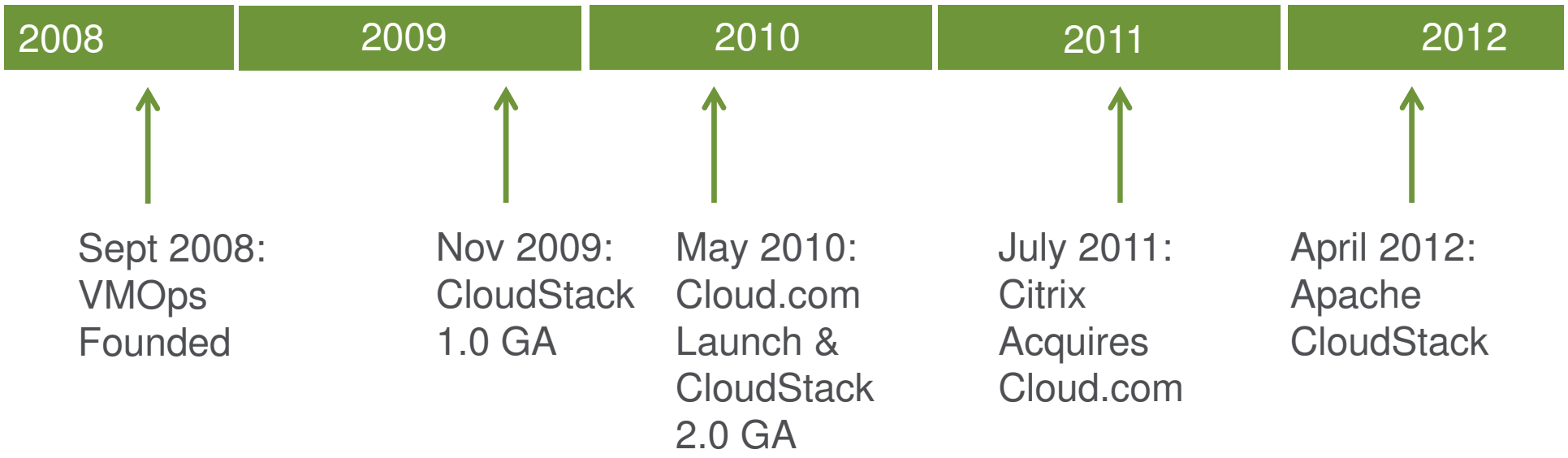
Architecting for the cloud: lessons learned from 100 CloudStack deployments

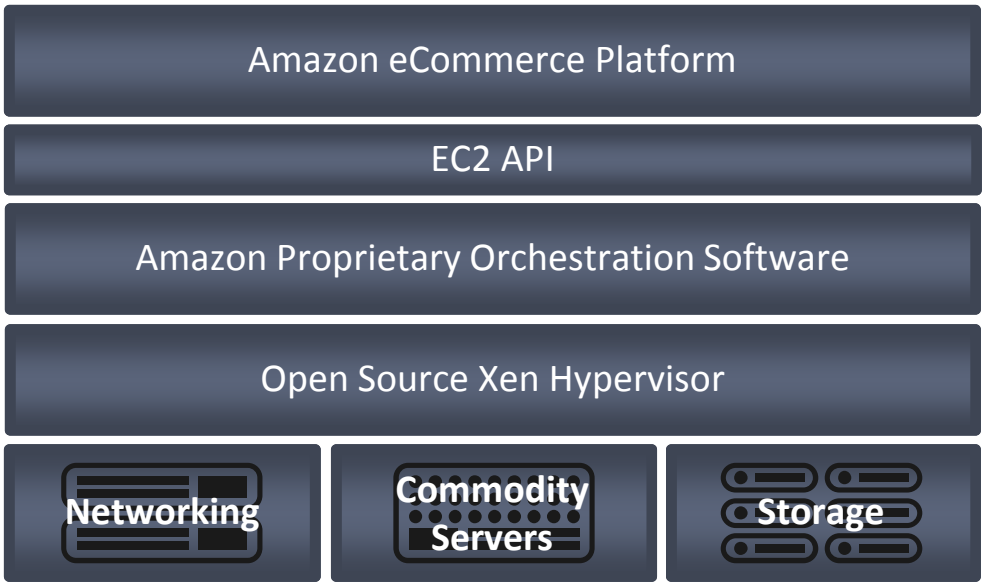
Sheng Liang

CTO, Cloud Platforms, Citrix

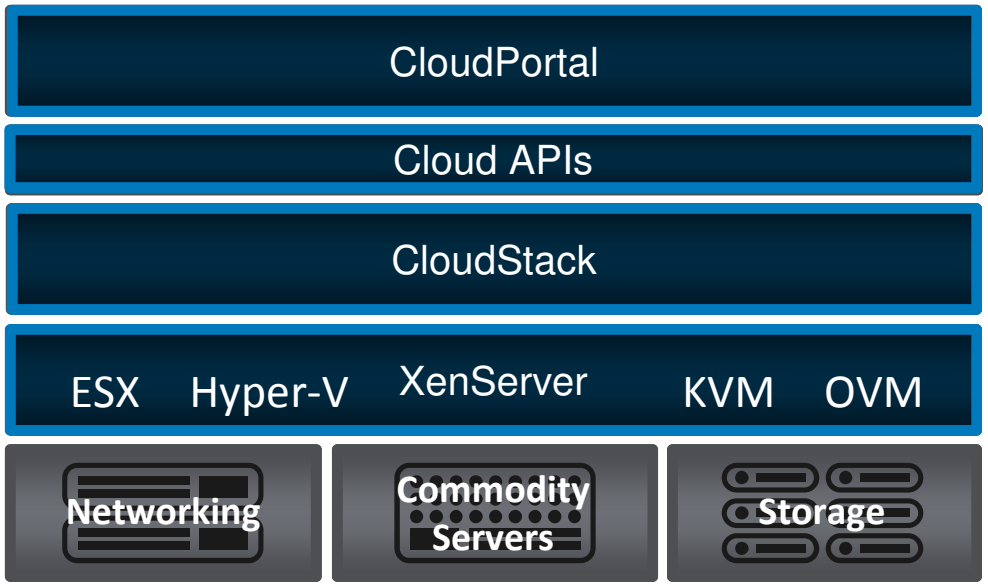
CloudStack History



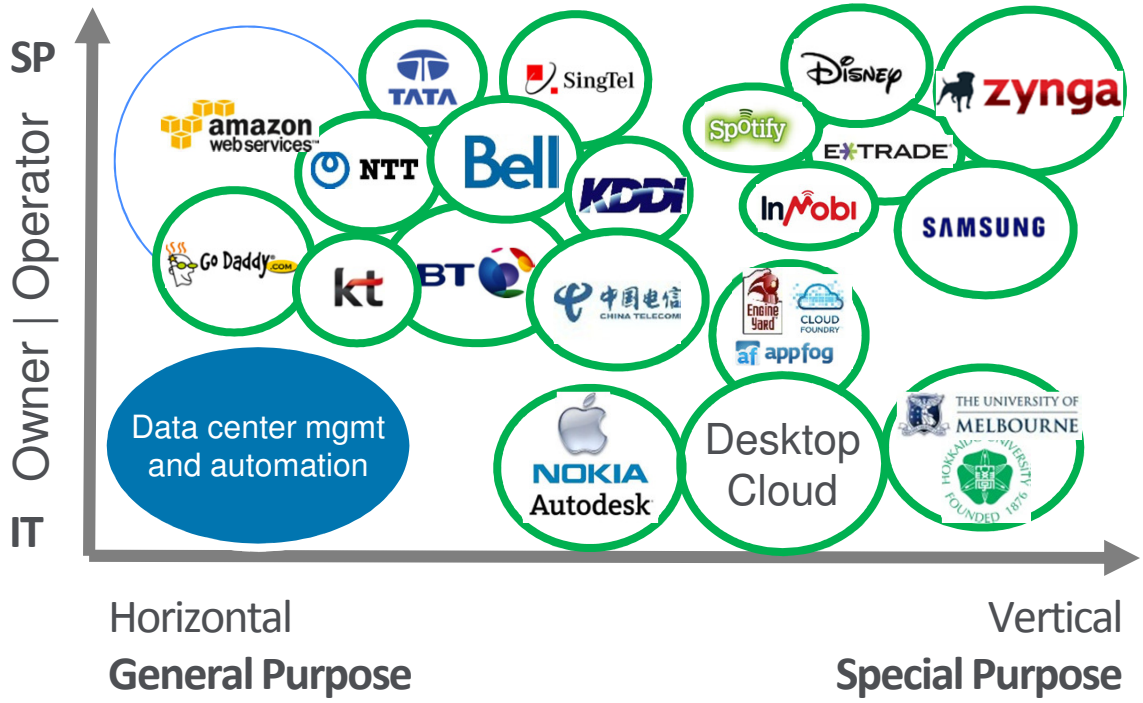
The inventor of IaaS cloud – Amazon EC2



CloudStack is inspired by Amazon EC2



There will be 1000s of clouds



Learning from 100s of CloudStack deployments

What is the biggest difference between traditional-style data center automation and Amazon-style cloud?

NIST
National Institute of
Standards and Technology
U.S. Department of Commerce

Special Publication 800-145

The NIST Definition of Cloud Computing

Recommendations of the National Institute of Standards and Technology

Peter Mell
Timothy Grance

2. The NIST Definition of Cloud Computing

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

Essential Characteristics:

On-demand self-service. A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

Broad network access. Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Resource pooling. The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

Rapid elasticity. Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Measured service. Cloud systems automatically control and optimize resource use by leveraging a metering capability[1] at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

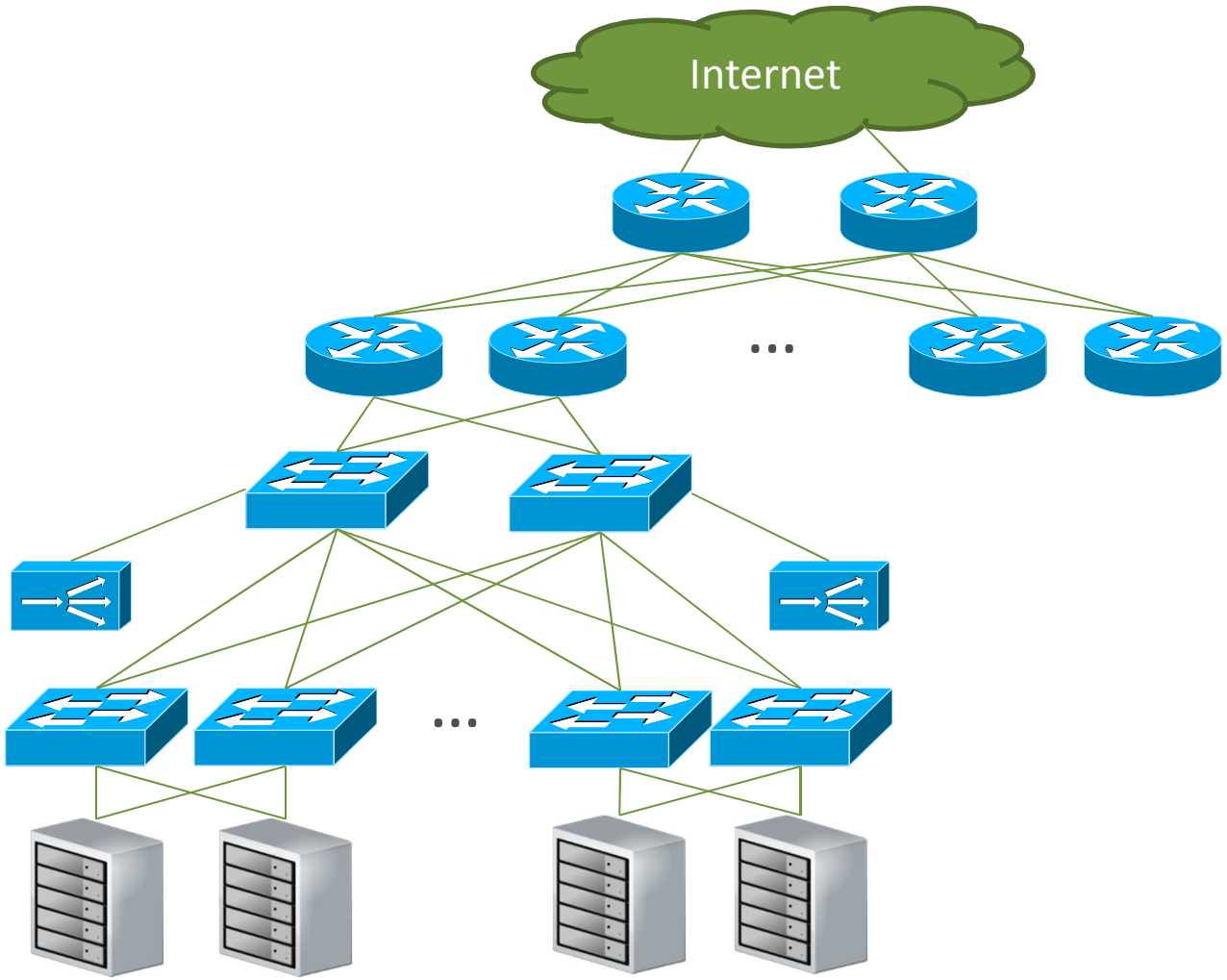
How to handle failures

8%

Annual Failure Rate of servers

Kashi Venkatesh Vishwanath and Nachiappan Nagappan, **Characterizing Cloud Computing Hardware Reliability,** *SoCC'10*

- Server failure comes from:
  - 70% - hard disk
  - 6% - RAID controller
  - 5% - memory
  - 18% - other factors
- Application can still fail for other reasons:
  - Network failure
  - Software bugs
  - Human admin error



Core Routers

Access Routers

Aggregation Switches

Load Balancers

Top of Rack Switches

Servers

40%

Effectiveness of network redundancy in reducing failures

Phillipa Gill, Navendu Jain & Nachiappan Nagappan, **Understanding Network Failures in Data Centers: Measurement, Analysis and Implications**, SIGCOMM 2011

- Bugs in failover mechanism
- Incorrect configuration
- Protocol issues such as TCP back-off, timeouts, and spanning tree reconfiguration

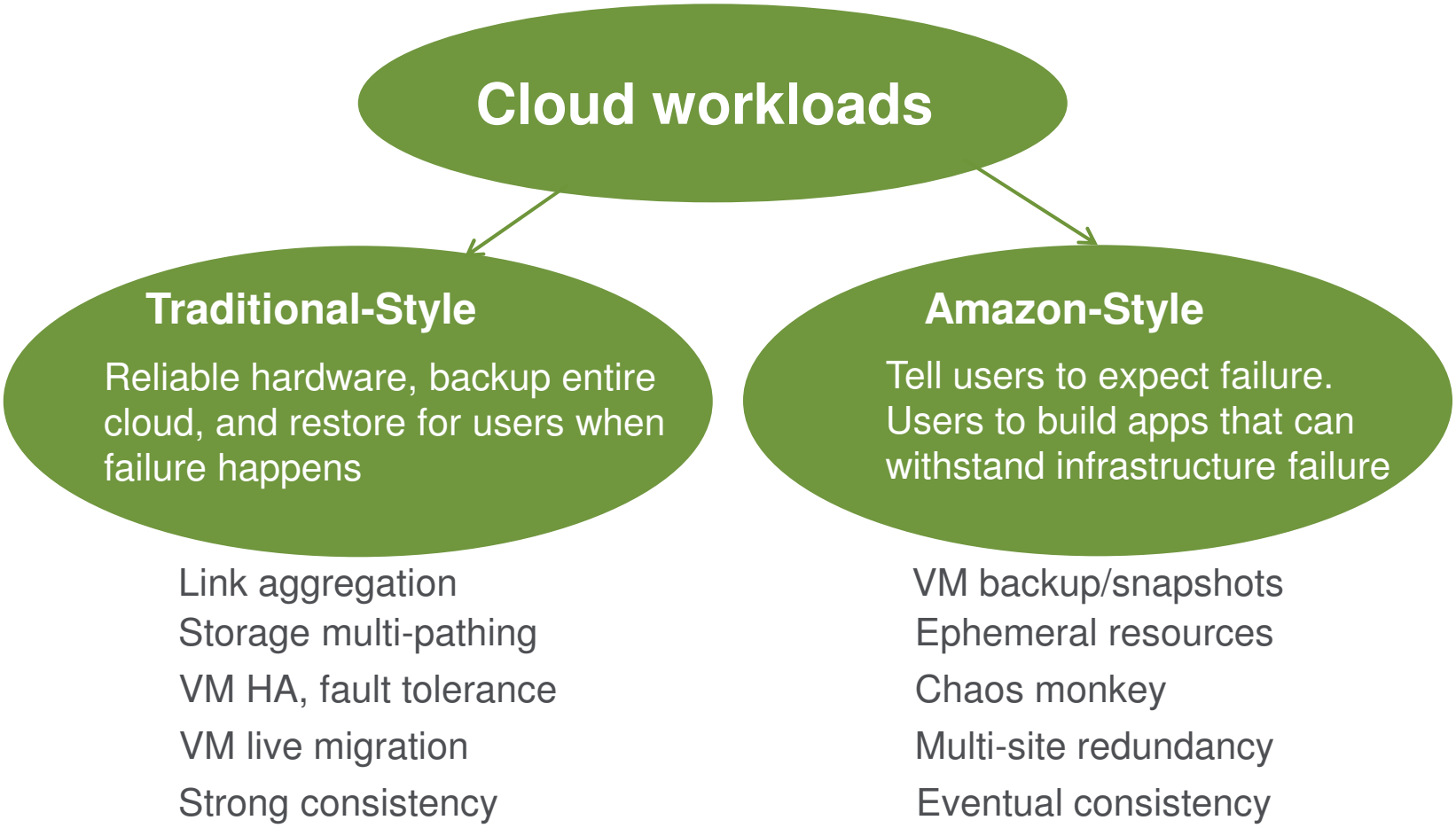
A. Promise users VM, storage, and networking will never fail -- no strategy to handle failures

B. Backup VM for users and restore for users when failure happens

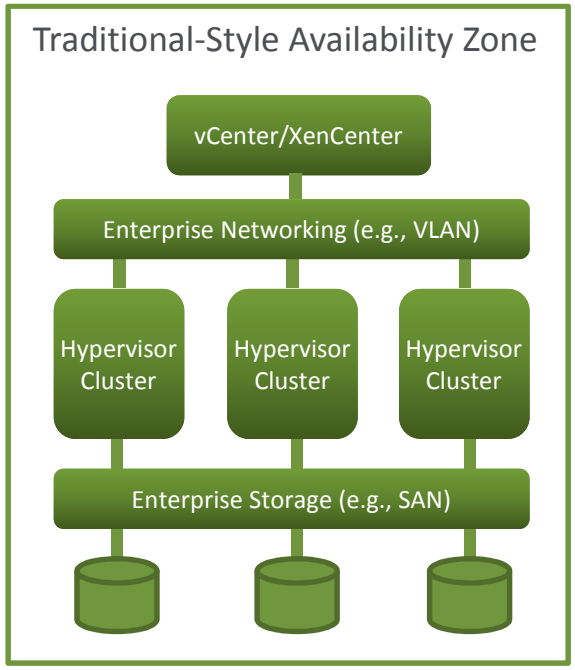
C. Tell users to expect failure. Users to backup VM and handle failure themselves


FARMVILLE
zCloud
West
Zone
AWS
West
Zone
zCloud
East
Zone
AWS
East
Zone

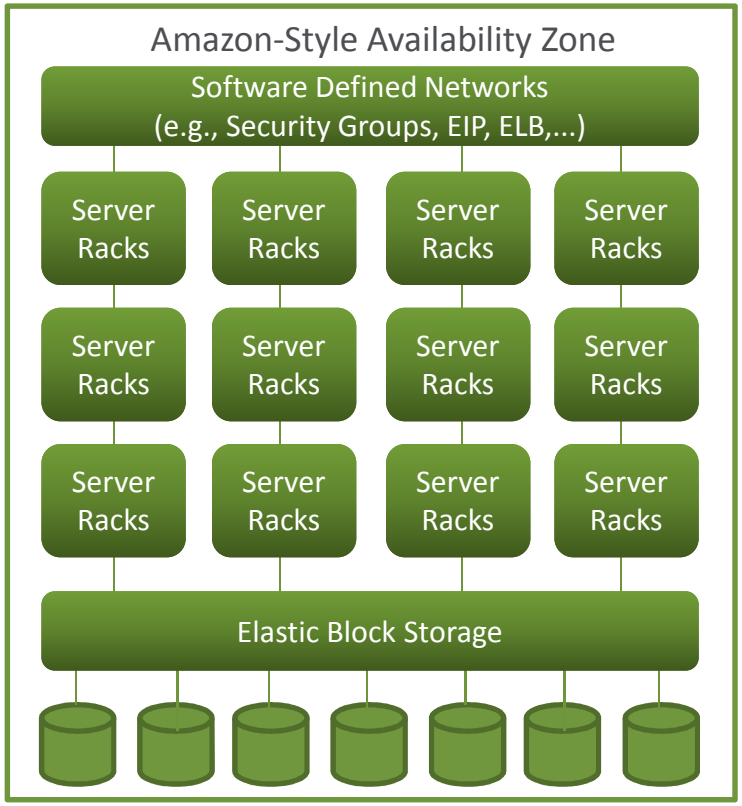

FARMVILLE
zCloud
West
Zone
AWS
West
Zone
Design for
Failure
zCloud
East
Zone


Cloud workloads
Traditional-Style
Reliable hardware, backup entire cloud, and restore for users when failure happens
Amazon-Style
Tell users to expect failure. Users to build apps that can withstand infrastructure failure
Link aggregation
Storage multi-pathing
VM HA, fault tolerance
VM live migration
Strong consistency
VM backup/snapshots
Ephemeral resources
Chaos monkey
Multi-site redundancy
Eventual consistency

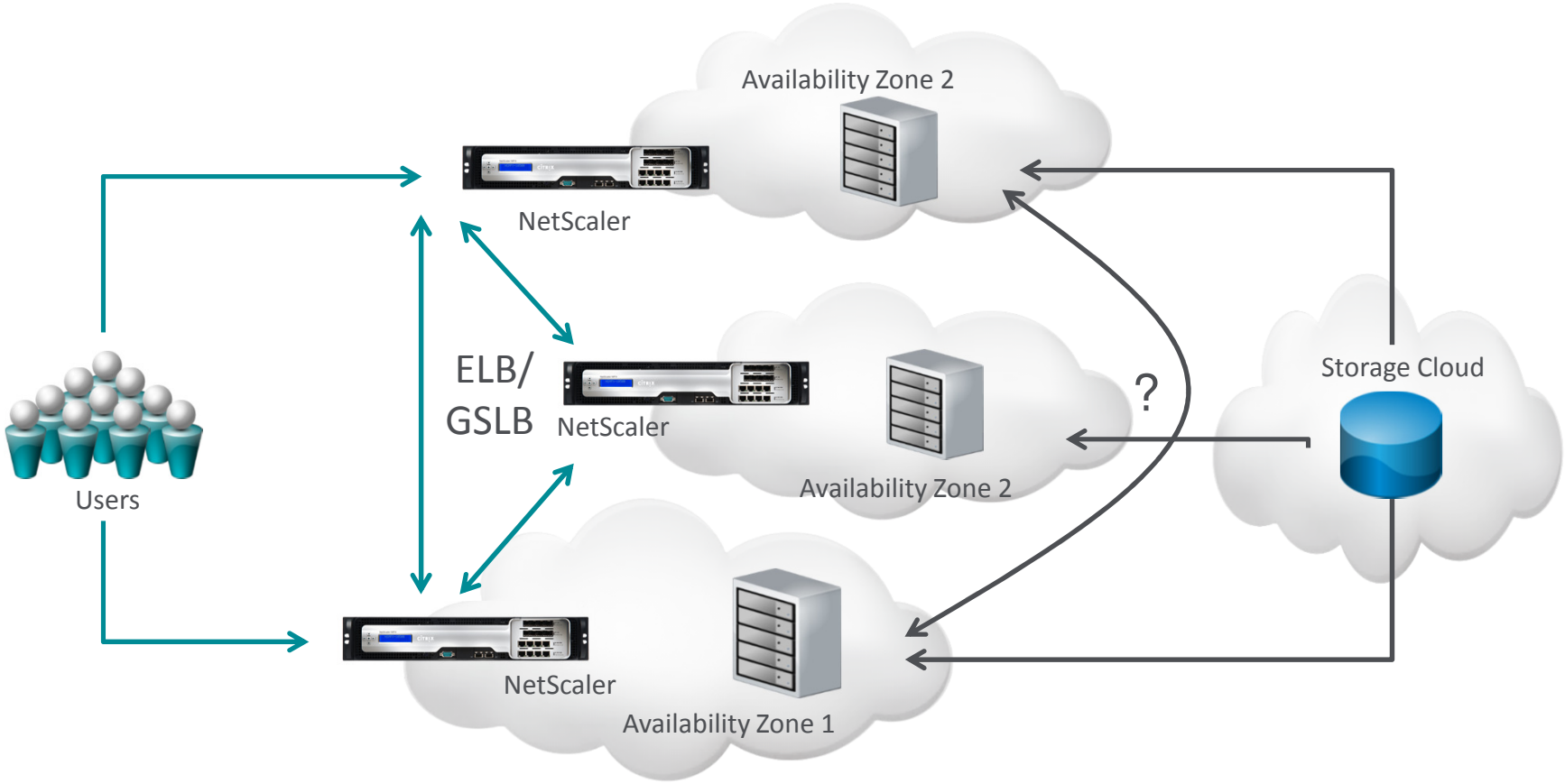
Designing a zone for a traditional workload



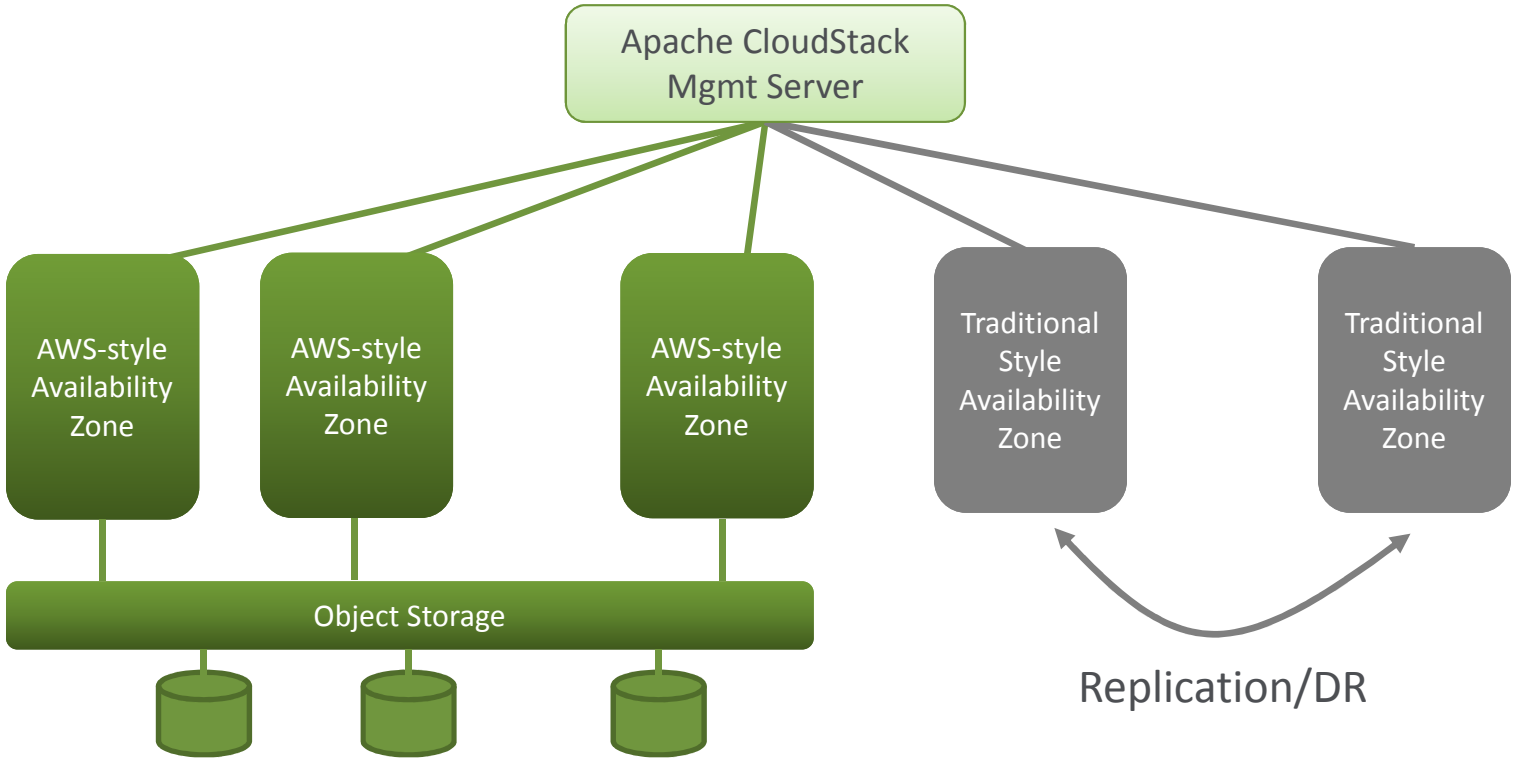
Designing a zone for an Amazon-style workload



Object store is critical for Amazon-style cloud



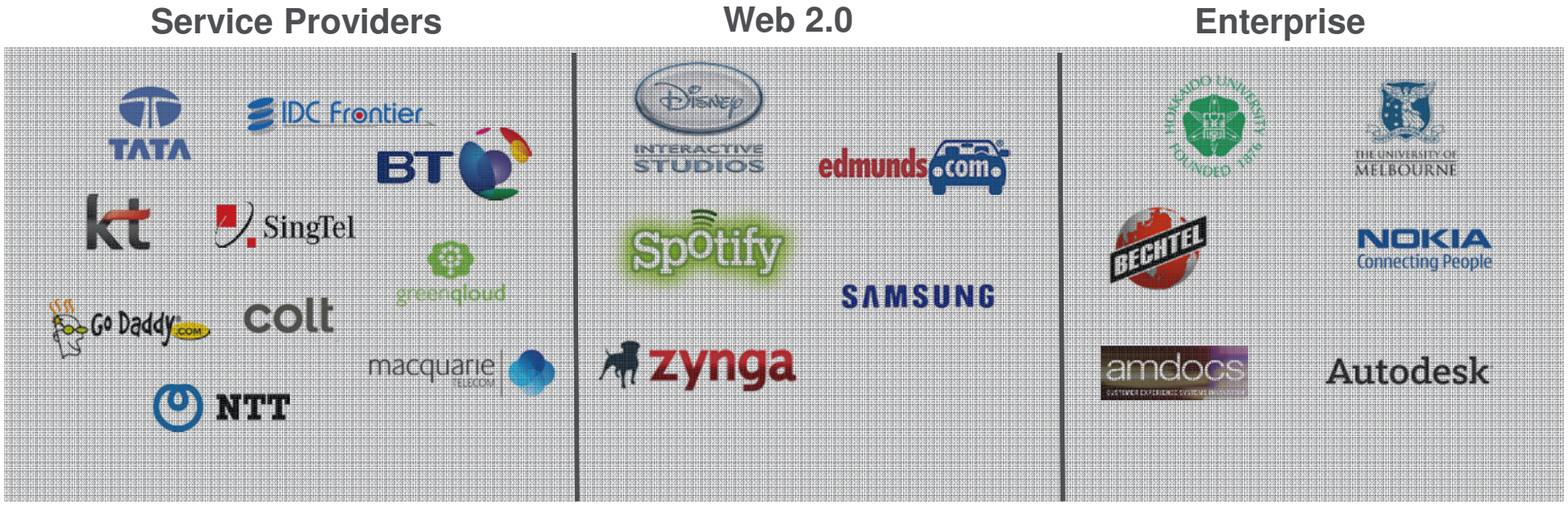
Same Cloud can Support Both Styles



Tests for a “true” cloud app

- Does it require SAN or VLAN?
- Does it run in multiple data centers?
- Does it involve a distributed object store?
- Is there a single point of failure?

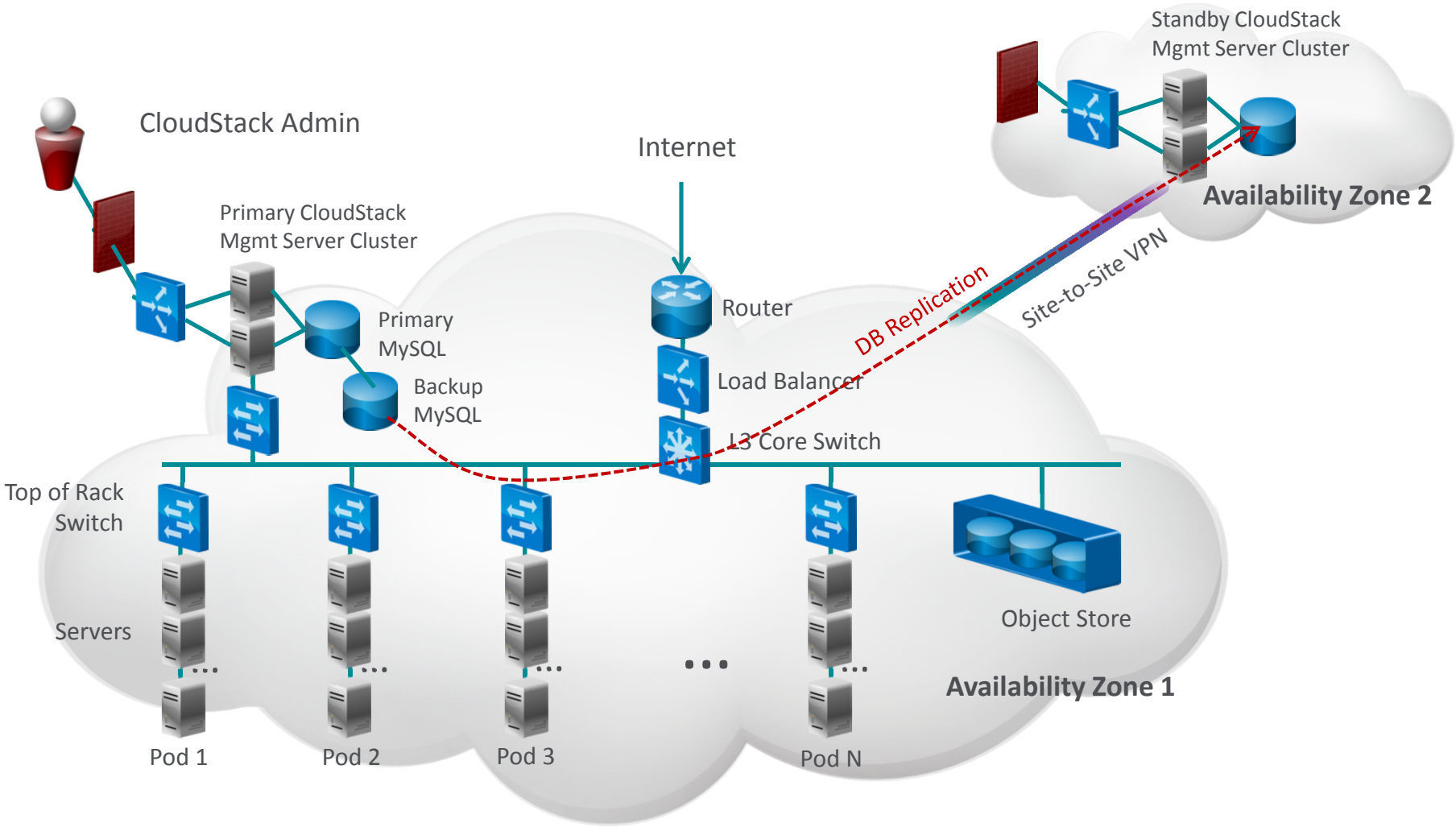
Learning from 100s of CloudStack deployments



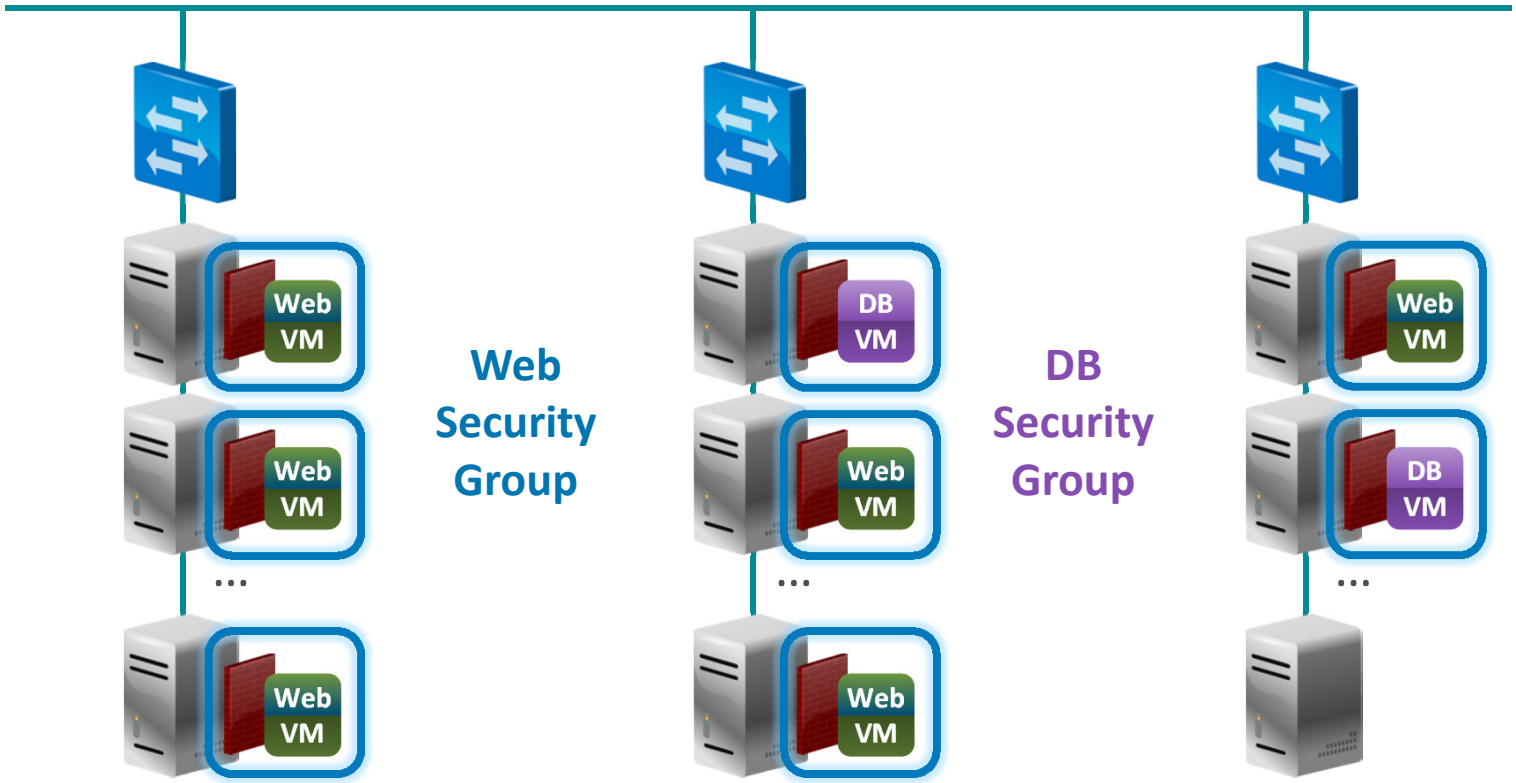
Traditional-style

Mostly Amazon-style

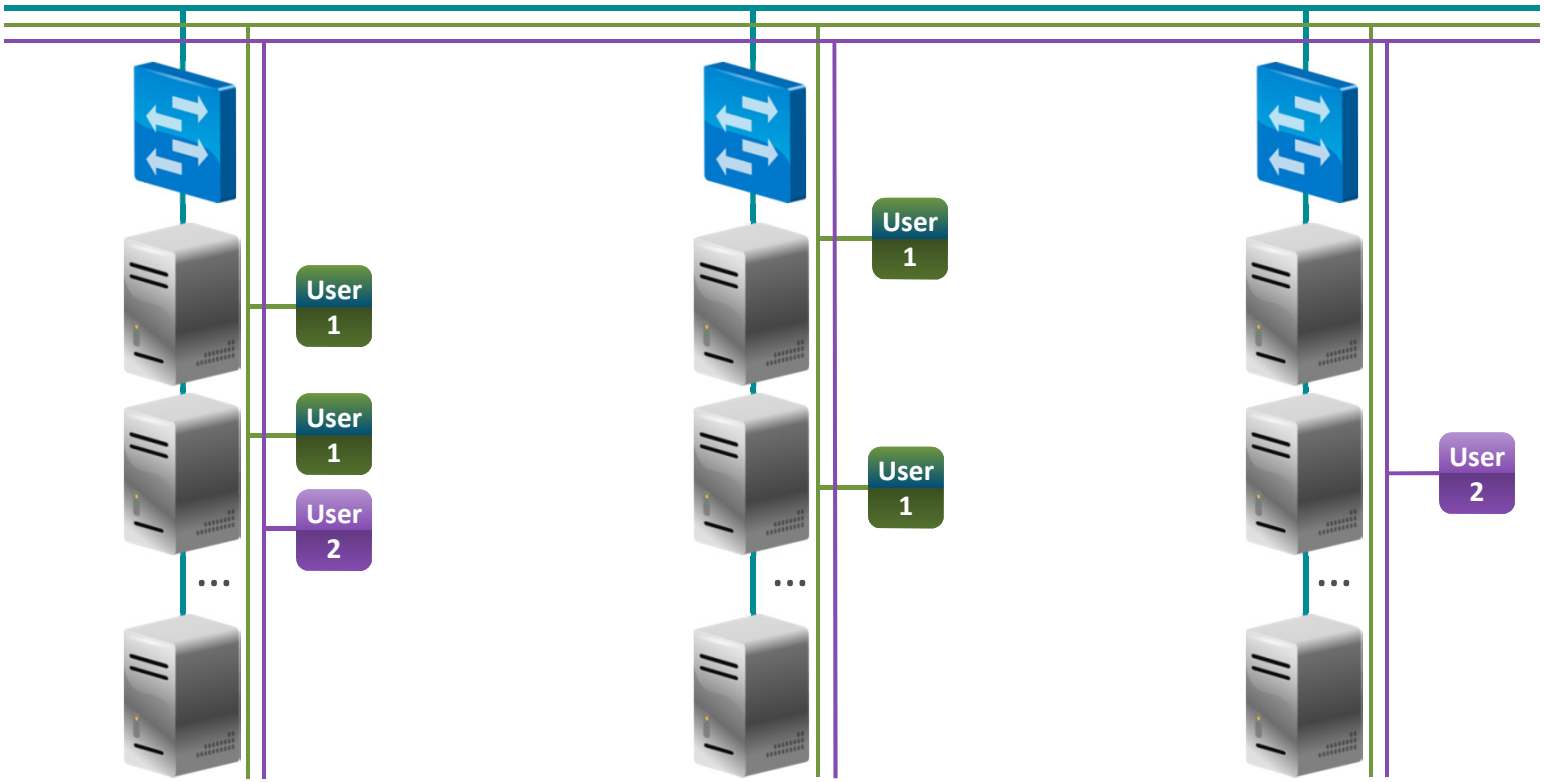
Mostly traditional style


CloudStack Admin
Internet
Standby CloudStack
Mgmt Server Cluster
Availability Zone 2
Primary CloudStack
Mgmt Server Cluster
Primary
MySQL
Backup
MySQL
Router
Load Balancer
L3 Core Switch
DB Replication
Site-to-Site VPN
Top of Rack
Switch
Servers
Object Store
Availability Zone 1
Pod 1
Pod 2
Pod 3
Pod N

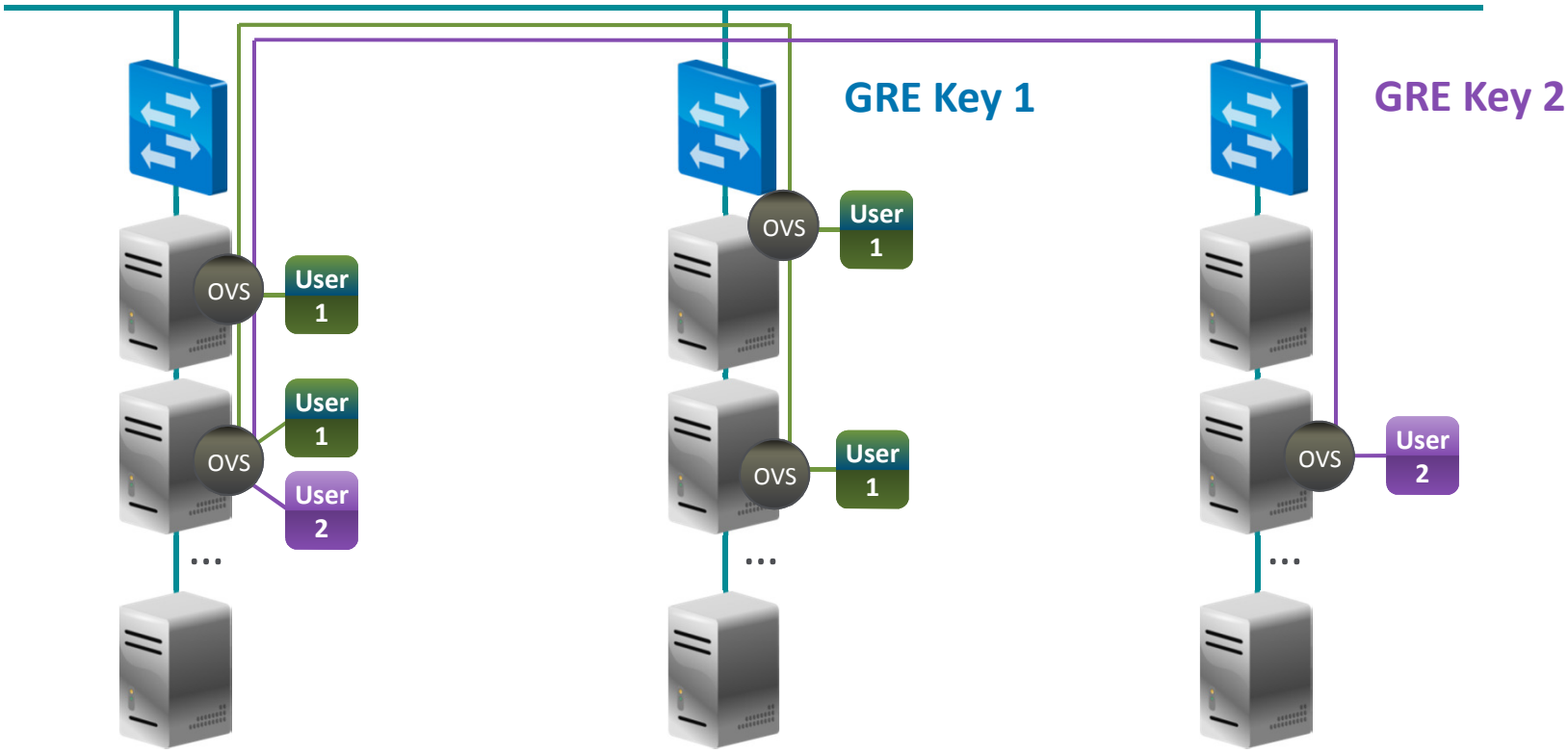
Layer 3 cloud networking (security groups)



Layer 2 VLAN networking



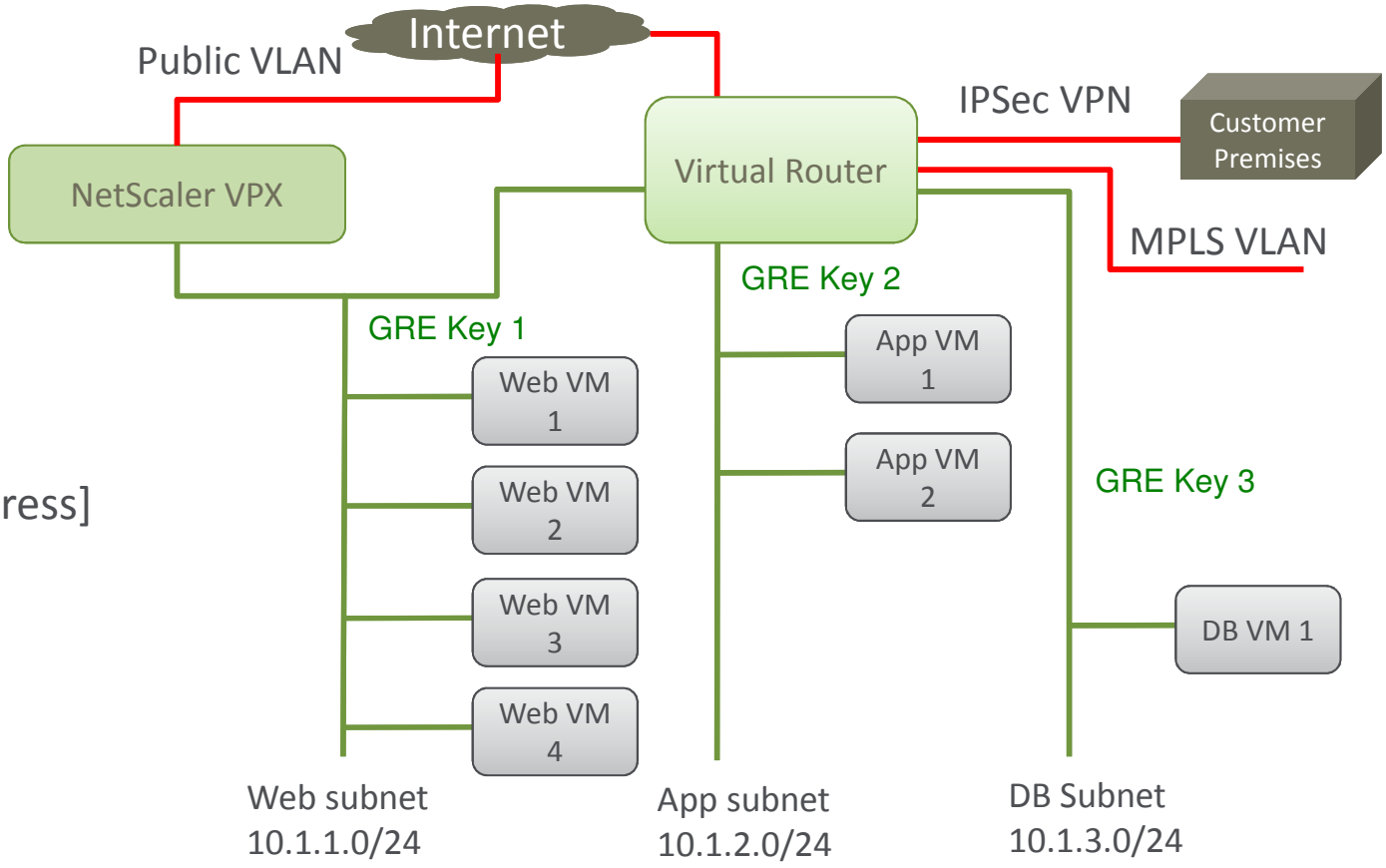
OVS networking



Multi-tier virtual networking

Network Services

- IPAM
- DNS
- LB [intra]
- S-2-S VPN
- Static Routes
- ACLs
- NAT, PF
- FW [ingress & egress]
- BGP



Network flexibility

Network Services

- ✓ L2 connectivity
- ✓ IPAM
- ✓ DNS
- ✓ Routing
- ✓ ACL
- ✓ Firewall
- ✓ NAT
- ✓ VPN
- ✓ LB
- ✓ IDS
- ✓ IPS

Service Providers

- ✓ Virtual appliances
- ✓ Hardware firewalls
- ✓ LB appliances
- ✓ SDN controllers
- ✓ IDS /IPS appliances
- ✓ VRF
- ✓ Hypervisor

Network Isolation

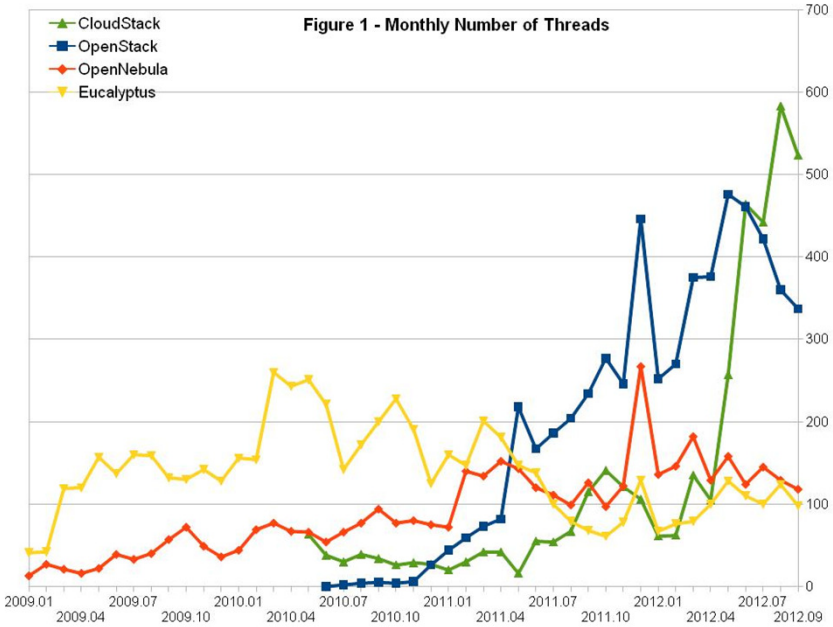
- ✓ No isolation
- ✓ VLAN isolation
- ✓ SDN overlays
- ✓ L3 isolation

“The Apache Way”

- Collaborative software development
- Commercial-friendly standard license
- Consistently high quality software
- Respectful, honest, technical-based interaction
- Faithful implementation of standards
- Security as a mandatory feature

Apache CloudStack Community

	Pre Apache Move (Jan 2012)	June Actuals
# of companies endorsing project	1	68
# of companies participating	10	140
# of developers working on project	40	238



Apache CloudStack community projects

- SDN
  - Nicira
  - Midokura
  - Big Switch Networks
  - Stratosphere
- Backup/DR
  - Sungard
- Networking
  - Cisco
  - Brocade (ADX)
- Smart Storage
  - Hadoop + S3 API for object store
  - NetApp (FlexPod, object store)
  - Basho RIAK CS
  - Caringo object store
  - Cloudian S3
- PaaS
  - CloudFoundry implementation through IronFoundry and Stackato teams
  - Engine Yard
  - Cumulogic
  - GigaSpaces

Workload requirements drive cloud architecture

There is real demand for SDN in cloud infrastructure

Open source developers drive cloud adoption

More info

http://cloudstack.org