

JÖRG KINZEBACH

STATISCHE WEBSEITEN

KNF Kongress 2023

JÖRG KINZEBACH

VORWORT

KNF Kongress 2023

TIMELINE

Apache httpd

performanter als der CERN
httpd und deutlich mehr
Optionen

erster Web-Accelerator

erste Firma mit “close to customer”
Web-Caches um Webseiten-Zugriffe
zu beschleunigen

JAMStack

Webentwicklungsarchitektur mit
“statischen Webseiten” und hoher
Performance

1990

CERN httpd

erster Webserver, der zuerst nur Dateien
(statische Inhalte) ausliefern konnte

1995

PHP

entwickelt für die Erstellung von
dynamischen Webseiten

1995

1998

WordPress

freies Content Management System
(CMS) ursprünglich für Blogs
entwickelt

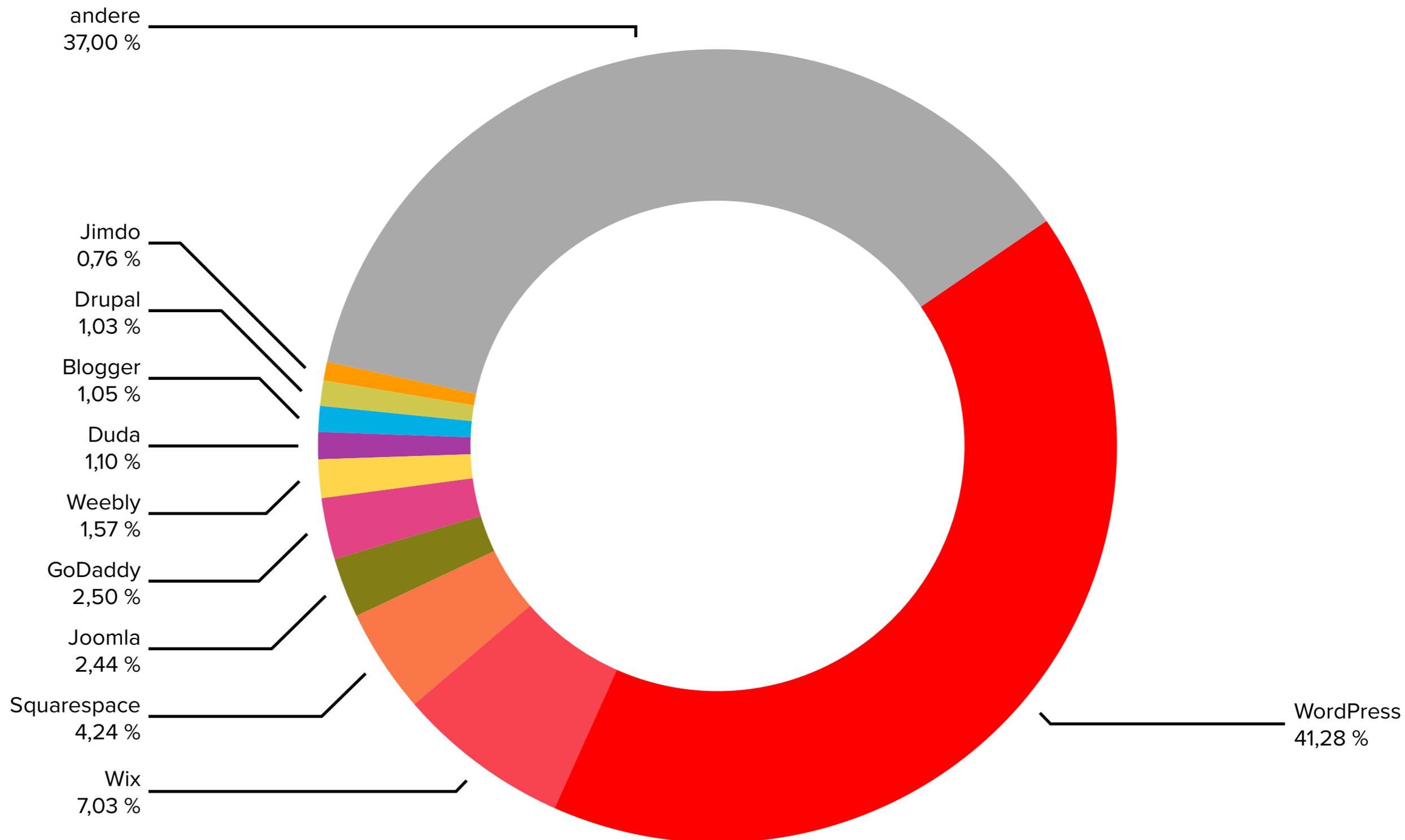
2003

2015

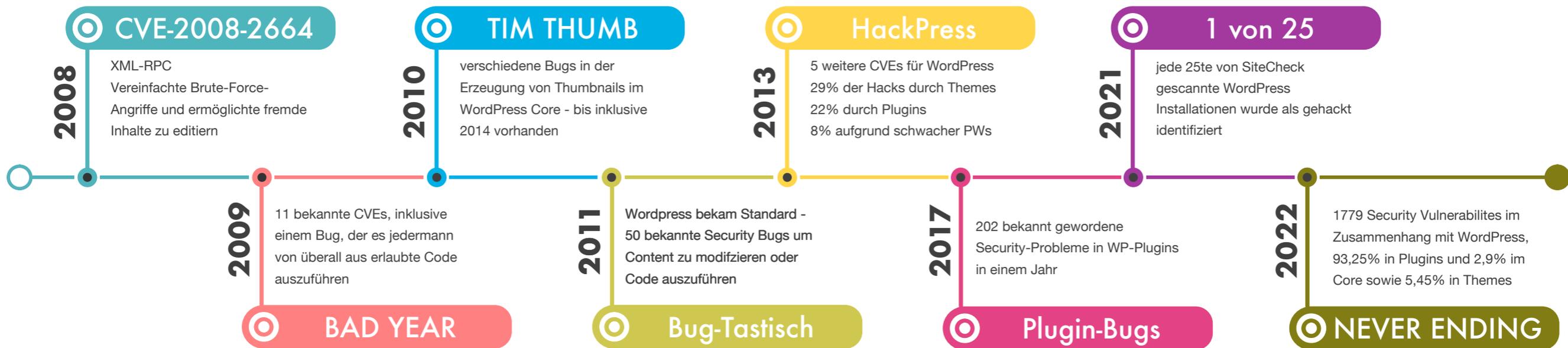
**NIEMAND WILL HEUTE NOCH DUTZENDE
VON SEITEN MIT HTML UND NAVIGATIONS-
LINKS VON HAND EDITIEREN/PFLEGEN**

... EIN MANAGEMENT SYSTEM MUSS(TE) HER

CMS VERBREITUNG



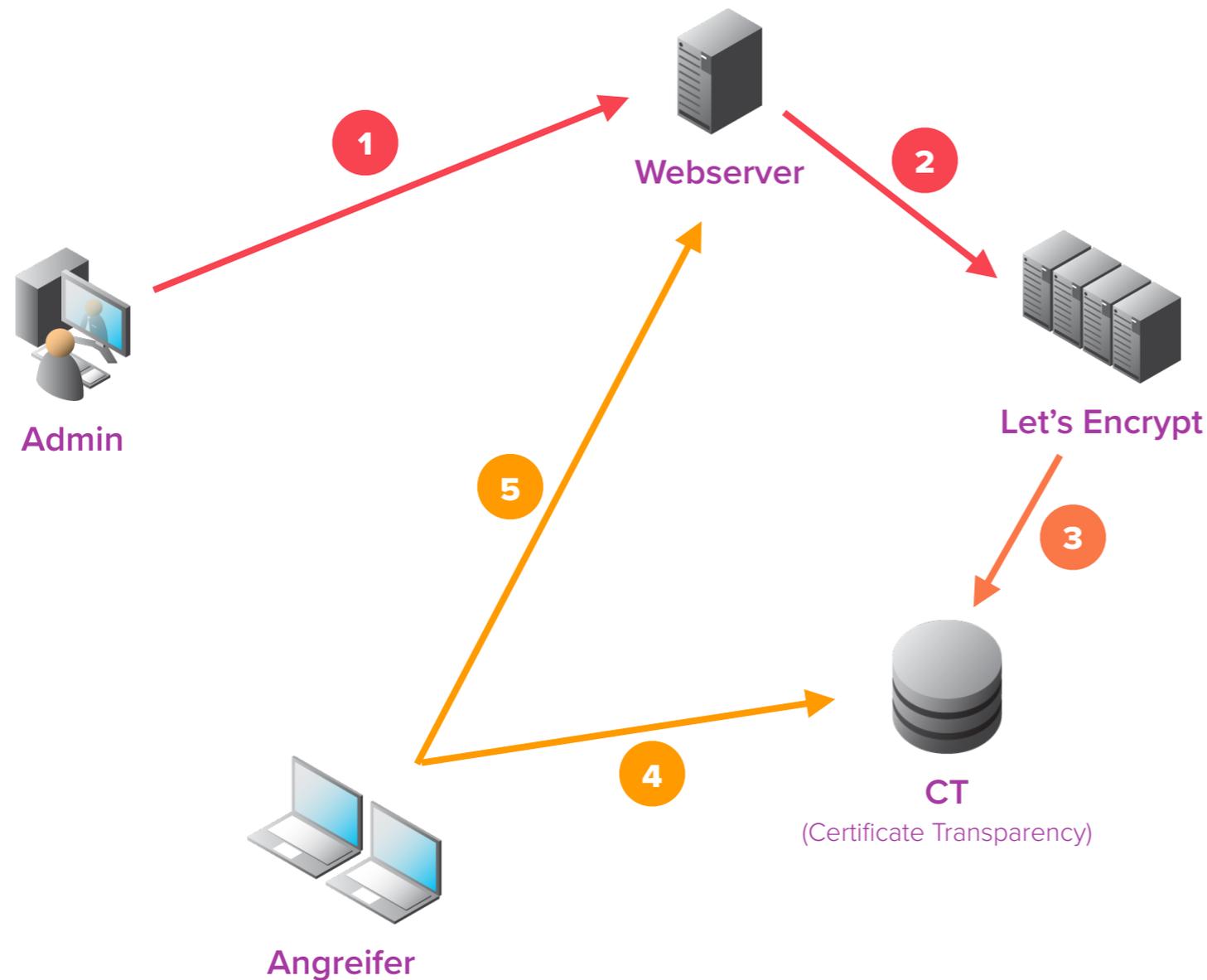
WORDPRESS & SECURITY



SSL

MACHT ALLES NOCH VIEL SCHLIMMER

ZERO MINUTE ATTACK



- 1) Installationskript wird von Admin auf den Server geladen (und gestartet)
- 2) Server holt sich Zertifikat von Let's Encrypt
- 3) CT Datenbank bekommt Eintrag für Hostname des Servers von Let's Encrypt
- 4) Angreifer lernt Hostname eines frisch installierten Servers aus der CT Datenbank
- 5) Angreifer kann austesten, ob sich bekannte und angreifbare Installation auf dem Server befindet (Default-Passwörter u.ä.)

PERFORMANCE

VS DYNAMIK

CODEZEILEN

 **TYP03** 850.000

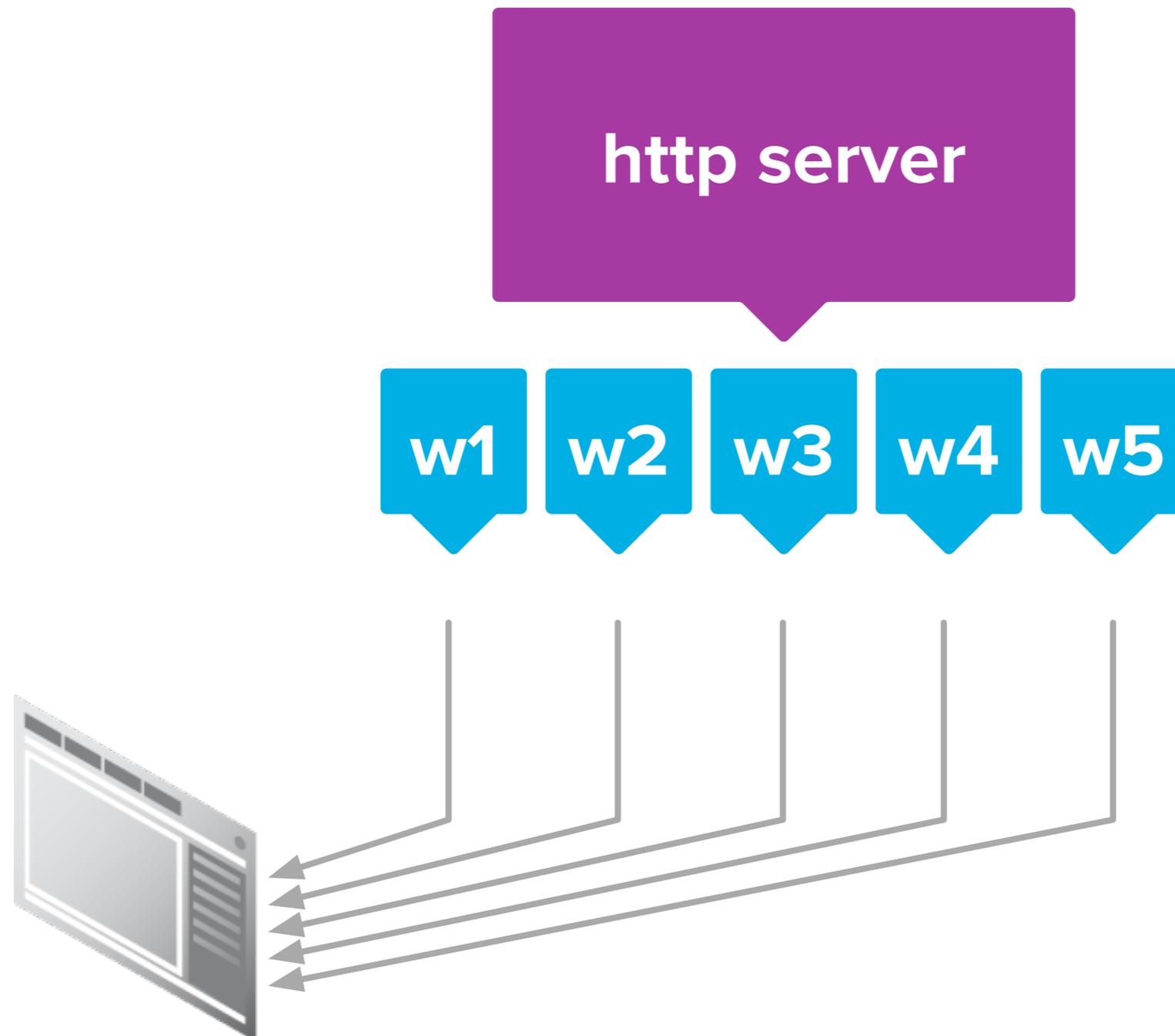
 **Drupal™** 1.850.000

 **Joomla!®** 1.156.117

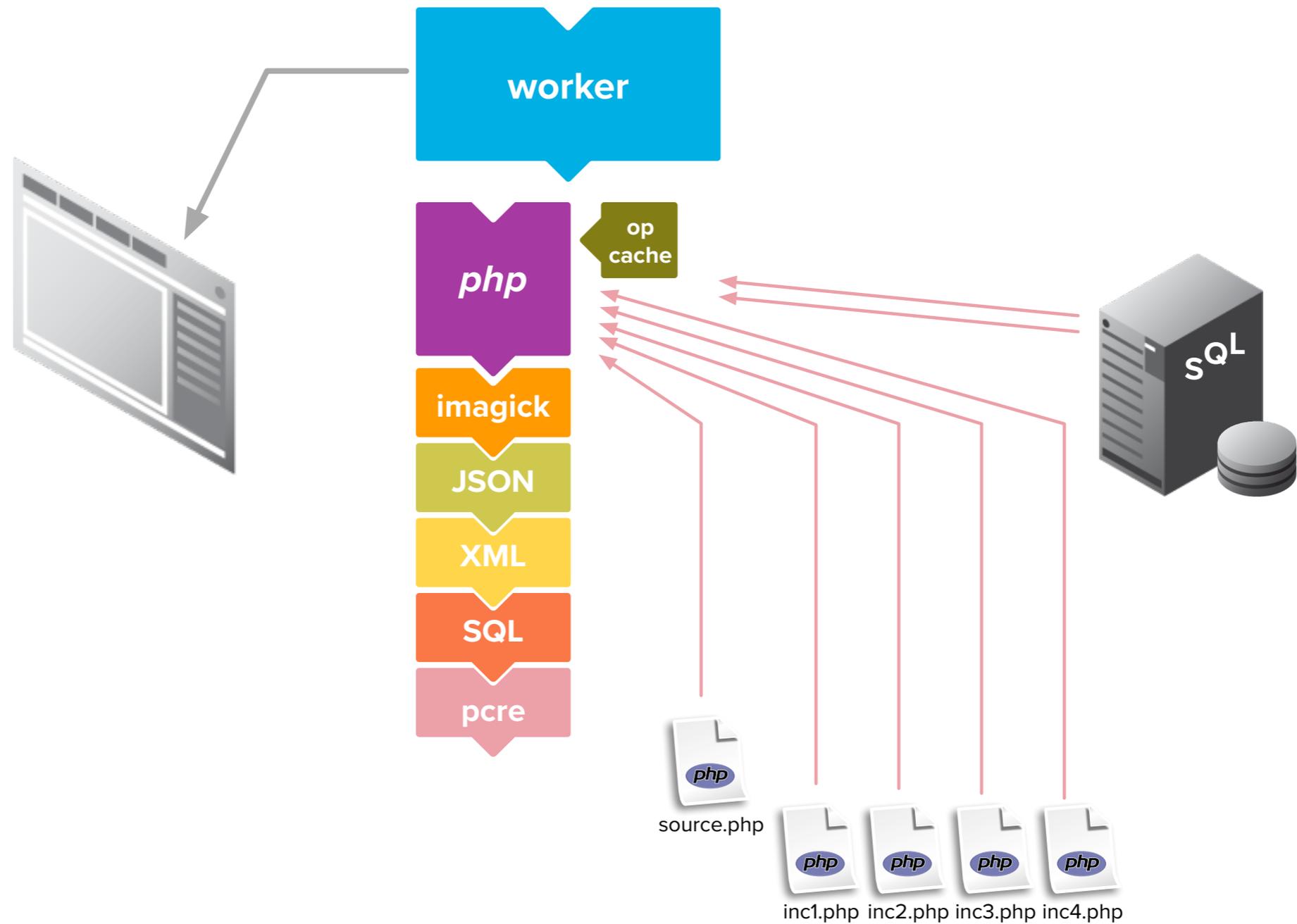
 **WORDPRESS** 1.126.110

jeweils ohne Plugins/Extensions

DYNAMISCHER CONTENT



DYNAMISCHER CONTENT



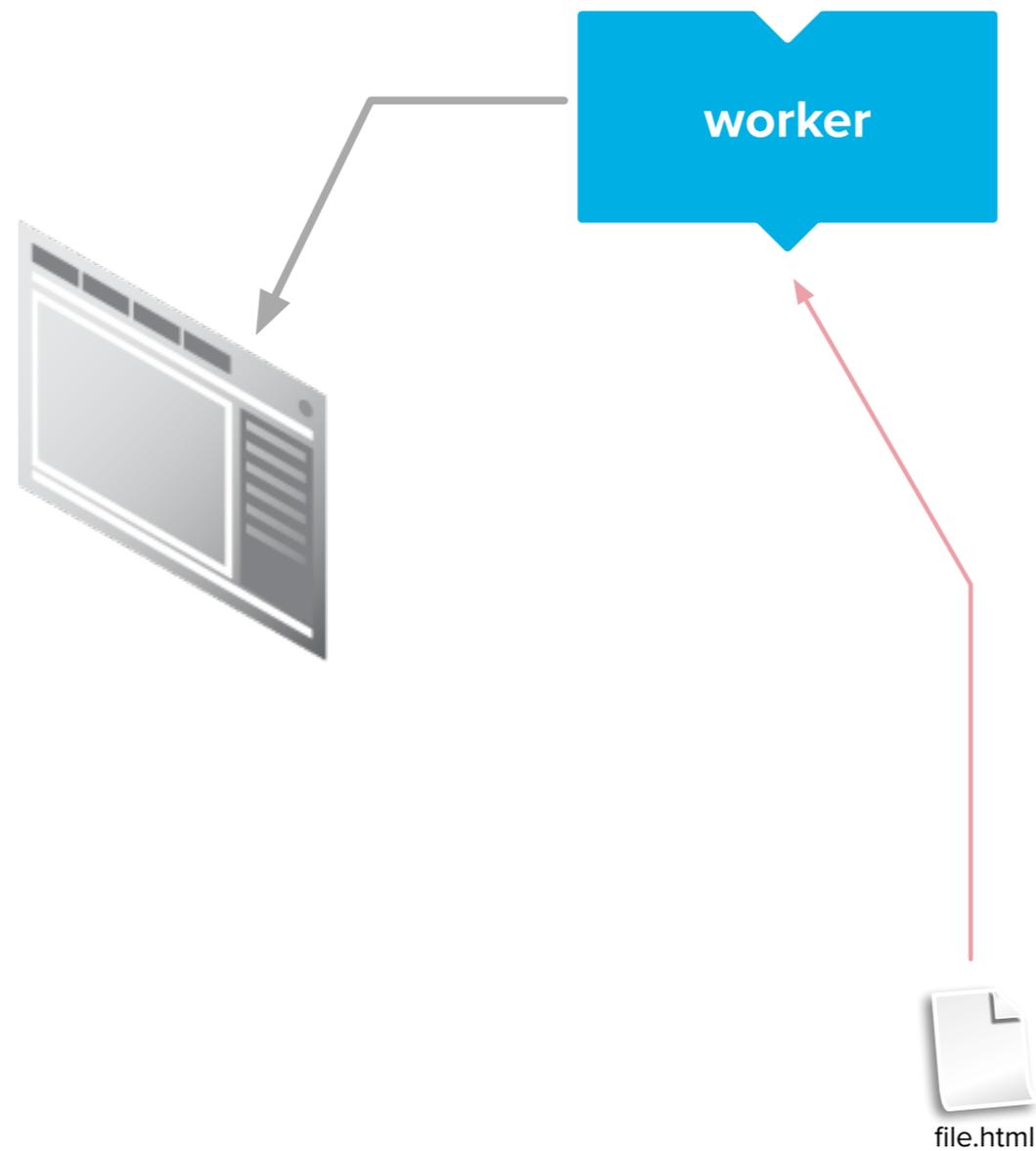
LAMP

LINUX - APACHE - MYSQL - PHP

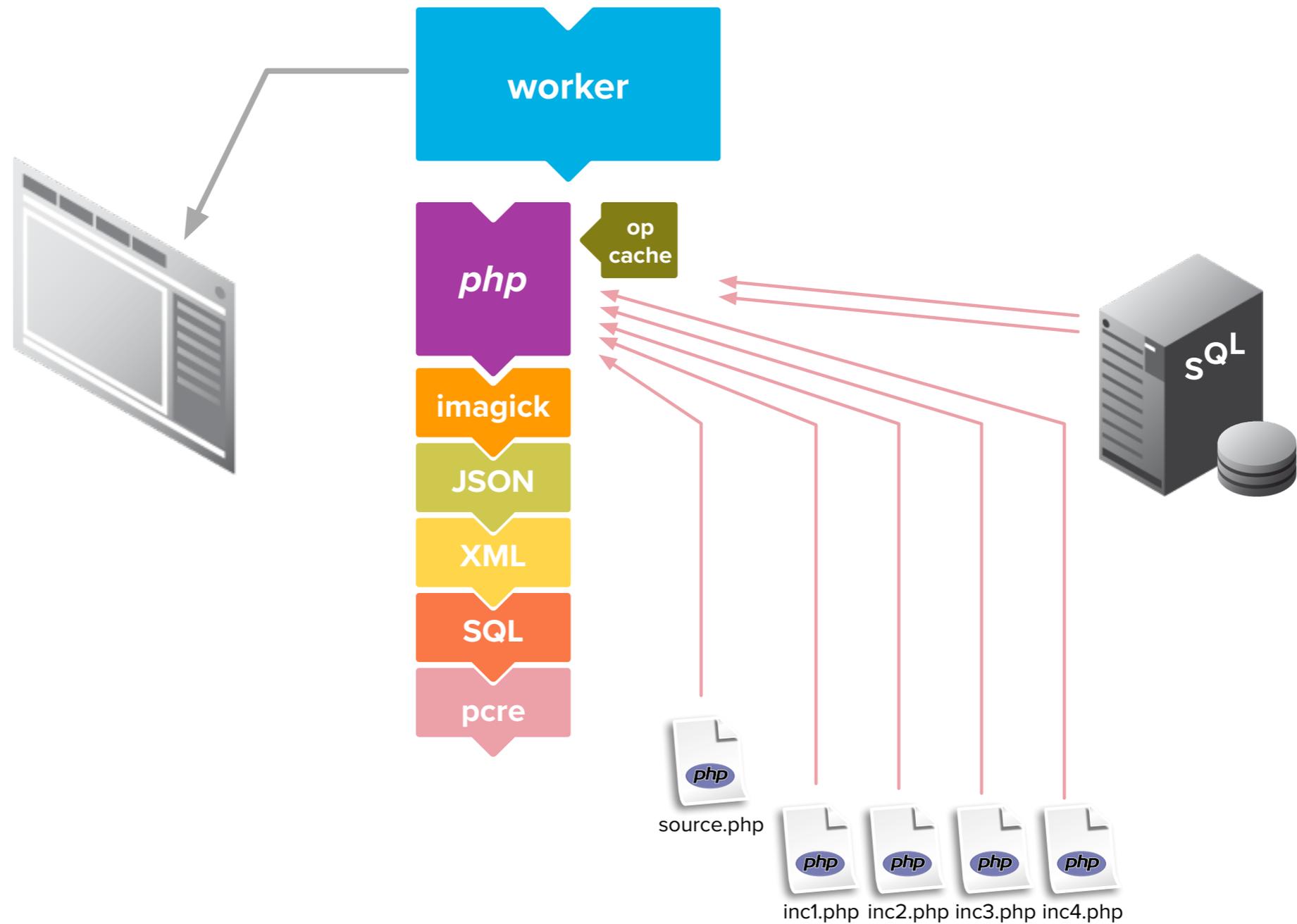
STATISCH

STATISCHE FILES, DER SERVER INTERPRETIERT NICHTS

STATISCHER CONTENT



DYNAMISCHER CONTENT



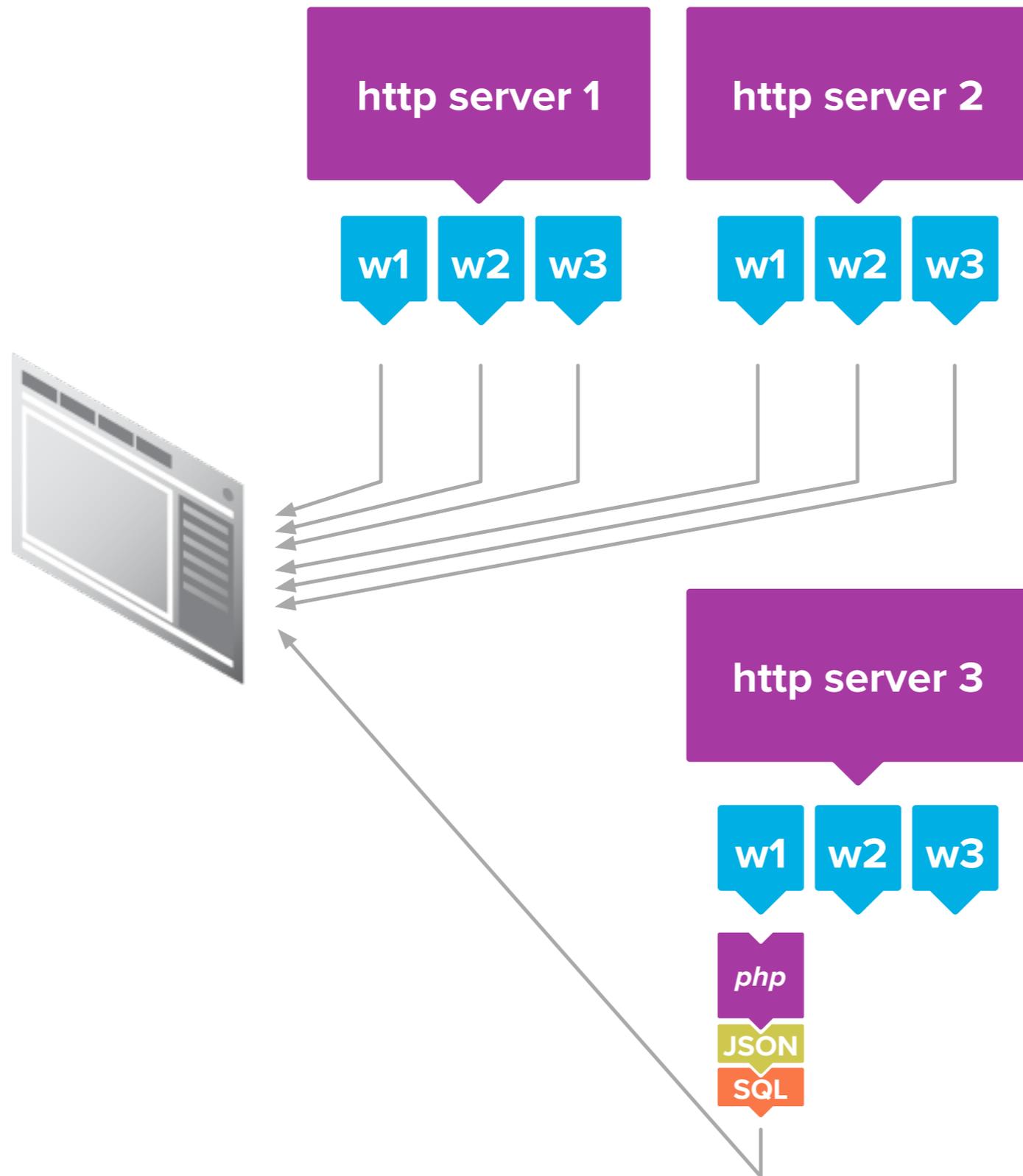
VORTEILE VON STATISCH

- **Security**
- **Antwortzeiten**
 - keine Datenbank-Abfragen
 - keine CPU-Zyklen für Berechnung
- **Server-CPU**
 - Mehr Ressourcen für mehr Auslieferungen
- **Caching**
 - Browser-Cache und Proxies brauchen keine Magie um zu entscheiden ob bereits vorhandene Inhalte noch aktuell sind
- **Skalierbarkeit**
 - Statische Files können parallel auf vielen Servern liegen und von dort ausgeliefert werden
 - Keine Abhängigkeit an Software-Versionen auf dem Server

**ABER NICHT ALLES
IST STATISCH?**

NUR DYNAMISCHE INHALTE MÜSSEN DYNAMISCH ERZEUGT WERDEN

DYNAMISCHE ELEMENTE



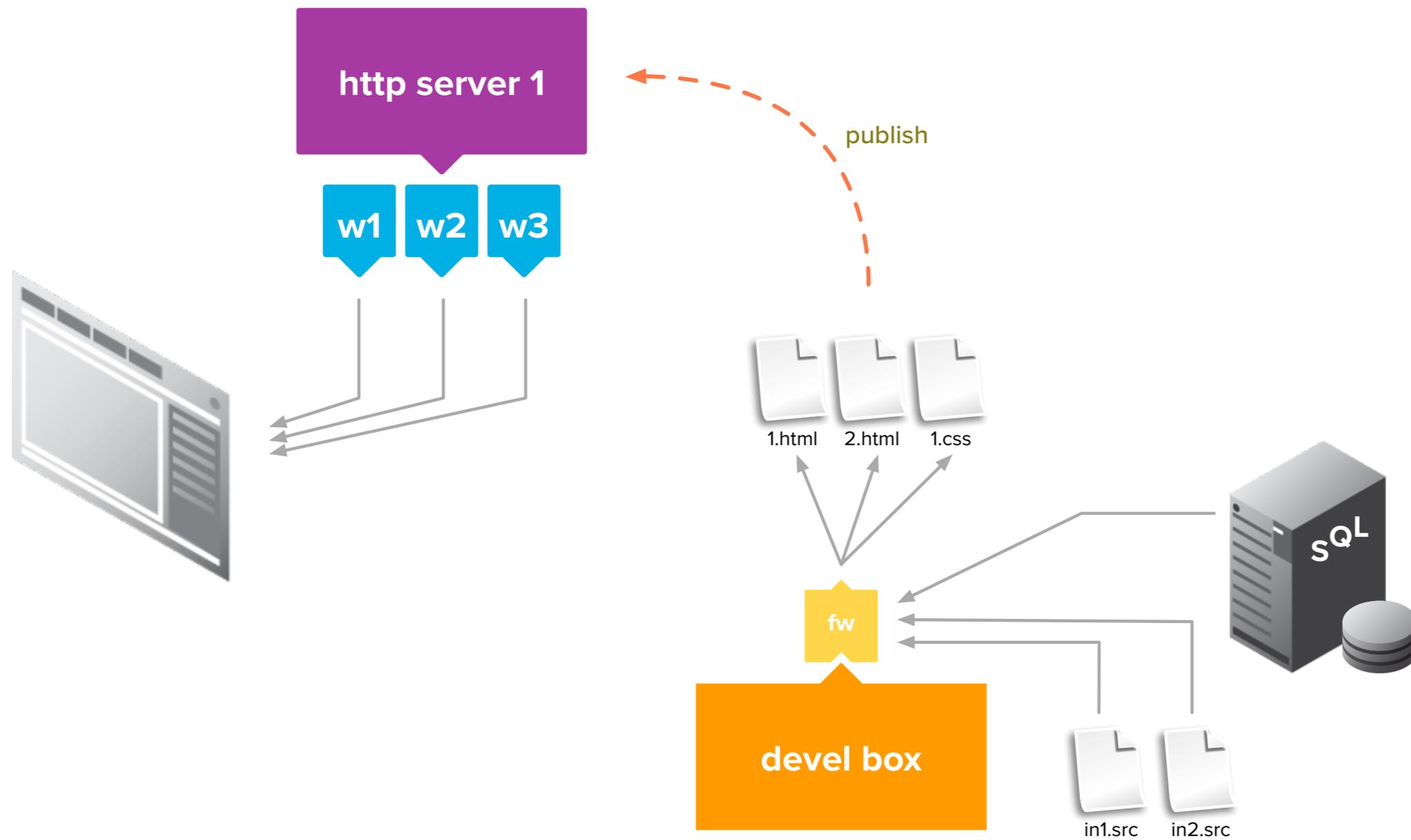
JAMSTACK

JAVASCRIPT, APIS UND MARKUP

JAMSTACK WEBSITE/APP

- Daten und Darstellung werden getrennt
- Input wird von der „Hauptseite“ verarbeitet und in eine Darstellung für den Nutzer umgewandelt
- Große Teile des States einer Webseite/Formularverarbeitung können im Browser gehalten werden und müssen nicht umständlich über Cookies oder URL-Rewriting mit dem Server ausgetauscht werden
- Nur spezielle unbedingt dynamisch nötige Daten einer Webseite werden dynamisch angefordert und geliefert
- Daten werden in einem Roh-Format geliefert (z.B. JSON oder XML)
 - Daten können potentiell auch wieder in einfachen Flat-Files stehen (z.B. News-Artikel, Kalender-Einträge) und brauchen somit nicht zwingend eine SQL-artige Datenbank oder ein programmiertes Backend

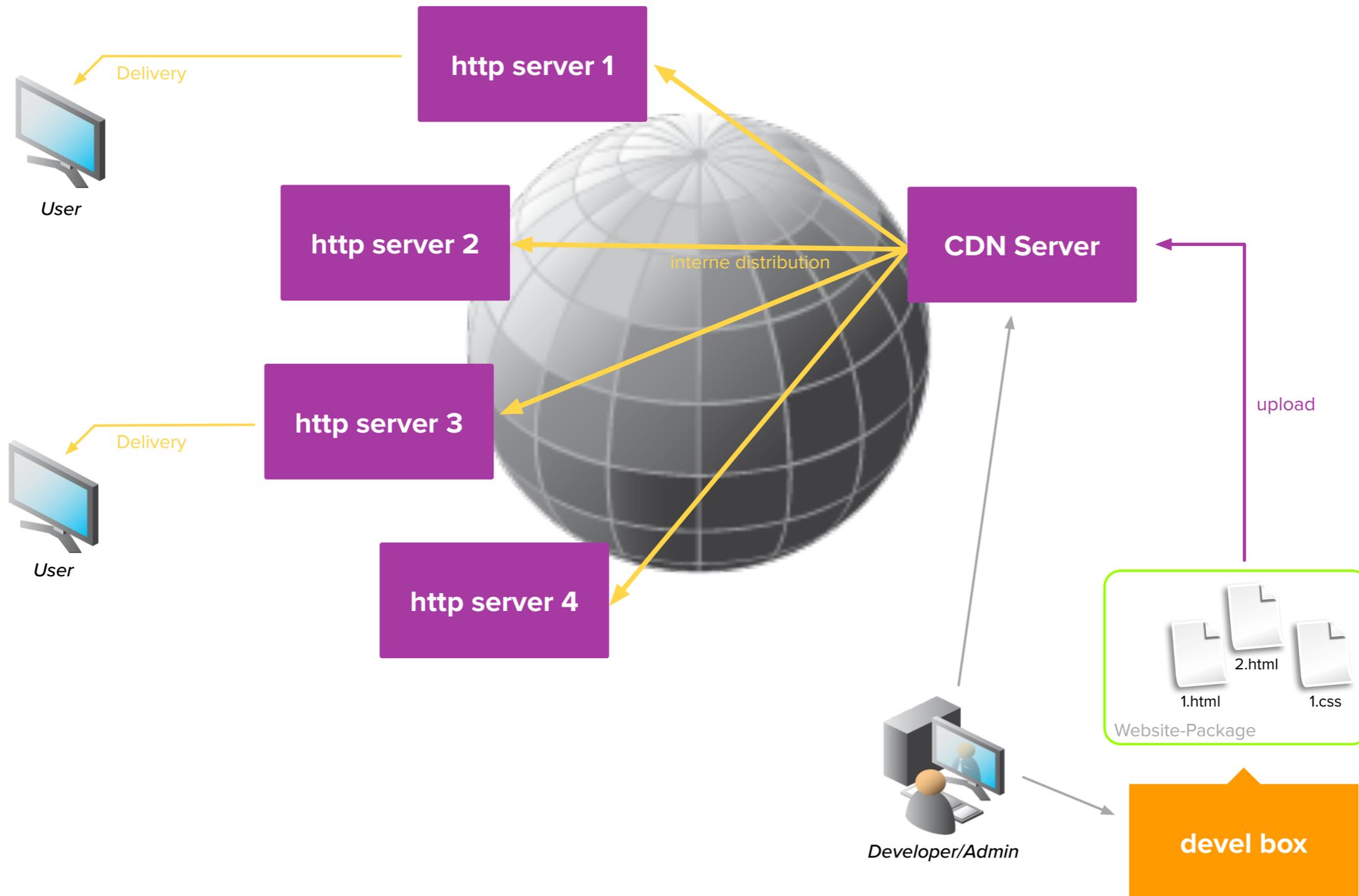
JAMSTACK



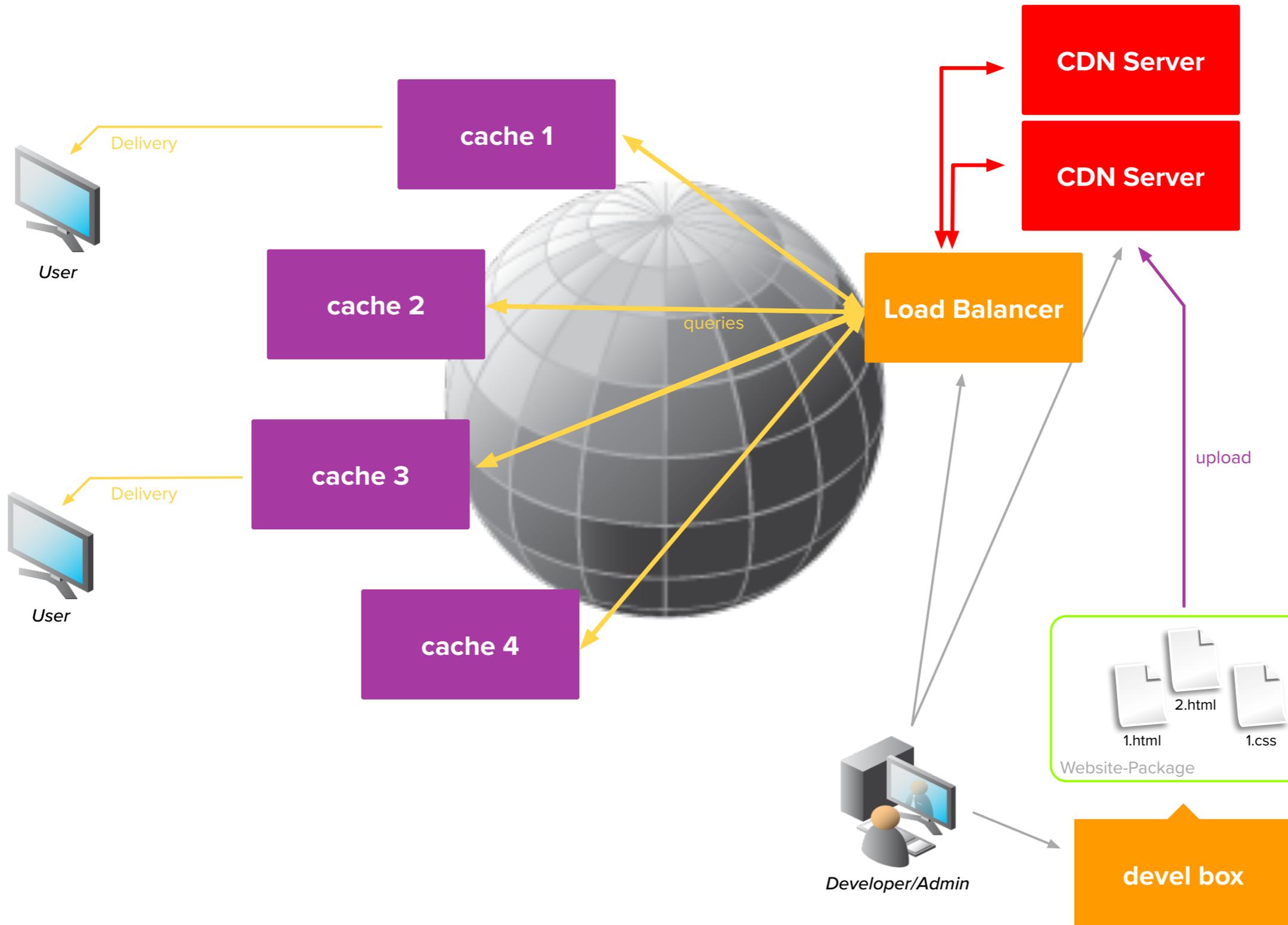
JAMSTACK

- JAMstack ist prinzipiell ein Konzept und keine Technologie (Begriff geprägt durch Netlify)
- Skalierbarkeit durch überall identische statische Daten
- unabhängig vom Host
- Trennung von Daten und Design
- Reaktive Webseiten durch Anwendung im Browser möglich
- schnellere Ladezyklen (potentiell keine Seitenwechsel/
Nachladen von Seiteninhalten nötig)
- einfache Versionierung der Webseiten

JAMSTACK UND CDN



LAMP UND EDGE CACHING



CDN UPLOAD

- Theoretisch sind viele Upload-Techniken möglich
- Viele JAMstack focused CDN bieten nur noch GIT an
 - CDN hat direkten Zugriff auf GitHub/GitLab/Bitbucket/Azure DevOps Repositories und holt sich Daten aus dem Verwaltungssystem
 - Deploy/Publish kann optional automatisch nach GIT push gestartet werden
 - CDN kann vor einer Veröffentlichung auch eine Vorschau auf einer Preview-URL bereitstellen
- Einige CDN Anbieter können die Webseite direkt aus den JAMstack Sourcen bauen
 - Erkennen ob eine Webseite den jeweiligen Stack-Standards entspricht oder nur auf dem Rechner des Entwickler/Designers gebaut werden kann
 - Erlaubt einfachere Fullstack-Anwendungen
 - Erzeugt wieder eine potentielle Abhängigkeit auf Anbieter mit diesem Feature-Set

**WIE MACHT MAN
DAS?**

MIT DIESEM JAMSTACK

JAMSTACK FRAMEWORKS

■ REACT

- 40.14% beliebtestes front-end framework
- bei facebook entwickelt und 2011 erstmals für deren Newsfeed eingesetzt
- an XML angelehnte Template-Sprache zur Deklaration von Komponenten

■ ANGULAR

- 22.96% zweitbeliebtestes front-end framework
- wird angeführt von Google durch eine Software-Community entwickelt
- basiert auf TypeScript
- Hauptfokus auf mobilen Anwendungen

■ VUE

- 18.97% drittb Liebtestes front-end framework

■ ...

VUE

- Fokus auf Single-Page Web-Anwendungen
- Einfacher zu erlernen als Angular oder REACT
- Entwickelt von Evan You in 2014 (Erscheinungsdatum) mit dem Ziel mächtig aber leichter und unkomplizierter als Angular zu sein
- Ein- und Ausgaben können direkt an Datenquellen gebunden werden ohne z.B. jQUERY zu benötigen
 - 40.14% beliebtestes front-end framework
 - bei facebook entwickelt und 2011 erstmals für deren Newsfeed eingesetzt
 - an XML angelehnte Template-Sprache zur Deklaration von Komponenten
- Instanzen von VUE-Anwendungen können mit Komponenten gefüllt werden, welche sich baumartig verschachteln lassen

VUE

- **Frameworks, die auf VUE aufsetzen**
 - Nuxt
 - Vuetify
 - Gridsome
 - VuePress
- **Viele VUE Komponenten und Plugins verfügbar**
 - vuetify: vorgefertigte Komponenten um Design-Arbeit zu sparen
 - vue-bootstrap: um das Design- und Komponentenframework „Bootstrap“ mit Vue zu verheiraten (bootstrap-vue-next für nuxt)
 - vorgefertigte Komponenten um Bilder-Galerien zu erzeugen
 - vorgefertigte Komponenten zur Abfrage von Datenquellen

FINDET MAN VUE IN FREIER WILDBAHN?

EINE KLEINE AUSWAHL

VUE

KVELERTAK THE INTERRUPTERS HANABIE SON

DE



ROCK IM
PARK

[JUNE 7-9, 2024
NÜRNBERG]

[FRI, JUNE 7]

GREENDAY **BROILERS** *BILLY
TALENT*

BABYMETAL · BAD OMENS · ELECTRIC CALLBOY

311 · AGAINST THE CURRENT · ANTILOPEN GANG · DOGSTAR · DONOTS
JAZMIN BEAN · KVELERTAK · L.S DUNES · PENDULUM · PINKSHIFT
ROYAL REPUBLIC · SCHIMMERLING · TEAM SCHEISSE · THE INTERRUPTERS
THE LAST INTERNATIONALE · THE SCRATCH · TRETSMANN · UNDEROATH

VUE

- **GitLab**
- **Nintendo**
 - (MyNintendo: für das Rewards-Programm)
- **Adobe**
 - (Behance: Nutzer-Showcase)
- **Xiaomi**
- **BMW**
 - (für den Wagen-Konfigurator)

GANZ KURZ NOCH...

■ Node.js

- plattformübergreifende JavaScript-Laufzeitumgebung

■ npm

- (früher: Node Package Manager)
- über das NPM Registry wird ein Archiv von JavaScript-Paketen bereitgestellt das mehr als 350.000 Komponenten beinhaltet
(das Repository wird von npm, Inc. betrieben, welche seit 2020 GitHub gehört [Microsoft])

■ yarn

- JavaScript Package Manager
- 2016 von einem facebook-Mitarbeiter entwickelt
- Wird in Zusammenarbeit von Meta, Expo.dev, Google und Tilde gepflegt

■ router

- mapping von URL auf Template und View, mehrere URLs können den gleichen View und/oder das gleiche Template verwenden

■ view

- Eine Ansicht/Darstellung von einer oder mehreren Komponenten

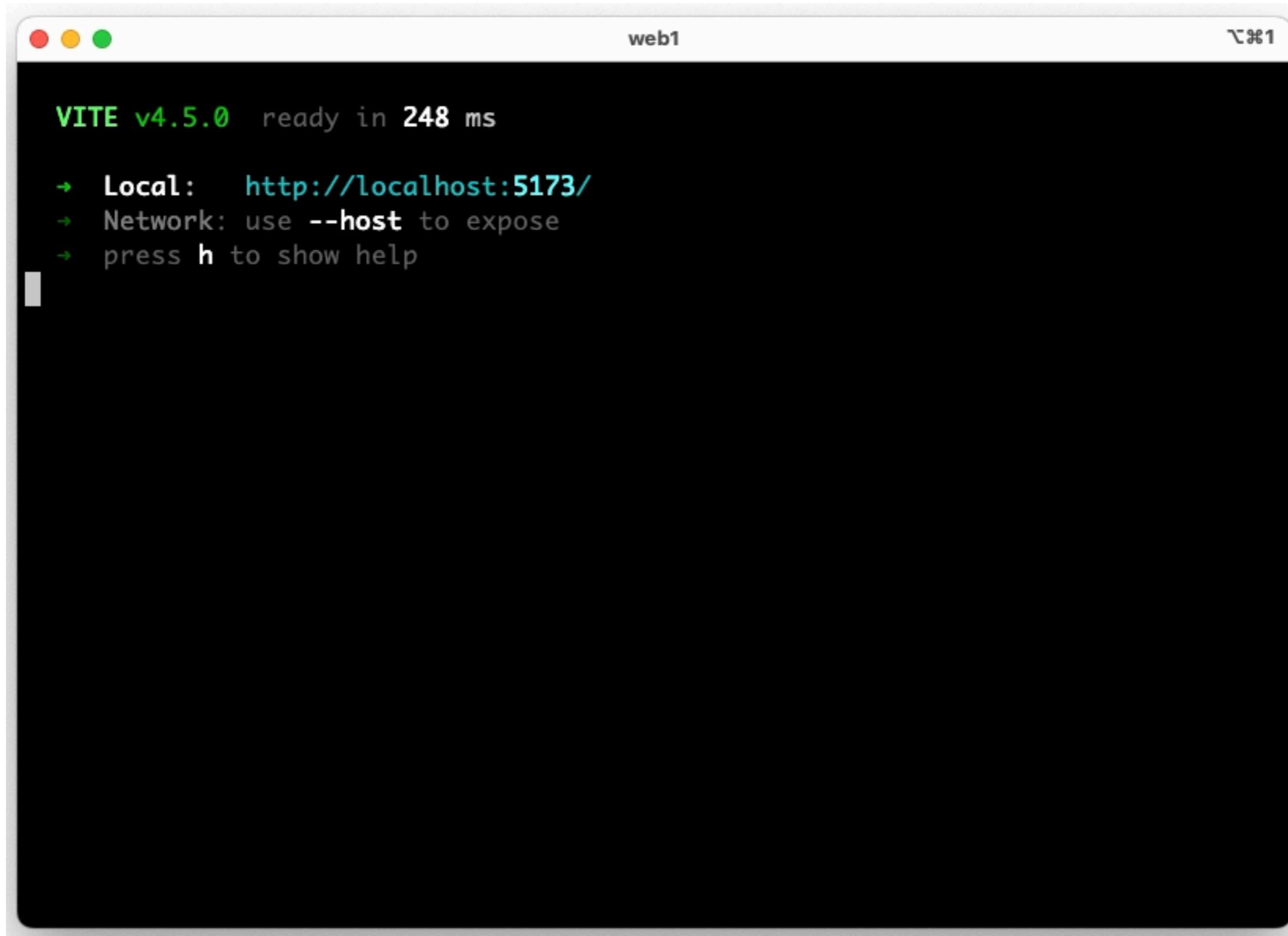
■ Page

- Eine einzelne (Unter-) Seite mit potentiell eigener URL

ERSTE SCHRITTE



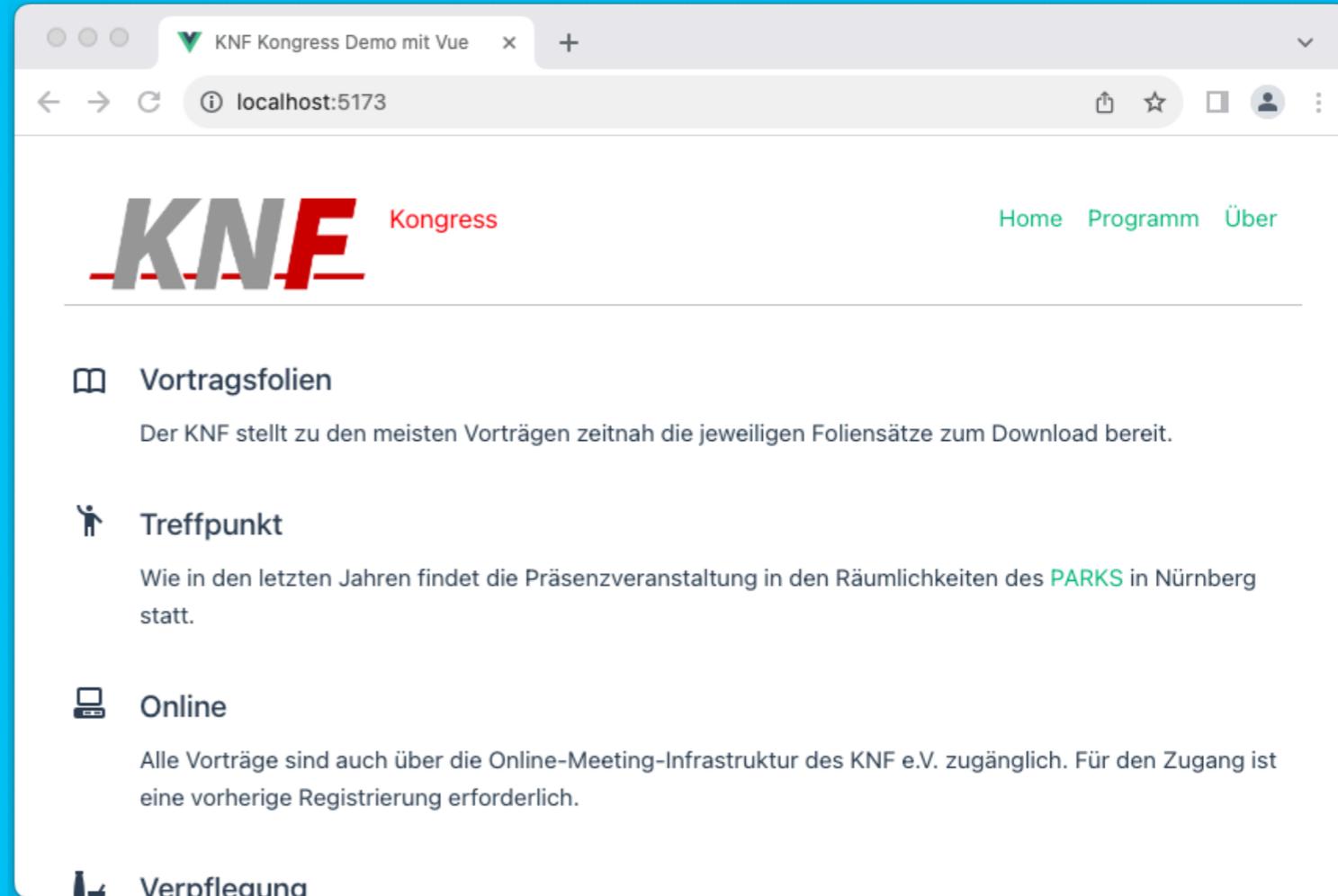
PUBLISH



```
web1 11:11 AM 11/11/2023  
VITE v4.5.0 ready in 248 ms  
→ Local:   http://localhost:5173/  
→ Network: use --host to expose  
→ press h to show help
```

VUE

KONGRESS BEISPIEL



APP.VUE

```
App.vue (~/knf/src) - VIM  3
<template>
  <app-header />
  <router-view/>
</template>

<script>
import AppHeader from '@components/AppHeader.vue'

export default {
  components: {
    AppHeader
  }
}
</script>
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
1  src/App.vue  1
"src/App.vue" 14L, 182B
```

APPHEADER.VUE

```
AppHeader.vue (~/knf/src/components) - VIM  3

<template>
  <header>
    
    <p class="kongress">Kongress</p>
    <nav>
      <ul>
        <li><router-link to="/">Home</router-link></li>
        <li><router-link to="/program">Programm</router-link></li>
        <li><router-link to="/about">Über</router-link></li>
      </ul>
    </nav>
  </header>
</template>

<style lang="scss">
  header {
    display: flex;
    border-bottom: 1px solid #ccc;
    padding: .5rem 1rem;

    p {
      margin-left: 1rem;
    }

    .kongress {
      color: #ff0000;
      font-size: 110%;
    }
  }
</style>

1 src/components/AppHeader.vue 1
"src/components/AppHeader.vue" 43L, 764B
```

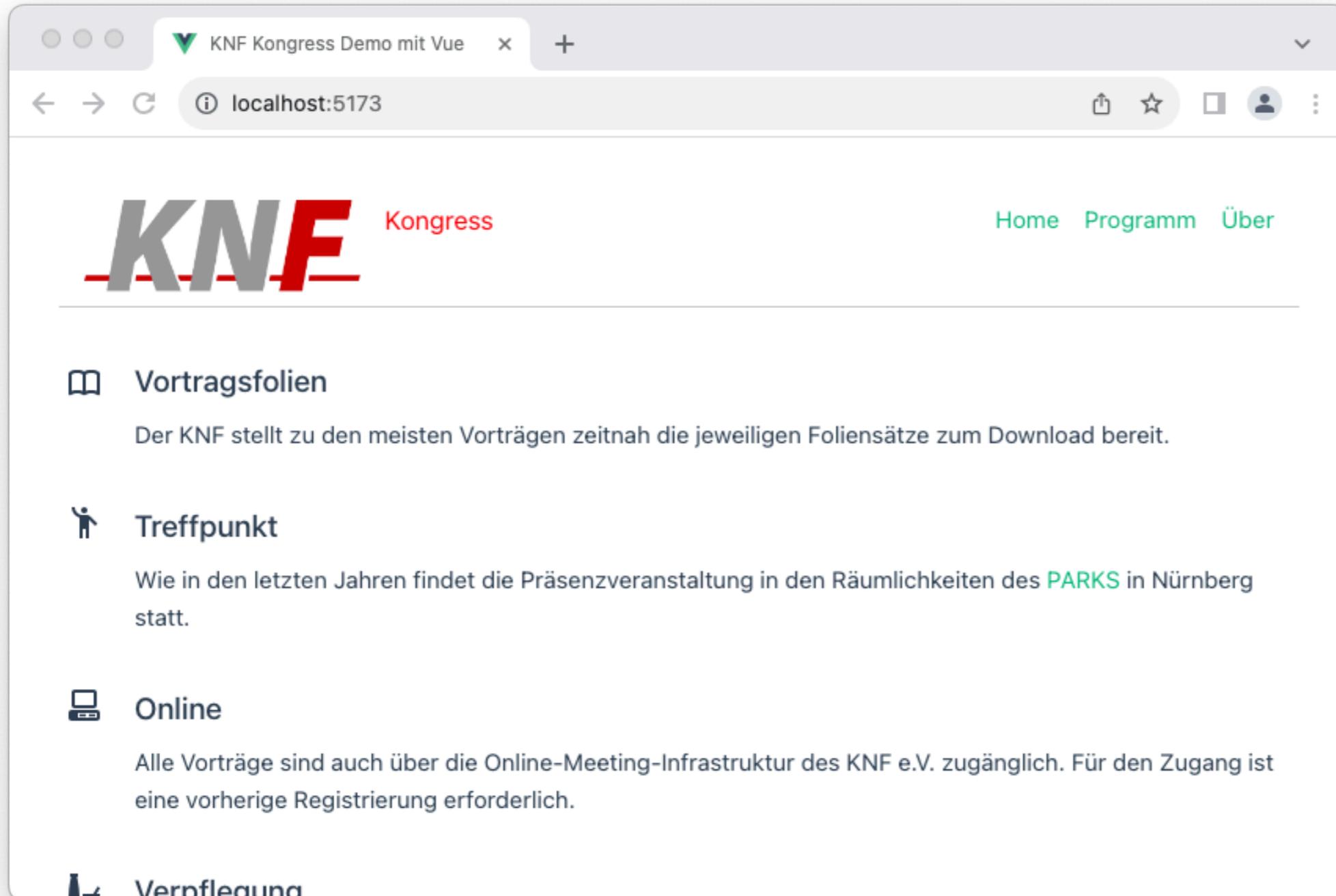
ROUTER

```
index.ts (~/knf/src/router) - VIM  3
import { createRouter, createWebHistory } from 'vue-router'
import HomeView from '../views/HomeView.vue'
import ProgramView from '../views/ProgramView.vue'

const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  routes: [
    {
      path: '/',
      name: 'home',
      component: HomeView
    },
    {
      path: '/program',
      name: 'program',
      component: ProgramView
    },
    {
      path: '/about',
      name: 'about',
      // route level code-splitting
      // this generates a separate chunk (About.[hash].js) for this route
      // which is lazy-loaded when the route is visited.
      component: () => import('../views/AboutView.vue')
    }
  ]
})

1 src/router/index.ts 1
"src/router/index.ts" 29L, 727B
```

HOME



PROGRAMM

KNF Kongress

Home Programm Über

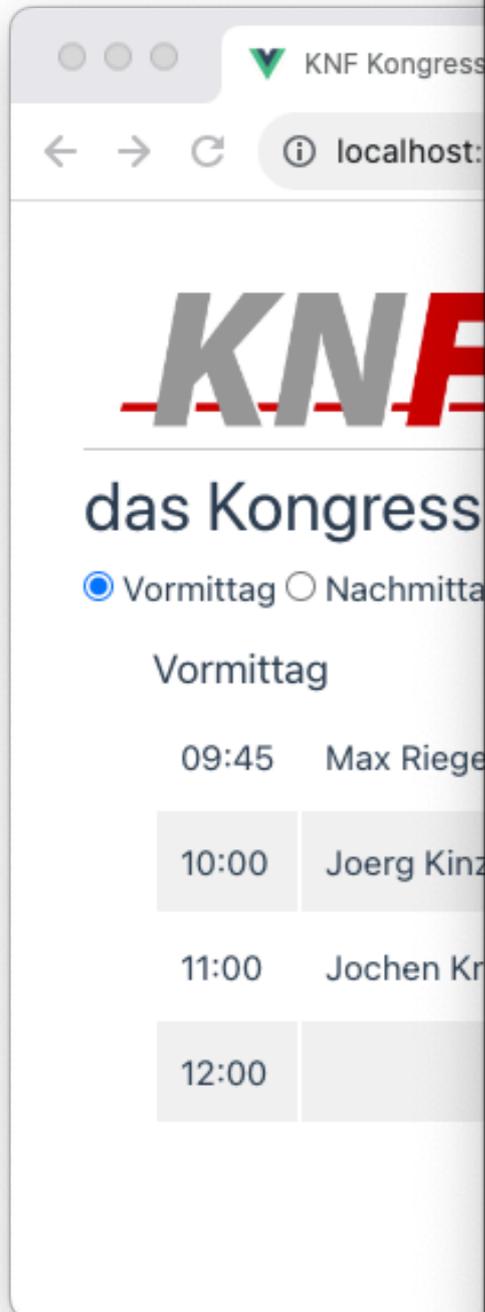
das Kongress Programm

Vormittag Nachmittag

Vormittag

09:45	Max Riegel	Beginn und Einführung
10:00	Joerg Kinzebach	Statische Webseiten
11:00	Jochen Krapf	Altbewährtes Erbe
12:00		=== Mittagspause ===

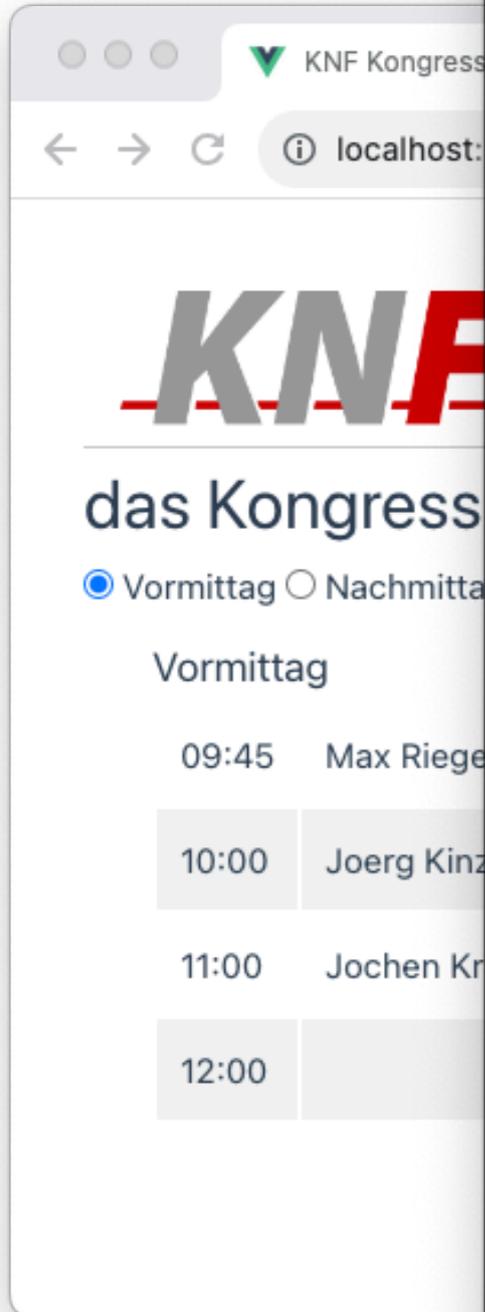
PROGRAMVIEW.VUE



```
ProgramView.vue (~/knf/src/views) - VIM
<script setup>
  import { shallowRef } from 'vue'
  import pV from '../components/pVormittag.vue'
  import pN from '../components/pNachmittag.vue'

  const activeComponent = shallowRef(pV)
</script>
<template>
  <div class="program">
    <h1>das Kongress Programm</h1>
    <label>
      <input type="radio" v-model="activeComponent" :value="pV"> Vormittag
    </label>
    <label>
      <input type="radio" v-model="activeComponent" :value="pN"> Nachmitta
    </label>
    <div class="programbox">
      <Transition name="fade" mode="out-in">
        <component :is="activeComponent"></component>
      </Transition>
    </div>
  </div>
</template>
<style>
@media (min-width: 1024px) {
1 src/views/ProgramView.vue
"src/views/ProgramView.vue" 49L, 957B
```

PROGRAMCOMPONENT

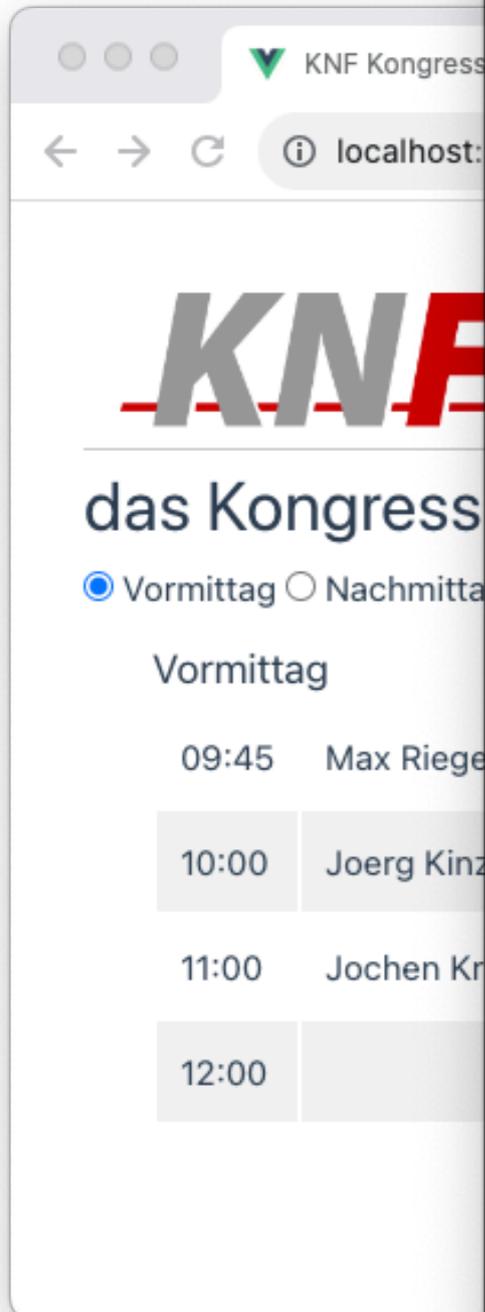


```
ProgramComponent.vue (~/knf/src/components) - VIM 3
<script setup>
  const props = defineProps({
    list:Array
  })
</script>

<template>
  <table>
    <tr v-for="item in list">
      <td>{{ item.time }}</td>
      <td>{{ item.person }}</td>
      <td>{{ item.name }}</td>
    </tr>
  </table>
</template>

<style>
th,td {
  text-align: left;
  padding: 10px;
}
tr:nth-child(even) {
  background-color: #f2f2f2;
}
</style>
~
~
~
1 src/components/ProgramComponent.vue 1
"src/components/ProgramComponent.vue" 25L, 352B
```

PVORMITTAG.VUE

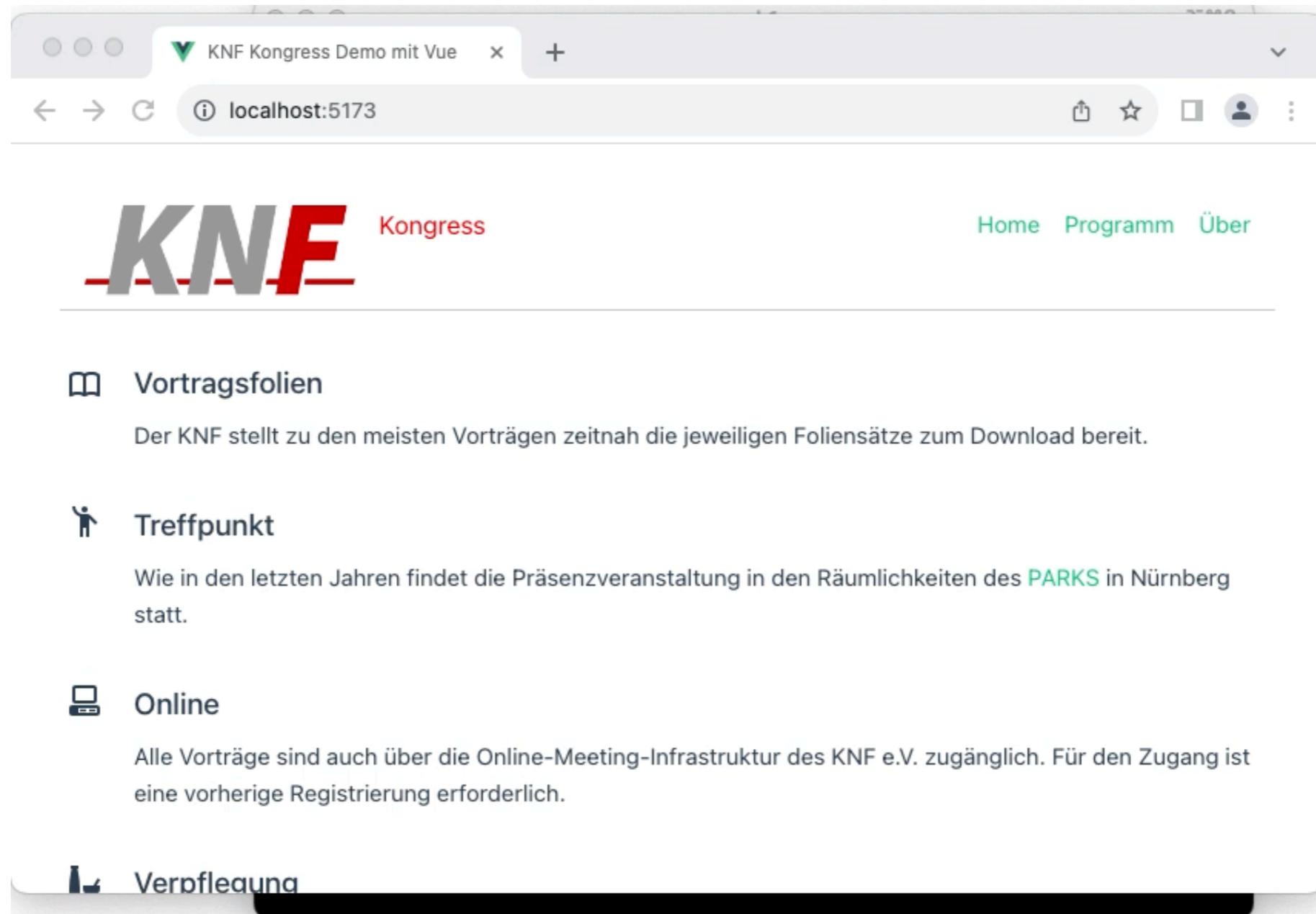


```
pVormittag.vue (~/knf/src/components) - VIM 3
<script setup>
import { ref } from 'vue'
import ProgramComponent from '../components/ProgramComponent.vue'

const vortraege = ref([
  {time: '09:45', person: 'Max Riegel', name: 'Beginn und Einführung'},
  {time: '10:00', person: 'Joerg Kinzebach', name: 'Statische Webseiten'},
  {time: '11:00', person: 'Jochen Krapf', name: 'Altbewährtes Erbe'},
  {time: '12:00', person: '', name: '== Mittagspause =='}
])
</script>

<template>
  <div>
    <h3>Vormittag</h3>
    <ProgramComponent :list=vortraege />
  </div>
</template>
~
~
~
~
~
~
~
~
~
~
1 src/components/pVormittag.vue 1
"src/components/pVormittag.vue" 19L, 524B
```

ERGEBNIS



DEPLOY/PUBLISH

The screenshot shows the Netlify team overview page for a team named 'KNF Kongress'. The page displays usage statistics for the 'Starter' plan during the current usage period from November 24 to December 23. The statistics are as follows:

Metric	Usage	Limit
Bandwidth	806 KB	100 GB
Build minutes	2	300
Concurrent builds	0	1
Team members	1	-

The interface includes a search bar, a team name 'KNF Kongress', and a navigation menu. The browser address bar shows the URL 'app.netlify.com/teams/duckiew2/overview'.

FAZIT

PERSÖNLICHES

FAZIT

- **Lernkurve entspricht nicht dem „in 5 Minuten läuft die Website“**
- **nacktes Vue erzeugt erstmal SPA und/oder SSR (fullstack)**
 - praktische zum Spielen oder für echte APP-Anwendungen, aber nicht so geschickt für Websites da nicht SEO-tauglich
- **um SSG zu erzeugen entweder Plugins nötig oder gleich ein übergeordnetes Framework**
 - vike / vite-ssg (Plugin für SSG)
 - Astro
- **VuePress**
 - Inhalte werden primär über Markdown erstellt
- **Nuxt**
 - kann per Config SSG erzeugen
 - macht Router-Konfiguration überflüssig
- **ein Grafik-Framework (bootstrap, material-design, ...) ist praktisch**
 - Integration jedoch teilweise frickelig, Plugins im Allgemeinen haben teils Abhängigkeiten, die sich beissen mit anderen Plugins oder bestimmte Versionen von Vue oder Nuxt
- **Aber man kann Hilfe im Netz finden, anders als bei selbstgestricktem!**

FRAGEN?

LINKSAMMLUNG

- Vue

<https://vuejs.org>

- Nuxt

<https://nuxt.com>

- bootstrap-vue-next

<https://github.com/bootstrap-vue-next/bootstrap-vue-next>

- VuePress

<https://vuepress.vuejs.org>

- netlify

<https://netlify.com>

- Vercel

<https://vercel.com>

- VuePress

<https://vuepress.vuejs.org>

BEGRIFFSERKLÄRUNGEN

Vite	lokaler Development Server, genutzt vom Vue und React Projekt. Hat eingebauten Support für SSR.	Hydrate	Eine Web-Applikation lädt dynamisch Inhalte nach um sich selbst korrekt und später neue Inhalte darstellen zu können
SSR	Server-side Rendering. Die Ansicht einer Anwendung wird auf dem Server gerendert und an den Browser ausgeliefert	Router	Element einer SPA, die ein Mapping von Sub-URLs auf Views/Pages herstellt
SPA	Single-Page Application. Eine Web-Anwendung die interaktiv mit dem Nutzer agiert und dabei eine einzelne Webseite mit neuen Daten füllt und neu rendert.	TypeScript	Erweiterung von JavaScript
SSG	Server Side Generation. Eine bereits Vorgenerierte „statische“ Seite erzeugt die Web-Anwendung, alle Clients laden das gleiche bereits erzeugte HTML und Javascript und führen dann den Hydrate durch	SASS	<u>S</u> yntactically <u>A</u> wesome <u>S</u> tyles <u>h</u> eets, ein CSS-Präprozessor mit Variablen, Schleifen und Erweiterungen, die CSS nicht selbst beinhalten
MPA	Multi-Page Application. Eine klassische Web-Anwendung, die von Seite zu Seite springt um Inhalte nachzuladen.	SCSS	SassyCSS, eine Erweiterung bzw. verbesserte Version von SASS
SEO	Search Engine Optimization	CDN	Content Delivery Network/Content Distribution Network - ein meist regional verteilter Auslieferungsdienst von Inhalten

KNF KONGRESS DEMO

```
web1 ㄿ#3
user@web1[~] 14>git clone https://github.com/duckiew2/kongress.git
Cloning into 'kongress'...
remote: Enumerating objects: 76, done.
remote: Counting objects: 100% (76/76), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 76 (delta 23), reused 68 (delta 15), pack-reused 0
Receiving objects: 100% (76/76), 54.12 KiB | 2.35 MiB/s, done.
Resolving deltas: 100% (23/23), done.
user@web1[~] 15>cd kongress
user@web1[~/kongress] 16>npm install

added 281 packages, and audited 282 packages in 4s

76 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 9.6.3 -> 10.2.4
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.4
npm notice Run npm install -g npm@10.2.4 to update!
npm notice
user@web1[~/kongress] 17>npm run dev
```

■ GIT repository:

<https://github.com/duckiew2/kongress.git>

■ Demo auf netlify:

<https://kongress2023.netlify.app/>