Single Page Applications without REST API

Simon Martinelli
Twitter: @simas_ch

About me

Java

1972 | 1995 | 2000 | 2007 | 2009 | 2011 | 2012 | 2013 | 2021

JAVA USER GROUP CH

COBOL

SBB CFF FFS

Bern University of Applied Sciences

JSR-352
Java Batch

JSR-354
Money/Currency

vaadin}>

hilla

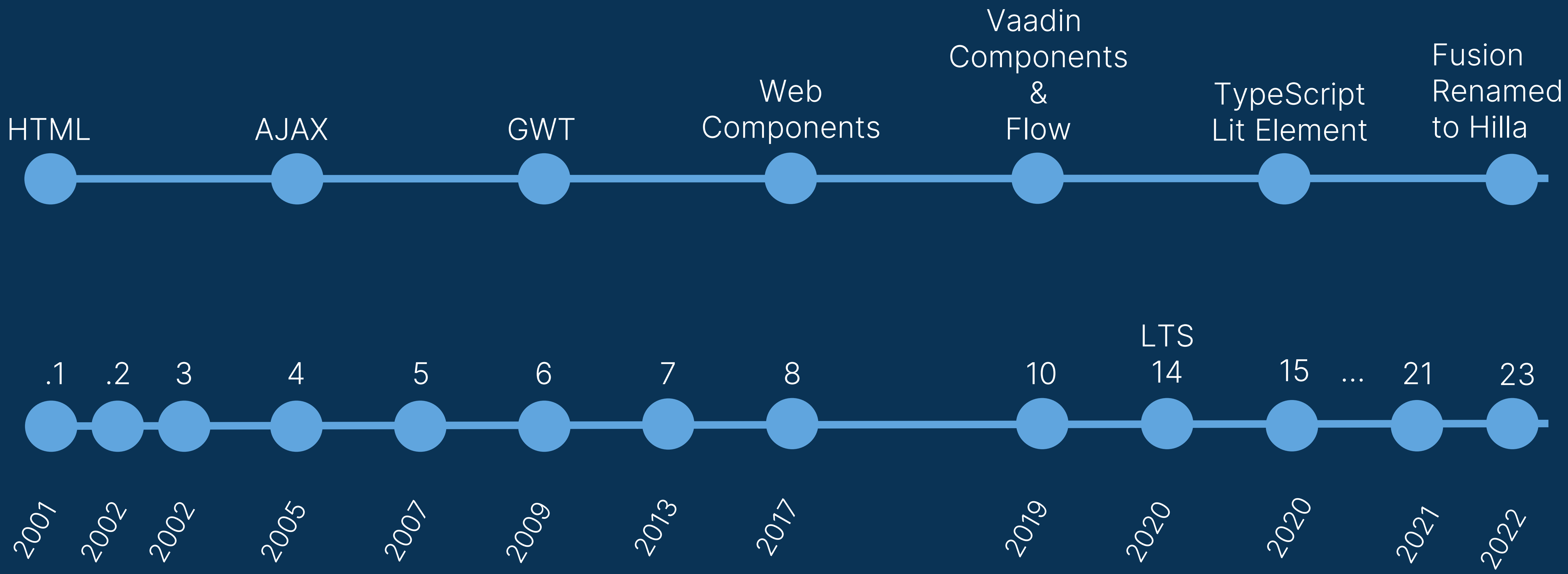
Why Vaadin?

- **A great fit for data-centric business applications**
- Has a **rich component model**
- Brings everything you need for this type of application
  - **Grids with paging, sorting and filtering**
  - **Forms with validation and conversion**

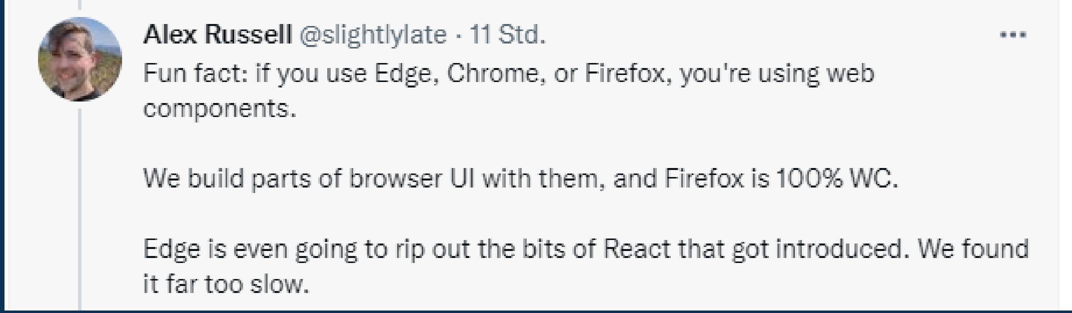
Vaadin as we know it

```java
@Route
public class HelloView extends VerticalLayout {

    public HelloView() {
        TextField textField = new TextField("Your Name");
        Label label = new Label();

        Button button = new Button("Greet",
                      e -> label.setText("Hello, " + textField.getValue()));

        add(textField, label, button);
    }

}
```

History


HTML
AJAX
GWT
Web Components
Vaadin Components & Flow
TypeScript Lit Element
Fusion Renamed to Hilla
.1
.2
3
4
5
6
7
8
10
LTS 14
15
...
21
23
2001
2002
2002
2005
2007
2009
2013
2017
2019
2020
2020
2021
2022

Web Components?



Alex Russell @slightlylate · 11 Std.

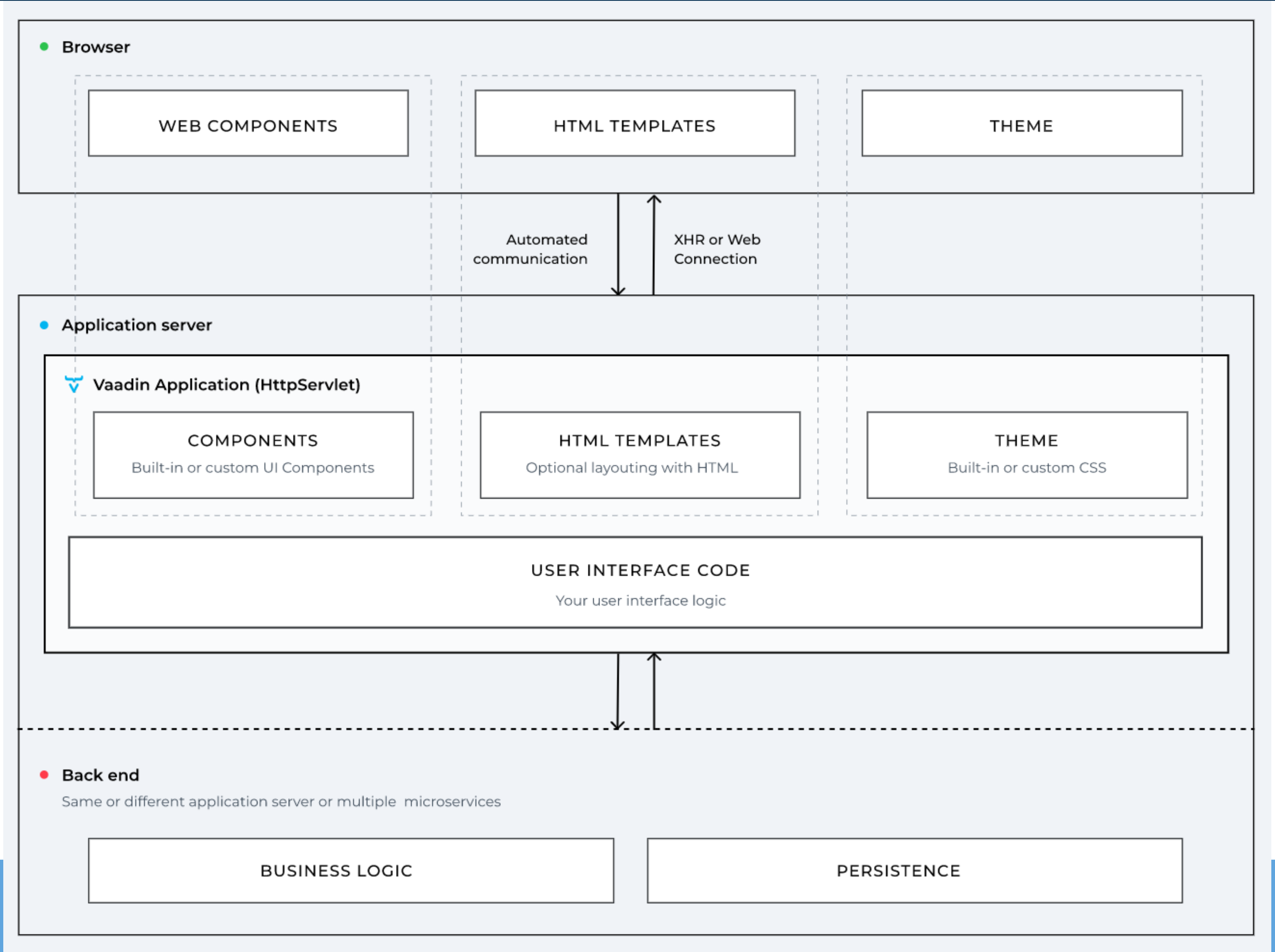
Fun fact: if you use Edge, Chrome, or Firefox, you're using web components.

We build parts of browser UI with them, and Firefox is 100% WC.

Edge is even going to rip out the bits of React that got introduced. We found it far too slow.

- Web Components allowing you to create reusable custom elements and utilize them in your web apps.
  - Custom elements: A set of JavaScript APIs
  - Shadow DOM: To keep an element's features private, so they can be scripted and styled without the fear of collision with other parts of the document.
  - HTML templates: These can then be reused multiple times as the basis of a custom element's structure.

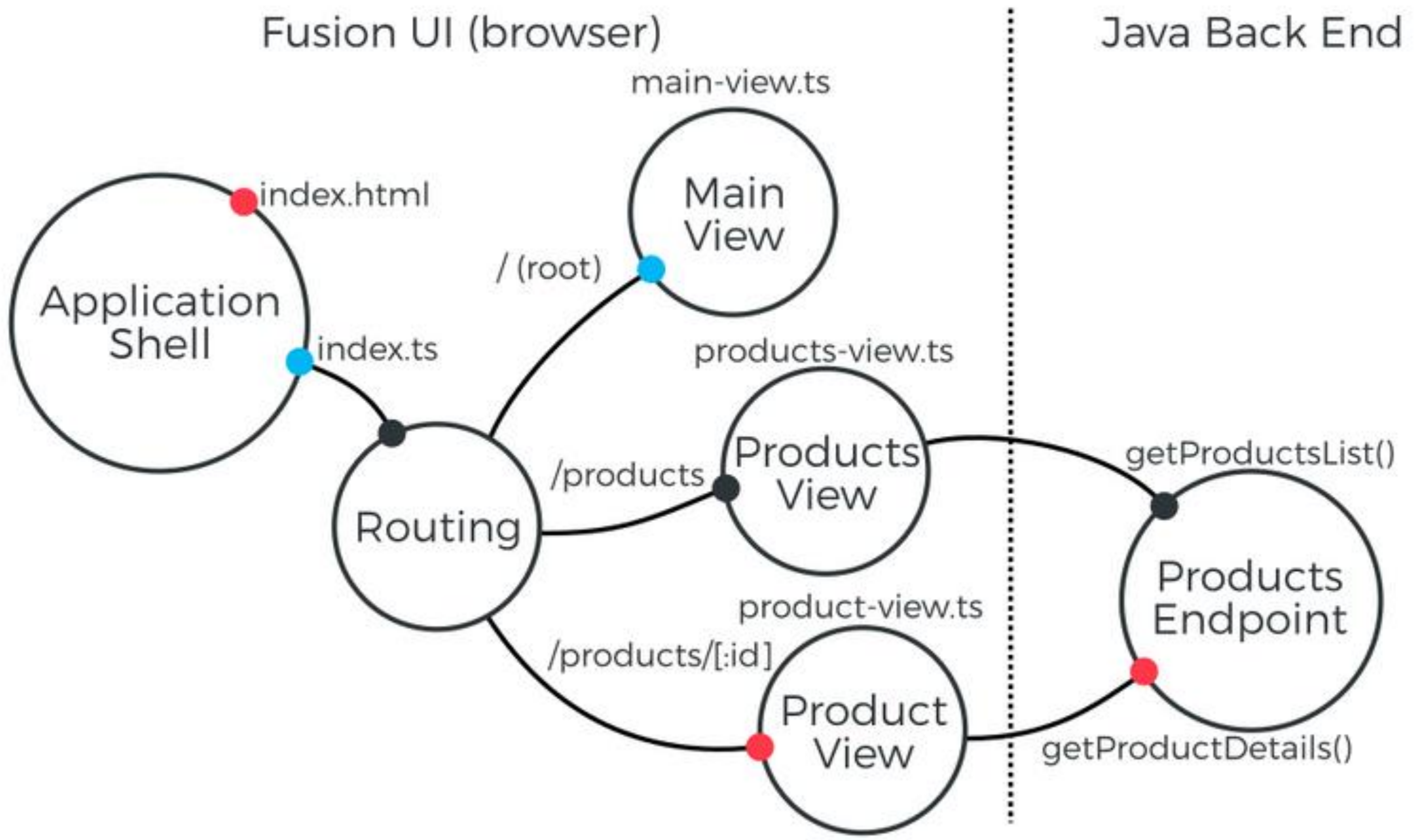
Vaadin Architecture



Vaadin

- Type-safe **Java-UI-Component-API** on the server side
- Uses **Web Components**
- **No REST API necessary**
  - Direct acces from UI code to services and repositories
- **Bi-directional Data Binding**
  - If the user interface changes on the client or the server, the changes are automatically applied to the other side

Hilla Architecture



Hilla

- **Integrates** a Spring Boot **Java backend** with a reactive **TypeScript frontend**
  - Build UIs from web components
  - Use a reactive programming model for updating the UI
  - Use routing to display views and resources
- **No REST-API necessary**
  - REST API and client code are generated
  - Manage security on the server-side
- **Fully stateless**
  - TypeScript views can be loaded without creating a server session

What is Lit?

- Lit is a **simple library for building fast, lightweight web components**
- At Lit's core is a boilerplate-killing component base class that provides **reactive** state, scoped styles, and a **declarative template system** that's tiny, fast and expressive

- https://lit.dev/

Frontend Example

```
@customElement('hello-world-view')
export class HelloWorldView extends View {
  name = '';

  render() {
    return html`
      <vaadin-text-field label="Your name" @value-changed=${this.nameChanged}>
      </vaadin-text-field>
      <vaadin-button @click=${this.sayHello}>Say hello</vaadin-button>
    `;
  }
  nameChanged(e: CustomEvent) {
    this.name = e.detail.value;
  }
  sayHello() {
    showNotification(`Hello ${this.name}`);
  }
}
```

Endpoints are secure by default

```
@Endpoint
public class MyEndpoint {

  @PermitAll
  public void permittedToAllMethod() {
    // Any authenticated user can access
  }

  @RolesAllowed("ROLE_ADMIN")
  public void permittedToRoleMethod() {
    // Only users with admin role can access
  }
}
```

Vaadin or Hilla?

- **Vaadin** for **full-stack development with Java**
- **Hilla** if you want **to avoid server state**

Let's see some code

https://hilla.dev

Conclusion

- With Hilla you **don't** have to **care about client-server communication**
- You can **concentrate on building the application** because the build just works
- You get a **single deployment** which is in many cases very convenient

Let's Keep in Touch

Blog	martinelli.ch
Twitter	@simas_ch

LinkedIn.com/in/simonmartinelli/

Thank you!