

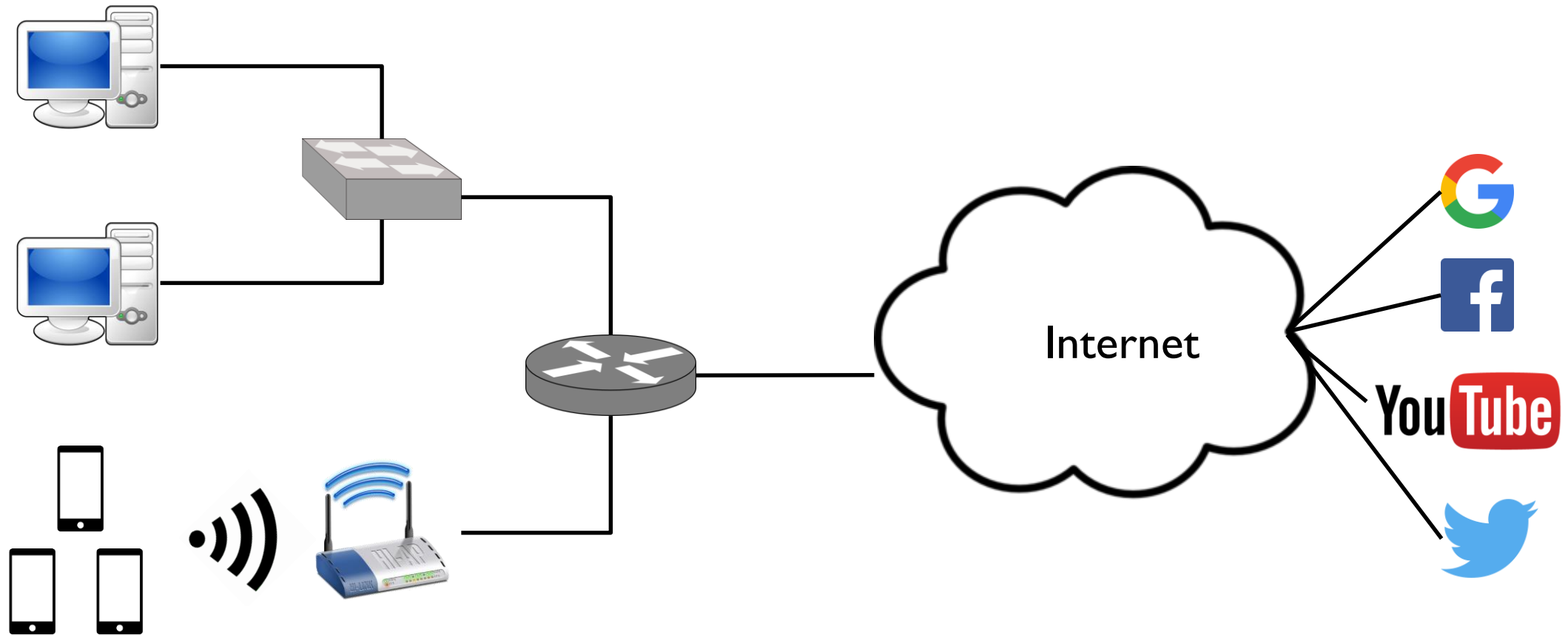
Management & Orchestration of Service Function Chains

Shihabur R. Chowdhury

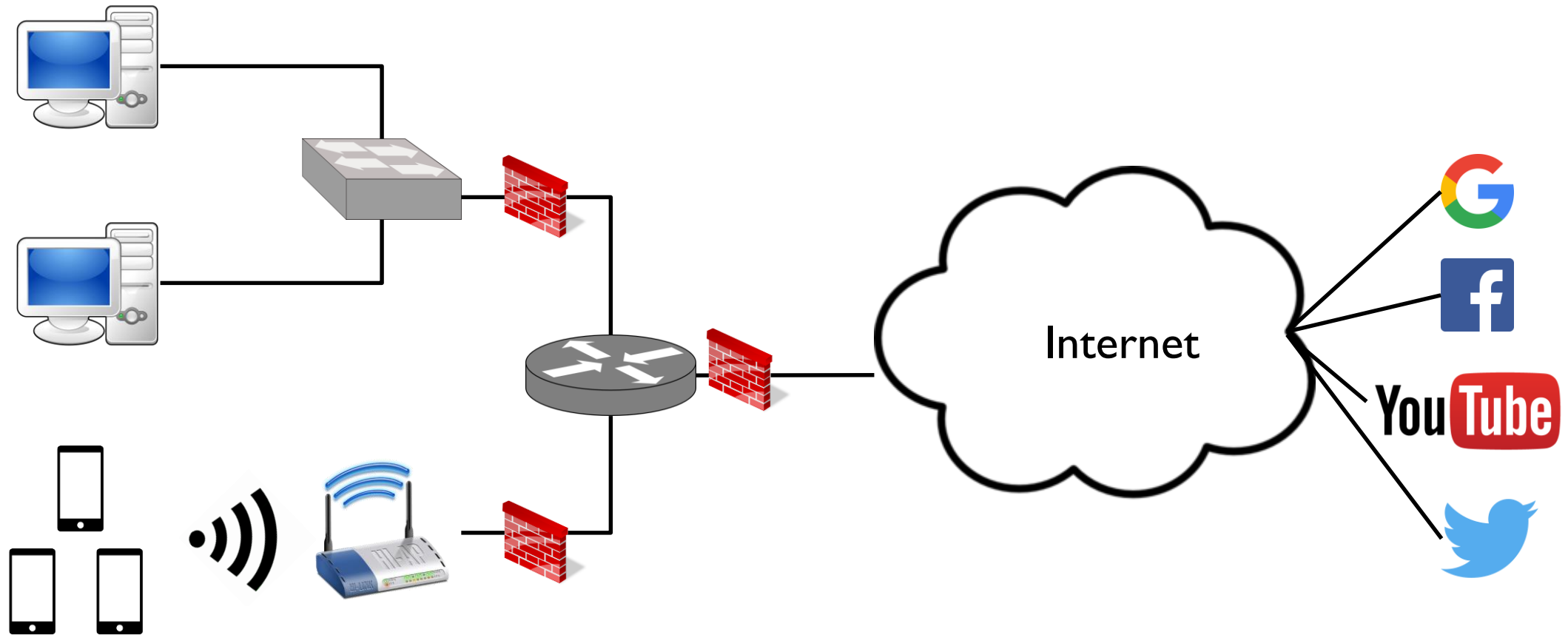


UNIVERSITY OF WATERLOO
FACULTY OF MATHEMATICS
David R. Cheriton School
of Computer Science

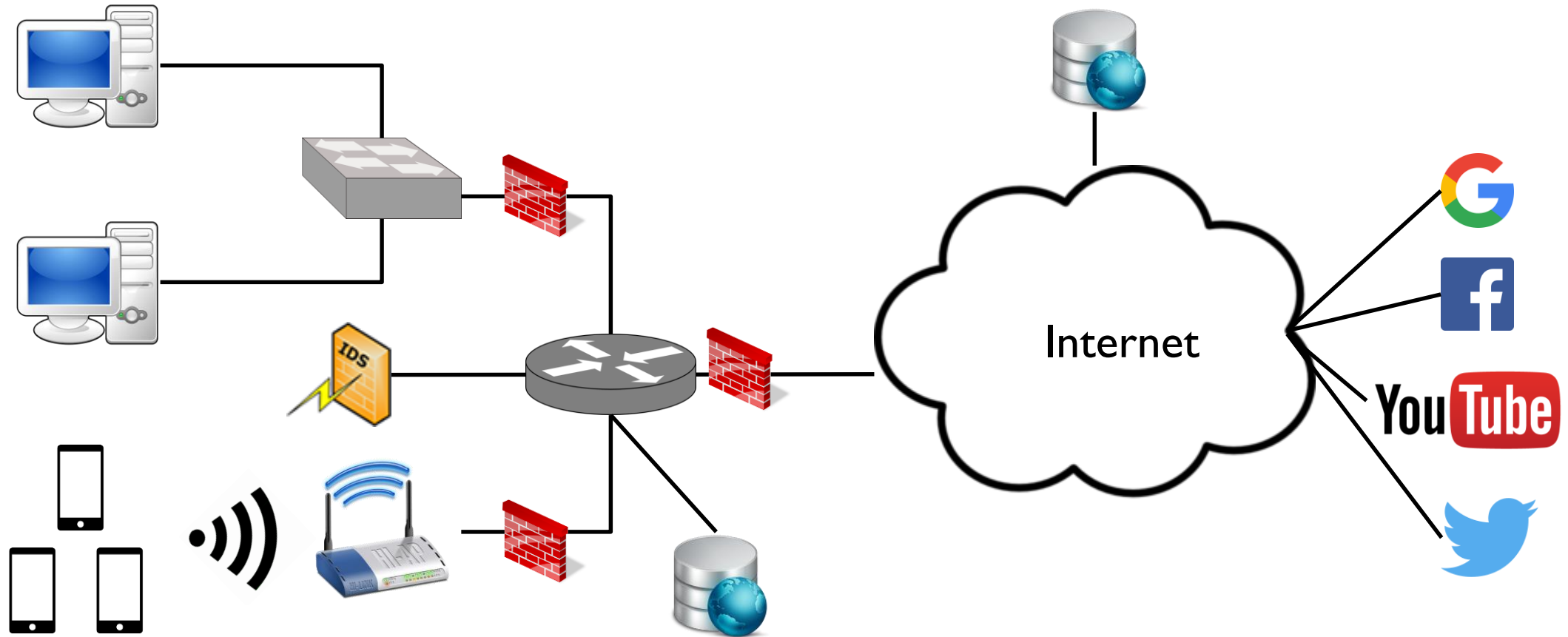
Network Function & Middlebox



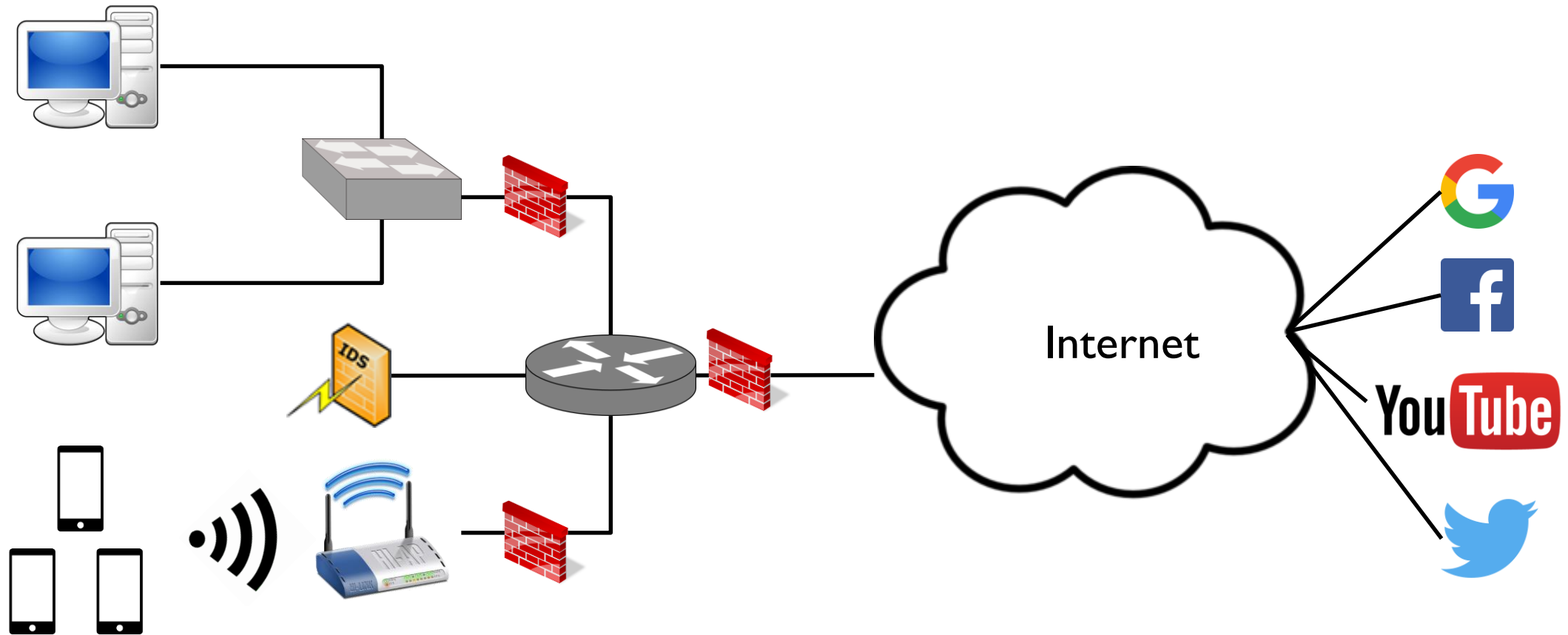
Network Function & Middlebox



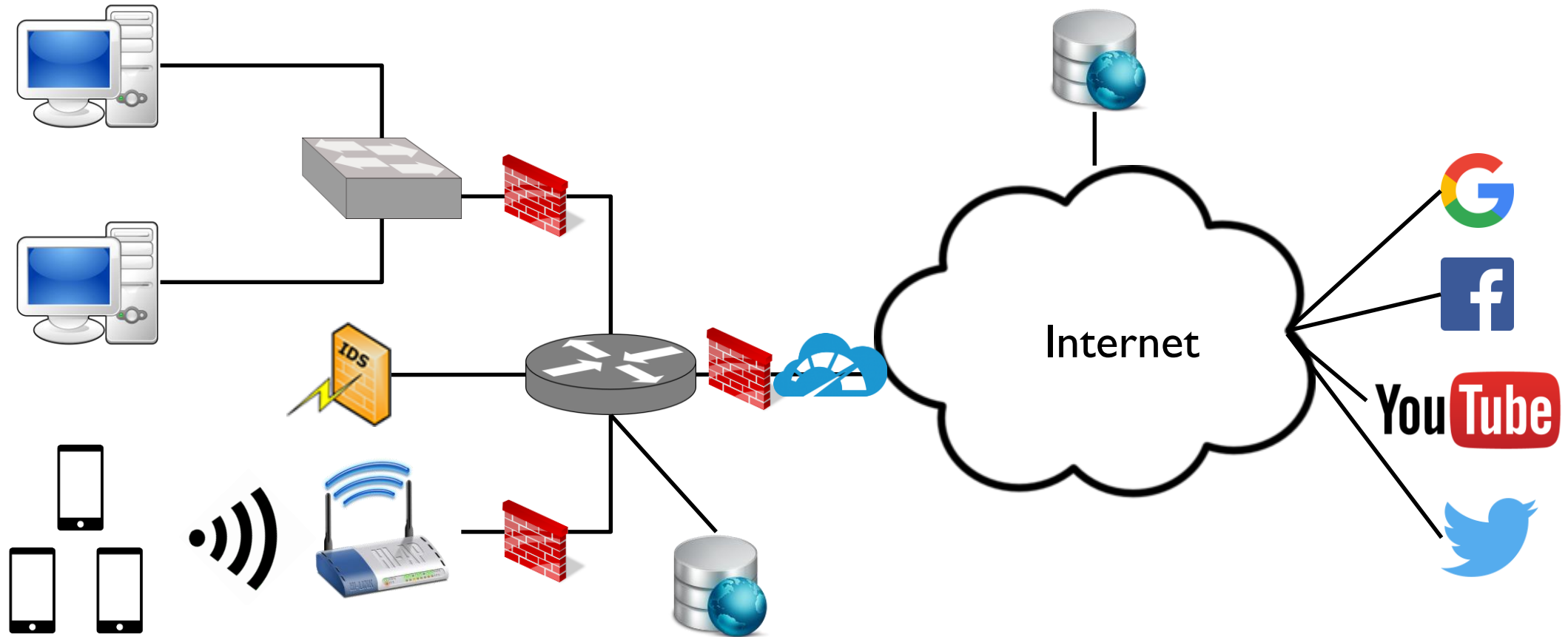
Network Function & Middlebox



Network Function & Middlebox



Network Function & Middlebox



What's the Fuss About Middleboxes?

Expensive: High CAPEX & OPEX

Proprietary & Vertically Integrated: Vendor lock-in

What's the Fuss About Middleboxes?

Expensive: High CAPEX & OPEX

Proprietary & Vertically Integrated: Vendor lock-in



As many middleboxes as routers and switches in enterprise networks*

* J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar. Making middleboxes someone else's problem: network processing as a cloud service. ACM SIGCOMM 2012, pp. 13-24

Networks are full of *vertically integrated* and *proprietary* devices that *slow down innovation* and are *expensive to buy and maintain*

Service Function Chain (SFC)

Network policies often require packets to traverse an ordered set of Network Functions



e.g., Service chain for a Web Service



e.g., Service chain for 3G Data Network

SFC with Hardware Middleboxes

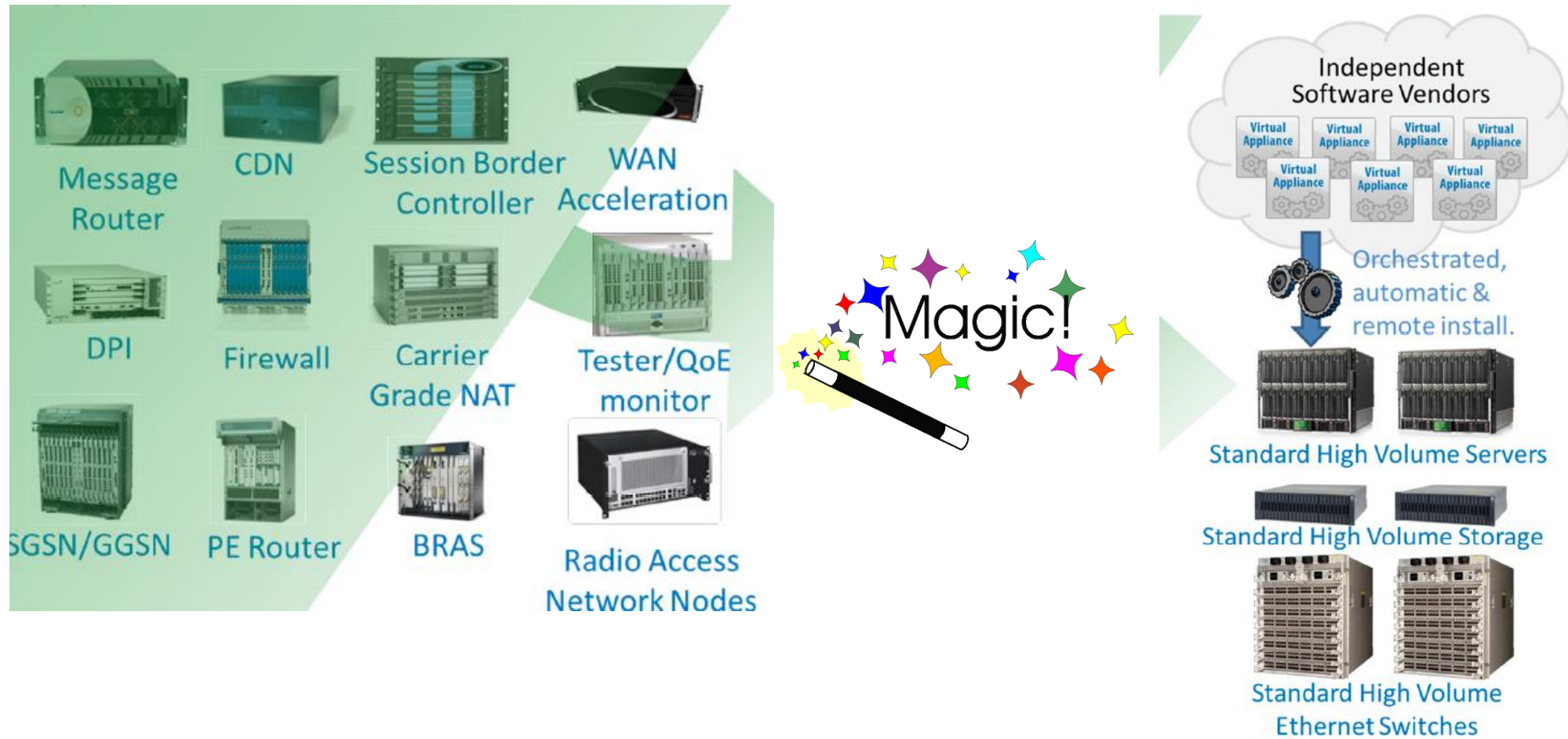


H/W Middleboxes have fixed locations



Fixed locations restrict routing optimization

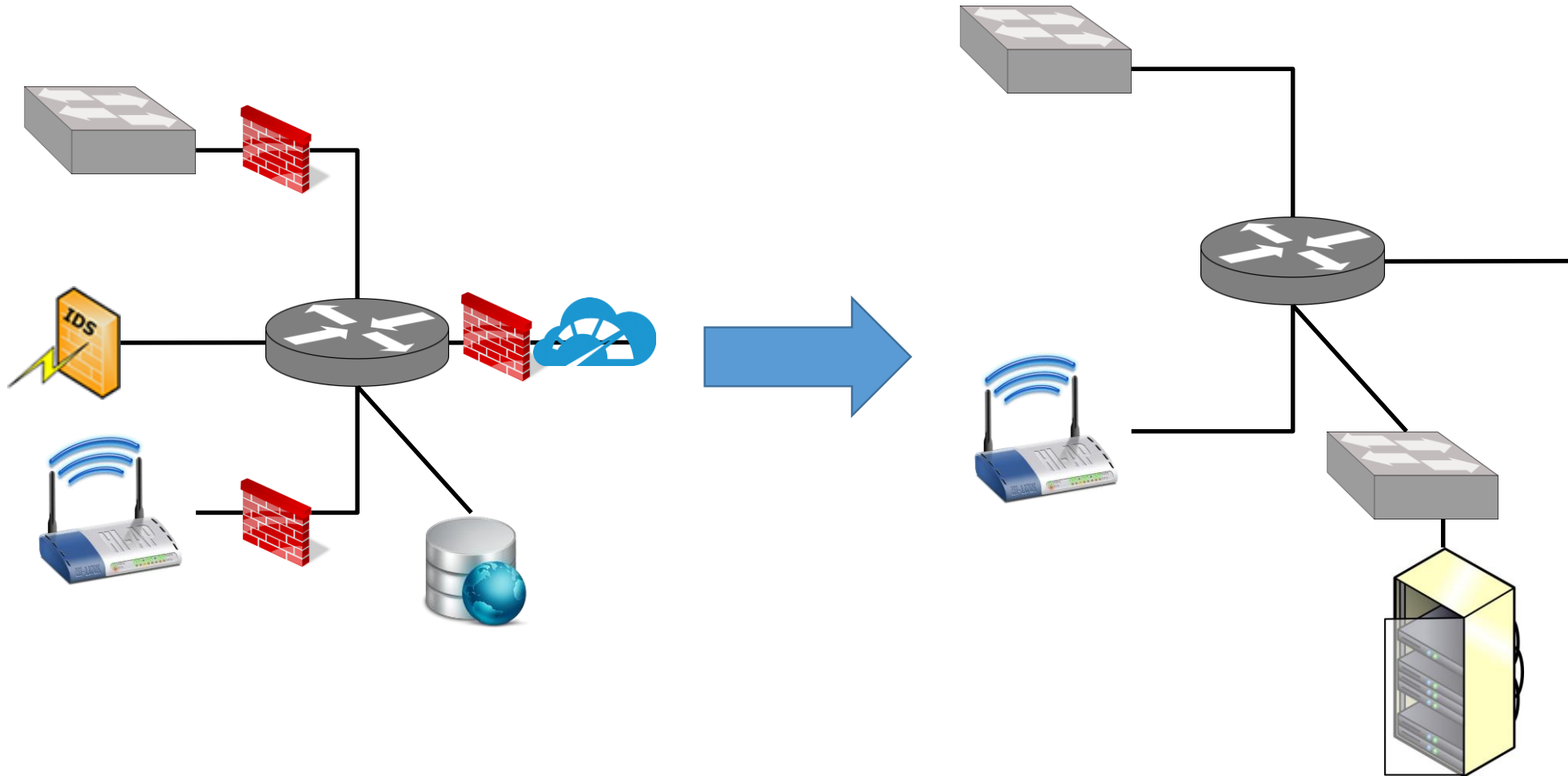
Network Function Virtualization (NFV)



Hardware Middleboxes

NFV

Network Function Virtualization



Hardware Middleboxes vs. NFV

Hardware Middleboxes

Provisioned for peak traffic

Fixed network attachment
point

Limited traffic steering
optimization

NFV

Provisioned on-demand

More flexible network
attachment point

Optimized VNF* placement
and traffic steering

*VNF = Virtual Network Function, i.e., the virtualized middlebox

Our Contributions

VNF Orchestration: Coordinated compute resource allocation and traffic steering Service Function Chains*

nf.io: Management interface for deploying, configuring, and monitoring VNFs and chains**

* M.F. Bari, S.R. Chowdhury, R. Ahmed, R. Boutaba, and O. Duarte. Orchestrating Virtualized Network Functions. IEEE Transactions on Network and Service Management (To Appear)

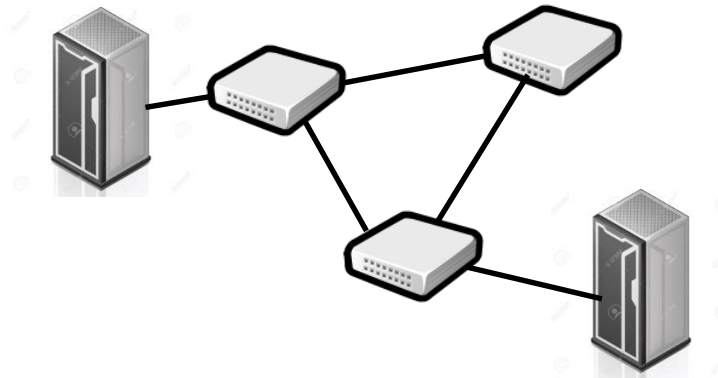
** M.F. Bari, S.R. Chowdhury, R. Ahmed, R. Boutaba. nf.io: A File System Abstraction for NFV Orchestration. IEEE NFV-SDN Conference 2015, pp. 135 - 141

VNF Chain Orchestration

Given



A set of VNF chain requests



A set of Physical Resources
(Servers, Switches, Links)

VNF Chain Orchestration

Determine

How Many VNFs to deploy?

Where to deploy the VNFs?

How to route traffic between VNFs?

VNF Chain Orchestration

Determine

How Many VNFs to deploy?

Where to deploy the VNFs?

How to route traffic between VNFs?

VNF Chain Orchestration

Determine

How Many VNFs to deploy?

Where to deploy the VNFs?

How to route traffic between VNFs?

VNF Chain Orchestration

Determine

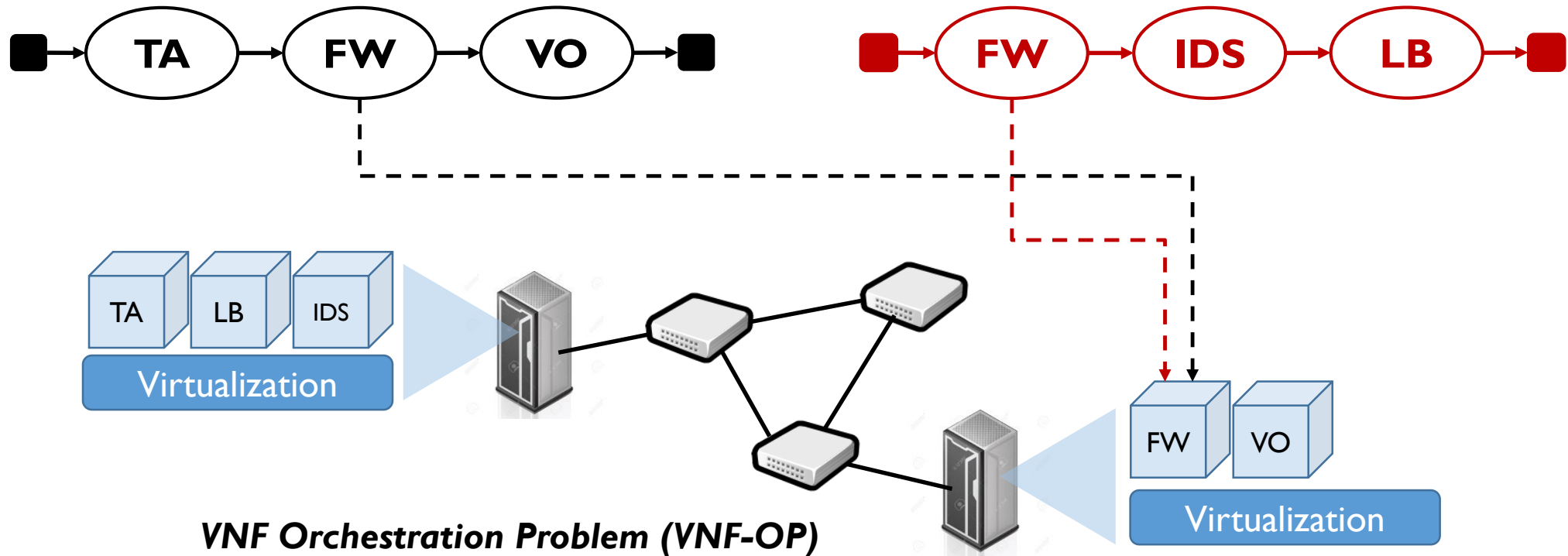
How Many VNFs to deploy?

Where to deploy the VNFs?

How to route traffic between VNFs?

Objective: Minimize Operational Expenditure (OPEX)

VNF Chain Orchestration



Proposed Solution: Optimal

Combination of 3 Problems

Allocate Resource for VM/Containers

Assign VNFs to VMs

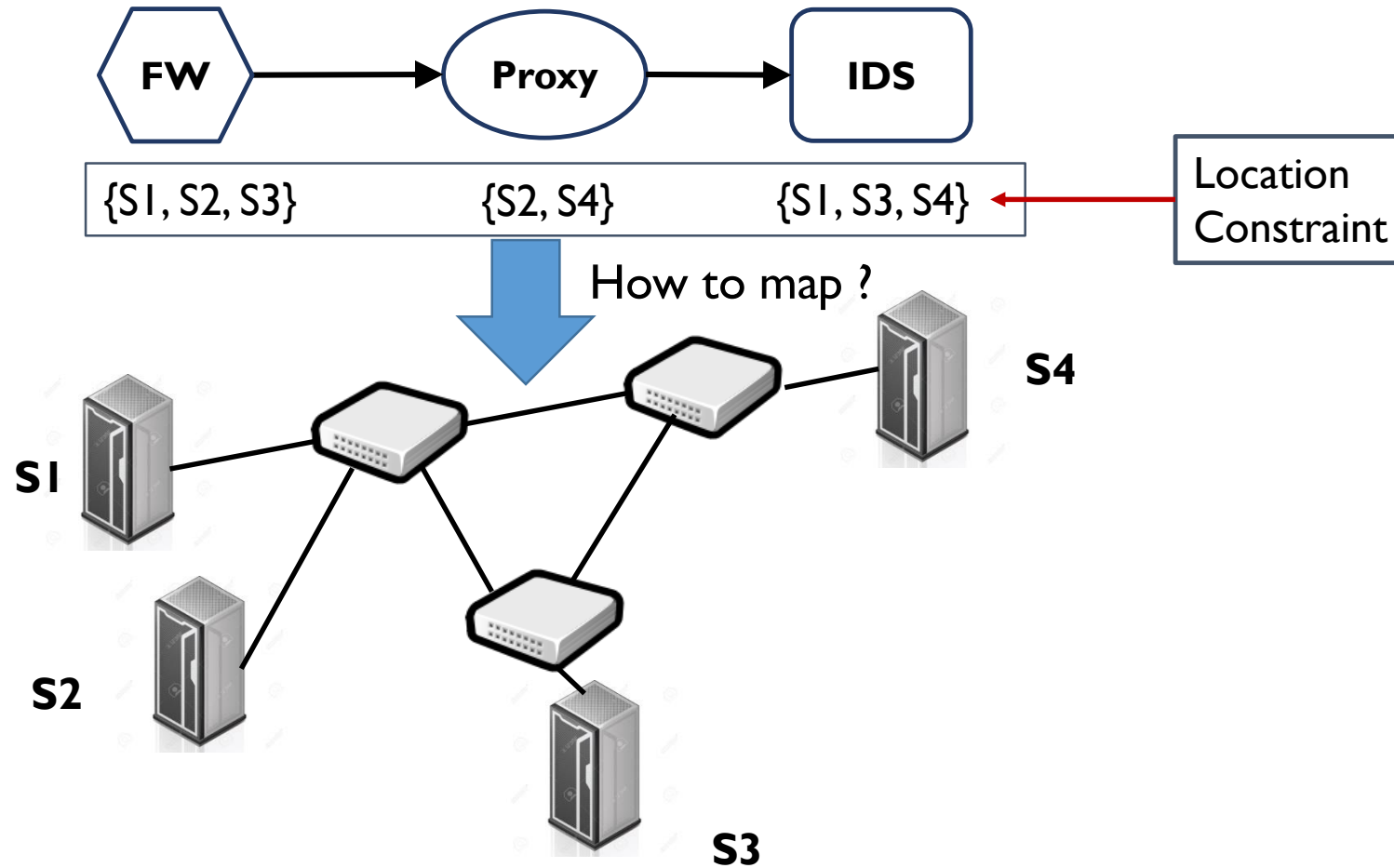
Route traffic between VNFs

Proposed Solution: Optimal

NP-Hard: Reduction from *Capacited Plant Location Problem with Single Source Constraint*

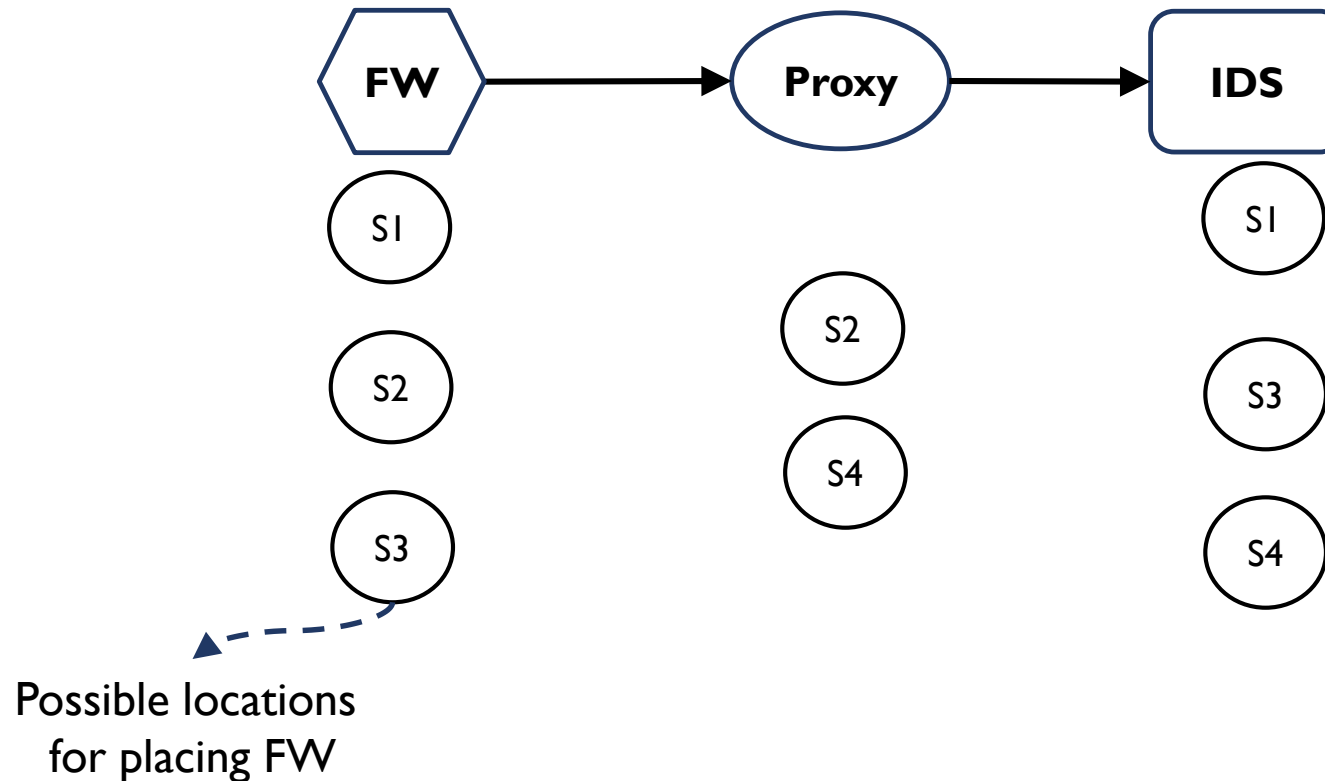
Integer Linear Program (ILP) model for
optimal solution

Proposed Solution: Heuristic



Proposed Solution: Heuristic (cont.)

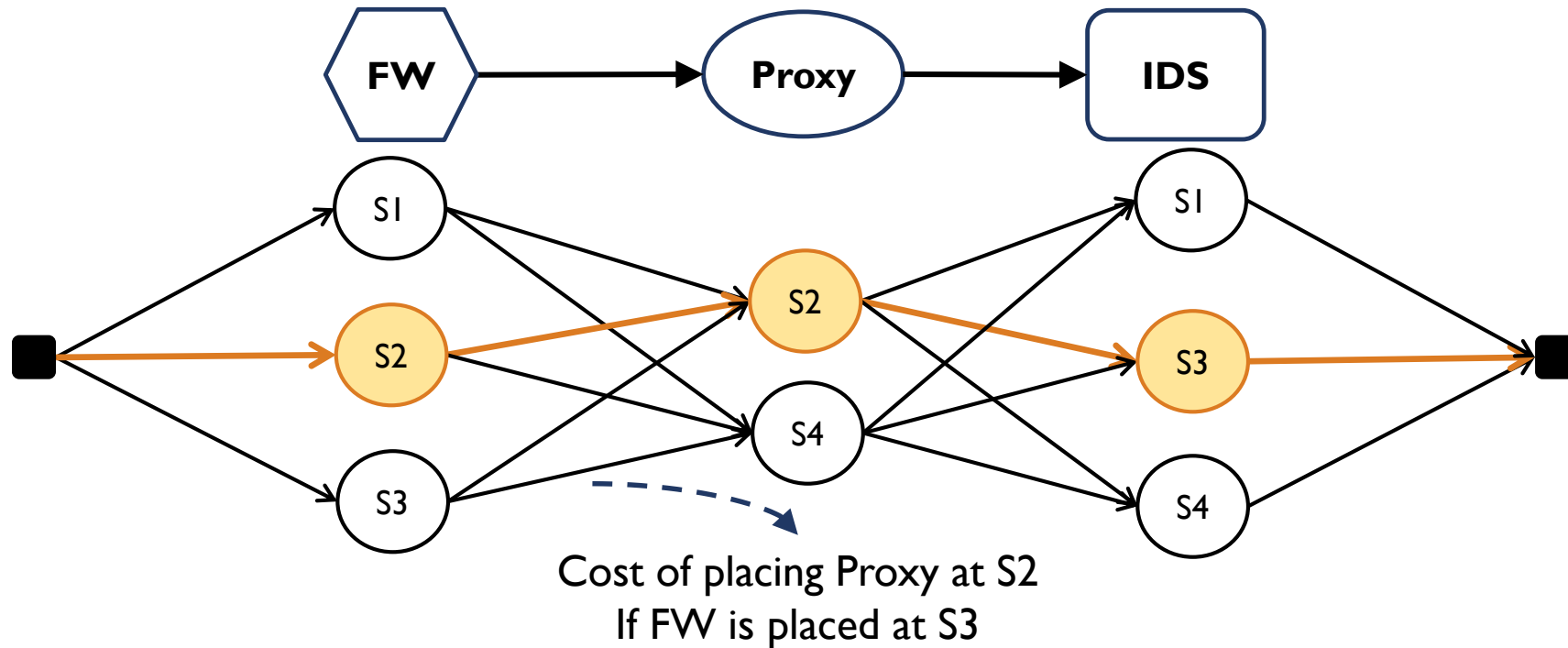
Place location constraints on a grid



Objective: Select exactly one node from each stage and minimize total cost

Proposed Solution: Heuristic (cont.)

Create a multi stage graph (one stage for each VNF)



Objective: Select exactly one node from each stage and minimize total cost

Solution: Use **Viterbi algorithm** to find the minimum cost path from leftmost to right most stage

Evaluation: Setup

Real Data Center and WAN Topologies and Traffic Traces

Server energy ratings from datasheets

Hardware Middlebox data from a Manufacturer

Evaluation: Key Result

Hardware Middlebox

vs.

VNF-OP (Optimal)



vs.

VNF-OP (Heuristic)

Evaluation: Key Result

Hardware Middlebox



vs.

VNF-OP (Optimal)



vs.

VNF-OP (Heuristic)

Evaluation: Key Result

Hardware Middlebox



vs.

VNF-OP (Optimal)

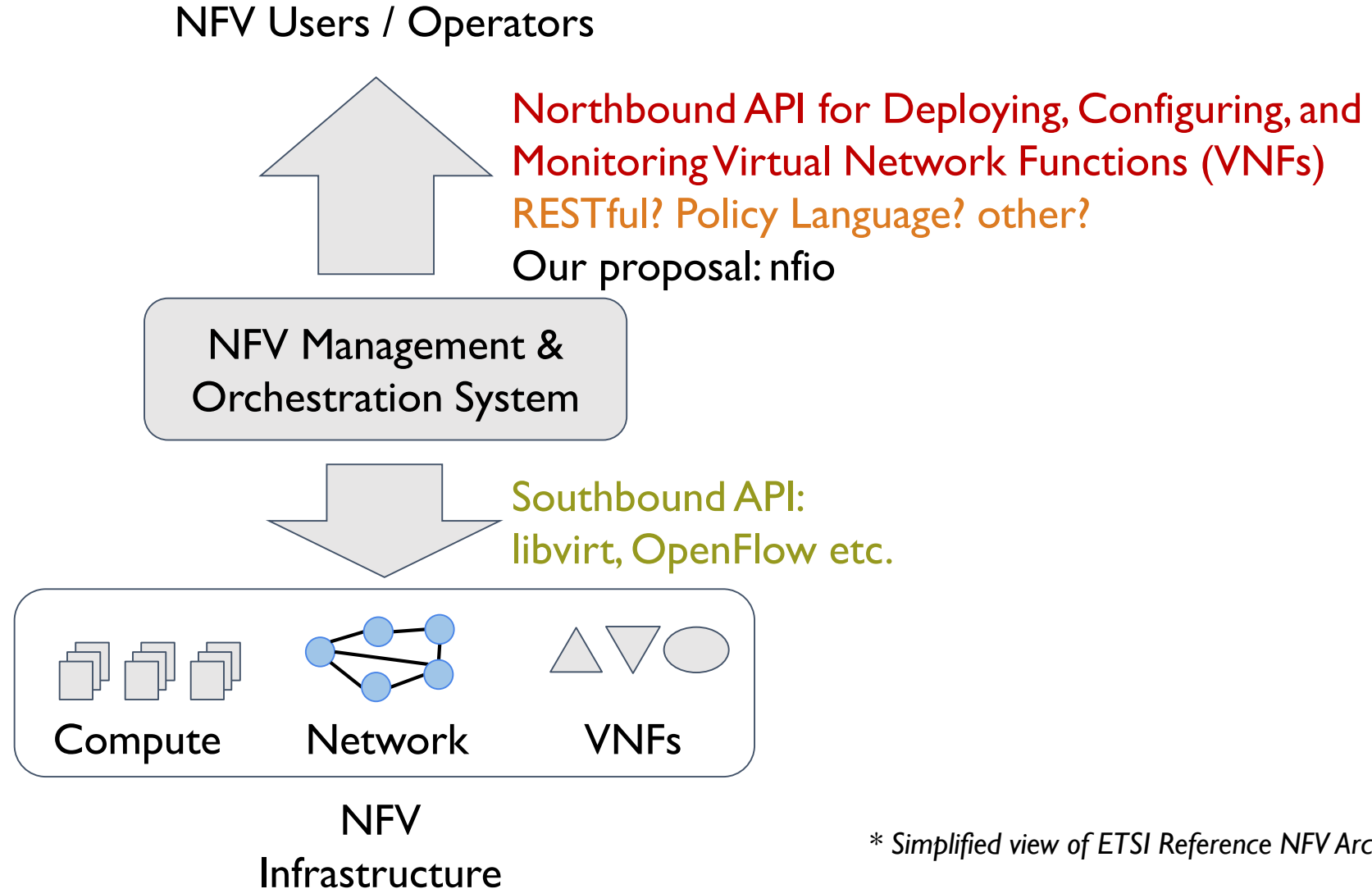


vs.

VNF-OP (Heuristic)



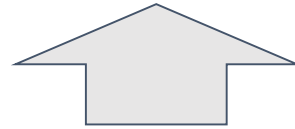
nf.io: NFV Management and Orchestration API



* Simplified view of ETSI Reference NFV Architecture

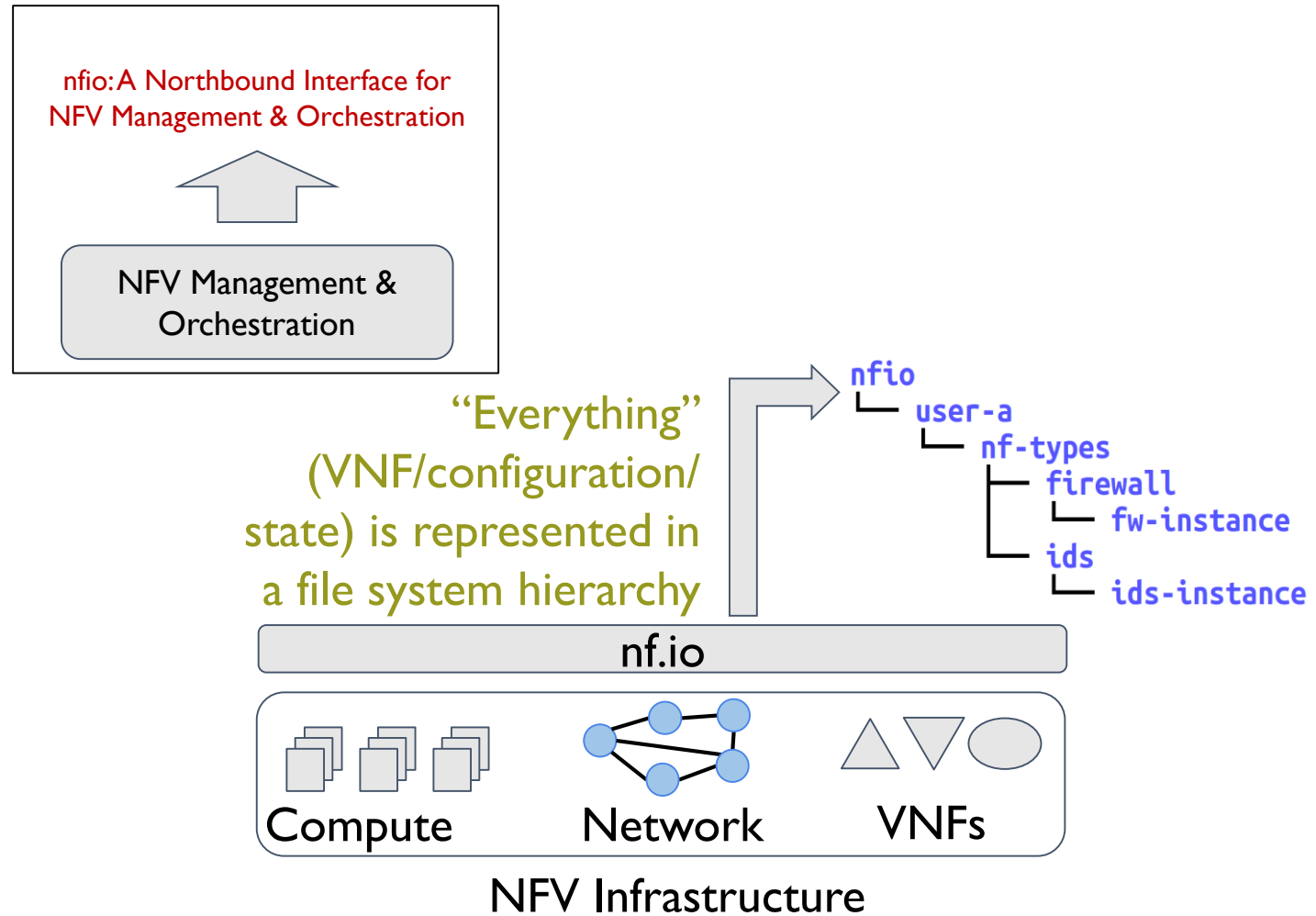
What is nf.io?

nf.io: A Northbound Interface for NFV
Management & Orchestration

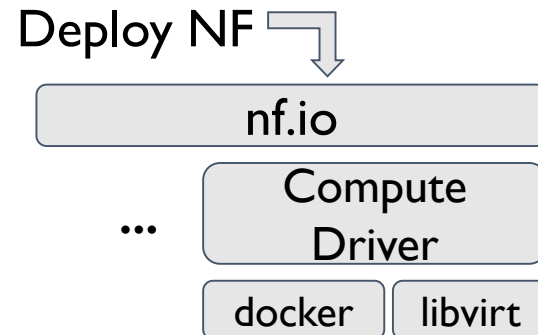
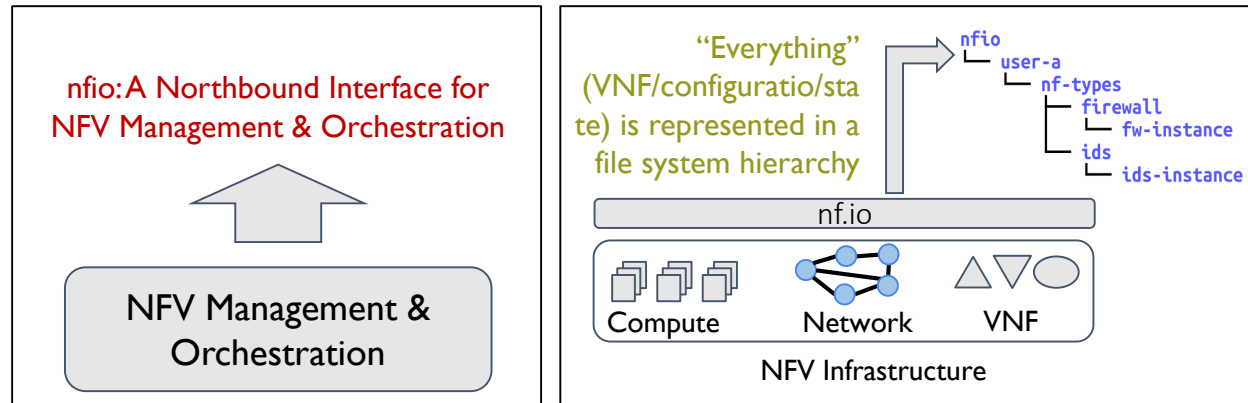


NFV Management &
Orchestration

What is nf.io?

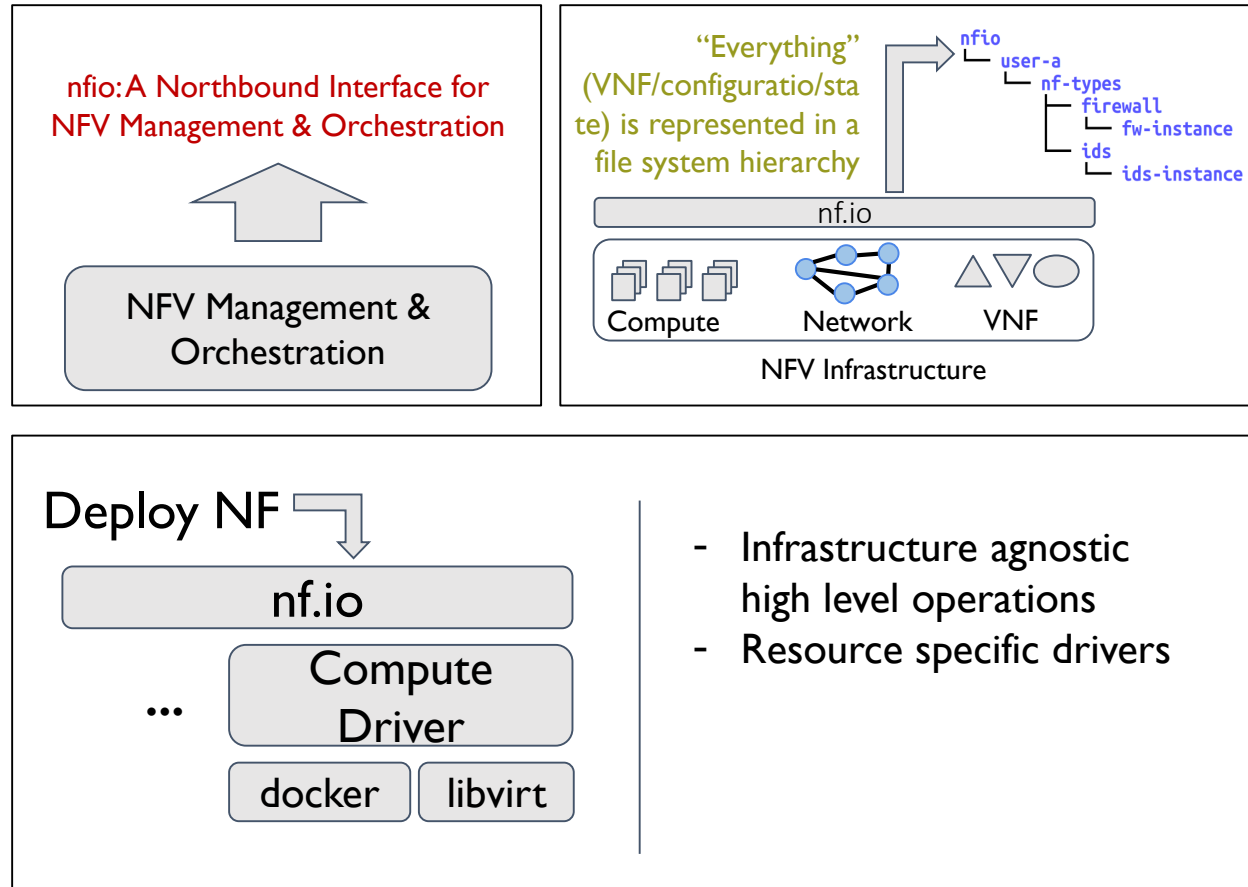


What is nf.io?



- Infrastructure agnostic high level operations
- Resource specific drivers

What is nf.io?



Why File System Abstraction?

Centralized view of a distributed system

Rich set of familiar tools

Extensive management support in Configuration Management Systems

Why File System Abstraction?

Centralized view of a distributed system

Rich set of familiar tools

Extensive management support in Configuration Management Systems

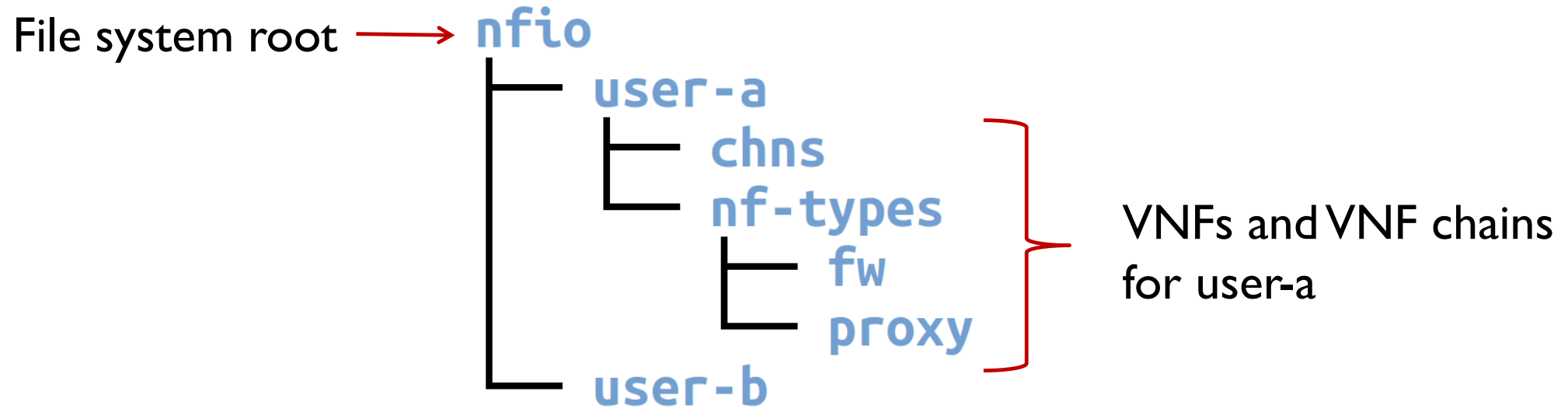
Why File System Abstraction?

Centralized view of a distributed system

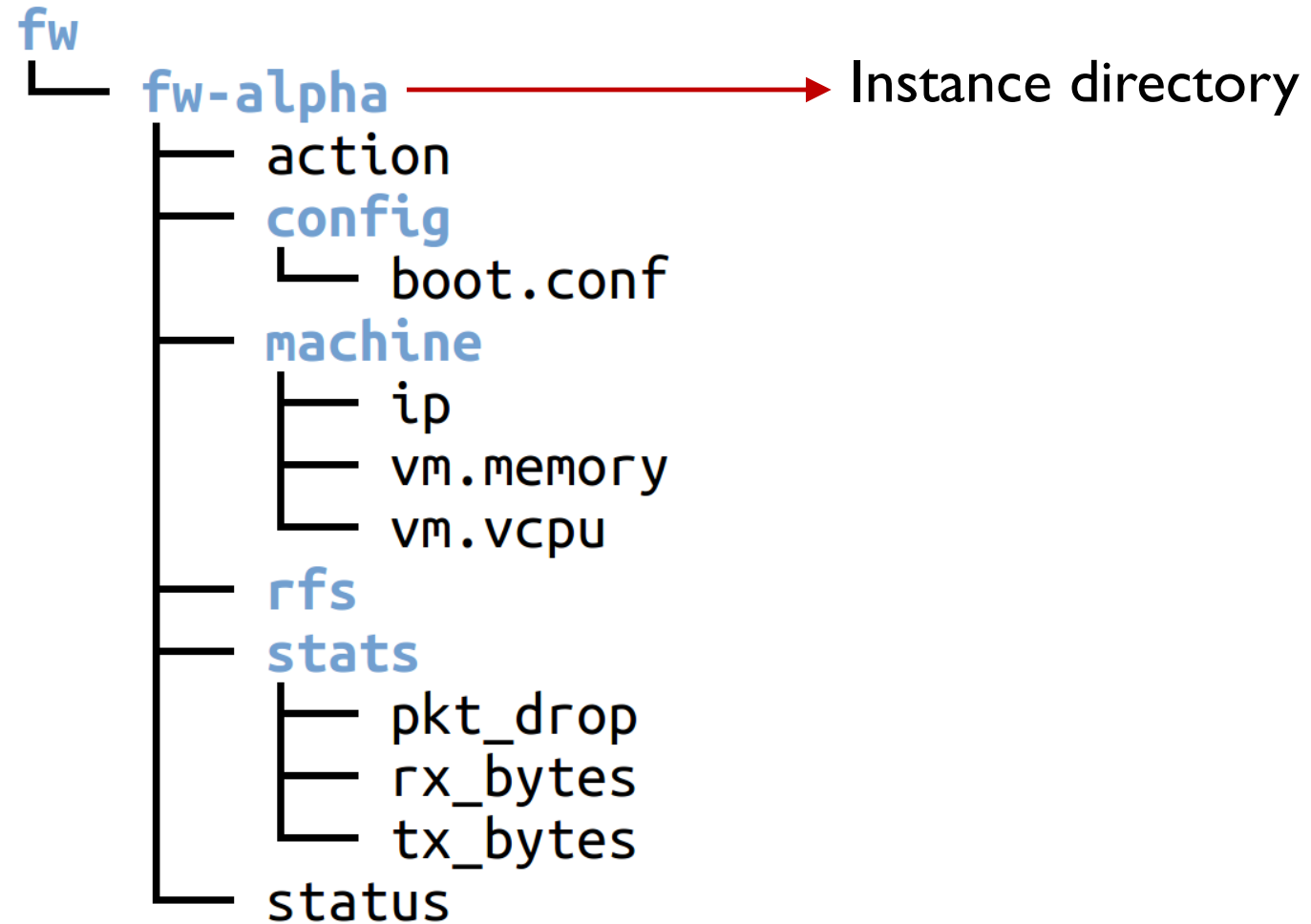
Rich set of familiar tools

Extensive management support in Configuration Management Systems

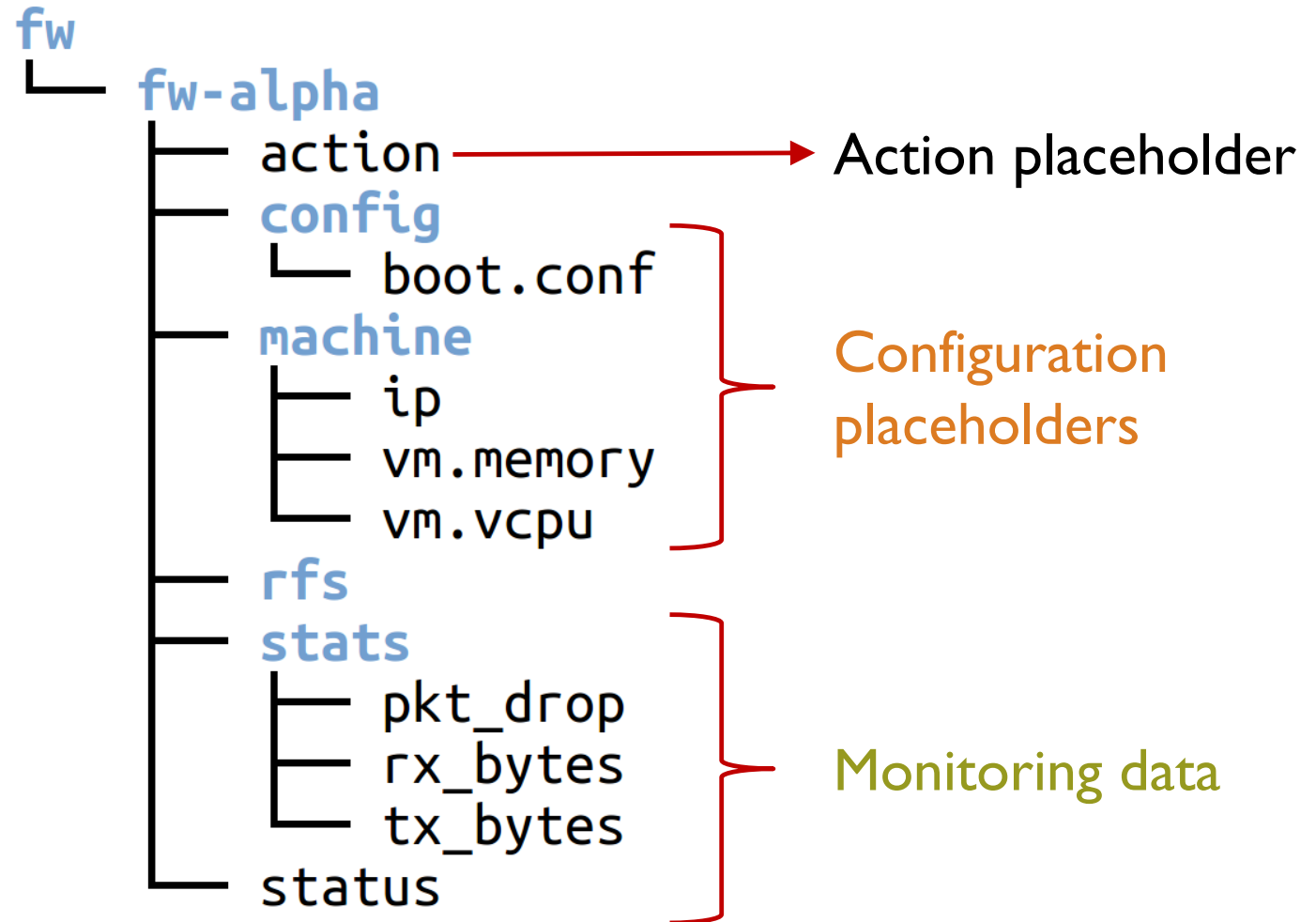
File System Model



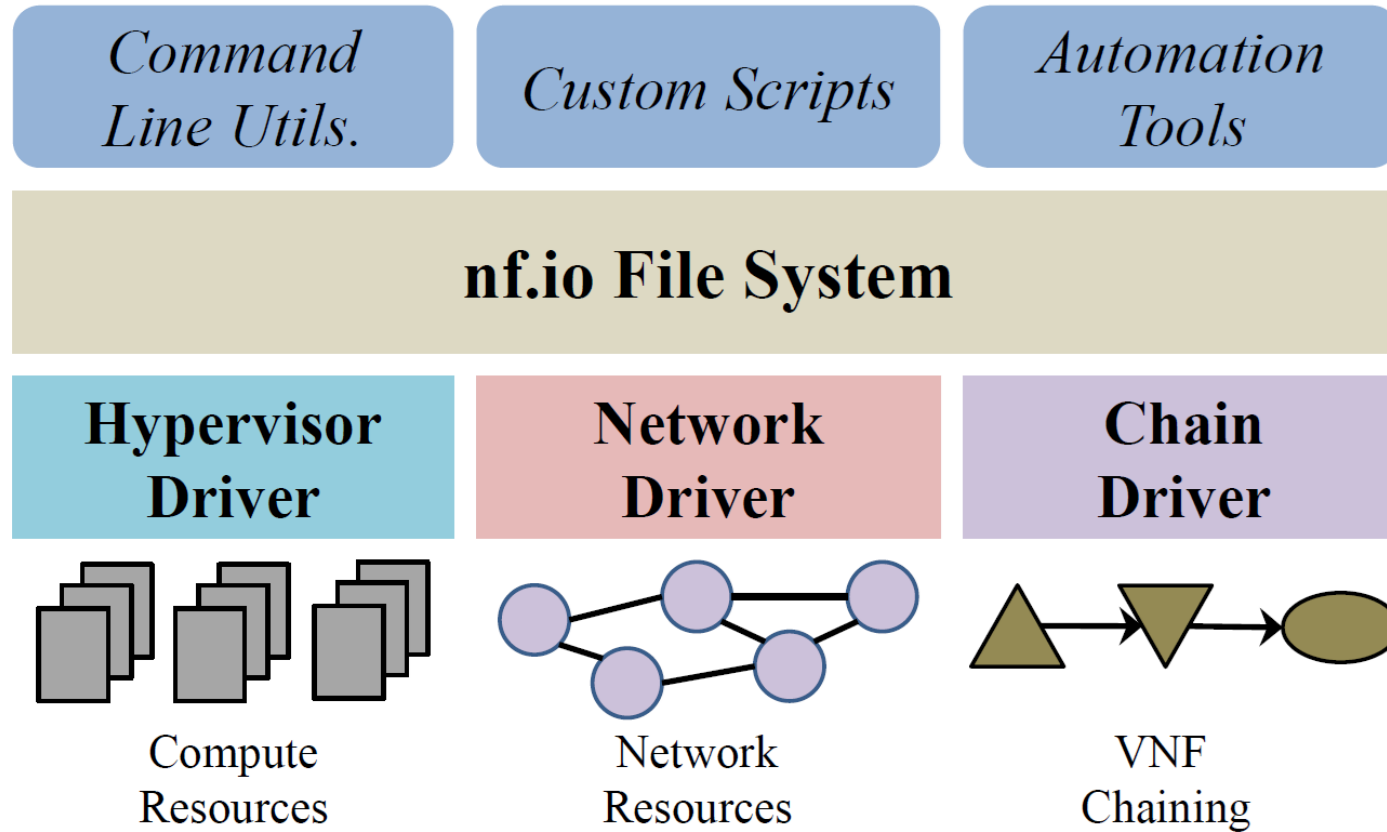
File System Model (cont.)



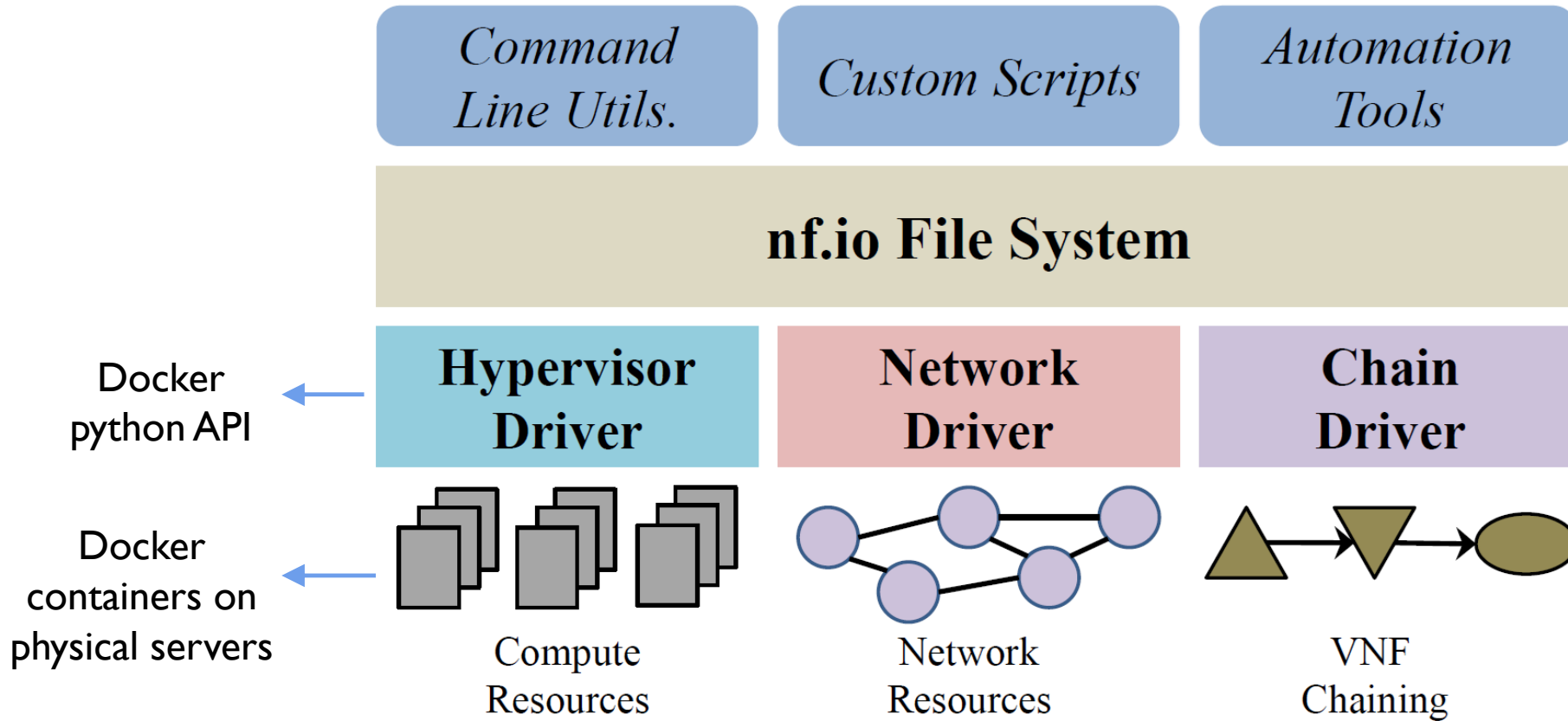
File System Model (cont.)



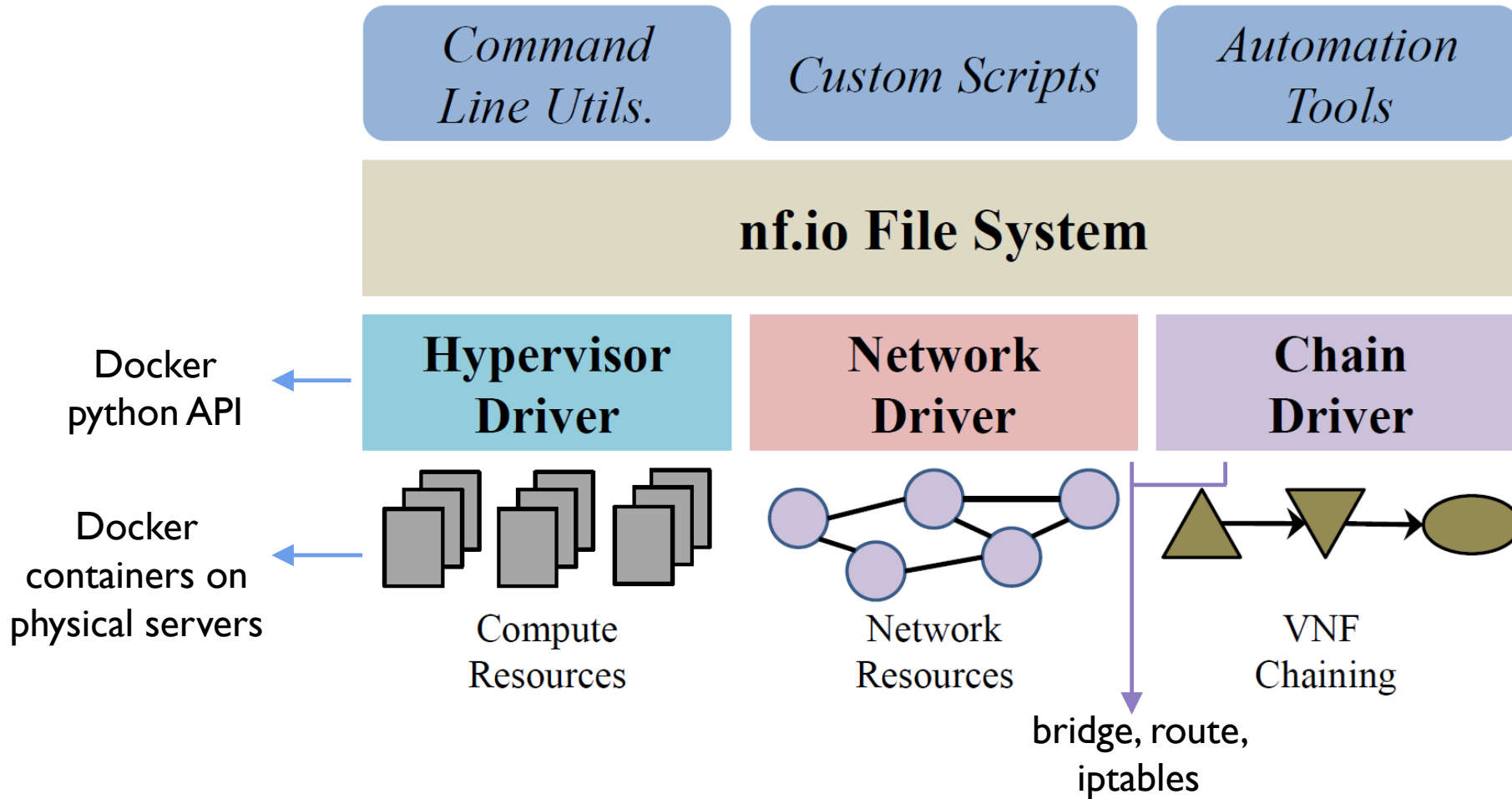
System Architecture



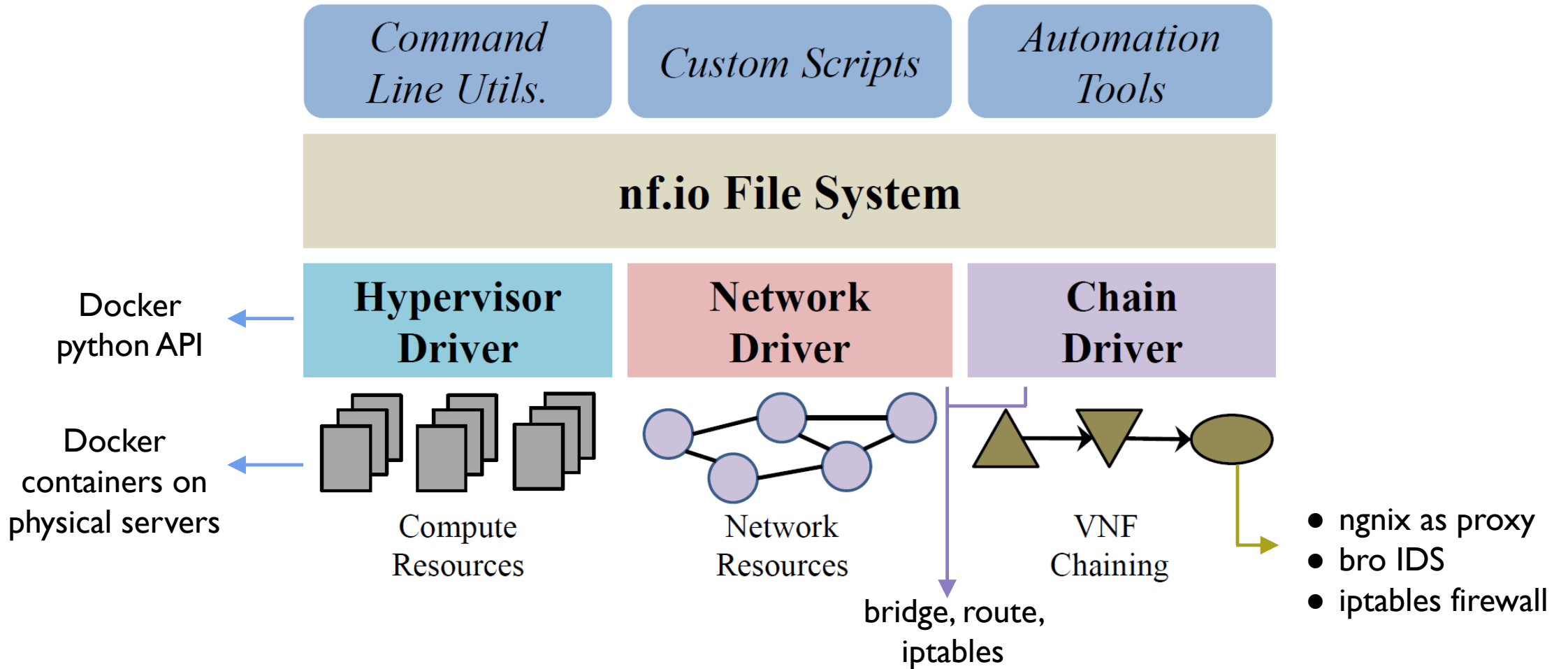
System Architecture



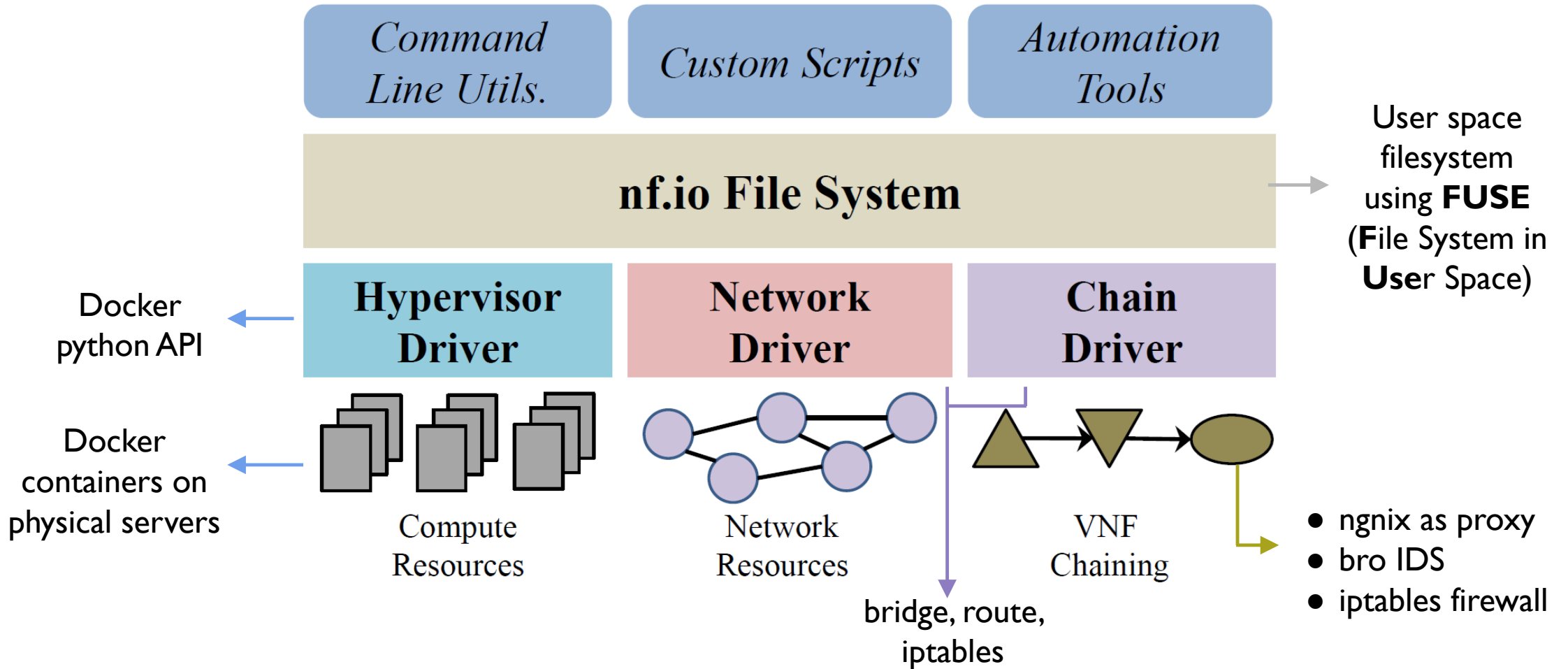
System Architecture



System Architecture



System Architecture



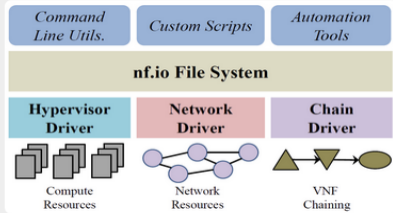
More about nf.io: <http://watnfv.github.io/nf.io/>

nf.io Get Started Downloads Documentation Developers Credits

Overview

nf.io provides a file system abstraction for Virtual Network Function (VNF) management and Orchestration. It exposes the Linux file system interface as the northbound API, enabling user to manage and orchestrate VNFs by performing simple file and directory manipulation operations.

Architecture



The diagram illustrates the nf.io architecture. At the top, three blue boxes represent user interaction methods: 'Command Line Utils.', 'Custom Scripts', and 'Automation Tools'. These interact with the 'nf.io File System', shown as a yellow bar. Below this, three colored boxes represent resource drivers: 'Hypervisor Driver' (blue), 'Network Driver' (red), and 'Chain Driver' (purple). The Hypervisor Driver manages 'Compute Resources' (represented by server icons), the Network Driver manages 'Network Resources' (represented by a network diagram), and the Chain Driver manages 'VNF Chaining' (represented by a flow diagram with triangles and circles).

A high-level view of the nf.io architecture is shown in the above figure. The nf.io File System is a virtual file system layered on top of the OS file system. VNF operations are triggered when a user writes special strings (e.g., 'activate', 'stop', etc.) in action files. nf.io performs these operations by using three resource drivers: (i) Hypervisor Driver, (ii) Network Driver, and (iii) Chain Driver. The hypervisor and network drivers manage the compute and network resources, respectively. The chain driver manages VNF chains by configuring traffic forwarding rules between VNFs.

Publications

- Md. Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, and Raouf Boutaba, **nf.io: A File System Abstraction for NFV Orchestration**, *IEEE NFV-SDN 2015*, San Francisco, USA, 18-21 November, 2015. [PDF](#) [PPT](#)
- Md. Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, and Raouf Boutaba, **nf.io: A File System Abstraction for NFV Orchestration**, *ACM SIGCOMM 2015*, London, UK, August 17-21, 2015: 361-362. [PDF](#)

Source Code

The source code is available at <https://github.com/WatNFV/nf.io>

QUESTION, YOU HAVE?



ANSWER, I SHALL.

memegenerator.net