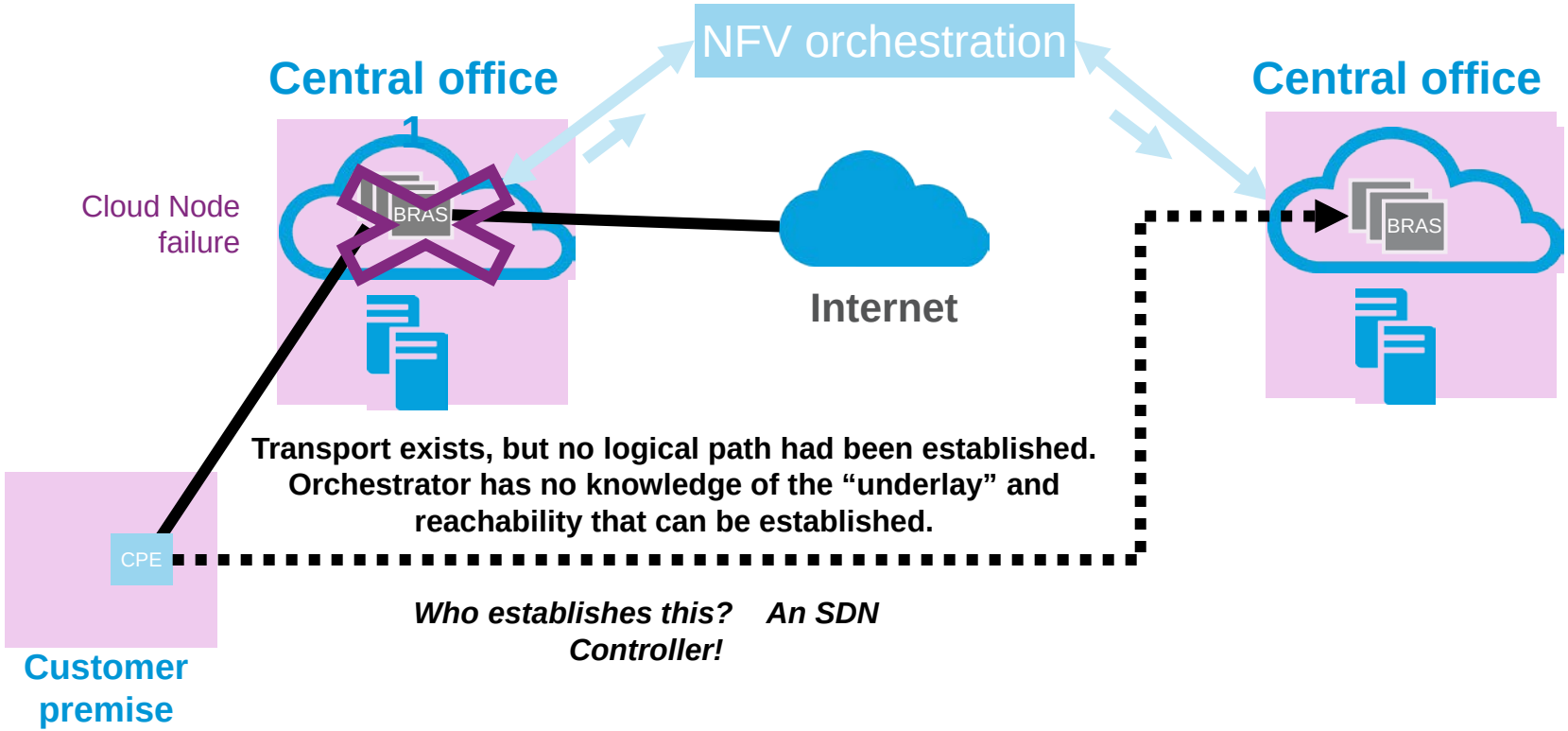
Controllers and Orchestrators

Cacophony or Harmony?

Prodip Sen
CTO, HP NFV
Chair, OPNFV Board of Directors
July 2015

An NFV scenario



NFV and SDN are complementary in a WAN

NFV

- Virtualization of network functions
- Eventual cloudification of the network infrastructure

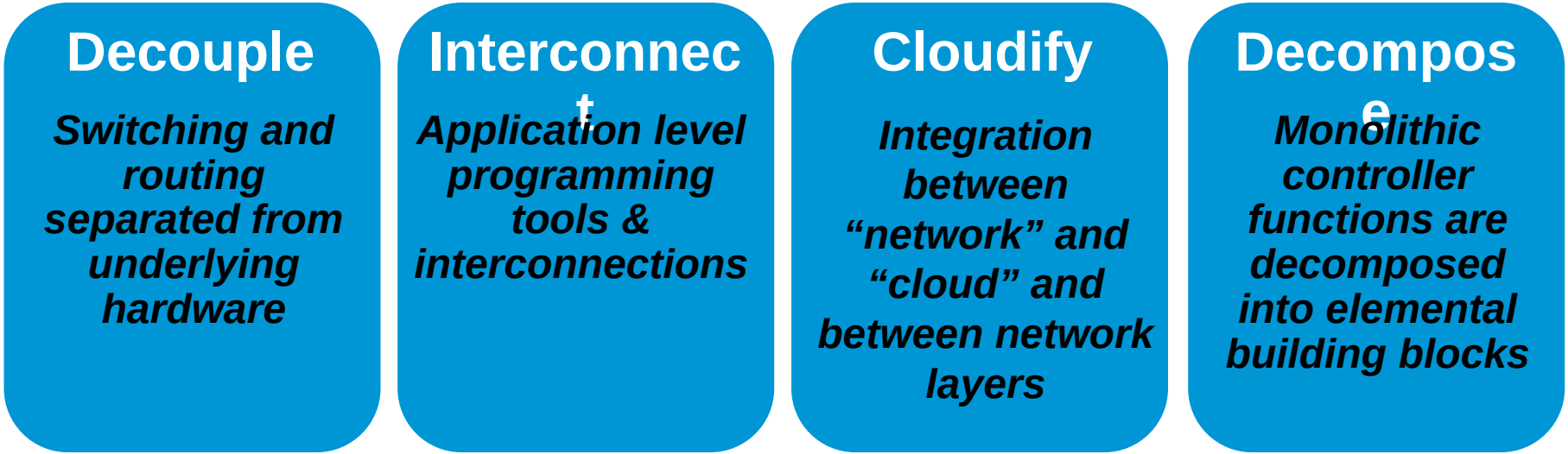
SDN

- Dynamic programmability of the network topology
- Service/application-aware traffic handling

- NFV is the use case for SDN – when you move or re-establish a function, you have to dynamically re-establish the topology
- Virtualization creates several new combinations of switching locations within/between data processing sites that need to be programmed in concert with assignment of compute & storage

We need a holistic view of SDN and NFV.

SDN technology is evolving in stages



- Each stage represents a sequence of technology developments
- Each stage will drive more innovation & faster delivery of applications and services
- Deployments can start/jump ahead to any stage as the technology matures
- Similar to NFV evolution stages discussed elsewhere

1 | Decouple

Switching and routing separated from hardware

Run switch and router control planes in software separated from the HW switches

Implement standardized protocols between planes, and flow-based traffic handling

Implement forwarding plane in software

Islands of programmable switching domains

Benefits

Application aware programmability of physical and virtualized switch

Simpler management of campus and data center networking

2 | Interconnect

Application level programming tools & interconnections

- Software-based function chaining, topology management, monitoring and visualization … available to service layer
- Interconnection of controller domains
- Data plane interconnection with underlay networks and limited control plane interconnectivity

Benefits

- Dynamic policy based network behavior
- Network programmability exposed to service layer

3 | Cloudify

Integration between "network" and "cloud" and between network layers

- Service friendly network programming constructs
- Policy-based (intent) interfaces
- Summarized SLA-based state visibility between network and cloud layers
- Control and data plane integration between network layers

Benefits

- Automation of policy based network behavior
- Integration of network programmability into cloud-based resource orchestration

4 | Decompose

Monolithic controller functions are decomposed into elemental building blocks that are available to applications

- Federated control and orchestration structure
- PaaS APIs
- Composable service development environment at all layers
- Control and data plane integration between network layers

Benefits

Increased capacity to rapidly deliver new, innovative services.
Ability to point/click to compose services that provide a seamless integration of network, compute and storage resources dynamically
Efficient sharing of underlying resources.

The 4 stages of SDN: NFV Applicability

Decouple

Campus and data center networking

Interconnect

Gi-LAN, vCPE

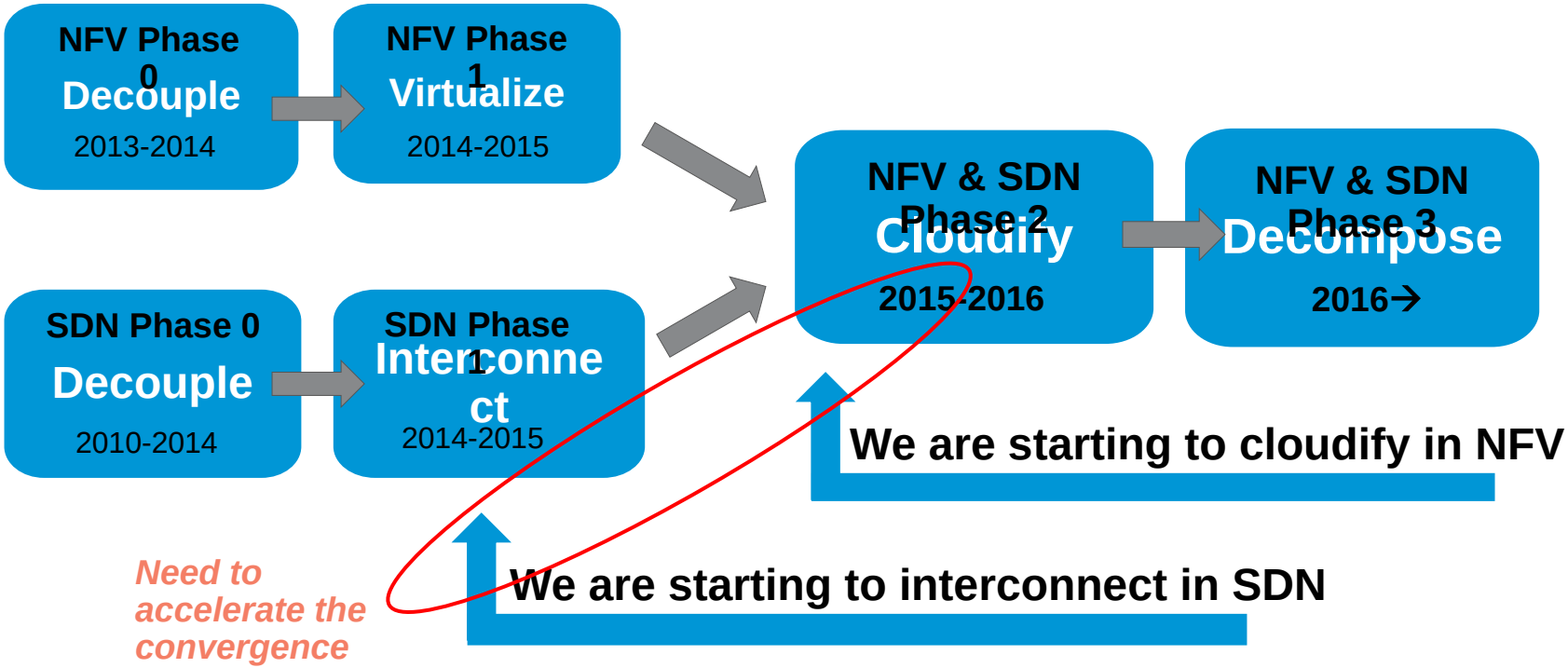
Cloudify

vEPC, MVNOs

Decompose

vRAN

The convergence of SDN and NFV stages



What kind of networking do we need for NFV?

Standard mechanism	Static overlay networks using tunnels
SDN controllers for overlay/service chaining	Dynamic establishment of overlay connectivity
Apps	Programmability of the overlay
State awareness	Connectivity between underlay and overlay
Operational design for SDN/NFV	Controller / orchestrator hierarchy & federation

Cacophony or Harmony ?

- Between controllers and orchestrators (across hierarchies and domains):
  - Be independent but cooperative
  - Request / inform but don't dictate
  - Communicate what is necessary, don't drown in information
- Focus on Intent, Policy, SLA based abstractions
- Keep core models and intelligence unified – not fragmented into "plug-ins"