Fachhochschule Nordwestschweiz
Hochschule für Technik

artha
Safer Water, Safer Life

Artha - Data-Hub

Semesterprojekt IP5

Windisch, HS24

Studentin/Student	Damjan Stojanovic Stefan Simic
Fachbetreuer/in	Norbert Seyff Nitish Patkar
Auftraggeberin	Patrick Frei
Projektnummer	24HS IIT22

Fachhochschule Nordwestschweiz, Hochschule für Technik

Abstract

Das Unternehmen *Artha* ist spezialisiert auf die Bereitstellung innovativer Technologien für die Wasseraufbereitung und -überwachung. Es bietet massgeschneiderte Lösungen zur Analyse der Wasserqualität in Klär- und Wasseraufbereitungsanlagen an. Durch den Einsatz von UV-Spektrometern werden detaillierte Messdaten erfasst, die wichtige Informationen über die Zusammensetzung und Qualität des Wassers liefern.

Im Rahmen des Projekts *Artha - Watersense* wurde bereits eine Cloud-basierte Lösung implementiert, die den bisherigen, lokal begrenzten Ansatz überwindet. Die neue Lösung ermöglicht es, die von den UV-Spektrometern erfassten Daten in eine zentralisierte Cloud-Datenbank hochzuladen. Über ein intuitives Dashboard können die Daten visualisiert und analysiert werden. Diese Entwicklung hat die Datenverfügbarkeit und -nutzung verbessert, da Messwerte ortsunabhängig und nahezu in Echtzeit zugänglich sind.

Trotz dieser Fortschritte bestehen weiterhin Herausforderungen in den Bereichen Skalierbarkeit, Erweiterbarkeit und Benutzerfreundlichkeit. Das Ziel des vorliegenden Projekts ist es daher, ein Konzept für einen erweiterten Data-Hub zu entwickeln, der diese Schwächen adressiert. Gleichzeitig wird eine Teilintegration des Konzepts vorgenommen, um die praktische Umsetzbarkeit und den Mehrwert für die bestehende Lösung zu überprüfen.

Contents

List of Figures

List of Tables

1 Einleitung

1.1 Hintergrund und Motivation

Das Projekt *„Artha - Data-Hub“* adressiert zentrale Herausforderungen im Bereich der Wasserqualitätsanalyse. Derzeitige Lösungen, wie sie im Projekt *Artha - Watersense* zum Einsatz kommen, bieten grundlegende Funktionen zur Messdatenspeicherung und Analyse. Allerdings sind diese Systeme oft starr und lokal begrenzt, was zu ineffizienten Prozessen und erhöhtem Verwaltungsaufwand führt. Mit der Entwicklung eines flexiblen und skalierbaren Data-Hubs soll die Grundlage geschaffen werden, um Messdaten effizienter zu verwalten, zu visualisieren und nutzbar zu machen. Dies bietet sowohl für Betreiber von Kläranlagen als auch für Trinkwasserversorger deutliche Vorteile, insbesondere in der Reaktionsgeschwindigkeit auf Abweichungen.

1.2 Zielsetzung und Vision des Projekts

Das Hauptziel des Projekts ist es, ein Konzept für einen Data-Hub zu entwickeln, der bestehende technische und organisatorische Einschränkungen überwindet. Ein Fokus liegt auf der Integration von Echtzeit-Datenquellen, einer verbesserten Skalierbarkeit und der benutzerfreundlichen Visualisierung von Messdaten. Durch die Teilintegration in die bestehende Lösung *Artha - Watersense* wird überprüft, ob das Konzept praxistauglich ist und wie es als Grundlage für zukünftige Erweiterungen dienen kann.

1.3 Relevanz des Projekts und Anwendungsbereich

Die Relevanz des Projekts ergibt sich aus den steigenden Anforderungen an die Wasseraufbereitung und -überwachung, insbesondere in Hinblick auf Nachhaltigkeit, Effizienz und Datensicherheit. Mit einem Data-Hub können Betreiber von Wasseraufbereitungsanlagen ihre Ressourcen effizienter einsetzen, indem sie Trends und Anomalien frühzeitig erkennen und entsprechende Massnahmen ergreifen. Zudem trägt das Projekt zur Optimierung bestehender Lösungen wie *Artha - Watersense* bei und bietet eine Grundlage für weitergehende Entwicklungen, wie die Integration von KI-gestützten Analysen.

1.4 Aufbau der Dokumentation

Diese Dokumentation beschreibt systematisch die Konzeption und Teilintegration des Data-Hubs. Nach der Darstellung der Ausgangslage und der theoretischen Grundlagen folgt die detaillierte Beschreibung des entwickelten Konzepts. Anschliessend wird die Teilintegration in die bestehende Lösung diskutiert. Den Abschluss bilden ein Fazit und ein Ausblick auf zukünftige Arbeiten. Ziel ist es, die Ergebnisse klar und nachvollziehbar zu präsentieren und gleichzeitig die Relevanz des Projekts im Kontext moderner Wasseraufbereitungstechnologien hervorzuheben.

Aufbau von Konzeptlösungen

Die Konzeptlösungen für die verschiedenen Aspekte des Data-Hubs werden in etwa gleich aufgebaut sein um den Lesefluss zu fördern.
Dazu gehören beispielsweise:

- Motivation / Hintergrund
- Anforderungen (Funktional und Nichtfunktional)
- Fazit

1.5 Recherchevorgehen

Unser Rechercheprozess basiert auf einer Kombination bewährter und moderner Ansätze, um eine umfassende und fundierte Grundlage für unser Projekt zu schaffen. Zu den Hauptinstrumenten gehören:

- Google Scholar: Für die gezielte Suche nach wissenschaftlichen Artikeln, Publikationen und anderen akademischen Quellen, die relevante theoretische Erkenntnisse liefern.
- Allgemeine Google-Recherche: Die Google Recherche wird verwendet um breiter gefasste Informationen und praktische Anwendungen zu finden.
- KI-basierte Chatbots: Durch den Einsatz moderner KI-Tools wie ChatGPT werden Informationen strukturiert und erweitert. Dieser Ansatz ermöglicht es uns auch alternative Perspektiven einzuholen.

In den jeweiligen Kapiteln des Dokuments wird der spezifische Rechercheansatz beschrieben, um die gesammelten Informationen nachvollziehbar zu machen.

2 Ausgangslage / Theoretische Grundlage für einen Data-Hub

2.1 Ausgangslage

Hintergrund und Motivation

Die Qualität von Wasser ist entscheidend für den Schutz der Gesundheit und die effiziente Nutzung von Ressourcen in Klär- und Wasseraufbereitungsanlagen. Im Rahmen des Projekts *Artha - Watersense* wurde bereits eine Cloud-basierte Lösung implementiert, die wesentliche Verbesserungen gegenüber traditionellen lokalen Systemen bietet. Diese Lösung ermöglicht es, Messdaten, die von UV-Spektrometern erfasst werden, direkt in die Cloud hochzuladen und über ein zentrales Dashboard zu visualisieren. Die Cloudlösung stellt einen grossen Fortschritt dar, da sie die ortsgebundene Analyse aufhebt und die Daten nun unabhängig von Standort und Zeit zugänglich macht.

Aktuelle Lösung und ihre Stärken

Die bestehende Cloudlösung basiert auf der Google Cloud Platform und umfasst folgende wesentliche Funktionen:

- **Zentralisierte Datenspeicherung:** Messdaten werden in einer Cloud-SQL-Datenbank gespeichert, wodurch sie von verschiedenen Endgeräten aus zugänglich sind.
- **Dashboard zur Visualisierung:** Ein intuitives Dashboard ermöglicht die Anzeige von Trends und Anomalien über Visualisierungen wie Liniendiagramme und Heatmaps.
- **Zeitnahe Datenverfügbarkeit:** Messdaten stehen innerhalb von Minuten nach der Erfassung für Analysen bereit.

Durch diese Funktionen hat die Lösung den Zugang zu Messdaten erheblich erleichtert und die Effizienz in der Datenverarbeitung und -überwachung gesteigert.

Herausforderungen und Verbesserungsbedarf

Trotz der bisherigen Erfolge gibt es immer noch einige Herausforderungen und Verbesserungspotenziale, die angegangen werden müssen:

- **Skalierbarkeit:** Die aktuelle Datenbankstruktur ist an feste Datenformate gebunden, was die Integration neuer Sensoren oder Anlagen aufwendig macht.
- **Datenqualität und Flexibilität:** Es fehlen Mechanismen zur automatischen Sicherstellung und Überprüfung der Datenqualität, insbesondere bei der Einbindung neuer Datenquellen.
- **Erweiterbarkeit:** Die Architektur der Lösung ist nicht ausreichend darauf ausgelegt, zukünftige Anforderungen wie die Integration von KI-gestützten Analysen oder zusätzlichen Visualisierungen effizient umzusetzen.
- **Benutzerzentrierung:** Obwohl das Dashboard grundlegende Visualisierungen bietet, besteht Potenzial für eine benutzerfreundlichere Gestaltung, die komplexe Daten einfacher interpretierbar macht.

Relevanz der Weiterentwicklung

Die bestehende Cloudlösung bildet eine solide Grundlage, doch um den steigenden Anforderungen gerecht zu werden, ist eine Weiterentwicklung notwendig. Insbesondere die Flexibilität und Skalierbarkeit der Lösung müssen verbessert werden, um neue Sensoren und Datenquellen einfacher zu integrieren. Ein optimiertes Dashboard mit erweiterten Visualisierungen und intuitiver Navigation wird zudem die Nutzerfreundlichkeit erhöhen und die Entscheidungsfindung beschleunigen. Die geplante Konzeptentwicklung und die Teilintegration neuer Ansätze sollen dazu beitragen, diese Ziele zu erreichen und die Lösung zukunftsfähig zu machen.

2.2 Theoretische Grundlagen

Das Ziel dieses Projekts ist es, die Basis für einen skalierbaren und flexiblen Data-Hub zu schaffen. Da es sich in erster Linie um die Entwicklung einer Konzeptlösung handelt, ist es entscheidend, ein solides theoretisches Fundament durch eine fundierte Literaturrecherche zu legen. Diese Recherche hilft dabei, bewährte Ansätze, aktuelle Technologien und relevante wissenschaftliche Erkenntnisse zu identifizieren und für die Konzeptentwicklung nutzbar zu machen.

Evaluierung eines Data-Hubs

Die präzise Definition eines Data-Hubs stellt eine Herausforderung dar, da keine einheitliche, allgemeingültige Definition existiert. Stattdessen wird ein Data-Hub häufig durch seine wesentlichen Merkmale und Funktionen beschrieben. Laut Gadepally et al. [1] gehören dazu:

- **Datenkataloge:** Diese ermöglichen eine zentrale Übersicht über vorhandene Datenquellen und deren Metadaten, wodurch die Auffindbarkeit und Nachvollziehbarkeit verbessert wird.
- **Verknüpfung mit Dateneigentümern:** Ein Data-Hub bietet Mechanismen, um Daten mit den jeweiligen Verantwortlichen zu verbinden, wodurch eine klare Verantwortlichkeit sichergestellt wird.
- **Datenvisualisierung:** Die Möglichkeit, Daten verständlich und übersichtlich darzustellen, ist ein zentraler Bestandteil eines Data-Hubs, insbesondere für die Entscheidungsfindung.
- **Integration und Zugriffskontrolle:** Neben der Darstellung von Daten umfasst ein Data-Hub Funktionen zur Integration verschiedener Datenquellen sowie zur Steuerung des Zugriffs durch Rollen- und Berechtigungskonzepte.

Ein Data-Hub unterscheidet sich somit von traditionellen Datenspeichersystemen, da er nicht nur als zentraler Speicherort fungiert, sondern auch eine Plattform zur Integration, Organisation und Analyse von Daten bietet. Er stellt sicher, dass Daten nutzbar gemacht werden können, indem sie kontextualisiert, zugänglich und verständlich aufbereitet werden.

Vergleich: Data Hub, Data Lake und Data Warehouse

Für ein Datenmanagementsystem eignen sich nebem dem *Data Hub* auch noch andere Konzepte wie der *Data Lake* und das Data Warehouse. In diesem Abschnitt wollen wir herausfinden wieso sich für unser Projekt, ein Data Hub am besten eignet. Die Konzepte *Data Hub*, *Data Lake* und *Data Warehouse* spielen eine zentrale Rolle in der modernen Datenarchitektur, unterscheiden sich jedoch erheblich in ihrem Zweck, ihrer Struktur und ihren Anwendungsbereichen [2].

Ein **Data Hub** dient primär der Integration und Zentralisierung von Daten aus verschiedenen Quellen. Es ermöglicht den Austausch und die Synchronisation von Daten zwischen Systemen, ohne die Daten vollständig zu migrieren. Data Hubs sind besonders geeignet, wenn der Fokus auf Interoperabilität, Datenverwaltung und der Bereitstellung kontextbezogener Daten für mehrere Anwendungen liegt. Sie betonen Flexibilität und Zugänglichkeit, indem sie Daten dynamisch verknüpfen und nicht duplizieren.

Im Gegensatz dazu ist ein **Data Lake** darauf ausgelegt, grosse Mengen unstrukturierter und strukturierter Daten zu speichern. Es handelt sich um einen kosteneffizienten Ansatz, bei dem die Daten in ihrem Rohformat vorgehalten werden, um sie später für Analysen, maschinelles Lernen oder andere datenintensive Prozesse aufzubereiten. Während Data Lakes grosse Flexibilität bieten, ist ihre Datenqualität oft uneinheitlich, da keine strengen Standards für die Datenspeicherung gelten.

Ein **Data Warehouse** hingegen bietet eine strukturierte und kuratierte Umgebung, die speziell für analytische Zwecke und Business Intelligence optimiert ist. Daten in einem Data Warehouse sind bereinigt, aggregiert und in einer einheitlichen Struktur organisiert, wodurch schnelle und konsistente Berichte ermöglicht werden. Die Hauptnutzung eines Data Warehouses liegt in der Unterstützung strategischer Entscheidungen durch historische Analysen.

Bewertung und Auswertung:

Da bei Artha mit heterogenen Daten aus unterschiedlichen Sensoren und Systemen gearbeitet wird, eignet sich der Data Hub am besten. Er ermöglicht eine flexible Integration und Verwaltung dieser Daten, ohne sie zu duplizieren, und bietet gleichzeitig Mechanismen für Echtzeit-Datenverarbeitung und Zugriffskontrolle. Diese Eigenschaften machen ihn zur idealen Lösung für die Anforderungen eines Data-Hubs.

Data-Hub Types und ihre Stärken / Schwächen

Die Architektur eines Data-Hubs kann verschiedene Formen annehmen, je nach spezifischen Anforderungen an die Datenintegration und -bereitstellung. Im Folgenden werden vier gängige Typen von Data-Hubs beschrieben, die jeweils unterschiedliche Vor- und Nachteile bieten [3]:

Verteilte Hubs

Bei verteilten Hubs verbleiben die Daten bei den jeweiligen Produzenten, die für die Konvertierung in ein standardisiertes Format verantwortlich sind. Der Data-Hub dient als zentraler Metadatenkatalog, der die Datenquellen verknüpft und die Daten in Echtzeit abrufbar macht.

Vorteile:

- Geringe Speicher- und Rechenanforderungen beim Hub.
- Lokale Kontrolle über die Daten bleibt erhalten.

Nachteile:

- Begrenzte Abfragefunktionen.
- Erfordert hohe technische Kapazitäten bei den Produzenten.

Zentralisierte Hubs

Hier werden die Daten von den Produzenten an den Hub übertragen, der sie speichert, standardisiert und über einen zentralen Metadatenkatalog bereitstellt. Dies erleichtert die Datenverwaltung und sorgt für eine einheitliche Datenstruktur.

Vorteile:

- Daten sind jederzeit verfügbar, unabhängig vom Status der Produzenten.
- Ermöglicht umfangreiche Datenabfragen und Analysen.

Nachteile:

- Hohe Speicher- und Rechenanforderungen für den Hub.
- Abhängigkeit von der zentralisierten Infrastruktur.

Gemischte Hubs: Produzenten senden Daten an den Hub

Ein hybrider Ansatz, bei dem Produzenten ihre Daten an den Hub übertragen, jedoch weiterhin lokal Zugriff auf die Daten haben. Der Hub fungiert sowohl als Datenspeicher als auch als zentraler Verknüpfungspunkt.

Vorteile:

- Unterstützt sowohl lokale als auch zentrale Datenzugriffe.
- Daten sind bei Ausfällen des Hubs lokal verfügbar.

Nachteile:

- Komplexere Infrastruktur und Datenabstimmung.
- Abhängigkeit von der Standardisierung durch die Produzenten.

Gemischte Hubs: Hub holt Daten von Produzenten

In diesem Modell ruft der Hub die Daten bei den Produzenten ab und speichert sie zentral. Dies reduziert die technische Last bei den Produzenten und erhöht die Verfügbarkeit der Daten.

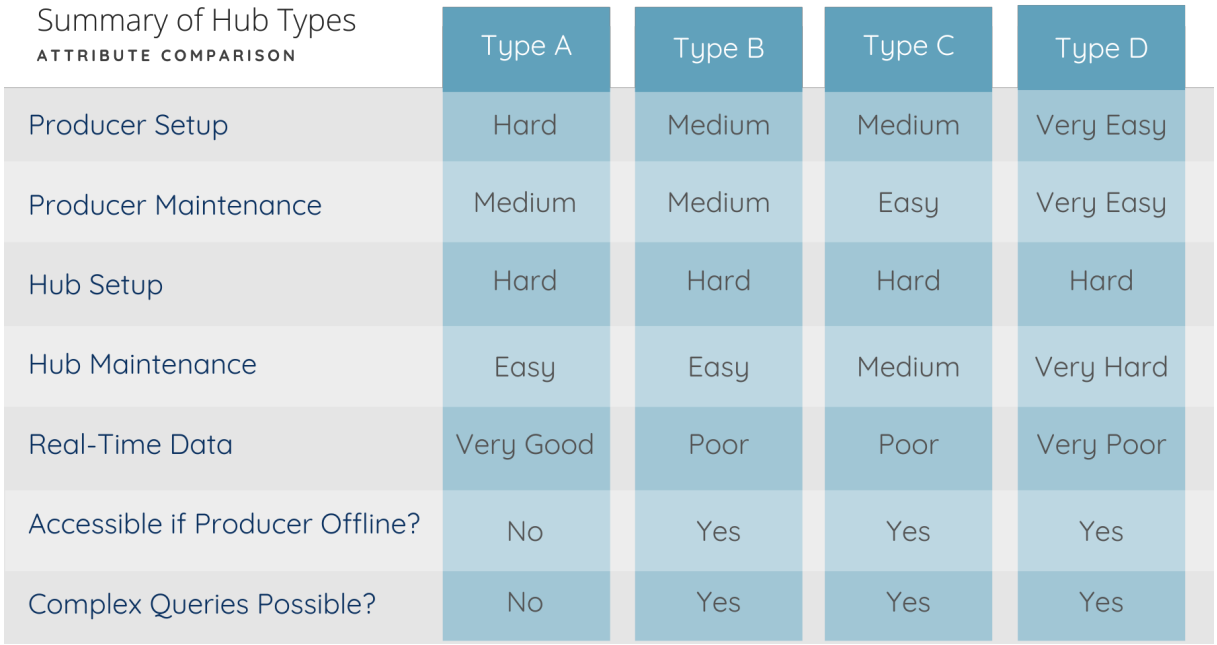
Vorteile:

- Minimale Anforderungen an die Produzenten.
- Zentralisierte Kontrolle über Datenformate und Qualität.

Nachteile:

- Erhöhte Rechen- und Speicheranforderungen beim Hub.
- Risiko von Verzögerungen beim Abruf grosser Datenmengen.

Die folgende Grafik von Internet of Water [3] gibt einen Überblick über weitere Informationen und Eigenschaften von den verschiedenen Data-Hub Typen, wobei gilt:
Type A: Verteilte Hubs
Type B: Gemischte Hubs: Produzenten senden Daten an den Hub
Type C: Gemischte Hubs: Hub holt Daten von Produzenten
Type D: Zentralisierte Hubs



Summary of Hub Types ATTRIBUTE COMPARISON	Type A	Type B	Type C	Type D
Producer Setup	Hard	Medium	Medium	Very Easy
Producer Maintenance	Medium	Medium	Easy	Very Easy
Hub Setup	Hard	Hard	Hard	Hard
Hub Maintenance	Easy	Easy	Medium	Very Hard
Real-Time Data	Very Good	Poor	Poor	Very Poor
Accessible if Producer Offline?	No	Yes	Yes	Yes
Complex Queries Possible?	No	Yes	Yes	Yes

Figure 2.1: Attribute der unterschiedlichen Data Hubs

Auswahl des geeigneten Data-Hub-Typs

Die Wahl des richtigen Data-Hub-Typs hängt von den spezifischen Anforderungen des Projekts ab. Faktoren wie die technischen Kapazitäten der Datenproduzenten, die erforderliche Skalierbarkeit und die Komplexität der Datenintegration sollten sorgfältig abgewogen werden, um eine optimale Lösung zu gewährleisten. Die schlussendliche Entscheidung für einen geeigneten Data-Hub-Typs für Artha wird in den folgenden Kapiteln diskutitiert und evaluiert.

3 Abstraktes Konzept eines Data Hubs

3.1 Hintergrund und Motivation

In der modernen datengetriebenen Welt sind Data Hubs essenzielle Systeme, die es Organisationen ermöglichen, Daten aus verschiedenen Quellen zentral zu integrieren, zu verarbeiten und für unterschiedliche Anwendungen bereitzustellen. Sie bieten eine einheitliche Plattform für den Umgang mit Daten in einer zunehmend fragmentierten und vernetzten Datenlandschaft. Data Hubs dienen nicht nur als Datenspeicher, sondern auch als Dreh- und Angelpunkt für Datenverarbeitung, Anonymisierung, Zugriffskontrolle und Verteilung an berechtigte Nutzer oder Systeme.

Der Bedarf an solchen zentralen Datenplattformen wächst mit der zunehmenden Digitalisierung von Prozessen und der steigenden Menge an generierten Daten. Dies gilt insbesondere in Bereichen wie dem Gesundheitswesen, der Forschung und der Umweltüberwachung, wie Wasserdaten, wo Daten aus unterschiedlichsten Quellen aggregiert und aufbereitet werden müssen.

3.2 Ziel des Konzepts

Das Ziel dieses Konzepts ist es, eine allgemeine, abstrakte Grundlage für die Planung und den Aufbau eines Data Hubs zu schaffen. Dabei werden folgende Aspekte besonders berücksichtigt:

1. **Integration von Datenquellen:** Einheitliche Anbindung unterschiedlicher Datenquellen (z. B. Sensoren, Datenbanken, APIs).
2. **Datenmanagement:** Effiziente Verarbeitung, Speicherung und Klassifizierung von Daten (z. B. öffentlich, privat, anonymisiert).
3. **Zugriffskontrolle:** Sicherstellung, dass nur berechtigte Nutzer auf sensible Daten zugreifen können.
4. **Benutzerinteraktion:** Bereitstellung einer intuitiven und sicheren Schnittstelle (z. B. Dashboard) für die Verwaltung und Visualisierung von Daten.
5. **Flexibilität und Skalierbarkeit:** Anpassungsfähigkeit an wachsende Anforderungen und verschiedene Anwendungsfälle.

3.3 Abgrenzung und Fokus

Dieses Konzept bleibt bewusst abstrakt und universell, um für verschiedene Szenarien und Branchen anwendbar zu sein. Es liefert keine technologischen oder implementierungsbezogenen Details, sondern fokussiert sich auf **allgemeine Prinzipien** und **Best Practices**. Erst in der Konzeptlösung werden mögliche technische Umsetzungen (z. B. Service-Architekturen oder Datenanonymisierungsmethoden) detailliert betrachtet.

3.4 Anforderungen an den Data Hub

Ein moderner Data Hub muss eine Vielzahl von Anforderungen erfüllen, um eine effiziente, sichere und flexible Datenverwaltung zu gewährleisten. Diese Anforderungen lassen sich in zwei Hauptkategorien unterteilen:

- **Funktionale Anforderungen:** Beschreiben die konkreten Aufgaben und Funktionen des Systems.
- **Nicht-funktionale Anforderungen:** Stellen die Qualität und Betriebsfähigkeit des Systems sicher.

Funktionale Anforderungen an einen Data Hub

Anforderung	Beschreibung	Beispiel
FA-1: Integration von Datenquellen	Der Data Hub muss in der Lage sein, Daten aus verschiedenen Quellen (z. B. Datenbanken, APIs, IoT-Geräten) zu integrieren und einheitlich aufzubereiten.	Einbindung von Echtzeit-Sensordaten und historischen Datenbanken, um eine zentrale Datenquelle bereitzustellen.
FA-2: Datenverwaltung	Der Data Hub muss Funktionen zur Datenbereinigung, Validierung und Konsistenzprüfung bereitstellen, um die Qualität der Daten sicherzustellen.	Automatische Duplikaterkennung und Validierung von Datensätzen bei der Integration neuer Quellen.
FA-3: Datenklassifizierung	Der Data Hub muss in der Lage sein, Daten nach Sichtbarkeitsstufen (öffentlich, privat, anonymisiert) zu klassifizieren.	Datenquellen können als öffentlich markiert werden, während andere nur anonymisiert verfügbar sind.
FA-4: Zugriffskontrolle	Der Data Hub muss sicherstellen, dass nur autorisierte Nutzer auf bestimmte Daten zugreifen können, z. B. durch rollenbasierte Zugriffskontrolle (RBAC).	Nur Administratoren dürfen private Datenquellen verwalten, während Analysten nur anonymisierte Daten einsehen dürfen.
FA-5: Datenbereitstellung	Der Data Hub muss Daten in Echtzeit oder Batch-Verfahren für unterschiedliche Nutzer und Systeme bereitstellen.	Echtzeit-Übertragung von Sensordaten an eine Visualisierungssoftware.
FA-6: Analyse- und Abfragefunktionen	Der Data Hub muss grundlegende Abfrage- und Analysefunktionen bieten, um Nutzern die Verarbeitung und Interpretation von Daten zu erleichtern.	Filterung von Datensätzen nach Attributen wie Zeitstempeln oder bestimmten Kategorien.

Table 3.1: Funktionale Anforderungen an einen Data-Hub

Nicht-funktionale Anforderungen an einen Data Hub

Anforderung	Beschreibung	Beispiel
NFA-1: Skalierbarkeit	Der Data Hub muss mit wachsenden Datenmengen und Nutzerzahlen skalieren können.	Unterstützung von horizontaler Skalierung durch Containerisierung.
NFA-2: Sicherheit	Der Data Hub muss die Daten vor unbefugtem Zugriff schützen und eventuell Datenschutzstandards (z. B. DSGVO, HIPAA) einhalten.	Verschlüsselung von Daten im Ruhezustand und während der Übertragung.
NFA-3: Flexibilität und Erweiterbarkeit	Das System muss modular aufgebaut sein, um neue Anforderungen und Datenquellen integrieren zu können.	Einführung neuer Datenformate oder Zugriffskontrollmethoden.
NFA-4: Benutzerfreundlichkeit	Die Benutzeroberfläche muss intuitiv sein, um die Bedienung für Administratoren und Endnutzer zu erleichtern.	Dashboard mit klaren Funktionen zur Verwaltung von Nutzern und Datenquellen.
NFA-5: Zuverlässigkeit und Verfügbarkeit	Der Data Hub muss hochverfügbar sein und Datenverluste vermeiden.	Einsatz von redundanten Speicherlösungen und regelmässigen Backups.
NFA-6: Performance	Der Data Hub muss geringe Latenz bei Datenanfragen gewährleisten, insbesondere bei Echtzeit-Datenquellen.	Caching von häufig angefragten Daten.

Table 3.2: Nicht-funktionale Anforderungen an einen Data-Hub

3.5 Stakeholder

Ein Data Hub dient unterschiedlichen Interessengruppen, die jeweils spezifische Anforderungen und Erwartungen an das System haben. Diese Stakeholder beeinflussen die Konzeption, Entwicklung und den Betrieb des Data Hubs massgeblich. In diesem Abschnitt werden die wichtigsten Stakeholder identifiziert und ihre Ziele sowie Erwartungen an den Data Hub beschrieben.

Nutzergruppen

Das System richtet sich an verschiedene Stakeholder-Gruppen, die spezifische Anforderungen und Erwartungen an die Funktionen eines Data Hubs haben. Zu den Stakeholdern gehören Administratoren, Datenanalysten, Fachexperten, die IT-Abteilung, die Geschäftsleitung, externe Nutzer, Regulierungsbehörden sowie Personen oder Systeme, die Daten in den Data Hub hochladen. Jede dieser Gruppen hat unterschiedliche Bedürfnisse, die von der Verwaltung von Datenquellen über den sicheren Zugriff auf Daten bis hin zur Geschäftsleitung reichen. In der Praxis können einzelne Personen mehrere Rollen gleichzeitig einnehmen, z. B. als Daten Uploader und Administrator oder als Datenanalyst und Fachexperte. Die folgende Tabelle beschreibt die Anforderungen und Ziele der Stakeholder und zeigt auf, wie der Data Hub diese effektiv unterstützen kann.

Stakeholder-Analyse

Stakeholder	Beschreibung	Anforderungen
Administratoren	Personen, die für die Verwaltung und den Betrieb des Data Hubs verantwortlich sind.	- Hinzufügen und Verwalten von Datenquellen. - Definition und Verwaltung von Benutzerrollen und Zugriffsrechten. - Überwachung der Systemleistung und Sicherheit.
Kunde/Daten Uploader	Personen oder Systeme, die neue Datenquellen in den Data Hub hochladen.	- Einfacher und sicherer Upload von Daten. - Möglichkeit zur Angabe von Metadaten (z. B. Sichtbarkeitsstufen, Quelle). - Rückmeldung über den Upload-Status (z. B. Fehlerberichte, Validierungsergebnisse).
Datenanalysten	Nutzer, die Daten aus dem Data Hub analysieren und daraus Erkenntnisse ableiten.	- Zugriff auf relevante Datenquellen (anonymisiert oder vollständig). - Möglichkeit zur Filterung und Analyse der Daten. - Export von Datensätzen für externe Analysen.
Fachexperten	Fachspezifische Nutzer, die die Daten für operative Entscheidungen verwenden.	- Einfacher Zugang zu ausgewählten Datenquellen. - Übersichtliche Darstellung der Daten im Dashboard. - Möglichkeit, Berichte oder Visualisierungen zu erstellen.
IT-Abteilung	Technische Experten, die für die Infrastruktur und technische Umsetzung des Data Hubs zuständig sind.	- Sicherstellung der Skalierbarkeit und Zuverlässigkeit. - Unterstützung bei der Integration neuer Datenquellen. - Implementierung von Sicherheits- und Datenschutzmassnahmen.
Geschäftsleitung	Führungskräfte, die den Data Hub zur Entscheidungsfindung nutzen.	- Zugriff auf aggregierte und anonymisierte Daten.

		- Berichte und Visualisierungen zur Unterstützung strategischer Entscheidungen. - Nachvollziehbarkeit der Datenquellen.
Externe Nutzer	Partner, Kunden oder Behörden, die Zugang zu bestimmten Daten benötigen.	- Zugriff auf anonymisierte oder öffentliche Datenquellen. - Sichere und kontrollierte Freigabe der Daten durch den Data Hub.

Table 3.3: Stakeholderanalyse Data-Hub

Lösungsansatz

Das Ziel eines Data Hubs ist es, Daten aus verschiedenen Quellen zentral zu integrieren, zu verwalten und für unterschiedliche Anwendungen oder Nutzer bereitzustellen, während Sicherheits- und Datenschutzstandards eingehalten werden. Er dient als zentrale Plattform, die den Datenfluss optimiert, Analysefunktionen ermöglicht und eine effiziente Nutzung der Daten sicherstellt.

Dementsprechend wird der Datahub in verschiedene Komponenten aufgeteilt, welche verschiedene Aufgaben übernehmen. Somit kann sichergestellt werden, dass alle Anforderungen erfüllt werden, da die Komponenten jeweils diese Aufgaben haben.

Es gibt die Sensoren, welche Daten liefern. Diese kommen in das System des Datahubs. Dort gibt es verschiedene Layers durch welche die Daten gehen und transformiert, oder ergänzt werden, bevor sie persistiert werden. Die Datenbank kann die Daten zur Weiterverarbeitung und Analyse senden und die Ergebnisse speichern. Falls ein Endnutzer auf die Daten zugreift, geht dieser über den Frontend-Layer (Dashboard) auf die Daten, welche zuerst durch Sicherheits-Layers durchgehen und evtl. anonymisiert werden.

Architektur

Der Data Hub wird als modular aufgebaute Plattform realisiert, die folgende Hauptkomponenten umfasst:

1. **Datenquellenintegration:**
   - Nutzung von standardisierten Schnittstellen (z. B. REST-APIs, Webhooks) und Konnektoren, um Daten aus verschiedenen Quellen zu integrieren.
   - Unterstützung von Echtzeit-Datenstreams und Batch-Verarbeitung.
2. **Datenverarbeitung und -klassifizierung:**
   - Implementierung eines **Service-Layers**, der die Daten filtert, verarbeitet und klassifiziert.
   - Automatische Klassifizierung von Daten in die Kategorien öffentlich, privat und anonymisiert.
   - Einsatz von Anonymisierungsalgorithmen (z. B. Hashing, Pseudonymisierung), um Datenschutzvorgaben zu erfüllen.
3. **Zugriffskontrolle:**
   - Integration von **Role-Based Access Control (RBAC)** für eine klare, rollenbasierte Zugriffskontrolle.
   - Erweiterung mit **Attribute-Based Access Control (ABAC)** für komplexere Anforderungen (z. B. zeit- oder ortsbasierter Zugriff).
   - Daten werden im Service-Layer gefiltert und anonymisiert.

4. **Datenbereitstellung und Benutzerinteraktion:**
   - Entwicklung eines intuitiven Dashboards, das folgende Funktionen bietet:
   - Bereitstellung von APIs zur externen Nutzung von Daten.

Folgende Grafik veranschaulicht eine simple Data-Hub-Architektur:

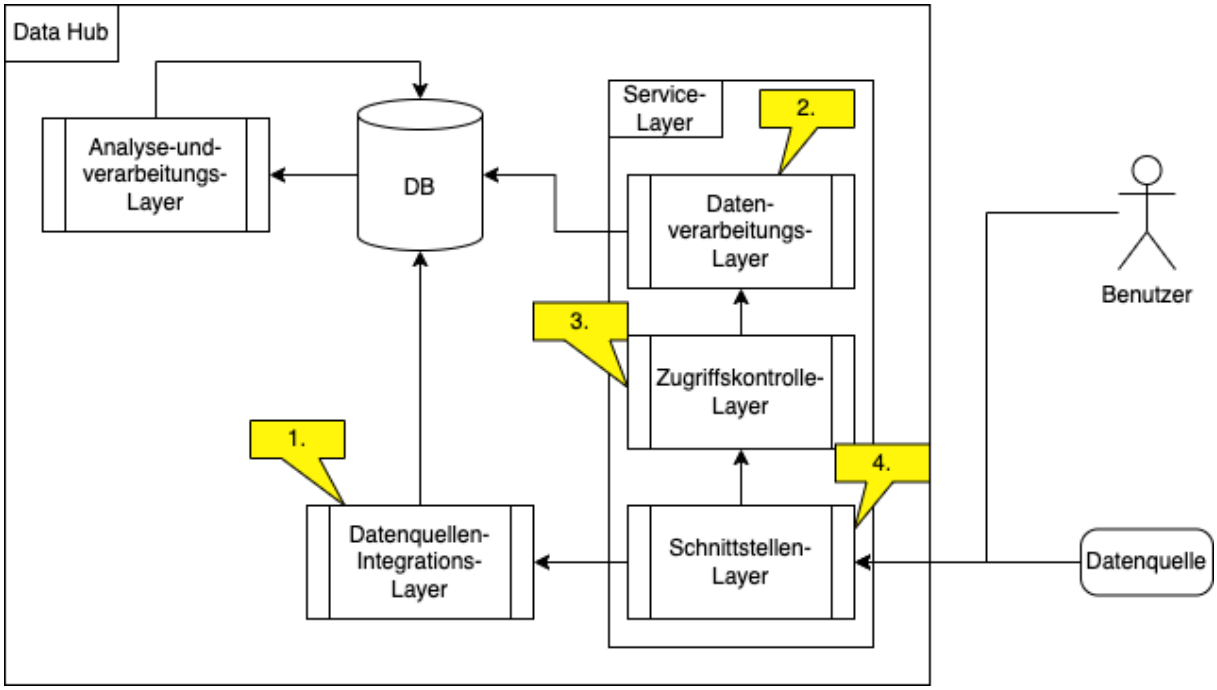


Figure 3.1: Architektur eines Data-Hubs

Der verschiedenen Schichten sind modulare Schichten, welche bestimmte Aufgaben erfüllen und die Daten oder Anfragen dann weiter in der Pipeline senden.

Skalierbarkeit und Flexibilität

1. **Modulare Architektur:**
   - Trennung der Komponenten (z. B. Datenintegration, Service-Layer, Dashboard), um Änderungen und Erweiterungen unabhängig voneinander vornehmen zu können.
2. **Horizontale Skalierbarkeit:**
   - Nutzung von Cloud-Diensten oder Container-Technologien wie Kubernetes, um wachsende Datenmengen und Nutzerzahlen zu bewältigen.
3. **Erweiterbarkeit:**
   - Einfache Integration neuer Datenquellen, Rollen oder Sichtbarkeitsstufen durch konfigurierbare Module.

4 Konzeptlösung eines Data Hubs

Aufbauend auf den Anforderungen und Zielen des Konzepts werden in diesem Kapitel allgemeine, jedoch konkrete Lösungsansätze für die Implementierung eines Data Hubs beschrieben. Diese Lösungen adressieren die zentralen Herausforderungen der Datenintegration, Verwaltung und Bereitstellung und konzentrieren sich auf bewährte Methoden zur Skalierbarkeit, Sicherheit und Benutzerfreundlichkeit. Sie basieren auf aktuellen
Forschungsergebnissen und industriellen Best Practices und bieten eine Grundlage, um den Anforderungen moderner Datenarchitekturen gerecht zu werden.

Am Ende des Kapitels wird eine Beurteilung abgegeben, ob die aktuelle Architektur von Artha geeignet ist für ein Data-Hub und was allenfalls angepasst werden muss.

4.1 Literaturrecherche zu Datahubs

Die Implementierung eines Data Hubs erfordert die Integration bewährter Methoden und aktueller
Forschungsergebnisse, um den vielfältigen Anforderungen gerecht zu werden. Im Folgenden werden zentrale Aspekte und Empfehlungen aus der Literatur und der Industrie vorgestellt, die für die Entwicklung eines effektiven Data Hubs von Bedeutung sind.

1. Architektur und Hauptkomponenten eines Data Hubs
Wie in folgendem Artikel [4] beschrieben, dient ein Data Hub als zentrale Plattform zur Verwaltung, Verarbeitung und Verteilung grosser
Datenmengen aus unterschiedlichen Quellen. Die Hauptkomponenten umfassen Datenintegration, Datenverarbeitung, Speicherung und Bereitstellung. Eine modulare Architektur ermöglicht es, Daten effizient zu integrieren und in Echtzeit bereitzustellen.
2. Best Practices für die Datenintegration
Die Integration heterogener Datenquellen erfordert standardisierte Schnittstellen und flexible Datenmodelle. Der Einsatz von Data Vault Modeling bietet hierbei Vorteile wie hier [5] erwähnt wird, da es eine skalierbare und anpassungsfähige Methode zur Speicherung und Verwaltung von Daten darstellt. Dieses Modell trennt Geschäftslogik von historischen Daten und ermöglicht so eine bessere Nachvollziehbarkeit und Flexibilität, dies kann für gewisse Arten von Datahubs essenziell sein und für andere nicht nötig.
3. Sicherheit und Datenschutz
Die Einhaltung von Datenschutzrichtlinien und die Sicherstellung der Datenintegrität sind essenziell. Plattformen wie der Datahub Europe [6] legen besonderen Wert auf sichere Datenbereitstellung und -nutzung, um vertrauenswürdige KI-Anwendungen zu fördern. Dies unterstreicht die Bedeutung von Sicherheitsmassnahmen und Datenschutz in modernen Datenplattformen.
4. Skalierbarkeit und Flexibilität
Die Fähigkeit, mit wachsenden Datenmengen und sich ändernden Anforderungen umzugehen, ist entscheidend, so Barker [7]. Cloud-basierte Lösungen, wie sie von Google Cloud Plattform angeboten werden, bieten nahezu unbegrenzte Skalierbarkeit und integrierte KI- und Machine-Learning-Funktionen, die für den Aufbau moderner Data Hubs genutzt werden können.
5. Datenarchitektur
Laut diesem Artikel [8] ist eine klare Datenarchitektur notwendig, um Datenqualität und -konsistenz sicherzustellen. Ansätze wie Data Mesh betonen die dezentrale Verantwortung für Datenprodukte und fördern eine skalierbare Datenverwaltung in grossen Organisationen.
6. Technologische Umsetzung
Die Auswahl geeigneter Technologien und Tools ist für die erfolgreiche Implementierung eines Data Hubs entscheidend. Open-Source-Projekte wie DataDock [9] bieten flexible und anpassbare Lösungen für spezifische Anforderungen im Forschungsbereich und können als Grundlage für massgeschneiderte Data-Hub-Lösungen dienen.

Diese Bereiche werden nun einzeln betrachtet und analysiert, welche Best Practices und Industriestandards gelten.

4.2 Architektur und Hauptkomponenten

Ein Data Hub ist eine zentrale Plattform, die Daten aus verschiedenen Quellen zusammenführt, verarbeitet, speichert und für verschiedene Anwendungen oder Nutzer bereitstellt. Anders als klassische Datenspeicher wie Data Warehouses oder Data Lakes legt ein Data Hub den Schwerpunkt auf die Verarbeitung und Weitergabe von Daten in Echtzeit. Die Struktur eines Data Hubs ist so gestaltet, dass er leicht erweitert, angepasst und gewartet werden kann. Dabei werden verschiedene Aufgaben in getrennte Bereiche aufgeteilt, die jeweils spezifische Funktionen übernehmen. Standardisierte Schnittstellen sorgen dafür, dass die einzelnen Teile der Plattform miteinander reibungslos kommunizieren können. So kann der Data Hub Daten effizient verwalten und gleichzeitig hohe Anforderungen an Sicherheit und Datenschutz erfüllen.

Kernfragen der Analyse

- Welche Hauptkomponenten benötigt ein Data Hub?
- Wie kann ein Data Hub flexibel und skalierbar gestaltet werden?
- Welche Standards und Best Practices sollten berücksichtigt werden?

Hauptkomponenten eines Data-Hubs

Laut diesem Artikel [4] erfodert die Implementierung eines Data Hubs bestimmte Hauptkomponenten, die in der Fachliteratur als Standard gelten. Diese Komponenten gewährleisten eine effiziente Integration, Verarbeitung, Speicherung und Bereitstellung von Daten.

Datenintegration: Ein zentrales Modul für die Aufnahme und Harmonisierung von Daten aus verschiedenen Quellen. Die Integration heterogener Datenquellen erfordert standardisierte Schnittstellen und flexible Datenmodelle. Der Einsatz von Data Vault Modeling bietet hierbei Vorteile, da es eine skalierbare und anpassungsfähige Methode zur Speicherung und Verwaltung von Daten darstellt. Dieses Modell trennt Geschäftslogik von historischen Daten und ermöglicht so eine bessere Nachvollziehbarkeit und Flexibilität.

Datenverarbeitung: Diese Schicht ermöglicht die Transformation und Analyse der integrierten Daten. Die Verarbeitungsschicht klassifiziert, transformiert und analysiert die eingehenden Daten. Ein effektiver Data Hub sollte Mechanismen zur Datenklassifizierung und -anonymisierung enthalten, um Datenschutzrichtlinien einzuhalten und die Datenqualität zu sichern.

Datenspeicherung: Ein Speicherbereich, der für die sichere und effiziente Verwaltung der Daten verantwortlich ist. Die Speicherung dient der sicheren und effizienten Verwaltung von Daten im Hub. Die Auswahl geeigneter Speichertechnologien, wie relationale Datenbanken oder NoSQL-Lösungen, ist entscheidend für die Skalierbarkeit und Zuverlässigkeit des Systems.

Zugriffskontrolle: Dieses Modul stellt sicher, dass nur autorisierte Benutzer auf bestimmte Daten zugreifen können. Die Implementierung von Sicherheits- und Berechtigungsmechanismen, wie Role-Based Access Control (RBAC) oder Attribute-Based Access Control (ABAC), gewährleistet den Datenschutz und die Einhaltung gesetzlicher Vorgaben.

Datenbereitstellung: Diese Komponente stellt die aufbereiteten Daten den Endnutzern oder Anwendungen zur Verfügung. Die Bereitstellung von APIs und Dashboards ermöglicht es Benutzern, auf die

Daten zuzugreifen und sie für Analysezwecke zu nutzen. Ein intuitives Benutzerinterface erleichtert die Interaktion mit dem Data Hub.

Diese Komponenten sind in der Fachliteratur als Standard etabliert und bilden die Grundlage für die Entwicklung eines effektiven Data Hubs. Sie ermöglichen eine strukturierte und sichere Datenverwaltung, die den Anforderungen moderner Datenarchitekturen gerecht wird.

Die folgende Tabelle dient noch als Zusammenfassung der oben genannten Hauptkomponenten:

Komponente	**Beschreibung**	**Kernaufgaben**
Datenintegration	Diese Komponente ist verantwortlich für die Verbindung zu externen Datenquellen, wie Datenbanken, APIs und IoT-Geräten.	- Integration von Echtzeit- und Batch-Daten. - Standardisierung der Datenformate. - Bereinigung und Validierung der Daten während des Imports.
Datenverarbeitung	Verarbeitungsschicht, die die eingehenden Daten klassifiziert, transformiert und analysiert.	- Klassifizierung der Daten (öffentlich, privat, anonymisiert). - Anonymisierung und Pseudonymisierung von sensiblen Daten.
Speicherung	Die Speicherung dient der sicheren und effizienten Verwaltung von Daten im Hub.	- Speicherung von Daten im Ruhezustand, je nach Datahub einschliesslich historischer Daten. - Unterstützung von skalierbaren Speicherlösungen (z. B. Cloud, NoSQL).
Zugriffskontrolle	Implementierung von Sicherheits- und Berechtigungsmechanismen.	- Verwaltung von Benutzerrollen und Berechtigungen (z. B. RBAC, ABAC). - Sicherstellung, dass nur autorisierte Nutzer Zugriff auf sensible Daten haben.
Datenbereitstellung	Diese Schicht stellt die Daten den Nutzern und Anwendungen zur Verfügung.	- Bereitstellung von APIs und Schnittstellen für externe Anwendungen. - Unterstützung von Dashboards zur Visualisierung und Analyse der Daten.

Table 4.1: Hauptkomponenten eines Data-Hubs

Fazit Hauptkomponenten

Die Implementierungsweise der einzelnen Komponenten hängt von den Zielen des Datahubs ab, so kann bei der Zugriffskontrolle eine RBAC sinnvoll sein, aber auch eine ABAC, je nach den Anforderungen des Datahubs. Die genannten Komponenten und Kernaufgaben dienen als Standard und können für spezifische Lösungen abweichen.

4.3 Data-Hub flexibel und skalierbar entwerfen

Die Flexibilität und Skalierbarkeit eines Data Hubs sind entscheidende Merkmale wie in diesem Artikel [4] beschrieben, um mit wachsenden Datenmengen, steigenden Nutzerzahlen und sich verändernden Anforderungen Schritt zu halten. Ansätze wie modulare Architekturen, Cloud-basierte Technologien, Containerisierung oder Datenpartitionierung tragen dazu bei, einen zukunftssicheren und anpassungsfähigen Data Hub zu entwickeln. Im Folgenden werden einige dieser Ansätze beschrieben.

Modulare Architektur
Eine modulare Architektur zerlegt die Funktionen eines Data Hubs in separate Komponenten, wie Datenintegration, Verarbeitung, Speicherung und Bereitstellung. Dadurch können einzelne Module unabhängig voneinander entwickelt, erweitert oder ersetzt werden, ohne das gesamte System zu beeinträchtigen. Beispielsweise kann eine Datenbank, die nicht mehr den Leistungsanforderungen entspricht, durch eine neue ersetzt werden, während die restliche Architektur unverändert bleibt. Diese Trennung erleichtert die Wartung, Anpassung an neue Anforderungen und Integration neuer Technologien.

Cloud-basierte Skalierung
Cloud-Dienste wie AWS, Google Cloud oder Microsoft Azure bieten nahezu unbegrenzte horizontale Skalierbarkeit. Sie ermöglichen es, Ressourcen flexibel an den aktuellen Bedarf anzupassen, wodurch sowohl Kosten als auch technische Herausforderungen reduziert werden. Automatische Skalierung von Speicherressourcen bei wachsenden Datenmengen oder der Einsatz serverloser Funktionen für kurzfristige Spitzenlasten sind typische Beispiele für Cloud-basierte Skalierung.

Containerisierung und Orchestrierung
Technologien wie Docker und Kubernetes bieten eine Grundlage für die Portabilität und Skalierbarkeit von Anwendungen. Durch die Containerisierung können einzelne Komponenten des Data Hubs unabhängig von der zugrunde liegenden Infrastruktur betrieben und bei Bedarf automatisch hoch- oder herunterskaliert werden. Dies sorgt nicht nur für Konsistenz zwischen Entwicklungs- und Produktionsumgebungen, sondern auch für eine effiziente Ressourcenverwaltung.

Datenpartitionierung und verteilte Speicherung
Die Aufteilung grosser Datenmengen auf mehrere Knoten verbessert die Effizienz bei Speicherung und Verarbeitung. Verteilte Datenbanken wie Apache Cassandra oder Amazon DynamoDB ermöglichen eine schnellere Verarbeitung durch Parallelisierung und bieten gleichzeitig Redundanz und Ausfallsicherheit. Dies reduziert Engpässe und sorgt für eine bessere Leistung bei grossen Datenvolumen.

API-gesteuerte Architekturen
Durch standardisierte APIs wird eine flexible Schnittstelle geschaffen, über die externe Anwendungen und Systeme auf die Daten des Data Hubs zugreifen können. Dies erleichtert die Integration neuer Anwendungen und erlaubt Anpassungen ohne Auswirkungen auf bestehende Funktionen. Beispiele hierfür sind die Implementierung von REST- oder GraphQL-APIs.

Echtzeit-Datenverarbeitung
Streaming-Technologien wie Apache Kafka oder AWS Kinesis erlauben die Verarbeitung grosser Datenströme in Echtzeit. Dies ist besonders für zeitkritische Anwendungen wie IoT-Datenverarbeitung von Vorteil, da Daten aus mehreren Quellen gleichzeitig verarbeitet werden können, um schnelle Entscheidungen zu ermöglichen.

Best Practices

- Cloud-native Architekturen bieten maximale Flexibilität und Skalierbarkeit. Laut Gadepally et al. (2020) erlauben sie eine dynamische Anpassung an Nutzerbedürfnisse und Datenlast.
- Containerisierung ermöglicht konsistente und portable Anwendungen, die sich effizient über verschiedene Umgebungen hinweg bereitstellen lassen (Bitgrip).
- Datenpartitionierung verbessert die Leistung und Skalierbarkeit durch die Verteilung grosser Datenmengen über verschiedene Speicherorte.
- Echtzeit-Datenverarbeitung erhöht die Reaktionsfähigkeit moderner Anwendungen durch Technologien wie Apache Kafka (Springer Link).

Fazit flexibler Entwurf

Es gibt viele Möglichkeiten ein Datahub-System flexibel und skalierbar zu gestalten. Jede Komponente muss entsprechend so gebaut sein, dass sie nicht zum Flaschenhals wird und das System verlangsamt. Es ist wichtig einen Überblick über die Architektur und deren Stärken und Schwächen zu haben, damit bei ungenügender Leistung die Komponente angepasst oder ausgetauscht wird. Das grösste Risiko ist bei der Datenbank, dass diese zu viele Daten hat und somit langsam auf Anfragen antwortet. Dies kann durch Cloud-Lösungen, sowie Containerisierung gehandhabt werden, was jedoch zu höheren Betriebskosten führt. Somit ist der aller wichtigste Punkt bei einem Datahub, dass die einzelnen Komponenten modular und unabhängig voneinander gebaut werden, damit sie leicht und mühelos ausgetauscht werden können, ohne grossen Aufwand bei den anderen Komponenten. Wenn dies sichergestellt wird, dann ist die Flexibilität im Datahub sicher. Und mit der Flexibilität des Datahubs kann bei steigender Datenfrage (und somit bei der Skalierung) entsprechend gehandelt werden und bspw. ein geeigneter Cloud-Anbieter für die Datenbank benutzt werden oder der Service-Layer kann erweitert oder ersetzt werden.

Standards und Best Practices

Standards und Best Practices in der Entwicklung von Data Hubs sind entscheidend, um eine effiziente, skalierbare und sichere Datenverarbeitung sicherzustellen. Durch den Vergleich mehrerer etablierter Data Hubs lassen sich wichtige Gemeinsamkeiten und bewährte Ansätze identifizieren, die als Grundlage für die Konzeption eines modernen Data Hubs dienen können. In diesem Abschnitt werden einige bekannte Data Hubs untersucht, ihre Ansätze verglichen und daraus Schlussfolgerungen abgeleitet.

Data Hub	**Hauptmerkmale**	**Technologien/Ansätze**
SAP Data Hub [10]	- Unterstützung für Datenmanagement und -integration in komplexen Landschaften. - Verarbeitung von strukturierten und unstrukturierten Daten. - Automatisierung von Datenpipelines und Workflows.	- Container-basierte Architektur für Skalierbarkeit und Flexibilität. - Integration von Daten aus SAP- und Nicht-SAP-Systemen. - Unterstützung für Machine-Learning-Modelle und Big Data Technologien.
Google Cloud Platform (BigQuery)	- Fokus auf skalierbare Datenanalysen. - Echtzeit-Datenverarbeitung. - Globale Datenverfügbarkeit.	- DWH-ähnliche Architektur. - Unterstützung von SQL-ähnlichen Abfragen. - Nutzung von Cloud-native Technologien.

Azure Data Factory	- Pipeline-basierte Datenintegration. - Integration heterogener Datenquellen. - Visuelle Workflow-Erstellung.	- Integration mit Power BI. - Nutzung von Batch- und Echtzeit-Verarbeitungsfunktionen. - Rollenbasierte Zugriffskontrolle.
Snowflake Data Cloud	- Cloud-native Lösung für Datenintegration und -verarbeitung. - Hohe Performance durch Datenkomprimierung und -partitionierung.	- Nutzung von Multi-Cluster-Architekturen. - Unterstützung von Semi- und Unstructured Data. - SQL-ähnliche Abfragen.

Table 4.2: Standards und best practices

Schlussfolgerungen

Aus der Analyse und dem Vergleich der untersuchten Data Hubs lassen sich zentrale Standards und bewährte Ansätze für die Entwicklung moderner Data Hubs ableiten. Diese Schlussfolgerungen bieten eine solide Grundlage für die Konzeption flexibler, skalierbarer und effizienter Datenplattformen.

1. Modularität und Skalierbarkeit: Alle untersuchten Data Hubs setzen auf modulare und skalierbare Architekturen. SAP Data Hub nutzt beispielsweise container-basierte Technologien, während Snowflake mit Multi-Cluster-Architekturen arbeitet. Dies zeigt, dass eine flexible und erweiterbare Architektur entscheidend ist, um sowohl aktuelle Anforderungen zu erfüllen als auch zukünftige Entwicklungen zu unterstützen.
2. Effiziente Datenverarbeitung: Technologien wie Datenkomprimierung und -partitionierung bei Snowflake oder die Nutzung von SQL-ähnlichen Abfragen bei BigQuery zeigen, dass effiziente Datenverarbeitung ein zentrales Ziel ist. Diese Techniken sorgen für eine optimale Performance, selbst bei grossen Datenmengen.
3. Sicherheit und Zugriffskontrolle: Rollenbasierte Zugriffskontrolle, wie sie Azure Data Factory bietet, unterstreicht die Bedeutung von Sicherheitsmassnahmen in der Datenverwaltung. Diese Funktion gewährleistet, dass sensible Daten nur von autorisierten Personen genutzt werden können.

Fazit

Ein moderner Data Hub muss flexibel, skalierbar und anpassungsfähig sein, um den Anforderungen einer datengetriebenen Welt gerecht zu werden. Die wichtigsten Merkmale sind eine modulare Architektur, effiziente Datenverarbeitung, Echtzeit-Fähigkeiten, Automatisierung und eine starke Integration unterschiedlicher Datenquellen. Unternehmen können durch die Berücksichtigung dieser Best Practices leistungsstarke Datenplattformen schaffen, die den technologischen und geschäftlichen Herausforderungen der Zukunft gewachsen sind.

4.4 Bezug auf die Anforderungen des abstrakten Konzepts

Die Konzeptlösung des Data Hubs wurde entwickelt, um die in der Analyse definierten Anforderungen [Im Kapitel 3.4] umfassend zu erfüllen. Dieses Kapitel zeigt auf, wie die vorgeschlagenen Ansätze und Technologien die funktionalen und nicht-funktionalen Anforderungen umsetzen und wie die Hauptkomponenten des Data Hubs gezielt auf die Bedürfnisse der Stakeholder eingehen.

Funktionale Anforderungen (FA)

Anforderung	Beschreibung	Umsetzung
FA-1: Integration von Datenquellen	Der Data Hub muss Daten aus verschiedenen Quellen (z. B. Datenbanken, APIs, IoT-Geräten) integrieren und einheitlich aufbereiten.	- Nutzung von standardisierten APIs (REST, GraphQL) für die Datenintegration. - Modularer Ansatz ermöglicht die einfache Integration neuer Datenquellen. - Cloud-native Architektur unterstützt Echtzeit- und Batch-Datenströme.
FA-2: Datenklassifizierung und -sicherung	Daten müssen basierend auf Sichtbarkeitsstufen (öffentlich, privat, anonymisiert) klassifiziert und zugänglich gemacht werden.	- Implementierung einer Klassifizierung im Service-Layer. - Anonymisierungs- und Pseudonymisierungsverfahren zur Sicherstellung des Datenschutzes. - Rollenbasierte Zugriffskontrolle regelt den Zugang zu sensiblen Daten.
FA-3: Datenzugriffskontrolle	Zugriffskontrollmechanismen wie RBAC oder ABAC müssen implementiert werden, um Berechtigungen zu verwalten.	- Einsatz von RBAC für einfache Rollenverwaltung. - Erweiterung durch ABAC für kontextbasierte Zugriffskontrolle. - Sicherstellung von Zugriffstransparenz durch Protokollierung und Auditing.
FA-4: Verwaltung von Datenquellen	Administratoren müssen Datenquellen hinzufügen, verwalten und Sichtbarkeitsstufen zuweisen können.	- Dashboard bietet intuitive Oberfläche zur Verwaltung von Datenquellen. - APIs unterstützen automatisierte Änderungen und Verwaltung. - Sichtbarkeitsstufen werden durch Metadatenmanagement unterstützt.
FA-5: Analyse- und Abfragefunktionen	Der Data Hub soll grundlegende Abfrage- und Analysefunktionen bieten.	- Unterstützung von SQL-ähnlichen Abfragen (z. B. durch BigQuery). - Echtzeit-Analysen durch Streaming-Technologien wie Kafka. - Visualisierungen und Filterfunktionen im Dashboard integriert.

Table 4.3: Funktionale Anforderungen mit Bezug auf das abstrakte Konzept

Nicht-funktionale Anforderungen (NFA)

Anforderung	Beschreibung	Umsetzung
NFA-1: Skalierbarkeit	Der Data Hub muss mit wachsenden Datenmengen und Nutzerzahlen skalieren können.	- Nutzung von Cloud-Diensten wie AWS oder GCP für horizontale Skalierung. - Containerisierung und Orchestrierung mit Docker und Kubernetes. - Verteilte Speichersysteme (z. B. S3, Cassandra) unterstützen die Skalierung.
NFA-2: Sicherheit	Der Data Hub muss die Daten vor unbefugtem Zugriff schützen und Datenschutzstandards (z. B. DSGVO, HIPAA) einhalten.	- Verschlüsselung der Daten im Ruhezustand und während der Übertragung. - Multi-Faktor-Authentifizierung (MFA) für Benutzerzugriffe. - Regelmässige Sicherheitsprüfungen und Audits.
NFA-3: Flexibilität und Erweiterbarkeit	Das System muss modular aufgebaut sein, um neue Anforderungen und Datenquellen integrieren zu können.	- Modularität ermöglicht die Erweiterung einzelner Komponenten. - Standardisierte Schnittstellen wie REST-APIs fördern die einfache Integration neuer Anwendungen. - Erweiterbare Rollen- und Zugriffskontrollmechanismen.
NFA-4: Benutzerfreundlichkeit	Die Benutzeroberfläche muss intuitiv sein, um die Bedienung für Administratoren und Endnutzer zu erleichtern.	- Nutzerzentriertes Dashboard-Design mit klarer Navigation. - Mehrsprachigkeit und barrierefreie Standards für Endnutzer. - Bereitstellung von Tutorials und Benutzerhilfen.
NFA-5: Zuverlässigkeit und Verfügbarkeit	Der Data Hub muss hochverfügbar sein und Datenverluste vermeiden.	- Redundante Speicherlösungen und regelmässige Backups sichern Dateninteg rität. - Verwendung von Load-Balancing zur Vermeidung von Engpässen. - Automatische Fehlerkennung und -behebung.
NFA-6: Performance	Der Data Hub muss geringe Latenz bei Datenanfragen gewährleisten, insbesondere bei Echtzeit-Datenquellen.	- Caching-Mechanismen zur Beschleunigung häufiger Anfragen. - Parallelisierung von Abfragen durch verteilte Architekturen. - Unterstützung von Streaming-Technologien für Echtzeitverarbeitung.

Table 4.4: Nicht-funktionale Anforderungen mit Bezug auf das abstrakte Konzept

Risiken und Herausforderungen

Die Implementierung und der Betrieb eines Data Hubs bringen neben den vielen Vorteilen auch potenzielle Risiken und Herausforderungen mit sich. Diese müssen frühzeitig identifiziert und adressiert werden, um die Funktionalität, Sicherheit und Skalierbarkeit des Systems langfristig sicherzustellen. In diesem Kapitel werden die zentralen Risiken und Herausforderungen beschrieben, zusammen mit Vorschlägen für deren Bewältigung.

Die Integration unterschiedlicher Datenquellen ist oft komplex und fehleranfällig. Standardisierte Schnittstellen wie REST-APIs und ETL-Tools erleichtern die Harmonisierung der Daten. Leistungsprobleme bei grossen Datenmengen können durch skalierbare Cloud-Technologien, Caching-Mechanismen und verteilte Datenbanken reduziert werden. Containerisierung und Load-Balancer sichern zudem die Skalierbarkeit und Verfügbarkeit des Systems.

Unzureichende Zugriffskontrollen und Sicherheitsverletzungen stellen Datenschutzrisiken dar. RBAC/ABAC, Datenverschlüsselung und Multi-Faktor-Authentifizierung helfen, sensible Daten zu schützen. Die Einhaltung von Datenschutzrichtlinien wie DSGVO kann durch Anonymisierungsstandards und Zusammenarbeit mit Datenschutzbeauftragten gewährleistet werden.

Benutzerakzeptanz ist ein weiterer wichtiger Punkt. Ein intuitives Dashboard-Design sowie Schulungen fördern die Nutzung. Die Wartung modularer Systeme kann durch Open-Source-Technologien und regelmässige Architektur-Reviews effizient gestaltet werden.

Strategische Risiken wie technologische Abhängigkeiten und schlechte Datenqualität lassen sich durch offene Standards, automatisierte Datenvalidierung und regelmässige Überprüfungen minimieren. So wird ein Data Hub nachhaltig und zukunftssicher.

Fazit zur Konzeptlösung des Data-Hubs

Die erfolgreiche Implementierung eines Data Hubs hängt entscheidend davon ab, wie gut die identifizierten Risiken und Herausforderungen bewältigt werden. Durch den Einsatz moderner Technologien, die Einhaltung bewährter Sicherheits- und Datenschutzstandards sowie die kontinuierliche Überwachung und Anpassung des Systems können diese Risiken effektiv minimiert werden. Ein strategisches Vorgehen gewährleistet die langfristige Funktionalität und Akzeptanz des Data Hubs. Jedoch hat nicht jedes Data Hub dieselben Anforderungen und Ziele, was bedeutet, dass es keine universelle Lösung für den Aufbau gibt. Jede modulare Komponente eines Data Hubs muss spezifisch an die individuellen Anforderungen angepasst werden. Hierin liegt zugleich die Stärke und das Risiko: Die Modularität erlaubt eine hohe Flexibilität und Anpassungsfähigkeit, bringt jedoch auch die Herausforderung mit sich, dass jede Komponente exakt auf die jeweiligen Anforderungen abgestimmt sein muss. Änderungen im System erfordern Anpassungen oder sogar den Austausch einzelner Module, was ein komplexes und sorgfältiges Management voraussetzt. Die grosse Stärke der Modularität ist somit auch ihre grösste Herausforderung, da jedes Data Hub eine einzigartige, massgeschneiderte Lösung benötigt, die stets den spezifischen Anforderungen gerecht wird.

5 Konzept der Visualisierung im Data-Hub

5.1 Hintergrund und Motivation

Wie in diesem Artikel [11] beschrieben, ist es wichtig in unserer heutigen Zeit, in der Daten eine immer grössere Rolle spielen, Informationen verständlich darzustellen. Gute Visualisierungen helfen dabei, Entscheidungen zu treffen und komplexe Themen besser zu verstehen. Besonders in einem Data-Hub sind Visualisierungen nicht nur ein Extra, sondern ein wichtiger Bestandteil. Sie sorgen dafür, dass die gesammelten und verarbeiteten Daten ihren vollen Nutzen entfalten können. Eine durchdachte Visualisierungsstrategie im Data-Hub hilft dabei, grosse Datenmengen einfach und schnell verständlich zu machen. Da moderne Unternehmen immer mehr Daten sammeln und diese immer komplexer werden, ist es wichtig, Muster und Auffälligkeiten auf den ersten Blick erkennen zu können. Gute Visualisierungen zeigen Zusammenhänge, die in Tabellen oder reinem Text oft übersehen werden können. Zudem sorgen klare Visualisierungen dafür, dass Daten für alle Mitarbeitenden zugänglich werden. Sie ermöglichen es auch Personen ohne tiefes technisches Wissen, datenbasierte Entscheidungen zu treffen. Das unterstützt das Ziel eines Data-Hubs: Daten verständlich und für jeden nutzbar zu machen. Ein gut gestaltetes Dashboard dient dabei als zentrales Werkzeug, um diese Visualisierungen effektiv bereitzustellen und den Nutzern eine intuitive Plattform zu bieten, über die sie die aufbereiteten Daten schnell und einfach analysieren und nutzen können.

5.2 Anforderungen an die Visualisierung

Die Entwicklung eines Visualisierungskonzepts für einen Data-Hub erfordert die Berücksichtigung verschiedener Anforderungen, die sowohl technischer als auch nutzerzentrierter Natur sind:

Funktionale Anforderungen

Damit Visualisierungen in einem Data-Hub effektiv sind, müssen sie verschiedene Aufgaben erfüllen:

- **Datenvielfalt:** Das System muss unterschiedliche Datentypen und -formate verarbeiten und darstellen können, einschliesslich strukturierter und unstrukturierter Daten.
- **Skalierbarkeit:** Visualisierungen sollen auch bei grossen Datenmengen reibungslos funktionieren und ohne Verzögerungen dargestellt werden.
- **Interaktivität:** Benutzer sollten Filter anwenden, Details erkunden und mit den Visualisierungen interagieren können.
- **Anpassbarkeit:** Benutzer sollten ihre Dashboards anpassen können, um sie an individuelle Bedürfnisse anzupassen.

Nicht-funktionale Anforderungen

Neben den funktionalen Anforderungen sind weitere Aspekte wichtig, die die Benutzerfreundlichkeit und Performance beeinflussen:

- **Benutzerfreundlichkeit:** Visualisierungen sollten intuitiv und leicht verständlich sein, damit auch Nutzer ohne technisches Fachwissen davon profitieren können.
- **Konsistenz:** Einheitliche Designs und Interaktionsmuster verbessern die Navigation und sorgen für ein besseres Nutzungserlebnis.
- **Barrierefreiheit:** Die Gestaltung sollte barrierefrei sein, sodass möglichst viele Menschen Zugang haben.
- **Performance:** Schnelle Ladezeiten und eine stabile Performance sind entscheidend, um die Nutzung reibungslos zu gestalten.

5.3 Lösungsansätze und Gestaltungsprinzipien

Ein effektives Dashboard zeichnet sich durch eine klare und übersichtliche Struktur aus, die den Nutzern hilft, wichtige Informationen schnell zu erfassen und zu analysieren. Das folgende Beispiel von diesem Artikel [12] zeigt ein einfach gestaltetes Dashboard, das verschiedene Visualisierungselemente kombiniert, um unterschiedliche Datentypen und -muster darzustellen. Die Bereiche (Area 1–4) illustrieren, wie Balkendiagramme, Liniendiagramme, Tabellen und andere grafische Darstellungen gezielt eingesetzt werden können, um komplexe Daten verständlich aufzubereiten. Diese Vielfalt an Visualisierungen ermöglicht es, sowohl Vergleiche als auch Trends und Detailanalysen auf einen Blick zu präsentieren.

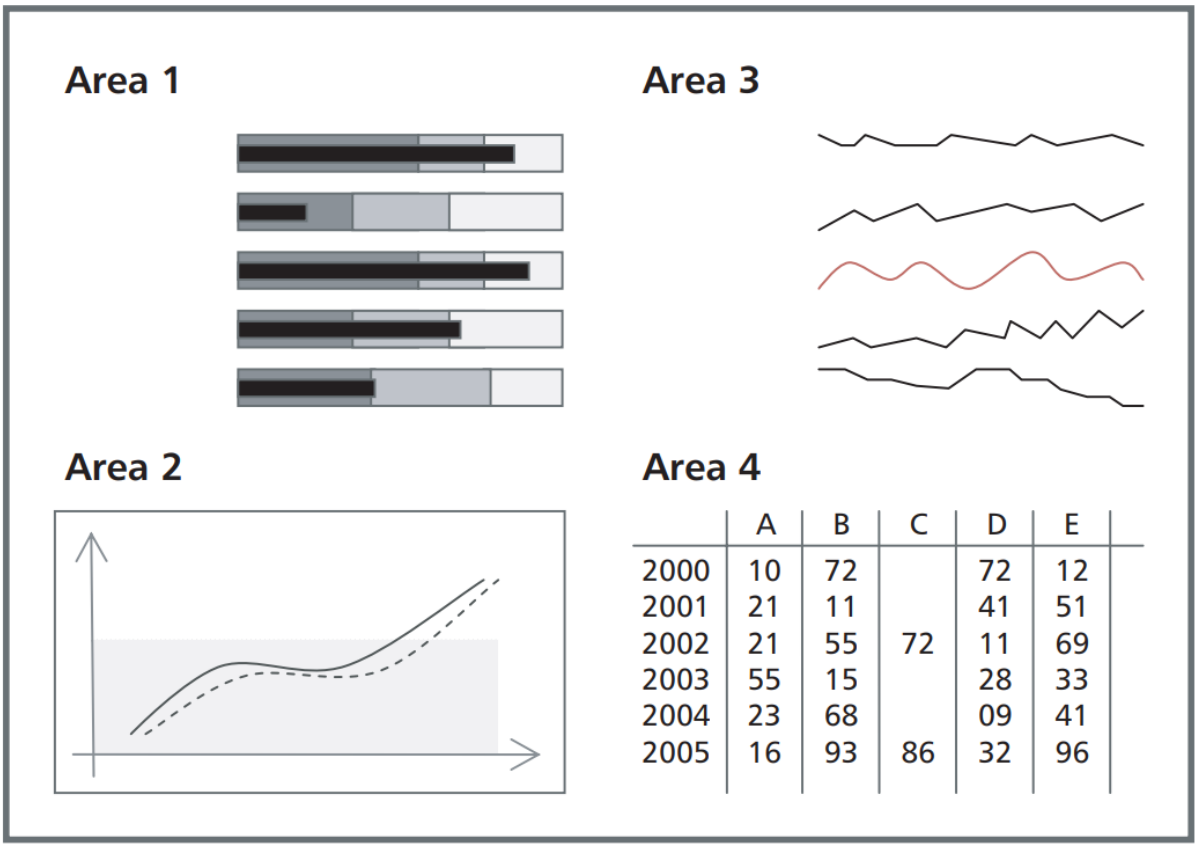


Figure 5.1: Beispiel eines simplen Dashboards

Laut diesem Artikel [12] macht ein gutes Dashboard wichtige Informationen einfach sichtbar und leicht verständlich. Damit es effektiv ist, sollten die Daten sorgfältig ausgewählt werden. Nur relevante Informationen gehören ins Dashboard, damit Benutzer nicht abgelenkt werden. Zu viele unnötige Daten können den Zweck verfehlen und die Übersichtlichkeit beeinträchtigen.

Die Darstellung der Daten ist genauso wichtig wie die Auswahl. Die passenden Visualisierungen – wie Diagramme, Tabellen oder andere Grafiken – helfen, Muster und Zusammenhänge klar erkennbar zu machen. Dabei sollte jede Visualisierung so gewählt sein, dass sie den dargestellten Daten entspricht. Zum Beispiel eignen sich Balkendiagramme gut, um Vergleiche zu zeigen, während Liniendiagramme Trends darstellen.

Bei der Gestaltung eines Dashboards ist es wichtig, dass es einfach und übersichtlich bleibt. Überflüssige Elemente sollten vermieden werden. Einheitliche Farben und Schriftarten erleichtern die Navigation und verbessern die Benutzererfahrung. Zudem sollten Dashboards interaktiv sein. Filter oder die Möglichkeit, Details auf Anfrage anzuzeigen, machen das Dashboard flexibler und nützlicher.

Eine weitere wichtige Eigenschaft ist die Anpassbarkeit. Benutzer sollten in der Lage sein, ihre Dashboards nach ihren Bedürfnissen zu gestalten. Zum Beispiel könnten sie entscheiden, welche Informationen sie sehen möchten und wie diese dargestellt werden.

Die Gestaltung eines Dashboards ist ein laufender Prozess. Es sollte regelmässig überprüft und angepasst werden, um sicherzustellen, dass es weiterhin die Bedürfnisse der Benutzer erfüllt. Dabei hilft es, direktes Feedback von den Benutzern einzuholen und bei Bedarf Änderungen vorzunehmen. So bleibt das Dashboard relevant und hilfreich.

Verschiedene Visualisierungsmöglichkeiten in einem Data-Hub Dashboard

Im Kontext eines Data-Hub-Dashboards spielen Visualisierungen eine entscheidende Rolle, um grosse Datenmengen verständlich aufzubereiten und Nutzern eine schnelle Entscheidungsfindung zu ermöglichen. Unterschiedliche Diagrammtypen können je nach Anwendungsfall spezifische Einsichten liefern. Im Folgenden werden die gängigsten Diagrammtypen vorgestellt, ihre Vor- und Nachteile erläutert und ihr Nutzen in einem Data-Hub beschrieben. Die folgenden Informationen und Bilder stammen von folgender Quelle: [13]

Balkendiagramm

Balkendiagramme eignen sich hervorragend, um kategorische Daten zu vergleichen, wie z. B. die Anzahl von Datenquellen oder die Datenvolumen pro System. Sie helfen Nutzern, Unterschiede zwischen Kategorien schnell zu erfassen.

Einsatz im Data-Hub: Balkendiagramme können z. B. verwendet werden, um die Datenintegrationsleistung verschiedener Datenquellen zu analysieren oder die monatliche Anzahl an hochgeladenen Datensätzen nach Quelle darzustellen.

Vorteile:

- Einfache und klare Darstellung von Unterschieden zwischen Kategorien.
- Leicht verständlich und schnell interpretierbar.

Nachteile:

- Bei zu vielen Kategorien kann das Diagramm unübersichtlich werden.
- Nicht geeignet für die Darstellung von zeitlichen Verläufen.

Beispielbild:

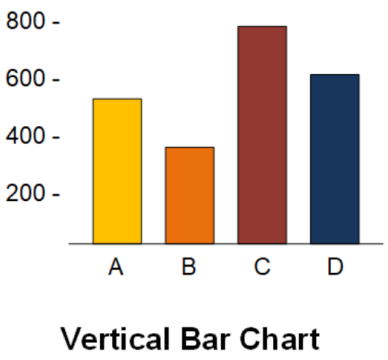


Figure 5.2: Beispiel für ein Balkendiagramm: Anzahl der Datenquellen pro Monat.

Liniendiagramm

Liniendiagramme sind ideal zur Darstellung von Trends und zeitlichen Entwicklungen, wie z. B. die Datenverarbeitungsgeschwindigkeit oder den Speicherverbrauch über die Zeit.

Einsatz im Data-Hub: Sie können verwendet werden, um den Anstieg von Echtzeit-Datenströmen oder die Nutzung von Speicherressourcen über verschiedene Zeiträume zu visualisieren.

Vorteile:

- Hervorragend zur Darstellung von Entwicklungen und Trends.
- Gut geeignet für kontinuierliche Daten.

Nachteile:

- Weniger effektiv bei der Darstellung von Kategorien.
- Kann bei zu vielen Linien unübersichtlich werden.

Beispielbild:

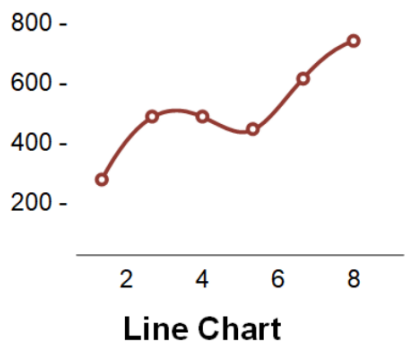


Figure 5.3: Beispiel für ein Liniendiagramm: Entwicklung der Datenübertragungsraten im Data-Hub.

Kreisdiagramm

Kreisdiagramme eignen sich besonders, um Anteile eines Ganzen darzustellen, wie z. B. die prozentuale Verteilung der Datennutzung zwischen verschiedenen Abteilungen.

Einsatz im Data-Hub: Sie sind hilfreich, um zu zeigen, welche Abteilung oder welcher Prozess den grössten Anteil an der Datennutzung hat, oder wie sich die Sichtbarkeitsstufen (öffentlich, privat, anonymisiert) verteilen.

Vorteile:

- Veranschaulicht Anteile und Proportionen auf einen Blick.
- Gut geeignet für wenige Kategorien.

Nachteile:

- Bei vielen Kategorien schwer lesbar.
- Unterschiede zwischen ähnlichen Anteilen sind schwer zu erkennen.

Beispielbild:

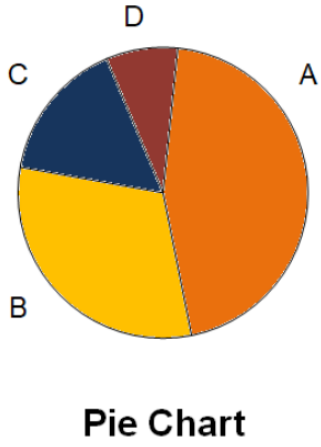


Figure 5.4: Beispiel für ein Kreisdiagramm: Verteilung der Datenquellen nach Abteilungen.

Flächendiagramm

Flächendiagramme sind ideal, um kumulative Werte über die Zeit darzustellen, z. B. die Entwicklung des genutzten Datenvolumens.

Einsatz im Data-Hub: Sie können genutzt werden, um die Gesamtzahl von Datenströmen oder Datensätzen zu verschiedenen Zeitpunkten anzuzeigen und Trends zu verdeutlichen.

Vorteile:

- Zeigt kumulative Werte über die Zeit.
- Gut geeignet, um das Verhältnis zwischen Teilen und dem Ganzen darzustellen.

Nachteile:

- Kann bei mehreren Kategorien unübersichtlich werden.
- Farbwahl ist entscheidend für die Lesbarkeit.

Beispielbild:

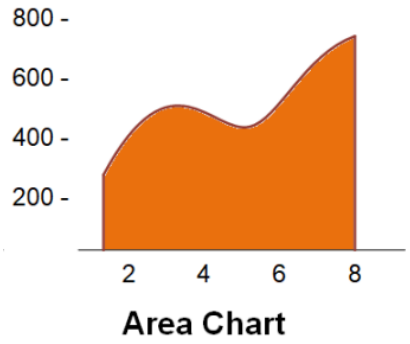


Figure 5.5: Beispiel für ein Flächendiagramm: Kumulierte Datenströme über die Zeit.

Fazit: Die Wahl des richtigen Diagrammtyps in einem Data-Hub-Dashboard hängt stark von den darzustellenden Daten und den Bedürfnissen der Nutzer ab. Eine durchdachte Auswahl der Visualisierungen ermöglicht es, Daten effizient zu analysieren und fundierte Entscheidungen zu treffen.

Innovative Ansätze

Neue Technologien bieten zusätzliche Möglichkeiten zur Datenvisualisierung:

- **KI-gestützte Dashboards:** Diese heben automatisch Muster und Anomalien hervor und bieten tiefere Einblicke.
- **Immersive Technologien:** Virtual Reality (VR) und Augmented Reality (AR) ermöglichen die Darstellung komplexer Daten in drei Dimensionen.
- **Kollaborative Dashboards:** Mehrere Nutzer können gleichzeitig an Visualisierungen arbeiten und Erkenntnisse in Echtzeit austauschen.

5.4 Technische Umsetzung

Die technische Umsetzung eines Visualisierungskonzepts erfordert die Auswahl moderner Technologien:

- **Frontend-Frameworks:** React oder Angular bieten eine gute Grundlage für interaktive Benutzeroberflächen.
- **Visualisierungsbibliotheken:** D3.js oder Plotly.js eignen sich für datengesteuerte Visualisierungen.
- **WebGL-Lösungen:** Three.js kann für komplexe 3D-Darstellungen verwendet werden.
- **Daten-Pipelines:** Effiziente Datenaufbereitung durch ETL-Prozesse (Extract, Transform, Load) ist essenziell.
- **Streaming-Technologien:** Tools wie Apache Kafka helfen bei der Verarbeitung von Echtzeitdaten.

Mit diesen Technologien kann der Data-Hub sowohl leistungsstark als auch benutzerfreundlich gestaltet werden. Dies stellt sicher, dass die Visualisierungen flexibel und effizient sind.

6 Konzept der Datensicherheit

6.1 Einleitung

Hintergrund und Motivation

Artha ist ein Unternehmen, das sich auf die Live-Überwachung und Analyse von Wasserdaten spezialisiert hat. Die gesammelten Daten werden auf eine Cloud hochgeladen und über ein Dashboard bereitgestellt, damit Kunden diese in Echtzeit einsehen und analysieren können. Aktuell betreibt Artha zwei Anlagen: eine am Flughafen Zürich und eine in einer Kläranlage. Die derzeitige Datenverwaltung weist jedoch Schwächen auf:

- **Zugriffskontrolle** – Es fehlt eine Zugriffskontrolle, wodurch Kunden die Daten anderer Kunden einsehen können.
- **Anonymisierung** – Die aktuellen Mechanismen bieten keine Möglichkeit, Daten anonymisiert für Forschung oder öffentliche Zwecke bereitzustellen, ohne sensible Informationen preiszugeben.

Diese Mängel stellen sowohl ein Sicherheitsrisiko als auch eine Einschränkung für den flexiblen Einsatz der Daten dar. Um den steigenden Anforderungen gerecht zu werden, soll ein System entwickelt werden, das:

- den Zugriff auf sensible Daten strikt kontrolliert, um unautorisierten Zugriff zu verhindern
- die Bereitstellung anonymisierter Daten für Forschungs- und öffentliche Zwecke ermöglicht
- flexibel bleibt, um zukünftige Erweiterungen (z. B. neue Anlagen oder Nutzergruppen) zu unterstützen.

Dieses Konzept ist entscheidend um den Anforderungen von der Industrie und der Forschung gerecht zu werden.

Zielsetzung

Das Konzept verfolgt zwei Hauptziele:

1. Sicherstellung, dass sensible Daten vor unautorisiertem Zugriff geschützt sind und nur autorisierten Nutzern zugänglich gemacht werden.
2. Möglichkeit zur Bereitstellung anonymisierter Daten für die ÖMentlichkeit, ohne die Privatsphäre oder Vertraulichkeit der Originaldaten zu gefährden.

Das Konzept soll gewährleisten, dass die Datenablage sowohl sicher als auch flexibel bleibt, damit die Daten von der Industrie, wie auch für die Forschung genutzt werden können.

Um die definierten Ziele zu erreichen, wurden spezifische Anforderungen an das System formuliert. Diese Anforderungen sind in funktionale und nicht-funktionale Kategorien unterteilt. Während funktionale Anforderungen beschreiben, welche konkreten Funktionen das System erfüllen muss, spezifizieren die nicht-funktionalen Anforderungen Qualitätsmerkmale wie Sicherheit, Skalierbarkeit und Benutzerfreundlichkeit. Diese Anforderungen bilden die Grundlage für die Entwicklung und Bewertung der technischen Lösung.

6.2 Anforderungen

Die hier beschriebenen Anforderungen definieren die grundlegenden Ziele und Eigenschaften des Systems auf einer abstrakten Ebene. Technische Details zur Umsetzung werden in der Konzeptlösung spezifiziert.

Funktionale Anforderungen

Nr.	Anforderung	Beschreibung
DFA-1	Die Datenablage muss die Vertraulichkeit sensibler Informationen gewährleisten.	Die Sichtbarkeit von Daten muss gewährleistet werden, damit nicht autorisierte Nutzer private Daten nicht sehen. Öffentliche oder anonymisierte Daten können weiterhin eingesehen werden.
DFA-2	Die Sicherheitsmechanismen müssen anpassbar sein, um neue Anforderungen wie zusätzliche Nutzer oder veränderte Datenschutzvorgaben zu berücksichtigen.	Neue Nutzer sollen hinzugefügt werden können, und bestehende Nutzer anpassbar sein, sowie neue Daten sollen angepasst werden.
DFA-3	Mehrere Nutzer können dieselben Daten sehen.	Es muss gewährleistet werden, dass eine n:m Beziehung möglich ist. Damit ist gemeint, dass mehrere Nutzer dieselben privaten Daten einsehen können und ebenfalls ein Nutzer mehrere Anlagen einsehen kann.
DFA-4	Daten können jeweils als öffentlich, privat oder anonymisiert öffentlich gehandhabt werden.	Öffentlich bedeutet, jeder kann diese sehen. Privat bedeutet, dass nur berechtigte Nutzer die Daten sehen können. Anonymisiert bedeutet, jeder kann die Daten sehen, aber sie sind anonymisiert (Daten wie Ursprung sind verborgen) und Nutzer mit entsprechender Berechtigung können die Daten komplett einsehen.

Table 6.1: Funktionale Anforderungen und Beschreibungen

Nicht-funktionale Anforderungen

Nr.	Anforderung	Beschreibung
NDFA-1	Sicherstellen, dass autorisierte Nutzer jederzeit Zugriff haben.	Die Datenablage muss sicherstellen, dass autorisierte Nutzer jederzeit Zugriff auf ihre berechtigten Daten haben, ohne Unterbrechungen durch Sicherheitsmassnahmen.
NDFA-2	Effizienz bei wachsender Nutzeranzahl und Datenvolumen.	Die Sicherheitsmassnahmen müssen auch bei wachsender Anzahl von Nutzern und Daten weiterhin zuverlässig und effizient funktionieren.
NDFA-3	Benutzerfreundlichkeit darf nicht beeinträchtigt werden.	Die Implementierung von Sicherheitsmassnahmen darf die Nutzererfahrung nicht beeinträchtigen und muss intuitiv bedienbar sein.

Table 6.2: Nicht-funktionale Anforderungen

6.3 Stakeholder

Nutzergruppen

Das System richtet sich an verschiedene Stakeholder-Gruppen, die jeweils spezifische Anforderungen und Erwartungen an die Datenverwaltung haben. Zu den Stakeholdern gehören Datenerfasser, Datenanalysten, Fachexperten, Administratoren, die Öffentlichkeit und Kunden. In der Praxis können bei Artha einzelne Personen mehrere Rollen gleichzeitig einnehmen, z. B. als Datenanalyst, Administrator, Datenerfasser und Fachexperte. Die folgende Tabelle beschreibt die Bedürfnisse und Ziele der einzelnen Stakeholder und zeigt auf, wie das System diese unterstützen kann.

Stakeholder-Analyse

Stakeholder	Beschreibung	Bedürfnisse und Erwartungen	Wie das System hilft
Datenerfasser	Personen, die Daten sammeln und hochladen (z. B. Techniker vor Ort).	- Einfache Erfassung und Upload der Sensordaten. - Möglichkeit, Sichtbarkeitsstufen für Daten festzulegen.	- Bietet eine intuitive Oberfläche für Datenuploads. - Ermöglicht einfache Sichtbarkeitssteuerung (öffentlich, privat, anonymisiert).
Datenanalysten	Experten, die mit den Daten Analysen und Berichte erstellen (z. B. interne Analysten, Forscher).	- Zugriff auf anonymisierte Daten für Analysen. - Möglichkeit, relevante Datensätze einfach zu filtern.	- Liefert anonymisierte Daten nach Bedarf. - Ermöglicht schnelles und gezieltes Arbeiten durch Filteroptionen.
Fachexperten	Verantwortliche, die Anomalien überwachen und Entscheidungen treffen (z. B. Qualitätsmanager).	- Benachrichtigung bei Ausreissern oder Problemen. - Möglichkeit, Nutzerrollen und Berechtigungen zu verwalten.	- Automatisches Alarmsystem bei Anomalien. - Benutzerfreundliche Verwaltungsoberfläche für Zugriffsberechtigungen.
Administratoren	Personen, die das System verwalten (z. B. IT-Administratoren bei Artha).	- Rollen- und Rechteverwaltung. - Sicherstellung der Datensicherheit und Compliance.	- Flexible und skalierbare Verwaltung von Nutzern und Berechtigungen. - Unterstützung durch Sicherheitsmechanismen (z. B. Audit-Logs).
Öffentlichkeit	Personen oder Institutionen, die anonymisierte Daten einsehen (z. B. Forscher).	- Zugriff auf relevante Informationen ohne Kompromittierung der Privatsphäre.	- Stellt anonymisierte Daten bereit, die keine Rückschlüsse auf Nutzer oder Orte erlauben.
Kunden	Auftraggeber von Artha (z. B. Betreiber von Anlagen).	- Gewährleistung der Datensicherheit für ihre Anlagen. - Einhaltung vertraglicher und rechtlicher Vorgaben.	- Sichere Datenbereitstellung durch Zugriffskontrolle und Verschlüsselung. - Übersicht über Zugriffs- und Nutzungsmuster.

Table 6.3: Stakeholder, ihre Bedürfnisse und wie das System unterstützt

6.4 Lösungsansatz

In diesem Kapitel wird ein Lösungsansatz für die Datenverwaltung beschrieben, der für eine allgemeine Datenablage gültig ist.

Eine Datenablage in einem modernen System besteht aus mehreren zentralen Komponenten, die zusammenarbeiten, um eine effiziente, sichere und skalierbare Verwaltung von Informationen zu ermöglichen. Diese Komponenten sind eng miteinander verzahnt, um Anforderungen wie Datensicherheit, Zugriffskontrolle und Anonymisierung zu erfüllen. Der folgende Ansatz beschreibt diese Komponenten und deren Zusammenspiel auf einer abstrakten Ebene.

Datenquellen

- Die Daten werden von verschiedenen Sensoren und Geräten erfasst, die Rohdaten in das System einspeisen.
- **Beispiele:** Wassersensoren, Überwachungsgeräte.

Datenbank

- Eine zentrale Ablage, in der alle Roh- und/oder verarbeiteten Daten sicher gespeichert werden.
- Klassifikation der Daten in Kategorien: privat, öffentlich, anonymisiert.
- Unter bestimmten Umständen kann hier auch eine grundlegende Datenverwaltung integriert sein.

Service-Layer

- Der Service-Layer ist das Herzstück der Datenverwaltung. Er regelt den Zugriff auf die Daten basierend auf Benutzerrollen und Sichtbarkeitsregeln.
- **Aufgaben:**
  - Überprüfung, welche Daten für welche Nutzer sichtbar sind.
  - Durchführung von Anonymisierungen sensibler Informationen.

Frontend (Dashboard)

- Das Dashboard bietet den Nutzern eine visuelle Oberfläche, um Daten in Echtzeit einzusehen.
- Unterschiedliche Benutzergruppen sehen nur die Daten, die für sie freigegeben oder relevant sind.

Benutzerrollen und Zugriffskontrolle

- Die Rollen definieren, welche Aktionen ein Nutzer durchführen darf (z. B. Daten hochladen, ansehen, verwalten).
- Zugriffskontrollen sorgen dafür, dass nur autorisierte Nutzer Zugriff auf sensible oder private Daten erhalten.

Anonymisierungsschicht

- Daten, die öffentlich verfügbar gemacht werden sollen, durchlaufen eine Anonymisierung, bei der sensible Informationen entfernt oder maskiert werden.

6.5 Grafische Darstellung

Die folgende Grafik zeigt eine abstrahierte Darstellung des Datenflusses und der Komponenten des Systems:

- **Eingabe:** Sensordaten gelangen über eine Eingabeschnittstelle in die Datenbank. Hierfür werden sie zuerst validiert und entsprechende Metadaten werden hinzugefügt (wie Zeit, Ort, Kontext-Daten wie beispielsweise Wetter).
- **Datenbank:** Die Daten werden dort gespeichert, klassifiziert und bei Bedarf weitergeleitet. Periodisch oder auch manuell können hier Datenanalysen und Weiterverarbeitungen gestartet werden. Diese nehmen Daten aus der Datenbank verarbeiten sie und speichern sie erneut ab.
- **Service-Layer:** Dieser Layer filtert und anonymisiert die Daten basierend auf Benutzerrollen. Dieser Layer ist ebenfalls dafür zuständig, die Daten dem Frontend zur Verfügung zu stellen. Dementsprechend werden Daten über ein Kontroll-Service oder Layer überprüft und später aufbereitet und zurückgegeben
- **Frontend:** Nutzer greifen über das Dashboard auf die gefilterten oder anonymisierten Daten zu. Dies wurde in der Illustration entsprechend ausserhalb der Datenablage gezeichnet, damit verdeutlicht wird, dass das Frontend durch eine beliebige Website oder Schnittstellen-Abfrage ersetzt werden kann. Das Frontend beinhaltet selbst keine Logik und verändert auch keine Daten (ausser Benutzerdaten).

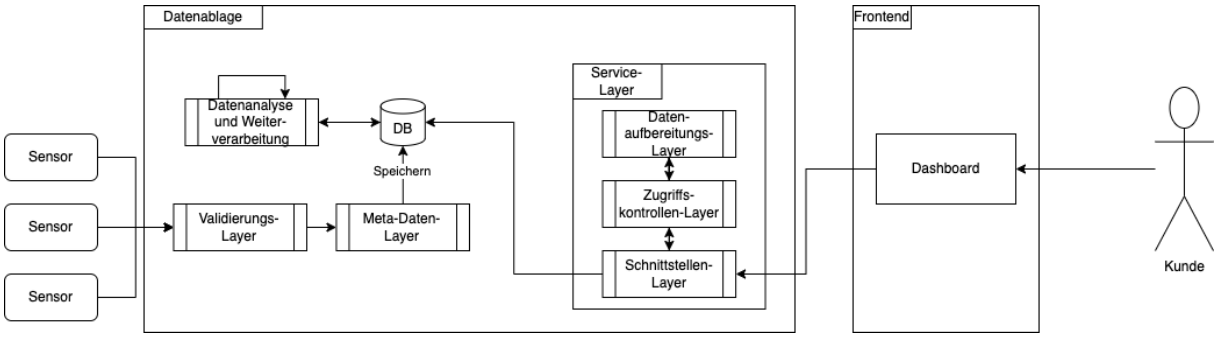


Figure 6.1: Konzept Datenablage

Beispiel-Zugriff

Die folgende Grafik stellt vereinfacht dar, wie eine Datenabfrage auf die Datenablage aussehen könnte, ohne komplexe Zwischentabellen und Datenstrukturen. Ersichtlich ist, dass Daten jeweils aus einer Anlage kommen und die Nutzer Berechtigungen auf bestimmte Anlagen haben.
Aus dem Ergebnis der Abfragen wird sichtbar, dass man die Daten vollständig sieht, für welche die Berechtigungen stimmen (oder die Daten öffentlich sind). Anonymisierte Daten verstecken sensible Informationen mit einer (zufälligen) Variable, damit bekannt bleibt welche Datensätze zusammengehören (Bspw. bleibt der Ort bei AG stets O1, in einer Abfrage. Bei einer zweiten Abfrage könnte es bspw. auch O2 sein).

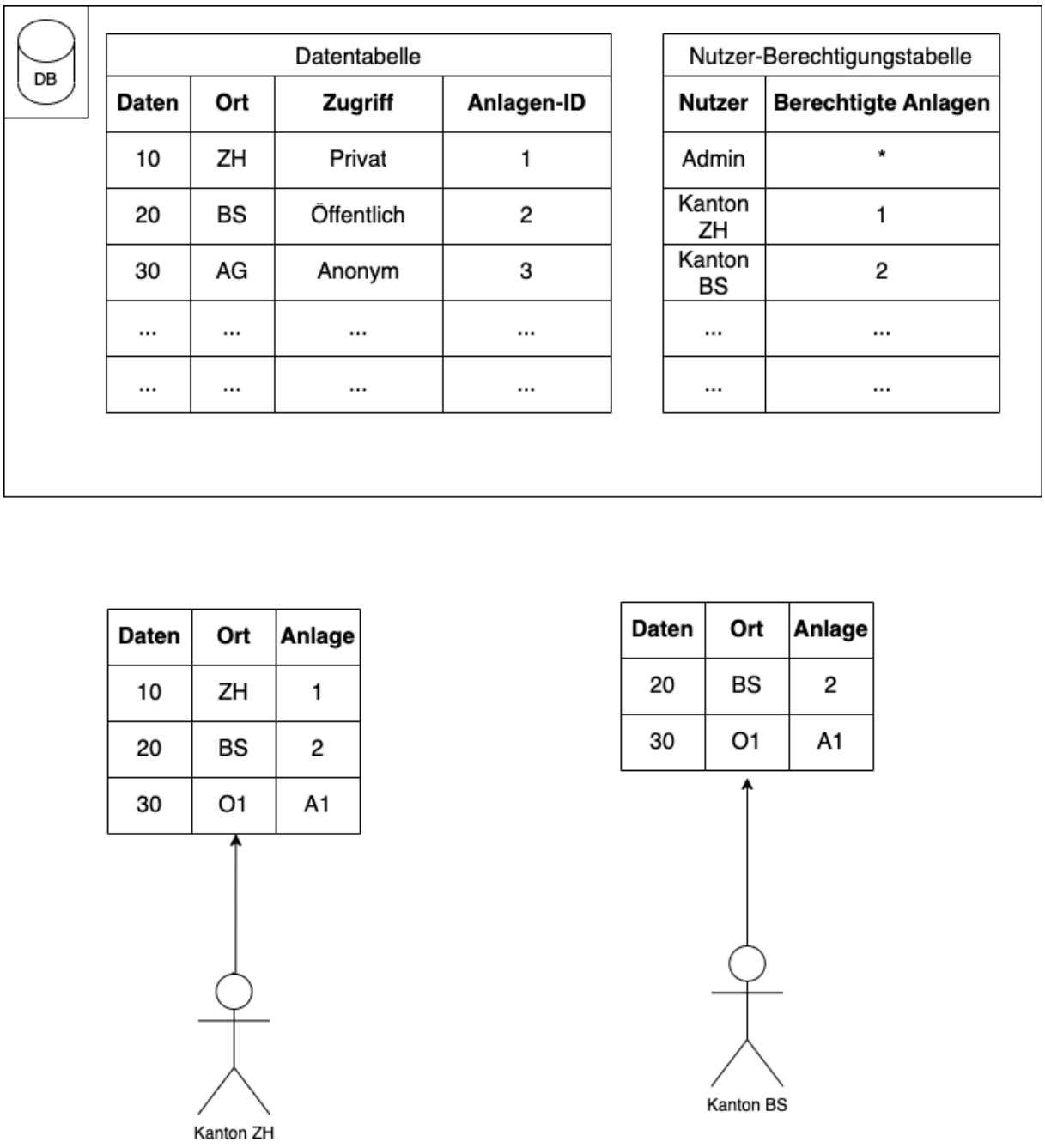


Figure 6.2: Datenverwaltungskonzept

Mit diesem Konzept können flexibel Daten verwaltet werden, da die Grundfunktionalität einer Datenablage gewährleistet wird, dadurch dass Daten öffentlich oder anonym zugänglich sind durch alle Nutzer. Durch das Spezifizieren von privaten Daten können Benutzer zusätzlich feingranularer entscheiden, wie mit ihren Daten umgegangen wird. Dadurch wird das Ziel einer Datenablage (viele Daten sammeln und bereit stellen für viele Nutzer) erfüllt.

7 Konzeptlösung der Datenverwaltung

Aufbauend auf den Anforderungen und Zielen des Konzepts werden in diesem Kapitel allgemeine, jedoch konkrete Lösungsansätze für die Datenverwaltung in Datahubs beschrieben. Diese Lösungen konzentrieren sich auf die effiziente und sichere Verwaltung von Daten, einschliesslich Zugriffssteuerung, Anonymisierung und Datenklassifizierung, und lassen sich auf verschiedene Architekturen anwenden.

7.1 Literaturrecherche zur Datenverwaltung

Eine fundierte Datenverwaltung in Datahubs erfordert eine sorgfältige Analyse bestehender Ansätze und Technologien. Die folgende Literaturrecherche beleuchtet bewährte Praktiken aus Forschung und Industrie, insbesondere in den Bereichen Zugriffssteuerung, Anonymisierung und Datenklassifizierung. Ziel ist es, auf dieser Grundlage Lösungsansätze zu entwickeln, die allgemeingültig und flexibel für verschiedene Szenarien einsetzbar sind. Dabei wird besonderer Wert daraufgelegt, wie diese Ansätze die im Konzept definierten Anforderungen erfüllen können.

Die Datenverwaltung in Data Hubs ist ein zentrales Element, das die Funktionalität und Sicherheit des gesamten Systems massgeblich beeinflusst. Um allgemeine und praxistaugliche Lösungsansätze zu entwickeln, wird eine umfassende Literaturrecherche durchgeführt, die sich in drei zentrale Themenbereiche gliedert:

- **Datenverwaltung (Software-Architektur):**
  Der Fokus liegt auf der optimalen Platzierung und Gestaltung der Logik innerhalb der Systemarchitektur. Ziel ist es, Ansätze für die Klassifizierung, Speicherung und Verarbeitung von Daten zu analysieren, die flexibel und skalierbar sind.
- **Datenzugriff:**
  In diesem Bereich werden Prinzipien für die effektive und sichere Handhabung von Daten untersucht. Hierbei liegt ein besonderer Schwerpunkt auf der Zugriffssteuerung und den Mechanismen, die gewährleisten, dass nur autorisierte Nutzer auf sensible oder geschützte Daten zugreifen können.
- **Login-Mechanismen und Datenverwaltung im Dashboard:**
  Dieser Bereich widmet sich der Benutzerinteraktion mit dem System. Dabei werden allgemeine Prinzipien für sichere, aber benutzerfreundliche Authentifizierungsprozesse sowie für die Gestaltung von Dashboards untersucht. Ziel ist es, Lösungsansätze für die intuitive Verwaltung von Nutzerrollen, Zugriffsrechten und Datenvisualisierungen zu entwickeln.

Diese drei Bereiche werden sowohl getrennt analysiert als auch im Zusammenspiel betrachtet, um ein ganzheitliches Bild moderner Datenverwaltungssysteme zu schaffen. Die gewonnenen Erkenntnisse dienen als Grundlage für die Entwicklung allgemeingültiger Lösungsansätze, die flexibel auf verschiedene Data Hub-Projekte angewendet werden können.

7.2 Software-Architektur der Datenverwaltung

Die zentrale Herausforderung der Datenverwaltung in Data Hubs besteht darin, sicherzustellen, dass Daten basierend auf Sichtbarkeitsstufen (privat, anonymisiert, öffentlich) gefiltert und bereitgestellt werden. Die Frage, wo diese Logik – insbesondere für das Filtern und Anonymisieren – implementiert wird, beeinflusst massgeblich die Flexibilität, Sicherheit und Performance des Systems. In diesem Abschnitt werden die Vor- und Nachteile verschiedener Implementierungsebenen untersucht, um allgemeingültige Empfehlungen abzuleiten.

Kernfragen der Analyse

1. Wo sollte die Datenfilterung und Anonymisierung implementiert werden: auf der Datenbank-, Service- oder Frontend-Ebene?

2. Welche Auswirkungen haben die Implementierungsorte auf Skalierbarkeit, Sicherheit und Wartbarkeit?

Analyse der Implementierungsebenen

Implementierung auf Datenbank-Ebene

Beschreibung: Die Filterung und Anonymisierung erfolgt direkt in der Datenbank. Die Datenbank führt Anfragen nur mit den entsprechenden Sichtbarkeitsregeln aus und liefert gefilterte Daten zurück.

Vorteile:

- **Effizienz:** Anfragen werden direkt in der Datenbank verarbeitet, wodurch Daten nur nach Bedarf übertragen werden.
- **Integrierte Sicherheitsfunktionen:** Viele moderne Datenbanken unterstützen rollenbasierte Zugriffskontrollen und Verschlüsselung.
- **Skalierbarkeit:** Datenbanken können grosse Datenmengen effizient verarbeiten.

Nachteile:

- **Komplexität:** Die Anonymisierungs- und Filterlogik wird in die Abfragesprache (z. B. SQL) integriert, was zu Wartungsproblemen führen kann. Es müssen evtl. komplexe Abfrage-Statements erstellt werden, welche Fehlerallfällig sein können oder auch die Performance beeinträchtigen.
- **Vendor Lock-in:** Die Lösung ist oft stark an die spezifischen Funktionen des Datenbankanbieters gebunden. Wechsel zu anderem Anbieter ist mit Aufwand verbunden.
- **Verteilung der Business-Logik:** Die Business-Logik wird auf Datenbankund Service-Ebene verteilt, was zu Seiteneffekten führen kann, sowie zu erschwertem Debugging, da über mehrere Schichten der Programmfluss geprüft werden muss.

Implementierung auf Service-Layer-Ebene

Beschreibung: Die Logik für Filterung und Anonymisierung wird in einer Zwischenschicht zwischen der Datenbank und dem Frontend umgesetzt. Der Service-Layer verarbeitet alle Datenanforderungen und sorgt dafür, dass nur berechtigte Daten an die Nutzer geliefert werden. Grundsätzlich besitzt der Service-Layer die Business-Logik.

Vorteile:

- **Flexibilität:** Die Logik ist unabhängig von der Datenbank und kann leicht angepasst werden.
- **Zentrale Steuerung:** Alle Regeln und Berechtigungen sind in einer Schicht gebündelt.
- **Unabhängigkeit:** Die Lösung ist nicht an einen spezifischen Datenbankanbieter gebunden.

Nachteile:

- **Leistungsanforderungen:** Der Service-Layer kann bei hohen Datenvolumina zum Flaschenhals werden, wenn nicht ausreichend skaliert wird.
- **Komplexität:** Die gesamte Logik muss selbst implementiert und gewartet werden.

Implementierung auf Frontend-Ebene

Beschreibung: Die Filterung und Anonymisierung erfolgt im Frontend. Alle Daten werden an den Client gesendet, und die Filterlogik wird dort ausgeführt.

Vorteile:

- **Einfachheit:** Keine Änderungen an der Datenbank oder dem Service-Layer erforderlich.
- **Direkte Kontrolle:** Nutzer können selbst Filteroptionen anpassen.

Nachteile:

- **Unsicherheit:** : Alle Daten, auch sensible, müssen an den Client übertragen werden, wodurch sie potenziell kompromittiert werden könnten. Durch einfaches Netzwerküberwachen im Browser können bereits ungefilterte und nicht anonymisierte Daten eingesehen werden.
- **Netzwerklast:** Grosse Datenmengen müssen vollständig übertragen werden, bevor sie gefiltert werden können
- **Unwartbarkeit:** Jedes neue Frontend erfordert eine erneute Implementierung der Logik.

Als Überblick dient folgende Tabelle:

	Datenbank	Service	Frontend
Aufwand	Mittel	Mittel	Gut
Flexibilität & Skalierbarkeit	Gut	Gut	Mittel
Abhängigkeit	Schlecht	Sehr gut	Sehr Schlecht
Performance	Sehr gut	Gut	Sehr Schlecht
Sicherheit	Gut	Gut	Sehr Schlecht

Figure 7.1: Überblick Implementierungsebenen

Hier ist auf einen Blick klar, dass der Service Grundsätzlich gut abschneidet. Die Datenbank hat den grossen Nachteil, dass es das Datahub sehr abhängig macht, beim Service ist dies nicht der Fall, da dieser Grundsätzlich nicht gewechselt werden muss (in den meisten Fällen). Es kommt viel öfter vor, dass Datenbanken gewechselt werden, aus Performance oder Preisgründen. Das Frontend schneidet auf allen Ebenen schlecht ab, ausser beim Aufwand, da dies relativ einfach umgesetzt werden kann. Folgende Erkenntnis spiegelt sich in der Forschung wider, wie im nächsten Kapitel klar wird.

Forschung aus der Industrie

Die Implementierung von Datenfilterung und Anonymisierung auf der Service-Ebene wird in der Industrie häufig bevorzugt, da sie Flexibilität und zentrale Steuerung bietet. Mehrere Studien und Artikel unterstützen diesen Ansatz:

Service-orientierte Ansätze für Datenanonymisierung

Der Artikel [14] im IEEE Xplore beschreibt eine neuartige Architektur, die auf einer serviceorientierten Basis arbeitet und die Echtzeit-Anonymisierung von Daten ermöglicht. Dies gewährleistet, dass autorisierte Benutzer auf die Daten zugreifen können, während Datenschutzstandards eingehalten werden.

Echtzeit-Anonymisierung sensibler persönlicher Daten mittels Service-Architektur:

Ein weiterer Beitrag [15] im IEEE Xplore schlägt eine servicebasierte Architektur für die Echtzeit-Anonymisierung vor, die sicherstellt, dass Daten für autorisierte Nutzer zugänglich bleiben und gleichzeitig die Privatsphäre gewahrt wird.

Anonymisierung im grossen Massstab: Vom Datensatz zur Pipeline:
Ein Artikel [16] von Privacy Analytics betont die Bedeutung einer robusten und praktischen Herangehensweise an die Datenanonymisierung, insbesondere in gross angelegten Datenpipelines, um den Datenschutz zu gewährleisten.

Datenanonymisierung: Schutz der Privatsphäre in gross angelegten Analysen:
Ein Beitrag [17] der Data Science Council of America untersucht, wie Datenanonymisierung es ermöglicht, datenschutzfreundliche Analysen durchzuführen, indem persönliche Identifikatoren entfernt und Anonymität in Datensätzen sichergestellt wird.

Architektur von Datahubs:
Die Architektur von Data Hubs wird in der Fachliteratur umfassend behandelt, insbesondere hinsichtlich der Implementierung von Datenfilterung und Anonymisierung auf der Service-Ebene. Ein herausragendes Beispiel hierfür ist die "Data Hub Guide for Architects" [18] von Progress Software. Dieses Dokument bietet einen tiefen Einblick in die Prinzipien der Data Hub-Architektur und betont die Bedeutung einer zentralen Schicht für Datenintegration und -verwaltung. Es wird erläutert, wie ein Data Hub als zentrale Vermittlungsstelle zwischen verschiedenen Datenquellen und -konsumenten fungiert und dabei Flexibilität, Skalierbarkeit und Sicherheit gewährleistet.

Diese Quellen unterstreichen die Vorteile der Implementierung von Datenfilterungs- und Anonymisierungslogik auf der Service-Ebene, insbesondere in Bezug auf Flexibilität, zentrale Steuerung und die Fähigkeit, Datenschutzstandards effektiv einzuhalten. Für die Recherche haben wir auf Google Scholar nach folgenden Begriffen gesucht:

- "Service-oriented data anonymization"
- "Real-time data anonymization service architecture"
- "Data anonymization pipelines in data engineering"
- "Privacy-preserving data analytics service layer"

Fazit der Entscheidung

Auf Grundlage der durchgeführten Analyse und der Literaturrecherche wird die Implementierung der Datenfilterung und Anonymisierung im Service-Layer als bevorzugte Lösung für die Datenverwaltung in Data Hubs empfohlen. Diese Entscheidung gewährleistet eine optimale Balance zwischen Flexibilität, Sicherheit und Skalierbarkeit und entspricht den Anforderungen moderner Data Hub-Architekturen.

7.3 Datenzugriff bei Datahubs

Der Datenzugriff ist eine der zentralen Herausforderungen bei der Verwaltung sensibler Daten in Data Hubs. Ziel ist es, Daten basierend auf ihrem Zugriffsstatus – öffentlich, privat oder anonymisiert – gezielt und sicher bereitzustellen. Dieses Kapitel beleuchtet die Standardpraktiken, die in der Industrie und insbesondere bei sensiblen Datenablagen wie im medizinischen Bereich etabliert sind. Die Analyse stützt sich auf Literaturrecherchen und zeigt, wie diese Prinzipien die Anforderungen moderner Data Hubs erfüllen.

Kernfragen der Analyse

1. Wie wird der Datenzugriff in modernen Data Hubs organisiert?
2. Welche Standards und Prinzipien sind bei sensiblen Daten, wie z. B. in der Medizin, etabliert?
3. Wie wird sichergestellt, dass die Sichtbarkeitsstufen (öffentlich, privat, anonymisiert) korrekt und effizient umgesetzt werden?

Prinzipien des Datenzugriffs

In Data Hubs wird der Datenzugriff häufig durch drei zentrale Sichtbarkeitsstufen organisiert:

Öffentlich:

- Daten, die als öffentlich markiert sind, können von jedem Nutzer ohne Einschränkung eingesehen werden.
- **Beispiele:** Aggregierte Daten zur Umweltüberwachung oder allgemeine Statistiken, die keine personenbezogenen Informationen enthalten.

Privat:

- Privat gekennzeichnete Daten sind nur für berechtigte Personen oder Gruppen zugänglich.
- **Beispiele:** Personenbezogene Gesundheitsdaten, die nur behandelnden Ärzten zugänglich sind, Wasserdaten zu Anlagen, welche nicht von anderen Anlagen gesehen werden sollten.

Anonymisiert:

- Anonymisierte Daten können von allen Nutzern eingesehen werden, allerdings ohne Rückschluss auf persönliche oder sensible Informationen.
- Berechtigte Personen können jedoch die vollständigen, nichtanonymisierten Daten einsehen.
- **Beispiele:** Forschungsdaten in der Medizin, bei denen die Identität der Patienten anonymisiert, die klinischen Details aber sichtbar sind. Bei Wasserdaten können Werte aus verschiedenen Kläranlagen ausgelesen werden, ohne dabei Daten herauszugeben, welche Rückschluss auf Ort der Kläranlage bieten.

Lösungsansätze und Literaturrecherche

Standards für Datenzugriffsmodelle in Data Hubs:

Studien zeigen, dass Datenzugriffsmodelle in Data Hubs stark auf **Role-Based Access Control (RBAC)** oder **Attribute-Based Access Control (ABAC)** basieren, um Zugriffsebenen klar zu definieren.
Laut einer Arbeit von Zhang et al. (2020) wird RBAC häufig in Data Hubs eingesetzt, um Zugriffskontrollrichtlinien zentral zu verwalten und die Sicherheit bei sensiblen Daten zu gewährleisten.
Im medizinischen Bereich betont eine Studie von Smith et al. (2019), dass ABAC besonders geeignet ist, um flexibel auf verschiedene Anforderungen einzugehen, z. B. den Zugriff basierend auf Standort, Zeit und Nutzerattributen zu steuern.

Datenzugriff und Anonymisierung im medizinischen Bereich:

Die Implementierung von Anonymisierungsmechanismen ist besonders in der Medizin von hoher Relevanz, um die Privatsphäre der Patienten zu schützen und gleichzeitig die Forschung zu ermöglichen.
Laut einer Arbeit von Kumar et al. (2021) wird die Kombination aus anonymisierten und privaten Daten immer häufiger angewendet, um Forscher und Ärzte gleichermassen zu unterstützen.
Eine Studie von Privacy Analytics (2020) hebt hervor, dass die Pseudonymisierung ein gängiger Standard ist, um Daten anonymisiert bereitzustellen, wobei autorisierte Nutzer weiterhin auf die ursprünglichen Daten zugreifen können.

Sicherheit und Effizienz in der Umsetzung:

Moderne Data Hubs nutzen Mechanismen wie Datenverschlüsselung und Protokollierung, um sicherzustellen, dass nur berechtigte Nutzer auf sensible Daten zugreifen können.
Laut einer Untersuchung von IEEE Xplore (2022) ist die Implementierung von Verschlüsselung sowohl auf Transport- als auch auf Speicherebene essenziell für den Datenschutz.
Khan et al. (2021) schlagen vor, dass Protokollierungsmechanismen unerlaubte Zugriffsversuche dokumentieren und Sicherheitsverletzungen frühzeitig erkennen können.

Role-Based Access Control (RBAC)

Definition:
RBAC ist ein Zugriffskontrollmodell, bei dem Berechtigungen basierend auf den Rollen eines Nutzers innerhalb des Systems zugewiesen werden. Eine Rolle repräsentiert eine Sammlung von Berechtigungen, die mit einer bestimmten Funktion oder Position in der Organisation verbunden sind (z. B. „Administrator“, „Analyst“, „Gast“).

Wie es funktioniert:

1. Rollen zuweisen: Jede Rolle wird vordefiniert und mit spezifischen Berechtigungen verknüpft (z. B. Zugriff auf bestimmte Datensätze oder Aktionen wie Lesen, Schreiben, Löschen).
2. Benutzerrollen: Nutzer werden einer oder mehreren Rollen zugeordnet.
3. Zugriffsprüfung: Bei jeder Aktion prüft das System, ob die Rolle des Nutzers die erforderlichen Berechtigungen besitzt.

Beispiel:

- Ein „Analyst“ darf anonymisierte Daten einsehen und Berichte generieren, hat jedoch keinen Zugriff auf private Daten oder administrative Funktionen.
- Ein „Administrator“ hat vollständigen Zugriff auf alle Daten und kann neue Nutzerrollen definieren.

Vorteile:

- Einfachheit: RBAC ist leicht zu implementieren und zu verwalten.
- Skalierbarkeit: Änderungen an Berechtigungen können zentral über die Rollen vorgenommen werden, ohne alle Nutzer einzeln anzupassen.

Nachteile:

- Eingeschränkte Flexibilität: In komplexen Szenarien mit vielen Ausnahmen (z. B. zeit- oder standortbasierte Zugriffe) wird RBAC weniger geeignet.

Attribute-Based Access Control (ABAC)

Definition:
ABAC ist ein flexibles Zugriffskontrollmodell, bei dem Berechtigungen basierend auf einer Kombination von Attributen des Nutzers, der Ressourcen und des Kontexts festgelegt werden. Attribute sind Eigenschaften wie Nutzerattribute (z. B. Abteilung, Standort), Ressourcenattribute (z. B. Datentyp, Sensitivität) oder Umgebungsattribute (z. B. Zeit, IP-Adresse).

Wie es funktioniert:

1. Attribute definieren: Attribute können sowohl statisch (z. B. Nutzerrolle) als auch dynamisch (z. B. Uhrzeit) sein.
2. Zugriffsrichtlinien: Richtlinien werden erstellt, die kombinierte Bedingungen enthalten (z. B. „Zugriff nur zwischen 9:00 und 18:00 Uhr, wenn der Nutzer in der Abteilung ‚Medizin‘ ist“).
3. Zugriffsprüfung: Bei jeder Anfrage überprüft das System, ob alle Bedingungen der Richtlinie erfüllt sind.

Beispiel:

- Ein Arzt in der Abteilung „Onkologie“ darf nur auf die Krankenakten von Patienten dieser Abteilung zugreifen, und dies nur während der Arbeitszeiten.

- Ein Forscher darf anonymisierte Daten sehen, wenn er von einer zugelassenen IPAdresse aus auf das System zugreift.

Vorteile:

- Flexibilität: ABAC ermöglicht die Definition granularer und kontextbasierter Zugriffsrichtlinien.
- Dynamische Kontrolle: Zugriffsentscheidungen können in Echtzeit auf Grundlage aktueller Attribute getroffen werden.

Nachteile:

- Komplexität: Die Verwaltung von Attributen und Richtlinien kann aufwendig sein.
- Performance: In Echtzeit müssen viele Attribute und Bedingungen geprüft werden, was die Systemlast erhöhen kann.

Fazit der Entscheidung

Auf Basis der Literatur und der etablierten Standards wird empfohlen, den Datenzugriff in Data Hubs nach folgenden Prinzipien zu gestalten:

1. RBAC oder ABAC für Zugriffskontrollen nutzen:
   - RBAC eignet sich für Szenarien mit klar definierten Nutzerrollen. Dies ist vermindert den administrativen Aufwand, da keine komplexen Zugriffskontrollen gebaut werden müssen.
   - ABAC bietet mehr Flexibilität für dynamische Umgebungen wie im medizinischen Bereich. Dieser Ansatz kann für Datahubs genutzt werden, welche strenge regulieren mit spezifischen Edge-Cases haben, um zu jederzeit die Datensicherheit zu schützen, aber auch die Möglichkeit zu haben schnell zu reagieren.
2. Anonymisierung und Pseudonymisierung kombinieren:
   - Anonymisierte Daten für öffentliche Nutzer bereitstellen, während autorisierte Nutzer auf vollständige Datensätze zugreifen können.
   - Die Kombination aus Anonymisierung und Verschlüsselung bietet maximale Sicherheit.

Grundsätzlich haben all diese Standards ihre Berechtigung und sind je nach Szenario sinnvoll einsetzbar. Beim Aufbau eines Data Hubs ist es entscheidend, die spezifischen Anforderungen zu analysieren und basierend darauf das passende Zugriffskontrollmodell auszuwählen.

Im Fall von Artha ist RBAC die geeignetere Wahl, da klar definierte Rollen existieren und der administrative Aufwand minimal gehalten werden soll. Nicht autorisierte Nutzer sollen ausschliesslich anonymisierte Daten einsehen können, während autorisierte Nutzer vollständigen Zugriff auf die Daten haben.

7.4 Login-Mechanismen und Datenverwaltung im Dashboard

Die Benutzerinteraktion mit einem Data Hub erfolgt in der Regel über ein Dashboard, welches zentrale Funktionen wie die Benutzerverwaltung, die Zuweisung von Sichtbarkeitsstufen zu Datenquellen und die Steuerung von Zugriffsberechtigungen bietet. Ein weiteres Schlüsselelement ist ein sicherer und benutzerfreundlicher Loginmechanismus, der den Zugang zu den richtigen Funktionen und Daten sicherstellt. Dieses Kapitel analysiert Standards in der Industrie und zeigt, wie diese Anforderungen in Dashboards moderner Data Hubs umgesetzt werden.

Kernfragen der Analyse

1. Welche Login-Mechanismen gelten als sicher und benutzerfreundlich?
2. Wie sind Dashboards in der Industrie strukturiert, um Datenquellen effizient zu verwalten?

3. Wie lassen sich Benutzerrollen, Sichtbarkeitsstufen und Datenquellen intuitiv verknüpfen?

Industriestandards für Login-Mechanismen

Moderne Data Hubs setzen auf bewährte Standards, um Benutzer sicher und effizient anzumelden. Zu den gängigsten Mechanismen gehören:

1. Multi-Faktor-Authentifizierung (MFA):
   - Beschreibung: Neben dem Passwort wird eine zweite Authentifizierungsmethode (z. B. ein Einmalpasswort oder ein biometrisches Merkmal) verwendet.
   - Vorteile: Erhöhte Sicherheit gegen unautorisierte Zugriffe.
   - Industriestandard: Laut einer Studie von NIST ist MFA bei sensiblen Datenanwendungen wie im Gesundheitswesen ein Muss.
2. Token-basierte Authentifizierung (z. B. OAuth 2.0):
   - Beschreibung: Nutzer erhalten nach der Anmeldung ein Token, das für den Zugriff auf Ressourcen verwendet wird.
   - Vorteile: Trennung von Authentifizierung und Autorisierung, erhöhte Sicherheit bei der API-Nutzung.
   - Beispiele: Systeme wie Google und Microsoft verwenden OAuth für sichere Benutzeranmeldungen.
3. Single Sign-On (SSO):
   - Beschreibung: Nutzer können sich einmal anmelden und auf mehrere Systeme zugreifen, ohne sich erneut authentifizieren zu müssen.
   - Vorteile: Reduzierter Aufwand für Nutzer, konsistente Sicherheitsrichtlinien
   - Beispiele: SSO wird häufig in Unternehmensanwendungen verwendet.

Struktur und Funktionen eines Dashboards

Ein modernes Dashboard für Data Hubs bietet intuitive Benutzeroberflächen und klare Strukturen, um zentrale Aufgaben wie Benutzer- und Datenverwaltung zu ermöglichen.
Wichtige Funktionen im Dashboard:

1. Benutzerverwaltung:
   - Beschreibung: Das Dashboard sollte ermöglichen, Benutzer anzulegen, zu bearbeiten und zu löschen.
   - Funktionen: Rollen zuweisen (z. B. Administrator, Analyst), Berechtigungen individuell oder gruppenbasiert festlegen.
   - Beispiel: Ein Administrator kann einem Nutzer die Rolle „Analyst“ zuweisen und ihn nur für spezifische Datenquellen autorisieren.
2. Verwaltung von Sichtbarkeitsstufen:
   - Beschreibung: Jede Datenquelle kann ein Sichtbarkeitslevel erhalten (öffentlich, privat, anonymisiert).
   - Funktionen: Datenquellen kategorisieren und den entsprechenden Sichtbarkeitslevel zuweisen, Anonymisierungsregeln direkt im Dashboard definieren.

Beispiele aus der Praxis:

- Microsoft Azure Data Hub: Bietet ein zentrales Dashboard für Benutzer- und Datenquellenverwaltung. Nutzer können auch Sichtbarkeitsstufen und Zugriffsrechte mit wenigen Klicks festlegen.
- Amazon Web Services (AWS) Data Lake: Unterstützt detaillierte Zugriffskontrollregeln, basierend auf Attributen und Rollen. Dashboards bieten auch Visualisierungen zur Datenquelle und deren Zugriffshistorie.
- Beispiel aus dem Gesundheitswesen: Dashboards in medizinischen Data Hubs, wie die von Privacy Analytics, ermöglichen es, Datenquellen zu kategorisieren (öffentlich, anonymisiert, privat) und Zugriffsberechtigungen direkt zu steuern.

Fazit der Entscheidung

Auf Basis der Literatur und Industriestandards sollten folgende Prinzipien beachtet werden:

1. Sicherer Login:
   - Implementierung von MFA, SSO und Token-basierten Authentifizierungsmethoden, um die Sicherheit zu gewährleisten.
   - Nutzung bewährter Protokolle wie OAuth 2.0 für die Authentifizierung.
2. Intuitive Benutzer- und Datenverwaltung:
   - Benutzerrollen und Datenquellen sollten in einer zentralen Benutzeroberfläche verwaltet werden können.
   - Sichtbarkeitsstufen (öffentlich, anonymisiert, privat) sollten klar definiert und einfach zuweisbar sein.

Bei Artha macht eine Token-basierte Authorisierung am meisten Sinn, da SSO keinen Nutzen hat, da Artha lediglich ein Dashboard hat und ein MFA ist zuviel Aufwand, mit zu wenigen Vorteilen. Dementsprechend aufgrund der guten Sicherheit und dem relativ einfachen Implementierung von Token-basierten Authorisieren.

7.5 Bezug auf die Anforderungen

Die im Konzept definierten Anforderungen dienen als Grundlage für die Entwicklung eines Data Hubs, der die zentrale Verwaltung von Daten, die sichere Zugriffskontrolle und die intuitive Benutzerinteraktion ermöglicht. In diesem Kapitel wird dargelegt, wie die vorgeschlagene Konzeptlösung diese Anforderungen erfüllt und welche spezifischen Massnahmen getroffen wurden, um die definierten Ziele zu erreichen. Die folgende Tabelle zeigt, wie die Anforderungen (funktional und nicht-funktional) durch die Konzeptlösung umgesetzt werden:

Anforderung	**Beschreibung**	**Umsetzung**
DFA-1: Zugriffsbeschränkungen	Es darf nur autorisierten Nutzern erlaubt sein, auf private Daten zuzugreifen.	• Implementierung von RBAC für rollenbasierte Zugriffskontrolle. • Anonyme Nutzer erhalten nur Zugriff auf anonymisierte Daten. • Verschlüsselung sensibler Daten im Ruhezustand und während der Übertragung.
DFA-2: Anonymisierung	Nicht autorisierte Nutzer sollen anonymisierte Daten einsehen können, während autorisierte Nutzer die vollständigen Daten sehen können.	• Anonymisierung der Daten im Service-Layer, basierend auf Sichtbarkeitsstufen (öffentlich, privat, anonymisiert). • Bereitstellung von Anonymisierungs- und Pseudonymisierungsfunktionen im Dashboard.
DFA-3: Datenquellen-Management	Administratoren müssen in der Lage sein, neue Datenquellen hinzuzufügen, zu verwalten und mit Sichtbarkeitsstufen zu versehen.	• Integration eines Datenquellen-Managements im Dashboard. • Übersichtliche Darstellung aller Datenquellen mit Filter- und Zuweisungsfunktionen.

Anforderung	Beschreibung	Umsetzung
NDFA-1: Sicherheit	Das System muss Datenschutzgesetze (z. B. DSGVO, HIPAA) einhalten und sicher vor unautorisierten Zugriffen sein.	• Einsatz von Verschlüsselungstechnologien (z. B. TLS). • Regelmässige Überprüfung von Zugriffskontrollen durch Audit-Protokollierung. • Konformität mit Datenschutzstandards durch Anonymisierung und Protokollierung.
NDFA-2: Benutzerfreundlichkeit	Das System muss intuitiv bedienbar sein, um die Benutzerakzeptanz zu fördern.	• Design eines nutzerzentrierten Dashboards mit einfacher Navigation. • Direkte Zuweisung von Rollen und Datenquellen durch Administratoren möglich. • Design-Besprechungen mit dem Kunden.
NDFA-3: Skalierbarkeit	Das System muss mit steigender Nutzeranzahl und Datenvolumen skalierbar bleiben.	• Nutzung von horizontaler Skalierung und Lastverteilung im Service-Layer. • Einsatz von Caching-Techniken zur Verbesserung der Performance. • Realistische Nutzungslastenschätzung.

Table 7.1: Datenverwaltung: Bezug auf die Anforderungen

Nächste Schritte

Im Kapitel ≪Technische Implementation / Analyse der Lösung≫ wird eine für Artha-spezifische Lösung implementiert und entsprechend dokumentiert, welche stark auf dieses hier beschriebene Lösungskonzept eingeht.

8 Technische Implementation / Analyse der Lösung

8.1 Analyse der aktuellen Lösung von Artha (Watersense)

Basierend auf der Analyse der aktuell vorhandenen Lösung von Artha (Watersense) und den Aspekten eines Data-Hubs aus unserer wissenschaftlichen Arbeit lässt sich wie folgt beurteilen, ob Watersense die Kriterien eines Data-Hubs erfüllt:

Zentralisierte Datenspeicherung

Watersense verwendet eine Cloud-basierte Lösung, bei der Messdaten automatisch in eine zentralisierte SQL-Datenbank hochgeladen werden. Dies erfüllt das Merkmal eines zentralen Datenspeicherorts, das für einen Data-Hub essenziell ist.

Positiv: Die Daten sind standortunabhängig verfügbar und können in einem zentralen Dashboard visualisiert werden. Dies vereinfacht den Zugriff für Kunden und Systemadministratoren erheblich.

Verbesserungspotenzial: Eine Erweiterung der Datenbank, um heterogene Datenquellen besser zu integrieren, würde die Flexibilität weiter steigern.

Integration von Datenquellen

Ein Data-Hub zeichnet sich durch die Integration verschiedenster Datenquellen aus. Watersense fokussiert sich auf die Messdaten eines UV-Spektrometers.

Positiv: Die aktuellen Sensoren werden effizient integriert.

Fehlend: Eine flexible Architektur zur Einbindung weiterer Sensortypen oder Datenquellen scheint aktuell nicht vollständig umgesetzt zu sein.

Visualisierung von Daten

Watersense bietet ein Dashboard mit Funktionen zur Visualisierung von Trends und Anomalien durch Diagramme wie Liniendiagramme und Heatmaps.

Positiv: Das System ermöglicht eine benutzerfreundliche Datenvisualisierung, die für die Entscheidungsfindung notwendig ist.

Verbesserungspotenzial: Weitere interaktive und anpassbare Visualisierungsoptionen könnten die Benutzerfreundlichkeit erhöhen und den Anforderungen eines modernen Data-Hubs besser gerecht werden.

Skalierbarkeit

Die Skalierbarkeit ist ein essenzieller Aspekt eines Data-Hubs. Watersense nutzt eine Cloud-Plattform, die grundsätzlich skalierbar ist.

Positiv: Die Cloud-Architektur ermöglicht eine gewisse Skalierbarkeit.

Fehlend: Mechanismen, die ein dynamisches Wachstum von Nutzern und Datenquellen ermöglichen, sind nicht detailliert beschrieben. Auch Echtzeit-Verarbeitung könnte stärker in den Fokus rücken.

Sicherheit und Zugriffskontrolle

Ein modernes Data-Hub-System sollte robuste Sicherheits- und Zugriffskontrollmechanismen bieten. Watersense ermöglicht ortsunabhängigen Zugriff, doch detaillierte Informationen zu Sicherheitsmassnahmen fehlen in der aktuellen Lösung.

Verbesserungspotenzial: Die Einführung granularer Rollen- und Zugriffskonzepte (z. B. RBAC oder ABAC) würde das System robuster machen.

Flexibilität und Erweiterbarkeit

Die Möglichkeit, das System durch neue Funktionen, wie KI-gestützte Analysen oder zusätzliche Visualisierungen, zu erweitern, ist ein weiterer essenzieller Punkt.

Positiv: Die Architektur von Watersense ist prinzipiell erweiterbar.

Fehlend: Eine detaillierte Roadmap oder Implementierung für zukünftige Funktionen wurde nicht vollständig beschrieben.

Fazit

Watersense enthält viele Elemente eines Data-Hubs, insbesondere in den Bereichen Datenspeicherung, Visualisierung und Skalierbarkeit. Es fehlen jedoch noch einige zentrale Aspekte:

- **Datenintegration:** Mehr Flexibilität bei der Anbindung verschiedener Datenquellen.
- **Erweiterbarkeit:** Ein klares Konzept für zukünftige Technologien wie KI oder Echtzeit-Datenverarbeitung.
- **Sicherheit:** Verbesserte Zugriffskontrolle und Datenschutzmechanismen.
- **Interaktivität und Nutzerfokus:** Erweiterte und anpassbare Visualisierungen, die die Benutzerfreundlichkeit steigern.

Mit diesen Erweiterungen könnte Watersense ein vollständiger und moderner Data-Hub werden, der die Anforderungen unseres Konzepts vollumfänglich erfüllt.

8.2 Mock-ups der Applikation

Auf Wunsch des Kunden haben wir zunächst einen Prototyp entwickelt, der die grundlegenden Funktionen einer Datenverwaltung demonstriert. Anschließend wurde dieser Prototyp als Basis genutzt, um eine umfassende Lösung zu implementieren, die Daten nach Sichtbarkeitsstufen wie öffentlich, privat und anonymisiert organisiert und effizient verwaltet. Die Daten, welche anonymisiert werden, wären die Sensoren, da diese sensible Daten wie Location und Seriennummer gespeichert haben. Später werden für unberechtigte die Sensoren, so wie die Daten der Sensoren gefiltert, damit diese nicht gesehen werden können. Die Sensordaten selber sind nicht sensibel, da diese lediglich eine Sensor-ID-Referenz (Datenbank ID, welche nicht, wie die Seriennummer, auf dem Gerät selbst ist) haben und mit dieser Information lässt sich kein Rückschluss auf Sensor oder Ort des Sensors machen. Jedoch wenn der Sensor auf anonymisiert gestellt ist, dann wird die Information des Sensors verschleiert und nicht die Sensordaten selbst, da sie, wie gesagt, selbst keine sensiblen Daten beinhalten.

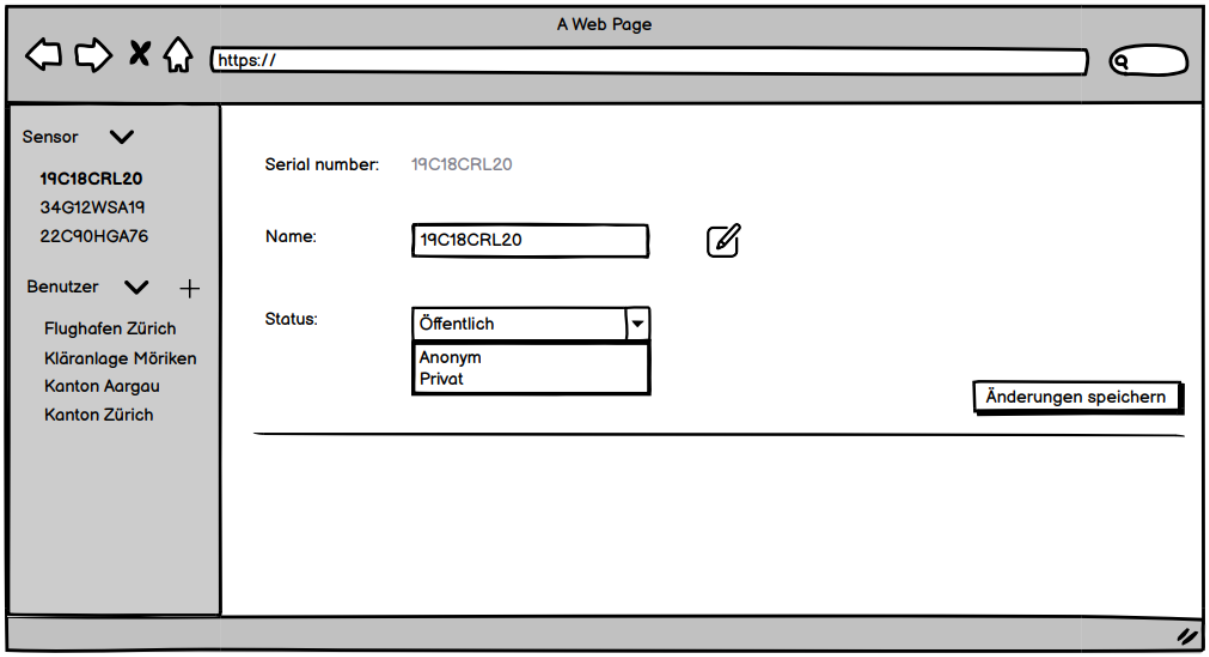


Figure 8.1: Sensorseite

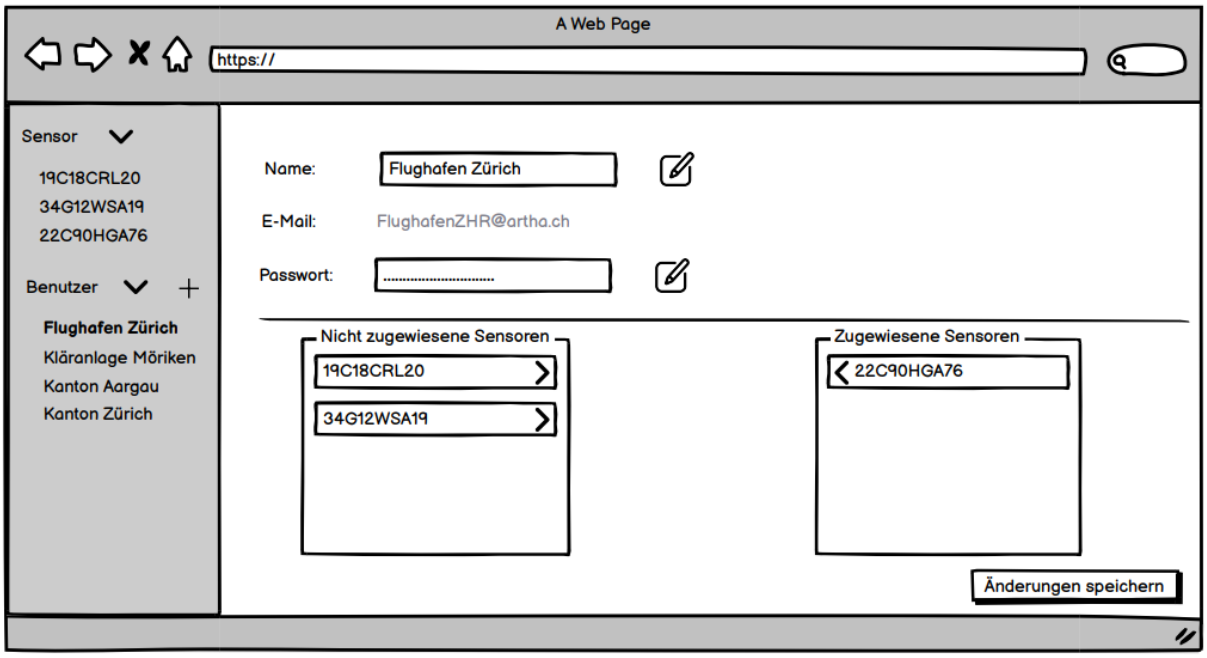


Figure 8.2: Benutzerverwaltung

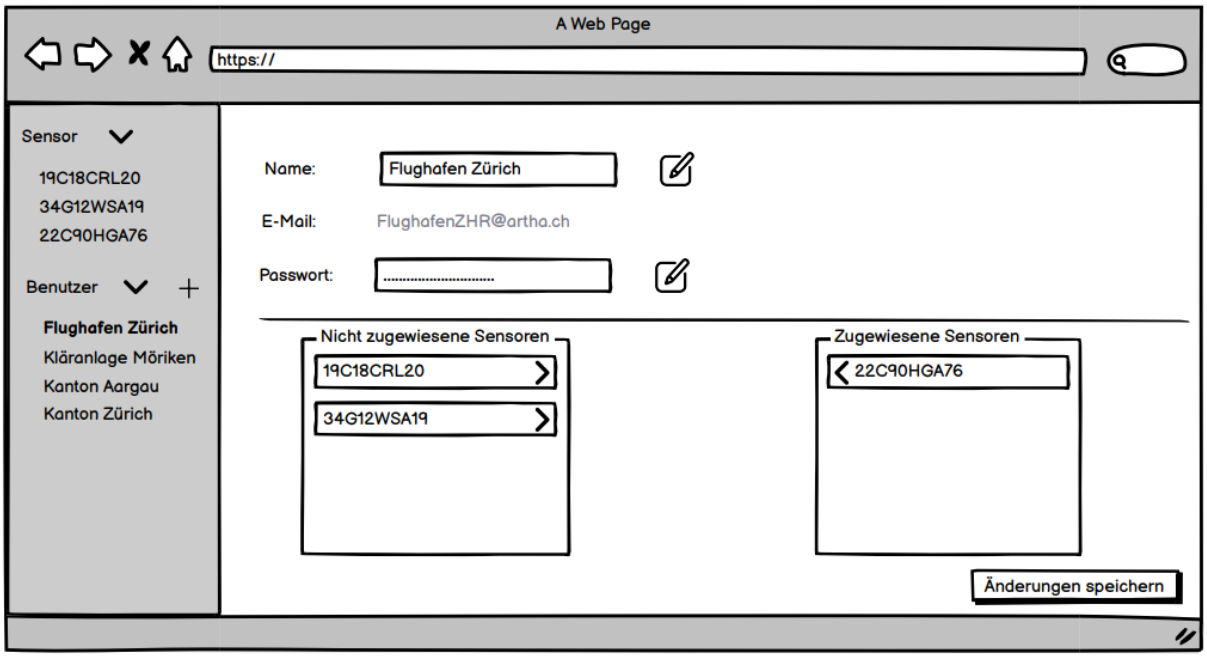


Figure 8.3: Benutzererstellung

8.3 Technische Implementation

In diesem Kapitel wird die technische Umsetzung der Datenverwaltungslösung detailliert beschrieben. Der Fokus liegt darauf, wie Daten basierend auf ihren Sichtbarkeitsstufen
(öffentlich, privat, anonymisiert) gefiltert und verarbeitet werden. Zudem wird erläutert, wie Administratoren neue Benutzer anlegen, Rollen zuweisen und spezifische Sensoren oder Datenquellen verwalten können.

Datenbank

Das bestehende Datenbankschema wurde um neue Tabellen erweitert, ohne die bestehenden Tabellen und Datensätze zu verändern. Da sich Artha im Produktivbetrieb befindet und keine Testumgebung verfügbar ist, war es essenziell, die Datenintegrität zu gewährleisten und die bestehenden Daten unverändert zu lassen.

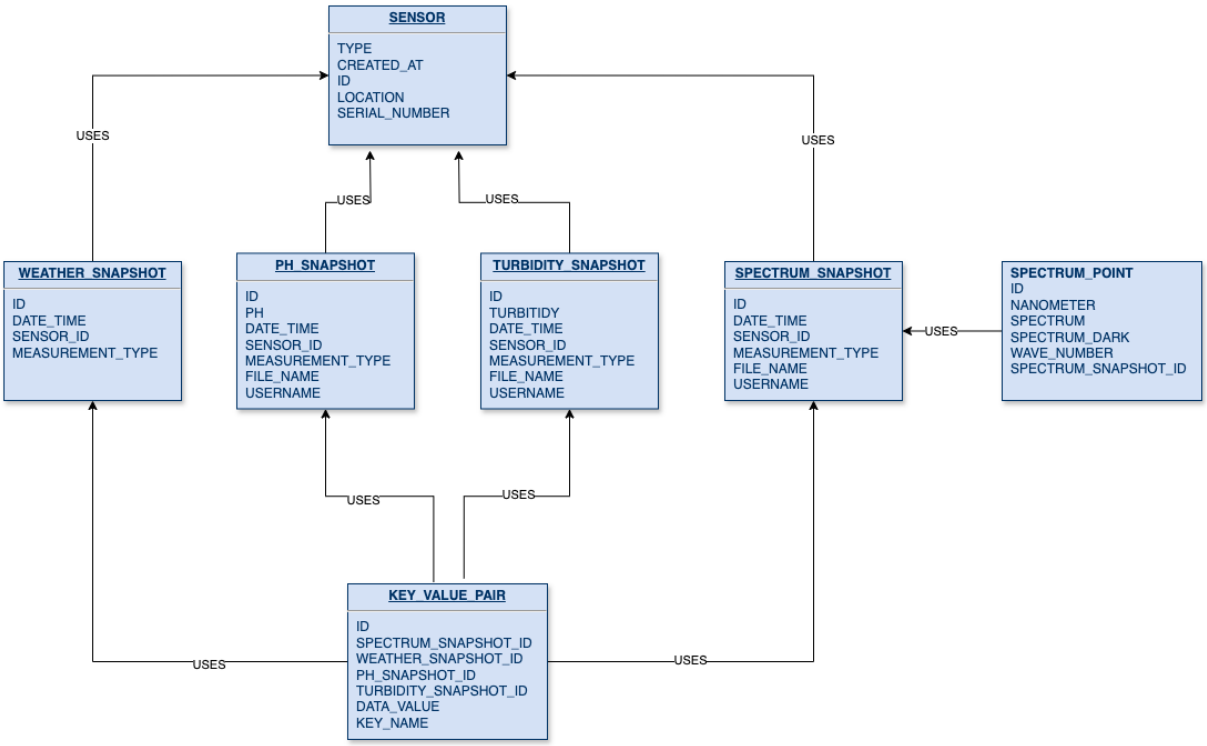


Figure 8.4: Datenbank vorher

Dies ist das Datenbankschema vor der Änderung. Es besteht aus einem Sensor, welcher jeweils Daten liefert und diese werden in den Snapshots gespeichert, je nach Art der Daten. Zusammen im KEY_VALUE_PAIR werden dann die Daten gespeichert. Das bedeutet, dass die Snapshots jeweils die Meta-Daten beinhalten (an welchem Tag, welcher Sensor, Dateiname usw.) und im KEY_VALUE_PAIR werden dann diese Daten mit einem Wert gesetzt. Dieses Schema wurde erweitert durch die Benutzer und Sensibilitätssicht.

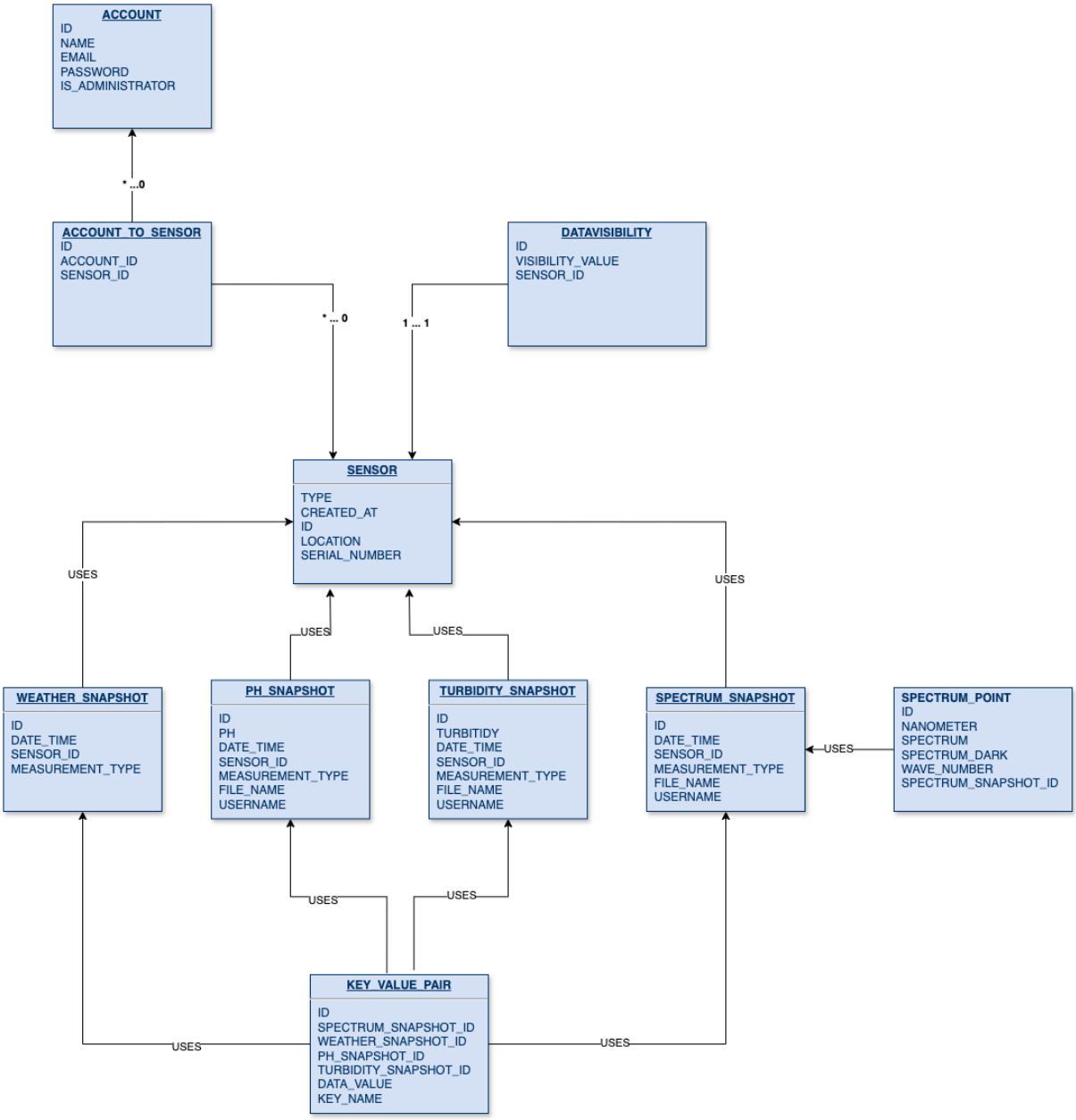


Figure 8.5: Datenbank aktuell

Somit kommen neu drei Tabellen dazu. `Account` speichert die Benutzerdaten, also den Namen, die E-Mail und das Passwort. Dazu ist es wichtig zu wissen, ob der Benutzer ein Administrator ist, da nur dieser auf die Data-Management-Seite Zugriff hat, um andere Benutzer zu erstellen und ihnen Sensoren zuzuweisen. Die Zwischentabelle (`ACCOUNT_TO_SENSOR`) dient als Zuweisung von `Account` zu `Sensor`. Ein `Account` kann mehrere `Sensoren` haben, und ein `Sensor` kann bei mehreren `Accounts` sein. Die Tabelle `DATAVISIBILITY` hat die Funktion eines Enums in Java. Sie hat drei Werte, welche die Werte öffentlich, privat und anonymisiert repräsentieren. Zusätzlich wird ein `Sensor` genau zu einem Wert zugewiesen, damit je ein `Sensor` immer nur maximal einen Sichtbarkeits-Level hat. Dadurch, dass `DATAVISIBILITY` im Service-Layer durch Jakarta umgesetzt wurde, wird `VISIBILITY_VALUE` in Java mit einem Enum befüllt. Dieses Enum kann die eben genannten Werte öffentlich, privat oder anonymisiert haben.

Die Funktionalität des Datenzugriffs wird im Service-Layer implementiert.

Schnittstellen (Beschreibung)

Neu:

- Benutzererstellung und Mutation
- Sensormutation und Sichtbarkeitsstufe setzen

Angepasst:

- Sensordaten holen (wird zuvor gefiltert)

Datenfilterung und Datenanonymisierung

Die Endpunkte, welche vom Benutzer aufgerufen werden (SensorController) werden zusätzlich nun vom AccountService vor der Rückgabe des Wertes gefiltert. Der AccountService überprüft, ob die Daten des Sensors öffentlich, privat oder anonym sind und filtert die Daten aus. Falls die Daten anonymisiert werden müssen, werden sie dem DataAnonymizationService übergeben. Dieser codiert die sensitiven Daten.

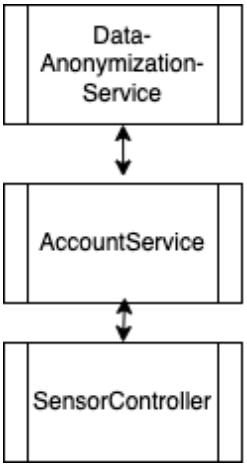


Figure 8.6: Anonymisierungsprozess der Daten

Der DataAnonymizationService nimmt die Location des Sensors und die Seriennummer und speichert sie in eine HashMap. Der Schlüssel ist jeweils die Location oder Seriennummer und der Wert (value) ist eine zufällige Zahl, welche in das HEX-Format umgewandelt wird. Dadurch wird bei mehreren Aufrufen des Endpunkts der gleiche zufällige Wert für den gleichen Sensor zurückgegeben. Dies hat den Vorteil, dass es einfacher ist die Daten zu verstehen, da jeweils der gleiche Wert benutzt wird. Nach 24 Stunden wird die HashMap geleert und es wird ein neuer zufälliger Wert benutzt für einen Sensor.

Der AccountService nimmt den Sensor entgegen und falls dieser zum Benutzer gehört oder öffentlich ist, wird er zurückgegeben. Wenn er als anonym definiert ist, wird er durch den DataAnonymizationService geschickt und dort bearbeitet und anonymisiert zurückgegeben. Wenn der Sensor als privat deklariert ist und nicht dem Benutzer gehört wird er ausgefiltert.

Webapplikation

Die Webseite wurde analog dem Mockup implementiert.

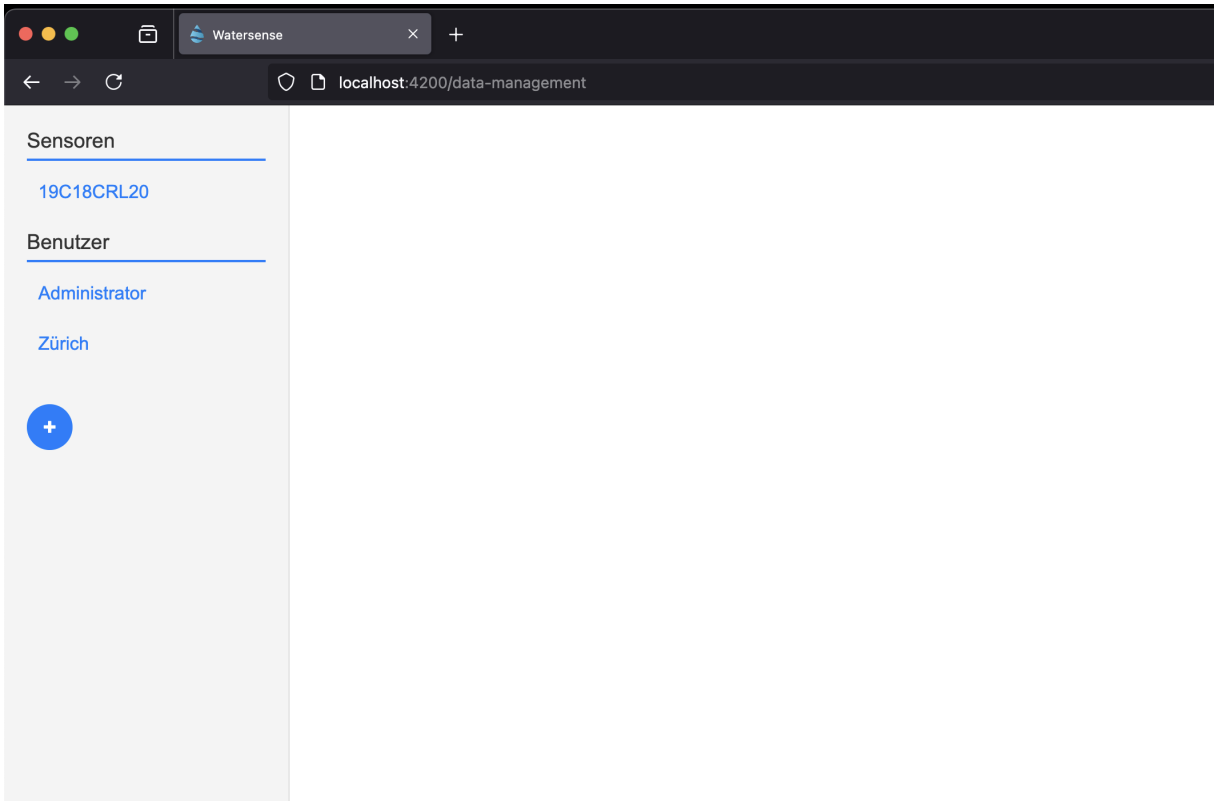


Figure 8.7: Navigationsleiste

So wurde auf der linken Seite eine Navigationsleiste implementiert. Dort werden die Sensoren und Benutzer angezeigt. Mit einem Mausklick auf einen Sensor oder Benutzer wird die Detailsicht auf der rechten Seite angezeigt. Beim Klicken auf den Plus-Knopf wird die Detailsicht für das Erstellen eines neuen Benutzers aufgemacht.

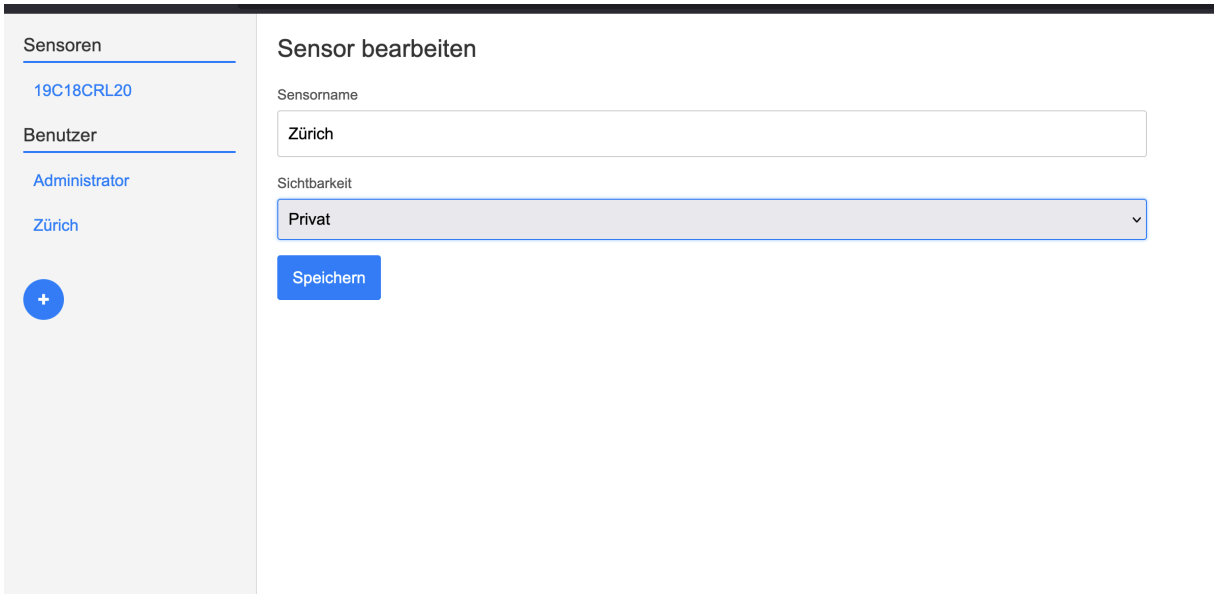


Figure 8.8: Sensorseite

Beim Sensor kann der Sensorname gesetzt werden und die Sichtbarkeit.

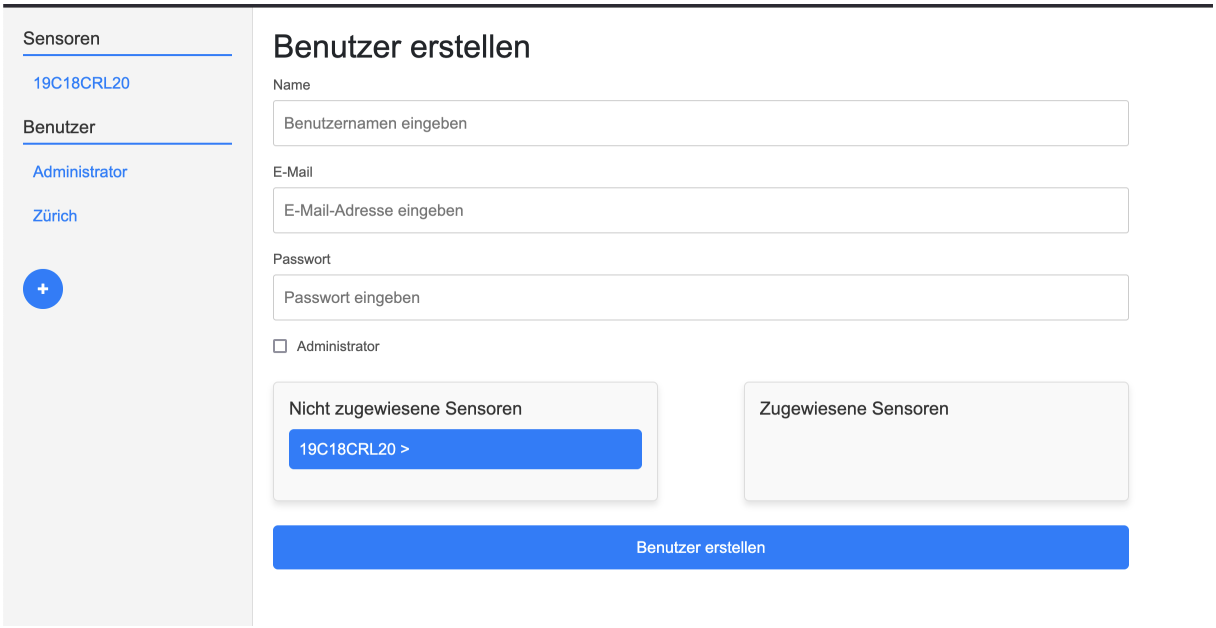


Figure 8.9: Benutzererstellung

Beim Erstellen des Benutzers müssen Name, E-Mail und Passwort gesetzt werden. Optional kann der Benutzer auch als Administrator gehandhabt werden, dann kann der Benutzer alle Sensoren und Daten sehen und andere Benutzer erstellen. Darunter können verfügbare Sensoren in einem einfachen Layout dem Benutzer zugewiesen werden.

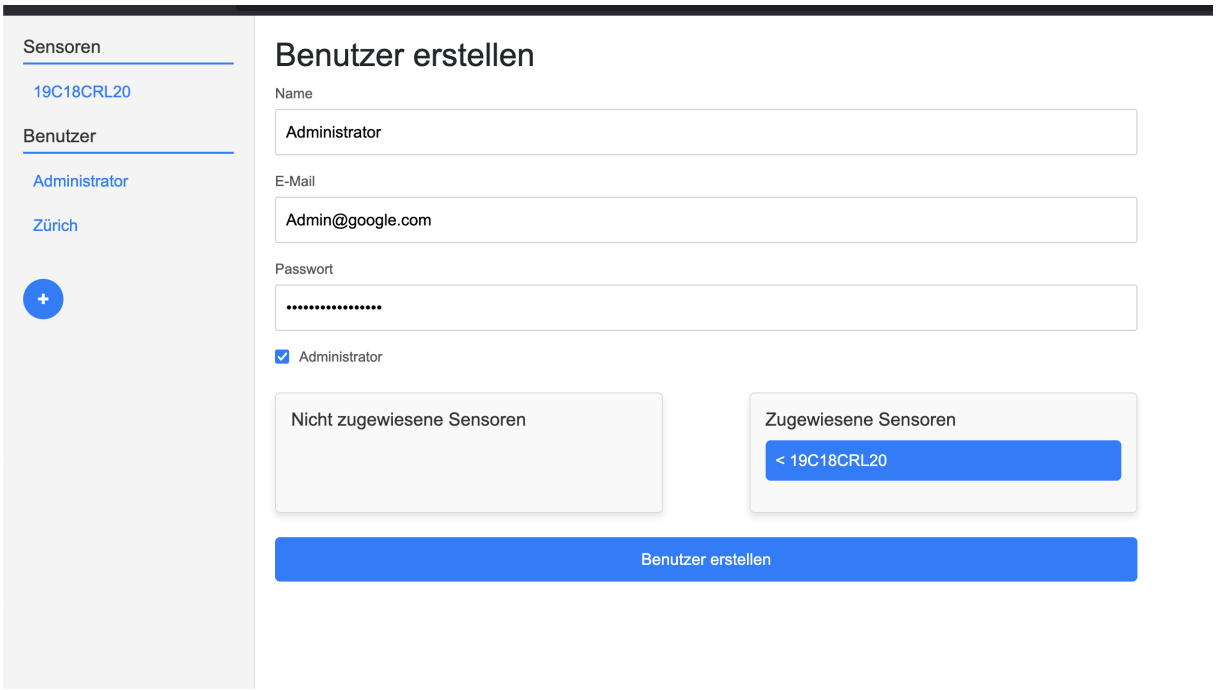


Figure 8.10: Benutzerbearbeitung

Das Mutieren eines bestehenden Benutzers wird analog der Erstellung im gleichen Fenster gemacht, nur dass hier eben kein Benutzer erstellt wird, sondern nur der ausgewählte mutiert.

9 Ergebnis

9.1 Zusammenfassung der Ergebnisse

Während dieser wissenschaftlichen Arbeit haben wir uns intensiv mit der Konzeption eines Data-Hubs auseinandergesetzt und dabei drei verschiedene Konzeptlösungen entwickelt. Diese Konzepte bilden die Grundlage für den Aufbau eines modernen Data-Hubs und decken unterschiedliche Anforderungen ab. Sie adressieren technische Herausforderungen und bieten praxistaugliche Ansätze für Datenintegration, Visualisierung und Sicherheit.

Im Mittelpunkt stand der theoretische Aufbau eines skalierbaren und flexiblen Data-Hubs. Ziel war es, eine zentrale Plattform zu schaffen, die Daten aus verschiedenen Quellen integriert, verarbeitet und bereitstellt. Der Fokus lag auf einer modularen Architektur, die es ermöglicht, Komponenten wie Datenintegration, Verarbeitung und Speicherung unabhängig voneinander zu entwickeln und zu erweitern. Hauptfunktionen wie die Nutzung standardisierter Schnittstellen (z. B. REST-APIs), Echtzeitdatenverarbeitung und Rollen-basierte Zugriffskontrollen (RBAC) wurden integriert. Dieser Ansatz gewährleistet eine einfache Integration neuer Datenquellen und unterstützt zukünftige Erweiterungen wie KI-gestützte Analysen. Zudem wurden alle Aspekte eines Data-Hubs evaluiert und mit Best Practices aus der Literatur ergänzt.

Die zweite Lösung konzentrierte sich auf die Sicherheit und Verwaltung von Daten im Hub. Ein zentraler Aspekt war die Anonymisierung sensibler Daten, um den Datenschutz zu gewährleisten und die Daten für öffentliche oder wissenschaftliche Zwecke bereitzustellen. Dies wurde durch die Implementierung von Anonymisierungsalgorithmen erreicht. Die Zugriffskontrolle wurde durch die Kombination von RBAC und Attribute-Based Access Control (ABAC) flexibel gestaltet, um sowohl standardisierte als auch kontextabhängige Zugriffsrechte zu ermöglichen. Diese Massnahmen tragen dazu bei, sowohl die Sicherheit der Daten als auch deren Nutzbarkeit zu optimieren.

Die dritte Konzeptlösung befasste sich mit der Entwicklung eines benutzerfreundlichen Dashboards für die Visualisierung der Daten. Das Ziel war es, komplexe Daten verständlich und intuitiv zugänglich zu machen. Verschiedene Diagrammtypen wurden integriert, um Trends und Zusammenhänge darzustellen. Besonderes Augenmerk wurde auf die Benutzerfreundlichkeit gelegt, um auch Nicht-Technikern den Zugang zu erleichtern. Das Dashboard dient als zentrale Schnittstelle für Datenanalysen und Entscheidungsprozesse und wurde als anpassbares Tool konzipiert, das verschiedene Nutzerbedürfnisse erfüllt.

Die drei Konzeptlösungen ergänzen sich und bilden gemeinsam eine ganzheitliche Grundlage für einen modernen Data-Hub. Der Aufbau sorgt für technische Skalierbarkeit und Modularität, die Datenverwaltung gewährleistet Sicherheit und Flexibilität, und die Visualisierung stellt sicher, dass die Daten für verschiedene Nutzergruppen verständlich und nutzbar sind. Diese Arbeit legt den Grundstein für ein zukunftsfähiges System, das auf die steigenden Anforderungen einer datengetriebenen Welt zugeschnitten ist.

Im Rahmen unserer praktischen Arbeit lag auch ein Fokus auf der Implementierung der zuvor entwickelten Konzeptlösung der Datenverwaltung. Dabei konnten wir wesentliche Funktionen erfolgreich umsetzen, auch wenn einige Bereiche wie der Login-Mechanismus noch unvollständig sind. Ein zentraler Fortschritt war die Umsetzung der Anonymisierung von Daten. Mechanismen wurden implementiert, die sicherstellen, dass sensible Informationen anonymisiert werden, bevor sie an unautorisierte Nutzer weitergegeben werden. Hierbei wurden beispielsweise Standortdaten und Seriennummern von Sensoren entweder vollständig entfernt oder durch Pseudonyme ersetzt, sodass keine Rückschlüsse auf die Quelle möglich sind. Ein weiterer wichtiger Erfolg war die Zugriffskontrolle für private Daten. Diese werden zuverlässig vor unbefugtem Zugriff geschützt, sodass Nutzer ausschliesslich die Daten einsehen können,

für die sie explizit berechtigt sind. Daten, die als privat gekennzeichnet sind, bleiben für andere Nutzer vollständig unsichtbar. Zusätzlich haben wir die Verwaltung von Sichtbarkeitsstufen implementiert, die es ermöglicht, Datenquellen in Kategorien wie öffentlich, privat oder anonymisiert einzuteilen. Dadurch wird die Sichtbarkeit und Verfügbarkeit der Daten flexibel und sicher gesteuert. Diese Funktionalität wurde direkt in der Datenverwaltungsschicht und im Frontend umgesetzt, um Konsistenz und Sicherheit zu gewährleisten. Der Login-Mechanismus konnte allerdings nicht mehr fertiggestellt werden, wodurch die Nutzeridentifikation und -authentifizierung noch fehlen. Dennoch bietet die realisierte Datenverwaltung eine solide Grundlage für die weitere Entwicklung und demonstriert die praktische Umsetzbarkeit der zuvor erarbeiteten theoretischen Konzepte.

9.2 Beantwortung der Fragestellungen

Was sind die wesentlichen Funktionen eines Data Hubs, welche Features sind notwendig und welche optional?

Die wesentlichen Funktionen eines Data Hubs wurden im Konzept und in der Konzeptlösung detailliert erarbeitet, um einen umfassenden Überblick über notwendige und optionale Features zu bieten. Ein Data Hub hat primär die Aufgabe, Daten aus unterschiedlichen Quellen effizient zu integrieren, zu verarbeiten, zu speichern und bereitzustellen. Zu den notwendigen Funktionen zählen die Integration von Datenquellen über standardisierte Schnittstellen wie APIs, die Klassifizierung von Daten nach Sichtbarkeitsstufen (öffentlich, privat, anonymisiert), die sichere Speicherung der Daten sowie die Bereitstellung durch flexible APIs oder Dashboards. Ebenso unverzichtbar sind Mechanismen zur Zugriffskontrolle wie Role-Based Access Control (RBAC), um die Datensicherheit zu gewährleisten. Optionale Features hingegen erweitern die Funktionalität und passen den Data Hub spezifisch an die Bedürfnisse des Anwendungsfalls an. Dazu gehören die Integration von Machine-Learning-Tools zur automatisierten Analyse der Daten, Echtzeit-Datenverarbeitung für zeitkritische Anwendungen und zusätzliche Visualisierungsoptionen im Dashboard. Weitere optionale Features können erweiterte Rollen- und Berechtigungsmodelle, benutzerdefinierte Workflows oder Tools zur Datenqualitätssicherung umfassen. Die Modularität des Data Hubs ermöglicht es, diese optionalen Funktionen je nach Bedarf zu integrieren oder zu erweitern. Die vorgestellten Funktionen und Features bilden eine flexible Basis, um die Anforderungen moderner Datenarchitekturen zu erfüllen, wobei der Fokus stets darauf liegt, notwendige und optionale Komponenten individuell an die spezifischen Anforderungen anzupassen.

Welche Visualisierungen und Funktionen sind am besten geeignet, um komplexe Daten in einem Data-Hub-Dashboard benutzerfreundlich und effektiv darzustellen?

Die Frage, welche Visualisierungen und Funktionen am besten geeignet sind, um komplexe Daten in einem Data-Hub-Dashboard benutzerfreundlich und effektiv darzustellen, wurde im Konzept und in der Konzeptlösung eingehend behandelt. Ein benutzerfreundliches Dashboard sollte darauf ausgelegt sein, komplexe Daten in einer klaren und anpassbaren Form zu präsentieren, sodass sowohl technische als auch nicht-technische Nutzer effektiv damit arbeiten können. Zentrale Visualisierungen umfassen interaktive Diagramme wie Balken-, Linien- und Kreisdiagramme, die es ermöglichen, Trends, Verteilungen und Korrelationen schnell zu erkennen. Die Kombination aus klaren Visualisierungen, interaktiven Funktionen und einer intuitiven Benutzeroberfläche ermöglicht es, komplexe Daten effektiv darzustellen und den Mehrwert eines Data Hubs optimal zu nutzen. Die Modularität des Systems erlaubt es zudem, diese Funktionen je nach Anwendungsfall flexibel zu erweitern oder anzupassen.

Wie lässt sich gewährleisten, dass sensible Daten geschützt bleiben, während gleichzeitig die Vorteile des Data-Hubs optimal genutzt werden können?

Die Frage, wie sensible Daten geschützt bleiben können, während gleichzeitig die Vorteile eines Data Hubs optimal genutzt werden, wurde im Konzept und in der Konzeptlösung eingehend analysiert. Der Schutz sensibler Daten erfordert eine Kombination aus technischen, organisatorischen und architektonischen Massnahmen, die individuell auf die Anforderungen des jeweiligen Data Hubs abgestimmt sind. Zu den zentralen Ansätzen gehört die Implementierung von Zugriffskontrollmechanismen wie

Role-Based Access Control (RBAC) oder Attribute-Based Access Control (ABAC), die sicherstellen, dass nur autorisierte Nutzer auf private oder sensible Daten zugreifen können. Darüber hinaus ist die Anonymisierung oder Pseudonymisierung sensibler Daten ein essenzieller Bestandteil, der es ermöglicht, Daten für nicht autorisierte Nutzer zugänglich zu machen, ohne die Privatsphäre zu gefährden. Diese Prozesse werden typischerweise im Service-Layer implementiert, um eine dynamische Anpassung der Sichtbarkeitsstufen zu ermöglichen. Die optimale Nutzung der Vorteile des Data Hubs wird durch die Modularität des Systems gewährleistet, da sie die flexible Anpassung der Sicherheitsmassnahmen an spezifische Anforderungen erlaubt. Diese Massnahmen stellen sicher, dass sensible Daten geschützt bleiben, während der Data Hub gleichzeitig seine zentrale Funktion als Plattform für die effiziente Integration, Verarbeitung und Bereitstellung von Daten erfüllt.

10 Reflexion und Ausblick

10.1 Reflexion

Rückblickend sind wir mit unserer Arbeit grundsätzlich zufrieden, auch wenn uns bewusst ist, dass wir nicht alles vollständig umsetzen konnten. Positiv ist, dass wir dort, wo wir nicht fertig wurden, eine solide Grundlage geschaffen haben, auf der in Zukunft weitergearbeitet werden kann. Im nächsten Kapitel beschreiben wir detailliert, welche Schritte noch notwendig sind, um die fehlenden Teile abzuschliessen. Mit den Konzepten, die wir erarbeitet haben, sind wir sehr zufrieden. Sie sind durchdacht und decken die zentralen Anforderungen eines modernen Data-Hubs ab.
Allerdings hat uns die Recherche viel Zeit gekostet. Es schien, als gäbe es eine überwältigende Menge an Informationen, und wir fühlten uns zunächst nicht ausreichend vorbereitet, um eine klare Meinung zu den verschiedenen Ansätzen zu entwickeln. Wir haben viel Zeit gebraucht, um die verschiedenen Perspektiven einzuordnen und eine fundierte Basis für unsere Entscheidungen zu schaffen. Deswegen haben später als geplant mit der praktischen Umsetzung begonnen.
Ein weiterer Punkt, der uns mehr Zeit gekostet hat als erwartet, war die Aufgabenvereinbarung zu Beginn des Projekts. Wir sind von der ursprünglichen Projektausschreibung ausgegangen und hatten den Eindruck, dass es vor allem um das Dashboard bei Artha geht. Als wir dann festgestellt haben, dass der Fokus auf dem Aufbau eines Data-Hubs liegt, mussten wir uns erst in das Thema einarbeiten. Data-Hubs waren für uns ein völlig neues Gebiet, und es hat länger gedauert als geplant, das Thema zu verstehen und mit den Dokumenten und der Recherche richtig anzufangen. Trotzdem sind wir stolz darauf, wie wir uns in das komplexe Thema hineingearbeitet haben und ein solides Ergebnis präsentieren können.

10.2 Ausblick

Die nächsten Schritte sind sowohl die Fertigstellung unserer begonnenen Implementierungen als auch die Erweiterung unseres Projekts durch zusätzliche Analysen und Vergleichen. Zunächst ist es essenziell, die noch ausstehenden Teile unserer Datenverwaltung abzuschliessen, insbesondere die Integration des Login-Mechanismus. Dabei soll bei jedem API-Call die Authentifizierung des Nutzers geprüft werden, sodass sicherstellt wird, dass nur autorisierte Nutzer Zugriff auf die ihnen zugewiesenen Daten erhalten. Sobald diese Funktionalität implementiert ist, muss umfangreich getestet werden, ob die Daten wie vorgesehen korrekt gefiltert und bereitgestellt werden. Obwohl wir davon ausgehen, dass unsere bisherigen Lösungen in Kombination mit dem Login funktionieren, ist es wichtig, alle Anforderungen systematisch zu prüfen und potenzielle Schwachstellen zu identifizieren.
Für die zukünftige Arbeit haben wir mit unserem Projekt eine solide Grundlage gelegt, die als Vorlage für den Aufbau weiterer Data-Hub-Systeme genutzt werden kann. Unsere bisherigen Ergebnisse und die erarbeiteten Konzepte bieten einen strukturierten Ansatz, der zukünftige Projekte in diesem Bereich leiten kann. Bei der Weiterentwicklung oder dem Aufbau neuer Data-Hubs ist es jedoch entscheidend, dass die Anforderungen im Vorfeld detailliert definiert werden. Diese klare Spezifikation bildet die Basis für alle weiteren Entscheidungen und ermöglicht es, den optimalen Typ des Data-Hubs entsprechend der spezifischen Bedürfnisse und Rahmenbedingungen auszuwählen.
Wie in unserer Dokumentation beschrieben, sollte dabei sorgfältig analysiert werden, welche Art von Data-Hub – zentralisiert, verteilt oder gemischt – den Anforderungen am besten entspricht. Nur durch diese fundierte Auswahl kann sichergestellt werden, dass das System sowohl technisch als auch organisatorisch optimal implementiert wird. Unsere Arbeit liefert hierfür nicht nur theoretische Grundlagen, sondern auch praktische Ansätze, die direkt auf ähnliche Projekte angewendet werden können. Sie bietet einen Leitfaden, um die wesentlichen Fragen zu Architektur, Skalierbarkeit, Sicherheitsmassnahmen und Datenverwaltung zu beantworten und dabei die spezifischen Gegebenheiten des jeweiligen Anwendungsfalls zu berücksichtigen.

Quellenverzeichnis

[1] V. Gadepally and J. Kepner, *Technical report: Developing a working data hub*, 2020. arXiv: 2004.00190 [cs.DB]. [Online]. Available: https://arxiv.org/abs/2004.00190.

[2] S. Lindsey, „Data hub vs. data lake vs. data warehouse: A comparative analysis“, 2024. [Online]. Available: https://www.cdata.com/blog/data-hub-vs-data-lake-vs-data-warehouse.

[3] Internet of Water, *Internet of water hubs*, Zugriff am 13. Januar 2025, 2025. [Online]. Available: https://internetofwater.org/resources/hubs/.

[4] admin-otomatic, *Data hub definition: Alles, was sie über data hubs wissen müssen*, Zuletzt abgerufen am 17. Januar 2025, 2024. [Online]. Available: https://iatechnologie.com/de/data-hub-definition-alles-was-sie-uber-data-hubs-wissen-mussen/.

[5] C. Giebler, C. Gröger, E. Hoos, H. Schwarz, and B. Mitschang, „Modeling data lakes with data vault: Practical experiences, assessment, and lessons learned“, in *Conceptual Modeling: 38th International Conference, ER 2019, Salvador, Brazil, November 4–7, 2019, Proceedings 38*, Springer, 2019, pp. 63–77.

[6] DFKI, *Datahub europe gestartet*, Zuletzt abgerufen am 17. Januar 2025, 2024. [Online]. Available: https://www.dfki.de/web/news/datahub-europe-gestartet.

[7] S. Barker, „Building an enterprise-grade data hub on google cloud“, Oct. 2024, Zugriff am 17. Januar 2025. [Online]. Available: https://expertbeacon.com/building-an-enterprise-grade-data-hub-on-google-cloud/.

[8] T. Priebe, S. Neumaier, and S. Markus, *Von data warehouse bis data mesh: Ein wegweiser durch den dschungel analytischer datenarchitekturen*, 2022. arXiv: 2212.03612 [cs.DB]. [Online]. Available: https://arxiv.org/abs/2212.03612.

[9] L. Whalen and H. Valafar, *Datadock: An open source data hub for research*, 2024. arXiv: 2406.16880 [cs.DB]. [Online]. Available: https://arxiv.org/abs/2406.16880.

[10] Meinolf Schäfer, *Sap data hub – funktionen und vorteile*, Zuletzt abgerufen am 17. Januar 2025, n.d. [Online]. Available: https://www.gambit.de/wiki/sap-data-hub/.

[11] Norstat. „Ein leitfaden für forscher zu dashboards“. Zugriff am 16. Januar 2025. [Online]. Available: https://norstat.co/de/loesungen/dashboards/ein-leitfaden-fuer-forscher-zu-dashboards/.

[12] A. Janes, A. Sillitti, G. Succi, *et al.*, „Effective dashboard design“, *Cutter IT journal*, vol. 26, no. 1, pp. 17–24, 2013.

[13] D. Orlovskyi and A. Kopp, „A business intelligence dashboard design approach to improve data analytics and decision making.“, in *IT&I*, 2020, pp. 48–59.

[14] F. Giampaolo *et al.*, „A privacy preserving service-oriented approach for data anonymization through deep learning“, in *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2023, pp. 0738–0746. DOI: 10.1109/DASC/PiCom/CBDCom/Cy59711.2023.10361409.

[15] F. Giampaolo, S. Izzo, S. Siccardi, A. Polimeno, V. Bellandi, and F. Piccialli, „Real-time anonymization of sensitive personal data using a service-based architecture“, in *2023 IEEE International Conference on Web Services (ICWS)*, 2023, pp. 701–703. DOI: 10.1109/ICWS60048.2023.00090.

[16] L. Arbuckle, *Anonymization at scale: From dataset to pipeline*, Accessed: 2023-01-14, 2024. [Online]. Available: https://privacy-analytics.com/resources/articles/anonymization-at-scale-from-dataset-to-pipeline/.

[17] D. .-. D. S. C. of America, *Data anonymization: Protecting privacy in large-scale analytics*, Accessed: 2023-01-14, 2024. [Online]. Available: https://www.dasca.org/world-of-data-science/article/data-anonymization-protecting-privacy-in-large-scale-analytics.

[18] K. Krupa, *Data Hub Guide for Architects*. Progress Software, n.d. Accessed: 2023-01-14. [Online]. Available: https://www.progress.com/docs/default-source/marklogic-docs/data-hub-guide-for-architects.pdf.

Ehrlichkeitserklärung

Hiermit erklären wir, die vorliegende Projektarbeit selbständig und nur unter Benutzung der angegebenen Quellen verfasst zu haben. Die wörtlich oder inhaltlich aus den aufgeführten Quellen entnommenen Stellen sind in der Arbeit als Zitat bzw. Paraphrase kenntlich gemacht. Diese Projektarbeit ist noch nicht veröffentlicht worden. Sie ist somit weder anderen Interessierten zugänglich gemacht noch einer anderen Prüfungsbehörde vorgelegt worden.

Windisch, 17. Januar 2025

Name: Stefan Simic

Unterschrift: 

Name: Damjan Stojanovic

Unterschrift: 

A Appendix 1

Fachhochschule Nordwestschweiz

Windisch, 17.01.25

Informationen zum Projektablauf & Projektvereinbarung IP5, Artha – The new way of water

Betreuer: Norbert Seyff
Nitish Patkar

Auftraggeber: Patrick Frei, Artha

Projektdauer: 16.09.2024 bis 17.01.2025

Aufgabenstellung

1. Einarbeitung

1.1 Erwartungen zum Projektablauf

Termine
Fixieren Sie Termine frühzeitig, d.h. Reviews mit dem Kunden und ca. alle 2-3 Wochen einen Besprechungstermin mit Ihren Betreuern. Klären Sie allfällige Abwesenheiten gleich zum Projektstart.

Meetings
Meetings sind grundsätzlich dazu vorgesehen, den aktuellen Projektstand zu besprechen, Fragen zu klären, Ideen zu diskutieren und die nächsten Schritte zu planen.
Senden Sie vorgehend eine Traktandenliste sowie alle weiteren nötigen Unterlagen an die Betreuenden. Erläutern Sie zu Beginn jedes Projektmeetings den aktuellen Projektstand, die Fortschritte und Probleme sowie die geplanten Schritte.
Sie können die Meetings nach Absprache und bei Bedarf auch für spezifische Fragestellungen nutzen (z.B. Micro-Teaching, Brainstorming, Präsentation von Ergebnissen oder Mentoring).
Kommen Sie jedoch mit möglichst konkreten Fragestellungen an eine Besprechung.
Bitte halten Sie die besprochenen Inhalte und Entscheide zeitnah protokollarisch fest.

1.2 Vorgaben für die Vereinbarung

Als erste Aufgabe in Ihrer Arbeit müssen Sie diese Vereinbarung (vgl. Punkt 3) vervollständigen. Eine erste Version ist bis ca. 2-4 Wochen (BB 4-6 Wochen) nach dem Kickoff zu erstellen. Bei Projekten die technische Analyse benötigen, kann es sinnvoll sein eine erste Implementationsiteration vor der Abgabe der Projektvereinbarung durchzuführen. Bitte füllen Sie folgende Punkte aus:

Ausgangslage
Formulieren Sie das Projekt und die Ausgangslage in eigenen Worten.

Projektvision
Beschreiben Sie, welche Ziele und Resultate mit dem Projekt erreicht werden sollen. Die Vision dient der Ableitung von Qualitätskriterien.

Projektspezifische Fragestellungen

Formulieren Sie zusätzlich zu den allgemeinen Fragestellungen 2-3 projektspezifische Fragestellungen. Diese dienen Ihnen als Basis für eine wissenschaftlich strukturierte Recherche und die Ableitung geeigneter Lösungsansätze.
Beispiele von Fragestellungen und Lösungsansätzen:

- Mit welchen Ansätzen erreichen Sie die definierte Zielgruppe?
  Lösungsansatz: Entwicklung von Konzepten für nutzerzentrierte Ansätze und Umsetzung des User Interface der Applikation, z.B. in Form von Storyboards mit einer durchgehenden User Story oder GUI-Prototypen.
- Mit welchem technischen Konzept erreichen Sie die gewünschte Lösung?
  Lösungsansatz: Technologie-Evaluation, Entwicklung technisches Lösungskonzept (PoC), Definition von Subsystemzerlegung, Architekturstil und Technologien.
- Welche Interaktionskonzepte, Interfacegestaltungen und Bildsprachen eignen sich für Ihren Ansatz?
  Lösungsansatz: Entwicklung von Interaktionskonzepten und graphisch sorgfältig gestalteter, klar strukturierter Bildsprache für das Interface Design, welche den Anforderungen an eine innovative User Experience gerecht werden.
- Mit welcher technischen Umsetzung erfüllen Sie die Anforderungen an Funktionalität, Benutzbarkeit, Zuverlässigkeit, Effizienz und Wartbarkeit?
  Lösungsansatz: Implementation einer lauffähigen Applikation für ein zuvor evaluiertes Setup und definierte Nutzungsszenarien basierend auf geeigneten Technologien und Frameworks
- Für die erfolgreiche Einführung der Software sind die Korrektheit, die Benutzbarkeit und die Zuverlässigkeit zentral. Wie können Sie diese sicherstellen und testen?
  Lösungsansatz: Eingehendes Testing von Korrektheit, Benutzbarkeit und Zuverlässigkeit, Dokumentation von Testresultaten, Demonstration der Erfüllung der Anforderungen mittels Live-Test.

Methodik
Beschreiben Sie, wie die Ziele erreicht werden. Welche Methodiken setzen Sie dafür ein (z.B. Scrum, Agile, wissenschaftliches Vorgehen, etc.).

Planung
Erstellen Sie eine initiale Projektplanung. Definieren Sie Arbeitspakete sowie deren Deliverables.

Risiko Assessement
Identifizieren und bewerten Sie Risiken innerhalb des Projektes und entwickeln Sie Strategien, wie Sie mit diesen umgehen.

2. Dokumentation

2.1 Schriftliche Dokumentation (Thesis Rapport)

Dokumentieren Sie schriftlich und elektronisch Ihre Vorgehensweise, den theoretischen Hintergrund, die Anwendung von Methoden und Konzepten, die Implementierungen und Testresultate. Überprüfen Sie auch den geplanten mit dem tatsächlichen Zeitplan, die Zielerreichung und reflektieren Sie Erfahrungen.
Achten Sie unbedingt darauf, dass Sie persönliche Kommentare von Fakten strikte trennen. **Der Hauptteil der Dokumentation ist vollständig faktenbasiert**. Das bedeutet, dass keine Sätze der Art „Dann hatten wir das Problem x und versuchten es mit y zu lösen." auftreten dürfen. Falls ein solches Problem x aber wirklich existiert und nicht nur Sie damit nicht gleich zu Rande kamen, dann sollen Sie schreiben: „Tests z haben klar gezeigt, dass ein Problem x besteht. Mögliche Ansätze, um das Problem x zu lösen, sind a, b und c. Wir haben uns aus den Gründen e und f für Variante c entschieden." Erst in einem Extraabschnitt können Sie Ihre persönlichen Eindrücke, Erlebnisse, Probleme und dergleichen formulieren.
Wichtig ist auch, dass eine gute Dokumentation auch noch nach vielen Jahren gelesen werden können muss und dass sie dem Leser ein gut abgerundetes Bild vermittelt, auch dann, wenn er nicht direkt an der Arbeit beteiligt war. Bitte legen Sie auch grossen Wert auf sprachliche Qualität.

Das Zielpublikum dieser Dokumentation sind die Betreuer, die Experten, der Auftraggeber und zukünftige Studierende, welche in diesem Bereich weiterarbeiten wollen.
Die Dokumentation wird im Projektverlauf erstellt. Für das zweite Coaching Meeting soll ein Inhaltsverzeichnis des Berichts vorbereitet werden, damit dieses mit den Betreuenden rückgesprochen werden kann. **Die Teile zur Recherche und Analyse sind nach dem ersten Projektdrittel zu präsentieren.**

Auf dem Web-Portal der FHNW erstellen Sie eine Projektpräsentation (Web-Summary). Für Bachelorarbeiten im Frühlingssemester erstellen Sie zusätzlich ein Plakat für die Ausstellung. Beide Artefakte sind vor Veröffentlichung mit den Betreuenden zu besprechen.

Folgende Informationen sind auf allen Publikationen zu nennen:
- Logo FHNW
- Semesterprojekt IP5 bzw. Bachelorthesis (IP6)
- Projektname
- Frühlings- oder Herbstsemester 202x, Studiengang Informatik (Profilierung iCompetence), Hochschule für Technik, Fachhochschule Nordwestschweiz
- Vorgelegt von: Name Studierende
- Eingereicht bei: Name Betreuende
- Auftraggeber: Firma / Institution
- Datum

Weitere Informationen bezüglich des Verfassens von Berichten finden sie auch auf der Plattform Informationskompetenz

2.2 Präsentationen

Präsentationen finden in Absprache mit den Betreuenden und dem Auftraggeber statt. Bei der Verteidigung Ihrer Bachelorthesis wird auch die Expertin oder der Experte anwesend sein.

Präsentationen verschaffen einerseits einen Überblick über das gesamte Projekt und die erreichten Ergebnisse und vertiefen ein oder zwei wichtige interessante Fragestellungen. Ebenfalls Teil der Präsentation ist eine prägnante Demonstration der Benutzung Ihrer Software. Bei den Zuhörern dürfen Sie von einem technisch versierten Fachpublikum ausgehen. Planen Sie 30’ für die Präsentation und Demonstration ein und reservieren Sie 30’ für Fragen und Diskussion.

2.3 Publikation der Projektresultate

Werden die Arbeit oder Teile der Arbeit veröffentlicht, sind alle Namen der Projektbeteiligten (Studierende, Betreuende, Auftraggeber) sowie der Name der Institution (FHNW) zu nennen. Vor jeder Veröffentlichung müssen Betreuende und Auftraggeber vorgängig um ihr Einverständnis gebeten werden.

2.4 Protokolle

Protokolle bilden einen wichtigen Teil der Dokumentation. Professionell geführte Protokolle enthalten folgende Punkte:
- Datum, Raum, Zeit, Teilnehmende, Entschuldigte
- Traktanden
- Projektstand (ggf. mit Screenshots, Skizzen, o.ä; Stand gemäss Planung)
- Inhalt (faktenbasiert, thematisch strukturiert und inhaltlich nachvollziehbar; Entscheidungen sind festgehalten)
- Offene Fragen
- Nächste Schritte; Termine & Aufgaben (wer, was & bis wann)

2.5 Dokumentenablage

Richten Sie für die Betreuer Zugriffe auf Ihre Dokumentenablage ein. Falls keine zwingenden Gründe dagegen sprechen verwenden Sie dafür die Gitlab Infrastruktur der FHNW[1].
Verwenden Sie diese Dokumentablage auch, um zusätzliche Dokumentation abzulegen, z.B. wie Ihr Code ausgeführt werden kann.
Stellen Sie sicher, dass eine adäquate Commit-History für die Betreuenden sichtbar ist.

2.6 Abgabe

Die Projektabgabe umfasst (sofern nicht anders mit dem Projektbetreuer definiert) die folgenden Artefakte:

- Schriftliche Dokumentation (Thesis Rapport)
- Projektvereinbarung (in der Regal als Anhang in der Thesis)
- Codebase (dokumentiert & mit readme zur Erläuterung des Setup), gehostet auf Gitlab der FHNW (https://gitlab.fhnw.ch/iit-projektschiene/[Semester]/[Projekt]) und als ZIP-Archiv
- Link zum Projektauftritt auf dem Web-Portal der FHNW
- weitere Artefakte, falls vorhanden (Screencast empfohlen, …)

[1] https://gitlab.fhnw.ch/

3. Projektspezifische Vereinbarung

3.1 Ausgangslage

Das Projekt Artha – The new way of water zielt darauf ab, die aktuellen Herausforderungen in der Wasserqualitätsanalyse zu bewältigen, beispielsweise in Kläranlagen. Das bestehende System, das mit UV-Spektrometern Wasserproben analysiert, weist eine wesentliche Einschränkung auf: Die Daten sind nur lokal verfügbar und können nur vor Ort ausgewertet werden. Dadurch entsteht ein hoher Verwaltungsaufwand und eine verzögerte Reaktionsfähigkeit bei Wasserqualitätsabweichungen.

Dieses Problem führt dazu, dass Anlagen oft länger und intensiver betrieben werden, um mögliche Probleme zu vermeiden, was zu unnötigen Betriebskosten führt. Das Artha Watersense-Projekt hat bereits erste Fortschritte erzielt, indem es die Messdaten in die Cloud hochlädt und ein Dashboard zur Visualisierung der Daten bereitstellt.

Die Daten in der Datenbank (Google Cloud-Dienst “Cloud SQL”) sind an eine feste Struktur für bestimmte Sensoren und Anlagen gebunden, was die Skalierbarkeit erschwert. Google bietet diverse Dienste an wie zum Beispiel eine Aufbereitung der Daten durch künstliche Intelligenz, was jedoch momentan noch nicht integriert ist. Auch Hinzufügen von neuen Sensoren oder ganzen Anlagen führt zu einem grossen Mehraufwand und benötigt das Fachwissen von den Entwicklern der Applikation, was den gesamten Prozess verlangsamt.

Die Herausforderung besteht darin, die Stelle, an der die Daten zusammenfließen, so flexibel und gleichzeitig strukturiert zu gestalten, dass sie effizient verarbeitet und zugänglich gemacht werden können. Es muss ein System geschaffen werden, das neue Datenquellen integriert und dabei eine sinnvolle Organisation der Informationen gewährleistet. Dies ermöglicht eine einfache Weiterverarbeitung der Daten, ohne dass die Struktur bei jeder Erweiterung aufwendig angepasst werden muss.

3.2 Projektvision

Das Ziel dieses Projekts ist es, eine Grundlage für einen späteren Data-Hub zu legen und den Prozess der Erweiterbarkeit damit zu verbessern. Das bedeutet auch eine Evaluierung der momentanen Cloud-Lösung sowie auch des Dashboards, um sicherzustellen, dass es sowohl die neuen technischen Anforderungen wie auch die Nutzerbedürfnisse erfüllt, mit einem Fokus auf die Herausforderungen wie Echtzeit-Datenintegration, Skalierbarkeit und die Verbesserung der Benutzererfahrung.

Durch die Implementierung der neuen Lösung wird die Einschränkung der starren Datenstrukturen überwunden und eine flexible Plattform geschaffen, die als zentrale Grundlage für das Sammeln, Verarbeiten und Analysieren großer Datenmengen dient. Diese Plattform ermöglicht das Einbinden neuer Geräte und Sensoren, so wie auch neue Kunden und Anlagen an neuen Standorten, ohne dass umfangreiche Anpassungen erforderlich sind.

3.3 Fragestellungen

A. **Was sind die wesentlichen Funktionen eines Data Hubs, welche Features sind notwendig und welche optional?**
Ein Data Hub dient als zentrale Plattform zur Integration und Verwaltung von Daten aus verschiedenen Quellen. Um die wesentlichen Funktionen eines Data Hubs sowie die Unterscheidung zwischen notwendigen und optionalen Features zu ermitteln, wird ein methodisches Vorgehen gewählt. Zunächst erfolgt eine Literaturrecherche, um zentrale Anforderungen und Merkmale aus wissenschaftlichen Artikeln und Branchenstandards zu extrahieren. Anschließend werden bestehende Data Hub-Lösungen analysiert, um zu identifizieren, welche Features als notwendig und welche als optionale Ergänzungen gelten.
Die gewonnenen Erkenntnisse werden in zwei Kategorien unterteilt: notwendige Funktionen für den Betrieb eines Data Hubs und optionale Features, die den Nutzen je nach Anwendungsfall erweitern. Ergänzend erfolgt eine Anforderungsanalyse, die branchenspezifische Standards und Nutzerbedarfe berücksichtigt. Auf dieser Basis wird ein Konzept entwickelt, das die wesentlichen Funktionen priorisiert und eine optimale Lösung vorschlägt.

B. **Welche Visualisierungen und Funktionen sind am besten geeignet, um komplexe Daten in einem Data-Hub-Dashboard benutzerfreundlich und effektiv darzustellen?**
Das Dashboard in einem Data-Hub ist die zentrale Nutzungsschnittstelle für den Endnutzer, weswegen es wichtig ist alle wichtigen und relevanten Funktionen einzubinden, um die Nutzung des Data-Hubs einfach und übersichtlich zu gestalten. Um die Frage zur Gestaltung eines Data-Hub-Dashboards zu beantworten, wird ein strukturiertes Vorgehen gewählt. Zunächst werden die Anforderungen analysiert, um die relevanten Benutzerprozesse und Funktionen zu identifizieren. Dabei werden die spezifischen Bedürfnisse der Nutzer und die verschiedenen Rollen im System berücksichtigt. Im nächsten Schritt erfolgt eine Literaturrecherche, die sich auf Best Practices und konzeptionelle Ansätze zur Gestaltung von Dashboards konzentriert. Dabei werden wissenschaftliche Erkenntnisse und bestehende Lösungen untersucht, um allgemeine Gestaltungsprinzipien und funktionale Anforderungen zu ermitteln.
Die gewonnenen Erkenntnisse werden genutzt, um ein Konzept zu entwickeln, das Benutzerfreundlichkeit und die Abdeckung aller relevanten Prozesse vereint. Ziel ist eine klare Struktur, die die komplexen Anforderungen eines Data Hubs optimal unterstützt.

C. **Wie lässt sich gewährleisten, dass sensible Daten geschützt bleiben, während gleichzeitig die Vorteile des Data-Hubs optimal genutzt werden können?**
Der Data-Hub sammelt Daten von verschiedenen Nutzern und Endpunkten. Um eine sichere Verwaltung dieser Daten zu ermöglichen, ist es entscheidend, den Zugang sorgfältig zu steuern. Einige Daten sollten privat bleiben, während andere eventuell öffentlich oder für bestimmte Benutzer zugänglich gemacht werden können. Andere Daten dürfen nur anonym für Auswertungen benutzt werden. Es muss klar definiert sein, welche Daten für wen zugänglich sind und unter welchen Bedingungen der Zugang erlaubt oder eingeschränkt wird. Dafür werden bereits bestehende Lösungen verglichen und Anforderungen an das System gestellt, aus welchem ein Konzept entwickelt wird, das sowohl den Schutz sensibler Daten gewährleistet als auch eine flexible und sichere Datenverwaltung innerhalb des Data-Hubs ermöglicht.

3.4 Methodik

Für die Zielerreichung setzen wir auf Methodiken, welche folgende Schlüsselaspekte abdecken:

Projektplanung: Kombination von Wasserfallmodell und Kanban: Für die Grobplanung des Projekts wird das Wasserfallmodell verwendet, um klare Phasen, Meilensteine und Abhängigkeiten zu definieren. Dies stellt sicher, dass ein strukturierter Ablauf und eine klare Orientierung über den gesamten Projektzeitraum hinweg gewährleistet sind. Kanban ergänzt dies durch eine agile Verwaltung der Aufgaben, die es erlaubt, flexibel auf Veränderungen einzugehen. Mithilfe von Kanban-Boards wird der Fortschritt kontinuierlich visualisiert, wodurch Transparenz geschaffen und die Reaktionsfähigkeit auf neue Anforderungen oder Herausforderungen verbessert wird.

Anforderungsanalyse (Requirements Engineering): Um die Bedürfnisse und Erwartungen der Nutzer sowie die technischen Anforderungen zu definieren, wird ein iterativer Prozess angewendet. Dabei werden in regelmäßigen Meetings mit Kunden und Betreuern Anforderungen erhoben, die dokumentiert werden. Diese dienen als Grundlage für die Entwicklung und helfen dabei, klare Ziele zu formulieren. Der Fokus liegt auf der Beantwortung zentraler Fragestellungen. Ebenfalls werden durch Literaturrecherche und Konkurrenzvergleiche Anforderungen an das Produkt erhoben.

User Centred Design: Ein nutzerzentrierter Ansatz wird verfolgt, um sicherzustellen, dass das Dashboard benutzerfreundlich, intuitiv und funktional ist. Iterative Prototypen dienen als Grundlage für regelmäßige Usability-Tests, bei denen Feedback der potenziellen Anwender eingeholt und integriert wird.

Wissenschaftliches Arbeiten und Literaturrecherche: Zur Beantwortung spezifischer Forschungsfragen und Anforderungen, wird eine fundierte Literaturrecherche durchgeführt. Dabei werden wissenschaftliche Artikel, technische Berichte und Best Practices analysiert, so wie Konkurrenzprodukte verglichen. Der Austausch mit Fachexperten liefert zusätzliche Perspektiven und ermöglicht es, theoretische Erkenntnisse in praktische Lösungen zu übersetzen.

3.5 Planung

Das Wasserfallmodell besteht aus aufeinander folgenden Phasen, die nacheinander durchlaufen werden. Zuerst werden in der **Anforderungsanalyse** alle Projektanforderungen erfasst und dokumentiert. In der **Entwurfsphase** wird das System auf Basis dieser Anforderungen geplant und strukturiert. Anschließend erfolgt in der **Implementierungsphase** die Programmierung. Nach der Umsetzung wird das System in der **Testphase** überprüft, um Fehler zu finden und sicherzustellen, dass es den Anforderungen entspricht. Danach erfolgt die **Einführung**, bei der das System in Betrieb genommen wird. Schließlich sorgt die **Wartungsphase** dafür, dass es langfristig funktioniert und aktualisiert wird.

Phasen

Nr.	Phase	Dauer	Von	Bis
P-1	Projektmanagement	18 Wochen	16.09.2024	17.01.2025
P-2	Anforderungsanalyse	7 Wochen	16.09.2024	01.11.2024
P-3	Entwurf	3 Wochen	01.11.2024	22.11.2024
P-4	Implementierung	4 Wochen (Inklusive Projektwoche)	22.11.2024	20.12.2024
P-5	Testing	2 Wochen	20.12.2024	03.01.2025
P-6	Einführung	1 Woche	03.01.2025	10.01.2025
P-7	Wartung	1 Woche	10.01.2025	17.01.2025

Meilensteine

Nr.	Meilensteine	Phase	Termin
M-1	Aufgabenvereinbarung genehmigt	P-1	01.11.20224
M-2	Data-Hub Konzept entwickelt	P-3	22.11.2024
M-3	Data-Hub Prototyp getestet und ausgewertet	P-4	20.12.2025
M-4	MVP-Einführung in Kundenumgebung	P-5	03.01.2025
M-5	Abgabe IP5	P-6	10.01.2025

Arbeitspakete

Nr.	Arbeitspaket	Dauer	Phase	Deliverable	Aktivitäten
AP-1	Projektmanagement	50h	P-1	Aufgabenvereinbarung, Protokolle, Projektplanung	Dokumentation, Aufgabenvereinbarung, Planung
AP-2	Projektdokumentation	80h	P-1	Schriftliche Dokumentation	Dokumentation
AP-3	Analyse Google Cloud-Dienste	30h	P-2	Interner Bericht für unsere Dokumentation, Protokoll der Ergebnisse	Recherche, Konkurrenzvergleich, Dokumentation, Auswertung
AP-4	Analyse des Technologie Stacks	35h	P-2	Interner Bericht für unsere Dokumentation, Protokoll der Ergebnisse	Recherche, Auswertung und Analyse des aktuellen Standes, Dokumentation
AP-5	Implementierung der Datensicherheit	40h	P-3, P-4	Interner Bericht für unsere Dokumentation, Konzeptlösung, Implementierung im Artha Projekt	Literaturrecherche, Dokumentation, Lösungskonzept entwickeln, Implementierung des Lösungskonzepts
AP-6	Konzeptlösung für Data-Hub	85h	P-3, P-4 P-5	Interner Bericht für unsere Dokumentation, Konzeptlösung, Prototyp, Artha Projekt	Literaturrecherche, Konzeptentwicklung, Dokumentation, Auswertung und Analyse aktueller Technologie, Prototyp
AP-7	Konzeptlösung für Dashboard	40h	P-3, P-4	Interner Bericht für unsere Dokumentation, Prototypen, Designs, Artha Projekt	Literaturrecherche, Vergleich mit anderen Dashboards, Dokumentation, Auswertung und Analyse des aktuellen Dashboards, Prototyp

Zeitplan

Folgende Tabelle zeigt zu welchem Zeitpunkt wir uns in welcher Phase befinden, sowie wann an welchen Arbeitspaketen gearbeitet wird. Die Meilensteine sind jeweils rot eingezeichnet. Der Zeitplan ist noch nicht vollständig und wird laufend ergänzt.

			September			Oktober				November				Dezember					Januar	
Phase	Arbeitspakete	Dauer (h)	16	23	30	7	14	21	28	4	11	18	25	2	8	16	23	30	6	13
P-1 Projektmanagement	AP-1 Projektmanagement	40	6	6	6	6	4	4	4	4	1	1	1	2	2	2	2	2	4	2
	AP-2 Projektdokumentation	40				4	4	4	6	4	6	6	6	M-1	6	6	6	6	10	12
P-2 Anforderungsanalyse	AP-3 Analyse Google Cloud-Dienste	20								4	6	6	12							
	AP-4 Analyse des Technologie Stacks	15									6	6	12	8						
P-3 Entwurf	AP-5 Implementierung Datensicherheit	10											12	8	8	8				
	AP-6 Konzeptlösung Data-Hub	20											12	8	8	M-2	5	5		
	AP-7 Konzeptlösung Dashboard	10														8	8	8		
P-4 Implementierung	AP-5 Implementierung Datensicherheit	20																8	8	8
	AP-6 Konzeptlösung Data-Hub	25																8	8	
	AP-7 Konzeptlösung Dashboard	15																4	4	
P-5 Testing	AP-6 Konzeptlösung Data-Hub	5																	6	M-3
P-6 Einführung																			3	M-4, M-5
P-7 Wartung																				

3.6 Risiko Assessement

Nr.	Risiko	Beschreibung
R-01	Kommunikationsprobleme	Unzureichende Kommunikation innerhalb des Teams, die zu Missverständnissen führt. Auch zwischen Kunde und Betreuern.
R-02	Projektüberschreitung	Eine Projektüberschreitung, welche durch eine unzureichende Zeitplanung verursacht wird.
R-03	Sicherheitsprobleme	Sicherheitsprobleme durch unzureichende Datenschutzmassnahmen bei der Integration von neuen Datenquellen.
R-04	Ausfall eines Teammitglieds	Der Ausfall eines Teammitglieds kann zu Einschränkungen im Fortschritt und zu Verzögerungen führen.
R-05	Einbindung von neuen Anlagen	Schwierigkeiten beim Einbinden von neuen Anlagen durch unerwartete Komplikationen.
R-06	Probleme mit dem bestehenden System	Unsaubere Arbeit und mangelnde Dokumentation können unseren Arbeitsprozess verlangsamen.
R-07	Änderung der Anforderungen	Zusätzliche Wünsche von Betreuern / Kunde können Änderungen im Zeitplan auslösen.

Risiko Bewertung

Eintritts-wahrscheinlichkeit	Sehr wahrscheinlich				
	wahrscheinlich			R-03	
	möglich			R-02, R-05	
	unwahrscheinlich		R-06	R-01, R-07	R-04
	Sehr unwahrscheinlich				
		Sehr gering	gering	kritisch	Sehr kritisch
		Schadensausmass			

Massnahmen

Nr.	Risiko	Massnahme
R-01	Kommunikationsprobleme	Regelmässige Meetings, klare Protokolle und Nutzung von Kommunikationsplattformen.
R-02	Projektüberschreitung	Erstellung eines detaillierten Projektplans mit Milestones und regelmäßige Überprüfungen der Ressourcen und des Fortschritts.
R-03	Sicherheitsprobleme	Implementierung strenger Sicherheitsrichtlinien, Durchführung von Penetrationstests und regelmäßige Sicherheitsupdates.
R-04	Ausfall eines Teammitglieds	Regelmässiger Austausch und das Hochladen von der Arbeit in gemeinsamen Ablagen können dieses Problem dämpfen.
R-05	Einbindung von neuen Anlagen	Frühzeitige Tests mit den Sensoren und Anlagen können die Kompatibilitätsprobleme vorzeitig erkennen.
R-06	Probleme mit dem bestehenden System	Frühe Rücksprache mit den ehemaligen Projektmitgliedern und Kunden helfen offene Fragen zu klären.
R-07	Änderung der Anforderungen	Offene / ehrliche Kommunikation und eine klare Zieldefinierung von Anfang an beugen dieses Problem vor.

4. **Schlussbestimmung**
   Die Unterzeichneten anerkennen, den Text gelesen und verstanden zu haben und verpflichten sich mit Ihrer Unterschrift die aufgeführten Punkte und die allgemeine Sorgfaltspflicht einzuhalten.

Windisch, den 14.01.2025

Betreuer

Norbert Seyff .....

Nitish Patkar .....

Studierende

Damjan Stojanovic .....

Stefan Simic .....