



Patrick Scheidegger, Anja Friedrich

# Inhalt

---

**01** *Idee*

**02** *Design Decisions*

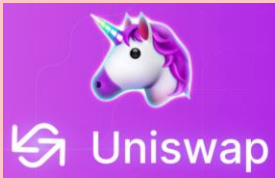
**03** *Architektur*

**04** *Demo*

**05** *Takeaways*

# Idee

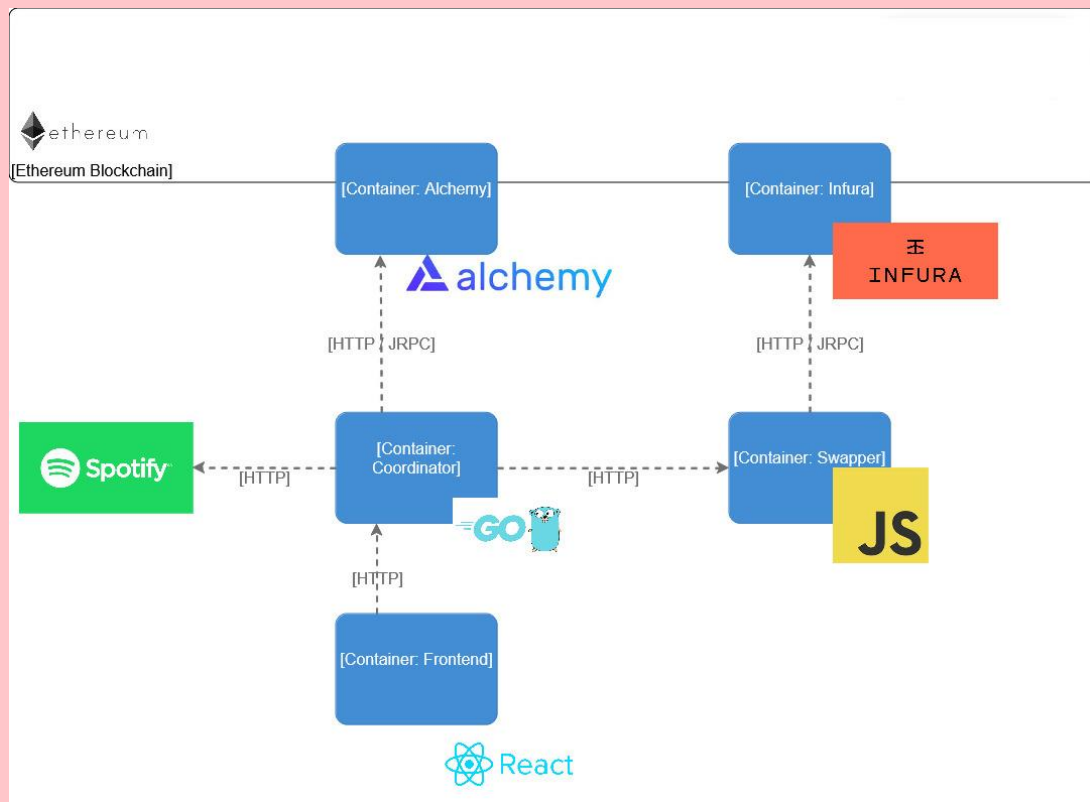
- Fiktive TaylorAG (vorher: Spotify-Popularität)
- Kauf und Verkauft fiktiver Aktien verändert Wert der “TaylorCoin”
- Ethereum-Sepolia-Netzwerk
- Uniswap als DEX
- Pool mit stabilen “TaylorCoins” und “AnjaCoins” wird verwendet



# Design Decisions

- Verwenden von zwei dedizierten Kryptowährungen, damit wir die volle Kontrolle über den Pool haben.
- JavaScript verwenden, damit Uniswap über Code angesprochen werden kann.
- Das Backend hat eine dedizierte Kryptowallet. Diese Wallet hat auch beide Währungen erstellt.

# Architektur



# Demo



# Takeaways

- Wir haben die Annahme getroffen, dass die Uniswap-Schnittstellen gut dokumentiert sind und es entsprechende Beispiele gibt. Diese falsche Annahme hat uns viel Zeit gekostet.
- Die Swapfunktion selber zu implementieren in einem SmartContract ist noch aufwändiger als bestehende APIs zu verwenden.
- SmartContracts können nur sehr schwer debugged werden. Es ist quasi ein Muss, während der Entwicklung eine lokale Blockchain zu verwenden.
- Arithmetik mit BigInt in JavaScript erfordert, dass alle Zahlen von Typ BigInt sind.



# – Fragen? –

