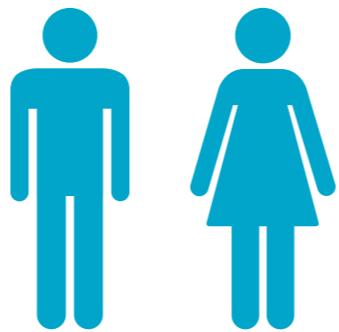


# Micro Frontends

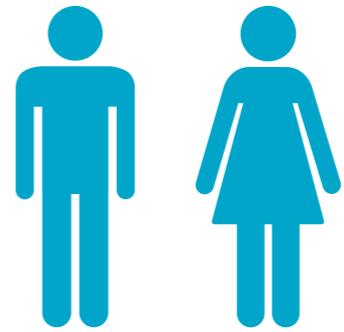
Extending the Micro Services  
Architecture to Web Development

Jan 2018 | Frontend Developer Love @ Usabilla

Monolith



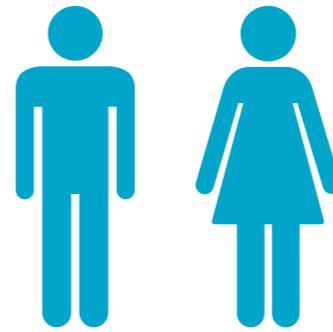
Monolith



- + Single codebase
- + Easy quick setup
- + No version management
- + Single CI/CD pipeline

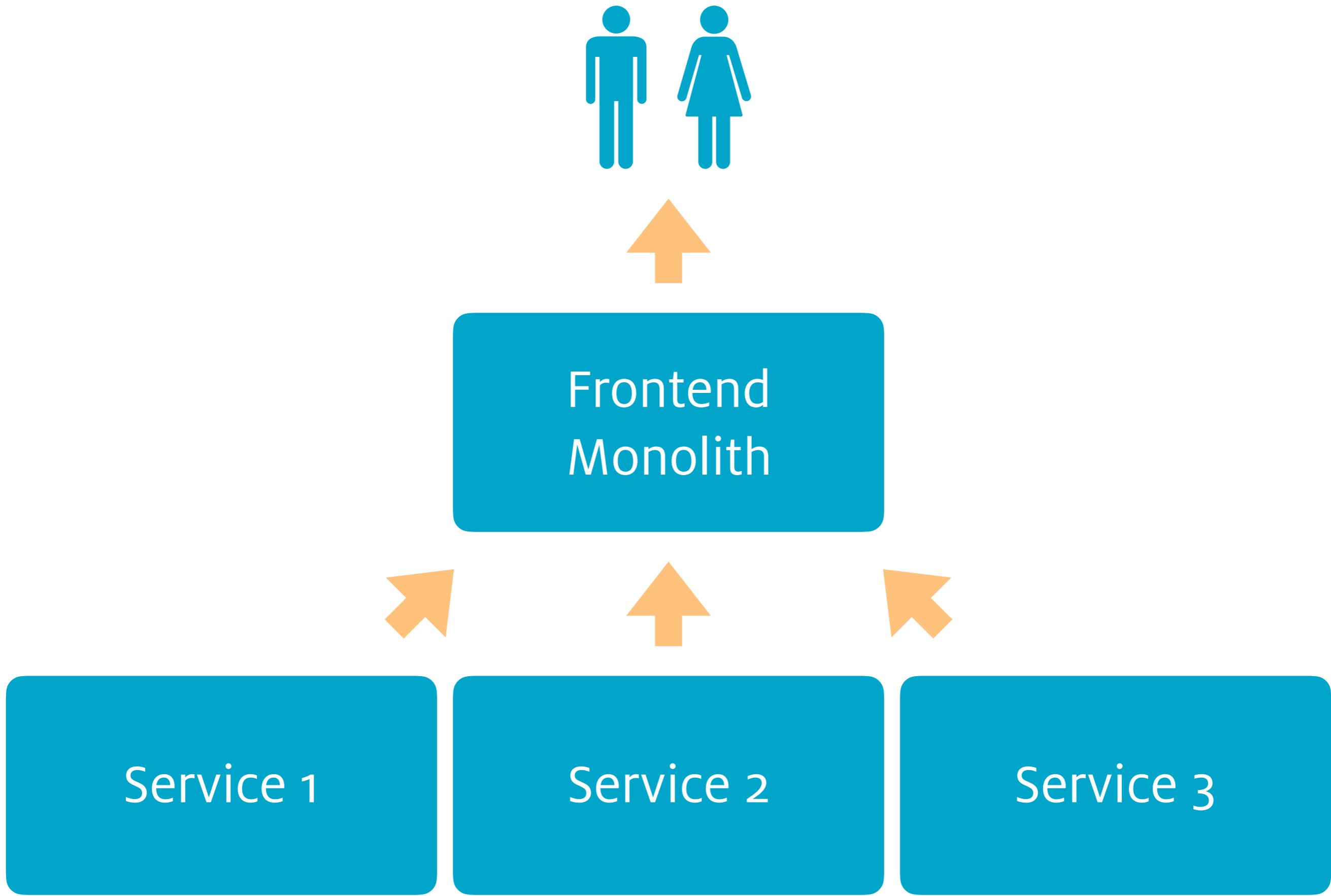
Monolith

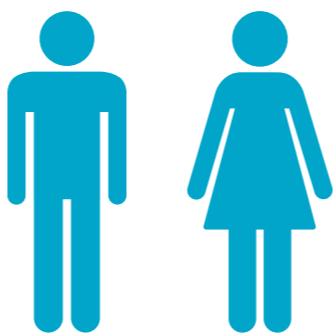
- + Single codebase
- + Easy quick setup
- + No version management
- + Single CI/CD pipeline



- Refactoring is difficult
- Try new stack
- Experiment
- Long build times
- Long development times
- Long deployment cycles

Monolith





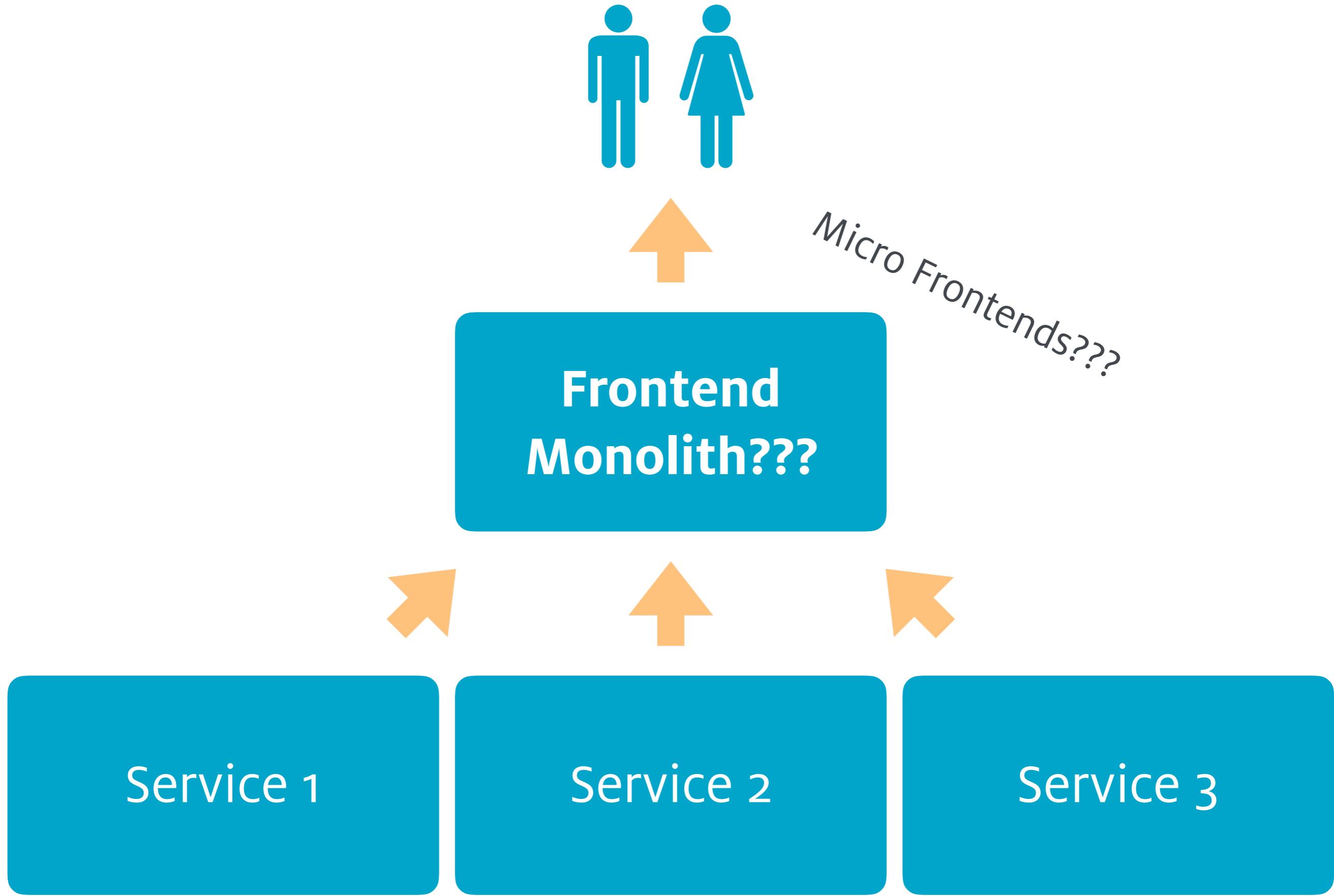
Frontend  
Monolith???



Service 1

Service 2

Service 3





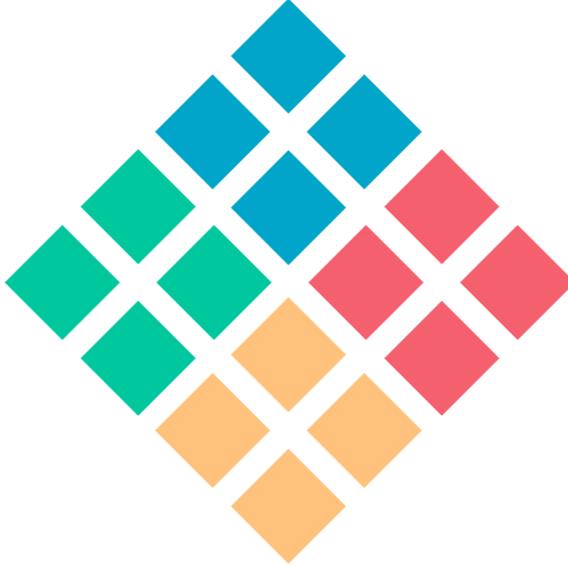
# Spyros loakeimidis

## Software Engineer



[github](#)  
[@spirosikmd](#)

[twitter](#)  
[@spyros\\_io](#)



# Chapter 1: What? Why?



## Technology Radar

### Techniques

#### Micro frontends

TRIAL ?

<https://www.thoughtworks.com/radar/techniques/micro-frontends>



● micro frontends

Search term

Oct 2 - Oct 8 2016

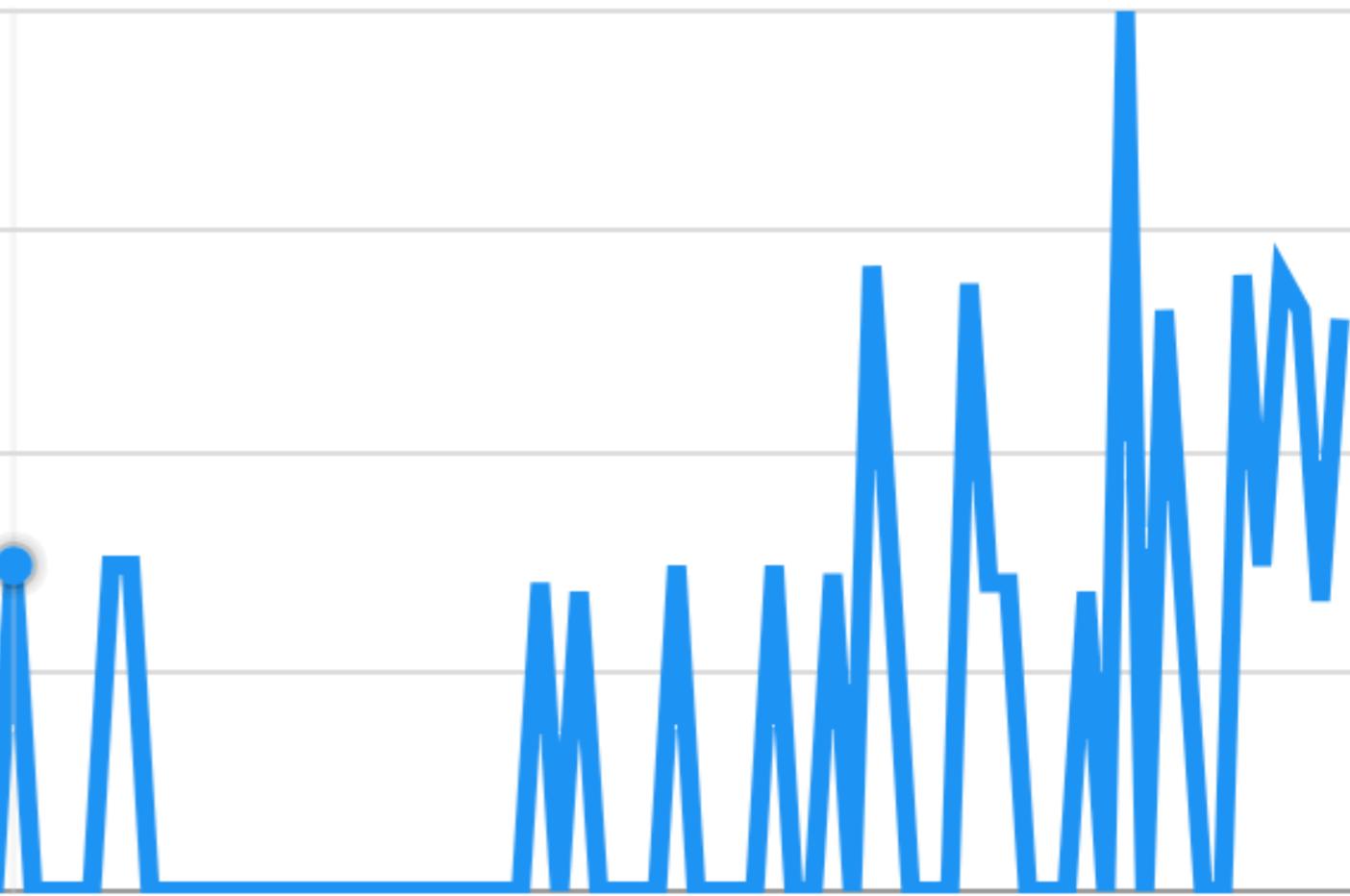
micro frontends

37

Note

Jan 31, 2016

Jul 30, 2017





**svdberg**

@svdberg

Follow



Interesting post about micro frontends. What  
is your opinion?  
[blog.xebia.com/2015/07/27/the ...](http://blog.xebia.com/2015/07/27/the ...)

9:36 AM - 28 Jul 2015

**2** Retweets **1** Like



2



1



Tweet your reply

<https://twitter.com/svdberg/status/625932935204610048>



mir

@mircealungu

Follow



i don't know #microfrontends are, and i hope that you and I can retire without ever finding out.

12:01 AM - 30 Dec 2017

1 Like

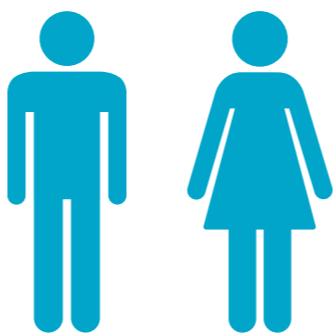


1



Tweet your reply

<https://twitter.com/mircealungu/status/946878978463289344>



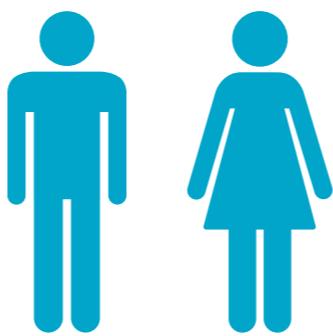
Frontend  
Monolith

Service 1

Service 2

Service 3





Frontend 1

Frontend 2

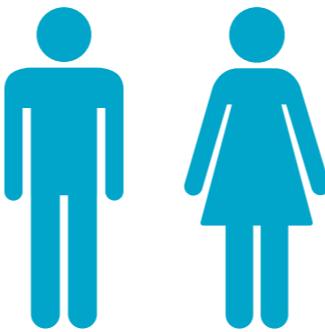
Frontend 3

Service 1

Service 2

Service 3

- > Reusable Code
- > Agility
- > Reduce risk
- > Easier Testing



Frontend 1

Frontend 2

Frontend 3

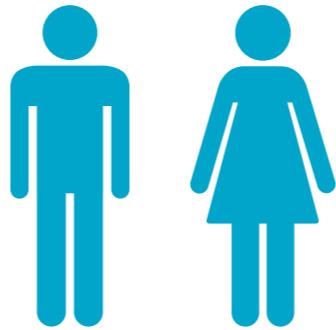


Service 1

Service 2

Service 3

- + Verticalised teams
- + Self contained
- + Vertical decomposition
- + UI Composition



Frontend 1

Frontend 2

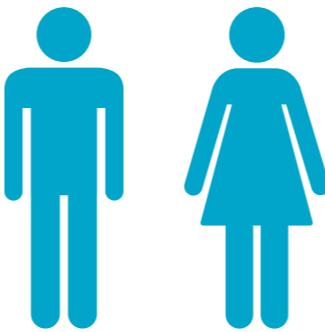
Frontend 3

Service 1

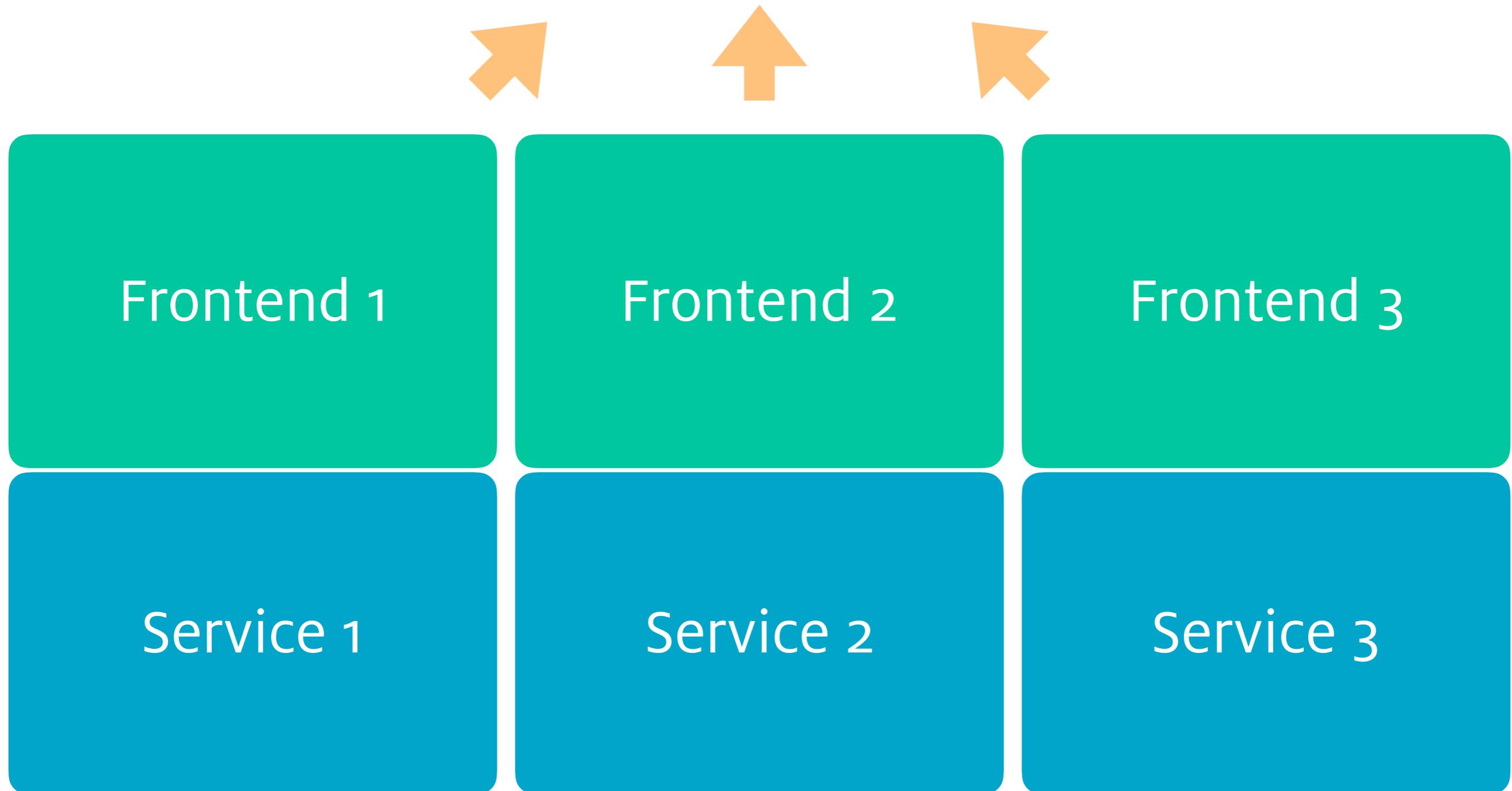
Service 2

Service 3

- + Verticalised teams
- + Self contained
- + Vertical decomposition
- + UI Composition



- Proprietary solution
- Fit current architecture
- Load times
- Redundancy risk



Reporting

Security

---

## Single Responsibility

---

Account  
Management

Interactions

Reporting

Security

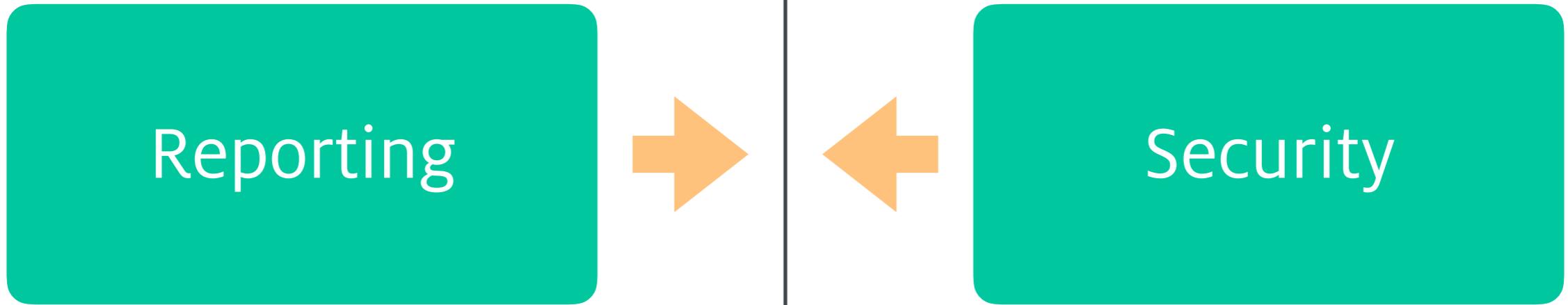
---

## Consistent Look & UX

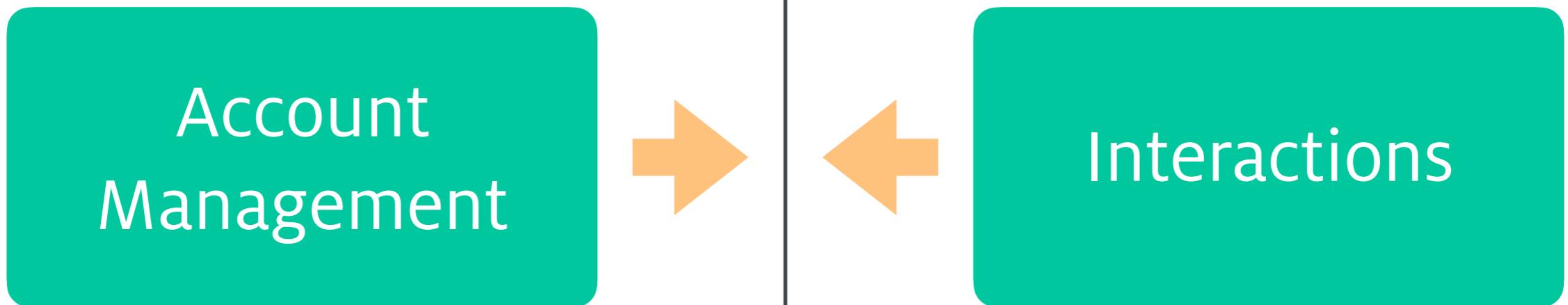
---

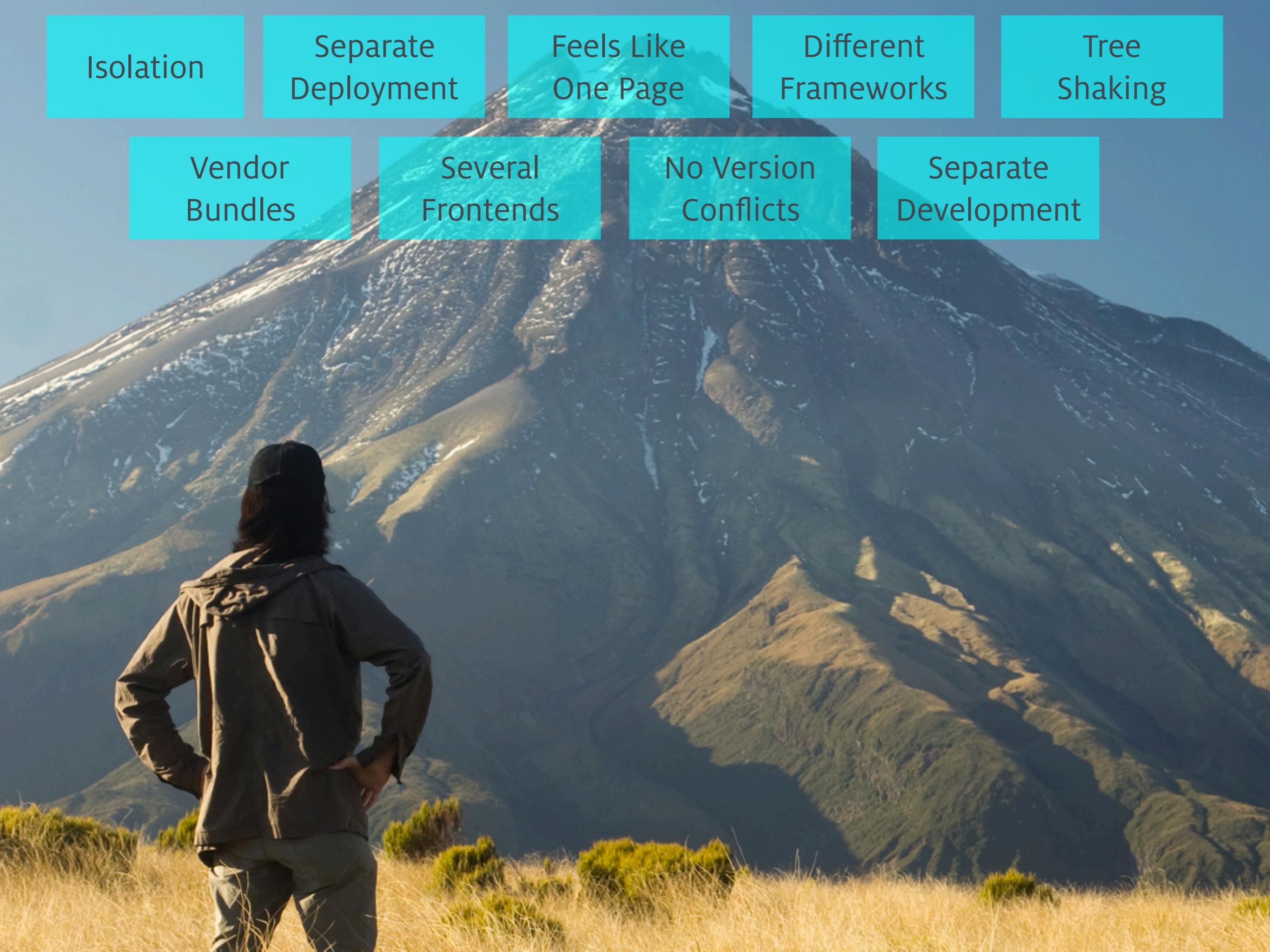
Account  
Management

Interactions



Common  
Interface



A photograph of a person from behind, wearing a dark jacket and backpack, standing on a grassy hillside and looking towards a massive, rugged mountain range under a clear blue sky.

Isolation

Separate Deployment

Feels Like One Page

Different Frameworks

Tree Shaking

Vendor Bundles

Several Frontends

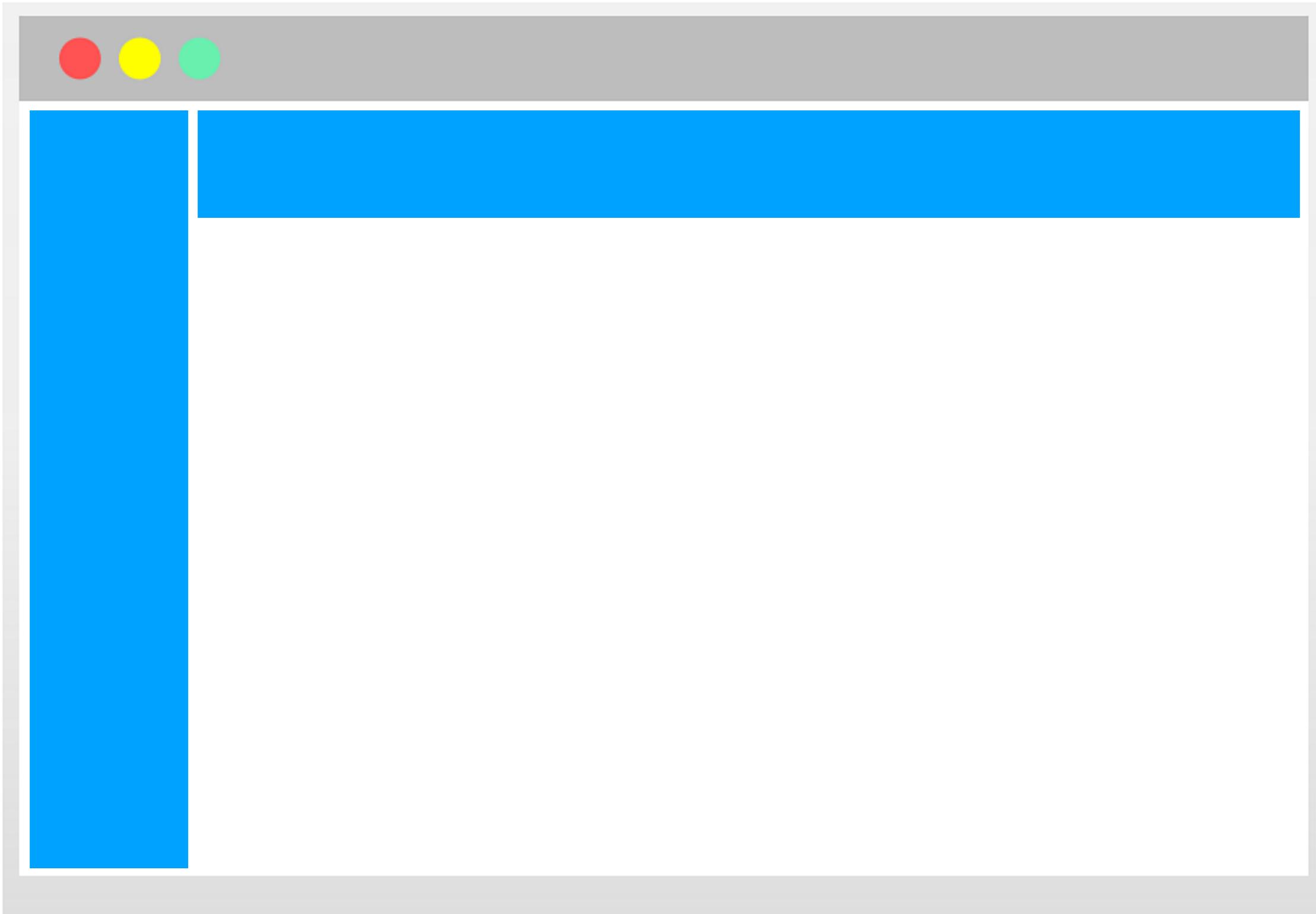
No Version Conflicts

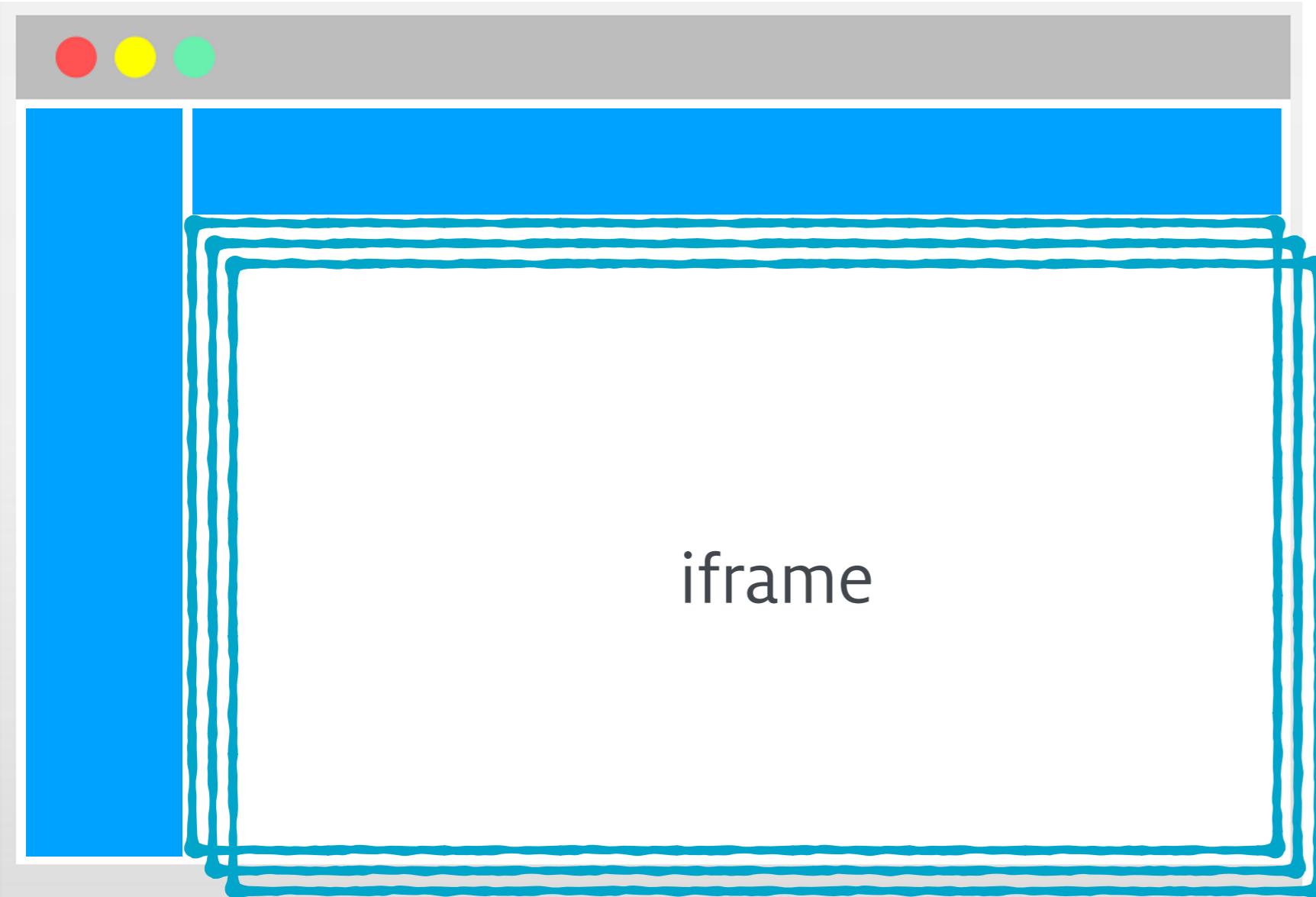
Separate Development



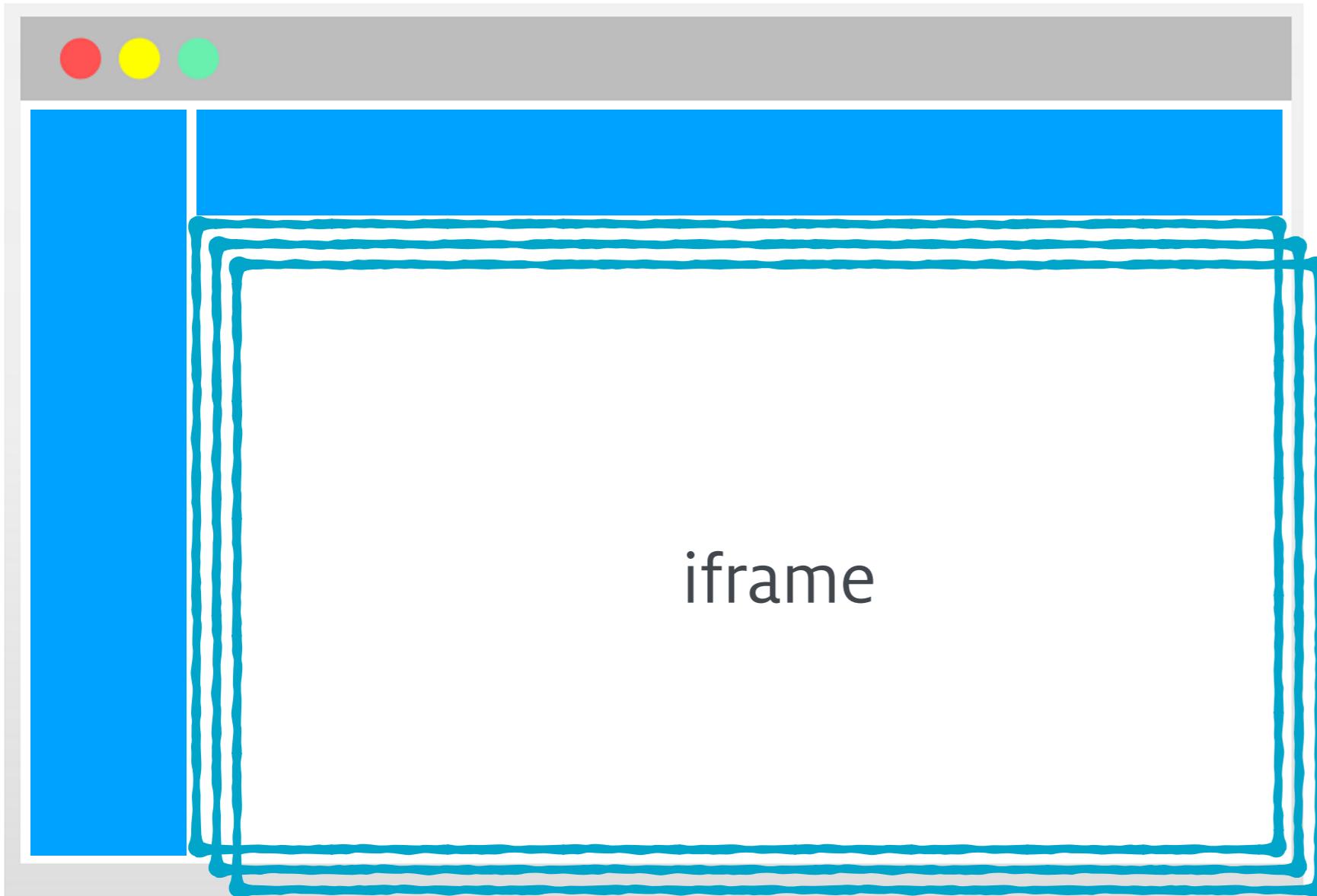
# Chapter 2: How?

# App Shell FTW!





App Shell  
- State  
- Routing



App Shell  
- State  
- Routing

Isolation  
Separate Deployment  
Feels like one page  
Different Frameworks  
Tree Shaking  
Several Frontends  
No Version Conflicts  
Separate Development

Vendor Bundles

```
import {MetaRouter} from 'meta-spa-router';

const config = [
  {
    path: 'a',
    app: '/app-a/dist'
  }, {
    path: 'b',
    app: '/app-b/dist'
  }];

```

```
window.addEventListener('load', () => {
  const router = new MetaRouter();
  router.config(config);
  router.init();
  router.preload();

  document
    .getElementById('link-a')
    .addEventListener('click', () => router.go('a'));

  document
    .getElementById('link-b')
    .addEventListener('click', () => router.go('b'));

  document
    .getElementById('link-aa')
    .addEventListener('click', () => router.go('a', 'a'));

  document
    .getElementById('link-ab')
    .addEventListener('click', () => router.go('a', 'b'));
});
```

```
import {MetaRouter} from 'meta-spa-router';

const config = [
  {
    path: 'a',
    app: '/app-a/dist'
  }, {
    path: 'b',
    app: '/app-b/dist'
  }
];
```

```
<div>
  <a id="link-a">Route to A</a> |
  <a id="link-b">Route to B</a> |
  <a id="link-aa">Jump to A within A</a> |
  <a id="link-ab">Jump to B within A</a>
</div>

<!-- placeholder for routed apps -->
<div id="outlet"></div>
```

```
window.addEventListener('load', () => {
  const router = new MetaRouter();
  router.config(config);
  router.init();
  router.preload();

  document
    .getElementById('link-a')
    .addEventListener('click', () => router.go('a'));

  document
    .getElementById('link-b')
    .addEventListener('click', () => router.go('b'));

  document
    .getElementById('link-aa')
    .addEventListener('click', () => router.go('a', 'a'));

  document
    .getElementById('link-ab')
    .addEventListener('click', () => router.go('a', 'b'));
});
```

```
import { Router, NavigationEnd } from '@angular/router';
import { Component } from '@angular/core';
import { filter } from 'rxjs/operators';
import { RoutedApp } from 'meta-spa-router';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})

export class AppComponent {
  title = 'app';

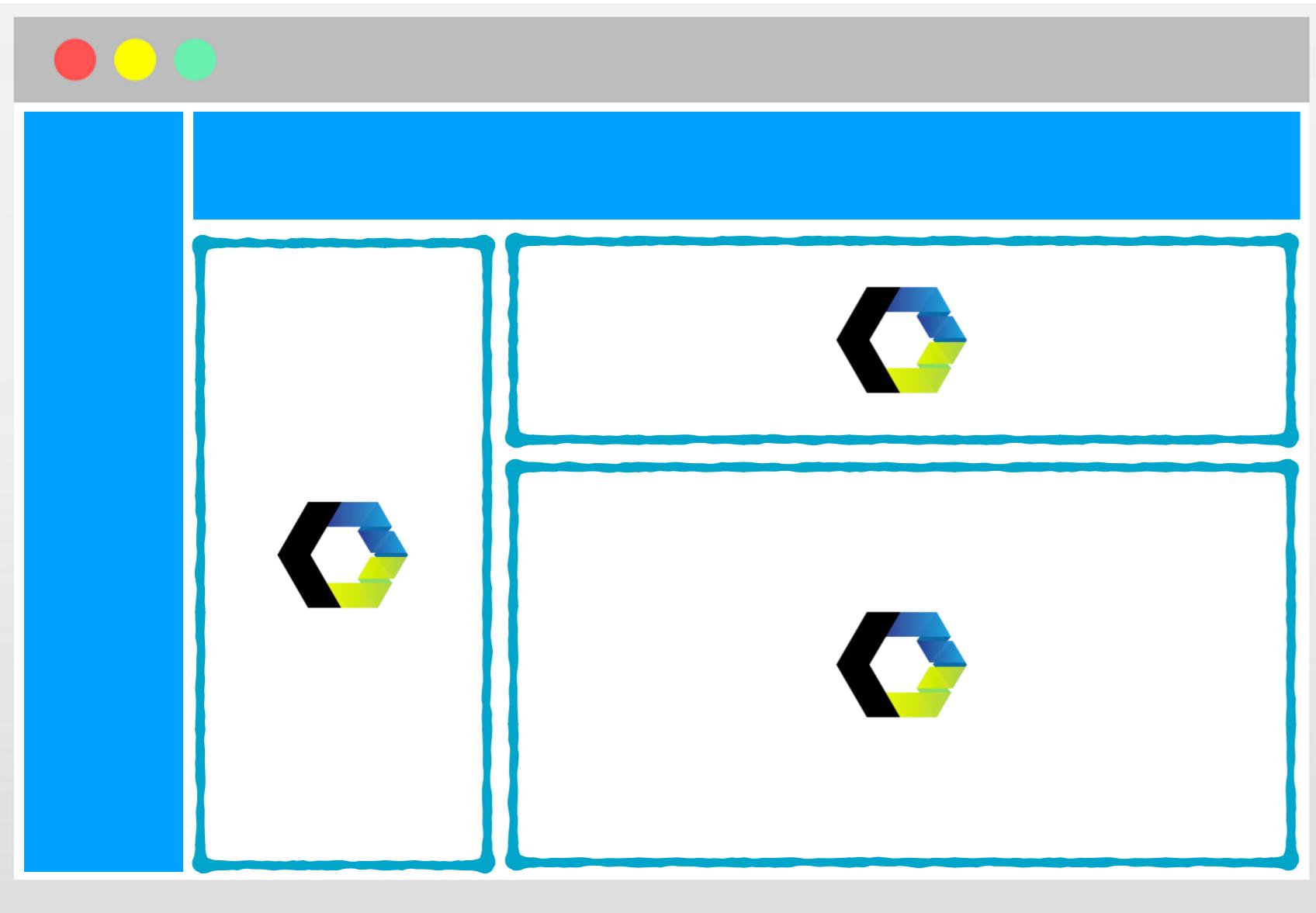
  constructor(
    private router: Router,
    @Inject(ROUTES_APP) private routedApp: RoutedApp) {
    this.initRoutedApp();
  }

  initRoutedApp() {
    this.routedApp.config({ appId: 'a' });
    this.routedApp.init();

    this.router.events.pipe(filter(e => e instanceof NavigationEnd)).subscribe((e: NavigationEnd) => {
      this.routedApp.sendRoute(e.url);
    });

    this.routedApp.registerForRouteChange(url => this.router.navigateByUrl(url));
  }
}
```

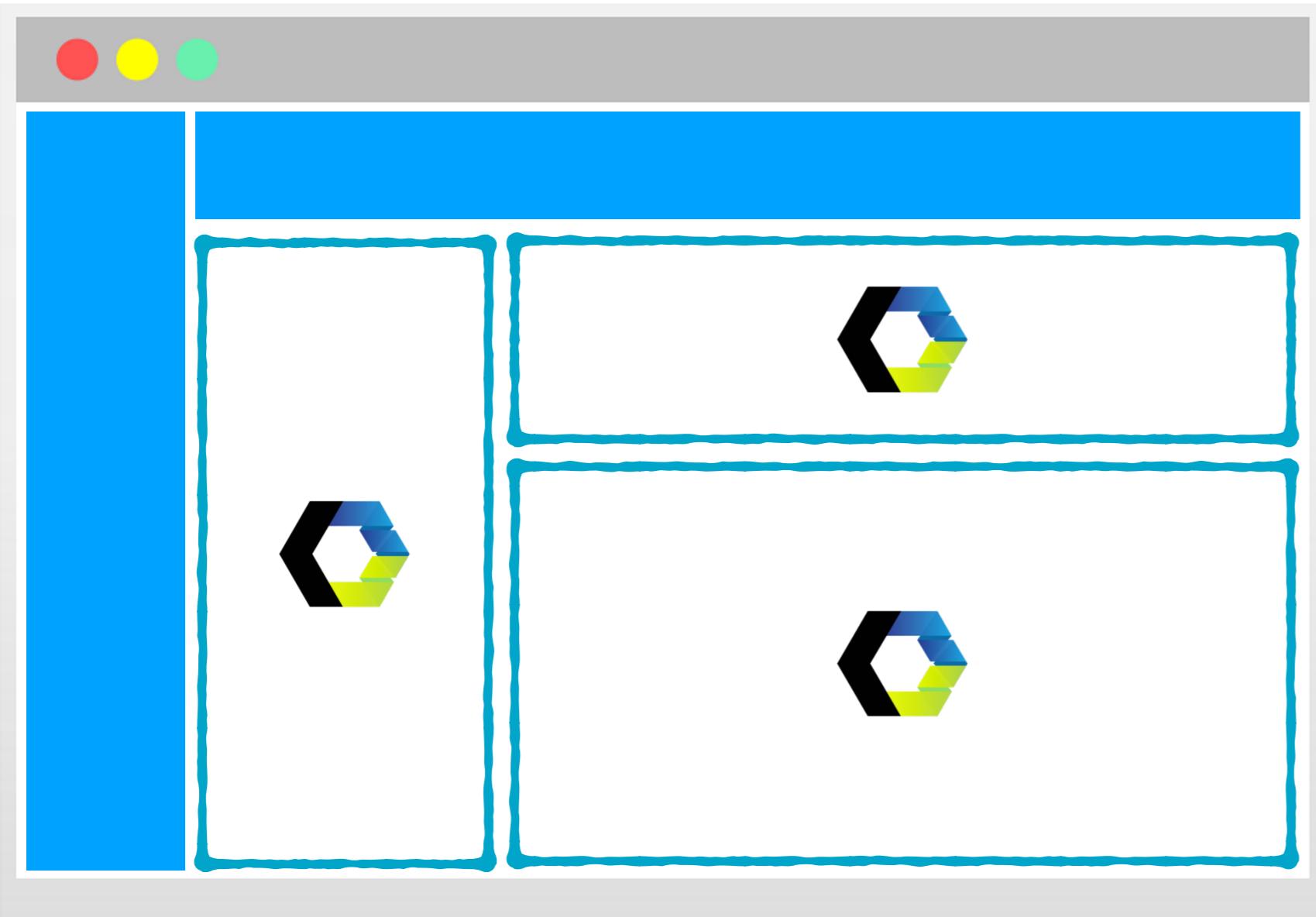
We need to register routing in the micro frontend!



## Web Components

Each app a component

Again App Shell is the orchestrator



## Web Components

Each app a component

Again App Shell is the orchestrator

Separate Deployment  
Feels like one page  
Different Frameworks  
Tree Shaking  
Several Frontends  
Separate Development

Isolation  
Vendor Bundles  
No Version Conflicts

imports

```
import {declareChildApplication, start} from 'single-spa';

declareChildApplication('angular1',
  () => import('./angular1/angular1.app.js'), pathPrefix('/angular1')
);
declareChildApplication('react',
  () => import('./react/react.app.js'), pathPrefix('/react')
);

start();

function pathPrefix(prefix) {
  return function(location) {
    return location.pathname.indexOf(`#${prefix}`) === 0;
  };
}
```



```
import singleSpaAngular1 from 'single-spa-angular1';
import angular from 'angular';

const domElementGetter = () => document.getElementById('angular1');

const angularLifecycles = singleSpaAngular1({
  angular,
  domElementGetter,
  mainAngularModule: 'single-spa-app',
  uiRouter: true,
  preserveGlobal: true,
});

export const bootstrap = [angularLifecycles.bootstrap];

export const mount = [angularLifecycles.mount];

export const unmount = [angularLifecycles.unmount];
```

<https://github.com/CanopyTax/single-spa>

<https://github.com/medikoo/event-emitter>

```

import {declareChildApplication, start} from 'single-spa';

declareChildApplication('angular1',
  () => import('./angular1/angular1.app.js'), pathPrefix('/angular1')
);

declareChildApplication('react',
  () => import('./react/react.app.js'), pathPrefix('/react')
);

start();

function pathPrefix(prefix) {
  return function(location) {
    return location.pathname.indexOf(`/${prefix}`) === 0;
  };
}

```

imports



```

import singleSpaAngular1 from 'single-spa-angular1';
import angular from 'angular';

const domElementGetter = () => document.getElementById('angular1');

const angularLifecycles = singleSpaAngular1({
  angular,
  domElementGetter,
  mainAngularModule: 'single-spa-app',
  uiRouter: true,
  preserveGlobal: true,
});

export const bootstrap = [angularLifecycles.bootstrap];

export const mount = [angularLifecycles.mount];

export const unmount = [angularLifecycles.unmount];

```

<https://github.com/CanopyTax/single-spa>

<https://github.com/medikoo/event-emitter>

imports



```

import React from 'react';
import ReactDOM from 'react-dom';
import singleSpaReact from 'single-spa-react';
import rootComponent from './root.component.js';

const reactLifecycles = singleSpaReact({
  React,
  ReactDOM,
  rootComponent,
  domElementGetter: () => document.getElementById('react-app')
});

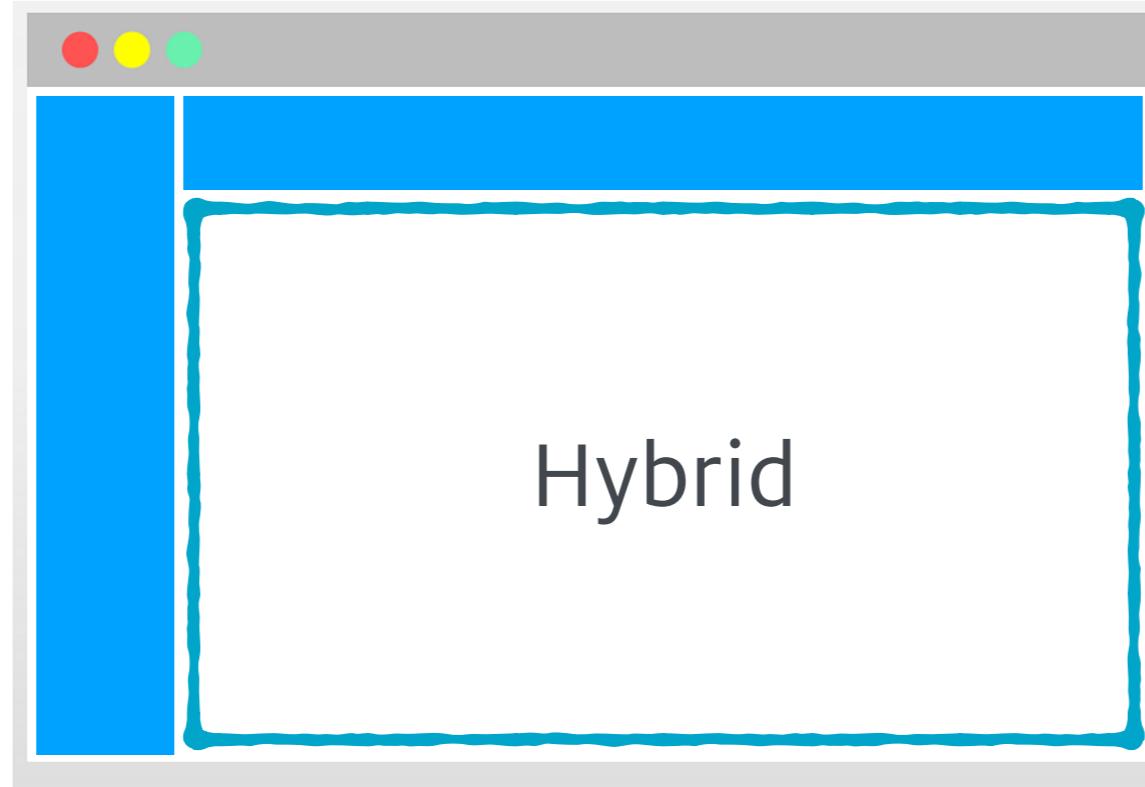
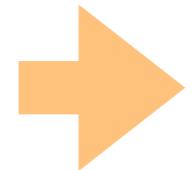
export const bootstrap = [reactLifecycles.bootstrap];

export const mount = [reactLifecycles.mount];

export const unmount = [reactLifecycles.unmount];

```

/reporting  
/account  
/editor



Reporting

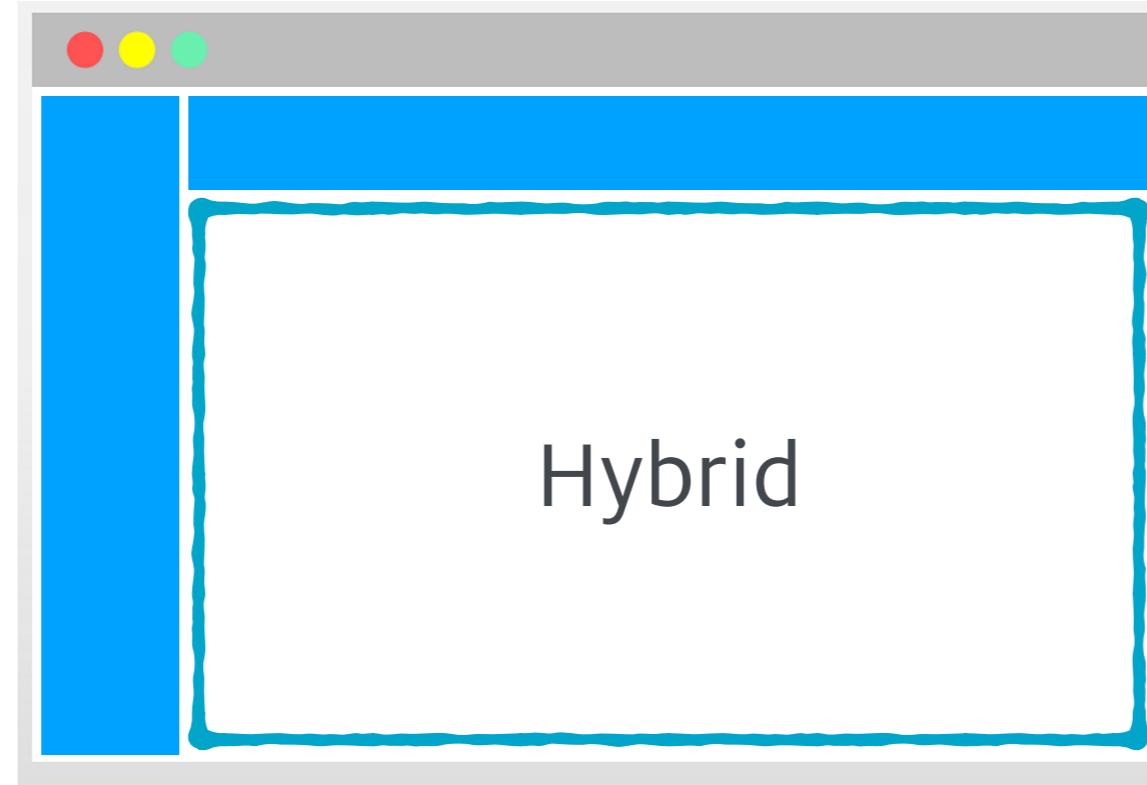
Security

Interactions

Account  
Management



/reporting  
/account  
/editor



Reporting

Security

Interactions

Account Management

Hybrid



Isolation  
Separate Deployment  
Feels like one page  
Different Frameworks

Tree Shaking (per app)  
Several Frontends  
Separate Development  
No Version Conflicts

Vendor Bundles

# Move responsibility to micro frontends

```
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="https://static.site.com/styleguide.css">
  </head>
  <body>
    <div id="root"></div>
    <script src="https://static.site.com/manifest.app.json"></script>
  </body>
</html>
```



loads

```
{
  "bundle": "https://static.site.com/bundle.min.js",
  "vendor": "https://static.site.com/vendor.min.js",
  "app": "https://static.site.com/app.min.js"
}
```



# Chapter 3: Challenges?

# Integration!



# Is this a micro frontend?

The screenshot shows the Usabilla Feedback Detail View. On the left, there's a sidebar with navigation links for 'Websites' (Buttons & Forms, Dashboard, **Feedback**, Benchmarking, Setup), 'Campaigns' (Overview), and 'In-Page Widgets' (Dashboard, **Feedback**, Setup). The main area has a title 'Detailed view Item 37 out of 2154'. To the right of the title is a list of feedback items with small icons and timestamps. One item is highlighted with a blue border. The detailed view on the right shows a summary of the selected feedback item, including the date (July 9, 2017 - 11:39), URL (<https://usabilla.com/partners>), and a large red 'REQUEST INFO' button. Below this are sections for 'Details', 'Subject' (Suggestion), 'User Comment' (knopje werkt niet!), 'NPS SCORE' (3), 'Button' (Usabilla.com), 'Labels' (Add label, Alpha, adjskdlaks..), 'Feedback ID' (123456789 123456789101112131415), 'Technical Info' (Desktop, Windows 8, Chrome 43.0, 1349 x 634, Average 2.02 sec, IP 10.10.10.1), and 'Integrations' (Clicktale, Tealeaf, Jira, Google Analytic, Adobe Analytics).

Feedback Details:

- July 9, 2017 – 11:39  
<https://usabilla.com/partners>
- SUBJECT: Suggestion
- BUTTON: Usabilla.com
- LABELS: Add label, Alpha, adjskdlaks..
- FEEDBACK ID: 123456789 123456789101112131415
- USER COMMENT: knopje werkt niet!
- TECHNICAL INFO:
  - Desktop
  - Windows 8
  - Chrome 43.0
  - 1349 x 634
  - Average 2.02 sec
  - IP 10.10.10.1
- NPS SCORE: 3

# Or are these micro frontends?

The screenshot shows the Usabilla interface. On the left, a sidebar menu includes sections for Websites, Buttons & Forms, Campaigns, and In-Page Widgets, with 'Feedback' selected. The main area displays a list of feedback items and a detailed view of one item.

**Feedback List:**

- 55 sec ago: Improve the performance of your websites, apps or emails
- new 55 sec ago: Easy Navigation
- 55 sec ago: Intelligent Data Watch your data respond intuitively to smart filters.
- 55 sec ago: Improve the performance of your websites, apps or emails
- 10 min ago: Report bugs and share their opinion all from within your app
- 29 Jan 2016: We've redesigned your NPS dashboard with simplicity
- 55 sec ago: **QUE** This button is not working!
- 55 sec ago: Our new summary section means you can instantly see
- 16 Oct 2016: Improve the performance of your websites, apps or emails
- 55 sec ago: Improve the performance of your websites, apps or emails

**Detailed View:**

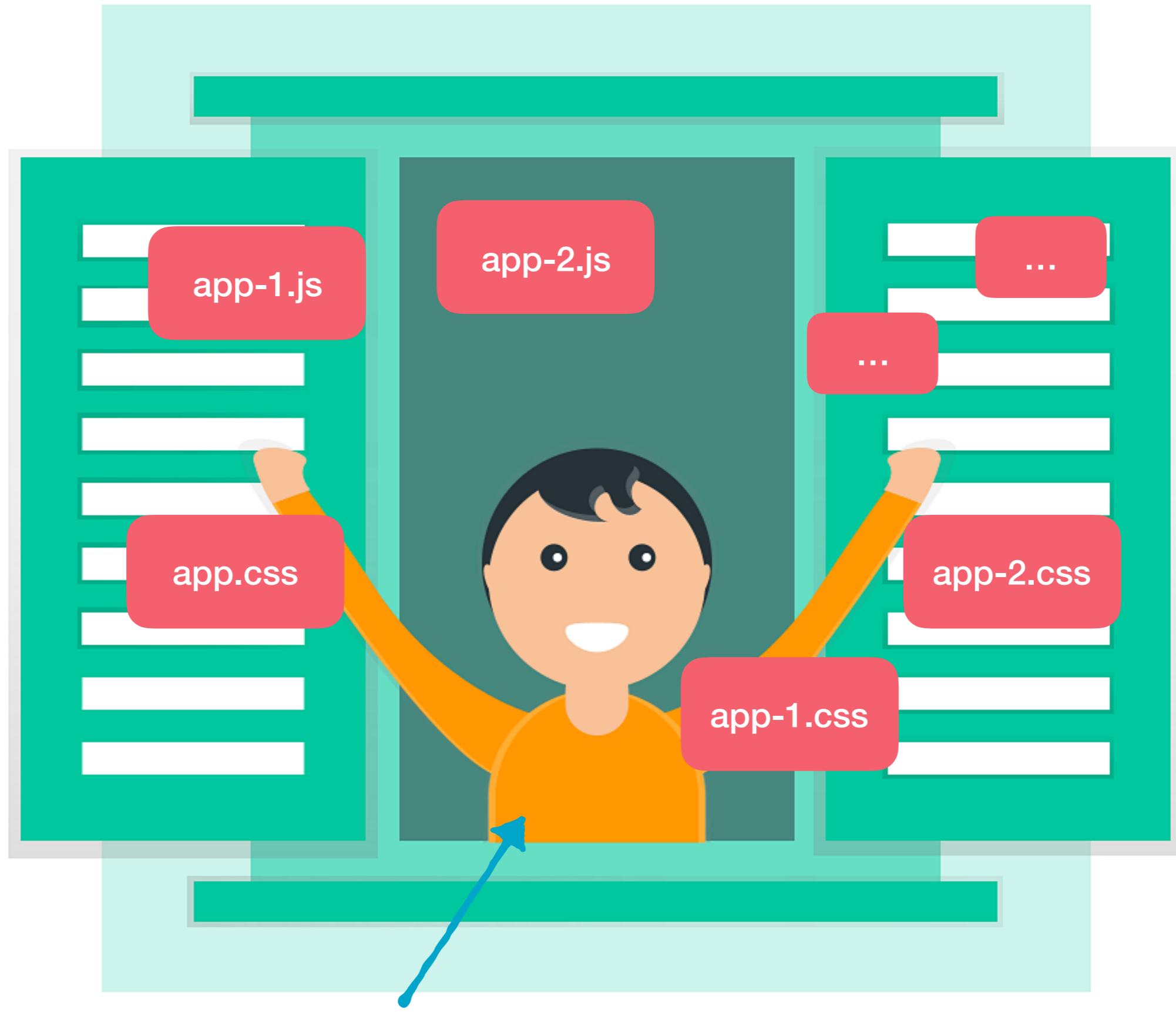
**Feedback Item Details:**

- SUBJECT:** Suggestion
- BUTTON:** Usabilla.com
- LABELS:** Add label, Alpha, adjskdlaks..
- FEEDBACK ID:** 123456789 123456789101112131415
- TECHNICAL INFO:**
  - Desktop
  - Windows 8
  - Chrome 43.0
  - 1349 x 634
  - Average 2.02 sec
  - 10.10.10.1
- INTEGRATIONS:**
  - Clicktale
  - Tealeaf
  - Jira
  - Google Analytic
  - Adobe Analytics

**User Comment:** knopje werkt niet!

**NPS Score:** 3

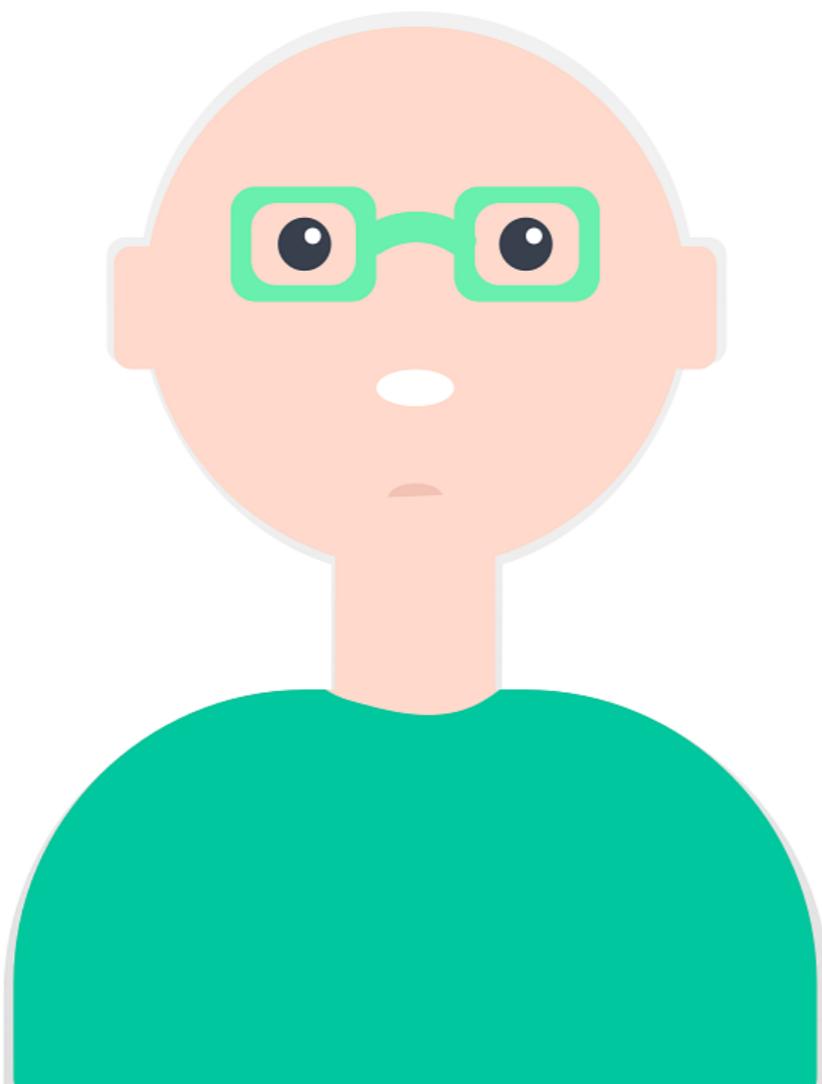
**Request Info Button:** REQUEST INFO



This person suffers

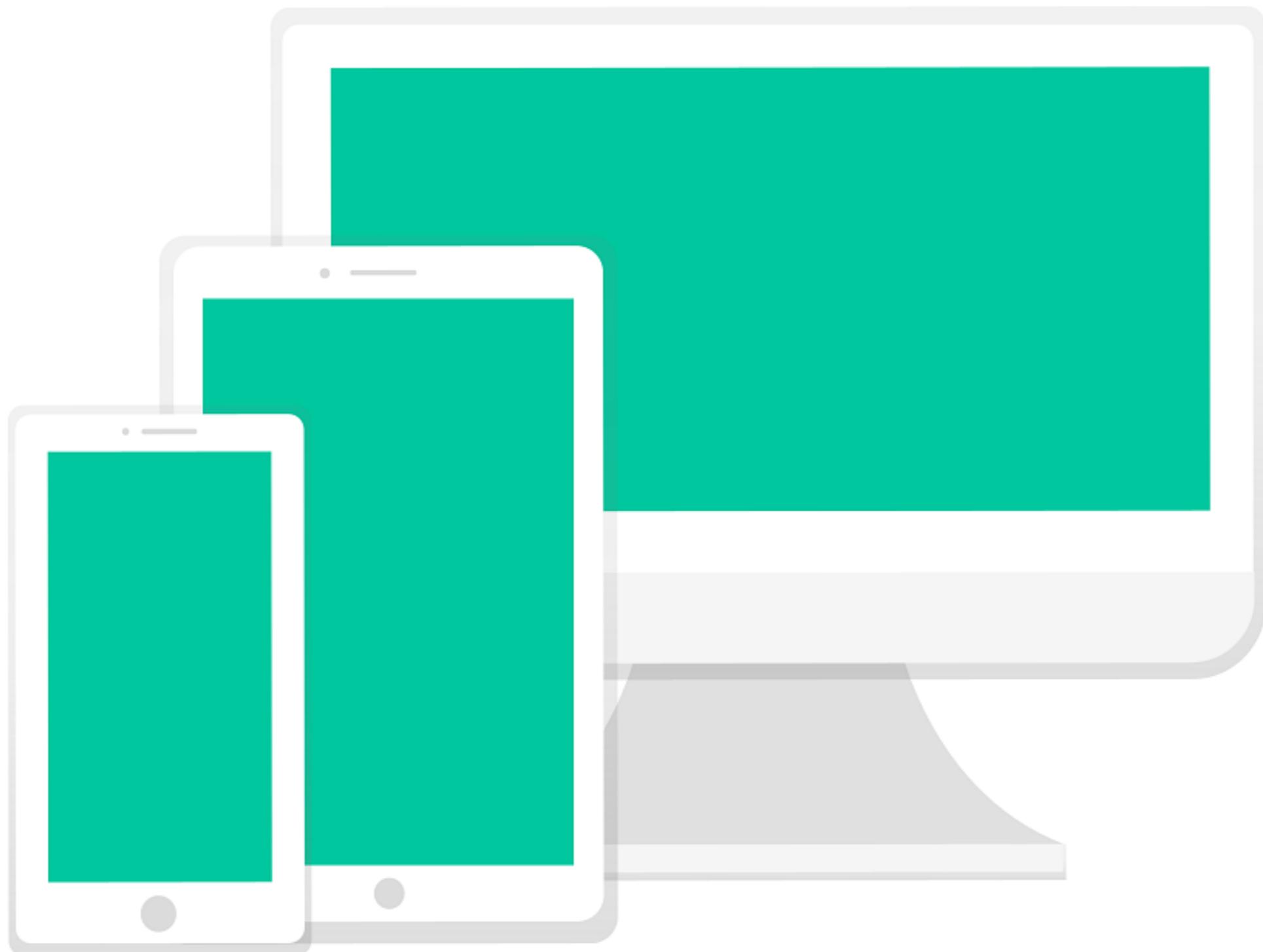
Its your time to figure out what  
the previous developers had in mind!

We use docker, it should work!



And this person suffers

# Adaptive, Responsive & Accessible Design



Integration  
Testing is  
probably your  
best bet!



**A software tester walks  
into a bar and then  
runs into the bar...  
strolls into the bar...  
gallops into the bar...  
saunters into the bar...**

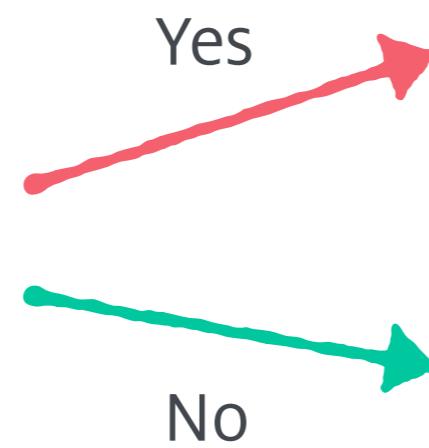
by Jasmine Harpley

[www.ministryoftesting.com](http://www.ministryoftesting.com)



# Chapter 4: Thoughts?

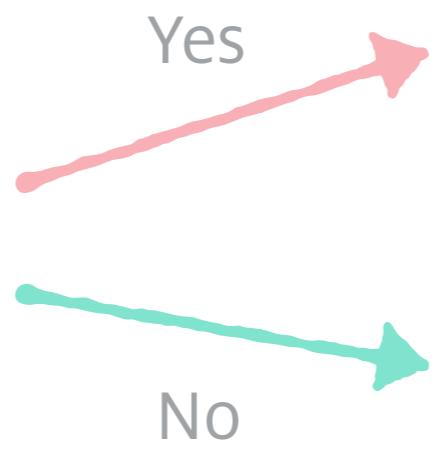
Can you fit your team  
around a large table?



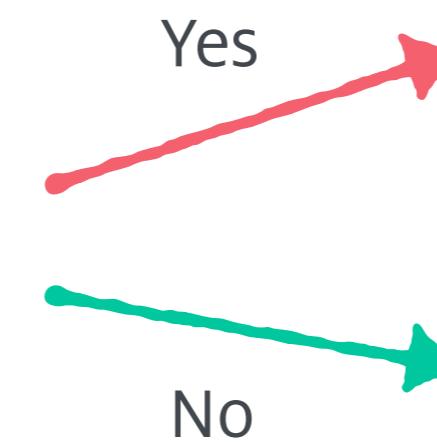
You might not need  
micro frontends yet

Micro frontends  
might help you

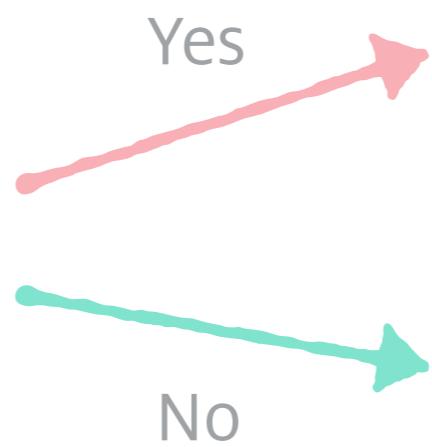
Can you fit your team  
around a large table?



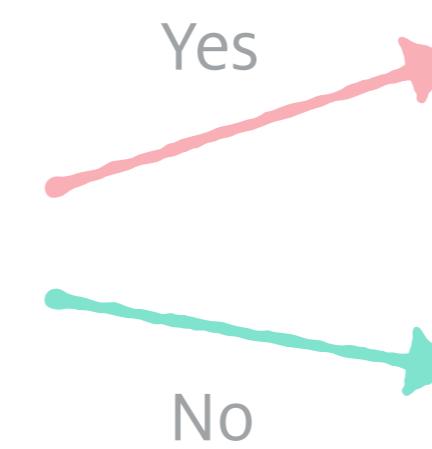
Do you have monolithic  
dependencies?



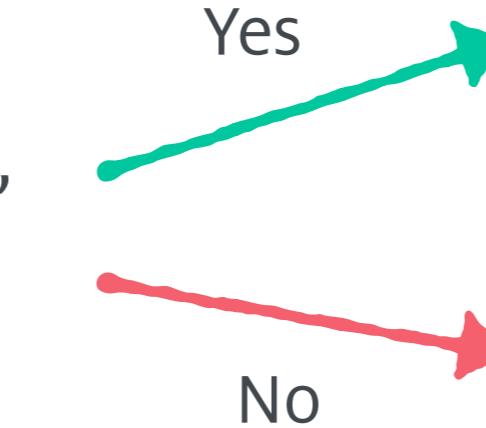
Can you fit your team around a large table?



Do you have monolithic dependencies?



Got container, orchestration, devops experience?





Extend the monolith?

“Architecture in micro frontends is **domain driven**, and low level components can be technical driven.

It makes sense to follow a **top down approach**.

Thinking of how to split your monolith in apps according to your domain. **No big design upfront** as things might go wrong.”





**Mattias P Johansson** @mpjme · 19 Sep 2017

Replying to [@derberq](#)

Not there any more, but it's not secret they are moving away from that architecture to a React/Redux architecture.



**Łukasz Górnicki** @derberq · 19 Sep 2017

so basically no more technology independence? each team must use react?



**Mattias P Johansson** @mpjme · 19 Sep 2017

Yes. Downside: Harder to introduce new tech. Upside: Not shipping every JS framework in the world with the client. :)



**Mattias P Johansson** @mpjme · 19 Sep 2017

Also the ability to share build pipeline, shared libraries and styling.



**Mattias P Johansson** @mpjme · 19 Sep 2017

I was actually a fan of the micro frontend and multi-repo but even I must admit that in hindsight it's so much better.



<https://twitter.com/derberq/status/910056617881817089>

# Thank you!

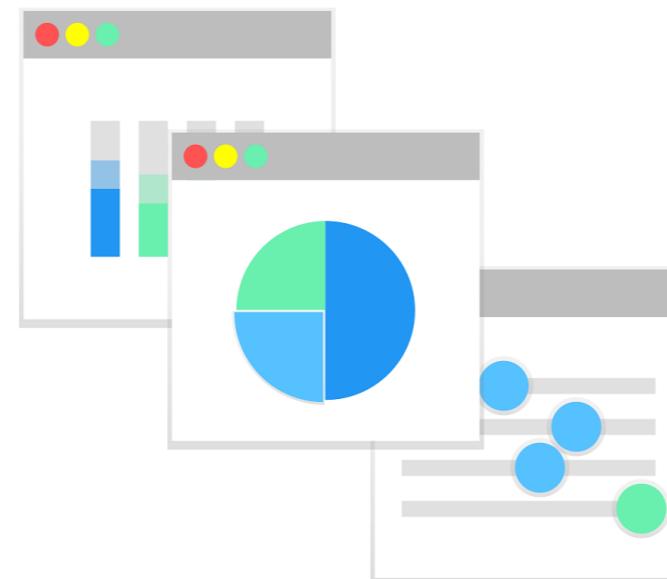


<https://jobs.usabilla.com/>

MIT licensed illustrations for every project you can imagine and create

Browse now

Created by Katerina Limpitsouni



<https://undraw.co/>

## Check these out!

- > <https://medium.com/@gilfink/why-im-betting-on-web-components-and-you-should-think-about-using-them-too-8629396e27a>
- > <https://softwarearchitekt.at/post/2017/12/28/a-software-architect-s-approach-towards-using-angular-and-spas-in-general-for-microservices-aka-microfrontends.aspx>
- > <https://www.upwork.com/blog/2017/05/modernizing-upwork-micro-frontends/>
- > <https://x-team.com/blog/micro-frontend>
- > <https://medium.com/@tomsoderlund/micro-frontends-a-microservice-approach-to-front-end-web-development-f325ebdac16>
- > <https://www.packtpub.com/books/content/what-micro-frontend>
- > <https://technologyconversations.com/2015/08/09/including-front-end-web-components-into-microservices>
- > <https://www.tikalk.com/js/introduction-to-micro-frontends/>
- > <https://micro-frontends.zeef.com/elisabeth.engel?ref=elisabeth.engel&share=ee53d51a914b4951ae5c94ece97642fc>