

Inleiding Logboek dataverwerkingen

Logius Praktijkrichtlijn

Werkversie 04 november 2025

**Deze versie:**

<https://logius-standaarden.github.io/logboek-dataverwerkingen-inleiding/>

Laatst gepubliceerde versie:

<https://logius-standaarden.github.io/logboek-dataverwerkingen-inleiding/>

Laatste werkversie:

<https://logius-standaarden.github.io/logboek-dataverwerkingen-inleiding/>

Redacteurs:

Vedran Bilanovic ([Ministerie van Binnenlandse Zaken en Koninkrijksrelaties](#))

Eelco Hotting ([Ministerie van Binnenlandse Zaken en Koninkrijksrelaties](#))

Auteurs:

Nil Barua ([Logius](#))

Martin van der Plas ([Logius](#))

Tim van der Lippe ([Logius](#))

Jeroen Mulder ([Ministerie van Binnenlandse Zaken en Koninkrijksrelaties](#))

Doe mee:

[GitHub Logius-standaarden/logboek-dataverwerkingen-inleiding](#)

[Dien een melding in](#)

[Revisiehistorie](#)

[Pull requests](#)

Dit document is ook beschikbaar in dit niet-normatieve formaat: [PDF](#)



Dit document valt onder de volgende licentie:

[Creative Commons Attribution 4.0 International Public License](#)

Status van dit document

Dit is een werkversie die op elk moment kan worden gewijzigd, verwijderd of vervangen door andere documenten. Het is geen door het TO goedgekeurde consultatieversie.

Inhoudsopgave

Status van dit document

Conformiteit

1. Feedback en Issues

Samenvatting

2. Inleiding

2.1 Aanleiding en context van Logboek dataverwerkingen

2.2 Scope van de Standaard Logboek dataverwerkingen

2.3 Totstandkoming van de standaard

3. Architectuur

3.1 Context

3.2 De relatie tussen logboekelementen, waarom eigenlijk?

4. Besluitenlijst

5. Voorbeelden

6. FAQ

A. Referenties

A.1 Normatieve referenties

§ Conformiteit

Naast onderdelen die als niet normatief gemarkeerd zijn, zijn ook alle diagrammen, voorbeelden, en noten in dit document niet normatief. Verder is alles in dit document normatief.

De trefwoorden *AANBEVOLEN* en *MOET* in dit document moeten worden geïnterpreteerd als in [BCP 14](#) [[RFC2119](#)] [[RFC8174](#)] als, en alleen als deze in hoofdletters zijn weergegeven, zoals hier getoond.

§ 1. Feedback en Issues

Dit onderdeel is niet normatief.

We moedigen gebruikers aan om meldingen of suggesties aan te maken via [GitHub](#). Mocht dit niet mogelijk zijn, dan kunt u ook een e-mail sturen naar api@logius.nl.

Samenvatting

404: Not Found

§ 2. Inleiding

Het idee is dat het Logboek Dataverwerkingen een basis biedt om te zorgen dat de overheid precies de data logt die zij nodig heeft om verantwoording af te leggen over haar taken. Niet meer, maar ook niet minder. En om te zorgen dat organisaties data zodanig loggen dat zij zich niet alleen over een eigen handelen kunnen verantwoorden, maar ook over hun gezamenlijk handelen als “de overheid”.

§ 2.1 Aanleiding en context van Logboek dataverwerkingen

Informatiehuishouding van de overheid moet op orde worden gebracht. De overheid werkt ten dienste van burgers en bedrijven. De overheid verwerkt daarvoor informatie van deze burgers en bedrijven. Het is belangrijk dat de informatiehuishouding van de overheid op orde is, zodat de overheid transparant en aanspreekbaar is, en zich daarover goed kan verantwoorden.

Dat werkt als een soort vliegwiel, omdat daardoor ook de kwaliteit van de informatie beter wordt. De overheid kan daarmee betere dienstverlening bieden en ook zorgen dat de burger minder met fouten wordt geconfronteerd, of dat overheden fouten beter en sneller kunnen herstellen als deze zich onverhoopt voordoen.

Eenduidige en integrale verantwoording over dataverwerkingen door de overheid. Belangrijk is dat overheidsorganisaties op een eenduidige manier met informatie omgaan en op een eenduidige manier informatie met elkaar uitwisselen.

Voorafgaand aan informatie-uitwisseling is het belangrijk dat transparant is waarom dat gebeurt en, achteraf moet de overheid verantwoording kunnen afleggen over de data en de wijze waarop de data is verwerkt. Zo kunnen eventuele fouten of onregelmatigheden worden hersteld en kunnen burgers hun rechten op grond van de AVG geldend maken (oa. inzage en correctie). Het gaat daarbij niet alleen om overheidsorganisaties afzonderlijk, maar het gaat er ook – juist - om dat “dé overheid” zich als geheel ten opzichte van de burger kan verantwoorden.

Een belangrijk instrument om verbetering van de informatiehuishouding te bereiken is standaardisatie. Op diverse aspecten is daarom standaardisatie nodig en worden deze ontwikkeld. Een van deze aspecten is de wijze waarop overheden zich verantwoorden. Standaardisatie daarvan vormt daarmee een puzzelstukje in het bredere geheel. Hiermee kunnen overheden hun

dataverwerkingen op dezelfde wijze verantwoorden en deze verantwoording onderling relateren, zodat de keten van dataverwerkingen tussen organisaties compleet inzichtelijk kan worden gemaakt.

§ 2.1.1 Standaard Logboek dataverwerkingen

Om eenduidige verantwoording over dataverwerkingen te regelen en te zorgen dat deze verantwoording over overheidsorganisaties heen relateerbaar is, is de standaard Logboek Dataverwerkingen in ontwikkeling. De standaard heeft tot doel geautomatiseerd eenduidige verslaglegging van dataverwerkingen binnen organisaties te bevorderen, en dataverwerkingen tussen organisaties aan elkaar te relateren.

§ 2.2 Scope van de Standaard Logboek dataverwerkingen

Dit beschrijft een overzicht van de scope van de standaard, inclusief de zaken die wel en niet binnen de scope van de standaard vallen.

§ 2.2.1 In scope van deze standaard

Logging over dataverwerkingen in overheidssystemen. Het uitgangspunt van deze standaard is de verantwoordingsplicht van de overheid over de uitvoering van haar taken en de wetten en kaders die daarbij horen.

§ 2.2.2 Buiten scope van deze standaard

De volgende zaken worden **niet** behandeld in deze standaard:

- **Toegangsbeheer:** Logging rondom toegang tot systemen en data, waarbij zowel succesvolle als mislukte pogingen om toegang te krijgen worden vastgelegd. Deze logs zijn bedoeld voor het controleren van wie toegang heeft tot gevoelige informatie en voor het detecteren van ongeautoriseerde toegang.
- **Toegangsverlening Logboek:** De standaard specificeert geen functionaliteit rondom het aanmaken en beheren van toegangs- en onderhoudsprofielen ten behoeve van het logboek. Voor meer informatie zie [Federatieve Toegangsverlening](#).

- **Gebruikersactiviteiten:** Logging van namen van gebruikers die data gebruiken of verwerken.
- **Beveiligingsincidenten:** Specifieke logs voor incidenten die de beveiliging kunnen beïnvloeden, zoals malware-detectie, aanvallen of misbruik. Dit soort logging is van groot belang voor het identificeren van bedreigingen en het kunnen reageren op incidenten.
- **Technische en Systeemlogs:** Logging van systeemfouten, configuratiewijzigingen en technische problemen. Deze logs helpen bij het waarborgen van de stabiliteit en betrouwbaarheid van IT-systemen en ondersteunen het oplossen van technische problemen.
- **Logging ten behoeve van motivatie totstandkoming besluitvorming:** anders dan uitgevoerde dataverwerkingen (niet: beslisregels, algoritmes, et cetera).
- **Correcties op- en verwijdering van verwerkte data:** dit wordt gezien als verwerking en volgt de gewone route van datalogging. Voor meer informatie zie [besluit 4.5](#).
- **Beperkingen op informatieplichten (bijvoorbeeld indien er een strafrechtelijk onderzoek plaatsvindt):** het is aan de organisatie zelf om procedures te implementeren om beperking van inzage uit te voeren. Voor meer informatie zie [besluit 4.6](#).

§ 2.3 Totstandkoming van de standaard

§ 2.3.1 aanleiding en achtergrond

Het Logboek Dataverwerkingen is een doorontwikkeling van de conceptstandaard GEMMA Verwerkingenlogging, die door VNG Realisatie is gemaakt met als doel de naleving van AVG-verplichtingen rondom de verwerking van persoonsdata te verbeteren.

In 2023 heeft het Ministerie van Binnenlandse Zaken, in samenwerking met verschillende overheidspartijen, een project gestart om de GEMMA Verwerkingenlogging-standaard verder te ontwikkelen. Het uitgangspunt was het vergroten van de transparantie van de overheid en het verbeteren van de informatiepositie van de burger. Vanaf 2024 werd breder gekeken dan alleen de AVG; wettelijke kaders, zoals verantwoordingsverplichtingen, werden als uitgangspunt genomen voor het vormgeven van de standaard. Om aan deze eisen te voldoen, is de standaard aangepast en hernoemd tot Logboek Dataverwerkingen.

§ 2.3.2 Interdisciplinaire aanpak

Voor de ontwikkeling van de standaard Logboek Dataverwerkingen was het essentieel dat de verschillende aspecten (juridische beleidskaders, techniek, inhoud en beheer) goed op elkaar

werden afgestemd. Daartoe werkte het project met een interdisciplinair team: juristen, beleidsmakers en adviseurs van BZK werkten nauw samen met technische experts van Digilab en medewerkers van Logius, de beoogde beheerder. Deze interdisciplinaire aanpak zorgde ervoor dat de standaard aansluit op juridische randvoorwaarden, eenvoudig te beheren en te implementeren is, én effectief functioneert in de praktijk. Dit laatste aspect werd getest in Digilab, waar de standaard in verschillende simulatieomgevingen (Fieldlabs) werd ingebouwd en beproefd op praktische toepasbaarheid.

§ 2.3.3 Beheer en doorontwikkeling

Om de overgang tussen ontwikkeling en beheer soepel te laten verlopen, was Logius vanaf een vroeg stadium betrokken bij het project. De inzet van Logius is in de loop van de tijd uitgebreid, zodat in 2025 het beheer van de standaard volledig kan worden overgedragen. Dit beheer wordt ingericht volgens de BOMOS-methodologie (Beheer- en OntwikkelModel voor Open Standaarden). Het opzetten van een goede governance-structuur is een integraal onderdeel van het beheer. Hierbij zullen, naast de gebruikers van de standaard, belangrijke rollen zijn weggelegd voor MIDO (Meerjarenprogramma Infrastructuur Digitale Overheid) en het Forum Standaardisatie. Deze gremia zullen naar verwachting respectievelijk de standaard vaststellen en deze opnemen op de Pas-Of-Leg-Uit-lijst. Het Ministerie van Binnenlandse Zaken blijft opdrachtgever voor het beheer van de standaard.

§ 3. Architectuur

§ 3.1 Context

De standaard Logboek dataverwerkingen levert geen kant-en-klare softwareoplossing. Wel biedt de standaard richtlijnen waar het Logboek dataverwerkingen van een applicatie aan moet voldoen. Dit document geeft aan hoe de standaard zich verhoudt tot de [Architectuur Digitale Overheid 2030](#) en de *Domeinarchitectuur Gegevensuitwisseling*

§ 3.1.1 Bedrijfsarchitectuur

- *Diensten en producten*
 - De standaard Logboek dataverwerkingen is als product voornamelijk gericht op het verantwoorden van dataverwerkingen door overheden in het kader van hun taken (**Regie**

op Gegevens en de Transparante Overheid). Dit betekent dat het gebruik van deze standaard door overheidsorganisaties de informatiepositie van burgers en bedrijven ten opzichte van de overheid sterk verbetert zodat zij meer grip op en inzicht in hun persoonsgebonden data hebben (**inzicht in data over jezelf**). Daarnaast draagt deze standaard aan bij **Verantwoord datagebruik en ruimer meervoudig gebruik data**. Implementatie van deze standaard draagt bij aan de verantwoording over, en het doelmatig gebruik van data.

- *Kanalen*
 - Het Logboek dataverwerkingen is een service behorend bij een applicatie die specifieke data verwerkt waarover uiteindelijk verantwoording moet kunnen worden afgelegd (bijvoorbeeld persoonsdata of data over geografische objecten). De standaard geeft geen richtlijnen ten aanzien de ontsluiting van deze logdata richting belanghebbenden, wel geeft het een richting ten aanzien van de inrichting en het gedrag van het Logboek dataverwerkingen.
- *Organisatie*
 - De Overheid bestaat uit vele (semi)autonome organisaties. Door gezamenlijke afspraken te maken ten aanzien van logging van verwerkte data, ondersteunt de standaard het doel om naar de burger toe als één organisatie te kunnen verantwoorden.
- *Processen*
 - Voor het verwerken van data zijn vaak ook data nodig van andere (overheids)organisaties. De implementatie van de standaard Logboek dataverwerkingen zorgt er voor dat loggings tracing-metadata bevat zodat altijd kan worden nagegaan wat de bron van specifieke data was. De standaard ondersteunt hiermee het uitgangspunt dat (overheids)organisaties zorgen voor onderlinge samenhang van data.
- *Bedrijfsfuncties*
 - Overheidsfuncties moeten eenduidig een helder belegd zijn, het moet helder zijn welke (overheids)organisaties verantwoordelijk zijn voor het leveren van product of dienst. Door het gebruiken van de standaard Logboek datalogverwerkingen door alle dataverwerkende (overheids)organisaties op een soortgelijke manier wordt het duidelijk welke data gebruikt zijn en door wie.

§ 3.1.2 Informatiearchitectuur (Information systems architecture)

- *Api's / Services*
 - Naast richtlijnen voor de inrichting en het gedrag van het Logboek dataverwerkingen, biedt deze standaard ook een aantal voorbeeld API's:
 - **Inzicht API:** deze service geeft de mogelijkheid een query uit te voeren op loggings van dataverwerkingen (nog niet beschikbaar).

- **Register van de Verwerkingsactiviteiten:** deze service geeft de mogelijkheid de data van een Register van Verwerkingsactiviteiten te bekijken (nog niet beschikbaar).

De API's zijn ontworpen en ontwikkeld volgens de standaard [Rest-API Design Rules](#).

- *Applicaties*

- De standaard biedt geen applicatie aan, wel biedt het richtlijnen ten aanzien van het gedrag en invulling van het Logboek dataverwerkingen. Hiermee geeft de standaard de vrijheid aan organisaties om zelf op basis van de specifieke implementatie van een dataverwerkende Applicatie een Logboek te ontwikkelen wat qua gedrag en (meta)data gelijkvormig is over alle (overheids)organisaties heen.

- *Berichtenverkeer / datauitwisseling*

- Het berichten verkeer met betrekking tot het Logboek dataverwerkingen heeft geen directe connectie met de burger. Wel is het van belang bij opvraag van data bij andere organisaties traceringsdata worden verstuurd en opgeslagen in het Logboek zodat altijd duidelijk wat de bron is van data die verwerkt zijn. Deze standaard biedt een traceringsmethodiek aan zodat de datauitwisseling tussen organisaties *vastgelegd en verantwoord* kan worden.

- *Data*

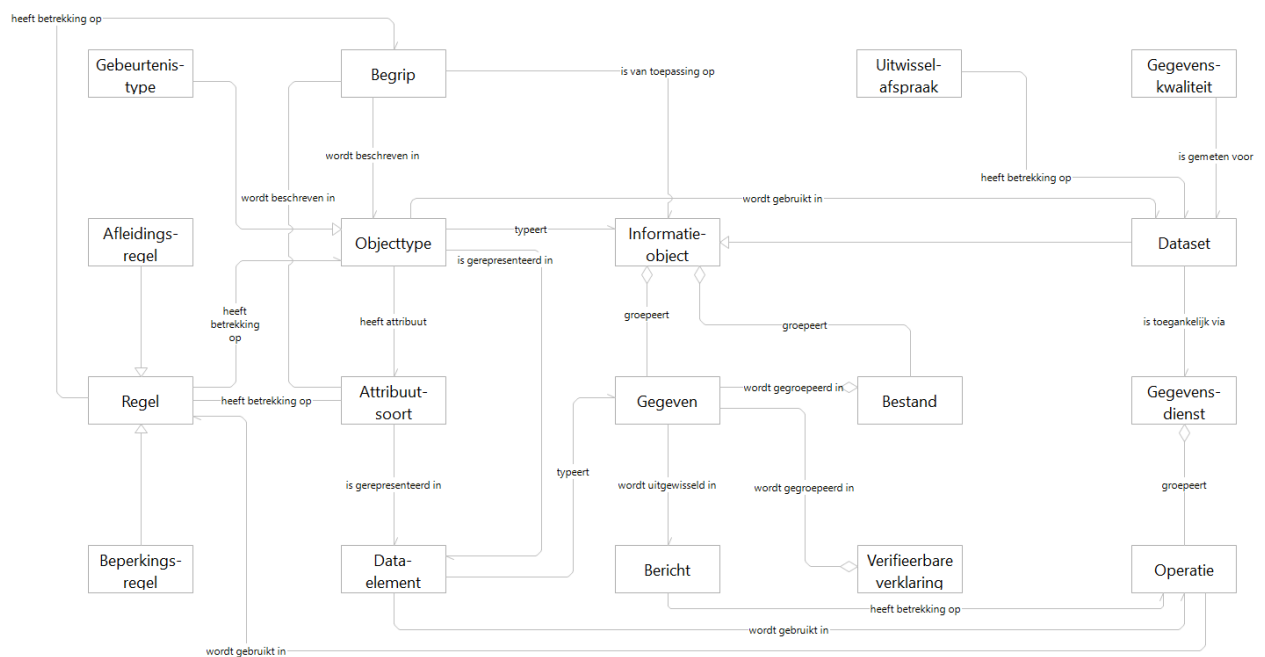
- De Nederlandse Basisregistraties vervullen een essentiële rol in het vastleggen en gecontroleerd beheren van data. Organisaties kunnen aan elkaar data ter beschikking te stellen vanuit 'kernregistraties'. De standaard Logboek dataverwerkingen biedt een richtlijn voor het loggen van verwerkte data van al deze basisregistraties.

Onderstaande stelselplaat geeft een globaal overzicht van de bronhouders, de aanbieders en afnemers van data.



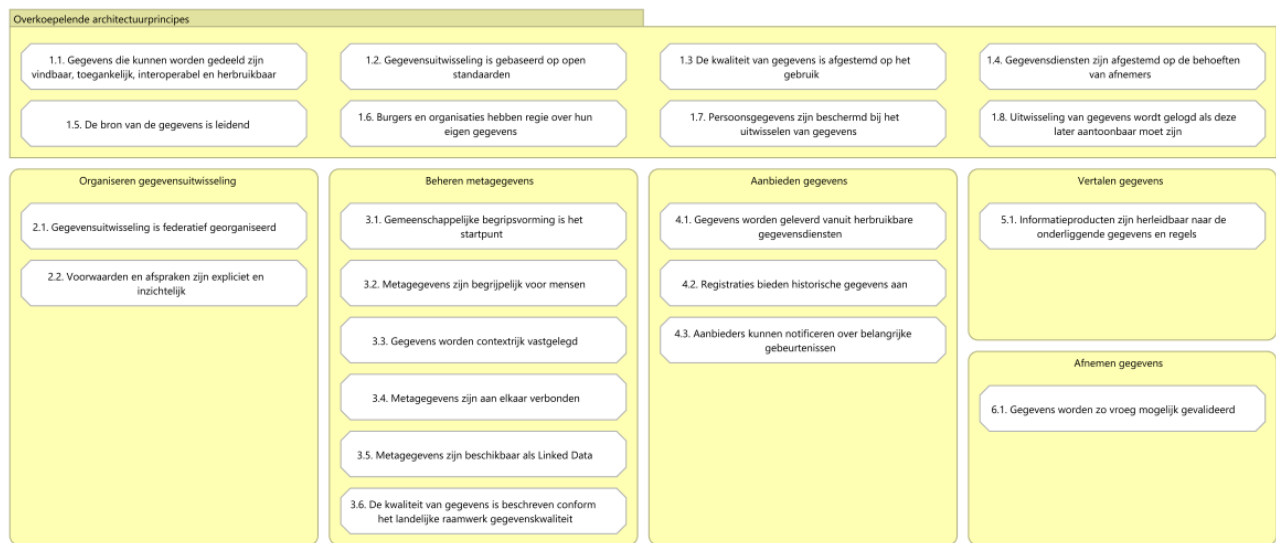
[bron: Architectuur Digitale Overheid 2030]

Een belangrijk kader voor de standaard Logboek dataverwerkingen is de uitwerking van het [GDI meerjarenvisie](#) op basis van de [Architectuur Digitale Overheid 2030](#) met als specifiek onderwerp Gegevensuitwisseling. De standaard Logboek dataverwerking kan gepositioneerd worden in de GDI Gegevensuitwisseling als standaard waarin een 'Uitwisselingsafpraak' geformaliseerd wordt. Waarbij de daadwerkelijke logging betrekking heeft op de 'Operatie' in de modelplaat [GDI-Gegevensuitwisseling](#).



[Bron: GDI – Gegevensuitwisseling]

Onderstaand figuur geeft een overzicht van de architectuurprincipes uit het [GDI meerjarenvisie](#) en hun relatie met de belangrijkste functie voor data uitwisseling.



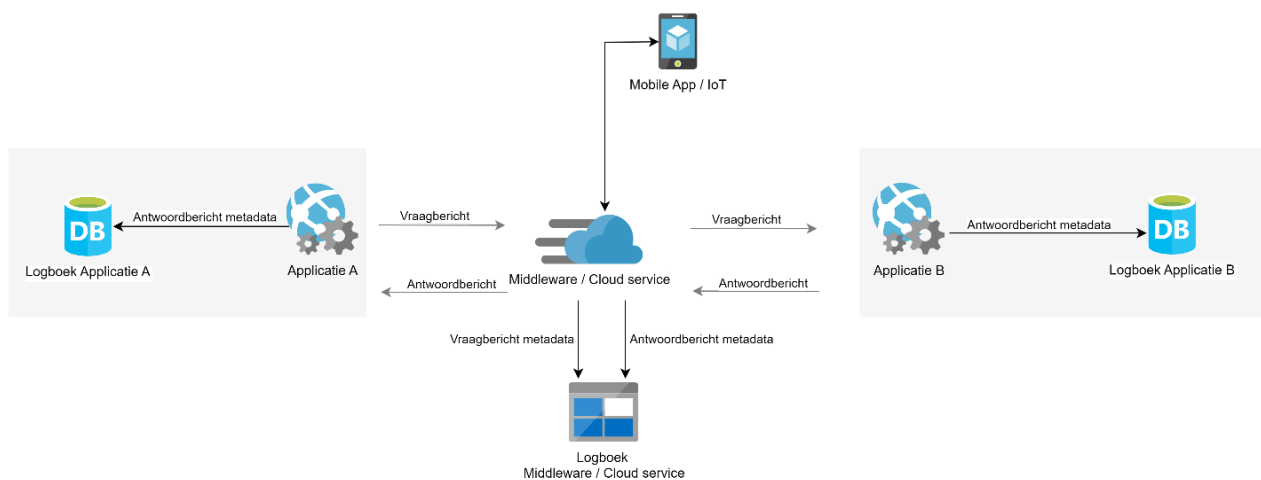
[Bron: MIDO/GDI Domeinarchitectuur Gegevensuitwisseling]

De relatie en invulling van de standaard Logboek dataverwerkingen staat uitgewerkt in de volgende paragraaf.

§ 3.1.3 Technische architectuur (Technical architecture)

§ Netwerken en slimme apparaten

De standaard Logboek dataverwerkingen kan ook worden toegepast in een middleware- of cloud-omgeving. Het netwerkcomponent logt binnenkomende en uitgaande berichten.



Ook voor mobiele Apps en IoT (Internet of Things) geldt dat het netwerkcomponent de databerichten logt.

§ 3.1.4 Software architectuur

Hoofdstuk [2.2 Componenten](#) geeft een globaal overzicht van de benodigde softwarecomponenten om de standaard te implementeren.

§ Platformen voor dagelijkse exploitatie en huisvesting

De standaard Logboek dataverwerkingen gaat er vanuit dat de het Logboekcomponent op een beveiligd platform in een beveiligd datacenter is geïnstalleerd.

§ Aspectgebieden

- *Informatiebeveiliging*
 - De standaard Logboek dataverwerkingen gaat er vanuit dat zowel het Logboekcomponent als de data in het Logboek beveiligd zijn volgens de BIO (=Baseline Informatiebeveiliging Overheid) – zie ook [Beleidskader](#).
- *Beheer en exploitatie*
 - Logius verzorgt het beheer en onderhoud van deze standaard volgens het [BOMOS](#)-model.

§ 3.1.5 Relaties GDI architectuurprincipes en de standaard

| Architectuurprincipe | Relatie met de standaard |
|--|--|
| 1.1. Data die kunnen worden gedeeld zijn vindbaar, toegankelijk, interoperabel en herbruikbaar | <ul style="list-style-type: none">- Logregels zijn voorzien van metadata ten aanzien van tracering zodat gerelateerde Logregels altijd gevonden kunnen worden.- Identificatoren worden aangemaakt zodat deze over de gehele wereld uniek zijn.- Het <code>processing_activity_id</code> is gerelateerd aan het Register van Verwerkingsactiviteiten zodat per Logregel altijd verwezen |

| Architectuurprincipe | Relatie met de standaard |
|---|--|
| | kan worden naar een activiteit van een (overheids)organisatie en daarmee de context direct duidelijk wordt. |
| 1.2. Datauitwisseling is gebaseerd op open standaarden | <ul style="list-style-type: none"> - API's gerelateerd aan deze standaard moeten worden ontworpen en gebouwd volgens de standaarden REST-API Design Rules, OpenAPI en DigiKoppeling. - Het metadatamodel van deze standaard is gebaseerd op OpenTelemetry. Dit is een internationale standaard voor het genereren, verzamelen en exporteren van telemetrie data (metriecken, logging en tracering). |
| 1.3 De kwaliteit van data is afgestemd op het gebruik | <ul style="list-style-type: none"> - Door de registratie van verwerkte data in een Logboek kan er op een later moment inzicht gegeven worden aan andere (overheids)organisaties en burgers. Eventuele foutieve data komen dan direct aan het licht en kunnen hersteld worden. |
| 1.4. Datadiensten zijn afgestemd op de behoeften van afnemers | <ul style="list-style-type: none"> - Data die gelogd worden bij een dataverwerking zijn afgestemd op het doel waarvoor er gelogd moet worden (bijvoorbeeld de data die gevraagd worden op basis van de AVG-wetgeving). Er wordt niet minder opgeslagen, meer zeker niet meer dan nodig is. - Het ontwerp van het datamodel van deze standaard is gebaseerd op OpenTelemetry, vertaling van data is dus niet nodig. |
| 1.5. De bron van de data is leidend | <ul style="list-style-type: none"> - Overheidsapplicaties moeten rekening houden met de onderhoud van data bij de bron. Dit betekent dat data niet zonder meer gekopieerd opgeslagen mogen worden. Bij sommige dataverwerkingen zijn data nodig van andere databronnen (in de eigen organisatie of bij een andere organisatie). De standaard Logboek Dataverwerkingen schrijft een traceringsmechanisme voor zodat kopiëren van specifieke data naar het Logboek niet nodig is, er kan altijd nagegaan worden waar data vandaan kwamen en welke data er gebruikt werden. - De standaard verwijst zo veel mogelijk naar bestaande databronnen elders in plaats van de data te dupliceren (zie besluit 4.2 en 4.4) |
| 1.6. Burgers en organisaties hebben regie over hun eigen data | <ul style="list-style-type: none"> - Burgers, (overheids)organisaties en parlement hebben recht om inzicht te krijgen in verwerkte data. Door toepassing van deze standaard kan er een rapportage gemaakt worden die voldoet aan die informatiebehoefte. - De standaard Logboek dataverwerkingen biedt geen handreiking ten aanzien van de manier waarop datainzicht plaats moet vinden richting belanghebbende, wel op de inhoud van het datainzicht. |

| Architectuurprincipe | Relatie met de standaard |
|--|--|
| 1.7. Persoonsdata zijn beschermd bij het uitwisselen van data | - Deze standaard gaat er vanuit dat autorisatie- en beveiligingsmechanismen worden toegepast rondom Applicatie en Logboek, daarom zijn er geen extra richtlijnen op dit vlak. Zie ook Beleidskader . |
| 1.8. Uitwisseling van data wordt gelogd als deze later aantoonbaar moet zijn | - Logging van verwerkte data is de kern van deze standaard. Door gebruikt te maken van een traceringsmechanisme en unieke identificatoren, kan er altijd worden voldaan aan de eis dat ontvangen en verzonden data aan elkaar gerelateerd kunnen worden. |
| 2.1. Datauitwisseling is federatief georganiseerd | <p>- Logboek Dataverwerkingen maakt het mogelijk om in een gefedereerde omgeving en in informatieketens verantwoording te kunnen afleggen over datauitwisseling. Hiervoor wordt tracing ingezet, een concept dat gebaseerd op de open standaard OpenTelemetry. Zie voor enkele juridische en beleidsmatige uitgangspunten het Juridisch Beleidskader hoofdstuk 6 en hoofdstuk 8.</p> <p>- Nadere invulling t.a.v. datauitwisseling in het kader van inzage zal volgen als de extensie voor inzage wordt gemaakt.</p> |
| 2.2. Voorwaarden en afspraken zijn expliciet en inzichtelijk | - Afspraken in het kader van deze standaard zullen vooral gemaakt worden op het gebied van tracering van data binnen organisaties en over organisaties heen. |
| 3.1. Gemeenschappelijke begripsvorming is het startpunt | - De data die gebruikt worden in deze standaard, staan vermeld en uitgelegd in het Canoniek datamodel. Het is van belang dat alle (overheids)organisatie die gebruik maken van de standaard hetzelfde beeld hebben ten aanzien specifieke data en uitwisseling daarvan om foutsituaties en verwarring te voorkomen. |
| 3.2. Metadata zijn begrijpelijk voor mensen | - De metadata zijn veelal ontstaan op basis van de internationale standaard OpenTelemetry. Daarnaast worden de begrippen ook uitgelegd in het Canoniek Datamodel. |
| 3.3. Data worden contextrijk vastgelegd | - Het gebruik van metadata in deze standaard is essentieel. De context wordt uitgelegd in het Canoniek Datamodel. |
| 3.4. Metadata zijn aan elkaar verbonden | - Metadata tussen de verschillende Logboeken zijn aan elkaar gerelateerd middels de beschrijvingen en afspraken zoals gespecificeerd in de standaard . |
| 3.5. Metadata zijn beschikbaar als Linked Data | - De gedefinieerde metadata is gerelateerd aan de standaard NL-SBB – standaard voor het beschrijven van begrippen . |

| Architectuurprincipe | Relatie met de standaard |
|--|--|
| 4.1. Data worden geleverd vanuit herbruikbare datadiensten | <ul style="list-style-type: none"> - Nadere invulling volgt bij het ontwerp van de Inzage extensie. - Data in het Logboek mogen niet fysiek worden verwijderd; als ze niet meer geldig zijn dan wordt alleen vastgelegd dat ze niet meer geldig zijn (tenzij ze om juridische redenen vernietigd moeten worden). - De historische integriteit van Logboekdata is geborgd; oude data mogen niet worden niet overschreven. |
| 4.2. Registraties bieden historische data aan | <ul style="list-style-type: none"> - Nieuwe formele overheidsregistraties die worden ontwikkeld, moeten de formele historie van datawerking vastleggen conform de standaard Logboek dataverwerking. Ook de wijzigingen in Register van Verwerkingsactiviteiten worden toevoegd als nieuwe verwerkingsactiviteiten met een eigen unieke identicator. Bestaande verwerkingsactiviteiten mogen niet wijzigen of verwijderd worden. Hierdoor blijven de oude verwijzingen uit de Logboek Dataverwerking intact. |
| 4.3. Aanbieders kunnen notificeren over belangrijke gebeurtenissen | <ul style="list-style-type: none"> - N.v.t. Standaard beschrijft geen notificatiemechanisme voor wijzigingen in de Log. |
| 5.1. Informatieproducten zijn herleidbaar naar de onderliggende data en regels | <ul style="list-style-type: none"> - Door gebruikt te maken van een traceringsmechanisme en unieke identificatoren, kan er altijd worden voldaan aan de eis dat ontvangen en verzonden data aan elkaar gerelateerd kunnen worden. De bron, en daarmee de kwaliteit en betrouwbaarheid van verwerkte data, kunnen snel en eenvoudig worden opgehaald. - Wordt eventueel nader ingevuld bij de ontwikkeling van een Inzage extensie. |
| 6.1. Data worden zo vroeg mogelijk gevalideerd | <ul style="list-style-type: none"> - Validatie van logdata is een taak van de Applicatie zelf, deze standaard geeft hiervoor geen handreiking. |

§ 3.2 De relatie tussen logboekelementen, waarom eigenlijk?

Logging van dataverwerkingen kunnen vaak en veelvuldig plaatsvinden. Het geheel kan groot en complex worden want sommige Logregels zijn aan elkaar gerelateerd. Deze relaties kunnen gelegd worden met Logregels met andere applicaties binnen dezelfde organisatie of met logregels van applicaties van andere organisaties.

Maar ook zijn er relaties nodig met activiteiten in het Register van Verwerkingsactiviteiten. Wat nu als alle Logregels zonder relaties worden opgeslagen? Bij een rapportage (bijvoorbeeld een verzoek tot inzage van een burger) moet nu handmatig worden uitgezocht welke dataverwerkingen bij elkaar horen en er moet, in het ernstigste geval, ook contact worden gezocht met andere organisaties om te onderzoeken of daar ook de nodige dataverwerkingen zijn uitgevoerd. Als er bij elke Logregel de nodige relatiedata worden bijgevoegd, kan de rapportage snel en accuraat worden gegenereerd.

§ 3.2.1 Welke relatiedata moeten er dan worden opgeslagen per Logregel?

Om er zeker van te zijn dat de relatie tussen Logregels gelegd kan worden, moeten de volgende data worden geregistreerd per Logregel:

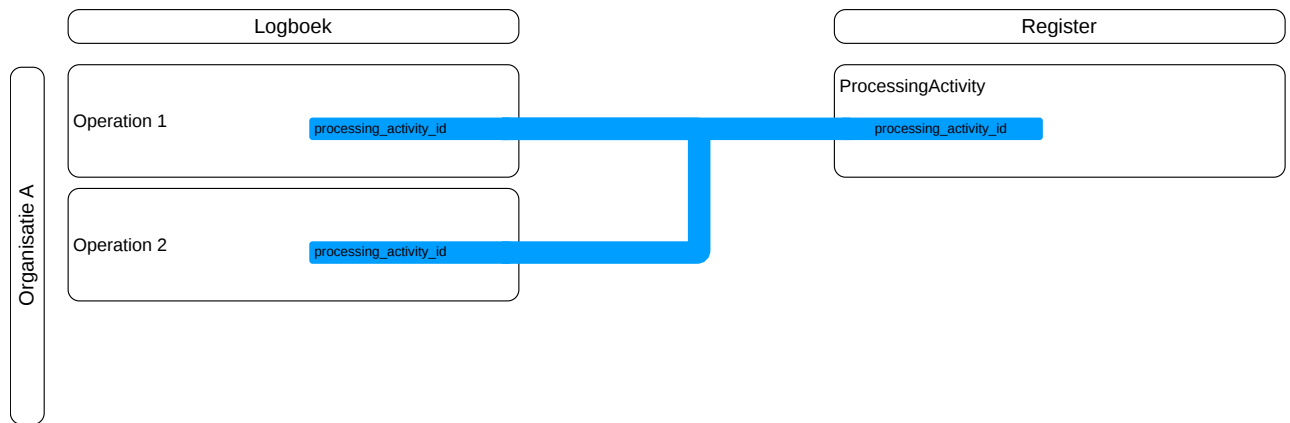
- `processing_activity_id`: elke dataverwerking die een organisatie doet, moet bekend zijn in het Register van Verwerkingsactiviteiten. Het `processingActivity` legt de relatie tussen de dataverwerking door een applicatie, en de activiteit gedefinieerd in het Register.
- `trace_id`: alle logregels die voor een specifieke dataverwerking bij elkaar horen, krijgen een `trace_id`. De `trace_id`-waarde voor alle Logregels die bij elkaar horen is hetzelfde.
- `span_id`: elke individuele Logregel (Operation) krijgt een eigen, unieke `span_id` (net zoals elk databaserecord dat ook krijgt).

In werkelijkheid worden alle relaties door de Applicatie in een fractie van een seconde (in parallel) gelegd. Om het grote geheel beter te begrijpen, worden alle relaties hieronder stap voor stap uitgelegd.

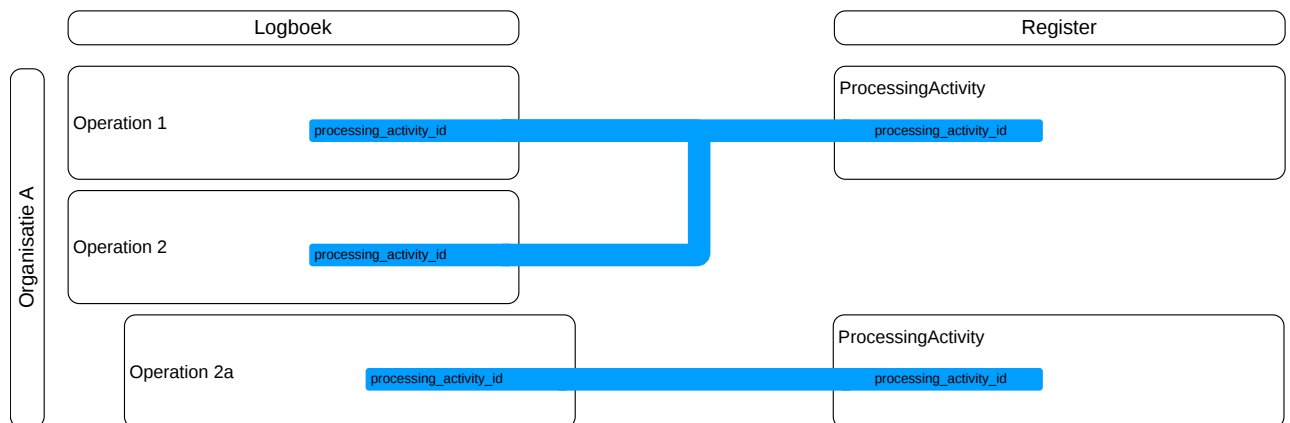
§ 3.2.2 Het logboek en het Register van Verwerkingsactiviteiten

Als er een Dataverwerking plaatsvindt, moet dit altijd een relatie hebben met het Register van Verwerkingsactiviteiten. In dit Register staat informatie over de data die een organisatie verwerkt. Het Register is verplicht, een geautomatiseerde koppeling met het Logboek niet.

Bij elke Dataverwerking wordt door het Logboek een relatie gelegd met het Register door middel van het `processing_activity_id`. Als er meerdere dezelfde Dataverwerkingen ('Operations') zijn, krijgen deze dus allemaal dezelfde `processing_activity_id`.

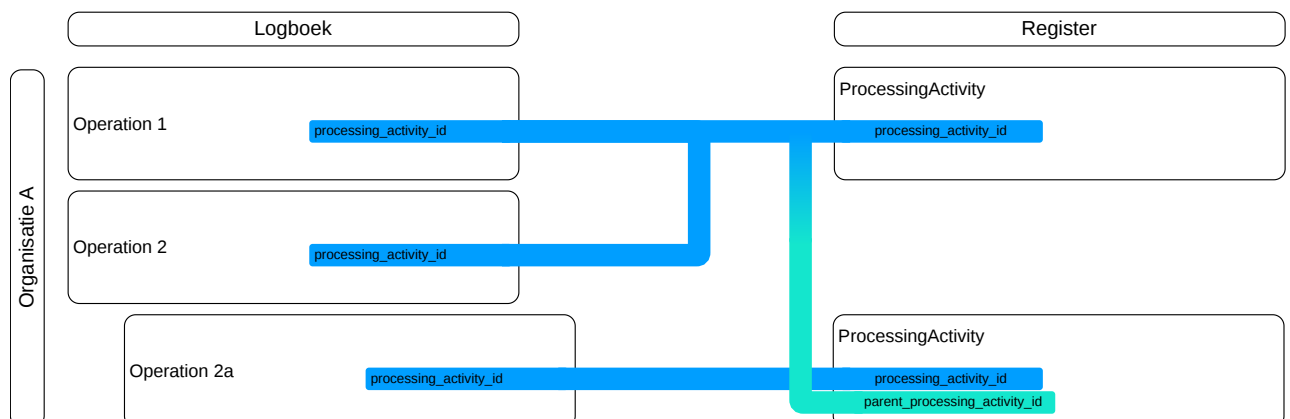


In het geval er een Dataverwerking plaatsvindt ter ondersteuning van een andere Dataverwerking (suboperation), dan kan deze ondersteunende Dataverwerking een eigen `processing_activity_id` krijgen. Deze kan anders zijn dan het `processing_activity_id` van de 'hoofdprocessingActivity'.



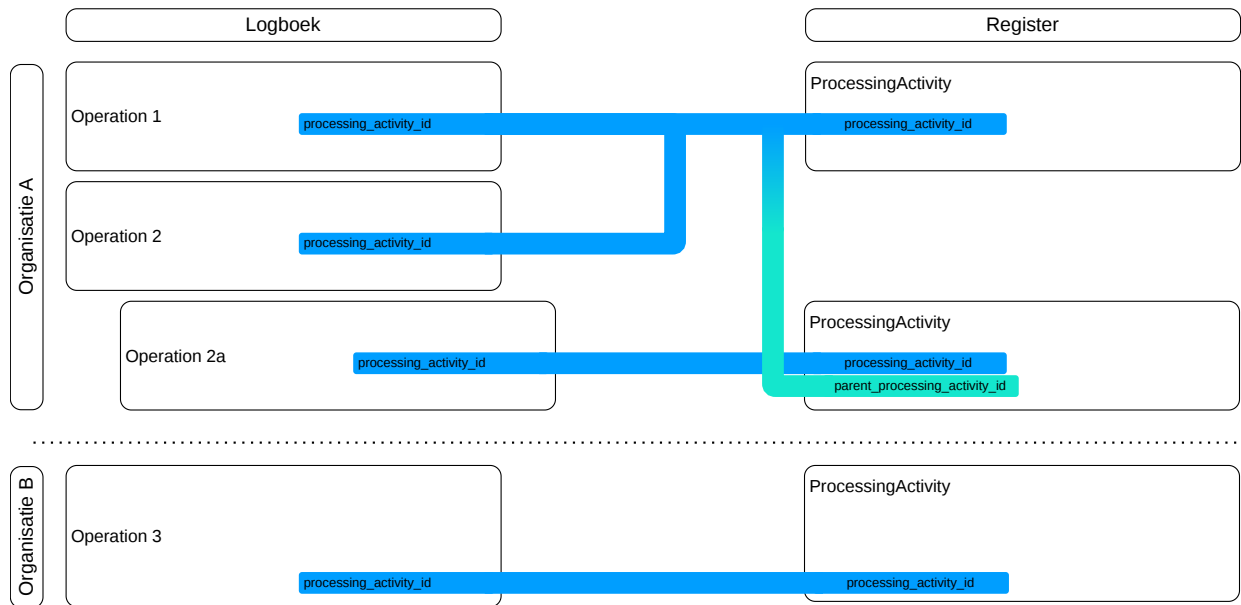
De subOperation heeft nu een eigen `processing_activity_id` gekregen, maar het is nog niet duidelijk aan welke hoofdprocessingActivityId deze gekoppeld is. Om dit op te lossen, wordt ook een 'parentProcessingActivityId' geregistreerd.

Bij de subOperation wordt in dit geval naast de `processing_activity_id` ook een `parentProcessingActivityId` geregistreerd. De waarde van deze `parentProcessingActivityId` is gelijk aan de waarde van het hoofdProcessingActivityId.



Bij een Dataverwerking kan het zijn dat data moeten worden opgevraagd bij een andere organisatie. Deze organisatie heeft zelf ook een Register van Verwerkingsactiviteiten. In dit Register staat beschreven dat een specifieke organisatie specifieke gegevens mag opvragen als aparte operation.

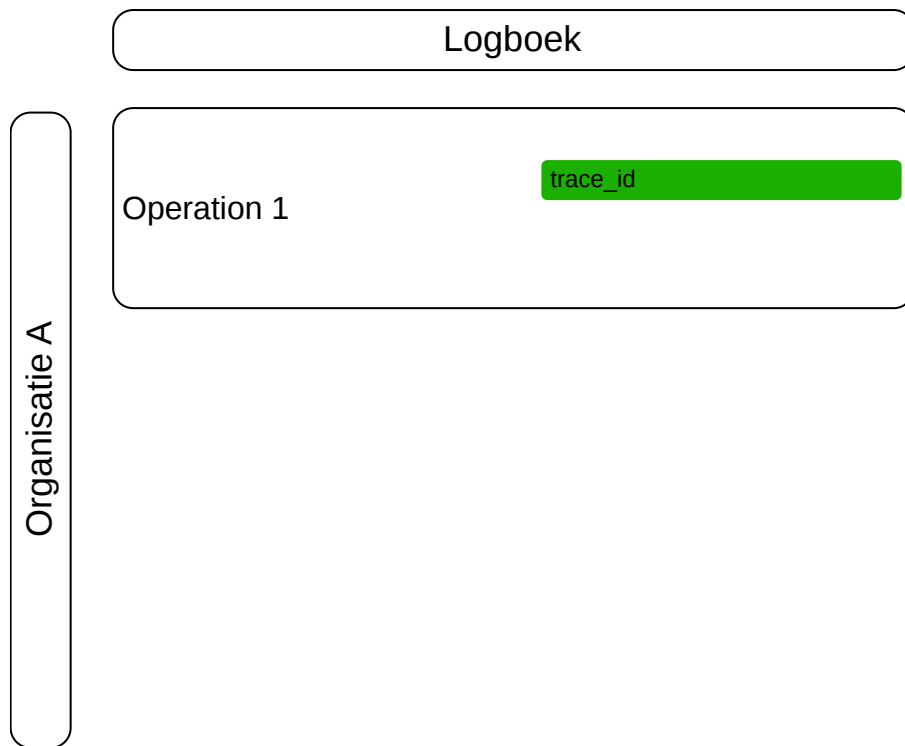
Bij het verstrekken van deze data aan de aanvragende organisatie, wordt het `processing_activity_id` van de gegevensverstreckende organisatie geregistreerd. Er is dus GEEN rechtstreekse koppeling tussen het Register van de aanvragende en het Register van de verstreckende organisatie.



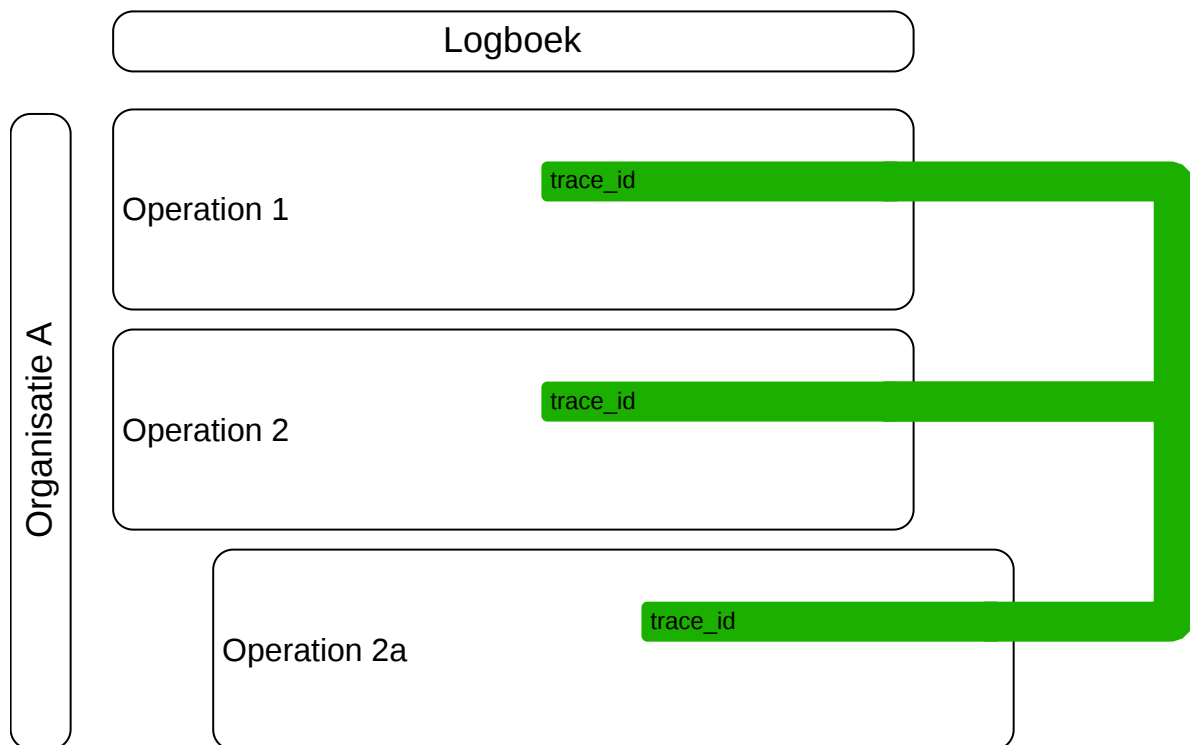
§ 3.2.3 `trace_id` als grootste gemene deler

Operations kunnen bestaan uit meerdere (sub)Operations binnen de eigen organisatie maar ook over organisaties heen. Het geheel kan een grote en ingewikkelde constructie worden. Om toch het overzicht te kunnen behouden, is het noodzakelijk een `trace_id` te introduceren per (sub)Operation.

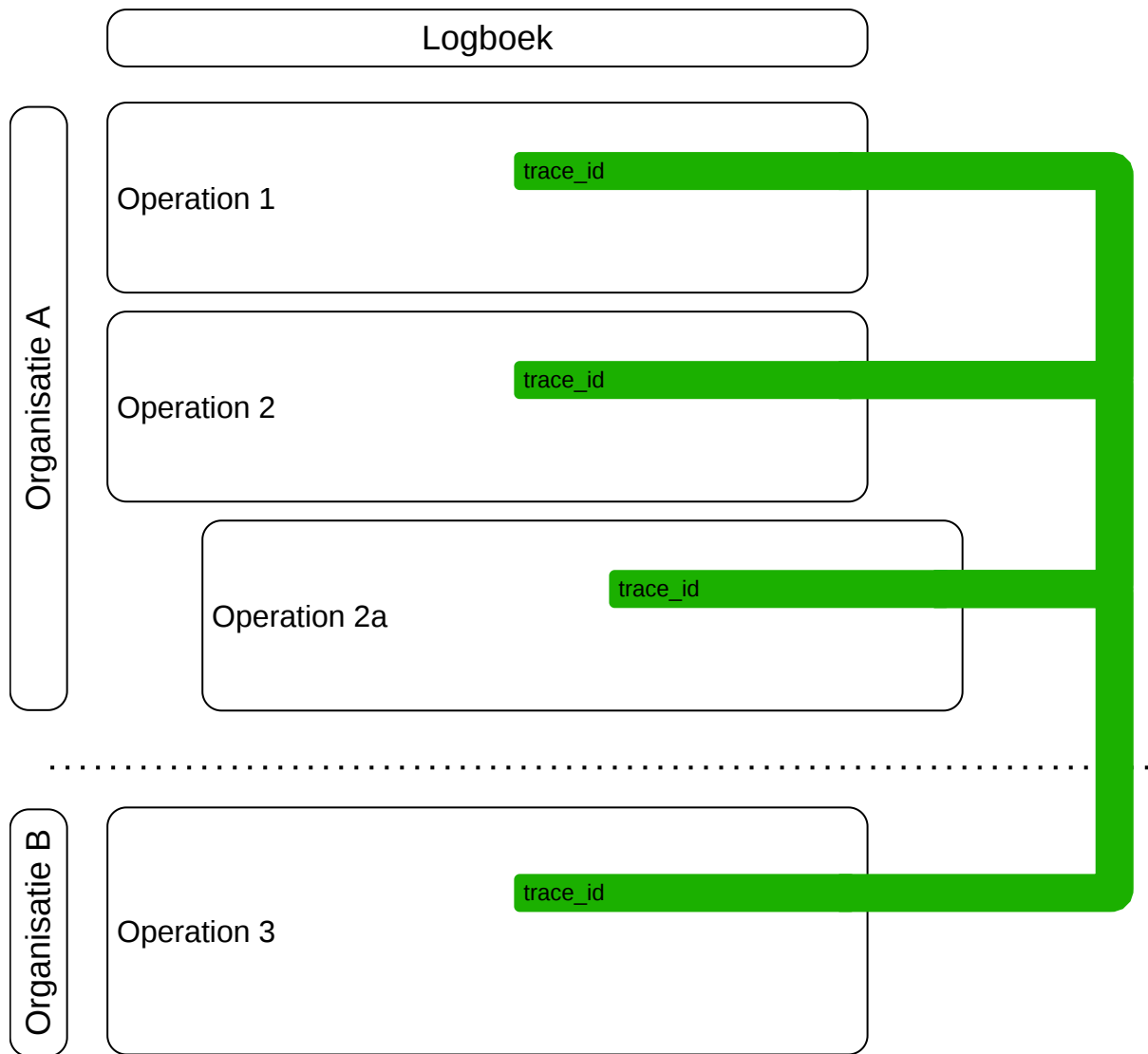
Het `trace_id` is als het ware de 'lijm' tussen alle (sub)Operations. Als er nog geen `trace_id` bekend is, wordt deze automatisch gegenereerd voor de eerste Operation.



Alle bij elkaar horende (sub)Operations, krijgen vervolgens dezelfde `trace_id`-waarde.

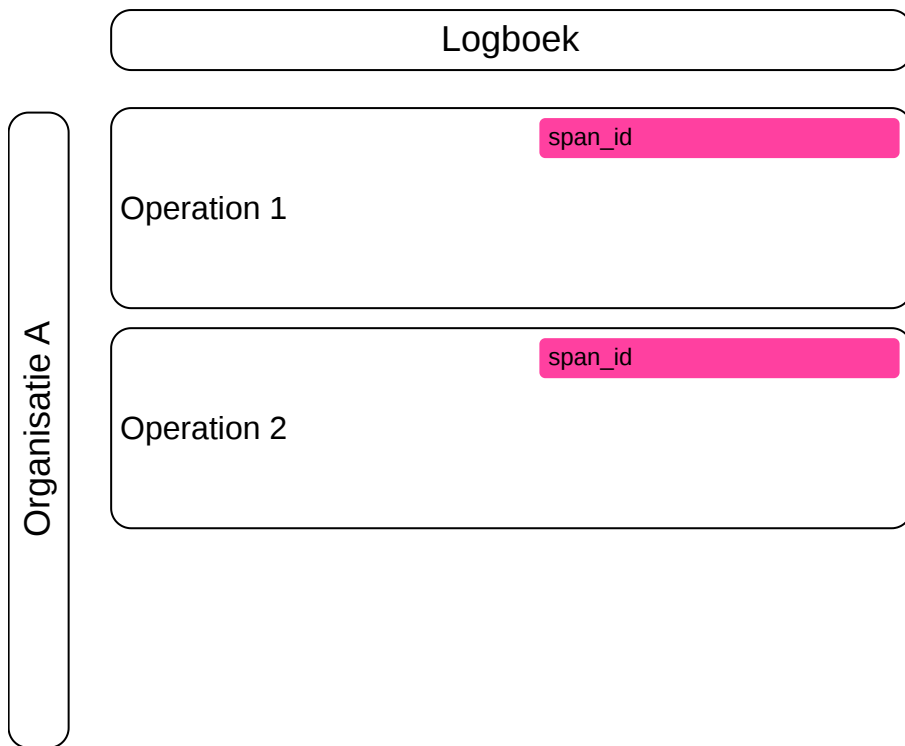


In het geval er data wordt opgevraagd aan een andere organisatie, krijgt elke operation bij verstreckende organisatie een `trace_id`. Om de relatie te leggen tussen de vragende en de verstreckende organisatie, wordt bij elke Operation van de verstreckende organisatie een `trace_id` geregistreerd. De waarde van de `trace_id` van de verstreckende organisatie is gelijk aan de waarde van `trace_id` van de vragende organisatie.

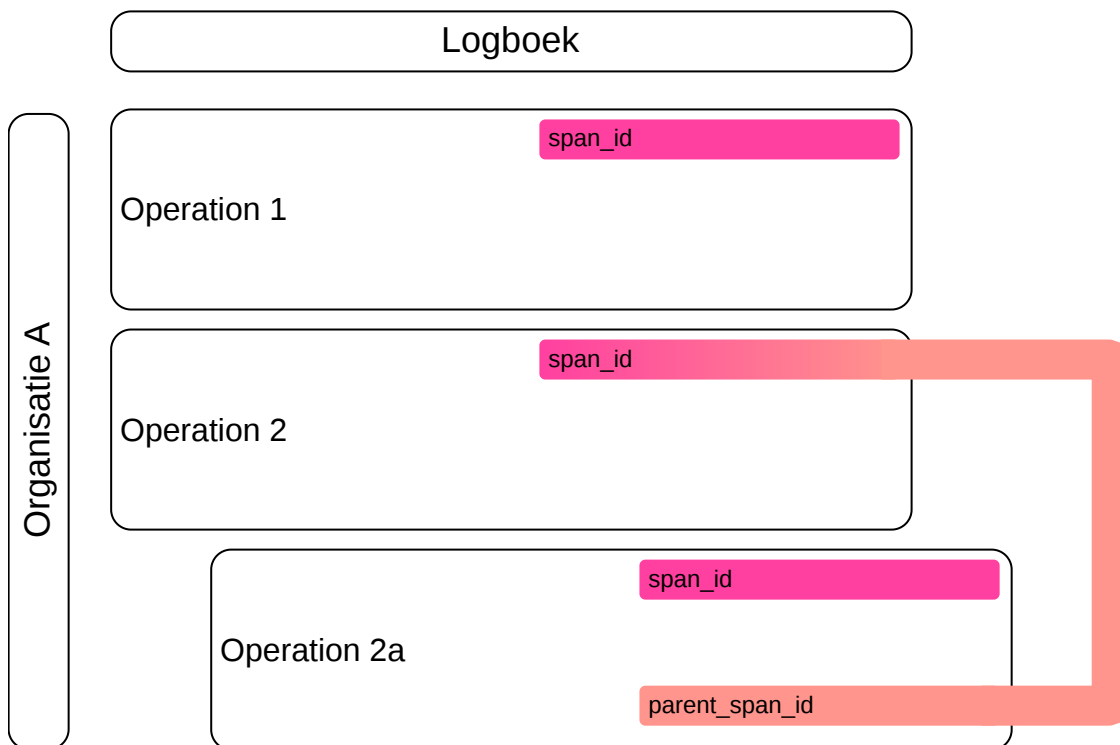


§ 3.2.4 Relatie tussen (sub)Operations

Elke (sub)Operation krijgt een eigen, unieke `span_id`. Hiermee zijn alle loggings altijd uniek traceerbaar. Ook subOperations krijgen een eigen, unieke `span_id`.

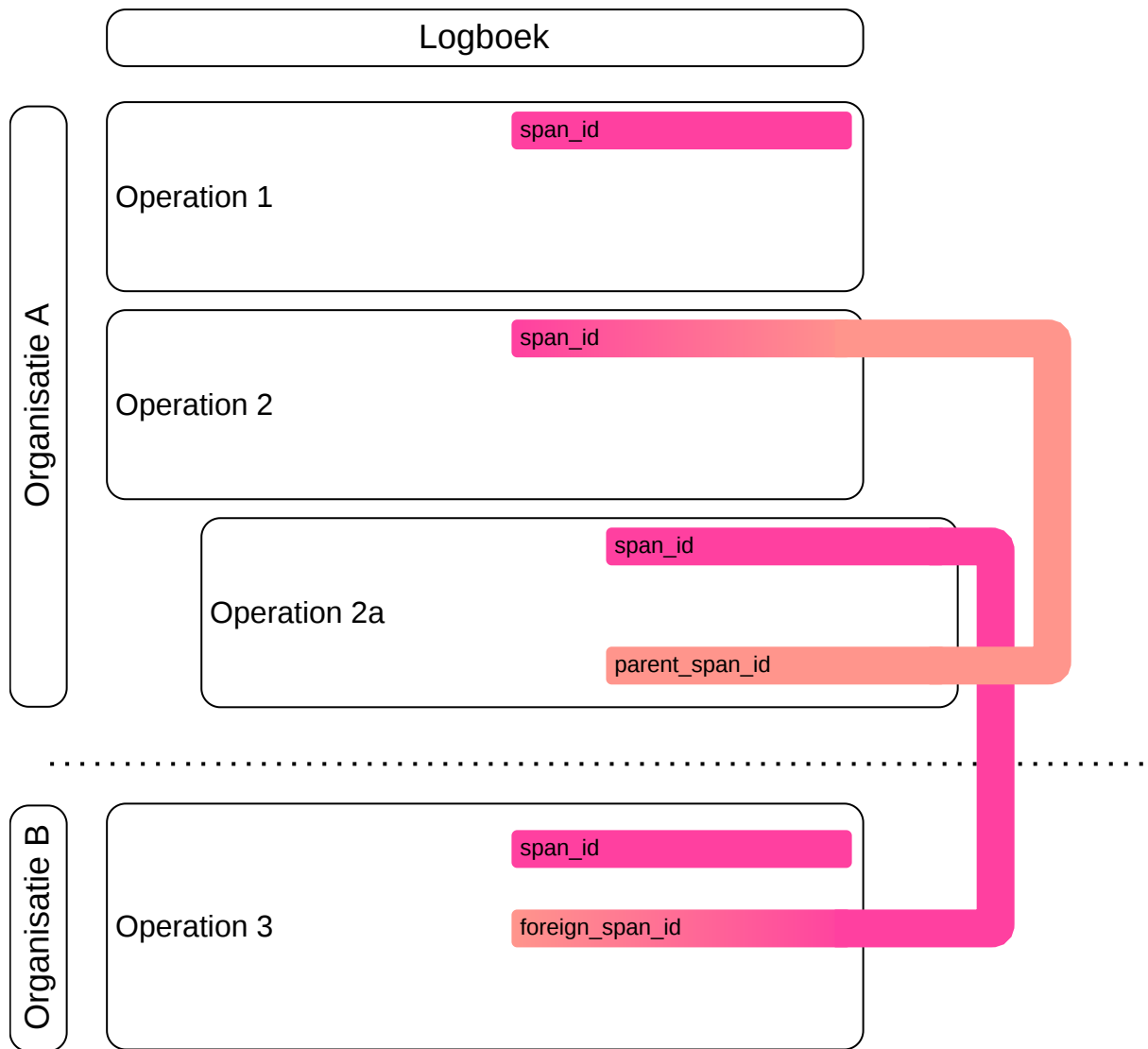


Als er ook subOperations plaatsvinden, moet er ook een `parent_span_id` worden geregistreerd om de koppeling met de hoofdOperation te realiseren.



In het geval er data nodig is van een andere organisatie, krijgt de Operation van de verstrekende organisatie ook een eigen, unieke `span_id`.

Daarnaast wordt bij deze Operation ook het `span_id` geregistreerd die het verzoek voor informatie geïnitieerd heeft (vanuit de vragende organisatie). Deze specifieke `span_id` wordt het `foreign_operation._span_id` genoemd en krijgt de waarde gelijk aan het `span_id` van de initiërende Operation van de vragende organisatie.



§ 3.2.5 Voorbeeld van een traceringsconstructie

Het nu volgende voorbeeld is volledig fictief en is puur bedoeld om een beeld te schetsen ten behoeve van een traceringsconstructie in een logboek.

§ 3.2.6 Situatieschets

Een persoon heeft een parkeervergunning in een gemeente. Er is een nieuwe auto aangeschaft, het kenteken moet worden aangepast ten behoeve van de vergunning. De persoon kan het kenteken online wijzigen in de 'mijnGemeente' applicatie. Om het voorbeeld eenvoudig te houden, worden foutsituaties buiten beschouwing gelaten.

§ 3.2.7 Procesgang

1. Persoon logt in gemeenteapplicatie.
2. Gemeente toont huidige parkeervergunning.
3. Persoon wijzigt kenteken in de gemeenteapplicatie.
4. Gemeenteapplicatie vraagt het RDW om data op basis van de tenaamstelling.
5. RDW geeft de gevraagde data terug aan de gemeenteapplicatie.
6. Gemeenteapplicatie accepteert de wijziging van de persoon.

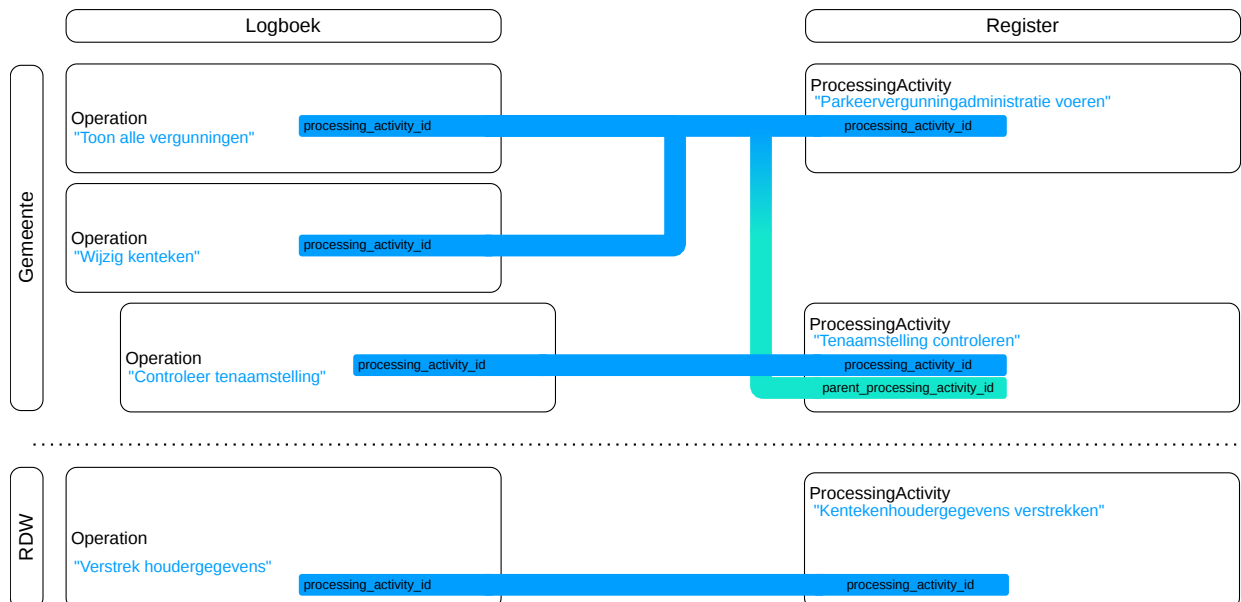
De traceringsdata worden als volgt vastgelegd:

§ 3.2.8 processing_activity_id

In de gemeenteapplicatie worden de volgende Operations uitgevoerd die een relatie hebben met het Register van Verwerkingsactiviteiten van de gemeente:

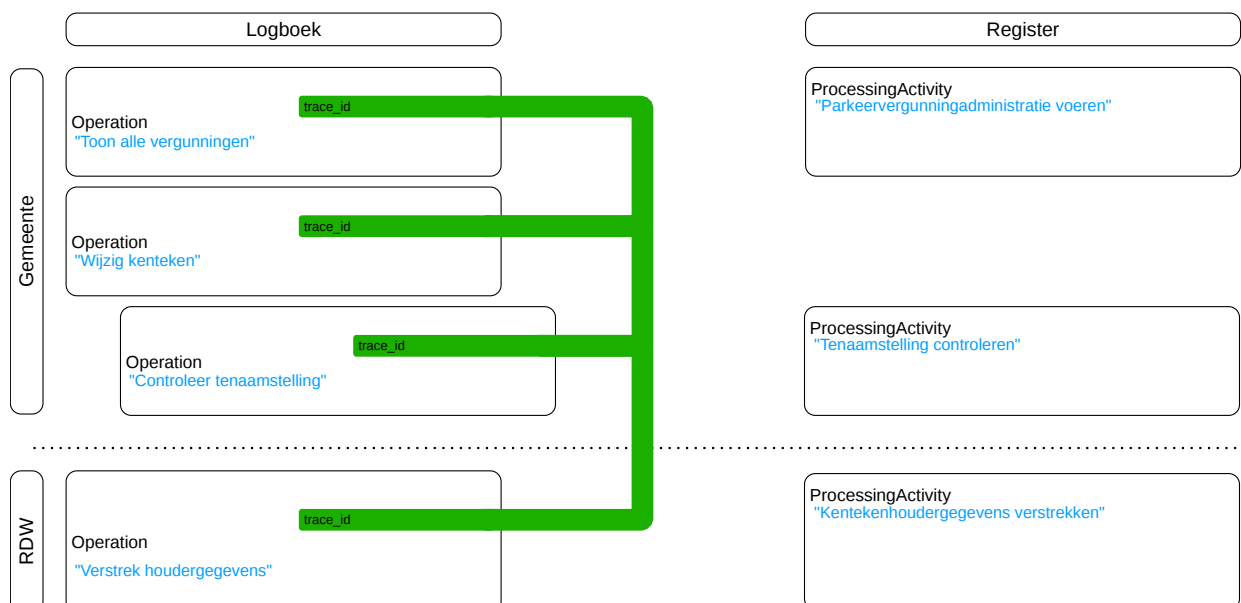
- **Toon alle vergunningen:** na het inloggen, worden de parkeervergunningen van de persoon getoond. Deze Operation is gerelateerd aan de processingActivity **Parkeervergunningadministratie voeren**.
- **Wijzig kenteken:** het wijzigen van het kenteken valt ook onder de processingActivity **Parkeervergunningadministratie voeren**. Hierdoor is het processing_activity_id hetzelfde als die van de Operation **Toon alle vergunningen**.
- **Controleer tenaamstelling:** deze Operation zorgt voor de aanvraag van data richting het RDW en controle van de teruggontvangen data. Deze Operation is een subOperation van **Wijzig kenteken** en krijgt een processingActivity wat hoort bij de processingActivity in het Register genaamd **Tenaamstelling controleren**. De processingActivity is op zijn beurt weer een subprocessingActivity van **Parkeeradministratie voeren**. Om deze relatie te leggen, moet ook een parentProcessingActivityId worden geregistreerd. De waarde hiervan is gelijk aan de waarde van het processing_activity_id van **Parkeervergunningadministratie voeren**.

In de RDW-applicatie wordt het verstrekken van data aan de gemeenteapplicatie ook geregistreerd. De Operation **Verstrek houdergegevens** is gerelateerd aan de processingActivity **Kentekenhoudergegevens verstrekken**. Merk op dat er hier dus GEEN directe relatie is tussen het Register van Verwerkingsactiviteiten van de gemeente en die van het RDW.



3.2.9 trace_id

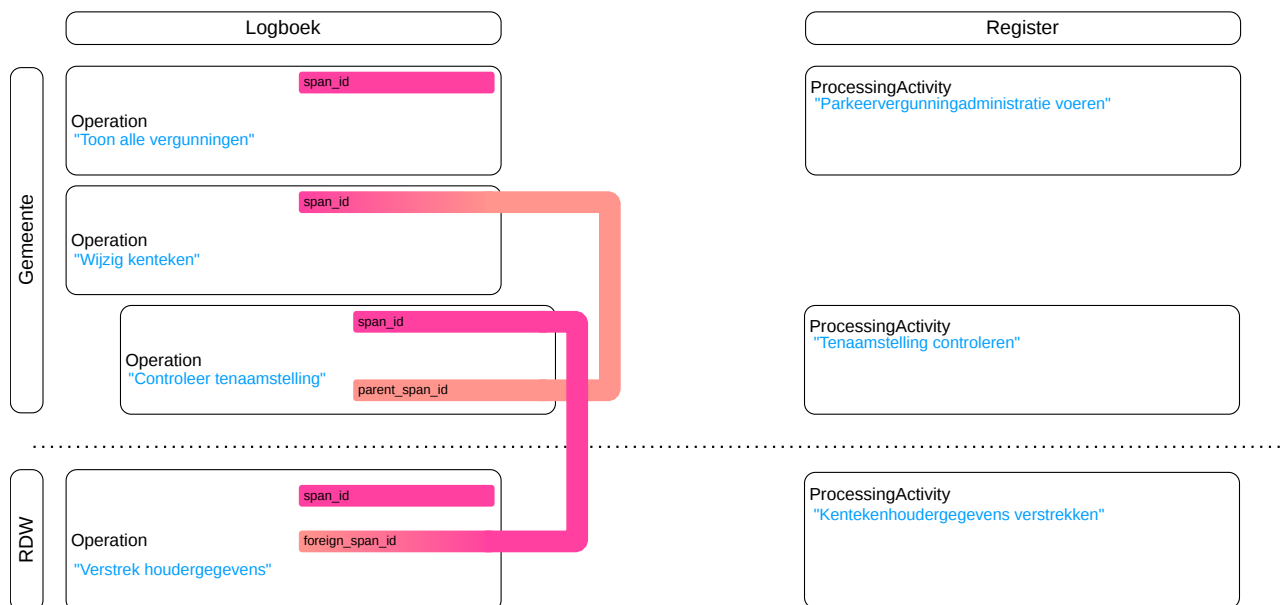
- De gemeenteOperations **Toon alle vergunningen**, **Wijzig kenteken** en **Controleer tenaamstelling** behoren tot dezelfde handeling (met als eindresultaat het wijzigingen van het kenteken op de vergunning). Deze Operations krijgen allemaal dezelfde trace_id.
- De RDW-Operation **Verstrek houdergegevens** krijgt een eigen trace_id.
- Om het geheel te koppelen over de organisaties heen, wordt bij het RDW ook een trace_id opgeslagen, de waarde hier van is gelijk aan de waarde van de trace_id van de Operation **Controleer tenaamstelling**.



§ 3.2.10 span_id

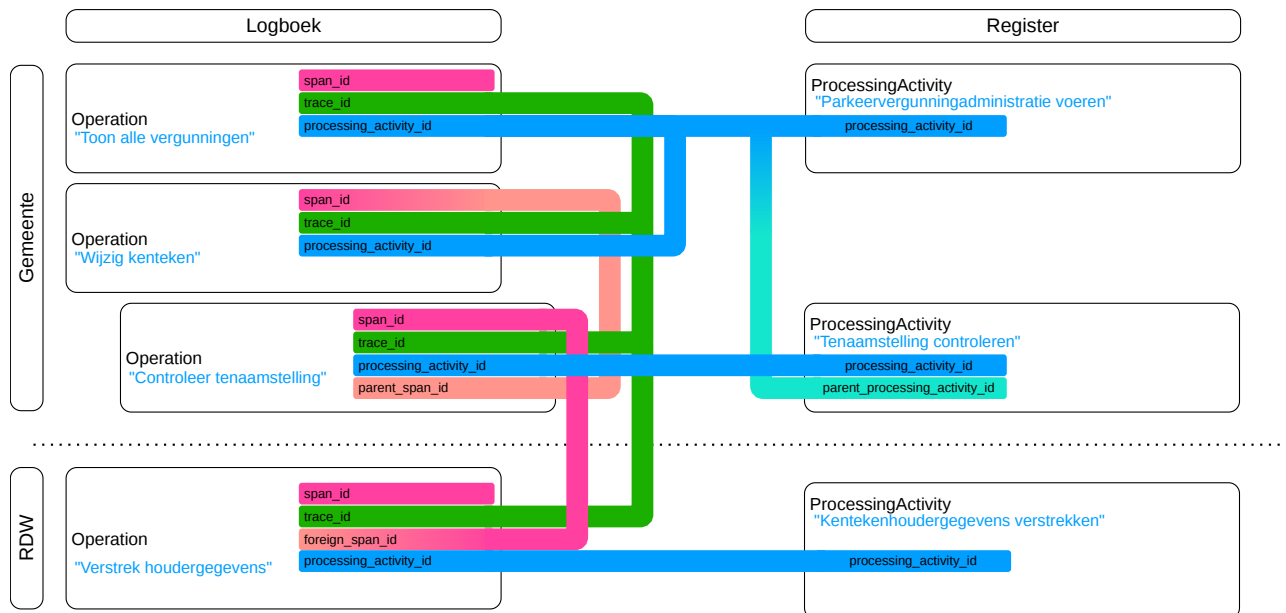
In de gemeente-applicatie krijgt elke (sub)Operation een eigen, unieke span_id.

- De (sub)Operation **Controleer tenaamstelling** krijgt daarnaast ook nog een parent_span_id met de waarde van span_id van de **Operation Wijzig** kenteken om een relatie te leggen.
- Ook de RDW-Operation **Verstrek houdergegevens** krijgt een eigen unieke span_id.
- Om de relatie over de organisaties heen te leggen, wordt er bij de RDW-Operation **Verstrek houdergegevens** ook een foreign_operation.span_id moeten worden vastgelegd. De waarde van deze foreign_operation.span_id is gelijk aan de waarde van de span_id van de gemeente-Operation **Controleer tenaamstelling**.



§ 3.2.11 Totaalbeeld

Als alle relaties gelegd zijn, ziet de traceringsconstructie er als volgt uit:



Meer gedetailleerde voorbeelden staan beschreven in de [inleiding](#) van de standaard logboek dataverwerkingen.

§ 4. Besluitenlijst

Deze sectie is *tijdelijk* en *niet normatief*, bedoeld om informatie te geven over achterliggende afwegingen bij de standaard.

In de definitieve standaard wordt deze lijst niet opgenomen, omdat veel afwegingen specifiek zijn voor de context van de Nederlandse overheid waarin deze standaard is ontstaan. De standaard is breder inzetbaar, en voor de implementatie is het niet relevant om de afwegingen bij alle aspecten van de standaard in de context van de Nederlandse overheid te kennen.

Een aantal besluiten en overweging hebben relatie met het [Juridische Beleidskader](#)

§ Logregels bevatten alleen wat nodig is voor verantwoording door verantwoordelijke

Dit onderdeel is niet normatief.

- Status: draft
- Datum: 2024-06-24

§ Context en probleemstelling

Vanuit de wens om zoveel mogelijk context vast te leggen om zo een compleet beeld te schetsen van wat er is gebeurd rond een [Dataverwerking](#) kan de neiging ontstaan om informatie uit andere organisaties vast te leggen in de logregels.

Hierdoor kom je (juridisch gezien) al snel in lastig vaarwater. Er worden dan zaken vastgelegd die niet noodzakelijk zijn voor het verantwoorden van het handelen. Het is namelijk mogelijk om een compleet beeld te krijgen door de informatie te laten in de organisatie waar een dataverwerking is uitgevoerd. Vanuit het oogpunt van dataminimalisatie is dit een betere oplossing.

Voor de uitoefening van het Inzagerecht is de consequentie dat de Betrokkene informatie uit alle organisaties moet ophalen en deze volgens een paar relatief eenvoudige businessrules aan elkaar moet relateren voor het verkrijgen van een compleet beeld. Dit kan door alle organisaties te bevragen, of door gericht bij één organisatie te beginnen en vervolgens de [URI's](#) te volgen naar Logregels in andere organisaties.

Het kan zijn dat organisatie A de logs wel op orde heeft, en organisatie B (nog) niet. Dan is het resultaat dat geen compleet beeld kan worden gegeven. Daarmee komt de prikkel tot verbetering op de juiste plek, namelijk bij de organisatie die het Logboek nog niet op orde heeft.

§ Besluit

Logregels bevatten alleen wat nodig is voor verantwoording door de Verantwoordelijke.

§ Gevolgen

- In logregels wordt alleen een identifier vastgelegd van gerelateerde Dataverwerkingen in een andere context (bijv. een andere organisatie), geen inhoudelijke informatie
- Voor een analyse, bijvoorbeeld in het kader van een audit of uitoefening inzage-recht, is het nodig om op dat moment de URI's naar logs in andere organisaties te volgen

§ Logregels bevatten geen gegevens die al vastliggen in een Register

Dit onderdeel is niet normatief.

- Status: proposed
- Datum: 2024-02-01

§ Context en probleemstelling

Om logs zo begrijpelijk mogelijk te maken is het aantrekkelijk om de benodigde informatie redundant weg te schrijven in elk logrecord, zodat er geen afhankelijkheid bestaat van andere bronnen.

Dit heeft nadelen, zoals:

- Wanneer de statische data (zoals bewaartermijn, verantwoordelijke, etc.) wijzigen, zou dit moeten worden aangepast in alle Logregels. Dat verhoudt zich slecht tot het 'immutable' (onveranderbaar) zijn van deze Logregels.
- De grote vrijheid in alle clients om invulling te geven aan deze data leidt er vrijwel zeker toe dat verdere divergentie optreedt. Dit heeft o.a. tot gevolg dat het lastig wordt om te rapporteren uit de logs
- De API voor het wegschrijven van logs wordt ingewikkeld en relatief traag voor het wegschrijven van records
- We niet voldoen aan alle uitgangspunten die horen bij dataminimalisatie

In de gewenste situatie:

- staan alle statische data in het Register van de Verwerkingsactiviteiten (RvVA), en bevatten Logregels verwijzingen naar dat Register. Specifiek gaat dit om de resources '[verwerkingsactiviteiten](#)' en 'organisaties'.
- kan bij het configureren van clients in de RvVA-API worden opgezocht welke organisaties en verwerkingsactiviteiten van toepassing zijn
- kunnen wijzigingen in verwerkingsactiviteiten worden doorgevoerd zonder dat Logregels gewijzigd behoeven te worden

Met name het wegschrijven van logs kan op deze manier met hogere performance worden uitgevoerd. Dit kan nog verder worden geoptimaliseerd door niet te vereisen dat dit middels REST

API calls gebeurt, maar een interface te definiëren die kan worden geïmplementeerd met bijvoorbeeld gRPC of andere streaming protocollen.

Wanneer het aan de gebruiker is om in de software die de Logboek API aanroept de namen van acties, de vertrouwelijkheid en de bewaartermijn te bepalen, zal de invulling daarvan op allerlei manieren uiteen gaan lopen. Door dit in het RvVA te bepalen zal eerder uniformering plaatsvinden. De vulling van RvVA's kan waarschijnlijk zelfs in hoge mate gestandaardiseerd worden.

Met meer gestandaardiseerde namen en bewaartermijnen en een eenduidige omgang met vertrouwelijkheid is het ook eenvoudiger om eenduidige te communiceren naar de [Betrokkene](#). Bijvoorbeeld: een portaal dat aan de Betrokkene toont hoe de persoonsdata zijn verwerkt, is lastig vorm te geven wanneer in de praktijk blijkt dat software-leveranciers verschillende interpretaties hebben van het niveau waarbij sprake is van een verwerking, handeling of actie. Eenduidige interpretatie is cruciaal, en dit kan waarschijnlijk alleen in het RvVA.

Overigens werkt het conceptueel wél wanneer men geen API op het RvVA aanbiedt, deze link kan ook handmatig worden gelegd iedere keer als deze informatie nodig is, en het RvVA bijvoorbeeld alleen bestaat als Excel document.

§ Besluit

Logregels bevatten geen informatie over Verwerkingsactiviteiten en Verantwoordelijkheden die al vastliggen in een Register

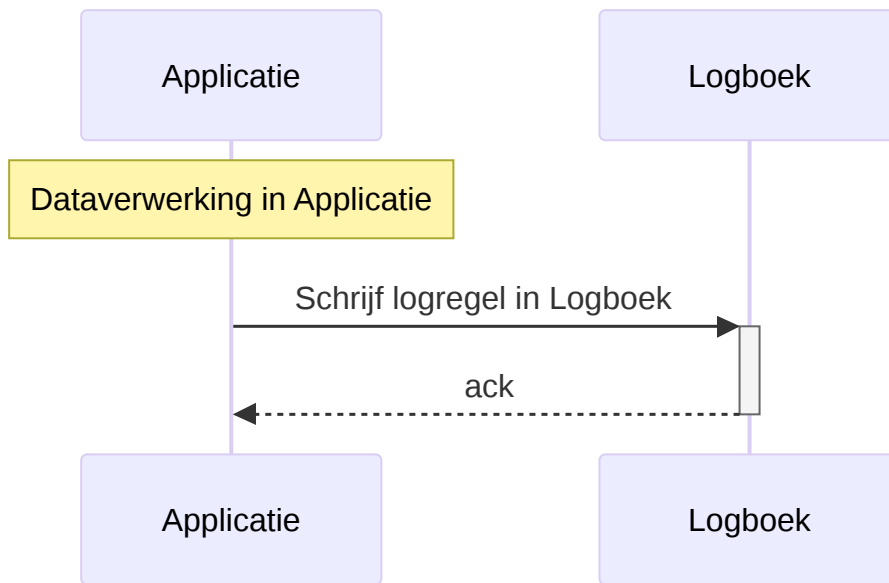
§ Gevolgen

- In de standaard Logboek Dataverwerkingen is het nodig om ook de benodigde interface op de RvVA te standaardiseren. Dit is nodig om de logs geautomatiseerd en realtime te kunnen interpreteren: zonder een gestandaardiseerde manier om informatie over verwerkingsactiviteiten op te vragen kan men aan logregels niet zien of het verwerkingen, handelingen of acties betreft.

Met de volgende sequentie diagrammen wordt in beeld gebracht wat de gevolgen zijn voor de diverse flows in het gebruik van de standaard.

§ Loggen van een verwerking

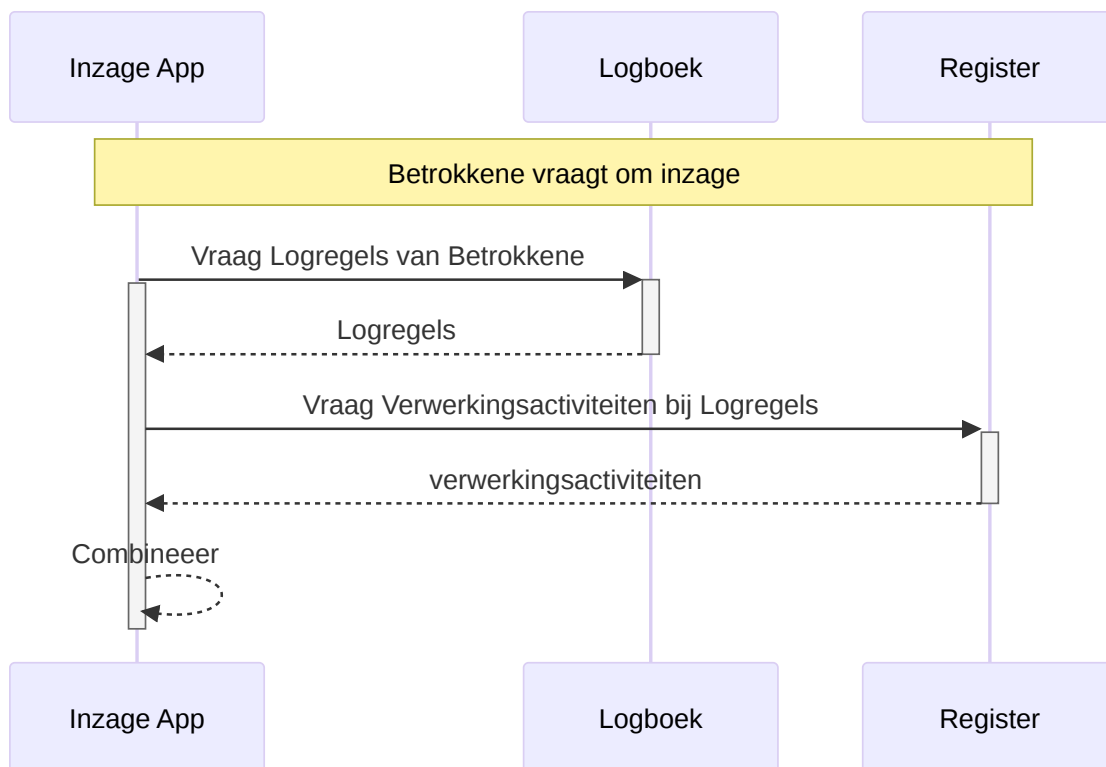
Het wegschrijven van een verwerking in de log-API is uiterst simpel:



Deze transactie is geoptimaliseerd op eenvoud en snelheid, want deze heeft rechtstreeks invloed op de snelheid van verwerkingen. Deze transactie moet schaalbaar zijn naar bijv. tienduizenden transacties per seconde.

§ Tonen van een verwerking

Om verwerkingen op een significante manier te tonen aan bijvoorbeeld een Betrokkene, is het noodzakelijk om data op te vragen uit zowel de logs als het RvVA. Deze flow mag wat complexer zijn, omdat deze niet voor alle vastgelegde data wordt uitgevoerd en het belang van de bevraging rechtvaardigt dat een bevraging wat langer kan duren.



§ Bewaartermijnen worden in het Profiel vastgelegd

Dit onderdeel is niet normatief.

- Status: draft
- Datum: 2024-05-23

§ Context en probleemstelling

Logregels moeten op enig moment worden verwijderd. Wanneer?

Voor vrijwel alle vastgelegde data geldt dat deze op enig moment moeten worden vernietigd of overgebracht naar een archief. Dit geldt ook voor Logregels.

Anders dan bij data over rechtsfeiten zullen Logregels typisch allemaal dezelfde bewaartermijn hebben. Het kan zijn dat de Dataverwerking waar het logrecord betrekking op heeft leidt tot data waarvoor complexe bewaartermijnen gelden (bijvoorbeeld een dynamische termijn die duurt totdat Betrokkene is overleden gevolgd door een statische termijn van enkele tientallen jaren). De Logregels die de Dataverwerking beschrijven kennen deze complexe bewaartermijn niet, deze

kunnen statisch zijn en generiek worden vastgesteld per organisatie of eventueel per verwerkingsactiviteit. Het is aan de organisatie zelf om daarin keuzes te maken.

Voor samenwerkende organisaties die zich ten doel stellen om gezamenlijk op eenduidige manier te verantwoorden over dataverwerkingen kan het nuttig zijn afspraken voor bewaartermijnen vast te leggen in een Profiel.

§ Besluit

Bewaartermijnen worden in het Profiel vastgelegd.

§ Gevolgen

- In de Logregel liggen geen data vast over bewaartermijnen.
- Vanuit een beheercomponent (een applicatie die functionaliteit biedt voor beheren van logboek. Is een kwestie van implementatie en valt buiten scope van deze standaard) kunnen Logregels worden verwijderd door te kijken naar de datum van de Logregel in relatie tot de bewaartermijn die de organisatie hanteert voor Logregels. Deze bewaartermijn kan gezamenlijk zijn afgesproken en ligt dan vast in het Profiel.

§ Geen data over gebruikers in logregels

Dit onderdeel is niet normatief.

- Status: draft
- Datum: 2024-05-23

§ Context en probleemstelling

Om te verantwoorden dat een dataverwerking correct is uitgevoerd is het nodig te weten wie de dataverwerking heeft geïnitieerd, zodat kan worden nagegaan dat dit met de juiste autorisatie is gedaan.

De wens zou kunnen bestaan om in elke logregel vast te leggen welke gebruiker een rol heeft gehad bij de betreffende Dataverwerking.

Echter, de vastlegging van een handeling van een gebruiker als medewerker van een organisatie betreft ook een Dataverwerking die onder de AVG valt, waardoor rechten ontstaan voor de betreffende gebruiker om Inzage te verkrijgen. De vastlegging van de betrokkenheid van de gebruiker is een Dataverwerking op zich. Door een dergelijke vastlegging in de logregels te doen ontstaat een ongewenste recursiviteit.

Ook is de relatie van de gebruiker tot de Dataverwerking niet eenvoudig eenduidig te modelleren, o.a. omdat bij een enkele Dataverwerking meerdere gebruikers in meerdere rollen betrokken kunnen zijn.

Daarnaast kan het goed zijn dat de Dataverwerking in het Audit Log (Een log file die activiteiten van gebruikers, uitzonderingen en informatiebeveiligingsgebeurtenissen vastgelegd. Dit is o.a. vanuit BIO verplicht) onder een andere Verantwoordelijke valt dan de Dataverwerking die op dat moment door de gebruiker wordt uitgevoerd. Bijvoorbeeld:

- Een Dataverwerking wordt door een gebruiker bij een [Verwerker](#) uitgevoerd
- De Dataverwerking valt onder verantwoordelijkheid van de [Verantwoordelijke](#), namelijk de organisatie die de Verwerker heeft ingehuurd
- De Audit Log is een aparte Dataverwerking die valt onder verantwoordelijkheid van de Verwerker, in de rol van Verantwoordelijke over de eigen bedrijfsvoering.

Het is daarom zuiverder om een andere oplossing te kiezen, namelijk:

- Betrokkenheid van gebruiker wordt vastgelegd in een Audit Log (buiten scope van deze standaard)
- In het Audit Log kan eventueel een relatie worden gelegd met het Processing ID dat ook in het Logboek Dataverwerkingen wordt gebruikt
- Iedere keer dat in het Audit Log data over een gebruiker worden vastgelegd, moet tevens een Dataverwerking worden gelogd in het Logboek Dataverwerkingen.

Let wel, deze Dataverwerking is een *andere Dataverwerking* dan de Dataverwerking die op dat moment wordt uitgevoerd door de Gebruiker, heeft een eigen Trace Context, en wordt gerelateerd aan een andere Verwerkingsactiviteit.

§ Besluit

In logregels worden geen identificerende data over gebruikers van de Applicaties vastgelegd.

§ Gevolgen

- In gevallen dat het nodig is te achterhalen welke gebruiker een specifieke Dataverwerking heeft uitgevoerd, moet dit worden achterhaald door de Dataverwerking te koppelen aan het Audit Log (buiten scope van de standaard).
- Het koppelen van Dataverwerking aan Audit Log is mogelijk door in Audit Logs hetzelfde Processing ID op te nemen als in de logregel die in het Logboek Dataverwerkingen wordt opgenomen.

§ Standaard beschrijft geen interface voor verwijderen van logs

Dit onderdeel is niet normatief.

- Status: draft
- Datum: 2024-05-24

§ Context en probleemstelling

Logregels moeten op enig moment worden vernietigd. Moet er een interface in de standaard worden gedefinieerd voor het verwijderen van vastgelegde Logregels?

De wijze waarop Logregels worden weggeschreven is sterk afhankelijk van de keuzes die een organisatie maakt bij de implementatie van de standaard.

Interoperabiliteit is daarbij niet relevant, omdat het wijzigen of verwijderen van Logregels niet gebeurt vanuit de applicatie die oorspronkelijk de dataverwerking uitvoerde en het wegschrijven van het logrecord veroorzaakte. Wijzigen en verwijderen gebeurt vanuit een beheercomponent. Deze zijn vaak hard gekoppeld aan de voor logging gekozen oplossing, waardoor het voorschrijven van een interface tot onnodige complexiteit leidt.

§ Besluit

- De standaard beschrijft geen interface voor het wijzigen of verwijderen van Logregels

§ Gevolgen

- Iedere organisatie kan een bij de eigen implementatie passende oplossing kiezen voor het verwijderen van Logregels
- Het wijzigen van Logregels is in principe ongewenst maar kan op soortgelijke manier opgelost worden

§ Vertrouwelijkheid wordt vastgelegd per Verwerkingsactiviteit

Dit onderdeel is niet normatief.

- Status: draft
- Datum: 2024-05-23

§ Context en probleemstelling

Alle verwerkingen worden gelogd. Een deel van deze verwerkingen mag bekend worden bij Betrokkenen, een deel niet. Hoe moet dit onderscheid geïmplementeerd worden?

Voorbeeld:

- Opsporingsinstantie A bevraagt bij Overheidsorgaan B data op over Betrokkene X in het kader van opsporingsactiviteiten rond een misdrijf
- Betrokkene krijgt geen inzage in / wordt niet geïnformeerd over de verwerking van Opsporingsinstantie A, dit zou het onderzoek hinderen
- Als Betrokkene wel inzage krijgt / wordt geïnformeerd over de verwerking van Overheidsorgaan B, zou Betrokkene alsnog zien dat Opsporingsinstantie A deze data heeft opgevraagd, met hetzelfde ongewenste effect.

Er zijn meerdere oplossingen denkbaar. Wat is de gewenste oplossingsrichting, hoe wordt deze gespecificeerd?

Mogelijke oplossingsrichtingen:

1. Ken aan iedere Dataverwerking een status toe waarmee de vertrouwelijkheid wordt aangeduid, en geef deze status mee in de verwerking zodat alle betrokken organisaties dit in de logs kunnen verwerken

2. Leg vertrouwelijkheid meer categorisch vast op het niveau van Verwerkingsactiviteiten (in het Register)

Overwegingen:

Vertrouwelijke verwerkingen moeten meer strikt gescheiden moeten worden van niet-vertrouwelijke verwerkingen. Als een bevraging zowel vertrouwelijk als niet-vertrouwelijk kan zijn, zoals bij het opvragen van eigenaarddata van een voertuig, moeten hiervoor twee gescheiden processen bestaan. De vertrouwelijke variant moet apart worden gelogd en aan strengere regels voldoen. Dit omvat bijvoorbeeld eisen aan betrokken beheerders, de classificatie van data en andere specifieke voorschriften.

Pogingen om het geschetste probleem op te lossen door op logrecord-niveau vast te leggen of een verwerking vertrouwelijk is leiden tot veel complexiteit en uitzonderingsgevallen in de implementatie van de standaard. Een aantal voorbeelden van ongewenste complexiteit:

- Vertrouwelijkheid vastleggen per logrecord betekent dat deze vertrouwelijkheid ook moet kunnen worden opgeheven
- Logregels zijn dan niet langer 'immutable' tenzij ingewikkelde constructies worden gekozen waarbij een logrecord logisch wordt vervangen door een nieuw record toe te voegen
- Er zou een interface gedefinieerd moeten worden voor het wijzigen van de status 'vertrouwelijkheid'
- Vertrouwelijkheid van een handeling aan het einde van een proces zou gevolgen kunnen hebben voor reeds vastgelegde Logregels

Bovendien geldt dat Overheidsorganisatie B op impliciete wijze zou leren dat Betrokkene X onderwerp is van een opsporingsonderzoek, terwijl dit beter op expliciete wijze geregeld kan worden. Door het expliciet te regelen kan Overheidsorganisatie B alle benodigde maatregelen nemen om te zorgen dat de vertrouwelijkheid ook in die organisatie geborgd is.

§ Besluit

Vertrouwelijkheid wordt vastgelegd per Verwerkingsactiviteit

§ Gevolgen

- Vertrouwelijkheid wordt niet vastgelegd in Logregels
- Vertrouwelijkheid wordt per logrecord afgeleid uit wat over vertrouwelijkheid is vastgelegd bij de bijbehorende Verwerkingsactiviteit

- Vertrouwelijkheid wordt NIET uitgewisseld tussen organisaties
- Wanneer een verwerking niet langer vertrouwelijk is, bijvoorbeeld na verjaring, dan volgt dit uit data die vastliggen in het Register (bijvoorbeeld status vertrouwelijkheid, duur vertrouwelijkheid) en wat vastligt in een logrecord (verwerkingsactiviteit_id en datum)
- Organisaties moeten vooraf borgen dat vertrouwelijke Dataverwerkingen worden uitgevoerd op een manier die verantwoord kan worden, door dit te regelen op het niveau van Verwerkingsactiviteit. Dit kan tot gevolg hebben dat twee aparte processen nodig zijn voor het vertrouwelijk en niet-vertrouwelijk opvragen van gegevens.

§ Verwijzingen naar Registers zijn zo los mogelijk

Dit onderdeel is niet normatief.

- Status: proposed
- Datum: 2024-05-24

§ Context en probleemstelling

In de Logregels staat zo min mogelijk inhoudelijke informatie. Informatie over verwerkingsactiviteiten ligt vast in specifieke registers.

- Er kunnen meerdere van deze Registers zijn
- Deze kunnen ook van andere organisaties zijn
- Naar welk Register wordt verwezen is afhankelijk van het type dataverwerking. Verwerkingen in het kader van de Algemene Verordening Gegevensbescherming (AVG) verwijzen naar een Register van Verwerkingsactiviteiten zoals beschreven in AVG art. 30.
- Het Register van Verwerkingsactiviteiten (RvVA) is voor veel organisaties verplicht vanuit AVG art. 30, echter niet voor alle organisaties
- Als een Register bestaat, betekent dit niet dat het ook ontsloten wordt met een API. In de huidige praktijk bestaat het vaak alleen in een statisch document.

De standaard voor logging moet functioneren gegeven bovenstaande feiten.

§ Besluit

De link naar de uitwerking van een verwerkingsactiviteit bestaat uit een *identifier* en daarnaast een URI, URL of URN, in de vorm van *key value pairs*. Eventuele nadere invulling voor het verwijzen naar specifieke Registers (zoals het RvVA) wordt uitgewerkt in extensies.

§ Gevolgen

Door de verwijzingen naar de registers los te houden van de Registers wordt voorkomen dat er in de logs directe afhankelijkheden ontstaan van de registers.

§ Log Sampling is niet toegestaan

Dit onderdeel is niet normatief.

- Status: proposed
- Datum: 2024-06-20

§ Context en probleemstelling

Een bij logging veelgebruikte techniek is het zogenaamde 'Log Samplen', waarbij bijvoorbeeld slechts 1 op de 10 of 1 op de 100 acties die een log zouden veroorzaken daadwerkelijk worden weggeschreven. Dit wordt gedaan uit overwegingen van performance, opslagruimte en/of kosten. Voor veel toepassingen is het voldoende om uit deze logs trends te destilleren om zo fouten op te sporen of voorstellen voor verbetering te kunnen doen.

Wanneer dit zou worden toegepast bij onderhanden standaard, zou kunnen worden betoogd dat verantwoording nog altijd slaagt, omdat data voor een relevante, gerandomiseerde steekproef beschikbaar is. Echter, gelet op het belang van de verantwoording, en de wettelijke verplichtingen waaraan met de standaard invulling wordt gegeven, is dit onwenselijk voor het Logboek Dataverwerkingen. De Logregels vormen o.a. de basis voor de Informatieplicht en het Inzagerecht uit de AVG. Daarvoor is het nodig om over iedere Dataverwerking metadata vast te leggen.

§ Besluit

Log Sampling is niet toegestaan.

§ Gevolgen

- Iedere logregel wordt weggeschreven in het LogBoek Dataverwerkingen
- Wanneer een techniek voor loggen wordt toegepast waarbij Log Sampling is ingericht, moet ervoor worden gewaakt dat dit niet geldt voor de logregels die beschreven worden in deze standaard.

§ Pseudonimiseren data

Dit onderdeel is niet normatief.

- Dit onderdeel is niet normatief.
- Status: final
- Datum: 2024-12-04

§ Context

Bij vastleggen betrokkene wordt in veel gevallen een BSN vastgelegd, of een ander nummer bijvoorbeeld vreemdelingennummer of objectnummer. Wanneer we inhoud van verwerkte data-objecten gaan vastleggen dan betreft dat ook dit soort data. Uit wet en regelgeving (bijvoorbeeld AVG) volgt dat pseudonimiseren wenselijk is.

§ Vraag:

Moeten we in de standaard beschrijven hoe men pseudonimisering moet aanpakken?

§ Overwegingen:

- Pseudonimisering en het weer 'ont-pseudonimiseren' is iets wat binnen de organisatie relevant is, om onder andere de inzage-API te kunnen laten functioneren.
- Het is niet nodig om pseudoniemen buiten de grenzen van de organisatie uit te wisselen.
- Als burger Inzage-API's gebruikt, bestaat dat uit een reeks achtereenvolgende calls met organisaties waaruit client-side wordt afgeleid hoe het een en ander zich tot elkaar verhoudt.
- Door organisaties vrij te laten in hun implementatie, is het eenvoudiger om de standaard te implementeren.
- Pseudonimiseren is ook relevant buiten deze standaard, daarom kunnen we niet in deze standaard definiëren hoe het geïmplementeerd moet worden.

§ Besluit

- Juridisch is het pseudonimiseren van data niet VERPLICHT. Er is een verplichting om persoonsdata adequaat te beveiligen. Pseudonimisering is een mogelijke maatregel. We kiezen daarom voor een AANBEVELING in plaats van een VERPLICHTING.
- In de standaard zetten we de aanbeveling om te standaardiseren, waarbij de implementatie aan de organisatie is ("Het is AANBEVOLEN om ...").

§ `foreign_trace_id` wordt niet meer opgeslagen

Dit onderdeel is niet normatief.

- Status: draft
- Datum: 2025-08-15

§ Context en probleemstelling

Tijdens de workshop casus met RINIS is gebleken dat het veld `foreign_trace_id` overbodig is en geen praktische toepassing kent binnen de standaard. Dit inzicht heeft geleid tot heroverweging van de noodzaak ervan.

§ Besluit

- `foreign_trace_id` wordt **niet** meer opgeslagen

§ Gevolgen

- De aangeroepen organisatie (verwerker) krijg als `trace_id` de `trace_id` van de aanroepende organisatie (verantwoordelijke)
- `foreign_span_id` wordt wel opgeslagen
- De aangeroepen organisatie (verwerker) krijgt een eigen `span_id` per activiteit

§ 5. Voorbeelden

In dit hoofdstuk worden er voorbeelden beschreven van hoe de standaard gebruikt kan worden in verschillende scenario's om de lezer een beter beeld te geven van de standaard in zijn echte werking. Hierbij zijn er vier voorbeelden met elk een schets van de situatie, de uitgangspunten, het globale proces, de relatie tussen data, de relatie met het Logboek Dataverwerkingen en het gedrag van de applicatie.

§ Parkeervergunning - inzien

§ Situatieschets (Parkeervergunning - inzien)

Een persoon heeft bij een gemeente een parkeervergunning in gebruik en wil de data van deze vergunning bekijken.

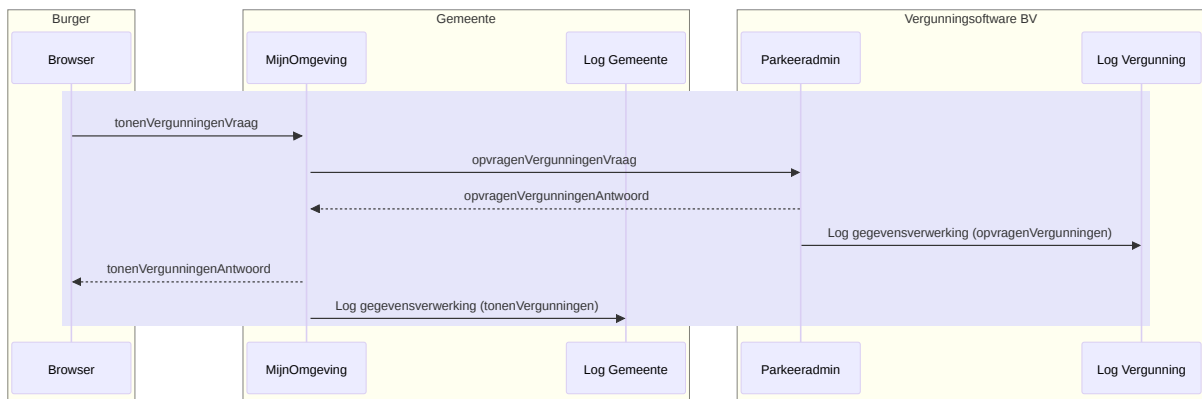
§ Uitgangspunten (Parkeervergunning - inzien)

- Het beschreven proces is een voorbeeld, het werkelijke proces kan anders verlopen.
- Het proces is een 'happy flow', dit betekent dat validaties en eventuele foutsituaties in dit voorbeeld niet in ogenschouw worden genomen.
- Autorisatieprocessen zijn in dit voorbeeld niet meegenomen.
- Een Loggingsregel wordt toegevoegd aan het logboek per **geheel** afgeronde transactie. Er wordt dus **geen** aparte logregel aangemaakt per ontvangen of verstuurd bericht.

§ Globaal proces (Parkeervergunning - inzien)

1. Een persoon vraagt in zijn 'MijnOmgeving' van de gemeente om de bestaande parkeervergunningdata.
2. De 'MijnOmgeving' van de gemeente verzoekt de parkeervergunningapplicatie om de actuele parkeervergunningdata van de persoon.
3. Het parkeervergunningsysteem voert dit verzoek uit. Daarna verzendt de parkeervergunningapplicatie de gevraagde data naar de gemeente. Het parkeervergunningensysteem logt dat er data verzonden is naar de gemeente.
4. De gemeente toont de data aan de persoon en logt dat deze data is getoond aan de persoon.

Schematisch ziet dit proces er als volgt uit:



§ Logging van data (Parkeervergunning - inzien)

De volgende data worden gelogd in de diverse logmomenten:

- **Log opvragenVergunningen (log vergunningenapplicatie)**

| Attribuut | Waarde |
|---------------------------|--|
| span_id | 8451dcd9ede037cb |
| name | opvragenVergunningen |
| parent_span_id | <leeg> |
| trace_id | ccf5064a324163ed939bfa09c2bcb210 |
| start_time | 2024-05-30 08:40:37.000 |
| end_time | 2024-05-30 08:40:37.000 |
| status_code | OK |
| resource.name | Parkeeradmin |
| resource.version | 2.1.6 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | rva:12f2ec2a-0cc4-3541-9ae6-219a178fcfe4 |
| attributeKey | <leeg> |
| attributeValue | <leeg> |
| foreign_operation.span_id | 9f8971bfd093637d |

- **Log opvragenVergunningen (log gemeente)**

| Attribuut | Waarde |
|-----------|----------------------------------|
| span_id | ccf5064a324163ed939bfa09c2bcb210 |

| Attribuut | Waarde |
|------------------|--|
| name | tonenVergunningen |
| parent_span_id | <leeg> |
| trace_id | c7a26dcd0bee0c8900e2174c43c3393c |
| start_time | 2024-05-30 10:40:37.821 |
| end_time | 2024-05-30 10:40:37.845 |
| status_code | OK |
| resource.name | MijnOmgeving |
| resource.version | 1.0.5 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | rva:11x2ec2a-0774-3541-9b16-21ba179fcf15 |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | rva:13j2ec27-0cc4-3541-9av6-219a178fcfe5 |

§ Relatie tussen data (Parkeervergunning - inzien)

Om uiteindelijk alle data te kunnen rapporteren, is het van belang dat data op een bepaalde manier aan elkaar gekoppeld is. In dit voorbeeld is de data op de volgende manier gekoppeld:



§ Relatie met de standaard Logboek dataverwerkingen (Parkeervergunning - inzien)

De relatie met de doelstellingen die gesteld zijn in de standaard Logboek dataverwerkingen worden, op basis van dit voorbeeld, als volgt concreet gerealiseerd:

- **het wegschrijven van logs van dataverwerkingen:** In dit voorbeeld is het de betrokkene zelf die via een portaal zijn eigen data kan bekijken. Deze actie is een dataverwerking en wordt gelogd bij zowel de gemeenteapplicatie (data wordt getoond aan de betrokkene) als bij de vergunningenapplicatie (verstrekking specifieke informatie aan de gemeenteapplicatie).
- **het aan elkaar relateren van logs van dataverwerkingen:** Er zijn in dit voorbeeld twee applicaties nodig om het totaal aan gevraagde informatie te kunnen tonen aan de betrokkene. Beide applicaties hebben een logboek voor verwerkte data. Om een totaalbeeld van de gelogde data te kunnen construeren, is een relatie tussen de logs nodig. In dit voorbeeld wordt de koppeling gelegd door het `span_id` en `trace_id` uit het gemeentelogboek te linken aan het `foreign_operation.span_id` en `trace_id` uit het vergunningenlogboek.
- **het aan elkaar relateren van dataverwerkingen over de grenzen van systemen:** Naast het koppelen van logs van diverse applicaties, wordt ook een koppeling gelegd met het Register van verwerkingsactiviteiten. Dit gebeurt per applicatie op basis van het `processing_activity_id` (register) te koppelen aan `dpl.core.processing_activity_id` (logboek). De diverse registers hebben **geen** directe koppeling met elkaar.

Standaard Logverwerkingen: **paragraaf 3.3.1 Gedrag**

1. *De applicatie MOET een Trace starten voor iedere Dataverwerking waarvan nog geen Trace bekend is.* Bij elke start van een verwerking wordt een `trace_id` aangemaakt. Bijvoorbeeld: in het voorbeeld komt er een bericht binnen bij de 'MijnOmgeving' van de gemeente (opvragenVergunningenVraag). Er wordt direct een `trace_id` aangemaakt.
2. *De applicatie MOET voor iedere Dataverwerking een logregel wegschrijven in een Logboek. Log Sampling is niet toegestaan.* Een dataverwerking wordt opgeslagen als deze volledig is afgerond. In het voorbeeld is te zien dat een logregel wordt geschreven op het moment dat de vraag- en het antwoordbericht zijn afgerond.
3. *De applicatie MOET bijhouden of een Dataverwerking geslaagd of mislukt is en dit per Dataverwerking als status meegeven aan het Logboek.* Bij elke logregel in het voorbeeld staat de `status_code` vermeld ('OK').
4. *Als een Dataverwerking meerdere Betrokkenen heeft dan MOET de applicatie voor iedere betrokkene een aparte logregel wegschrijven. Een logregel kan naar 0 of 1 betrokkenen verwijzen.* In het voorbeeld gaat het om één betrokkene (`dpl.core.data_subject_id`), er wordt steeds één logregel aangemaakt.
5. *Als een applicatie aangeroepen kan worden vanuit een andere applicatie MOET de applicatie Trace Context metadata accepteren bij een dergelijke aanroepen deze metadata kunnen omzetten naar een foreign_operation bericht.* Bij een externe verwerking (bijvoorbeeld opvragenVergunningen) geeft de 'MijnOmgeving' de `trace_id` en `span_id` mee aan de Vergunningenapplicatie. De vergunningenapplicatie registreert de `trace_id`, en `span_id` als `foreign_operation.span_id`.

§ Parkeervergunning - wijzigen

§ Situatieschets (Parkeervergunning - wijzigen)

Een persoon heeft bij een gemeente een parkeervergunning in gebruik en wil de data van het kenteken van deze vergunning wijzigen.

§ Uitgangspunten (Parkeervergunning - wijzigen)

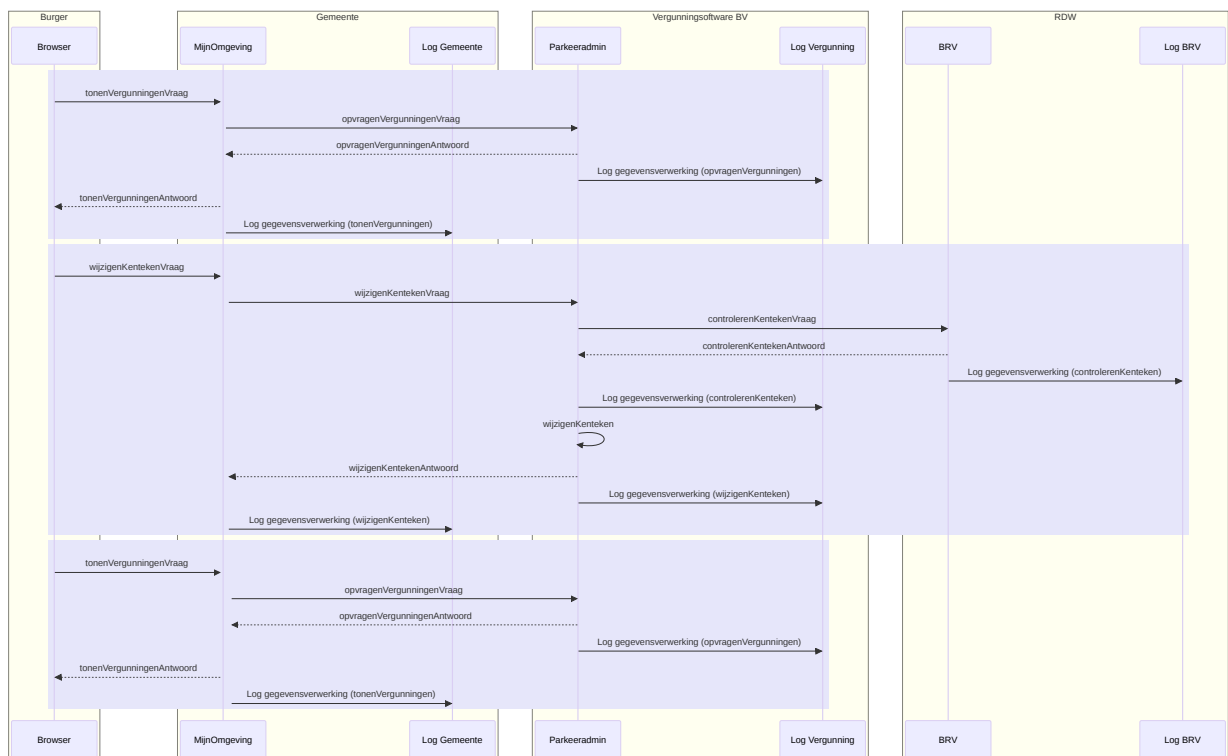
- Het beschreven proces is een voorbeeld, het werkelijke proces kan anders verlopen.
- Het proces is een 'happy flow', dit betekent dat validaties en eventuele foutsituaties in dit voorbeeld niet in ogenschouw worden genomen.
- Autorisatieprocessen zijn in dit voorbeeld niet meegenomen.
- Een Loggingsregel wordt toegevoegd aan het logboek per **geheel** afgeronde transactie. Er wordt dus **geen** aparte logregel aangemaakt per ontvangen of verstuurd bericht.

§ Globaal proces (Parkeervergunning - wijzigen)

1. Een persoon vraagt in zijn 'MijnOmgeving' van de gemeente om de bestaande parkeervergunningdata.
2. De 'MijnOmgeving' van de gemeente verzoekt de parkeervergunningapplicatie om de actuele parkeervergunningdata van de persoon.
3. De parkeervergunningapplicatie voert dit verzoek uit. Daarna verzendt de parkeervergunningapplicatie de gevraagde data naar de gemeente. De parkeervergunningapplicatie logt dat er data verzonden is naar de gemeente.
4. De gemeente toont de data aan de persoon en logt dat deze data is getoond aan de persoon.
5. De persoon wijzigt het kenteken in de 'MijnOmgeving' van de gemeente.
6. De 'MijnOmgeving' van de gemeente verzoekt de parkeervergunningapplicatie om de wijziging af te handelen.
7. De parkeervergunningapplicatie verzoekt het RDW te controleren of het kenteken ook daadwerkelijk bij de persoon hoort.

8. Het RDW stuurt een antwoord terug naar de parkeervergunningapplicatie en logt de dataverwerking.
9. De parkeervergunningapplicatie logt het controleverzoek aan het RDW.
10. De parkeervergunningapplicatie wijzigt het kenteken van de persoon en logt het wijzigingsverzoek van de persoon.
11. Nadat de wijziging is gedaan in de parkeervergunningapplicatie, wordt het wijzigingsverzoek gelogd in de 'MijnOmgeving' van de gemeente.
12. De persoon vraagt in zijn 'MijnOmgeving' van de gemeente om de bestaande parkeervergunningdata.
13. De 'MijnOmgeving' van de gemeente verzoekt de parkeervergunningapplicatie om de actuele parkeervergunningdata van de persoon.
14. De parkeervergunningapplicatie voert dit verzoek uit. Daarna verzendt de parkeervergunningapplicatie de gevraagde data naar de gemeente. De parkeervergunningapplicatie logt dat er data verzonden zijn naar de gemeente.
15. De gemeente toont de data aan de persoon en logt dat deze data is getoond aan de persoon.

Schematisch ziet dit proces er als volgt uit:



§ Logging van data (Parkeervergunning - wijzigen)

De volgende data worden gelogd in de diverse logmomenten:

- **Log opvragenVergunningen (log vergunningenapplicatie)**

| Attribuut | Waarde |
|---------------------------|--------------------------------------|
| span_id | 8ee7b01aca8d01d9 |
| name | opvragenVergunningen |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:16:49.000 |
| end_time | 2024-07-29 08:16:49.000 |
| status_code | OK |
| resource.name | Parkeeradmin |
| resource.version | 2.1.6 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 12f2ec2a-0cc4-3541-9ae6-219a178fcfe4 |
| attributeKey | <leeg> |
| attributeValue | <leeg> |
| foreign_operation.span_id | b2e339a595246e01 |

- **Log tonenVergunningen (log gemeente)**

| Attribuut | Waarde |
|------------------|--------------------------------------|
| span_id | b2e339a595246e01 |
| name | tonenVergunningen |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 10:16:49.690 |
| end_time | 2024-07-29 10:16:49.723 |
| status_code | OK |
| resource.name | MijnOmgeving |
| resource.version | 1.0.5 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 11x2ec2a-0774-3541-9b16-21ba179fcf15 |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | 13j2ec27-0cc4-3541-9av6-219a178fcfe5 |

- **Log controlerenKenteken (log RDW)**

| Attribuut | Waarde |
|-----------------------------------|--------------------------------------|
| span_id | 433f276975204ccf |
| name | controlerenKenteken |
| parent_span_idcontrolerenKenteken | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:17:02 |
| end_time | 2024-07-29 08:17:02 |
| status_code | OK |
| resource.name | BRV |
| resource.version | 2.0 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 8c714e4a-a538-36f7-8b1f-37a6884cc68c |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | <leeg> |
| foreign_operation.span_id | 414514cf1d40d6b2 |

- **Log controlerenKenteken (log vergunningenapplicatie)**

| Attribuut | Waarde |
|------------------|--------------------------------------|
| span_id | 414514cf1d40d6b2 |
| name | controlerenKenteken |
| parent_span_id | 7a95b6989d2b28c7 |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:17:02.000 |
| end_time | 2024-07-29 08:17:02.000 |
| status_code | OK |
| resource.name | Parkeeradmin |
| resource.version | 2.1.6 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 19u2dd2a-0cb7-3541-9ae6-217a178fc9e6 |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | <leeg> |

| Attribuut | Waarde |
|---------------------------|------------------|
| foreign_operation.span_id | ba7cac7ca0489e42 |

- **Log wijzigenKenteken (log vergunningenapplicatie)**

| Attribuut | Waarde |
|---------------------------|--------------------------------------|
| span_id | 7a95b6989d2b28c7 |
| name | wijzigenKenteken |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:17:02.000 |
| end_time | 2024-07-29 08:17:02.000 |
| status_code | OK |
| resource.name | Parkeeradmin |
| resource.version | 2.1.6 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 0b1ff20a-3ecb-34bf-8cf5-e4cbacb046ab |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | <leeg> |
| foreign_operation.span_id | df524ee2a3fd5ddf |

- **Log wijzigenKenteken (log gemeente)**

| Attribuut | Waarde |
|------------------|--------------------------------------|
| span_id | df524ee2a3fd5ddf |
| name | wijzigenKenteken |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 10:17:02.010 |
| end_time | 2024-07-29 10:17:02.039 |
| status_code | OK |
| resource.name | MijnOmgeving |
| resource.version | 1.0.5 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 12c21c2a-0875-3543-9b16-21ja179fcf16 |

| Attribuut | Waarde |
|---------------------------|--------------------------------------|
| attributeKey | dpl.core.data_subject_id |
| attributeValue | 13j2ec27-0cc4-3541-9av6-219a178fcfe5 |
| foreign_operation.span_id | <leeg> |

- **Log opvragenVergunningen (log vergunningenapplicatie)**

| Attribuut | Waarde |
|---------------------------|--------------------------------------|
| span_id | 6042d706f53fec76 |
| name | opvragenVergunningen |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:17:02.000 |
| end_time | 2024-07-29 08:17:02.000 |
| status_code | OK |
| resource.name | Parkeeradmin |
| resource.version | 2.1.6 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 12f2ec2a-0cc4-3541-9ae6-219a178fcfe4 |
| attributeKey | <leeg> |
| attributeValue | <leeg> |
| foreign_operation.span_id | ba7cac7ca0489e42 |

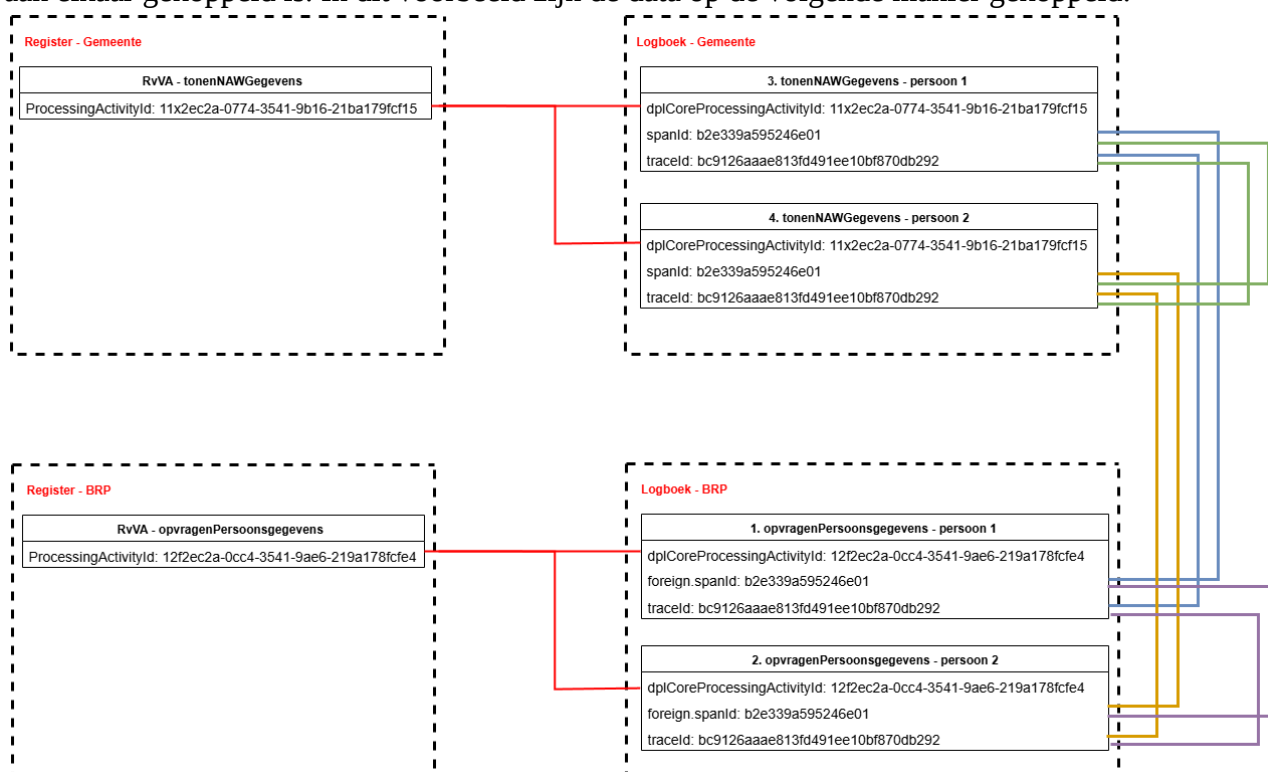
- **Log tonenVergunningen (log gemeente)**

| Attribuut | Waarde |
|------------------|---------------------------------|
| span_id | ba7cac7ca0489e42 |
| name | tonenVergunningen |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 10:17:02.274 |
| end_time | 2024-07-29 10:17:02.291 |
| status_code | OK |
| resource.name | MijnOmgeving |
| resource.version | 1.0.5 |

| Attribuut | Waarde |
|----------------|--------------------------------------|
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 11x2ec2a-0774-3541-9b16-21ba179fcf15 |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | 13j2ec27-0cc4-3541-9av6-219a178fcfe5 |

§ Relatie tussen data (Parkeervergunning - wijzigen)

Om uiteindelijk alle data te kunnen rapporteren, is het van belang dat data op een bepaalde manier aan elkaar gekoppeld is. In dit voorbeeld zijn de data op de volgende manier gekoppeld:



§ Relatie met de standaard Logboek Dataverwerkingen (Parkeervergunning - wijzigen)

De relatie met de doelstellingen die gesteld zijn in de standaard Logboek dataverwerkingen worden, op basis van dit voorbeeld, als volgt concreet gerealiseerd:

- **het wegschrijven van logs van dataverwerkingen:** In dit voorbeeld is het de betrokkene zelf die via een portaal zijn eigen data kan bekijken en wijzigen. Deze acties zijn dataverwerkingen en worden gelogd bij zowel de gemeenteapplicatie (data wordt getoond aan de betrokkene) als bij de vergunningenapplicatie (verstrekking specifieke informatie aan de gemeenteapplicatie).

- **het aan elkaar relateren van logs van dataverwerkingen:** Er zijn in dit voorbeeld twee applicaties nodig om het totaal aan gevraagde informatie te kunnen tonen aan de betrokkene. Beide applicaties hebben een logboek voor verwerkte data. Om een totaalbeeld van de gelogde data te kunnen construeren, is een relatie tussen de logs nodig. In dit voorbeeld wordt de koppeling gelegd door het `span_id` en `trace_id` (gemeentelogboek) te linken aan het `foreign_operation.span_id` en `trace_id` (vergunningenlogboek).
- **het aan elkaar relateren van dataverwerkingen over de grenzen van systemen:** Naast het koppelen van logs van diverse applicaties, wordt ook een koppeling gelegd met het Register van verwerkingsactiviteiten. Dit gebeurt per applicatie op basis van het `processing_activity_id` (register) te koppelen aan `dpl.core.processing_activity_id` (logboek). De diverse registers hebben **geen** directe koppeling met elkaar.

§ Standaard Logverwerkingen paragraaf 3.3.1 Gedrag

1. *De applicatie MOET een Trace starten voor iedere Dataverwerking waarvan nog geen Trace bekend is.* Bij elke start van een verwerking wordt een `trace_id` aangemaakt. Bijvoorbeeld: in het voorbeeld komt er een bericht binnen bij de 'MijnOmgeving' van de gemeente (opvragenVergunningenVraag). Er wordt direct een `trace_id` aangemaakt.
2. *De applicatie MOET voor iedere Dataverwerking een logregel wegschrijven in een Logboek. Log Sampling is niet toegestaan. Een dataverwerking wordt opgeslagen als deze volledig is afgerond.* In het voorbeeld is te zien dat een logregel wordt geschreven op het moment dat de vraag- en het antwoordbericht zijn afgerond.
3. *De applicatie MOET bijhouden of een Dataverwerking geslaagd of mislukt is en dit per Dataverwerking als status meegeven aan het Logboek.* Bij elke logregel in het voorbeeld staat de `status_code` vermeld ('OK').
4. *Als een Dataverwerking meerdere Betrokkenen heeft dan MOET de applicatie voor iedere betrokkene een aparte logregel wegschrijven. Een logregel kan naar 0 of 1 betrokkenen verwijzen.* In het voorbeeld gaat het om één betrokkene (`dpl.core.data_subject_id`), er wordt steeds één logregel aangemaakt.
5. *Als een applicatie aangeroepen kan worden vanuit een andere applicatie MOET de applicatie Trace Context metadata accepteren bij een dergelijke aanroepen deze metadata kunnen omzetten naar een foreign_operation bericht.* Bij een externe verwerking (bijvoorbeeld opvragenVergunningen) geeft de 'MijnOmgeving' de `trace_id` en `span_id` mee aan de Vergunningenapplicatie. De vergunningenapplicatie registreert de `trace_id`, en `span_id` als `foreign_operation.span_id`.

§ Registratie Verhuizing - Eenvoudig, traditioneel systeem

§ Situatieschets (Registratie Verhuizing - Eenvoudig)

Deze case beschrijft de binnengemeentelijke verhuizing van een persoon. De beschrijving is functioneel zo eenvoudig mogelijk. De burger komt aan de balie en er is geen sprake van meeverhuizende gezinsleden.

§ Uitgangspunten (Registratie Verhuizing - Eenvoudig)

- Het beschreven proces is een voorbeeld, het werkelijke proces kan anders verlopen.
- Het proces is een 'happy flow', dit betekent dat validaties en eventuele foutsituaties in dit voorbeeld niet in ogenschouw worden genomen.
- Autorisatieprocessen zijn in dit voorbeeld niet meegenomen.
- Een Loggingsregel wordt toegevoegd aan het logboek per **geheel** afgeronde transactie. Er wordt dus **geen** aparte logregel aangemaakt per ontvangen of verstuurd bericht.

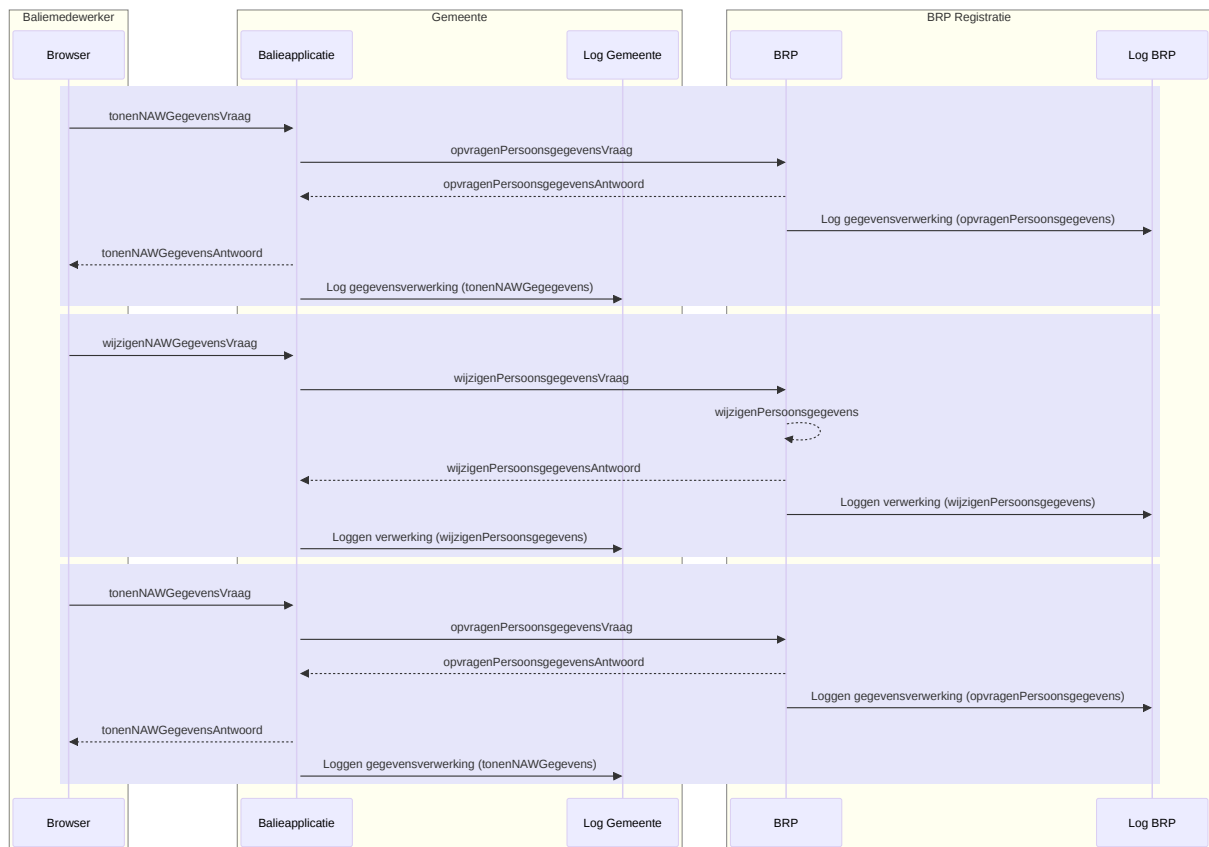
§ Globaal proces (Registratie Verhuizing - Eenvoudig)

Schematisch ziet dit proces er als volgt uit:

1. De Baliemedewerker voert BSN van de burger in.
2. De Browser vraagt om persoonsdata bij de gemeentelijke Balieapplicatie.
3. De gemeentelijke Balieapplicatie vraagt persoonsdata bij het BRP-systeem.
4. Het BRP systeem stuurt gevraagde data naar de gemeentelijke Balieapplicatie en logt de aanvraag.
5. De gemeentelijke Balieapplicatie stuurt de data naar de Browser en worden getoond aan de Baliemedewerker. De aanvraag wordt gelogd door de Balieapplicatie.
6. De Baliemedewerker voert de wijziging in en de Browser verstuurt de data naar de gemeentelijke Balieapplicatie.
7. De gemeentelijke Balieapplicatie verstuurt de data naar het BRP-systeem.

8. Het BRP-systeem verwerkt de wijziging, stuurt bevestiging terug naar de gemeentelijke Balieapplicatie en logt de verwerkingsactie.
9. De Browser vraagt de actuele persoonsdata op de gemeentelijke Balieapplicatie.
10. De gemeentelijke Balieapplicatie vraagt de persoonsdata op bij het BRP-systeem.
11. Het BRP-systeem stuurt de persoonsdata naar de gemeentelijke Balieapplicatie en logt de aanvraag.
12. De gemeentelijke Balieapplicatie stuurt de persoonsdata naar de Browser en logt de aanvraag.

Schematisch ziet dit proces er als volgt uit:



§ Logging van data (Registratie Verhuizing - Eenvoudig)

De volgende data worden gelogd in de diverse logmomenten:

- **Log opvragenPersoonsgegevens (log BRP)**

| Attribuut | Waarde |
|-----------|--------------------------------------|
| span_id | 7a22eb38-bca6-463f-9955-54ab040287cb |
| name | opvragenPersoonsgegevens |

| Attribuut | Waarde |
|---------------------------|--------------------------------------|
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:16:49.000 |
| end_time | 2024-07-29 08:16:49.000 |
| status_code | OK |
| resource.name | BRP |
| resource.version | 2.0 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 12f2ec2a-0cc4-3541-9ae6-219a178fcfe4 |
| attributeKey | <leeg> |
| attributeValue | <leeg> |
| foreign_operation.span_id | b2e339a595246e01 |

- **Log tonenNAWGegevens (log gemeente)**

| Attribuut | Waarde |
|------------------|--------------------------------------|
| span_id | b2e339a595246e01 |
| name | tonenNAWGegevens |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 10:16:49.690 |
| end_time | 2024-07-29 10:16:49.723 |
| status_code | OK |
| resource.name | Balieapp |
| resource.version | 1.0.5 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 11x2ec2a-0774-3541-9b16-21ba179fcf15 |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | 13j2ec27-0cc4-3541-9av6-219a178fcfe5 |

- **Log wijzigenPersoonsgegevens (log BRP)**

| Attribuut | Waarde |
|-----------|------------------|
| span_id | 433f276975204ccf |

| Attribuut | Waarde |
|---------------------------|--------------------------------------|
| name | wijzigenPersoonsgegevens |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:17:02 |
| end_time | 2024-07-29 08:17:02 |
| status_code | OK |
| resource.name | BRP |
| resource.version | 2.0 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 8c714e4a-a538-36f7-8b1f-37a6884cc68c |
| attributeKey | <leeg> |
| attributeValue | <leeg> |
| foreign_operation.span_id | 414514cf1d40d6b2 |

- **Log wijzigenPersoonsgegevens (log gemeente)**

| Attribuut | Waarde |
|---------------------------|--------------------------------------|
| span_id | 414514cf1d40d6b2 |
| name | wijzigenPersoonsgegevens |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:17:02.000 |
| end_time | 2024-07-29 08:17:02.000 |
| status_code | OK |
| resource.name | Balieapp |
| resource.version | 1.0.5 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 19u2dd2a-0cb7-3541-9ae6-217a178fc9e6 |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | 13j2ec27-0cc4-3541-9av6-219a178fcfe5 |
| foreign_operation.span_id | <leeg> |

- **Log opvragenPersoonsgegevens (log BRP)**

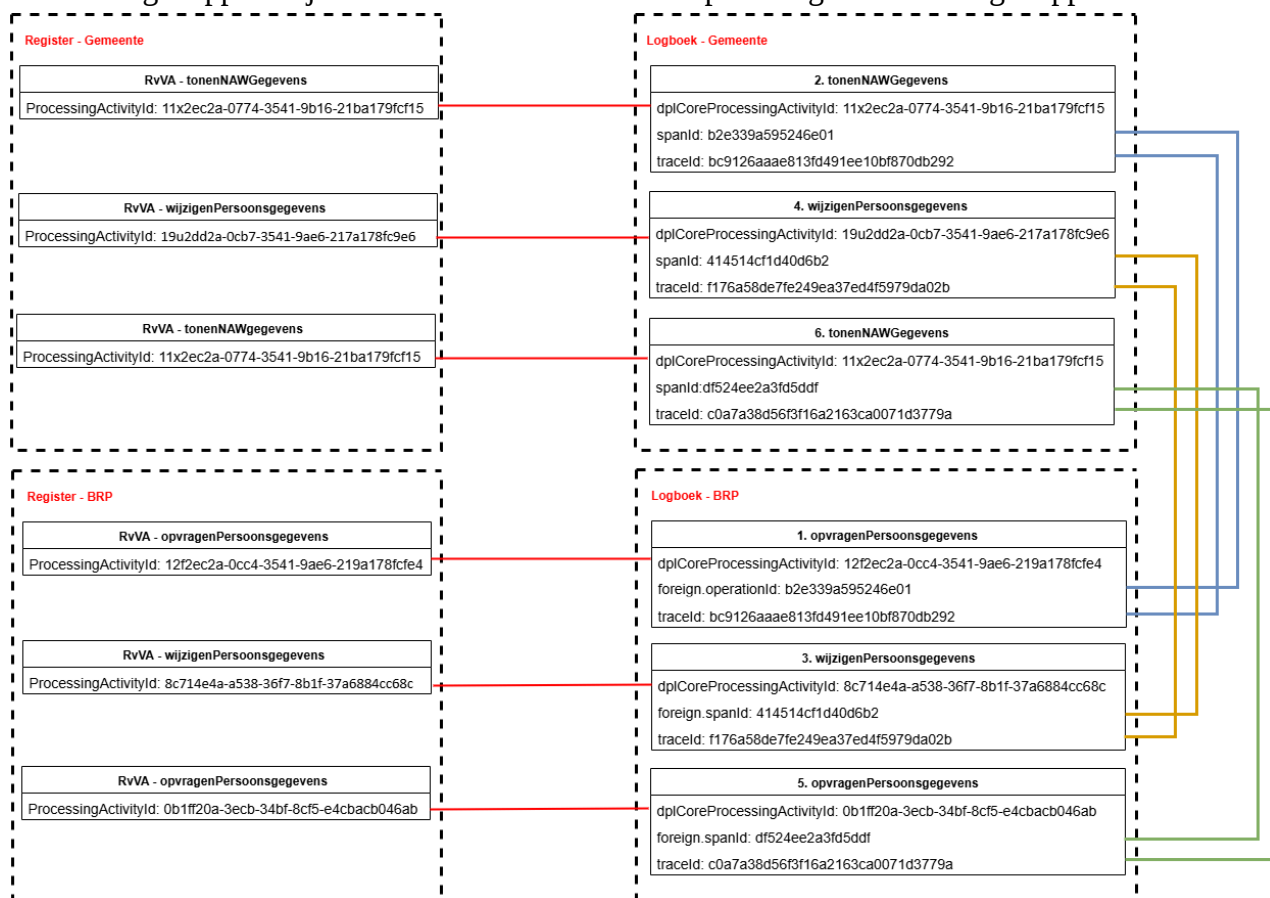
| Attribuut | Waarde |
|---------------------------|--------------------------------------|
| span_id | 7a95b6989d2b28c7 |
| name | opvragenPersoonsgegevens |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:17:02.000 |
| end_time | 2024-07-29 08:17:02.000 |
| status_code | OK |
| resource.name | BRP |
| resource.version | 2.0 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 0b1ff20a-3ecb-34bf-8cf5-e4cbacb046ab |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | <leeg> |
| foreign_operation.span_id | df524ee2a3fd5ddf |

- **Log tonenNAWGegevens (log gemeente)**

| Attribuut | Waarde |
|---------------------------|--------------------------------------|
| span_id | df524ee2a3fd5ddf |
| name | tonenNAWGegevens |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 10:17:02.010 |
| end_time | 2024-07-29 10:17:02.039 |
| status_code | OK |
| resource.name | Balieapp |
| resource.version | 1.0.5 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 12c21c2a-0875-3543-9b16-21ja179fcf16 |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | 13j2ec27-0cc4-3541-9av6-219a178fcfe5 |
| foreign_operation.span_id | <leeg> |

§ Relatie tussen data (Registratie Verhuizing - Eenvoudig)

Om uiteindelijk alle data te kunnen rapporteren, is het van belang dat data op een bepaalde manier aan elkaar gekoppeld zijn. In dit voorbeeld is de data op de volgende manier gekoppeld:



§ Relatie met de standaard Logboek dataverwerkingen (Registratie Verhuizing - Eenvoudig)

De relatie met de doelstellingen die gesteld zijn in de standaard Logboek dataverwerkingen worden, op basis van dit voorbeeld, als volgt concreet gerealiseerd:

- **het wegschrijven van logs van dataverwerkingen:** In dit voorbeeld is het de Baliemedewerker die via een Balieapplicatie de data van een Betrokkene kan bekijken en wijzigen. Deze acties zijn dataverwerkingen en worden gelogd bij zowel de Balieapplicatie als bij het BRP-systeem.
- **het aan elkaar relateren van logs van dataverwerkingen:** Er zijn in dit voorbeeld twee applicaties nodig om het totaal aan gevraagde informatie te kunnen tonen aan de betrokkene. Beide applicaties hebben een logboek voor verwerkte data. Om een totaalbeeld van de gelogde data te kunnen construeren, is een relatie tussen de logs nodig. In dit voorbeeld wordt de koppeling gelegd door het span_id en trace_id (gemeentelogboek) te linken aan het foreign_operation.span_id (BRP-logboek).

- **het aan elkaar relateren van dataverwerkingen over de grenzen van systemen:** Naast het koppelen van logs van diverse applicaties, wordt ook een koppeling gelegd met het Register van verwerkingsactiviteiten. Dit gebeurt per applicatie op basis van het `processing_activity_id` (register) te koppelen aan `dpl.core.processing_activity_id` (logboek). De diverse registers hebben **geen** directe koppeling met elkaar.

Standaard Logverwerkingen: **paragraaf 3.3.1 Gedrag**

1. *De applicatie MOET een Trace starten voor iedere Dataverwerking waarvan nog geen Trace bekend is.* Bij elke start van een verwerking wordt een `trace_id` aangemaakt. Bijvoorbeeld: in het voorbeeld komt er een bericht binnen bij de Balieapplicatie van de gemeente (tonenNAWGegevens). Er wordt direct een `trace_id` aangemaakt.
2. *De applicatie MOET voor iedere Dataverwerking een logregel wegschrijven in een Logboek. Log Sampling is niet toegestaan.* Een dataverwerking wordt opgeslagen als deze volledig is afgerond. In het voorbeeld is te zien dat een logregel wordt geschreven op het moment dat de vraag- en het antwoordbericht zijn afgerond.
3. *De applicatie MOET bijhouden of een Dataverwerking geslaagd of mislukt is en dit per Dataverwerking als status meegeven aan het Logboek.* Bij elke logregel in het voorbeeld staat de `status_code` vermeld ('OK').
4. *Als een Dataverwerking meerdere Betrokkenen heeft dan MOET de applicatie voor iedere betrokkene een aparte logregel wegschrijven. Een logregel kan naar 0 of 1 betrokkenen verwijzen.* In het voorbeeld gaat het om één betrokkene (`dpl.core.data_subject_id`), er wordt steeds één logregel aangemaakt.
5. *Als een applicatie aangeroepen kan worden vanuit een andere applicatie MOET de applicatie Trace Context metadata accepteren bij een dergelijke aanroepen deze metadata kunnen omzetten naar een `foreign_operation` bericht.* Bij een externe verwerking (bijvoorbeeld opvragenPersoonsgegevens) geeft de Balieapplicatie de `trace_id` en `span_id` mee aan het BRP-systeem. Het BRP-systeem registreert de `trace_id`, en `span_id` als `foreign_operation.span_id`.

§ Registratie verhuizing – Opvragen meerdere BSN's

§ Situatieschets (Registratie verhuizing)

Deze case beschrijft de samenstelling van een huishouding op een bepaald adres. De beschrijving is functioneel zo eenvoudig mogelijk, een burger komt aan de balie en er is geen sprake van wijzigingen in de huishouding.

§ Uitgangspunten (Registratie verhuizing)

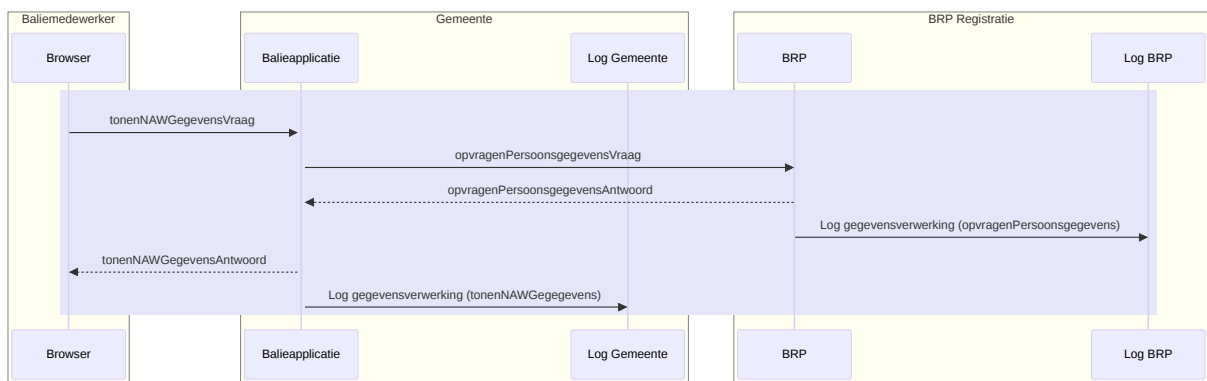
- Het beschreven proces is een voorbeeld, het werkelijke proces kan anders verlopen.
- Het proces is een 'happy flow', dit betekent dat validaties en eventuele foutsituaties in dit voorbeeld niet in ogenschouw worden genomen.
- Autorisatieprocessen zijn in dit voorbeeld niet meegenomen.
- Een Loggingsregel wordt toegevoegd aan het logboek per **geheel** afgeronde transactie. Er wordt dus **geen** aparte logregel aangemaakt per ontvangen of verstuurd bericht.
- Het is optioneel om het BSN (`dpl.core.data_subject_id`) te versleutelen ten behoeve van extra databaseveiliging. In dit voorbeeld wordt versleuteling van data toegepast.

§ Globaal proces (Registratie verhuizing)

Schematisch ziet dit proces er als volgt uit:

1. De Baliemedewerker voert adres van de burger in.
2. De Browser vraagt om persoonsdata bij de gemeentelijke Balieapplicatie.
3. De gemeentelijke Balieapplicatie vraagt persoonsdata bij het BRP-systeem.
4. Het BRP systeem stuurt gevraagde data naar de gemeentelijke Balieapplicatie en logt de aanvraag.
5. De gemeentelijke Balieapplicatie stuurt de data naar de Browser en worden getoond aan de Baliemedewerker. De aanvraag wordt gelogd door de Balieapplicatie.

Schematisch ziet dit proces er als volgt uit:



§ Logging van data (Registratie verhuizing)

De volgende data worden gelogd in de diverse logmomenten:

- **Log opvragenPersoonsgegevens (log BRP) persoon 1**

| Attribuut | Waarde |
|---------------------------|---------------------------------------|
| span_id | 7a22eb38-bca6-463f-9955-54ab040287cb |
| name | opvragenPersoonsgegevens |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:16:49.000 |
| end_time | 2024-07-29 08:16:49.000 |
| status_code | OK |
| resource.name | BRP |
| resource.version | 2.0 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 12f2ec2a-0cc4-3541-9ae6-219a178fcfe4 |
| foreign_operation.span_id | b2e339a595246e01 |
| <u>BSN 1</u> | <leeg> |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | ddj2ey299-0cf4-3541-9ar6-21ia178fcfrr |
| span_id | r2e3229059BG246e01 |
| parent_span_id | 7a22eb38-bca6-463f-9955-54ab040287cb |

- **Log opvragenPersoonsgegevens (log BRP) persoon 2**

| Attribuut | Waarde |
|----------------|--------------------------------------|
| span_id | 7a22eb38-bca6-463f-9955-54ab040287cb |
| name | opvragenPersoonsgegevens |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 08:16:49.000 |
| end_time | 2024-07-29 08:16:49.000 |
| status_code | OK |

| Attribuut | Waarde |
|---------------------------|---|
| resource.name | BRP |
| resource.version | 2.0 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 12f2ec2a-0cc4-3541-9ae6-219a178fcfe4 |
| foreign_operation.span_id | b2e339a595246e01 |
| <u>BSN 2</u> | <leeg> |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | f4j2ey299-3er4-3aa41-9ar6-21ia178fc3tyy |
| span_id | 9as5y3t-3ca7-463f-wwt9a5-54ab0402rft |
| parent_span_id | 7a22eb38-bca6-463f-9955-54ab040287cb |

- **Log tonenNAWGegevens (log gemeente) persoon 1**

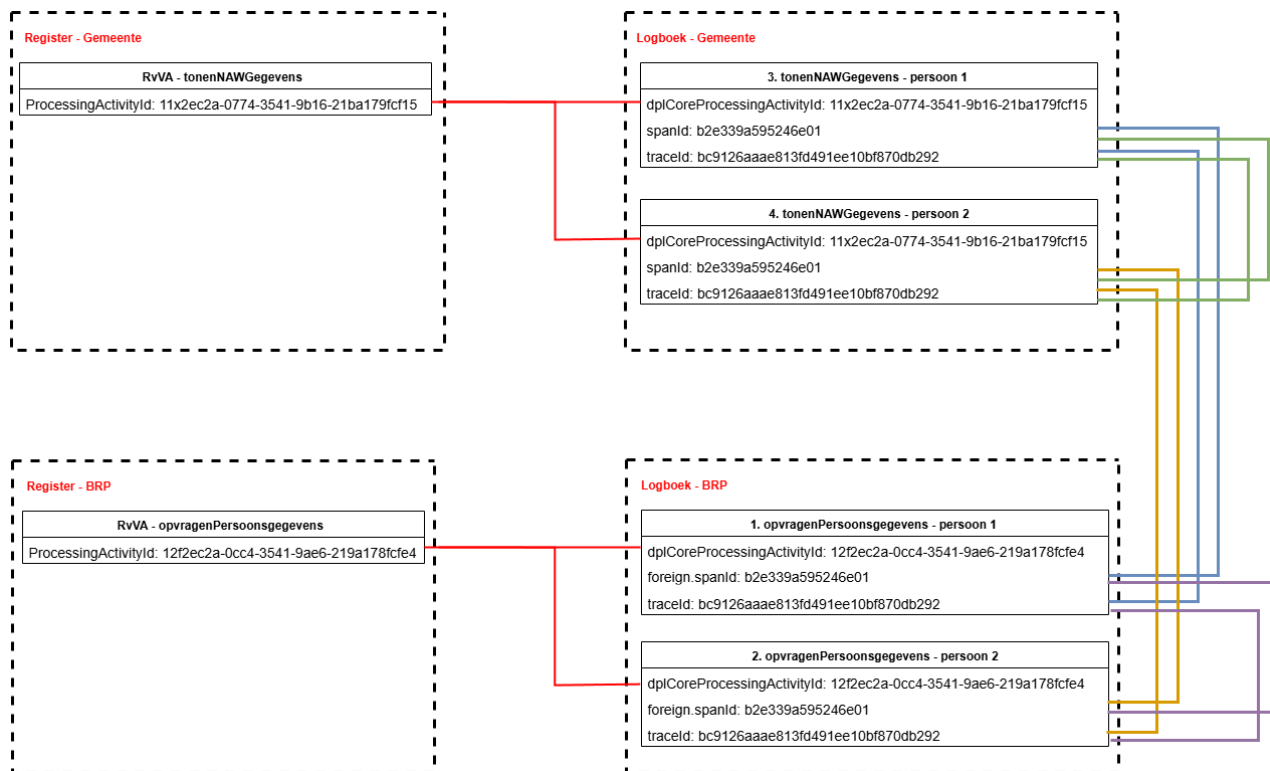
| Attribuut | Waarde |
|------------------|--------------------------------------|
| span_id | b2e339a595246e01 |
| name | tonenNAWGegevens |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 10:16:49.690 |
| end_time | 2024-07-29 10:16:49.723 |
| status_code | OK |
| resource.name | Balieapp |
| resource.version | 1.0.5 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 11x2ec2a-0774-3541-9b16-21ba179fcf15 |
| <u>BSN 1</u> | <leeg> |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | 13j2ec27-0cc4-3541-9av6-219a178fcfe5 |
| span_id | 42f33gfa595246ert |
| parent_span_id | b2e339a595246e01 |

- **Log tonenNAWGegevens (log gemeente) persoon 2**

| Attribuut | Waarde |
|------------------|--------------------------------------|
| span_id | b2e339a595246e01 |
| name | tonenNAWGegevens |
| parent_span_id | <leeg> |
| trace_id | c6adf4df949d03c662b53e95debd411 |
| start_time | 2024-07-29 10:16:49.690 |
| end_time | 2024-07-29 10:16:49.723 |
| status_code | OK |
| resource.name | Balieapp |
| resource.version | 1.0.5 |
| attributeKey | dpl.core.processing_activity_id |
| attributeValue | 11x2ec2a-0774-3541-9b16-21ba179fcf15 |
| <u>BSN 2</u> | <leeg> |
| attributeKey | dpl.core.data_subject_id |
| attributeValue | 342ec27-aa41-dav6-219a178f5ty6 |
| span_id | aef53rfa59e240ert |
| parent_span_id | b2e339a595246e01 |

§ Relatie tussen data (Registratie verhuizing)

Om uiteindelijk alle data te kunnen rapporteren, is het van belang dat data op een bepaalde manier aan elkaar gekoppeld zijn. In dit voorbeeld zijn de data op de volgende manier gekoppeld:



§ Relatie met de standaard Logboek dataverwerkingen (Registratie verhuizing)

De relatie met de doelstellingen die gesteld zijn in de standaard Logboek dataverwerkingen worden, op basis van dit voorbeeld, als volgt concreet gerealiseerd:

- **het wegschrijven van logs van dataverwerkingen:** In dit voorbeeld is het de Baliemedewerker die via een Balieapplicatie de data van een Betrokkene kan bekijken. Deze acties zijn dataverwerkingen en worden gelogd bij zowel de Balieapplicatie als bij het BRP-systeem.
- **het aan elkaar relateren van logs van dataverwerkingen:** Er zijn in dit voorbeeld twee applicaties nodig om het totaal aan gevraagde informatie te kunnen tonen aan de betrokkene. Beide applicaties hebben een logboek voor verwerkte data. Om een totaalbeeld van de gelogde data te kunnen construeren, is een relatie tussen de logs nodig. In dit voorbeeld wordt de koppeling gelegd door het span_id en trace_id (gemeentelogboek) te linken aan het foreign_operation.span_id en foreign_operation.trace_id (BRP-logboek). De aanroep van de gemeente-applicatie naar het BRP betreft één opvraag op basis van één adres, één span_id en één trace_id. Het resultaat is meervoudig en moeten naar dezelfde span_id en trace_id leiden van de gemeente-applicatie. Het onderscheid zit in de verschillende BSN's van de personen die via een parent_span_id gekoppeld zijn.
- **het aan elkaar relateren van dataverwerkingen over de grenzen van systemen:** Naast het koppelen van logs van diverse applicaties, wordt ook een koppeling gelegd met het Register van verwerkingsactiviteiten. Dit gebeurt per applicatie op basis van het

`processing_activity_id` (register) te koppelen aan `dpl.core.processing_activity_id` (logboek). De diverse registers hebben **geen** directe koppeling met elkaar.

§ Paragraaf 3.3.1 Gedrag

1. *De applicatie MOET een Trace starten voor iedere Dataverwerking waarvan nog geen Trace bekend is.* Bij elke start van een verwerking wordt een `trace_id` aangemaakt. Bijvoorbeeld: in het voorbeeld komt er een bericht binnen bij de Balieapplicatie van de gemeente (tonenNAWGegevens). Er wordt direct een `trace_id` aangemaakt.
2. *De applicatie MOET voor iedere Dataverwerking een logregel wegschrijven in een Logboek. Log Sampling is niet toegestaan.* Een dataverwerking wordt opgeslagen als deze volledig is afgerond. In het voorbeeld is te zien dat logregels worden geschreven op het moment dat de vraag- en het antwoordbericht zijn afgerond.
3. *De applicatie MOET bijhouden of een Dataverwerking geslaagd of mislukt is en dit per Dataverwerking als status meegeven aan het Logboek.* Bij elke logregel in het voorbeeld staat de `status_code` vermeld ('OK').
4. *Als een Dataverwerking meerdere Betrokkenen heeft dan MOET de applicatie voor iedere betrokkene een aparte logregel wegschrijven. Een logregel kan naar 0 of 1 betrokkenen verwijzen.* In het voorbeeld gaat het om twee betrokkenen (`dpl.core.data_subject_id`), er wordt één logregel aangemaakt per BSN.
5. *Als een applicatie aangeroepen kan worden vanuit een andere applicatie MOET de applicatie Trace Context metadata accepteren bij een dergelijke aanroepen deze metadata kunnen omzetten naar een `foreign_operation` bericht.* Bij een externe verwerking (bijvoorbeeld `opvragenPersoonsgegevens`) geeft de Balieapplicatie de `trace_id` en `span_id` mee aan het BRP-systeem. Het BRP-systeem registreert de `trace_id`, en `span_id` als `foreign_operation.span_id`.

§ Voorbeeldapplicaties

§ Register van de verwerkingsactiviteiten (RvVA)

Dit project biedt een overzicht van dataverwerkingen binnen de overheid, waaronder het Register van de verwerkingsactiviteiten (RvVA). Dit register documenteert hoe data worden verwerkt,

waarschijnlijk ter ondersteuning van transparantie en naleving van de Algemene Verordening Gegevensbescherming (AVG).

Je kunt de voorbeeldapplicatie van het logboek en de specifieke RvvaA-sectie hier bekijken:

- gitlab.com/digilab.overheid.nl/ecosystem/logboek-dataverwerkingen/rva
- gitlab.com/digilab.overheid.nl/ecosystem/logboek-dataverwerkingen

§ Use Cases

Dit hoofdstuk bevat voorbeelden van use cases, gebaseerd op praktijkscenario's uit workshops. Ze illustreren hoe de standaard Logboek Dataverwerkingen kan worden toegepast in samenwerkingen tussen organisaties. De voorbeelden zijn niet-normatief en dienen ter verduidelijking voor organisaties met vergelijkbare processen.

§ Use Case 01: Moet de retentieperiode (bewaartermijn) van de logging in de logregel staan?

In **AVG artikel 30-1f** wordt de volgende maatregel benoemd: *Elke verantwoordelijke en, in voorkomend geval, de vertegenwoordiger van de verantwoordelijke houdt een register van de verwerkingsactiviteiten die onder hun verantwoordelijkheid plaatsvinden. Dat register bevat alle volgende gegevens: indien mogelijk, de beoogde termijnen waarbinnen de verschillende categorieën van gegevens moeten worden gewist.*

De concrete datum waarop een dataverwerking moet worden gewist uit het logboek, kan bepaald worden door middel van het bewaartermijn in het register en de eindtijd waarop een dataverwerking is gelogd in het logboek. Daardoor is het onnodig om de concrete verwijderdatum van een dataverwerking te registreren in het logboek.

Wanneer een logregel verwijderd moet worden, is afhankelijk van de situatie van een organisatie en het beleid van bewaarperiodes:

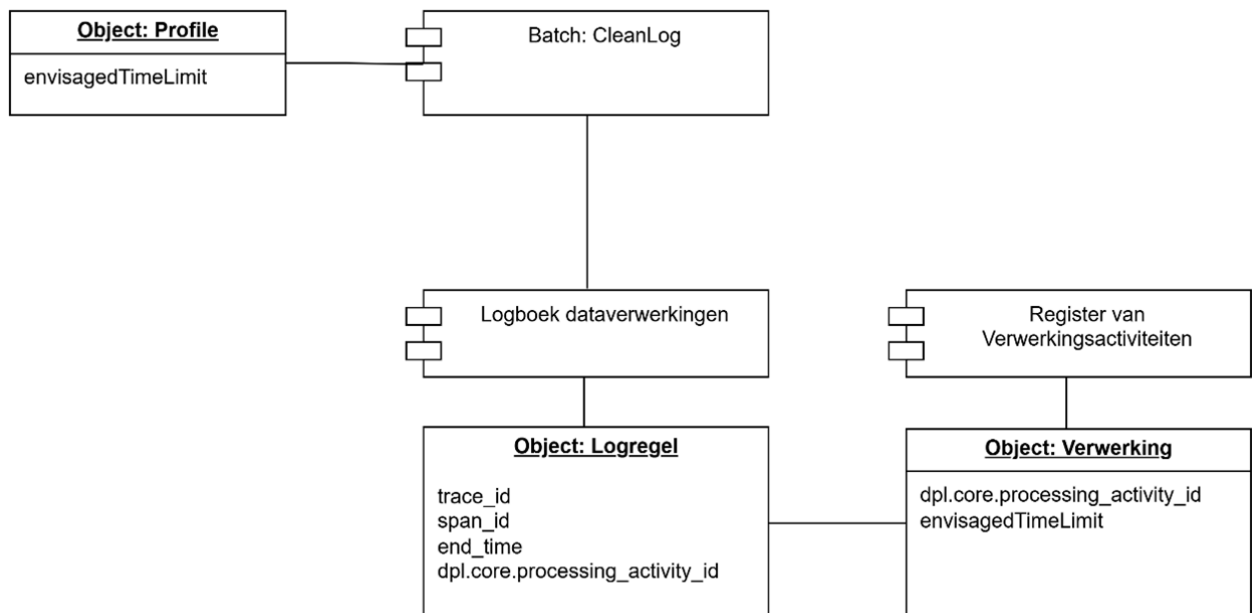
- **Gelijke bewaartermijn:** Als het beleid is dat alle logregels dezelfde bewaartijd hebben, kan er bijvoorbeeld elke dag een batch draaien die kijkt naar de eindtijd van een logregel en berekent of deze verwijderd kan worden.
- **Verskillende bewaartermijnen:** Als er per activiteit een andere bewaartermijn geldt, moet dat duidelijk worden aangegeven in het *Register van Verwerkingsactiviteiten* per activiteit in het veld `envisagedTimeLimit`.

Voorbeelden van waarden van het veld `envisagedTimeLimit`:

- **Datum (Date) of Datum/Tijd (DateTime):** 2025-02-23T00:00:00
- **Geheel getal (Integer):** 15 (dagen/maanden/jaren)
- **Tekst (String):** "20 jaar na onherroepelijk worden besluit"

Het belangrijkste is dat de organisatie duidelijk kan aantonen (*verantwoordingsplicht*) waarom een bepaalde bewaartermijn is gekozen en dat deze termijn in lijn is met de AVG. Dit betekent dat de keuze van het datatype minder cruciaal is dan de heldere vastlegging en naleving van de bewaartermijn zelf.

Concreet zou de logverwijderingssituatie er als volgt uit kunnen zien:



Scenario 1: Als het is toegestaan om een vaste retentieperiode voor alle logregels te hanteren, dan zou deze kunnen worden vastgelegd in de `envisagedTimeLimit` in een profiel. Dagelijks wordt een batch gedraaid om te bepalen of een logregel mag worden verwijderd. Als $\text{Huidige datum} - \text{envisagedTimeLimit} < \text{end_time}$ dan mag de logregel worden verwijderd.

Voorbeeld:

- Huidige datum: 1-8-2025
- `envisagedTimeLimit`: 6 maanden
- `end_time`: 1-1-2025

$1-8-2025 - 6 \text{ maanden} = 1-2-2025$, de logregel mag dus verwijderd worden.

Scenario 2: Als het **niet** is toegestaan om een vaste retentieperiode voor alle logregels te hanteren, dan moet deze worden vastgelegd in de `envisagedTimeLimit` in het *Register van Verwerkingsactiviteiten* per activiteit. De batch moet nu op basis van

`dpl.core.processing_activity_id` de `envisedTimeLimit` opzoeken in het *Register van Verwerkingsactiviteiten* en bepalen of de logregel verwijderd mag worden.

NOOT

Let op:

- De standaard Logboek Dataverwerkingen schrijft **niet** voor hoe het mechanisme van het verwijderen van logregels zou moeten werken. Het ontwerp en de architectuur moeten door de organisatie zelf bepaald worden.
- De standaard Logboek Dataverwerkingen schrijft **niet** voor wat de retentietijd is voor activiteiten. Het is de taak van de organisatie om te bepalen (op basis van de wettelijke basis) wat de bewaartijd is van een logregel.
- Het veld `envisedTimeLimit` in het Register van Verwerkingsactiviteiten moet **altijd** worden ingevuld, ook al is de bewaartijd voor alle activiteiten hetzelfde.

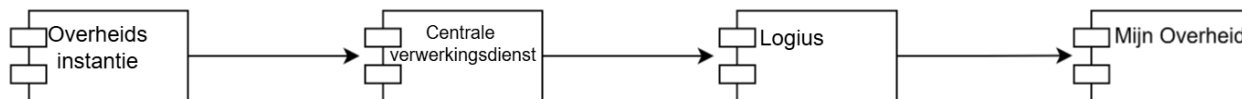
§ Use Case 02: Wordt er een vlag gelogd in de logregel, zodat ik weet dat de gegevens in deze logregel niet getoond mogen worden in het geval van inzageverzoek?

Nee, in het Logboek Verwerkingsgegevens worden geen vlaggen gelogd waardoor kan worden gezien dat de gegevens niet getoond mogen worden aan een burger. Het is aan de organisatie om procedures op te stellen om te regelen dat in specifieke gevallen data niet getoond mag worden aan een burger.

§ Use Case 03: Berichten versturen naar de burger vanuit een overheidsinstantie via een intermediaire organisatie

§ Procesbeschrijving Use Case 03

1. Een overheidsinstantie stuurt mededelende berichten in batchvorm naar een centrale verwerkingsdienst.
2. De centrale verwerkingsdienst (intermediair) verwerkt de batch en maakt hier individuele bestanden van. Deze individuele bestanden worden verstuurd naar Logius.
3. Logius verstuurt het individueel bestand naar de juiste inbox van de burger in MijnOverheid.



§ Logging

1. Voor zowel de verwerking van de batch als het verzenden van de (individuele) berichten wordt een logregel aangemaakt (in beide gevallen komt een BSN ‘tevoorschijn’).
2. De `trace_id` wordt aangeleverd door de overheidsinstantie, er wordt door de centrale verwerkingsdienst (intermediair) geen aparte `trace_id` aangemaakt noch wordt er een `foreign_trace_id` gelogd.
3. De centrale verwerkingsdienst (intermediair) heeft een eigen Register van Verwerkingsactiviteiten (via `dpl.core.processing_activity_id`).
4. De allereerste logregel geldt als ‘kapstok’, alle logregels daarna refereren naar de `span_id` van deze allereerste logregel via `parent_span_id`.
5. Elk individueel BSN krijgt een eigen logregel.

| Veld | Logregel 1 | Logregel 2 |
|-------------------------------|---|---|
| <code>trace_id</code> | bc9126aaae813fd491ee10bf870db292 | bc9126aaae813fd491ee10bf870db292 |
| <code>span_id</code> | b2e339a595246e01 | 414514cf1d40d6b2 |
| <code>parent_span_id</code> | NA | b2e339a595246e01 |
| <code>name</code> | VerwerkBatch | VerzendBericht |
| <code>start_time</code> | 2024-07-29 10:16:49.690 | 2024-07-29 10:16:49.690 |
| <code>end_time</code> | 2024-07-29 10:16:49.723 | 2024-07-29 10:16:49.723 |
| <code>status_code</code> | OK | OK |
| <code>resource.name</code> | BatchVerwerking | ZendDienst |
| <code>resource.version</code> | 1.0 | 2.1 |
| <code>attributeKey1</code> | <code>dpl.core.processing_activity_id</code> | <code>dpl.core.processing_activity_id</code> |
| <code>attributeValue1</code> | 11x2ec2a-0774-3541-9b16 | 12f2ec2a-0cc4-3541-9ae6 |
| <code>attributeKey2</code> | <code>dpl.core.data_subject_id</code> | <code>dpl.core.data_subject_id</code> |
| <code>attributeValue2</code> | 13j2ec27-0cc4-3541-9av6 | 19u2dd2a-0cb7-3541-9ae6-217 |
| <code>attributeKey3</code> | <code>dpl.core.data_subject_id_type</code> | <code>dpl.core.data_subject_id_type</code> |
| <code>attributeValue3</code> | BSN | BSN |
| <code>attributeKey4</code> | <code>dpl.core.foreign_operation.span_id</code> | <code>dpl.core.foreign_operation.span_id</code> |

| Veld | Logregel 1 | Logregel 2 |
|-----------------|----------------------------------|----------------------------------|
| attributeValue4 | 8ccfd3c567c51d68937c263e00a352be | 8ccfd3c567c51d68937c263e00a352be |

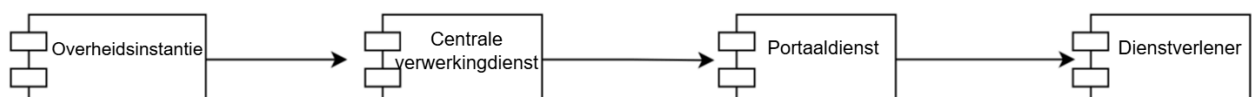
NOOT

Als het bericht 1 op 1 zou worden doorgestuurd, zou één logregel kunnen volstaan (geen persoonsgegevens zichtbaar).

§ Use Case 04: Service Bericht Mededeling vanuit een overheidsinstantie naar een dienstverlener via een intermediair

§ Procesbeschrijving Use Case 04

1. Een overheidsinstantie stuurt mededelende berichten in batchvorm naar een centrale verwerkingsdienst (intermediair).
2. De centrale verwerkingsdienst verwerkt batch en maakt hier individuele files van. De individuele files worden verstuurd naar een portaaldienst (bijv. Digipoort).
3. De portaaldienst verstuurt de individuele file naar de juiste dienstverlener.



NOOT

Als er geen HTTP protocol wordt gebruikt, moet er op een bepaalde manier toch headerinformatie worden verzonden.

§ Use Case 05: Persoonsgebeurtenisberichten via een intermediair

§ Procesbeschrijving Use Case 05

1. Een werkgeversdienst zendt individuele berichten naar een overheidsinstantie met betrekking tot personen met een werkverleden.
2. De overheidsinstantie bundelt persoonsgebeurtenisberichten in een batch en zendt deze naar een centrale verwerkingsdienst (intermediair).
3. De centrale verwerkingsdienst verwerkt de batch en maakt hier individuele files van en zendt deze naar het juiste EU-land.



Het proces kan ook andersom:

1. EU-land komt een persoonsgebeurtenisbericht (bijvoorbeeld van een persoon in het buitenland) en zendt deze naar de centrale verwerkingsdienst.
2. De centrale verwerkingsdienst bundelt persoonsgebeurtenisberichten van diverse EU-landen en stuurt deze als batch naar de overheidsinstantie.
3. De overheidsinstantie stuurt de batch door naar de werkgeversdienst.

NOOT

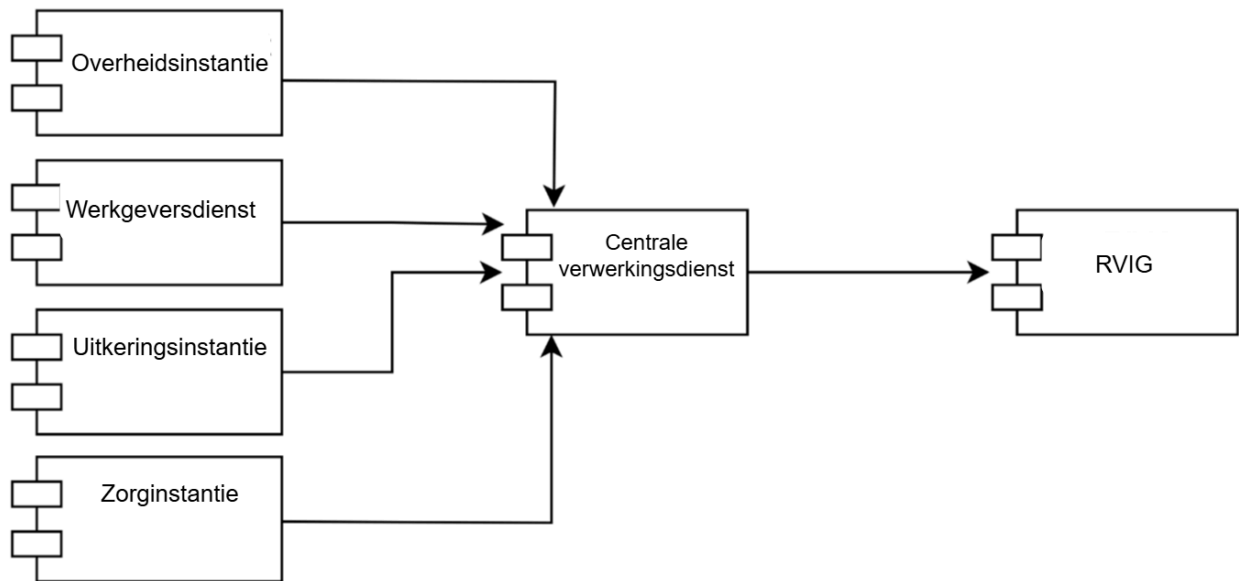
- De organisatie die als centrale verwerkingsdienst acteert zou een `trace_id` aan moeten maken op het moment dat er een bericht vanuit een EU-land komt.
- De overheidsinstantie in deze afbeelding is verantwoordelijke ook al komt het initiële bericht vanuit de werkgeversdienst.
- Niet elke organisatie geeft een acknowledgement terug.

§ Use Case 06: Register van Niet-Ingezeten (RNI)

§ Procesbeschrijving Use Case 06

1. Een overheidsinstantie/werkgeversdienst/uitkeringsinstantie/zorginstantie stuurt aanpassing ten aanzien van RNI naar een centrale verwerkingsdienst.

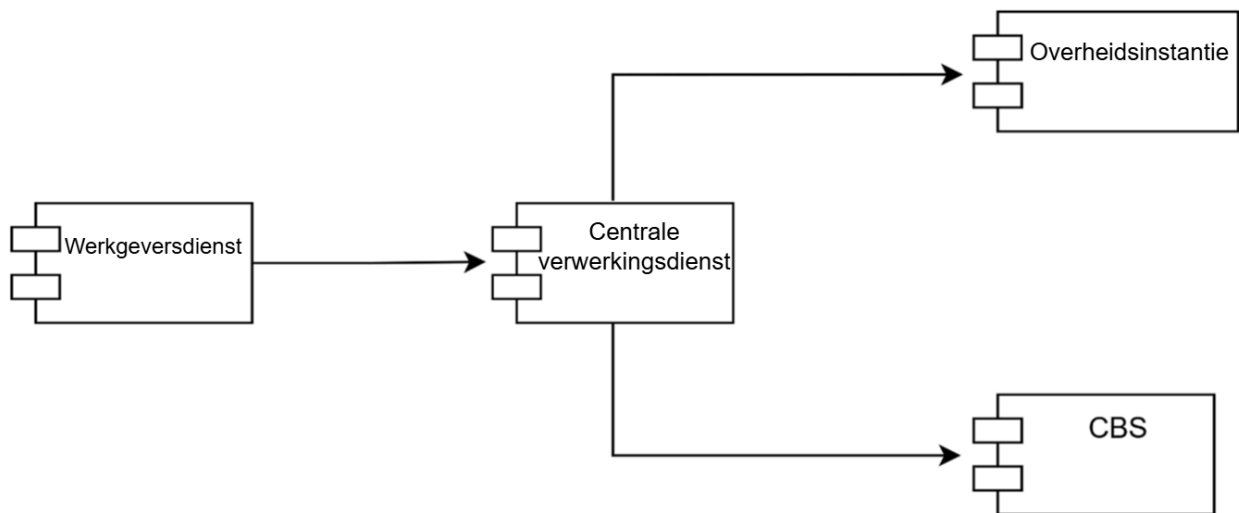
2. De centrale verwerkingsdienst stuurt aanpassing naar de RVIG (Rijksdienst voor identiteitsgegevens). In dit voorbeeld maakt de centrale verwerkingsdienst alleen individuele berichten indien aanpassingen in batchvorm zijn aangeleverd.
3. De RVIG voert de aanpassing uit in de RNI.



§ Use Case 07: Statistische Informatie via een intermediair

§ Procesbeschrijving Use Case 07

1. Een werkgeversdienst verstuurt statistische informatie over burgers bedoeld voor zowel een overheidsinstantie als het CBS (twee aparte berichten) in batch via een centrale verwerkingsdienst (intermediair).
2. De centrale verwerkingsdienst verwerkt de batch en stuurt individuele berichten naar zowel de overheidsinstantie als het CBS.
3. Het CBS anonimiseert de aangeleverde data.



NOOT

Het CBS moet de verwerking loggen van verwerkingen persoonsgegevens om persoonsgegevens te anonimiseren.

§ 6. FAQ

Dit hoofdstuk bevat veelgestelde vragen over de standaard Logboek Dataverwerking en in dit hoofdstuk worden deze veelgestelde vragen beantwoord en toegelicht.

§ Wat is het doel van de standaard Logboek Dataverwerking?

De overheid moet zich verantwoorden over haar handelen. Daar valt ook onder het verantwoorden van het gebruik van data. Met het gebruik van de standaard Logboek Dataverwerking is een organisatie in staat de logging van de verwerking van data gestandaardiseerd uit te voeren. Dit geldt zowel voor verwerkingen binnen de eigen organisatie als voor verwerkingen die tussen organisaties plaatsvinden.

§ Wie kan de standaard Logboek Dataverwerking gebruiken?

Elke organisatie die data verwerkt kan de standaard Logboek Dataverwerking inzetten bij processen waar logging en monitoring, bijvoorbeeld vanuit de wetgeving wenselijk is.

§ Is de standaard Logboek Dataverwerking alleen bedoeld voor de verwerking van persoonsdata?

Nee, de standaard kan ook worden gebruikt voor verwerking van andere typen data, zoals bijvoorbeeld de registratie van (geografische) objecten.

§ Is in de standaard Logboek Dataverwerking opgenomen hoe logging plaatsvinden ten aanzien van beveiligingsincidenten (denk ook aan technische systeemactiviteiten en toegangsbeheer)?

Nee, deze materie is buiten scope van de standaard Logboek Dataverwerkingen.

§ Wie is eigenaar en beheerder van de standaard Logboek Dataverwerking?

Het ministerie van Binnenlandse Zaken en Koninkrijksrelaties is verantwoordelijk voor deze standaard en de doorontwikkeling ervan. Het beheer wordt door Logius uitgevoerd.

§ Is mijn organisatie verplicht de standaard Logboek Dataverwerking te implementeren?

Er zijn momenteel geen verplichtingen voor gebruik van de standaard. Indien de standaard ooit verplicht wordt zal dit worden gepubliceerd bij het Forum Standaardisatie op de Pas-toe-of-leg-uitlijst. Hiervoor dient eerst de gehele procedure te worden doorlopen.

§ Wat is de relatie van Audit Log met de standaard Logboek Dataverwerking?

In de standaard Logboek Dataverwerking wordt geen identificerende data opgeslagen over gebruiker van het systeem (bijv. de ambtenaar die het systeem gebruikt). We gaan ervan uit dat daar in de organisatie een Audit log voor is ingericht, aangezien dat verplicht is vanuit BIO. Vanuit Audit Log kan wel een relatie gelegd worden met een verwerking in de standaard Logboek Dataverwerking door te verwijzen naar de `span_id` die de verwerking identificeert.

Voor redenering hiërarchisch, zie [besluit 4.4](#)

Daarnaast is het van belang om te beseffen dat het vastleggen van data over een gebruiker in de Audit Log, ook een dataverwerking is. Immers, de data van de gebruiker (bijv. de ambtenaar die het systeem heeft gebruikt) worden daarbij opgeslagen (verwerkt). Dat is dus een eigen, aparte, dataverwerking die gelogd dient te worden in de Logboek Dataverwerkingen van de verwerker.

§ Zijn er dingen die je moet aanpassen aan je Audit Log als je de standaard Logboek Dataverwerking implementeert?

In de logging worden geen identificerende data opgeslagen over gebruiker van het systeem (bijv. de ambtenaar die het systeem gebruikt). Om de link tussen een gebruiker en de standaard Logboek Dataverwerking te maken, kan de Audit Log worden aangepast door te verwijzen naar de `span_id` die de dataverwerking identificeert die door de gebruiker is uitgevoerd.

§ Kan je de standaard Logboek Dataverwerking implementeren als je een cloud (SaaS) applicatie gebruikt?

Ja dat kan. Het is wel van belang een duidelijk onderscheid te maken tussen verantwoordelijke en een verwerker van de data. Dit bepaalt bijvoorbeeld de Register van verwerkingsactiviteiten waarnaar u dient te verwijzen bij implementatie van de standaard Logboek Dataverwerking.

Voor meer informatie over de rol van een verantwoordelijke en een verwerker kunt u de website van [Autoriteit Persoonsgegevens](#) raadplegen.

§ Is de performance van de standaard Logboek Dataverwerking getest? / Zal de standaard Logboek Dataverwerking de performance van mijn applicaties beperken?

Ja, de performance is getest met een aantal demo-applicaties. De testen toonden aan dat er weinig tot geen performanceverlies was op geraakte applicaties.

§ Hoe werken het Register van verwerkingsactiviteiten (RvvA) en het Logboek Dataverwerkingen samen?

In het Register van verwerkingsactiviteiten (art 30 AVG) zijn de binnen de organisatie uit te voeren taken en processen waarin verwerkingen worden uitgevoerd benoemd. In de **Logboek Dataverwerkingen standaard** wordt de relatie gelegd tussen de beschreven processen in het register en de daadwerkelijk uitgevoerde activiteit waarbij **data** zijn verwerkt. **Door deze relatie ontstaat** inzicht in de taak en activiteit waarvoor de **data** verwerkt zijn.

§ Moet ik mijn RvvA aanpassen als ik deze standaard Logboek Dataverwerking implementeer?

Voor de implementatie van deze standaard is het noodzakelijk dat iedere verwerkingsactiviteit in uw RvvA uniek te identificeren is. Mocht dat nog niet het geval zijn, voeg dan een unieke identifier toe aan alle dataverwerkingen.

Het is aanbevolen – maar niet verplicht – om de RvvA benaderbaar te maken met een API. Dat voorkomt dat de identificatoren van de verwerkingsactiviteiten “hardcoded” moeten worden toegevoegd aan de logging en dat bij inzage, handmatig data uit de RvvA moeten worden toegevoegd in de logging.

Het is van belang dat, als de RvvA wordt aangepast, de wijzigingen toegevoegd worden als nieuwe verwerkingsactiviteiten met een eigen unieke identifier. Bestaande verwerkingsactiviteiten mogen niet wijzigen of verwijderd worden. Hierdoor blijven de oude verwijzingen uit de Logboek Dataverwerking intact.

§ Mijn organisatie heeft geen RvvA API. (Hoe) Kan ik dan nog steeds de standaard Logboek Dataverwerking implementeren?

Ja, dat kan nog steeds. Het is niet verplicht een RvvA API te implementeren, de RvvA is uiteraard wel verplicht in het geval van persoonsdataverwerking. Voor de implementatie van de Logboek Dataverwerkingen is het van belang dat iedere verwerkingsactiviteit te identificeren is met een unieke identifier.

Stel de RvvA is uitgewerkt in een MS-Exceldocument en het systeem heeft daar geen API-toegang toe. Daarnaast zijn de dataverwerkingen in de RvvA niet uniek te identificeren met een identifier.

In dat geval zal er een kolom moeten worden toegevoegd aan het MS-Exceldocument waar iedere dataverwerking een unieke identicator krijgt. Deze identificatoren van de dataverwerkingen in uw RvvA zullen dan 'hardcoded' moeten worden toegevoegd in de logging. Bij inzage zullen de data uit de RvvA, die horen bij een dataverwerking, handmatig moeten worden opgezocht.

§ Hoe gedetailleerd moet mijn RvvA zijn om de standaard Logboek Dataverwerking te implementeren? / Heeft de detailniveau van mijn RvvA invloed op de werking van deze standaard?

De standaard Logboek Dataverwerking gaat er alleen vanuit dat er een RvvA is. Hoe gedetailleerd de RvvA is, is een beslissing van de organisatie zelf. Uiteraard is het wel zo dat hoe gedetailleerder de RvvA is opgezet, hoe transparanter er kan worden gerapporteerd naar burger en (overige) overheid.

§ Wat gebeurt er als ik mijn RvvA wil wijzigen na de implementatie van de Logboek Dataverwerkingenstandaard?

Het is van belang dat als de RvvA aangepast moet worden, de wijzigingen toegevoegd worden als nieuwe verwerkingsactiviteit met een eigen unieke identicator. Bestaande verwerkingsactiviteiten mogen niet worden aangepast of verwijderd.

Hierdoor blijven de oude verwijzingen uit de Logboek Dataverwerking intact.

§ De standaard leunt zwaar op OpenTelemetry, waarom wordt OpenTelemetry niet voorgeschreven?

De keuze om aan te sluiten bij OpenTelemetry betekent dat het voor veel organisaties eenvoudiger is om de standaard te implementeren. Als het echter niet mogelijk is om OpenTelemetry te gebruiken, bijvoorbeeld omdat men volledig gebruik maakt van proprietary log collectors die dat niet ondersteunen, kan men wel de kleine benodigde subset van de OpenTelemetry interfaces zelf implementeren zodat aan de standaard wordt voldaan.

Door OpenTelemetry aan te bevelen als een eenvoudige implementatie houdt men de vrijheid om daar waar dat nuttig is een andere oplossing te kiezen dan een OpenTelemetry collector.

§ Hoe verhoudt de standaard zich tot andere logging standaarden zoals NEN 7513?

De [NEN 7513 norm](#) legt vast welke data moet worden gelogd van acties op persoonlijke gezondheidsinformatie. De norm beschrijft alle verschillende aspecten van gezondheidsinformatie die bij verschillende soorten activiteiten moet worden vastgelegd en verduidelijkt wat deze aspecten inhouden. Hierbij legt de norm niet het formaat van logging vast, enkel de vereiste informatie. Deze standaard is daarom inzetbaar in combinatie met de NEN 7513 norm, door vast te leggen in welk formaat dat past.

Hiervoor is, als onderdeel van de totstandkoming van deze standaard, allereerst [een modelanalyse van de norm](#) gemaakt in relatie tot welke informatie de core standaard bevat. Alle missende informatie is beschreven in [een potentiële extensie](#), om na te gaan of dit een haalbare aanpak is. Deze extensie zal niet worden vastgesteld en is enkel bedoeld als proof-of-concept tijdens de totstandkoming. Mocht in de toekomst er interesse zijn om de NEN 7513 norm te implementeren op basis van deze standaard, dan kan deze extensie als startpunt van het traject gebruikt worden.

§ Als mijn organisatie besluit de Logboek Dataverwerkingen-standaard te implementeren, moeten wij dan historische logregels aanpassen?

Nee, organisaties hoeven historische logregels niet aan te passen wanneer zij starten met het implementeren van deze standaard. De standaard is ontworpen zonder verplichting tot het herstructureren van bestaande data.

§ Is er een risicoanalyse beschikbaar voor deze standaard?

Ja, deze is beschikbaar:

- [De tekstuele beschrijving](#)
- [De bijbehorende tabel met dreigingen en classificaties](#)

§ A. Referenties

§ A.1 Normatieve referenties

[RFC2119]

Key words for use in RFCs to Indicate Requirement Levels. S. Bradner. IETF. March 1997.
Best Current Practice. URL: <https://www.rfc-editor.org/rfc/rfc2119>

[RFC8174]

Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words. B. Leiba. IETF. May 2017.
Best Current Practice. URL: <https://www.rfc-editor.org/rfc/rfc8174>