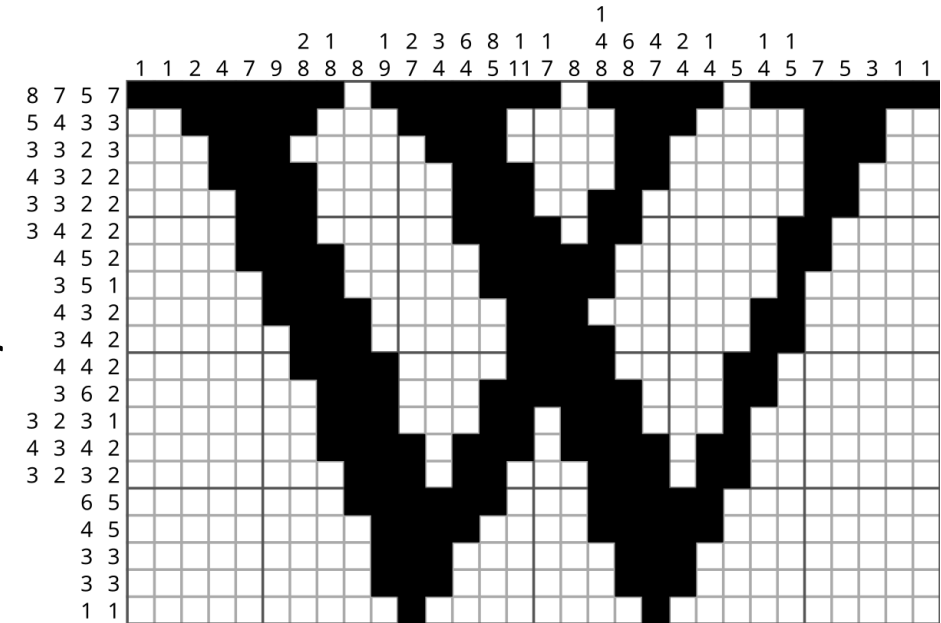


Nono*Rush*

Patrick Brander, Silas Schmid, Quentin Weber

Projektidee

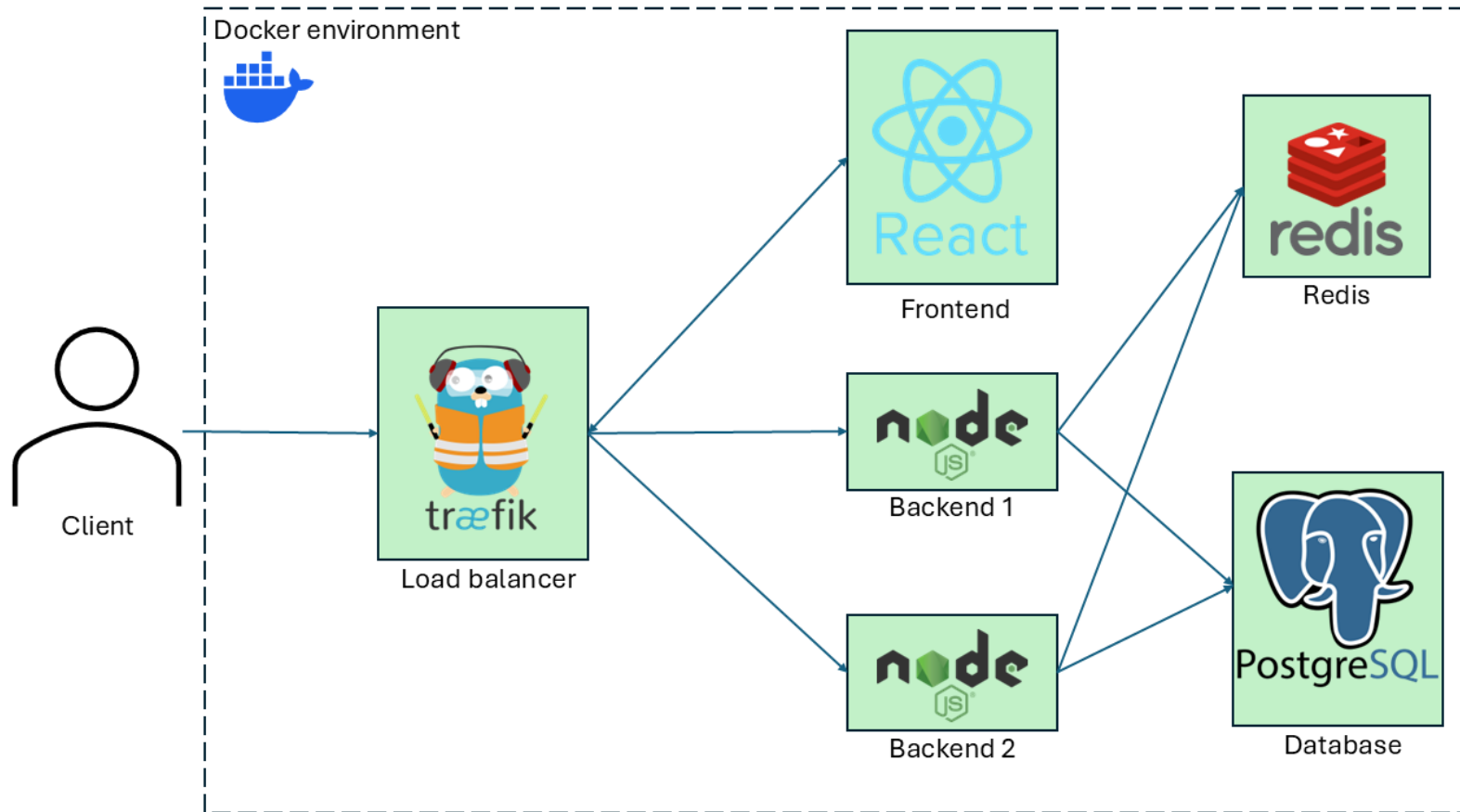
- Nonogram
 - Puzzle game mit einem Pixel Grid
 - Zahlen am Rand des Grids geben die Anzahl der befüllten Pixel an
 - Vollständig gelöst ist meistens ein Motiv erkennbar
- Idee
 - Nonogram – aber gegeneinander
 - Zwei oder mehr Spieler versuchen dasselbe Puzzle als erster zu lösen



Ablauf

- Host erstellt Lobby
- Spieler treten der Lobby per code/link bei
- Host startet Runde
- Alle Spieler erhalten dasselbe Puzzle
- Wenn ein Spieler fertig ist, endet die Runde

Architektur



Redis

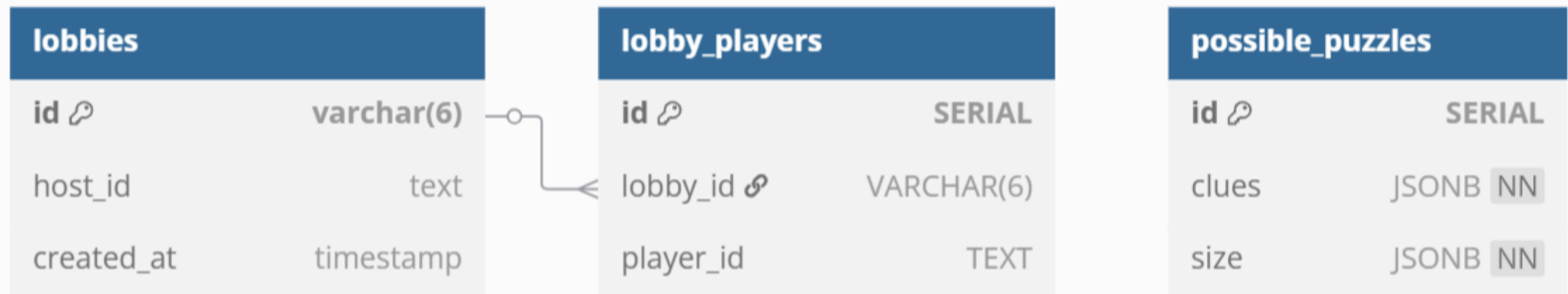


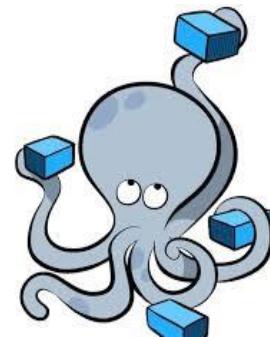
- Eigentlich in-memory database (cache)
- Bei uns nur als message broker für SSE
 - Clients abonnieren ihre Lobby mit `redis.subscribe()`
 - Lobbies senden Events an den Client mit `redis.publish()`
 - Ermöglicht Kommunikation zwischen den beiden Backends

Datenbank



- Drei Tabellen für Spieler, Lobbies, und Puzzles
- Einige Standardpuzzles werden beim Erstellen der Datenbank eingefügt





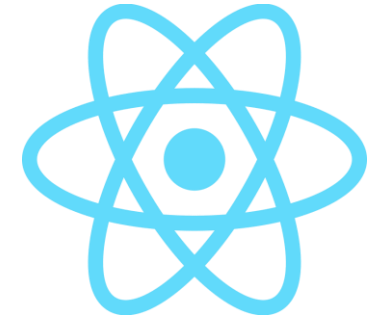
Docker Compose

- Dockerized Umgebung
- Eigener Container für jeden Service

```
docker compose -f docker-compose.yml up --scale backend=2
```

☐	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
☐	▼ ☒ <u>dsysprojekt</u>	-	-	-	148.62%	1 second ago	☒ ⋮ <input type="trash"/>
☐	☒ dsys-traefik	06bcc1076626	traefik:late	80:80 ↗ Show all ports (2)	0%	2 seconds ago	☒ ⋮ <input type="trash"/>
☐	☒ redis	50ffffabe8be	redis:7-alpi	6379:6379 ↗	0.35%	2 seconds ago	☒ ⋮ <input type="trash"/>
☐	☒ dsys-db	1ae2a6a9b11f	postgres	5432:5432 ↗	4.53%	2 seconds ago	☒ ⋮ <input type="trash"/>
☐	☒ backend-2	9f429a285420	dsysprojekt		0%	1 second ago	☒ ⋮ <input type="trash"/>
☐	☒ backend-1	3a2a1ebb9a5c	dsysprojekt		0%	1 second ago	☒ ⋮ <input type="trash"/>
☐	☒ dsys-frontend	4a308152f6d7	dsysprojekt		143.74%	1 second ago	☒ ⋮ <input type="trash"/>

Frontend/ React JS



```
frontend/  
├─ api/  
│   ├── GameAPI.js  
│   └── LobbyAPI.js  
├─ components/  
│   └── PuzzleGrid.js  
├─ hooks/  
│   └── useSSE.js  
├─ pages/  
│   ├── Home.js  
│   ├── AdminPage.js  
│   ├── GamePage.js  
│   └── LobbyPage.js  
└─ styles/
```




Backend/ Node.js

backend/

|— controllers/

| |— lobbyController.js

| |— puzzleController.js

|— routes/

| |— lobbyRoutes.js

| |— puzzleRoutes.js

|— sseClients.js

|— redisClient.js

|— db.js

|— server.js

Projekt Struktur

NonoRush/

├─ frontend/ (*React/ JS*)

├─ backend/ (*Node.js*)

├─ db/ (*PostgreSQL*)

│ └─ init.sql

└─ docker-compose.yml

Design Choices – Low Level

- Best Practices
 - Separation of Concerns (UI, API, Data)
 - Modularisierung (z.B. Grid Component)
- Docker Compose für Dockerized Setup
 - Angelehnt an Microservices Architektur
- SSE für einfache und zuverlässige Echtzeit Updates
- Redis als Broker für SSE
- PostgreSQL für permanenter Speicher
- Traefik für reverse proxy routing und load balancing

Design Choices – High Level

- Über Link beitreten

Nonorush

Join

Create

Player 1

Lobby ID

Join Lobby

Create Puzzle

<http://{hostaddress}/lobby/7I2J1X>

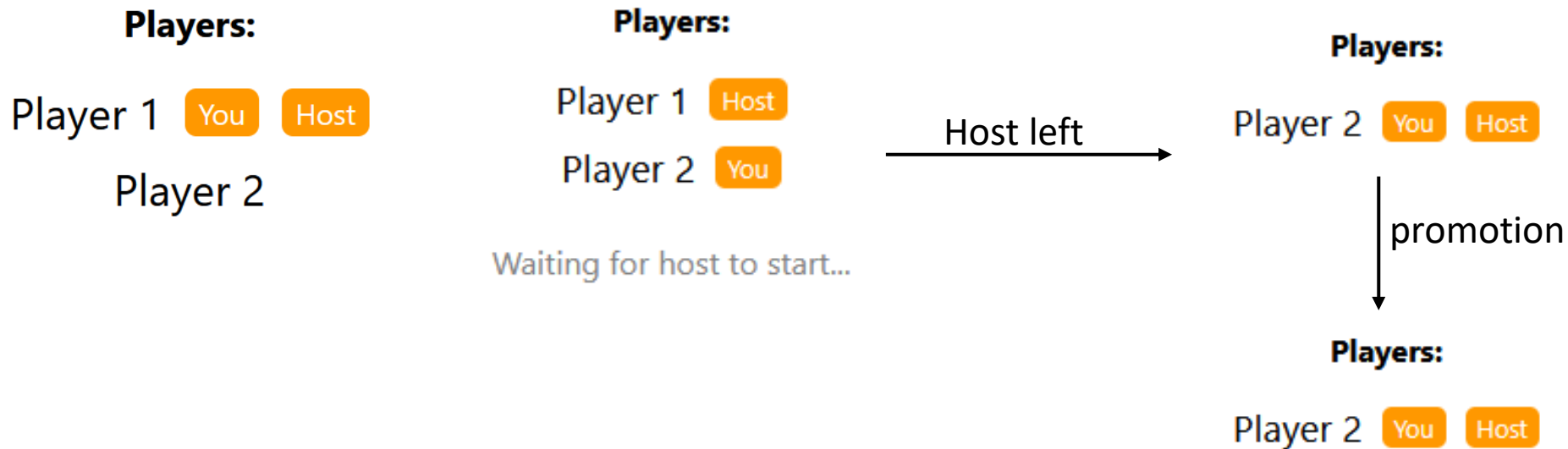
Nonorush: Lobby 7I2J1X

Player 1

Join

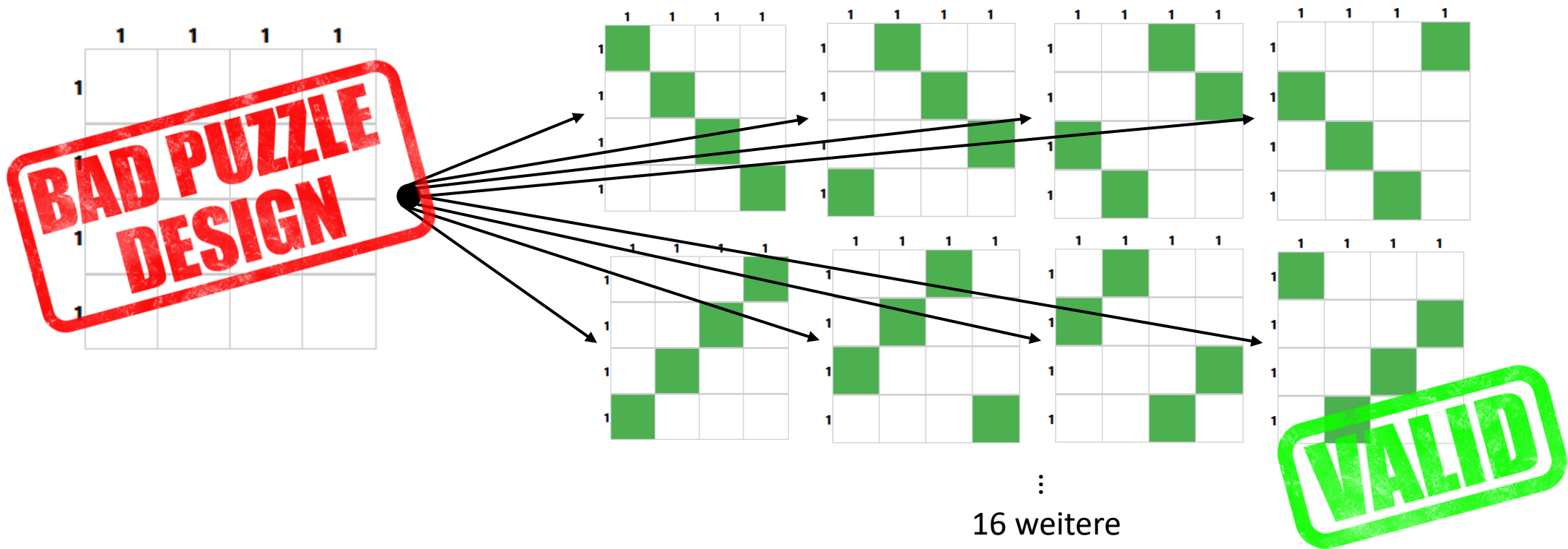
Design Choices – High Level

- Lobby Cleanup
- Remaining Host Promotion



Design Choices – High Level

- Lösung nach “Clues” und nicht fix Lösung.



Live*Demo*