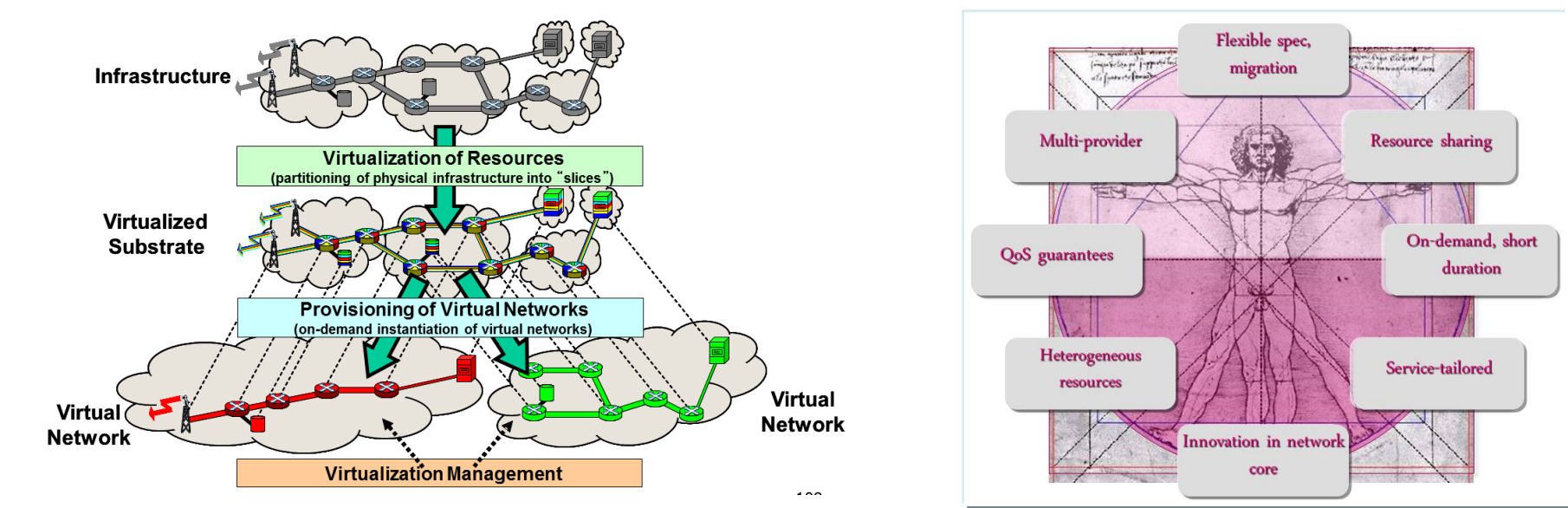# Networking Tomorrow: Virtualized, Software-Defined, Distributed
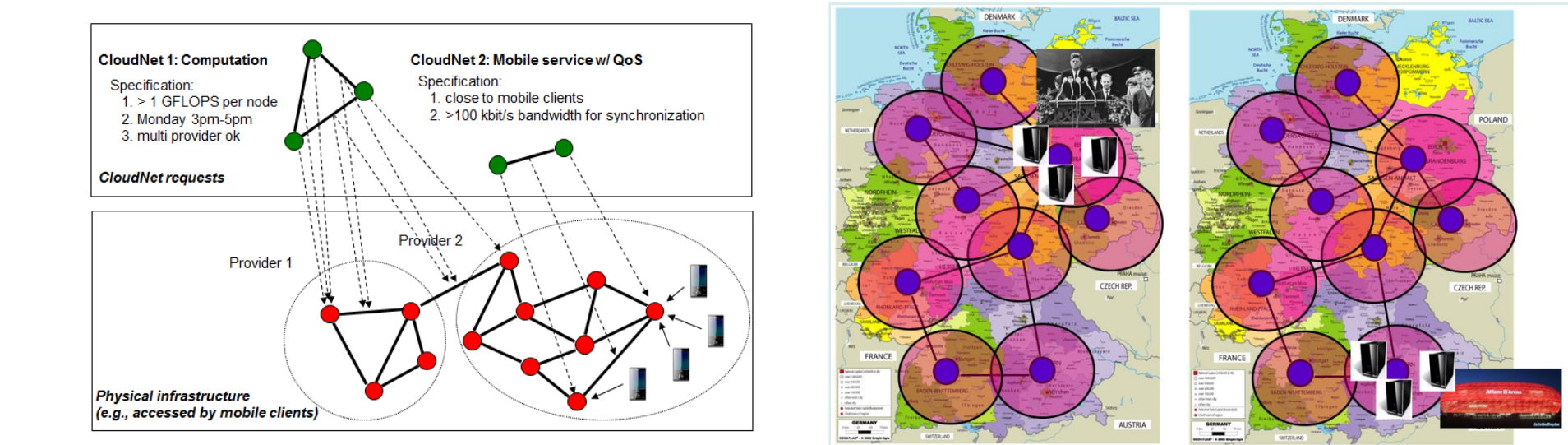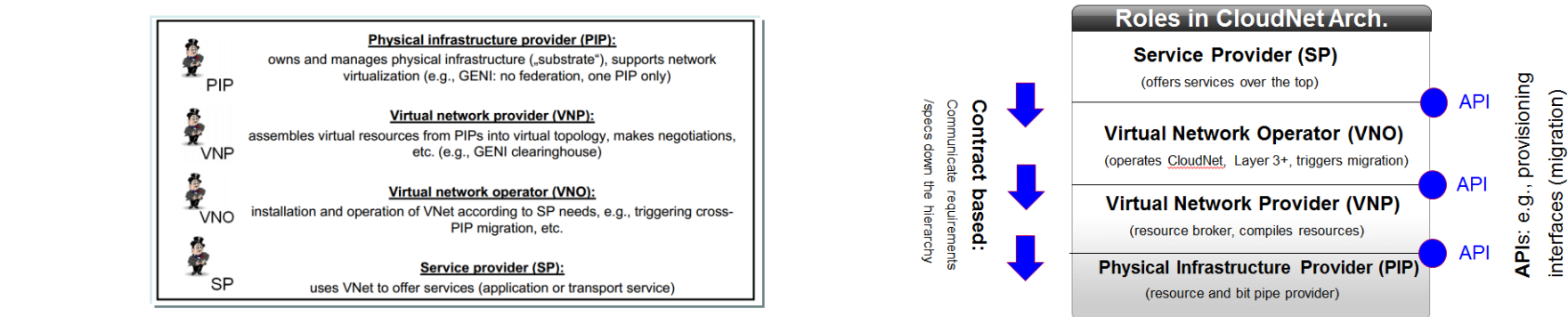
Stefan Schmid et al.

1. Virtualized!

Vision: CloudNets as flexibly specifiable, service-specific and adaptive virtual networks connecting heterogeneous resources.
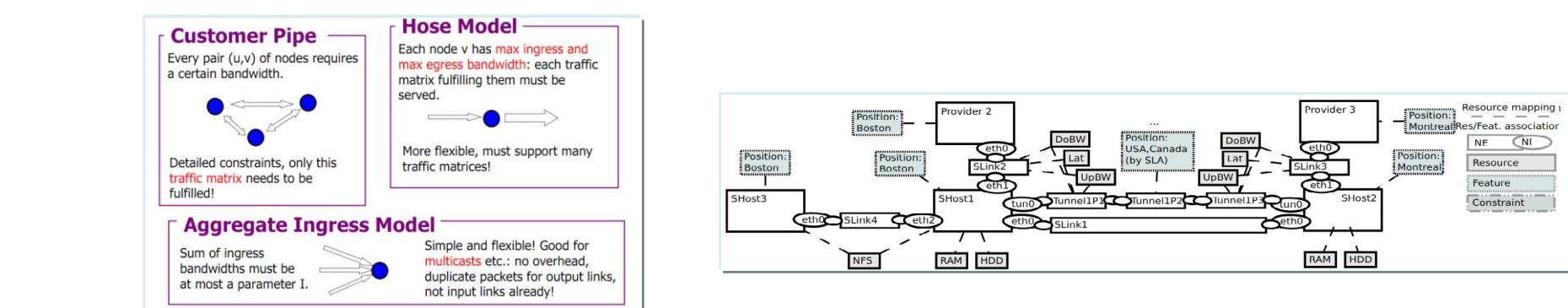

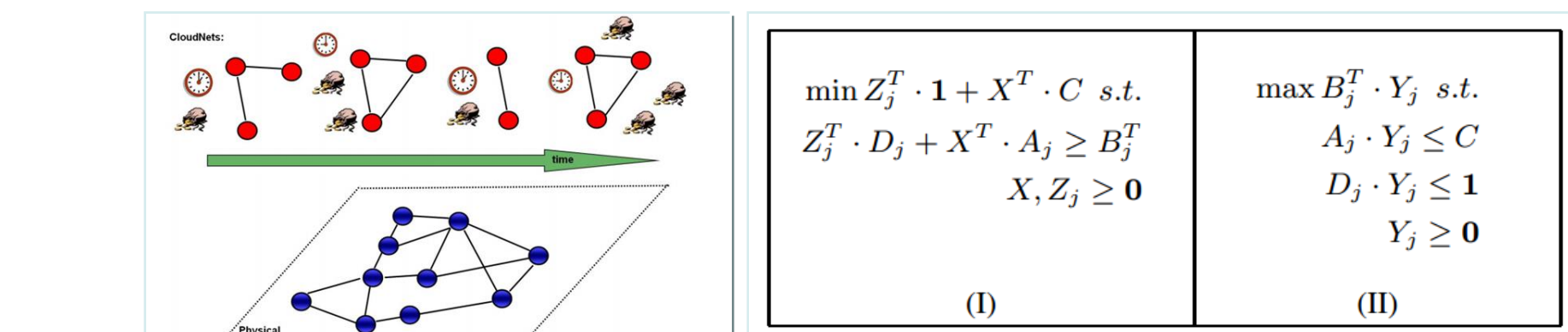Use cases: flexible resource allocation, migration, elastic computing

Challenges:

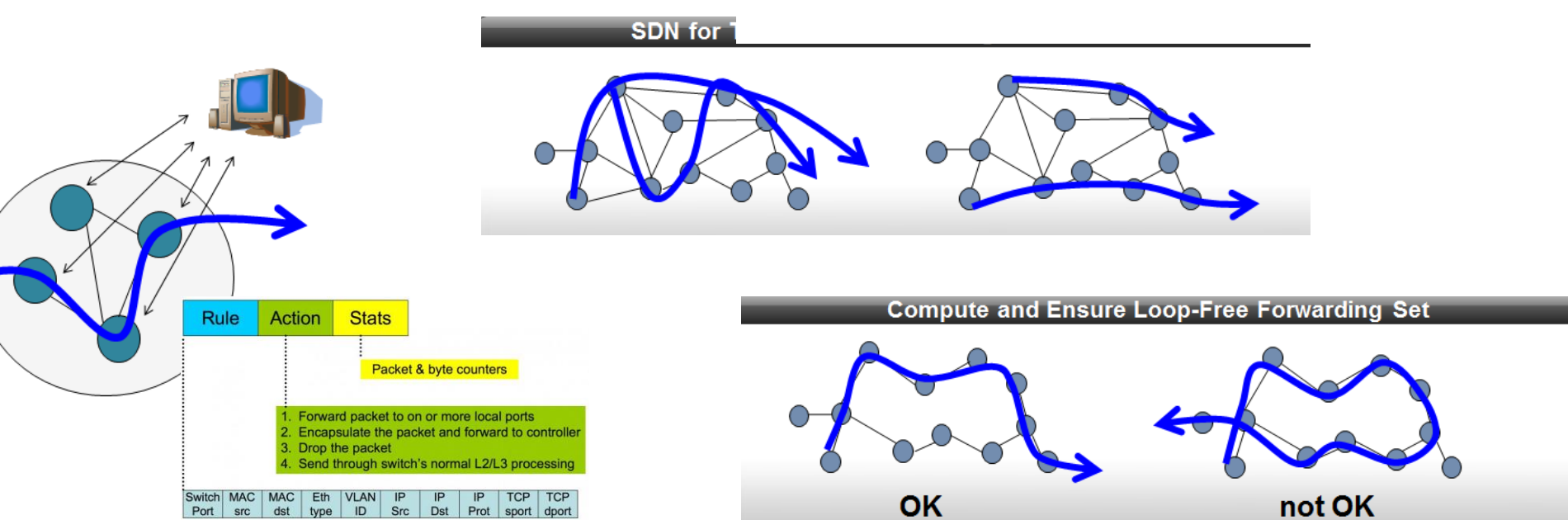Independent Economic Players

Specification

$$\min Z_j^T \cdot \mathbf{1} + X^T \cdot C \quad s.t.$$
$$Z_j^T \cdot D_j + X^T \cdot A_j \geq B_j^T$$
$$X, Z_j \geq \mathbf{0}$$
(I)

$$\max B_j^T \cdot Y_j \quad s.t.$$
$$A_j \cdot Y_j \leq C$$
$$D_j \cdot Y_j \leq \mathbf{1}$$
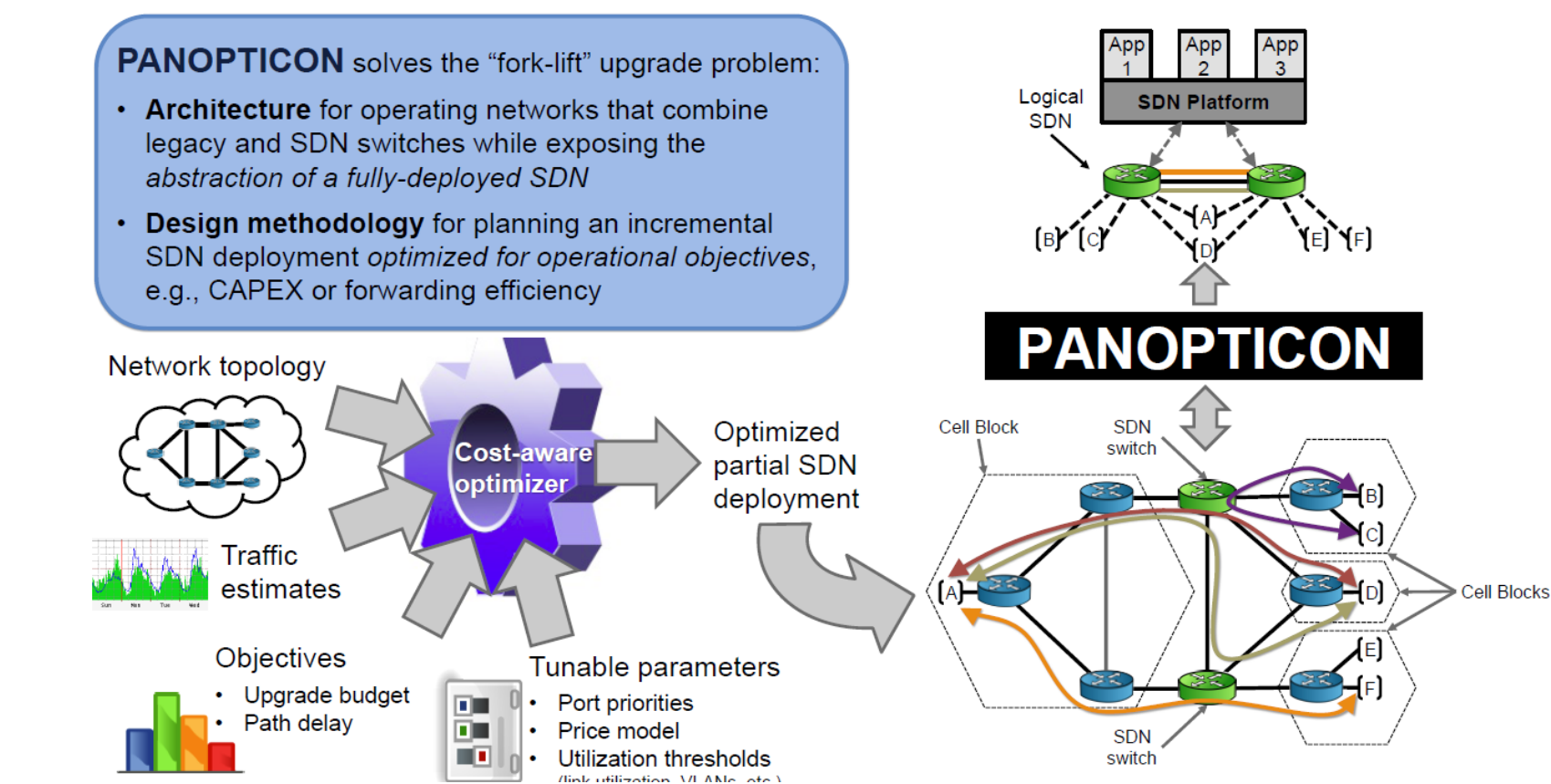$$Y_j \geq \mathbf{0}$$
(II)

Optimization

2. Software-Defined!

Software—defined networking: software controlled network management

Use Cases: traffic engineering, QoS, network management, …

Compute and Ensure Loop-Free Forwarding Set

OK   not OK

Challenge: how to get there? Which parts to upgrade first?

PANOPTICON solves the "fork-lift" upgrade problem:
- **Architecture** for operating networks that combine legacy and SDN switches while exposing the *abstraction of a fully-deployed SDN*
- **Design methodology** for planning an incremental SDN deployment *optimized for operational objectives*, e.g., CAPEX or forwarding efficiency

3. Distributed!

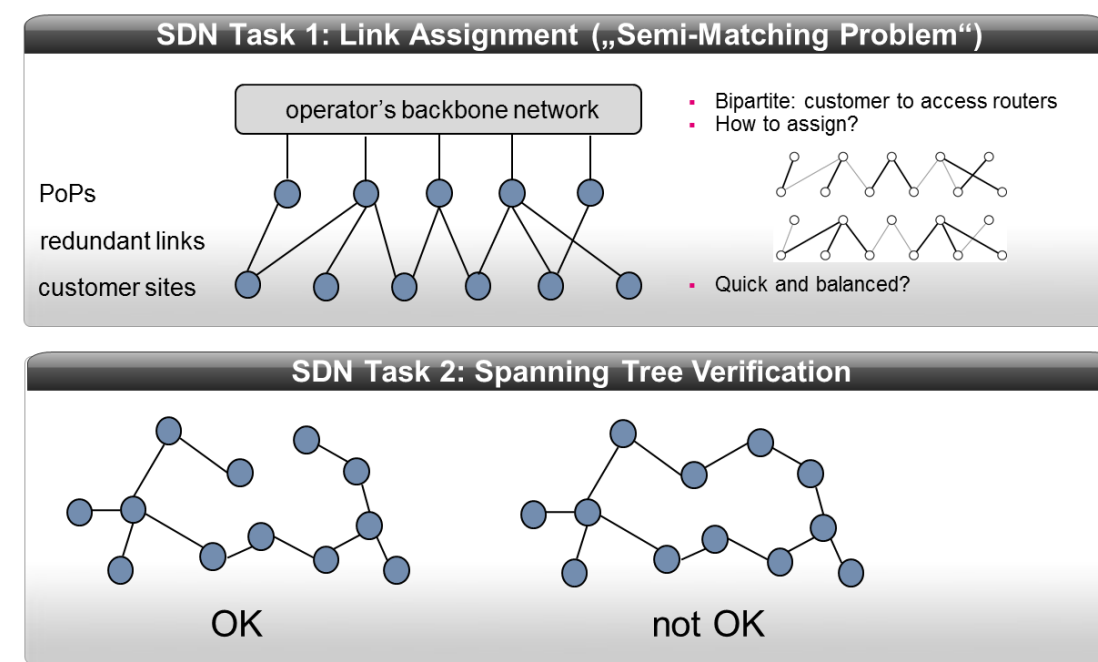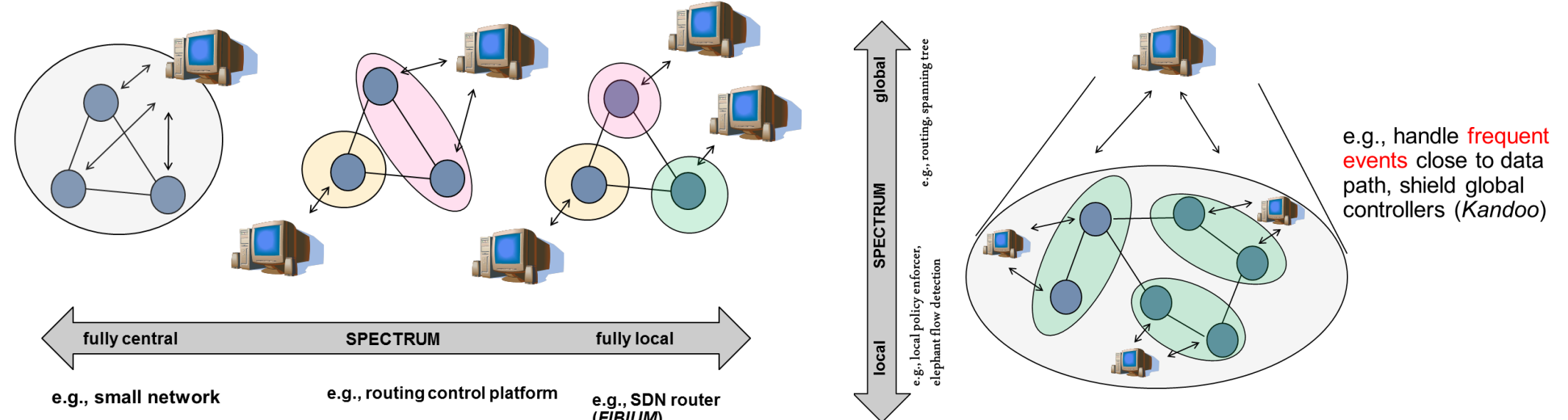Example: One SDN switch enough for SDN abstraction, but more needed for flexible control.

Insight #1:
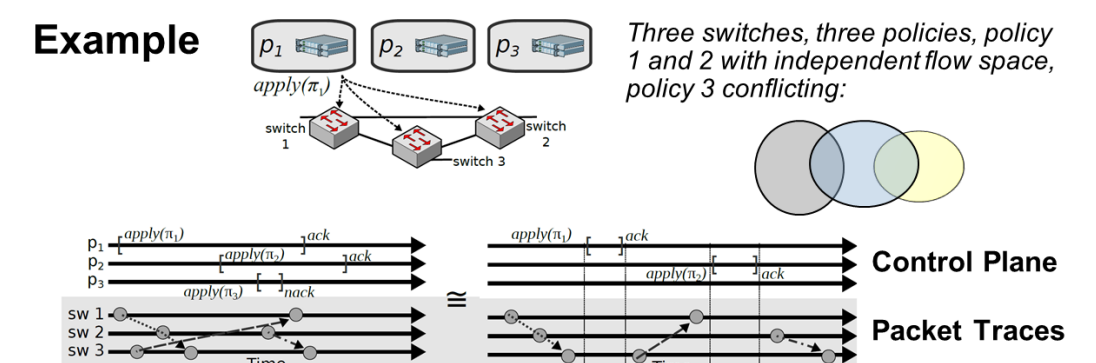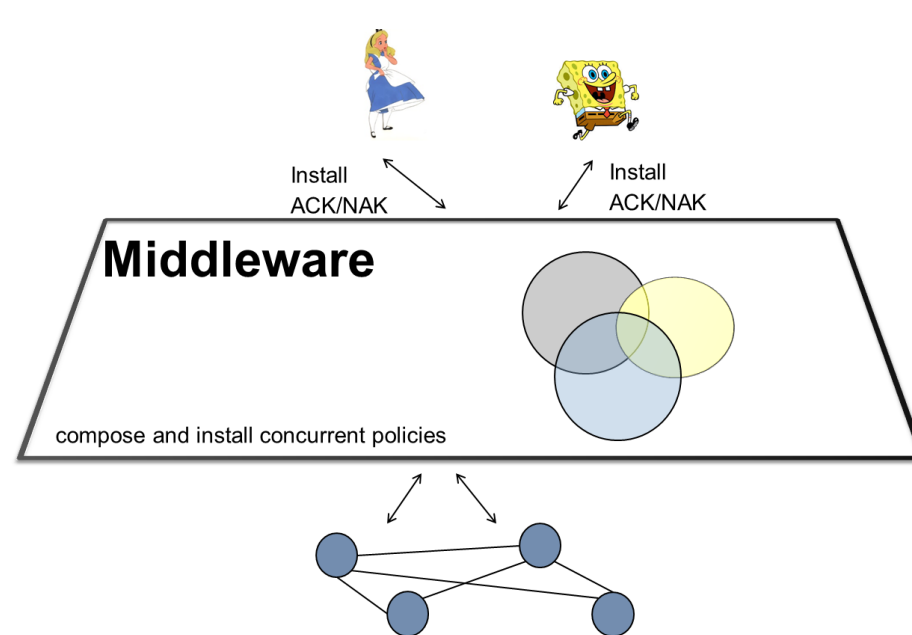≥ 1 SDN switch →
Policy enforcement

Access control
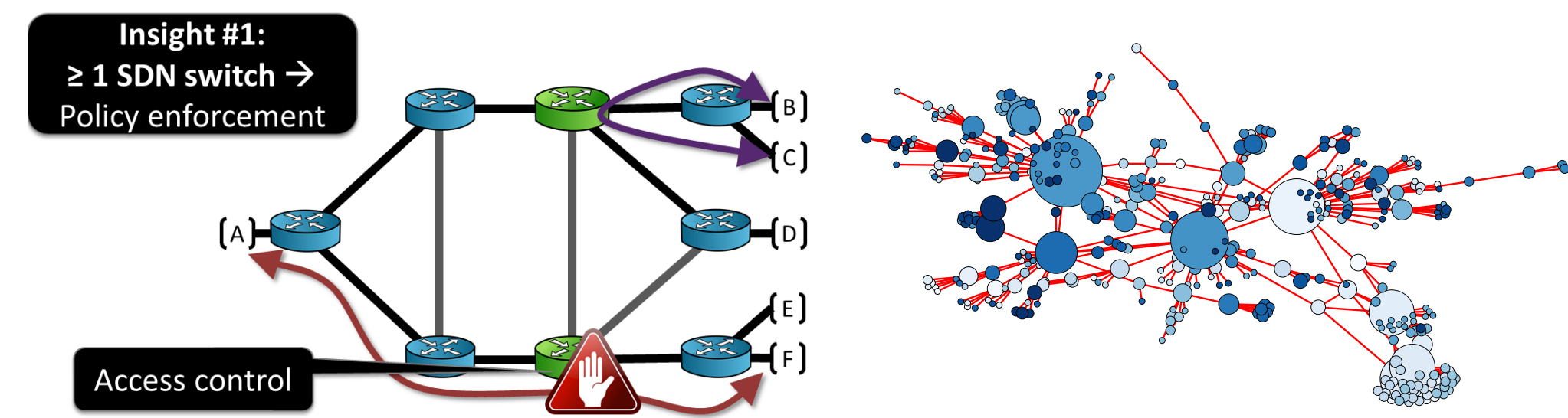
Distributed control of virtualized networks:

e.g., handle frequent events close to data path, shield global controllers (*Kandoo*)

fully central   SPECTRUM   fully local

e.g., small network   e.g., routing control platform   e.g., SDN router (*FIBIUM*)

Approximation and verification easier than computation!

SDN Task 1: Link Assignment ("Semi-Matching Problem")

- Bipartite: customer to access routers
- How to assign?
- Quick and balanced?

PoPs
redundant links
customer sites

$$f(\text{ }) = \text{No}$$

SDN Task 2: Spanning Tree Verification

OK   not OK

Neighbor with same distance alert!
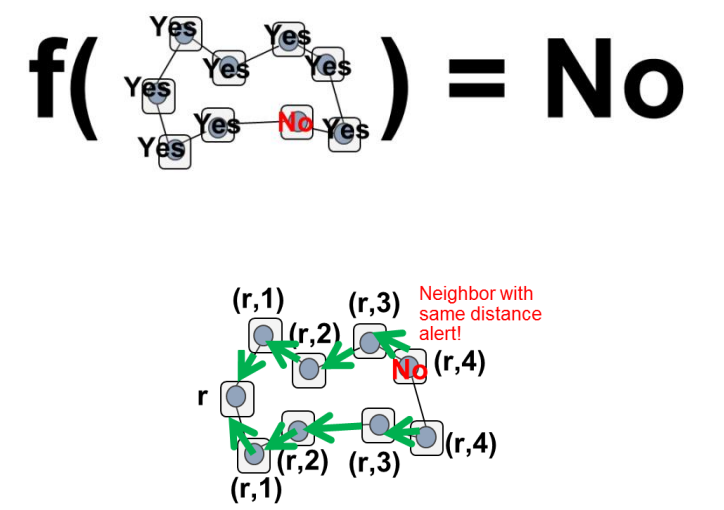
Concurrency and multi-authorship:

Install ACK/NAK   Install ACK/NAK

Middleware

compose and install concurrent policies

Example — Three switches, three policies, policy 1 and 2 with independent flow space, policy 3 conflicting:

Control Plane
Packet Traces

Left: Concurrent history: 3rd policy aborted due to conflict.
Right: In the sequential history, no two requests applied concurrently. No packet is in flight while an update is being installed.

No packet can distinguish the two histories! So as though the application of policy updates is atomic and packets cross the network instantaneously.

internal ports   internal ports
ingress port   ingress port

4. Demos: e.g., YouTube

For further information

Please contact **Stefan Schmid** (stefan.schmid@tu-berlin.de). This poster as well as more information on this and related projects can be obtained at website: http://www.net.t-labs.tu-berlin.de/~stefan/