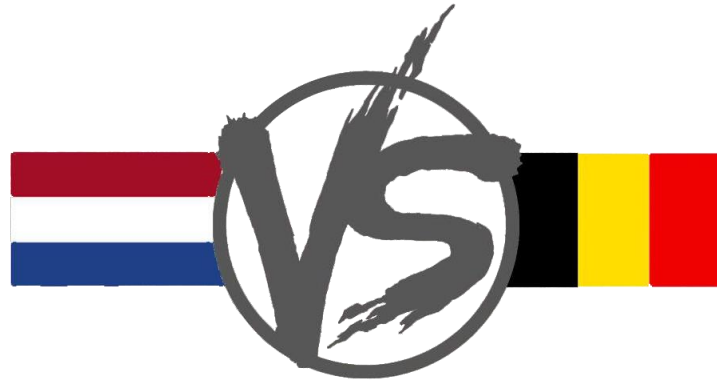




Professionele Bachelor Toegepaste Informatica



Energiedatamodel: België versus Nederland

Thomas Simaels

Promotoren:

Mevr. Ilse Melotte
Dhr. Lowie Vangaal

Nrgfin
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2017-2018



Professionele Bachelor Toegepaste Informatica



Energiedatamodel: België versus Nederland

Thomas Simaels

Promotoren:

Ilse Melotte
Lowie Vangaal

Nrgfin
Hogeschool PXL Hasselt



Bachelorpaper Academiejaar 2017-2018

Dankwoord

Bij Nrgfin heb ik een stageplaats gevonden, waarvoor mijn gemeente dank. In het bijzonder wil ik mevrouw Ilse Melotte hiervoor bedanken. Zij stond altijd klaar voor mij wanneer er vragen waren over de stageopdracht of het bedrijf zelf.

De collega's die ik heb leren kennen bij Nrgfin ben ik ook dankbaar, voor aangeboden gastvrijheid en omdat zowel management als collega's altijd ter beschikking waren wanneer ik allerlei vragen had, technologiegerelateerd of in verband met de interne processen of huisregels van Nrgfin.

Ook de mensen van KEN, de eindklant waarvoor ik Qlik-applicaties heb moeten ontwerpen, wil ik bedanken voor het vertrouwen dat ze me schonken. Ik mocht de applicaties voor de Nederlandse energiemarkt ontwikkelen op hun data. Tijdens het klantbezoek in Barendrecht bij Rotterdam waren de werknemers en het management van KEN ter beschikking. Op die manier heb ik veel kunnen bijleren over het Nederlands datamodel. Zij waren ook altijd beschikbaar via mail of telefoon wanneer ik nog vragen had of documentatie nodig had over bepaalde processen. Ook aan hen mijn diepgemeende dank. Hierdoor verliep mijn stage een heel stuk vlotter.

Als laatste wil ik graag de heer Lowie Vangaal bedanken voor de nuttige feedback en de vele tips die hij me gaf voor en tijdens de stage. Dankzij zijn raad en opvolging heb ik een nauwkeurig onderzoek kunnen opleveren en een vlotte stage kunnen genieten.

Dankzij de hulp en begeleiding van alle bovenstaande mensen heb ik een heel interessante en leervolle stageperiode achter de rug waarvoor mijn oprechte dank.

Abstract

Mijn stageopdracht omvat het bouwen van een aantal Qlik-applicaties voor de Nederlandse energiemarkt, vertrekkend vanaf bestaande Belgische energiefunctionaliteiten. De stage betrof het ontwerpen, converteren en het aanpassen van de code zodat deze gebruikt kan worden voor de Nederlandse energiemarkt.

Het datamodel voor de Nederlandse energiemarkt is heel verschillend van het Belgisch datamodel. Onze noorderburen gebruiken ander typen bestanden en gegevens. Hiervoor dienen bijgevolg nieuwe scripts en aangepaste code geschreven te worden. De achterliggende data in de applicaties moet ook aangepast worden of zelfs helemaal herschreven worden in een aantal gevallen.

In het onderzoek wordt er gedetailleerd gekeken naar het datamodel, de werkwijze en de databronnen die in Nederland worden gebruikt en worden deze vergeleken met het datamodel in België. De werkwijzen van beide landen worden naast elkaar gelegd en de databronnen verder geanalyseerd.

Een groot deel van dit onderzoek bestaat uit literatuurstudie. Het dataproces van Nederland wordt geanalyseerd, maar ook dat van België, om vervolgens beide diepgaand te kunnen vergelijken.

Buiten de literatuurstudie is er ook een praktische studie. Beide scripttalen, Powershell en Perl, worden op performantie en gebruik van computergeheugen getest op verschillende soorten en groottes van bestanden. Dit zijn onder andere:

- het filteren van bestanden en verplaatsen naar de aanliggende folder;
- Het converteren van bestandtypes (bijvoorbeeld: .txt -> .csv);
- Het samenvoegen van verschillende bestanden tot één groot eindbestand.

Inhoudsopgave

Dankwoord	ii
Abstract	iii
Inhoudsopgave	iv
Lijst van gebruikte figuren	vi
Lijst van gebruikte tabellen	vii
Lijst van gebruikte afbeeldingen	viii
Lijst van gebruikte afkortingen.....	ix
Inleiding.....	1
I. Stageverslag.....	2
1 Bedrijfsvoorstelling.....	2
2 Voorstelling stageopdracht	4
2.1 Stageopdracht	4
2.2 Doelstellingen.....	4
2.3 Omgeving & structuur	5
2.3.1 Qlik.....	5
2.3.2 SQL-server	6
2.3.3 Jira en Tempo	6
2.3.4 Slack.....	7
3 Uitwerking stageopdracht.....	8
3.1 Klanten totaal applicatie	8
3.1.1 Voorbereiding.....	8
3.1.2 Ontwikkeling van de applicatie (backend)	10
3.1.3 Ontwikkeling van de applicatie (frontend).....	11
3.1.4 Validatie van data	13
3.1.5 Herladen van de applicatie.....	14
3.2 Overige applicaties	14
3.2.1 Billed	14
3.2.2 Billing OPS.....	15
3.2.3 Aankoop.....	16
3.3 Opdrachten dagelijkse werking.....	17
3.3.1 Website en sociale media onderhouden.....	17
3.3.2 Grafisch werk.....	17
4 Besluit	18

II. Onderzoekstopic.....	19
1 Onderzoeksvraag.....	19
2 Onderzoeksmethode.....	19
3 Uitvoering.....	20
3.1 België.....	20
3.1.1 Dataproces.....	20
3.1.2 Datasources.....	24
3.1.3 IT-infrastructuur.....	26
3.2 Nederland.....	29
3.2.1 Dataproces.....	29
3.2.2 Datasources.....	30
3.2.3 IT-infrastructuur.....	30
3.3 Datatypes.....	32
3.3.1 XML.....	32
3.3.2 EDI.....	34
3.3.3 CSV.....	38
3.3.4 TXT.....	39
4 Scripttalen.....	40
4.1 Perl.....	40
4.2 Powershell.....	40
5 Testen: Powershell versus PERL.....	41
5.1 Voorbereiding.....	41
5.2 Powershell.....	42
5.2.1 Transformatie.....	43
5.2.2 Verplaatsing.....	44
5.2.3 Samenvoegen.....	45
5.3 Perl.....	46
5.3.1 Transformatie.....	49
5.3.2 Verplaatsing.....	50
5.3.3 Samenvoegen.....	50
5.4 Resultaten.....	52
Conclusie.....	55
Bibliografie.....	56
Bijlagen.....	57

Lijst van gebruikte figuren

Figuur 1 Infrastructuur voor ontvangen bestanden.....	26
Figuur 2 ETL-proces voor SQL-server.....	27
Figuur 3 ETL-proces voor QVD-bestanden	28
Figuur 4 Ontvangst Nederlandse data.....	30
Figuur 5 ETL-proces Nederlandse bestanden.....	31
Figuur 6 Boomstructuur XML	33

Lijst van gebruikte tabellen

Tabel 1 Verbruiksprofielen Belgische energiemarkt	20
Tabel 2 Verbruiksprofielen Nederlandse energiemarkt.....	29
Tabel 3 Voorbeeld EDI-bericht energiemarkt.....	34
Tabel 4 Voorbeeldfactuur energie	35
Tabel 5 Voorbeeldfactuur met data elementen.....	35
Tabel 6 Voorbeeldfactuur met segmenten	36
Tabel 7 Voorbeeld CSV-bestand	38

Lijst van gebruikte afbeeldingen

Afbeelding 1 Locatie Nrgfin	2
Afbeelding 2 Organisatiestructuur van Nrgfin	3
Afbeelding 3 Qlik Desktop dashboard	5
Afbeelding 4 Jira Kanban	6
Afbeelding 5 Tempo planning en actuals	7
Afbeelding 6 Slack communicatiegroep	7
Afbeelding 7 Datamodel 2 XML's	8
Afbeelding 8 Powershell script voor samenvoeging XML	9
Afbeelding 9 Klantenbestanden	10
Afbeelding 10 Voorbeeld dashboard Qlik	11
Afbeelding 11 Berekening met variabele	11
Afbeelding 12 Drilldown: klant naar contract	12
Afbeelding 13 Mastervisualisatie filter	13
Afbeelding 14 Filterbalk	13
Afbeelding 15 Tabel export in Qlik	13
Afbeelding 16 Export getoond in Excel	13
Afbeelding 17 KPI-dashboard Billed	14
Afbeelding 18 Uitleg Billing OPS termen	15
Afbeelding 19 Dashboard Billing OPS	15
Afbeelding 20 PV en leveranciersentiteit	16
Afbeelding 21 Backend website Comigoo	17
Afbeelding 22 Enkelvoudige meter	21
Afbeelding 23 Dubbele meter	22
Afbeelding 24 Gasmeter	23
Afbeelding 25 SFTP-shema	24
Afbeelding 26 Voorbeeld interne structuur XML	32
Afbeelding 27 Versleutelingsregel	34
Afbeelding 28 Model over verzending EDI	38
Afbeelding 29 Powershell script tijdsduurmeting	42
Afbeelding 30 Powershell script procesinformatie	42
Afbeelding 31 Powershell script transformatie	43
Afbeelding 32 Powershell script verplaatsing	44
Afbeelding 33 XML-berichten en mappenstructuur	44
Afbeelding 34 Powershell script samenvoeging	45
Afbeelding 35 Module succesvol geïnstalleerd	46
Afbeelding 36 Perl script tijdsduurmeting	47
Afbeelding 37 Perl script procesinformatie	48
Afbeelding 38 Perl script transformatie	49
Afbeelding 39 Perl script verplaatsing	50
Afbeelding 40 Perl script samenvoeging	51
Afbeelding 41 KPI's gemiddelde tijd	52
Afbeelding 42 Lijngrafiek gemiddelde tijd	52
Afbeelding 43 KPI's gemiddeld geheugenverbruik	53
Afbeelding 44 Lijngrafiek gemiddeld geheugenverbruik	53
Afbeelding 45 KPI's gemiddeld CPU verbruik	54
Afbeelding 46 Lijngrafiek gemiddeld CPU verbruik	54

Lijst van gebruikte afkortingen

Afkorting	Begrip	Uitleg
BI	Business Intelligence	Business Intelligence is het geheel van oplossingen en technieken voor het verzamelen, opslaan en analyseren van bedrijfskritische data uit uiteenlopende bedrijfsplatformen.
CSV	Comma seperated values	Een tabelbestand dat voornamelijk in Excel gebruikt wordt. In de bestand zijn de waardes per rij achtereen geplakt met een speciaal karakter tussen iedere waarde. Meestal is dit een komma (,) maar kan ook een koppelteken (-) of puntkomma (;) zijn.
DEV	Development	Het ontwikkelen van een bepaalde applicatie.
EABO	EnergyAssets BackOffice	Een Nederlands bedrijf dat administratieve bijdrages levert aan andere bedrijven. Zij zorgen ervoor dat de data binnen het bedrijf georganiseerd blijft en makkelijk te exporteren/importeren is voor uitwisseling. Tijdens dit verslag zal er dus vaak gesproken worden over EABO-bestanden, oftewel bestanden die door EABO geleverd worden.
EAN	European Article Numbering	Een EAN-code is een unieke cijfercombinatie waarmee een voorwerp op basis van een Europees systeem wordt geïdentificeerd. Dit kan gelden voor een product, een leverancier enz. EAN-codes worden ook gebruikt om een leveringspunt voor gas of elektriciteit te identificeren en tellen in dat geval achttien cijfers in plaats van dertien.diele
EDI	Electronic Data Interchange	EDI is een standaard dat is opgesteld voor het elektronisch uitwisselen van bedrijfsdocumenten. Het is een soort template die opgesteld is en waaraan de bedrijven die dit standaard gebruiken zich aan moeten houden.
EDIEL	Electronic Data Interchange in de Electricity Industrie	EDIEL-berichten zijn een meer gespecificeerde vorm van EDI-berichten. EDIEL is een standaard dat is ontworpen specifiek voor de energiemarkt en dit voor heel België.
EDSN	Energie Data Services Nederland	Een Nederlands bedrijf dat energiebedrijven helpt om een IT-infrastructuur te ontwerpen.
ETL	Extract, Transform, Load	Het benoemt de processen die gebruikt worden om gegevens uit verschillend gestructureerde databases te verenigen in een andere database, veelal een datawarehouse. Het gaat de gegevens ophalen, transformeren en inladen.

INVOIC	INVOIC	Het INVOIC bericht wordt gebruikt voor het communiceren van de details aan de facturen met betrekking tot het gebruik van het distributienetwerk
KEN	Kleinverbruik energie der Nederlanden	Nederlands bedrijf met als activiteiten: handel in elektriciteit en in gas via leidingen. Deze branche heeft als hoofdcategorie: 'Productie en distributie van en handel in elektriciteit, aardgas, stoom en/of gekoelde lucht' en is in dit geval verder onderverdeeld bij: 'Productie en distributie van en handel in elektriciteit, aardgas, stoom en gekoelde lucht', subcategorie 'Productie van elektriciteit; transmissie en distributie van elektriciteit en aardgas'.
kVA	Kilovoltampère	Eenheid voor het elektrisch vermogen bij elektrisch apparaat.
kWh	Kilowattuur	Eenheid die het hoeveelheid energie per uur weergeeft dat een elektrisch apparaat nodig heeft.
PV	Provincie verantwoordelijke	De taken van een provincie verantwoordelijke is om ervoor te zorgen dat er voor zijn of haar aangewezen provincie een juiste hoeveelheid stroom en gas aangekocht en verkocht wordt. De bedoeling hiervan is dus dat er niet teveel wordt aangekocht want dit veroorzaakt een verlies. Aan de andere kant mag er niet te weinig worden aangekocht want dan kan niet iedereen in de provincie beleverd worden.
QVD	QlikView Data	Een QVD bestand is een datatabel die geëxporteerd is vanuit Qlik zelf. Een QVD bestand staat al in het Qlik formaat waardoor deze sneller de data zal inlezen en laden in het script.
SLP	Synthetic Load Profiles	Het is een curve of cijfertabel die per kwartier (voor elektriciteit) of per uur (voor aardgas) over een volledig jaar het relatieve verbruik weergeeft voor een bepaald type van klanten (dus uitgedrukt in% waarbij de som van alle 100% of 1 is.
SFTP	Secure File Transfer Protocol	De SFTP gaat een beveiligde verbinding maken tussen twee digitale plaatsen waarin bestanden uitgewisseld kunnen worden.
UTILMD	UTILities Master Data message	UTILMD zal gebruikt worden voor het uitwisselen van relatief stabiele gegevens tussen verschillende partijen binnen de energiemarkt.
UTILTS	UTILities Time Series message	Voor transactionele uitwisselingmeetgegevens wordt het UTILTS bericht gebruikt. Een "Time Series" bevat een opeenvolging van controles van één enkel proces, meestal genomen met gelijke tussenpozen.

XML	eXtensible Markup Language	XML is ontworpen voor het bewaren en het transporteren van data. Het kan verschillende soorten gegevens eenvoudig en makkelijk versturen, dit kan gaan over financiële data tot een geschreven tekst van een eindwerk bijvoorbeeld.
------------	----------------------------	---

Inleiding

In de informaticawereld veranderen er elke dag processen en technologieën. IT-bedrijven moeten steeds mee zijn met de nieuwste technische toepassingen. Grote verandering in een bedrijf of proces gebeuren in projectmodus. Elk project heeft een goede voorbereiding nodig. Zonder een diepgaande voorbereidende fase lopen projecten vaak vertraging op.

Tijdens de stageperiode zijn er nieuwe applicaties ontwikkeld moeten worden voor een nieuwe Nederlandse Nrgfinklant. Doordat er in Nederland een ander datamodel is als België dient er een goede analyse te gebeuren van de 'Big Data' dat aangeleverd wordt. Nieuwe scripts, zowel backend als frontend, zijn aan de hand van de scripts voor Belgische klanten geschreven en aangepast waar er veranderingen zijn tussen beide datamodellen.

Tegenwoordig zijn er ook zoveel verschillende soorten programma's en programmeertalen om dagelijkse berekeningen en taken uit te voeren dat we niet meer weten welke de beste of de snelste zijn voor het bedrijf. Hierbij is tijd één van de belangrijkste criteria - al dan niet het belangrijkste – op basis waarvan het bedrijf beslissingen neemt. "Tijd is geld" horen we vandaag de dag in de bedrijfswereld.

Hiervoor is het dus van groot belang dat er intern in een bedrijf zo snel mogelijk wordt gewerkt, dat betekent op een zo efficiënt mogelijke manier.

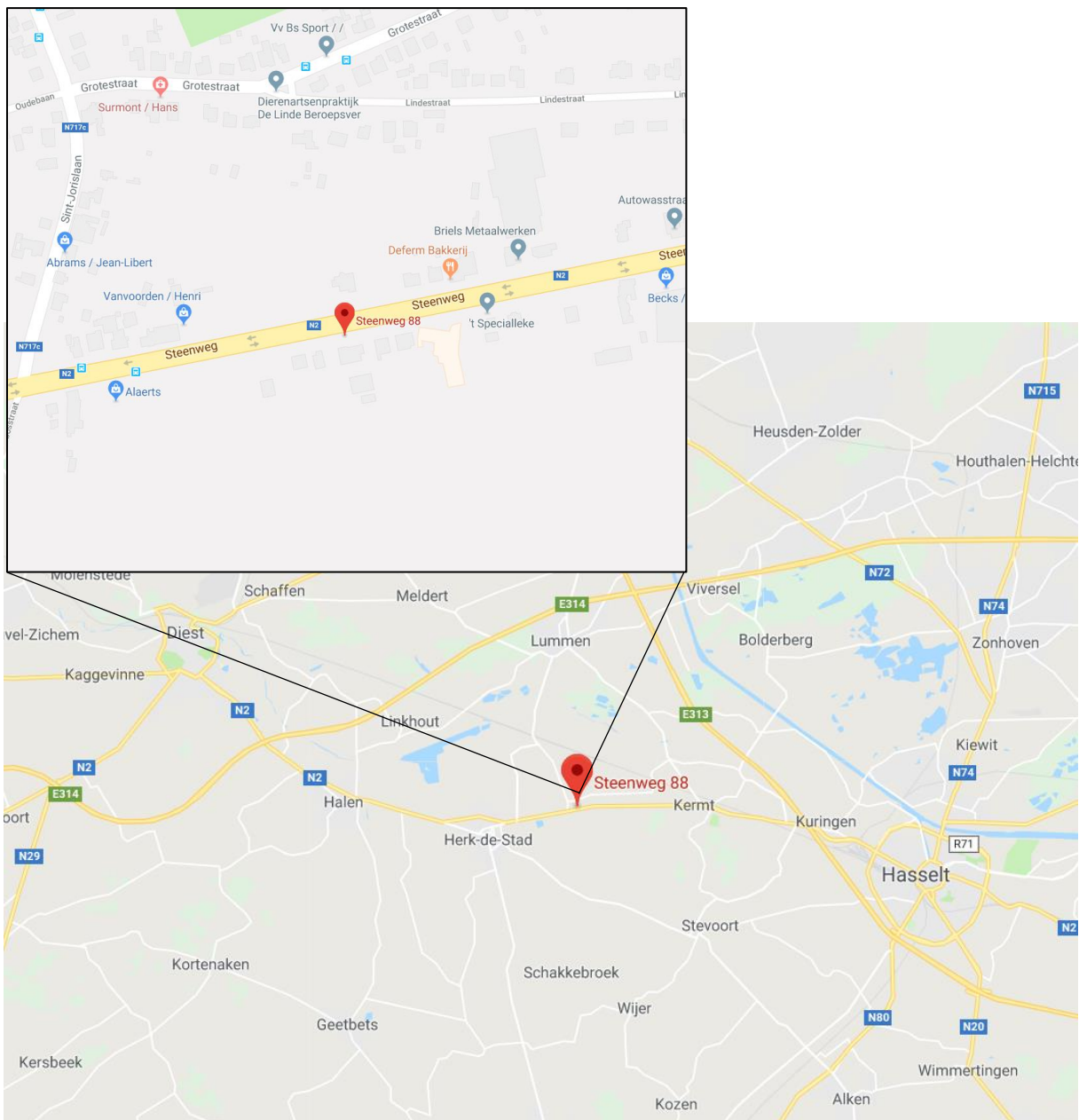
Tijdens dit onderzoek leert u het energiedatamodel van België en Nederland kennen en wordt u op de hoogte gesteld over de gelijkenissen en vooral de verschillen tussen beide datamodellen. De programmeertalen Perl en Powershell zijn ook vergeleken met elkaar op snelheid en verbruik van computergeheugen.

Uit het laatste praktijkonderzoek heb ik een conclusie kunnen vormen, via een aantal uitgevoerde testen, omtrent welke programmeertaal het snelste en het efficiëntst werkt. Dit kan een bedrijf drastisch helpen op het gebied van besteding van werkuren. En dat betekent kostenbesparing.

I. Stageverslag

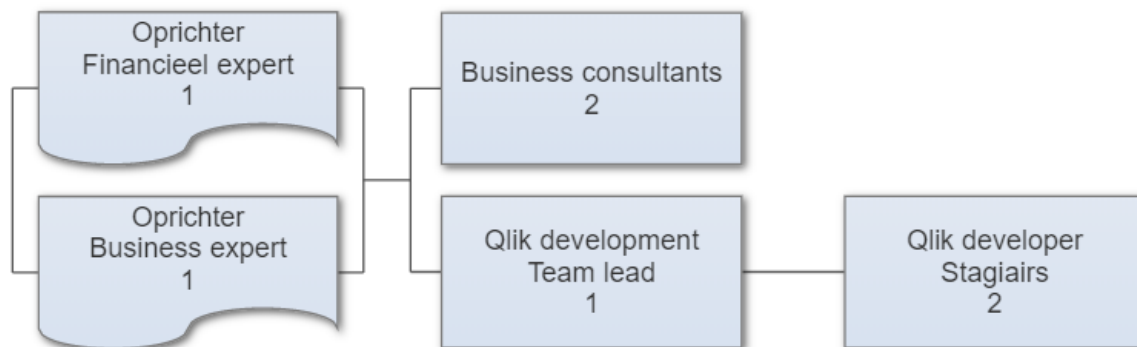
1 Bedrijfsvoorstelling

Nrgfin is een onderneming gelegen in Berbroek [Afbeelding 1] bestaande uit 5 werknemers. Het bedrijf biedt strategische consulting en businessanalyseoplossingen aan energieleveranciers. De klanten kunnen via het Livello platform verschillende soorten applicaties gebruiken waardoor ze inzicht verkrijgen in hun strategische, financiële en operationele data en dit op basis van visualisaties en tabellen. Data afkomstig uit verschillende bronnen wordt getransformeerd in MS SQL en geanalyseerd, geïntegreerd en gevisualiseerd in Qlik Sense, een softwareoplossing die dient om big data om te zetten tot nuttige en overzichtelijke informatie.



Afbeelding 1 Locatie Nrgfin

Een handig weetje is dat Nrgfin pas drie jaar geleden opgericht werd in België en dat het bedrijf momenteel al aan het uitbreiden is naar Nederland. Op dit moment bestaat Nrgfin uit één business en één financieel expert, twee energiebusiness consultants, één team lead en twee stagiairs. Zoals te zien is op onderstaande afbeelding [Afbeelding 2] is de hiërarchie niet verticaal maar horizontaal, wat een heel aangename manier van werken is. Op de werkvloer heerst een heel familiale werksfeer is. Mevrouw Melotte (oprichter) gedraagt zich niet als een baas die iedereen commandeert en zichzelf boven iedereen ziet staan. Integendeel, ze ziet in iedere werknemer zijn of haar unieke kwaliteiten en behandelt iedereen gelijk.



Afbeelding 2 Organisatiestructuur van Nrgfin

Voordat de stage begon heb ik zeven maanden bij Nrgfin gewerkt als jobstudent. Tijdens deze periode werd ik vertrouwd met de werkwijzen, structuren en applicaties binnen Nrgfin. Hierdoor werd ik zelf ook wel al opgenomen in de dagelijkse werking van Nrgfin. Concreet wil dit zeggen dat tijdens de stage er nog taken uitgevoerd moesten worden die niets te maken hebben met de stageopdracht maar met de dagelijkse werking van Nrgfin, zoals het herladen van bestaande apps voor hun Belgische energiekanten. Ook was ik verantwoordelijk voor het onderhoud van website en social media.

2 Voorstelling stageopdracht

2.1 Stageopdracht

De opdracht die uitgevoerd is tijdens de stageperiode bestaat erin om een aantal Qlik-applicaties, die voor de Belgische energiemarkt werden ontworpen, te converteren zodat deze gebruikt kunnen worden voor de Nederlandse energiemarkt. Concreet betreft dit dus het aanpassen van bepaalde scripts of achterliggende code. Het kan ook zijn dat er bepaalde benamingen in de bestanden aangepast zijn waar rekening mee moet gehouden worden of dat er visualisaties aangepast moeten worden.

Voor de Nederlandse klant 'Kleinverbruik Energie der Nederlanden' (KEN) zullen er drie applicaties ontwikkeld worden op basis van hun datagegevens.

Buiten de opgelegde stageopdracht is het de bedoeling dat de andere taken die voor Nrgfin geleverd moeten worden nog steeds uitgevoerd worden. Dit zijn onder andere: de website onderhouden, nieuwe blogs ontwerpen, de sociale media updaten en grafisch design ontwerpen.

2.2 Doelstellingen

De belangrijkste doelstelling is ervoor te zorgen om nieuwe Qlik-applicaties te bouwen die gebruikt kunnen worden voor de Nederlandse energiemarkt. Momenteel bestaan er applicaties die voor België gebruikt worden. Deze moeten dus geconverteerd worden zodat deze ook voor de Nederlandse energiemarkt in gebruik kunnen worden genomen.

Een andere doelstelling is professioneel om te gaan met de aangeleverde data van de Nederlandse klant. Hiermee wordt bedoeld dat er geen informatie wordt gelekt en dat de gegevens ten alle tijde bewaard worden op een veilige omgeving.

Als laatste moet er een validatie komen voor elke applicatie en zijn scripts. Deze scripts moeten ook nauwkeurig getest worden. Als er eventueel een error voorkomt, zal deze opgevolgd en verholpen worden.

2.3 Omgeving & structuur

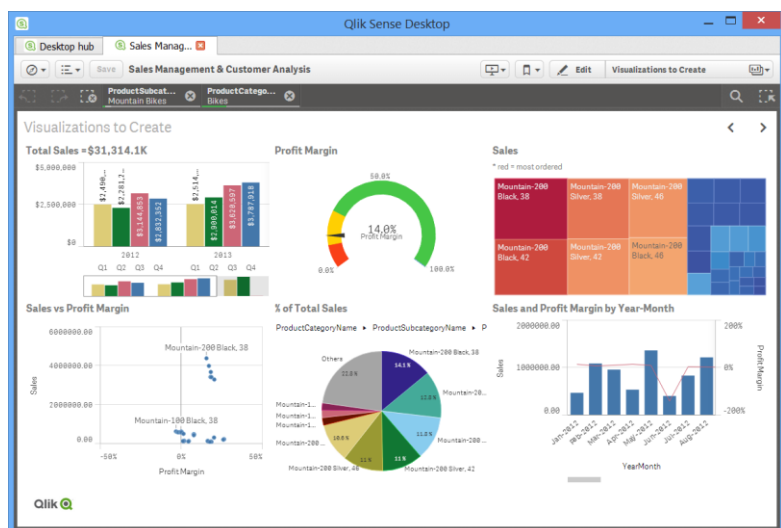
De stageopdracht zal uitgevoerd worden in een Qlik-omgeving. De aangeleverde data behandelen en het voorbereidend werk zal in de backend gebeuren door middel van SQL- en Qlikcode, het ontwikkelen en visualiseren van de applicaties in de frontend.

2.3.1 Qlik

Qlik is een platform dat businessanalyseoplossingen levert aan bedrijven van alle sectoren. Het vergemakkelijkt het conversieproces van data uit alle soorten bronnen naar meer begrijpelijke en inzichtelijke visualisaties en dashboards. Met Qlik is het heel eenvoudig om flexibele en vooral interactieve datavisualisaties te maken met het handige drag-and-dropsysteem. Hiermee kunnen visualisaties waarvan een template is opgeslagen eenvoudig worden toegevoegd aan een dashboard en moet er enkel nog meegegeven worden welke velden of functies er getoond moeten worden.

Tijdens de stageopdracht wordt er gebruikgemaakt van twee Qlik- producten: Qlik Sense Desktop (versie november 2017) en Qlik Sense Enterprise (versie 3.2)

Qlik Sense Desktop [Afbeelding 3] is de lokale versie van Qlik en is gratis te downloaden. Deze versie wordt voornamelijk gebruikt om gegevens van de applicaties snel te valideren. Verder kan hierin, wanneer er een nieuwe applicatie moet aangemaakt worden, eerst een voorbereiding worden gemaakt, zodat de server niet onnodig wordt belast.



Afbeelding 3 Qlik Desktop dashboard

Het tweede product dat gebruikt wordt bij Nrgfin is Qlik Sense Enterprise, oftewel Qlik Server. Via dit product kunnen klanten de applicaties die voor hen aangemaakt zijn, bekijken en gebruiken. Wat betreft functionaliteiten zijn Qlik Sense Desktop en Qlik Server bijna hetzelfde, alleen is het zo dat de desktopversie lokaal en de serverversie online staat. Sommige visualisaties werken niet in beide programma's.

2.3.2 SQL-server

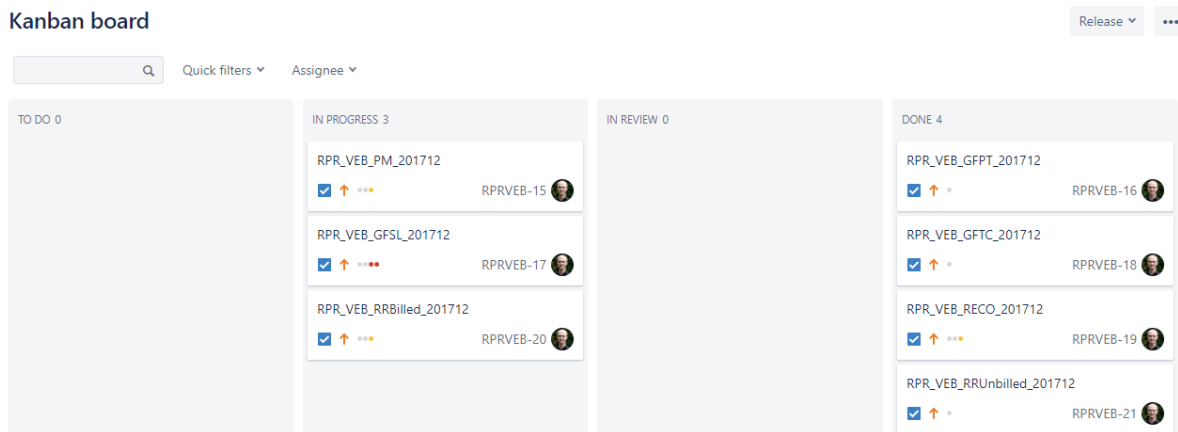
SQL-server wordt gebruikt om alle ontvangen data in op te slaan. Hierdoor is er een back-up van de data en is het makkelijk werken wanneer er bepaalde data nodig is, omdat deze op één plaats verzameld is.

Momenteel wordt bij Nrgfin SQL server 2016 gebruikt met versie 13.0.4451.0.

In deze server worden alle Belgische EDI-berichten opgeladen zodat op deze gegevens berekeningen en functionaliteiten kunnen uitgewerkt worden voor de energiemarkt. Tijdens mijn stageperiode heb ik geen SQL server gebruikt omwille dat KEN geen Belgische klant is en XML-bestanden aanlevert, deze kunnen nog niet in de server geladen worden.

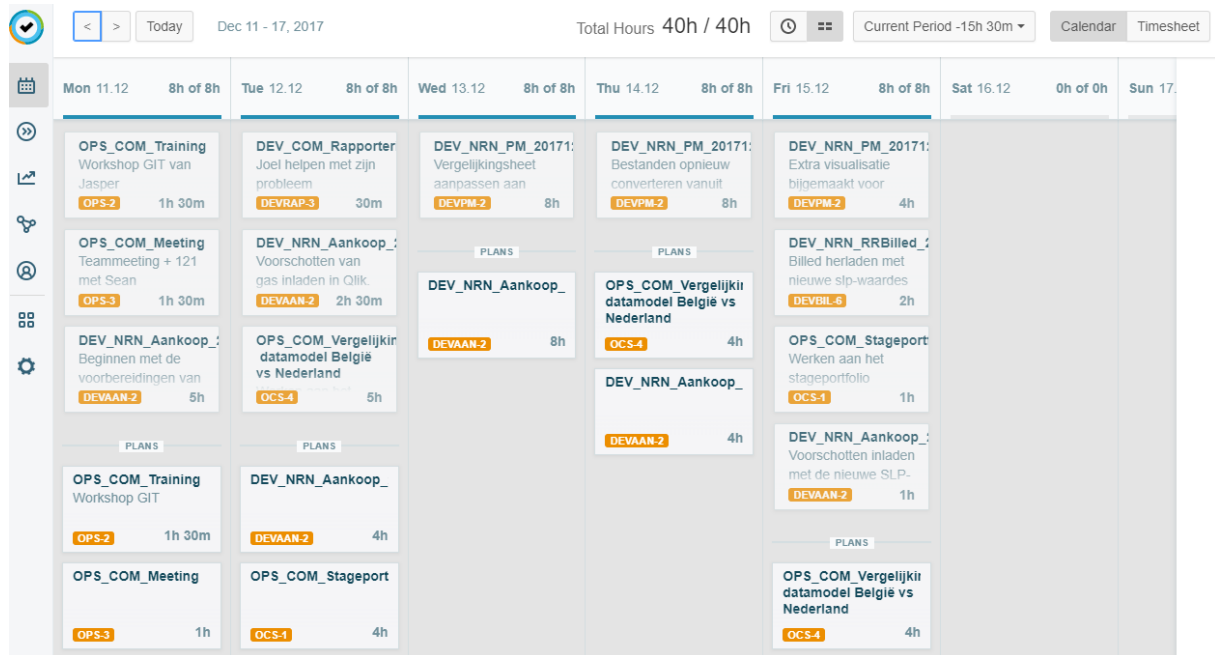
2.3.3 Jira en Tempo

Voor de interne structuur en opvolging wordt er gebruikgemaakt van Jira en Tempo als projectmanagementsysteem. Jira is een krachtige tool waarin projecten en issues aangemaakt kunnen worden. [Afbeelding 4] Issues zijn ticketjes die elke taak beschrijven die nog moeten gebeuren om een project of product te voltooien. Er kunnen verschillende en/of meerdere medewerkers gekoppeld worden aan deze projecten en issues.



Afbeelding 4 Jira Kanban

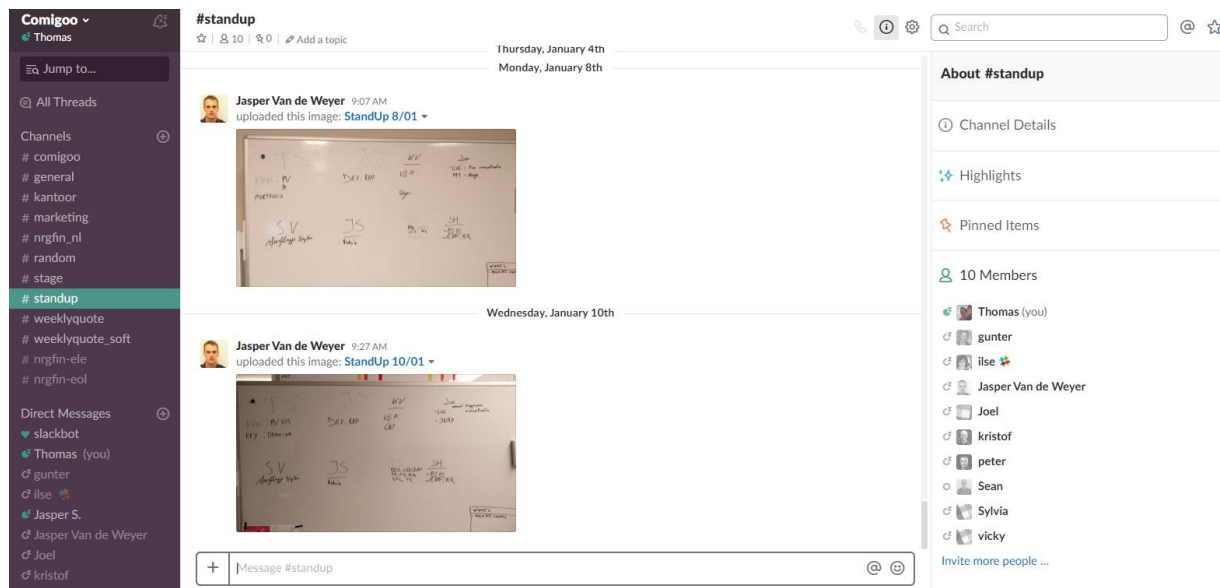
Tempo is een extensie voor Jira waarin er op de projecten en issues gepland en gelogd kan worden. Op deze manier wordt er bijgehouden wie op welk project en welk issue gepland is en hoeveel er effectief op is gewerkt. [Afbeelding 5]



Afbeelding 5 Tempo planning en actuals

2.3.4 Slack

Als communicatiemiddel binnen het bedrijf wordt er gebruikgemaakt van Slack. Slack is een klein communicatieprogramma waarin medewerkers berichten naar elkaar kunnen sturen, maar waarin ook groepen aangemaakt kunnen worden waarin meerdere medewerkers zitten. Dit programma wordt vooral gebruikt wanneer er vragen gesteld moeten worden. [Afbeelding 6]



Afbeelding 6 Slack communicatiegroep

3 Uitwerking stageopdracht

Aangezien er verschillende applicaties ontwikkeld zijn voor de Nederlandse klant, zal hieronder de belangrijkste applicatie gedetailleerd beschreven worden. De rest wordt enkel summier toegelicht.

3.1 Klanten totaal applicatie

De eerste applicatie die ontworpen wordt voor de Nederlandse klant is om inzicht te krijgen in de bewegingen van de klantenportefeuille. Dit wordt de ‘Klantentotaal’ –applicatie genoemd. In deze applicatie wordt alle data opgeladen die klantgegevens bevatten. Er zijn verschillende types bestanden (CSV en XML) die ingeladen moeten worden. Deze bestanden moeten op een correcte manier gelinkt worden aan elkaar, zodat de data in de applicatie interactief en correct is.

3.1.1 Voorbereiding

Aan het begin van elke ontwikkeling moet er een voorbereiding zijn. De requirements van de applicatie worden genoteerd en de verschillende soorten bestanden die zijn aangeleverd worden geanalyseerd. Van elk type bestandstype, werd een datamodel gemaakt zodat het later gemakkelijker is om te weten welke data te vinden is in de verschillende soorten bestanden.

[Afbeelding 7]

GBU DRQ ATT			GBU DRQ		
Name Field	Structure	Example	Name Field	Structure	Example
CreationTimestamp	DATE(DD/MM/YYYY HH:MM)	1/08/2017 10:00	CreationTimestamp	date(YYYY-MM-DD timezone)	2017-03-02T03:03:07.757+01:00
ReceiverID	varchar(13)	1234567891234	Authority	varchar	EAN.UCC
Service	[empty]		ContactTypeIdentifier	varchar	DDQ_M
MimeTypeQualifierCode	[empty]		ReceiverID	varchar(13)	8,52159E+12
UniformResourceIdentifier	[empty]		MimeTypeQualifierCode	varchar	tekst/plain
NumberOfItems	number	1	UniformResourceIdentifier	varchar	/u02/PRD01/data/files/fileexchan
MessageID	varchar(6)	545454	NumberOfItems	number	1
SenderID	varchar(13)	9,85237E+12	MessageID	varchar	9450b662-1bb0-4eea-a0dd-3f328
ID	varchar(13)	8,52148E+12	Authority	[empty]	
DateFrom	DATE(DD/MM/YYYY)	1/08/2016	ContactTypeIdentifier	varchar	EDSN
DateTo	DATE(DD/MM/YYYY)	1/08/2017	SenderID	varchar(13)	8,52259E+12
Currency	varchar	EUR	FileID	varchar	0510537e-50ba-42be-9eb9-887f5
AmountExVAT4Decimal	number	410,2549	FileSize	number	546546
AmountType	varchar	TYPES	FileType	varchar	KOS Verplichting
VATNumber	varchar	NL987654321B01	Group	varchar	ADC
ID	varchar	12345	OrganisationID	varchar(13)	8,52137E+12
EANID	varchar(18)	8,52365E+17	OrganisationName	varchar	energiemaatschappij
DayTariff	number	1,2345	OrganisationID	varchar(13)	8,7459E+12
AmountExVAT4Decimal	number	15,6543	OrganisationName	varchar	EDSN
AmountType	varchar	TYPES	Status	varchar	opgehaald
TotalDays	number	30	DateTime	date(YYYY-MM-DD timezone)	2017-03-02T03:03:07.757+01:00
CapTarCode	varchar(13)	8,52357E+12			

Afbeelding 7 Datamodel 2 XML's

Sommige bestanden bestaan uit meer dan 5000 berichten die allemaal gebruikt moeten worden. Om dit op een efficiënte manier te kunnen gebruiken in Qlik zijn er scripts geschreven in Powershell om bestanden van hetzelfde type en dezelfde soort samen te voegen. Hierdoor is er maar één bestand dat wordt opgeladen in Qlik. Zo een bestand wordt ook wel een masterbestand genoemd.

Er worden verschillende soorten scripts aangemaakt naargelang de functionaliteit die nodig is. De twee voornaamste scripts die gebruikt zijn tijdens de stage, dienen om enerzijds bestanden te verplaatsen naar hun desbetreffende plaats op de server en anderzijds om de bestanden die hetzelfde type en structuur hebben samen te voegen tot één masterbestand. [Afbeelding 8]

```
$xmlDoc = new-object xml
$merged = "merged"
$DestinationRoot= "$pwd\"
$rootnode = $xmlDoc.createelement("xml")
$xmlDoc.appendChild($rootnode)
$finalxml = $null
$files = gci *.xml
$count =0

foreach ($file in $files) {
    [xml]$xmlstuff = gc $file.fullname
    $innerel = $xmlstuff.selectnodes("/*/*")
    $count++
    foreach ($inone in $innerel) {
        $inone = $xmlDoc.importnode($inone, $true)
        $rootnode.appendChild($inone)
    }
    New-Item "$DestinationRoot$last" -ItemType Directory -Force | Out-Null
    Move-Item -Path $file.fullname -Destination $Destination -Force
    "file $file moved"
}
foreach ($t34 in $rootnode.selectnodes("//*/text()")) {
    $t34.innertext = [regex]::replace($t34.innertext,"\\s+"," ")
}

# create and set xmlwritersettings
$xws = new-object system.xml.XmlWriterSettings
$xws.Indent = $true
$xws.indentchars = "`t"
$xtw = [system.xml.XmlWriter]::create("$pwd\\final_Master.xml", $xws)
$xmlDoc.WriteContentTo($xtw)
$xtw.flush()
$xtw.dispose()

"bestanden gemerged: $count"
Read-Host -Prompt "Press Enter to exit"
```

Afbeelding 8 Powershell script voor samenvoeging XML

De verschillende klantbestanden [Afbeelding 9] die nodig zijn voor de ‘Klantentotaal’ applicatie worden verzameld. In deze bestanden wordt onderzocht welke velden er nodig zijn en welke berekeningen er eventueel moeten worden gebruikt. Niet alle velden uit de bestanden worden opgeladen omdat dit een vertraging zou betekenen voor de applicatie. Het is onnodig om de applicatie te vullen met data die niet gebruikt wordt.

Agreements	14/12/2017 11:46	Bestandsmap
ClientAdresses	15/12/2017 12:06	Bestandsmap
ContractedDays	14/12/2017 11:46	Bestandsmap
Contracts	14/12/2017 11:45	Bestandsmap
EABO_All_201611.qvd	15/12/2017 11:39	QVD-bestand
EABO_All_201612.qvd	15/12/2017 11:40	QVD-bestand
EABO_All_201701.qvd	15/12/2017 11:40	QVD-bestand
EABO_All_201702.qvd	15/12/2017 11:41	QVD-bestand
EABO_All_201703.qvd	15/12/2017 11:41	QVD-bestand
EABO_All_201704.qvd	15/12/2017 11:42	QVD-bestand
EABO_All_201705.qvd	15/12/2017 11:42	QVD-bestand
EABO_All_201706.qvd	15/12/2017 11:43	QVD-bestand
EABO_All_201707.qvd	15/12/2017 12:23	QVD-bestand
EABO_All_201708.qvd	17/12/2017 19:57	QVD-bestand

Afbeelding 9 Klantenbestanden

3.1.2 Ontwikkeling van de applicatie (backend)

Het ontwikkelen van een Qlik -applicatie wordt meestal in twee stappen opgesplitst: backend en frontend. Eerst wordt er achterliggend geprogrammeerd. Hierin wordt er gezorgd dat enkel de data die nodig is voor de applicatie wordt opgeladen. Dit gebeurt voornamelijk door SQL-code met bepaalde functies die ontwikkeld zijn door Qlik zelf. Hierin worden ook de meeste berekeningen en functies aangemaakt zodat deze niet meer in de visualisaties moeten worden gecreëerd.

De data kan op verschillende manieren worden opgehaald. Dit kan rechtstreeks van de bestanden die zijn aangeleverd of er kan een verbinding worden gemaakt naar een SQL-database.

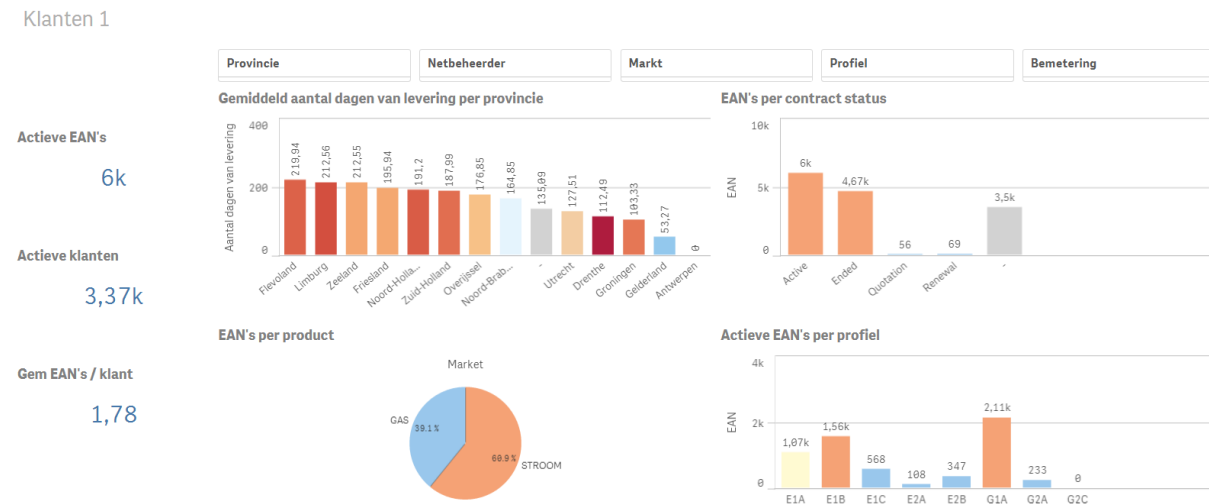
Data die tijdens dit proces is opgehaald en gebruikt wordt in berekeningen en functies kan ook opnieuw geëxporteerd worden naar een QVD-bestand. Een QVD (QlikView Data)-bestand is een bestand van Qlik zelf. Deze bestanden zijn ingedeeld volgens een logica van Qlik en kunnen ook enkel geschreven en gelezen worden door QlikView of Qlik Sense. QVD-bestanden hebben vaak een kleinere bestands grootte dan andere soorten bestanden en zijn 10 tot 100 keer sneller ingelezen in Qlik Sense dan een andere gegevensbron.

Bepaalde informatie die voor meerdere applicaties gebruikt wordt, wordt opgeladen in Qlik: hier zijn de nodige berekeningen aan toegevoegd en deze data is vervolgens geëxporteerd naar een QVD-bestand. Dit bestand wordt verder gebruikt om op te laden in de verschillende applicaties die deze informatie nodig hebben.

3.1.3 Ontwikkeling van de applicatie (frontend)

De dashboards zijn verzamelingen van verschillende soorten interactieve visualisaties die de klant kan bekijken en gebruiken. Alle visualisaties hierin moeten worden opgebouwd.

Via het drag-and-dropsysteem kan er een grafiektype naar het dashboard worden gesleept, waarna de velden die gebruikt worden, meegegeven dienen te worden. Voorbeelden van gebruikte grafieken zijn KPI's, cirkel-, staaf- en lijndiagrammen, draaitabellen en histogrammen. [Afbeelding 10]



Afbeelding 10 Voorbeeld dashboard Qlik

In deze visualisaties kan er gebruikgemaakt worden van de velden die zijn opgeladen tijdens de eerste fase van de ontwikkeling van de applicatie en ook van de functie. Het is nog steeds mogelijk om in de visualisatie zelf berekeningen te programmeren en er kan hier ook gebruik worden gemaakt van variabelen. [Afbeelding 11]



Afbeelding 11 Berekening met variabele

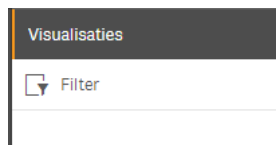
In de visualisatie kan er ook gebruikgemaakt worden van 'Drill-downs'. Dit is een veldnaam waarin meerdere veldnamen zitten. Als voorbeeld uit de 'klanten totaal'- applicatie is er een tabel waarin de naam van de klant staat, samen met de inschrijfdatum van de klant. Wanneer er een bepaalde klant geselecteerd wordt, zal er dieper ingegaan worden op deze klant en zien we de verschillende contracten die gekoppeld zijn aan deze klant. Hier is een Drill-down gebruikt met als eerste veld de klantnaam en als tweede veld het contractnummer. [Afbeelding 12]

The screenshot shows a configuration interface for a visualization. At the top left, there are two radio buttons: 'Eén' (unselected) and 'Drill-down' (selected). Below this is a 'Filteren op tabel' section with a dropdown menu set to 'Alle tabellen'. A search bar labeled 'Zoeken' is present. A list of fields is shown, including AgreementNumber, AgreementStatus, CalculatedEndAgreement, City, ClientCode, ClientName, Contractnumber, ContractStatus, and ContractType. On the right, the 'Veld:' section contains two selected fields: '=ClientName' and '=Contractnumber', each with a delete icon and a function icon 'fx'. Below this is a 'Naam:' field containing the text 'Drilldown Clientname - Contract'. A 'Beschrijving:' field is empty. There is a 'Labels:' section with an empty input field and a plus icon. At the bottom right, there is a 'Klaar' button.

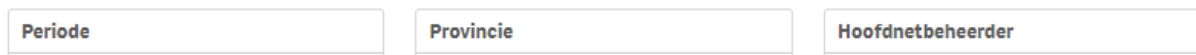
Afbeelding 12 Drilldown: klant naar contract

Wanneer er visualisaties, functies of velden zijn die meermaals gebruikt zullen worden, kunnen deze ook opgeslagen worden als een masteritem. Als een visualisatie helemaal opgevuld is met velden en functies en deze opgeslagen is als een masteritem kan deze rechtstreeks gebruikt worden in een ander dashboard.

Wanneer er wijzigingen moet gebeuren in deze visualisatie, moet er enkel in de mastervisualisatie een wijziging worden gemaakt, waarna deze in alle masteritems zal worden doorgevoerd. In het volgend voorbeeld is de filterbalk [Afbeelding 14] opgeslagen als mastervisualisatie [Afbeelding 13]. Deze komt op elk werkblad terug.



Afbeelding 13 Mastervisualisatie filter



Afbeelding 14 Filterbalk

3.1.4 Validatie van data

De data die ingeladen wordt in de applicatie kan door de berekeningen en de filters die geprogrammeerd worden verschillen van de originele, soms op een verkeerde manier. Dit kan voorvallen als er verkeerde selecties zijn gemaakt of berekeningen die foutief zijn geprogrammeerd. Tijdens het ontwikkelen van de visualisaties wordt de data gevalideerd: er wordt met de originele data vergeleken of deze correct is. Indien er opgemerkt wordt dat dit niet het geval is, moet er teruggegaan worden naar de achterliggende code en moeten de nodige wijzigingen gebeuren om de foutieve data te corrigeren.

De data die gevalideerd moet worden is gevisualiseerd in een tabel. Deze kan simpel geëxporteerd worden [Afbeelding 15] naar een Excel-bestand [Afbeelding 16], waarna de data gemakkelijk vergeleken kan worden met de originele data. Deze export is ook nuttig wanneer er data in de applicatie verstuurd moet worden naar andere personen of bedrijven.

Province	Q	Result	Q	Count(Result)	ClientCode	Q	EffectivePeriod	Q	ContractStatus	Q	AgreementStatus	Q	Avg(Result)	ContractType
Totalen				121852									320,95488	
Drenthe		3		1	2013		201611		Contract closed		Ended		3	Vast
Drenthe		3		1	2013		201612		Contract closed		Ended		3	Vast
Drenthe		3		1	2013		201701		Contract closed		Ended		3	Vast
Drenthe		3									Ended		3	Vast
Drenthe		3									Ended		3	Vast
Drenthe		3									Ended		3	Vast
Drenthe		3									Ended		3	Vast
Drenthe		3									Ended		3	Vast
Drenthe		3									Ended		3	Vast
Drenthe		3									Ended		3	Vast
Drenthe		3									Ended		3	Vast

Afbeelding 15 Tabel export in Qlik

	A	B	C	D	E	F	G	H	I
	Province	Result	ClientCode	EffectivePeriod	ContractStatus	AgreementStatu	ContractType	Count(Result)	Avg(Result)
2	Drenthe	3	2013	201611	Contract closed	Ended	Vast	1	3
3	Drenthe	3	2013	201612	Contract closed	Ended	Vast	1	3
4	Drenthe	3	2013	201701	Contract closed	Ended	Vast	1	3
5	Drenthe	3	2013	201702	Contract closed	Ended	Vast	1	3
6	Drenthe	3	2013	201703	Contract closed	Ended	Vast	1	3
7	Drenthe	3	2013	201704	Contract closed	Ended	Vast	1	3
8	Drenthe	3	2013	201705	Contract closed	Ended	Vast	1	3
9	Drenthe	3	2013	201706	Contract closed	Ended	Vast	1	3
10	Drenthe	3	2013	201707	Contract closed	Ended	Vast	1	3
11	Drenthe	3	2013	201708	Contract closed	Ended	Vast	1	3
12	Drenthe	30	2011	201611	Contract closed	Ended	Vast	2	30
13	Drenthe	30	2011	201612	Contract closed	Ended	Vast	2	30
14	Drenthe	30	2011	201701	Contract closed	Ended	Vast	2	30

Afbeelding 16 Export getoond in Excel

3.1.5 Herladen van de applicatie

Een applicatie is nooit voltooid. De klant zal bepaalde data op een andere manier willen bekijken en andere data niet meer. Wijzigingen moeten de hele tijd doorgevoerd worden in de applicatie.

Maar het belangrijkste is dat de applicatie steeds herladen moet worden, omdat er nieuwe gegevens ter beschikking komen. De klant stuurt meestal maandelijks nieuwe data door waarna deze in de applicatie moet worden opgeladen. Om deze data toe te voegen moeten de nieuwe gegevens worden samengevoegd met de oude data, omdat deze ook nog altijd zichtbaar moet zijn (incrementele data worden opgeladen).

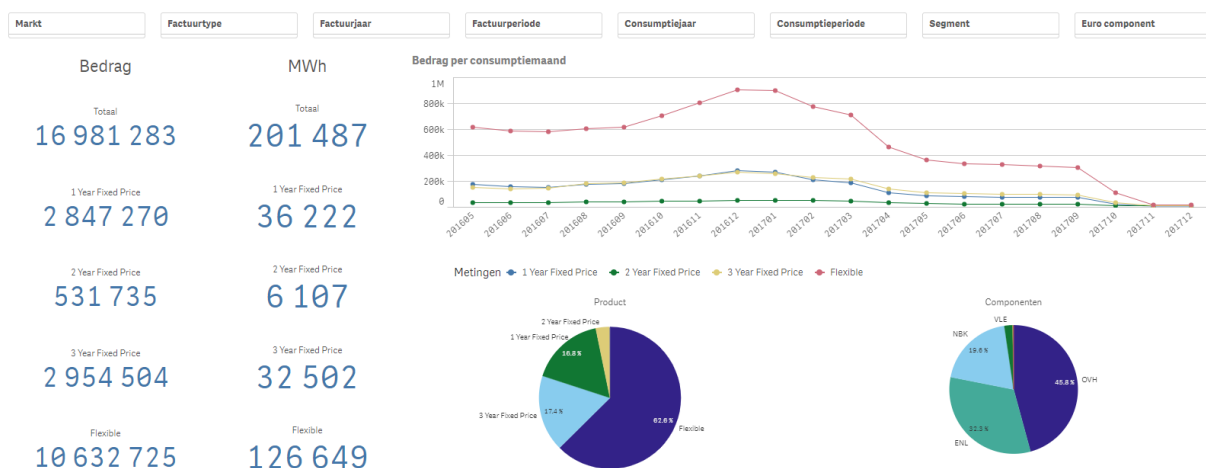
Tijdens dit proces kunnen er fouten optreden, omdat het bestand dat nieuw is aangeleverd een wijziging kan hebben in de structuur. Zo kan er een kolom van naam veranderd zijn of er kunnen nieuwe kolommen zijn bijgekomen. Hierdoor moeten er aanpassingen worden gemaakt in de achterliggende code, zodat deze bestanden en toekomstige bestanden nog steeds ingeladen kunnen worden.

3.2 Overige applicaties

Buiten de klantenapplicatie zijn er nog andere applicaties ontwikkeld: de Billed app, Billing OPS app en de Aankoop app. Deze zijn allemaal voor dezelfde klant gebouwd.

3.2.1 Billed

In de Billed- applicatie staan de klantenfacturen centraal. Hierin worden alle data van de facturen ingeladen. Alle informatie die op een energiefactuur staat ("the invoice lines"), worden in deze app geanalyseerd. In de klantenapplicatie wordt er meer globaal gekeken naar de masterdata en contracten (actief/inactief) van de klanten. In deze applicatie gebeurt een diepgaande analyse op de facturen. Deze worden op het laagste niveau van detail ingeladen en dit zowel voor de hoeveelheden als de bedragen (KWh'en en EURO's): dus niet enkel de totaalsom maar elk component of factuurlijn apart wordt ingeladen. Hierdoor wordt deze applicatie vooral gebruikt om meer inzicht te krijgen in de omzet van het bedrijf en de correctheid van de facturatie. Deze opbrengsten kunnen geanalyseerd worden per facturatiemaand of per consumptiemaand. [Afbeelding 17]



Afbeelding 17 KPI-dashboard Billed

3.2.2 Billing OPS

In deze applicatie wordt nagegaan of de aanrekeningen volledig en tijdig zijn. We vertrekken van de regels die in de energiemarkten in voege zijn met betrekking tot voorschotten en afrekeningen. We bekijken het theoretische facturatieproces, dus eigenlijk de situatie "Should Be". Wanneer wordt een voorschot of een afrekening verwacht bij bestaande klanten en bij nieuwe klanten? Dit wordt vergeleken met de lopende situatie ("As Is"). [Afbeelding 18] en [Afbeelding 19]

- A = Afrekening
 - V = Voorschot
 - E = Eindafrekening
 - AV = Afrekening verwacht maar er is voorshot
 - AE = Afrekening verwacht maar er is eindafrekening
 - EV = Eindafrekening verwacht maar er is voorshot
 - EA = Eindafrekening verwacht maar er is afrekening
 - VA = Voorschot verwacht maar er is afrekening
 - VE = Voorschot verwacht maar er is eindafrekening
 - A- = Afrekening verwacht maar er is niets gefactureerd
 - V- = Voorschot verwacht maar er is niets gefactureerd
 - E- = Eindafrekening verwacht maar er is niets gefactureerd
 - A- = Afrekening verwacht maar er is niets gefactureerd
 - V- = Voorschot verwacht maar er is niets gefactureerd
 - E- = Eindafrekening verwacht maar er is niets gefactureerd
 - A = Niets verwacht maar er is afrekening
 - V = Niets verwacht maar er is voorschot
 - E = Niets verwacht maar er is eindafrekening
- Groen = Correct, geen actie nodig
Geel = Onverwacht, er werd gefactureerd maar het type komt niet overeen met de verwachting
Rood = Actie ondernemen want er werd geen factuur opgemaakt

Afbeelding 18 Uitleg Billing OPS termen

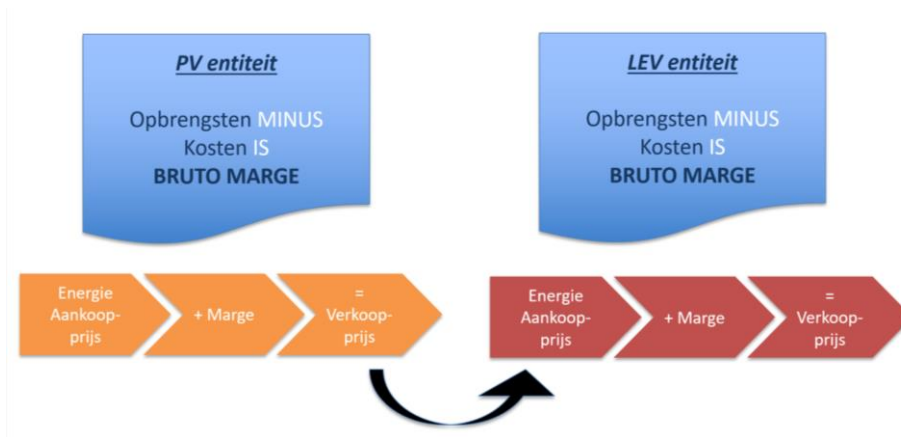
Contractnumber	InvoicePeriod					
	201611	201612	201701	201702	201703	201704
200	A	VA	VA	VA	VA	EA
200	V	V	VA	V	V	A
201	VA	VA	VA	VA	VA	A
201	VA	VA	VA	VA	VA	A
201	VA	VA	VA	VA	VA	A
201	VA	VA	VA	VA	VA	A
201	V	V	VA	V	V	A
201	VA	VA	VA	VA	VA	A
201	VA	VA	VA	VA	VA	A
201	VA	VA	VA	VA	VA	VA
200	V	V	VA	V	V	A
201	VA	VA	VA	VA	VA	A
200	VA	VA	VA	VA	VA	A
201	VA	VA	VA	VA	VA	VA
201	VE	E	E	-	-	-

Afbeelding 19 Dashboard Billing OPS

3.2.3 Aankoop

Er zijn verschillende scenario's met betrekking tot de aanrekening van de aangekochte energie in Nederland. Elk scenario volgt een ander afrekeningsproces. Het doel van deze app is om al deze scenario's in kaart te brengen. Elke wijze van afrekening zal worden gesimuleerd wat KEN (de leveringsentiteit) dient te betalen aan de PV entiteit (of energiegroothandel). Belangrijk is daarbij om een duidelijk onderscheid te maken tussen de verschillende methodes.

Afbeelding 20 illustreert het verschil tussen de PV entiteit en de leveranciersentiteit. De scope van deze app is om de verkoopprijs van de PV entiteit (dus de prijs waarop de leverancier zijn energie aankoopt).



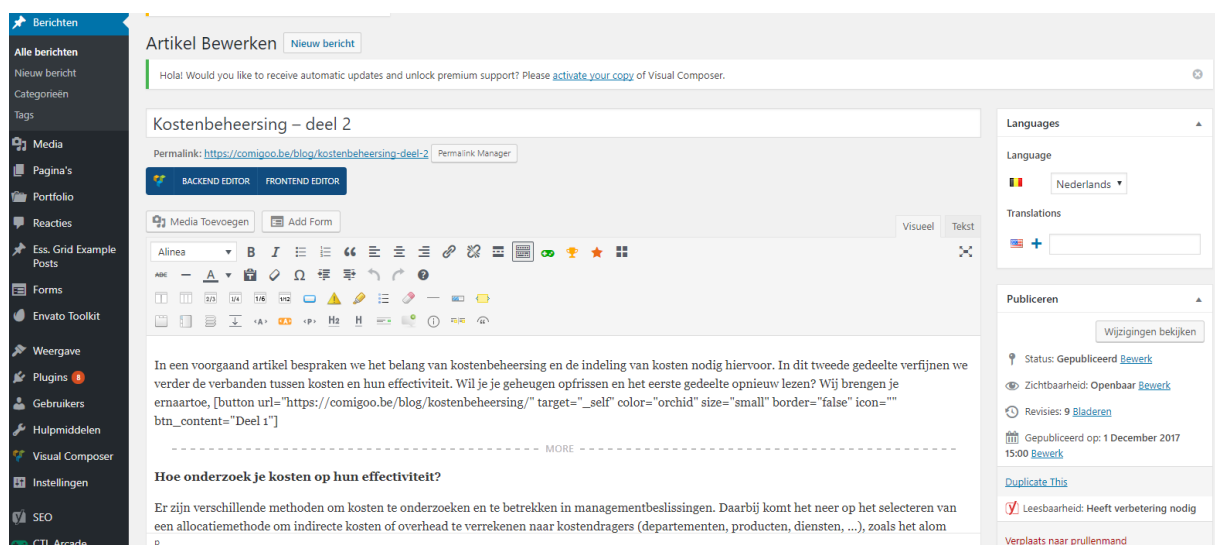
Afbeelding 20 PV en leveranciersentiteit

3.3 Opdrachten dagelijkse werking

Buiten de stageopdracht zijn er nog verschillende soorten opdrachten die uitgevoerd worden voor de dagelijkse werking van het bedrijf. Dit omvat: de website en sociale media onderhouden, nieuwe blogs aanmaken en grafisch werk ontwerpen. Deze taken moeten vooral gebeuren voor Comigoo. Comigoo is een tweede bedrijf van dezelfde aandeelhouders, dat is ontstaan door een opsplitsing van Nrgfin. De klantenmarkt voor Nrgfin was beperkt en hierdoor is er tijdens de stageperiode een afsplitsing gekomen van de IT-afdeling binnen Nrgfin. Deze afdeling is uiteindelijk Comigoo geworden. De bijkomende taken zullen kort beschreven worden.

3.3.1 Website en sociale media onderhouden.

De website en de sociale media (Facebook, LinkedIn en Twitter) van Nrgfin en Comigoo moeten worden onderhouden. Dit omvat updates doorvoeren, bepaalde teksten op de website aanpassen en nieuwe blogs schrijven. [Afbeelding 21]



Afbeelding 21 Backend website Comigoo

Deze blogs moeten ook gepubliceerd worden op de sociale media samen met interessante weetjes van het bedrijf. Een contactname via de website of sociale media, wordt zo snel mogelijk beantwoord of doorgestuurd naar de juiste medewerker. De links van de website en de sociale media's zijn te vinden in bijlage C.

3.3.2 Grafisch werk

Omdat het bedrijf aan het uitbreiden is moet er veel promotiemateriaal ontwikkeld worden zoals flyers, brochures en folders. Deze worden ontwikkeld in Adobe Photoshop (versie 17.0.0) en Illustrator (versie 17.0.1). Hierin worden ook afbeeldingen ontworpen die gebruikt kunnen worden op de website, de blogs en op sociale media. In bijlage D zijn enkele uitgewerkte voorbeelden te vinden.

4 Besluit

De stageopdracht op zich was een vrij eenduidige opdracht die bovendien zeer goed toegelicht werd door het bedrijf. Dankzij de hulp van mijn bedrijfspromotor en collega's was het veel makkelijker om de applicaties te kunnen ontwikkelen voor de Nederlandse klant KEN.

Dit was een eerste en unieke ervaring voor mij, waarbij ik de ontwikkeling van een nieuw product, of in dit geval vier producten, moest uitvoeren voor een buitenlandse klant. Door deze opdracht heb ik veel meer kennis opgedaan in verschillende technologieën: Qlik, SQL, Powershell, Perl, etc.

Niet enkel technologieën heb ik bijgeleerd maar heb ook uiteenlopende ervaringen opgedaan in de bedrijfswereld. Zo heb ik onder andere de interne organisatie en structuur leren kennen van Nrgfin, hoe binnen dit bedrijf alle processen en teammanagement gestructureerd is en hoe de processen verlopen. Ook heb ik het privilege gehad om mee te gaan naar Nederland om de applicaties zelf te tonen aan de eindklant. Hier werden de applicaties besproken, en kon ik zelf de tevredenheid van de klant ervaren. Ook werden kleine details genoteerd die de eindklant op een andere manier wenst te zien.

Daarnaast heb ik ook nog dagelijkse opdrachten moeten uitvoeren die niets bij mijn stageopdracht te maken hadden, wat ik helemaal niet erg vond. Integendeel, ik vond ik het heel fijn omdat mijn werk hierdoor varieerde. Elke dag dezelfde opdracht uitvoeren is niets voor mij. De bijkomende opdrachten lagen bovendien in mijn interessedomein (grafische ontwikkeling en webdesign).

Dat is een pluspunt dat ik bij Nrgfin fijn vind. Natuurlijk moet het dagelijks werk gebeuren maar de verschillende kwaliteiten van elk werknemer worden zo optimaal mogelijk gebruikt binnen het bedrijf. Een bijzondere reden waardoor ik, met de andere redenen, het een zeer leervolle en geslaagde stage vond bij Nrgfin.

II. Onderzoekstopic

1 Onderzoeksvraag

Tijdens dit onderzoek wordt gekeken naar de werkwijze en de databronnen die in Nederland worden gebruikt en deze worden vergeleken met de werking van het Belgisch datamodel. Hierdoor kunnen de overeenkomsten en verschillen gevisualiseerd worden en kunnen de soorten bronnen worden geanalyseerd.

Hierbij wordt specifiek nagegaan of de methodes die door Nrgfin Nederland momenteel gebruikt beter of minder efficiënt zijn als die van Nrgfin België, en wordt er gekeken of de performantie van de huidige scripts hieronder lijdt. In België wordt gebruikgemaakt van PERL scripts om de berichten te converteren naar CSV maar in Nederland wordt een ander soort bericht gebruikt. Perl is momenteel een oude programmeertaal die niet zo bekend is als Powershell en minder vaak updates heeft. Er wordt onderzocht of Powershell een alternatief zou kunnen zijn voor Nrgfin België. In punt 5. worden Powershell en PERL getest op verschillende domeinen zoals snelheid, performantie en gebruik van computergeheugen. Hieruit zal een besluit gevormd kunnen worden welke het optimaalste werkt voor de aangeleverde bestanden te converteren.

Ook moet er bekeken worden via welke scripts de Nederlandse bestanden op de beste manier naar de SQL server kunnen geïmporteerd worden.

2 Onderzoeksmethode

Een groot deel van dit onderzoek is voornamelijk literatuurstudie. Het data proces van Nederland wordt helemaal geanalyseerd maar ook dat van België om deze twee te kunnen vergelijken. De verschillende soorten types van bestanden zullen geanalyseerd moeten worden en worden gevisualiseerd in een model.

Buiten de literatuurstudie worden er ook praktische studies uitgevoerd. Er zullen testen uitgevoerd worden op verschillende soorten en groottes van bestanden met de verschillende scripts van PERL en Powershell. Bij deze studie zal de snelheid, performantie en het gebruik van computergeheugen bij beide scripttalen getest en vergeleken worden.

3 Uitvoering

3.1 België

3.1.1 Dataproces

3.1.1.1 Synthetische lastprofielen

In de energiemarkt zijn er veel verschillende processen die lopen voor de stakeholders in de markt. Ook de eindklanten worden opgesplitst in groepen op basis van hun verbruiksprofielen wat een belangrijk gegeven is in de energiemarkt. Om dit op een gestructureerde manier te laten verlopen wordt er gebruik gemaakt van SLP's oftewel synthetische lastprofielen.

Ieder lastprofiel staat voor een bepaalde categorie van verbruikers. De profielen geven een typisch profiel weer voor iedere verbruiker, per kwartier voor elektriciteit en per uur voor gas (vb professionele klanten en residentiële klanten gebruiken gas en elektriciteit op andere momenten).

In Tabel 1 bevinden zich de meest voorkomende lastprofielen, de volledige lijst van lastprofielen is opgenomen in Bijlage A. [1]

SLP	Beschrijving
S10	'RLP' Real Load Profile Electricity
S11	'SLP' Industrial < 56 kVA Electricity
S12	'SLP' Industrial >= 56 kVA Electricity
S21	'SLP' Residential night/day <1,3, also only day - Electricity
S22	'SLP' Residential night/day >= 1,3 - Electricity
S31	'SLP' < 150 000 kWh {Industrial heating} - Gas
S32	'SLP' >= 150 000 kWh {Industrial Process} - Gas
S41	'SLP' Residential Gas

Tabel 1 Verbruiksprofielen Belgische energiemarkt

In België zijn de SLP's gebaseerd op het aantal kilovoltampère (kVA) en kilowattuur (kWh). De SLP's zijn opgebouwd door middel van een 'S' gevolgd door een nummering. Via deze nummering weet de energiemarkt over welk verbruiksprofiel het gaat.

3.1.1.2 Metertypes

Er zijn verschillende soorten metertypes voor elektriciteit en maar één soort voor gas. Elke klant kan zelf bepalen welk soort elektriciteitsmeter hij of zij wilt plaatsen. Elektriciteitsmeters meten in kWh en gasmeters in kubieke meter (m³).

Enkelvoudige meter

Wanneer er maar één elektriciteitsmeter is, wordt er gesproken over een enkelvoudige meter. Het elektriciteitsverbruik wordt zowel overdag als 's nachts geregistreerd op deze meter. Klanten met deze meter hebben dan één energieprijs. [1] [Afbeelding 22]

Een handig weetje is dat een tweevoudige meter die uw dag- en nachtverbruik apart registreert voordeliger is als u vooral 's nachts en tijdens het weekend verbruikt. Eigenaars van zonnepanelen daarentegen zijn dikwijls beter af met een enkelvoudige meter. [1]

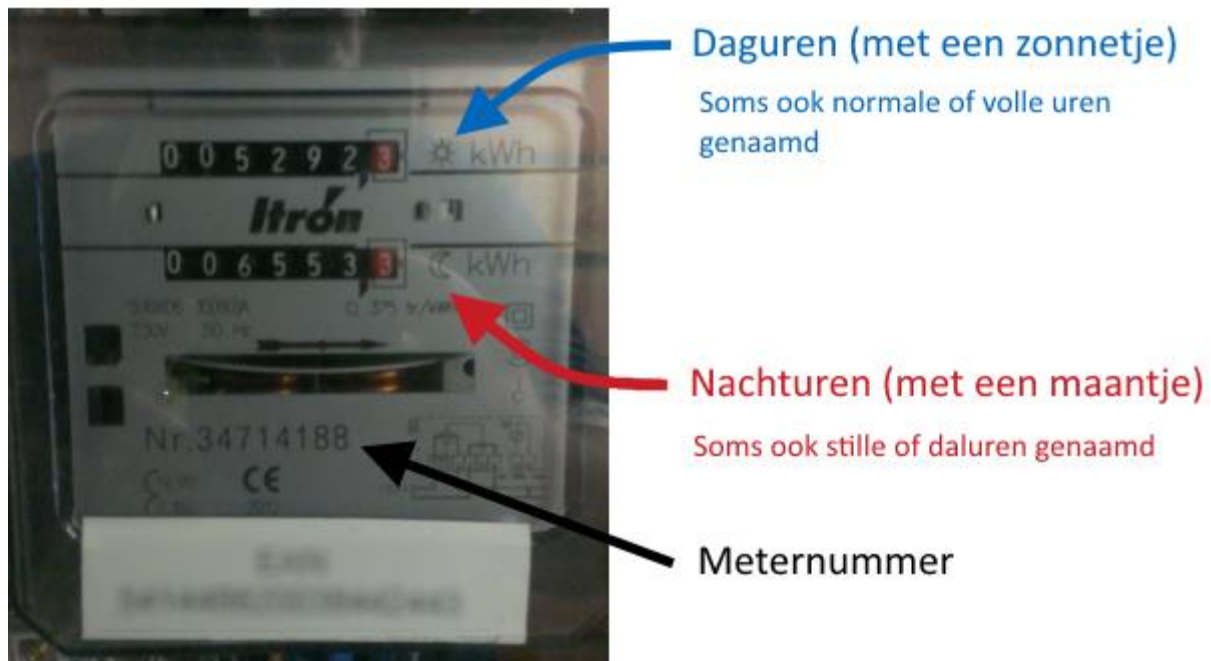


Afbeelding 22 Enkelvoudige meter

Dag- en nachtmeter (dubbele meter)

Een dubbele meter meet het elektriciteitsverbruik op een andere manier als een enkelvoudige meter. De dagmeter meet het elektriciteitsverbruik tussen 6 en 21 u. of tussen 7 en 22 u., afhankelijk van uw gemeente en netbeheerder. De nachtmeter meet tussen 21 en 6 u. of tussen 22 en 7 u. Ook het verbruik tijdens het weekend wordt op uw nachtmeter geregistreerd. Hier is één uitzondering op: het verbruik op een wettelijke feestdag die tijdens de week valt, wordt op de dagmeter geregistreerd. [1]

Deze meter [Afbeelding 23] schakelt automatisch van dag naar nacht en omgekeerd. Het verschil tussen de enkelvoudige meter en deze is dat er een aparte prijs is voor het dag- en nachttarief. Het nachtverbruik is goedkoper omdat 's nachts minder elektriciteit verbruikt wordt. [1]



Afbeelding 23 Dubbele meter

Uitsluitend nachtmeter

Deze meter telt het verbruik via een afzonderlijk elektriciteitscircuit waarop uitsluitend verwarmingstoepassingen mogen worden aangesloten (bijvoorbeeld accumulatieverwarming en warmwaterboilers).

Dit circuit is actief tijdens de nachturen en zorgt voor het opladen van de verwarmingstoepassingen. Het exacte tijdstip van aan- en uitschakelen hangt eveneens af van de gemeente en netbeheerder. Het tarief voor een uitsluitend nachtmeter is doorgaans goedkoper dan het dag- en nachttarief.

Budgetmeter

Bij een budgetmeter moet de elektriciteitsmeter opgeladen worden door vooraf te betalen. U zou het kunnen vergelijken met de prepaidkaarten van gsm's.

Een budgetmeter geeft de mogelijkheid om de energiekosten goed te beheren en uw energie rationeel te gebruiken. Er kan voor deze meter niet zelf gekozen worden. Deze meter wordt enkel geïnstalleerd bij verbruikers die schulden hebben bij hun netbeheerder.

Meter met terugdraaiende teller

Terugdraaiende meters komen enkel voor in woningen die meer elektriciteit opwekken dan verbruiken. Huizen met zonnepanelen kunnen bijvoorbeeld meer stroom opwekken dan dat ze verbruiken.

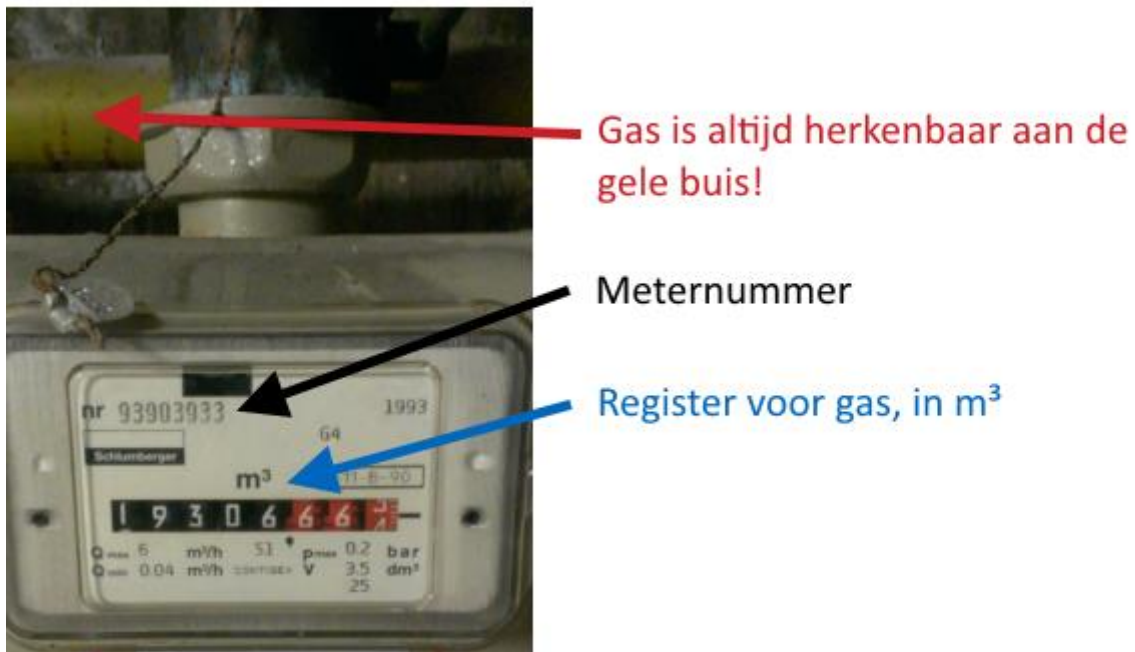
Gasmeter

Gasmeters [Afbeelding 24] zijn vaak snel te onderscheiden van elektriciteitsmeters omdat deze groter zijn en er een gele buis op aangesloten is. Deze buis dient voor de aanvoer van het gas en is altijd geel geschilderd. Net zoals de enkelvoudige elektriciteitsmeter is er voor gas maar één tarief. Meestal wordt dit op de energiefactuur omgezet naar kWh via een bepaalde formule op maat van de situatie.

De berekening om m^3 naar kWh te vormen is het aantal m^3 te vermenigvuldigen met de omzettingcoëfficiënt. Het omzettingcoëfficiënt hangt af van het type gas dat aangeleverd wordt en verandert elke maand. Er zijn twee gastypes: laagcalorisch en hoogcalorisch.

Wie laagcalorisch gas heeft, verbruikt voor eenzelfde verwarming in volume (m^3) meer gas dan iemand die hoogcalorisch gas heeft maar er wordt een lagere omrekeningscoëfficiënt (van m^3 naar kWh) gebruikt. [1] Het omgekeerde geldt dan voor hoogcalorisch gas.

Ter verduidelijking met een voorbeeld. Een klant die een gasverbruik van $1000 m^3$ heeft en een laagcalorisch gastype (omzettingcoëfficiënt = 10,22) heeft als berekening: $1000 \times 10,22 = 10\ 220$ kWh. [1]



Afbeelding 24 Gasmeter

3.1.2 Datasources

De data wordt voornamelijk op twee manieren aangeleverd binnen Nrgfin België: via SFTP en Haugazel. Het bestandstype dat aangeleverd wordt, varieert van energieleverancier tot energieleverancier. De bestandstypes die het meest gebruikt worden zijn: EDIEL, CSV, Excel en TXT.

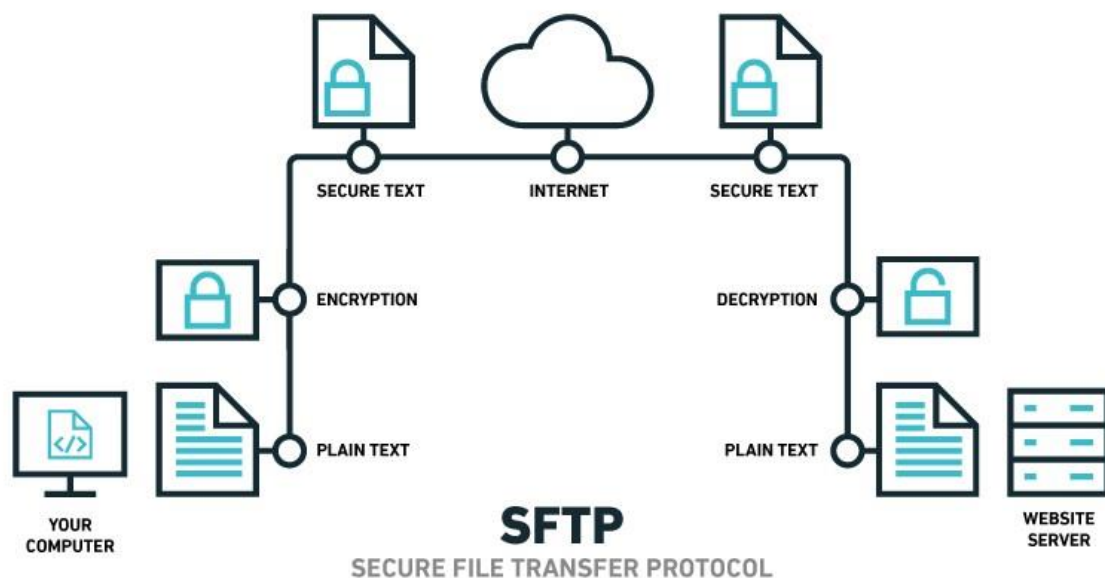
3.1.2.1 SFTP

De data van de energieleveranciers kan via SFTP geleverd worden. De SFTP, oftewel 'Secure File Transfer Protocol', maakt een beveiligde verbinding tussen twee digitale plaatsen waarin bestanden uitgewisseld kunnen worden.

SFTP biedt bovendien functionaliteit voor extern bestandsysteembeheer aan. Hierdoor kunnen onderbroken bestandsoverdrachten hervat worden, de inhoud van externe mappen kan worden weergegeven en externe bestanden kunnen worden verwijderen.

SFTP wordt voornamelijk gebruikt samen met een SFTP Client zoals FileZilla of Cyberduck waarin je enkel het IP-adres en de inloggegevens moet invullen en dan de hele mappenstructuur kan zien en uitwisselen. Dit kan natuurlijk ook helemaal automatisch ingesteld worden.

Zoals in Afbeelding 25 wordt getoond staat het bestand dat verzonden wordt in gewoon tekstformaat. Door middel van een encryptie van SFTP zal de tekst omgevormd worden tot een beveiligde tekst die niet meer leesbaar is zonder de sleutel. Deze beveiligde tekst wordt dan via het internet verzonden naar de server waar de ontvanger het bestand kan afhalen. Deze heeft het wachtwoord om de beveiligde tekst te ontcijferen waardoor het bestand weer een gewoon tekstformaat wordt en leesbaar is op de ontvanger zijn of haar computer.



Afbeelding 25 SFTP-schema

3.1.2.2 Haugazel

Haugazel is een platform dat is opgericht door Haulogy, een IT-bedrijf dat is gespecialiseerd in IT-oplossingen voor de energiemarkt. Via Haugazel kunnen energiebedrijven drie grote toepassingen uitvoeren:

- Energiegerelateerde berichten bijhouden
- Statussen van processen bekijken
- Berichten en documenten op een eenvoudige manier versturen en ontvangen.

In deze situatie wordt er vooral gekeken naar de laatste functionaliteit, het versturen en ontvangen van de energiegerelateerde berichten. Het principe van Haugazel is hetzelfde als dat van SFTP; het enige verschil hierbij is dat er met Haugazel enkel berichten verstuurd kunnen worden die te maken hebben met energie.

Hierbij staat er dus een limiet van extensies die verzonden kunnen worden met Haugazel. Bij de klanten van Nrgfin is dit zelfs enkel EDIEL-berichten. Een powerpoint die een extensie '.pptx' heeft zal via Haugazel niet verzonden kunnen worden.

3.1.2.3 EDIEL

EDIEL-berichten zijn een meer gespecificeerde vorm van EDI-berichten. EDI (Electronic data Interchange) is een standaard die is opgesteld voor het elektronisch uitwisselen van bedrijfsdocumenten. Het is een soort template die opgesteld is en waaraan de bedrijven die dit standaard gebruiken zich moeten houden.

EDIEL-berichten (Electronic data Interchange in the Electricity Industrie) [2] is een standaard die is ontworpen specifiek om te gebruiken in de energiemarkt en dit voor heel België. Binnen EDIEL zijn er dan nog eens drie opsplitsingen die gebruikt worden, namelijk:

- UTILMD
- UTILTS
- INVOIC

UTILMD zal gebruikt worden voor het uitwisselen van relatief stabiele gegevens tussen verschillende partijen binnen de energiemarkt. Het gebruik van de UTILMD zal gerelateerd zijn aan de leverancierswissel met uitwisseling van Master Data, die betrekking heeft op een toegangspunt of een installatie. [1]

Voor transactionele uitwisseling van gegevens zal het UTILTS -bericht gebruikt worden. Een "Time Series" is een opeenvolging van controles van één enkel proces, meestal genomen met gelijke tussenpozen. Deze "time series" omvatten de gemeten waarden, voorspellingen, ramingen, enz. Aan elke "time series" zal er technische en administratieve informatie gekoppeld zijn, zoals meterkarakteristieken enz. [4] [2]

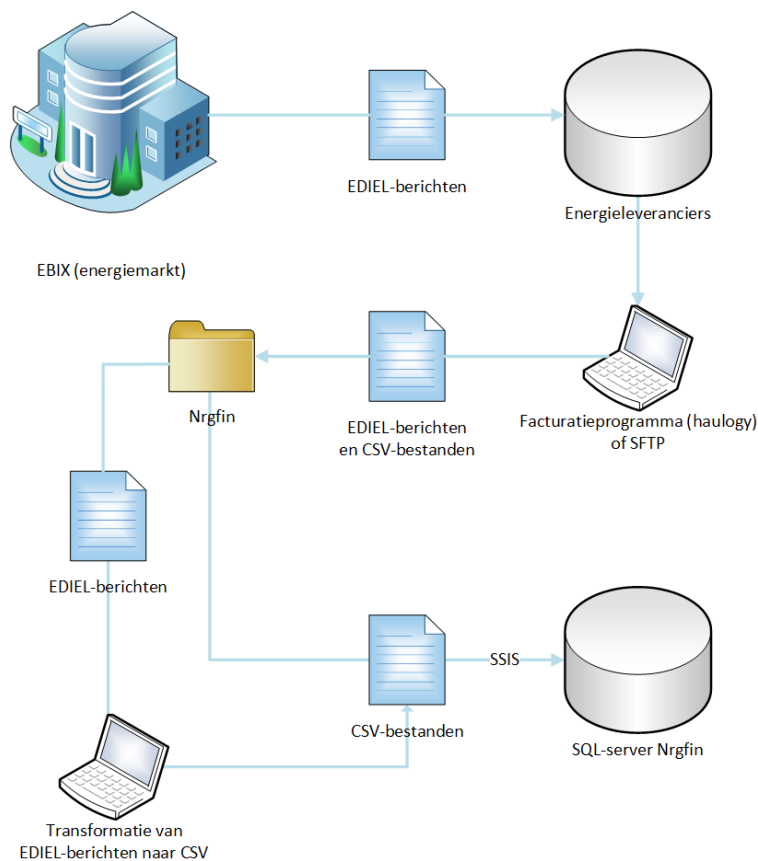
Het INVOIC- bericht wordt gebruikt voor het communiceren van de details op de facturen met betrekking tot het gebruik van het distributienetwerk." [1]

3.1.3 IT-infrastructuur

3.1.3.1 Ontvangst klantenbestanden

In Figuur 1 is het proces weergegeven hoe de bestanden van de klanten terechtkomen op de SQL-server van Nrgfin. Nrgfin ontvangt deze bestanden via een SFTP-server of een gespecificeerd facturatieprogramma zoals Haugazel. Afhankelijk van het bestandstype kunnen de bestanden rechtstreeks ingeladen worden in de SQL-server of moet er eerst een transformatie gebeuren van het bestandstype.

Bestanden met een EDI-extensie kunnen niet rechtstreeks ingeladen worden op de SQL-server van Nrgfin en moeten hiervoor getransformeerd worden naar CSV-bestanden.



Figuur 1 Infrastructuur voor ontvangen bestanden

3.1.3.2 ETL-proces

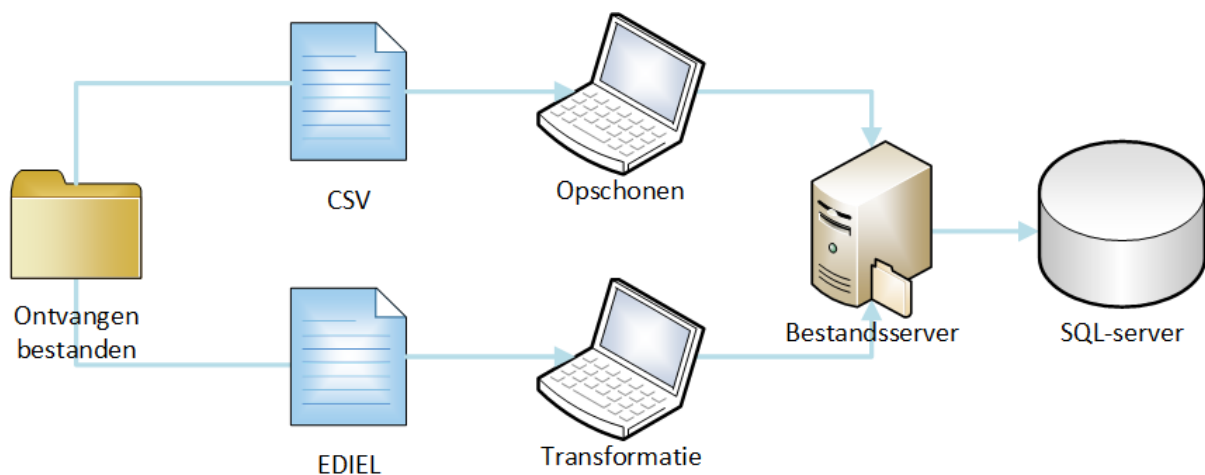
De ontvangen bestanden stromen door één of verschillende ETL-processen (Extract, Transform, Load) om te gebruiken in de applicaties. Met een ETL-proces worden de gegevens uit de bestanden opgehaald, deze worden getransformeerd naar het gewenste formaat en kan dan geladen worden in de gewenste applicaties.

Bij Nrgfin worden er voornamelijk twee processen gebruikt: één voor de bestanden op de SQL-server te plaatsen en één voor de gegevens te converteren naar een QVD.

3.1.3.3 SQL-server

CSV-bestanden die opgeladen moeten worden in de SQL-server moeten een bepaalde structuur en dataformaat hebben, is dit niet het geval zal er een foutmelding komen van SQL en wordt het bestand niet opgeladen. Hiervoor gaan de CSV-bestanden door een cleaningprocess en worden de bestanden opgeschoond. Hierna zijn de CSV's gereed voor naar SQL te laden. [Figuur 2]

EDIEL-bestanden moeten getransformeerd worden naar CSV's. Dit gebeurt door scripts die geschreven zijn in Perl. Deze scripts gaan de EDIEL-berichten automatisch opschonen en transformeren naar CSV's.

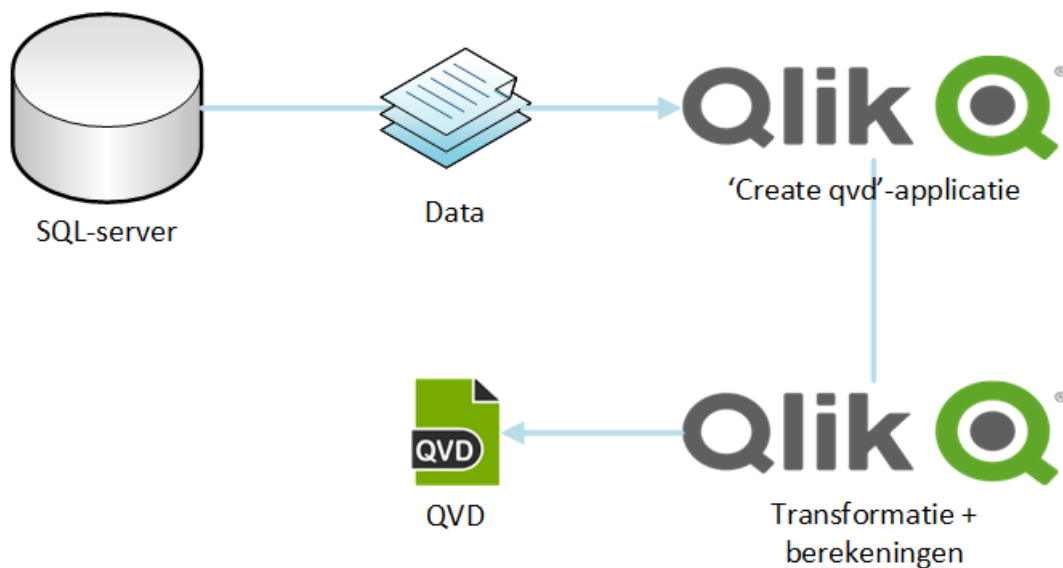


Figuur 2 ETL-proces voor SQL-server

3.1.3.4 QVD

Voor QVD-bestanden (QlikView Data) is er een apart ETL-proces. Er wordt gebruik gemaakt van dit bestandstype zodat de algemene berekeningen die moeten gebeuren telkens niet opnieuw moeten worden uitgevoerd. Deze bestanden worden ook 10 tot 100 keer sneller ingelezen in Qlik dan andere bestandstypes.

Meestal heeft elke klant een eigen 'Create QVD'-applicatie die gebruikt wordt om het ETL-proces uit te voeren. De data die nodig is wordt opgehaald uit de SQL-server, de benodigde transformaties en berekeningen worden uitgevoerd in de applicatie en op het einde wordt deze data geëxporteerd naar een QVD-bestand. Deze QVD-bestanden worden gebruikt in de verschillende applicaties van de klant.



Figuur 3 ETL-proces voor QVD-bestanden

3.2 Nederland

3.2.1 Dataproces

Bij de Nederlandse energiemarkt zijn er, net zoals bij de Belgische energiemarkt, veel verschillende processen ten behoeve van de verschillende stakeholders. Wat in België als SLP's wordt genoemd zijn in Nederland Profielcategorieën.

3.2.1.1 Verbruikprofielen

In Nederland zijn de verbruiksprofielen gebaseerd op ampère. Deze profielen beginnen ook met de letter voor welke levering ze staan. Als er elektriciteit wordt geleverd is dit een 'E', bij gas een 'G'. Dit wordt nog eens opgevolgd door een cijfer en een letter die het type profiel weergeven: dit kan een particulier of een bedrijf zijn met een enkele of dubbele meter. Alle profielen die in Nederland gebruikt worden zijn te vinden in Tabel 2. [5]

Elektriciteit		Gas	
SLP	Beschrijving	SLP	Beschrijving
E1A	aansluiting met een meter die beschikt over maar 1 actief telwerk of onbemeten aansluiting (met uitzondering van gedimensioneerde profielen)	G1A	Kleinverbruikers zonder meetinrichting. Kleinverbruikers met een standaard jaarverbruik < 5000 m3 en met een gasmeter ≤ G6
E1B	aansluiting die beschikt over een meter met 2 actieve telwerken. Het eerste telwerk meet op werkdagen van 7:00 uur tot 23:00 uur. Het 2de telwerk is actief op de overige tijden (nachttarief schakeling).	G2A	Kleinverbruikers die niet voldoen aan de criteria voor profielcategorie G1A.
E1C	aansluiting die beschikt over een meter met 2 actieve telwerken. Het eerste telwerk meet op werkdagen van 7:00 uur tot 21:00 uur. Het 2de telwerk is actief op de overige tijden (avondtarief schakeling).	G2C	Grootverbruikers met een afname < 170.000 m3 per jaar die niet zijn voorzien van uurmeting en dagelijkse of uurlijkse uitlezing.
E2A	enkeltarief		
E2B	aansluiting die beschikt over een meetinrichting die het nacht- of avondtarief meet		
E3A	Bedrijfstijd ≤ 2000 uur		
E3B	Bedrijfstijd > 2000 uur, Bedrijfstijd ≤ 3000 uur		
E3C	Bedrijfstijd > 3000 uur, Bedrijfstijd < 5000 uur		
E3D	Bedrijfstijd ≥ 5000 uur		
E4A	alle gemeten aansluitingen geschakeld op het stuursignaal openbare verlichting met een aangesloten vermogen van minder dan 100 kW		

Tabel 2 Verbruiksprofielen Nederlandse energiemarkt

3.2.1.2 Metertypes

Nederland heeft dezelfde metertypes als België, alleen heeft de dubbele meter in Nederland twee versies: analoog en digitaal. Analoog is de standaardversie en de digitale versie is de nieuwe versie, ook wel de slimme meter genoemd. Via deze meter kan de netbeheerder rechtstreeks aan de energieverbruiker en deze doorsturen naar de energieleverancier. De klant kan ook via een app zijn verbruiken controleren en zo energiebewust worden.

3.2.2 Datasources

Bij de klanten van Nrgfin Nederland wordt gebruikgemaakt van EDSN en EABO. Via deze twee softwareoplossingen verzamelt de leverancier (of leveringsentiteit) zijn documenten en bewerkt ze indien nodig. Deze kunnen hierna via SFTP verzonden worden naar externe bedrijven zoals Nrgfin.

3.2.2.1 EDSN

EDSN, oftewel Energie Data Services Nederland, is een bedrijf dat samenwerkt met de Nederlandse energieverdelers om een goede uitwisseling van data te hebben in de energiesector. EDSN heeft een IT-infrastructuur ontwikkeld voor een snelle en efficiënte uitwisseling van marktgegevens en documenten tussen de energieproducenten, energieleveranciers en eventueel externe bedrijven die bepaalde documenten ook moeten gebruiken.

EDSN is enkel een infrastructuur; voor het werkelijk versturen van de bestanden is aparte software nodig, zoals SFTP en Filezilla.

3.2.2.2 EABO

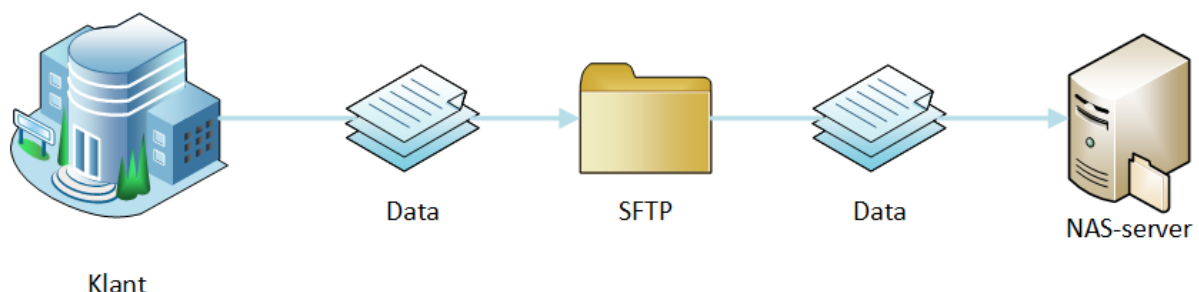
EABO is een bedrijf dat zorgt voor een efficiënt administratief beleid door administratieve processen op te volgen, te optimaliseren en te reduceren. Dat omvat onder andere het ontvangen en lezen van marktdata, het factureren van eindklanten en het opvolgen van slechte betalers. De filosofie van dit bedrijf is: "Alle werkzaamheden die meer dan één keer uitgevoerd worden, moeten geleidelijk zo veel mogelijk worden geautomatiseerd binnen één Europees administratief platform, ontwikkeld voor en door intensieve samenwerking met verschillende energieleveranciers." [3]

3.2.3 IT-infrastructuur

3.2.3.1 Ontvangst klantenbestanden

Nrgfin heeft momenteel één Nederlandse klant waarvan de bestanden ontvangen worden via een SFTP-server. Deze bestanden worden opgeslagen op een lokale NAS-server als back-up en de bestanden die nodig zijn voor de applicaties worden, nadat deze door het ETL-proces zijn gestroomd, op een serveropslagplaats gekopieerd. De meest voorkomende bestandstypes zijn CSV, XLSX, TXT en XML-bestanden, deze soorten worden in een komende sectie verder uitgelegd. [Figuur 4]

Momenteel wordt de Nederlandse data bewaard op een NAS-server en wordt deze niet opgeladen in de SQL-server. De gegevens worden rechtstreeks in Qlik geladen.



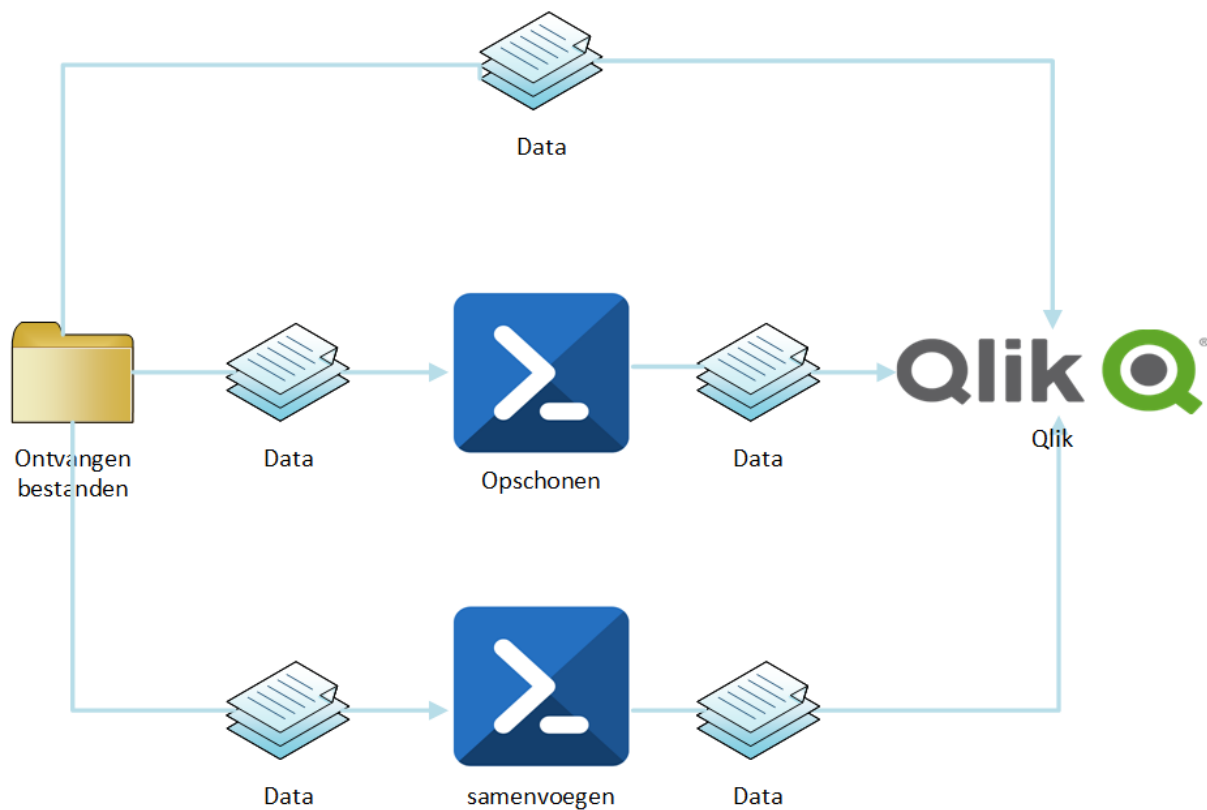
Figuur 4 Ontvangst Nederlandse data

3.2.3.2 ETL-processen

XML, CSV, XLSX en TXT

De ontvangen bestanden zitten bij ontvangst vaak allemaal bij elkaar in één grote map, het eerste proces dat hierop wordt uitgevoerd is elk soort bericht te verplaatsen in een eigen foldermap. Een volgende stap eventueel is om de bestanden ook nog te splitsen per jaar en maand. De tweede stap is afhankelijk van het type bericht en de applicatie waarin deze gebruikt zal worden.

- De bestanden kunnen rechtstreeks gebruikt worden in Qlik, er moet geen verdere uitvoering op volgen.
- De structuur binnen het bestand is niet bruikbaar en moet opgeschoond worden.
- Verschillende soorten bestanden kunnen per dag verschillende versies hebben, waardoor sommige berichttypes oplopen tot 10.000 bestanden voor één maand. Dit kan moeilijk opgeladen worden in Qlik, hierom wordt er een Powershell-script gebruikt om de bestanden met hetzelfde berichttype samen te voegen tot één groot bestand. Dit bestand, ook wel masterfile genoemd, is veel gemakkelijker en sneller om op te laden in Qlik. [Figuur 5]



Figuur 5 ETL-proces Nederlandse bestanden

3.3 Datatypes

3.3.1 XML

Het grootste deel van de bestanden die in Nederland gebruikt worden zijn XML bestanden. XML staat voor eXtensible Markup Language. XML is vooral ontstaan voor het bewaren en het transporteren van data. Het kan verschillende soorten gegevens eenvoudig en makkelijk versturen: dit kan gaan van financiële data tot een geschreven tekst van een eindwerk bijvoorbeeld.

Hier volgt een voorbeeld van hoe de data in een XML- bestand wordt opgeslagen:

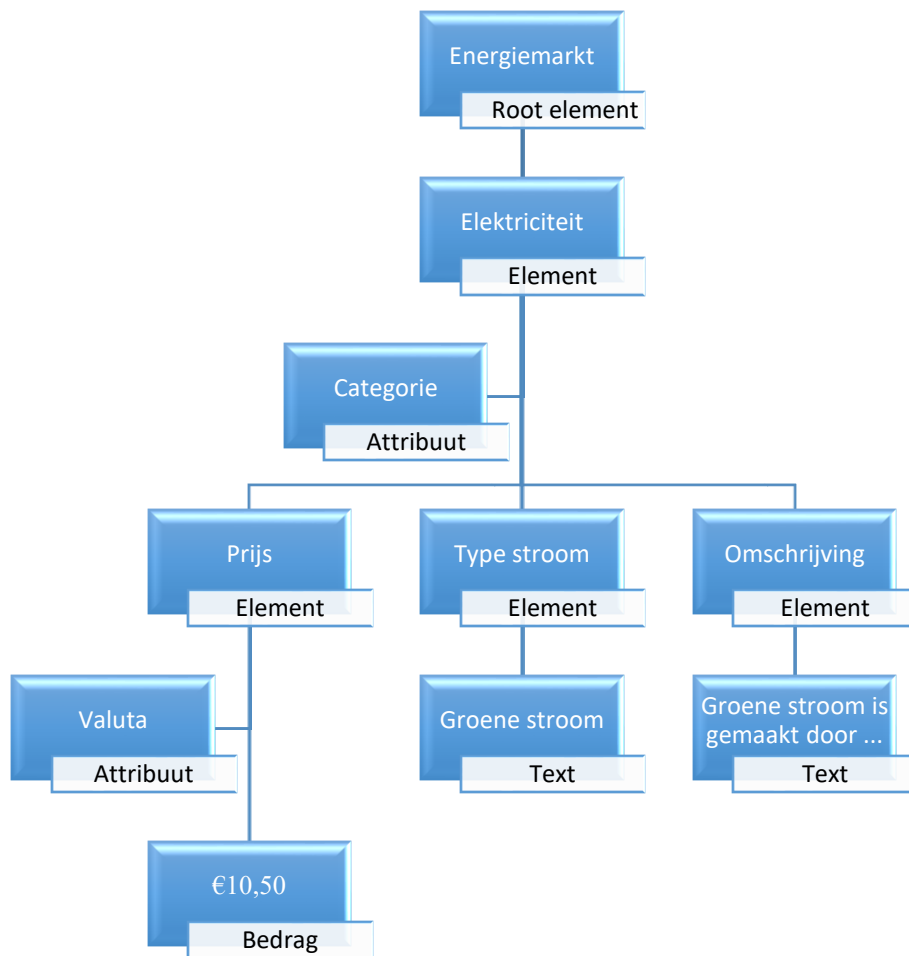
```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Vicky</to>
  <from>Thomas</from>
  <heading>Voorbeeld</heading>
  <body>Dit is een voorbeeld van de opmaak van een XML.</body>
</note>
```

Afbeelding 26 Voorbeeld interne structuur XML

3.3.1.1 XML- structuur

Een XML -bestand is opgemaakt in een 'boomstructuur' die begint met de 'root element' en zo verder wordt opgedeeld in verschillende 'child elements'. Dit wil dus zeggen dat er een hoofdelement is en dat deze onderverdeeld wordt in één of meerdere elementen. Deze elementen kunnen dan nog eens onderverdeeld worden in nog andere elementen. Ieder element kan ook een attribuut bevatten dat meer info geeft over het element.

Als voorbeeld nemen we de energiemarkt waar elektriciteit en gas beschikbaar is. In ons voorbeeld [Figuur 6] is de energiemarkt ons 'root element'. Dit is het basiselement dat alle andere element zal omsluiten. Onder dit element kan er dan bijvoorbeeld het element elektriciteit staan. Als attribuut kan dit element dan bijvoorbeeld 'categorie' hebben. Een ander element dat onder categorie kan staan is dan bijvoorbeeld gas. Onder het element elektriciteit kan je dan nog bijvoorbeeld de elementen: type elektriciteit, omschrijving en prijs hebben. De prijs kan dan als attribuut 'valuta' of 'muntsoort' hebben. Hier wordt dan euro ingevuld voor Europa of dollar voor Amerika. In onderstaande afbeeldingen is er een visualisatie gecreëerd ter illustratie van de boomstructuur over dit voorbeeld.



Figuur 6 Boomstructuur XML

3.3.1.2 XML syntax

De syntax die gebruikt wordt in een XML bestand is vrij eenvoudig en logisch. Elk bestand moet een basiselement bezitten. Daarnaast kan een XML een prolog bevatten, maar dit is optioneel. In een prolog staat extra informatie over het XML bestand zelf. Hierin kan bijvoorbeeld het versienummer en de versleuteling inzitten.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Afbeelding 27 Versleutelingsregel

Wanneer hier gebruik van wordt gemaakt, moet deze regel wel als eerste [4] [7] in het bestand komen. Om onnodige fouten te voorkomen is het aangeraden om als versleuteling standaard UTF-8 te nemen. Dit is de standaardversleuteling die voor XML bestanden wordt gebruikt. Dit is ook de standaardversleuteling voor andere programmeertalen zoals HTML, CSS, Javascript en SQL.

Een andere regel [4] [7] binnen XML is dat ieder element een begin en een eind label of tag bezit. Ook zijn deze tags hoofdlettergevoelig. Dus het begin en eind label moet met dezelfde hoofdletters en kleine letters geschreven worden. Het is dan ook normaal dat deze tags op een juiste manier genest worden. Dit wil dus zeggen dat de laatst geopende tag als eerst terug moet sluiten.

Een leuk weetje is dat in XML berichten de witruimtes behouden worden en dat deze niet afgekapt worden tot één witruimte zoals in meerdere andere programmeertalen.

3.3.2 EDI

De afkorting EDI staat voor Electronic Data Interchange. Het wordt voornamelijk gebruikt als computer-tot-computer uitwisselingsbron voor bedrijfsdocumenten, deze staat in een standaard elektronisch formaat afgesproken tussen de twee zakenpartners.

Met computer-tot-computer wordt bedoeld dat het bericht van een computer direct naar een andere computer wordt verstuurd zonder de tussenkomst van mensen. Dus EDI vervangt eigenlijk het mailen en faxen. Doordat de tussenkomst van de mensen bij EDI niet aanwezig is gaat het proces veel sneller en is de kans op fouten ook een heel stuk lager. [2]

De inhoud van een EDI-bericht is bij elk type dat wordt besloten tussen de zakenpartners anders. De inhoud van een standaard EDI-bericht dat de energiemarkt gebruikt is als volgt:

```
UNJ+ FAC5+FACTN:543926841'  
FACTD+101:25102017:49'  
Bedrijf+347 ENI  
Laanstraat+1000+Brussel'  
BED+28:35+31:49:ELK+GAS'  
KLAN+2:10325'  
CON+101:ultra'  
UNK+31+FAC5'
```

Tabel 3 Voorbeeld EDI-bericht energiemarkt

In de energiemarkt specifiek wordt er gesproken over EDIEL-berichten. Hier staat EDIEL voor “Electronic Data Interchange in the Electricity industry”. De gehele energiemarkt in België hanteert dit als standaard.

3.3.2.1 EDI-structuur

Een EDI-bestand bestaat uit drie componenten: data elementen, segmenten en enveloppen. Deze drie componenten zijn geformatteerd op basis van de regels van het EDI-standaard dat de zakenpartners intern hebben besloten. Onderstaande tabel zal ik gebruiken als voorbeeld om de verschillende componenten beter uit te leggen. De tabel geeft een energiefactuur weer voor de maand oktober.

Gelieve het volgende verschuldigd bedrag te betalen op BE01 2345 6789 1011.		
Factuurnummer 54392684166		Factuurdatum 25/10/2017
Omschrijving	Periode	Bedrag
Elektriciteit	01/10/2017 – 31/10/2017	€ 28,35
Gas	01/10/2017 – 31/10/2017	€ 31,49
BTW		€ 12,57
Totaal factuurbedrag		€ 72,41

Uw persoonlijke gegevens		Klantendienst
Klantnummer	103252458	www.test.be
Leveringsadres	Kilowattstraat 10 3500 Hasselt	info@test.be 078/111111
Huidig contract	Elektriciteit ultra	Laanstraat 1 1000 Brussel

Tabel 4 Voorbeeldfactuur energie

3.3.2.2 Data elementen

Een data element in een EDI-bericht is elk afzonderlijk informatief item dat in het bestand te vinden is. Ieder data element krijgt ook een beschrijving mee zodat men bijvoorbeeld weet welk type het item is of hoe groot dit is. Het type kan numeriek, alfanumeriek, datum of tijd zijn bijvoorbeeld waarbij de lengte niet groter kan zijn dan 20 karakters. Deze data elementen worden gegroepeerd in segmenten. In ons voorbeeld zijn dus de klantnummer, het adres, de woonplaats, het type contract allemaal verschillende data elementen. Ieder data element heeft in persoonlijke gegevens een andere kleur meegekregen [2].

Gelieve het volgende verschuldigd bedrag te betalen op BE01 2345 6789 1011.		
Factuurnummer 54392684166		Factuurdatum 25/10/2017
Omschrijving	Periode	Bedrag
Elektriciteit	01/10/2017 – 31/10/2017	€ 28,35
Gas	01/10/2017 – 31/10/2017	€ 31,49
BTW		€ 12,57
Totaal factuurbedrag		€ 72,41

Uw persoonlijke gegevens		Klantendienst
Klantnummer	103252458	www.elactrabel.be
Leveringsadres	Kilowattstraat 10 3500 Hasselt	info@eni.be 078/111111
Huidig contract	Elektriciteit ultra	Laanstraat 1 1000 Brussel

Tabel 5 Voorbeeldfactuur met data elementen

3.3.2.3 Segmenten

Een segment is een groepering van data elementen. Data elementen kunnen altijd gegroepeerd worden in bepaalde categorieën die gemeenschappelijke eigenschappen of kenmerken hebben. Elke groep of segment zoals bij EDI-berichten gebruikt wordt verteld een bepaald soort informatie.

In onze voorbeeldfactuur zijn er zo dus data elementen die wij in segmenten kunnen plaatsen:

Persoonlijke gegevens
Klantendienst
Factuurdetails
Factuurgegevens

Gelieve het volgende verschuldigd bedrag te betalen op BE01 2345 6789 1011.		
Factuurnummer 54392684166		Factuurdatum 25/10/2017
Omschrijving	Periode	Bedrag
Elektriciteit	01/10/2017 – 31/10/2017	€ 28,35
Gas	01/10/2017 – 31/10/2017	€ 31,49
BTW		€ 12,57
Totaal factuurbedrag		€ 72,41

Uw persoonlijke gegevens		Klantendienst
Klantnummer	103252458	www.elactrabel.be
Leveringsadres	Kilowattstraat 10	info@eni.be
	3500 Hasselt	078/111111
Huidig contract	Elektriciteit ultra	Laanstraat 1 1000 Brussel

Tabel 6 Voorbeeldfactuur met segmenten

3.3.2.4 Enveloppen

EDI-bestanden maken gebruik van drie enveloppen waar de data in terecht komt. De drie enveloppen zijn de berichtenvelop, groepenvelop en de uitwisselingvelop. In de berichtenvelop komt de data van één bericht, bij ons voorbeeld dus een energiefactuur. In de groep envelop staan dan bijvoorbeeld alle energiefacturen. En als deze en nog andere groep enveloppen moeten verstuurd worden naar een ontvanger worden deze allemaal in de uitwisselingsvelop gestoken.

3.3.2.5 EDI-proces

Het proces om een EDI-bestand te versturen bestaat uit drie stappen:

- Het document klaarmaken voor te verzenden
- Het document vertalen naar EDI-formaat
- Een connectie leggen met uw zakenpartner en het EDI-bericht verzenden

Document klaarmaken voor verzending

Eerst moet er gekeken worden welke documenten allemaal verzonden moeten worden naar de ontvanger. Deze moeten allemaal verzameld en georganiseerd worden. Deze documenten kunnen op verschillende manieren en verschillende middelen opgebouwd worden. De data in deze documenten kunnen bijvoorbeeld opgebouwd zijn door:

- Mensen die gegevens invullen via een computer
- Elektronische bestanden geformatteerd in databestanden
- Een database exporteren
- Software dat geïntegreerde modules heeft voor EDI-bestanden

Document vertalen naar EDI-formaat

Wanneer de documenten allemaal verzameld zijn is het tijd om deze te vertalen naar EDI-formaat. De documenten gaan door een specifieke software die de inhoudelijke data converteert in een EDI-formaat. Hierbij worden dan de juiste segmenten en data elementen aangemaakt.

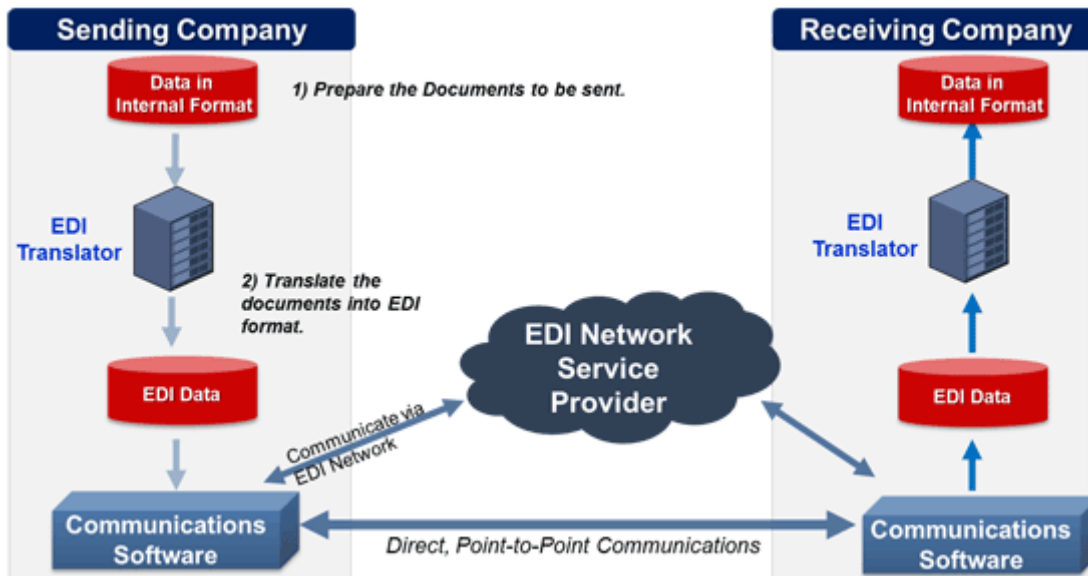
De software is op twee manieren beschikbaar: oftewel koop je een volledig vertaalsoftware die werkt met jouw gewenste EDI-formaat. Er zijn verschillende soorten software op de markt, met vaak heel uiteenlopende prijzen.

Een andere manier is om een service leverancier te gebruiken. Wanneer je voor deze optie kiest, moet je je data verzenden naar de leverancier, deze behandelt dan de vertaling van jouw documenten naar EDI-formaat. Wanneer alles klaar is, stuurt de leverancier de EDI-berichten terug.
[2]

Connectie leggen naar zakenpartner en EDI versturen

Als alle benodigde documenten vertaald zijn naar een EDI-formaat is het tijd om deze te verzenden naar de ontvanger. Er zijn verschillende manieren om de EDI-berichten te versturen.

De meest gebruikte manier is om de bestanden te versturen over het internet met een beveiligd internetprotocol zoals SFTP (Secure File Transfer Protocol). Een andere manier die vaak gebruikt wordt is door middel van een EDI-netwerkaanbieder als tussenpersoon te gebruiken. [Afbeelding 28] Deze persoon zal dan een beveiligde connectie leggen tussen de verzender en de ontvanger van de bestanden. [2]



Afbeelding 28 Model over verzending EDI

3.3.3 CSV

Een vrij bekend type van bestand onder alle computergebruikers. Een CSV of comma-separated values bestand is eigenlijk een Excel bestand maar waarin alle gegevens van een rij achter elkaar worden gestoken en worden gescheiden door een komma (,). Standaard worden de gegevens in een rij opgedeeld door een komma maar dit kan evengoed een punt(.), puntkomma (;) of zelfs een slash (/) of backslash (\) zijn.

Ter verduidelijk is hieronder een voorbeeld opgemaakt:

Datagegevens				CSV
Naam	Leeftijd	Prijs	Datum	<u>Naam, Leeftijd, Prijs, Datum</u>
Thomas	22	35.34	24/11/2017	Thomas,22,35.34,24/11/2017
Bob	34	103.49	10/02/2016	Bob,34,103.49/10/02/2016
Ann	25	56.97	19/08/2016	Ann,25,56.97,19/08/2016

Tabel 7 Voorbeeld CSV-bestand

3.3.4 TXT

Tekstbestanden zijn eenvoudige bestanden die meestal bestaan uit platte tekst. Dit wil zeggen dat er enkel letters, spaties, cijfers en leestekens in voorkomen. Er zijn dus geen speciale tekens of opmaak aanwezig. Dit is ook een type bestand dat vrij bekend is onder de computergebruikers.

Een tekstbestand bestaat uit een reeks elektronische tekstlijnen waarin de gegevens staan. Op het einde van het tekstbestand staan meestal een reeks dezelfde karakters, dit wordt een eindbestandmarkering (EOF – End Off File) genoemd om het einde van het tekstbestand aan te kondigen.

3.3.4.1 TXT-formaten

Tekstbestanden kunnen we voornamelijk in drie formaten terugvinden: Windows, Unix en Apple Macintosh tekstbestanden. Ieder formaat hangt af van het besturingssysteem waarop het tekstbestand gebruikt wordt maar ieder formaat heeft dezelfde eigenschappen en functionaliteiten. Deze bestanden kunnen op een eenvoudige manier geïmporteerd worden in een Excel bestand.

3.3.4.2 Windows

Bij een Windows tekstbestand wordt elke regel gescheiden door een combinatie van twee karakters: CR en LF. CR staat voor Carriage Return en LF voor Line Feed. Deze twee karakters zorgen ervoor dat het systeem naar het begin van een regel gaat. Maar dit wordt niet meer gebruikt en het wordt zelfs niet meer automatisch toegevoegd door bepaalde softwares zoals Kladblok of Notepad.

Tekstbestanden op een Windows besturingssysteem zijn te herkennen aan het achtervoegsel (extensie) van de naam van het bestand, dit is “.txt”.

3.3.4.3 Unix

Bij tekstbestanden op een Unix besturingssysteem wordt de tekstbestandsindeling gedetailleerd beschreven. Bij een Unix besturingssysteem wordt een tekstbestand beschouwd als een bestand karakters bevat die in nul of meer lijnen zijn georganiseerd en een beëindigde nieuwe lijn karakter, wat normaal LF is.

3.3.4.4 Apple Macintosh

Op het besturingssysteem van MacOS worden tekstbestanden beschouwd wanneer er in de resourcefork, oftewel een sectie in het bestand, “TEXT” bevat.

4 Scripttalen

4.1 Perl

Perl, ook wel bekend als Practical Extraction and Report Language, is een programmeertaal die gebruikt wordt voor een heel gevarieerd aantal taken. Oorspronkelijk werd Perl ontwikkeld voor tekstmanipulatie uit te voeren. Momenteel kan het ook gebruikt worden in de webontwikkeling, systeem- en netwerkbeheer en nog veel andere domeinen.

Perl wordt vaak op twee manieren geschreven: Perl en perl. Wanneer we het over de programmeertaal zelf hebben, spreken we over Perl met een hoofdletter. Wanneer we over perl met een kleine letter spreken gaat het over de naam van het programma dat Perl scripts gaat uitvoeren.

4.2 Powershell

Powershell is een scripttaal ontwikkeld door Microsoft. Het is automatisch geïnstalleerd op Windows besturingssystemen maar kan manueel ook geïnstalleerd worden op Linux-, macOS- en Unixsystemen. Deze scripttaal is specifiek ontworpen voor de administratietaken te kunnen automatiseren. De eindgebruikers zijn hier meestal systeembeheerders en gebruikers die alle rechten bezitten in het computersysteem.

5 Testen: Powershell versus PERL

Powershell en Perl zijn twee verschillende soorten programmeertalen. Deze twee talen zijn uitvoerig getest op functionaliteit, snelheid en performantie op verschillende soorten bestanden. Alle testen zijn uitgevoerd op een HP Elitebook 850 G4. Deze laptop heeft:

- I7-7500u processor (2.70 GHz)
- 16 GB RAM-geheugen
- Windows 10 pro versie 1709

Beide talen zijn getest op drie functionaliteiten: het transformeren van bestanden, het verplaatsen van bestanden op basis van de bestandsnaam en als laatste om meerdere bestanden samen te voegen tot één groot bestand, ook wel een masterbestand genoemd.

Deze drie scripts zijn telkens getest op 1, 2, 3, 5, 10, 100, 1.000, 10.000 en/of 100.000 bestanden. In het begin is de test 1.000 keer herhaald per groepering van bestanden. Hieruit bleek snel dat de meetgegevens die verkrijgt waren eenduidig waren na tien testen. Hierom zijn er in de analyse telkens tien bestanden ingeladen voor de analyse.

5.1 Voorbereiding

Voor de testen uitgevoerd konden worden is er een voorbereiding moeten gebeuren voor beide talen.

Voor Powershell zijn er drie scripts geschreven die de voorgenoemde taken kunnen uitvoeren alsook de prestaties van het script zelf exporteren naar een CSV-bestand zodat dit later gebruikt kan worden in de analyse. In totaal was er 8u40m nodig om de drie scripts te schrijven zodat deze alle functionaliteiten op een correcte manier toepaste. Powershell is automatisch geïnstalleerd op Windows besturingssystemen, hierdoor is er geen verdere voorbereiding nodig voor Powershell. De versie die gebruikt is van Powershell is 10.0.162.299.15.

Perl heeft, zoals Powershell, drie scripts nodig voor de functionaliteiten uit te voeren. Perl is minder bekende en vernieuwde taal als Powershell en hierdoor is er minder informatie te vinden op het internet. De tijd dat er nodig geweest is om dezelfde scripts te schrijven heeft 21 uur en 10 minuten in beslag genomen. Ook werkt het script voor de samenvoeging niet volledig correct. Dit wordt verder uitgelegd in '5.3 Perl'. Perl moet een omgeving hebben op het Windows systeem en deze moet apart geïnstalleerd worden. Nrgfin maakt gebruik van 'Strawberry Perl' en hiervoor is deze omgeving gebruikt voor de testen in uit te voeren. De versie die gebruikt is tijdens de testen is 5.26.1.1. Het programma neemt ongeveer 700MB in beslag van de harde schijf. Buiten deze voorbereiding moeten er aparte modules geïnstalleerd worden, voor ieder script zijn dit andere modules. De benodigde modules worden bij elke sectie verder uitgelegd.

5.2 Powershell

In de drie scripts van Powershell zijn telkens twee stukken code toegevoegd die de prestaties van het script zelf registreren en deze exporteren naar een CSV-bestand. De code in Afbeelding 29 wordt gebruikt om de tijdsduur te meten bij het uitvoeren van het script. In het begin wordt de starttijd geregistreerd en op het einde van het uitvoeren script de eindtijd. Hierna wordt de huidige datum, de computernaam, de tijdsduur in seconden en in milliseconden weggeschreven in een CSV-bestand dat gebruikt kan worden voor analyse. Dit bestand krijg als benaming 'Time' met hierachter de datum en tijd wanneer dit bestand is aangemaakt. Hierdoor kan er later altijd gekeken worden wanneer welke test is uitgevoerd.

```
$startDTM = (Get-Date)

    Code die gemeten moet worden

$endDTM = (Get-Date)

$Props = @{
    "Date" = Get-Date -Format f
    "Computer Name" = $env:COMPUTERNAME
    "Seconds" = $($endDTM-$startDTM).totalseconds)
    "Milliseconds" = $($endDTM-$startDTM).totalmilliseconds)
}
$Start= Get-Date -Format "yyyyMMddHHmmss"
$MyObject = New-Object -TypeName psubject -Property $Props

$FilePath = "$pwd\" + "Time_$Start" + ".csv"
$MyObject | Export-Csv $FilePath -NoTypeInfo
```

Afbeelding 29 Powershell script tijdsduurmeting

Net zoals voor de tijdsduur is er een stuk code in de scripts geschreven die het gebruik meet van de processor en het geheugen. Zoals te zien is in Afbeelding 30 zijn dit maar twee regels code. De eerste regel definieert het pad waar het CSV-bestand opgeslagen zal worden. De tweede regel zal alle procesinformatie ophalen van het huidige proces, in dit geval dus Powershell.exe. Deze gegevens worden dan ook weggeschreven in een CSV-bestand met als benaming 'Usage' met hierachter de datum en de tijd van wanneer dit CSV-bestand is aangemaakt.

```
$FilePathUsage = "$pwd\" + "Usage_$Start.csv"
get-process -Id $PID | Export-Csv $FilePathUsage -notype
```

Afbeelding 30 Powershell script procesinformatie

5.2.1 Transformatie

Het eerste script [Afbeelding 31] dat geschreven is voor een bestand te transformeren of te converteren naar een andere bestandsextensie. In dit geval wordt een XLSX-bestand geconverteerd naar een CSV-bestand. De reden dat er een CSV-bestand nodig is in plaats van een XLSX is omdat er functies in Qlik zijn die niet werken op XLSX-bestanden maar wel op CSV-bestanden. Eén van deze functies is de 'FileName()', deze gaat de bestandsnaam ophalen van de bestanden die ingeladen worden. De testen zullen uitgevoerd worden om één, twee, drie, vijf en tien bestanden tegelijk te transformeren.

Voor de bestanden van XLSX naar CSV te converteren is het volgende script gebruikt voor Powershell:

```
$Directory = "$pwd"
$FolderToCreate = "$pwd\CSV\"

if (!(Test-Path $FolderToCreate -PathType Container)) {
    New-Item -ItemType Directory -Force -Path $FolderToCreate
}

Function ExcelCSV ($File)
{
    $excelFile = "$pwd\" + $File
    $Excel = New-Object -ComObject Excel.Application
    $Excel.Visible = $false
    $Excel.DisplayAlerts = $false
    $wb = $Excel.Workbooks.Open($excelFile)
    foreach ($ws in $wb.Worksheets)
    {
        $ws.SaveAs("$pwd\CSV\" + $File.BaseName + ".csv", 6)
    }
    $Excel.Quit()
}

Foreach ($Files in (Get-ChildItem -Path $Directory -Filter "*.xlsx"))
{
    ExcelCSV($Files)
}
```

Afbeelding 31 Powershell script transformatie

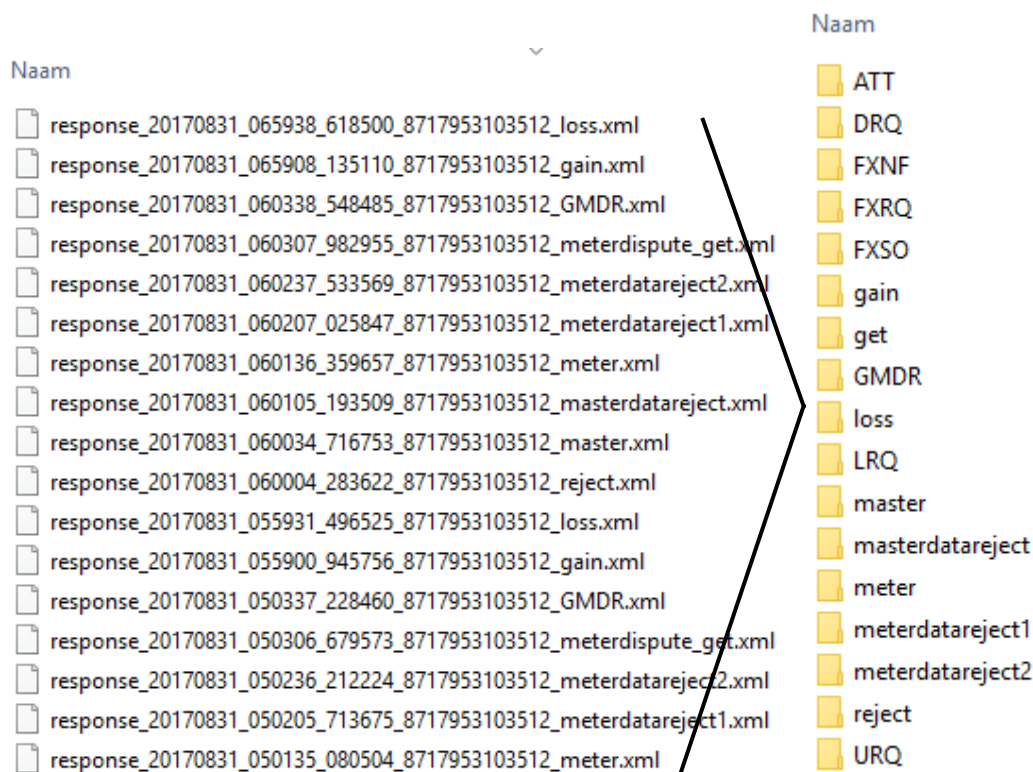
Het script gaat elk bestand ophalen in de huidige map die als extensie '.xlsx' heeft. Powershell maakt per bestand een nieuw Excelobject aan en gaat hierin elke regel uit het XLSX-bestand opnemen. Hierna schrijft het script deze in het nieuw aangemaakte CSV-bestand. Dit proces blijft zich herhalen tot alle regels in alle bestanden zijn weggeschreven in het CSV bestand.

5.2.2 Verplaatsing

In dit script wordt eerst een variabele aangemaakt die het begin bevat van de locatie waar de bestanden naar verplaatst worden. Hierna is te zien in Afbeelding 32 dat alle XML-bestanden in de folder opgehaald worden waarin het Powershell script opgeslagen is. Elk XML-bestand zijn bestandsnaam wordt opgehaald en deze wordt eerst opgesplitst in twee stukken. Het eerste gedeelte is alles voor het puntsymbool (.), het tweede gedeelte is alles na het puntsymbool. Dit gebeurt zodat de extensie van het bestand niet meegenomen wordt. Vervolgens wordt de naam nog eens opgedeeld, deze keer op basis van een underscore (_), hier wordt het laatste gedeelte van meegenomen. Dit gedeelte omvat het onderwerp van het bestand en wordt gebruikt om het bestand te verplaatsen naar zijn betreffende folder [Afbeelding 33]. Dit script is uitgevoerd op 100,1.000,10.000 en 100.000 XML-bestanden.

```
$DestinationRoot = "$pwd\splitsing\"  
  
foreach ($file in gci *.xml){  
  
    $fname = ($file.name).split('.')[0]  
    $last = $fname.split('_')[-1]  
    $Destination = "$DestinationRoot$last"  
  
    Move-Item -Path $file.fullname -Destination $Destination -Force  
}
```

Afbeelding 32 Powershell script verplaatsing



Afbeelding 33 XML-berichten en mappenstructuur

5.2.3 Samenvoegen

XML-bestanden samenvoegen tot één groot bestand was het moeilijkste script om te schrijven. Zoals eerder vermeld bestaat een XML-bericht uit een boomstructuur waardoor de data op verschillende niveaus staat binnen het bericht. Zoals te zien is op Afbeelding 34 wordt eerst een XML-object aangemaakt samen met een lege boomstructuur. Vervolgens worden alle XML-bestanden in de huidige map opgehaald en worden deze opgeslagen in de variabele '\$files'.

Elk bestand wordt opengemaakt en de interne boomstructuur wordt bekeken. Per niveau wordt elk regel opgehaald en opgeslagen in de variabele '\$rootnode'. Wanneer een XML-bestand het hele proces heeft doorlopen wordt deze verplaatst naar een andere folder genaamd 'Merged'. Dit is een archiefmap zodat het aparte XML-bestand nog altijd bewaard blijft.

Als alle XML-bestanden door het proces zijn gegaan wordt het masterbestand aangemaakt en de opgeslagen data hierin geschreven. Als laatste wordt de verbinding om te kunnen schrijven naar het masterbestand afgesloten.

Dit script is uitgevoerd op 1.000, 5.000 en 10.000 XML-bestanden.

```
$xmlDoc = new-object xml
$rootnode = $xmlDoc.createelement("xml")
$xmlDoc.appendChild($rootnode)
$files = gci *.xml

foreach ($file in $files) {
    [xml]$xmlstuff = gc $file.fullname
    $innerel = $xmlstuff.selectnodes("/*/*")
    foreach ($inone in $innerel) {
        $inone = $xmlDoc.importnode($inone, $true)
        $rootnode.appendChild($inone)
    }
    New-Item "$pwd\Merged\" -ItemType Directory -Force | Out-Null
    Move-Item -Path $file.fullname -Destination "$pwd\Merged\" -Force
}
foreach ($t34 in $rootnode.selectnodes("//*/text()")) {
    $t34.innertext = [regex]::replace($t34.innertext, "\s+", " ")
}

$xws = new-object system.xml.XmlWriterSettings
$xws.Indent = $true
$xws.indentchars = "`t"
$xtw = [system.xml.XmlWriter]::create("$pwd\final_Master.xml", $xws)
$xmlDoc.WriteContentTo($xtw)
$xtw.flush()
$xtw.dispose()
```

Afbeelding 34 Powershell script samenvoeging

5.3 Perl

Zoals eerder vermeld heeft Perl extra modules nodig om bepaalde functionaliteiten te kunnen uitvoeren in zijn scripts. Voor de drie scripts die gebruikt zijn tijdens de testen zijn de volgende modules apart geïnstalleerd:

- Spreadsheet::XLSX
 - Module om bewerkingen uit te voeren op alle soorten Excelbestanden.
- TEXT::CSV
 - Deze module is specifiek gebruikt om CSV-bestanden te kunnen creëren. Dit is voornamelijk gebruikt om de meetresultaten in op te slaan en in het script voor transformatie.
- File::Find
 - Een module om een gedetailleerde zoekopdracht te creëren naar specifieke bestanden.
- Time::HiRes
 - Deze module is enkel gebruikt voor de meetresultaten, meer bepaald om de tijd te registreren hoelang een script loopt.
- Win32::Process::Info
 - Perl heeft verschillende functies voor Linuxsystemen en Windowssystemen. Bij Linux is er een lokale functie die gebruikt kan worden op de procesgegevens op te halen. Bij Windowssystemen is dit niet aanwezig, hiervoor dient deze module.

Om deze modules zo eenvoudig mogelijk te installeren moeten de volgende stappen uitgevoerd worden:

1. Open de Opdrachtenprompt (CMD)
2. Typ: App::cpanminus
 - a. Dit commando gaat een connectie leggen naar de database waar alle modules beschikbaar zijn. Hierdoor moet er niet specifiek gezocht worden naar de modules en moeten deze niet manueel geïnstalleerd worden. Het enige wat er moet geweten zijn is de modulenaam.
3. Typ: cpanm 'gevolgd door de modulenaam'
 - a. cpanm Spreadsheet::XLSX
 - b. cpanm Text::CSV
4. Wacht tot er staat 'Distribution installed' en de module kan gebruikt worden in Perl scripts.
[Afbeelding 35]

```
C:\Users\Simaes>cpanm Spreadsheet::XLSX
--> Working on Spreadsheet::XLSX
Fetching http://www.cpan.org/authors/id/M/MI/MIKEB/Spreadsheet-XLSX-0.15.tar.gz ... OK
Configuring Spreadsheet-XLSX-0.15 ... OK
Building and testing Spreadsheet-XLSX-0.15 ... OK
Successfully installed Spreadsheet-XLSX-0.15
1 distribution installed
```

Afbeelding 35 Module succesvol geïnstalleerd

Om de tijdsduur te bepalen van de Perl-scripts is hetzelfde principe gevolgd als bij Powershell. De code die getest wordt staat tussen twee stukken code. Zoals te zien is op Afbeelding 36 gaat de starttijd opgeslagen worden in een variabele. De code begint zijn functies te starten en op het einde wordt de eindtijd opgeslagen.

De datum en tijd worden ook opgeslagen wanneer de test voorbij is. De variabele waarin de tijd opgeslagen is worden de dubbele punten (:) eruit gehaald omdat deze gebruikt wordt voor de bestandsnaam en deze geen dubbele punten accepteert.

De eerste die aangemaakt wordt zijn de kolomnamen: Execution time, date en time. Hieronder volgen de argumenten die zijn opgeslagen tijdens de test. Er wordt gekeken of de foldermap al bestaat waarin deze CSV-bestanden in opgeslagen worden, indien dit niet het geval is gaat deze map aangemaakt worden.

Als laatste wordt het CSV-bestand aangemaakt waarin de eerste en de tweede rij in opgeslagen zullen worden en waarin de snelheidsresultaten dus instaan. [8] [9]

```
my $start = Time::HiRes::gettimeofday();

    Code die getest moet worden

my $end = Time::HiRes::gettimeofday();
my $timeScript = $end - $start;

$dt = DateTime ->now;
my $date = $dt->ymd;
my $time = $dt->hms;
$time =~ s/[[:]]+//g;

my @firstline = ("Execution time, date, time");
my @arr = ($timeScript,$date,$time);

if (! -e $timeloc) {
    mkdir($timeloc);
}

my $csv = Text::CSV->new ( { always_quote => 1,eol => $/ } );
open my $fh, '>', $timeloc . '/' . $date . ' ' . $time . '.csv' or die $!;
$csv->print( $fh, \@firstline);
$csv->print( $fh, \@arr );
```

Afbeelding 36 Perl script tijdsduurmeting

Voor de verbruikresultaten van het proces te verkrijgen is meer code nodig. Op Afbeelding 37 is te zien dat de module voor de procesinformatie wordt opgehaald. Hieruit wordt een lijst verkregen met alle processen die momenteel actief zijn op de computer. Met de '%ofInterest'-functie wordt er gespecificeerd dat enkel de informatie van Perl.exe verder meegenomen moet worden. [10]

```

use Win32::Process::Info;
my $pio_2 = Win32::Process::Info->new ();
my @pids = $pio_2->ListPids ();
my @proc_info;

my %ofInterest = map { $_ => 1 } qw[
    perl.exe
];

my @fields = qw[ Name WorkingSetSize UserModeTime KernelModeTime ];

my @firstline = ("WorkingSetSize, UserModeTime, KernelModeTime");
$location= '10 files';
$usageloc = 'Usage' . $location;
$timeloc= 'Time execution' . $location;

if (! -e $usageloc) {
    mkdir($usageloc);
}

$dt = DateTime ->now;
my $date = $dt->ymd;
my $time = $dt->hms;
$time =~ s/[:]+//g;

my $csv = Text::CSV->new ( { always_quote => 1,eol => $/ } );
open my $fh, '>', $usageloc . '/' . $date . ' ' . $time . '.csv' or die $!;
$csv->print( $fh, \@firstline);

for my $pid (@pids) {
    @proc_info = $pio_2->GetProcInfo( { no_user_info => 1 }, $pid);
    for my $info (
        grep{ exists $ofInterest{ lc $_->{ Name } } } @proc_info
    ) {
        my $WorkingSetSize = $info->{ WorkingSetSize };
        my $UserModeTime = $info->{ UserModeTime };
        my $KernelModeTime = $info->{ KernelModeTime };

        my @arr = ($WorkingSetSize,$UserModeTime,$KernelModeTime);
        $csv->print( $fh, \@arr );
    }
}

```

Afbeelding 37 Perl script procesinformatie

Zoals bij de snelheidsmeting wordt hier een eerste rij aangemaakt met de kolomnamen. Wanneer de Foldermap voor deze resultaten nog niet bestaat gaat deze ook aangemaakt worden. Het proces om de meetresultaten naar het CSV-bestand te schrijven verloopt anders als bij de snelheidsmeting. Hier wordt het CSV-bestand al aangemaakt voordat de meetresultaten in een variabele staan. De eerste rij met de kolomnamen wordt al in het bestand geplaatst maar de meetresultaten nog niet.

De procesgegevens worden elk apart opgeslagen in een variabele, die nadien in een array gestoken worden. Deze array gaat als laatste stap dan opgeslagen worden in het CSV-bestand.

5.3.1 Transformatie

In dit script worden alle bestanden opgehaald in de huidige map die als extensie 'xlsx' hebben. Perl maakt per maar één nieuw Excelobject aan en gaat hierin elke regel uit het XLSX-bestand opnemen. Het verschil bij Powershell is dat Perl niet de volledige rij ophaalt maar veld per veld. Hierdoor moeten de velden op dezelfde rij samengevoegd worden met een komma (,) tussenbeide. Deze rijen worden dan uiteindelijk in een CSV-bestand geschreven. [Afbeelding 38] [8] [9] [10]

```
use Spreadsheet::XLSX;
use File::Copy;
use File::Basename;
use Time::HiRes;
use Text::CSV;
use DateTime;
use Cwd qw(cwd);

my @files = glob("*.xlsx");

foreach $file (@files) {
    $name = fileparse($file, qr/\.[^.]*/);
    my $dir = cwd . "/" . $file;

    my $excel = Spreadsheet::XLSX->new("$dir");
    my $sheet = ${$excel->{Worksheet}}[0];

    my ($row_min, $row_max) = $sheet->row_range();
    my ($col_min, $col_max) = $sheet->col_range();

    open (FH,'>', 'converted/' . $name . '.csv') or die '$!';
    for my $row ($row_min..$row_max)
    {
        print FH join (',', map {$sheet->get_cell($row,$_)->value()} ($col_min..$col_max));
        print FH "\n";
    }

    close (FH);
}
```

Afbeelding 38 Perl script transformatie

5.3.2 Verplaatsing

Het Powershell-script en het Perl-script verschillen bijna niets van structuur bij de verplaatsing functionaliteit. Alle XML-bestanden worden opgehaald, deze worden opgesplitst op het punt (.) en de underscore (_) en hiervan wordt het laatste gedeelte genomen. Deze benaming wordt gebruikt om een specifieke folder te creëren voor dit type bestand indien het nog niet bestaat en vervolgens wordt het XML-bestand verplaatst naar deze locatie. [Afbeelding 39] [8] [10]

```
use Time::HiRes;
use Text::CSV;
use DateTime;
use Cwd qw(cwd);

@files = glob("*.xml*");

foreach $file (@files) {
    $name = fileparse($file, qr/\.[^.]*/);
    $basename = basename($file);

    my $foo = (split '_', $name)[-1];
    $newloc = cwd . "/Splitsing/" . $foo . "/" . $basename;
    $folder = "Splitsing/" . $foo;

    if (! -e $folder) {
        mkdir($folder);
    }

    move($file, $newloc);
}
```

Afbeelding 39 Perl script verplaatsing

5.3.3 Samenvoegen

Het samenvoegen van XML-bestanden is Perl is niet simpel. Het script dat gebruikt wordt tijdens de testen werkt ook niet volledig correct en hierbij wordt rekening gehouden met het eindresultaat en de conclusie.

Eerst wordt er gekeken of de array ARGV nul is, ARGV is een speciale array binnen Perl die argumenten kan bewaren van de commandolijn. Indien deze nul is wordt het pad meegegeven waarin de XML-bestanden te vinden zijn.

Hierna wordt het masterbestand aangemaakt, dit is echter een Combined-bestand dan een XML-bestand. In dit masterbestand zal alle data terechtkomen die in de XML-bestanden staat. De XML-bestanden worden opgehaald in de folder die opgeslagen is in de ARGV variabele. Per XML-bestand wordt de interne data opgehaald en opgeslagen in het masterbestand. [Afbeelding 40]

Echter zijn hier een paar problemen waardoor deze test niet volledig accuraat zal zijn als bij de andere testen. Het masterbestand is een Combined-bestand en deze moet manueel nog omgevormd worden tot een XML-bestand. Ook neemt de Perlfunctionaliteit niet alle regels mee uit het XML-bestand of koppelt hij de verkeerde velden bij elkaar. Zoals uitgelegd in punt 3.3.1.1. is een XML opgebouwd uit een boomstructuur. Wat er nu verkeerd gaat is dat de Perlfunctionaliteit de verschillende takken van de boom niet langs elkaar zet in het masterbestand maar elke tak een aparte rij geeft of wanneer er één veld gemeenschappelijke is de takken op een verkeerde hoogte langs elkaar zetten.

Hierdoor is deze test dus niet volledig accuraat waarbij er zeker rekening zal gehouden worden bij de conclusie.

```
#!/usr/lib/perl
use strict;
use warnings;
use Carp;
use File::Find;
use File::Spec::Functions qw( canonpath );
use Spreadsheet::XLSX;
use File::Copy;
use File::Basename;
use Time::HiRes;
use Text::CSV;

if ( @ARGV == 0 ) {
    push @ARGV, 'C:\Users\Simae\Documents\Onderzoek\Merging';
    warn "Using default path $ARGV[0]\n Usage: $0 path ...\n";
}

open( my $allxml, '>', "all_xml_contents.combined" )
    or die "can't open output xml file for writing: $!\n";
print $allxml '<?xml version="1.0" encoding="UTF-8"?>', "\n<all_xml_contents>\n";

find(
    sub {
        return unless ( /[.]xml\z/i and -f );
        extract_information();
        return;
    },
    @ARGV
);

print $allxml "</all_xml_contents>\n";

sub extract_information {
    my $path = $_;

    if ( open my $xmlin, '<', $path ) {
        local $_ = <$xmlin>;
        print $allxml $_ unless ( /<\?xml/ );
        while ( <$xmlin> ) {
            print $allxml $_;
        }
    }
    return;
}
```

Afbeelding 40 Perl script samenvoeging

5.4 Resultaten

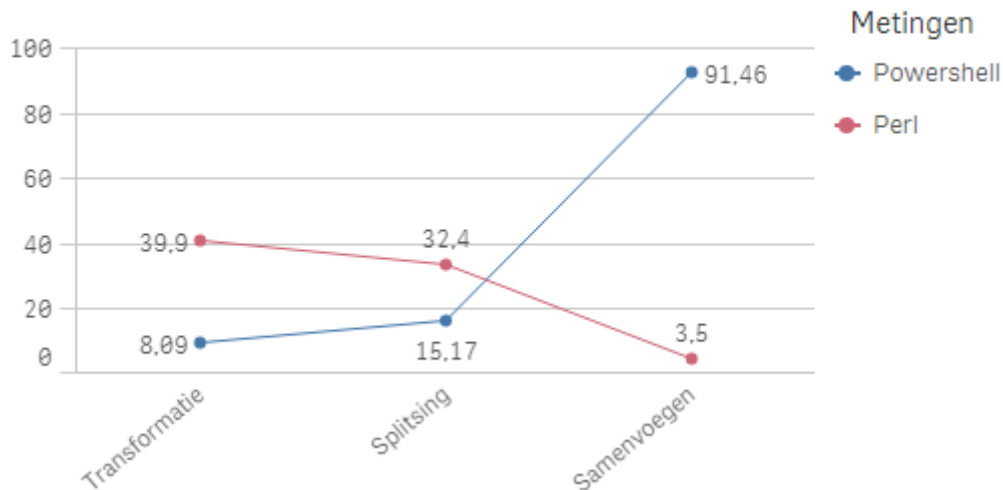
Alle resultaten zijn gecombineerd en opgeladen in een Qlik-applicatie. Hierin kan er eenvoudig aangetoond worden welke programmeertaal het snelste en/of efficiënt werkt. In de volgende afbeeldingen kan het overzicht te zien zijn van alle resultaten, deze zijn door middel van Perl te vergelijken met Powershell. Indien het procent negatief is betekent dit dat Powershell beter is, indien positief is Perl beter.

Als we kijken naar Afbeelding 41 en Afbeelding 42, dan zien we dat Powershell op gebied van het script uit te voeren veel sneller is als Perl. Bij de transformatie gemiddeld 79% en bij de splitsing 53%. Echter zijn wij dat bij het samenvoegen Powershell 2520% trager is als Perl, dit komt grotendeels door de problemen die bij et Perl-script voorkomen. Hierom zijn deze gegevens in het oranje geplaatst. Als we deze test opzij leggen zien we in Afbeelding 42 dat Powershell een heel stuk onder Perl ligt over het aantal seconden dat het script nodig heeft om helemaal uit te voeren.



Afbeelding 41 KPI's gemiddelde tijd

Tijdsduur



Afbeelding 42 Lijngrafiek gemiddelde tijd

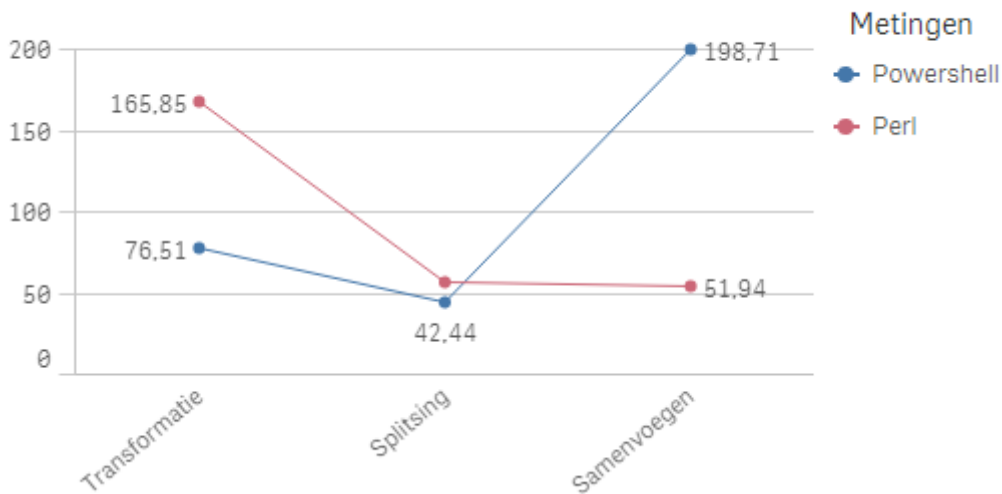
Hetzelfde op het gebied van inbeslagname van geheugen, Powershell gebruikt 22% tot 53% minder geheugen tijdens de uitvoering dan Perl. Hierdoor wordt er gemiddeld 10 tot 90MB bespaard. Dit maakt een groot verschil als er bijvoorbeeld tien processen aan het lopen zijn. Tijdens de uitvoer van deze processen is er 100 tot 900 MB geheugen vrij. Dat is 10% extra werkkraft op een HP Elitebook G4. [Afbeelding 43] en [Afbeelding 44]

Echter is hier het verschil bij het samenvoegen weer groot maar dit komt grotendeels door de problemen in het Perl-script voor XML-bestanden samen te voegen.



Afbeelding 43 KPI's gemiddeld geheugenverbruik

Computergeheugen



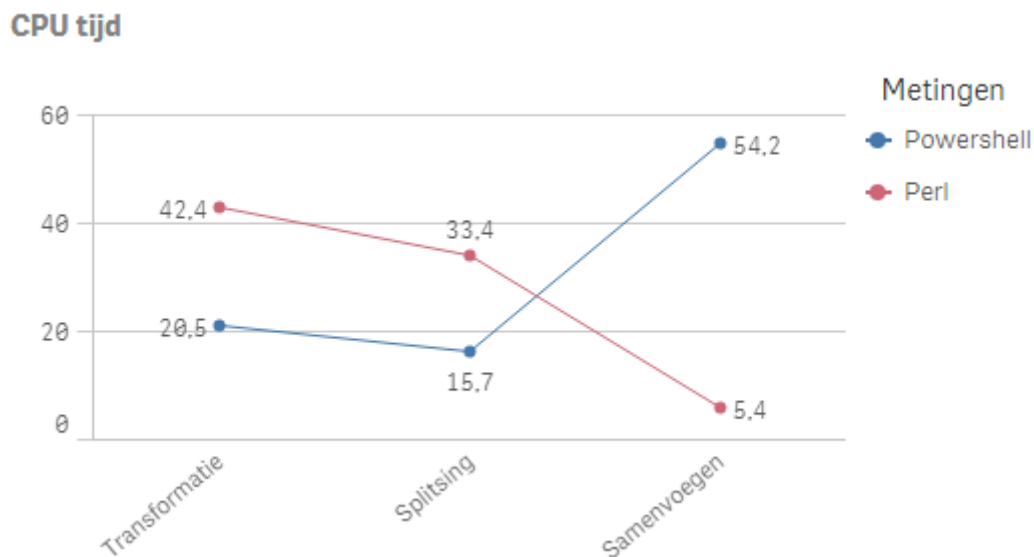
Afbeelding 44 Lijngrafiek gemiddeld geheugenverbruik

Als laatste is er gekeken naar de tijd dat de CPU in beslag genomen is en zoals bij de andere twee domeinen is Powershell hier meer als 50% efficiënter dan Perl. Powershell belast de CPU twee keer zo weinig als Perl zou doen. Dit is vooral voordelig zodat de huidige processen die aan het lopen zijn niet gaan vasthangen of zelfs crashen. Indien een proces crasht moet de hele uitvoering opnieuw gebeuren wat een ramp is voor een bedrijf. [Afbeelding 45] en [Afbeelding 46]

Een deel van de aparte resultaten zijn opgenomen in de bijlagen. Voor de resultaten van Powershell kan er gekeken worden naar Bijlages E,F en G. De resultaten van Perl zijn te vinden in Bijlages H, I en J.



Afbeelding 45 KPI's gemiddeld CPU verbruik



Afbeelding 46 Lijngrafiek gemiddeld CPU verbruik

Conclusie

In het eerste gedeelte van dit onderzoek zijn de energiedatamodellen van België en Nederland geanalyseerd en vergeleken geweest bij elkaar. Dit komt omdat het datamodel van Nederland op verschillende gebieden verschilt met dat van België.

Tijdens dit onderzoek is er ondervonden dat er domeinen zijn die België en Nederland hetzelfde hebben of uitvoeren maar ook verschillen.

De domeinen die hetzelfde zijn voor beide landen zijn:

- Type elektriciteits- en gasmeters (op één na, de digitale meter)
- Het gebruik van volgende bestanden:
 - XLSX
 - CSV
 - TXT

Echter zijn er ook verschillen tussen beide modellen. Het grootste verschil is dat België gebruikmaakt van EDIEL-berichten en Haugazel om deze te versturen. De verbruiksprofielen zijn gebaseerd op kWh en deze beginnen allemaal met een 's' gevolgd door een nummer.

Nederland maakt gebruik van XML-berichten, België gaat dit in de toekomst ook invoeren dus op dit domein is Nederland sneller. Ook maakt Nederland momenteel gebruik van een SFTP om de bestanden te versturen. De verbruiksprofielen van Nederland zijn gebaseerd op ampère en de opdeling hiervan is sneller te herkennen voor een leek. De profielen voor elektriciteit beginnen met een 'E' en de profielen van gas met een 'G'. Als laatste ligt het Nederlands energiedatamodel licht voor op gebied van meters. Nederland maakt al gebruik van de slimme meter die in België waarschijnlijk pas in 2019 ingevoerd zal worden.

Bij de praktijkstudie zijn Powershell en Perl vergeleken met elkaar op basis van snelheid en gebruik van computergeheugen. Dit op basis van drie functionaliteiten: transformeren, verplaatsing en samenvoeging van respectievelijk: 1, 2, 3, 5, 10, 100, 1.000, 10.000 en/of 100.000 bestanden.

Uit deze studie is Powershell als snelste en efficiëntste programmeertaal voortgekomen op de functionaliteiten: transformeren en verplaatsing. Powershell duurde 50%-80% minder lang als Perl en gebruikte 50% minder geheugen en CPU. Het enige gebied waar Powershell het slechter heeft gedaan als Perl is bij de samenvoeging van bestanden. Echter werkt het Perl-script voor de samenvoeging niet volledig waardoor deze resultaten niet accuraat kunnen zijn.

Ook op vlak van het creëren van de scripts en de voorbereiding die nodig geweest is voor de tests was bij Powershell makkelijker en sneller.

Hierdoor is besloten dat Powershell sneller en efficiënter is als Perl.

Bibliografie

- [1] „Atrias publicaties,” Atrias, [Online]. Available: http://www.atrias.be/NL/Paginas/Publications_UMIG41.aspx. [Geopend 11-01 2017-2018].
- [2] „VREG,” [Online]. Available: <http://www.vreg.be/nl/metertypes>. [Geopend 20 12 2017].
- [3] „EDI,” [Online]. Available: <https://www.edibasics.com/what-is-edi/>. [Geopend 10-01 2017-2018].
- [4] „Atrias,” [Online]. Available: <http://www.atrias.be/NL/Paginas/Home.aspx>. [Geopend 10-01 2017-2018].
- [5] „Nedu profielen,” [Online]. Available: <http://www.nedu.nl/portfolio/verbruiksprofielen/>. [Geopend 11-01 2017-2018].
- [6] „XMLFiles,” [Online]. Available: <https://www.xmlfiles.com/xml/xml-syntax/>. [Geopend 11-01 2017-2018].
- [7] „w3schools,” [Online]. Available: https://www.w3schools.com/xml/xml_syntax.asp. [Geopend 10-01 2017-2018].
- [8] „Perlmaven,” [Online]. Available: <https://perlmaven.com/>. [Geopend 01 2018].
- [9] „Perldoc,” [Online]. Available: <http://perldoc.perl.org/>. [Geopend 01 2018].
- [10] „Perlmonks,” [Online]. Available: <http://www.perlmonks.org>. [Geopend 01 2018].
- [11] „EABO,” EnergyAssets BackOffice, [Online]. Available: <http://www.eabo.nl/>. [Geopend 10-01 2017-2018].
- [12] „Metacpan,” [Online]. Available: <https://metacpan.org/>. [Geopend 01 2018].
- [13] „Tempo,” JIRA, [Online]. Available: <https://nrgfin.atlassian.net>. [Geopend 11-12 2017].

Bijlagen

- A. Synthetische lastprofielen**
- B. Volledig datastructuur Belgsich energiedatamodel**
- C. Links website en social media**
- D. Grafisch design**
- E. Meetresultaten transformatie Powershell**
- F. Meetresultaten verplaatsing Powershell**
- G. Meetresultaten samenvoeging Powershell**
- H. Meetresultaten transformatie Perl**
- I. Meetresultaten verplaatsing Perl**
- J. Meetresultaten samenvoeging Perl**

A. Synthetische lastprofielen

SLP	Beschrijving
B17	Monthly
B18	Yearly
BA0	Injection / Infeed
E13	Continuous
E17	Consumption
E18	Production
S10	'RLP' Real Load Profile Electricity
S11	'SLP' Industrial < 56 kVA Electricity
S12	'SLP' Industrial >= 56 kVA Electricity
S21	'SLP' Residential night/day <1,3, also only day - Electricity
S22	'SLP' Residential night/day >= 1,3 - Electricity
S30	'RLP' Real Load Profile Gas
S31	'SLP' < 150 000 kWh {Industrial heating} - Gas
S32	'SLP' >= 150 000 kWh {Industrial Process} - Gas
S41	'SLP' Residential Gas
S70	S70 'LP' Net Losses - Electricity
S71	'LP' Net Infeed - Electricity
S78	'KCF' Climate Correction Factor – Gas
S79	'KCF' Climate Correction Factor – Electricity - Is provided even if equal "1"
S80	'LP' Sum S10 Consumption + S70
S88	'RF' Residu Factor - Gas
S89	'RF' Residu Factor - Electricity
S98	'LP' Sum of all RLP + SLP Gas
S99	'LP' Sum of all RLP + SLP Electricity

B. Links website en sociale media

Nrgfin

<http://www.nrgfin.be/>

<http://www.nrgfin.nl/>

<https://www.linkedin.com/company/dobe-energy/>

Comigoo

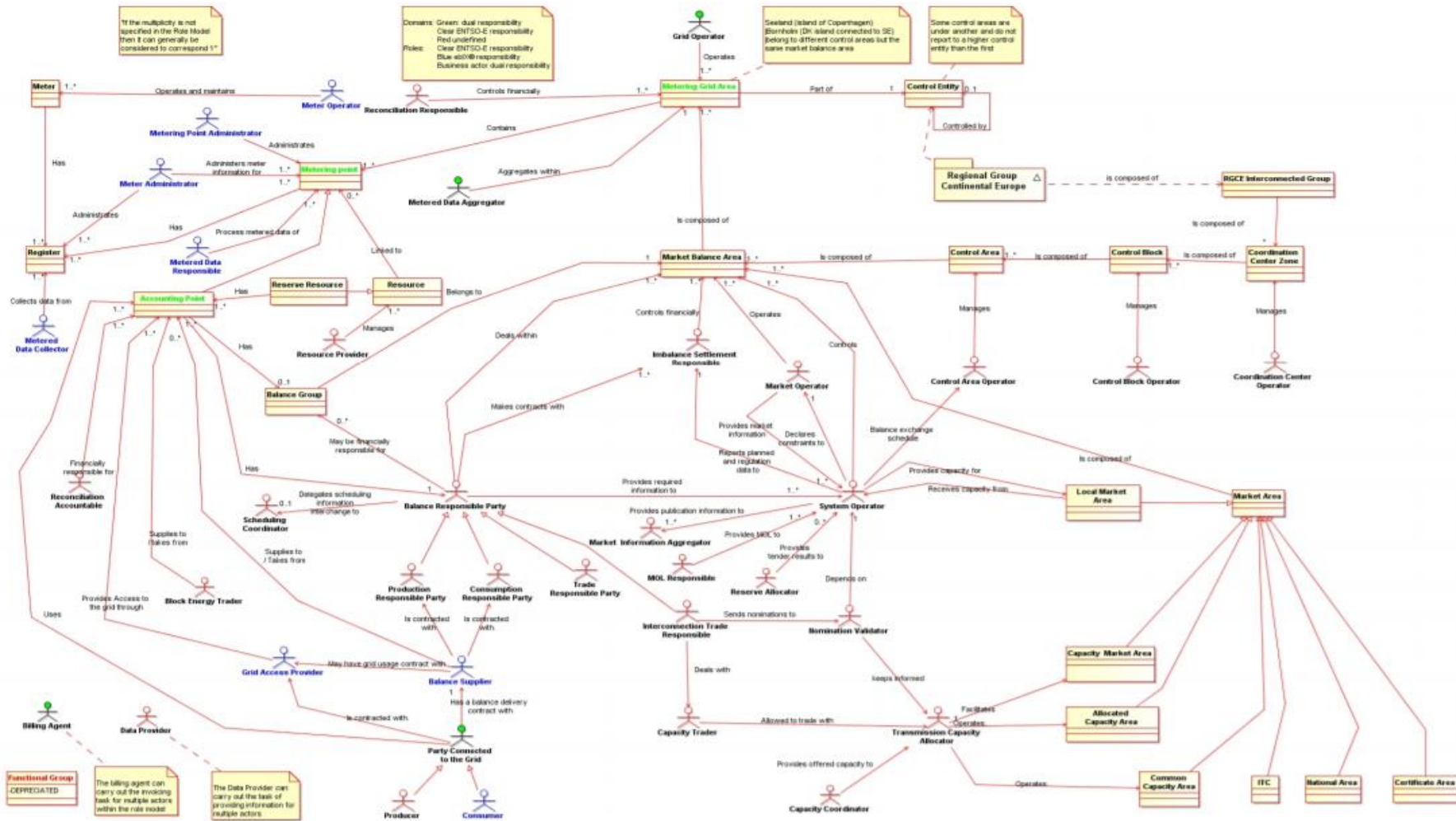
<https://comigoo.be/>

<https://www.facebook.com/ComigooBE/>

<https://www.linkedin.com/company/comigoo/?originalSubdomain=nl>

<https://twitter.com/ComigooBE>

C. Omschrijving Bijlage



D. Grafisch design



E. Meetresultaten transformatie Powershell

	A	B		A	B	C	D
1	Datum	Seconden	1	Datum	CPU	WS MB	WS kB
2	woensdag 17 januari 2018 14:28	3,4204786	2	woensdag 17 januari 2018 14:28	5,828125	76,10546875	77932
3	woensdag 17 januari 2018 14:28	3,654952	3	woensdag 17 januari 2018 14:28	6,078125	71,953125	73680
4	woensdag 17 januari 2018 14:30	4,0894708	4	woensdag 17 januari 2018 14:30	5,4375	71,36328125	73076
5	woensdag 17 januari 2018 14:31	3,2446411	5	woensdag 17 januari 2018 14:31	5,328125	71,59375	73312
6	woensdag 17 januari 2018 14:31	3,2504889	6	woensdag 17 januari 2018 14:31	5,4375	71,7734375	73496
7	woensdag 17 januari 2018 14:31	3,3320149	7	woensdag 17 januari 2018 14:31	5,78125	72,046875	73776
8	woensdag 17 januari 2018 14:31	3,3899248	8	woensdag 17 januari 2018 14:31	6,109375	71,71875	73440
9	woensdag 17 januari 2018 14:33	3,4439619	9	woensdag 17 januari 2018 14:33	5,703125	73,6796875	75448
10	woensdag 17 januari 2018 14:38	3,3215672	10	woensdag 17 januari 2018 14:38	5,359375	72,56640625	74308
11	woensdag 17 januari 2018 14:38	3,3687893	11	woensdag 17 januari 2018 14:38	6	77,00390625	78852
12	woensdag 17 januari 2018 15:43	5,7444782	12	woensdag 17 januari 2018 15:43	6,21875	76,875	78720
13	woensdag 17 januari 2018 15:43	5,7872693	13	woensdag 17 januari 2018 15:43	6,328125	77,62109375	79484
14	woensdag 17 januari 2018 15:43	6,0293278	14	woensdag 17 januari 2018 15:43	6,375	77,609375	79472
15	woensdag 17 januari 2018 15:43	6,1856812	15	woensdag 17 januari 2018 15:43	6,453125	77,59765625	79460
16	woensdag 17 januari 2018 15:44	5,7032764	16	woensdag 17 januari 2018 15:44	6,53125	77,75	79616
17	woensdag 17 januari 2018 15:44	5,9022279	17	woensdag 17 januari 2018 15:44	6,578125	77,73828125	79604
18	woensdag 17 januari 2018 15:44	5,9120525	18	woensdag 17 januari 2018 15:44	6,625	77,78515625	79652
19	woensdag 17 januari 2018 15:44	5,9778533	19	woensdag 17 januari 2018 15:44	6,6875	77,75	79616
20	woensdag 17 januari 2018 15:44	6,001188	20	woensdag 17 januari 2018 15:44	6,71875	77,65234375	79516
21	woensdag 17 januari 2018 15:44	6,0388527	21	woensdag 17 januari 2018 15:44	6,78125	77,86328125	79732
22	woensdag 17 januari 2018 16:56	4,8610782	22	woensdag 17 januari 2018 16:56	4,90625	71,1796875	72888
23	woensdag 17 januari 2018 16:56	5,1452763	23	woensdag 17 januari 2018 16:56	5,0625	72,63671875	74380
24	woensdag 17 januari 2018 16:56	5,1481666	24	woensdag 17 januari 2018 16:56	5,171875	74,25	76032
25	woensdag 17 januari 2018 16:56	5,1938855	25	woensdag 17 januari 2018 16:56	5,1875	74,3828125	76168
26	woensdag 17 januari 2018 16:56	5,2192978	26	woensdag 17 januari 2018 16:56	5,3125	74,39453125	76180
27	woensdag 17 januari 2018 16:56	5,2568209	27	woensdag 17 januari 2018 16:56	5,359375	74,515625	76304
28	woensdag 17 januari 2018 16:56	5,7506035	28	woensdag 17 januari 2018 16:56	5,375	74,484375	76272
29	woensdag 17 januari 2018 16:57	4,7198479	29	woensdag 17 januari 2018 16:57	5,421875	73,91015625	75684

F. Meetresultaten verplaatsing Powershell

	A	B		A	B	C	D
1	Datum	Seconden	1	Datum	CPU	WS MB	WS kB
2	donderdag 18 januari 2018 22:25	14,7353098	2	donderdag 18 januari 2018 14:02	0,796875	37,7546875	38660,8
3	donderdag 18 januari 2018 22:26	12,7385031	3	donderdag 18 januari 2018 14:04	0,671875	37,48125	38380,8
4	donderdag 18 januari 2018 22:27	19,035615	4	donderdag 18 januari 2018 14:04	0,703125	37,5359375	38436,8
5	donderdag 18 januari 2018 22:28	14,5970855	5	donderdag 18 januari 2018 14:04	0,75	37,596875	38499,2
6	donderdag 18 januari 2018 22:29	12,7387679	6	donderdag 18 januari 2018 14:04	0,765625	37,625	38528
7	donderdag 18 januari 2018 22:29	13,4233726	7	donderdag 18 januari 2018 14:04	0,796875	37,54375	38444,8
8	donderdag 18 januari 2018 22:30	13,7439556	8	donderdag 18 januari 2018 14:04	0,8125	37,578125	38480
9	donderdag 18 januari 2018 22:35	13,8815043	9	donderdag 18 januari 2018 14:05	0,78125	37,6171875	38520
10	donderdag 18 januari 2018 22:41	12,9925189	10	donderdag 18 januari 2018 14:15	0,625	37,1359375	38027,2
11	donderdag 18 januari 2018 22:41	13,4680717	11	donderdag 18 januari 2018 14:15	0,671875	36,875	37760
12	donderdag 18 januari 2018 22:49	129,2524681	12	donderdag 18 januari 2018 14:15	0,671875	36,9734375	37860,8
13	donderdag 18 januari 2018 22:52	85,4431737	13	donderdag 18 januari 2018 14:15	0,859375	37,221875	38115,2
14	donderdag 18 januari 2018 22:58	83,4943433	14	donderdag 18 januari 2018 14:16	0,640625	37,1703125	38062,4
15	donderdag 18 januari 2018 23:02	83,0664873	15	donderdag 18 januari 2018 14:16	0,65625	37,15	38041,6
16	donderdag 18 januari 2018 23:07	79,5328984	16	donderdag 18 januari 2018 14:16	0,671875	37,09375	37984
17	donderdag 18 januari 2018 23:10	85,6823168	17	donderdag 18 januari 2018 14:16	0,671875	37,3	38195,2
18	donderdag 18 januari 2018 23:13	83,7454486	18	donderdag 18 januari 2018 14:16	0,6875	37,096875	37987,2
19	donderdag 18 januari 2018 23:17	80,028544	19	donderdag 18 januari 2018 14:16	0,6875	37,1109375	38001,6
20	donderdag 18 januari 2018 23:26	81,7485072	20	donderdag 18 januari 2018 14:16	0,6875	37,2609375	38155,2
21	donderdag 18 januari 2018 23:30	83,7060024	21	donderdag 18 januari 2018 14:16	0,703125	36,9828125	37870,4
22	vrijdag 19 januari 2018 13:18	115,6576774	22	donderdag 18 januari 2018 14:16	0,71875	37,378125	38275,2
			23	donderdag 18 januari 2018 14:16	0,8125	37,0484375	37937,6
			24	donderdag 18 januari 2018 14:17	0,625	37,2734375	38168
			25	donderdag 18 januari 2018 14:17	0,65625	37,103125	37993,6
			26	donderdag 18 januari 2018 14:17	0,6875	37,090625	37980,8
			27	donderdag 18 januari 2018 14:17	0,703125	37,1	37990,4
			28	donderdag 18 januari 2018 14:17	0,703125	37,1328125	38024
			29	donderdag 18 januari 2018 14:17	0,703125	37,1953125	38088

G. Meetresultaten samenvoeging Powershell

	A	B		A	B	C	D
1	Datum	Seconden	1	Datum	CPU	WS MB	WS kB
2	woensdag 17 januari 2018 14:28	3,4204786	2	donderdag 18 januari 2018 22:25	9,609375	87,421875	89520
3	woensdag 17 januari 2018 14:28	3,654952	3	donderdag 18 januari 2018 22:26	9,671875	87,359375	89456
4	woensdag 17 januari 2018 14:30	4,0894708	4	donderdag 18 januari 2018 22:27	9,703125	93,5078125	95752
5	woensdag 17 januari 2018 14:31	3,2446411	5	donderdag 18 januari 2018 22:28	8,609375	87,1875	89280
6	woensdag 17 januari 2018 14:31	3,2504889	6	donderdag 18 januari 2018 22:29	7,859375	87,734375	89840
7	woensdag 17 januari 2018 14:31	3,3320149	7	donderdag 18 januari 2018 22:29	8,59375	86,96875	89056
8	woensdag 17 januari 2018 14:31	3,3899248	8	donderdag 18 januari 2018 22:30	8,5	87,3671875	89464
9	woensdag 17 januari 2018 14:33	3,4439619	9	donderdag 18 januari 2018 22:35	8,453125	86,91796875	89004
10	woensdag 17 januari 2018 14:38	3,3215672	10	donderdag 18 januari 2018 22:41	8,5	87,5703125	89672
11	woensdag 17 januari 2018 14:38	3,3687893	11	donderdag 18 januari 2018 22:41	8,609375	87,04296875	89132
12	woensdag 17 januari 2018 15:43	5,7444782	12	donderdag 18 januari 2018 22:49	54,984375	214,7578125	219912
13	woensdag 17 januari 2018 15:43	5,7872693	13	donderdag 18 januari 2018 22:52	46,59375	206,5	211456
14	woensdag 17 januari 2018 15:43	6,0293278	14	donderdag 18 januari 2018 22:58	47,171875	217,3125	222528
15	woensdag 17 januari 2018 15:43	6,1856812	15	donderdag 18 januari 2018 23:02	45,78125	211,7851563	216868
16	woensdag 17 januari 2018 15:44	5,7032764	16	donderdag 18 januari 2018 23:07	45,578125	212,4414063	217540
17	woensdag 17 januari 2018 15:44	5,9022279	17	donderdag 18 januari 2018 23:10	47,765625	211,7265625	216808
18	woensdag 17 januari 2018 15:44	5,9120525	18	donderdag 18 januari 2018 23:13	46,40625	217,7773438	223004
19	woensdag 17 januari 2018 15:44	5,9778533	19	donderdag 18 januari 2018 23:17	45,203125	211,2265625	216296
20	woensdag 17 januari 2018 15:44	6,001188	20	donderdag 18 januari 2018 23:26	47,15625	215,4570313	220628
21	woensdag 17 januari 2018 15:44	6,0388527	21	donderdag 18 januari 2018 23:30	47,890625	203,6992188	208588
22	woensdag 17 januari 2018 16:56	4,8610782	22	vrijdag 19 januari 2018 13:18	103,140625	295,671875	302768
23	woensdag 17 januari 2018 16:56	5,1452763	23	vrijdag 19 januari 2018 13:24	103,15625	296,1289063	303236
24	woensdag 17 januari 2018 16:56	5,1481666	24	vrijdag 19 januari 2018 13:31	103,03125	296,3515625	303464
25	woensdag 17 januari 2018 16:56	5,1938855	25	vrijdag 19 januari 2018 13:36	104,203125	295,8632813	302964
26	woensdag 17 januari 2018 16:56	5,2192978	26	vrijdag 19 januari 2018 13:53	105,484375	296,4140625	303528
27	woensdag 17 januari 2018 16:56	5,2568209	27	vrijdag 19 januari 2018 13:59	105,203125	294,9335938	302012
28	woensdag 17 januari 2018 16:56	5,7506035	28	vrijdag 19 januari 2018 14:04	108,078125	295,953125	303056
29	woensdag 17 januari 2018 16:57	4,7198479	29	vrijdag 19 januari 2018 14:17	112,53125	294,1953125	301256

H. Meetresultaten transformatie Perl

	A	B		A	B	C	D
1	Datum&tijd	Seconden	1	Datum&tijd	CPU	WS MB	WS kB
2	2018-01-19 215141	8,94497895240784	2	2018-01-19 215032	11,640625	159,2148438	163036
3	2018-01-19 215339	9,08106994628906	3	2018-01-19 215115	11,875	159,5351563	163364
4	2018-01-19 215236	9,14411211013794	4	2018-01-19 215141	11,046875	159,265625	163088
5	2018-01-19 215209	9,16112279891968	5	2018-01-19 215209	11,671875	158,9648438	162780
6	2018-01-19 215437	9,2121570110321	6	2018-01-19 215236	11,78125	159,2109375	163032
7	2018-01-19 215306	9,23116993904114	7	2018-01-19 215306	11,828125	159,2265625	163048
8	2018-01-19 215115	9,2476270198822	8	2018-01-19 215339	11,109375	159,0625	162880
9	2018-01-19 215412	9,32923698425293	9	2018-01-19 215412	11,5625	158,9257813	162740
10	2018-01-19 215032	9,37826800346375	10	2018-01-19 215437	11,640625	159,2070313	163028
11	2018-01-19 215512	13,1349720954895	11	2018-01-19 215512	16,53125	159,1328125	162952
12	2018-01-19 215615	16,0267128944397	12	2018-01-19 215615	18,328125	157,7148438	161500
13	2018-01-19 220842	16,6481280326843	13	2018-01-19 215706	26,53125	158,1210938	161916
14	2018-01-19 220654	17,2065010070801	14	2018-01-19 215813	25,6875	157,7773438	161564
15	2018-01-19 220112	20,282753944397	15	2018-01-19 215925	27,90625	157,7578125	161544
16	2018-01-19 220805	21,5747740268707	16	2018-01-19 220025	25,109375	157,5273438	161308
17	2018-01-19 215813	22,4038751125336	17	2018-01-19 220112	23,34375	158,0664063	161860
18	2018-01-19 220025	22,6492409706116	18	2018-01-19 220348	25,765625	157,8203125	161608
19	2018-01-19 220348	23,2638700008392	19	2018-01-19 220654	20,28125	157,6601563	161444
20	2018-01-19 215706	24,2083909511566	20	2018-01-19 220805	23,921875	158,0273438	161820
21	2018-01-19 215925	25,1085798740387	21	2018-01-19 220842	19,53125	157,828125	161616
22	2018-01-19 222716	25,2989089488983	22	2018-01-19 221541	33,09375	165,2109375	169176
23	2018-01-19 222804	25,3149211406708	23	2018-01-19 221648	29	165,2382813	169204
24	2018-01-19 222401	25,5450749397278	24	2018-01-19 221913	28,953125	165,3476563	169316
25	2018-01-19 222634	25,5480771064758	25	2018-01-19 222205	31,3125	164,9804688	168940
26	2018-01-19 221648	26,2125210762024	26	2018-01-19 222401	27,53125	165,0234375	168984
27	2018-01-19 221913	26,5607550144196	27	2018-01-19 222451	32,984375	165,3476563	169316
28	2018-01-19 222205	28,5346930027008	28	2018-01-19 222543	36,546875	165,3867188	169356
29	2018-01-19 221541	30,0476400852203	29	2018-01-19 222634	27,921875	165,3125	169280

I. Meetresultaten verplaatsing Perl

	A	B		A	B	C	D
1	Datum&tijd	Seconden	1	Datum&tijd	CPU	WS MB	WS kB
2	2018-01-19 234901	0,135884046554565	2	2018-01-19 234901	2,578125	51,375	52608
3	2018-01-20 114727	0,153788805007935	3	2018-01-20 113632	3,0625	51,41796875	52652
4	2018-01-20 113700	0,157106161117554	4	2018-01-20 113700	2,53125	51,53125	52768
5	2018-01-20 113758	0,159923076629639	5	2018-01-20 113728	3,125	51,5234375	52760
6	2018-01-20 113728	0,163134098052979	6	2018-01-20 113758	2,953125	51,37109375	52604
7	2018-01-20 113632	0,164635181427002	7	2018-01-20 113828	2,703125	51,4296875	52664
8	2018-01-20 113828	0,167891025543213	8	2018-01-20 114002	2,90625	51,4296875	52664
9	2018-01-20 114619	0,174349069595337	9	2018-01-20 114049	4,359375	51,6171875	52856
10	2018-01-20 114002	0,182392120361328	10	2018-01-20 114619	2,5	51,5078125	52744
11	2018-01-20 114049	0,269557952880859	11	2018-01-20 114727	2,25	51,54296875	52780
12	2018-01-20 115955	1,42795205116272	12	2018-01-20 115206	3,875	51,39453125	52628
13	2018-01-20 115855	1,47198390960693	13	2018-01-20 115320	3,8125	51,57421875	52812
14	2018-01-20 115206	1,60106992721558	14	2018-01-20 115433	4,546875	51,33203125	52564
15	2018-01-20 115758	1,60807609558105	15	2018-01-20 115531	4,25	51,55078125	52788
16	2018-01-20 115320	1,61007690429688	16	2018-01-20 115709	4,515625	51,953125	53200
17	2018-01-20 115433	1,77618598937988	17	2018-01-20 115758	4,125	51,578125	52816
18	2018-01-20 120210	1,80320382118225	18	2018-01-20 115855	3,71875	51,515625	52752
19	2018-01-20 115531	1,82021689414978	19	2018-01-20 115955	3,578125	51,82421875	53068
20	2018-01-20 115709	1,82622194290161	20	2018-01-20 120109	4,875	51,82421875	53068
21	2018-01-20 120109	2,03435897827148	21	2018-01-20 120210	4,359375	51,81640625	53060
22	2018-01-20 141144	14,7588810920715	22	2018-01-20 140041	17,734375	53,8359375	55128
23	2018-01-20 141313	14,7918870449066	23	2018-01-20 140305	17,78125	53,6640625	54952
24	2018-01-20 141613	14,9189629554749	24	2018-01-20 140536	17,546875	53,62109375	54908
25	2018-01-20 141443	14,9511168003082	25	2018-01-20 140754	16,484375	54,6484375	55960
26	2018-01-20 140754	14,9730081558228	26	2018-01-20 141015	16,9375	53,83984375	55132
27	2018-01-20 141015	15,1221070289612	27	2018-01-20 141144	16,515625	53,70703125	54996
28	2018-01-20 140305	15,7265119552612	28	2018-01-20 141313	16,453125	53,49609375	54780
29	2018-01-20 140536	15,8205749988556	29	2018-01-20 141443	16,6875	53,5390625	54824

J. Meetresultaten samenvoeging Perl

	A	B		A	B	C	D
1	Datum&tijd	Seconden	1	Datum&tijd	CPU	WS MB	WS kB
2	2018-01-20 215828	0,555432081222534	2	2018-01-20 215535	2,21875	51,55078125	52788
3	2018-01-20 215744	0,56036901473999	3	2018-01-20 215555	2,75	51,44921875	52684
4	2018-01-20 215555	0,560376882553101	4	2018-01-20 215611	2,40625	51,59375	52832
5	2018-01-20 215709	0,562373876571655	5	2018-01-20 215637	2,4375	51,390625	52624
6	2018-01-20 215725	0,562376976013184	6	2018-01-20 215653	2,640625	51,56640625	52804
7	2018-01-20 215812	0,563374996185303	7	2018-01-20 215709	2,484375	51,53125	52768
8	2018-01-20 215653	0,564377069473267	8	2018-01-20 215725	2,484375	51,65625	52896
9	2018-01-20 215611	0,565379858016968	9	2018-01-20 215744	2,65625	51,359375	52592
10	2018-01-20 215637	0,566377878189087	10	2018-01-20 215812	2,46875	51,26953125	52500
11	2018-01-20 215535	0,576382875442505	11	2018-01-20 215828	2,65625	51,61328125	52852
12	2018-01-20 215957	3,5574688911438	12	2018-01-20 215938	5,296875	52,01953125	53268
13	2018-01-20 220015	3,57338714599609	13	2018-01-20 215957	5,359375	51,671875	52912
14	2018-01-20 220034	3,58539700508118	14	2018-01-20 220015	5,625	51,69921875	52940
15	2018-01-20 220349	3,59140086174011	15	2018-01-20 220034	5,3125	51,765625	53008
16	2018-01-20 220328	3,59540486335754	16	2018-01-20 220052	5,84375	51,8046875	53048
17	2018-01-20 220409	3,60340905189514	17	2018-01-20 220227	5,890625	51,8515625	53096
18	2018-01-20 220227	3,61890411376953	18	2018-01-20 220310	5,796875	51,58984375	52828
19	2018-01-20 215938	3,63542890548706	19	2018-01-20 220328	5,5625	51,984375	53232
20	2018-01-20 220310	3,67645502090454	20	2018-01-20 220349	5,5625	51,703125	52944
21	2018-01-20 220052	3,73568201065063	21	2018-01-20 220409	5,859375	51,8359375	53080
22	2018-01-20 221852	6,24717712402344	22	2018-01-20 220638	8,59375	52,7421875	54008
23	2018-01-20 221830	6,25818419456482	23	2018-01-20 221336	8,140625	52,5546875	53816
24	2018-01-20 221919	6,26418805122375	24	2018-01-20 221503	8,3125	52,46875	53728
25	2018-01-20 221555	6,28120112419128	25	2018-01-20 221533	8,21875	52,4765625	53736
26	2018-01-20 221808	6,28220105171204	26	2018-01-20 221555	8,28125	52,6171875	53880
27	2018-01-20 221533	6,29020404815674	27	2018-01-20 221808	7,765625	52,38671875	53644
28	2018-01-20 221503	6,29320812225342	28	2018-01-20 221830	8,28125	52,578125	53840
29	2018-01-20 221946	6,29820799827576	29	2018-01-20 221852	7,859375	52,5703125	53832

