

# Workshop Web API's

door Willy Tadema, 14 februari 2019

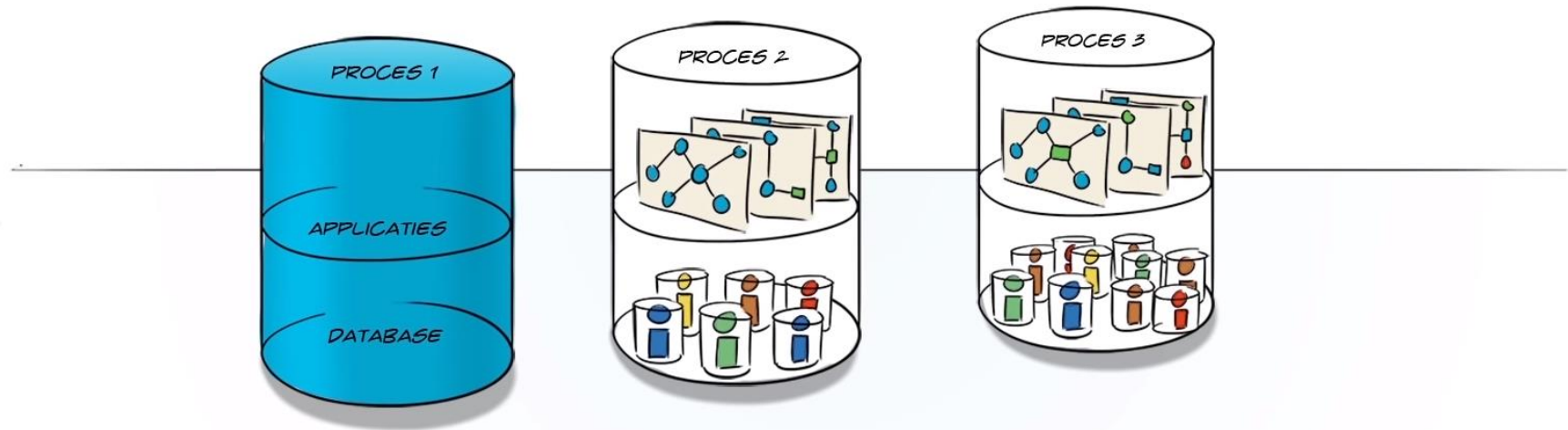
# Programma

- Introductie Web API's (Willy)
- Vernieuwde data warehouse & API's (Kevin)
- Technische uitleg (Willy)
- Zelf aan de slag!
  - ArcGIS REST API
  - PDOK LocatieServer
  - PDOK REST API
  - Dataloket API

**WHY?**

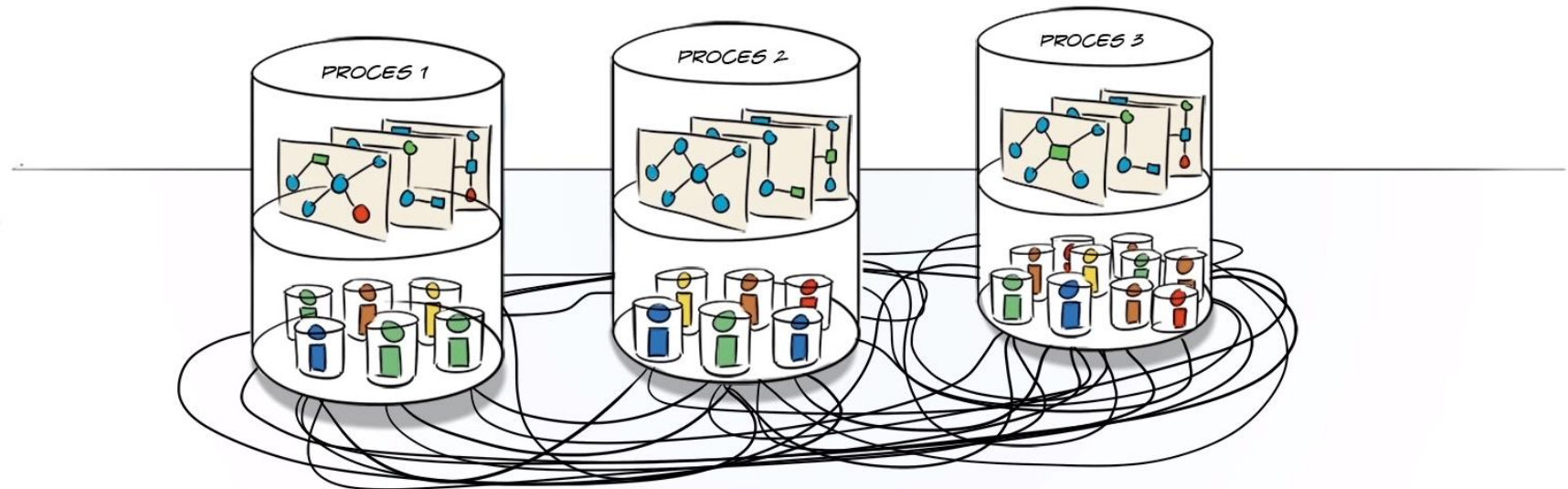
# Wat is de situatie nu?

- Informatievoorziening is vaak verknoopt met processen
- Iedere applicatie heeft een eigen database
- Zo ontstaan informatiesilo's



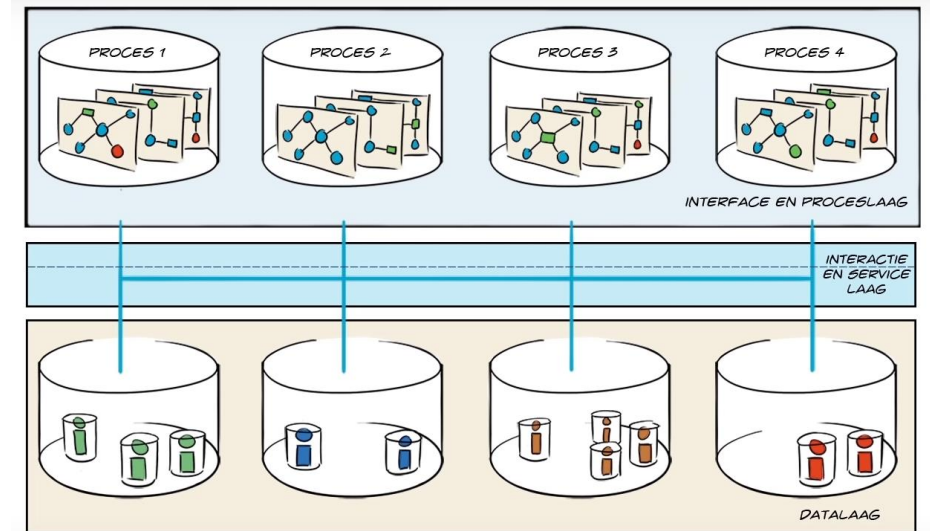
# Hoe lossen we dit in de praktijk op?

- Gegevens kopiëren van de ene naar de andere silo
- Spaghetti aan koppelingen en meervoudige opslag van data
- Foutgevoelig en inefficiënt, en belemmering voor innovatie

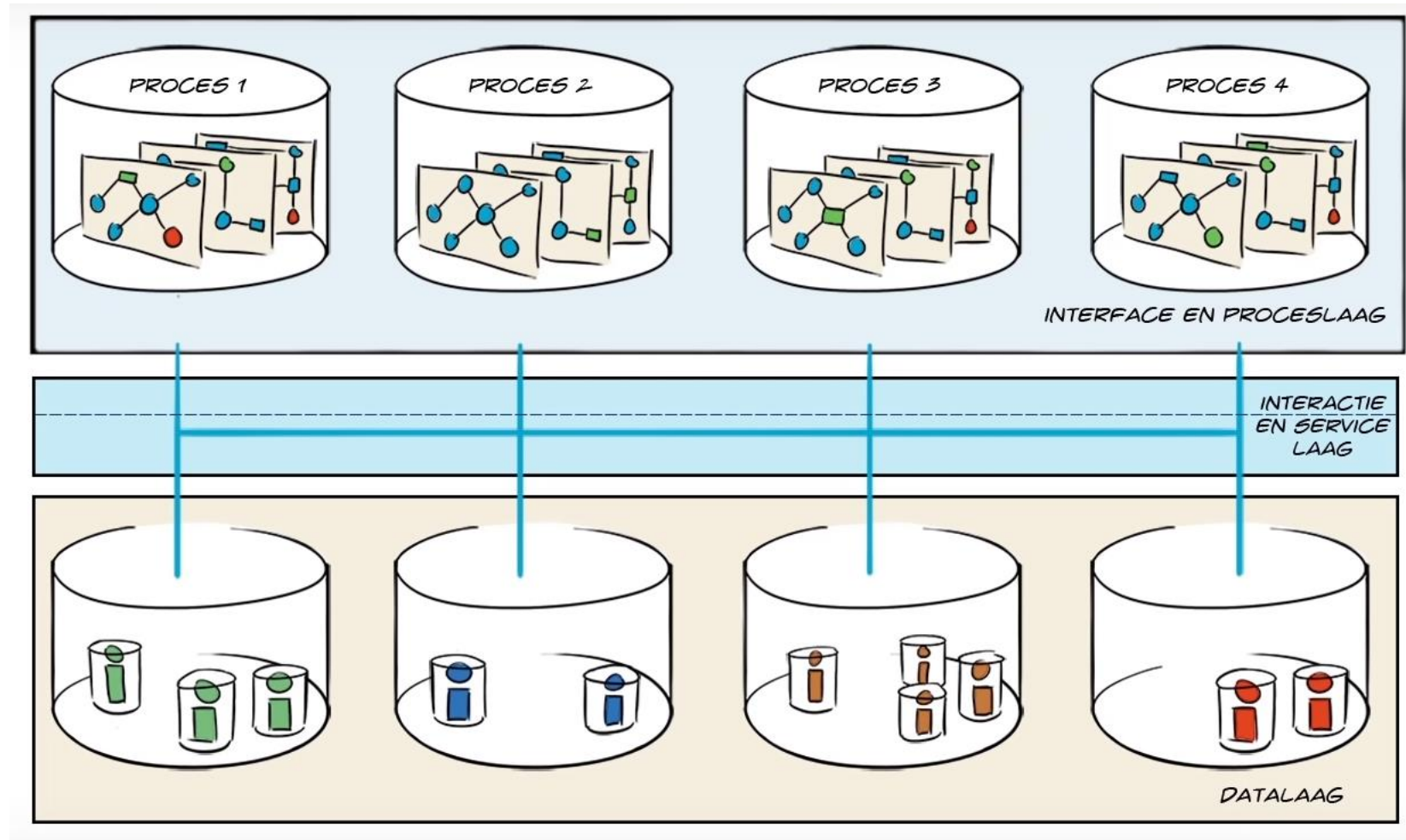


# Waar willen we naar toe?

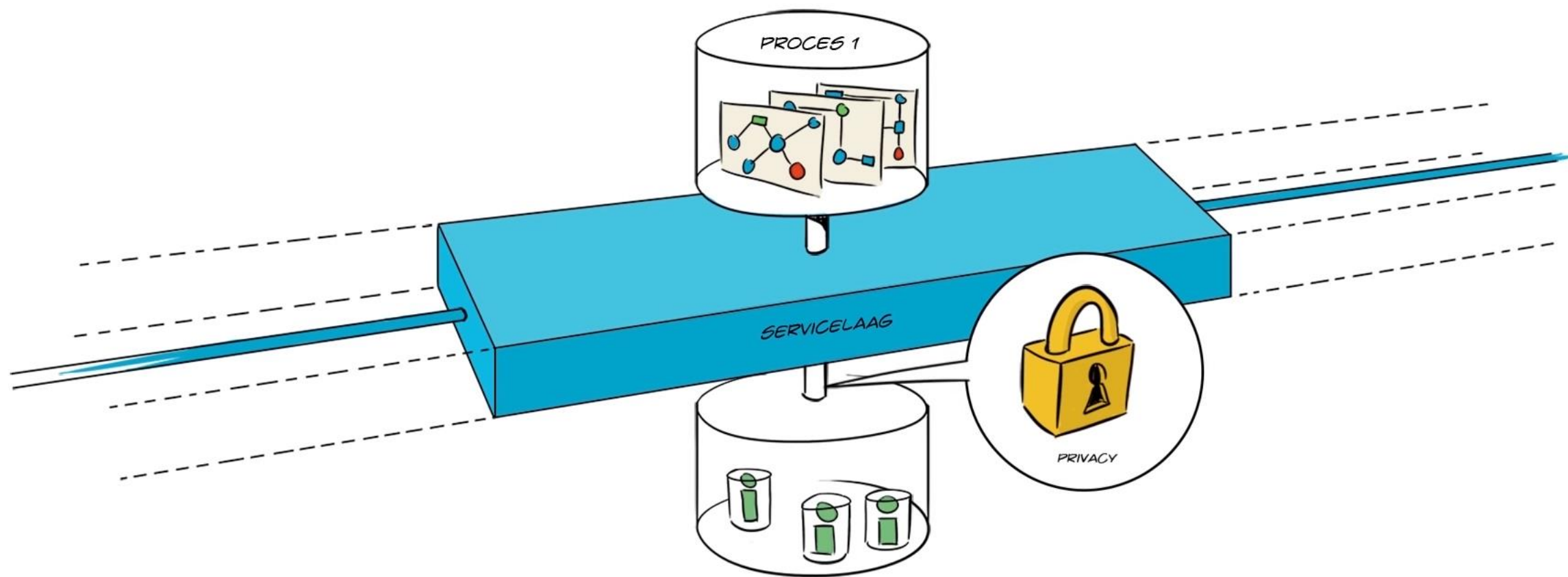
- Hergebruik van gegevens en functionaliteit
  - Eenmalig opslaan, meervoudig gebruiken
  - Eenmalig ontwikkelen, meervoudig gebruiken
- Inrichting in lagen:
  - Interface en proceslaag
  - Interactie en serviceslaag
  - Datalaag
- Informatieveiligheid en privacy waarborgen in de serviceslaag



# Inrichting gebaseerd op lagen



# Enkelvoudige opslag





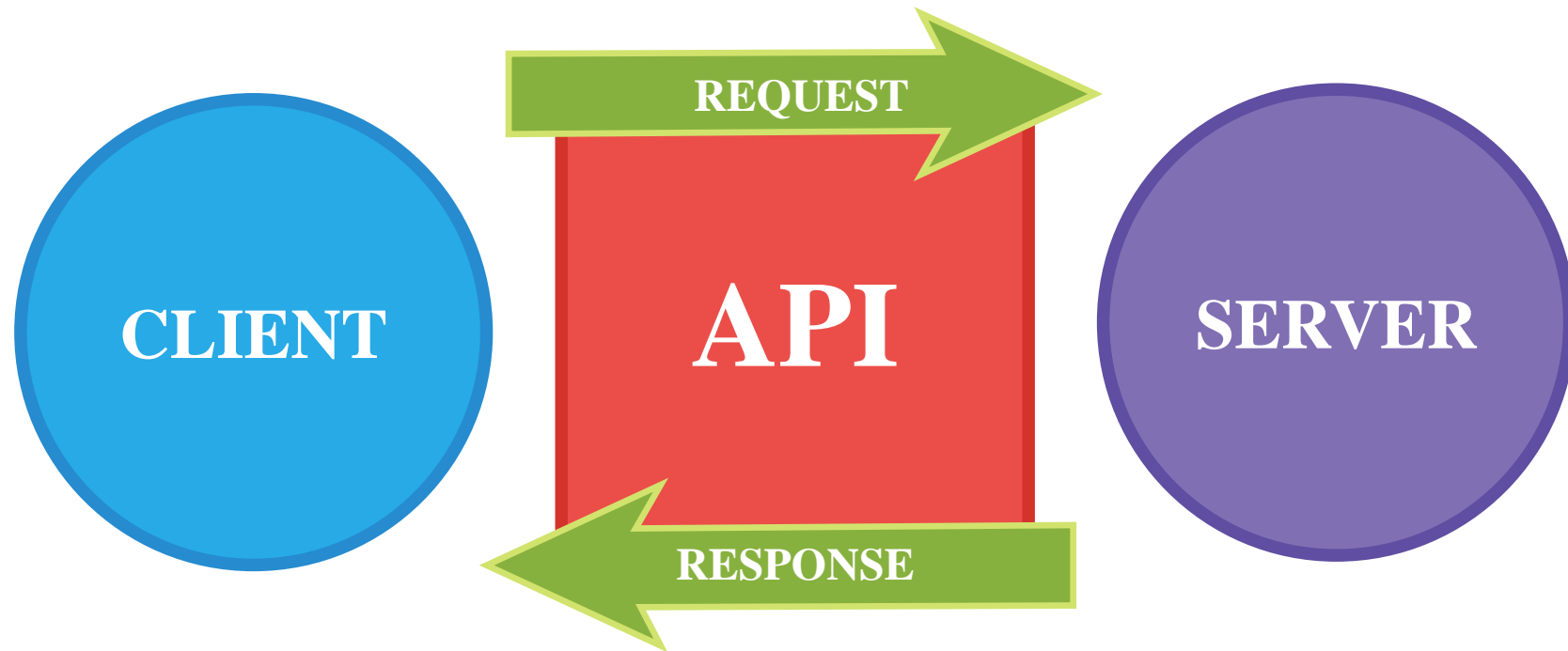
# Wat heeft dit met API's te maken?

- API is technische uitwerking van interactie en services laag
- API staat voor *Application Programming Interface*
- Een API maakt communicatie mogelijk tussen verschillende (software) systemen
- Bij een *web* API vindt uitwisseling plaats via het web (HTTP protocol)

# Waarom API's?

- Efficiënt omgaan met data
- Snel en flexibel vernieuwen
- Beheersbaar en aanpasbaar
- Waarborgen informatieveiligheid en privacy

# Communicatie via request & response

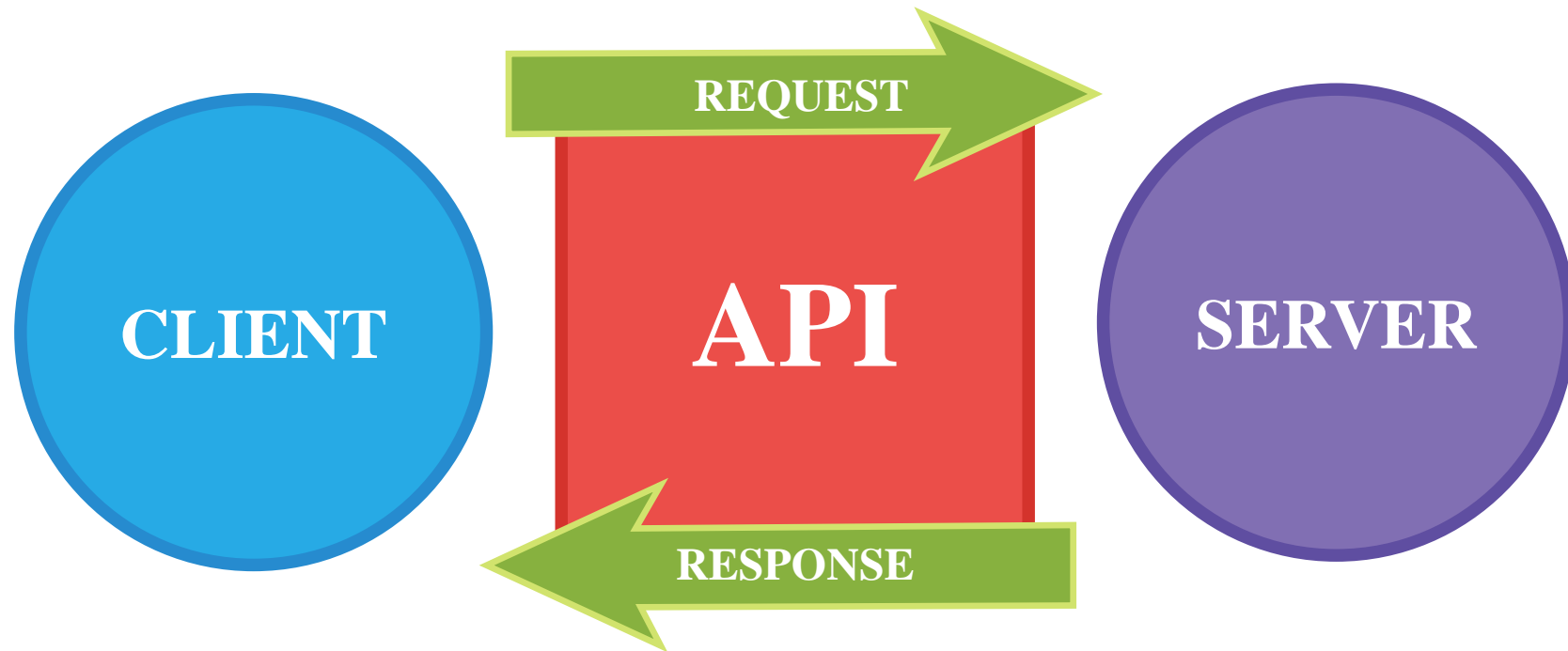


# Wanneer is een API oké?

- ~~• Het is een API en *dus* goed~~
- API's zijn géén bijproduct!
- Bij ontwerp en bouw rekening houden met hergebruik
- Generieke interface, betekenisvol voor (potentiële) afnemers
- Documentatie op orde
- Versiebeheer op orde
- Voorzieningen voor zoeken & vinden van API's



# Request & Response



# Voorbeeld request

HTTP methode                      Pad naar de resource op de webserver                      Query parameters

↓

**GET** https://dataloket.zuid-holland.nl/api/search?query=energie&offset=1

Header {

- Host: dataloket.zuid-holland.nl
- User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) ...
- Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIs ...

# Voorbeeld request





# Voorbeeld request

HTTP methode

Pad naar de resource op de webserver

**DELETE** `https://dataloket.zuid-holland.nl/api/documents/upload/9000168888`

Header {  
Host: dataloket.zuid-holland.nl  
User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) ...  
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIs ...

The diagram illustrates an HTTP DELETE request. It features three main annotations: 'HTTP methode' with a downward arrow pointing to the 'DELETE' method; 'Pad naar de resource op de webserver' with a bracket pointing to the URL; and 'Header' with a bracket pointing to the list of header fields.

