

The background features a dark gradient with several vertical, slightly wavy light streaks in shades of red, orange, and yellow. Scattered across the background are several circles of different sizes and colors, including green, orange, pink, and yellow.

# Softwareengineering Demo

Maximilian Simon, Steven Wahl, Nicolas Maiwald, Leon  
Andreesen

# Gliederung

---

Selling Points

---

Architektur-Design

---

Gesamtüberblick Projektstruktur

---

Techstack Backend & Frontend

---

CI/CD

---

Metrics

---

Lessons learned

# Selling Point



Einzigartige Spielekombination:  
Integration von BlackJack, Snake und  
CookieClicker auf einer Website



Benutzerfreundlichkeit: Intuitive  
Benutzeroberfläche und einfaches  
Spielprinzip für alle Altersgruppen



Revolutionäre Spiele: klassische Spiele  
erhalten einen neuen Anstrich. So zum  
Beispiel das neue SnakeExtreme

# Architektur-Design



REST-API Wahl: Entscheidung für eine REST-API zur Trennung von Frontend und Backend, was die Entwicklung und Wartung erleichtert



Skalierbarkeit: REST-API unterstützen horizontale Skalierung, was bei wachsender Nutzerzahl wichtig ist

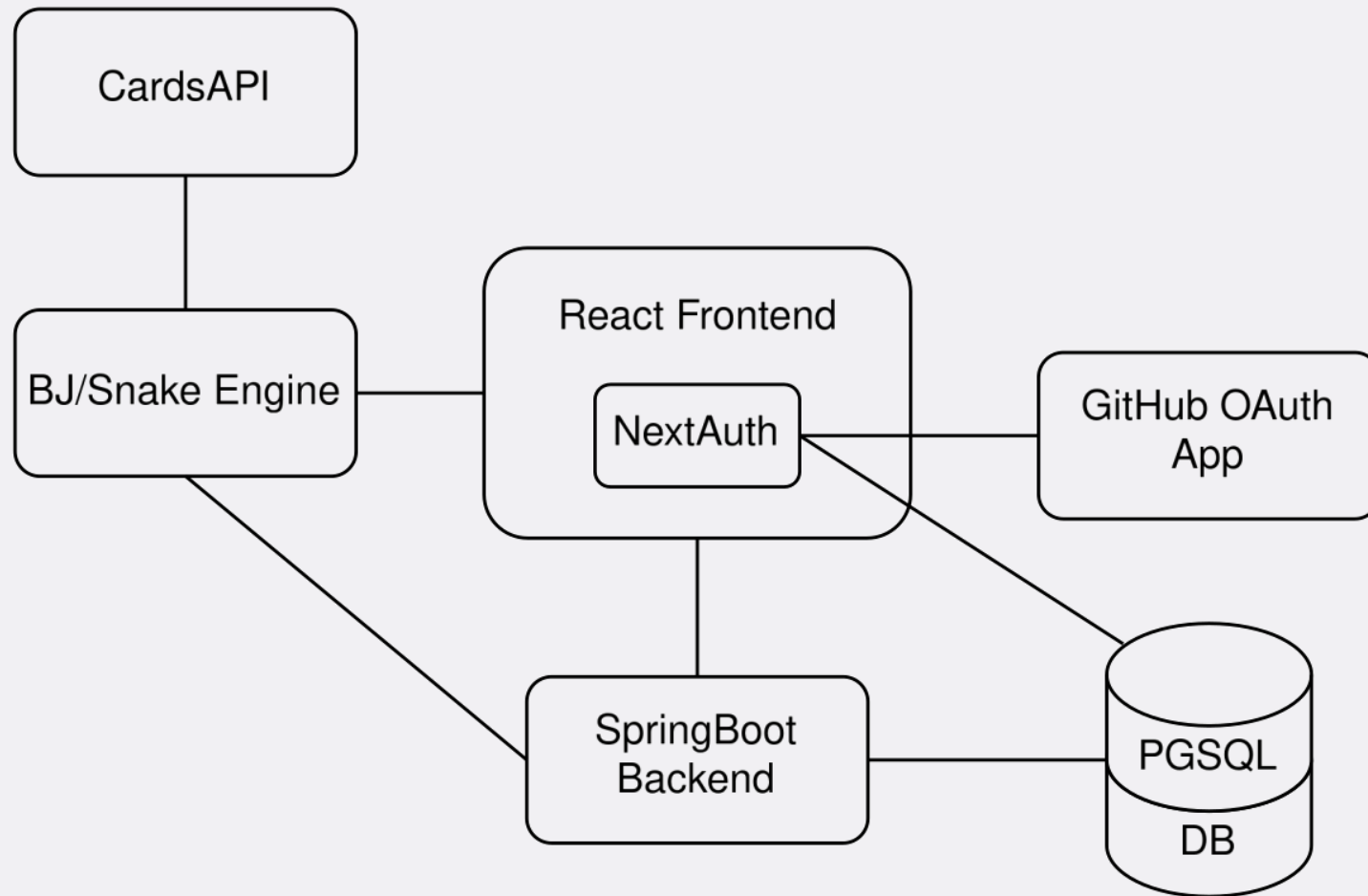


Plattformunabhängigkeit: REST-API ermöglichen eine einfache Integration und Kommunikation zwischen verschiedenen Plattformen und Geräten



Sicherheit: Durch den Einsatz von HTTPS und OAuth2 für die Authentifizierung wird die Sicherheit erhöht

# Gesamtüberblick



# Tech Stack: Backend

Verwendung von  
Spring Boot (Java)  
für die Backend-  
Entwicklung,  
gebaut mit Gradle

Einsatz einer  
PostgreSQL-  
Datenbank zur  
Speicherung der  
Spieldaten

Einsatz von  
Docker zur  
Containerisierung  
auf einer  
Debian12-Instanz  
bei DigitalOcean

Cloudflare für  
Domain, Reverse  
Proxy  
und Analytics

# Spring Boot Backend

- Speichern von Highscores in der Datenbank (Nach jedem Spielende)
- Laden von Highscores (für die Leaderboards)

POST	/highscores	Save a new highscore in the db	🔒 ▼
GET	/highscores	Find highscores by username and game	🔒 ▼
GET	/highscores/games/{game}	Get highscores by specific game	🔒 ▼
GET	/highscores/users/{username}	Get highscores by username	🔒 ▼
GET	/highscores/secret	Returns 42	🔒 ▼

# Cloudflare

- Managen von Subdomains für die verschiedenen Container (Frontend, Backend, Engines)
- Reverse Proxy hinter der Domain sorgt für Sicherheit und Robustheit
- Vielzahl an Traffic Analytics (Unique Visitor IPs, Anzahl an Requests, User-Herkunft, etc.)





# Tech Stack: Frontend

Pixi. JS OpenGL  
Renderer: Verwendet  
für die Darstellung der  
Grafiken in den Spielen

HTML und CSS:  
Grundlegende  
Technologien zur  
Gestaltung und  
Strukturierung der  
Webinhalte

React und Next.JS:  
Genutzt für die  
Entwicklung der  
Website und des  
Cookie Clickers

NextAuth: OAuth2-  
basierte  
Authentifizierung via  
GitHub

# Cards API

- DeckofCards-API Funktion:

- Kartenrückseite anzeigen
- Neues Deck erstellen
- Karten ziehen
- Karten mischen
- Spezielles Deck erstellen
- Ablagestapel erstellen
- Ablagestapel mischen
- Vom Ablagestapel ziehen

# Cards API

- Kartenrückseite anzeigen
  - <https://www.deckofcardsapi.com/static/img/back.png>
  - Statischer API Aufruf um Kartenrückseite darzustellen
- Neues Deck erstellen
  - [https://www.deckofcardsapi.com/api/deck/new/shuffle/?deck\\_count=6](https://www.deckofcardsapi.com/api/deck/new/shuffle/?deck_count=6)
  - Shuffle mischt das erstellte Deck
  - Deck\_count gibt die Anzahl der verwendeten Decks an
  - Gibt deck\_id zurück
- Karten ziehen
  - [https://www.deckofcardsapi.com/api/deck/<<deck\\_id>>/draw/?count=2](https://www.deckofcardsapi.com/api/deck/<<deck_id>>/draw/?count=2)
  - Deck\_id gibt das Deck an aus dem gezogen wird
  - Count gibt die Anzahl der gezogenen Karten an
  - Gibt Karten als JSON zurück

# Cards API

```
{
  "success": true,
  "deck_id": "kxozasf3edqu",
  "cards": [
    {
      "code": "6H",
      "image": "https://deckofcardsapi.com/static/img/6H.png",
      "images": {
        "svg": "https://deckofcardsapi.com/static/img/6H.svg",
        "png": "https://deckofcardsapi.com/static/img/6H.png"
      },
      "value": "6",
      "suit": "HEARTS"
    },
    {
      "code": "5S",
      "image": "https://deckofcardsapi.com/static/img/5S.png",
      "images": {
        "svg": "https://deckofcardsapi.com/static/img/5S.svg",
        "png": "https://deckofcardsapi.com/static/img/5S.png"
      },
      "value": "5",
      "suit": "SPADES"
    }
  ],
  "remaining": 50
}
```



# PixiJS

- Großes JavaScript Modul um vorgefertigte Spielbausteine zu nutzen
- Erstellen und darstellen von Sprite-Objekten
- Implementierung des Black Jack und Snake Spiels
- Genaueres in der Live Demo



# React Frontend

- React-Frontend für die Overview/Games/Stats/Profil Seiten
- Page-Routing mit Next.JS für intuitive Strukturierung des Codes
- Implementierung des Röhrig-Clickers
- NextAuth-Integration für den Login

```

└─ app
  └─ (auth)
    └─ profile
      └─ page.tsx
  └─ (default)
    └─ page.tsx
  └─ (games)
    └─ games
      └─ blackjack
        └─ karten
          └─ page.tsx
          └─ page.tsx
        └─ roehrig
          └─ page.tsx
        └─ roehrigclicker
          └─ page.tsx
        └─ snake
          └─ page.tsx
          └─ page.tsx
  └─ (stats)
    └─ stats
      └─ page.tsx
  └─ [...not_found]
    └─ page.tsx
  > api
  > css
  └─ impressum
    └─ page.tsx
    └─ layout.tsx
  > components
```

# Next Auth

- Ver
- Us
- Da



Auth flow



Sign in with GitHub

User logs in  
via GitHub

TIME2SHINE

[Home](#) [Games](#) [Stats](#) [My Profile](#)

[Sign out](#)

## Profile Page



Name: Tech.NIC

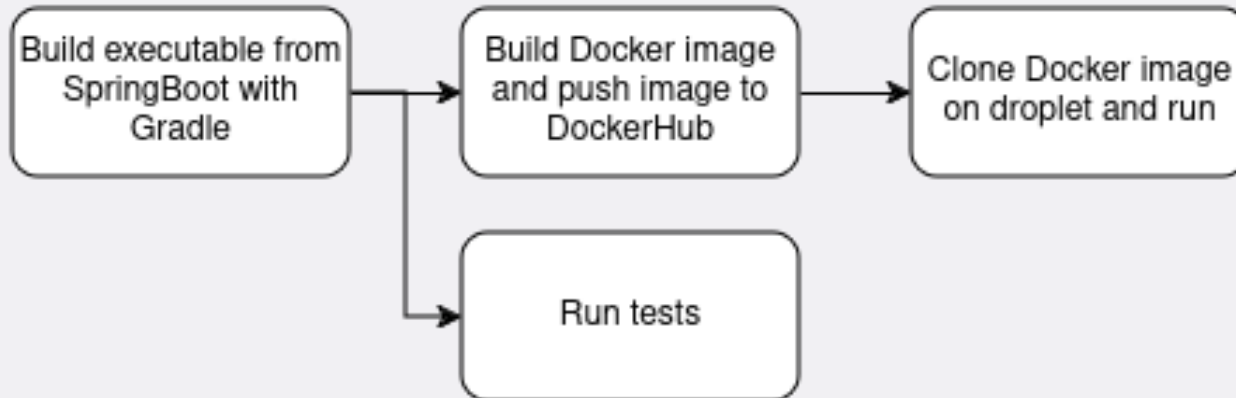
Email: nicolasward@gmail.com

User gets redirected  
to t2s w/o needing  
separate credentials

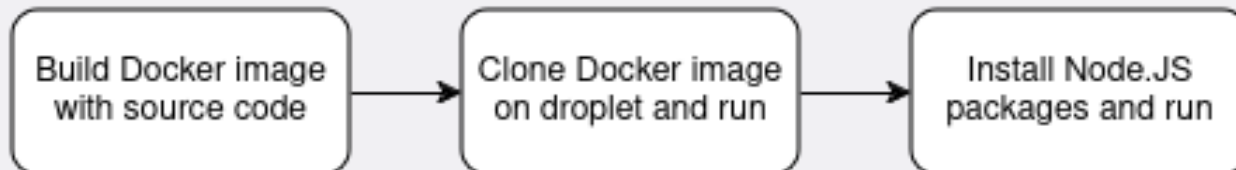
# CI/CD-Pipelines

- Automatisierte Builds der Komponenten
- Deployment in Docker Containern

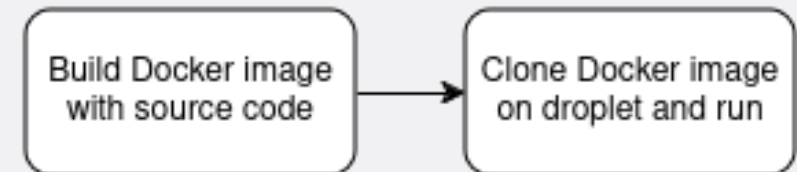
## Backend Pipeline



## Frontend Pipeline



## Engine Pipeline





# Metrics

Performance grade

**B 88**

Page size

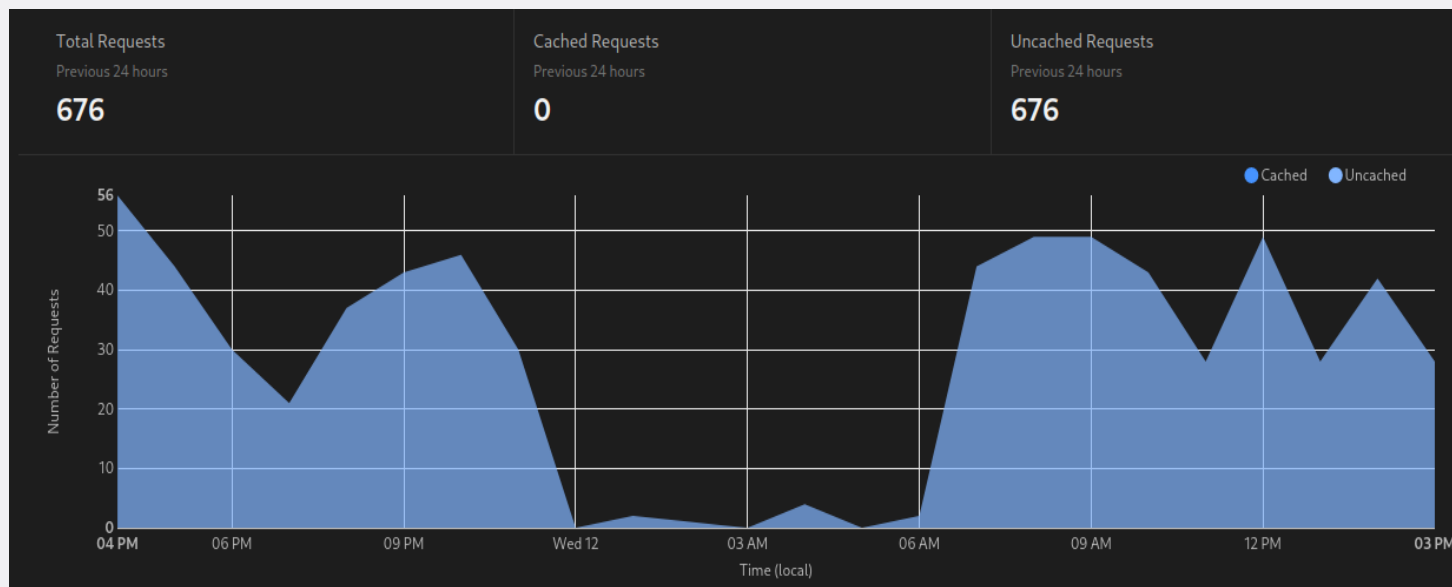
**2.0 MB**

Load time

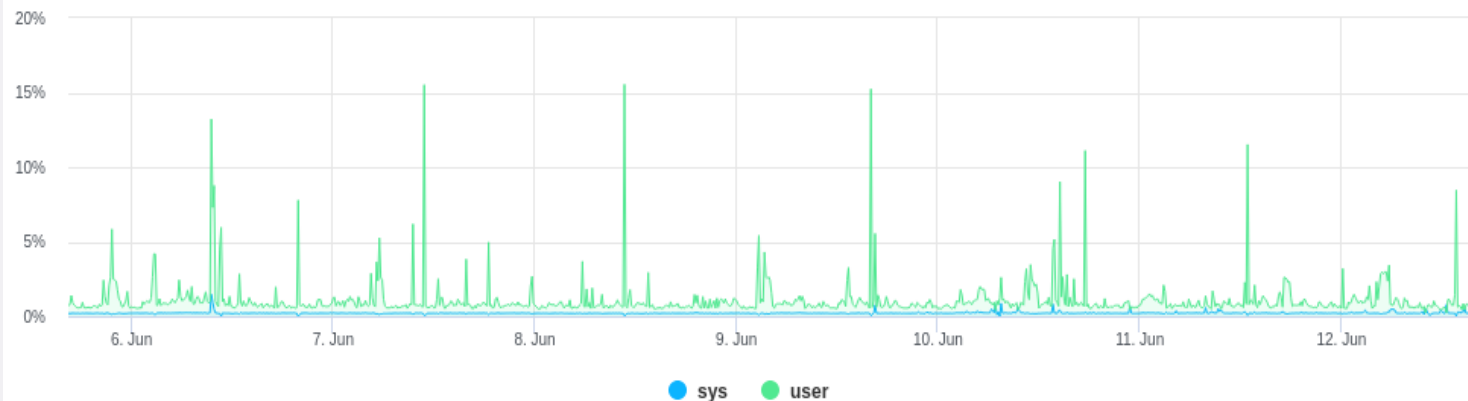
**593 ms**

Requests

**17**



## CPU Usage



# Metrics: Page Load Time

Performance grade

**B 88**

Page size

**2.0 MB**

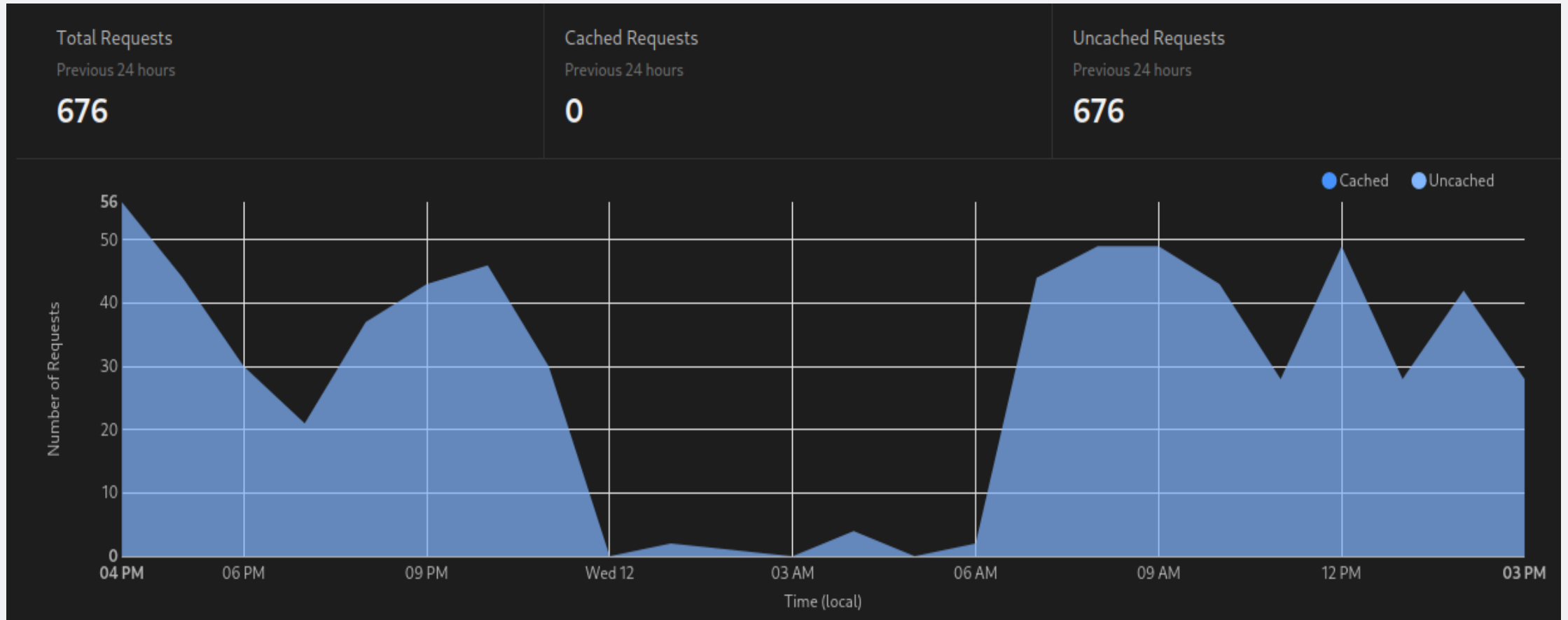
Load time

**593 ms**

Requests

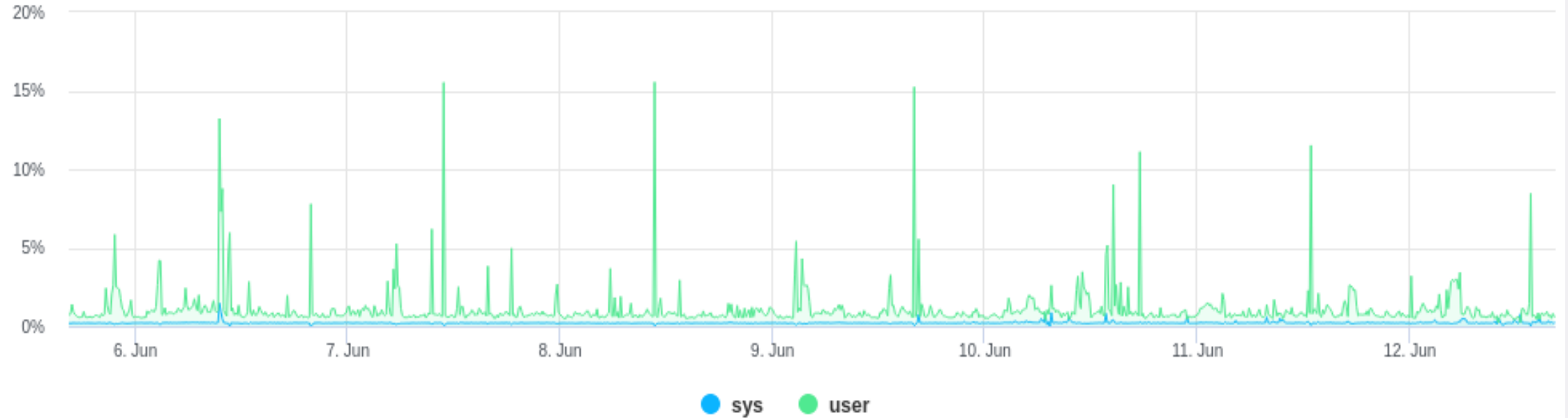
**17**

# Metrics: User traffic



# Metrics: Server load

CPU Usage



# Lessons learned



UMGANG MIT  
GIT(HUB)



MOCKUPS ERSTELLEN



OAUTH2 FLOW

# time2shine games.

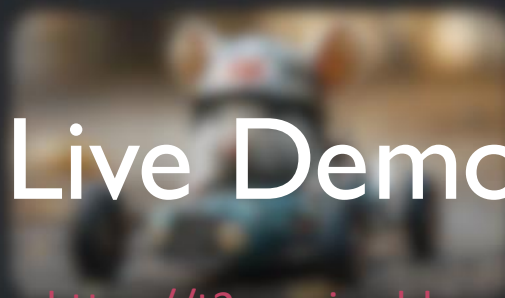
An overview of our current and upcoming games.



Blindspot is a first-person shooter that  
against the odds, where you  
use a magic hand of cards to  
conquer it and strike gold!

[Play Blindspot](#)

[Learn more](#)



In "Racing Chase", become  
Professor Wilson's experimental  
subject, frantically pushing every  
one else at a time!

[Play Racing Chase](#)

[Learn more](#)



In Snake Extreme, crank up the  
speed and outsmart your own  
lengthening serpent, getting  
past the point for a high score  
bounty!

[Play Snake Extreme](#)

[Learn more](#)

## Live Demo

<https://t2s.maiwald.cc>

**DANKE FÜR EURE AUFMERKSAMKEIT**

