



Solution Design | STARLight 2.0

Versie 1.0

Datum: 08/09/2023
Status: Goedgekeurd

Documenthistorie

Datum	Versie	Beschrijving	Auteur
26/06/2023	0.1	Initiële versie	Tom van Poppelen Protomic
09/07/2023	0.2	Update openstaande punten en aanvulling PvA en planning	Tom van Poppelen Protomic Leon van Lierop Ster
14/07/2023	0.3	Verwerken feedback 1	Tom van Poppelen Protomic
08/09/2023	0.4	Verwerken feedback 2	Tom van Poppelen Protomic
18/09/2023	1.0	Goedgekeurd	Leon van Lierop & Robbert van den Hoogen

Inhoudsopgave

1.	Inleiding	4
1.1	Aanleiding	4
1.2	Doel van dit document	4
1.3	Referenties	4
2.	Begrippen	5
3.	Huidige situatie (IST)	7
3.1	Inleiding	7
3.2	STARLight Database	8
3.3	STARLight Windows Applicatie	9
3.4	Interfaces	10
3.4.1	SOAP vs RESTful services	10
3.4.2	STARLight API	11
3.4.3	STARLight AFAS API	11
3.4.4	STARLight CRM API	11
3.4.5	STARLight B2B API	11
3.5	STARLight Batch Scheduler	12
3.6	NPO WhatsOn	12
3.6.1	NPO WhatsOn Business API	12
3.7	NPO Media Gateway	12
3.7.1	NPO TV Playlists – Miranda API	12
3.7.2	NPO Radio Playlists – OmniRadio API	12
3.7.3	NPO Mediabestanden – Signiant/FTP	12
3.8	Digital	13
3.8.1	DAPI	13
3.9	Klantportaal	13
3.9.1	Klantportaal API	14
3.10	Infrastructuur	14
3.11	PDF Generatie - Crystal Reports	14
4.	Gewenste situatie (SOLL)	15
4.1	Inleiding	15
4.2	STARLight 2.0 Database	16
4.3	STARLight 2.0 Web Applicatie	16
4.3.1	React/TypeScript vs Blazor	16
4.3.2	UI Web Componenten Bibliotheek	17
4.4	Interfaces	18
4.4.1	STARLight API	18
4.4.2	STARLight AFAS API	18
4.4.3	STARLight CRM API	18
4.4.4	STARLight B2B API	18
4.5	STARLight Batch Scheduler	19
4.5.1	Azure Web Jobs	19
4.5.2	Hangfire	19
4.6	Digital	20
4.7	Klantportaal	20
4.8	Afsplitsing in separate modules en/of microservices	20

4.8.1	Microservices	21
4.9	Infrastructuur	21
4.9.1	Azure App Services	21
4.9.2	Azure SQL Services	21
4.9.3	Azure Blob Storage	21
4.10	PDF Generatie – Aspose.PDF	22
5.	Migratie (IST naar SOLL)	23
5.1	Inleiding	23
5.2	Aandachtspunten	24
5.2.1	Digitale module	24
5.2.2	Kennisoverdracht STARLight business logic	24
6.	Concept Solution Design	25
6.1	High Level Design	25
6.2	Azure Services	25
6.2.1	Azure App Services	25
6.2.2	Azure SQL Services	25
6.2.3	Azure Blob Storage	25
6.2.4	Azure API Management Application Gateway	26
6.3	Privacy en veiligheid	26
6.4	Authenticatie en autorisatie	26
6.5	Beschikbaarheid	27
6.6	Schaalbaarheid	27
6.7	Monitoring/Logging	27
6.8	OTAP/CI/CD	27

1. Inleiding

1.1 Aanleiding

Het huidige Ad Sales systeem van Ster 'STARLight' is gebaseerd op enigszins verouderde Microsoft technologie, bevat 3^e partij (legacy) softwarecomponenten en is na 20+ jaar doorontwikkeling technisch inhoudelijk op meerdere onderdelen aan een herziening toe. Om de doorontwikkeling en het beheer van STARLight in de toekomst te kunnen waarborgen is een technologische vernieuwing van het systeem gewenst.

1.2 Doel van dit document

Dit document beschrijft een mogelijk inzetbare toekomstbestendige doelarchitectuur en een conceptueel solution design t.b.v. vernieuwing van het Ad Sales systeem naar versie STARLight 2.0 gebaseerd op up-to-date Microsoft (Azure Cloud) technologie.

1.3 Referenties

Dit document is mede tot stand gekomen op basis van onderstaande informatiebronnen gerelateerd aan het bestaande en gewenste IT en applicatielandschap van Ster m.b.t. STARLight:

- Aanbestedingsstukken vernieuwing STARLight
- Beschikbare functionele en technische documentatie van de huidige versie STARLight (1.0)
- Inhoudelijke gesprekken en interviews met (IT) medewerkers van Ster.
 - Robbert van den Hoogen | IT Manager
 - Leon van Lierop | Projectleider IT
 - Willem de Jonge | Applicatie Architect | STARLight
 - Herman Oskam | Applicatie Architect | STARLight
 - Jelle de Jong | Applicatie Ontwikkelaar | Klantportaal
 - Vincent Osinga | Applicatie Ontwikkelaar | Klantportaal
 - Senta Gerwig | Functioneel beheerder CRM | Digital
 - Floris Brouwer | Webdeveloper | Digital | DAPI
 - Marc Punte | IT Beheerder | Ster Netwerk | Azure
 - Tom Zwart | Developer | IT Beheerder | Ster Netwerk | Azure
- STARLight en Klantportaal broncode in Ster Azure DevOps Repositories

2. Begrippen

Azure

Azure is het Cloudplatform van Microsoft, waarbinnen alle Microsoft Cloudservices via een abonnementsvorm beschikbaar worden gesteld.

API

API staat voor Application Programming Interface en is een verzameling van definities op basis waarvan een applicatie of service kan communiceren met een andere applicatie of service.

C#.Net

C#.Net is de primaire ontwikkeltaal op het Microsofts .Net ontwikkelplatform. De oorsprong van C# ligt in een combinatie van het beste van diverse (oudere) ontwikkeltalen, zoals Java, C++ en Pascal.

CI/CD

CI/CD staat voor Continuous Integration en Continuous Delivery/Deployment. Dit is een modern softwareontwikkelconcept, waarmee software updates geautomatiseerd een ontwikkelstraat doorlopen en uitgerold kan worden op (klant)omgevingen.

DBMS

DBMS staat voor Database Management System en is software waarmee gebruikers database gegevens kunnen opslaan, ophalen en bevragen.

DMZ

DMZ staat voor Demilitarized Zone en is een speciaal beperkt afgeschermd stukje netwerk wat naar de buitenwereld staat opengesteld en een bescherming biedt voor verdere toegang naar het interne (klant)netwerk.

Entity Framework

Microsoft .Net Entity Framework is een gestandaardiseerde en geoptimaliseerde data access laag, waarmee CRUD (Create, Read, Update, Delete) operaties op een relationele database, via een object model (geautomatiseerd) beschikbaar wordt gesteld aan een ontwikkelaar. Een zogenaamd Object-Relation-Mapper (ORM) framework. Hiermee krijgt de ontwikkelaar een hoger abstractieniveau tot zijn beschikking, om database operaties mee uit te voeren, en behoeven deze niet zelf uit geprogrammeerd.

Foreign key en constraints

Een foreign key is een unieke sleutel, waarmee relatie(s) tussen database tabellen worden gedefinieerd. In combinatie met optionele constraints (beperkingen) kan op relaties worden ingesteld hoe gerelateerde records tussen tabellen moeten reageren bij een update of verwijder actie.

.Net Framework

.Net Framework is het (verouderde) Microsoft ontwikkel framework. Met dit framework kunnen alleen applicaties worden ontwikkeld voor het Microsoft Windows besturingssysteem. Het .Net Framework wordt nog wel ondersteund, maar niet meer doorontwikkeld.

.Net (Core)

.Net (Core) is het huidige Microsoft ontwikkel-framework. Dit framework is Cross-Platform en hiermee kunnen applicaties worden ontwikkeld voor zowel het Microsoft Windows, Linux en MacOS besturingssystemen.

OTAP

OTAP is de afkorting voor Ontwikkel, Test, Acceptatie en Productie en wordt toegepast om de fases aan te geven die een applicatie doorloopt in een ontwikkelstraat.

Primary key

De primary key is de primaire unieke sleutel, waarmee records in een database tabel kunnen worden geïdentificeerd.

Relationele database

Een relationele database is een verzameling van tabellen met onderlinge relaties welke via foreign keys en constraints zijn gemodelleerd.

React

React is een Javascript bibliotheek ontwikkeld door Facebook, waarmee een op componenten gebaseerde, interactieve gebruikersinterface (UI) kan worden gerealiseerd binnen een Web Applicatie.

RESTful Service

Een RESTful Service is een API gebaseerd op het HTTP Protocol (GET/POST/PUT/DELETE), waarmee berichten in een JSON-format worden uitgewisseld.

SOAP Service

Een SOAP Service is een API gebaseerd op het SOAP Protocol, waarmee berichten in een XML-format worden uitgewisseld.

SQL Server

SQL Server is Microsoft Database Management Systeem (DBMS) wat draait op een Microsoft Windows Server.

Typescript

Typescript is een programmeertaal gebaseerd op Javascript. Typescript compileert naar Javascript. Het biedt de ontwikkelaar sterk getypeerde functionaliteiten in de code-editor aan, waardoor fouten sneller ondervangen worden en robuuster geprogrammeerd kan worden.

VB.Net

VB.Net is een verouderde ontwikkeltaal op het Microsoft .Net ontwikkelplatform. De oorsprong ligt in het oude Visual Basic.

Web Applicatie

Een webapplicatie is een applicatie die via een webbrowser kan worden geopend en gebruikt.

Windows Forms Applicatie

Een Windows Forms Applicatie is een applicatie, die alleen geïnstalleerd en gebruikt kan worden op een Microsoft Windows besturingssysteem.

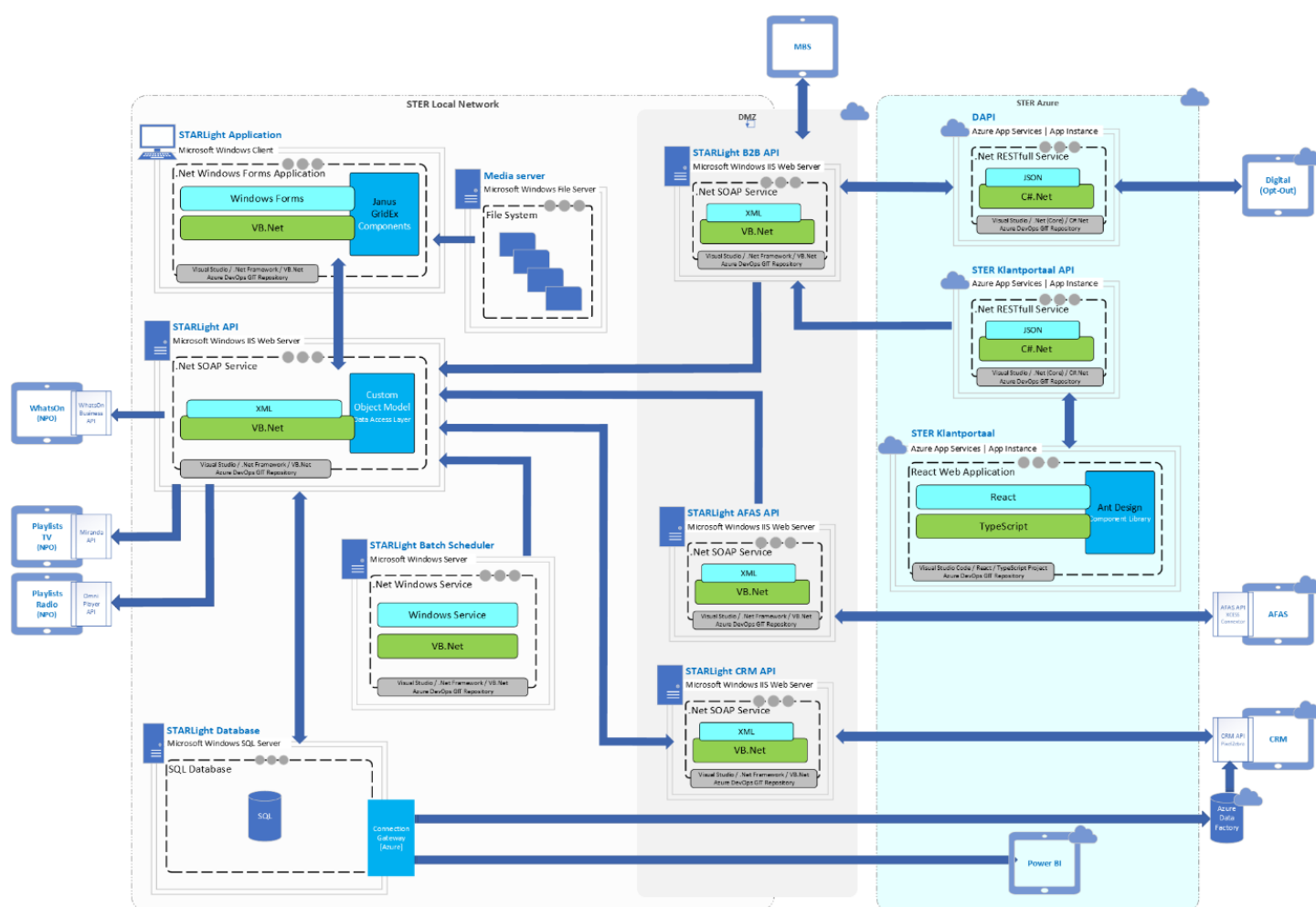
3. Huidige situatie (IST)

3.1 Inleiding

In algemene zin wordt het huidige STARLight systeem als goed werkend ervaren. Het doet wat het moet doen en biedt in de kern alle benodigde functionaliteiten, schermen, views, configuratie mogelijkheden, en interfaces boven op de STARLight database. Ster medewerkers kunnen dagelijks hun Ad Sales werk goed uitvoeren via de STARLight Windows applicatie en zowel interne als externe partijen en applicaties communiceren (vaak real-time) met STARLight via de diverse interne en externe interfaces.

Wat speelt is dat het huidige STARLight systeem een doorontwikkeling is geweest op basis van technologische keuzes van 10+ jaar geleden. Diverse toegepaste technieken, componenten en methodiek(en) zijn inmiddels verouderd. Om gebruik en doorontwikkeling van STARLight in de toekomst te kunnen waarborgen is een technologische vernieuwing gewenst.

De IST-situatie beschrijft het huidige technologische landschap van STARLight.



3.2 STARLight Database

De huidige STARLight database structuur bestaat uit één (1) enkele database met ongeveer 300 tabellen. Die database is van oudsher relationeel gemodelleerd. Het model is functioneel kloppend, waar het gaat om de aanwezige tabellen en relaties tussen die tabellen. Op het databasemodel behoeft in de basis in principe geen wijziging te worden doorgevoerd, dit met de kanttekening dat inmiddels gewijzigde, niet meer gebruikte en/of verwijderde functionaliteiten binnen het STARLight systeem ertoe geleid heeft dat tabellen of tabeldefinities herzien, of misschien zelfs verwijderd kunnen worden.

Sinds de eerste ontwikkelingen van STARLight (30+ jaar geleden) is de database een aantal keer overgegaan van databasemanagementsysteem (DBMS) en door de jaren heen ook uitgebreid. Momenteel draait de STARLight database op basis van DBMS Microsoft SQL Server. Microsoft SQL Server is een up-to-date DBMS en behoeft in die zin niet vervangen te worden door een ander DBMS. SQL Server wordt in de huidige situatie op het lokale Ster netwerk gehost en beheerd.

Een punt van aandacht, met betrekking tot de technische uitwerking van het huidige relationele databasemodel binnen SQL Server, is de afwezigheid van fysieke relaties in de vorm van Foreign Keys en eventueel bijbehorende Constraints. De afwezigheid hiervan ligt met name in de geschiedenis van de ontwikkeling van de database en de diverse (oude) DBMS'en waarop de database heeft gedraaid. In eerdere versies van SQL Server was het o.a. vanwege performance redenen, waardoor fysieke relaties zijn vermeden. In de loop der jaren is SQL Server sterk verbeterd, maar is dit nooit herzien. Het aanbrengen van fysieke relaties heeft te allen tijde de voorkeur om de data integriteit op database niveau correct geregeld te hebben. In de huidige implementatie van STARLight is de relatie logica tussen tabellen primair uitgewerkt in de data access code laag.

Een ander punt van aandacht binnen het databasemodel is de implementatie en generatie van primaire (unieke) sleutelwaarden binnen tabellen. Ook hier is vanuit het verleden een mechanisme ontwikkeld vanwege het gebrek aan (goede) functionaliteit binnen oudere DBMS'en.

In een technische vernieuwing van de database heeft het grote voorkeur om de implementatie van Primaire en Foreign keys te herzien en vooral de kracht te gebruiken van een modern DBMS, waarvan men mag verwachten dat deze inmiddels geoptimaliseerd zijn op deze punten. Tijdens een herziening kan eveneens gekeken worden naar (meer) inzet van SQL Stored Procedures en Views om performance (extra) te optimaliseren.

3.3 STARLight Windows Applicatie

De huidige STARLight applicatie is een Microsoft Windows Forms Applicatie gebaseerd op basis van Microsoft .Net Framework technologie. De applicatie wordt alleen beschikbaar gesteld op Microsoft Windows werkstations van Ster medewerkers (laptops met Windows10/11) en functioneert alleen binnen het Ster Netwerk. Een medewerker logt met zijn eigen Ster netwerkaccount in op de applicatie en krijgt vervolgens op basis van zijn functie/rol/rechten functionaliteiten beschikbaar om zijn/haar werkzaamheden uit te voeren.

Onderstaand een overzicht van beschikbare functionaliteiten binnen de STARLight Applicatie:

- Gebruikersbeheer (functies/rollen/rechten)
- Beheer van scheduled batched/queued verwerkingsprocessen
- Beheer stamtabellen (Doelgroepen, Zendercombinaties, Kortingstypen, etc)
- Materiaal
- Verkoop
- Programmering
- Klanten
- Facturatie
- Rapportages
- Filter en zoekopties

De applicatie is gebaseerd op het Microsoft .Net Framework en programmeertaal VB.Net. Het Microsoft Visual Studio project is van het type Windows Forms applicatie, waarbinnen gebruik wordt gemaakt van een 3^e partij componenten bibliotheek Janus Winforms/GridEx. Het Janus Winforms/GridEx control vormt de UI basis voor alle schermen binnen de STARLight applicatie, maar is een sterk verouderde bibliotheek waar geen updates meer op plaatsvinden sinds 2016. Dit is een van de hoofdredenen om de STARLight applicatie tot een vernieuwing te brengen.

Andere overwegingen om de applicatie tot een vernieuwing te brengen:

- Het gebruik van de programmeertaal VB.Net. Hiervan is de support binnen de nieuwste Microsoft .Net development tools niet meer standaard aanwezig. Een vernieuwing naar C#.Net, de primaire Microsoft .Net ontwikkeltaal, zal bij vernieuwing een logische stap zijn om meer toekomstbestendig te kunnen zijn.
- De afhankelijkheid van het Microsoft Windows OS bij het type .Net Framework Windows Forms Applicatie. Door te vernieuwen naar een type Web Applicatie ontstaat meer flexibiliteit in het OS of device, waarmee medewerkers met de applicatie kunnen werken. Het enige wat dan benodigd is, is een supported webbrowser. Web technologie is tevens waar Microsoft met de recentste .Net (core) technologie en Azure cloud platform de focus heeft liggen en toekomstbestendigheid biedt.

De applicatie communiceert niet rechtstreeks met de STARLight database. Data access van en naar de database verloopt via de STARLight API. Via de API worden de benodigde datasets t.b.v. het Janus GridEx control opgehaald en ook het wegschrijven van data vanuit de applicatie naar de database geregeld. In principe zit alle business logica verwerkt in de STARLight API en onderkent de oplossing het basis ontwikkelprincipe van scheiding van verantwoordelijkheden (Separation of Concerns, SoC), deze opzet kan daarom ook meegenomen worden bij een vernieuwing.

3.4 Interfaces

Het STARLight systeem kent één hoofd-interface, de STARLight (interne) API, deze bevat in feite alle STARLight business logic en regelt primair de communicatie van en naar de STARLight database. Er zijn in principe geen applicaties en/of andere interfaces die rechtstreeks met de STARLight database communiceren. Er wordt in principe altijd gebruik gemaakt van de STARLight API, zodat centrale business logic ervoor zorgt dat gegevens op een consistente wijze worden uitgelezen en weggeschreven naar de STARLight database.

Ten behoeve van externe systemen, die van buiten het netwerk van de Ster moeten kunnen communiceren met STARLight zijn aparte interfaces ontwikkeld. De interne STARLight API is afgeschermd en niet rechtstreeks benaderbaar van buitenaf. We onderkennen drie interfaces, die via de DMZ-zone van het Ster netwerk voor systemen en services van buitenaf beschikbaar zijn gesteld:

- STARLight B2B API: Interface met specifieke STARLight API end-points beschikbaar gesteld voor communicatie vanaf:
 - o MBS – Mediabureaus
 - o Digital (Opt-out) service (via de DAPI)
 - o Ster Klantportaal (via de Klantportaal API)
- STARLight AFAS API: Interface met specifieke STARLight API end-points beschikbaar gesteld voor communicatie vanaf AFAS, via de (door XCESS ontwikkelde) AFAS Connextor API.
- STARLight CRM API: Interface met specifieke STARLight API end-points beschikbaar gesteld voor communicatie vanaf CRM, via de (door PixelZebra ontwikkelde) CRM API.

Er zijn twee uitzonderingen op het gebruik van de STARLight API t.b.v. het ophalen van datasets:

- Power BI koppeling: Rapportages in Power BI maken gebruik van rechtstreekse koppelingen met de STARLight database (alleen lezen).
- CRM-koppeling: Vanuit CRM wordt dagelijks via een rechtstreekse koppeling met de STARLight database een View uitgelezen t.b.v. synchronisatie van klant gerelateerd gegevens vanuit de STARLight database naar CRM (alleen lezen).

Deze opzet (en scheiding) van interfaces kan bij vernieuwing in principe aangehouden worden.

In de volgende paragrafen worden de toegepaste technieken van de interfaces verder beschreven.

3.4.1 SOAP vs RESTful services

Het merendeel van de STARLight interfaces is gerealiseerd op basis van het SOAP protocol. SOAP is een ouder protocol, maar wordt nog wel ondersteund, request en response zijn gebaseerd op (ingepakte) XML-berichten.

De huidige standaard voor Web API's zijn interfaces gebaseerd op het RESTful services principe. Bij vernieuwing van API's gaat de voorkeur uit naar het beschikbaar stellen van de API's in de vorm van RESTful services. RESTful services is een op HTTP Protocol (GET/POST/PUT/DELETE) gebaseerde communicatiemechanisme met request en response gebaseerd op JSON-berichten. Het grote voordeel van een RESTful service is het gebruik van minder bandbreedte en daarmee de snelheid van berichtuitwisseling. Berichten zijn vele malen kleiner vergeleken met het SOAP Protocol.

Een punt van aandacht bij vernieuwing van interfaces is het (tijdelijk) beschikbaar houden van SOAP services. Bijvoorbeeld vanwege bestaande koppelingen van externe afnemers, die niet zomaar hun applicatie kunnen ombouwen naar RESTful communicatie. Bij interfaces waarbij SOAP beschikbaar moet blijven houdt vernieuwing in dat de API in eerste instantie SOAP blijft, maar wel een vernieuwing ondergaat, zoals het inzetten van C#.Net als programmeertaal. Denk hierbij bijvoorbeeld aan de STARLight B2B API t.b.v. de externe mediabureaus.

3.4.2 *STARLight API*

De STARLight (interne) API is een SOAP Web Service, gebaseerd op het Microsoft .Net Framework en programmeertaal VB.Net. Binnen de API is zowel STARLight business logic als de fysieke database communicatie uitgewerkt. De API regelt alle database bewerkingen (CRUD), maar handelt ook de database integriteit af, deze data access laag is in de huidige oplossing volledig zelf uitgeprogrammeerd en een doorontwikkeling geweest sinds het ontstaan van STARLight in de jaren 90.

Een vernieuwing van deze API is o.a. gewenst op het gebruik van VB.Net als programmeertaal, het beschikbaar stellen als RESTful service en het vervangen van de custom data access laag door een data access laag gebaseerd op het gestandaardiseerde Entity Framework, waarmee de ontwikkelaar een hoger niveau van abstractie tot zijn beschikking krijgt om database acties uit te voeren en deze niet zelf hoeft uit te programmeren.

3.4.3 *STARLight AFAS API*

De STARLight AFAS API is een SOAP Web Service, gebaseerd op het Microsoft .Net Framework en programmeertaal VB.Net. Via de API worden een aantal specifieke end-points van de STARLight (internal) API ontsloten, waarmee financiële informatie tussen STARLight en AFAS kan worden uitgewisseld, zoals gefactureerde en betaalde facturen.

De STARLight AFAS API wordt fysiek gehost in een afgeschermd DMZ op het Ster netwerk.

Een vernieuwing van deze API is o.a. gewenst op het gebruik van VB.Net als programmeertaal en het beschikbaar stellen als RESTful service.

3.4.4 *STARLight CRM API*

De STARLight CRM API is een SOAP Web Service, gebaseerd op het Microsoft .Net Framework en programmeertaal VB.Net. Via de API worden een aantal specifieke end-points van de STARLight (internal) API ontsloten, waarmee informatie tussen STARLight en CRM kan worden uitgewisseld, zoals klantcontactgegevens.

De STARLight CRM API wordt fysiek gehost in een afgeschermd DMZ op het Ster netwerk.

Een vernieuwing van deze API is o.a. gewenst op het gebruik van VB.Net als programmeertaal en het beschikbaar stellen als RESTful service.

3.4.5 *STARLight B2B API*

De STARLight B2B API is een SOAP Web Service, gebaseerd op het Microsoft .Net Framework en programmeertaal VB.Net. Via de API worden een aantal specifieke end-points van de STARLight (internal) API ontsloten, waarmee informatie tussen STARLight en externe mediabureaus (MBS), externen cloud service Digital (via de DAPI) en het STARLight Klantportaal kan worden uitgewisseld.

De B2B API wordt fysiek gehost in een afgeschermd DMZ op het Ster netwerk.

Een vernieuwing van deze API is gewenst op o.a. het gebruik van VB.Net als programmeertaal, het beschikbaar stellen als RESTful service.

Een aandachtspunt bij de B2B API is het beschikbaar houden van een SOAP interface voor de externe mediabureaus.

3.5 STARLight Batch Scheduler

De STARLight Batch Scheduler is een Windows Service, gebaseerd op het Microsoft .Net Framework en programmeertaal VB.Net. Via de STARLight applicatie kunnen jobs worden geconfigureerd en gescheduled. De job engine is door Ster zelf ontwikkeld en bevat o.a. time-schedules en logica t.b.v. afhankelijkheid en volgorde van jobs. Er zijn o.a. (nachtelijke) batch jobs voor het inroosteren en inplannen van TV- en Radiocommercials.

Een vernieuwing van de Batch Scheduler is o.a. gewenst op het type applicatie .Net Framework Windows Service en het gebruik van VB.Net als programmeertaal.

3.6 NPO WhatsOn

NPO WhatsOn is het NPO Broadcast Management system voor TV en Radiocommercials. De STARLight Batch Scheduler draait daily jobs, waarmee TV en Radioblokkenschema's worden bepaald voor de NPO. Gegevensuitwisseling tussen STARLight en WhatsOn vindt plaats via de WhatsOn Business API. STARLight haalt op gezette tijden benodigde schedule informatie op uit WhatsOn en schrijft blokkenschema's terug. De communicatie wordt altijd geïnitieerd vanuit STARLight, WhatsOn initieert geen communicatie richting STARLight.

3.6.1 NPO WhatsOn Business API

De WhatsOn Business API is een SOAP Service en in beheer van NPO. STARLight spreekt deze SOAP Service aan om te communiceren met WhatsOn.

Op het moment van schrijven van dit document, wordt gewerkt aan een vernieuwing van de WhatsOn Business API naar een RESTful service. De communicatie vanuit STARLight naar WhatsOn zal in de toekomst omgezet moeten gaan worden van SOAP naar REST.

3.7 NPO Media Gateway

Via de NPO Media Gateway interface worden TV- en Radiocommercial playlists en fysieke mediabestanden (commercials) vanuit de STARLight omgeving naar de NPO verstuurd. De STARLight Batch Scheduler draait o.a. een daily job, waarmee TV en Radio playlists worden samengesteld en verstuurd. Via de STARLight Applicatie kunnen medewerkers van Ster ook real-time (on-demand) commercial playlist informatie ophalen en versturen. De commercial playlist gegevensuitwisseling tussen STARLight en NPO vindt plaats via de Media Gateway. Voor TV gaat dat via de Miranda API en voor Radio via de OmniRadio API. Overdracht van de mediabestanden gaat via Signiant/FTP. Communicatie wordt altijd geïnitieerd vanuit STARLight, NPO initieert geen communicatie richting STARLight.

3.7.1 NPO TV Playlists – Miranda API

De Miranda API is een SOAP Service en in beheer van NPO. STARLight spreekt deze SOAP Service aan om de tv-commercial playlist gegevens heen en weer te sturen.

3.7.2 NPO Radio Playlists – OmniRadio API

De Miranda API is een RESTful Service en in beheer van NPO. STARLight spreekt deze RESTful Service aan om de Radiocommercial playlist gegevens heen en weer te sturen.

3.7.3 NPO Mediabestanden – Signiant/FTP

Signiant/FTP (File Transfer) software wordt ingezet om de fysieke commercial mediabestanden van het Ster netwerk over te zetten naar de NPO.

3.8 Digital

Digital is een externe (cloud) service, welke door Ster wordt afgenomen bij Opt-Out Advertising. Deze service voorziet Ster, naast hun eigen TV en Radio advertising functionaliteiten in het kunnen aanbieden van cookieloos digitaal adverteren op het internet. De Digital (Opt-Out) service is een compleet pakket inclusief beheer van digitale campagnes, hosting (streaming) en het meten van de digitale advertenties.

In STARLight is (beperkte) functionaliteit ingebouwd om digitale campagnes in het externe Digital systeem te kunnen beheren/bekijken. Om gegevens van en naar Digital te communiceren heeft Ster een eigen interface gebouwd en heeft deze ook in eigen beheer, de DAPI (Digital API). Via de DAPI worden een aantal specifieke end-points van de STARLight (internal) API ontsloten, waarmee informatie tussen STARLight en Digital kan worden uitgewisseld, zoals digitale campagne gegevens.

Ster heeft de wens uitgesproken om de digital campagne functionaliteiten, welke nu geboden worden door de Digital (Opt-Out) service, zelf te gaan realiseren en op te nemen binnen het STARLight systeem. Op dit moment is het nog niet helder om welke functionaliteiten het precies gaat. Is het idee om alle Digital functionaliteiten in eigen beheer te nemen inclusief de hosting, streaming en metingen van de digitale advertenties, dan zal hier waarschijnlijk een kostenoverweging aan ten grondslag liggen. De vraag is of men hiermee inderdaad 'winst' kan behalen en zal door Ster nader bepaald moeten worden. Realisatie van een eigen 'Digital' valt buiten de scope van dit solution design.

3.8.1 DAPI

De DAPI is ontwikkeld als een RESTful Service, gebaseerd op recente Microsoft .Net development technologie. Hosting vindt plaats in de Azure Cloud, binnen het Azure App Services landschap. Via de DAPI worden een aantal specifieke end-points van de STARLight (internal) API ontsloten, waarmee informatie tussen STARLight en Digital kan worden uitgewisseld, zoals digitale campagne gegevens.

Development vindt plaats in Visual Studio en C#.Net is de ontwikkeltaal. Versiebeheer van broncode vindt plaats via Azure DevOps repositories en deployment van nieuwe versies wordt gedaan via Azure DevOps pipelines.

Gezien de huidige development stack van de DAPI en de hosting in Azure is deze (voorlopig) niet onderhevig aan een vernieuwingsslag en past deze in de beoogde nieuwe doelarchitectuur.

3.9 Klantportaal

Het Ster klantportaal is een moderne webapplicatie gebaseerd op up-to-date Microsoft .Net web development technologie. De applicatie is via het internet publiekelijk toegankelijk op adres <https://portal.ster.nl>. Om te kunnen werken in het portaal heeft de gebruiker een login account nodig. Zowel externe klanten als interne medewerkers van Ster kunnen inloggen op het klantportaal.

Het klantportaal biedt de klant functionaliteiten om zijn eigen reclamecampagnes in te zien en te beheren. Het klantportaal is een moderne web interface boven op het STARLight systeem. Gegevens met betrekking tot de klant reclamecampagnes worden via de STARLight B2B API uitgelezen of weggeschreven. Communicatie tussen Web Applicatie en de STARLight B2B API vindt plaats via een eigen Klantportal API.

De webapplicatie is gebaseerd op React met TypeScript en gebruikt de Ant Design componenten bibliotheek voor UI elementen en look and feel. Development vindt plaats in Visual Studio (Code). Versiebeheer van broncode vindt plaats via Azure DevOps repositories en deployment van nieuwe versies wordt gedaan via Azure DevOps pipelines.

Het klantportaal heeft ook een eigen SQL-database voor Klantportaal specifieke informatie, welke niet wordt opgeslagen in de STARLight database. Communicatie met de Klantportaal database vindt plaats via de Klantportaal API.

Hosting vindt plaats in de Azure Cloud, binnen het Azure App Services landschap. Voor de Klantportaal database wordt gebruik gemaakt van Azure SQL Services.

Gezien de huidige development stack van het Klantportaal en de hosting in Azure is deze (voorlopig) niet onderhevig aan een vernieuwingsslag en past deze in de beoogde nieuwe doelarchitectuur.

3.9.1 Klantportaal API

De Klantportaal API is ontwikkeld als een RESTful Service, gebaseerd op recente Microsoft .Net development technologie. Hosting vindt plaats in de Azure Cloud, binnen het Azure App Services landschap. Via de Klantportal API worden een aantal specifieke end-points van de STARLight (internal) API ontsloten, waarmee informatie tussen het Klantportal en STARLight worden uitgewisseld, zoals reclamecampagne gegevens en inzage van facturen.

Development vindt plaats in Visual Studio en C#.Net is de ontwikkeltaal. Versiebeheer van broncode vindt plaats via Azure DevOps repositories en deployment van nieuwe versies wordt gedaan via Azure DevOps pipelines.

Gezien de huidige development stack van de Klantportaal API en de hosting in Azure is deze niet onderhevig aan een vernieuwingsslag en past deze in de beoogde nieuwe doelarchitectuur.

3.10 Infrastructuur

De huidige infrastructuur van het STARLight systeem en onderdelen, die aan vernieuwing onderhevig zijn, bevindt zich op het eigen lokale Ster netwerk en is daarbij ook volledig in eigen (server)beheer.

- Interne Windows Server met SQL Server Software t.b.v. de STARLight database
- Interne Windows Server met IIS Web Server Software t.b.v. de interne STARLight API's
- Interne Windows Server t.b.v. de STARLight Batch Scheduler (Windows Service)
- 'External' Windows Server met IIS Web Server in DMZ t.b.v. de externe STARLight API's B2B, AFAS en CRM
- Werkstations/Laptops met Windows 10/11 waar de STARLight Windows Applicatie op geopend kan worden
- De STARLight Windows Applicatie kan alleen geopend worden als een gebruiker zich rechtstreeks of via een VPN op het Ster netwerk bevindt.

3.11 PDF Generatie - Crystal Reports

Het STARLight systeem maakt gebruik van een pdf-generator voor rapportages en facturatie. Het huidige systeem maakt gebruik van Crystal Reports for .Net om rapporten en facturen vorm te geven en te vullen met gegevens vanuit de STARLight database. Die Crystal Reports formulieren kunnen als PDF worden gedownload en worden geprint.

De vernieuwing van het STARLight systeem vraagt om herziening van de inzet en afhankelijkheid van Crystal Reports t.b.v. rapportage opbouw en PDF-generatie.

4. Gewenste situatie (SOLL)

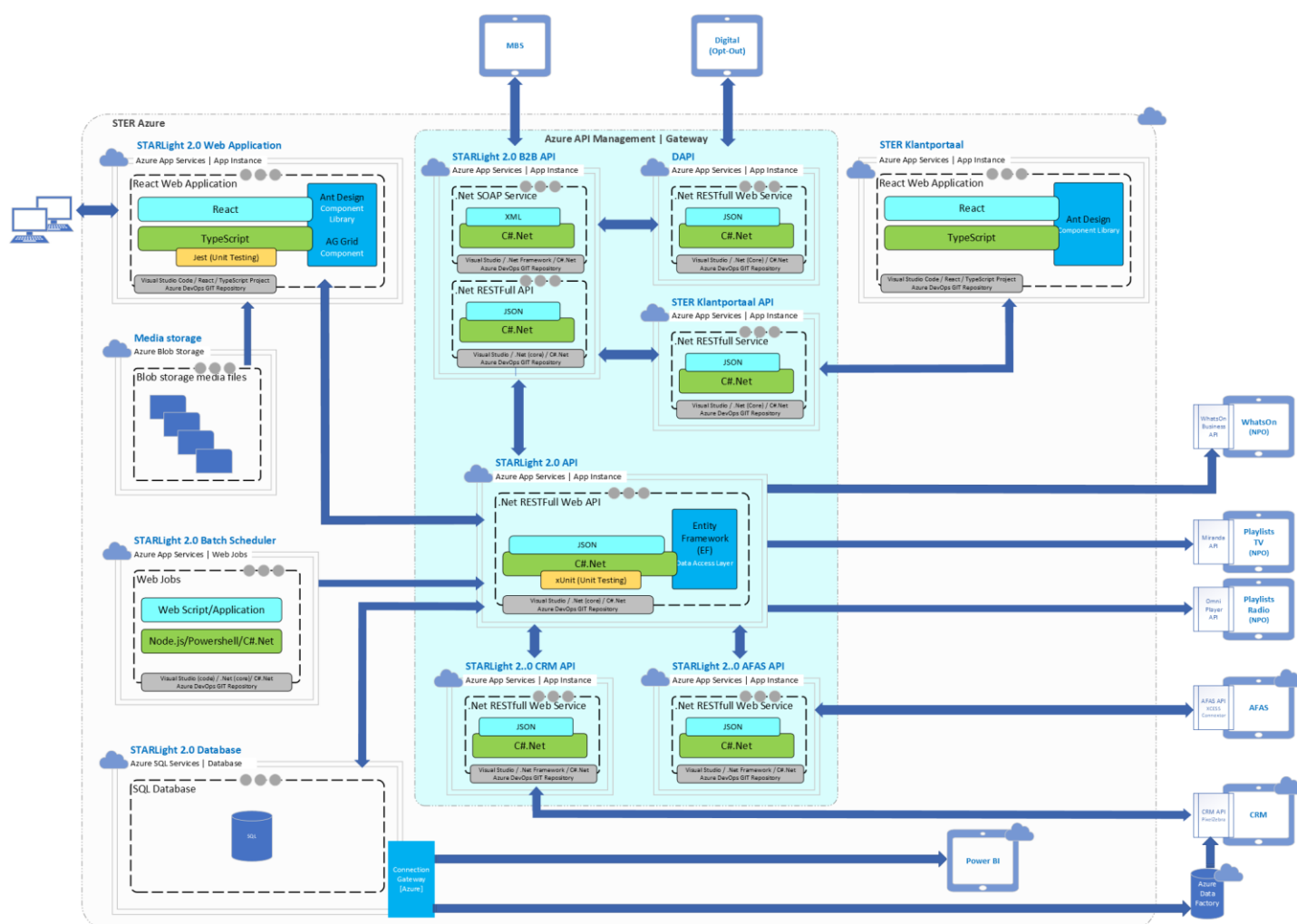
4.1 Inleiding

De gewenste situatie is een STARLight landschap, welke toekomstbestendig is op het vlak van development technologie, doorontwikkeling, infrastructuur beheer en schaalbaarheid. Het huidige STARLight landschap is primair gebaseerd op Microsoft (on-premise) development technologie en dat geldt ook voor de infrastructuur.

Een logische stap voor Ster is om de vernieuwing van het STARLight landschap binnen de Microsoft technologie te houden, want hiervoor is in feite al de nodige kennis in huis, zowel op het vlak van development als ook infrastructuur.

Kijken we naar de nieuwere generatie applicaties van de Ster, zoals o.a. het Klantportaal en de DAPI, dan zijn deze al gebaseerd op up-to-date Microsoft development technologie en Microsoft (Azure Cloud) infrastructuur.

De gewenste situatie die hier dan ook verder beschreven wordt is een vernieuwing van het STARLight landschap naar up-to-date Microsoft development technologie gebaseerd op Microsoft (Azure Cloud) services en infrastructuur.



4.2 STARLight 2.0 Database

Het STARLight databasemodel zal worden herzien. Er dient gekeken te worden naar opschoning van niet meer gebruikte tabellen en/of kolommen. Database integriteit zal beter in het database model verankerd moeten worden door correct gebruik van Primary, Foreign keys, constraints en indexen op de SQL Database zelf. De redenen om dit in het verleden niet te doen (zoals performance) zijn niet meer van toepassing in de huidige SQL Server versies. Azure SQL Services is gebaseerd op de laatste SQL Server versie.

4.3 STARLight 2.0 Web Applicatie

Web en cloud based is waar de Microsoft focus en visie ligt, hiermee wordt toekomstbestendigheid verzekerd vanuit het Microsoft perspectief. Ster wil met de vernieuwing van STARLight toekomstbestendig garanderen en zal daarom de Microsoft web en cloud based visie volgen.

De STARLight Applicatie als Windows Forms Applicatie komt daarmee te vervallen en zal worden omgebouwd naar een Web Applicatie. Dit is een eis van Ster. De nieuwste .Net (core) development stack van Microsoft biedt standaard zelfs al geen Visual Studio Windows Forms Project types meer aan.

In principe moeten alle onderdelen, zoals nu beschikbaar in de huidige STARLight applicatie, op basis van functionele werking en business logica exact terugkomen in de nieuwe Web Applicatie.

Functionaliteiten die niet meer in gebruik zijn, worden uiteraard verwijderd en in samenspraak met de business units kunnen ook nieuwe functionaliteiten op de backlog komen, waarbij m.b.v. een mvp gemanaged wordt op scope en tijd.

Door de overgang van een windows forms naar een webapplicatie zal user interface technische niet alles 1-op-1 kunnen worden nagebouwd. In de huidige windows forms applicatie bepaalt het Janus Winforms/GridEx control grotendeels de UI. Een dergelijke geavanceerd grid control is hoogstwaarschijnlijk in geen webcomponent bibliotheek terug te vinden. Om tot een gedegen keuze te komen tot inzet van een webcomponent bibliotheek zal onderzoek moeten plaatsvinden in de vorm van een Proof of Concept (POC). Een of meerdere webcomponent bibliotheken kunnen hierbij onderzocht worden om te bepalen welke bibliotheek het beste aan de grid eisen van de STARLight applicatie voldoet.

4.3.1 React/TypeScript vs Blazor

Binnen het huidige Microsoft .Net web ontwikkellandschap kan gekozen worden tussen zeer diverse (oudere) en nieuwe webtechnieken om tot een webapplicatie te komen. O.a. ASP.Net Web Forms, ASP.Net MVC Razor, ASP.NET MVC Blazor en React/Typescript of Angular/Typescript. Echter zit de huidige C# web ontwikkelaar momenteel op een van twee hoofdstromingen, zij het een webapplicatie op basis van React/Typescript ofwel ASP.Net MVC Blazor. Al het andere wordt of bestempeld als verouderd, zoals ASP.Net Web Forms of MVC Razor of het wordt niet breed genoeg gedragen binnen de Microsoft community zoals Angular. Angular kan zeker, maar Microsoft biedt meer (native) support voor React in deze.

ASP.Net MVC Blazor is een nieuwe opkomende specifieke Microsoft techniek, maar eigenlijk nog iets te nieuw. De ontwikkelaar heeft bij dit type project zeer specifieke Microsoft MVC skills nodig. De basis is Web Assembly, een gecompileerde applicatie dll, welke in de browser van de client draait. Het nadeel is dat je hiervoor een zeer moderne browser nodig hebt die Web Assembly ondersteunt (dat doen ze nog niet allemaal) en de dll moet eerst volledig gedownload worden voordat het kan beginnen. Het voordeel is dat deze eenmaal ingeladen in de browser een snelle gebruikerservaring geeft. Communicatie naar een back-end service gaat via RESTful API's.

React/Typescript is een bewezen en zeer breed toegepaste techniek binnen de web development community in zijn algemeenheid. De ontwikkelaar bij dit type project heeft in feite geen specifieke Microsoft development skills nodig als het gaat om React zelf. Er is dus bredere support als het gaat om ontwikkelaars met kennis. De component based basis van React maakt het zeer overzichtelijk en testen hebben aangetoond dat de snelheid van de React rendering engine juiste bij grotere en complexere applicaties beter is dan Blazor. Er zijn in feite weinig nadelen aan React. Communicatie naar back-end services gaan via RESTful API's, idem aan Blazor. De support in (oudere) browsers is groter dan bij Blazor en de Microsoft support is groot. Daarbij is het Ster Klantportaal ook ontwikkeld op basis van React/Typescript, kennis en kunde binnen Ster is reeds aanwezig. Het is een logische keuze om het nieuwe STARLight op basis van React/Typescript te ontwikkelen.

4.3.2 *UI Web Componenten Bibliotheek*

Een van de grootste uitdagingen bij de vernieuwing naar het web zit in het vervangen van het Janus Winforms/GridEx control, dit is een zeer geavanceerd Windows Forms Grid control. Gezien de functionaliteiten die het biedt zal hier niet direct een één-op-één vervanger voor gevonden kunnen worden t.b.v. een web interface.

Als we kijken naar de beschikbare (React) Web Control libraries, dan zijn hier zeer veel smaken in. Zo biedt Microsoft zijn eigen Fluent UI componenten bibliotheek met een eigen Data-Grid control. Fluent UI is vanuit Microsoft perspectief de standaard (en meest voor de hand liggende bibliotheek) als het gaat om User Interfaces, waarbij men de gebruiker een herkenbaar Microsoft (Office) experience wil geven. Zo was het Janus GridEx control destijds ook een Office Outlook look-alike.

Het Ster Klantportaal maakt gebruik van de Ant Design componenten bibliotheek, deze bibliotheek heeft een eigen Table control, welke mogelijk ingezet kan worden als grid control.

Om tot een passende keuze te komen m.b.t. de toe te passen componentenbibliotheek (of een combinatie van bibliotheken/componenten) is tijdens de totstandkoming van het Solution Design aan Ster geadviseerd om een Proof-of-Concept traject te doorlopen. Ster heeft dit Proof-of-Concept traject inmiddels doorlopen. Hierbij zijn meerdere React bibliotheken en/of combinaties van bibliotheken/componenten tegen elkaar uitgezet en in een Proof-of-Concept React webapplicatie bekeken en beoordeeld.

De uitkomst van dit Proof-of-Concept traject is een sterke voorkeur voor de inzet van de bij Ster reeds bekende Ant Design componenten bibliotheek t.b.v. de algemene look and feel en layout van webpagina's en componenten aangevuld met een geavanceerd web grid control van AG Grid. Het AG Grid React Web Control. In te zetten React componenten bibliotheek en web grid:

- [Ant Design React component library](#)
- [AG Grid React Web Control](#)

4.4 Interfaces

Alle STARLight interfaces komen in het nieuwe landschap primair beschikbaar als een RESTful Service. Als het geval zich voordoet dat een SOAP service nog noodzakelijk blijkt, zoals bijvoorbeeld voor de STARLight B2B API, omdat afnemende externe partijen niet zomaar via REST kunnen gaan communiceren. Dan zal er naast de RESTful variant ook een SOAP variant beschikbaar moeten blijven.

Daar waar API-code logica gelijk kan blijven aan bestaande API-code logica zal er sprake zijn van het porten van VB.Net syntax naar C#.Net. Herziening en opschoning van code dient ten allen tijde in acht te worden genomen.

4.4.1 STARLight API

Ombouw van .Net Framework SOAP Service naar .NET (core) RESTful Service (C#.Net). De grootste herziening hierbij is het gaan gebruiken van Microsoft's gestandaardiseerd high-level data access laag op basis van Entity Framework. Hiervoor dient het database model eerst op orde te zijn.

Het Entity Framework voorziet standaard in alle CRUD-methodes en geoptimaliseerde technieken om (grote sets aan) data in te laden en ook weg te schrijven. In combinatie met een integer uitgewerkt database model is dit een krachtig en data-safe model.

Vernieuwing:

- .Net Framework vervangen door .Net (Core)
- Programmeertaal VB.Net vervangen door C#.Net
- Beschikbaar stellen als RESTful service
- Toepassen gestandaardiseerde en geoptimaliseerde data access laag op basis van Entity Framework
- Hosting in Azure App Service

4.4.2 STARLight AFAS API

Ombouw naar .NET (core) RESTful API (C#.Net).

Vernieuwing:

- .Net Framework vervangen door .Net (Core)
- Programmeertaal VB.Net vervangen door C#.Net
- Beschikbaar stellen als RESTful service
- Hosting in Azure App Service

4.4.3 STARLight CRM API

Ombouw naar .NET (core) RESTful API (C#.Net).

Vernieuwing:

- .Net Framework vervangen door .Net (Core)
- Programmeertaal VB.Net vervangen door C#.Net
- Beschikbaar stellen als RESTful service
- Hosting in Azure App Service

4.4.4 STARLight B2B API

Ombouw naar .NET (core) RESTful API (C#.Net).

De SOAP Service variant zal beschikbaar moeten blijven voor bestaande afnemende externe partijen. .Net (Core) biedt standaard geen SOAP Service templates. Een vernieuwing zal worden uitgewerkt op basis van laatste versie .Net Framework WCF techniek, waarbij de bestaande VB.Net code wordt geport naar C#.NET.

Het idee is dat de afnemende externe partijen in de loop van tijd overgaan naar de RESTful variant, waarop de SOAP variant kan worden uitgefaseerd.

Vernieuwing:

- Programmeertaal VB.Net vervangen door C#.Net
- Beschikbaar stellen als .Net (core) RESTful service
- Beschikbaar stellen als .Net Framework WCF SOAP service
- Hosting in Azure App Service

4.5 STARLight Batch Scheduler

De huidige Batch Scheduler is ontwikkeld als een Windows Service, een Windows Service kan niet gehost worden in Azure en zal omgezet moeten worden naar een applicatie of service, welke gehost kan worden binnen het Azure applicatie landschap.

Het basis uitgangspunt van de vernieuwing is de overgang van oude .Net Framework Windows Service naar een .Net (Core) Azure App Services oplossing.

Vernieuwing:

- Programmeertaal VB.Net vervangen door C#.Net
- .Net (core) Azure App Services oplossing
- Hosting in Azure App Services

Er worden twee technologische opties onderkent, waarmee de Batch Scheduler binnen Azure gerealiseerd kan worden. Een oplossing gebaseerd op Azure Web Jobs en/of op basis van Hangfire. Hieronder een beschrijving en een vergelijk van de twee technieken.

Het inzetten van Hangfire heeft in principe de voorkeur, vanwege de extra job opties t.o.v. Azure Web Jobs. De extra Hangfire job opties hebben direct toegevoegde waarde voor de Batch Scheduler uitwerking op Azure.

Het is niet zo dat het een het ander uitsluit. Zo kan een Hangfire oplossing als Azure Web Job worden opgestart. Een combinatie van de twee behoort ook tot de mogelijkheden als blijkt dat hier voordeel uit gehaald kan worden.

4.5.1 Azure Web Jobs

Binnen het Azure Apps Services landschap kan gebruik gemaakt worden van Azure Web Jobs. Azure Web Jobs voorziet in het scheduled uitvoeren van jobs in de vorm van executables en/of scripts. Zeer vergelijkbaar met de Windows Scheduled Tasks.

De standaard Azure Web Jobs opties kent wel beperkingen. Zo is er bijvoorbeeld geen mogelijkheid om jobs afhankelijk van elkaar te laten starten. Een dergelijke logica zou apart ontwikkeld moeten worden binnen een Azure Web Jobs oplossing.

4.5.2 Hangfire

Hangfire is een job scheduler package welke geïntegreerd kan worden binnen een Azure .Net (Core) (web) applicatie. Een groot voordeel van Hangfire t.o.v. Azure Web Jobs is o.a. de mogelijkheid om jobs afhankelijk van elkaar te laten starten. Hangfire ondersteunt de volgende job types:

- Fire and Forget
- Continuation Jobs
- Logical Retries

Een ander voordeel van Hangfire is dat Ster ontwikkelaars reeds kennis hebben van Hangfire.

4.6 Digital

Blijft voorlopig de service die wordt afgenomen bij Opt-Out. Volledig eigen bouw zou een toekomstig project kunnen zijn, na de vernieuwing van het STARLight landschap.

De DAPI valt reeds binnen nieuwe Microsoft technologie. Development technologie blijft voorlopig as-is. Aanpassingen zijn noodzakelijk als de DAPI met een of meerdere STARLight API's communiceert welke omgebouwd zijn van SOAP naar RESTful service of waarvan de authenticatie wijze is aangepast.

4.7 Klantportaal

Het Klantportaal valt al binnen nieuwe Microsoft technologie. De development technologie van het huidige klantportaal sluit aan bij de nieuw in te zetten development technologie van STARLight. Aanpassingen zullen noodzakelijk zijn op de onderdelen waar het Klantportaal met één of meerdere STARLight API's communiceert welke omgebouwd zijn van SOAP naar een RESTful service of waarvan de authenticatie wijze is aangepast.

4.8 Afsplitsing in separate modules en/of microservices

In principe is het uitgangspunt van de vernieuwing, zoveel mogelijk as is, waarbij de database één database blijft en de api's en applicaties 1-op-1 worden vernieuwd. Hiermee is ook de snelste doorlooptijd te behalen voor een complete vernieuwing van het hele STARLight landschap.

Dat wil niet wegnemen dat tijdens het vernieuwen niet gekeken kan/mag worden naar mogelijke afsplitsing van onderdelen naar separate modules. Als tijdens het vernieuwen of herzien van onderdelen blijkt dat een afsplitsing mogelijk (of misschien wel logisch) is, omdat er bijvoorbeeld geen enkele relatie voor een onderdeel is t.o.v. de overige onderdelen, dan kan een afsplitsing overwogen worden.

Fictief voorbeeld: Het onderdeel TV Playlists binnen STARLight blijkt compleet op zichzelf staand te zijn. Het onderdeel gebruikt 6 tabellen binnen de database, deze tabellen hebben onderling een relatie, maar hebben verder geen relatie met andere tabellen in de database. Deze tabellen zouden dan afgesplitst kunnen worden naar een eigen database. Binnen de STARLight (interne) API kan dan ook de keuze gemaakt worden om voor de TV Playlists functionaliteit een afsplitsing te maken naar een eigen API.

Met een dergelijke afsplitsing kan dan eventueel een modulaire systeem opgebouwd worden, waarbij onderdelen onafhankelijk van elkaar functioneren en beheerd kunnen worden.

4.8.1 *Microservices*

Microservices wordt geroepen als men het heeft over losstaande, op zichzelf functionerende onderdelen. Het is ook erg hip. Een afsplitsing naar een aparte db en eigen api kan ook zeker de basis zijn voor één microservice/container, maar het grote verschil met bovenstaande mogelijke 'on-the-fly' afsplitsing en een echte microservices architectuur zit hem in het feit dat bij het opzetten van een microarchitectuur een as-is start niet als uitgangspunt genomen kan worden. Bij een microarchitectuur zal van meet af aan de architectuur uitgedacht en opgezet moeten worden en elk STARLight onderdeel zal eerst onder de loep genomen moeten worden. Dit betekent een complete ombouw van het bestaande en is een zeer tijdrovend proces. Het is een grote vraag of STARLight überhaupt microarchitectuur waardig is. Want wanneer krijgt een microarchitectuur toegevoegde waarde? Hieronder twee punten waarom men een microarchitectuur zou kunnen overwegen.

1. Per onderdeel (microservice) een eigen roadmap, eigen ontwikkelteam en eigen ontwikkelstraat
2. Veel variatie in gebruikers aantallen en piekmomenten (opschalen/afschalen met extra containers)

Voor STARLight spelen bovenstaande punten helemaal niet. Heel STARLight zal door een enkel ontwikkelteam worden beheerd. De gebruikersgroep is zeer constant en zeer voorspelbaar. Opsplitsing is op voorhand eigenlijk niet nodig, want de applicatie structuur, zoals hij nu is, werkt prima. Een microarchitectuur voor STARLight is overbodig.

4.9 **Infrastructuur**

Het volledige vernieuwde STARLight applicatie landschap zal gaan landen binnen de Azure Cloud.

Door gebruik te maken van Azure Services, zoals het bestaande Klantportaal en de DAPI dat ook al doen, zullen de diverse STARLight onderdelen als volgt geplaatst kunnen worden binnen Azure:

4.9.1 *Azure App Services*

Apps en Api's worden gehost in Azure App Services. De toegang tot de Apps en Api's worden geregeld via de Authenticatie en Autorisatie mogelijkheden binnen Azure App Services.

- STARLight Web Applicatie
- STARLight RESTful Api's
- STARLight SOAP Services
- STARLight Web Jobs

4.9.2 *Azure SQL Services*

SQL Databases worden gehost in Azure SQL Services. De toegang tot de database wordt geregeld via de Authenticatie en Autorisatie mogelijkheden binnen Azure SQL Services.

- STARLight SQL Database

4.9.3 *Azure Blob Storage*

Opslag van (grote) mediabestanden, zoals de commercial video's in origineel (raw) format, welke via een webprotocol gedownload kunnen worden om vervolgens af te kunnen spelen via client software, wordt gehost in Azure Blob Storage. De toegang tot de bestanden wordt geregeld via de Authenticatie en Autorisatie mogelijkheden binnen Azure Blob Storage.

- STARLight Mediabestanden (Commercial video's)

4.10 PDF Generatie – Aspose.PDF

Het STARLight systeem maakt gebruik van PDF generatie voor o.a. rapportages en facturatie. Het huidige systeem maakt gebruik van Crystal Reports rapportages voor PDF generatie. Crystal Reports is in feite een volwaardig rapportagesysteem op zichzelf met een visual (design) interface, maar wordt binnen de STARLight systeem eigenlijk maar beperkt ingezet enkel alleen voor het samenstellen van rapporten om uiteindelijk een PDF te kunnen genereren. Als alternatief voor PDF generatie kan een meer licht gewicht tool worden ingezet in de vorm van Aspose.PDF for .Net.

Aspose.PDF for .Net is een development API, die zich puur focused om vanuit code een PDF te genereren. Volledig te integreren in het STARLight 2.0 eco-systeem om de gewenste pdf-bestanden on-the-fly samen te stellen en vervolgens ter download (of print) beschikbaar te stellen binnen de STARLight applicatie.

5. Migratie (IST naar SOLL)

5.1 Inleiding

De migratie van de bestaande naar de gewenste situatie is met name gericht op het in technische zin ombouwen van STARLight naar nieuwe Microsoft technologie en het landen in de Azure Cloud. De bestaande STARLight Microsoft .net development stack heeft een Windows based en lokaal netwerk focus. De gewenste doelarchitectuur heeft een web based en Azure cloud focus. In dit hoofdstuk wordt in hoofdlijnen uitgeschreven hoe de migratie van oud naar nieuw plaats kan vinden.

We onderkennen de onderstaande stappen:

1. Vernieuwing van de STARLight Database

Het relationele databasemodel zal worden herzien. Focus hierbij op een betere verankering van database integriteit en data consistentie in het databasemodel zelf. O.a. het fysiek toepassen van foreign keys en constraints.

Daar waar mogelijk vindt opschoning plaats van verouderde overbodige tabellen en/of kolommen.

De database(s) zullen landen in Azure SQL Database Services

2. STARLight API's op basis van SOAP services ontwikkeld op verouderde ontwikkeltaal vb.net worden omgebouwd naar RESTful services op basis van up-to-date .NET (core) Web API technologie en ontwikkeltaal c#.net.

De RESTful API's zullen landen in Azure App Services.

3. Daar waar STARLight API's op basis van SOAP services nog beschikbaar moeten blijven (o.a. de B2B API) wordt een nieuwe SOAP service ontwikkeld op basis van de laatste .Net Framework WCF technologie en ontwikkeltaal c#.net.

De SOAP Services zullen landen in Azure App Services.

4. De STARLight Windows Forms front-end applicatie op basis van verouderde componenten en ontwikkeltaal vb.net zal worden omgebouwd naar een Web front-end applicatie met up-to-date componenten gebaseerd op moderne web development stack React en Typescript.

De STARLight (React) Web Application zal landen in Azure App Services.

5. De STARLight Batch Scheduler op basis van een Windows Service en verouderde ontwikkeltaal vb.net zal worden omgebouwd naar een applicatie op basis nieuwe microsoft tech, hier kan nog een definitieve keuze in worden gemaakt, bijvoorbeeld een oplossing op basis van .Net (core) C# of mogelijk een scripted oplossing op basis van powershell. Een oplossing zal vervolgens op basis van Azure Web Jobs en/of een Azure App i.c.m. Hangfire continous of scheduled de STARLight (daily) batches uitvoeren.

De STARLight Batch Scheduler zal landen in Azure App Services.

6. Hosting van mediafiles zal verplaatst worden van een interne file server naar Azure Blob Storage. Mediafiles worden altijd eerst lokaal gedownload alvorens een gebruiker deze afspeelt via een lokaal geïnstalleerde mediaspeler. Azure Blob Storage heeft grote opslagcapaciteit en biedt een standaard download over http.

5.2 Aandachtspunten

5.2.1 *Digitale module*

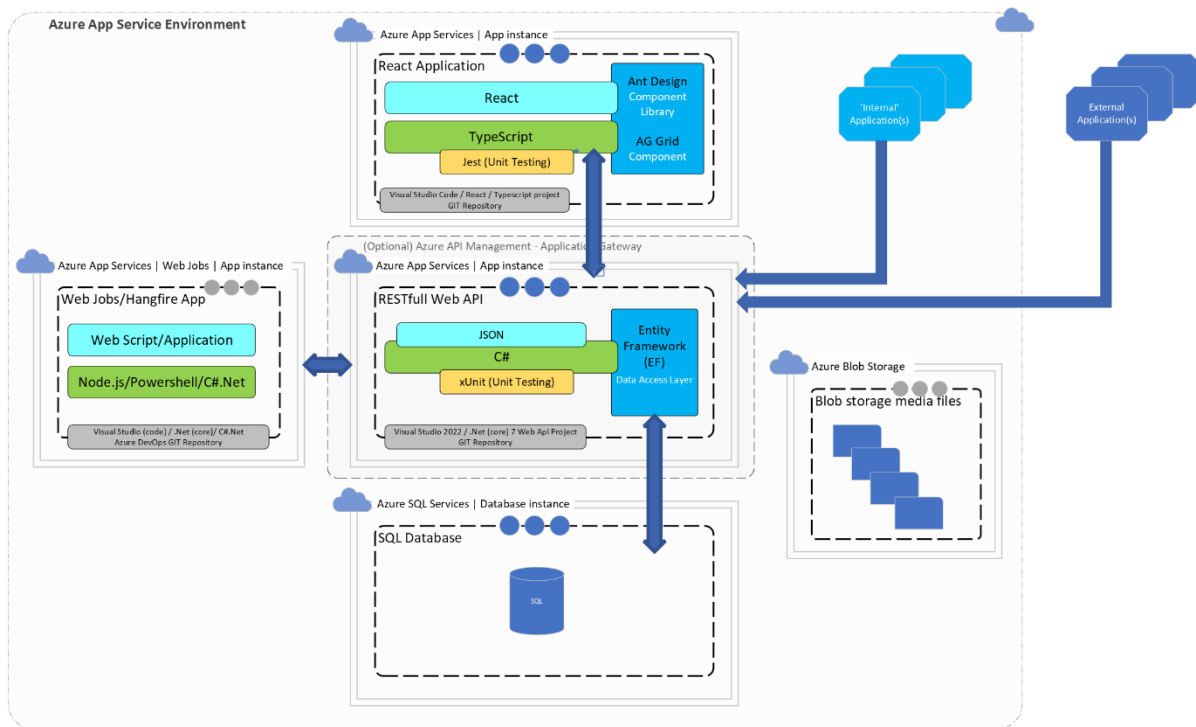
Indien ervoor gekozen wordt om de digitale module te vervangen en na te bouwen met een product development team van Ster, is aan te bevelen om deze activiteiten in een apart – doch gerelateerd - (deel-)project te laten plaatsvinden met een eigen product development team. De voornaamste reden hiervoor is dat op deze manier prioriteit gegeven kan worden aan de ontwikkeling van STARlight 2.0 die hogere prioriteit heeft. Daarbij komt dat de ontwikkeling van de digitale module gemakkelijk separaat kan plaatsvinden. Op deze manier wordt het risico geminimaliseerd op uitloop.

5.2.2 *Kennisoverdracht STARLight business logic*

Het grootste deel van de kennis van STARLight business logic in algemene zin, maar ook in de (technische) diepte, zit in de hoofden van de twee huidige STARLight ontwikkelaars Herman en Willem. Het wordt sterk aangeraden om Herman en Willem tijdens de herziening van o.a. het database model en de STARLight functionaliteiten in de periode dat ze nog werkzaam zijn voor Ster mee te laten draaien als zijnde 'Business Consultants'.

6. Concept Solution Design

6.1 High Level Design



6.2 Azure Services

6.2.1 Azure App Services

De nieuwe STARLight 2.0 web api's, web applications en web jobs worden ontwikkeld om gehost te kunnen worden in een Azure App Services omgeving.

- STARLight 'interne' API
- STARLight B2B API
- STARLight AFAS API
- STARLight CRM API
- STARLight Web Application
- STARLight Batch Scheduler (Web Jobs)

6.2.2 Azure SQL Services

De nieuwe STARLight 2.0 database wordt gehost in de Azure SQL Database services omgeving.

Inzet van Azure SQL High Availability services wordt aanbevolen t.b.v. hogere mate van verzekerde beschikbaarheid.

6.2.3 Azure Blob Storage

Mediafiles worden opgeslagen in Azure Blob Storage. Hier kunnen grote bestanden worden opgeslagen en Azure Blob Storage biedt standaard download over http(s).

6.2.4 Azure API Management / Application Gateway

Azure API Management en/of Application Gateway services kunnen ingezet worden als er behoefte is aan meer centrale controle over de toegang- en beveiliging van de diverse API's van buitenaf. Azure API Management biedt ook geavanceerde API-monitoring en rapportages over gebruik.

Met de Application Gateway krijgt men geavanceerde opties op het gebied van toegang en routing naar API's en Apps.

6.3 Privacy en veiligheid

Door gebruik te maken van de vele standaard beschikbare Azure privacy en veiligheidsconfiguratie opties kan een zeer hoog level van beveiliging van dataverkeer en bescherming van informatie worden gerealiseerd. De infrastructuur van Azure voorziet in de allerlaatste beveiligingstechnieken en wordt continue gemonitored en up-to-date gehouden.

Aanbevelingen:

- Maak gebruik van ssl certificaten voor beveiligd dataverkeer via https en/of ftps.
- Maak gebruik van (extra) encryptie van gevoelige data, welke over de lijn gaat. Ook als deze binnen de Azure omgeving zelf gaat. Azure services is in basis een Shared omgeving.
- Maak gebruik van IP-restricties (ip-whitelisting) om te toegang te beperken tot een specifiek netwerk.
- Maak (optioneel) gebruik van een geïsoleerde (private) Azure App Service Environment om volledig afgeschermd te zijn van overige (gedeelde) Azure omgevingen. In combinatie met een Azure Virtual Network kunnen apps en services alleen benaderbaar worden gemaakt binnen een eigen Azure afgeschermd (v)an omgeving.
- Maak gebruik van Azure Gateways (App, SQL) met meer geavanceerde configuratieopties m.b.t. traffic routing, monitoring en o.a. het afvangen van o.a. DDoS aanvallen.
- Sla wachtwoorden en sleutels nooit op in de code, maar maak gebruik van encrypted configuratie opties zoals o.a. Azure Key Vault.

6.4 Authenticatie en autorisatie

Door gebruik te maken van de standaard Azure authenticatie en autorisatie mogelijkheden voor toegang naar apps en services is het level voor veilige toegang zeer hoog en de mate van configuratieoptie hierin zeer hoog.

Anonieme toegang naar apps en services is zeker mogelijk maar wordt sterk afgeraden.

Voor gebruikerstoegang tot apps of services maak zoveel mogelijk gebruik van de Azure Active Directory (AAD) services i.c.m. het Microsoft Identity platform. AAD authenticatie en autorisatie is de standaard binnen de Azure services, toegangsrestricties op AAD gebruikersniveau kan standaard worden ingeregeld binnen de diverse Azure services. AAD is de default user store voor Microsoft 365 gebruikers.

Optioneel ondersteund Azure Services ook de OpenId standaard, waarmee met andere login providers kan worden gekoppeld. Denk hierbij bijvoorbeeld aan Google of Facebook-accounts om gebruikers te laten inloggen op een App.

Voor system to system (ofwel Business to Business) koppelingen maak gebruik van Client Certificate authenticatie voor het hoogste level van beveiliging tussen systemen.

6.5 Beschikbaarheid

De hoogte van het servicelevel wordt met name bepaald op basis van het type Azure service abonnement wat wordt afgenomen. Bij een Standard/Premium abonnement wordt een SLA up-time van 99.95% gegeven. Een free/shared abonnement kent geen SLA. Ondanks de hoge up-time bij een standaard productie abonnement (Standard/Premium/Isolated) zal er altijd rekening gehouden moeten worden met de minimale kans op downtime.

Om een hoger beschikbaarheid level te halen kan o.a. gebruik gemaakt worden van Azure Availability Zones. Zones zijn fysiek en logisch gescheiden en bij het uitvallen van een zone wordt automatisch overgeschakeld naar een andere zone om downtime verder te minimaliseren. Data wordt beveiligd en encrypted in sync gehouden tussen de zones. De beschikbaarheid kan hiermee opgehoogd worden naar 99.99%.

Binnen Azure Apps en SQL-services is High Availability beschikbaar vanaf het Standard abonnement.

6.6 Schaalbaarheid

Schaalbaarheid komt beschikbaar vanaf het Azure Basis abonnement. Onder schaalbaarheid wordt onder meer verstaan het tijdelijk of permanent opschalen van resources binnen een Azure App of SQL-service naar meer of snellere resources om een groter gebruikersaantal of een andere (tijdelijke) piekbelasting te ondervangen.

Automatisch laten schalen van resources op basis van gebruik of tijdstip behoort ook tot mogelijkheden. Vanaf het Azure Standard abonnement komen auto-scaling configuratieopties beschikbaar.

6.7 Monitoring/Logging

Monitoring en logging zit standaard ingebakken in de Azure App en SQL-services. Azure Applications Insight geeft inzicht in het gebruik en verbruik van een app of service. Als er behoefte is aan meer inzicht of gevanceerde (overkoepelende) dashboards en rapportages kan gebruik gemaakt worden van Azure Monitoring.

6.8 OTAP/CI/CD

Door gebruik te maken van Azure DevOps is er een compleet uitgewerkte ontwikkel en deploy omgeving beschikbaar in te zetten bij alle ontwikkel- en beheerwerkzaamheden. Broncode wordt op een centrale plek met versiebeheer (GIT) centraal opgeslagen. Ontwikkelteams kunnen overzichtelijk samenwerken op dezelfde code base.

Iteratief samenwerken (SCRUM) op basis van backlogs, boards en stories zit standaard ingebakken in de Azure DevOps omgeving.

(Geautomatiseerd) compileren van code, het testen van builds, klaarzetten van releases en het uitrollen van releases op diverse omgevingen test, acceptatie en productie (OTAP) kan gerealiseerd worden door gebruik te maken van Azure DevOps pipelines. Hiermee kan een Continuous Integration en Continuous Deployment (CI/CD) straat op zeer hoog niveau worden gerealiseerd.