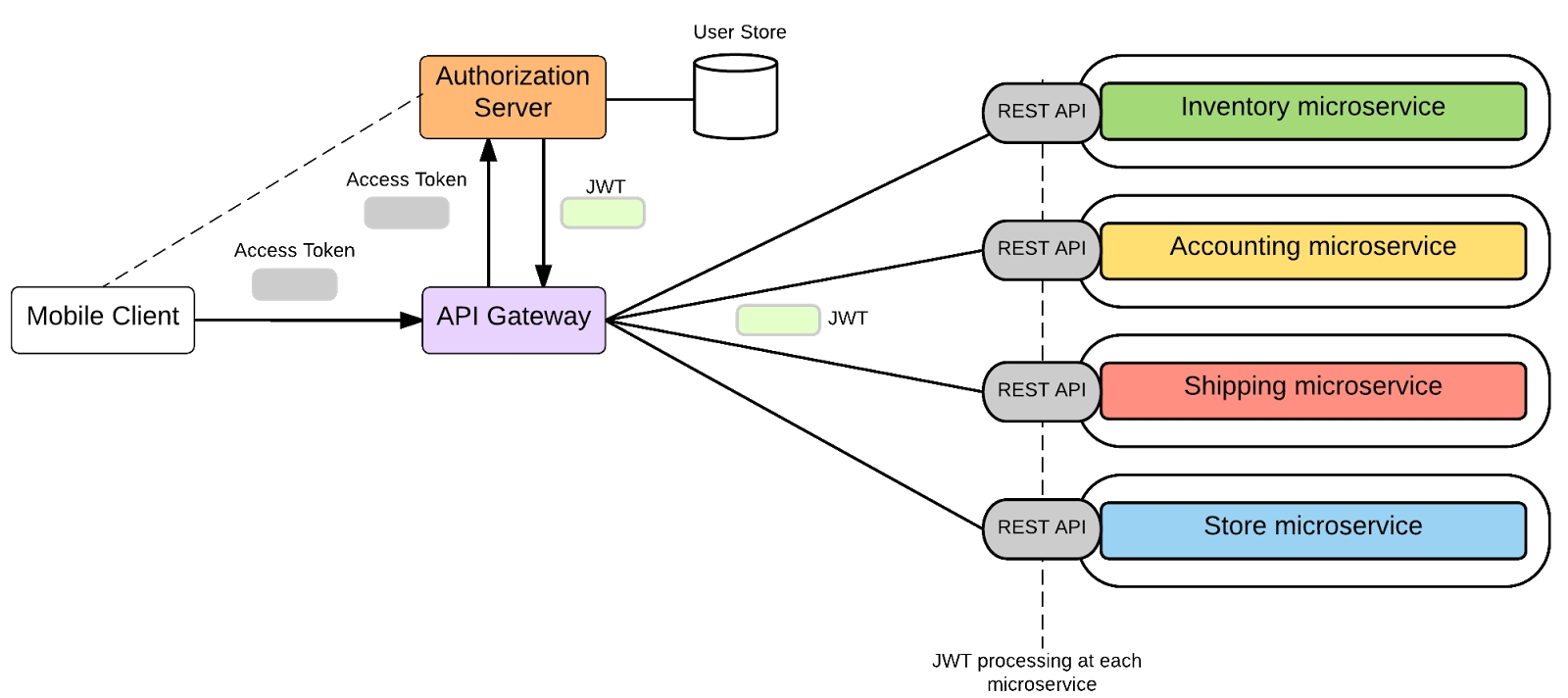
# Microservice arkitektur

Af: Anders

## Hvad er microservice?

Microservice er en software arkitektur som består af en eller flere små applikationer som for det meste vil være implementeret i skyen som eksempelvis en virtuel maskine. Ved at de køre på en virtuel maskine giver det bedre mulighed for at administrere hvor meget CPU samt hukommelse der bruges af servicen. Microservice arkitektur bruges til at lave enkelte applikationer som passer til andre servicer. Hver microservice vil normalt have deres egen database frem for en stor samlet.  
For at der kan kommunikeres mellem de andre services bruges der API’er.



## Fordele ved at bruge microservice med budgetmanager:

Ved brugen af microservice vil man som udviklere arbejde i mindre fokuseret grupper da der kun skal fokuseres på en service. Det kan også have effekt på koden da der er betydeligt mindre end hvis vi stod med en almindelig service, der er mindre kode at sætte sig ind i når det skal implementeres.

Hvis en service går ned betyder det at det kun er den ene service der ikke virker mere, da en service er uafhængig af alle andre services.

Det giver mulighed for at opdatere en service uden at man er nød til at ligge hele applikationen ned, og hvis der under opdateringen skulle ske mindre fejl, vil det betyde at det stadig kun er denne ene del der ikke er aktiv.

## Ulemper ved at bruge microservice med budgetmanager:

Det skal sikres at den microservice vi laver ikke kan være skyld i at hele systemet går offline, fordi vores microservice er afhængig af andre services. Hvis f.eks. en anden microservice er blevet opdateret kunne det ske at en bestemt service ikke er kompatibelt mere.

Der vil være risiko for at hver microservice er skrevet i et nyt sprog eller med nyt framework, hvilket kan være med til at gøre det svært at opretholde servicen. Derfor kan det være en fordel at der er sat nogle få regler op som man skal holde sig indenfor, det vil dog være i strid mod at hvert udviklingsteam selv bestemmer deres arbejdsmetode.