

# Projekt: Optimierung einer VueJS-Web-App

Andreas Nicklaus

Web Performance Optimierung

20.01.2024

[web-performance.andreasnicklaus.de](http://web-performance.andreasnicklaus.de)



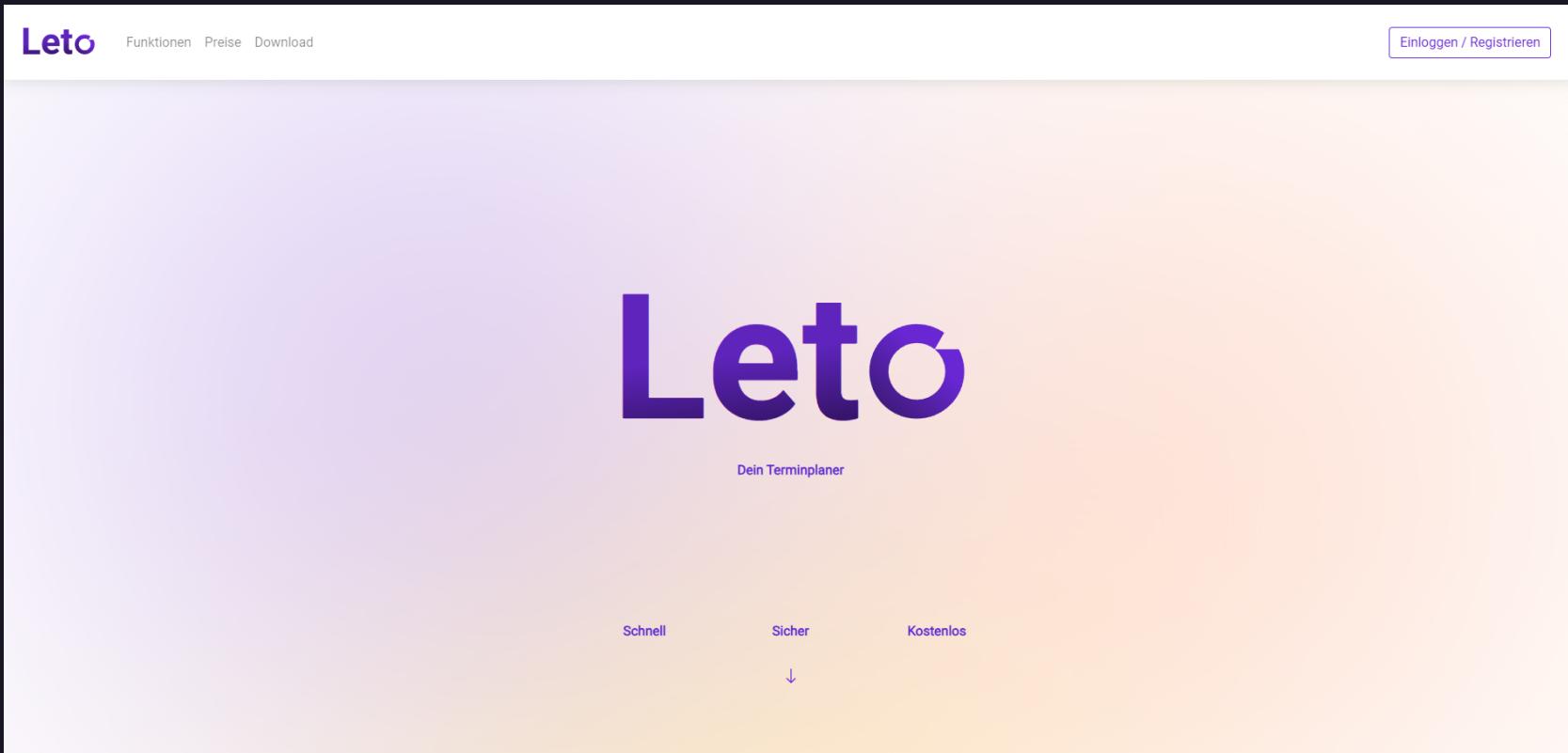
# Agenda

1. Probleme der Webseite
2. Gemachte Änderungen
3. Ergebnisse
4. Was fehlt noch?



# Die Webseite

<https://leto.andreasnicklaus.de>



# Infos zur Seite

- Marketing-Webseite für Desktopanwendung
- 6 animierte Elemente in der Startansicht
- 2 PNG-Bilder auf der Startseite
- Development:
  -  VueJS
  -  Bootstrap-Vue
  -  Vuetify
- Deployment:
  -  AWS EC2
  -  Docker
  -  Nginx Proxy Manager
  -  Nginx



# Seitenstruktur

- 9 statische & indexierte Seiten
- 3 dynamische & nicht-indexierte Seiten
  - Profilansicht
  - Admin-Dashboard
  - Checkout



# Probleme der Webseite



Time To Interactive	4,6s
Largest Contentful Paint	5,2s
Lighthouse Performance	33
Page Weight	2,789 MB



Page Weight	Bytes
Bilder	2 099 825
JS	471 052
CSS	221 358
Fonts	15 744
HTML	515
<hr/> <b>Gesamt</b>	<b>~2,7 MB</b>



# Änderungen 0-3

## **Version Änderungen**

Original -

v01 Prerendering für statische Seiten

v02 Render-Blocking Stylesheets entfernt

v03 Updated Docker Build Workflow



# Änderungen 4-7

## **Version Änderungen**

- v04      Verbessertes Access Management
- v05      AVIF-Bilder, Chained Request für Fonts entfernt
- v06      Lineare Verteilung der Bildgrößen
- v07      Import nur der genutzten Icons anstelle von allen

# Änderungen 8-11

## **Version Änderungen**

v08 JS-Chunks gesplittet

v09 Animationen entfernt

v10 Alle Bilder werden lazy-loaded

v11 Zuerst sichtbare SVGs werden lazy-loaded

# v01: Prerendering

```
new PrerenderSpaPlugin({
  staticDir: path.join(__dirname, 'dist'),
  routes: routes.filter(r => r.meta?.prerender).map(r => r.path),
  renderer: new PrerenderSpaPlugin.PuppeteerRenderer({
    inject: {},
    renderAfterElementExists: '[data-view]',
  }),
  postProcess: (renderedRoute) => {
    renderedRoute.html = renderedRoute.html
      .replace(/<script\ (.*)?>/g, '<script $1 defer>')
      .replace('id="app"', 'id="app" data-server-rendered="true"');

    return renderedRoute;
  }
})
```



# v01: Prerendering

	<b>Original</b>	<b>v01</b>
Time To Interactive	4,6s	14,5s
Largest Contentful Paint	5,2s	3,3s
First Contentful Paint	4,6s	3,3s
Lighthouse Performance	33	57

# v02: Renderblocking-Stylesheets entfernt

```
postProcess: (renderedRoute) => {
  renderedRoute.html = renderedRoute.html
    .replace(
      /<link href="(.*?)" rel="stylesheet"/g,
      `<link rel="preload" href="$1" as="style" onload="this.onload=null;this.rel='stylesheet'>
<noscript>
  <link rel="stylesheet" href="$1">
</noscript>`
    )
    .replace(
      /<link rel="stylesheet" (.*?)>/g,
      `<link rel="preload" $1 as="style" onload="this.onload=null;this.rel='stylesheet'>
<noscript>
  <link rel="stylesheet" $1>
</noscript>`
    )
    .replace(/<script (.*?)>/g, '<script $1 defer>')
    .replace('id="app"', 'id="app" data-server-rendered="true"');

  return renderedRoute;
}
```



# v02: Renderblocking-Stylesheets entfernt

	<b>Original</b>	<b>v01</b>	<b>v02</b>
<b>Time To Interactive</b>	<b>4,6s</b>	14,5s	10,4s
<b>First Contentful Paint</b>	4,6s	3,3s	<b>0,9s</b>
<b>Largest Contentful Paint</b>	5,2s	3,3s	<b>2,6s</b>
<b>Lighthouse Performance</b>	33	<b>57</b>	45

# v05: Chained Request für Fonts entfernt

Bevor in `App.vue`:

```
<style lang="scss">
  @import url("https://fonts.googleapis.com/css2?family=Roboto&display=swap");
</style>
```

Danach in `index.html`:

```
<link
  rel="stylesheet"
  href="https://fonts.googleapis.com/css2?family=Roboto&display=swap"
/>
```



# v05: AVIF-Bilder (alt)

```

```

# v05: AVIF-Bilder (neu)

```
<picture>
  <source
    :srcset="`${appleDevices_avif_1} 200w, ${appleDevices_avif_2} 783w,
              ${appleDevices_avif_3} 1123w, ${appleDevices_avif} 1920w`"
    sizes="(max-width: 768px) 100vw, 50vw"
  />
  <source
    :srcset="`${appleDevices_webp_1} 200w, ${appleDevices_webp_2} 783w,
              ${appleDevices_webp_3} 1123w, ${appleDevices_webp} 1920w`"
    sizes="(max-width: 768px) 100vw, 50vw"
  />
  
</picture>
```



# v05: Format Beispiel

- PNG: **611 kB**  
(nicht verwendet)
- WEBP:
  - **126 kB** (Original)
  - **112 kB** (klein)
- AVIF:
  - **374 kB** (Original)
  - **48 kB** (klein)



# v05: Format Beispiel

	<b>Original</b>	<b>v04</b>	<b>v05</b>
<b>Time To Interactive</b>	<b>4,6s</b>	13,3s	6,1s
<b>Speed Index</b>	7,5s	4,1s	<b>2,5s</b>
<b>Page Weight</b>	2,789 MB	2,738 MB	<b>1,140 MB</b>
<b>Cumulative Layout Shift</b>	<b>0,751</b>	0,936	0,780

# v06: Lineare Verteilung der Bildgrößen

vorher:

```
<source
  :srcset="`${appleDevices_avif} 1920w, ${appleDevices_avif_3} 1123w,
${appleDevices_avif_2} 783w, ${appleDevices_avif_1} 200w`"
  sizes="(max-width: 768px) 100vw, 41.67vw"
/>
```

nachher:

```
<source
  :srcset="`${appleDevices_avif} 1920w, ${appleDevices_avif_1} 1600w,
${appleDevices_avif_2} 1280w, ${appleDevices_avif_3} 960w,
${appleDevices_avif_4} 640w, ${appleDevices_avif_5} 320w`"
  sizes="(max-width: 768px) 100vw, 41.67vw"
/>
```



# v07: Import nur der genutzten Icons

```
import { BootstrapVueIcons } from 'bootstrap-vue'  
Vue.use(BootstrapVueIcons)
```

```
import { BootstrapVue, BIIconBoxArrowUpRight, BIIconPerson, ... } from 'bootstrap-vue'  
Vue.component("b-icon-box-arrow-up-right", BIIconBoxArrowUpRight)  
Vue.component("b-icon-person", BIIconPerson)  
...
```



# v08: JS-Chunks gesplittet

```
configureWebpack: (config) => {
  config.optimization = {
    runtimeChunk: 'single',
    splitChunks: {
      chunks: 'all',
      maxInitialRequests: Infinity,
      maxSize: 500_000,
    }
  }
}
```



# v08: JS-Chunks gesplittet

	<b>v06</b>	<b>v07</b>	<b>v08</b>
<b>Anzahl JS-Dateien</b>	30	30	<b>62</b>
<b>Raw Größe JS-Dateien</b>	5,75 MB	<b>4,95 MB</b>	<b>4,95 MB</b>
<b>Page Weight</b>	778 kB	<b>612 kB</b>	628 kB



# v09: Animationen entfernt

mit Animationen (um 1100ms verzögert)

```
<div  
  data-aos="fade-up"  
  data-aos-delay="1100"  
  data-aos-anchor-placement="bottom"  
>  
  <b-button><b-icon-arrow-down variant="primary"/></b-button>  
</div>
```

ohne Animationen:

```
<div>  
  <b-button><b-icon-arrow-down variant="primary"/></b-button>  
</div>
```



# Alle Bilder (v10) & SVGs (v11) werden lazy-loaded

v10:

```
<picture>
  <source .../>
  <source .../>
  <img
    ...
    loading="lazy"
  />
</picture>
```

v11:

```
<img
  ...
  loading="lazy"
/>
```



# v10 & v11: Lazy-loading

**v09    v10    v11**

**Lighthouse Performance**                    **54    44    54**

**Speed Index**                                **2,6s    3,9s    2,5s**

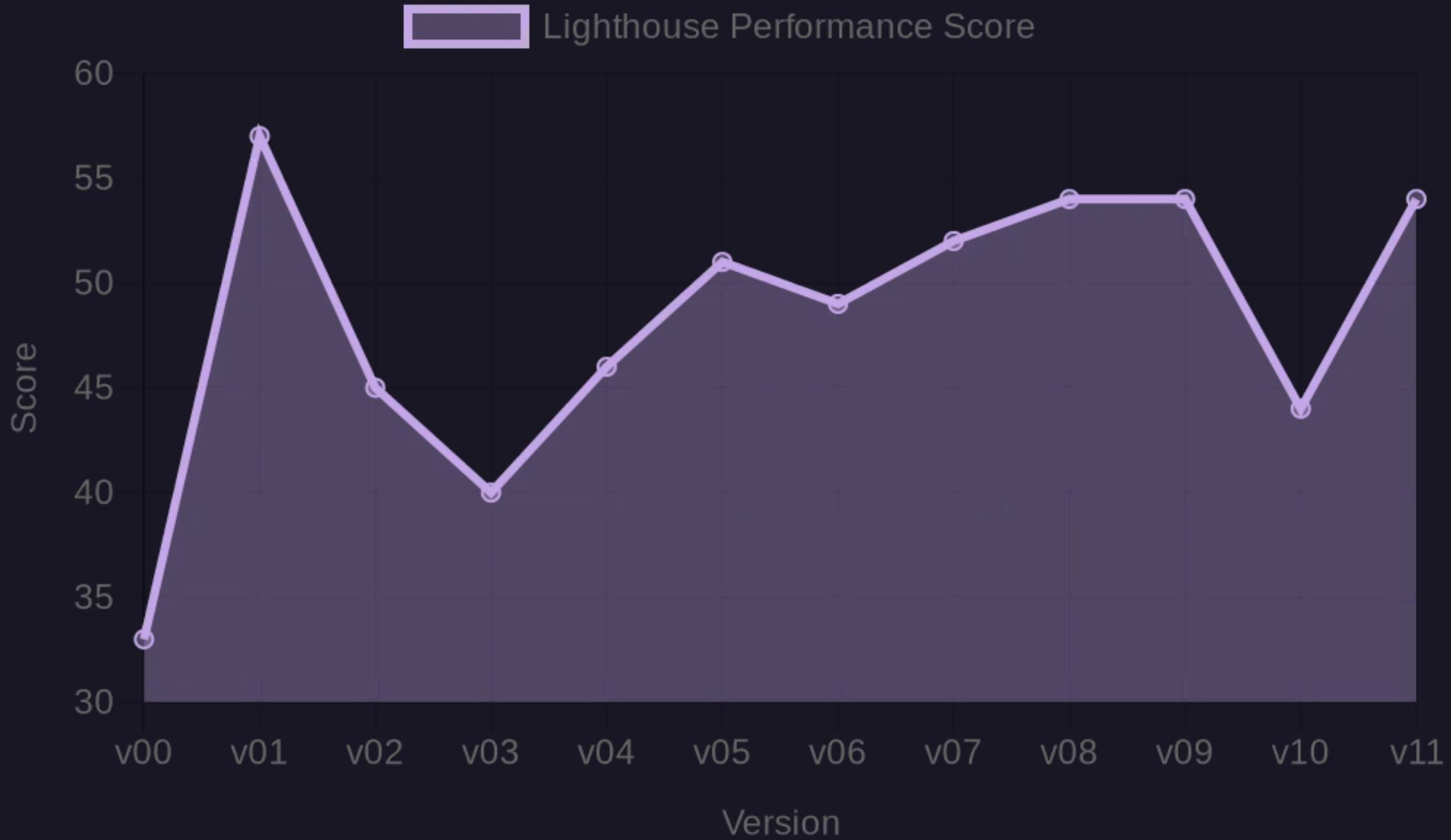
**Largest Contentful Paint**                    **2,7s    4,1s    2,6s**



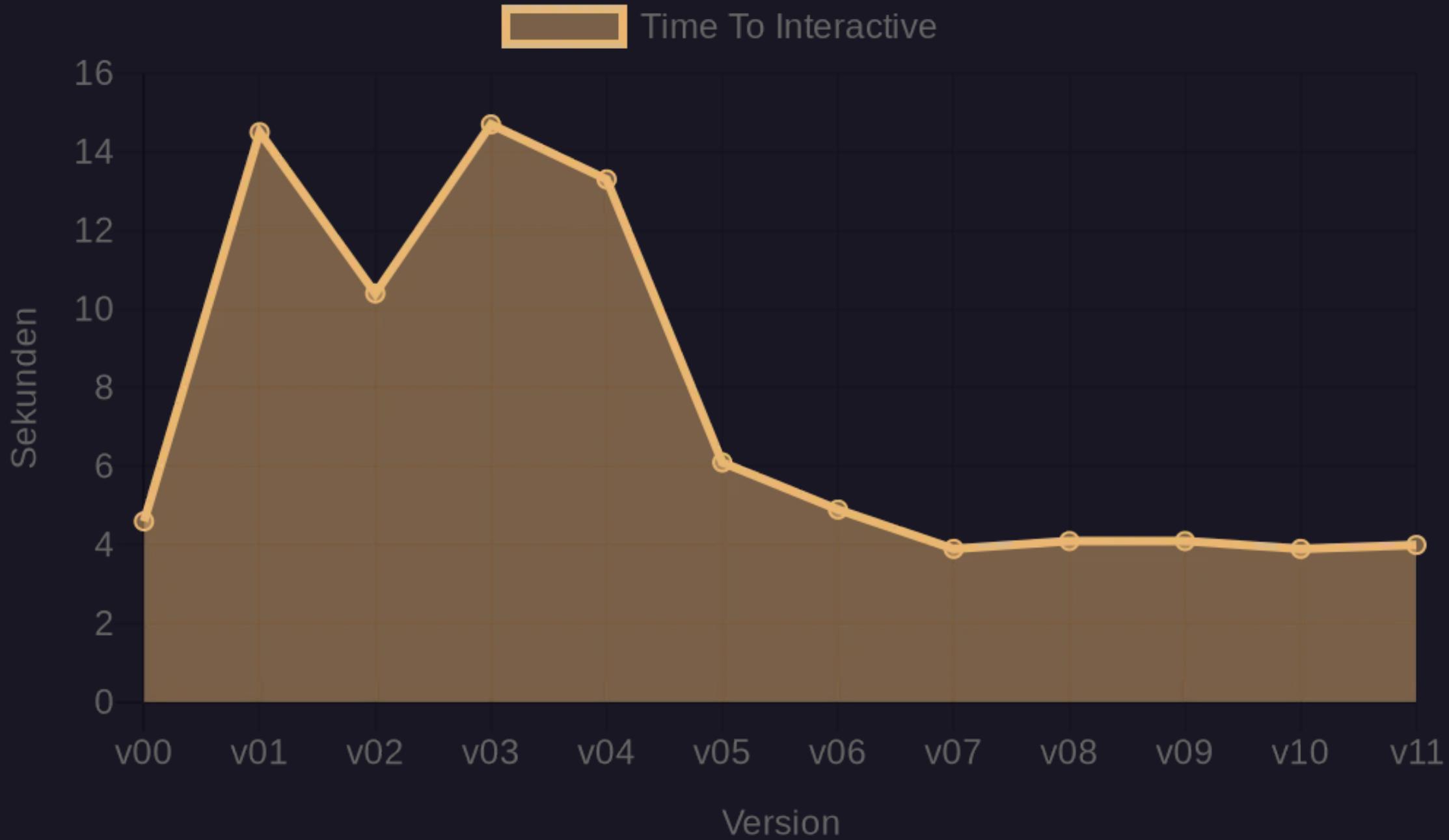
# Ergebnisse - Visualisierungen

Lighthouse Performance	Score
Original	33
v01: Prerendering	57
v11: Lazy-Loading	54



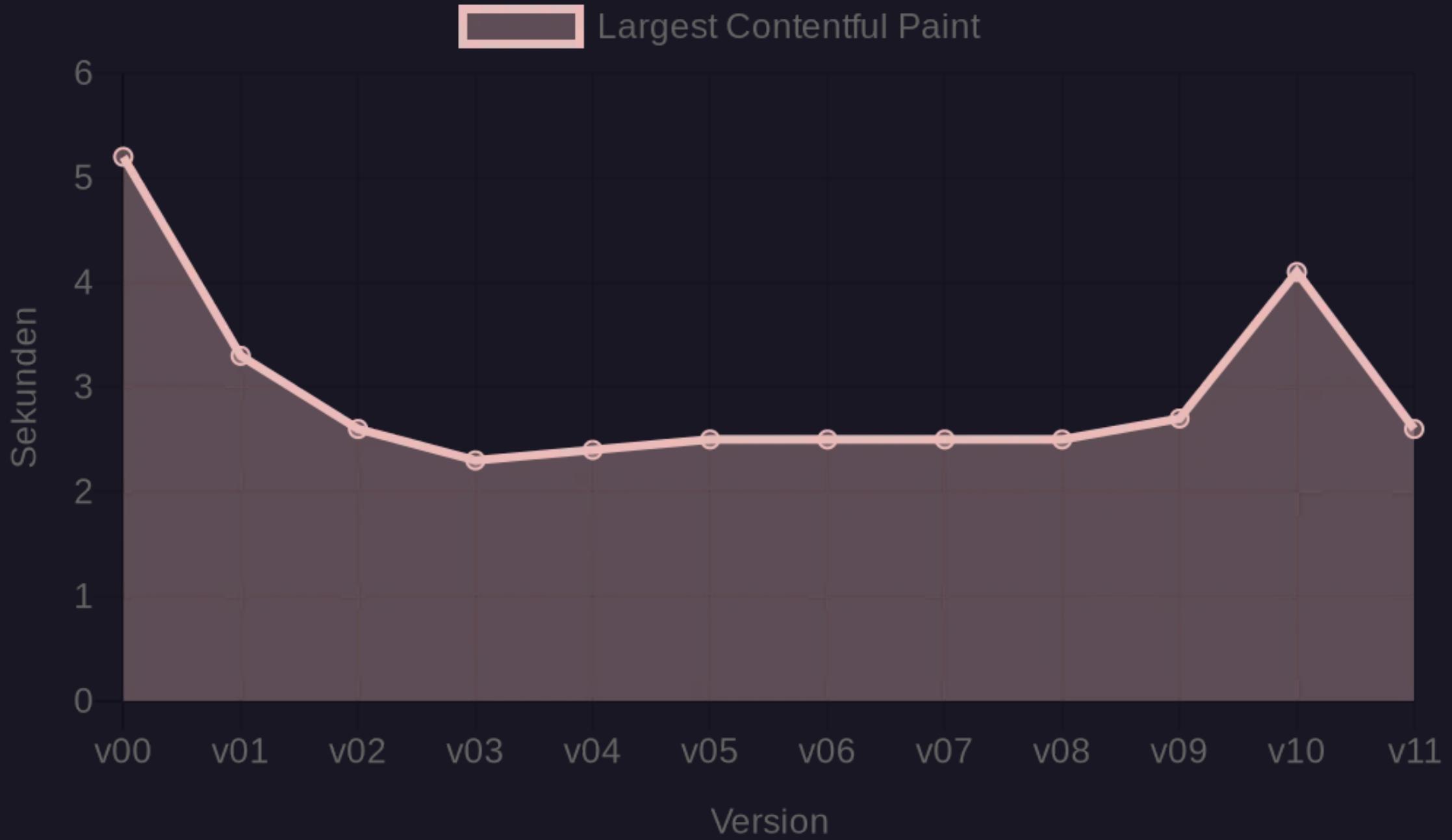


<b>Time To Interactive</b>	<b>Score</b>
Original	4,6s
v10: lazy-loaded Images	3,9s



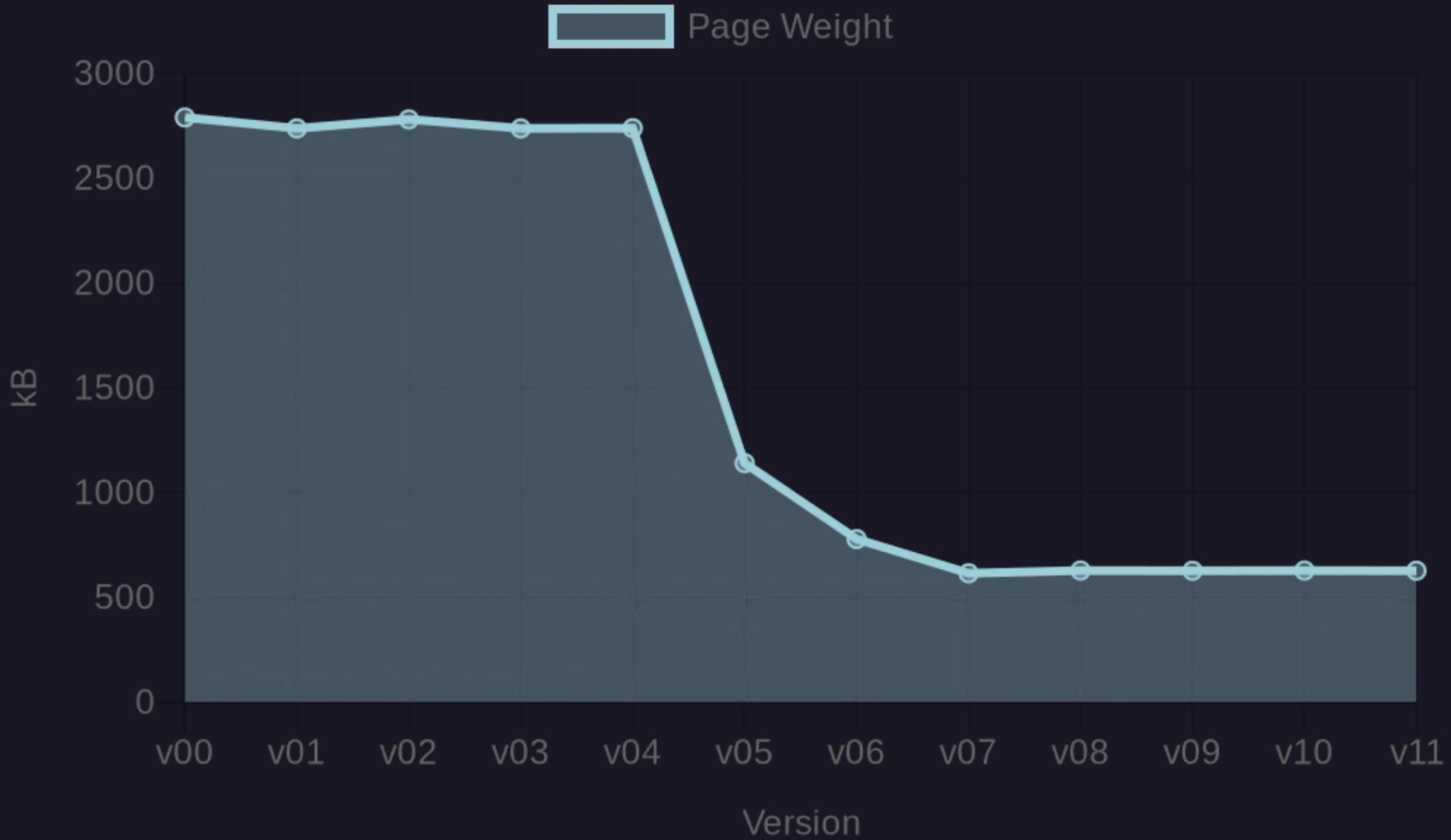
Largest Contentful Paint	Score
Original	5,2s
v05: AVIF, Chained Requests	2,5s
v11: lazy-loaded SVGs	2,6s





<b>Page Weight</b>	<b>Bytes</b>
Original	2789 kB
v05: Bilder im AVIF-Format	-1649 kB
v06: lineare Bilder-Größenverteilung	-362 kB
v07: Importiere nur genutzte Icons	-166 kB
<hr/> <b>Gesamt</b>	<b>612 kB</b>





# Was fehlt noch?

Nach dem Lighthouse Performance Report:

- CSS Pruning
- Automatisiertes Treeshaking
- "Minimize main thread work"
- Preloading LCP-Bild
- Effizienteres Caching mit CDN

Nach Image Linter:

- Größere Bildversionen für große Bildschirme



# Das wars!

Versions-URLs: vXX.leto.andreasnicklaus.de

# **Andreas Nicklaus**



@andreasnicklaus



andreasnicklaus



andreasnicklaus.de