

API URI richtlijnen

Titel : API URI richtlijnen
Datum : 01-05-2023
Versie : 2.0
Status : Vastgesteld
Opdrachtgever : MFF-BAS Technische Commissie

Inhoudsopgave

Colofon	3
Versiegeschiedenis	3
Verspreiding.....	3
Gerefereerde documenten	4
Begrippen- en afkortingenlijst.....	4
1	Inleiding..... 6
2	Algemene uitgangspunten 8
3	Naamgeving URI's 10
3.1	Inleiding naamgevingen..... 10
3.2	Verplichte opbouw van URI's..... 10
3.3	Onderdeel <Protocol>..... 10
3.4	Onderdeel <Autoriteit> (domeingedeelte)..... 11
3.4.1	Beveiligingskenmerken 11
3.4.2	Wijze van ontsluiten (techniek)..... 11
3.4.3	Omgevingsaanduiding..... 13
3.4.4	Routeerfunctie 13
3.4.5	Hoofddomeinnaam 16
3.5	Onderdeel <Pad> (resource-gedeelte) 17
3.5.1	Hoofdniveau 17
3.5.2	Eerste subniveau binnen <Pad> 19
3.5.3	Tweede subniveau binnen <Pad> 19
3.5.4	Naam API binnen <Pad>..... 20
3.5.5	Versie binnen <Pad>..... 21
3.5.6	Collectie binnen <Pad>..... 21
3.6	Onderdeel <Query> 22
3.7	Onderdeel <Fragment>..... 23
4	Resulterende opbouw van een energysector-URI 24
4.1	Naamconventie energysector-URI..... 24
4.2	Voorbeelden energysector-URI's..... 24

Colofon

Versiegeschiedenis

Versienummer	Status	Markering/wijzigingen
0.1	Concept	Initiële versie
0.2	Concept	Toevoeging routeerfunctie, aanpassing API-naam naar primaire resource
0.5	Concept	Resources baseren op bedrijfsobjecten, verwerken commentaar
0.7	Concept	Feedback van overleg 9 december2020 verwerkt
0.9	Ter vaststelling	Laatste feedback verwerkt
0.91	Ter vaststelling	Review SI
0.92	Ter vaststelling	Query paragraaf aangepast
1.0	Vastgesteld	Versie na vaststelling in ALV NEDU van 16 december 2020
1.1	Concept	Gedeeltelijk herschreven: conventie is nu gebaseerd op NBility bedrijfsobjecten Template gewijzigd in MFF-BAS Ter review aangeboden aan werkgroep en EA
1.9	Ter vaststelling	Feedback van werkgroep en EA verwerkt
2.0	Vastgesteld	Versie na vaststelling in ALV MFFBAS van 1 mei 2023

Verspreiding

Naam	Datum	Versie
NEDU Technische Commissie	17-12-2019	0.7
NEDU Technische Commissie	10-02-2020	0.9
NEDU	01-10-2020	0.92
NEDU	16-12-2020	1.0
API werkgroep en EA	8-12-2022	1.1
MFFBAS	10-2-2023	1.9
MFFBAS	01-05-2023	2.0

Gerefereerde documenten

Nummer	Omschrijving	Datum	Versie	Auteur
1.	API-strategie voor de Nederlandse overheid API Strategie Algemeen (Nederlandse API Strategie I) API Design Rules (Nederlandse API Strategie IIa) API Designrules Extensions (Nederlandse API Strategie IIb)	09-03-2022 09-07-2020 13-10-2021	Handreiking 1.0 Handreiking	Geonovum e.a. Logius e.a. Geonovum e.a.
2.	API ontwerprichtlijnen MFFBAS https://edns.sharepoint.com/sites/EDSN/SysteemDocumentatie/ServiceDesign/MFF%20BAS%20API%20Ontwerprichtlijnen%20v3.0%20(vastgesteld).pdf	10-11-2022	3.0	MFF-BAS werkgroep API-Strategie
3.	Lijst open standaarden Overheid https://www.forumstandaardisatie.nl/open-standaarden/lijst	n.v.t.	n.v.t.	Bureau Forum Standaardisatie
4.	API- en URI-strategie DSO https://aandeslagmetdeomgevingswet.nl/digitaal-stelsel/technisch-aansluiten/standaarden/api-uri-strategie/	26-03-202	2.0	A.J. Sloos e.a.
5.	Dissertatie Roy Fielding https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf	2000	n.v.t.	Roy Fielding
6.	IETF Internet Standard RFC3986 URI Generic Syntax https://www.ietf.org/rfc/rfc3986.txt	Januari 2005	Internet Standard	IETF
7.	NBility model van de samenwerkende netbeheerders https://www.edsn.nl/nbility-model	April 2022	2.0	NBNL
8.	Google Cloud documentatie https://cloud.google.com/apis/design/resources#what_is_a_rest_api	n.v.t.	n.v.t.	Google

Begrippen- en afkortingenlijst

Begrip	Omschrijving
API	Application Programming Interface
ART	Agile Release Train
BOM	Business Object Model
C-AR	Centraal Aansluitingen Register
C-ARM	Centrale Allocatie Reconciliatie en Meetdata
CMF	Centrale Markt facilitering
CIM	Common Information Model
CPS	Centraal Postbus Systeem
CRUD	Create Read Update Delete
DNS	Domain Name System
DSO	Digitaal Stelsel Omgevingswet
ldata	Linked Data
MFF-BAS	Markt Faciliterings Forum - Beheerder Afspraken Stelsel
MMC	Markt-Markt Communicatie hub
Nexus	Berichtenhub voor wholesale gas berichten

Begrip	Omschrijving
NC RFG	Netcode Requirements For Generators
REST	REpresentational State Transfer
SOAP	Simple Object Acces Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name

1 Inleiding

Dit document hoort bij het document MFF-BAS API ontwerprichtlijnen [2]. Alhoewel deze laatste is gebaseerd op de API-strategie voor de Nederlandse Overheid [1], voorzag deze niet in een URI-strategie. Daarom is de keuze gemaakt door de Werkgroep API-Strategie om zelf te voorzien in afspraken omtrent het gebruik en toepassen van URI's. Het resultaat daarvan is het voorliggende document.

Om REST API's¹ opvraagbaar te maken is het nodig om deze een digitaal adres te geven. Binnen de context van het Internet worden deze adressen URI's genoemd, waarbij de meest voorkomende vorm een URL is, ook wel bekend als een link. Er is gekozen om de URI richtlijnen te inspireren op de URI-strategie van de digitaal stelsel omgevingswet [4]. Deze strategie is volwassen en goed doordacht. Daarnaast levert de sector zelf ook informatie aan de overheid. Diverse partijen in de sector moeten zelf aansluiten op het digitaal stelsel van de Nederlandse overheid.

De voorliggende URI-strategie is aldus gebaseerd op de volgende bronnen²:

- API-strategie voor de Nederlandse overheid [1];
- API-strategie van de digitaal stelsel omgevingswet [4];
- URI-strategie van de digitaal stelsel omgevingswet [4];
- Inzichten van Alliander, Enexis, Stedin en EDSN.

URI's kunnen voor meerdere zaken worden gebruikt. De overheid onderscheidt de volgende vier verschillende categorieën [4]:

1. Webpagina's van gebruikerstoepassingen;
2. SOAP webservices;
3. REST API's;
4. Linked Data.

Aangezien deze URI-strategie zich richt op API gebruik, beperken we ons (vooralsnog) tot REST API URI's. Er wordt echter al wel rekening gehouden met het toepassen van de hier besproken conventie voor andere manieren van ontsluiten (zie paragraaf [Wijze van ontsluiten \(techniek\)](#)).

Noodzaak URI richtlijnen

URI's zijn onlosmakelijk verbonden met API's. Elke informatie en/of functionaliteit die wordt aangeboden via een API heeft een identificatie (=URI) nodig, waardoor een REST API vindbaar en toegankelijk wordt. URI richtlijnen geven een praktische vertaling voor het identificeren van API's door een structuuroopbouw uit te werken.

Samenhang begrippen URI, URN, URL en resource [3]

Om dit onderwerp goed te kunnen begrijpen is een stukje uitleg nodig over de definitie van URI's, URN's en URL's. URI's bieden een mechanisme om naar resources te verwijzen ongeacht waar deze zich op het Internet bevinden. URI's zijn voor mensen leesbaar omdat dit de interpretatie en het leggen van relaties vereenvoudigt.

Een "resource" kan elk mogelijk ding zijn, zoals fysieke objecten (transformator, kabel, geografische

¹ Waar in dit document over API's wordt gesproken, worden REST API's verstaan.

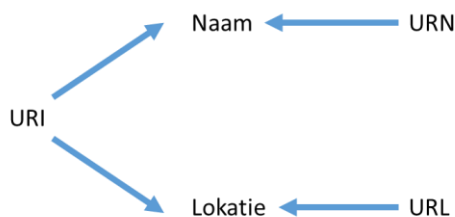
² Delen van deze documenten zijn overgenomen op deze pagina.

locatie), abstracties (begrip, model) alsook informatie-elementen (webpagina). De afkortingen URN's, URI's en URL's bevatten allen de term "Resource". De resource is het basis concept waarop de gehele naamconventie is gebaseerd.

Een **Uniform Resource Identifier** (URI) is een gestandaardiseerde manier om bronnen van informatie (webpages, tekst, afbeeldingen, etc.) op het Internet te identificeren. Een URI identificeert een bron van informatie ("resource") aan de hand van een hiërarchische beschrijving (reeks tekens) die meestal een locatie op een netwerk representeert. URI's zijn unieke verwijzingen naar digitale objecten. Identificatie aan de hand van een URI maakt interactie met verschillende bronnen van informatie via een netwerk (zoals het World Wide Web) mogelijk. De meest voorkomende vorm van een URI is de Uniform Resource Locator (URL), vaak aangeduid als een webadres. URL en URN zijn beiden een specifieke vorm van een URI.

Een **Uniform Resource Name** (URN) beschrijft eenduidig de naam van een "resource". Een URN is een URI, maar geen URL. Dat betekent dat een URN zonder inbedding in een URI of URL niet vindbaar is op het Internet.

Een **Uniform Resource Locator** (URL) beschrijft eenduidig de locatie van een "resource" en deze locatie is wel vindbaar op het Internet. Een URL wordt gebruikt om anderen toegang te geven tot een bron. Binnen het Digitaal Stelsel Omgevingswet (DSO) [4] worden URL's gebruikt voor SOAP-services, REST-API's en Linked Data resources die sectorbreed worden aangeboden.



Figuur 1 Samenhang URI, URL en URN

Een URI die een naam specificceert is een URN.

Een URI die een locatie specificceert is een URL.

Een URI die een naam en een locatie specificceert is een URI.

Voorbeelden van URI's:

- <ftp://ftp.is.co.za/rfc/rfc1808.txt>
- <http://www.ietf.org/rfc/rfc2396.txt>
- [ldap://\[2001:db8::7\]/c=GB?objectClass?one](ldap://[2001:db8::7]/c=GB?objectClass?one)
- <mailto:John.Doe@example.com>
- <telnet://192.0.2.16:80/>
- <urn:oasis:names:specification:docbook:dtd:xml:4.1.2>

Scope

Dit document beschrijft alleen de URI-strategie en gaat niet in op de inhoudelijke toepassing ervan.

Voor informatie over het toepassen van API's wordt verwezen we naar de MFF-BAS API ontwerprichtlijnen [2].

De toepassing van deze naamconventie is leidend en beperkt tot het gebruik binnen het CMF. In de praktijk komt dit neer op alle REST API's die EDSN bouwt en onderhoudt. Andere marktrollen worden geadviseerd om zoveel mogelijk deze richtlijnen te volgen.

2 Algemene uitgangspunten

REST API's zijn populair omwille van hun eenvoud en hergebruik van het HTTP(S) protocol dat gebruik maakt van een subset van de hiervoor besproken generieke URI's, te weten URL's.

Voor alle URI's gelden de volgende uitgangspunten:

ID	01
Uitgangspunt	URI's zijn resources-georiënteerd
Implicatie(s)	Resources vertegenwoordigen altijd dingen, objecten (zelfstandige naamwoorden). De functie (werkwoord) die op een Resource aangrijpt, zoals gebruikelijk is bij de naamgeving van webservices, is in deze naamgeving dus niet toegestaan.
Rationale	Om informatie op globale schaal eenvoudig te kunnen delen was een hernieuwd inzicht in het ontwikkelen van netwerk-georiënteerde applicaties nodig. Het combineren van de internet-standaarden HTTP (RFC2616) en URI (RFC3986) hebben in 2000 door Roy Fielding's dissertatie [5] geleid tot de hedendaagse resource-oriëntatie.
Voorbeelden (illustratief)	market-agreements (CIM), power-generating-modules (NC RFG)

ID	02
Uitgangspunt	URI's zijn langdurig stabiel
Implicatie(s)	URI's bevatten geen organisatienamen, systeemnamen, projectnamen, servernamen, etc.
Rationale	Om de wendbaarheid van de API's c.q. het IT-landschap te vergroten is het nodig om de impact bij verandering zo klein mogelijk te houden. Duurzame URI's dragen hieraan bij.
Voorbeelden (illustratief)	<p>Een voorbeeld van een slechte URI: https://pnp-webdisp.bedrijf.local:8190/irj/portal/urenverantwoording geeft de webapplicatie voor het schrijven van uren.</p> <p>Deze URI is, naast het feit dat hij niet gemakkelijk te lezen en te onthouden is, erg applicatiespecifiek en toont implementatie-complexiteit. Daarmee is hij dus niet langdurig stabiel. Wat als we overstappen op andere software, een andere poort? Dan moeten gebruikers hun links aanpassen. Als dit kan worden voorkomen, heeft dit de voorkeur.</p> <p>Bedrijven kunnen fuseren of nieuwe namen krijgen. Een langdurige stabiele URI voorziet in dit soort scenario's.</p>

ID	03
Uitgangspunt	Centrale URI's worden organisatie-onafhankelijk benoemd
Implicatie(s)	Lokale diensten zijn de verantwoordelijkheid van de organisatie die deze dienst aanbiedt. Voor centraal aangeboden URI's zal een centrale organisatie in de

	energiesector verantwoordelijk moeten zijn voor het bijhouden van de DNS-records (anders kans op wildgroei, onbedoelde botsingen van namen).
Rationale	Alleen diensten die een landelijk of centraal karakter binnen de energiesector hebben, dienen zich te houden aan afspraken die in dit document gemaakt worden. Dezelfde/vergelijkbare diensten die door één of meerdere aanbieders lokaal worden aangeboden zijn buiten scope.
Voorbeelden (illustratief)	Landelijke, centrale dienst die in scope is: CMF Decentraal: services die sector leden zelf uitvoeren zoals klantenservices, asset management, etc.

ID	04
Uitgangspunt	De schrijfwijze van URI's is, behalve het <Query>-gedeelte, volledig in kleine letters (lower case)
Implicatie(s)	Doordat hoofdletters niet zijn toegestaan kan de leesbaarheid van paden in sommige gevallen verminderen. Een koppelstreepje (hyphen) kan dan gebruikt worden om de leesbaarheid te vergroten (tezamen ook wel kebab case genoemd)
Rationale	URI's bestaan uit een domeingedeelte en een resource-gedeelte. De internetstandaard RFC3986 stelt kleine letters verplicht voor het domeingedeelte. In het resource-gedeelte zouden hoofdletters toegepast kunnen worden, maar omwille van duidelijkheid en eenduidigheid wordt in deze conventie gekozen voor het verplichten van kleine letters, ook voor het resource gedeelte
Voorbeelden (illustratief)	https://api.cmf.energysector.nl/energy-market/market-participant/market-agreement/contract-management/v1/market-agreements/879999999999999999

3 Naamgeving URI's

3.1 Inleiding naamgevingen

Om tot een goede naamgeving van objecten (in deze context: resources) te komen is geen triviale zaak. De reden hiervoor is dat een naamgeving alleen betekenis heeft voor mensen. Een naamgeving is dus volledig arbitrair te noemen.

De hier voorgestelde namen voor URI-objecten, is opgebouwd uit verschillende kenmerken. Per kenmerk zal een overweging gemaakt worden of het in de context van de energiesector zinvol is en wat de beste invulling hiervoor is.

3.2 Verplichte opbouw van URI's

De basis opbouw van de MFF BAS API URI's volgt de internetstandaard RFC3986 [6]:

```
<Protocol>:"//"<>Auteuriteit>"/">Pad>["?">Query>]"/"["#">Fragment>]
```

Onderdeel	Toelichting
<Protocol >	Geeft het protocol aan waarmee de URI bereikt kan worden, bijvoorbeeld "https" of "ftps".
<Auteuriteit> (domeingedeelte)	De autoriteit (of domeinnaam) voor API's. Dit gedeelte geeft de locatie weer binnen het internetdomein.
<Pad> (resource-gedeelte)	Het unieke pad dat een resource identificeert of een locatie van een resource aangeeft binnen het domeingedeelte. Hierbij zijn de onderdelen binnen het pad gescheiden door een slash ("/"), waarmee de resource verder hiërarchisch ingedeeld kan worden.
<Query>	Dit gedeelte bevat niet-hiërarchische data die samen met het <Pad> de resource identificeert. Het gaat hier vaak om parameters die als filter werken.
<Fragment>	Een fragment identificatie is een identificatie van een resource die ondergeschikt is aan een andere primaire resource.

Bovenstaande generieke opzet zal worden gebruikt als startpunt voor de specificatie van de REST URI's.

3.3 Onderdeel <Protocol>

Allereerst dient een keuze gemaakt te worden voor het <Protocol> (in de RFC "scheme" genoemd). Hier kiezen we simpelweg voor de defacto standaard.

Keuze:

voor het onderdeel <Protocol> wordt *https* gekozen

Motivatie:

HTTPS biedt beveiliging van het transport van data en is een defacto standaard.

3.4 Onderdeel <Autoriteit> (domeingedeelte)

Het onderdeel <Autoriteit> zal uit een aantal kenmerken bestaan, die in deze paragraaf worden besproken.

3.4.1 Beveiligingskenmerken

Er kan gekozen worden om beveiligingskenmerken toe te voegen aan de URI of niet. Dit heeft voordelen en nadelen.

Optie	Voordelen	Nadelen
1. Beveiligingskenmerk niet opnemen in de URI	Flexibel bij wijziging van beveiligingsniveau	Beveiligingsniveau is niet af te leiden uit de URI.
2. Beveiligingskenmerk opnemen in URI	Het beveiligingsniveau is op te maken uit de URI.	Het meegeven van beveiligingsdetails in de URI wordt als minder veilig ervaren. Minder gemakkelijk om te beveiligen op basis van rollen. Wijziging van de classificatie betekent aanpassen van de URI. Dit kan leiden tot een minder stabiele oplossing.

Voorbeelden met beveiligingskenmerk:

- <https://<Autoriteit>/openbaar/<overige URI onderdelen>>
- <https://<Autoriteit>/intern/<overige URI onderdelen>>

Voorbeeld zonder beveiligingskenmerk:

- <https://<Autoriteit>/<overige URI onderdelen>>

We kiezen hier voor optie 1:

Beveiligingskenmerken van API's niet in de URI opnemen

Motivatie:

Gebruikers kunnen zelf achterhalen wat hun rechten zijn bij het benaderen van de API. Het toevoegen van een beveiligingskenmerk maakt de API zelf ook niet veilig, maar er is wel een kans dat dit kenmerk in de loop van tijd verandert.

In de praktijk wordt in beide gevallen (zowel optie 1 als 2) de API's beveiligd door bijvoorbeeld een API gateway die per resource rechten kan toekennen (buiten scope van de URI-strategie).

3.4.2 Wijze van ontsluiten (techniek)

Functionaliteiten kunnen binnen een gekozen <Autoriteit> (=domeinnaam) op verschillende technische wijzen worden aangeboden. Voorbeelden hiervan zijn REST API's, SOAP webservices,

Linked data, etc. Om dit onderscheid aan te duiden zijn er verschillende mogelijkheden:

1. Neem de wijze van ontsluiten op in het onderdeel <Autoriteit>. Voorbeeld:
https://<subdomein-aanduidingen> . **api** . <hoofdomein van Autoriteit>/<overige Pad-kenmerken>
2. Neem de wijze van ontsluiten op in het onderdeel <Pad>. Voorbeeld:
https://<Autoriteit>/<andere Pad-kenmerken>/**api**/<overige Pad-kenmerken>

De voor- en nadelen van beide opties:

Optie	Voordelen	Nadelen
1. Neem de wijze van ontsluiten van functionaliteit op als kenmerk van de <Autoriteit> (in de vorm van een subdomeinnaam).	Wijze van ontsluiten van services is direct duidelijk vanuit de domeinnaam. Maakt het mogelijk om één resource op verschillende wijzen te ontsluiten. Routing door de gateway naar de backend is eenvoudiger.	Er dient overeenstemming te worden gevonden over het ontsluiten van een nieuw type service binnen het domein.
2. Neem de wijze van ontsluiten van de functionaliteit op in het <Pad> (het resource-gedeelte)	Domeinnaam blijft altijd gelijk.	Wijze van ontsluiten van services is niet direct duidelijk vanuit de (sub) domeinnaam.

We kiezen hier voor optie 1:

Wijze van ontsluiten toevoegen als subdomein in het onderdeel <Autoriteit>

Motivatie:

Deze keuze maakt het mogelijk om verschillende protocollen naast elkaar te ondersteunen voor dezelfde resources.

Voorbeelden:

- api
- service

Opmerking: alhoewel de scope van dit document alleen de URI's betreft m.b.t. de API ontwerprichtlijnen [2] voorziet de hier voorgestelde naamconventie ook in webservices. Dit wil echter niet zeggen dat instemming met deze conventie betekent dat alle bestaande webservices een andere URI moeten krijgen. Voorlopig zal dus alleen de term "api" gebruikt worden in het domein gedeelte.

3.4.3 Omgevingsaanduiding

In navolging van de DSO gebruiken we ook een omgevingsaanduiding. We gebruiken hiervoor de bestaande afspraken binnen EDSN.

Voorgesteld wordt om deze als volgt in te vullen:

Omgeving	Afkorting
Productie	prd
Acceptatie	acc
Test	tst
Ontwikkel	dev

Intern kan de aanduiding “prd” gebruikt worden maar volgens goed gebruik laten we hierbij de aanduiding “prd” achterwege voor onze externe gebruikers (niemand surft immers naar <https://www.prd.google.com>).

3.4.4 Routeerfunctie

Bij het stelsel omgevingswet van de overheid [**Error! Reference source not found.**] is het idee dat er één landelijke ingang is voor omgevingswet-vragen en dat deze gerouteerd worden naar de juiste gemeente/provincie/waterschap etc. Om dit mogelijk te maken worden er vaste URL's gebruikt die aan de achterkant de vraag doorzetten naar de relevante overheidsinstelling. Deze instelling zorgt dan voor verdere afhandeling.

Binnen de energiesector is geografische ligging veel minder bepalend zoals bij de omgevingswet. Wat wel bepalend is, is dat bepaalde centrale, landelijke systemen nu onderhouden worden door verschillende organisaties (bijv.: CPS bij TenneT, CMF-diensten bij EDSN) en dat er zonder routeerfunctie geen mogelijkheid bestaat om deze diensten organisatie-onafhankelijk te adresseren.

Het voordeel van een routeerfunctie binnen de naamgeving van de energiesector is dus dat er geen afhankelijkheid meer zal zijn met namen van bedrijfsonderdelen of uitvoerende organen. Als hier iets in verandert, hoeft alleen het eindpunt van de routing aangepast te worden, de dienst zal door de afnemer nog steeds d.m.v. dezelfde URI opvraagbaar zijn.

Er zijn 2 methodes om te routeerfunctie op te nemen in de URI:

1. opnemen in het onderdeel <Autoriteit> (het domeingedeelte)
2. opnemen in het onderdeel <Pad> (het resource-gedeelte)

De voor- en nadelen van beide opties:

Optie	Voordelen	Nadelen
1. opnemen in het onderdeel <Autoriteit> (het domeingedeelte)	<p>Locaties van diensten zijn onafhankelijk van organisatienamen zoals liander.nl, edsn.nl</p> <p>Ondersteunt routeren op basis van DNS-naam</p> <p>Eenvoudig en goedkoop; voor aanpassing in routing is slechts een wijziging in DNS-records nodig</p> <p>Opvoeren in DNS dwingt unieke namen af voor de hele sector</p>	<p>Er is een generieke autoriteitsnaam nodig</p> <p>Een centrale organisatie in de energiesector zal verantwoordelijk moeten zijn voor het bijhouden van de DNS-records (anders kans op wildgroei).</p> <p>Decentrale organisaties zullen waarschijnlijk niet de controle over de DNS records uit handen willen geven aan de centrale organisatie.</p>
2. opnemen in het onderdeel <Pad> (het resource-gedeelte)	Eenduidigheid: er is maar één domeinnaam voor de hele energiesector	Routing vereist een eigen routeringsinfrastructuur en onderhoud

We kiezen hier voor optie 1:

Routeer-informatie opnemen in het onderdeel <Autoriteit>

Motivatie:

Het routeren op basis van een subdomein is een eenvoudige manier om verkeer te kunnen routeren, terwijl gebruikgemaakt wordt van één vaste (hoofd)domeinnaam.

Voor de invulling van een subdomein zijn er verschillende mogelijkheden:

1. subdomein kiezen op basis van bedrijfsfunctie
2. subdomein kiezen op basis van organisatie
3. subdomein kiezen op basis van bestaande hubs of portalen
4. subdomein kiezen op basis van toekomstige hubs of portalen
5. subdomein kiezen op basis van register

De voor- en nadelen van de verschillende opties:

Optie	Voordelen	Nadelen
1. subdomein kiezen op basis van bedrijfsfunctie	<p>Zeer langdurig stabiel. De kans dat een bedrijfsfunctie in de energiesector verandert is klein en als de functies vervallen is de dienst</p>	<p>Veel impact, vooral bij de afnemers van energie-diensten: alle URI's zullen wijzigen (mitigatie: sterfhuiscconstructie, oud en nieuw naast elkaar laten bestaan)</p>

Optie	Voordelen	Nadelen
	waarschijnlijk ook niet meer nodig.	
2. subdomein kiezen op basis van organisatie	Herkenbaar wie de dienst aanbiedt Weinig impact: sluit goed aan bij huidige situatie	Niet langdurig stabiel. Door fusies of veranderende wetgeving kunnen en zullen organisatienamen veranderen
3. subdomein kiezen op basis van bestaande hubs of portalen	Weinig impact en kosten Kan direct ingevoerd worden	Minder langdurig stabiel dan bij optie 1
4. subdomein kiezen op basis van toekomstige hubs of portalen	Toekomst hubs is onzeker Kans op rework en dus verspilling	Minder langdurig stabiel dan bij optie 1
5. subdomein kiezen op basis van register; Voorbeeld: Het centraal aansluit register	Weinig impact en kosten Kan direct ingevoerd worden Registers worden vaak aangeboden door 1 partij	Minder langdurig stabiel dan bij optie 1 Resources kunnen van register veranderen

We kiezen hier voor optie 3:

Routeren op basis van bestaande hubs of portalen

Motivatie:

Deze keuze is goed langdurig stabiel, tegen geringe impact en kosten. Optie 1 is nog meer langdurig stabiel, maar dit betekent o.a. het opstellen van een uitputtende lijst van bedrijfsfuncties en het toepassen daarvan in URI's, en bestaande webservices.

Overzicht van centrale hubs of portalen:

Onderwerp hub (NL)	Onderwerp hub (EN)	Afkorting	Doelsysteem
Groothandel elektriciteit	Wholesale electricity	wse	routeert naar CPS (Tennet, CPS wordt opgevolgd door MMC)
Centrale Allocatie Reconciliatie en Meetdata	Allocation Reconciliation and Measurementdata	arm	routeert naar C-ARM (EDSN)
Groothandel gas	Wholesale Gas	wsg	routeert naar Nexus (EDSN)
Centrale marktfacilitering	Central Market Facilitation	cmf	routeert naar C-AR/Portaal en het wendbare (cloud) landschap (EDSN)

3.4.5 Hoofddomeinnaam

Bij het kiezen van de hoofddomeinnaam van de <Autoriteit> zijn de volgende opties in beschouwing genomen:

1. De domeinnaam van de API aanbieder; bv in geval van MFF-BAS: “mffbas.nl”
2. Een organisatie specifieke landelijke domeinnaam voor alle API’s zoals bijv. “edsn.nl” of “netbeheernederland.nl”
3. Een nieuw landelijk te kiezen generieke domeinnaam zoals bijv. “netbeheer.nl” of “nutssector.nl”
4. Internationale persistente domeinnamen zoals “purl.org” of “w3id.org”

De volgende tabel geeft de onderbouwing voor de keuze tussen de verschillende alternatieven weer:

Optie	Voordelen	Nadelen
1. Domeinnaam van de API aanbieder	Herkenbaar voor buitenstaanders wat de bron/aanbieder is (welk bedrijfsonderdeel) en wie de verantwoordelijkheid heeft	API’s die dezelfde functionaliteit bieden kunnen over tijd door meerdere partijen worden aangeboden. Organisatienamen kunnen veranderen door het samenvoegen of splitsen van bedrijven.
2. Een organisatie specifieke landelijke domeinnaam voor alle API’s.	Eén centraal punt voor API toegang.	Landelijke organisatienamen kunnen veranderen door het samenvoegen of splitsen van de organisaties. Minder herkenbaar wie de aanbieder is.
3. Te kiezen landelijke generieke domeinnaam die los staat van een organisatie	Bij het samenvoegen of splitsen van netbeheerders of andere sector deelnemers zal de URI gelijk blijven	Er zijn afspraken en afstemming nodig hoe deelnemende partijen informatie zullen publiceren.
4. Internationale domeinnaam zoals: https://w3id.org/netbeheer/def/EStationcomplex/	Bij het samenvoegen of splitsen van netbeheerders zal de URI gelijk blijven	Afhankelijkheid van de aanbieder (minder controle dan een eigen domeinnaam).

We kiezen hier voor optie 3:

Toepassen van een landelijk generieke domeinnaam: ***energysector.nl***
(domein is door EDSN gereserveerd)

Motivatie:

Met deze keuze is de URI het meest persistent en weerbaar voor organisatieveranderingen. Aangezien (veel) functionaliteit landelijk wordt aangeboden kan er per project/initiatief afstemming worden gezocht over de exacte functionaliteit en wijze van aanbieden.
Ter info: het domein “energysector.nl” was al geregistreerd door Eneco.

De gemaakte keuzes in de paragrafen 3.4.1 t/m 3.4.5 resulteren in de volgende opbouw van het onderdeel <Autoriteit>:

`https:// api . <omgeving> . < routeerfunctie > . energysector . nl`

Voorbeelden van het onderdeel <Autoriteit>:

- api.acc.cmf.energysector.nl
- api.cmf.energysector.nl (productieomgeving, komt voort uit api.prd.cmf.energysector.nl)

3.5 Onderdeel <Pad> (resource-gedeelte)

Resources kunnen plat of hiërarchisch ingedeeld worden. Het hiërarchisch indelen geeft echter direct inzicht over de context waarin de resource geplaatst is. We kiezen dus voor een hiërarchische indeling met meerdere niveaus.

3.5.1 Hoofdniveau

Voor het invullen van de hoofdniveaus zijn er meerdere opties:

1. op basis van het bedrijfs onderdeel dat de resources beheert
2. op basis van de afdeling die de resources beheert
3. op basis van het team dat de resources ontsluit
4. op basis van de bedrijfsfunctie (zoals Bedrijfsvoeren, Asset-onderhoud, etc.) waarin de resources gebruikt worden
5. op basis van bedrijfsobjecten uit NBility
6. op basis van zogenaamde 'Bounded Contexts'
7. op basis van Waardenstromen of ART's van EDSN

Optie	Voordelen	Nadelen
1. Bedrijfs onderdeel	Duidelijk voor de afnemer wie de API aanbiedt.	Bedrijfs onderdelen zijn aan verandering onderhevig. Wellicht dat we ons in de toekomst gaan organiseren op basis van bijv. ketens. Bedrijfs onderdelen per bedrijf kunnen verschillen en zijn wellicht niet bekend voor de consument.
2. Afdelingen	Duidelijk voor de afnemer wie de API aanbiedt.	Afdelingen veranderen geregeld. Afdelingen per bedrijf kunnen verschillen en zijn wellicht niet bekend voor de consument.
3. Teams	Duidelijk voor de afnemer wie de API aanbiedt. Zeer herkenbaar voor "lokale" consumenten.	Teams veranderen geregeld. Teams per bedrijf kunnen verschillen en zijn wellicht niet bekend voor de consument.
4. Bedrijfsfunctie	Onafhankelijk van de organisatiestructuur van de	Verantwoordelijke is niet direct te achterhalen.

Optie	Voordelen	Nadelen
	<p>aanbiedende organisatie.</p> <p>De bedrijfsfuncties van de onderliggende partijen zijn erg stabiel en geven enige hint waar de functie zich kan bevinden in het bedrijf.</p>	<p>Een functie past niet goed op de gewenste resource-benadering</p> <p>Er zijn veel bedrijfsfuncties, gebruik hiervan levert veel entries op</p>
5. Bedrijfsobject (niveau 1 uit het BOM-model van NBility)	<p>Bedrijfsobjecten zijn inherent geschikt om te dienen als resource.</p> <p>Bedrijfsobjecten zijn erg stabiel in de tijd.</p> <p>Bedrijfsobjecten zijn herkenbaar voor alle partijen.</p> <p>Bedrijfsobjecten zijn beschreven en afgestemd met de drie grootste regionale netbeheerders in het capability-model (NBility [7])</p>	<p>Discussies over scope en granulariteit van bedrijfsobjecten.</p>
6. Bounded Context	<p>Bounded Contexts zorgen voor een goede afbakening binnen de organisatie</p> <p>Indien goed gekozen, zijn BC's erg stabiel in de tijd</p>	<p>BC's zijn minder goed herkenbaar voor externe partijen</p>
7. Waardestroom	<p>Juist gekozen waardestromen zorgen voor onafhankelijkheid van de ontwikkelteams onderling. Dit bevordert snellere ontwikkeltijden en eenvoudiger onderhoud.</p>	<p>Het is moeilijk om waardestromen goed te kiezen</p> <p>Momenteel worden veranderingen in de waardestromen verwacht, waarbij er meerdere teams tot 1 waardestroom zullen gaan behoren. Verwachting is dat hierdoor een indeling gebaseerd op waardestroom te grof zal zijn om als hoofdniveau te dienen.</p>

We kiezen hier voor optie 5:

Gebruik een NBility bedrijfsobject (niveau 1) voor het hoofdniveau van de <Resource>

Motivatie:

Bedrijfsobjecten zijn het meest stabiel en geven context aan de te ontwikkelen API's en gebruikte begrippen. Daarnaast is men hierdoor het beste voorbereid op de toekomst wanneer er eventuele

nieuwe bedrijfsobjecten gedurende de energietransitie ontstaan.

Met de komst van NBility [7] in september 2021 hanteren de gezamenlijke netbeheerders dezelfde benamingen voor hun bedrijfsobjecten. Volgens afspraak hanteren we voor de toepassing in URI's steeds de Engelse vertalingen van deze bedrijfsobjecten. Omdat bedrijfsobjecten een sterke relatie hebben met bedrijfsfuncties en -processen die op deze bedrijfsobjecten inwerken, worden deze in NBility als zogenaamde capabilities beschreven. Een capability is afgeleid uit bedrijfsdoelstellingen en representeert al het benodigde om de gestelde bedrijfsresultaten te behalen. Simpel gezegd: in de context van URI naamgeving omvat een capability de CRUD (Create, Read, Update en Delete) operaties op een bedrijfsobject.

Dit zijn de vastgestelde namen voor de bedrijfsobjecten op niveau 1 binnen NBility:

Aanduiding in NBility	BOM niveau 1 (EN)
1.	customer
2.	energy-transport
3.	energy- grid
4.	measurement
5.	work
6.	energy-market

3.5.2 Eerste subniveau binnen <Pad>

Er zijn twee subniveaus onder het hoofdniveau om de resources verder in te delen. Het eerste subniveau bestaat uit de namen voor bedrijfsobjecten op NBility niveau 2:

Aanduiding in NBility	BOM niveau 2 (EN)
1.1.	customer
1.2.	customer-agreement
1.3.	customer-settlement
1.4.	customer-case
2.1.	transport-control
2.2.	transport-deviation
2.3	transport-planning
3.1.	energy-grid-change *
3.2.	energy-grid-quality
3.3.	energy-grid-policy
4.1. + 4.2.	measurement
5.1.	work-portfolio
5.2.	work-order
5.3.	work-activity
5.4.	work-resource
6.1.	market-participant
6.2.	connection
6.3.	market-process
6.4	data-provision

* kleine afwijking t.o.v. NBility, NBility v2.0 benoemt 3.1 als "Grid change" i.p.v. het meer consistente "Energy grid change".

3.5.3 Tweede subniveau binnen <Pad>

Het tweede subniveau bestaat uit de namen voor bedrijfsobjecten op niveau 3. Omdat deze lijst vrij

lang is, wordt hier volstaan met de objecten uit de voor CMF meest relevante groep “energy-market” (groep 6). Voor de overige groepen wordt verwezen naar NBility [7].

Aanduiding in NBility	BOM niveau 3 (EN)
6.1.1.	market-participant
6.1.2.	market-case
6.1.3.	market-agreement
6.1.4.	energy-market
6.2.1.	connection
6.2.2	facility
6.2.3	congestion-area
6.3.1	market-request
6.3.2	energy-schedule *
6.3.3	energy-exchange
6.3.4	market-invoice
6.3.5	regulation-demand *
6.3.6	regulation-offer *
6.4.3	consent-from-system-user

* kleine afwijking t.o.v. NBility. NBility geeft voor deze niveau’s twee benamingen als invulling van het niveau. Omwille van eenduidigheid is hier gekozen voor de best bruikbare optie m.b.t. naamgeving.

3.5.4 Naam API binnen <Pad>

Dit niveau representeert alle bij elkaar horende resources en de bijbehorende business-functie. De naam dient zo generiek mogelijk te zijn om later de API uit te kunnen breiden, zonder een nieuwe API te hoeven benoemen. Als leidraad kan uitgegaan worden van een samentrekking van het NBility object op niveau 3 en de capability op niveau 3.

Niv.	Capability niveau 3	BOM niveau 3 (EN)	<naam API> (EN) = samentrekking BOM+capability niveau 3
6.1.1	Relatie met marktpartijen onderhouden	market-participant	market-participant-management
6.1.2	Vragen van marktpartijen afhandelen	market-case	market-case-management
6.1.3	Contracten voor marktdiensten afsluiten	market-agreement	contract-management
6.1.4	Energiemarkten beheren	energy-market	energy-market-management
6.2.1	Aansluitingen/allocatiepunten beheren	connection	connection-management
6.2.2	Installaties achter aansluitingen beheren	facility	facility-management
6.2.3	Congestiegebieden beheren	congestion-area	congestion-area-management
6.3.1	Marktprocedures uitvoeren	market-request	market-request-management
6.3.2	Energieplanning uitwisselen	energy-schedule	energy-schedule-communication
6.3.3	Energie-uitwisseling vaststellen	energy-exchange	energy-exchange-quantification
6.3.4	Energie-uitwisseling toewijzen aan marktpartijen	market-invoice	market-invoice-assignment
6.3.5	Opregel- en afregelbehoefte communiceren	regulation-demand	regulation-demand-notification
6.3.6	Opregel- en afregelaanbod beheren	regulation-offer	regulation-offer-management
6.4.3	Toestemmingen beheren	consent-from-system-user	user-consent-management

De samentrekkingen in de laatste kolom zijn geen officieel onderdeel van het NBility model.

Om toekomstige naamgevingen te vergemakkelijken en om naamgeving zo uniform mogelijk uit te voeren, volgt hier een lijstje met Engelse sleutelwoorden, die leidend zijn voor de API-naam:

Sleutelwoord (EN)	Toepassing, betekenis
management	als de API alle stadia van de levenscyclus van een resource beïnvloedt (alle CRUD-operaties)
retrieval	als de API een opvragend karakter heeft
assignment	als de API iets toewijst aan een partij
quantification	Als de API iets cijfermatig vaststelt
notification	als de API enkel informatie publiceert
communication	als de API bi-directionele uitwisseling van informatie betreft, waarbij geen sprake is van mutatie van het gerelateerde bedrijfsobject (vraag-antwoord)

In voorkomende gevallen kunnen ook andere API-namen toegepast worden, als de standaard indeling volgens NBility niet toereikend is. Dit is o.a. het geval als:

- niveau 3 nog te grofmazig is. Bijvoorbeeld: de verdere onderverdeling van Power Generating Modules in Type A en Type BCD
- als de API meerdere resources ontsluit die in NBility onder verschillende objecten zijn ingedeeld. Bijvoorbeeld: de EANcodeboek-API bevat zowel de netbeheerdersnaam (object 6.1.1 market-participant) als de EAN18 van de aansluiting (object 6.2.1 connection)

3.5.5 Versie binnen <Pad>

Het URI-kenmerk <versie> specificeert het versienummer van de API. Dit kenmerk is opgebouwd uit de letter "v" gevolgd door een nummer. Dit nummer is het major versienummer van de API. Het versienummer begint bij 1 en wordt met 1 opgehoogd voor elke major release waar het koppelvlak wijzigt (en backward compatibiliteit niet wordt gegarandeerd). Minor versie-nummers staan in het bericht zelf. Het toevoegen van een niet-verplicht attribuut aan de payload is een voorbeeld van een wijziging die backward compatible is.

3.5.6 Collectie binnen <Pad>

De onderverdeling naar collecties is nodig om de verschillende resources die tot één API behoren, van elkaar te onderscheiden. Dit is het laagste niveau van resources. In navolging van anderen, zoals bv Google, wordt dit een collectie [8] genoemd. Het URI-component <Collectie> identificeert de collectie-resources waaruit gegevens worden teruggegeven. Een collectie is vergelijkbaar met de classes in object oriëntatie en de objecten zijn de daadwerkelijke resources.

De gemaakte keuzes in de paragrafen 3.5.1 t/m 3.5.6 resulteren in de volgende opbouw van het onderdeel <Pad>:

```
/<hoofdniveau>/<1° subniveau>/<2° subniveau>/<naam API>/<versie>/<collectie>
```

Voor de invulling van de resources van een collectie kan het beste uitgegaan worden van reeds bestaande standaarden of modellen. Deze invulling zal altijd in de meervoudsvorm gesteld worden.

Voor het bepalen van de namen van collecties of resources zijn er meerdere opties mogelijk:

1. Definities uit de Nederlandse/Europese wetgeving
2. Definities uit het CIM
3. Definities uit NBility
4. Zelf vaststellen door een afvaardiging van netbeheerders en EDSN

Optie	Voordelen	Nadelen
1. Definities uit de Nederlandse wetgeving	Sluit het beste aan bij de Nederlandse situatie	Veelal gebaseerd op de Nederlandse taal
2. Definities uit het CIM	Sluit het beste aan bij internationale toepassingen	Definities kunnen afwijken van de Nederlandse situatie, wat verwarrend kan werken
3. Definities uit NBility	Sluit het beste aan bij de capabilities van de netbeheerders. Bij toenemend gebruik van NBility neemt de herkenbaarheid en positionering in het geheel, toe	NBility is niet bedoeld om te voorzien in alle situaties (zoals bijv. samengestelde API's waar resources uit meerdere domeinen beschikbaar worden gesteld)
4. Zelf vaststellen door een afvaardiging van netbeheerders en EDSN	Kan altijd toegepast worden, bijv. als standaarden of modellen hierin niet (kunnen) voorzien	Vereist extra definitie/toelichting omdat niet naar standaarden of modellen verwezen kan worden

Omdat resources in heel veel verschillende contexten voorkomen en de betekenis ervan afhankelijk is van de context waar ze gebruikt worden, wordt hier geen keuze gemaakt voor één van de opties, maar een voorkeur van toepassen in de gegeven volgorde.

Bij het benoemen van resources prevaleert dus de Nederlandse wetgeving boven de definities uit de CIM-standaard, die weer prevaleren boven de business objecten uit NBility.

Als laatste optie kan gekozen worden om zelf een resource-naam te kiezen (door een afvaardiging van netbeheerders en EDSN) als de modellen/standaarden hier niet in (kunnen) voorzien. Deze naam zal dan toegelicht moeten worden voor de gebruikers.

De naam van een resource zal vastgesteld moeten worden door een afvaardiging van netbeheerders en EDSN. Momenteel (begin 2023) ligt die taak bij de WG API Strategie, die behalve de naam ook de specificatie van API's t.b.v. het CMF-domein, controleert op het correct toepassen van de API en URI richtlijnen.

3.6 Onderdeel <Query>

Het URI-onderdeel <Query> specificeert optioneel aanvullende parameters die als filter op het resultaat van de API-aanroep werken.

Mogelijke toepassingen van query's zijn:

- *Filtering/Zoeken*: wanneer de gebruiker van de API geïnteresseerd is in een deel van de dataset kan dit d.m.v. een query worden bewerkstelligd. Dit is vergelijkbaar met een where-statement in een SQL-query.

Vanuit de DSO nemen we de beperking over om maximaal 5 query parameters per call te specificeren. Indien er meer dan 5 nodig zijn, moet een POST gebruikt worden.

- *Paginatie/Sortering*: zie afspraken hierover in MFF-BAS API ontwerprichtlijnen [2].

3.7 Onderdeel <Fragment>

Het URI-onderdeel <Fragment> specificeert optioneel een verwijzing naar de identificatie van een resource binnen de collectie conform het uitgangspunt: hergebruik de identificatie-mechanismes waar mogelijk zoals in de referentiemodellen beschreven.

4 Resulterende opbouw van een energysector-URI

4.1 Naamconventie energysector-URI

Alle in hoofdstuk 3 genoemde onderdelen samenvoegend, levert de volgende naamconventie op:

```
https:// api . <omgeving> . <routing> . energysector . nl /<hoofdniveau>/<1° subniveau>/<2° subniveau>/<naam API>/<Versie>/<Collectie>/<ID}<?<Query>#<Fragment>
```

4.2 Voorbeelden energysector-URI's

Als eerste voorbeeld benoemen we een API waarmee contracten voor marktdiensten beheerd kunnen worden. Het gaat hier om contracten voor balanceringscapaciteit, redispatch capaciteit of vraagsturing.

Wijze van ontsluiten: "api" (geldt voor alle API's)
 Omgeving: "prd" (productie omgeving, dus weglaten voor afnemers)
 Routing naar dienst: "cmf" (cloud applicatie in het landschap van EDSN)
 Hoofdniveau: "energy-market" (NBility BOM niveau 1)
 1° subniveau: "market-participant" (NBility BOM niveau 2)
 2° subniveau: "market-agreement" (NBility BOM niveau 3)
 Naam API: "contract-management" (samentrekking van capability en BOM niveau 3)
 Versie: v1
 Collectie: "market-agreements" (conform CIM en NBility 6.1.3.)
 Resource identifier: 879999999999999999

Een URI op basis van bovenstaand voorbeeld wordt dan als volgt gevormd:

<https://api.cmf.energysector.nl/energy-market/market-participant/market-agreement/contract-management/v1/market-agreements/87999999999999999999>

Andere praktijkvoorbeelden:

BOM nive 1 (EN)	BOM niveau 2 (EN)	BOM niveau 3 (EN)	<naam API> (EN) = samentrekking BOM+capability niveau 3	Collectie (EN)
energy-market	market-participant	market-case	market-case-management	tickets
energy-market	market-participant	market-agreement	contract-management	market-agreements
energy-market	connection	connection	eancode-retrieval *	ecbinfoset
energy-market	connection	facility	pgm-type-bcd-management *	power-generating-modules
energy-market	market-process	energy-exchange	profile-fractions-retrieval *	profile-fractions
energy-market	market-process	energy-exchange	conversion-factors-retrieval *	gridarea-calorific-conversion-factors
energy-market	market-process	energy-exchange	readings-retrieval *	24months-readings

* samenstelling wijkt af van de standaard afleiding uit paragraaf 3.5.4