

FRONTENDOVÁ ODYSEA

Vývoj, frameworky a korporátní džungle

Michal Kozubík

Development Expert

SAP Signavio, SAP Labs Brno

michal.kozubik@sap.com



AGENDA

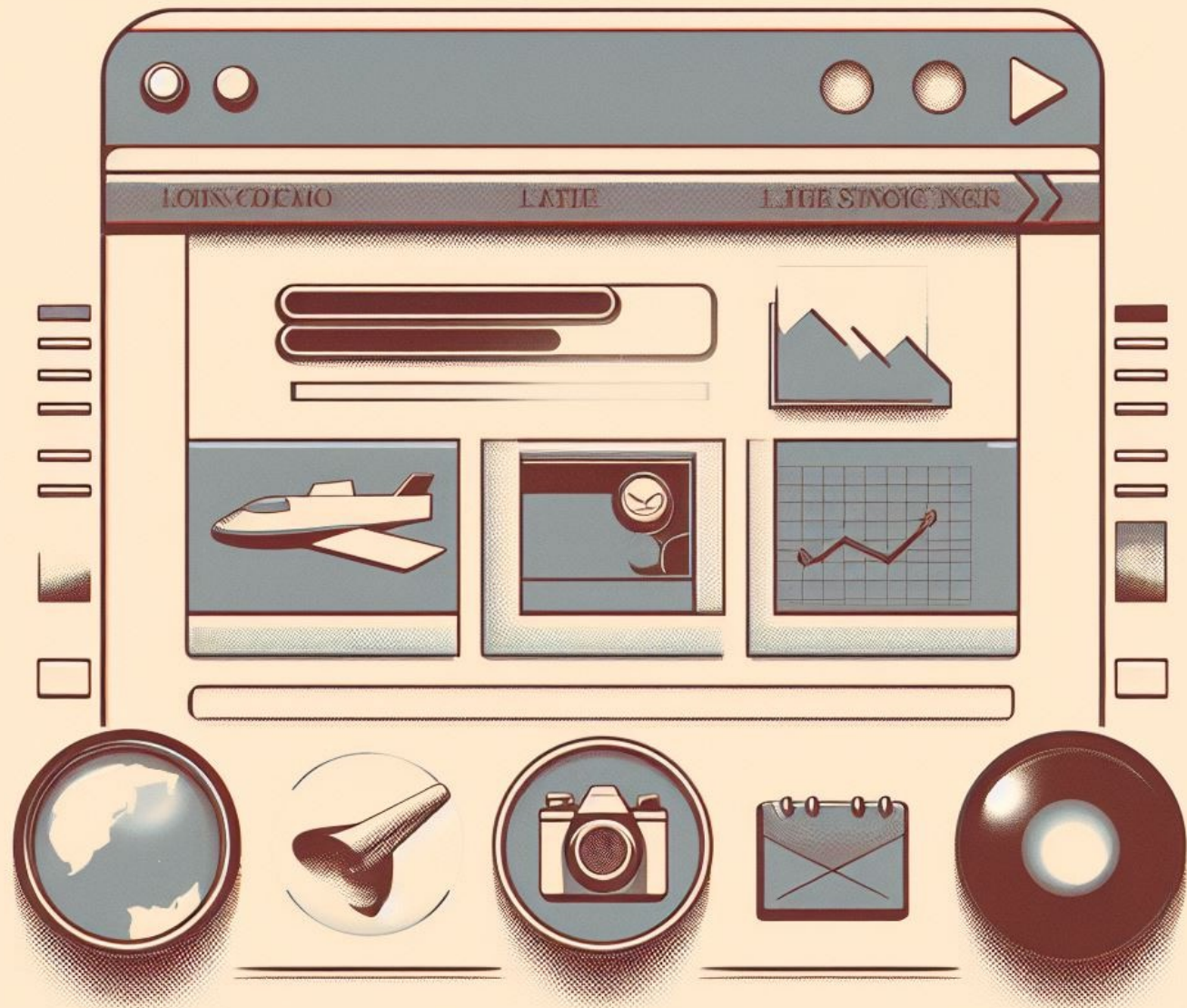
- Úvod
- Evoluce frontendu a webových aplikací
- (Mrazivá) tajemství Frameworků
- Vývoj webových aplikací pro velké korporace

WHY ARE SKELETONS SO CALM?

WHY ARE SKELETONS SO CALM?

BECAUSE NOTHING GETS UNDER THEIR SKIN.

EVOLUCE FRONTENDU



EVOLUCE FRONTENDU - ZAČÁTKY

- HTML (1993)

```
<html>
<head>
  <title>Vítejte na mé první webové stránce</title>
</head>
<body>
  <h1>Ahoj světe!</h1>
  <p>Toto je moje první webová stránka vytvořená pomocí HTML 1.0.</p>
  <p>Ještě mnoho vylepšení a novinek přijde!</p>
</body>
</html>
```

PHP (1995)

```
<?php
// Simulace načítání dat z API
$response = file_get_contents('https://api.example.com/weather');
$data = json_decode($response, true);
$temperature = $data['temperature'];
?>

<div>
    <h1>Aktuální teplota: <?php echo $temperature; ?>°C</h1>
</div>
```

EVOLUCE FRONTENDU - ASYNC

- JavaScript (1995, ECMA 1997)
- iframe (1996, Microsoft), XMLHttpRequest (1999, Microsoft)
- Ajax (2005, Google) - Asynchronous Javascript and XML
- Přejít k dynamickým stránkám
- Zvýšená interaktivita, klient nabírá hmotnost (svaly)

```
var xhr = new XMLHttpRequest();
xhr.open("GET", "server-data.json", true);
xhr.onreadystatechange = function () {
    if (xhr.readyState == 4 && xhr.status == 200) {
        var data = JSON.parse(xhr.responseText);
        // Manipulace s daty
    }
};
xhr.send();
```


JQUERY (2006)

```
<div id="weather-container">
  <h1>Aktuální teplota: <span id="temperature"></span>°C</h1>
</div>

<script>
$(document).ready(function () {
  // Simulace načítání dat z API
  $.getJSON('https://api.example.com/weather', function (data) {
    // Aktualizace teploty na stránce
    $('#temperature').text(data.temperature);
  });
});
</script>
```

EVOLUCE FRONTENDU – MODERNÍ ÉRA

- TypeScript, React, Angular...
- Konec webových prezentací, začátek webových aplikací
- Vysoká interaktivita
- Obézní klient

REACT

```
import React, { useState, useEffect } from 'react';

function WeatherApp() {
  const [temperature, setTemperature] = useState(null);

  useEffect(() => {
    // Simulace načítání dat z API
    fetch('https://api.example.com/weather')
      .then(response => response.json())
      .then(data => setTemperature(data.temperature));
  }, []);

  return (
    <div>
      <h1>Aktuální teplota: {temperature}°C</h1>
    </div>
  );
}

export default WeatherApp;
```

EVOLUCE FRONTENDU – POST-MODERNÍ ÉRA

- Metaframeworks
- Next, Nuxt, SvelteKit, Astro
- Zjištění, že obezita není OK
- Přesun funkcionality zpět na server
- Menší datová náročnost

NEXT.JS

```
const getTemperature = async () => {
  const response = await fetch("https://api.example.com/weather");
  const data = await response.json();

  return data.temperature;
};

const WeatherApp = async () => {
  const temperature = await getTemperature();

  return (
    <div>
      <h1>Aktuální teplota: {temperature}°C</h1>
    </div>
  );
};

export default WeatherApp;
```

EVOLUCE FRONTENDU – UZAVŘENÍ CYKLU

```
<?php
// Simulace načítání dat z API
$apiResponse = file_get_contents('https://api.example.com/weather');
$data = json_decode($apiResponse, true);
$temperature = $data['temperature'];
?>

<div>
  <h1>Aktuální teplota: <?php echo $temperature; ?>°C</h1>
</div>
```

```
const getTemperature = async () => {
  const response = await fetch("https://api.example.com/weather");
  const data = await response.json();

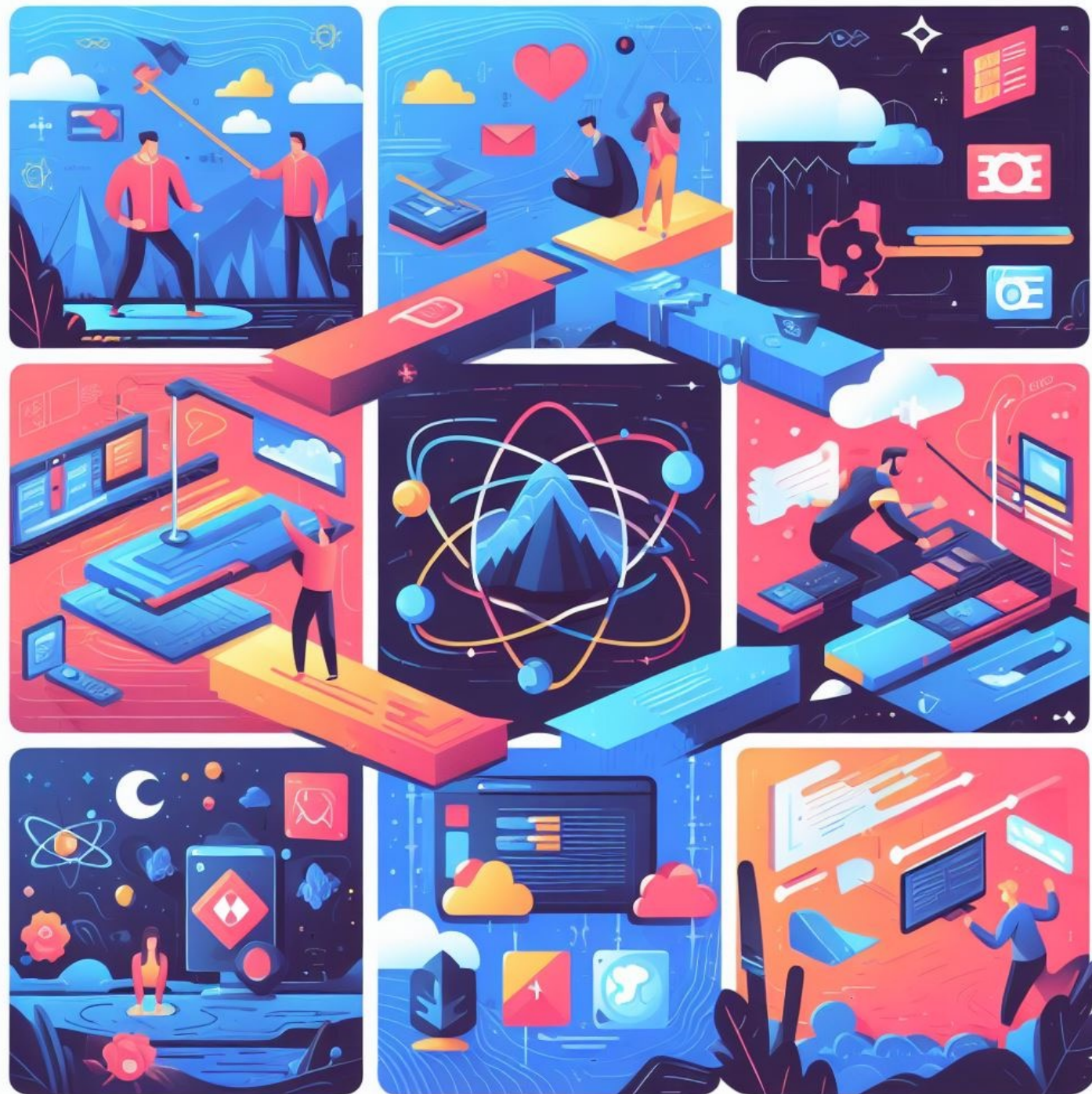
  return data.temperature;
};

const WeatherApp = async () => {
  const temperature = await getTemperature();

  return (
    <div>
      <h1>Aktuální teplota: {temperature}°C</h1>
    </div>
  );
};

export default WeatherApp;
```

TAJEMSTVÍ FRAMEWORKŮ



FRAMEWORK NEBO METAFRAMEWORK?

- Framework
 - Soubor abstrakcí, knihoven, nástrojů a pravidel, které usnadňují a standardizují proces vývoje software
- Metaframework
 - Poskytuje abstrakce a nástroje pro správu stavu aplikace, routování, optimalizaci výkonu a další.
 - Umožňuje snadněji vytvářet sofistikované webové aplikace a současně abstrahuje některé technické detaily a složitosti.

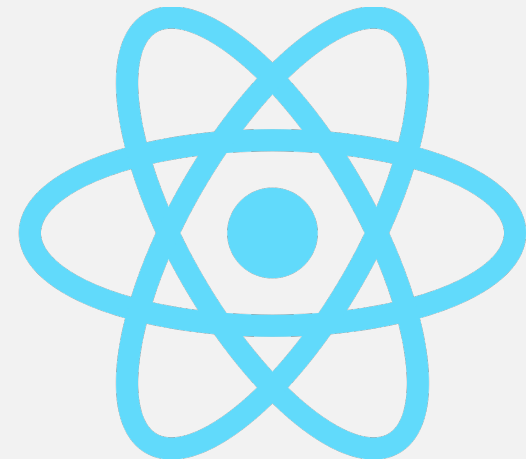
FRAMEWORK - REACT

- “Meta framework”



FRAMEWORK - REACT

- 2013
- Není to framework...
- Virtual DOM a komponentová architektura
- JSX formát
- Interaktivní uživatelské rozhraní
- Velká síla
- Velká zodpovědnost vývojáře
- Metaframework: Next.js



FRAMEWORK REACT

```
import React, { useState, useEffect } from 'react';

function WeatherApp() {
  const [temperature, setTemperature] = useState(null);

  useEffect(() => {
    // Simulace načítání dat z API
    fetch('https://api.example.com/weather')
      .then(response => response.json())
      .then(data => setTemperature(data.temperature));
  }, []);

  return (
    <div>
      <h1>Aktuální teplota: {temperature}°C</h1>
    </div>
  );
}

export default WeatherApp;
```

FRAMEWORK - ANGULAR

- 2016 (AngularJS 2010)
- Kompletní nástroje pro vývoj, testování a nasazování aplikací
- TypeScript out of the box
- Typicky velké a komplexní projekty
- Metaframework: Analog



FRAMEWORK ANGULAR

```
import { Component, OnInit } from '@angular/core';
import { ApiService } from './api.service';

@Component({
  selector: 'app-weather',
  template: '<h1>Aktuální teplota: {{ temperature }}°C</h1>',
})
export class WeatherComponent implements OnInit {
  temperature: number | null = null;

  constructor(private apiService: ApiService) {}

  ngOnInit() {
    // Simulace načítání dat z API
    this.apiService.getWeather().subscribe(data => {
      this.temperature = data.temperature;
    });
  }
}
```

FRAMEWORK – VUE.JS

- 2014
- Jednoduchý a flexibilní
- Snadná integrace do existujících projektů
- Vlastní tooling – Vite
- v3 velmi odlišná od v2
- Metaframework: Nuxt



FRAMEWORK VUE.JS

```
<template>
  <div>
    <h1>Aktuální teplota: {{ temperature }}°C</h1>
  </div>
</template>

<script>
import { ref, onMounted } from 'vue';

export default {
  setup() {
    const temperature = ref(null);

    onMounted(() => {
      // Simulace načítání dat z API
      fetch('https://api.example.com/weather')
        .then(response => response.json())
        .then(data => temperature.value = data.temperature);
    });

    return { temperature };
  },
};
</script>
```

FRAMEWORK – SVELTE

- 2016
- Zaměření na výkon
- V podstatě se nejedná o framework, ale o kompilátor
- Minimalizace množství potřebného JavaScriptu
- Co nejjednodušší zápis kódu
- Metaframework: SvelteKit



FRAMEWORK SVELTE

```
<script>
  import { onMount } from 'svelte';

  let temperature = null;

  onMount(async () => {
    // Simulace načítání dat z API
    const response = await fetch('https://api.example.com/weather');
    const data = await response.json();
    temperature = data.temperature;
  });
</script>

<div>
  <h1>Aktuální teplota: {temperature}°C</h1>
</div>
```

FRAMEWORK
????



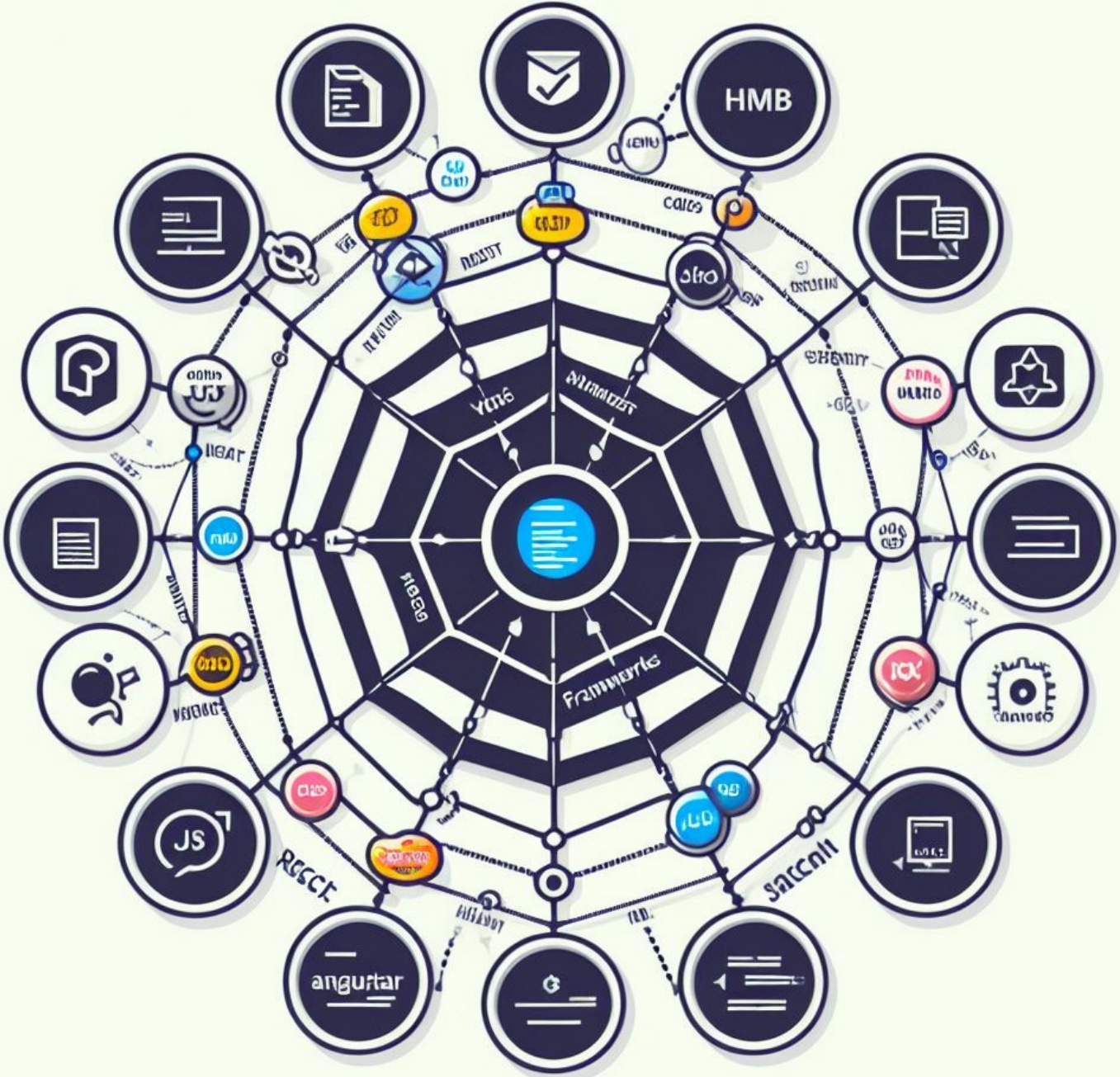
JavaScript Developers



„STYLOVÁ“ ODBOČKA

- CSS vs preprocessors
 - LESS
 - SASS
 - SCSS
- CSS frameworks
 - Component-based vs Utility-first
 - Bootstrap, Tailwind, Bulma
- Čisté CSS se vrací do módy

ABY TOHO NEBYLO MÁLO



FRONTENDOVÉ ZKRATKY

- JS, TS
- CSS, LESS, SASS, SCSS
- SSG, SSR, ISR, CSR
- SPA, MPA

FRONTENDOVÉ ZKRATKY - SSG

- SSG – Static Site Generation
 - Stránky jsou předgenerovány během fáze sestavení
 - Obsah je vygenerován do statických HTML souborů
- Výhody
 - Rychlé načítání stránek
 - Snadné cachování a distribuce obsahu (CDN)
 - Tenký klient
- Statické weby, blogy, marketingové stránky

FRONTENDOVÉ ZKRATKY - CSR

- CSR – Client Side Rendering
 - Veškeré vykreslování na klientovi
 - Server pošle prázdný kontejter, který se naplní u koncového uživatele
- Výhody
 - Rychlá odezva na akce
 - Možnost fungování offline
- Většina současných webobých aplikací

FRONTENDOVÉ ZKRATKY - RDJ

- RDJ – Random Dad Joke

WHAT'S THE ASTRONAUT'S FAVORITE PART OF A
COMPUTER?

WHAT'S THE ASTRONAUT'S FAVORITE PART OF A
COMPUTER?

THE SPACEBAR.

FRONTENDOVÉ ZKRATKY - SSR

- SSR – Server Side Rendering
 - Obsah je generován dynamicky na straně serveru při každém požadavku
 - Klient obdrží kompletní HTML document (+ JavaScript, když je třeba)
- Výhody
 - Dynamicky generovaný obsah na základě aktuálních dat
 - Lepší interaktivita a možnost přizpůsobení obsahu uživateli
 - SEO
 - Tenčí klient
- Aplikace s dynamickým obsahem, e-shopy

FRONTENDOVÉ ZKRATKY - ISR

- ISR – Incremental Static Regeneration
 - Kombinuje statickou generaci a dynamické generování obsahu
 - Klient obdrží kompletní HTML document (+ JavaScript, když je třeba)
- Výhody
 - Rychlost SSG + aktuálnost SSR
- Stránky s obsahem, který není nutné aktualizovat s každým požadavkem, ale mění se časem nebo podle konkrétních podmínek (blog atp.)

ISR

[Illegal Street Racing](#)

"yo, you [cats](#) down for some ISR?"

by [Rich B](#) August 31, 2006

FRONTENDOVÉ ZKRATKY - SPA

- SPA – Single Page Application
 - Jedna HTML stránka, dynamický obsah dle virtuální “routy”
- Výhody
 - Interaktivita
 - Plynulejší přechod mezi stránkami

FRONTENDOVÉ ZKRATKY - MPA

- MPA – Multi Page Application
 - Každá stránka má svůj HTML soubor
 - Při navigaci se načte nová stránka
- Výhody
 - Snažší správa obsahu – každá stránka je samostatný dokument
 - Lepší pro vyhledávače

KORPORÁTNÍ DŽUNGLE



MODELOVÝ PŘÍKLAD – SAP SIGNAVIO

- 2021: SAP + Signavio = SAP Signavio
- 2022: Process Intelligence v SAP Labs Brno
 - 2 hlavní FE aplikace
 - Několik Microfrontendů
- 2022 + 2023: Rapid Growth
 - 3 -> cca 10 týmů
- Tech Stack
 - JavaScript + TypeScript
 - React v18 (function components, hooks)
 - Apollo GraphQL
 - Jest, Cypress, Percy

```
describe('My First Test', () => {
  it('Gets, types and asserts', () => {
    cy.visit('https://example.cypress.io')

    cy.contains('type').click()

    // Should be on a new URL which
    // includes '/commands/actions'
    cy.url().should('include', '/commands/actions')

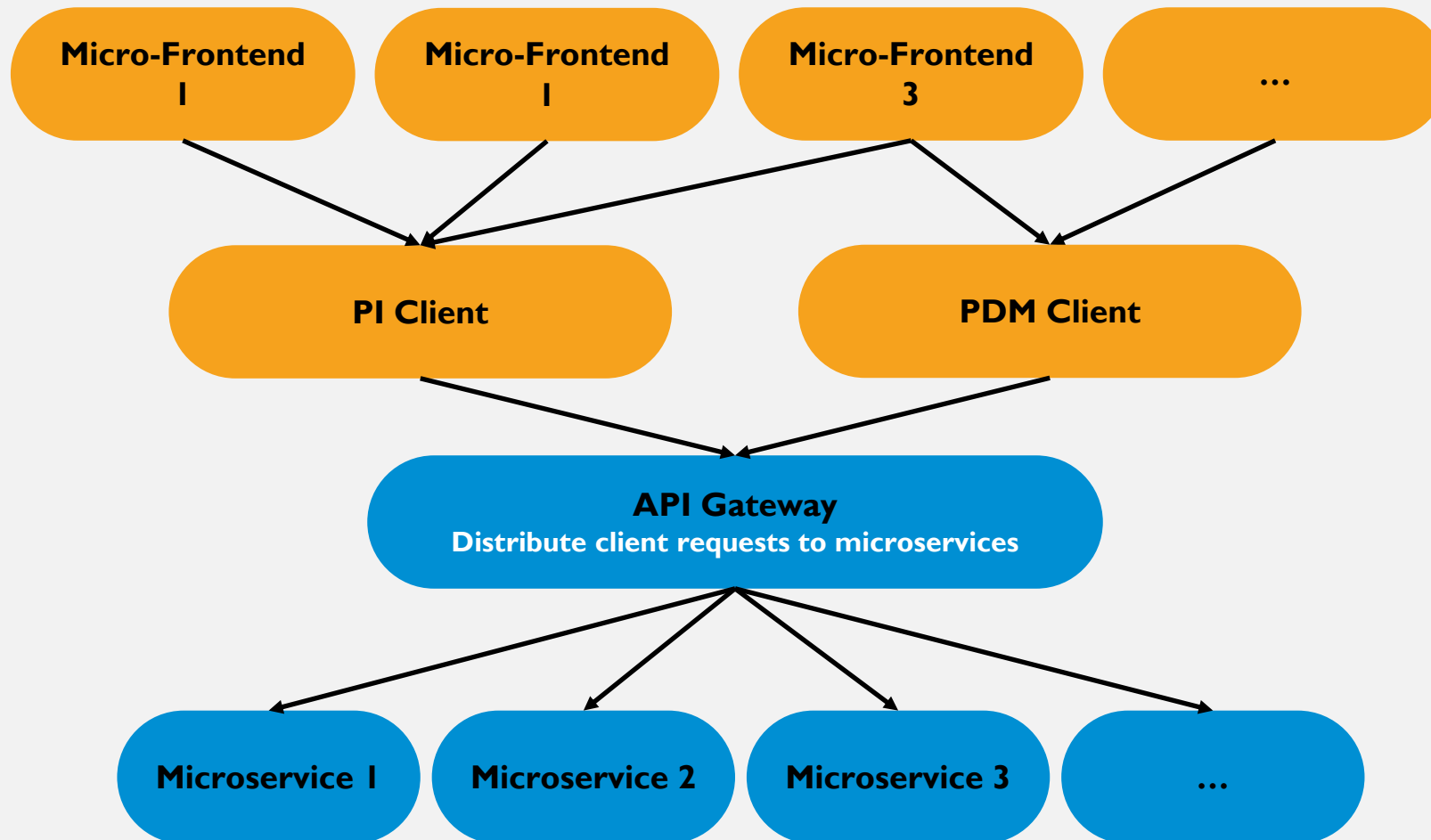
    // Get an input, type into it and verify
    // that the value has been updated
    cy.get('.action-email')
      .type('fake@email.com')
      .should('have.value', 'fake@email.com')
  })
})
```


INFRASTRUKTURA

- Kubernetes
 - Náročný na znalosti
- CircleCI
- Vercel
 - Preview deployments
- Continuous Deployment
 - Deploy to prod. ASAP
- Trunk based development
 - Feature flags



ARCHITEKTURA



TECHNICÉ VÝZVY

- Škálování infrastruktury
- Zajištění plynulosti vývoje
 - Flaky testy – ovlivní spoustu lidí
- Zajištění kvality kódu
 - Potřeba automatizace, skenů a reportů
- Dokumentace
 - Včetně zaznamenávání klíčových rozhodnutí
 - ADR – Architecture Decision Record
- Používání firemních technologií
 - UI5 Web Components, Fiori
- Logování

WHY ARE ELEVATOR JOKES SO GOOD?

WHY ARE ELEVATOR JOKES SO GOOD?

THEY WORK ON SO MANY LEVELS.

TECHNICÉ VÝZVY – UI5

- SAPUI5 / OpenUI5
 - Proprietární framework pro SAP byznysové aplikace
 - MVC (Model View Controller)
 - XML Views

FRAMEWORK SAPUI5

```
<mvc:View xmlns:mvc="sap.ui.core.mvc" xmlns="sap.m" xmlns:core="sap.ui.core"
  controllerName="myApp.controller.Main">
  <VBox>
    <Text text="Aktuální teplota: {weatherModel>/temperature}°C" />
  </VBox>
</mvc:View>
```

FRAMEWORK SAPUI5

```
sap.ui.define([
    "sap/ui/core/mvc/Controller",
    "sap/ui/model/json/JSONModel"
], function(Controller, JSONModel) {
    "use strict";

    return Controller.extend("myApp.controller.Main", {
        onInit: function() {
            // Inicializace modelu pro data o počasí
            var oWeatherModel = new JSONModel();
            this.getView().setModel(oWeatherModel, "weatherModel");

            // Simulace načítání dat z API při inicializaci
            this.loadWeatherData();
        },

        loadWeatherData: function() {
            // Simulace načítání dat z API
            var that = this;
            fetch('https://api.example.com/weather')
                .then(response => response.json())
                .then(data => that.getView().getModel("weatherModel").setData(data));
        }
    });
});
```


TECHNICÉ VÝZVY – UI5

- SAPUI5 / OpenUI5
 - Proprietární framework pro SAP byznysové aplikace
 - MVC (Model View Controller)
 - XML Views
- UI5 Web Components
 - Odlehčený framework používající web componenty

WEB COMPONENTS



WEB COMPONENTS

- 2012*
- Standardizované stavební bloky webových stránek, umožňující vytváření hermeticky uzavřených, opakovaně použitelných komponent s vlastním skriptem, šablonou a styly.
- 3 základní části
 - Custom Elements
 - Vlastní HTML elementy s vlastním chováním
 - Shadow DOM
 - Izolovaný kontejner pro DOM elementy
 - Chrání styl a chování komponenty
 - HTML Templates
- Nezávislé na frameworku

WEB COMPONENT

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="CounterComponent.js" defer></script>
</head>
<body>

  <!-- Použití vlastní webové komponenty -->
  <counter-component></counter-component>

</body>
</html>
```

WEB COMPONENT

```
customElements.define('counter-component', CounterComponent);
```

```
class CounterComponent extends HTMLElement {
  constructor() {
    super();

    this.attachShadow({ mode: 'open' });

    this._count = 0;

    this.shadowRoot.innerHTML = `
      <style>
        div {
          font-size: 24px;
          text-align: center;
          margin: 20px;
        }

        button {
          font-size: 18px;
          margin: 0 5px;
          padding: 5px 10px;
        }
      </style>
      <div>
        <button id="decrement">-</button>
        <span id="count">${this._count}</span>
        <button id="increment">+</button>
      </div>
    `;

    this._decrementButton = this.shadowRoot.getElementById('decrement');
    this._countDisplay = this.shadowRoot.getElementById('count');
    this._incrementButton = this.shadowRoot.getElementById('increment');

    this._decrementButton.addEventListener('click', () => this._decrement());
    this._incrementButton.addEventListener('click', () => this._increment());
  }
}
```

TECHNICÉ VÝZVY – UI5

- SAPUI5 / OpenUI5
 - Proprietární framework pro SAP byznysové aplikace
 - MVC (Model View Controller)
 - XML Views
- UI5 Web Components
 - Odlehčený framework používající web componenty
- UI5 Web Components for React
 - React obálka nad UI5WC

TECHNICÉ VÝZVY – DESIGN GUIDELINES

- Fiori
 - Set pravidel pro všechny SAP frontendové aplikace
 - UI5: Podpora out of the box



PROCESNÍ VÝZVY

- Rozdělení vlastnictví částí aplikaci
 - Zamezení merge conflict
 - Když se něco rozbije, víme, na koho ukázat
 - Refactoring, npm packages, MicroFrontends
- Budování (Coding) Guidelines
 - Unifikace kódu napříč aplikací
 - Standardizace UI/UX
- Community of Practice

PERSONÁLNÍ VÝZVY

- Onboarding nových lidí
 - Nutná podpora “veteránů”
- Týmová spolupráce
 - Efektivní komunikace
- Pravidelné alignment
 - Různé časové zóny



KORPORÁTNÍ VÝZVY

- Product Standards
 - Např. accessibility
- Security
 - Používání správných verzí ověřených nástrojů a balíčků
- Schválený Software
 - -> migrace, migrace, migrace
- Mandatory scans
 - Code coverage, quality checks...
- Design Guidelines
 - Fit into the corporate ecosystem
- GitHub Copilot



Corporate needs you to find the differences between this picture and this picture.

They're the same picture.

KORPORÁTNÍ VÝHODY

- Chyba neznamená konec
- Dostupnost (drahých) profesionálních nástrojů
- Pracovní prostředí bez stresu
 - Work Life Balance
- Služební auto

MICROFRONTENDS



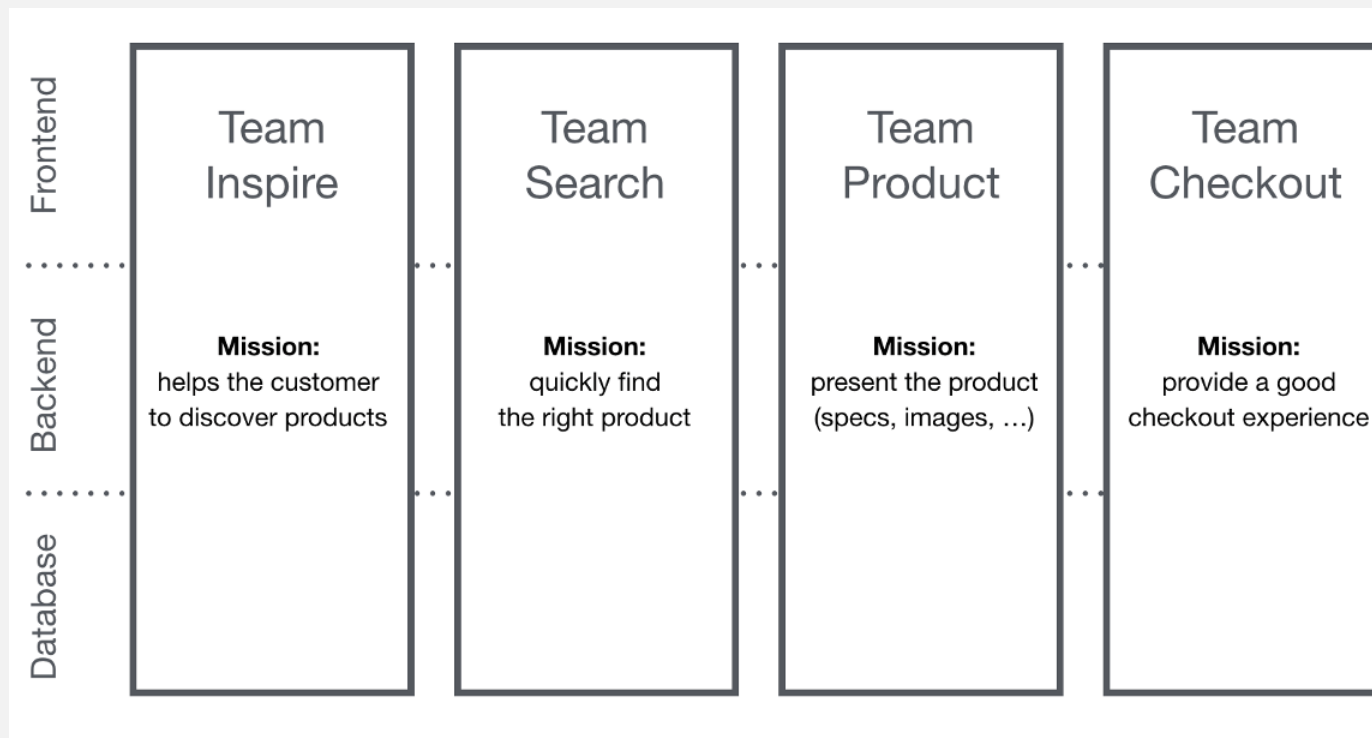
MICROFRONTEND

- Microservice in the Frontend World



MICROFRONTEND – ZÁKLADNÍ MYŠLENKY

- Nezávislé na technologii
 - Každý tým si určí svůj tech stack
- Izolace kódu
 - Žádný sdílený/globální stav
- Týmové prefixy a správné názvosloví
 - CSS, local storage, cookies
- Nativní API prohlížečů
 - KISS



MICROFRONTEND – SAP SIGNAVIO

The screenshot displays the SAP Signavio interface. At the top, a dark blue header bar contains the 'SAP Signavio' logo and 'Process Intelligence' text on the left. On the right side of this bar, the word 'Notifications' is written in yellow, followed by a search icon, a bell icon (highlighted with a yellow circle), a grid icon, a question mark icon, and a user profile icon labeled 'MK'. Below the header, a light gray bar contains the text 'All processes' with a help icon, the text 'Global Header' in red, and three buttons: 'Action Triggers' (highlighted with a green circle), 'Manage data', and 'New process'. The word 'Triggers' is written in green below the 'Action Triggers' button. The main content area below is a large white rectangle.

DĚKUJI ZA POZORNOST

Michal Kozubík

Development Expert

SAP Signavio, SAP Labs Brno

michal.kozubik@sap.com

