

Projektdokumentation

Maschinenlesbare Bereitstellung von
Pflanzenschutzmittelzulassungsdaten
für die Anwendung im Precision
Farming



Pflanzenschutzmittelzulassungsdatenbank

Executive Summary/Case Study	4
1 Pflanzenschutzmittelzulassungsdatenbank	5
1.1 Ausgangslage	5
1.2 Über Tech4Germany	6
1.3 Arbeitsweise	6
2 Analyse	8
2.1 Kern Fragestellungen	8
2.2 Vorgehen	8
2.3 Erkenntnisse aus den Expert:inneninterviews	9
3 Konsolidierung und Erkenntnisse	9
3.1. Systemkarte	9
3.2. Marktübersicht	11
3.3. Anwender:innenübersicht	13
3.4. Personas	14
3.5. User-Journey - Status Quo	16
3.6. Kernerkenntnisse	17
3.7. Priorisierung der Handlungsfelder	21
4 Ideenfindung	22
4.1 Brainstorming Sessions	22
4.1.1 Lösungsraum abstecken	23
Kritische Bewertung der möglichen Richtungen:	24
4.1.2 Potentielle Lösungen für unseren Fokus	24
4.2 Definition Roadmap	25
5 Entwicklung	27
5.1 Proof of Concept - ORDS API	27
5.2 Prototype ORDS API - Iteration 1	28
5.3 Workshop mit der BVL IT	29
5.4 Prototype ORDS API - Iteration 2	31
5.5 Weitere Inkremente	32
5.5.1 Deploymentskript & Dokumentation	32
5.5.2 Skript für die Inkonsistenzprüfung im BVL	32
5.5.3 Problemfeld Maschinenlesbarkeit	33
6 Ergebnis	33
6.1 Prototyp API	33
6.2 KPIs & Monitoring	33
6.3 Prüfskript für interne Inkonsistenzprüfung	35
6.4 Vorarbeit Maschinenlesbarkeit	35
6.4.1 Definition "Maschinenlesbarkeit"	36

6.4.2 Umfrage zum Thema "Maschinenlesbarkeit"	37
6.4.3 Prozessvorschlag - Definition eines Schemas	37
6.5 Übersicht: Arbeitspakete hinter einer API	39
7 Nächste Schritte	42
7.1 Unmittelbare Folgen aus unserem Projekt	42
7.1.1 Prototyp veröffentlichen	42
7.1.1.1 Prozess	42
7.1.1.2 Kommunikation & Markteinführung	43
7.1.2 Nutzungsanalysen	44
7.1.3 Access ablösen	46
7.2 Weiterreichende Schritte	46
7.2.1 API weiterentwickeln	46
7.2.2 Über eine API hinaus	50
7.2.2.1 Infozupf Weiter- / Neuentwicklung	51
7.2.2.2 Maschinenlesbarkeit der Daten verbessern	53
7.2.2.3 Prozess anpassen	55
7.3 Langfristige Vision bilden	56
8 Danksagung, Team & Kontakt	58
9. Anhang	59
9.1 Übersicht Tools	59
9.2 Warum haben wir uns für ORDS entschieden?	59
9.3 Was kann ORDS und wo stoßen wir an Grenzen?	61

Executive Summary/Case Study

In unserem 12 wöchigen Projekt beschäftigten wir uns mit der Pflanzenschutzmittelzulassungsdatenbank des Bundesamts für Verbraucherschutz und Lebensmittelsicherheit (BVL). Ziel der Projektes war es im Rahmen der Datenstrategie des Bundes zu zeigen, wie eine Datenbank maschinenlesbar und anwender:innenfreundlich gestaltet werden kann.

Die Pflanzenschutzmittelzulassungsdatenbank (PSM-DB) des BVL wird bereits digital veröffentlicht. Aus diesem Grund wurde dieser Anwendungsfall durch den Projektpartner Bundesministerium für Ernährung und Landwirtschaft (BMEL) gewählt. Es galt die Bereitstellung zu analysieren und an aktuelle Anforderungen anzupassen. Dies sollte mit einem minimalinvasiven Eingriff in die Prozesse des BVL erreicht werden. Als Ausgangspunkt unserer Analyse haben wir untersucht, welche Probleme im Umgang mit der PSM-DB bestehen.

Wir entschlossen uns, den Prozess der Datenbereitstellung nutzer:innenfreundlicher zu gestalten - sowohl innerhalb, wie auch außerhalb des BVL. Eine andere Möglichkeit wäre gewesen die Daten selbst und deren Struktur (Maschinenlesbarkeit) anzupassen. Wir erwarten durch diesen Fokus auf eine Prozessverbesserung (Bereitstellung), allen Anwender:innen, nicht nur hochspezialisierten, gleichermaßen zu helfen. Um den Bereitstellungsprozess zu verbessern, haben wir eine digitale Schnittstelle (API) entwickelt, die die Daten leichter erreichbar zur Verfügung stellt und dabei möglichst wenig Wartung im BVL bedarf. Diese API haben wir mit Nutzer:innen getestet, deren Anregungen eingeholt und kontinuierlich eingearbeitet.

Innerhalb unseres Projektes wurde bereits ein Großteil des Integrationsprozesses des entwickelten API-Prototypen in die BVL-IT durchgeführt. Wir empfehlen, mit einer Freischaltung in eine Testphase zu starten. Hierdurch kann im Dialog mit den Anwender:innen ein klares Verständnis für die Anforderungen an eine Weiterentwicklung der API und deren Betrieb aufgebaut werden.

Darüber hinaus schlagen wir Optimierungen der internen Prozesse sowie eine Vereinfachung der Datenstruktur und Dateninterpretierbarkeit vor. Details zu all unseren Handlungsempfehlungen finden Sie in [7 Nächste Schritte](#).

1 Pflanzenschutzmittelzulassungsdatenbank

1.1 Ausgangslage

Die Bundesregierung entwickelt aktuell eine Datenstrategie um Deutschland für eine datenbasierte Zukunft zu wappnen. Der Staat soll eine Vorreiterrolle in der Verfügbarmachung von Daten einnehmen und die Datenschätze der Behörden künftig besser für die Öffentlichkeit zugänglich machen, um Innovationen zu fördern. Das Bundesamt für Verbraucherschutz und Lebensmittelsicherheit (BVL) hat aufgrund seiner Aufgaben diverse Datenbanken, die perspektivisch anwender:innenfreundlich bereitgestellt werden sollen. Das übergeordnete Bundesministerium für Ernährung und Landwirtschaft (BMEL) treibt eine starke Digitalisierungsvision voran.

Aktuell fehlt es sowohl dem Ministerium (BMEL) als auch dem BVL an Erfahrung ein solches Digitalisierungsvorhaben im Datenkontext anzugehen. Mit der Teilnahme bei Tech4Germany möchte man früh erste Erfahrungen sammeln und von modernen Ansätzen lernen. Ziel ist es mit ersten Erfahrungen die breite Umsetzung der Datenstrategie besser angehen zu können. Zum gemeinsamen Lernen wurde ein konkreter Use Case aus dem BVL ausgewählt, bei dem in der zugehörigen Anwender:innenlandschaft bereits konkrete Vorstellungen zum Thema Datenverfügbarkeit existieren. Unter anderem, da im Rahmen der Möglichkeiten der Behörde bereits Daten digital zur Verfügung gestellt werden: Einmal in Form eines [Webinterfaces](#), um spezifische Informationen abzufragen und zusätzlich in Form eines monatlich per E-Mail verschickten [Datenbank Dumps im Access Format](#).

Die Anwender:innenseite in Form von Verbänden, Industrie und Landwirtschaft verfolgt das übergeordnete Ziel, mit der besseren Datenbereitstellung den Einsatz von Pflanzenschutzmitteln zu präzisieren und damit zu reduzieren. Ein Weg hierzu ist Precision Farming: "Das richtige Mittel, in der richtigen Menge, am richtigen Ort, zur

richtigen Zeit.“ Dieses Ziel möchten sie mit der Unterstützung von Hard- und Software verfolgen. Die Technologie liefert zum Beispiel Applikationsdüsen für Pflanzenschutzmittel, die einzeln angesteuert werden können um Bereiche auszusparen. In der Software eines Farm-Management-Systems laufen alle Informationen zusammen, die der Landwirt braucht um den Einsatz des Pflanzenschutzmittels richtig zu planen, es kann künftig so etwas wie das “Business Intelligence System“ für den Acker werden.

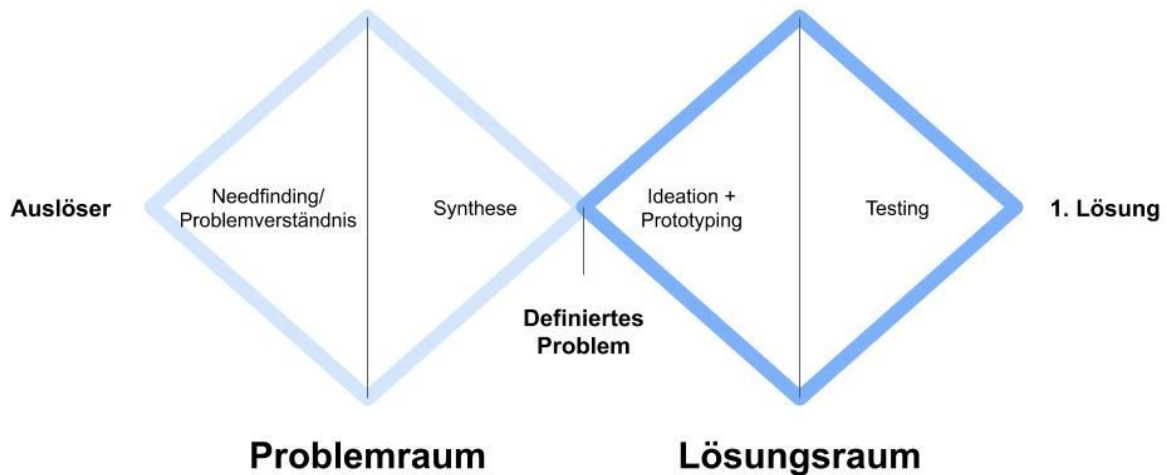
Neben der korrekten Ausbringung von Pflanzenschutzmitteln können durch die bessere Verfügbarkeit der Daten wiederkehrende Tätigkeiten besser automatisiert und die Anwender:innen bei der komplexen Anwendung von Pflanzenschutzmitteln unterstützt werden. Zudem erhöhen digitale Lösungen, wie Entscheidungshilfen, die Sicherheit bei der Anwendung. Durch die Verfügbarkeit der aktuell gültigen Information zur Anwendung wird die korrekte Ausbringung unterstützt. Dies hilft dem Umweltschutz und letztendlich dem Verbraucher.

1.2 Über Tech4Germany

Das Ziel von Tech4Germany ist ein digitaler Staat, der einerseits Bürger:innen-zentrierte Services, also einfach zu bedienende Leistungen, anbietet, andererseits seine Mitarbeitenden dazu befähigt, gute digitale Produkte zu entwickeln. Hierzu bringt das Tech4Germany Fellowship jedes Jahr die landesweit besten Digital-Talente und kreativen Köpfe in einem 12-wöchigen Programm mit Behörden und Ministerien zusammen, um gemeinsam mit modernen Arbeitsweisen und konsequenter Nutzer:innenzentrierung prototypische digitale Produkte zu entwickeln. Das Fellowship steht unter der Schirmherrschaft des Chef des Bundeskanzleramts, Prof. Dr. Helge Braun.

1.3 Arbeitsweise

Wir haben das Projekt mithilfe eines Design Thinking Prozesses umgesetzt. In diesem ergründet man zuerst das Problem bevor man sich mit der Lösung beschäftigt. Näheres zur Methode finden Sie in der Methodendokumentation in den Projektressourcen. Dort ist auch der Double Diamond beschrieben, der im Folgenden als Analogie verwendet wird. Hier nur eine Abbildung dessen:



Dieses Dokument spiegelt unseren, und damit den Design Thinking, Prozess wider. So beschreibt Kapitel 2 die Analyse des uns mitgeteilten Problems und weitere Erkenntnisse, die wir dadurch gewonnen haben ("Problemraum öffnen/ Problemverständnis" im Double Diamond). In Kapitel 3 werden diese gefundenen Probleme dann konsolidiert und Erkenntnisse zusammengefasst ("Synthese" im Double Diamond), bevor wir uns in Kapitel 4 damit beschäftigen mögliche Lösungsideen zu entwickeln ("Ideate" im Double Diamond). In den Kapiteln 5 und 6 beschreiben wir die Entwicklung und die Ergebnisse dessen ("Testing" im Double Diamond). Letztlich sprechen wir in Kapitel 7 Empfehlungen für weitere Schritte aus.

2 Analyse

2.1 Kern Fragestellungen

Da der Themenbereich "Pflanzenschutzmittel" komplex ist und viele, unterschiedliche Anwender:innen hat, war unser Ziel zunächst 1) uns in den Themenbereich der Pflanzenschutzmittel und deren Anwendung einzuarbeiten und 2) herauszufinden was in der Verwendung der Daten zu Pflanzenschutzmittelzulassungen gut läuft und was noch verbessert werden kann.

Spezifischer leiteten uns folgende Fragestellungen:

- Wer sind die Bezieher:innen der Pflanzenschutzmittelzulassungsdaten?
- Wer benötigt welche Daten?
- Welche konkreten Use Cases müssen abgebildet werden?
- Was ist gut an der aktuellen Datenbereitstellung?

- Ist der Prozess/die Umsetzung aktuell ausreichend?
- Wie können Daten des BVL für die Außenwelt zugänglicher gemacht werden?
- Wo ist der Ursprung des Problems?

2.2 Vorgehen

Um den Kontext der Pflanzenschutzmittel und die damit verbundenen Probleme zu erkunden, haben wir offene Expert:inneninterviews geführt. Zielsetzung war es, einen schnellen und tiefen Einblick in den Kontext der BVL PSM Datenbank, ihrer Anwender:innen und deren Bedürfnisse zu bekommen. Neben der vorgefertigten Stakeholder/Expert:innenliste von Tech4Germany haben wir weitere Kontakte durch Interviewpartner:innen vermittelt bekommen. Es wurde mit 18 Expert:innen aus unterschiedlichen Organisationen gesprochen. Hierbei wurden neben den Verbänden als Interessenvertreter:innen auch Vertreter:innen von Organisationen aus Industrie, Forschung, Beratung und dem BVL selbst interviewt. Die Interviewleitfäden befinden sich in den in den Projektressourcen. Die Notizen der Interviews finden sich in anonymisierter Form in einem digitalen Whiteboard als PDF in den Projektressourcen und unter folgendem Link:

https://miro.com/app/board/o9J_kkPogK4=?moveToWidget=3074457350285004945&cot=12

2.3 Erkenntnisse aus den Expert:inneninterviews

Neben den Erkenntnissen, welche in Kapitel 3 beschrieben werden, haben wir weitere Erkenntnisse auf allgemeiner Ebene gesammelt:

- Die Anzahl der Kernpersonen im Umfeld der PSM-DB ist überschaubar → Es wäre möglich einen kleinen Expert:innenpool zusammenzustellen, der viel Wissen vereinen würde und Interoperabilität anstrebt. Einen Vorschlag der Zusammenstellung, finden Sie in den Projektressourcen.
- Die treibenden Kräfte sind mit viel Leidenschaft involviert und haben große intrinsische Motivation etwas voran zu bringen → Sie sind offen für einen Dialog und Zusammenarbeit.
- Deutschland ist bezüglich der Verfügbarkeit der PSM-Daten im europäischen Vergleich vorne → Dennoch gibt es viel Potential.

- Die Tatsache, dass das BVL die PSM-Daten überhaupt als Gesamtauszug der Datenbank zur Verfügung stellt, finden alle gut. → Eine vereinfachte Bereitstellung kann große Auswirkungen haben.
- Man hat sich mit dem aktuellen Prozess der Bereitstellung der PSM-DB arrangiert → Das bedeutet aber nicht, dass er als gut empfunden wird.
- Es ist eine sehr heterogene Anwender:innenlandschaft mit diversen Bedürfnissen → Man wird es nicht allen recht machen können.

3 Konsolidierung und Erkenntnisse

Aus den Interviews konnten wir viel Wissen gewinnen und viele Probleme aufdecken, die aber aufgrund der Menge und Komplexität der Informationen noch konsolidiert werden mussten. Dafür haben wir verschiedene Methoden verwendet, die im folgenden beschrieben sind:

3.1. Systemkarte

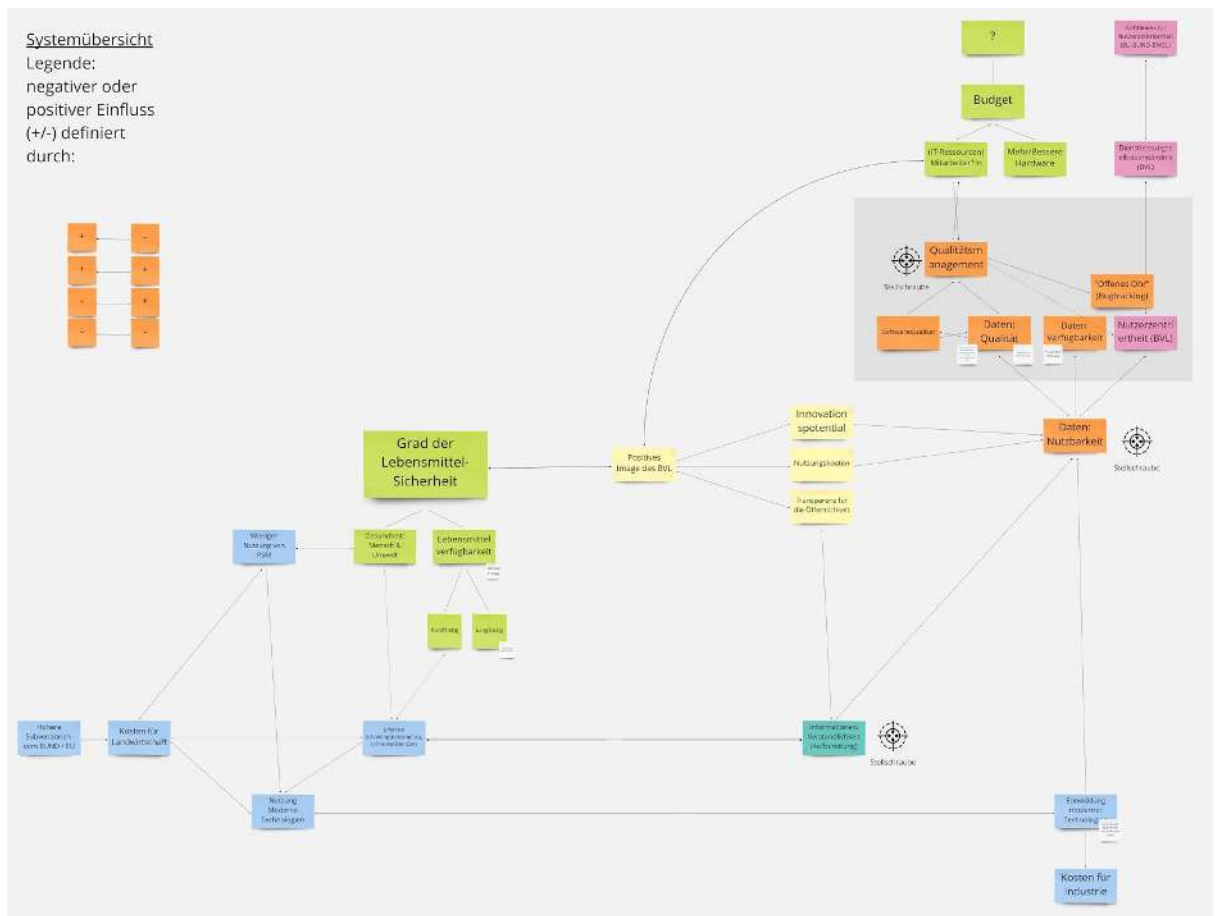
Die [Systemkarte](#) zeigt auf, wie Wechselwirkungen innerhalb eines Systems zusammenhängen. Es verbindet unterschiedliche Komponenten und Akteure und zeigt auf, wie sich einzelne Elemente des Systems verändern, wenn an einer Stellschraube gedreht wird. Ziel der Systemkarte ist es, Zusammenhänge aufzuzeigen und Hebel zu erkennen.

Im Rahmen unserer Recherche konnten wir drei Hebel erkennen:

1. Das IT-Qualitätsmanagement im BVL - Ein besseres Qualitätsmanagement wirkt sich unmittelbar auf die Software-, und Datenqualität aus - und somit auf die Nutzbarkeit der Daten. Durch eine höhere Nutzer:innenzentriertheit durch die Beachtung externer Anfragen, können Probleme aufgedeckt und behoben werden. Hierfür muss jedoch (IT-)Personal verfügbar sein.
2. Die Nutzbarkeit der Daten, die vom BVL im Rahmen der PSM Zulassung veröffentlicht werden - Durch eine bessere Nutzbarkeit der zur Verfügung gestellten Daten, steigt das Vertrauen und Verständnis und damit das Image des BVL. Es könnten sich, mit dem BVL als attraktivem Arbeitgeber, mehr Personen

bewerben und somit mehr IT-Ressourcen akquiriert werden. Es kann eine Qualitätsverbesserung angestellt werden.

- Die Verständlichkeit der Informationen, die das BVL nach außen gibt - Die Verständlichkeit der Daten wirkt sich wiederum auf die Nutzbarkeit der Daten aus und zahlt somit auf die oben genannten Kreisläufe ein. Eine bessere Verständlichkeit wirkt sich unmittelbar auf eine präzisere Nutzung von PSM aus und damit auf eine Reduzierung von möglichen Fehleranfälligkeiten.



Für die Detailansicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:
https://miro.com/app/board/o9J_kkPogK4/?moveToWidget=3074457350285005002&ot=12

3.2. Marktübersicht

Das Umfeld der PSM DB beinhaltet eine recht komplexe Marktsituation. Es gibt zahlreiche Akteure, die Interesse daran haben, hochwertige, strukturierte Daten rund

um die Pflanzenschutzmittelzulassung aus einer zentralen und rechtssicheren Quelle zu bekommen. Die Kette gestaltet sich wie folgt:

1. Chemiekonzerne
2. BVL / Bewertungsbehörden
3. Forschungsinstitute
4. Verbände
5. Industrie (Maschinenbau) / Dienstleistung (Software)
6. Berater:innen (Landwirtschaft)
7. PSM (Bau-)Märkte
8. Landwirt:innen
9. Kontrolleur:innen (Landwirtschaft)
10. Lebensmittel Großhandel
11. Lebensmittel (Super-)Märkte
12. Konsument:innen

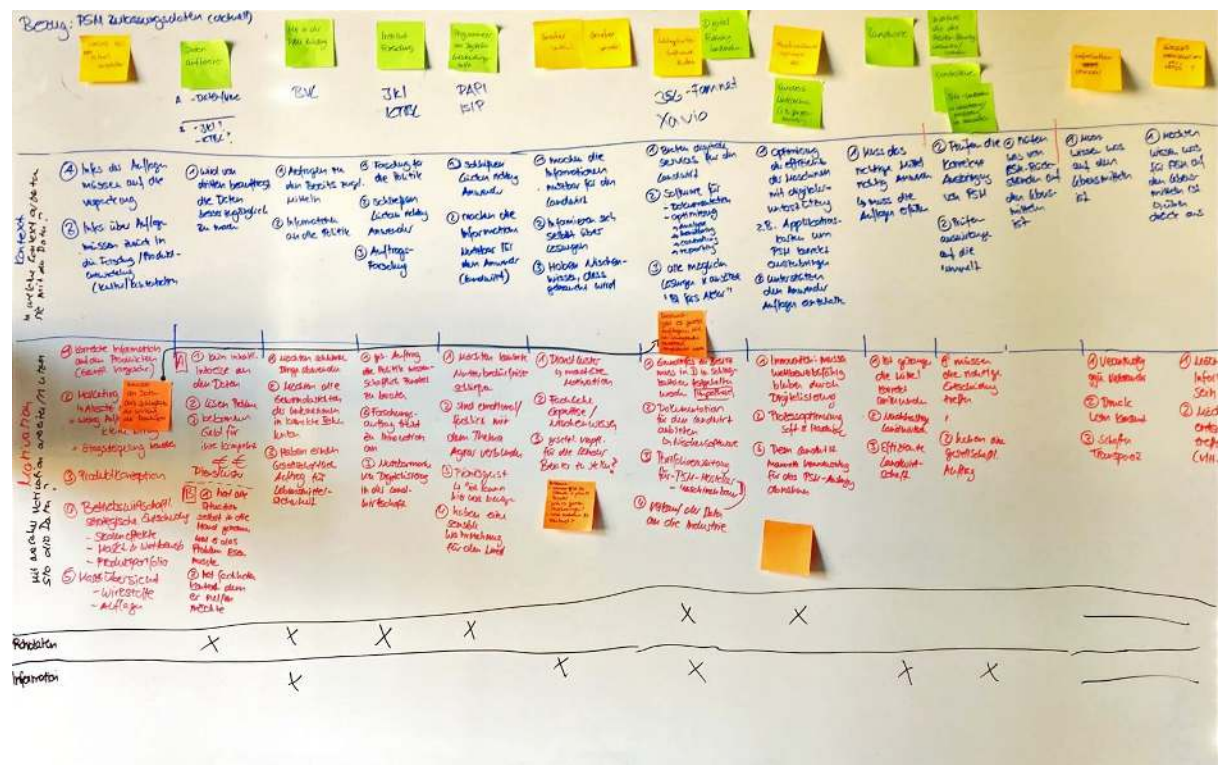


Für die Detailansicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:

Aus der Marktübersicht nehmen wir mit, dass das Thema der Pflanzenschutzmittel sehr weitreichend ist. Der Prozess startet bei Industriefirmen und gelangt über viele Institutionen letztendlich zum Verbraucher.

3.3. Anwender:innenübersicht

Im weiteren Verlauf wurden die Anwender:innen der PSM Daten herausgearbeitet. Hierbei wurden jeweils jene Perspektiven unterschieden, die sich in Kontext und Motivation der BVL PSM Daten differenzieren. Beim Gegenüberstellen der einzelnen Perspektiven kristallisierte sich heraus, dass Sie sich in zwei übergeordnete Gruppen einordnen lassen. Zum einen die Anwender:innen, die die Daten benötigen um Sie weiterzuverarbeiten (z.B. Softwareanbieter, die Abstandsauflagen in die digitale Schlagkartei ergänzen möchten). Zum anderen jene Anwender:innen, die die Informationen aus den Daten benötigen um danach zu handeln (z.B. Berater:innen, die Landwirt:innen für die passende Indikation beraten möchten). Es gibt also Anwender:innen, die die Daten beziehen möchten und jene, die die Informationen aus den Daten brauchen.



Für die Detailansicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:
https://miro.com/app/board/o9J_kkPogK4=?moveToWidget=3074457350285002882&cot=14

SCOPING Perspektive 1: Aufgrund der Kürze der Projektzeit wurde im weiteren Verlauf der Fokus auf die Gruppe jener Anwender:innenperspektiven gelegt, die die Daten unmittelbar benötigen und keine daraus zu interpretierenden Informationen. Dadurch wird die Gruppe jener Anwender:innen in den Mittelpunkt gerückt, die bereits Lösungen für die bessere Bereitstellung von Informationen liefern. Es wurde die Annahme getroffen, dass die andere Gruppe, die nur die Informationen benötigt, indirekt profitiert.

3.4. *Personas*

Personas sind ein Instrument um unterschiedliche Anwender:innenperspektiven greifbar zu machen. Es handelt sich dabei nicht um reale Personen, sondern die Zusammenfassung verschiedener Gruppen. Details zu allen Personas finden sich in den Projektdateien. Eine detaillierte Erläuterung zur Methodik "Personas" finden Sie in der Methodendokumentation von Tech4Germany 2020, ebenfalls in den Projektressourcen. Sie dienen in unserem Fall dazu, ein geteiltes Verständnis der Adressat:innen der Projektarbeit zu bekommen. Es ist also eine gemeinsame Projektionsfläche für Probleme und Bedürfnisse der Anwender:innen. Sie basieren nicht auf quantitativ belastbaren sondern qualitativ interpretierten Daten.



Tara
 Tara ist eine 35-jährige Frau, die in der IT-Branche arbeitet. Sie ist eine Einzelkämpferin und hat eine hohe Arbeitsbelastung. Sie ist sehr detailorientiert und hat eine hohe Genauigkeit bei ihrer Arbeit. Sie ist eine Einzelkämpferin und hat eine hohe Arbeitsbelastung. Sie ist sehr detailorientiert und hat eine hohe Genauigkeit bei ihrer Arbeit.



Oliver
 Oliver ist ein 45-jähriger Mann, der in der IT-Branche arbeitet. Er ist ein Teamplayer und hat eine hohe Arbeitsbelastung. Er ist sehr detailorientiert und hat eine hohe Genauigkeit bei seiner Arbeit. Er ist ein Teamplayer und hat eine hohe Arbeitsbelastung. Er ist sehr detailorientiert und hat eine hohe Genauigkeit bei seiner Arbeit.



Nico
 Nico ist ein 30-jähriger Mann, der in der IT-Branche arbeitet. Er ist ein Teamplayer und hat eine hohe Arbeitsbelastung. Er ist sehr detailorientiert und hat eine hohe Genauigkeit bei seiner Arbeit. Er ist ein Teamplayer und hat eine hohe Arbeitsbelastung. Er ist sehr detailorientiert und hat eine hohe Genauigkeit bei seiner Arbeit.



Ben
 Ben ist ein 25-jähriger Mann, der in der IT-Branche arbeitet. Er ist ein Teamplayer und hat eine hohe Arbeitsbelastung. Er ist sehr detailorientiert und hat eine hohe Genauigkeit bei seiner Arbeit. Er ist ein Teamplayer und hat eine hohe Arbeitsbelastung. Er ist sehr detailorientiert und hat eine hohe Genauigkeit bei seiner Arbeit.



Orin
 Orin ist ein 35-jähriger Mann, der in der IT-Branche arbeitet. Er ist ein Teamplayer und hat eine hohe Arbeitsbelastung. Er ist sehr detailorientiert und hat eine hohe Genauigkeit bei seiner Arbeit. Er ist ein Teamplayer und hat eine hohe Arbeitsbelastung. Er ist sehr detailorientiert und hat eine hohe Genauigkeit bei seiner Arbeit.



Igor
 Igor ist ein 40-jähriger Mann, der in der IT-Branche arbeitet. Er ist ein Teamplayer und hat eine hohe Arbeitsbelastung. Er ist sehr detailorientiert und hat eine hohe Genauigkeit bei seiner Arbeit. Er ist ein Teamplayer und hat eine hohe Arbeitsbelastung. Er ist sehr detailorientiert und hat eine hohe Genauigkeit bei seiner Arbeit.



Für die Detailsicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:
https://miro.com/app/board/o9J_kkPogK4/?moveToWidget=3074457350339034842&ot=14

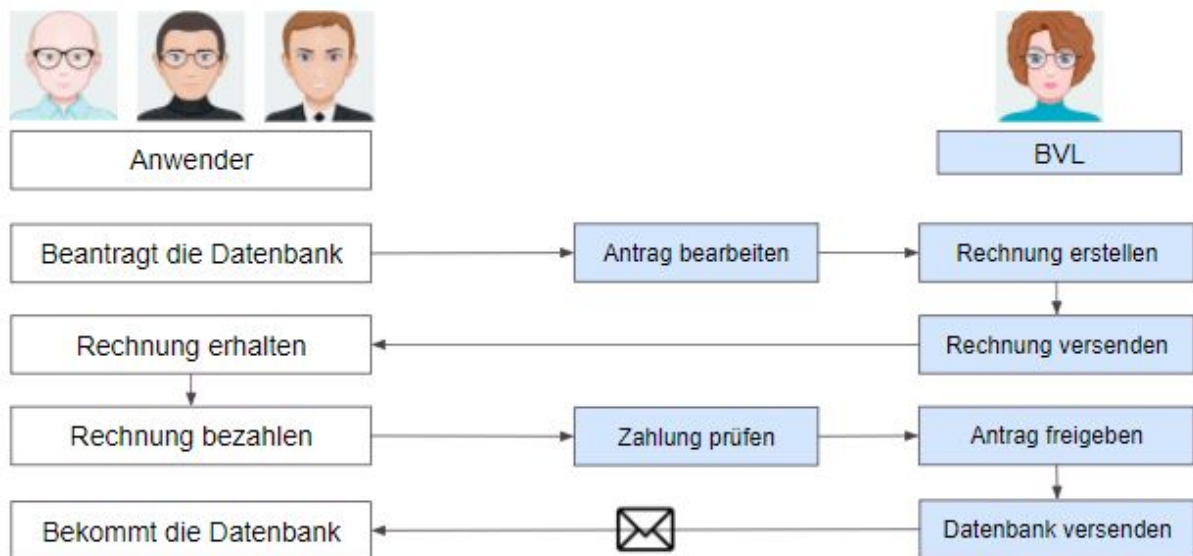
Erkenntnisse aus der Persona Synthese:

- Sehr heterogene Nutzer:innenbedürfnisse → diesen allen gerecht zu werden, wäre nicht möglich, nur Teilen gerecht zu werden nicht nachhaltig, es gilt den kleinsten gemeinsamen Nenner zu finden.
- Je nachdem welche Persona man in den Fokus seiner weiteren Arbeit rückt kann es passieren, dass einzelne Branchen oder Industriezweige bevorteilt werden.

SCOPING Perspektive 2: Im weiteren Verlauf des Projekts beschäftigen wir uns primär mit zwei Personas: (1) Barbara Behörde und (2) Sascha Software. Die Mehrheit der Probleme dieser wird an der Schnittstelle des BVLs mit der Außenwelt offenkundig. Die beiden Personas bilden die jeweiligen Seiten dieser Schnittstelle ab.

3.5. User-Journey - Status Quo

Die Daten zur Pflanzenschutzmittelzulassung können monatlich bezogen werden. Hierzu wird ein Microsoft Access Dump, der die Daten enthält, per E-Mail an registrierte Empfänger:innen der Datenbank geschickt. Um den Dump zu erhalten, müssen die Nutzer:innen Kontakt zum BVL aufnehmen und sich in den Kreis der Bezieher:innen eintragen lassen. Ein Entgelt wird erhoben, um den Aufwand, der dem BVL durch diese Form der Bereitstellung der Daten entsteht, zu kompensieren. Für hoheitliche Aufgaben anderer Behörden sind diese Daten kostenlos. Der Prozess enthält viele Teilschritte. Zudem sind mehrere Kanalwechsel aus Anwender:innensicht enthalten (E-Mail → Zip entpacken → MS Access umwandeln → ins Zielsystem einarbeiten).



Für mehr Informationen bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:

https://miro.com/app/board/o9J_kkPogK4=?moveToWidget=3074457350285004960&cot=12

Einordnung: Der Bereitstellungsprozess an der Schnittstelle ist kleinteilig. Um die Kommunikation zwischen dem BVL (Barbara Behörde) und Außenwelt (Sascha Software) zu vereinfachen, sollten die Schritte im Prozess der Datenübermittlung reduziert werden.

3.6. Kernerkenntnisse

Unsere Kernerkenntnisse haben wir in Form von Handlungsfeldern zusammengetragen. Hierzu wurde die WhatIf-Methode genutzt, um neben dem Aufzeigen des Problems auch aufzuzeigen, wie es besser sein könnte, ohne jedoch schon konkrete Lösungsvorschläge zu nennen. Bei der WhatIf-Methode handelt es sich um eine Problemsynthesierungsmethode, aus dem Repertoire des Service Designs. Sie wird genutzt, um aus einer Situation, in welcher man durch die Frage "Wie löse ich..." auf Grund der Masse an Problem nicht weiterkommt und daher die Frage in eine Form "Was wäre wenn..." umformt. Es können somit Handlungsfelder offen formuliert werden, ohne direkt eine Lösung für die einzelnen Probleme parat zu haben.

Im Folgenden sind die identifizierten acht Handlungsfelder beschrieben.

Die Handlungsfelder fassen die Probleme auf unterschiedlichen Granularitätsstufen zu einem übergeordneten Thema zusammen. Durch diese Synthese werden übergeordnete Hebel definiert, an denen man potentielle Lösungen ansetzen kann. Die Handlungsfelder werden beschrieben mit einer Gegenüberstellung der aktuellen Situation sowie einer potentiellen Veränderung in der Zukunft. Dem werden die jeweiligen Probleme und Zitate aus den Interviews zugeordnet, die auf die einzelnen Handlungsfelder einzahlen.

Die einzelnen Handlungsfelder sind miteinander verknüpft. Das Angehen eines einzelnen Handlungsfeldes kann sich auch auf benachbarte Handlungsfelder auswirken. Details hierzu finden Sie im Handout der Zwischenpräsentation und in den Projektressourcen.

Handlungsfeld 1: Rechtssicherheit

Was ist: Die Rechtssicherheit externer Lösungen fehlt.

Was wäre wenn: Die Rechtssicherheit der digitalen Interpretation hergestellt wird.

Hintergrund: Die vom BVL veröffentlichten Anwendungsaufgaben sind komplex und in einer Form verfügbar, die von Menschen teilweise schwer verständlich und von Maschinen nur teilweise interpretierbar ist. Digitale Entscheidungshilfen (Softwarelösungen, die Landwirt:innen bei der Entscheidung helfen sollen, welches Mittel, wie, angewendet werden darf) versuchen, diese Lücke zu schließen, Sie müssen hierfür die BVL Daten überarbeiten. Dadurch geht die Rechtssicherheit verloren, da die Programme nicht mehr auf den Originaldaten operieren. Würde das BVL die Daten in einer Form zur Verfügung stellen, die besser von Maschinen interpretierbar ist, könnte man die Rechtssicherheit besser gewährleisten.

Handlungsfeld 2: Verschiedene Sprachen

Was ist: Die Behörde (BVL) und externe Anwender:innenwelt sprechen nicht die gleiche Sprache.

Was wäre wenn: Die Behörde (BVL) und externe Anwender:innenwelt sprechen eine Sprache.

Hintergrund:

Während unserer Interviews haben wir herausgefunden, dass unterschiedliche Erwartungen bei Behörde und Außenwelt existieren. Die Anwender:innenwelt erwartet einige Softwarelösungen, die so in der Behörde aufgrund von Personalmangel schlicht nicht umsetzbar ist. Hierbei könnte der gemeinsame Dialog helfen. Ein spezifisches Beispiel dieser unterschiedlichen Erwartungen und anderer Sprachen ist der Begriff der Maschinenlesbarkeit, der auf beiden Seiten unterschiedlich aufgefasst wird (Interpretierbarkeit vs. Einlesbarkeit).

Handlungsfeld 3: Verständnis für Sichtweise des Staats

Was ist: Das Verständnis der Landwirte für den Staat ist eingeschränkt.

Was wäre wenn: Das gegenseitige Verständnis der Parteien existiert.

Hintergrund:

Anwendungsaufgaben werden als kompliziert und unübersichtlich wahrgenommen. So haben viele Landwirte Mühe die Aufgaben in juristischem Deutsch zu verstehen. Eine rechtssichere Anwendung kann somit erschwert werden, da die Aufgaben meist

interpretiert werden müssen. Die Datengrundlage für eine informierte Anwendung ist somit nicht optimal. Da jedoch die Konsequenzen durch eine Fehlanwendung beim Landwirt hoch sind, fühlen sich Landwirte hier nicht ausreichend unterstützt. Die Einhaltung der Vorgaben bei der Verwendung von PSM wird durch eine teilweise schwierige Verständlichkeit der juristischen Aufagentexte, laut Landwirten, unnötig erschwert. So benennen in einer Umfrage durch die Bitkom, den Bauernverband und der Rentenbank im Bericht zur digitalen Landwirtschaft 2020, 92% der Landwirte die größte Herausforderung in ihrem Betrieb als "Vorschriften und Regelungen bremsen uns aus", die mangelnde Digitalisierung jedoch nur 58%.

Handlungsfeld 4: Dienstleistungsmindset

Was ist: Das BVL versteht sich nicht als Dienstleister für die Nutzer:innen der Daten

Was wäre wenn: Die Bereitstellung der Daten wird im Nutzer:innen-Dialog umgesetzt.

Hintergrund: Das BVL stellt die Daten zur Verfügung. Dies kann grundsätzlich als eine Dienstleistung verstanden werden. Die Form der Daten und deren Bereitstellung sind jedoch auf die internen Prozesse des BVL abgestimmt und daraufhin optimiert - nicht mit Blick auf die Anwender:innen. Zudem gibt es keine klar definierten Kommunikationskanäle für den Austausch über die angebotene Dienstleistung der Datenbereitstellung mit der Außenwelt, wodurch nur wenige von außen motivierte Veränderungen durchgeführt werden können.

Handlungsfeld 5: Standards

Was ist: Es gibt wenig Interoperabilität, da es keinen offiziellen Standard gibt. Es definiert daher jeder seinen eigenen Standard.

Was wäre wenn: Es gibt einen offiziellen Standard, an dem man sich orientieren kann.

Hintergrund: Für das digitale Verarbeiten der Informationen in Softwareanwendungen müssen die Daten in strukturierter Form vorliegen. Da es aktuell auf keiner Ebene (national, EU, weltweit) einen Standard gibt, definiert sich jeder Softwarehersteller seinen eigenen Verarbeitungsstandard. Hierdurch werden die Lösungen nur schwer vergleichbar und können kaum miteinander verknüpft werden. Das wiederum hemmt die Digitalisierung in der Landwirtschaft, da es aktuell noch keine sinnvolle Lösung gibt, die von allen anerkannt wird.

Handlungsfeld 6: Komplizierte Prozesse

Was ist: Im BVL gibt es diverse komplizierte Prozesse.

Was wäre wenn: Die Prozesse werden verschlankt und (teil-)automatisiert.

Hintergrund: Es gibt mehrere sehr komplizierte Prozesse im BVL, welche viele manuelle Schritte verlangen (Bereitstellungsprozess, Prüfprozess). Diese können durch die Nutzung von digitaler Technologie verschlankt werden, bzw. komplett oder teilweise automatisiert ablaufen.

Handlungsfeld 7: IT Ressourcen

Was ist: Es fehlt an IT Personal-Ressourcen im BVL.

Was wäre wenn: Das BVL hat mehr IT Mitarbeiter.

Hintergrund: Die IT-Systemlandschaft im BVL ist durch die vielfältigen Aufgaben kompliziert. Darüber hinaus ist sie historisch gewachsen. Neue Anforderungen wurden pragmatisch umgesetzt, da für einen geplanten Ausbau und eine Weiterentwicklung die personellen Ressourcen fehlen. Aktuell sind nur ein Teil der ausgeschriebenen IT Stellen besetzt.

Handlungsfeld 8: Datenqualität

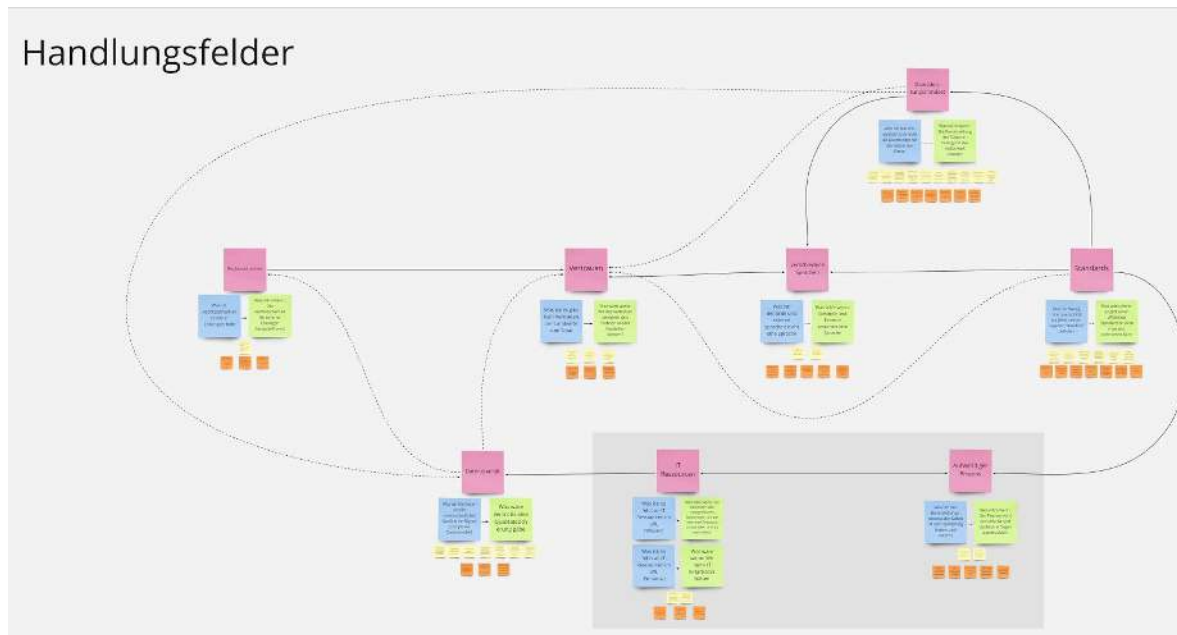
Was ist: Die Qualität der veröffentlichten PSM Daten variiert stark.

Was wäre wenn: Es gibt eine bessere Qualitätssicherung.

Hintergrund: Es gibt unterschiedliche Veröffentlichungsprozesse für Informationen rund um Pflanzenschutzmittel (PSM Access, PSM Web, Dokument mit Sonderzulassung, Dokument mit Aufbrauchfristen, Oracle DB für BVL-nahe Institute). Wo Daten als Datenbankdump veröffentlicht werden, gibt es viele Herausforderungen für Externe im Hinblick auf die Datenqualität. Zwei Beispiele sind ein schwer verständliches Datenmodell oder das Fehlen von Fremdschlüsseln. Darüber hinaus wünscht sich die Anwender:innenwelt eine zentrale Informationsquelle, in der alle Elemente konsolidiert sind.

Zusammenfassung

Die genannten Handlungsfelder haben auch untereinander Wechselwirkungen. Diese und die Probleme und Zitate aus Interviews aus denen die Handlungsfelder entstanden sind, haben wir grafisch zusammengetragen:



Für die Detailansicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen: https://miro.com/app/board/o9J_kkPogK4/?moveToWidget=3074457350285005001&ot=12

3.7 Priorisierung der Handlungsfelder

Aufgrund der Kürze der Zeit war es notwendig, einen Fokus zu setzen. Im ersten Schritt wurden jene Handlungsfelder ausgeklammert, auf die wir im Rahmen des Fellowships keinen Einfluss nehmen können. Grundlage dieser Beurteilung war die Frage, ob Sie in unserem Wirkungskreis liegen und zum Skillset des Teams passen. Konkret ausgeklammert haben wir die Handlungsfelder: Verständnis für Sichtweise des Staats, Rechtssicherheit und IT Ressourcen (Personal).

Mit den übrigen Handlungsfeldern gab es einen Priorisierungsworkshop, gemeinsam mit unseren Digitallotsen, um den Fokus für den weiteren Projektverlauf zu setzen. In diesem wurde der Weg der Recherchephase aufgezeigt und die Handlungsfelder vorgestellt. Um jedem die Chance zu geben, sich in Ruhe einzudenken wurde die Entscheidung auf einen weiteren Termin vertagt. Vor der Entscheidungsfindung via Punktevergabe wurden die Vor- und Nachteile bzw. Stärken und Schwächen der einzelnen Handlungsfelder diskutiert. Die beiden Workshopsessions fanden remote

auf dem digitalen Whiteboard statt. Als Ergebnis der Session wurden die folgenden Handlungsfelder priorisiert:



- 1) Datenqualität
- 2) Standards
- 3) Aufwändiger Prozess

Für detailliertere Informationen bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:

https://miro.com/app/board/o9J_kkPogK4=?moveToWidget=3074457350290710410&cot=12

Die Präsentationsfolien der Zwischenpräsentation finden Sie ebenfalls in den Projektressourcen.

4 Ideenfindung

In der Ideenfindungsphase gilt es, mögliche Lösungen für das Problem zu finden, auf das man sich konzentrieren will.

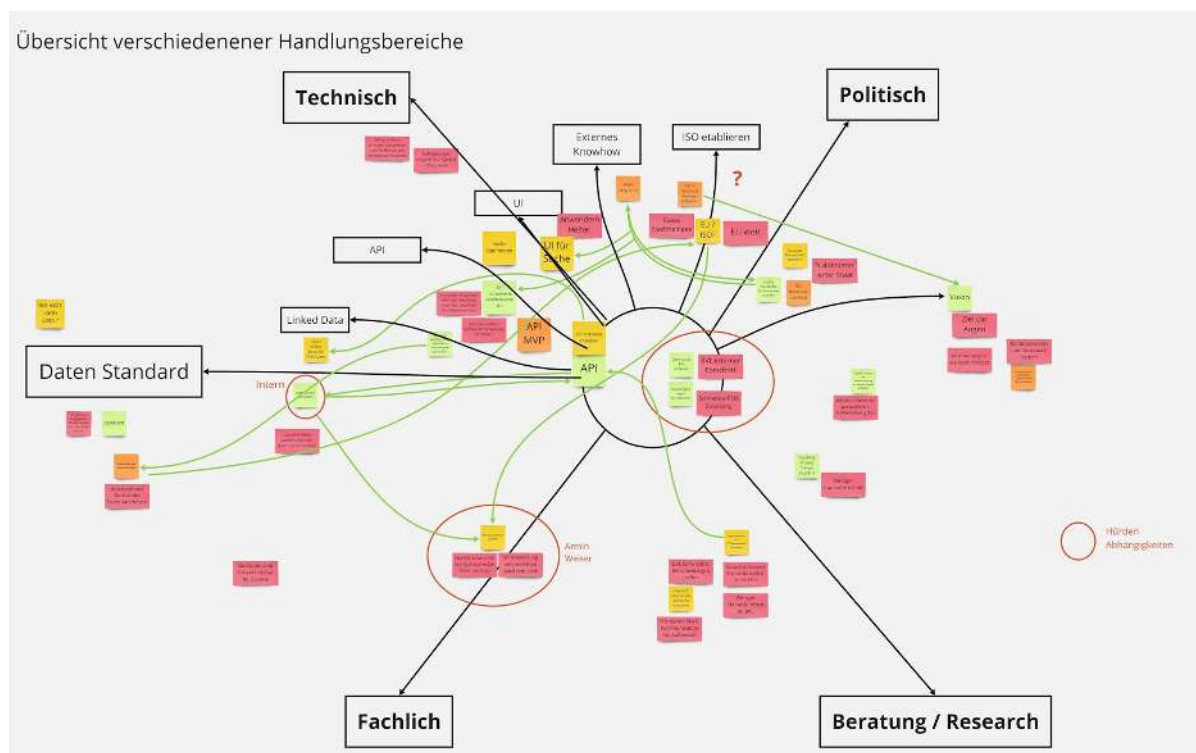
4.1 Brainstorming Sessions

Im Rahmen der Expert:inneninterviews haben wir viele Informationen gesammelt. Mit diesem Expert:innenwissen im Kopf galt es im nächsten Schritt potentielle Lösungsideen zu sammeln, zu strukturieren und zu priorisieren. Als Vorarbeit hierzu haben wir ein gemeinsames Verständnis von Richtungen erarbeitet, in denen unsere Lösung liegen kann.

4.1.1 Lösungsraum abstecken

Bevor wir uns näher mit potentiellen Lösungen befassten, haben wir uns ein gemeinsames Verständnis des Lösungsraums erarbeitet. Auf Basis des gewonnenen Wissens aus den Interviews haben wir überlegt in welche Richtung potentielle Lösungen gehen könnten und welche Aufgaben auf dem Weg in diese Richtung liegen. Als mögliche Richtungen haben wir definiert: Technisch, Politisch, Beratung/Research, und Fachlich. Der Datenstandard wurde gesondert hervorgehoben, da er viele Unterpunkte inkorporierte.

Ziel der Session war neben dem gemeinsamen Verständnis, die Ideen aus dem Team zusammenzutragen und zu externalisieren. Nach einer individuellen Brainstormingphase wurden die Ideen den jeweiligen Richtungen zugeordnet. Dabei sind kleinere Inkremente näher zur Mitte angeordnet, größere weiter in die entsprechende Richtung der Ausrichtung. In der dritten Phase haben wir die Ideen geclustert und in Abhängigkeit zueinander gebracht. Hierbei kristallisierte sich heraus, dass eine API sich positiv auf viele andere technische Bereiche auswirken und dabei als Grundlage dienen kann.



Für die Detailansicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:

https://miro.com/app/board/o9J_kkPogK4=?moveToWidget=3074457350285004993&ot=12

Kritische Bewertung der möglichen Richtungen:

Darüber hinaus zeigt sich, dass einige Richtungen aufgrund von Restriktionen wie Dauer des Fellowships, Größe der Aufgabe oder auch Skillset des Teams außer Reichweite sind. Ein weiterer Aspekt ist die Ausrichtung des Programms Tech4Germany. Im Sinne der Vision "Show, don't tell!" war es uns als Team wichtig eine technische Lösung zu schaffen, die tatsächlich anwendbar ist. Durch die Zusammensetzung des Teams (2 Engineers (Entwicklung), 1 Engineer (Data Science), 1 Product) war es naheliegend als Rahmen für die Lösung einen Weg im Bereich der Richtungen "Technisch" und "Daten Standard" anzusiedeln.

4.1.2 Potentielle Lösungen für unseren Fokus

Im Rahmen des Absteckens des Lösungsraums kristallisierte sich die API als Voraussetzung für viele weitere Aufgaben heraus. Aus diesem Grund haben wir sie in den Fokus unserer potentiellen Lösung gesetzt.

Auch hier erfolgte eine Brainstorming Session, in der wir die Ideen externalisiert haben. Im zweiten Schritt erfolgte ein Clustering und das ins Verhältnis setzen der einzelnen Ideen.



Für die Detailansicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:
https://miro.com/app/board/o9J_kkPogK4/?moveToWidget=3074457350290710412&ot=12

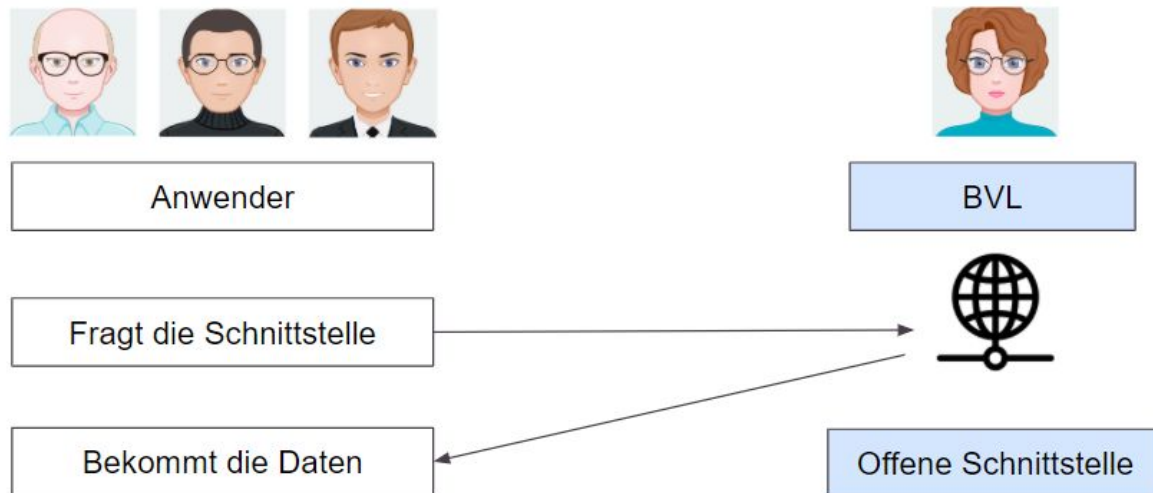
In einer zweiten Phase, wurden die Elemente in Aufgabenpakete eingeteilt, um aufzuzeigen wie sinnvolle Iterationen aussehen könnten.

4.2 Definition Roadmap

Als Ergebnis der Brainstorming Sessions (4.1) wurde festgestellt, dass der erste Schritt unserer Lösung eine simple API sein soll, die die Daten, so wie sie aktuell sind, nach außen gibt. Es werden also die gleichen Daten veröffentlicht, die auch durch den Access E-Mail Prozess abgebildet sind. Dies hat für die bereits existierenden Bezieher:innen der Daten den Vorteil, dass sich die Datenstruktur noch nicht ändert. Damit können bestehende Lösungen im ersten Schritt nur den Weg, wie die Daten in Ihre Systeme gelangen, auf die Schnittstelle umstellen. Das heißt, es wird die Möglichkeit geschaffen in kleinen Schritten zu migrieren.

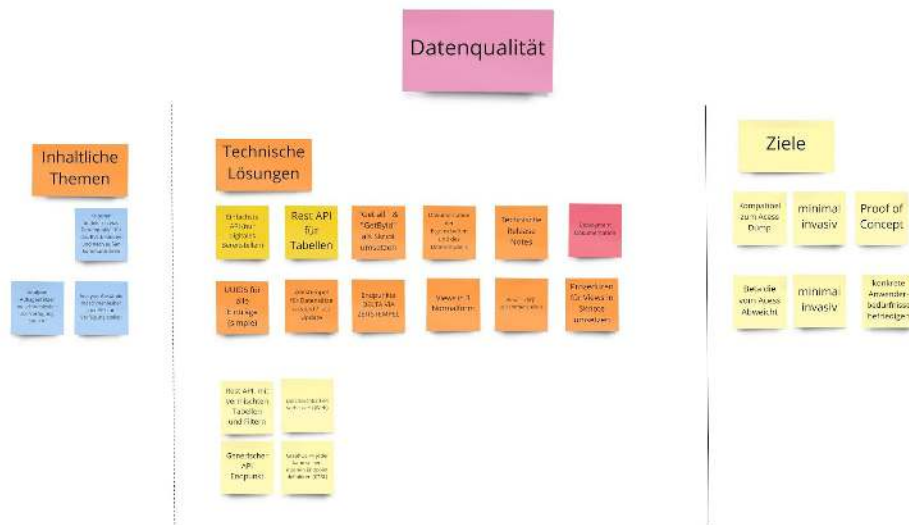
Der Austausch zwischen BVL und Außenwelt ist aufwändig (Vgl. User Journey (3.5)). Den Prozess zu verschlanken ist aus unserer Sicht eine Voraussetzung für alle nachfolgenden potentiellen Lösungen. Dadurch kann im ersten Schritt bewiesen werden, dass Anwender:innen die simple API überhaupt nachfragen. Sollte keine

Nachfrage für die simple API da sein, lohnt es eventuell nicht, eine aufwändigere zu bauen.



<http://api.bvl.bund.de/psm/mittel>

Erst in weiteren Iterationen soll versucht werden, die zur Verfügung gestellten Daten anzupassen. Dies könnte zum Beispiel bedeuten, sie mit Zeitstempeln zu versehen. Um das weitere Vorgehen zu strukturieren, haben wir die einzelnen Lösungsideen in Aufgabenpakete zusammengefasst. Neben technischen Aufgabenpaketen wurden auch inhaltliche Themen identifiziert, die parallel angestoßen werden sollten. Auf technischer Ebene umfasst die Roadmap zwei Versionen des API Prototyp. (v1) soll eine simple API sein, die die Tabellen in ihrer aktuellen Struktur nach außen trägt. Erst in (v2) soll dann die Struktur der Daten nach ihrer Entstehung angepasst werden, um Anwender:innenbedürfnisse besser zu befriedigen.



Für die Detailansicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:
https://miro.com/app/board/o9J_kkPogK4=?moveToWidget=3074457350290710412&ot=12

5 Entwicklung

Im Rahmen der Entwicklung haben wir uns auf die folgende Ausgangshypothese fokussiert: Durch eine simple API kann im ersten Schritt allen Parteien geholfen werden. Wir sehen den Mehrwert in vier Punkten:

1. Langfristige Reduktion der Prozessschritte auf beiden Seiten
2. Verfügbarmachung der Daten für die Öffentlichkeit in einem modernen und frei verfügbarem Standardformat
3. Minimalinvasiver Eingriff in die aktuelle IT-Struktur
4. Geringer Migrationsaufwand für Anwender:innen.

5.1 Proof of Concept - ORDS API

Im Rahmen unserer Recherche sind wir auf eine Funktion von Oracle gestoßen (mit Oracle Database als Basistechnologie hinter der aktuellen PSM-DB des BVL), mit der man eine REST API ohne viel Aufwand aktivieren kann. Details hierzu finden sich in der technischen [Dokumentation von ORDS](#).

Zielsetzung: Prüfen ob man dem BVL mit vorhandenen Mitteln helfen kann

Durchführung: Testen ob es in kurzer Zeit möglich ist, mit ORDS eine Schnittstelle zu entwickeln, die dann Grundlage für Nutzer:innentests und somit auch als Validierung der eingeschlagenen Route genutzt werden kann

Ergebnis: Es hat geklappt, dies scheint ein sinnvoller Weg zu sein, eine erste REST API zur Verfügung zu stellen mit Technologie, die im BVL verfügbar ist.

Hindernisse: Es war eine besondere Herausforderung veraltete und proprietäre Software auf unserer Entwicklungsumgebung zum Laufen bekommen. Dies wird anderen Entwickler:innen ähnlich gehen.

5.2 Prototype ORDS API - Iteration 1

Nach dem erfolgreichen Proof of Concept gilt es im nächsten Schritt einen nutzbaren Prototypen der API zu bauen, der als Grundlage für Anwender:innentests dienen kann. Im abschließenden Anwender:innentest dieser Iterationstufe wurde die Entwicklung mit 6 Anwender:innen getestet und Feedback eingeholt. 5 dieser Anwender:innen wurden aus dem Kreis der aktuellen Bezieher:innen des Access Dumps gewonnen. Hierbei wurde der Fokus auf die technischen Anwender:innen gelegt. Ein weiterer Tester war ein domänenfremder Entwickler um zu prüfen wie gut eine "externe" Person mit dem aktuellen Stand zurecht kommt.

Zielsetzung: Aus dem Proof of Concept einen nutzbaren Prototypen bauen, der die Zugänglichkeit der Daten für die Anwender:innen verbessert und das BVL möglichst wenig Aufwand kostet.

Umsetzung: Im Rahmen der Umsetzung wurden die folgenden Features zur Verfügung gestellt:

- Feature 1: Tabellen aus dem Dump in JSON zur Verfügung stellen
- Feature 2: Für alle Tabellen die Abfragen "get all" und "get by ID" ermöglichen
- Feature 3: OpenAPI-Swagger Interface auf die API setzen
- Feature 4: Auf einem Server deployen um es öffentlich zugänglich zu machen

Nutzer:innentesting: Im Rahmen des Nutzer:innentesting wurden die folgenden Hypothesen überprüft:

- H1: Die Ablösung des Access Dumps durch JSON verbessert die Zugänglichkeit der Daten
- H2: Die Ablösung des Push Prozesses durch eine Pull Option erhöht die Zugänglichkeit der Daten und schafft Mehrwert für die Anwender:innen

- H3: Der Wegfall der monatlichen E-Mail als Reminder die Daten zu aktualisieren ist kein Problem für die Anwender:innen
- H4: Die aktuell vom BVL zur Verfügung gestellte Dokumentation ist nicht ausreichend um das Datenformat schnell zu verstehen

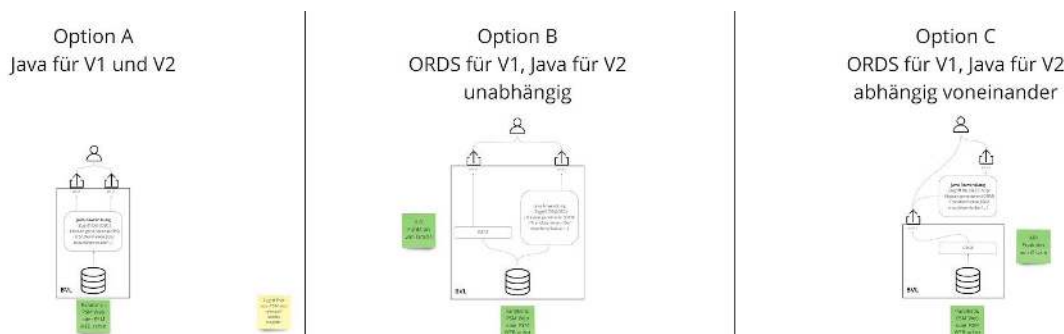
Ergebnis: Beim Nutzer:innentesting haben wir die folgenden Erkenntnisse gewonnen:

- H1: bestätigt → Die Verfügbarkeit in JSON erspart Prozessschritte
- H2: bestätigt → Die permanente Verfügbarkeit der API ermöglicht es den Anwender:innen sich mit den Daten auseinander zu setzen und reduziert die Einstiegsbarriere.
- H3: bestätigt → Die Anwender:innen haben die Fähigkeit selbst die API in regelmäßigen Abständen abzurufen, um informiert zu werden, dass die Daten aktualisiert sind.
- H4: bestätigt → Es wird eine Anwender:innendokumentation benötigt, die die API, die Endpunkte und das Datenmodell erklärt. Dies kam insbesondere beim Interview mit dem domainenfremden Anwender zum Vorschein.
- Ferner müssen die Daten vorerst nicht tagesaktuell sein, solange der Aktualisierungszyklus klar kommuniziert wird.

5.3 Workshop mit der BVL IT

Nachdem der Prototyp in der ersten Testrunde bestätigt wurde, haben wir uns auf die Implementierungsmöglichkeiten im BVL fokussiert. Klares Ziel des Fellowships seitens des Projektpartners BMEL ist es, am Ende des Projekts eine Lösung implementiert zu haben. Um hierbei einen Ansatz zu wählen, der mit dem BVL abgestimmt ist, haben wir einen Anforderungsworkshop mit der operativen Ebene der BVL IT durchgeführt.

Zielsetzung des Workshops: Wir möchten gemeinsam 3 mögliche Implementierungsstrategien für unsere Lösung diskutieren und ein Verständnis für die Vor- und Nachteile der Ansätze bekommen.



Die drei Implementierungsstrategien:

Option A: Java REST Service für v1 und v2 - **Eine** Anwendung, die auf die Datenbank zugreift, die Daten in Klassen umwandelt, die Struktur ggf anpasst und anreichert und zwei Schnittstellen zur Verfügung stellt: v1 mit den unverarbeiteten/unangepassten Daten und v2 mit den angepassten Daten.

Option B: ORDS für v1 und Java für v2 unabhängig - Es gibt ein System, das die API v1 bereitstellt mit der low-code Methode ORDS und ein weiteres System, das in einer Java Anwendung auf die Datenbank zugreift, die Daten in Klassen umwandelt, die Struktur ggf anpasst und anreichert. Dies stellt dann API v2 bereit.

Option C: ORDS für v1, Java für v2 abhängig voneinander - Hierbei wird v2 nicht als Webservice sondern als Client Library außerhalb des BVL bereitgestellt, indem Verarbeitung und Transformation der Daten aufbauend auf der ORDS Schnittstelle (v1) durchgeführt werden.

Eine Abwägung der verschiedenen Optionen und damit eine Begründung, warum wir ORDS verwenden und die Grenzen von ORDS, finden Sie im [9. Anhang](#).

Ergebnis des Workshops:

Nach der Diskussion der einzelnen Implementierungsstrategien hatten wir eine gute Übersicht über die Vor- und Nachteile aus Sicht der BVL IT zu den Ansätzen. Darüber hinaus konnten wir offene Fragen identifizieren, die über den Projekterfolg bestimmen können. Neben Klärungen der Zuständigkeiten und der Einstellung des BVL zu Open Source kristallisierte sich eine essentielle Frage heraus: Können die Daten per API kostenlos zur Verfügung gestellt werden? Diese Frage wurde vor Projektende durch das BVL positiv beantwortet. Die Schnittstelle wird kostenlos.

Im Rahmen der Auswertung des Workshops haben wir uns für Implementierungsvariante C entschieden. Ausschlaggebend für diese Entscheidung war zum einen das Ziel, der BVL-IT möglichst wenig zusätzlichen Aufwand zu verursachen, sowie die Möglichkeit, eine zweite Variante der API gemeinsam mit der Open Source Community zu entwickeln. Dadurch könnte externes Know How verwendet werden. Nichtsdestotrotz werden Ressourcen im BVL benötigt, um die Entwicklung zu betreuen.

5.4 Prototype ORDS API - Iteration 2

In der zweiten Iterationsstufe des Prototypen sollten die Erkenntnisse aus der ersten Iteration eingebaut und erneut mit Anwender:innen getestet werden. In dieser Runde haben wir mit 3 Anwender:innen aus dem Kreis der Bezieher:innen des aktuellen PSM Access Dumps gesprochen. Die Anwender:innen waren aus dem gleichen Kreis ausgewählt wie in der ersten Testphase.

Zielsetzung: Den Prototypen anwender:innenfreundlicher gestalten mit den Feedbacks aus den ersten Nutzer:innentests

Umsetzung:

- Feature 5: Inhaltliche Erläuterungen zu den API Endpunkten ergänzen
- Feature 6: Dokumentation der Abfrageparameter ergänzen
- Feature 7: Weitere Abfrageparameter ergänzen, die sinnvoll in anderen Endpunkten verwendet werden können
- Feature 8: Unittests für die API
- Feature 9: Dokumentation zusätzlicher Filtermöglichkeit durch JSON-Queries

Nutzer:innentest: Im Rahmen des Nutzer:innentests werden die folgenden Hypothesen überprüft.

- H5: Die Dokumentation ist nicht ausreichend um das komplexe Datenmodell zu verstehen
- H6: Die Filtermöglichkeit in ORDS bietet zusätzlich zum OpenAPI-Swagger Mehrwert, der dem User transparent gemacht werden sollte.

Ergebnis:

- H5: Teilweise bestätigt, die Tabellen- und Parameter Dokumentationen schaffen einen großen Mehrwert. Was noch fehlt, ist die Dokumentation, wie die einzelnen Tabellen verknüpft sind. Hier zeigt sich, dass das Datenmodell schwer verständlich und wenig anwender:innenfreundlich ist.
- H6: Bestätigt, allerdings zeigt sich, dass der Hinweis zu dieser Funktion zu dezent ist. Da die Tester die Testoberfläche alle kennen, rechnen Sie nicht mit neuen Funktionen und lesen den Kopf der OpenAPI-Swagger Dokumentation nicht aufmerksam. Zudem ist die Filterfunktion wenig intuitiv, man muss sich tiefer in die Dokumentation einarbeiten um zu verstehen, was man wie abfragen kann.

5.5 Weitere Inkremente

Neben den einzelnen Iterationen des Prototypen haben wir im Rahmen der Entwicklungsphase weitere Inkremente entwickelt. Die folgend aufgelisteten Dokumente schaffen direkt oder indirekt Mehrwert für die Handlungsfelder rund um die API. Sie sind nicht der Hauptfokus des Projektes, wurden von uns dennoch erstellt, da sie Mehrwert für die Gesamtunternehmung liefern. Einige dieser Inkremente konnten wir vervollständigen. Andere sind nicht vollständig. Unabhängig von dem Bearbeitungsstand am Ende unserer Projektlaufzeit, übergeben wir all diese Dokumente mit den Projektressourcen.

5.5.1 Deploymentskript & Dokumentation

Zielsetzung: Inbetriebnahme der Lösung sichern/ unterstützen

Bedürfnis BVL: Ressourcenschonendes Deployment

Mehrwert: Die BVL IT wird beim Test-Deploy unterstützt und kann das Deploy anschließend eigenständig durchführen. Darüber hinaus wird der Deploymentprozess klar dokumentiert und kann im Bedarfsfall reproduziert werden. Zusätzlich haben wir das Deploymentskript im Git-Repository des Projekts veröffentlicht um dem Open Source Gedanken des Fellowships nachzukommen. Es wird angestrebt, einen Oracle Dump mit zu veröffentlichen, sodass jede:r das Projekt aus dem Repository nachbauen kann.

5.5.2 Skript für die Inkonsistenzprüfung im BVL

Zielsetzung: Alternative für die Access Prüfroutinen schaffen

Bedürfnis BVL: Wenn Access abgelöst werden soll, wird eine Alternative für die Inkonsistenzprüfung benötigt

Mehrwert: Durch das SQL Skript können die Prüfroutinen in Access abgelöst werden. Es könnte die BVL IT dazu befähigen, eigenständig zu prüfen, ob die publizierten Daten Inkonsistenzen enthalten. Dadurch wird der Prüfschritt näher an die Entstehung der Daten geschoben, was die Fehlerfolgekosten reduziert. Aktuell fällt der Fehler erst kurz vor der Veröffentlichung auf, wodurch im Zweifelsfall viele Arbeitsschritte wiederholt werden müssen. Das Skript unterstützt lediglich die weiterhin notwendige fachliche Prüfung.

5.5.3 Problemfeld Maschinenlesbarkeit

Zielsetzung: Wunsch der Außenwelt nach "maschinenlesbaren Daten" verstehen und konkrete Handlungsmaßnahmen erarbeiten.

Bedürfnis Außen: "Maschinenlesbare Daten" um PSM Ausbringung zu automatisieren

Bedürfnis BVL: Klärung ob das BVL überhaupt "maschinenlesbare Daten" zur Verfügung stellen muss

Erkenntnis: Das Problem liegt tiefer in den nicht einfach verständlichen Auflagentexten. Hier ist zu viel Interpretationsspielraum für Mensch und Maschine. Durch den Ruf nach "Maschinenlesbaren Daten" wird sich eine Vereinfachung erhofft. Der Begriff selbst birgt jedoch viel Interpretationsspielraum (siehe [6.4.2 Umfrage zum Thema "Maschinenlesbarkeit"](#)). Das Problem ist größer als gedacht, im Rahmen des Fellowships kann nur ein Startpunkt aufgezeigt werden.

6 Ergebnis

6.1 Prototyp API

Im Rahmen des Tech4Germany Projektes ist der technische Prototyp einer API entstanden, der es dem BVL ermöglicht, weitere Erfahrungen zu sammeln. Ziel eines Prototypen ist es, konkretes Anwender:innenfeedback mit einer schnell zusammengestellten Lösung einzuholen. Alle weiteren Features sollten dann iterativ ergänzt werden in enger Abstimmung mit den Anwender:innen. Der Code kann in folgendem Repository eingesehen werden:

<https://github.com/tech4germany/psm.api.v1>

6.2 KPIs & Monitoring

Im Rahmen agiler Produktentwicklung werden Key Performance Indicators (kurz KPIs) genutzt um beim iterativen Vorgehen zu prüfen, ob der aktuelle Entwicklungsstand den gewünschten Mehrwert liefert. Sie sind also das Steuerungstool für kleinteilige Produktentwicklung und sollten regelmäßig konsultiert werden.

Es wird empfohlen mit einem Set von 4-6 KPIs zu starten, wobei eine davon den Fokus bildet und die anderen nur ergänzende Perspektiven abbilden sollen. Hier gilt der

Grundsatz "weniger ist mehr". Durch zu viele KPIs und dem Fehlen einer klaren Zielstruktur nimmt man sich die Macht der Entscheidungshilfe.

Im Laufe des Lebenszyklus eines Produkts können sich die KPIs auch verändern. Hier gilt es die jeweils richtige Metrik zum nächsten strategischen Ziel oder Entwicklungsschritt zu finden. Es ist ein Prozess, der mit wachsender Erfahrung feiner und granularer wird. Neben den KPIs sollten auch 1-3 Schwellenwerte und dazugehörige Handlungsempfehlungen (z.B. Gut - nichts tun / Gefährdet - Analyse / Schlecht - Handeln) definiert werden.

Die folgenden KPIs könnten für die unterschiedlichen Phasen genutzt werden:

Phase 1: Etablieren der API mit Prototyp

1. Reduzierte Anzahl der Prozessschritte zur Bereitstellung der Datenbank (BVL und Anwender:innenperspektive)
2. Anzahl unterschiedlicher Besucher auf der API (der genaue Wert hängt von der Kommunikation ab*)
3. Verhältnis Anzahl Access Nutzer:innen zu API Nutzer:innen (hier spielen Faktoren wie Gebühren und Kommunikation ein)

* Je Höher die Reichweite der Kommunikation desto mehr Nutzer:innen können potenziell auf der API landen. Wenn man nur 30-40 Leuten (der Abonnent:innenkreis) Bescheid sagt, sollte man nicht als Schwellenwert 50 Nutzer:innen definieren.

Phase 2: Entwicklung von API V2

→ Hängt vom Fokus der Weiterentwicklung ab (siehe [7 Nächste Schritte](#)). Im Zweifelsfall können die bestehenden KPIs übernommen und ergänzt werden.

Weiterführende Links:

<https://nordicapis.com/an-introduction-to-setting-kpis-for-your-api-strategy>

<https://apimetrics.io/understanding-api/api-kpis/>

<https://www.zdnet.com/article/10-key-performance-indicators-for-measuring-api-business-value/>

<https://cloud.google.com/blog/products/api-management/kpis-for-apis-12-key-metrics-for-api-programs>

Beispiel aus dem Verwaltungskontext von Gov.UK:

<https://www.gov.uk/performance>

<https://www.gov.uk/service-manual/measuring-success>

6.3 Prüfskript für interne Inkonsistenzprüfung

Ist-Zustand

Aktuell werden bereits Inkonsistenzprüfungen vor der monatlichen Veröffentlichung der Daten gemacht. Die Inkonsistenzprüfungen sind in Access implementiert und liefern momentan die Anzahl potenzieller Inkonsistenzen zu einem Thema, sowie eine detaillierte Ausgabe der betroffenen Datensätze als auch eine Kombination in Form eines Reports. Es handelt sich hierbei teilweise um lediglich potenzielle Inkonsistenzen, die eine fachliche Prüfung erfordern.

Soll-Zustand

Ziel war es diese Prüfungen in ein SQL Skript zu überführen. Der Informationsgehalt und die Filtermöglichkeiten sollen quantitativ und qualitativ nicht geringer sein als die bereits vorhandene Funktionalität. In einem ersten Schritt wurde ein SQL Skript erstellt, das die Übersicht über die Anzahl der verschiedenen potentiellen Inkonsistenzen in einer Datenbank-View darstellt. Analog dazu soll ebenfalls eine Detailansicht geschaffen werden. Die Ausgabe der Skripte kann in ein Excel-Dokument oder CSV überführt werden und könnte somit die Reports über Access ablösen. Es ist intern noch abschließend zu klären wer die Verantwortung für die Skripte übernimmt. Die Skripte könnten somit einen weiteren Beitrag unsererseits dazu leisten, unabhängig von der Access Lösung zu werden. Eine Evaluation der Adaption steht aus.

6.4 Vorarbeit Maschinenlesbarkeit

Im Rahmen des Projekts wurde ebenso auf das Thema "Maschinenlesbarkeit" eingegangen. Hier stellte sich heraus, dass verschiedene Motivationen und auch Definitionen dieses Begriffs existieren. Die vollständige Auflösung der internen Verständnisse aller Beteiligten war uns in der Projektlaufzeit nicht möglich.

6.4.1 Definition "Maschinenlesbarkeit"

Um ein gemeinsames Verständnis zu erhalten, sind wir mit unseren Projektpartnern zu dem Schluss gekommen, dass eine allgemeingültige Definition nicht möglich ist. Wir haben uns für den Rahmen des Projekts daher die folgende Definition erarbeitet. Es besteht kein Anspruch auf Vollständigkeit oder Allgemeingültigkeit. Ausgangspunkt dafür war die Definition, aus der Richtlinie (EU) 2019/1024 des europäischen Parlaments und des Rates:

"Maschinenlesbares Format" ist ein Dateiformat, das so strukturiert ist, dass Softwareanwendungen konkrete Daten, einschließlich einzelner Sachverhaltsdarstellungen und deren interner Struktur, leicht identifizieren, erkennen und extrahieren können"

Die von uns vorgeschlagene Definition gliedert sich in zwei Teile: um was handelt es sich bei der "Maschine", die lesen soll und was bedeutet "lesbar". Da der Begriff der Maschinenlesbarkeit vorgeprägt ist, führen wir mehrere neue Begriffskombinationen ein:

Der Teilbegriff "Maschinen-" wird ersetzt durch:

- Landmaschinen - ein System, das direkt und ohne Veränderung die Daten verstehen muss.
- PC - ein System, auf welchem Nutzer:innen die Daten verändern können.

Das Einlesen auf eine Landmaschine erfolgt vollautomatisch. Es können im Gegensatz zu einem PC keine Änderungen an den Daten vorgenommen werden. Die Daten müssen direkt in einer Form vorliegen, die von Landmaschinen interpretierbar ist.

Der Teilbegriff "-lesbarkeit" wird ersetzt durch:

- Einlesbarkeit - die Daten liegen digital vor und können in das System geladen werden.
- Verarbeitbarkeit - die Daten liegen in solcher Struktur vor, dass sie ohne Transformation in andere Strukturen überführt werden kann.
- Interpretierbarkeit - die Daten sind syntaktisch und semantisch derart aufbereitet, dass Maschinenbefehle aus ihnen abgeleitet werden können.

Um Interpretierbarkeit zu erreichen, müssen drei Datenaufbereitungsstufen genommen werden:

1. Boolescher Wert - Existiert eine Auflage? Ja oder nein.
2. Integer - Welche Werte hat eine Auflage? 5 Meter.
3. Semantik - Welche Abhängigkeiten hat eine Auflage? Nur wenn die Sonne scheint.

Mit dieser Definition können wir einheitlicher beschreiben, was mit dem Begriff "Maschinenlesbar" in einem gewissen Kontext gemeint ist. Die Definition stellt allerdings keine vollständige Abbildung aller Möglichkeiten dar, der Begriff der "Maschinenlesbarkeit" lässt weiterhin Interpretationsspielraum.

6.4.2 Umfrage zum Thema "Maschinenlesbarkeit"

Um ein besseres Gefühl für die Sichtweise der Anwender:innen zum Thema "Maschinenlesbarkeit" zu bekommen, haben wir eine qualitative Umfrage durchgeführt. Diese Umfrage wurde an den Kreis der Projektinteressierten und Interviewpartner:innen versendet. Während die Begriffe "Maschinenlesbarkeit" und "Interpretierbarkeit" bei Beantwortung der Befragung synonym genutzt wurden, waren sich mehr als zwei Drittel der Befragten einig, dass die Daten in der aktuellen Form nicht "maschinenlesbar" sind. Beispiele einer Definition des Begriffs der Maschinenlesbarkeit durch die Befragten, sowie eine detaillierte Auswertung der Umfrage selbst liegt den Projektressourcen bei.

6.4.3 Prozessvorschlag - Definition eines Schemas

Dieser Abschnitt soll eine Handlungsempfehlung für die weitere Bearbeitung der Struktur liefern. Es handelt sich um eine grobe Richtlinie, welche für drei Auflagen ausgearbeitet wurde. Sie dient als Orientierung für die Ausarbeitung einer Struktur, welche für den aktuellen Satz an Auflagen gültig ist. Zur eigenständigen Behandlung dieser Handlungsempfehlungen benötigt die bearbeitende Person bestmöglich Verständnis für hierarchische Strukturen, technisches und fachliches Verständnis.

Komplexitätstreiber bei den vorhandenen Auflagen sind die juristischen Auflagentexte. Diese sind von Mensch und Maschine schwer interpretierbar. Neben der besseren

Verständlichkeit für Maschinen fördert die Aufbereitung auch eine bessere Zugänglichkeit der Informationen für die Menschen, die damit arbeiten.

Als erstes sollte sich eine neue bearbeitende Person ein Grundverständnis über die Struktur verschaffen. Die Struktur ist hierarchisch, aber nicht technisch. Die Klammern dienen lediglich zur Visualisierung.

Es sollten alle extrahierbaren Informationen in das Schema überführt werden, auch wenn diese zur Zeit keine technische Anwendung finden. Ziel ist es eine Liste an Anwendungen zu schaffen, welche theoretisch in eine Datenbank- oder Kommunikationsstruktur überführt werden kann.

Das Skelett des aktuellen Schemas ist:

```
ANWENDUNG: {
  IF Bedingung der Auflage (Wann gilt die Auflage): {
    Bedingung Typ: TYP {
      Auflage gilt explizit wenn (Explicit): {
      }
      Auflage gilt nicht wenn (Exclude): {
      }
    }
  }
  THEN Auflagen: {
  }
}
```

Der empfohlene Ablauf ist wie folgt:

1. Definieren Sie eine kleine Auswahl an Auflagen. Dazu wurde im Rahmen unseres Projekts schon eine mögliche Auswahl genannt. Diese finden Sie in den Projektressourcen.
2. Entwickeln sie ein hierarchisches Schema für die Auflagen. Achten Sie dabei darauf mögliche Dopplungen ("A gilt wenn C, B gilt wenn C" heißt "wenn C: gilt A und B") zu konsolidieren und Inhalte kleinstmöglich (z.B. Wert und Einheit einzeln) abzubilden. Sollte sich eine Auflage nicht abbilden lassen, brechen Sie nach Ermessen ab. Liefern Sie eine Begründung.
3. Überführen Sie die neu ausgearbeiteten Auflagen in das Gesamtdokument.
4. Prüfen Sie, ob das Gesamtschema noch anwendbar ist und iterieren Sie über alle Auflagen, ob ein Feld hinzugefügt werden muss oder ähnliches.

5. Iterieren Sie und passen das Schema kontinuierlich an. Unter Umständen wird das Schema dabei sehr komplex um alle Fälle in den Daten abzubilden.

6.5 Übersicht: Arbeitspakete hinter einer API

Ein Teilziel unseres Projektes ist, eine Blaupause für andere Projekte dieser Art im BVL/BMEL zu liefern. Es gibt noch viele weitere Datenbanken, die per API zur Verfügung gestellt werden sollen. Daher haben wir hier aus der Erfahrung unseres Projekts heraus, eine Auflistung der möglicher Arbeitspakete für die Implementierung solcher APIs im BVL zusammengestellt.

Die folgende Darstellung ist sehr simplifiziert. Das sind die groben Schritte und Fragen, die man sich in jedem Fall bei der Implementierung einer API stellen sollte. Dabei erheben wir jedoch keinen Anspruch auf Vollständigkeit.

Einige dieser Schritte dauern viel länger als man annehmen würde. Um beispielsweise die Daten in eine "landmaschineninterpretierbare" Form zu bringen (hier als ein Stichpunkt dargestellt), wird viel Arbeit benötigt, um die Datenstruktur herunterzubrechen, vorhandene Fließtexte zu interpretieren und in eine allgemein gültige Datenstruktur zu überführen, die alle Anwendungsfälle abbildet.

1. Zuerst sollte man sich innerhalb der Behörde, im Austausch mit Nutzer:innen, bewusst werden, was das Ziel ist:
 - a. Die reine Bereitstellung per Schnittstelle (sofern noch nicht vorhanden) von Daten in der bereits existierenden Form, kann ein guter Schritt sein und ermöglicht Anwender:innen einen "sanften" Übergang.
 - b. Die Bereitstellung von Daten in einer aufgearbeiteten Form (z.B. landmaschinen-interpretierbar, in intuitiver Struktur). Für diese Entscheidung sollten in jedem Fall Anwender:innen befragt werden, ob sie eine andere Struktur brauchen als die, die aktuell vorliegt.
2. Wenn man sich dessen bewusst ist, kann es darum gehen, eine geeignete Form der API zu finden. Es kann auch von Vorteil sein, verschiedene Systeme zu haben, die auf eine Bereitstellungsform spezialisiert sind.
 - a. Reine Bereitstellung
 - i. Relationale (Oracle?) Datenbank: analoges Vorgehen zu unserem Projekt. Bereitstellung über REST, bspw. via ORDS oder per Java Application. Dokumentation via OpenAPI-Swagger.
 - ii. Graphdatenbank: Bereitstellung via SparQL Endpoint.

b. Aufgearbeitete Bereitstellung

i. Nutzer:innen Befragung + Testing:

- Welche sinnvollen Endpunkte soll es geben? Wo steigen die Nutzer:innen also ein? (Beispiel PSM: meist über den Endpunkt /mittel, aber auch über eine Indikation)
- Welche Attribute braucht man in den verschiedenen Endpunkten?
- Welche Parameter sind sinnvoll, um direkt nach ihnen filtern zu können?
- Dient die API primär Suchzwecken, können die Daten für die Suche aufbereitet werden bspw. durch Denormalsierung bei strukturierten Daten oder durch den Einsatz einer SearchEngine (zB. Elasticsearch)
- Statt vordefinierte Endpunkte bereitzustellen, kann auch mit der Technologie GraphQL (muss nicht nur auf Graphdatenbanken aufgesetzt werden) experimentiert werden. Hier können sich Nutzer:innen ihre Endpunkte eigenständig "zusammenstellen".

ii. Content Arbeit

- Welche Informationen liegen noch nicht in gewünschter Form vor?
- Zeitstempel an Datensätzen (created, updated)?
- Wie sieht die gewünschte Form aus? → Schema entwickeln (sehr eng mit Datenbankdesign verbunden)
- Die identifizierten Stellen in das entwickelte Schema bringen (gerne erst in Excel o.ä. um diese neue Struktur dann testen zu können: Wird sie von Nutzer:innen verstanden? In der Form gebraucht?)

iii. In der unterliegenden Datenbank selbst (empfohlen) oder in der API Bereitstellungssoftware die identifizierten Stellen tatsächlich programmatisch anpassen + anreichern.

3. Es sollte eine Weiterentwicklungsstrategie entwickelt werden, um die Modernität der Schnittstelle zu gewährleisten.
4. Wer übernimmt Verantwortung für was? Zuständigkeiten der (Weiter-) Entwicklung, Wartung und Support sollten geklärt sein.

5. Es sollte eine Versionierungsstrategie geben, um bei Anwender:innen keine Fehlermeldungen zu verursachen, wenn sich die Version der API ändert: Wann gibt es eine neue Version der API? Wie wird diese abgebildet? Wie wird kommuniziert, dass es eine neue Version gibt? Wann werden alte Versionen nicht mehr unterstützt?
6. Es sollte eine Kommunikationsstrategie geben, um Anwender:innen auf dem Laufenden zu halten:
 - Auf welchem Weg wird den Nutzer:innen mitgeteilt, dass es die API gibt?
 - Wie kommen sie auf die entsprechende Seite?
 - Auf welchem Weg geschieht die Rückkopplung, also das Feedback an die Lösung?
 - Gibt es Release Notes, wer schreibt diese?
 - Wie wird der Stand der Daten/Aktualität kommuniziert?
7. Sicherheitsmaßnahmen sollten ergriffen werden:
 - Müssen Daten aufgrund von Datenschutz ausgeschlossen werden?
 - Braucht es Rate Limits um DoS Attacken zu verhindern?
 - Ist vor SQL Injection geschützt?
 - Stress-Tests sollten durchgeführt werden.
8. Technische Details:
 - Was passiert wenn die Schnittstelle "Scale" erreicht?
 - Pagination sollte von Anfang an implementiert sein.
 - Wartbarkeit durch Modularisierung?
 - Finden sich genügend Entwickler für die Lösung (Aktualität der Programmiersprache)?
9. Letztendlich muss die API implementiert und ins bestehende System integriert werden.
10. Es wird eine Abnahme durch die IT und die Fachseite benötigt.
11. Es sollte eine gute, ausführliche Dokumentation der Schnittstelle und aller Endpunkte geben, wir empfehlen dafür OpenAPI-Swagger zu nutzen, wodurch der [OpenAPI Standard](#) umgesetzt wird, um die Dokumentation direkt bei der API verfügbar zu haben.
12. Die API sollte sobald sich das unterliegende Datenmodell ändert, ebenfalls geändert werden und eine neue Version zugeschrieben bekommen. Alte Versionen sollten für eine Übergangszeit verfügbar bleiben.

7 Nächste Schritte

Im Rahmen der Zwischenpräsentation haben wir eine Übung mit den anwesenden Kollegen aus Behörde und Ministerium gemacht. Nach Vorbild des "Amazon Press Release" wurden die Teilnehmer darum gebeten zu unterschiedlichen Zeitpunkten ihre Vision für die Datenbereitstellung im BVL (in 5 Jahren), ihr Schreckensszenario (in 5 Jahren) und ihre realistische Vorstellung für das Tech4Germany Projekt festzuhalten und zu teilen. Details zu den einzelnen Beiträgen der Teilnehmer finden sich in anonymisierter Form in den Projektunterlagen. Bemerkenswert war, dass die größte Angst bei der Teilmenge der Beteiligten, die uns ihre Visionen bereitgestellt haben, jene war, dass **"wieder nichts passiert"**. Als Ursachen können hier die Prozesse in der Verwaltung herausgelesen werden und die Herangehensweise an IT Projekte im Wasserfallmodell. In diesem Abschnitt möchten wir konkrete Handlungsempfehlungen teilen, um Wege und Möglichkeiten in den von uns behandelten Themen aufzuzeigen in kleinen Schritten voran zu kommen.

7.1 Unmittelbare Folgen aus unserem Projekt

7.1.1 Prototyp veröffentlichen

Der Deploymentprozess der von uns entwickelten API wurde bereits während unseres Projektes in die Wege geleitet und zu großen Teilen umgesetzt. Die ausstehenden Schritte und weitere Empfehlungen sind im Folgenden aufgelistet.

7.1.1.1 Prozess

Wie im Gespräch mit der BVL-IT am 06.10.20 abgesprochen, sind die bis zur Produktivnahme vorzunehmenden Schritte:

- Unit-Tests prüfen und einbauen
- Definieren, wo auf der Seite der Link zur API veröffentlicht wird (Pressestelle / Online Kommunikation auf Anweisung der Fachseite)
- Content für die BVL Website erstellen (Pressestelle auf Anweisung der Fachseite)
- Finale Bestätigung der Fachseite, dass es deployed & veröffentlicht werden darf

- Geplant ist die Bereitstellung des aktuellen Prototyps durch Tech4Germany mit einem **nicht aktualisierbaren** Stand von August. Die Bereitstellung erfolgt maximal bis die offizielle Freischaltung durch das BVL erfolgt ist. Zu finden ist der Prototyp unter folgender Adresse: <https://psm.tech4germany.org/>

Nach fachlicher Freigabe:

- Domain anmelden (2 Tage)
- Abschließendes Deployment auf einem öffentlich zugänglichen System

Testbetrieb:

- Monatliche Aktualisierung → Die API kann in den Aktualisierungsprozess der BVL-PSM-Websuche integriert werden. Wird die Datenbank der Weboberfläche aktualisiert, so bekommt auch die API einen neuen Stand. Dies spart Ressourcen.

7.1.1.2 Kommunikation & Markteinführung

Um die Erwartungshaltung der Öffentlichkeit zu steuern, macht es Sinn, eine klare Kommunikationsstrategie zu entwickeln. Durch transparente Kommunikation kann dem Entstehen von falschen Erwartungshaltungen entgegengewirkt werden. Durch das Tech4Germany Projekt ist die interessierte Öffentlichkeit aufmerksamer als üblich.

Wir empfehlen daher eine achtsame Kommunikation rund um die Veröffentlichung des Prototypen. Im Folgenden möchten wir einige Ideen teilen, wie eine solche Kommunikationsstrategie aussehen könnte.

Kernaussage: Das BVL hat eine simple API in den Testbetrieb genommen um Erfahrungen im Bereitstellen der Daten via API zu sammeln. Diese Information sollte auch an alle aktuellen Bezieher:innen der Access-Datenbank verteilt werden.

Ergänzende Aussagen:

- Der Testbetrieb ist zunächst auf die Dauer von X Monaten begrenzt.
- Die API wird in dieser Zeit nicht geplant weiterentwickelt
- Am Ende der X Monate werden die Erkenntnisse zusammengetragen und entschieden wie es weitergeht

Phase 1: Einführung - Testbetrieb

Im ersten Schritt sollten nur die aktuellen Bezieher:innen des Access Dumps über die neue API informiert werden. Hierdurch kann zum Beispiel in den ersten 4 Wochen ein Test des Betriebs erfolgen. Die Außenkommunikation sollte aufzeigen, dass die API eine Alternative zum aktuellen Access Dump ist.

Die Anwender:innen werden das Bedürfnis haben, Feedback zu geben. Durch das Anbieten eines Kanals kann das Anwender:innenfeedback kanalisiert und effizient ausgewertet werden. Gängige Methoden hierfür sind Feedbackformulare, man kann aber auch eine extra E-Mailadresse hierfür einrichten. Im Rahmen der Einführung kann auch der Feedbackkanal erprobt und ggf. noch nachgesteuert werden. Am Ende der Phase sollte eine Reflexion stehen, in der geprüft wurde, ob die Kommunikation erfolgreich war.

Phase 2: Skalierung - Testbetrieb

Nachdem die Einführung mit den Stammnutzer:innen erfolgreich abgeschlossen ist und die Stolpersteine aus dem Weg geräumt wurden, sollte die Kommunikation breiter aufgestellt werden. Zu diesem Zeitpunkt ergibt es Sinn Kanäle zu nutzen, die eine große Reichweite haben und die erweiterte Zielgruppe der interessierten Branche und Öffentlichkeit adressieren. Mögliche Kanäle können die BVL Website und Pressemitteilungen sein. Darüber hinaus könnte man die Interessenverbände und Landesorganisationen als Multiplikatoren einschließen.

Damit die Kommunikation nicht missverständlich ist, sollte sie transparent das Vorhaben darstellen und den Leser:innen eine klare Aufgabe adressieren (wie zum Beispiel "Jetzt testen" oder "an interessierte Kolleg:innen aus der IT weiterleiten").

Losgelöst von der Kommunikationsstrategie muss noch konkret definiert werden, wie und wo der Link zum Prototypen der API auf der BVL Seite platziert wird. Aktuell ist die Nutzer:innenführung recht unübersichtlich. Im Kern haben die Nutzer:innen künftig 3 Optionen: (1) Web-Suche Gesamtsuche, (2) Web-Suche Schrittweise (3) API-Prototyp. Diese 3 Optionen sollten klar werden. Zudem sollte darauf geachtet werden, dass die Unterseite der API für Suchmaschinen optimiert ist. Sofern die Neuigkeit sich verbreitet, ist es ein realistisches Szenario, dass interessierte Anwender:innen versuchen via Google weitere Informationen zu bekommen.

7.1.2 Nutzungsanalysen

Um pragmatisch Transparenz zu schaffen, kann auf dem Webserver ein Log Analyzer installiert werden, der Informationen zur Nutzung der API und der OpenAPI-Swagger-Dokumentation liefert. Um die Adaption der Schnittstellen oder auch der Websuche zu messen. Ein solches Tool visualisiert den Inhalt der Server Logs in grafischer Form. Es werden keine zusätzlichen Informationen gesammelt oder verarbeitet.

Ein Tool unserer Wahl ist <https://goaccess.io>. Eine Anleitung zur Installation und Einrichtung ist auf der Website gegeben. So können Informationen sowohl über Webseitenaufrufe, wie auch über API-Aufrufe durch verschiedene Endpunkte visualisiert werden. Selbiges gilt auch bei Aktivierung für die bereits bestehenden Websuchen.

Es können die folgenden Informationen von Interesse sein:

- Visitors: Gibt eine Auskunft über die Aufrufe auf der Website/API, welche IPs machen besonders viele Aufrufe.
- Welche Endpunkte abgefragt wurden: Welche Mittel/Auflagen werden öfter abgefragt?
- Welche Website besucht wurde: Wie oft wird die OpenAPI-Swagger Dokumentation abgerufen, wie oft die API selbst?
- Hits: Wie viele Anfragen wurden gestellt?
- Datenvolumen: Wie viele Megabyte an Traffic wurden über die Schnittstelle verursacht?
- Geolocation: Woher kommen die Website Nutzer:innen und gibt es Bots, die gesperrt werden müssen?
- Zeitliches: Wann wird abgefragt und gibt es besondere Zeiten mit hoher Last?
- Wird z.B. monatlich wenn die neuen Zulassungen zur Verfügung gestellt werden abgefragt oder im Verlauf jedes einzelnen Tages?

Zur **Fehleranalyse** kann genutzt werden, wie oft Anfragen geklappt haben und mit welchem Response Code sie quittiert wurden.

Datenschutz: Das Tool visualisiert nur bereits vorhandene Informationen aus den Server Logs. Es sammelt nichts, was nicht ohnehin schon da ist. Sobald die Log-Datei

gelöscht wird, sind die Informationen auch aus dem Tool gelöscht. Es werden keine historischen Daten gespeichert. Eine Anwendbarkeit sollte vom BVL evaluiert werden.

7.1.3 Access ablösen

Wenn die API freigeschaltet wird, erwarten wir aufgrund des Feedbacks der Nutzer:inneninterviews, dass nach und nach die Prozesse der Anwender:innen auf die API umgestellt werden. So werden Sie den monatlichen Access Dump abbestellen, da durch die API die gleichen Daten zur Verfügung stehen. Für einen noch zu bestimmenden Zeitraum, der klar kommuniziert werden sollte, sollte die API parallel zum Access Dump zur Verfügung stehen. Wir schlagen ein Jahr vor. In diesem Zeitraum können sich Konfigurationen, wie beispielsweise Rate Limits einpendeln und die Bezieher:innen schrittweise ihren Prozess anpassen. In dieser Übergangszeit wird Mehraufwand entstehen. Dieser sollte durch Ablösung von Access langfristig mindestens kompensiert werden.

Sobald sich die Zahl der Access Bezieher:innen auf einen hinreichend kleinen Wert reduziert hat, schlagen wir vor, den Prozess der monatlichen Update E-Mails mit Access Dump abzuschaffen und stattdessen komplett auf die API umzusteigen.

Zusätzlich zur Bereitstellung der Datenbank per E-Mail, wird Access auch an weiteren Stellen verwendet, die uns wie folgt mitgeteilt wurden: BVL-extern wird die Access-Datenbank noch monatlich in das FIS-VL (<https://fis-vl.bvl.bund.de>) hochgeladen und Behörden zur Erfüllung hoheitlicher Aufgaben unentgeltlich zur Verfügung/zum Download bereitgestellt. Alle diese Verwendungen können unserer Einschätzung nach durch die API abgerufen werden und benötigen keine Repräsentation als Access. In den allermeisten Fällen, lassen sich die Anfragen durch einen einzigen Request an einen Endpunkt abbilden. So könnte auch die interne Kommunikation über vollautomatische Schnittstellen laufen.

7.2 Weiterreichende Schritte

7.2.1 API weiterentwickeln

Da es perspektivisch nicht bei einem Prototypen bleiben soll, folgt noch eine Zusammenfassung der Wünsche der Anwender:innen und der weiteren Ideen wo und

wie zukünftig Weiterentwicklung stattfinden könnte und was dafür noch hilfreich sein kann.

Eine Digitalstrategie ist ein guter Startpunkt um vorweg eine längerfristige Richtung abzustecken. Gibt es bereits eine Digitalstrategie in die sich das Projekt eingliedern lässt? Gibt es darüber hinaus schon eine Strategie zur Weiterentwicklung der Schnittstelle? Teil einer solchen Strategie für die Schnittstelle könnte es bspw. sein, die Konzeption und Anpassung der Schnittstelle nach größtmöglichen Mehrwert für alle Beteiligten bzw. Betroffenen in den Vordergrund zu stellen. Umsetzung und Arbeitspakete lassen sich einfacher und zielgerichtet auf einer gemeinsam geteilten Strategie definieren. Um die Weiterentwicklung der API langfristig voranzutreiben, braucht es weiterhin ein Team, das sich (Vollzeit) mit diesem Thema beschäftigt. Dieses Team muss noch gebildet werden.

Generell sollten alle Änderungen an der Schnittstelle versioniert werden, d.h., die Veröffentlichung von einer durch Features erweiterten Schnittstelle sollte immer zu einer neuen Versionsnummer führen. Verschiedene Versionen können so (bspw. für eine notwendige Adaptionszeit) parallel betrieben werden. Kleine Schritte halten den zusätzlichen Aufwand für nachgelagerte Anpassung gering und bieten Gelegenheit möglichst konkretes Feedback zu bekommen.

Ein Beispiel für konkretes Feedback aus den Nutzer:innentests: Es sei gut zunächst nur den Kanal der Bereitstellung von E-Mail auf Web-Schnittstelle zu ändern, dadurch bliebe der Aufwand für den Umstieg überschaubarer.

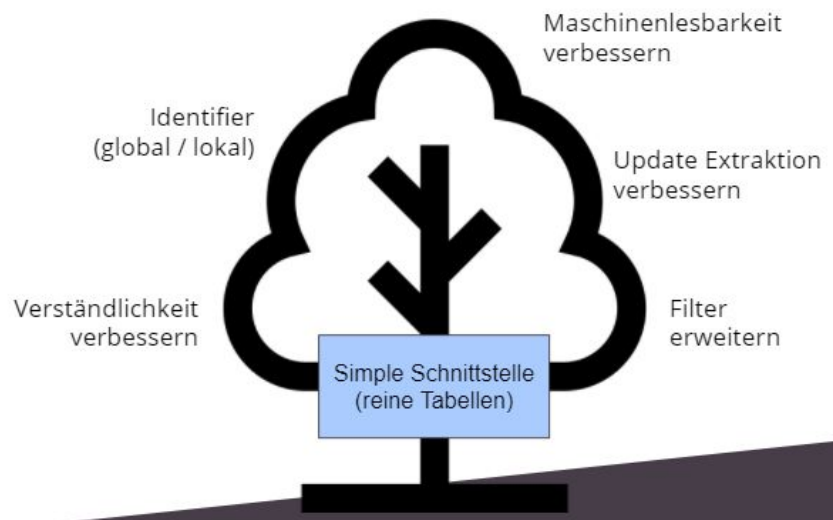
Konkrete Wünsche und Featureideen von Nutzer:innen:

1. Ein Endpunkt zur Abfrage der Änderungen seit einem bestimmten Datum (Delta) würde die zu übertragende Datenmenge zwecks Update anderer Systeme verringern.
2. Zwei Zeitstempel an jedem Datensatz: "Erstelldatum" und "Datum der letzten Aktualisierung" ermöglichen jedem/jeder, sich ein eigenes Delta für die Daten, die von interesse sind zu bilden (Ähnlich wie Punkt 1).
3. Die einleitende "allgemeine" Dokumentation auf dem OpenAPI-Swagger Interface interessanter gestalten, sodass sie auch gelesen wird. In Nutzer:innentests

wurde das Feature der Json Queries, die viel mächtiger sind als die einzelnen Parameter Filter, wiederholt nicht gefunden.

4. "Vertriebsfirmen" werden nicht beim Endpunkt ADRESSE gesucht. Hier ist also eine Umbenennung angeraten in bspw. "Firma" oder "Company".
5. Die Datenstruktur wurde als kompliziert wahrgenommen, weil häufig Tabellen zusammengefügt werden müssen. (Zitat: "Habe die Befürchtung das ist wieder nur eine Verknüpfungstabelle")

Um die API weiterzuentwickeln, gibt es zusätzlich noch Ideen, die wir bereits in der Zwischenpräsentation vorgestellt haben. Einige davon sind strukturverändernde Maßnahmen, die für Nachhaltigkeit zuerst in Infozupf umgesetzt werden sollten, statt sie im Nachgang zu verändern.



Konkrete Vorschläge für Anpassungen:

- **Sprechende Namen** für Eigenschaften / Spalten / Tabellennamen
Bspw. ist nicht sofort erkenntlich wofür acronyme wie "AWG" oder "HUK" stehen.
- **Konsequente Verwendung von Identifiern** und deren Veröffentlichung inkl. einer Beschreibung für die Anwender:innen. Als Beispiel können hier die Identifizier aus dem Kulturbaum oder die Identifizier der Schadorganismen

genannt werden. Der Kulturbaum verwendet EPPO Codes als Identifier, wird aber zusätzlich durch interne eigene Codes ergänzt, eine Dokumentation der daraus entstehenden Gesamtmenge an einem zentral zugänglichen Ort fehlt. So können auch an anderen Stellen allgemeingültige Identifier eingeführt werden.

- **Zeitstempel** für das Erstellungsdatum und den Aktualisierungszeitpunkt zu jedem Datensatz einführen (zwei neue Spalten je Tabelle).
- **Views** für häufig zusammenhängende Abfragen erstellen (bspw. wenn Tabellen zusammengefügt werden müssen um an konkrete Informationen zu kommen).

Die Anforderungen von Externen an die Form der veröffentlichten Daten sind so divers, dass es schwierig sein wird, alle zufriedenzustellen. Tendenziell ist es natürlich im Sinne des nutzer:innenzentrierten Vorgehens für Anwendungsfälle im einzelnen speziell zu optimieren. Es ist jedoch ratsam, vorher zu evaluieren, welche der möglichen Optimierungen den meisten Mehrwert in der Fläche liefern (Low Hanging Fruits).

Einzelne Anwendungsfälle für die Optimiert werden kann:

- **Such- und Filtermöglichkeiten**

Wie schon erwähnt, bildet die aktuelle Websuche die Wünsche der Anwender:innen nur lückenhaft ab. Für eine Optimierung ist es ratsam, noch etwas tiefer zu forschen. Es kann in Erfahrung gebracht werden, ob sich das Zusammenlegen von Tabellen bspw. in Datenbank-Views schon lohnt, oder ob der Einsatz einer Suchmaschinentechologie mehr Sinn ergibt.

- **Vollständige Datensätze** beinhalten alle expliziten sowie impliziten Daten, die zu einem Mittel existieren. In der Websuche sind bspw. implizite Auflagen durch Hierarchien (Vererbung der Auflage durch eine übergeordnete Kultur) mit einer einzelnen Suchanfrage nicht zu finden. Auf der Datenbank können solche Baumstrukturen ggf. durch rekursive Abfragen ermittelt werden, über die Schnittstelle ist dies aktuell nur durch viele Requests abzubilden. Derartige Baumstrukturen wären in einem vollständigen Datensatz nachvollziehbar.

- **Maschinenlesbarkeit verbessern**

Hierbei ging es primär um die juristischen Auflagentexte in denen bspw. die Anwendungsaufgaben zu einem Mittel festgelegt sind. Herausforderung hierbei

ist es, diese Texte in ein Schema zu überführen, welches möglichst keine Interpretation der Auflage mehr zulässt. Es soll klar aus einzelnen atomaren Eigenschaften ablesbar sein wer? wann? was? unter welchen Auflagen etwas darf bzw. nicht darf.

- **Semantic Data** ist ein weiterer Ansatz, Daten für Maschinen verständlicher aufzubereiten. Semantik versucht bspw. über global eindeutige Identifier die ggf. durch die Verwendung oder Schaffung von Ontologien gebildet werden, die Verarbeitung durch Maschinen zu erleichtern. Die PAM Projekte in denen JKI und KTBL involviert waren, hat hier bereits Erfahrung mit der Aufbereitung von Abstandsaufgaben für Precision-Farming. Das KTBL hat diesbezüglich auch Erfahrung mit der Herausgabe der Daten in einem JSON Format.

Außerhalb der konkreten Vorschläge und Ideen sollten vorhandene und übliche Lösungen betrachtet werden, um ggf. Ideen zu adaptieren und das Verständnis für potenzielle Anwender:innen zu schärfen. So stellt beispielsweise das Dienstleistungszentrum Ländlicher Raum RheinPfalz (DLR RLP) als untere Landesbehörde die Website pflanzen-schutz-information.de bereit, welche die Zulassungsdaten für die Anwender:innen verständlicher darstellt. Dies sei erwähnt, da es sich um eine behördliche Bereitstellung handelt. Es existieren weitere privatwirtschaftliche Lösungen. Das DLR RLP stellt in einem Dokument die Nutzung und Probleme mit der aktuellen PSM-DB dar. Dieses Dokument liegt den Projektressourcen bei.

7.2.2 Über eine API hinaus

Abgeleitet aus den Handlungsfeldern, haben wir uns auch mit anderen Bereichen beschäftigt als der API v1 und v2. Deshalb sind die nächsten Grafiken farbcodiert nach Handlungsfeldern auf die sie einzahlen:

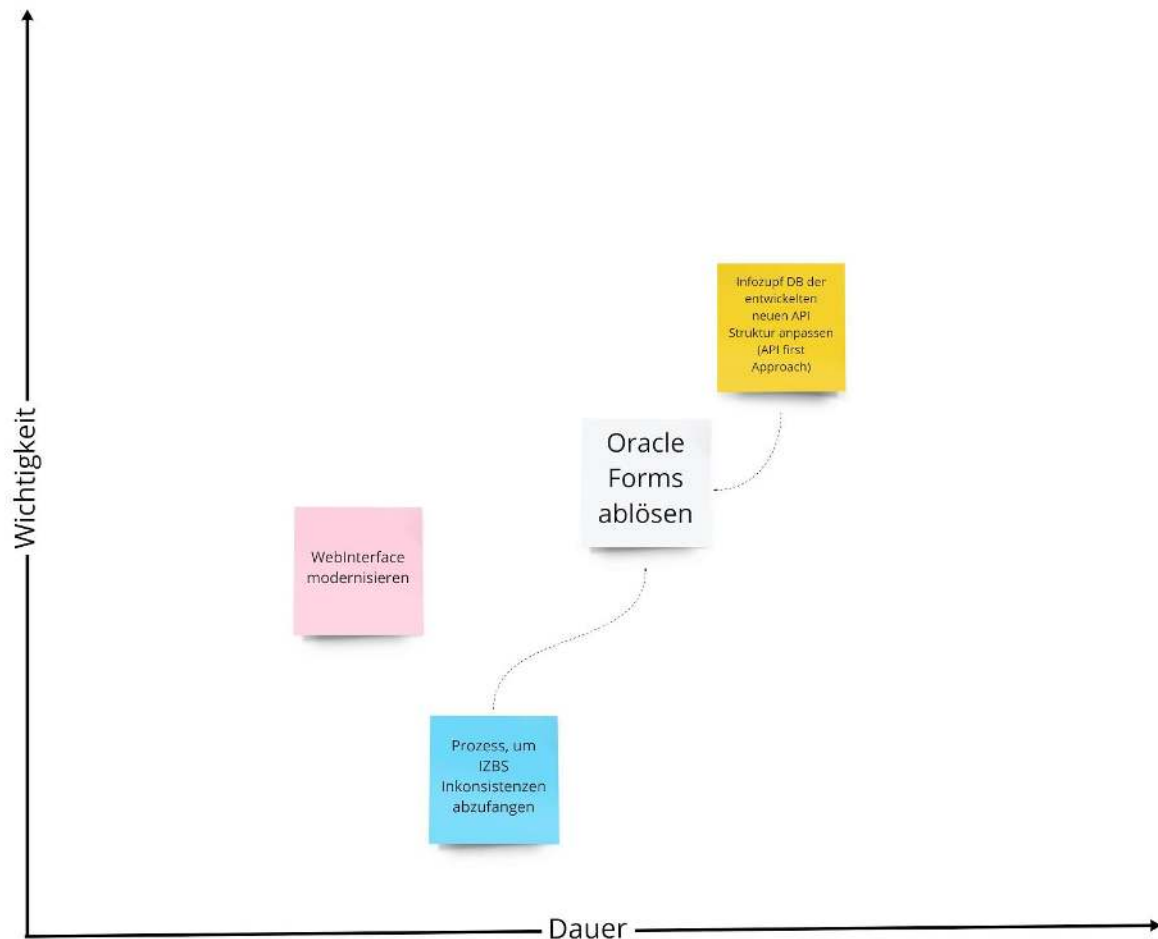


Die Grafiken sind dabei folgendermaßen zu lesen: Je höher eine Handlung (ein Post-It) angesiedelt ist, desto weitreichender schätzen wir es in seiner Bedeutung ein. Je weiter rechts es ist, desto länger dauert die Bearbeitung der Handlung unserer Einschätzung nach. Weiterhin haben wir Abhängigkeiten (durchgezogene Pfeile) und Synergien (gestrichelte Pfeile) zwischen den einzelnen Handlungen eingezeichnet, die den Zusammenhang darstellen.

7.2.2.1 Infozupf Weiter- / Neuentwicklung

Aus einem angepassten Datenmodell für die Weiterentwicklung der API kann es angeraten oder notwendig werden, auch das Datenmodell in Infozupf und damit die Anwendung Infozupf, als Möglichkeit der Dateneingabe, anzupassen. Dabei sollte evaluiert werden, ob eine Neuentwicklung wirtschaftlich ist, bzw. ob der aktuelle Zulassungsprozess und das angepasste Datenmodell noch zusammen passen.

Es wurde aus dem BVL an uns herangetragen, dass die Infozupf Anwendung als veraltet gilt. Von außerhalb haben wir zudem gehört, dass die Datenstruktur kontraintuitiv ist und überarbeitet werden sollte. Wir empfehlen daher dringend das Thema Infozupf und die dahinterliegende Datenbank so schnell wie möglich anzugehen.



Für die Detailansicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:
https://miro.com/app/board/o9J_kkPogK4=?moveToWidget=3074457350613385888&cot=12

Erläuterungen:

- Der erste Schritt könnte dabei sein, einen neuen Prozess zu etablieren, der die vorhandenen Konsistenzprüfungen erleichtert/obsolet macht. Mehr dazu wie dieser aussehen kann, wird weiter unten beschrieben.
- Der konsequente Einsatz von Fremdschlüsseln zur Wahrung der referentiellen Integrität kann in vielen Fällen verhindern, dass Inkonsistenzen überhaupt erst entstehen.
- Da Oracle Forms, worauf die aktuelle Anwendung basiert, bald nicht mehr unterstützt wird, sollte auch die Eingabe der Daten in die Infozupf Datenbank überarbeitet werden. Hierzu Zitat von Oracle: Oracle Forms durch Oracle APEX ablösen: "Oracle hat bereits im Dezember 2010 den Support für Oracle Forms 10g beendet."

Wechsel auf Oracle Reports: "Bereits im Oktober 2016 hat Oracle angekündigt, dass Oracle Fusion Middleware 12.2.1.3.0 das letzte Reports-Release sein wird. Zudem erlischt spätestens 2025 (2023) vollständig der Support-Anspruch seitens Oracle."

- Sobald eine neue Datenstruktur entwickelt wurde (siehe "7.5.1 Konkrete Handlungsempfehlungen zur API"), kann dieses für die API entwickelte Datenmodell (nach dem modernen API-first Ansatz) auch in Infozupf etabliert werden.
- Die aktuelle Websuche ist nicht modern und nicht für die Nutzung durch mobile Geräte konzipiert und kann daher auch, dass ein Großteil der Websuchen mittlerweile von Smartphones oder Tablets getätigt werden, nicht gerecht werden. Auf die API kann eine moderne Websuche aufgesetzt werden.

Mehr zum vorgeschlagenen Prüfungsprozess:

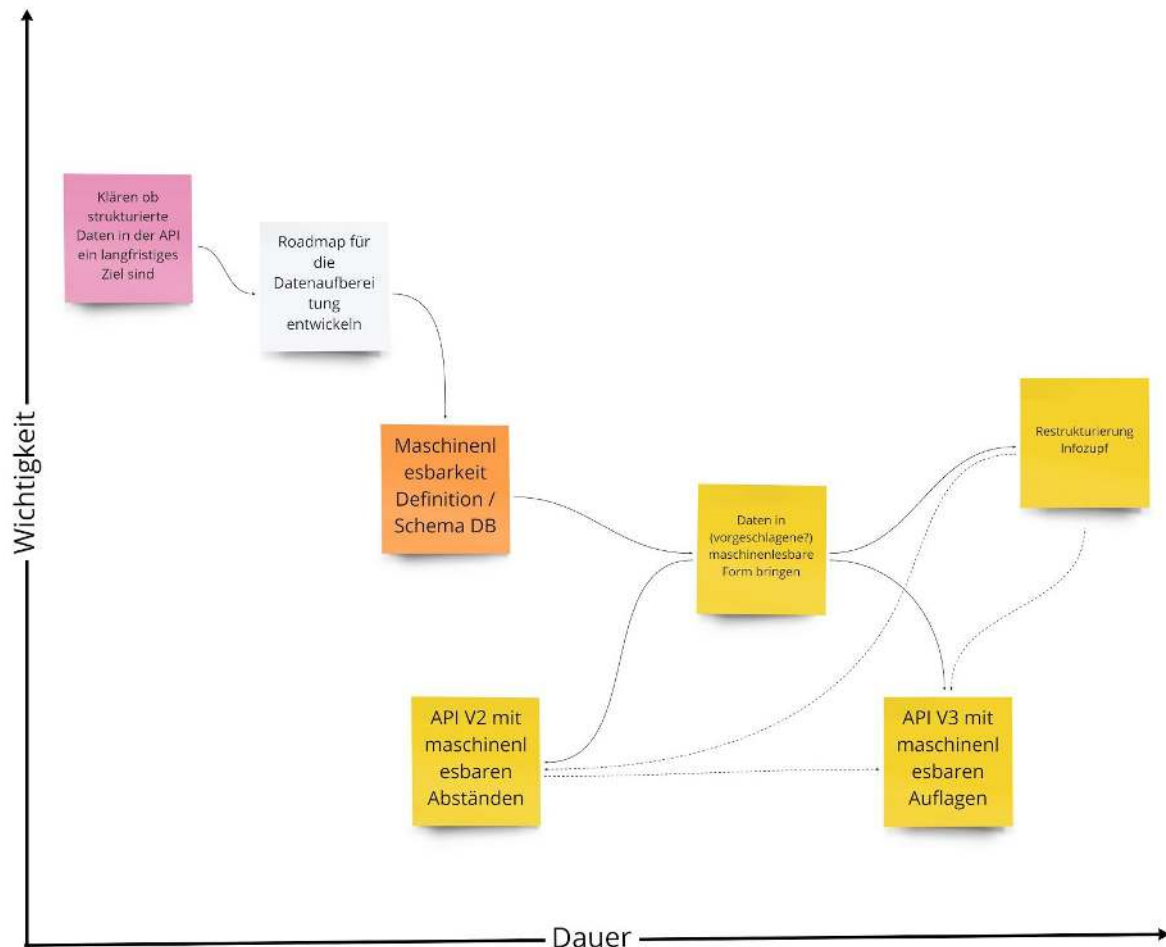
Um die fachliche Prüfung noch über die Möglichkeiten des Prüfskripts für Inkonsistenzen (siehe Kapitel 5.5.2) hinaus durch vorhandene Software zu unterstützen, müssten in die Infozupf Datenbank zusätzliche Felder eingeführt und die Infozupf Software um spezielle Validierungen erweitert werden. Potenzielle Inkonsistenzen können dann bereits beim Speichern in der Datenaufnahme geprüft werden und zu Warnungen führen. Das abschließende Festschreiben der Daten dürfte dann nur durch Eingabe einer Begründung erfolgen (die Begründung wird dann in dem neuen Datenbankfeld zu dem korrespondierenden Datensatz gespeichert). Spätestens bei der letzten Abnahme einer Zulassung vor Veröffentlichung des Bescheids, können diese Felder dann gesondert betrachtet und technisch freigeben werden. Diese technische Freigabe sollte ebenfalls in der Datenbank in einem korrespondierenden Feld hinterlegt werden. So ist für die Inkonsistenzprüfung transparent warum die Daten so vorliegen und dass diese auch technisch abgenommen wurden.

Vorteil: Dies schafft einen klaren Prozess wie mit ggf. fachlich notwendigen "Inkonsistenzen" bzw. Abweichungen zu den per Definition "konsistenten Zuständen" umzugehen ist. Die Häufigkeit rückwirkend Fehlersuche zu betreiben könnte dadurch reduziert werden.

7.2.2.2 Maschinenlesbarkeit der Daten verbessern

Von Seiten des BMEL und verschiedener Forschungseinrichtungen wurde an uns herangetragen, dass es eine gemeinsame Vision ist, die PSM Datenbank als eine

vollständig maschinen-interpretierbare Datenbank über eine Schnittstelle zur Verfügung zu stellen.



Für die Detailansicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:
https://miro.com/app/board/o9J_kkPogK4=?moveToWidget=3074457350613385888&cot=12

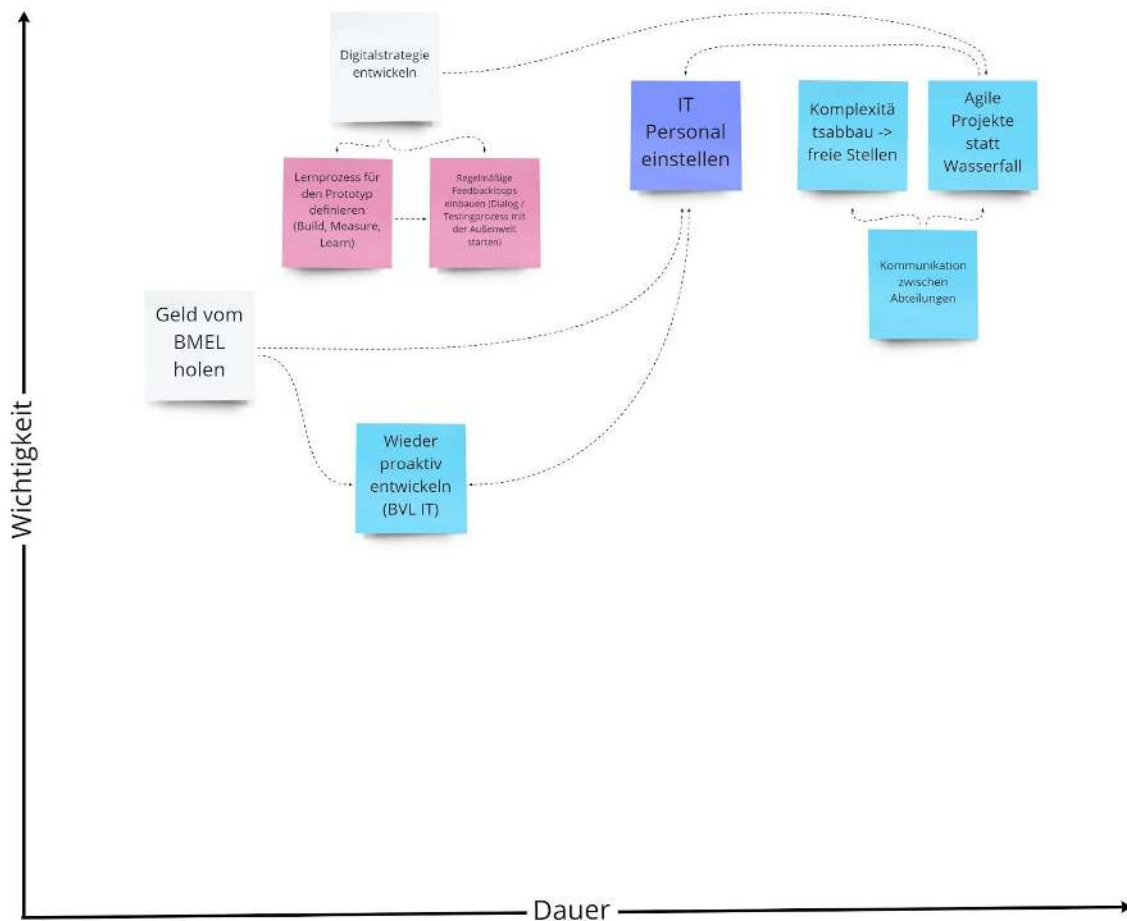
Erläuterungen:

- Als allerersten Schritt empfehlen wir, sich einmal in einer gemeinsamen Runde zusammzusetzen und mit **allen** Beteiligten, insbesondere dem BVL, auszutauschen, ob und warum die Datenbank maschinen-interpretierbar zur Verfügung gestellt werden soll. Dies kann helfen, um im BVL die gemeinsame Stoßrichtung zu stärken und die Ziele der Datenstrategie zu verständlichen. Auch sollte dabei klargestellt werden, um welche Form der Maschinenlesbarkeit es perspektivisch geht, sodass ein gemeinsames Verständnis geschaffen wird. Dabei kann es helfen, den Begriff kleiner und greifbarer zu machen.
- Darauf aufbauend können Strategien entwickelt werden.

- Der von uns gelieferte erste Ansatz einer maschinenlesbaren Strukturierung (Schema) kann weiterentwickelt und für sämtliche Auflagen nach und nach ausgefüllt werden. Dafür sollten zunächst die juristischen Formulierungen in den Auflagen in verständliche Texte heruntergebrochen werden.
- Darauf aufbauend, sollte zunächst die unterliegende Datenstruktur in der Datenbank (somit auch in Infozupf) angepasst werden, ehe man diese Daten dann per API bereitstellen kann.
- Auch hier schlagen wir ein **schrittweises** Vorgehen vor: Zunächst nur die Abstände zu extrahieren, zu **testen**, ob diese Änderung gut aufgenommen wird und dann erst andere Auflagen nachzuziehen.

7.2.2.3 Prozess anpassen

Um dabei zu helfen, die empfohlenen Schritte durchzuführen, ist es notwendig, dass interne BVL Prozesse angepasst und modernisiert werden. Anpassungen sollten auf einer gemeinsamen Digitalstrategie aufbauen, die klare Ziele und eine klare Richtung vorgibt. Eine Anpassung und Modernisierung von Prozessen ohne eine übergeordnete Strategie kann zur Fehlleitung einzelner Maßnahmen führen.



Für die Detailansicht bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:
https://miro.com/app/board/o9J_kkPogK4=/?moveToWidget=3074457350613385888&cot=12

Erläuterungen:

Die vielleicht wichtigste Erkenntnis, die wir über das BVL erhalten haben, ist, dass es stark an personellen IT Ressourcen mangelt. Um langfristige, digitale Visionen umsetzen zu können braucht es mehr Personal.

Um dieses Personal einstellen zu können, müsste unserer Ansicht nach an zwei Stellen angepackt werden:

- Geldakquise um wettbewerbsfähige Gehälter zu gewährleisten
- Die Umstrukturierung vom Wasserfallmodell in agile Entwicklung

Uns ist bewusst, dass dies große Anpassungen sind, die viel Zeit in Anspruch nehmen können und ggf. von höherer Stelle in die Wege geleitet werden müssen. Dennoch halten wir diese Anpassungen für essentiell, um langfristig als attraktiver Arbeitgeber für IT Personal zu werden.

Außerdem können Prozesse, wie der Freigabeprozess und der Inkonsistenzprüfungsprozess entkompliziert werden, wenn daran gearbeitet wird, dass unterschiedliche Abteilungen, beispielsweise die IT und die Fachabteilungen, Aufgaben im Hinblick auf den Gesamtprozess eher parallelisiert und gemeinsam abarbeiten als sequentiell. Dies ist in erster Linie kein Technikproblem.

Zusätzlich sollte ein Prozess etabliert werden mit dem man die aktuell implementierten Lösungen validieren und weiterentwickeln kann. Das kann so einfach sein, wie ein zentrales E-Mail Postfach für Feedback einzurichten, das regelmäßig überprüft und die Vorschläge abgewogen werden. Besser wäre noch die aktive Suche nach Kommunikationspartner:innen, die bereit sind ihre Erfahrungen zu teilen (Stichwort Nutzer:inneninterviews). Die Personen, die sich uns gegenüber bereits für so etwas bereit erklärt haben, haben wir in einer Liste zusammengeführt und liefern diese mit den Projektressourcen.

7.3 Langfristige Vision bilden

Wie in der Einleitung erwähnt, haben wir in der Zwischenpräsentation die Anwesenden gefragt, was ihre Vision für das BVL in fünf Jahren ist. Die Antworten lassen sich in folgender Vision zusammenfassen:

“Das BVL ist Vorreiter in Sachen Datenbereitstellung und Digitalisierung ihrer Prozesse”

Unserer Einschätzung nach, ist diese Vision durchaus zu verwirklichen und keine Utopie! Es muss nur gestartet werden. Kleine Schritte führen zum letztendlich großen Erfolg. Denn die gemeinsame Angst, die sich aus den zusammengetragenen Mitteilungen der Zwischenpräsentation ergeben hat, war folgende: “Trotz großer Bemühungen passiert gar nichts, aufgrund veralteter, langwieriger, starrer Prozesse”

Um dieser Schreckensnachricht entgegenzuwirken, haben wir oben stehende Handlungsempfehlungen formuliert, die Schritt für Schritt zu einem BVL als Vorreiter der Digitalisierung führen können.

Die nächste Vision auf dem Weg zum BVL als Digitalisierungsvorreiter kann also sein:

"Eine allgemein zugängliche und maschinenlesbare und -durchsuchbare PSM-Datenbank, die die anwendungsrelevanten Daten der Zulassung des jeweils zu nutzenden Pflanzenschutzmittels elektronisch zur Verfügung stellt."

8 Danksagung, Team & Kontakt

An dieser Stelle möchten wir uns gerne bei allen Projektbeteiligten bedanken. Danke an das Bundesministerium für Landwirtschaft und Ernährung, das Bundesamt für Verbraucherschutz und Lebensmittelsicherheit sowie das Julius Kühn Institut als Partnerorganisationen in diesem Projekt. Wir möchten uns bei Armin Weiser (BMEL), unserem Projektleiter, sowie Peter Ahlbrecht (BVL), Armin Wiese (BVL), Alexander Pfaff (BVL) und Burkhard Golla (JKI) bedanken, die uns als Digitallosten durch dieses Projekt begleitet haben.

Wir möchten Danke sagen für ihren Mut und Ihre Flexibilität neue Wege zu gehen und das uns entgegengebrachte Vertrauen im Laufe des Projekts. Ein großes Dankeschön gebührt auch der BVL-IT die uns mit sehr viel Geduld durch die Tiefen der IT-Infrastruktur des BVL gelotst und einen großen Beitrag zum Projekterfolg geleistet haben.

Außerdem möchten wir uns bei allen Personen bedanken, die uns so sympathisch und engagiert für Interviews jeglicher Art zur Verfügung standen. Sowohl in den Expert:inneninterviews als auch in den Nutzer:innentests des Prototypen durften wir sehr viel Leidenschaft und Motivation erleben unser Projekt voran zu bringen, das hat uns immer wieder motiviert, DANKE!

Lara Albrecht
Product Fellow

Kai Bitterschulte
Engineering Fellow

Frederike Ramin
Engineering Fellow

David Redlich
Engineering Fellow

Peter Ahlbrecht
Digitallotse
BVL

Burkhard Golla
Digitallotse
JKI

Alexander Pfaff
Digitallotse
BVL

Armin Wiese
Digitallotse
BVL

Armin Weiser
Projektleiter
BMEL

9. Anhang

9.1 Übersicht der von uns genutzten Tools

[Asana](#) - Tool, um die Arbeit im Team zu strukturieren. Es können Aufgabenpakete als "Tickets" oder "Karten" definiert werden, wobei bei jedem Ticket klar ist, wer dafür zuständig ist und in welchem Stadium es sich gerade befindet (zB: "noch nicht angefangen", "in Arbeit" oder "fertig"). Das schafft einen guten Überblick darüber, was wer gerade und als nächstes tut.

[G-Drive](#)/G-Suite - Digitaler Ort um sämtliche Ressourcen abzulegen. Von Fotos über Protokolle bis hin zu Dokumentation.

[Slack](#) - Chatprogramm in dem man private Nachrichten, aber auch Gruppennachrichten senden kann. Die Kommunikation organisiert sich in "Channels" in denen vorrangig über ein bestimmtes Thema gesprochen wird.

[Miro](#) - Digitales Whiteboard. Da wir viel mit Post-Its arbeiten, ist Miro eine gute Lösung, um unsere Ergebnisse zu digitalisieren. Das Ganze funktioniert auch kollaborativ, sodass das ganze Team gleichzeitig und gemeinsam an einem Whiteboard arbeiten kann.

[Github](#) - Versionsverwaltungstool für Code. Macht die kollaborative Arbeit am gleichen Produkt um einiges einfacher, durch leichtes Mergen von Ständen und Versionsverwaltung des Codes.

9.2 Warum haben wir uns für ORDS entschieden?

Unser Ziel war es, in enger Zusammenarbeit mit dem BVL zu klären welche technologischen Umsetzungen in Frage kommen. Seit Projektstart war Java als Programmiersprache im Gespräch, da in dem Bereich Wissen und Interesse in der BVL IT vorhanden ist.

Es wurde allerdings seit dem ersten Gespräch mit der BVL IT an uns herangetragen, dass im BVL sehr wenig personelle Ressourcen zur Verfügung stehen um eine von uns (teilweise) entwickelte Lösung weiterzuentwickeln und zu warten.

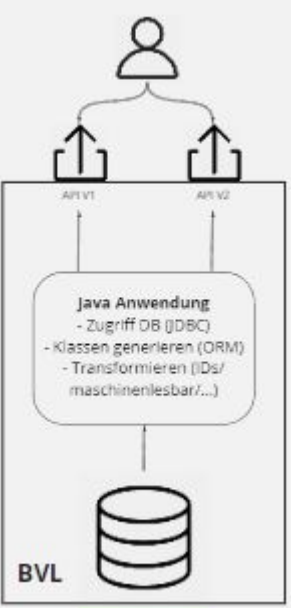
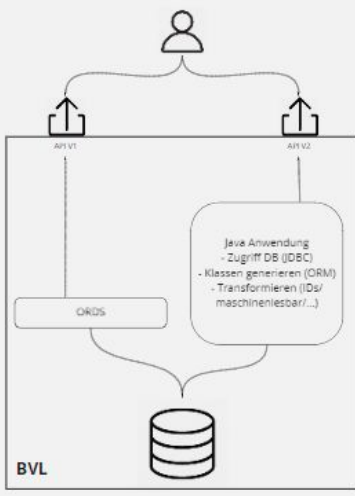
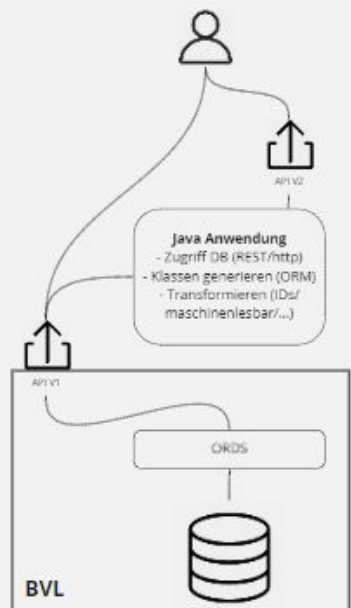
Daher waren die Ansprüche an unsere Lösung:

- So weit wie möglich auslieferungsfertig (um Weiterentwicklung zu vermeiden)
- Minimalinvasiv (um Wartungsaufwand gering zu halten)
- In enger Abstimmung mit der BVL IT entwickelt

Wir haben daher einen gemeinsamen Workshop mit der BVL IT am 10.09.2020 durchgeführt. Ziel des Workshops sollte sein, verschiedene Infrastrukturen gegeneinander abzuwägen. Dabei haben wir zwischen der Version 1 (V1) der API (reine Weitergabe von BVL PSM Tabellen per API) und einer Version 2 (V2) der API (Umstrukturierung/Aufbereitung der Daten) unterschieden.

In Vorbereitung auf den Workshop sind wir auf die Technologie Oracle REST Data Services (ORDS) gestoßen, die direkt von Oracle bereitgestellt wird und auf jeder Oracle Datenbank (die so im BVL vorhanden ist) mithilfe eines einzelnen SQL Skriptes aktiviert werden kann.

Von uns wurden drei verschiedene Infrastrukturkonzepte entwickelt, die im Workshop mit der IT vorgestellt wurden. Dafür wurden gemeinsam Pro- und Kontra-Argumente gesammelt.

Option A: Java für V1 und V2	Option B: ORDS für V1, Java für V2, unabhängig	Option C: ORDS für V1, Java für V2, abhängig
		
+ einheitlich	+ erster Schritt ist sehr	+ wenig BVL Aufwand

+ gut erweiterbar + kein Vendor-Lock-In	unaufwändig und schnell zu machen + zwei System können arbeitsteilig besser aufgeteilt werden + erster Schritt ist sehr unaufwändig und schnell zu machen	+ minimalinvasiv + Open Source Ansatz
- höherer Initialaufwand (Klassen schreiben) - höherer Aufwand für das BVL im Vergleich zu ORDS	- V2 braucht dann Arbeit und Pflege - zwei Systeme	- Rechtssicherheit von V2 nicht gewährleistet - V2 wird extern vielleicht nicht gewartet

Für alle Pro- und Kontra Argumente bitte in das Miro Board wechseln oder die beiliegende PDF öffnen:

https://miro.com/app/board/o9J_kkPogK4=?moveToWidget=3074457350557140298&cot=12

Anhand dieser Argumente haben wir uns letztendlich aufgrund des minimalen Aufwandes durch das BVL dafür entschieden, Option C zu wählen und zunächst nur an Version 1 der API mit ORDS zu arbeiten.

9.3 Was kann ORDS und wo stoßen wir an Grenzen?

ORDS ist darauf ausgelegt, die Inhalte einer Oracle Datenbank mit möglichst wenig Aufwand als API zur Verfügung zu stellen. Dabei kann mithilfe eines SQL Skripts, eine REST Schnittstelle für das ganze Datenbankschema erstellt werden.

Mit besagten Skript lassen sich einzelne Endpunkte inkl. konkreter Abfrageparameter definieren. Die Daten, die an einem Endpunkt abrufbar sind, können durch SQL Statements in beliebiger Komplexität definiert werden. Auch die Verwendung von PL/SQL ist möglich. Außerdem bietet ORDS von Haus aus die Möglichkeit mithilfe von Json Queries ([ORDS Dokumentation](#)) Filter auf die Endpunkte anzuwenden. ORDS liefert Jetty für einen Standalone Betrieb aus und es wird kein weiterer Server benötigt. Mindestens eine Schicht zwischen Datenbank und Webschnittstelle fällt im Vergleich zu traditionellen Webservices weg. Änderungen können einfach zur Laufzeit eingespielt werden.

Bei der Betrachtung des Tools, haben wir festgestellt, dass es sich für einfache Anwendungen sehr gut eignen kann. Allerdings wird es bei komplexen

Anwendungsfällen schwierig. Irgendwann wird ein Punkt erreicht an dem es ORDS an Flexibilität mangelt:

- Businesslogik in SQL bzw PL/SQL umzusetzen ist nicht unmöglich aber sicherlich wenig komfortabel.
- Weniger Kontrolle darüber was zwischen Abfrage und Rückgabe der Daten passiert.
- PL/SQL ist sicherlich weniger verbreitet ist als Java. Somit sind Entwickler:innen rar.
- Wartbarkeit kann bei komplexen und aufwändigen Implementierungen schnell zum Problem werden.
- Die Unterstützung durch Tools bspw. zum Debuggen von Code ist im Umfeld von Programmiersprachen wie Java deutlich vorteilhafter.
- Eingeschränkte Konfiguration - man kann ORDS beispielsweise nicht so konfigurieren, dass es XML zurückliefert. Die Rückgabe ist immer im Json Format.
- Die resultierende Schnittstelle ist abhängig von einer proprietären Technologie, auf dessen Weiterentwicklung kein Einfluss besteht.
- Fehlerbehandlung in PL/SQL umzusetzen ist nur bedingt intuitiv.
- ORDS generiert eine grundlegende OpenAPI-Swagger-Dokumentation, eine fein granulare Dokumentation der Schnittstelle, der Rückgabe, der Parameter etc. ist jedoch bis dato nicht möglich.
- Die Hauseigenen Query Filter von ORDS können im OpenAPI-Swagger nicht dokumentiert werden, bzw. tauchen in der von ORDS generierten OpenAPI-Swagger Dokumentation nicht auf, weshalb wir das in die allgemeine Dokumentation am Anfang der Seite einfügen mussten.
- Die Dokumentation für OpenAPI-Swagger, kann nur über sehr allgemein gehaltene Kommentarfelder vorgenommen werden.
- Die Formatierung der Beschreibung der Endpunkte etc. ist nicht intuitiv. Markdown wird nur teilweise unterstützt (z.B. Links, italic, bold), die Syntax weicht teilweise vom Standard ab und ist nicht dokumentiert (z.B. Code Snippets mit tab Einrückung statt ```) und einiges wird nicht unterstützt (z.B. Überschriften).