

NEDU API URI richtlijnen

Titel : NEDU API URI richtlijnen
Project :
Datum : 21 december 2020
Versie : 1.0
Status : Vastgesteld ALV NEDU
Opdrachtgever : NEDU Technische Commissie
:

Inhoudsopgave

Colofon	3
Versiegeschiedenis	3
Verspreiding.....	3
Gerefereerde documenten	3
1 Inleiding.....	5
2 Algemene uitgangspunten	7
3 Naamgeving URI's	9
3.1 Inleiding naamgevingen.....	9
3.2 Verplichte opbouw van URI's.....	9
3.3 Onderdeel <Protocol>.....	9
3.4 Beveiligingskenmerken	10
3.5 Routeerfunctie	11
3.6 Wijze van ontsluiten (techniek)	13
3.7 Onderdeel <Autoriteit> (domeinnaam).....	14
3.8 Primaire resource groep	16
3.9 Secundaire resource groepen.....	18
3.10 Versie	18
3.11 Onderdeel <Pad>	18
3.12 Onderdeel <Query>	18
3.13 Onderdeel <Fragment>.....	19
4 Resulterende opbouw van een energie-URI.....	20
4.1 Naamconventie energie-URI.....	20
4.2 Voorbeeld energie-URI	20

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 december verwerkt
0.9	Ter vaststelling	Laatste feedback verwerkt
0.91	Ter vaststelling	Review SI
0.92	Ter vaststelling	Query paragraaf aangepast
1.0	Vastgesteld	

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	21-12-2020	1.0

Gerefereerde documenten

Nummer	Omschrijving	Datum	Versie	Auteur
1.	API strategie voor de Nederlandse overheid:			

	API Strategie Algemeen (Nederlandse API Strategie I)	04-02-2020	Vastgesteld	Geonovum e.a.
	API Designrules (Nederlandse API Strategie IIa)	17-01-2020	Ter vaststelling	Geonovum e.a.
	API Designrules Extensions (Nederlandse API Strategie IIb)	17-01-2020	Consultatie-versie	Geonovum e.a.
2.	NEDU API ontwerprichtlijnen	01-10-2020	0.95	NEDU API werkgroep
3.	https://www.forumstandaardisatie.nl/open-standaarden/lijst	n.v.t.	n.v.t.	Bureau Forum Standaardisatie
4.	https://aandeslagmetdeomgevingswet.nl/digitaal-stelsel/technisch-aansluiten/standaarden/api-uri-strategie/	12 maart 2018	1.1	Deelprogramma DSO
5.	https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf	2000	n.v.t.	Roy Fielding
6.	https://www.ietf.org/rfc/rfc3986.txt	Januari 2005	Internet Standard	IETF
7.	https://cloud.google.com/apis/design/resources#what_is_a_rest_api	n.v.t.	n.v.t.	Google
8.	Referentie Architectuur	6 juli 2018	2.0	AG CMF

1 Inleiding

Dit document hoort bij het document NEDU API ontwerprichtlijnen [2]. Alhoewel deze laatste is gebaseerd op de API strategie voor de Nederlandse Overheid [1], voorziet deze niet in een URI-strategie. Daarom is de keuze gemaakt door de WG API Strategie om zelf te voorzien in afspraken omtrent het gebruik en toepassen van URI's. Het resultaat daarvan is het voorliggende document. Om API's opvraagbaar te maken is het nodig om deze een digitaal adres te geven. Binnen de context van internet worden deze adressen URI's genoemd waarbij de meest voorkomende vorm een URL is, ook wel bekend als een link. Er wordt 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 URI strategie is gebaseerd op de volgende bronnen¹:

- 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 en EDSN.

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

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

Aangezien deze URI strategie zich richt op API gebruik, beperken 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 3.6).

Waarom 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 structuuropbouw uit te werken.

Wat is een URI/URN/URL/resource? [3]

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

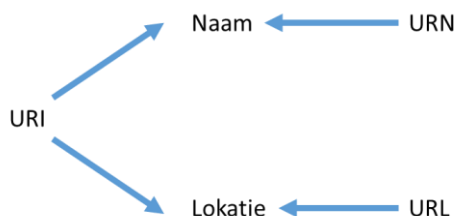
¹ Delen van deze documenten zijn overgenomen op deze pagina.

Een “resource” kan elk mogelijk ding zijn, zoals fysieke objecten (transformator, kabel, geografische locatie), abstracties (begrip, model) alsook informatie-elementen (webpagina). De afkortingen URN’s, URI’s en URL’s bevatten allen de term “Resource”.

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 het 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 locatie specificeert is een URL.

Een URI die een naam specificeert is een URN.

Een URI die een naam en een locatie specificeert 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
- <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 inhoudelijk toepassing ervan. Voor verdere informatie over het toepassen van queries in API's verwijzen we naar de NEDU API ontwerprichtlijnen [2].

2 Algemene uitgangspunten

REST API's zijn populair omwille van hun eenvoud en hergebruik van het http(s) protocol die zoals gezegd een subset zijn van de hiervoor besproken generieke URI'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)	MeteringPoints, EndDevices, MeteringData

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 / 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)	Een voorbeeld van een slechte URI: https://pnp-webdisp.bedrijf.local:8190/iri/portal/urenverantwoording geeft de webapplicatie voor het schrijven van uren. Deze URI is, naast het feit dat hij niet gemakkelijk te lezen en te onthouden is, erg applicatie specifiek 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. Bedrijven kunnen fuseren of nieuwe namen krijgen. Een langdurige stabiele URI voorziet in dit soort scenario's.

ID	03
Uitgangspunt	Centrale URI's worden organisatie onafhankelijk benoemd
Implicatie(s)	Lokale diensten zijn verantwoordelijkheid van de organisatie die deze dienst aanbiedt. Er is een generieke autoriteitsnaam nodig. Een centrale organisatie in de energiesector zal verantwoordelijk moeten zijn voor het bijhouden van de DNS-records (anders kans op wildgroei).
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 diensten die in scope zijn: CMF, CPS / MMC Hub (Tennet), etc. Decentraal: services die sector leden zelf uitvoeren zoals klantenservices, asset management, etc.

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 oorzaak hiervoor is dat een naamgeving alleen betekenis heeft voor mensen. Een naamgeving is dus volledig arbitrair. 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 de NEDU API URI volgt de internetstandaard RFC3986 [6]:

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

Onderdeel	Toelichting
<Protocol >	Geeft het protocol aan waarmee de URI opgevraagd kan worden, bijvoorbeeld HTTPS of FTPS.
<Autoriteit> (domeinnaam)	De autoriteit (of domeinnaam) voor API's
<Pad>	Het unieke pad dat een resource identificeert of een locatie van een resource aangeeft. Hierbij zijn de onderdelen binnen het pad gescheiden door een slash ("/"). De slash wordt gebruikt om hiërarchie aan te geven.
<Query>	Het query deel 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>

Beveiliging van API's heeft verschillende aspecten. Allereerst dient een keuze gemaakt te worden voor het Protocol (of channel).

Keuze:

voor het onderdeel <Protocol> wordt *https* gekozen

Motivatie:

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

3.4 Beveiligingskenmerken

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

	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 classificatie:

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

Voorbeeld zonder beveiligingsclassificatie:

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

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).

3.5 Routeerfunctie

Bij het stelsel omgevingswet van de overheid [1] 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 (vb: 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 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 routeren:

1. opnemen in het onderdeel <Autoriteit>
2. opnemen in het onderdeel <Pad>

De voor- en nadelen van beide opties:

	Voordelen	Nadelen
1. opnemen in het onderdeel <Autoriteit>	Locaties van diensten zijn onafhankelijk van organisatienamen zoals liander.nl, edsn.nl Ondersteunt routeren op basis van DNS-naam Eenvoudig en goedkoop; voor aanpassing in routing is slechts een wijziging in DNS-records nodig Opvoeren in DNS dwingt unieke namen af voor de hele sector	Er is een generieke autoriteitsnaam nodig Een centrale organisatie in de energiesector zal verantwoordelijk moeten zijn voor het bijhouden van de DNS-records (anders kans op wildgroei)
2. opnemen in het onderdeel <Pad>	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 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:

	Voordelen	Nadelen
1. subdomein kiezen op basis van bedrijfsfunctie	Zeer langdurig stabiel. De kans dat een bedrijfsfunctie in de energiesector verandert is klein en als de functies vervallen is de dienst waarschijnlijk ook niet meer nodig.	Veel impact, vooral bij de afnemers van energiediensten: alle URI's zullen wijzigen (mitigatie: sterfhuiscstructie, oud en nieuw naast elkaar laten bestaan)
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

	Voordelen	Nadelen
		register veranderen

We kiezen hier voor optie 3:

Routeren op basis van bestaande hubs of portalen
--

Motivatie:

Deze keuze is het meest 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 (Engels)	Afkorting	Doelsysteem
Centraal MeetData Systeem	Central Measurement Data Services	cmds	routeert naar portaal-CTS
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 (EDSN)
marktfacilitering	Market Facilitation	mf	routeert naar SIV (EDSN)

3.6 Wijze van ontsluiten (techniek)

Functionaliteiten kunnen binnen een gekozen autoriteit(domeinnaam) op verschillende technische wijzen worden aangeboden. Voorbeelden hiervan zijn OAS API's, SOAP webservices, Linked data, etc. Om hierin onderscheid te maken zijn er verschillende mogelijkheden:

1. Neem de wijze van ontsluiten van de functionaliteit op in het onderdeel <Pad>

Voorbeeld: <https://<Autoriteit>/api/<rest van het Pad>>

2. Neem de wijze van ontsluiten van functionaliteit op als onderdeel van het onderdeel <Autoriteit>

Voorbeeld: <https://api.<hoofddomein van Autoriteit>/<Pad>>

De voor- en nadelen van beide opties:

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

We kiezen hier voor optie 2:

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
- ldata

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 en Linked Data. Dit wil echter niet zeggen dat instemming met deze conventie betekent dat alle bestaande webservices een andere URI moeten krijgen.

3.7 Onderdeel <Autoriteit> (domeinnaam)

De gemaakte keuzes in de paragrafen 3.4 t/m 3.6 resulteren in de volgende opbouw van het onderdeel <Autoriteit>:

<wijze van ontsluiten> . <routeerfunctie> . <hoofddomein naam>

In deze opbouw ontbreekt nog een onderdeel wat de omgeving weergeeft van de resources (OTAP). Dit onderdeel zal vooraan in het onderdeel <Autoriteit> geplaatst worden:

<omgeving> . <wijze van ontsluiten> . <routeerfunctie> . <hoofddomein naam>

Voorgesteld wordt om dit als volgt in te vullen:

Productie	prd
Acceptatie	acc
Test	tst
Ontwikkel	dev

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

1. De domeinnaam van de API aanbieder; bv in geval van de NEDU: NEDU.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:

	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. Organisatie namen 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 organisatie namen 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	Afspraken/afstemming hoe alle partijen informatie publiceren.

	Voordelen	Nadelen
4. Internationale domeinnaam zoals: https://w3id.org/netbeheer/def/ESstationcomplex/	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. Als alternatief is nutssector.org ook geregistreerd. Ter info: energiesector.nl is geregistreerd door Eneco.

Voorbeelden van <Autoriteit>:

- prd.api.mf.energysector.nl (voor bv CERES)
- acc.service.mf.energysector.nl (voor bv RaadplegenAansluiting)

3.8 Primaire resource groep

Resources kunnen plat of hiërarchisch ingedeeld worden. In beide gevallen is er een hoofd- of primair niveau nodig die een groep resources typeert.

Primaire resources zijn groepen van resources die logisch bij elkaar horen en zich onderscheiden van andere primaire resources.

Voor het invullen van de hoofdresources zijn er meerdere opties:

1. op basis van het bedrijfsonderdeel 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

	Voordelen	Nadelen
1. Bedrijfsonderdeel	Duidelijk voor de afnemer wie de API aanbiedt.	Bedrijfsonderdelen zijn aan verandering onderhevig. Wellicht dat we ons in de toekomst gaan organiseren op basis van bijv. ketens. Bedrijfsonderdelen per bedrijf kunnen

	Voordelen	Nadelen
		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 aanbiedende organisatie. De bedrijfsfuncties van de onderliggende partijen zijn erg stabiel en geven enige hint waar de functie zich kan bevinden in het bedrijf.	Verantwoordelijke is niet direct te achterhalen. Een functie past niet goed op de gewenste resource benadering Er zijn veel bedrijfsfuncties, gebruik hiervan levert veel entries op
5. Bedrijfsobject	Bedrijfsobjecten zijn inherent geschikt om te dienen als resource. Bedrijfsobjecten zijn erg stabiel in de tijd. Bedrijfsobjecten zijn herkenbaar voor alle partijen. Bedrijfsobjecten zijn beschreven en afgestemd met de drie grootste regionale netbeheerders in de CMF (ReferentieArchitectuur [8])	Discussies over scope en granulariteit van bedrijfsobjecten. Aansluiten op een model als CIM kan dit mitigeren.

We kiezen hier voor optie 5:

Gebruik een bedrijfsobject 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 zijn we het beste voorbereid op de toekomst wanneer eventuele nieuwe

bedrijfsobjecten gedurende de energietransitie ontstaan.

Voorbeelden bedrijfsobjecten (uit de CMF ReferentieArchitectuur [8]) en hun vertaling:

- | | |
|-----------------|-------------------|
| - Aansluitingen | -> MeteringPoints |
| - Contracten | -> Agreements |
| - Meetdata | -> MeteringData |
| - Apparaten | -> EndDevices |

3.9 Secundaire resource groepen

In sommige gevallen is een verdere onderverdeling nodig omdat alleen het hoofdniveau te generiek is voor de context. Zoals bijvoorbeeld bij SysteemElementen. Zonder verdere onderverdeling worden alle soorten systeemelementen (energieleverende, energie-opnemende en neutrale netelementen) bij elkaar geplaatst. In deze gevallen kan de URI uitgebreid worden met een verdere onderverdeling. In navolging van anderen, zoals bv Google, wordt dit een collectie [7] genoemd. De collectie identificatie is een identificatie van een resourcegroep die ondergeschikt is aan de primaire resourcegroep op hoofdniveau.

Het URI-component <collectie> identificeert de collectie resources waaruit gegevens worden teruggegeven.

Dit kan een bestaande dataset zijn (bijvoorbeeld “/switches”), maar ook een collectie die dynamisch wordt opgebouwd (bijvoorbeeld “/adressen”); eventueel op basis van een opgegeven query. Een collectie is vergelijkbaar met de classes in object oriëntatie en Objecten zijn de daadwerkelijke resources.

De verdere invulling kan per specifieke dienst worden geadresseerd. Zaken zoals servernamen, organisatienamen etc. blijven verboden.

3.10 Versie

Het URI-component <versie> specificeert het versienummer van de API.

De opbouw is 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 (niet backward compatible). Minor versienummers staan in het bericht zelf. Het toevoegen van een eindpunt of een niet verplicht attribuut aan de payload zijn voorbeelden van wijzigingen die backward compatible zijn.

3.11 Onderdeel <Pad>

De gemaakte keuzes in de paragrafen 3.8 t/m 3.10 resulteren in de volgende opbouw van het onderdeel <Pad>:

`/<primaire resource groep>/<versie>/<collectie>/`

3.12 Onderdeel <Query>

Het URI-component <query> specificeert optioneel aanvullende parameters die als filter op de resultset werken.

Mogelijke toepassingen van query's zijn:

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

Voorbeeld (slimme meters in Error status):

<https://prd.api.mf.energysector.nl/endDevices/v1/Meters?Status=Error>

Paginatie: met deze query functie kan door een resultset worden gebladerd op basis van een in de query vastgestelde "pagina-grootte". Om botsingen met resource-properties te voorkomen wordt er geprefixed met een underscore ("_").

Voorbeeld (alleen de resultaten retourneren op de 2e "pagina", dus genummerd tussen 200 en 300):

https://prd.api.mf.energysector.nl/endDevices/v1/Meters?_limit=100&_page=2

Sortering: met deze query functie kan sortering op de resultset worden toegepast. Om botsingen met resource-properties te voorkomen wordt er geprefixed met een underscore ("_").

Voorbeeld (slimme meters in Error status, gesorteerd op afnemende prioriteit):

https://prd.api.mf.energysector.nl/endDevices/v1/Meters?Status=Error&_sort=priority,asc

3.13 Onderdeel <Fragment>

Het URI-component <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 referentie modellen beschreven.

4 Resulterende opbouw van een energie-URI

4.1 Naamconventie energie-URI

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

`https:// <omgeving> . <wijze van ontsluiten> . <routing> . energysector . nl
/hoofdniveau>/<versie>/<collectie>/?<query>#<fragment>`

4.2 Voorbeeld energie-URI

Als voorbeeld benoemen we een resource die een meter bij een aangeslotene representeert.

Omgeving:	prd
Wijze van ontsluiten:	api
Routing naar dienst:	mf (Wendbaar Landschap)
Hoofdniveau:	endDevices (“apparaten” uit ReferentieArchitectuur, vertaald naar CIM)
Versie:	1
Collectie:	Meters
Resource identifier:	0d03d817-f7b7-48ea-9cc1-fff7ff1c89f5

Een URL op basis van bovenstaand voorbeeld wordt als volgt gevormd:

`https://prd.api.mf.energysector.nl/endDevices/v1/Meters/0d03d817-f7b7-48ea-9cc1-fff7ff1c89f5`