



**Hewlett Packard  
Enterprise**

# Overview of HPE AI Strategy and Platform

Gaurav Kaul and Sorin-Cristin Cheran

EPCC Vendor Day



# Winning with AI now



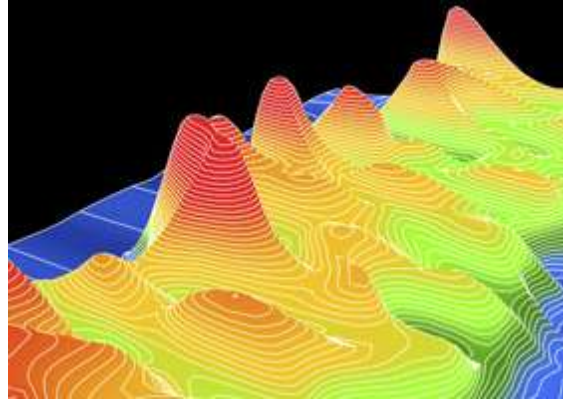
## Financial services

Fraud detection, ID verification



## Government

Cyber-security, smart cities and utilities



## Energy

Seismic and reservoir modeling



## Retail

Video surveillance, shopping patterns



## Health

Personalized medicine, image analytics



## Consumer tech

Chatbots



## Service providers

Media delivery



## Manufacturing

Predictive and prescriptive maintenance

# HPE puts AI in Action



Shaving milliseconds off race times



Transforming manufacturing quality assurance using deep learning from edge to cloud



Automating worker safety and facility condition monitoring with predictive maintenance and video analytics

# Accelerate AI adoption

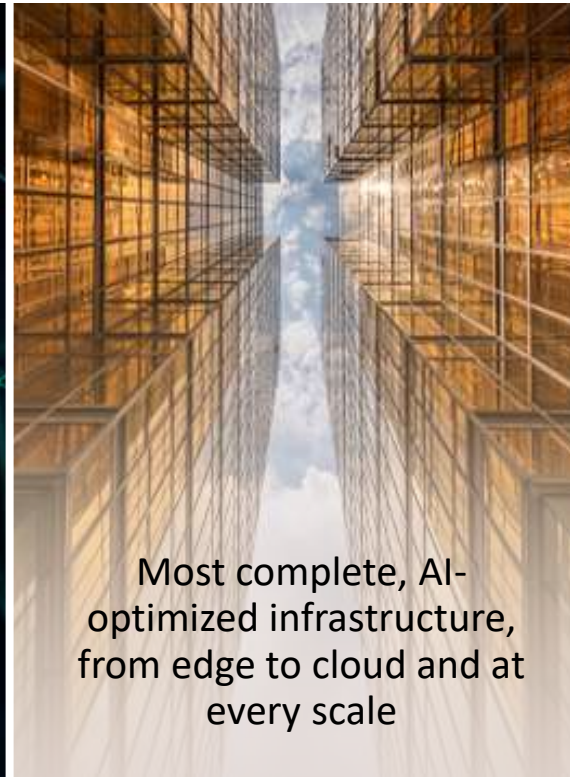
Speed the design and deployment of your AI strategy



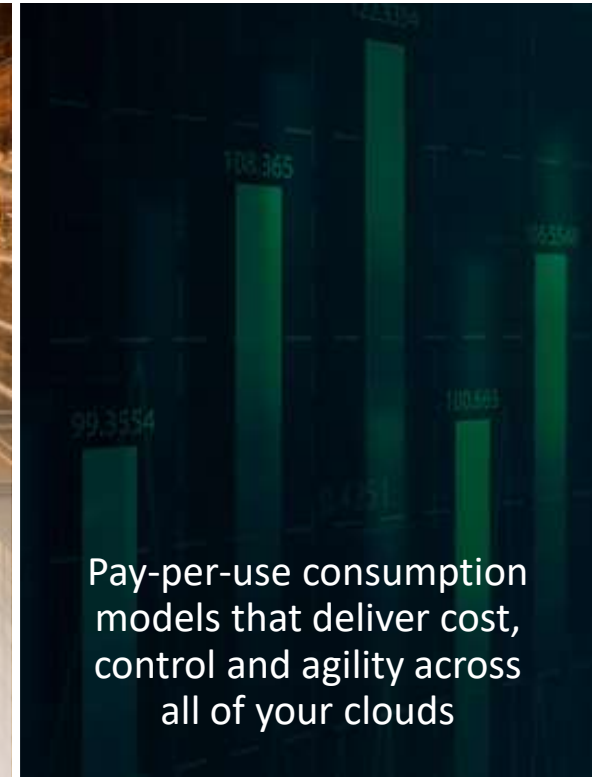
Give your data science teams instant access to AI tools and data




Build your AI models at scale with less complexity



Execute your AI strategy fast, cost-effectively, with less risk



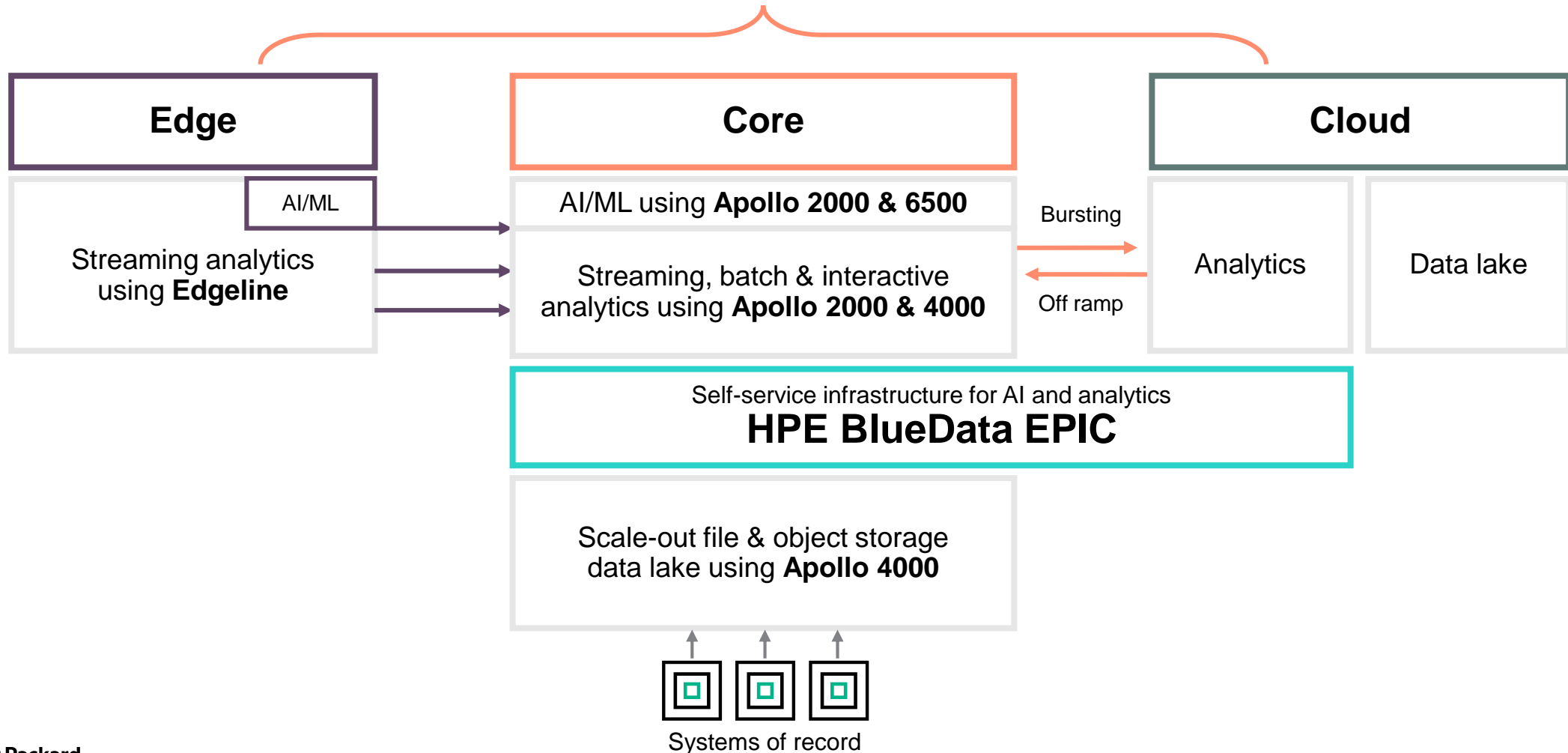
# HPE AI Vision



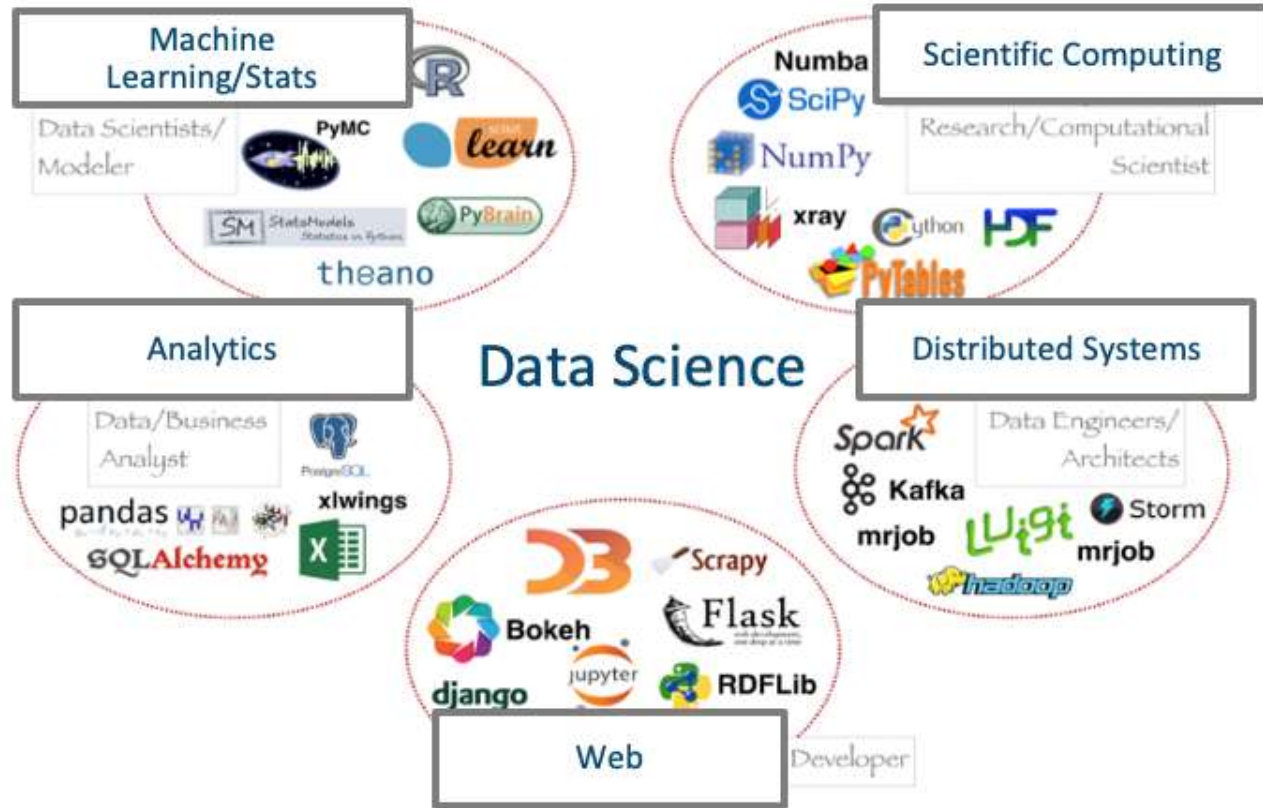
**Transform** the deployment  
and consumption experience  
for **AI and analytics**  
whether **on-premises** or in a **hybrid cloud**

# BlueData role within HPE portfolio

Enables **HPE POINTNEXT**



# Operationalizing ML models is hard

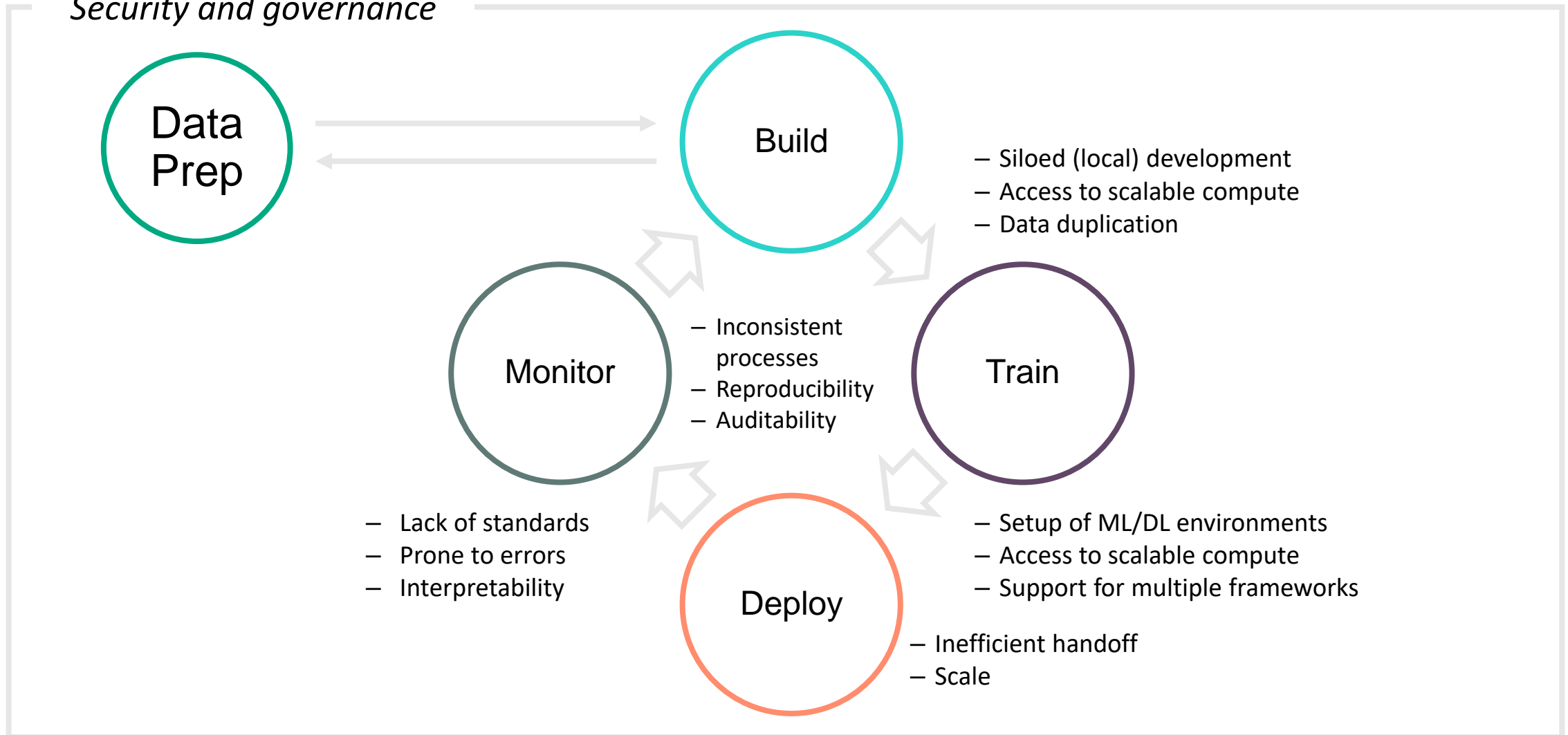


- A zoo of tools and frameworks for data processing, model development, model training, and visualization
- Not enough effort in standards and processes for operationalizing models
- Multiple barriers to adoption at scale

**Only operational ML pipelines deliver business value**

# Challenges with ML in the enterprise

*Security and governance*



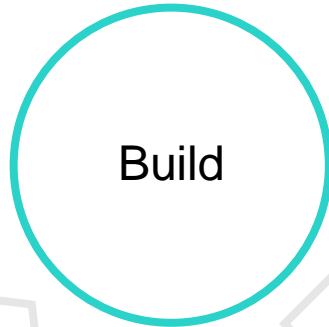
# How does HPE ML Ops address these challenges



*Enterprise grade security and control*



Data  
Prep



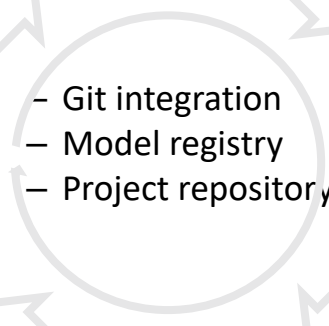
Build

- Containerized sandbox environments
- Choice of ML/DL tools, interfaces and frameworks
- Secure access to shared data

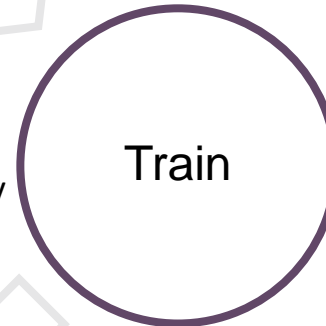


Monitor

- End-to-end visibility
- 3rd party integrations

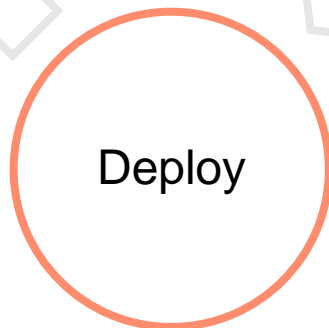


- Git integration
- Model registry
- Project repository



Train

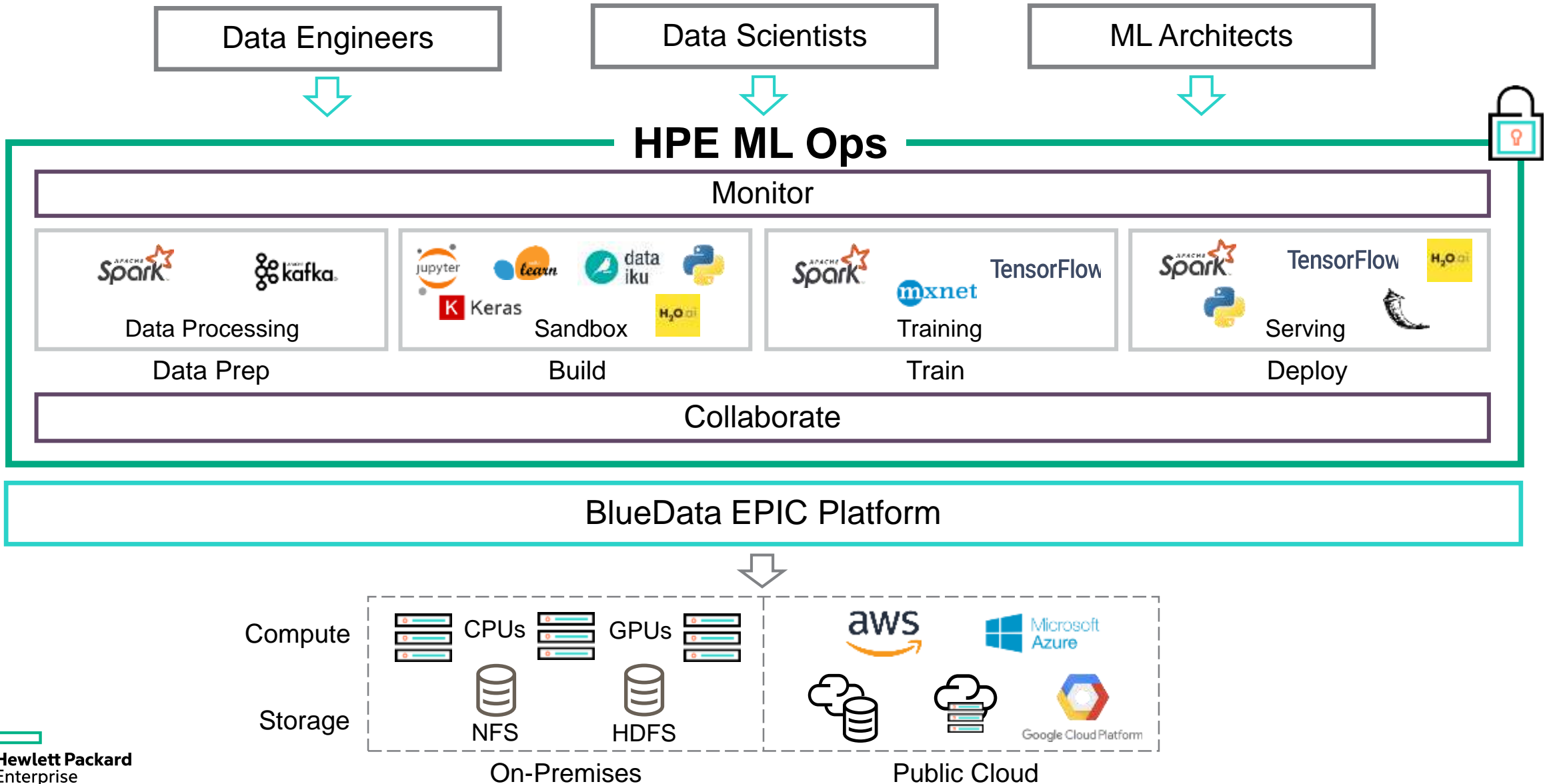
- Containerized ML/DL environments
- Elastic clusters
- Python, Spark, TensorFlow, ....



Deploy

- Containerized deployment with scalable endpoints
- Out of the box runtime engines
- Autoscaling and load balancing


# HPE ML Ops Architecture



# Future Proofed Big Data Architecture

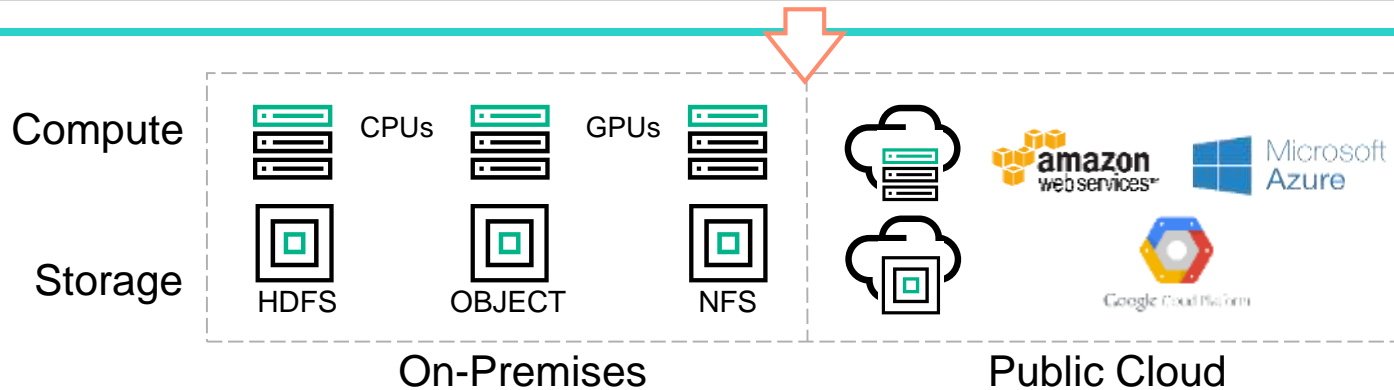
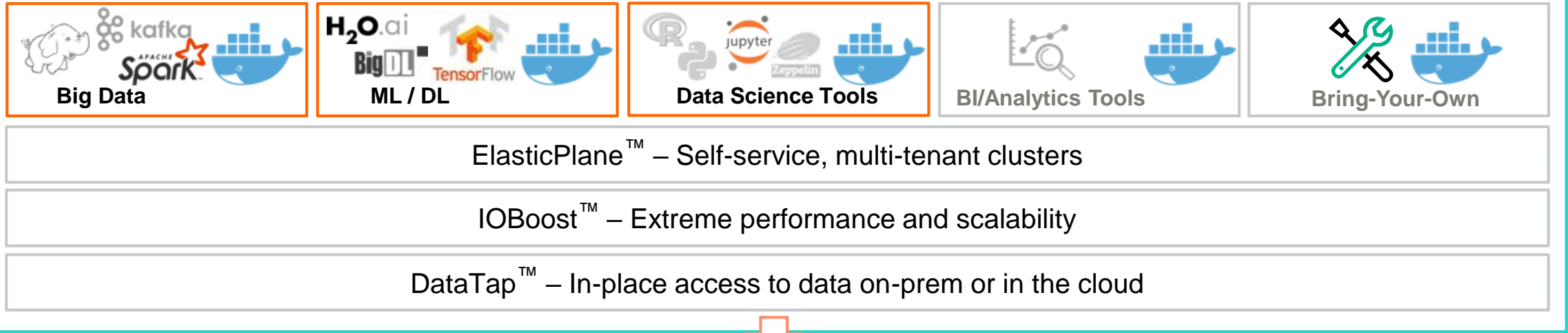
  
Data Scientists

  
Developers

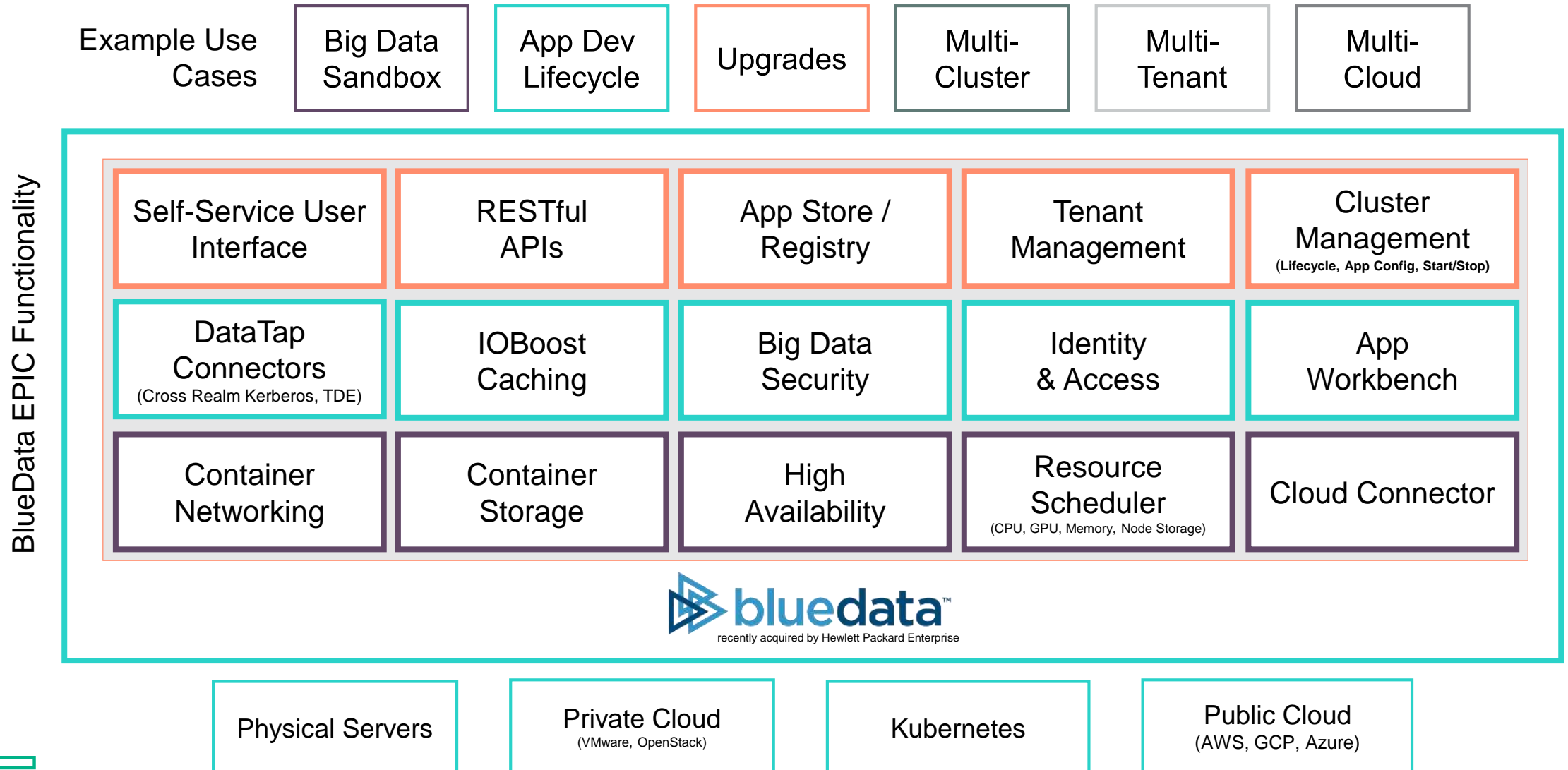
  
Data Engineers

  
Data Analysts

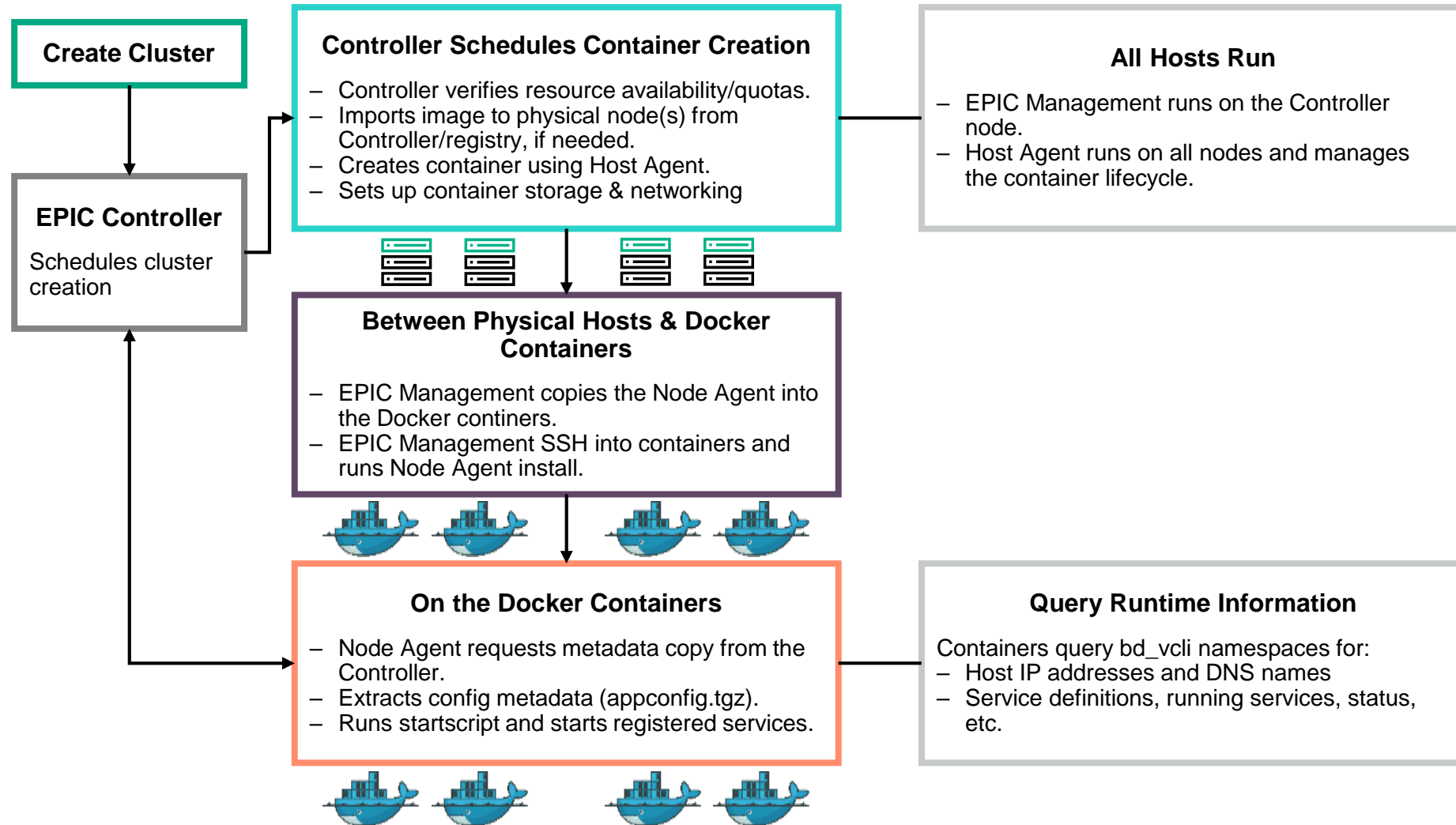
## BlueData EPIC™ Software Platform



# Key Software Components (Turnkey)

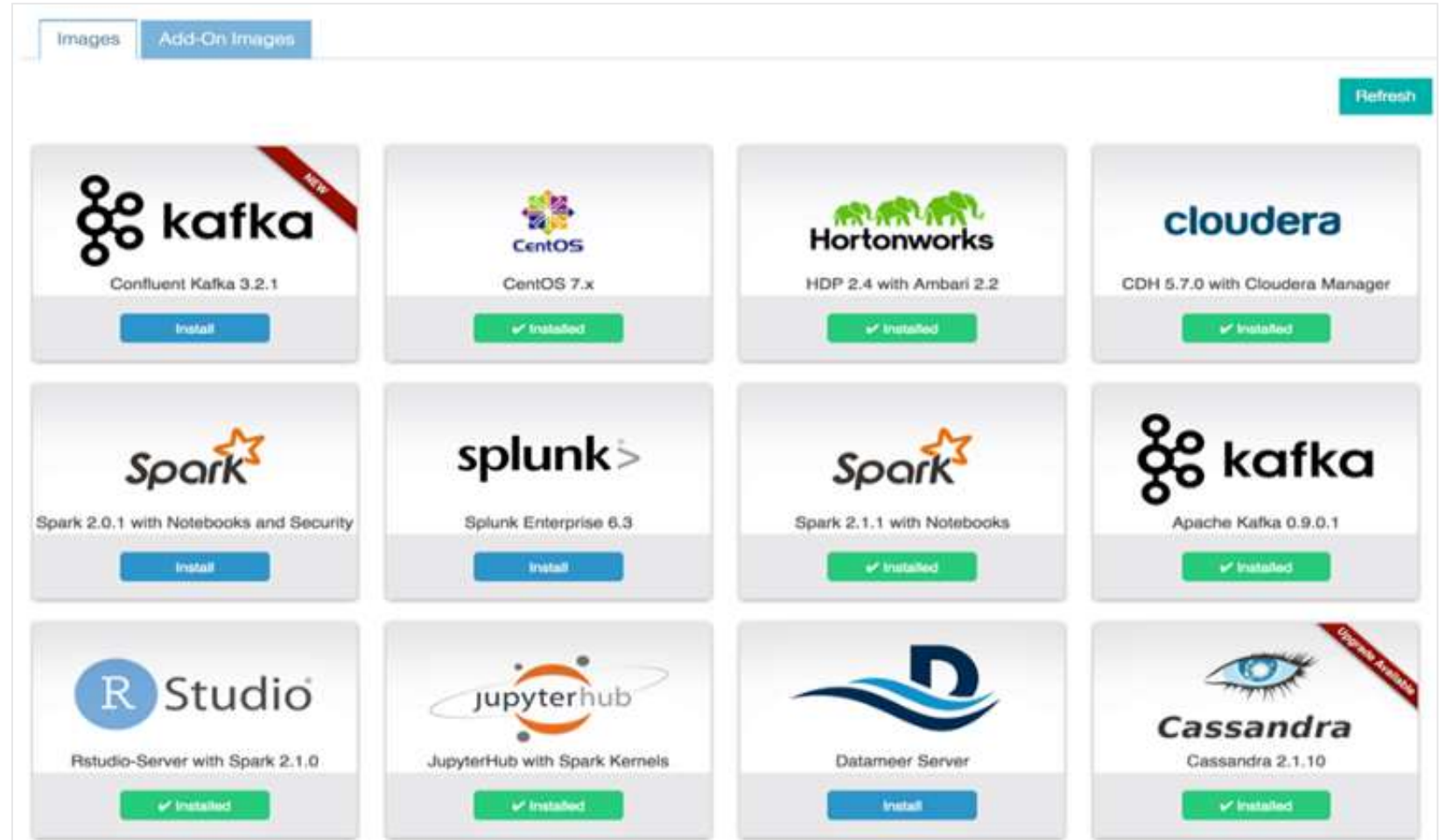


# Big Data Cluster Management (Containers)



# App Store

- The App Store in BlueData EPIC includes a variety of pre-configured Docker-based application images
- BlueData provides an App Workbench to configure custom images and modify or upgrade existing application images



# HPE "PROJECT ARTEMIS" CONTAINER PLATFORM

Data Engineers

ML Architects

App Developers

DevOps

## HPE "PROJECT ARTEMIS" CONTAINER PLATFORM

Spark CLUSTERA MAPR kafka  
Big Data Analytics

TensorFlow MxNet H2O R Studio  
AI / ML and Data Science

Java .NET php  
App Modernization

Jenkins GitHub Bitbucket  
CI / CD Pipelines

Bring Your Own / More


Self-service, multi-tenant containerized applications & services  
(BlueData + MapR, with Kubernetes orchestration engine)



In-place access to data on-prem or in the cloud  
(BlueData DataTap)

Multi-protocol, high performance persistent data services  
(MapR Data Fabric)

On-Premises

Compute   
Storage 

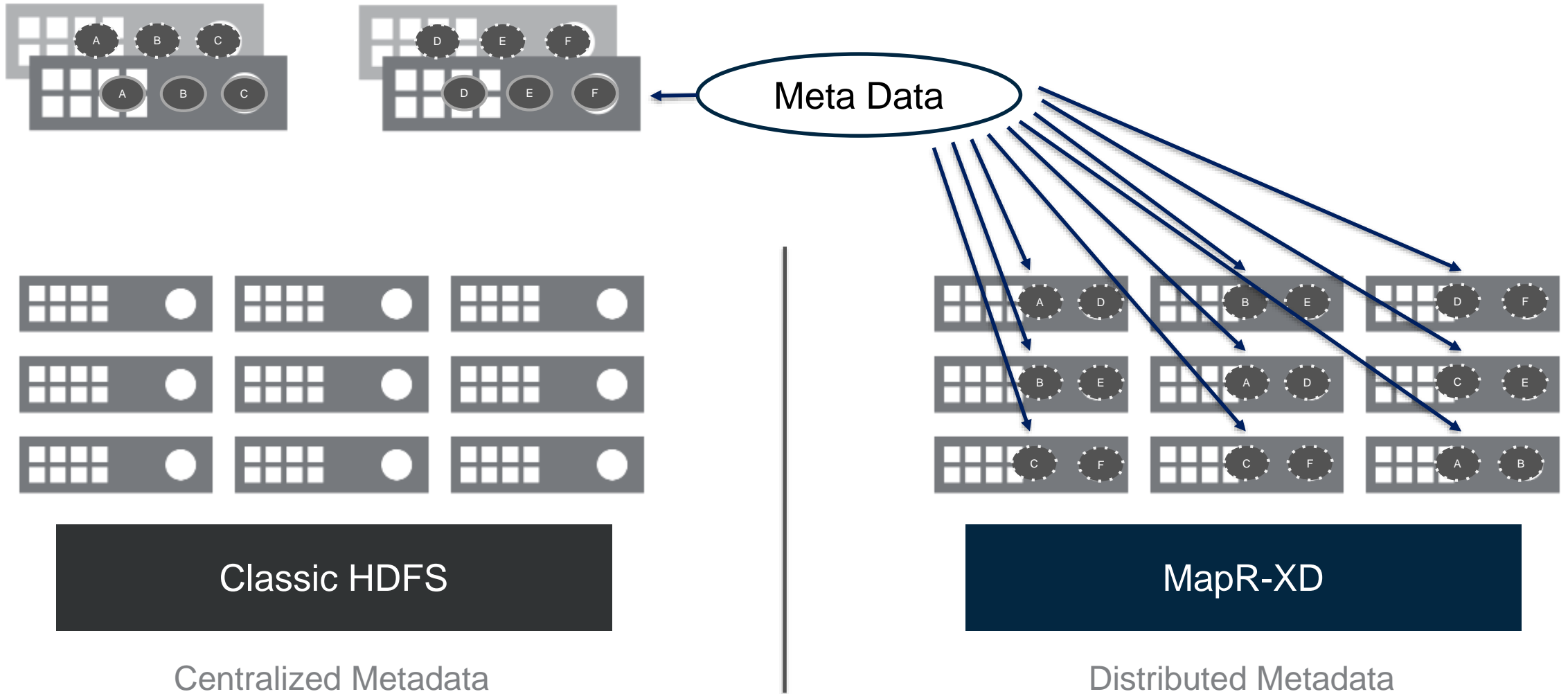
aws

Azure

Google Cloud Platform

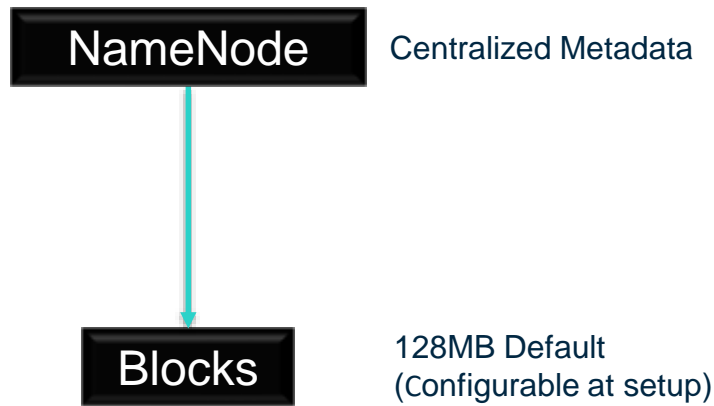
Public Cloud

# Summary: HDFS vs. MapR's "No NameNode Architecture"™



# MapR is Architected for Scalability and Speed

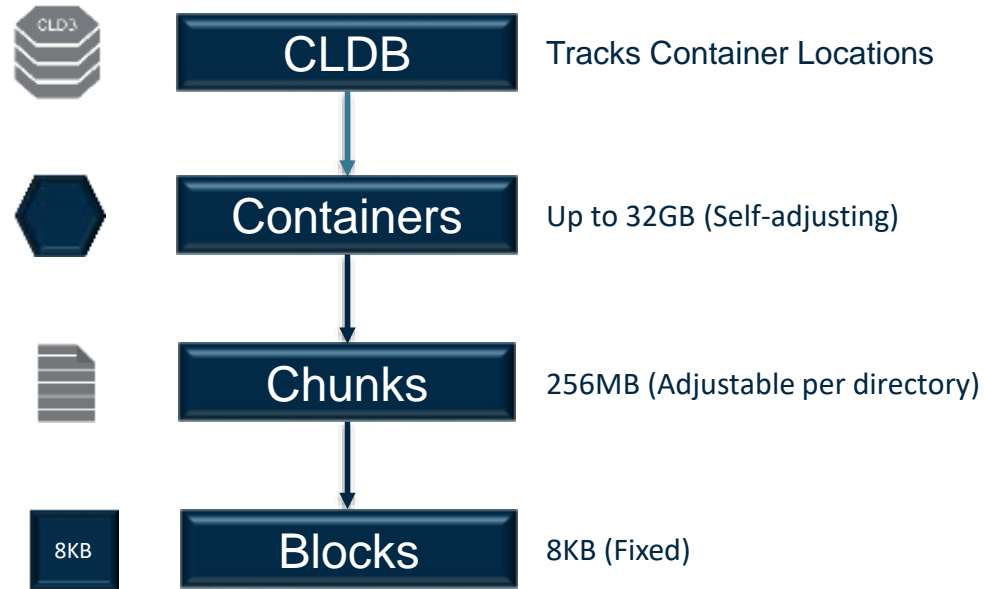
## CLASSIC HDFS



HDFS's NameNode is a single process containing an in-memory table which stores all file metadata and the locations of all Blocks in the filesystem. The 128MB Block size was selected for batch programs.

**Scalability = (# Blocks x Block Size)**

## MAPR-FS



The CLDB stores the locations of Containers. Containers are replicated data structures which live in Storage Pools. Containers hold File Chunks and Directories and the locations of File Chunks. This architecture has an extra step of indirection, but it introduces scale and parallelism. Containers are infrequently created/moved, so the CLDB is relatively stable. Containers are also relatively large (and therefore, relatively few), so the CLDB is relatively small. The CLDB's data is stored in a Container and is replicated, so it is highly available.

**Scalability = (# Containers x Container Size)**

Detailed video at: <http://bit.ly/2j2j8wx>

### MapR-FS Design Benefits:

- High Parallelism
- Extreme Scalability
- Extreme Throughput
- Simpler Multitenancy

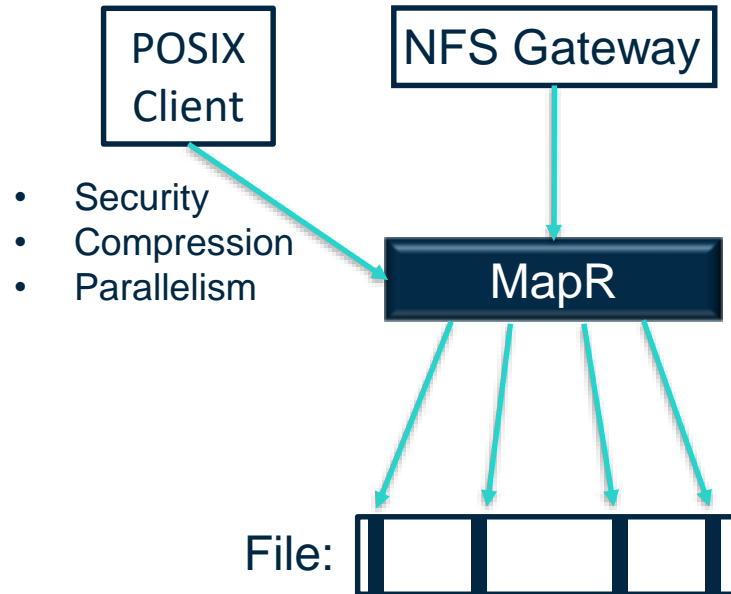
# MapR and HDFS – Other Similarities and Differences

## Similarities:

- Implement HDFS API
- Distributed File System
- Automatic Replication of Data
- Commodity Servers
- Local Disks

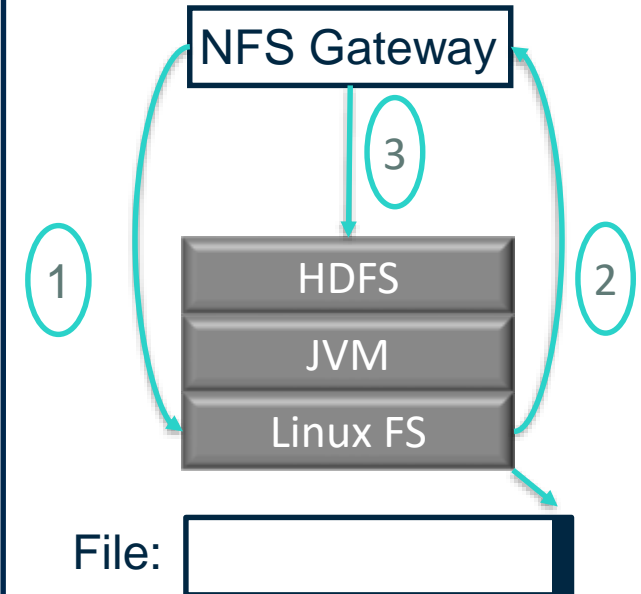
## Differences:

- MapR-FS: Fully Read/Write
- MapR-FS: No Java, No JVM
- MapR-FS: No Linux File System
- MapR-FS: Direct NFS Writes
- MapR-FS: POSIX client



- Security
- Compression
- Parallelism

- Fully Read/Write
- Direct Disk I/O
- Real-time NFS access

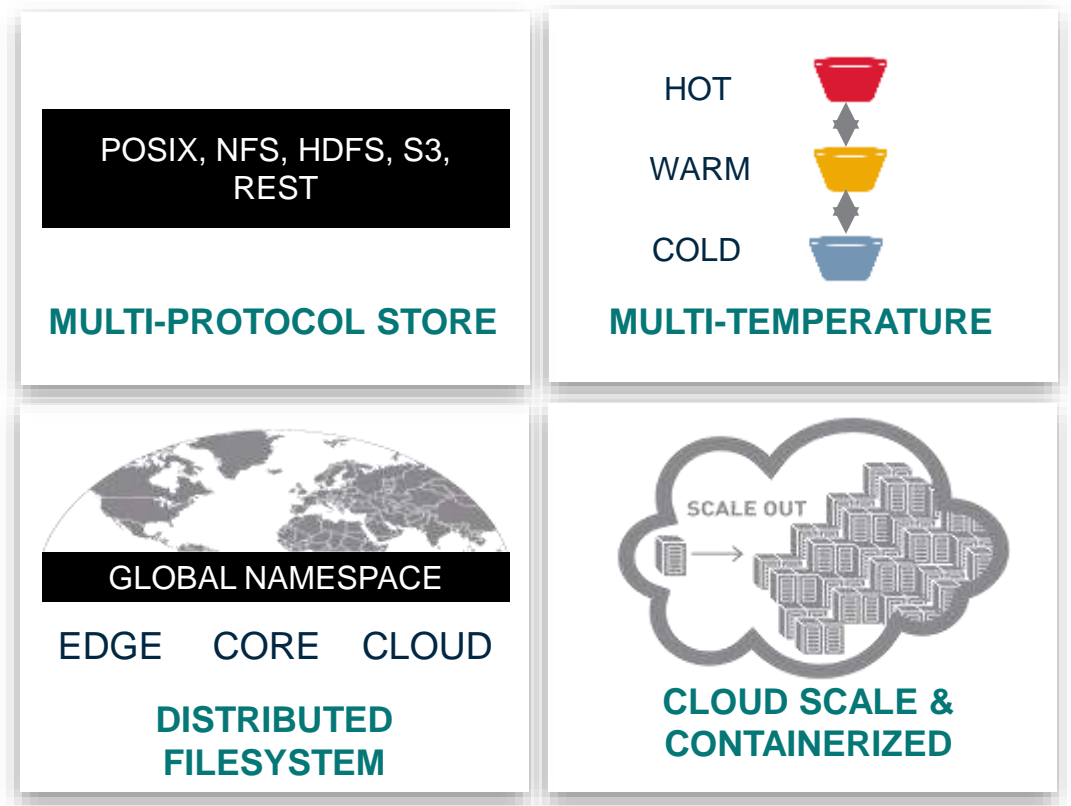


- Write Once, Append
- Needs JVM and Linux FS
- Byte-ordering issues with NFS

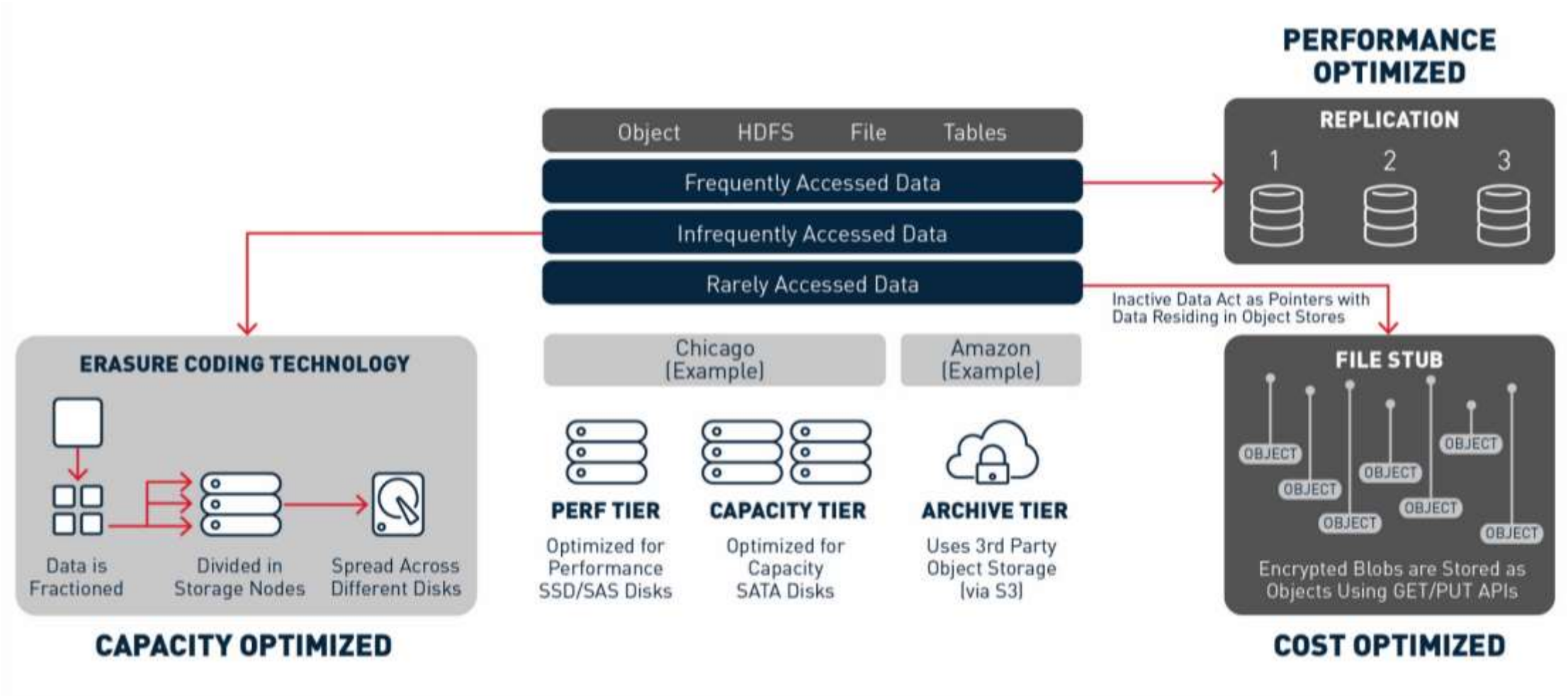
Detailed video at: <http://bit.ly/2Af4A8A>

# MapR FS Today

- Patented distributed Exabyte scale filesystem
- Read/write with POSIX, NFS, HDFS, S3, REST interfaces
- Global namespace across locations, clouds, edge
- Policy based tier into S3 object store while preserving metadata, with transparent, auto or scheduled recall
- Designed for high-scale, high-reliability
- Optimized for 'in-place' analytics, ML, AI workloads
- Data protection, point in time recovery with snapshots, disaster recovery with mirroring
- Data placement topology control
- SSD, NVME optimized



# Overview of Storage Tiering in MapR



---

# Call to Action

1. AI HPC “convergence” still at infrastructure level. Work on going to go up the tsack at workflow and application level
2. Co-design opportunity for h/w and s/w stack to make it possible to cater the fast moving requirments in this space
3. HPE platform for AI and HPC will incorporate elements from BlueData, MapR, Cray and existing portfolio to create turnkey and vertical specific architecture
4. More accoucements coming at SC19 esp. around Cray. Please join us in Denver and HPCast!



**Hewlett Packard**  
Enterprise

**Thank you**

Q&A



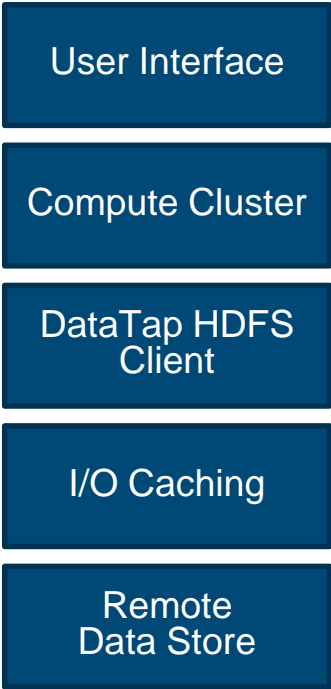
**Hewlett Packard**  
Enterprise

# Backup

BlueData/MapR

# Architecture: Remote Storage Access

## DataTap + IOBoost (Caching) Service



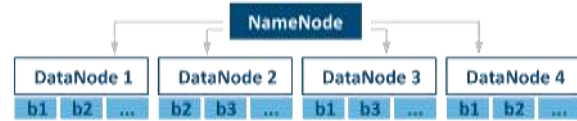
```
hadoop jar cdh-examples.jar wordcount dtap://TenantStorage/tab1/tab1.csv
```



Resource Manager and Node Managers open up HDFS connections to get tab1.csv data

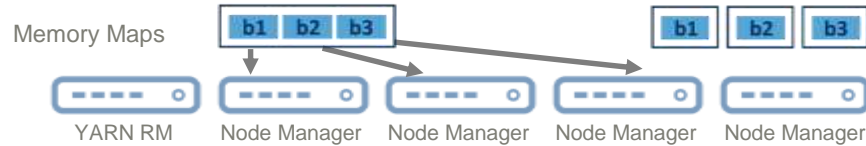
**DTAP multi-threaded C Client** (HDFS version independent)  
Uses protobuf\* 2.5 to connect with external storage

“Where are the data blocks?”



“They are here!”

Data blocks (64MB or 128MB) from remote HDFS are read and streamed to I/O Cache  
Read ahead caching (b1, b2, b3 cached) and write behind vs. sequential read one block at a time

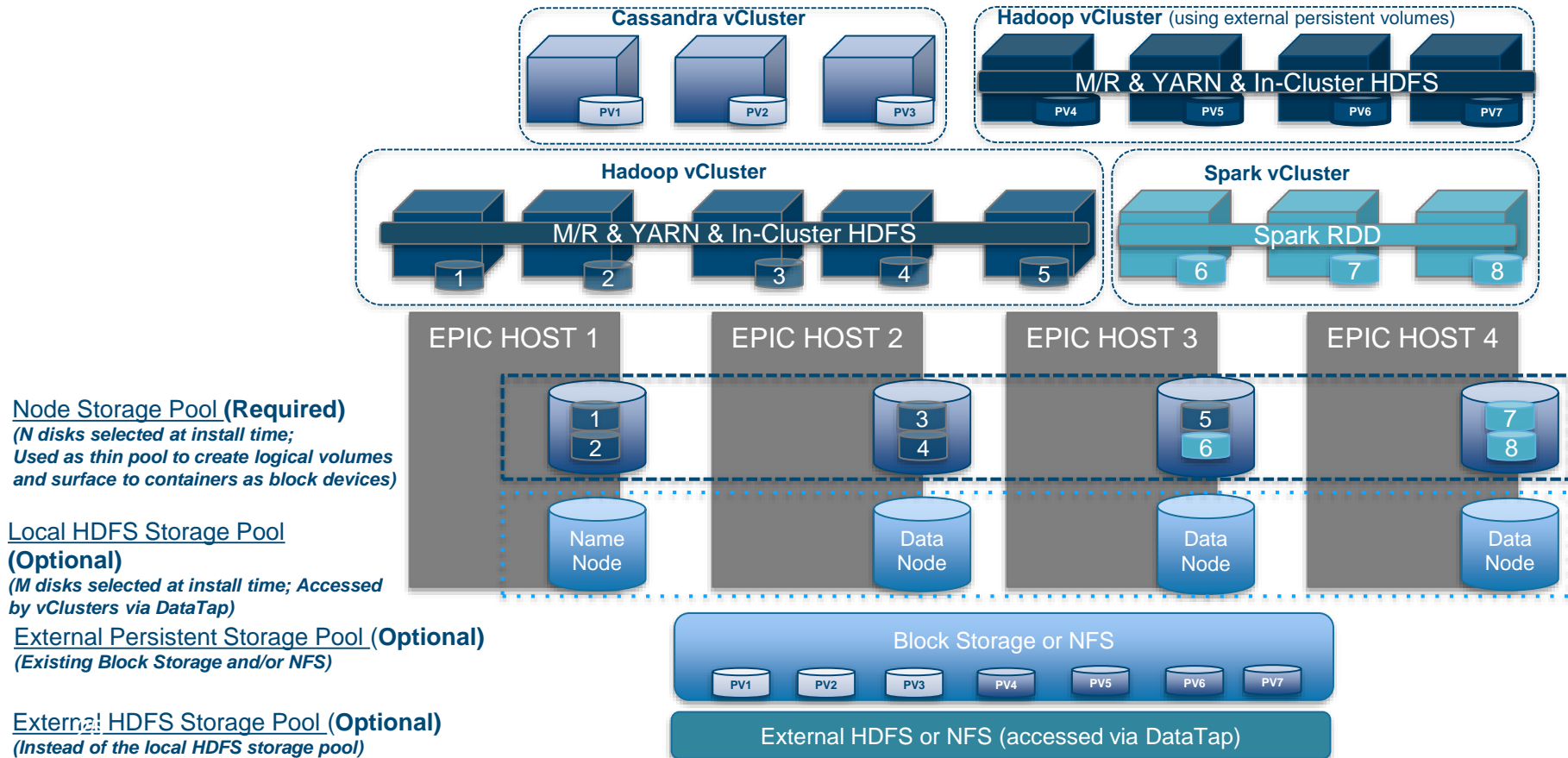


Blocks supplied to YARN cluster for computation using direct memory map – bypass virtual network stack to get higher I/O throughput

\*Protocol buffers: Protocol Buffers are a binary wire format which is compact, forward-compatible, and backward-compatible  
HPE Restricted – HPE and Authorized Partner Internal Use Only

Kerberos and TDE Support

# Storage Architecture



**Node Storage Pool (Required)**  
*(N disks selected at install time; Used as thin pool to create logical volumes and surface to containers as block devices)*

**Local HDFS Storage Pool (Optional)**  
*(M disks selected at install time; Accessed by vClusters via DataTap)*

**External Persistent Storage Pool (Optional)**  
*(Existing Block Storage and/or NFS)*

**External HDFS Storage Pool (Optional)**  
*(Instead of the local HDFS storage pool)*