



Schweizerische Eidgenossenschaft  
Confédération suisse  
Confederazione Svizzera  
Confederaziun svizra

Eidgenössisches Departement für Verteidigung,  
Bevölkerungsschutz und Sport VBS  
Bundesamt für Landestopografie swisstopo  
**Geodäsie und Eidgenössische Vermessungsdirektion**

Bediener:		19.07.2021
-----------	--	------------

## Ko-Produktion STDL

### Technischer Beschrieb «Prototyp zentrale Datenhaltung»



## Inhaltsverzeichnis

Der Prototyp .....	5
Lösungsfindung .....	5
GIS System in virtueller Umgebung.....	5
Zentrale Datenhaltung und dezentrale Arbeitsplätze .....	5
Prototyp – Systemübersicht.....	6
Zentrale Datenhaltung .....	6
CDE - Common Data Environment.....	6
Messages.....	6
Dezentrale Arbeitsplätze (GIS Arbeitsplätze) .....	7
Technischer Beschrieb der Umsetzung .....	8
CDE – Zentrale Datenhaltung .....	8
Technische Implementation Prototyp .....	8
Tabelle «cms_imp_version» .....	8
Interlis2-OID als Datensatzschlüssel .....	8
CDE – Web Service Endpunkte .....	9
Technische Implementation Prototyp .....	9
REST-API .....	9
Daten «Check-Out» .....	9
Daten «Check-In» .....	9
Datensatz Versionierung .....	9
Messages .....	10
Technische Implementation Prototyp .....	10
Meldung eintragen .....	10
Bezug von Meldungen .....	11
GIS Arbeitsplätze .....	12
Daten «Check-Out» .....	12
Daten «Check-In» .....	12
Datensätze Löschen .....	12
Geometrien .....	12
Weitere technische Überlegungen .....	13
Zentrale Datenhaltung .....	13
Datenbank.....	13
Geometrien Handhabung .....	13



Datenmodell .....	13
Web Service Endpunkte .....	14
Programmiersprache PHP .....	14
Standard REST .....	14
Sicherheit der Endpunkte .....	14
Versionierung der Endpunkte .....	15
Datenqualität und Prüfungen .....	15
Überlegungen zu REST, JSON, XML und Interlis2 .....	15
Standard REST .....	15
Body als JSON, XML oder Interlis2 .....	15
Nachteil gegenüber Interlis2 .....	16
Kontoverwaltung und Berechtigungssystem .....	16
Eigene Zugangsverwaltung vs. Drittanbieter .....	16
Messages - Benachrichtigungen .....	16
Technische Möglichkeiten .....	16
Lesebestätigung, Meldungen Quittieren .....	17
Prozesse und Meldungen .....	17
Handhabung der Informationsebenen .....	17
Abgrenzung und Abdeckung von Nutzen .....	18
Anhang .....	19
Bild: JSON-Objekt Methode GET .....	19
Bild: JSON-Objekt Methode POST .....	20



## Änderungen am Beschrieb

Beschreibung	Datum	Verfasser
Initialbericht	02.07.2021	U. Sommerhalder
Finalbericht	19.07.2021	U. Sommerhalder



## Der Prototyp

Der Prototyp (Sandbox) soll die Machbarkeit aufzeigen, wie möglichst gemeinsam an Daten gearbeitet werden kann. Dies basierend auf den «once only» Prinzip.

## Lösungsfindung

### GIS System in virtueller Umgebung

Als technisch einfachste Umsetzung, wäre ein GIS System, dass komplett virtuell betrieben wird und eine gemeinsame Datenhaltung (z.B. Datenbank) anbietet.

Die Daten könnten über einen virtuellen Arbeitsplatz erfasst und geändert werden.

Hierbei stellt sich die Frage nicht, ob dies auf einer Umgebung «On Premise» (Rechenzentrum) oder auf einer «Hosting» Umgebung (Cloud Anbieter) betrieben wird.

Die meisten GIS System Hersteller oder Anbieter können dies technisch umsetzen.

### Zentrale Datenhaltung und dezentrale Arbeitsplätze

Eine Umsetzung mit einem GIS systemunabhängigen Ansatz verfolgt die Lösung mit einer zentralen Datenhaltung und einem Datenaustauschmechanismus.

Die zentrale Datenhaltung bietet eine Schnittstelle an, die durch die GIS Arbeitsplätze angesprochen werden kann und worüber die Daten bezogen und wieder abgegeben werden können.

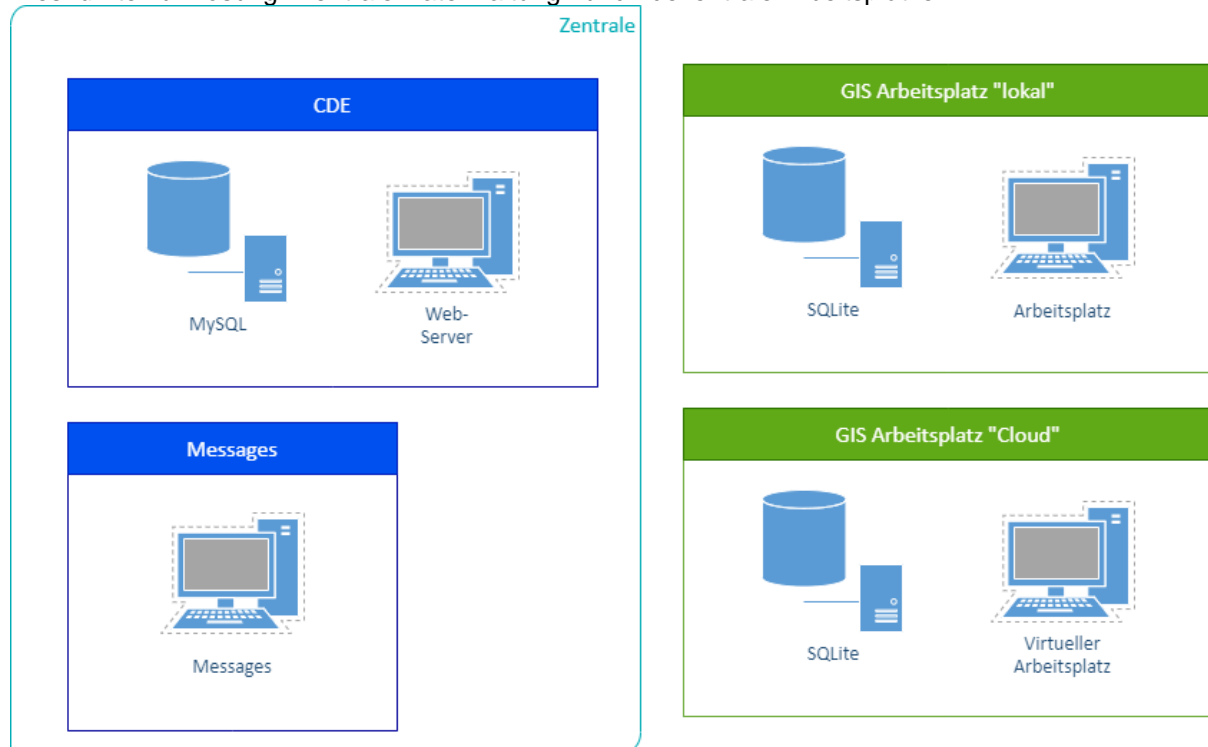


## Prototyp – Systemübersicht

Für die Auswahl der Lösung spielten einige Gedanken und Anforderungen aus dem Projektteam mit:

- Wahrung der Systemneutralität.
- Bestehende GIS Applikationen können weiterverwendet werden.
- Möglichst hohe Benutzerakzeptanz erreichen (z.B. Wie wird eine Cloud Umgebung von den Benutzenden akzeptiert?).

Dies führte zur Lösung «zentrale Datenhaltung» und «dezentrale Arbeitsplätze».



## Zentrale Datenhaltung

Die Zentrale umfasst verschiedene Komponenten.

### CDE - Common Data Environment

Zentrale Datenhaltung mit einer Datenbank und den Web Service-Endpunkten für den Datenbezug und die Datenabgabe.

### Messages

Kommunikationskomponente für die Benachrichtigungen innerhalb der Prozesse (Mutationsabläufe).



## Dezentrale Arbeitsplätze (GIS Arbeitsplätze)

Innerhalb des Prototyps wurde ein Arbeitsplatz «lokal», sprich in einer bestehenden IT-Umgebung in einem Unternehmen installiert und verwendet. Also eine Installation, wie sie heute bei den meisten Unternehmungen im Einsatz ist.

Ein weiterer Arbeitsplatz wurde virtuell (auf einer Cloud Umgebung) installiert und verwendet.

Mit dieser Umgebung soll aufgezeigt werden, dass die technische Umsetzung sämtliche Umgebungen und Kombinationen zulässt:

- Betrieb komplett in einer virtuellen Umgebung.
- Hybridbetrieb zwischen virtuellen und lokalen Umgebungen.
- Betrieb komplett in einer lokalen Umgebung.



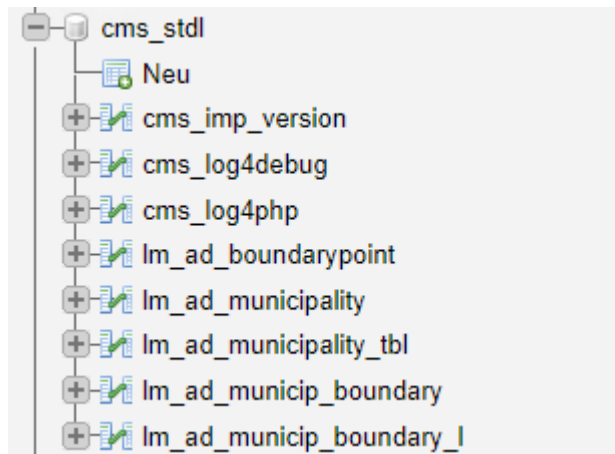
## Technischer Beschrieb der Umsetzung

### CDE – Zentrale Datenhaltung

#### Technische Implementation Prototyp

Die zentrale Datenhaltung wurde in einer MariaDB **Datenbank** umgesetzt.

Das Datenbankschema wurde vom GIS Datenmodell übernommen zusätzlich mit einer Tabelle für die Verwaltung der Importe und Datensatzversionen (cms\_imp\_version) versehen.



#### Tabelle «cms\_imp\_version»

In dieser Tabelle werden die Importe der Daten in der Datenbank protokolliert, die ID wurde auf den einzelnen Datensätzen als Version eingetragen und somit der Bezug zur Änderung festgehalten:

Tabelle	Aktion
<input type="checkbox"/> cms_imp_version	★ Anzeigen Struktur St
<input type="checkbox"/> cms_log4debug	★ Anzeigen Struktur St
<input type="checkbox"/> cms_log4php	★ Anzeigen Struktur St
<input type="checkbox"/> lm_ad_boundarypoint	★ Anzeigen Struktur St
<input type="checkbox"/> lm_ad_municipality	★ Anzeigen Struktur St
<input type="checkbox"/> lm_ad_municipality_tbl	★ Anzeigen Struktur St
<input type="checkbox"/> lm_ad_municip_boundary	★ Anzeigen Struktur St
<input type="checkbox"/> lm_ad_municip_boundary_I	★ Anzeigen Struktur St

8 Tabellen Gesamt

#### Interlis2-OID als Datensatzschlüssel

Als Datensatzschlüssel wurde die Interlis2-OID verwendet.





## CDE – Web Service Endpunkte

### Technische Implementation Prototyp

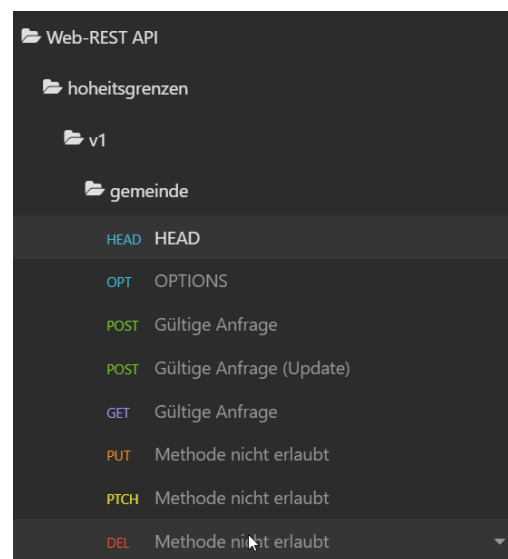
Der Webserver wurde auf einem Windows Betriebssystem mit der PHP-Entwicklungsumgebung XAMPP (<https://www.apachefriends.org>) bereitgestellt.

### REST-API

Die REST-API wurde in PHP geschrieben und bietet im Prototyp einen Endpunkt «gemeinde» an.

Umgesetzt wurden die Methoden:

- **POST**
- **GET**



### Daten «Check-Out»

Die Daten können auf dem Endpunkt mit der GET-Methode abgeholt werden und in der GIS Applikation verwendet werden.

Es werden alle ungelöschten Daten als JSON-Objekt zurückgegeben. Sozusagen die ganze Schweiz im rechtsgültigen Zustand.

### Daten «Check-In»

Die Daten können auf dem Endpunkt mit der Methode POST abgegeben werden und werden in der zentralen Datenbank eingepflegt.

Es werden die geänderten und gelöschten Datensätze als JSON-Objekt übergeben. Die Änderungen wurden im GIS System vorgemerkt um diese Objekte anschliessend zu übertragen.

### Datensatz Versionierung

Im Prototyp wurde ein einfaches Datensatz Versionierungssystem implementiert. Beim «Check-In» wurde eine Versionsnummer (Laufnummer hochgezählt) erstellt und diese in der Tabelle «cms\_im\_version» gespeichert.

Jedes Objekt wurde beim Import in die Tabelle mit dieser «Import-ID» (Version) ergänzt und als eigenständiger Datensatz gespeichert. Somit konnten die Änderungen an den einzelnen Datensätze reproduziert werden.

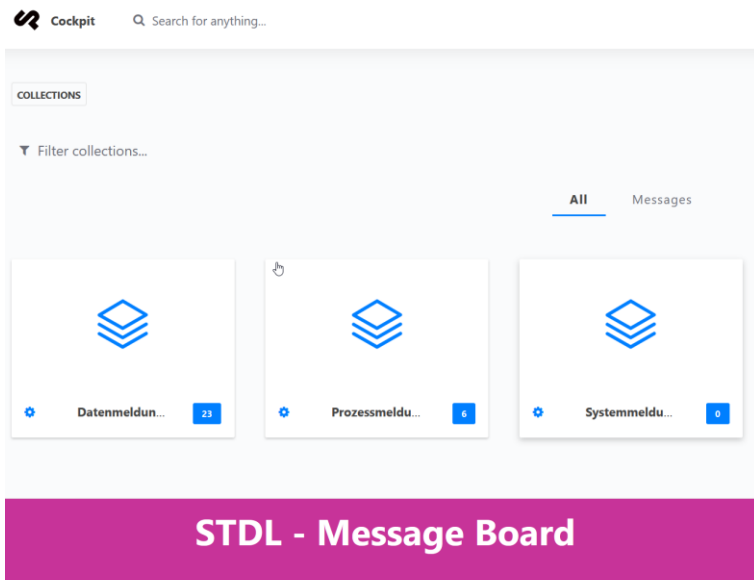


## Messages

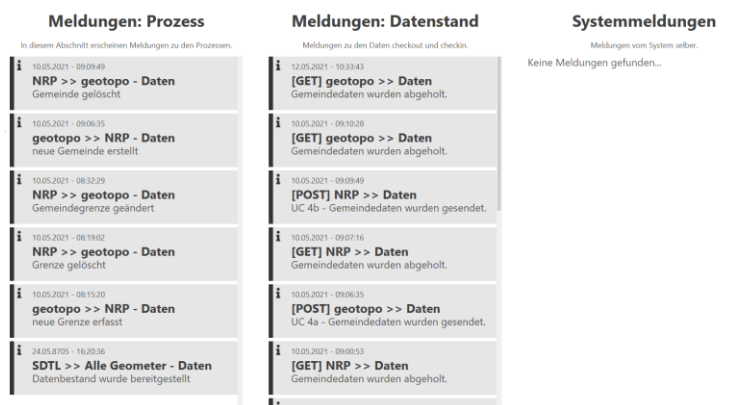
### Technische Implementation Prototyp

Für eine schnelle und rasche Implementation wurde ein Headless CMS (Cockpit) eingesetzt.

Dieses umfasst verschiedene Log-Tabellen (Collections), die direkt als REST-Endpunkt zur Verfügung stehen.

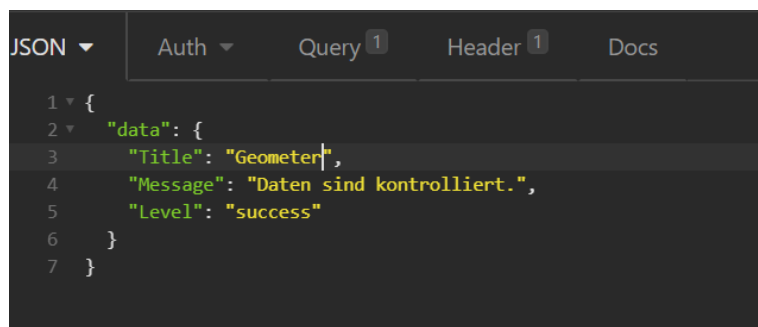


Für die Anzeige der Meldungen wurde eine eigene Webseite mit PHP und JavaScript erstellt.



### Meldung eintragen

Eine Meldung kann auf einem REST-Endpunkt mit einer Nachricht im JSON-Format erfolgen.



Beispiel URL für Endpunkte:

- Datenmeldungen
- Systemmeldungen

[https://myDomain.tld/api/collections/save/msg\\_datasets](https://myDomain.tld/api/collections/save/msg_datasets)  
[https://myDomain.tld/api/collections/save/msg\\_system](https://myDomain.tld/api/collections/save/msg_system)



## Bezug von Meldungen

Auf diesen REST-Endpunkten können die Meldungen jeweils wieder als JSON-Objekte bezogen werden.

Beispiel URL: [https://myDomain.tld/api/collections/get/msg\\_datasets](https://myDomain.tld/api/collections/get/msg_datasets)

```
Preview ▾ | Header 10 | Cookie | Timeline
1 {
2   "fields": { ↵ 3 ↵ },
26  "entries": [
27    {
28      "$type": "GEOBOX.IM.STDL.Plugin.Model.MessageBoard, GEOBOX.IM.STDL.Plugin",
29      "Title": "[POST] SDTL >> Daten",
30      "Message": "Daten - Gemeindedaten wurden gesendet.",
31      "Level": "info",
32      "_by": null,
33      "_modified": 1620368705,
34      "_created": 1620368705,
35      "_id": "6094dd415aab1c47c15eda95"
36    },
37    {
38      "$type": "GEOBOX.IM.STDL.Plugin.Model.MessageBoard, GEOBOX.IM.STDL.Plugin",
39      "Title": "[GET] geotopo >> Daten",
40      "Message": "Gemeindedaten wurden abgeholt.",
41      "Level": "info",
42      "_by": null,
43      "_modified": 1620627166,
44      "_created": 1620627166,
45      "_id": "6098cede9bd60bf8ff0cc421"
46    },
47  ]
}
```

Somit ist es frei, wo diese Meldungen angezeigt werden.



## GIS Arbeitsplätze

Die GIS Arbeitsplätze für den Prototyp wurden mit der Applikation «GEOBOX Amtliche Vermessung CH» basierend auf dem Autodesk AutoCAD Map 3D betrieben.

Für den Datenaustausch zwischen der Fachschale und dem «Web Service-Endpunkt» wurde ein eigenständiges Plug-In programmiert.

Das Plug-In ermöglichte es, direkt aus der GIS Applikation die Daten vom Web Service-Endpunkt zu beziehen und auch wieder abzugeben.

Die Daten wurden für die Bearbeitung in das GIS System eingetragen, also sozusagen importiert. Somit sind nach einem «Check-Out» die Daten auch im GIS vorhanden, unabhängig davon, ob der Web Service-Endpunkt weiterhin erreichbar ist oder nicht.

Die Fachschale wurde als sogenannte «dateibasierte Fachschale» verwendet, die aus technischer Sicht gesehen, eine eingelagerte SQLite Datenbank, die als Container-Objekt in einer DWG-Datei ist.

## Daten «Check-Out»

Im GIS System wurden, beim «Check-Out», die entsprechenden Objektklassen (Tabellen/Daten) geleert und die aktuellen Daten ab dem Web Service-Endpunkt gespeichert.

## Daten «Check-In»

Die Daten aus dem GIS wurden zusammengestellt und dem Web Service-Endpunkt übergeben.

## Datensätze Löschen

Damit der Übertrag (Daten «Check-In») aus dem GIS System auch die gelöschten Objekte übermitteln konnte, durften diese nicht aus dem Datenbestand (lokal) entfernt werden. Diese mussten über ein Attribut markiert werden und wurden beim Übertrag entsprechend auf dem zentralen Datenbestand verarbeitet.

## Geometrien

Die Geometrien wurden mit dem Plug-In jeweils vom SQLite-Spatial-Datentyp (GIS System interner Datentyp) in eine FGF-Geometrie (Feature Geometry Format) konvertiert.

Die FGF-Geometrie erweitert das WKB-Format (Well-known Binary).



## Weitere technische Überlegungen

Der Prototyp lässt vieles offen oder unbeantwortet. In diesem Abschnitt sind einige Gedanken und Konzepte für eine mögliche Implementierung.

### Zentrale Datenhaltung

#### Datenbank

Die gewählte Datenbank «MariaDB» ist für den Betrieb auf einem Standard Web-Hosting geeignet und wurde daher für den Prototyp gewählt.

Für eine produktive Implementation kann jedoch eine Datenbank eingesetzt werden, die durch die Web Server Komponente verwendet werden kann.

Beispielsweise: Oracle Database, PostgreSQL oder Microsoft SQL Server.

Die Wahl der Datenbank hängt von verschiedenen Faktoren ab, die beantwortet werden müssen:

- Lizenzen (Kosten und zulässige Betriebsmöglichkeiten)
- Datenmengen und Performanz
- Verfügbarkeitsanforderungen (Cluster, Synchrone Systeme, ...)
- Betriebsort und Betriebssystem
- Eingesetzte Programmiersprache der Webplattform

### Geometrien Handhabung

Im Prototyp wurde eine einfache Geometrie-Konvertierung gewählt, damit der Übertrag zwischen den beiden Komponenten verlustfrei bewerkstelligt werden konnte.

Der eingesetzte FGF-Datentyp eignet sich im Systemumfeld der Autodesk Software und ist nach erster Einschätzung unter dem Aspekt «Systemneutralität» eher ungeeignet.

Daher ist eine Definition für die Codierung der Geometrien für den Datenaustausch zu finden.

### Datenmodell

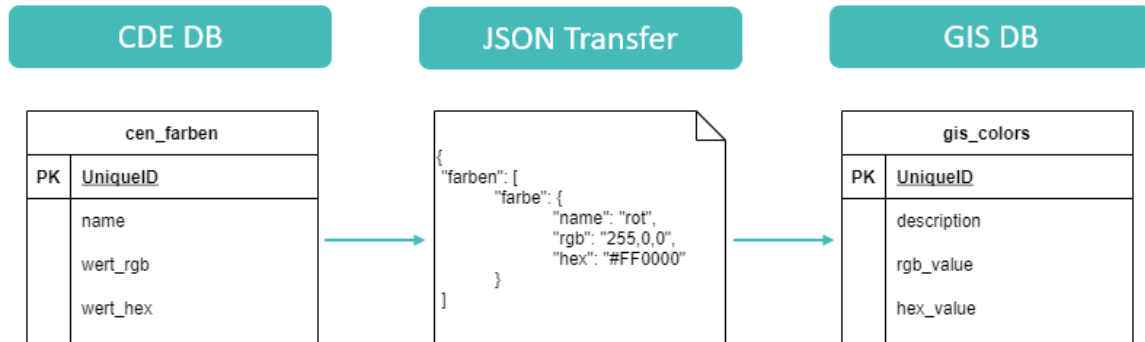
Innerhalb des Prototyps wurde das Datenmodell 1:1 zum Datenmodell der GEOBOX Fachschale (internes Datenmodell in der GIS Applikation) gewählt, dies aufgrund einer einfachen und raschen Umsetzung.

Ein Datenmodell, dass unabhängig von jedem GIS System ist und spezifisch auf die Anforderungen der zentralen Datenhaltung ausgelegt ist, wäre besser geeignet.

Auf die zentrale Datenhaltung sollte keine GIS Applikation oder Darstellungsdienst direkt zugreifen, sondern jeweils über den Web Service-Endpunkt die Daten beziehen und dann für die entsprechenden Aufgaben verwenden.

Eine Möglichkeit wäre beispielsweise, das Datenmodell nahe am INTERLIS-Modell (z.B. DM.flex) zu definieren und die entsprechenden Attributnamen zu übernehmen.

Eine Zuweisung (Mapping) der verschiedenen Beteiligten muss vom jeweiligen Systemanbieter ermöglicht werden.



Unter der Voraussetzung, dass der Datentransfer über eine REST-Schnittstelle angeboten wird, können die GIS Systemanbieter die entsprechenden Werkzeuge für Ihre Kunden anbieten. Dies beinhaltet auch, dass von jedem GIS Systemanbieter eine Zuweisung vom «JSON Transfer» in die interne Struktur gemacht werden muss, daher spielt es keine Rolle, ob die Struktur INTERLIS nahe oder komplett eigenständig ist.

## Web Service Endpunkte

### Programmiersprache PHP

Einfachheitshalber wurde PHP eingesetzt, damit der Prototyp zeitnah umgesetzt werden konnte.

Der Prototyp lässt offen, ob dies die geeignete Sprache und Umgebung ist. Es kann eine beliebige Technologie und Programmiersprache verwendet werden.

### Standard REST

Der Standard REST ist geeignet, um einen Datenaustausch anzubieten. Diese Technologie wird auch bei vielen weiteren Web-Services verwendet.

Als Transfer-Format wurde JSON (JavaScript Object Notation) verwendet.

Viele Applikationen und Programmierframeworks bieten direkte Unterstützung von diesen beiden Technologien an, dies kann ein positiver Einflussfaktor bei einer Umsetzung bezüglich Zeit (schnellere Umsetzung), Kosten (Einsparungsmöglichkeit), Benutzerakzeptanz und Datennutzung (vielfältigere Möglichkeiten) sein.

### Sicherheit der Endpunkte

Beim Prototyp wurde die Sicherheit der Endpunkte lediglich mit einem API-Key (Verbindungsschlüssel) gewährleistet.

Für eine Umsetzung müsste dies mit einer entsprechenden Systematik umgesetzt werden, welche den Anforderungen entspricht und die Sicherheit gewährleistet. Beispielsweise über ein Benutzeridentifikationsverfahren mit OAuth Standard.



## Versionierung der Endpunkte

Eine Entkopplung des Datenmodelles auf der CDE Datenbank und der Datentransferstruktur lässt eine Versionierung der Endpunkte zu, somit können Anpassungen am Datenmodell gemacht werden und über eine neue Version bereitgestellt werden.

Beispiel:

URL:	Datenlieferung:
<b>gisapi/v1/gemeinde</b> Liefert den Namen der Gemeinde.	«gemeinde»: { «NAME»: «Müllheim» }
<b>gisapi/v2/gemeinde</b> Ergänzung um die BFS-Nummer der Gemeinde.	«gemeinde»: { «NAME»: «Müllheim», «BFS»: «4831» }

## Datenqualität und Prüfungen

Die Überprüfung der Daten hinsichtlich Qualität und Konsistenz ist ein wichtiges Instrument, um einen wertvollen Datenbestand zu haben.

Im Prototyp wurde dies technisch **nicht** umgesetzt, respektive die Annahme getroffen, dass die Daten, die geliefert wurden, korrekt sind.

In Abhängigkeit zu den weiteren Anforderungen können verschiedene Lösungen zielführend sein:

- Weiternutzen der bestehenden Check-Services auf Basis INTERLIS.
- Eigene Prüfungen in der Plattform.

Eine zeitnahe Rückmeldung an die Benutzenden, wenn möglich in Echtzeit, wirkt sich unter Umständen positiv auf die Benutzerakzeptanz aus, da dies eine Verbesserung gegenüber der aktuellen Arbeitsweise darstellen kann.

## Überlegungen zu REST, JSON, XML und Interlis2

### Standard REST

REST kommuniziert über den Standard HTTP (Hypertext Transfer Protocol), also auf der Technologie, die im WWW und für die meisten Webseiten verwendet wird.

Das Protokoll ist in zwei Bereiche aufgeteilt, in den Kopf (Nachrichtenkopf, Header) und einem Nachrichtenrumpf (Body, teilweise auch als Payload bezeichnet).

Dies als einfache Umschreibung des Standards.

### Body als JSON, XML oder Interlis2

Bei einer REST Implementation wird die Definition des Bodys (Payload) selbst definiert, somit lässt der Standard zu, dass die Daten im Format JSON, XML oder Interlis2 übertragen werden.

Für eine hohe Benutzerakzeptanz, sowie für eine möglichst breite Datennutzung wäre ein Standard wie JSON oder XML zu bevorzugen. Diese werden in der Regel besser durch Programmiersprachen und Frameworks unterstützt.



Dies auch unter dem Aspekt, dass weitere Applikationen (nicht GIS Systeme) diese Daten verwenden können.

Wenn weitere Informationsebenen angeboten werden, sind auch Szenarien wie die Nachführung von Daten mit Mobilien Applikationen denkbar.

### Nachteil gegenüber Interlis2

Nachteile von JSON und XML gegenüber von Interlis2 ist die Definition der Geometrie, diese ist im INTERLIS Standard exakt definiert.

Für die weiteren Formate müsste dieser Umstand genau untersucht werden, es können verschiedene Lösungsansätze zielführend sein. Beispielsweise die INTERLIS Definition in eine JSON Notation überführen oder die Geometrie mit einem passenden Standard (GeoJSON, KML/GML, OGC-Standard, ...) definieren.

### Kontoverwaltung und Berechtigungssystem

Innerhalb des Prototyps wurde auf eine Kontoverwaltung und Berechtigungssystem verzichtet. Dies ist aber für eine produktive Umgebung Pflicht.

Bei der Kontoverwaltung kann ein System auf Stufe «Unternehmen» oder bis zum eigentlichen Mitarbeitenden der jeweiligen Unternehmen gewählt werden.

Für die Umsetzung müssten die Anforderungen klar definiert werden, damit das geeignete System erstellt und implementiert werden kann.

Für das Berechtigungssystem fliessen folgende Gedanken ein:

- Wer darf was auf den Daten (lesen, schreiben, löschen)?
- Müssen diese Berechtigungen auf geografische Gebiete eingeschränkt werden?
- Gibt es zu den Berechtigungen lesen und schreiben eine Eingrenzung in den Datenmengen?  
Beispielsweise lesen = immer alle Daten, schreiben = nur die mir Zugewiesenen
- Gibt es eine spezielle Berechtigung für Datenverifikationen und/oder Datenfreigaben?

### Eigene Zugangsverwaltung vs. Drittanbieter

Eine eigene Zugangsverwaltung mit den entsprechenden Berechtigungen muss sowieso innerhalb der Lösung erstellt werden.

Für die Anmeldung durch die Benutzenden (Authentifizierung) kann aber auf einen Drittanbieter zurückgegriffen werden, beispielsweise die SwissID.

### Messages - Benachrichtigungen

Die Lösung der «Messages» ist mit einer einfachen Plattform und der Anzeige der Meldungen im Prototyp gelöst.

Die Prozesse (Use Cases) zeigen auf, dass die Benachrichtigungen eine hohe Komplexität aufweisen und mit diversen Fallunterscheidungen in den Einsatz kommen müssen.

Für die Realisierung einer möglichen Lösung müssten noch einige Anforderungen definiert werden.

### Technische Möglichkeiten

Neben der einfachen Anzeige sind einige weitere technische Möglichkeiten anzuschauen:

- Meldungen per E-Mail versenden.





- Meldungen per eCH Standards ausgeben. (Beteiligte Verwaltungssysteme?)
- Weitere Möglichkeiten wie RSS-Feed zum Abonnieren, REST Endpunkte für den Bezug durch weitere Applikationen, ...
- Meldungen per BCF (BIM Collaboration Format).

### Lesebestätigung, Meldungen Quittieren

Je nach Anforderung sind Überlegungen für die Bestätigung von Meldungen (Lesebestätigungen) zu machen.

### Prozesse und Meldungen

Die Fragestellungen:

- Wer wird benachrichtigt?
- Was wird benachrichtigt?
- Zu welchem Zeitpunkt wird benachrichtigt?

müssen durch die Prozesse beantwortet werden und durch eine allfällige Messagekomponente entsprechend abgebildet werden können.

Das Benachrichtigungsverfahren hat einen direkten Einfluss auf die Anforderungen der Kontoverwaltung.

### Handhabung der Informationsebenen

Aktuell sind bezüglich der Amtlichen Vermessung die Informationsebenen vorgegeben, welche teilweise in direkter Abhängigkeit zueinander definiert sind.

Als Beispiel die Definition der «Liegenschaft»: Im INTERLIS Datenmodell ist bei der Geometrie der Hinweis bezüglich den gültigen Stützpunkten beschrieben::

- Geometrie nur LFP1, LFP2, LFP3, Grenzpunkte oder Hoheitsgrenzpunkte

Weitere Vorschriften sind bezüglich Liegenschaftsbeschriebe vorhanden, die mit Bemerkungen oder Anmerkungen verlangen die LFP aufzuführen.

In Bezug auf die Datenlieferungen an die Grundbuchsysteme mittels AVGBS gibt es weitere bekannte Abhängigkeiten.

Beispiele für die Liegenschaften mit den Hoheitsgrenzpunkten:

- Wie werden diese überprüft? (Topologieprüfung im GIS System, im INTERLIS)
- Muss das GIS System weiterhin in der Lage sein, diese Informationen als INTERLIS abzugeben, weiterzugeben?
  - Wer übernimmt welche Verantwortlichkeiten?
  - Gibt es entsprechende Entschädigungen für Aufwendungen?
- Bezieht das Grundbuchsystem diese Information direkt von der zentralen Datenhaltung und generiert es die Information selber?

Hinsichtlich einem Ausbau der zentralen Datenhaltung mit weiteren Themen stellen sich diese Fragen immer wieder.

- Beispiel PLZ Ortschaften



- Liegenschaftsbeschriebe
- AVGBS Lieferungen
- ...

Trennung der Informationsebenen als mögliche Lösung?

Könnte zielführend sein, dass beispielsweise gemeinsame Stützpunkte der Liegenschaftsgeometrie in jeder Informationsebene erfasst werden.

Dies würde zu einer Möglichkeit führen, dass jede Informationsebene jeweils komplett und eigenständig erfasst, geprüft und übertragen (z.B. Interlis2-Export) werden kann. Führt aber zwangsläufig zu redundanten (doppelten) Daten.

Der Umstand der Redundanzen könnte mit einem Top-Down Prinzip entschäft werden, als Beispiel der Hoheitsgrenzpunkt ist mit allen geforderten Attributen erfasst und gilt als Top Element. Alle weiteren in Abhängigkeit stehenden Punkte, wie beispielsweise der Grenzpunkt ist lediglich mit der Geometrie für die topologischen Prüfungen und einer Referenz auf den Hoheitsgrenzpunkt eingetragen.

Ob dieser Lösungsansatz, für alle Abhängigkeiten anwendbar ist, wäre zu prüfen.

### Abgrenzung und Abdeckung von Nutzen

Eine Definition für die Abgrenzung der Aufgaben der zentralen Datenhaltung muss klar definiert werden.

Im Zusammenhang mit den Datenbeständen steht immer eine Nutzung der Daten. Daher ist eine Abgrenzung wichtig, um auch den Umfang klar zu definieren.

Beispielweise beinhaltet die zentrale Datenhaltung auch Darstellungsdienste wie WMS? Oder wird diese durch einen unabhängigen Dienst oder Service bereitgestellt, der lediglich die Daten bezieht und aufbereitet.



## Anhang

Bild: JSON-Objekt Methode GET

```
{
  "sender": "Zentrale Datenbank",
  "jobname": "Status genehmigt",
  "featureclasses": [
    { ← 2 → },
    { ← 2 → },
    {
      "featureclass": "LM_AD_MUNICIP_BOUNDARY",
      "rows": [
        {
          "values": {
            "ID": "1",
            "GEOM": "AQAAAAEAAACoxkvX7MREQYGVQ4vOcTNBAAAAAAAAA=",
            "ORIENTATION": "133.802",
            "AREA": null,
            "DATE_LEGALIZED": "01.04.2021 12:43:01",
            "FID_AD_MUNICIPALITY": "356",
            "FID_AD_MUTPERIMETER": null,
            "ILI2_OID": "ch21yubs8UQOfukn",
            "ROW_DELETED": "0",
            "ROW_VERSION": "1",
            "UFID": null
          }
        }
      ]
    }
  ]
}
```



## Bild: JSON-Objekt Methode POST

```
"featureclasses": [  
  {  
    "featureclass": "LM_AD_MUNICIPALITY",  
    "rows": [ ↵ 2 ➡ ]  
  },  
  {  
    "featureclass": "LM_AD_MUNICIP_BOUNDARY",  
    "rows": [  
      {  
        "values": {  
          "GEOM": "x'0100000000000000a8c64bd7ecc444418195438bce713341'",  
          "ORIENTATION": "133.802276324186",  
          "DATE_LEGALIZED": "01.04.2021 12:43:01",  
          "FID_AD_MUNICIPALITY": "356",  
          "ILI2_OID": "ch21yubs8UQOfukn",  
          "ROW_DELETED": "0",  
          "ROW_VERSION": "1"  
        }  
      },  
      {  
        "values": {  
          "GEOM": "x'0100000000000000f2d24d9255bb44419eefa7a6656e3341'",  
          "ORIENTATION": "133.802276324186",  
          "DATE_LEGALIZED": "01.04.2021 13:14:15",  
          "FID_AD_MUNICIPALITY": "355",  
          "ILI2_OID": "ch21yubs8UQOfukS",  
          "ROW_DELETED": "0",  
          "ROW_VERSION": "1"  
        }  
      }  
    ]  
  },  
]
```