

# Application Modernization – End-to-End Example

## From Java EE in 2008 to Quarkus in 2021

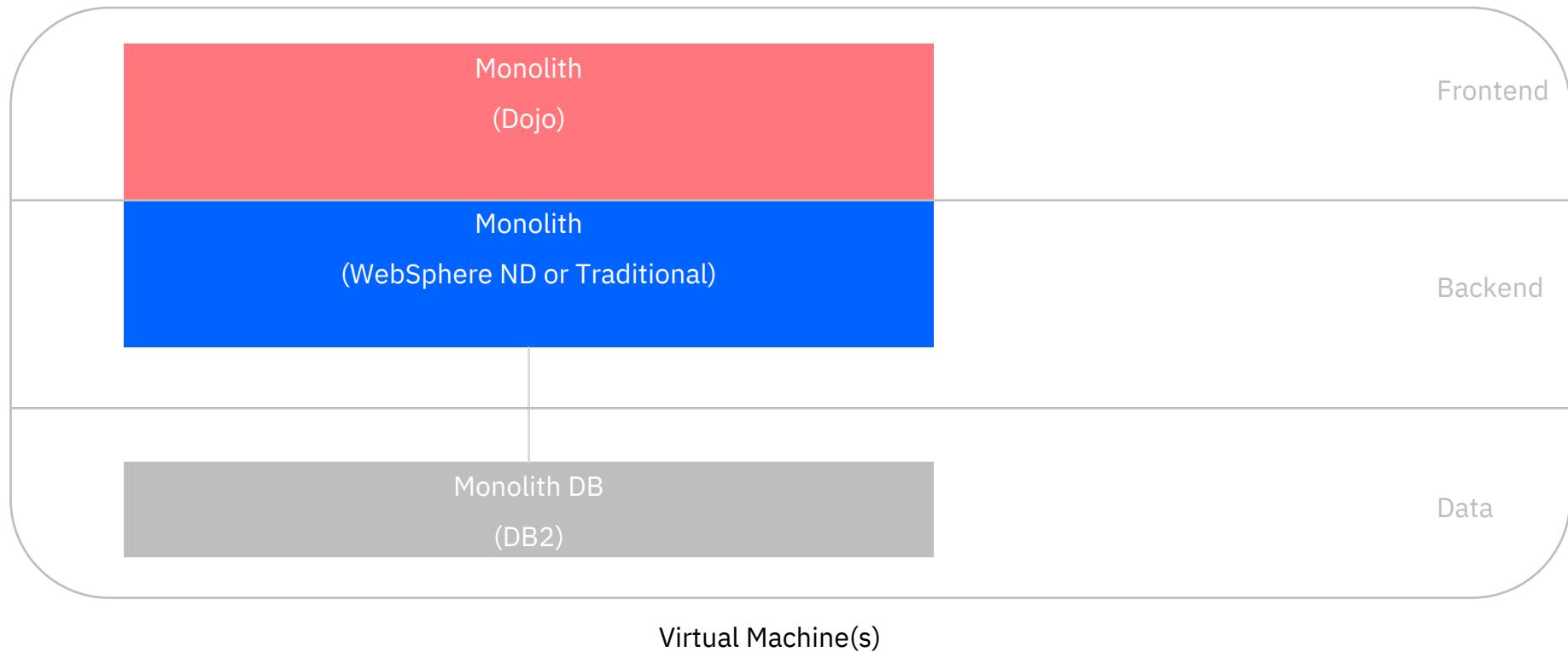
Niklas Heidloff  
Developer Advocate, IBM  
@nheidloff

Draft only  
Work in progress ...

# Step 1 / Starting Point: Monolith running in VM(s)

Description	How did we get here?	Used Technologies	Containers
Java EE 6 application	N/A – starting point	WebSphere ND or Traditional	None
Built around 2008		EJBs	Two Virtual Machines
WebSphere 8.5.5		REST APIs (BFF, not RESTful)	
Typical three tier architecture		Db2	
No CI/CD		Dojo Toolkit	

# Step 1 / Starting Point: Monolith running in VM(s)



# Step 1 / Starting Point: Monolith running in VM(s)

Functionality:  
Browse Products

**Electronic and Movie Depot**

Shop Cart Order History Account

Entertainment ► Electronics ►

**Movies**

Product	Price
Star Wars Episode VI: Return of the Jedi	29.99
Star Wars Episode V: The Empire Strikes Back	29.99
Star Wars Episode IV: A New Hope	29.99
Sony DVD Player	100

Current Shopping Cart

- Empire Strikes Back (2)
- DVD Player (2)
- Return of the Jedi (4)
- BlackBerry Curve (1)
- Sony Ericsson (1)
- New Hope (1)

Order Total: 539.97

# Step 1 / Starting Point: Monolith running in VM(s)

Functionality:  
Add Products to Shopping Cart

**Electronic and Movie Depot**

Shop    Cart    Order History    Account

Entertainment ▾  
Electronics ▾

**Movies**

Star Wars	Return of the Jedi
Episode VI: Return of the Jedi	50

Star Wars

Return of the Jedi

Current Shopping Cart

- Empire Strikes Back (1)
- Return of the Jedi (1)

Order Total: 59.98

Return of the Jedi

Drag Image To Cart

+

Star Wars

Episode VI:  
Return of the Jedi

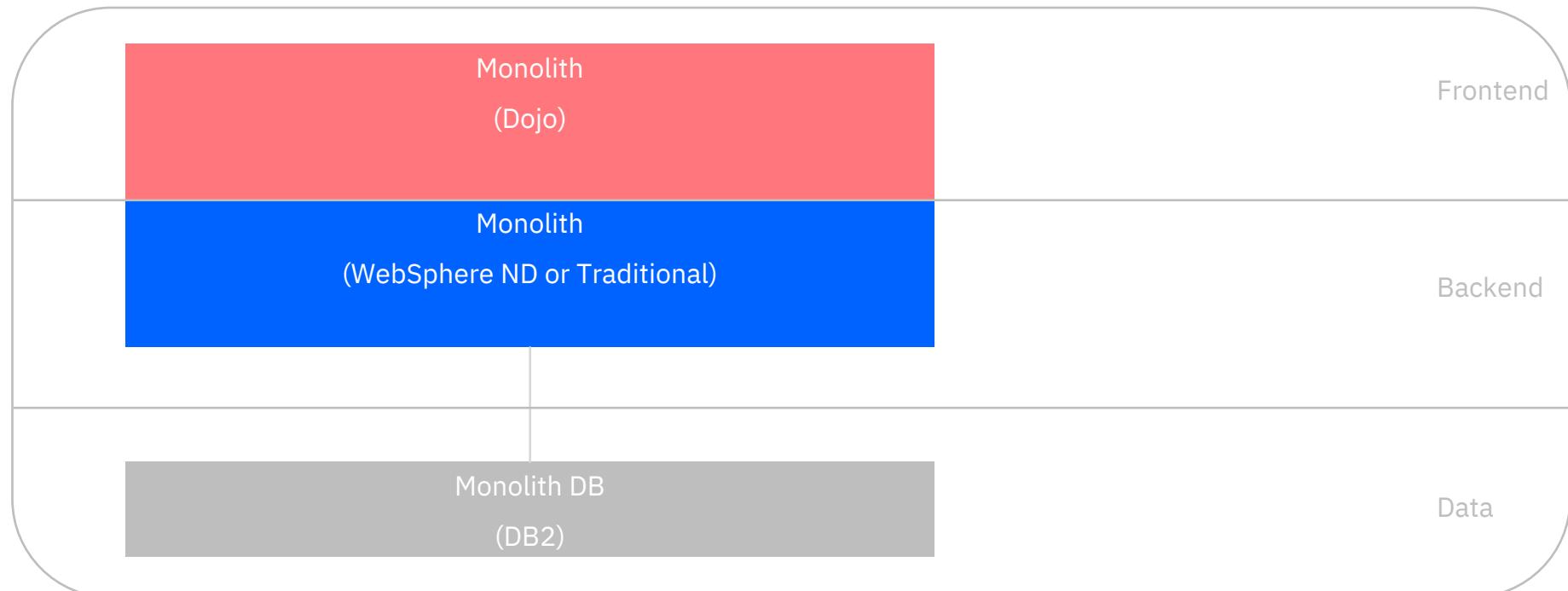
Star Wars

Episode VI:  
Return of the Jedi

# Step 2: Monolith running in Container on WebSphere Traditional

Description and Value	How did we get here?	Used Technologies	Containers
Monolith running in Container  -> Can run on orchestration platforms like OpenShift  -> Consistent management of all applications  -> CI/CD possible	Transformation Advisor generated Dockerfile  Manually created Db2 container	WebSphere Traditional 9	Monolith Db2

# Step 2: Monolith running in Container on WebSphere Traditional



# Step 2: Monolith running in Container on WebSphere Traditional

IBM Cloud Transformation Advisor

Workspace + Home / workspace / collection855

workspace ▾

Collections +/- collection855 +

collection90

collection855

Source environment IBM WebSphere Application Server Profile AppSrv01 Migration target on Cloud Pak for Apps ⓘ Version: 8.5.5.17 WebSphere traditional ⓘ

Upload options ▾

Java applications (2)

Business apps +

0 apps created

Search Filter New Up

Name	Migration Target	Complexity	Issues	Estimated dev cost in days	⋮
CustomerOrderServicesApp.ear	WebSphere traditional	Simple	● ● ● 8	0	⋮
query.ear	WebSphere traditional	Moderate	■ 1 ● 4	1	⋮

# Step 2: Monolith running in Container on WebSphere Traditional

## Migration Files

These files are generated by Transformation Advisor

[server\\_config.py](#) ↓

[install\\_app.py](#) ↓

[Dockerfile](#) ↓

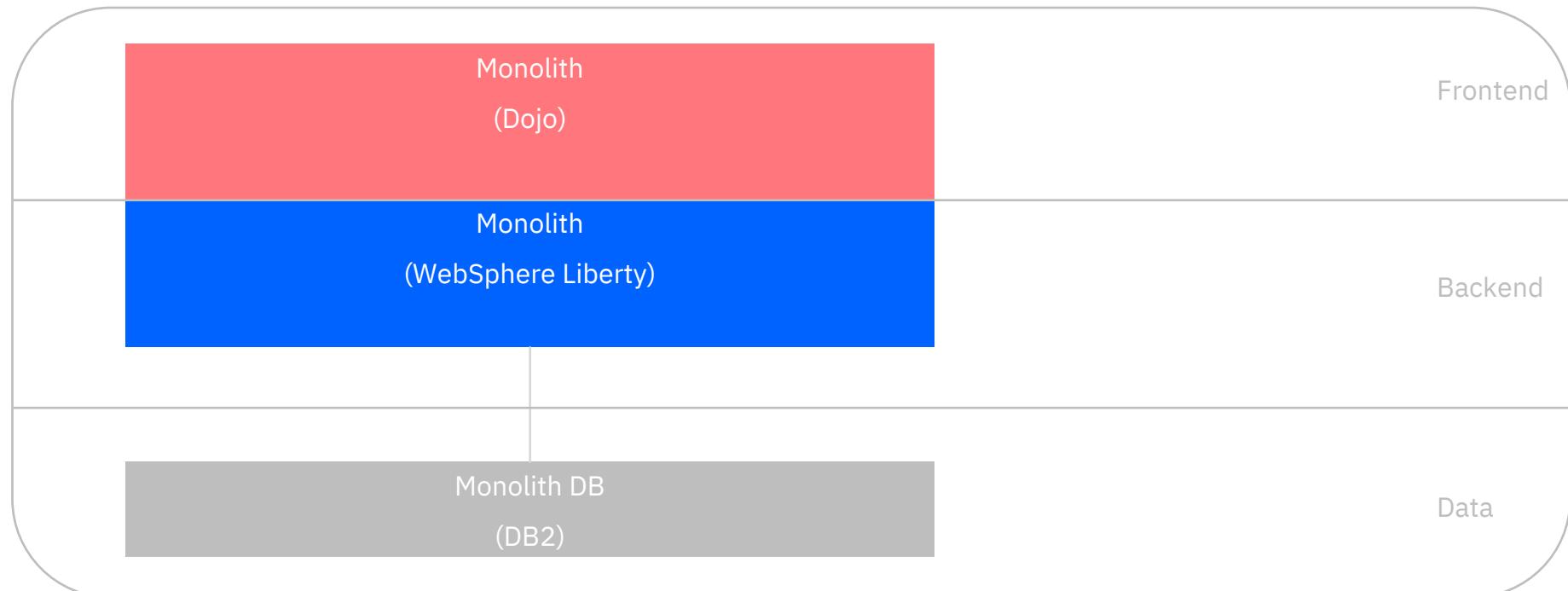
[Pipeline resources](#) ↓

**Question:**  
**Why don't the generated files work?**

# Step 3: Monolith running in Container on WebSphere Liberty

Description and Value	How did we get here?	Used Technologies	Containers
Monolith running on Liberty  -> More efficient runtime for container workloads	Transformation Advisor generated Dockerfile and server.xml  Eclipse Migration Tools for code changes (?)	WebSphere Liberty	Monolith Db2

# Step 3: Monolith running in Container on WebSphere Liberty



OpenShift / Kubernetes

# Step 3: Monolith running in Container on WebSphere Liberty

Connectivity Rules Summary											Jump To Rule	▼
This table summarizes the flagged connectivity rules for each Java archive. Select the links in the column header to view all detailed results for that rule. Select the number links within this table to view the detailed results for that specific Java archive.												
	Databases	Enterprise information systems (EIS)	Java EE security	Java Message Service (JMS)	JavaMail server	Message-Driven Beans (MDB)	Remote EJB lookups	Remote EJB providers	Remote web services	Third-party security	Vendor specific messaging	
+ CustomerOrderServicesApp.ear	1		3									

## Detailed Results by Rule

Expand all | Collapse all

### Severe Rules

WebSphere traditional to Liberty

#### 🚫 Behavior change on lookups for Enterprise JavaBeans (4)

Show rule help | Close results

##### Results

FILE NAME	REFERENCE DETAILS	MATCH CRITERIA	LINE NUMBER
CustomerOrderServicesApp.ear/CustomerOrderServicesTest-0.1.0-SNAPSHOT.war			
WEB-INF/ibm-web-bnd.xml		ejb-ref.binding-name=ejblocal:org.pwte.example.service.ProductSearchService	10
CustomerOrderServicesApp.ear/CustomerOrderServicesWeb-0.1.0-SNAPSHOT.war			
WEB-INF/classes/org/pwte/example/resources/CategoryResource.class	Constructor	ejblocal:org.pwte.example.service.ProductSearchService	28
WEB-INF/classes/org/pwte/example/resources/CustomerOrderResource.class	Constructor	ejblocal:org.pwte.example.service.CustomerOrderServices	53
WEB-INF/classes/org/pwte/example/resources/ProductResource.class	Constructor	ejblocal:org.pwte.example.service.ProductSearchService	38

# Step 3: WebSphere Liberty

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure with files like AbstractCustomer.java and CustomerOrderResource.java.
- Code Editor:** Displays the Java code for CustomerOrderResource.java, specifically focusing on EJB lookups.
- Java Code Review:** A tool window showing inspection results:
  - View as:** Tree
  - Java EE 7 [10 results in 14ms]:**
    - CDI [2 results in 10ms]
      - CDI recognizes implicit bean archives [2 results in 1ms]
    - JAX-RS 2.0 [8 results in 0ms]
      - org.codehaus.jackson packages are not available [8 results in 0ms]
        - AbstractCustomer.java:21 org.codehaus.jackson packages are not available
        - Category.java:16 org.codehaus.jackson packages are not available
        - Category.java:17 org.codehaus.jackson packages are not available
        - LineItem.java:19 org.codehaus.jackson packages are not available
        - Order.java:21 org.codehaus.jackson packages are not available
        - Product.java:16 org.codehaus.jackson packages are not available
        - Product.java:17 org.codehaus.jackson packages are not available
        - CustomerServicesApp.java:23 org.codehaus.jackson packages are not available
    - Java EE 8 [1 result in 19ms]:**
      - JAX-RS 2.1 [1 result in 10ms]
        - The com.ibm.json packages are not available in the jaxrs-2.1 feature [1]
        - CustomerOrderResource.java:41 The com.ibm.json packages are not available
    - Liberty migration [3 results in 32ms]:**
      - WebSphere traditional to Liberty [3 results in 32ms]
        - Behavior change on lookups for Enterprise JavaBeans [3 results in 0ms]
          - CategoryResource.java:29 Behavior change on lookups for Enterprise JavaBeans
          - CustomerOrderResource.java:53 Behavior change on lookups for Enterprise JavaBeans
          - ProductResource.java:38 Behavior change on lookups for Enterprise JavaBeans
  - Annotations:** A panel on the right showing annotations for the code editor.
  - Help:** A panel on the right showing help resources.

# Step 3: WebSphere Liberty

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows the project structure with files like AbstractCustomer.java, CustomerOrderResource.java, and various CustomerOrderServices\*.java files.
- Editor:** Displays the Java code for CustomerOrderResource.java, which includes imports for org.pwte.\* and com.ibm.json.\* packages, and a constructor that looks up a CustomerOrderServices bean.
- Bottom Bar:** Contains tabs for XML File Review, JSP Code Review, File Review, and Java Code Review, with Java Code Review selected.
- Bottom Panel:** Shows a static analysis report titled "was-to-lib (21/01/2014:11:26) [24]" with a tree view of findings categorized by Java EE version and feature.
- Right Panel:** A tooltip provides information about the absence of com.ibm.json packages in the jaxrs-2.1 feature.

**Java Code Review Findings:**

- Java EE 7:** CDI [2 results in 10ms], JAX-RS 2.0 [8 results in 0ms].
  - org.codehaus.jackson packages are not available [8 results in 0ms]: AbstractCustomer.java:21, Category.java:16, Category.java:17, LineItem.java:19, Order.java:21, Product.java:16, Product.java:17, CustomerServicesApp.java:23.
- Java EE 8:** JAX-RS 2.1 [1 result in 19ms].
  - The com.ibm.json packages are not available in the jaxrs-2.1 feature [1 result in 10ms]: CustomerOrderResource.java:41.
- Liberty migration:** WebSphere traditional to Liberty [3 results in 32ms].
  - Behavior change on lookups for Enterprise JavaBeans [3 results in 0ms]: CategoryResource.java:29, CustomerOrderResource.java:53, ProductResource.java:38.

**Right Panel Tooltip:**

The com.ibm.json packages are not available in the jaxrs-2.1 feature

The json4j com.ibm.json packages were previously included in the jaxrs-1.1 and jaxrs-2.0 Liberty features. However, the packages are no longer included in the jaxrs-2.1 feature.

Note: This rule is flagged once per Eclipse project or Java archive.

**Option 1 (Recommended): Refactor your application to use JSON-P**

It is recommended to switch your application from using JSON4J to using the JSON Processing (JSON-P) APIs provided by the Java EE specification. Once you have made the necessary application changes, no further changes to the Liberty server.xml configuration is required since the jsonp-1.1 feature is enabled by the jaxrs-2.1 feature.

For more information on JSON-P, see [Java API for JSON Processing](#).

**Option 2: Add the json-1.0 feature in server.xml**

If you want to continue using the json4j com.ibm.json packages, add the json-1.0 feature to your Liberty server.xml configuration file.

For more information on the json-1.0 feature, see [JavaScript Object Notation for Java](#).

# Step 3: WebSphere Liberty

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows several projects including "CustomerOrderServices", "CustomerOrderServicesA", "CustomerOrderServicesW", and "StaticAnalysisProject".
- Code Editor:** Displays the file `AbstractCustomer.java` with the following code snippet:

```
11 import javax.persistence.Entity;
12 import javax.persistence.FetchType;
13 import javax.persistence.Id;
14 import javax.persistence.Inheritance;
15 import javax.persistence.InheritanceType;
16 import javax.persistence.JoinColumn;
17 import javax.persistence.OneToMany;
18 import javax.persistence.OneToOne;
19 import javax.persistence.Table;
20
21 import org.codehaus.jackson.annotate.JsonIgnore;
22
23 @Entity
24 @Table(name = "CUSTOMER")
25 @Inheritance(strategy = InheritanceType.SINGLE_TABLE)
26 @DiscriminatorColumn(name = "TYPE", discriminatorType = DiscriminatorType.STRING)
27 public abstract class AbstractCustomer implements Serializable {
28
29     private static final long serialVersionUID = -4988539679853665972L;
30
31     public AbstractCustomer() {
32
33     }
34 }
```

- Java Code Review:** A detailed report of migration issues:

  - Java EE 7 [10 results in 14ms]:**
    - CDI [2 results in 10ms]
      - CDI recognizes implicit bean archives [2 results in 1ms]
    - JAX-RS 2.0 [8 results in 0ms]
      - org.codehaus.jackson packages are not available [8 results in 0ms]
        - AbstractCustomer.java:21 org.codehaus.jackson packages are not available
        - Category.java:16 org.codehaus.jackson packages are not available
        - Category.java:17 org.codehaus.jackson packages are not available
        - LineItem.java:19 org.codehaus.jackson packages are not available
        - Order.java:21 org.codehaus.jackson packages are not available
        - Product.java:16 org.codehaus.jackson packages are not available
        - Product.java:17 org.codehaus.jackson packages are not available
        - CustomerServicesApp.java:23 org.codehaus.jackson packages are not available
    - Java EE 8 [1 result in 19ms]:**
      - JAX-RS 2.1 [1 result in 10ms]
        - The com.ibm.json packages are not available in the jaxrs-2.1 feature [1 result in 10ms]
          - CustomerOrderResource.java:41 The com.ibm.json packages are not available
    - Liberty migration [3 results in 32ms]:**
      - WebSphere traditional to Liberty [3 results in 32ms]
        - Behavior change on lookups for Enterprise JavaBeans [3 results in 0ms]
          - CategoryResource.java:29 Behavior change on lookups for Enterprise JavaBeans
          - CustomerOrderResource.java:53 Behavior change on lookups for Enterprise JavaBeans
          - ProductResource.java:38 Behavior change on lookups for Enterprise JavaBeans
  - Help:** Shows a message: "org.codehaus.jackson packages are not available".
  - Annotations:** Shows a warning about JAX-RS 2.1 annotations.

# Step 3: Monolith running in Container on WebSphere Liberty

## Migration Files

These files are generated by Transformation Advisor

server.xml [↓](#)

pom.xml [↓](#)

Operator resources [↓](#)

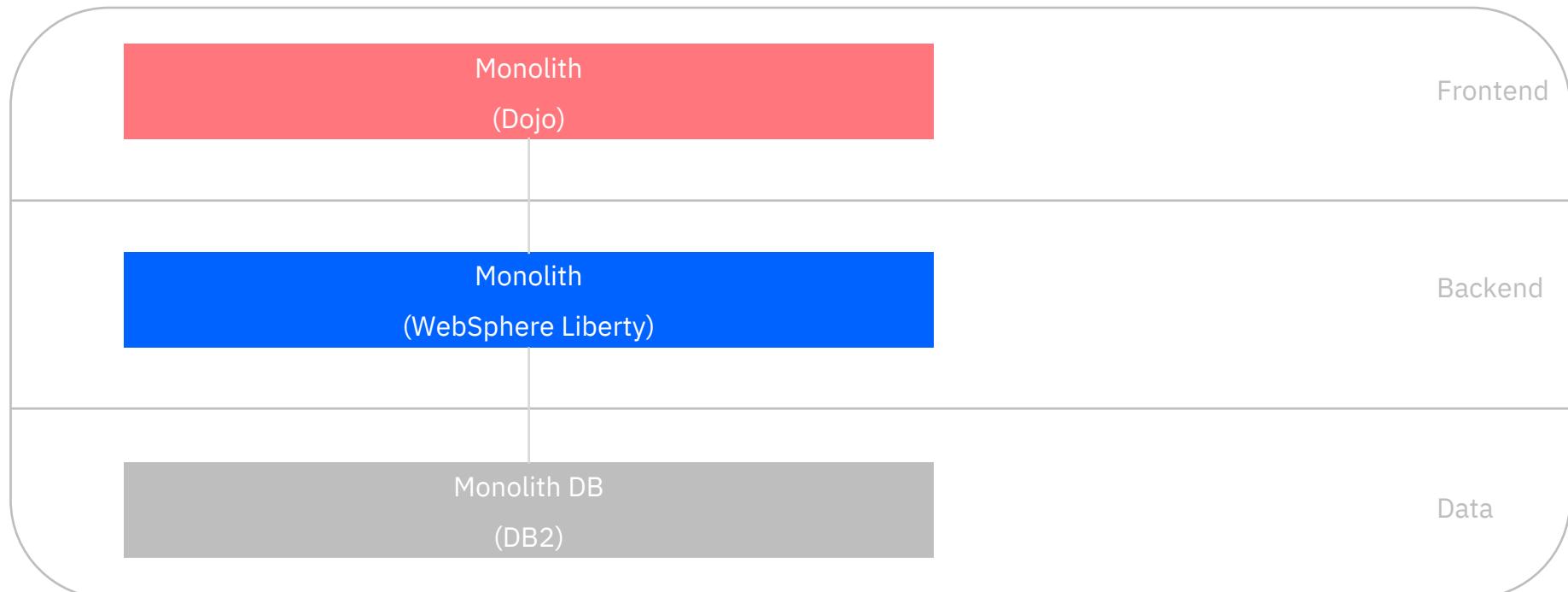
Dockerfile [↓](#)

**Question:**  
**Why don't the generated files work?**

# Step 4: Separated Frontend

Description and Value	How did we get here?	Used Technologies	Containers
Separate container for Dojo frontend  -> Easier extensions and potential replacement of frontend code	Manually	WebSphere Liberty Dojo	Backend monolith Frontend monolith Db2

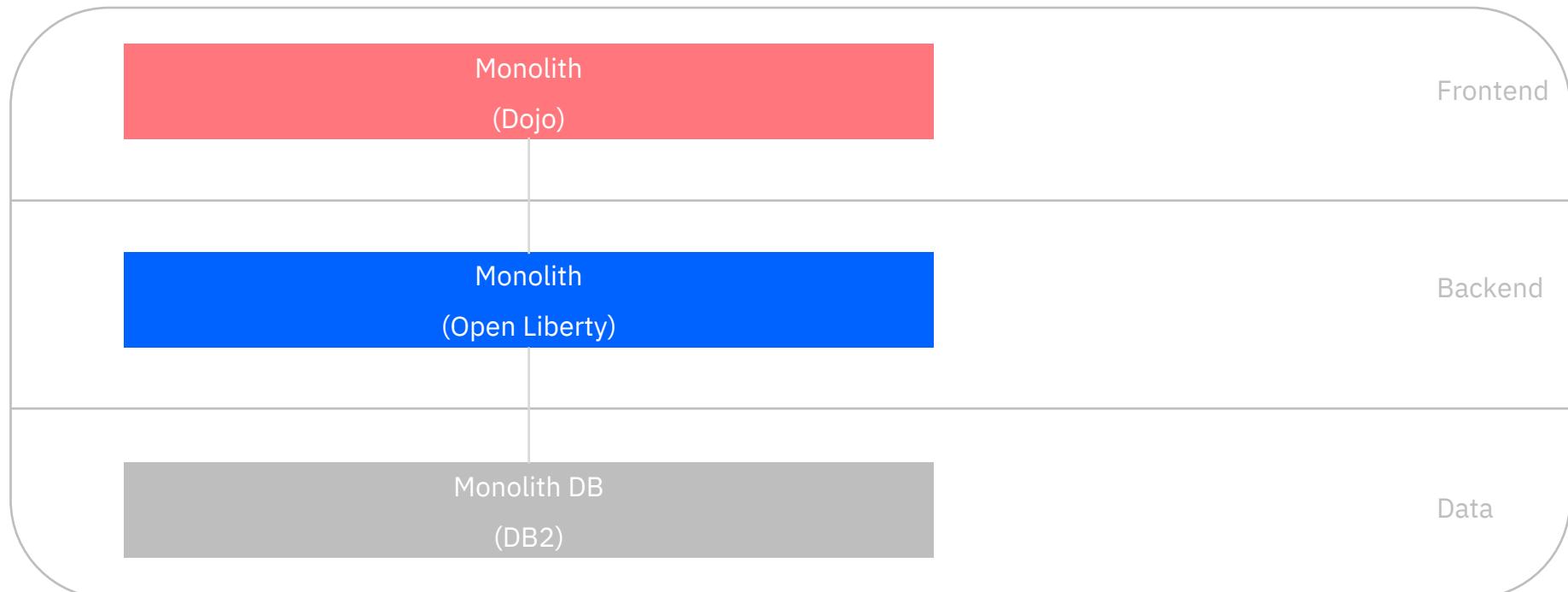
# Step 4: Separated Frontend



# Step 5: Open Liberty Monolith

Description and Value	How did we get here?	Used Technologies	Containers
Usage of Open Liberty -> License cost savings	Manually (primarily changing image only)	Open Liberty MicroProfile	Backend monolith Frontend monolith Db2
-> Leverage latest MicroProfile and Open Liberty versions early and leverage open source communities			

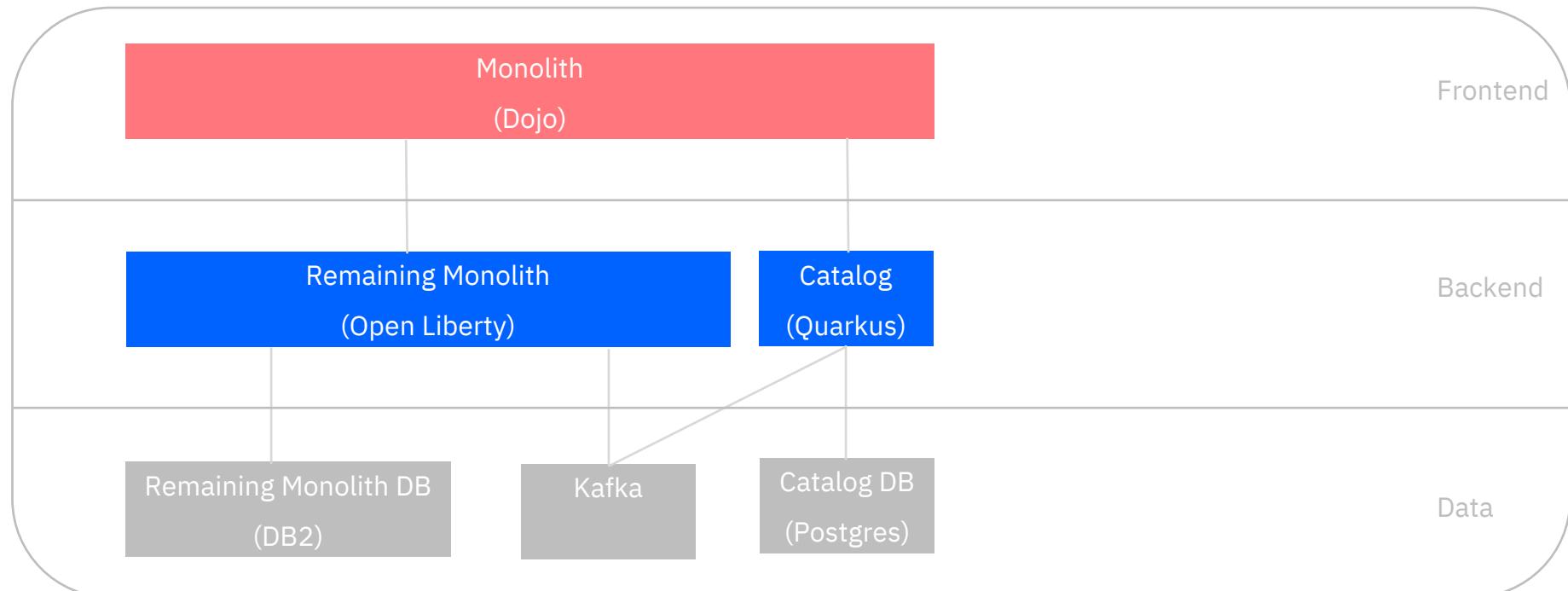
# Step 5: Open Liberty Monolith



# Step 6: Strangled Catalog Service and Open Liberty Monolith

Description and Value	How did we get here?	Used Technologies	Containers
Strangled catalog service <ul style="list-style-type: none"><li>- Quarkus</li><li>- Kafka for CQRS</li><li>- Panache</li></ul>	Inspired by Mono2Micro	Quarkus OpenJ9	Backend monolith (Liberty)
Open Liberty monolith <ul style="list-style-type: none"><li>- Modern project structure</li><li>- No EJBs, but CDI</li><li>- Kafka for CQRS</li></ul>	Manually	Kafka Open Liberty MicroProfile	Catalog service Frontend monolith Db2 Kafka Postgres
-> Separate scalability			

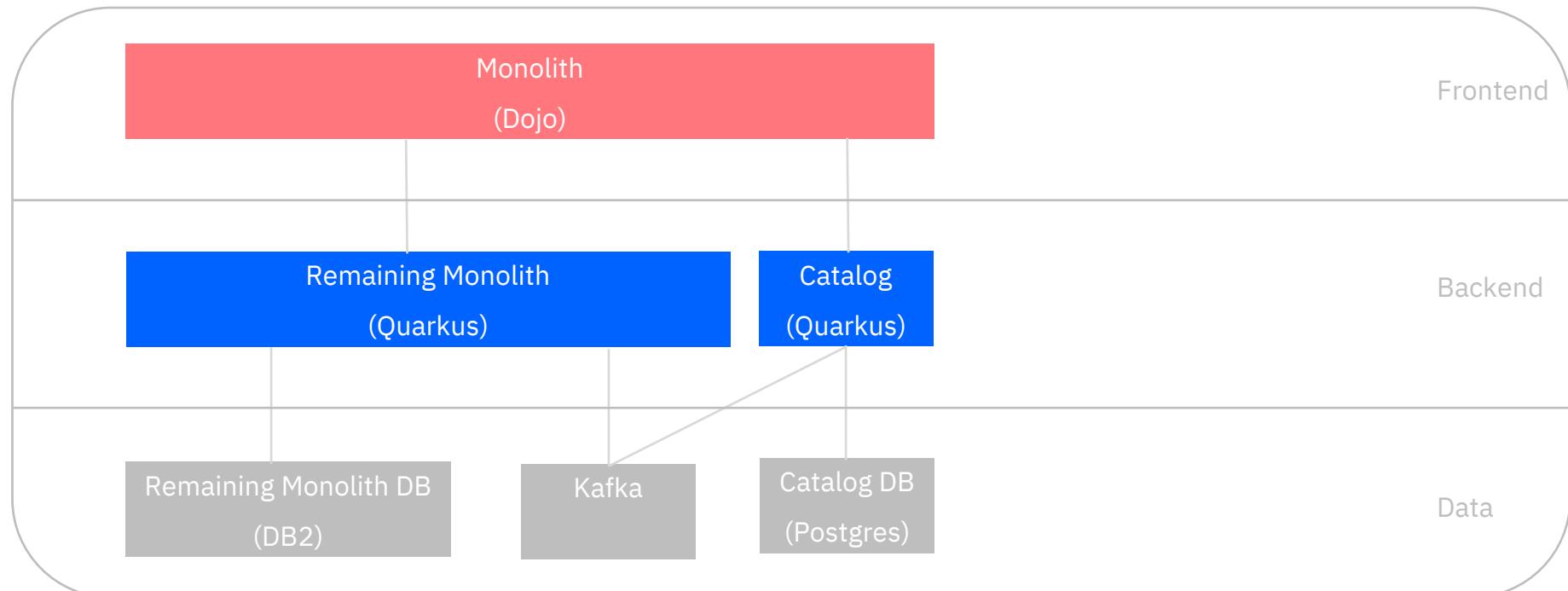
# Step 6: Strangled Catalog Service and Open Liberty Monolith



# Step 7: Strangled Catalog Service and Quarkus Monolith

Description and Value	How did we get here?	Used Technologies	Containers
Strangled catalog service <ul style="list-style-type: none"><li>- Reactive endpoints</li><li>- Reactive Postgres access</li></ul> <p>-&gt; More efficient than synchronous code</p>	Manually	Quarkus OpenJ9 Kafka MicroProfile	Backend monolith (Quarkus) Catalog service Frontend monolith Db2 Kafka Postgres
Quarkus monolith <ul style="list-style-type: none"><li>- Compiled to native binary</li></ul> <p>-&gt; Faster startup time and less memory usage</p>			

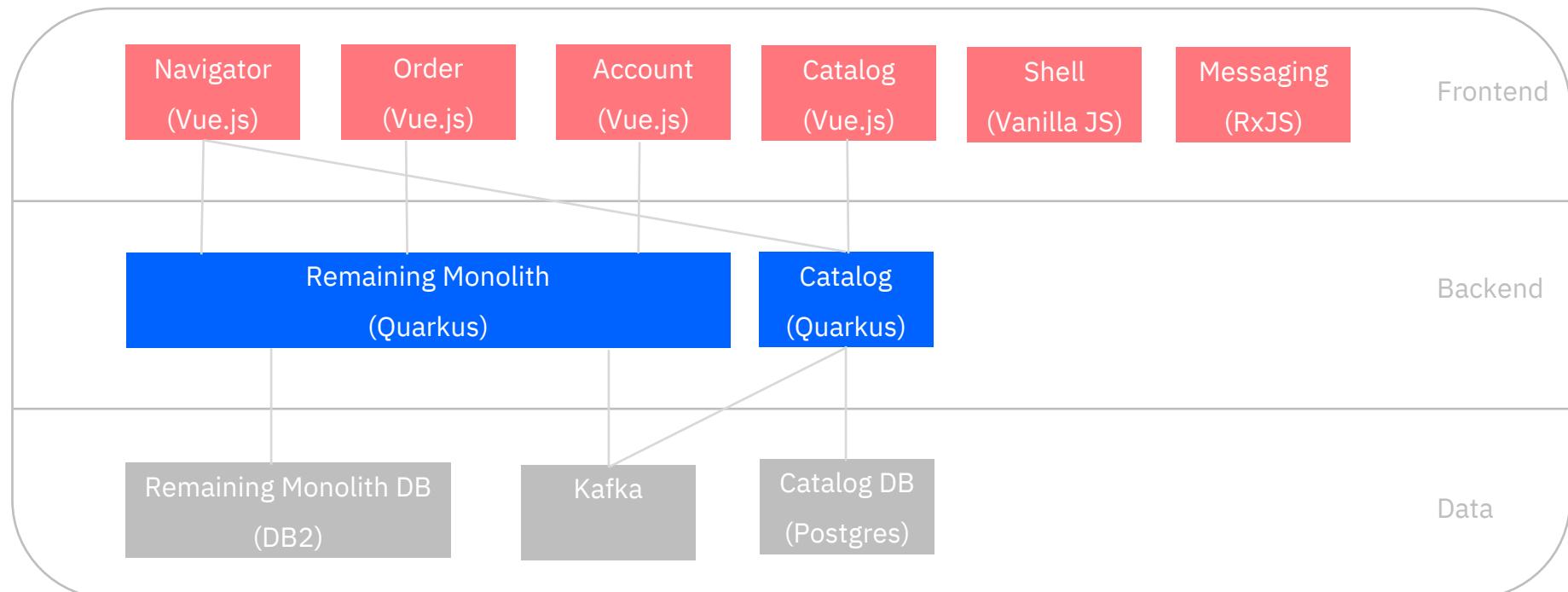
# Step 7: Strangled Catalog Service and Quarkus Monolith



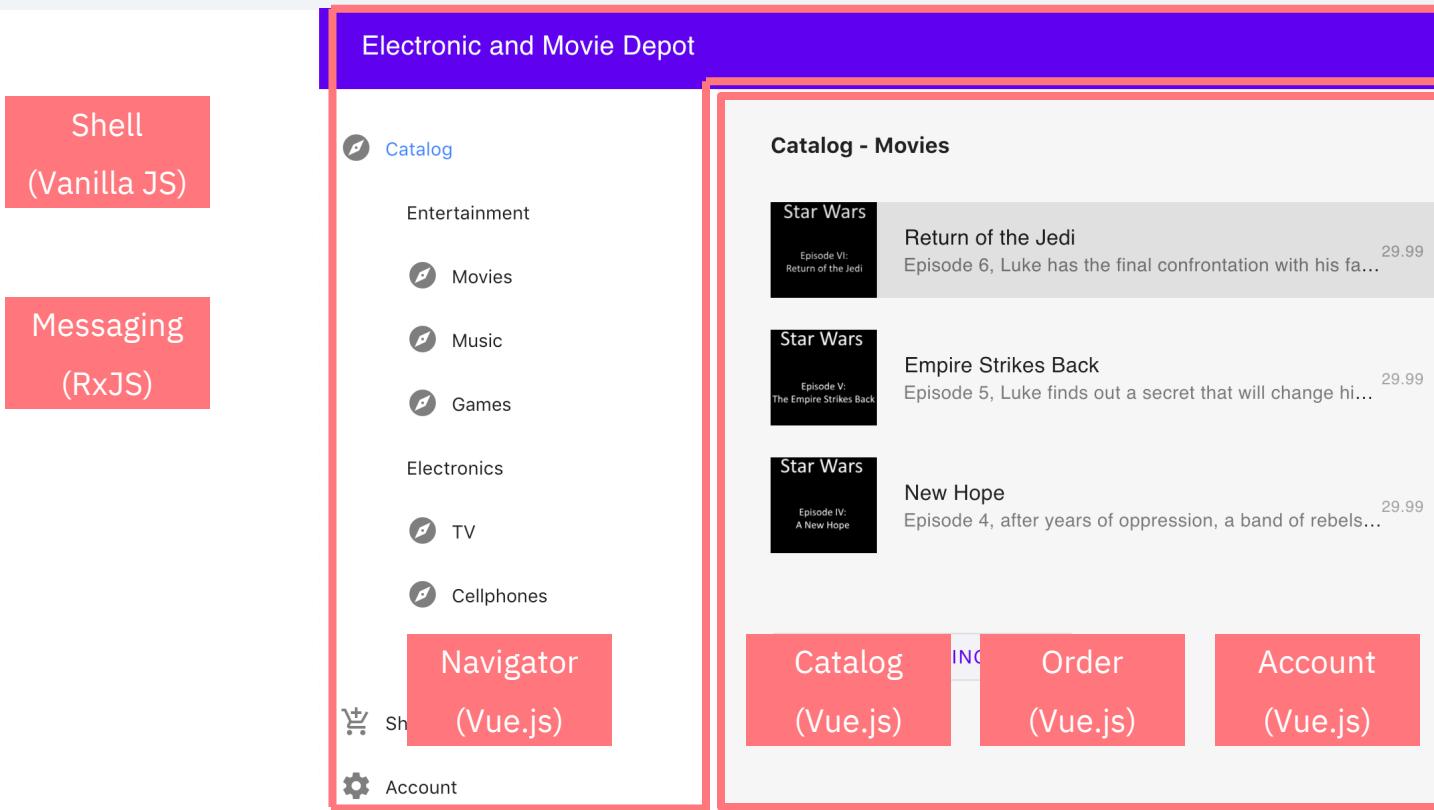
# Step 8: Quarkus Backend and Micro Frontends

Description and Value	How did we get here?	Used Technologies	Containers
Break down frontend monolith in micro frontends  -> Independent updates to separate pages	Manually	single-spa Vue.js RxJS	Backend monolith (Quarkus) Catalog service Frontend monolith Db2 Kafka Postgres Navigator, catalog, account, order, shell, messaging

# Step 8: Quarkus Backend and Micro Frontends



# Micro Frontends via single-spa



# Micro Frontends via single-spa

Navigator  
(Vue.js)

Catalog  
(Vue.js)

Order  
(Vue.js)

Account  
(Vue.js)

Shell  
(Vanilla JS)

```
import { registerApplication, start } from "single-spa";

registerApplication({
  name: "@vue-app-mod/navigator",
  app: () => System.import("@vue-app-mod/navigator"),
  activeWhen: ["/"],
});

registerApplication({
  name: "@vue-app-mod/order",
  app: () => System.import("@vue-app-mod/order"),
  activeWhen: ["/", "/catalog", "/order"],
});

registerApplication({
  name: "@vue-app-mod/account",
  app: () => System.import("@vue-app-mod/account"),
  activeWhen: "/account",
});

registerApplication({
  name: "@vue-app-mod/catalog",
  app: () => System.import("@vue-app-mod/catalog"),
  activeWhen: ["/", "/catalog"],
});

start();
```

Messaging  
(RxJS)

```
export default {
  ...
  getObservable(microFrontendName) {
    switch (microFrontendName) {
      case MICRO_FRONTEND_NAVIGATOR:
        if (!observableForNavigator) {
          observableForNavigator = new Subject()
        }
        return observableForNavigator
    }
  },
  send(message) {
    let topic = message.topic
    let commandId = message.commandId

    switch (topic) {
      case TOPIC_COMMAND_ADD_ITEM:
        if (observableForOrder) {
          observableForOrder.next(message)
        }
        break
    }
  }
},
```

# Micro Frontends via single-spa

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta name="importmap-type" content="systemjs-importmap" />
  <script type="systemjs-importmap">
    {
      "imports": {
        "single-spa": "https://cdn.jsdelivr.net/npm/single-spa@5.5.1/lib/system/single-spa.min.js",
        "vue": "https://cdn.jsdelivr.net/npm/vue@2.6.11/dist/vue.min.js",
        "vue-router": "https://cdn.jsdelivr.net/npm/vue-router@3.1.6/dist/vue-router.min.js",
        "rxjs": "http://cdnjs.cloudflare.com/ajax/libs/rxjs/6.6.3/rxjs.umd.min.js",

        "@vue-app-mod/shell": "http://localhost:8080/vue-app-mod-shell.js",
        "@vue-app-mod/messaging": "http://localhost:9001/vue-app-mod-messaging.js",
        "@vue-app-mod/navigator": "http://localhost:8501/js/app.js",
        "@vue-app-mod/order": "http://localhost:8504/js/app.js",
        "@vue-app-mod/catalog": "http://localhost:8503/js/app.js",
        "@vue-app-mod/account": "http://localhost:8502/js/app.js"
      }
    }
  </script>
  <script src="https://cdn.jsdelivr.net/npm/import-map-overrides/dist/import-map-overrides.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/systemjs/dist/system.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/systemjs/dist/extras/amd.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/systemjs/dist/extras/named-exports.js"></script>
  <link href="https://unpkg.com/material-components-web@latest/dist/material-components-web.min.css" rel="stylesheet">
  <script src="https://unpkg.com/material-components-web@latest/dist/material-components-web.min.js"></script>
  <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
</head>
<body>
  <div style="display: flex;flex-direction: row;">
    <div id="navigator" style="order: 1;flex-grow: 2;max-width: 300px; width: 300px"></div>
    <div id="catalog" style="order: 2; flex-grow: 10;left: 320px;position: fixed;"></div>
    <div id="order" style="order: 2; flex-grow: 10;left: 320px;position: fixed;"></div>
    <div id="account" style="order: 2; flex-grow: 10;left: 320px;position: fixed;"></div>
  </div>
  <script>
    System.import('@vue-app-mod/shell');
  </script>
  <import-map-overrides-full show-when-local-storage="devtools" dev-libs></import-map-overrides-full>
</body>
</html>
```

**Catalog**

- Entertainment
- Movies
- Music
- Games
- Electronics
- TV
- Cellphones

## Catalog - Movies

Image	Title	Description	Price
Star Wars Episode VI: Return of the Jedi	Return of the Jedi	Episode 6, Luke has the final confrontation with his fa...	29.99
Star Wars Episode V: The Empire Strikes Back	Empire Strikes Back	Episode 5, Luke finds out a secret that will change hi...	29.99

**@vue-app-mod/navigator**

Default URL: <http://localhost:8501/js/app.js>

Override URL:  [\(X\)](#)

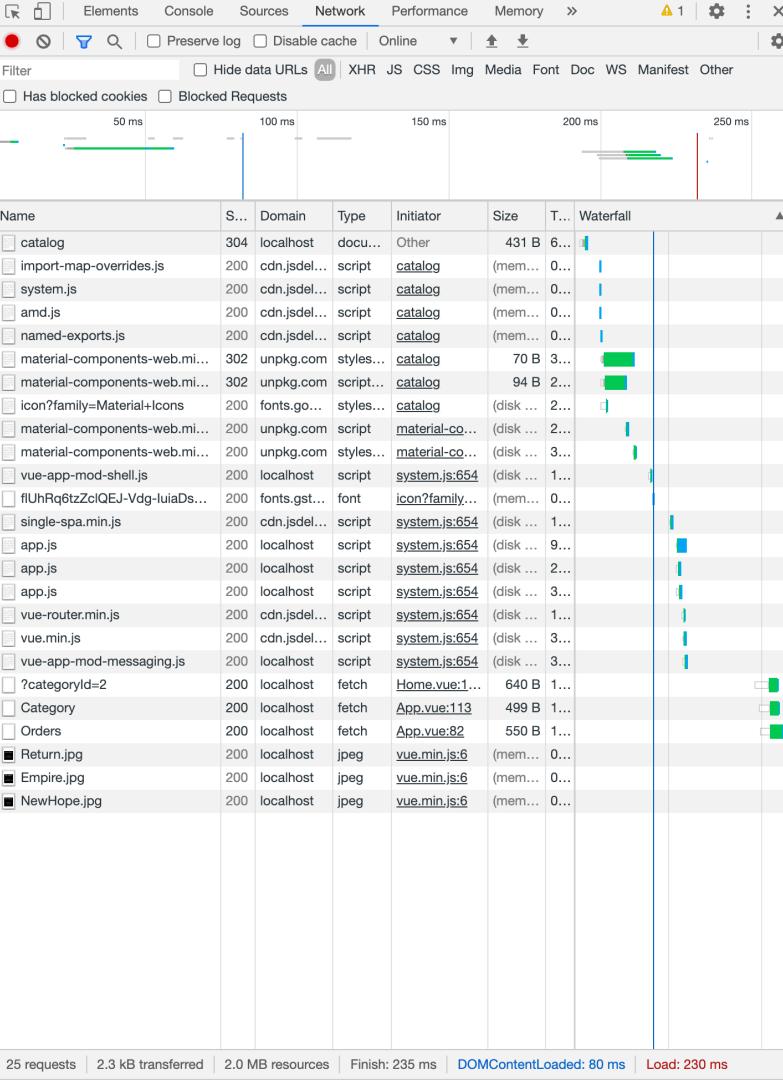
[Cancel](#) [Reset to default](#)

[ADD TO SHOPPING CART](#)

override. See documentation for more info.

Search modules [Add new module](#) [Add import map](#) [Reset all overrides](#)

Module Status	Module Name	Domain	Filename
● Inline Override	single-spa	cdn.jsdelivr.net	single-spa.dev.js
● Inline Override	vue	cdn.jsdelivr.net	vue.js
● Inline Override	vue-router	cdn.jsdelivr.net	vue-router.js
● Default	rxjs	cdnjs.cloudflare.com	rxjs.umd.min.js
● Default	@vue-app-mod/shell	localhost:8080	vue-app-mod-shell.js
● Default	@vue-app-mod/messaging	localhost:9001	vue-app-mod-messaging.js
● Default	@vue-app-mod/navigator	localhost:8501	app.js



# Micro Frontends – Messaging via RxJS for loosely Coupling Command Pattern – Example: Add Item to Cart

Catalog -> Order

```
messaging - messaging.js - send invoked - topic: TOPIC_COMMAND_ADD_ITEM commandId: messaging.js:49  
1611049822817
```

```
▶ {__ob__: we} messaging.js:50
```

```
order - App.vue - created - message: App.vue:20
```

```
Topic: TOPIC_COMMAND_ADD_ITEM App.vue:21
```

```
CommandId: 1611049822817 App.vue:22
```

```
ProductId: 1 App.vue:23
```

```
Product has been added to order App.vue:47
```

```
Products in shopping cart: 1 App.vue:54
```

```
Line items in shopping cart: 2 Order -> Navigator App.vue:55
```

```
messaging - messaging.js - send invoked - topic: messaging.js:49  
TOPIC_EVENT_AMOUNT_LINE_ITEMS_CHANGED commandId: undefined
```

```
▶ {topic: "TOPIC_EVENT_AMOUNT_LINE_ITEMS_CHANGED", payload: {...}} messaging.js:50
```

```
navigator - App.vue - amountLineItems: 2 App.vue:74
```

```
messaging - messaging.js - send invoked - topic: TOPIC_COMMAND_RESPONSE_ADD_ITEM messaging.js:49  
commandId: undefined
```

```
▶ {topic: "TOPIC_COMMAND_RESPONSE_ADD_ITEM", payload: {...}} messaging.js:50
```

```
catalog - Home.vue - message: Home.vue:69
```

Order -> Catalog

# Appendix

# lorem

## lorem

“lorem”

lorem

@nheidloff

# lorem

lorem

lorem

lorem

lorem

