

ÖREB-Kataster

Handbuch

Amt für Geoinformation Kanton Solothurn <agi@bd.so.ch>
Version 2.0.28, 2024-07-26 12:23

Inhaltsverzeichnis

| | |
|--|----|
| Vorwort | 1 |
| 1. Organisation | 2 |
| 1.1. Katasterverantwortliche Stelle (KVS) | 2 |
| 1.2. Zuständige Stelle für die Bereitstellung der Daten | 2 |
| 1.3. Staatskanzlei | 2 |
| 2. Systemübersicht | 3 |
| 2.1. Architektur | 3 |
| 2.2. Komponenten | 3 |
| 2.3. Systemumgebungen (Technisches Staging) | 7 |
| 3. Betrieb | 8 |
| 3.1. Durchführung und Überwachung des Betriebs | 8 |
| 3.2. Ausfall und Wiederherstellung des Betriebs | 9 |
| 4. Integrationsprozesse | 10 |
| 4.1. Inbetriebnahme (Daten-Erstintegration) | 10 |
| 4.2. Datenintegration | 11 |
| 4.3. Nachführung der ÖREB-Daten | 11 |
| 4.4. Nachführung der kantonalen Konfiguration | 13 |
| 4.5. Nachführung Daten der amtlichen Vermessung (inkl. Grundbuchkreise und amtliches Ortschaftsverzeichnis) | 13 |
| 5. Changemanagement | 14 |
| 5.1. National | 14 |
| 5.2. Kantonal | 14 |

Vorwort

Das ÖREB-Kataster-Handbuch des Kantons Solothurn dient der Organisation und Koordination der verschiedenen Abläufe des ÖREB-Katasters. Vertiefte (technische) Informationen zu den einzelnen Komponenten finden sich immer in den jeweiligen Github-Repositories.

Zum besseren Verständnis dienen folgende Dokumente:

- ÖREB-Kataster V2.0: «Infrastruktur- und Betriebskonzept» (Version 1.0 vom 12. Januar 2022)
- ÖREB-Kataster V2.0: «Neue Themen» (Version 1.3 vom 28. Februar 2022)
- ÖREB-Kataster V1.0: «Infrastruktur- und Betriebskonzept» (Version 1.0 vom 28. Januar 2019)
- ÖREB-Kataster V1.0: «Konzept Ersterfassung, Aufbereitung und Nachführung der ÖREB-Daten» (Version 1.0 vom 28. Januar 2019)

1. Organisation

1.1. Katasterverantwortliche Stelle (KVS)

Die katasterverantwortliche Stelle nach Art. 17 Abs. 2 ÖREBKV ist das Amt für Geoinformation (AGI).

In Zusammenarbeit mit dem Amt für Informatik und Organisation (AIO) stellt sie die ÖREB-Kataster-Infrastruktur bereit und betreibt diese selbständig. Sie nimmt die Daten der zuständigen Stellen in den Kataster auf, gewährleistet die Verfügbarkeit der Daten und stellt den Inhalt des Katasters als dynamischen und statischen Auszug via Geoportal zur Verfügung. Sie verknüpft im Auftrag der zuständigen Stellen die gesetzlichen Grundlagen mit den ÖREB-Daten.

Sie definiert die technischen Rahmenbedingungen für die Erhebung und Nachführung der ÖREB-Daten.

1.2. Zuständige Stelle für die Bereitstellung der Daten

Die Bereitstellung der Daten obliegt den nach der jeweiligen Fachgesetzgebung zuständigen Stellen von Bund, Kanton und Gemeinden.

Daten in Zuständigkeit des Kantons resp. der Gemeinden werden im kantonalen Datenmodell oder MGDM des Bundes inklusive der Rechtsvorschriften erfasst und in der GDI verwaltet. Die zuständige Stelle prüft die Daten vor der Publikation auf deren Vollständigkeit und Richtigkeit. Die Publikation im ÖREB-Kataster erfolgt spätestens zehn Arbeitstage nach Eintritt der Rechtskraft.

1.3. Staatskanzlei

Die Staatskanzlei meldet der KVS Änderungen in den gesetzlichen Grundlagen. Die KVS verwaltet im Auftrag der Staatskanzlei die gesetzlichen Grundlagen im Rahmenmodell.

2.1. Architektur

[illegible]

Abbildung 1. Architektur ÖREB-Kataster

Vorhandene kantonale Daten werden in der Edit-DB in einem dafür vorgesehenen Schema (`<amt>_<thema>_oerebv2`) in eine dem Rahmenmodell (Teilmodell Transferstruktur) äquivalente Struktur gebracht und anschliessend exportiert und in die ÖREB-DB importiert. Daten aus Fachanwendungen (Kataster der belasteten Standorten) und Bundesdaten werden direkt in die ÖREB-DB importiert, da sie bereits im Rahmenmodell bereitgestellt werden. Ein Datenumbau entfällt. Sämtliche Daten (inkl. amtliche Vermessung, Konfigurationsdaten, ...) werden in das gleiche Schema importiert. Es werden zwei identische Schemen erstellt: *live* und *stage*. Das *stage*-Schema dient der Validierung der Daten durch die dafür zuständigen Stellen. Ist diese erfolgreich, werden die Daten in das *live*-Schema importiert.

Der ÖREB-Kataster des Kantons Solothurn besteht aus den folgenden Komponenten:

- ÖREB-Datenbank
- ÖREB-WMS
- ÖREB-Webservice (inkl. pdf4oereb-Bibliothek)
- ÖREB-Iconizer
- ÖREB GRETL-Jobs
- ÖREB Web GIS Client (Bestandteil des Web GIS Clients)
- Dokumentenablage (bereits bestehende Komponente für die digitale Nutzungsplanung)
- ÖREB-Handbuch (vorliegendes Handbuch)

Nicht speziell aufgelistet sind die verschiedenen Arbeits-Datenbank-Schemen in der Edit-DB, die zur Herstellung der Daten im INTERLIS-Rahmenmodell benötigt werden. Dazu wird im Kapitel [Datenintegration](#) näher eingegangen. [GRETL-Jenkins](#), das zur Steuerung der GRETL-Jobs verwendet wird, wird ebenfalls nicht in diesem Handbuch detailliert behandelt und beschrieben, da es sich um eine Standardkomponente der GDI handelt.

2.2.1. ÖREB-Datenbank

Code-Repository:

<https://github.com/sogis/oereb-db>

Docker-Image:

<https://hub.docker.com/r/sogis/oereb-db>

In der ÖREB-Datenbank werden in zwei Schemen (*live* und *stage*) jeweils sämtliche für den Betrieb des ÖREB-Katasters notwendigen Daten gespeichert. Dazu gehören die eigentlichen ÖREB-Daten (inkl. kantonaler Gesetze und Verordnungen und Bundesgesetze und -verordnungen) sowie die amtliche Vermessung (im Bundesmodell), das amtliche Ortschaftsverzeichnis und die Konfigurationsdaten.

Die Datenstruktur entspricht einem mit *ili2pg* angelegten Schema und Tabellen und sind entsprechend normalisiert. Für die Bereitstellung via WMS werden zusätzlich denormalisierte, «flachgewalzte» Tabellen eingesetzt. Dazu wurde ein sehr einfache Datenmodell (SO_AGI_OeREB_WMS_20220222) geschrieben. Die Dokumente, welche in der Tabelle als JSON-Array gespeichert werden, wurden nicht als Strukturen ausmodelliert, da dies in der Datenbank zu JSON-Spalten führen würde, die wiederum mittels View zu Text gecastet werden müssen, weil QGIS-Server nicht damit umgehen kann. Somit wird das JSON-Array als einfacher Text modelliert und gespeichert.

Das *stage*-Schema dient der Validierung der Daten durch die zuständigen Stellen.

Die Kernfunktionalität des Repositories ist das (aus den Datenmodellen) automatische Herstellen der DDL-Queries zum Aufsetzen der Datenbank. Dieser Prozess wird mit einem [jBang-Java-Skript](#) durchgeführt.

Für Test- und Entwicklungszwecken wird mit Github Action ein [Docker-Image](#) der ÖREB-Datenbank erzeugt.

Im produktiven Betrieb wird nicht das Docker-Image verwendet, sondern die DDL-Queries werden mit *Ansible* auf dem bereits in der GDI vorhandenen Datenbankserver und -cluster deployed. Es ist die einzige Softwarekomponente, die nicht mittels *Docker* betrieben wird.

Benutzernamen und Passwörter der DB-Benutzer werden durch Setzen der Umgebungsvariablen PG_USER, PG_PASSWORD, PG_WRITE_USER, PG_WRITE_PASSWORD, PG_READ_USER und

PG_READ_PASSWORD definiert. Das Passwort für den DB-Benutzer postgres wird durch die Umgebungsvariable PG_ROOT_PASSWORD gesetzt.

2.2.2. ÖREB-WMS

Code-Repository:

<https://github.com/sogis/oereb-wms>

Docker-Image:

<https://hub.docker.com/r/sogis/oereb-wms>

Der ÖREB-WMS dient dazu die «flachgewalzten» Tabellen (siehe [ÖREB-Datenbank](#)) aus der ÖREB-Datenbank als WMS-Layer zu publizieren. Es werden nur die kantonalen Daten publiziert. Für die Bundes-ÖREB-Daten wird der WMS-Dienst des Bundes (GetMap-Request gespeichert in den Daten) direkt verwendet.

Der WMS-Server exponiert zwei Endpunkte:

- <https://geo.so.ch/wms/oereb>: WMS-Layer der kantonalen ÖREB-Daten
- <https://geo.so.ch/wms/oereb-symbols>: «Dummy»-Layer für die Generierung der Symbole der Eigentumsbeschränkungen (siehe [ÖREB GRETl-Jobs](#)). Die Symbole sind Bestandteil der Transferstruktur.

Sämtliche Konfiguration, insbesondere die QGIS-Projektdateien und die GeoPackage-Datei für den Dummy-Layer werden in das [Docker-Image](#) gebrannt. Die PostgreSQL-Verbindungsparameter inklusive Benutzername und Passwort und einer Option, die den `search_path` (default-Schemaname) definiert, werden in einem PostgreSQL Service File vorgehalten. Es wird in einem Secret platziert und unter `/etc/postgresql-common` in den Docker-Container gemountet.

Für den produktiven Einsatz wird somit nicht der bereits in der GDI vorhandene WMS-Server verwendet, sondern es wird bewusst ein zusätzlicher WMS-Server in Betrieb genommen.

Nach jedem Commit wird mit einer Github Action das Image neu gebuildet und innerhalb von 15 Minuten auf der [Test- und Integrationsumgebung](#) von *OpenShift* deployed.

Es wird QGIS 3.16 LTR eingesetzt. Das Dockerimage für die ARM64-Architektur verwendet QGIS 3.10 LTR aus dem offiziellen Ubuntu-Repository.

2.2.3. ÖREB-Webservice

Code-Repository:

<https://github.com/claeis/oereb-web-service>

<https://github.com/sogis/oereb-web-service-docker>

Docker-Image:

<https://hub.docker.com/r/sogis/oereb-web-service>

Der ÖREB-Webservice ist die M2M-Schnittstelle des ÖREB-Katasters und dient dem Bezug des ÖREB-Katasterauszuges (XML und PDF) als Downloaddienst. Das Bundesamt für Landestopografie hat dazu zwei Weisungen («[ÖREB-Webservice \(Aufruf eines Auszugs\)](#)» und «[ÖREB-Kataster - DATA-Extract](#)») erlassen.

Die Umwandlung des XML nach PDF übernimmt die im Webservice integrierte Bibliothek [pdf4oereb](#).

Der ÖREB Web Service des Kantons Solothurn unterstützt nur das Ausgabeformat XML.

Alle benötigten Daten müssen in einem einzigen Schema in einer PostgreSQL-Datenbank vorliegen. Die Konfiguration (inkl. der Datenbank-Verbindungsparameter) wird mittels ENV-Variablen gesteuert.

Jeder Commit im Code-Repository stösst einen Build-Prozess des Docker-Image-Repositories an. Das Docker-Image wird anschliessend automatisch in der [Test- und Integrationsumgebung](#) von *OpenShift* deployed.

2.2.4. ÖREB-Iconizer

Code-Repository:

<https://github.com/sogis/oereb-iconizer>

Der *ÖREB-Iconizer* ist ein Java-Programm, das zum Herstellen der einzelnen ÖREB-Symbole (als Bestandteil der Transferstruktur), verwendet wird. Die Symbole werden in einem manuellen Prozess hergestellt und als INTERLIS-Transferdatei zu den jeweiligen ÖREB-Gretl-Jobs kopiert. Während des Datenumbaus «kantonale Daten - ÖREB-Rahmenmodell» wird diese INTERLIS-Transferdatei importiert und das Symbol wird dem jeweiligen Symbol-Record des Rahmenmodells in der Datenbanktabelle zugewiesen. Da die Symbole nicht häufig ändern, ist dieser manuelle Herstellungsprozess der Symbole vertretbar.

Die Befehle für die Herstellung der INTERLIS-Transferdatei sind im Github-Repository beschrieben.

2.2.5. ÖREB GRETL-Jobs

Code-Repository:

<https://github.com/sogis/oereb-gretljobs>

Die ÖREB-GRETL-Jobs werden für den Datenfluss eingesetzt. Dazu gehören der Umbau der Daten in der Edit-DB, der Export in das Rahmenmodell, die Prüfung der INTERLIS-Transferdatei und der Import in die ÖREB-Datenbank. Daten, die bereits im Rahmenmodell vorliegen, müssen nur noch geprüft und in die ÖREB-Datenbank importiert werden.

2.2.6. Web GIS Client Werkzeug «Grundstücksinformation»

Service:

<https://github.com/qwc-services/>

Frontend:

<https://github.com/sourcepole/qwc2-extra>

Das Werkzeug ist ein Bestandteil des Web GIS Client und hat eigene Konfigurationensparameter. Diese Werkzeug ruft den ÖREB-Auszug (XML oder PDF) für das betroffene Grundstück auf und stellt den WMS zum passenden ÖREB-Katasterthema im Kartenfenster dar.

2.2.7. Dokumentenablage

Dokumentenablage:

https://geo.so.ch/docs/ch.so.arp.zonenplaene/Zonenplaene_pdf/

Für die Ablage und das Bereitstellen sämtlicher Dokumente wird die bestehende Lösung des AGI verwendet: Sie besteht aus einem klassischen Filesystem, das in die verschiedenen Desktop- und Serverumgebungen eingebunden werden kann und von den berechtigten Benutzern verwendet werden kann. Dieses Filesystem wird mittels API-Gateway (*nginx* Webserver) als HTTP-Ressource exponiert.

2.3. Systemumgebungen (Technisches Staging)

Es stehen drei vollständige Systemumgebung zur Verfügung:

- **Test:** Zum Testen neuer Funktionen und Bugfixes. Jeder Commit in einer Software-Komponente stösst die Build-Pipeline an (Github Action). Ist der Build und das Testing erfolgreich, wird die Komponente nach maximal 15 Minuten neu deployed und steht dem Benutzer zur Verfügung.
- **Integration:** Die Integrationsumgebung ist sehr nahe der Produktionsumgebung und dient vor allem für Abnahmetests und Systemintegrationstests. Manuelles Deployment.
- **Produktion:** Produktionsumgebung. Manuelles Deployment.

3. Betrieb

Das Amt für Geoinformation ist der Betreiber der ÖREB-Kataster-Infrastruktur und seinen verschiedenen Komponenten. Die IT-Grundinfrastruktur (Netzwerk, Storage, virtuelle Server, OpenShift-Infrastruktur) wird vom Amt für Informatik und Organisation (AIO) bereitgestellt und betrieben. Controlling- und Vorgabestelle ist das AGI in Zusammenarbeit mit dem AIO (§5 Abs. 1 GeoIV). Die Vorgaben zur Abnahme des Systems sind im Abnahmeprotokoll (Dokument «ÖREB-Kataster - Abnahmeprotokoll zur Systemabnahme») enthalten.

3.1. Durchführung und Überwachung des Betriebs

3.1.1. Service-, Reaktionszeiten und Verfügbarkeiten

Die Servicezeit ist der Zeitraum, während dem die vertraglich vereinbarte Verfügbarkeit der Leistung sichergestellt und ausgewiesen wird. Sie ist Basis für die Verfügbarkeitsmessung. Die Systeme stehen grundsätzlich auch ausserhalb der definierten Servicezeiten zur Verfügung (Betriebszeit 7 x 24 Stunden), jedoch ohne Gewährleistung der Verfügbarkeiten (vgl. unten).

Servicezeit: Montag bis Freitag, 07:00 – 12:00 und 13:00 – 17:30 Uhr, allgemeine Feiertage werden wie Sonntage behandelt.

Die Verfügbarkeit entspricht dem Prozentsatz, zu dem die vereinbarten Leistungen während der definierten Servicezeit (vgl. oben) erbracht werden.

Verfügbarkeit: 98% von Montag bis Freitag, 07:00 – 12:00 und 13:00 – 17:30 Uhr.

Die Störungsbehebungszeit bezeichnet die Zeitdauer während der Servicezeit zwischen dem Eingang der Störungsmeldung bei der Supportorganisation und der Wiederherstellung der Anwendung beziehungsweise Dienstes, inklusive der Information des Leistungsbezügers.

Störungsbehebungszeit: 16 Stunden

3.1.2. Betriebsüberwachung mit Monitoring, Alarmierung etc.

Die Überwachung des Betriebs des ÖREB-Katasters erfolgt auf unterschiedlichen Stufen:

Das AIO überwacht den Betrieb des *OpenShift*-Clusters in dem sämtliche vom AGI betriebenen ÖREB-Kataster-Komponenten (mit Ausnahme der Datenbank) deployed sind. Die ÖREB-Kataster-Komponenten werden vom AGI mit *OpenShift*-Bordmitteln überwacht (Live- und Readyness-tests, Selfhealing, etc.). Zusätzlich wird mit [Statuscake](#) ein Uptime- und Performancemonitoring betrieben. Fehler werden per E-Mail an die entsprechenden Personen mitgeteilt.

Die ÖREB-Datenbank wird nicht in *OpenShift* betrieben, sondern im bestehenden Datenbankcluster des AGI auf virtuellen Servern. Die Überwachung wird mit *NAGIOS* sichergestellt.

Störungsmeldungen bei den Daten-Integrationsprozessen werden der zuständigen Stellen und dem AGI per E-Mail mitgeteilt.

3.1.3. Datensicherung

Die Geobasisdaten der kantonalen ÖREB-Themen sind in der kantonalen GDI abgelegt und werden entsprechend durch bereits vorhandene Prozesse jeden Tag gesichert. Die Dokumente auf dem Filesystem werden durch übergeordnete Backupprozesse des AIO gesichert.

Jede Datenintegration in den ÖREB-Kataster geschieht mittels Rahmenmodell. Diese INTERLIS-Transferdatei wird archiviert.

3.1.4. Kontrolle zum Datenschutz

Die Geobasisdaten der ÖREB-Katasterthemen enthalten ausschliesslich nicht-sensitive Sachdaten und unterliegen deshalb nicht dem Datenschutz.

Die Rechtsvorschriften werden ohne sensible Daten publiziert. Die entsprechende Bearbeitung der Rechtsvorschriften liegt in der Verantwortung der zuständigen Stelle.

3.1.5. Statistiken, Kennzahlen und Messgrössen

Die Katasterabfragen (dynamischer Auszug) und die PDF-Katasterauszüge über den Web GIS Client werden mit [Matomo](#) protokolliert und können ausgewertet werden. Die Logdateien des API-Gateway werden zudem mittels [logit.io](#) ausgewertet. Diese Auswertungen (*Matomo* und *logit.io*) dienen als Grundlage für die Berichterstattung zuhanden der Katasteraufsicht und für die Tätigkeitsberichte der kantonalen Verwaltung.

3.1.6. Vorgehen im bei Datenfehler (inkl. organisatorische Aspekte wie Kommunikation etc.)

Fehler in den Geobasisdaten und Rechtsvorschriften, welche der KVS gemeldet werden, werden der zuständigen Stelle zur Korrektur gemeldet. Die Fehler sind je nach Art so schnell wie möglich zu beheben.

3.2. Ausfall und Wiederherstellung des Betriebs

Durch den Betrieb der meisten Komponenten in *OpenShift* werden diese automatisch selber neu gestartet. Falls das «Selfhealing» nicht funktioniert und bei der klassisch betriebenen ÖREB-Datenbank wird innerhalb von 30 Minuten nach Eingang der Meldung die Anwendung neu gestartet.

Ausgenommen davon ist die Bereitstellung der ÖREB-Daten des Bundes. In diesem Fall ist die Behebung durch die zuständige Stelle beim Bund abzuwarten.

Das komplette Neuaufsetzen der ÖREB-Katasterinfrastruktur ist dank des Betriebs mittels *Docker* und *OpenShift* einerseits und der ebenfalls als Code vorgehaltenen und in einem VCS verwalteten Konfigurationen bereits zu grossen Teilen automatisiert können und dauert maximal einen Arbeitstag. Das Einspielen sämtlicher Daten mit den ÖREB-Integrationsprozessen in die ÖREB-Datenbank dauert circa 2 Stunden. Zusätzlich kann die Datenbank selbstverständlich bei Bedarf mit einem Datenbankdump wiederhergestellt werden.

Reihenfolge der Inbetriebnahme der Komponenten bei einem Neuaufsetzen:

1. ÖREB-Datenbank
2. ÖREB-WMS
3. Deployen der ÖREB-GRETL-Jobs in Jenkins
4. Importieren der Konfiguration und Grundlagedaten. Siehe [Inbetriebnahme \(Daten-Erstintegration\)](#)
5. Importieren der ÖREB-Transferdaten
6. Installation des Werkzeuges im Web GIS Client.

4. Integrationsprozesse

Die Integrationsprozesse sind dafür verantwortlich, dass die für den Betrieb des ÖREB-Katasters notwendigen Daten in der ÖREB-Datenbank integriert werden. Konkret sind das:

- ÖREB-Daten
- Codelisten (nur Bund)
- Bundesgesetze und -verordnungen, sowie kantonale Gesetze und Verordnungen
- ÖREB-Rahmenmodell Teilmodell «Konfiguration»
- AV-Daten (inkl. Grundbuchkreise und amtliches Ortschaftsverzeichnis)

4.1. Inbetriebnahme (Daten-Erstintegration)

Bei der Daten-Erstintegration in die Edit-DB für spätere Bearbeitung der kantonalen Konfigurationsdaten ist die Importreihenfolge der verschiedenen Datensätze entscheidend. Es handelt sich ausschliesslich um INTERLIS-Transferdateien. Diese sind teilweise voneinander abhängig, so dass in der Datenbank Fremdschlüssel entstehen. Dies trifft vor allem auf die Konfigurationen zu. Es ergibt sich folgende Reihenfolge, wobei ab (6.) die Reihenfolge egal ist:

1. Zuständige Stellen (Kanton)
2. Gesetze (Kanton)
3. Gesetze (Bund)
4. Themen (Bund)
5. Themen (Kanton)
6. Logos (Kanton)
7. Texte (Kanton)
8. Verfügbarkeit (Kanton)
9. Grundbuchkreise (Kanton)

Siehe auch [DEVELOP.md](#) für die Dataset-Identifizierung und ili2pg-Befehle.

Daraus ergibt sich auch die Reihenfolge wie die Gretl-Jobs für die Konfigurationen ausgeführt werden müssen, um Daten von der Edit-DB in die ÖREB-DB zu transferieren:

1. oerebv2_bundesgesetze
2. oerebv2_bundeskonfiguration
3. oerebv2_konfiguration_zustaendigestellen
4. oerebv2_konfiguration_gesetze
5. oerebv2_konfiguration_themen
6. oerebv2_konfiguration_logo
7. oerebv2_konfiguration_text
8. oerebv2_konfiguration_verfuegbarkeit
9. oerebv2_konfiguration_grundbuchkreis

Die Reihenfolge der eigentlichen Daten (inkl. AV und PLZ/Ortschaft) spielt keine Rolle mehr.

4.2. Datenintegration

Für die Datenintegration sämtlicher Daten in den ÖREB-Kataster werden GRETJ-Jobs verwendet. Ein GRETJ-Job kann je nach Datensatz unterschiedliche Aufgaben wahrnehmen:

Werden die ÖREB-Daten in der Edit-DB nachgeführt, werden diese zuerst in ein Rahmenmodell-äquivalentes Schema in der Edit-DB umgebaut (*arp_nutzungsplanung_oerebv2*), anschliessend in eine INTERLIS-Datei exportiert, validiert und in die ÖREB-Datenbank importiert. In der ÖREB-Datenbank sind zwei Schemen vorhanden: *stage* und *live*. Zuerst werden die Daten in das *stage*-Schema importiert. Die Daten müssen von der zuständigen Stelle visuell validiert werden und freigegeben werden. Die WMS-Layer aus dem *stage*-Schema sind als geschützte Layer im Web GIS Client freigeschaltet, wo die Daten kontrolliert werden können.

Das Starten eines solchen Jobs wird in [GRETJ-Jenkins](#) gemacht. Nur berechtigte Personen können den Job starten. Nach erfolgter visueller Validierung muss die berechtigte Person den Import in das *live*-Schema freigeben.

Sämtliche Aktionen in *GRETJ-Jenkins* werden transparent geloggt und sind nachvollziehbar. Die erzeugten Daten im Rahmenmodell werden archiviert und zusätzlich in S3 gespeichert.

4.3. Nachführung der ÖREB-Daten

4.3.1. Bundesdaten, Bundeskonfiguration, Bundesgesetze und -verordnungen

GRETJ-Jobs:

oerebv2_bundesdaten: ÖREB-Daten im Transfermodell. Werden jede Nacht importiert.

oerebv2_bundesgesetze: Gesetzlichen Grundlagen (Gesetz und Verordnungen) des Bundes. Werden manuell, «on demand» importiert.

oerebv2_bundeskonfiguration: Themen, Logos und Texte des Bundes. Werden manuell, «on demand» importiert.

Bundesdaten werden durch das zuständige Bundesamt nachgeführt und auf der Webseite <https://data.geo.admin.ch/> im entsprechenden Unterordner bereitgestellt. Aus nicht nachvollziehbaren Gründen sind die ÖREB-Unterordner jedoch nicht direkt aufgelistet, sondern die Unterordner-Url muss mühsam selber zusammengestöpselt werden (aus Basis-Url und Themennamen).

Die Bundeskonfiguration und die Bundesgesetze und -verordnungen sind auf der INTERLIS-Modell- und Datenablage zu finden (http://models.geo.admin.ch/V_D/OeREB/).

4.3.2. Kantonale Daten

Nutzungsplanung

GRETJ-Jobs:

oerebv2_nutzungsplanung: Kommunale Nutzungsplanung. Job wird manuell, «on demand» ausgelöst. *oerebv2_nutzungsplanung_kanton*: Kantonale Nutzungsplanung. Job wird manuell, «on demand» ausgelöst.

Die Daten der Nutzungsplanung (inkl. Waldabstandslinien und Lärmempfindlichkeitsstufen in Nutzungszonen) werden durch das AGI in den Datensatz in der Edit-DB eingepflegt.

Planungszonen

GRETJ-Jobs:

oerebv2_planungszonen: Planungszonen. Job wird manuell, «on demand» ausgelöst.

Die Daten werden in der Nutzungsplanung geführt und werden durch das AGI in den Datensatz in der Edit-DB eingepflegt.

Gewässerraum

GRETJ-Jobs:

oerebv2_gewaesserraum: Gewässerraum. Job wird manuell, «on demand» ausgelöst.

Die Daten des Gewässerraums liegen nicht in jeder Gemeinde mit Nutzungsplanung rechtsgültig vor. Aus diesem Grund muss das Thema bei der jeweiligen Gemeinde zuerst in der Edit-DB im Schema `agi_konfiguration_oerebv2` freigeschaltet werden.

Waldgrenzen

GRETJ-Jobs:

oerebv2_waldgrenzen: Job wird manuell, «on demand» ausgelöst.

Die Waldgrenzen werden in einem kantonalen Datenmodell durch das Amt für Geoinformation (AGI) in der Edit-DB nachgeführt.

Waldreservate

GRETJ-Jobs:

oerebv2_waldreservate: Job wird manuell, «on demand» ausgelöst.

Die Waldreservate werden als Übergangslösung in einem kantonalen Datenmodell durch das Amt für Geoinformation (AGI) in der Edit-DB nachgeführt. Später werden sie im Waldportal erfasst.

Kataster der belasteten Standorte

GRETJ-Jobs:

oerebv2_belastete_standorte: Job wird manuell, «on demand» ausgelöst.

Für die Nachführung des Katasters der belasteten Standorte setzt das Amt für Umwelt (AfU) die Software *altlast4web* ein. Die Software läuft ausserhalb der GDI.

Die Daten des Kataster der belasteten Standorte werden durch *altlast4web* als HTTP-Ressource (nur innerhalb des Kantonsnetzes verfügbar) bereitgestellt. Ein Datenumbau ist somit nicht notwendig. Der GRETJ-Job muss die Daten nur validieren und in das *stage*- resp. *live*-Schema importieren. Eine visuelle Validierung durch die zuständige Stelle im *stage*-Schema findet jedoch ebenfalls statt.

Planerischer Gewässerschutz

GRETJ-Jobs:

oerebv2_grundwasserschutz: Job wird manuell, «on demand» ausgelöst.

Der planerische Gewässerschutz wird im MGDM durch das Amt für Geoinformation (AGI) in der Edit-DB nachgeführt.

Schützenswerte Objekte (Denkmal) / Geotope / Naturreservate

GRETJ-Jobs:

oerebv2_einzelschutz_denkmal: Job wird manuell, «on demand» ausgelöst.

oerebv2_einzelschutz_geotop: Job wird manuell, «on demand» ausgelöst.

oerebv2_einzelschutz_naturreservat: Job wird manuell, «on demand» ausgelöst.

Denkmal: Diese ÖREB-Kataster-relevanten Daten werden durch das Amt für Archäologie und Denkmalschutz in der Fachanwendung *ArtPlus* ausserhalb der kantonalen GDI nachgeführt. Eine Teilmenge der Daten wird täglich in die Edit-DB importiert.

Geotope werden in einem kantonalen Datenmodell durch das Amt für Umwelt (AFU) in der Edit-DB nachgeführt.

Naturreservate werden in einem kantonalen Modell durch das Amt für Geoinformation (AGI) in der Edit-DB nachgeführt.

4.4. Nachführung der kantonalen Konfiguration

GRETl-Jobs:

oerebv2_konfiguration_zustaendigestellen
oerebv2_konfiguration_gesetze
oerebv2_konfiguration_themen
oerebv2_konfiguration_logo
oerebv2_konfiguration_text
oerebv2_konfiguration_verfuegbarkeit
oerebv2_konfiguration_grundbuchkreis

Bis auf oerebv2_konfiguration_verfuegbarkeit sind sämtliche Konfigurations-Jobs «on demand»-Jobs. Die Verfügbarkeit muss täglich ausgeführt werden, damit das Datum des Standes der amtlichen Vermessung aktualisiert wird.

Die Daten werden durch das Amt für Geoinformation (AGI) nachgeführt. Siehe dazu auch eine ausführliche **Anleitung:** H:\BJSVW\Agi\ÖREB-Kataster\Nachführung\Konfiguration_Annex\Anleitung
freischalten_Version2.0.docx. ÖREB-Themen

4.5. Nachführung Daten der amtlichen Vermessung (inkl. Grundbuchkreise und amtliches Ortschaftsverzeichnis)

GRETl-Jobs:

oereb_av: Job wird jede Nacht automatisch ausgeführt.
oereb_plzo: Job wird jede Nacht automatisch ausgeführt.

Die Daten werden mit einem GRETl-Job aus der Edit-DB direkt («Db2Db-Task») in die ÖREB-Datenbank kopiert. Die AV-Daten werden von den zuständigen Nachführungsgeometern mindestens einmal wöchentlich geliefert und anschliessend automatisch in die Edit-DB importiert.

5. Changemanagement

5.1. National

Geregelt in Gesetzen, Verordnungen und Weisungen des Bundes resp. der Swisstopo. Es wird dazu nicht näher eingegangen.

5.2. Kantonal

Feedback, Änderungswünsche und Weiterentwicklungsideen zum ÖREB-Kataster werden im Ticketsystem des AGI resp. bei der direkt betroffenen Softwarekomponente verwaltet. Deren Umsetzungen werden durch die Amtsleitung geplant und freigegeben. Grössere Anpassungen werden als Projekt behandelt.