



# Frontend Monoliths: Run if you can!



Daniel Wiehl  
Developer mtrail GmbH  
[daniel.wiehl@mtrail.ch](mailto:daniel.wiehl@mtrail.ch)



Jonas Bandi  
Freelance Developer & Teacher  
[jonas.bandi@ivorycode.com](mailto:jonas.bandi@ivorycode.com)  
 @jbandi

# Agenda

From Frontend Monolith to Micro-Frontends

Introducing the SCION Workbench

SCION Workbench Demo

In 2016 the average webpage size passed the size of Doom



 **ronan cremin**  
@xbs



In about 7 months average web page size will be same as  
Doom install image.

Well done us! Onwards & upwards!

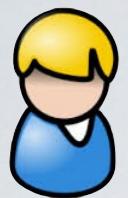
5:48 PM - Jul 30, 2015

 1,761  3,907 people are talking about this

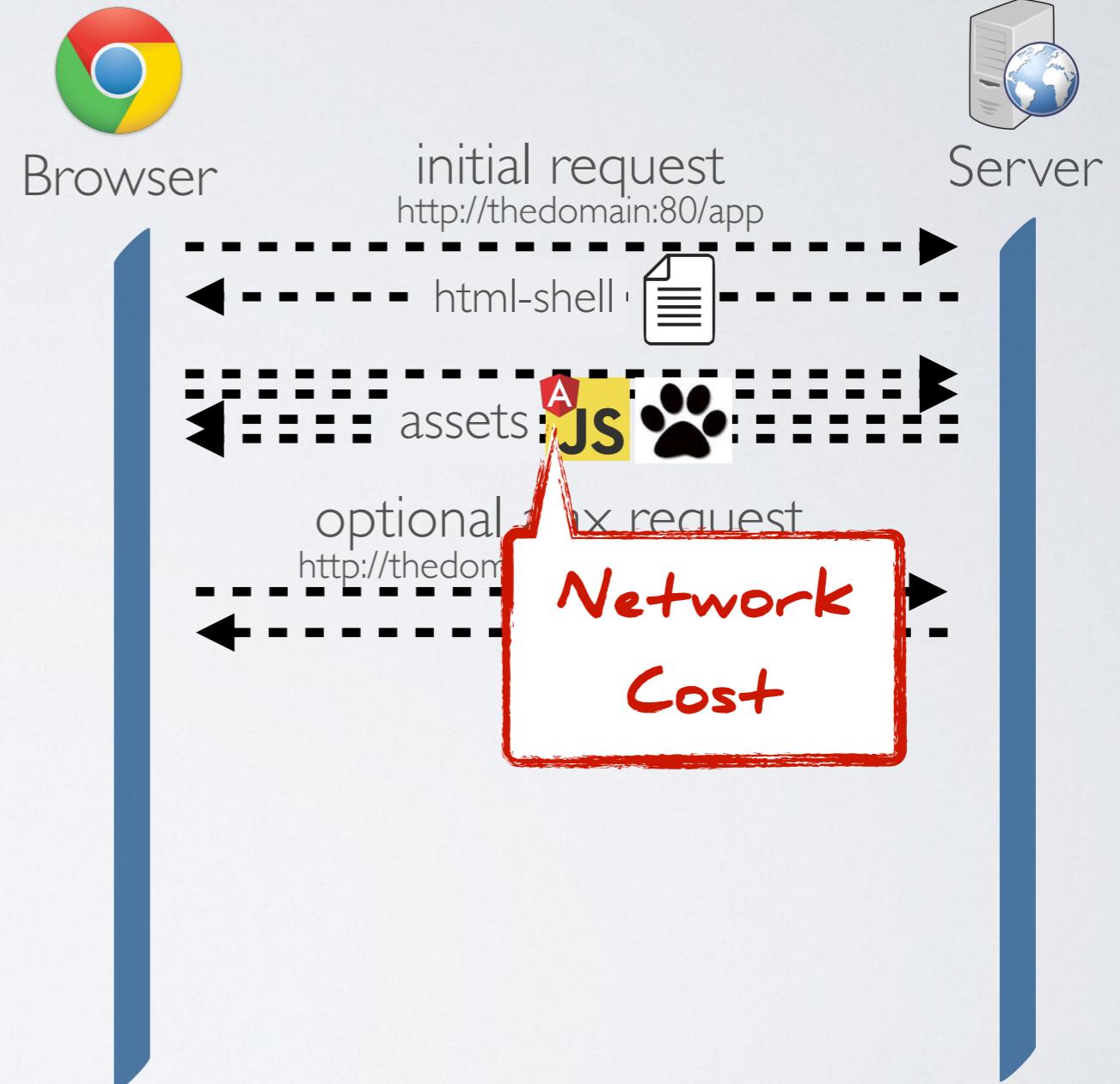
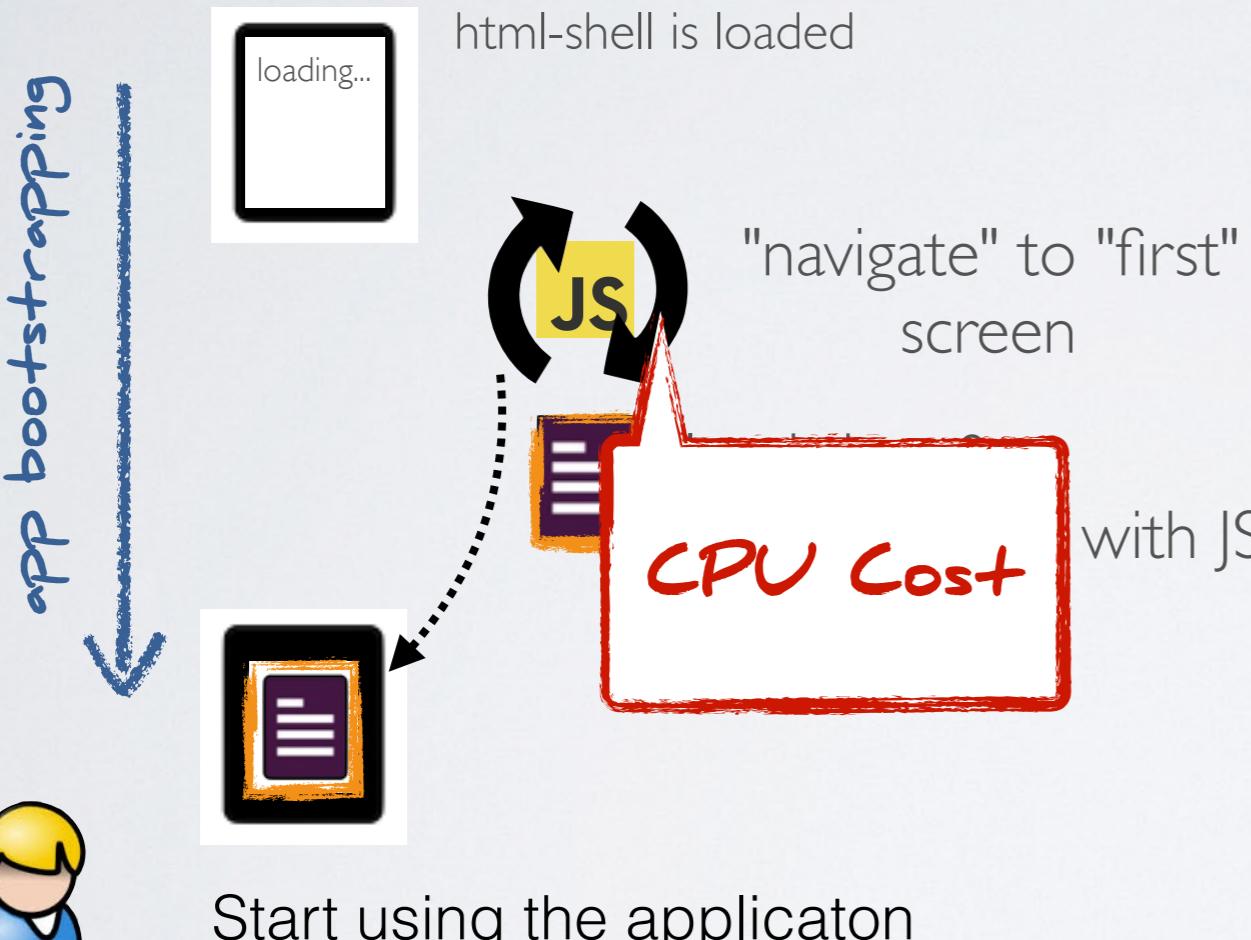


Size does Matter!

# Single Page Applications (SPA)

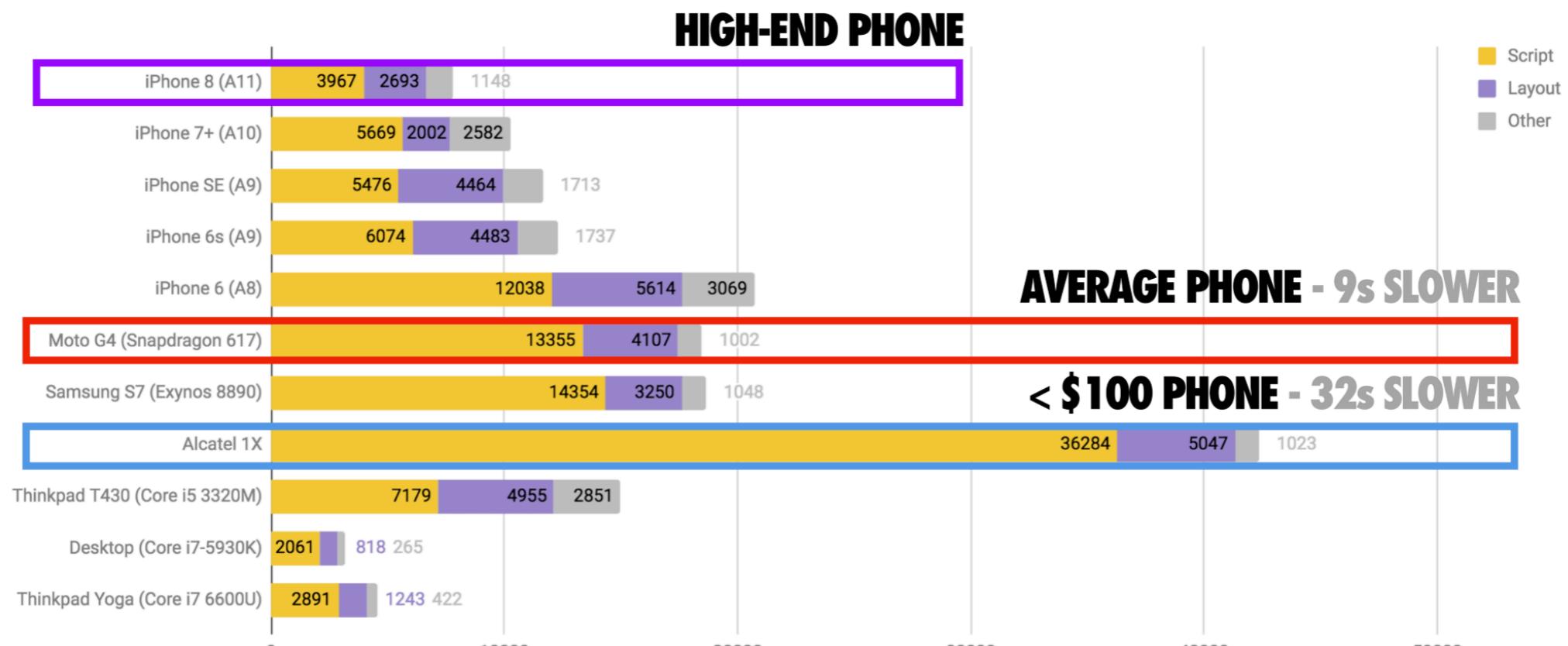


http://thedomain:80/app#first



# The Cost of JavaScript

## JS PROCESSING FOR CNN.COM



For mobile, aim for a JS budget of < 170KB minified/compressed.



JAVASCRIPT

DEVELOPERS

WEBSITE

# Speed == Money

Amazon did tests that showed they would lose \$1.6 BILLION every year if they slowed down by just one second.

Walmart Study: For every 100 ms of improvement, they grew incremental revenue by up to 1%.

A photograph of a large, open-plan office environment. The space is filled with numerous white cubicles, each containing a desk, a computer monitor, and some papers. A woman with long blonde hair, wearing a pink top, is visible from behind, working at her cubicle. The office appears to be quite busy and spacious.

But ...

... Enterprise Applications?



 Christmas.reloadData() 🎄

@AdamRackis

Follow



The volume of devrel drivel fetishizing tiny initial bundle size is exhausting.

If you work on a team of developers shipping line of business software, and your initial bundle is over 200K, and business is good, then you're doing fine.  
Keep shipping, and getting paid 😎

12:47 PM - 9 Dec 2018



Ville Lahdenvuo @tuhoojabotti · Dec 7

Our bundle is just shy of 400k (we're still working on it) but 140k of that is just Angular. Hoping that ivy will make things better. There's also probably some stuff we could eliminate already manually.

1

1

1

1

1



Jake Archibald

@jaffathecake

Following

Replying to @tuhoojabotti @bmeurer and 2 others

If 140k is Angular, Angular has put you in a position where you lose by default. I really hope things improve here.

Preact let us hit first-interaction in 8k of JS.

1:29 PM - 7 Dec 2018

Jake Archibald, Developer Advocate at Google

<https://twitter.com/jaffathecake/status/1071079244246237184>

For typical enterprise-applications, the performance of big JavaScript bundles is not the primary problem:

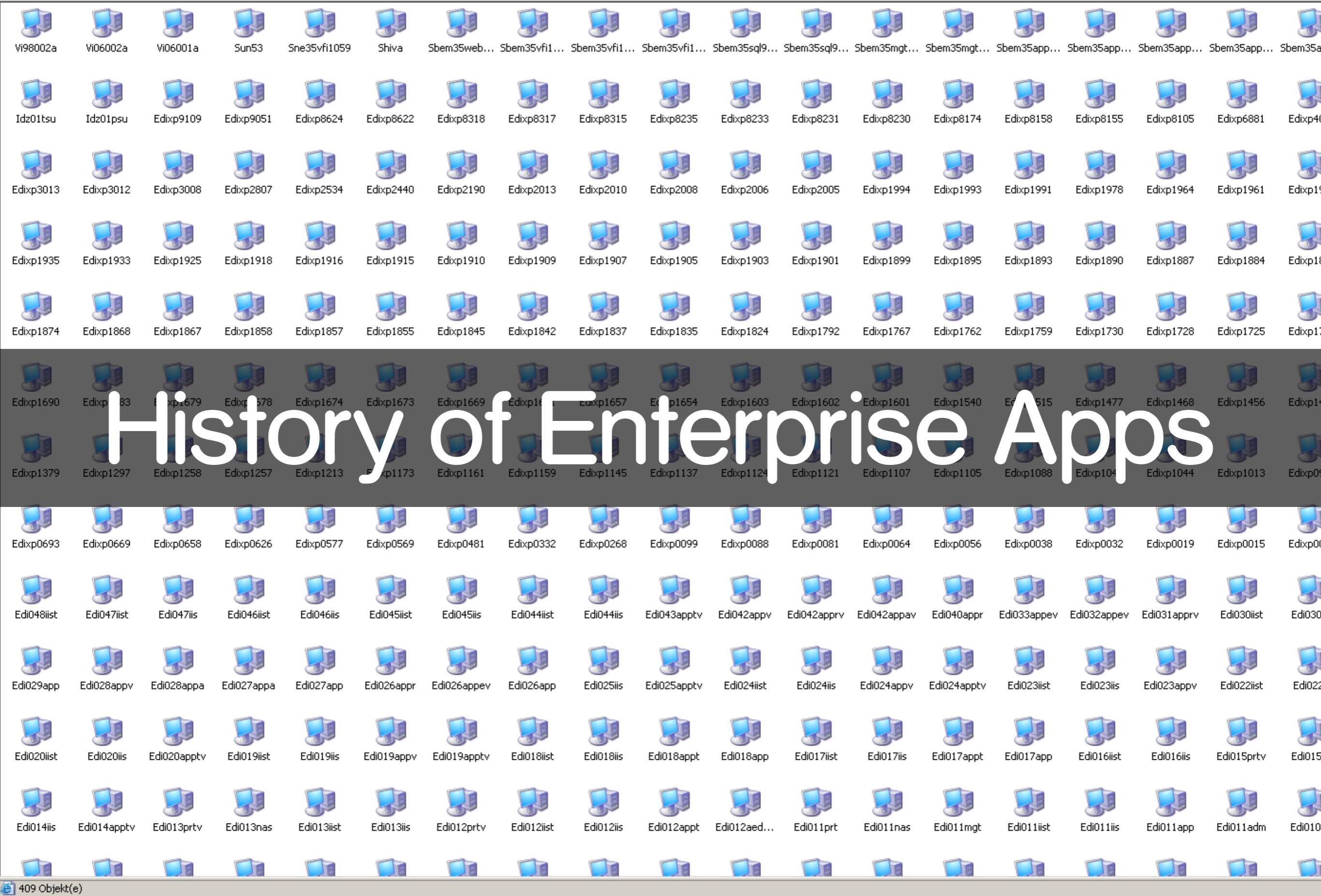
- network and processing power is typically good enough
- often there are other performances bottlenecks (data-fetching, back-end performance ...)



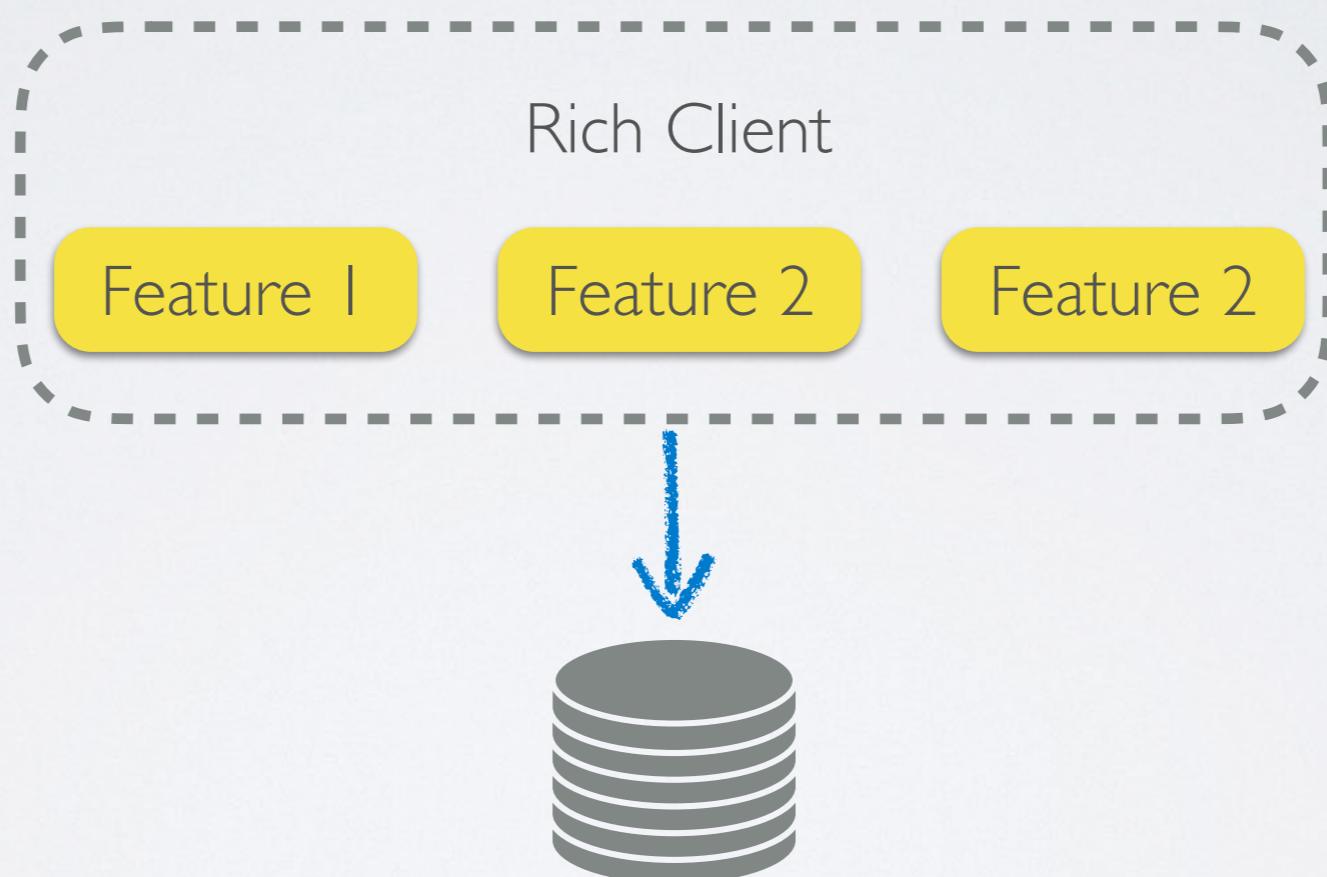
A

Size still might be a problem ...

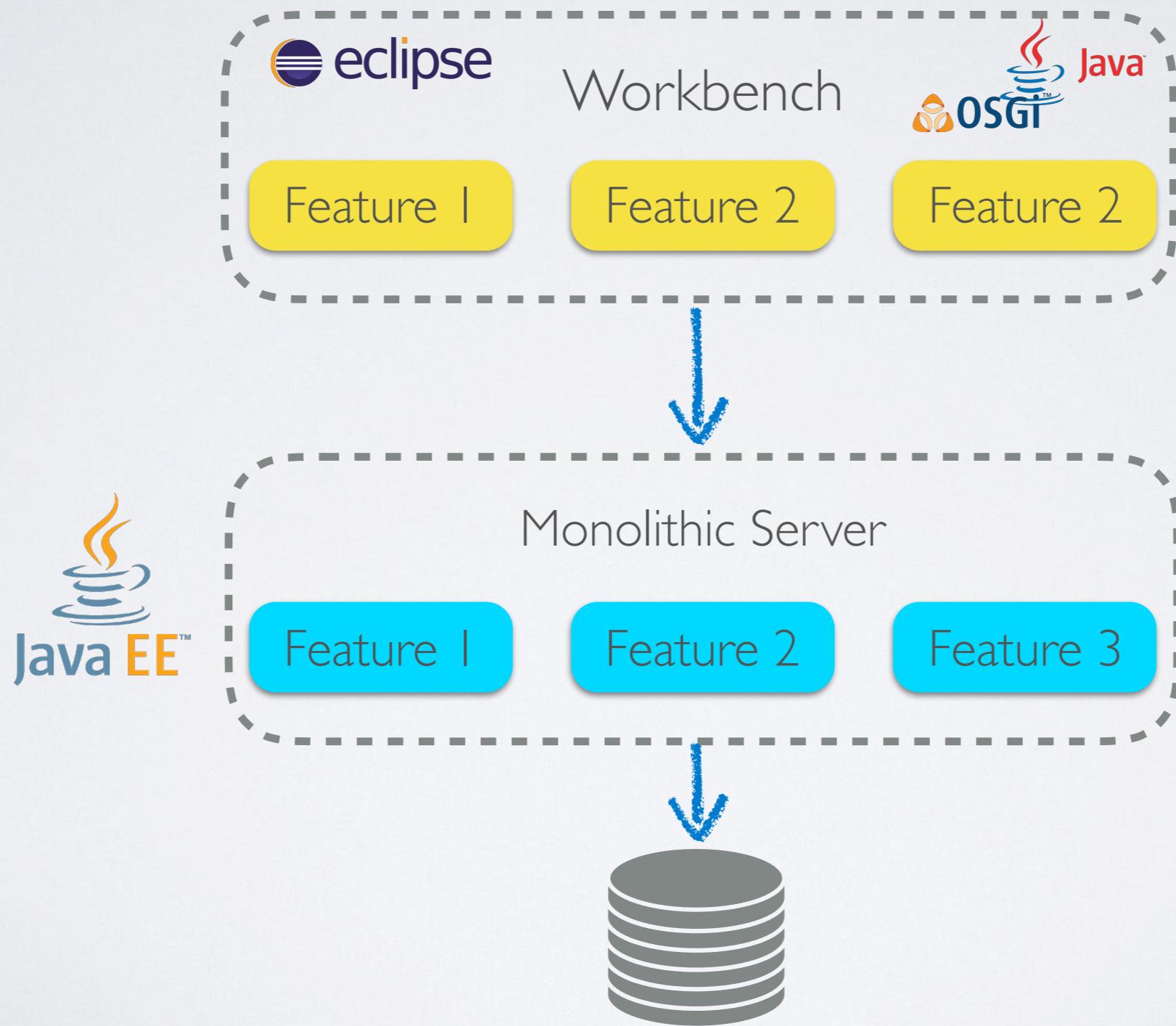
Datei Bearbeiten Ansicht Favoriten Extras ?

Adresse [Edi](#)

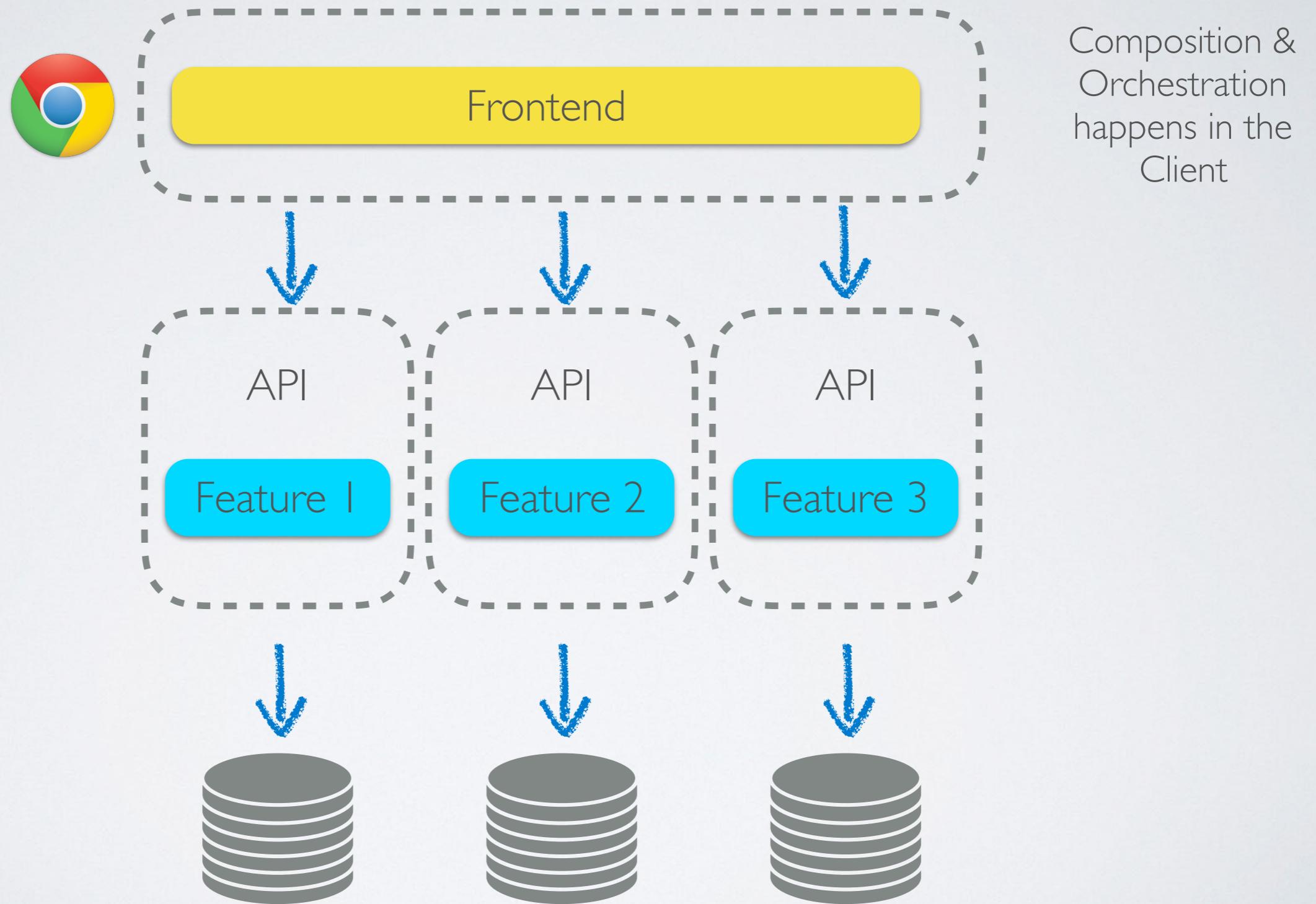
# The Fat Client



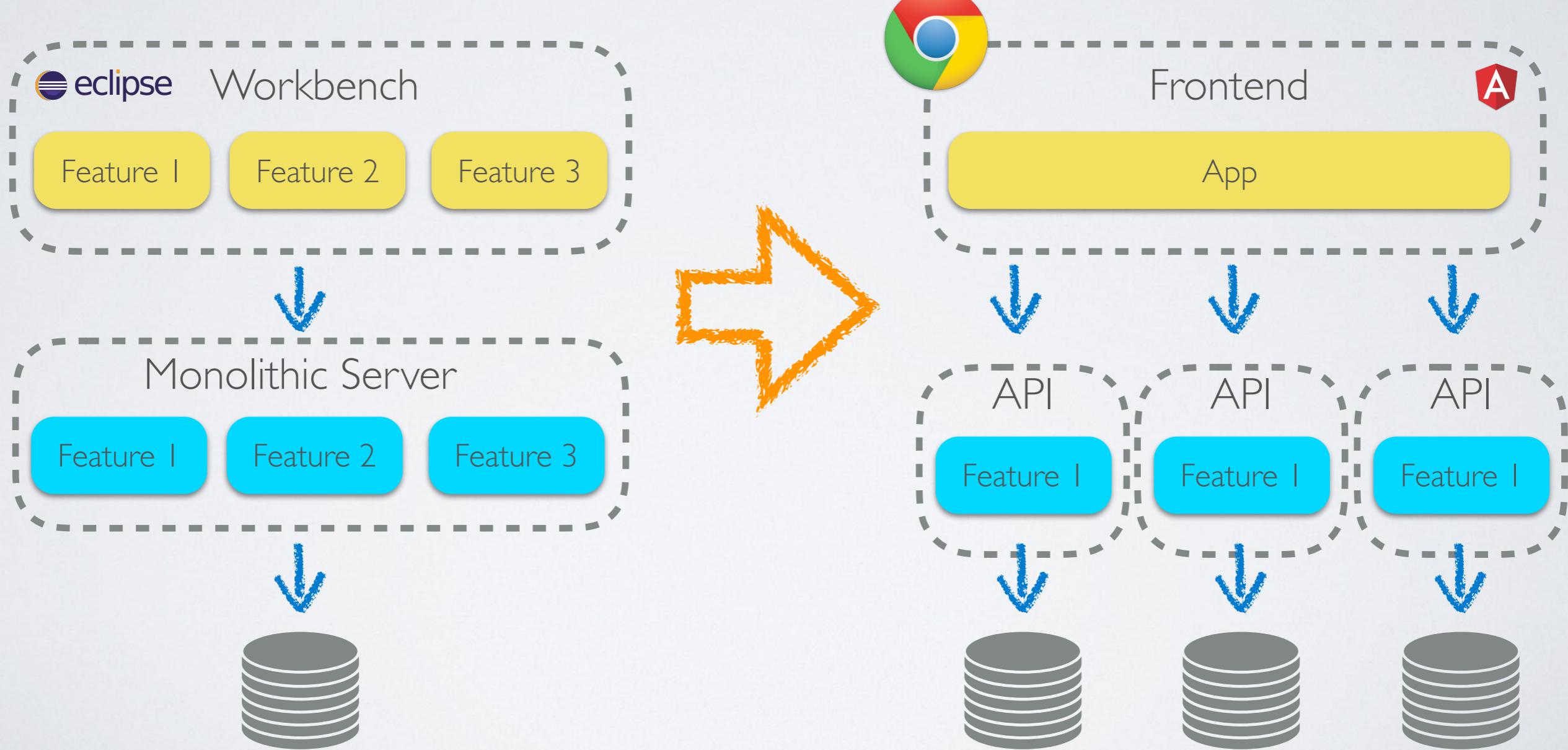
# Workbench & Monolith

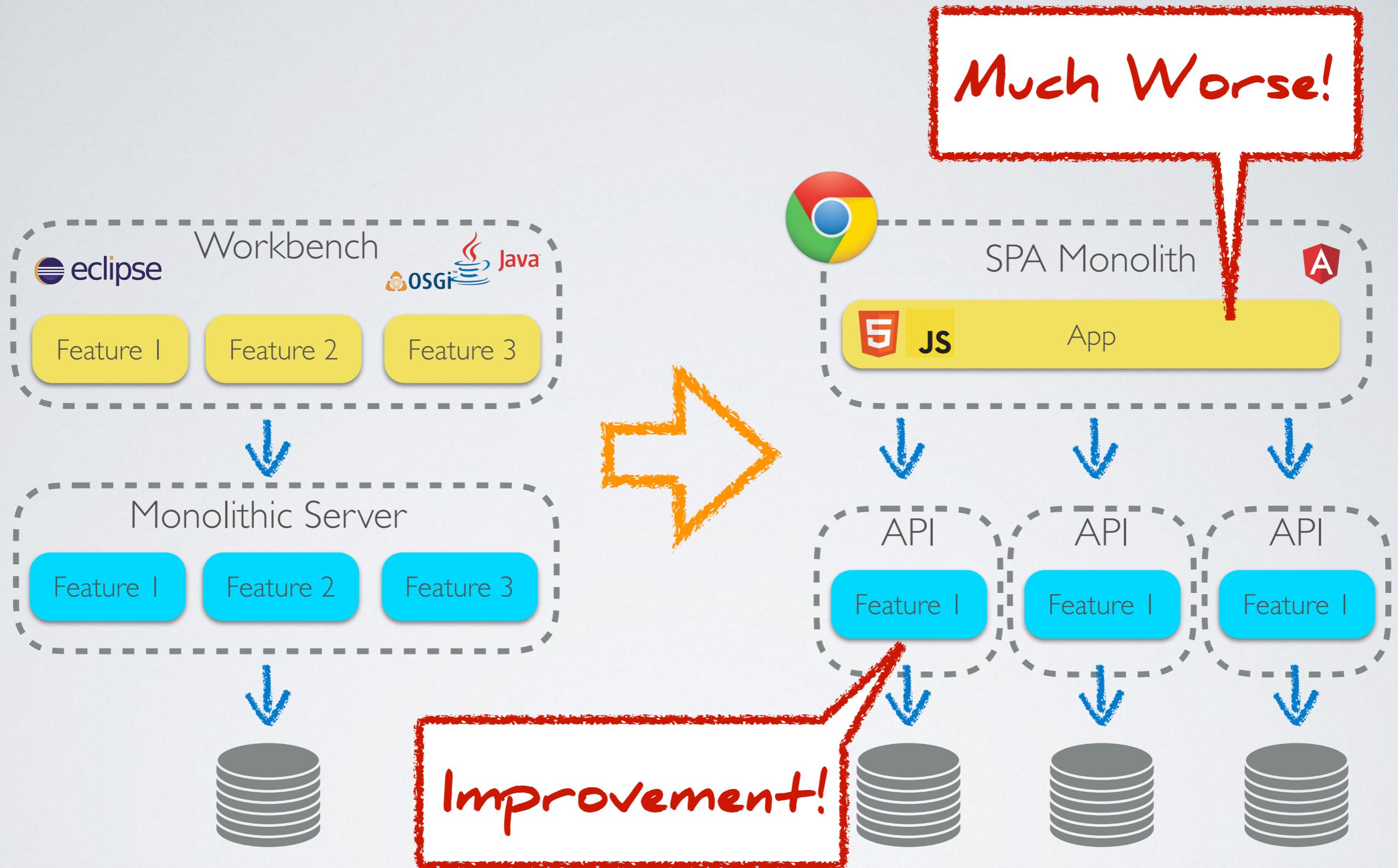


# Microservices



# Rich Client Migrations: The Rise of the Frontend Monoliths

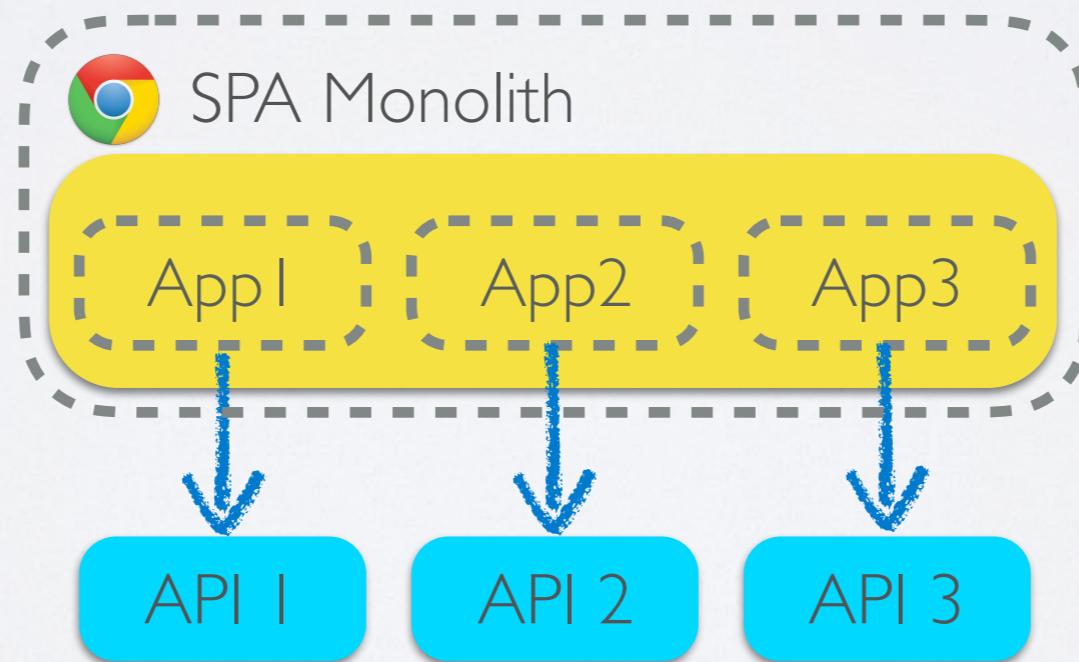




# Beware of the Monolith

Unfortunately, we've seen many teams create front-end monoliths - a single, large and sprawling browser application - on top of their back-end services.

- ThoughtWorks Technology Radar, 2017



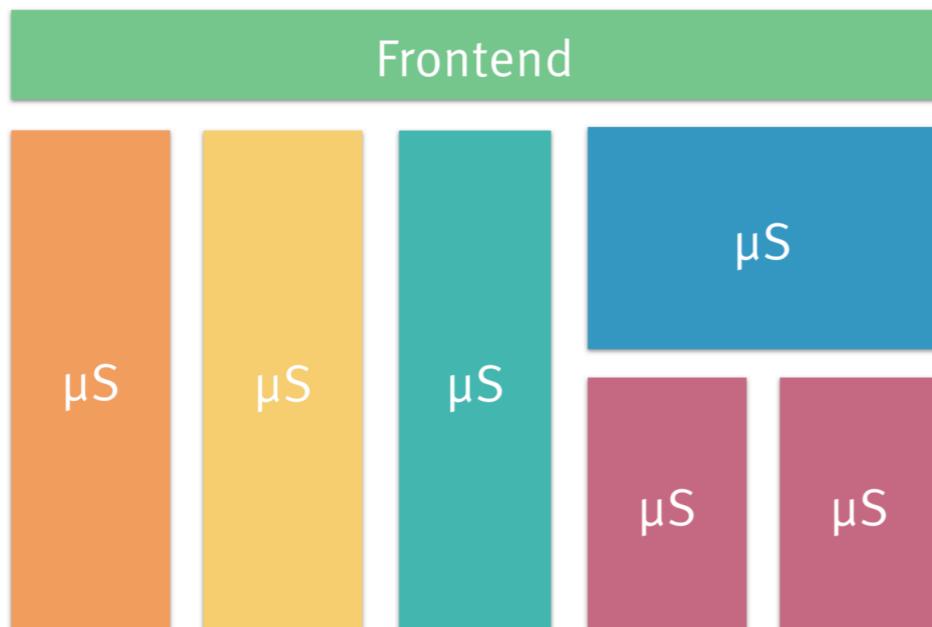


The real problem of  
the monolith:

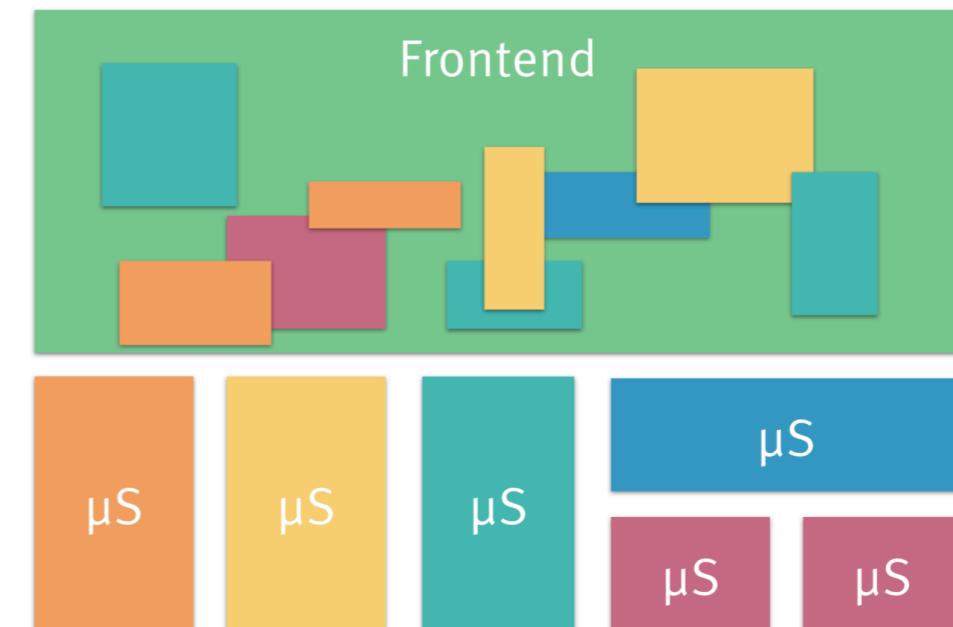
Coupling

# The frontend is not an implementation detail. It is a critical part of a microservices architecture.

Frontend + services in a backend architect's mind



Frontend + services in the real world



Stefan Tilkov :Wait, what!? Our microservices have actual human users?

<https://www.youtube.com/watch?v=pUlglgXA0rfwc>

<https://speakerdeck.com/stilkov/wait-what-our-microservices-have-actual-human-users-1>

# The goal of microservices: enable autonomy

- independent & parallel development
- independent deployments/lifecycles
- autonomous operation / runtime isolation
- technology agnostic

Frontend Monolith

# Size does matter ...



... especially in the enterprise.

# Break the Monolith!



Isolation at development time  
Isolation at run time

A close-up portrait of a man with dark, wavy hair. He has a serious, contemplative expression, looking slightly to his left. The lighting is dramatic, with strong highlights on his forehead and hair, while the rest of his face and the background are in shadow. The background appears to be a dark, textured surface, possibly a wall or a piece of furniture.

The first rule of huge  
enterprise applications is:  
You do not build huge enterprise  
applications.



The second rule of huge  
enterprise applications is:  
You do not build huge  
enterprise applications.



How?

A close-up photograph of a man with dark hair and a serious expression. He is wearing a dark suit jacket over a light-colored shirt. The lighting is dramatic, with strong highlights on his face and shoulders, while the background is dark and out of focus.

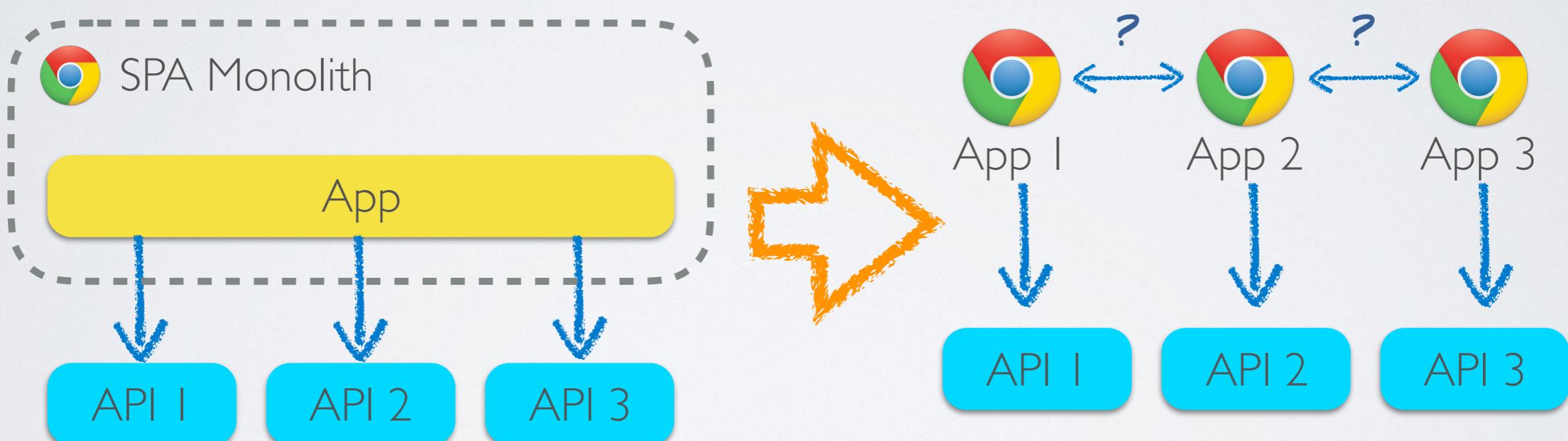
# Micro Frontends:

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

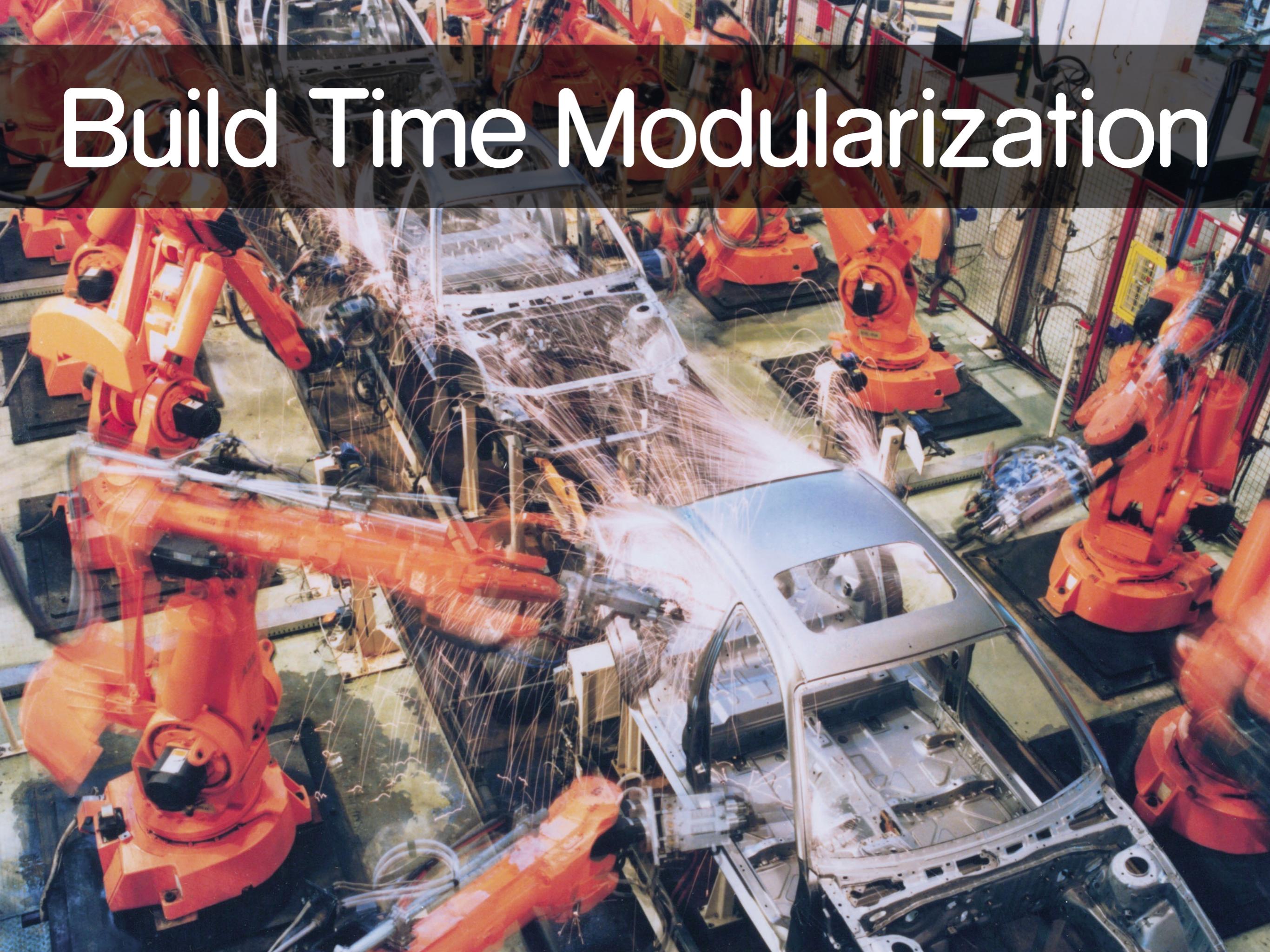


# Micro Frontends

The million dollar question:  
*Modularizing frontend applications.*



# Build Time Modularization



# Code Splitting

A Angular:

```
const routes: Routes = [
  { path: 'customers', loadChildren: './customers/customers.module#CustomersModule' },
  {path: 'orders', loadChildren: './orders/orders.module#OrdersModule' } ]  
  
@NgModule({
  imports: [RouterModule.forChild(routes)],
  ...
});
```

<https://angular.io/guide/lazy-loading-ngmodules>



Angular:

```
const LazyComponent = React.lazy(() => import('./OtherComponent'));
```



React:

```
const LazyComponent = Vue.component('my-comp', () => import('./Foo.vue'))
```

What we get: performance at runtime (incremental loading)

parallel development

independent lifecycles / deployments

runtime isolation

<https://webpack.js.org/guides/code-splitting/>

# Libraries

package.json

```
"dependencies": {  
    "@angular/material": "^6.0.1",  
    "@ngrx/store": "^4.1.1",  
    "my-common-widgets": "^3.1.0",  
    "my-feature-1": "^2.0.1",  
    "my-feature-2": "^7.1.0",  
    ...  
}
```

Libraries can be developed and versioned independently.  
Tools like *yarn workspaces*, *lerna*, *angular cli* or *Nx nrwl* extensions  
can help to manage libraries in a mono repo.

What we get:



parallel development



independent lifecycles / deployments



runtime isolation



Build Time Modularization does not solve the core problems of the frontend monolith.

- lifecycle / releases are coupled
  - no isolation at run time



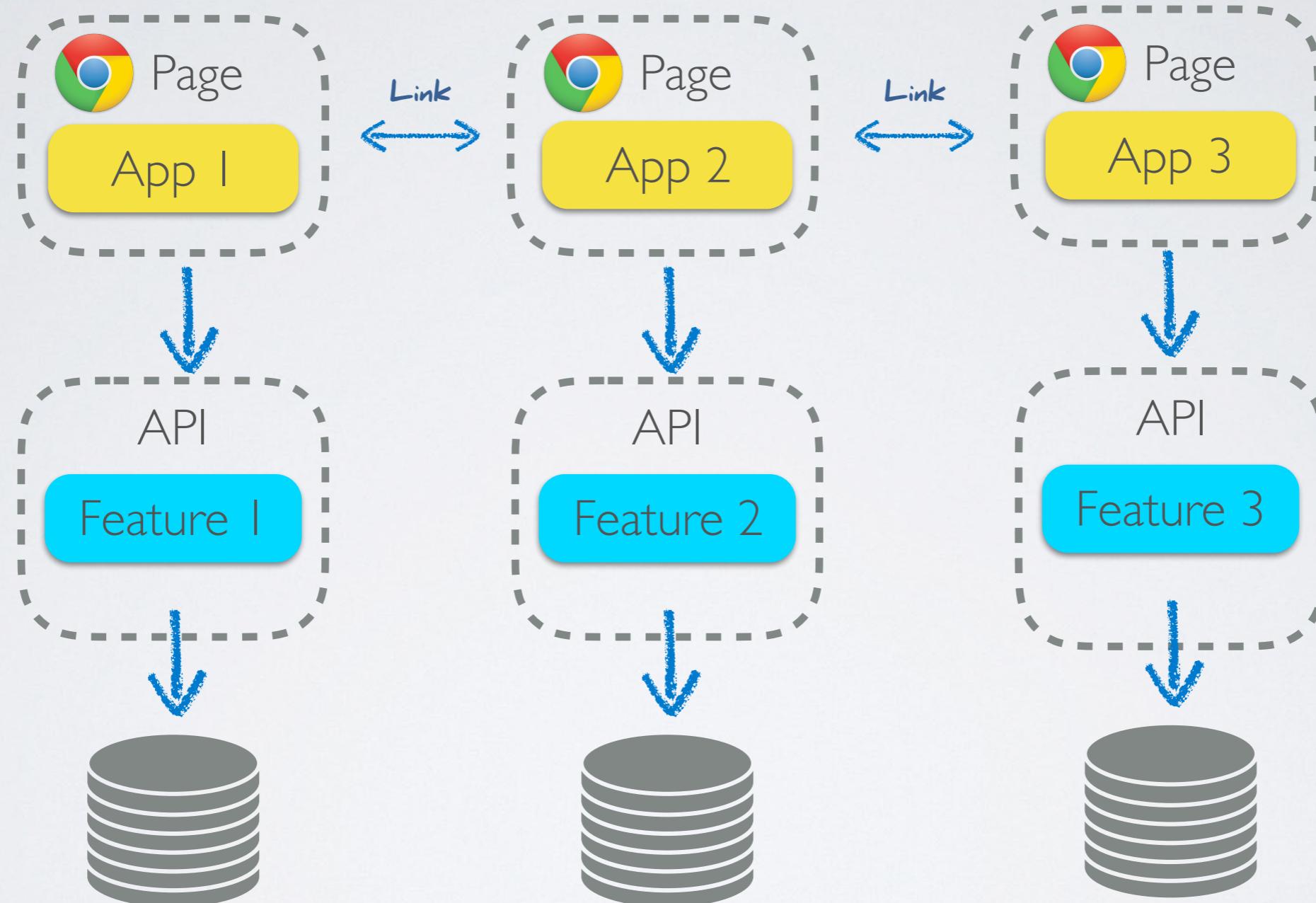
# Runtime Modularization

```
<a href="..." >
```

The Link ...

... the magical integration pattern of the web!

# Microfrontend Flavors: Mini SPAs

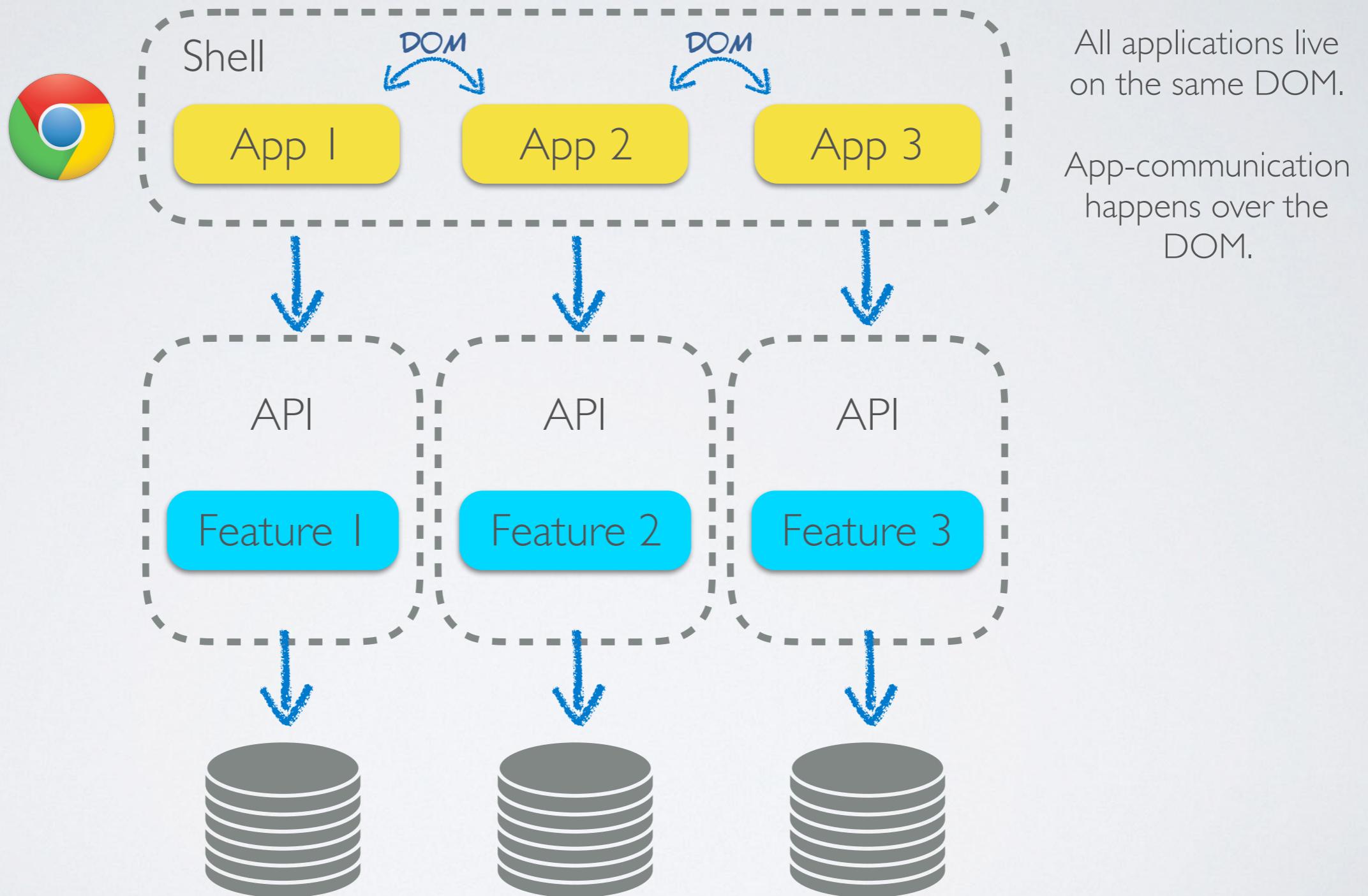


You have complete isolation. But jumping between apps is a full page load!

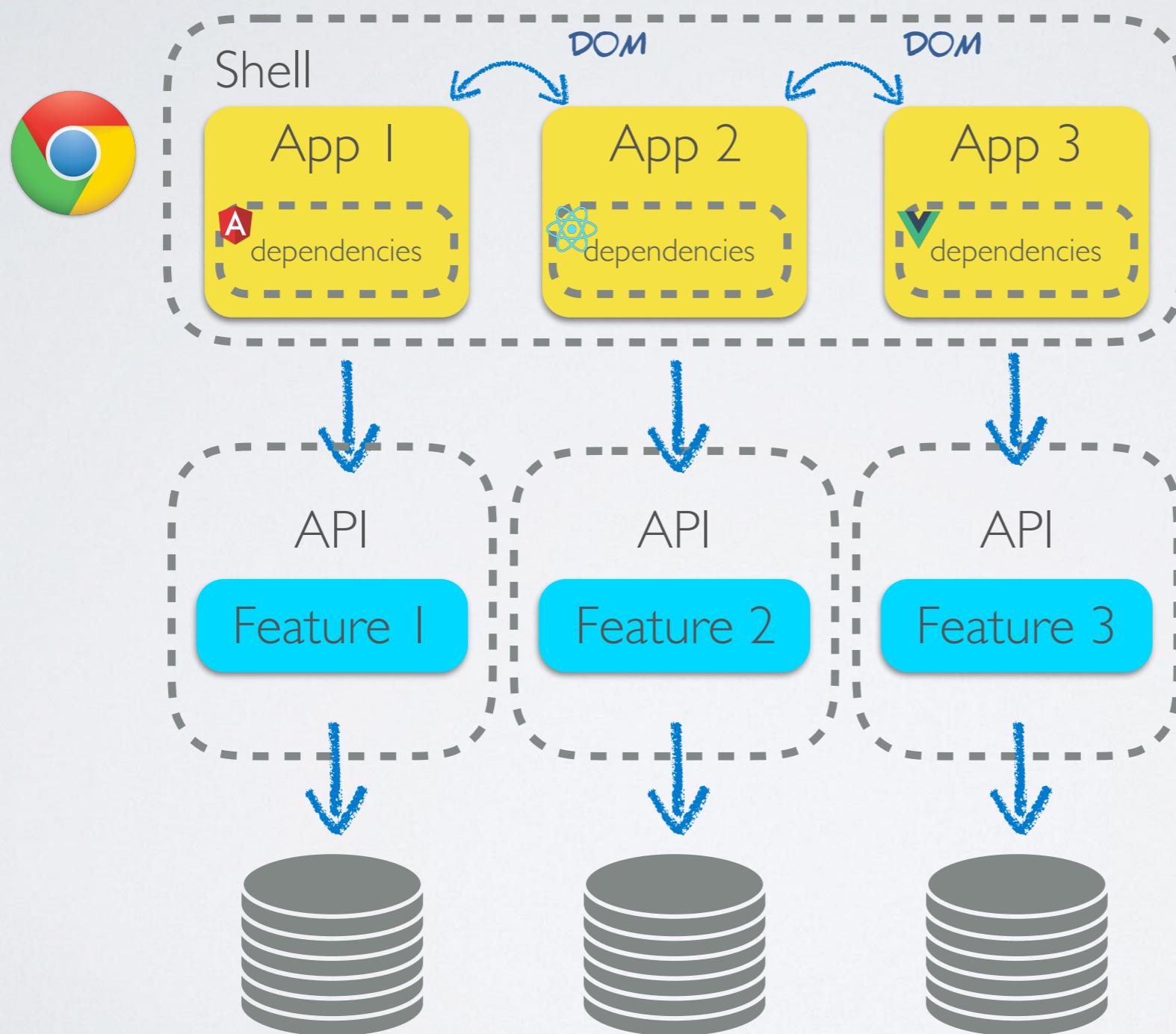


But I need a single SPA!

# Microfrontend Flavors: Single SPA



# Single SPA: One Document



Each application bundles its own dependencies.  
Therefore applications are isolated as long as dependencies don't pollute the global scope.

Note: Angular comes with Zone.js, which heavily pollutes the global scope!

Browser can leverage caching, but for each app all dependencies have to be parsed, compiled and executed.

# DEMO: Single-Spa

<https://single-spa.surge.sh/>

# Shared DOM

```
<html>
  ><head>...</head>
  ><body>
    ><div id="navbar">...</div>
    ><div id="home"></div>
    ><div id="angularjs"></div>
    ><div id="react-app"></div>
...
  ><div id="angular"> == $0
    ><example-ng2-app ng-version="5.1.1">
      ><div>...</div>
      </example-ng2-app>
    </div>
    <div id="vue-app"></div>
    <div id="svelte-app"></div>
    <div id="cycle-app"></div>
    <div id="preact-app"></div>
    <div id="vanillajs"></div>
    <div id="inferno-app"></div>
    <div id="ember-app"></div>
    <div class="hiddendiv common"></div>
  ><div id="sideNav">...</div>
    <div class="drag-target" data-sidenav="mobile-de
  </body>
</html>
```

empty

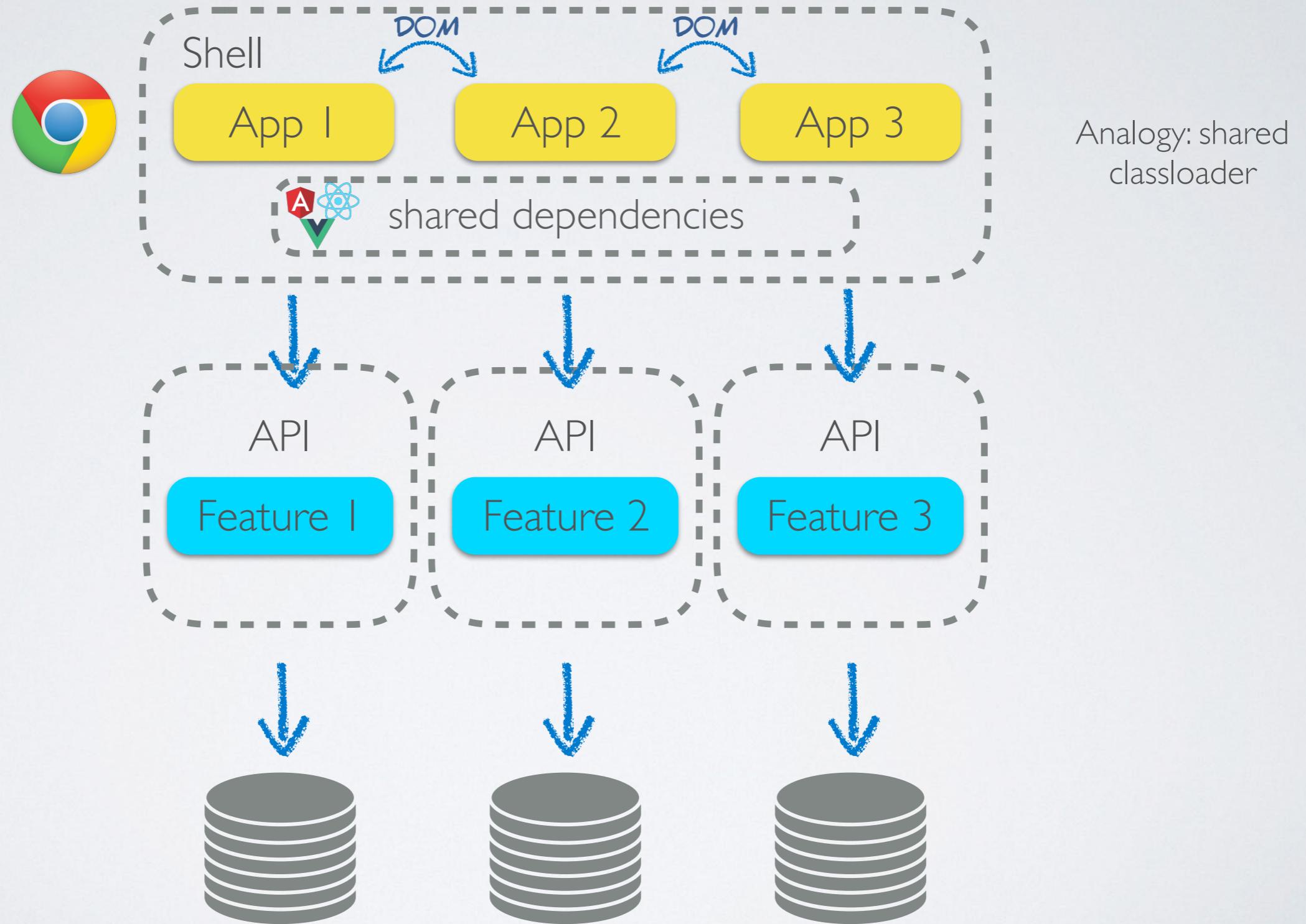


```
<html>
  ><head>...</head>
  ><body>
    ><div id="navbar">...</div>
    ><div id="home"></div>
    ><div id="angularjs"></div>
...
  ><div id="react-app"> == $0
    ><div>
      ><div>
        ><div style="padding: 10px 20px; overflow: h
        ><div class="container">...</div>
      </div>
    </div>
    <div id="angular"></div>
    <div id="vue-app"></div>
    <div id="svelte-app"></div>
    <div id="cycle-app"></div>
    <div id="preact-app"></div>
    <div id="vanillajs"></div>
    <div id="inferno-app"></div>
    <div id="ember-app"></div>
    <div class="hiddendiv common"></div>
  ><div id="sideNav">...</div>
    <div class="drag-target" data-sidenav="mobile-de
  </body>
</html>
```

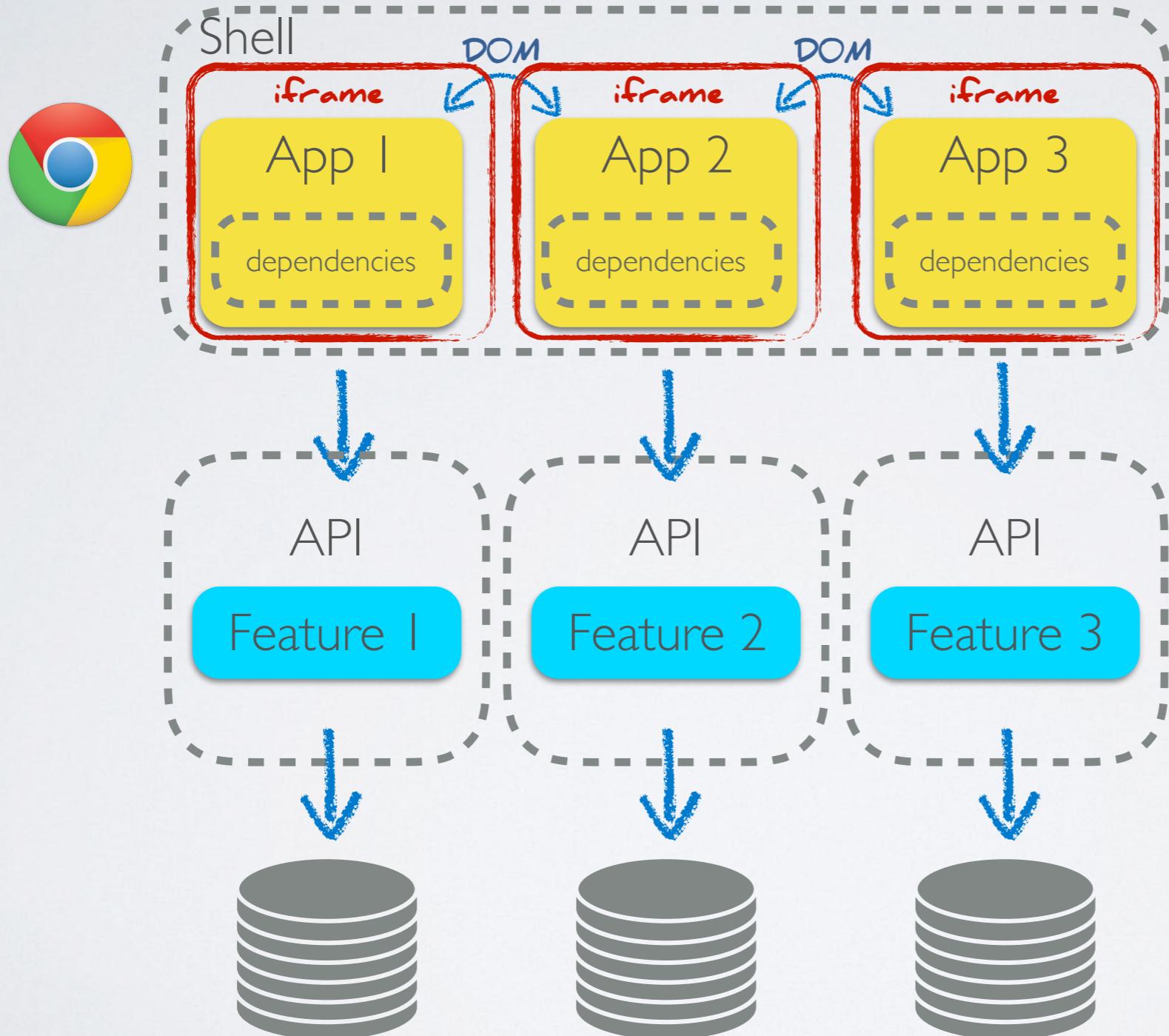
empty



# Optimization: Sharing Dependencies



# Single SPA: iframes



Analogy: complete classloader isolation.

Browser can leverage caching, but for each app all dependencies have to be parsed, compiled and executed.

Apps are constrained to the iframe, which results in challenges for layout (resizing, dialogs, tooltips ...)



There is no Silver Bullet for  
Frontend Modularization



WebComponents ... Rescue?

Google jb

Secure | https://www.google.com

Gmail Images

# Google

are web component

are web components **dead**  
are web components **ready**  
are web components **the future**

[Learn more](#)

Google offered in: Deutsch Français Italiano Rumantsch

Switzerland

Advertising Business About Privacy Terms Settings



# Web Components

WebComponents is a series of browser standards for creating reusable custom elements:

Shadow DOM	Encapsulation
HTML templates	markup that is not initially rendered and can be instantiated.
Custom elements	JavaScript API to define custom elements that can be used in html and their behavior

Custom Elements API:

```
customElements.define('my-component', class extends HTMLElement { ... })
```

JavaScript

html:

```
<div> <my-component></my-component> </div>
```

Framework integrations:



Angular elements



Vue & React wrappers

Browser support:

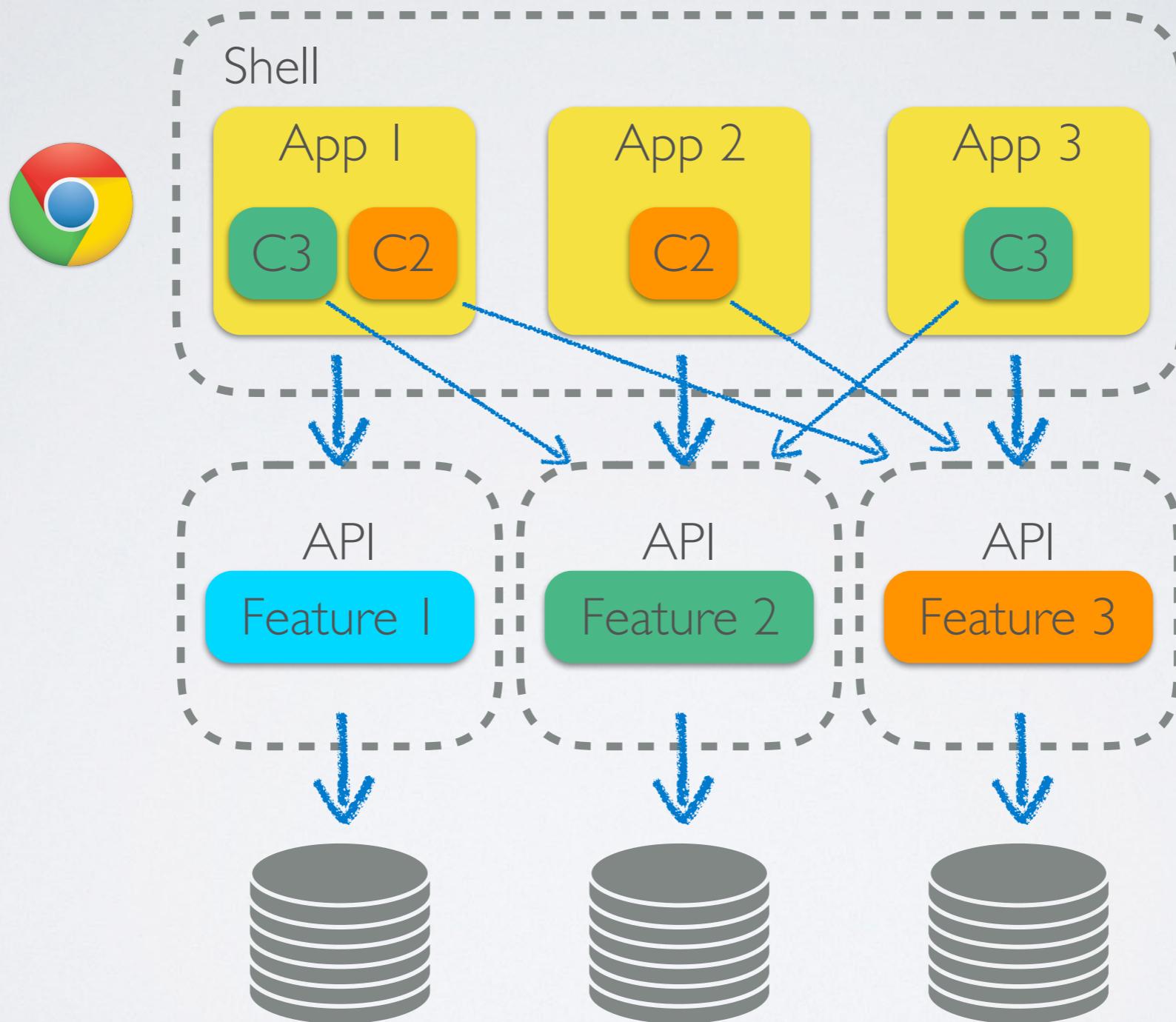
native: Chrome, Safari

polyfill: Firefox, Edge, IE11

[https://developer.mozilla.org/en-US/docs/Web/Web\\_Components](https://developer.mozilla.org/en-US/docs/Web/Web_Components)

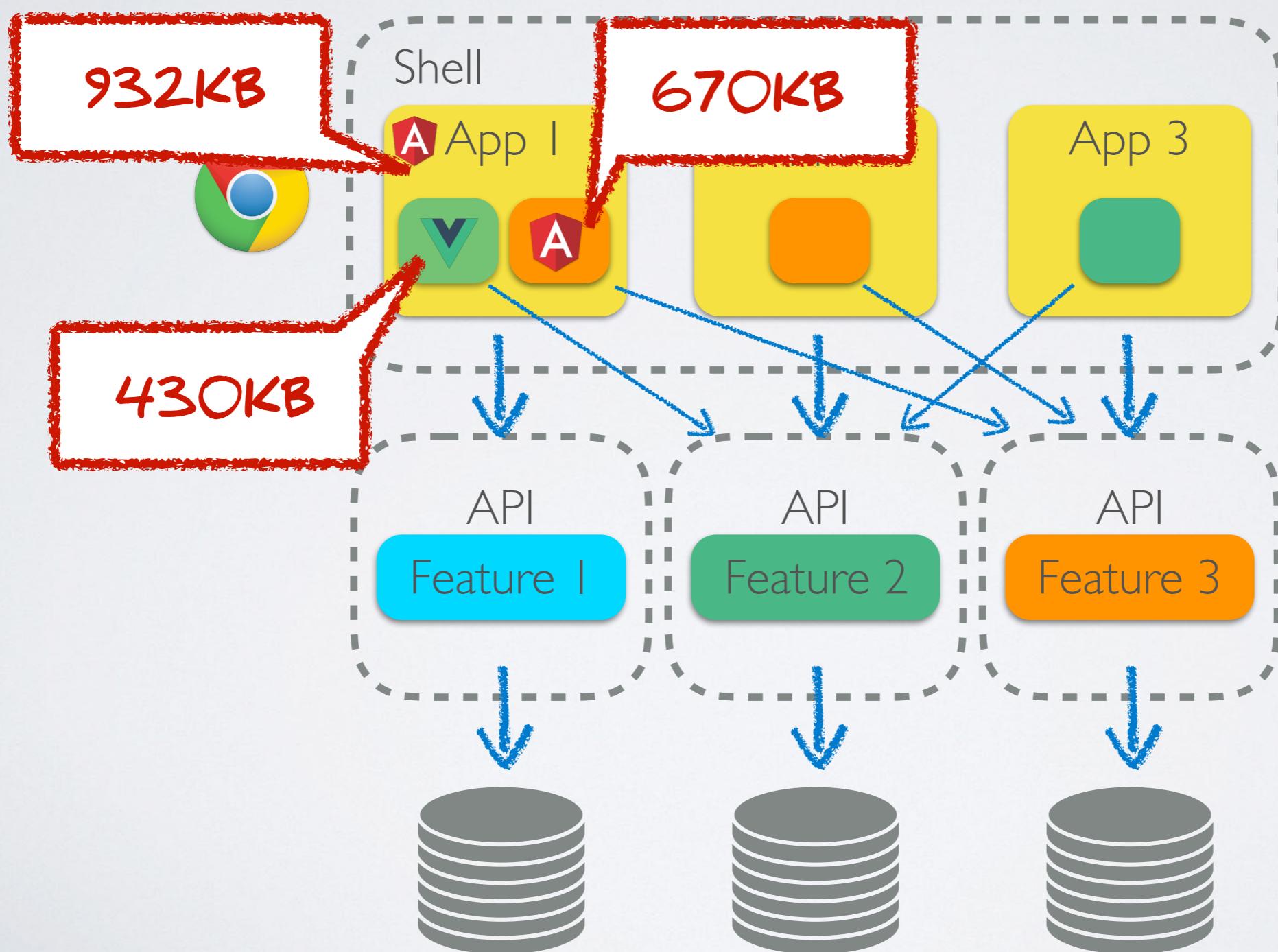
<https://developer.mozilla.org/en-US/docs/Web/API/Window/customElements>

# Promise: Frontend Composition

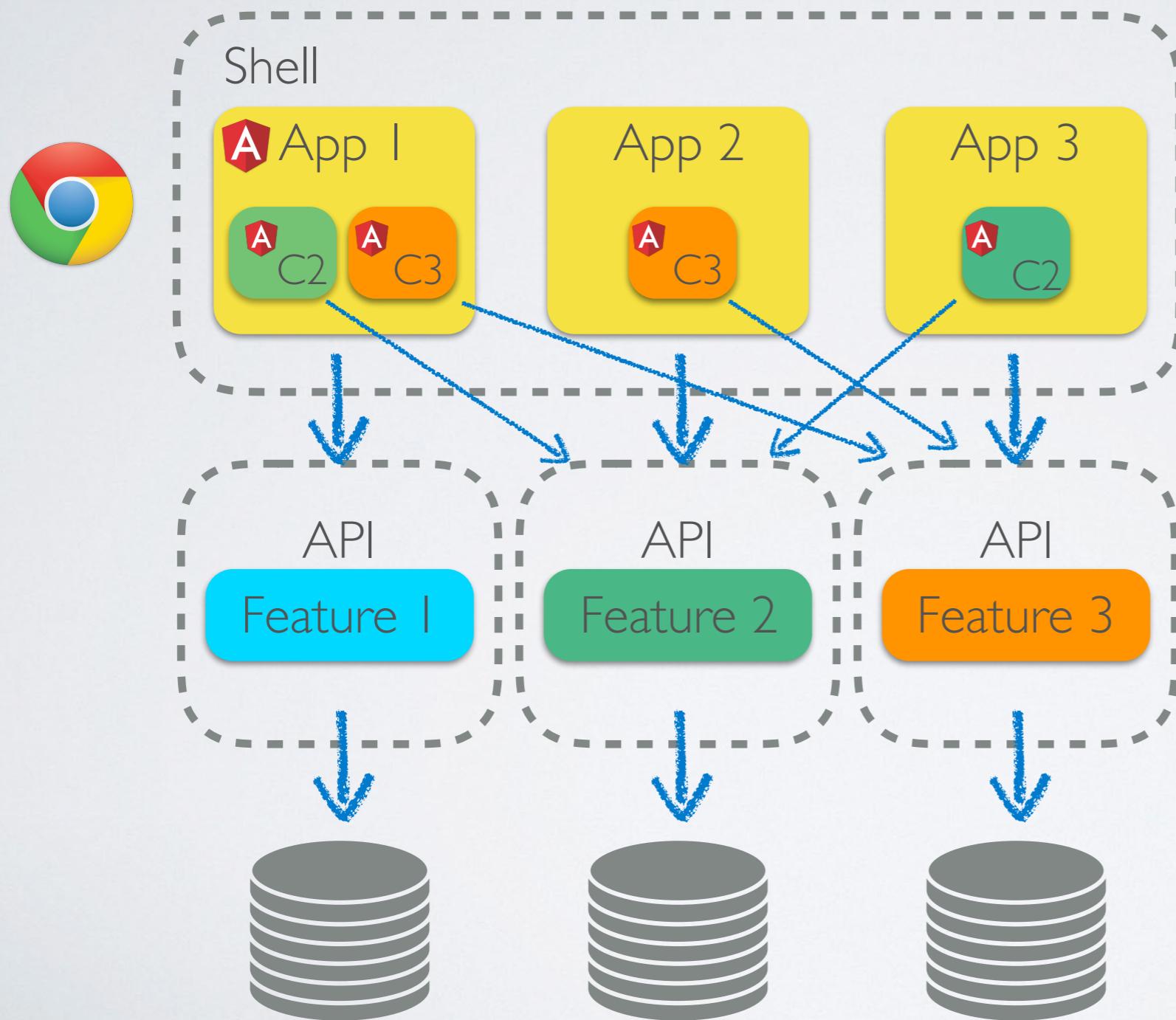


# Size Does Matter!

If a component should be autonomous,  
it must bring it's own dependencies!



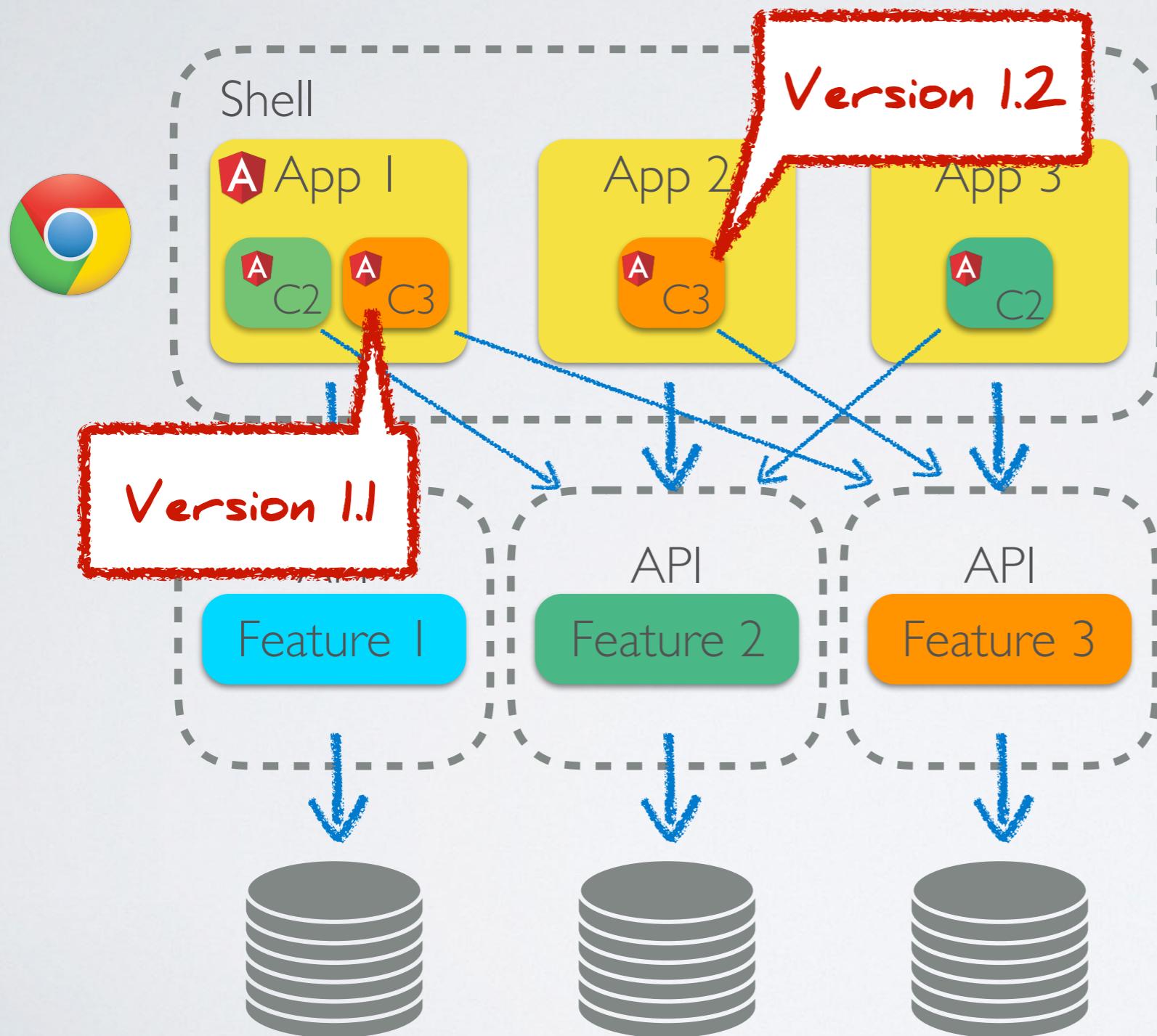
# WebComponents do not provide JavaScript isolation!



The Shadow DOM only provides encapsulation for styling not for JavaScript!  
There is still one single global scope per document.

- There is no guarantee that the components do not have side-effects on each other.

# WebComponents are global!



WebComponents are registered on the document by name.

```
customElements.define(  
  'my-component',  
  class extends HTMLElement { ... })
```

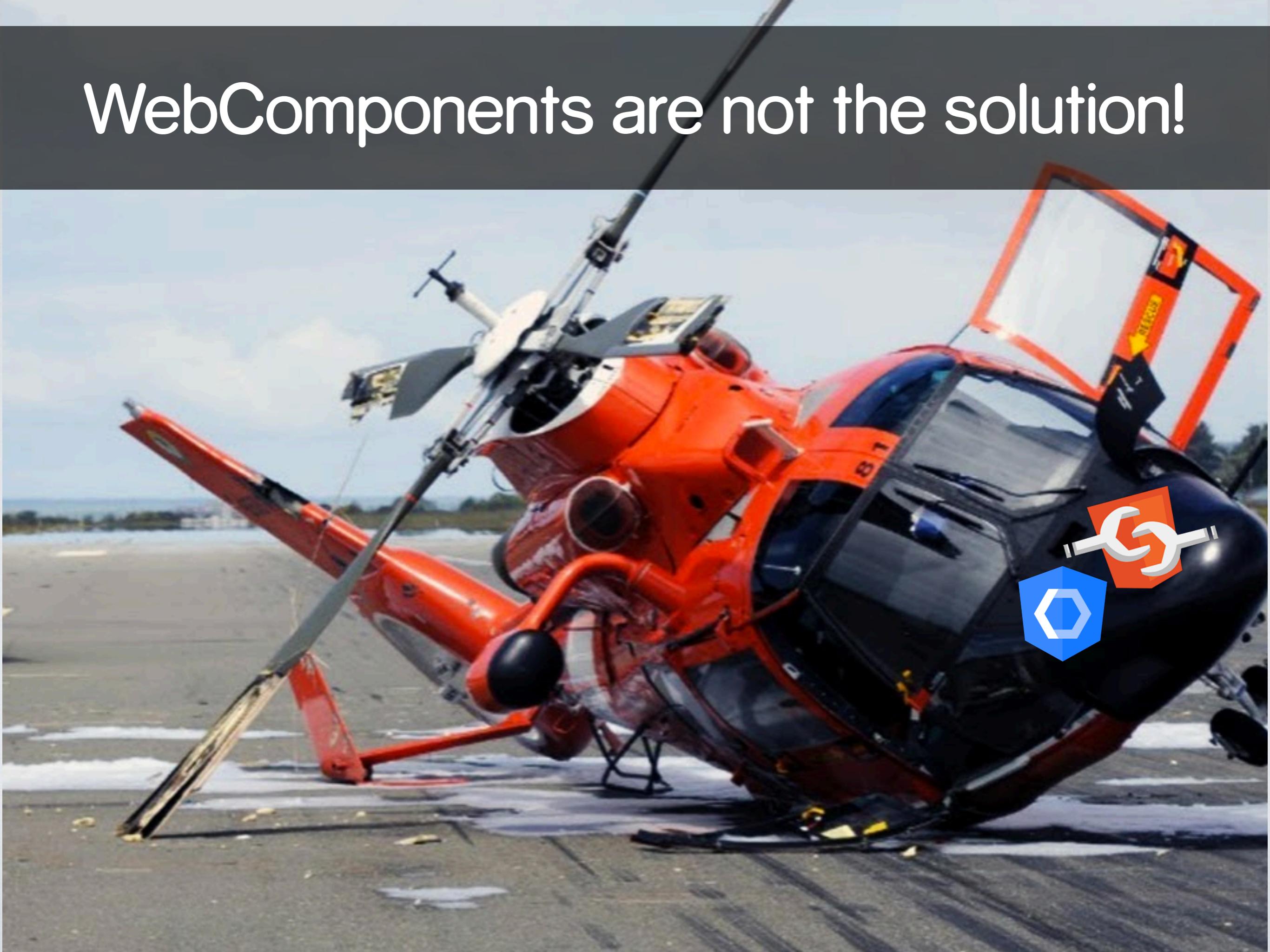
All applications must use the same version of a given component!

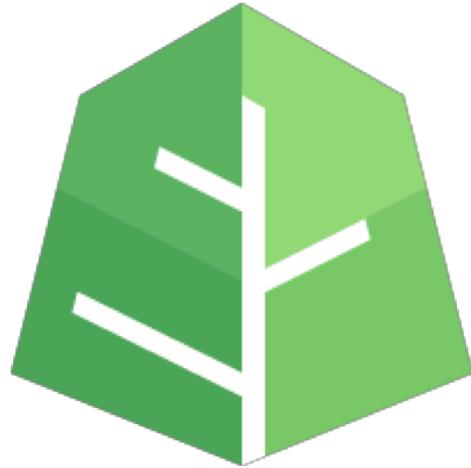
# Web Components: The Reality

What we get:

- ✓ simple consumption/embedding
- ✓ parallel development (when packaged as a library)
- ✗ independent lifecycles / deployments
- ✗ runtime isolation (Shadow DOM only isolates styling)

# WebComponents are not the solution!





# scion workbench

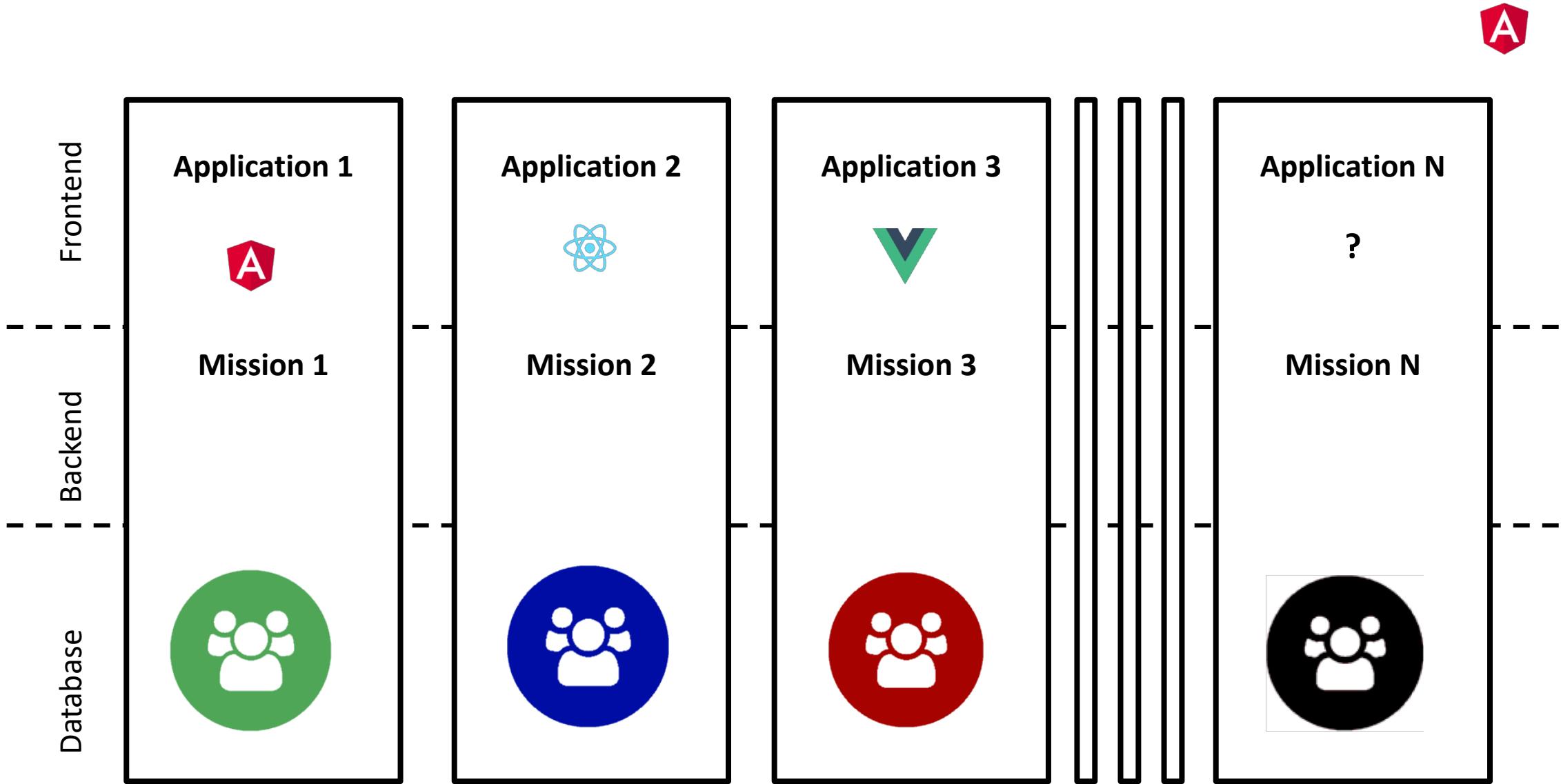


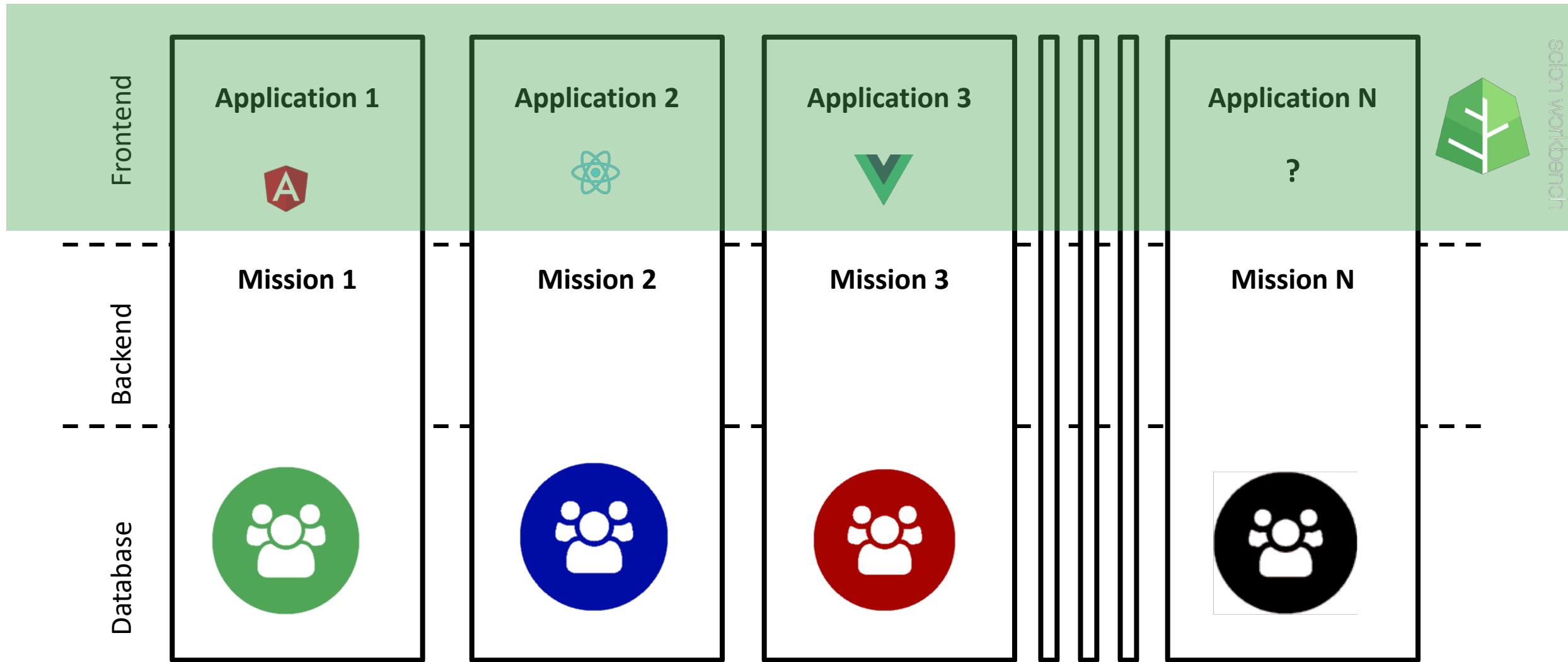
<https://github.com/SchweizerischeBundesbahnen/scion-workbench>



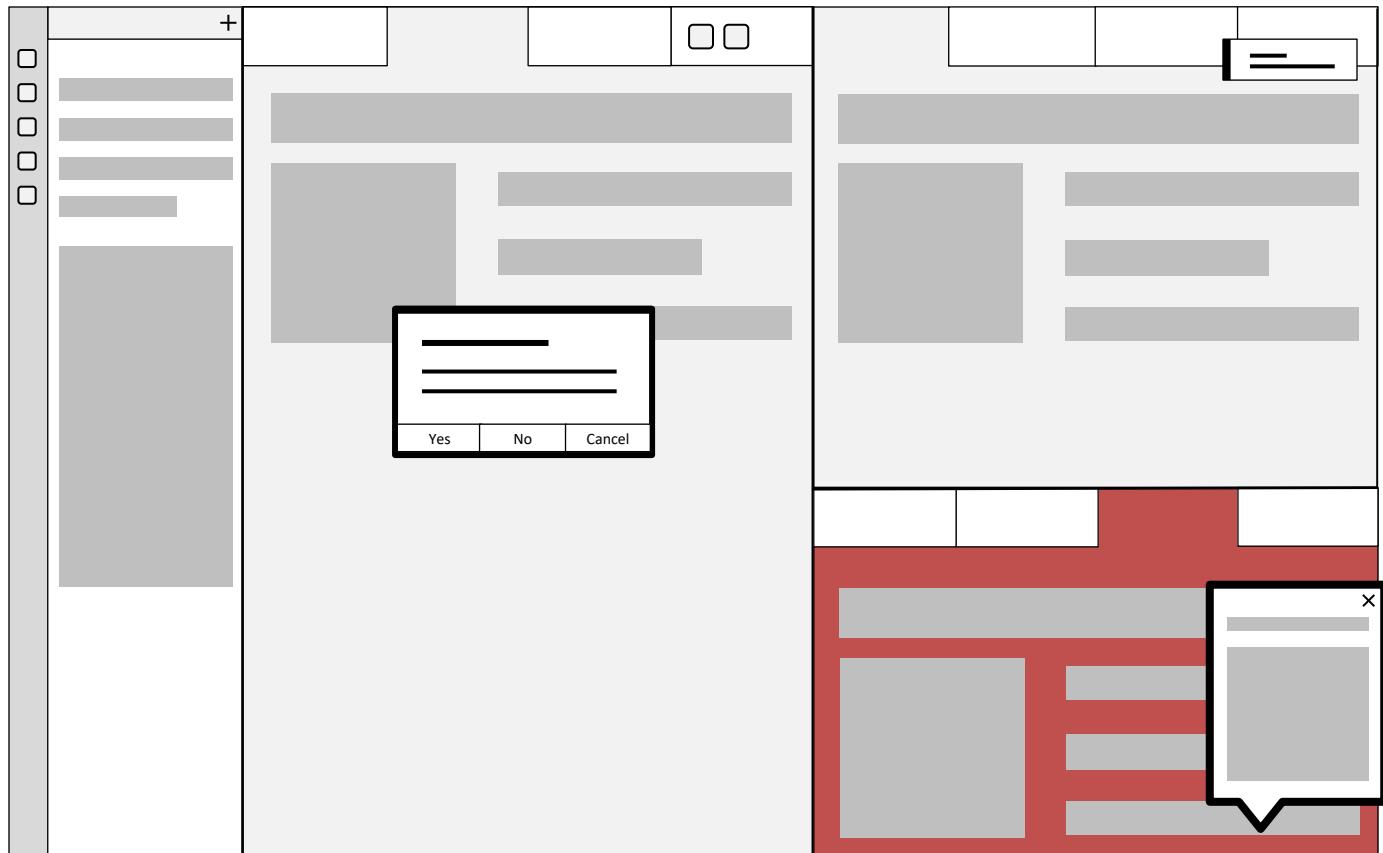
SBB CFF FFS

mtrail





[https://www.app.com/#/\(view.1:person/39//view.2:person/38\)?viewgrid=WyJ2aWV3cGFydC4xliw](https://www.app.com/#/(view.1:person/39//view.2:person/38)?viewgrid=WyJ2aWV3cGFydC4xliw)



# SCION Workbench



Provides a **lightweight application frame** and helps to build **multi-view Angular applications**.

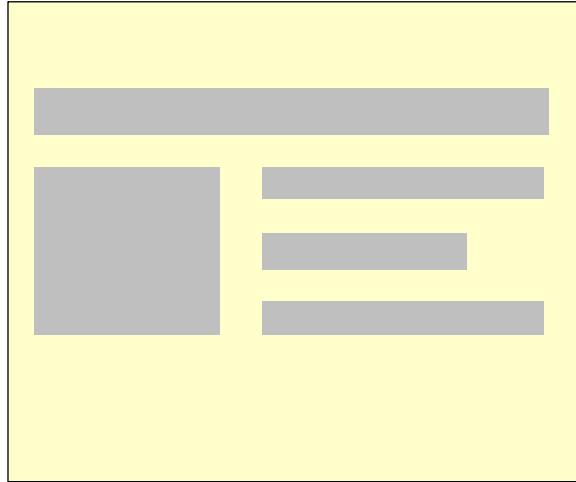
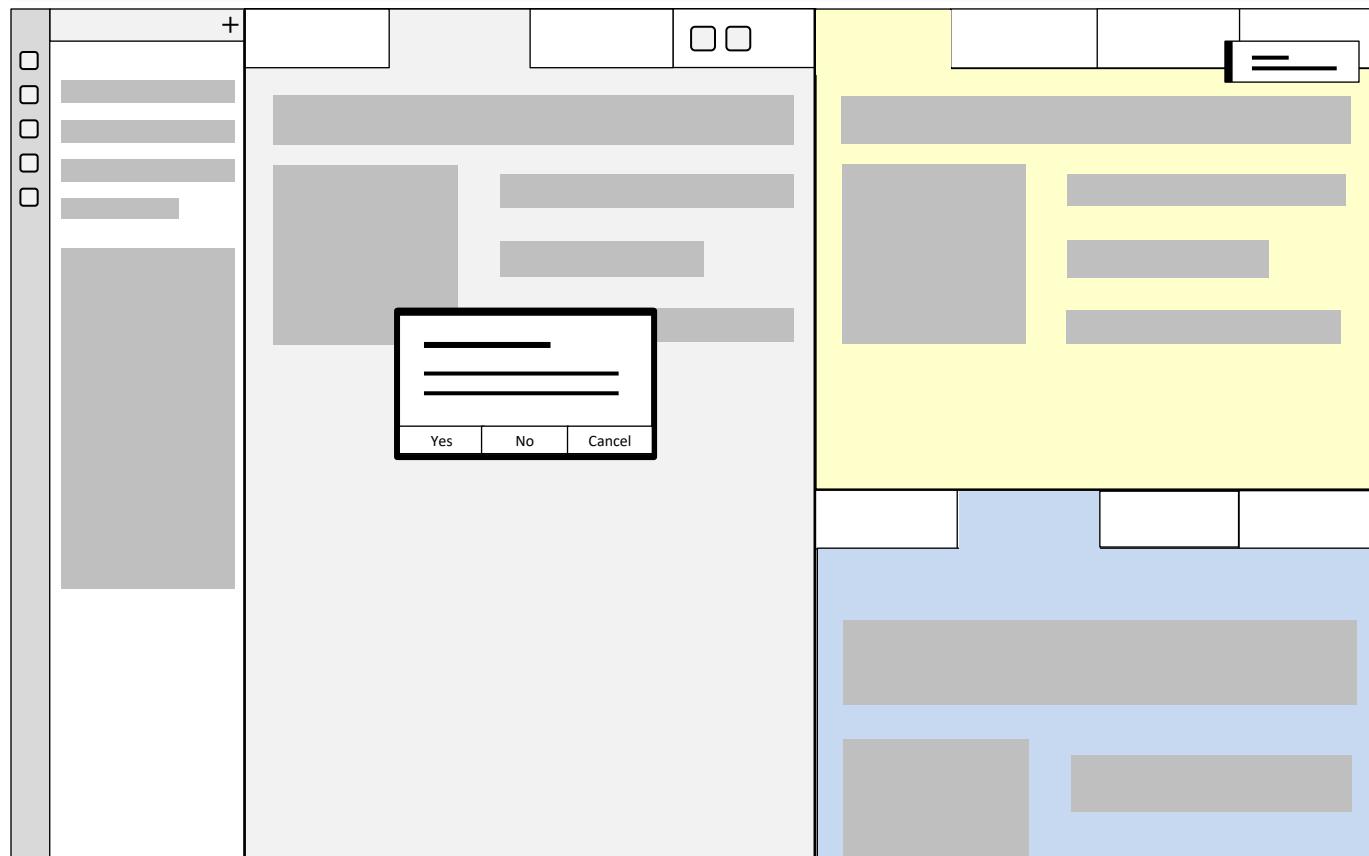
The screenshot displays a multi-view application built with SCION Workbench. It includes:

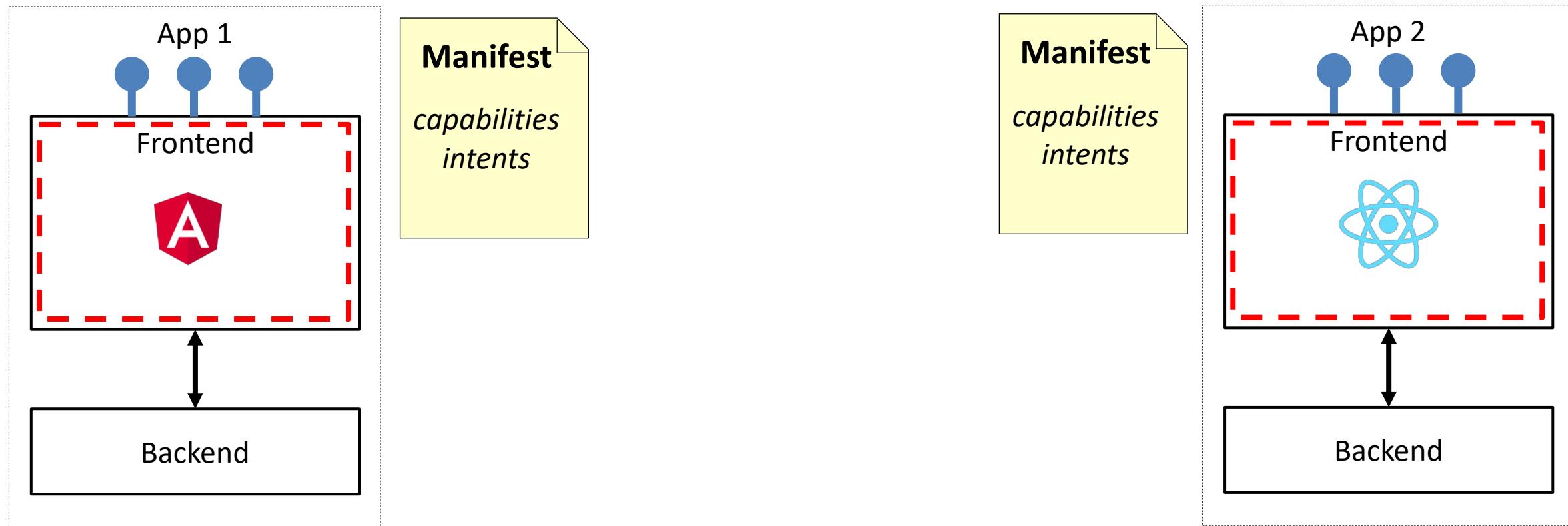
- A left sidebar titled "PERSONS" listing various people with their names and locations.
- A central "Person" view for "Adeline McEnhill" showing a profile picture and a "PEXELS" watermark.
- A "Person" view for "Clarke Noden" with fields for Firstname, Lastname, Street, City, Email, Phone, and Profession, also featuring a "PEXELS" watermark.
- A "Friends" section where users can enter friend names.
- A "Heatmap" view showing a map of North America with colored dots indicating the density of data points across the United States, Canada, and Mexico.
- A "Person" view for "Alaster Sancto" showing a profile picture and a "PEXELS" watermark.



<https://github.com/SchweizerischeBundesbahnen/scion-workbench>

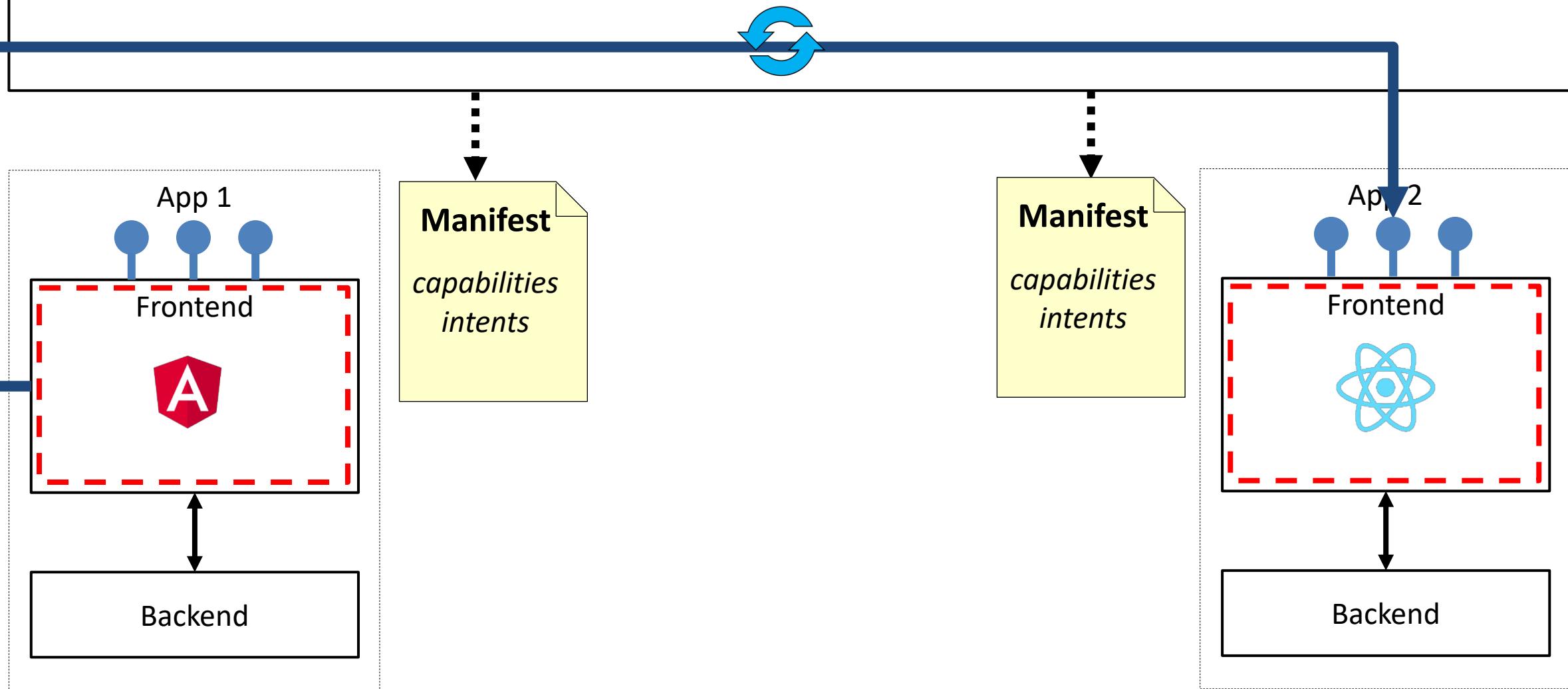
[https://www.app.com/#/\(view.1:person/39//view.2:person/38\)?viewgrid=WyJ2aWV3cGFydC4xliw](https://www.app.com/#/(view.1:person/39//view.2:person/38)?viewgrid=WyJ2aWV3cGFydC4xliw)





# Intent Router

Workbench Application Platform in Host App



## MANIFEST of App 1

```
"intent": {  
  "type": "view",  
  "qualifier": {  
    entity: "person",  
    id: "*"  
  }  
}
```

## MANIFEST of App 2

```
"capability": {  
  "type": "view",  
  "qualifier": {  
    entity: "person",  
    id: "*"  
  },  
  "properties": {  
    "path": "/persons/:id",  
  }  
}
```

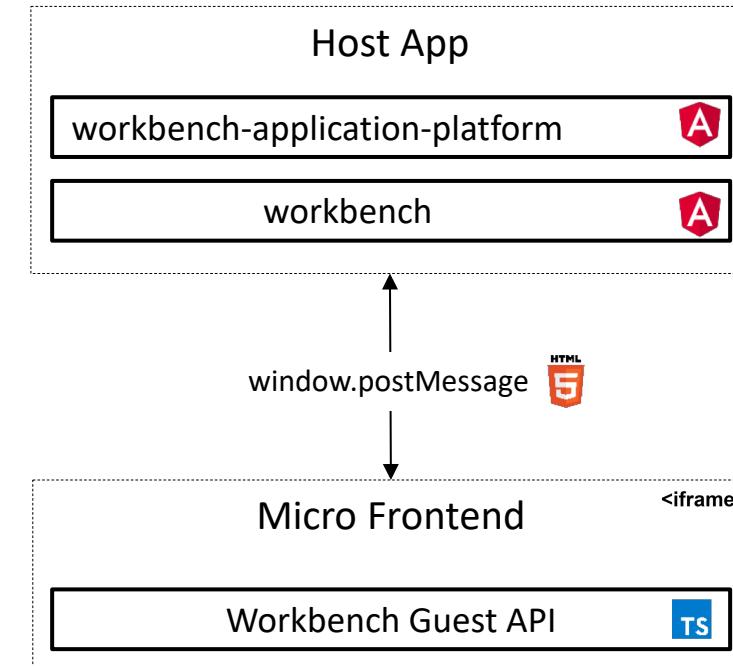
<a [wbRouterLink]={"entity: 'person', id: '5' }">Open Person</a>

# SCION Workbench Application Platform



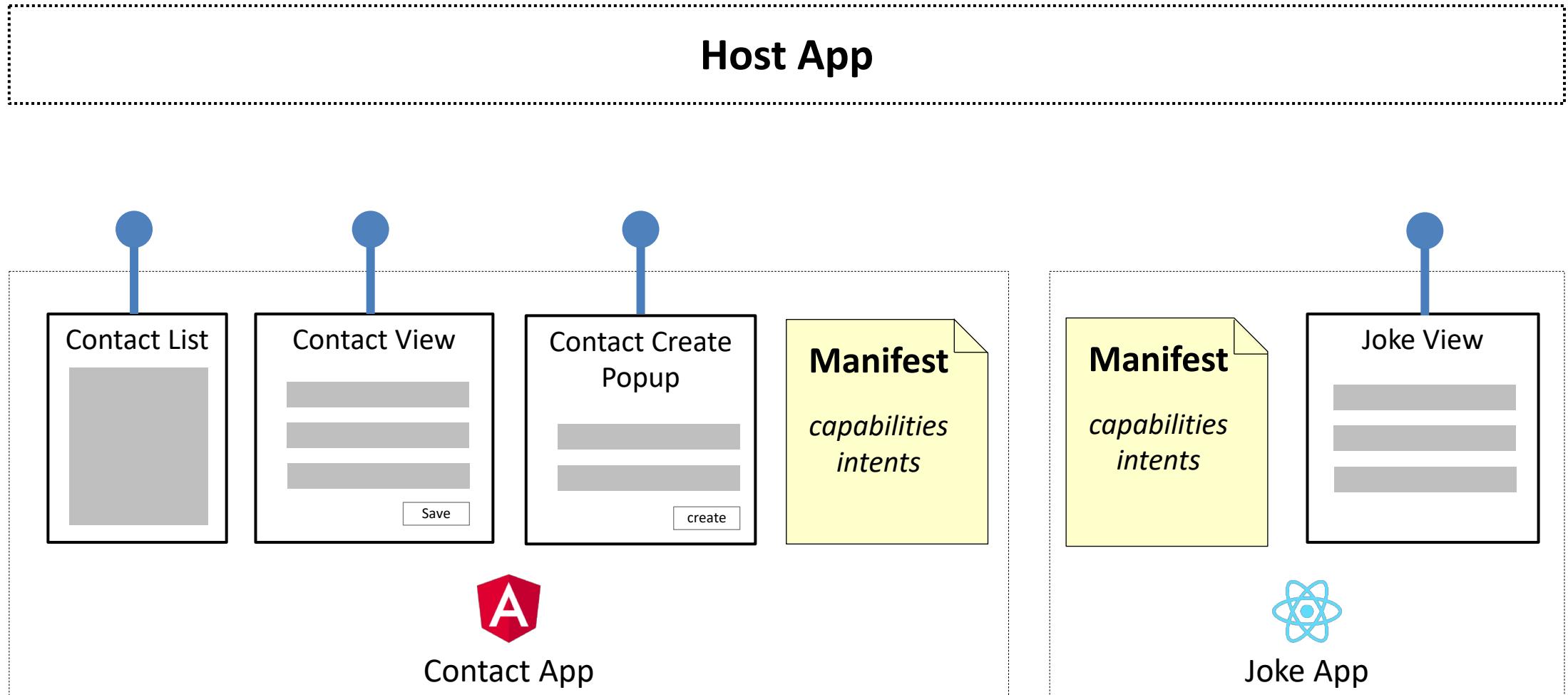
Enables a micro frontend architecture by  
**integrating content from multiple web applications in a coherent way.**

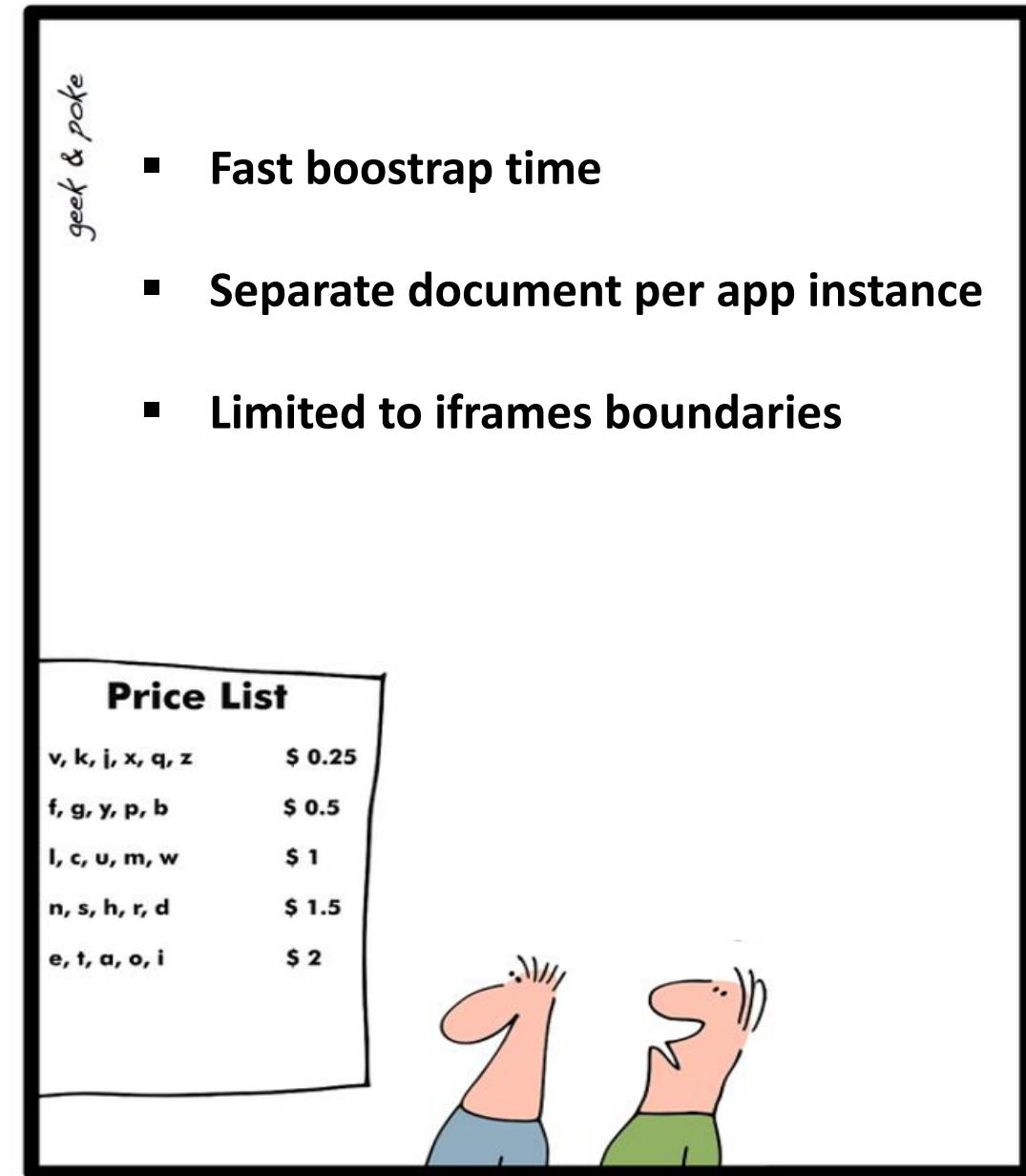
The screenshot shows a contacts application interface. On the left, a list of contacts is displayed. In the center, a detailed view for 'Alvin Zambon' is shown, including fields for Firstname, Lastname, Street, City, Email, and Phone. Below this, a 'New communication' modal is open, prompting for a recipient and message content. A red dashed box highlights this communication interface.



<https://github.com/SchweizerischeBundesbahnen/scion-workbench>

# SCION Workbench Application Platform Live Hacking









# scion workbench



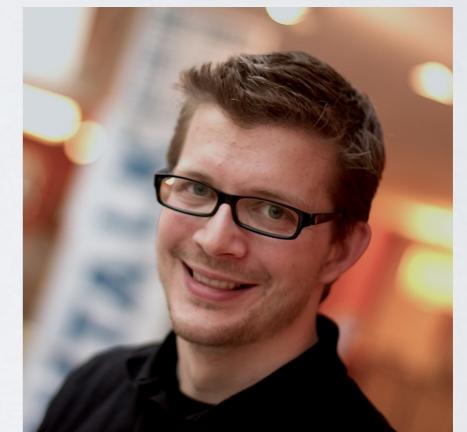
<https://github.com/SchweizerischeBundesbahnen/scion-workbench>

# Questions?

<https://github.com/jbandi/voxxed-days-zuerich-2019>



Daniel Wiehl  
mtrail AG  
SCION Workbench  
[daniel.wiehl@mtrail.ch](mailto:daniel.wiehl@mtrail.ch)



Jonas Bandi  
JavaScript / Angular / React  
Courses, Consulting, Reviews,  
Project-Setup, Proof-of-Concept  
[jonas.bandi@ivorycode.com](mailto:jonas.bandi@ivorycode.com)