Microservice Architectures (and more)

Michael Hilton **Rohan Padhye**

Inspirations:
Martin Fowler (http://martinfowler.com/articles/microservices.html)
Josh Evans @ Netflix (https://www.youtube.com/watch?v=CZ3wIuvmHeM)
Matt Ranney @ Uber (https://www.youtube.com/watch?v=kb-m2fasdDY)
Christopher Meiklejohn & Filibuster (http://filibuster.cloud)

institute for SOFTWARE RESEARCH
Carnegie Mellon University
School of Computer Science

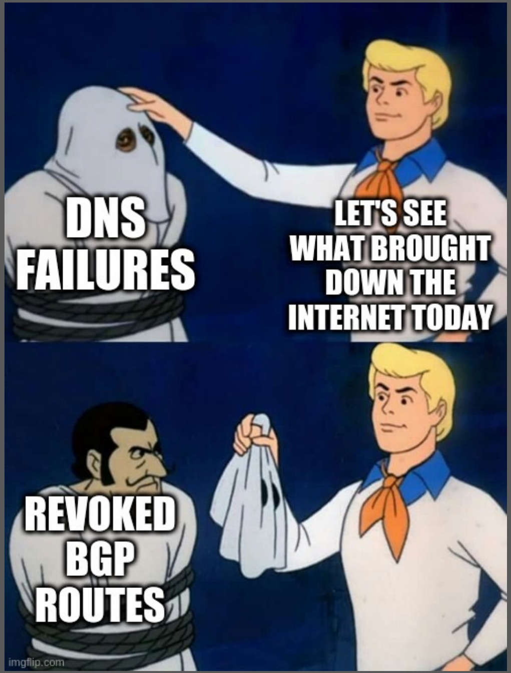
Administrativia

- Homework 2 due Friday (Oct 7).
- Recitation this week: midterm review (**come prepared**!)
    - Work through problems on the previous midterms – many students found this helpful.
    - Any questions on the previous midterm questions – bring them to recitation to discuss as a class.
- Midterm on October 12th (in class, regular timing).

Learning Goals

- Contrast the monolithic application design with a modular design based on microservices.
- Reason about how architectural choices affect software quality and process attributes.
- Reason about tradeoffs of microservices architectures.

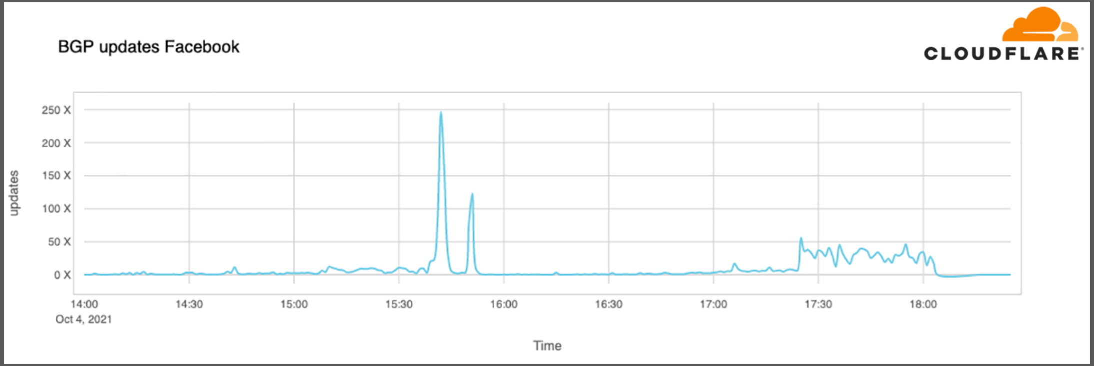
There has been a failure
on the INTERNETS

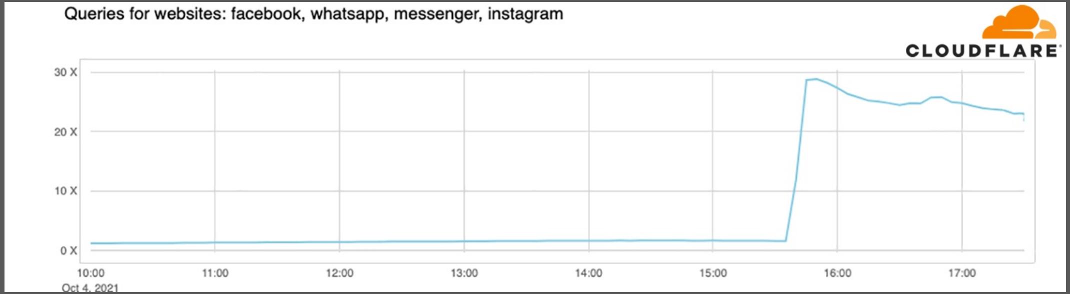

DNS FAILURES
LET'S SEE WHAT BROUGHT DOWN THE INTERNET TODAY
REVOKED BGP ROUTES
imgflip.com

Facebook Network Engineering Team after doing `git push` of BGP changes:
This little maneuvre is gonna cost us 6 billion

Facebook on Oct 4, 2021

Source: https://blog.cloudflare.com/october-2021-facebook-outage/

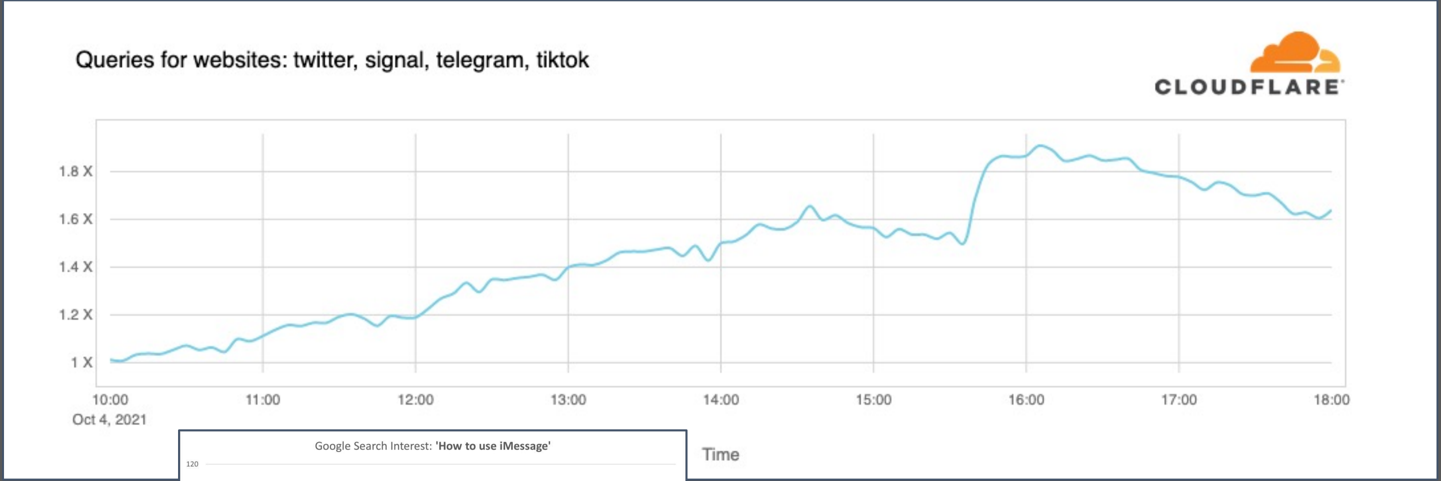


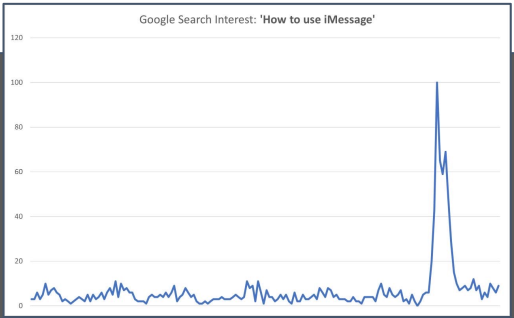


Carnegie Mellon University
School of Computer Science

Facebook on Oct 4, 2021

Source: https://blog.cloudflare.com/october-2021-facebook-outage/



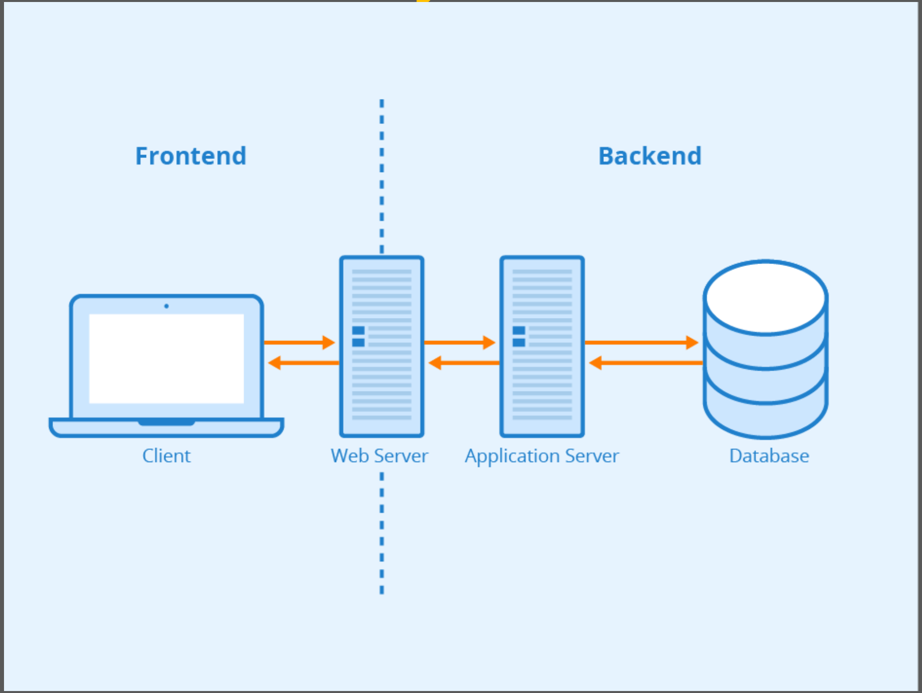


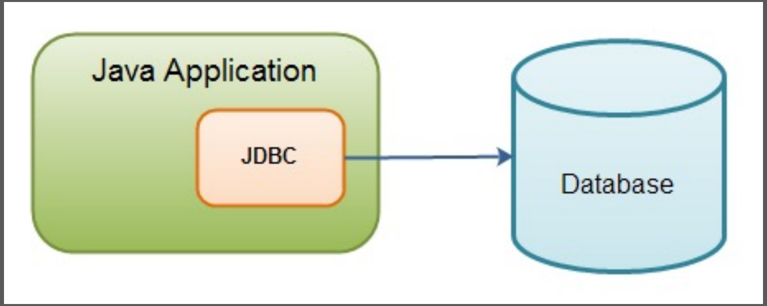
Some interesting insights about the dependency web of the Web:
https://www.synergylabs.org/yuvraj/docs/Kashaf_IMC2020_WebDep.pdf

Microservice architectures

MONOLITHS

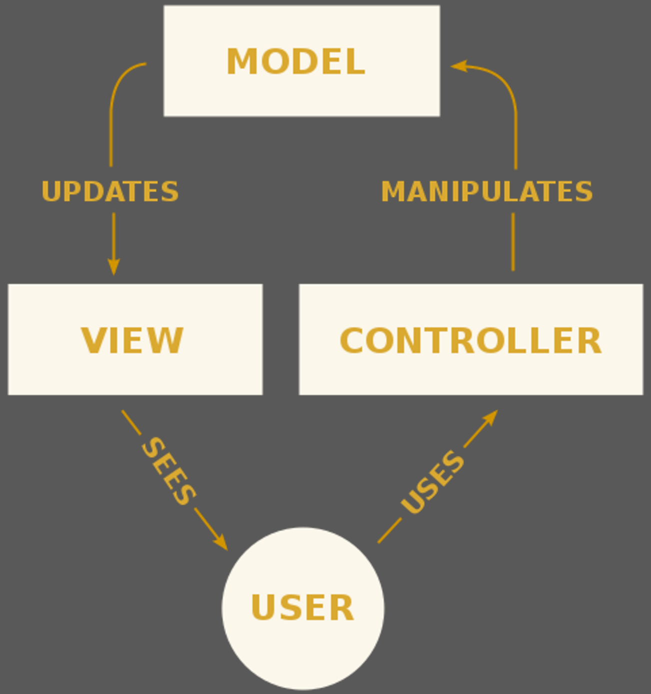
Monolithic styles





Source: https://www.seobility.net (CC BY-SA 4.0)

Monolithic styles: MVC Pattern (e.g. Mayan)



Monoliths

What are the consequences of this architecture? On:

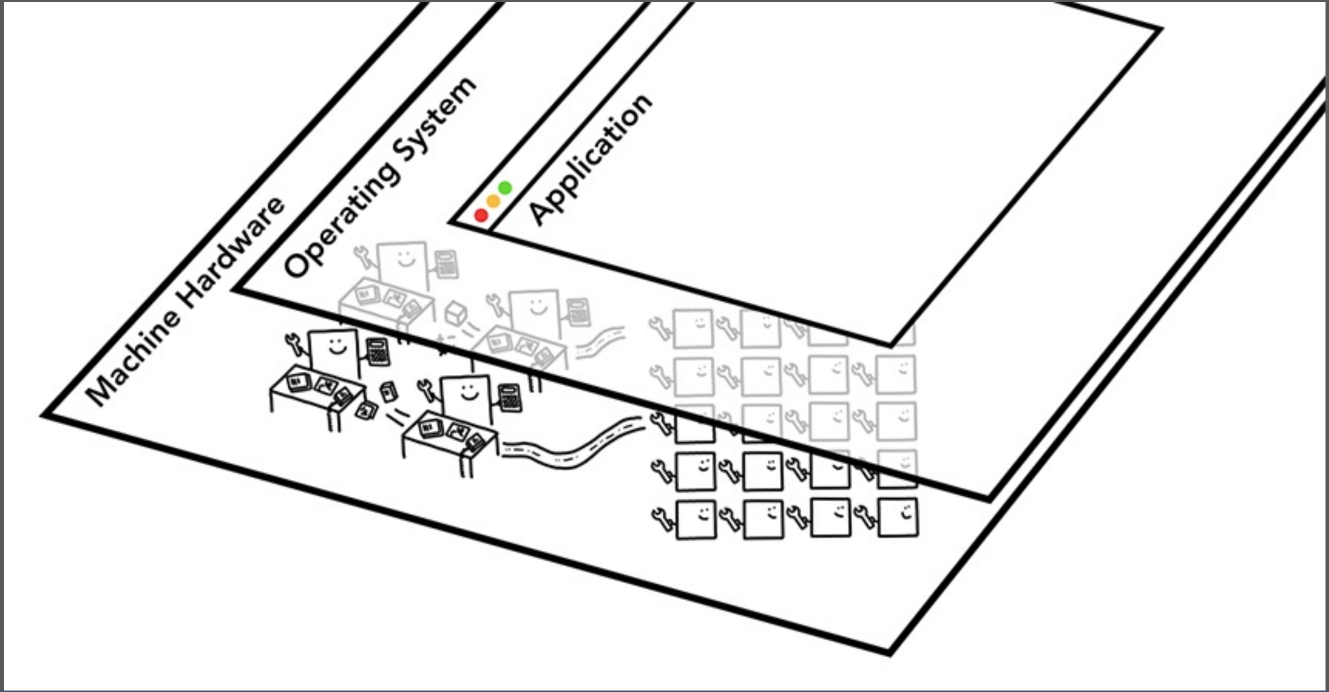
- Scalability
- Reliability
- Performance
- Development
- Maintainability
- Evolution
- Testability
- Ownership
- Data Consistency

Separation of concerns

SERVICE-BASED ARCHITECTURE

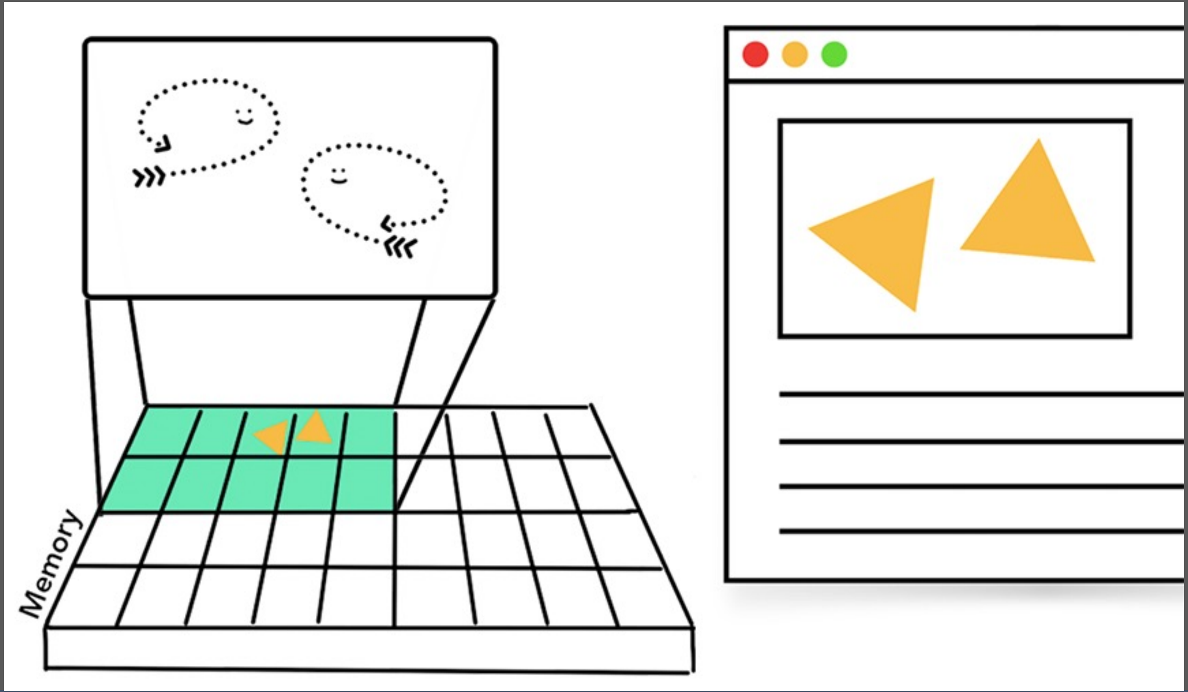
Chrome

Web Browsers



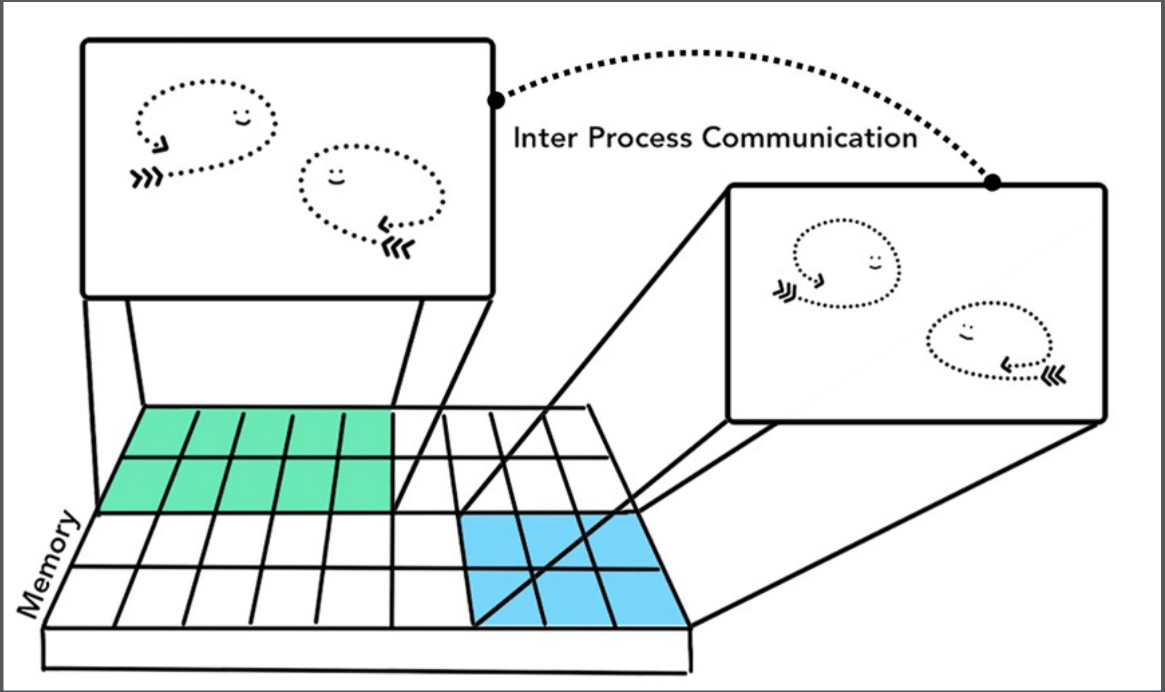
Source: https://developers.google.com/web/updates/2018/09/inside-browser-part1 (CC BY 4.0)

Browser: A multi-threaded process



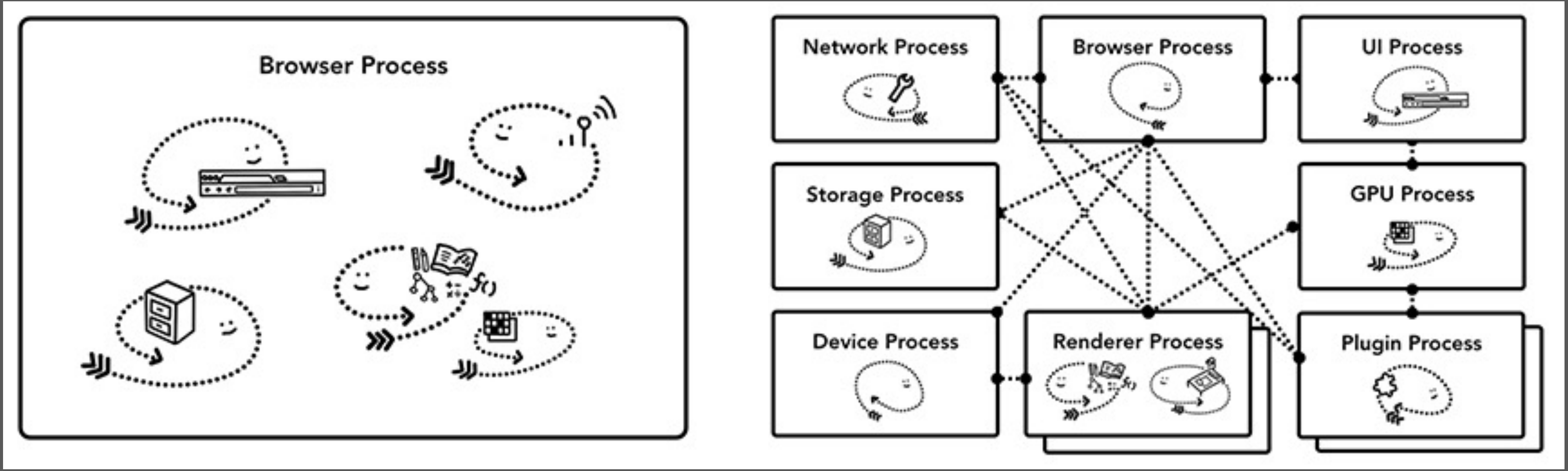
Source: https://developers.google.com/web/updates/2018/09/inside-browser-part1 (CC BY 4.0)

Multi-process browser with IPC



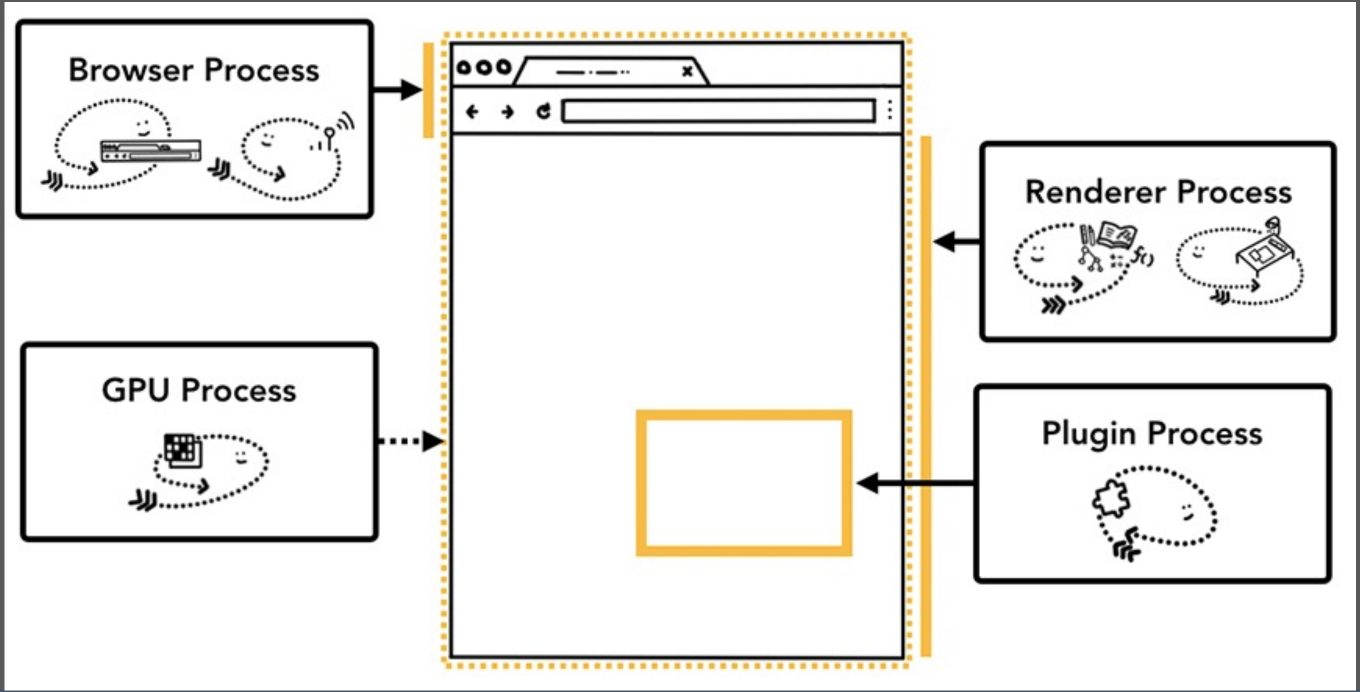
Source: https://developers.google.com/web/updates/2018/09/inside-browser-part1 (CC BY 4.0)

Browser Architectures



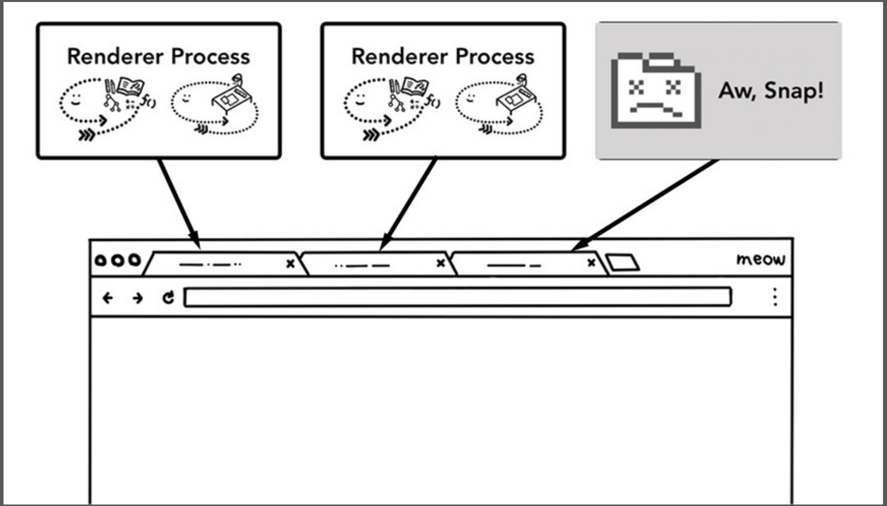
Source: https://developers.google.com/web/updates/2018/09/inside-browser-part1 (CC BY 4.0)

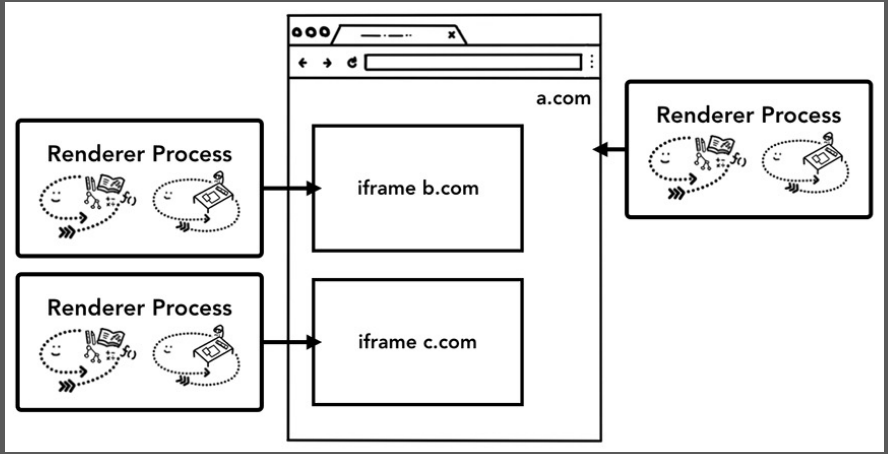
Service-based browser architecture



Source: https://developers.google.com/web/updates/2018/09/inside-browser-part1 (CC BY 4.0)

Service-based browser architecture





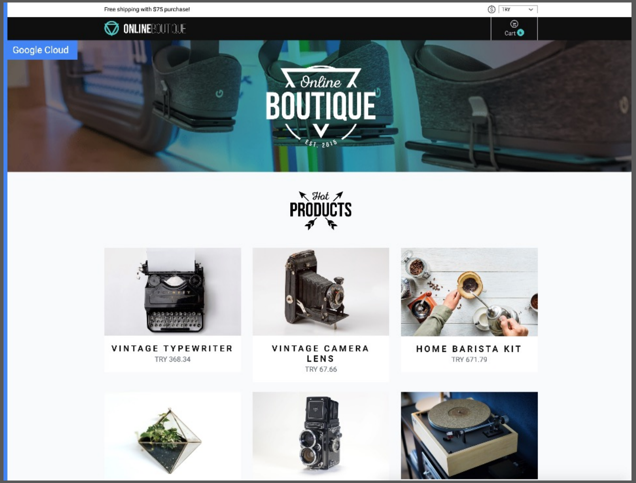
Source: https://developers.google.com/web/updates/2018/09/inside-browser-part1 (CC BY 4.0)

Taking it further

MICROSERVICES

Hipster Shop

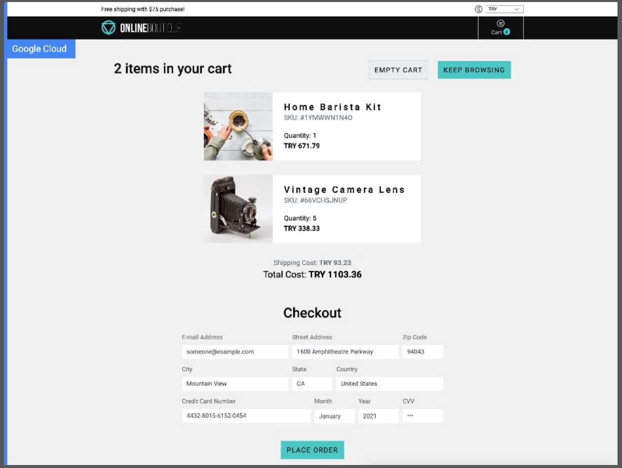
Hipster Shop User Interface



VINTAGE TYPEWRITER
TRY 368.34

VINTAGE CAMERA LENS
TRY 67.66

HOME BARISTA KIT
TRY 671.79



2 items in your cart

Home Barista Kit
SKU: #1YMWWN1N4O
Quantity: 1
TRY 671.79

Vintage Camera Lens
SKU: #66VCHSJNUP
Quantity: 5
TRY 338.33

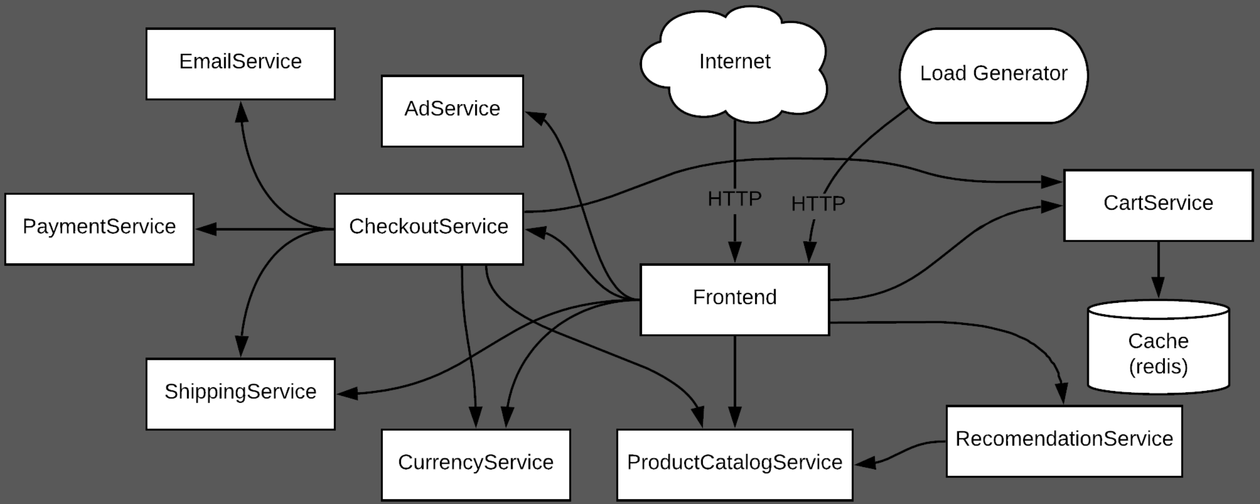
Shipping Cost: TRY 93.23
Total Cost: TRY 1103.36

Checkout

E-mail Address: someone@example.com
Street Address: 1600 Amphitheatre Parkway
Zip Code: 94043
City: Mountain View
State: CA
Country: United States
Credit Card Number: 4432-8015-6152-0454
Month: January
Year: 2021
CVV: •••

https://github.com/GoogleCloudPlatform/microservices-demo

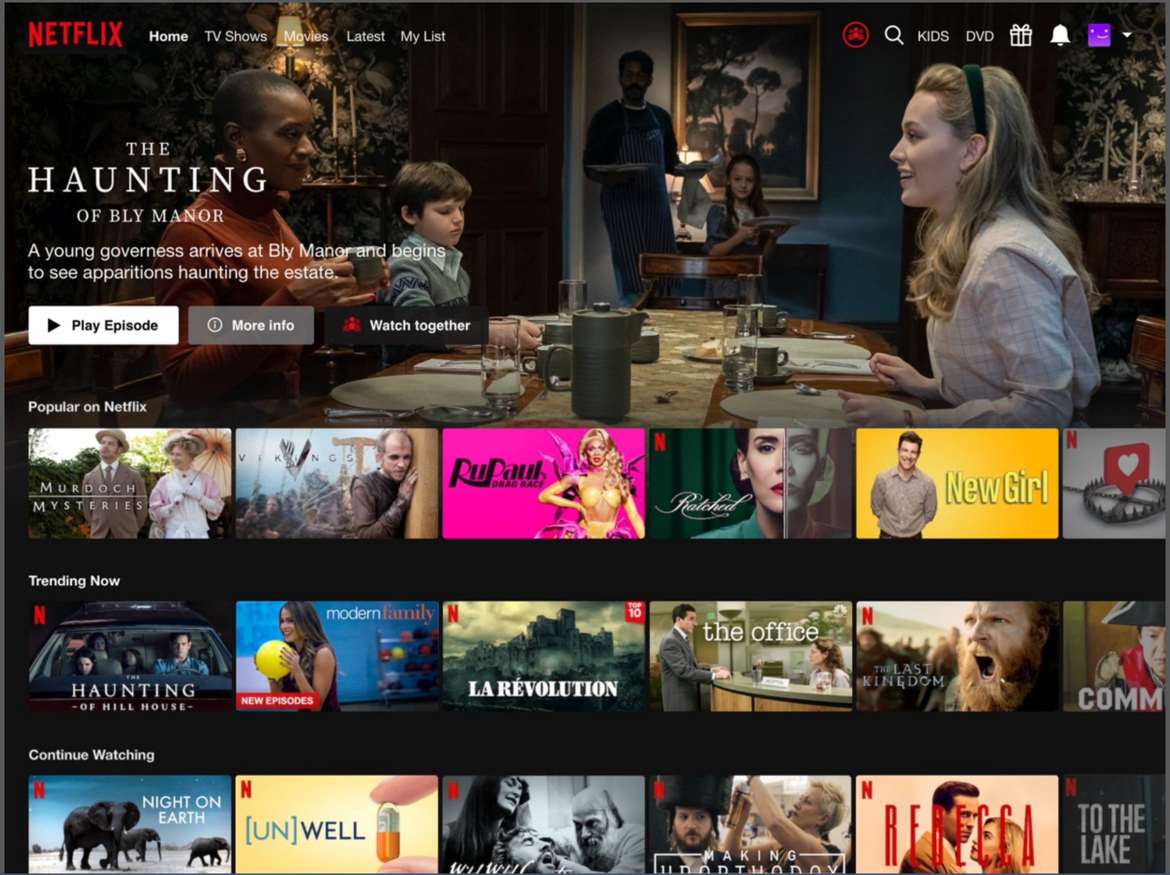
Hipster Shop Microservice Architecture



https://github.com/GoogleCloudPlatform/microservices-demo

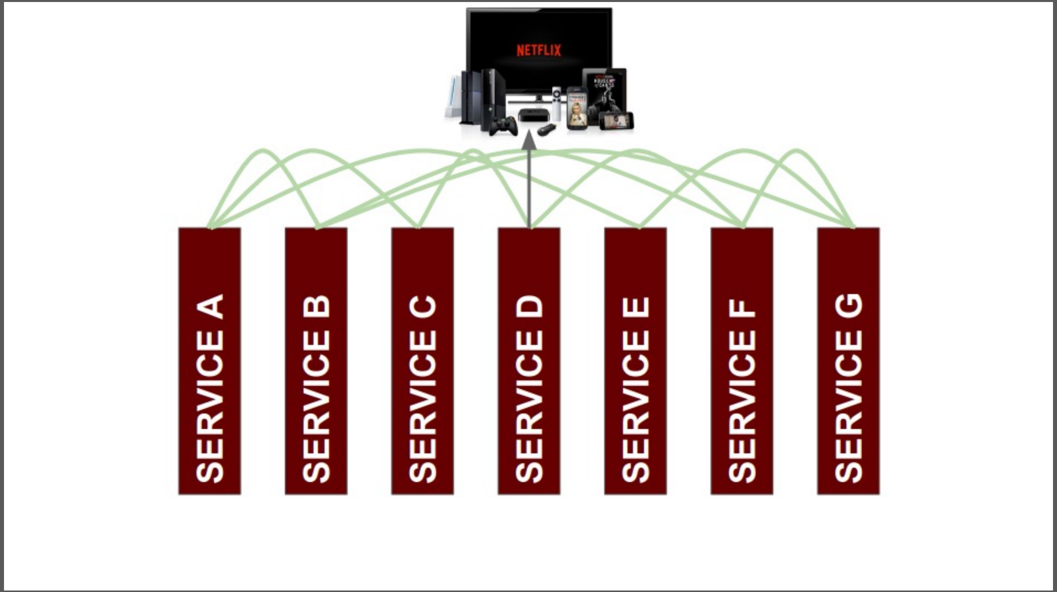
Netflix

Netflix



A young governess arrives at Bly Manor and begins to see apparitions haunting the estate.

AppBoot



Bookmarks

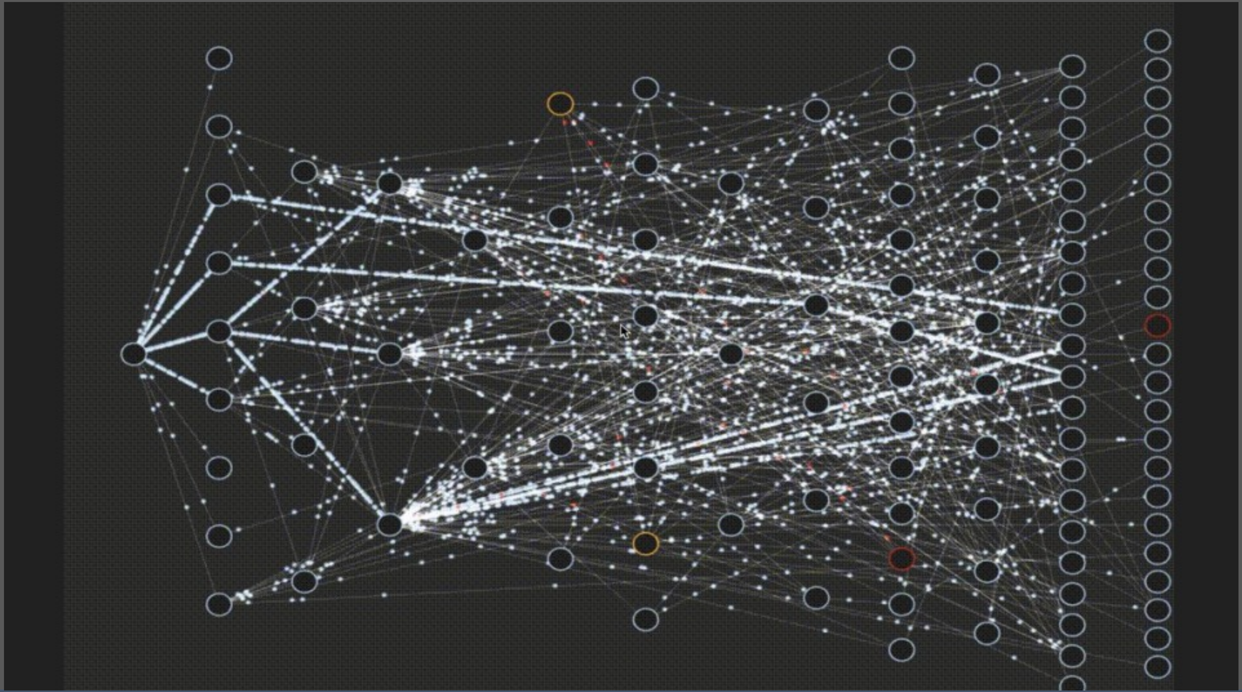
Recommendations

My List

Metrics

(as of 2016)

Netflix Microservices



(as of 2016)

https://www.youtube.com/watch?v=CZ3wIuvmHeM

Who uses Microservices?

COMCAST

NETFLIX

ebay

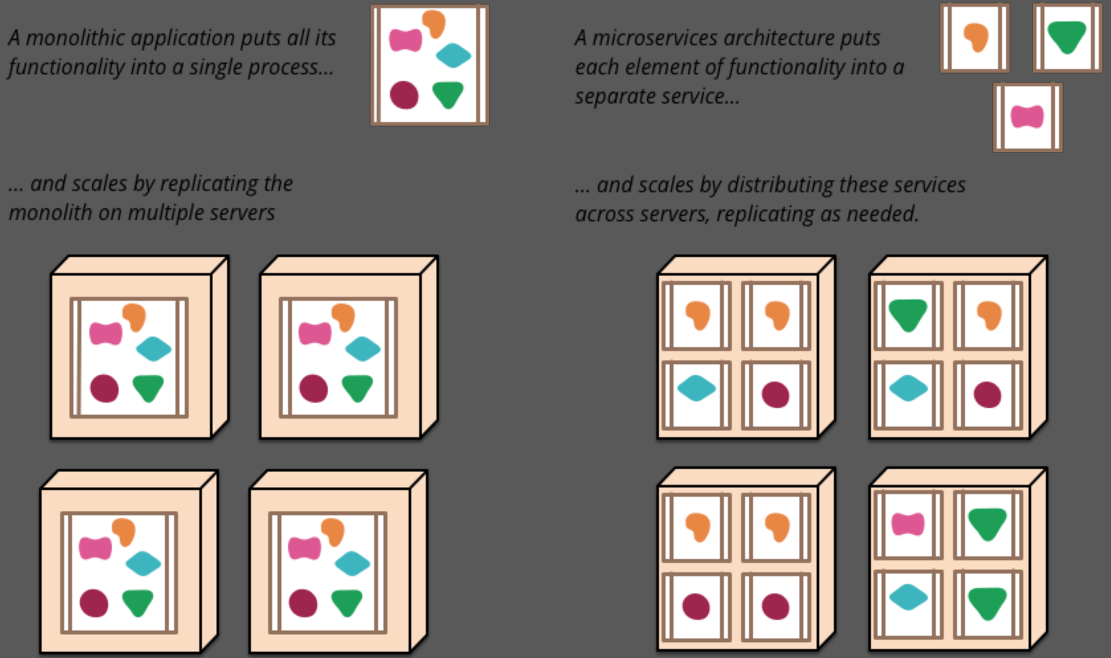
UBER GROUPON

Microservices

What are the consequences of this architecture? On:

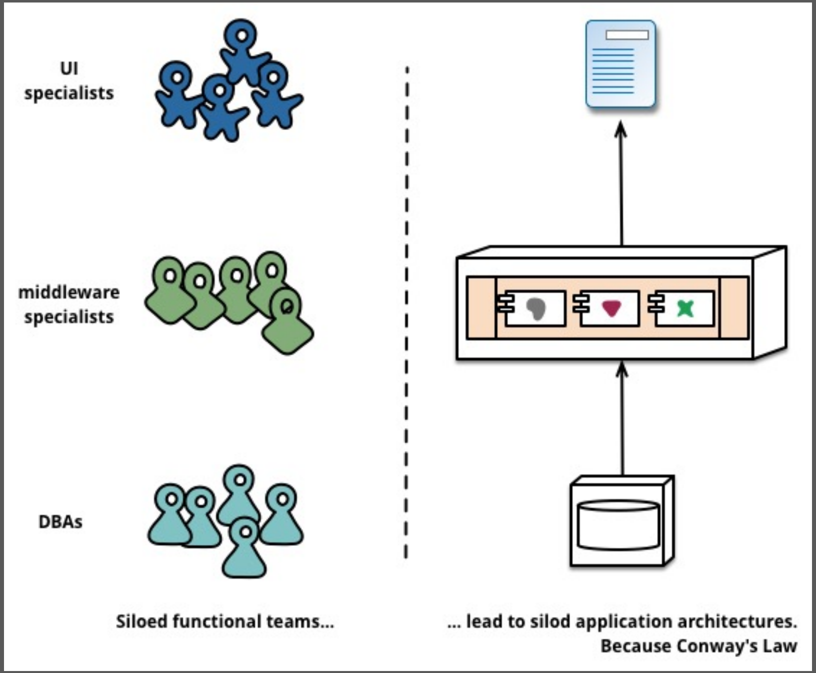
- Scalability
- Reliability
- Performance
- Development
- Maintainability
- Evolution
- Testability
- Ownership
- Data Consistency

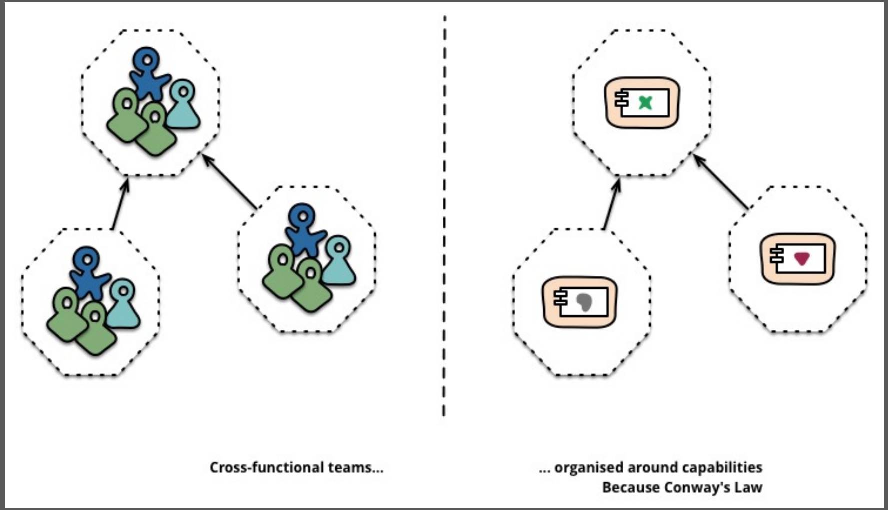
Scalability


A monolithic application puts all its functionality into a single process...
... and scales by replicating the monolith on multiple servers
A microservices architecture puts each element of functionality into a separate service...
... and scales by distributing these services across servers, replicating as needed.

Source: http://martinfowler.com/articles/microservices.html

Team Organization (Conway's Law)

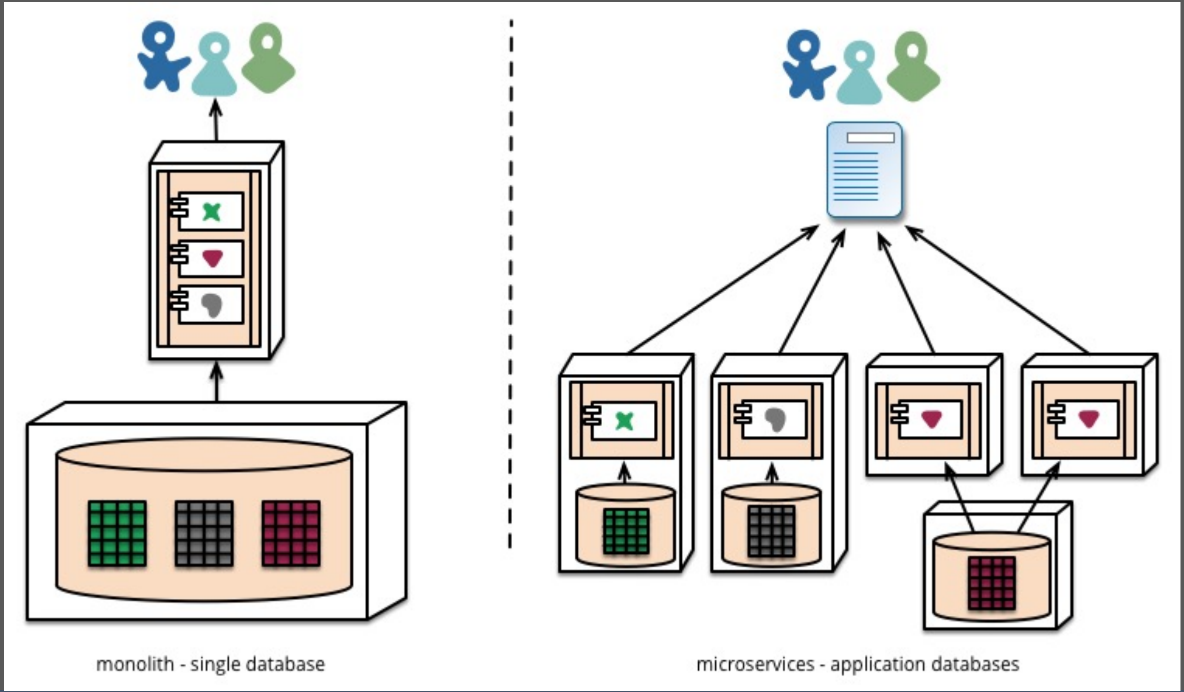

UI specialists
middleware specialists
DBAs
Siloed functional teams...
... lead to silod application architectures.
Because Conway's Law



"Products" not "Projects"

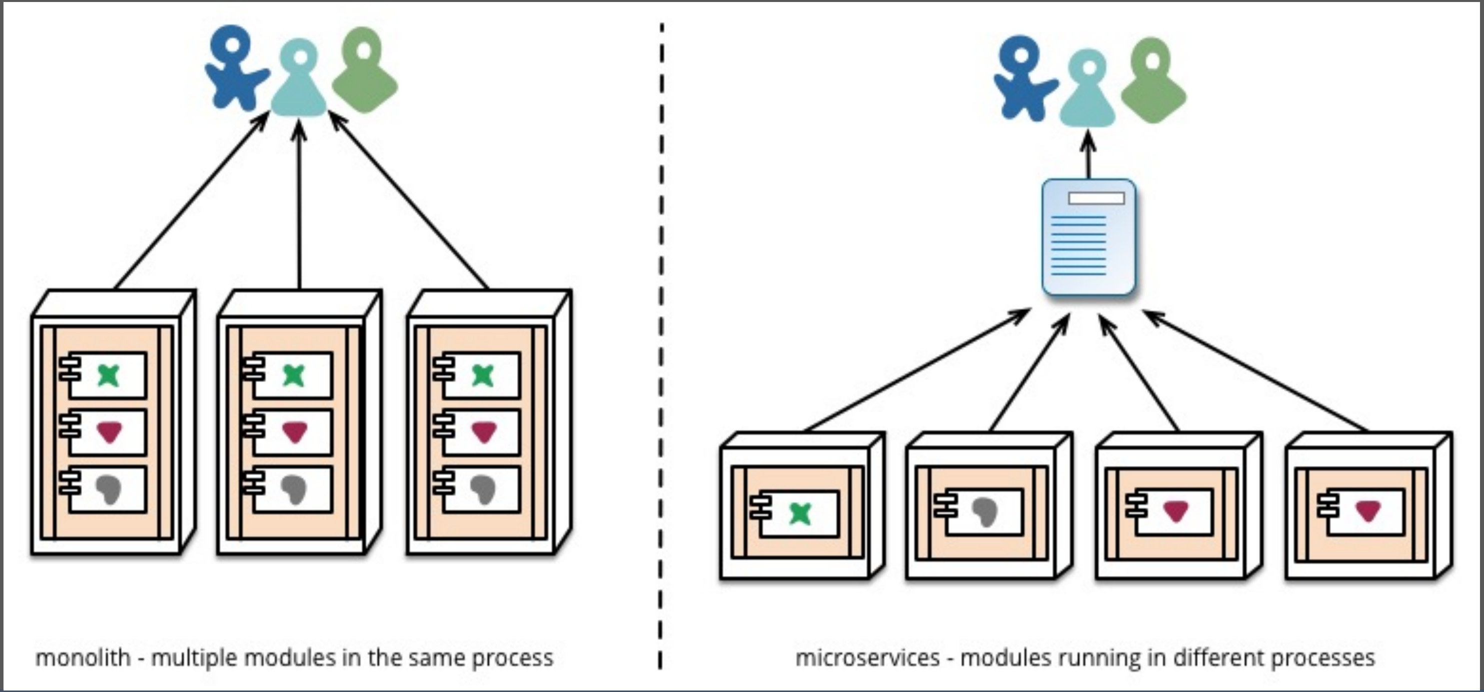
Source: http://martinfowler.com/articles/microservices.html

Data Management and Consistency


monolith - single database
microservices - application databases

Source: http://martinfowler.com/articles/microservices.html

Deployment and Evolution


monolith - multiple modules in the same process
microservices - modules running in different processes

Source: http://martinfowler.com/articles/microservices.html

Microservices

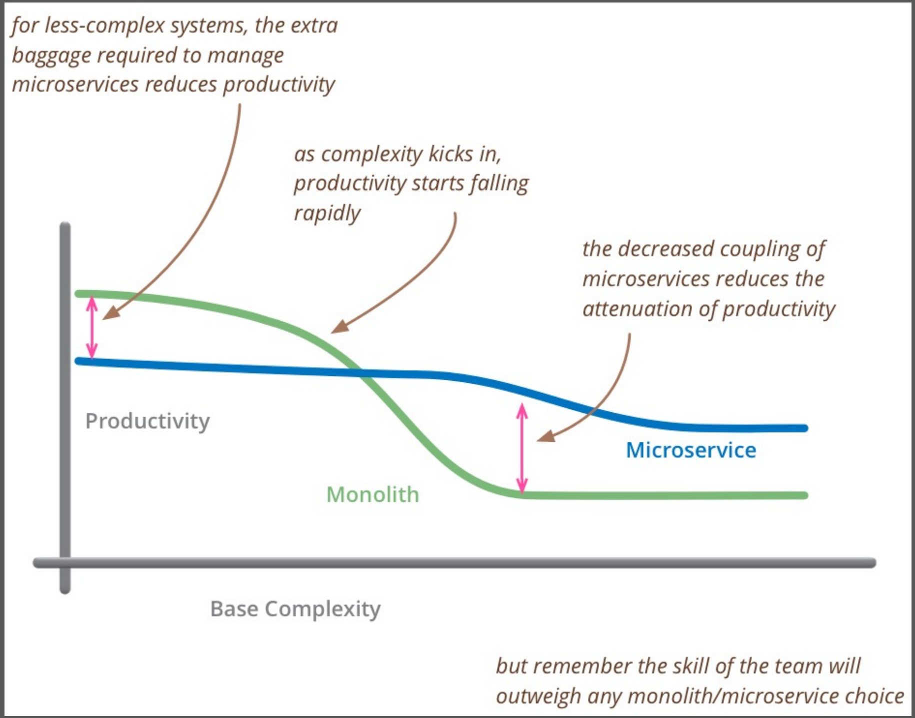
- Building applications as suite of small and easy to replace services
  - fine grained, one functionality per service (sometimes 3-5 classes)
  - composable
  - easy to develop, test, and understand
  - fast (re)start, fault isolation
  - modelled around business domain
- Interplay of different systems and languages
- Easily deployable and replicable
- Embrace automation, embrace faults
- Highly observable

Technical Considerations

- HTTP/REST/JSON/GRPC/etc. communication
- Independent development and deployment
- Self-contained services (e.g., each with own database)
  - multiple instances behind load-balancer
- Streamline deployment

Are microservices always the right choice?

Microservices overhead



Microservice challenges

- Complexities of distributed systems
  - network latency, faults, inconsistencies
  - testing challenges
- Resource overhead, RPCs
  - Requires more thoughtful design (avoid ”chatty” APIs, be more coarse-grained)_
- Shifting complexities to the network
- Operational complexity
- Frequently adopted by breaking down monolithic application
- HTTP/REST/JSON communication
  - Schemas?

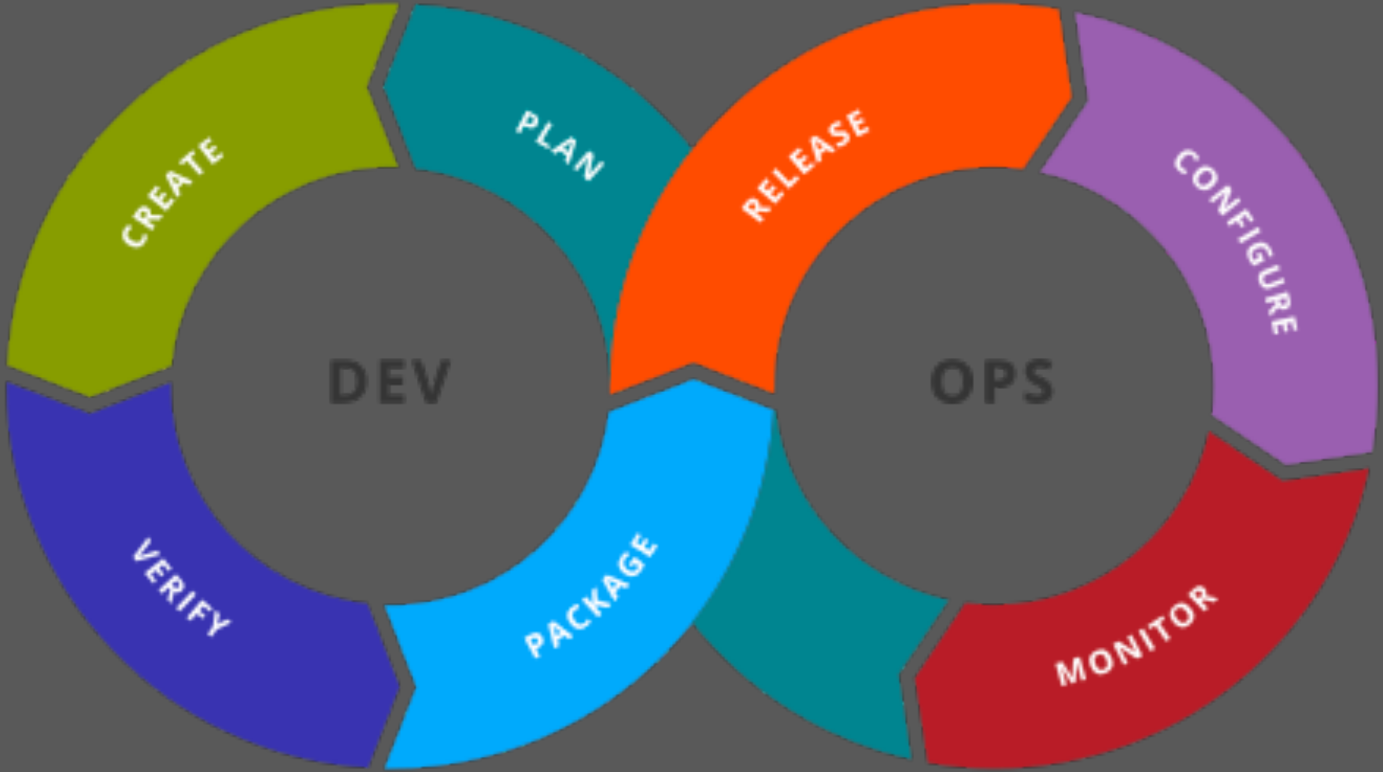
Taking it to the extreme

SERVERLESS

Serverless (Functions-as-a-Service)

- Instead of writing minimal services, write just functions
- No state, rely completely on cloud storage or other cloud services
- Pay-per-invocation billing with elastic scalability
- Drawback: more ways things can fail, state is expensive
- Examples:
  AWS lambda, CloudFlare workers, Azure Functions
- What might this be good for?

- (New in 2019/20) Stateful Functions:
  Azure Durable Entities, CloudFlare Durable Objects


PLAN
CREATE
VERIFY
PACKAGE
DEV
RELEASE
CONFIGURE
MONITOR
OPS