

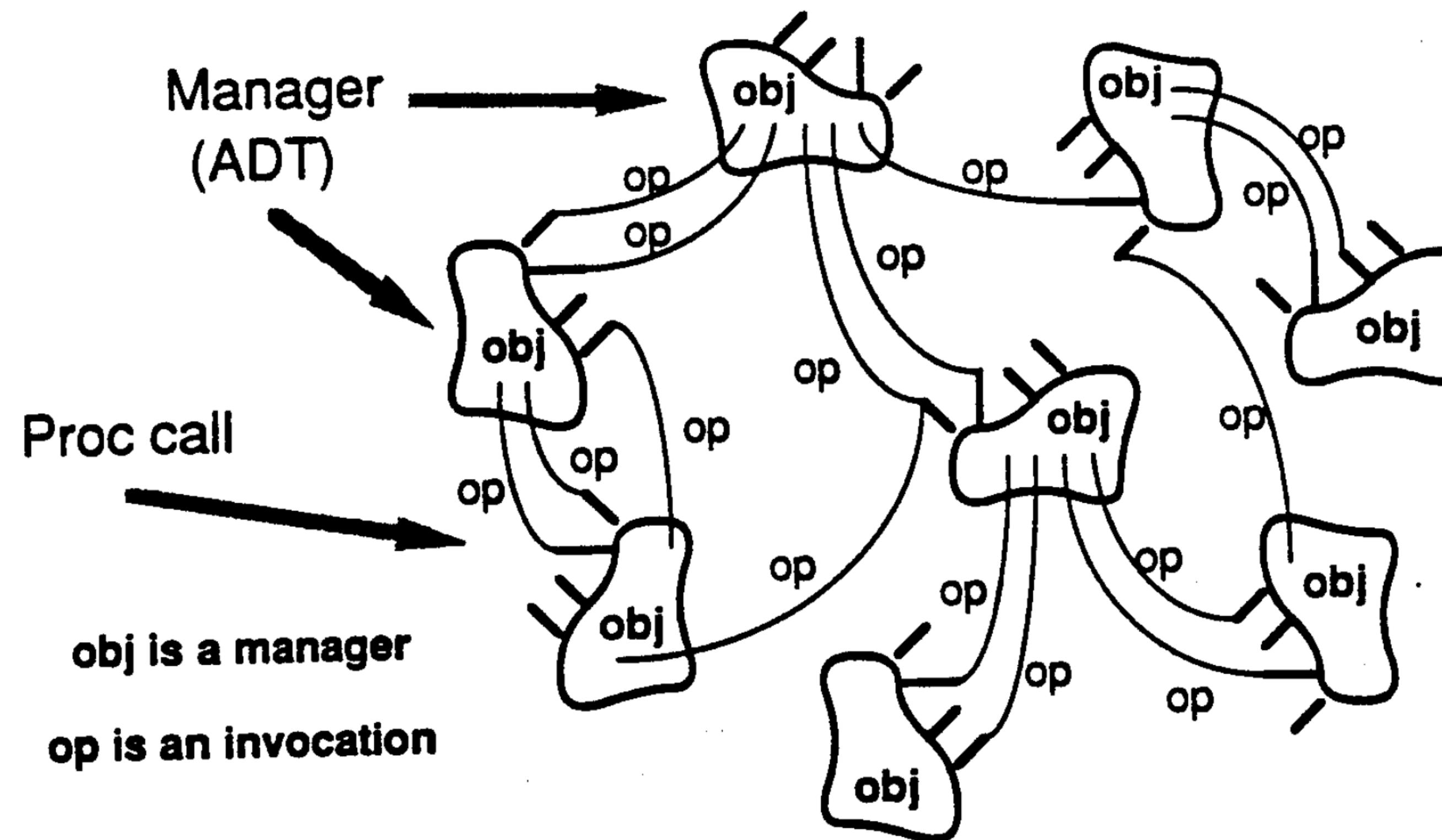
Introduction to Software Architecture, Part 2

Michael Coblenz



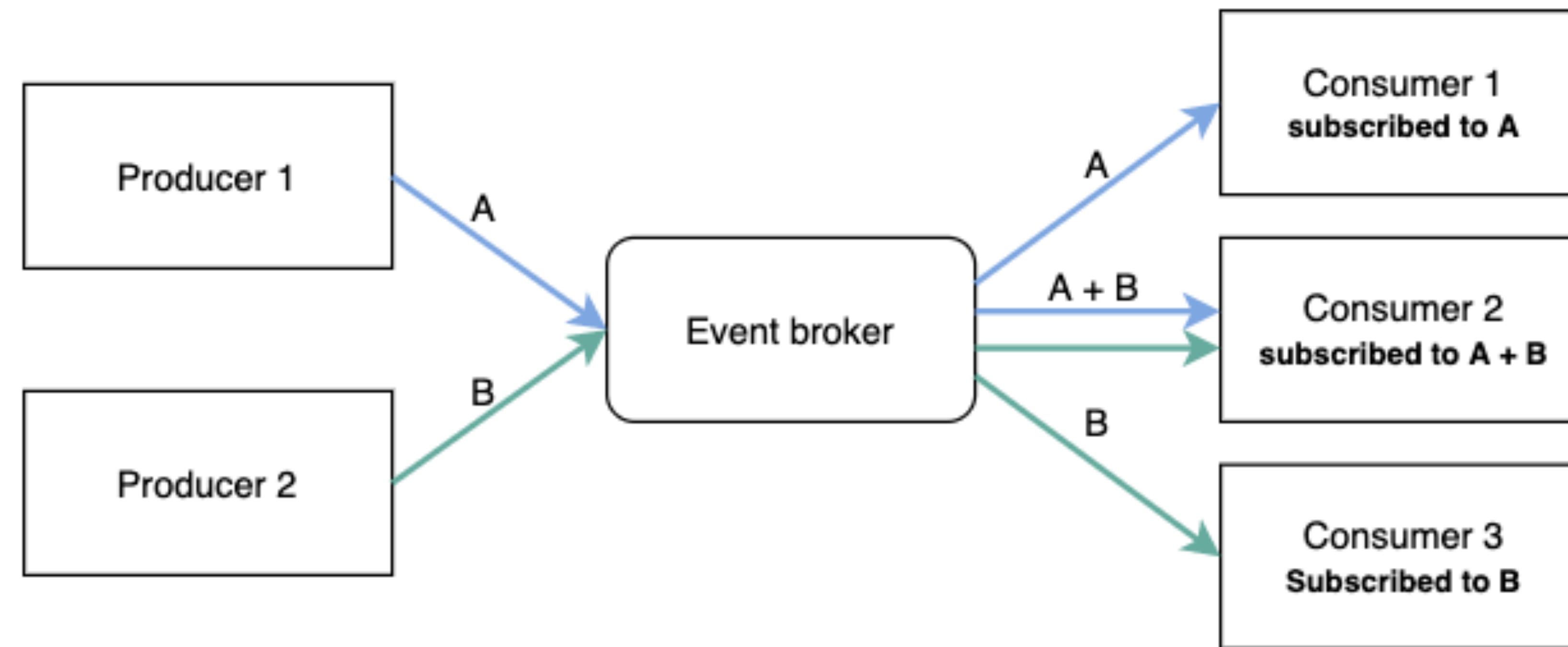
Slide credit: Michael Hilton at CMU

2. Object-Oriented Organization

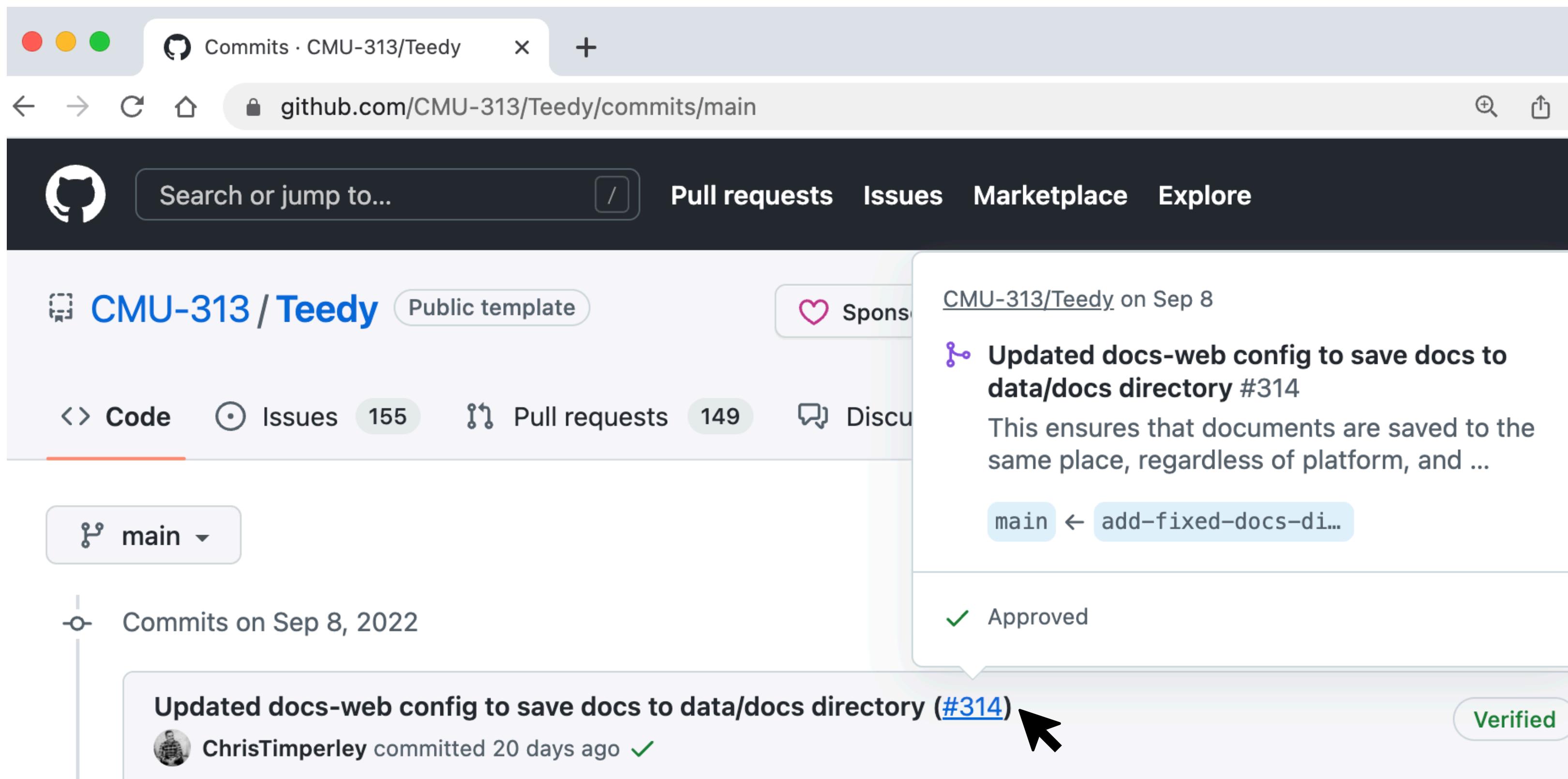


© David Garlan and Mary Shaw, CMU/SEI-94-TR-021

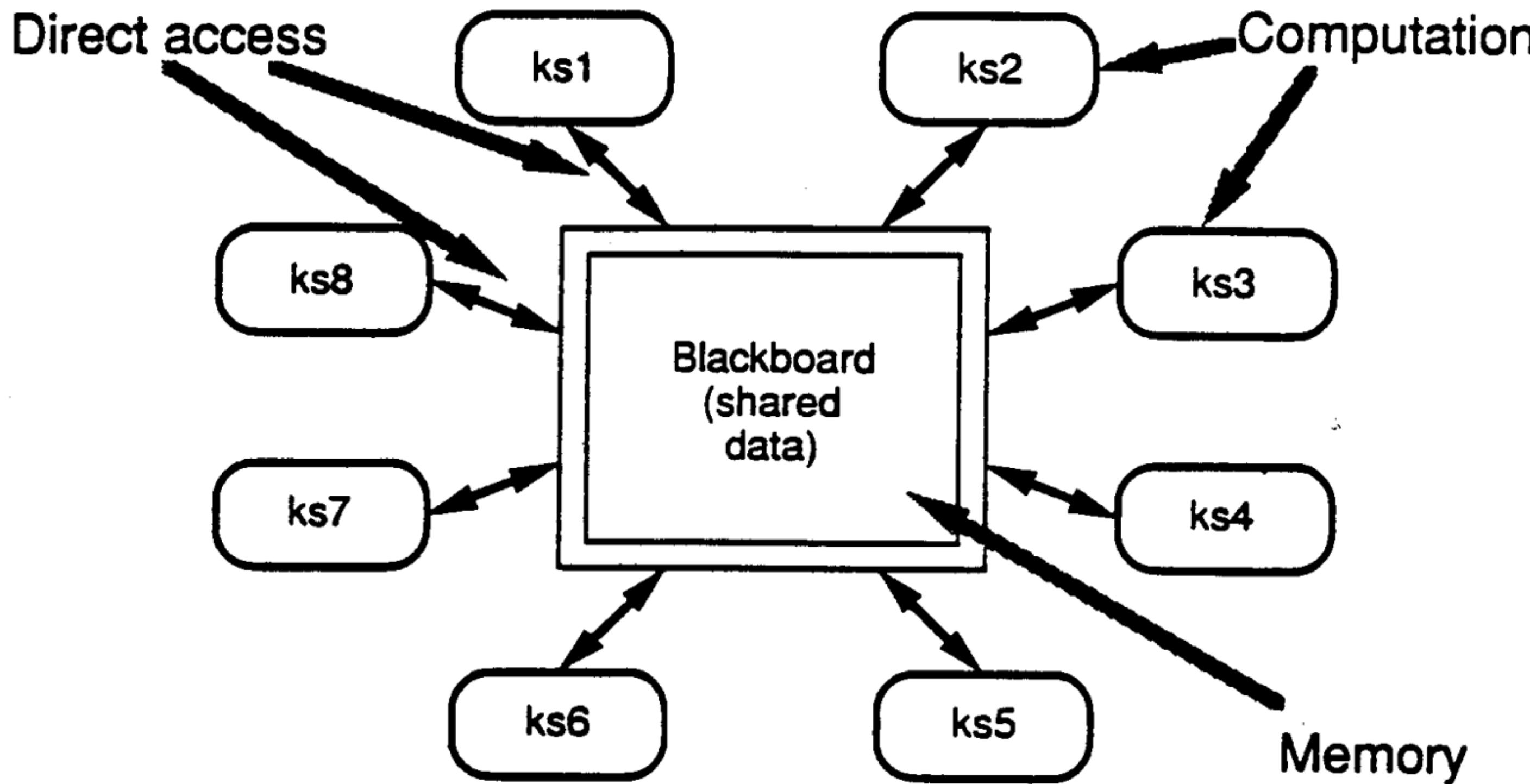
3. Event-Driven Architecture



Example: HTML DOM + JavaScript

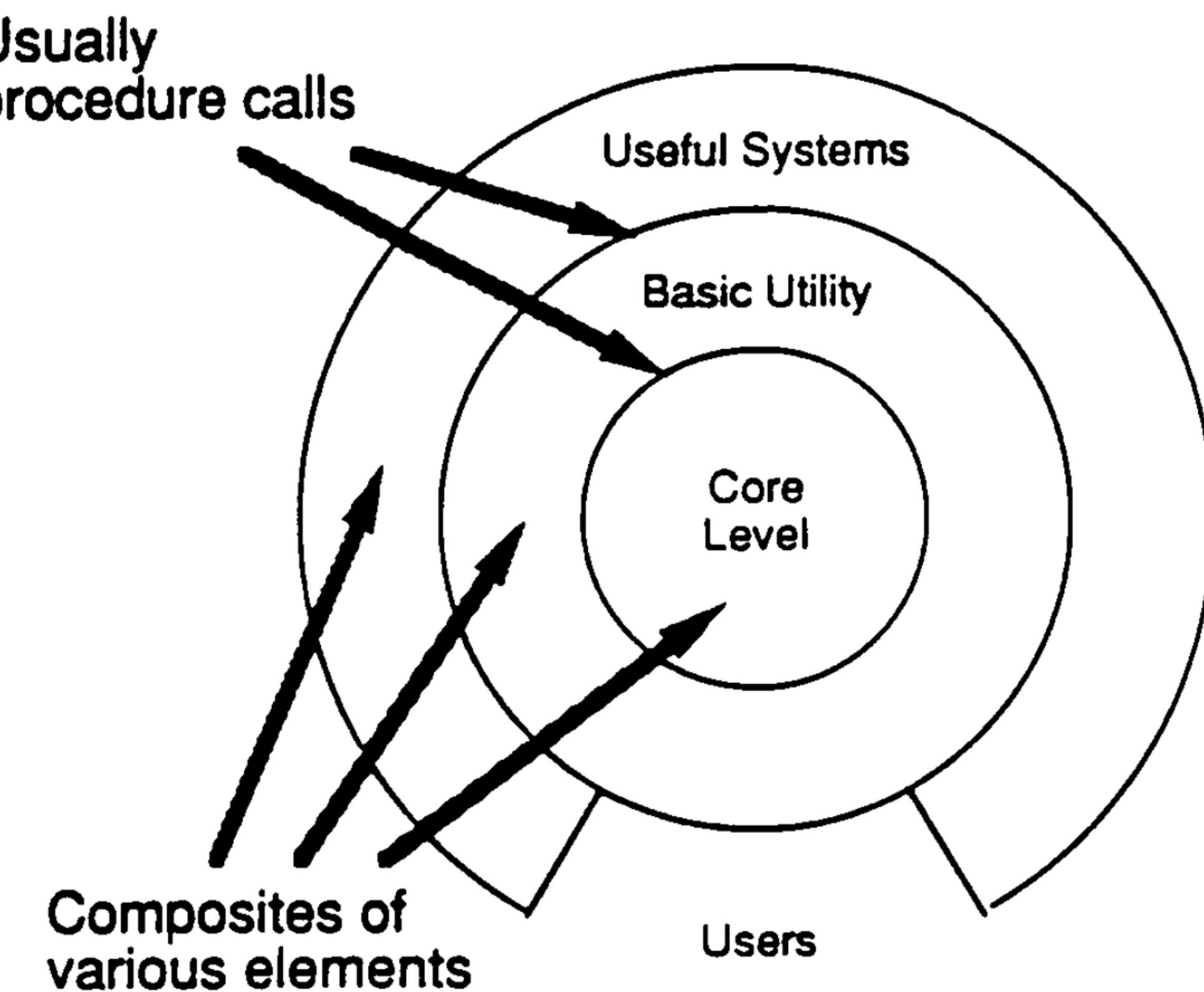


4. Blackboard Architecture



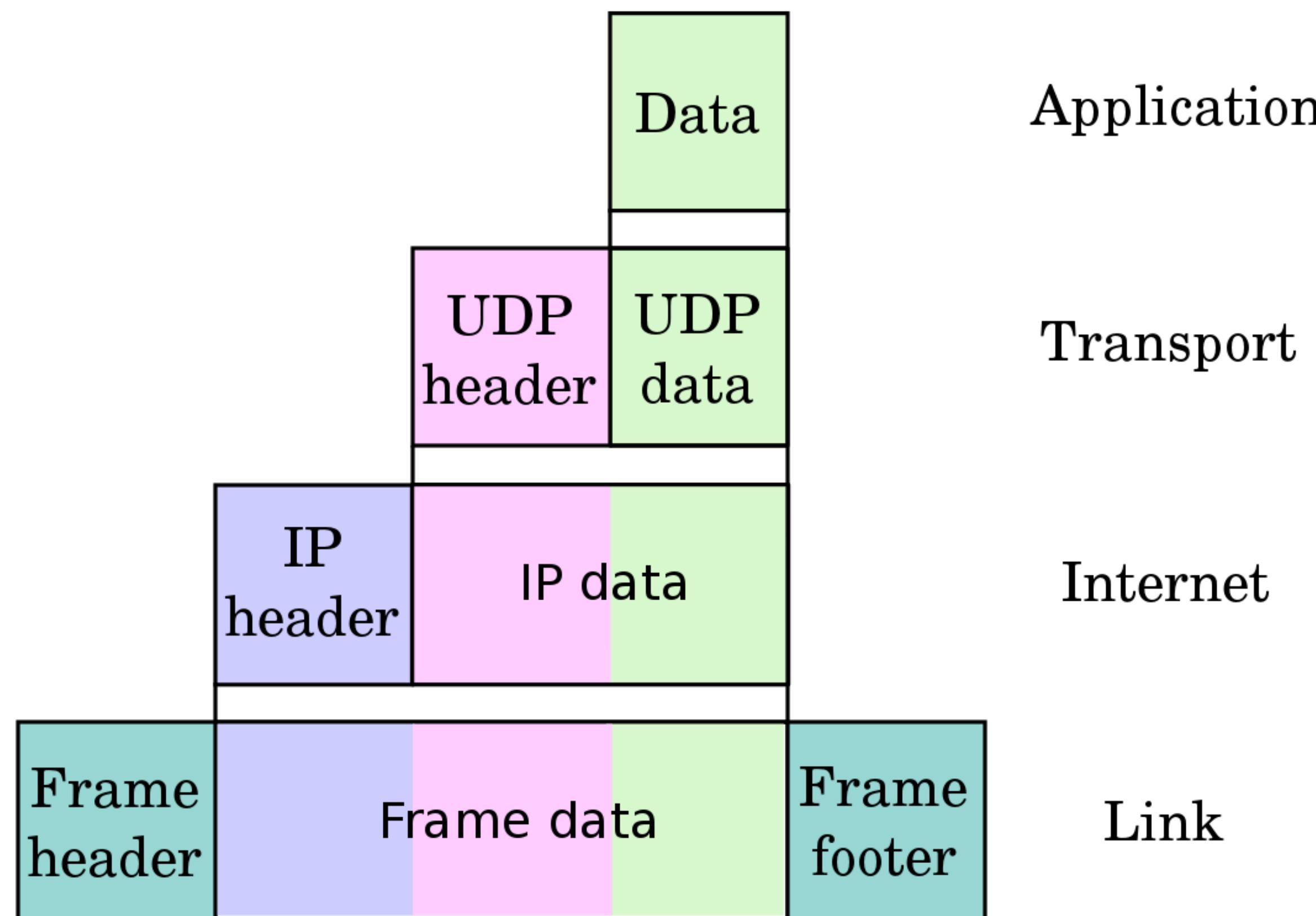
© David Garlan and Mary Shaw, CMU/SEI-94-TR-021

5. Layered Systems



© David Garlan and Mary Shaw, CMU/SEI-94-TR-021

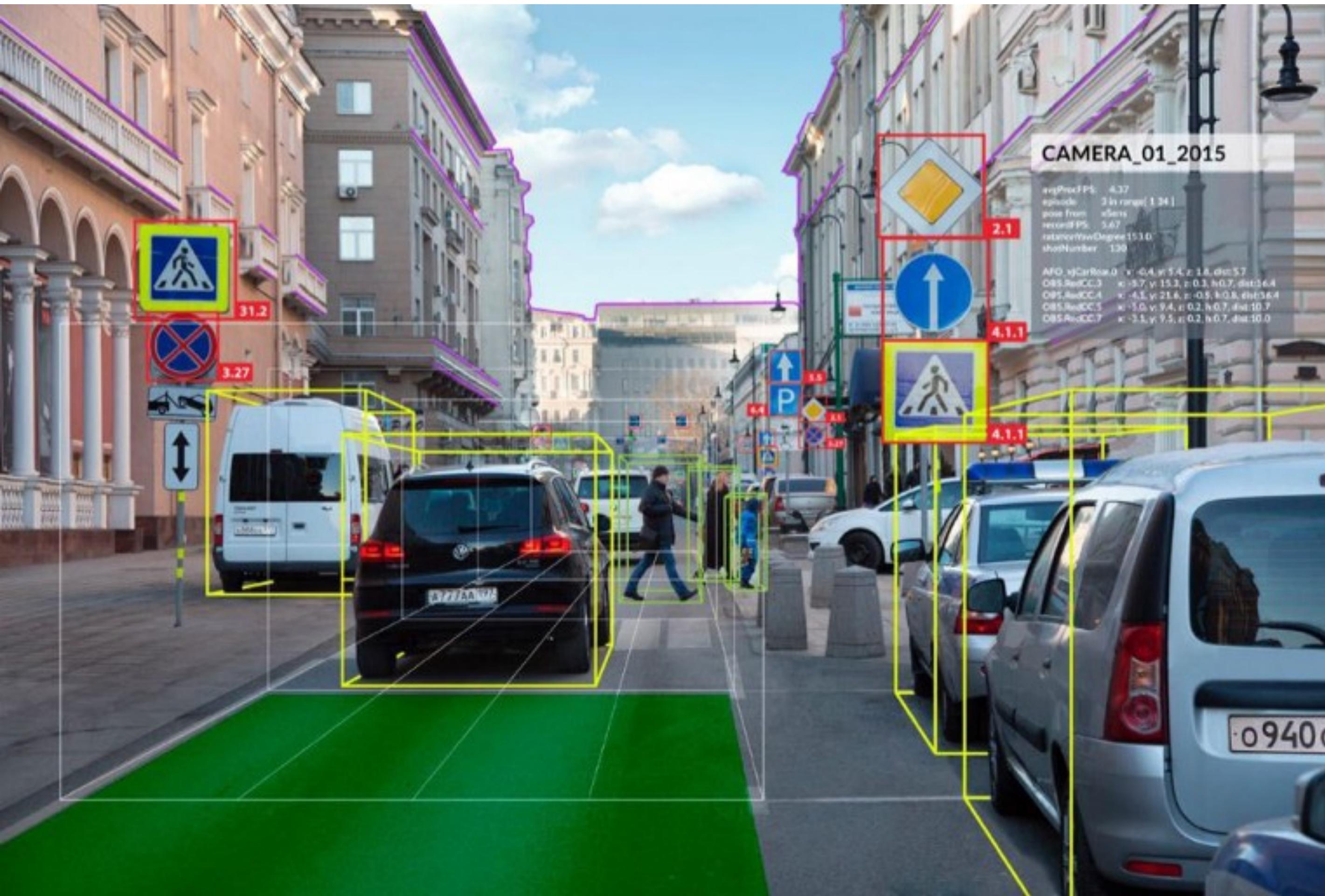
Example: Internet Protocol Suite



Guidelines for selecting a notation

- Suitable for purpose
- Often visual for compact representation
- Usually boxes and arrows
- UML possible (semi-formal), but possibly constraining
 - Note the different abstraction level – Subsystems or processes, not classes or objects
- Formal notations available
- Decompose diagrams hierarchically and in views
- Always include a legend
- Define precisely what the boxes mean
- Define precisely what the lines mean
- Do not try to do too much in one diagram
 - Each view of architecture should fit on a page
 - Use hierarchy

Case Study: Autonomous Vehicle Software



Case Study: Apollo

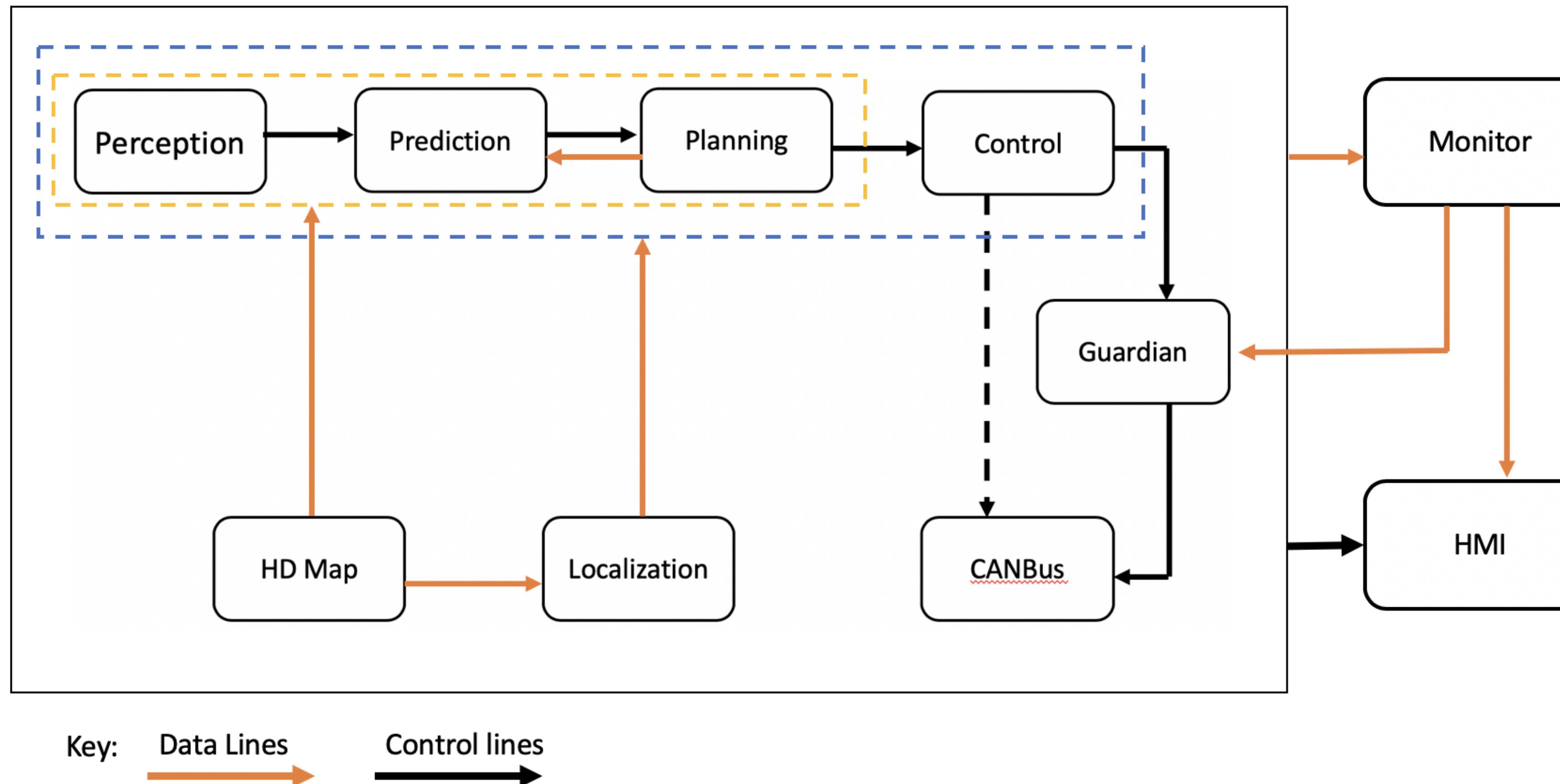
Check out the “side pass” feature from the video:

<https://www.youtube.com/watch?v=BXNDUtNZdM4>

Source: <https://github.com/ApolloAuto/apollo>

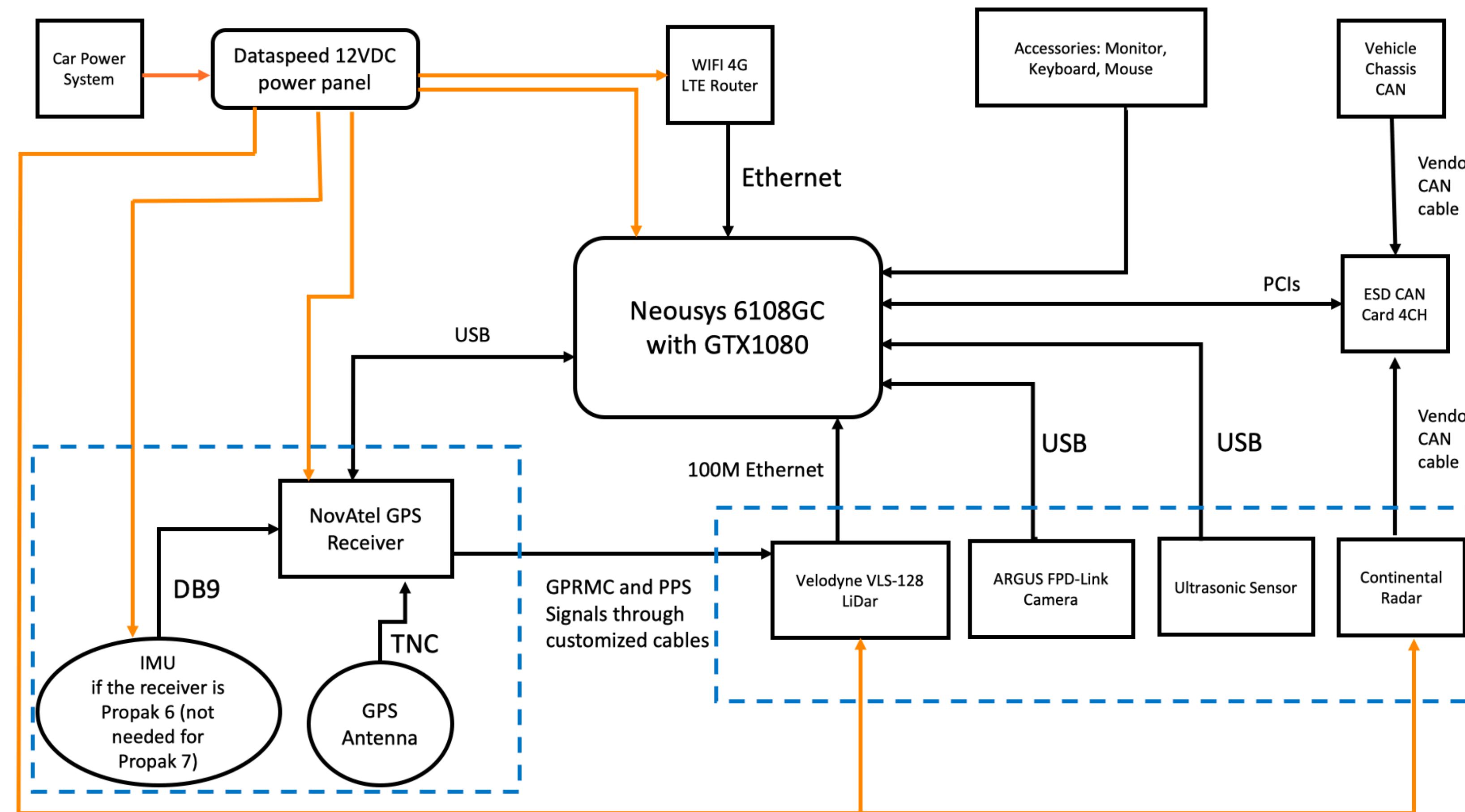
Doxygen: <https://hidetoshi-furukawa.github.io/apollo-doxygen/index.html>

Apollo Software Architecture



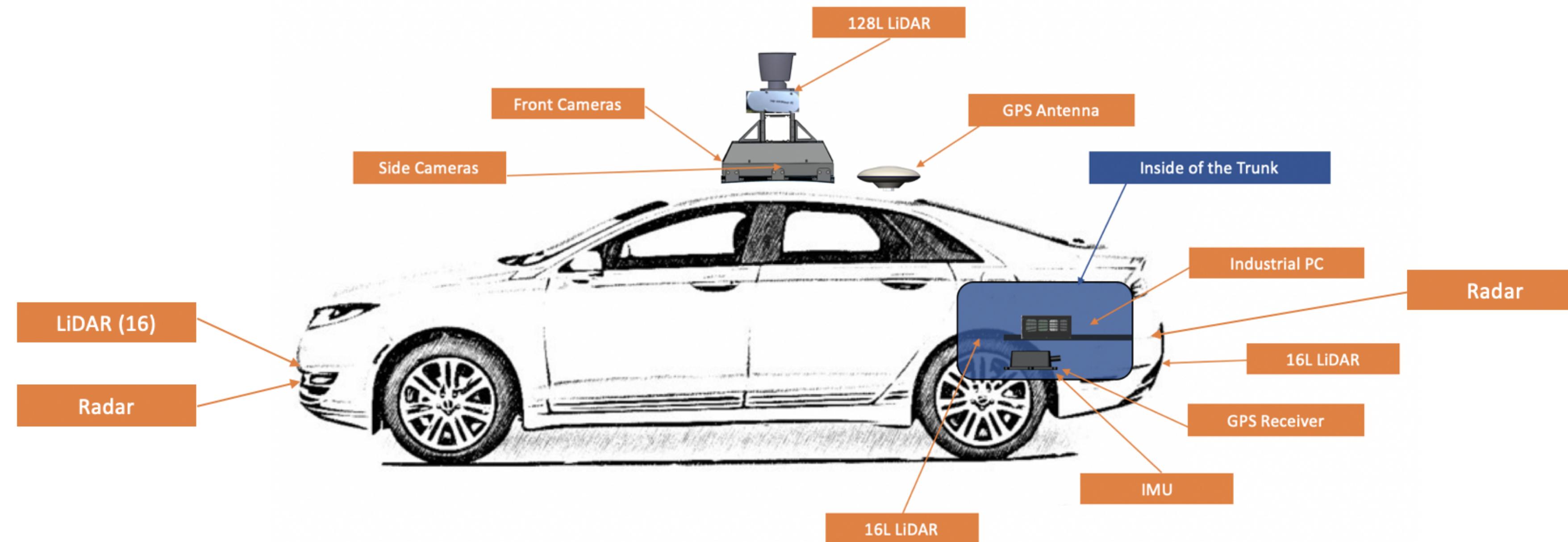
Source: https://github.com/ApolloAuto/apollo/blob/v6.0.0/docs/specs/Apollo_5.5_Software_Architecture.md

Apollo Hardware Architecture



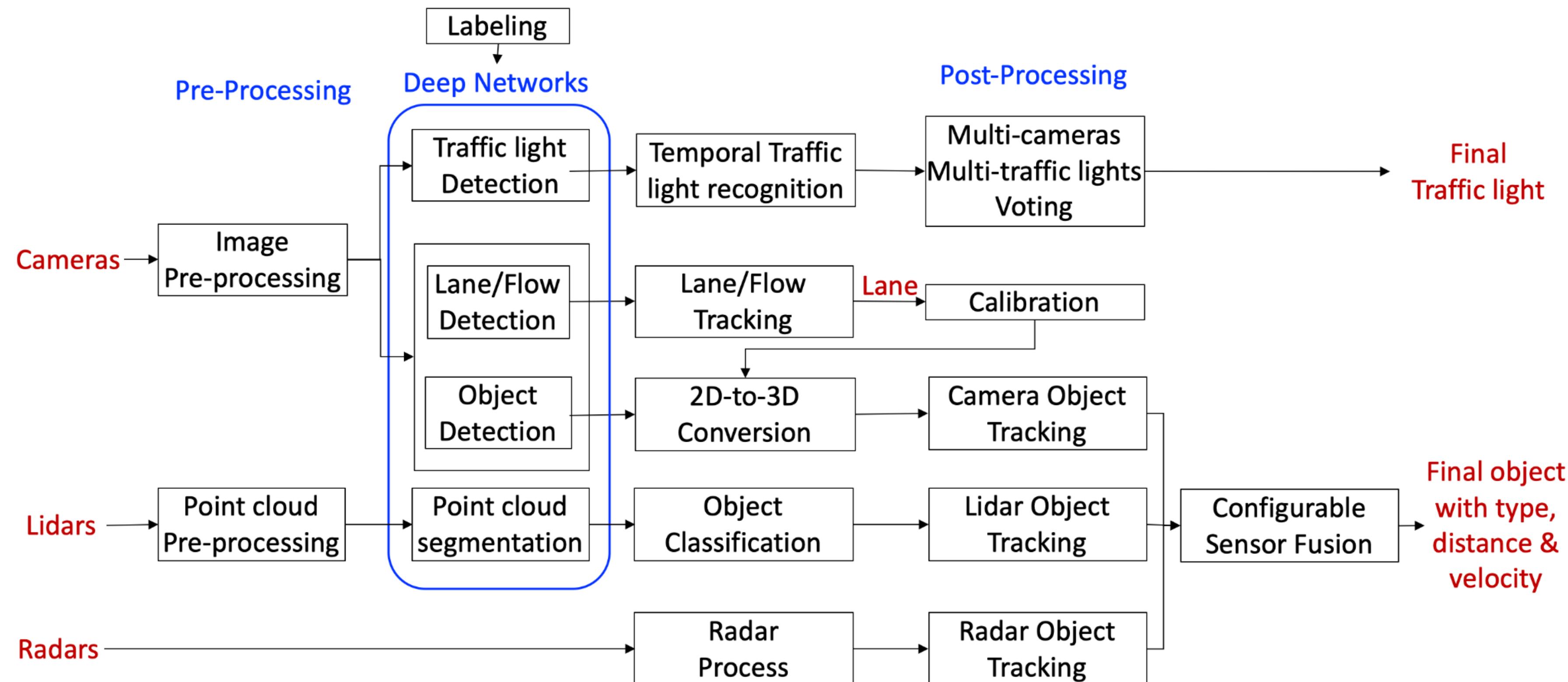
Source: <https://github.com/ApolloAuto/apollo/blob/v6.0.0/README.md>

Apollo Hardware/Vehicle Overview

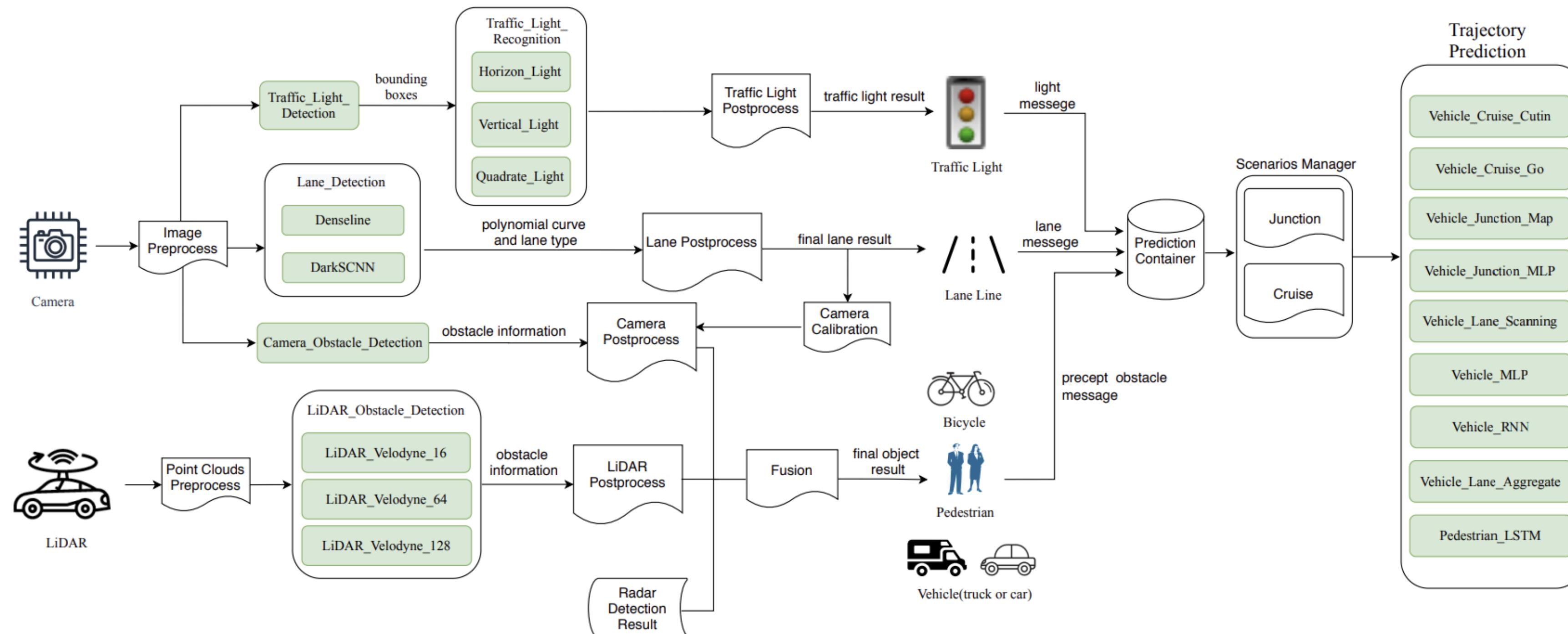


Source: <https://github.com/ApolloAuto/apollo/blob/v6.0.0/README.md>

Apollo Perception Module



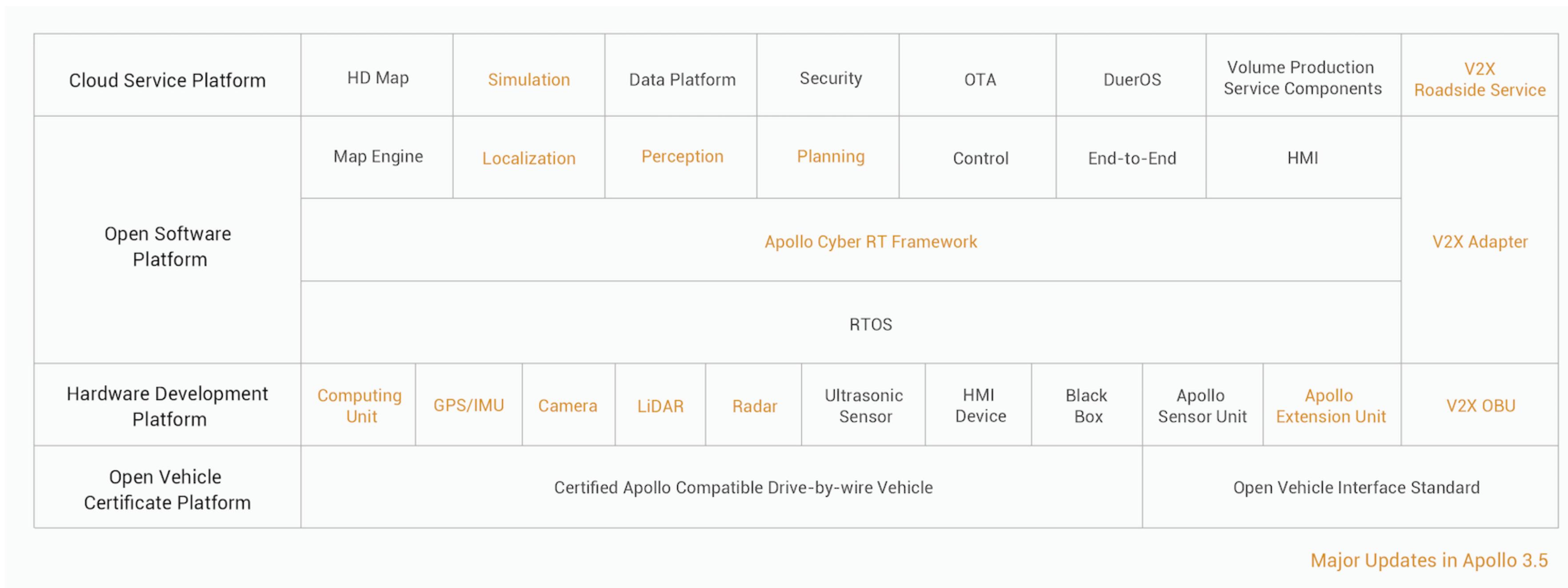
Apollo ML Models



Source: Zi Peng, Jinqiu Yang, Tse-Hsun (Peter) Chen, and Lei Ma. 2020. A First Look at the Integration of Machine Learning Models in Complex Autonomous Driving Systems: A Case Study on Apollo. In Proceedings of the 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '20), <https://doi.org/10.1145/3368089.3417063>

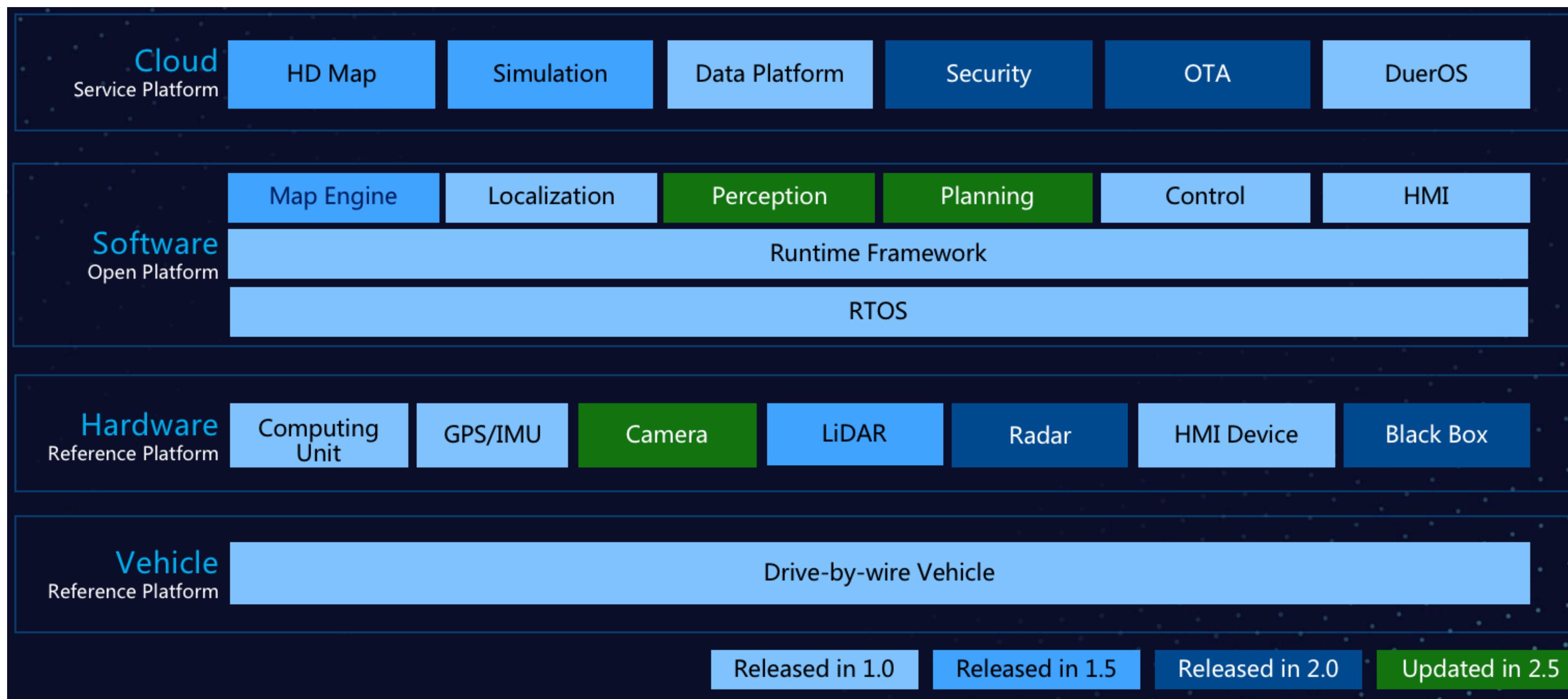
Apollo Software Stack

Convention: components can use components in layers below
(sometimes: only the layer immediately below)



Source: <https://github.com/ApolloAuto/>

Feature Evolution (Software Stack View)



Source: <https://github.com/ApolloAuto/apollo>

Monolithic Design vs. Microservices

Before we get to microservices...

How might these apps be architected?

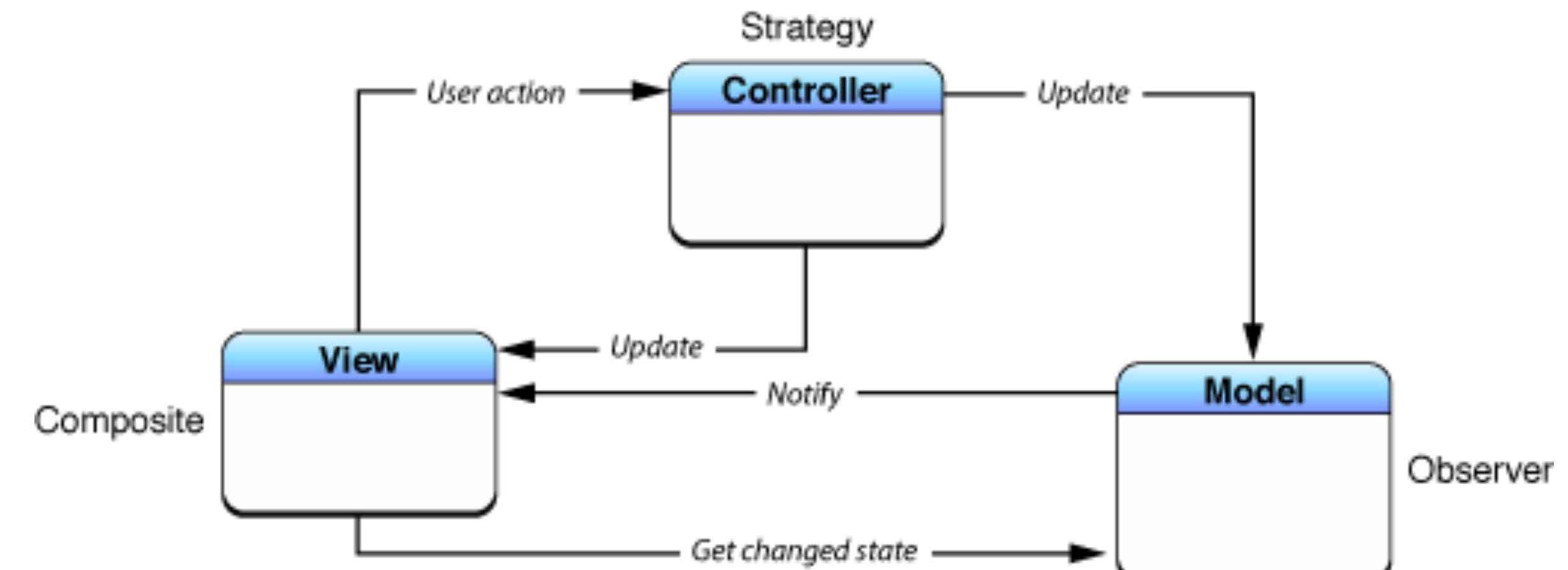
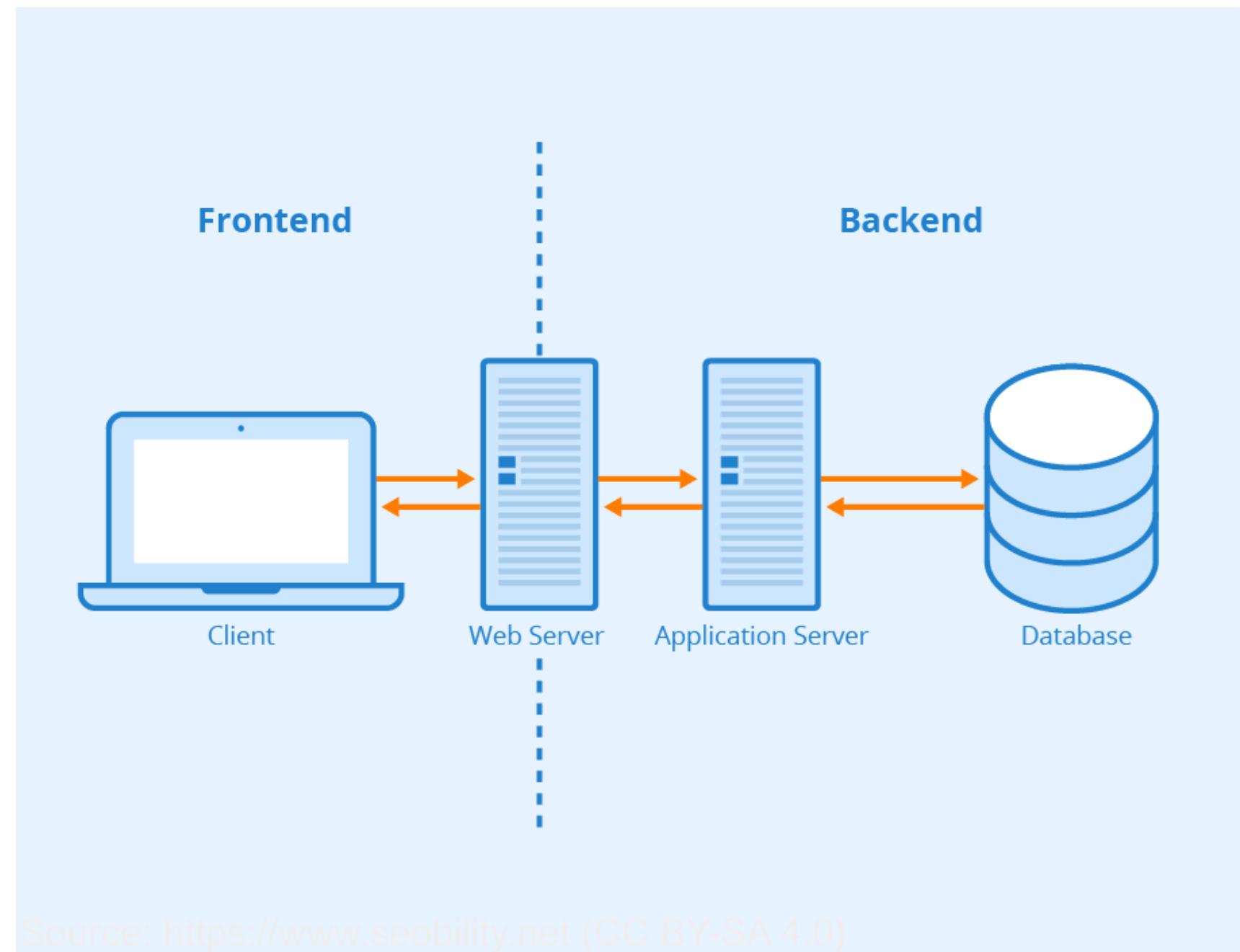
The Netflix homepage features a large banner for the TV show 'The Haunting of Bly Manor'. Below the banner, there are sections for 'Popular on Netflix' and 'Trending Now'. The 'Popular on Netflix' section includes thumbnails for 'Murdoch Mysteries', 'Vikings', 'RuPaul's Drag Race', 'Ratched', and 'New Girl'. The 'Trending Now' section includes thumbnails for 'The Haunting of Hill House', 'Modern Family', 'La Révolution', 'the office', 'The Last Kingdom', and 'COMM'.

The Simeics Docs interface shows a document titled 'Tourism' created on 11/18/17 by 'admin'. The document has a table of contents and several attachments, including '29789.pdf', '2998_no_90_p_2048w_7534.jpg', 'lyon-immobilier-neuf-ct.jpg', and 'Lyon-Riverfront.jpg'. There are also sections for 'Contributors' and 'Comments'.

The first screenshot shows a person riding a scooter with the text 'During the journey' and 'Rent a scooter when traveling around the world'. The second screenshot shows a map of San Francisco with a scooter route marked. The third screenshot shows payment options: 'Your scooter is checked and ready to ride', with options for 'VISA' (5421), 'Apple Pay' (instant payment with credit card), 'Add card' (add new payment method in app), and 'Scan QR code' (the money will be debited from your card).

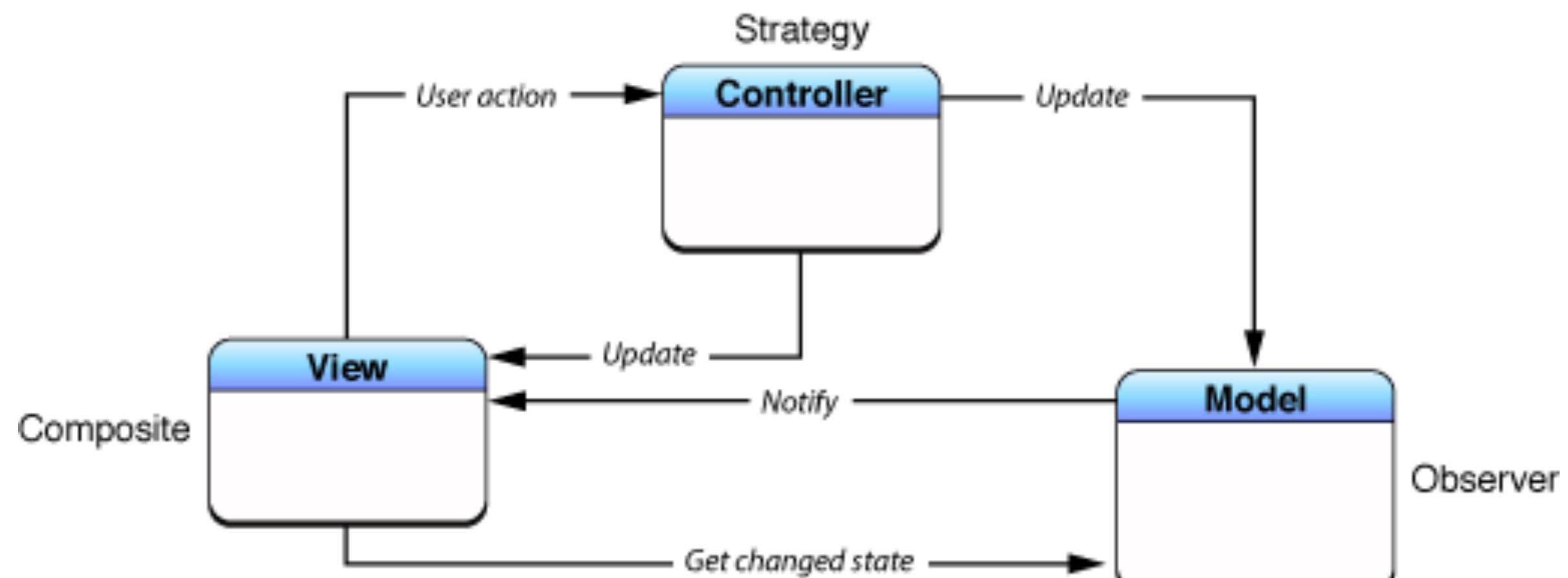
The Facebook news feed shows posts from 'Andy Chung' and 'Vivian Wang'. 'Andy Chung' posted 'What's on your mind?' and 'Colorful day on the boardwalk!'. 'Vivian Wang' posted 'Just now - Colorful day on the boardwalk!'. The feed also includes trending news items like 'Paul Rudd: Lip Sync Battle with Paul Rudd' and 'Ben & Jerry's: Ben and Jerry's Announces Four New "Core" Flavors'.

Monolithic styles: Client-server or MVC



Brief digression: MVC (Model-View-Controller)

- Views:
 - Reusable views promote consistency
 - Modularity promotes reusability
- Model: separate to allow representation independence
- Controller: "business logic"; very custom



Monoliths make trade-offs on software quality

Several consequences of this architecture on:

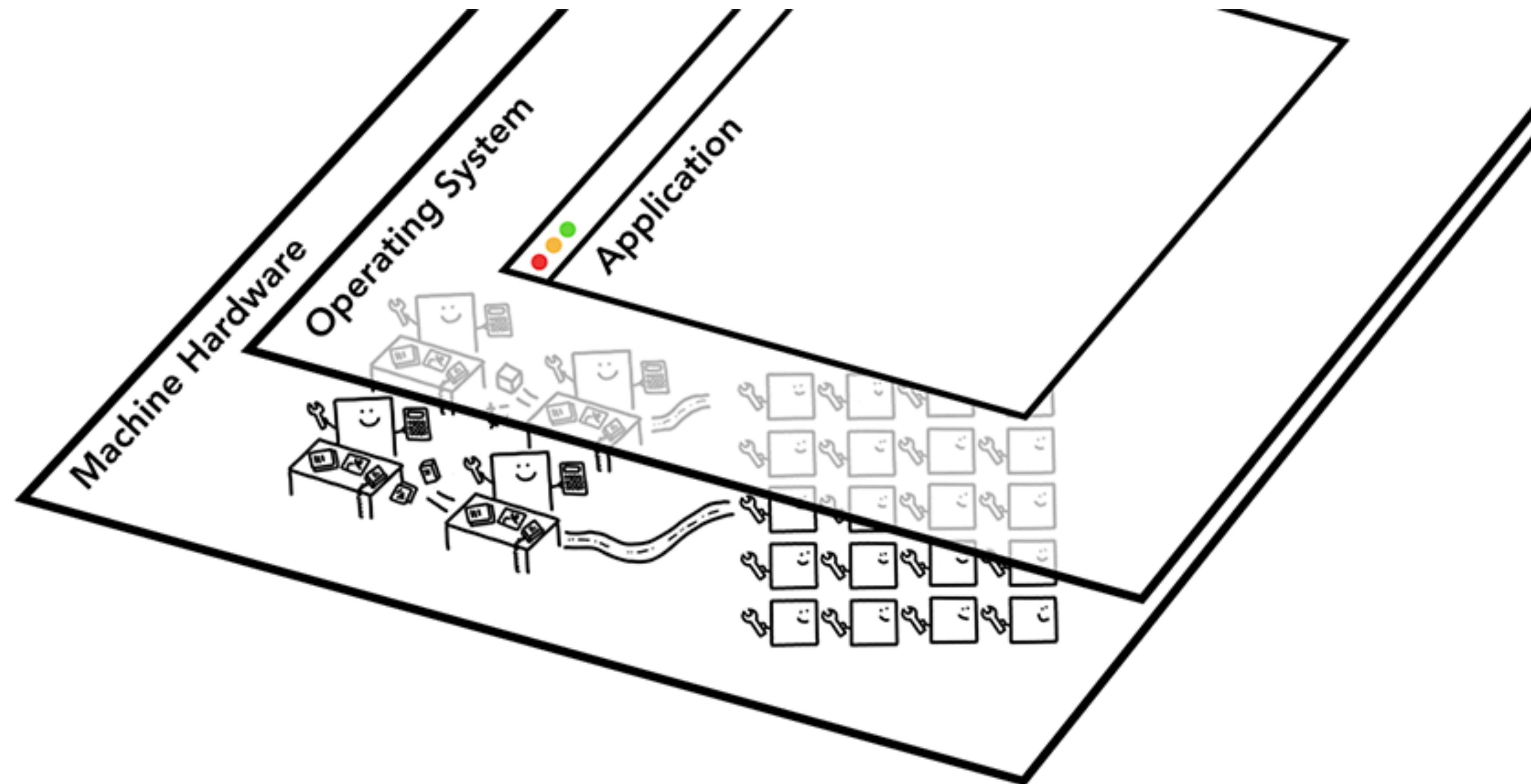
- Scalability
- Reliability
- Performance
- Development
- Maintainability
- Evolution
- Testability
- Ownership

The image contains three screenshots of monolithic software interfaces:

- Screenshot 1 (Top): Facebook News Feed.** Shows a user's news feed with posts from friends like Andy Chung, Vivian Wang, and Matt Visconti, along with trending news items about Paul Rudd, Ben & Jerry's, and Paco de Lucia.
- Screenshot 2 (Bottom Left): Sismics Docs search results.** Shows a search interface for documents, listing results for terms like "Authoritarianism", "Auto-segregation", "Discrimination based on skin color", "Collectivism", "Indigo Era (economical)", "Dhat syndrome", "Korean ethnic nationalism", "Individualism", "Anarchism", and "Music".
- Screenshot 3 (Bottom Right): Sismics Docs document view.** Shows a detailed view of a document titled "Tourism" created by "admin" on 11/18/17. The view includes tabs for Content, Workflow, Permissions, and Activity, and displays four images related to tourism.

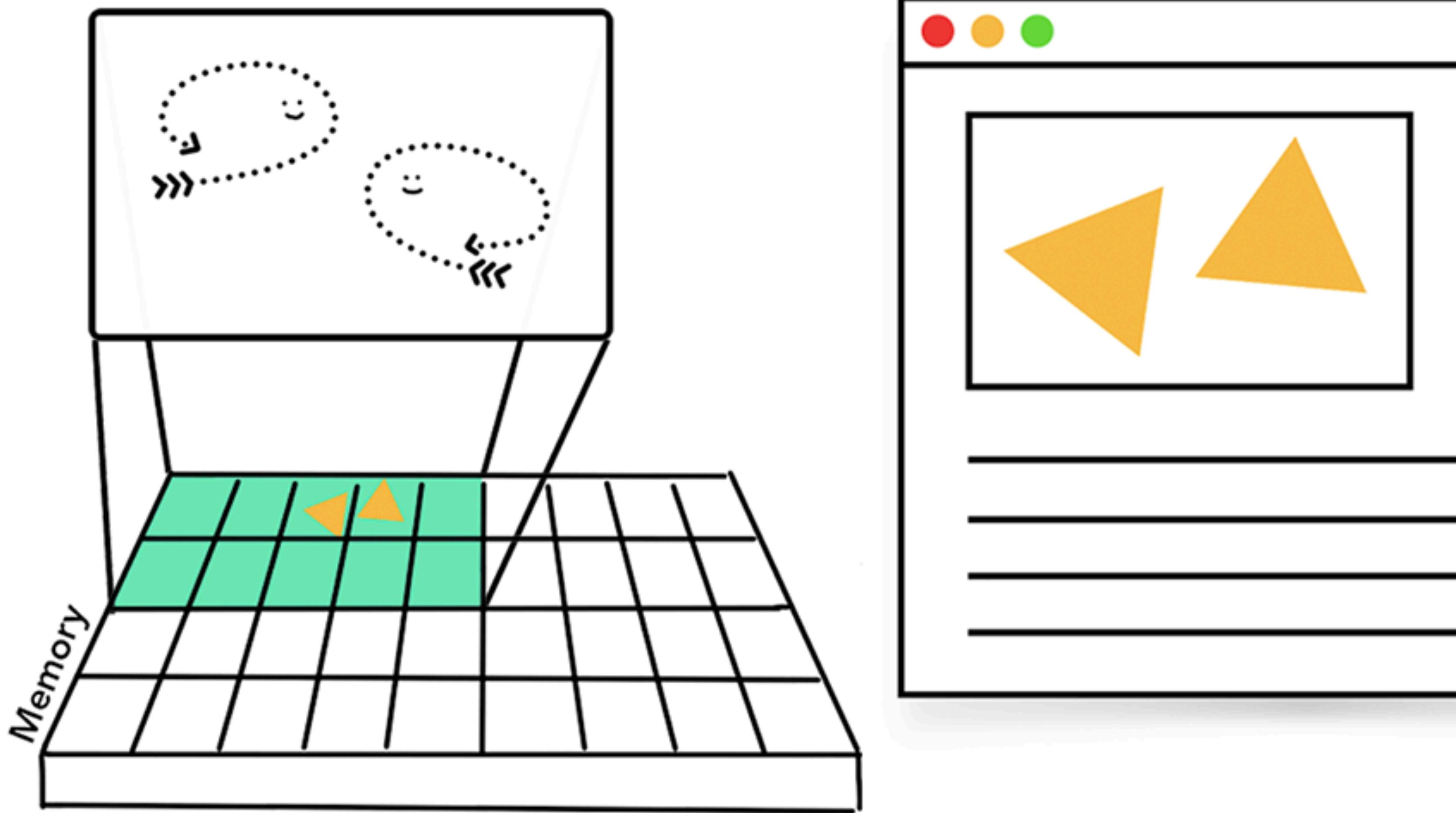
Service-based architecture – Chrome

Web Browsers



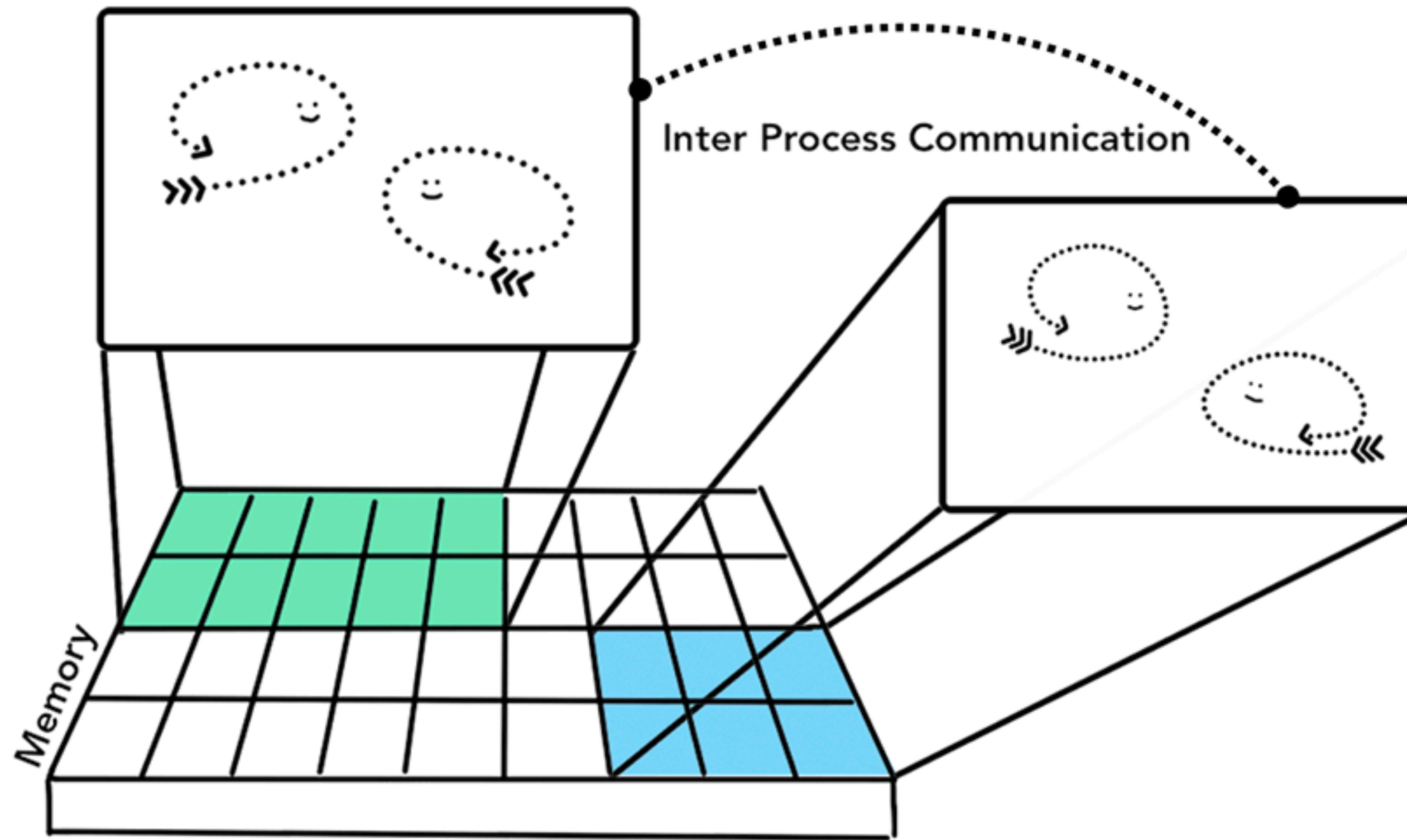
Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Browser: A multi-threaded process



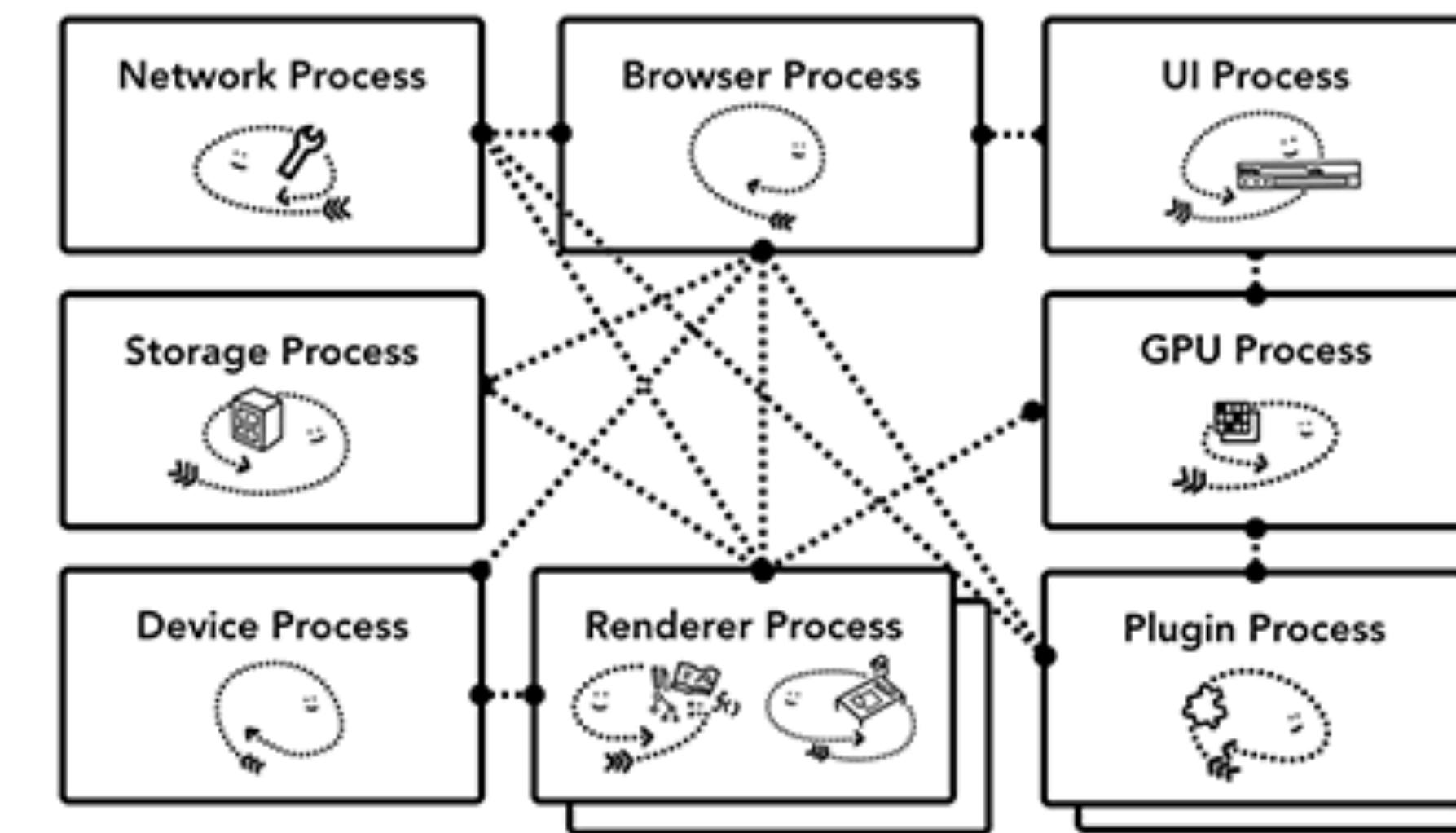
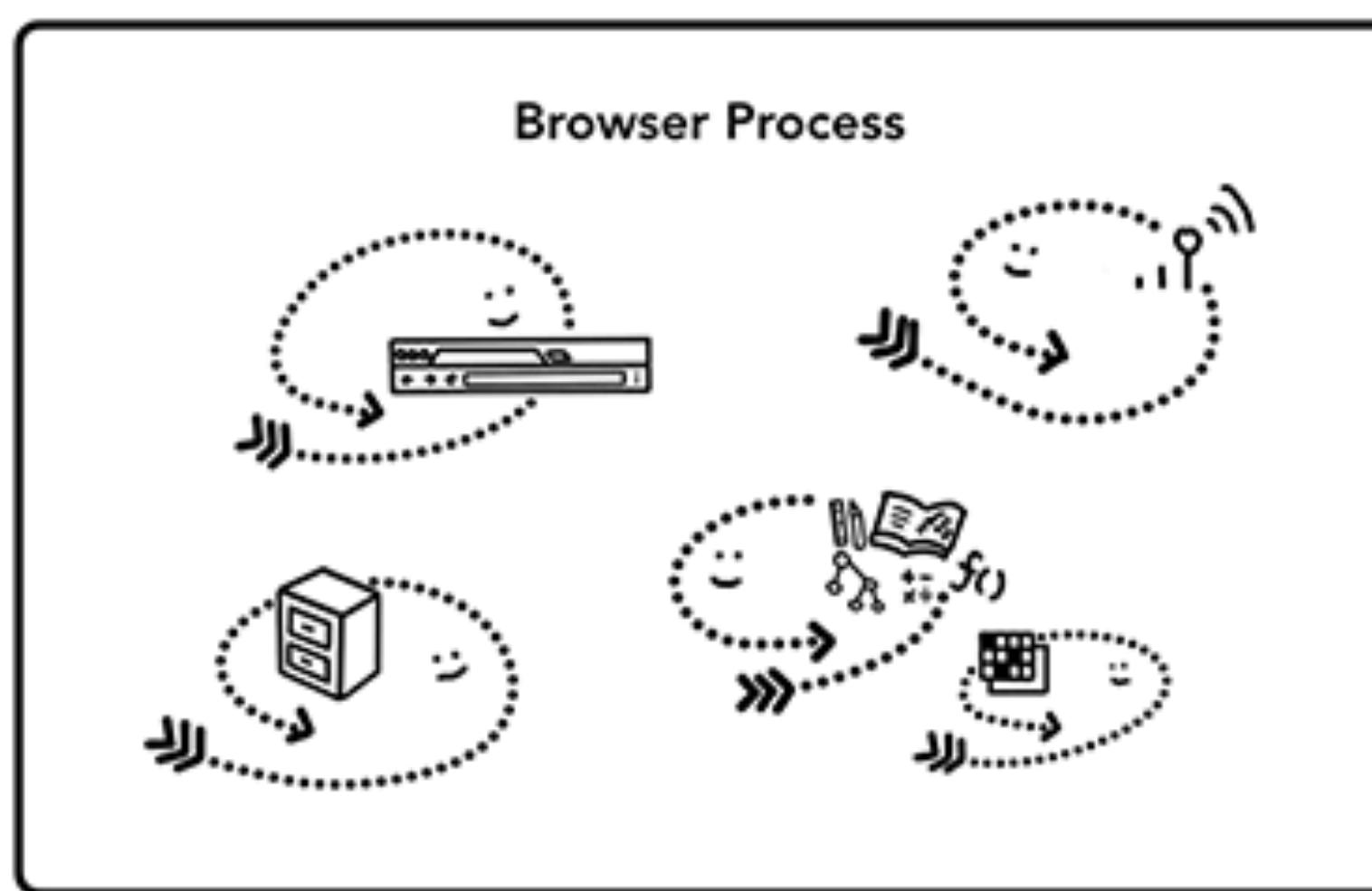
Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Multi-process browser with IPC



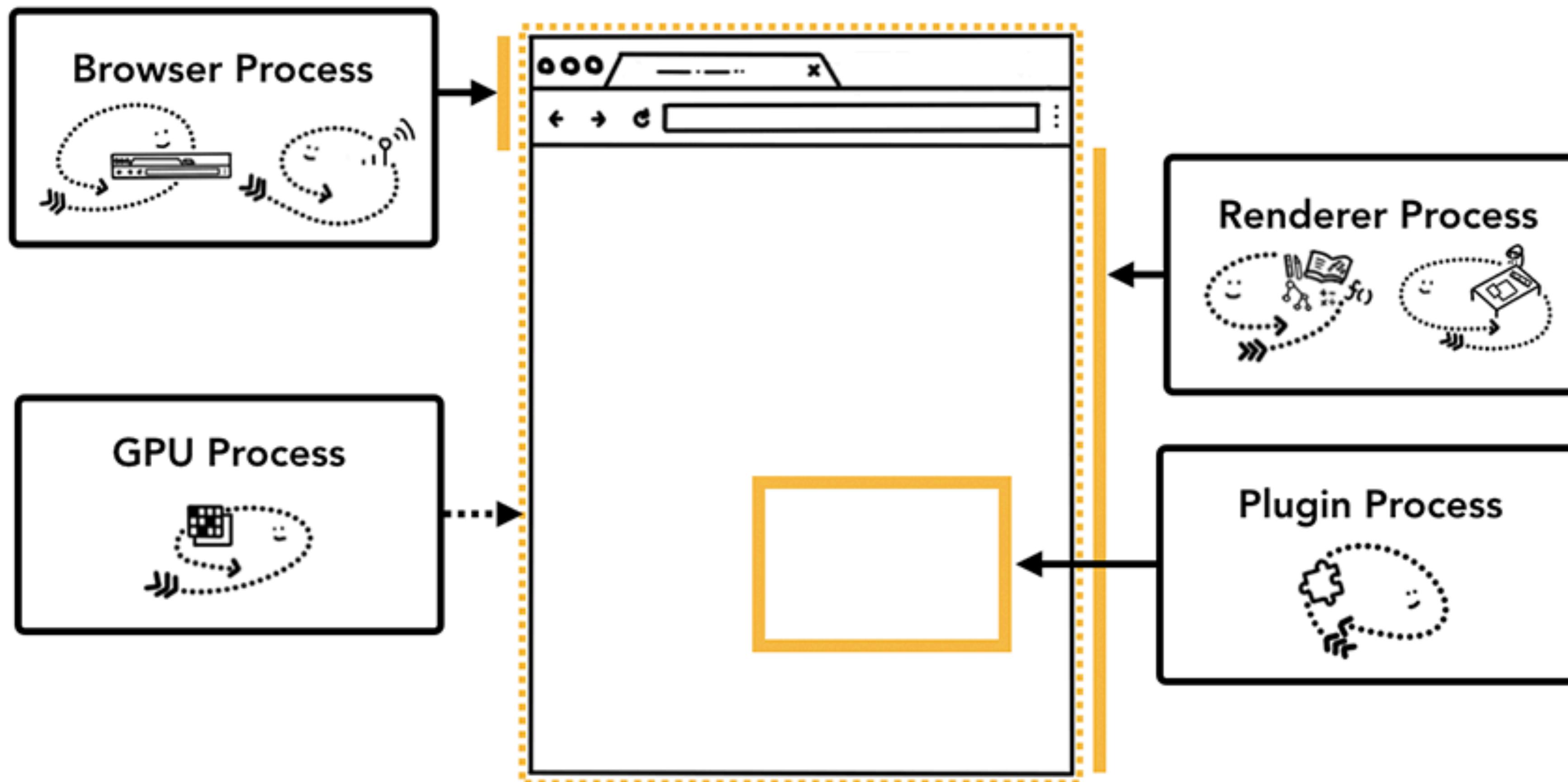
Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Browser Architectures



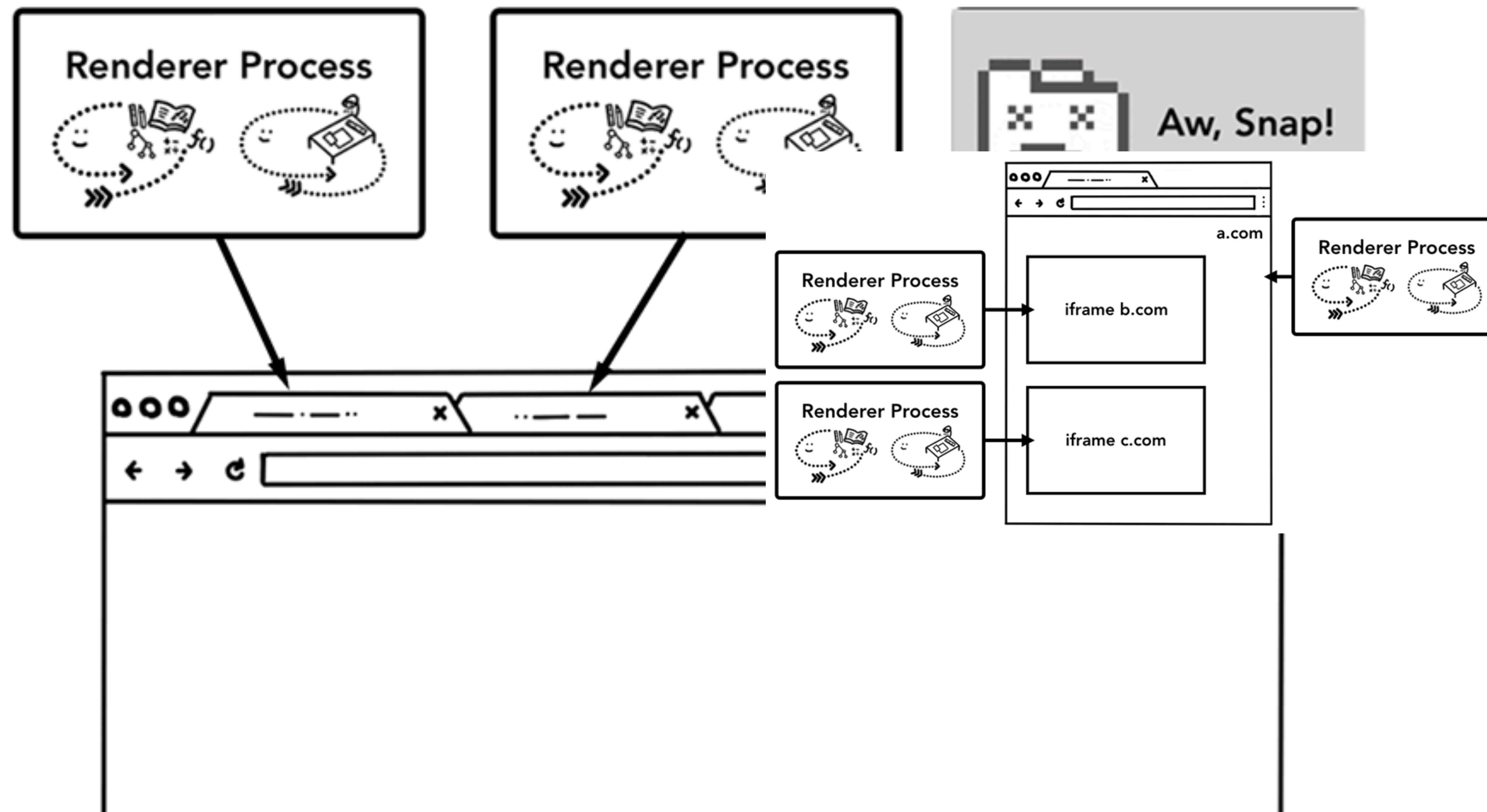
Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Service-based browser architecture



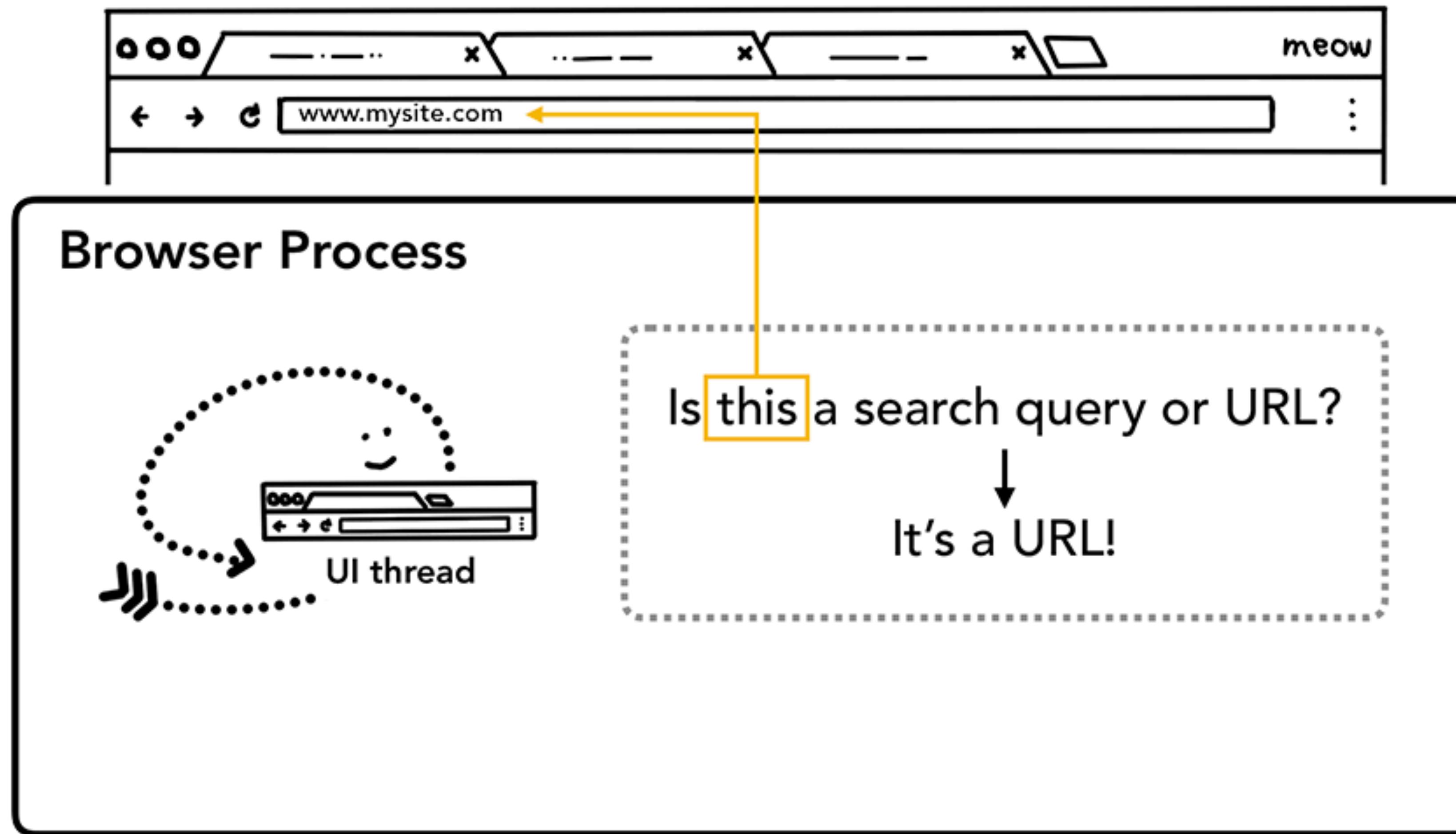
Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Service-based browser architecture

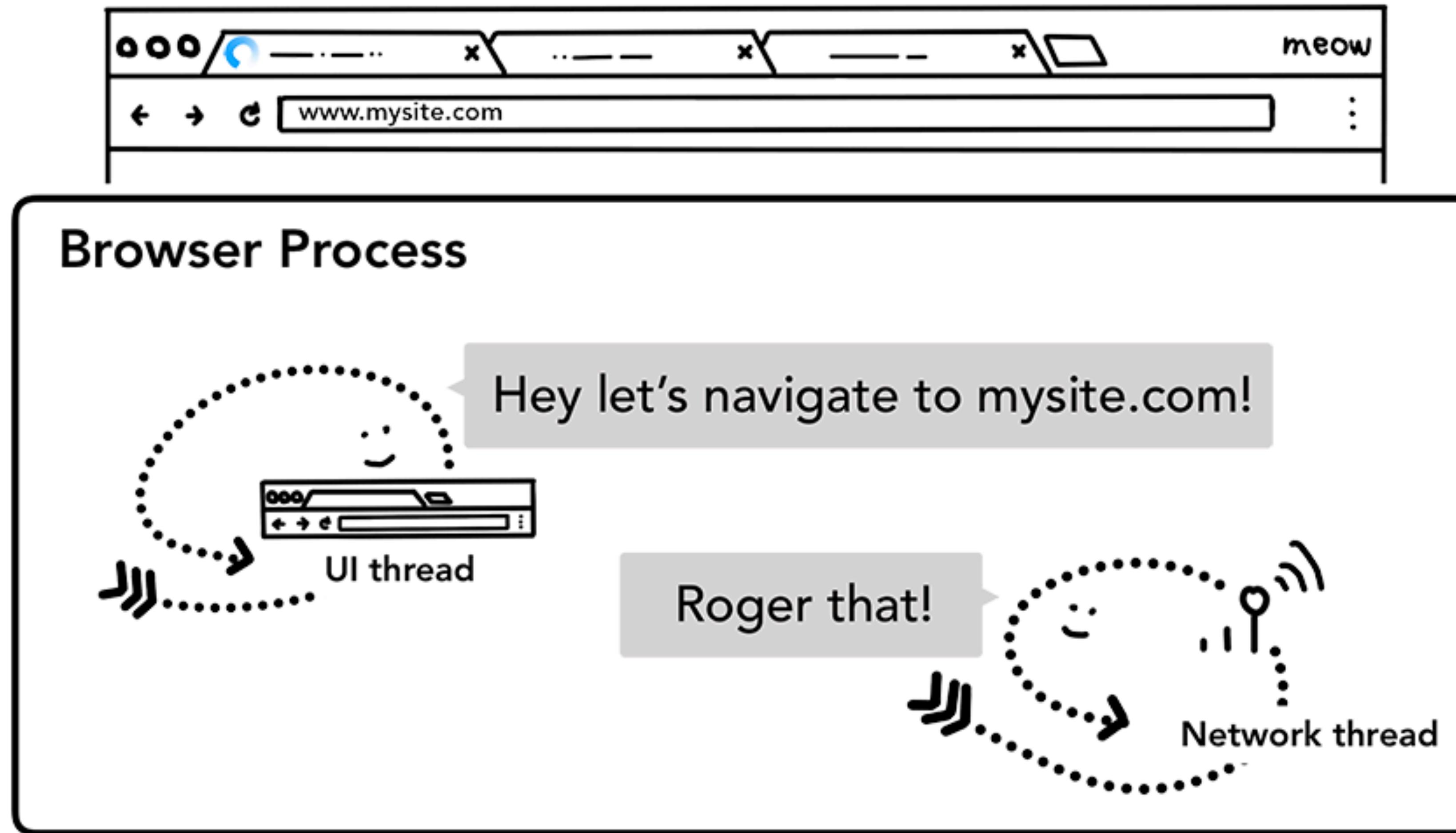


Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

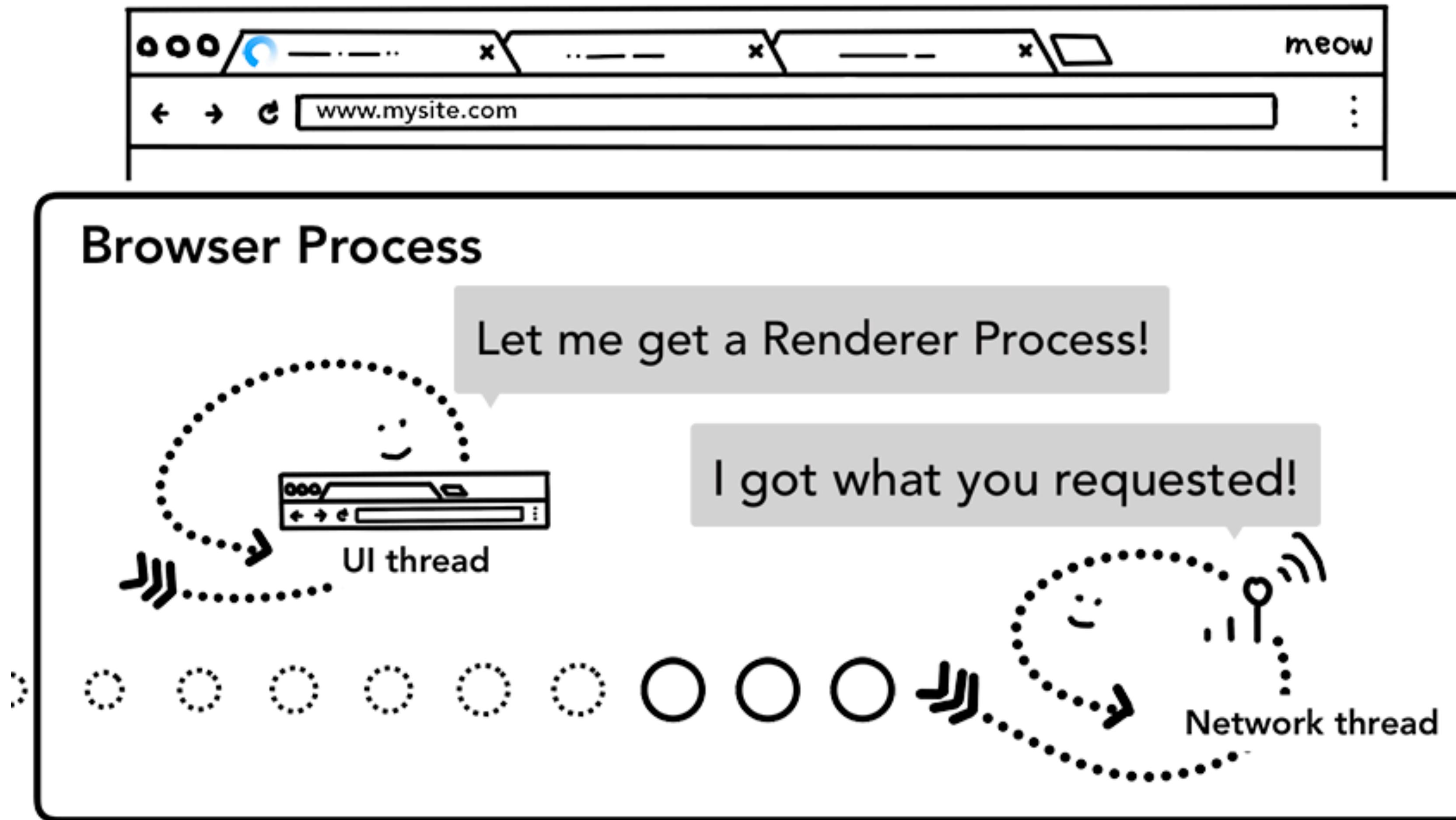
Navigating to a web site uses service requests



Navigating to a web site uses service requests

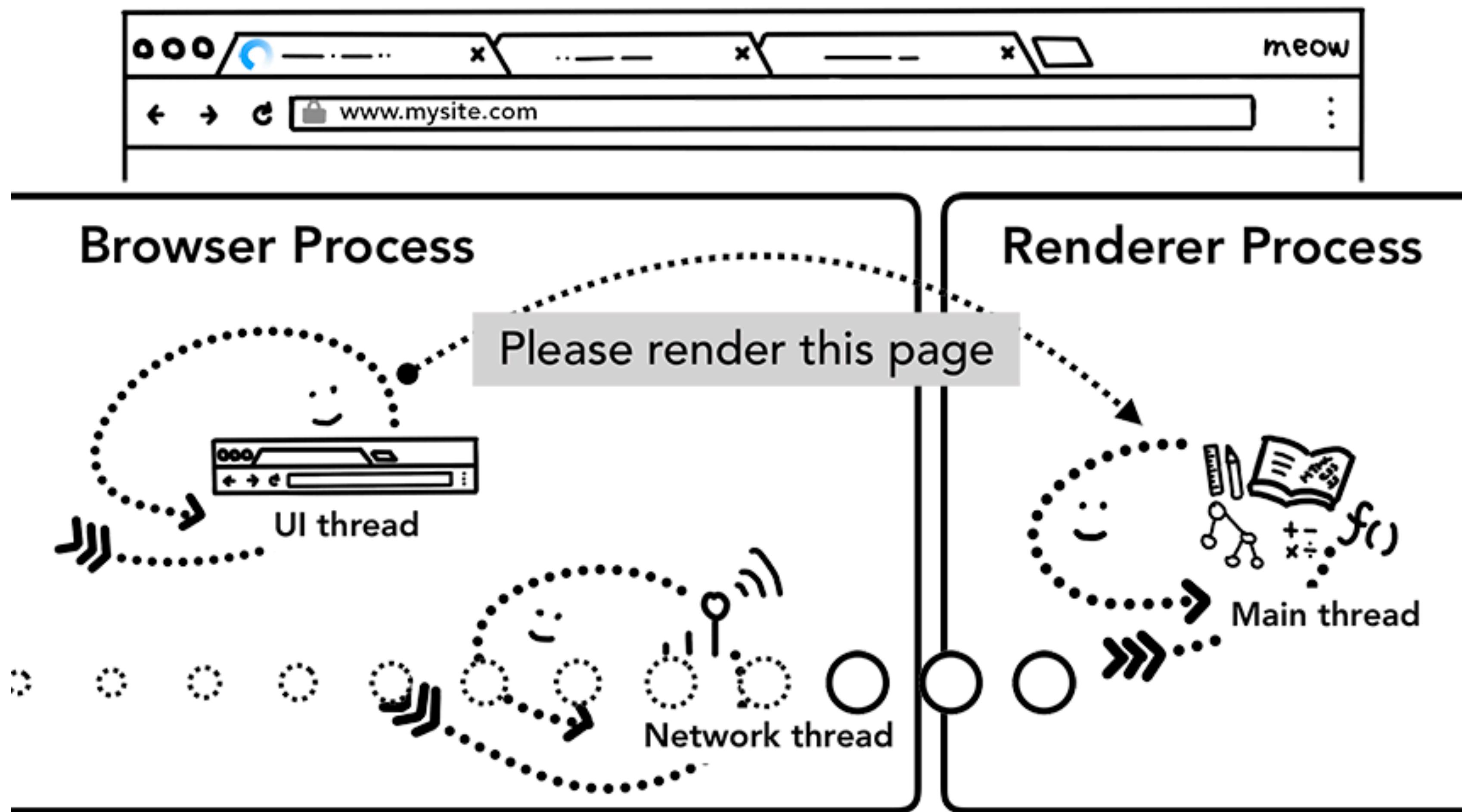


Navigating to a web site uses service requests



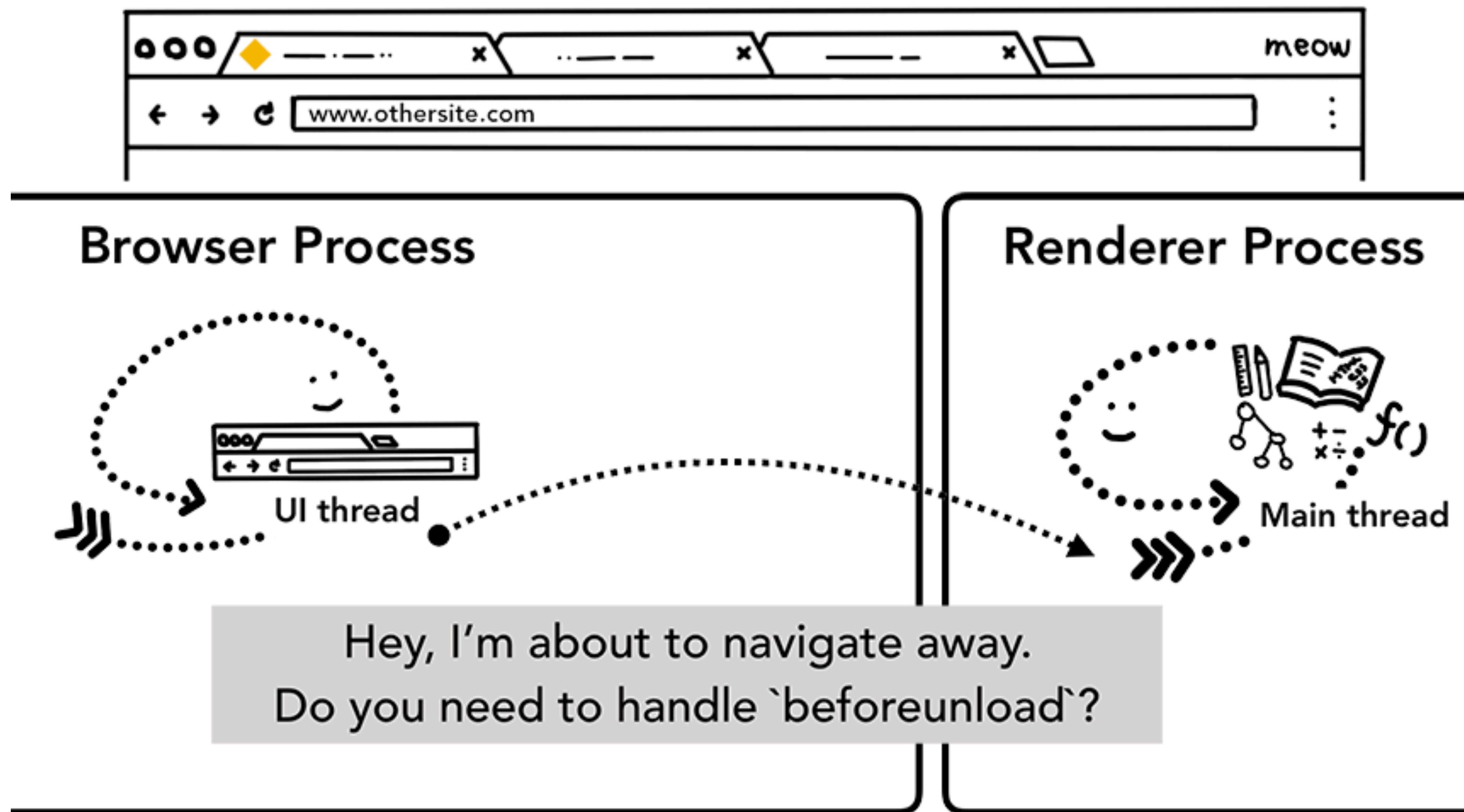
Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Navigating to a web site uses service requests

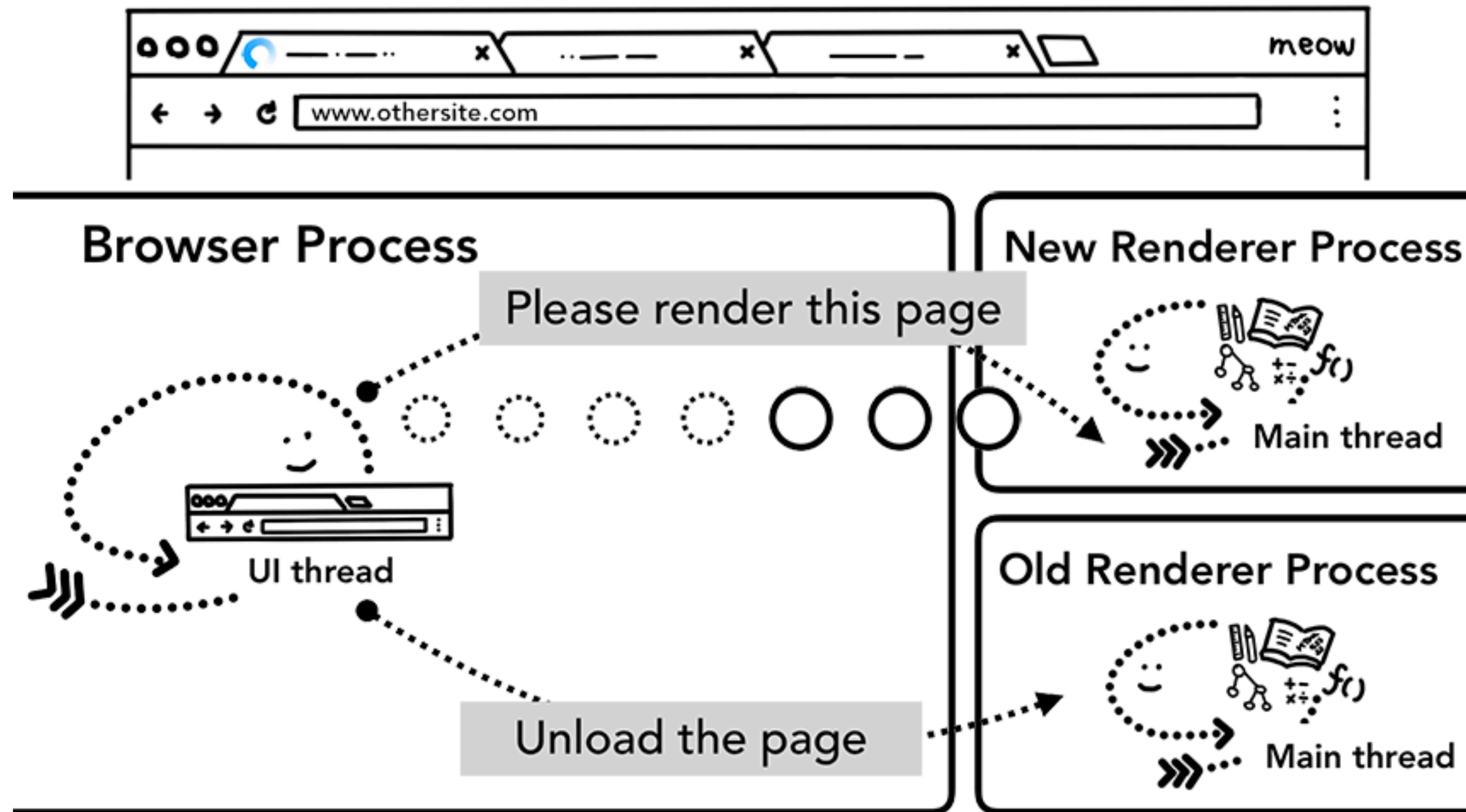


Source: <https://developers.google.com/web/updates/2018/09/inside-browser-part1> (CC BY 4.0)

Navigating to a web site uses service requests

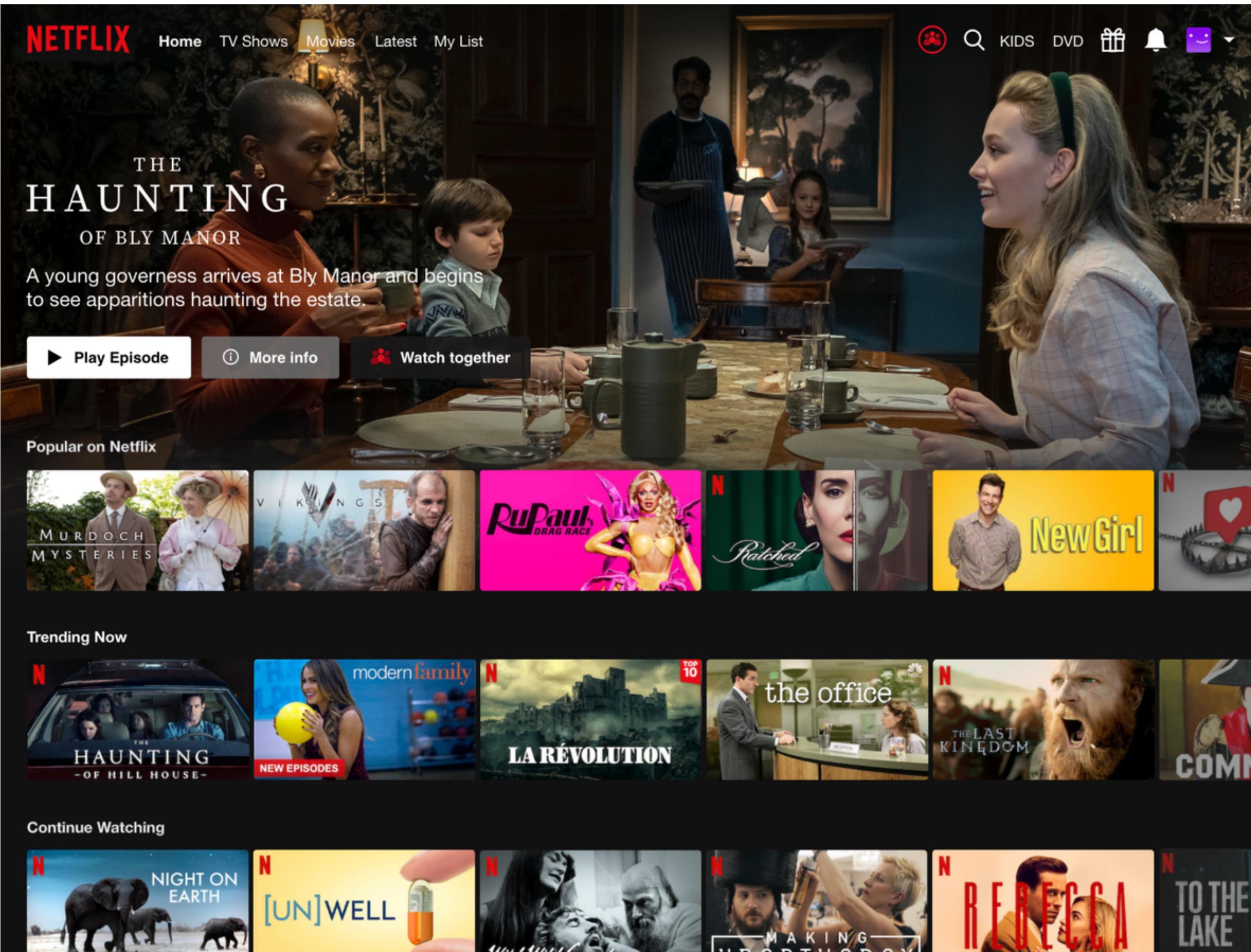


Navigating to a web site uses service requests

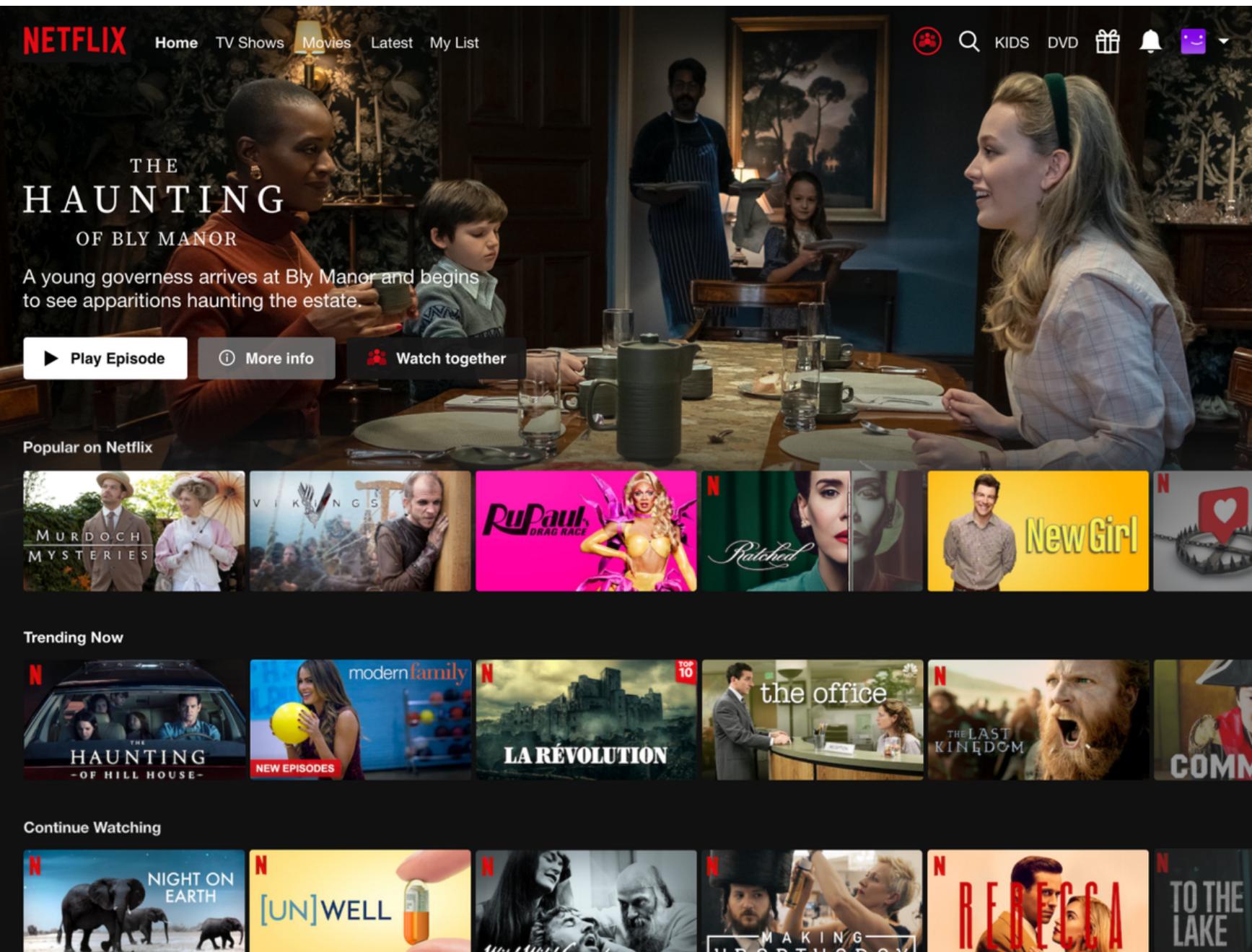


Microservice architecture - Netflix

Netflix



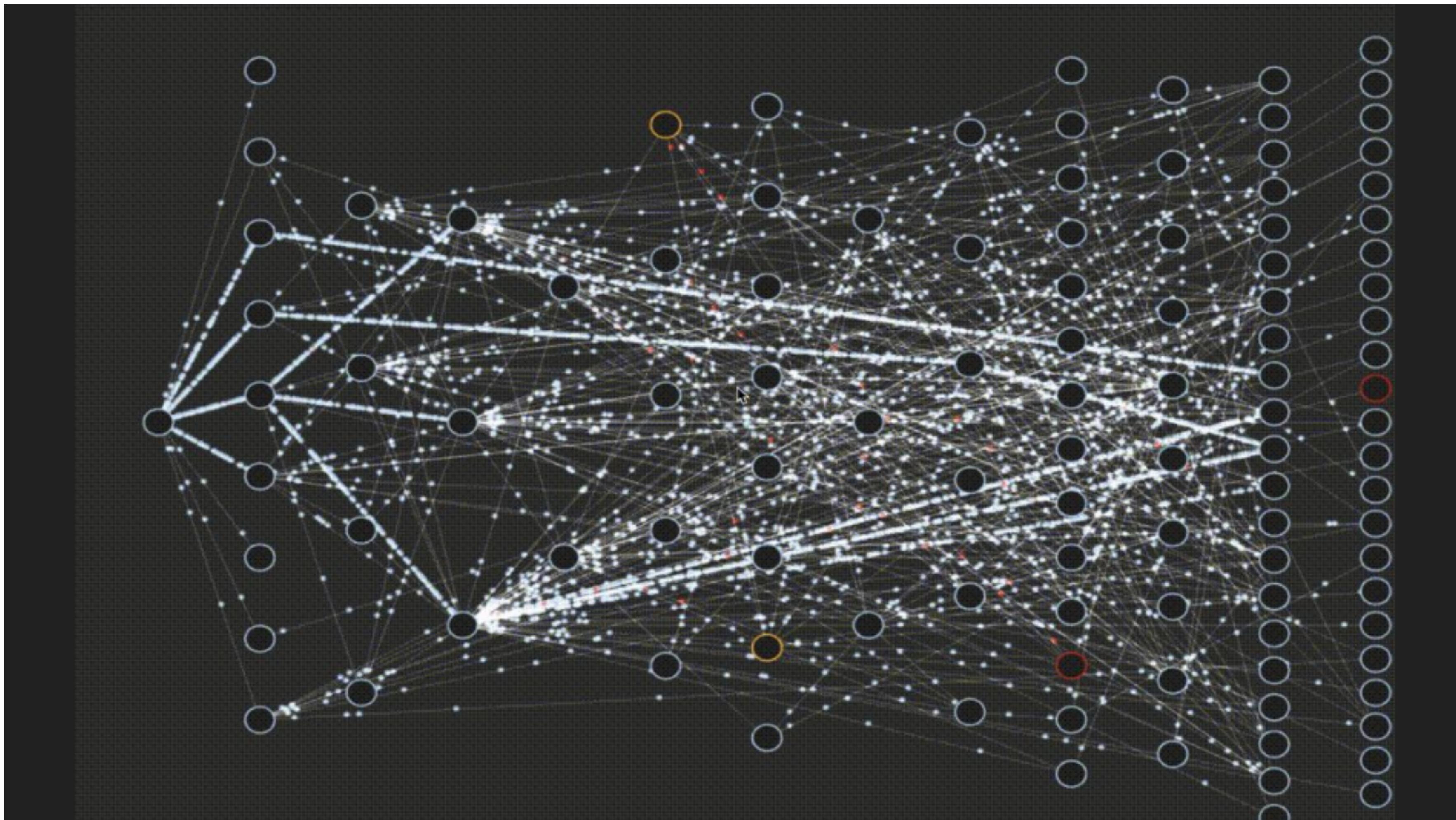
Netflix Microservices - App Boot



- Recommendations
- Trending Now
- Continue Watching
- My List
- Metrics

(as of 2016)

Netflix Microservices – One Request



(as of 2016)

<https://www.youtube.com/watch?v=CZ3wluvmHeM>

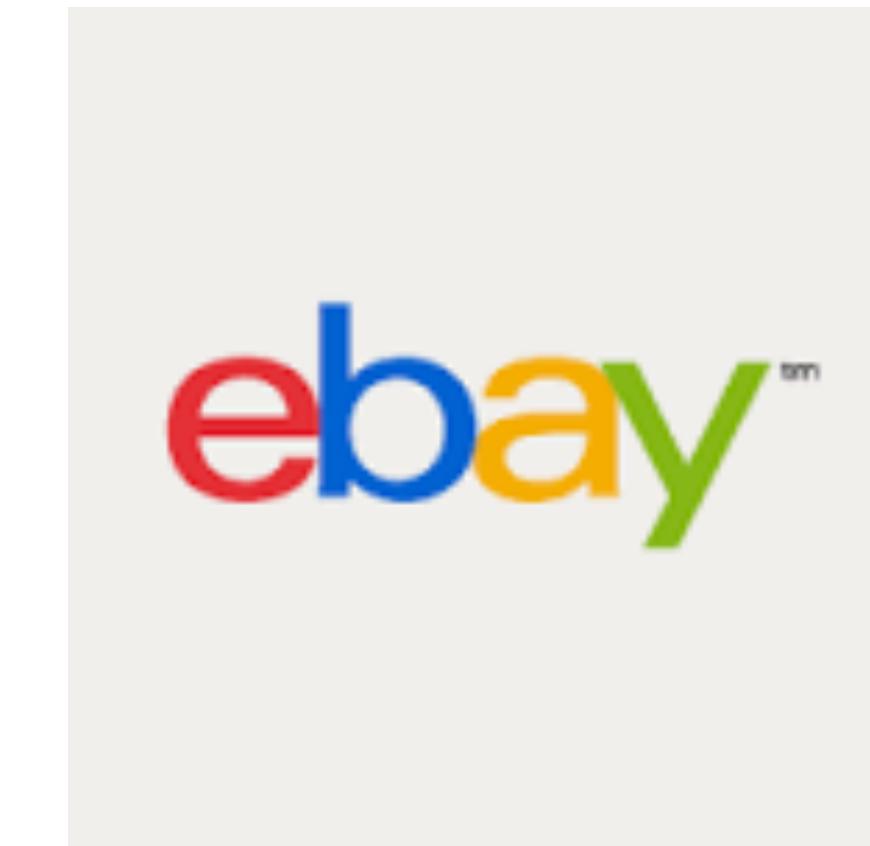
Who uses Microservices?



UBER



COMCAST



GROUPON®



Microservices – The Hipster Shop Example

Hipster Shop: Guess some microservices

The image displays two screenshots of the Online Boutique website, illustrating a microservices architecture.

Left Screenshot (Homepage):

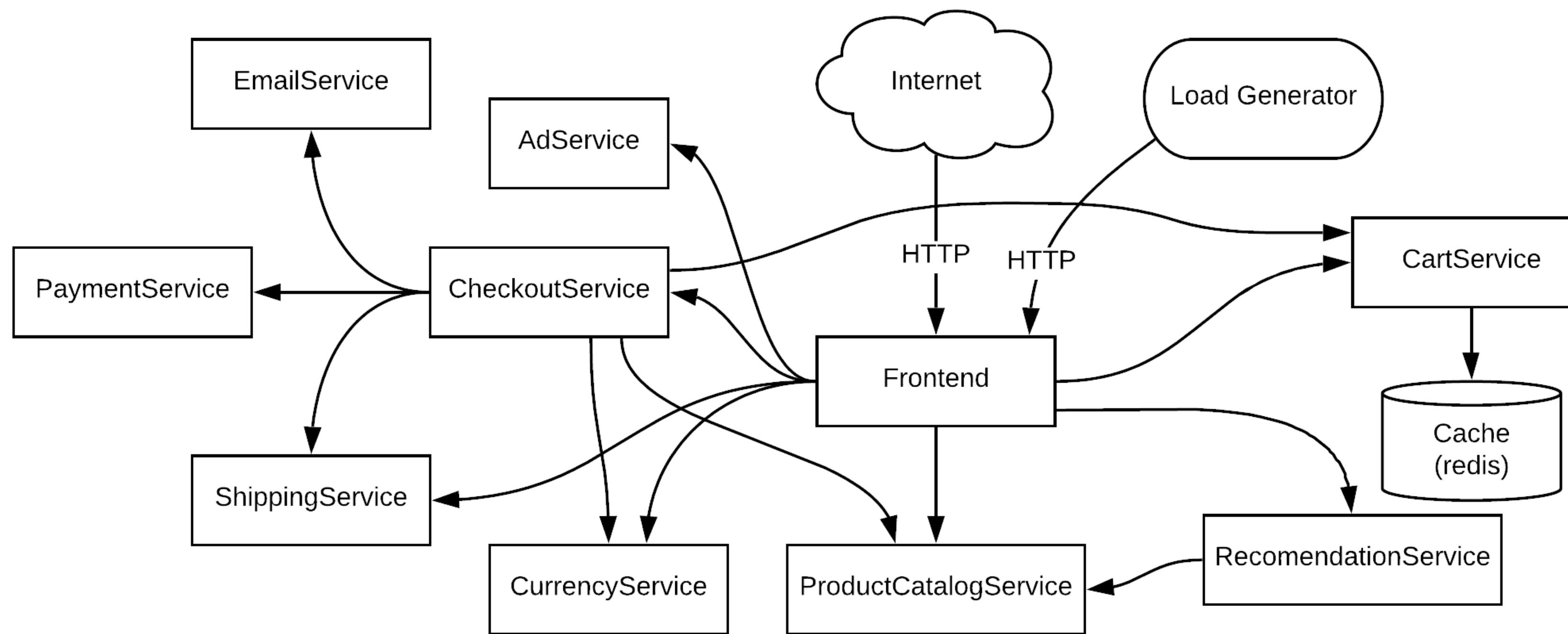
- Header: "Free shipping with \$75 purchase!"
- Logo: "ONLINEBOUTIQUE" with a teal icon.
- Cart: "Cart" button with a count of 0.
- Google Cloud badge.
- Banner: "Online BOUTIQUE EST. 2013" featuring a vintage typewriter.
- Section: "Hot PRODUCTS" with three items:
 - VINTAGE TYPEWRITER (TRY 368.34)
 - VINTAGE CAMERA LENS (TRY 67.66)
 - HOME BARISTA KIT (TRY 671.79)
- Bottom row: Three more products shown as thumbnails.

Right Screenshot (Shopping Cart):

- Header: "Free shipping with \$75 purchase!"
- Logo: "ONLINEBOUTIQUE" with a teal icon.
- Cart: "Cart" button with a count of 2.
- Google Cloud badge.
- Text: "2 items in your cart".
- Items:
 - Home Barista Kit (SKU: #1YMWNN1N40) - Quantity: 1, TRY 671.79
 - Vintage Camera Lens (SKU: #66VCHSJNUP) - Quantity: 5, TRY 338.33
- Text: "Shipping Cost: TRY 93.23" and "Total Cost: TRY 1103.36".
- Section: "Checkout".
- Form fields:
 - 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: ...
- Buttons: "PLACE ORDER" (teal), "EMPTY CART", and "KEEP BROWSING".

<https://onlineboutique.dev>

Hipster Shop Microservice Architecture



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

Microservices

What are the consequences of this architecture? On:

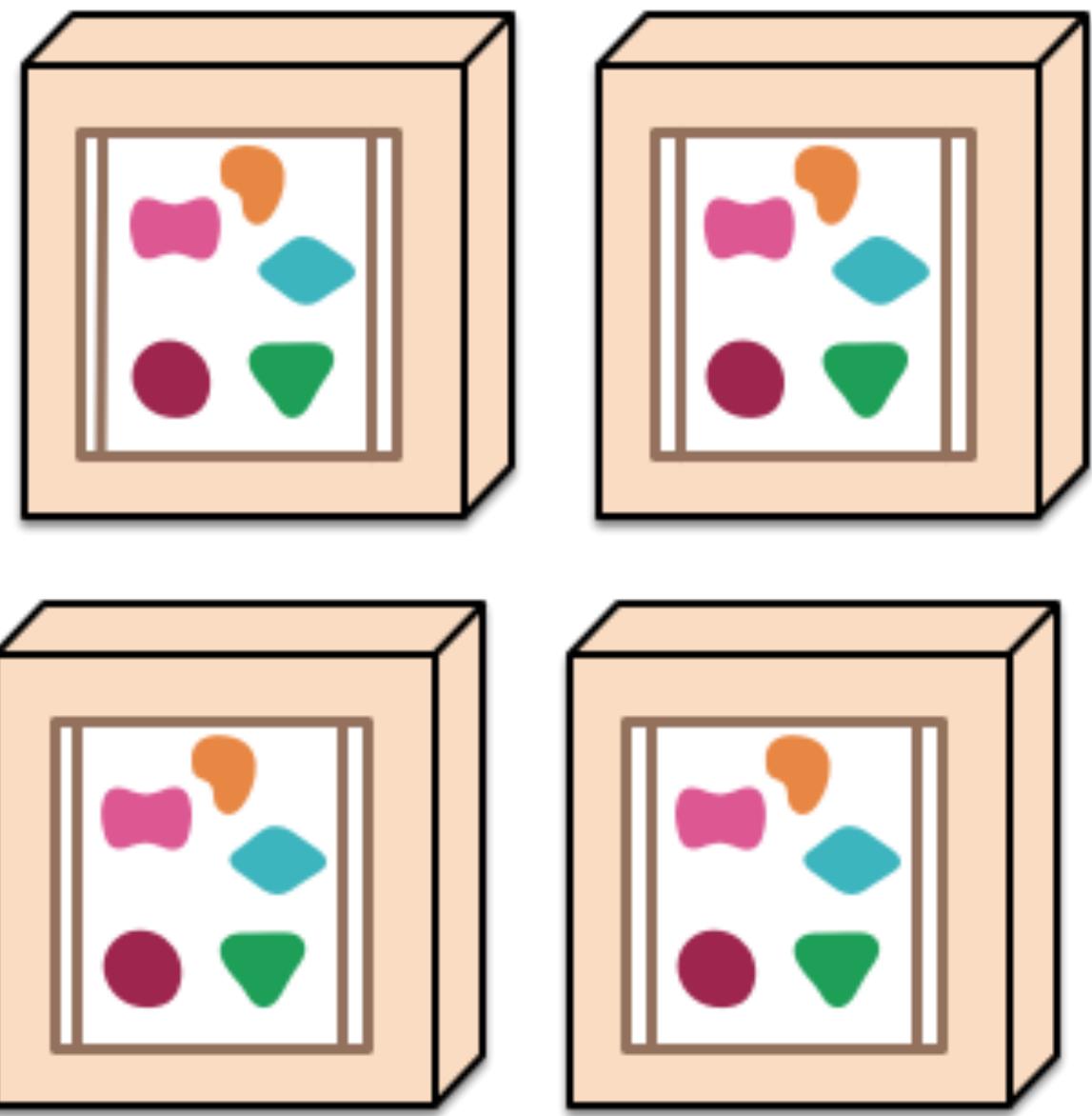
- Scalability
- Reliability
- Performance
- Development
- Maintainability
- Evolution
- Testability
- Ownership
- Data Consistency

Scalability

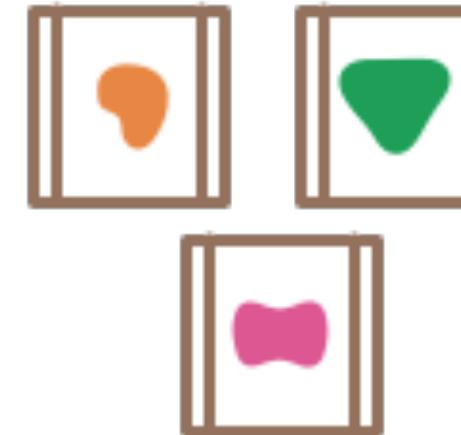
A monolithic application puts all its functionality into a single process...



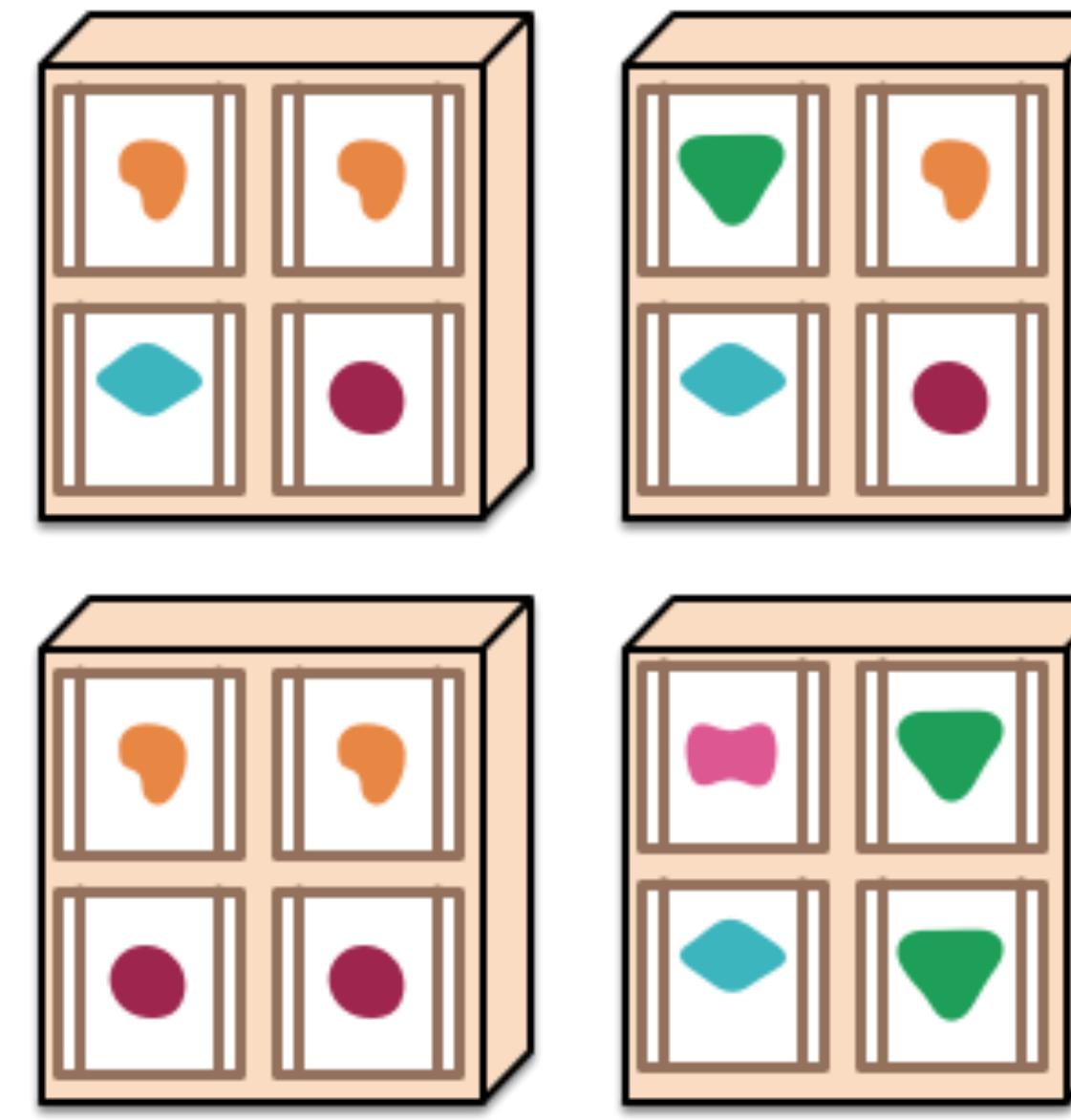
... and scales by replicating the monolith on multiple servers



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

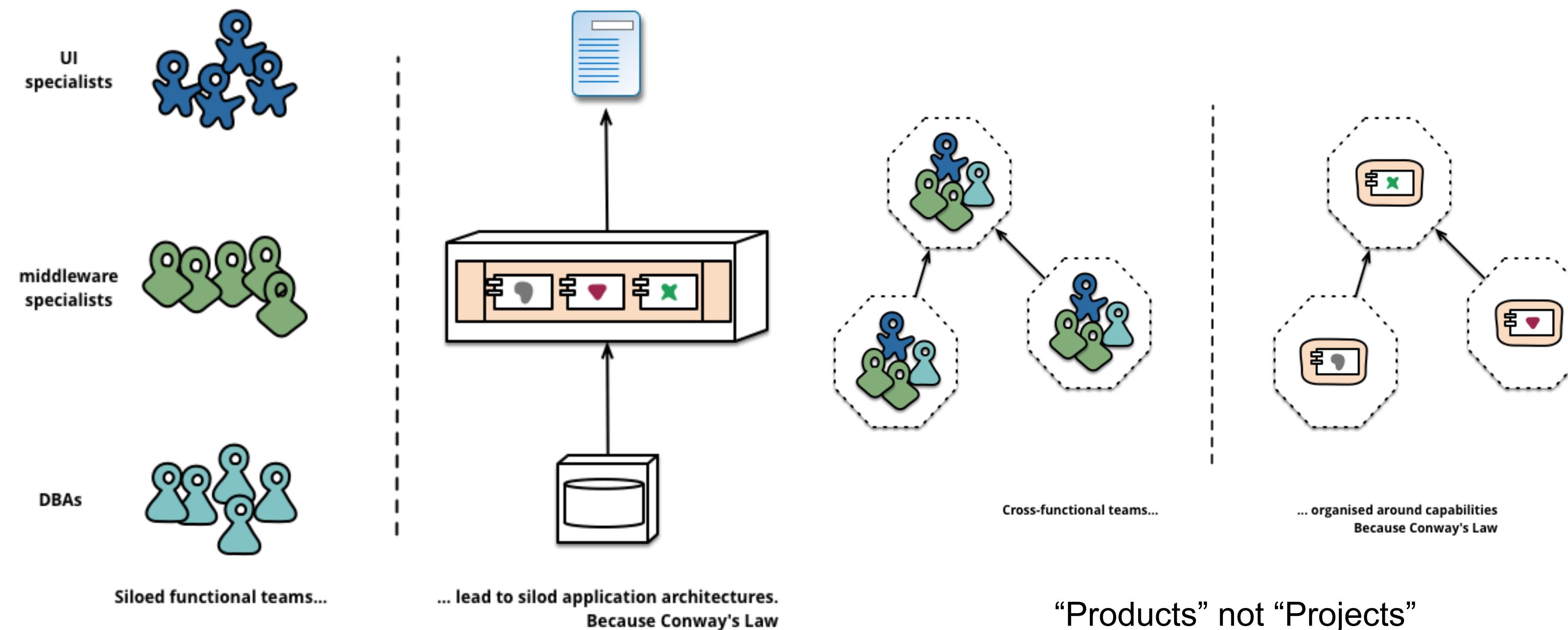


... and scales by distributing these services across servers, replicating as needed.



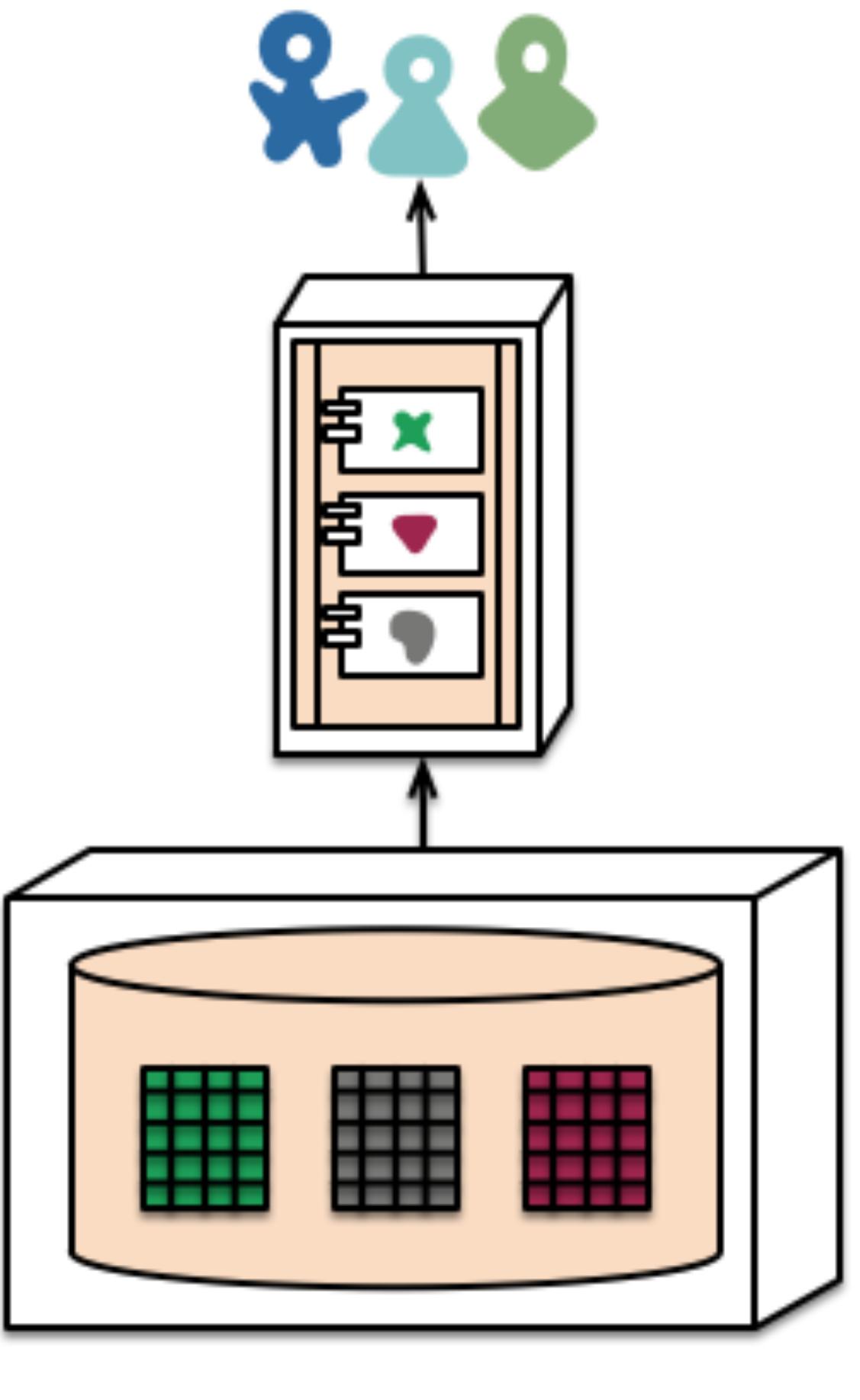
Source: <http://martinfowler.com/articles/microservices.html>

Team Organization (Conway's Law)

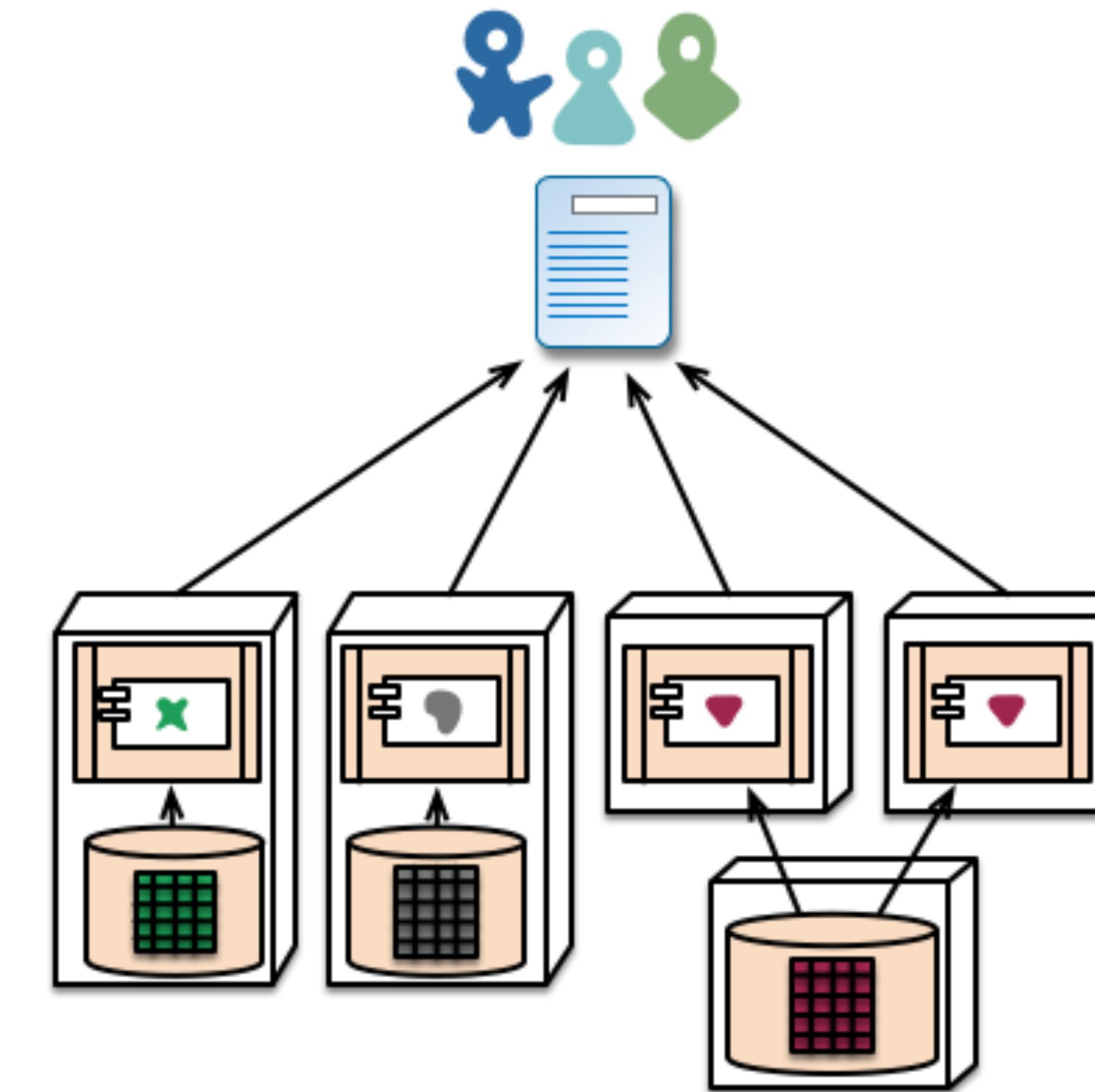


Source: <http://martinfowler.com/articles/microservices.html>

Data Management and Consistency



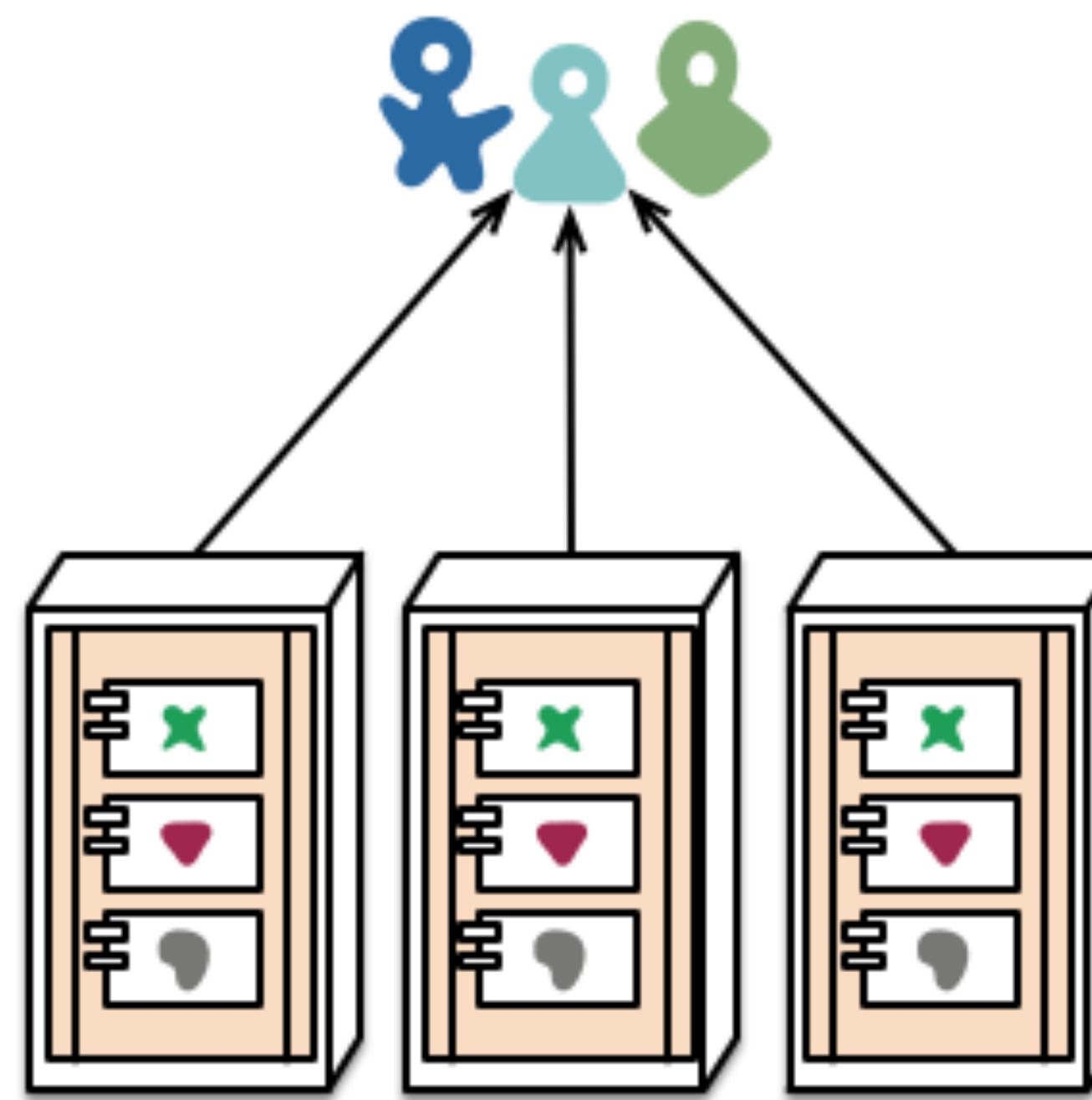
monolith - single database



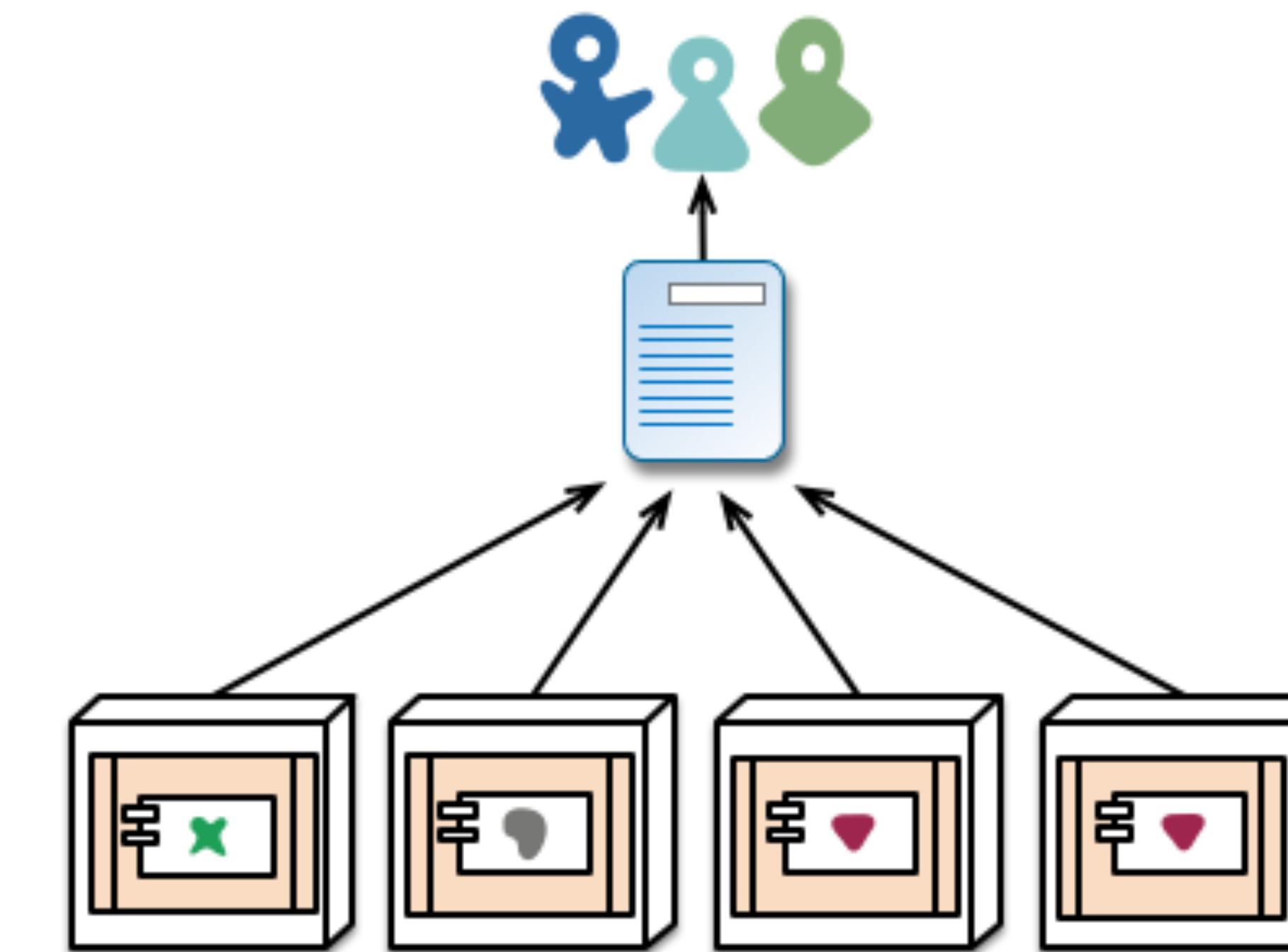
microservices - application databases

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>

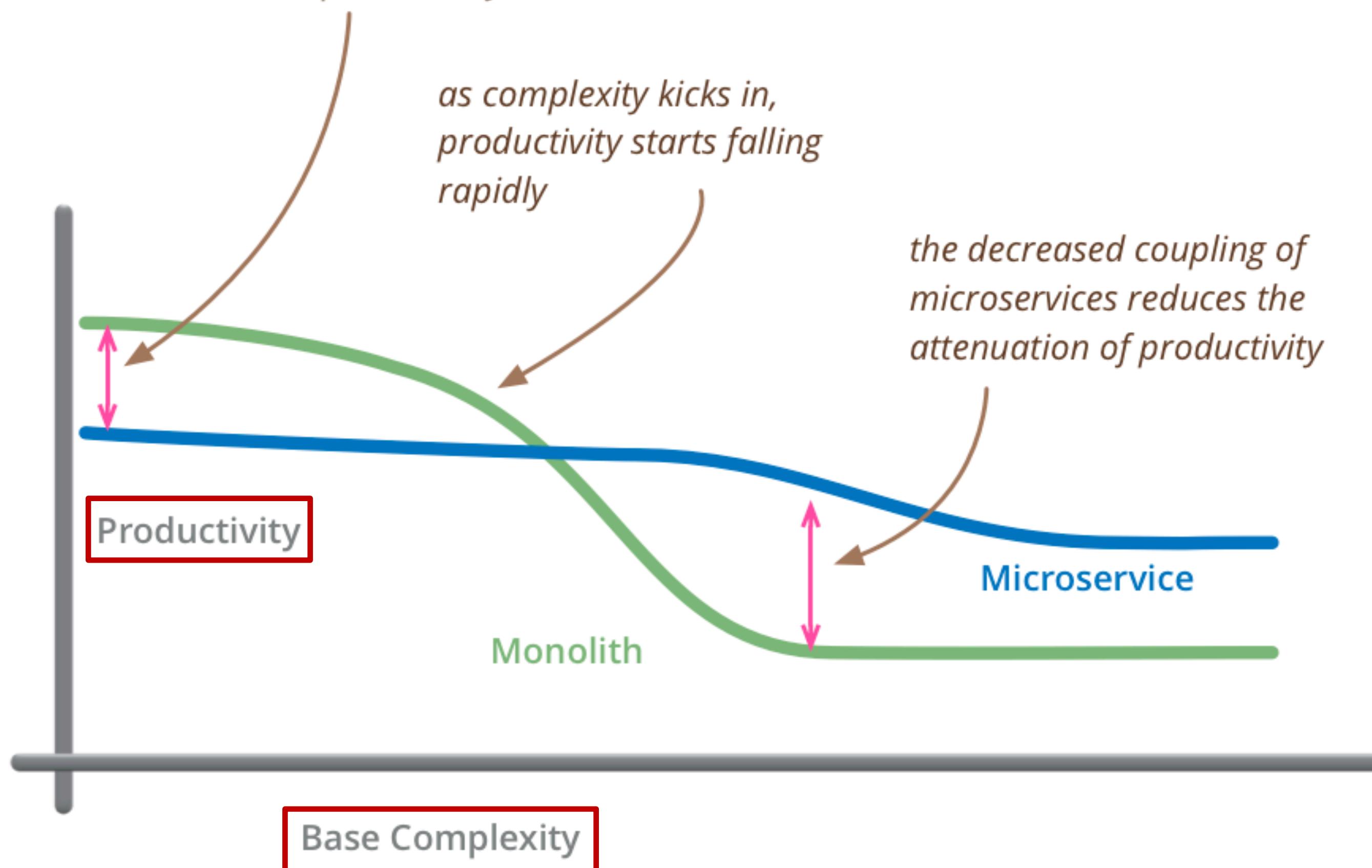
Microservices

- 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

Are microservices always the right choice?

Microservices overhead

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



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?

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?

More in: API testing and DevOps

