HET LOGISCH ONTWERP BRP ALS TOEGANGSBELEID

Van autorisatiebesluiten naar Linked Data policies

Werkgroep Register Toegangsbeleid — 31 maart 2026
Michiel Trimpe (FTV)

1. HET LOGISCH ONTWERP BRP

De systeemspecificatie van de Basisregistratie Personen

WAT IS HET LO BRP?

- Systeemspecificatie van RvIG voor de BRP
- Beschrijft **interfaces, berichten en voorwaarden** voor gegevensuitwisseling
- Elk kwartaal een nieuwe versie (huidig: 2026.Q1)
- Substantieve wijzigingen op 1 jan & 1 jul

> Het LO BRP is feitelijk het **beleidsregister avant la lettre** — alle autorisatieregels staan erin, inclusief de technische semantiek.

BRP DATAMODEL: DE PERSOONSLIJST

Elke persoon heeft een **Persoonslijst (PL)** met een drielaags hiërarchie:

Nr	Categorie
01	Persoon
02	Ouder 1
03	Ouder 2
04	Nationaliteit
05	Huwelijk/partnerschap
06	Overlijden
07	Inschrijving
08	Verblijfplaats
09	Kind
10	Verblijfstitel
11	Gezagsverhouding
12	Reisdocument
13	Kiesrecht

Categorie (CC)
↓ bevat
Groep (GG)
↓ bevat
Element / Rubriek (EE)

Rubrieknummer: `CC.GG.EE`
Bijv. `01.02.40` = Geslachtsnaam

Historische categorieën: 51–66
Informatieproducten: PA, PH, PX

HET AUTORISATIEMODEL

Drie voorwaarden voor toegang tot BRP-gegevens:

1. Wettelijke basis
Publiekrechtelijke taak of aangewezen belang

2. Doelbinding
Gegevens alleen voor het opgegeven doel

3. Proportionaliteit
Niet meer dan noodzakelijk

Resulteert in een **autorisatiebesluit**:

- Genomen door de Minister van BZK
- Gepubliceerd in de Staatscourant
- Per afnemer: welke rubrieken, welke koppelvlakken, onder welke voorwaarden
- Technisch vertaald in **Tabel 35**

TABEL 35: DE AUTORISATIETABEL

De technische kern van het autorisatiebeleid — per afnemer een rij met:

Kolom	Inhoud	Voorbeeld
Afnemersindicatie	6-cijferige code	001401
Afnemernaam	Organisatie	Gemeente Groningen
Versienummer	Oplopend	5
Datum ingang / einde	Geldigheid	2024-01-01 / —
Ad hoc rubrieken	Velden voor bevraging	01.01.10, 01.02.10, ...
Spontaan rubrieken	Velden voor push	08.09.10, 08.11.10, ...
Selectie rubrieken	Velden voor bulk	01.01.10, 01.02.40, ...
Voorwaarderegel	Condities op personen	`08.09.10 GA1 0363`
Sleutelrubrieken	Triggers voor spontaan	08.09.10, 08.11.10

DRIE KOPPELVLAKKEN

AD HOC (BEVRAGING)

Vraag-antwoord, via de HaalCentraal BRP API

- Raadpleeg met BSN
- Zoek met postcode + huisnummer
- Zoek met naam + geboortedatum
- etc.

SPONTAAN (PUSH)

Automatische melding bij wijziging

- Afnemersindicatie op PL plaatsen
- Sleutelrubrieken triggeren bericht
- Voorwaarderegel bepaalt selectie

SELECTIE (BULK)

Eenmalig of periodiek bestand

- Alle personen die aan criteria voldoen
- Via CSV of netwerklevering

VOORWAARDEREGELS

Booleaanse expressies die bepalen welke personen in scope zijn:

```
08.09.10 GA1 0363
```

→ Gemeente van inschrijving = Amsterdam (0363)

```
01.03.10 KDOG1 19.89.20 – 000300 ENVWD 01.03.10 GD1 19.89.20 – 001800
```

→ Leeftijd tussen 3 en 18 jaar

Operator	Betekenis	ODRL mapping
GA1 / GAA	Gelijk aan (1/alle)	brp:ga1 / brp:gaa
OGA1	Ongelijk aan	brp:oga1
GD1 / KD1	Groter / kleiner dan	brp:gd1 / brp:kd1
OFVGL / ENVGL	Een van / alle van lijst	odrl:isAnyOf / odrl:isAllOf
KV / KNV	Komt voor / niet voor	brp:kv / brp:knv
ENVWD / OFVWD	EN / OF	odrl:and / odrl:or

2. VAN TABEL 35 NAAR ODRL

github.com/mtrimpe/brp-odrl

WAT HEBBEN WE GEBOUWD?

~1.400 actuele autorisatiebesluiten als ODRL policies

4.575 voorwaarderegels volledig geparsed (100% coverage)

~2.600 afnemers met metadata

OWL ontologie + **ODRL profiel** met custom operatoren

Volledig reproduceerbaar:

```
make generate
```

Bouwt alles uit de RvIG brontabellen (Tabel 32–56)

Plus:

- Temporeel versiebeheer (ODRL Temporal Profile)
- DCAT-AP-NL 3.0 catalogus
- RvIG referentietabellen als Linked Data
- Compact versie (categorie/groepniveau)

BRP → ODRL MAPPING

BRP concept	ODRL concept	Voorbeeld
Autorisatiebesluit	`odrl:Set`	`brpaut:001401-v5`
Afnemer	`odrl:assignee`	`brpafn:001401`
Koppelvlak	`odrl:Action`	`brp:adHocVerstrekking`
API query type	`odrl:Action` (sub)	`brp:raadpleegMetBsn`
Rubrieken	`odrl:target`	`brprub:voornamenPersoon`
Voorwaarderegel	`odrl:Constraint`	`odrl:and`, `odrl:or`, `brp:ga1`
Temporele geldigheid	`tpl:TemporalSet`	`brpaut:autorisatiebesluiten`

ACTIE-HIËRARCHIE (`odrl:includedIn`)

De drie BRP-koppelvlakken worden ODRL acties, met sub-acties via `odrl:includedIn`:

```
odrl:use
 └─ brp:spontaneVerstrekking
      (push bij wijziging)

odrl:distribute
 ├─ brp:adHocVerstrekking
 │    (vraag-antwoord)
 │    ├─ brp:raadpleegMetBsn
 │    ├─ brp:zoekMetPostcodeEnHuisnr
 │    ├─ brp:zoekMetNaamEnGeboortedatum
 │    ├─ brp:zoekMetNaamEnGemeente
 │    ├─ brp:zoekMetStraat...
 │    ├─ brp:zoekMetNummeraanduiding
 │    ├─ brp:zoekMetAdresseerbaarObject
 │    └─ brp:adresVraag
 │
 └─ brp:selectieVerstrekking
      (bulk)
      ├─ brp:selectieGegevensVerstrekking
      ├─ brp:selectiePlaatsingIndicatie
      ├─ brp:selectieLogischVerwijderen
      ├─ brp:selectieVoorwaardelijkVerw.
      └─ brp:selectieOnvoorwaardelijkVerw.
```

Oud ↔ nieuw
Tabel 35 autoriseert op *koppelvlak*-niveau: spontaan, selectie, ad hoc. De HaalCentraal BRP API introduceert specifieke *PersonenQuery types* (`RaadpleegMetBsn`, `ZoekMetPostcode...`) die niet in Tabel 35 voorkomen.

`odrl:includedIn` overbrugt dit: de nieuwe API query types vallen onder het bestaande ad hoc koppelvlak.

Standaard ODRL — `odrl:includedIn` is geen extensie. De evaluator weet zo dat een `raadpleegMetBsn` request valt onder de `adHocVerstrekking` autorisatie.

VOORBEELD: MINISTERIE VAN DEFENSIE

TABEL 35 (BRON)

```
Afnemer: 890001
Min. van Defensie / Dienstplicht

Voorwaarderegel ad hoc:
(01.03.10 KDOG1 19.89.30 – 0017)
  ENVWD
(04.05.10 GA1 0001)
```

→ Geboortedatum ≤ vandaag − 17 jaar

EN

→ Nationaliteit = 0001 (Nederlands)

ODRL (GEGENEREERD)

```
brpaut:890001-v11 a odrl:Set ;
  odrl:permission [
    odrl:action brp:adHocVerstrekking ;
    odrl:assignee brpafn:890001 ;
    odrl:target brprub:voornamenPersoon,
      brprub:geslachtsnaamPersoon,
      brprub:geboortedatumPersoon,
      ... ;
    odrl:constraint [
      a odrl:LogicalConstraint ;
      odrl:and (
        [ odrl:leftOperand
            brprub:geboortedatumPersoon ;
          odrl:operator brp:kdog1 ;
          odrl:rightOperand "P17Y"^^xsd:duration ;
          odrl:rightOperandReference
            brprub:vandaagdatumSysteem ]
        [ odrl:leftOperand
            brprub:nationaliteitNationaliteit ;
          odrl:operator brp:ga1 ;
          odrl:rightOperand "0001" ]
      ) ] ] .
```

NAMESPACE-ONTWERP

Prefix	Namespace	Bevat
`brp:`	data.rijksoverheid.nl/brp/def#	Ontologie, acties, operatoren
`brpcat:`	.../brp/categorie/	Categorieën (01–17, 51–66)
`brpgrp:`	.../brp/groep/	Groepen
`brprub:`	.../brp/rubriek/	Rubrieken (CC.GG.EE)
`brpafn:`	.../brp/afnemer/	Afnemers
`brpaut:`	.../brp/autorisatie/	Autorisatiebesluiten

Open vraag: Deze namespace is door ons gekozen, niet door RvIG.
Hoe gaan we om met RDF-definities *voor* een registratiehouder?

3. AMSTERDAM: GEMEENTELIJK BELEID

Twee lagen autorisatie bovenop elkaar

AMSTERDAM: SCOPE-GEBASEERD MODEL

Amsterdam verdeelt de nationale BRP-autorisatie in fijnmazige JWT scopes:

WAT (GEGEVENSSETS)

29 gegevenssets als `odrl:AssetCollection` met hiërarchie:

```
# Gegevensset = JWT scope
amsveld:basisPersoonsgegevens
    a odrl:AssetCollection ;
    skos:notation "benk-brp-gegevensset-1" .

# Deelcollecties
amsveld:adres
    odrl:partOf amsveld:basisPersoonsgegevens .
amsveld:naamEnAdressering
    odrl:partOf amsveld:basisPersoonsgegevens .
amsveld:geboorteDatumEnGeslacht
    odrl:partOf amsveld:basisPersoonsgegevens .
amsveld:overlijden
    odrl:partOf amsveld:basisPersoonsgegevens .

# Concrete rubrieken in een deelcollectie
brprub:burgerservicenummerPersoon
    odrl:partOf amsveld:adres .
brpgrp:verblijfplaats-adres
    odrl:partOf amsveld:adres .
```

WIE (POPULATIE)

3 standaardbeperkingen, ophefbaar via behavioral scopes

```
amspol:alleenAmsterdam
    a odrl:Constraint ;
    skos:notation "benk-brp-landelijk" ;
    odrl:leftOperand brprub:gemeente... ;
    odrl:operator odrl:eq ;
    odrl:rightOperand gem:gm0363 .

amspol:nietOverleden ...
amspol:nietGeheim ...
```

TWEE-LAAGS EVALUATIE

Een API request wordt tegen **twee policies** geëvalueerd. Beide moeten slagen.

Laag 1 — Nationaal (RvIG)
Policy: autorisatiebesluit uit Tabel 35

- Mag deze afnemer deze rubrieken opvragen?
- Via dit koppelvlak?
- Voldoet de persoon aan de voorwaarderegel?

Laag 2 — Gemeentelijk (Amsterdam)
Policy: gegevensset + populatieconstraints

- Heeft de app de juiste gegevensset-scope?
- Alleen Amsterdammers? (tenzij scope landelijk)
- Geen overledenen/geheimhouding? (tenzij scope)

Het strengste beleid wint — de gemeente kan de nationale autorisatie alleen **verder inperken**, niet verruimen.

Een gedeelde Linked Data definitie (zoals `brprub:`) helpt hier dubbel: dezelfde URI's worden gebruikt in de **beleidsdefinitie** (odrl:target) én in het **individuele verzoek** (odrl:Request). Dat maakt evaluatie mogelijk zonder mapping-stap.

4. END-TO-END EVALUATIE

API Request → PDP-adapter → ODRL Request + Policy → FORCE → Besluit

VAN API CALL NAAR ODRL REQUEST

HTTP REQUEST

```
POST /bevragingen/v1/personen
Authorization: Bearer <JWT 001401>

{
  "type": "RaadpleegMetBsn",
  "burgerservicenummer":
      ["999993653"],
  "fields": [
    "naam.voornamen",
    "geboorte.datum"
  ]
}
```

AUTHZEN EVALUATION REQUEST

```
{
  "subject": {
    "type": "identity",
    "id": "<JWT.sub>"
  },
  "action": {
    "name": "POST",
    "properties": {
      "body": {
        "type": "RaadpleegMetBsn",
        "burgerservicenummer":
            ["999993653"],
        "fields": [
          "naam.voornamen",
          "geboorte.datum"]
      }
    }
  },
  "resource": {
    "type": "route",
    "id": "/bevragingen/v1/personen"
  }
}
```

VAN AUTHZEN NAAR ODRL REQUEST

MAPPING

```
subject.id       → odrl:assignee
action.body.type → odrl:action
action.body.fields → odrl:target
PL-gegevens      → SotW (constraints)
```

De PDP-adapter haalt de persoonslijst op (gemeente, geboortedatum, etc.) en zet die als `sotw:context` constraints op het ODRL Request.

ODRL REQUEST

```
<request> a odrl:Request ;
  odrl:permission [
    odrl:assignee brpafn:001401 ;
    odrl:action
      brp:raadpleegMetBsn ;
    odrl:target
      brprub:voornamenPersoon,
      brprub:geboortedatumPersoon
  ] .
```

FORCE EVALUATIE: TWEE SCENARIO'S

Scenario 1: Amsterdam
BSN 999993653
Gemeente: gm0363 (Amsterdam)

Constraint: `gemeente = 0363`
→ **SATISFIED**

Resultaat: **TOEGANG VERLEEND**

Scenario 2: Den Haag
BSN 999993654
Gemeente: gm0518 (Den Haag)

Constraint: `gemeente = 0363`
→ **NOT-SATISFIED**

Resultaat: **TOEGANG GEWEIGERD**

Beperking FORCE evaluator: ODRL constraints gebruiken een `leftOperand` om aan te geven *welk* gegeven je vergelijkt (bijv. `brprub:gemeenteVanInschrijving`). FORCE accepteert alleen ODRL-standaard operands (zoals `odrl:dateTime`) en weigert domein-specifieke. Wij moesten die whitelist verwijderen om BRP rubrieken te laten werken.

SCOPE-COMPOSITIE: HET KERNPROBLEEM

ODRL kent geen "verwijder een constraint". Maar het Amsterdam-model vereist precies dat.

Optie	Aanpak	Trade-off
A	Strengste constraints, compositie als conventie	Simpel, maar niet evalueerbaar
B	Alle 2^n combinaties als permissions	Evalueerbaar, maar 29×8 = 232 permissions
C	Gescheiden policies + profiel-extensie	Uitbreidbaar, maar niet standaard ODRL
D	Action refinements	Semantisch zuiver, maar FORCE ondersteunt het niet
E	Offer/Agreement patroon	Standaard ODRL, maar past conceptueel niet

Vraag aan de groep: Moeten we "evalueerbaar door een standaard ODRL evaluator" als doel stellen? Of blijft ODRL primair een *beschrijvingstaal* en laten we evaluatie per implementatie over?

5. RELATIE TOT HET REGISTER TOEGANGSBELEID

WAT LEERT BRP ONS OVER HET REGISTER?

ANALYSEDIMENSIES (17 MAART)

Dimensie	BRP score
Mensleesbaar	Ja (Staatscourant)
Machineleesbaar	Ja (Tabel 35 CSV, nu ODRL)
Hiërarchie	Registratie → afnemer → koppelvlak
Granulariteit	Rubriekniveau + voorwaarderegels
Historisering	Effectieve + materiële historie
Semantiek	Gedefinieerd in LO BRP

LESSEN VOOR HET REGISTER

- **Versiebeheer** is essentieel — BRP heeft temporele versies per besluit
- **Semantiek** moet gedeeld zijn — vocabulaire als voorwaarde voor interoperabiliteit
- **Granulariteit** varieert — van rubriek tot informatieproduct
- **Meerdere lagen** — nationaal + gemeentelijk

WAT ODRL GOED DOET

Extensible via profielen
Domein-specifieke operatoren (`brp:ga1`), acties (`brp:adHocVerstrekking`) en vocabulaires toevoegen zonder de standaard te breken.

Temporal Profile
Versiebeheer van policies over tijd. Elk autorisatiebesluit als `tpl:TemporalSet` met versies per periode.

Conceptueel beleid
Beschrijft het *wat* en *waarom*. Complementair aan technische regelsets (Rego, Cedar) voor het *hoe*.

AssetCollections
Hiërarchische groepering via `odrl:partOf`. Past bij dataminimalisatie: van rubrieken naar informatieproducten en gegevenssets.

Constraints
Voorwaarderegels als logische expressies (`odrl:and`, `odrl:or`, custom operatoren). Alle 4.575 BRP voorwaarderegels zijn 1-op-1 te mappen.

6. OPEN VRAGEN

MEER DAN AUTORISATIEBESLUITEN

Het LO BRP definieert veel toegangsbeleid dat **niet** in Tabel 35 staat:

Verplichting	Bron (LO BRP)	Aard
Aansluittoets	§A.2.3, §A.6.2.4	Technische poort: toets op proefomgeving vóór productietoegang
Terugmeldplicht (TMV)	§6.3, art. 2.34 Wet BRP	Wettelijke plicht: vermoede fouten melden via Digimelding
PKI/TLS-eisen	§6.2.7.4, §A.2.5	Technisch: goedgekeurde PKIO-certificaten, code-inspectie door RvIG
Berichten ophalen	§A.7.1	Operationeel: minimaal 1× per werkdag berichten ophalen
Protocollering	§3.3.12, §A.5.2	Wettelijk: alle verstrekkingen 20 jaar loggen (inzagerecht)
Beveiligingseisen	§A.7.2	Technisch: AVG- en BIO-compliance voor afnemerssystemen

OBLIGATIONS IN ODRL: STRUCTUREEL PROBLEEM

ODRL kent `odrl:duty` — maar alleen op Policy- of Permission-niveau.

Het probleem

Terugmeldplicht, aansluittoets, protocollering etc. gelden voor **alle** afnemers. In ODRL zou je ze op *elk* autorisatiebesluit als `odrl:duty` moeten toevoegen. Er is geen concept van "stelsel-brede obligations".

Hetzelfde compositie-probleem

Net als bij scope-compositie: ODRL kent geen overerving of cross-cutting beleid.

Opties:

- Duty op elke policy (herhaling)
- Basis-policy + `odrl:inheritFrom`
- Profiel-extensie (zoals optie C)
- Conventie buiten ODRL

TEMPORALITEIT: WELKE TIJDDIMENSIES?

BRP kent drie vormen van historie. Het ODRL Temporal Profile ondersteunt er één.

Dimensie	Wat	BRP
Effectief	Wanneer geldt het besluit?	Kolom 99.98 / 99.99
Materiëel	Wanneer inhoudelijk vastgesteld?	Publicatiedatum
Formeel	Wanneer in systeem verwerkt?	Niet expliciet

ODRL Temporal Profile ondersteunt één tijddimensie (`tpl:TemporalSet` met periodes).

Voor audit-trails en reconstructie kan materiële/formele historie wenselijk zijn.

Vraag: Is effectieve tijd voldoende voor het Register?

BELEIDSVERWIJZING IN DE KETEN

Moet er verwezen worden naar het beleid in de technische request/response-keten?

Bij toekenning

- API-antwoord benoemt welk autorisatiebesluit-versie gold

  (bijv. `brpaut:001401-v5` in response header)
- ADL record verwijst naar het beleid
- Traceerbaarheid: achteraf reconstrueerbaar welk beleid gold

Bij afwijzing

- Reden voor afkeuring benoemen

  (bijv. "mTLS-certificaat ontbreekt", "voorwaarderegel niet voldaan")
- Welke verplichting niet is nagekomen?
- Verwijzing naar het beleid dat de afwijzing veroorzaakt

ODRL UITBREIDEN?

We lopen op twee plekken tegen de grenzen van ODRL aan:

Constraints verwijderen
ODRL kent geen mechanisme om een constraint conditioneel weg te halen.

- Amsterdam: behavioral scope heft constraint op
- Meerdere lagen beleid combineren

Stelsel-brede constraints/obligations
Terugmeldplicht, aansluittoets, PKI-eisen etc. gelden voor *alle* afnemers. ODRL kent geen cross-cutting beleid — je zou ze op elke policy moeten herhalen.

Opties

- **Profiel-extensie** voorstellen bij W3C/ODRL community

  (compositiemodel, stelsel-policies)
- **Basis-policy** met `odrl:inheritFrom`

  (maar: bedoeld voor Offer/Agreement, niet overerving)
- **Conventie buiten ODRL**

  (compositie in de PAP/PDP, niet in het beleid zelf)
- **Accepteren** dat ODRL dit niet doet

  (ODRL = beschrijvingstaal, compositie = implementatie)

FORCE EVALUATOR: DOEL OF MIDDEL?

We hebben aangetoond dat FORCE een echt BRP autorisatiebesluit kan evalueren — met patches.

Wat werkt

- Nationaal beleid (Tabel 35) is evalueerbaar
- SotW-context patroon voor PL-gegevens
- End-to-end pipeline draait

Beperkingen FORCE

- **Left-operand whitelist:** accepteert alleen ODRL-standaard operands, weigert domein-specifieke (zoals `brprub:`)
- **Action refinements:** nog niet geïmplementeerd
- Scope-compositie niet evalueerbaar

Vraag: Willen we "evalueerbaar door een standaard ODRL evaluator" nastreven? Of blijft ODRL primair een beschrijvingstaal?

VOCABULAIREBEHEER: WIE GAAT EERST?

> De BRP ODRL-representatie gebruikt `brp:` en `brprub:` namespaces onder `data.rijksoverheid.nl` — maar deze zijn door ons gedefinieerd, niet door RvIG.

Het kip-ei probleem:

Afnemer wil policies in ODRL modelleren → heeft vocabulaire nodig → registratiehouder publiceert (nog) geen Linked Data

Opties:

- Community-vocabulaire (zoals nu)
- Registratiehouder publiceert URI's
- Centraal stelsel-vocabulaire

Vraag: Hoe organiseren we vocabulairebeheer in het stelsel? Relatie met de conclusie van 4 maart: "het valt of staat bij een gedeelde vocabulaire".

DISCUSSIE

OVERZICHT PRAATPUNTEN

Wat ODRL goed doet

- Extensible via profielen
- Temporal Profile voor versiebeheer
- AssetCollections voor dataminimalisatie
- Constraints voor voorwaarderegels
- Gedeelde Linked Data URI's in beleid én verzoek

Open vragen

- Technisch beleid in ODRL — wens of niet?
- Missende Linked Data definities — wie gaat eerst?
- Is effectieve historisering voldoende?
- Gedeelde constraints/obligations — hoe?
- ODRL uitbreiden (constraints verwijderen)?
- FORCE evaluatie — doel of middel?
- Beleidsverwijzing in de request-keten?