



THE
DEVELOPER'S
CONFERENCE

Mantendo seu monolito vivo em um ambiente de alta disponibilidade

Guilherme Carreiro - Senior Software Engineer at Red Hat

Paulo Martins - Senior Software Engineer at Red Hat

TDC Porto Alegre

28.11.2019



THE
DEVELOPER'S
CONFERENCE

Mantendo seu monolito vivo em um ambiente de alta disponibilidade

Guilherme Carreiro - Senior Software Engineer at Red Hat

Paulo Martins - Senior Software Engineer at Red Hat

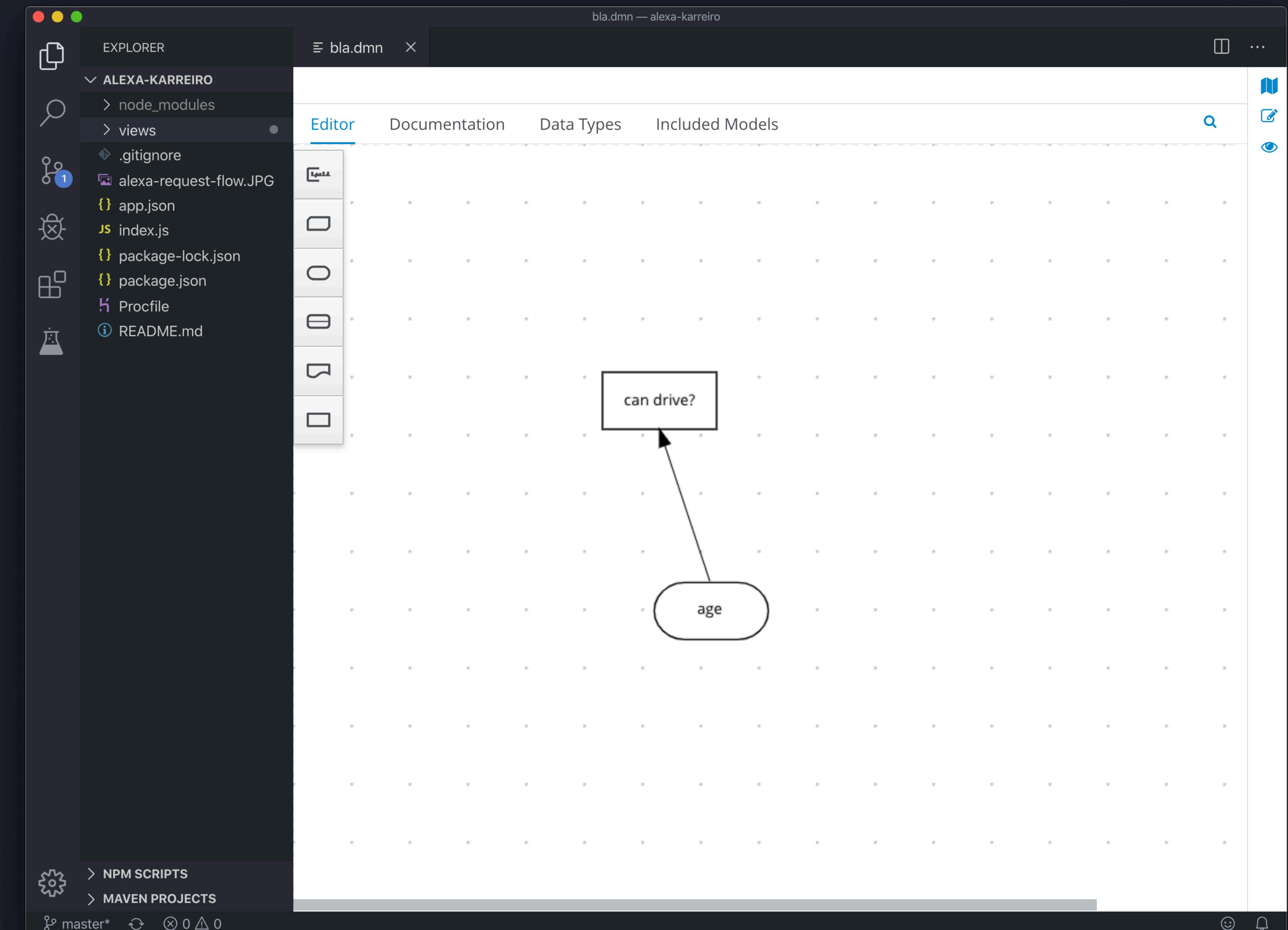
TDC Porto Alegre

28.11.2019



Guilherme Carreiro - @karreiro

- São Paulo, SP
- DMN
- Go language
- Data Privacy
- Weekly blogger - karreiro.com
- Red Hat / Middleware (Decision Tooling)





Paulo Martins - @paulovmr

- Campinas, SP
- Computer Scientist
- React
- Foundation Blog - <https://medium.com/kie-foundation>
- Red Hat / Middleware (Foundation)



localhost

Business Central [Menu](#)

Spaces » workspace » project » master

project

Assets **2** Change Requests **0** Contributors **1** Metrics Settings

All 1-2 of 2 < 1 > of 1 Import Asset Add Asset

 model	DMN	Last modified 1 week ago	Created 1 week ago
 Person	Data Objects	Last modified 1 week ago	Created 1 week ago

The Monolith

16Gb of RAM to run a single app 😍



Monolith features

- Simple deployment
- Built-in tracing
- Simple packaging
- Easy reusability

Monolith features

- Deployment coupling
- Code coupling
- Inflexible scalability
- Lack of elasticity

The Monolith perspective

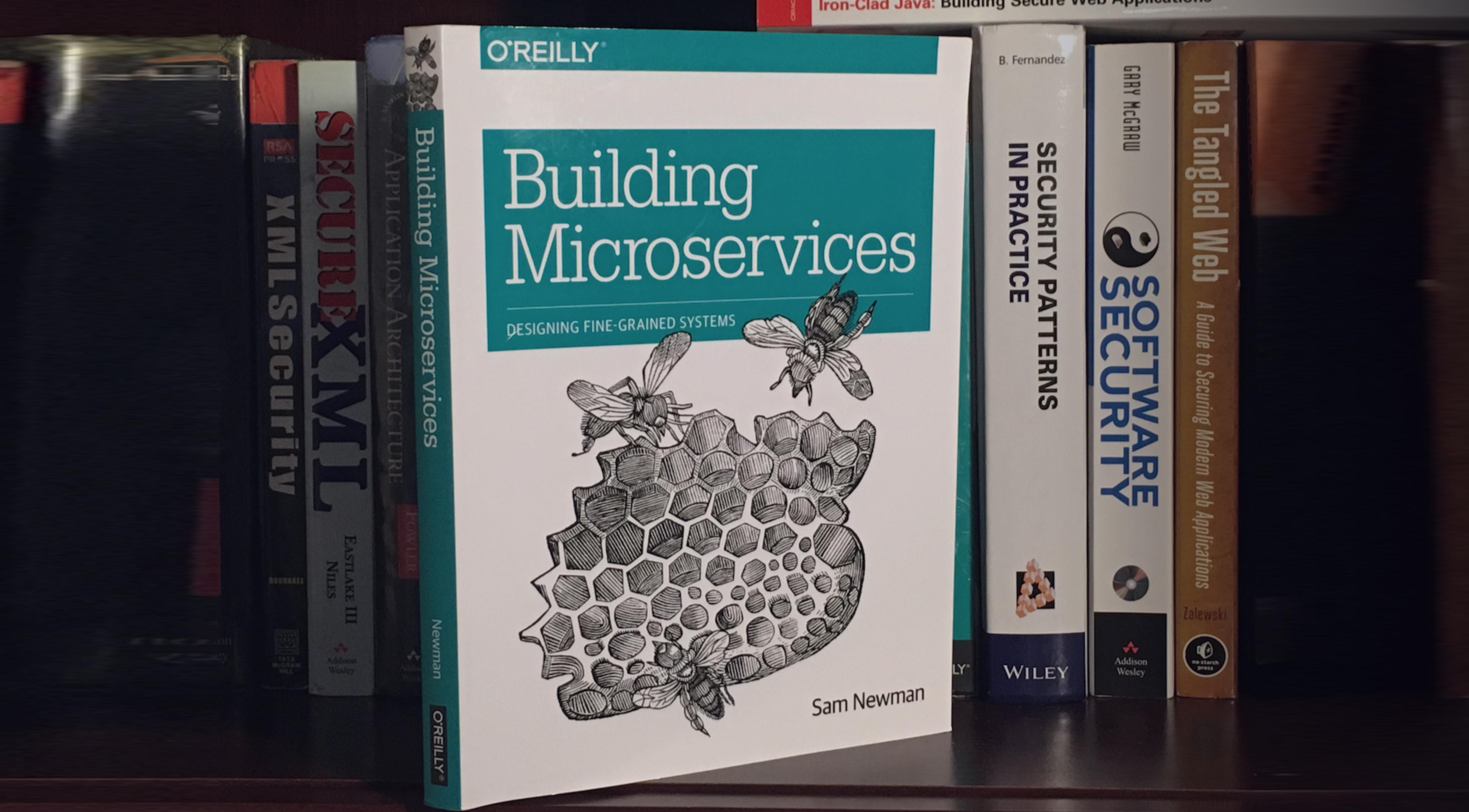
The Monolith perspective

Cloud-native apps and Containers

The Microservices sweet spot

Linux Containers (LXC)

Namespaces, Control Groups, and
Copy-on-write filesystem



Containers with Microservices

- Independent deployments
- Flexible scalability
- Horizontal scalability
- High elasticity

Containers with Microservices

- Clear understanding of the domain
- Tracing
- Orchestration vs. Choreography

The Cloud-native apps perspective

We want it.

...and nobody is gonna die!

The Monolith perspective

The Monolith
perspective

Cloud-native
apps and
Containers

Business Central

Old but gold – not legacy.

Business Central

188Mb WAR:

- more than 8 years of code base
- more than 1 million of LOC
- more than 30 developers
- more than 150 sub-projects
- **very good** balance between coupling and cohesion

Business Central platform

The screenshot shows the Business Central platform interface. At the top, there is a header bar with the title "Business Central" and a URL "localhost:8080/business-central/kie-wb.jsp#HomePerspective%7Corg.kie.workbench.common.screens.home.client.HomePresenter". On the right side of the header are icons for help, settings, and user authentication.

The main content area has a dark background with a faint, abstract pattern. It features a central "Welcome to Business Central" message and a sub-instruction: "Business Central offers a set of flexible tools that support the way you need to work. Select a tool below to get started."

The interface is divided into four main sections, each represented by an icon and a title:

- Design**: Represented by a document icon. Below it, text says "Create and modify [projects](#) and [pages](#)".
- Deploy**: Represented by a gear icon. Below it, text says "Administer [provisioning](#) and [servers](#)".
- Manage**: Represented by a briefcase icon. Below it, text says "Access [process definitions](#), [process instances](#), [tasks](#), [jobs](#) and [executions errors](#)".
- Track**: Represented by a chart icon. Below it, text says "View [task inbox](#), [process reports](#) and [task reports](#)".

Business Central filesystem

The screenshot shows the Business Central web application interface. The title bar reads "Business Central" and the address bar shows "localhost:8080/business-central/kie-wb.jsp#LibraryPerspective%7C\$ProjectScreen,DataModelerEditor?path_uri=default://master@workspace/project/src/main/java/com/workspace/project/Person.java&file_name=Person.java&has_version_support=true,AddAssetsS...". The main navigation bar includes "Business Central", "Menu", and "admin". Below the navigation is a breadcrumb trail: "Spaces > workspace > project > master". The main content area is titled "project" and displays two assets: "model" (DMN) and "Person" (Data Objects). Both assets were last modified today and created today. Action buttons for "Test", "Build", "Deploy", and "View Alerts" are visible at the top right of the asset list.

Asset Type	File Name	Last Modified	Created
DMN	model	Today	Today
Data Objects	Person	Today	Today

Business Central assets

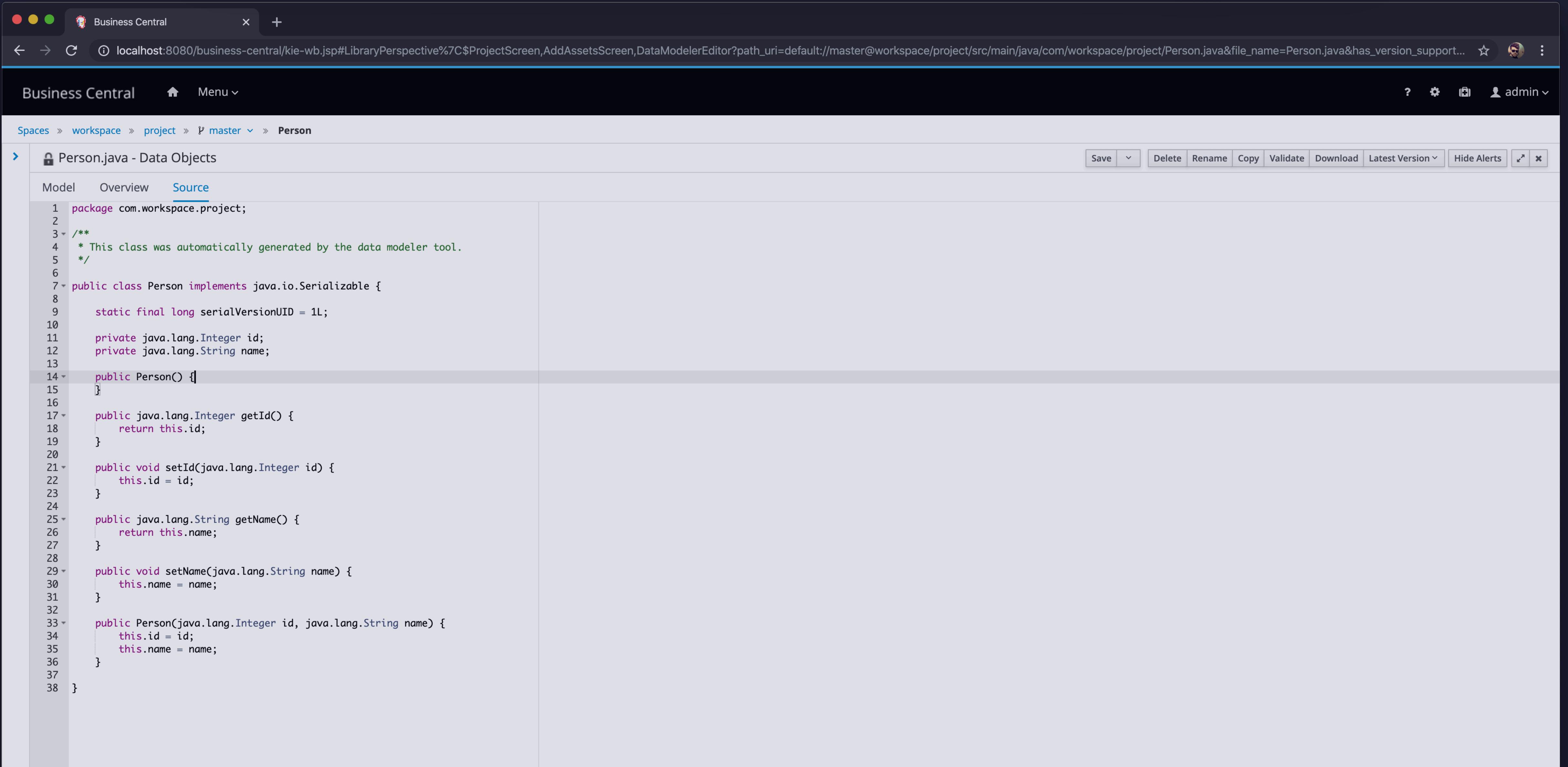
Screenshot of the Business Central interface showing the "Add Asset" screen.

The title bar shows "Business Central" and the URL "localhost:8080/business-central/kie-wb.jsp#LibraryPerspective%7C\$ProjectScreen,AddAssetsScreen". The top navigation includes "Spaces", "workspace", "project", "master", "Menu", and "admin".

The main content area is titled "Add Asset" with a "Cancel" button. A dropdown menu "All" is open. The screen displays a grid of asset types:

Icon	Type	Description
	Business Process	Process
	Business Process (legacy)	Process
	Data Object Model	
	Decision Table (Spreadsheet)	Decision
	DMN	Decision
	DRL file	Decision
	DSL definition	Decision
	Enumeration	Model
	Form	Form
	Global Variable(s)	Decision
	Guided Decision Table	Decision
	Guided Decision Table Graph	Decision
	Guided Decision Tree	Decision
	Guided Rule	Decision
	Guided Rule Template	Decision
	Guided Score Card	Decision
	Package	Others
	Score Card (Spreadsheet)	Decision
	Solver configuration	Optimization
	Test Scenario	Decision
	Test Scenario (Legacy)	Decision
	Work Item definition	Others
	Case definition (legacy)	Process

Business Central editors (Data Objects)



The screenshot shows the Business Central Data Modeler Editor interface. The title bar reads "Business Central". The URL in the address bar is "localhost:8080/business-central/kie-wb.jsp#LibraryPerspective%7C\$ProjectScreen,AddAssetsScreen,DataModelerEditor?path_uri=default://master@workspace/project/src/main/java/com/workspace/project/Person.java&file_name=Person.java&has_version_support...". The main content area displays the Java code for the Person class:

```
1 package com.workspace.project;
2
3 /**
4  * This class was automatically generated by the data modeler tool.
5  */
6
7 public class Person implements java.io.Serializable {
8
9     static final long serialVersionUID = 1L;
10
11    private java.lang.Integer id;
12    private java.lang.String name;
13
14    public Person() {
15    }
16
17    public java.lang.Integer getId() {
18        return this.id;
19    }
20
21    public void setId(java.lang.Integer id) {
22        this.id = id;
23    }
24
25    public java.lang.String getName() {
26        return this.name;
27    }
28
29    public void setName(java.lang.String name) {
30        this.name = name;
31    }
32
33    public Person(java.lang.Integer id, java.lang.String name) {
34        this.id = id;
35        this.name = name;
36    }
37
38 }
```

The code editor has tabs for "Model", "Overview", and "Source", with "Source" being the active tab. At the top right of the code area are buttons for "Save", "Delete", "Rename", "Copy", "Validate", "Download", "Latest Version", and "Hide Alerts". The status bar at the bottom shows "File: Person.java" and "Line 1, Col 1".

Business Central editors (DMN models)

Screenshot of the Business Central Data Modeler Editor interface for a DMN model named "model.dmn".

The interface includes:

- Header:** Business Central window title, URL (localhost:8080/business-central/kie-wb.jsp#LibraryPerspective%7C\$ProjectScreen,DataModelerEditor?path_uri=default://master@workspace/project/src/main/java/com/workspace/project/Person.java&file_name=Person.java&has_version_support=true,AddAssetsS...), and user admin.
- Breadcrumbs:** Spaces > workspace > project > master > model.
- Left Sidebar:** Decision Navigator showing "Decision Graphs" with "model" containing "Decision-1" and "InputData-1".
- Right Main Area:** "model.dmn - DMN" tab, Model tab selected. The canvas displays a decision graph with an oval node labeled "InputData-1" and a rectangular node labeled "Decision-1" connected by a directed edge.
- Toolbar:** Top right features standard file operations (Save, Delete, Rename, Copy, etc.) and a "View Alerts" button.

Business Central runtime

The screenshot shows the Business Central runtime management interface. The left sidebar lists 'SERVER CONFIGURATIONS' with a 'server' configuration selected. The main panel displays the 'Status' tab for the 'server' configuration, which includes sections for 'Capabilities' (Decision, Process checked, Planner unchecked) and 'DEPLOYMENT UNITS'. A deployment unit named 'com.myspace:aaa:1.0.0-SNAPSHOT' is listed under 'DEPLOYMENT UNITS' and is highlighted with a blue bar. The right side of the screen shows a message 'No Remote Servers' with a note: 'This Deployment Unit is currently not in use by any Remote Servers.' and a 'Refresh' button.

Business Central

localhost:8080/business-central/kie-wb.jsp#

Business Central

Menu

admin

SERVER CONFIGURATIONS

+ New Server Configuration

server

Capabilities:

- Decision
- Process
- Planner

DEPLOYMENT UNITS

+ Add Deployment Unit

com.myspace:aaa:1.0.0-SNAPSHOT

REMOTE SERVERS

[Group Id:com.myspace | Artifact Id:aaa]

Status Version Configuration Process Configuration

Start Stop Deactivate Remove Refresh

No Remote Servers

This Deployment Unit is currently not in use by any Remote Servers.

Refresh

Monolith-to-Cloud challenges

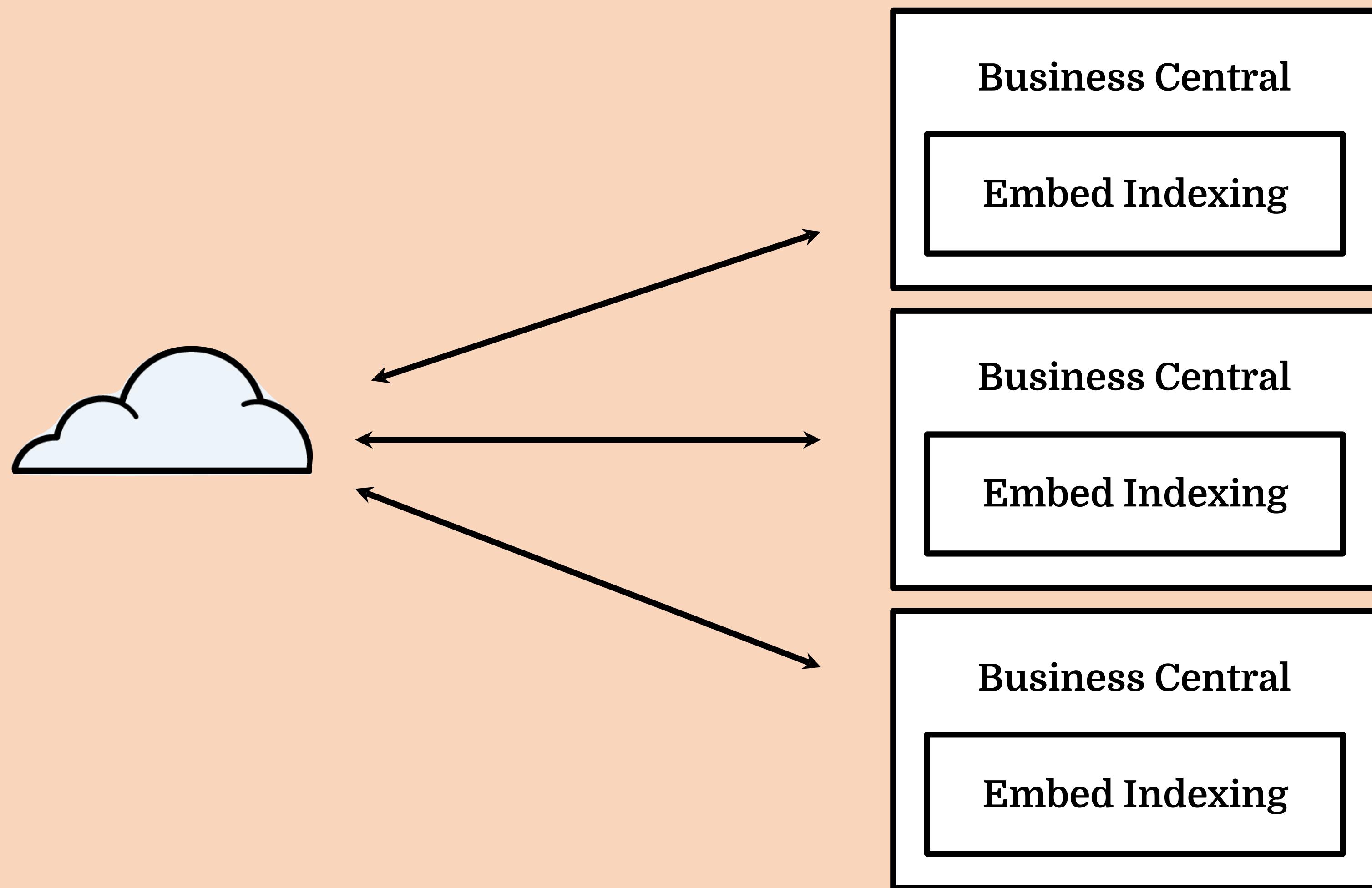
Here we go!

Monolith-to-Cloud challenges

1. Redundant processing
2. Global lock
3. Keep the programming paradigm

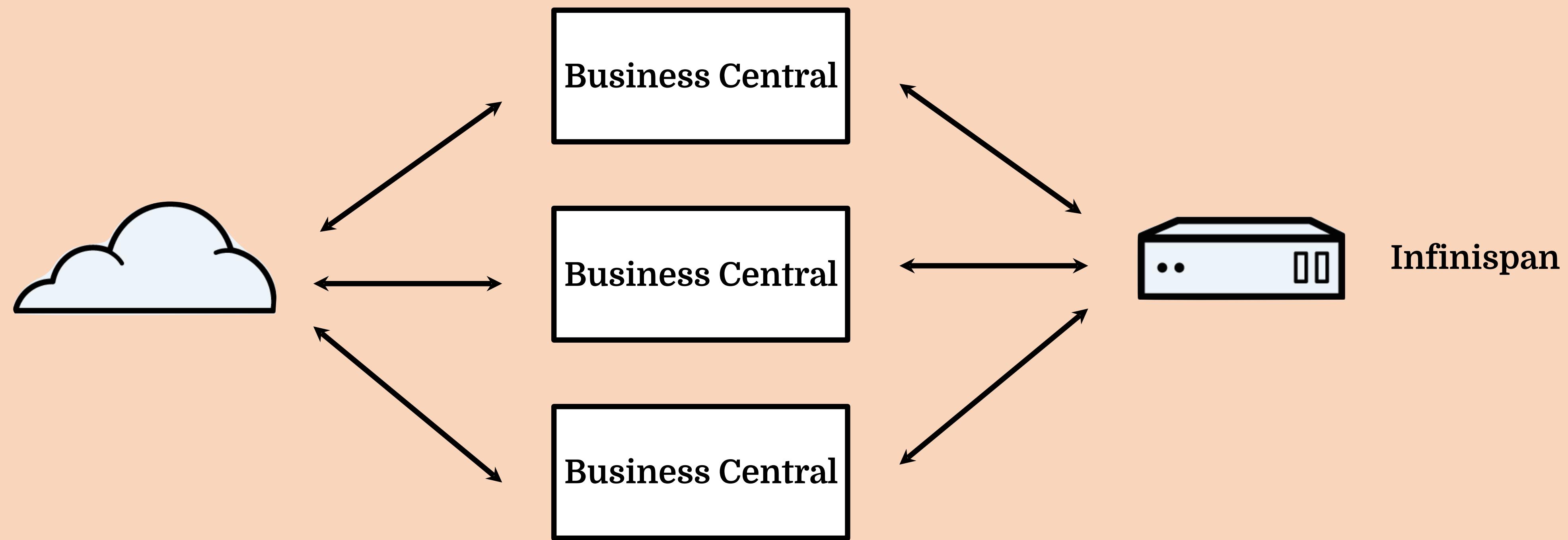
1. Redundant processing

DRYing the monolith



Indexing as a service

Ctrl + Alt + M



Lighter monolith and fine tuning

Win-win 

2. Global lock

Do not stop the world!!!

Bounded Contexts

“A context means a specific responsibility. A bounded context means that responsibility is enforced with explicit boundaries” *

* <https://blog.sapiensworks.com/post/2012/04/17/DDD-The-Bounded-Context-Explained.aspx>

Bounded Contexts



The locked contexts

Smaller contexts, smaller locks

3. Keep the programming paradigm

The agnostic code base

http://localhost:8180/...

This screenshot shows the Business Central interface for the 'HR' space. The top navigation bar includes 'Business Central', 'Menu', and user information for 'admin'. Below the header, the breadcrumb navigation shows 'Spaces > HR'. A prominent red 'Delete Space' button is located in the top right corner. The main content area displays a large 'Nothing Here' message with a plus sign icon, indicating no projects are available. Below this message, a note states: 'There are currently no projects available for you to view or edit. To get started, create a new project or import one.' At the bottom, there are three buttons: 'Add Project' (highlighted in blue), 'Try Samples', and 'Import Project'.

http://localhost:8180/...

This screenshot shows the Business Central interface for the 'HR' space, but with a different user logged in ('admin2'). The top navigation bar includes 'Business Central', 'Menu', and user information for 'admin2'. The rest of the interface is identical to the first screenshot, displaying an empty space with a 'Nothing Here' message, a note about starting a project, and three action buttons at the bottom.

CDI Events 101

CDI Events 101

```
class NewProjectEvent {  
    Project project;  
  
    NewProjectEvent(Project p) {  
        this.project = project;  
    }  
  
    Project getProject() {  
        return this.project;  
    }  
}
```

CDI Events 101

```
class NewProjectEvent {  
    Project project;  
  
    NewProjectEvent(Project p) {  
        this.project = project;  
    }  
  
    Project getProject() {  
        return this.project;  
    }  
}
```

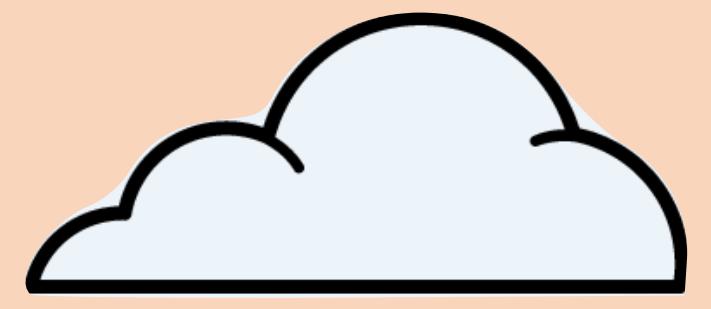
```
class ProjectsController {  
    @Inject  
    Event<NewProjectEvent> newProjectEvent;  
  
    void createProject(Project project) {  
        fileSystem.save(project);  
        newProjectEvent.fire(new NewProjectEvent(project))  
    }  
}
```

CDI Events 101

```
class NewProjectEvent {  
    Project project;  
  
    NewProjectEvent(Project p) {  
        this.project = project;  
    }  
  
    Project getProject() {  
        return this.project;  
    }  
}
```

```
class ProjectsController {  
    @Inject  
    Event<NewProjectEvent> newProjectEvent;  
  
    void createProject(Project project) {  
        fileSystem.save(project);  
        newProjectEvent.fire(new NewProjectEvent(project))  
    }  
}
```

```
class ProjectScreen {  
    onNewProject(@Observes NewProjectEvent e) {  
        updateProjectList(e.getProject())  
    }  
}
```



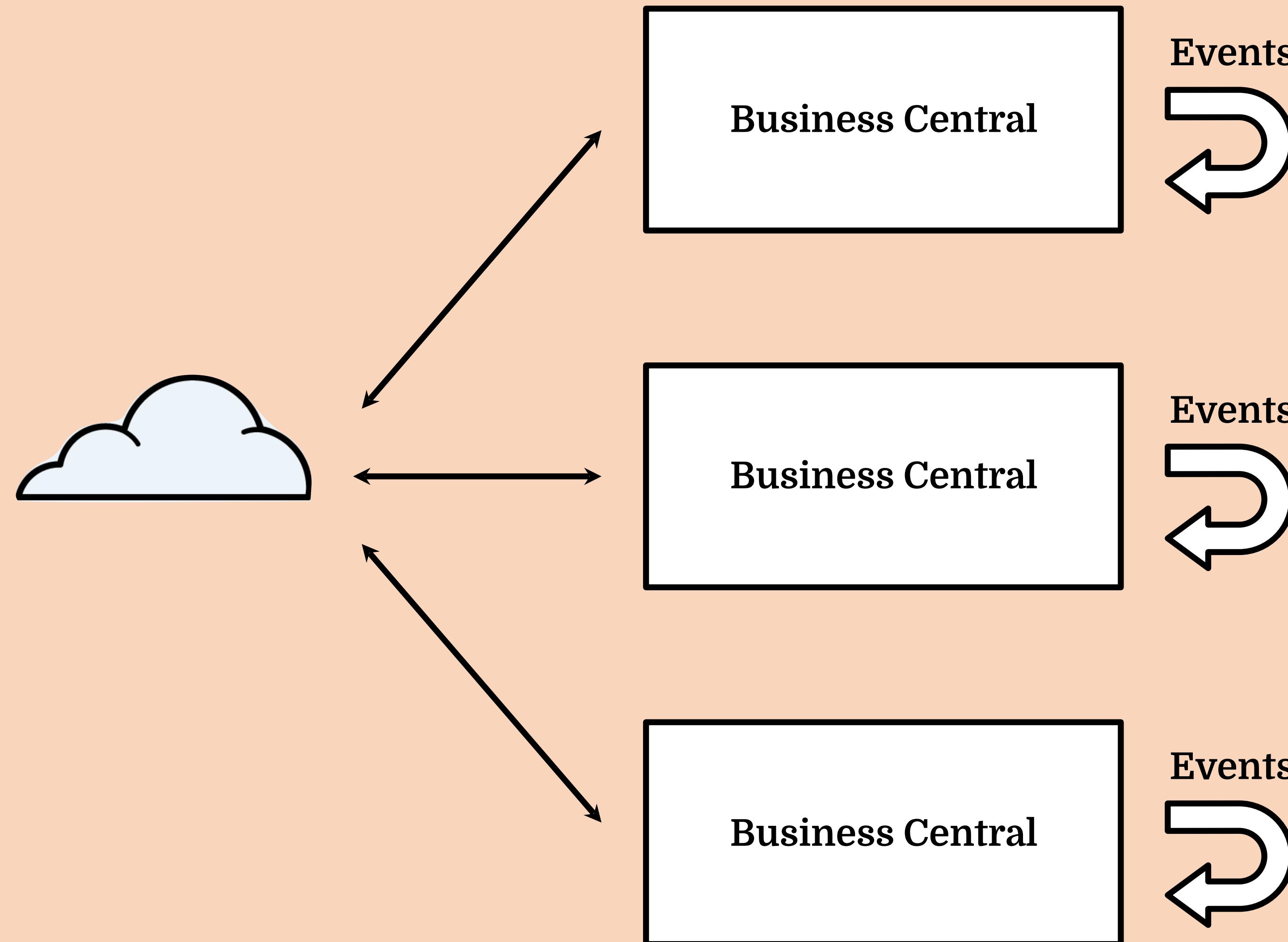
Events

http://localhost:8180/...

This screenshot shows the Business Central interface for the 'HR' space. The top navigation bar includes 'Business Central', 'Menu', and user information for 'admin'. Below the header, the breadcrumb navigation shows 'Spaces > HR'. A prominent red 'Delete Space' button is located in the top right corner. The main content area displays a large 'Nothing Here' message with a plus sign icon, indicating no projects are available. Below this message, a note states: 'There are currently no projects available for you to view or edit. To get started, create a new project or import one.' At the bottom, there are three buttons: 'Add Project' (highlighted in blue), 'Try Samples', and 'Import Project'.

http://localhost:8180/...

This screenshot shows the Business Central interface for the 'HR' space, but with a different user logged in, 'admin2'. The top navigation bar includes 'Business Central', 'Menu', and user information for 'admin2'. Below the header, the breadcrumb navigation shows 'Spaces > HR'. A red 'Delete Space' button is located in the top right corner. The main content area displays a large 'Nothing Here' message with a plus sign icon, indicating no projects are available. Below this message, a note states: 'There are currently no projects available for you to view or edit. To get started, create a new project or import one.' At the bottom, there are three buttons: 'Add Project' (highlighted in blue), 'Try Samples', and 'Import Project'.



http://localhost:8180/...

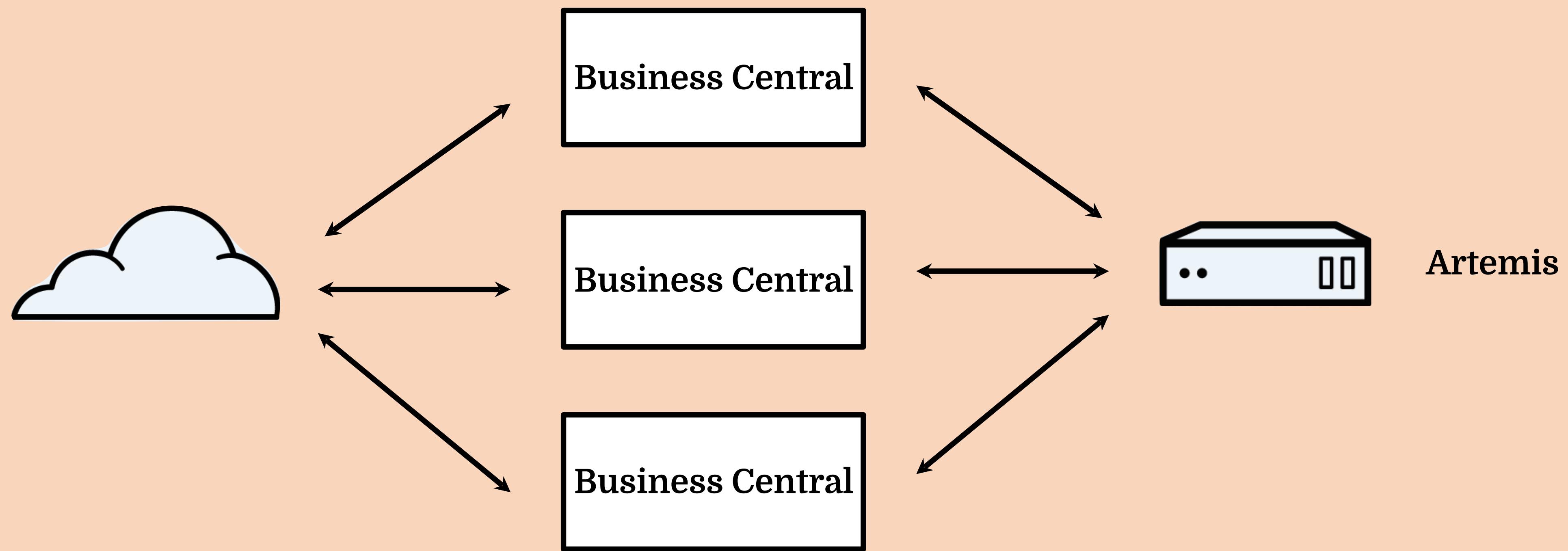
This screenshot shows the Business Central interface running on port 8180. The browser title bar reads "Business Central" and the URL is "localhost:8180/business-central/kie-wb.jsp#LibraryPerspective%7C\$LibraryScreen". The main content area is titled "HR" and displays a message "Nothing Here" with a large plus sign icon. Below the message, a note states: "There are currently no projects available for you to view or edit. To get started, create a new project or import one." At the bottom, there are three buttons: "Add Project" (highlighted in blue), "Try Samples", and "Import Project". The top navigation bar includes links for "Spaces", "Menu", and user "admin".

http://localhost:8280/...

This screenshot shows the Business Central interface running on port 8280. The browser title bar reads "Business Central" and the URL is "localhost:8280/business-central/kie-wb.jsp#LibraryPerspective%7C\$LibraryScreen". The main content area is titled "HR" and displays a message "Nothing Here" with a large plus sign icon. Below the message, a note states: "There are currently no projects available for you to view or edit. To get started, create a new project or import one." At the bottom, there are three buttons: "Add Project" (highlighted in blue), "Try Samples", and "Import Project". The top navigation bar includes links for "Spaces", "Menu", and user "admin2".

Distributed CDI Events

- Publish–subscribe pattern



Distributed CDI Events

- Publish–subscribe pattern
- Clustered events

```
@Clustered
class NewProjectEvent {
    Project project;

    NewProjectEvent(Project p) {
        this.project = project;
    }

    Project getProject() {
        return this.project;
    }
}
```

Distributed CDI Events

- Publish–subscribe pattern
- Clustered events
- Event propagation

```
@ApplicationScoped
class ClusterEventObserver {
    String nodeId = UUID.randomUUID().toString();

    // ...

    void consumeMessage(Event<Object> eventBus,
                        EventMessage message) {
        if (!message.getNodeId().equals(nodeId)) {
            eventBus.fire(fromJSON(message));
        }
    }

    void observeAllEvents(@Observes Object event,
                          EventMetadata metaData) {
        if (shouldObserveThisEvent(event, metaData)) {
            broadcast(event);
        }
    }
    // ...
}
```

```
@ApplicationScoped
class ClusterEventObserver {
    String nodeId = UUID.randomUUID().toString();

    // ...

    void consumeMessage(Event<Object> eventBus,
                        EventMessage message) {
        if (!message.getNodeId().equals(nodeId)) {
            eventBus.fire(fromJSON(message));
        }
    }

    void observeAllEvents(@Observes Object event,
                          EventMetadata metaData) {
        if (shouldObserveThisEvent(event, metaData)) {
            broadcast(event);
        }
    }
    // ...
}
```

```
@ApplicationScoped
class ClusterEventObserver {
    String nodeId = UUID.randomUUID().toString();

    // ...

    void consumeMessage(Event<Object> eventBus,
                        EventMessage message) {
        if (!message.getNodeId().equals(nodeId)) {
            eventBus.fire(fromJSON(message));
        }
    }

    void observeAllEvents(@Observes Object event,
                          EventMetadata metaData) {
        if (shouldObserveThisEvent(event, metaData)) {
            broadcast(event);
        }
    }
    // ...
}
```

```
@ApplicationScoped
class ClusterEventObserver {
    String nodeId = UUID.randomUUID().toString();

    // ...

    void consumeMessage(Event<Object> eventBus,
                        EventMessage message) {
        if (!message.getNodeId().equals(nodeId)) {
            eventBus.fire(fromJSON(message));
        }
    }

    void observeAllEvents(@Observes Object event,
                          EventMetadata metaData) {
        if (shouldObserveThisEvent(event, metaData)) {
            broadcast(event);
        }
    }
    // ...
}
```

```
@ApplicationScoped
class ClusterEventObserver {
    String nodeId = UUID.randomUUID().toString();

    // ...

    void consumeMessage(Event<Object> eventBus,
                        EventMessage message) {
        if (!message.getNodeId().equals(nodeId)) {
            eventBus.fire(fromJSON(message));
        }
    }

    void observeAllEvents(@Observes Object event,
                          EventMetadata metaData) {
        if (shouldObserveThisEvent(event, metaData)) {
            broadcast(event);
        }
    }

    // ...
}
```

http://localhost:8180/...

This screenshot shows the Business Central interface running on port 8180. The browser title bar reads "Business Central" and the address bar shows "localhost:8180/business-central/kie-wb.jsp#LibraryPerspective%7C\$LibraryScreen". The main content area is titled "HR" and displays a message "Nothing Here" with a large plus sign icon. Below the message, a note states: "There are currently no projects available for you to view or edit. To get started, create a new project or import one." At the bottom, there are three buttons: "Add Project" (highlighted in blue), "Try Samples", and "Import Project". The top navigation bar includes links for "Business Central", "Menu", "Spaces > HR", and "Delete Space".

http://localhost:8280/...

This screenshot shows the Business Central interface running on port 8280. The browser title bar reads "Business Central" and the address bar shows "localhost:8280/business-central/kie-wb.jsp#LibraryPerspective%7C\$LibraryScreen". The main content area is titled "HR" and displays a message "Nothing Here" with a large plus sign icon. Below the message, a note states: "There are currently no projects available for you to view or edit. To get started, create a new project or import one." At the bottom, there are three buttons: "Add Project" (highlighted in blue), "Try Samples", and "Import Project". The top navigation bar includes links for "Business Central", "Menu", "Spaces > HR", and "Delete Space".

The Monolith
perspective

Cloud-native
apps and
Containers

The Monolith perspective

Cloud-native
apps and
Containers

Business
Central

Monolith to
Cloud
challenges

Cloud-ready

The pragmatic good enough.

Co-Transformation to Cloud-Native Applications

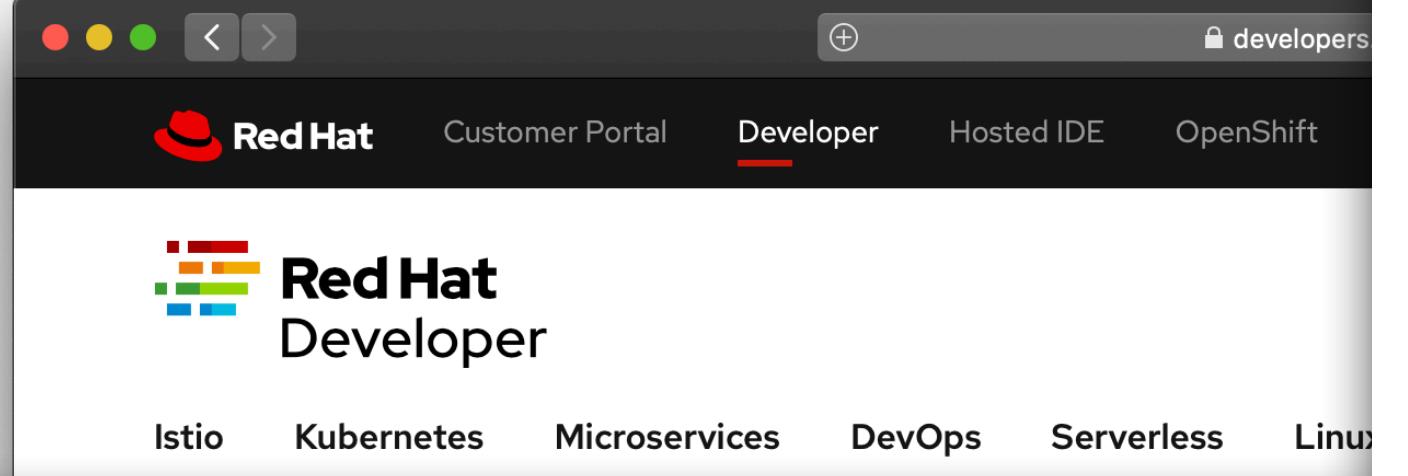
Development Experiences and Experimental Evaluation

Josef Spillner¹, Yessica Bogado², Walter

¹*Service Prototyping Lab, Zurich University of*

²*Information and Communication Technology Center, It*

josef.spillner@zhaw.ch {yessica.bogado.walter}@zhaw.ch



About the Complexity to Transfer Cloud Applications at Runtime and how Container Platforms can Contribute?

Nane Kratzke

Lübeck University of Applied Sciences,
Center for Communication, Systems and Applications (CoSA), Germany,
nane.kratzke@fh-luebeck.de

Review

A Brief History of Cloud Application Architecture

Nane Kratzke 

Lübeck University of Applied Sciences, Department of Electrical Engineering and Computer Science
23562 Lübeck, Germany; nane.kratzke@fh-luebeck.de

A screenshot of a web browser displaying the entando.com website. The top navigation bar includes links for Back to overview, WHAT WE DO, TECHNOLOGY, and SUPPORT. The main content area features the entando logo and the text "What is the Difference between Cloud-Ready and Cloud-Native?". At the bottom, there is a footer with the text "Tuesday, April 17, 2018 | ENGINEERING_TEAM" and a link to "Engineering".

Abstract. Cloud-native applications are often designed for only one specific cloud infrastructure or platform. The effort to port such kind of applications into a different cloud is usually a laborious one time exercise. Modern Cloud-native application architecture approaches make use of popular elastic container platforms (Apache Mesos, Kubernetes, Docker Swarm). These kind of platforms contribute to a lot of existing cloud engineering requirements. This given, it astonishes that these kind of platforms (already existing and open source available) are not considered more consequently for multi-cloud solutions. These platforms provide inherent multi-cloud support but this is often overlooked. This paper presents a software prototype and shows how Kubernetes and Docker Swarm clusters could be successfully transferred at runtime across public cloud infrastructures of Google (Google Compute Engine), Microsoft (Azure) and Amazon (EC2) and further cloud infrastructures like OpenStack. Additionally, software engineering lessons learned are derived and some astonishing performance data of the mentioned cloud infrastructures is presented that could be used for further optimizations of IaaS transfers of Cloud-native applications.

Keywords: cloud-native application, multi-cloud, elastic platform, container, portability, transferability, MAPE, AWS, GCE, Azure, OpenStack

Co-Transformation to Cloud-Native Applications

Development Experiences and Experimental Evaluation

Josef Spillner¹, Yessica Bogado², Walte

¹Service Prototyping Lab, Zurich University of Applied Sciences.

“Cloud-ready apps are applications initially developed to run on static environments, but modified to take advantage of a limited number of cloud features”

About the Complexity to Transfer Cloud Applications at Runtime and how Container Platforms can Contribute?

out modified to take

Abstract. Cloud-native applications are often designed for only one specific cloud infrastructure or platform. The effort to port such kind of applications into a different cloud is usually a laborious one time exercise. Modern Cloud-native application architecture approaches make use of popular elastic container platforms (Apache Mesos, Kubernetes, Docker Swarm). These kind of platforms contribute to a lot of existing cloud engineering requirements. This given, it astonishes that these kind of platforms (already existing and open source available) are not considered more consequently for multi-cloud solutions. These platforms provide inherent multi-cloud support but this is often overlooked. This paper presents a software prototype and shows how Kubernetes and Docker Swarm clusters could be successfully transferred at runtime across public cloud infrastructures of Google (Google Compute Engine), Microsoft (Azure) and Amazon (EC2) and further cloud infrastructures like Open-Stack. Additionally, software engineering lessons learned are derived and some astonishing performance data of the mentioned cloud infrastructures is presented that could be used for further optimizations of IaaS transfers of Cloud-native applications.

Keywords: cloud-native application, multi-cloud, elastic platform, container, portability, scalability, MADE, AWS, GCE, Azure, OpenShift

Cloud-ready

1. Each instance is agnostic of environment
2. Redundant tasks as a service
3. Horizontal scalability

The Monolith perspective

Monolith to Cloud challenges

Cloud-native apps and Containers

Cloud-ready enough

Business Central

The Monolith perspective

Monolith to Cloud challenges

Cloud-native apps and Containers

Cloud-ready enough

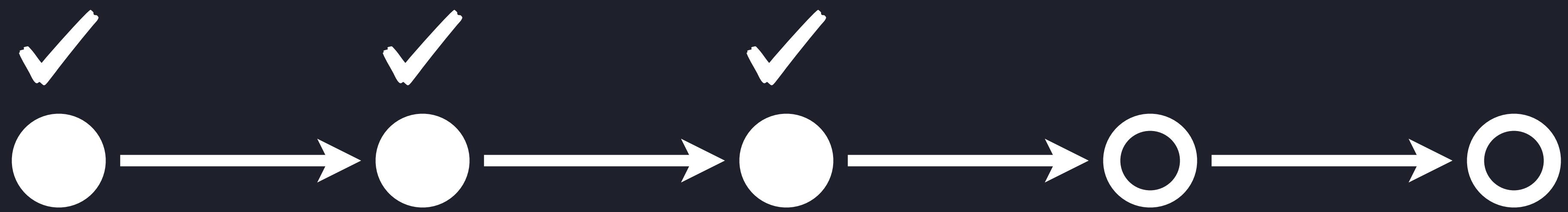
Business Central

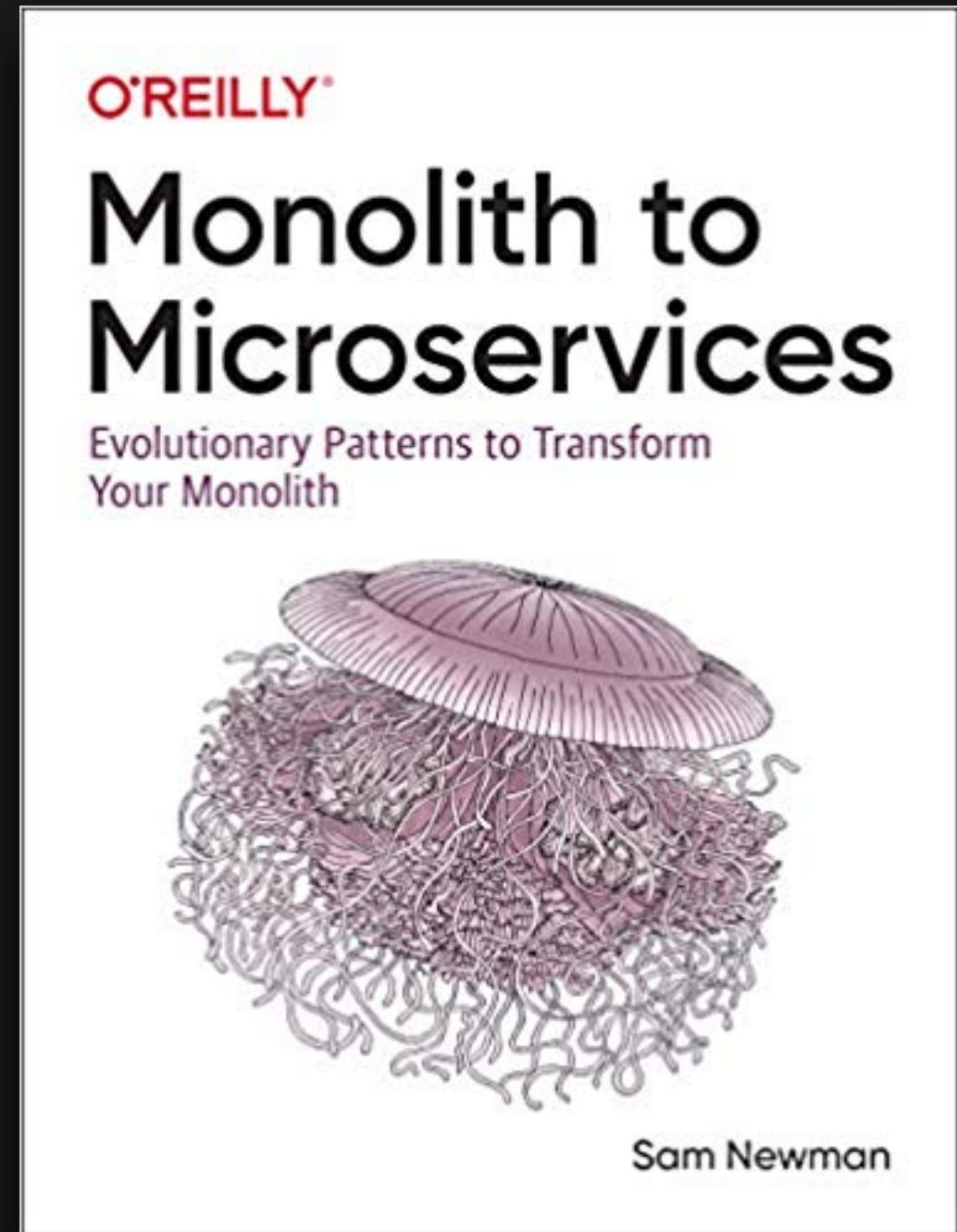
Look at your app

Look at your app

It's not that ugly, neither that good – probably.





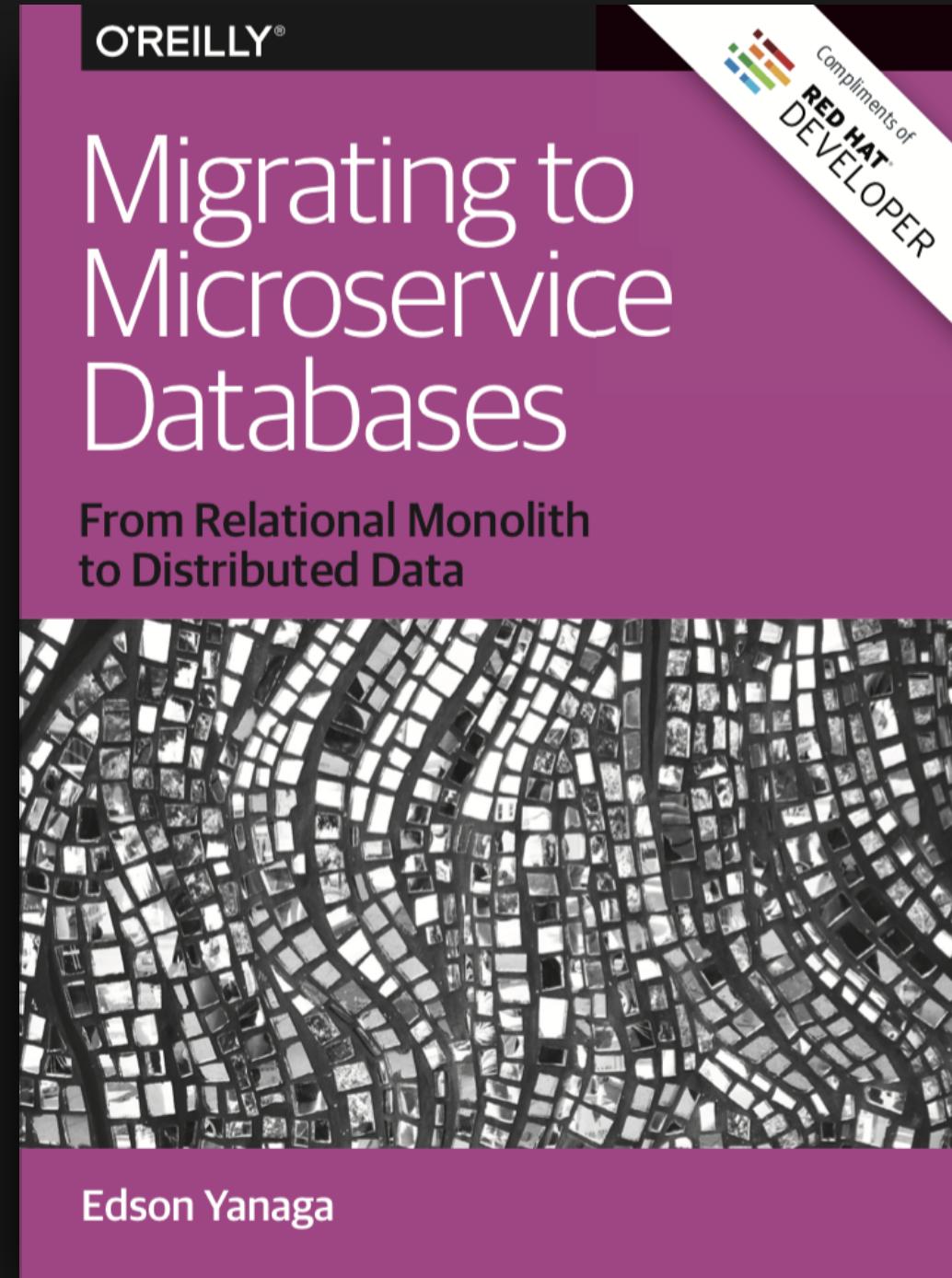


Monolith to Microservices

by Sam Newman

Release Date: November 2019

<http://shop.oreilly.com/product/0636920233169.do>

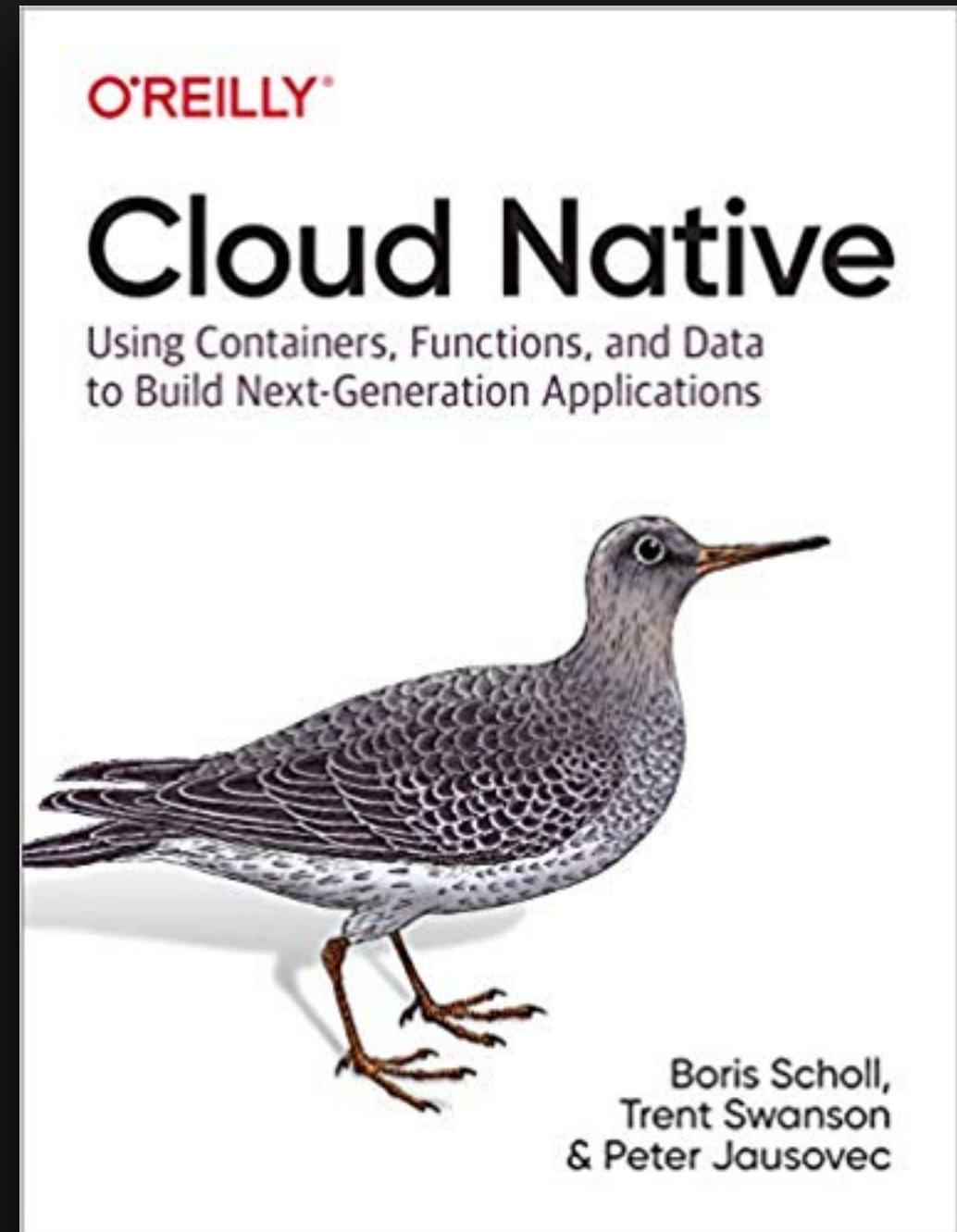


Migrating to Microservice Databases

by Edson Yanaga

Release Date: April 2017

<https://developers.redhat.com/books/migrating-mi...>

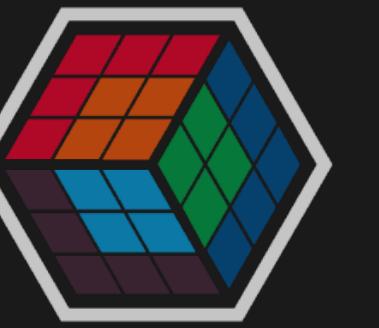


Cloud Native

by Peter Jausovec, Trent Swanson, Boris Scholl

Release Date: August 2019

<http://shop.oreilly.com/product/0636920261704.do>



THE
DEVELOPER'S
CONFERENCE

Thank you! 😊

@karreiro - karreiro.com/talks

@paulovmr