Virtualized deployments:

Why this flexibility is key to future media workflows

Greg Shay, CTO
The Telos Alliance

Telos Alliance®

LOVE WHAT YOU HEAR

Telos Alliance: Our Vision

The Broadcast Facility Implemented using Modern Server Software Technology

- **Why IT Technologies:** What is driving this shift in technology?
- **Today's IT Solution:** Enabling new solutions in broadcast
- **Telos Alliance:** Enabling today and tomorrow's solutions

Information Technology is the foundation...

Recall a sharp focus:

***Why did the IT industry evolve to virtualized software*:**

1. Hardware is unreliable.

2. Human labor is expensive to react, reroute, reconfigure and repair.

3. Many copies of identical devices are less expensive than many different unique devices.

Broadcast has all these same needs…

Server based virtualized software solves these problems with:

- On-the-fly, automated replacement of failures from a pool of identical resources.
- Operations automation, less labor.
- Shorten fault response time and improve preparedness with automated recovery for the primary function.
- Orchestration of resource pool use, and deployment as needed
- Streamlined spares
- Improved operational efficiency in operations, configuration, maintenance & monitoring

The more mission critical the function is, (i.e. the airchain), the more these methods are needed.

This is more than the flexibility of "anywhere".

How are virtualized software products deployed?

There are two main server platforms for deployment:

Hypervisors
(Virtual Machines)

docker

Containerized
Software

docker

What is Docker?

Docker is a lighter weight method of OS-Level Virtualization, that allows software to be deployed portably, quickly, and reliably, *under the control of deployment software.*

Software running software = automation = less labor

How does it work?

- Relies on containers, which package everything needed for software to run.
- Containers eliminate the overhead associated with a full virtual machine.
- Multiple containers can be run on a single server with a single OS.

Why Docker? Designed for facility deployments

- Linux server software deployed as a **Docker** container.
- Can be run on premises or in the cloud (AWS, Google, Azure,…)
- As many instances as wanted can be run.
- If you prefer VM’s, Docker hosting containers can be run virtualized in a VM.

Note: Setting up these systems has its own complexity and special knowledge requirements: "DevOps" and "Infrastructure as Code“.

Redundancy with containerized products:

Redundancy means ‘keeping the product function running’.

The general strategy for redundancy using software in containers is:

- If a server stops, start up the required containers on another server.
- The very design of containers themselves is to make this as reliable, quick and easy as possible.
- *Kubernetes* is an orchestration system often used.

Economics:

Server uniformity and flexible network connectivity:

- Saving labor: Orchestration software controlling software
  One person can supervise the operation of thousands of machines.

- The work to be done can be moved away from failed hardware, using 1 for N redundancy.

- Can use geographic zones far enough apart to isolate from disaster outages and failures. (e.g. AWS Multi-AZ availability zones)

Additional benefits:

Standardized virtualization containers allow the use of pools of servers, without rigid central resource planning.

The same container can run on premises, or in the cloud, and can move back and forth controlled by automation on demand as needed.

This gives the flexibility to use the server resource that fits the economic equation best:

- Seasonal or on-demand consumption of 3rd party (cloud) resources.
- Fixed contract for a professionally manage resources, either on premises (hybrid cloud) or remote.
- Fixed 24/7 365 use constructed and managed in house.

Deployment Locations

Broadcast facilities require a range of deployments…

Future SW Products & Services

Historic Software in a Box

Cloud

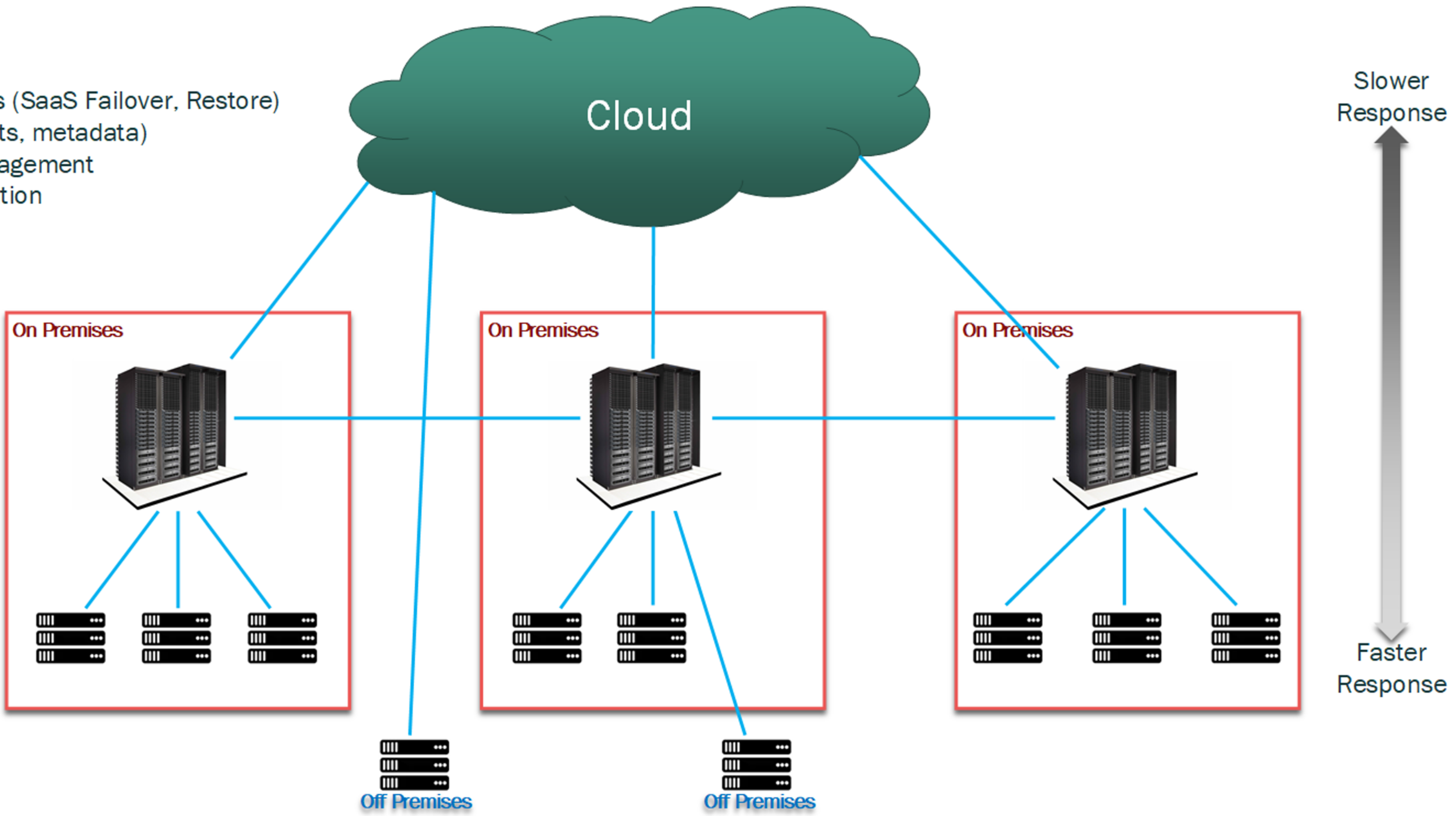
SaaS Applications
Disaster Recovery Solutions (SaaS Failover, Restore)
Data Sharing (config, presets, metadata)
SW Update Push/Pull/Management
Content Ingest and Distribution
Connection Management
Telos Marketplace

Fog

"Private or Hybrid Cloud"
Software VM and Dockers
Cloud Traffic Mgmt.
Local Disaster Recovery
Low Latency Audio

Edge

Connects to Cloud or Fog
HW I/O Devices (Nodes)
HW Control Interfaces
Local/User DSP/Audio
Highest Responsivity



Workflow, economy, and time-of-flight latency all influence *where* to deploy.

Some reality biting:

The virtualized/containerized automated transitions take time measured in handfuls of seconds, they are not instantaneous.

There is a fundamental economic question that cannot be avoided:

Can the broadcast or media workflow be defined as successful with a rare multi-second outage?

If so, can take full advantage of the economics of the best in class IT industry high availability by automation.

More reality…

If a rare outage of ‘seconds’ cannot be acceptable:

Operations are going to cost more and consume more resources for the deployment of simultaneous redundant end to end flows.

However, the high availability automation (everything previously talked about) is still *critically essential* for keeping all redundant paths up.

Thank You

Telos Alliance®

LOVE WHAT YOU HEAR