

Architecture: Modularity & Microservices

17-313 Fall 2024

Foundations of Software Engineering

<https://cmu-313.github.io>

Michael Hilton and Rohan Padhye

Administrivia

- Project 2C: Second sprint is on!
- Teamwork assessments due every Friday
- Reminder: Midterm on October 8th in class
 - We will release sample / practice exams next week

Smoking Section

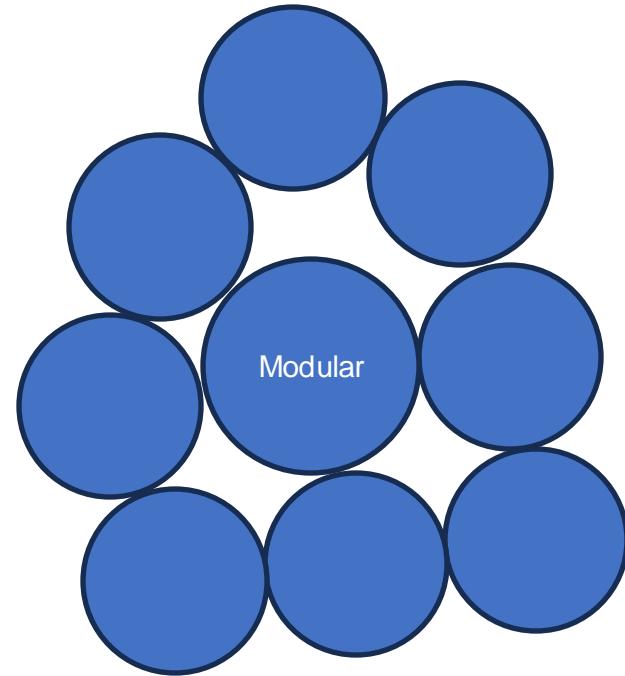
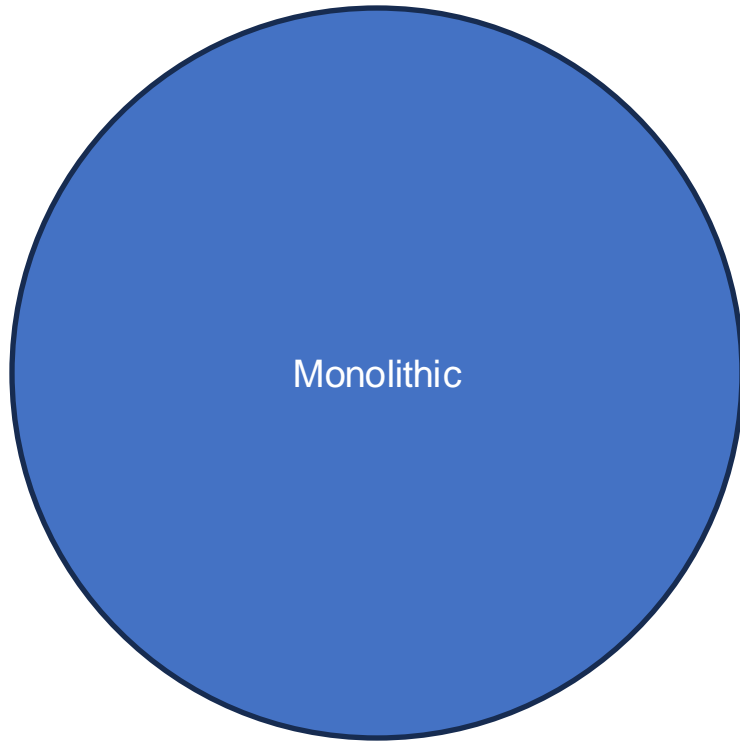
- Last **two** full rows



Learning Goals

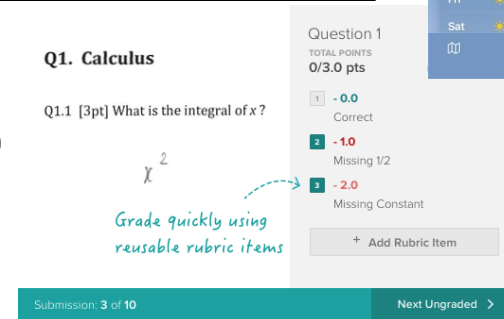
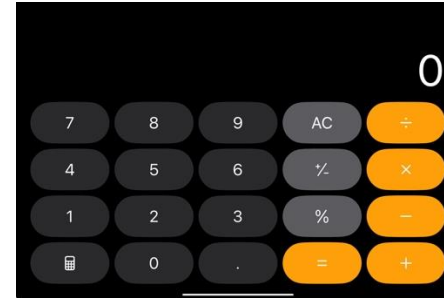
- Contrast monolithic vs. modular software architectures.
- Enumerate various types of modularity including plug-in architectures, service-oriented architectures, and microservices.
- Reason about tradeoffs of modularity: how to benefit from separation of concerns and what pitfalls to be wary of.

Monolithic vs. Modular architecture



Monoliths are the “default”

- Git (command-line interface)
- Calculator app
- PDF Reader
- Mobile weather app
- Grading web app
- Stock exchange
- Music/podcast player (e.g. Spotify)
- Video calling app (e.g. Zoom)
- Self-driving car (e.g. Apollo)



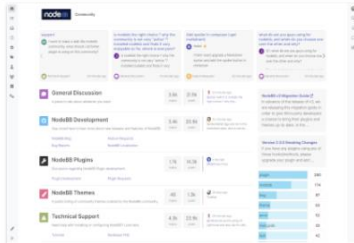
Modularity comes in many ways

- Plug-in architectures
 - Distinct code repositories, linked-in to a monolithic run-time
 - Examples:
 - Linux kernel modules
 - Themes in NodeBB, WordPress
 - Language packs for Visual Studio, IntelliJ, Sublime Text
 - Separates development, but runs as “one”.
- Service-oriented architectures
 - Distinct processes communicating via messages (e.g., Web browsers)
 - Separates run-time resource management and failure / security issues.
- Distributed micro-services
 - Independent, autonomous services communicating via web APIs
 - Separates almost all concerns

NodeBB Themes

Themes

Revert Theme

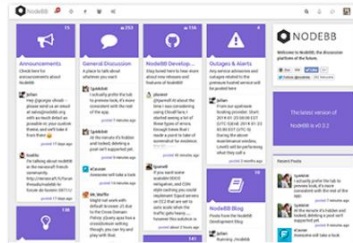


Harmony

The default theme for NodeBB. Uses a standard approach to forum design.

[Homepage](#)

Current Theme



Lavender

A theme that utilizes a masonry (magazine-like) homepage.

[Homepage](#)

Select Theme



Peace

A clean theme, optimized for use with Bootswatch skins.

[Homepage](#)

Select Theme

CATEGORIES		
Announcements	41 TOPICS	622 POSTS
General Discussion	1.5k TOPICS	11.1k POSTS
NodeBB Development	1.7k TOPICS	11.6k POSTS

NodeBB Themes

- **Activity:** Write down 2 pros and 2 cons of the NodeBB theme architecture
- Work in groups of 3-4
- Write down names and Andrew IDs!

Before going into “micro-services”, let’s discuss...

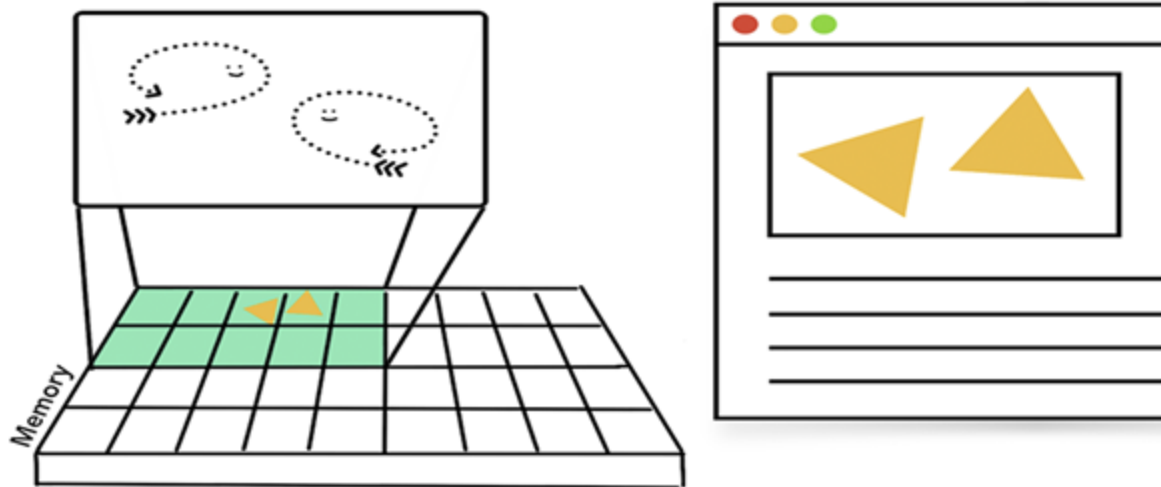
Service-oriented architecture

Case Study: Web Browsers



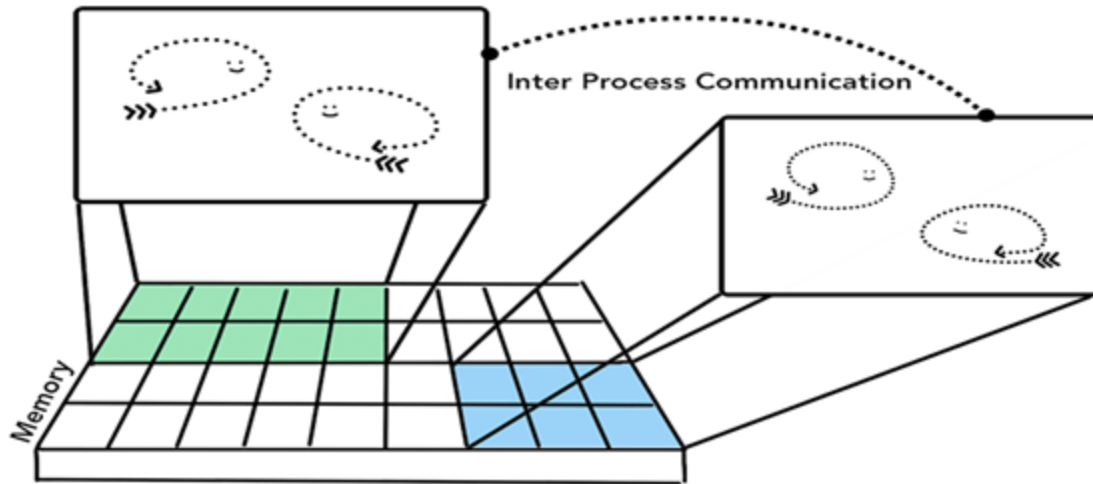
Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Multi-threaded browser in single process



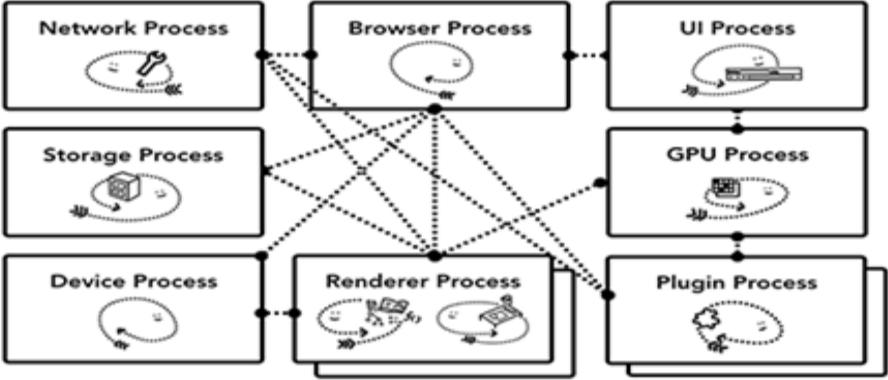
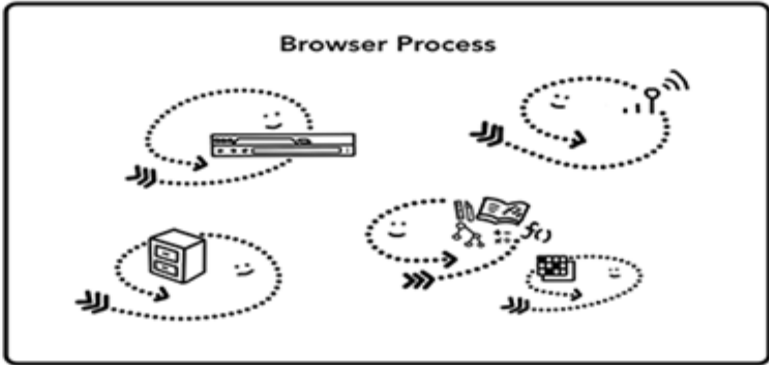
Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Multi-process browser with IPC



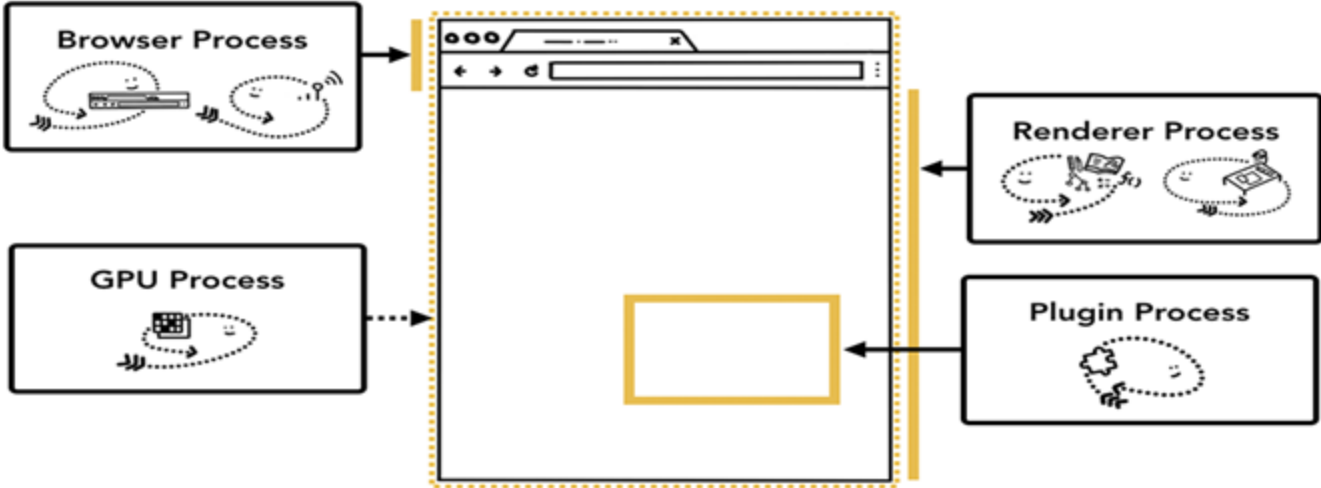
Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Service-based browser architecture



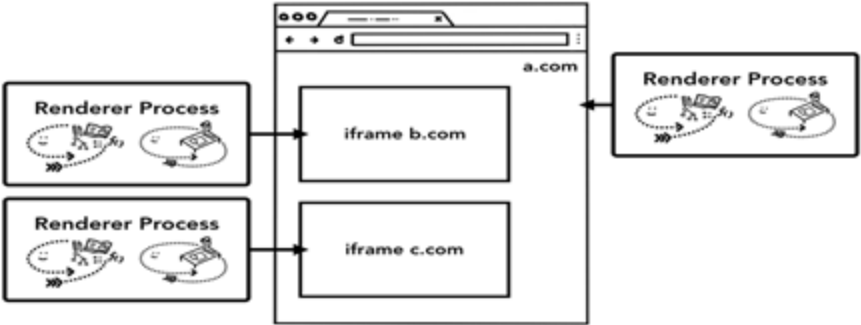
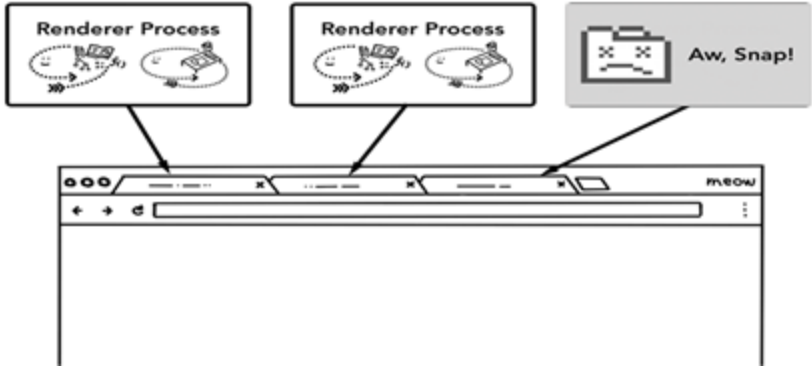
Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Service-based browser architecture

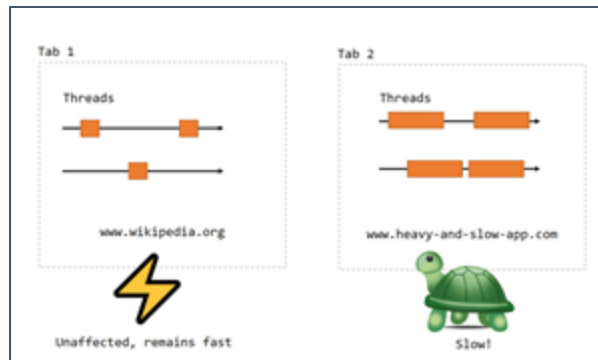
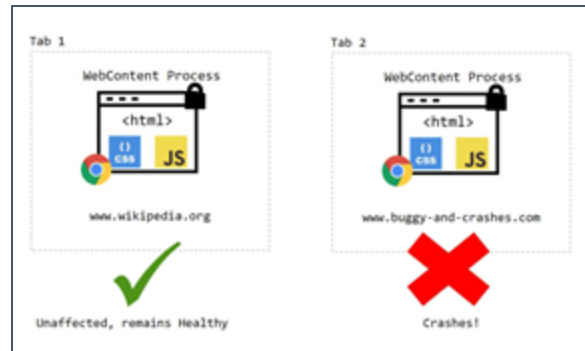
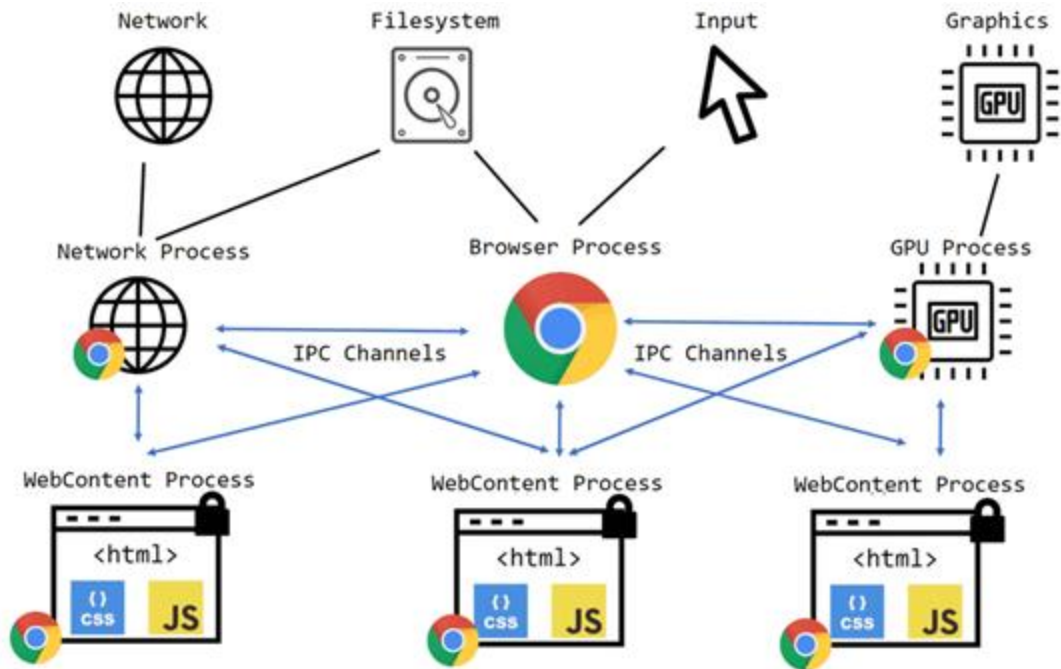


Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Service-based browser architecture



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



<https://webperf.tips/tip/browser-process-model/>

MICROSERVICES

“Small autonomous services that work well together”

Sam Newman

Microservices



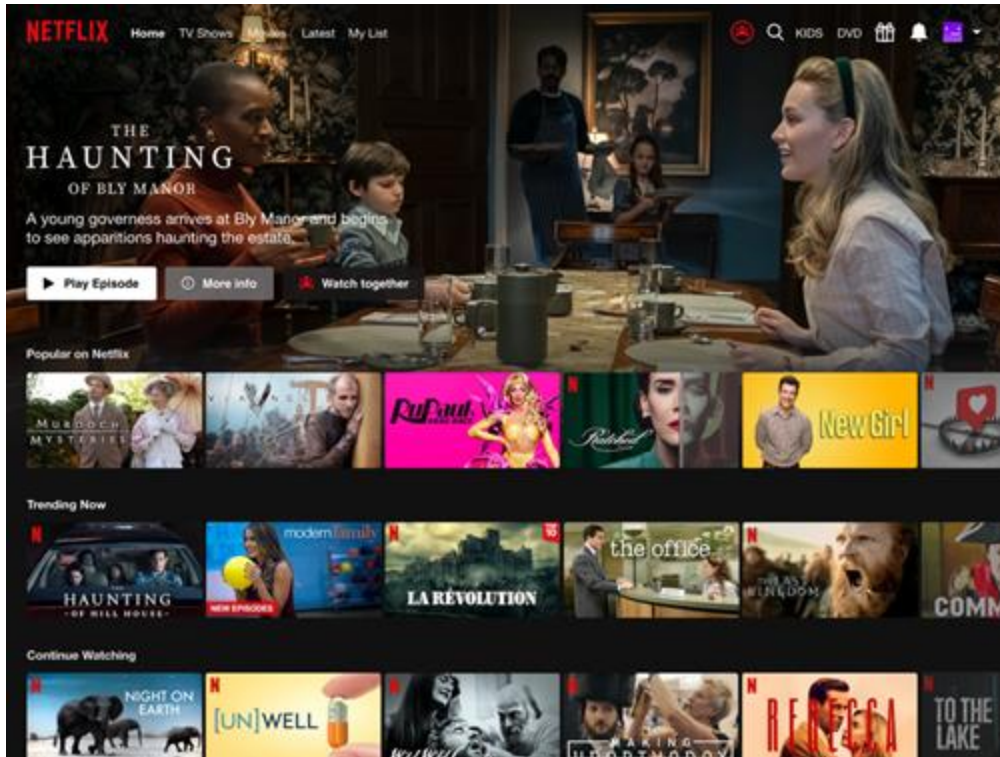
COMCAST



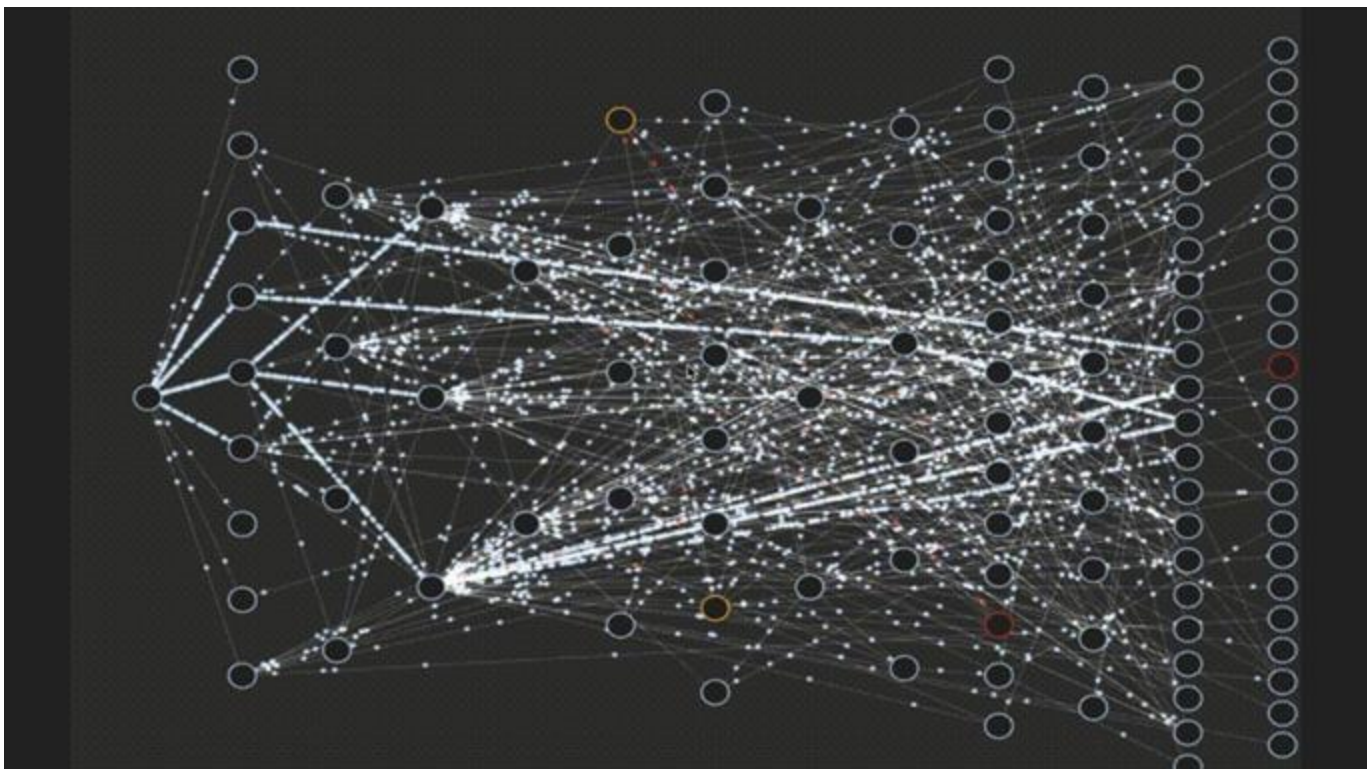
UBER

GROUPON®

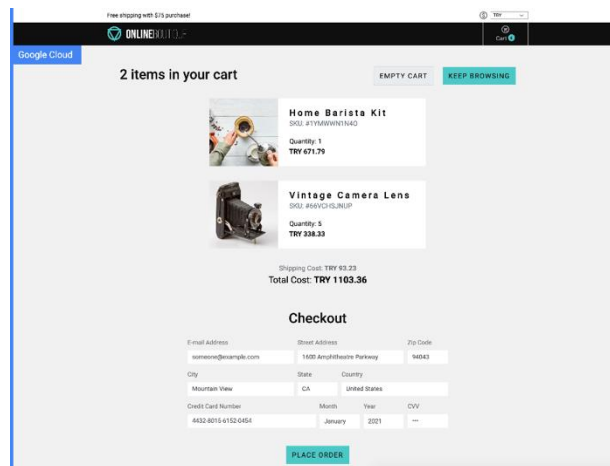
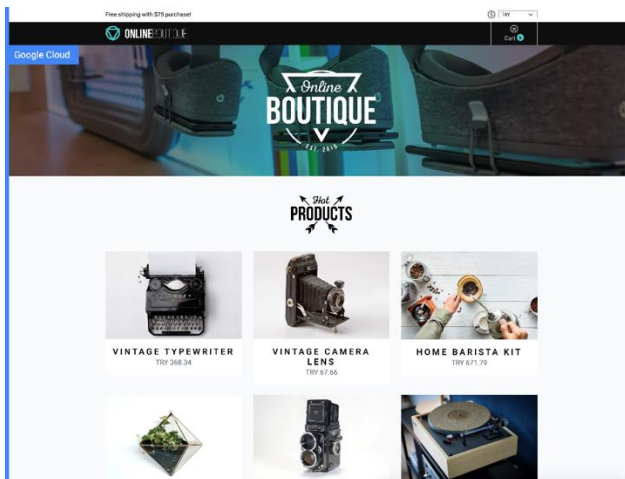
Netflix Microservices



- User subscriptions
- Banner Ad
- Popular Shows
- Trending Now
- Continue Watching
- My List (saved shows)
- Notifications
- Show info
- Trailers metadata
- Episodes metadata
- Video content

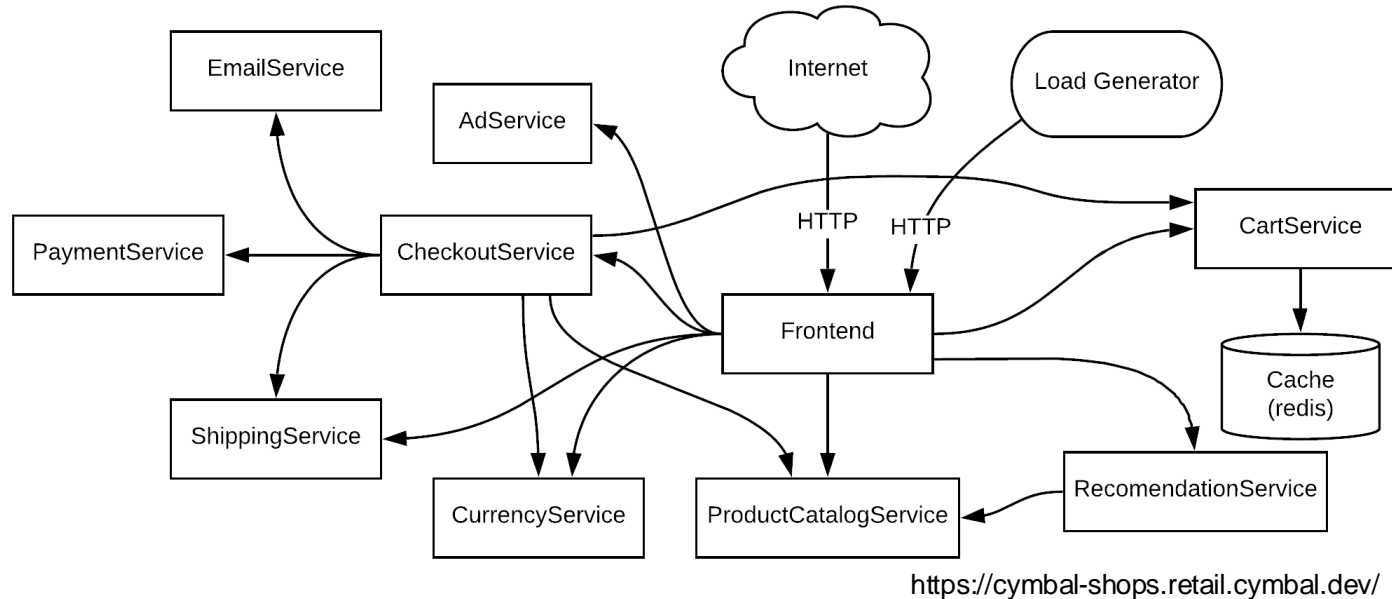


Online Boutique: Guess some microservices



<https://cymbal-shops.retail.cymbal.dev/>

Online Boutique: Microservice Architecture



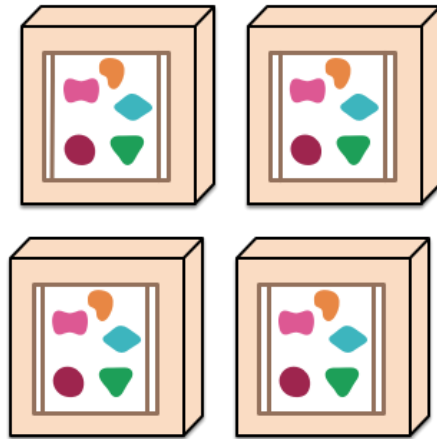
Service	Description
frontend	Exposes an HTTP server to serve the website. Does not require signup/login and generates session IDs for all users automatically.
cartservice	Stores the items in the user's shopping cart in Redis and retrieves it.
productcatalogservice	Provides the list of products from a JSON file and ability to search products and get individual products.
currencyservice	Converts one money amount to another currency. Uses real values fetched from European Central Bank.
paymentservice	Charges the given credit card info (mock) with the given amount and returns a transaction ID.
shippingservice	Gives shipping cost estimates based on the shopping cart. Ships items to the given address (mock)
checkoutservice	Retrieves user cart, prepares order and orchestrates the payment, shipping and the email notification.

Scalability

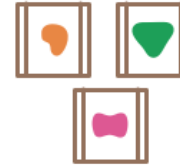
A monolithic application puts all its functionality into a single process...



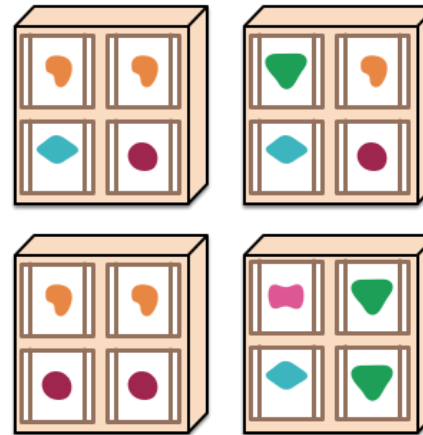
... and scales by replicating the monolith on multiple servers



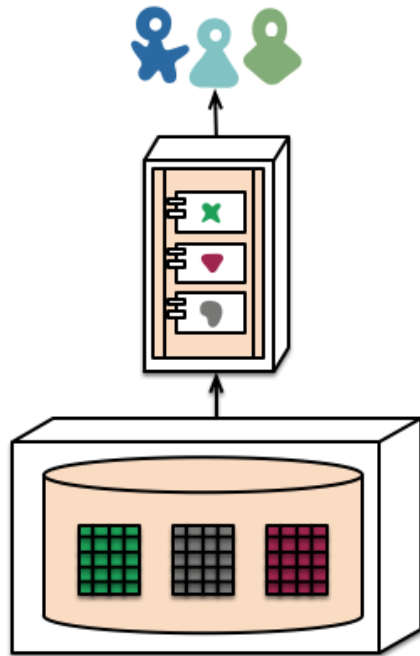
A microservices architecture puts each element of functionality into a separate service...



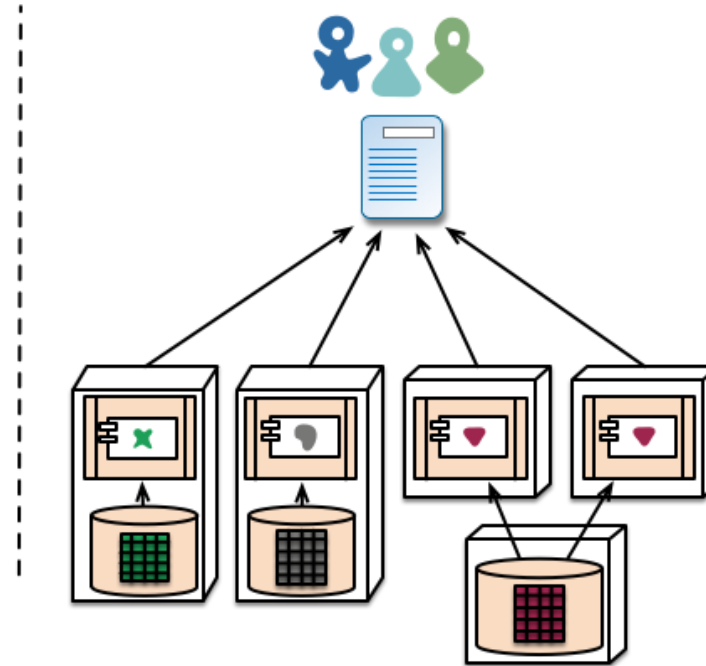
... and scales by distributing these services across servers, replicating as needed.



Data Management and Consistency



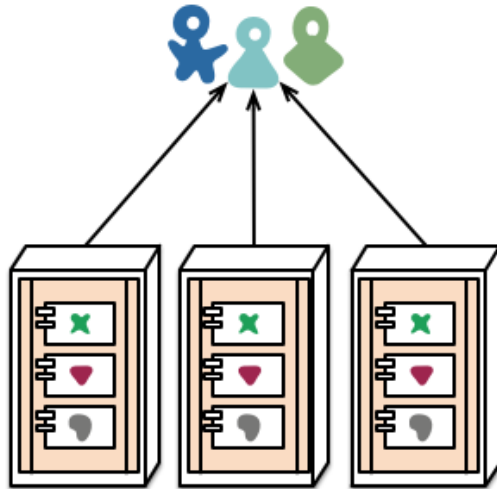
monolith - single database



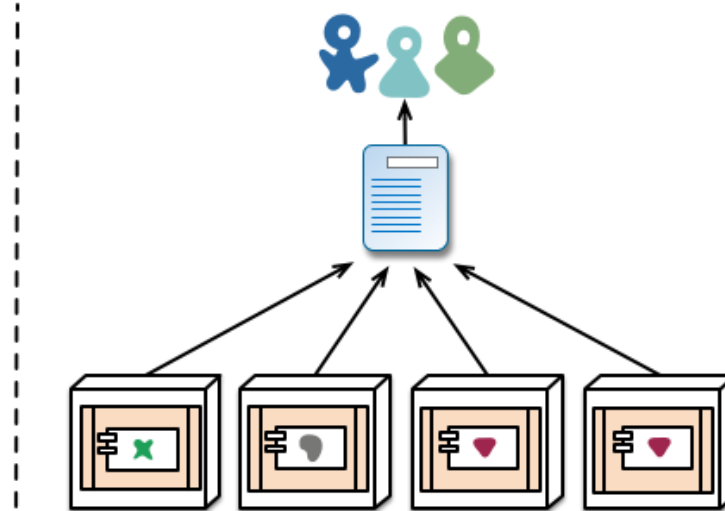
microservices - application databases

Source: <http://martinfowler.com/artides/microservices.html>

Deployment and Evolution



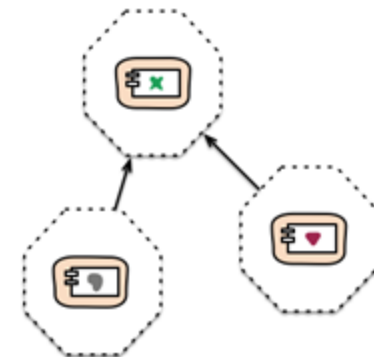
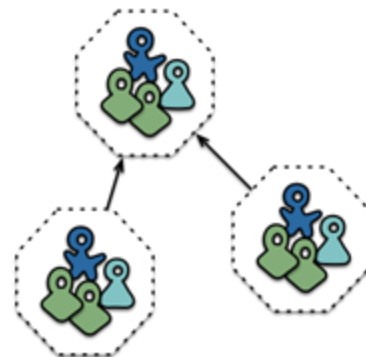
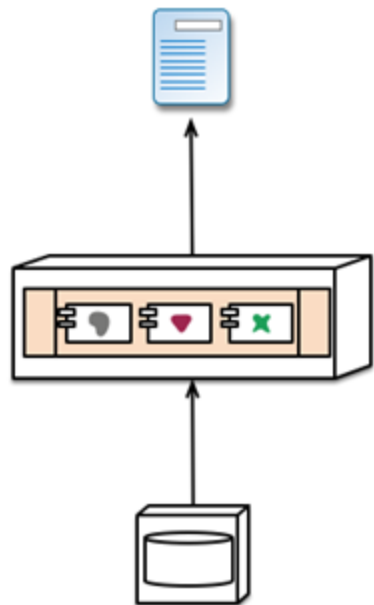
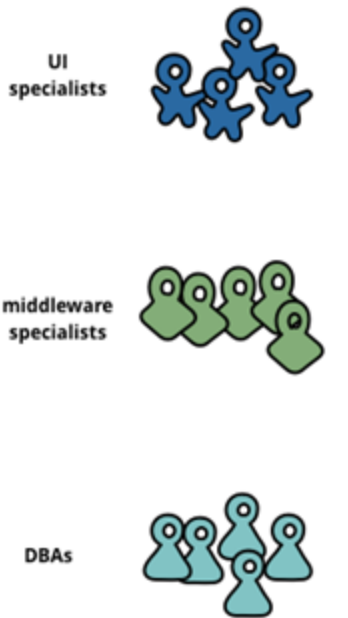
monolith - multiple modules in the same process



microservices - modules running in different processes

Conway's Law

"Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure."



"Products" not "Projects"

Advantages of Microservices

- Better alignment with the organization
- Ship features faster and safer
- Scalability
- Target security concerns
- Allow the interplay of different systems and languages, no commitment to a single technology stack
- Easily deployable and replicable
- Embrace uncertainty, automation, and faults

Microservice challenges

- Too many choices
- Delay between investment and payback
- Complexities of distributed systems
 - network latency, faults, inconsistencies
 - testing challenges
- Monitoring is more complex
- More system states
- More points of failure
- Operational complexity
- Frequently adopted by breaking down a monolithic application

Microservices overhead

for less-complex systems, the extra baggage required to manage microservices reduces productivity

as complexity kicks in, productivity starts falling rapidly

the decreased coupling of microservices reduces the attenuation of productivity

