Why IBM Cloud Private ?

Hybrid Cloud, Containers & Micro-service approach for IBM i Modernization

Benoit Marolleau – Cloud Architect IBM Systems
benoit.marolleau@fr.ibm.com

IBM **Systems** IBM **Cloud**

Agenda

- Introduction - DevOps & Continuous Innovation
- New Development Models & Paradigms
  - Cloud Computing, Containers & Technologies Docker, Kubernetes , Microservices
- IBM Cloud Private
- Integration w/ my IBM i Apps
- **Demo :**

  Agile Deployment & Hybrid Cloud w/ IBM Cloud Automation Manager

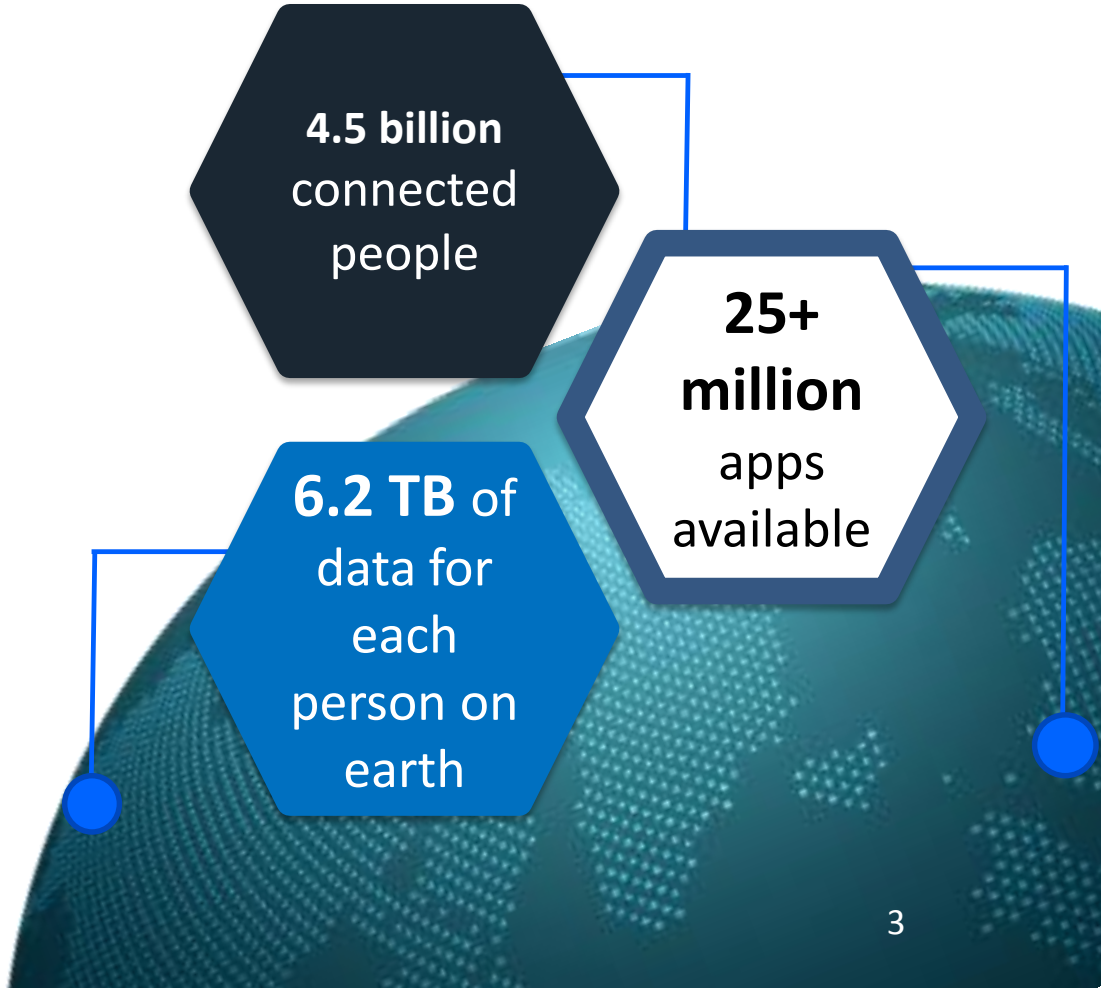
The world is becoming more connected than ever

Businesses must be ready to face the challenge

To win in the connected economy, **enterprises are focusing on interactions and value exchange** across a partner ecosystem

You need:

- A better, more compelling customer experience
- An infrastructure that scales out autonomously
- To bring teams together across a partner ecosystem
- Continuous innovation to deliver software faster, consistently, and reliably



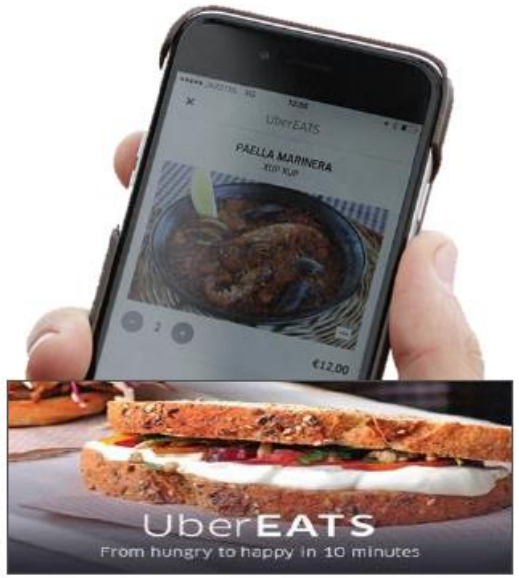
Source:[1] IDC: The Digital Universe of Opportunities: Rich Data and the Increasing Value of Internet of Things, April 2014

Source:[2] RisingStack: How Enterprises Benefit From Microservices Architecture, February 2016

Customers and ecosystem partners expect innovative and personalized experiences

Location Aware

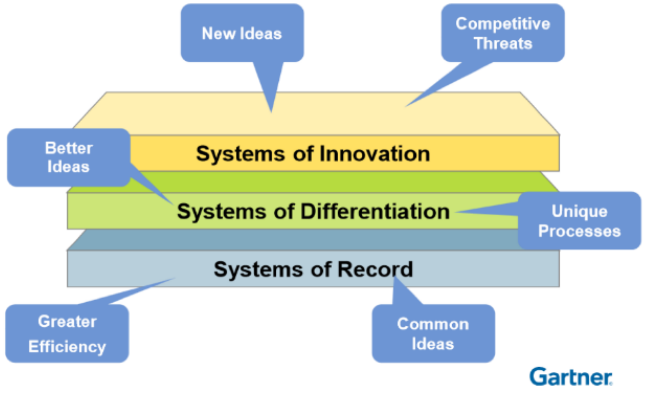
On Demand

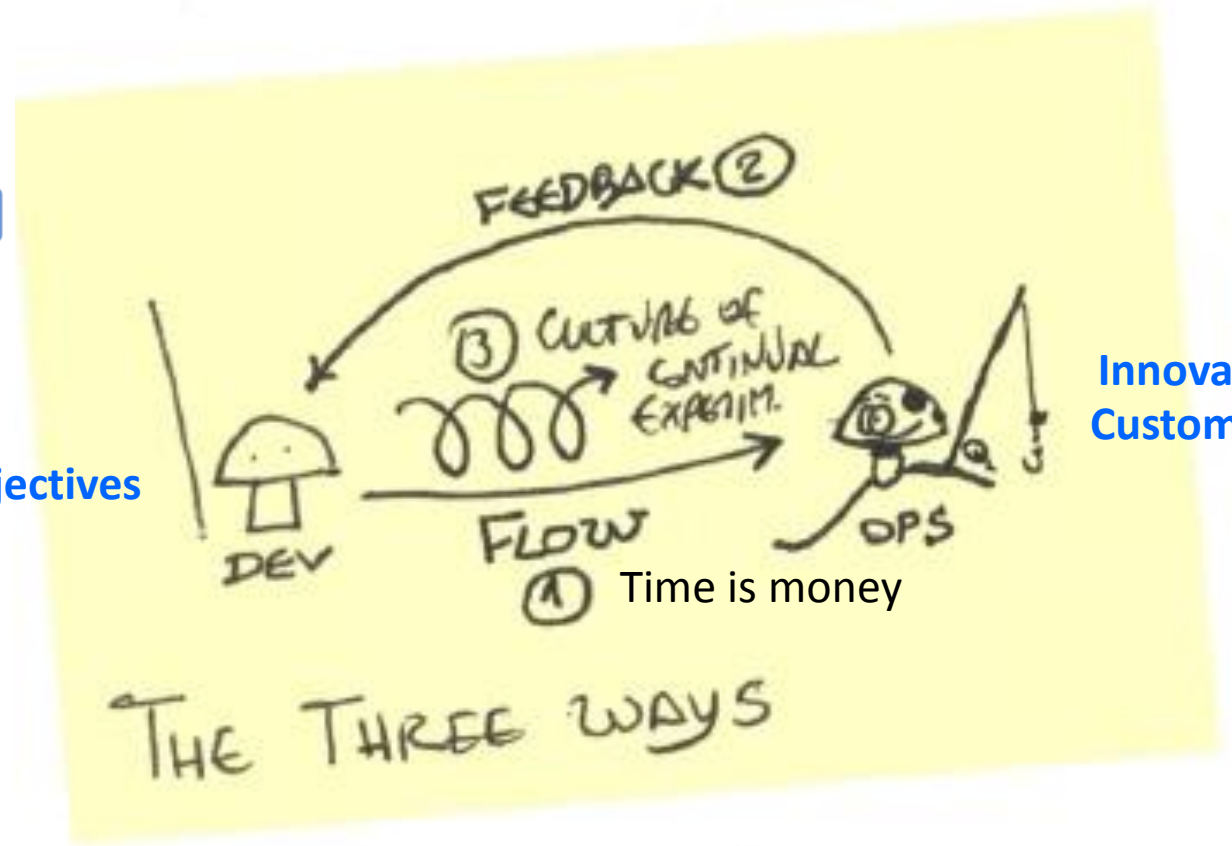


Personalized and Engaging

DevOps & Innovation continue

- DevOps “3 Ways”





Business Objectives & Ideas

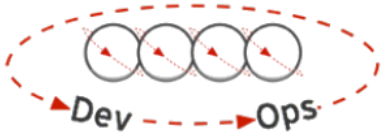
1. Accelerate Delivery
2. Feedback Loop
3. Continuous Innovation

Innovation Customer Experience

IT aligned with new business needs

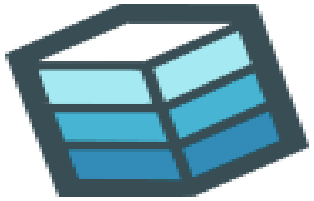
thirty years

Development Process	Application Architecture	Deployment & Packaging	Application Infrastructure
Waterfall	Monolithic	Physical Servers	Datacenter
Agile	N-Tier	Virtual Servers	Hosted
DevOps	**Microservices**	**Containers**	**Cloud**

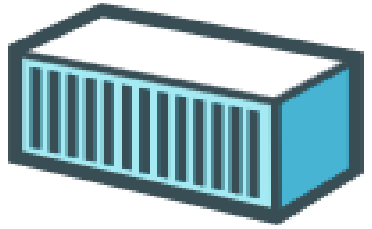


Containers

Containers vs. VM



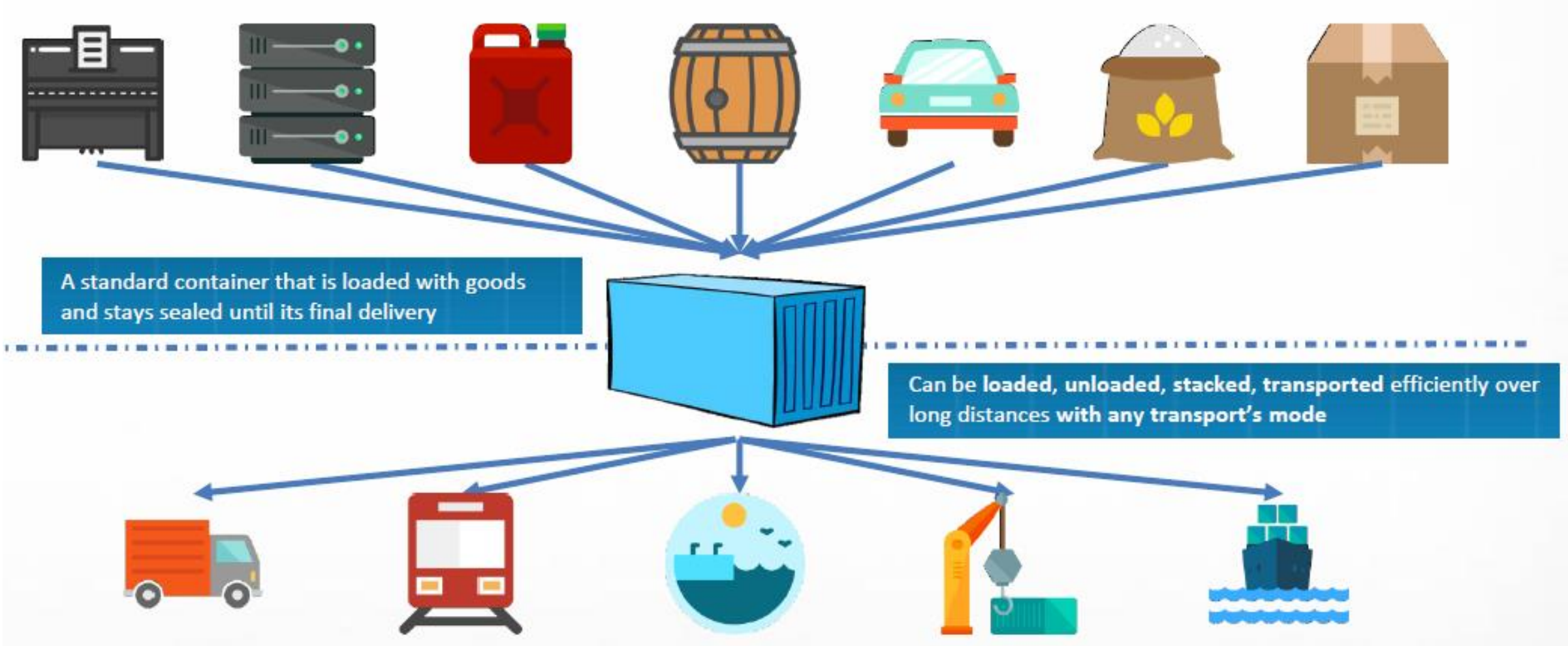
Build



Ship

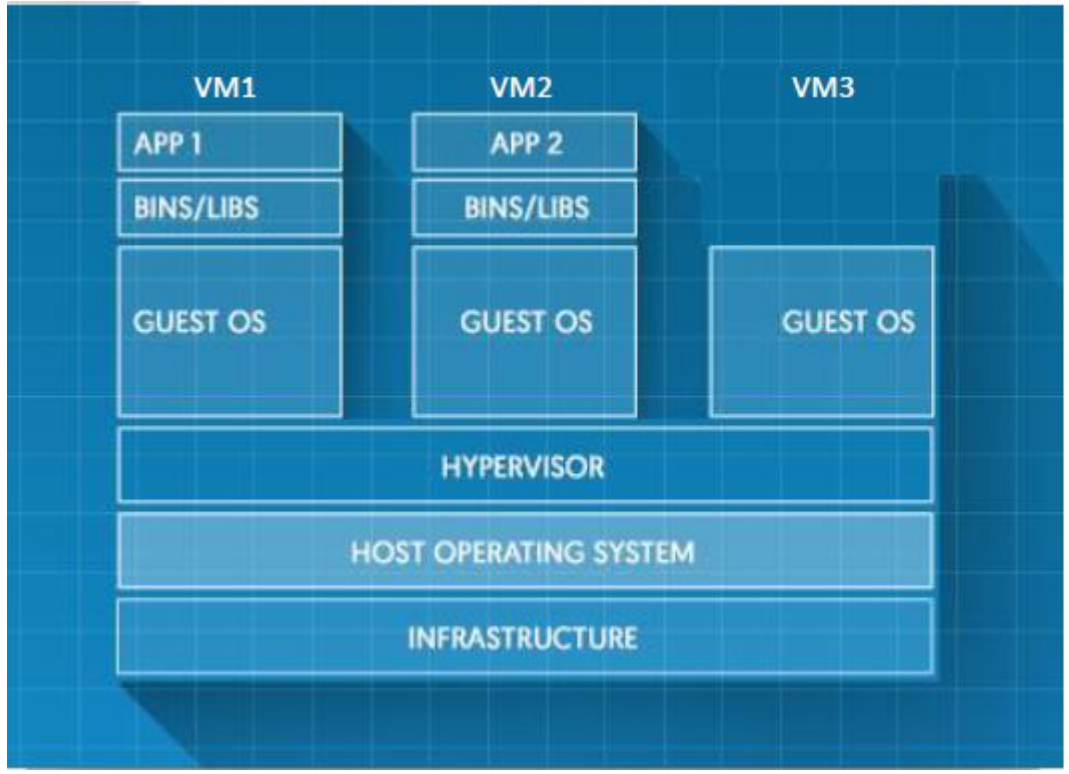
Run

Docker: Application portability



Containers vs VM

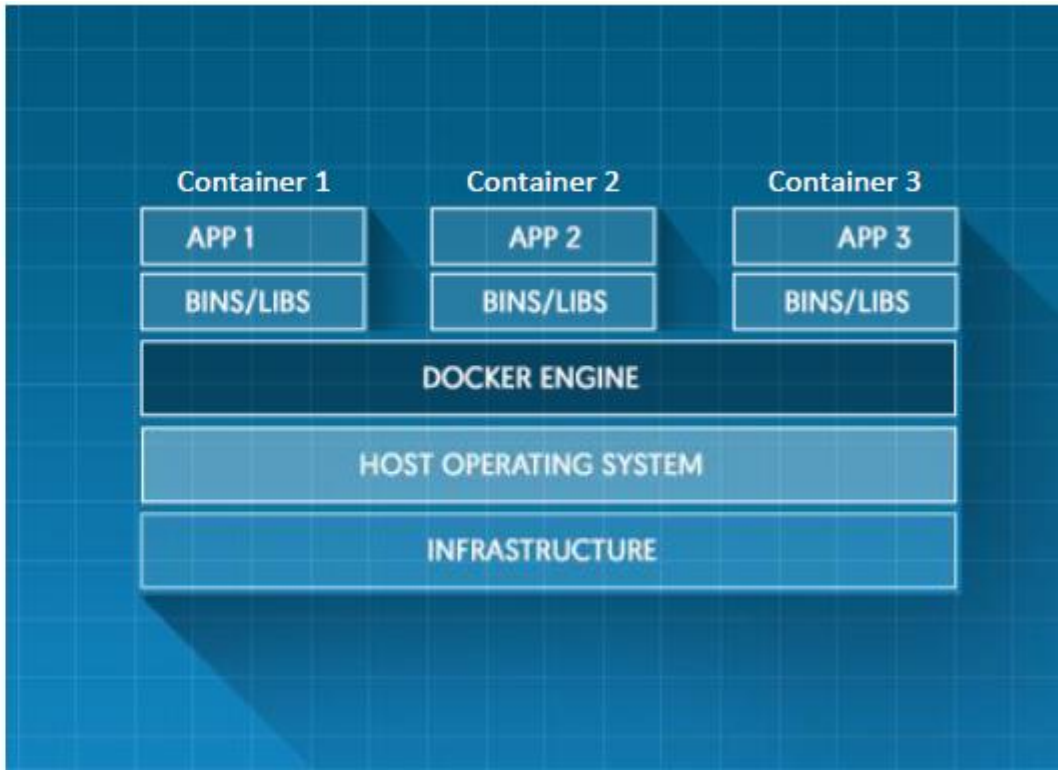
Virtual Machines



Virtualization Pro:

- Better Security / Isolation
- Allow different Kernel between VMs
- Not Limited to Linux OS

Containers



Containers Pro:

- Better resources utilization
- Less overhead compare to VM
- Light compare to VM
- Very FAST START : No Boot
- No special hypervisor mode access required

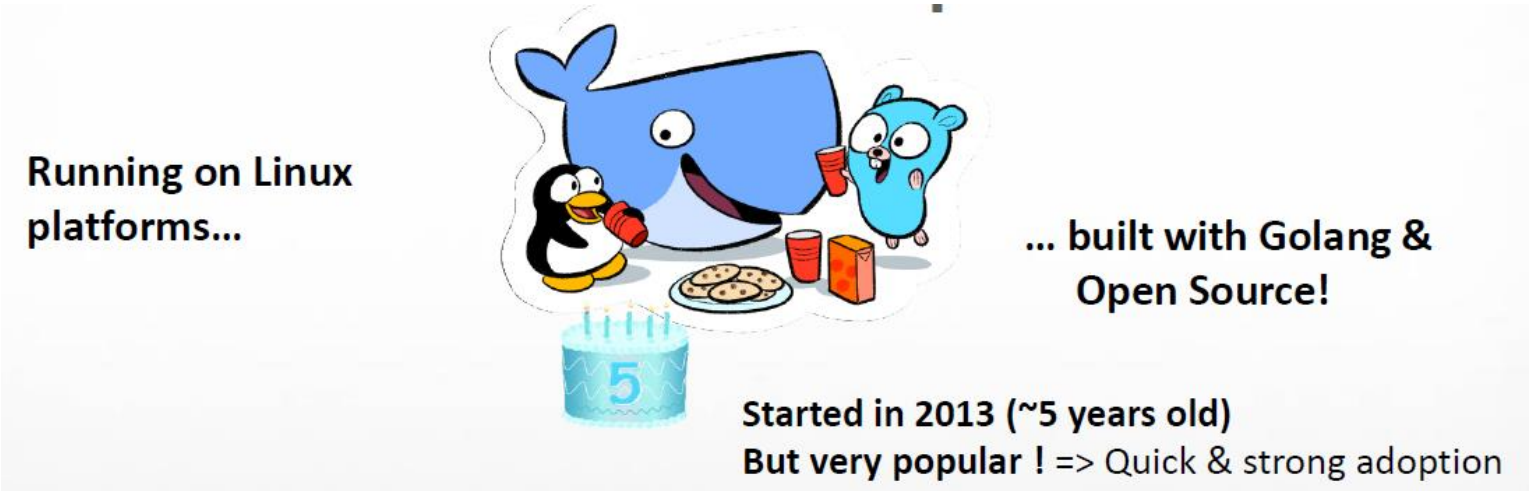
=> could be nested without performance impact.

Docker : Concepts

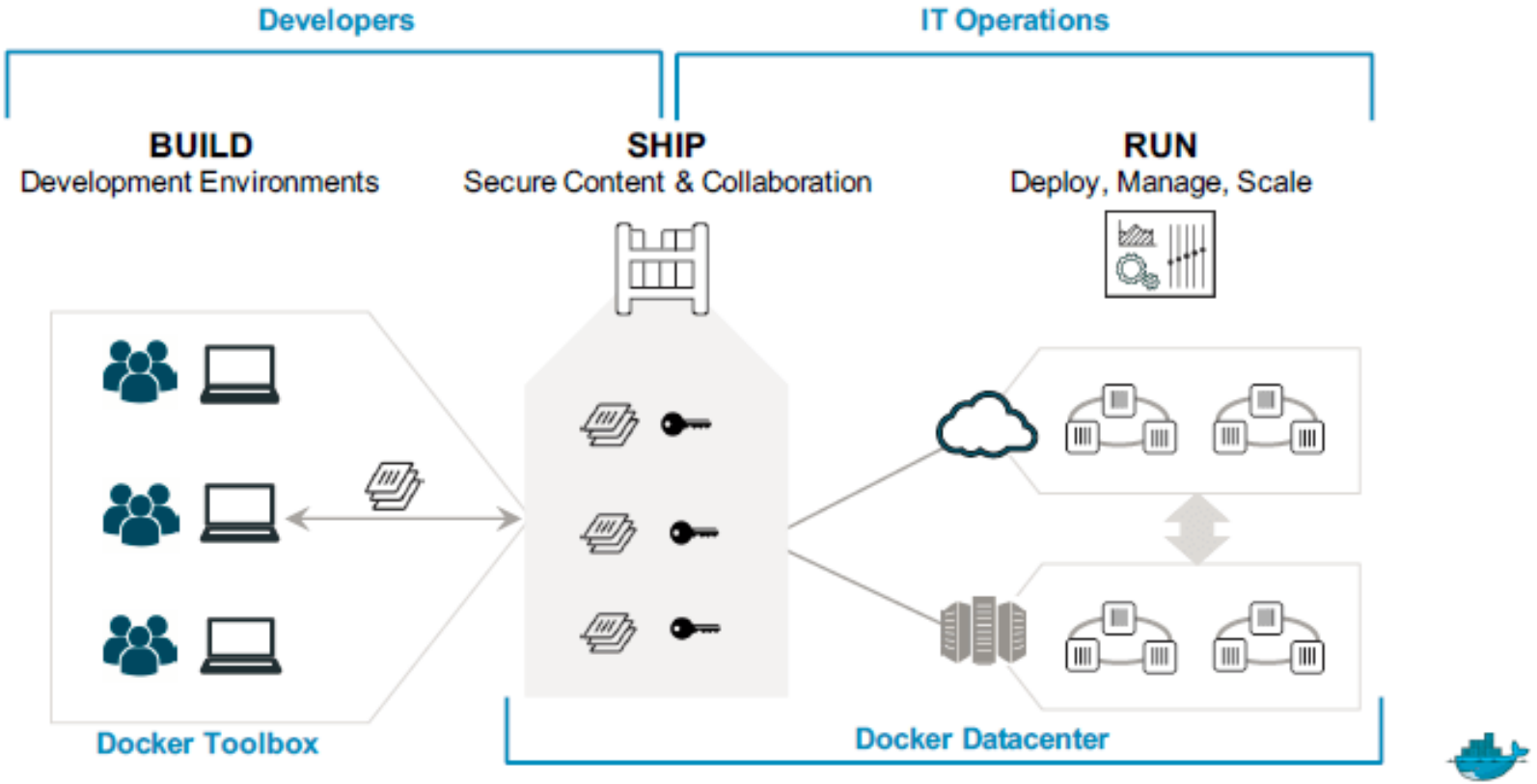
- **Engine**: Runs on Linux, it provides the operating environment for Docker containers.
- **Image:** Read-only templates for containers, stored and managed in a registry.

  Once instantiated a container is created.
- **Dockerfile:** Defines a Docker image as if it was code; used to re-build an image
- **Registry:** A service that allows to store and manage Docker images
- **Container:** Standard unit to package an application and its dependencies: binaries, libraries, system tools…

  So that it can be moved between environments and run without changes.



Containers & DevOps



IBM Cloud amazon webservices™ Microsoft Azure

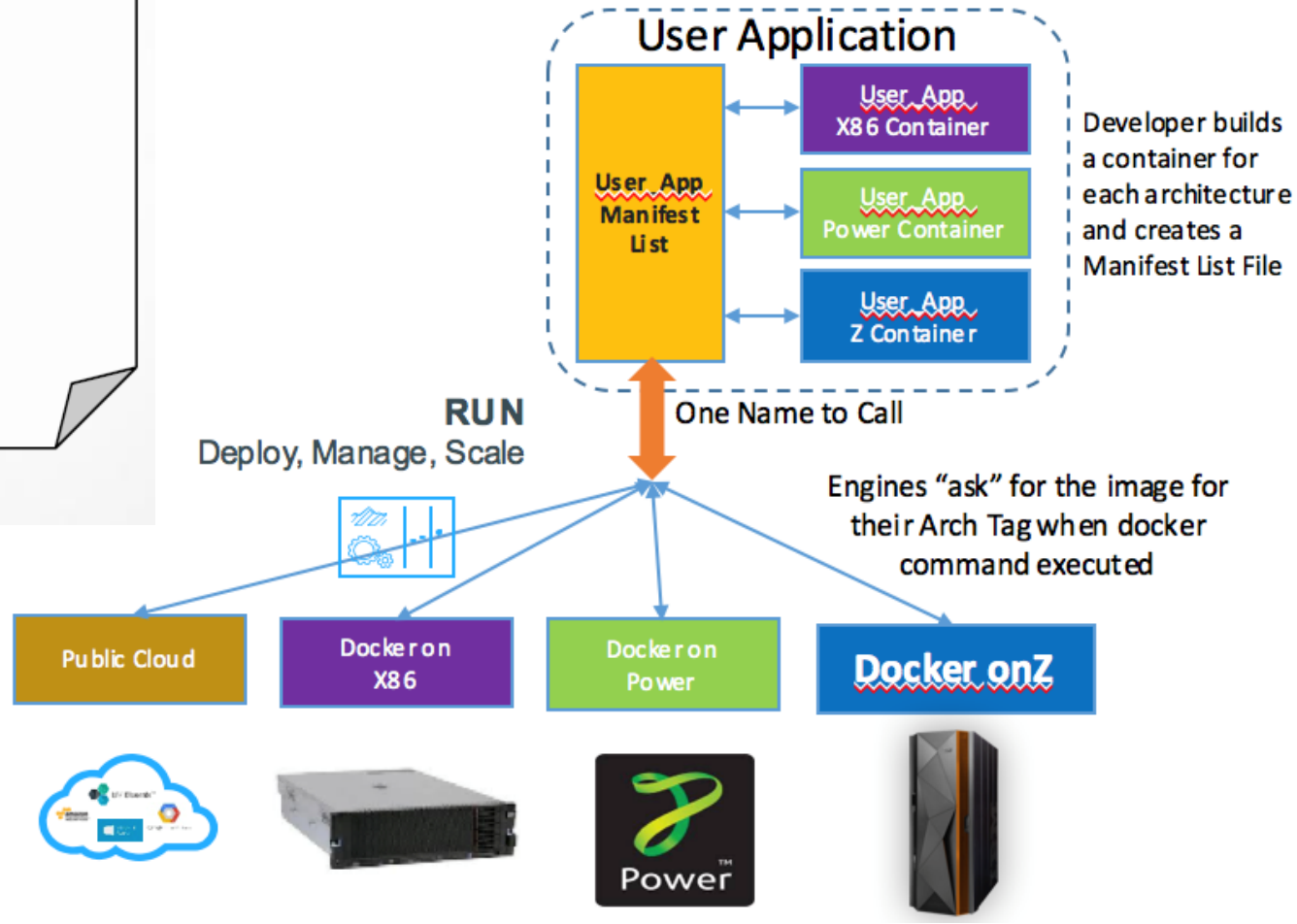
DevOps: Break down barriers between Dev and Ops teams to improve the app development process

CI/CD: Enable developers to develop and test applications more quickly and within any environment

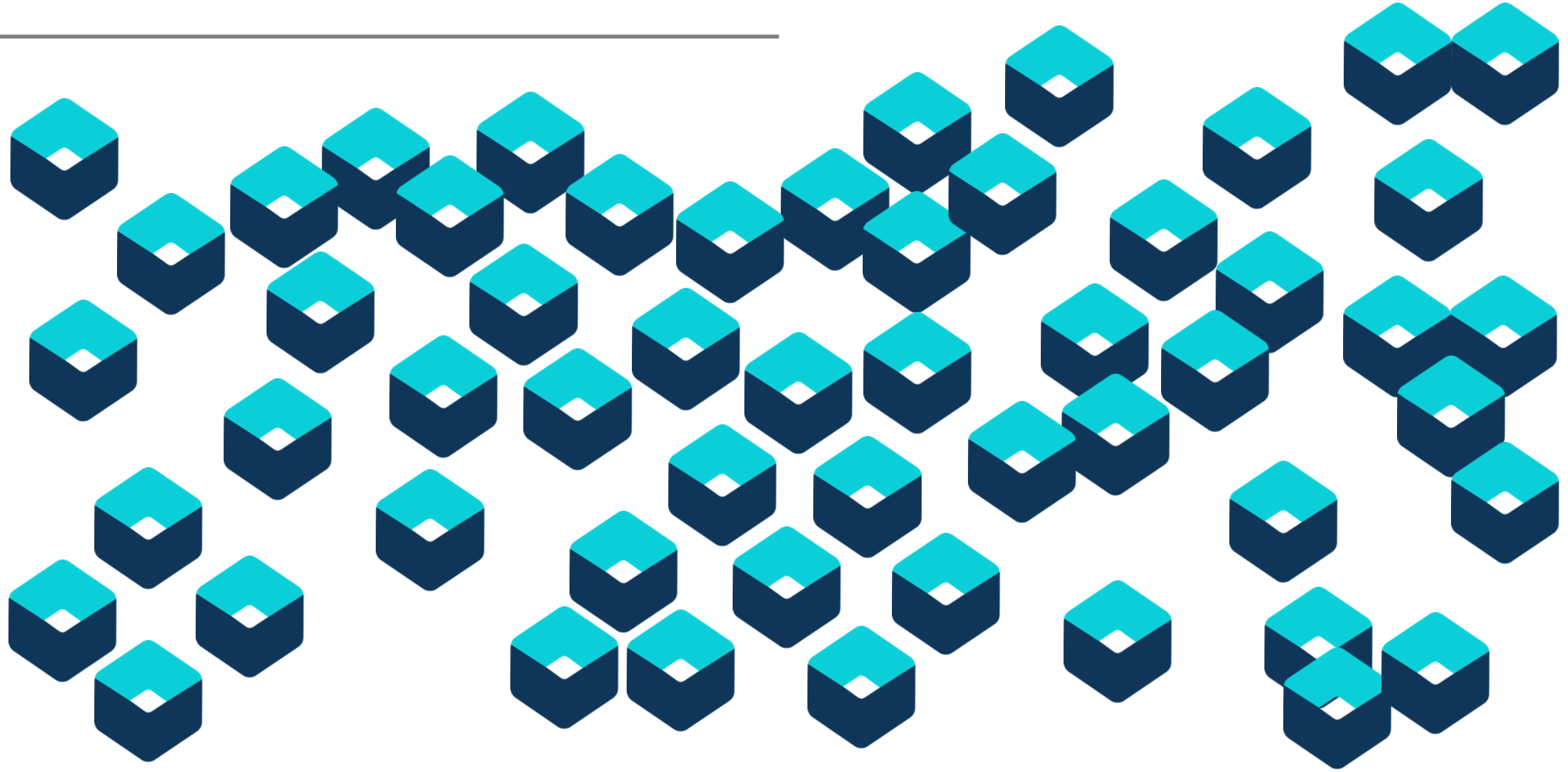
Multi-Arch & Multi-Cloud Enablement for Docker

```
From ubuntu:14.04

RUN apt-get install <your_app>

ADD <files like conf>

EXPOSE <tcp/udp port>

CMD <start_your_app>
```

Dockerfile example



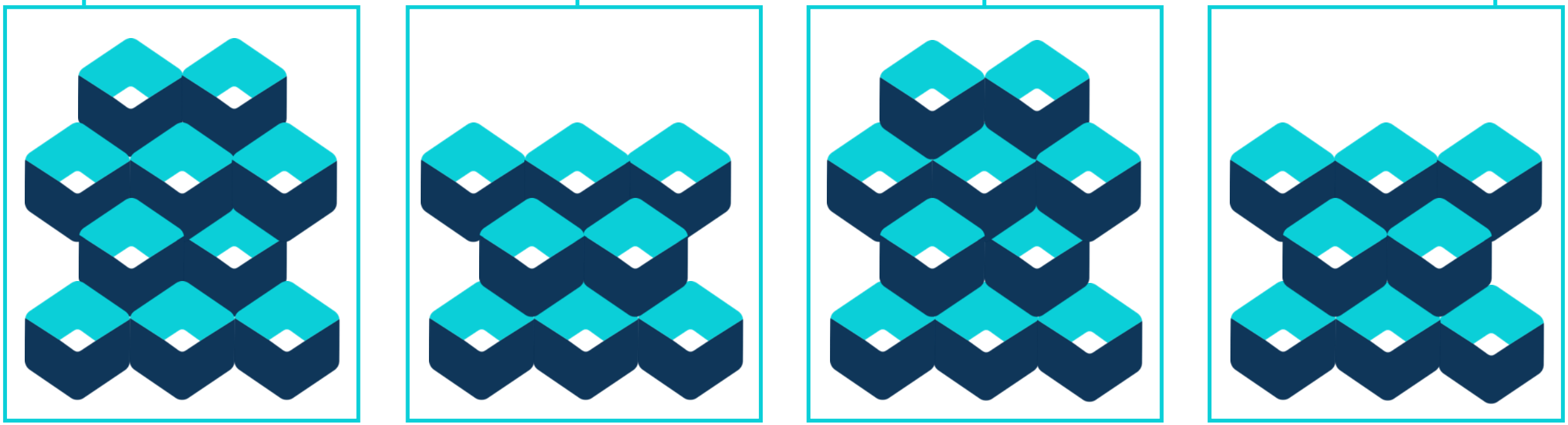
Containers are great but … can lead into lack of control & chaos



Kubernetes – (Κυβερνήτης - Captain in Greek)

Regain control with Containers and Kubernetes

- Organize and Govern the Container Chaos



What do Kubernetes really offer ?

Intelligent Scheduling		Automatically places containers based on their resource requirements and other constraints, while not sacrificing availability. Mix critical and best-effort workloads in order to drive up utilization and save even more resources.
Self Healing		Restarts containers that fail, replaces and reschedules containers when nodes die, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.
Horizontal Scaling		Scale your application up and down with a simple command, with a UI, or automatically based on CPU usage.
Service Discovery and Load Balancing		No need to modify your application to use an unfamiliar service discovery mechanism. Kubernetes gives containers their own IP addresses and a single DNS name for a set of containers, and can load-balance across them.
Automated rollout and rollback		Kubernetes progressively rolls out changes to your application, while monitoring application health to ensure it doesn't kill all your instances at the same time. If something goes wrong, Kubernetes will rollback the change for you. Take advantage of a growing ecosystem of deployment solutions.
Secret and configuration management		Deploy and update secrets and application configuration without rebuilding your image and without exposing secrets in your stack configuration.

Kubernetes Concepts

- Declarative Configuration (YAML) & Decoupling
  - Services, loosely coupled apps
- Consistency / Scaling
  - Application SLA vs. OS SLA
- Abstraction layer
  - K8s is present in all Cloud Providers
  - Pods, or groups of containers =
  - Kubernetes services =
  - Namespaces =
- Efficiency
  - Machine usage optimization – distribution of application

A group of co-located containers

A replication controller ensures that
a specified number of pod replicas
are running at any one time.

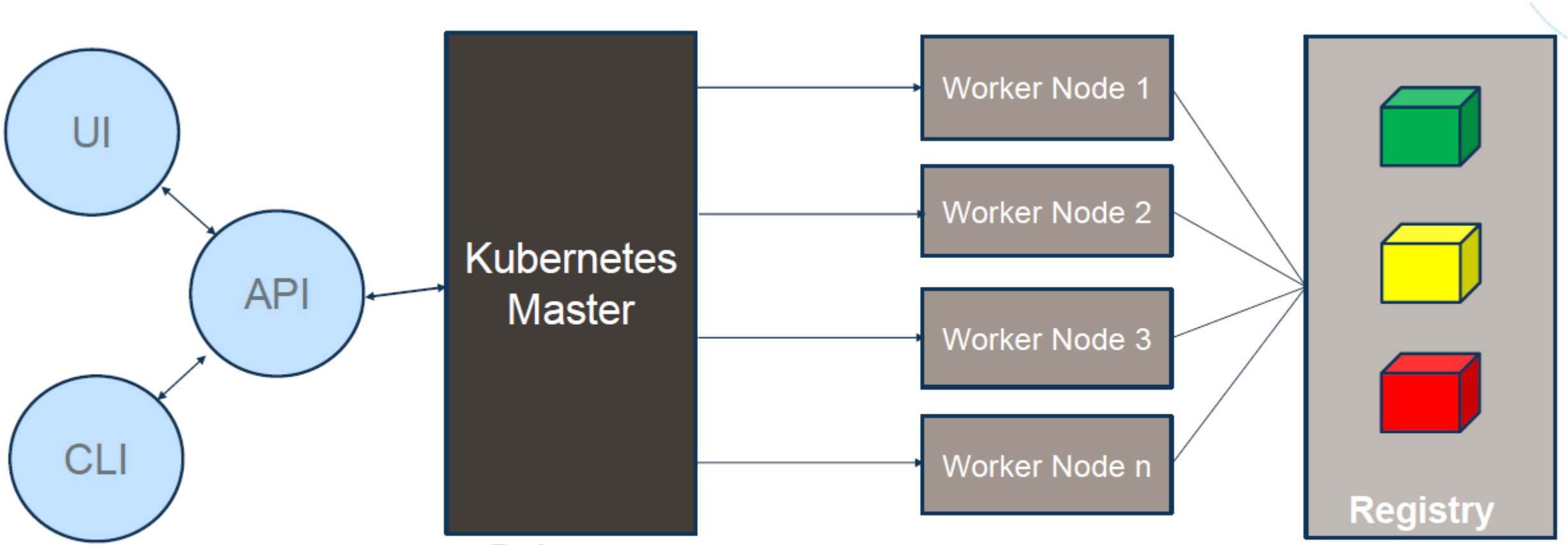
A volume is a directory, possibly with some data in it, which is accessible to a Container as part of its filesystem.

A StatefulSet is a Controller that provides a unique identity to its Pods. It provides guarantees about the ordering of deployment and scaling.

A service defines a set of pods and a means by which to access them, such as single stable IP address and corresponding DNS name.

A label is a key/value pair that is attached to a resource, such as a pod, to convey a user-defined identifying attribute.

Kubernetes Architecture



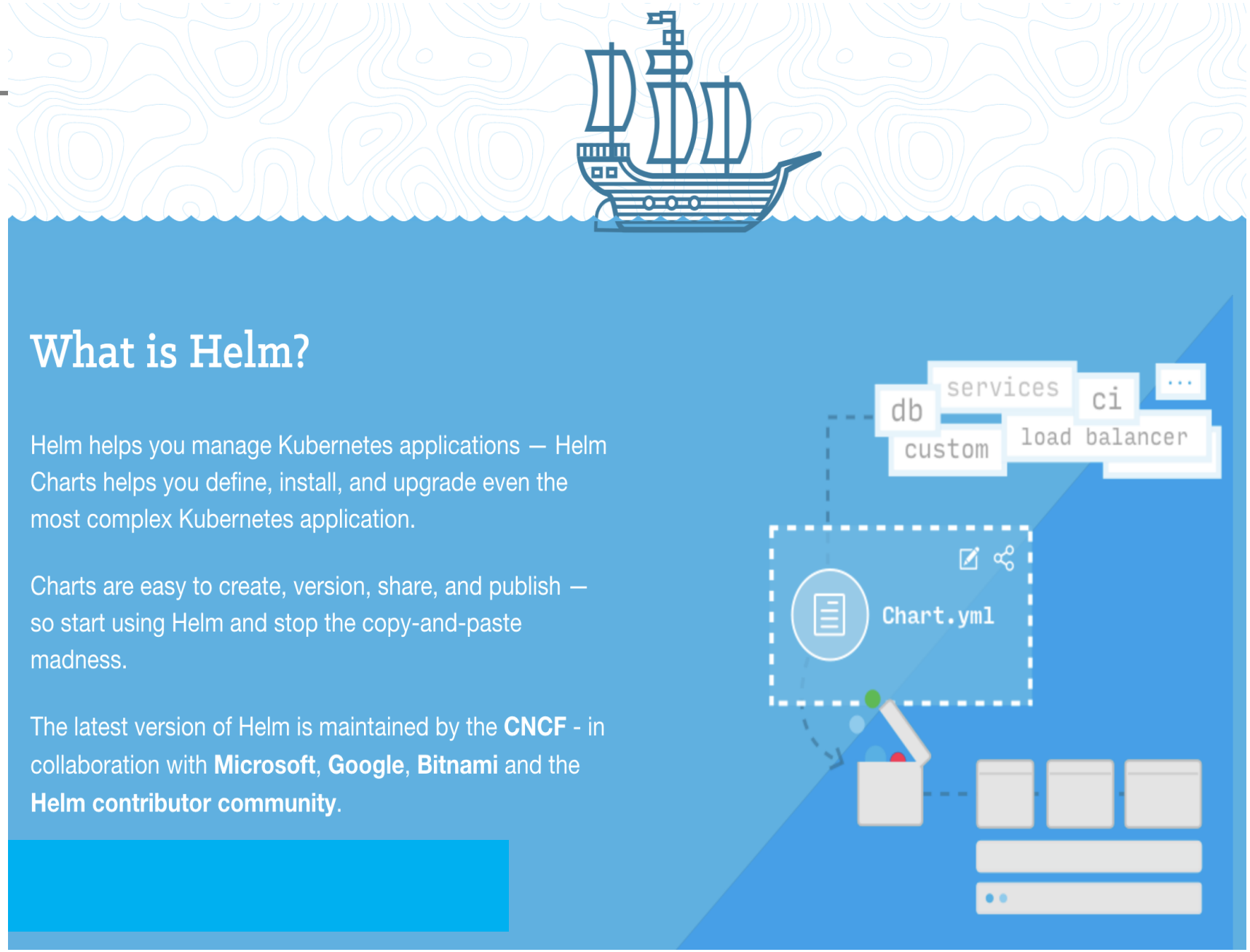
- Etcd
- API Server
- Controller Manager Server
- Scheduler Server

And HELM is …

The package manager for Kubernetes

Helm is the best way to find, share, and use software built for Kubernetes.

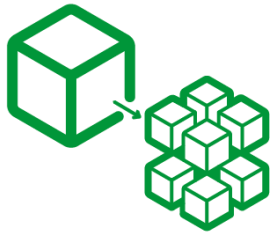
Tells Kubernetes all it needs to know about an application its parameters and dependencies

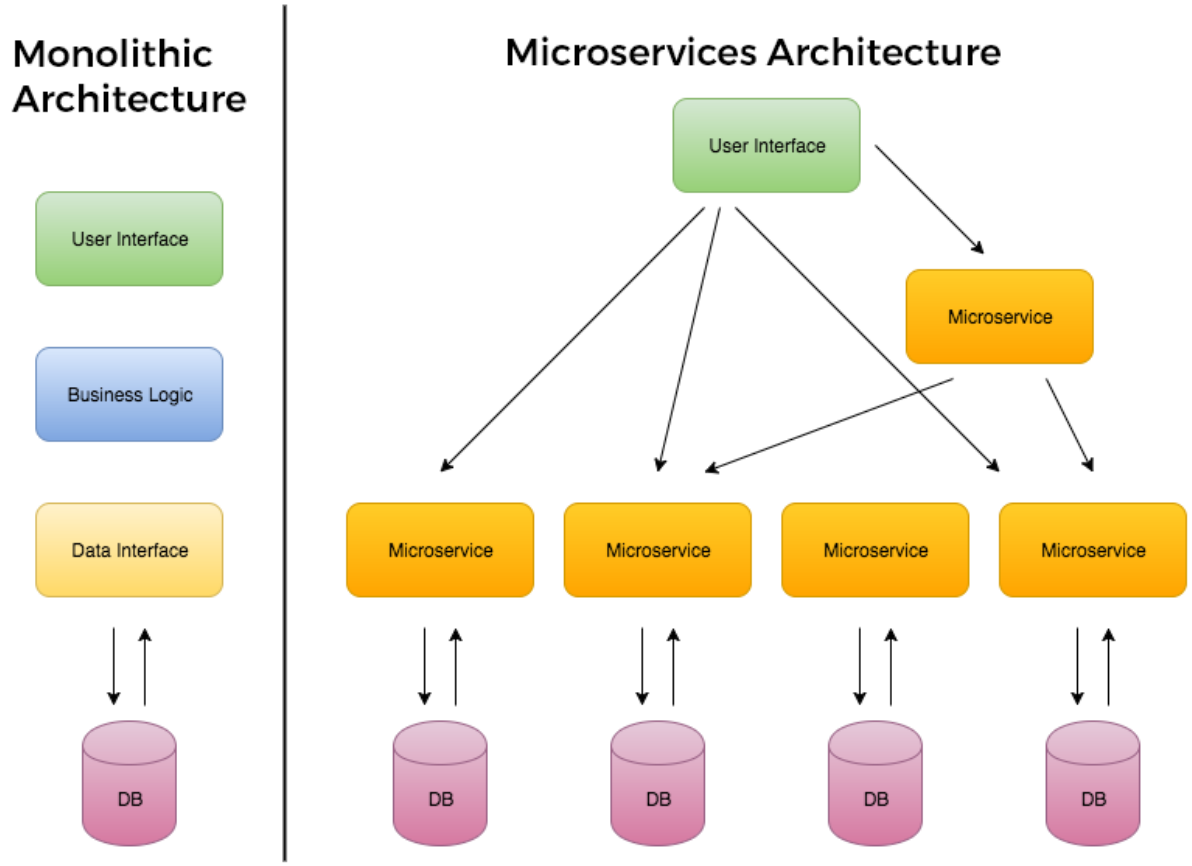


Microservices

Microservice Approach

Microservices & Cloud Native Apps





Why microservices?

The microservices revolution

Connecting people and digital apps better than ever before

A microservices architecture is gaining traction for developing and delivering cloud-native workloads across public, private, and hybrid application environments

60% of new apps will use cloud-enabled continuous delivery and cloud-native application architectures to enable faster innovation and business agility*

IDC FutureScape: Worldwide Cloud 2016 Predictions – Master the Raw Material of Digital Transformation, November 2015

Why?

- ✓ Decomposed into small pieces
- ✓ Loosely coupled
- ✓ Easier to scale development
- ✓ Improved fault isolation
- ✓ Each service can be developed and deployed independently
- ✓ Eliminates any long-term commitment to a technology stack

Why 12 factor apps?

I. Codebase
One codebase tracked in revision control, many deploys

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing services
Treat backing services as attached resources

V. Build, release, run
Strictly separate build and run stages

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

X. Dev/prod parity
Keep development, staging, and production as similar as possible

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

https://12factor.net/

Code

I. Codebase
One codebase tracked in revision control, many deploys

V. Build, release, run
Strictly separate build and run stages

X. Dev/prod parity
Keep development, staging, and production as similar as possible

Deploy

II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing services
Treat backing services as attached resources

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding

Operate

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

Code

I. Codebase
One codebase tracked in revision control, many deploys

V. Build, release, run
Strictly separate build and run stages

X. D
Keep development, staging, and production as similar as possible

Deploy

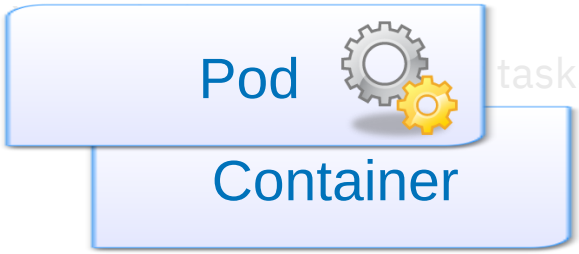
II. Dependencies
Explicitly declare and isolate dependencies

III. Config
Store config in the environment

IV. Backing services
Treat backing services as attached resources

VI. Processes
Execute the app as one or more stateless processes

VII. Port binding
Export services via port binding



Operate

VIII. Concurrency
Scale out via the process model

fast
own

XI. Logs
Treat logs as event streams

tasks as

Code

I. Codebase
One codebase tracked in revision control, many deploys

V. Build, release, run
Strictly separate build and run stages

X. Dev/prod parity
Keep development, staging, and production as similar as possible

Deploy

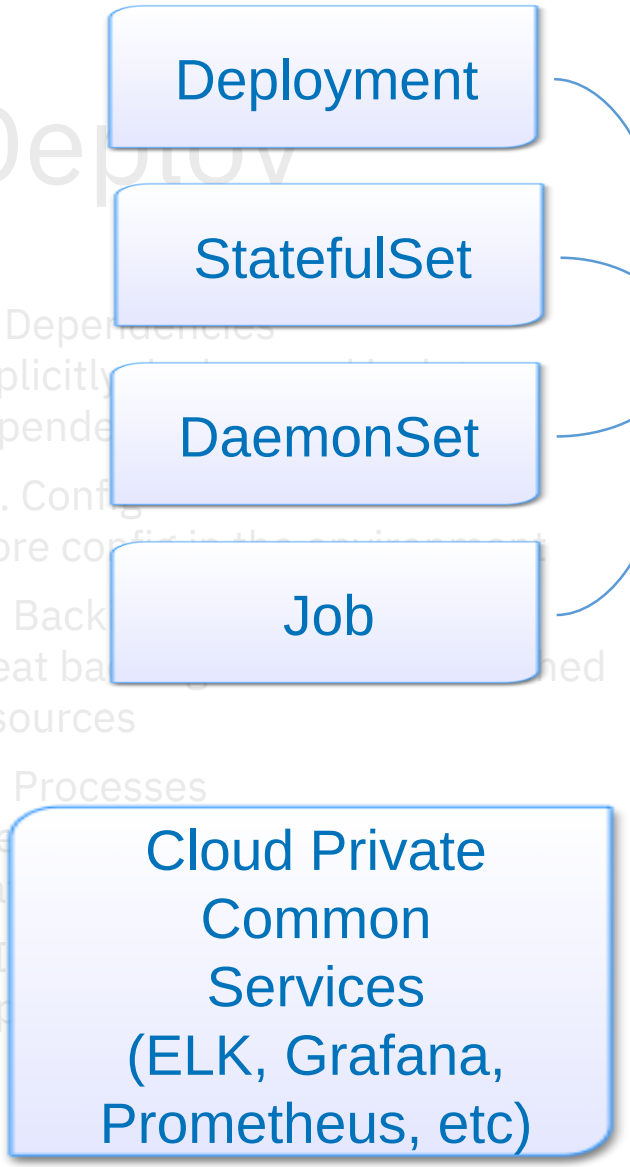
II. Dependencies
Explicitly
depende

III. Conf
Store co

IV. Back
Treat ba
resources

VI. Processes
Exe
sta

VII
Exp



Operate

VIII. Concurrency
Scale out via the process model

IX. Disposability
Maximize robustness with fast startup and graceful shutdown

XI. Logs
Treat logs as event streams

XII. Admin processes
Run admin/management tasks as one-off processes

IBM Cloud Private

IBM Cloud Private

- A private cloud platform for enterprises to develop and run their workloads locally
- An integrated platform consisting of PaaS and developer services necessary to create, run, and manage cloud applications
- Container infrastructure, orchestration and management
  - ✓ Resource management
  - ✓ Application life-cycle management/schedule/deployment
  - ✓ Scaling, rolling upgrade
  - ✓ Service registry/discovery
  - ✓ Distributed storage management
  - ✓ Image/software repository management
  - ✓ Configuration management
  - ✓ User/Account management

kubernetes

Kubernetes based container platform

Industry leading container orchestration platform

Common Services

Simplify operations management, DevOps, and hybrid integration

IBM Middleware, Data and Analytics services

Optimize current investments and rapidly innovate

IBM **Cloud** Private Solution Overview

IBM Middleware & Open Source – e.g. Data, Analytics and Developer Services

Cloud-enabled middleware, application runtimes, messaging, databases & analytics to optimize current investments and rapidly innovate

IBM WebSphere · IBM DB2 · IBM Data Science Experience · open source

Core Operational Services

To simplify Operations Management, Security, DevOps, and hybrid integration

Grafana · Prometheus

kubernetes

HELM

Kubernetes-based Container Platform

Industry leading container orchestration platform across private, dedicated & public clouds

CLOUDFOUNDRY

Cloud Foundry

For prescribed application development & deployment

HashiCorp Terraform

Terraform (CAM)

Infrastructure as Code for provisioning on public and on-prem cloud

IBM Cloud · Microsoft · amazon web services · openstack · vmware

Runs on existing IaaS: vmware · openstack · Power Systems · System Z · IBM Spectrum · Dell, Cisco, NetApp, Lenovo, ...

IBM Cloud Private – Specific Use Cases

Airline

- Develop Chat app between tarmac personnel and pilot crews
- Makes use of on-prem APIs of flight logistics & scheduling info
- Requires low-latency
- Solution: Running microservices within containers on-prem

Industrial Client

- Requires standard deployment of software within factory environment
- Factories are geographically disperse and isolated with limited technical resources
- Solution: Leverage small ICP footprint with ability to synchronize catalogue content and approved workloads

Hospital

- Desire to leverage IBM Voice Gateway using on-prem environment for HA scalable deployment
- Analysis of text, roundtrip application
- Solution: Run IBM Voice Gateway and ICP for cloud native workloads

Bank

- Currently running main web portal on a self-managed (vanilla) Kubernetes deployment
- Need vendor to provide Kubernetes deployment in order to improve support and security posture
- Solution: Deploy ICP with full support of Kubernetes, Docker images, patch process, etc

Internal and Business Partner Use Only

IBM Cloud Private – Specific Use Cases

Use Case #1

Modernize and optimize existing applications

- Time to market acceleration
- Legacy or monolithic apps
- Existing WAS, MQ, DB2 infrastructure / migration
- DevOps initiatives and enterprise developers
- x86, Power and zLinux

Use Case #2

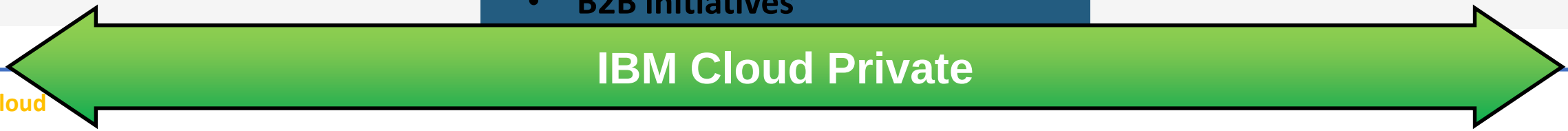
Opening up enterprise data centers to work with cloud services

- Securely open your datacenter
- GDPR
- API Economy
- Integrate public cloud services securely with your local cloud
- new web/mobile presence
- customer loyalty
- B2B initiatives

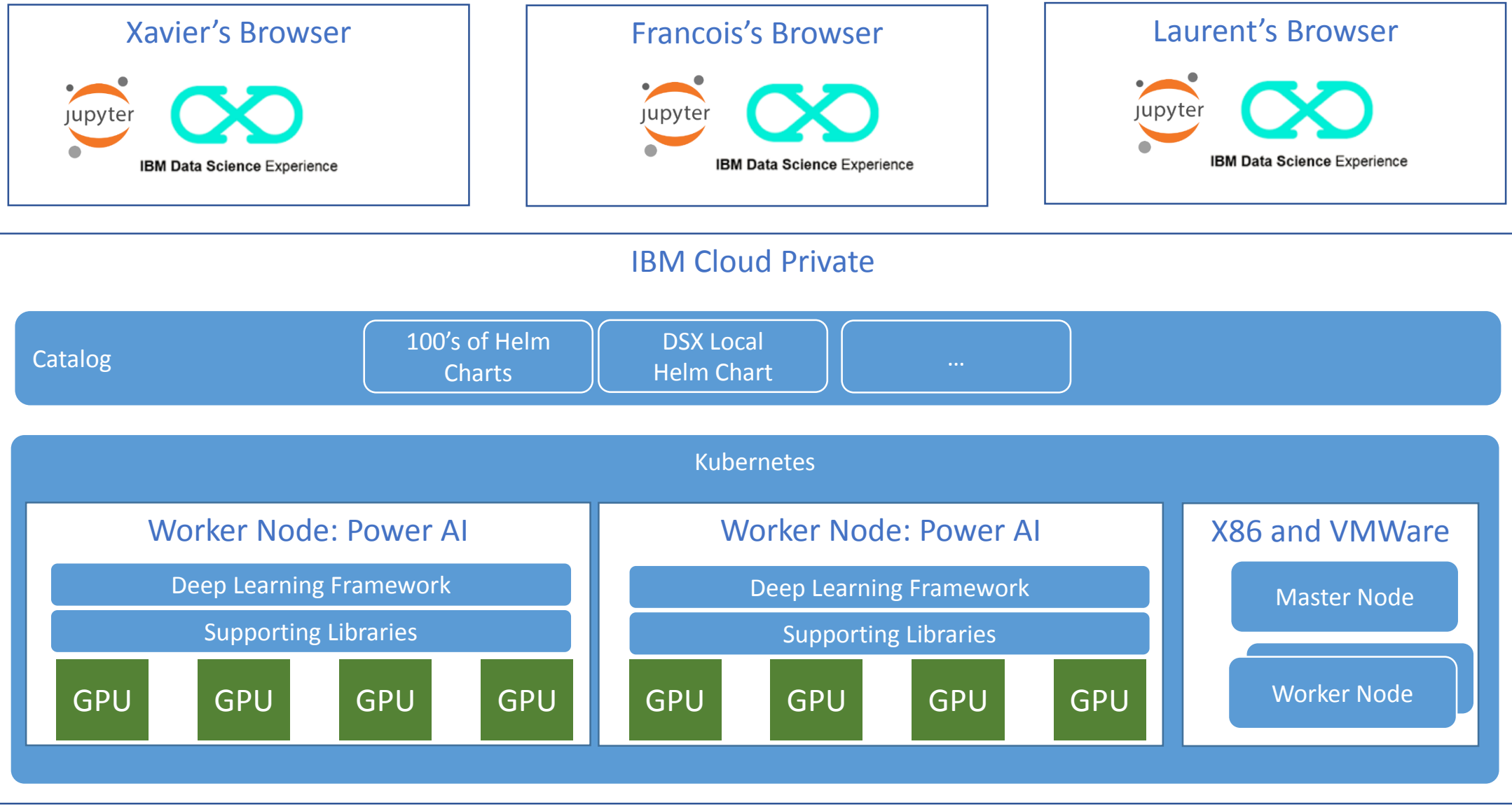
Use Case #3

Create new cloud-native applications

- New use cases
- IoT
- Blockchain
- Machine Learning
- Data science experience
- Building MicroServices



ICP Use case for Data Scientists



DSX Local Cluster & ICP – Internals (as of today: DSX 1.1.1.00 ppc64le)

Namespaces:
sysibmadm-data , sysibm-adm , dsxl-ml , ibm-private-cloud
PODs:
cloudant, redis, usermgmt, dsx-core, and ibm-nginx
Images:
27 images

Listing of key Components in DSX Local
(see under /wdp/k8s in the master node)

- **devtest-helpers** - Utility scripts to help with deployments
- **dsx-local-proxy** - the primary NGINX based server– serves up port 443 and reverse proxies to all other DSX Local service URLs
- **docker-registry** - Docker registry running as a Daemon Set in all hosts and service all needed docker images
- **cloudant-repo** - Cloudant repository database used to house metadata and projects etc.
- **redis-repo** - Redis in-memory Key value store – used for session storage in the web/UI micro services
- **swift-objectstore** - Openstack Swift container used to store csv data assets
- **usermgmt** - Supports management of users, authentication and working an external LDAP server
- **spark** - Spark cluster – master & worker daemon set
- **wdp-deploy-dashboard** – Backend and Front-end Admin components (IBM Data Platform Manager)
- **wdp-logs-elk** - Elastic Search, LogStash and Kibana – for Logging, Indexing
- **wdp-metrics-prometheus** - Monitoring metrics with Prometheus
- **dsx-local-k8s** - web-ui and api microservices (such as portal-main, projects api etc.)
- **docplexcloud-service** - Decision optimization / Deep Learning deployment

19 © 2017 IBM Corporation

Good news: ICP/K8s manages everything for you ☺

Prefix/Suffix	image.repository	image.tag
cloudantRepo	privatecloud-cloudant-repo	v3.13.428
dsxConnectionBack	dsx-connection-back	1.0.4
dsxCore	dsx-core	v3.13.10
dsxScriptedML	privatecloud-dsx-scripted-ml	v0.01.2
filemgmt	filemgmt	1.0.2
hdpzeppelinDsxD8a2ls2x	hdpzeppelin-dsx-d8a2ls2x	v1.0.10
jupyterDsxD8a2ls2x	jupyter-dsx-d8a2ls2x	v1.0.11
jupyterDsxD8a3ls2x	jupyter-dsx-d8a3ls2x	v1.0.7
jupyterGpuPy35	jupyter-gpu-py35	v1.0.9
mlOnlineScoring	privatecloud-ml-online-scoring	v3.13.6
mlPipelinesApi	privatecloud-ml-pipelines-api	v3.13.4
mllib	ml-libs	v3.13.30
nginxRepo	privatecloud-nginx-repo	v3.13.6
pipeline	privatecloud-pipeline	v3.13.3
portalMachineLearning	privatecloud-portal-machine-learning	v3.13.20
portalMlaas	privatecloud-portal-mlaas	v3.13.17
redisRepo	privatecloud-redis-repo	v3.13.431
repository	privatecloud-repository	v3.13.2
rstudio	privatecloud-rstudio	v3.13.8
spark	spark	1.5.1
sparkClient	spark-client	v1.0.2
sparkaasApi	sparkaas-api	v1.3.14
spawnerApiK8s	privatecloud-spawner-api-k8s	v3.13.5
usermgmt	privatecloud-usermgmt	v3.13.5
utilsApi	privatecloud-utils-api	v3.13.5
wmlBatchScoring	wml-batch-scoring	v3.13.2
wmlIngestion	privatecloud-wml-ingestion	v3.13.2

IBM Cloud Private Editions

Community

Platform
- Kubernetes (+ Helm)
- Core services
- Content catalog (Containers)

Freely Available in Docker Hub

Cloud Native

Platform
- Kubernetes (+ Helm)
- Core services
- Content catalog (Containers)

Cloud Foundry (Optional)

IBM Enterprise Software
- Microservice Builder
- WebSphere Liberty
- IBM SDK for node.js
- Cloud Automation Manager

Enterprise

Platform
- Kubernetes (+Helm)
- Core services
- Content catalog (Containers)

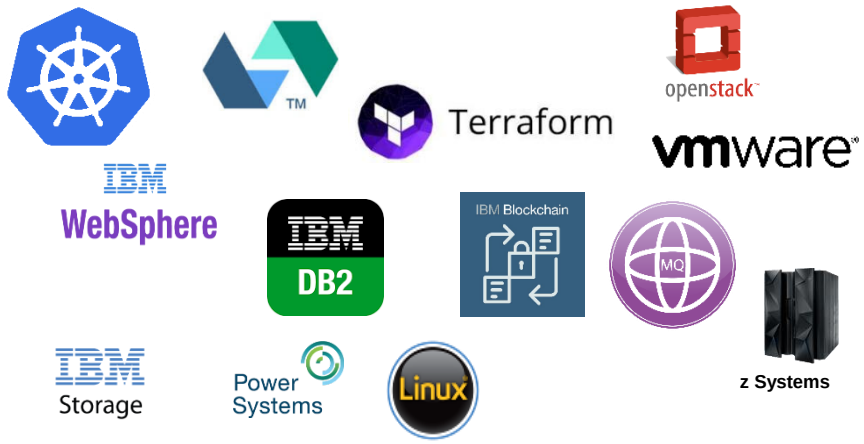
Cloud Foundry (Optional)

IBM Enterprise Software
Cloud Native Edition, plus:
+ WAS ND
+ MQ Advanced
+ API Connect Professional

37
IBM Cloud / © 2018 IBM Corporation

IBM Cloud Private v2.1 (Example on POWER)

Exceptional density, performance and economics for the next generation of business transformation and optimized cognitive services



- A rich stack of built-in tools and services for Developers and powerful enterprise-grade management tools for Operators
- Single Kubernetes-based platform to address new application development as well as optimize existing applications– supporting developer agility and operational simplicity
- Automate, deploy, scale and manage containerized applications across multiple architectures, delivering better performance, density and efficiency
- Built-in dashboards and analytics – simplifying operations and management
- Add-in services to connect APIs and data, monitor and manage events - delivering enhanced integration with critical enterprise applications

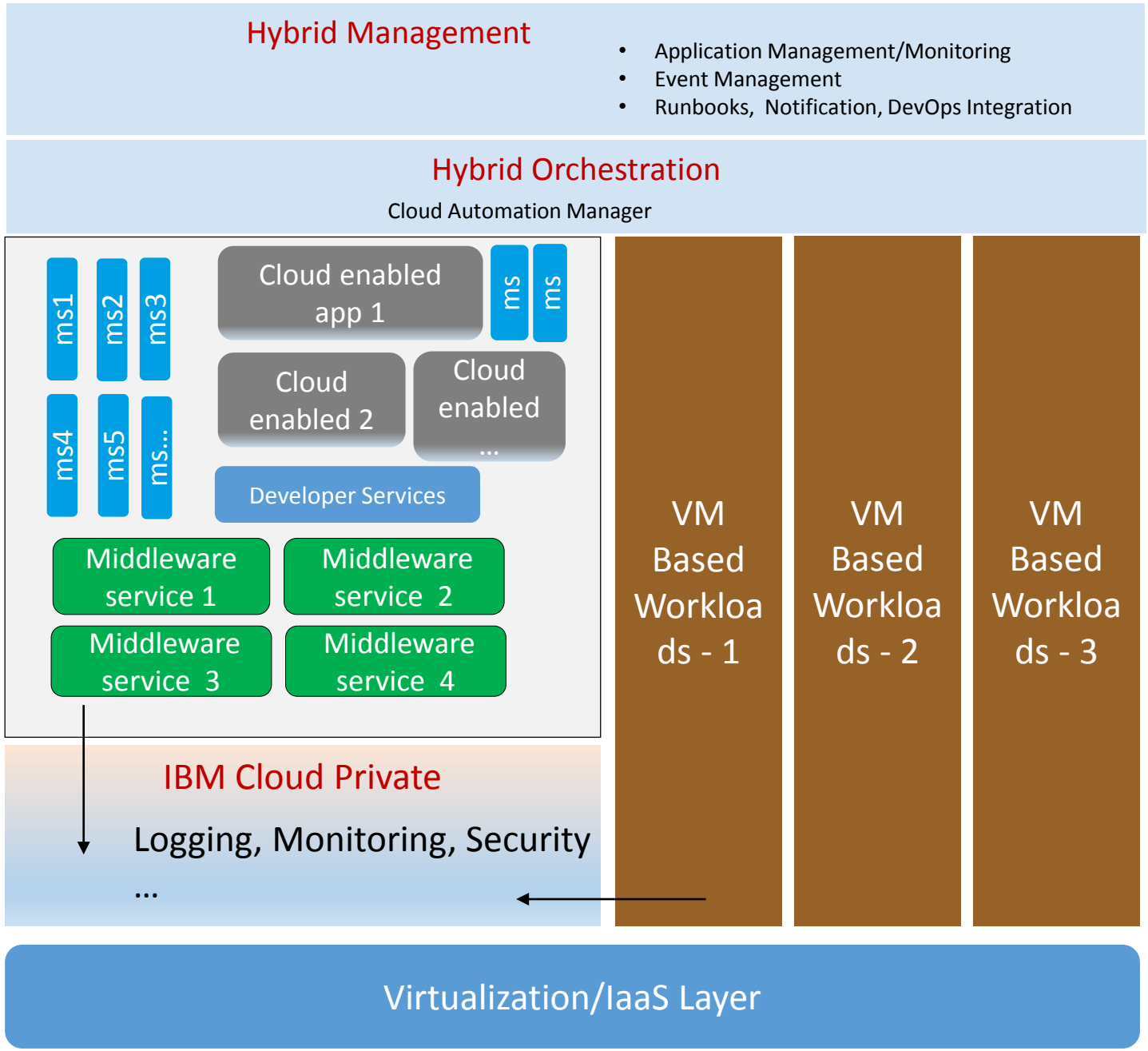
- Deploy in any LE Linux partition including PowerVM, KVM on Power and AHV with Nutanix
- Deliver better performance for data and cognitive services, i.e.:
  - ✓ 157% higher container density and 145% more throughput compared to x86 when running WAS Liberty*
  - ✓ Record setting speed and accuracy for deep learning training – 16 days down to 7 hours -- a 58x speedup!*

GA: 10/24/17 Announce: 11/1/17

Cloud Native	Data Science
Platform • Kubernetes • Core services (logging, etc) • Cloud Automation Manager • Content catalog	**Platform** • Kubernetes/Cloud Foundry • Core services (logging, etc) • Cloud Automation Manager • Content catalog
Bundled IBM Software • Microservice Builder • Liberty	**Bundled IBM Software** • DB2 Warehouse Enterprise Edition • Data Science Experience Enterprise Edition

IBM and Business Partner Confidential until Announce on 11/1/17

The Architect's view - Bringing it all together at the Enterprise level



Cloud Native application logic (Innovating)

- Microservices

Middleware Services (some IBM and some from the open source world)

- Each instance supports 1..n microservices
- IIB, DB2, Open Databases like Redis, Mongo, Messaging, API C, Datapower

Cloud Enabled (Modernizing)

- Monolithic applications made to run in containers
- Brought from WAS ND or WAS Base to Liberty

Add new function, expose APIs

VM Based Workloads

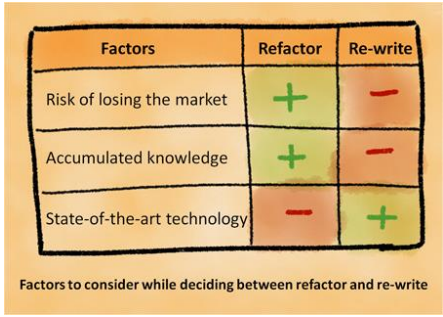
- WAS Base, WAS ND, BPM and others as necessary, CAM provisioned

All holistically plugged into existing Enterprise Facilities for Management, Monitoring and Security

IBM Cloud Private & IBM i

IBM i Apps & Cloud Native Apps

- ICP = based on Open Standards , for creating & deploying quickly new apps & new micro-services
- Need to complement this solution with a “DevOps” approach (automation, test..)
  - CI/CD : Microservice Builder, …
- On the IBM i side , a few Challenges :
  1. Existing Apps understanding
  2. Modernize my Apps, good practices for the existing & new ones
     - Rewriting vs. Refactoring
     - Data Centric Approach – Use Db2 for i & OS features,…
     - Modular Design vs. Architectural Monolith
     - **DevOps** Approach– Toolchain CI/CD (mandatory?) integrating your **IBM i environment**.
  3. IBM i Integration with Cloud Native Apps
     - Open Source Tools & frameworks on IBM i.
     - Expose your apps & data on IBM i via standards - Web Services & API / Microservices on IBM i



Factors to consider while deciding between refactor and re-write

IBM i Apps & Cloud Native Apps

- Why a modular approach / micro services on IBM i ?
  - Collaborative devs & application integration made easier
  - Time saver for innovative projects & frequent apps changes & enhancements
  - Relevant for pure IBM i devs (RPG ILE, Java, Node.js, PHP) or hybrid (IBM i + Cloud – SoE – SoR).

- Part of a global **DevOps** methodology & mindset **, for any Cloud Apps (SoE) & IBM i Apps**
  → requires a DevOps mindset…and appropriate tooling : **Toolchain, Delivery Pipeline**
  → IBM Solutions , Open Source & Third Party

IBM i Apps & Cloud Native Apps

Architectural Evolution

Spaghetti Architecture

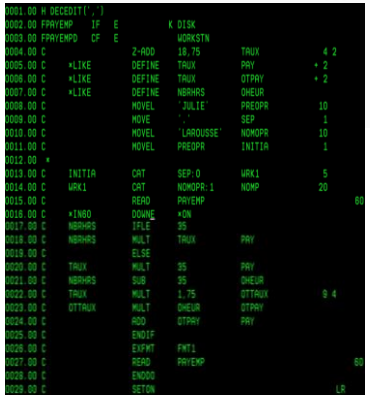
Cut & Paste
(1990's)

Lasagna Architecture

Layered Monolith
(2000's)

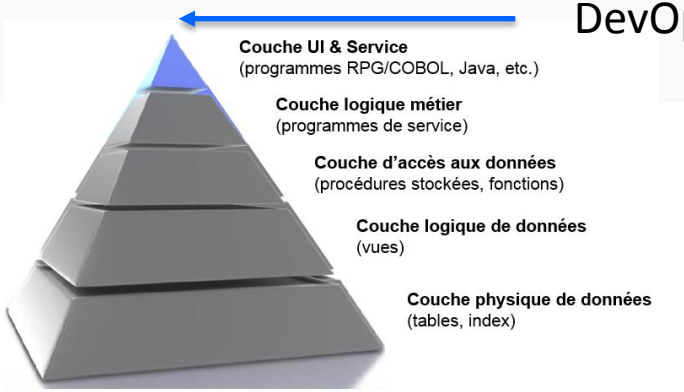
Ravioli Architecture

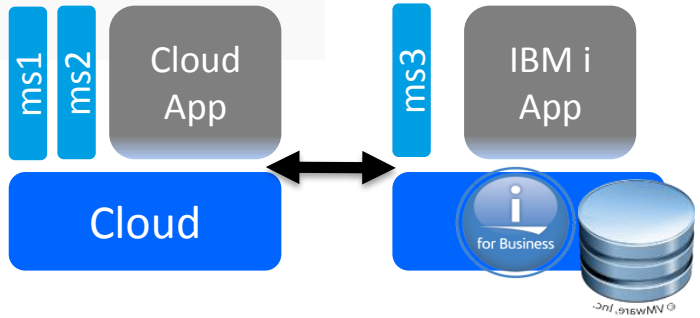
Microservices
(2010's)



App Centric Monolith,
Single Program

DevOps Ready





Data Centric, Modular & Layered, Modern Techno, Design Patterns (MVC…)

IBM i Apps & Cloud Native Apps

Everything is ready for building Cloud Native & Microservices solutions integrated with any IBM i applications:

- ❑ New languages & frameworks on IBM i : RPG Free, Python, Ruby, Node.js and many others (.NET) etc.

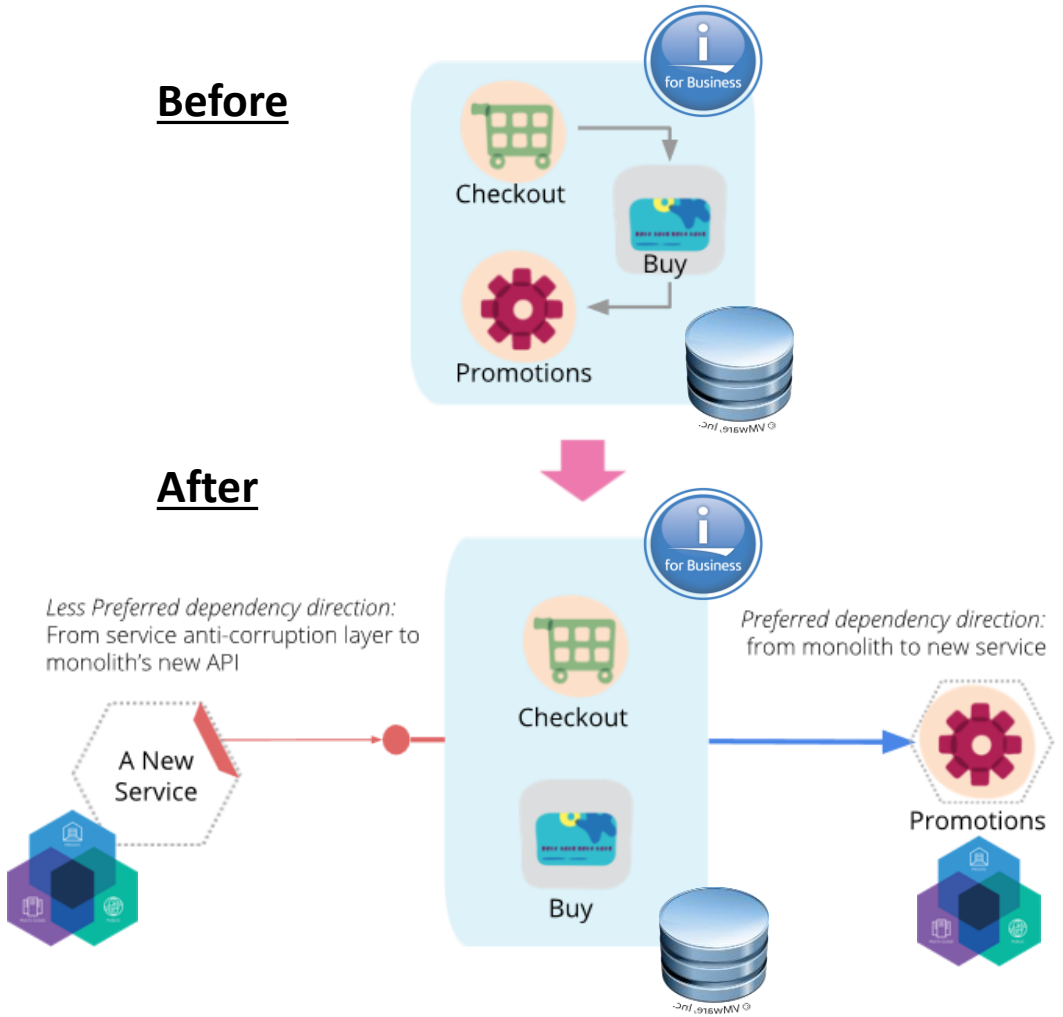
- ❑ Out of the box Integration technologies available on IBM i
  - Integrated Web Service Server (WebSphere Liberty)
  - Integrated Application Server (WebSphere Liberty)
  - Open Source frameworks (Node.js, NGINX) with native access to objects & the database

IBM i Apps & Cloud Native Apps

Everything is ready for building Cloud Native & Microservices solutions integrated with any IBM i applications:

- ❑ New languages & frameworks on IBM i : RPG Free, Python, Ruby, Node.js and many others (.NET) etc.

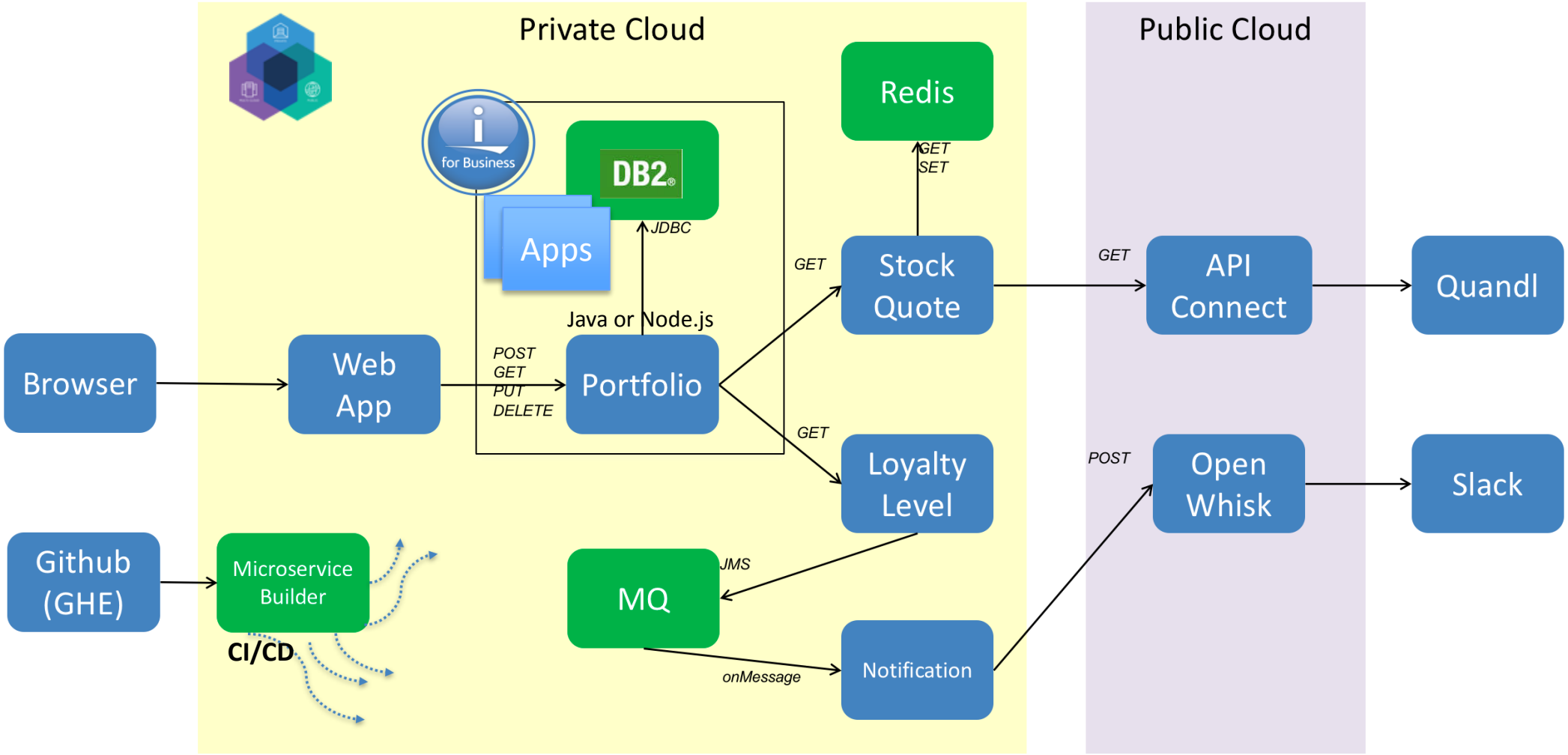
- ❑ Out of the box Integration technologies available on IBM i
  - Integrated Web Service Server (WebSphere Liberty)
  - Integrated Application Server (WebSphere Liberty)
  - Open Source frameworks (Node.js, NGINX) with native access to objects & the database



IBM i Apps & Cloud Native Apps

micro-services Application w/ IBM I integration

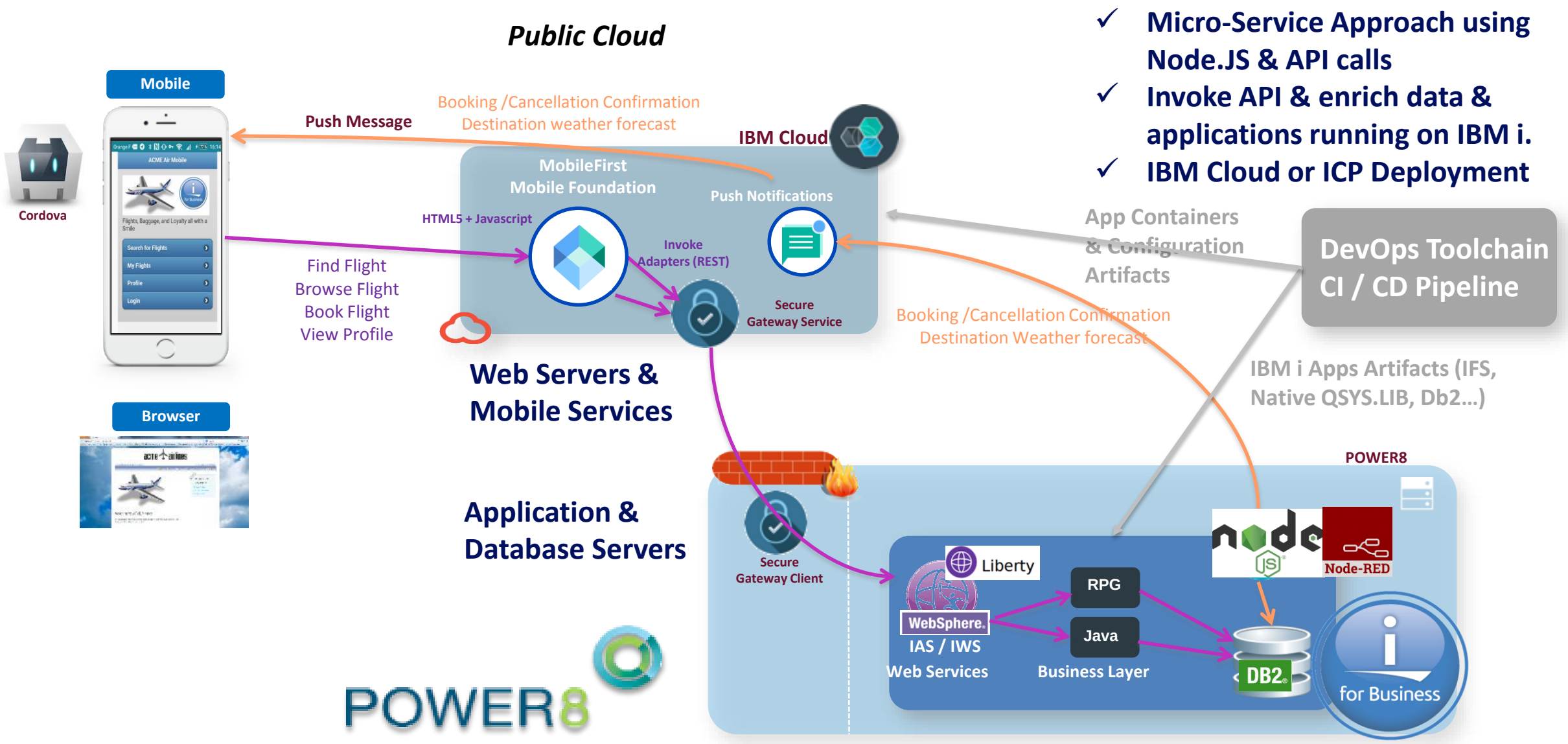
Stock Trader App



Original source: https://github.com/IBMStockTrader/

Example: Develop new Mobile Services on IBM i with IBM Cloud

Hybrid Application – DevOps & Microservices

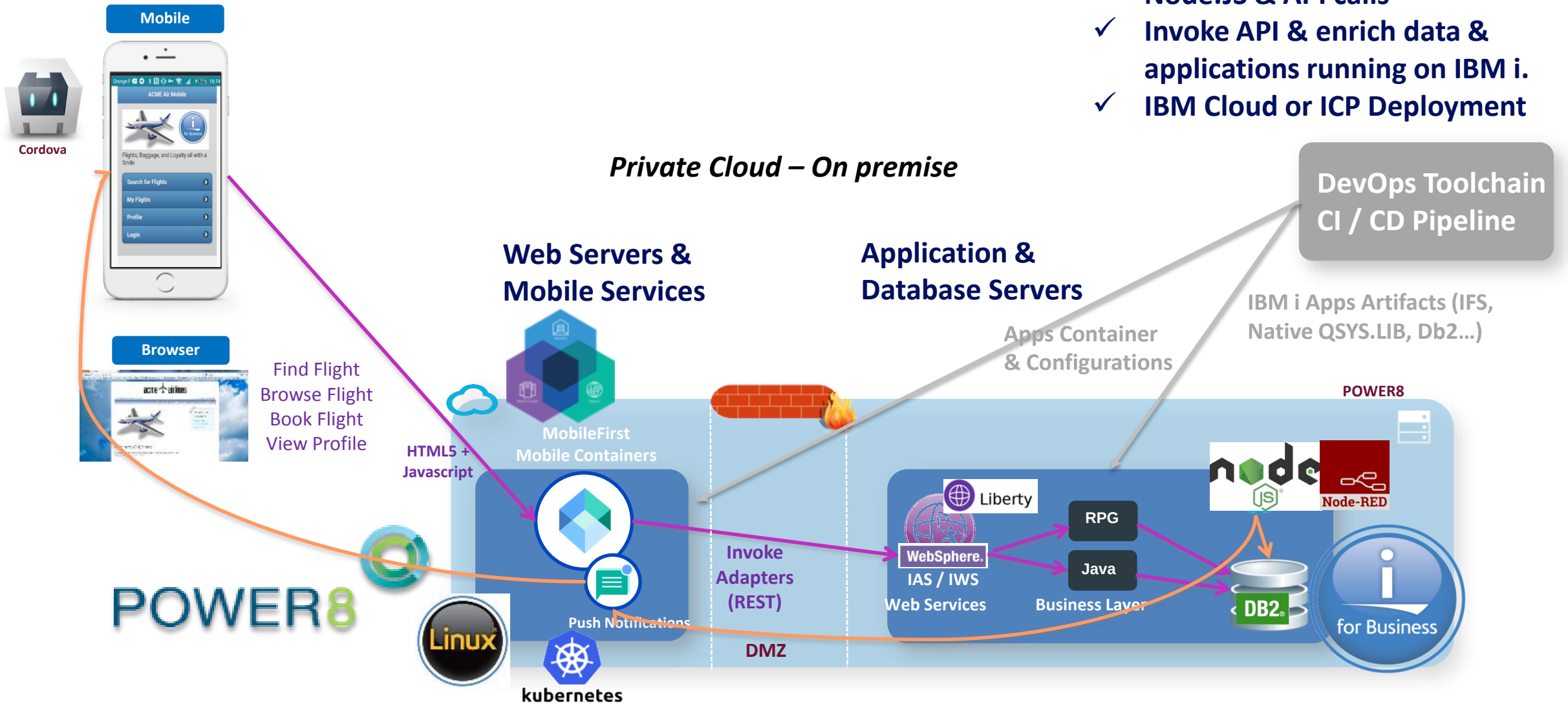


- ✓ Micro-Service Approach using Node.JS & API calls
- ✓ Invoke API & enrich data & applications running on IBM i.
- ✓ IBM Cloud or ICP Deployment

Example: Develop new Mobile Services on IBM i with IBM Cloud

Hybrid Application – DevOps & Microservices

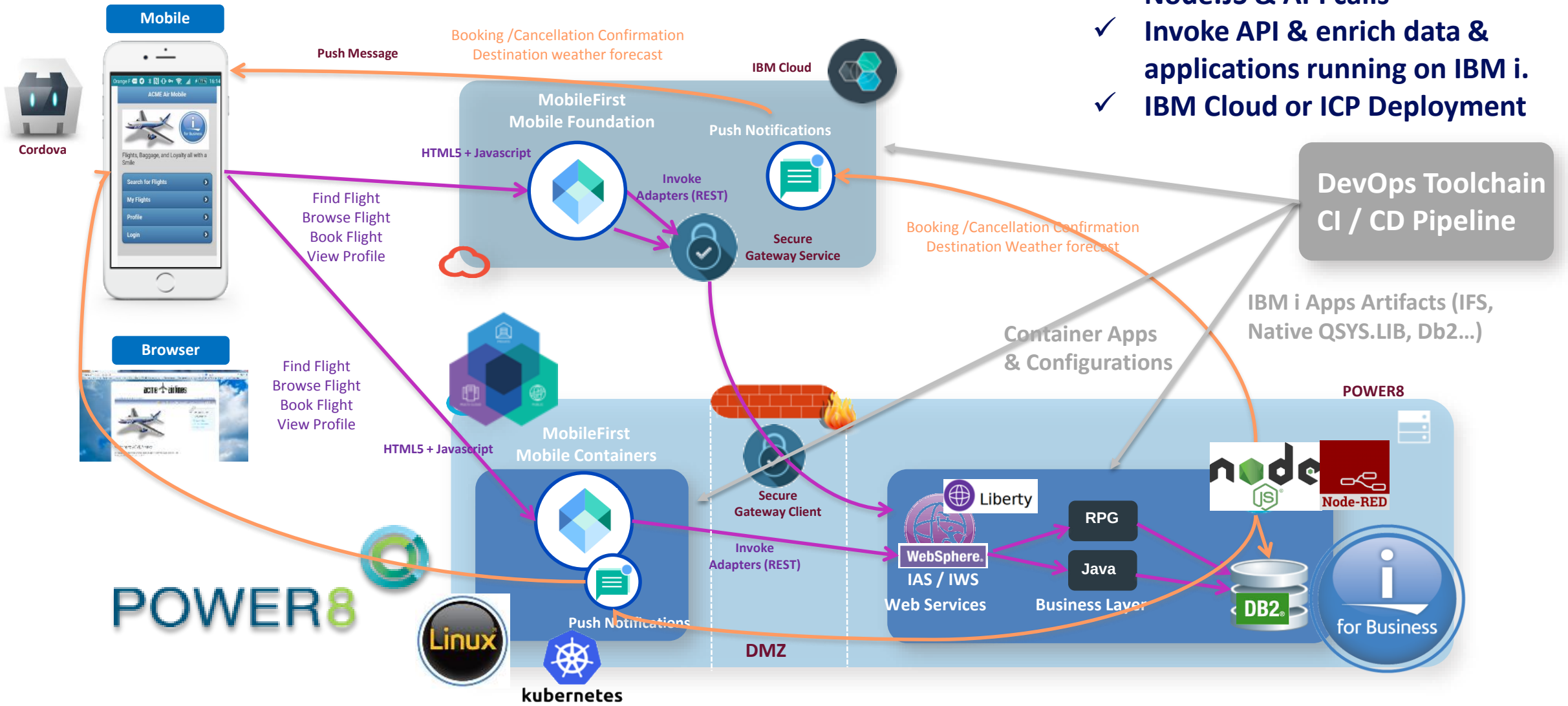
- ✓ Micro-Service Approach using Node.JS & API calls
- ✓ Invoke API & enrich data & applications running on IBM i.
- ✓ IBM Cloud or ICP Deployment



Example: Develop new Mobile Services on IBM i with IBM Cloud

Hybrid Application – DevOps & Microservices

- ✓ Micro-Service Approach using Node.JS & API calls
- ✓ Invoke API & enrich data & applications running on IBM i.
- ✓ IBM Cloud or ICP Deployment

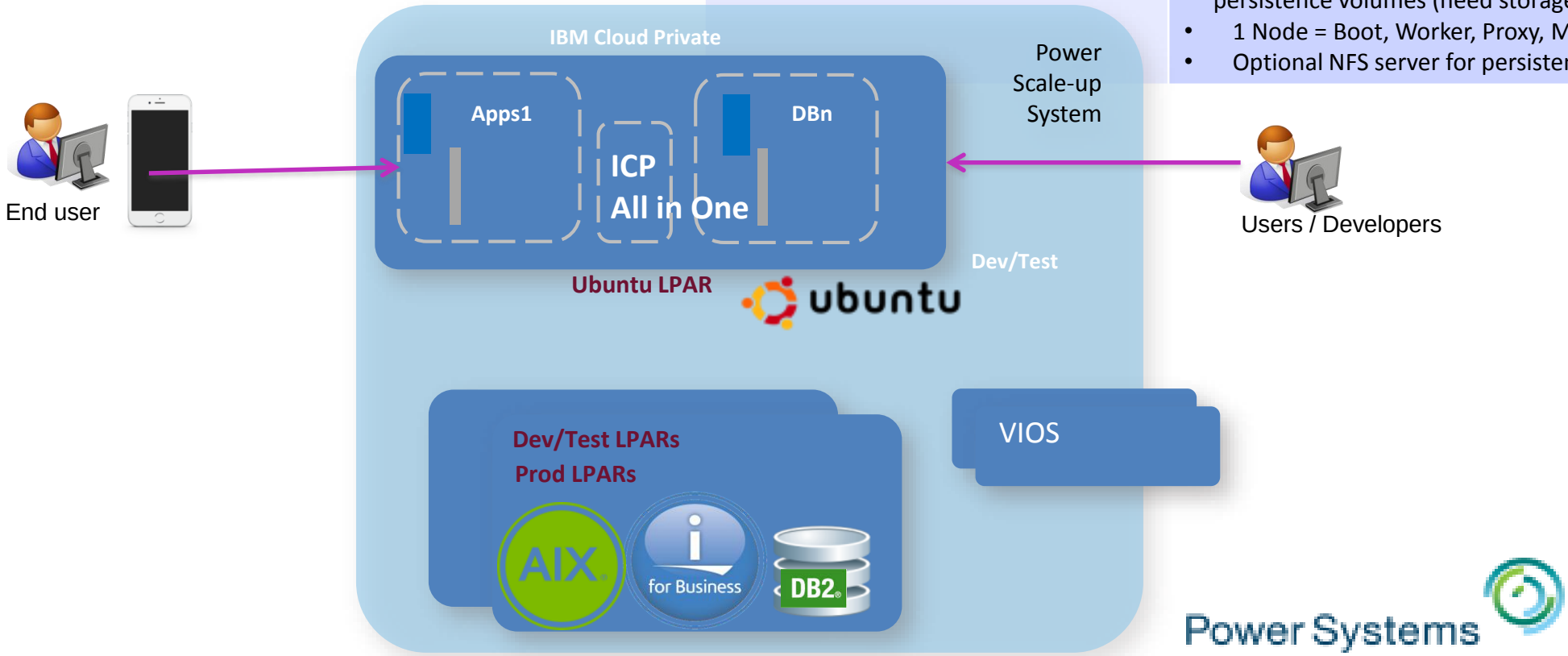


How to get Started?

Starter configuration – Single node “All in one”

- **CE (Free) Edition. For Dev/Test, no HA.**
- Cloud or Enterprise Editions = catalog++
- ICP on PowerVM (LPAR) or 1 Scale-out L / LC / CS System (VM)
- Cloud Foundry on x86 nodes only (1H 2018)

Current	Target*
IBM i / AIX / Linux LPARs production	IBM i / AIX / Linux LPARs production
IBM i / AIX / Linux LPARs dev/test	IBM i / AIX / Linux LPARs dev/test
	• **ICP 3.1 on Power Community Edition** • **1 Node: LPAR Ubuntu 16.04 LTS** • **8 core / 16GB / disk 200+ GB** Documentation: 4 cores / 8GB • SAN Storage or internal disks • hosted by VIOS (SAN/LAN), NFS Server for persistence volumes (need storage) • 1 Node = Boot, Worker, Proxy, Master • Optional NFS server for persistence



To go further…

Multi-VM config with HA – Config Example

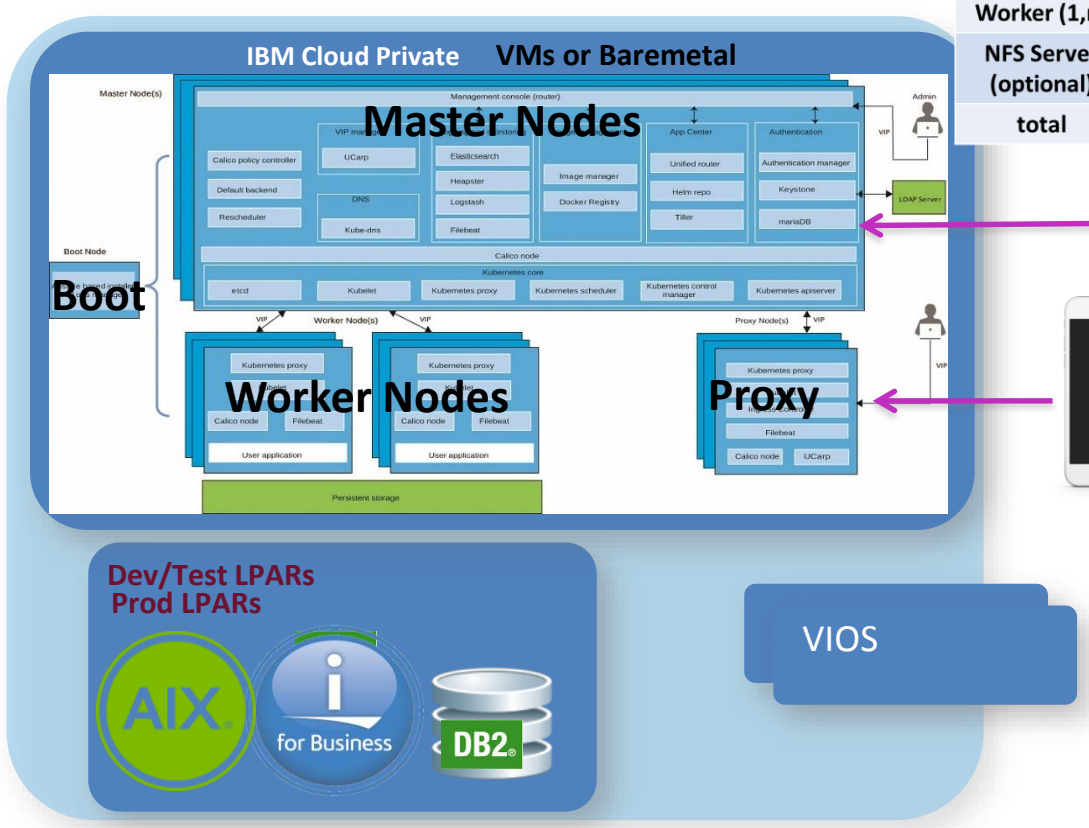
- ICP on PowerVM (LPARs) / Scale-out Systems / Nutanix HCI
- HA Mode for Master & Proxy nodes

Note: Worker nodes can support mixed architectures. You can add worker nodes into a single cluster that run on Linux™ 64-bit, Linux™ on Power® 64-bit LE and IBM® Z platforms.

- Persistence:
  - NFS v4
  - Prefer Glusterfs
- Node = VM (LPAR or KVM VM) or Baremetal
- Example on PowerVM:
  - 11 nodes on 11 LPARs
- Example on L / CS / LC servers:
  - 11 nodes on 4 servers

Current	Target*
IBM i / AIX / Linux LPARs production	IBM i / AIX / Linux LPARs production
	ICP 3.1 on Power - Cloud or Enterprise Edition **11 x OS: Ubuntu 16.04 LTS or RHEL 7.4+**

	Minimal 4 Node Config				11 Node Config Example			
Role	#Nodes	CPU (VP)	Mem (GB)	Disk (GB)	#Nodes	CPU (VP)	Mem (GB)	Disk (GB)
Boot (1)	1	1	4	100	1	2	8	80
Master(1,3,5)	1	1	4	150	3	4	16	60
Proxy (1,3,5)	1	2	4	50	3	2	4	60
Management (1, optional)	0	4	8	100	1	8	16	150
Worker (1,n)	1	1	4	100	3	2	8	100
NFS Server (optional)					1	2	8	1024
total	4 nodes	**5 VP**	**16 GB**	**400 GB**	11 nodes	**32 VP**	**116 GB**	**900 GB**




Developers


End Users

ubuntu

Power Systems

Demo

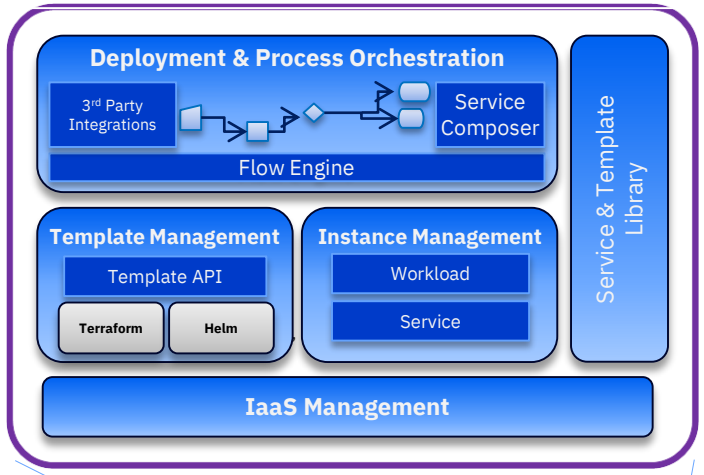
IBM **Cloud** Automation Manager on IBM **Cloud** Private

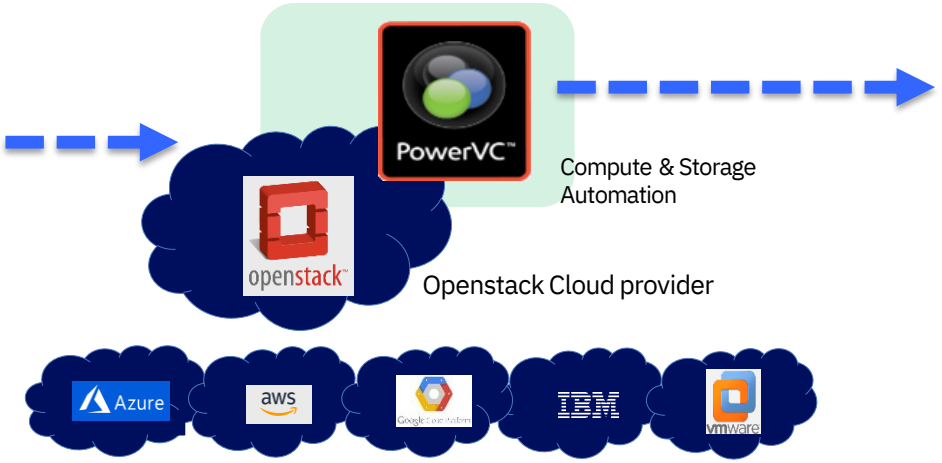
Value Proposition for Enterprise Power Systems & IBM i :

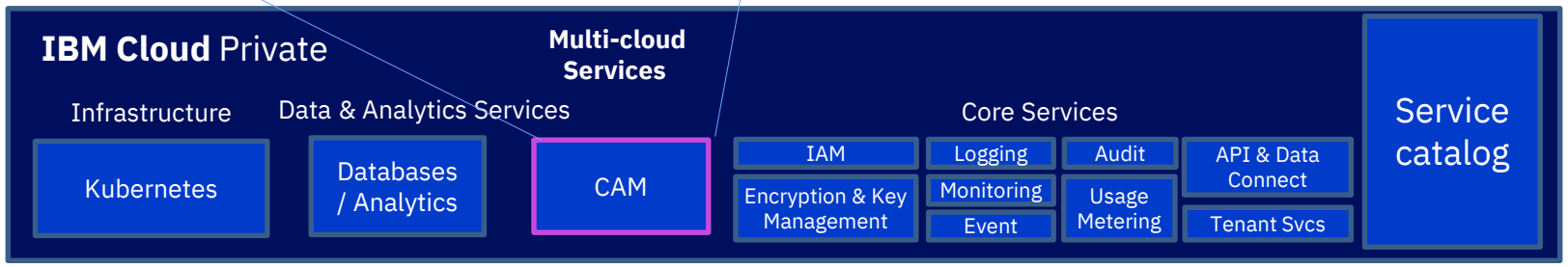
→ Optimize developer's need for speed with organization's need for governance

Containerized cloud native application
CAM is installed into an IBM **Cloud** Private worker node with a Helm chart
Leverages IBM **Cloud** Private services for enterprise capabilities

IBM **Cloud** Automation Manager







thirty years

OSDB @ Montpellier Cognitive Systems Lab

Positioning IBM Power Systems at the heart of the Cognitive Era

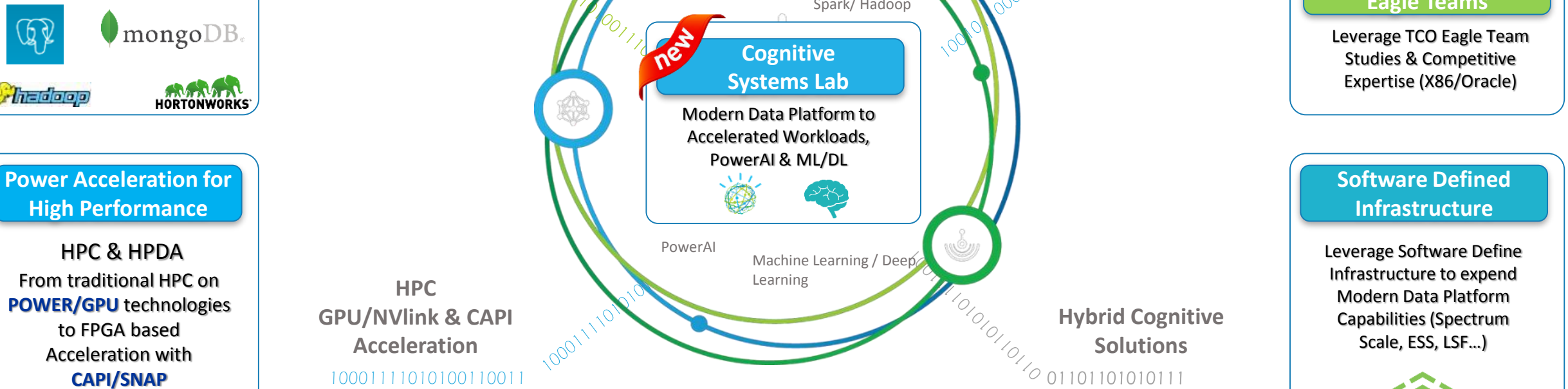
Power System Linux Center

Engage with Clients, ISVs & Partners to leverage the benefits of Linux on Power, using **Open Sources** Solutions

Power Acceleration for High Performance

HPC & HPDA

From traditional HPC on **POWER/GPU** technologies to FPGA based Acceleration with **CAPI/SNAP**



Power Acceleration for ISV's

Leverage OpenSource Databases in the Competition with Oracle.

Competitive & TCO Eagle Teams

Leverage TCO Eagle Team Studies & Competitive Expertise (X86/Oracle)

Software Defined Infrastructure

Leverage Software Define Infrastructure to expend Modern Data Platform Capabilities (Spectrum Scale, ESS, LSF…)

What we will deliver

Highly Technical Skills & Hardware (S822LC for HPC & Big Data) available

Talk & Demonstrate	Exploration & Design Workshops	PoT/ POC/Benchmarks	Advanced Technical Support