Full-stack Microservices

William Bartlett

@ w.bartlett@treeptik.fr @bartlettstarman

punkstarman punkstarman

Treeptik

15 November, 2018

whoami?

William Bartlett, Level 23 programmer

- Agile Coach, Java dev, Container enthusiast
- Web Component nut (Polymer)
- Former doctoral student

“Use the right tool for the job”

TREEPTIK.

Section 1

Introduction

Microservices

Advantages

- separation (*"divide and conquer"*)
- autonomy
- automation
- modularity
- scaling
- tech stack transitions

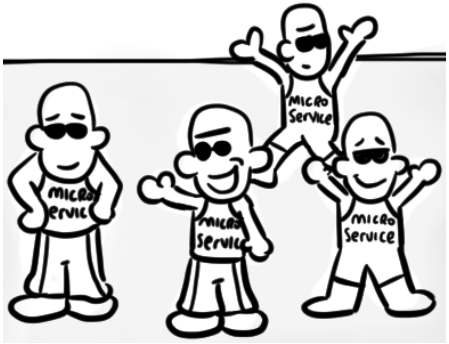
Microservices

Price to pay

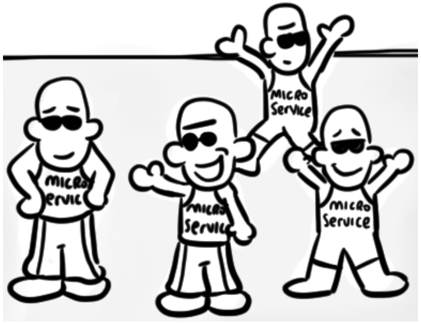
- complex integration
- consistency or availability
- hard boundaries

Microservices

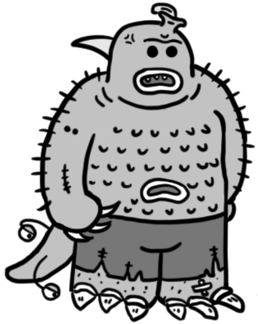
Microservices are cool
☺



Microservices

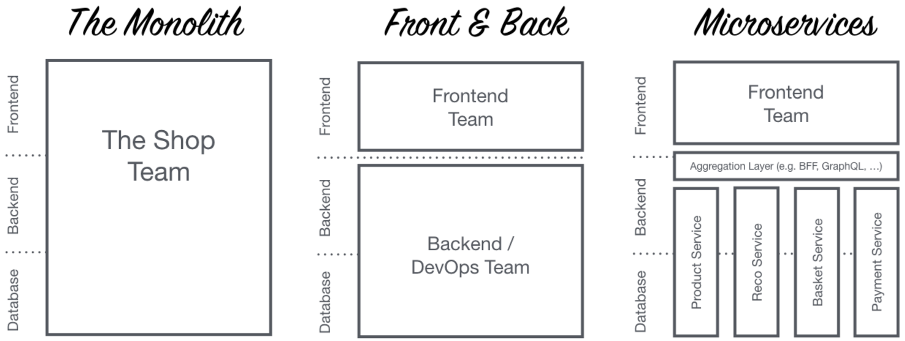


Server-side



Client-side

Microservices



Micro Frontends: https://micro-frontends.org/

Issue

Issue

How to build a large product with entirely autonomous multi-disciplinary teams?

Bring microservices to the front-end

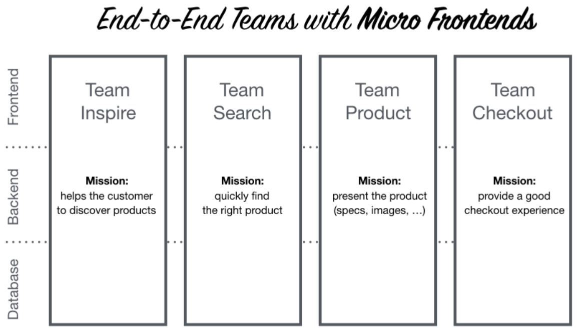
- Tech solution: distributed web components
- Org solution: Atomic Design

Outline

Section 2

It's All Relative

Micro Frontends


End-to-End Teams with Micro Frontends
Frontend
Backend
Database
Team Inspire
Mission: helps the customer to discover products
Team Search
Mission: quickly find the right product
Team Product
Mission: present the product (specs, images, ...)
Team Checkout
Mission: provide a good checkout experience

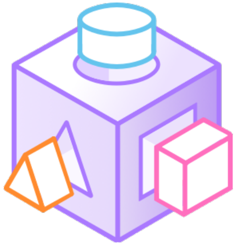
Micro Frontends : https://micro-frontends.org/

Project Mosaic, Zalando

- Skipper (router), InnKeeper (router config API)
- Tailor (layout), Quilt (layout storage)
- Shaker (components)
- Tessellate (SSR)

`https://www.mosaic9.org/`

OpenComponents, OpenTable



- API for UI components
- SSR or not

https://opencomponents.github.io/

TREEPTIK.

Metaframework, CanopyTax

Single SPA

- Runtime stitching of micro-frontends.

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

Example : https://single-spa.surge.sh/

Microservice Websites, Gustaf Nilsson Kotte

Microservice Websites

Scalable development of an evolvable system with great mobile performance.

- performance
- autonomy $\Rightarrow$ heterogeneity and scalability

- Manifesto
- Article
- Tools:
  - Edge-Side Includes
  - Client-Side Includes
  - Caching (Varnish)

Other options

- JSP/ASP.Net Fragments

Other options

- JSP/ASP.Net Fragments
- JSP Tag Library

Other options

- JSP/ASP.Net Fragments
- JSP Tag Library
- Struts Tiles

Other options

- JSP/ASP.Net Fragments
- JSP Tag Library
- Struts Tiles
- Portlets (Liferay)

Other options

- JSP/ASP.Net Fragments
- JSP Tag Library
- Struts Tiles
- Portlets (Liferay)
- `<iframe>`

Section 3

Web Components

Web Components

Web Components

Reusable, modular components for the web.

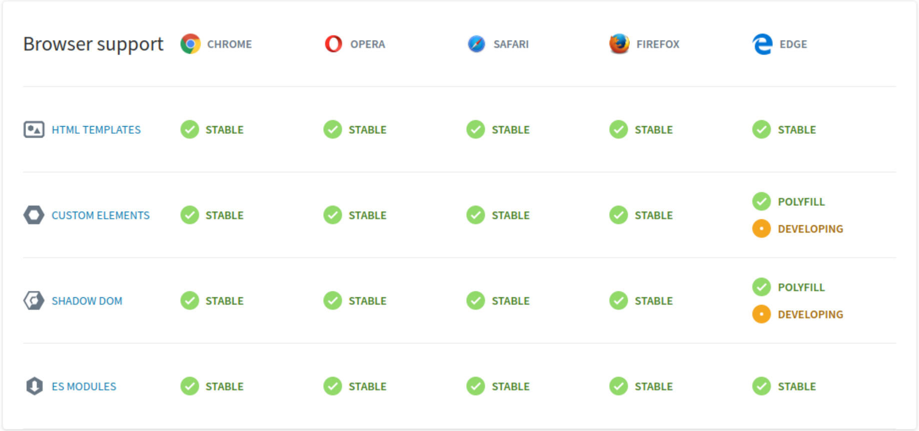
- 4 W3C specifications
- started in 2011

Demo

Find this demo on CodePen

https://codepen.io/punkstarman/project/editor/AEKqQg

Custom Elements



Browser support	CHROME	OPERA	SAFARI	FIREFOX	EDGE
HTML TEMPLATES	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE
CUSTOM ELEMENTS	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE	✓ POLYFILL ● DEVELOPING
SHADOW DOM	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE	✓ POLYFILL ● DEVELOPING
ES MODULES	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE

It's an HTML element

- attributes
- properties
- events
- styling

Frameworks

Frameworks:

- Polymer, SkateJS, Slim.js, x-tag, Bosonic, Stencil, ...
- React, Angular, Vue, ...

https://custom-elements-everywhere.com/

Added bonuses:

- two-way data-binding
- boilerplate reduction

Component libraries

`https://www.webcomponents.org/`

- Polymer (iron, paper, app, gold)
- Vaadin
- Google (Maps, YouTube)
- Predix UI

Web Components in the Wild

- Google (Polymer)
  - Chrome
  - YouTube, Drive, Contacts
  - Example app: Shop (https://shop.polymer-project.org/)
- Electronic Arts
- GitHub
- Simpla

Section 4

Atomic Design

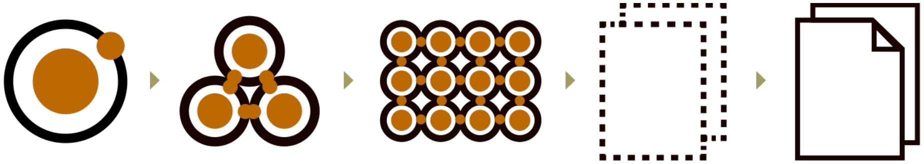
Atomic Design

Atomic Design

Methodology for creating and maintaining a graphic design system

Brad Frost, 2016
`http://atomicdesign.bradfrost.com/`

Atomic Design



atoms molecules organisms templates screens

TREEPTIK.

Atom

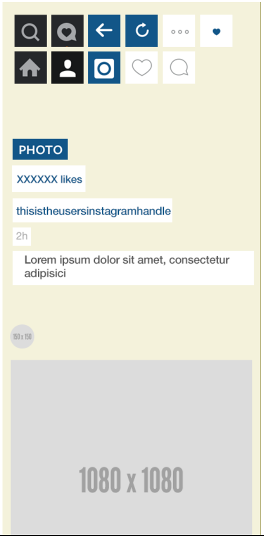
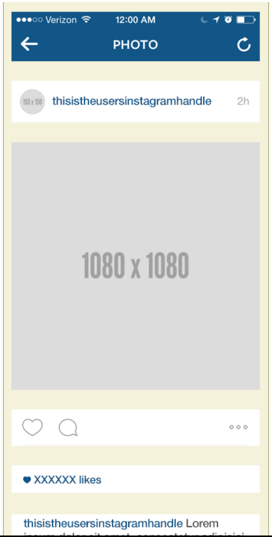


PHOTO
XXXXXX likes
thisistheusersinstagramhandle
2h
Lorem ipsum dolor sit amet, consectetur adipisici
150 x 150
1080 x 1080

- indivisible
- graphical identity

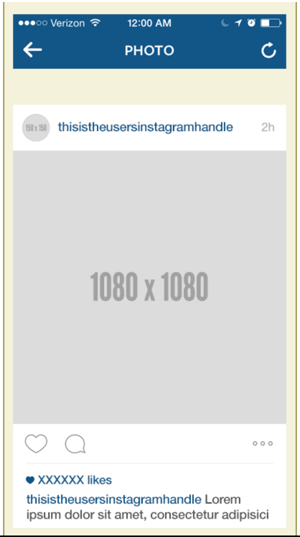
Molecule


●●●○○ Verizon
12:00 AM
PHOTO
150 x 150
thisistheusersinstagramhandle
2h
1080 x 1080
XXXXXX likes
thisistheusersinstagramhandle Lorem

- connected atoms
- purpose

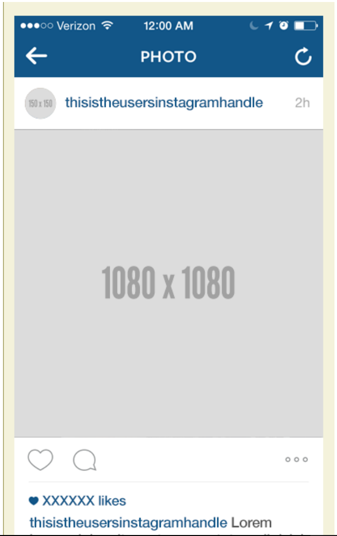
TREEPTIK.

Organism


●●●○○ Verizon
12:00 AM
PHOTO
150 x 150
thisistheusersinstagramhandle
2h
1080 x 1080
XXXXXX likes
thisistheusersinstagramhandle Lorem ipsum dolor sit amet, consectetur adipisici

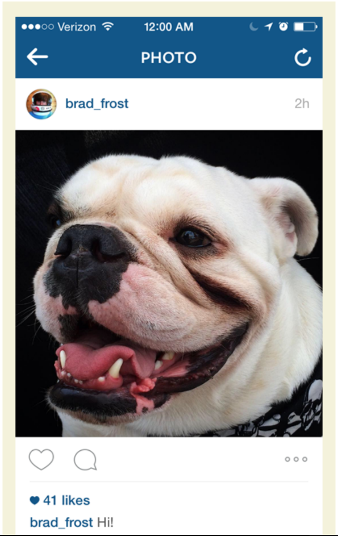
- composed atoms, molecules or other organisms
- complexity

Template


●●●○○ Verizon
12:00 AM
PHOTO
150 x 150
thisistheusersinstagramhandle
2h
1080 x 1080
XXXXXX likes
thisistheusersinstagramhandle Lorem

- layout of organisms
- generic page

Page



- template + data
- proof of concept

Section 5

Full-stack Microservices

Microservices + Web Components

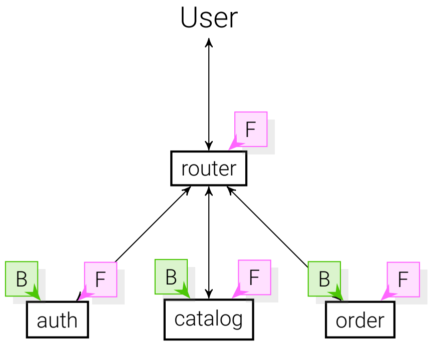
- integration over HTTP
- lazy loading
- fault tolerance

Web Components + Atomic Design

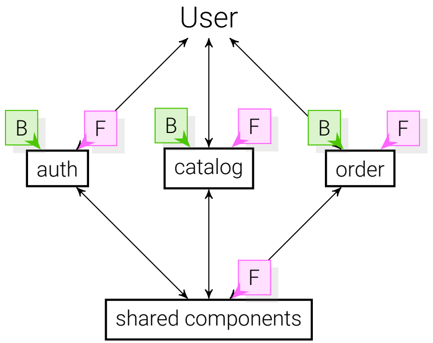
Atomic Design is the methodology needed to develop a system of components efficiently.
Web Components are a solution that can enable Atomic Design.

DNF: integration components (AJAX, state)

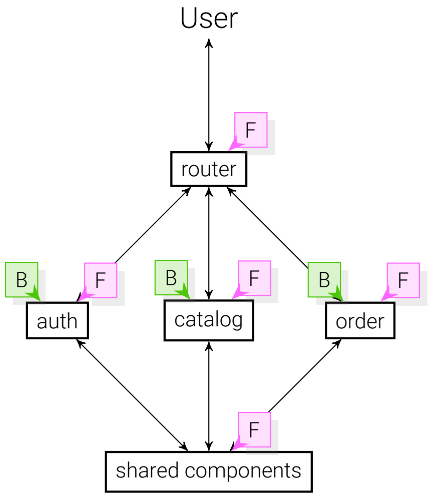
Full-stack Microservices



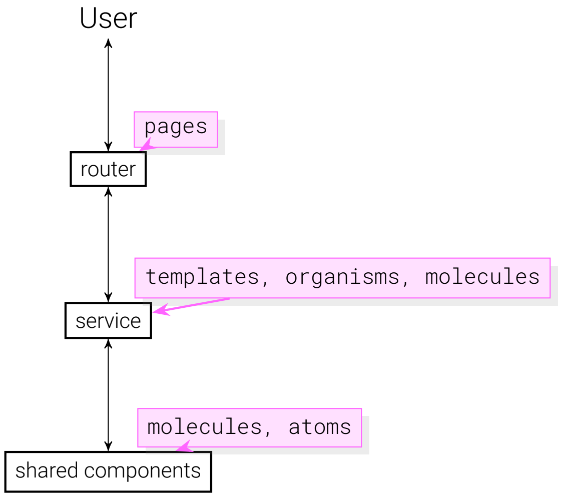
Full-stack Microservices



Full-stack Microservices



Full-stack Microservices



Section 6

Case Study

O gods of Demo,
we beseach thee

Section 7

Conclusion

Conclusion

- + Autonomous teams
- + expose API and components
- + RAD
- + choice of framework

Conclusion

- + Autonomous teams
- + expose API and components
- + RAD
- + choice of framework

- — strong coupling between API and UI
- — library design > application design
- — difficult to isolate dependencies

Future Work

- Dependency and Configuration Injection

Future Work

- Dependency and Configuration Injection
- Encapsulation of third party services:

Future Work

- Dependency and Configuration Injection
- Encapsulation of third party services:
  - Auth0, Keycloak, Okta, ...

Future Work

- Dependency and Configuration Injection
- Encapsulation of third party services:
  - Auth0, Keycloak, Okta, ...
  - Paypal

Future Work

- Dependency and Configuration Injection
- Encapsulation of third party services:
  - Auth0, Keycloak, Okta, ...
  - Paypal
- OAuth2 via Web Components: trust?

Future Work

- Dependency and Configuration Injection
- Encapsulation of third party services:
  - Auth0, Keycloak, Okta, ...
  - Paypal
- OAuth2 via Web Components: trust?
- Logs, monitoring, instrumentation ...