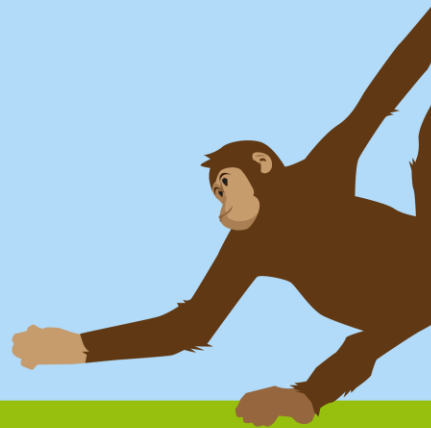


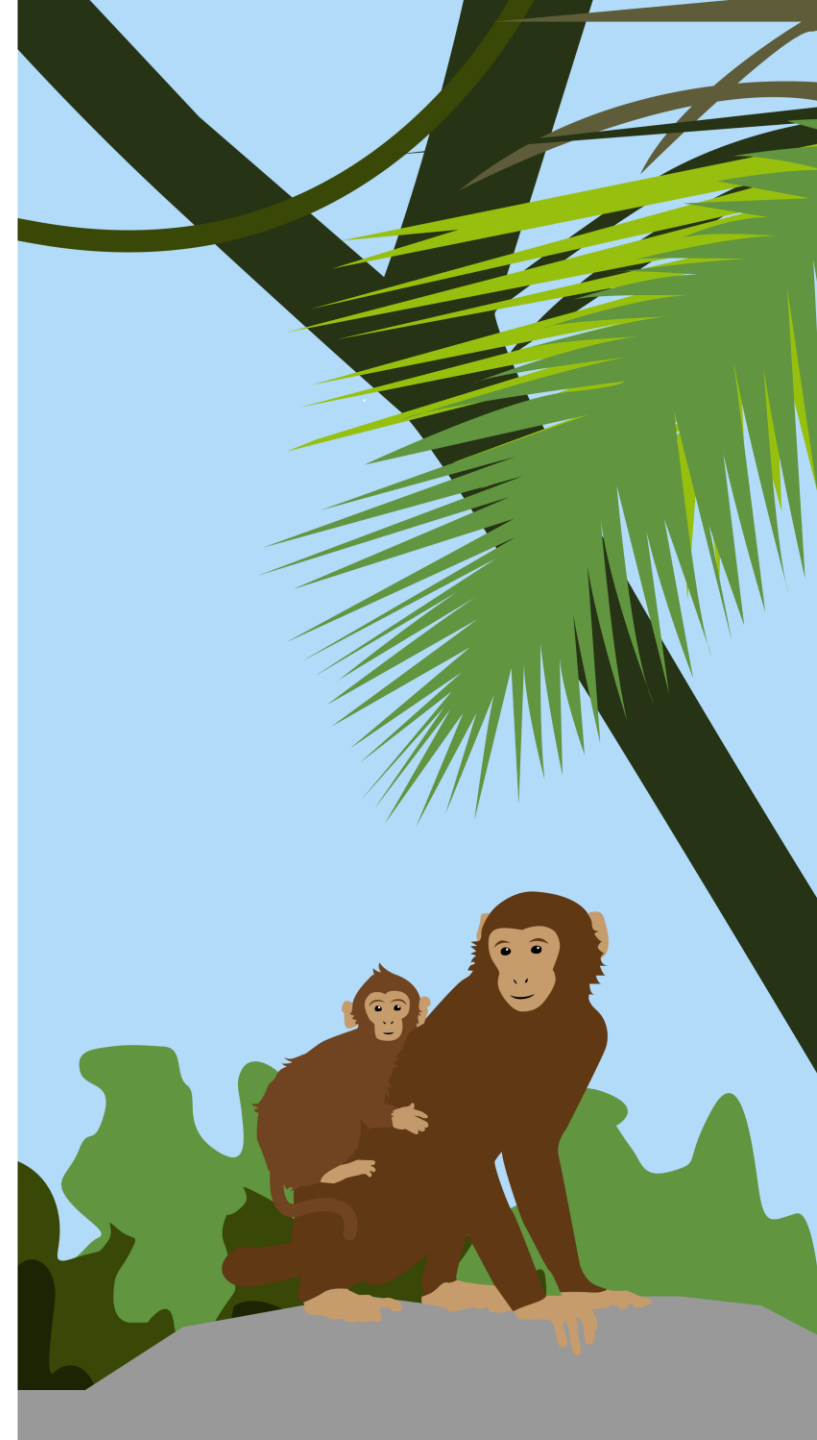
Resultaten Werkgroep Orkestratie

5 november 2025



Agenda

- Architectuur
- Beveiliging / Access control
- Batching module

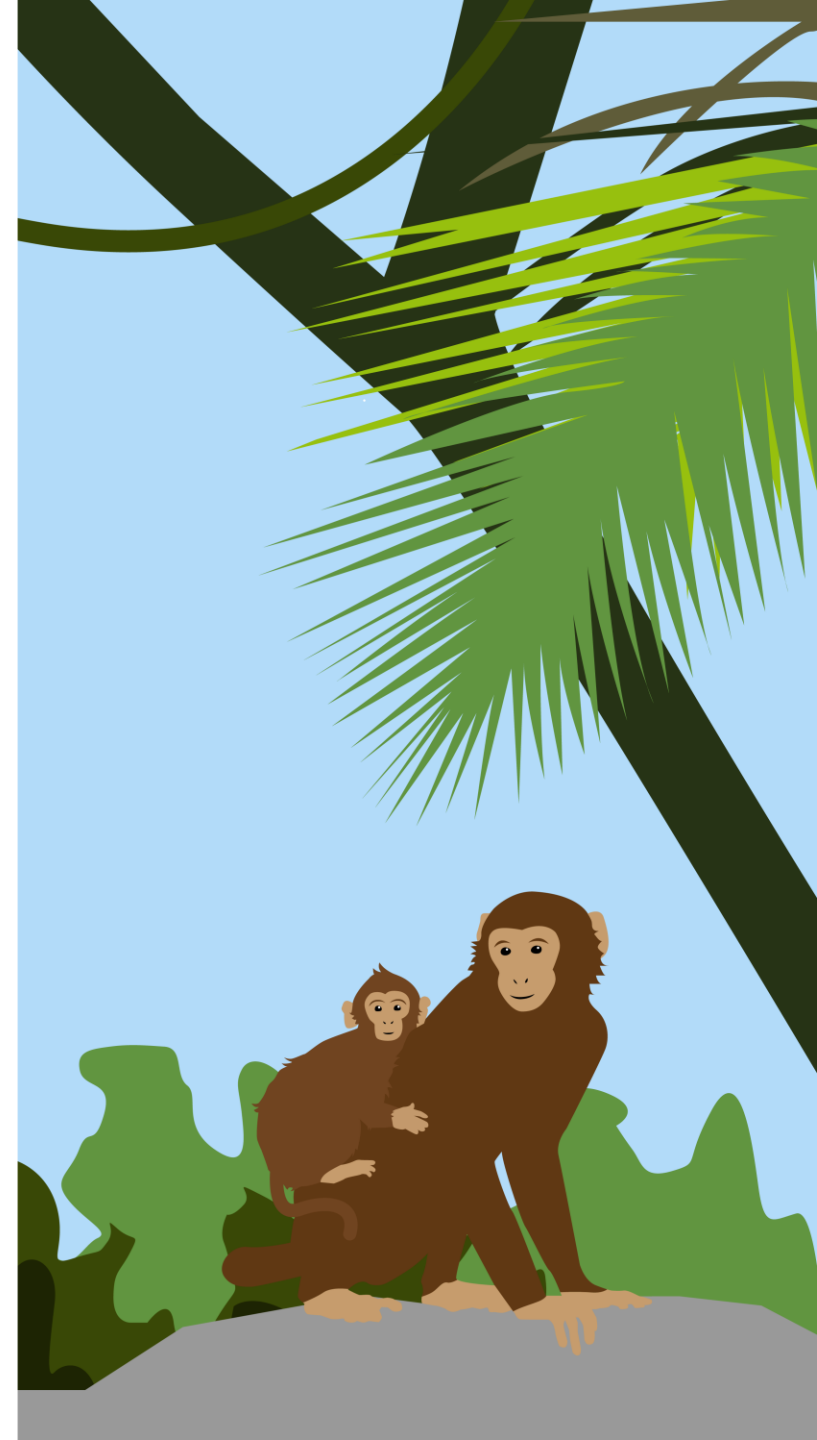


Architectuur

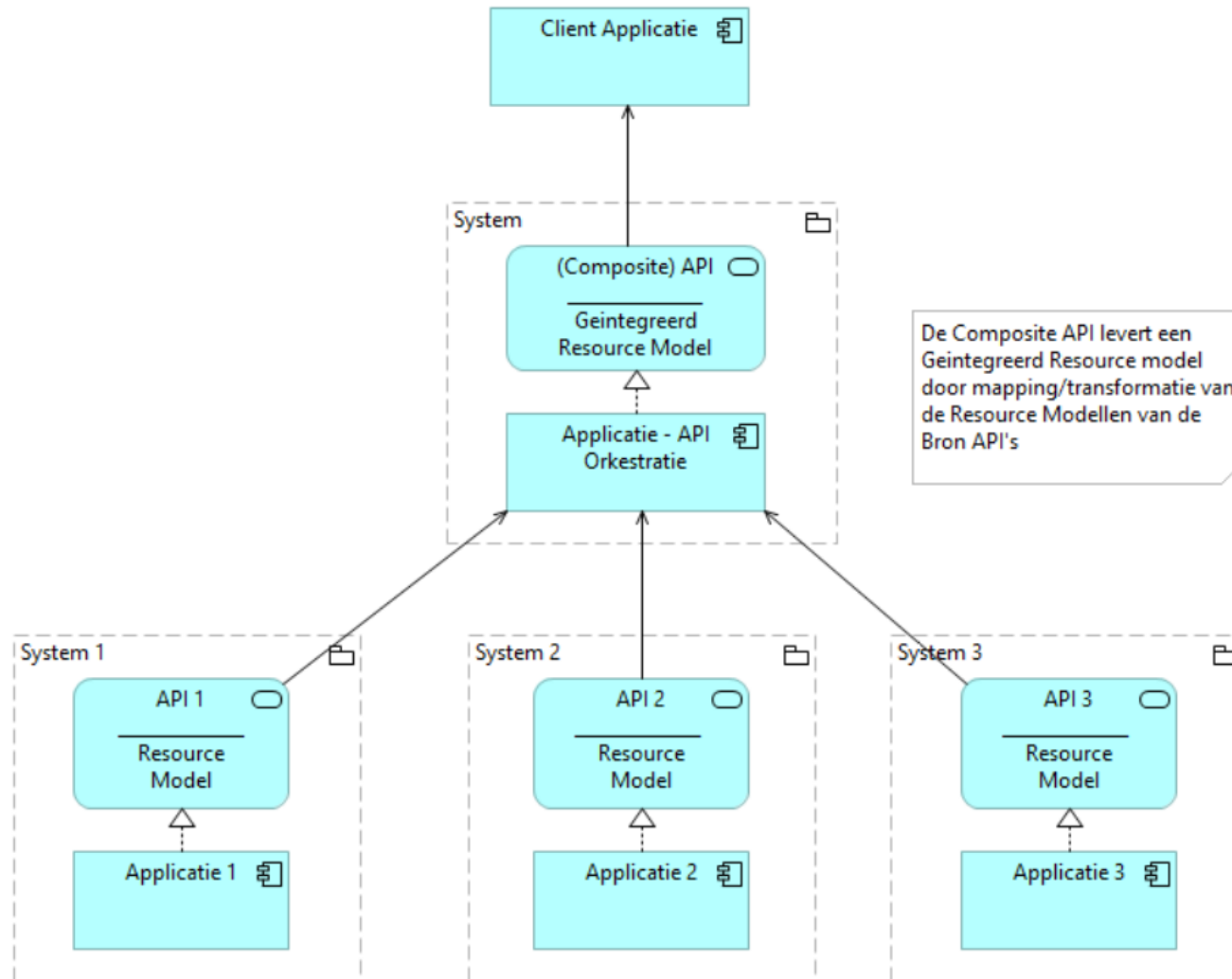
Orkestratie

API orkestratie verwijst naar het proces waarbij meerdere API-aanroepen worden gecoördineerd om een specifieke taak of workflow uit te voeren.

Dit kan bijvoorbeeld inhouden dat gegevens van verschillende bronnen worden samengevoegd of dat meerdere services in een bepaalde volgorde worden aangeroepen om een einddoel te bereiken.



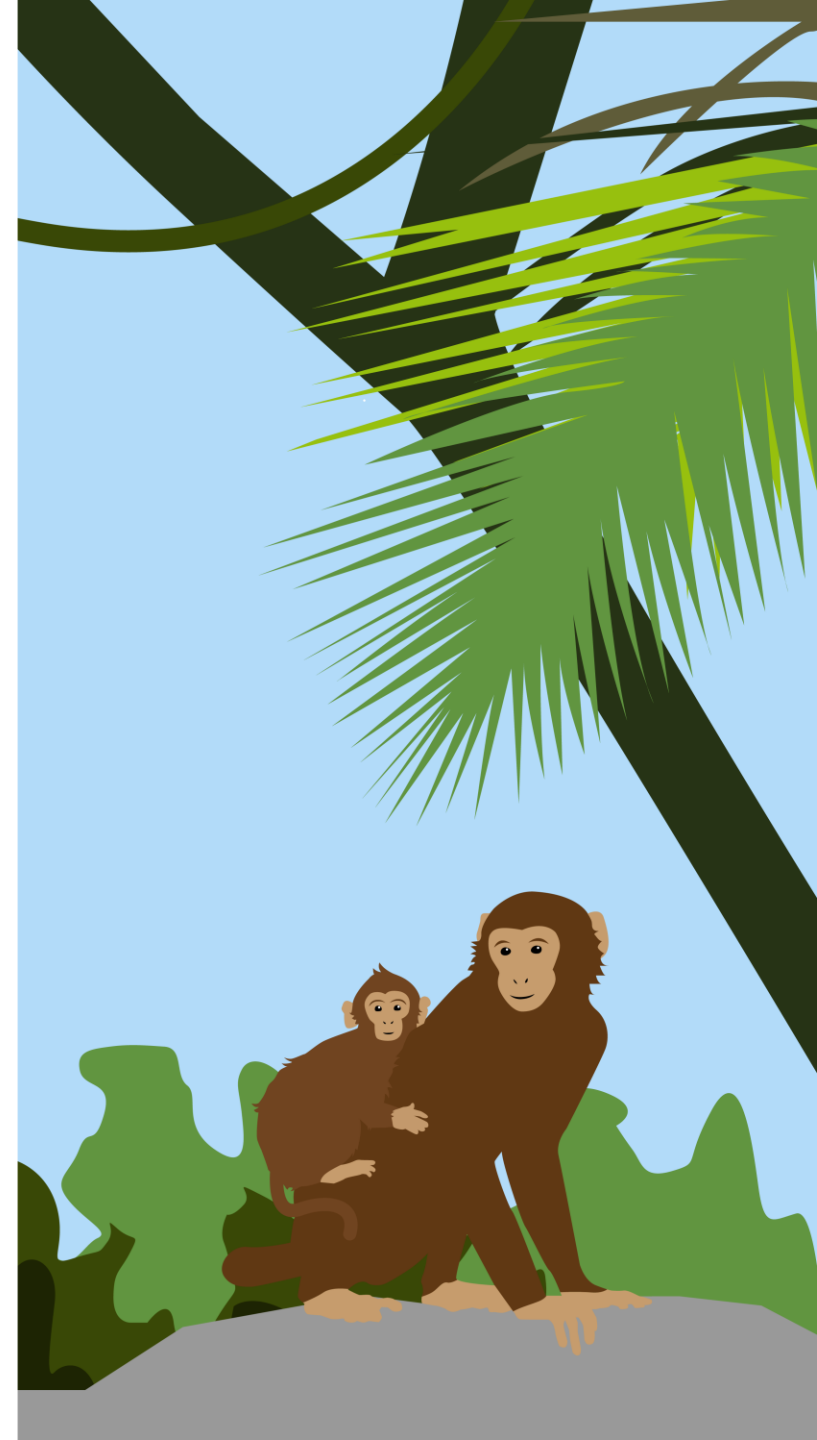
Informatiemodel & API



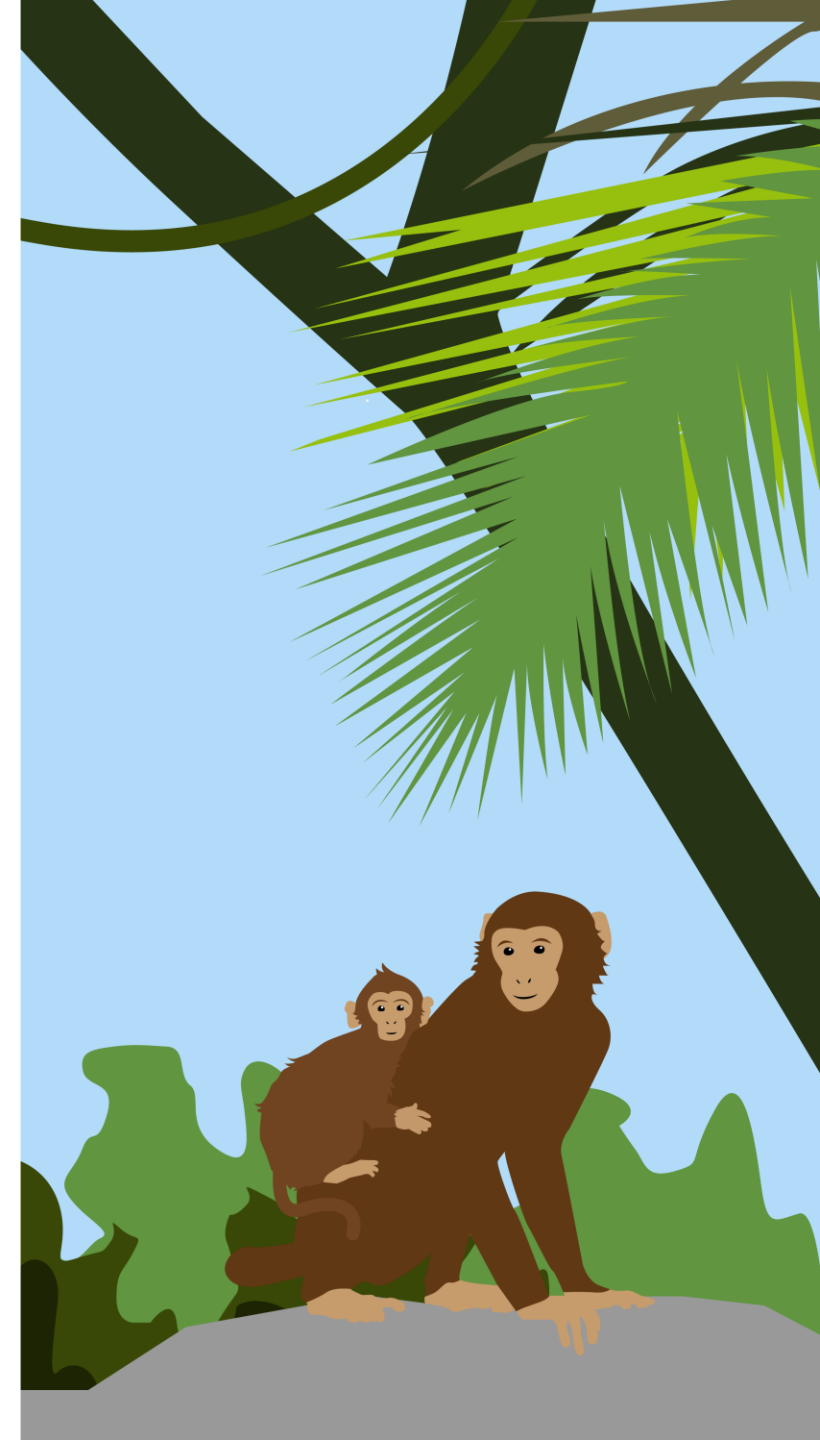
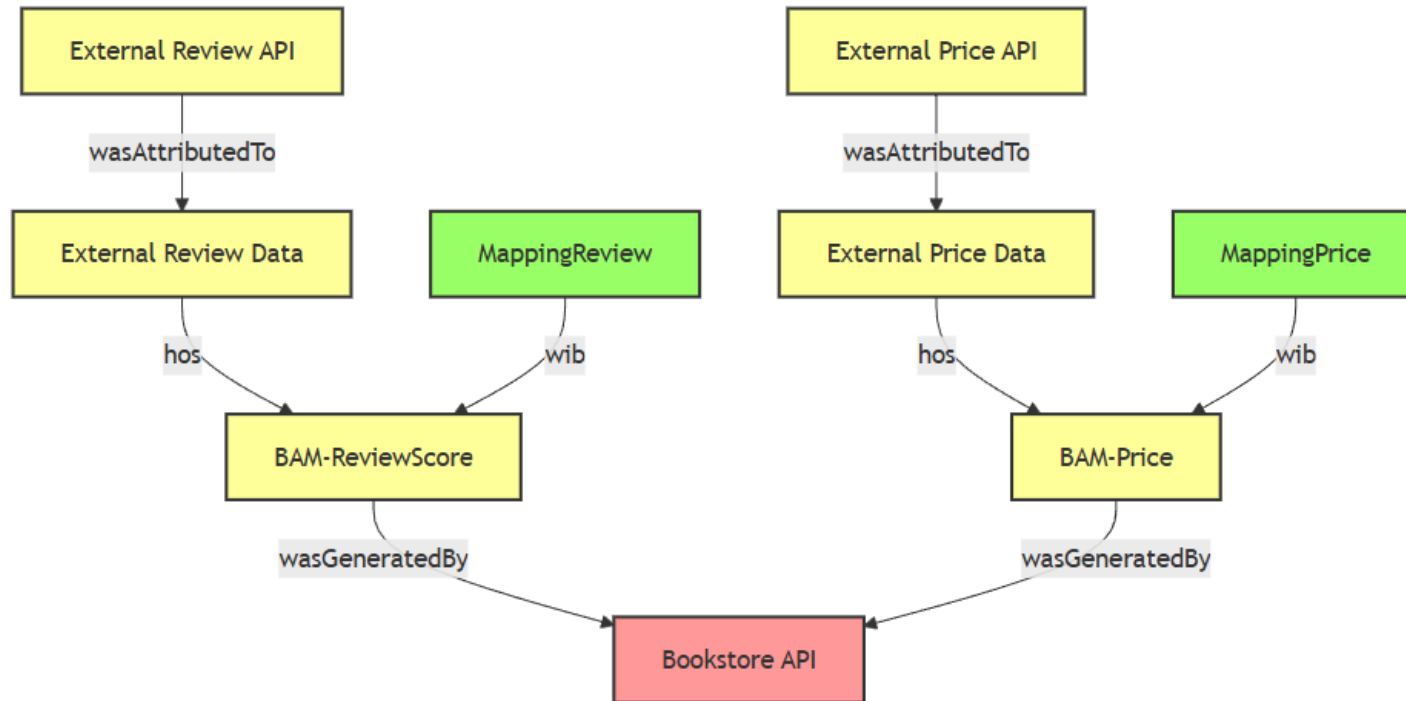
Relaties tussen API's vastleggen mbv de w3c PROV standaard

Kernconcepten van W3C PROV

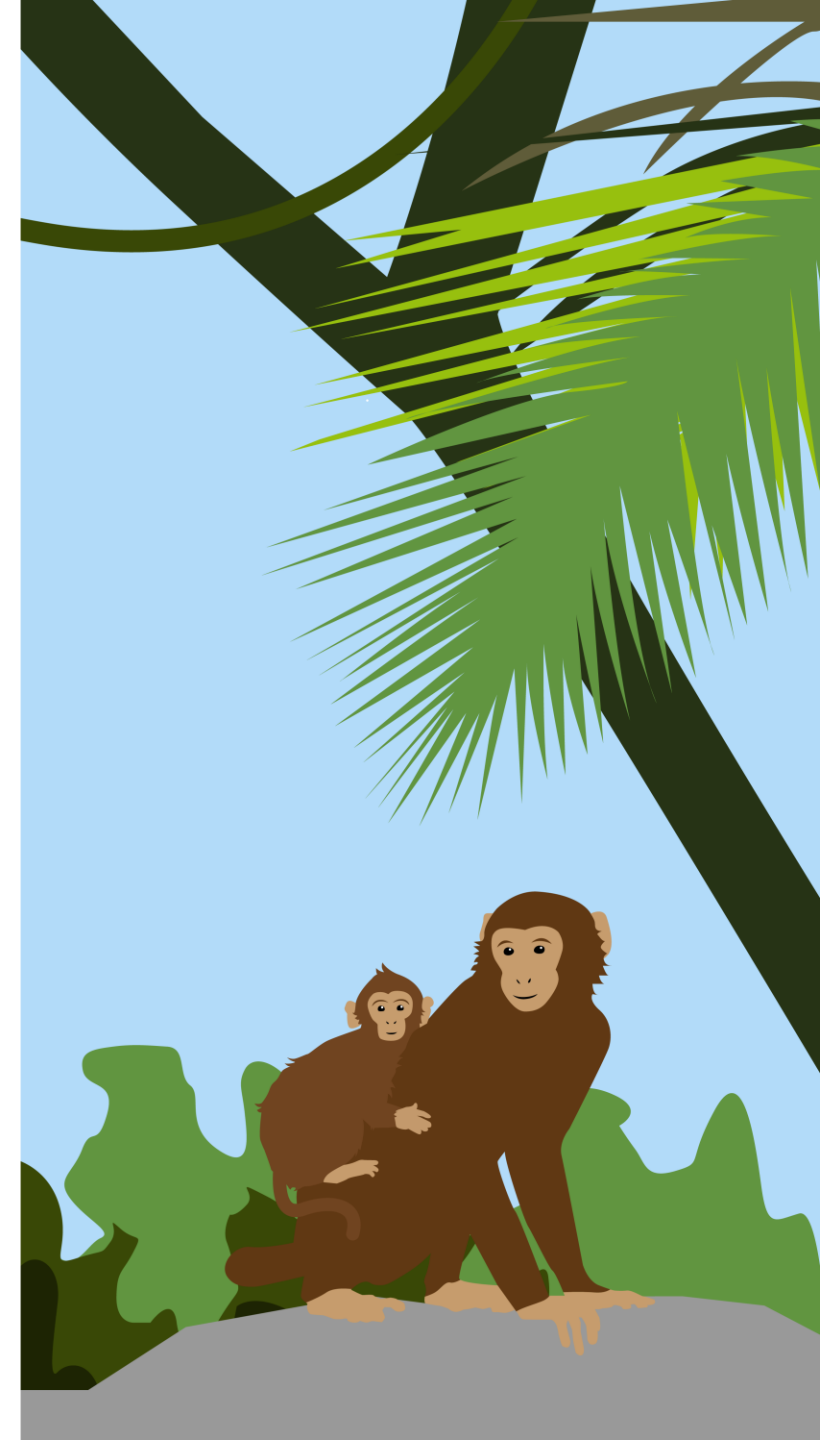
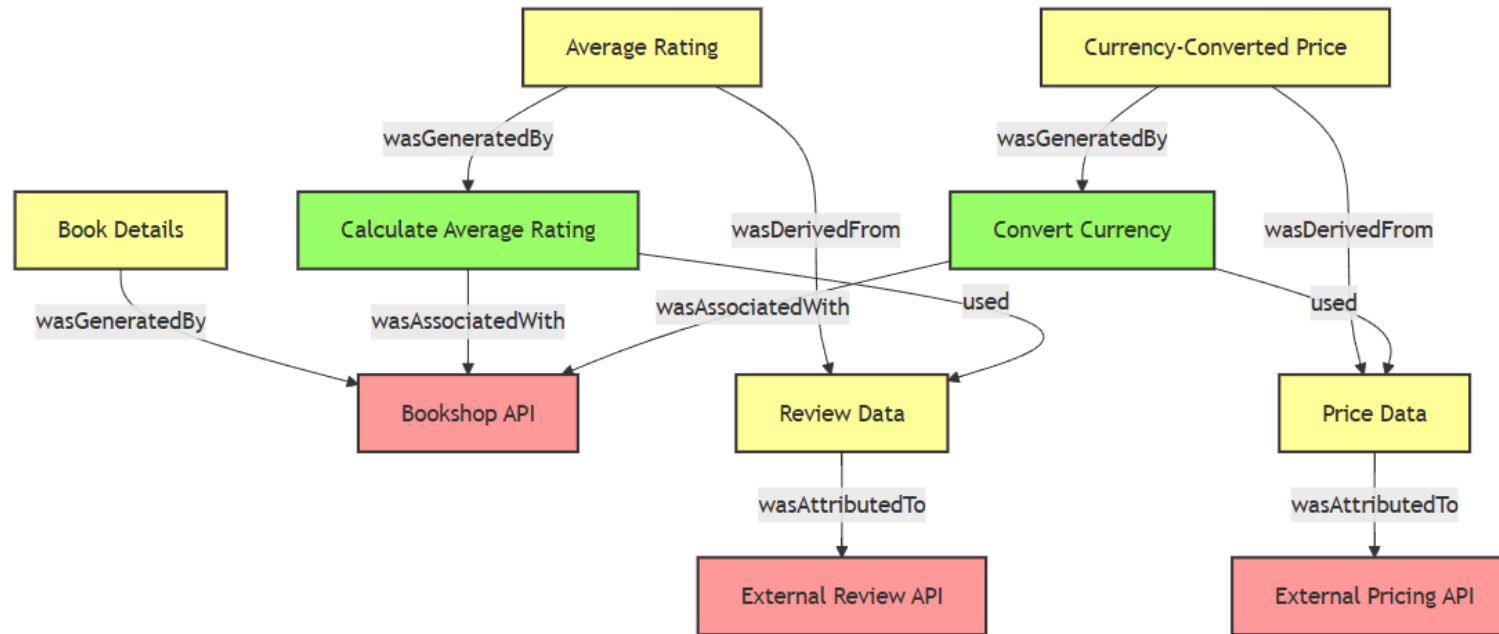
- **Entiteiten:** Alles wat beschreven kan worden, zoals API-resources, databaserecords of bestanden.
- **Activiteiten:** Bewerkingen of processen die van invloed zijn op entiteiten (bijv. datatransformaties, berekeningen).
- **Agenten:** De personen, organisaties of systemen die verantwoordelijk zijn voor activiteiten.
- **Relaties:** De verbindingen tussen entiteiten, activiteiten en agents, zoals wasGeneratedBy, used of wasDerivedFrom.



Boekwinkel voorbeeld

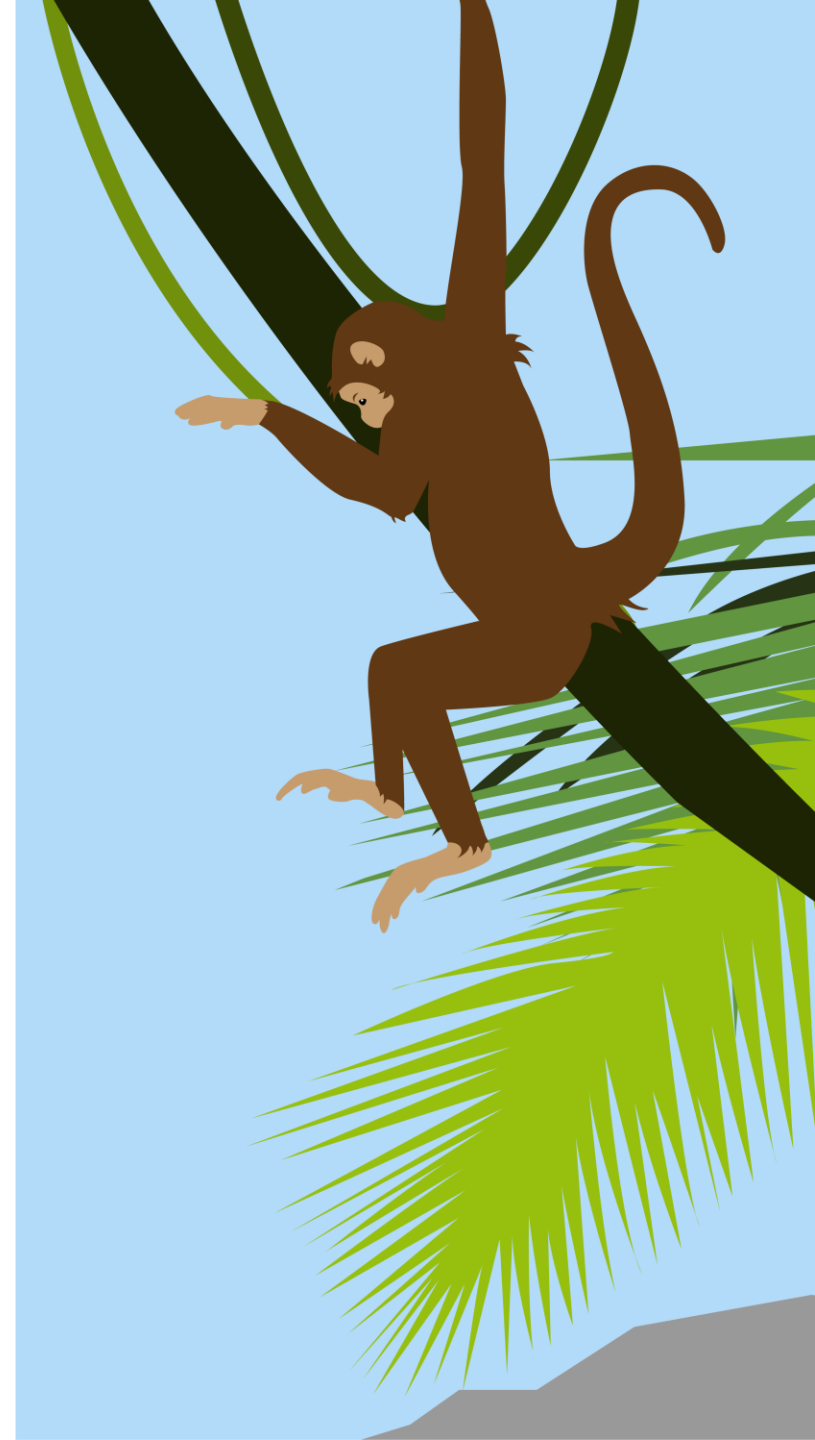


Boekwinkel voorbeeld (berekening)



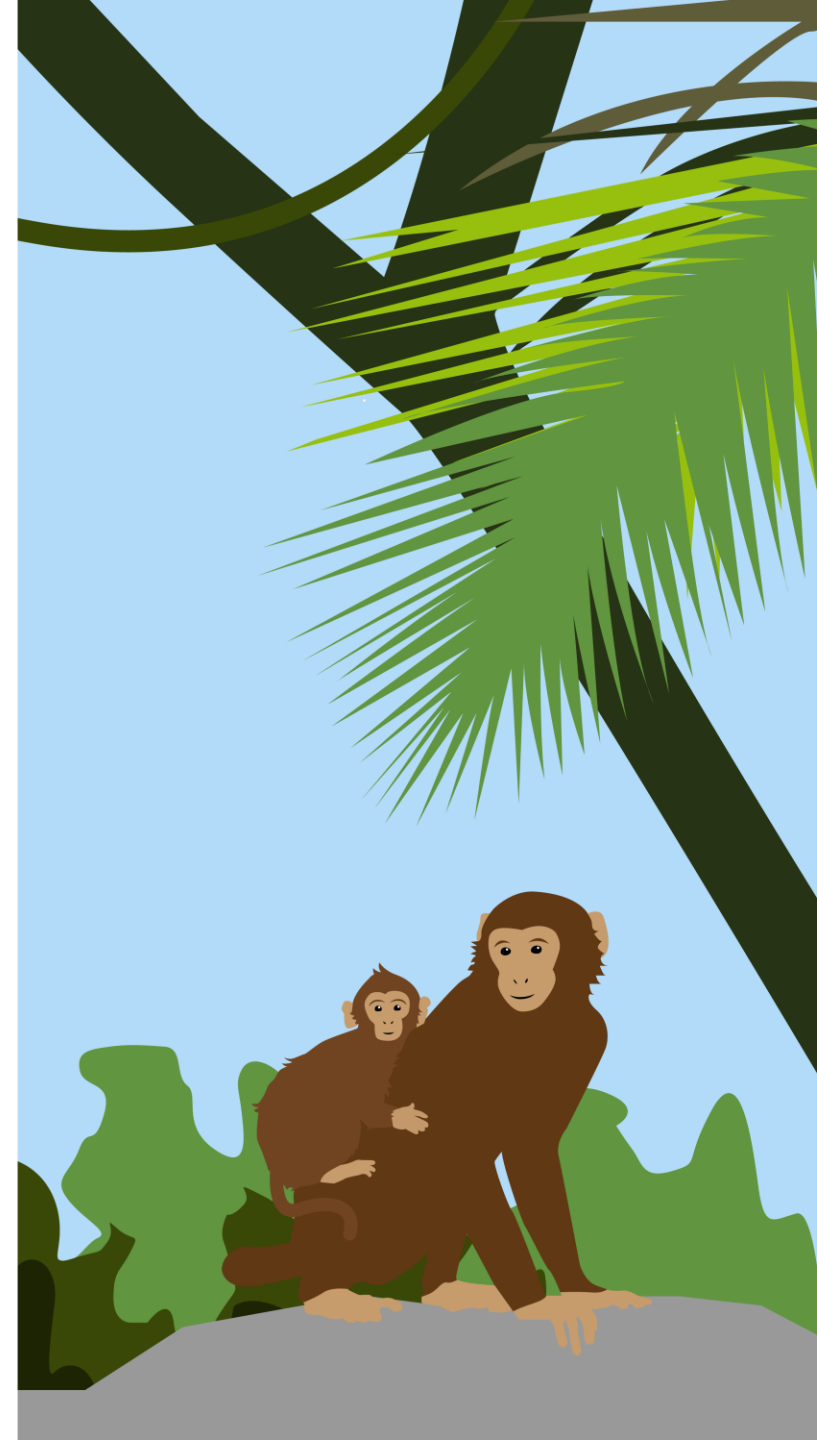
Beveiliging / Access control

- Module wordt bijgewerkt
 - Input vanuit werkgroep orkestratie
 - Input vanuit werkgroep beveiliging
- Module integreren in developer.overheid.nl

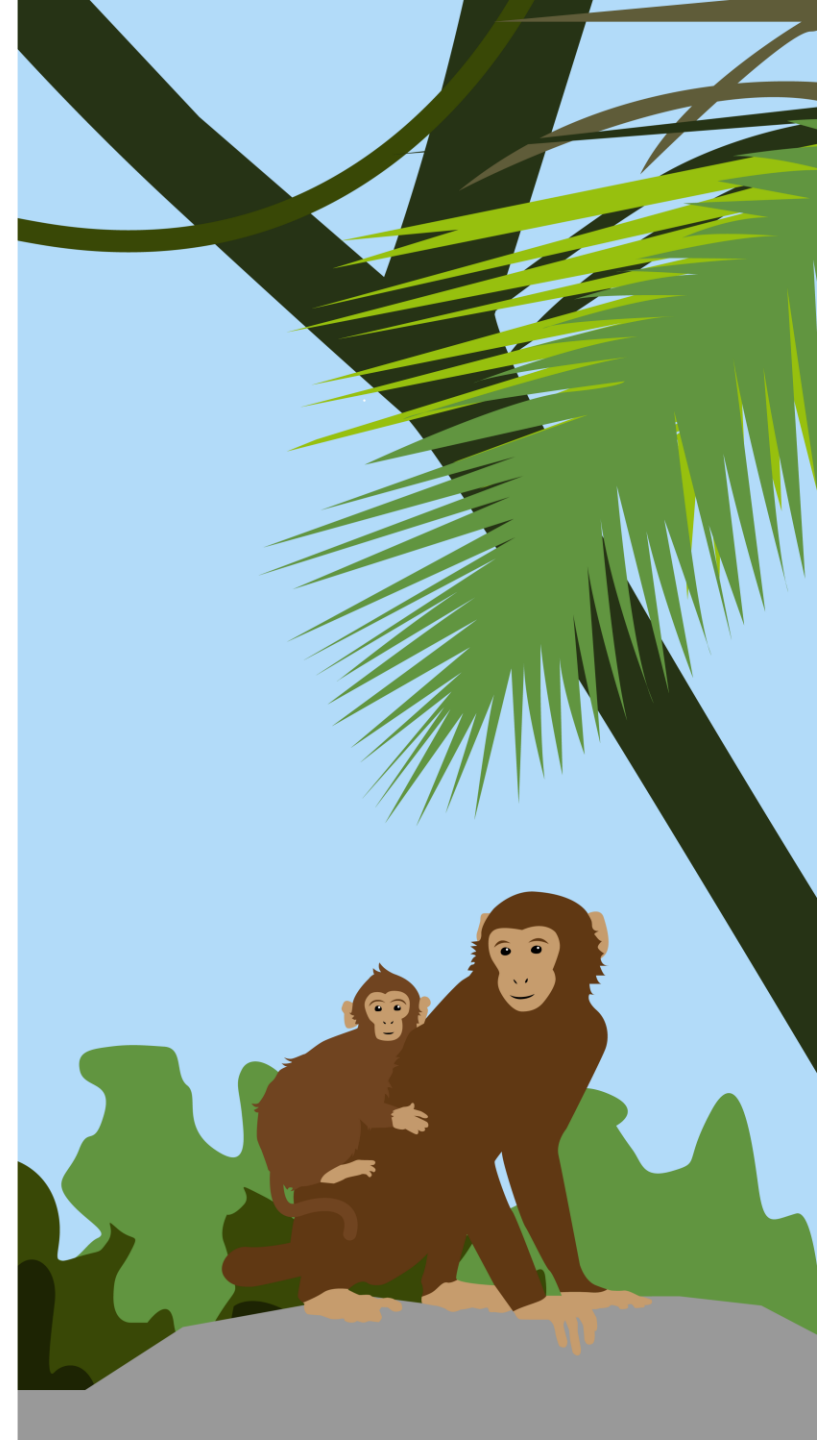
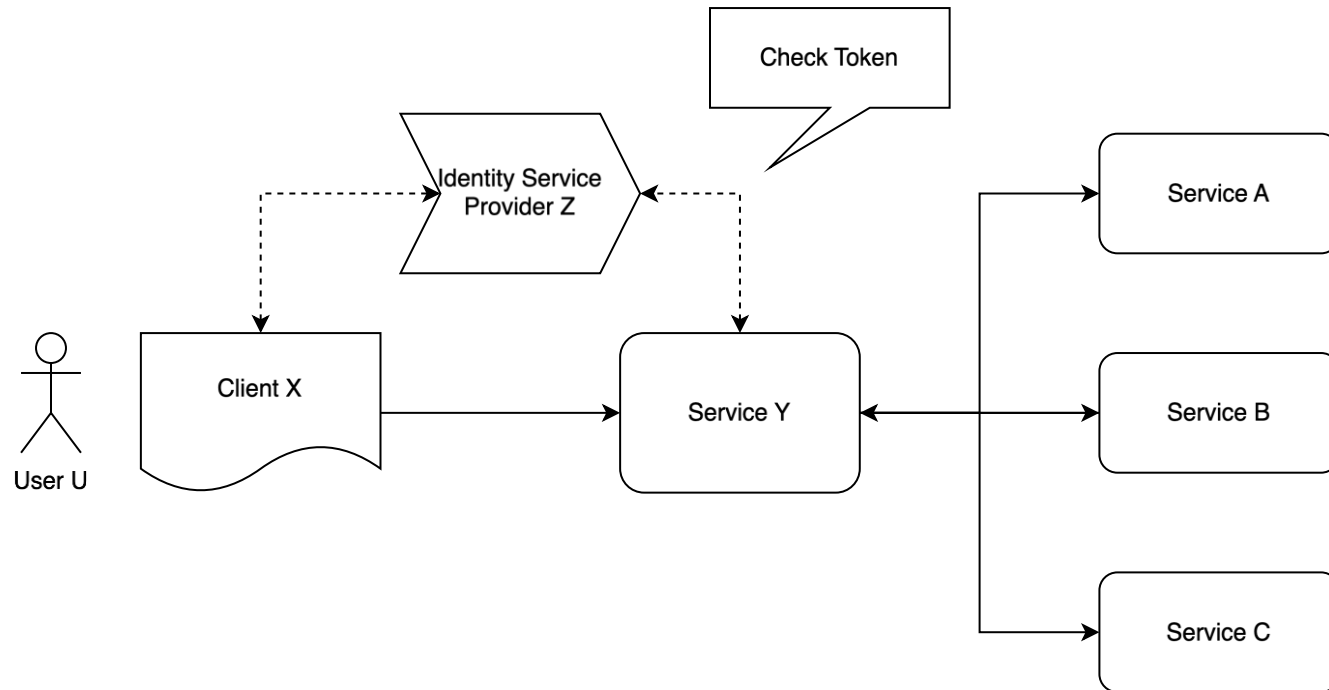


Input vanuit Orkestratie

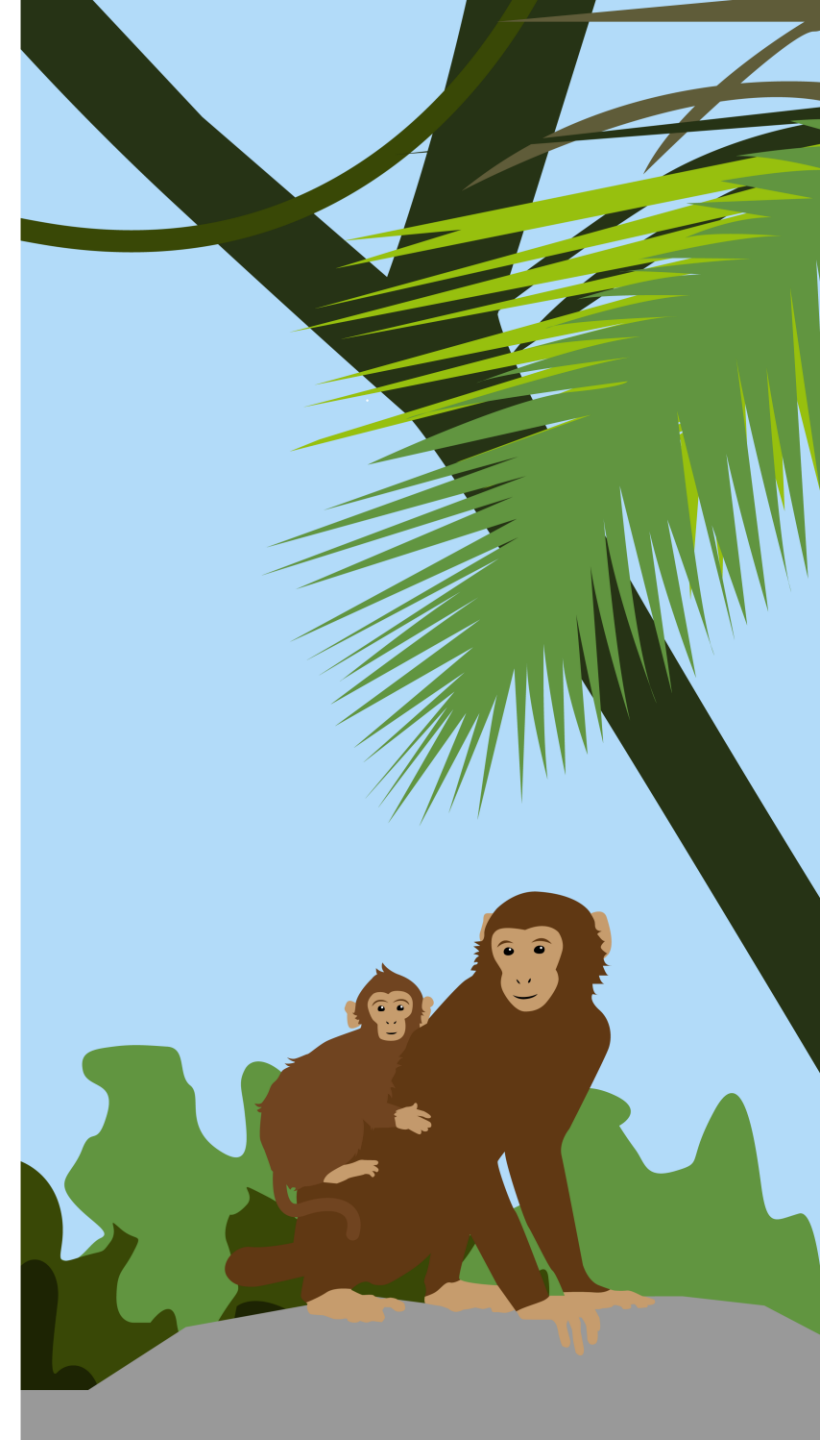
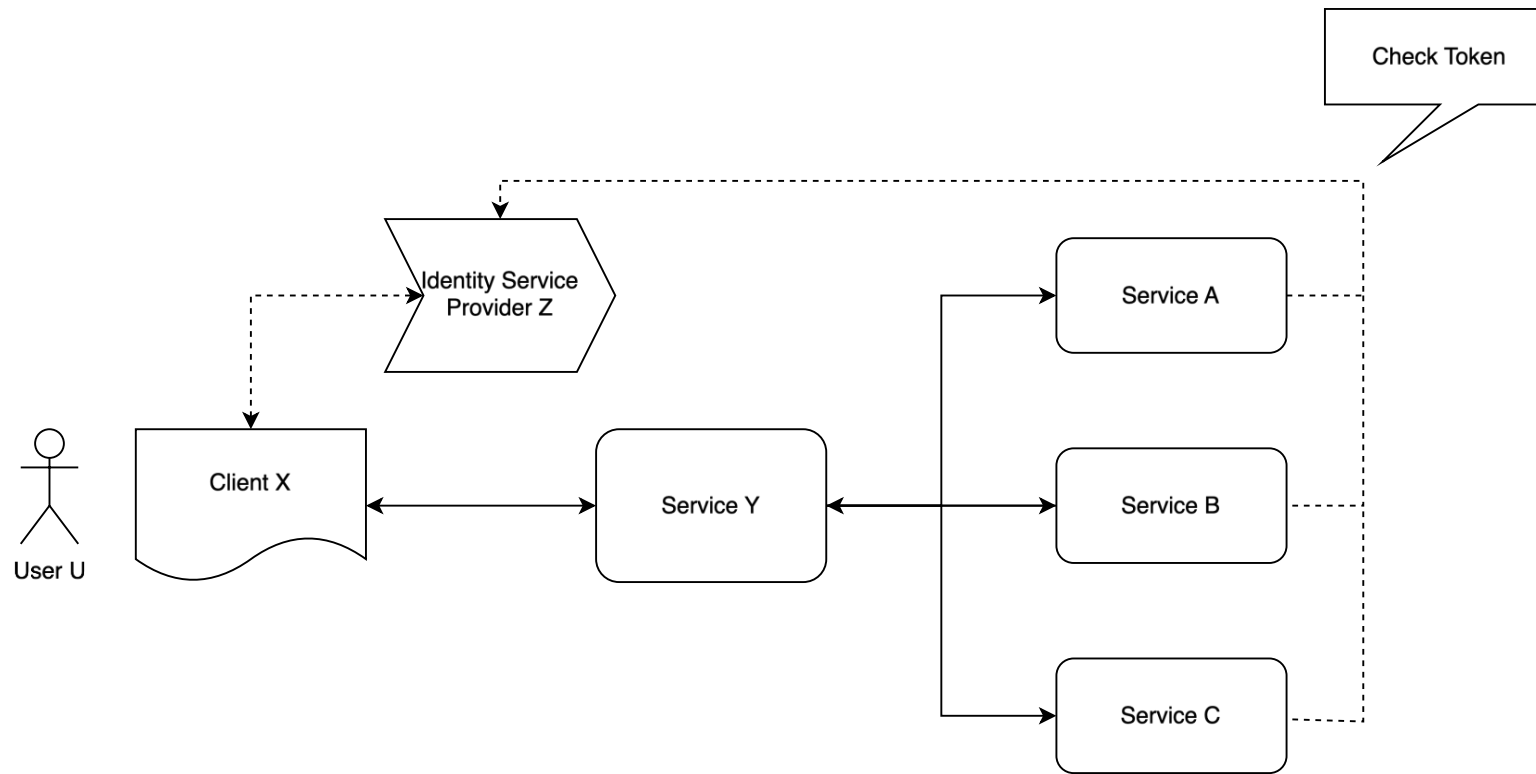
- Toevoegen patronen:
 - Intermediairs
 - Transparant
 - Niet transparant (Opaque)
 - Orchestration
 - Implicit identity re-use
 - Identity propagation
 - Federated identity propagation



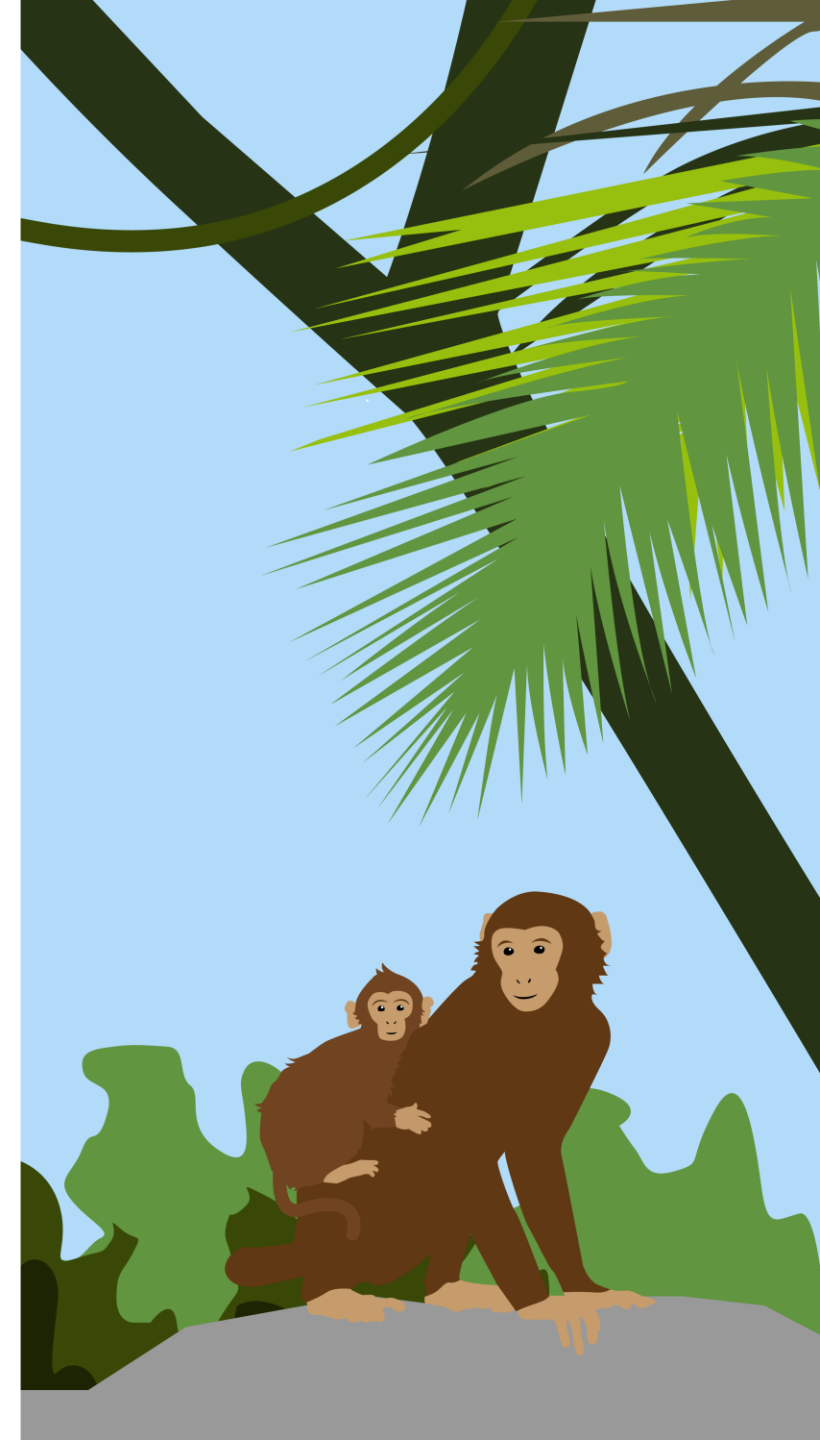
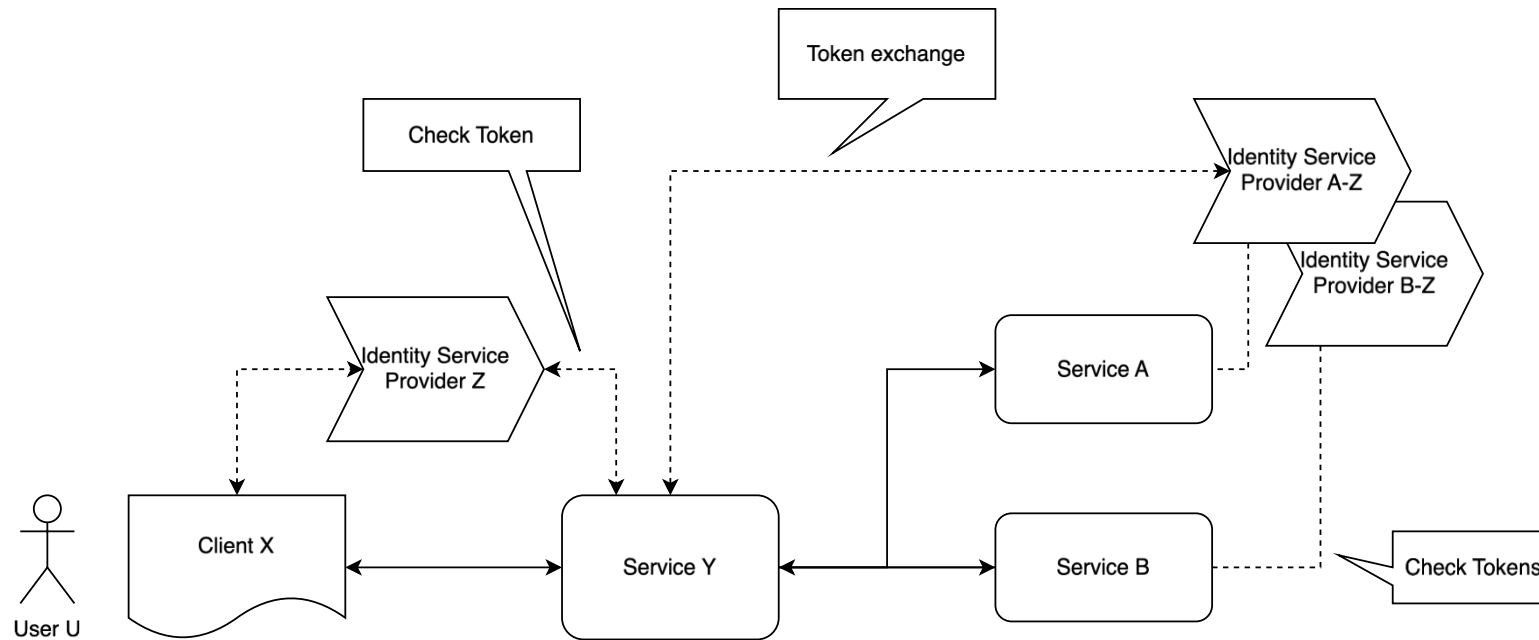
Implicit identity reuse



Identity propagation

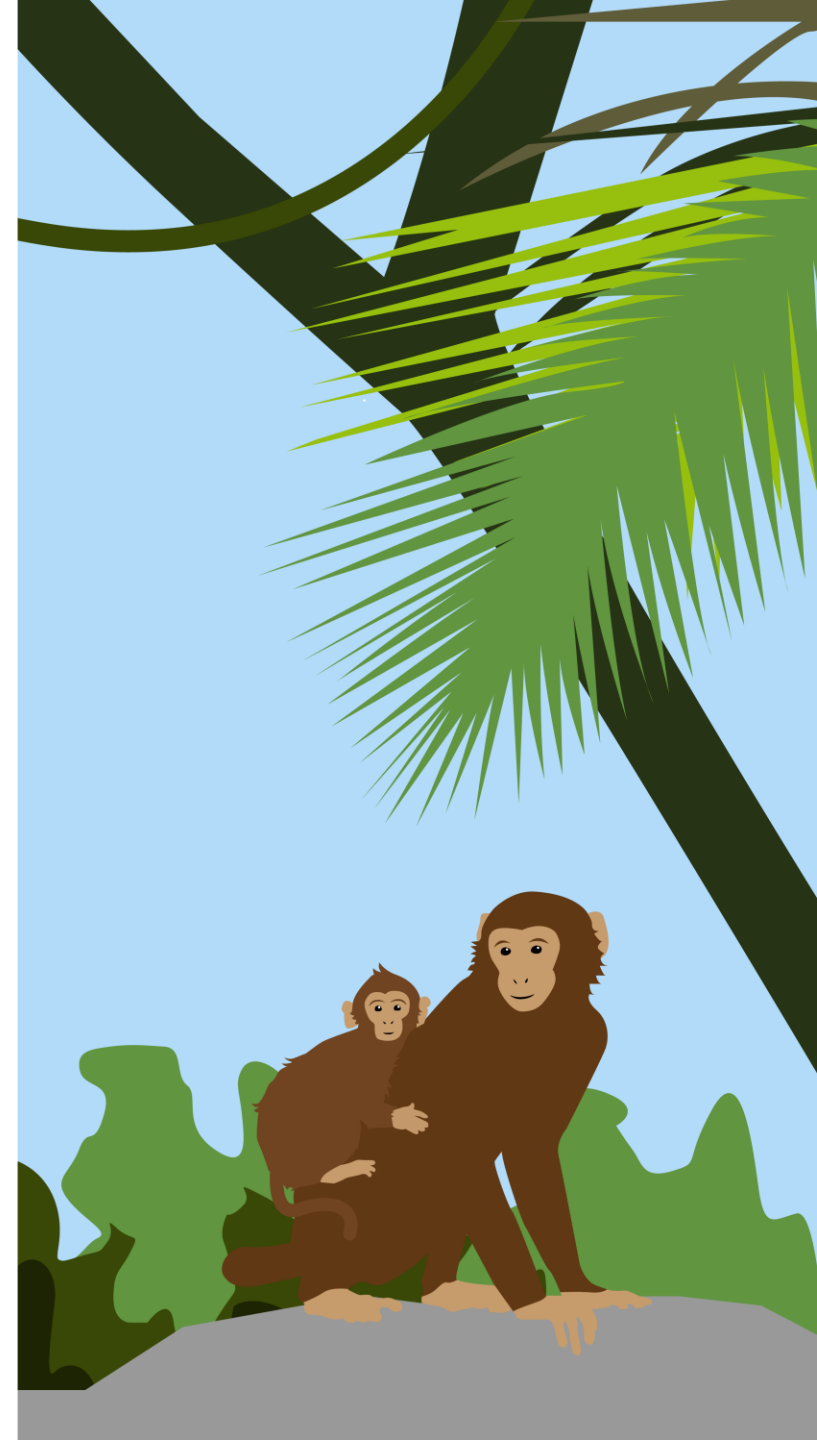


Federated identity propagation



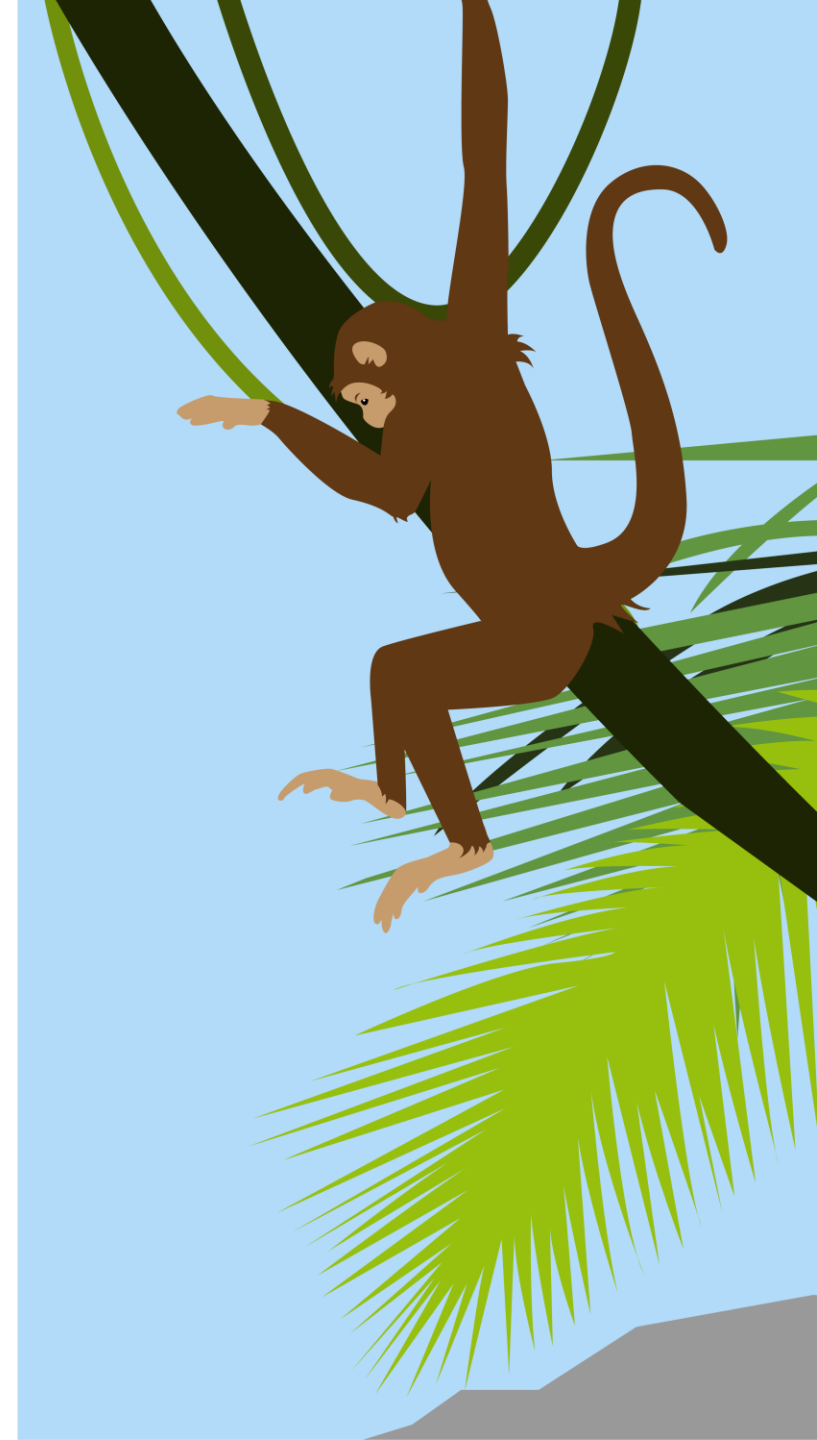
Input vanuit werkgroep beveiliging

- Verwijzing naar standaard Federatieve Toegang verlening (AuthZen)
- Verwijzing naar Signing & Encryptie modules
- API Access layers



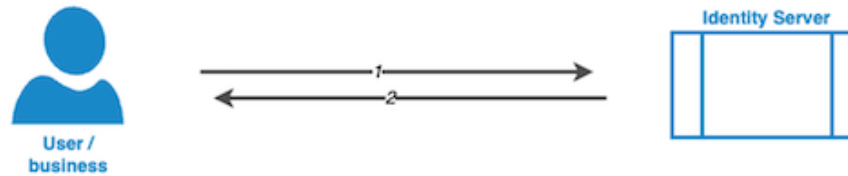
Federatieve toegangsverlening

- Standaard voor toegangsverlening (communicatie tussen PeP en PdP)
- Gebaseerd op AuthZen
- Momenteel in consultatie
 - <https://www.logius.nl/actueel/publieke-consultatie-nlgov-authzen-authorization-api-v10>

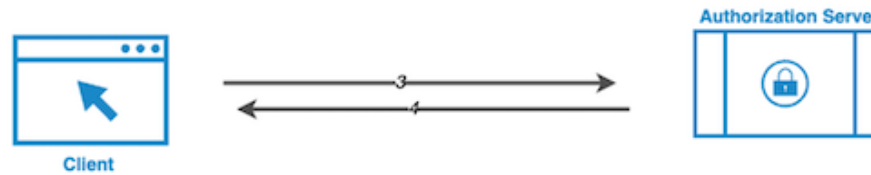


API Access layers

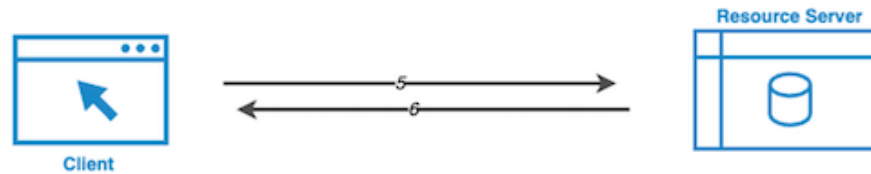
Layer - onboarding



Layer - Client registration



Layer - Runtime Resource access



Integratie developer.overheid.nl

developer.overheid.nl

Ontwikkelaarsportaal van de Nederlandse overheid

Home

Kennisbank

Communities

Blog

API's

Open Source

Open Data

Geodata

Kennisbank

Security

Standaarden

Tools

OpenKAT

Best practices

De drie lagen van API-toegang

Runtime toegangscontrolle en foutafhandeling

>

Kennisbank

>

Best practices

Best practices voor API-toegangsbeheer

Deze pagina is een eerste voorstel om de Geonovum-module [API Access Control](#) om te vormen tot een reeks best practices binnen de kennisbank Security. Doel: de inhoud beter vindbaar maken voor Nederlandse overheden en te laten aansluiten op de opzet van developer.overheid.nl.

In deze testopzet laten we zien hoe twee onderdelen van de module er in DON-stijl uit kunnen zien:

1. [De drie lagen van API-toegang](#) – van onboarding tot runtime-controle.
2. [Runtime toegangscontrolle en transportbeveiliging](#).

Dit is een conceptversie voor review. Feedback op structuur, toon of extra context is welkom voordat we de overige hoofdstukken migreren.

Afbakening en licentie

- Inhoudsbron: Geonovum, [API Access Control module](#) (laatste versie geraadpleegd op 2025-10-16).
- Licentie: [Creative Commons Zero \(CC0\)](#), zoals in de oorspronkelijke repository vermeld.
- Aanpassingen: tekstueel geherstructureerd, vertaald naar het Nederlands en aangevuld met relevante links naar DON en Nederlandse standaarden.

Iets aan dit artikel verbeteren?

Heb je nog andere ideeën of suggesties?

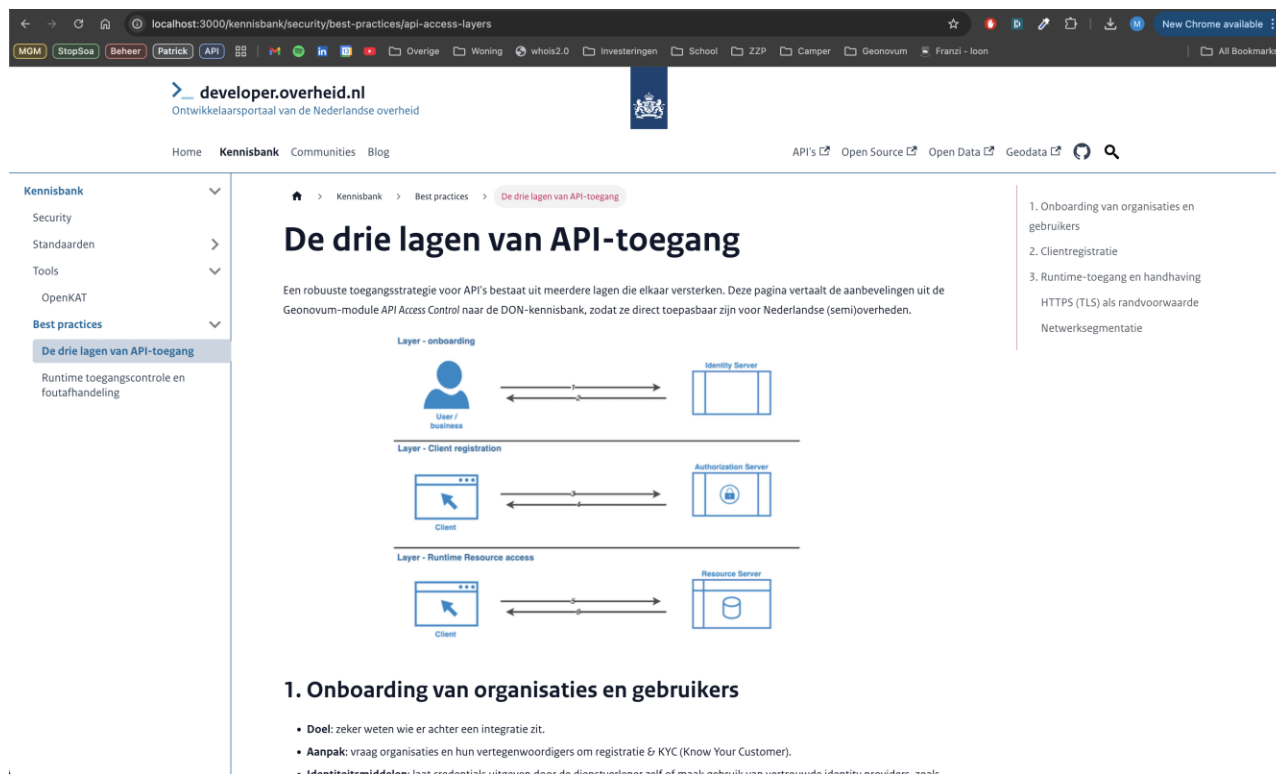
Vorige

< OpenKAT

Volgende

De drie lagen van API-toegang >

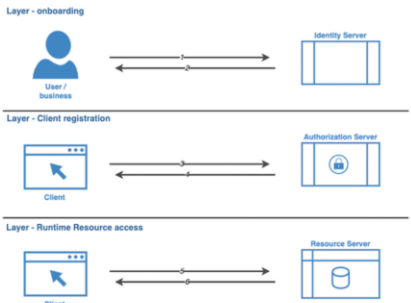
Integratie developer.overheid.nl



The screenshot shows a web browser at localhost:3000/kennisbank/security/best-practices/api-access-layers. The page title is 'developer.overheid.nl' with the subtitle 'Ontwikkelaarsportaal van de Nederlandse overheid'. The navigation bar includes 'Home', 'Kennisbank', 'Communities', and 'Blog'. The 'Kennisbank' sidebar lists 'Security', 'Standaarden', 'Tools', and 'OpenKAT'. The 'Best practices' section is active, showing 'De drie lagen van API-toegang'. The main content area features the article title 'De drie lagen van API-toegang' and a brief description: 'Een robuuste toegangsstrategie voor API's bestaat uit meerdere lagen die elkaar versterken. Deze pagina vertaalt de aanbevelingen uit de Geonovum-module API Access Control naar de DON-kennisbank, zodat ze direct toepasbaar zijn voor Nederlandse (semi)overheden.' The diagram illustrates three layers: 1. Layer - onboarding (User / business to Identity Server), 2. Layer - Client registration (Client to Authorization Server), and 3. Layer - Runtime Resource access (Client to Resource Server). The right sidebar lists the three layers: 1. Onboarding van organisaties en gebruikers, 2. Clientregistratie, and 3. Runtime-toegang en handhaving (HTTPS (TLS) als randvoorwaarde, Netwerksegmentatie).

De drie lagen van API-toegang

Een robuuste toegangsstrategie voor API's bestaat uit meerdere lagen die elkaar versterken. Deze pagina vertaalt de aanbevelingen uit de Geonovum-module API Access Control naar de DON-kennisbank, zodat ze direct toepasbaar zijn voor Nederlandse (semi)overheden.



Layer - onboarding

User / business ↔ Identity Server

Layer - Client registration

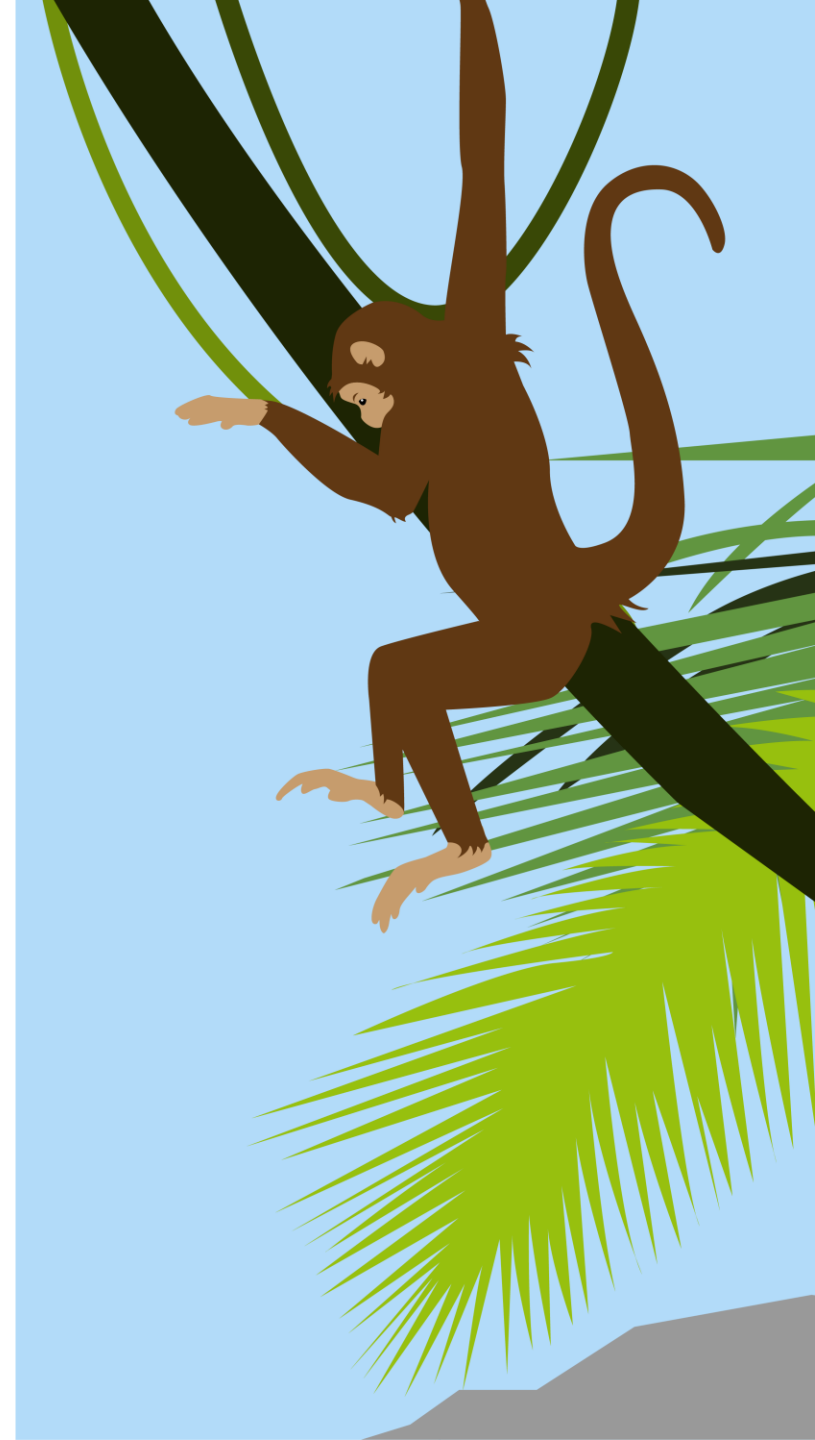
Client ↔ Authorization Server

Layer - Runtime Resource access

Client ↔ Resource Server

1. Onboarding van organisaties en gebruikers

- **Doel:** zeker weten wie er achter een integratie zit.
- **Aanpak:** vraag organisaties en hun vertegenwoordigers om registratie & KYC (Know Your Customer).
- **Identiteitsmiddelen:** laat credentials uitvoeren door de dienstverlener zelf of maak gebruik van vertrouwde identiteits providers, zoals



Integratie developer.overheid.nl



developer.overheid.nl

Ontwikkelaarsportaal van de Nederlandse overheid



Home

Kennisbank

Communities

Blog

API's

Open Source

Open Data

Geodata





Kennisbank

Security

Standaarden

Tools

OpenKAT

Best practices

De drie lagen van API-toegang

Runtime toegangscontrole en foutafhandeling

Runtime toegangscontrole en foutafhandeling

Tijdens runtime bepaalt de API of een verzoek daadwerkelijk mag worden uitgevoerd. Onderstaande aanbevelingen helpen om tokens veilig te verwerken, consistente foutcodes terug te geven en logging op orde te houden.

Tokens als enige bron van waarheid

- Stateless:** baseer autorisatie op het meegeleverde token, niet op serversessies.
- Formaat:** pas waar mogelijk OAuth 2.0 of OpenID Connect toe; SAML 2.0 is een alternatief voor federatieve ketens.
- Delegatie:** gebruik OAuth-scopes om rechten te begrenzen. Zie ook de [NL GOV Assurance-profielen](#).

Headers voor geautoriseerde en niet-geautoriseerde clients

Scenario	Headervoorbeeld	Toelichting
Geautoriseerd	Authorization: Bearer <token>	OAuth 2.0 bearer token met scopes/claims.
Niet-geautoriseerd	X-API-Key: <api-key>	Alleen gebruiken voor beperkt risico en read-only.

Zet tokens nooit in queryparameters: webservers loggen deze vaak en vergroten zo het risico op datalekken.

Foutafhandeling: 401, 403 of 404?

Geef zo min mogelijk informatie prijs, maar wees wel voorspelbaar voor legitieme afnemers:

Tokens als enige bron van waarheid

Headers voor geautoriseerde en niet-geautoriseerde clients

Foutafhandeling: 401, 403 of 404?

HTTPS en netwerkbeveiliging

Logging en monitoring

Headers voor geautoriseerde en niet-geautoriseerde clients

Scenario	Headervoorbeeld	Toelichting
Geautoriseerd	Authorization: Bearer <token>	OAuth 2.0 bearer token met scopes/claims.
Niet-geautoriseerd	X-API-Key: <api-key>	Alleen gebruiken voor beperkt risico en read-only.

Zet tokens nooit in queryparameters: webservers loggen deze vaak en vergroten zo het risico op datalekken.

Foutafhandeling: 401, 403 of 404?

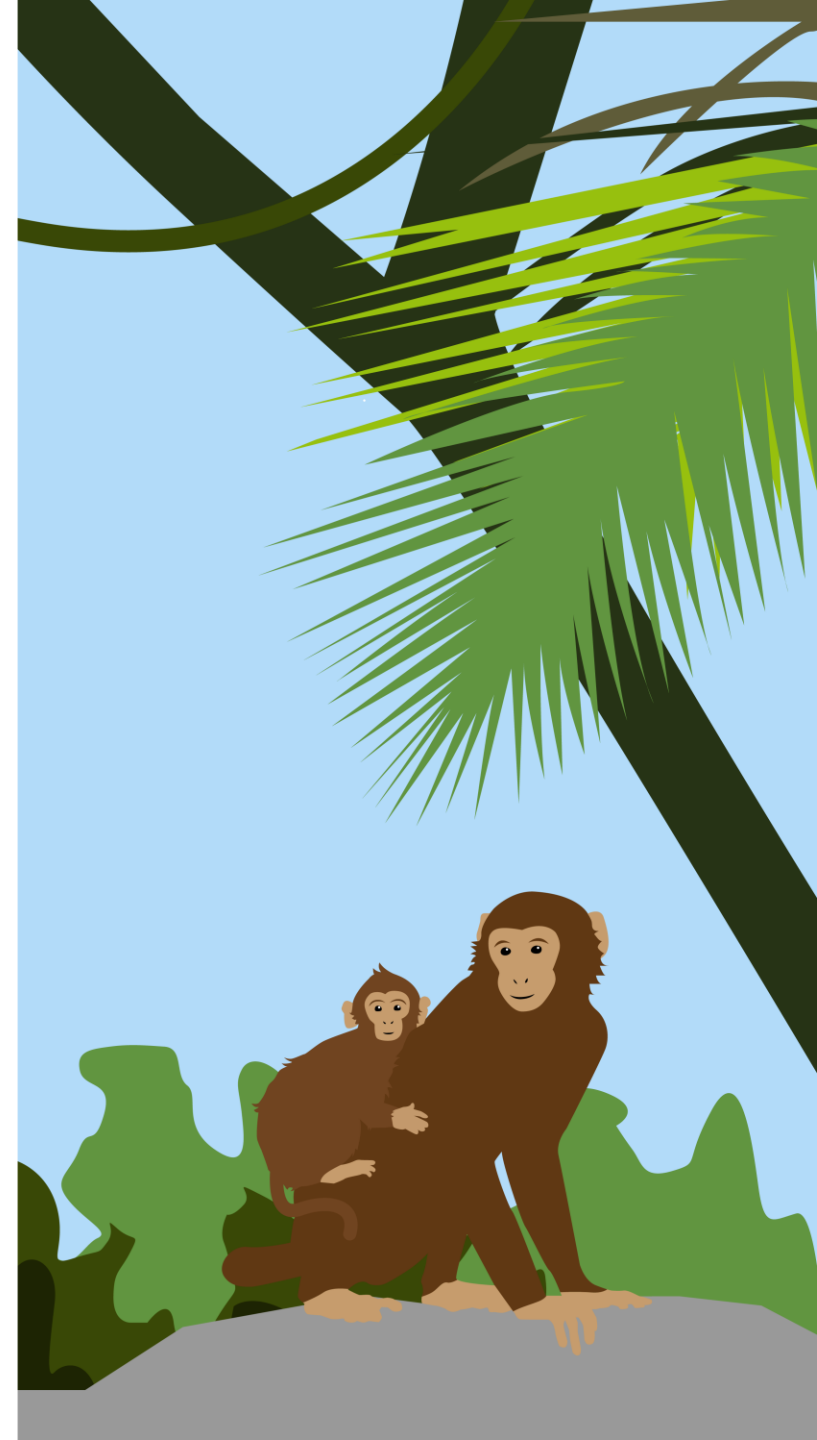
Geef zo min mogelijk informatie prijs, maar wees wel voorspelbaar voor legitieme afnemers:



Batching module

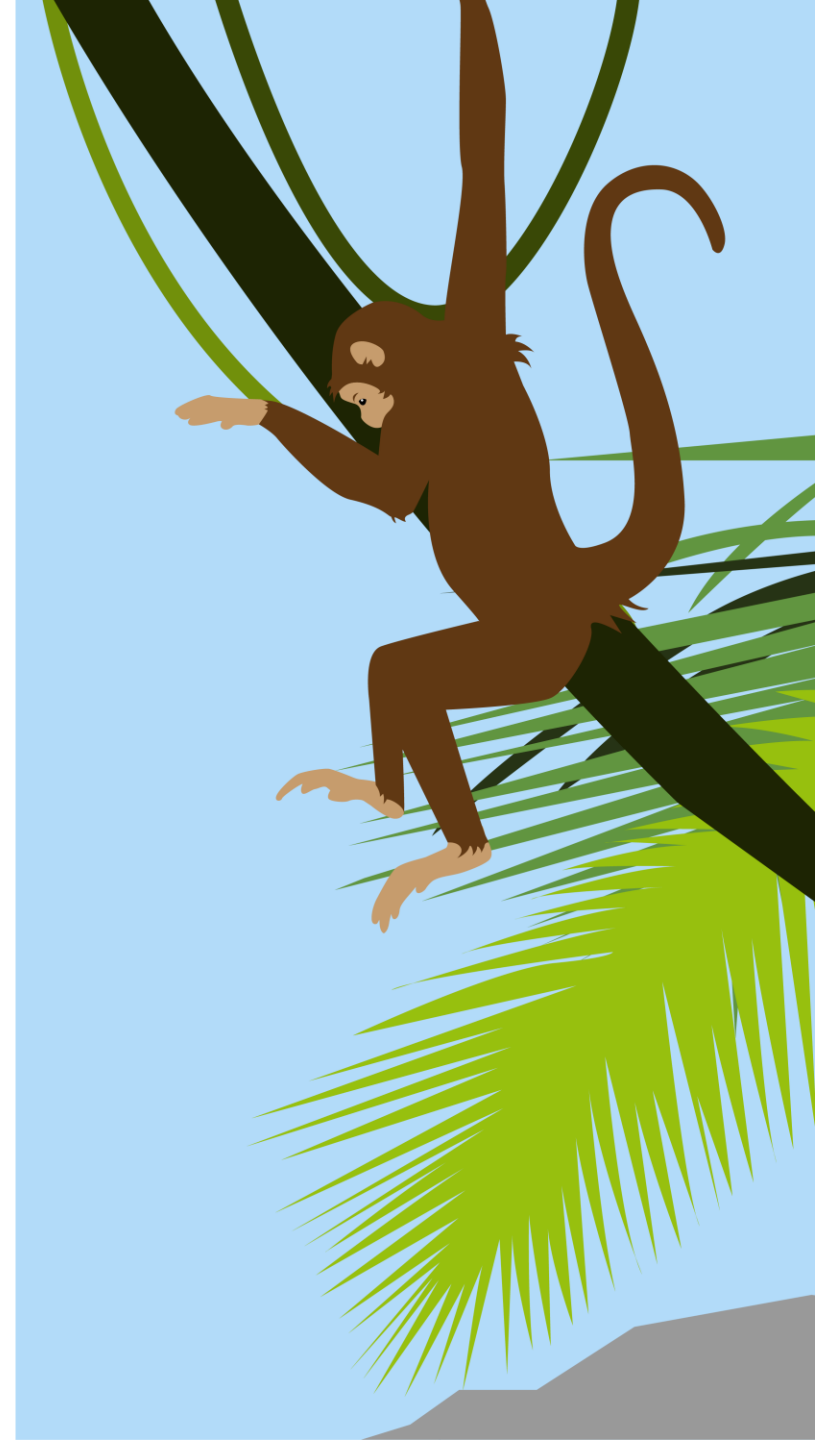
Wat is batching?

Batching is een patroon waarbij meerdere afzonderlijke API-verzoeken in één enkel request worden gebundeld.



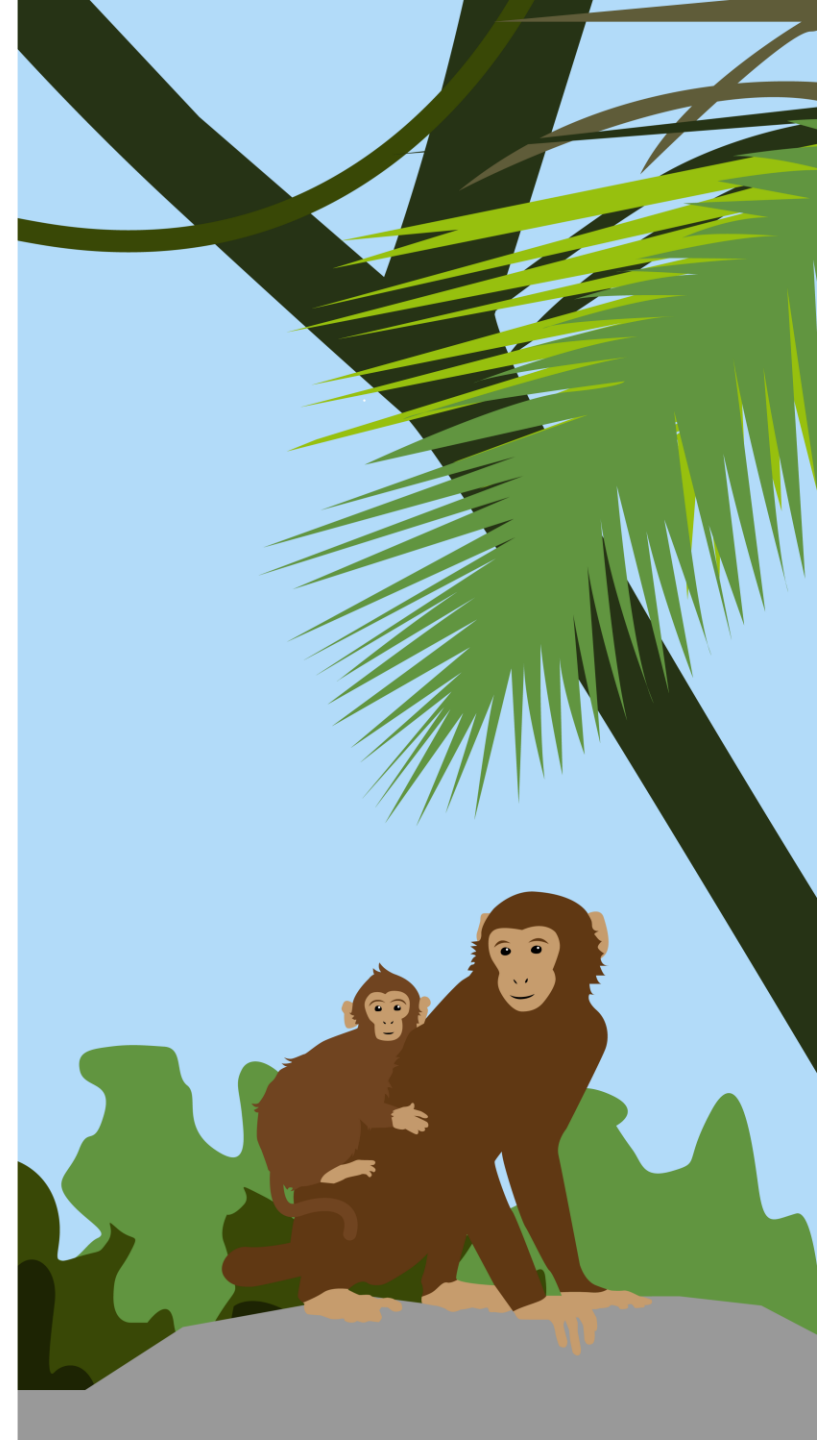
Use cases

- Orkestratie over gedistribueerde databronnen
- Synchronisatie extern systeem
- Parallele evaluatie van collectie filters



Voordelen

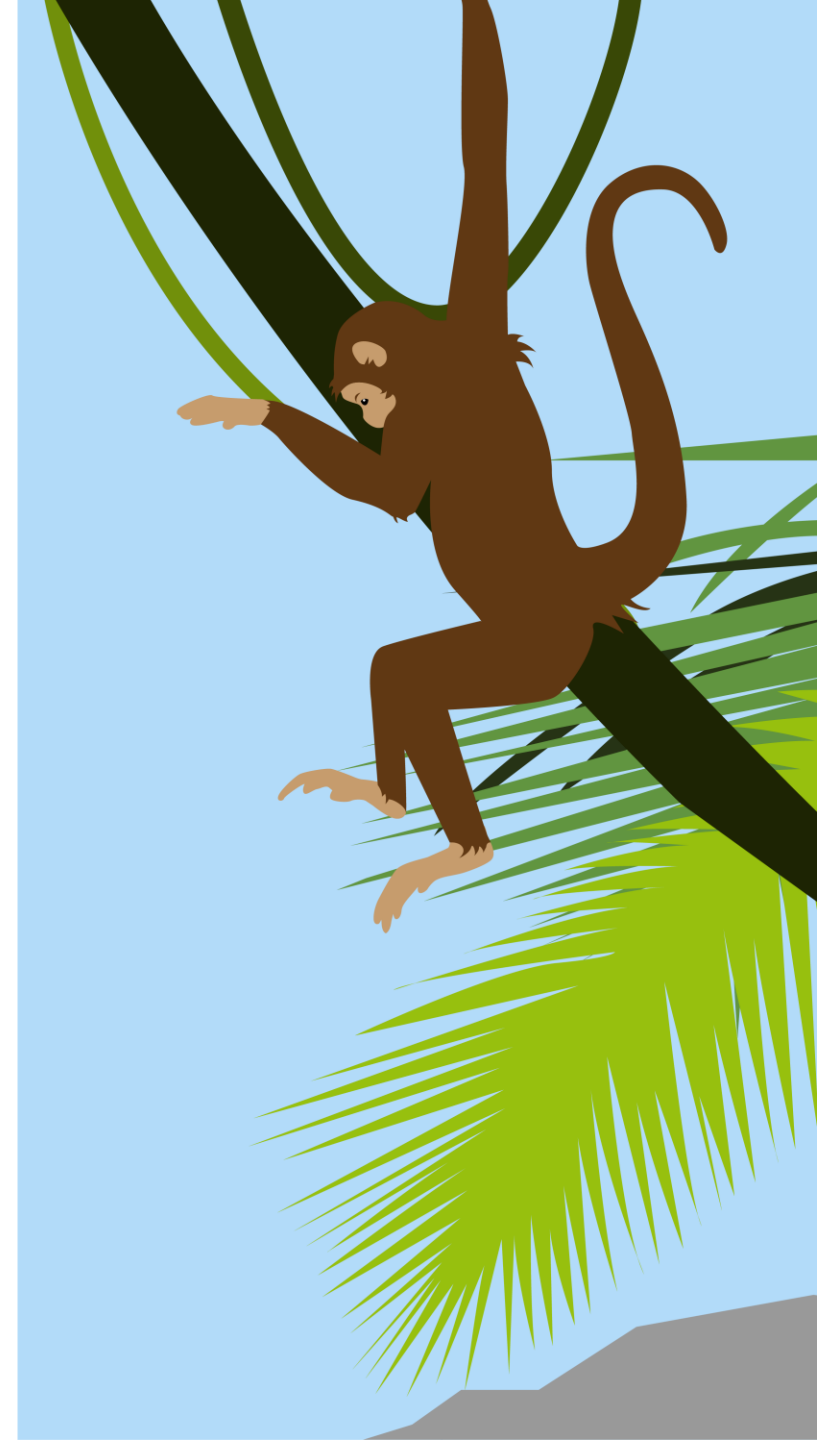
- Voorkomt het N+1 probleem
- Minder netwerkverkeer, hogere performance
- Minder belasting voor aanbieder
- Lagere complexiteit door minder "chatter"



Batch request

- Endpoint: POST /<collectie>/_batch
- Request bevat `requests[]`

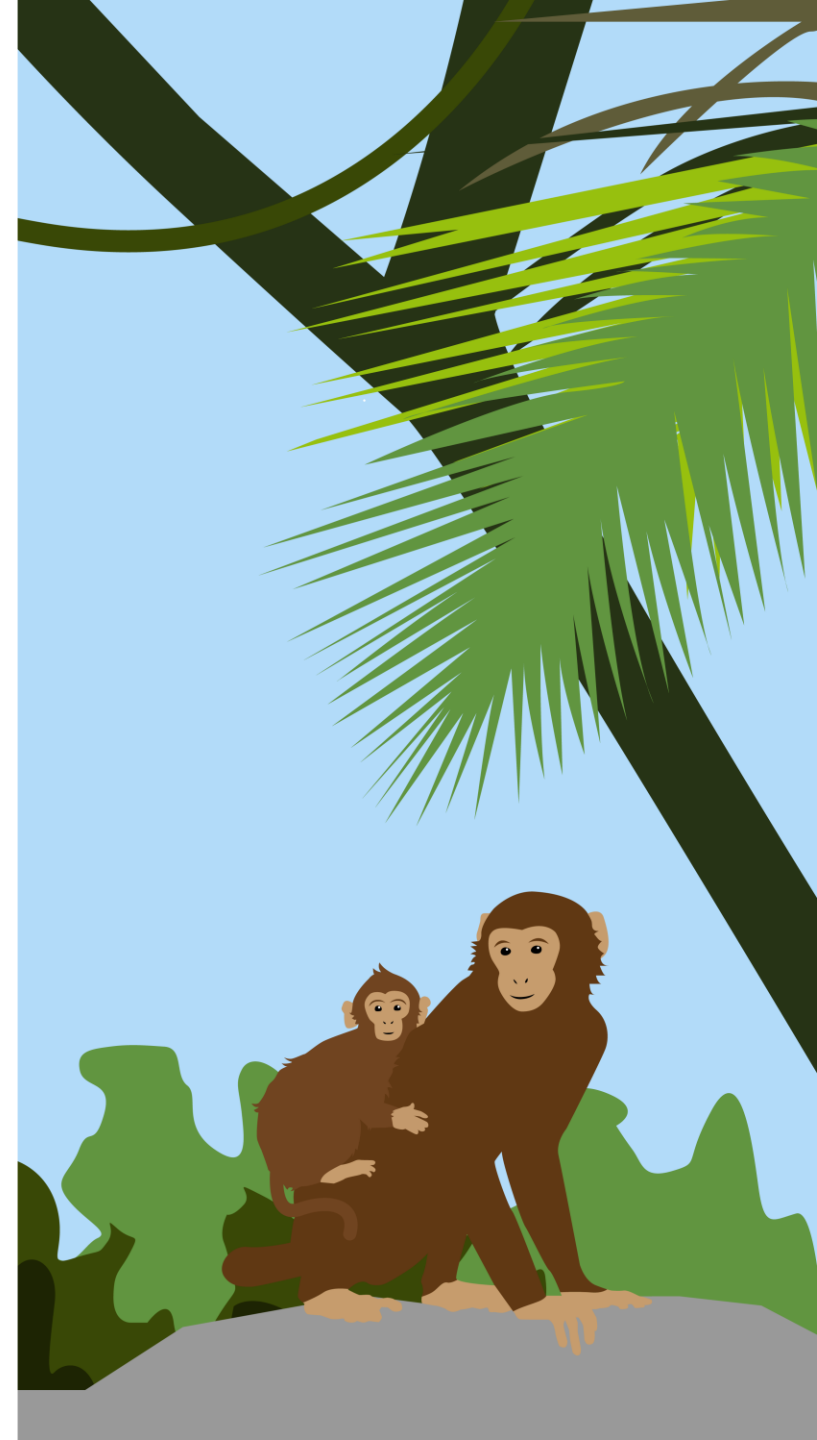
```
// POST https://api.example.org/v1/gebouwen/_batch
{
  "requests": [
    { "key": "3b9710c4-6614-467a-ab82-36822cf48db1" },
    { "key": "609b0651-acad-4091-9144-432621df8bf8" }
  ]
}
```



Batch response

- Response bevat `results[]`
- Volgorde gelijk aan `requests[]`
- `null` betekent niet bestaand/geautoriseerd

```
{  
  "results": [  
    {  
      "identificatie": "3b9710c4-6614-467a-ab82-36822cf48db1",  
      "naam": "Stadhuis",  
      "bouwjaar": 1978  
    },  
    null  
  ]  
}
```



Design rules

/batching/path: Expose a standardized batch endpoint for collections

/batching/req-format: Use the standardized payload format for batch requests

/batching/res-format: Use the standardized payload format for batch responses

/batching/req-singular: Specify a list of singular requests when batching singular resources

/batching/res-singular: Return a list of singular results when batching singular resources

/batching/req-collection: Specify a list of collection requests when batching collection resources

/batching/res-collection: Return a list of collection results when batching collection resources

/batching/err-req-invalid: Reject invalid batch requests using problem+json

/batching/err-req-limit: Reject batch requests that exceed the maximum number of items

/batching/err-res-limit: Reject batch requests that would produce oversized responses

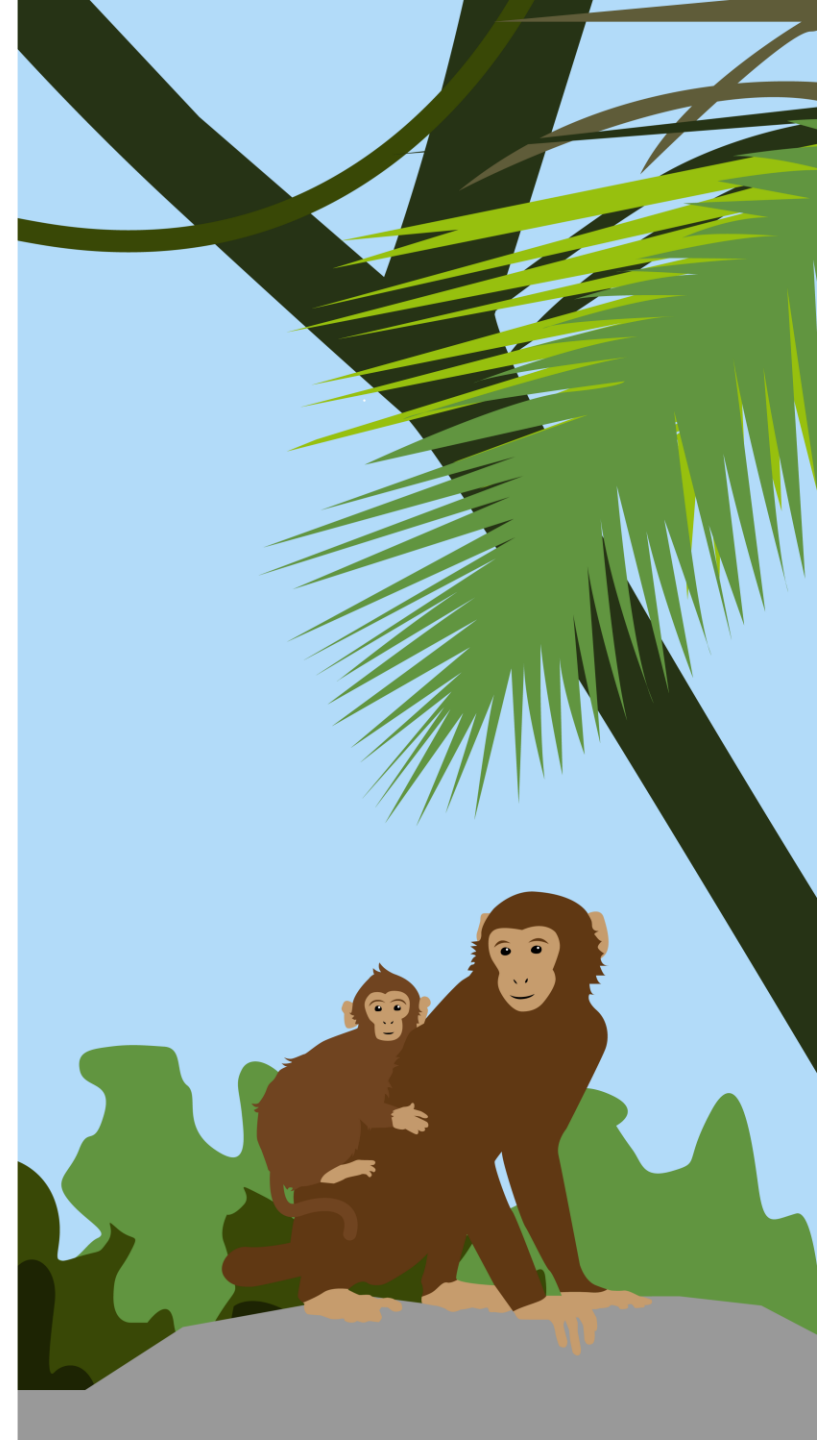
/batching/err-invalid keys: Reject singular batch requests with invalid keys

Publieke consultatie

Consultatieversie:

<https://geonovum.github.io/KP-APIs/API-strategie-modules/batching/>

Aanvang: a.s. maandag (onder voorbehoud)





Vragen?