

The background of the slide is a blue-toned image of a perspective view into a tunnel. The walls of the tunnel are covered in a pattern of binary code (0s and 1s). A yellow line runs along the bottom of the tunnel, leading towards a bright light at the end.

ACHIEVING AVAILABILITY AND RESILIENCY IN OPENSTACK FOR NFV

2015 Annual IEEE CQR International Workshop, May 13, 2015

Ali Kafel

Senior Director, Business Development

Ali.Kafel@Stratus.com

Defining Availability, Reliability and Resiliency

■ Availability

- % of time an equipment is in an operable state ie. access information or resources
- $\text{Availability} = \text{Uptime} / \text{Total time}$

■ Reliability

- How long a system performs its intended function.
- $\text{MTBF} = \text{total time in service} / \text{number of failures}$

■ Resiliency

- The ability to recover quickly from failures, **to return to its original form, state, etc. (just before the failure)**

- **Therefore, a Highly Available (HA) system may not be Highly Reliable (HRel) or Highly Resilient (HRes)**
- **A Fault Tolerant (FT) system is = Highly Available, Highly Reliable and Highly Resilient (state is preserved)**

Three ways to provide Stateful FT in VNFs

In the Hardware

In the Applications

In the Software Infrastructure

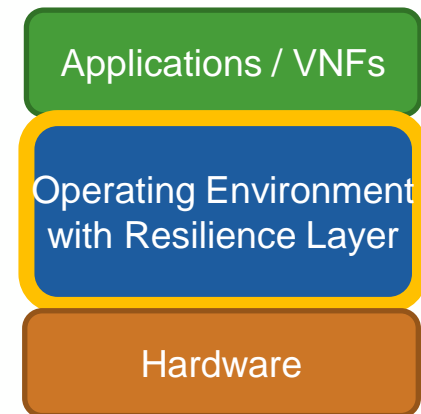
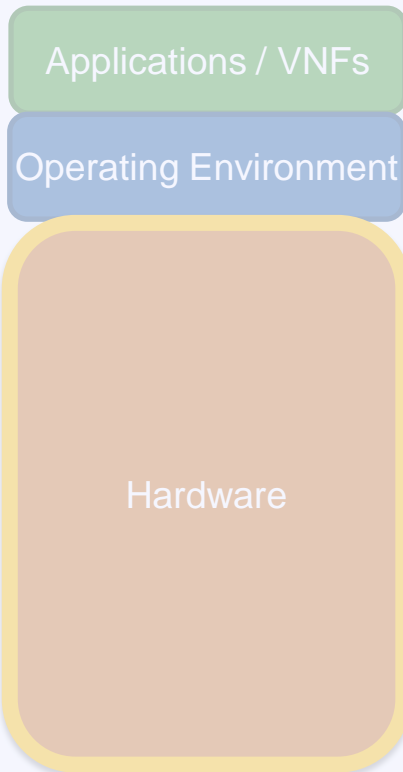
Pros

- Transparent – no code change
- Fast & Simple Deployment
- No special App Software

- App specific state can be Customized

- Transparent – no code change
- Fast & Simple Deployment
- No special App Software – deploy any
- No Special Hardware – use commodity
- Multiple Levels of Resiliency Supported
- Higher efficiency of resiliency – N+k

Costs & Resources ↑



Cons

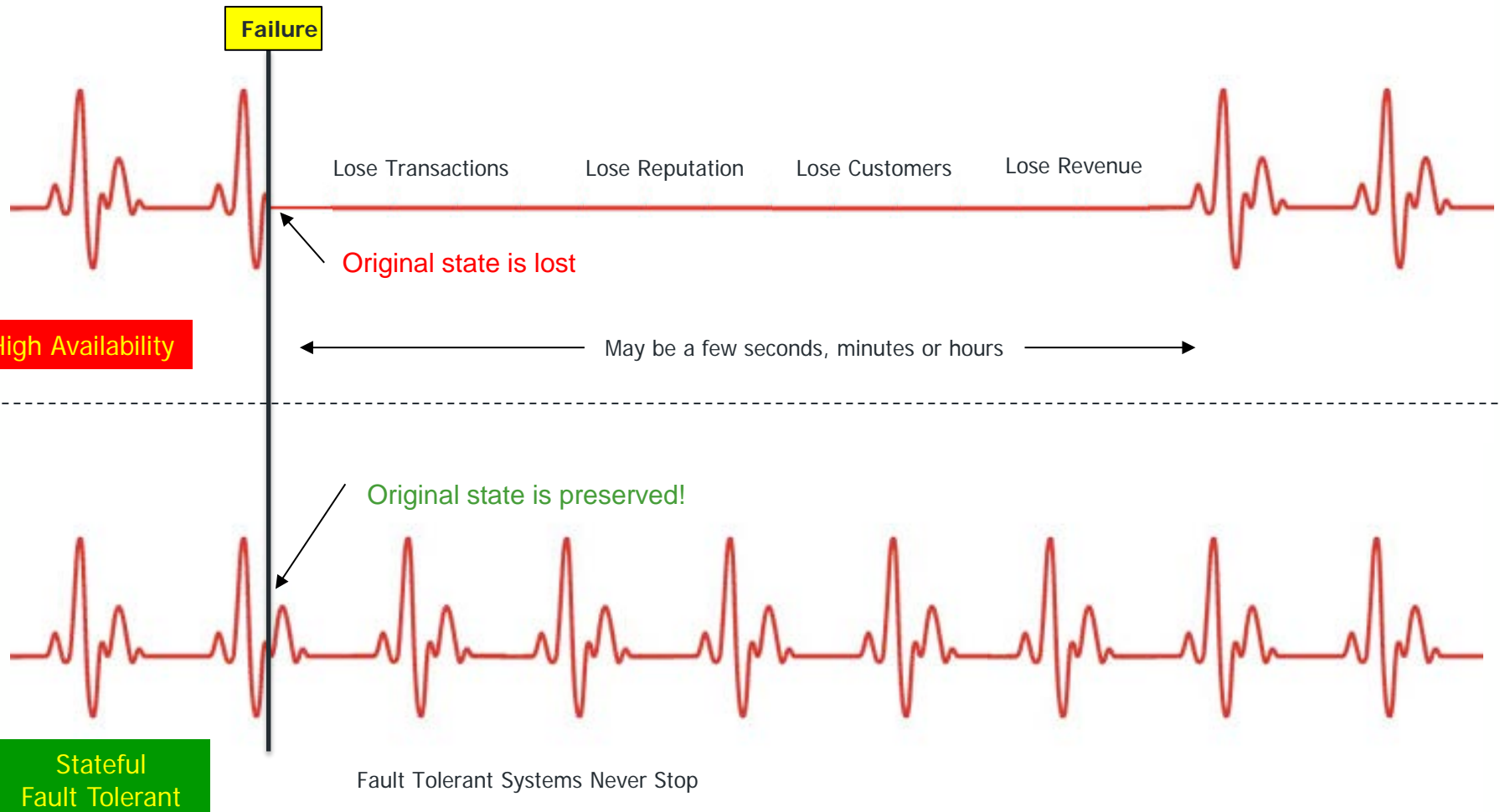
- Very expensive
- Inefficient utilization
- Special Hardware
- Rigid

- Every App must be modified
- Longer time to deploy
- Complex
- Rigid

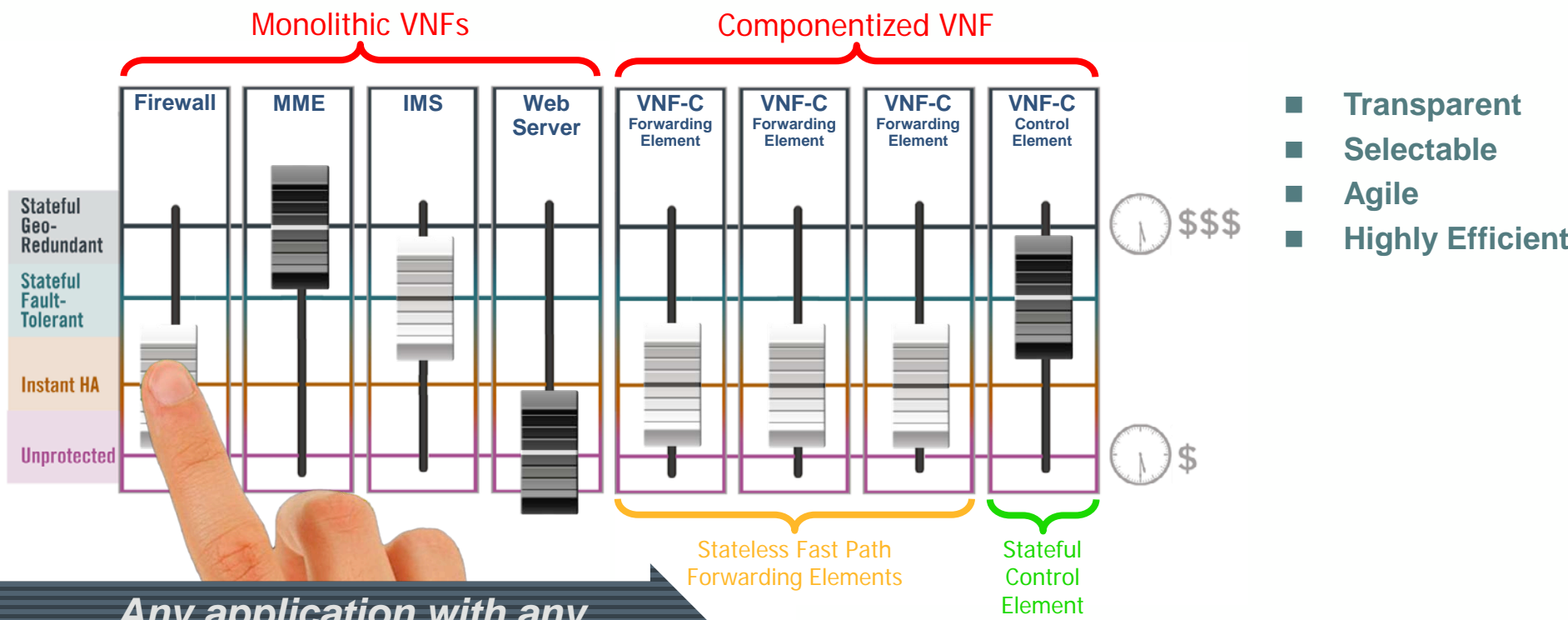
- Higher efficiency may not be possible on very large monolithic Apps

Stateful Fault Tolerance = HA + HRel + HRes

When Seconds Count... Loss of Revenue, Reputation, Safety, Life



The Right way: A Software Defined Availability (SDA) approach that is Transparent, Selectable, Agile and High Efficient



Any application with any availability need with application transparency

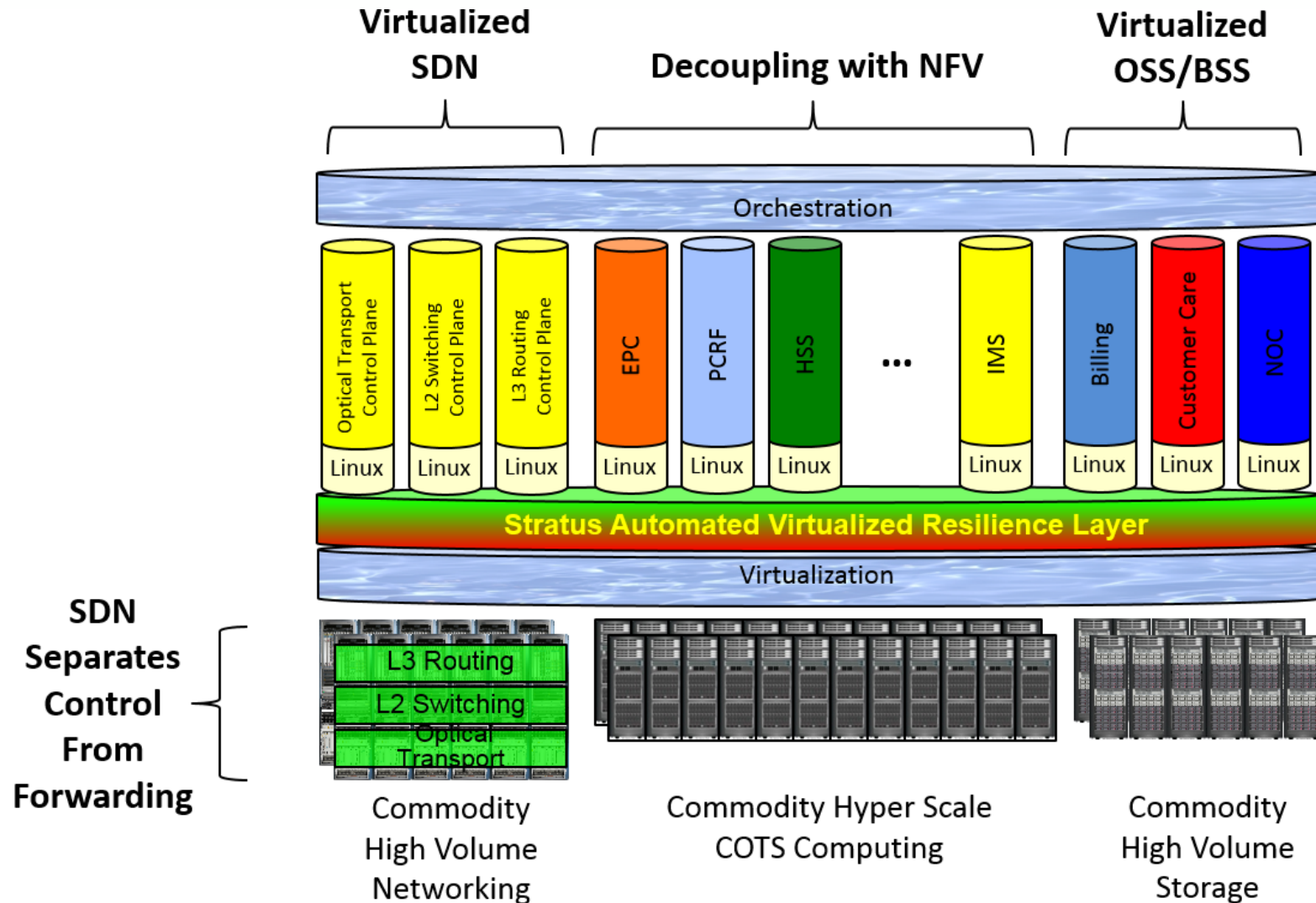
The Right Availability for Each Component

- Stateful Control Elements can be Fault Tolerant or even Geo-Redundant
- Stateless Fast Path Elements may be unprotected or Instant HA

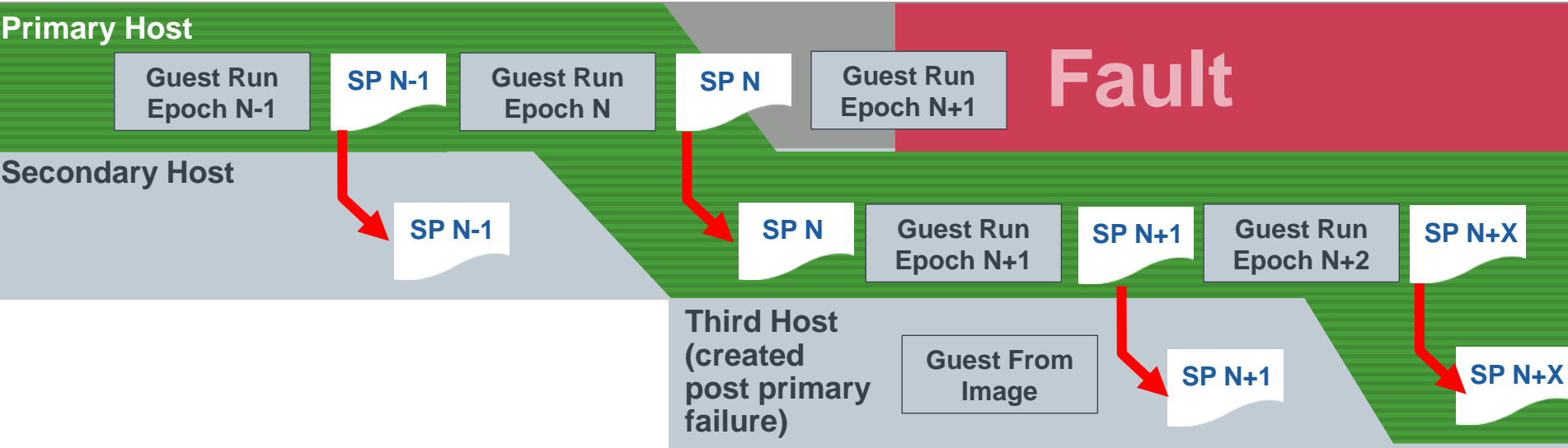
Highly Efficient

- Maximum Resilience where it matters
- Maximum Performance for Fast Path Forwarding

Am Automated Virtualized Resilience Layer for NFV and SDN

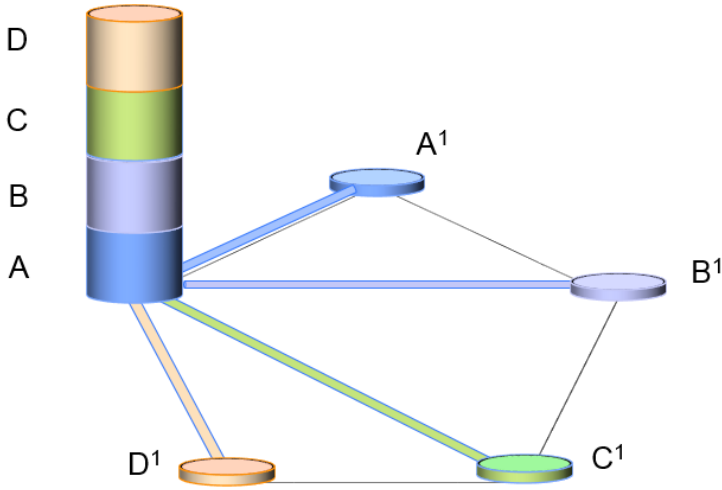


Protected VM Operation - Statepointing



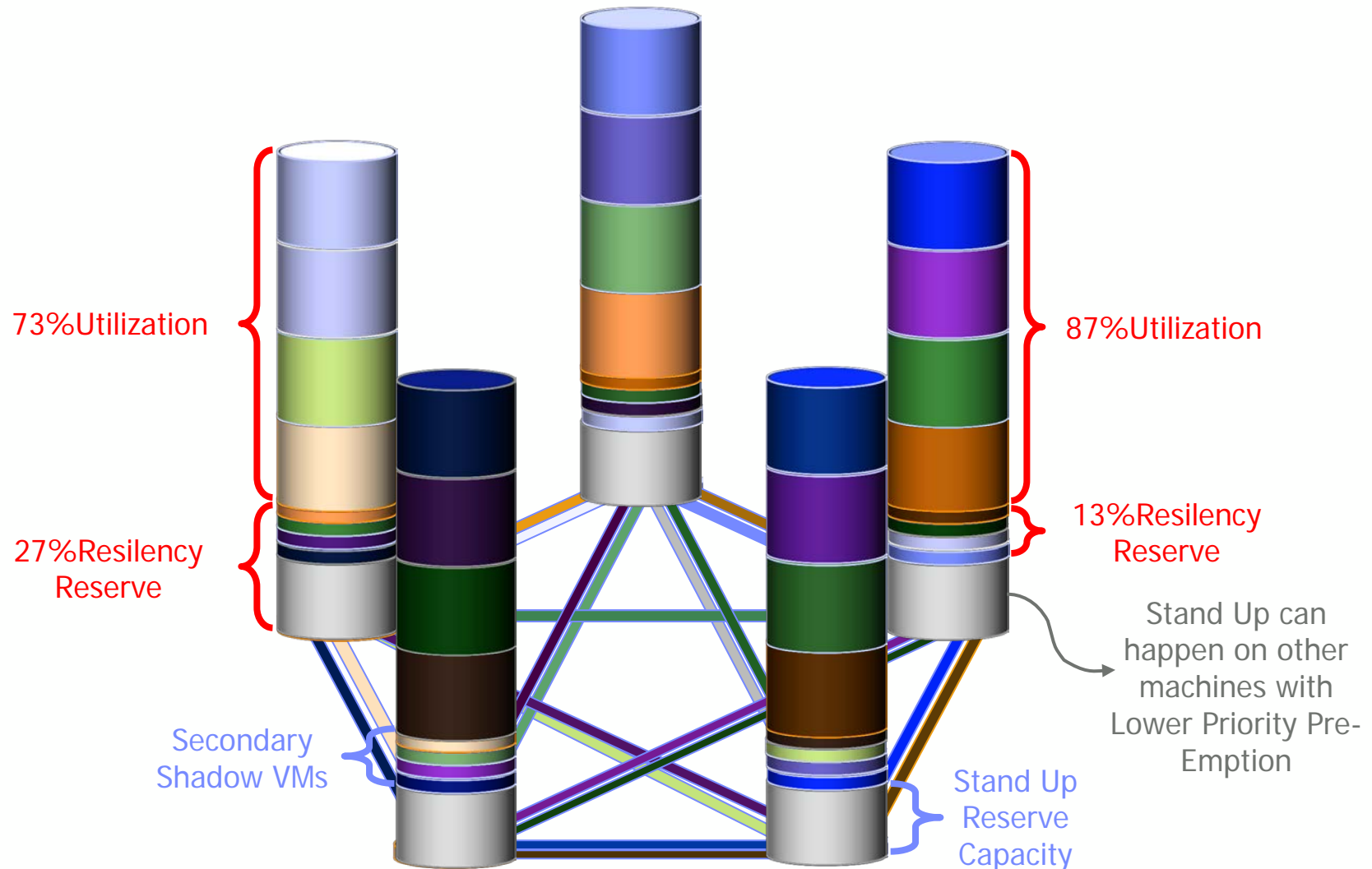
N+k De-Clustered Redundancy

*Each Server's Apps (ABCD) Are Backed Up On
Separate Servers
Which could be anywhere in the Pool of Servers*



Primaries are “Live Migrated” to New Server to Balance the Load

Yielding High Utilization, and Low Additional Reserve Capacity



The SDA approach provides a highly resilient NFVI that accelerates the “Cloudification” of the Telco Network

■ Transparency

- Software Defined, transparent Service Continuity, performed automatically by the infrastructure, without Application code changes

■ Selectability

- Fully automated selectable levels of resilience to Cloud workloads, including High Availability and Fault Tolerance, with Geo-Redundancy

■ Agility and Simplicity

- Deploy existing or new non-redundant VNFs as redundant fault-tolerant, with No complex code development, testing and support – for optimal partner ecosystem

■ Efficiency of Redundancy

- Unlike traditional approaches for Fault Tolerance, which limit Utilization to sub-50%, get dramatic increase in Efficiency of Redundancy, at 80% to 90%