Unwrapping Cloud Native Services

John H Terpstra
Director Cloud Architecture and Software Engineering

John.Terpstra@Dell.Com

@JohnHTerpstra

OpenShift Commons Briefing – April 6, 2017

DELL EMC

Today's Agenda

The elevator moment

- The opportunity to gain mind-share on an important initiative

Setting the Stage

- Digital trends and changes
- Key software application engineering challenges
- Characteristics of deployment methods

Digital transformation opportunity

Elevator Opportunities

Unplanned - unexpected opportunities to communicate key information

Need to be ready

- Present a succinct case for digital transformation
- Garner interest
- Create readiness to invest

Seek commitment

- Investment in key factors
- Problem that is being solved
- Proposed solution
- Return on investment

Setting the Stage
Digital Trends and Changes

DELL EMC

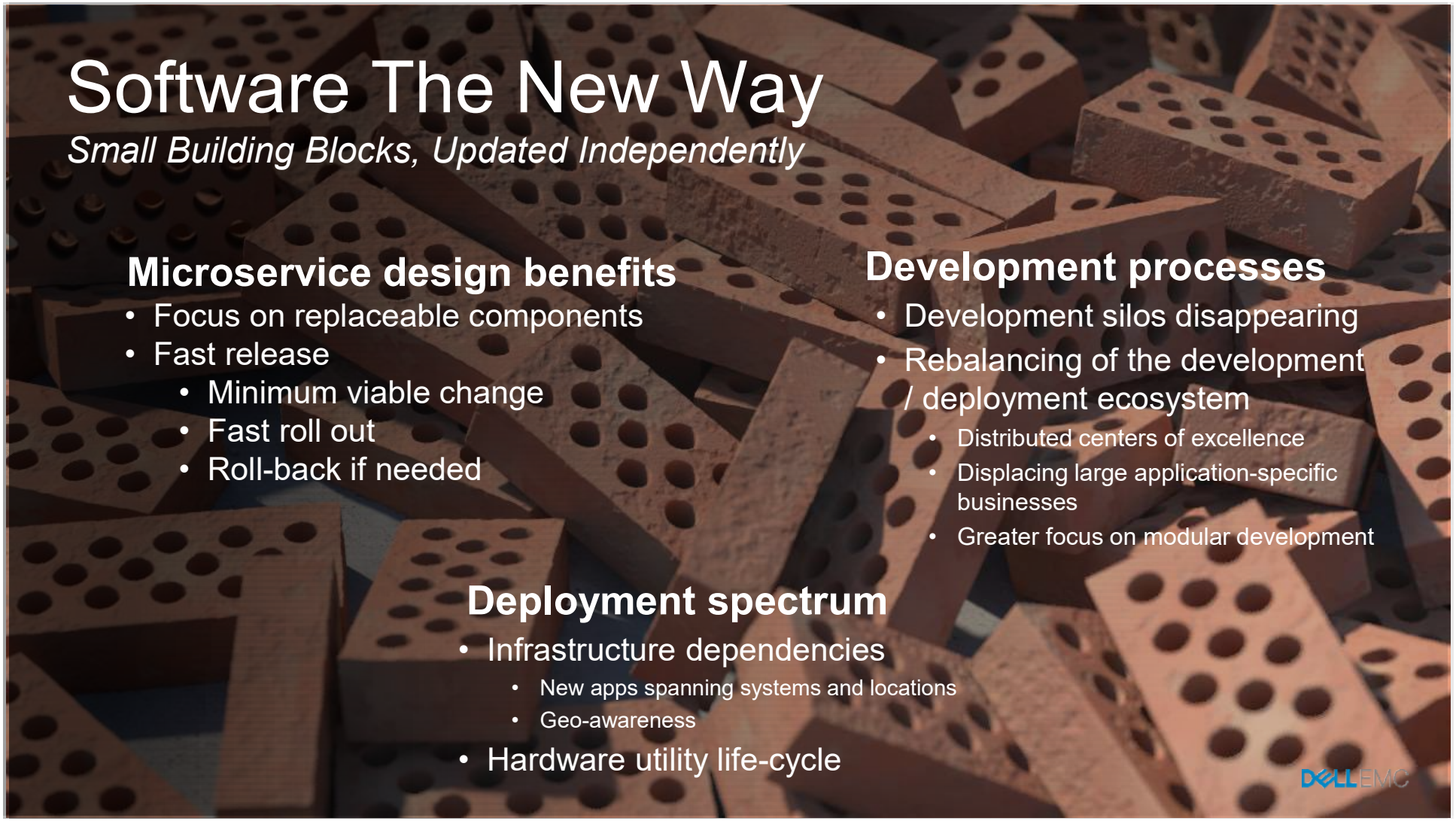

Consumers
Social Media
Digital Natives
Employment
WFH
Work-Life Balance
Organizational
Agility
Globalization
Confluence of FORCES of CHANGE
Technology
Cloud
IoT, Analytics
Digital Transformation
Software
Automation
Innovation
Disruption
Incumbents vs. start ups

Key software application engineering challenges

Something happened between 2005 and 2010

- Web and Internet technology and principles collided with Enterprise technology
- Business and Technology are moving faster
  - Commoditization
    - Processors, memory, and storage got faster and cheaper
    - Software got cheaper
    - More applications are used and developed
  - Web/Internet became “The Cloud”
    - Scale out is better - Distributed, parallel, loose coupling
    - SaaS and all it’s variations is here to stay
  - Direct impacts on software development and deployment
    - More choices in technology
    - More and improved architectures


Software The Old Way
Monolithic
Software application engineering
• Monolithic systems lack agility
• Large infrastructure overheads
• High cost of maintenance over time
• Minor failure places the whole at risk
• Complexity and waterfall development
• Development cycles are long and costly
• Inadequate mobility
• Pace of change v's opportunity life-cycles
DELL EMC


Software The New Way
Small Building Blocks, Updated Independently
Microservice design benefits
• Focus on replaceable components
• Fast release
• Minimum viable change
• Fast roll out
• Roll-back if needed
Development processes
• Development silos disappearing
• Rebalancing of the development / deployment ecosystem
• Distributed centers of excellence
• Displacing large application-specific businesses
• Greater focus on modular development
Deployment spectrum
• Infrastructure dependencies
• New apps spanning systems and locations
• Geo-awareness
• Hardware utility life-cycle
DELL EMC

Characteristics of Deployment Methods

Comparison of workload execution efficiencies

Deployment method	Bare metal	Virtual machines	Containers on Bare metal	Containers on Virtual machines
CPU and Memory Usage	Limited by workload	Maximized by VMs	Maximized	Maximized by VMs
Execution overhead	None	2-10%	2-3%	5-10%
Unit Outage impact	Whole machine	The Virtual Machine	The executing Pod	Pods within the VM
IO Capacity	Device limited	Shared across VMs	Allocated across Pods	VM resources shared by Pods
Compute efficiency	Highest	NUMA limited	Thread limited	NUMA/thread limited
Security Risk	Physical isolation	Virtual isolation	RBAC limited	Virtual isolation and RBAC limits
Tenancy	Single	Multiple	Multiple	Multiple
QoS	Process limited	Resource allocation	Noisy-neighbor risk	Noisy neighbors within VM allocation

Bare metal v's virtualization

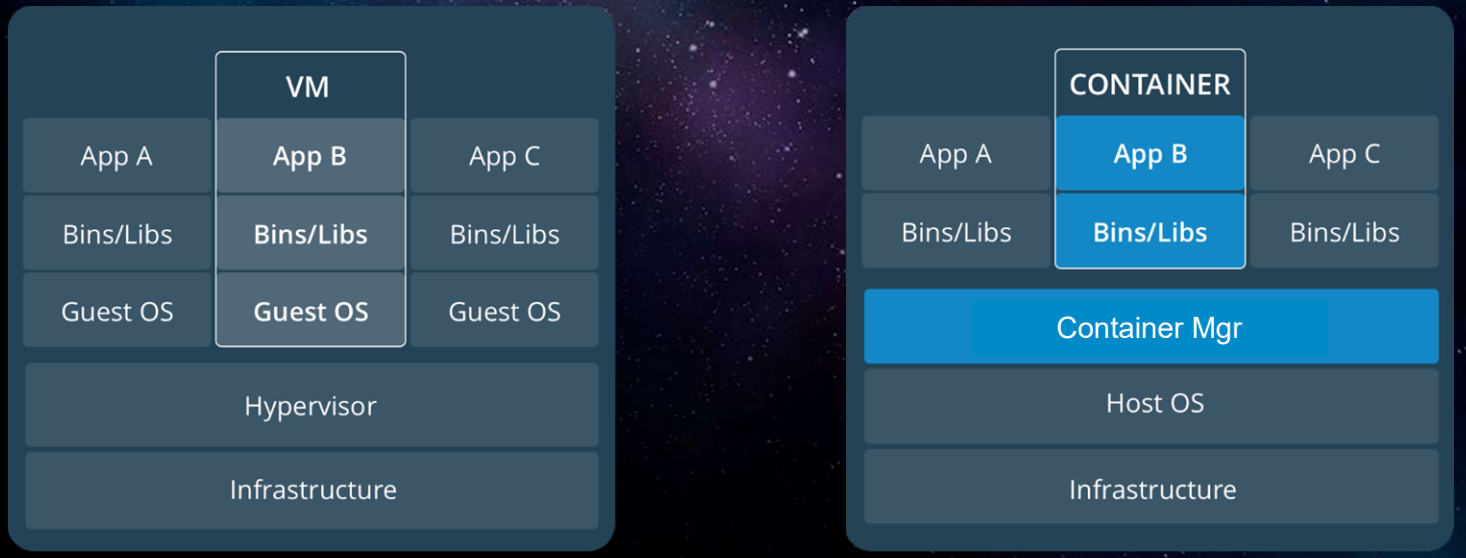
Virtualization Benefits

- Reduces capital costs and operating costs
  - Server consolidation, isolation, floor space
  - Improved energy efficiency
  - Task automation
- Minimizes downtime
  - Improved redundancy
- Increases
  - productivity, efficiency, agility, responsiveness
- Faster provisioning/deployment
- Better business continuity
- Simplifies data center management
  - Reduce hardware dependencies
  - Network virtualization
- Permits Software Defined Data Center (SDDC)

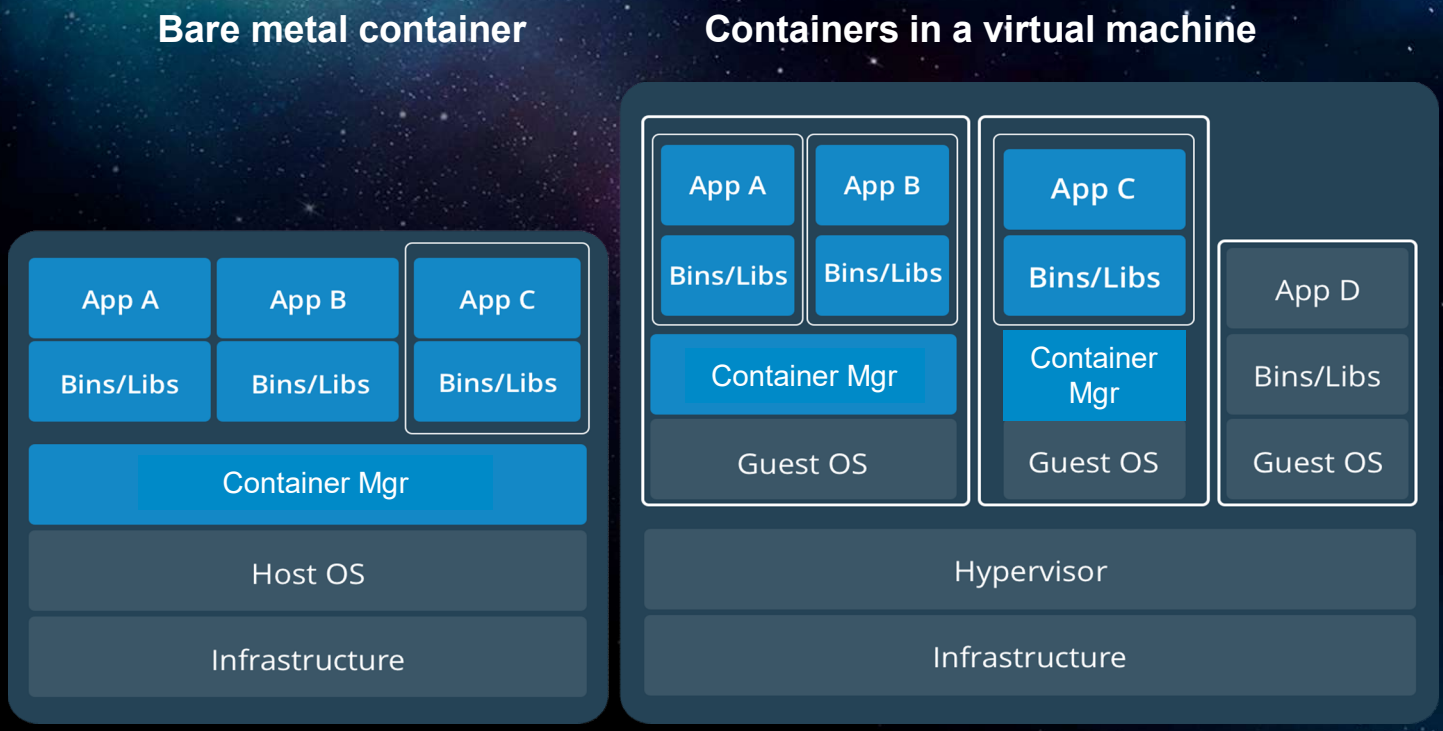
Disadvantages of Virtualization

- Hypervisor licensing costs
- Performance limits
  - High CPU or IOPS workloads don't virtualize well
  - Virtualization overheads
- More costly infrastructure
- Compatibility with older software
- Retraining costs
- Complexity of deployment
  - Server sprawl
  - Data storage management more critical
  - System Backup/Restore essential
  - Root-cause analysis can be more difficult
- Fragility
  - System failures can have greater impact

Resource layers: VMs vs. Containers



Resource layers: bare metal v's VMs



Containers v's VMs

Speed: Containers lower spin up latency

Management: Containers maintain service integrity of the application

Storage Footprint: Containers package deltas only, so smaller than VMs

Mutability: Containers layer on top of each other. Lower layered containers are immutable.

Code Decay: Containers preserve the relationship between layered dependencies. With VMs this must be managed independently

Hardware Dependencies: VM's are specific to CPU architecture. Containers are layered services that auto-match the host operating system CPU architecture

Damage Control: Container fault or failure affects only the local service instance. VM failures can have much higher outage impact

Rebuilds: Deployment of Containerized service will automatically rebuild the service instance from scratch. Rebuilding a VM can be very time-consuming

Launch times - VMs vs. containers

Instance Type	Start time (sec)	Stop time (sec)
Virtual Machines	30-50	5-20
OpenStack VMs	20-100	10-50
Containers	< 0.05	< 0.05

[illegible] of the memory and processor

Pros and Cons of Containers

Benefits

- Resource efficiency
  - No OS duplication
    - Requires less memory and fewer CPU cycles
    - Deploy 2-3x more apps per machine
- Provides consistent portable dev environment
- Faster startup/shutdown
  - Orchestration-friendly
- Separate name spaces
  - Process, Network, Mount, HostID, Memory

Liabilities

- Security
  - Containers and Cgroups live under /sys
    - Superuser privilege within the host or the container can violate security
- Trojan horse risks
  - Security of published containers is uncontrolled
- Excessive container packing
  - Heavily packed containers are more like VMs without the benefits of resource isolation

Guidelines for Containerized Cloud Services

Design Suggestions

- Single Responsibility
- Isolation via APIs
- Modular Resilience
- Launch location independence
- Stateless modules
- Minimum viable functionality
- Low deployment overhead
- Ensure modules are replaceable
- Automation – CI/CD

Digital transformation opportunity

DELL EMC

71% of customers agree that if they **do not embrace IT Transformation**, their firm will **no longer be competitive** in their market.


The business agenda *is* technology
IT = Competitive Advantage
Business Provides Incremental IT Budget
Software DNA Is Critical
DELL EMC

Dynamics of code change

4 R's

Retire high liability code	Retire
Migration to new processes	Retain
Pay down technical debt	Re-purpose
It as Digital production house	Re-engineer

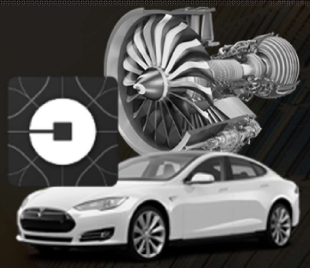
Cloud native microservices enable more flexible, higher performing, agile code – but at the cost of increased complexity. Complexity can be squashed by automation.

Intelligent Transformation

- **Monolithic code:** more time is spent on integration and testing than on delivering new functionality
- **Microservice code:** (aka Cloud Native code) – small time spend in development, fast deployment
- **Minimum viable principle:** Make changes small, release often, release early, get prompt feedback
- **Versioned APIs:** Permit expansion of modular functionality, isolate service dependencies
- **Auto-detect execution backpressure:** Gracefully recover from slow, failing, or failed services
- **Use machine learning:** Microservices are more complex to debug, log every operation

Faster Cycles = Competitive Advantage

Agile Development With Continuous Delivery
Accelerates improvement



New Applications & Smart Devices
Transform the Business

Data Analytics
Deliver New Insights

DELL EMC

THANK YOU

DELL EMC


DELL EMC