[Mailadresse]

Resumé

[Indfang din læser med et engagerende og kort resumé af dokumentet.   
Når du er klar til at tilføje dit indhold, skal du klikke her og skrive.]

IT6ONK - TSEIS

[Dokumentets undertitel]

Indhold

[Indledning 1](#_Toc515273490)

[Design 1](#_Toc515273491)

[Udførsel 1](#_Toc515273492)

[Særligt gode/bemærkelsesværdige implementeringer 1](#_Toc515273493)

[Mangler 1](#_Toc515273494)

[(Sammenligning af platforme måske, Service Fabric vs. Kubernetes) 1](#_Toc515273495)

[Konklusion 1](#_Toc515273496)

[Litteraturliste 1](#_Toc515273497)

Antal sider 10 ca. afhængig af brug af figurer o. Lign i rapporten, herudover kommer sider til indholdsfortegnelse, reference- og litteraturlister, forside og bilag. Husk sidenummerering!

Format PDF, navngivet ITONKF18Gr[nr].pdf eks ITONKF18Gr5.pdf.

Rapporten beskriver

# Indledning

I faget IT6ONK arbejdes der med distribuerede systemer, mere specifikt omhandler faget forbindelserne mellem objekter/microservices. I sammenhæng med dette er der i løbet af semesteret udviklet på et distribueret system til udveksling af aktier, kaldet The Stock Exchange Interaction System (TSEIS). Denne applikation har baseret sig på de opstillede Mountain-Goat krav til systemet og det oplæg til design af applikationen der blev skrevet i starten af kurset. Følgende rapport vil indeholde en beskrivelse af designet af TSEIS-applikationen, hvordan dette er blevet udført gennem Azure Service Fabric, en opsummering af særlige gode features, ligeledes en beskrivelse af systemets fejl/mangler og slutteligt en konklusion på hele processen og det færdige system.

# Design

Til at beskrive design af vores distribuerede system har vi opsat en N+1 arkitektur. Denne er inddelt i et Development view, et Physical view, et Proccess view og et Use case view. Vi har valgt at udlade det logiske view, da vi ikke følte det bragte værdi til beskrivelsen af designet. De førnævnte views vil blive beskrevet i føgende afsnit:

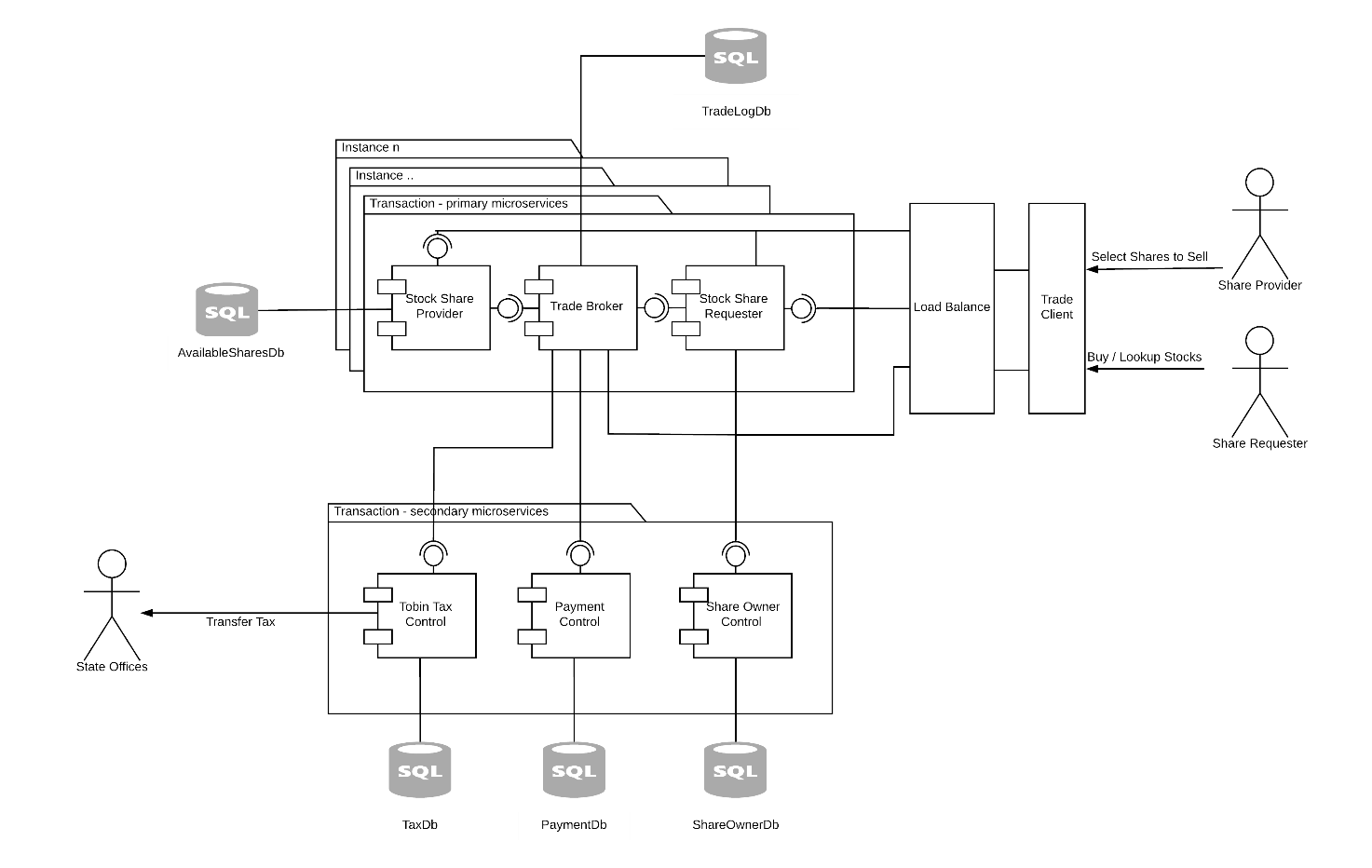
## Use Case View

Vores Use Case View tager sit udgangspunkt i oplægget til projektet, mere specifikt de opstillede Mountain-Goat krav til systemet. Disse ses nedenfor:

* As a "Tobin Taxer" I want to tax **all** transaction with a charge af 1%  of the total transaction value for then to be able to send the tax to the state.
* As a "Public Share Own Control" I want to be informed of any changes in the share ownerships for then to be able to track who is the owner of any share and series of shares.
* As a "Stock Trader Broker" I want to be the broker between the seller (provider) og the buyer (requester) for then to be able to intermediate the trade of one or more shares or one or more series of shares between one or more providers and/or one or more requesters.
* As a "Stock Trader Broker", "Stock Share Provider" or a "Stock Share Requester", I want to inform the Tobin Tax Control, that a transaction of a certain value has been commited, for then to be able obey the public rules about taxing and paying the 1% Tobin Tax.
* As a "Stock Trader Broker", "Stock Share Provider" or a S"tock Share Requester", I want to inform the The Public Share Owner Control, that a share, many shares or a series of shares or many series of shares has changes ownership, for the to able to inform the public about who own the shares.

## Development View

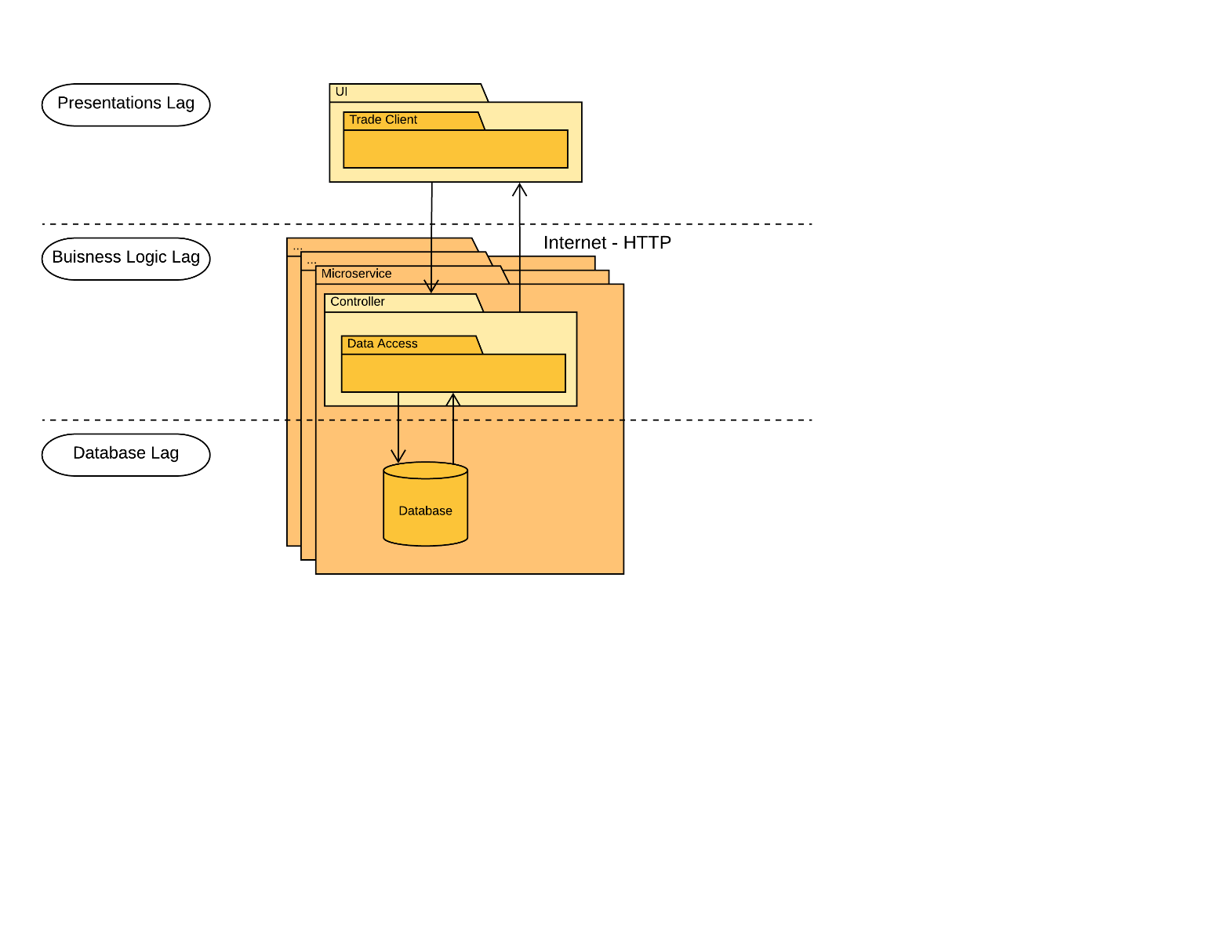
Figur 1 viser et component diagram for vores TSEIS applikation, det skal her bemærkes at kald imellem de forskellige microservices er udeladt, da dette kun ville skabe forvirring i diagrammet. I stedet kan en beskrivelse af alle de forskellige api-kald ses efterfølgende (måske i bilag):



Figur : Component diagram for TSEIS

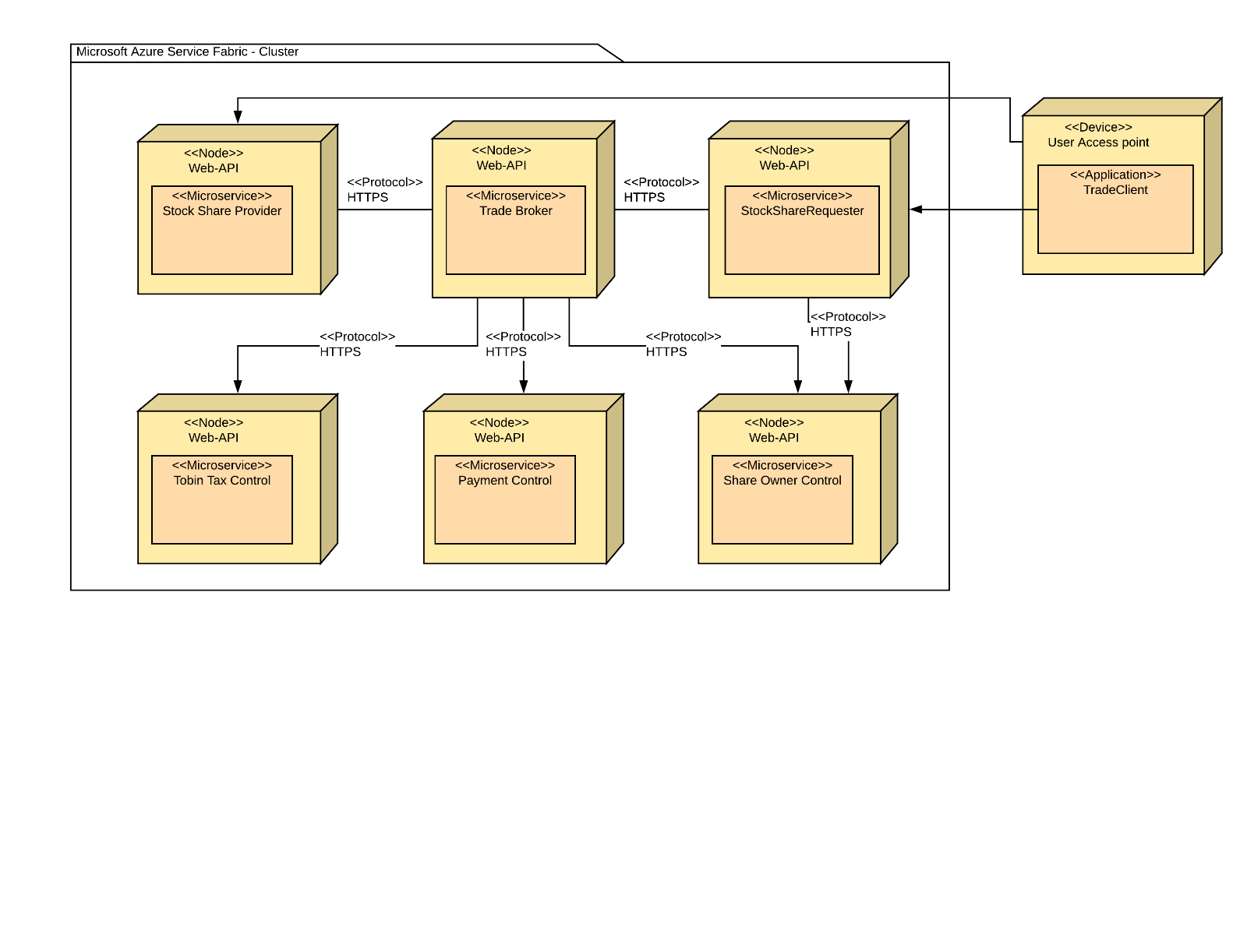
**ShareOwnerControl-microservice API-kald:**

|  |  |  |  |
| --- | --- | --- | --- |
| **HttpType:** | **Metode-navn:** | **Parametre:** | **Returværdi:** |
| Get | GetAllSharesForUser( ) | Guid userId | List<ShareOwnerDataModel |
| Get | Verify |  |  |



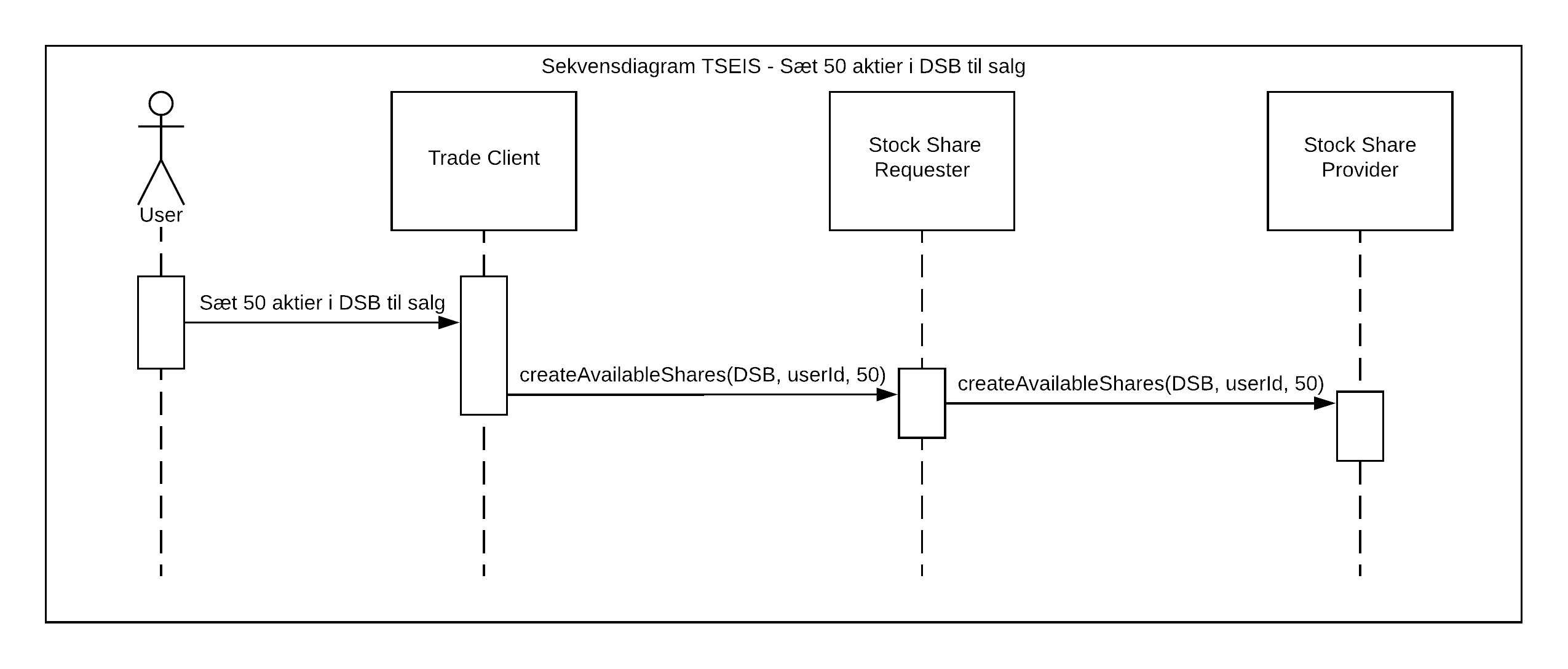
Figur 2: Package diagram for TSEIS

## Physical View

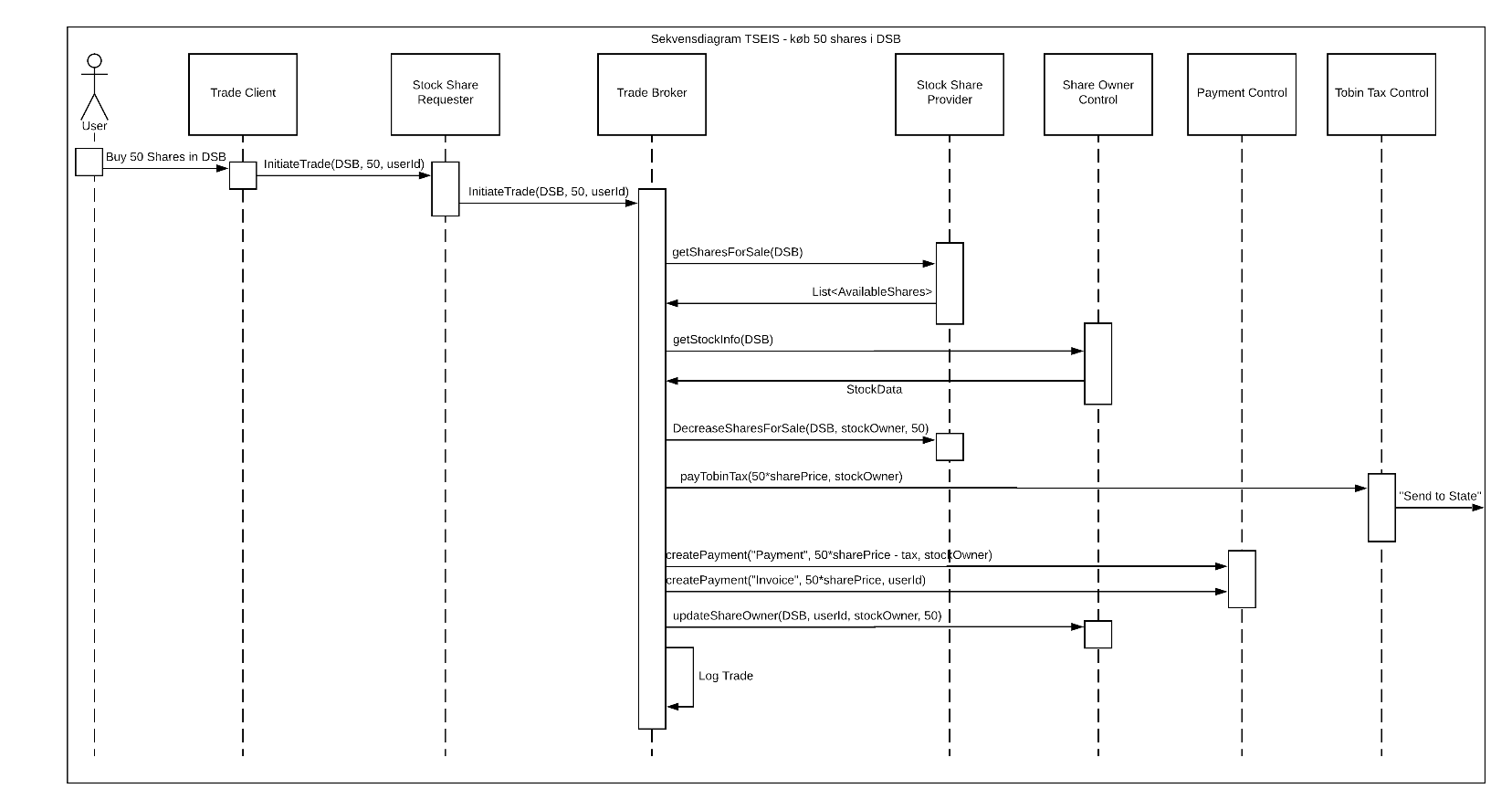


Figur : Deployment diagram for TSEIS

## Proccess View



Figur : Sekvensdiagram - Sæt aktier til salg



Figur : Sekvensdiagram - køb aktier

* I design lægges der primært vægt på at beskrive opbygningen af applikationens infrastruktur på den valgte cluster samt overvejelser for opdelingen af funktionalitet og microservices.
  + Selve ”business logic’en” er som sådan sekundær (kræver ikke den store dokumentation), idet den forventes at være brugt, som ”driver” for design af infrastruktur (microservices og cluster).
* Til UC- Diagram (not needed..)
  + I would say if the DB is internal to your app (i.e. it is used solely by this app and the external world doesn't interact with it directly in anyway, only via your app), you shouldn't represent it as a separate actor.
  + If the DB is shared and used directly by external users/systems, it is probably better to represent it as a separate actor.

# Udførsel - (Sammenligning af platforme måske, Service Fabric vs. Kubernetes)

* Udførelse på valgte platform, Azure Service Fabric eller/og Google Kubernetes, og inddrag her en overordnet en beskrivelse af udviklingsmiljø.

# Særligt gode/bemærkelsesværdige implementeringer

* Fremhævelse af ”guldkorn (/ …bamsefrø) ” for gennemførelse af tekniske løsning og en eller flere givne problemstillinger, dvs. beskriv en særlig smart, elegant eller på anden måde ”bemærkelsesværdig” løsning. Et eksempel herpå er inddragelsen af eksisterende standard services/komponenter til f.eks. persistering.
* Persistering er opsat via ormLite, alt opsætning findes derfor i controlleren
* Swagger og auto-generering af clients (NSwag)

# Mangler

* Opsummering af problemer, som ikke er løst eller udeståender som kan ses som mangler for projektet. Konkretiser om muligt betydningen heraf.
* ShareOwnerControl
* TradeBroker

# Konklusion

# Litteraturliste

* Brug referencer til brugte artikler og hjemmesider. Og gør det gerne i ”overdrevent” omfang, dels af hensyn til den videre brug af jeres projekt men også i forhold til andres læsning af rapporten (Eksaminator).

# Bilag???

* Synopsis???
* Mountain goat
* Feedback fra anden gruppe mybe? Don’t fucking know…