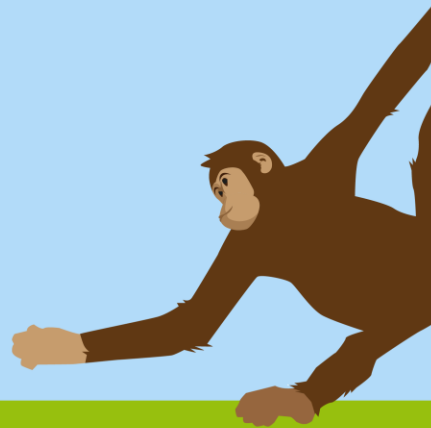


Werkgroep notificaties update

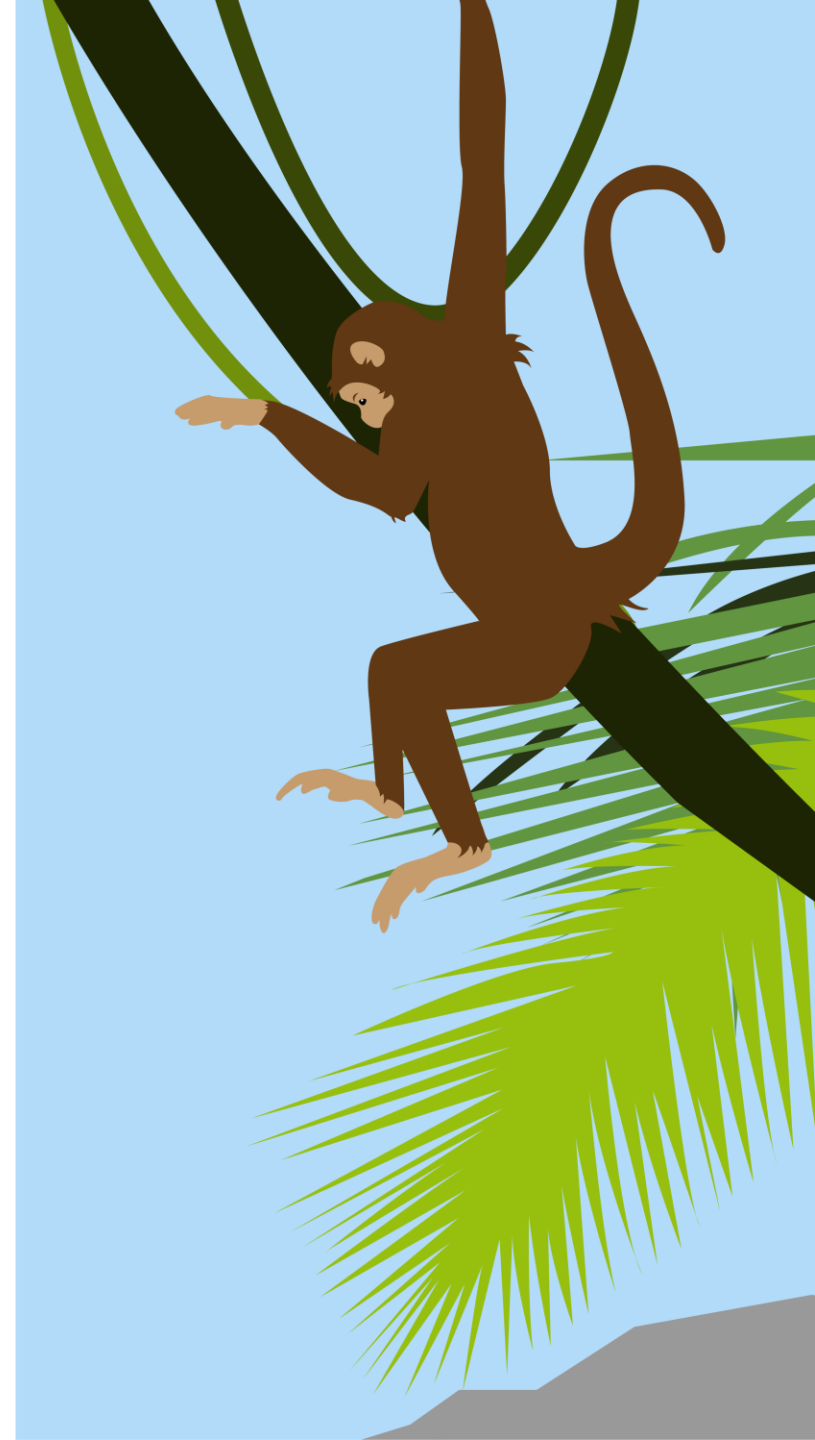
2025-06-05

Frank Terpstra & Jeffrey Gortmaker

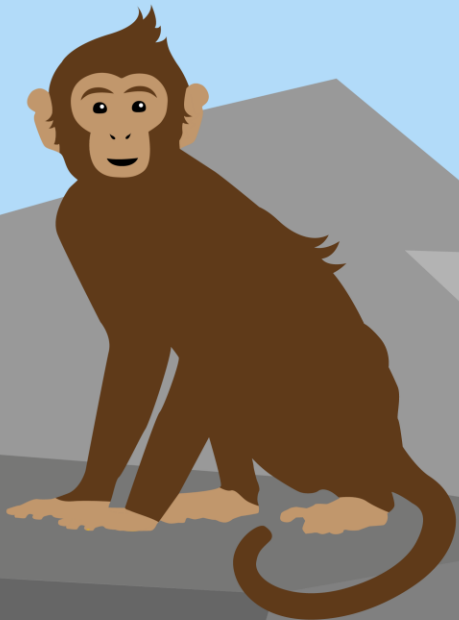


Agenda

- Historie
- Dimensies
- Categorieën
- Werkgroep resultaten
- AsyncAPI
- Architectuur
- Vervolg



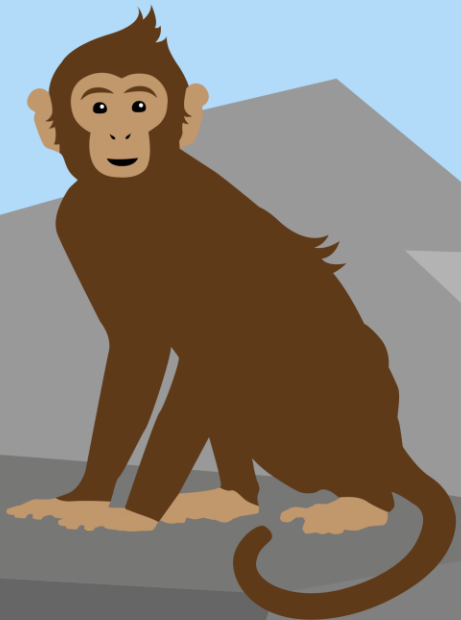
Historie



Historie

- Waarom notificaties?
- Historie binnen Kennisplatform
- Wat hebben we al?
 - Inhoud
 - CloudEvents
 - Metadata
 - Async API
 - Transport
 - AMQP
 - Webhooks
 -

Dimensions



Dimensies Metadata

- Metadata: waar en hoe beschrijf je notificaties
 - Machine leesbaar – Menselijk leesbaar
 - Bijvoorbeeld AsyncAPI – Developer portal

Dimensies berichtinhoud

- Berichtinhoud
 - Semantiek – Ontwerp - Structuur
 - Bijvoorbeeld MIM - Dataminimalisatie – Cloudevents

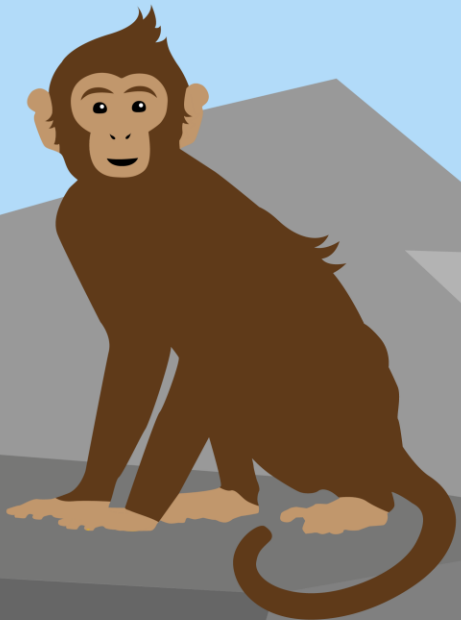
Dimensies functionaliteit

- Functionaliteit
 - Processlaag - Applicatielaag – Transportlaag – Netwerklaag
 - Synchron – A-synchron – Out of band
 - In welke laag regel je:
 - Onweerlegbaarheid (non-repudiation)
 - Betrouwbaar transport (reliability)
 - Berichtvolgorde
 - Beveiliging (encryptie)

Dimensies tijd

- Tijdigheid aflevering
 - Real-time – binnen een redelijke(wettelijke termijn)
- Mogelijkheid tijdreizen (formeel/materiële historie)
 - Is er een afhankelijkheid tussen verzonden notificatie en bijhouden historie aan de bron
 - Bij informatie arm notificeren, met tijdstempel informatie ophalen

Categorien



Categoriën

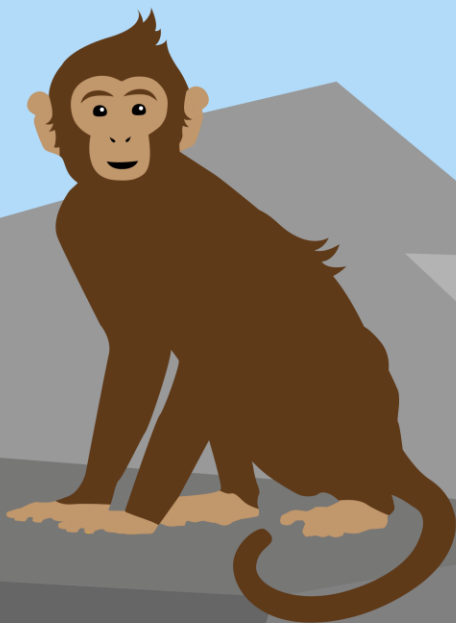
- Overzicht gebaseerd op VNG-model
- Huidig gebruik en/of wenselijk
- Gevolgen voor afnemers

Vorm van notificeren	Toelichting	Gevolgen afnemer
Gebeurtenis + identificatie (informatiearm)	Gegevens later/apart ophalen	
Gebeurtenis + gegevens (informatierijk)	Gebeurtenis is het uitgangspunt. Gegevens horen bij gebeurtenis.	

Vorm van notificeren	Toelichting	Gevolgen afnemer
Gebeurtenis + identificatie (informatiearm)	Gegevens later/apart ophalen	
Gebeurtenis + gegevens (informatierijk)	Gebeurtenis is het uitgangspunt. Gegevens horen bij gebeurtenis.	
Gegevensmutatie + Aanleiding	Oude & Nieuwe gegevens + Aanleiding (soort gebeurtenis)	
Gegevensmutatie + Afgeleide aanleiding		Aanleiding afleiden (waar mogelijk)

Vorm van notificeren	Toelichting	Gevolgen afnemer
Gebeurtenis + identificatie (informatiearm)	Gegevens later/apart ophalen	
Gebeurtenis + gegevens (informatierijk)	Gebeurtenis is het uitgangspunt. Gegevens horen bij gebeurtenis.	
Gegevensmutatie + Aanleiding	Oude & Nieuwe gegevens + Aanleiding (soort gebeurtenis)	
Gegevensmutatie + Afgeleide aanleiding		Aanleiding afleiden (waar mogelijk)
Gegevensmutatie	Oude & Nieuwe gegevens	Gegevensmutatie als CloudEvent
"Seintje"	Signaal als er iets gebeurt/verandert	Gegevensmutatie vertalen naar CloudEvent "object x gewijzigd"

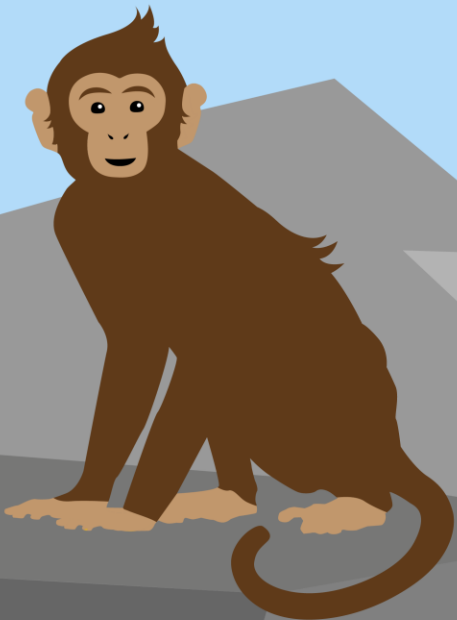
Werkgroep resultaten



Besproken onderwerpen

- Reliable messaging(staat van ebMS /AS4)
- Abboneren (meer nodig dan alleen cloudevents)
- Terugkoppeling VNG resultaten notificeren door Jeanot Bijpost
 - Abonneren is complex
 - Cloud events != event sourcing
 - Behoeftte aan semantische mapping
 - Combinatie Filters/abbonnementen en autorisaties lastig

AsyncAPI



Wat is AsyncAPI

- Het is de Asynchrone tegenhanger van Open API Specification(OAS, voorheen swagger)
- Open standaard, Onderdeel van Linux foundation
- Bestaat sinds 2020
- Heeft actieve community

Verschillen met OAS

- Bedoeld voor Asynchrone uitwisseling, maar architectuur agnostisch
 - Zo opgezet dat alle soorten asynchrone uitwisselingen erin gedefinieerd kunnen worden
- Beschrijft Server, Application, Sender, Receiver, Message, Channel, Protocol en Bindings
 - OAS beschrijft alleen Server
- Kan bestaan uit meerdere (naar elkaar verwijzende) bestanden
 - OAS is één bestand per API

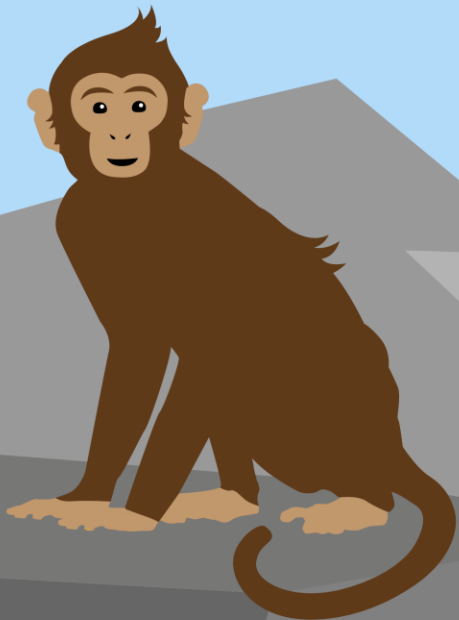
Wat kan je met AsyncAPI?

- Asynchrone interfaces definiëren (protocol agnostisch)
- Documentatie genereren voor zowel Client Als server
 - HTML Template
 - React component
- Code genereren voor zowel client als server
 - O.A. in Node.js, Java, Python, C#, RUST, Go en meer

Volgende stap

- Met werkgroep AsyncAPI bekijken
 - Voegt het waarde toe om deze standaard verplicht te stellen?
 - Heeft iemand hem al geïmplementeerd?
 - Is er een NL-Gov profiel nodig of is hij zoals Oauth direct bruikbaar

Architectuur





Notificeren

Notificeren is het door een aanbieder aan afnemers verstrekken van informatie over binnen zijn domein plaatsgevonden gebeurtenissen.

Een gebeurtenis is iets dat plaatsvindt op een bepaald moment in de tijd waardoor de toestand verandert.

De waarde van notificeren

- Geïnformeerd worden over gebeurtenissen
- Snel kunnen reageren op gebeurtenissen
- Hogere robuustheid, flexibiliteit en schaalbaarheid door ontkoppeling

Bron: Project Notificatie Services, deel 2 (2022): "Waarom notificeren?"



Gewenste capabilities

1. Definiëren en publiceren gebeurtenistypes
2. Publiceren van aanbieders van notificaties voor gebeurtenistypes
3. Abonneren op gebeurtenistypes bij een aanbieder (en kunnen opzeggen abonnement)
4. Versturen van notificaties naar abonnees
5. Autorisatie op niveau van gebeurtenistypes
6. Autorisatie op niveau van individuele gebeurtenissen
7. Beveiligde uitwisseling van notificaties – anonimiseren, pseudonimiseren, versleutelen
8. Filtering van notificaties
9. Precies één keer ontvangen
10. Afdwingen volgorde



Architectuurvraagstukken

1. Wanneer is een "intermediary" centraal of decentraal?
 - Decentraal is meer federatief en heeft daarom in het algemeen de voorkeur
 - Sommige gebeurtenistypes hebben meerdere aanbieders en kunnen om een centrale intermediary vragen
2. Is er een landelijke catalogus voor gebeurtenistypes nodig?
 - Hoe wordt de catalogus gevuld?
3. Is er een landelijke catalogus van aanbieders (per gebeurtenistype)?
4. Is er een generieke en waardevolle classificatie van gebeurtenistypes?

Vervolg

- Opsplitsen in twee werkgroepen
 - Architectuur
 - AsyncAPI
- Bijeenkomst 13-11 van 14:00-16:00