I4DAB Handin 4  
  
Gruppe 9  
  
Dato: 25/05-2018

|  |  |
| --- | --- |
| Navn | Studienummer |
| Armine Sanjarizadeh | 201607125 |
| Daniel Pat Hansen | 201601915 |
| Fatima Kodro | 201609565 |
| Martin Haugaard Andersen | 201605036 |
| Søren Bech | 201604784 |

Indhold

[Indledning 2](#_Toc515036407)

[Arbejdsplan 2](#_Toc515036408)

[Analyse og design 2](#_Toc515036409)

[Udkast til Databaserne 3](#_Toc515036410)

[Smart Grid Info DB 3](#_Toc515036411)

[Prosumer Info DB 4](#_Toc515036412)

[Trader Info DB 5](#_Toc515036413)

[Smart meter 7](#_Toc515036414)

[Blockchain 8](#_Toc515036415)

[Referencer 9](#_Toc515036416)

# Indledning

Der skal designes mindst 3 databaser med tilhørende REST API til brug for et afregningssystem, som i sidste ende kan benytte Bitcoins til at afregne salg og køb af strøm mellem en række borgere og/eller virksomheder i en lille landsby.

Der skal i opgaven arbejdes med en landsby som kaldes ”Village Smart Grid”. Denne landsby består af 33 husstande og 12 virksomheder/landbrug.

Hver Prosumer har et smart meter til at måle og opsamle de konkrete data omkring strømproduktion og strømforbrug. Det er muligt for Prosumers at købe og sælge en mængde kWh-blokke mellem hinanden eller til ”The National Smart Grid” som er et stort Smart Grid over resten af Danmark.

## Arbejdsplan

Grundet at der har været travlt i andre fag, har det været svært at lave en arbejdsplan, da det var svært at vide, hvornår der var god tid til opgaven. Der har dog været en planlagt arbejdsfordeling i gruppen.

Arbejdsfordelingen i gruppen er således, at hele gruppen i starten af opgaven var sammen om at analysere, hvordan alle databaser skulle se ud. Derefter er gruppen delt ud på databaserne og diagrammerne. Trader Info DB blev analyseret til at ville være den sværeste at implementere og derfor er der sat to personer på denne. De to andre databaser er der blevet en person på, da de ikke virkede svære at implementere. Den sidste person er sat til at arbejde på diagrammerne.

Selvom alle medlemmer havde sit eget ansvarsområde, blev der gjort godt udnytte af at få hjælp fra hinanden. Til sidst blev der samlet om at få skrevet en god rapport, der ville indeholde alle de overordnede krav.

# Analyse og design

Der skal oprettes 3 database: Trader Info DB, Prosumer Info DB og Smart Grid Info DB.

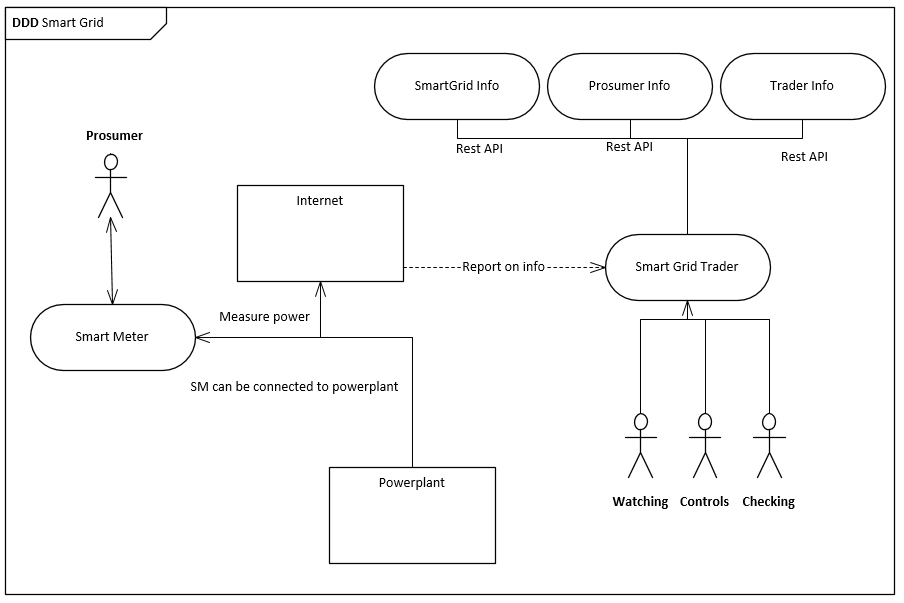
**Trader Info DB** skal indeholde informationer om de handler der er forekommet i fortiden, handler der forekommer lige nu og handler som skal ske i fremtiden mellem beboerne. Der beskrives beskrives hvordan fremtidige handler vil håndteres, de skal dog ikke håndteres.

**Prosumer Info DB** skal indeholde informationer om de enkelte prosumers i det givne Village Smart Grid, hvilket inkluderer, hvor mange kWh de gennemsnitligt forventer at forbruge de forskellige tidspunkter på dagen.

**Smart Grid Info DB** skal indeholde informationer om det givne Village Smart Grid, angående hvor mange husstande og/eller virksomheder der er i det givne Village Smart Grid. Samt skal det indeholde nogle informationer omkring hvilke konfigurationer de forskellige husstande/virksomheder har.

Som databaseteknik er der valgt at Trader Info DB skal være en Dokument database, da tabellerne hele tiden bliver opdateret med nye handler mellem beboerne.  
De to andre databaser er valgt til at være SQL databaser, da tabellerne i disse databaser er statiske for det meste, hvilket vil sige at de næsten aldrig ændrer sig. Det er data som f.eks. deres smartmeter, konfiguration og type.

Der vælges at lave REST API på Trader Info DB og Smart Grid Info DB, da det er disse databaser der er mest behov for at få fat i udefra.   
Der var fra start besluttet at Trader Info DB i hvert fald skulle laves et REST API på, da det er vigtigt at få informationer omkring hvordan handlerne er mellem beboerne.  
Det var herefter lidt svært at beslutte hvilken database der ellers skulle laves REST API på, men efter omtanke, gav det mest mening at lave det på Smart Grid Info DB, da det også er vigtigt at have fat i informationer omkring hvordan konfigurationen er i det givne Village Smart Grid.



Figur : DDD for smart grid

Som der kan ses på Figur 1 så har vi fra problemdomænet prøvet at sætte det ind i det tekniske domæne så vi kan analysere og udbygge systemet bedre. Ideen er at bag smart meter er der enten en Consumer, Producer eller Prosumer som giver eller modtager energi fra kraftværket eller en anden producer/prosumer. Derfor tillader vi at Prosumer (som laver og forbruger energi) at være alene efter Smart Meter som repræsentere enten en Consumer eller Producer.

# Udkast til Databaserne

## Smart Grid Info DB

Smart Grid Info DB skal, som tidligere nævnt, indeholde informationer om det givne Village Smart Grid, samt holde styr på konfigurationerne i de forskellige huse/virksomheder.

Der er tænkt at denne database kun indeholder en tabel som indeholder alle de prosumers der er i dette Smart Grid. Hver prosumer er identificeret med hver sit unikke id, som går fra 1 til 45, da dette er antallet af prosumers. Tabellen skal så indeholde informationer om hver prosumer omkring, hvilken type prosumer der er tale om, altså om det er en husstand, virksomhed eller noget andet. Det skal også indeholde, hvilket smartmeter hver prosumer har, da dette er afgørende for hvordan strømmen bliver målt. Til sidst skal der vides, hvilken produktionstype de forskellige prosumers har, altså om de får strøm fra en energikilde som f.eks. solceller eller vindmøller. Det regnes her med at hver prosumer kun kan have én produktionstype. På Tabel 1 ses et udkast til Smart Grid Info databasen.

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Type | Smartmeter | Produktionstype |
| 1 | Husstand | Smartmeter 2.7 | Solcelle |
| 2 | Virksomhed | SMARTMETER3000 | Vindmølle |

Tabel : Udkast til Smart Grid Info DB

## Prosumer Info DB

Prosumer Info DB indeholder, som nævnt tidligere, informationer om, hvor meget de forskellige prosumers forventer at bruge af kWh på forskellige tidspunkter på dagen.

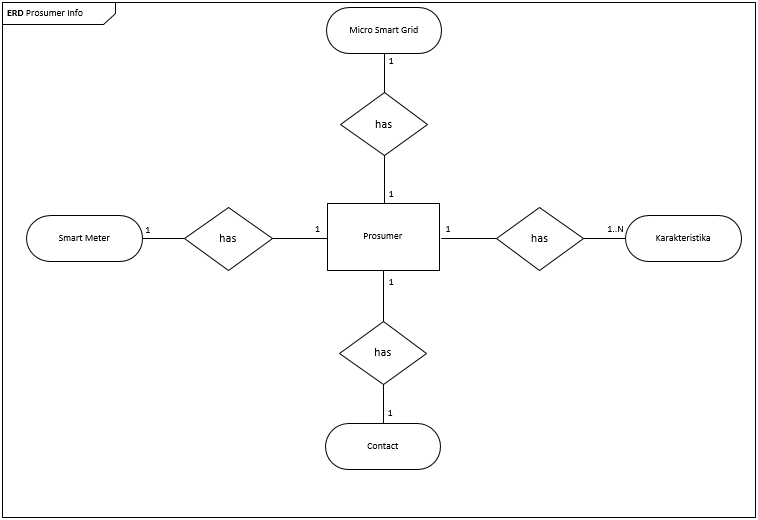
Denne database skal indeholde et vindue for hvert tidsinterval. I dette tilfælde er der bestem at der skal kigges på hver time på et døgn. Dette interval kan sagtens være mindre. Et mere realistisk interval kunne være et kvarter, men grunden til at der er valgt en time i dette tilfælde er for ikke at skulle bruge alt for meget tid på at oprette vinduer. Da denne database er en relationel database, er der valgt, at hvert vindue vises med en tabel. Samtidigt oprettes der en ekstra tabel, som indeholder alle de primary keys (Id) for Prosumers. Resten af tabellerne indeholder så foreign keys til de primary keys.

Der skal derfor oprettes 25 tabeller i denne database i alt. På Tabel 2 ses et udkast til en tabel over en bestemt tidsperiode i Prosumer Info databasen.

|  |  |
| --- | --- |
| Id | Forbrug(kWh) |
| 1 | 2 |
| 2 | 15 |

Tabel : Udkast til Prosumer Info DB

På Figur 2 ses et ERD for Prosumer info DB. Det giver et kort billede af, hvad den enkelte prosumer indeholder.

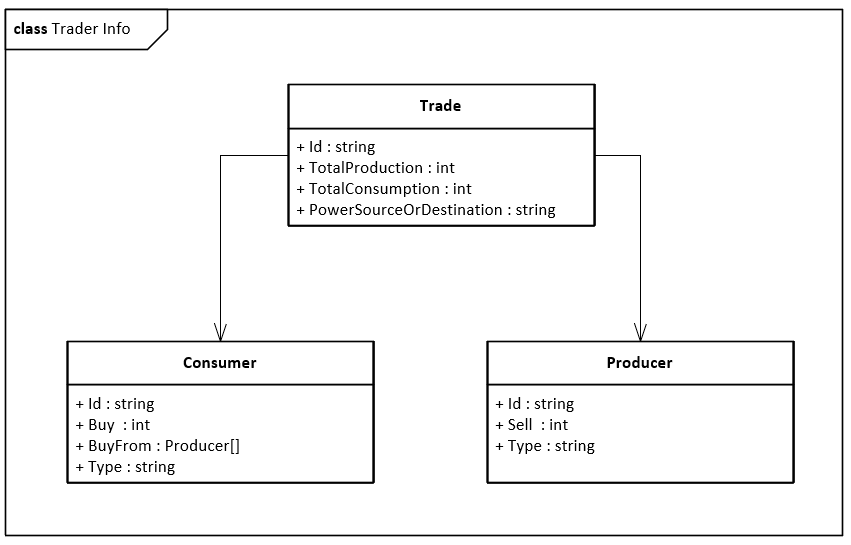


Figur : ERD for Prosumer info

## Trader Info DB

Trader Info DB indeholder som nævnt tidligere informationer om handlerne mellem prosumerne. Dette er både handler som er sket i fortiden, sker nu eller sker i fremtiden.

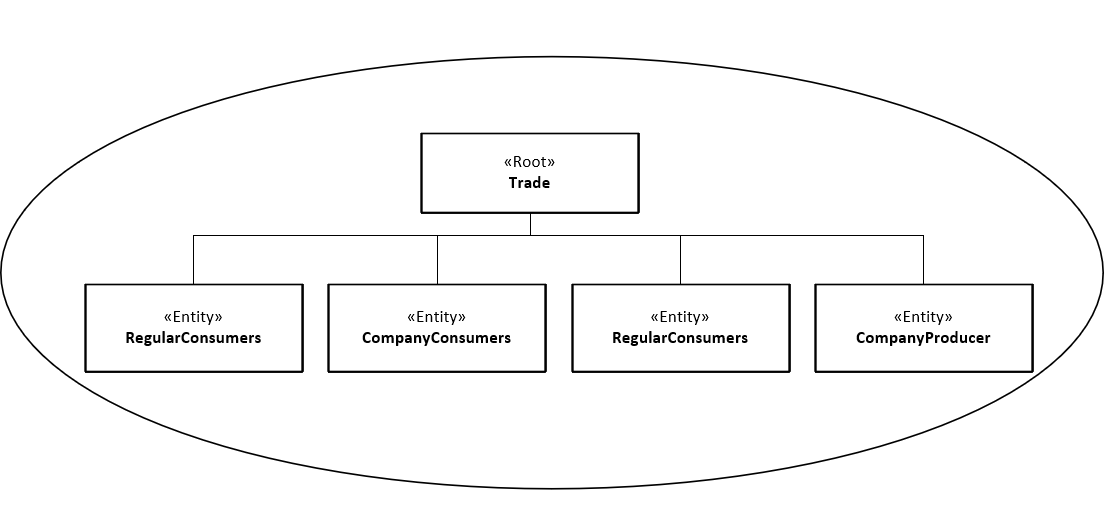
Denne database skal, ligesom i Prosumer Info DB, indeholde et vindue for hvert specifikt tidsrum. Da denne database er en dokument DB, bliver der lavet et dokument for hvert tidsrum. Der bliver herudover lavet fire lister: to lister over virksomheder som er producers og consumers og to lister over almindelig husstande som er producers og consumers. Producers er de prosumers, der producerer mere kWh end de forbruger, og comsumers er dem, der forbruger mere end de producerer. Figur 1 viser sammenhængen mellem en Trade og Producers og Consumers.



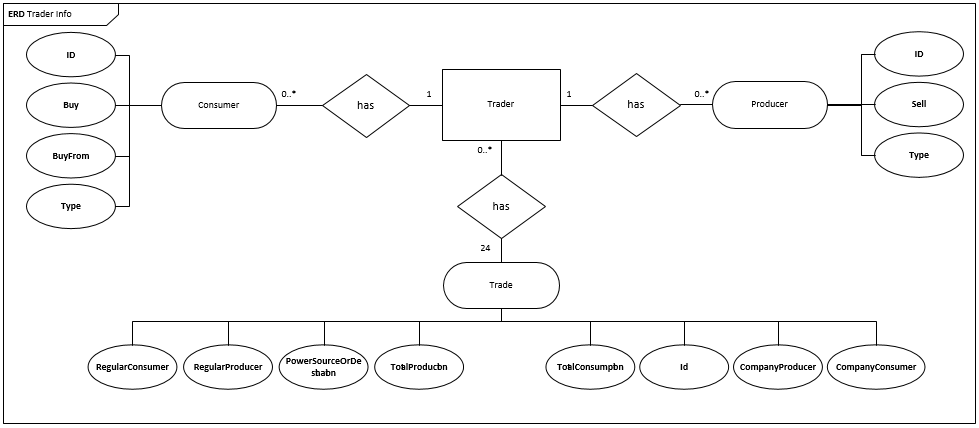
Figur 3 Klassediagram over Trader Info

En handel indeholder et Id som er et tidsstempel. Den totale produktion og det totale forbrug bliver beregnet. På basis af dette bliver en string ”PowerSourceOrDestination” sat. Denne beskriver hvad der skal ske med overskydende energi eller energi som er i underskud. Er der meget i overskud skal energien tilbage til National Smart Grid, men er der derimod lidt i overskud ryger dette i Dyppekogeren. Der er ikke et system eller program som håndterer energien: det er blot en beskrivelse af, hvad der skal ske. I hver Trade er der en række af Prosumers. En Prosumer har et Id og en type. En Producer har en ”Sell” værdi. Dette er hvor meget denne Prosumer producerer af kWh-blokke i overskud og gerne vil sælge. En Consumer har en ”Buy” værdi, dvs. hvor meget denne Consumer bruger mere end den producerer og derfor gerne vil købe. Desuden har den en liste af producers som den vil købe fra.

For at kunne generere data er et ”Smart Meter” simuleret. Denne genererer tilfældige tal for en række af Prosumers og indsætter dem i databasen. Dette er et program som anvender et REST API til at tilgå Trader Info. Programmet vil genere data til 24 tidsstempler, dvs. én dag. De 24 handler gemmes i databasen. Det er muligt at generere mere data og ikke overskrive eksisterende data, men for overskuelighedens skyld bliver dette overskrevet. Hvis man forbandt denne database sammen med Prosumer og Smart Grid Info databaserne kan man hente karakteristika for Prosumers og vide hvor meget hver Prosumer forbruger og producerer per dag, og på denne måde kunne udregne et udgangspunkt for fremtidige handler. Dette er ikke gjort da der er fokuseret på at implementere databaserne, ikke implementere avanceret logik for at generere data. Figur 2 viser sammenhængen mellem en Trade og de fire entities som er fire lister af Prosumers.



Figur 4 Aggregate over Trader Info



Figur : ERD for Trader info

På Figur 5 ses ERD for Trader Info. Den har en relation til Producer, Consumer og Trade.

Producer indeholder ID, salg og type, imens Consumer indeholder ID, køb, købt fra og type.

Trade indeholder lister med alle Consumers, Producers, produktion af strøm og forbrug af strømmen. Ved hjælp af denne Trade kan man se både hvad der bliver brugt og købt i løbet af en opdatering fra Smart Meter.

## Smart meter

Hver prosumer har deres eget smart meter, som måler strømproduktion og strømforbruget i beboelsen. Denne smart meter bliver simuleret, med at give nogle tilfældige værdier i form af kWh-blokke for hver prosumer. Da der er forskel på om den enkelte prosumer er en husstand eller virksomhed, om produktionstypen er en solcelle eller en vindmølle osv. bliver der genereret forskellige værdier baseret på typen af grunden.

Solceller og vindmøller tænkes at være de mest populære former for energikilder i dette mini smart grid. En artikel fundet fra (Thønnings, 2016), er det undersøgt, at vindmøller er mere økonomiske end solceller, så det kan bedst betale sig for en prosumer at få strøm fra en vindmølle.

# Blockchain

Blockchain er en kæde af transaktioner, der hver kan blive kaldt en blok. Princippet er at alle transaktioner bliver gemt, og de udgør en kæde af transaktioner, så man kan altid spore hvor transaktionen kommer fra.

Informationen i denne blok bliver kaldt en ”blockchain ledger”. Blockchain ledger kan bestå af information om senderen, modtageren og selve blokkens data i form af en hash. Denne information skal bruges til at identificere blokken.

Blockchain bliver brugt i bitcoin, til at holde styr på transaktionerne mellem de forskellige prosumers.

Hvis hele systemet i Village Smart Grid, skulle implementeres, ville der skulle bruges Blockchain i sammenhæng med Trader Info DB, til at holde styr på forespørgslerne og transaktionerne mellem prosumerne. Der ville her så blive brugt bitcoins i forbindelse med handlerne mellem prosumerne.

# Referencer

Thønnings, M. (2016). *Solceller eller vindmøller? Lerchenborg går forrest.* Sydsjælland Tidende.