Managing Security in Dynamic Networks

Alexander V. Konstantinou

Yechiam Yemini

Columbia University

Sandeep Bhatt

S. Rajagopalan

Telcordia Technologies

(formerly Bellcore)

Overview

1. Dynamic Network Example
2. Automating Network Configuration
3. NESTOR Architecture
4. Example Revisited
5. Future Work

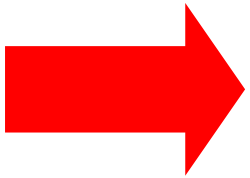
Dynamic Networks

- *Network*: elements, services, and policy
- *Dynamic Network* : components may change

Goal : manage configuration to maintain policy through change

Configuration Mgmt is Difficult

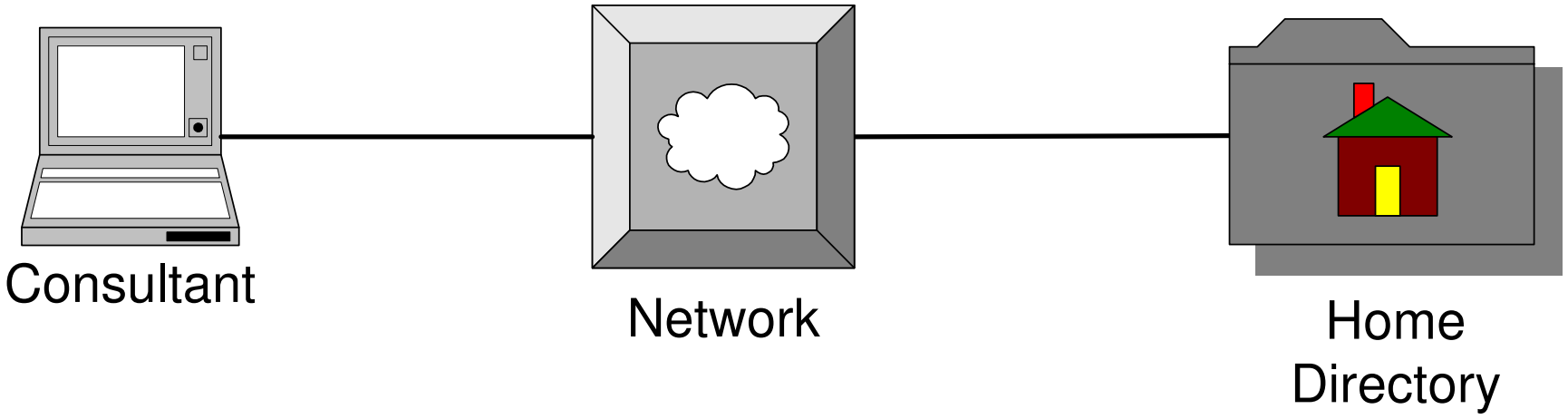
- Human-intensive
- Distributed heterogeneous data
  - Single task involves multiple elements
  - Duplication & dependencies
- No verification of integrity rules
- Manual recovery



Static configuration
& network failure

Dynamic Network Example

- Consultant visiting client needs to access home directory
- *Goal*: Plug laptop & double-click on home folder



Example Security Policies

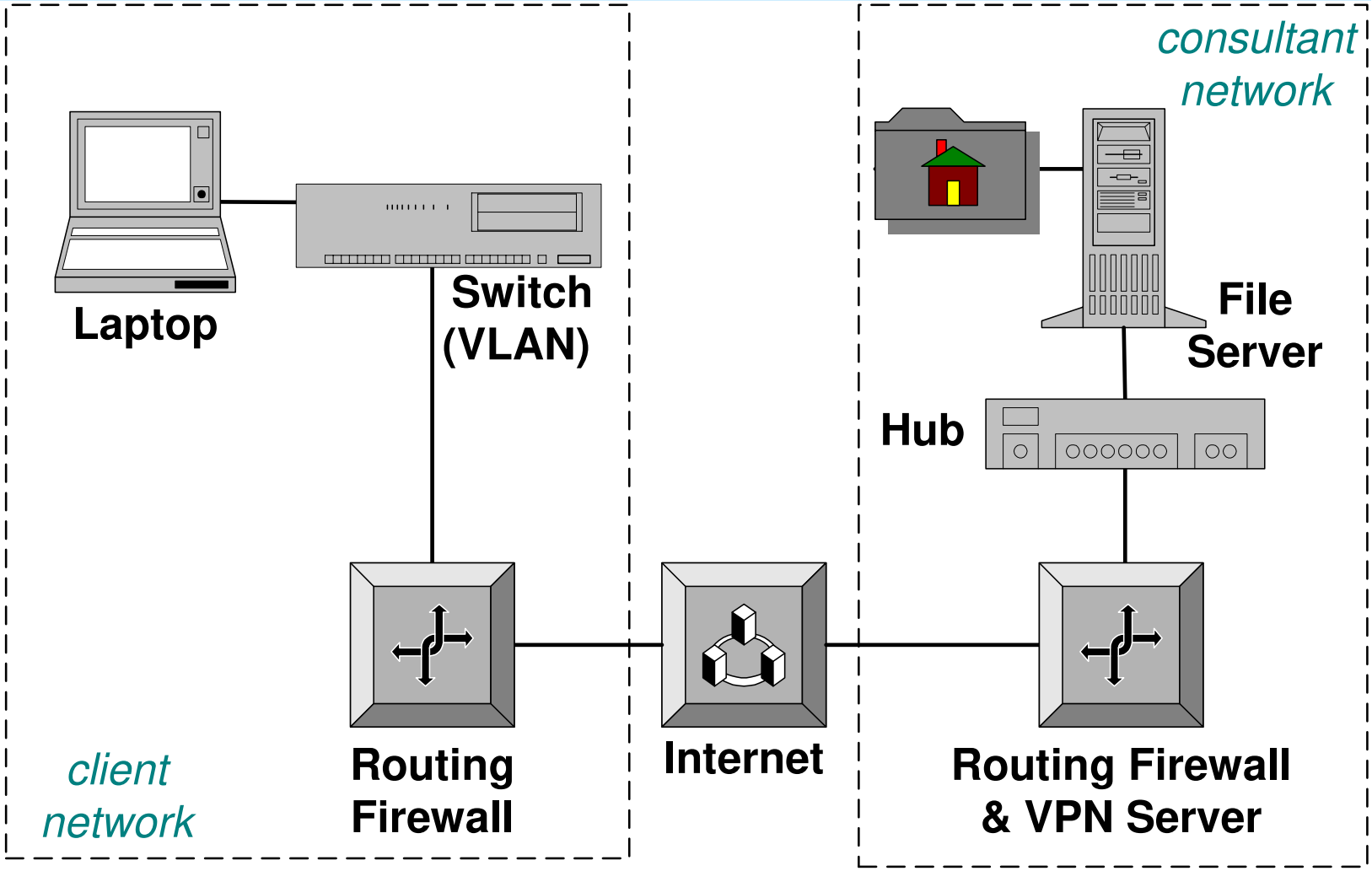
Client

- No visitor access to internal hosts
  - *switch, router, physical configuration*
- Restricted visitor Internet access
  - *firewall configuration*

Consultant

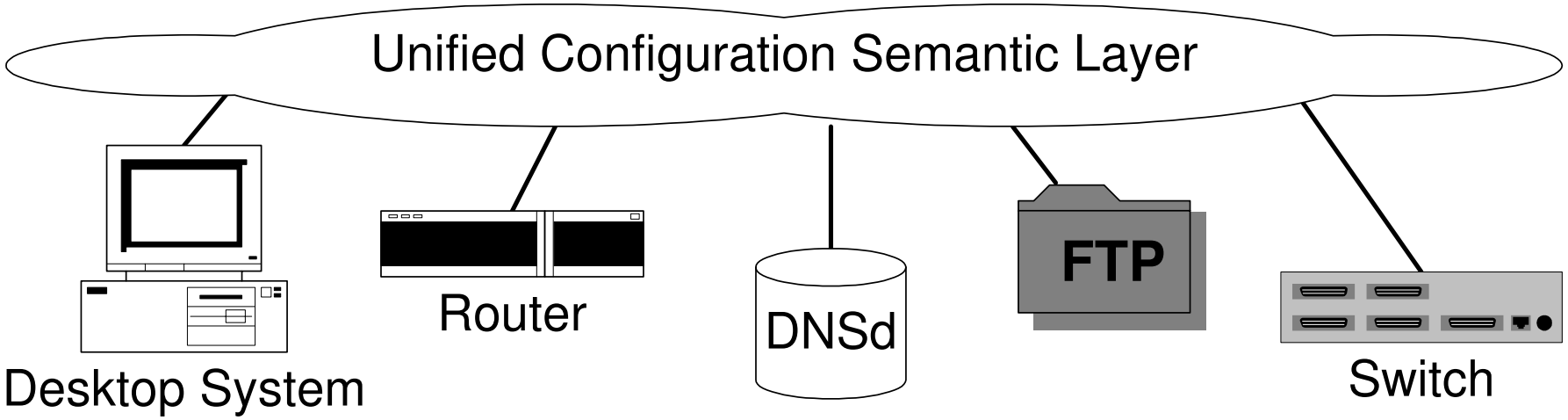
- VPN clients obtain restricted file access
  - *file, http, ftp server configuration*

Example Network (Detail)

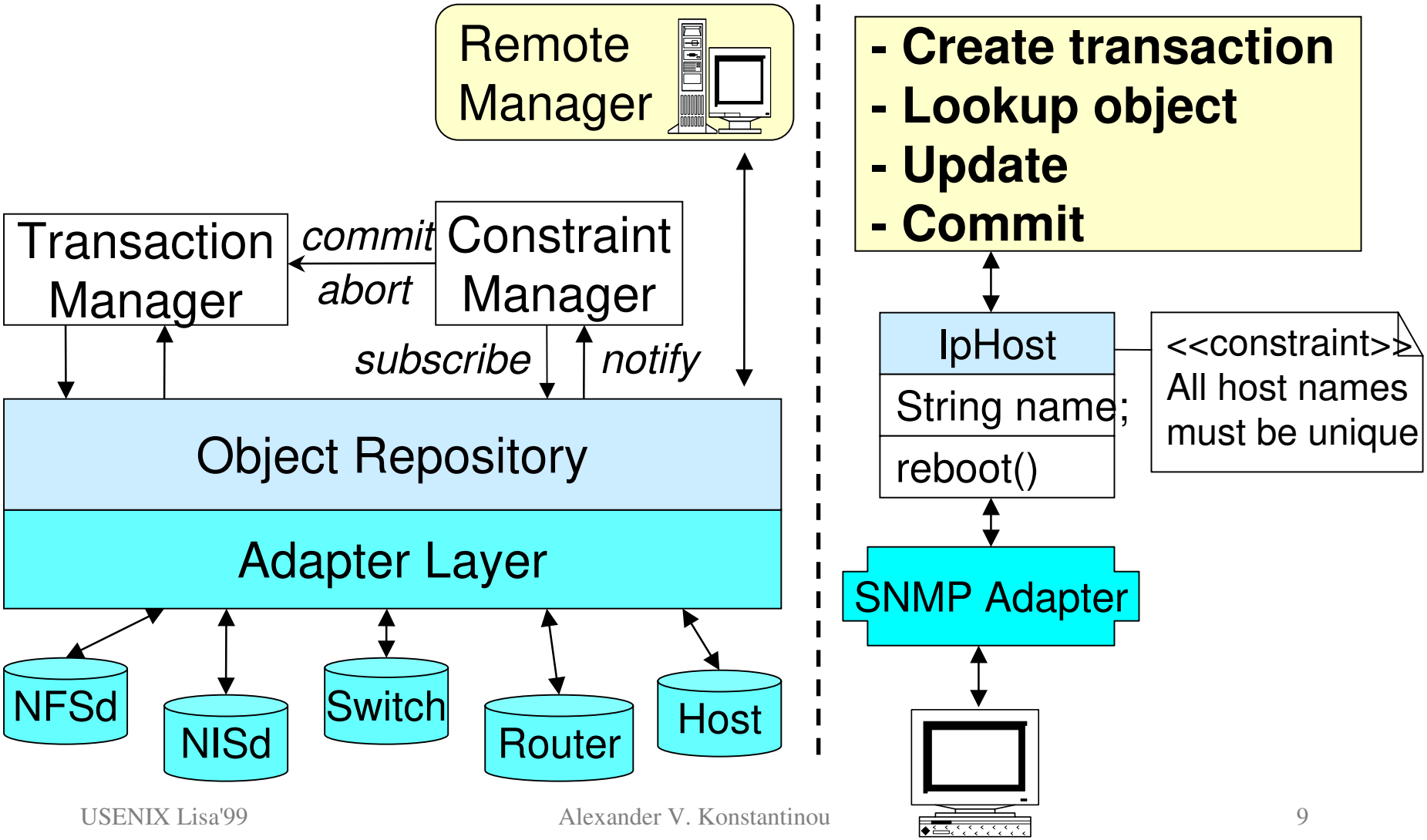


Solution : Unified Configuration Semantic Layer

- Unified object-relationship configuration model
- Consistency rules
- Change propagation
- Rollback and recovery



NESTOR: An Architecture for Network Self Management & Organization



Integrity Constraint Example

- Constraints expressed in OCL (Object Constraint Language -- part of UML)
- Example : “All nodes connected to an internal VLAN port should be *trusted* ”

```
EthernetVlanSwitchPort->allInstances
->select(port | port.isEnabled)
->forAll(port |
 if (port.securityMgr.isTrusted(port.vlanID))
   port.forwardsNodes->forAll
     (node | node.securityMgr.isTrusted(node))
```

Policy Script Example

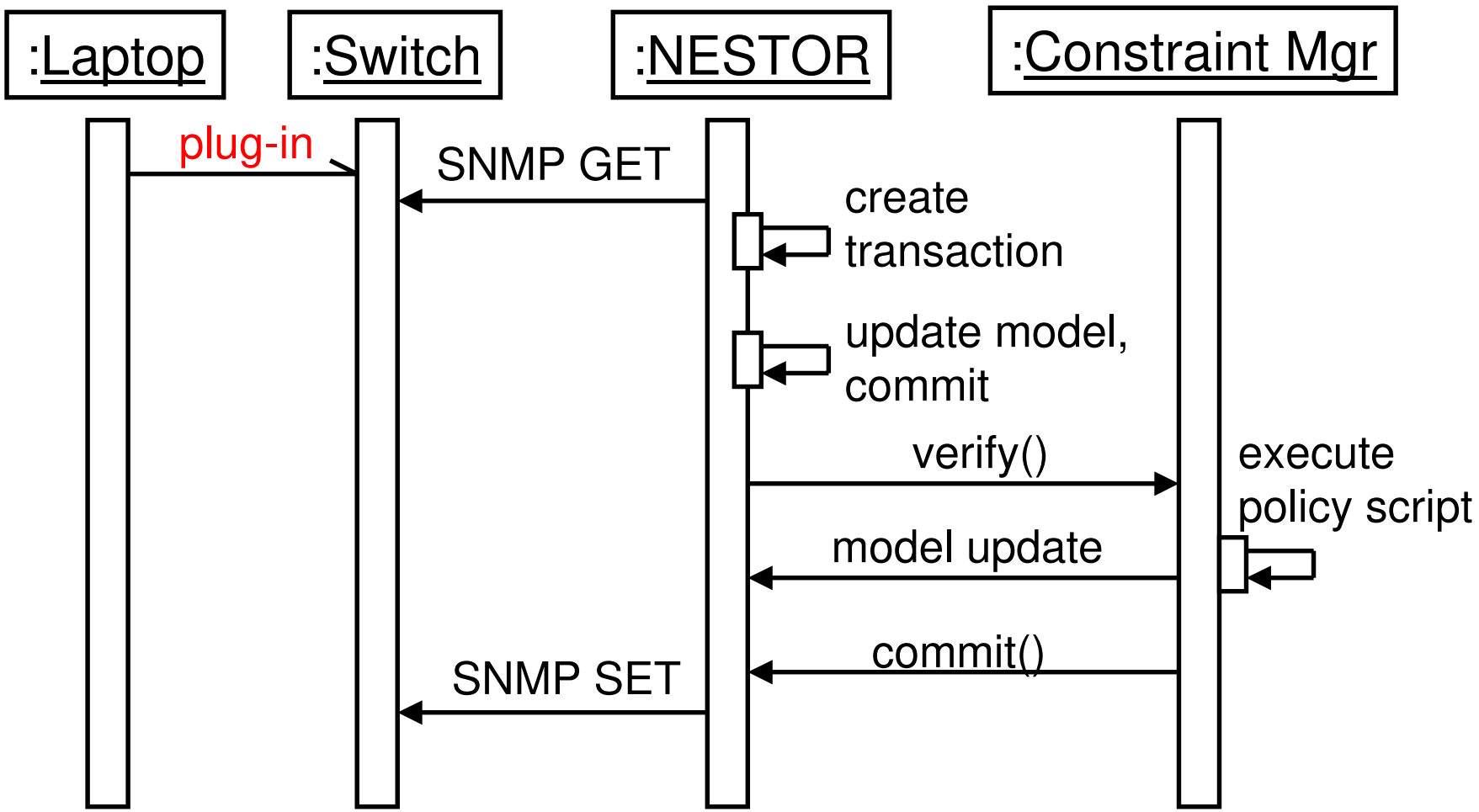
- Constraint violations handled by policy scripts (Java methods)
- Example (cont.): policy script changes the VLAN id of the violating port

```
public void constraintHandler
   (Object[] stack, Transaction trans) {
   EthernetVlanSwitchPort port =
      (EthernetVlanSwitchPort) stack[1];
   port.vlanID =
         port.securityMgr.getPublicVlanID();
}
```

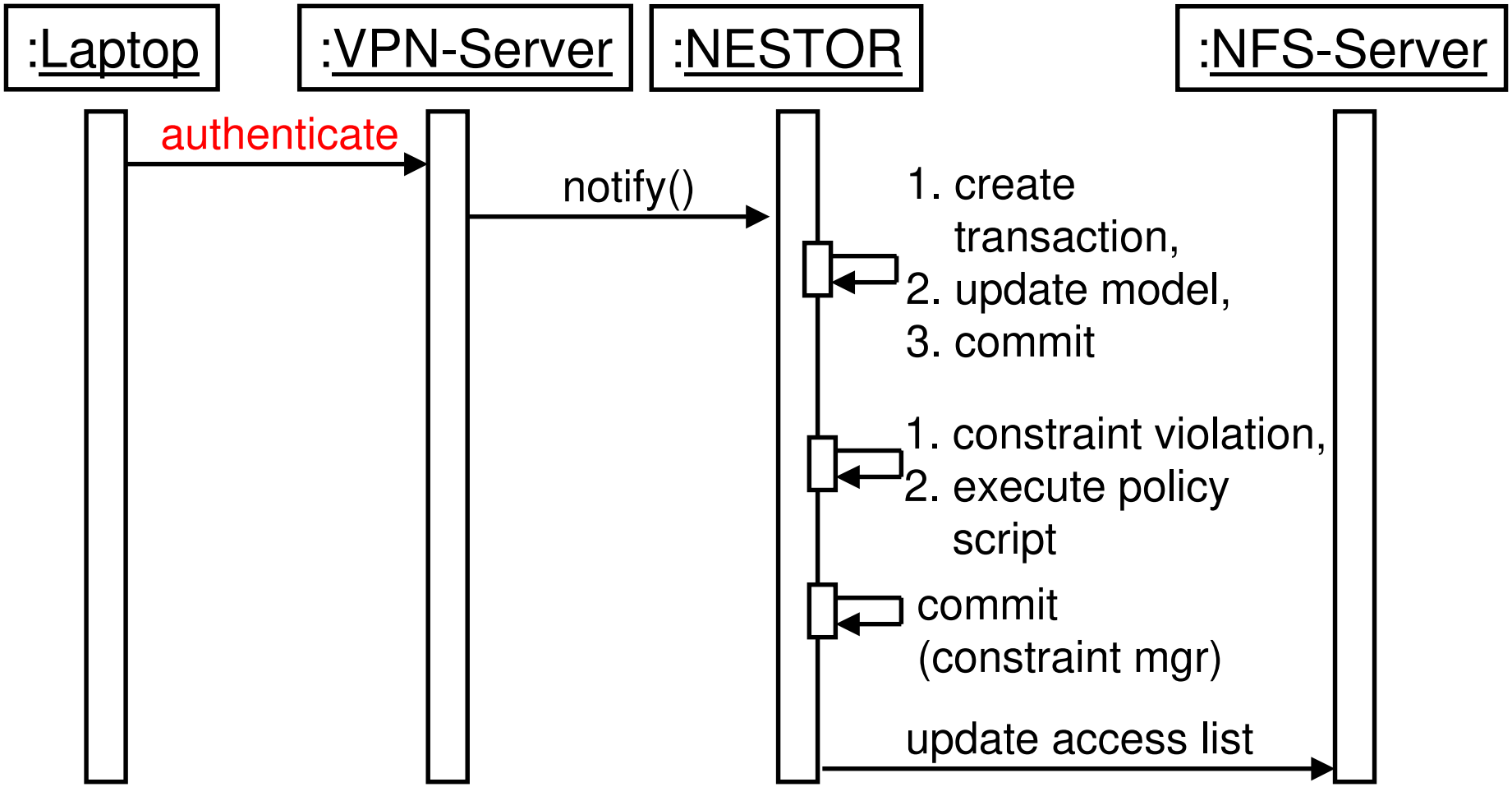
Dynamic Network Example Revisited

- High-level security policies
- Model network elements & services
- Instrument model interfaces
- Policies as constraints on configuration
- Policy scripts for change propagation
- Deploy and populate NESTOR server

Laptop Plug-In Interactions



Laptop Plug-In Interactions (2)



Summary

- Dynamic network challenges
- Solution: unified configuration semantic layer
- NESTOR architecture
- Policy-based dynamic network configuration

Future SA role: defining policies for change propagation

Future Work

- Translating high-level security policies to constraints on configuration (Telcordia)
- Model evolution (Telcordia project on reconfiguring networks of firewalls)
- Scalability
- NESTOR security model
- Distributing NESTOR/pushing down to device

Managing Security in Dynamic Networks

Alexander V. Konstantinou
akonstan@cs.columbia.edu

http://www.cs.columbia.edu/dcc/nestor

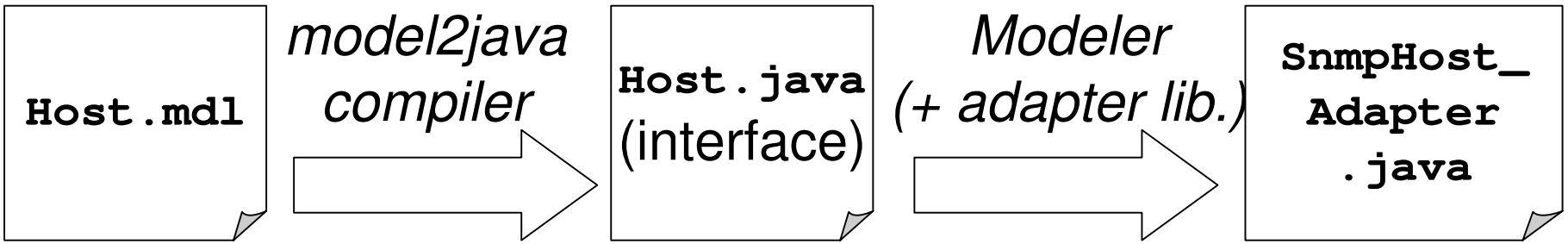
Yechiam Yemini (yemini@cs.columbia.edu)
Sandeep Bhatt (bhatt@research.telcordia.com)
S. Rajagapalan (sraj@research.telcordia.com)

Backup Slides

Configuration Modeling

- Model expressed in the MODEL language (SMARTS)
- MODEL extends IDL with relationships, problems ...

```
interface nestor::IpHost : nestor::ManagedObject {
    attribute String hostname "Name of host";
    relationshipset interfacedThrough,
                    IpNetworkInterface, partOf;  }
```



NESTOR Transactions

- Proxy repository objects
  - Implement model interfaces
  - Log all access
  - Updates not pushed to device
- Transaction commit
  - Effect all changes on proxy objects to adapter objects (same order)
  - On failure, roll-back
  - On roll-back failure, note in recovery log