### Inrichtingsprincipes Samenhangende Objectenregistratie

De inrichtinsprincipes voor de Samenhangende Objecten Registratie zijn, gebaseerd op DSO, best practices, en Common Ground, samengevat de volgende:

CONCEPT

#### Gegevens en functionaliteit zijn gescheiden

Gegevens kunnen alleen worden benaderd via dataservices. Benaderen is maken, lezen en aanpassen van gegevens. Gegevens worden nooit verwijderd, maar *gemarkeerd* als verwijderd. Applicaties kunnen alleen via de dataservices gegevens maken, lezen en aanpassen.

De Ontkoppeling en interoperabiliteit van de gegevens maken verandering mogelijk (middels API strategie en URI strategie)

#### Gegevens hebben meta-data

Voor elk gegeven in de centrale voorziening is meta-data (data over de gegevens zelf) beschikbaar. Deze metadata wordt zoveel mogelijk [automatisch](#dataservices-houden-metadata-up-to-date) bijgehouden. In de metadata staan ook links naar structuur, definities en relaties van de gegevens.

#### Gegevens in de centrale voorziening zijn altijd actueel

Alle mutaties vinden plaats op de centrale gegevensset. Een wijziging aan een gegeven wordt genotificeerd.

##### Gegevens worden eenmalig en op één plek bijgehouden

De gegevens in de centrale voorziening zijn altijd actueel. Het principe "één bron - één waarheid" geldt.

##### Dubbele opslag betekent synchroniseren

Partijen die, om wat voor reden dan ook, een kopie dataset "on premise" willen kunnen en mogen dat doen, maar dan zijn zij zelf verantwoordelijk voor het up-to-date houen van die gegevensset. Om zulke partijen tegemoet te helpen hun gegevensset up-to-dat te houden kan zo'n partij zich abbonneren op de notificatie services.

#### Gegevens kunnen alleen via Dataservices worden benaderd

Om te garanderen dat de gegevens blijven voldoen aan de gestelde kwalteit en actualiteit kunnen ze alleen benaderd worden via *Dataservices*. Directe toegang tot de gegevens vanuit applicaties is ten strengste verboden.

#### Dataservices controleren

Alle controles, of het nou gaat om toegangscontrole tot de gegevens, of kwaliteitscontroles worden gedaan door de dataservices, en niet door applicaties. Zoals al eerder gezegd: aplicaties benaderen de gegevens via de dataservices en niet direct. Applicaties zorgen wel voor een nette foutafhandeling: als een service een foutcode retourneet, zorgt de applicatie voor de afhandeling van die fout door het geven van een melding, of door een nieuwe aanroep van de dataservice.

##### Dataservices regelen de toegang tot de gegevens

Bij het aanroepen van de dataservices wordt gecontroleerd of de gebruiker de gegevens wel mag benaderen. Benaderen kan weer zijn: maken, lezen en aanpassen. De dataservice controleert of de gebruiker wel de rechten heeft om de gegevens te maken, te lezen of aan te passen. Wat een gebruiker mag wijzigen hangt ook van de *klassificatie* van de gegevens af.

##### Dataservices loggen de transacties op de gegevens

Alle (!) transacties op de gegevens worden gelogd. Dit is nodig om een *audit-trail* te kunnen opbouwen.

##### Dataservices borgen de kwaliteit van de gegevens

Alle (!) controles die moeten plaatsvinden voordat een gegeven wordt gewijzigd worden in de dataservice gedaan. Enkele voorbeelden van een kwailiteitscheck:

* domeinwaarde check: komt de nieuwe waarde voor in een lijst met geldige waarden, of ligt de nieuwe waarde tussen twee grenswaarden.
* datum check: is de waarde volgens een bepaald datum formaat.
* type check: is de nieuwe waarde wel van een bepaald datatype.
* referentie check: mag een gegeven worden gemarkeerd als verwijderd, terwijl het gegeven nog ergens anders bestaat?
* plausibiliteits check: is de nieuwe waarde wel te verwachten op basis van eerdere of andere attribuutwaarden?

##### Dataservices houden metadata up-to-date

Waar mogelijk wordt de metadata door de dataservices up-to-date gehouden. Denk hierbij aan datum laatste mutatie, door wie gemuteerd, enz.

CONCEPT