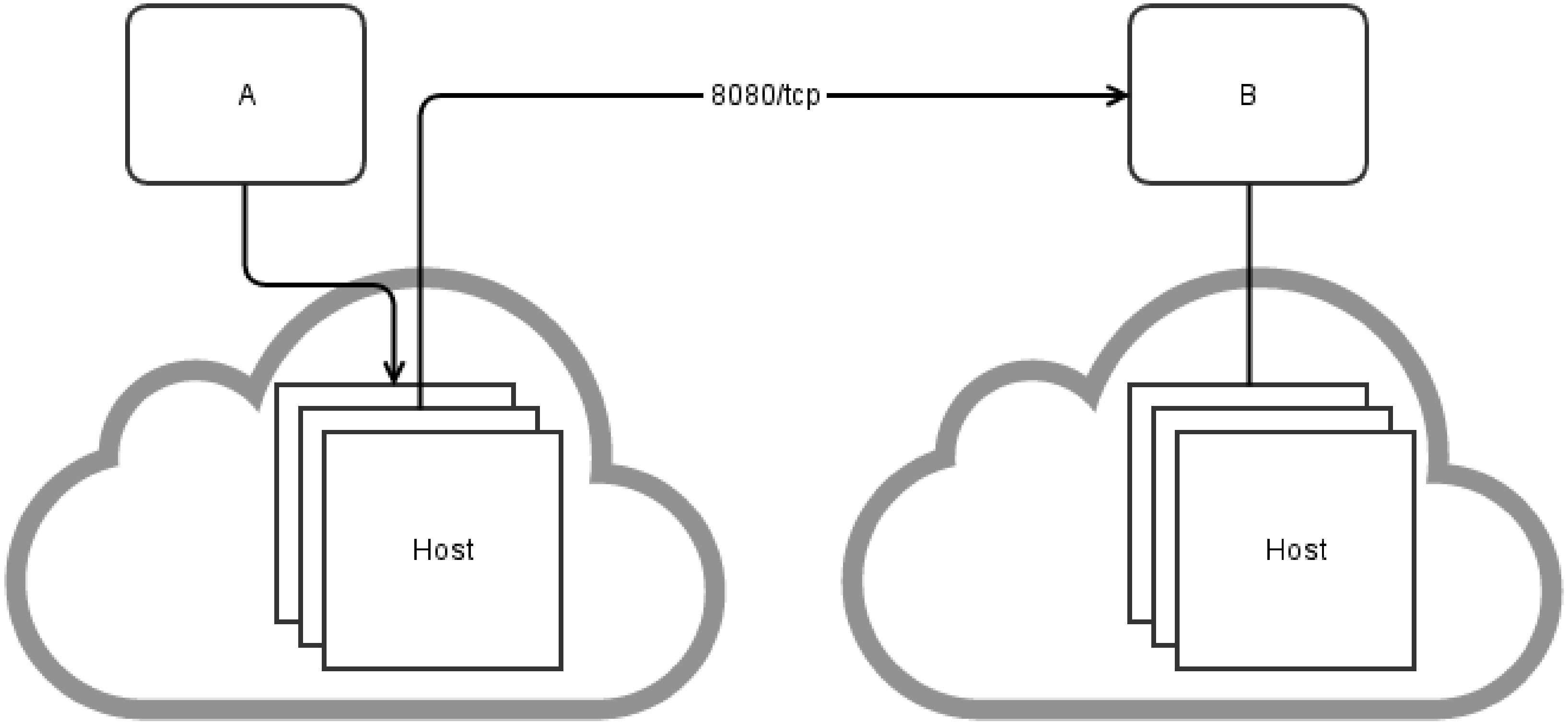
Transformation of System Architecture to Firewall Rules



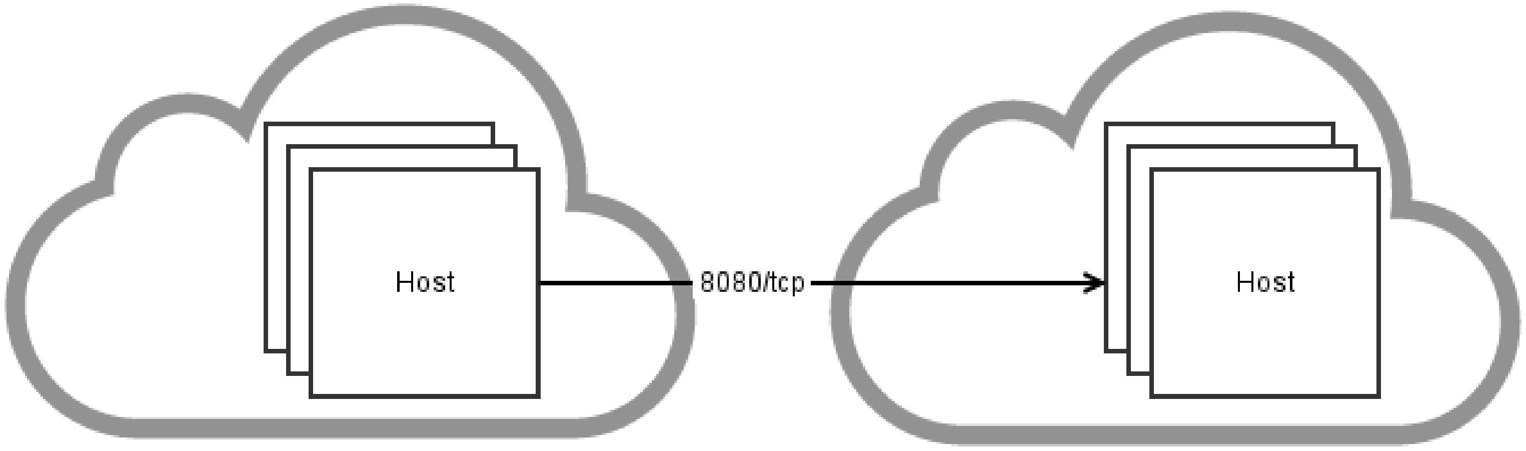
Transformation of System Architecture to Firewall Rules



Transformation of System Architecture to Firewall Rules



Transformation of System Architecture to Firewall Rules



Transformation of System Architecture to Firewall Rules

```
iptables -I INPUT --src 192.0.2.34 --dst 192.0.2.97 --dport 8443 -p tcp -j ACCEPT
iptables -I INPUT --src 192.0.2.34 --dst 192.0.2.56 --dport 8443 -p tcp -j ACCEPT
iptables -I INPUT --src 192.0.2.34 --dst 192.0.2.2 --dport 8443 -p tcp -j ACCEPT
iptables -I INPUT --src 192.0.2.38 --dst 192.0.2.97 --dport 8443 -p tcp -j ACCEPT
iptables -I INPUT --src 192.0.2.38 --dst 192.0.2.56 --dport 8443 -p tcp -j ACCEPT
iptables -I INPUT --src 192.0.2.38 --dst 192.0.2.2 --dport 8443 -p tcp -j ACCEPT
iptables -I INPUT --src 192.0.2.45 --dst 192.0.2.97 --dport 8443 -p tcp -j ACCEPT
iptables -I INPUT --src 192.0.2.45 --dst 192.0.2.56 --dport 8443 -p tcp -j ACCEPT
iptables -I INPUT --src 192.0.2.45 --dst 192.0.2.2 --dport 8443 -p tcp -j ACCEPT
```