RED HAT SUMMIT

From cloud to edge:

Why cloud-native application development matters in supporting IoT

David Bericat
Chief Architect Data Analytics (AI/ML) • Edge • IoT
Global Partner Solutions
Red Hat

Erik Jacobs
Principal Technical Marketing Manager, OpenShift
Cloud Platforms BU
Red Hat

Thursday May 9th

THE INTERNET IS ALL THINGS

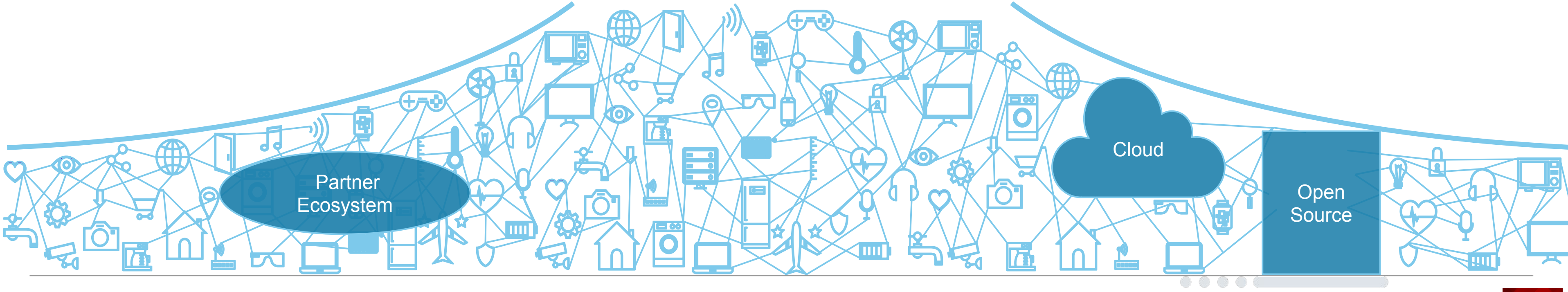
YOU SIMPLY DON'T REALIZE IT YET...

AGENDA

- What has Red Hat been doing in IoT
- From IoT to end-to-end Analytics (AI/ML) to Edge computing
- Red Hat’s next generation cloud-computing development
- Dreaming exercise: serverless & FaaS at the “edge”

RED HAT'S IoT STRATEGY OVERVIEW

- Customers use our parts of our portfolio in a variety of ways to address IoT challenges on a hybrid cloud infrastructure.
- We do not claim to provide an IoT platform. We work with our commercial partner ecosystem to define more complete solutions and bring them to market.
- We take a leadership role in open source communities to build IoT technology.



KEY FUNCTIONALITY

FOR AN END-TO-END IoT ARCHITECTURE

Device Management & Connectivity

Securely connect, authenticate and manage disparate connected devices that speak different protocols

Intelligent Edge Processing & Analytics

Apply analytics at the edge with machine learning and business rules to enable local, low-latency decision making

Advanced Analytics & Machine Learning

Centralize IoT data processing, analytics and machine learning to enable deep business insights and actionable intelligence

Business & Application Integration

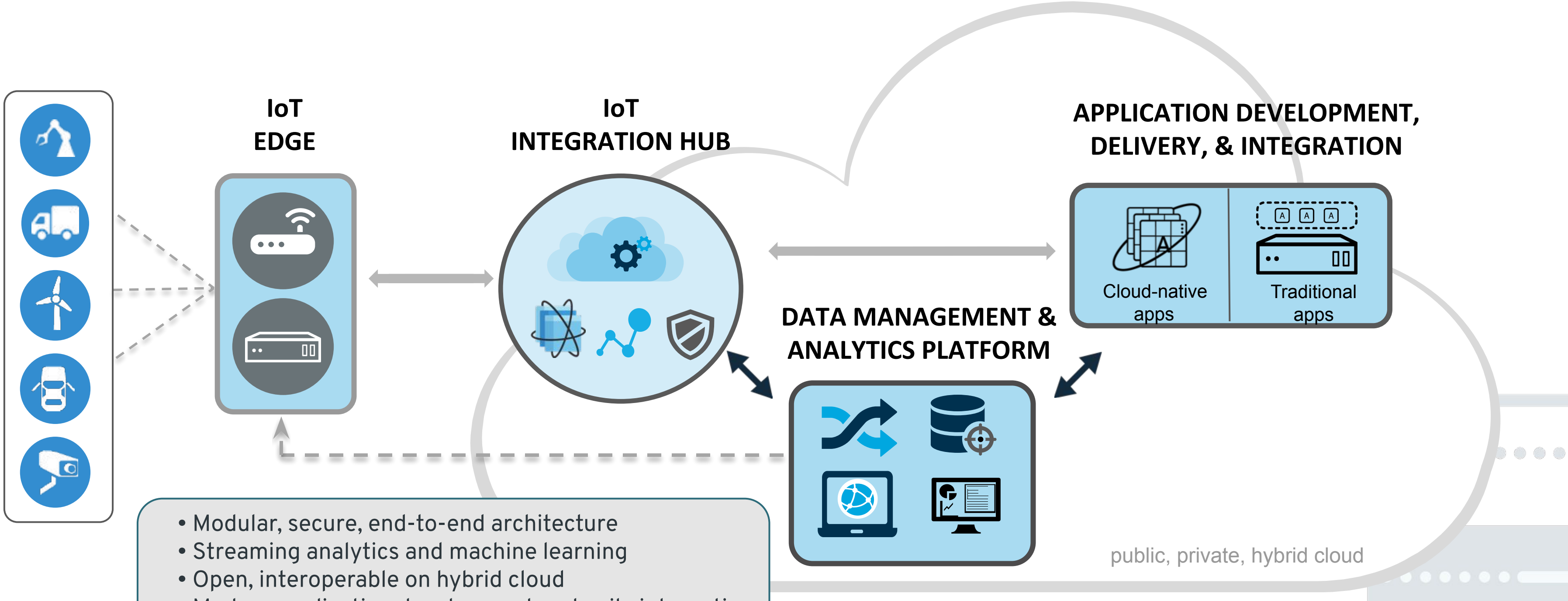
Enable integration with enterprise and business applications to bridge the gap between OT and IT and reduce complexity

End-to-End Security & Compliance

Tools to enable end-to-end data security, compliance, authorization and authentication

OPEN END-TO-END IoT ARCHITECTURE

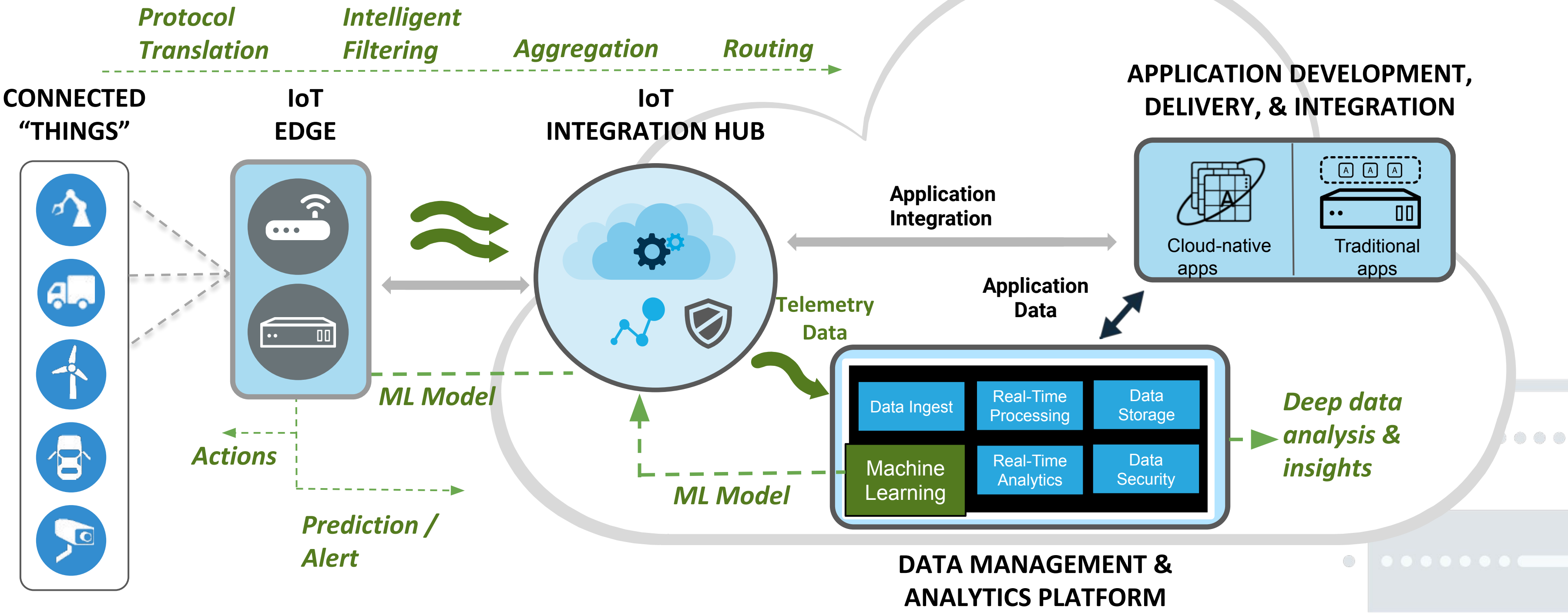
INTEGRATING OT, IT , DATA MANAGEMENT, ANALYTICS &, APPLICATIONS



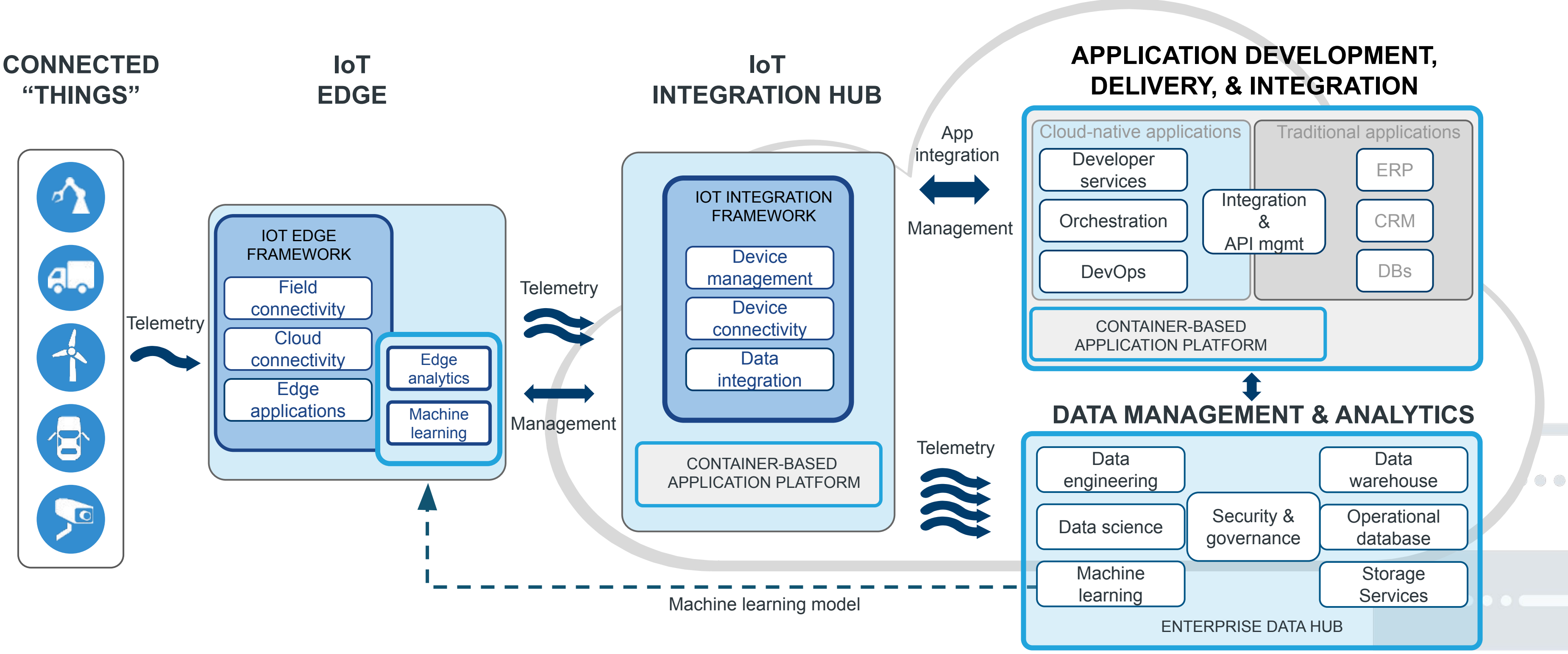
- Modular, secure, end-to-end architecture
- Streaming analytics and machine learning
- Open, interoperable on hybrid cloud
- Modern application development and agile integration

END-TO-END ANALYTICS

PUSHING AI/ML FROM CORE TO EDGE



FUNCTIONALITY OVERVIEW



CONNECTED "THINGS"

IoT EDGE

IoT INTEGRATION HUB

APPLICATION DEVELOPMENT, DELIVERY, & INTEGRATION

Telemetry

IOT EDGE FRAMEWORK

Field connectivity

Cloud connectivity

Edge applications

Edge analytics

Machine learning

Telemetry

Management

IOT INTEGRATION FRAMEWORK

Device management

Device connectivity

Data integration

CONTAINER-BASED APPLICATION PLATFORM

App integration

Management

Cloud-native applications

Developer services

Orchestration

DevOps

Integration & API mgmt

Traditional applications

ERP

CRM

DBs

CONTAINER-BASED APPLICATION PLATFORM

DATA MANAGEMENT & ANALYTICS

Telemetry

Data engineering

Data science

Machine learning

Security & governance

Data warehouse

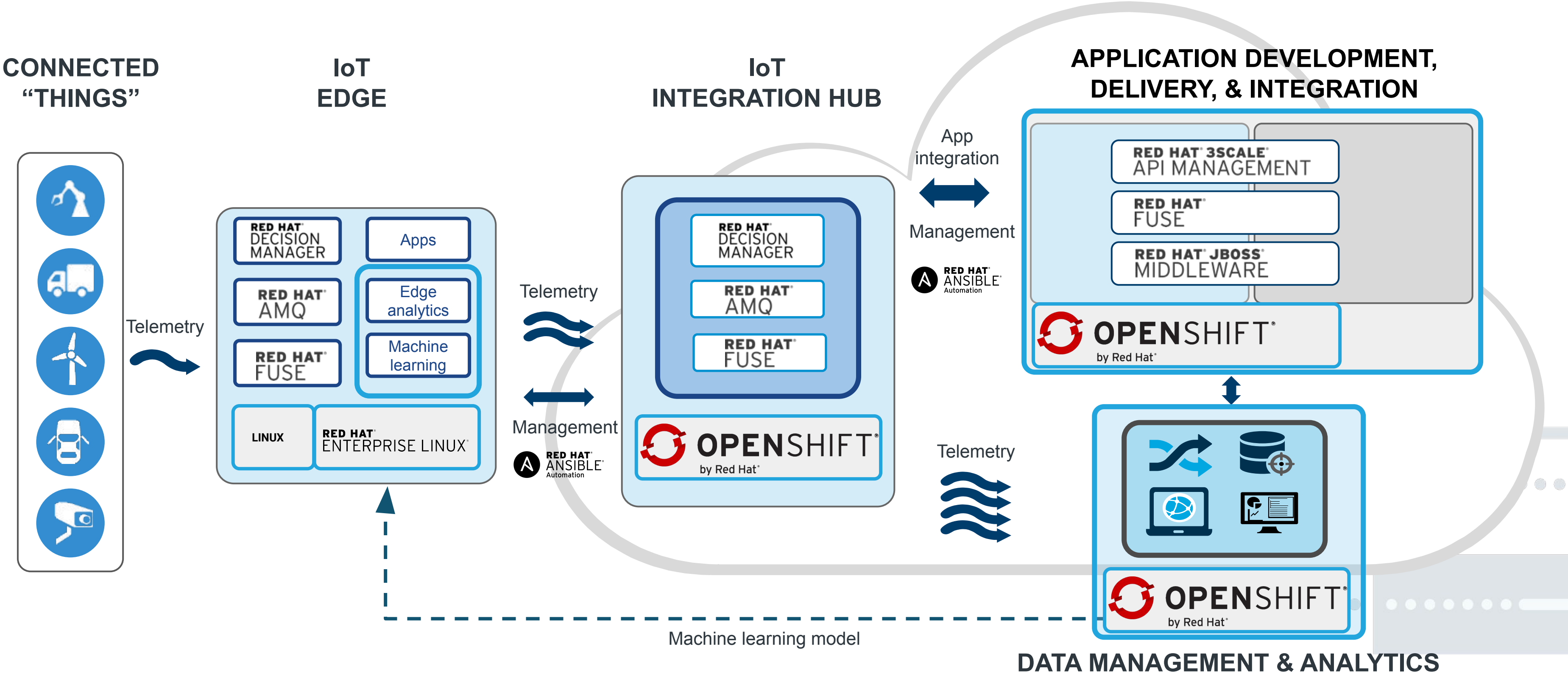
Operational database

Storage Services

ENTERPRISE DATA HUB

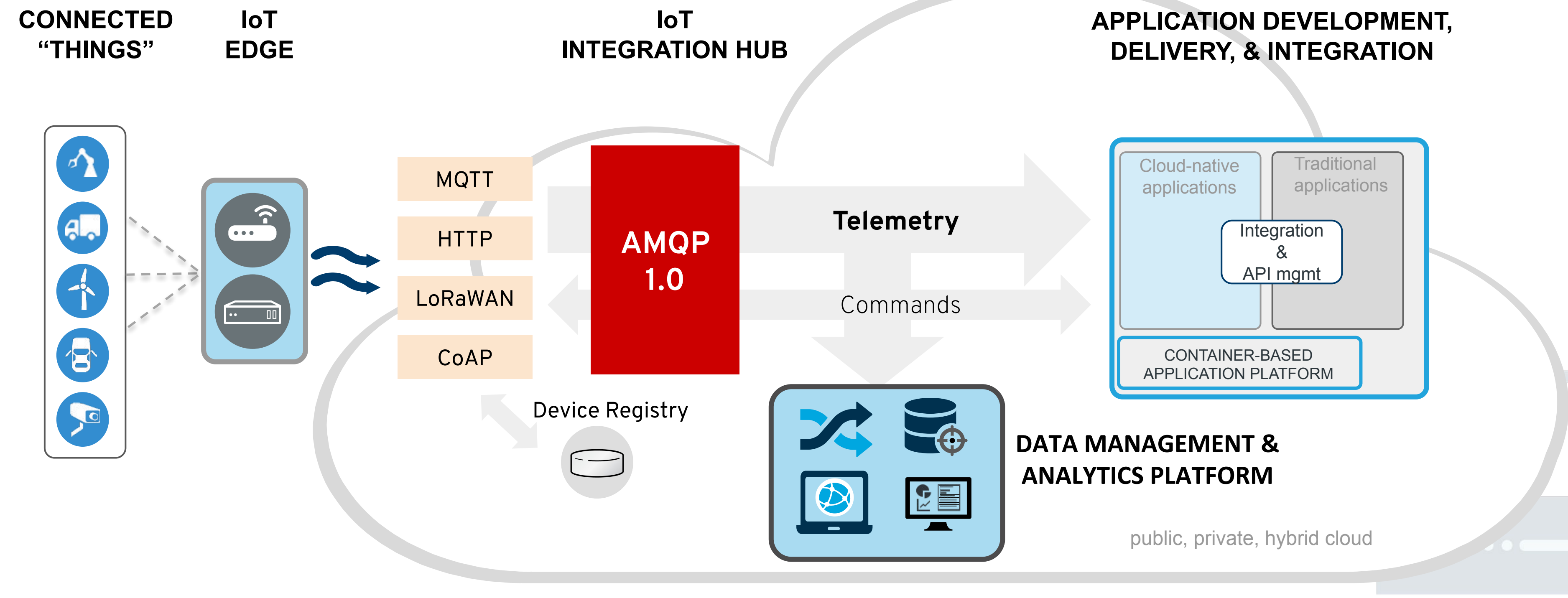
Machine learning model

HOW RED HAT PORTFOLIO CAN HELP



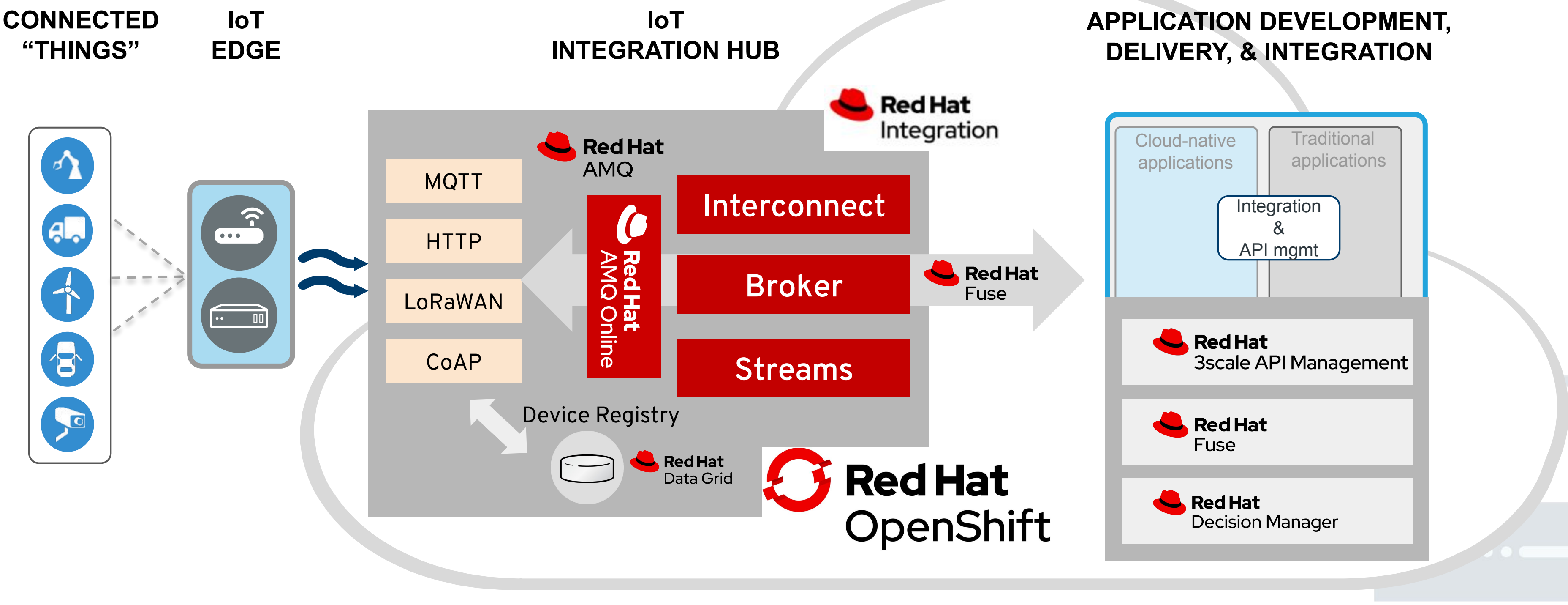
AGILE INTEGRATION

STANDARDIZING THE “WILD, WILD WEST”

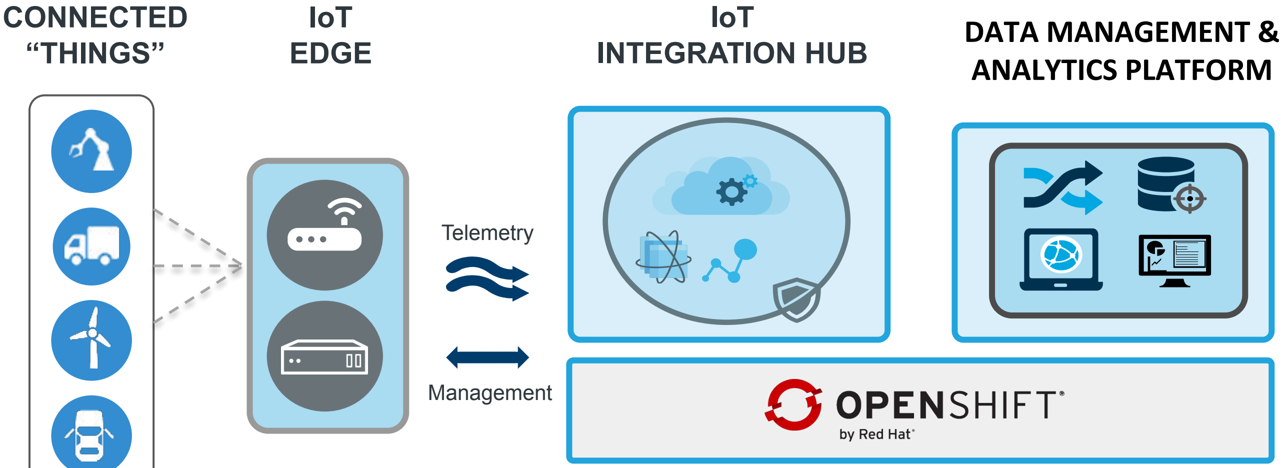

CONNECTED “THINGS”
IoT EDGE
IoT INTEGRATION HUB
APPLICATION DEVELOPMENT, DELIVERY, & INTEGRATION
MQTT
HTTP
LoRaWAN
CoAP
AMQP 1.0
Telemetry
Commands
Device Registry
Cloud-native applications
Traditional applications
Integration & API mgmt
CONTAINER-BASED APPLICATION PLATFORM
DATA MANAGEMENT & ANALYTICS PLATFORM
public, private, hybrid cloud

AGILE INTEGRATION

STANDARDIZING THE “WILD, WILD WEST”



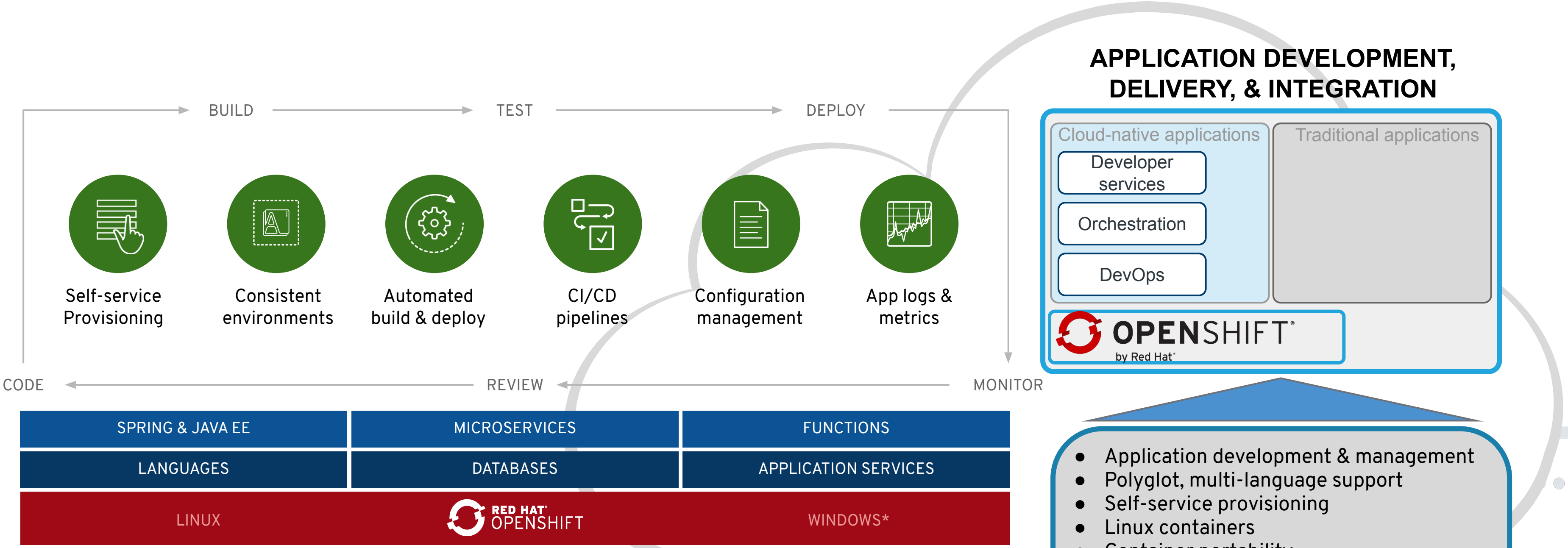
PORTABILITY ACROSS ANY INFRASTRUCTURE



Container	Container	Container	Container	Container
IoT Services	IoT Services	IoT Services	IoT Services	IoT Services
OS dependencies	OS dependencies	OS dependencies	OS dependencies	OS dependencies
RHEL	RHEL	RHEL	RHEL	RHEL
Guest VM		Virtual Machine	Virtual Machine	Virtual Machine
LAPTOP	BARE METAL	VIRTUALIZATION	PRIVATE CLOUD	PUBLIC CLOUD

- Linux containers
- Kubernetes-based orchestration
- Deployment flexibility
- Automation
- Scalability
- Persistent storage

APPDEV & CONTINUOUS DELIVERY



BUILD → TEST → DEPLOY

Self-service Provisioning

Consistent environments

Automated build & deploy

CI/CD pipelines

Configuration management

App logs & metrics

CODE ← REVIEW ← MONITOR

SPRING & JAVA EE	MICROSERVICES	FUNCTIONS
LANGUAGES	DATABASES	APPLICATION SERVICES
LINUX	RED HAT OPENSHIFT	WINDOWS*

APPLICATION DEVELOPMENT, DELIVERY, & INTEGRATION

Cloud-native applications

- Developer services
- Orchestration
- DevOps

Traditional applications

OPENSHIFT by Red Hat

- Application development & management
- Polyglot, multi-language support
- Self-service provisioning
- Linux containers
- Container portability
- Deployment flexibility
- Kubernetes-based orchestration
- Automation
- Scalability
- Persistent storage

* coming soon

APPDEV WITH CODEREADY WORKSPACES

PLANNING, CREATING, AND DEPLOYING HYBRID CLOUD SERVICES

The collaborative OpenShift-Native IDE. Free for any customer of OpenShift Dedicated or OpenShift Container Platform.

RED HAT® CODEREADY

Container Workspaces

Workspace replicas to end "works on my machine" and enable team collaboration.

DevOps Integrations

Reference developer workspaces from any issue, failed build, or git notification.

Protect Source Code

Full access to source code without any of it landing on hard-to-secure laptops.

Based on the open Eclipse Che project	Red Hat Linux and Application Infrastructure	Plugin model for extensibility	Serverless support (coming later)

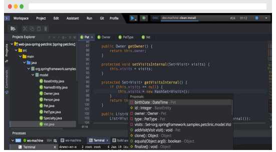
Use It To: Replace VDI for devs, and enable true container-based DevOps.

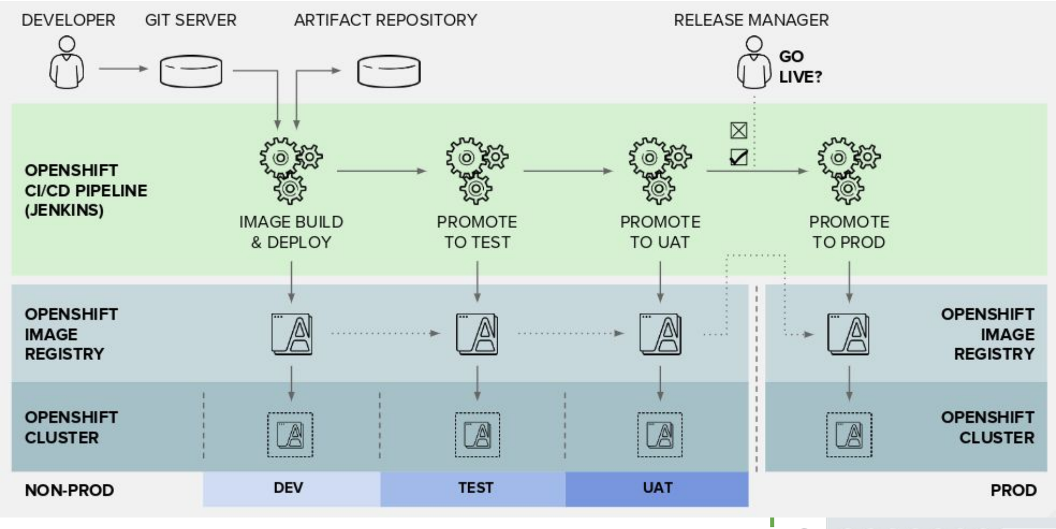
EDGE CONTINUOUS DEPLOYMENTS

1. Create App / microservice / function

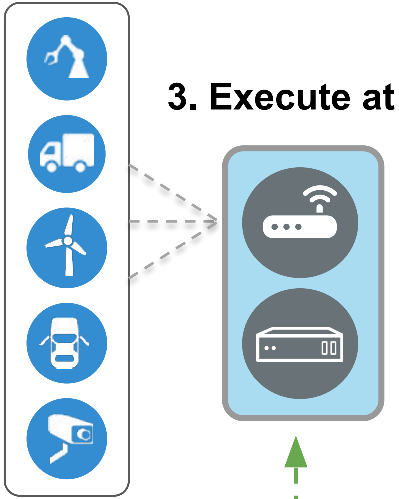
Who: Developers

Where: CodeReady Workspaces





3. Execute at the edge

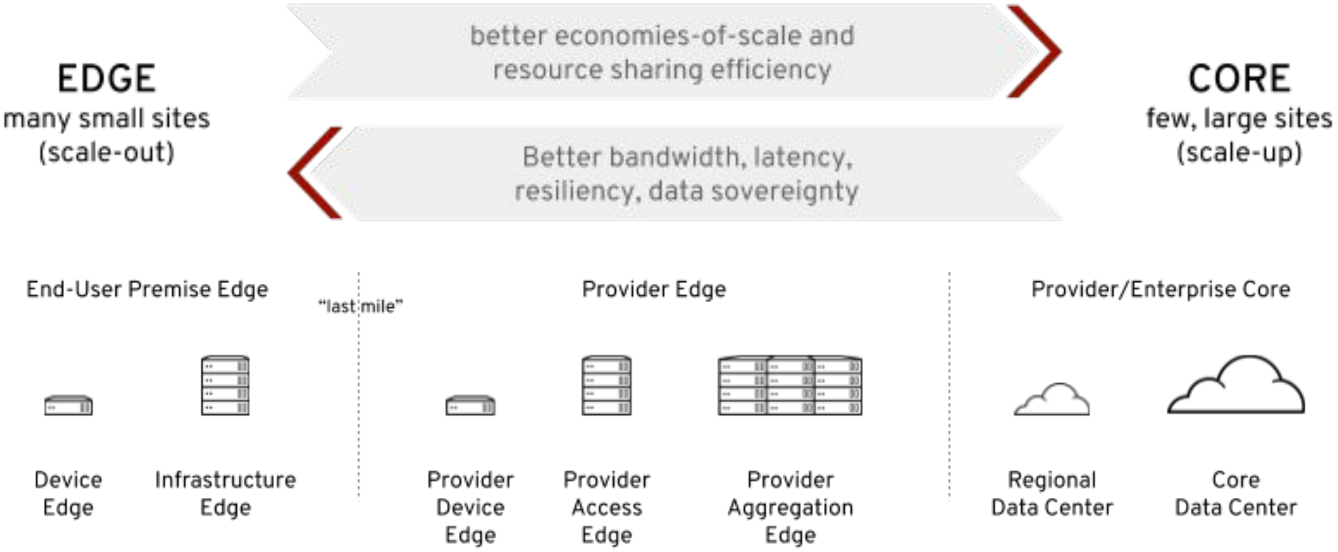


App / Microservice / Function

2. Push to the edge

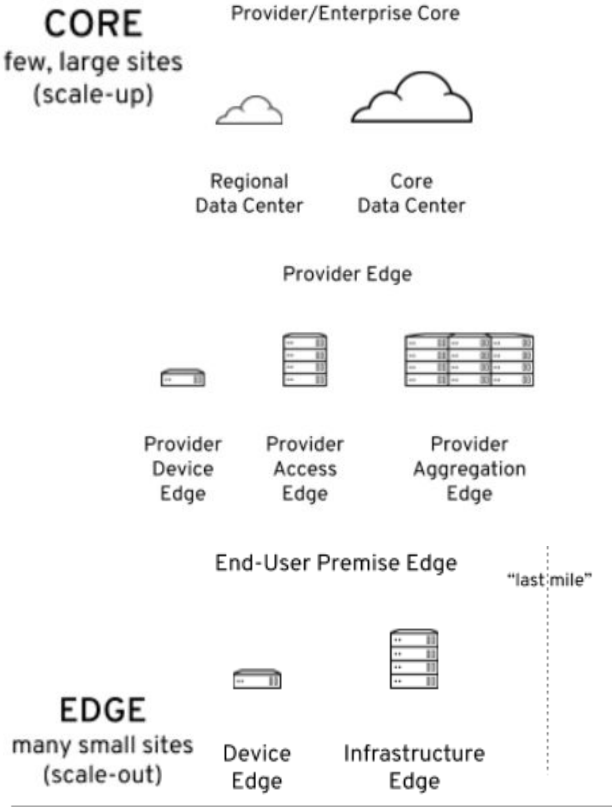
#redhat #rhsummit

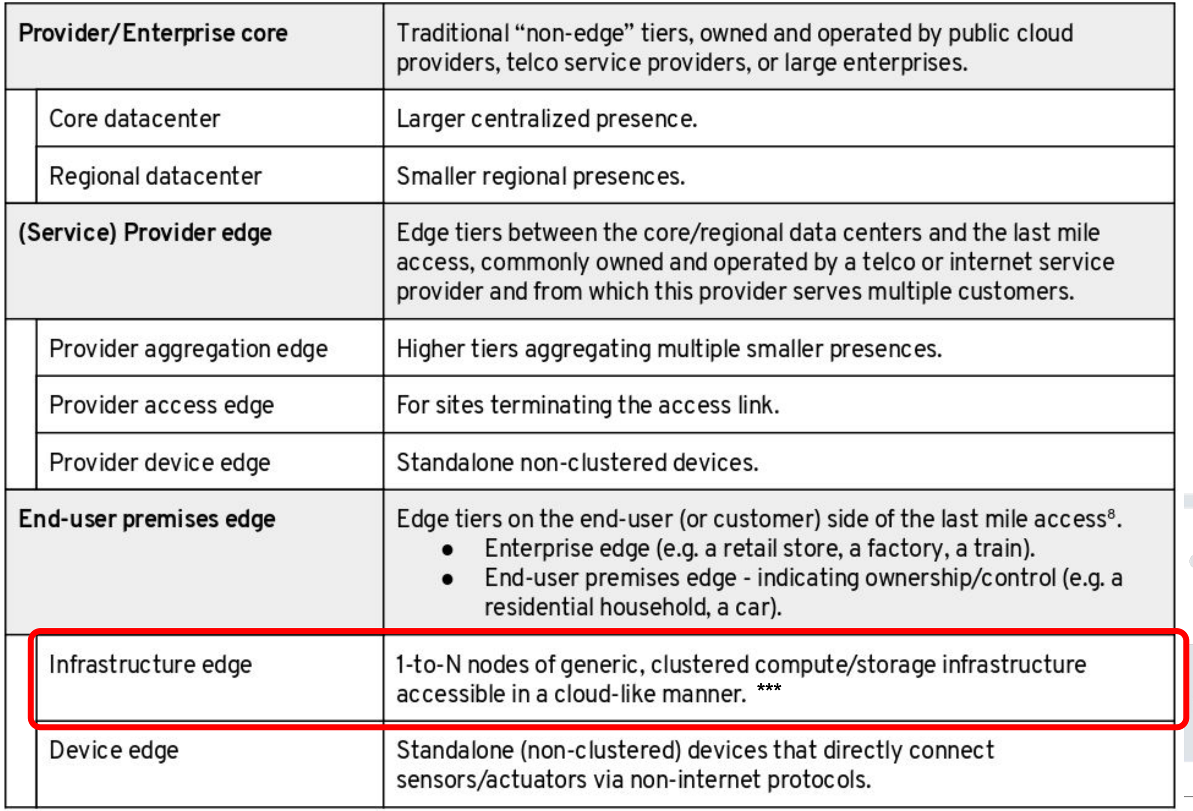
WHAT IS THE EDGE ANYWAYS?



[What is Edge Computing](#)
[Why choose Red Hat for Edge computing?](#)

RED HAT EDGE COMPUTING PoV

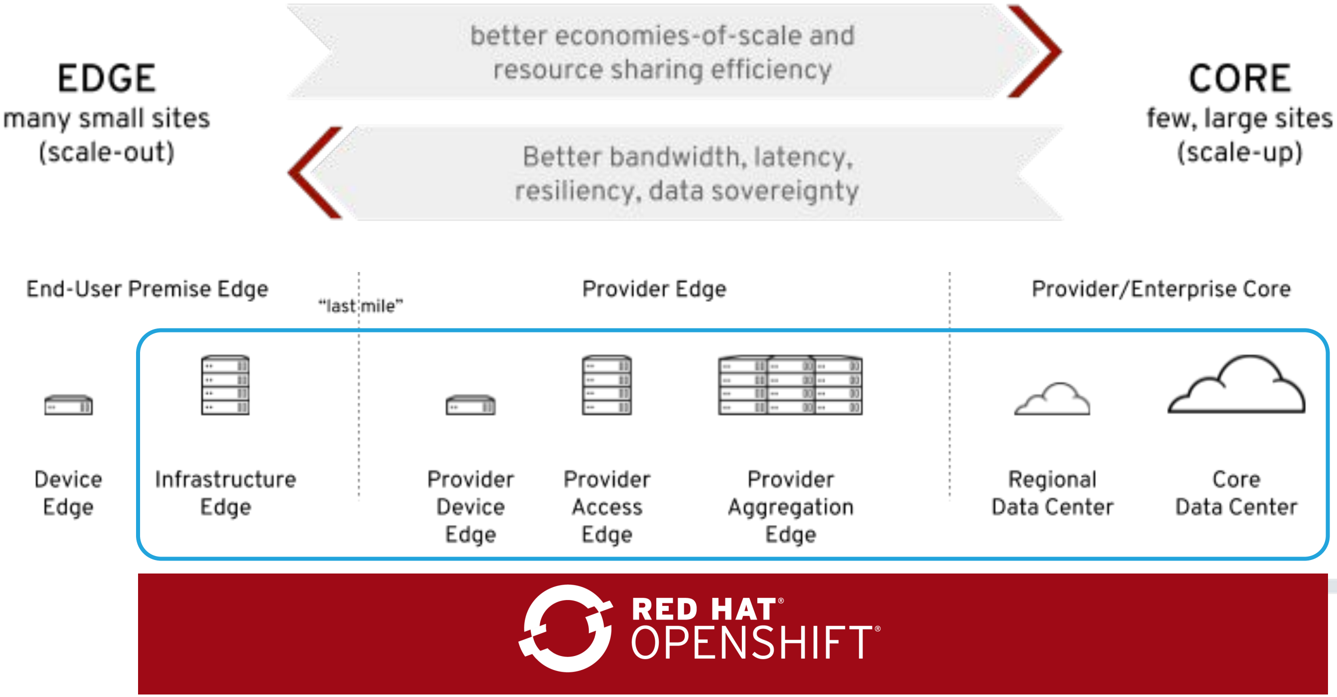




Provider/Enterprise core	Traditional "non-edge" tiers, owned and operated by public cloud providers, telco service providers, or large enterprises.
Core datacenter	Larger centralized presence.
Regional datacenter	Smaller regional presences.
(Service) Provider edge	Edge tiers between the core/regional data centers and the last mile access, commonly owned and operated by a telco or internet service provider and from which this provider serves multiple customers.
Provider aggregation edge	Higher tiers aggregating multiple smaller presences.
Provider access edge	For sites terminating the access link.
Provider device edge	Standalone non-clustered devices.
End-user premises edge	Edge tiers on the end-user (or customer) side of the last mile access[8]. ● Enterprise edge (e.g. a retail store, a factory, a train). ● End-user premises edge - indicating ownership/control (e.g. a residential household, a car).
Infrastructure edge	1-to-N nodes of generic, clustered compute/storage infrastructure accessible in a cloud-like manner. ***
Device edge	Standalone (non-clustered) devices that directly connect sensors/actuators via non-internet protocols.

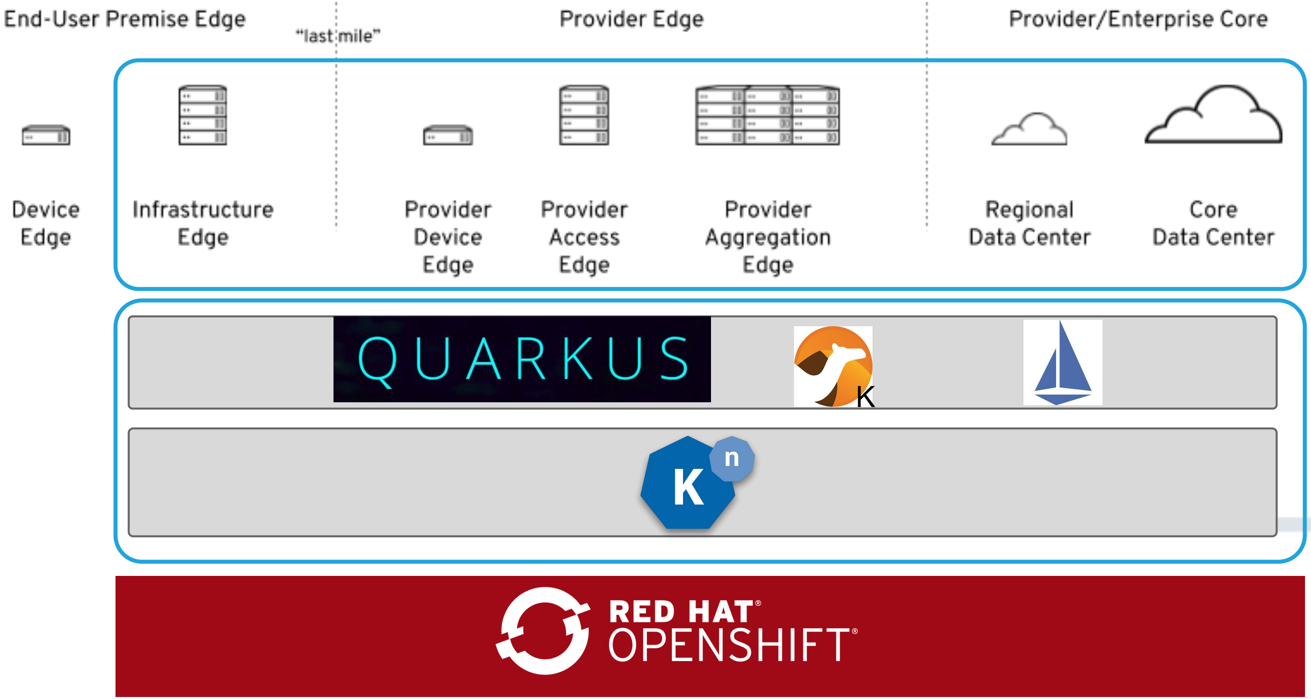
*** Minimum of 3 nodes required today

BEST EDGE COMPUTING FOUNDATION?



CLOUD-NATIVE DEVELOPMENT

DRIVING SERVERLESS / FaaS FROM CORE TO EDGE



KNATIVE

SERVERLESS BUILDING BLOCKS

"...an extension to Kubernetes exposing building blocks to build modern, source-centric, and container-based applications that can run anywhere".

Build

A pluggable model for building artifacts, like jar files, zips or containers from source code.

Serving

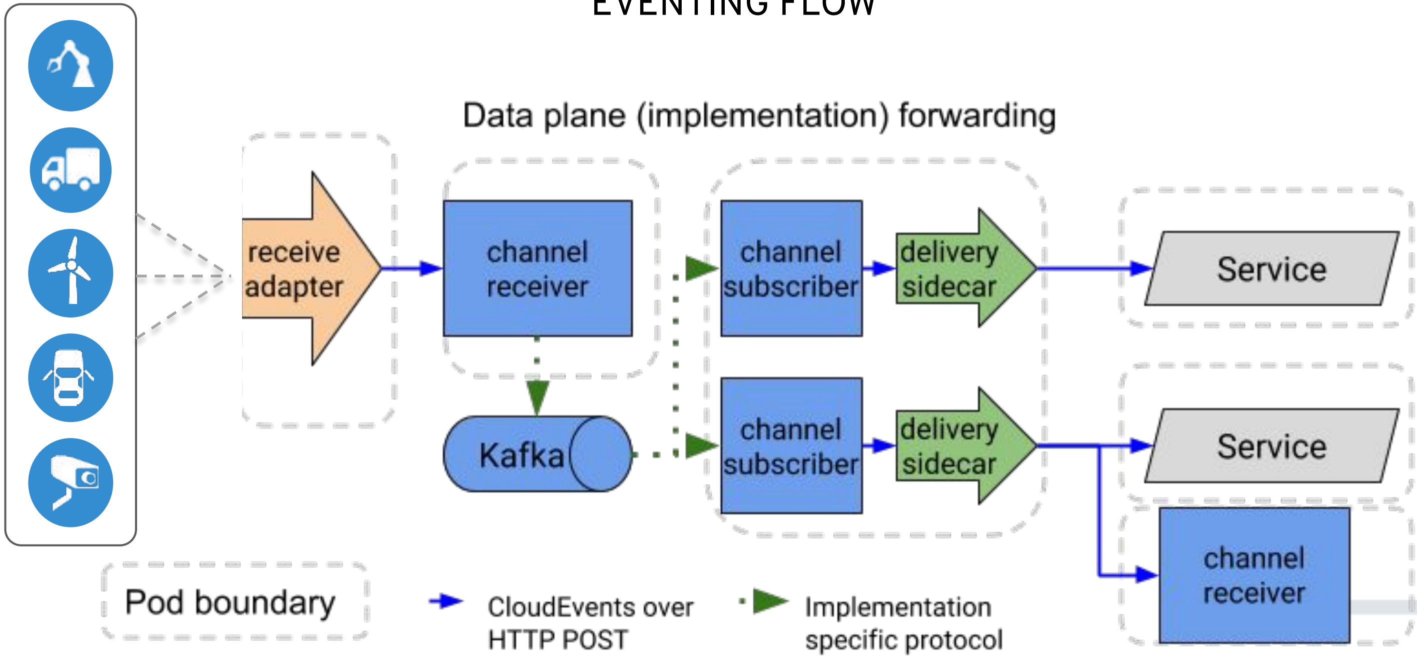
An event-driven model that serves the container with your application and can "scale to zero".

Eventing

Common infrastructure for consuming and producing events that will stimulate applications.

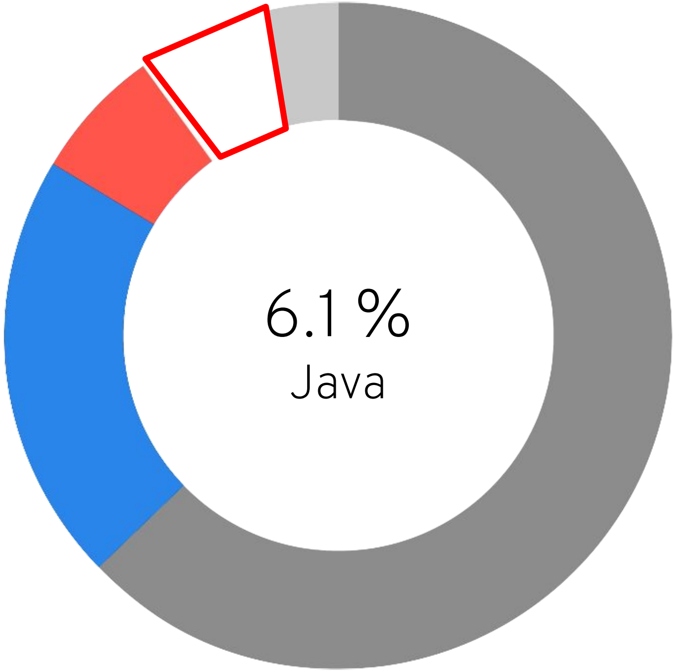
KNATIVE

EVENTING FLOW



SERVERLESS ADOPTION

OK, WE HAVE THE BUILDING BLOCKS, NOW WHAT?



- ● 62.9 % Node.js
- ● 20.8 % Python
- ● 6.4 % Go
- □ 6.1 % Java
- ● 3.8 % C#

https://serverless.com/blog/2018-serverless-community-survey-huge-growth-usage/

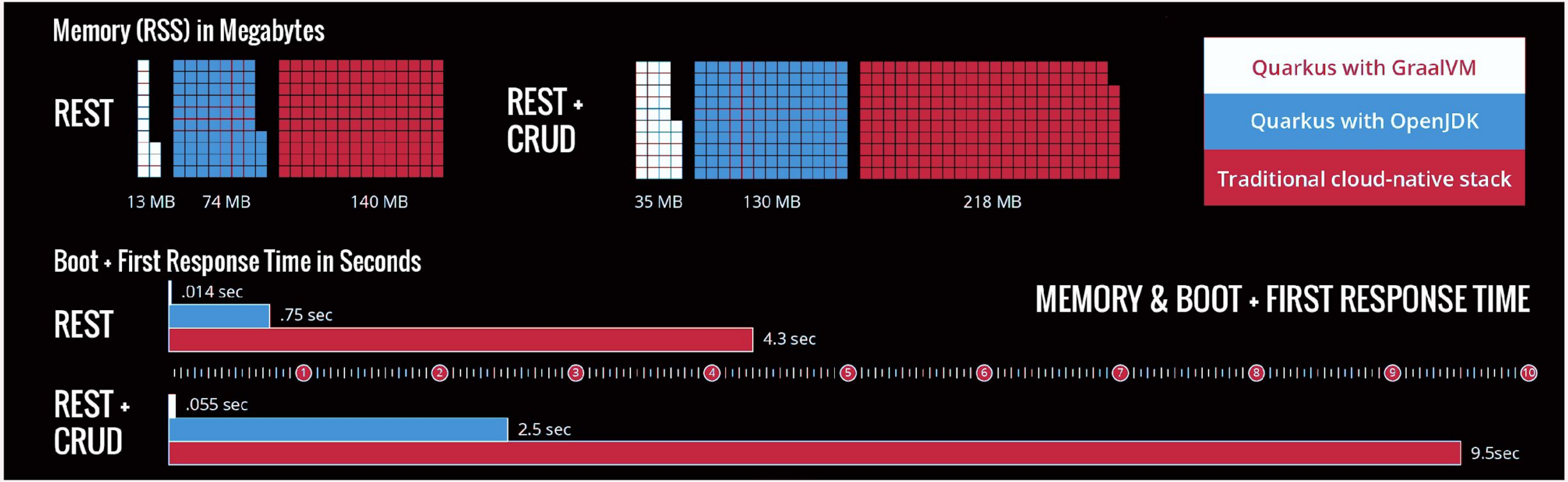
QUARKUS

SUPERSONIC SUBATOMIC JAVA

A Kubernetes Native Java stack tailored for GraalVM & OpenJDK HotSpot, crafted from the best of breed Java libraries and standards

100x faster startup, 10% of the memory in native mode

10x faster and ½ the of the memory on Hotspot/OpenJDK



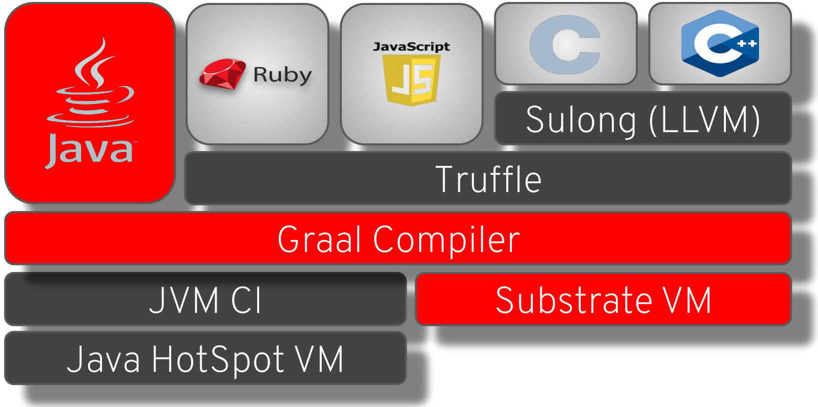
Minimal footprint Java applications -> Native vs Quarkus+OpenJDK JVM vs Traditional JVM

QUARKUS

HOW IS THIS POSSIBLE?

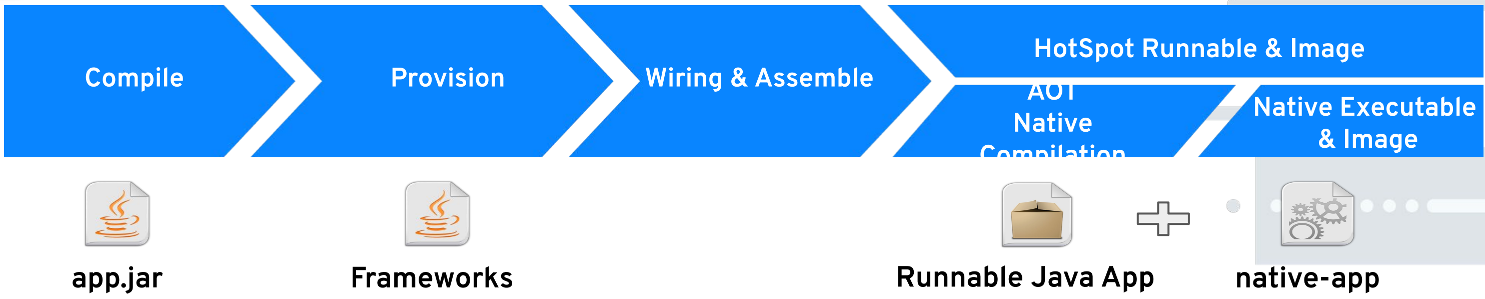
GraalVM™

Polyglot, Native or JVM, Embeddable



Quarkus enables Java developers to easily use the most popular frameworks and standards directly on SubstrateVM without any hassle.

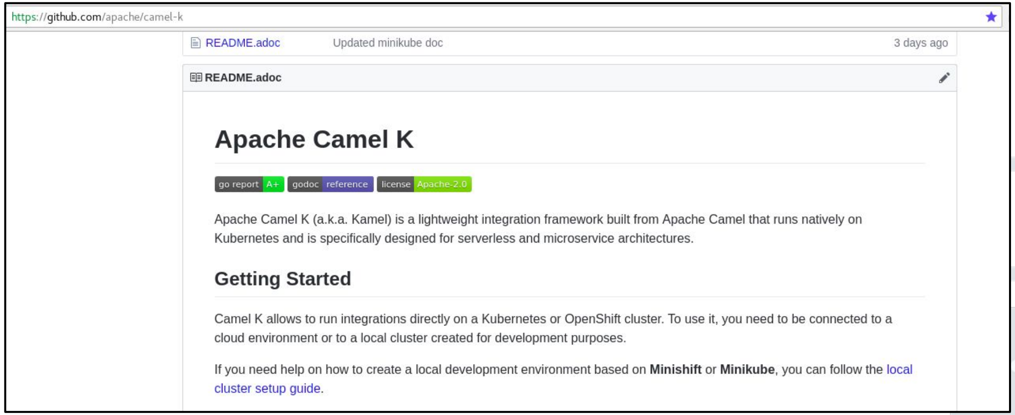
BUILD PROCESS



APACHE CAMEL K

SERVERLESS INTEGRATION

- **A lightweight platform for directly running Camel integration DSL in a cloud-native way**
- Based on (ex Core OS) **operator-sdk**
- Works on **Openshift** and **Kubernetes**
- Create **Event Sources**, use **EIPs**, define **Integration Functions**



Apache Camel K

Apache Camel K (a.k.a. Kamel) is a lightweight integration framework built from Apache Camel that runs natively on Kubernetes and is specifically designed for serverless and microservice architectures.

Getting Started

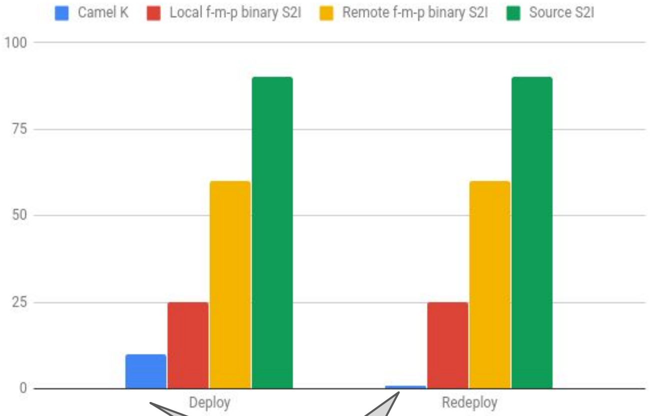
Camel K allows to run integrations directly on a Kubernetes or OpenShift cluster. To use it, you need to be connected to a cloud environment or to a local cluster created for development purposes.

If you need help on how to create a local development environment based on **Minishift** or **Minikube**, you can follow the local cluster setup guide.

CAMEL K

OVERVIEW

Time to run an integration using different strategies (in seconds)



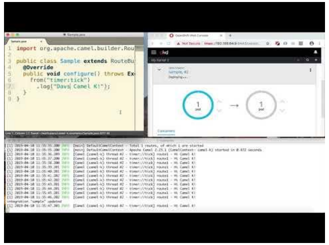
Lower is better :)

File routes.groovy

```
from(“telegram:bots/bot-id”)
  .transform()...
  .to(“kafka:topic”);

from(“kafka:topic”)
  .to(“http:my-host/api/path”);
```

```
$ kamel run routes.groovy
```



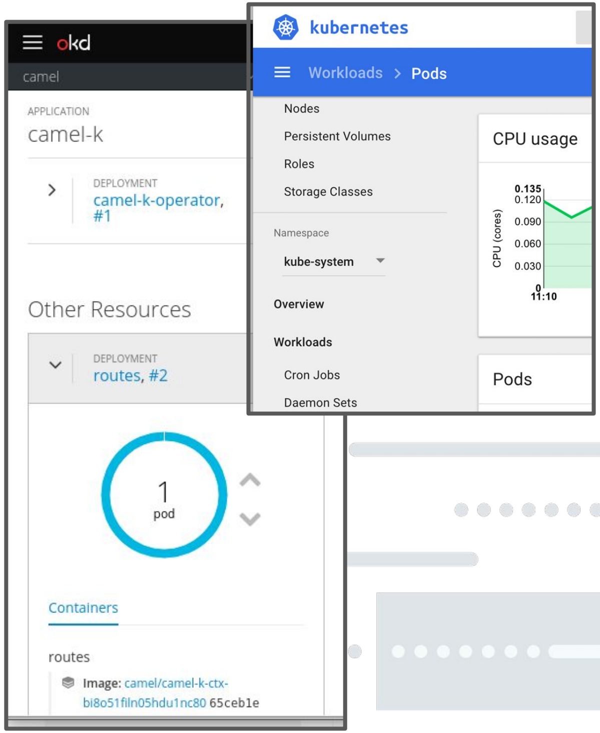
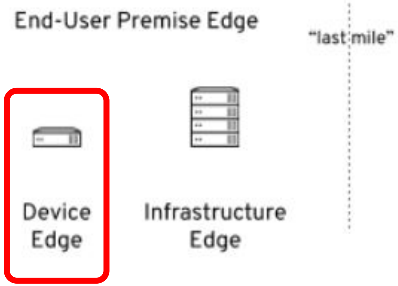


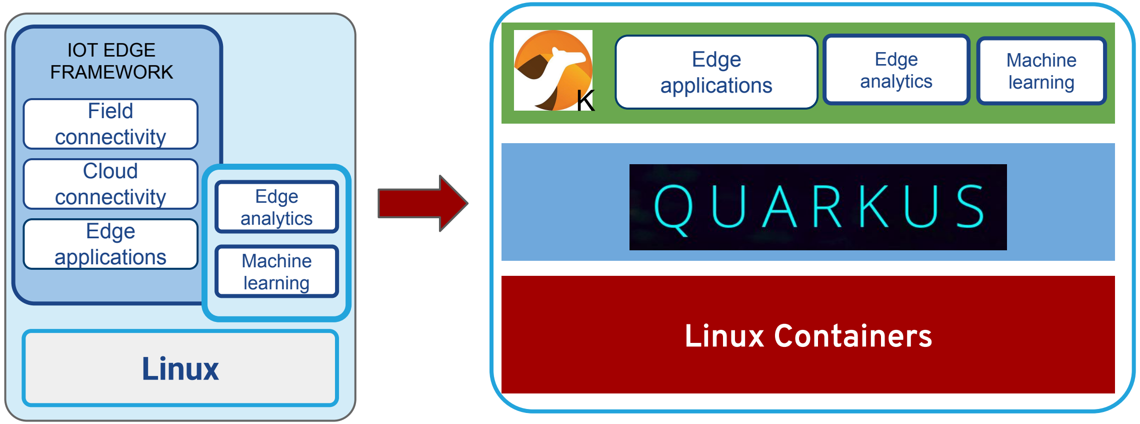
Image: camel/camel-k-ctx-bi8o51filn05hdu1nc80 65ceb1e

DEVICE EDGE

WHAT CAN WE DO WITHOUT KNATIVE/K8s?

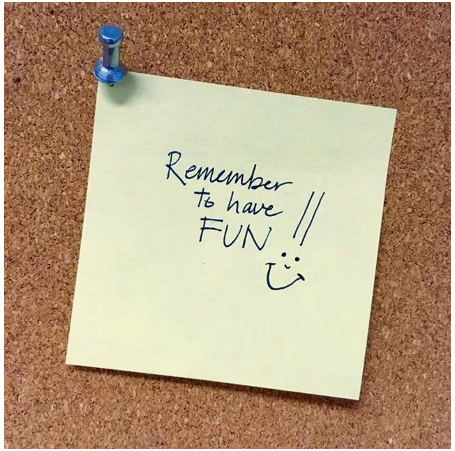


- Run native functions in containers at the edge
- Benefit from low footprint and fast startup



CALL TO ACTION

- https://try.openshift.com/
- https://developers.redhat.com/
- https://knative.dev/docs/
- https://quarkus.io/
- https://github.com/apache/camel-k



LET'S BUILD CLOUD NATIVE APPS

RED HAT
SUMMIT

THANK YOU

in linkedin.com/company/Red-Hat

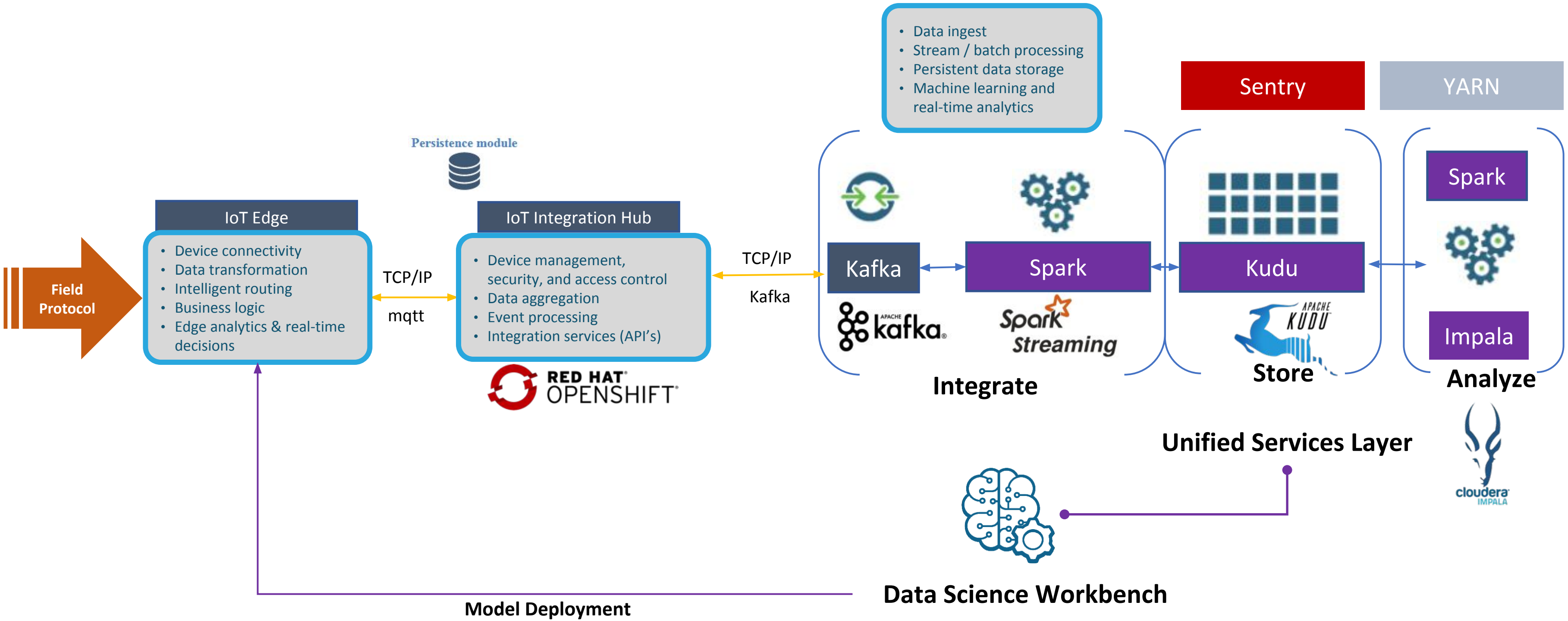
f facebook.com/RedHatinc

You Tube youtube.com/user/RedHatVideos

twitter.com/RedHat

CASE STUDY - Smart Manufacturing - Deep Dive

Integrating open source IoT operating technology, data management, analytics, and applications



Field Protocol

IoT Edge

- Device connectivity
- Data transformation
- Intelligent routing
- Business logic
- Edge analytics & real-time decisions

TCP/IP

mqtt

Persistence module

IoT Integration Hub

- Device management, security, and access control
- Data aggregation
- Event processing
- Integration services (API's)

RED HAT OPENSHIFT

TCP/IP

Kafka

- Data ingest
- Stream / batch processing
- Persistent data storage
- Machine learning and real-time analytics

Kafka

Spark

Spark Streaming

Integrate

Sentry

YARN

Kudu

Store

Spark

Impala

Analyze

Unified Services Layer

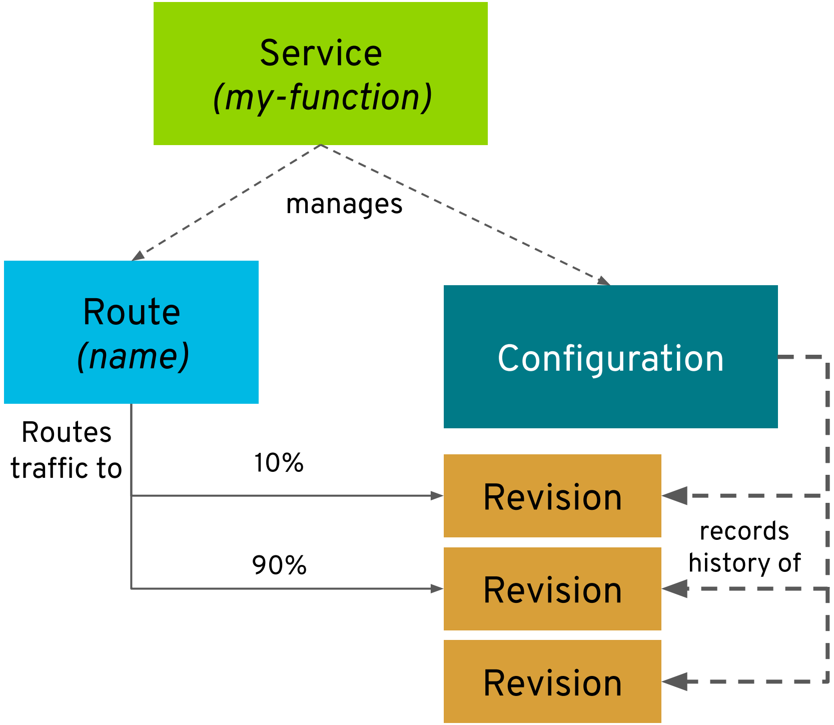
cloudera IMPALA

Data Science Workbench

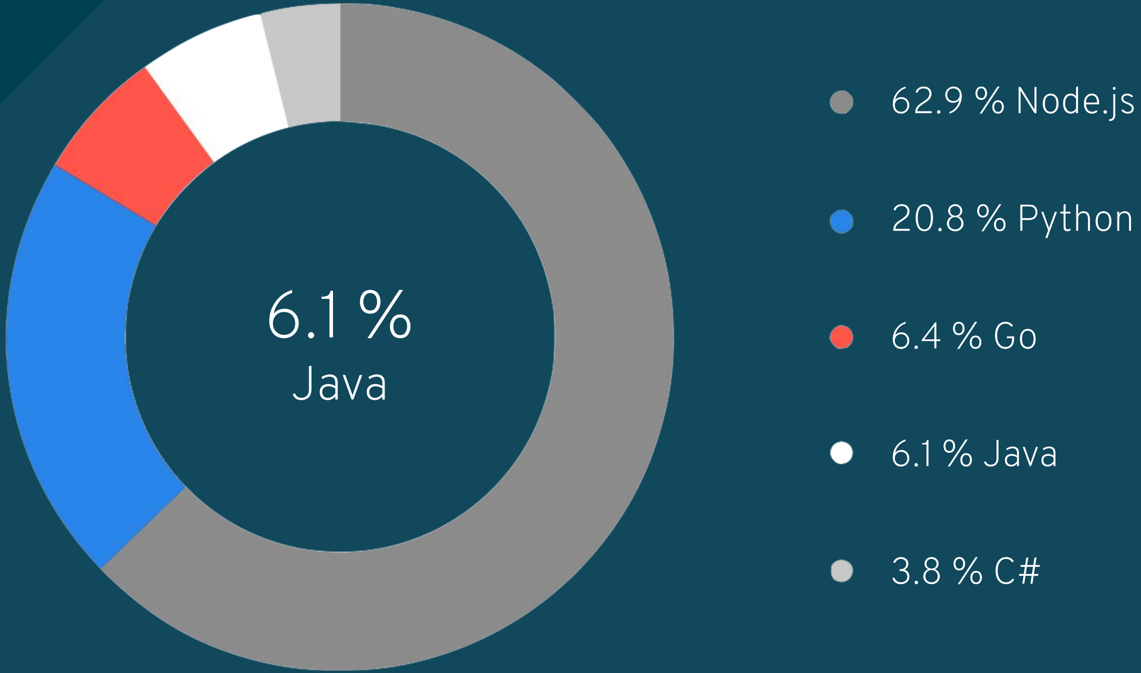
Model Deployment

Knative Overview - Serving

- **Configurations** represent the ‘floating HEAD’ of a history of **Revisions**
- **Revisions** represent immutable snapshot of code and configuration
- **Routes** configure ingress over a collection of Revisions and/or Configurations
- **Services** (nope, not K8s services) are top-level controllers that manage a set of Routes and Configurations to implement a network service

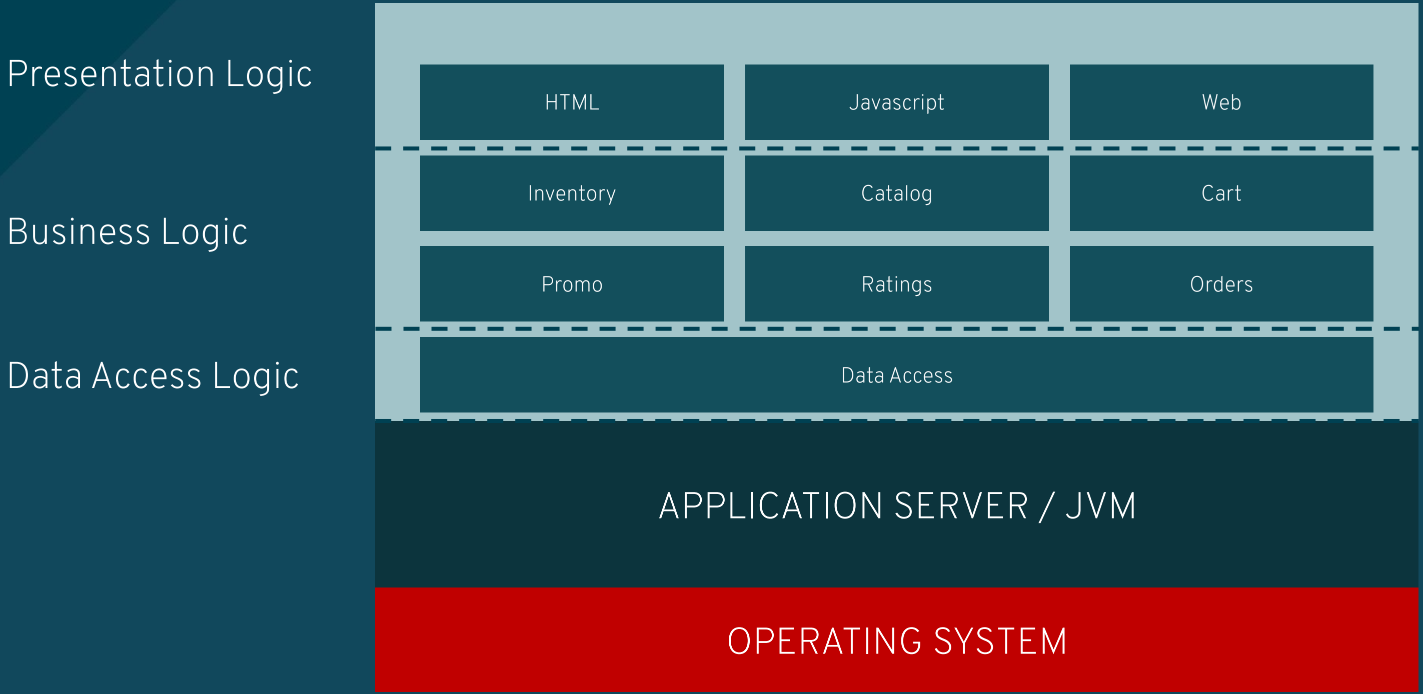


SERVERLESS ADOPTION

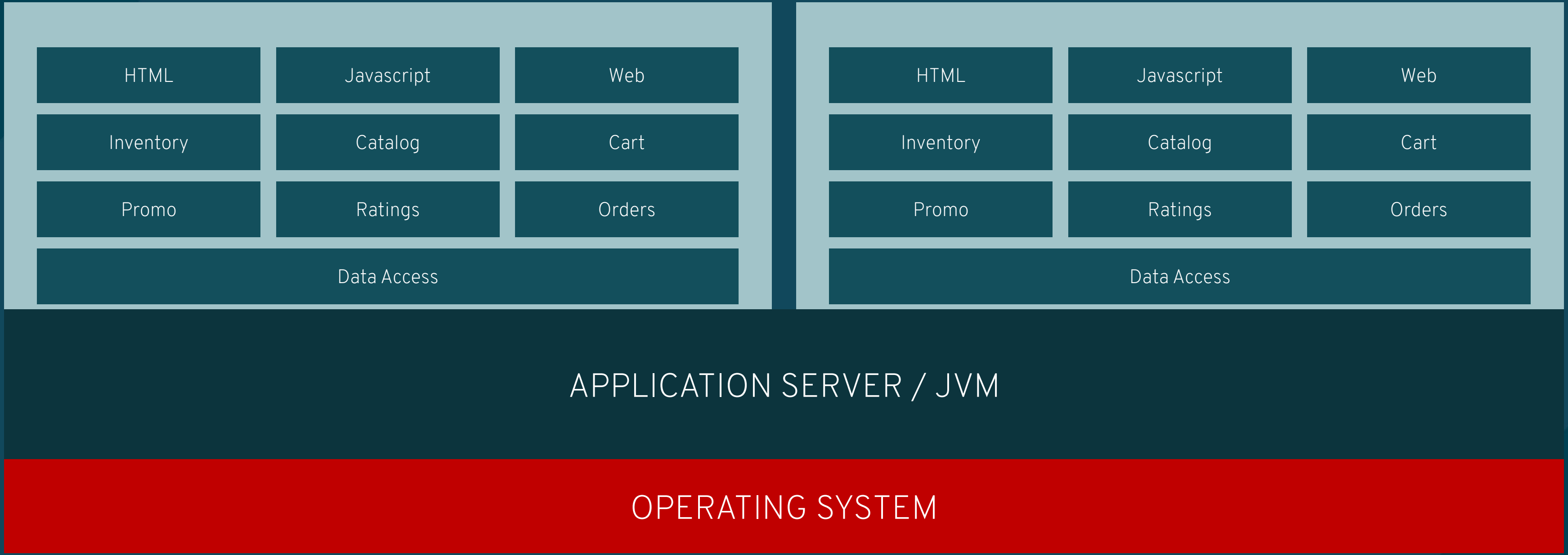


https://serverless.com/blog/2018-serverless-community-survey-huge-growth-usage/

ENTERPRISE JAVA WAS DESIGNED FOR 3-TIER ARCHITECTURE



HOW APPLICATION SERVERS DEPLOYED JAVA APPS



HTML Javascript Web

Inventory Catalog Cart

Promo Ratings Orders

Data Access

HTML Javascript Web

Inventory Catalog Cart

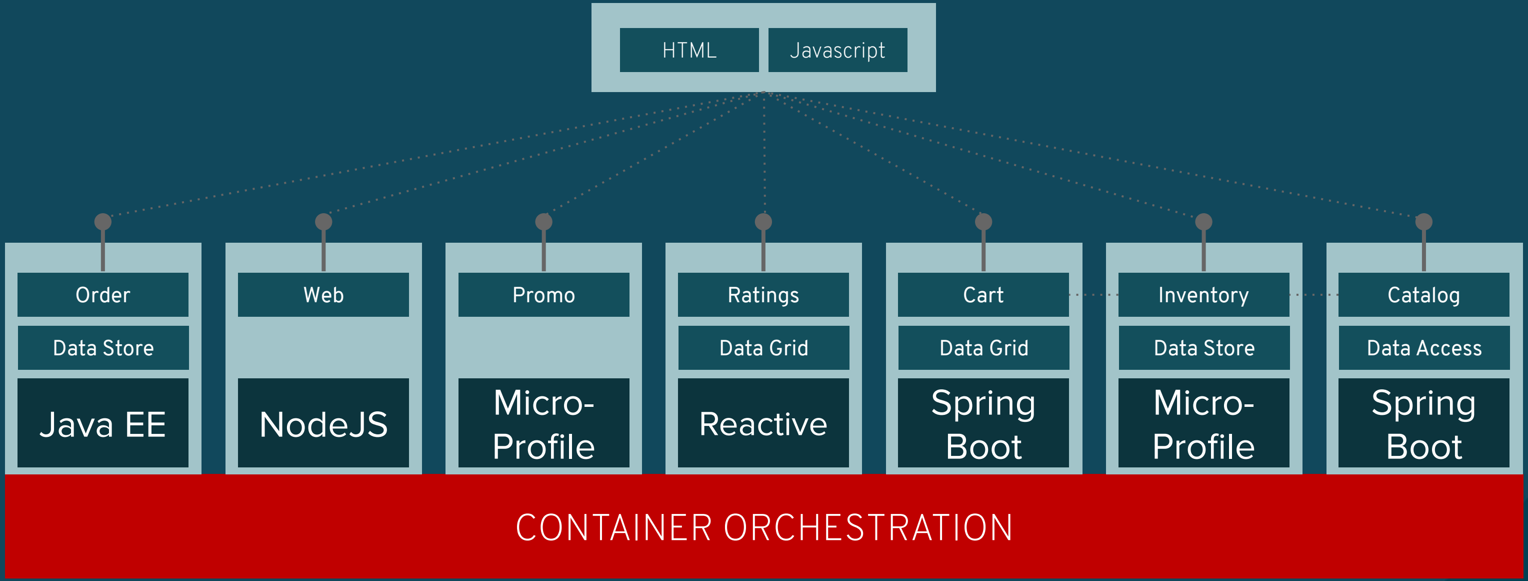
Promo Ratings Orders

Data Access

APPLICATION SERVER / JVM

OPERATING SYSTEM

MICROSERVICES CHANGED HOW WE DEPLOY APPS

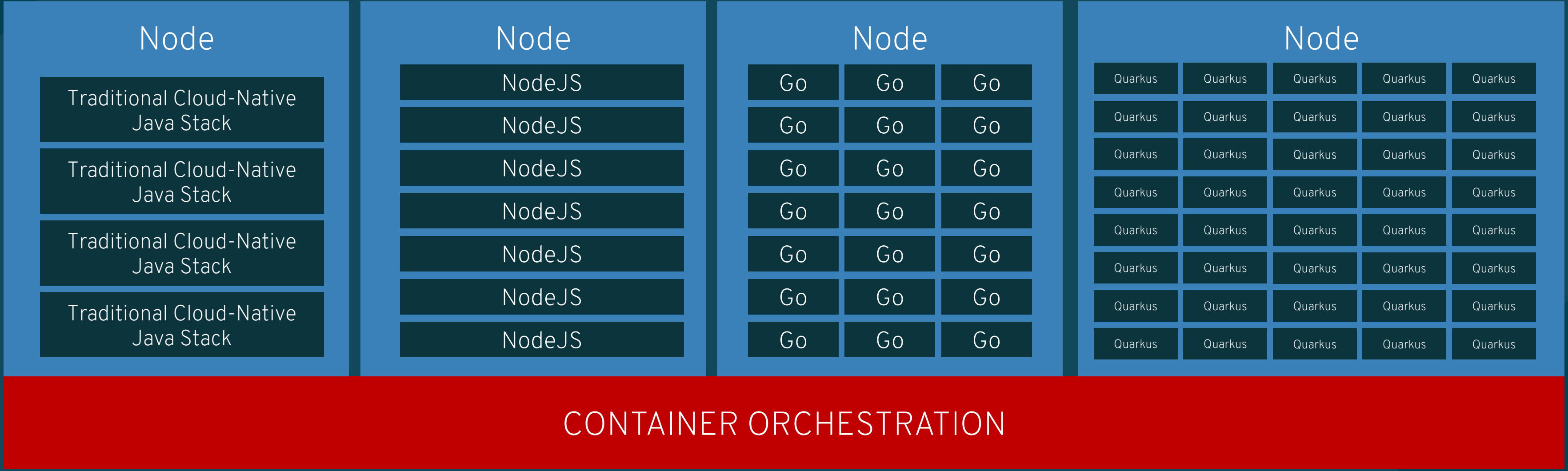


HTML | Javascript

Order	Web	Promo	Ratings	Cart	Inventory	Catalog
Data Store			Data Grid	Data Grid	Data Store	Data Access
Java EE	NodeJS	Micro-Profile	Reactive	Spring Boot	Micro-Profile	Spring Boot

CONTAINER ORCHESTRATION

THE HIDDEN TRUTH ABOUT JAVA IN CONTAINERS



Node

Traditional Cloud-Native Java Stack

Traditional Cloud-Native Java Stack

Traditional Cloud-Native Java Stack

Traditional Cloud-Native Java Stack

Node

NodeJS

NodeJS

NodeJS

NodeJS

NodeJS

NodeJS

NodeJS

Node

Go	Go	Go
Go	Go	Go
Go	Go	Go
Go	Go	Go
Go	Go	Go
Go	Go	Go
Go	Go	Go

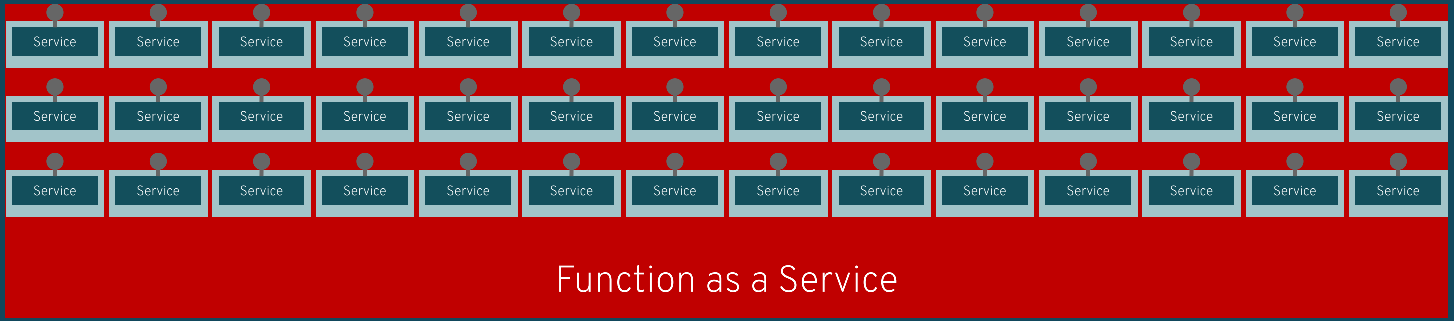
Node

Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus

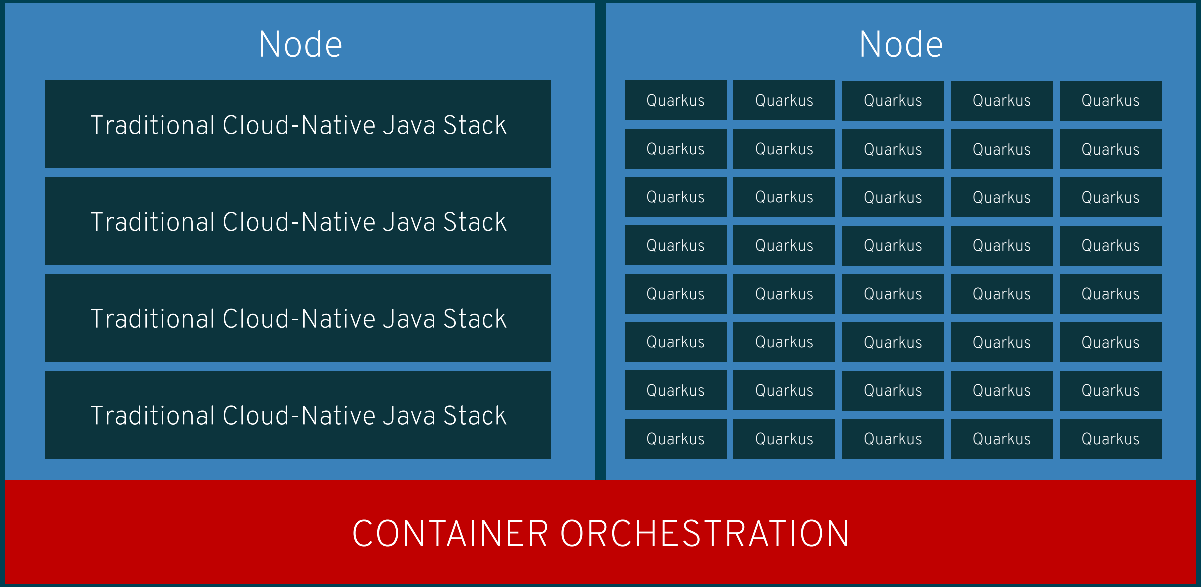
CONTAINER ORCHESTRATION

https://developers.redhat.com/blog/2017/03/14/java-inside-docker/

FAAS & SERVERLESS CHANGES HOW WE DEPLOY APPS IN THE FUTURE



QUARKUS NATIVE DEPLOYMENT



Node

Traditional Cloud-Native Java Stack

Traditional Cloud-Native Java Stack

Traditional Cloud-Native Java Stack

Traditional Cloud-Native Java Stack

Node

Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus
Quarkus	Quarkus	Quarkus	Quarkus	Quarkus

CONTAINER ORCHESTRATION

HOW IS THIS POSSIBLE?

GraalVM™

Polyglot, Native or JVM, Embeddable

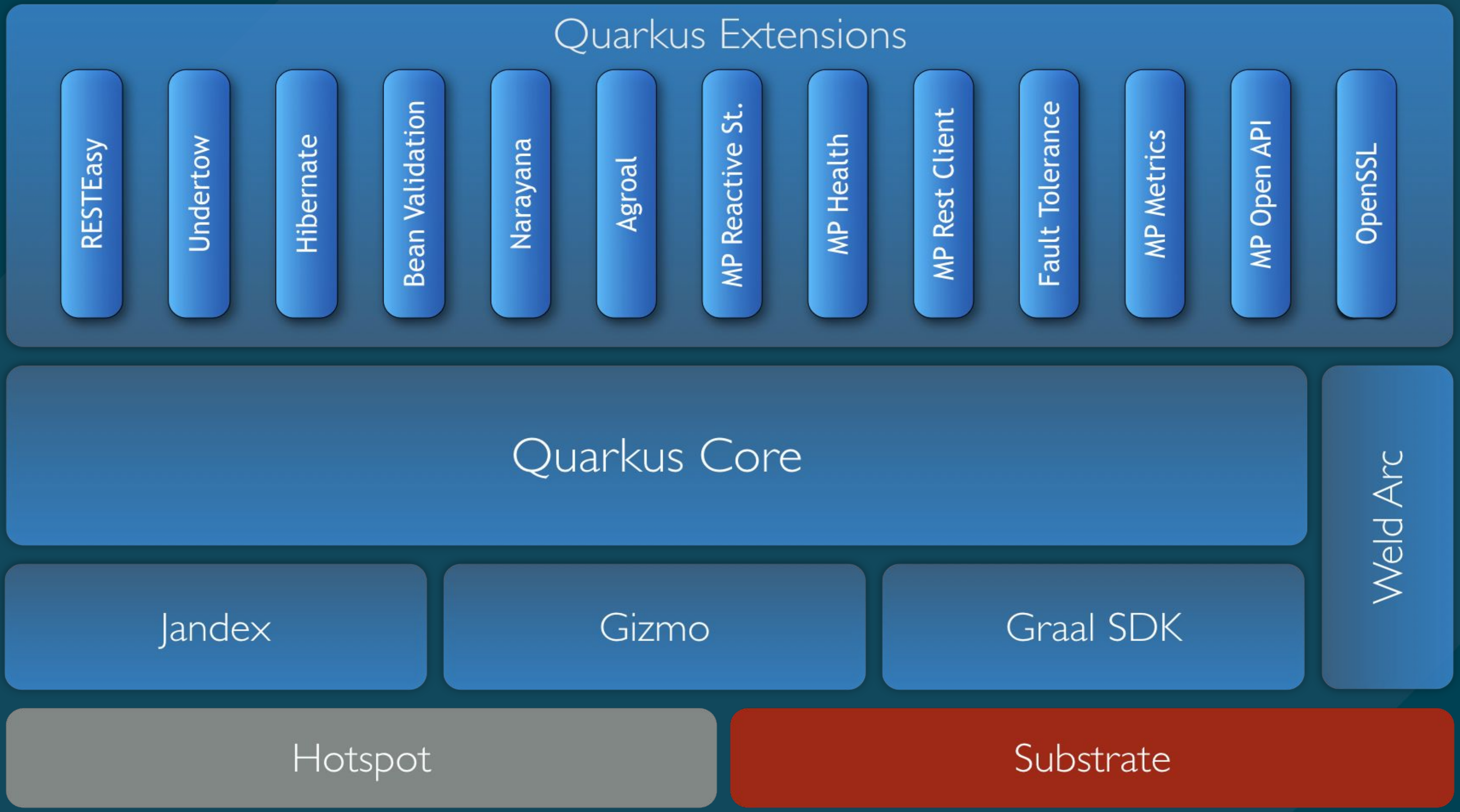
DEVELOPING NATIVE APPLICATIONS FOR SUBSTRATEVM IS PAINFUL

Quarkus enables Java developers to easily use the most popular frameworks and standards directly on SubstrateVM without any hassle.

FRAMEWORK OPTIMIZATIONS

- Moved as much as possible to build phase
- Minimized runtime dependencies
- Maximize dead code elimination
- Introduced clear metadata contracts
- Spectrum of optimization levels
  (all -> some -> no runtime reflection)

Optimizations benefit both GraalVM (SVM) and HotSpot



Quarkus Extensions

RESTEasy

Undertow

Hibernate

Bean Validation

Narayana

Agroal

MP Reactive St.

MP Health

MP Rest Client

Fault Tolerance

MP Metrics

MP Open API

OpenSSL

Quarkus Core

Weld Arc

Jandex

Gizmo

Graal SDK

Hotspot

Substrate

IMAGE CREDITS

bmw - https://www.flickr.com/photos/pinemikey/8850948516
android - https://www.flickr.com/photos/microsiervos/15350193299
nest - https://www.flickr.com/photos/161156731@N05/45096474951
question - https://www.flickr.com/photos/dharmabum1964/3108162671

SECTION BREAK

SECTION BREAK

SECTION BREAK

CONTENT SLIDE

RED HAT SUMMIT

THANK YOU

in linkedin.com/company/Red-Hat

f facebook.com/RedHatinc

You Tube youtube.com/user/RedHatVideos

twitter.com/RedHat