STORAGE DEVELOPER CONFERENCE

SDC 21

BY Developers FOR Developers

Virtual Conference
September 28-29, 2021

A SNIA Event

# Disaggregated Data Centers

## Challenges and Opportunities

Jason vanValkenburgh, Fungible

# Agenda

- Introducing Fungible
- The State of the Datacenter Today
- The Case for Disaggregation
- Our Approach and Results

*Objectives:*

- *Understand common challenges of modern data centers*
- *How Composable Infrastructure addresses these challenges*
- *Use Fungible's approach as an example*

# Of Interest...


SDC 21 Conference
COMPUTATIONAL STORAGE DIRECTIONS AT FUNGIBLE
September 29 | 12 PM PT
REGISTER HERE
Jai Menon
Chief Scientist, Fungible
FUNGIBLE

# About Fungible

**PRADEEP SINDHU**
Co-Founder, CDO

**BERTRAND SERLET**
Co-Founder, COO

**ERIC HAYES**
CEO

**Founded:** 2015 to revolutionize the *agility, security, performance*, *reliability,* and *economics* of ***all*** Cloud Data Centers

**Technology:** a new class of microprocessor, the *Fungible DPU™*, *software for the DPU*, *systems*, and *solutions* built using the DPU; 60+ patents, many fundamental and ground-breaking

**Intellectual Property:** *silicon, software, systems,* and *solutions*

**People:** world class, experienced team with deep expertise in *silicon, software* & *systems* across *compute*, *network* & *storage*

Top-tier investors

Mayfield | NORWEST VENTURE PARTNERS | SoftBank Investment Advisers
Battery | REDLINE | WRV CAPITAL

# Bottom Line

- Data center architectures need to transform just as applications have - the mismatch results in low efficiency, overprovisioning and excess cost
- Composable Disaggregated Infrastructure (CDI) is the path forward
- CDI must not come at the expense of performance, security, or cost
- This can only happen with new, purpose-build combinations of silicon, software, and systems

# Data centers are Under Pressure

Software architecture has changed, but datacenter architecture has not

- Cloud-native, microservices
- Scale-out applications
- Data services moved from dedicated devices (SAN arrays, etc.) to hosts (e.g. SDS, HCI)

Implications

- Increase in east-west traffic, network congestion as workload / platform domain size increases
- More CPU cores spent on storage and networking

# The Relationship between CPUs and I/O Has Changed

- Networks and NVMe storage are now significantly faster than servers
- Moore's Law Flattening
- Network speeds increasing - ~8 CPU cores to push 100G


Implications

- Low network utilization
- More and more resources used for the same capabilities and SLAs

# Storage and Server Utilization is Low

- Local NVME used for booting & performance
- Traditional storage deployed in clusters / pods to minimize performance domain

Implications

- Stranded local storage that is underutilized
- Applications deployed close to storage to ensure consistent experience
- Workload / platform silos, sized for peak, not average
- Low server utilization

# Tradeoffs Drive Decisions - Choose Wisely!

- SDS provides data durability, data reduction and other services via (slow and expensive) x86
- Local NVMe delivers performance but not necessarily durability
- NVMe over TCP support not available on all platforms

Implications

- IOPs are swallowed by data services running in software
- Need to choose between performance and durability up front

# Security is Paramount

- Rise of side channel attacks
- Encryption in flight and at rest is now a given
- Tenancy needs appropriate isolation

Implications

- Mitigations and security services sap performance, costing cores and $
- Hosts may not necessarily be trusted
- Choice between security & performance, and cost - pick two!

# The Results - Silos

Wasted Assets

Wasted Time

Wasted Power

Wasted Money


*It all comes down to a lack of trust that performance can be consistent and predictable when the deployment domain is the entire DC*

# Infrastructure Expectations Have Changed

- “Cloud-First” is Now “Cloud Best”
- Traditional Infrastructure brain drain
- Security & isolation are key

Implications

- No one wants to regress while repatriating - a cloud experience is table stakes
- Multi-tenancy matters

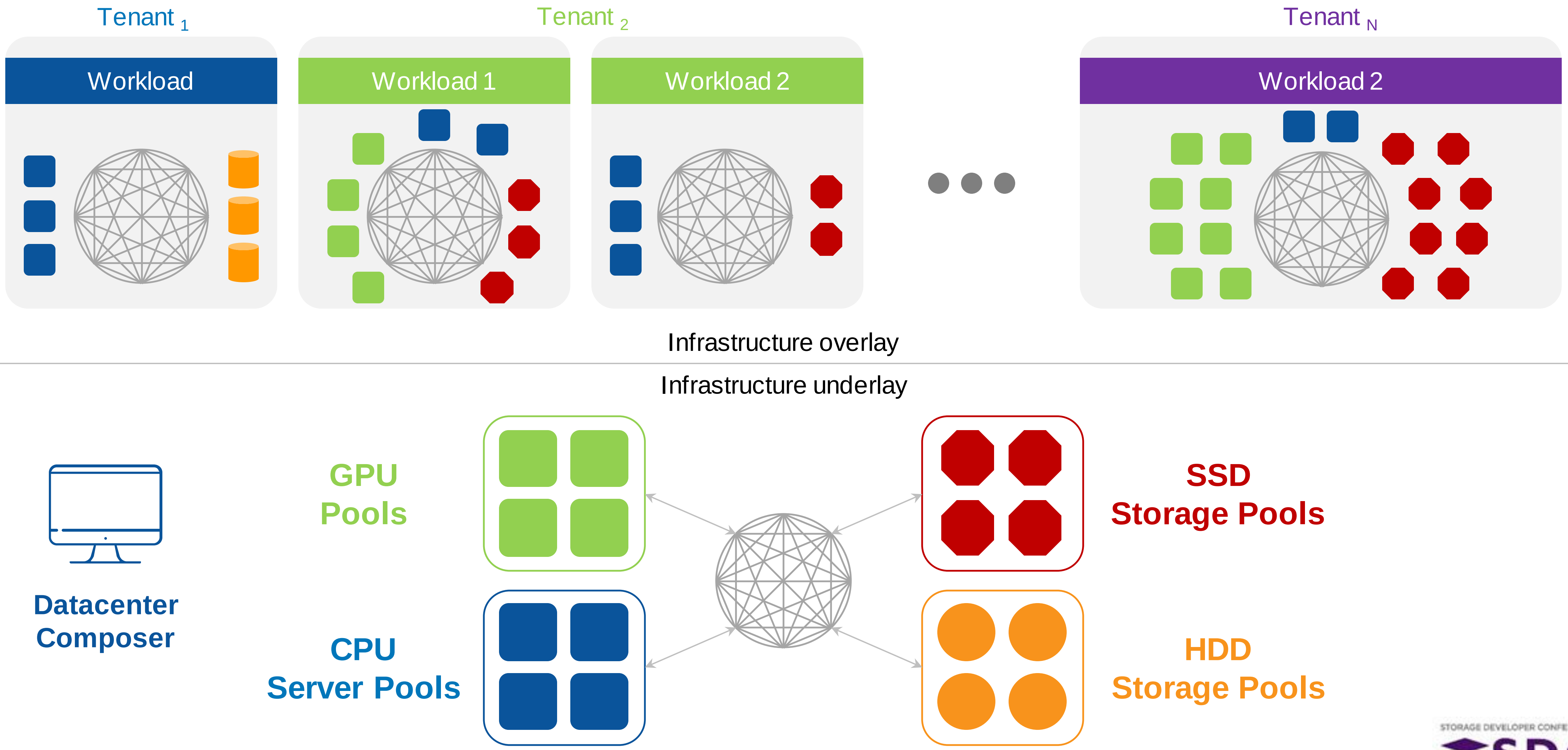


STORAGE DEVELOPER CONFERENCE SDC 21

# Relieving the Pressure

## The Case for Disaggregation

# The Disaggregated Data Center



# Composition Use Cases

## Composable Diskless Servers to Eliminate Local Storage and Reduce Cost

Servers see remote NVMe-over-TCP volumes as local and may boot from them

Eliminates instances of low-utilized local storage in favor of thin-provisioned storage from the fabric

## On-Demand, Pooled GPUs and FPGAs to Accelerate AI/ML Deployments

Dynamically assign GPUs to servers as needed, without servers needing special software (GPUs are “local”)

Unlock expensive assets, and improve performance by increasing GPU-to-CPU ratio (up to 32 GPUs to a host)

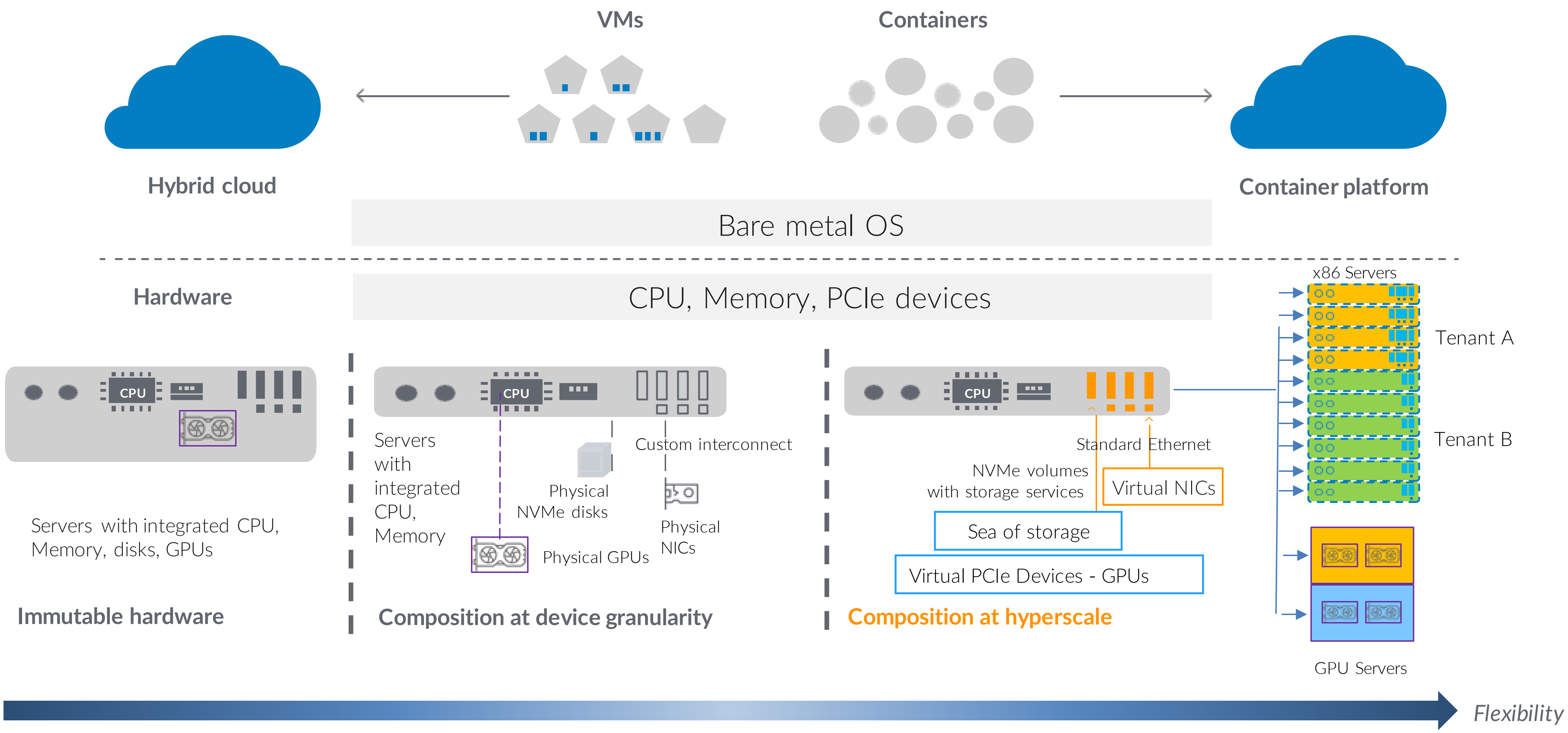
## Infrastructure- or- Metal as a Service to Automate and Delegate Workload Deployments

Tenant-level self-service portal to managing application and server deployments like the Cloud

Optional virtual networking and Network Controller provide server isolation and network flexibility

# Composition Makes Immutable Hardware Flexible



# Effects of Scalable Disaggregation and Composition

Automated / screwdriver-less adds/moves/changes

Fewer assets to serve the same workload - servers become highly interchangeable, storage pooled and thin provisioned

Relieves line-of-sight pressure on emerging and future requirements

No forklift upgrades - Independent asset / deployment life cycles between servers, GPUs, and storage

Environmental relief - put hot GPUs in a separate locale where you can better cool them, spare hot CPU cycles and replace with low-power, high performance DPU ones

# Prerequisites for Successful Disaggregation

Prerequisites

- Disaggregation must not come at the expense of performance and utility
- Must be invisible to the workload, as who's consuming IT is different than who's providing it
- Prevent Fabric Fatigue and exacerbating skills & staffing shortages

Conclusions

- Need high-speed storage without scale and deployment constraints
- We need a low-latency, congestion-free Ethernet fabric - use skills and equipment you have today
- Bare metal without special host software

Successful disaggregation comes down to scalability, network performance and ease of deployment

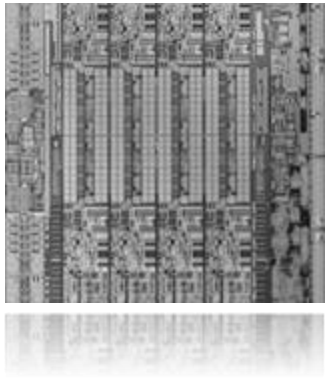
# The Fungible DPU

## A New Class of Microprocessor Purpose-Built for the Data-Centric Era

The Fungible DPU is a new class of programmable microprocessor that:

- Enables 10x more efficient execution of data-centric workloads
- Implements a scalable, low tail latency, congestion-free TrueFabric™ endpoint

CPU

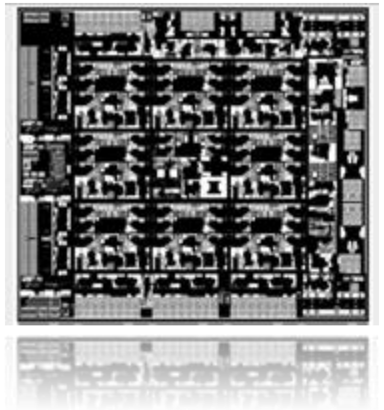


General-purpose

GPU

Vector floating point

Fungible DPU



Data-centric

# Foundational Technology


800G IO PROCESSING


200G IO PROCESSING

## PERFORMANCE & ECONOMICS

Executes data-centric workloads > 10x faster than a CPU, completely offloading network, storage and security from the CPU

## SCALABILITY

Addresses inefficient data interchange between nodes at large scale by enabling an ultra-low latency, tail latency network TrueFabric™

# Mix of Silicon and Software To Leverage Strengths of Each

Orchestration Software

Embedded Software

**Composer**

Topology
Configuration
Monitoring
Telemetry
Policy
Composition

**Control Plane**

- Agents
- Linux
- FunVisor

**Data Plane (FunOS)**

**Network**
- IPoE
- Fabric
- Full TCP
- Full RDMA
- Full TLS
- VXLAN
- Stateless Offload
- Encryption
- Granular QoS

**Storage**
- NVME-o-FCP
- NVME-o-TCP
- Initiator + Target for
  - Block Store
  - KV Store
  - File Store
- Erasure Coding
- Compression
- Encryption
- Pooling

**Virtualization**
- VXLAN
- V-Router
- V-Switch
- Per VF QoS
- Full BMV
- Full RPC Offload

*All Accelerators exposed via PCIE VF's*

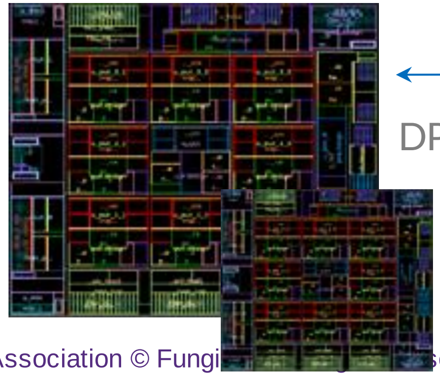
**Security**
- PUF
- Secure Enclave
- Key-Gen
- Key-Store
- TLS
- Regex
- Hash
- DLP

**Analytics**
- Sort
- Filter
- Join
- Select
- Map
- Reduce
- Lambda

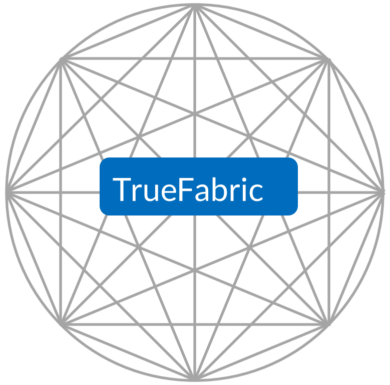
**FunOS Nucleus (scheduling, timer, memory management)**

**DPU**

TrueFabric™ Uniquely Enables Hyper-Disaggregation

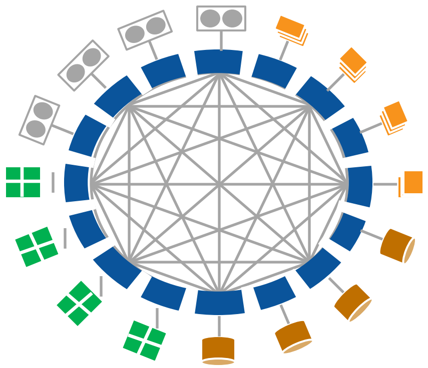


DPU Silicon



# DPU Powered Data Center Solutions

## Fungible Composer



- *Turnkey Multi-Tenanted Data Centers*
- *Agile & Efficient Resource Deployment*
- *No Network or Application changes Required*

## Fungible Compute Cards


- *Tightly coupled Security, Storage, Network HW Accelerators*
- *Fully Programable*
- *Multiple Form Factors – 50G, 100G, 200G*

## Fungible Storage Cluster

- *Cloud-Scale Architecture*
- *High Performance*
- *Efficient Durability & Cost Effective*
- *No Compromise Enterprise Features*

 Fungible Confidential

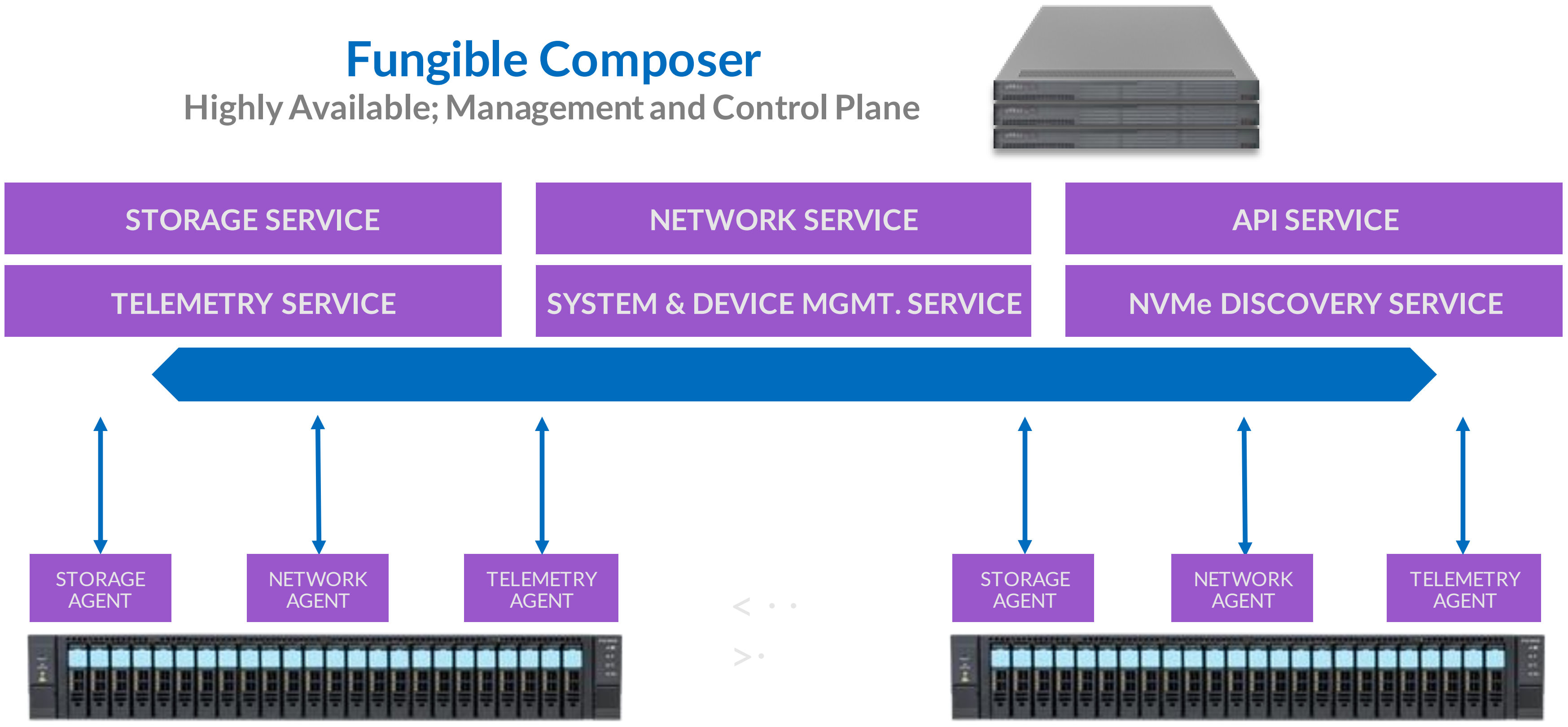
# Key Fungible Storage Cluster Features

| Feature | Benefit |
|---|---|
| Storage pooling | High storage utilization; Independent storage scaling |
| High performance (IOPS/GB, latency, block and file) | Networked storage @ local SSD performance |
| Supports VMs, containers, bare-metal | Workload consolidation |
| Multi-tenancy (per vol protection, encryption, QoS) | Workload consolidation |
| Scale out | Pay as you grow; manage storage cluster not box |
| Leadership compression (without performance loss) | TCO; high storage utilization |
| Encryption with minimal performance impact | Security; workload consolidation |
| REST API to manage for PBs of data | TCO |
| Rack scale resiliency @ low overhead | Very high reliability @ low cost |

# Cloud-Scale Architecture

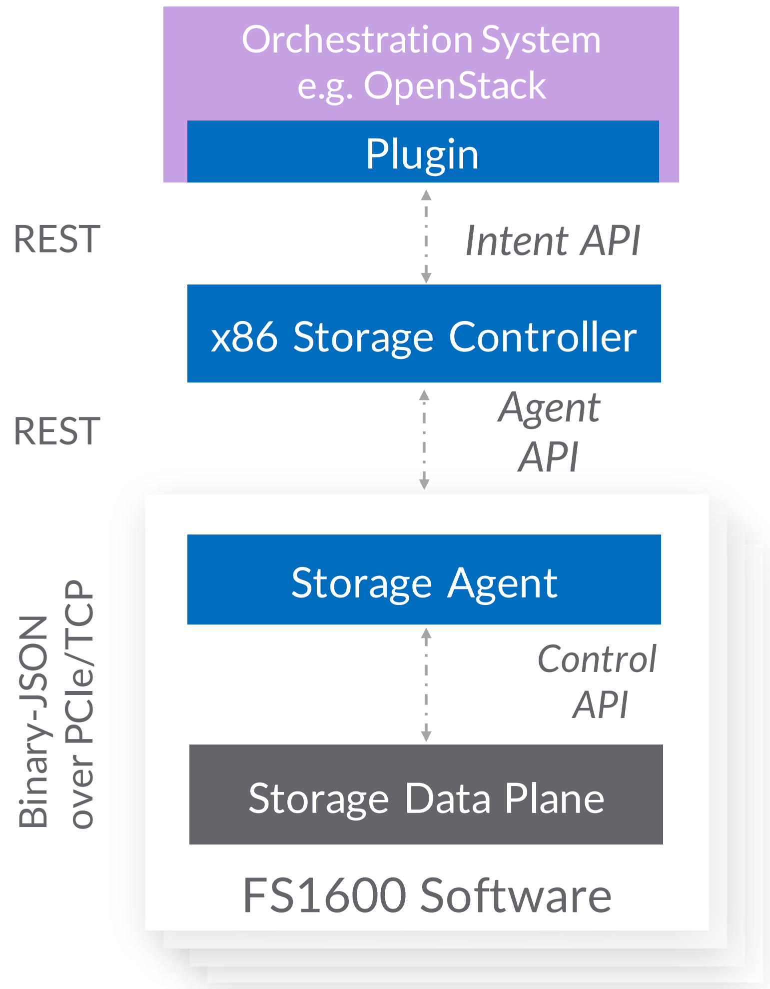
## MANAGEMENT & CONTROL PLANE
- API-First
- Intent-Based
- Micro-Services Based

## DATA PLANE
- Scale-Out
- High Performance
- Elastic Architecture



# Control API Hierarchy




**INTENT BASED**

*"Create Volume for VM Root"*

**HIGH LEVEL COMMANDS**

*"Create Raw Volume with Encryption and NVMEoF Access"*

**LOW LEVEL COMMANDS**

a. Create “Raw” Volume with Encryption
*b. Create NVMEoF Controller*
*c. Attach Volume to Controller as Namespace*

Resources: Device, Volume, Controller

Volume Management Commands: Create, Delete, List, Stats, etc.

Telemetry: per resource instance. E.g. IOPs, bandwidth, latency, capacity, errors, etc.

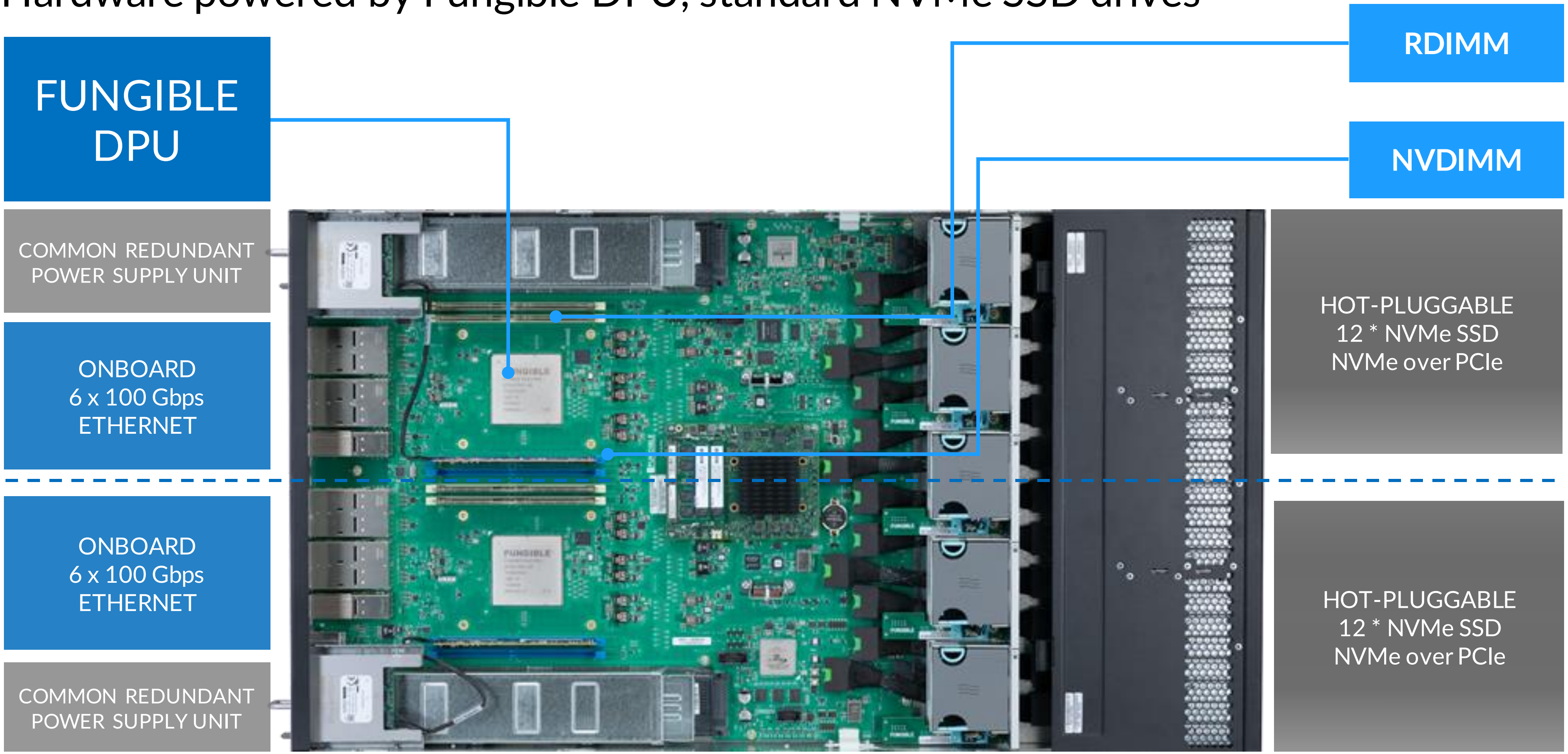
JSON based API

Volumes are tracked using UUIDs

# Under the Hood

Hardware powered by Fungible DPU, standard NVMe SSD drives



# Unrivaled Performance With The DPU

| | READ | WRITE |
|---|---|---|
| **Raw**<br>**Single node** | 15M IOPS<br>60 GBytes/sec | 4.4M* IOPS<br>24 GBytes/sec |
| **Network Protected (RF=2)**<br>**Two nodes**<br>(SSD and node failure protection) | 15M IOPS<br>60 GBytes/sec | 4M IOPS<br>16 GBytes/sec |
| **Network Protected 4+2 EC**<br>**6 nodes**<br>(SSD & node failure protection) | 20M IOPS<br>80 GBytes/sec | 6M IOPS<br>24 GBytes/sec |

**LINEAR PERFORMANCE SCALING MEASURED UP TO 16 NODES, EXPECT CONTINUED LINEAR SCALING BEYOND THIS**

* SSD limited

## Oracle RAC OLTP 70/30 Mixed Performance on Fungible FSC

## (Measured Test results)

*Using Linux NVMe over TCP Storage Initiator on Host*

| | SLOB - OLTP - 70/30 - Except [redacted] (75/25) | | | | |
|---|---|---|---|---|---|
| | Read IOPS | Write IOPS | Total IOPS | Read Lat(ms) | LFW Lat(ms) |
| 3FS - 12 x EC Vols - 2node RAC -384 users | 613,057 | 191,246 | 804,303 | 0.322 | 0.44 |
| 3FS - 12 x EC Vols - 4node RAC - 784users | 808,326 | 250,177 | 1,058,503 | 0.294 | 0.378 |
| [redacted] 10node RAC (upto 450 users) | 432,267 | 104,258 | 536,525 | 0.387 | 0.923 |
| [redacted] - 8node RAC - 512 users | 335,728 | 143,884 | 479,612 | 1.200 | N/A |
| [redacted] - 4node cluster - 4 Oracle Instances | 217,573 | 79,102 | 296,675 | 0.660 | 1.01 |
| [redacted] - 4node cluster - 8 Oracle Instances | 299,134 | 128,200 | 427,334 | ~0.800 | ~1.5 |
| [redacted] - 1 x Brick - 4node RAC - 96 users | 373,310 | 116,635 | 489,945 | 0.490 | 0.724 |

| Oracle RAC Node Components | Quantity / Description |
|---|---|
| Server Type | 4 x Supermicro |
| Memory | 256GB per server |
| CPU | 2x20 Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz per server |
| Network Card | 1 x Mellanox ConnectX-5 – 100GbE per server<br>1 x 10GbE per server |
| Direct Attach Storage | 1 x SATA SSD as Boot Disk per server |
| Disaggregated Storage | 12 EC volumes for ASM DATA Diskgroup for SLOB tablespace |
| Operating System | Red Hat Enterprise Linux 8.2 x86_64 |
| Oracle Grid / Database Software | Oracle 19c |

<u>Note:</u> To test physical I/O, the Oracle RAC SGA was configured with 8GB of memory. Specifically, db_cache_size was configured with 128MB.

# Ability to be Fast Alone Does Not Move the Needle

- End to End Speed pointless without ability to avoid congestion
- Multiple paths (“traffic lanes”) don’t help unless you can leverage them
- ECMP is not dynamic enough in the microservices world - hash collisions cause backup between “elephant and mice” flows
- We need to avoid congestion in the first place and open up all lanes!

“It’s more fun to drive a slow car fast, then to drive a fast car slow.” —Unknown

# TrueFabric™ Avoids Congestion Before it Starts

Packet switched, based on open standards (IP over Ethernet)

Full cross sectional any-to-any bandwidth with no constraints

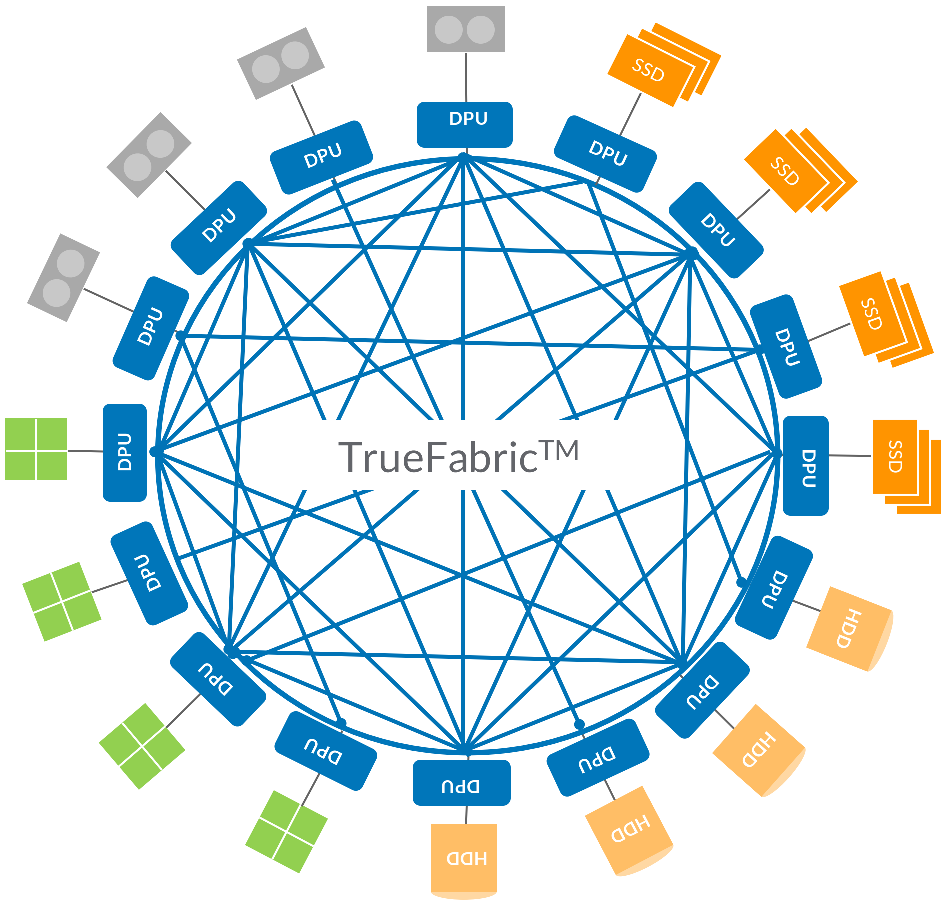
End-to-end congestion control and error control

Low zero load latency & excellent tail latency

Scalable from 10 to 100s of thousands of servers

End-to-end encryption

Software-defined connection topologies



# The Power of a Fungible DPU™ Network

## Cost Efficiency

- Fully standards compliant with IP/Ethernet enabling the use of standard TORs and Spines
- N:1 redundancy at the Spine layer and up to 8:1 redundancy at the TOR without extra hardware
- Support for well-known communication paradigms: message, RPC, byte stream, RDMA,…

**Leverages existing investments**

## Performance

- Full cross-section bandwidth from any server to any other up to 25.6 Petabits/sec
- Flat two-tier spine-leaf topology all the way to 25.6 Petabits/sec

**Unclogs East/West traffic**

## Scalability

- Ability to partition physical network into multiple disjoint Fabrics under software control
- Incrementally expandable while running live traffic

**Breaks down network silos**

## Security

- Built-in strong encryption end-end (AES); can be disabled if needed

**Secure from snooping**

## Predictability

- Low zero-load latency and excellent P99 tail latency even at > 90% offered load
- Very low jitter, typically under 300ns even at > 90% offered loads

**Avoids over-provisioning**

## Resiliency

- End-to-end congestion and error control at the packet level
- Ultra-fast failure detection and recovery from all failure types (< 100 microseconds)

**Self heals before applications notice**

# Multi-Tenancy and Workload Isolation Is Essential

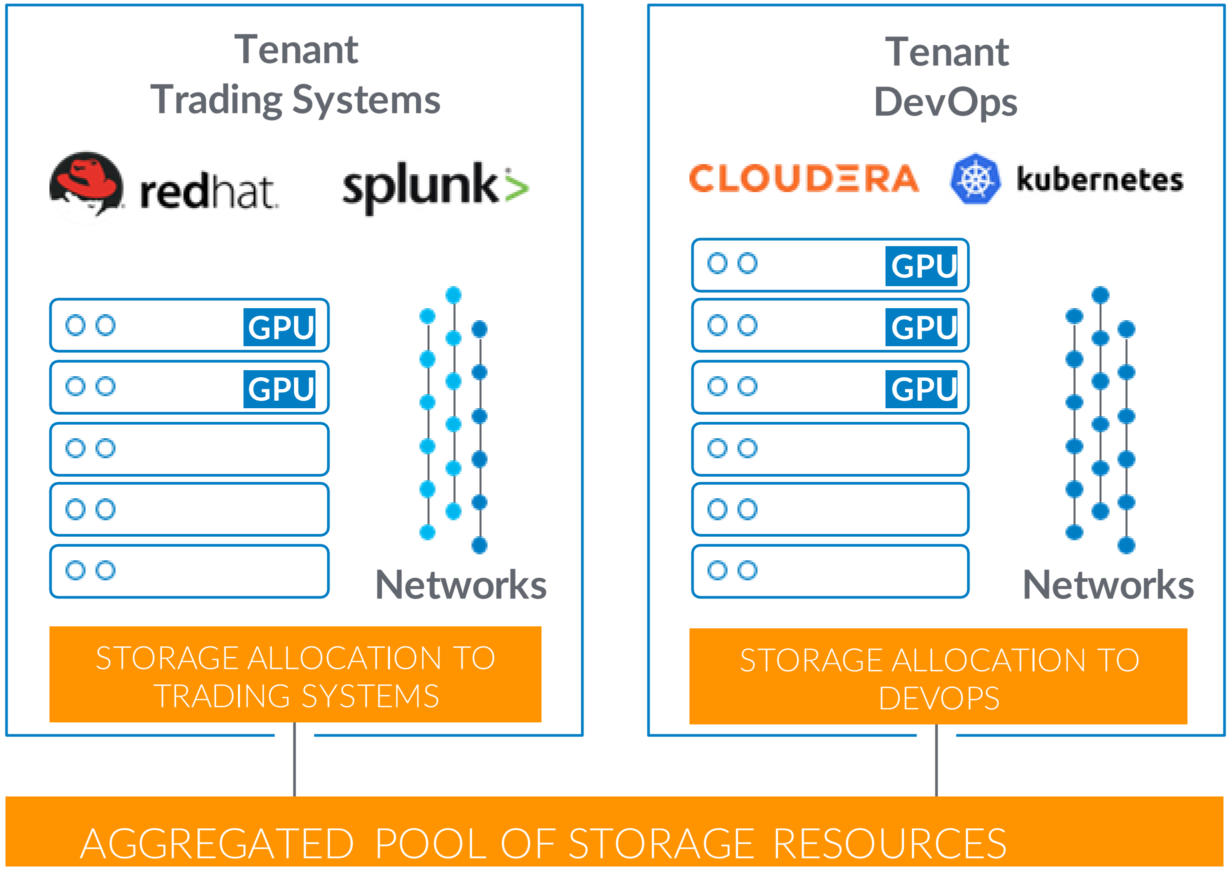
## Simplified IT through Self-Service

Partition server, storage, GPUs, and networking resources to customers

- Diskless Servers are hard partitioned
- Storage Capacity/IOPs and GPUs are allocated from pools
- Optional virtual networking

Tenants manage hardware within their isolated partition

True self-sufficiency via tenant's independent management portal, APIs and identity manager



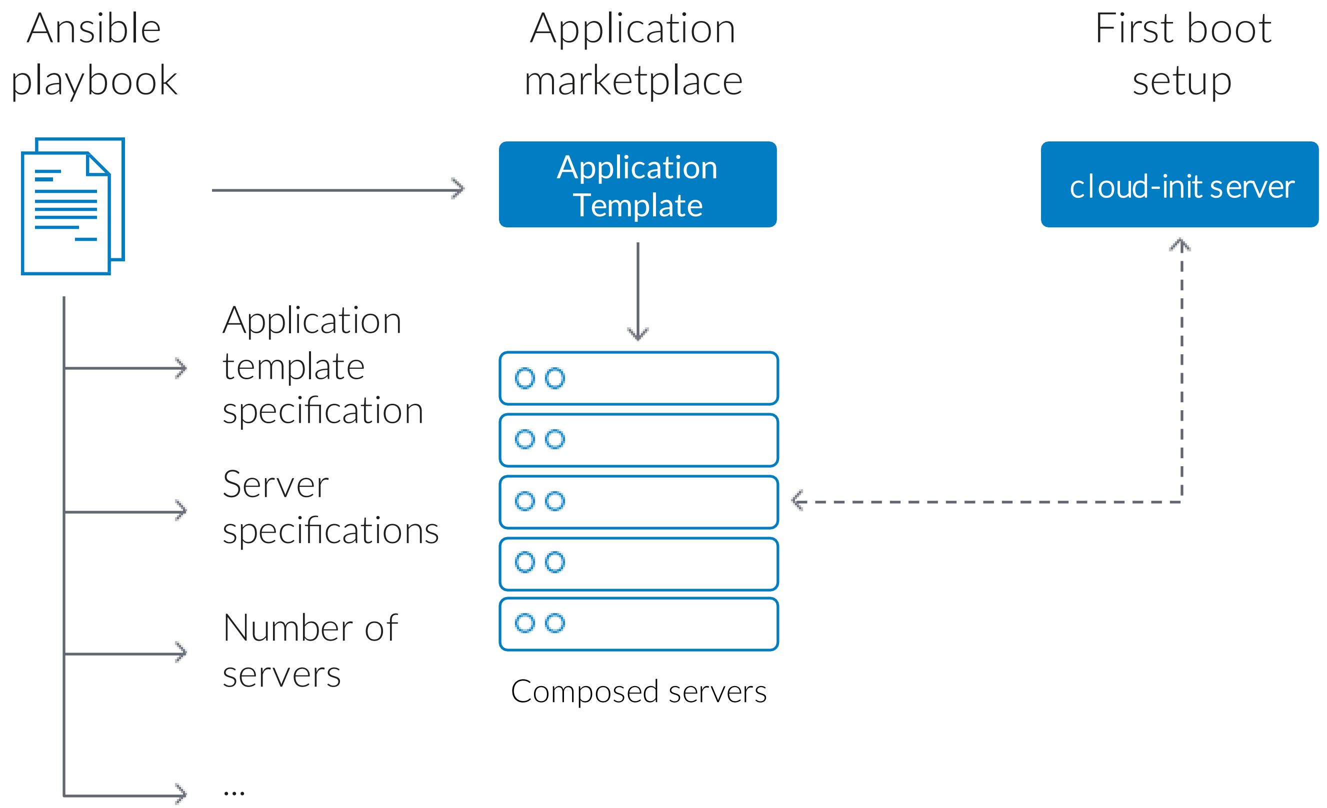

 

# Bringing Automation To Bare Metal

TerraForm/Ansible playbooks for deployment automation

Cloud-native first-boot configuration using Cloud-init

API-first approach




**MANAGE BARE-METAL INFRASTRUCTURE, LIKE YOU DO VMS**

**PLUG INTO EXISTING DEVOPS STACKS AND LEVERAGE EXISTING AUTOMATION**

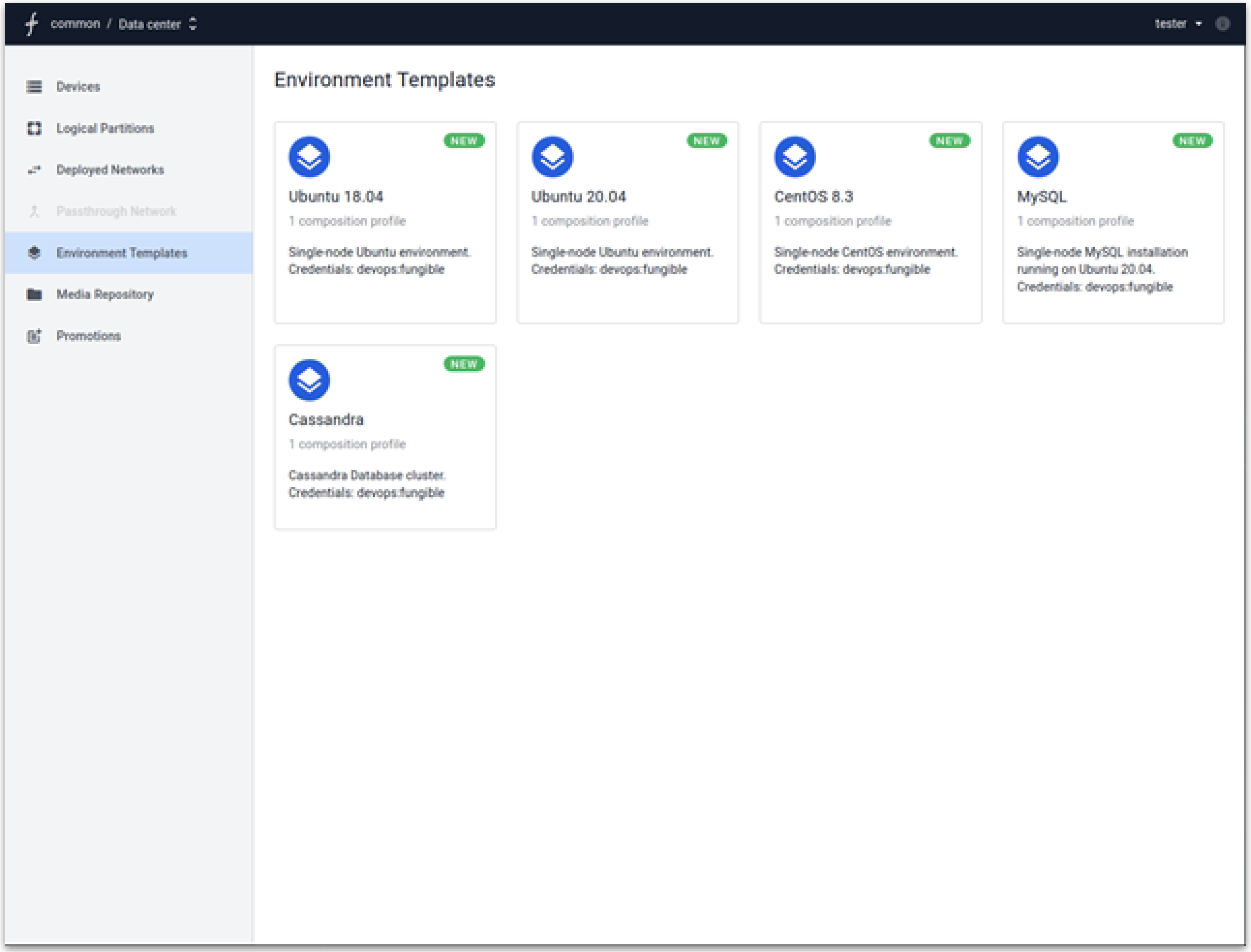
# One-Click Deployment Of Applications, Ease of Use

## Simplified App Deployment and Lifecycle Management,

Deploy complex workloads in minutes from pre-configured templates

Marketplace templates can be populated by service providers or their customers

- Each customer tenanted partition has its own marketplace visible just to themselves



# RECAP: BENEFITS OF FUNGIBLE DATA CENTERS

| | |
|---|---|
| PERFORMANCE | Market leading ***bare-metal performance*** for ***data-centric applications*** - enabled by fully offloading data-centric I/O processing to the Fungible DPU. Frees up cores for applications on host-side. |
| SECURITY | Independent ***hardware-accelerated security domains***, fine-grained segmentation, robust QoS, line rate encryption. |
| AGILITY | Reallocate compute, storage and network resources across workloads ***in minutes*** to handle workload hot spots. Adapt to changes in workloads to re-allocate high-demand items like GPUs. |
| SIMPLICITY | Deploy and manage ***turnkey multi-tenanted data centers*** with a single pane of glass management.<br>Deploy and manage complex scale-out workloads ***without any application changes***. |
| COST | Reduces server SKUs to a minimal set, gaining economies of scale and management simplicity.<br>Disaggregation and pooling of server, storage, network and GPU resources enables higher utilization (statistical multiplexing) under changing workload demands.<br>***Just-in-time composition*** of independent compute, storage, network and GPU resources optimizes consumption to exactly meet workload requirements with no wastage. |

# Thank You

Jason.vanvalkenburgh@fungible.com



STORAGE DEVELOPER CONFERENCE
SDC 21