



### Relational statt Dokumentenbasierte DB

PostgreSQL statt DynamoDB

- Strukturierte Queries
- Weniger performant

Vorteile DynamoDB (NoSQL)

Schnell und skalierbar

### CDN vs ohne CDN

Schnellere Ladezeiten

Geringere Latenz (geo)

Skalierbarer

Höhere Ausfallsicherheit (Fällt ein Server aus -> Weiterleitung zum nächsten)

Weniger Last am File Server

Sicherheit (DDoS Schutz)

Kosten

Komplexität (Konfiguration, Cache Invalidierung)

Zusätzliche Abhängigkeit (CDN Provider)

### Lambda -> SQS -> Lambda vs Lambda

Entkoppelt Verarbeitung (Async)

Fehlerbehandlung (retries)

Skalierbarer => mehr Lambdas die die Queue abarbeiten

Mehraufwand

### Caching - ohne Caching

Reduziert Datenbanklast => reduzierte Kosten

Schnellere Ladezeiten (in memory)

Mehraufwand durch Caching Logik erstellen/invalidieren

Erhöhter Speicheraufwand (redundante Speicherung)

Erhöhte Kosten (Abwägung DB Kosten vs Cache Kosten)

ElastiCache (Redis) Alternativen

Memcached, weniger Funktionen als Redis aber gut für einfaches Caching

DynamoDB Accelerator (DAX) in-memory DynamoDB (overkill)

### Lambda Alternativen

EC2

### AppSync statt API Gateway

Falls man GraphQL verwenden möchte

### S3 Alternativen

DynamoDB (teuer)

Elastic File System