Lessons learned after 18 months of Micro Frontends adoption in a large scale application

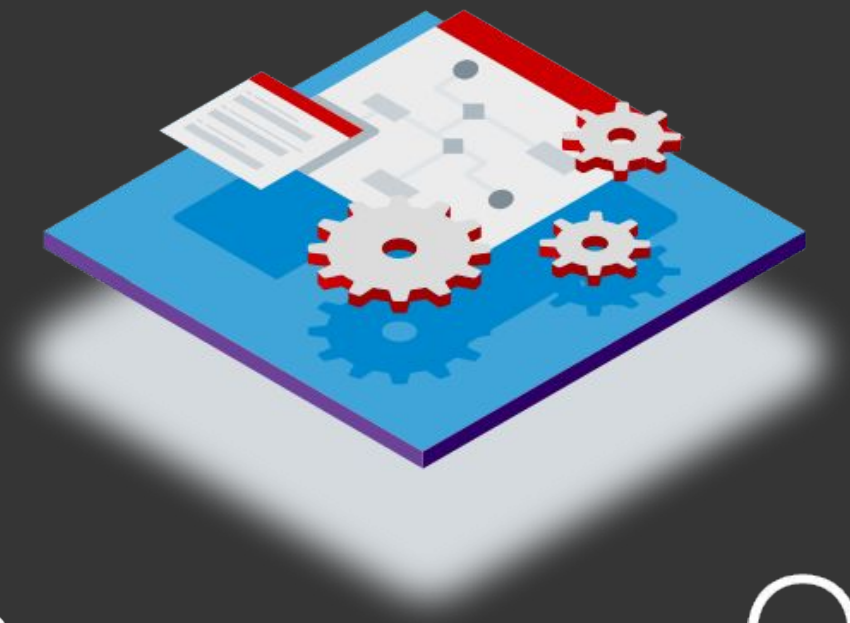
Eder Ignatowicz
Principal Software Engineer
@ederign

Alex Porcelli
Senior Principal Software Engineer
@porcelli

Red Hat

KIE

jBPM

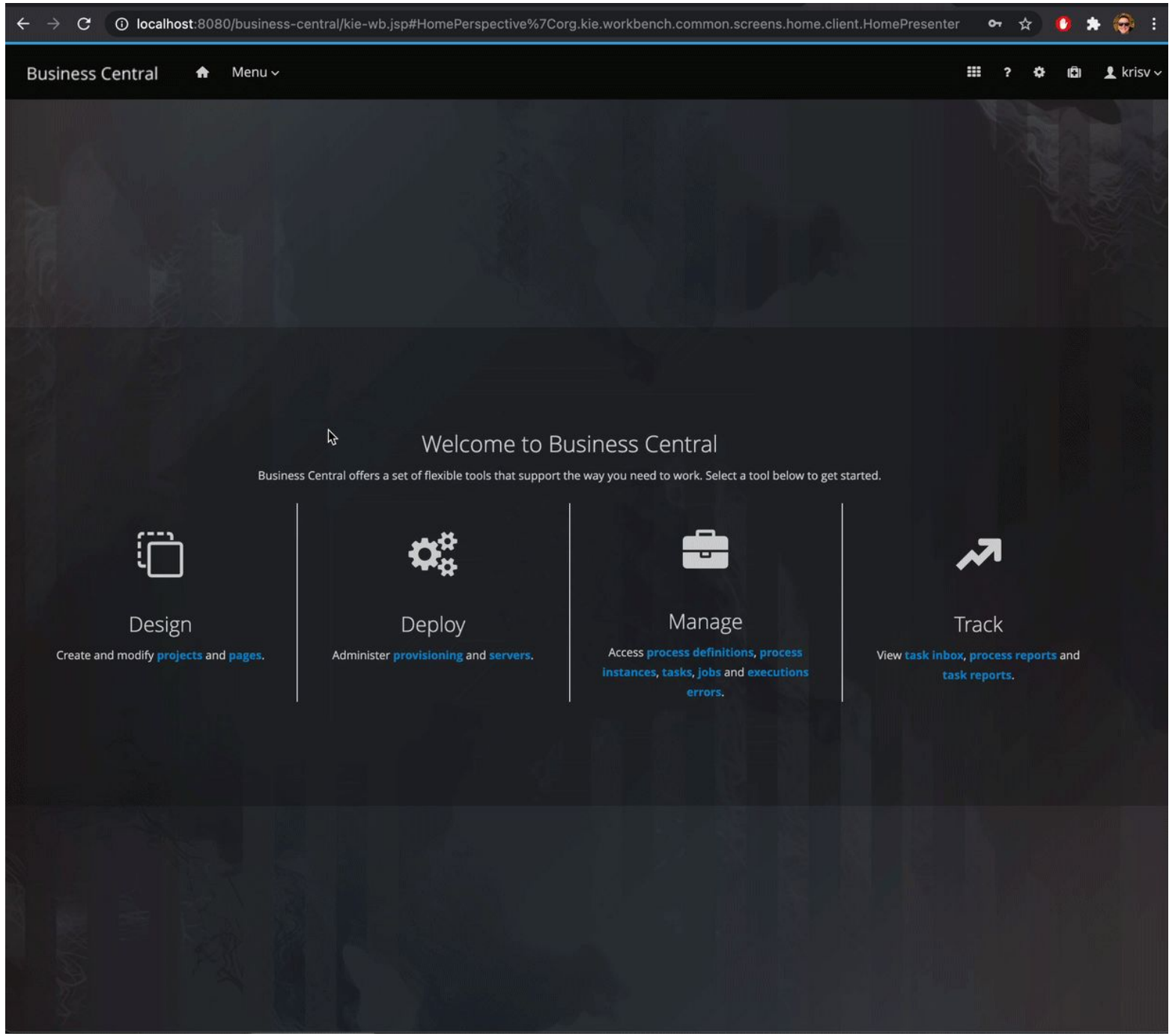


Drools

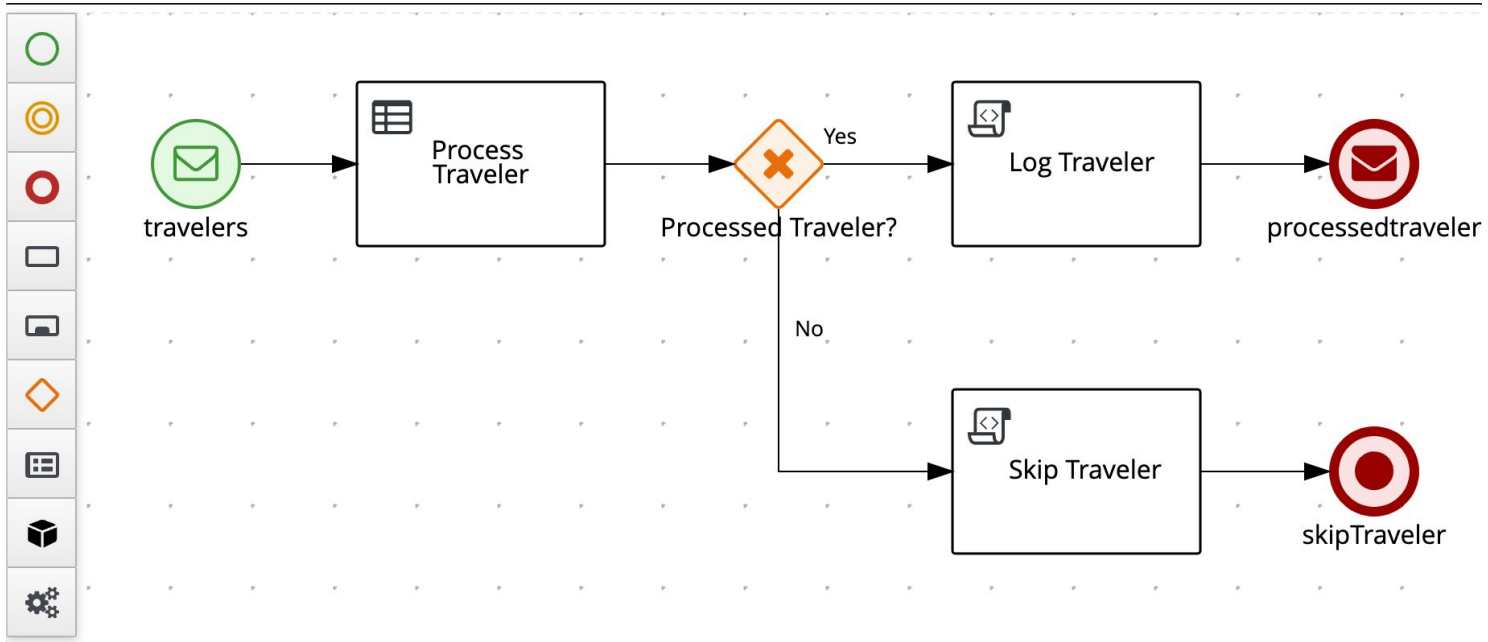
Kogito

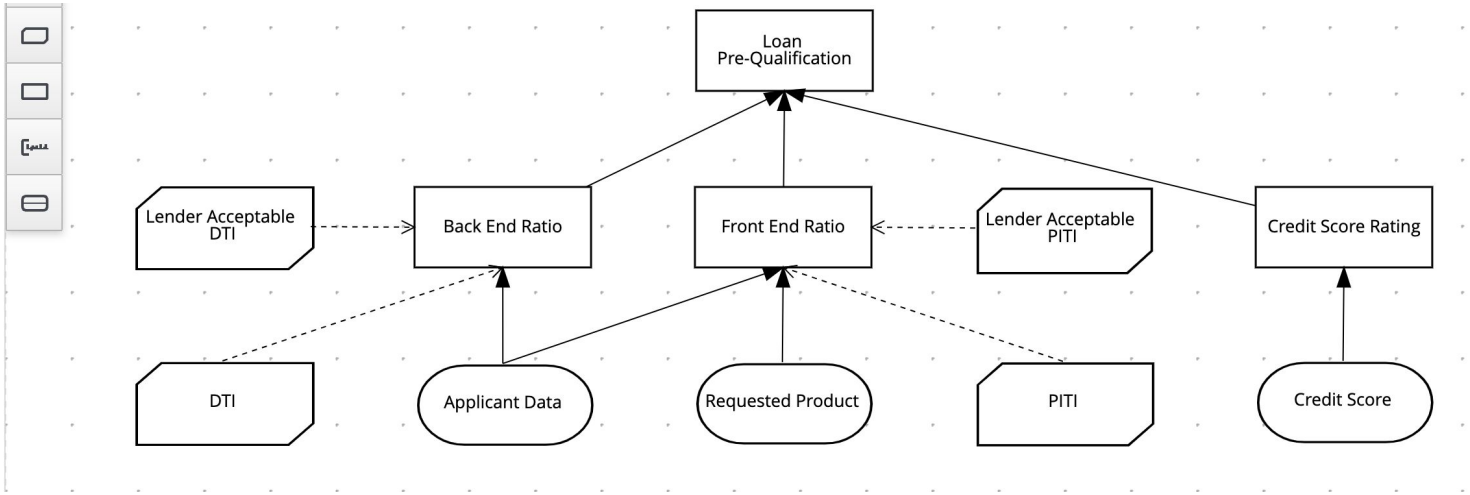
OptaPlanner

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.

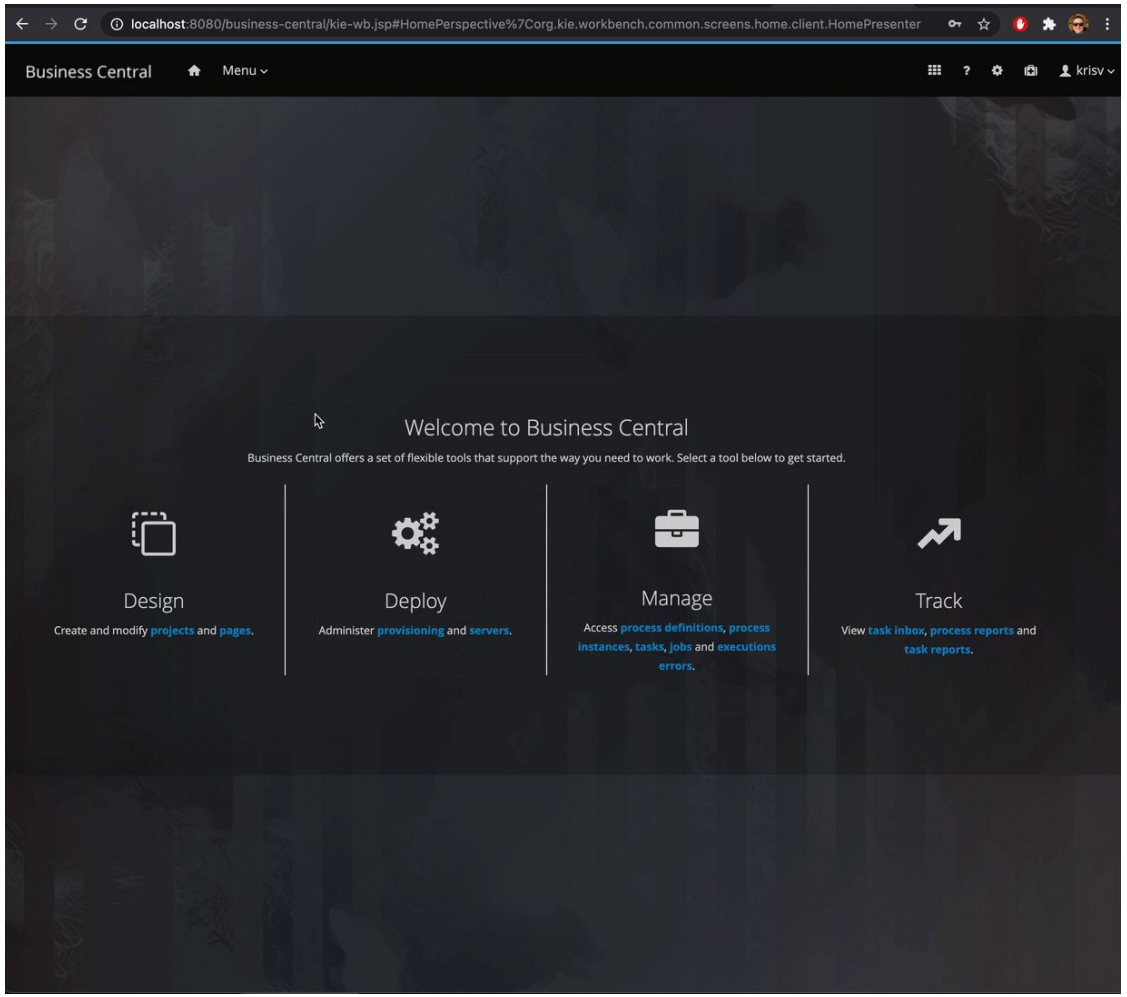
Red Hat


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

Red Hat

Red Hat



THE DEVELOPER'S CONFERENCE

QCon SÃO PAULO

JavaOne™

Red Hat

Kogito ergo automate

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

Red Hat

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?

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

Incremental upgrades

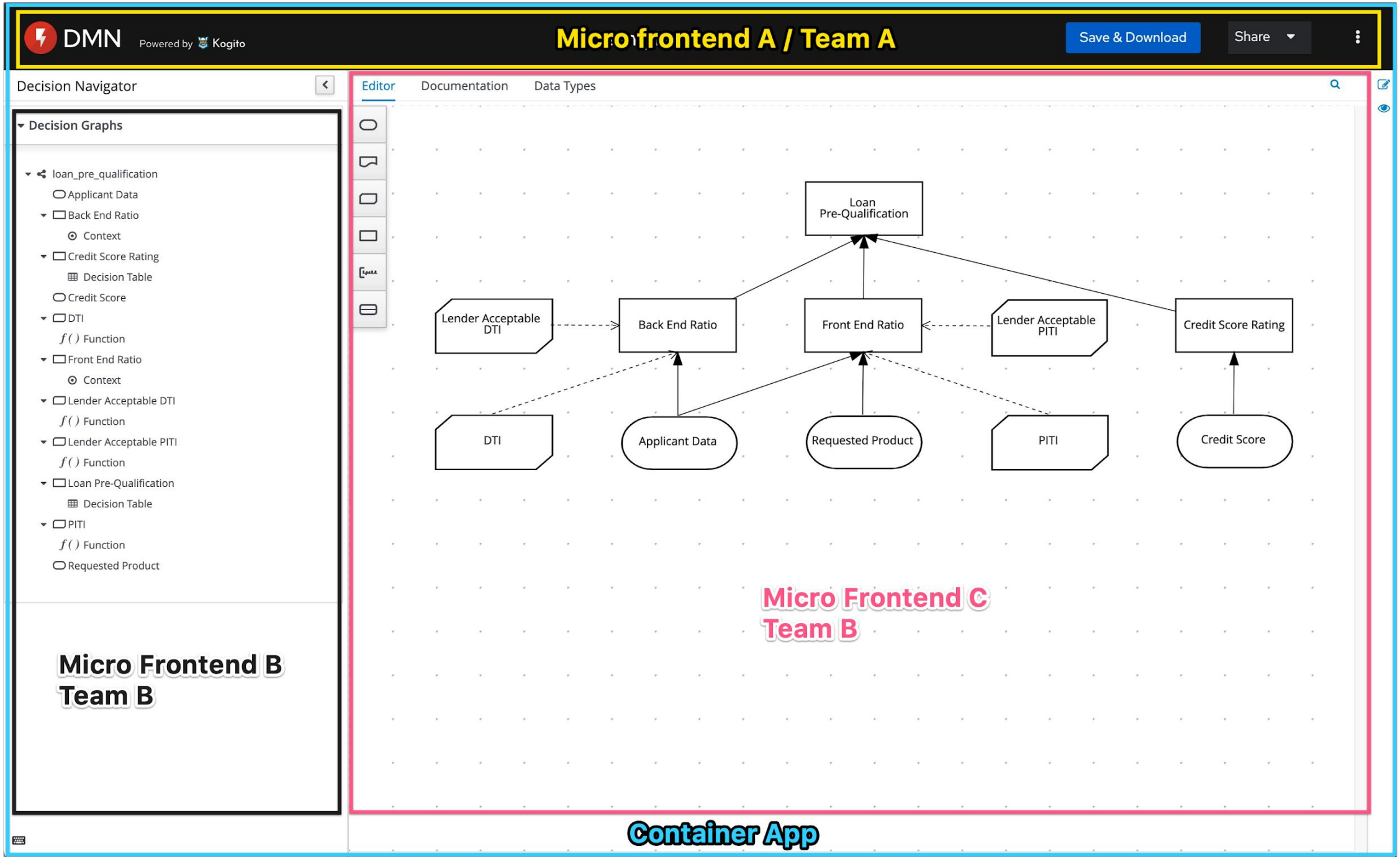
Simple, decoupled codebases

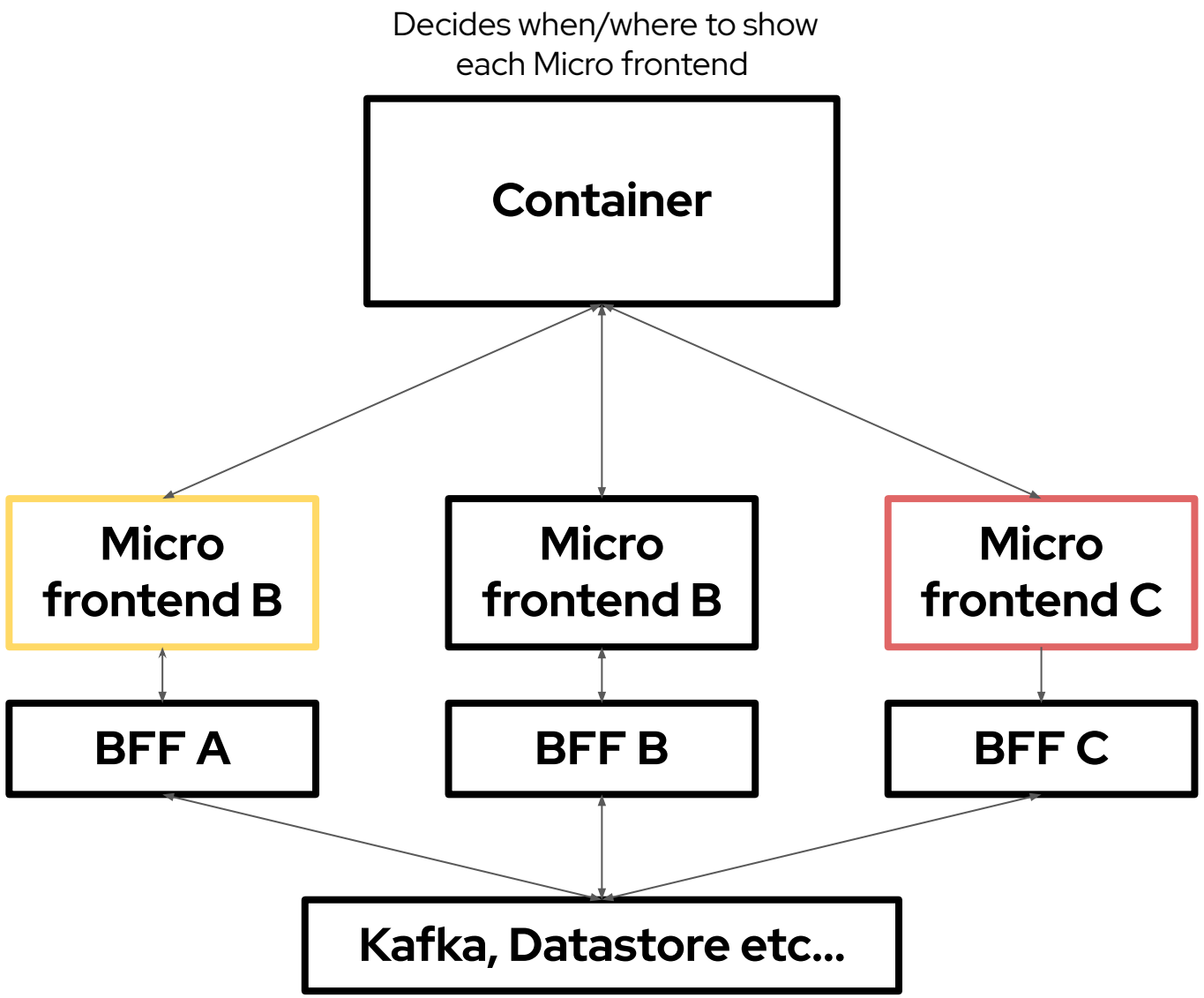
Each micro frontend can run as standalone

Independent deployment and releases

Autonomous Teams

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



No Micro frontend
communicate
directly to each other

Red Hat

Types of Integration

Run-Time integration

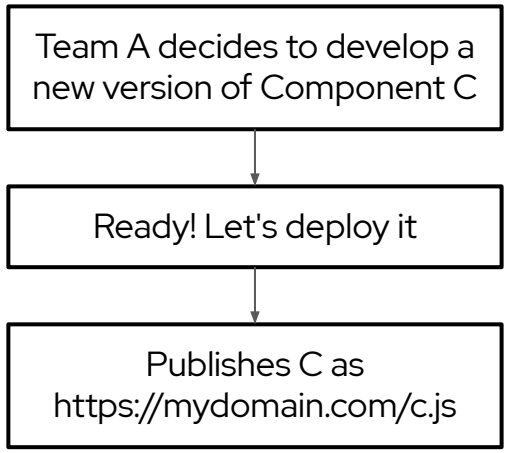
aka client-side integration:

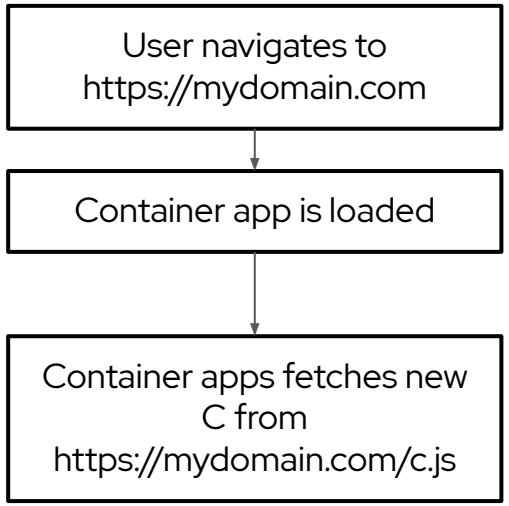
After the container gets loaded in the browser, it gets access to micro front end source code

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

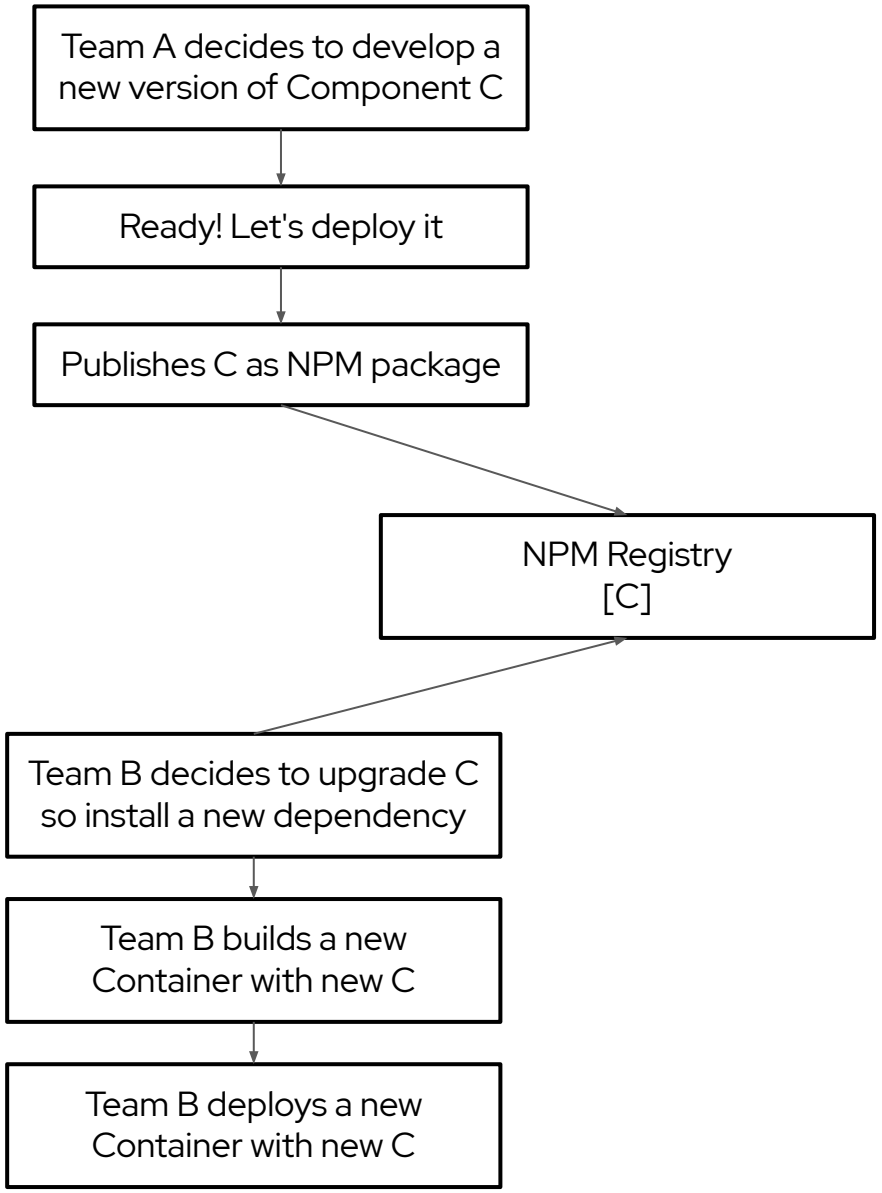
aka compile-time integration:

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

Foreign modules are accessible during build

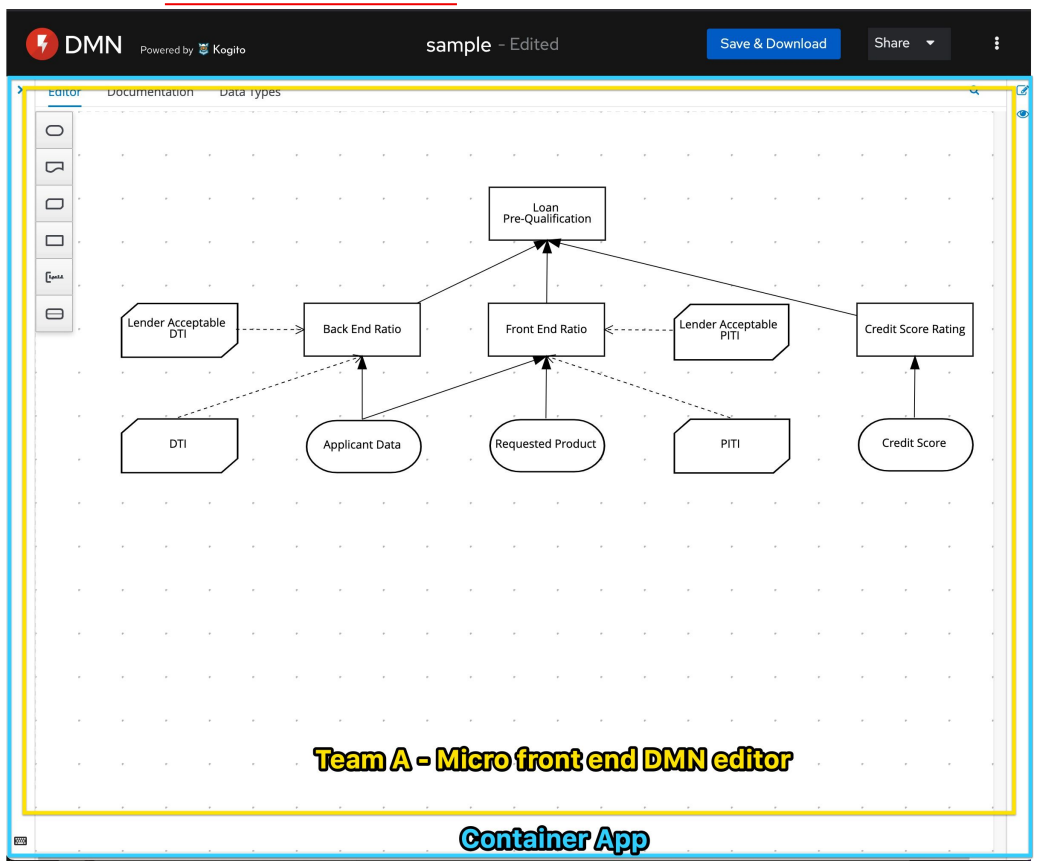
Pros: Easy to setup and understand

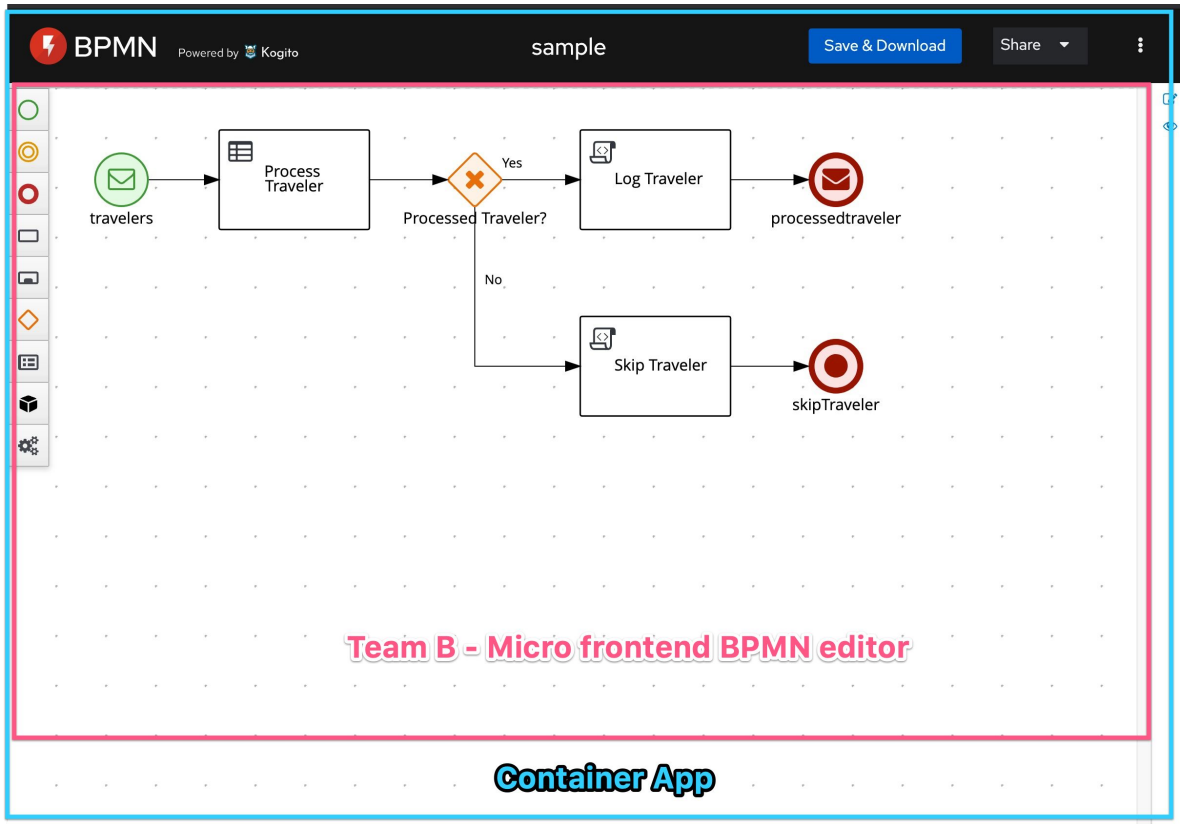
Cons: Container has to be re-deployed every time child has updated and tempting to tightly coupled Container + child together;



Red Hat

Build-time integration





Red Hat

Another concerns - Autonomous Teams

Autonomous teams

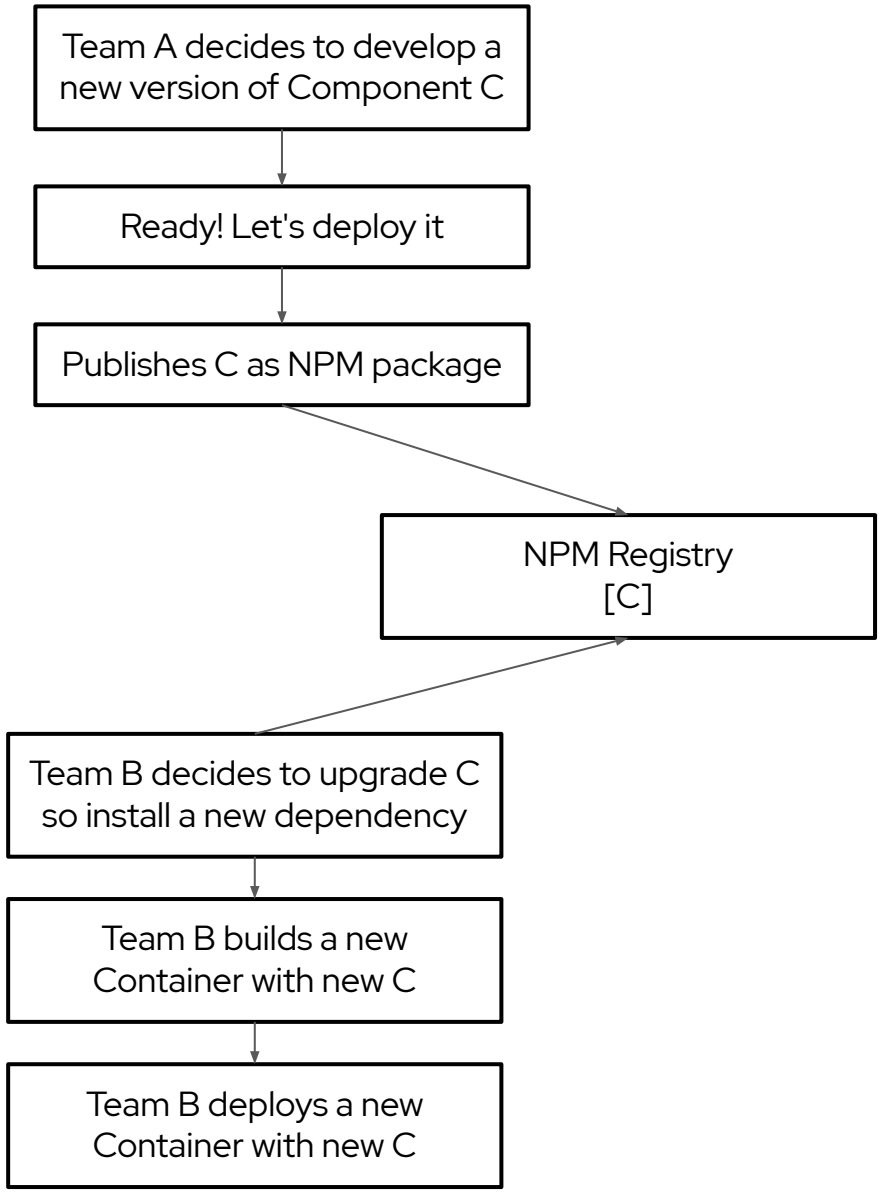
- Each team can run it's micro frontend in isolation

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

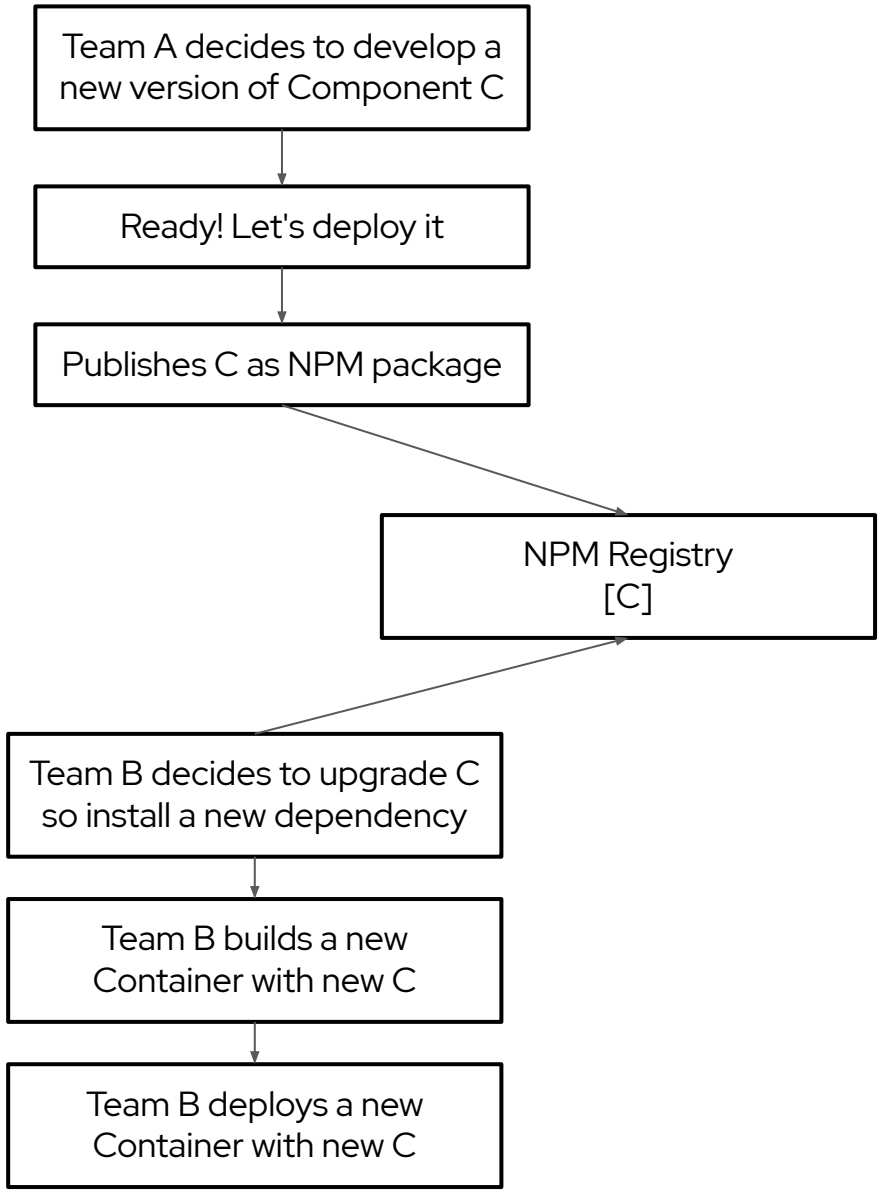


Red Hat

Another concerns - Styling

What you should do:

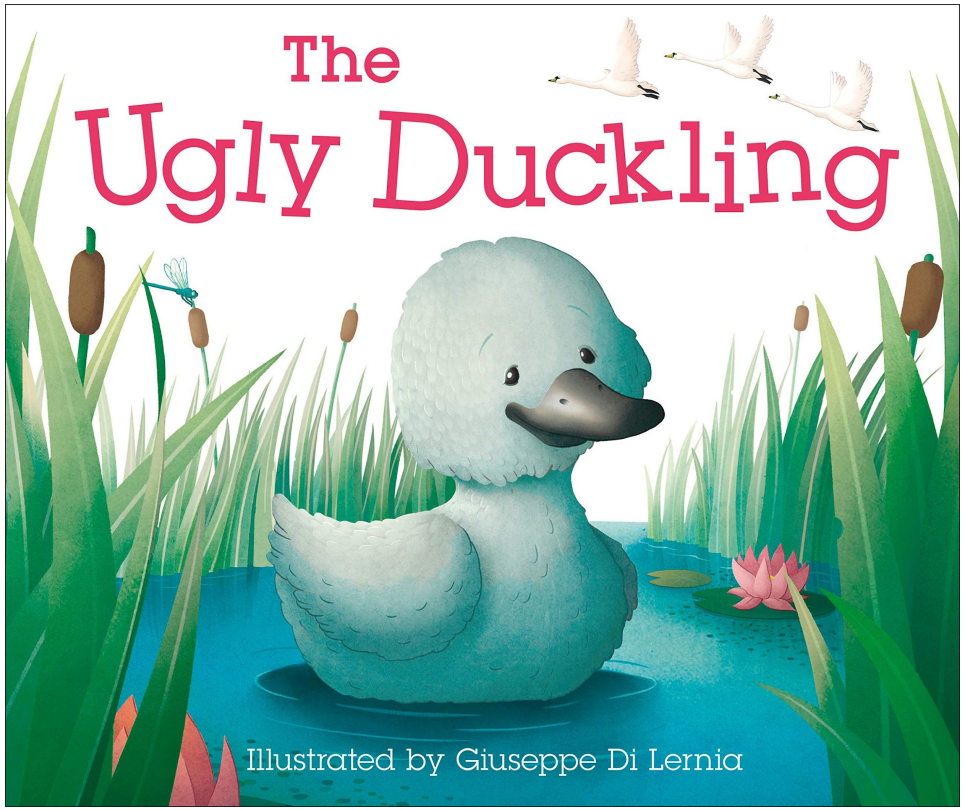
- *Custom CSS from your project:*
  - Use CSS-in-JS library
  - Use frameworks built-in component style scoping
    - Vue's and Angular has good ones
  - "Namespace" all your CSS
- *CSS coming from other libraries*
  - Use a component library that does css-in-js
  - Manually build the css library and 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

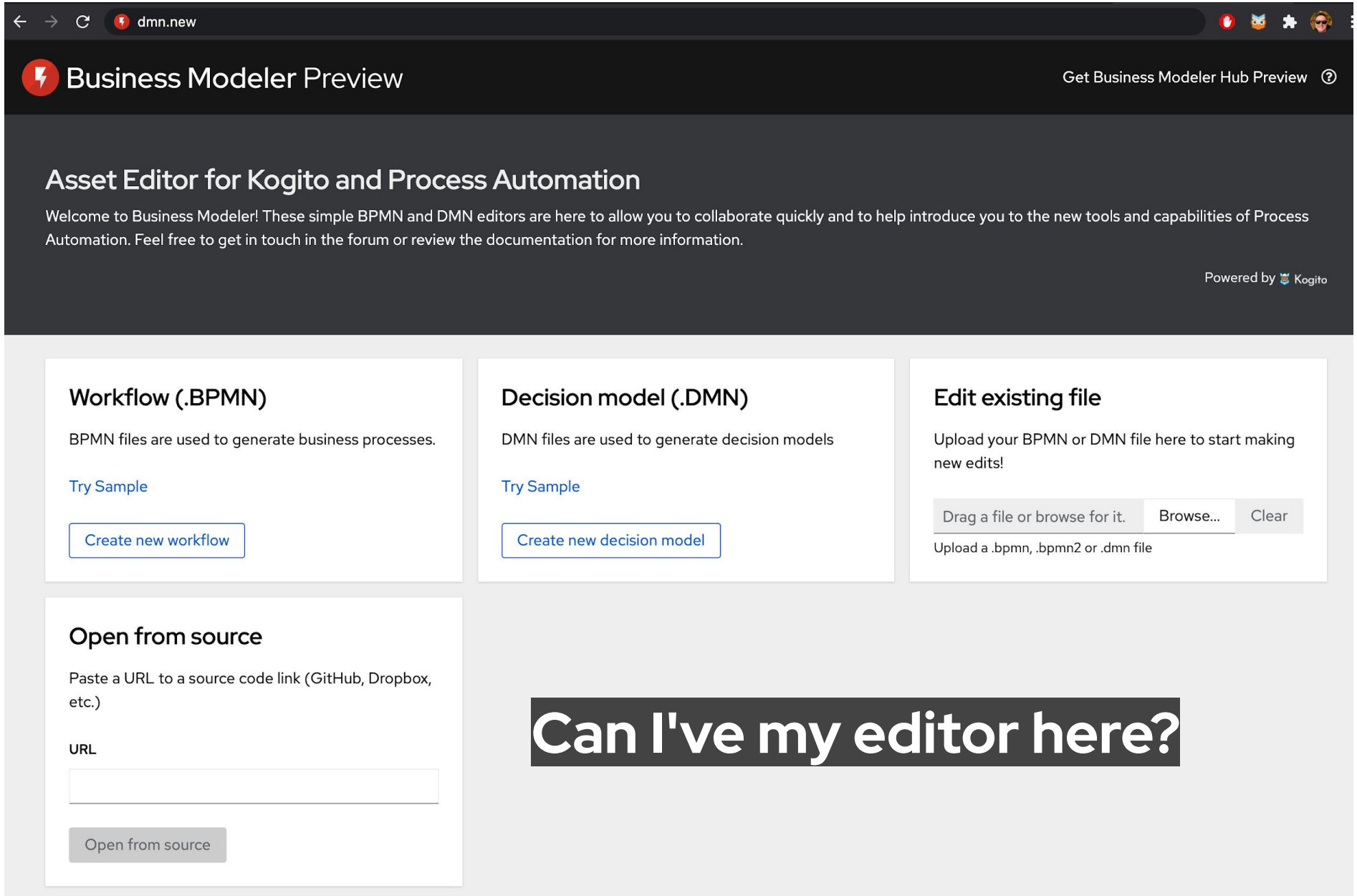
Debugging support for Trusted Type violations
Breakpoint on Trusted Type violations and link to more information in the Issues tab.

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

Kogito ergo automate

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

Red Hat



Can I've my editor here?

Red Hat

ederign.github.io

EXPLORER

OPEN EDITORS

EDERIGN.GITHUB.IO

- _includes
- _layouts
  - default-no-index.html
  - default.html
  - home.html
  - page.html
  - post.html
  - snippet.html
  - talks.html
  - wiki.html
- _lib
  - wiki_processor.rb
  - wiki.rb
- _posts
- _sass
- _site
- .sass-cache
- assets
- wiki
  - algorithms
  - java-and-jvm
  - raw
  - soft-skills
  - system-design
  - web-technologies
    - federated-modules.md U
    - index.md
    - microfrontends-with-react... M
    - talk-microfront-ends.md U
- _config.yml M
- .gitignore
- 404.html
- about.md

OUTLINE

TIMELINE

Can I also have the editors here? :)

Show All Commands ⇧ ⌘ P

Go to File ⌘ P

Find in Files ⇧ ⌘ F

Start Debugging F5

Toggle Terminal ^ `

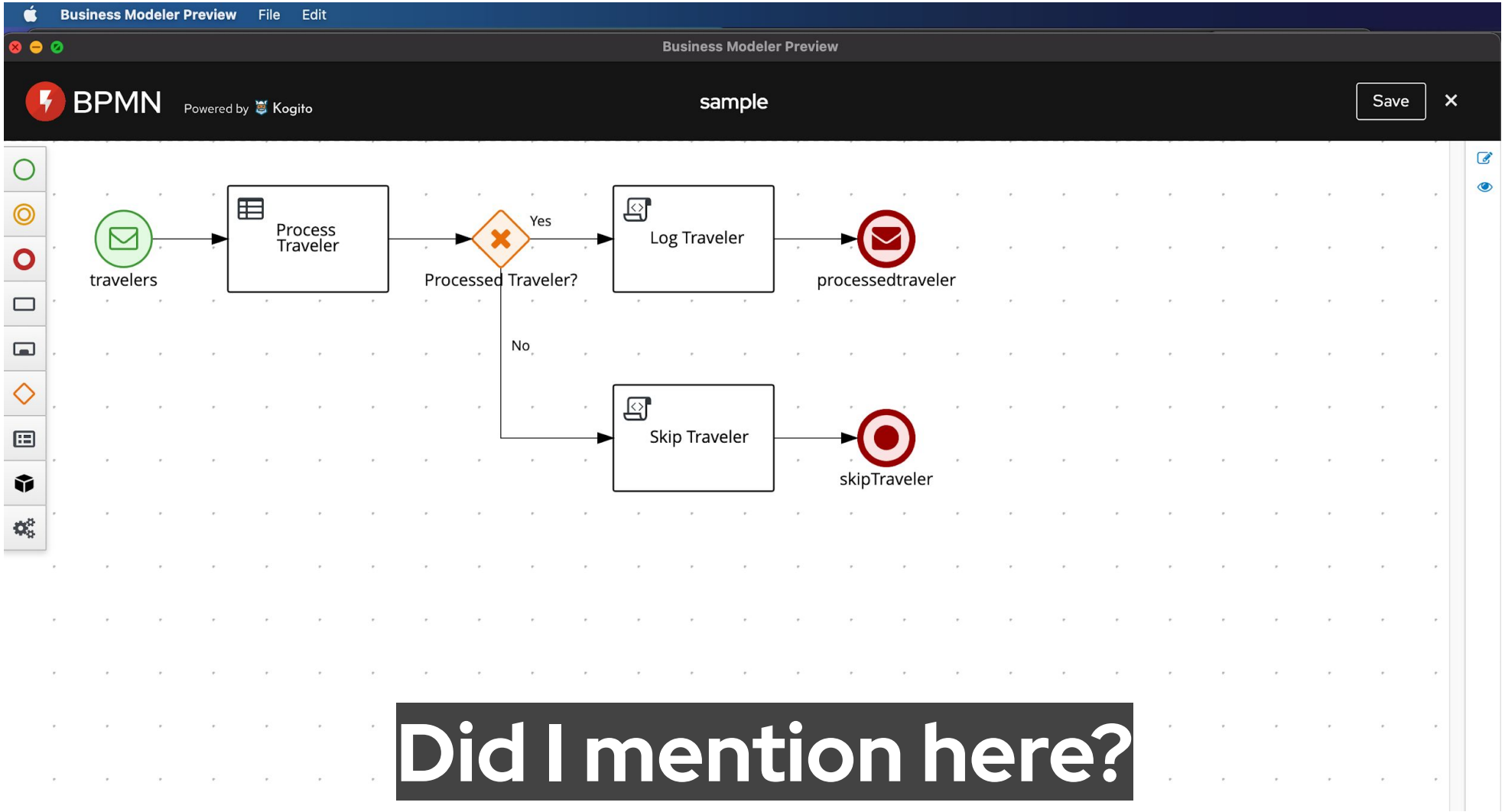
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: zsh

```
^C%
~/src/ederign.github.io master !4 ?6
~/src/ederign.github.io master !4 ?6
> 
```

2h 51m 46s

master* ⊗ 0 ⚠ 0 ESLint

Red Hat



Did I mention here?

Red Hat

github.com/ederign/KieDora/blob/master/persons.bpmn

Code | Issues | Pull requests 3 | Actions | Projects | Wiki | Security 2 | Insights | Settings

master | KieDora / persons.bpmn | Go to file

ederign Update persons.bpmn — Latest commit a2606cf on Mar 25, 2020 — History

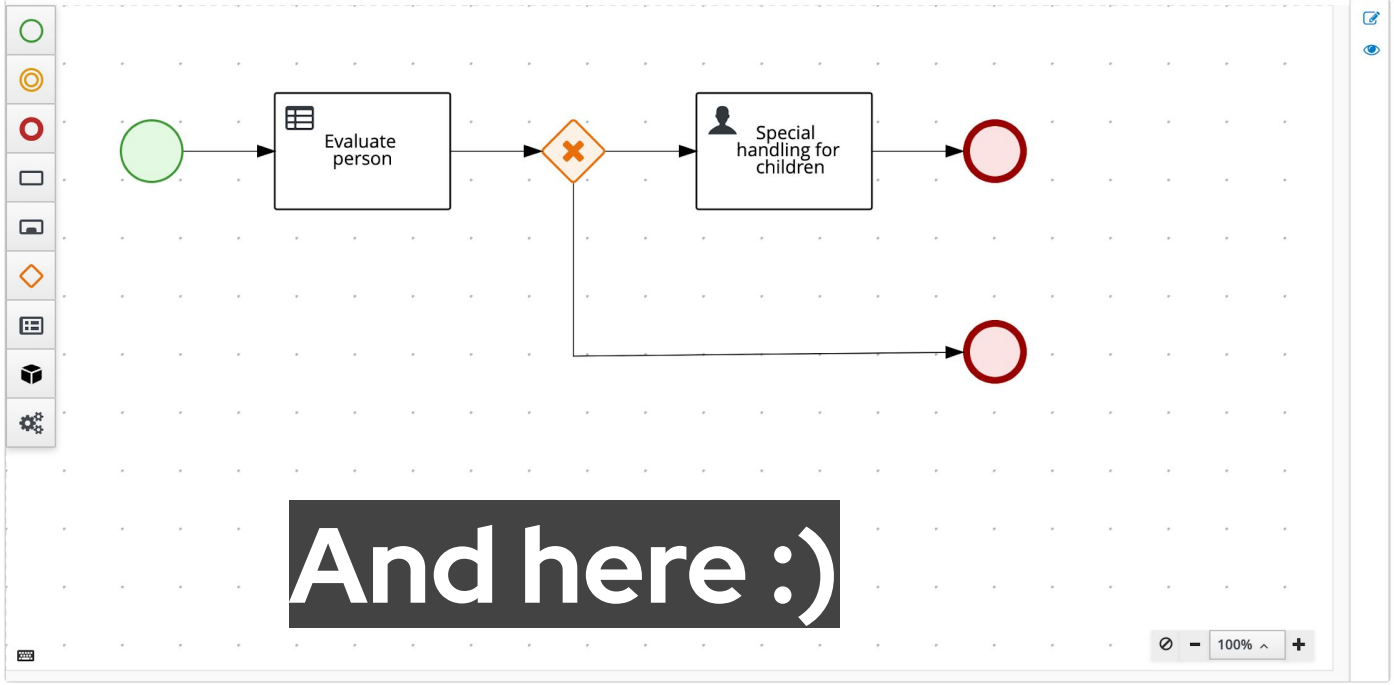
2 contributors

We found potential security vulnerabilities in your dependencies.
Only the owner of this repository can see this message.
See Dependabot alerts

This is a readonly visualization

See as source | Open in Online Editor | Copy link to Online Editor | Full Screen

240 lines (240 sloc) | 16.5 KB — Raw | Blame



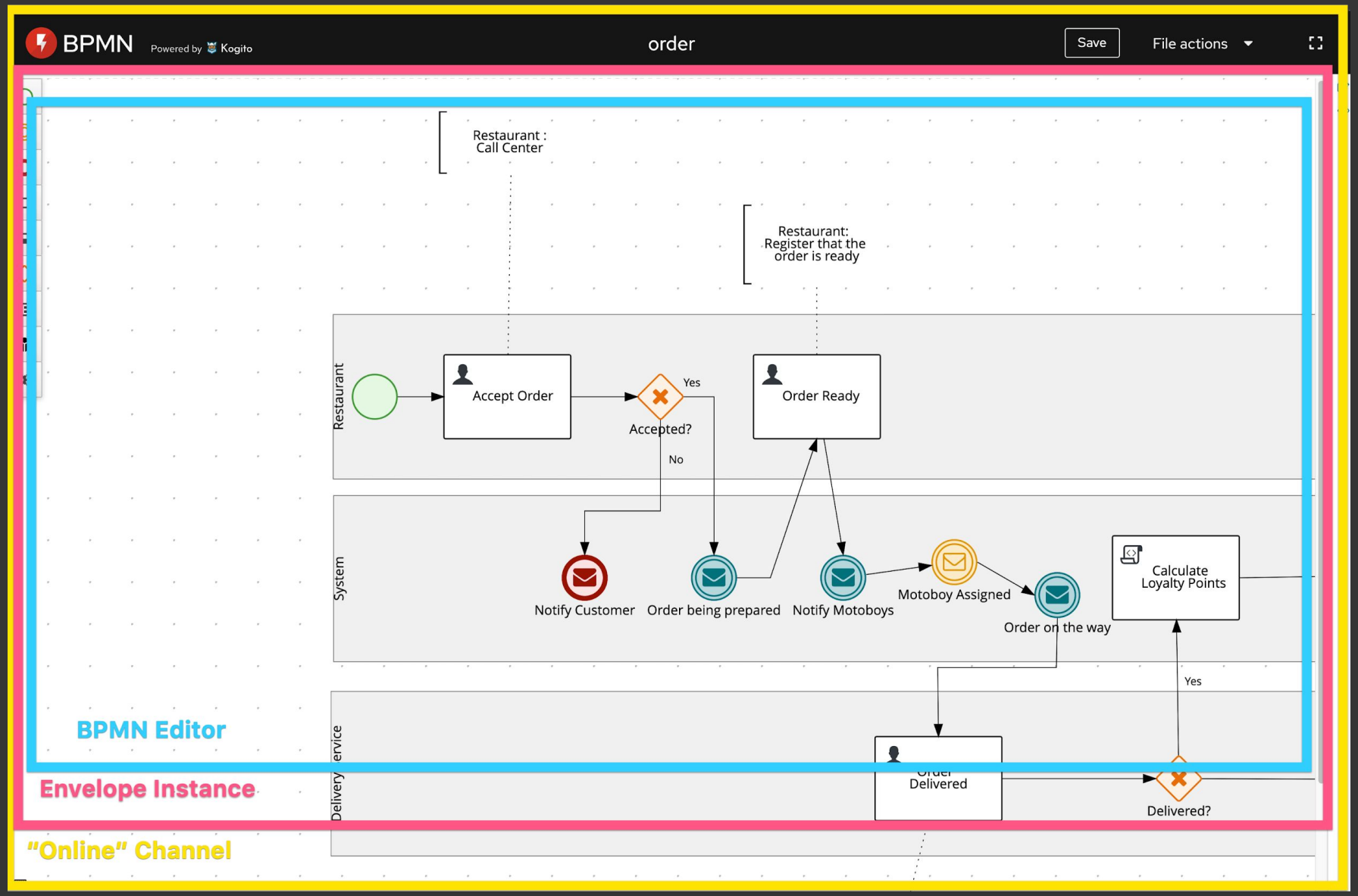
And here :)

Red Hat

Introducing Multiplying Architecture

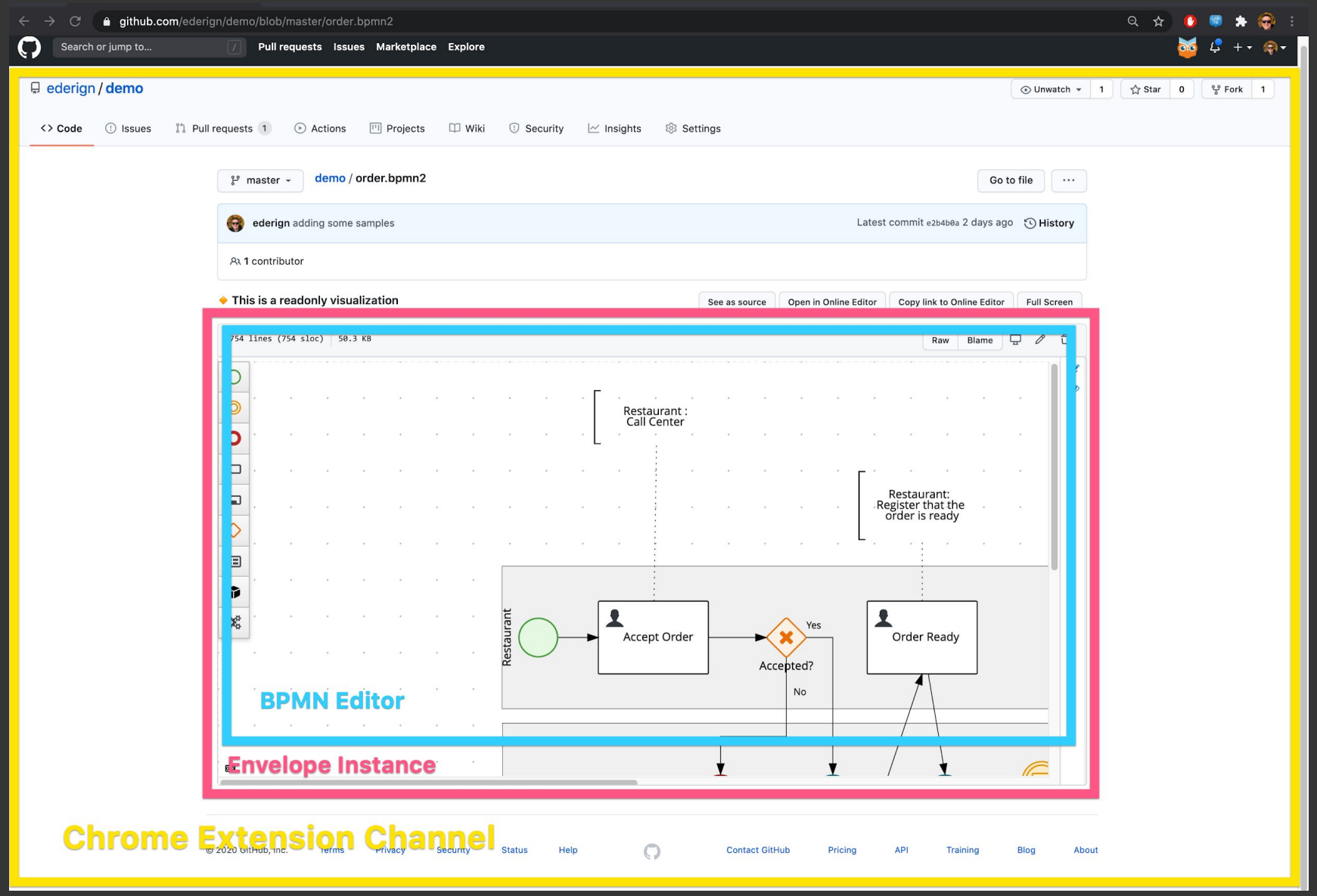
Red Hat

Online Channel



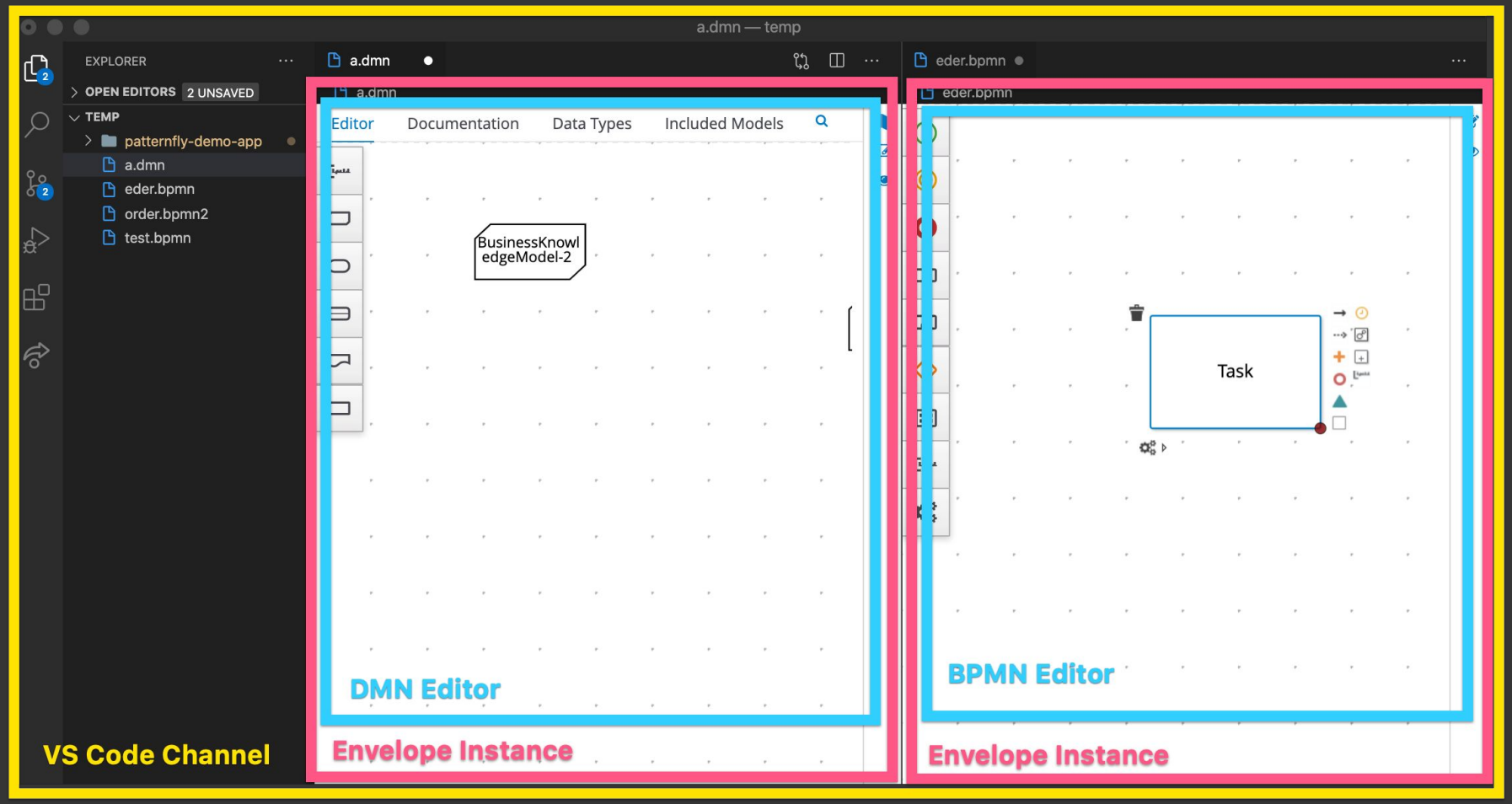
Red Hat

VSCode Channel

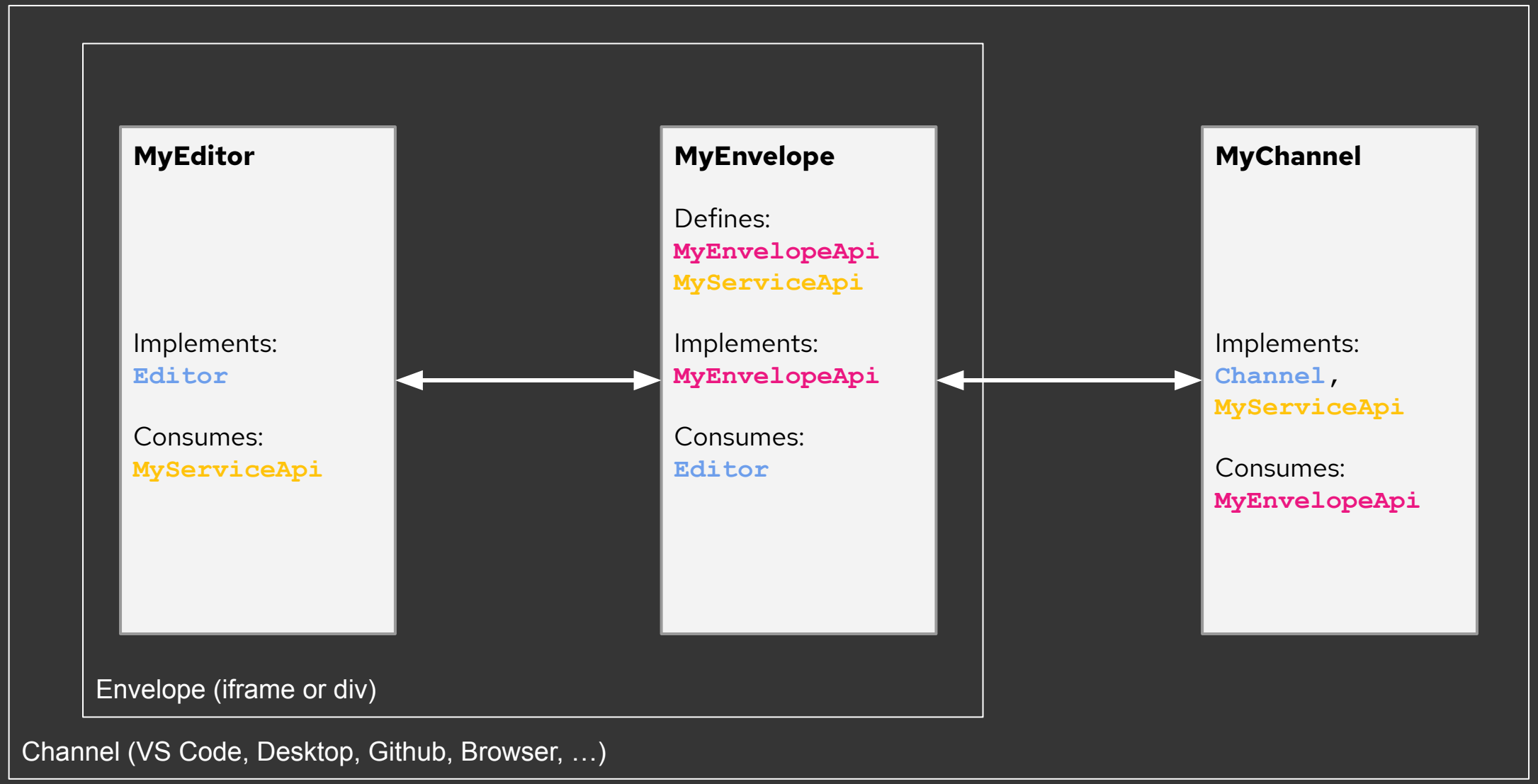


Red Hat

VSCode Channel



Red Hat


MyEditor
Implements:
Editor
Consumes:
MyServiceApi
MyEnvelope
Defines:
MyEnvelopeApi
MyServiceApi
Implements:
MyEnvelopeApi
Consumes:
Editor
MyChannel
Implements:
Channel,
MyServiceApi
Consumes:
MyEnvelopeApi
Envelope (iframe or div)
Channel (VS Code, Desktop, Github, Browser, …)

More on: Google for "Kie Live multiplying architecture"

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

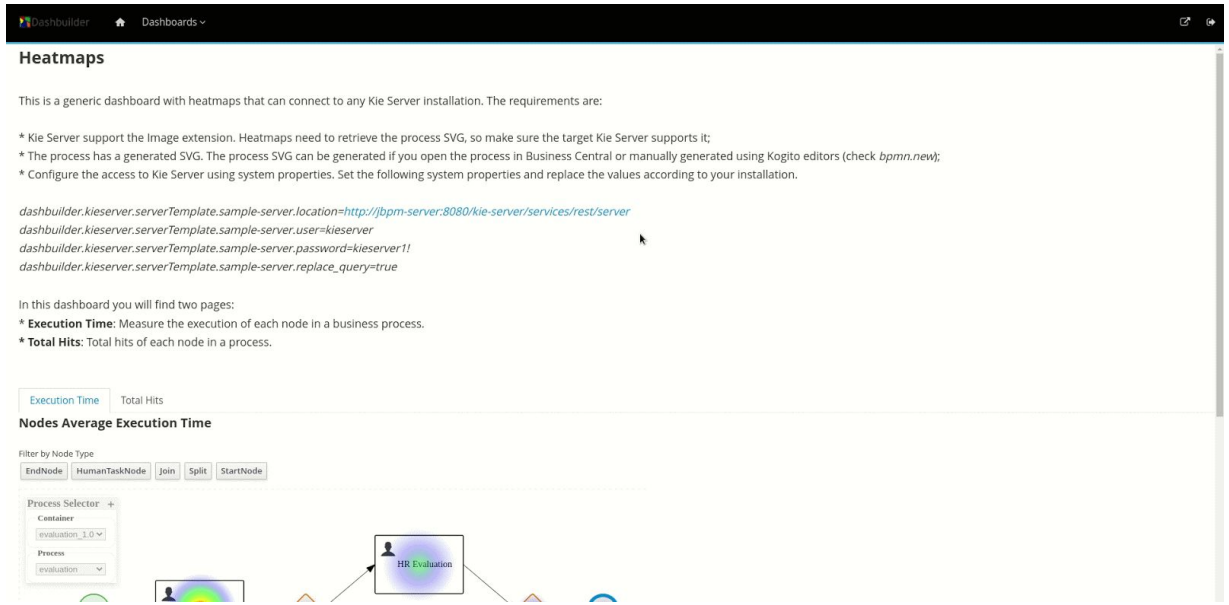
Where we are going?

Red Hat

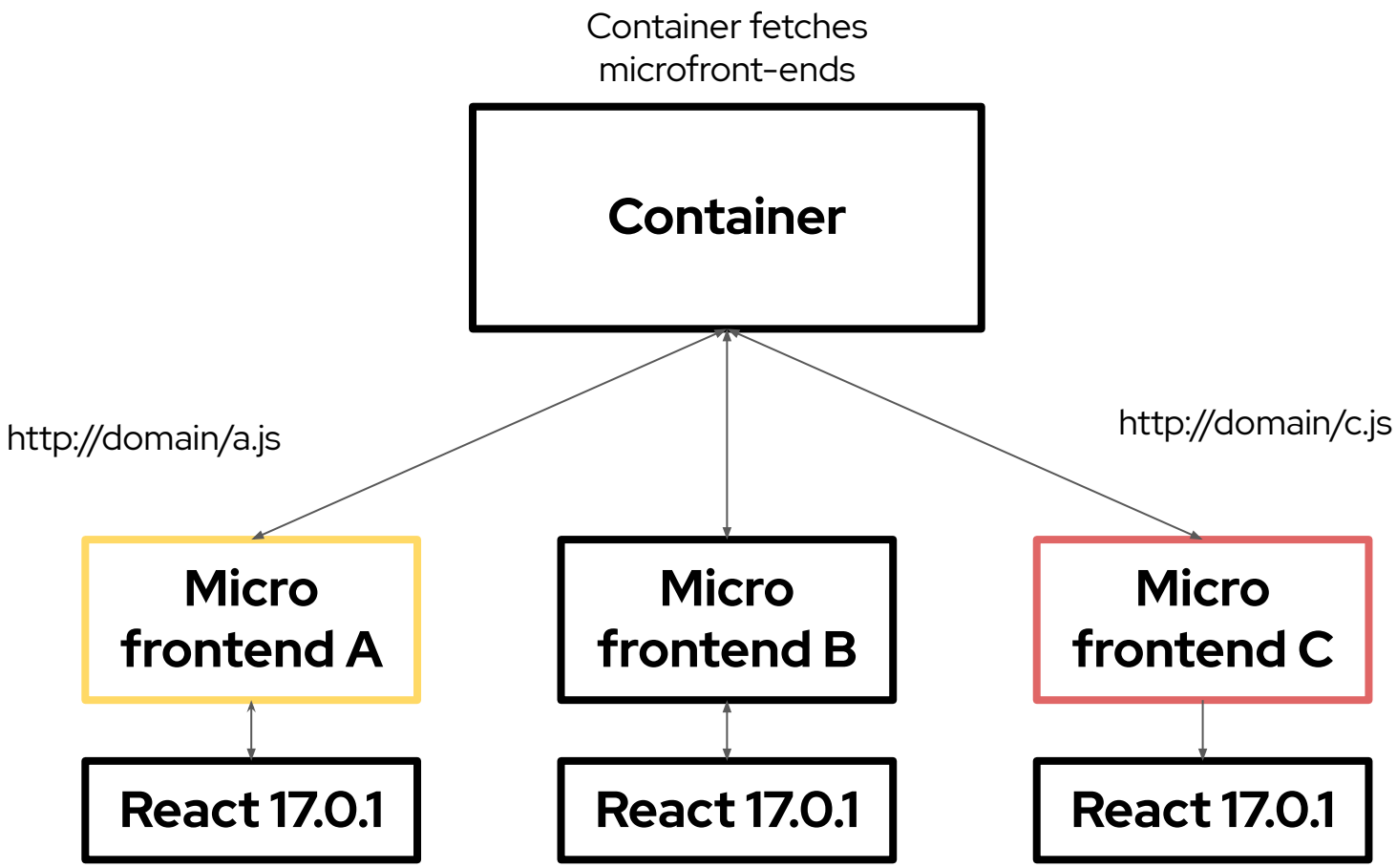
Built time x Runtime Integration

Build-time issues

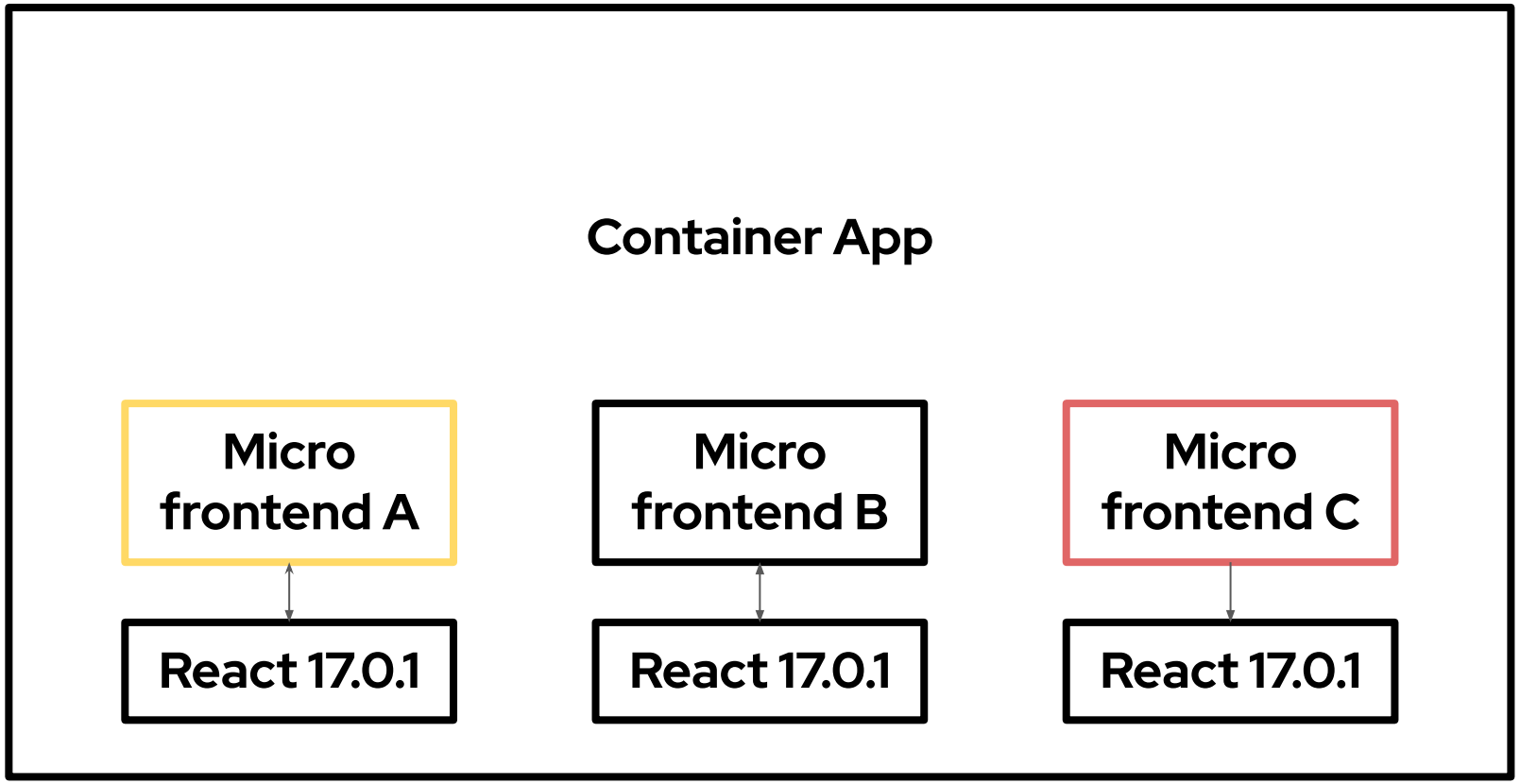
- Foreign modules are accessible during build
- Container has to be **re-deployed** every time child has updated and tempting to tightly coupled Container + child together;
- One single change to prod. requires full a long rebuild
- Dependency versions alignment
- No clear app/team isolation
- **Duplication of library loading**



Red Hat


Container fetches
microfront-ends
Container
http://domain/a.js
http://domain/c.js
Micro
frontend A
Micro
frontend B
Micro
frontend C
React 17.0.1
React 17.0.1
React 17.0.1

Red Hat


Container App
Micro frontend A
Micro frontend B
Micro frontend C
React 17.0.1
React 17.0.1
React 17.0.1

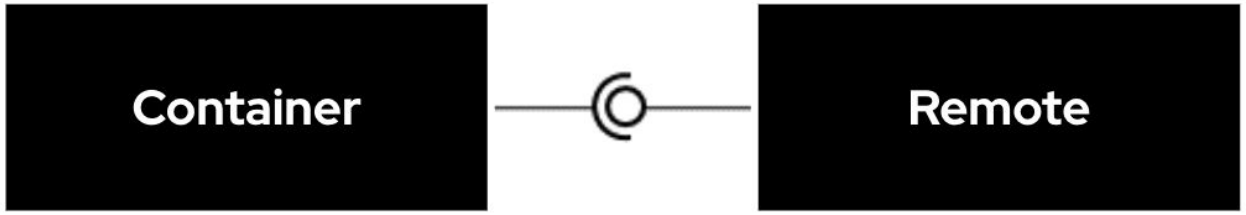
Red Hat

Federated Modules to the Rescue!

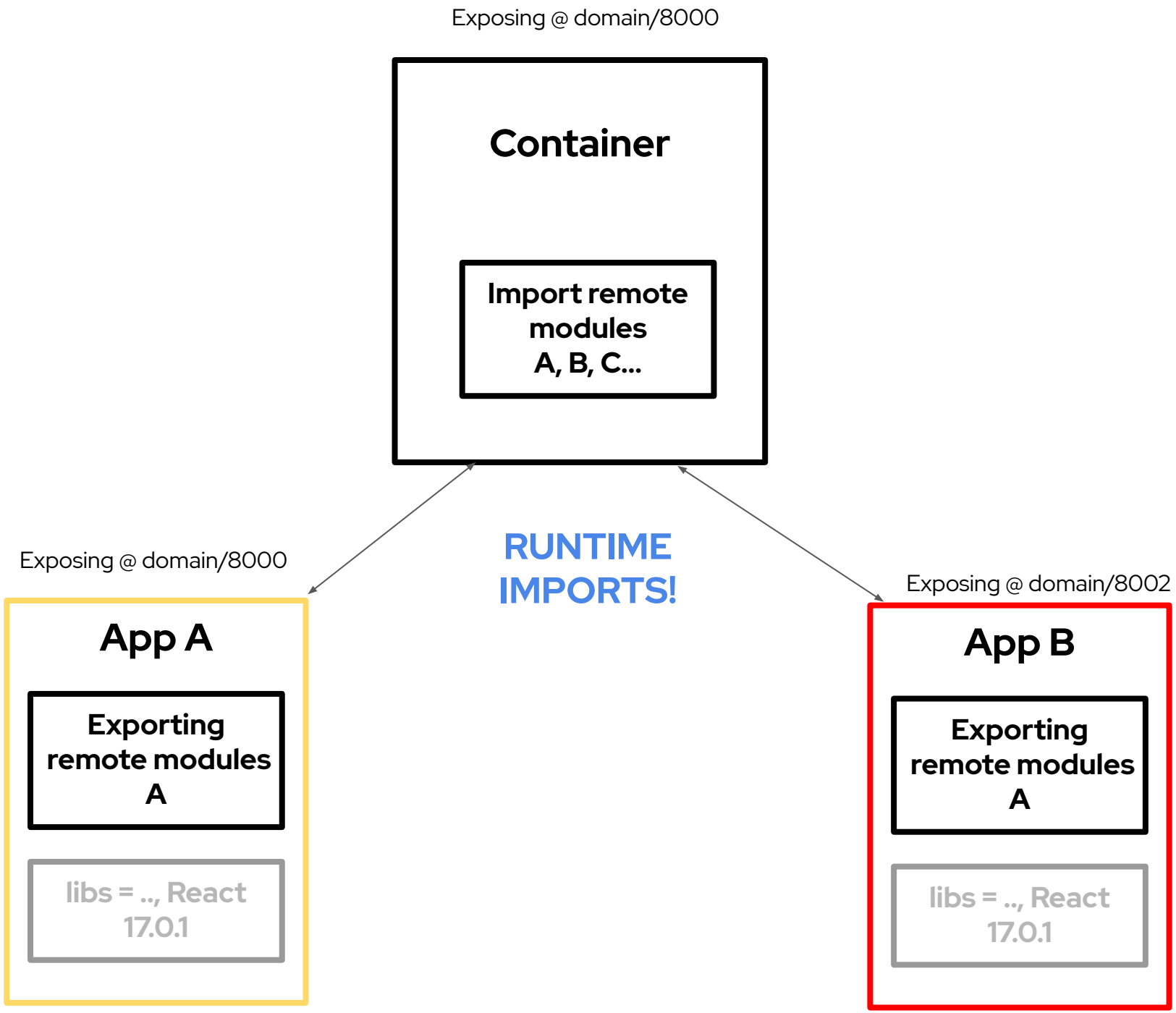
Red Hat

Federated Modules

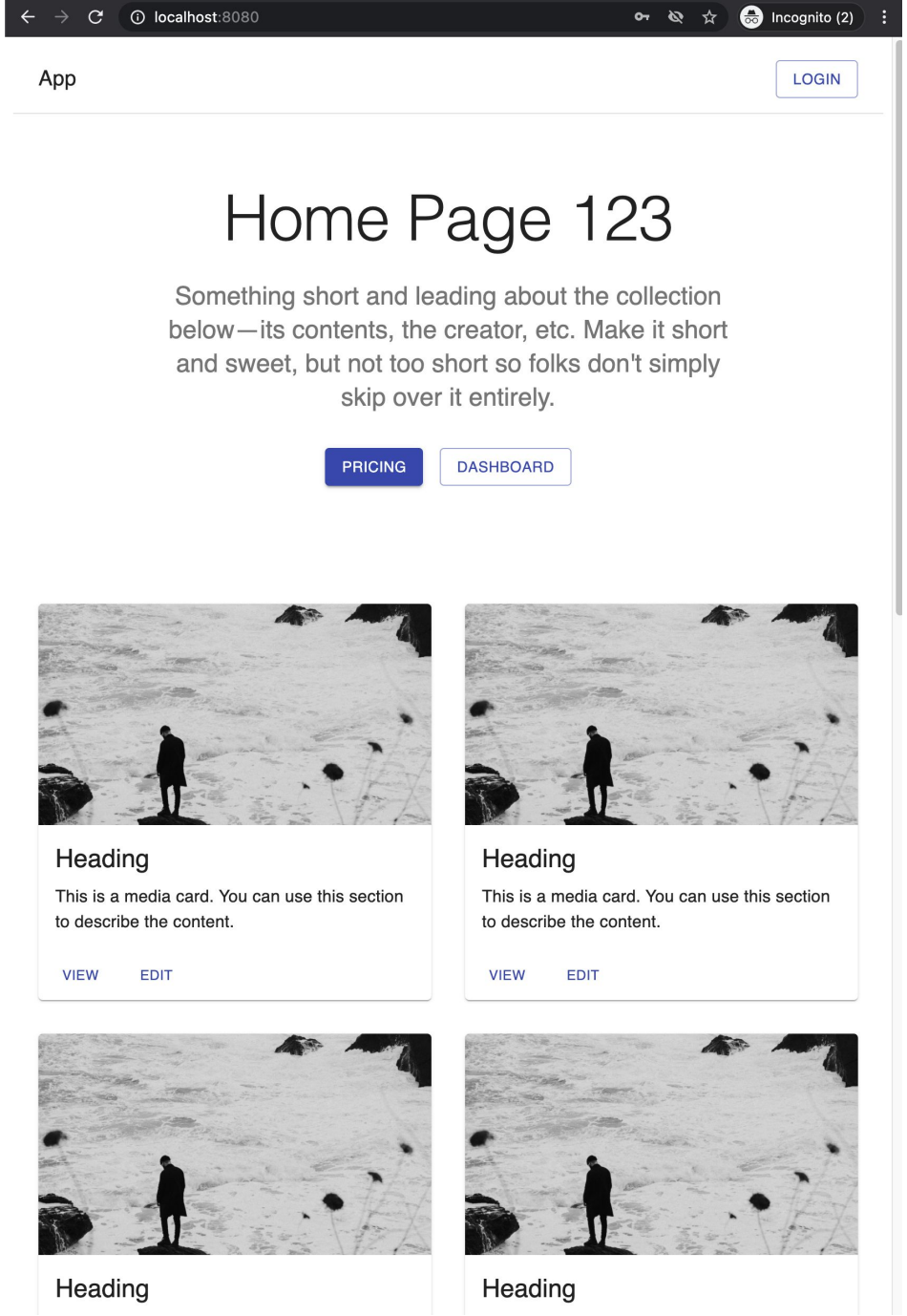
- Part of Webpack 5
- Allows loading separately compiled programs parts
- Solution for runtime integration of Micro frontends?
- Allow referencing program parts that are not yet known at compile time.
- Each micro frontend can run in isolation

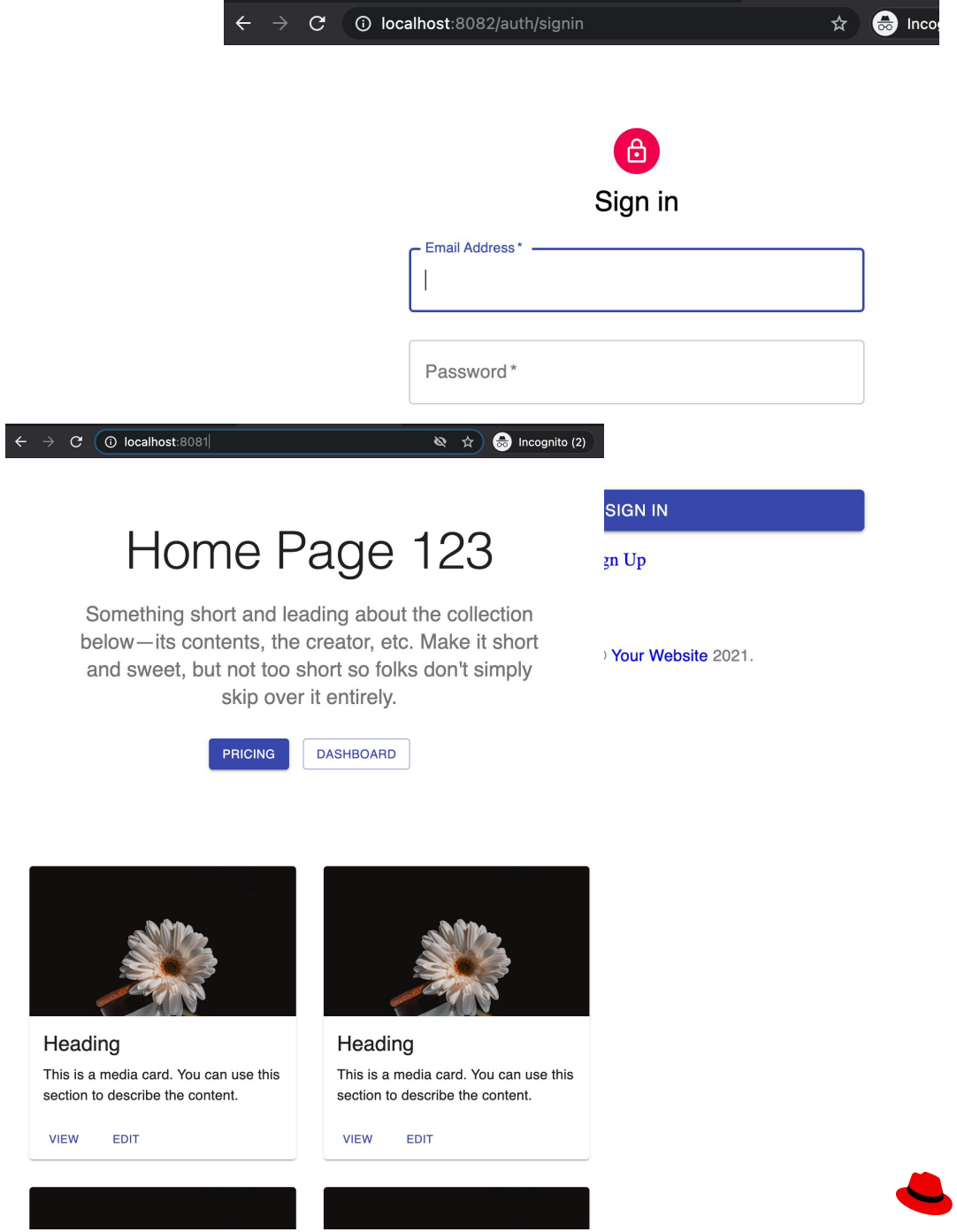


Red Hat


Exposing @ domain/8000
Container
Import remote modules A, B, C...
RUNTIME IMPORTS!
Exposing @ domain/8000
App A
Exporting remote modules A
libs = .., React 17.0.1
Exposing @ domain/8002
App B
Exporting remote modules A
libs = .., React 17.0.1

Red Hat


localhost:8080
Incognito (2)
App
LOGIN
Home Page 123
Something short and leading about the collection below—its contents, the creator, etc. Make it short and sweet, but not too short so folks don't simply skip over it entirely.
PRICING
DASHBOARD
Heading
This is a media card. You can use this section to describe the content.
VIEW
EDIT
Heading
This is a media card. You can use this section to describe the content.
VIEW
EDIT
Heading
Heading


localhost:8082/auth/signin
Sign in
Email Address *
Password *
SIGN IN
gn Up
Your Website 2021.
localhost:8081
Incognito (2)
Home Page 123
Something short and leading about the collection below—its contents, the creator, etc. Make it short and sweet, but not too short so folks don't simply skip over it entirely.
PRICING
DASHBOARD
Heading
This is a media card. You can use this section to describe the content.
VIEW
EDIT
Heading
This is a media card. You can use this section to describe the content.
VIEW
EDIT

Red Hat

```js
import React, { lazy, Suspense, useState, useEffect } from "react";
import { Router, Route, Switch, Redirect } from "react-router-dom";
...
const MarketingLazy = lazy(() => import("./components/MarketingApp"));
const AuthLazy = lazy(() => import("./components/AuthApp"));
const DashboardLazy = lazy(() => import("./components/DashboardApp"));
...

  return (
    <Router history={history}>
      <StylesProvider generateClassName={generateClassName}>
        <div>
          <Header
            isSignedin={isSignedIn}
            onSignOut={() => setIsSignedIn(false)}
          />
          <Suspense fallback={<Progress />}>
            <Switch>
              <Route path="/auth">
                <AuthLazy onSignIn={() => setIsSignedIn(true)} />
              </Route>
              <Route path="/dashboard">
                {!isSignedIn && <Redirect to="/" />}
                <DashboardLazy />
              </Route>
              <Route path="/" component={MarketingLazy} />
            </Switch>
          </Suspense>
        </div>
      </StylesProvider>
    </Router>
  );
};
```

-> Import of Federated Modules

-> Lazy loading via Route

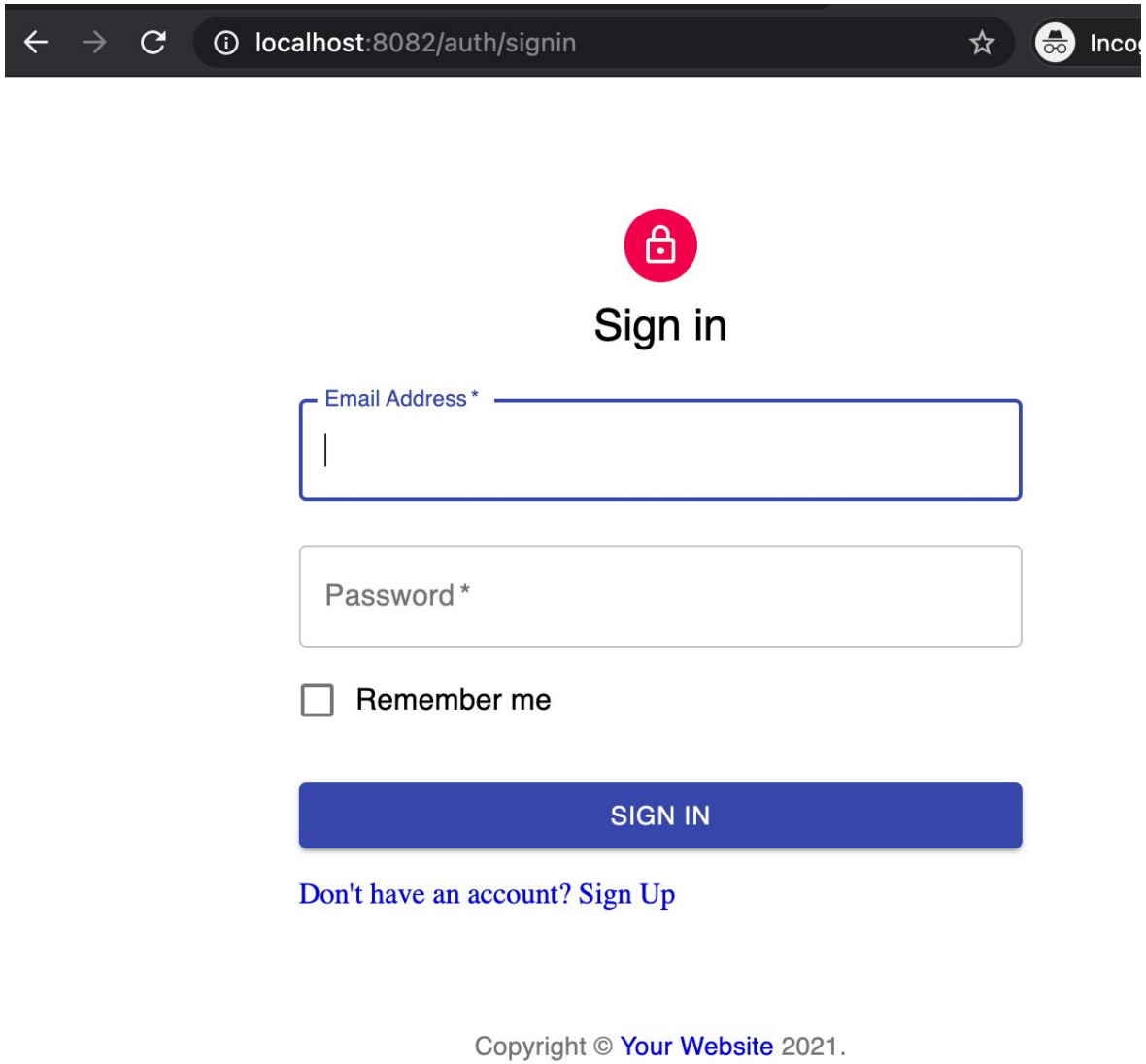
Red Hat

```
1  const { merge } = require("webpack-merge");
2  const HtmlWebpackPlugin = require("html-webpack-plugin");
3  const ModuleFederationPlugin = require("webpack/lib/container/ModuleFederationPlugin");
4  const commonConfig = require("./webpack.common");
5  const packageJson = require("../package.json");
6
7  const domain = process.env.PRODUCTION_DOMAIN;
8
9  const prodConfig = {
10   mode: "production",
11   output: {
12     filename: "[name].[contenthash].js",
13     publicPath: "/container/latest/",
14   },
15   plugins: [
16     new ModuleFederationPlugin({
17       name: "container",
18       remotes: {
19         marketing: `marketing@${domain}/marketing/latest/remoteEntry.js`,
20         auth: `auth@${domain}/auth/latest/remoteEntry.js`,
21         dashboard: `dashboard@${domain}/dashboard/latest/remoteEntry.js`,
22       },
23       shared: packageJson.dependencies,
24     }),
25   ],
26 };
27
28 module.exports = merge(commonConfig, prodConfig);
29
```

-> ModuleFederation Plugin

-> Remote Routes

Red Hat


localhost:8082/auth/signin
Inco
Sign in
Email Address *
Password *
Remember me
SIGN IN
Don't have an account? Sign Up
Copyright © Your Website 2021.

Red Hat

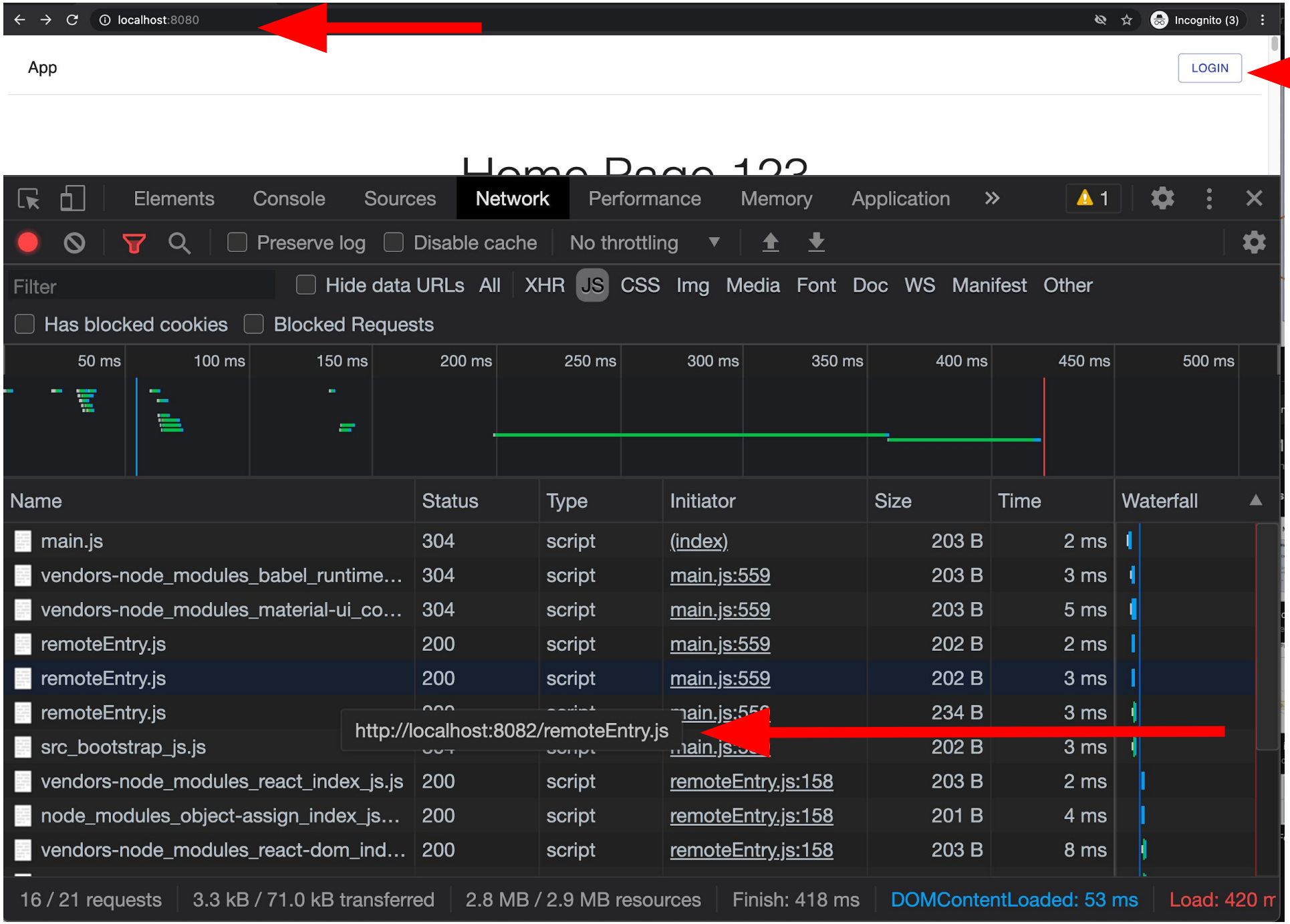
webpack.prod.js

```javascript
const { merge } = require("webpack-merge");
const ModuleFederationPlugin = require("webpack/lib/container/Modu
const packageJson = require("../package.json");
const commonConfig = require("./webpack.common");

const prodConfig = {
  mode: "production",
  output: {
    filename: "[name].[contenthash].js",
    publicPath: "/auth/latest/",
  },
  plugins: [
    new ModuleFederationPlugin({
      name: "auth",
      filename: "remoteEntry.js",
      exposes: {
        "./AuthApp": "./src/bootstrap",
      },
      shared: packageJson.dependencies,
    }),
  ],
};

module.exports = merge(commonConfig, prodConfig);
```

-> ModuleFederation Plugin

-> Exposed routes
(used by Container)

Red Hat


localhost:8080
Incognito (3)
App
LOGIN
Elements
Console
Sources
Network
Performance
Memory
Application
Preserve log
Disable cache
No throttling
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Has blocked cookies
Blocked Requests
50 ms
100 ms
150 ms
200 ms
250 ms
300 ms
350 ms
400 ms
450 ms
500 ms
Name
Status
Type
Initiator
Size
Time
Waterfall
main.js 304 script (index) 203 B 2 ms
vendors-node_modules_babel_runtime… 304 script main.js:559 203 B 3 ms
vendors-node_modules_material-ui_co… 304 script main.js:559 203 B 5 ms
remoteEntry.js 200 script main.js:559 202 B 2 ms
remoteEntry.js 200 script main.js:559 202 B 3 ms
remoteEntry.js 234 B 3 ms
http://localhost:8082/remoteEntry.js
src_bootstrap_js.js 202 B 3 ms
vendors-node_modules_react_index_js.js 200 script remoteEntry.js:158 203 B 2 ms
node_modules_object-assign_index_js… 200 script remoteEntry.js:158 201 B 4 ms
vendors-node_modules_react-dom_ind… 200 script remoteEntry.js:158 203 B 8 ms
16 / 21 requests
3.3 kB / 71.0 kB transferred
2.8 MB / 2.9 MB resources
Finish: 418 ms
DOMContentLoaded: 53 ms
Load: 420 m

Red Hat

localhost:8080/auth/signin

Sign in

Email Address *

Password *

Elements | Console | Sources | Network | Performance | Memory | Application

Preserve log | Disable cache | No throttling

Filter | Hide data URLs | All | XHR | JS | CSS | Img | Media | Font | Doc | WS | Manifest | Other

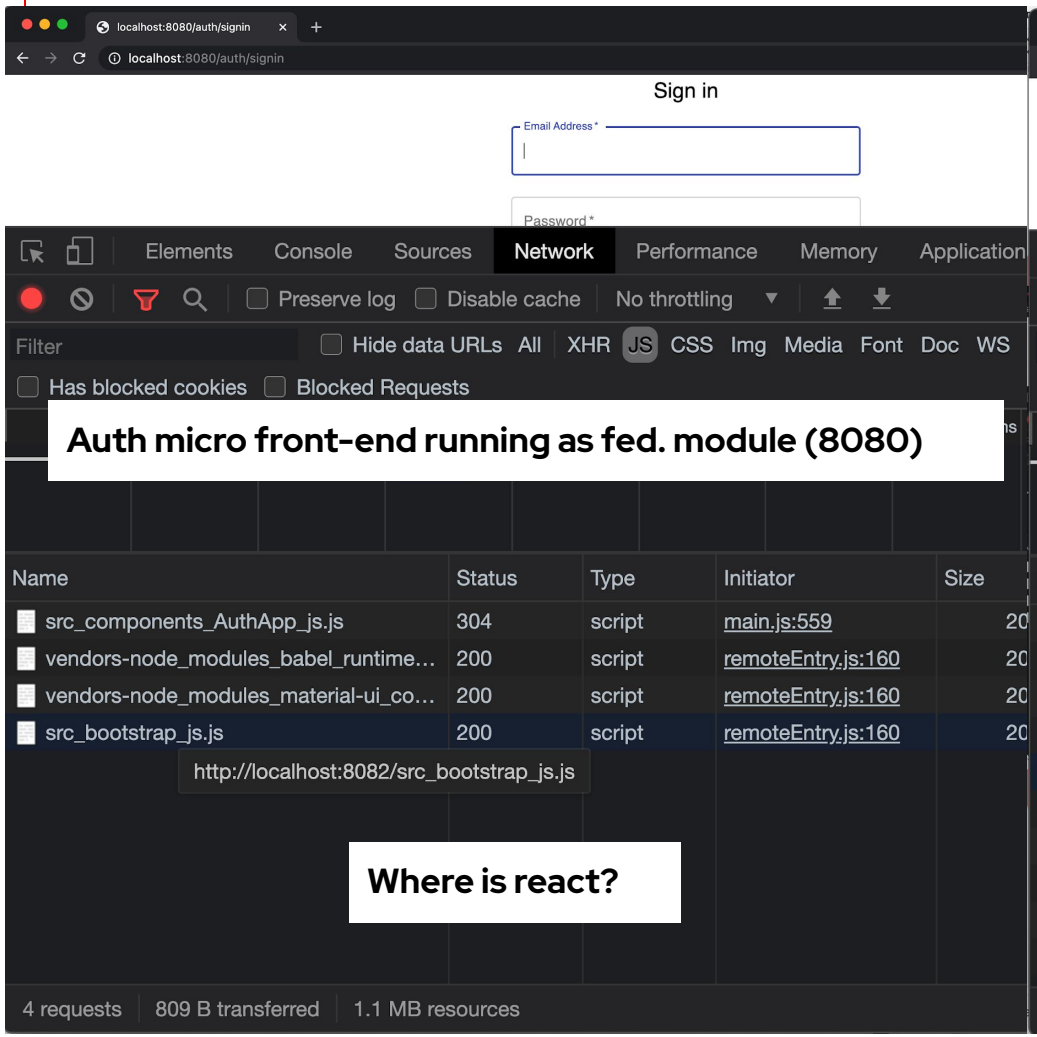
Has blocked cookies | Blocked Requests

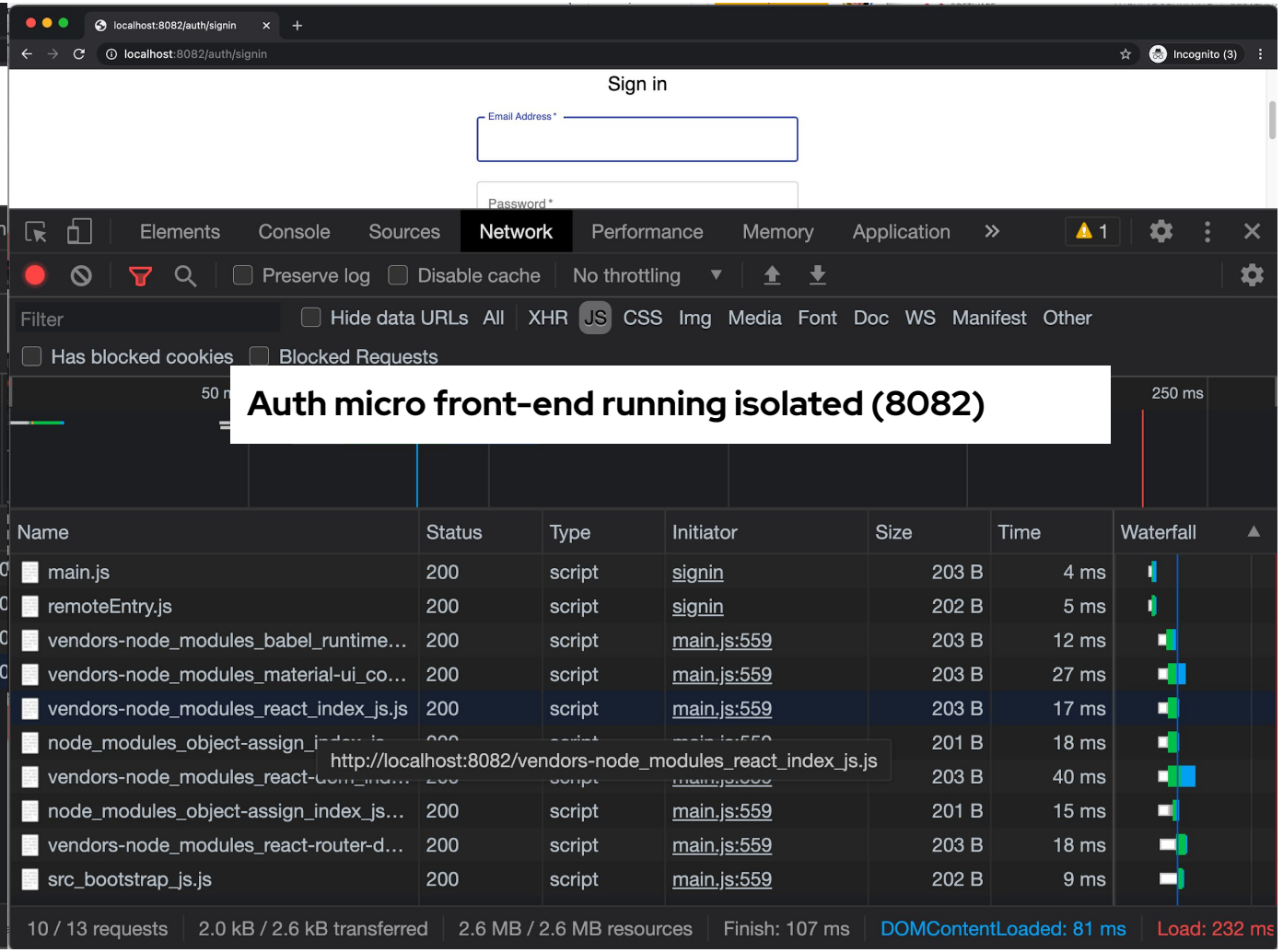
Name	Status	Type	Initiator	Size	Time	Waterfall
src_components_AuthApp_js.js	304	script	main.js:559	201 B	3 ms	
vendors-node_modules_babel_runtime…	200	script	remoteEntry.js:160	203 B	3 ms	
vendors-node_modules_material-ui_co…	200	script	remoteEntry.js:160	203 B	14 ms	
src_bootstrap_js.js	200	script	remoteEntry.js:160	202 B	7 ms	

http://localhost:8082/src_bootstrap_js.js

4 requests | 809 B transferred | 1.1 MB resources

Red Hat


localhost:8080/auth/signin
Sign in
Email Address *
Password *
Elements Console Sources Network Performance Memory Application
Preserve log Disable cache No throttling
Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS
Has blocked cookies Blocked Requests
Auth micro front-end running as fed. module (8080)
Name Status Type Initiator Size
src_components_AuthApp_js.js 304 script main.js:559
vendors-node_modules_babel_runtime... 200 script remoteEntry.js:160
vendors-node_modules_material-ui_co... 200 script remoteEntry.js:160
src_bootstrap_js.js 200 script remoteEntry.js:160
http://localhost:8082/src_bootstrap_js.js
Where is react?
4 requests 809 B transferred 1.1 MB resources


localhost:8082/auth/signin
Incognito (3)
Sign in
Email Address *
Password *
Elements Console Sources Network Performance Memory Application
Preserve log Disable cache No throttling
Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other
Has blocked cookies Blocked Requests
Auth micro front-end running isolated (8082)
250 ms
Name Status Type Initiator Size Time Waterfall
main.js 200 script signin 203 B 4 ms
remoteEntry.js 200 script signin 202 B 5 ms
vendors-node_modules_babel_runtime... 200 script main.js:559 203 B 12 ms
vendors-node_modules_material-ui_co... 200 script main.js:559 203 B 27 ms
vendors-node_modules_react_index_js.js 200 script main.js:559 203 B 17 ms
node_modules_object-assign_index_js... 200 script main.js:559 201 B 15 ms
vendors-node_modules_react-router-d... 200 script main.js:559 203 B 18 ms
src_bootstrap_js.js 200 script main.js:559 202 B 9 ms
http://localhost:8082/vendors-node_modules_react_index_js.js
10 / 13 requests 2.0 kB / 2.6 kB transferred 2.6 MB / 2.6 MB resources Finish: 107 ms DOMContentLoaded: 81 ms Load: 232 ms

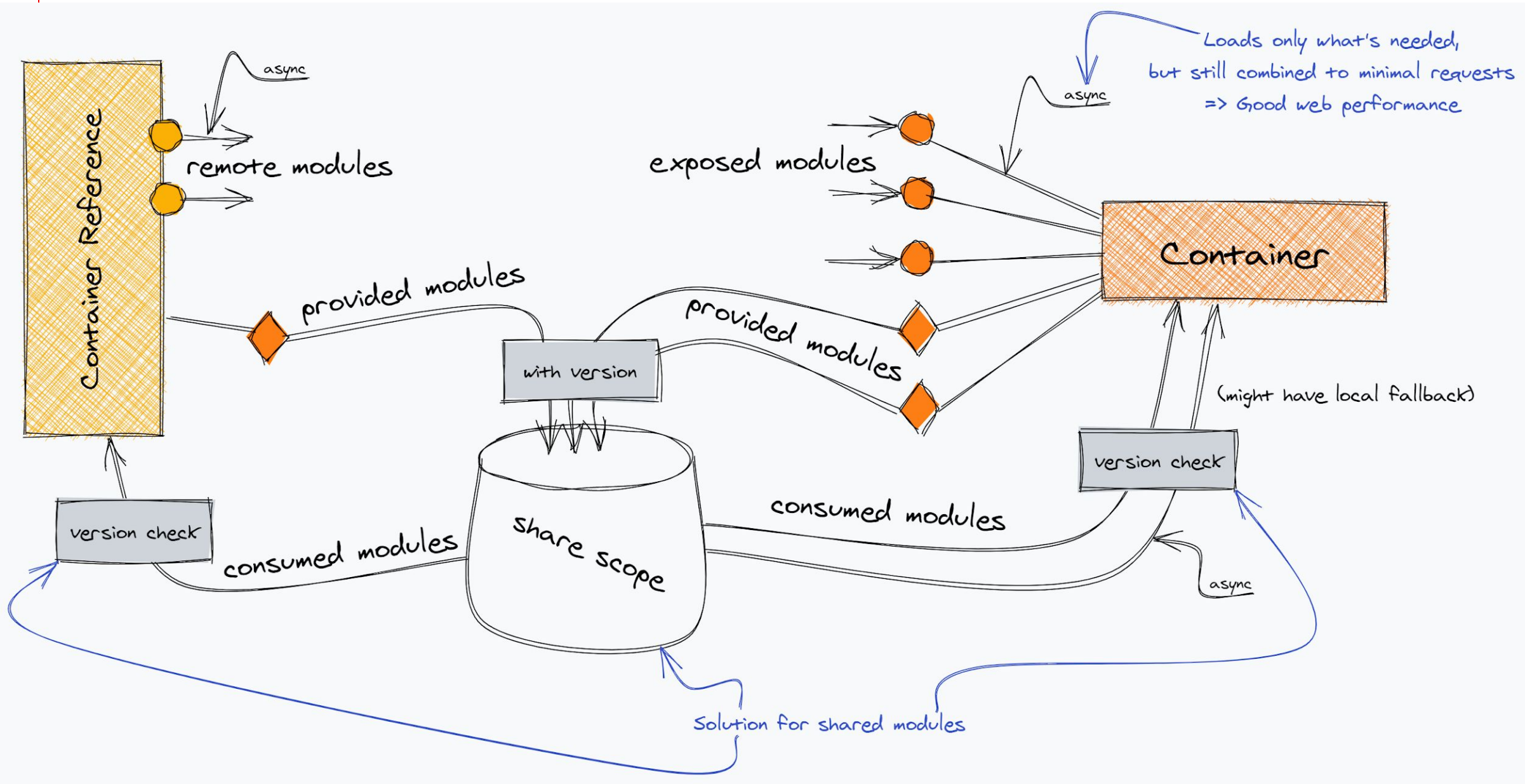
Red Hat

webpack.prod.js

```
1  const { merge } = require("webpack-merge");
2  const ModuleFederationPlugin = require("webpack/lib/container/Modu
3  const packageJson = require("../package.json");
4  const commonConfig = require("./webpack.common");
5
6  const prodConfig = {
7    mode: "production",
8    output: {
9      filename: "[name].[contenthash].js",
10     publicPath: "/auth/latest/",
11   },
12   plugins: [
13     new ModuleFederationPlugin({
14       name: "auth",
15       filename: "remoteEntry.js",
16       exposes: {
17         "./AuthApp": "./src/bootstrap",
18       },
19       shared: packageJson.dependencies,
20     }),
21   ],
22 };
23
24 module.exports = merge(commonConfig, prodConfig);
25
```

-> shared dependencies

Red Hat



https://github.com/sokra/slides/blob/master/content/ModuleFederationWebpack5.md

Red Hat

Federated Modules

Pros in our case

- **Microservices architecture**
- Able to finally take advantage of runtime integration
- Each team can build/deploy their own micro frontend
- No duplication of library loading
- Ability to deploy multiple pieces of your application to different servers without iframes
- Have a portion of an application getting too big and wants a dedicated team? Split it out.
- That split you just made was a bad idea? Merge it back together.
- Finally we are able to real decoupling.
- Be able to evolve tech stack independently

Federated Modules

Possible Con's for us

- **Microservices architecture**
- Distributed systems are hard
- Bleeding edge technology
- Complexity of deployment
  - If you don’t have a pretty solid CI/CD this will probably be a foot-gun for you
- Requires unbreakable API boundary that everyone agrees

Good frontend development is hard.

Red Hat

React 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

Principal Software Engineer

@ederign

Alex Porcelli

Senior Principal Software Engineer

@porcelli

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Red Hat