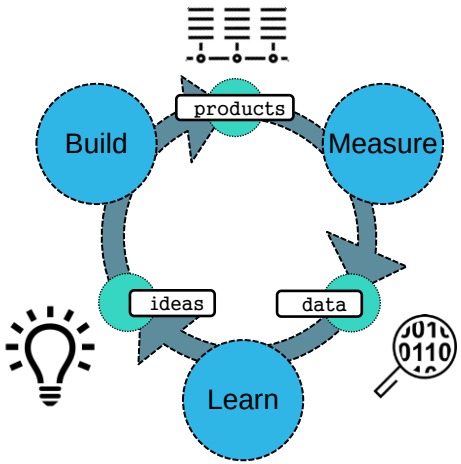
Software Defined Storage and Modern Cloud Platforms

Aaron Spiegel – Field CTO, Software Defined Storage

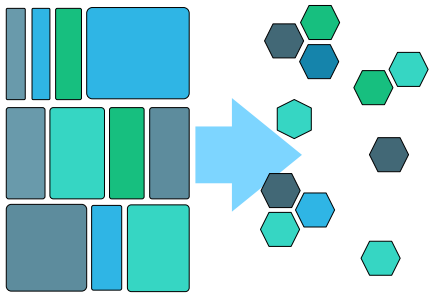
DELL EMC

Digital Transformation is Driving New IT Infrastructure

The Virtuous Cycle

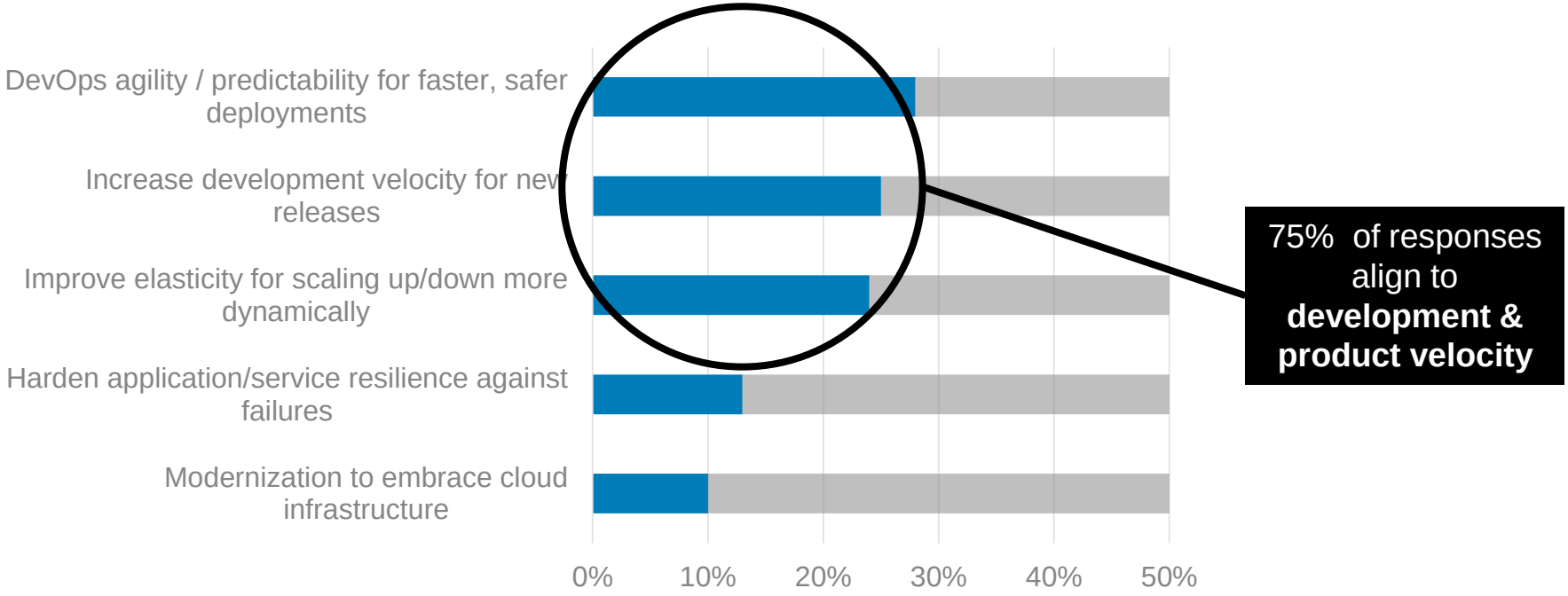


Micro-services



Modern Data Center

Which best describes the main reason WHY your organization is embracing microservices?



Source: *Enterprise Development Trends 2016*, Lightbend

Micro services are **not** a free lunch*

Frequent and more Varied Application Deployments

- Continuous Integration and Deployment
- Prefers PaaS & Schedulers
- Automated Infrastructure Becomes Vital
- Prefers Loose Coupling Between Host & Application

Distributed System Considerations

- Requires New Service Communication Patterns:
  - Service Registration
  - Circuit Breakers
  - Intelligent Load Balancing
- Eventually Consistent & Strongly Consistent Communications

Source: https://martinfowler.com/

Diverse Persistence Strategies

- Service Specific Databases
- Many Data Platform Architectures
  - RDBMS
  - Columnar
  - Document Oriented
- Service Specific Data SLAs:
  - Geo-Distributed
  - Performance
  - Data Sharing Requirements
  - Consistency

Docker's Initial Challenge

Static Website
nginx 1.5 + modsecurity + openssl + bootstrap 2

User DB
postgresql + pgv8 + v8

Queue
Redis + redis-sentinel

Analytics DB
hadoop + hive + thrift + OpenJDK

Background Workers
Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

Web frontend
Ruby + Rails + sass + Unicorn

API endpoint
Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

open container spec

Development VM

QA server

Contributor's laptop

Production Servers

Public Cloud

Service Provider

3rd Party Contributors

Source: *Why Docker?* Docker.io

Microservices also has an impact of Storage Requirements

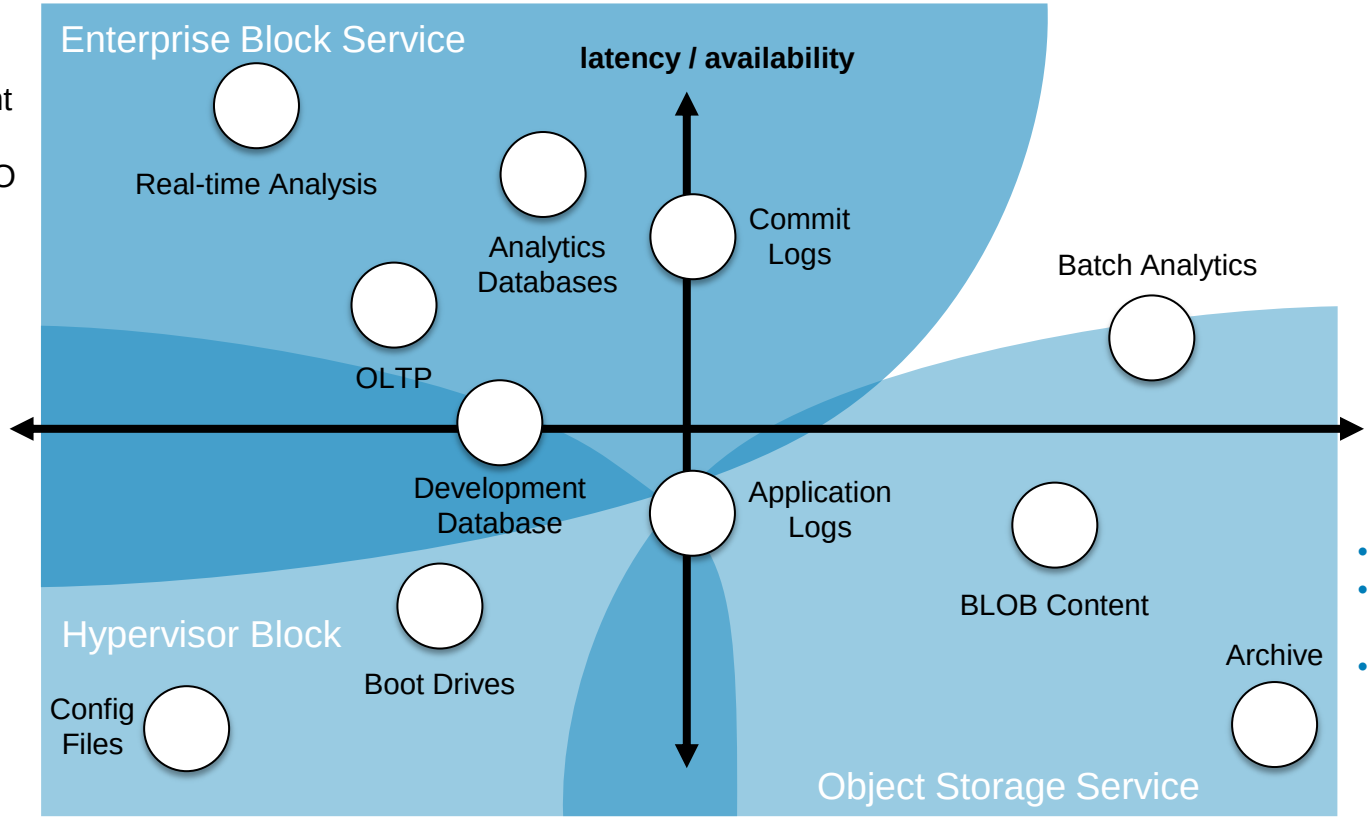
Creates momentum away from **Well-Rounded**, towards **Best-Fit** solutions

- Moderate Availability & Performance for App-Server, Boot etc.
- Low Latency, High Availability Block Storage for Database Platforms
- New Storage Formats for Unique Requirements:
  - Hyper-scale & Geo-Distributed Data Sets
  - Massively parallel workloads for analysis and writes
  - Full-text search platforms
  - Unique document, graph and event orientations with horizontal partitioning
  - Fine-grained memory allocations for K/V, session stores and atomic lists
  - Streaming Analytics for Real-Time Analytics and Next Generation Apps

Platform administrators challenged to *keep up* with developers' choices

Application Storage Requirements are Diverse

- AWS EBS
- Google Persistent Volume
- Dell EMC ScaleIO

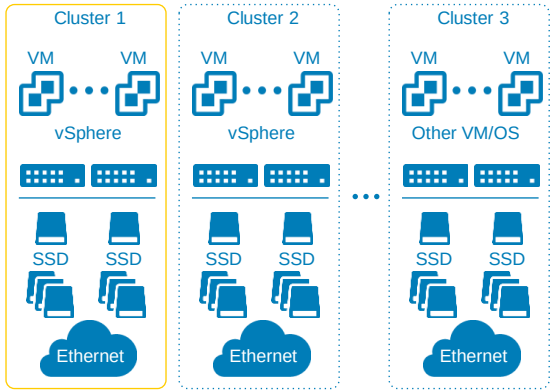


- AWS EC2 Instance Storage
- GCE Instance Storage
- Local Hypervisor Storage

- AWS S3
- Google Cloud Storage
- Dell EMC ECS

Cloud Environment Storage Primitives

Hypervisor Centric Block

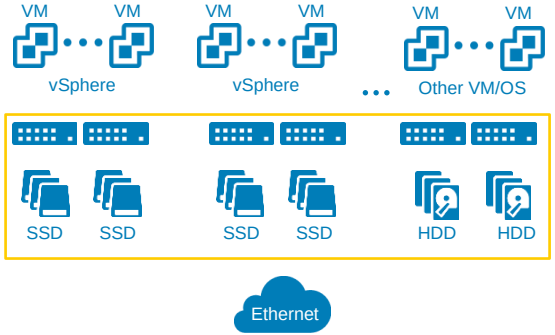


Tightly Coupled to Hypervisors

Optimized for Cost and Simplicity

Less Stringent SLAs

Enterprise Block Service



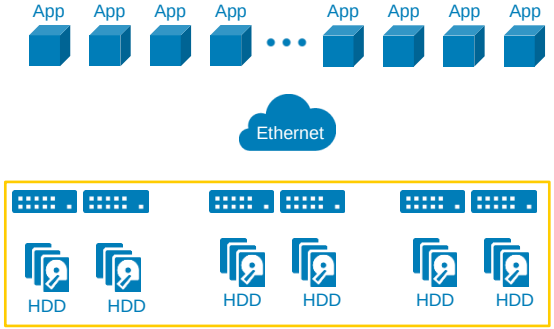
Loosely Coupled to Hypervisor

High Throughput / Low Latency

Most Stringent SLAs

Requires Flexibility to outlast single hypervisor cluster

Object Storage Service



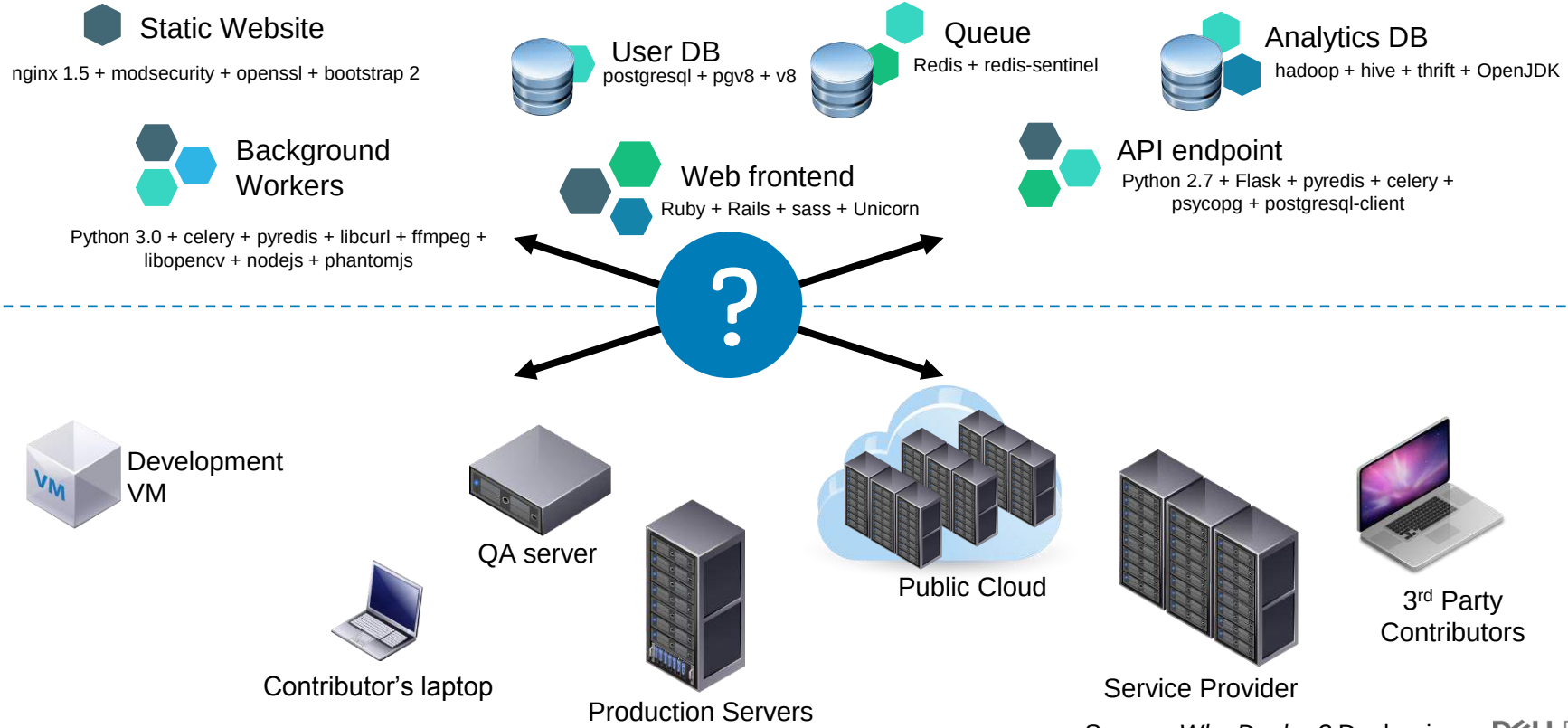
Loosely Coupled to Applications

Optimized for Price

Geo-Distribution Requirements

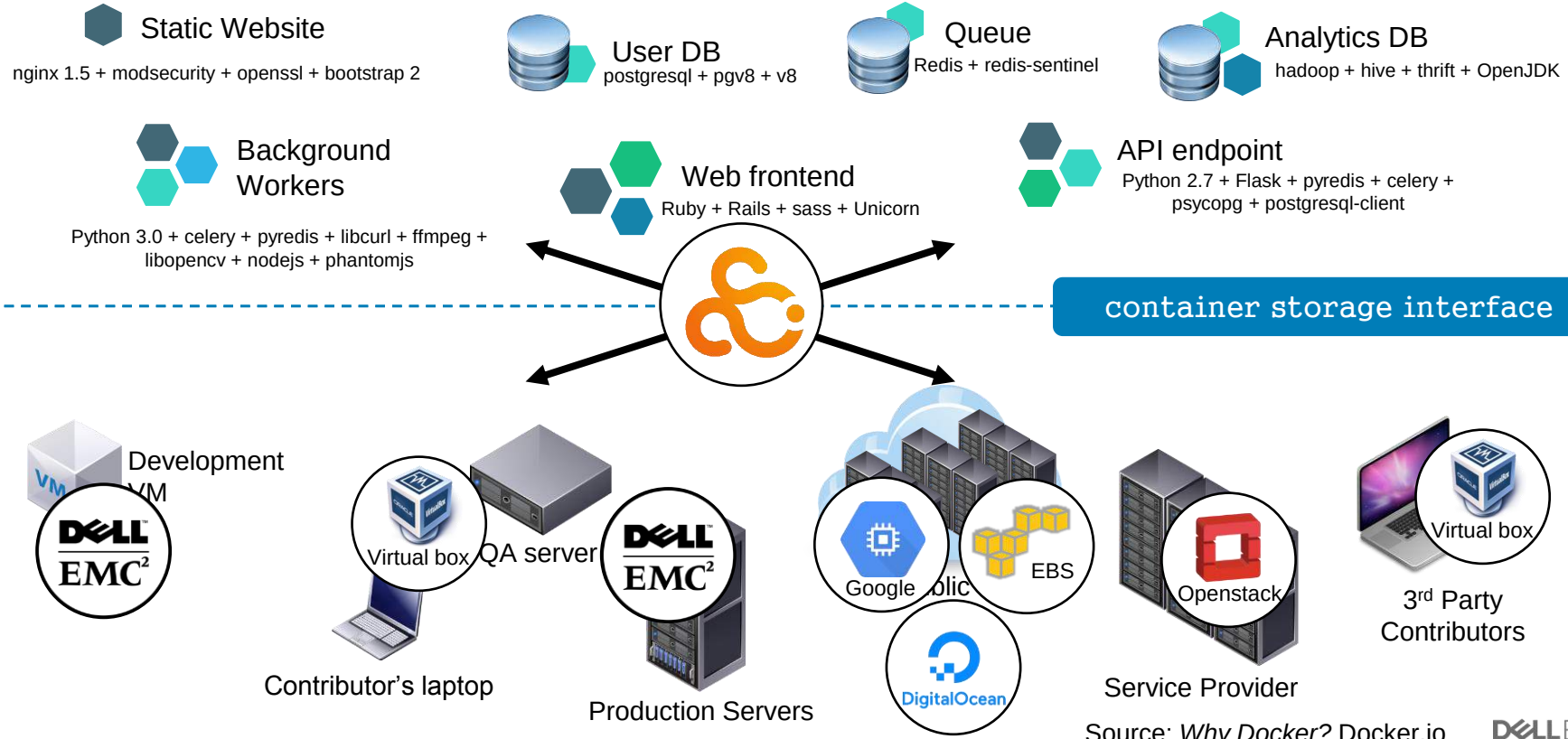
Support for Variety of App Protocols

Persistent Storage for Containers Equivalent



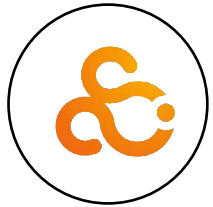
Source: *Why Docker?* Docker.io

Container Storage Interface as a Next Step to Support Data Platforms & State-ful Apps



Source: *Why Docker?* Docker.io

Container Storage Interface Specification



DOCKER

Use the stand alone Docker Engine to run a stateful application or combine it with Docker Swarm Mode to turn your application into a robust service.

APACHE MESOS

Use any framework that orchestrates containers such as Marathon or Aurora to provide persistent storage for stateful services.

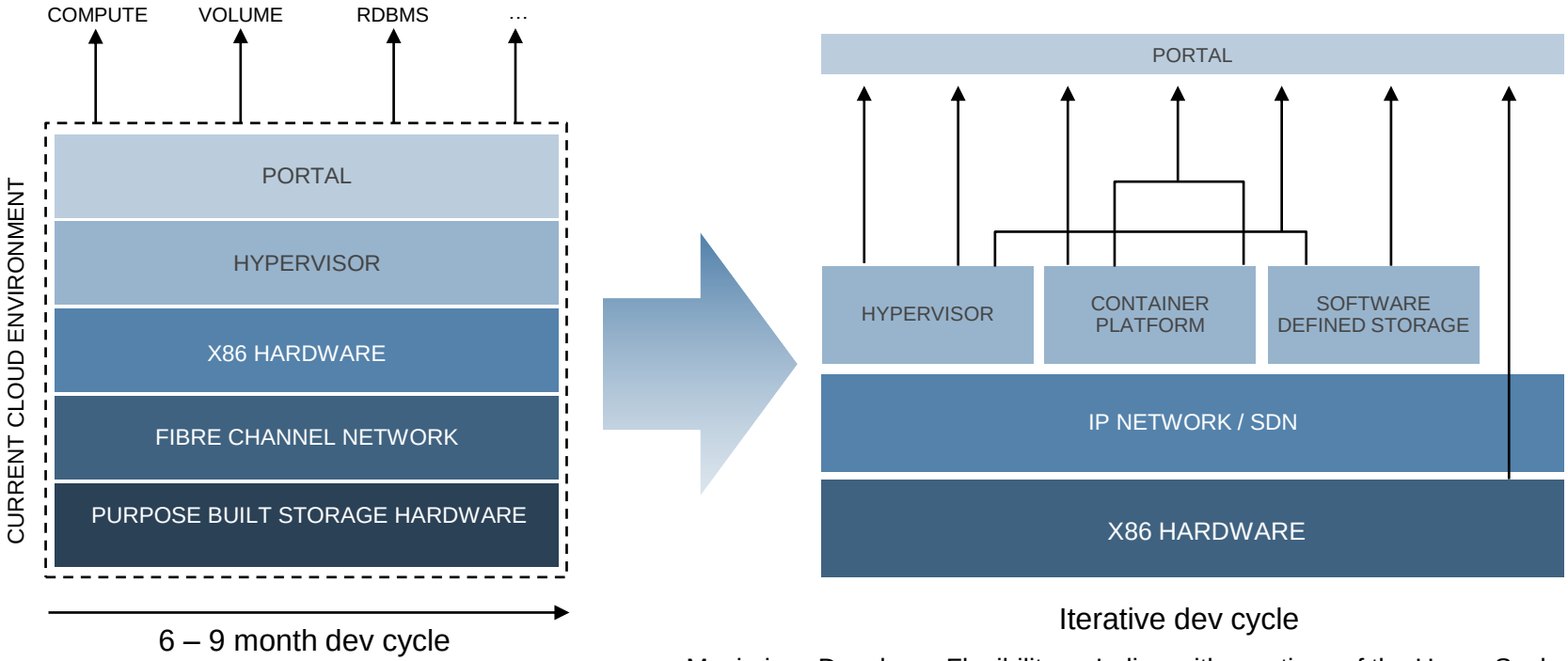
KUBERNETES AND OPENSHIFT

Run stateful applications in pods and stateful sets through FlexREX, and benefit from a broader set of storage platforms and CLI management capabilities.

PIVOTALCLOUD FOUNDRY AND PKS

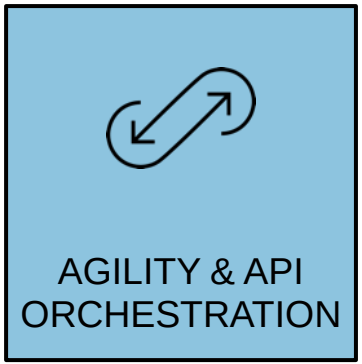
Run stateful applications using Pivotal Persistent Volume Support, or with the newly announced Pivotal Container Services (PKS).

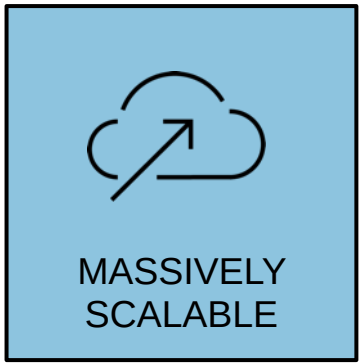
Transitioning from Template Centric to Portfolio of Services

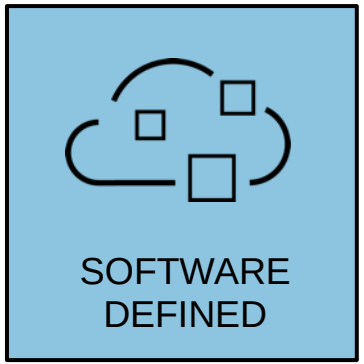


Maximizes Developer Flexibility · In-line with practices of the Hyper-Scale Public Cloud

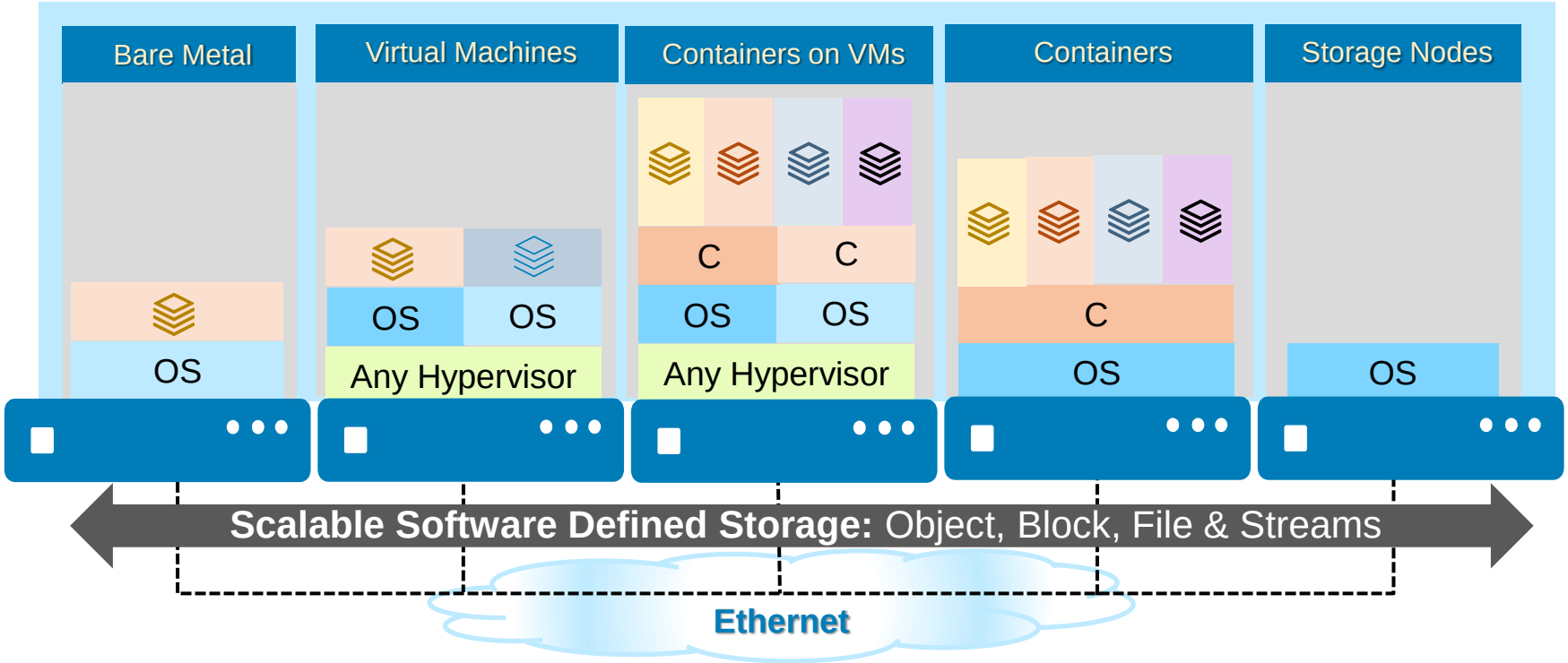
Characteristics of an Storage as a Service Offerings Differ From Traditional Storage



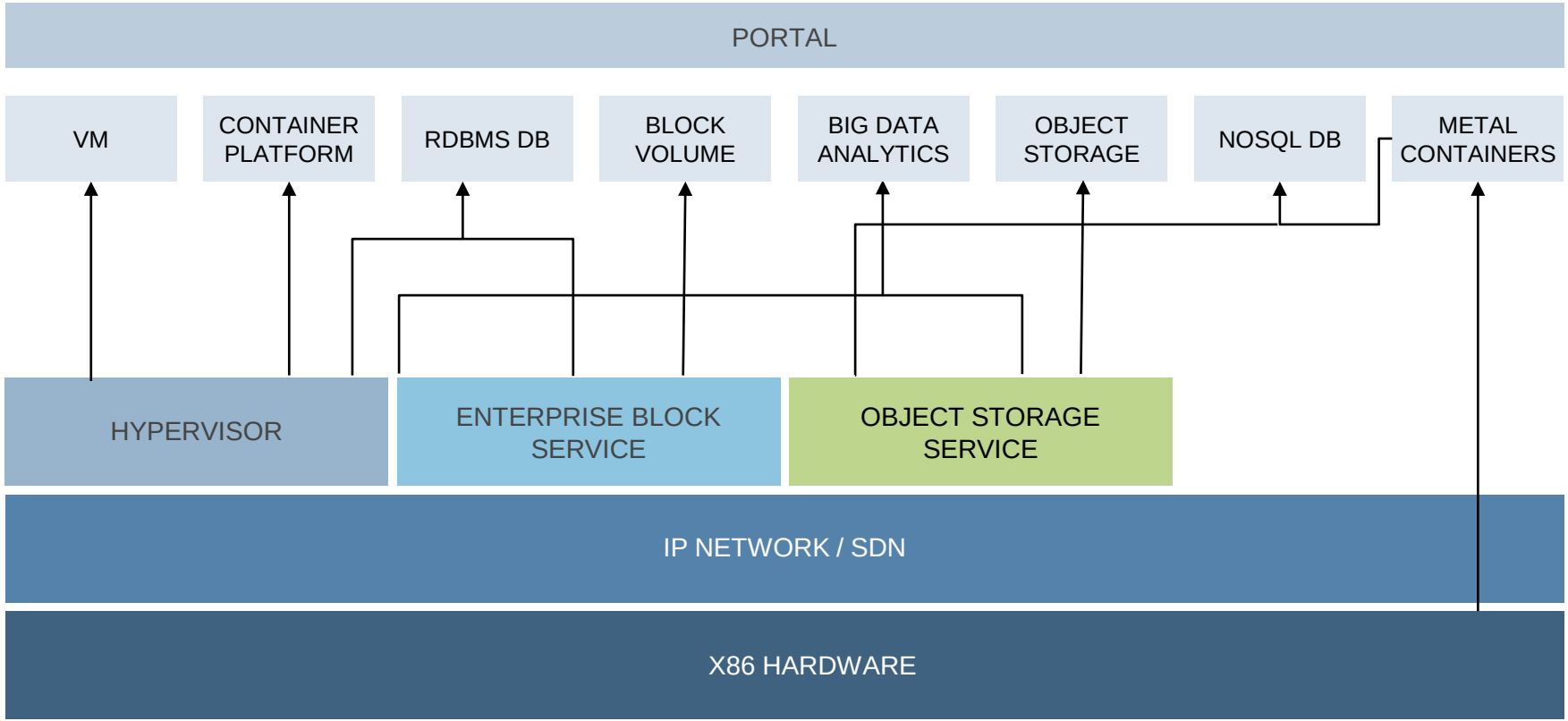




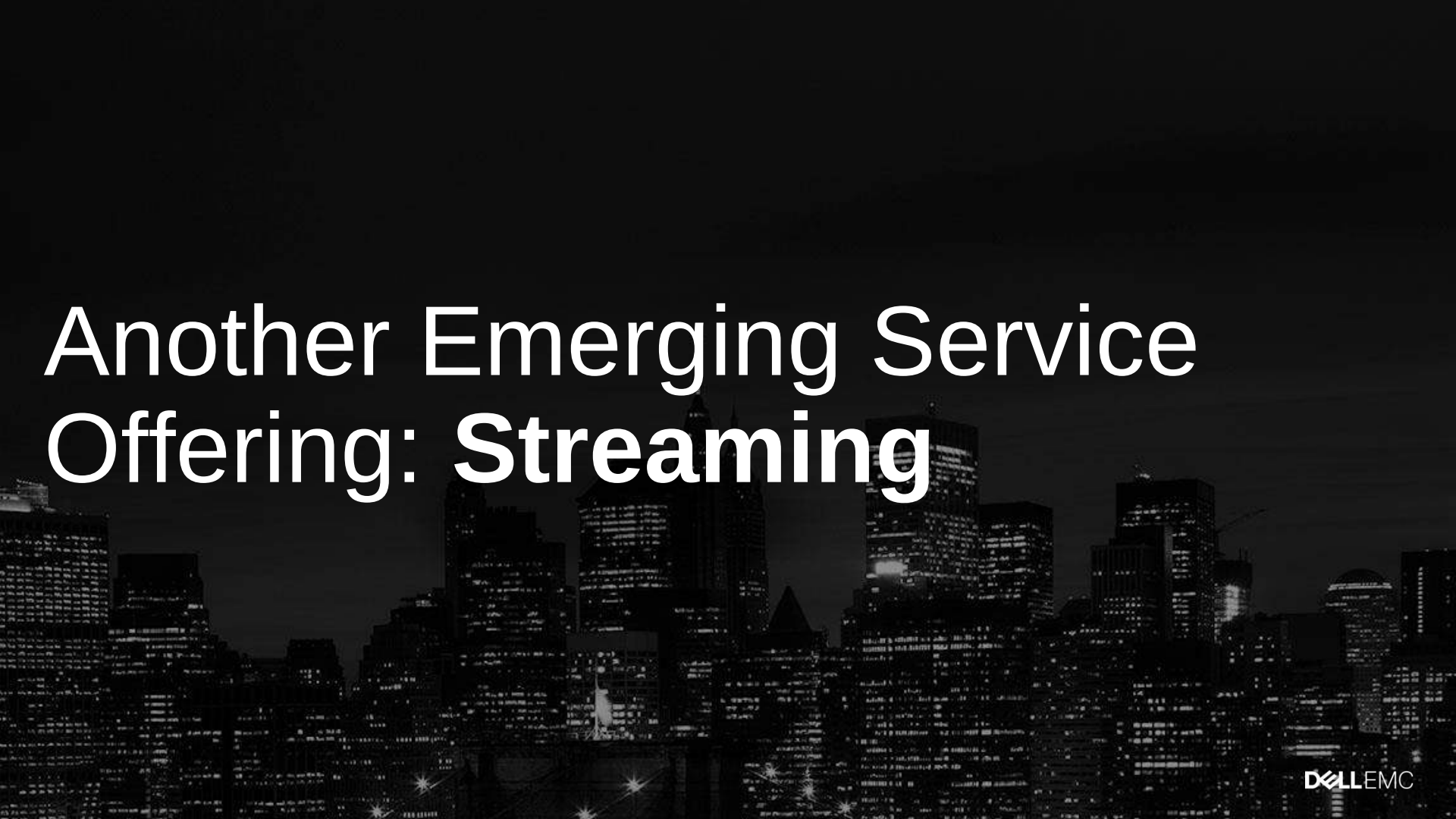
Works Flexibly Across Physical, Virtual and Container Environments



A Service Portfolio Approach to Cloud



Another Emerging Service Offering: **Streaming**



IN THE BEGINNING THERE WAS BATCH…

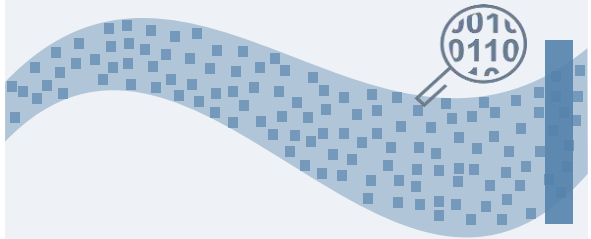
The Future is Based on Real-Time & Streams

Batch Processing

- Data trickles in over time
- Once data collection is ***complete***, processing can begin
- Processing post-collection adds **delays**
- Collection & processing phases create performance **peaks/valleys**

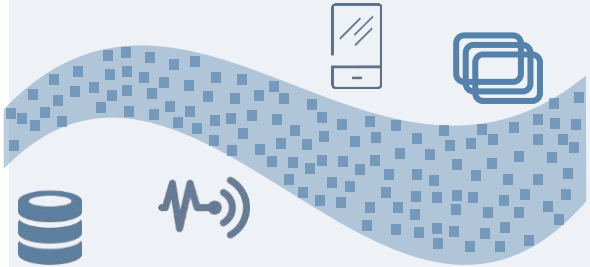
Real Time Processing

- Data is analyzed as it is received
- Intermediate Results in ***seconds***
- Data is understood to be ***continuous***
- More ***consistent*** resource usage



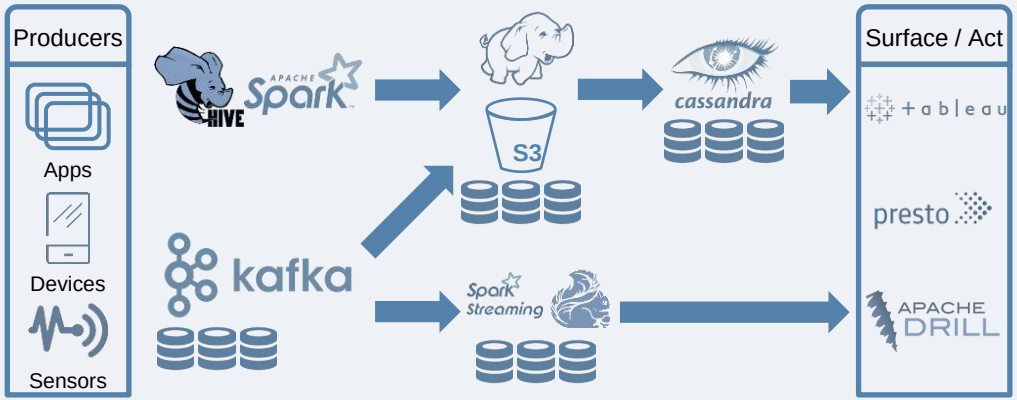
Stream-Based Applications

- Support **IoT** Applications
- New micro-service app architectures such as **CQRS**
- Flexible data platforms: caches, search-engines, DBs



CURRENT REAL-TIME ARCHITECTURES ARE VERY COMPLEX & WASTEFUL

Predominant Lambda Architecture



Challenges for the Infrastructure Admin

- Many Technologies to Setup, aintain & Scale
- Duplication of Data Ingest & Storage
- High Operational Overhead
- Disparate Security Models
- Individual DR for each Technology
- Complex Testing and Deployment Workflows

Challenges for the Data Scientist

- Duplicate Models & Code for Processing Across Batch & Real-Time Processing
- Data Silos between Stream & Long-term Storage
- Requires ETL & Movement Throughout Data Lifecycle

Not Suitable as a Cloud Service

- Inconsistencies During Upgrades & Outages
- Scaling Requires Client Application Changes

A SOLUTION IN TWO PARTS

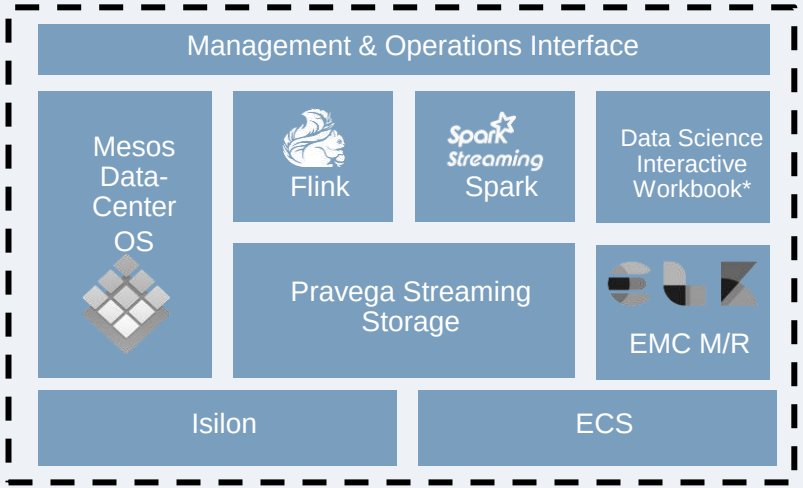
Pravega Streams

An Open Source Streaming Storage Platform. Designed from the ground up as a storage platform for streams.

- Unlimited Retention for Real-Time, Interactive & Batch
- Storage & Performance Efficient
- Transactions & Exactly-Once Guarantees
- Multi-Tier Policy-Based Storage
- Auto-Scaling with Dynamic Partitions

Nautilus

A Unified Platform for Big & Fast Data, built on streams. Available in Appliance or Software Delivery Models



* Planned for Post-GA release

A UNIFIED ARCHITECTURE FOR ANALYTICS

Strongly Consistent Storage ➔ Exactly Once Processing ➔ Unified Analytics

Unified Platform for Batch & Real-Time

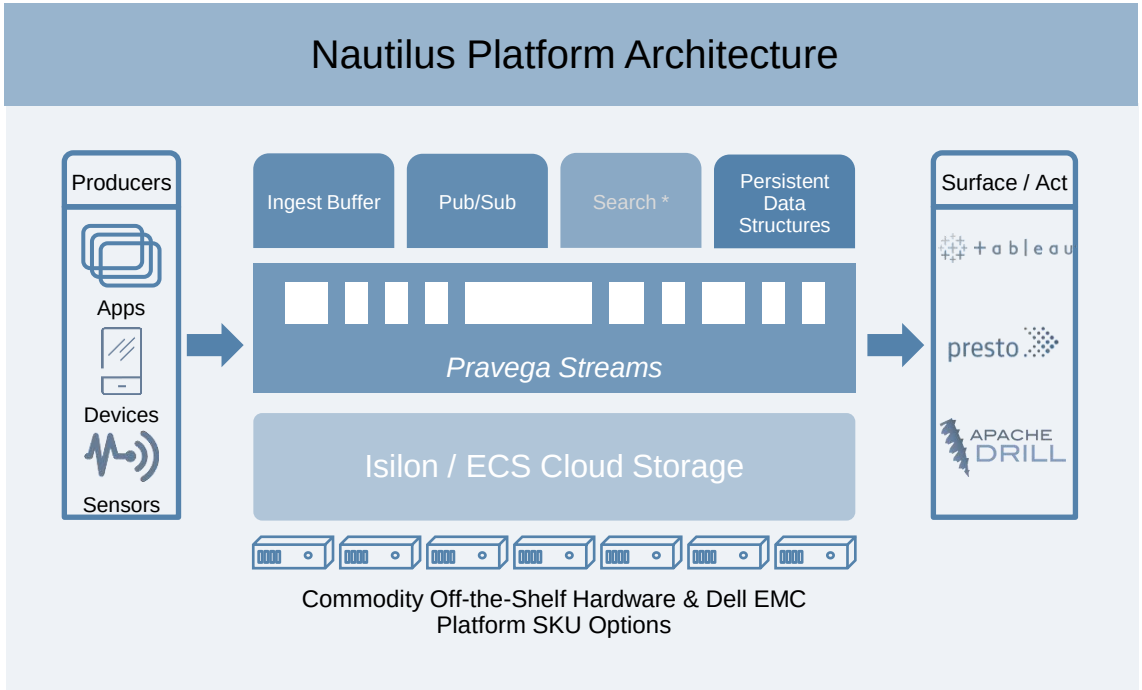
- Less Data Science & Development Effort
- Remove Data Silos and Duplication
- Less Data Staging and Movement
- Massively Improved Operations

Cloud Service Capable Platform

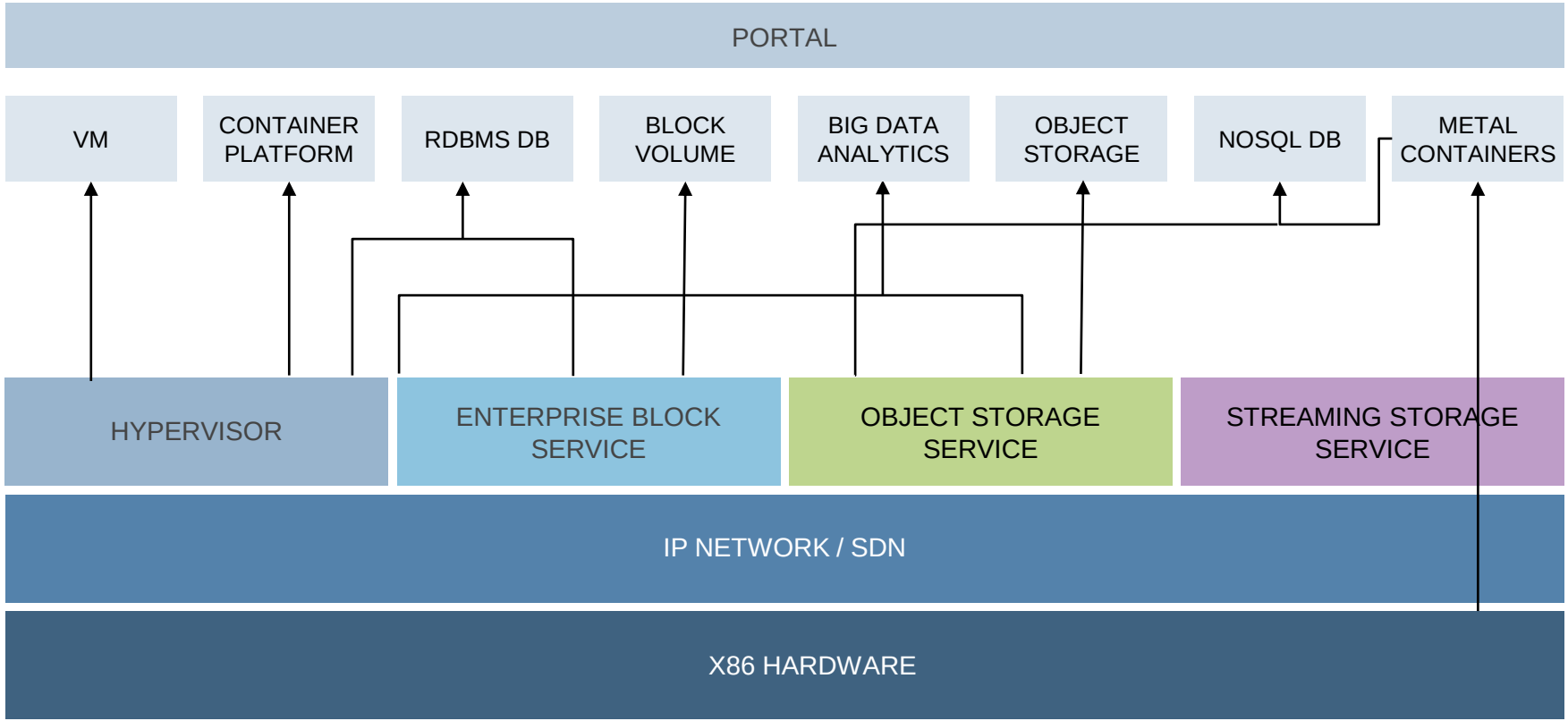
- Strong Data Consistency & Exactly-Once Delivery
- Atomic Transaction Support
- Dynamic Auto-Scaling Up & Down
- Automatic DR with Geo-Distributed Cloud Storage

Optimized for Performance & Scale

- High-Performance Durable Writes
- Optimized Storage for Tail & Catch-Up Reads
- Policy-based Multi-Tier Storage across SSD/NVMe & Cloud Storage Tiers
- Support for Infinite Retention or Policy-based Retention

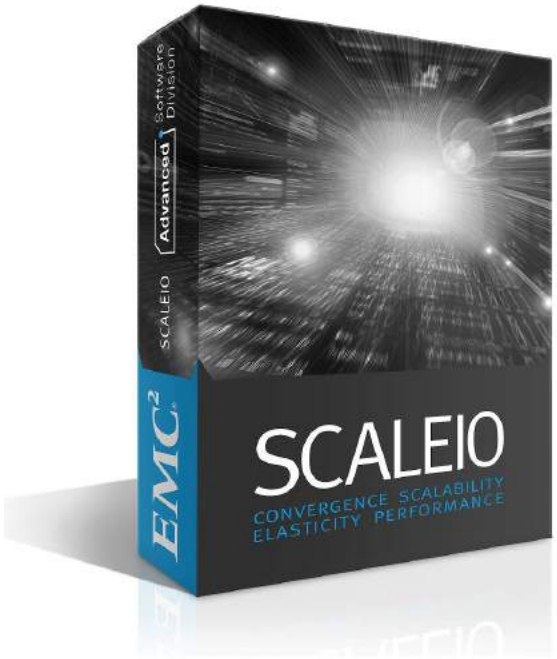


Streaming as an Enterprise Service




DELL EMC

ScaleIO – Software for an Enterprise Block Service



Enterprise Storage Platform based on Commodity X86 Hardware & IP Networking

Designed for Large Scale Heterogeneous Infrastructure as a Service

Enterprise Grade Performance, Reliability & Data Services

TECHNOLOGY EVOLVED TO SUPPORT TRANSFORMATION

Standard x86 building blocks vs proprietary hardware
CPU less expensive
Higher capacity
Flash drives

Network Ethernet vs FC
Bandwidth increasing
More affordable
More performant

Scale-out vs. scale-up
Just in time / just enough resources with no bottlenecks

Modern Data Center

Resilient, scalable and easy to deploy infrastructure platforms

Software-defined Storage

Storage-Only / Hyper-converged

DELL EMC

WHAT IS DELL EMC ELASTIC CLOUD STORAGE?

A MODERN, ENTERPRISE READY, SOFTWARE-DEFINED, S3-COMPATIBLE OBJECT STORAGE STORAGE SERVICE

Exabyte Scale

- No Logical or Physical Limitations
- Enterprise Management Built for Scale
- Capacity & WAN Efficient Data Protection
- Supports Large & Small Files Equally
- Interoperable Software & Appliance Options

Efficient

Globally Accessible

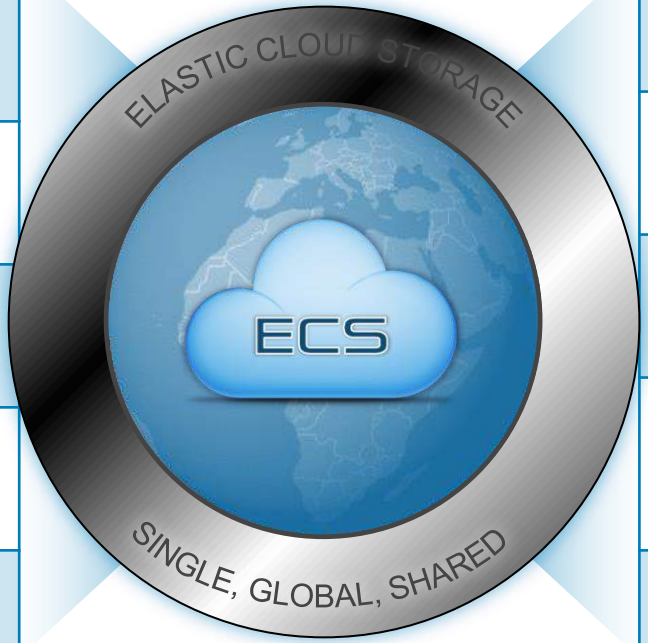
- Supports Swift, S3, NFS & more
- Strongly Consistent Multi-Region Replication
- Support 1 to 8 Regions
- Spans Private & Public Cloud Options

Flexible

Elastic Cloud Storage

A modern storage platform to bridge traditional and modern workloads.

✓	H/W Vendor Agnostic
✓	Exabyte Scale
✓	Multi-protocol
✓	Low overhead
✓	Multi-tenant



✓	WAN-efficient
✓	Active-active
✓	Geo-accessible
✓	Strong Consistency
✓	Large/Small Files

Docker's Initial Challenge

Static Website
nginx 1.5 + modsecurity + openssl + bootstrap 2

User DB
postgresql + pgv8 + v8

Queue
Redis + redis-sentinel

Analytics DB
hadoop + hive + thrift + OpenJDK

Background Workers
Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs

Web frontend
Ruby + Rails + sass + Unicorn

API endpoint
Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client

Development VM

QA server

Contributor's laptop

Production Servers

Public Cloud

Service Provider

3rd Party Contributors

Source: *Why Docker?* Docker.io

Docker's Initial Challenge

Resulting in the NxN Compatibility Nightmare

	Development VM	Contributor's Laptop	QA Server	Production Server	Public Cloud	Service Provider	3rd Party Contributor
Static Website	?	?	?	?	?	?	?
User DB	?	?	?	?	?	?	?
Queue	?	?	?	?	?	?	?
Background Workers	?	?	?	?	?	?	?
Web Frontend	?	?	?	?	?	?	?
Analytics DB	?	?	?	?	?	?	?
API Endpoint	?	?	?	?	?	?	?

Source: *Why Docker?* Docker.io