# Ontwerp

## Database Model

"Een datamodel laat de logische structuur van een database zien, inclusief de relaties en beperkingen die bepalen hoe gegevens opgeslagen en toegankelijk gemaakt kunnen worden. Een individuele database model wordt ontworpen op basis van de regels en concepten van een meer uitgebreid datamodel dat een ontwerper uitkiest om mee te werken." (*Wat is een datamodel?* Z.d.)

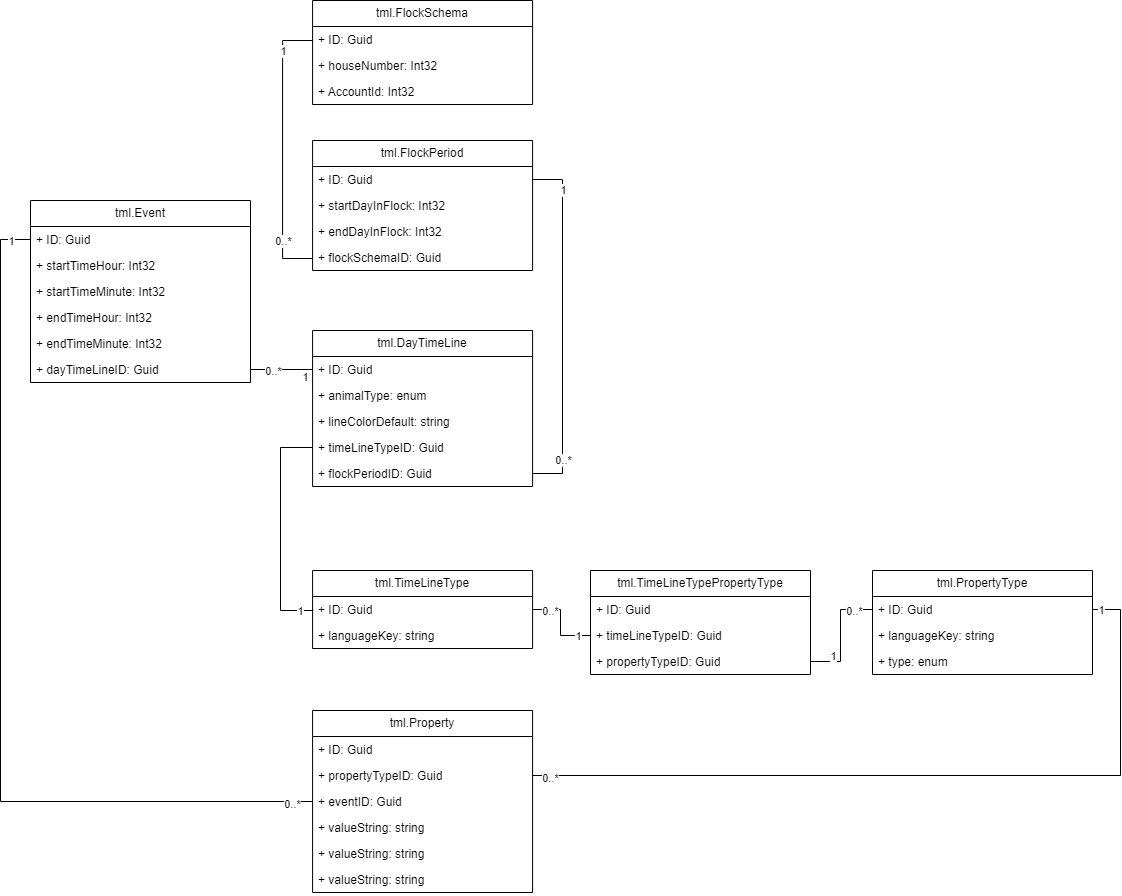
Dit diagram is gemaakt om de efficiëntie en consistentie van gegevensopslag te verbeteren en om te helpen bij het handhaven van de data-integriteit.

Een flockschema bestaat uit meerdere flockperioden, elk bestaande uit meerdere dagtijdlijnen, en elke dagtijdlijn bevat op zijn beurt meerdere evenementen. Elke dagtijdlijn is gekoppeld aan een specifiek type, dat vervolgens via een koppeltabel wordt gekoppeld aan meerdere propertytypes. Deze structuur maakt het mogelijk om te bepalen welke propertytypen aan welke dagtijdlijnsoorten worden toegewezen. Bijvoorbeeld, een dagtijdlijn met het type "licht" kan eigenschapstypen hebben zoals "fade-in/-out" en "lichtintensiteit". Om de werkelijke gegevens van de properties op te slaan, is er een propertytabel toegevoegd.

Niet alle properties worden opgeslagen in de propertytabel. Sommige properties, zoals de begintijd en eindtijd, zijn van belang voor alle soorten events en worden daarom rechtstreeks in de eventtabel opgeslagen.

Bij het ontwerp van het gegevensmodel is ervoor gekozen om de begintijd en eindtijd op te splitsen in afzonderlijke velden voor uren en minuten. Hierbij wordt geen gebruik gemaakt van het "DateTime"-type, omdat de datum geen relevante informatie is voor dit specifieke programma. Door de tijd op deze manier op te slaan, kan het systeem efficiënter werken en wordt overbodige informatie vermeden.

Aangezien de applicatie die wordt ontwikkeld beschikbaar moet zijn in verschillende talen wordt er een language key opgeslagen in de database. Er is ervoor gekozen om voor alle ID’s een guid type te gebruiken vanwege de uniciteit, beveiliging en gemakkelijke integratie.



Figuur 1 Database Model

## Dataflow

"Een gegevensstroomdiagram (DFD) brengt de gegevensstroom van een proces of systeem in beeld. Het is de meest gebruikte vorm om duidelijk te maken hoe de processen binnen een systeem zich afspelen." (*ICT Portal*, 2022)

Het TimeTables project staat in het midden van de dataflow en heeft verschillende input- en exportmethodes. Dit diagram is gemaakt om het duidelijk te maken voor alle partijen waar de timetable applicatie staat in het proces rondom het automeren van stalprogramma’s.

TimeTables moet verschillende inputmethodes krijgen: Handmatige input is belangrijk. Maar ook een automatische door middel van een API, deze moet data halen uit verschillende computers in een stal. Deze API valt buiten de scope van het TimeTables project.

Vervolgens moeten er ook verschillende exportmethodes ontwikkeld worden. Een Export API heeft een connectie met de voorheen genoemde computers in een stal. Ook moet er een Export PNG functie komen waarmee de gebruiker een visuele representatie van het dagschema als PNG kan exporteren en delen.

Afbeelding met tekst, diagram, Plan, Lettertype

Automatisch gegenereerde beschrijving

Figuur 2 Dataflow Model

## Ontwikkel Omgeving

Er gaat gewerkt worden binnen bepaalde modules die hieronder staan genoteerd. Er staat ook bij waarom erin daarin wordt gewerkt en wat ze doen.

### Meggsius Connect

Meggsius Connect is een bestaande front-end applicatie die oorspronkelijk is ontwikkeld door een externe partij. Het doel van de module die ontwikkeld gaat worden, is om een toevoeging te zijn op de bestaande applicatie. Meer informatie over de structuur van deze applicatie is te vinden in het gedeelte [Architectuur en ontwerpprincipes](#_Architectuur_en_ontwerpprincipes).

### DFE-platform API

De DFE API heeft twee belangrijke doelen:

1. Het fungeert als een tussenlaag tussen de front-end en de SPF API. De SPF API heeft directe toegang tot de VDS-configuratie database. Er is gekozen voor deze tussenlaag omdat het front-end, zoals eerder vermeld, is ontwikkeld door een externe partij. Op deze manier kan worden voorkomen dat het externe bedrijf wijzigingen aanbrengt in de API die door het softwareteam van Vencomatic zelf is ontwikkeld.
2. Het maakt verbinding met de Meggsius Connect database, waarin gegevens zijn opgeslagen die alleen relevant zijn voor de Meggsius Connect front-end.

### SPF API

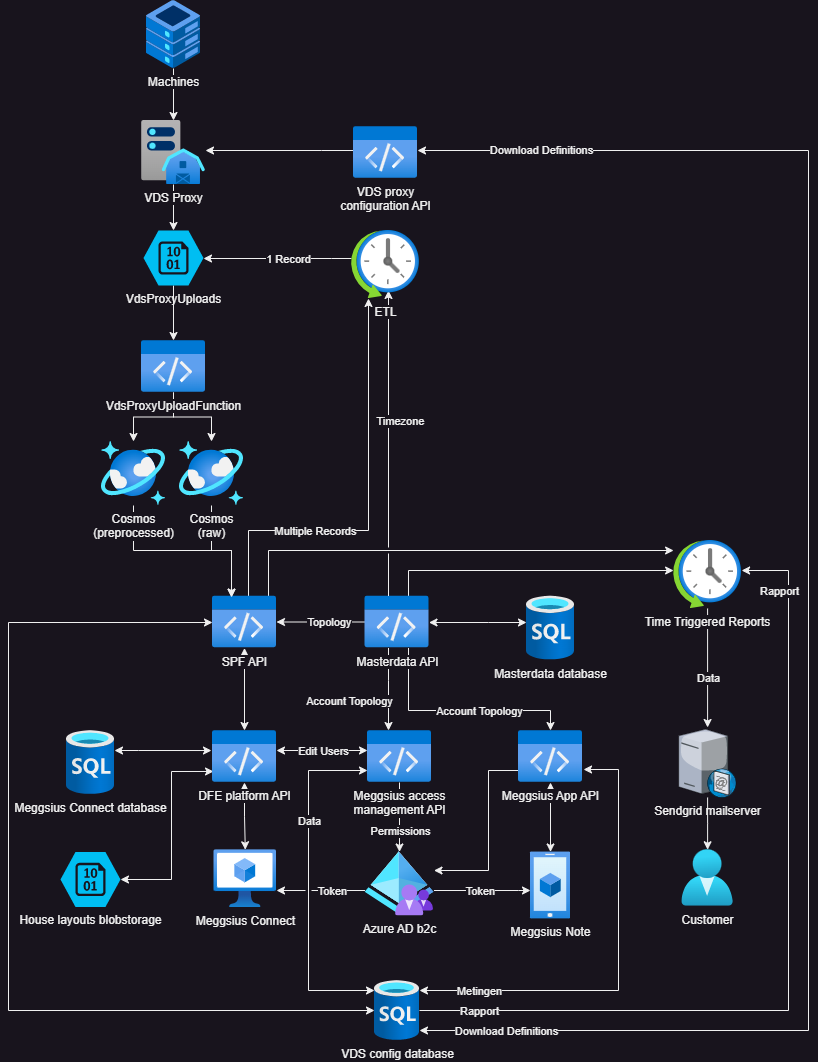
De SPF API heeft verschillende doelen:

1. Het maken van een verbinding met de VDS-configuratie database. Dit is het meest relevant voor dit project.
2. Het ophalen van accounttopologie door middel van een verbinding met de Masterdata API.
3. Het ophalen van gegevens zoals machinedata uit Cosmos en deze doorgeven aan de VDS-configuratie database.

Deze API wordt onder andere aangesproken door de DFE API.

### VDS config database

In deze database worden alle tabellen toegevoegd die relevant zijn voor dit project. Het dient als een algemene database die kan worden aangesproken door verschillende processen en API’s, zoals de SPF API, Meggsius access management API en Meggsius App API, elk met hun eigen doelen.



Figuur 3 Ontwikkel Omgeving

## Architectuur en ontwerpprincipes

Aangezien er een module wordt ontwikkeld binnen een applicatie die al voor een gedeelte af is wordt er verder gewerkt met dezelfde ontwerpprincipes en architectuurkeuzes. Er zal gewerkt worden binnen het Angular framework.

Angular maakt gebruik van TypeScript, een programmeertaal ontwikkeld door Microsoft die een bovenlaag vormt boven JavaScript. TypeScript voegt statische typering en andere functies toe aan JavaScript, wat de ontwikkeling van grote, complexe systemen kan vergemakkelijken. (*What is Angular?* Z.d.)  
Enkele redenen waarom er is gekozen voor deze Angular en Typescript zijn:

1. **Statische Typering**: TypeScript heeft statische typering die kan helpen bij het opsporen van fouten tijdens het ontwikkelingsproces in plaats van tijdens de uitvoering.
2. **Component-Based Architecture**: Angular gebruikt een component-based architectuur die het gemakkelijk maakt om herbruikbare, modulaire code te schrijven. Dit maakt de code beter te onderhouden en te testen.
3. **Tools en Ecosystem**: Angular heeft een uitgebreid ecosysteem met veel hulpmiddelen en bibliotheken, en wordt ondersteund door Google, wat vertrouwen en stabiliteit biedt.
4. **Two-Way Data Binding**: Angular ondersteunt two-way data binding, wat betekent dat de UI-elementen automatisch worden bijgewerkt wanneer de onderliggende data verandert, en vice versa.
5. **Support for Single Page Application**: Angular is ideaal voor het bouwen van complexe single page applications, waarbij de gebruiker een soepele ervaring heeft zonder de pagina te hoeven herladen. Dit is ook zeker handig voor het timetables project.

Tijdens het ontwikkelen van een project kan het vaststellen van ontwerpprincipes bijdragen aan het efficiënter maken van het ontwikkelproces

1. **Modulariteit**: TimeTables is ontworpen met een duidelijke scheiding tussen de UI, de backend services, en de database laag. Elk onderdeel is een onafhankelijke module die afzonderlijk kan worden ontwikkeld en onderhouden.
2. **Duidelijke Interfaces**: De communicatie tussen de modules in het project gebeurt via duidelijk gedefinieerde API's. Dit zorgt voor een soepele integratie en minimaliseert afhankelijkheden tussen de modules.
3. **Onderhoudbaarheid**: Zoals eerder vermeld wordt TimeTables is ontworpen met het oog op toekomstige uitbreidingen. De code moet netjes georganiseerd en goed gedocumenteerd zijn, wat het onderhoud vergemakkelijkt.
4. **Bruikbaarheid**: TimeTables krijgt een intuïtieve en gebruiksvriendelijke interface. Het is ontworpen om de gebruiker te helpen efficiënter te werken, met functies zoals eenvoudige navigatie. Meer hierover in het onderzoek document.

## Referenties

*Vivek, J. (2023, 8 mei). Een softwareontwerpdocument maken | Zuci Systems. Zuci Systems. Geraadpleegd op 1 juni 2023, van* [*https://www.zucisystems.com/nl/blog/een-softwareontwerpdocument-maken/*](https://www.zucisystems.com/nl/blog/een-softwareontwerpdocument-maken/)

*Wat is een datamodel? (Z.d.). Lucidchart. Geraadpleegd op 1 juni 2023, van* [*https://www.lucidchart.com/pages/nl/wat-is-een-datamodel*](https://www.lucidchart.com/pages/nl/wat-is-een-datamodel)

*ICT Portal. (2022, 29 maart). Hoe stel je een gegevensstroomdiagram (Data Flow Diagram) op? Geraadpleegd op 1 juni 2023, van* [*https://www.ictportal.nl/ict-lexicon/data-flow-diagram-dfd*](https://www.ictportal.nl/ict-lexicon/data-flow-diagram-dfd)

*What is Angular? (Z.d.). Angular. Geraadpleegd op 1 juni 2023, van* [*https://angular.io/guide/what-is-angular*](https://angular.io/guide/what-is-angular)

*Wat is Angular? (2021). Master IT. Geraadpleegd op 1 juni 2023, van* [*https://master-it.nl/blog/wat-is-angular/*](https://master-it.nl/blog/wat-is-angular/)