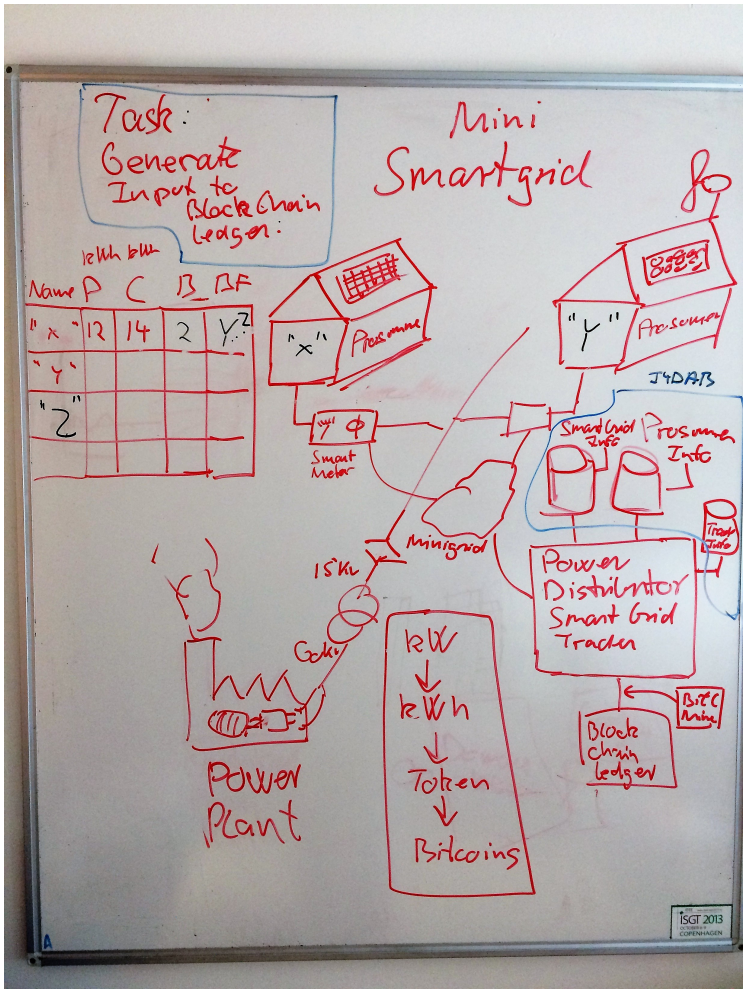
Efterår 2018
Ingeniørhøjskolen Aarhus Universitet

DAB - Handin 4

Gruppe 3

Village Smart Grid



Studienummer	Navn	Studieretning
Studienummer	Fornavn Efternavn	IKT
Studienummer	Fornavn Efternavn	IKT
Studienummer	Fornavn Efternavn	IKT
Studienummer	Fornavn Efternavn	IKT

Dato: 12-12-2018

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

Indholdfortegnelse

1 Indledning

1.1 Links til Databaser

Vi har benyttet samme MS SQL Database, til begge "databaser", da de alligevel kører uafhængigt af hinanden.

- SQL database:
- Azure DB:

1.2 Case

1.3 Krav

1.3.1 Problemdomænets definitoner

- Databasen Smart Grid Info: indeholder en beskrivels af konfigurationen af det givne Mini Smart Grid.
- Databasen Prosumer Info: indeholder oplysninger og beskrivelse af de "Prosumers"som er med i den given Mini Smart Grid.
- Databasen Trader Info: indeholder oplysninger om, hvorledes der er handlet, hvorledes der handles lige her og nu og hvorledes der skal handles fremover. Tabellen i øverste højre hjørne af billedet er en skematisk beskrivelse af nogle af oplysninger i Trader Info DB.
- Smart Meter: En IOT enhed eller mere konkret en enhed, som måler og opsamler de konkrete data omkring strøm, strømproduktion, strømforbrug og status iøvrigt for den pågældende Prosumer.
- Power Distributor/Smart Grid Trader: Systemet som er forsyningsselskabet bag Mini Smart Grid bruges til at overvåge, styre og kontrollere Smart Grid. Trader delen står for at håndtere de indbyrdes salg mellem de enkelte Prosumers.
- Power Plant: symbolisere en typisk central elforsyning uden for Mini Smart Grid.

1.3.2 Kravspecifikation

- Udvikles 3 databaser - Trader Info, Prosumer Info og Smart Grid Info.
- De nævnte databaser skal udstyres med Front End REST API, hvor nødvendige metoder afhænger af interaktionen med resten af systemet.
- MiniSmartGrid i denne opgave kaldes ’Village Smart Grid’, til 33 husstande og 12 virksomheder/landbrug.
- Der gælder en række karakteristika, som har relevants for MiniSmartGrid i forhold til hver overordnet type husstands-prosumer og virksomhedsprosumer. Disse skal kunne registreres.
- Afregning skal ske på basis af kWh-blokke. Smart Meter måler her og nu strøm og spænding på Prosumers elstik.
- Differenesen mellem ind- og udgående blokke aflæses i forhold til en given tidsperiode kaldte afregningsvinduet.
- Der er en "bryder"mellem "The Village Smart Grid"og resten af Danmark.
- Resten af Danmark skal betagtes som en et stort Smart Grid ’The National Smart Grid’.
- Hver Prosumer i Mini Smart Grid er kendt via sin kobberforbindelse (sit elstik), sine karakteristika og sit Smart Meter.
- Afregningsprincip et det princip som Prosumers bruger til at afregne kWh-blokke indbyrdes.

- Der kan inddrages et salgsvindue eller et købsvindue, hvor et salgsvindue er den tid og det tidsum en prosummer vil stille/stiller en mængde kWh-blokke til rådighed, og hvor et købsvindue er det er den tid og det tidsrum en prosumer vil købe kWh.blokke.
- I et givent afregninsgsvindue gælder en bestem pris for en kWh-blok. Hvis priserne for en kWh-blok ønskes dynamiske i forhold til et bestemt elmarked gøres aflæsningsvinduet kortere men ønskes en mere fast afregningsstruktur gøre afregningsvinduet længere.
- En Prosumer har mulighed for selv at bestemme, hvor mange kWh-blokke, der ønskes købt eller solgt, samt hvilke købs- og salgsvinduer der er gældende.
- Alle handler afsluttede, igangværende og kommende kendte handler skal registreres.
- Alle afregninger der er gennemført , under gennemførelse og som er kendt til at ville blive gennemført, skal registreres.

1.3.3 Tekniske Krav

- Den tekniske platform der udvikles på er Microsoft .NET. Nyeste gældende versioner.
- Enten .NET Standard eller .NET Core eller et miks af begge.
- ADO.NET Entity Framework, EF eller EF Core, benyttes mod SQL databaser. Nyeste gældende versioner.
- Azure Cosmos DB SQL API .NET benyttes Dokumentdatabaser. Nyeste gældende version.
- Der må gerne udvikles en eller flere testklienter som bruger de udviklede REST API'er.
- Tilsvarende må Swagger Swashbukle tilføjes REST API servererne for test af REST API.
- Tilsvarende med predefinerede request der sendes via REST klienter som Postman REST Client eller Advanced Rest Client må bruges.
- Der udvikles på SQL Express LocalDB og Azure CosmosDB SQL API Emulator.

1.4 Arbejdsfordeling

Efter at gruppen har kigget og undersøgt opgaven nærmere, er der besluttet at vi deler opgaven op således at vi har en mand på hver database udover dokumentdatabasen TraderInfo, hvor vi har valgt at sætte 2 mand på, da vi mente at det var der som vil være mest arbejde at lave.

2 Design

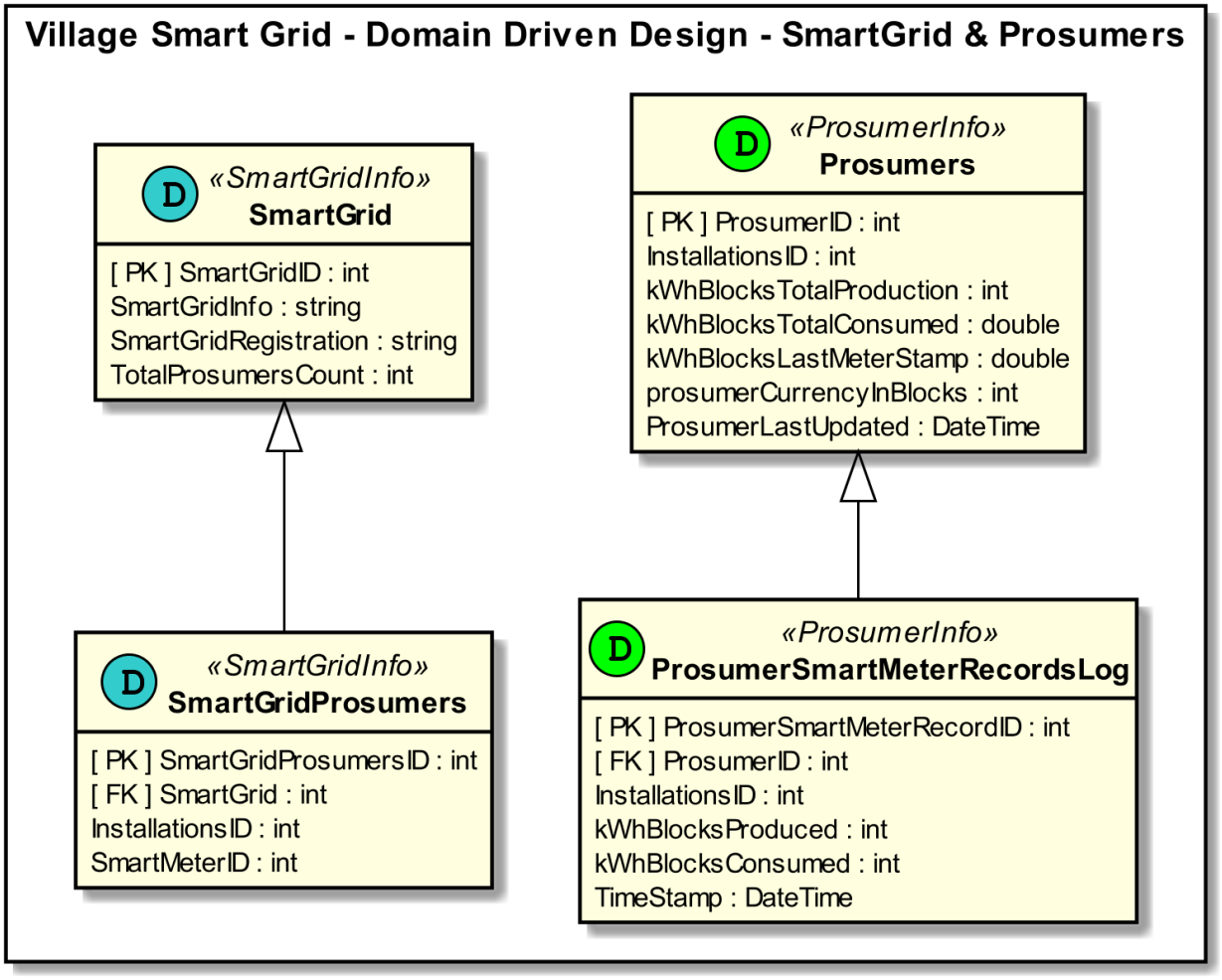
Der er til denne løsning benyttet 1 NoSQL-database og 2 SQL-databaser for at overholde kravet om mindst 1 af hver jf. kravsspec.

Gruppen har valgt at lade Traderdatabasen være NoSQL, da dokumentdatabasen blev vurderet som den mest fleksible, idet den ikke er så låst fast som SQL-databaserne.

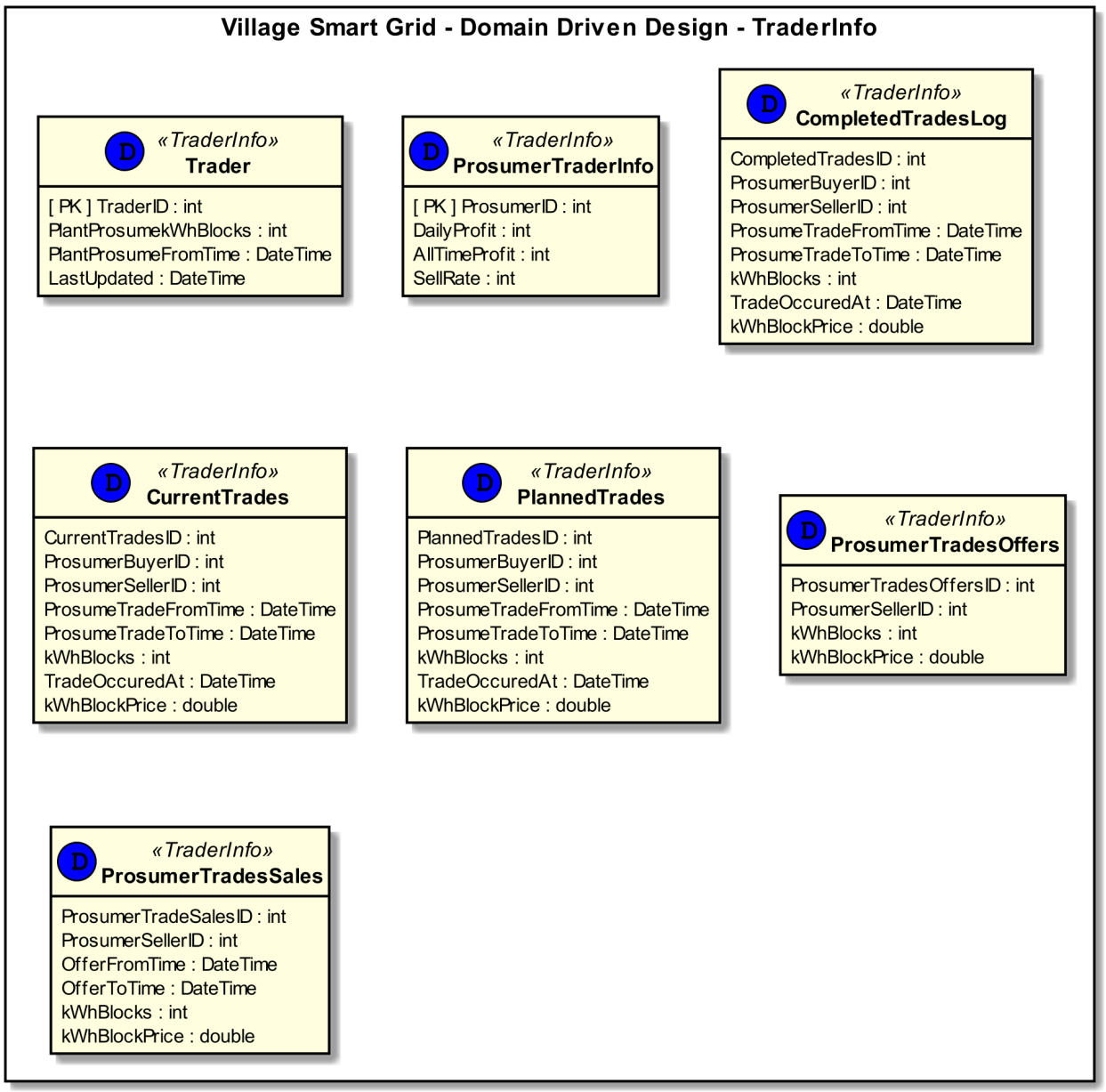
2.1 Entity Relationship Diagram (ERD)

2.2 Database Studio Design (DSD)

2.3 Domain Drive Design (DDD)



Figur 1: Opstilling af DDD-diagram af ProsumerInfo og SmartGridInfo.



Figur 2: Opstilling af DDD-diagram af TraderInfo.

smartgridinfo: -smartgrid info omkring dette grid, antal prosumers m.m, der vil kun være én række -smartgridprsoumers info omkring prosumers for hvem der er i dette grip, installations nr m.m

prosumerinfo: -prosumers: infoomkring total produktion af kWh blokke stamps for nyeste meter måling 1 række pr rosumer -prosumersmartmeterrecordslog tidsstemplede elementer, med prosumer id, for hver gang der er sendt data fra en given elmåler/smartmeter

traderinfo: -trader info omkring trader delen i forhold til grid, total brug af kWh fra plant(uden for systemet) eller hvis negativ viser den forbrug.

-prosumertraderinfo en række pr prosumer, giver info om totalproduktion og profit og sellrate

-completedtradeslog en ny række for hver enkel handel for en given periode, m. køber og sælger samt blokke og pris der er aftalt

-currenttrades igangværende trades, køber id og sælger id, samt pris og andtal kWh blokke, og fra og til stidsstamp

-plannedtrades fremtidige trades, køber id og sælger id, samt stamps fra og til, og antal blokke

-prosumertradessales liste over specielle tilbud fra en prosumer, om en bestemt tidspunkt og antal blokke(f.eks til en billigere pris om natten), bestemt af et tidsmarkør for hvornår folk kan købe til prisen

-prosumertradesoffers antal kWh blokke til salg og pris, men uden tidsperiode/interval

3 Implementering

3.1

4 Konklusion

Som der er gjort bemærket, så er det lykkedes at lave et system bestående af tre databaser:

- En documentDB
- To relationelle databaser