

## NEDU API strategie

Titel : API strategie  
Project :  
Datum : 16-12-2020  
Versie : 1.0  
Status : Vastgesteld  
Opdrachtgever : NEDU Technische Commissie  
:

# Inhoudsopgave

Versiegeschiedenis .....	3
Verspreiding.....	4
Gereferende documenten .....	4
<b>1</b>	<b>Inleiding..... 6</b>
1.1	Leeswijzer ..... 6
<b>2</b>	<b>API technologie ..... 7</b>
2.1	Wat is een API? ..... 7
2.2	Hoe verschilt een API van een website..... 8
2.3	De spelers ..... 10
2.4	De bron ..... 10
2.5	Type API's..... 10
2.6	Waarom nu? ..... 11
2.7	GDPR ..... 13
<b>3</b>	<b>API strategie..... 14</b>
3.1	API economie ..... 14
3.2	API value chain..... 14
3.3	API Classificatie NEDU..... 16
3.4	API categorieën..... 17
3.5	API First ..... 18
3.6	NEDU en API's ..... 18

## Colofon

### Versiegeschiedenis

Versienummer	Status	Markering/wijzigingen
0.1	Concept	Initiële versie opgesteld door de NEDU TC werkgroep API strategie
0.2	Concept	Addendum toegevoegd
0.3	Concept	Review op v0.2 verwerkt
0.4	Concept	Initiële versie "NEDU API strategie en ontwerprichtlijnen" opgesplitst in twee documenten: NEDU API strategie NEDU API ontwerprichtlijnen
0.5	Concept	Reviews verwerkt
0.6	Concept	Review van werkgroepleden verwerkt. Aanbevelingen API strategie toegevoegd
0.9	Ter vaststelling	Review van werkgroepleden verwerkt. Volgende paragrafen toegevoegd o.b.v. feedback toegevoegd : <ul style="list-style-type: none"> <li>• API's als een "enabler" voor innovatie en bimodal IT</li> <li>• GDPR</li> <li>• API categorieën</li> <li>• API First</li> <li>• NEDU en API's</li> </ul> Aanbeveling API strategie weer verwijderd en deels tekst verwerkt in paragraaf API First.
1.0	Vastgesteld	Versie na vaststelling in ALV NEDU van 16 december 2020

## Verspreiding

Naam	Datum	Versie
NEDU Technische Commissie	17-12-2019	0.6
API werkgroep	30-09-2020	0.9
NEDU	01-10-2020	0.9
NEDU	16-12-2020	1.0

## Gerefereerde documenten

Nummer	Omschrijving	Datum	Versie	Auteur
1.	API strategie voor de Nederlandse overheid: <ul style="list-style-type: none"> <li>• <a href="#">API Strategie Algemeen (Nederlandse API Strategie I)</a></li> <li>• <a href="#">API Designrules (Nederlandse API Strategie IIa)</a></li> <li>• <a href="#">API Designrules Extensions (Nederlandse API Strategie IIb)</a></li> </ul>	04-02-2020 17-01-2020 17-01-2020	Vastgesteld Ter vaststelling Consultatie-versie	Geonovum e.a. Geonovum e.a. Geonovum e.a.
2.	APIs: A Strategy Guide	December 2011	First Edition	Daniel Jacobson, Greg Brail, and Dan Woods
3.	NEDU API URI richtlijnen	16-12-2020	1.0	NEDU API werkgroep
4.	NEDU API ontwerprichtlijnen	16-12-2020	1.0	NEDU API werkgroep
5.	Enterprise API Management	Juli 2019	First Edition	Zdenek Z Nemeč, Luis Weir

## Begrippen- en afkortingenlijst

Begrip	Omschrijving
API	Application Programming Interface
ASNO	API strategie voor de Nederlandse overheid
OAS	Open API Specification
REST	REpresentational State Transfer
QoS	Quality of Service specificatie
WFS	Web Feature Service
JSON	JavaScript Object Notation

# 1 Inleiding

De energiesector beschikt over een enorme schat aan data, die bijzonder maatschappelijk relevant is om de energie gerelateerde uitdagingen aan te kunnen gaan.

Als energiesector wisselen we veel data uit volgens sectorafspraken en voor wettelijk verankerde processen. Daarnaast ontstaat door de energietransitie een geheel nieuw speelveld van lastig voorspelbaar gedrag. Denk aan zon en wind bij opwekken van energie, maar ook lokale ontwikkelingen of gedrag van verbruikers en producenten van energie. Gaan we meer decentraal ondernemen en hoe snel? Gaat opslag van energie een rol spelen en wanneer? Hoe snel wordt elektrisch rijden op grote schaal ingevoerd? Wanneer gaan we gas vervangen door elektriciteit of iets anders? Niemand weet vandaag de precieze antwoorden op deze vragen. De ontwikkeling zal organisch plaatsvinden en afhangen van factoren die we vandaag misschien nog niet eens kennen.

Door deze onvoorspelbaarheid is het een uitdaging om processen en systemen te bedenken die al deze ontwikkelingen kunnen faciliteren. Wat worden eigenlijk de spelregels<sup>1</sup> binnen nieuwe systemen en processen die nog moeten ontstaan? Dat is precies wat we nog niet weten. Wat we wel weten is hoeveel energie elk moment geproduceerd en verbruikt wordt, waar onbalans of congestie dreigt te ontstaan, wie hoeveel aan wie moet betalen voor energie en transport ervan, etc. Bottom line: deze informatie is de sleutel om te kunnen reageren op alle nieuwe ontwikkelingen in de sector, mits beschikbaar voor de juiste partijen op het juiste moment en op een veilige manier.

Het is onze rol om door het delen van informatie de energietransitie te faciliteren. API (Application Programming Interface) technieken zijn bij uitstek geschikt om onze data als digitale dienst beschikbaar te kunnen stellen. Data toegankelijk maken via API's is niet gelijk aan 'open data'. Met API technieken kan data passend bij het doel en op een gestandaardiseerde manier beschikbaar gesteld worden. Ook beheer van toegang tot die data en het inzicht in wie en wanneer welke data gebruikt, is goed geregeld binnen de API technieken.

Dit document beschrijft de API strategie als een lifecycle binnen de digitale economie. De API strategie voor de energiesector is in lijn met het "niet-technische" deel van de API strategie voor de Nederlandse overheid.

## 1.1 Leeswijzer

Deze API strategie is niet gericht op de technologen die API's bouwen of rechtstreeks gebruiken, maar is voornamelijk bedoeld voor belanghebbenden in de business binnen de energiesector, dus de leden v.d. issue-commissies inclusief de achterban die verantwoordelijk is voor de decentrale ICT strategie, die strategische/tactische beslissingen moeten nemen over de vraag of een digitale dienst in de vorm van een API een goed idee is als oplossing voor een business case/issue.

Het technische deel is uitgewerkt in twee separate documenten:

- NEDU API URI richtlijnen [3];
- NEDU API ontwerprichtlijnen [4].

---

<sup>1</sup> Eerste globale set aan regels binnen de EU zijn geformuleerd in de vorm van de netwerkcodes: [https://www.entsoe.eu/network\\_codes/](https://www.entsoe.eu/network_codes/). Aanvulling daarvan op landelijk en regionaal niveau zal voor het grootste deel nog echter plaats moeten vinden.

## 2 API technologie

API's zijn erg belangrijk en ze worden steeds belangrijker. Baanbrekende bedrijven zoals Google, Facebook, Apple en Twitter hebben verbazingwekkende technologische oplossingen aan het publiek onthuld, die bestaande bedrijven transformeren en nieuwe industrieën creëren. Centraal in de successen van deze bedrijven staan de API's die mensen en hun apparatuur verbinden met de onderliggende platforms die elk bedrijf voeden en die deze bedrijven achter de schermen met elkaar verbindt.

De wereld verandert continu. Denk aan de volgende voorbeelden :

- Salesforce.com heeft een groot en rijk partner-ecosysteem opgebouwd door de primaire services voor partners open te stellen voor innovatie en uitbreiding. Tegenwoordig komt er meer verkeer via de Salesforce API dan via de website.
- Amazon ontsloot zijn kerninfrastructuur als Amazon Web Services (AWS), die toegankelijk is via een aantal API's, en bedient nu meer via AWS dan via al zijn wereldwijde “storefronts” samen.
- Twitter is het meest zichtbare voorbeeld van een bedrijf dat bijna volledig gebaseerd is op een API en een ecosysteem van ontwikkelaarsapplicaties.
- Netflix heeft volledig opnieuw uitgevonden hoe we films en tv-programma's consumeren met streaming naar honderden verschillende apparaten, waarbij niet alleen de videoverhuurindustrie werd omgegooid, maar ook grote aangrenzende markten, zoals kabeltelevisie, werden beïnvloed. Met API's kan Netflix een groot aantal apparaten op een betaalbare manier ondersteunen.
- Een voorbeeld dichterbij huis zijn de belangrijke banken die apps bieden die ontsloten worden door API's.
- Een voorbeeld binnen de energiesector is de web app energieopwek.nl (een samenwerkingsverband tussen TenneT, Netbeheer Nederland, Gasunie New Energy, Entrance en het SER-Energieakkoord) die realtime inzicht geeft in duurzame energieproductie.

Meer voorbeelden in Nederland zijn te vinden in paragraaf 2.4 van de API strategie voor de Nederlandse overheid.

### 2.1 Wat is een API?

API staat voor application programming interface en is een gestructureerd en gedocumenteerd koppelvlak voor communicatie tussen applicaties.

API's zijn niet nieuw. In feite gaat het concept al lang terug en is het al aanwezig sinds de begindagen van distributed computing. Echter, de term zoals we die vandaag de dag kennen, verwijst naar een veel moderner type API, bekend als REST of Web API's, waarmee je applicaties over het Internet kan koppelen.

Met 'API technologie' bedoelen we in deze context specifiek de REST/JSON variant. Deze wordt tegenwoordig gezien als meest geschikt om informatie via internet te ontsluiten naar verschillende

(mobiele) kanalen. Dit heeft te maken met een aantal eigenschappen, zoals :

- Toegankelijkheid voor (web) ontwikkelaars;
- Vindbaarheid;
- Testbaarheid;
- Begrijpelijkheid;
- Beveiliging;
- Beheerbaarheid;
- Bondigheid.

Zogenaamde [REST](#) API's doen voor applicaties wat websites voor mensen doen. Websites presenteren informatie aan mensen, REST APIs maken applicaties en gegevens over het Internet beschikbaar voor andere applicaties. De technologie achter websites en REST APIs heeft daarom veel gemeen.

*De term REST API's werd voor het eerst geïntroduceerd in het jaar 2000 door Roy Fielding in zijn proefschrift "Architectural Styles and the Design of Network-based Software Architectures". In zijn proefschrift presenteerde Roy Representational State Transfer (REST) als een manier om computersystemen via het internet interoperabel te maken, door optimaal gebruik te maken van het reeds beschikbare Hypertext Transfer Protocol (HTTP).*

*Voor meer informatie, zie de volgende link:*

[https://en.wikipedia.org/wiki/Representational\\_state\\_transfer#History](https://en.wikipedia.org/wiki/Representational_state_transfer#History)

De meeste applicaties gebruiken APIs onder de motorkap zonder dat de eindgebruiker dat ziet. Als je een app opent voor het weer, dan merk je niet dat deze app gegevens ophaalt bij het [KNMI via een API](#). Degene die het meest direct met een API te maken krijgt is de ontwikkelaar van de applicatie die de API gebruikt. Zij moet begrijpen hoe de API werkt en hoe je deze vanuit de applicatie moet bevragen (zie par. 2.2 v.d. API strategie voor de Nederlandse overheid).

Een API biedt dus een ingang (hook) voor collega's, partners of externe ontwikkelaars om toegang te krijgen tot gegevens en diensten om snel applicaties zoals iPhone-apps te realiseren. De Twitter en Facebook API's zijn bekende voorbeelden. Er zijn API's die openstaan voor elke ontwikkelaar, API's die alleen openstaan voor partners, en API's die intern worden gebruikt om het bedrijf beter te laten functioneren en de samenwerking tussen teams te vereenvoudigen. Een API is dus in wezen een contract. Zodra zo'n contract is afgesloten, worden ontwikkelaars verleid om de API te gebruiken, omdat ze weten dat ze erop kunnen vertrouwen. Het contract verhoogt het vertrouwen, wat het gebruik verhoogt. Het contract maakt ook de verbinding tussen aanbieder en consument veel efficiënter omdat de interfaces gedocumenteerd, consistent en voorspelbaar zijn.

## 2.2 Hoe verschilt een API van een website

Een API is heel anders dan een website. Een website geeft informatie op aanvraag en bouw je voor de eindgebruiker die hem bezoekt. Een bedrijf plaatst de inhoud op de website, en mensen consumeren het. Websites hebben geen contracten of structuren rond het gebruik van content. Als



de inhoud van de website verandert, krijgen bezoekers die daarna de website benaderen de nieuwe inhoud te zien. Dit heeft geen impact op hun browsers, en elke verandering is transparant voor de gebruiker.

Een API maak je voor ontwikkelaars die er applicaties mee bouwen. Daarom is het gewoonte om API's aan te bieden op het *developer* subdomein van een website, zoals [developer.overheid.nl](https://developer.overheid.nl) voor API's van de overheid (zie par. 2.2 v.d. API strategie voor de Nederlandse overheid).

Een API is ook anders omdat het een contract heeft en programmatuur wordt gebaseerd op dat contract. Programma's zijn, in tegenstelling tot mensen, niet flexibel en bijna altijd slecht in het matchen van patronen. Als er iets verandert aan het contract van de API, dan is het effect op de apps die er op gebaseerd zijn, potentieel vrij groot.

Een API dient behandeld te worden als een softwareproduct, rekening houdend met versiebeheer, backwards compatibiliteit en de benodigde transitieperiode om eventuele nieuwe functies te kunnen gebruiken. Er moet een evenwicht zijn tussen het ondersteunen van de bestaande consumers en tegelijkertijd het bijhouden van de noodzakelijke veranderingen, zodat de API meegroeit met de business.

Dit betekent niet dat de API nooit kan veranderen. Integendeel, een API is een online product dat bijna constant kan veranderen om aan de behoeften van de business te voldoen, of om de huidige verkeersbelasting op de meest efficiënte manier te bedienen. Maar dit zijn veranderingen in de implementatie, niet in de interface. Als dit goed wordt gedaan, kan de implementatie van een API dagelijks veranderen, of zelfs vaker, terwijl de interface consistent blijft.

**De structuur van de API maakt deel uit van het contract. Het contract is bindend en kan niet zomaar worden gewijzigd.**

### **Maar API's en websites hebben veel gemeen....**

API's, zoals websites, zijn naar verwachting 24/7, 365 dagen per jaar beschikbaar. Ontwikkelaars, zoals websitegebruikers, hebben niet veel geduld voor "geplande downtime".

Dit alles kan een uitdaging vormen voor het bouwen van een API op een bestaande enterprise technologie-infrastructuur.

Succesvolle websites kunnen, en worden voortdurend bijgewerkt. Dit is mogelijk omdat websites live entiteiten op het netwerk zijn die eenvoudig kunnen worden gewijzigd zonder dat de clients aanpassing behoeven - er is geen noodzaak om een software-update naar de gebruikers te pushen.

API's zijn in dit opzicht niet veel anders. Ervan uitgaande dat de API backwards compatibel blijft, kan een API-programma nieuwe functies introduceren en de implementatie van bestaande functies wijzigen zonder de clients te "breken". Zolang het contract tussen de API en de ontwikkelaars die het gebruiken gehandhaafd blijft, kan de API veranderen op een "web schema" in plaats van op een "enterprise IT schema". Het resultaat is een beter en sneller reagerend API-programma.

## 2.3 De spelers

De organisatie die een API aanbiedt is de API provider. Deze API strategie is grotendeels gericht op API providers in de energiesector (of degenen die overwegen een API aan te bieden).

De partij die gebruik maakt van de API om applicaties te creëren zijn de service consumers. Het is waar dat veel soorten mensen geïnteresseerd kunnen zijn in een API, maar degene die uiteindelijk de applicaties zullen realiseren zijn de ontwikkelaars in de rol van service consumers. Zij zijn het primaire publiek voor een API.

Degene die gebruik maken van de door de ontwikkelaar gerealiseerde applicaties zijn de eindgebruikers. Zij zijn het secundaire publiek voor de API en vaak het publiek dat veel van de API-beslissingen bepaalt. Afhankelijk van de inhoud die via de API beschikbaar wordt gesteld, kunnen er bijzondere aandachtspunten zijn, zoals auteursrecht, legaal gebruik, etc., die betrekking hebben op dit secundaire publiek.

## 2.4 De bron

Het fundamentele concept in elke API is de bron, in API terminologie een resource. Een resource is een object met een type, bijbehorende gegevens en eventueel de relaties met andere resources. Resources kunnen worden gegroepeerd in collecties. Enkele voorbeelden van resources zijn aansluitingen, assets (bv. meters), aangesloten, events.

## 2.5 Type API's

We zien twee soorten API's:

- Private;
- Public.

Wat je ook in de media hoort, private API's zijn de meest voorkomende variant. U weet van de Facebooks en Twitters van de wereld en hun gebruik van API's. Wat u waarschijnlijk niet weet is dat diezelfde bedrijven waarschijnlijk in uitgebreide mate gebruik maken van hun eigen API's om hun websites, mobiele apps en andere klantgerichte producten aan te sturen. Onze ervaring is dat zichtbare public API's zoals deze slechts het topje van de ijsberg zijn. Net als de grote onderwatermassa van een ijsberg zijn de meeste API's private en niet waarneembaar, intern bij bedrijven, gebruikt door medewerkers en partners met contractuele afspraken. Dit gebruik van API's is de werkelijke drijfveer achter de API-revolutie. Beperk je dus niet tot de manier waarop API's kunnen worden gebruikt voor public voorbeelden zoals de App Store. Het intern gebruik en het gebruik voor partner-integratie van API's is vaak waardevoller. Veel van de discussie over API's gaat ervan uit dat ze open moeten staan voor het publiek om van waarde te zijn. Dit is niet het geval. Wij zijn van mening dat private API's meer invloed hebben op transformaties binnen de meeste bedrijven, in veel gevallen veel meer dan public API's.

## 2.6 Waarom nu?

Als energiesector zijn er verschillende redenen om nu op API's in te zetten :

### Procesvolwassenheid

API's gaan niet alleen over technologie. Zoals in veel zakelijke problemen, hebben we echt een “mensen” probleem. API's bieden een gemeenschappelijk patroon om mensen te helpen samen te werken.

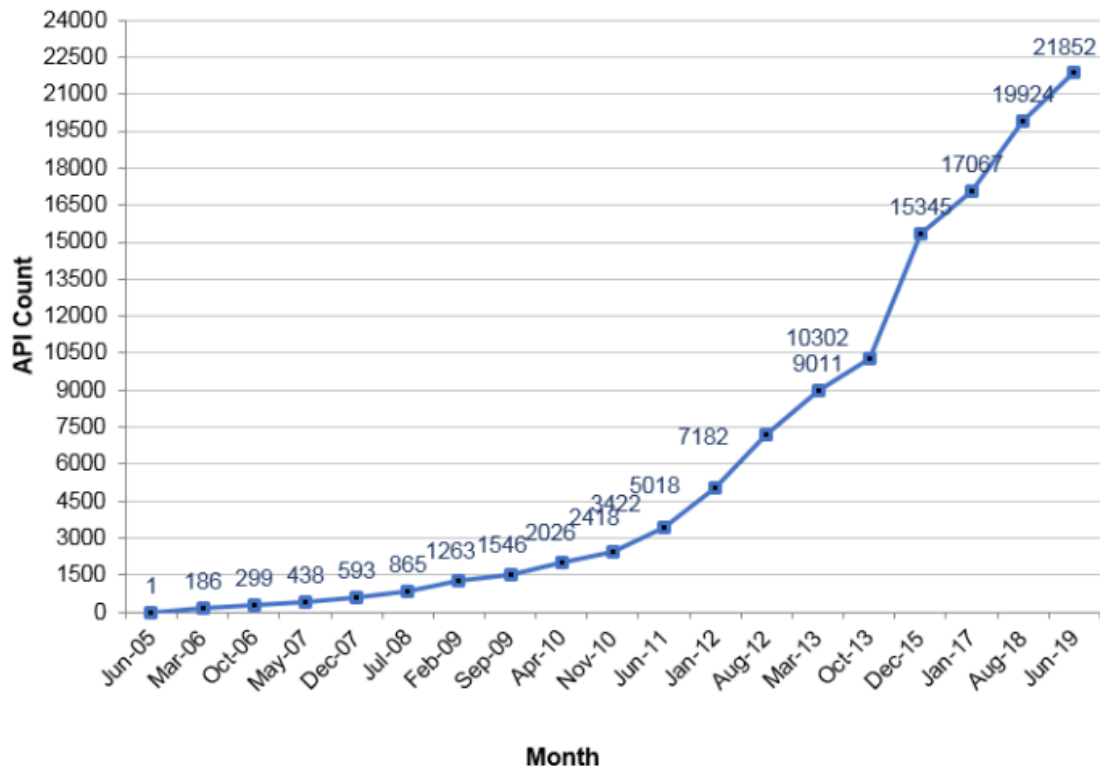
### Self-service

Waarom is open source geslaagd? Hoewel de beschikbaarheid van broncode vaak het onderwerp is van discussies over het succes van open source, is het idee van self-service veel belangrijker. Slechts een klein percentage van de ontwikkelaars wilde de broncode lezen of aanpassen. In plaats daarvan verving open-sourcesoftware commerciële software omdat ontwikkelaars niemand om toestemming hoefden te vragen om de software te nemen en ermee te draaien. Aanbieders van API's hebben geleerd van open source. Een succesvolle API moet beschikbaar zijn op basis van self-service en gemakkelijk te gebruiken zijn. Net als bij open-sourceprojecten hebben de beste API's een bloeiende online community, zowel binnen het bedrijf als in de grotere publieke ontwikkelaarsgemeenschap (of beide). In de meest succesvolle ontwikkelaarsgemeenschappen werken de meest actieve leden niet voor het bedrijf dat de API levert, maar ze helpen omdat de API van cruciaal belang is voor wat ze doen en ze houden ervan om anderen te helpen de waarde ervan te laten zien.

### Volwassen technologie en gemeengoed

Hoewel technologen al tientallen jaren gebruik maken van API's, beseffen maar weinig mensen dat de explosieve groei van de activiteit die zich voltrekt met Twitter, Netflix, en anderen online gebaseerd is op API's. Het eindresultaat dat mensen zien is veel verkeer, maar in feite is het geen webverkeer. Het is API-verkeer. Bedrijven zoals Google, Amazon, Twitter en duizenden anderen gebruiken API's om hun bedrijf te veranderen.

## Growth In [Public] Web APIs Since 2005



Figuur 1: Public APIs zoals vermeld op [programmableweb.com](http://programmableweb.com) in Augustus 2018

ProgrammableWeb ([programmableweb.com](http://programmableweb.com)), een register van web-API's, houdt sinds 2005 API's bij. In 2005 heeft ProgrammableWeb 105 API's gevolgd, waaronder Google, Salesforce, eBay en Amazon. Het aantal is in 2008 verzesvoudigd tot 601 API's, waarbij de belangstelling van sociale en traditionele mediabedrijven om gegevens aan externe partijen aan te bieden toeneemt. Eind 2010 waren er 2.500 API's. De online kleding- en schoenenwinkel Zappos publiceerde een REST API, en veel overheidsinstellingen en traditionele retailers sloten zich aan. De Britse multinational Tesco, een kruidenier en handelaar, stond toe om via API's te bestellen. De fotodelings-app Instagram werd de Twitter voor foto's. The Face introduceerde gezichtsherkenning als service. Met Twilio kon iedereen in een mum van tijd telefonietoepassingen maken. Het aantal openbare API's was in 2011 gestegen naar 5.000. Vanaf dit moment zijn er meer dan 21.000 API's geregistreerd bij ProgrammableWeb. Tegelijkertijd is de trend richting SOAP ten einde.

Dit is de reden en het moment voor de energiesector om een API-first strategie te hanteren en waar het kan SOAP gebaseerde services uit te faseren.

Tech blogger Robert Scoble vatte goed samen waar we nu zijn door drie tijdperken te definiëren :

- Web 1994 was het "get me a domain and a page" tijdperk;
- Web 2000 was het "make my pages interactive and put people on it" tijdperk;
- Web 2010 is het "get rid of pages and glue APIs and people together" tijdperk.

Op dit moment zitten we volop in het tijdperk dat het merendeel van de toepassingen (zoals apps, webapps en IOT programmatuur) ontsloten en gekoppeld worden via API's.

## 2.7 GDPR

In aanvulling op de EU-gegevensbeschermingsrichtlijn heeft het GDPR tot doel individuen (EU-burgers) meer controle, bescherming en privacy te geven over de manier waarop hun persoonlijke informatie wordt gebruikt en door wie.

De regelgeving is vrij uitgebreid en voor veel organisaties zal het bereiken van de GDPR-naleving een duur en lang proces zijn (of zijn geweest). De volledige GDPR-verordening is beschikbaar op : <https://www.itgovernance.eu/en-ie/eu-general-data-protection-regulation-gdpr-ie>

Vanwege de grote hoeveelheid privacy gevoelige informatie in de data die gebruikt wordt binnen de energiesector is deze richtlijn van groot belang. De ACM controleert of er conform deze richtlijn gehandeld wordt.

Hoe kunnen de API's helpen bij de naleving van de GDPR-verordening, nu persoonsgegevens een centrale plaats innemen in de GDPR-verordening ? Hoewel API's misschien niet het enige antwoord zijn, zorgt een goede API management oplossing voor een sterke toegangscontrole over wie toegang heeft tot welke informatie via API's, waardoor wordt gegarandeerd dat persoonlijke gegevens niet worden misbruikt of zonder voorafgaande toestemming worden benaderd. Naast deze controles moet de oplossing ook zorgen voor volledige zichtbaarheid en controleerbaarheid van de toegang tot de gegevens, wat betekent dat elke inbreuk op de gegevens zo snel mogelijk, of binnen de periode van 72 uur, zoals aangegeven in de verordening, aan klanten en autoriteiten kan worden gemeld.

## 3 API strategie

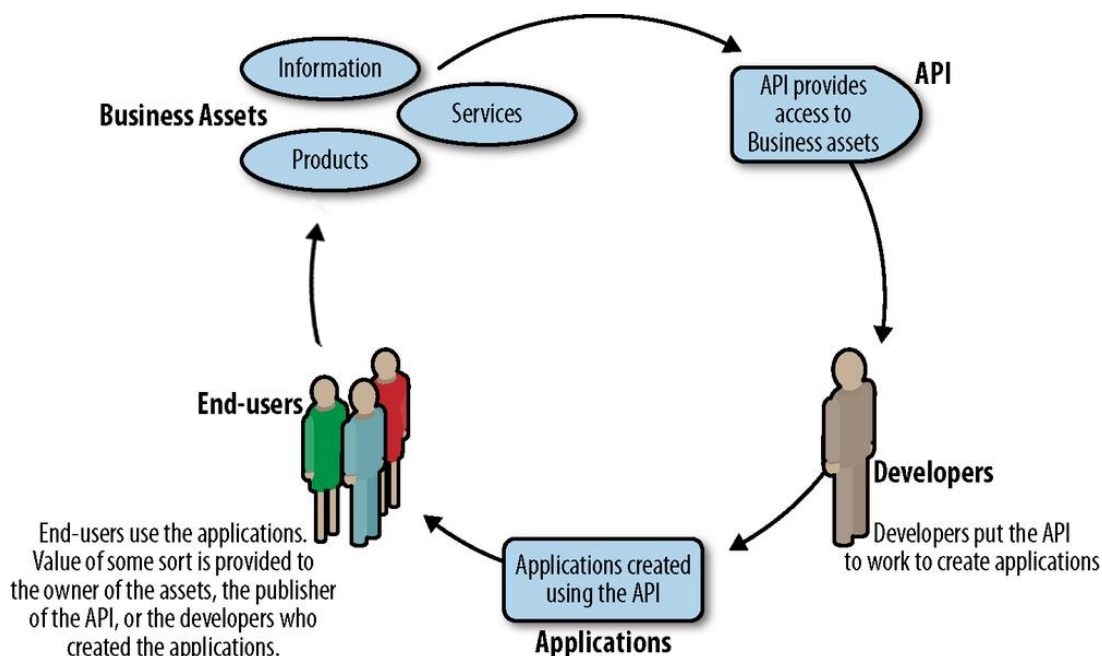
### 3.1 API economie

In een API economie verandert de manier waarop we data delen met de markt. Nu bouwen we alle kanalen zelf: web portalen, mobile apps, social media kanalen, etc. Dit is duur omdat deze vaak veranderen, afhankelijk van het publiek en van proces specifieke zaken. We kunnen onze aandacht beter besteden om onze data, en daarmee de energiesector, te presenteren via API's. Ontwikkelaars in de markt kunnen dan de specifieke kanalen bouwen, gebruik makend van onze APIs.

Startups, commerciële bedrijven en app ontwikkelaars kunnen bijvoorbeeld vanuit een financieel en duurzaam perspectief deel gaan nemen aan de API economie. Zij kunnen feedback leveren op basis van wie, wanneer en hoeveel er gebruik wordt gemaakt van de verschillende API's.

Met de API value chain (Figuur 2) kan het API landschap zich blijven ontwikkelen op voorwaarde dat partijen binnen de energie sector gezamenlijk (via de NEDU) de juiste data beschikbaar stellen via de verschillende API's. Daarmee helpen we het nieuwe systeem van marktprocessen incrementeel op te bouwen.

### 3.2 API value chain



Figuur 2: API Value chain (overgenomen uit [2])

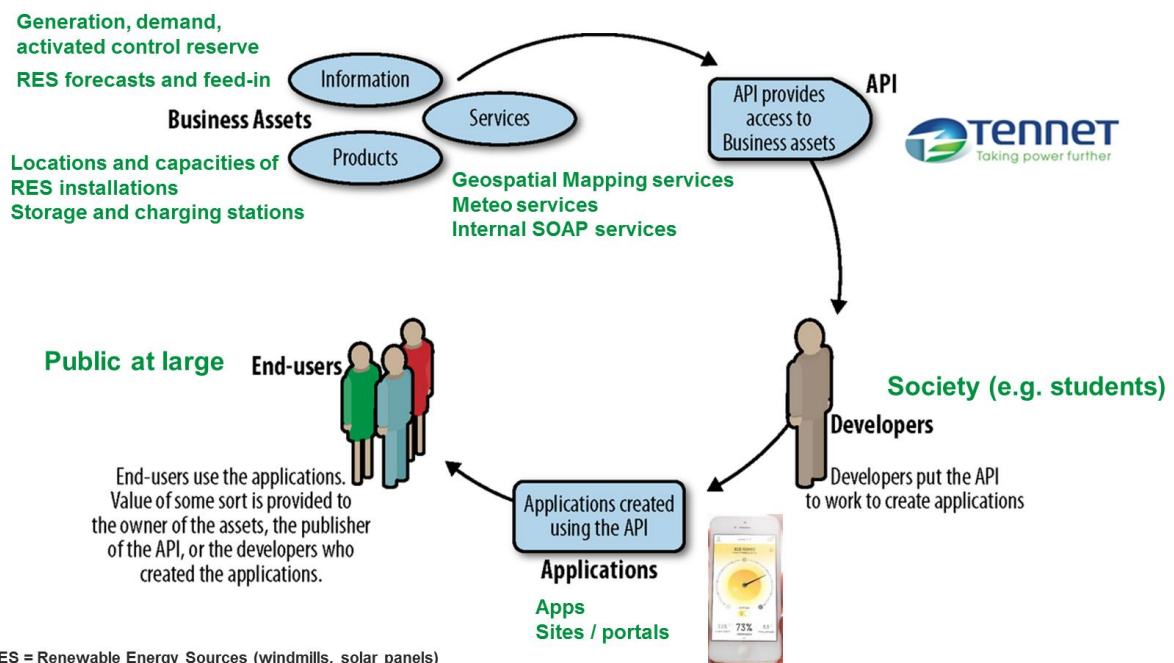
Zoals al eerder aangegeven is het aanbieden van een API niet alleen een technologisch probleem; het is ook een mensen- en procesprobleem. De hierboven afgebeelde API Value chain beschrijft de anatomie van dit kanaal met alle spelers en verschillende schakels.

Samenvattend gaat het om de volgende onderdelen uit De API Value chain:

1. Relevante business assets kiezen (data en diensten);
2. API's ontwerpen en laagdrempelig toegankelijk maken. Hiervoor zorgt de API provider;
3. API's gebruiken in markt toepassingen (marketing via overheid of commerciële kanalen);
4. Markt toepassingen gebruiken (marketing via overheid of commerciële kanalen);
5. Feedback verzamelen, verwerken en erop reageren als input voor stap 1.

Hieronder de invulling van een API value chain. De API provider is in dit geval TenneT. Er worden verschillende business assets gecombineerd ingezet om te ontsluiten via de API. Hierbij kan ook gebruik gemaakt worden van een third-party API provider, zoals een Geospatial mapping service. In dit voorbeeld worden ook interne services uit het achterlandschap gebruikt.

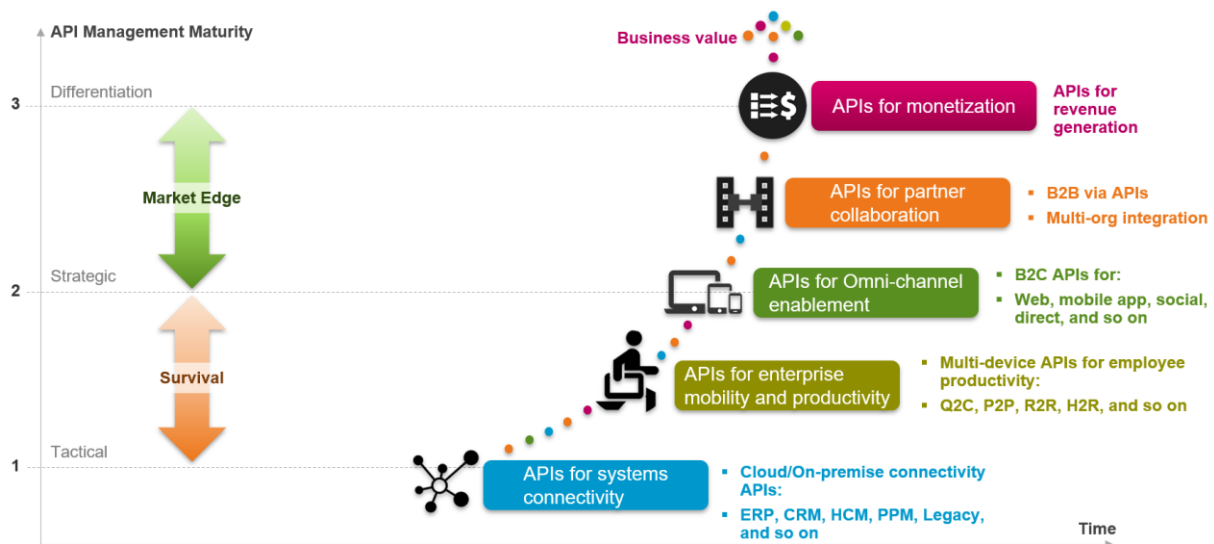
De API's uit onderstaand voorbeeld worden aangeboden aan de maatschappij zodat er toepassingen ontwikkeld kunnen worden door bijvoorbeeld studenten. Deze toepassingen worden vervolgens gebruikt door eindgebruikers, in dit geval het publiek in het algemeen.



RES = Renewable Energy Sources (windmills, solar panels)

Figuur 3: voorbeeld ingevulde API Value chain

### 3.3 API Classificatie NEDU



Figuur 4: API Value classification (overgenomen uit [5])

De hierboven afgebeelde API value classification classificeert API's in vijf hoofdgroepen. Elke groep wordt bepaald op basis van de bedrijfswaarde die het toevoegt, wat op zijn beurt volgens dit door business gedreven model ook de volwassenheid bepaalt.

Voor de NEDU hebben we in onderstaande tabel de hoofdgroepen opgenomen waar de NEDU zich op zou moeten focussen. Issues / business cases kunnen passen in één of meer hoofdgroepen.

Hoofdgroep	Rationale business value NEDU	Volwassenheid
APIs for system connectivity	Een "API led" strategie hanteren voor het ontsluiten v.d. bestaande systemen (APIs for systems connectivity) in NEDU verband, waardoor bestaande systemen binnen NEDU (bijv. Aansluitinformatie, CALGOS IC243) toegankelijk worden gemaakt via API's. Dit geeft een basis waarop andere API's kunnen voortbouwen.	tactisch
APIs for Omni-channel enablement	B2C (business to consumer) API's aanbieden voor de maatschappij, zodat op basis daarvan web apps, mobile apps etc. kunnen worden ontwikkeld voor de maatschappij.	strategisch
APIs for partner collaboration	Het gebruik maken en integreren van third party API's (geospatial API's of API's van partijen uit de energiesector). In specifieke gevallen kunnen API's ook ingezet worden voor B2B doeleinden (B2B berichtenverkeer).	differentiatie



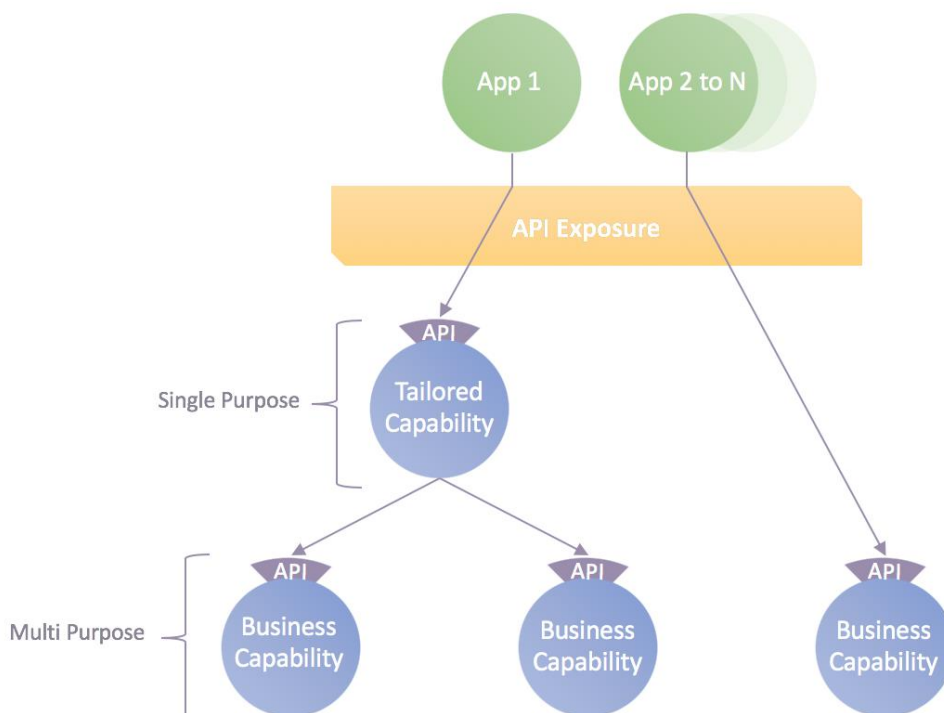
### 3.4 API categorieën

API's kunnen gecategoriseerd worden in verschillende type API's.

Sommige API's kunnen bijvoorbeeld worden gebouwd met een specifieke use case in gedachten en ter ondersteuning van een specifieke toepassing. Hierdoor kunnen dergelijke API's heel gespecialiseerd zijn en op maat gemaakt worden voor het doel waarvoor ze gebouwd zijn. Met andere woorden, ze hebben slechts één doel en zijn niet geschikt voor hergebruik buiten de context waarvoor ze zijn gebouwd.

Een veelgebruikte term voor dit soort (single-purpose) API's is Experience API's (zie par 3.4.2. v.d. API strategie voor de Nederlandse overheid), voornamelijk vanwege hun rol in het mogelijk maken van toepassingen waarmee mensen direct interacteren (bijvoorbeeld mobiele apps, webapps, etc.). Niet alle toepassingen die gespecialiseerde API's vereisen, hoeven echter met mensen te interacteren. In Industry 2.0 bijvoorbeeld, kunnen API's worden gebouwd ter ondersteuning van moderne industriële lijnen, of in de agrarische sector worden drones gebruikt om de bodemgesteldheid in grote gebieden te scannen, en worden API's gebruikt om gegevens in real time te verkrijgen en te versturen. Daarom geven wij de voorkeur aan een meer algemene term: single-purpose API's.

Andere API's zouden echter specifiek gebouwd kunnen worden met het oog op hergebruik. Zulke API's zullen meer generiek van aard zijn en zullen niet gebonden zijn aan een bepaalde use case. Daarom zijn deze API's multi-purpose, wat betekent dat ze kunnen worden gebruikt in een verscheidenheid van scenario's en dus in staat moeten zijn om vele toepassingen te dienen. Het volgende diagram illustreert dat API's die geen toegang bieden tot op maat gemaakte bedrijfscapaciteiten door veel toepassingen kunnen worden gebruikt om verschillende use-cases aan te pakken.



Figuur 5: API gestuurde communicatie (overgenomen uit [5])

### 3.5 API First

Het hanteren van een API First aanpak betekent dat er eerst een API ontwikkeld wordt voordat er een website, web, mobiele of single page applicatie wordt ontwikkeld.

Daarnaast betekent dit dat voor elk ontwikkelingsproject de API's behandeld worden als "first-class citizens". Dat alles in een project draait om het idee dat het eindproduct kan worden geconsumeerd door allerlei toepassingen, zoals mobiele apparaten en webapplicaties. Een API First benadering houdt in dat er API's worden ontwikkeld die consistent en herbruikbaar zijn, wat kan worden bereikt door een API-beschrijvingstaal te gebruiken om een contract op te stellen voor de manier waarop de API zich moet gedragen.

Een API-First Design bestaat uit 3 principes:

- De API is de eerste gebruikersinterface en interactie voor uw toepassing;
- De API komt eerst, dan de implementatie (de implementatie kan veranderen, maar uw API niet);
- Een API kan worden beschreven (en misschien zelfs zelfbeschrijvend).

Het advies is om een API First aanpak te hanteren in projecten waarin de ontwikkeling van API's één van de activiteiten is.

### 3.6 NEDU en API's

API's zijn een technologie waarmee de energiesector digitale diensten kunnen realiseren om de energietransitie te ondersteunen. De NEDU issue commissies kunnen (en moeten) door deze technologische ontwikkeling vrijelijk nadenken over deze digitale dienstverlening.

Daarnaast helpen API's om een wendbaardere IT omgeving te realiseren waarmee ontwikkelingen als een virtuele netbeheerder eenvoudiger kunnen worden ondersteund. Daarom is API management een relevante technologie voor architecten om te worden toegevoegd aan het palet aan IT integratie hulpmiddelen en zou kunnen uitgroeien tot het key integratiemiddel om een wendbare sector te realiseren (API tenzij).

API's zijn een belangrijke bouwsteen voor innovatie. Daarnaast is het bouwen van een API laagdrempelig en gemeengoed bij ontwikkelaars. Niet voor niets is het merendeel van grote organisaties al over op API's. Al met al komen we gezamenlijk als werkgroep tot de conclusie dat het advies moet zijn : **API's tenzij.**

Uitzonderingen zijn dan specifieke non-functional requirements waar nog geen bewezen , volwassen en algemeen geaccepteerde standaarden voor zijn, zoals bijvoorbeeld formele B2B informatie-uitwisseling waarin bijvoorbeeld onweerlegbaarheid op berichtniveau van belang is (de NEDU TC architectuur zal voor dit soort keuzes beleid opstellen).