# And what if microservices are just the beginning?

Micro... frontends?

Aleksander Orchowski

October 23, 2019

## Frontend that we know

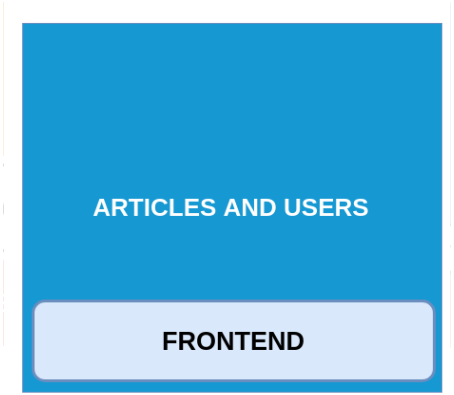
Now we are going to talk about approaches used at today's systems.

I use the example of the blog app:

- Service which manage users, authentication etc.
- Service for articles (listing, creating, editing, etc.)
- Frontend is SPA/SPA like

## Monolith

It's important to understand something about structure of application. So, we have one big block of code:



## Pros&Cons

+

- Everything at one place
- Probably - at one pull request we can add new feature
- Deployed once with a frontend. We are sure that both layers work fine together

## Pros&Cons

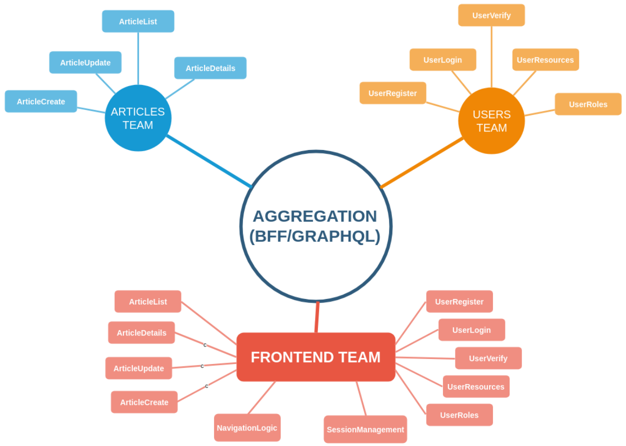
+

- Everything at one place
- Probably - at one pull request we can add new feature
- Deployed once with a frontend. We are sure that both layers work fine together

−

- Vertical scalability
- Complex inside
- A lot of legacy code can exist
- High entry threshold

## Microservices



## Pros&Cons

+

- Working parallel

## Pros&Cons

+

- Working parallel
- Many smaller codebases

## Pros&Cons

+

- Working parallel
- Many smaller codebases
- This codebases are really small

## Pros&Cons

+

- Working parallel
- Many smaller codebases
- This codebases are really small
- Microservices are trendy now

## Pros&Cons

+

- Working parallel
- Many smaller codebases
- This codebases are really small
- Microservices are trendy now
- We can use a lot of cool tools

## Pros&Cons

+

- Working parallel
- Many smaller codebases
- This codebases are really small
- Microservices are trendy now
- We can use a lot of cool tools
- We are skipping performance and scalability

## Pros&Cons

+

- Working parallel
- Many smaller codebases
- This codebases are really small
- Microservices are trendy now
- We can use a lot of cool tools
- We are skipping performance and scalability

−

- Many codebases - additional automation costs
- Integration needed (+ more testing etc.)
- Infrastructural costs
- Eventual consistency
- ...
- Just a lot of work more

## Why do we want to decouple everything?

## Why do we want to decouple everything?

Or in other words what's wrong with monolithic systems?

## Why do we want to decouple everything?

Or in other words what's wrong with monolithic systems?

- We can't understand a big amount of code

## Why do we want to decouple everything?

Or in other words what's wrong with monolithic systems?

- We can't understand a big amount of code
- So, we can't easily add new features

## Why do we want to decouple everything?

Or in other words what's wrong with monolithic systems?

- We can't understand a big amount of code
- So, we can't easily add new features
- They are complex

## Why do we want to decouple everything?

Or in other words what's wrong with monolithic systems?

- We can't understand a big amount of code
- So, we can't easily add new features
- They are complex
- One codebase can be developed effectively by 100 people?

## Why do we want to decouple everything?

Or in other words what's wrong with monolithic systems?

- We can't understand a big amount of code
- So, we can't easily add new features
- They are complex
- One codebase can be developed effectively by 100 people?
- Let's skip performance and scalability

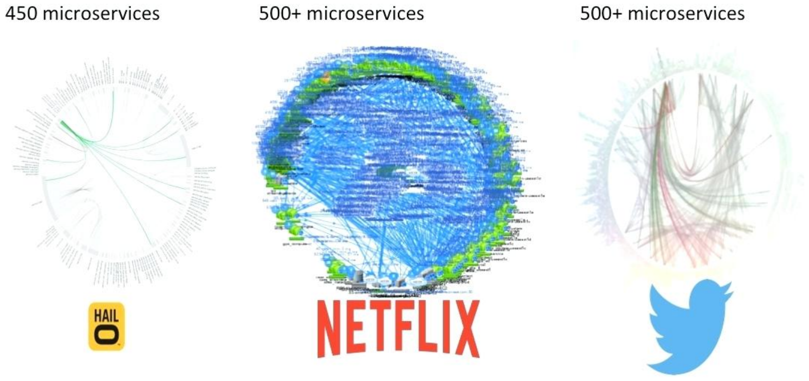
Always? It depends. The good architecture also relates to monoliths.

One of the biggest advantages of distributed systems is that knowledge is also distributed through teams. They can focus on a single part of business requirements.
Also, It's possible to easily **replace** each part.

But, everything is loosely coupled...

## Architectuure!

Do microservices solve your problems? ARE YOU SURE?



Source:
Netflix: http://www.slideshare.net/BruceWong3/the-case-for-chaos
Twitter: https://twitter.com/adrianco/status/441883572618948608
Hail-o: https://sudo.hailoapp.com/services/2015/03/09/journey-into-a-microservice-world-part-3/

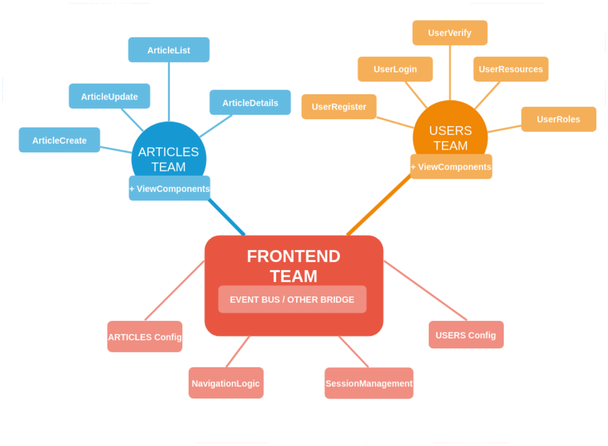
## Okay... 500 Microservices but...

Let's connect everything on UI...

## Okay... 500 Microservices but...

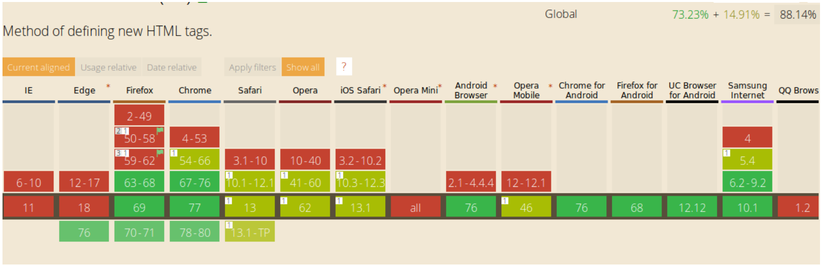
Let's connect everything on UI...

# Microfrontends



## Webcomponents

Additional HTML tags defined in java script files, which can be shared between a lot of pages.

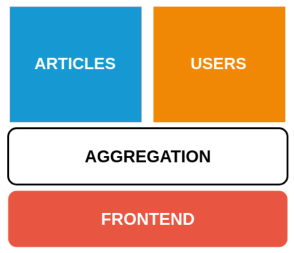


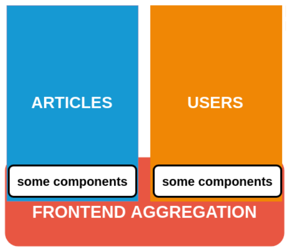
Method of defining new HTML tags.

Global 73.23% + 14.91% = 88.14%

| IE | Edge * | Firefox | Chrome | Safari | Opera | iOS Safari * | Opera Mini * | Android Browser * | Opera Mobile * | Chrome for Android | Firefox for Android | UC Browser for Android | Samsung Internet | QQ Brows |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 2-49 | | | | | | | | | | | | |
| | | [2] [1] 50-58 | 4-53 | | | | | | | | | | 4 | |
| | | [3] [1] 59-62 | [1] 54-66 | 3.1-10 | 10-40 | 3.2-10.2 | | | | | | | [1] 5.4 | |
| 6-10 | 12-17 | 63-68 | 67-76 | [1] 10.1-12.1 | [1] 41-60 | [1] 10.3-12.3 | | 2.1-4.4.4 | 12-12.1 | | | | 6.2-9.2 | |
| 11 | 18 | 69 | 77 | [1] 13 | [1] 62 | [1] 13.1 | all | 76 | [1] 46 | 76 | 68 | 12.12 | 10.1 | 1.2 |
| | 76 | 70-71 | 78-80 | [1] 13.1-TP | | | | | | | | | | |

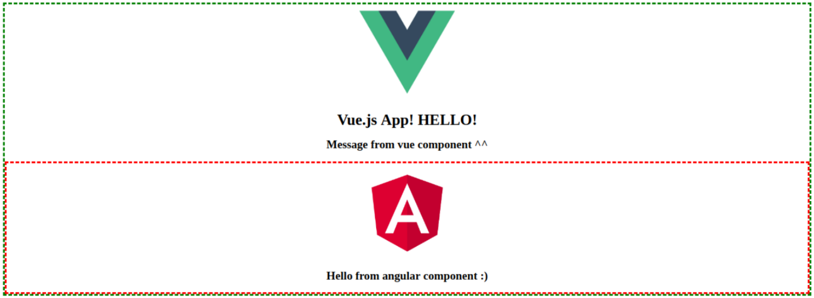
## Comparison


ARTICLES
USERS
AGGREGATION
FRONTEND


ARTICLES
USERS
some components
some components
FRONTEND AGGREGATION

## Multi-framework

# Multi-framework

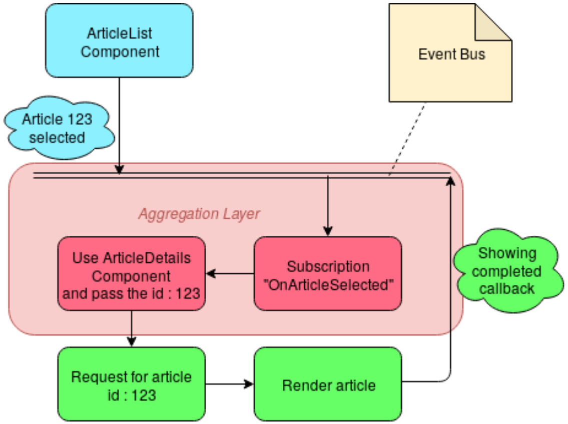

Vue.js App! HELLO!

Message from vue component ^^

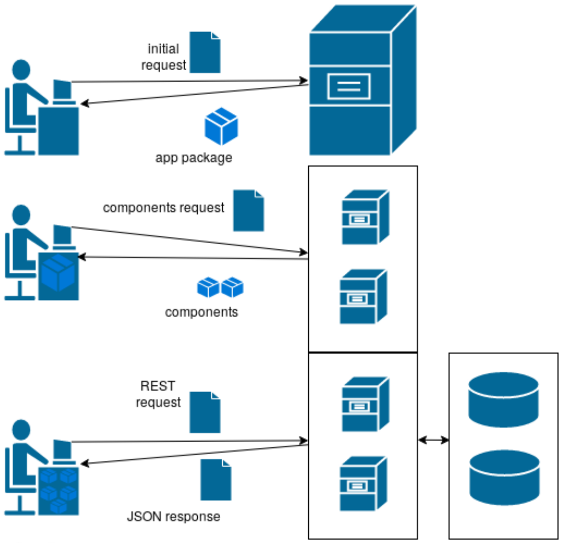
Hello from angular component :)

## DEMO TIME

## Components messaging


ArticleList Component
Event Bus
Article 123 selected
Aggregation Layer
Subscription "OnArticleSelected"
Use ArticleDetails Component and pass the id : 123
Showing completed callback
Request for article id : 123
Render article

## Component loading

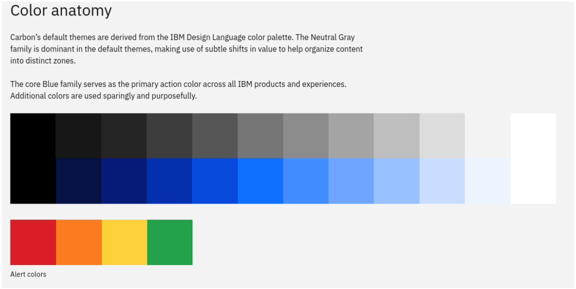


## UI Consistency

Many teams, many components. How to manage this mess?

## UI Consistency

Many teams, many components. How to manage this mess?

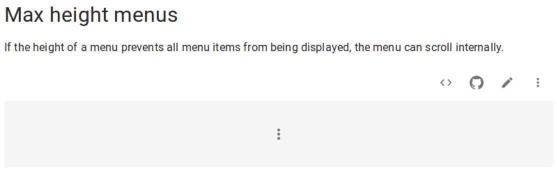

Color anatomy

Carbon's default themes are derived from the IBM Design Language color palette. The Neutral Gray family is dominant in the default themes, making use of subtle shifts in value to help organize content into distinct zones.

The core Blue family serves as the primary action color across all IBM products and experiences. Additional colors are used sparingly and purposefully.

Figure: IBM - color anatomy

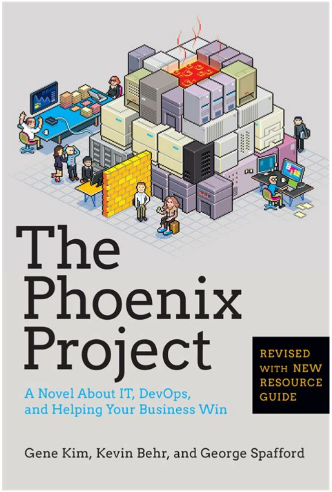
## UI Consistency


Max height menus

If the height of a menu prevents all menu items from being displayed, the menu can scroll internally.

Figure: Material UI - menu height

## For that - DevOps

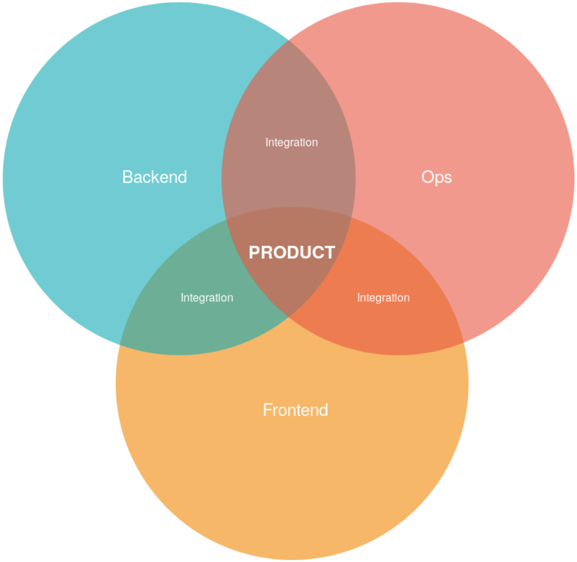

The Phoenix Project
A Novel About IT, DevOps, and Helping Your Business Win
REVISED WITH NEW RESOURCE GUIDE
Gene Kim, Kevin Behr, and George Spafford

## Dev & Ops

DevOps culture is mandatory at the microservices environment, but is it sufficient with microfrontends?

## Cross functional teams/Cross functional people

## Cross functional teams/Cross functional people



## Questions?

- My blog page: https://orchowskia.com
- Twitter: @orchowski