# **Zusammenfassung**

Folgende Schritte sind in der folgenden Reihenfolge abzuarbeiten.

Die Ausgangslage, techn. Zugriffe/Hilfsmittel, offene Fragen und Herausforderungen werden in den jeweiligen Abschnitten zu den einzelnen Schritten dann detailliert.

Schritte:

1. Quellinformationen für 1 Bauprojekt beschaffen, Syntax studieren, wfs-Dienst verstehen, Filtern lernen im Aufruf wfs  
   🡪 erstes Ergebnis: alle Informationen zu einem Bauprojekt per wfs beschaffen!
2. Zielinformationen für eine Übertragung von 1 EIV mit KOMs in neuer REST-Schnittstelle verstehen, neu anlegen, überschreiben und löschen, Rückantworten verstehen (Fehlermeldungen)  
   🡪 erstes Ergebnis: alle Informationen eines Bauprojektes in KSP übertragen
3. Übertragung von Quelle zu Ziel exemplarisch machen, verallgemeinern, parsen, GEo-Herausforderungen lösen   
   🡪 Ergebnis: 1 zu 1 – Übertragung implementieren (wahrscheinlich mit GeoTools)
4. Algorithmus für tägliche Übertragung fachlich formulieren und implementieren.
5. Anforderungen für Dokumentation und Monitoring und manuelle Eingriffe formulieren
6. Schnittstellen-Monitoring-Anwendung implementieren
7. Erste Übertragung aller KSP-ready-Baumaßnahmen ins KSP   
   (Ziel: bis 31.12.2024)
8. Aufnahme Betrieb der fortlaufenden Schnittstelle (Ziel: bis 1.3.2024)

# **Quelle wfs FlistraNeo**

## Fachliche Ausgangsanforderungen

In FlistraNeo hingegen finden wir

* Bauprojekte (Geometrie + Attribute), denen
* Maßnahmen (in einem Maßnahmestand) zugeordnet sind   
  (ohne Geometrie, aber mit Attributen zu Ausgangs- und Zielzuständen), denen wiederum
* Teilmaßnahmen zugeordnet sind (mit Geometrien, z.T. als Teilflächen)

(Stand bei den Altdaten – ist das anders bei aktuellen Datensätzen?)

Bei den Altdaten wurde die Teilflächenbildung autom. erzeugt wurde, d.h. die Teilflächen, wie der Name schon sagt, gehören zu einer Maßnahme.

LBM arbeitet in FLISTRAneo mit **3 Maßnahmenständen** für **Maßnahmen**:

* Baurecht
* 🡪 Ausführung
* 🡪 Bestand

Zu beachten ist, dass bei den Altdaten keine Geometrie für die Teilmaßnahmen abgegrenzt wurde. Diese Variante würde daher nur für die „Neueingaben“ funktionieren.

## Technischer Zugriff auf Server, um WFS von FlistraNeo zu erreichen

Roi Zugang erstellen, der über eine Citrix Plattform Putty, Winscp und einen Browser den Zugang bereit stellt. Android App nötig, und den im beiliegenden PDF (roisoft.pdf) beschriebenen Prozess durchlaufen.

Registrierungscode (Seite 4 des PDF, z. B. 03245-03267) ans LDI schicken (zum entsperren).

Sie können die folgenden Schritte in der PDF dann trotzdem schon abschließen, der Token funktioniert dann allerdings erst wenn ich ihn freigegeben habe.

Bsp. Daten Hr. Hessari für die Einrichtung der App:

Serial Number: 74680-61986

Activation Code: 9685-7295-3648-6159

Sobald der Token entsperrt ist können Sie sich dann wie in der PDF (Einwahl.pdf) beschrieben ist im LBM Netz anmelden und auf die oben beschriebenen Apps zugreifen. Ihr Benutzername ist [HessariBi@lsv.intra](mailto:HessariBi@lsv.intra) und das Initialpasswort ist \*\*\*\*\*

Registrierungscode melden:

Nach Aktivierung des Token:

Sobald Sie sich damit in citrix anmelden müssten Sie die Apps für Putty und Winscp sehen können. Auf dem Debian Server 10.100.210.227 habe ich schon einmal Postgresql, Postgis und JRE installiert (alles noch unkonfiguriert).

 Ihre Benutzerdaten sind:

Benutzername: hessari Passwort: \*\*\*\*\*\*\*\*

 Sie sind auch in sudo eingetragen damit Sie Root Rechte erhalten können.

WFS-Aufrufe die wir von der LISt erhalten haben:

Nutzer: gdi

Passwort: \*\*\*\*\*\*\*\*\*\*

## Technischer Zugriff auf WFS

Beispielaufraufe alle Bauprojekte, alle Teilmaßnahmen:

<https://flistra.lbm.rlp/geoserver/gdi/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=gdi%3Aksp_bauprojekt&maxFeatures=50&outputFormat=application%2Fjson>

<https://flistra.lbm.rlp/geoserver/gdi/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=gdi%3Aksp_teilmassnahme&maxFeatures=50&outputFormat=application%2Fjson>

Beispielaufraufe ein konkretes Bauprojekt:

<https://flistra.lbm.rlp/geoserver/gdi/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=>gdi%3Aksp\_bauprojekt&outputFormat=application%2Fjson&Filter=<Filter><PropertyIsEqualTo><PropertyName>bp\_id</PropertyName><Literal>{32ade62e-1719-74df-e200-39e897c580c2}</Literal></PropertyIsEqualTo></Filter>  
  
<https://flistra.lbm.rlp/geoserver/gdi/ows?service=WFS&version=1.0.0&request=GetFeature&typeName=>gdi%3Aksp\_bauprojekt&outputFormat=application%2Fjson&Filter=<Filter><PropertyIsEqualTo><PropertyName>projekttext</PropertyName><Literal>A1 (Wittlich) - B50a (Platten)</Literal></PropertyIsEqualTo></Filter>

**Ergebnis von einem Bauprojekt :**

{"type":"FeatureCollection",

"features":[{

"type":"Feature",

"id":"ksp\_bauprojekt.fid--6e66dbeb\_1924818a551\_45c3",

"geometry":{"type":"Point","coordiates":[348465.59914327,5535018.9602426]},

"geometry\_name":"geometrie",

"properties":{

"bp\_id":"{32ade62e-1719-74df-e200-39e897c580c2}",

"projekttext":"A1 (Wittlich) - B50a (Platten)",

"strasse":"B050",

"aktenzeichen":"02.2-1465-P/30/34/39/42",

"genehmigungsbescheid":"2000-12-27Z",

"bestandskraftdatum":"2003-07-07Z",

"verfahrensart":"Planfeststellung",

"verfahrensart\_kspnr":743743261,

"verfahrennach":null,

**"uebergabe\_ksp":"ja",**

"bearbeitetam":"2024-04-24Z",

"geo\_id":"{ff0470db-aed4-da5d-51dc-a7dac773d940}",

"geo\_bearbeitetam":null}

}],

"totalFeatures":1,

"numberMatched":1,

"numberReturned":1,

"timeStamp":"2024-10-03T17:07:06.402Z",

"crs":{"type":"name","properties":{"name":"urn:ogc:def:crs:EPSG::25832"

}}}

**Hinweise zur Filterkonstruktion:**

Das Prinzip ist eigentlich einfacht, man hängt einen Parameter "&Filter=" an die URL, der einen XML-Auszug nach der [OGC-Filter Spezifikation](https://www.ogc.org/standard/filter/) enthält (grün). In diesem Fall einfacher Vergleich. Der <PropertyName> gibt das WFS-Attribut bzw. Spaltennamen an, das <Literal>-Feld den zu suchenden Text. Man kann auch komplexere Filter bauen, das würde man aber sinnvollerweise über eine Post-Request mit einem WFS-Query xml machen.

Wichtig ist, dass ihr diese Abfragen nicht selber bauen müsst, das mach alles Geotools für euch. Ihr definiert dort einen Filter mit CQL.toFilter("bp\_id={32ade62e-1719-74df-e200-39e897c580c2}") und GeoTools macht dann daraus einen WFS-Filter oder ein SQL-Abfrage, je nachdem, was die Datenquelle ist (WFS oder Datenbank). D.h. man kann denselben Filter erst zur Postgis-DB schicken und wenn da noch nichts vorliegt, sich die Daten mit dem Filter vom WFS holen.

## toDo: Extraktion eines Bauprojektes mit allen zugehörigen Maßnahmen und Teilmaßnahmen und den Meta-Informationen

🡪 macht Hr. Boden in KW 44   
 (bitte auch neuen Zugriff für Server einrichten für Marcus Skowronek, ms@redlark.de)

# **Ziel KSP**

## Fachliche Ausgangsanforderungen

Die Fachlichen Anforderungen für die Attributierung von KOMs, EIVs und OEKs sind bekannt. Die Schnittstelle erlaubt die Neuanlage von Objekten, den update (komplettes überschreiben), das Löschen von Objekten.

## Technischer Zugriff auf die REST-Schnittstelle

**Zugriff über postman auf TEST-KSP:**

Im Abschnitt Headers zwei zusätzliche Einträge "Ksptoken" & "Kspuser" mit den credentials eintragen:

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Beispiel Body abgeschickt.

Bei Erfolg  bekommst du als Antwort eine id zurück, die für weitere GET uns PUT-Einträge verwenden kannst.

Beispiel id=73a65082-8f21-4a88-8adc-a0dd17a28f65

<https://konova.test.naturschutz.rlp.de/api/v1/intervention/73a65082-8f21-4a88-8adc-a0dd17a28f65>

Wenn diese URL mit einem GET-Request aufgerufen wird, kommt der übertragene Eingriff zurück.

{ "type": "MultiPolygon",

"coordinates": [

[ [ [ 7.593955993652341, 50.36244809879689 ],

[ 7.592582702636717, 50.36023051515941 ],

[ 7.601766586303708, 50.3582318633635 ],

[ 7.603225708007811, 50.3628587510637],

[ 7.599148750305174, 50.362557606415464],

[ 7.596273422241207, 50.362612360130036],

[ 7.593955993652341, 50.36244809879689] ]] ],

"properties": {

"title": "A1 (Wittlich) - B50a (Platten)",

"responsible": {

"registration\_office": "",

"registration\_file\_number": "",

"conservation\_office": "",

"conservation\_file\_number": "",

"handler": {

"type": "",

"detail": ""}},

"legal": {

"registration\_date": **null**,

"binding\_date": **null**,

"process\_type": **null**,

"laws": []}}}

Wenn am GeoJson nur eine Property geändert wird und diese mit einem PUT-Request an die o.g. URL sendest, wird die Änderung übernommen.

**Dokumentation KSP:**

**·** [**api v1 compensation post\_de - konova - Naturschutz RLP**](https://git.naturschutz.rlp.de/IT-Naturschutz/konova/wiki/api-v1-compensation-post_de)

**·** [**api v1 compensation get\_de - konova - Naturschutz RLP**](https://git.naturschutz.rlp.de/IT-Naturschutz/konova/wiki/api-v1-compensation-get_de)

**HINWEISE:**

Beachtet auch den Unterschied zwischen dem POST und dem GET Inhalt lt. Dokumentation. In den Beispielen seht ihr bspw., dass bei GET die Biotoptypen auch human-readable zurückgegeben werden. Die Info muss natürlich beim POST nicht angegeben werden. Dort reicht die Liste der Biotop AtomIDs. Ähnlich auch bei den Maßnahmen (actions).

Zwei GET responses der genannten Einträge hinten im Anhang.

EIVs und zugehörige KOMs in einem zu posten war ein erster Gedanke.

Aufgrund der daraus entstehenden Größe der einzelnen POST bodies (besonders bei solchen Projekten mit ein paar Dutzend KOMs hinten dran), ist die maximale Größe eines POST body nur schwer einzuschätzen und entsprechend schwammig in der Webserverkonfiguration zu erlauben. Da es ohnehin die Möglichkeit brauchte nachträglich KOMs hinzuzufügen, ist dies der elegantere Weg.

Nicht nur komplette Datensätze können übertragen werden, sondern auch unvollständige: Mindestumfang ist eine Geometrie und die Festlegung, ob es sich um eine EIV oder um ein KOM handelt.

## Fachliche Auswirkungen auf KSP

* Unvollständige,
* vollständige und
* freigegebene Datensätze

werden in unterschiedlichen layern im LANIS angezeigt.

So ist jeder übertragene Datensatz sichtbar im KSP und darf durch das LBM ohne Kenntnisnahme der uNB (bis zur Freigabeerteilung - bestandskräftig) weiter über die Schnittstelle aktualisiert werden.

Erst wenn der Maßnahmestand „freigegeben“ erreicht wird, kann die Schnittstelle nicht mehr wie bisher genutzt werden! Nun muss das Einverständis der uNB erwirkt werden (Freischaltung zur Aktualisierung), um noch weitere Änderungen nachzutragen.

Es gilt also zu unterscheiden

1. „unvollständige (Altfälle)“
2. „In Bearbeitung“, d.h. vollständige Fälle, aber noch nicht freigegebene (bestandskräftig) durch ZB
3. „nicht mehr änderbar“, d.h. Vollständige, freigegebene und bestandkräftige Fälle (Status BESTAND in Flistra)

Veranschaulichung der Spielregeln am Beispiel der EIVs (**ETS=uNB**)

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

1. Die oberste Ebene **EIV** ist die reguläre Ebene. Alle EIV-Objekte können diese erreichen. Diese Ebene wird nur erreicht, wenn
   * die **ZB eine Freigabe** dazu **erteilt** (das macht LBM zukünftig über die Schnittstelle),
   * der **Datensatz vollständig** ist (oder durch die ETS vollständig gemacht wird - das ist möglich nach Freigabe) und
   * die **ETS** den Datensatz letztendlich auch "**verzeichnet**" (nur möglich nach Freigabe).
2. Über die Schnittstelle landen LBM-Beiträge (im KSP, im LANIS) zunächst immer **in einer der beiden unteren Ebenen**:
   * Genehmigungsdatum eingetragen vor dem 12.06.2018  --> dann landet es in "unvollständige Altfälle"
   * kein Genehmigungsdatum eingetragen oder Genehmigungsdatum nach dem 12.06.2018 --> dann landet es in "in Bearbeitung"
3. Ein Übergang zwischen den beiden unteren Ebenen erfolgt also nur dann, wenn man am "Genehmigungsdatum" was verändert!
4. Ein Übergang in die obere Ebene erfolgt, wenn die Daten durch die ETS verzeichnet werden - dies geht nur,
   * wenn freigegeben und
   * der Datensatz vollständig ist.
5. Nach einer Freigabe hat auch die ETS die Möglichkeit, Daten zu ändern oder zu vervollständigen   
   (auch wenn das LBM das eigentlich nicht will).  
   Die ZB hat dann erst wieder **schreibenden Zugriff auf die Daten**, wenn die Daten von der ETS wieder "rückgestuft" wurden   
   (d.h. "verzeichnet" wird zurückgenommen).

# Transition Fachdaten (exemplarisch)

Einmalig können sukzessive alle Datensätze aus Flistra sukzessive ausgewählt und ins KSP erneut übertragen werden. Hierzu sind die bereist übertragenen Datensätze zu identifizieren und zu löschen.

Im folgenden werden Details und Hinweise gegeben, wie eine Baumaßnahme in ein EIV umgewandelt werden muss.

## Ausgangslage / Herausforderungen

Das KSP akzeptiert nur

* Multipolygone (während Flistra alle möglichen Geometrien (Punkte, Linien, Polygone) liefert).
* EIVs und KOMs

Flistra liefert Bauprojekte (BP), Maßnahmen (MA) und Teilmaßnahmen (TM).

Die Transformationsvorschrift ist grob:

* BP werden EIVS (aus Punkt wird Kreis mit Radius x)
* MA werden KOMs
* TM definieren die KSP-Maßnahmen in den KOMs

Frage an die Fachseite:

Wird **jede Teilmaßnahme (mit eigener) Geometrie ein eigenständiges KOM-Objekt?  
Wird jede Maßnahme ein KOM-Objekt und die Geometrien und der Teilmaßnahmen werden zusammengeführt (so wurde es bei Altdaten zwangsläufig gemacht)?**

## Tools

Ich würde das [Geotools-](https://docs.geotools.org)Framework verwenden, das eigentlich alle relevanten Funktionen liefert. Ein großer Vorteil dieses Paketes ist, dass Datenquellen (DataStore)   abstrahiert werden, so dass es egal ist, ob die Geodaten (Features)  aus einer DB (Postgis), einer Datei (Geojson) oder einem Dienst (WFS) kommen.

Geotools ist modular aufgebaut, bitte nicht das komplette Framework einbinden, sondern nur die benötigten Dependencies nutzen.

Geotools findet selbst den passenden [Datastore](https://docs.geotools.org) aufgrund einer Parameter-Map, die am besten über eine externe Properties-Datei für den [WFS](https://docs.geotools.org/latest/userguide/library/data/wfs-ng.html), oder die [Postgis](https://docs.geotools.org/latest/userguide/library/jdbc/postgis.html)-Datenbank bereitgestellt werden (So lassen sich die Einstellungen für Entw, Test und Prod am einfachsten anpassen).

Die Datastores stellen dann für die einzelnen Datenebenen (Layer bzw. Tabellen) je eine [FeatureSource](https://docs.geotools.org/latest/userguide/library/data/featuresource.html) bereit, von der dann die Datensätze (Feature) als Liste (FeatureCollection) abgefragt werden können. Für die Abfrage wird mit [CQL](https://docs.geotools.org/latest/userguide/library/cql/index.html) eine universelle Abfragesprache bereitgestellt, über die dann zum Beispiel nach Datum gefiltert werden kann. Der CQL-Ausdruck wird dann automatisch in eine Filter-Abfrage für WFS oder Postgis umgesetzt. Der Filter wird an ein [Query](https://docs.geotools.org/latest/userguide/tutorial/filter/query.html)-Objekt übergeben, bei dem dann das Ziel-Koordinatensystem des KSP  mit  *setCoordinateSystem​(CRS.decode("EPSG:4326")) gesetzt werden muss.*

Die FeatureSource liefert außerdem die Daten-Struktur der/des zugrundeliegenden Tabelle/Layer als [FeatureType](https://docs.geotools.org/latest/userguide/library/api/type.html)  zurück. Damit lassen sich dann neue leere Features für KOMs und EIVs erzeugen, die dann mit den Werten aus den Flistra-Features gefüllt werden müssen.

*GeoTools nutzt das Geometrie-Framework* [*JTS*](https://projects.eclipse.org/projects/locationtech.jts) *für die Geometrien. Mit JTS kann man jede Geometrie über die buffer()-Funktion in ein Polygon umwandeln. Dann erzeugt man ein neues* [*MultiPolygon*](http://locationtech.github.io/jts/javadoc/org/locationtech/jts/geom/MultiPolygon.html#MultiPolygon-org.locationtech.jts.geom.Polygon:A-org.locationtech.jts.geom.GeometryFactory-)*, dem übergibt das gepufferte Polygon als Array übergibt. Die GeometryFactory erhält man über die getFactory()-Methode des Polygons.*

Ist die Quelle writable (wie im Fall der Postgis-Tabellen), ist die zurückgegeben FeatureSource vom Typ FeatureStore, der entsprechende Schreib-Operationen bereitstellt.

GeoTools leistet nicht die Übertragung an die KSP-API.

Die Umgewandelten KSP-Objekte dann mit dem Ergebnis der KSP-Übertragung über den FeatureStore in die Postgis-Datenbank schreiben.

Man könnte auch versucht sein, einfach das erzeugte GeoJson als CLOB in die Datenbank zu schreiben. Davon würde ich aber abraten. Mit den GIS-Tabellen hat man die Möglicheit, z.B. mit QGIS darauf zu zugreifen und so den Status und evtl. Fehlerfälle zu visualisieren, ohne dass ein extra Tool dafür entwickelt werden müsste.

## Probleme:

Da wir keine direkte Entwicklungsumgebung haben, wird es schon Zeitintensiv sein, lokal zu entwickeln und den Code dann auf den LBM-Server zu übertragen und zu testen.

Man kann sich hier mit der freien Software [Geoserver](https://docs.geoserver.org/latest/en/user/) behelfen, die auch auf GeoTools basiert und u.a. WFS-Dienste bereitstellt. Wenn man die Flistra-WFS-Layer mit OUTPUTFORMAT=SHAPE-ZIP abfragt, kann man diese als Datenquelle in einer lokalen Geoserver-Installation als Quelle angeben und den Flistra-WFS damit simulieren, indem man die Datastore-Properties (s.o.) anpasst.

GeoTools ist sehr umfangreich und mächtig. Man muss keine eigenen GeoTools-Klassen, wie Builder oder Factories, implementieren. Alles was wir brauchen, ist über einfache Methodenaufrufe zu erreichen.

# Übertragungsalgorithmus (generelles Verfahren)

Es wird folgende Arten der Schnittstellennutzung geben:

1. Neuübertragung
2. Aktualsierung
3. Löschung
4. Freigabeerteilung

Täglich werden alle Datensätze aus Flistra ausgewählt, die entweder noch nicht übertragen oder aktualisiert wurden (relativ nach der letzen erfolgreichen Übertragung).

Fachlich ist festzulegen:

Wann übertragen wir (für) ein Bauprojekt mit allen (Teil-)Maßnahmen (kurz **BmaM**) ….

1. **erstmals**  --> wenn das KSP-Häckchen (KSP-fertig) in Flistra gesetzt ist.
2. **erneut** --> wenn es einen "Änderungshinweis" aus Flistra-Neo gibt (oder wir das in der Schnittstelle rausrechnen, z. B.    
   immer dann, wenn eine letze Änderung im Datensatz nach der Übertragung ans KSP vorgenommen wurde.)
3. als **freigegeben** zur "Verzeichnung durch eine ETS" --> wenn in Flistra das Attribut (Häckchen) ??? gesetzt ist (oder irgendein DATUM)
4. als **nicht mehr freigegeben** (Freigaberücknahme ist möglich, solange der Eintrag durch die ETS noch nicht verzeichnet wurde)
5. als **gelöscht**.

Alle Änderungsarten werden durch die Schnittstelle für die uNB nachvollziehbar im KSP im Datensatz dokumentiert – auch wenn dieser noch nicht freigegen ist (In Bearbeitung).

Bei der wfs-Abfrage der Flistra-Daten ist zunächst zu ermitteln, ob das Häckchen „KSP-ready“ gesetzt ist. Zudem ist zu prüfen, ob eine Aktualisierung (festgemacht am Aktualsierungsdatum) erfolgt ist nach der letzten erfolgreichen Übertragung.   
Dies sind z. B. alle Datensätze (Baumaßnahmen), deren letztes Aktualsierungsdatum später ist als das *letzte* Übertragungsdatum.

Bsp für eine Übertragung am 21.10.2024:

* + Letze Übertragung: 19.10.2024, 2h A.M.
  + Filtere alle Baumaßnahmen, deren letzte Aktualsierung **größer oder gleich** der letzten Übertragung (hier: 19.10.) ist.

Beachte: wenn es einen Übertragungsversuche gab, der „fail“ war (z. B. keine gültige Geometrie), dann gibt es eine Rückmeldung dazu. Dann muss in Flistra die Geometrie angepasst werden, das erzeugt ein neues Aktualsierungsdatum und dann wird der Datensatz erneut gefiltert und übertragen.

**1. Neuübertragung**

Projekte/ Einzelmaßnahmen, die „**"uebergabe\_ksp":"ja"**“ markiert sind, aber noch nicht ins KSP übertragen wurden, sind zu identifizieren. Dies erreichen wir durch eine Abfrage im KSP (gibt es schon ein EIV mit dieser Baumaßnahme-ID?).

**2. Aktualisierung**

Ergibt eine Abfrage im KSP (gibt es schon ein EIV mit dieser Baumaßnahme-ID?), das diese Baumaßnahme bereits übertragen wurde, so wird der Datensatz über die Schnittstelle nicht neu erzeugt, sondern aktualisiert.

**3. Löschung**

Projekte/ Einzelmaßnahmen, die mal eingegeben wurden, aber nie in die Umsetzung gingen, werden in FLISTRAneo gelöscht. Hier muss auch gewährleistet sein, dass sie auch aus KSP (damit auch aus LANIS) gelöscht werden!

Wer identifiziert / kommuniziert die gelöschten Baumaßnahmen, wie genau?

**4. Freigabeerlaubnis (Bestandskraft erreicht)**

Wer identifiziert / kommuniziert die gelöschten Baumaßnahmen, wie genau?

----------------

1. Wir müssen fachlich die Indikatoren für die verschiedenen Übertragungsarten ermitteln.
2. Täglich wird eine Liste der „heute **zu übertragenden Bauprojekte“** per wfs abgefragt und lokal als Datei abgespeichert. Dazu braucht es ein „letzes Übertragungsdatum“
3. Täglich wird die Übertragung aller Listenelemente ins KSP veranlasst.
4. Die Rückmeldungen werden ebenfalls lokal als Datei abgespeichert.
5. Dann wird das „letzte Übertragungsdatum aktualisiert.

Zuruf oder durch rücknahme der „besandskräftigkeit“.  
Diskussion: abstimmungsbedürftige Korrekturen und nicht abstimmungsbedürftige Korrekturen (z.B. kleine Korrekturen von Geometrien [Anpassen auf Flurstücksgrenzen], Ergänzung von Pflichtfeldern für KSP).

# Anforderungen an Dokumentation, Rückmeldungen, Monitoring, manueller Eingriff

< todo>

# Implementierung Anwendung „flistra2ksp“

Vorgesehen ist eine kleine Anwendung für ausgewählte Nutzer, um

* die Einzelübertragung einzubetten,
* alle Übertragungen zu dokumentieren (Übertragungszeitpunkt, Resultat)
* Rückmeldungen zu initiieren, wenn Fehler auftreten
* Manuelle Eingriffsmöglichkeiten vorzusehen, um Übertragungen zu wiederholen
* ……

Anwendung in react (JavaScript), mit Node JS ….

# Erstimport zum Stichtag 31.12.2024

# Implementierung Anwendung 31.3.2025