

Spezifikationsdokument

Erweiterung der Funktionen von GeoNode als

Forschungsdatenmanagement Plattform –

Software-Entwicklungsprojekt 2022

Florian Hoedt & Celina Thomas

Braunschweig, 03.05.2022



Florian Hoedt
Thünen-Institut, Zentrum für Informationsmanagement

Johann Heinrich von Thünen-Institut
Bundesforschungsinstitut für Ländliche Räume, Wald und Fischerei
Bundesallee 50
38116 Braunschweig

Tel.: +49 531 596-1428
E-Mail: florian.hoedt@thuenen.de

Inhaltsverzeichnis

Abbildungen	I
Tabellen	II
1 Ausgangssituation	1
1.1 Der Auftraggeber	1
1.2 Status Quo	1
1.3 Beteiligte Personen	2
1.4 Zielsetzung	2
1.5 Nutzergruppen	3
2 Spezifikation	3
2.1 Leistungsanforderungen	3
2.1.1 Synchronisierung der GeoNode-Instanzen	3
2.1.2 Implementierung eines UUID-Resolvers	3
2.1.3 Einfache benutzerdefinierte Applikation	3
2.1.4 EPIC Metadatenschema erweitern	6
2.1.5 ORCID als Identitätsprovider	10
2.1.6 EPIC Erweiterung der Datenexplorationswerkzeuge	10
2.1.7 EPIC Mapstore2 UI/UX Verbesserungen	14
2.1.8 Implementierung nicht-räumlicher Datensätze als neuer Ressource-Base-Subtyp	20
2.2 Dokumentation	20
2.3 Implementation	20
2.3.1 Layout und Design	20
2.3.2 Entwicklung	20
2.3.3 Testdaten	21
2.3.4 Deployment	21
Literaturverzeichnis	21

Abbildungen

Abbildung 1: Komponenten der Forschungsdateninfrastruktur.	1
Abbildung 2: Homepage.	4
Abbildung 3: Ausgewählter Atlas mit benutzerdefinierter Applikationsliste als Proof-of-Concept.	5
Abbildung 4: Homepage im Master Branch mit zusätzlichen benutzerdefinierten Applikationen als durchsuchbare Ressourcen.	5
Abbildung 5: Beispiel für die Detailansicht einer benutzerdefinierten Applikation.	6
Abbildung 6: Beispiel für die fehlende Multiplizität der Datumsangabe.	7
Abbildung 7: Beispiel für die Anzeige der Elemente als Teil der Ressourcendetailansicht.	9
Abbildung 8: Übersicht der Downloadoptionen.	10

Abbildung 9: Aktuelle Liniendiagramm-Implementierung zur Darstellung von Zeit und Durchschnittswerten für einen Datensatz. Es kann nur eine Farbe für das Diagramm ausgewählt werden, was zu einem einzigen Liniendiagramm führt.	11
Abbildung 10: Beispiel für ein Apache Superset Liniendiagramm mit Farbe als zusätzliche Gruppierungsebene, wodurch mehrere Linien erzeugt werden.	12
Abbildung 11: Beispiel für ein gestapeltes Balkendiagramm.	12
Abbildung 12: Beispiel für ein Sunburst-Diagramm. Die dargestellten Variablen sind der Bodenkohlenstoffbestand pro Ökoregion, die Landnutzungsart und die Bodenart (vom inneren zum äußeren Ring).	13
Abbildung 13: Dashboard ohne angewandte Filter.	13
Abbildung 14: Dashboard mit angewandtem Ökoregionsfilter. Die Widgets zeigen einen aktiven Filter und haben ihre Diagramme entsprechend angepasst.	14
Abbildung 15: Beispiel für die gegenseitige Ausschließung einzelner Ebenen auf Grund von Überschneidungen.	15
Abbildung 16: Aktuelle Nutzererfahrung der Legendenansicht bei Karten mit vielen Ebenen.	16
Abbildung 17: Beispiel für eine Legende mit den Elementen Zusammenfassung und Titel.	16
Abbildung 18: Beispiel für Datensätze mit mehreren Attributen zur Gestaltung von Kartenmerkmalen.	17
Abbildung 19: Beispiel der benötigten Klicks zur Änderung eines Ebenenstils.	18
Abbildung 20: Beispiel eines Dropdowns zum Umschalten des Ebenenstils.	18
Abbildung 21: Beispiel für die aktuelle Darstellung längerer Titel. Die Titel werden aktuell nicht vollständig angezeigt.	19
Abbildung 22: Beispiel für einen Zeilenumbruch in der Darstellung des Titels.	19

Tabellen

Tabelle 1: Übersicht der spezifischen Rollen.	7
---	---

1 Ausgangssituation

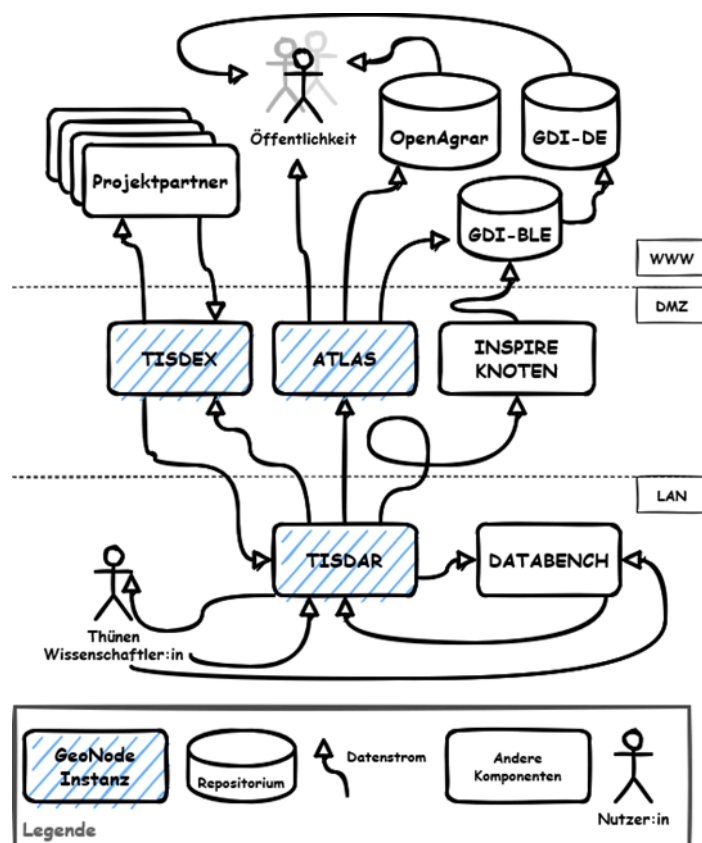
1.1 Der Auftraggeber

Der Auftraggeber ist eine Bundesbehörde im Geschäftsbereich des Bundesministeriums für Ernährung und Landwirtschaft (BMEL). Diese betreibt interdisziplinäre Forschung mit dem Ziel der nachhaltigen Entwicklung der Land-, Forst- und Holzwirtschaft sowie der Fischerei. Sie gliedert sich in einzelne Fachinstitute, die auf zehn Standorte in Norddeutschland verteilt sind. Der Hauptsitz ist in Braunschweig.

1.2 Status Quo

Um die steigende Menge an (Geo-)Daten zu verwalten, betreibt der Auftraggeber eine Forschungsdateninfrastruktur (FDI). Ein zentraler Bestandteil dieser FDI sind Geodatenportal-Anwendungen, die auf dem OSGeo-Projekt GeoNode basieren. Die FDI besteht aus mehreren miteinander verknüpften Komponenten, wie in der folgenden Abbildung dargestellt:

Abbildung 1: Komponenten der Forschungsdateninfrastruktur.



Quelle: Eigene Darstellung.

Zur Sicherung der Dienste und zur Bereitstellung modifizierter UI/UX für bestimmte Anwendungsfälle werden drei getrennte GeoNode-Instanzen eingesetzt (siehe Abbildung 1, blau markierte Knoten). Die TISDAR-Instanz wird im internen Netz (LAN) eingesetzt und ist die Hauptinstanz. TISDEX wird verwendet, um Datenaustauschfunktionen für Forschungsprojekte mit externen Partnern bereitzustellen. Es handelt sich um einen gesicherten Knoten mit Daten, die nicht öffentlich zugänglich sind. Die dritte Instanz ist der ATLAS, der offene Daten sowie Applikationen für die Datenexploration bereitstellt und öffentlich zugänglich ist.

Diese Portale veröffentlichen die räumlichen Informationen gemäß der vom Open Geospatial Consortium (OGC) definierten Schnittstellen, wie z.B.:

- Web-Map-Services (WMS),
- Web-Feature-Services (WFS),
- Web-Coverage-Services (WCS).

Um den Zugriff auf die oben genannten OGC-Dienste zu autorisieren, wird GeoFence, ein GeoServer-Plugin, verwendet. Zusätzlich kann OAuth zur Authentifizierung verwendet werden.

Die vom Auftraggeber verwendete Portalsoftware nutzt das Django-Webframework in der Version 2.2.24 und basiert auf dem Open-Source-Projekt GeoNode in der Version 3.3.2. Das Portal läuft auf einer Ubuntu 20.04 LTS VM unter Python 3.8 und verwendet als Backend eine PostgreSQL Datenbank in der Version 13 mit der räumlichen Erweiterung PostGIS in Version 3.1. Die OGC-Dienste werden über einen GeoServer in der Version 2.18.2 veröffentlicht. In diesem Portal wird die Mapping-Anwendung [MapStore2](#) als Web-GIS-Client und Kartenansicht verwendet. Die über GeoNode hochgeladenen Ebenen werden als Karten zusammengefasst und können mit sogenannten Widgets, wie Diagrammen und Tabellen, dargestellt werden. Zusätzlich können die Karten mit anderen Medien zu Applikationen, wie Dashboards oder Präsentationen, zusammengeführt werden (siehe [GeoStories](#)). Die Kommunikation zwischen MapStore2 und GeoNode erfolgt über einen Adapter, der die

Kartenkonfiguration in GeoNode an MapStore2 überträgt und Änderungen in MapStore2 persistent in das GeoNode Backend schreibt.

In naher Zukunft werden diese Portale auf die derzeit in der Entwicklung befindliche Version 4.0 von GeoNode aktualisiert, die Python 3.10.x, Django 3.2.x und GeoServer 2.20.x verwenden wird.

Neben diesen GeoNode-Instanzen wird ein dedizierter INSPIRE-Knoten (siehe Abbildung 1) verwendet, um INSPIRE konforme, harmonisierte Datensätze zu veröffentlichen. Dieser Knoten besteht aus einer hale-connect, einem GeoServer v2.19.x und einer GeoNetwork v3.12.1 Instanz.

Da TISDAR als Datenspeicher verwendet wird, bietet es keine Analysemöglichkeiten. Außerdem sollen die Datensätze in TISDAR nicht verändert werden, sondern in einem schreibgeschützten Status verbleiben, sobald sie fertiggestellt und durch Metadaten beschrieben sind. Daher wird ein Databench-Knoten bereitgestellt. Dieser besteht aus einem performanten PostgreSQL v13 DBMS und wird für die Abfrage und Analyse von Duplikaten von TISDAR-Datensätzen verwendet. Dieser Knoten befindet sich noch im Aufbau.

1.3 Beteiligte Personen

Die am Projekt beteiligten Personen können in zwei Gruppen unterteilt werden:

- Das **Projektmanagement** ist für die Definition und Weiterentwicklung der Projektvision verantwortlich und beantwortet Fragen zu den Projektzielen. Die Projektmanagementgruppe wird ausschließlich vom Auftraggeber besetzt.
- Die **Implementierungsgruppe** ist für Entwicklungsfragen zuständig. Sie legt die technischen Lösungen für die vom Projektmanagement definierten Anforderungen fest. Die Implementierungsgruppe setzt sich aus Mitgliedern des Auftragnehmers und des Auftraggebers zusammen.

1.4 Zielsetzung

Das Ziel ist es, die Fähigkeiten der beschriebenen Infrastruktur als Forschungsdatenmanagement-Infrastruktur zu verbessern. Um dieses Ziel zu erreichen, haben wir 8 Leistungsmerkmale festgelegt (siehe 2.1 Leistungsanforderungen), welche in die GeoNode basierten Knoten (TISDAR, TISDEX, ATLAS) integriert werden sollen.

1.5 Nutzergruppen

Die folgenden Nutzergruppen der gewünschten Anwendung können unterschieden werden:

Administratoren: Konfigurieren und verwalten die GeoNode-Instanz. Sie führen die Zugriffs- und Rechteverwaltung durch, konfigurieren Harvester und verteilen und aktualisieren Instanzen.

Ersteller: Sie erstellen Datensätze und laden sie in GeoNode hoch. Sie erstellen auch Dashboards als spezielle Visualisierungen für die Tabellendatensätze.

Betrachter: Suchen und betrachten Datensätze über die GeoNode-Portale. Sie interagieren mit den genannten Datensätzen, indem sie deren Metadaten abrufen und sie als Vorschau, innerhalb von Dashboards oder als Teil eines GIS-Clients wie QGIS betrachten.

2 Spezifikation

2.1 Leistungsanforderungen

2.1.1 Synchronisierung der GeoNode-Instanzen

Problem: Derzeit werden die TISDAR-Ressourcen manuell zwischen den Knoten veröffentlicht. Es gibt keine automatische Synchronisation. Die Daten und Metadaten einer TISDAR-Ressource werden verwendet, um eine Kopie in TISDEX oder im ATLAS zu erstellen. In GeoNode Version 4 ist es möglich Ressourcen von einem anderen Knoten zu kopieren, aber nur als Pull- und nicht als Push-Prozess. Daher ist ein Push der TISDAR-Datensätze nach TISDEX oder ATLAS nicht möglich, da die DMZ-Knoten nicht in der Lage sind (und auch nicht sein sollen), auf TISDAR in unserem firmeneigenen LAN zuzugreifen.

Zielstellung: Als Administrator möchte ich eine Push-basierte Synchronisation konfigurieren, die es ermöglicht, Ressourcen, die mit einem definierten Tag markiert sind, zu TISDEX und/oder ATLAS zu replizieren.

2.1.2 Implementierung eines UUID-Resolvers

Problem: Bei der Arbeit mit mehreren GeoNode-Instanzen ist der derzeitige Mechanismus zur Auflösung eines Datensatzes über `/type/datasetname` umständlich und erfordert Kenntnisse über den Datensatztyp. Außerdem kann der Name einer Ressource geändert werden, was zu einer URL führt, die auf den alten Titel als „datasetname“ verweist.

Zielstellung: Als Betrachter möchte ich die Detailansicht des Datensatzes mit Hilfe seiner UUID öffnen. Zum Beispiel dieser Datensatz:

https://tisdex.thuenen.de/layers/geonode_data:geonode:simplified_segmentation_ni

soll über diese Anfrage lösbar sein:

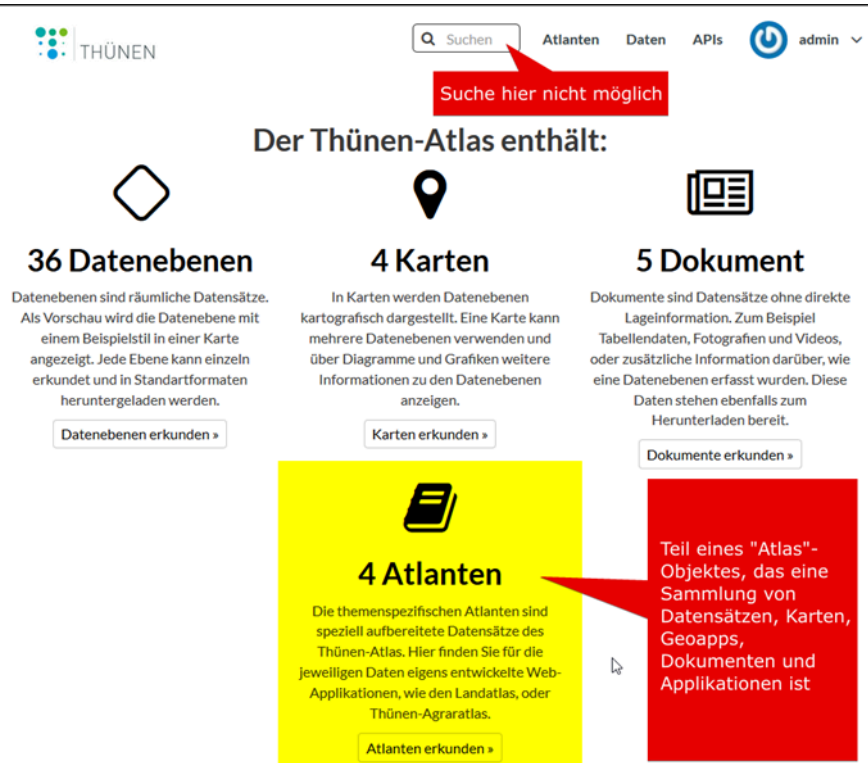
<https://tisdex.thuenen.de/resolver/a2abe8ce-83d1-4c42-b184-958a1b85b652>

2.1.3 Einfache benutzerdefinierte Applikation

Problem: Das aktuelle Portal verwendet ein generisches Applikationsmodell mit modularen (pluggable) Entitäten (siehe [GitHub](#)). Diese Funktionalität wird verwendet, um Dashboards und GeoStories innerhalb des Portals zu speichern und zu verwalten. Inzwischen verwendet der Auftraggeber seine eigenen benutzerdefinierten Applikationen, die keinen gemeinsamen Code mit dem Portal haben und nicht wie die aktuellen Portalapplikationen konfigurierbar sind. Stattdessen bieten diese benutzerdefinierten Applikationen spezielle Benutzeroberflächen und Funktionen für wissenschaftliche Datensätze. Derzeit gibt es keine Möglichkeit diese Applikationen innerhalb des Portals zu registrieren.

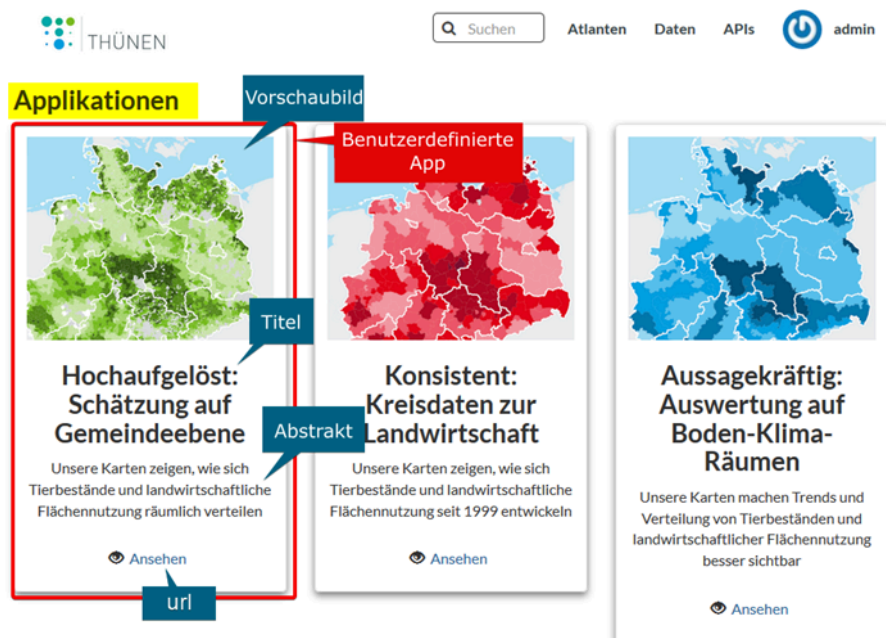
Daher können wir unsere benutzerdefinierten Applikationen innerhalb des Portals nicht auffindbar machen. Als Proof-of-Concept entwickelte der Auftraggeber zusätzliche Django-Modelle, um eine Sammlung von Datensätzen, Dokumenten, Geoapps, Karten und den genannten benutzerdefinierten Applikationen zu speichern. Diese ist im [ATLAS](#) veröffentlicht. Als Betrachter können Sie zu diesen Applikationen gelangen, indem Sie den [die Anwendungen eines der Atlanten betrachten](#), aber Sie können nicht nach diesen Applikationen suchen. Der Grund dafür ist, dass die benutzerdefinierten Applikationen nicht Teil des ResourceBase-Modells sind. Dies ist jedoch die Grundlage der Suche innerhalb des Portals. Sehen Sie sich dieses Beispiel der Benutzeroberfläche für diese Applikationen an:

Abbildung 2: Homepage.



Quelle: Eigene Darstellung.

Abbildung 3: Ausgewählter Atlas mit benutzerdefinierter Applikationsliste als Proof-of-Concept.

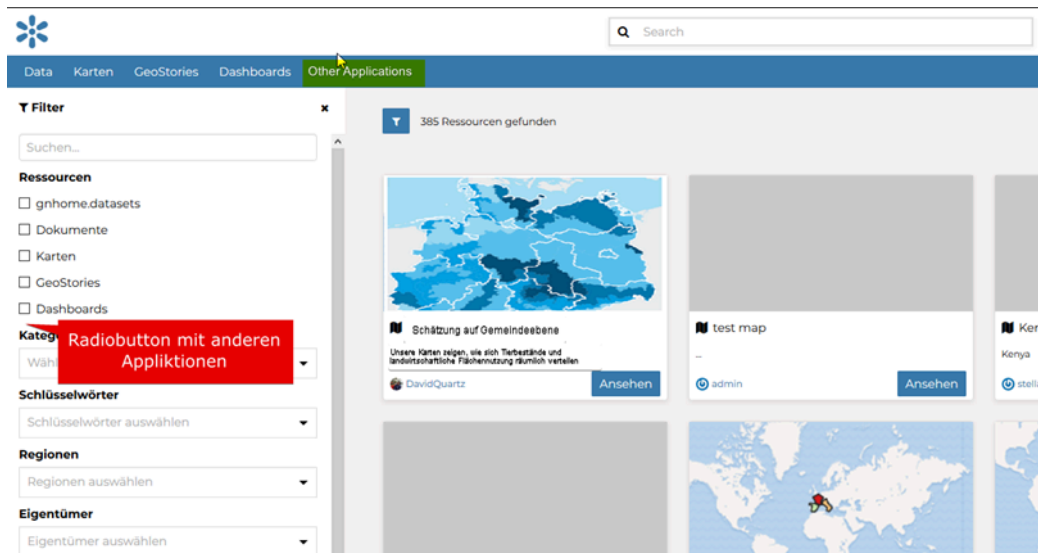


Quelle: Eigene Darstellung.

Zielstellung: Wir wollen unser derzeitiges Proof-of-Concept aufgeben und die benutzerdefinierten Applikationen in das GeoNode ResourceBase-Modell integrieren. Das derzeitige generische GeoApp-Modell soll um ein sehr einfaches benutzerdefiniertes Applikationsmodell erweitert werden. Dieses Modell soll die

[RessourceBase-Metadatenfelder](#) und eine URL zur Applikation selbst speichern. Betrachter müssen in der Lage sein, nach den Applikationen zu suchen. Schauen Sie sich hierzu dieses Beispiel an:

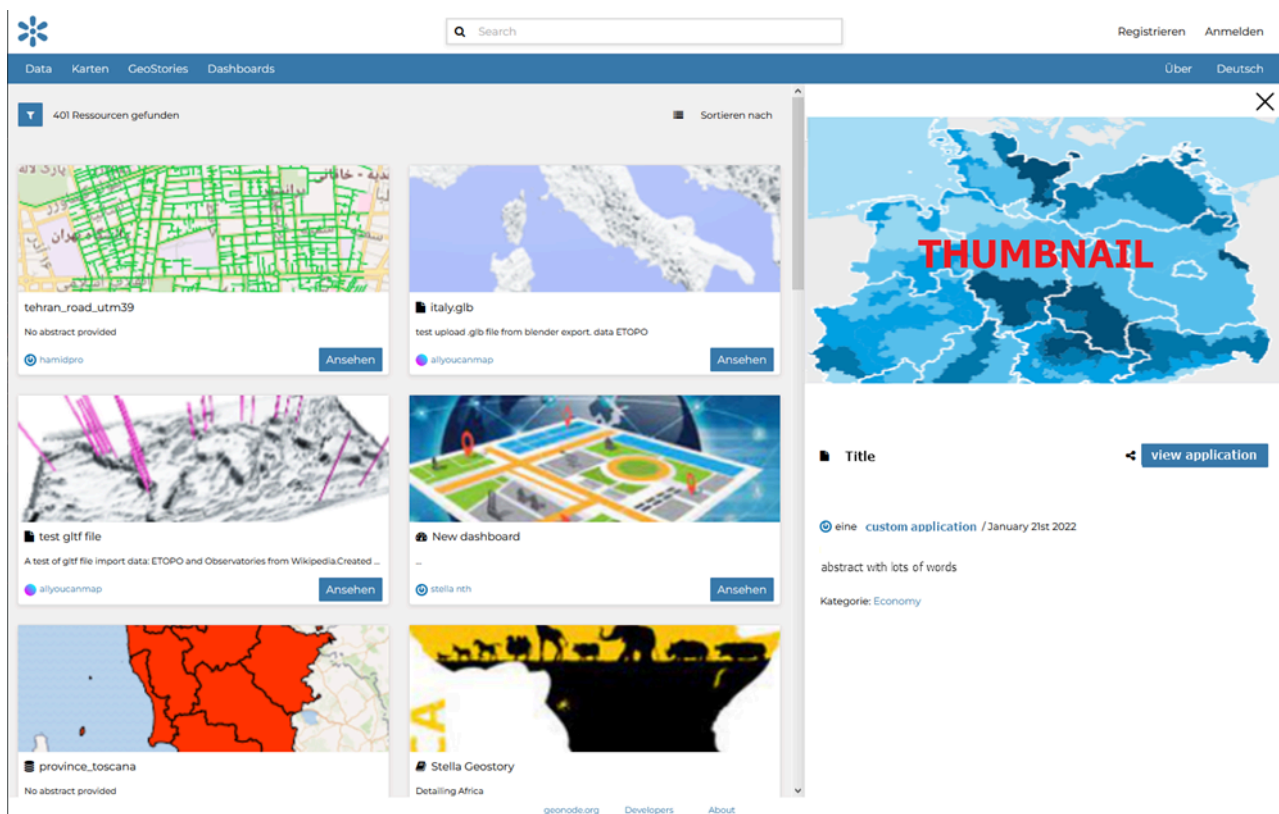
Abbildung 4: Homepage im Master Branch mit zusätzlichen benutzerdefinierten Applikationen als durchsuchbare Ressourcen.



Quelle: Eigene Darstellung.

Wenn ein benutzerdefiniertes Applikationsobjekt angeklickt wird, soll die Detailansicht das Vorschaubild, den Titel und die Zusammenfassung wie folgt anzeigen:

Abbildung 5: Beispiel für die Detailansicht einer benutzerdefinierten Applikation.



Quelle: Eigene Darstellung.

Durch Klicken auf „Anwendung anzeigen“ soll die URL geöffnet werden.

Die Erstellung einer neuen benutzerdefinierten Applikation funktioniert wie die Erstellung eines neuen Dokuments und die Festlegung einer URL anstelle des Hochladens des Dokuments selbst. Der Code und die Konfiguration der benutzerdefinierten Applikation sind nicht Teil der GeoNode-Benutzeroberfläche, und die Entwickler der benutzerdefinierten Applikation sind dafür verantwortlich, eine laufende Applikation hinter der URL bereitzustellen.

2.1.4 EPIC Metadatenschema erweitern

Problem: Das derzeit implementierte Metadatenschema und die Exportmöglichkeiten bieten nicht genügend Komplexität für Forschungsdaten. Die Bereitstellung dieser Funktionalität ist komplex und wird daher als ein EPIC beschrieben. Bei der Bearbeitung der geforderten API Erweiterungen soll [die bereits stattgefunden Diskussions](#) zur Modernisierung der pyCSW Komponente innerhalb des Projektes aufgegriffen werden. Dieses EPIC beinhaltet folgende Teilfunktionen:

2.1.4.1 Multiplizität für Datumsangaben

Problem: Das Metadatenschema erlaubt nur ein Datum mit einem bestimmten Datumstypen. Mehrfachangaben sind nicht möglich. Dies verhindert, dass ein Ersteller Datumsinformationen für alle drei Datumstypen (Erstellung, Veröffentlichung, Überarbeitung) eingeben kann.

Abbildung 6: Beispiel für die fehlende Multiplizität der Datumsangabe.

The screenshot displays the GeoNode metadata editor. At the top, there are tabs for 'Bearbeiten', 'Voransicht', and 'Einstellungen'. Below these, a progress bar indicates the status of different sections: 'Verpflichtend' (Required) for the first two sections and 'Optional' for the last two. The four sections are: 1. Grundlegende Metadaten (Basic Metadata), 2. Ort und Lizenzen (Location and Licenses), 3. Optionale Metadaten (Optional Metadata), and 4. Datensatz Attribute (Dataset Attributes). The 'Optionale Metadaten' section is currently active, showing fields for 'Datum Typ' (set to 'Publication'), 'Datum' (set to '2022-01-17 08:2'), 'Kategorie' (with a red error message stating 'Feld ist als verpflichtend im Metadatenschema gekennzeichnet'), 'Gruppe', and 'Freitext Schlüsselwörter'. The 'Grundlegende Metadaten' section shows a thumbnail of a green polygon and a 'Zusammenfassung' (Summary) section with a text editor.

Quelle: Eigene Darstellung.

Zielstellung: Ein Ersteller sollte die Möglichkeit haben, das Datumsfeld mehrfach zu verwenden und somit verschiedene Datumstypen zu nutzen. Als Betrachter möchte ich in der Lage sein, diese neuen Informationen über die AOI v2 und CSW-Endpunkte abzurufen.

2.1.4.2 Multiplizität für verantwortliche Personen und Differenzierung in spezifische Rollen

Problem: Der aktuelle Metadaten-Assistent erlaubt nur einen Eintrag für das Feld „verantwortliche Person“ und die Person muss ein registrierter Nutzer von GeoNode sein. Darüber hinaus bietet das aktuelle Schema keine weiteren Rollen von Personen an, die mit dem Datensatz in Verbindung stehen. Einige der in TISDAR gespeicherten Daten werden von anderen Forschungseinrichtungen oder Bundesbehörden (z.B. BKG) bereitgestellt. Es ist uns nicht möglich, diese Herkunft in TISDAR anzugeben und Ansprechpartner außerhalb unserer Plattform zu definieren.

Zielstellung: Ein Ersteller sollte die Möglichkeit haben, unterschiedliche „verantwortliche Personen“, sowie Personen mit weiteren Rollen einzutragen. Darüber hinaus soll der Ersteller die Möglichkeit haben, zugehörige Personen aus einer Liste hinzuzufügen, sowie über einen Freitext zugehörige Personen zu erfassen, die nicht am Thünen-Institut beschäftigt sind. Die folgenden Personentypen sollen ermöglicht werden:

Tabelle 1: Übersicht der spezifischen Rollen.

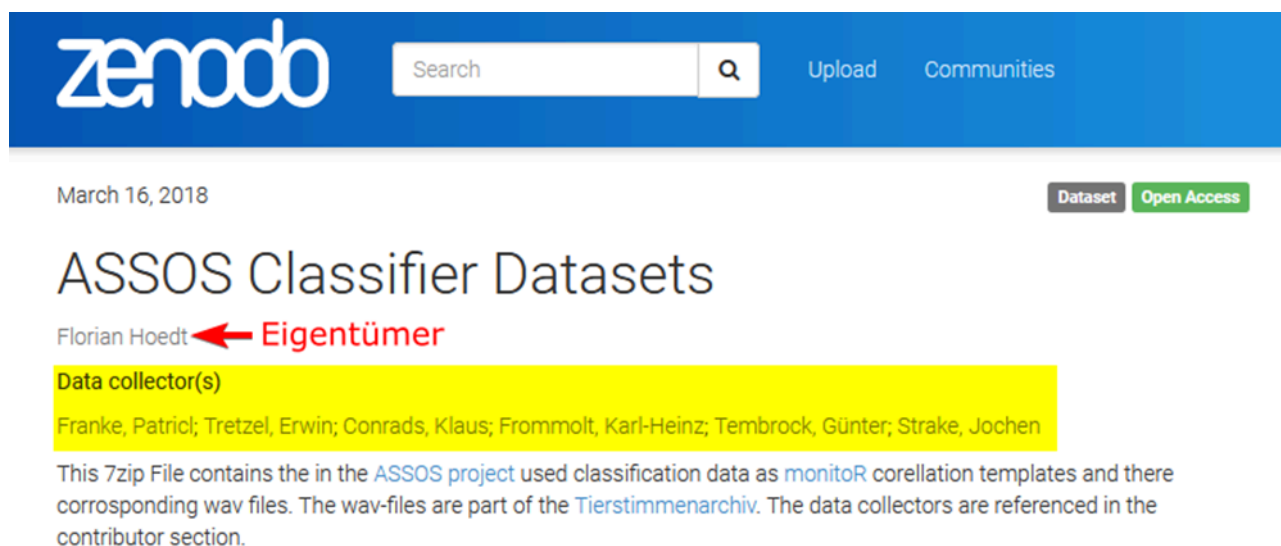
Rollen	Beschreibung	Zusätzliche Informationen	Quelle
Autor / Ersteller	Die maßgeblich an der Erstellung der Daten beteiligten Forschenden bzw. die Autoren der Veröffentlichung, in der Reihenfolge ihrer Priorität		BonaRes Metadatenchema DataCite Metadatascheme v4.4
Projektleitung	Person, die offiziell als Leiter*in eines Projektteams oder eines Teilprojektteams benannt ist, das an den für die Entwicklung der Ressource erforderlichen Arbeiten beteiligt ist		BonaRes Metadatenchema DataCite Metadatascheme v4.4
Datenerfasser*in	Person / Einrichtung, die für das Finden, Sammeln / Erheben von Daten unter der Leitung des Autors / der Autoren oder des Principal Investigators (PI) verantwortlich ist	Kann auch verwendet werden, um Umfrageleiter*innen, Interviewer*innen, Ereignis- oder Zustandsbeobachter*innen, Personen, die für die Überwachung wichtiger Instrumentdaten verantwortlich sind, zu berücksichtigen	BonaRes Metadatenchema DataCite Metadatascheme v4.4
Datenmanager*in	Person (oder Organisation mit einem Team von Datenmanagern, z.B. ein Datenzentrum), die für die Pflege der fertigen Ressource verantwortlich ist	Die Arbeit dieser Person oder Organisation stellt sicher, dass die Ressource regelmäßig „aktualisiert“ wird in Bezug auf Software-/Hardwareunterstützung, verfügbar gehalten oder vor unbefugtem Zugriff geschützt wird, in Übereinstimmung mit den Industriestandard aufbewahrt wird und in Übereinstimmung mit den für sie geltenden Anforderungen an das Aufbewahrungsmanagement behandelt wird.	DataCite Metadatascheme v4.4
Eigentümer	Eigentümer	-	BonaRes Metadatenchema
Andere	Jede Person oder Institution, die einen bedeutenden Beitrag zur Entwicklung und / oder Pflege der Ressource leistet, deren Beitrag aber nicht in ein anderes kontrolliertes Vokabular für den Typ „passt“	Dabei kann es sich um einen Fotografen, Künstler oder Schriftsteller handeln, dessen Beitrag dazu führte, die Ressource bekannt zu machen (im Gegensatz zu ihrer Erstellung), um eine(n) Reviewer*in der Ressource, um jemanden, der dem Autor administrative Dienste leistet (z.B. Hinterlegung von Aktualisierungen in einem Online-Repositorium,	BonaRes Metadatenchema DataCite Metadatascheme v4.4

Rollen	Beschreibung	Zusätzliche Informationen	Quelle
		Analyse der Nutzung usw.), oder eine von vielen anderen Rollen.	
Herausgeber	Der Name der Einrichtung, die die Ressource besitzt, archiviert, veröffentlicht, druckt, verteilt, freigibt, herausgibt oder produziert	Diese Eigenschaft wird für die Formulierung der Zitation verwendet, daher sollte die Bedeutung der Rolle berücksichtigt werden.	BonaRes Metadatenschema DataCite Metadataschema v4.4

Quelle: BonaRes Metadatenschema; DataCite Metadatenschema v4.4.

Als Anzeigeelement soll diese Information als Teil der Ressourcendetailansicht angezeigt werden. Ein Beispiel finden Sie hier:

Abbildung 7: Beispiel für die Anzeige der Elemente als Teil der Ressourcendetailansicht.



Quelle: Eigene Darstellung.

Außerdem möchte ich als Betrachter in der Lage sein, diese neuen Informationen über die API v2 und CSW-Endpunkte abzurufen.

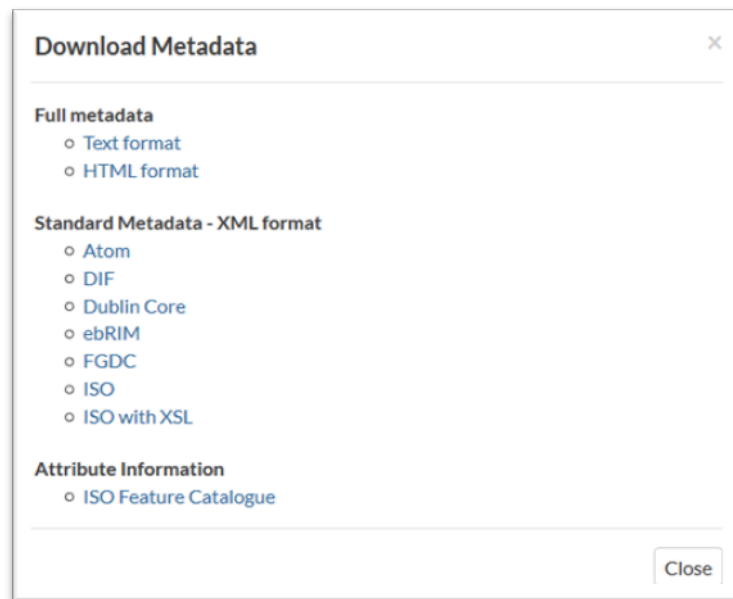
2.1.4.3 Feld für Herkunftsnachweis

Problem: Der aktuelle Metadaten-Assistent bietet kein Textfeld, in dem der Nutzer Informationen über die Datenherkunft eintragen kann.

Zielstellung: Als Ersteller möchte ich Herkunftsinformationen über ein Freitextfeld „Herkunftsangaben“ im Assistenten bereitstellen können. Als Anzeigeelement soll diese Information als Teil der Ressourcendetailansicht angezeigt werden. Außerdem möchte ich als Betrachter in der Lage sein, diese neuen Informationen über die API v2 und CSW-Endpunkte abzurufen.

2.1.4.4 DataCite Metadatenschema als Downloadoption hinzufügen

Problem: Die aktuellen Downloadoptionen von Metadatenstandardformaten umfassen nicht das DataCite Metadatenformat.

Abbildung 8: Übersicht der Downloadoptionen.

Quelle: Eigene Darstellung.

Zielstellung: Als Betrachter möchte ich Metadaten als DataCite konformes Ausgabeformat herunterladen. Für den Ersteller ermöglicht das neue Metadatenausgabeformat die problemlose Generierung von DOIs über DataCite.

2.1.4.5 Fehlende Werte in der ISO Metadatenausgabe korrigieren und sie vollständig ISO 19139 konform machen

Problem: Der Export der Metadaten in das ISO-Format erzeugt „missing values“ für die Kontaktinformationen. Der aktuelle ISO-Export ist nicht vollständig ISO 19139 konform.

Zielstellung: Als Betrachter möchte ich die Metadaten eines Datensatzes in einem gültigen ISO 19139 Format exportieren. Außerdem sollen die API v2 und die CSW diese Informationen in einer konformen Weise bereitstellen.

2.1.5 ORCID als Identitätsprovider

Problem: Die TISDEX GeoNode-Instanz wird für die Zusammenarbeit mit externen Partner*innen genutzt. Bei einigen dieser Partner*innen handelt es sich um Forschungsinstitute. Wir verfügen derzeit nicht über die Möglichkeit, diese zu autorisieren, ohne gesonderte Konten auf TISDEX für fremde Wissenschaftler*innen einzurichten.

Zielstellung: Als Administrator möchte ich einen ORCID Identitätsprovider konfigurieren (z.B. über keycloak). Als Betrachter möchte ich mich mit meinen ORCID-Zugangsdaten anmelden können. Als Betrachter möchte ich das ORCID-Handle auf der Profilsseite der Person sehen. Außerdem soll diese Information Teil der API v2 Antwort des /users/ Endpunktes sein.

2.1.6 EPIC Erweiterung der Datenexplorationswerkzeuge

Problem: Um eine umfassendere Datenexploration komplexer wissenschaftlicher Datensätze zu ermöglichen, muss die Diversität und Funktionalität der Datenexplorationswerkzeuge erweitert werden. Neben räumlichen Daten benötigen auch nicht-räumliche Datensätze webbasierte Werkzeuge, um Informationen abzurufen und

die Exploration zu fördern. Die Bereitstellung dieser Funktionalität ist komplex und wird daher als EPIC beschrieben, welches aus den folgenden Teilfunktionen besteht:

2.1.6.1 Farbe als dritten Visualisierungsfaktor für Liniendiagramme hinzufügen

Problem: Die derzeit verfügbaren Liniendiagramm- und Balkendiagramm-Widgets sind durch die ausschließliche Verwendung von X- und Y-Parametern eingeschränkt. So können Sie beispielsweise in einem Liniendiagramm bestehend aus *Zeit* x *Wert* x *Typ*, den Faktor *Typ* nicht als Farbe darstellen. Dieser zusätzliche Parameter würde zu einem mehrzeiligen Diagramm führen. Sehen Sie dieses aktuelle Beispiel:

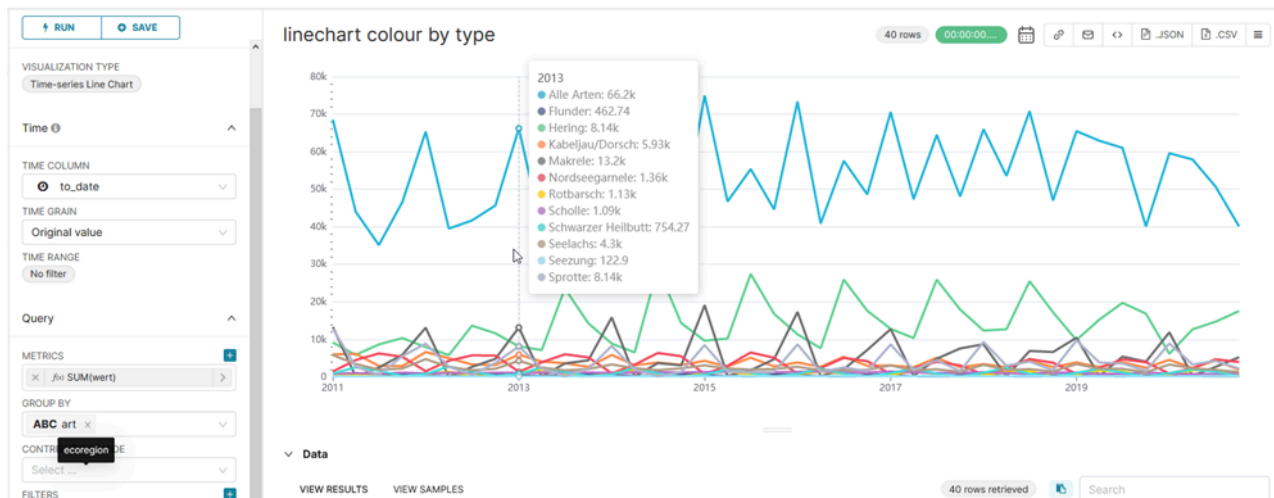
Abbildung 9: Aktuelle Liniendiagramm-Implementierung zur Darstellung von Zeit und Durchschnittswerten für einen Datensatz. Es kann nur eine Farbe für das Diagramm ausgewählt werden, was zu einem einzigen Liniendiagramm führt.



Quelle: Eigene Darstellung.

Zielstellung: Um die Gruppierung von Datensätzen in Diagrammen zu ermöglichen, sollte die Farbe für jede Linie / jeden Balken als Attribut auf der Grundlage kategorischer Feldwerte festgelegt werden. Sehen Sie sich hierzu dieses Beispiel desselben Datensatzes an, der als Apache Superset Liniendiagramm dargestellt wird:

Abbildung 10: Beispiel für ein Apache Superset Liniendiagramm mit Farbe als zusätzliche Gruppierungsebene, wodurch mehrere Linien erzeugt werden.

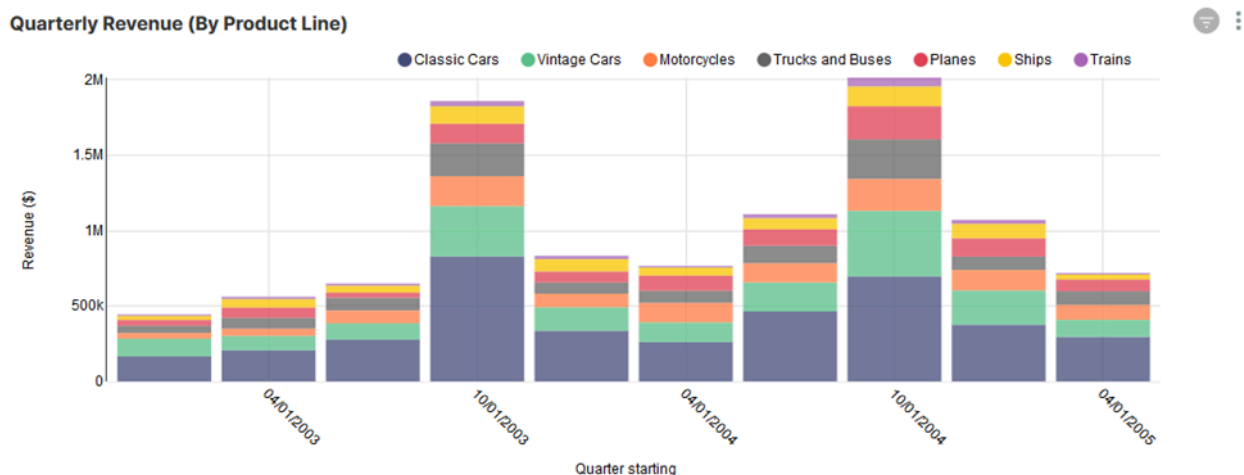


Quelle: Eigene Darstellung.

Die Metrik „Art“ wird hier zur Gruppierung verwendet, und es wird eine Farbrampe verwendet um ein mehrsträngiges Liniendiagramm zu erstellen.

Die gleiche Funktionalität soll für Balkendiagramme zur Verfügung gestellt werden, was zu gestapelten Diagrammen führt, wie hier dargestellt:

Abbildung 11: Beispiel für ein gestapeltes Balkendiagramm.



Quelle: Eigene Darstellung.

Die Funktion ist ebenfalls auf [GitHub](#) beschrieben und die Implementierung soll den dortigen Kommentaren folgen.

2.1.6.2 Hierarchiediagramme als Widgets hinzufügen (z.B. Sunburst)

Problem: Unsere Datensätze enthalten manchmal eine tiefere Kategorisierung zwischen mehreren Attributen. Ein Beispiel ist die durchschnittliche Menge an organischem Kohlenstoff, gruppiert nach Boden und Landnutzungsart in einer bestimmten Ökoregion. Derzeit bietet kein Widget Optionen zur Visualisierung dieser Hierarchie.

Zielstellung: Um eine mehrstufige Visualisierung von Datensätzen zu ermöglichen, soll ein Widget für hierarchische Diagramme entwickelt werden. Ein Beispiel für eine mehrstufige Visualisierung ist ein Sunburst-Diagramm. Die oben erwähnten Bodendaten könnten wie folgt abgebildet werden:

Abbildung 12: Beispiel für ein Sunburst-Diagramm. Die dargestellten Variablen sind der Bodenkohlenstoffbestand pro Ökoregion, die Landnutzungsart und die Bodenart (vom inneren zum äußeren Ring).



Quelle: Eigene Darstellung.

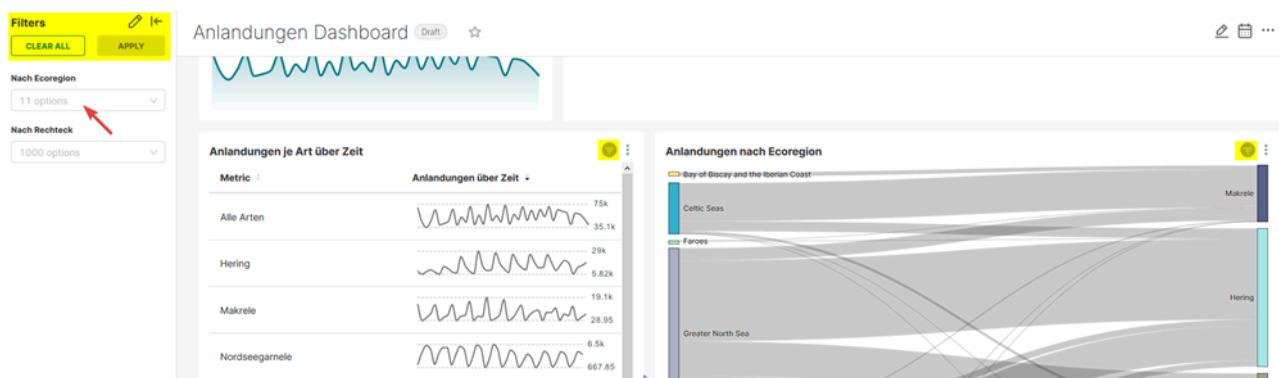
Diese neue Art von Widget soll als Teil von Dashboards konfigurierbar sein.

2.1.6.3 Implementierung von Cross-Widget Filter für Dashboards

Problem: Wissenschaftliche Datensätze sind meist komplex und schwer zu verstehen. Wenn diese Datensätze in Dashboards verwendet werden, ist die Menge der angezeigten Information sehr hoch, was ein einfaches Verständnis erschwert.

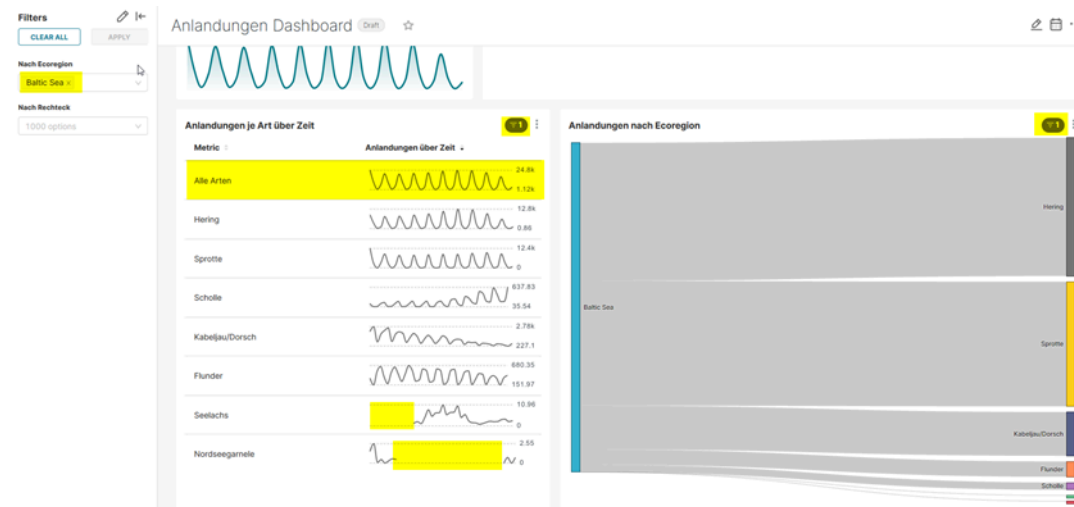
Zielstellung: Um die Datenexploration und -abfrage zu verbessern, sollten die Betrachter in der Lage sein, Dashboard-Widgets zu filtern. Eine funktionale Implementierung ist in Apache Superset zu sehen und wird in den folgenden Screenshots skizziert:

Abbildung 13: Dashboard ohne angewandte Filter.



Quelle: Eigene Darstellung.

Abbildung 14: Dashboard mit angewandtem Ökoregionsfilter. Die Widgets zeigen einen aktiven Filter und haben ihre Diagramme entsprechend angepasst.



Quelle: Eigene Darstellung.

Ersteller sollen in der Lage sein zu definieren, welche Art von Filtern verwendet werden kann und ob es Default-Werte geben soll und welche Widgets die Empfänger des Filters sein sollen. Ein Beispiel für die benötigte Funktionalität wird in dieser Demonstration auf YouTube gezeigt: <https://www.youtube.com/watch?v=Qzv8ZQvLyZg>

Die Funktion ist auch auf [GitHub](#) beschrieben und die Implementierung soll den dortigen Kommentaren folgen.

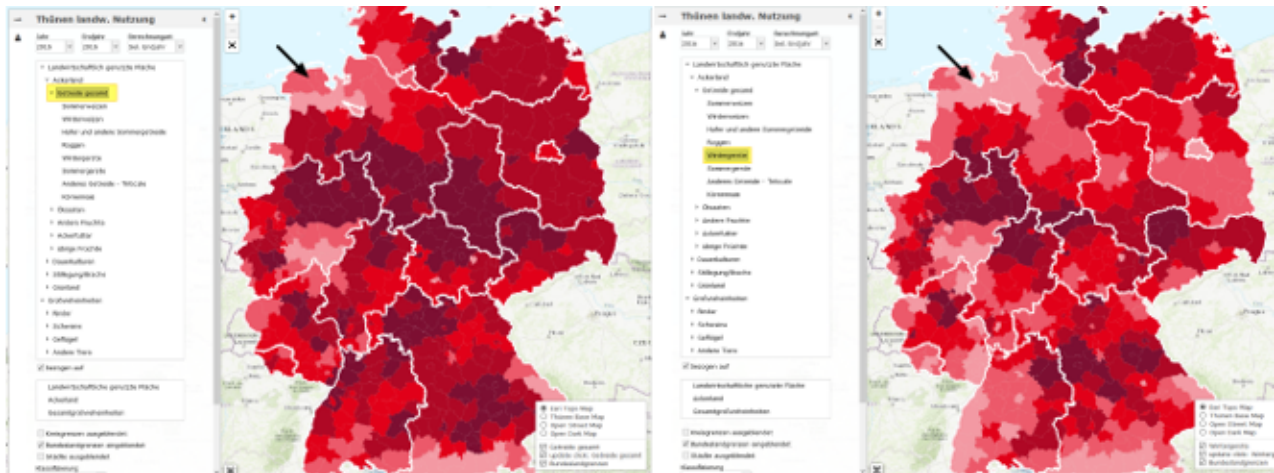
2.1.7 EPIC Mapstore2 UI/UX Verbesserungen

Problem: Der derzeit verwendete Mapping-Client bietet für einige Karten kein gutes Nutzererlebnis. Dies beeinträchtigt den Informationsabruf und führt zu einer geringen Anzahl von Besuchern. Daher muss die Benutzeroberfläche weiter verbessert und weiterentwickelt werden. Die Bereitstellung dieser Funktionalität ist komplex und wird daher als EPIC beschrieben, welches aus den folgenden Teilfunktionen besteht:

2.1.7.1 Einander ausschließende Ebenen

Problem: Bei der Darstellung ganzer Gebiete, wie z.B. aggregierte Daten für Gemeinden, funktioniert die gleichzeitige Anzeige mehrerer Ebenen nicht, da sich die Polygone überschneiden. Betrachten Sie hierzu dieses Beispiel mit zwei verschiedenen Ebenen:

Abbildung 15: Beispiel für die gegenseitige Ausschluss einzelner Ebenen auf Grund von Überschneidungen.



Quelle: Eigene Darstellung.

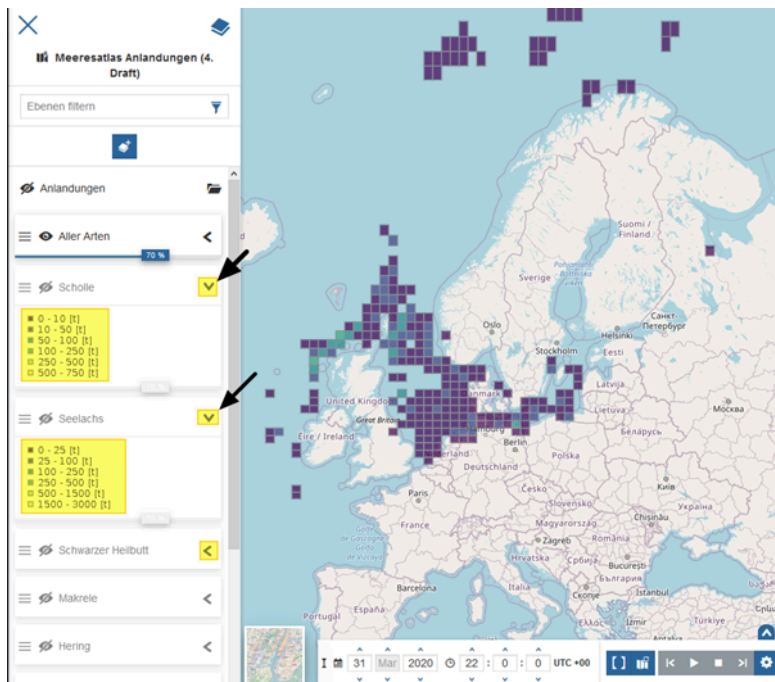
Die einzelnen Ebenen des Inhaltsverzeichnisses schließen sich für die Darstellung in der Kartenansicht gegenseitig aus. Diese Art von Verhalten ist derzeit in GeoNode-Karten nicht möglich.

Zielstellung: Jede Gruppe in einer GeoNode-Karte sollte einen Schalter haben, um festzulegen, ob die Ebenen innerhalb der Gruppe sich gegenseitig ausschließen sollen. Exklusive Ebenen sollten im Inhaltsverzeichnis der Karte ein eignes Aussehen haben, wie z.B. Radiobuttons. Diese Informationen müssen als Teil der Karte gespeichert werden. Ein funktionales Beispiel ist in der folgenden ‚custom App‘ zu sehen:

<https://atlas.thuenen.de/webpace/agraratlas/agraratlas/index.html?LP=2>

2.1.7.2 Schwebendes Legenden-Widget

Problem: Um die kombinierte Legende der sichtbaren Ebenen zu sehen, müssen Sie derzeit jede Ebene innerhalb einer GeoNode-Karte öffnen, was zu vielen Klicks und einer schlechten Nutzererfahrung bei Karten mit vielen Ebenen führt. Schauen Sie sich dieses Beispiel an:

Abbildung 16: Aktuelle Nutzererfahrung der Legendenansicht bei Karten mit vielen Ebenen.

Quelle: Eigene Darstellung.

Zielstellung: Implementierung einer schwebenden Legende, die die Legendenkomponenten der aktuell sichtbaren Ebenen dynamisch als Kartenkomponente wiedergibt. MapStore bietet Widgets an, die über der Kartenansicht schweben, und es gibt ein schwebendes Legenden-Widget in der Codebasis (siehe <https://groups.google.com/g/mapstore-developers/c/19trkb0IIN4>). Das Widget kann jedoch nicht als Teil einer GeoNode-Karte konfiguriert werden. Stattdessen sollte ein Ersteller in der Lage sein, das Widget für eine Karte zu verwenden und seinen Standardstatus (aktiv oder inaktiv) zu definieren. Das schwebende Legenden-Widget sollte die Konfiguration der SLD-Werte wie Zusammenfassung und Titel, wie hier dargestellt, ermöglichen:

Abbildung 17: Beispiel für eine Legende mit den Elementen Zusammenfassung und Titel.

Legende

Veränderung Kohlenstoffvorrat der Baumbiomasse [t/ha]; Zeitraum: 2002 - 2012

- $x \leq -10$
- $-10 < x \leq -5$
- $-5 < x \leq 0$
- $0 < x \leq 5$
- $5 < x \leq 10$
- $10 < x \leq 15$
- $15 < x$

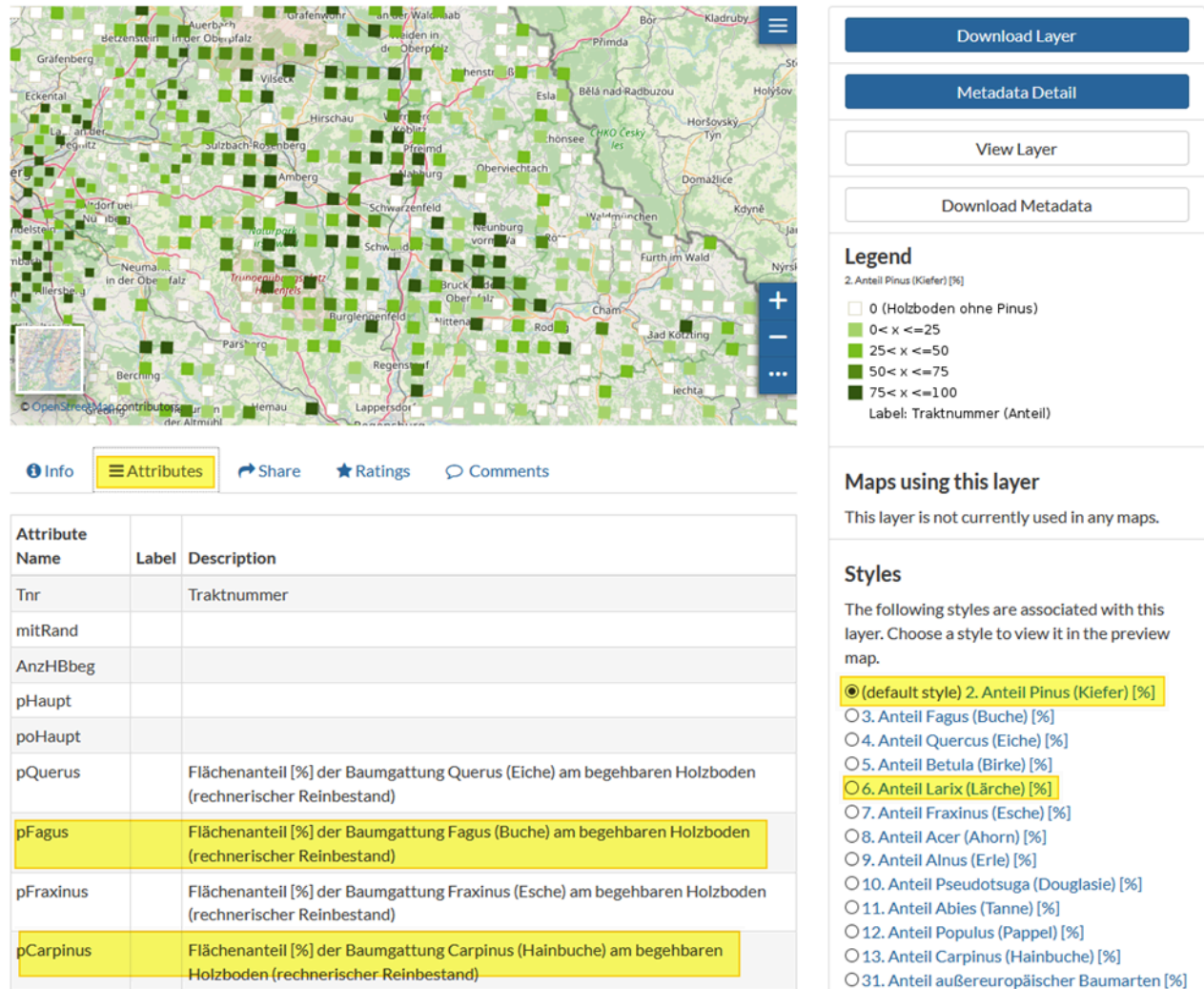
Quelle: Eigene Darstellung.

2.1.7.3 Layer Style Switch

Problem: Das Problem ist bereits auf [GitHub](#) beschrieben. Einige Datensätze bieten mehrere Attribute, die zur Gestaltung von Merkmalen verwendet werden können. Zum Beispiel könnte eine Probenauswahl von Rechtecken Informationen über den Flächenanteil verschiedener Baumarten enthalten, wie hier dargestellt:

Abbildung 18: Beispiel für Datensätze mit mehreren Attributen zur Gestaltung von Kartenmerkmalen.

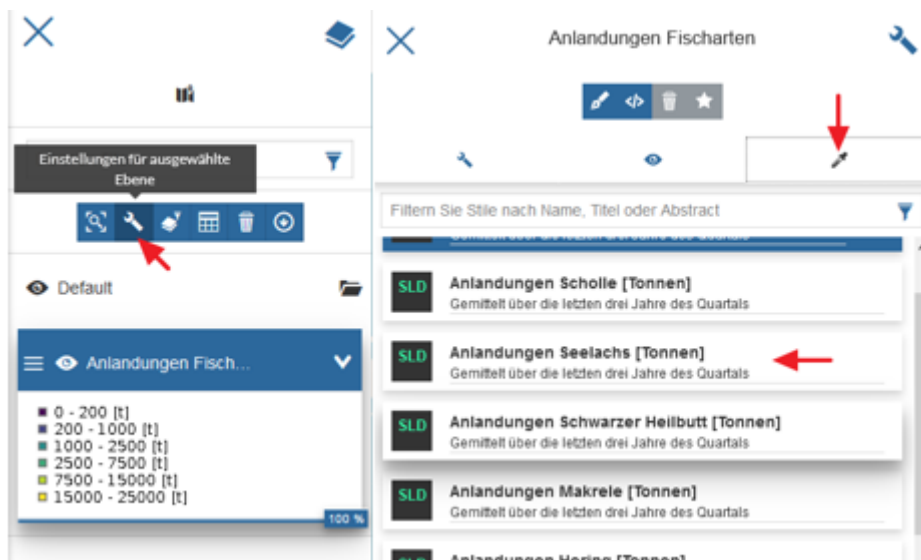
Baumartenverteilung (BWI-2012)



Quelle: Eigene Darstellung.

Um den Stil einer Ebene in einer Karte zu ändern, muss der Ersteller mehrere Klicks verwenden:

Abbildung 19: Beispiel der benötigten Klicks zur Änderung eines Ebenenstils.

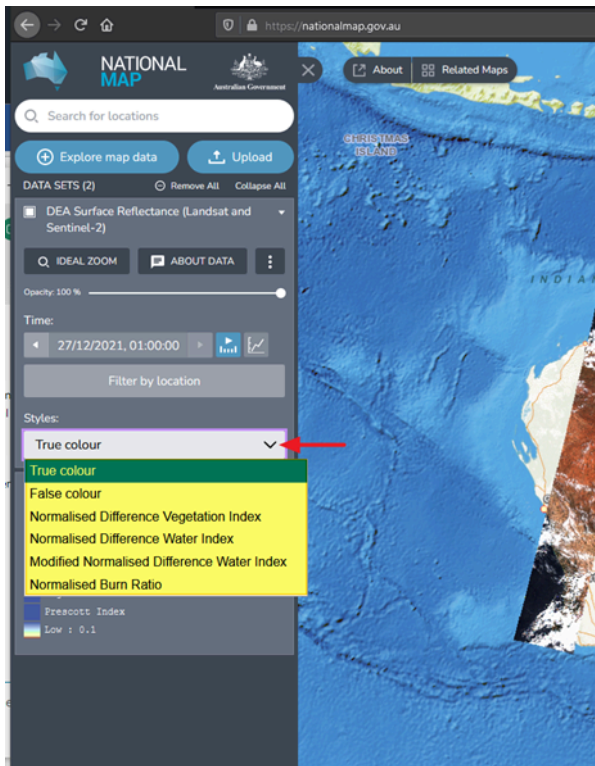


Quelle: Eigene Darstellung.

Ein Betrachter kann die Stile überhaupt nicht ändern.

Zielstellung: Bereitstellung eines Dropdowns zum Umschalten des Ebenenstils für GeoNode-Karten. Ein funktionierendes Beispiel ist Teil der terria.io Karte und sieht wie folgt aus:

Abbildung 20: Beispiel eines Dropdowns zum Umschalten des Ebenenstils.



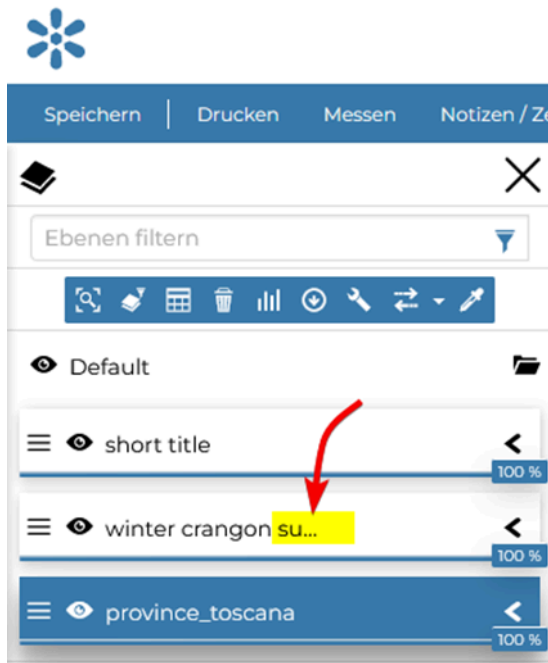
Quelle: Eigene Darstellung.

Der Ebenenstil-Switch sollte mögliche Stile für die gegebene Ebene lesen und deren Titel als Elemente der Dropdown-Liste anzeigen. Ein Betrachter sollte in der Lage sein, die Ebenenstile zu ändern.

2.1.7.4 Lange Titel in TOC umbrechen

Problem: Lange Ebenentitel sind im MapStore TOC nicht lesbar. Siehe:

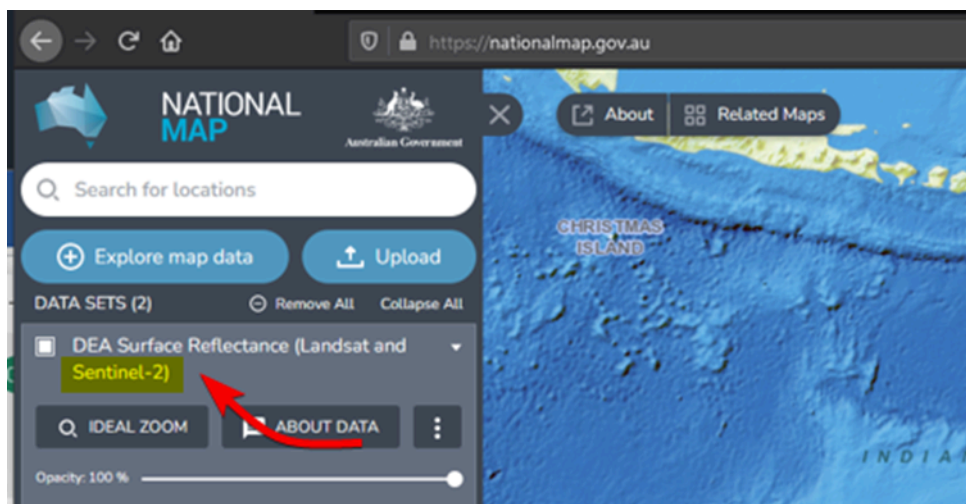
Abbildung 21: Beispiel für die aktuelle Darstellung längerer Titel. Die Titel werden aktuell nicht vollständig angezeigt.



Quelle: Eigene Darstellung.

Zielstellung: Als Betrachter sollte ich in der Lage sein, lange Titel in der Karte zu lesen. Deshalb sollten sie einen Zeilenumbruch verwenden, wie in der terria.io-Karte gezeigt. Siehe:

Abbildung 22: Beispiel für einen Zeilenumbruch in der Darstellung des Titels.



Quelle: Eigene Darstellung.

2.1.8 Implementierung nicht-räumlicher Datensätze als neuer Ressource-Base-Subtyp

Problem: Nicht-räumliche, aber strukturierte Datensätze wie CSV-/Excel-Dateien können als Dokumente hochgeladen werden. Als Dokumentobjekte erben diese Datensätze zwar die Metadatenmodelle der

RessourceBase, können aber nicht auf sinnvolle Weise angezeigt werden. Außerdem verwenden unsere Wissenschaftler*innen häufig PostgreSQL-Datenbanken und -Tabellen. Derzeit können diese in keiner Weise veröffentlicht werden. Wir haben keine Möglichkeit, diese nicht-räumlichen Datensätze in Dashboards oder Karten zu verwenden.

Zielsetzung: Implementierung einer neuen Kategorie von RessourceBase für strukturierte, nicht-räumliche Datensätze. Als Ersteller möchte ich, anstatt den GeoServer-Importer zu verwenden, um beispielsweise Shapefiles in das PostGIS-fähige Backend zu importieren, in der Lage sein, einen Verbindungsstring zu der zu verwendenden Tabelle zu definieren (wenn eine Datenbankverbindung verwendet wird), oder eine CSV-Datei zu importieren. Dieses Problem ist Teil eines GeoNode Improvement Proposal (GNIP) und soll wie auf GitHub angegeben implementiert werden:

<https://github.com/GeoNode/geonode/issues/8714>

Außerdem möchte ich als Ersteller diese Datensätze in Dashboards oder Karten als zusätzliche Datenquellen zur Darstellung von Diagrammen verwenden (siehe [GitHub-Kommentar](#)). Als Betrachter möchte ich die nicht-raumbezogenen Datensätze in einer einfachen tabellarischen Vorschau anzeigen.

2.2 Dokumentation

Die Konfiguration und Nutzung der genannten Funktionen soll in der GeoNode Dokumentation (<https://github.com/GeoNode/documentation>) als Teil des Benutzerhandbuchs (<https://docs.geonode.org/en/master/usage/index.html>) dokumentiert werden.

2.3 Implementation

2.3.1 Layout und Design

Die neuen Komponenten der Benutzeroberfläche basieren auf dem bestehenden MapStore2-Layout und verwenden, soweit möglich, bereits implementierte Anzeigekomponenten wie Schaltflächen, Felder und Texte.

2.3.2 Entwicklung

Die Entwicklung nutzt ein vom Auftraggeber zur Verfügung gestelltes GitLab zur Quellcode-Verwaltung. Abschließend wird die entwickelte Software dem Auftraggeber übergeben und unter der GNU GPLv3 Lizenz zur Verfügung gestellt.

Darüber hinaus müssen die Leistungsanforderungen:

2.1.1, 2.1.4, 2.1.6, 2.1.7, 2.1.8

Teil von Pull Requests für die jeweiligen Open-Source-Repositorien für [GeoNode](#), [MapStore2](#) und den [GeoNode MapStore2 Adapter](#) sein. Die Pull Requests sollen den dortigen Richtlinien entsprechen und akzeptiert werden. Der GitHub Ziel Branch ist hierbei master.

2.3.3 Testdaten

Als Testdaten sollen die über [ATLAS](#) veröffentlichten Datensätze, Dokumente und Karten verwendet werden.

2.3.4 Deployment

Das Zentrum für Informationsmanagement (ZI) des Auftraggebers verwendet die folgenden Komponenten, um die angepasste FDI bereitzustellen:

- Dedizierte Ubuntu 20.04 LTS VM (mindestens zwei Kerne, 4GB RAM),
- Python 3.8 (oder höher) Umgebung,
- GitLab für die Versionsverwaltung,
- NGINX Webserver,
- SSL Zertifikat für verschlüsselte Kommunikation,
- Docker und Docker-Compose

Der Auftragnehmer liefert Installationsanweisungen für eine minimal funktionierende Beispielanwendung, bestehend aus:

- Zusätzlich zu installierenden Softwarepaketen,
- Anleitungen für die Ersteinrichtung (z.B. Docker-Compose o.ä.),
- Instruktionen zum Starten eines lokalen Entwicklungsservers

Zu diesem Zweck hat der Auftragnehmer Zugang zum GitLab des AG. Das Deployment der angepassten Applikation in die Produktion ist nicht Teil dieser Leistungsbeschreibung und wird vom Team des ZI übernommen.

Literaturverzeichnis

DataCite Metadata Working Group. (2021). DataCite Metadata Schema Documentation for the Publication and Citation of Research Data and Other Research Outputs. Version 4.4. DataCite e.V. <https://doi.org/10.14454/3w3z-sa82>

Gärtner, P., Svoboda, N., Kühnert, T., Zoader, M. A., Heinrich, U. (2017). The BonaRes Metadata Schema. BonaRes Series. DOI: 10.20387/bonares-5pgg-8yyp