Micro Front-End @

Giacomo Magini
require('lx') 12/09/2019

onfido

onfido

Giacomo Magini

Software Engineer
(lots of Front-End)

onfido

Traditional Front-End

Your CTO

We need a dashboard!

You

Ok. React, angular or Vue?

Your CTO

I don’t care. Let’s use a design system!

You

fiiiiiine

yarn install

~~🔥~~ ~~🔥~~ ~~🔥~~

We build a feature...

👨‍💻

and another...

👨‍💻👨‍💻

and another...

👨‍💻👨‍💻👨‍💻

The problems begin!

Codebase has grown

Developers number has grown

Development friction has grown

Delivery time has grown

Risk of collision for package update has grown

About us...

📉 & 😩

Micro Front-End

Whaaaat?

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

Cam Jackson

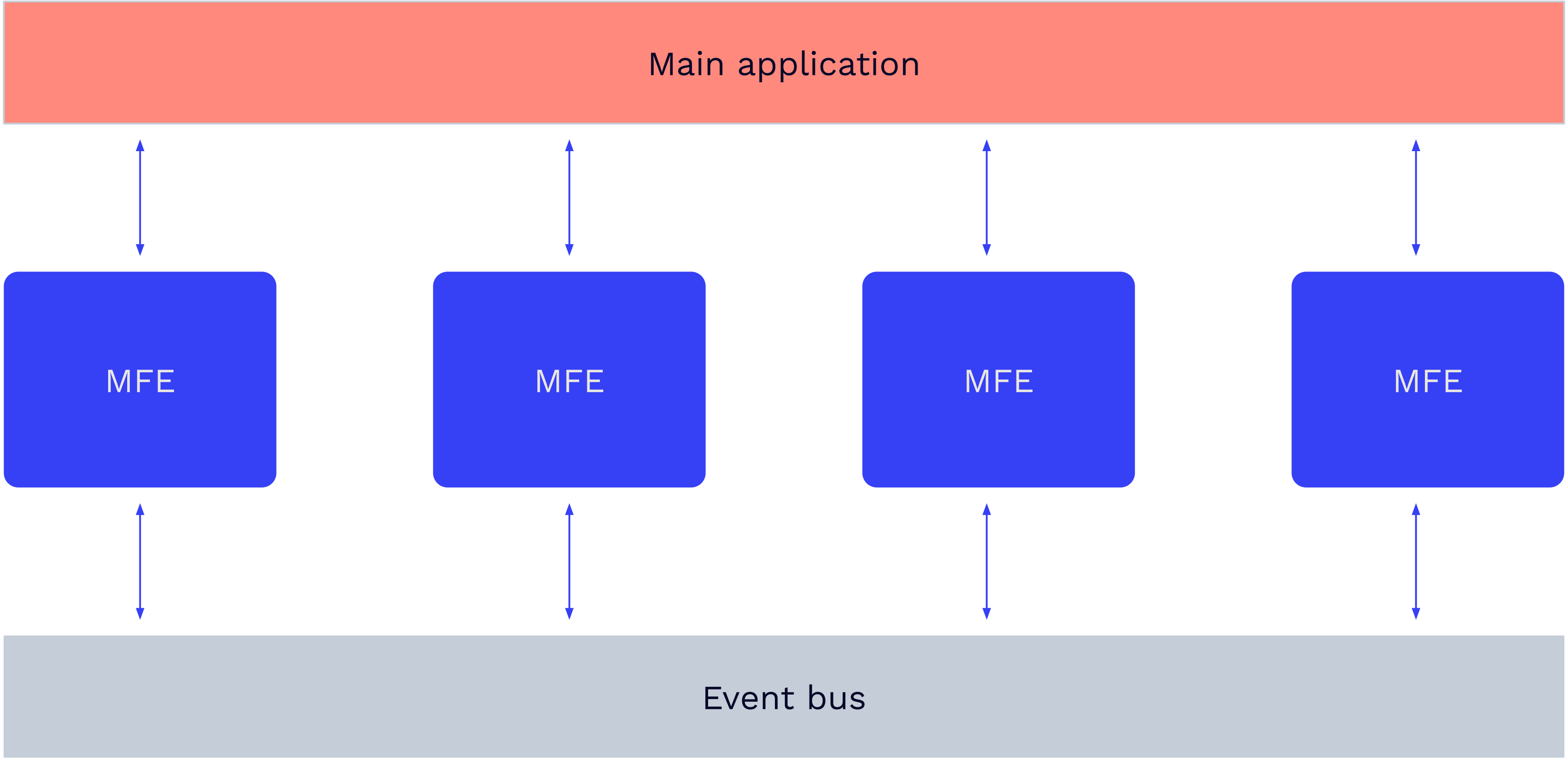
(martinfowler.com)

Horrible MFE, HERE WE ARE!!

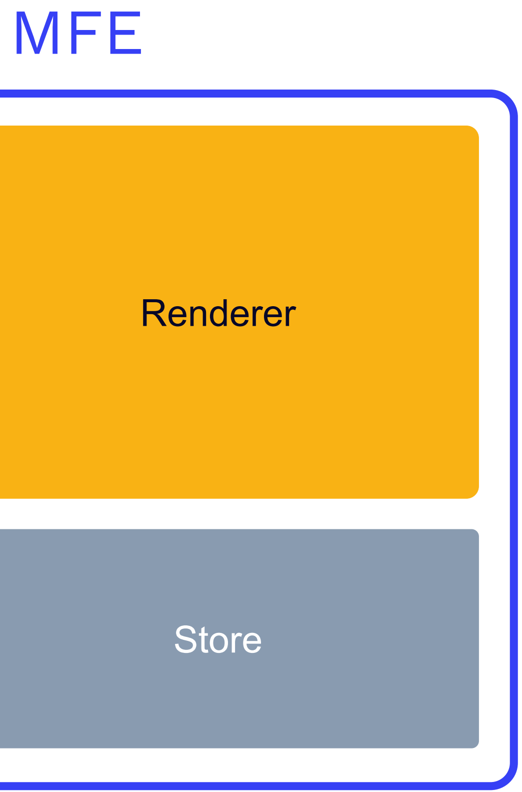
```
<html>
<body>
 <h1>Welcome to Onfido!</h1>
 <iframe src="https://mfe1.onfido.com"></iframe>
 <iframe src="https://mfe2.onfido.com"></iframe>
 <iframe src="https://mfe3.onfido.com"></iframe>
</body>
</html>
```

Onfido’s implementation

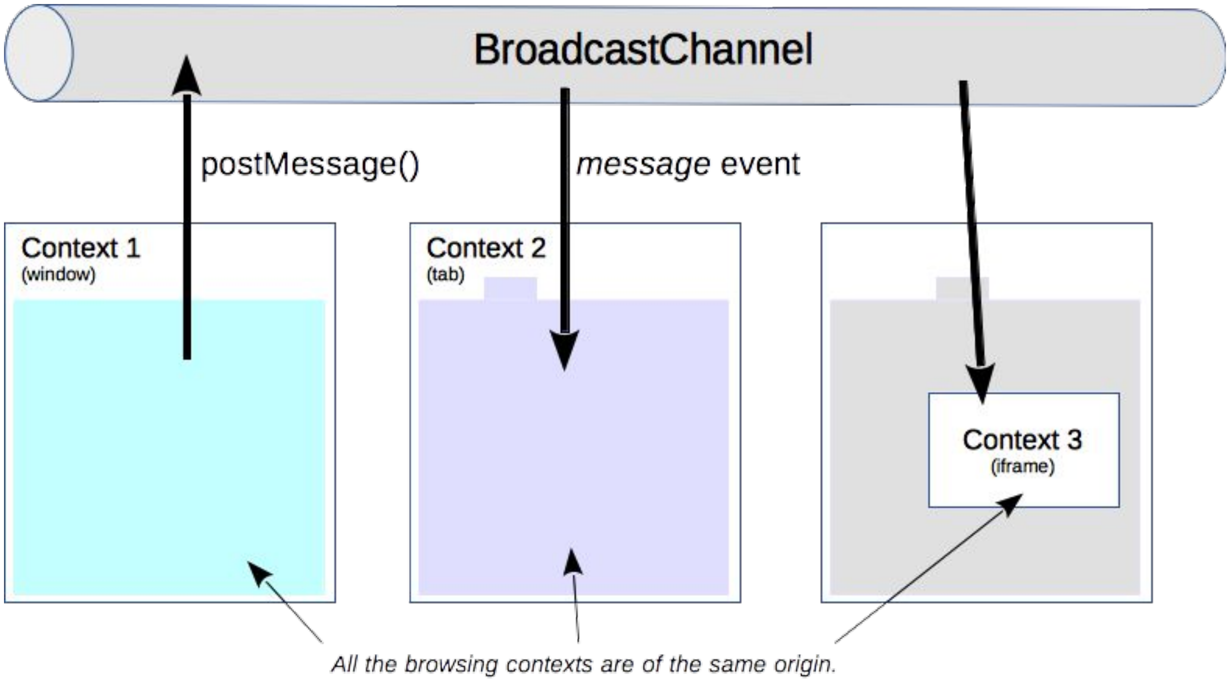
Application structure



What’s inside a MFE?



Events - Broadcast channel



Source: https://developer.mozilla.org/en-US/docs/Web/API/Broadcast_Channel_API

Nothing else to worry about?

MFE is a happy greenfield but...

Styling 👻- Stealthy rules

- We can occur in conflicts due to name clashes and global styles

- Pay attention to 3rd party libraries with global CSS

- Solution: namespace all of your styles either with CSS pre-processor, CSS-IN-JS or Shadow DOM

Global/Single instance- i.e. Sentry

- It really depends on the library implementation
- An “easy” solution could be Dependency Injection, if the library allows that.

Ship it (independently)

🎉🎉🎉🎉🎉🎉

S3 Bucket - AWS

- Static name for the entrypoint file of the bundle
- Cloudfront as CDN

Dynamic imports, lazy-loading

```
import(
 /* webpackIgnore: true */
`http://onfido.s3.aws.com/my-example-index.js`)
.then(() => {

  /* use your module here */

});
```

NOTE: this help us to update a MFE even if a user doesn’t refresh the page.

It’s all about trade-offs

What a news!

DRAWBACKS

- No dependencies sharing, overall size app will grow
- CSS is trickier
- Single/global instance libraries
- Easy to encounter in tech fragmentation
- Incurs an overhead that is not worthwhile for small apps. However, if not planned for, it could take a big effort to migrate to this approach

DRAWBACKS

- No dependencies sharing, overall size app will grow
- CSS is trickier
- Single/global instance libraries
- Easy to encounter in tech fragmentation
- Incurs an overhead that is not worthwhile for small apps. However, if not planned for, it could take a big effort to migrate to this approach

BENEFITS

- MFEs can be worked on and deployed independently
- Stability
- Deploys are much faster - no need to compile and test the entire app
- Different frameworks play nice together
- Every MFE is an independent app, it allows you to iterate over new technologies and learn it without impacting the entire app.
- No dependencies sharing - no potential for stealth dependency updates causing unexpected issues

Thanks for listening!

👏 👏 👏 👏

onfido