

DevOps with Containers

Noel O'Connor

Senior Principal Architect, Red Hat

noconnor@redhat.com

Mike Hepburn

Global Principal Architect, Red Hat

mhepburn@redhat.com

Container Native Applications

- ▶ Cloud Native (for today's world) leans towards building ***Systems***
 - For example, a cloud native container platform (i.e. OpenShift)
 - No longer preferable to deploy new applications onto VMs
- ▶ Container Native leans towards building ***Applications to run on Container platforms***
 - Applications that take advantage of the features of the underlying container platform
 - Abstracts developers as much as possible from infrastructure concerns
 - Resilient - not just during failure
 - Operationally consistent

Cloud Native...

Cloud Native Definition: <https://github.com/cncf/toc/blob/master/DEFINITION.md>

Cloud native technologies empower organizations to build and run **scalable** applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, **immutable infrastructure**, and declarative APIs exemplify this approach.

These techniques enable loosely coupled systems that are **resilient**, **manageable**, and **observable**. Combined with robust **automation**, they allow engineers to make high-impact changes **frequently** and predictably with minimal toil.



Container Native

- Focuses on business logic
- Uses platform provided services e.g. service discovery, configuration, HPA, PDB, Observability
- Operationally consistent

Cloud Native

- Independently Deployable Services
- Fault tolerant
- Suitable configuration & discovery

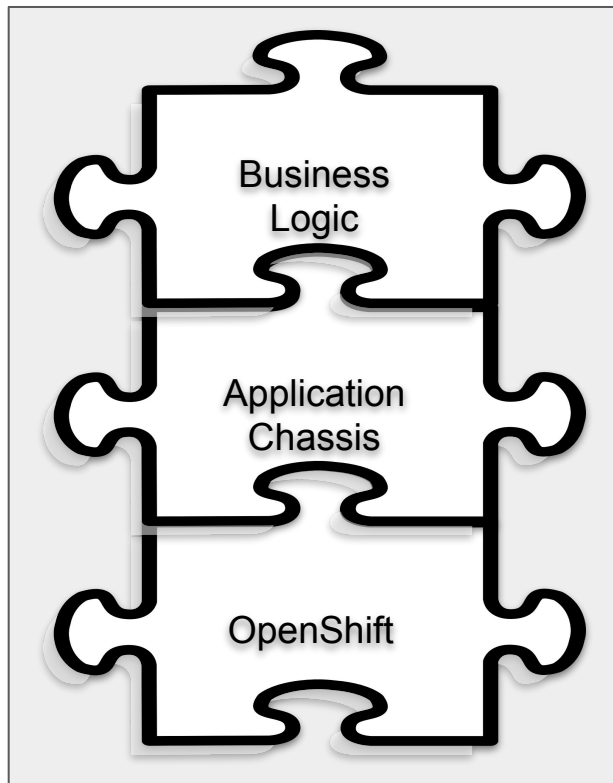
Cloud Ready

- Well Defined Services/Monolith
- Automated deployment
- Cloud suitable configuration

Cloud Thanos ;-)

- Manual deployment
- Network/Storage dependencies
- Manual service discovery
- Brittle, big-ball-of-mud

Container Native Application Layers



Container Native Ingredients (Platform)

Container native runtime

- ▶ Certified Base Images
- ▶ KNative
- ▶ OpenShift Virtualisation

Observability

- ▶ Centralized logging - EFK
- ▶ Metrics - Prometheus
- ▶ Tracing - Jaeger

Service Mesh

- ▶ Timeouts management
- ▶ Circuit breakers
- ▶ Fault injection
- ▶ Advanced rollouts

Build/CI

- ▶ Approved pipelines
 - Build
 - Promotion/Deploy
 - GitOps

Container Native Ingredients (Platform)

Security

- ▶ Automatic distribution, rotation, revocation of credentials
- ▶ Zero-trust network principles: microsegmentation and mTLS
- ▶ Native support of Oauth
- ▶ Credential management
- ▶ Policies

Stateful components

- ▶ Platform Managed services e.g. DBs, message brokers

Workload adaptability

- ▶ Auto-scaling
- ▶ Self-Healing
- ▶ Dynamic Storage

Container Native Ingredients (Application)

Application Endpoints

- ▶ Prometheus metrics scraping
- ▶ Probes - Liveness/Readiness
- ▶ Software Version
- ▶ Thread Dump generator
- ▶ Synthetic Transaction generator
- ▶ Dynamic logging level switch
- ▶ OpenAPI/Swagger

Metrics

- ▶ Connection Pools
- ▶ Last request timestamp
- ▶ Request/Error/Thread counts
- ▶ Garbage Collection (if applicable)

Container Native Ingredients (Application)

Configuration

- ▶ Runtime flags e.g. JVM -D
- ▶ ConfigMaps/Secrets/Env vars
- ▶ Service Serving Secrets
- ▶ Feature Flags
- ▶ Known ports - HTTP, debug etc
- ▶ Helm templates

MetaData

- ▶ Labels
- ▶ Annotations

Logging

- ▶ Consistent format
- ▶ Correlation Ids
- ▶ Default to stdout

Resiliency

- ▶ Shutdown signal handlers
- ▶ Circuit breakers status
- ▶ Timeouts and exponential backoff

Container Native Principles

Focus

- ▶ *"Applications not platform"*

Consistency

- ▶ In Development approach
- ▶ In Operational approach

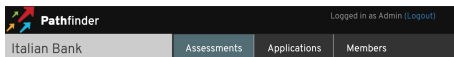
Avoid NIH

- ▶ Utilise platform features
- ▶ Use 3rd party services, managed via Operators

Automate

- ▶ Strive to remove manual interaction as much as is feasible

Pathfinder Assessment Tool



CONSUMER APP PORTAL ASSESSMENT

Application Details

1. Does the application development team understand and actively develop the application? 0

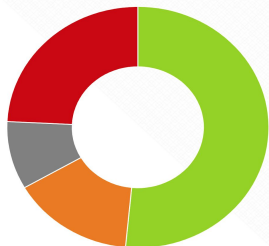
- ☐ Unknown
- ☐ External 3rd party or COTS application
- ☐ In maintenance mode, no app SME knowledge, poor documentation
- ☐ Maintenance mode, SME knowledge available
- ☐ Actively developed, SME knowledge available
- ☒ New Greenfield application

2. How is the application supported in Production? 0

- ☐ Unknown
- ☐ Application production support outsourced to 3rd party support provider. Ticket driven escalation process, no inhouse support resources.
- ☐ Production support provided by separate internal team, little interaction with development team.
- ☐ Multiple teams support the application using an established escalation model
- ☐ SRE based approach with knowledgeable and experienced operations team
- ☒ Pure DevOps model, the team that builds it is responsible for running it in Production



ARCHITECT REVIEW



CONSUMER APP PORTAL

Application Description: No description provided
Assessment Notes: Currently under development, still a work in progress.

Please use this section to provide your assessment of the possible migration/modernisation plan and an effort estimation.

Proposed Action: Re-host
Effort Estimate: Large
Business Criticality: 8
Work Priority: 8

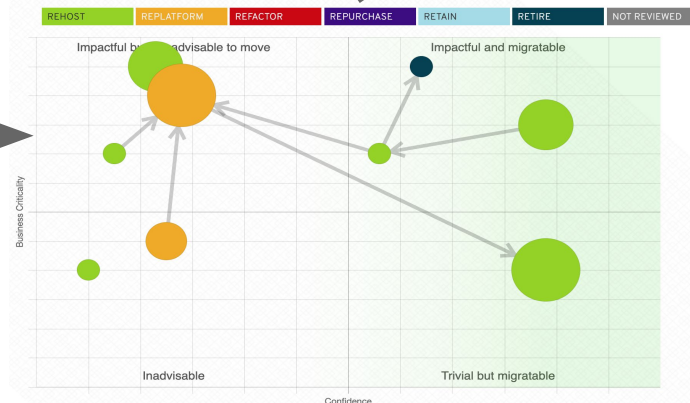
Submit Review

Question	Answer	Rating
Are 3rd party/vendor components supported in containers?	Not recommended to run component in containers	RED
Dependencies - (Incoming/Northbound)	No dependent systems	GREEN
Dependencies - (Outgoing/Southbound)	Limited processing available if dependencies are unavailable	GREEN

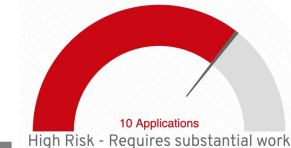
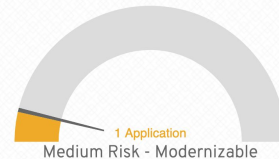
Username:

Password:

Submit



CURRENT LANDSCAPE



Pathfinder – Background

When **Migrating** to Containers:

What **Aspects** of my **Application** do I need to consider?

What **Changes** do I need to consider to my **Operational** processes?

What does **Good** look like on a **Container** platform?

Basically - Does my app look good in this?

Pathfinder – Purpose

*A **Red Hat** opinion on what Applications*

Should,

Could

and

Should Not

Run in containers

Assessment Areas

The assessment questions cover the following application aspects



- Architectural Suitability
- Dependencies
- Application resiliency
- Communication
- Compliance
- State Management
- Runtime profile
- Observability



- Level of ownership
- Service Discovery
- Deployment Complexity
- Application Testing
- Application Security
- Application Configuration
- Clustering
- Custom questions can also be easily added if required

Pathfinder - Demo



Frequently Asked Questions

- ▶ *Can I add my own questions?*

Pathfinder ships with a default set of questions out of the box. It is possible to create additional questions and add them to the assessment. See the following link

<https://github.com/redhat-cop/pathfinder/wiki/Adding-custom-questions>

- ▶ *Can I export my data?*

Yes, Pathfinder supports exporting the collected data via the User Interface.

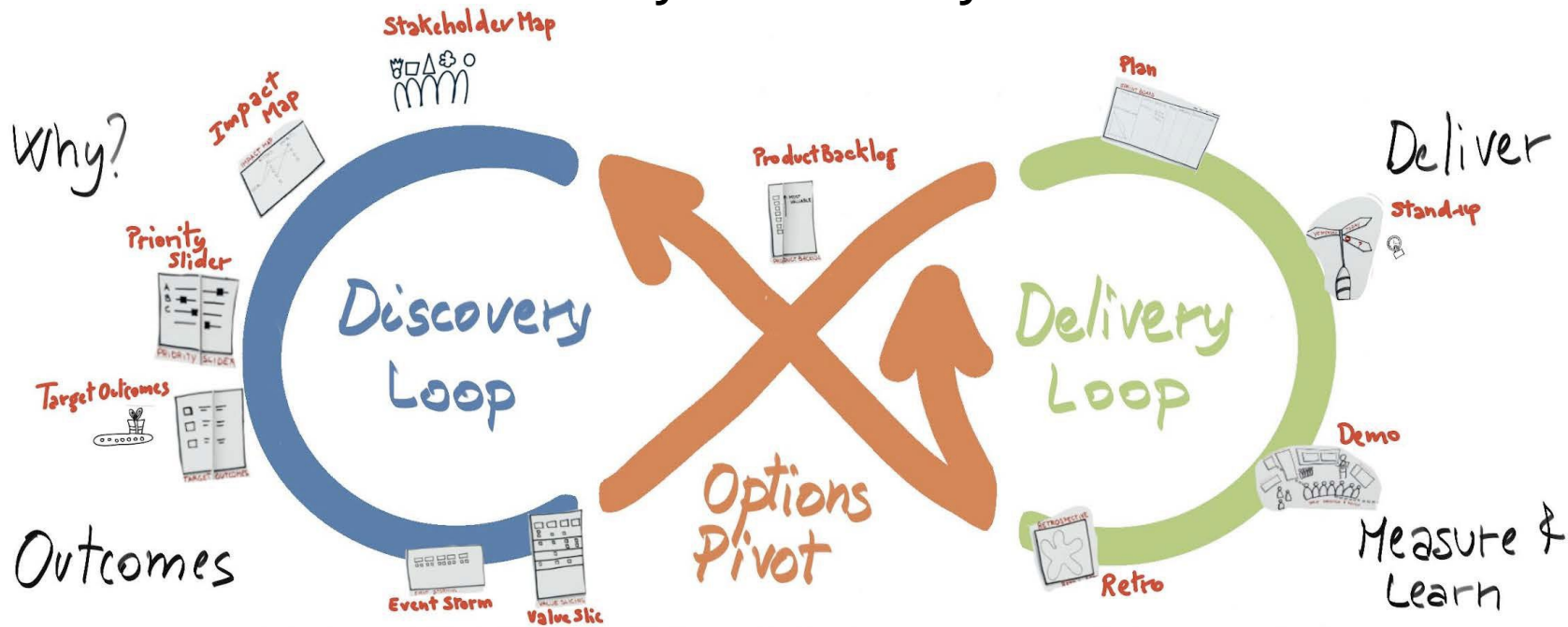
- ▶ *How much does the application it cost ?*

The Pathfinder application is fully open sourced see <https://github.com/redhat-cop/pathfinder>. The only thing we ask is to follow the *Creative Commons Attribution License* for the questions and answers.

- ▶ *Can I install it on Kubernetes ?*

Currently the Pathfinder installer relies on OpenShift templates to install MongoDB. However the application itself can run on any K8S instance if you deploy it manually.

Continuous Discovery & Delivery



Culture & Collaboration



Technical

Hat

Five failures



Leadership

Prevents change



Product

Builds things that
don't matter



Development

Builds wrong
things



Architecture

Builds things
wrong



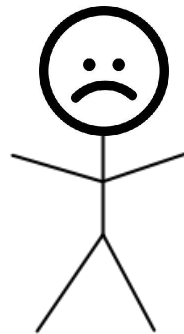
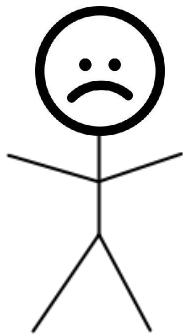
Operations

Incidents and
outages

The confusion is not understanding the other game



Development



Operations

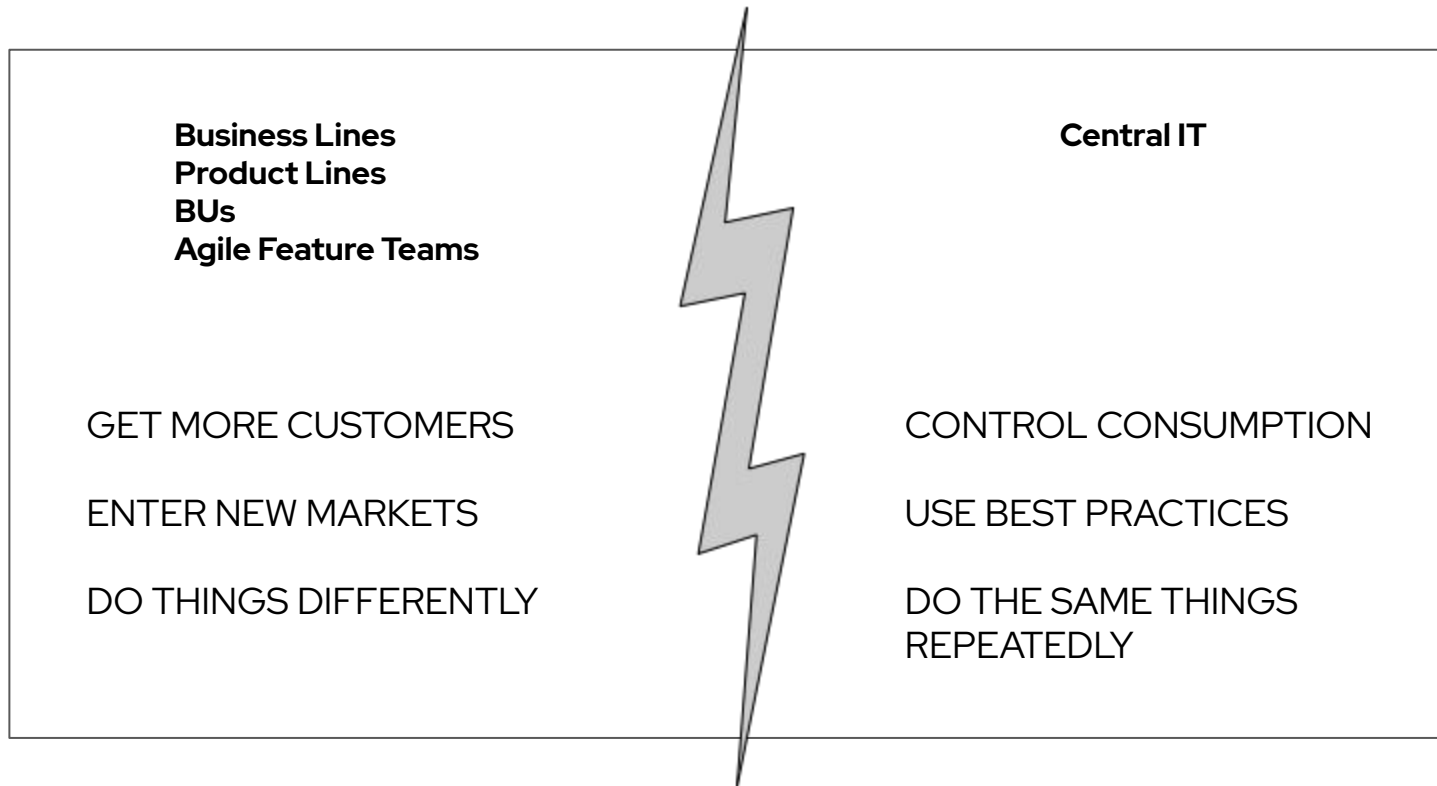
Wall of Confusion

Accelerate
Differentiation
and Creation of
Value



Operational
Excellence
and
Efficiency

Two Governing Theories



Differentiation

Accelerate

Speed and direction. Getting fast feedback from customers.

Novelty

Create opportunities for disruptive innovation.

Niche

Fast feedback to get product fit.

Experiment

Fail fast continuous experimentation.

Incubate

Invest in disruptive solutions with significant potential impact .



Scale

Regulate

Regulate velocity,

Reduce

Variety and Variability

Resilience

Build resilience into the system

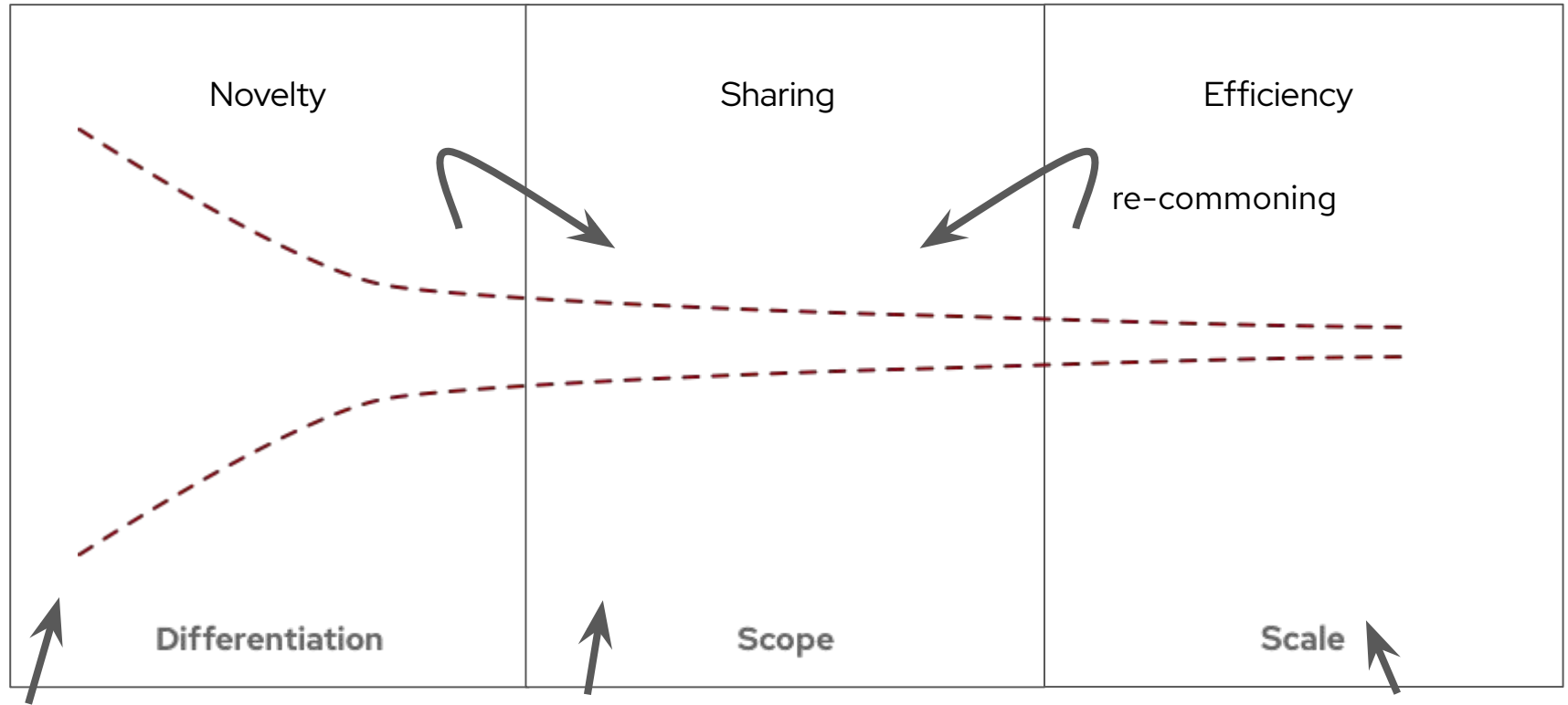
Reuse

Reuse of components

Consolidate

Consolidate infrastructure

Three Games



Some things that
start here...

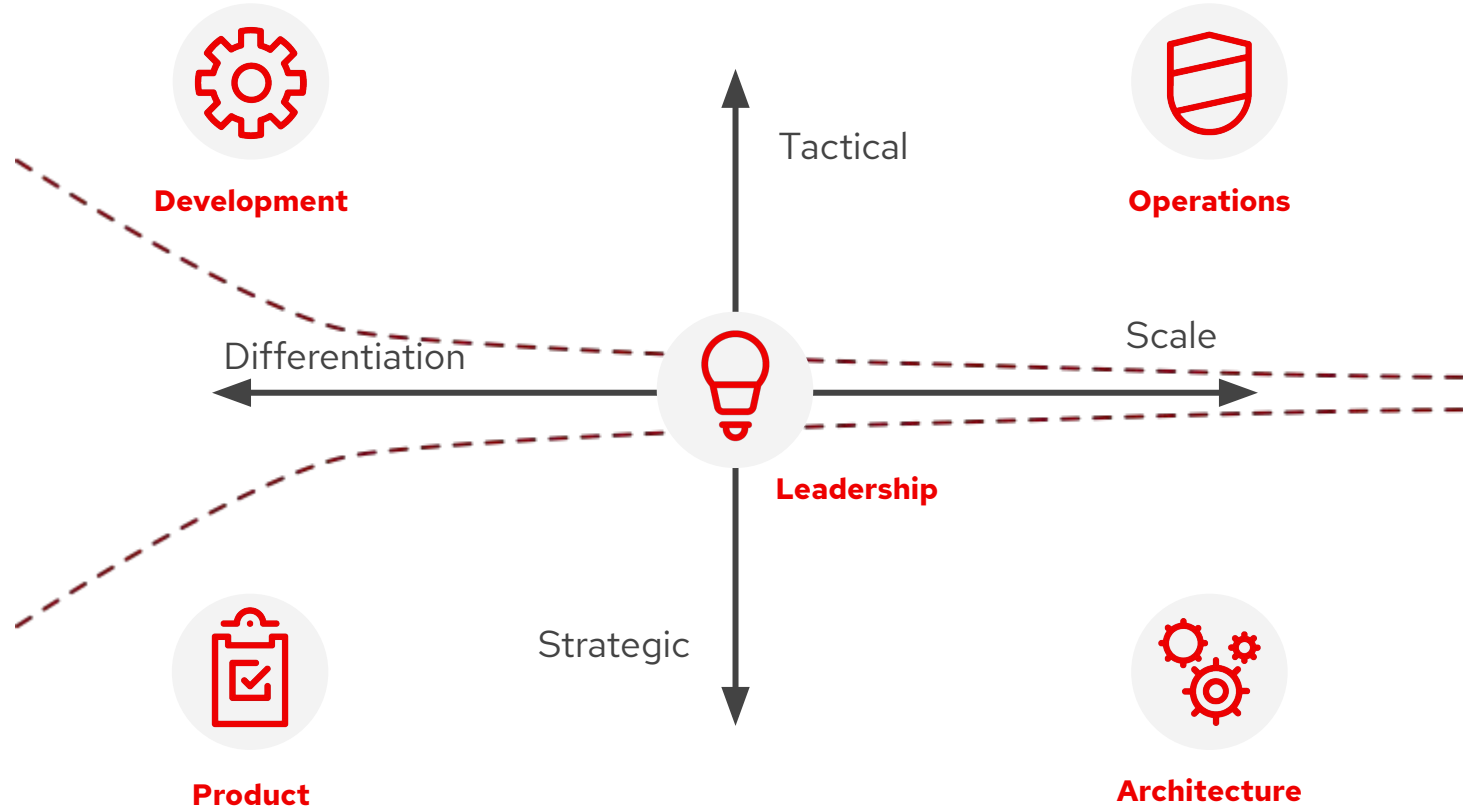
...may eventually be
more valuable here

But are being managed
like they are here

Cloud Native Workflows

Platform as an interface





SRE builds framework modules to implement canonical solutions for the concerned production area. As a result, development teams can focus on the business logic, because the framework already takes care of correct infrastructure use.

North Star

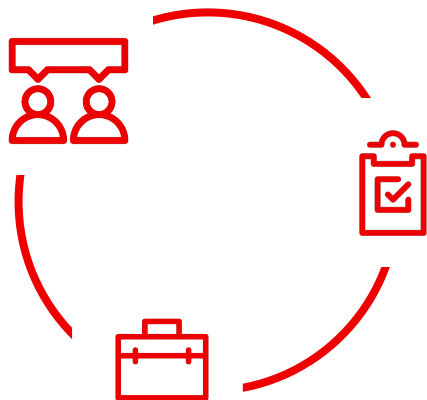
Bringing DDD to the Canonical Data Model

Executive Summary

Speed to business value is becoming a dominant driver in metrics driven technology transformations within organisations today. The reasoning is multifaceted, but primarily this is a direct response to the desire for vastly improved customer experience. A “North Star” Objective is a longer-term, high-level, aspirational goal that motivates, inspires, and uplifts. This positional paper describes concrete techniques to help bring “Domain Driven Design (DDD)”, microservices and distributed integration to an environment that is currently dominated by traditional ESB products and Canonical Data Models.



Accelerating Digital Transformation the Open Source Way



A community of **people** passionate about **helping others modernize** and migrate their **applications** to the hybrid cloud by **building tools and best practices** on how to **break down monoliths, adopt containers, and embrace Kubernetes**.

<https://konveyor.github.io/>

Pathfinder

A workload assessment tool used to determine an applications suitability for running on OpenShift/Kubernetes

build **passing** container none

Setup environment on OpenShift (incl. minishift)

The following commands will create a new project and deploy a mongo, pathfinder-server and pathfinder-ui instance into your environment

```
wget https://raw.githubusercontent.com/redhat-cop/pathfinder/master/pathfinder-full-template.yaml  
  
oc new-project <your-new-project-name>  
oc new-app --template=mongodb-persistent --param=MONGODB_DATABASE=pathfinder  
oc process -f pathfinder-full-template.yaml | oc create -f-
```

Login as admin/admin and start adding customers and apps. Once added assess the apps, review and decide what to do and then hit generate report to see the collective outcome and recommendations.

Usage Tracking

Pathfinder uses analytics to track usage of the application. This can be turned off by setting the "-DdisableTracking=true" JVM option on the Pathfinder UI application. See the file "run-local.sh" under the pathfinder-ui directory for an example of how to do this.

License

The code is made available under the Apache License, Version 2.0

The questions are made available under a Creative Commons Attribution-ShareAlike 4.0 International License.

OpenDevHour

<https://red.ht/OpenDevHour>

Upcoming events

- ▶ True Hybrid Cloud App Dev: Building Manageable Cross Cluster Microservices Architectures, JAN 14 | 16:00 CET
- ▶ More sessions are coming....stay tuned! #AI #Adoptium #Quarkus #GraalVM

Past events

- ▶ DevOps with Containers, DECEMBER 16
- ▶ Securing Microservices, NOV 17 | 16:00 CET
- ▶ Serverless stream processing of Debezium data change events with Kafka Streams and Knative, OCT 20 | 16:00 CEST (CANCELED, postponed to 2021)
- ▶ Supersonic Secure Java with Quarkus, SEP 14
- ▶ Helm for Developers, AUGUST 18
- ▶ Quarkus the black swan of Java, July 23

Markus Eisele

Developer Adoption Lead Red
Hat

markus@redhat.com

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

 linkedin.com/company/red-hat

 youtube.com/user/RedHatVideos

 facebook.com/redhatinc

 twitter.com/RedHat