

Micro Frontends

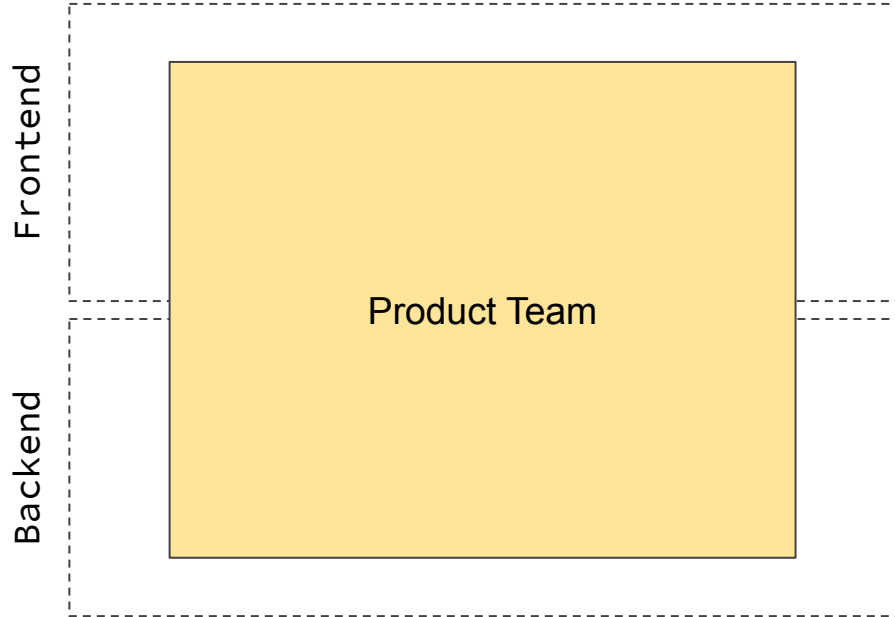
Creating complex and integrated distributed systems

Once upon a time...

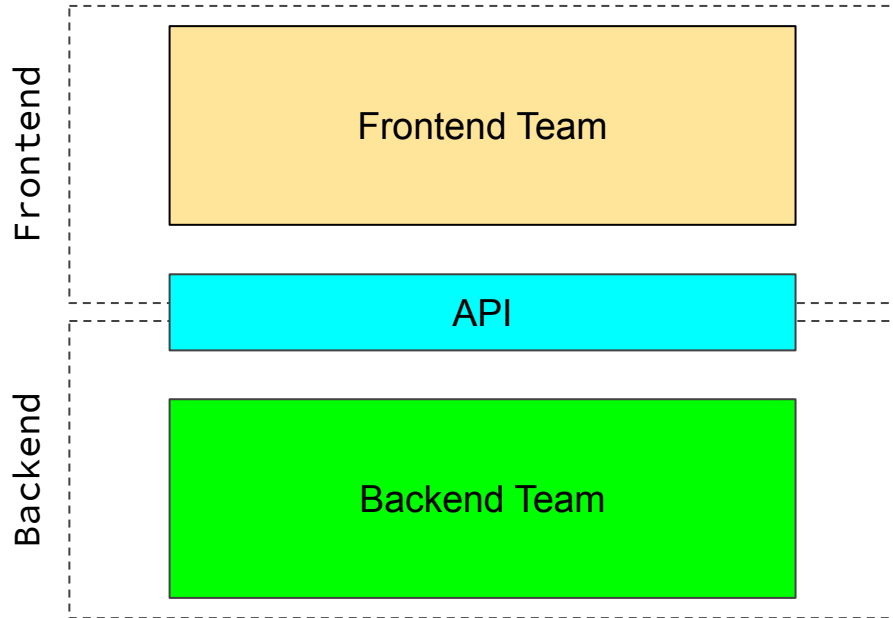
— — —

In recent years, microservices have exploded in popularity, with many organisations using this architectural style to avoid the limitations of large, monolithic backends. While much has been written about this style of building server-side software, many companies continue to struggle with monolithic frontend codebases

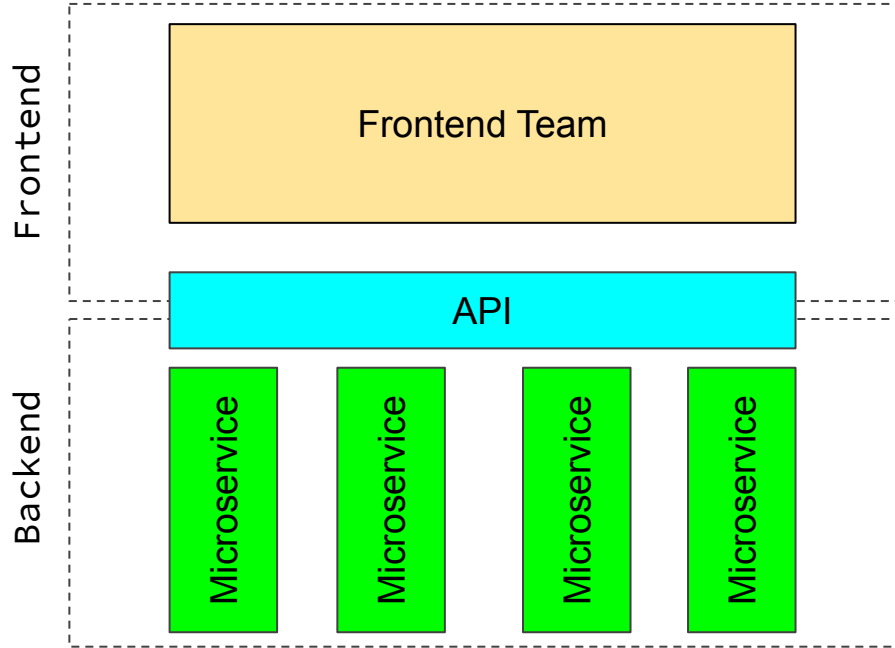
Monolith...



Front-end & Back-end...



Microservice



Micro frontend



Microfrontend

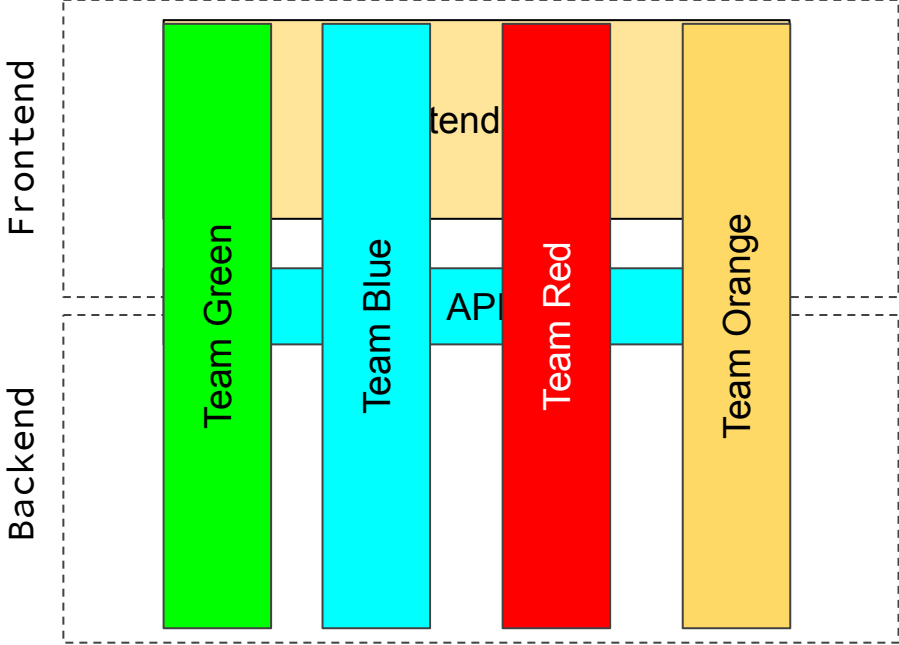
— — —

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

— Martin Fowler, 2016

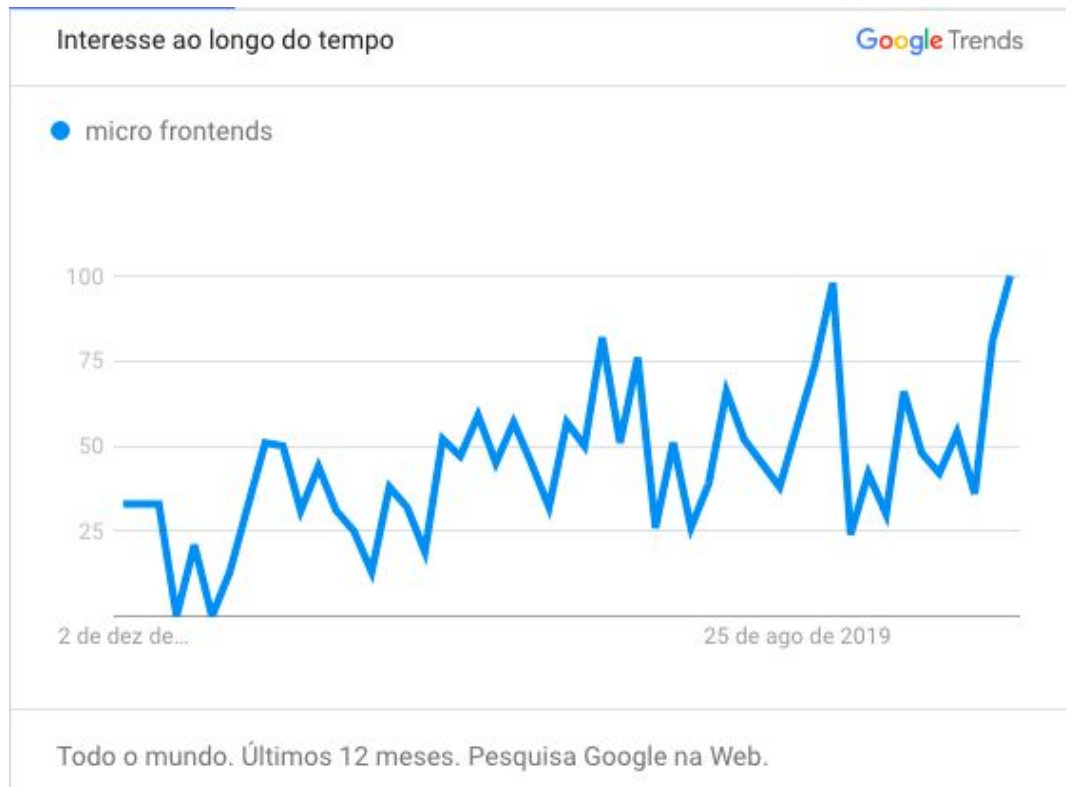


Microfrontend



Microfrontend

— — —



Microfrontend

TECHNOLOGY RADAR

🔍 Buscar Sobre o Radar Construa seu Radar Assine

Técnicas

Ferramentas

Plataformas

Linguagens & Frameworks

i Atualmente o conteúdo do nosso radar interativo está disponível apenas em inglês. Para visualizar o radar no seu idioma, faça download do PDF [aqui](#).

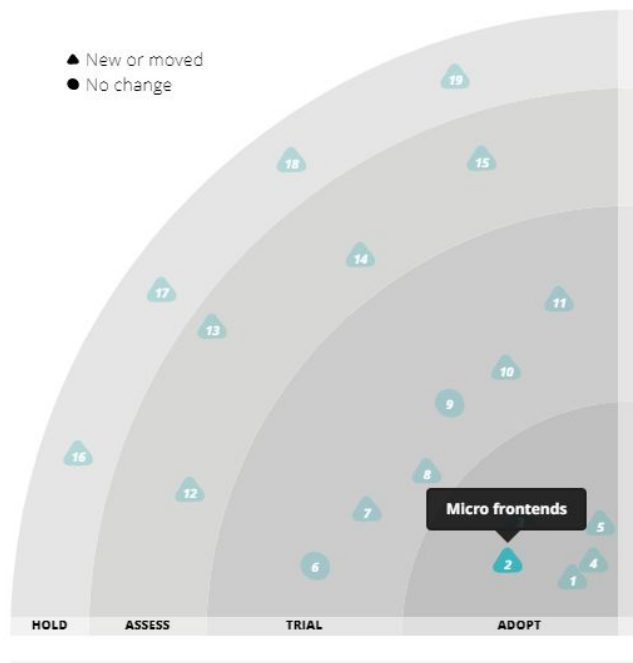
ADOPT ?

1. Four key metrics

2. Micro frontends

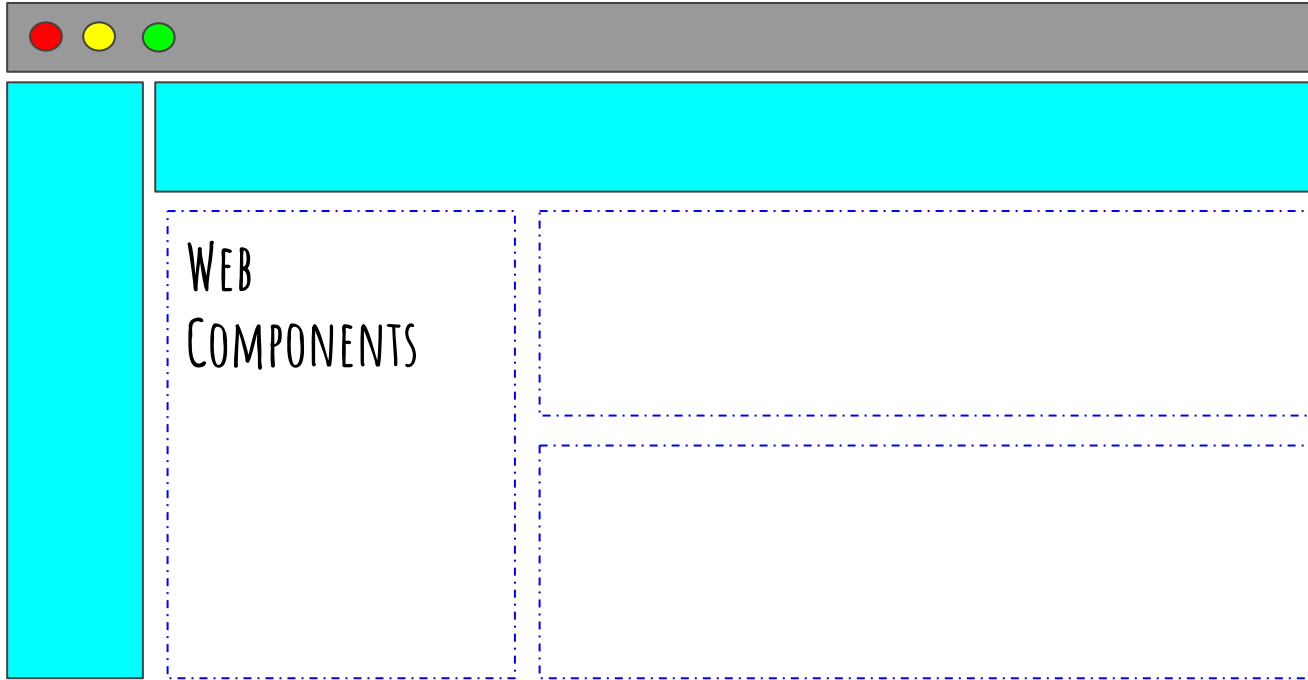
We've seen significant benefits from introducing [microservices](#), which have allowed teams to scale the delivery of independently deployed and maintained services. Unfortunately, we've also seen many teams create a frontend monolith — a large, entangled browser application that sits on top of the backend services — largely neutralizing the benefits of microservices. Since we first described **micro frontends** as a technique to address this issue, we've had almost universally positive experiences with the approach and have found a number of patterns to use micro frontends even as more and more code shifts from the server to the web browser. So far, [web components](#) have been elusive in this field, though.

▲ New or moved
● No change

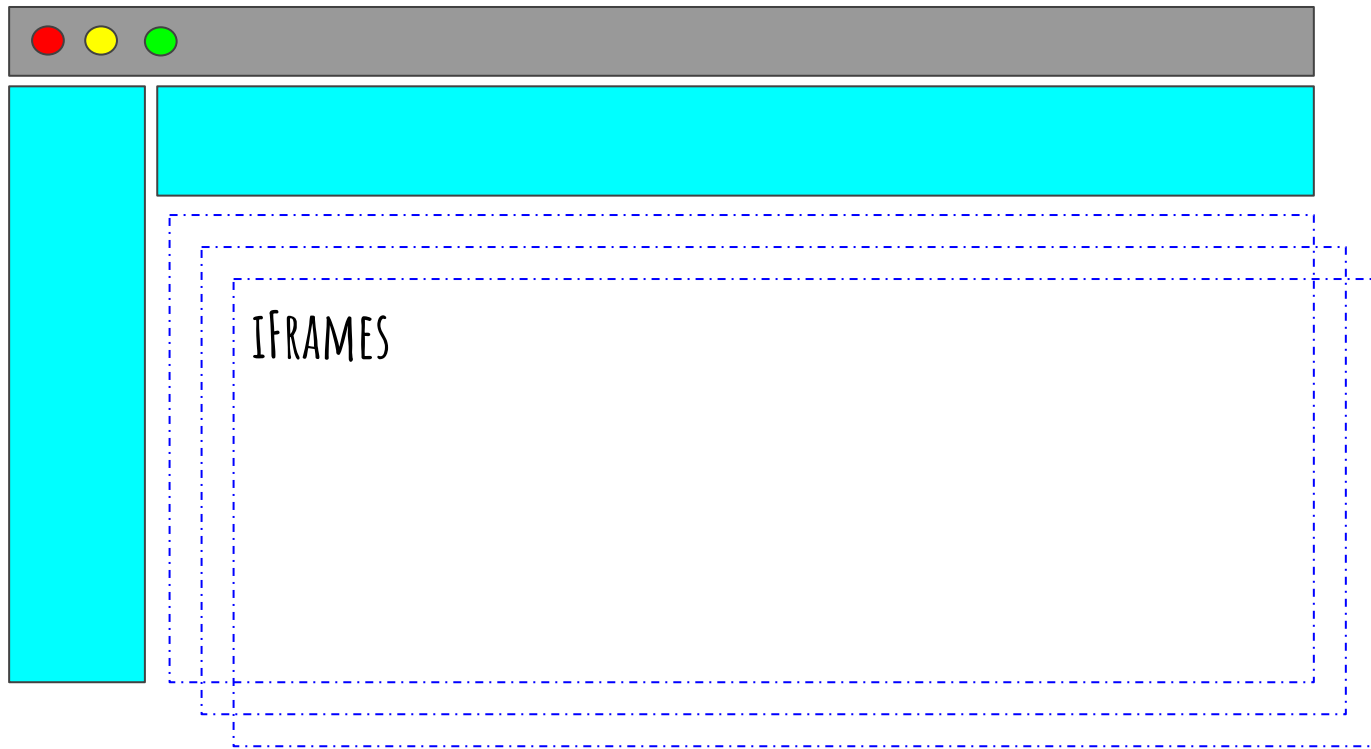


Integration approaches

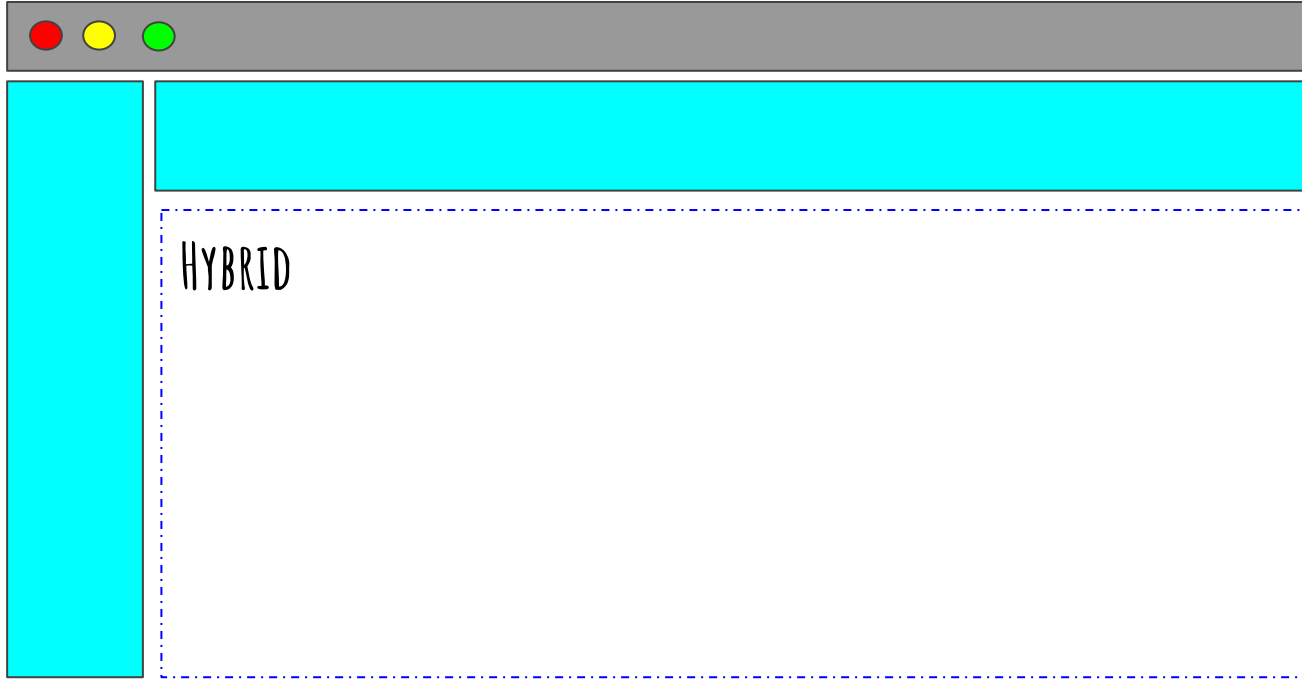
App shell



App shell



App shell



Real Example

APP SHELL - TEAM ORANGE

The screenshot displays a mobile application interface for 'TEAMORANGE'. The top navigation bar is dark red with a hamburger menu icon, the text 'TEAMORANGE', and a 'LOGIN' button. Below the navigation bar, there are two main sections: 'RECLAMAÇÕES' and 'COMENTÁRIOS'. Each section has a search bar, a 'Filtros' button, and a 'Resultado' button. The 'RECLAMAÇÕES' section also includes a 'Descrição:' field and a table with columns for 'Descrição', 'Data Início', and 'Data Fim'. The 'COMENTÁRIOS' section is currently empty. To the right of the 'COMENTÁRIOS' section, there is a pie chart titled 'Browser market shares at a specific website, 2014'. The pie chart shows the following data: Firefox (45%), IE (26.8%), Chrome (12.8%), Safari (8.5%), Opera (6.2%), and Others (0.7%). The bottom of the screen shows the version 'V 1.0' and logos for 'PROBPMS' and 'prodemge'.

TEAMORANGE LOGIN

RECLAMAÇÕES PESQUISAR LIMPAR + NOVO

Filtros

Descrição:

Resultado

Descrição Data Início Data Fim

COMENTÁRIOS PESQUISAR LIMPAR + NOVO

Filtros

Resultado

Browser market shares at a specific website, 2014

Firefox: 45 %

IE: 26.8 %

Chrome: 12.8 %

Safari: 8.5 %

Opera: 6.2 %

Others: 0.7 %

V 1.0 PROBPMS prodemge

TEAM BLUE

TEAM RED

“

No silver bullet.

– Fred Brooks, 1986



Microfrontend - Benefits vs Downsides

INCREMENTAL UPGRADES

SIMPLE, DECOUPLED CODEBASES

INDEPENDENTS DEPLOYMENT

AUTONOMOUS TEAMS

IN A NUTSHELL

PAYLOAD SIZE

ENVIRONMENT DIFFERENCES

OPERATIONAL AND GOVERNANCE
COMPLEXITY

Questions

Can you fit your team around a large table?

Yes

You might not need micro frontends yet

No

Micro frontends might help you

Do you have monolithic dependencies?

Yes

Integration might be complicated

No

Micro frontends may be easily integrated

Got devops experience?

Yes

Micro frontends might be valuable

No

Consider giving them a try first

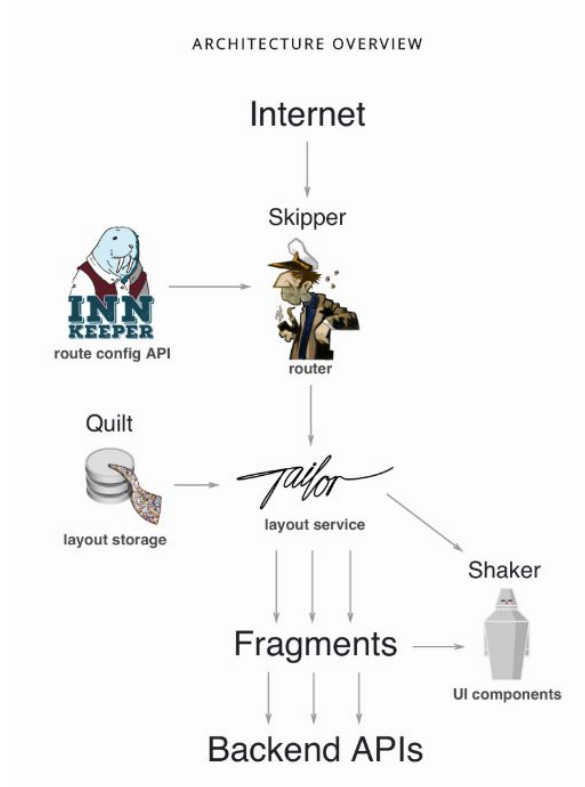
Projects Initiatives

Project Mosaic



Project Mosaic | **Microservices for the Frontend**

Mosaic is a set of services, libraries together with a specification that defines how its components interact with each other, to support a microservice style architecture for large scale websites. While decomposing the backend into microservices is a widely adopted approach to achieve flexibility in development and operation, most frontend solutions are still running as a monolithic application. Mosaic addresses this issue by using Fragments that are served by separate services and are composed together at runtime according to template definitions. As a result, Fragments can be iterated on very rapidly, be more flexible in technology choices, and can better benefit from the extreme development pace of today's frontend technologies.




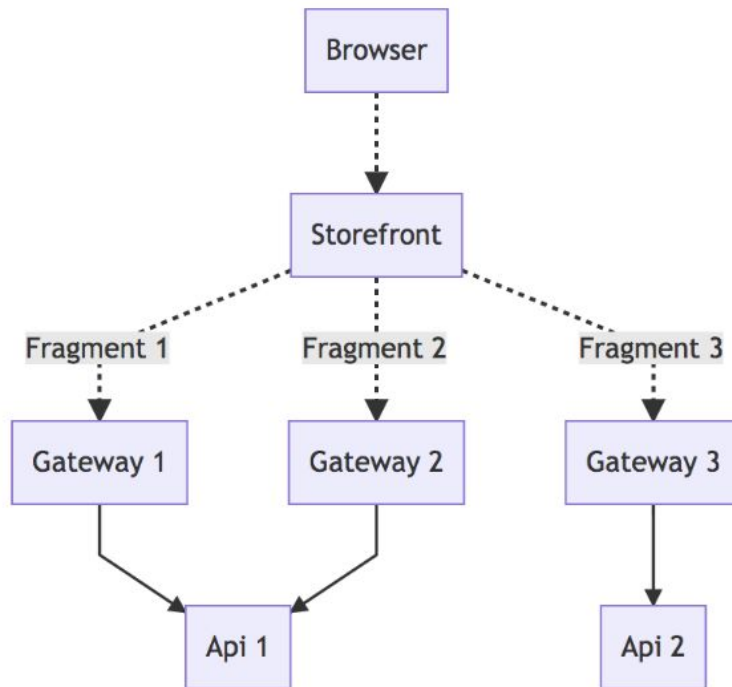
PuzzleJs



PuzzleJs Framework

Micro frontend framework for scalable and blazing fast websites.

 PASSED  downloads: 23k  npm v3.32.6  vulnerabilities: 0  codecov: 90%



Get our hands dirty!



References

<https://www.slideshare.net/YugoSakamoto1/state-of-micro-frontend-114560988>

<https://pt.slideshare.net/spyrosioakeimidis/micro-frontends-86962142>

<https://www.mosaic9.org/>

<https://github.com/puzzle-js/puzzle-js/blob/master/docs/guide.md#architecture>

<https://www.dwmkerr.com/the-death-of-microservice-madness-in-2018/>

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

<https://www.thoughtworks.com/pt/radar/techniques?blipid=1035>

<https://micro-frontends.org/>

<https://medium.com/javascript-in-plain-english/microfrontends-bringing-javascript-frameworks-together-react-angular-vue-etc-5d401cb0072b>