



**Jouke Mandemaker**

[jouke.mandemaker@codesharp.nl](mailto:jouke.mandemaker@codesharp.nl)

Code Sharp B.V.

**Software developer**

CTO and founder

An introduction to

# Micro frontends

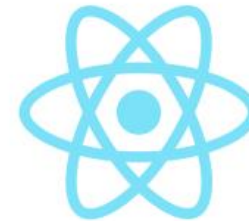
Extending the microservice idea to frontend development

# Agenda

- Introduction
- Solutions
- Demo
- Questions



svelte

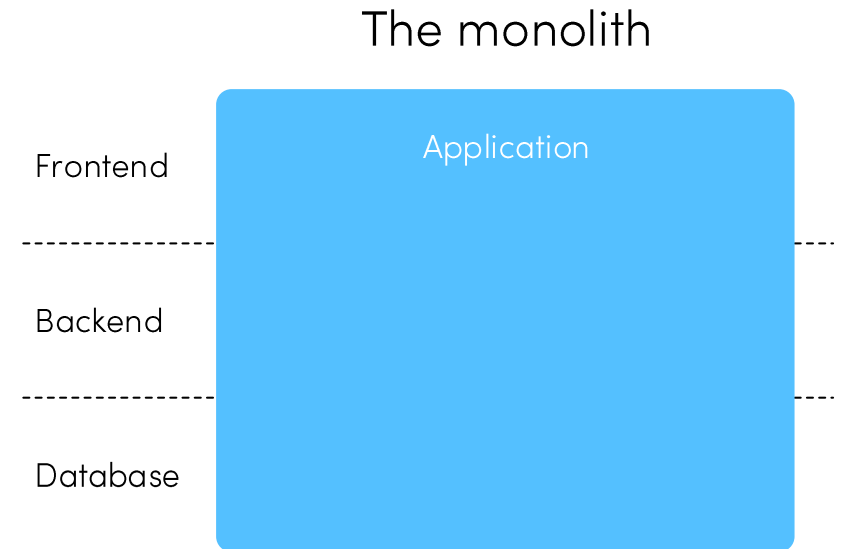


The monolith



# The monolith

- **Pros**
  - Well known
  - Easy deployment
- **Cons**
  - Hard to maintain
  - Complex
  - One stack



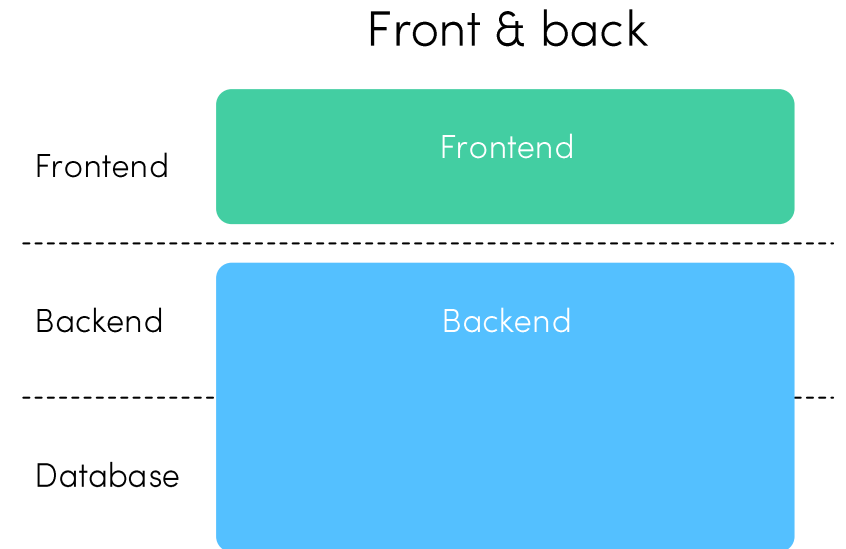
# Separate frontend

- **Pros**

- Well known
- Separate logic and UI

- **Cons**

- Hard to maintain
- Complex
- Harder deployment



# Microservices

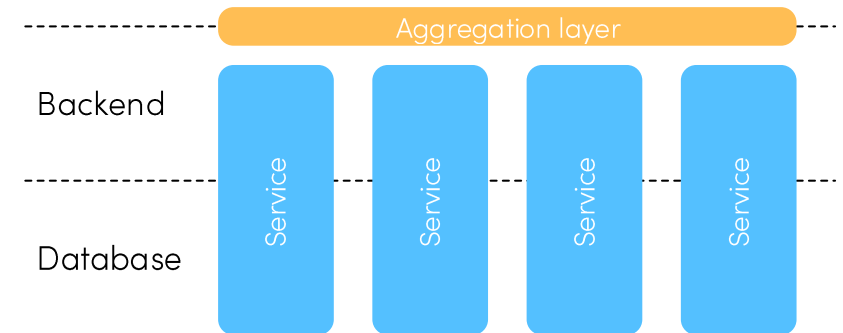
- **Pros**

- Scalability
- Resilience
- Flexible stack

- **Cons**

- Platform complexity
- Distributed transactions

## Microservices



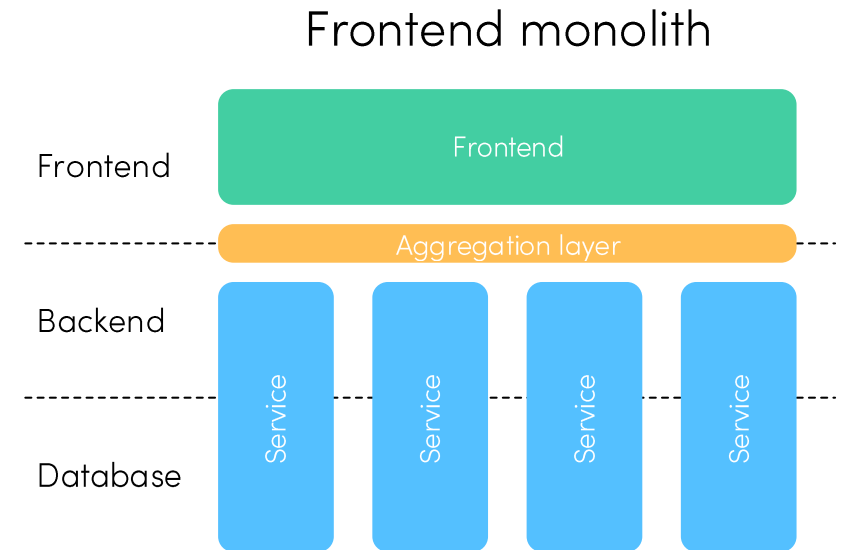
# The frontend monolith

- **Pros**

- Well known
- Framework support

- **Cons**

- Coordinated deployments
- Tight coupling
- Development bottleneck





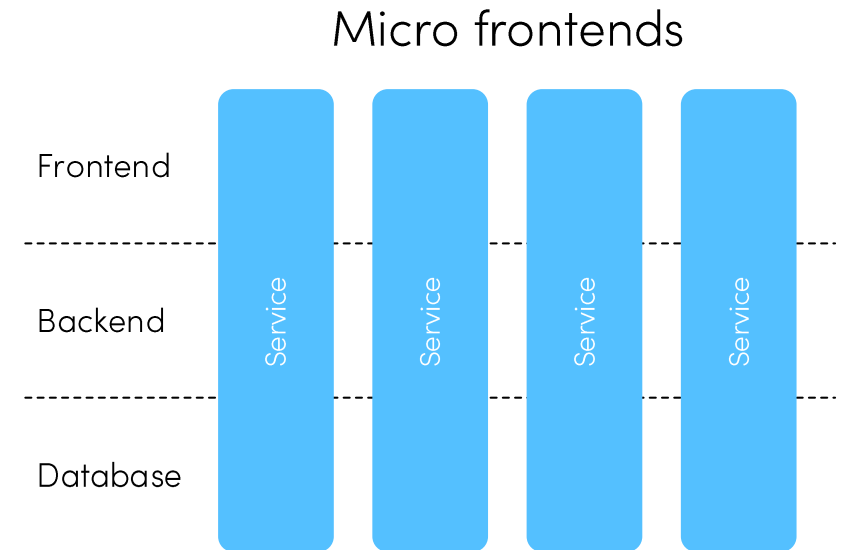
# Micro frontends

- **Pros**

- Independent deployments
- Decoupled codebases
- Team autonomy

- **Cons**

- Operational complexity
- Consistent experience



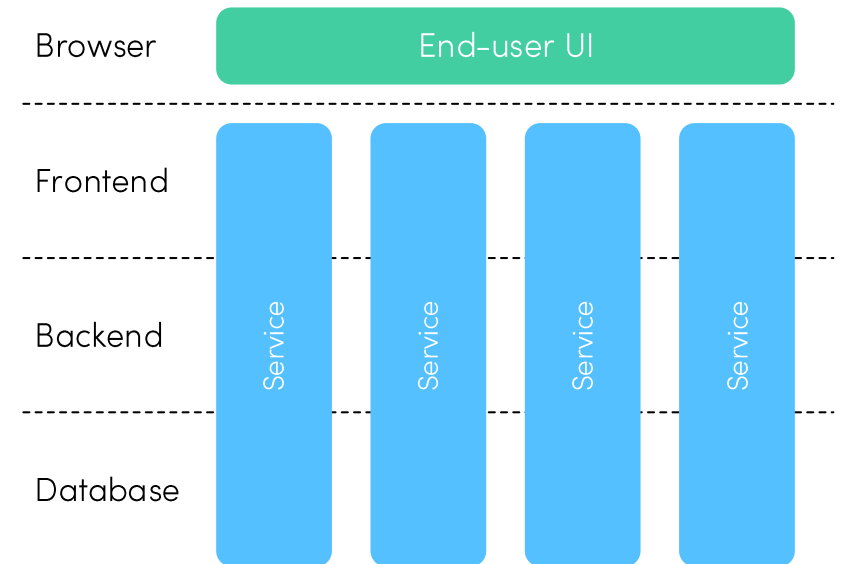
Breaking it apart



Breaking it apart

# The goal

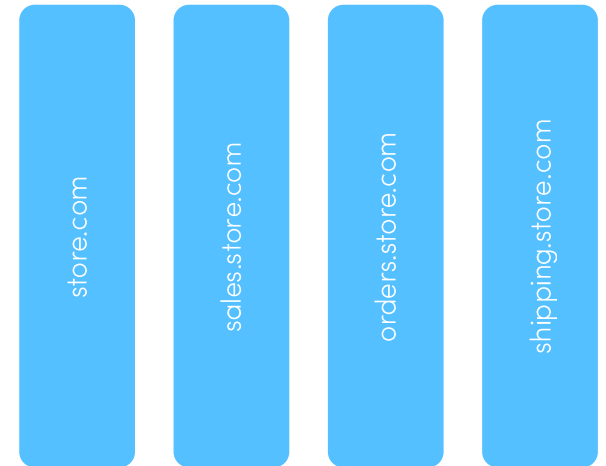
- Compose single UI
- Smooth end-user experience
- Hide backend complexity



Breaking it apart

# Don't compose

- Separate applications
- Share a little: CSS, fonts, images etc.



Where do we compose?

Browser

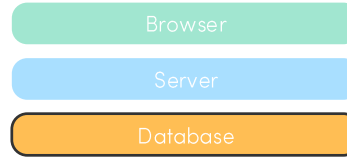
Server

Database

Breaking it apart

# Database

- Compose and aggregate data
- For example: Search



```
{
  "_id": "8a8f0736-a64a-43c0-8662-a9b3f571ed8e",
  "name": "Domain-Driven Design",
  "author": "Eric Evans",
  "formats": [
    { "format": "e-book", "price": "$30.22" },
    { "format": "hardcover", "price": "$54.78" },
  ],
  "Reviews": {
    "average": 0.8,
    "total": 114
  }
}
```

Catalog service

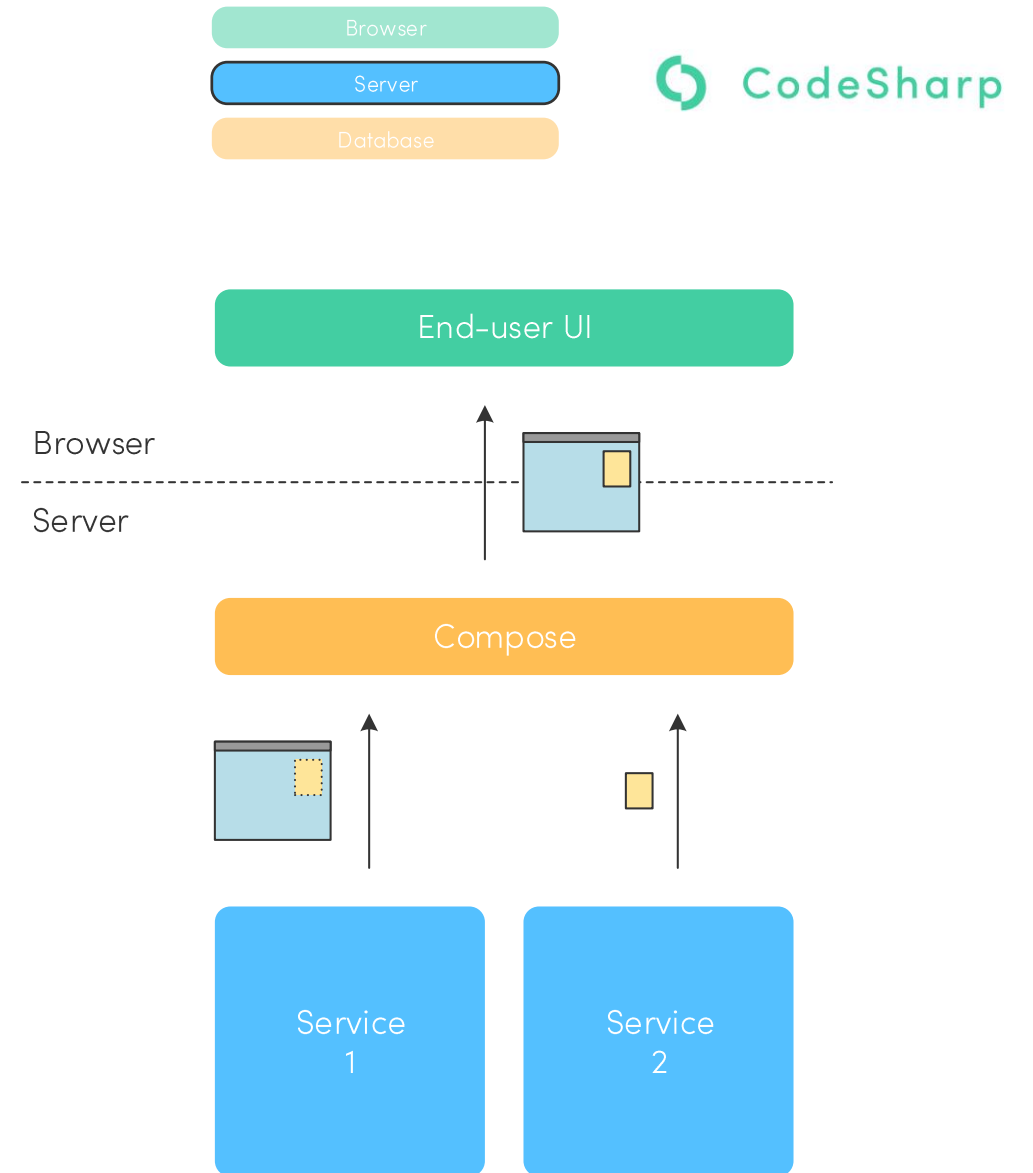
Format service

Review service

Breaking it apart

# Server

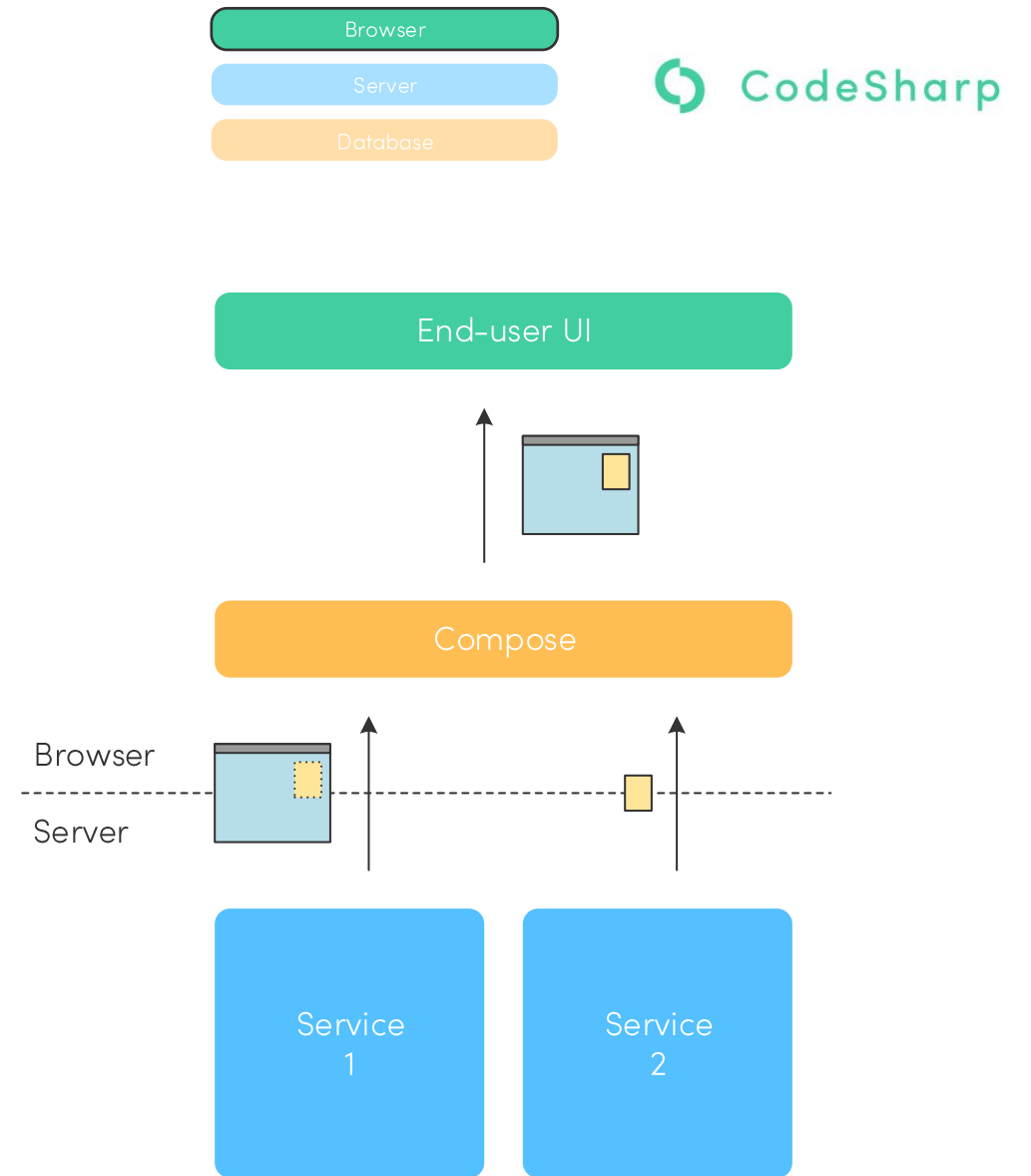
- Serves the HTML, JavaScript etc.



Breaking it apart

# Browser

- Renders the HTML
- Executes JavaScript





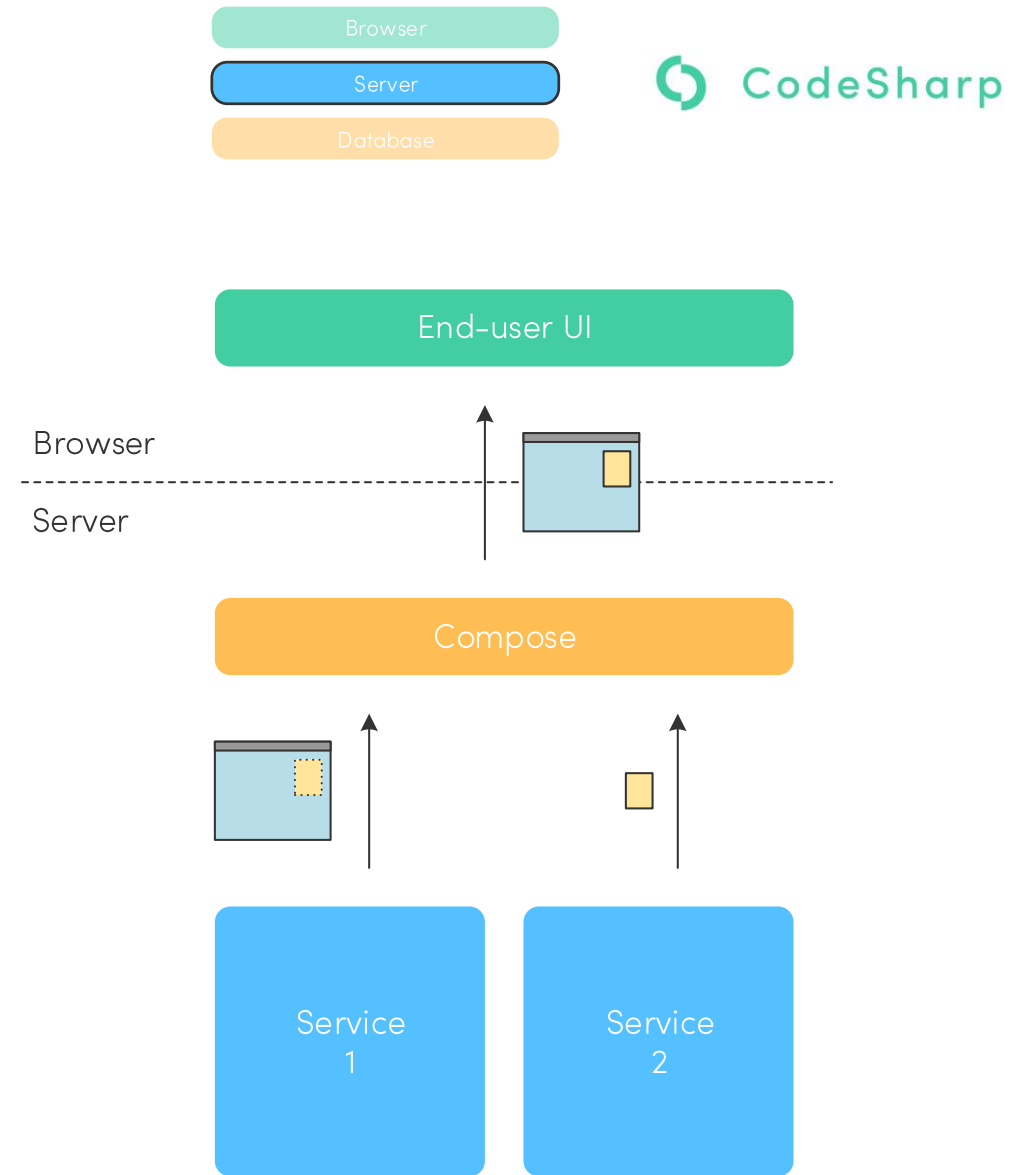
# Server composition



Server composition

# Server composition

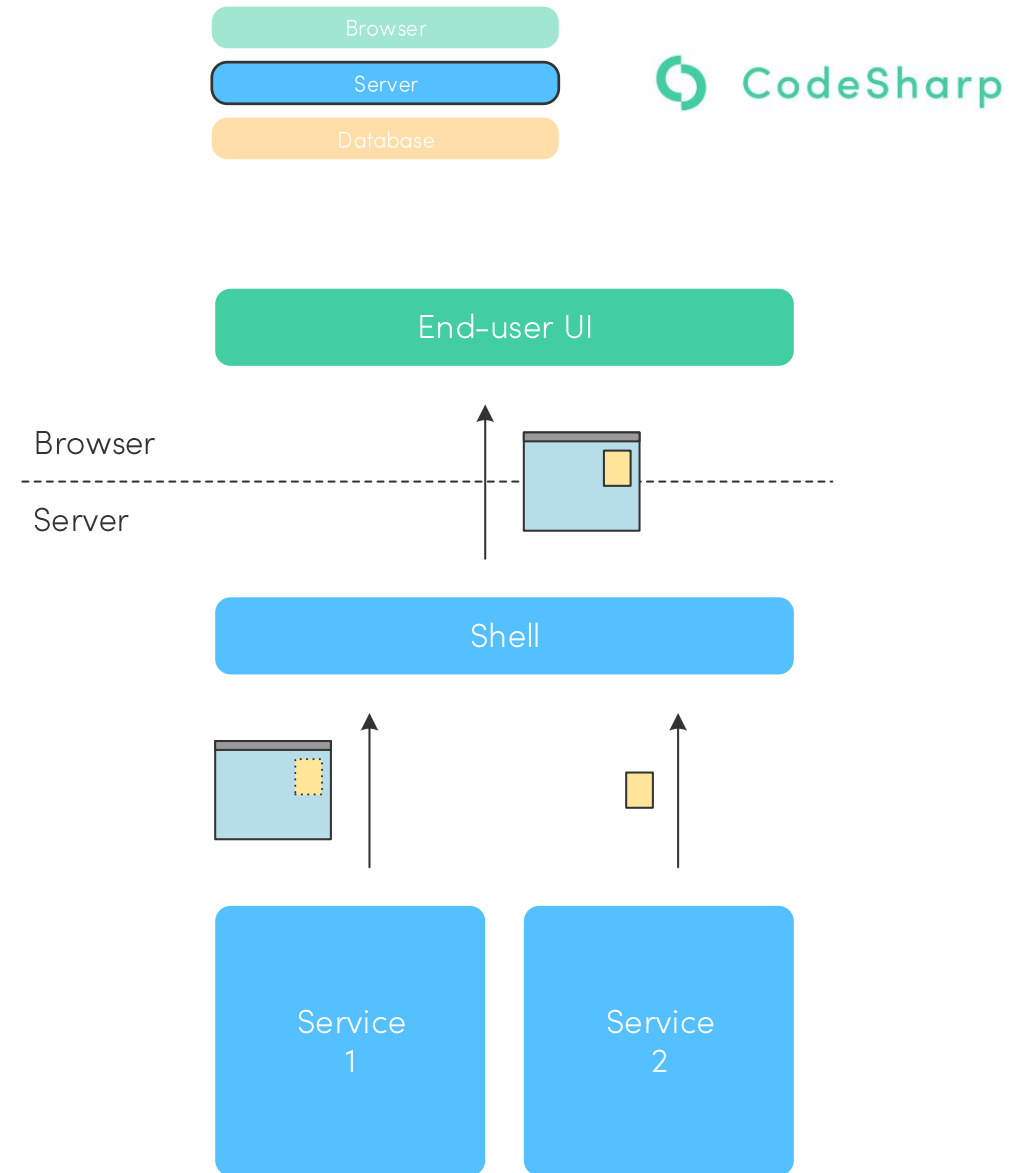
- In application integration
- Build-time integration
- ESI and SSI



## Server composition

# In application

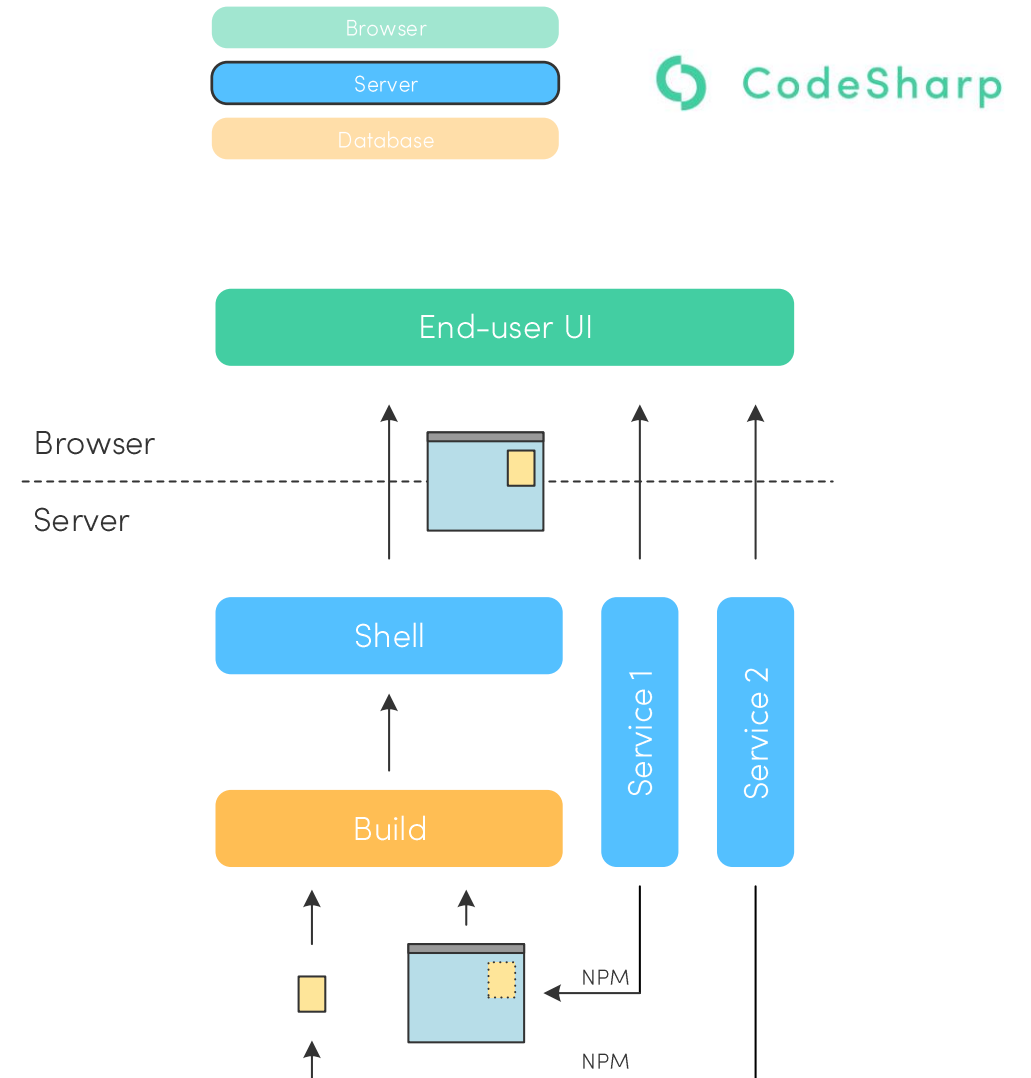
- For example ASP.NET MVC
  - Model
    - Collect data in dynamic view model
    - Subscribe route interceptors
  - View
    - Master pages, partial views, view components
  - *Controller*
    - Not a thing, just route handling
- Deliver parts with NuGet



## Server composition

# Build-time

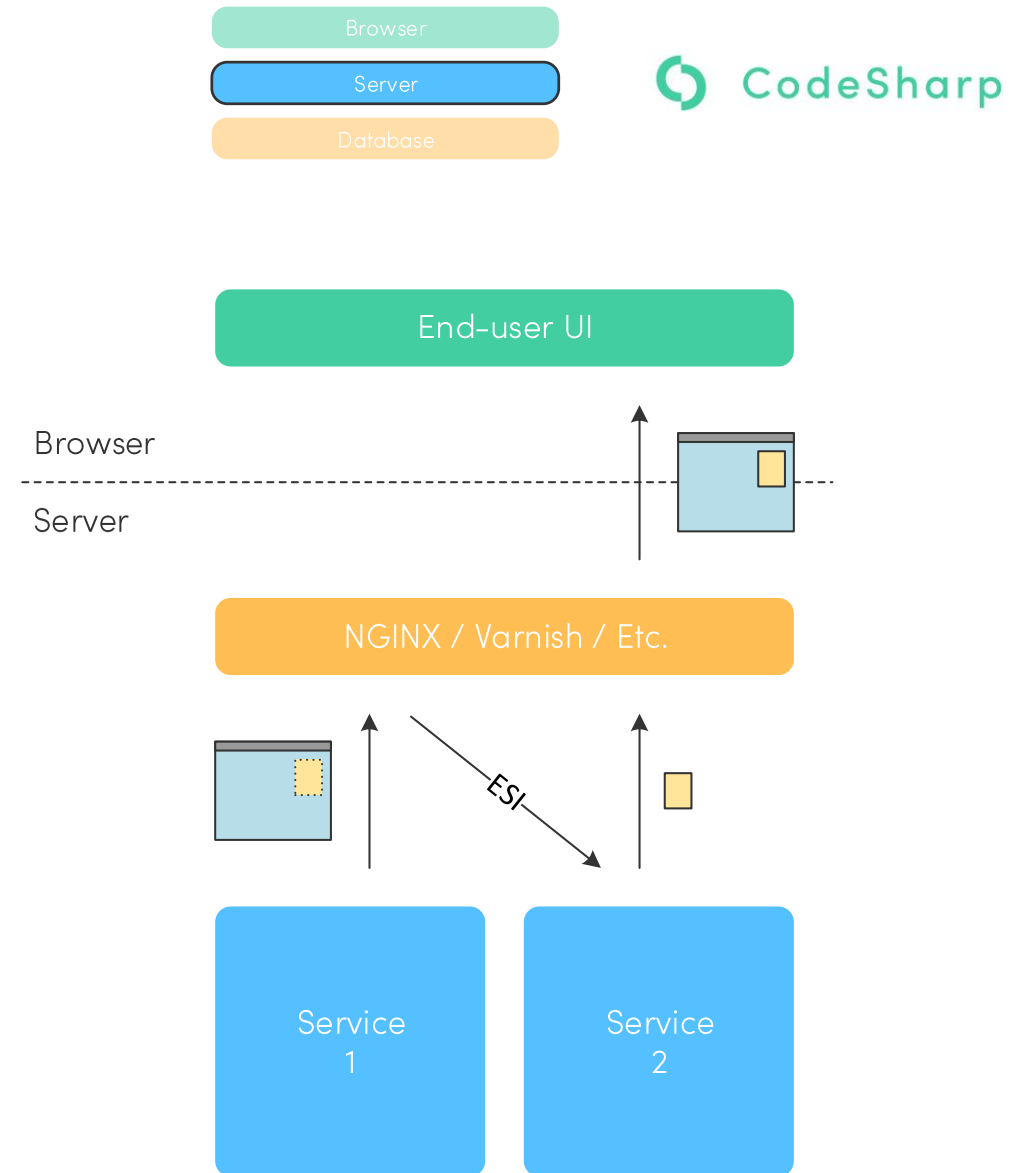
- NPM package per application
- Build shell for every change



## Server composition

# ESI and SSI

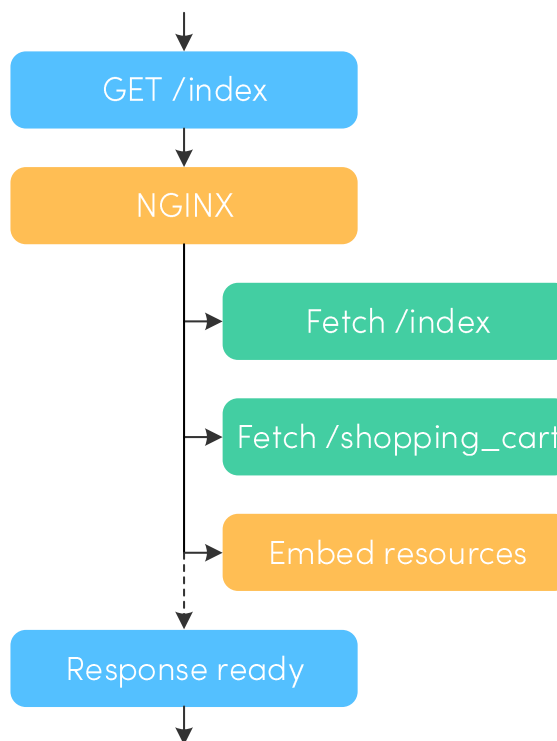
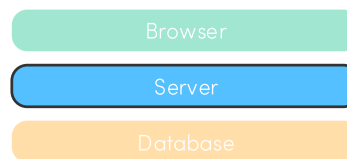
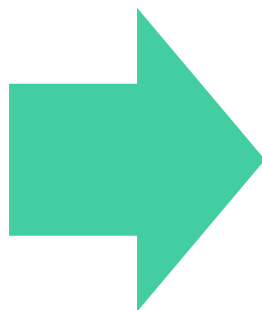
- Edge-side include (ESI)
  - Caches, reverse proxies, load balancers
  - Varnish, NGINX, etc.
- Server-side include (SSI)
  - Application, web server
  - Apache2, IIS, etc.



Server composition

# ESI and SSI

```
1 <html>
2   <head>
3     <title>Ecommerce site</title>
4   </head>
5   <body>
6     <div id="content">
7       <div id="header">
8         
9         <esi:include src="/shopping_cart" />
10      </div>
11      <div id="items">
12        <span id="item_1">...</span>
13        <span id="item_2">...</span>
14        <span id="item_n">...</span>
15      </div>
16    </div>
17  </body>
18 </html>
```



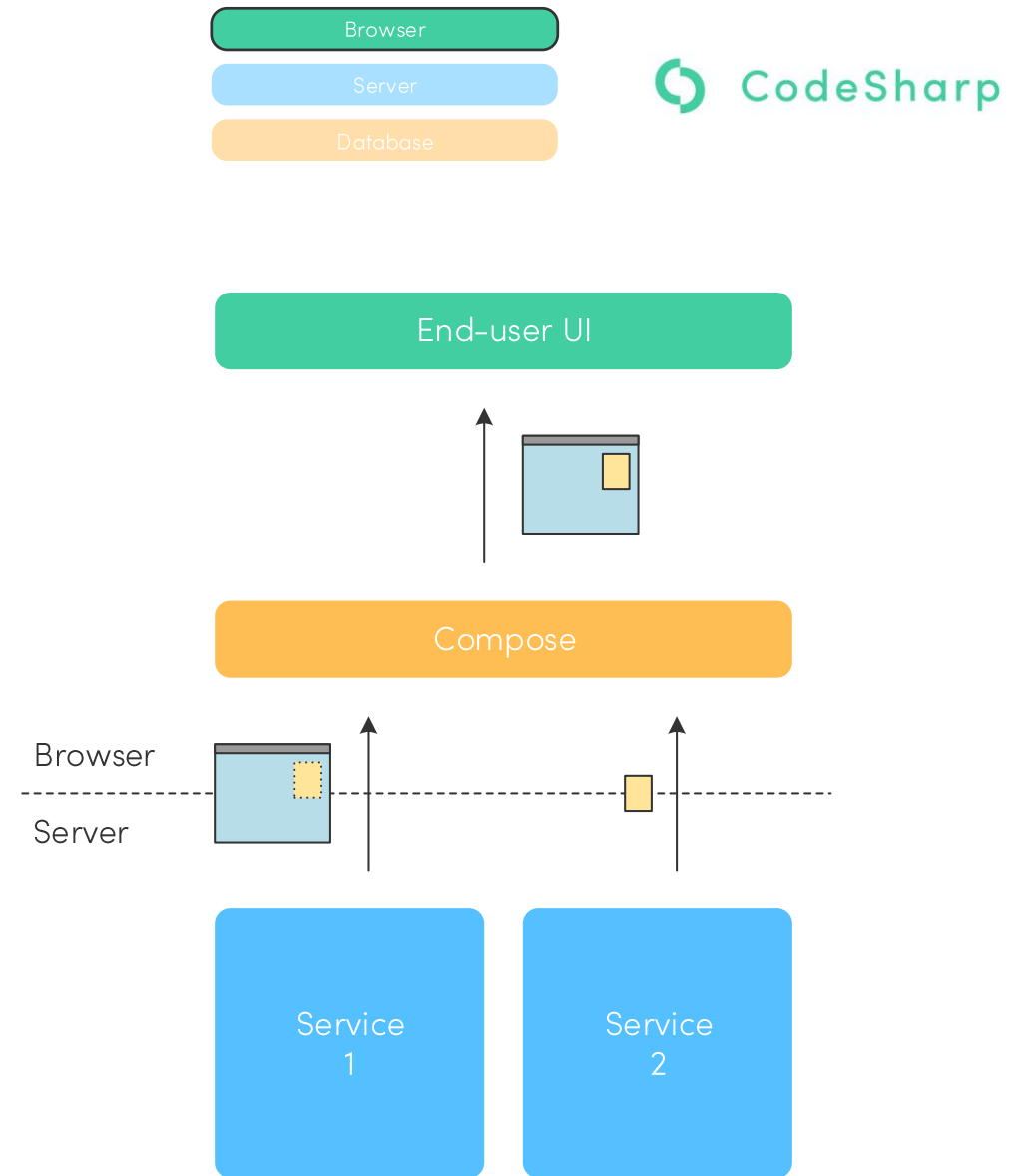
# Browser composition



Browser composition

# Browser composition

- Integration via iframes
- Integration via JavaScript
- Integration via web components

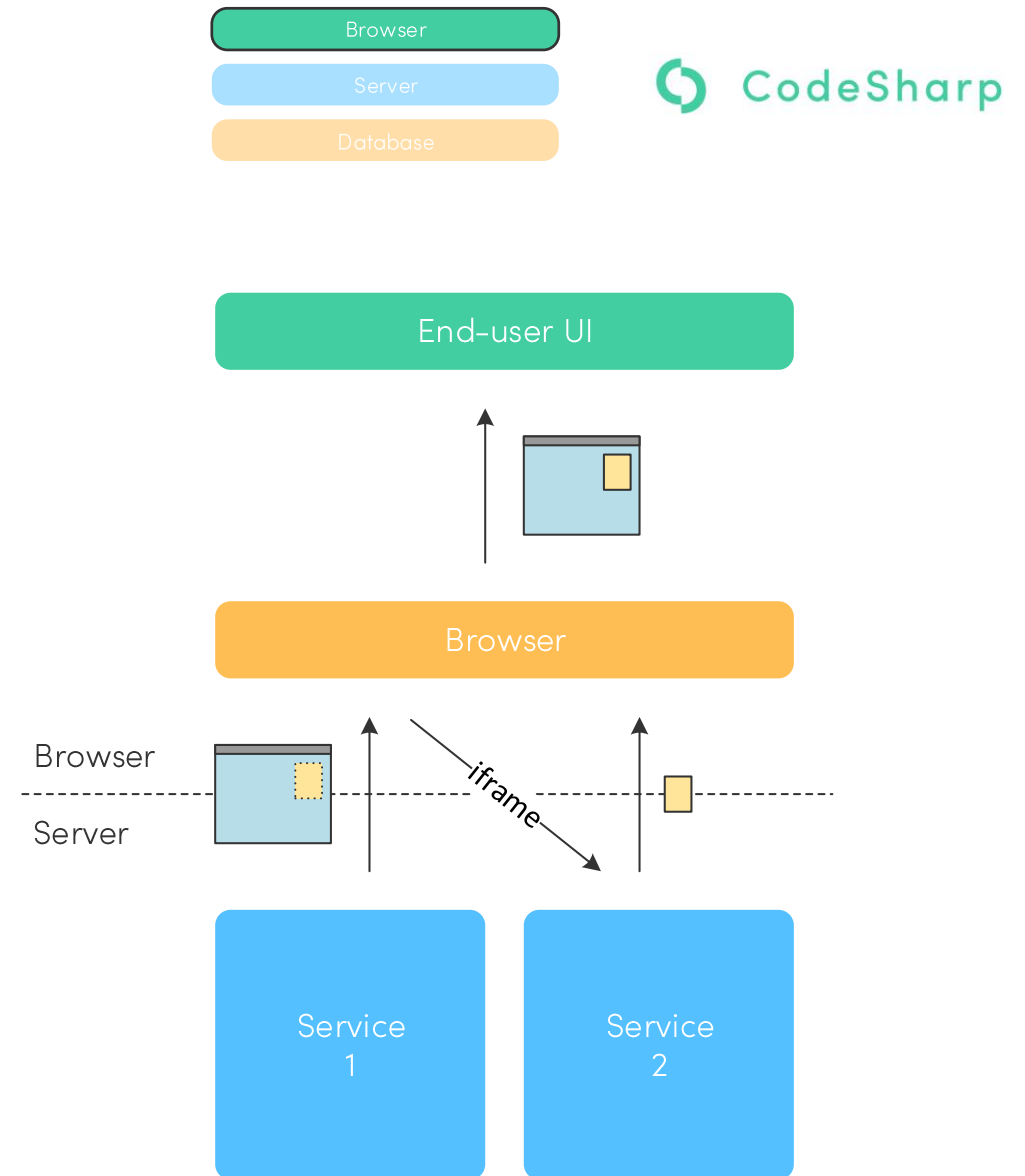




## Browser composition

# Iframes

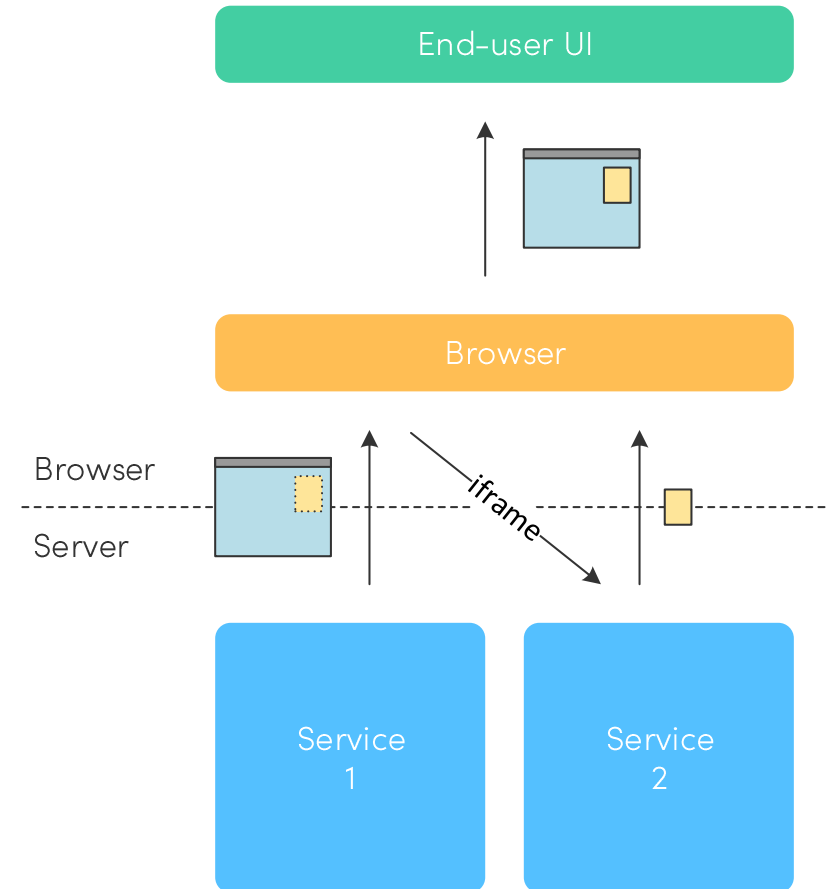
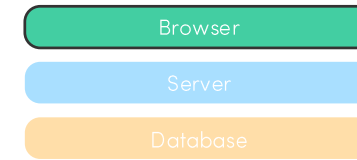
- Proven technology
- Good isolation
- Communication with events



## Browser composition

# Iframes

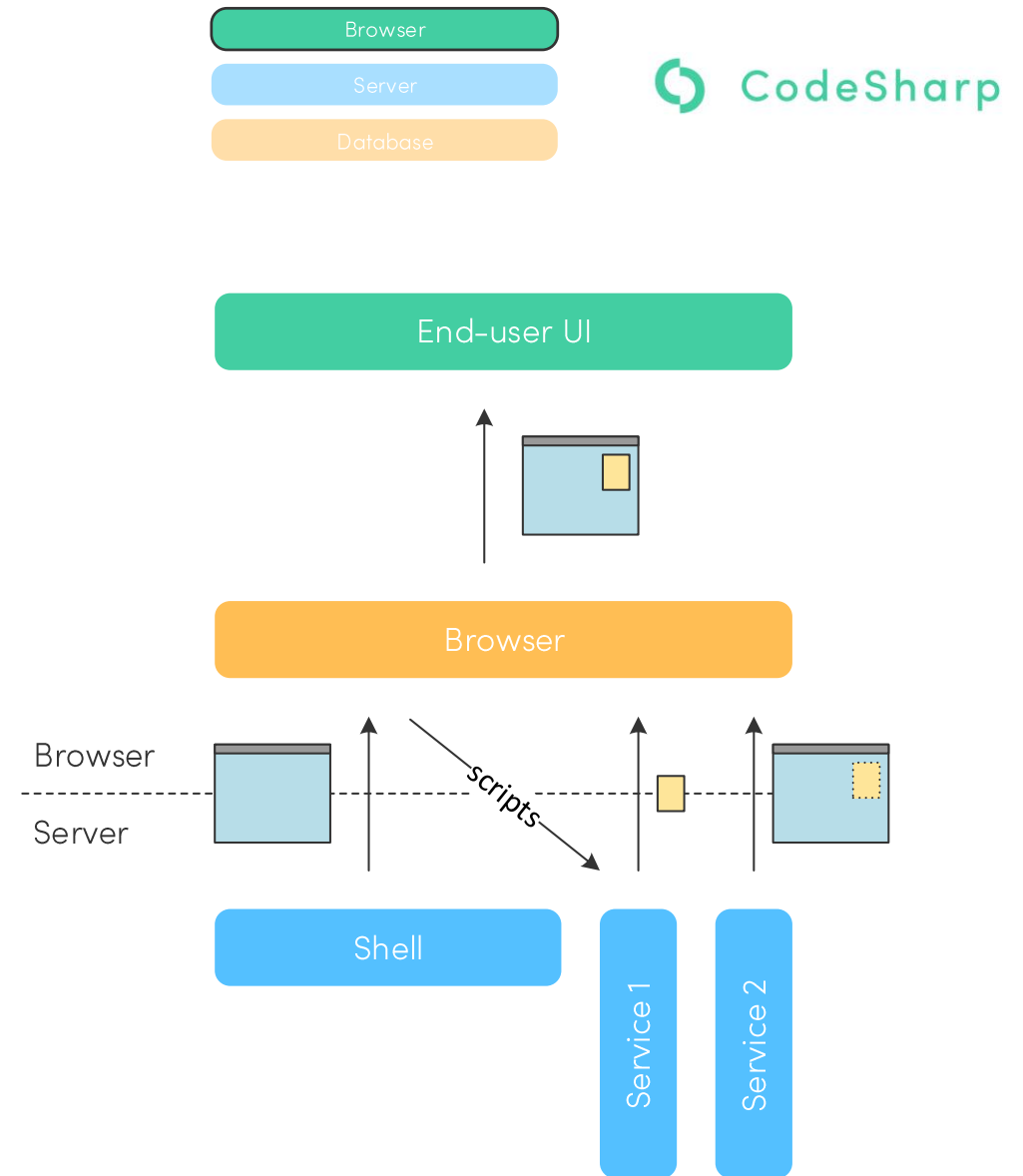
```
1 <html>
2   <head>
3     <title>Micro frontends</title>
4   </head>
5   <body>
6     <iframe id="app-container"></iframe>
7
8     <script type="text/javascript">
9       const routes = {
10         '/': 'https://store.com/index.html',
11         '/orders': 'https://orders.store.com/index.html',
12         '/sales': 'https://sales.store.com/index.html',
13         '/shipping': 'https://shipping.store.com/index.html',
14       };
15
16       const iframe = document.getElementById('app-container');
17       iframe.src = routes[window.location.pathname];
18     </script>
19   </body>
20 </html>
```



Browser composition

# JavaScript

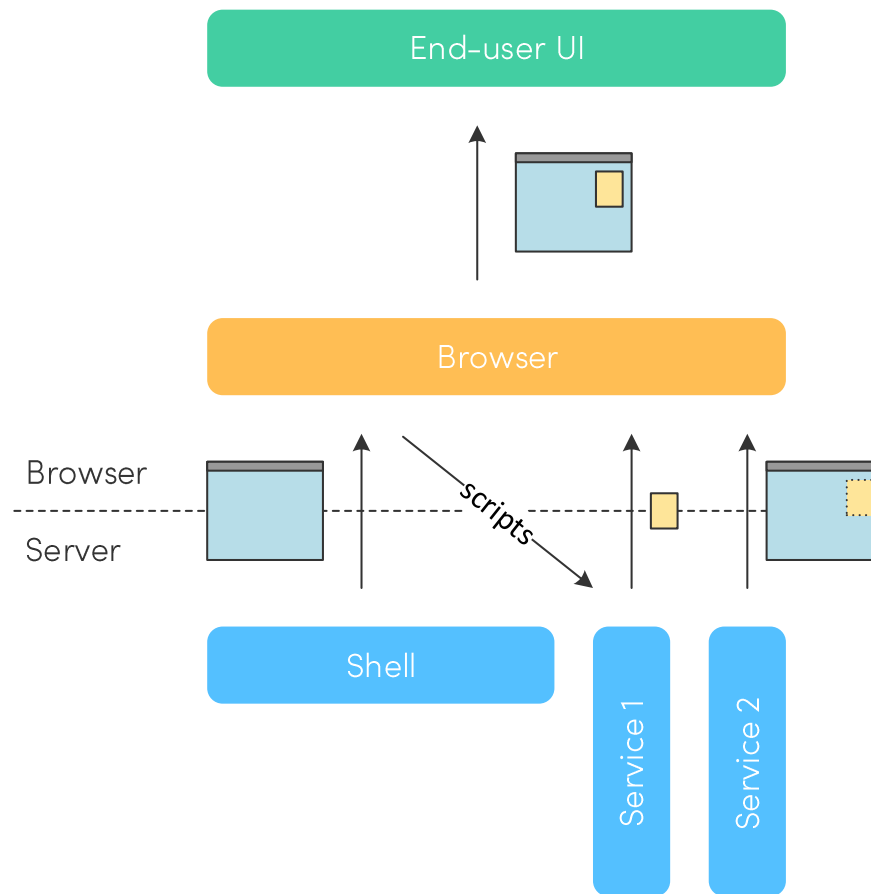
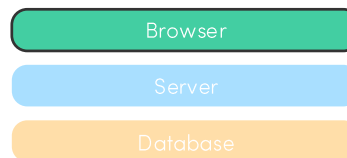
- Load scripts from microservices
- Shell application for base functionality
- Very flexible
- Most popular



## Browser composition

# JavaScript

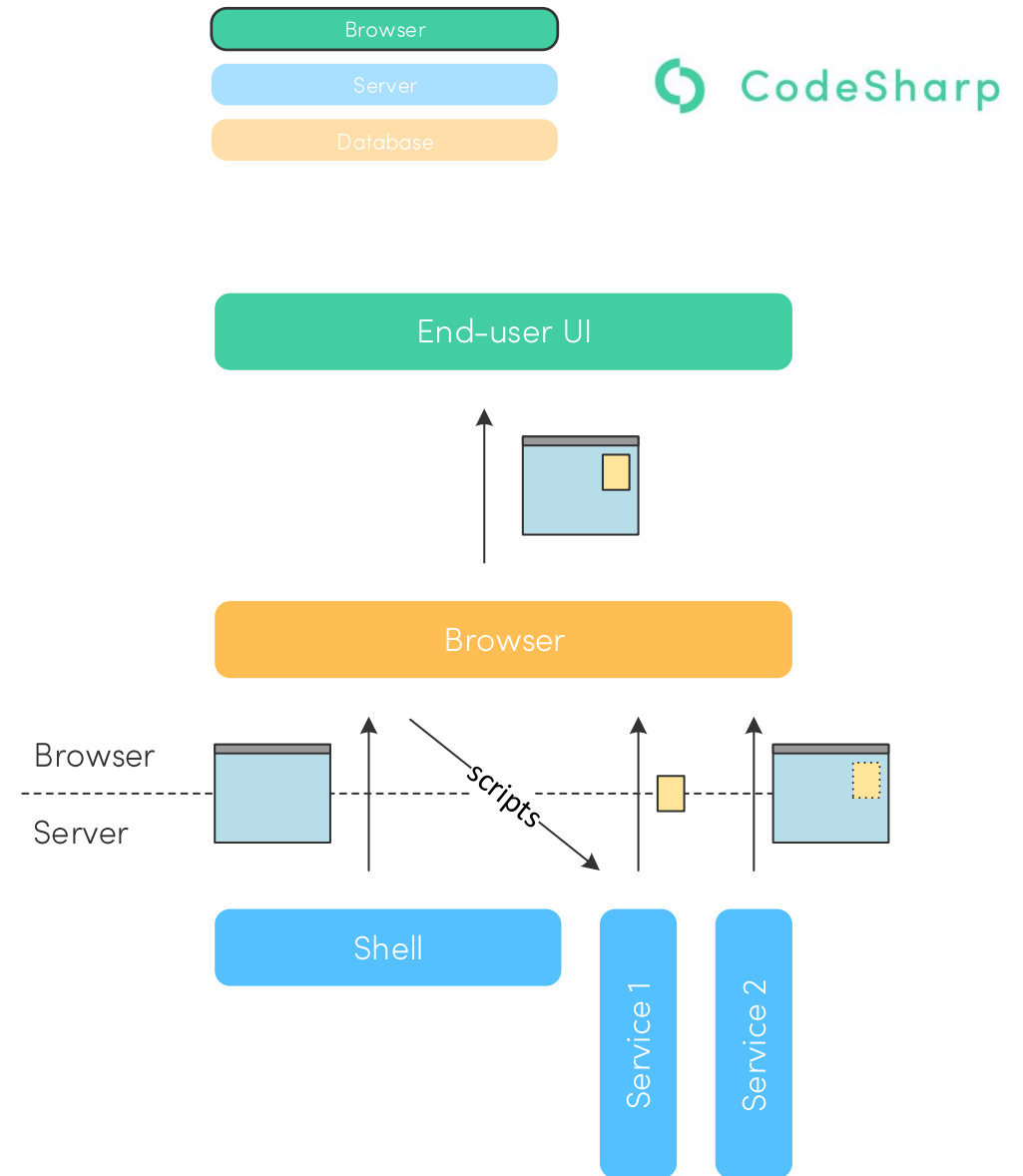
```
1 <html>
2   <head>
3     <title>Micro frontends</title>
4   </head>
5   <body>
6     <!-- Scripts register entry-points to 'window' -->
7     <script src="https://store.com/bundle.js"></script>
8     <script src="https://orders.store.com/bundle.js"></script>
9     <script src="https://sales.store.com/bundle.js"></script>
10    <script src="https://shipping.store.com/bundle.js"></script>
11
12    <div id="app-container"></div>
13
14    <script type="text/javascript">
15      const routes = {
16        '/':      window.renderStore,
17        '/orders': window.renderOrders,
18        '/sales':  window.renderSales,
19        '/shipping': window.renderShipping,
20      };
21
22      const renderFunction = routes[window.location.pathname];
23      renderFunction('app-container');
24    </script>
25  </body>
26 </html>
```



Browser composition

# Web components

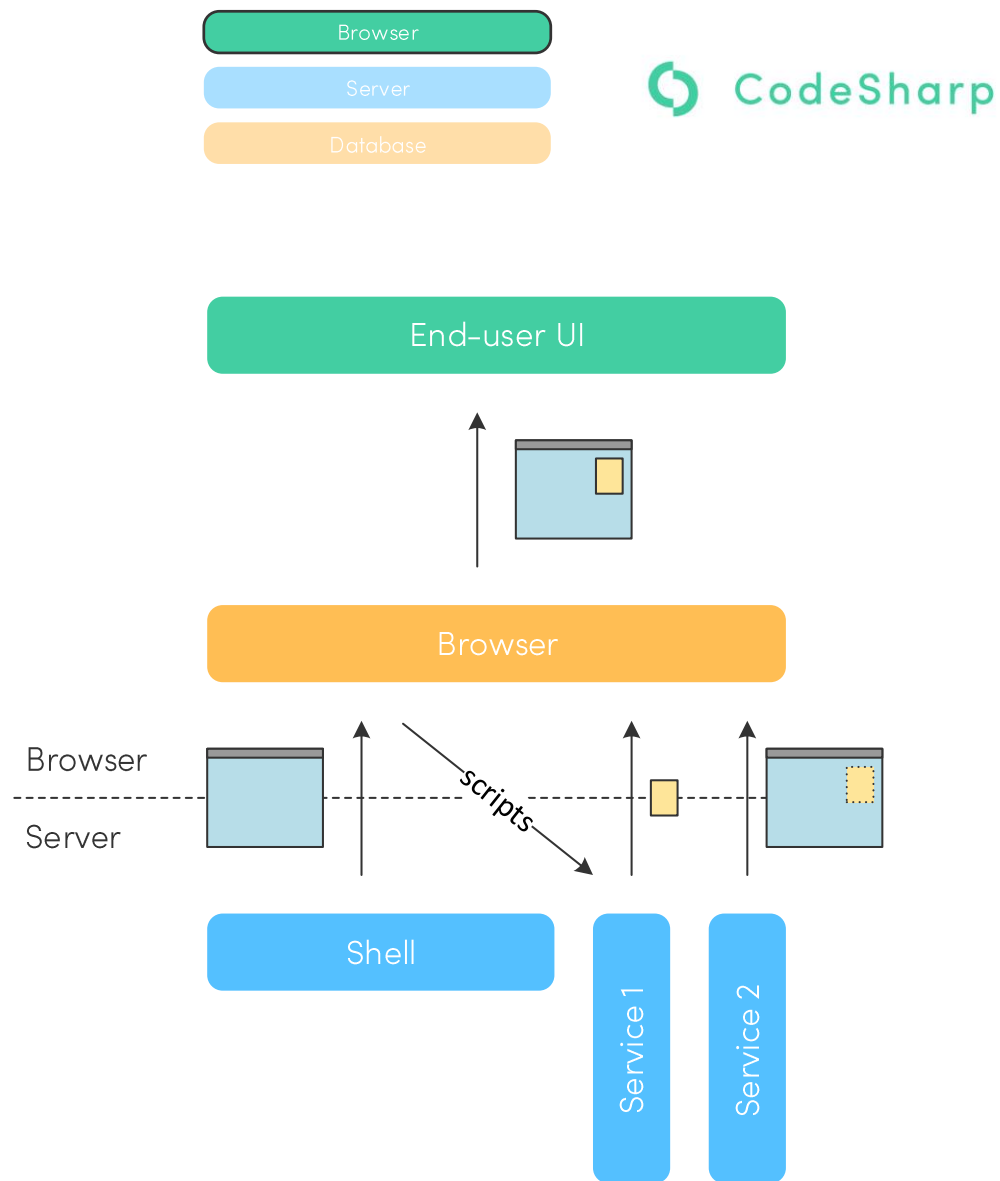
- Same as JavaScript
- Web component specification
- Use of browser capabilities



Browser composition

# Web components

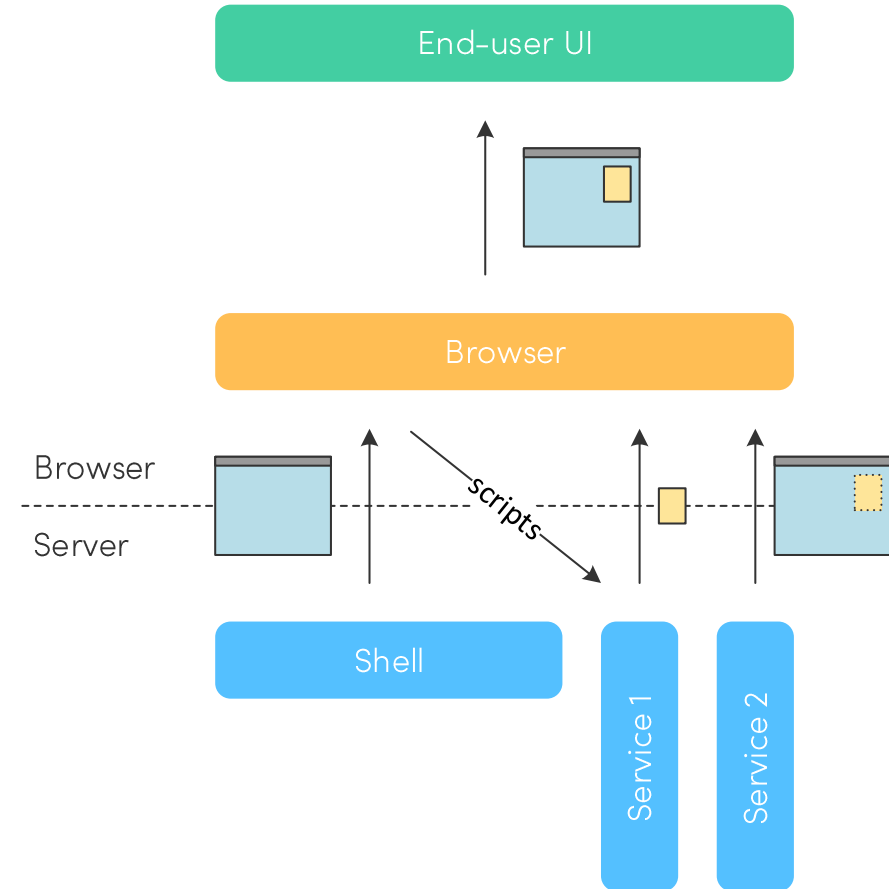
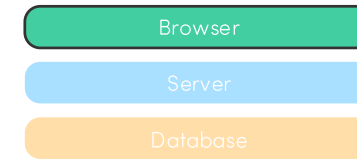
```
1 <html>
2 <head>
3   <title>Web components</title>
4
5   <script type="text/javascript">
6     class StoreButton extends HTMLElement {
7       constructor() { super(); } // When an element is created
8
9       connectedCallback() {} // When an element is inserted to the DOM
10      disconnectedCallback() {} // When an element is removed from the DOM
11
12      attributeChangedCallback(name, oldVal, newVal) {} // When an observed attribute changes
13      adoptedCallback() {} // When the element is moved to another document (iframes)
14
15      static get observedAttributes() { return ["color"]; } // Attributes to observe
16    }
17
18    window.customElements.define("store-button", StoreButton);
19  </script>
20 </head>
21 <body>
22   <store-button color="primary">Custom button</store-button>
23 </body>
24 </html>
```



Browser composition

# Web components

```
1 <html>
2   <head>
3     <title>Micro frontends</title>
4   </head>
5   <body>
6     <!-- Scripts define custom elements -->
7     <script src="https://store.com/bundle.js"></script>
8     <script src="https://orders.store.com/bundle.js"></script>
9     <script src="https://sales.store.com/bundle.js"></script>
10    <script src="https://shipping.store.com/bundle.js"></script>
11
12    <div id="app-container"></div>
13
14    <script type="text/javascript">
15      const routes = {
16        '/': 'store-home',
17        '/orders': 'store-orders',
18        '/sales': 'store-sales',
19        '/shipping': 'store-shipping',
20      };
21
22      const webComponentType = routes[window.location.pathname];
23      const root = document.getElementById('app-container');
24      const webComponent = document.createElement(webComponentType);
25      root.appendChild(webComponent);
26    </script>
27  </body>
28 </html>
```





# Hybrid composition

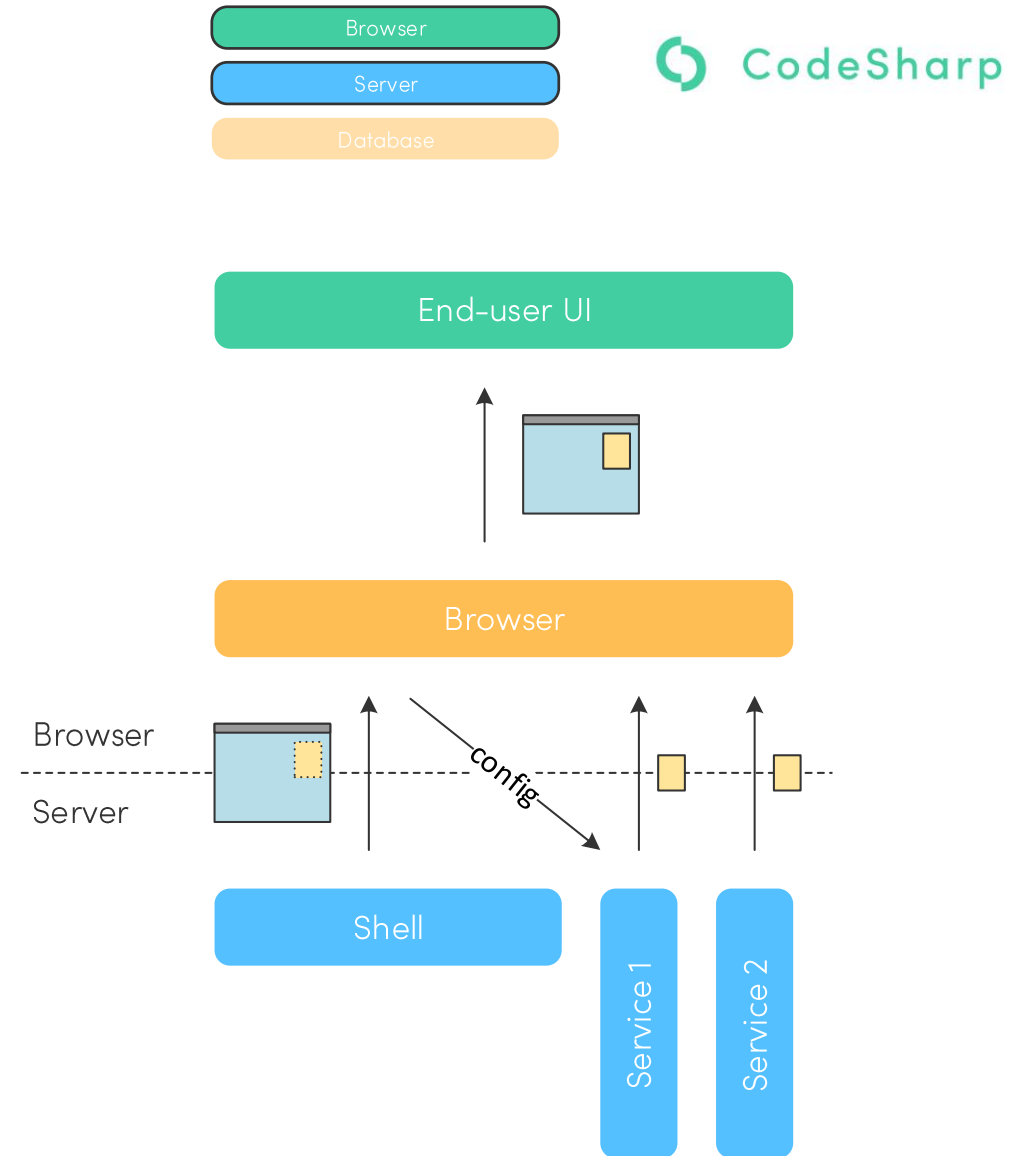




Breaking it apart

# Hybrid composition

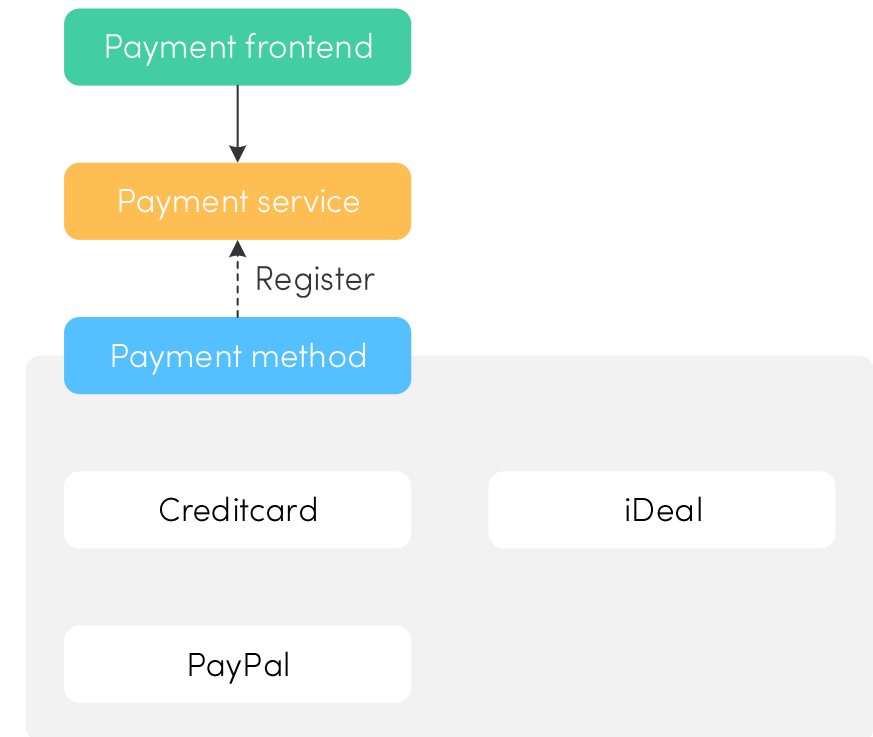
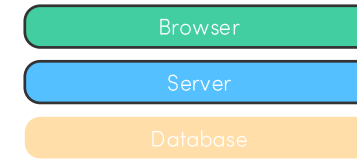
- Modelling composition
- Hypermedia



Breaking it apart

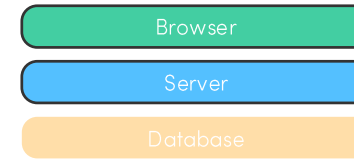
# Hybrid composition

- Modelling composition
  - Domain-driven design
  - Data-based composition



Breaking it apart

# Hybrid composition



- Hypermedia (protocol)

- HAL
- Siren
- Hydra

```
1  {
2      "@id": "https://store.com/products/2ca87e6b-21a1-4e9f-970b-93d8780d8c83",
3      "name": "Domain-driven design",
4      "author": "Eric Evans",
5      "price": 54.78,
6      "operations": [
7          {
8              "target": "https://store.com/products/2ca87e6b-21a1-4e9f-970b-93d8780d8c83",
9              "@type": "UpdateAction",
10             "expects": {
11                 "method": "PUT",
12                 "@type": "admin",
13                 "required": [ "name", "author", "price" ]
14             }
15         }
16     ]
17 }
```

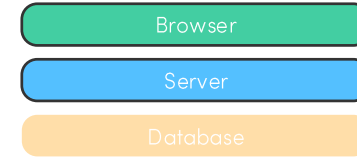
# Demo – Browser composition



Going to production

# Other things to consider

- Styling
- Communication
- State management
- Shared component libraries
- Testing
- Development environment



 CodeSharp



Questions?







**Jouke Mandemaker**

[jouke.mandemaker@codesharp.nl](mailto:jouke.mandemaker@codesharp.nl)

THANK YOU FOR YOUR TIME

**See you next time!**

**Code Sharp B.V.**

<https://codesharp.nl>

[info@codesharp.nl](mailto:info@codesharp.nl)