DevNation

Transparent web platform decoupling with Multiplying Architecture

Eder Ignatowicz

@ederign

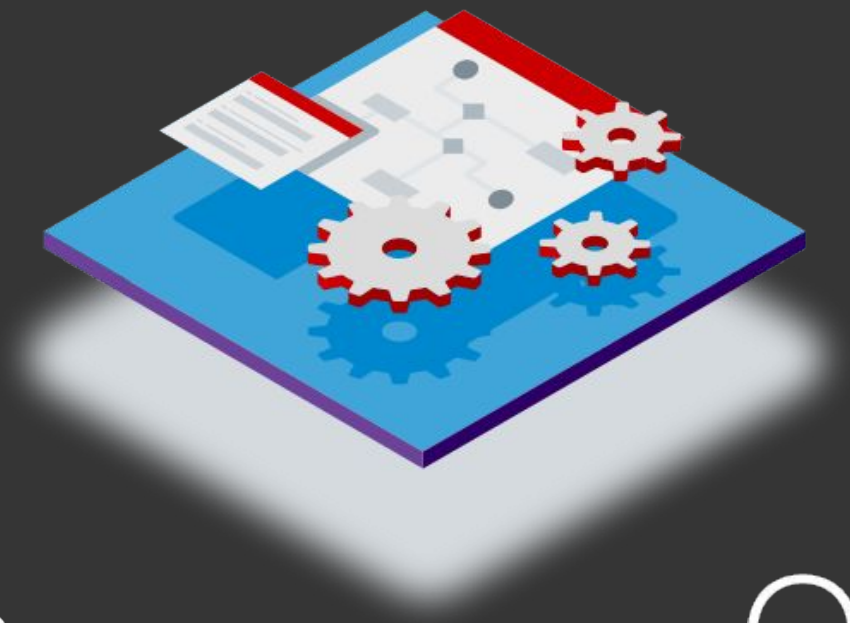
Guilherme Caponetto

@caponetto

Red Hat Developer

KIE

jBPM



Drools

Kogito

OptaPlanner

Red Hat

Serverless Logic Web Tools

Overview

Recent Models

Sample Catalog

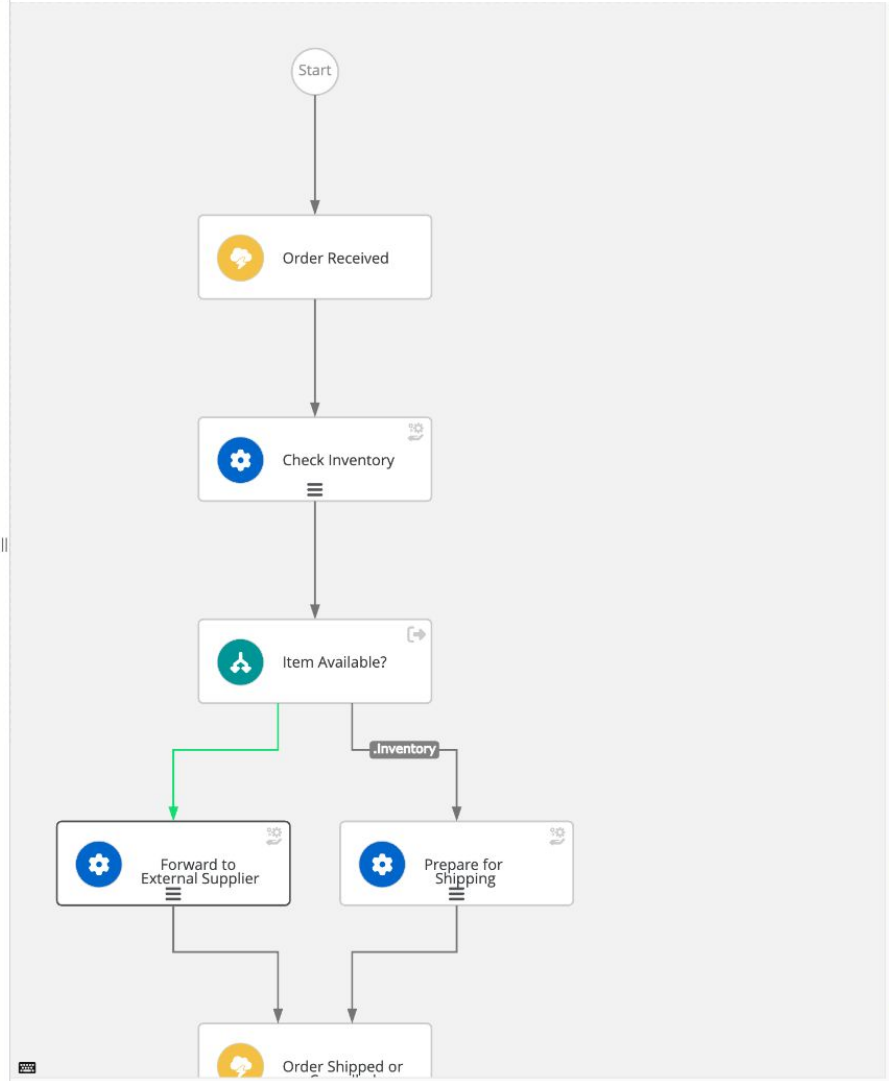
Documentation

Ephemeral order-swf

Serverless Workflow **order**

+ New file | Try on OpenShift | Share

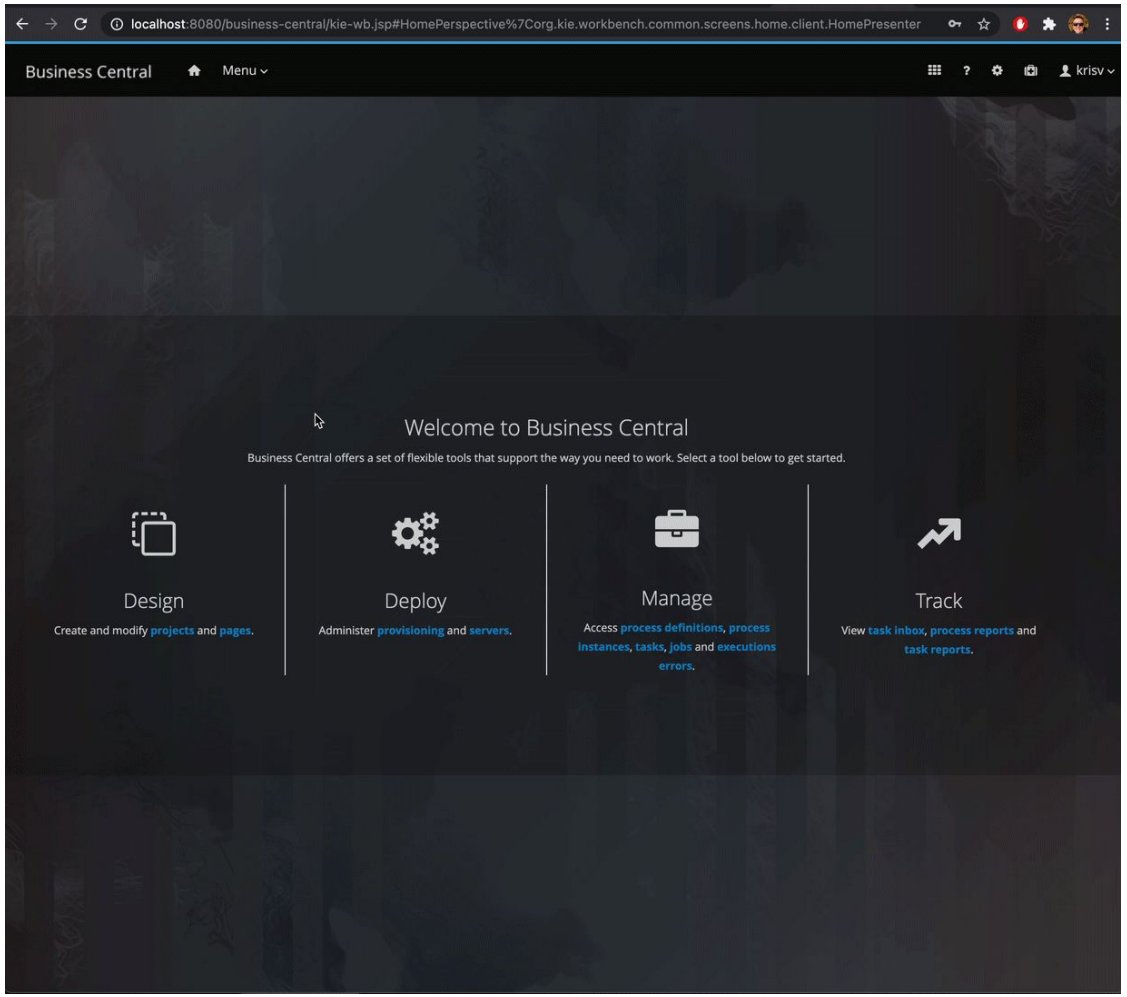
```
{
  "id": "order",
  "version": "1.0",
  "specVersion": "0.8",
  "name": "Order Workflow",
  "description": "Order Workflow Sample",
  "start": "Order Received",
  + Add function...
  "functions": [
    {
      "name": "printMessage",
      "type": "custom",
      "operation": "sysout"
    },
    {
      "name": "sendOrder",
      "operation": "specs/supplier.yaml#sendOrder",
      "type": "rest"
    },
    {
      "name": "cancelOrder",
      "operation": "specs/supplier.yaml#cancelOrder",
      "type": "rest"
    }
  ],
  + Add event...
  "events": [
    {
      "name": "orderEvent",
      "kind": "consumed",
      "type": "OrderEventType",
      "source": "Client",
      "correlation": [
        {
          "contextAttributeName": "orderid"
        }
      ]
    },
    {
      "name": "shippingEvent",
      "kind": "consumed",
      "type": "ShippingEventType",
      "source": "Shipper",
      "correlation": [
        {
          "contextAttributeName": "orderid"
        }
      ]
    },
    {
      "name": "cancelEvent",
      "kind": "consumed",
      "type": "CancelEventType",
```



Problems 0

https://start.kubesmarts.org/#/recent-models

Red Hat

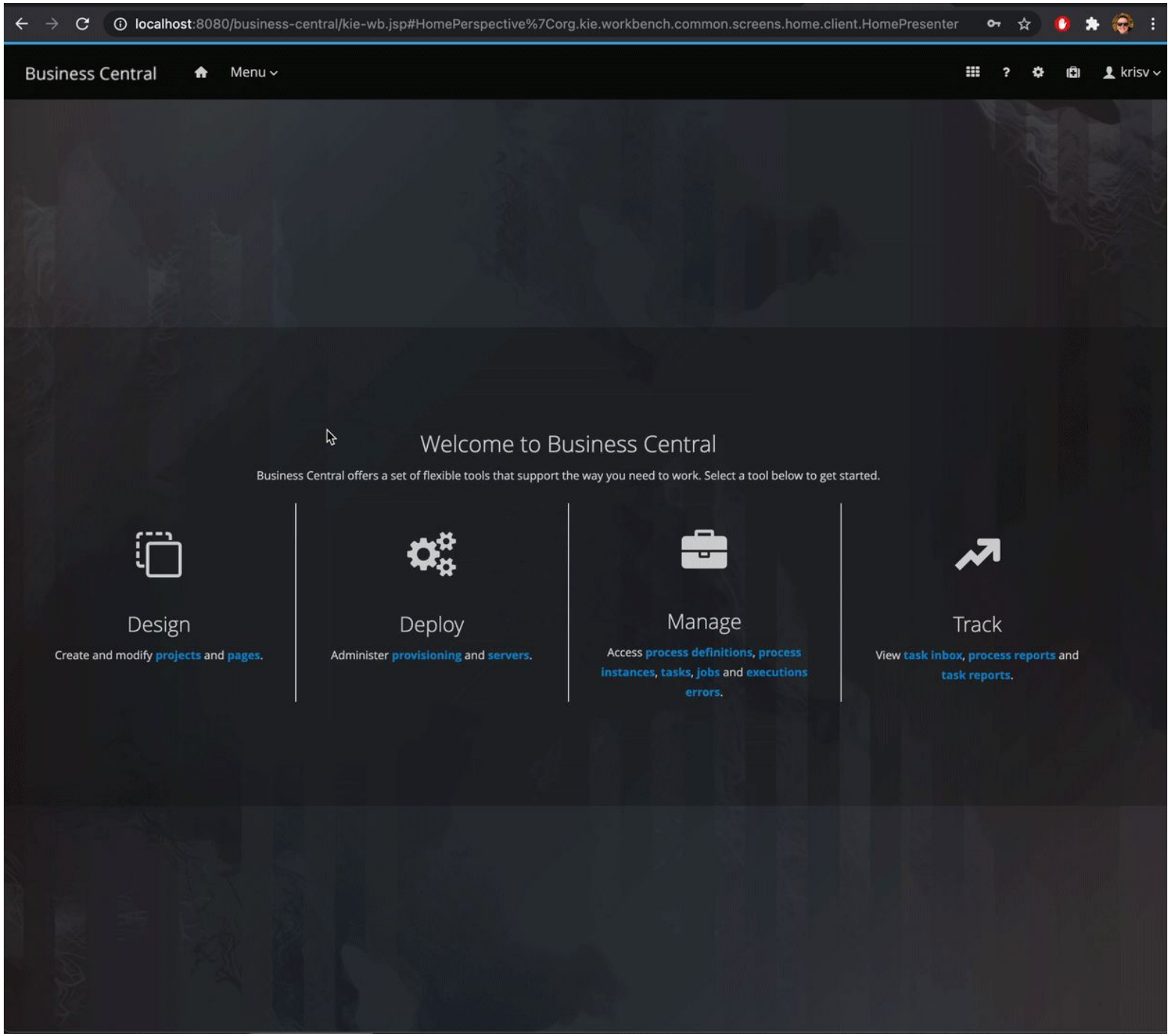


THE DEVELOPER'S CONFERENCE

QCon SÃO PAULO

JavaOne™

Red Hat


localhost:8080/business-central/kie-wb.jsp#HomePerspective%7Corg.kie.workbench.common.screens.home.client.HomePresenter
Business Central
Menu
krisv
Welcome to Business Central
Business Central offers a set of flexible tools that support the way you need to work. Select a tool below to get started.
Design
Create and modify projects and pages.
Deploy
Administer provisioning and servers.
Manage
Access process definitions, process instances, tasks, jobs and executions errors.
Track
View task inbox, process reports and task reports.

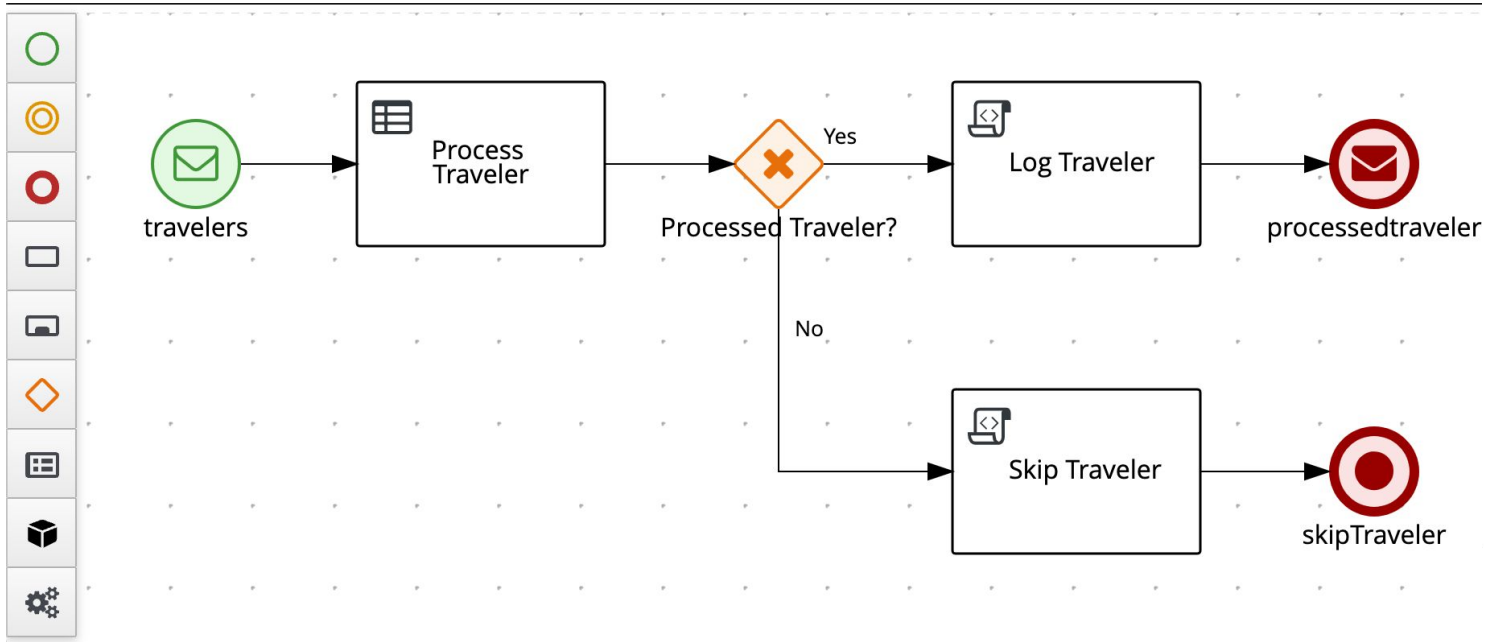
Red Hat

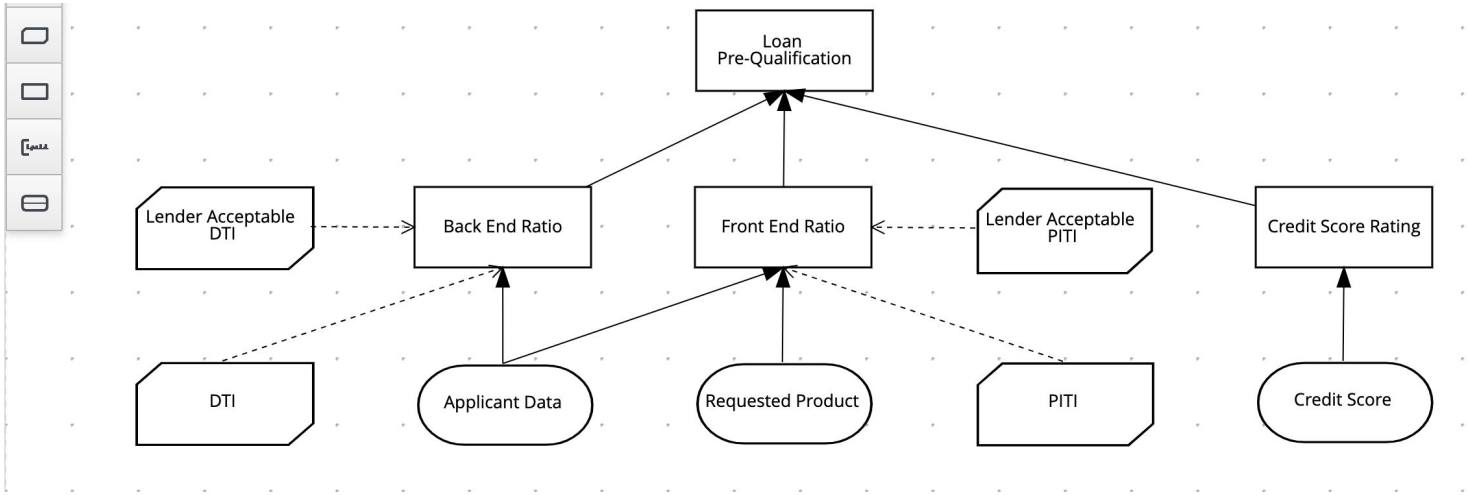
Red Hat

Kogito ergo automate

CLOUD-NATIVE BUSINESS AUTOMATION FOR BUILDING INTELLIGENT APPLICATIONS, BACKED BY BATTLE-TESTED CAPABILITIES.

Red Hat


travelers
Process Traveler
Processed Traveler?
Yes
No
Log Traveler
processedtraveler
Skip Traveler
skipTraveler


Loan Pre-Qualification
Lender Acceptable DTI
Back End Ratio
Front End Ratio
Lender Acceptable PITI
Credit Score Rating
DTI
Applicant Data
Requested Product
PITI
Credit Score

How to adapt a 10 years old legacy to modern web development?

Red Hat

From a handful of engineers to 6 different fullstack teams working independently on different fronts on this new initiative?

Red Hat

How to breakup my frontend monolith into many smaller manageable pieces?

Red Hat

Micro frontends

Red Hat

"An architectural style where independently deliverable frontend applications are composed into a greater whole"

–

Cam Jackson

https://martinfowler.com/articles/micro-frontends.html

Red Hat

Benefits

Incremental upgrades

Simple, decoupled codebases

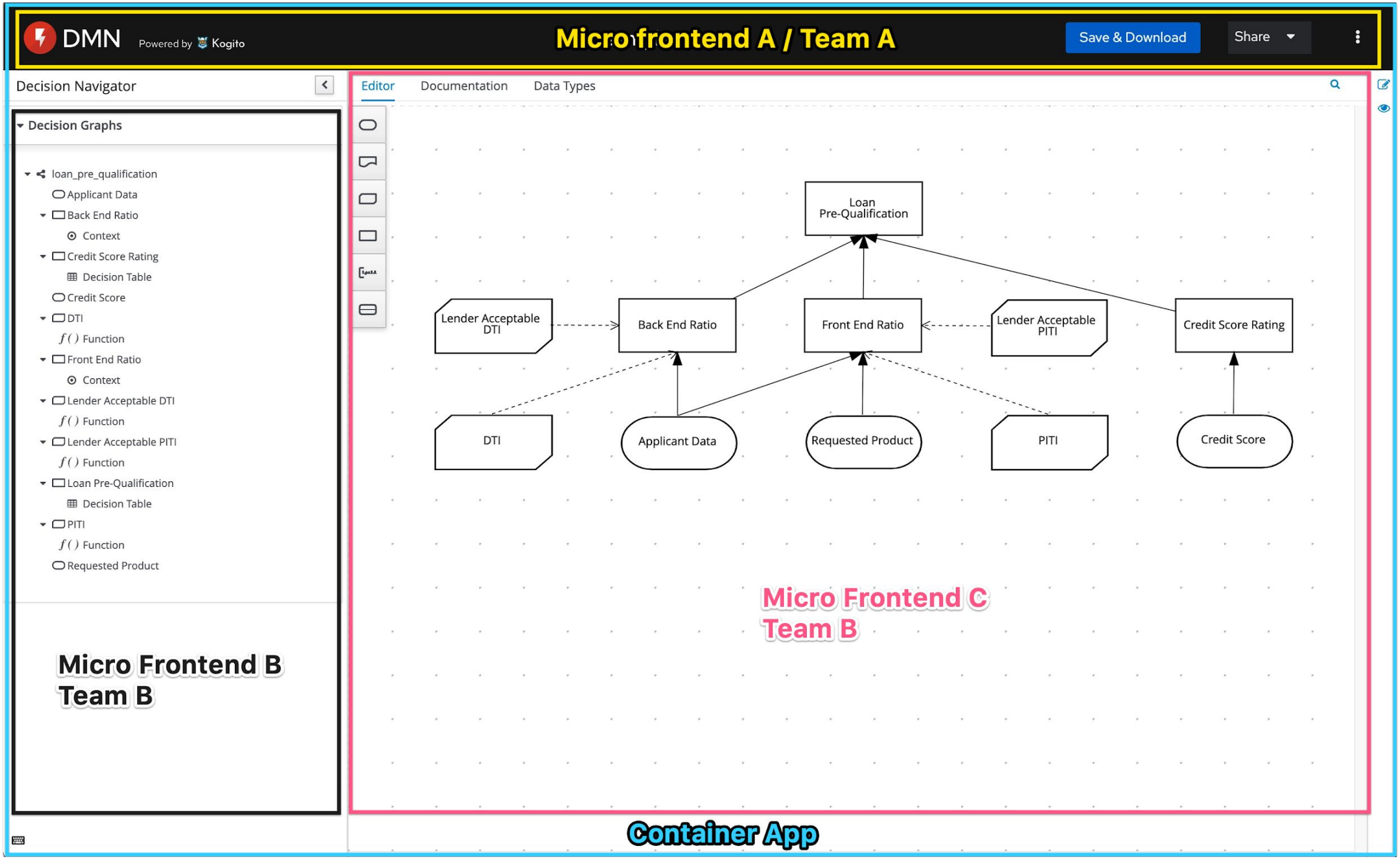
Each micro frontend can run as standalone

Independent deployment and releases

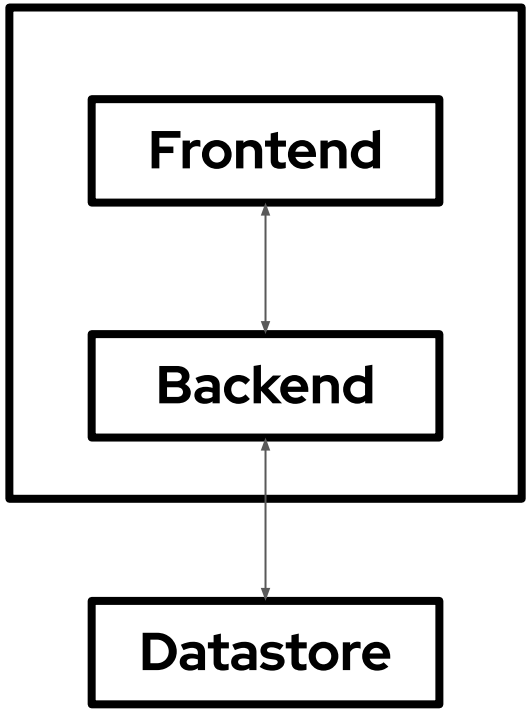
Autonomous Teams

Red Hat

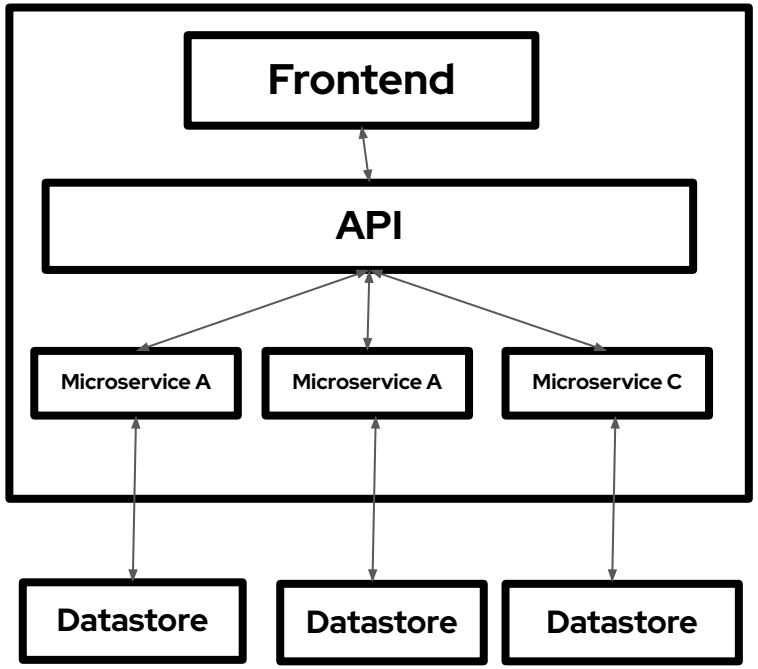
Example


DMN
Powered by Kogito
Micro frontend A / Team A
Save & Download
Share
Decision Navigator
Editor
Documentation
Data Types
Decision Graphs
loan_pre_qualification
Applicant Data
Back End Ratio
Context
Credit Score Rating
Decision Table
Credit Score
DTI
Function
Front End Ratio
Context
Lender Acceptable DTI
Function
Lender Acceptable PITI
Function
Loan Pre-Qualification
Decision Table
PITI
Function
Requested Product
Loan Pre-Qualification
Lender Acceptable DTI
Back End Ratio
Front End Ratio
Lender Acceptable PITI
Credit Score Rating
DTI
Applicant Data
Requested Product
PITI
Credit Score
Micro Frontend C
Team B
Micro Frontend B
Team B
Container App

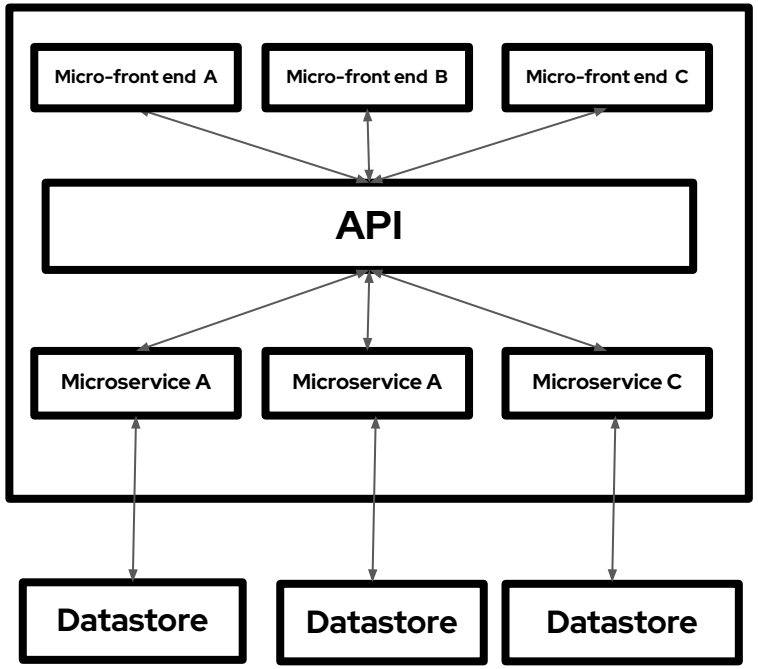
Monolith Web Application



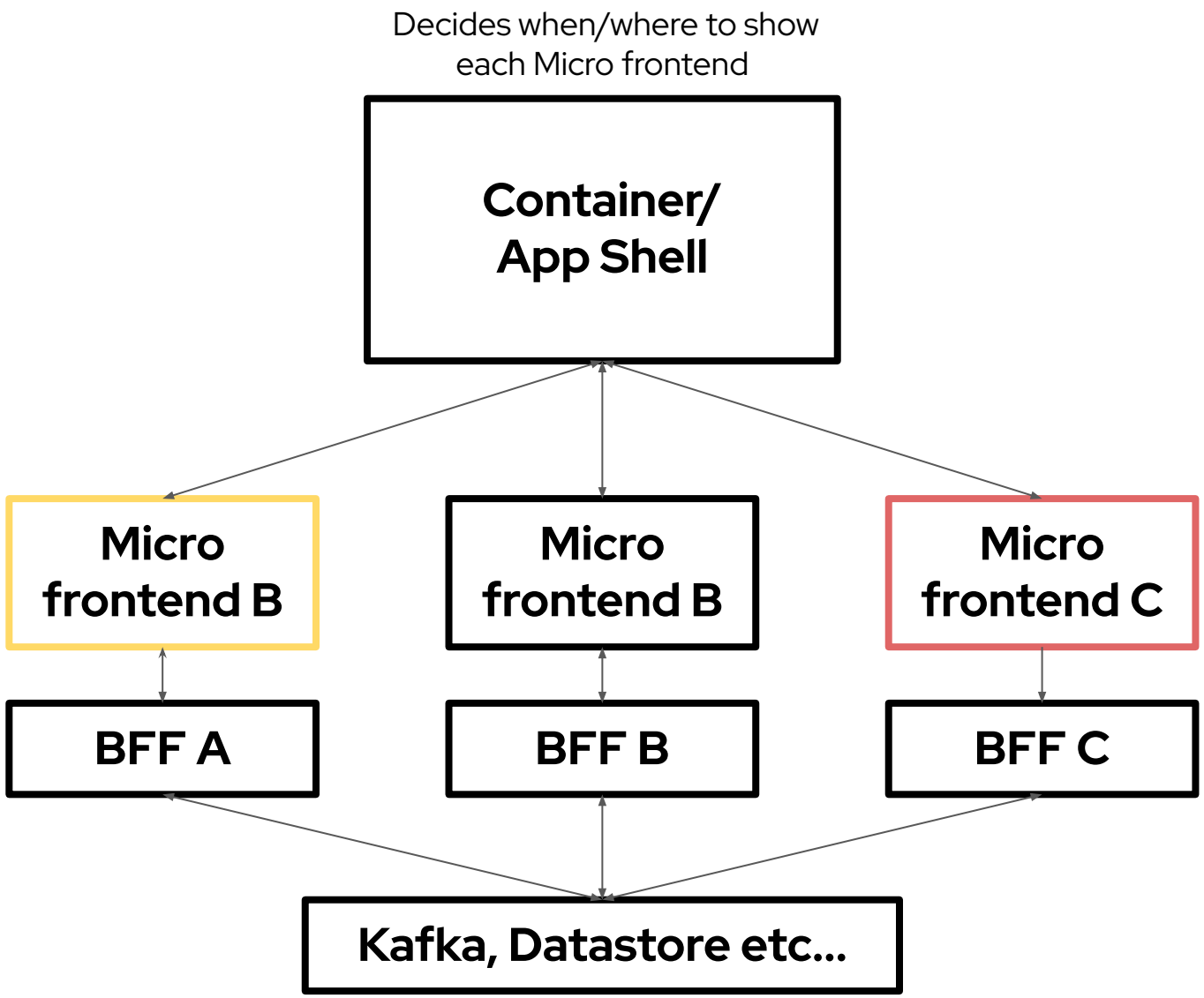
Web Application based on Microservices Architecture



Web Application based on Microservices and Micro-front end Architecture



Red Hat



No Micro frontend
communicate
directly to each other

Red Hat

Types of Integration

Run-Time integration

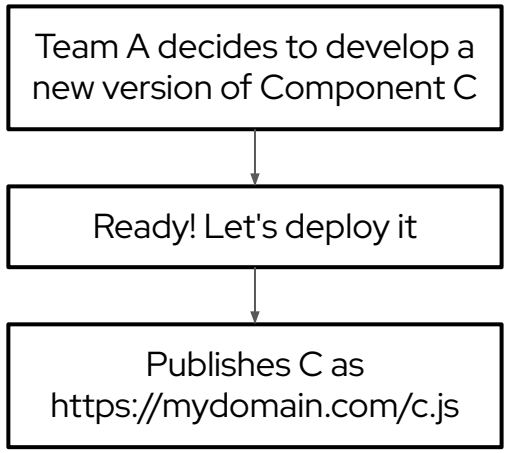
aka client-side integration:

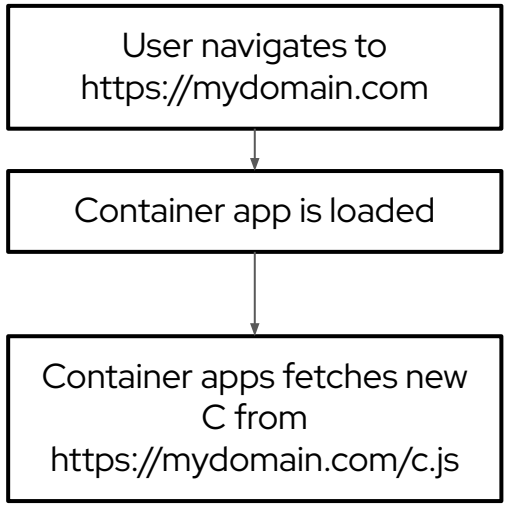
After the container gets loaded in the browser, it gets access to micro front end source code

Pros: A can be deployed independently at any time and can deploy different versions of it, and Container can decide which one to use

Cons: tooling + setup is far more complicated

Independent deployment makes it challenging to test/verify (build a good test suite for it)





Red Hat

Types of Integration

Build-time integration

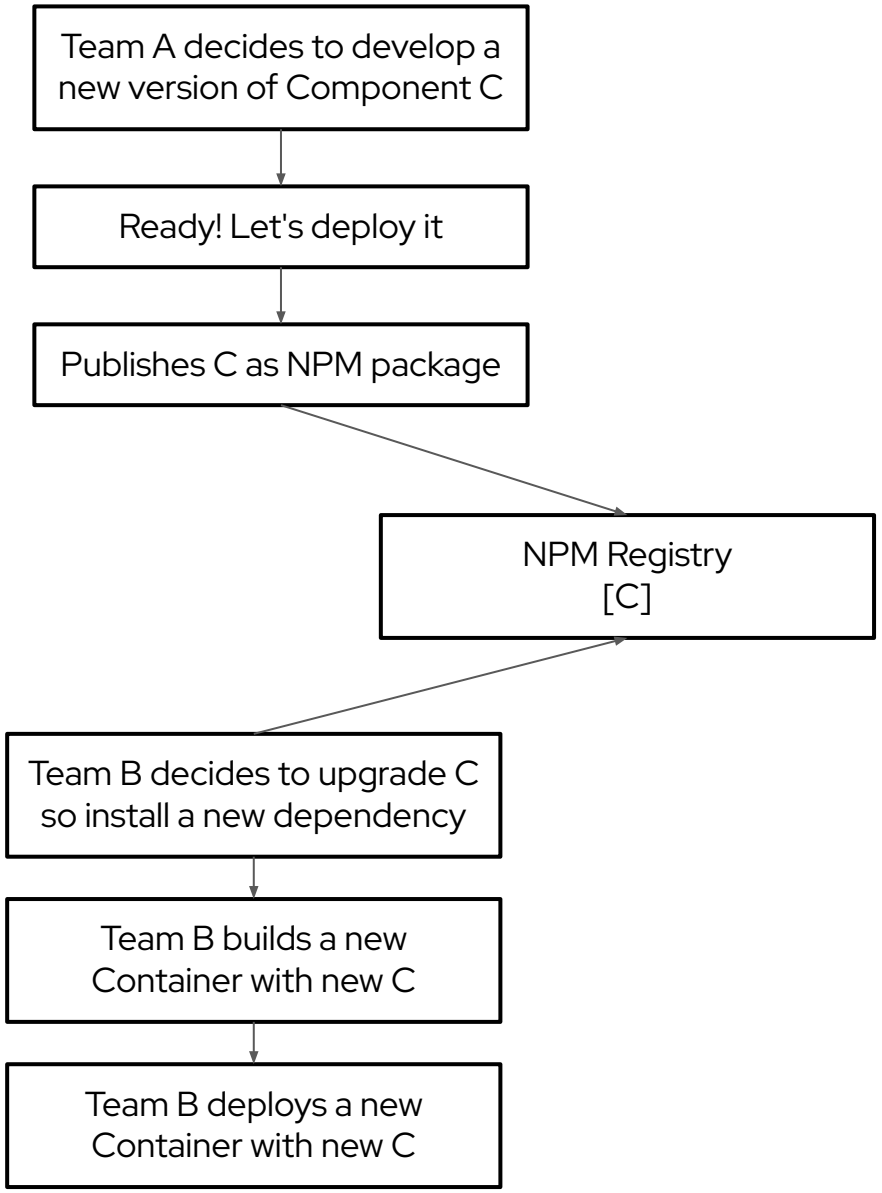
aka compile-time integration:

Before the container gets loaded in the browser, it gets access to micro frontend source code;

Foreign modules are accessible during build

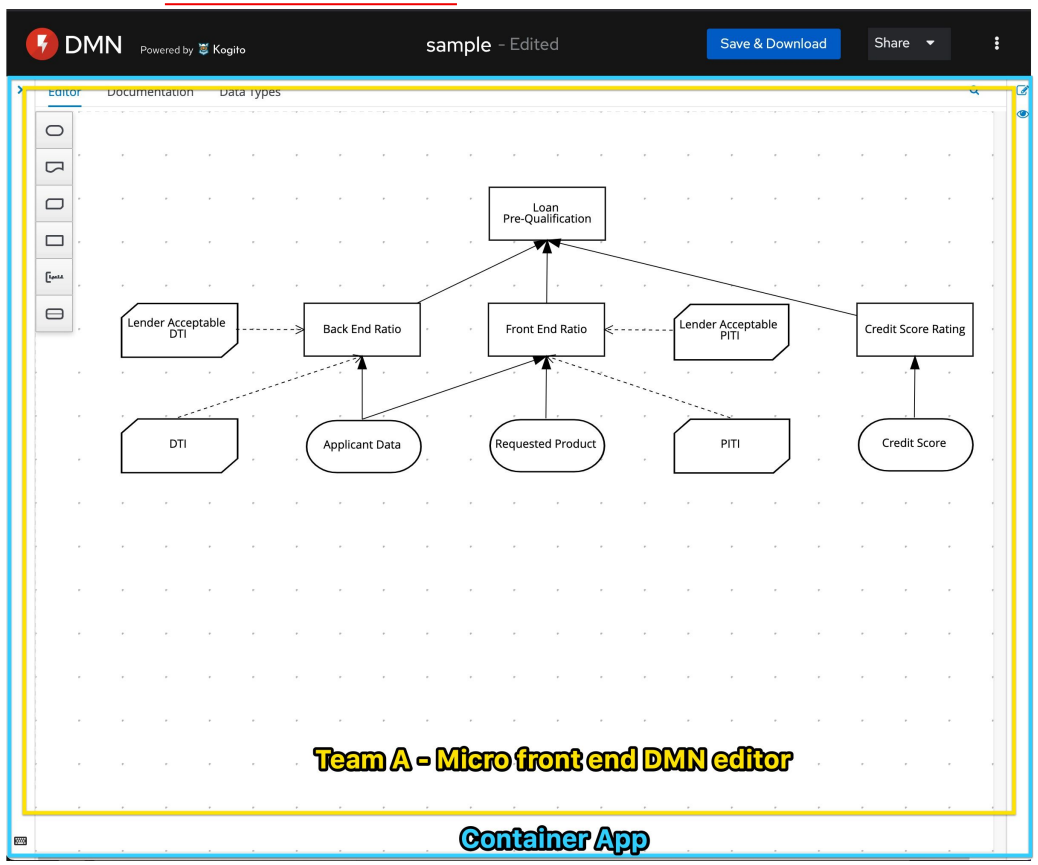
Pros: Easy to setup and understand

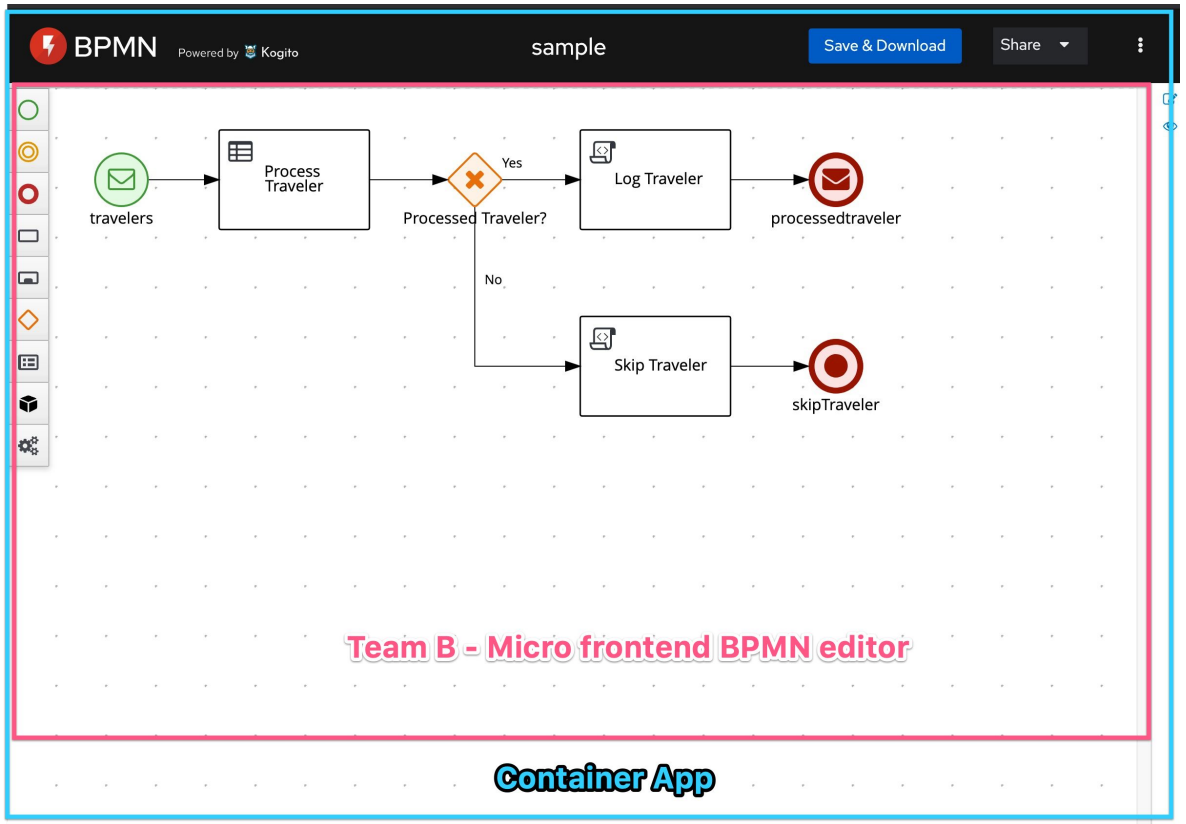
Cons: Container has to be re-deployed every time child has updated and tempting to tightly coupled Container + child together;



Red Hat

Build-time integration





Red Hat

Biggest Benefit - Autonomous Teams

Autonomous teams

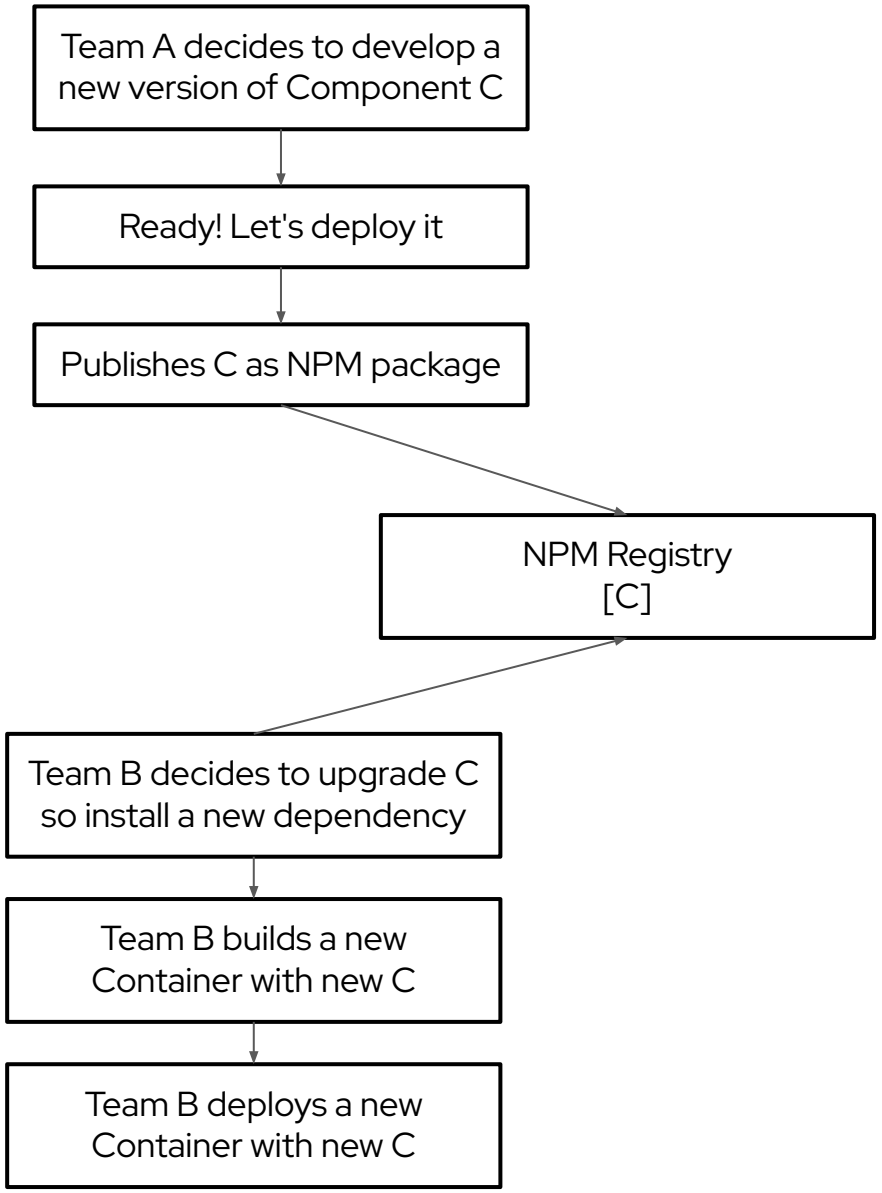
- Each team can run it's micro frontend in isolation

Pros

- Smaller/quickier build;
- Focus just on the problem;
- Less distraction, noise

Cons

- Bugs can appear just on container app
- Hard to run the complete experience;
- Hard to debug problems across entire system;
- Incoherent experiences;
- Our project:
  - Tricky issues can appears only on production

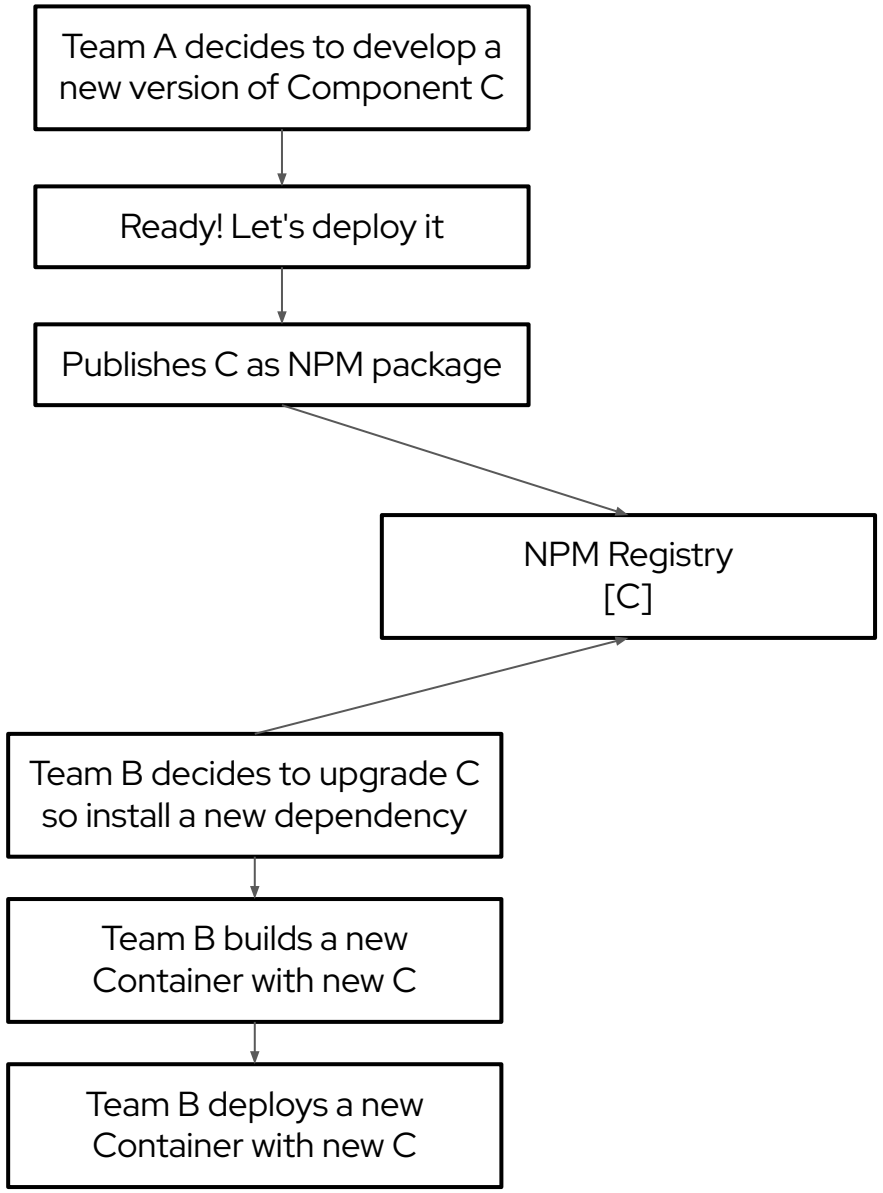


Red Hat

Another concerns - Styling

What you should do:

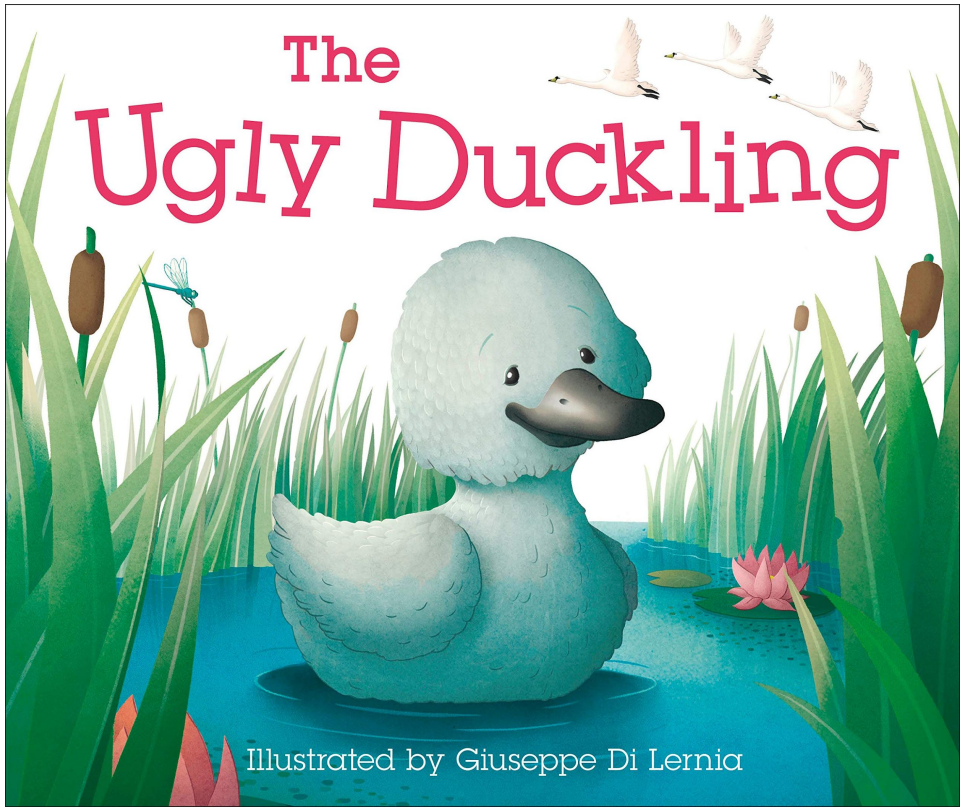
- *Custom CSS from your project:*
  - Use CSS-in-JS library
  - Use frameworks built-in component style scoping
    - Vue's and Angular has good ones
  - "Namespace" all your CSS
- *CSS coming from other libraries*
  - Use a component library that does css-in-js
  - Manually build the css library and apply namespacing techniques to it
  - Scope-it
    - Shadow DOM or **iframes**!



Red Hat

Context Isolation via iframes

- Nothing new, exciting, even a bit of 'yuck"
- **Pros**
  - Great degree of isolation;
    - Styling
    - Global variables
    - Shadow DOM was not a option in 2019
- Some libraries play directly with body of the page
- We only use it when necessary
- **Cons**:
  - Makes your app feel 'old'
  - Less flexible than other options
    - Hard to integrate routing, history;
    - Challenging to make the app responsive
    - Not Content-Security-Policy friendly
    - Harder to make apps communicate



Red Hat

index.html

Raw

```html
<html>
  <head>
    <title>Feed me!</title>
  </head>
  <body>
    <h1>Welcome to Feed me!</h1>

    <iframe id="micro-frontend-container"></iframe>

    <script type="text/javascript">
      const microFrontendsByRoute = {
        '/': 'https://browse.example.com/index.html',
        '/order-food': 'https://order.example.com/index.html',
        '/user-profile': 'https://profile.example.com/index.html',
      };

      const iframe = document.getElementById('micro-frontend-container');
      iframe.src = microFrontendsByRoute[window.location.pathname];
    </script>
  </body>
</html>
```


Example extracted from https://martinfowler.com/articles/micro-frontends.html

DMN Powered by Kogito

iframe#kogito-iframe 833 × 913

new-file

Editor Documentation Data Types

Welcome to this 5-minute tour

Learn more about the DMN online editor by taking this brief and interactive tour.

Let's go

Skip tour

Elements Console Sources

```
<body data-new-gr-c-s-check-loaded="14.1000.0" data-gr-ext-ins
gr-c-s-loaded="14.1000.0">
  <div id="app" style="height: 100vh;">
    <div class="pf-c-page">  grid
      <header class="pf-c-page__header kogito--editor__toolbar
      grid
      <main class="pf-c-page__main" tabindex="-1">
        <section class="pf-c-page__main-section pf-m-no-paddin
        style="flex-basis: 100%;">
          <iframe id="kogito-iframe" data-testid="kogito-iframe
          envelope.html" title="Kogito editor" data-envelope-cha
          style="display: flex; flex: 1 1 0%; flex-direction: co
          0%; height: 100%; border: none; margin: 0px; padding:
          dden;"> == $0
            #document
              <!DOCTYPE html>
              <html lang="en">
                <head>…</head>
                <body data-new-gr-c-s-check-loaded="14.1000.0"
                installed style="-webkit-tap-highlight-color: rgb
                  <div id="envelope-app">…</div>
                  <script src="envelope.js"></script>
                  <iframe id="org.kie.workbench.common.dmn.show
                  timeWebapp" tabindex="-1" style="position: absc
                  height: 0px; border: none; left: -1000px; top:
                  </iframe>
                    <div style="position: absolute; z-index: -327
                    idth: 10cm; height: 10cm; visibility: hidden;
                    <div style="position: absolute; inset: 0px;">
```

... c-page__main-section.pf-m-no-padding.pf-m-fill iframe#kogito-iframe

Styles Computed Layout Event Listeners DOM Breakpoints

Filter :hov .cls

```
element.style {
  display: flex;
  flex: 1 1 0%;
  flex-direction: column;
  width: 100%;
  height: 100%;
  border: none;
  margin: 0px;
  padding: 0px;
  overflow: hidden;
}
iframe {                                                <style>
  ~~border: 0;~~
}
img, embed, iframe, object, audio, video {              <style>
  max-width: 100%;
  ~~height: auto;~~
}
```

Console What's New

Highlights from the Chrome 89 update

Debugging support for Trusted Type violations

Breakpoint on Trusted Type violations and link to more information in the Issues tab.

Capture node screenshot beyond viewport

Capture node screenshot for a full node including content below the fold.

Red Hat

Gmail

is:starred

Compose

Inbox 4

Starred

Snoozed

Sent

Drafts

More

Meet

New meeting

Join a meeting

Hangouts

Kie

ignatowicz@gmail.com
Hello there!

No starred messages. Stars let you give messages a special status to make them easier to find. To star a message, click on the star outline beside any message or conversation.

0 GB of 15 GB used

Terms · Priv

Po

ignatowicz@gmail.com

Let's chat on Hangouts!

Hello there!

ignatowicz@gmail.com • Now

div.nH.aJl.nn 267 × 745

Elements Console 155 2

```
▸<div class="nH aJl nn" style="width: 6px; height: 1
display: none;">…</div>
▾<div class="nH aJl nn" style="width: 267px; height:
">
  ▾<div class="aJn">
    ▾<div id=":il">
      ▾<div class="nH aAl" style="height: 380px;">
        ▾<iframe src="https://hangouts.google.com/wel
        =1615964259&hl=en&pvt…Uq62Fv-M2woU8ZJMVmLx7Hi
        SSCij77kDw%3D%3D&prop=gmail#epreld" framebord
        "no" class="a7A" width="262" height="380"> ==
```

… nH div.no div.nH.aJl.nn div.aJn div#\:il div.nH.aAl iframe.a7A …

Styles Computed Layout Event Listeners DOM Breakpoints

Filter :hov .cls

```
element.style {
}
.aAl .a7A, .aAl.nH {                                <style>
  border-top-left-radius: 8px;
  border-top-right-radius: 8px;
}
.a7A, .Jc iframe {                                  <style>
  display: block;
}
iframe[Attributes Style] {
  border-top-width: 0px;
  border-right-width: 0px;
  border-bottom-width: 0px;
  border-left-width: 0px;
  width: 262px;
  height: 380px;
}
```

Console What's New

Highlights from the Chrome 89 update

Debugging support for Trusted Type violations
Breakpoint on Trusted Type violations and link to more information in the Issues tab.

Capture node screenshot beyond viewport
Capture node screenshot for a full node including content below the fold.

Context Isolation via iframes

Red Hat

Micro frontend Spectrum

Total independence ⟷ **Strategic collaboration**

Each team chooses tech stack

Each micro frontend makes it's own API calls

App is composed of fully functional micro apps

Each micro frontend has it's own CI/CD

Agrees on tech stack

Container handles all API calls

Share 'dumb' components

Shared CI/CD

https://twitter.com/housecor/status/1139504822930092033/photo/1

Going deeper on Multiplying Architecture

Red Hat

Overview | Serverless Logic We

start.kubesmarts.org/#/

Serverless Logic Web Tools

- Overview
- Recent Models
- Sample Catalog
- Documentation

Welcome to Serverless Logic Web Tools

Add-on service to create and synchronize your Serverless Workflow, Decision files, and Dashbuilder files

The Serverless Logic Web Tools is a web application that enables you to create and synchronize your Serverless Workflow, Serverless Decision, and Dashbuilder files in a single interface. Also, the Serverless Logic Web Tools application provides the integrations that are needed to deploy and test the Serverless Workflow models in development mode.

Get Started with Serverless Logic Web Tools

Create

Serverless Workflow

Define orchestration logic for services.

New Workflow: JSON YAML

Serverless Decision

Define decision logic for services.

New Decision: JSON YAML

Dashboard

Define data visualization from data extracted from applications.

New Dashboard: YAML

Import

From URL

Import a GitHub Repository, a GitHub Gist, or any other file URL.

URL

Import

Upload

Drag & drop files and folders here...

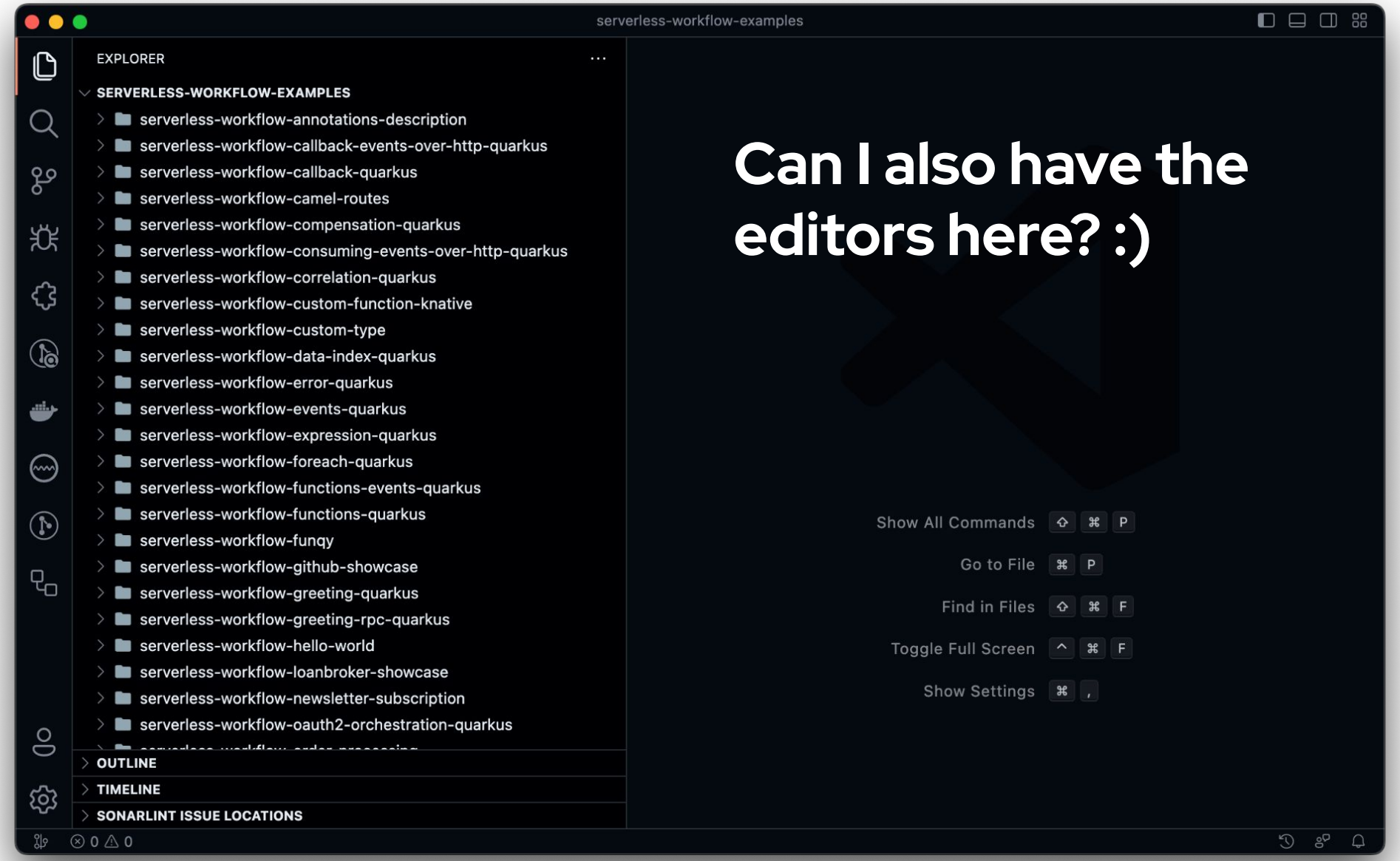
or

Select files...

Select folder...

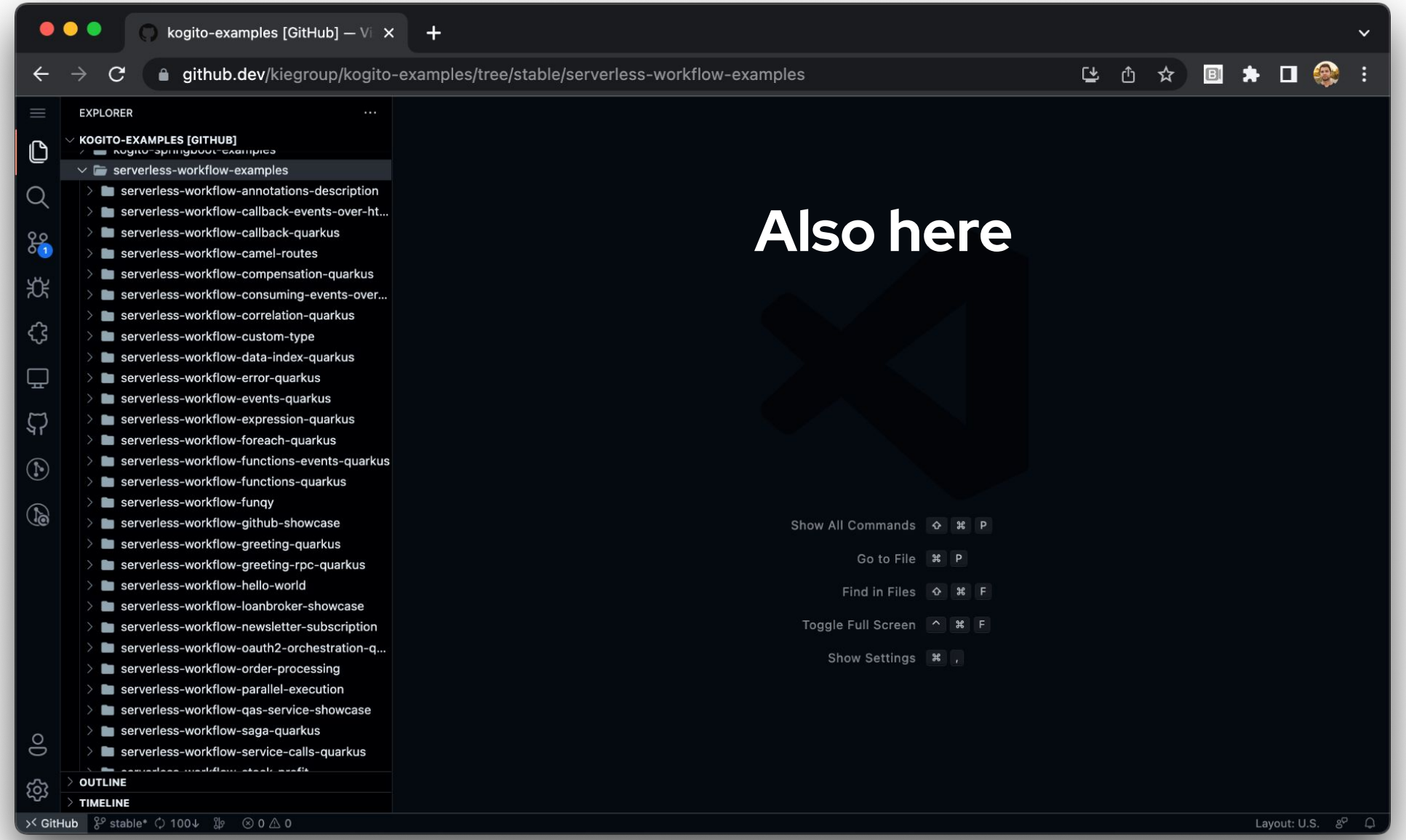
Can I have my editor here?

Red Hat



Can I also have the editors here? :)

Red Hat


kogito-examples [GitHub] — Vi
github.dev/kiegroup/kogito-examples/tree/stable/serverless-workflow-examples
EXPLORER
KOGITO-EXAMPLES [GITHUB]
serverless-workflow-examples
serverless-workflow-annotations-description
serverless-workflow-callback-events-over-ht...
serverless-workflow-callback-quarkus
serverless-workflow-camel-routes
serverless-workflow-compensation-quarkus
serverless-workflow-consuming-events-over...
serverless-workflow-correlation-quarkus
serverless-workflow-custom-type
serverless-workflow-data-index-quarkus
serverless-workflow-error-quarkus
serverless-workflow-events-quarkus
serverless-workflow-expression-quarkus
serverless-workflow-foreach-quarkus
serverless-workflow-functions-events-quarkus
serverless-workflow-functions-quarkus
serverless-workflow-funqy
serverless-workflow-github-showcase
serverless-workflow-greeting-quarkus
serverless-workflow-greeting-rpc-quarkus
serverless-workflow-hello-world
serverless-workflow-loanbroker-showcase
serverless-workflow-newsletter-subscription
serverless-workflow-oauth2-orchestration-q...
serverless-workflow-order-processing
serverless-workflow-parallel-execution
serverless-workflow-qas-service-showcase
serverless-workflow-saga-quarkus
serverless-workflow-service-calls-quarkus
OUTLINE
TIMELINE
Also here
Show All Commands
Go to File
Find in Files
Toggle Full Screen
Show Settings
GitHub
stable*
Layout: U.S.

Red Hat

kogito-examples/serverless-w

github.com/kiegroup/kogito-examples/blob/stable/serverless-workflow-examples/serverless-workflow-greeting-quarkus/src/main/resour...

kiegroup / kogito-examples

Type / to search

Code | Issues 22 | Pull requests 35 | Discussions | Actions | Projects | Wiki | Security | Insights | Settings

stable | kogito-examples / serverless-workflow-examples / serverless-workflow-greeting-quarkus / src / main / resources / jsongreet.sw.json | Go to file

hasys KOGITO-9027 Align workflows with specification and tooling exporter (#... a893520 · 2 months ago History

Code | Blame | 65 lines (65 loc) · 1.36 KB | Raw

```
{
  "id": "jsongreet",
  "version": "1.0",
  "name": "Greeting workflow",
  "description": "JSON based greeting workflow",
  "start": "ChooseOnLanguage",
  "functions": [
    {
      "name": "greetFunction",
      "type": "custom",
      "operation": "sysout"
    }
  ],
  "states": [
    {
      "name": "ChooseOnLanguage",
      "type": "switch",
      "dataConditions": [
        {
          "condition": "${ .language == \"English\" }",
          "transition": "GreetInEnglish"
        },
        {
          "condition": "${ .language == \"Spanish\" }",
          "transition": "GreetInSpanish"
        }
```

And here :)

Red Hat

Introducing Multiplying Architecture

Red Hat

The Abstractions

Core Components

Channel

Top level abstraction that represents the hosting environment, like a website or a desktop application.

Envelope

Enable transparent communication between Components (View/Editor) and Channel

View

View is a portable set of widgets that are exposed as an unit to the Channel through the Envelope.

Editor

Editor is a specialized type of View, that gets a file content as input and is able to serve the content state back to the Channel through the Envelope.

greeting.sw.json | Serverless L

start.kubesmarts.org/#/3f4c2a04-ae29-499e-964b-9bd43eae7b1c/file/greeting.sw.json

Serverless Logic Web Tools

Overview

Recent Models

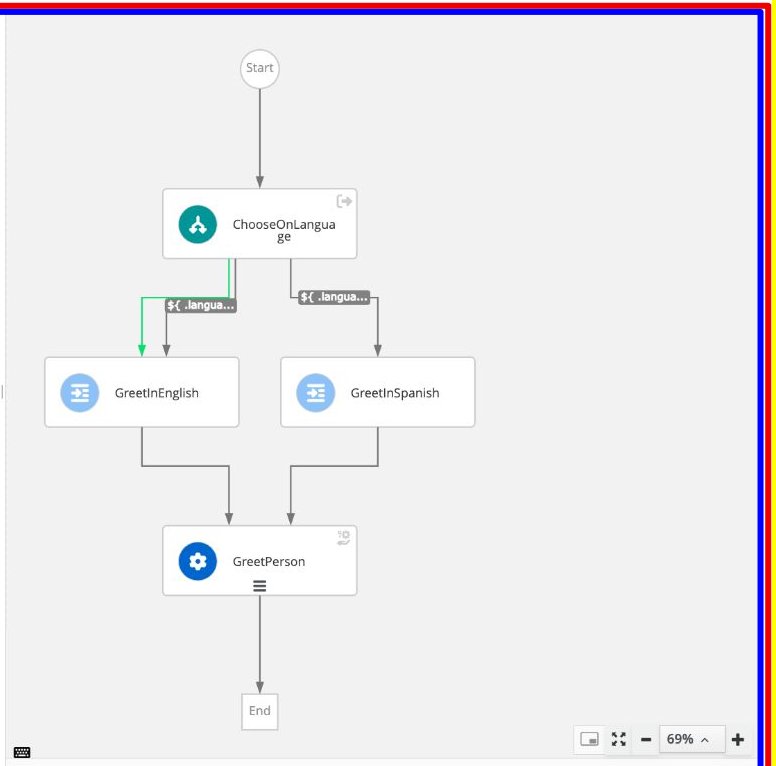
Sample Catalog

Documentation

Serverless Workflow **greeting**

New file | Try on OpenShift | Share

```
{
  "id": "jsongreet",
  "version": "1.0",
  "specVersion": "0.8",
  "name": "Greeting workflow",
  "description": "JSON based greeting workflow",
  "start": "ChooseOnLanguage",
  + Add function...
  "functions": [
    {
      "name": "greetFunction",
      "type": "custom",
      "operation": "sysout"
    }
  ],
  + Add state...
  "states": [
    {
      "name": "ChooseOnLanguage",
      "type": "switch",
      "dataConditions": [
        {
          "condition": "${ .language == \"English\" }",
          "transition": "GreetInEnglish"
        },
        {
          "condition": "${ .language == \"Spanish\" }",
          "transition": "GreetInSpanish"
        }
      ],
      "defaultCondition": {
        "transition": "GreetInEnglish"
      }
    },
    {
      "name": "GreetInEnglish",
      "type": "inject",
      "data": {
        "greeting": "Hello from JSON Workflow, "
```



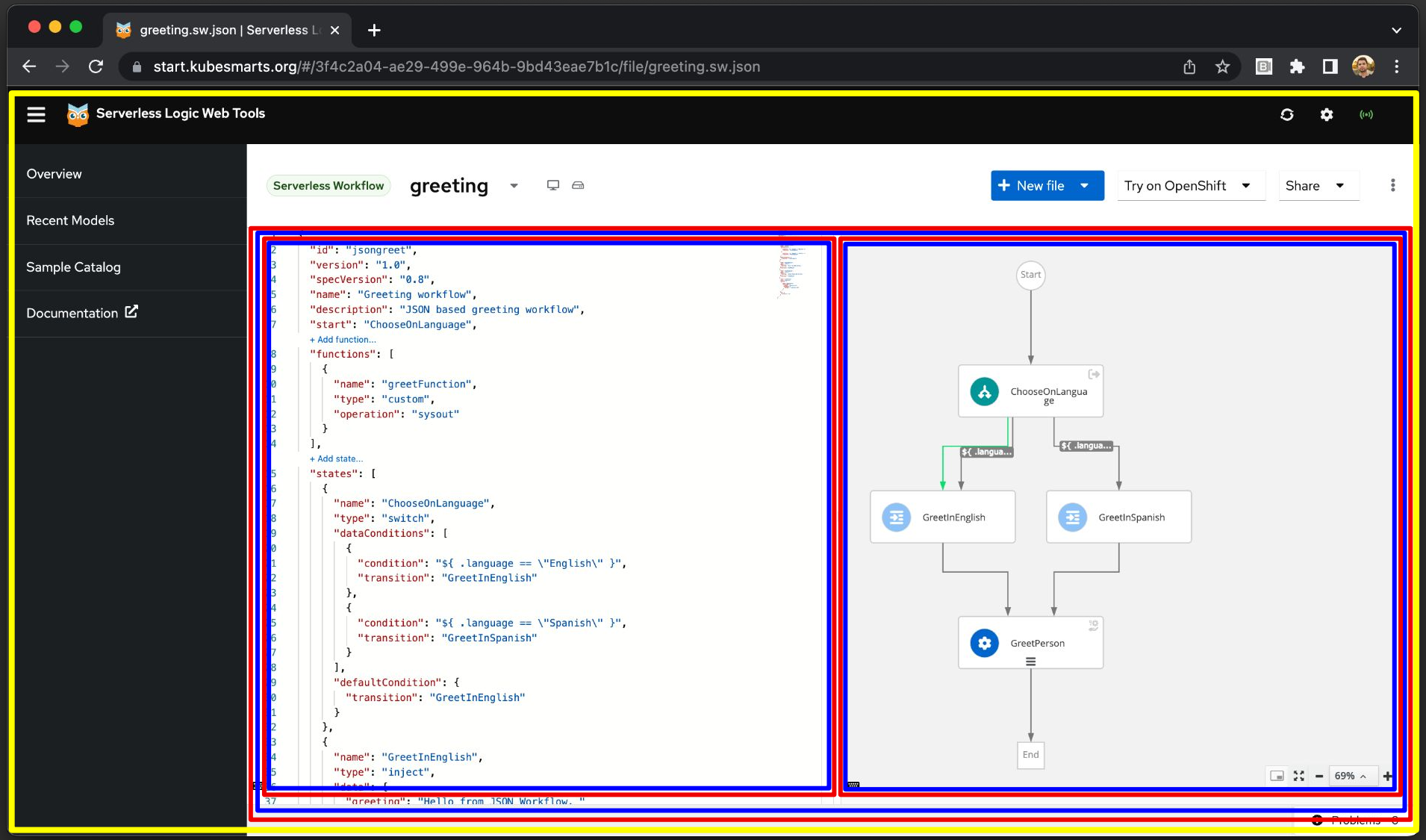
Problems 0

Online channel

Envelope instance

Editor

Red Hat



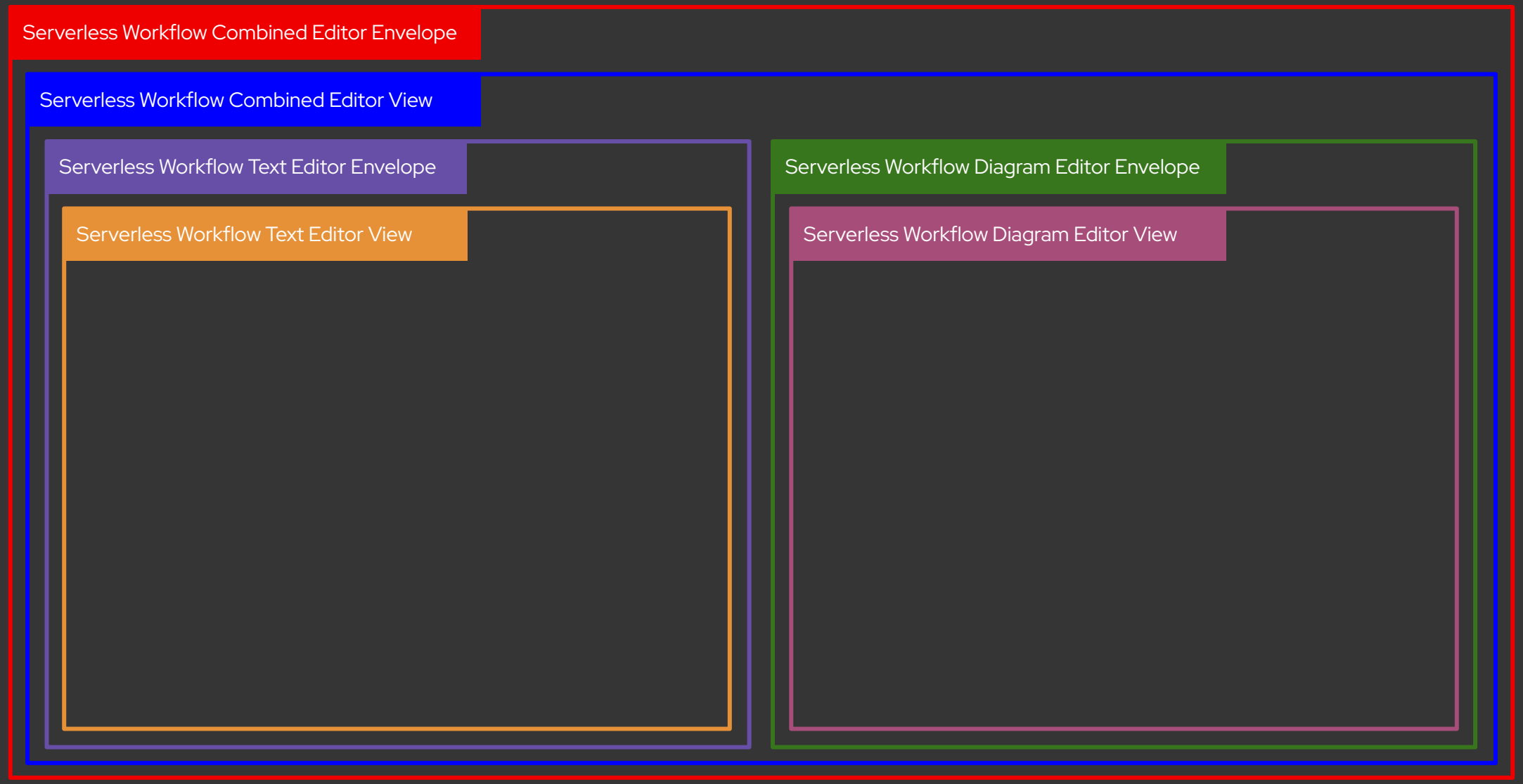
Online channel

Envelope instance

Editor

an envelope that combines two envelopes

Red Hat


Serverless Workflow Combined Editor Envelope
Serverless Workflow Combined Editor View
Serverless Workflow Text Editor Envelope
Serverless Workflow Text Editor View
Serverless Workflow Diagram Editor Envelope
Serverless Workflow Diagram Editor View

Red Hat

jsongreet.sw.json — serverless-workflow-examples

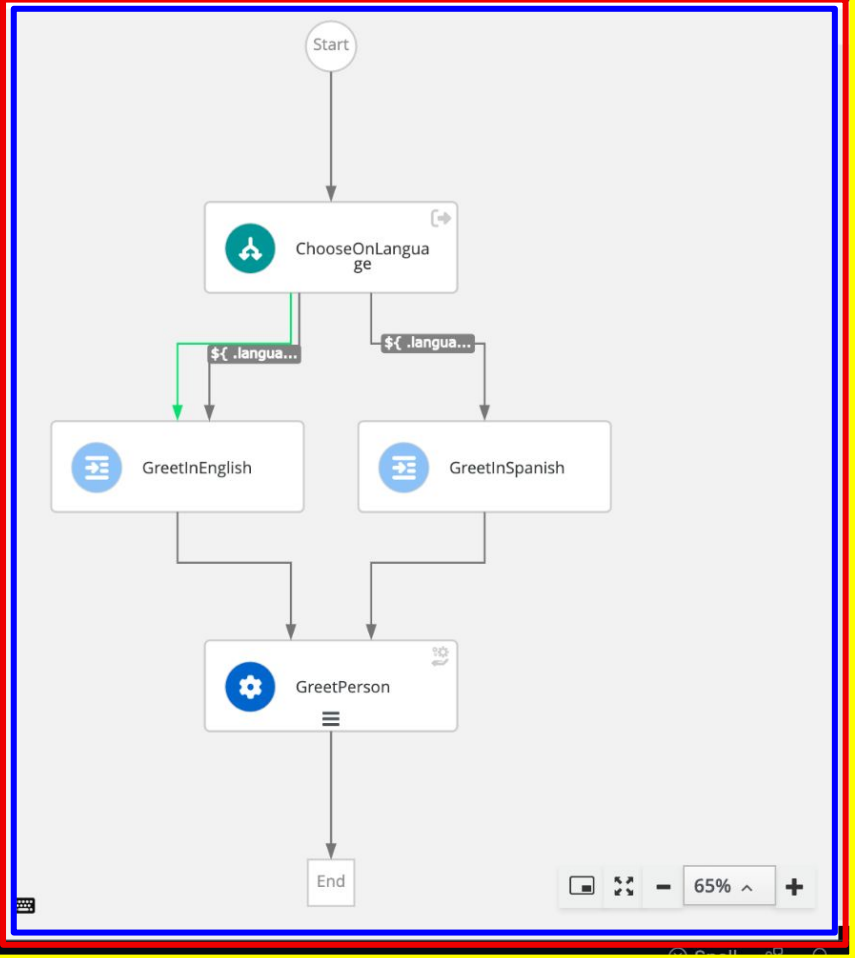
jsongreet.sw.json classes

serverless-workflow-greeting-quarkus > target > classes > jsongreet.sw.json > ...

```
1  {
2    "id": "jsongreet",
3    "version": "1.0",
4    "specVersion": "0.8",
5    "name": "Greeting workflow",
6    "description": "JSON based greeting workflow",
7    "start": "ChooseOnLanguage",
     + Add function... | ↪ Log in Service Registries...
8    "functions": [
9      {
10       "name": "greetFunction",
11       "type": "custom",
12       "operation": "sysout"
13     }
14   ],
     + Add state...
15   "states": [
16     {
17       "name": "ChooseOnLanguage",
18       "type": "switch",
19       "dataConditions": [
20         {
21           "condition": "${ .language == \"English\"
22           "transition": "GreetInEnglish"
23         },
24         {
25           "condition": "${ .language == \"Spanish\"
26           "transition": "GreetInSpanish"
27         }
28       ],
29       "defaultCondition": {
```

jsongreet.sw.json classes

serverless-workflow-greeting-quarkus > target > classes > jsongreet.sw.json



VS Code channel

Envelope instance

Editor

Red Hat

jsongreet.sw.json — kogito-exa

github.dev/kiegroup/kogito-examples/tree/stable/serverless-workflow-examples

jsongreet.sw.json resources 1, M

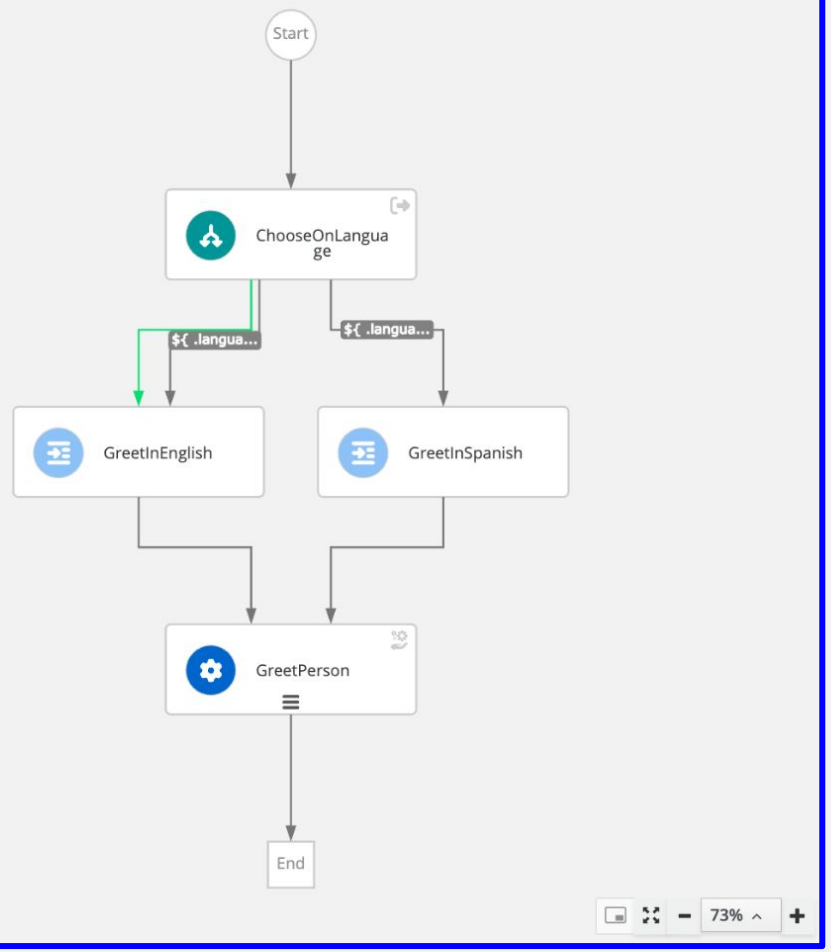
v-examples > serverless-workflow-greeting-quarkus > src > main > resources > jsongreet.sw.json > ...

```
1  {
2    "id": "jsongreet",
3    "version": "1.0",
4    "specVersion": "0.8",
5    "name": "Greeting workflow",
6    "description": "JSON based greeting workflow",
7    "start": "ChooseOnLanguage",
       + Add function...
8    "functions": [
9      {
10       "name": "greetFunction",
11       "type": "custom",
12       "operation": "sysout"
13     }
14   ],
       + Add state...
15   "states": [
16     {
17       "name": "ChooseOnLanguage",
18       "type": "switch",
19       "dataConditions": [
20         {
21           "condition": "${ .language == \"English\" }",
22           "transition": "GreetInEnglish"
23         },
24         {
25           "condition": "${ .language == \"Spanish\" }",
26           "transition": "GreetInSpanish"
27         }
28       ],
29       "defaultCondition": {
30         "transition": "GreetInEnglish"
31       }
32     },
33     {
34       "name": "GreetInEnglish",
35       "type": "inject",
```

GitHub stable* 100↓ 0 1 Layout: U.S.

jsongreet.sw.json resources 1, M

serverless-workflow-examples > serverless-workflow-greeting-quarkus > src > main > resources >



VS Code channel

Envelope instance

Editor

Red Hat

Browser extension channel

kiegroup / kogito-examples

Type / to search

Code | Issues 22 | Pull requests 35 | Discussions | Actions | Projects | Wiki | Security | Insights | Settings

stable | kogito-examples / serverless-workflow-examples / serverless-workflow-greeting-quarkus / src / main / resources / jsongreet.sw.json

Go to file

hasys KOGITO-9027 Align workflows with specification and tooling exporter (#...

a893520 · 2 months ago | History

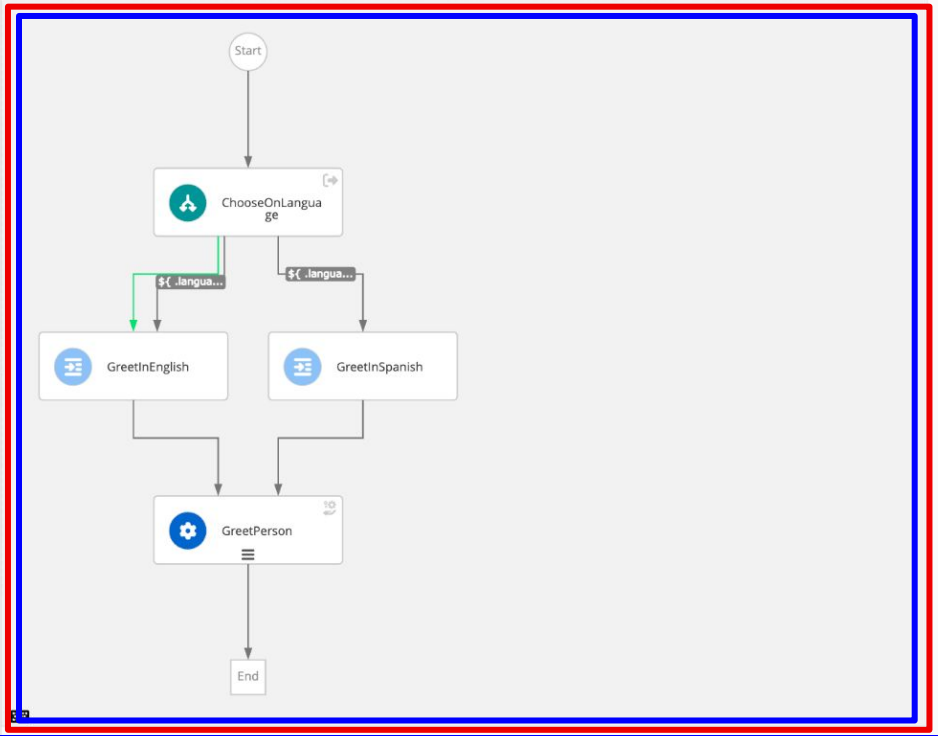
This is a read-only visualization.

Full Screen | See as source

Code | Blame | 65 lines (65 loc) · 1.36 KB

Raw

```
{
  "id": "jsongreet",
  "version": "1.0",
  "name": "Greeting workflow",
  "description": "JSON based greeting workflow",
  "start": "ChooseOnLanguage",
  "functions": [
    {
      "name": "greetFunction",
      "type": "custom",
      "operation": "sysout"
    }
  ],
  "states": [
    {
      "name": "ChooseOnLanguage",
      "type": "switch",
      "dataConditions": [
        {
          "condition": "${ .language == \"English\" }",
          "transition": "GreetInEnglish"
        },
        {
          "condition": "${ .language == \"Spanish\" }",
          "transition": "GreetInSpanish"
        }
      ],
      "defaultCondition": {
        "transition": "GreetInEnglish"
      }
    },
    {
```



Envelope instance

Editor

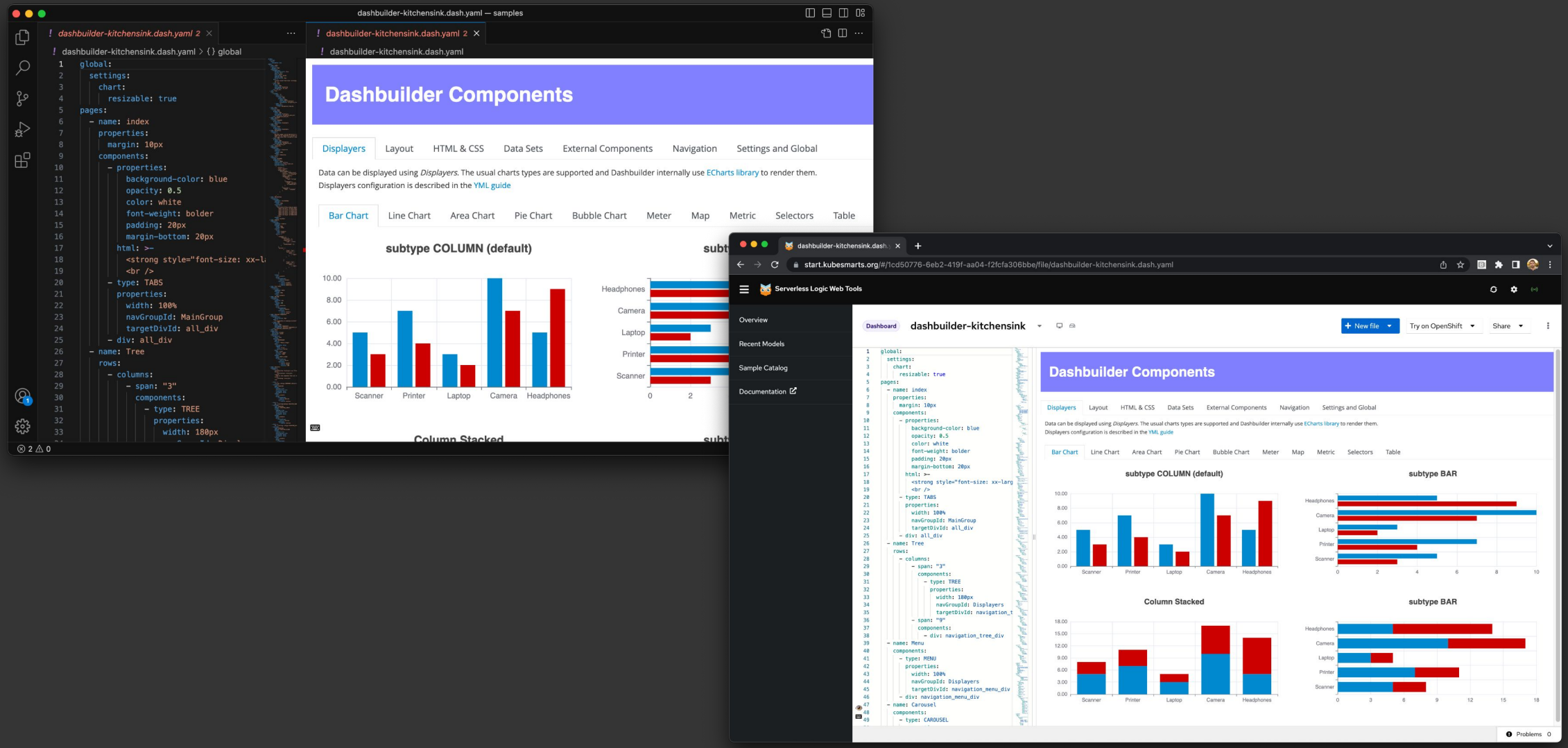
Red Hat

```
interface MyEnvelopeApi {
    undo(): void;
    redo(): void;
    contentRequest(): Promise<EditorContent>;
    previewRequest(): Promise<string>;
    ...
}
```

```
interface MyChannelApi {
    newEdit(edit: WorkspaceEdit): void;
    openFile(path: string): void;
    listResources(request: ResourceListRequest): Promise<ResourcesList>;
    setContentError(content: EditorContent);
    ...
}
```

More editors

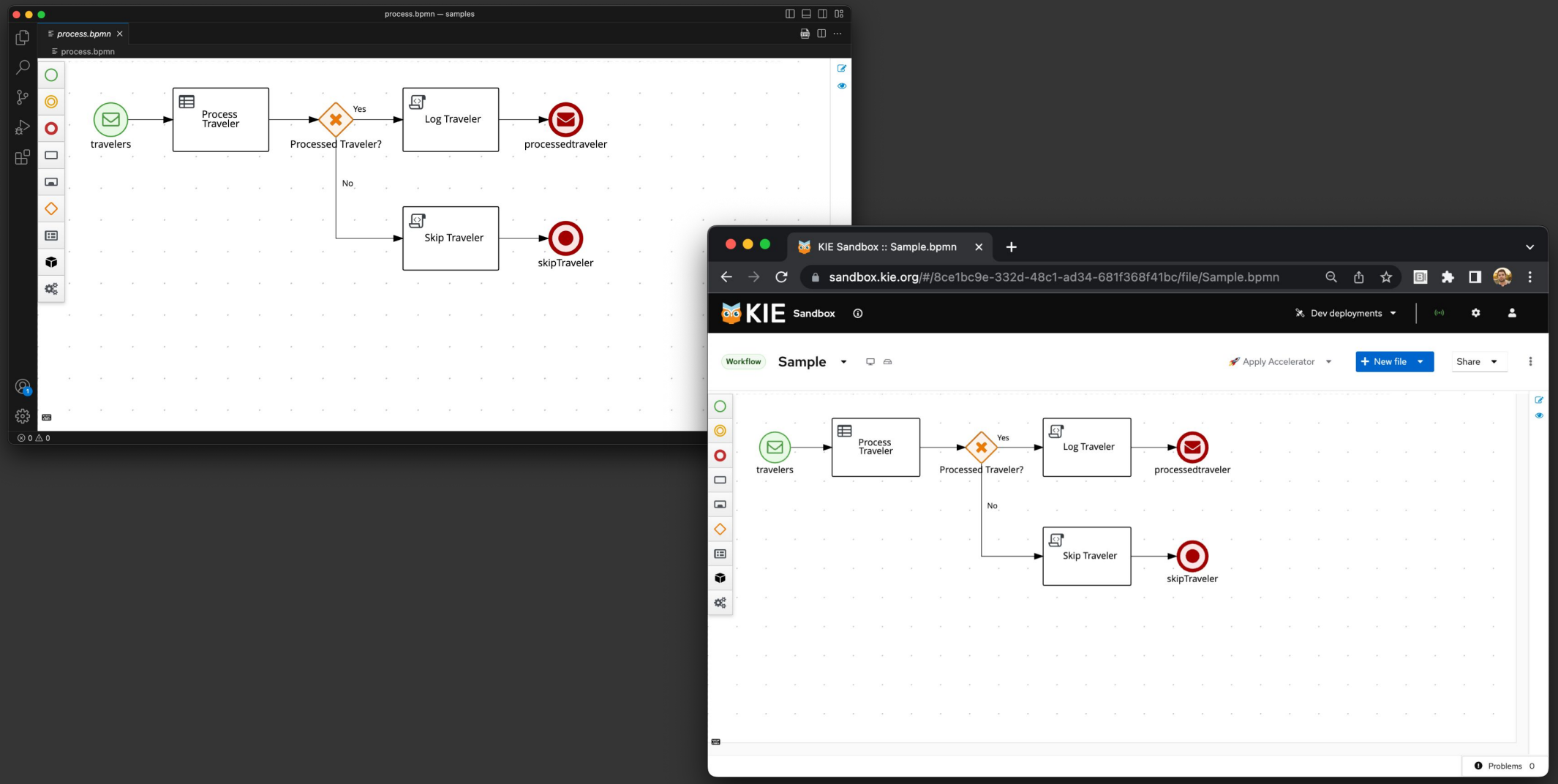
Dashbuilder editor



start.kubesmarts.org

Red Hat

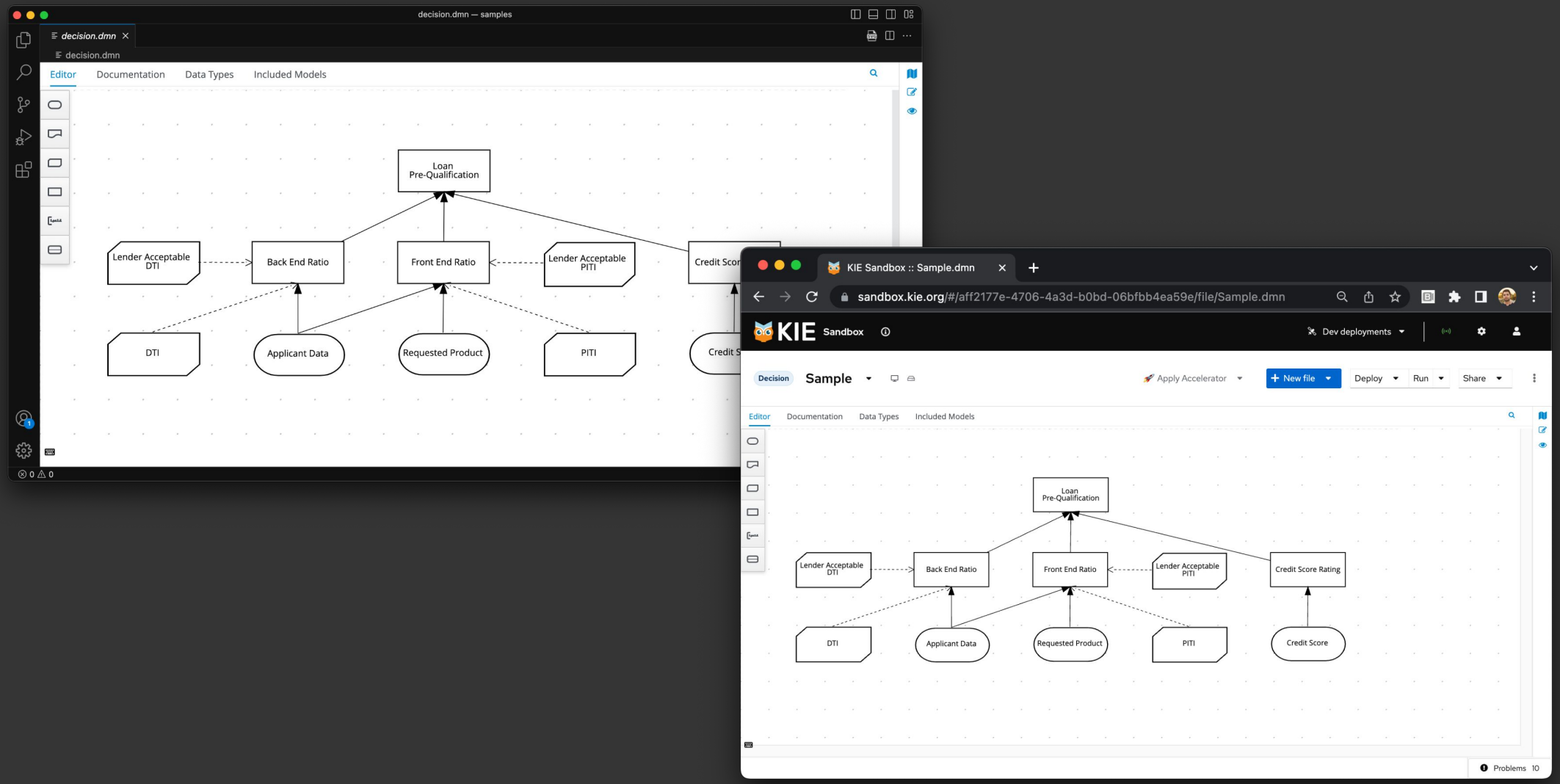
BPMN editor



bpmn.new

Red Hat

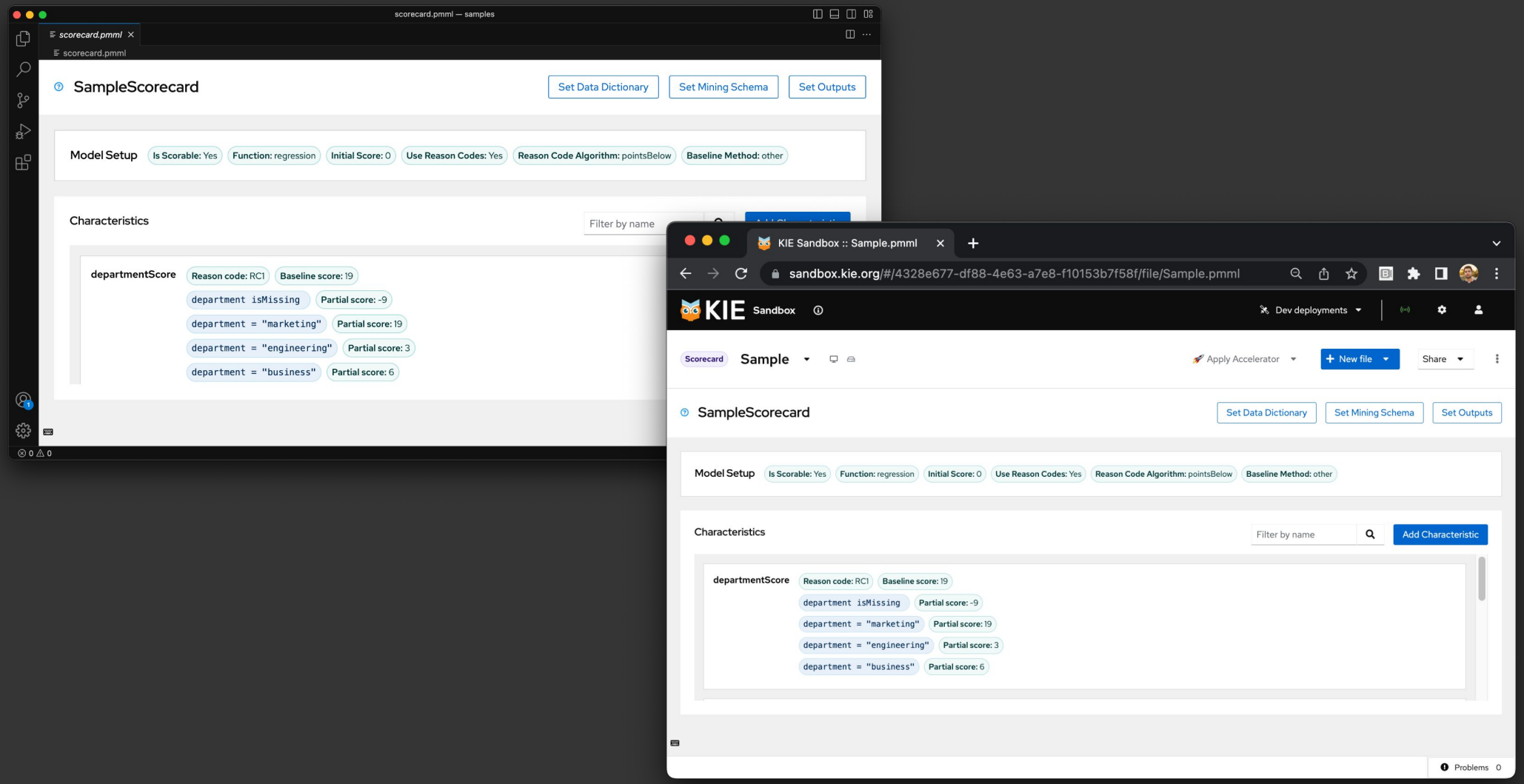
DMN editor



dmn.new

Red Hat

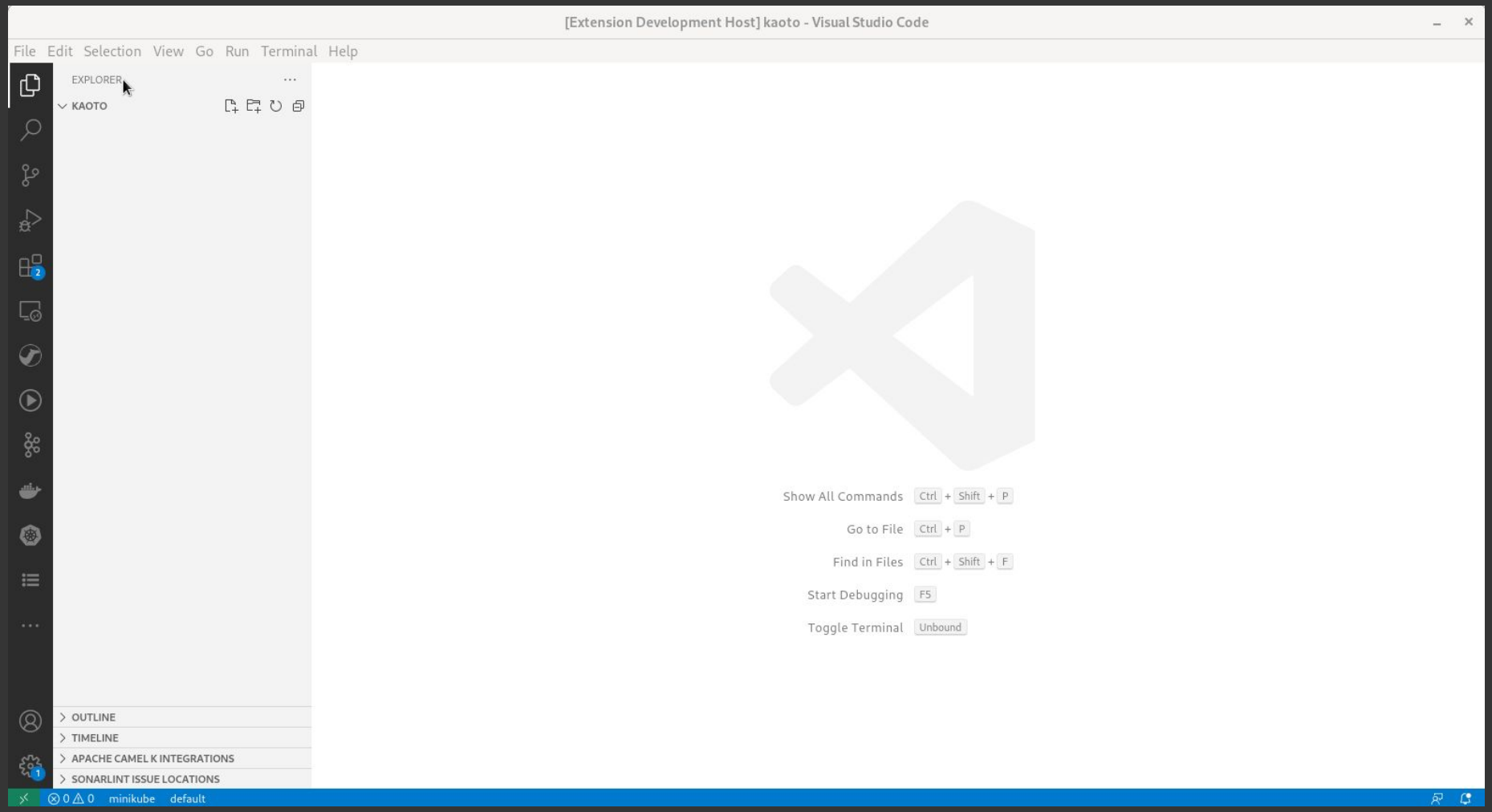
PMML editor



pmml.new

Red Hat

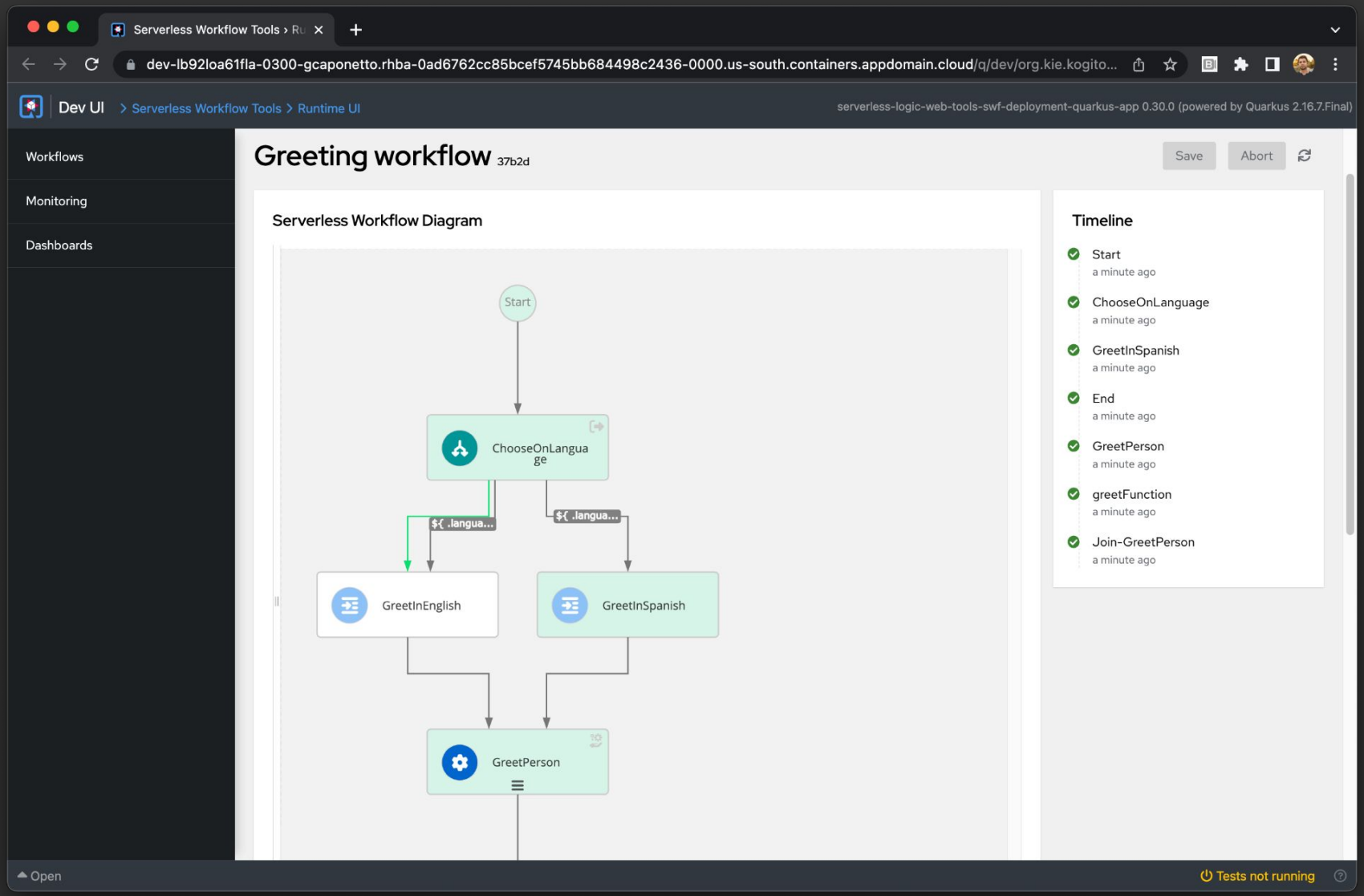
Kaoto editor



kaoto.io

Red Hat

Serverless Workflow Tools



Red Hat

More on:

github.com/kiegroup/kie-tools

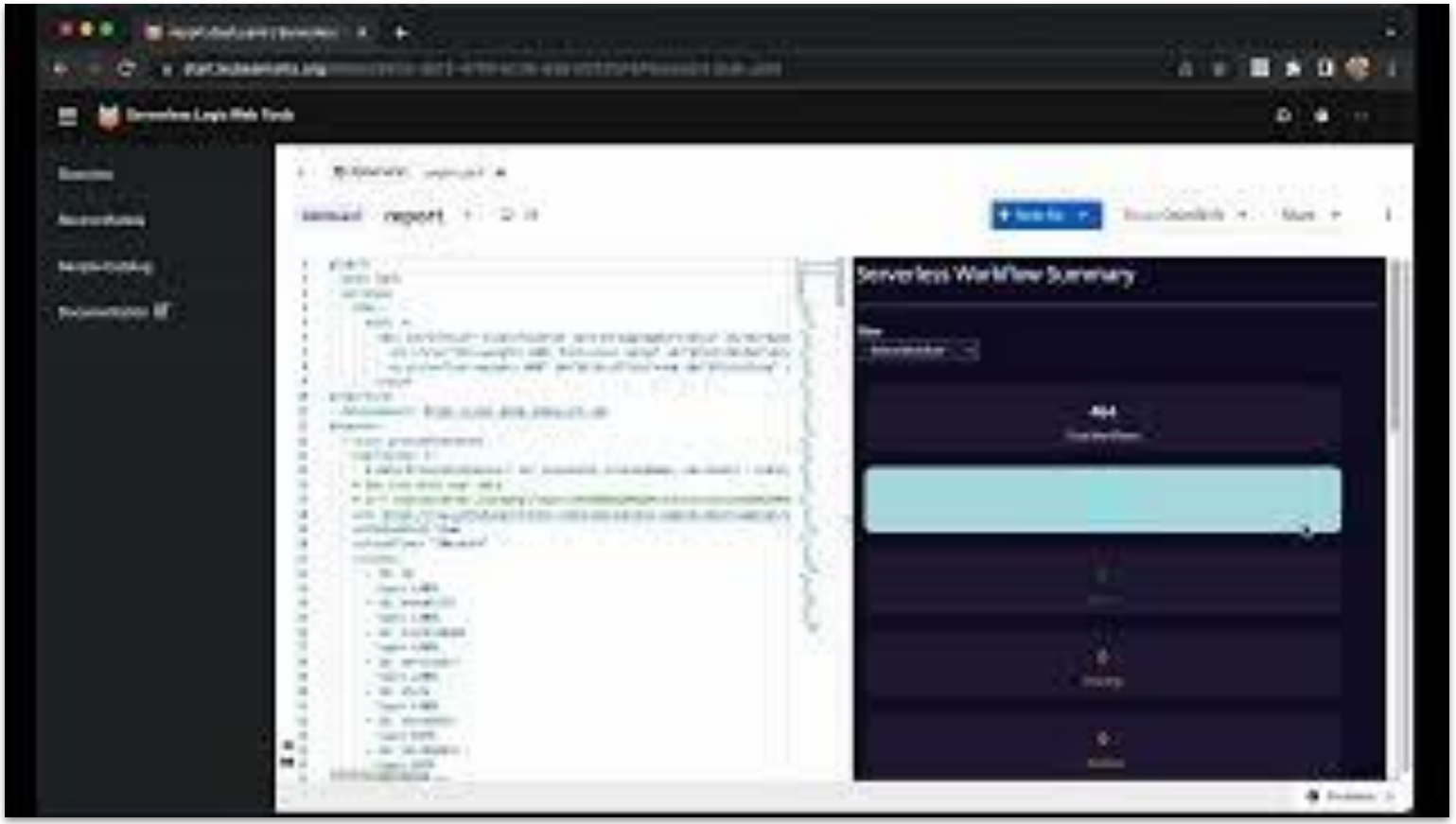
start.kubesmarts.org

sandbox.kie.org

Red Hat

Some authoring highlights

Red Hat

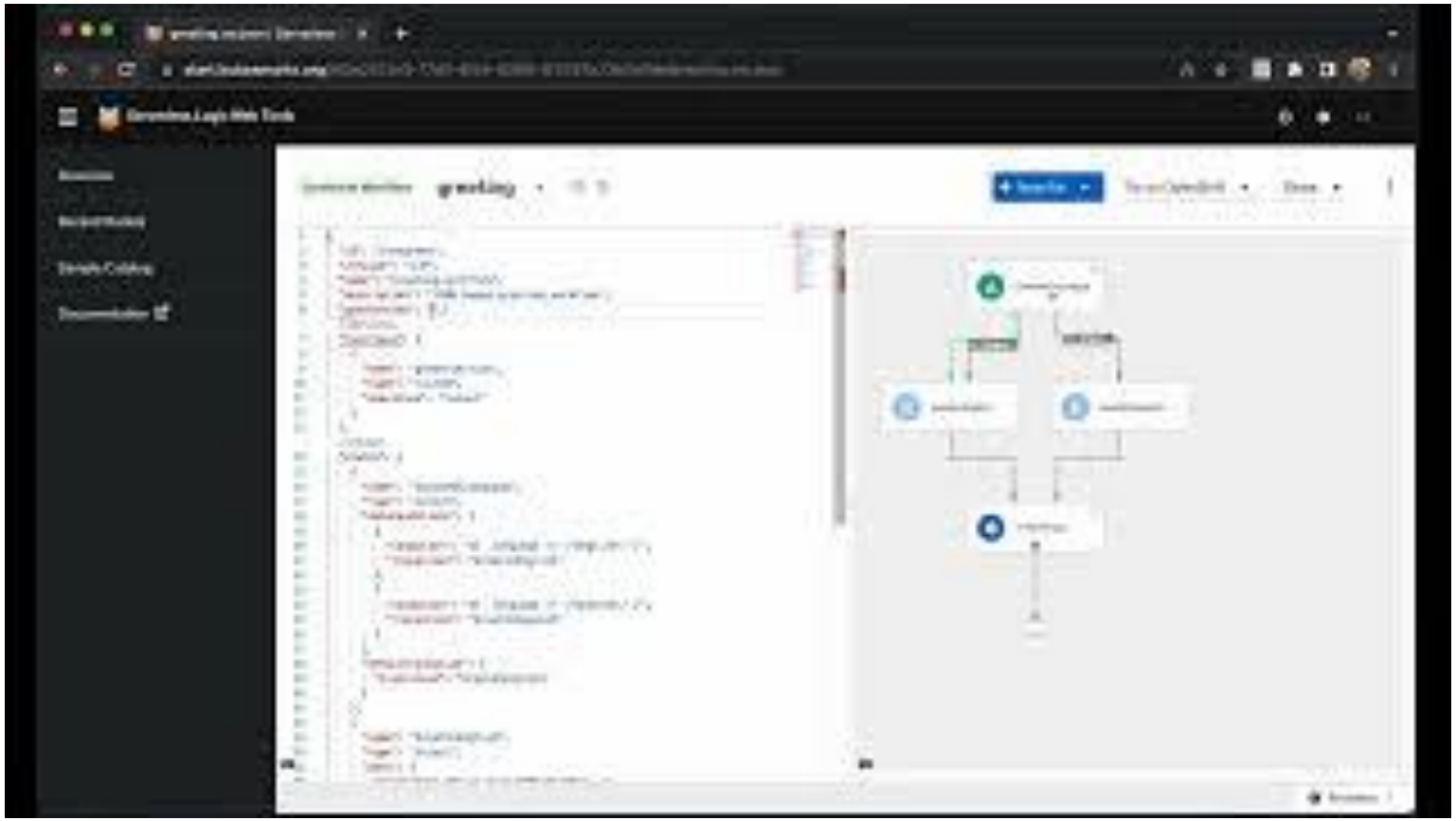


In-browser multi-file support

Users are able to upload, create and manage their files inside the browser.

This is an implementation for file system in the browser specific for the online channel.

Red Hat

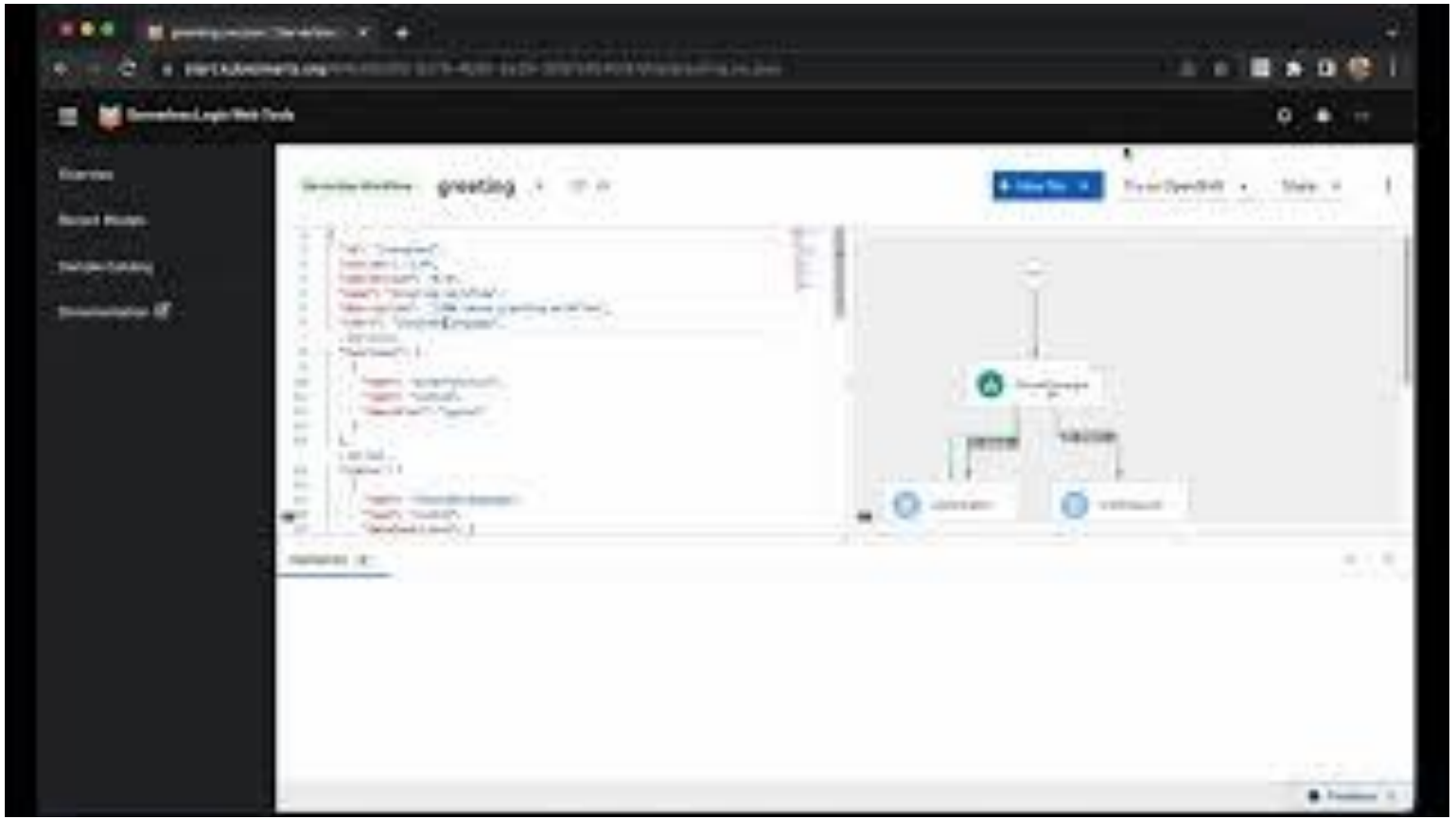


Autocomplete

Users can benefit from autocomplete for values, structures and even entire code snippets when using text editors.

Channels can use the same language service package to provide this feature.

Red Hat

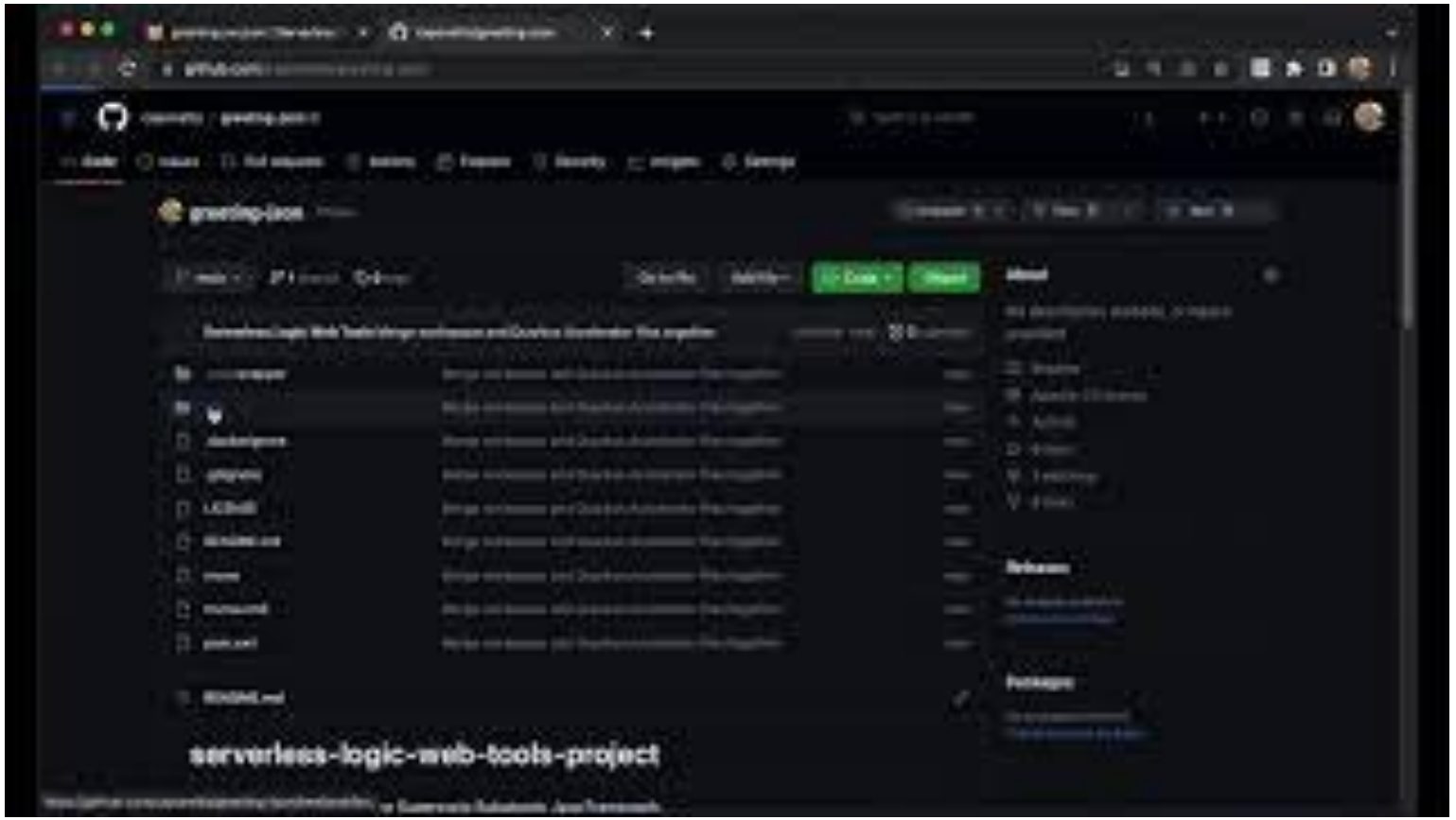


Validation

Users can be aided by real-time validation of their edits.

Each channel implements their own way of showing this information.

Red Hat

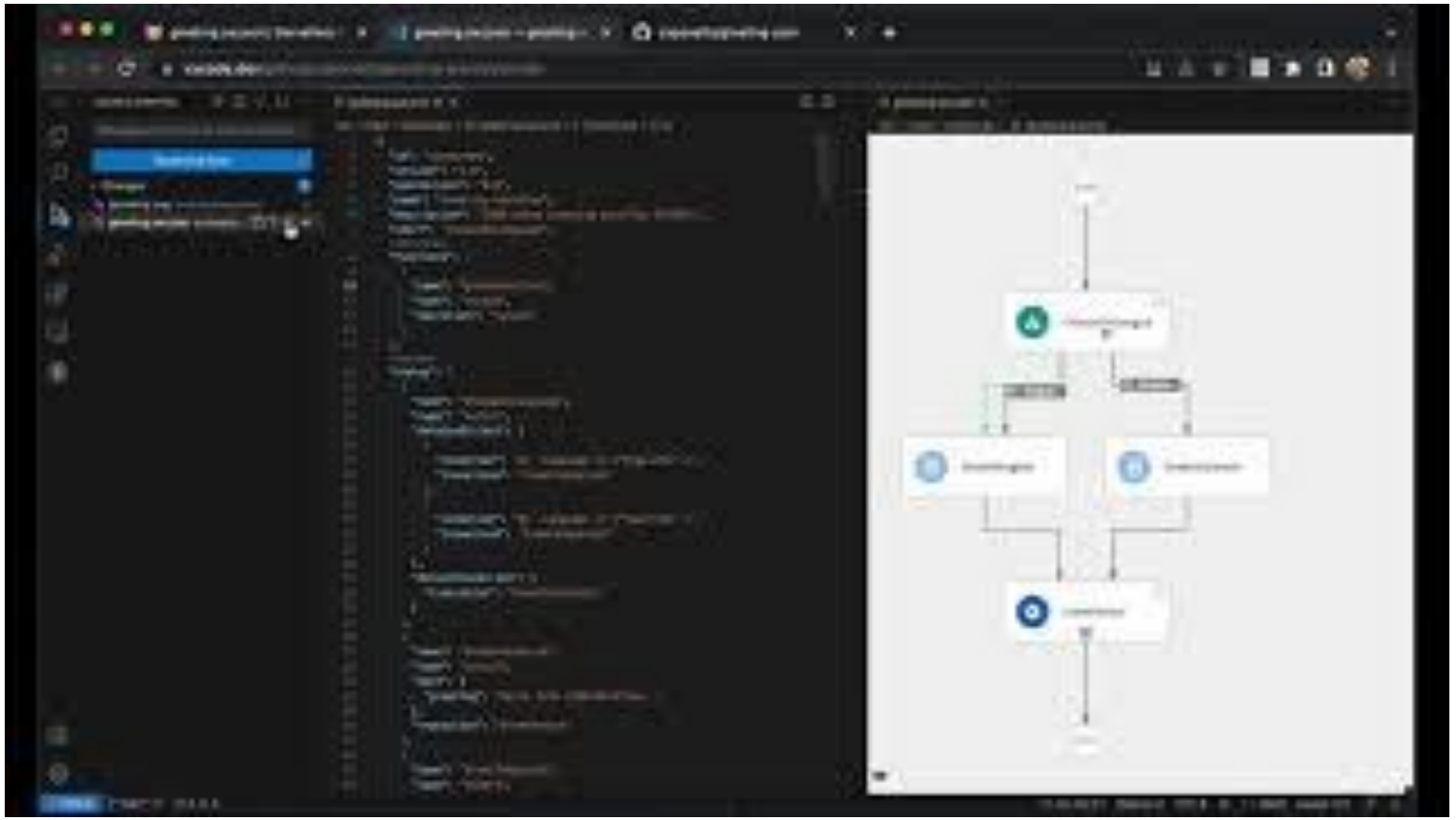


GitHub integration

Users can easily import their GitHub repositories and start working on the files.

As code can be pushed/pulled to/from the repository, it enables the online channel as another medium for collaboration.

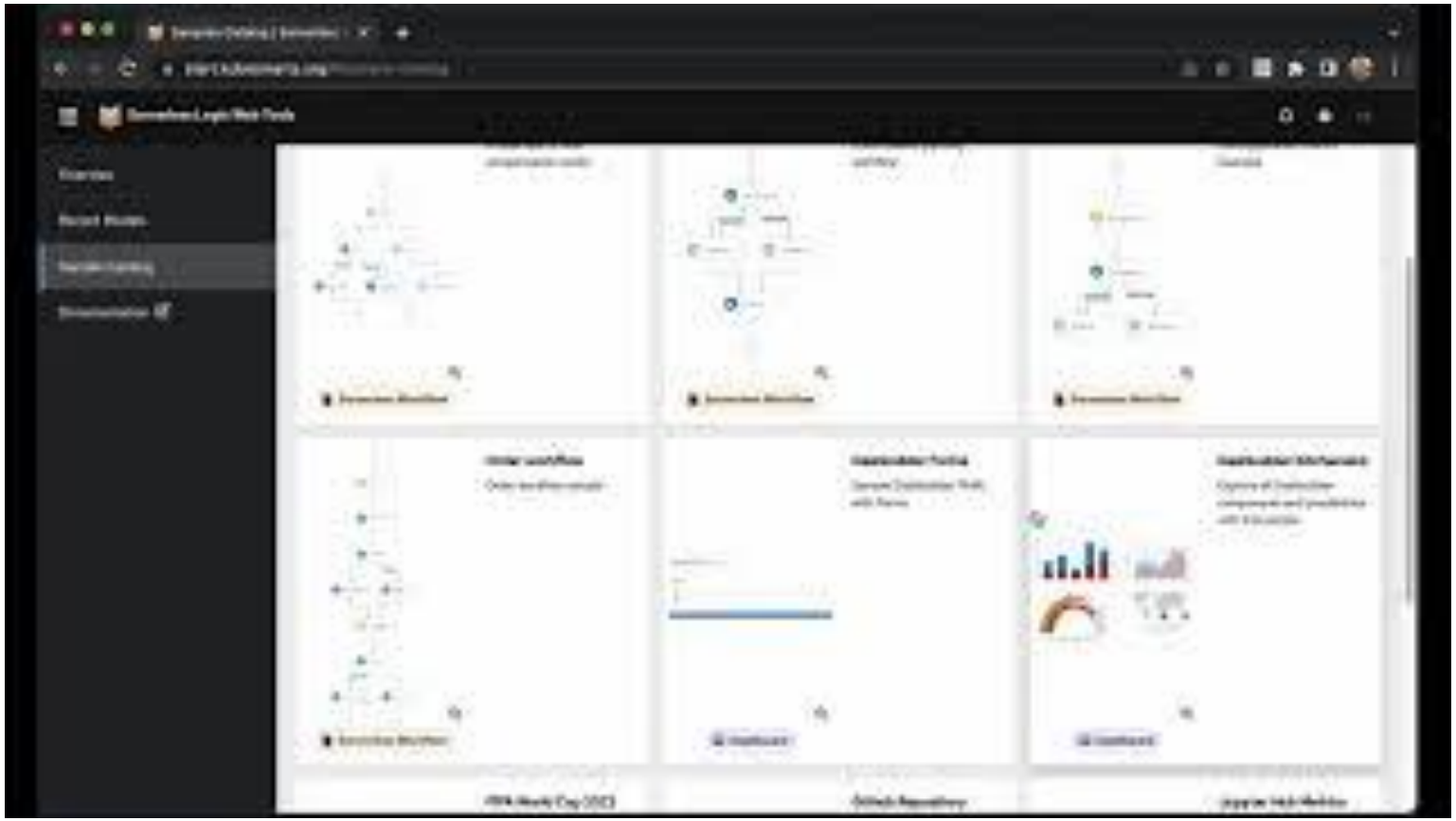
Red Hat



VS Code integration

Users can easily shift to either VS Code or VS Code web from the online channel.

Red Hat

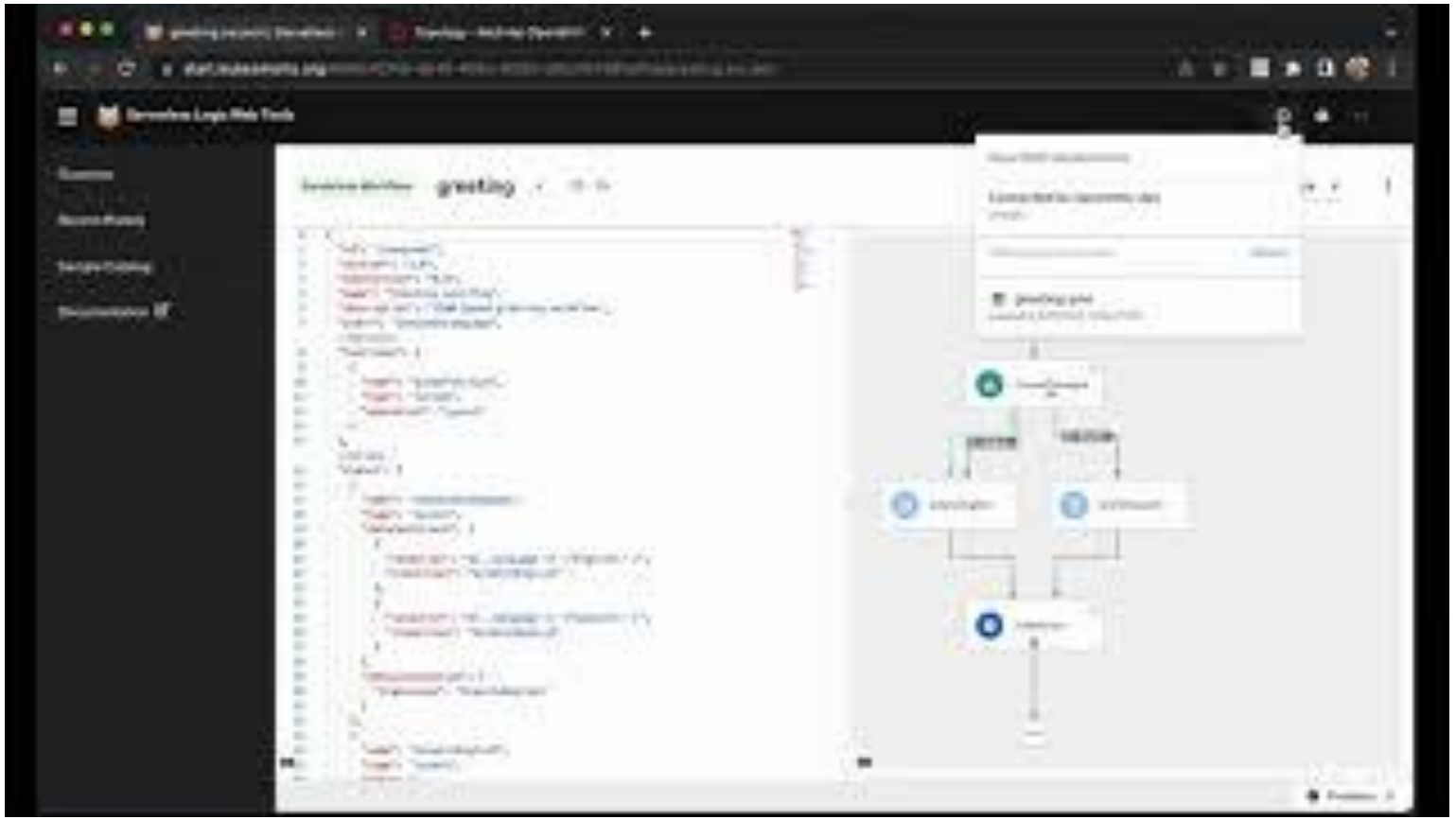


Samples

Users can try out various samples when starting to explore the editors.

Since the samples live in their own repository, users can also share their own samples and make them available to everyone.

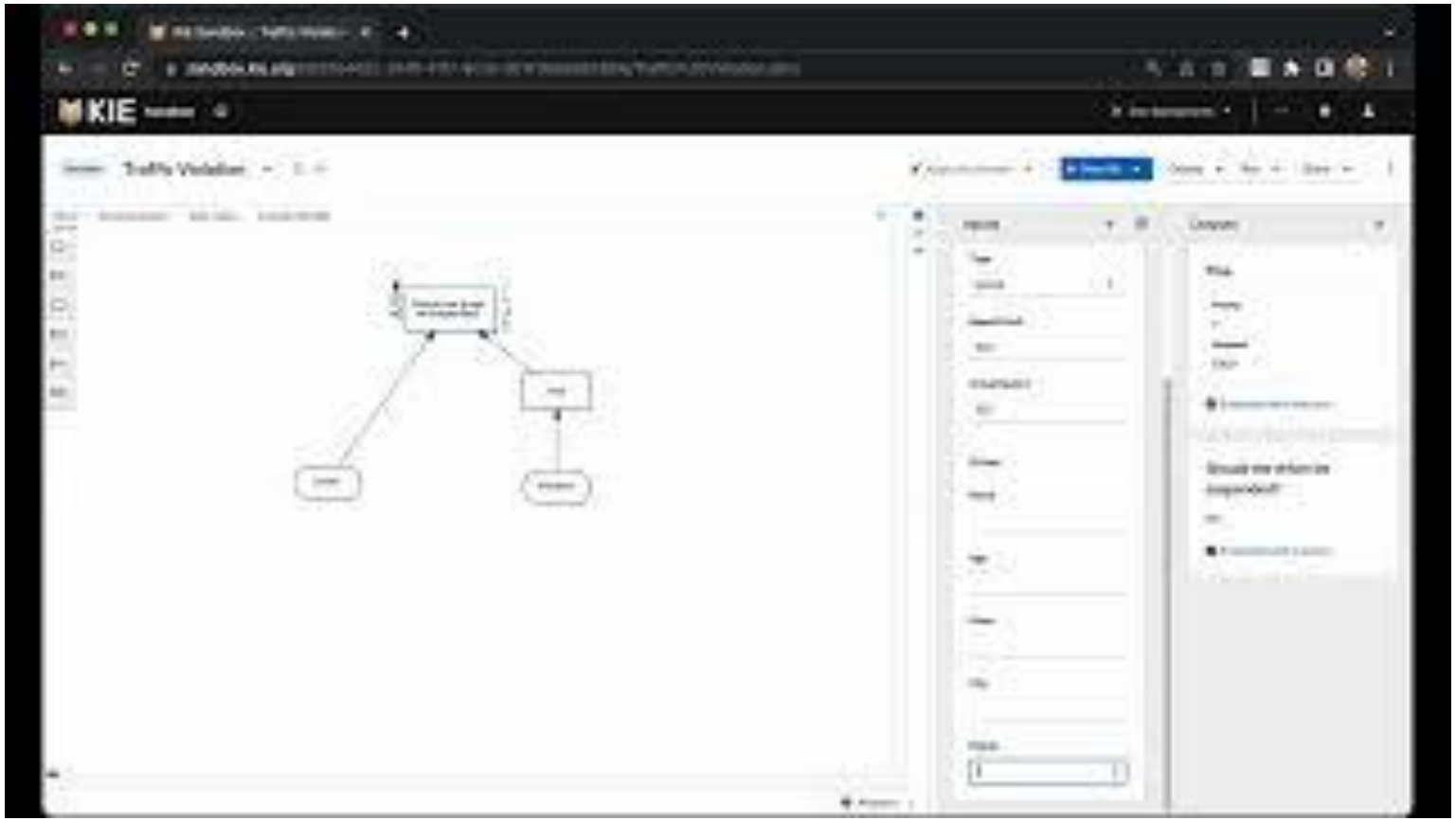
Red Hat



Deploy to OpenShift

Users can deploy their models to their OpenShift instance and share them with others.

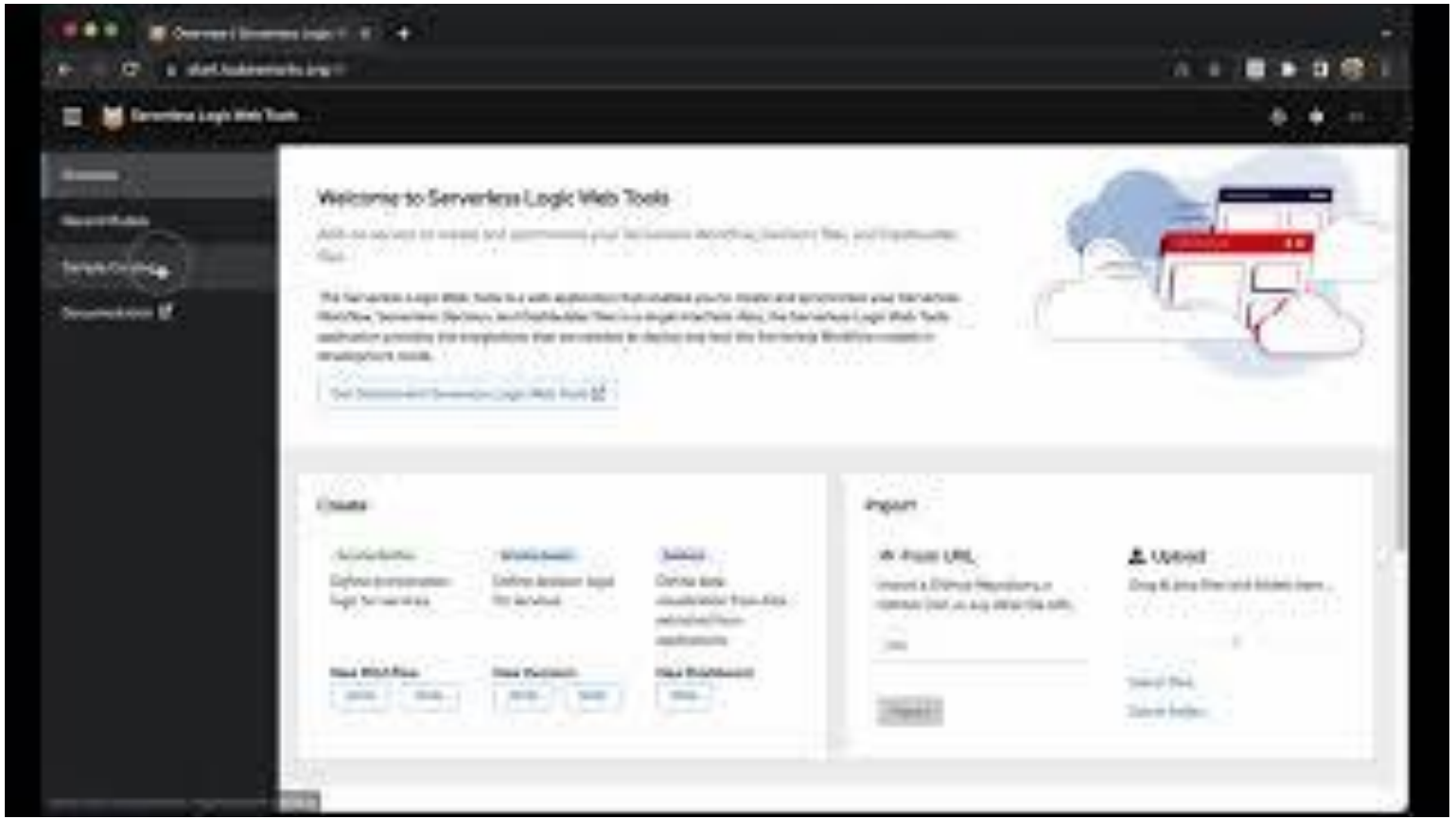
Red Hat



Backend services

Users can connect to backend services running either locally or remotely to augment their authoring experience.

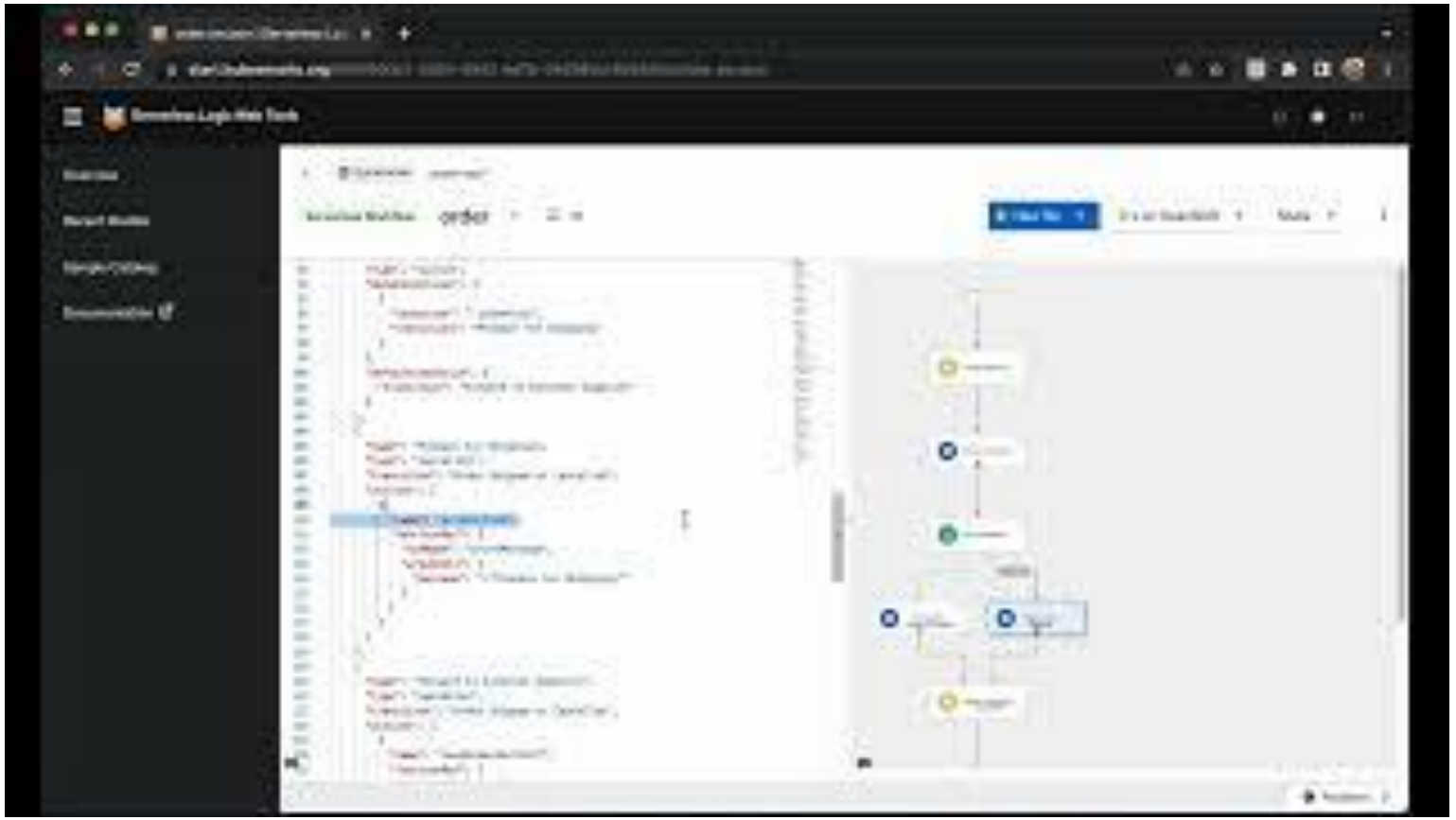
Red Hat



Dev Mode

Users can connect to their OpenShift instance and easily try out their changes in a Quarkus environment loaded up in dev mode.

Red Hat



Envelope communication

Users can benefit from inter-envelope communication to have a richer set of features.

Red Hat

Good frontend development is hard.

Red Hat

Your favourite web framework will not be here forever

Red Hat

Micro frontend Spectrum

Total independence ⟷ **Strategic collaboration**

Each team chooses tech stack

Each micro frontend makes it's own API calls

App is composed of fully functional micro apps

Each micro frontend has it's own CI/CD

Agrees on tech stack

Container handles all API calls

Share 'dumb' components

Shared CI/CD

https://twitter.com/housecor/status/1139504822930092033/photo/1

Thank you

Eder Ignatowicz

@ederign

Guilherme Caponetto

@caponetto

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat