



INOQ

Data Mesh Architektur

Aus der Engineering-
Perspektive

Jochen Christ • Simon Harrer • Larysa Visengeriyeva

Data Mesh Architektur

Aus der Engineering Perspektive

**Jochen Christ
Dr. Simon Harrer
Dr. Larysa Visengeriyeva**

innoQ Deutschland GmbH
Krischerstraße 100 · 40789 Monheim am Rhein · Germany
Phone +49 2173 33660 · www.INNOQ.com

Layout: Tammo van Lessen with X₃LaTeX
Design: Susanne Kayser. Typesetting: André Deuerling
Print: mediaprint solutions GmbH, Paderborn, Germany

Data Mesh Architektur – Aus der Engineering Perspektive

Published by innoQ Deutschland GmbH
1st Edition · April 2023

Copyright © 2023 Jochen Christ, Simon Harrer, Larysa
Visengeriyeva

Inhaltsverzeichnis

1	Vorwort	1
2	Data Mesh im Überblick	3
2.1	Was ist Data Mesh?	3
2.2	Was ist Data Mesh nicht?	5
2.3	Warum Data Mesh?	5
2.4	Für wen ist Data Mesh sinnvoll und für wen eher nicht? ..	8
3	Datenanalysen werden selbstverständlicher	11
4	Data Mesh am Beispiel	15
5	Data-Mesh-Architektur	19
5.1	Domänenorientierte Dezentralisierung	20
5.2	Datenprodukte	22
5.3	Datenplattform	27
5.4	Governance	30
5.5	Enabling	31
6	Data Mesh im Vergleich	35
6.1	Data Warehouse	35
6.2	Data Lake	35
6.3	Data Virtualization	36
6.4	Data Fabric	37
7	Fazit	39
8	Unser Angebot	41
	Literaturhinweise	43
	Über die Autor:innen	45

1 Vorwort

Data Mesh ist ein neuer soziotechnischer Ansatz für eine dezentrale Datenarchitektur mit dem Ziel, auch in großen Unternehmen Mehrwert aus Daten zu schöpfen. Der Ansatz basiert auf vier Prinzipien. Das „Domain-Ownership“-Prinzip setzt voraus, dass die Domänenteams die Verantwortung für ihre analytischen Daten übernehmen und annehmen, zusätzlich zur Betriebsverantwortung der operativen Systeme aus der DevOps-Bewegung.

Das Prinzip „Data as a Product“ wendet Product Thinking auf analytische Daten an. Das Domänenteam ist dafür verantwortlich, die Bedürfnisse anderer Domänen durch die Bereitstellung hochwertiger Daten als Datenprodukte zu befriedigen – ganz wie mit öffentlichen APIs.

Das dritte Prinzip, „Self-serve Data Platform“, besteht darin, Platform Thinking auf die Dateninfrastruktur zu übertragen. Ein spezielles Datenplattformteam stellt domänenagnostische Funktionen, Werkzeuge und Systeme für Erstellung und Konsum interoperabler Datenprodukte für alle Domänen bereit.

Das Prinzip „Federated Computational Governance“ stellt organisationsübergreifende Prozesse für Data Governance dar. Hiermit erreicht man die Interoperabilität aller Datenprodukte durch eine Standardisierung, die von einer Governance-Gilde bestimmt wird.

Das Hauptziel ist die Einhaltung der organisatorischen Regeln und der Regularien der Branche. Data Mesh führt die in der Softwareentwicklung gestartete, domänengetriebene Dezentralisierung und Skalierung im Bereich der Daten konsequent fort. Es löst die Abhängigkeit zu zentralen Datenteams auf und erlaubt Domänenteams, autonom Datenanalysen durchzuführen, um mit der Schnelligkeit der Geschäftswelt mithalten zu können.

Dieser Primer ist eine überarbeitete Version des im Oktober 2022 veröffentlichten TDWI eBook „Data Mesh“ [1].

2 Data Mesh im Überblick

In diesem ersten Teil wollen wir Ihnen eine Kurzübersicht über das Was und das Warum hinter Data Mesh geben und darauf eingehen, für wen dieser Ansatz sinnvoll ist und für wen nicht. Danach können Sie Data Mesh grob bewerten und in den Hype-Diskussionen mitreden.

2.1 Was ist Data Mesh?

Data Mesh ist ein soziotechnischer Ansatz für dezentralisierte Datenarchitekturen. Soziotechnisch bedeutet, dass sich organisatorische Strukturen, Menschen und Technologien gegenseitig beeinflussen. Data Mesh wurde für analytische Daten konzipiert. Es fokussiert sich also auf Datenanalysen und Machine Learning und überträgt den Domänenteams die Verantwortung für die Daten aus ihrem fachlichen Bereich. Dabei werden die Teams von einer Datenplattform befähigt, diese Aufgabe effizient und im Self-Service durchzuführen. Obwohl die Datenplattform wichtig ist, steht die Technologie bei Data Mesh nicht im Vordergrund.

Data Mesh überträgt die Ideen der Dezentralisierung, die in der Softwarearchitektur mit Domain Driven Design, Microservices und DevOps Einzug gehalten haben, auf die Datenwelt. Monolithische Strukturen werden dabei aufgebrochen und entlang der Domänen- und Team-Strukturen in unabhängige modulare Einheiten aufgeteilt. Die Domänenteams erhalten dabei die Verantwortung für die Bereitstellung der analytischen Daten aus ihrer Domäne.

Der Begriff Data Mesh wurde erstmals 2018 von Zhamak Dehghani [2] beschrieben und basiert auf folgenden vier aufeinander aufbauenden Prinzipien:

1. Das Prinzip der **Domain Ownership** verlangt, dass die Domänenteams die Verantwortung für ihre Daten übernehmen. Nach diesem Prinzip sollten analytische Daten um Domänen herum aufgebaut sein, ähnlich wie die Teamgrenzen, die sich an den Bounded Context des Systems anpassen. Im Rahmen der domänenorientierten verteilten Architektur wird die Verantwortung für

- die analytischen und operativen Daten nicht mehr vom zentralen Datenteam, sondern von den Domänenteams übernommen.
2. Das Prinzip **Data as a Product** wendet Product Thinking auf analytische Daten an. Es besagt, dass es über die Domäne hinaus Abnehmer für die Daten gibt. Das Domänenteam ist dafür verantwortlich, die Bedürfnisse anderer Domänen durch die Bereitstellung hochwertiger Daten zu befriedigen. Im Grunde sollten Domänendaten wie jede andere öffentliche API behandelt werden.
 3. Die Idee hinter der **Self-serve Data Platform** besteht darin, Platform Thinking auf die Dateninfrastruktur zu übertragen. Ein spezielles Datenplattformteam stellt domänenagnostische Funktionen, Werkzeuge und Systeme für die Erstellung und den Betrieb interoperabler Datenprodukte für alle Domänen bereit.
 4. Das Prinzip der **Federated Computational Governance** stellt die Interoperabilität, Qualität und Sicherheit aller Datenprodukte durch Standardisierung sicher. Vertreter:innen der Domänen und Fachexpert:innen definieren gemeinsam Vorgaben in Form von Policies, die von der Datenplattform automatisiert eingehalten werden.

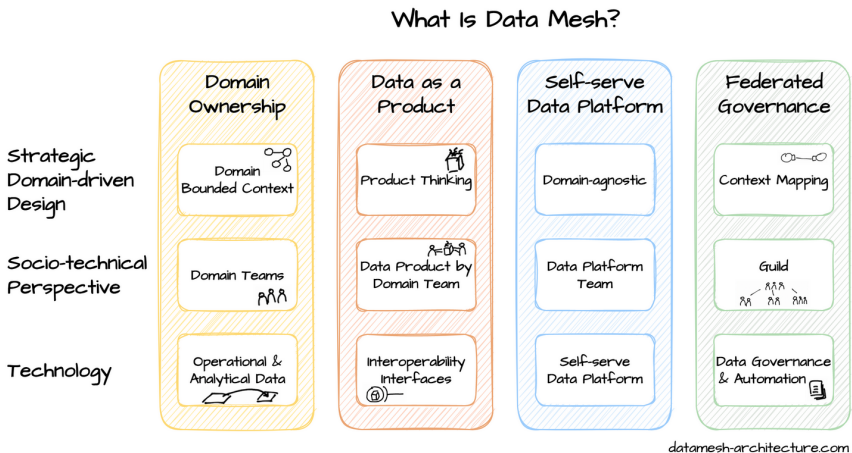


Abbildung 2.1: Was ist Data Mesh? [3]

Zusammengefasst: Data Mesh befähigt Domänenteams, eine datengetriebene Kultur zu entwickeln und datenbasierte Entscheidungen in ihrem Fachbereich selbst zu treffen.

2.2 Was ist Data Mesh nicht?

- Data Mesh ist keine Silver Bullet.
- Data Mesh ist keine Religion.
- Data Mesh ist nicht Plug-and-Play.
- Data Mesh ist kein Produkt, das man einfach kaufen kann.
- Data Mesh ist keine reine Datenplattform.
- Data Mesh kann nicht allein durch das Datenteam eingeführt werden.
- Data Mesh ist kein Konzept für operative Daten.
- Data Mesh ist keine Datenvirtualisierung.
- Data Mesh ist nicht der Nachfolger von Data Warehouse oder Data Lake.
- Data Mesh kann nicht als Big Bang schnell umgesetzt werden.
- Data Mesh ist kein Service Mesh für Daten.
- Und Data Mesh hat rein gar nichts mit Blockchain zu tun.

2.3 Warum Data Mesh?

Viele Unternehmen wollen datengetrieben arbeiten, um fundierte Entscheidungen zu treffen und neue Angebote zu entwickeln. Datenanalyst:innen sollen Erkenntnisse schnell und unkompliziert erzielen, damit das Produktmanagement kundenorientiert Produkte entwickeln und die Unternehmenssteuerung betriebswirtschaftliche Entscheidungen treffen kann. Mit Data Science und Machine Learning sollen neue, intelligente Dienste entstehen. Entwicklungsteams sollen durch Datenanalysen den Nutzen künftiger User Stories bewerten und priorisieren oder neue Features basierend auf ML-Modellen entwickeln können. Einige Beispiele: Lohnt es sich während der Black Week, kostenlosen Versand anzubieten? Würden Kund:innen längere, aber zuverlässigere Versandzeiten akzeptieren? Wie wirkt sich eine UX-Änderung der Produktseite auf das Kaufverhalten und die Retourenquote aus?

Daher haben Unternehmen viel in Datenplattformen und zentrale Datenteams mit Data Analysts, Data Scientists und Data Engineers investiert, in die sie hohe Erwartungen setzen. Nach einigen anfänglichen Quick-Wins mit Leuchtturmprojekten stellt man jedoch häufig fest, dass das zentrale Datenteam zum Engpass wird. Das Team kann die vielen analytischen Fragen des Managements, der Fachbereiche und der Entwicklungsteams nicht schnell genug oder genau genug beantworten. Wenn die fundierten Antworten spät kommen oder fachlich nicht passen, dann gehen Produktteams andere Wege und greifen beispielsweise wieder auf das intuitive Bauchgefühl zurück.

Aber warum wird das zentrale Team zum Engpass? Das zugrunde liegende Problem ist vielseitig. Erstens: Die Bereitschaft der Entwicklungsteams, eigene operative Daten zur Verfügung zu stellen, ist gering. Früher bekam das zentrale Datenteam einfach Zugriff auf eine Oracle-Datenbank, und damit war die Sache erledigt. In einer Microservice-Architektur hat jedes Entwicklungsteam eigene Datenbanken. Diese Teams wollen niemandem Direktzugriff auf ihre operativen Datenbanken geben und haben wenig Interesse daran, eine stabile API für das zentrale Datenteam bereitzustellen. Somit kommen die Datenteams nur schwer an die notwendige Datengrundlage, und ohne Daten können die Anfragen nicht beantwortet werden.

Zweitens: Wenn die Datenteams doch Zugriff auf die Datenbanken erhalten, müssen sie viel Zeit aufwenden, die immer wieder kaputtgehenden ETL-/ELT-Pipelines zu reparieren – wenn zum Beispiel in den Quellsystemen oder Quellnachrichten neue Felder hinzukommen oder Felder entfernt werden. Denn die Datenteams sind typischerweise allein für diese Pipelines zuständig, erfahren aber von Änderungen meist erst durch fehlgeschlagene Pipeline-Runs.

Drittens, und das ist vermutlich der schwerwiegendste Teil: Gute Datenanalysen erfordern fundiertes Wissen über die Fachlichkeit und die Systeme. Datenteams bestehen jedoch typischerweise eher aus technischen Datenexpert:innen und sind zu weit weg vom Business. Das notwendige Wissen müssen die Datenexpert:innen dann mühsam vom Fachbereich und den Entwicklungsteams einsammeln, die jedoch wiederum keine Anreize für einen solchen Wissenstransfer haben. Zusammengefasst: Es besteht ein systematisches Problem. Der zentralistische Ansatz stößt hier an seine Grenzen.

Lassen Sie uns in die operative Welt vor ein paar Jahren zurückblicken. Im Bereich des Betriebs und der Entwicklung von Software gab es fast identische Probleme. Ursprünglich unterhielt der Betrieb den alleinstehenden, riesigen Monolith der Softwareentwicklungsabteilung. Über die Zeit stieg der Druck auf die Software-Entwickler:innen, schneller mehr Software zu liefern. Die Lösung war, dass man die zentrale Softwareentwicklungsabteilung in Teams aufteilte, die ihre eigenen Systeme entwickelten.

Für den Betrieb, der weiterhin zentralistisch aufgebaut war, wurde dies zu einem Problem. Er sollte nun die immer zahlreicher werdenden Anwendungen betreiben und wurde zum Engpass. Die Folge war, dass der Betrieb nicht nur die Lieferung von neuen Features und Anwendungen ausbremste, sondern auch zur Innovationsbremse wurde. Denn neue Technologien wie NoSQL-Datenbanken waren unbekannt, und es gab auch keine Kapazität, das Wissen zu deren Nutzung aufzubauen.

Erst mit dem Kulturwechsel durch DevOps konnte dieser Engpass aufgelöst werden. Die Entwicklungsteams übernahmen die Verantwortung für den Betrieb der von ihnen entwickelten Anwendungen jetzt selbst, ganz nach dem Motto: „You build it, you run it.“ Das zentrale Betriebsteam wurde zu einem Plattformteam, um die Entwicklungsteams zu unterstützen und weiterhin zentrale Komponenten wie Netzwerke zu betreuen.

Letztlich herrschte damals ein Mismatch zwischen dezentralisierten Entwicklungsteams und zentralisiertem Betrieb. Jetzt gibt es diesen Mismatch zwischen dezentralisierten DevOps-Teams und dem zentralisierten Datenteam. Das Datenteam ist mit den vielen agilen, schnellen Projektteams konfrontiert und kann mit deren Veränderungsgeschwindigkeit und Analysebedürfnissen nicht mithalten. Der Engpass ist aufgrund dieses Mismatches systeminhärent. Wie DevOps das Problem bei Entwicklung und Betrieb gelöst hat, so setzt Data Mesh bei der Lösung zwischen Produktentwicklung und Datenanalyse an und führt die Dezentralisierung, die in den Entwicklungsteams vor einiger Zeit angefangen hat, konsequent fort. Wir täten gut daran, aus den Lehren der Einführung von DevOps zu lernen.

2.4 Für wen ist Data Mesh sinnvoll und für wen eher nicht?

Da Data Mesh ein neuer Begriff ist, gibt es bisher kaum Best Practices oder Erfahrungsberichte aus der Industrie. Ob Data Mesh wirklich ein sinnvoller Ansatz für Ihr Unternehmen ist, lässt sich nur mit einer genaueren Analyse feststellen. Aber mit einigen wenigen Fragen erhalten wir schnell eine Vorabindikation. Deswegen präsentieren wir hier eine unvollständige Liste von Heuristiken, die Ihnen helfen kann, einzuschätzen, ob Data Mesh für Ihr Unternehmen sinnvoll oder zumindest nachdenkenswert ist. Wir haben bewusst auf einen Scoring-Ansatz verzichtet, weil so etwas immer seine Schwächen hat, und stattdessen eine unsortierte Liste von Themen mit einer groben Einteilung der Selbsteinschätzungen in die drei Antwortkategorien „eher nicht“, „nachdenkenswert“ oder „sinnvoll“ erarbeitet. Sie bekommen damit ein Gefühl, wie bereit Ihr Unternehmen für Data Mesh ist.

	Selbsteinschätzung		
Aspekt	eher nicht	nachdenkenswert	sinnvoll
Domain Driven Design	Ist unbekannt oder spielt keine Rolle.	Teilweise bekannt und verstanden	Wird von allen verstanden. Es gibt Bounded Contexts und Context Maps.
Vertikalisierung des Software-Systems anhand von Domänen	Nein. Es gibt einen Monolith für die gesamte Fachlichkeit.	Die Transformation zu Microservices hat begonnen.	Das Software-System besteht aus fachlich geschnittenen Microservices.
Anzahl Softwareentwicklungsteams	<= 5	> 5	> 10

	Selbsteinschätzung		
Aspekt	eher nicht	nachdenkenswert	sinnvoll
Make- oder BuyStrategie für Software in den Kerndomänen	Standardsoftware (SAP, SaaS etc.)	Meist Individualsoftware, aber es gibt auch Standardsoftware.	Fast ausschließlich Individualsoftware
Cloud-Computing	Kein Thema, viele Bedenken	Mittendrin in der Cloud Transformation / Hybrid Cloud	Wir sind Cloud-native.
Teamstruktur	Getrennte Frontend-, Backend- und Betriebsteams	Gemischte Teams, bestehend aus Spezialist:innen	Kleine, funktionsübergreifende Teams, unterstützt durch ein Plattformteam
Vertrauen	Micromanagement durch Führungskräfte	Steuerung durch Ziele	Teams und Mitarbeitern wird vertraut, das Richtige zu tun.
DevOps-Kultur	Was ist CI/CD?	Wir etablieren gerade eine DevOps-Kultur.	Wir leben die DevOps-Prinzipien.
Produkt Thinking	Es wird in Projekten gedacht. Roadmaps geben Implementierung vor.	Produktentwicklung wird versucht, scheitert aber am Management.	Autonome Produktteams, die ihre Produkte selbst (weiter-) entwickeln.

	Selbsteinschätzung		
Aspekt	eher nicht	nachdenkenswert	sinnvoll
Wettbewerbsvorteil durch ML und Analytics	Nicht im Fokus	Potenzial ist erkannt und erste Leuchtturmprojekte sind gestartet.	Datengetriebene Produktentwicklung ist stark ausgeprägt.
Zentrales Datenteam als Engpass	Das Datenteam bearbeitet alle Anfragen in einer angemessenen Geschwindigkeit.	Analyse-Ergebnisse dauern oft mehrere Wochen oder Monate.	Product Owner stellen erst gar keine Analyse-Fragen.

Tabelle 2.1: Selbsteinschätzung, ob Data Mesh sinnvoll oder zumindest nachdenkenswert für Ihren Unternehmens- kontext ist. [2]

3 Datenanalysen werden selbstverständlicher

Bei der Entwicklung neuer Anwendungen oder einer technologischen Transformation, beispielsweise zu Microservices, entsteht aus eng getakteten Roadmaps und vorgegebenen Management-Zielen häufig ein hoher Druck, Features konsequent in Form von Apps, Webseiten und APIs fertigzustellen. Die Anwendungen erzeugen im Laufe der Zeit wertvolle Domänenendaten in Form von Datenbankeinträgen oder Domänen-Events. Datenanalysen spielen zu diesem Zeitpunkt aber oft keine Rolle, da sie nicht im Fokus der Roadmaps stehen und Product Owner sowie das Entwicklungsteam mit den notwendigen Daten-Tools und -Methoden normalerweise nicht vertraut sind.

Mit der Zeit kommt es aber immer wieder zu bestimmten analytischen Fragen, etwa, wie hoch der Umsatzverlust bei einem bestimmten Bug war. Bei einem außergewöhnlichen Anstieg der Log-Einträge muss die Ursache ermittelt werden, oder ein Product Owner stellt die Frage, wie häufig ein bestimmtes Feature genutzt wird und ob sich hier Optimierungen lohnen. Diese Analysen müssen vom Entwicklungsteam mangels Alternativen auf der Produktionsdatenbank selbst vorgenommen werden.

Das operative Datenmodell ist für solche Anfragen nicht optimiert, es fehlen die für die jeweilige Fragestellung notwendigen Indizes und man sorgt sich um Auswirkungen auf die Performance der Gesamtanwendung, wenn es zu einem Full-Table-Scan kommt. Solche Datenzugriffe sind auch aus Datenschutz-Aspekten problematisch, da sensible Daten auf der operativen Datenbank normalerweise nicht anonymisiert sind, solange sie noch gebraucht werden. Eine Ausnahmegenehmigung muss her. Das Entwicklungsteam hat erste Anwendungsfälle für Datenanalysen und kann auch gute Antworten geben, aber der Komfort und die Performance sind nicht auf eine regelmäßige, selbstverständliche Anwendung ausgelegt.

Das ändert sich nun mit Data Mesh: Ein Datenplattformteam stellt eine Datenplattform bereit, die von allen Entwicklungsteams leicht verwendet werden kann. Ein Enabling-Team zeigt den Entwicklungsteams, wie sie diese Plattform

verwendet können: Kafka-Nachrichten werden beispielsweise über vordefinierte Kafka-Connectoren in die Datenplattform gestreamt. Das Entwicklungsteam bekommt Zugriff auf seine bekannten Domänendaten und kann mit gewohnten SQL-Abfragen darauf zugreifen. Sensible Daten werden über definierte Tags automatisch anonymisiert und aufbereitet.

Die Datenplattform ist durch eine spaltenbasierte Datenarchitektur optimiert, um große Datenmengen zu verknüpfen und zu kombinieren. Das Entwicklungsteam kann jetzt Hypothesen formulieren oder analytische Fragen stellen, die es auf Basis seiner Domänendaten innerhalb weniger Sekunden beantworten kann, ohne erst mühsam Indizes anlegen zu müssen. Die Plattform stellt auch integrierte Möglichkeiten zur Visualisierung von Daten bereit, um Trends beispielsweise in Bar-Charts grafisch zu erfassen. Die Datenplattform ist einfach zu benutzen und bildet so die Grundlage für immer mehr datenbasierte Entscheidungen. Teams nutzen ihre vertrauten Programmiersprachen (wie SQL) und werden durch das Enabling-Team angeleitet, Datenanalysemethoden anzuwenden.

Die Datenplattform wird mit der Zeit immer intensiver genutzt. Dabei wird deutlich, dass immer öfter auch Daten benachbarter Domänen benötigt werden, etwa Kundendaten, Bestellinformationen oder Informationen zu Retouren. Die Daten liegen zwar meistens in Kafka vor, aber es wäre viel bequemer, direkt per SQL auf eine entsprechend aufbereitete Tabelle zuzugreifen.

Mit Data Mesh werden Datenprodukte eingeführt, also Datensätze, die ähnlich wie APIs für den Zugriff durch andere Teams konzipiert und entsprechend gepflegt werden. Datenprodukte werden durch eine Dokumentation semantisch beschrieben, sodass andere Teams sie ihren Anwendungsfällen entsprechend nutzen können. Diese Daten werden technisch referenziert und müssen nicht erst aufwendig importiert werden. Ein Entwicklungsteam ist bereit, selbst ein Datenprodukt anzubieten, wenn der Mehrwert hinter dem Bereitstellen des Datenprodukts erkennbar ist, wenn die bereitgestellten Datenprodukte auch wirklich genutzt werden und wenn Datenprodukte im Unternehmen den gleichen Stellenwert erhalten wie Features in der UI oder in Services.

Auch Data-Analytics- und Data-Science-Teams, die etwa Reports für das Top-Management oder das Controlling erstellen oder Machine-Learning-Modelle

trainieren, profitieren von Datenprodukten. Durch die klaren Kontrakte bleiben Datenstrukturen stabil und ermöglichen reproduzierbare Reports und Modelle. Durch die Dokumentation werden Daten verständlicher, es kommt zu weniger Fehlinterpretationen. Die Teams stellen ihre Ergebnisse ebenfalls als Datenprodukte zur Verfügung, die innerhalb des Unternehmens die Basis für betriebswirtschaftliche Entscheidungen auf Management-Ebene bilden oder basierend auf Vorhersagen mittels Machine Learning neue, intelligente Dienste ermöglichen. Auch die Organisation der Teams wandelt sich von einer technologisch aufgestellten Datenabteilung hin zu Domänenteams, die langfristig Datendienste für einen Fachbereich bereitstellen.

Data Mesh führt also zum einen zu stärkerer Nutzung von Datenanalysen in Anwendungsentwicklungen, die zu datenbasierten Entscheidungen in der Produktentwicklung führen. Zum anderen führt es bei Datenteams zu einer stärkeren Orientierung an den fachlichen Anforderungen einer Domäne. Beide Entwicklungen liefern einen Mehrwert für das Unternehmen.

4 Data Mesh am Beispiel

Ein Mesh entsteht, wenn Teams die Datenprodukte anderer Domänen nutzen. Die Verwendung von Daten aus vorgelagerten (englisch: upstream) Domänen vereinfacht Datenreferenzen und Lookups (z. B. Preis eines Artikels), während Daten aus nachgelagerten (englisch: downstream) Domänen die Analyse von Effekten ermöglichen, beispielsweise durch A/B-Tests, die die Auswirkungen auf die Conversion-Rate messen. Daten aus mehreren anderen Domänen können aggregiert werden, um umfassende Berichte und neue, innovative Datenprodukte zu erstellen.

Schauen wir uns das vereinfachte E-Commerce-Beispiel in Abbildung 4.1 an. In diesem Beispiel wird ein Online-Shop entlang der Customer Journey, von der Produktsuche über den Checkout bis zum Payment, in Domänen unterteilt. In einem Data Mesh veröffentlichen diese Domänen ihre Daten als Datenprodukte, damit andere auf sie zugreifen können. Die Entwickler:innen führen Analysen an ihren eigenen Daten durch, um ihre operativen Systeme zu verbessern und den Nutzen neuer Funktionen zu bewerten. Sie verwenden die Daten der benachbarten Domänen, um ihre Abfragen zu vereinfachen und Einblicke in die Auswirkungen in nachgelagerten Domänen zu erhalten. Diese Domänendaten können als quellenorientiert (englisch: source-aligned) bezeichnet werden, da die meisten ihrer veröffentlichten Datenprodukte eng mit den in ihren operativen Systemen erzeugten Domänen-Events und Entitäten übereinstimmen.

Bei komplizierten Teilsystemen (englisch: complicated subsystems) kann es effizient sein, wenn sich ein Team ausschließlich auf die Bereitstellung eines Datenprodukts konzentriert, das aus verschiedenen Datenprodukten aus anderen Domänen zusammengesetzt (englisch: aggregated) ist. Ein typisches Beispiel ist eine 360°-Kundensicht, die relevante Daten aus mehreren Domänen enthält, z. B. Kontodaten, Bestellungen, Sendungen, Rechnungen, Retouren, Kontostand und interne Bewertungen.

Eine umfassende 360°-Kundensicht ist im Hinblick auf verschiedene Bounded Contexts schwer zu erstellen, da jeder Bounded Context ein anderes Verständnis von dem Begriff „Kund:in“ hat, und gilt daher als Anti-Pattern. Trotzdem kann es

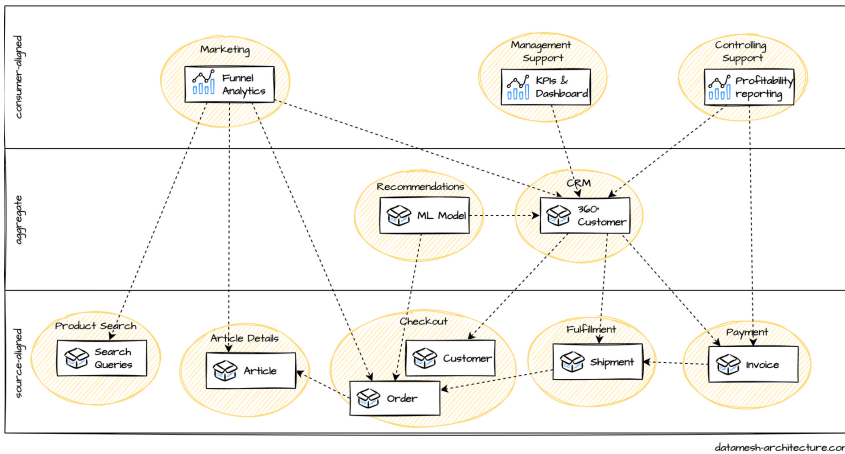


Abbildung 4.1: Beispiel für ein Data Mesh in einem E-Commerce-Unternehmen [3]

für viele Domänen hilfreich sein, darauf zu referenzieren, auch wenn es nicht perfekt ist. Ein weiteres Beispiel für ein kompliziertes Teilsystem ist der Aufbau anspruchsvoller ML-Modelle, die tiefgreifende Data-Science-Fähigkeiten erfordern. Es kann sinnvoll sein, dass ein Team von Data Scientists ein Empfehlungsmodell entwickelt und trainiert, indem es Daten aus der Kasse und der 360°-Kundensicht verwendet, damit ein anderes Team dieses Modell für Empfehlungen im Online-Shop oder in Werbe-E-Mails nutzen kann.

In Unternehmen gibt es auch Geschäftsabteilungen, die Daten aus der gesamten Wertschöpfungskette benötigen, um sinnvolle Entscheidungen treffen zu können, wobei die Mitarbeiter:innen in diesen Abteilungen zwar Geschäftsexpert:innen sind, aber keine Entwickler:innen oder Technologieexpert:innen. Management und Controlling benötigen detaillierte Berichte und KPIs aus allen Domänen, um Stärken und Abweichungen zu identifizieren. Das Marketing führt Funnel- und Webanalysen über alle Schritte der Customer Journey in optimierten Tools wie Google Analytics oder Adobe Analytics durch. In diesen Bereichen ist das Datenmodell für die Bedürfnisse einer bestimmten Abteilung optimiert und kann daher als konsumentenorientiert (englisch: consumer-aligned) bezeichnet werden.

Konsumentenorientierte Berichte waren oft eine der Hauptaufgaben der zentralen Datenteams. Mit Data Mesh konzentrieren sich (neue) konsumentenorientierte Domänenteams auf die Erfüllung des Datenbedarfs einer bestimmten Geschäftsdomäne, wodurch sie tiefes Domänenwissen erlangen und ständig bessere Analyseergebnisse erreichen können. Business und IT wachsen enger zusammen, entweder durch den Aufbau integrierter Domain-Teams oder durch Engineering-Teams, die Domänendaten als Service für das Business bereitstellen, z. B. zur Unterstützung der Unternehmensleitung oder des Controllings. Diese Daten werden in der Regel für eigene Analysen und Berichte verwendet, müssen aber nicht als Datenprodukte für andere Bereiche veröffentlicht und verwaltet werden.

5 Data-Mesh-Architektur

Eine Data-Mesh-Architektur ist ein dezentraler Ansatz, der es Domänenteams ermöglicht, domänenübergreifende Datenanalysen eigenständig durchzuführen. Die Architektur setzt die oben genannten Prinzipien um: Im Mittelpunkt steht die Domäne mit ihrem zuständigen Team und ihren operativen und analytischen Daten. Das Domänenteam erstellt aus den operativen Daten der Domäne analytische Datenmodelle, um seine eigenen Analysen durchzuführen. Auf Basis dieser analytischen Daten veröffentlicht das Team dann wiederum Datenprodukte für andere Domänen, wobei deren Zugriff und damit zusammenhängende Zusicherungen über Datenkontrakte abgebildet werden.

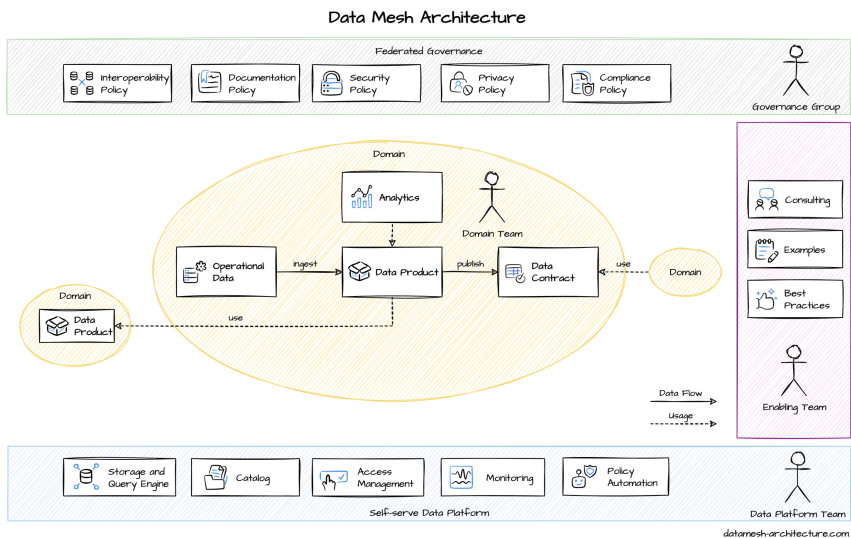


Abbildung 5.1: Data Mesh ist eine soziotechnische Architektur, bei der Domänenteams im Mittelpunkt stehen, die von der Datenplattform und einem Enabling-Team unterstützt werden. [3]

Das Domänenteam einigt sich in einer föderalen Governance-Gruppe mit anderen auf globale Polycys wie Interoperabilitäts-, Sicherheits- und Dokumentationsstandards. Die Gruppe ist häufig als Gilde organisiert und besteht aus Ver-

treter:innen der betroffenen Teams. Sie legen gemeinsam fest, wie sie die im Data Mesh verfügbaren Datenprodukte entdecken, verstehen und nutzen können. Die vom Datenplattformteam zur Verfügung gestellte, domänenagnostische Self-Service-Datenplattform ermöglicht es den Domänenteams, auf einfache Weise ihre eigenen Datenprodukte zu erstellen und ihre eigenen Analysen effektiv durchzuführen. Ein Enabling-Team leitet die Domänenteams bei der Modellierung analytischer Daten, der Nutzung der Datenplattform und der Erstellung und Pflege interoperabler Datenprodukte an.

5.1 Domänenorientierte Dezentralisierung

Die domänenorientierte Dezentralisierung ist der Kerngedanke hinter Data Mesh: Die Verantwortung für datenbasiertes Arbeiten wird nicht länger in einem zentralen Datenteam gesehen, sondern an die jeweiligen Teams verteilt, die am besten mit der zugrunde liegenden Fachlichkeit vertraut sind und deren operativen Systeme die entsprechenden Domänendaten generieren. Data Mesh greift dazu auf die Prinzipien von Domain Driven Design zurück.

Domain Driven Design ist ein Ansatz zur Modellierung von modularen Softwarearchitekturen und orientiert sich an fachlichen Grenzen, den sogenannten Domänen. Dabei wird anerkannt, dass Modelle nur innerhalb ihres begrenzten Bereichs (dem sogenannten Bounded Context) gültig sind. Solche Modelle umfassen insbesondere Begriffe, Geschäftslogik und Datenmodelle. Beispielsweise ist eine „Bestellung“ im Checkout-Prozess etwas anderes als eine „Bestellung“ neuer Ware bei einem Zulieferer. In der Logistik werden Bestellungen anders modelliert (hier ist die Lieferadresse relevant) als in der Finanzbuchhaltung (hier gibt es Kreditoren und Debitoren). Man versucht Domänen so abzugrenzen, dass ein funktionsübergreifendes Domänenteam, bestehend aus Fach- und IT-Expert:innen, eine (Sub-)Domäne eigenständig in Form eines unabhängigen IT-Systems umsetzen kann. Die Domänenteams erhalten dabei große Freiheiten, wie sie die Softwarekomponenten für ihre Domäne umsetzen und welche Features sie weiterentwickeln, denn sie sind am besten mit der Fachlichkeit vertraut und können den

Kundennutzen am besten beurteilen. Technisch wird dies dann beispielsweise in Form von Self-contained Systems umgesetzt.

In den Studien zu Accelerate haben Nicole Forsgren, Jez Humble und Gene Kim herausgefunden, dass Teams dann erfolgreich sind, wenn sie funktionsübergreifend arbeiten, also über alle notwendigen Fähigkeiten zu Entwurf, Entwicklung, Test, Deployment und Betrieb verfügen. Solche Teams können Informationen, die sie im Betrieb sammeln, nutzen, um sofort zu reagieren und die Software entsprechend der Kundenbedürfnisse anzupassen, ohne dass sie einen langwierigen, teamübergreifenden Abstimmungs- und Anforderungsprozess durchlaufen müssen. Eine Veränderung in der Software, wie die Implementierung einer neuen User Story, kann von einem einzigen Team ausgeführt werden.

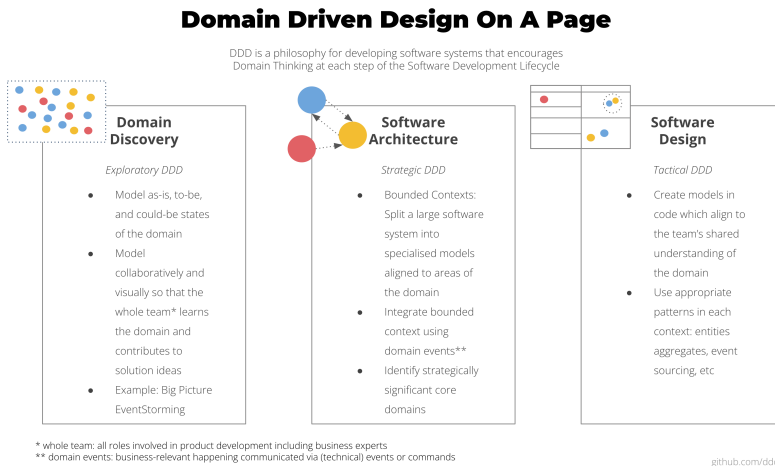


Abbildung 5.2: Domain Driven Design im Überblick. [4]

Data Mesh wendet dieses DevOps-Prinzip auch auf analytische Daten an. Das zuständige Domänenteam erhält auch die Verantwortung für seine analytischen Domänendaten. Es wird befähigt, analytische Auswertungen durchzuführen. Dies führt wieder zu höherer Geschwindigkeit: Das Team ist mit seiner Domäne vertraut und kann daher fachlich relevante Fragen stellen und Hypothesen bilden.

Die vorhandenen Domänenendaten werden genutzt, um Analysen vorzunehmen. Entsprechende Erkenntnisse können innerhalb des Teams sofort zu Änderungen in der Software führen, ein Handover zu einem anderen Team ist nicht nötig. Das vorhandene Domänenwissen im Team ist entscheidend, um aus den Daten semantisch korrekte Aussagen abzuleiten. Die Qualität der Analyseergebnisse wird besser. Man kann von einem DataDevOps-Team sprechen.

Das Domänenteam ist außerdem dafür zuständig, seine Domänenendaten für andere Teams aufzubereiten. Es ist dazu am besten geeignet, da die fachlich passenden Datensätze ausgewählt, aggregiert und semantisch beschrieben werden müssen. Wie das mit Datenprodukten umgesetzt werden kann, lesen wir im nächsten Kapitel.

Die Aufteilung in Domänen erfolgt nach den fachlichen Grenzen und den zuständigen Domänenteams, die auch die operativen Anwendungen umsetzen. Die Aufteilung wird daher nicht ausgehend von einem (bestehenden) analytischen Datenmodell vorgenommen. Auch bei konsumentenorientierten Domänenteams orientiert sich die Domäne an Abteilungsgrenzen der Konsumenten.

5.2 Datenprodukte

Das Prinzip **Data as a Product** wendet Product Thinking auf analytische Daten an. Es bedeutet, dass es nachgelagerte Teams gibt, die diese Datenprodukte konsumieren. Ein Datenprodukt lässt sich als eine autonome, lese-optimierte und standardisierte Dateneinheit definieren, die einen fachlichen Datensatz beinhaltet und die Bedürfnisse des eigenen oder anderer Domänenteams befriedigt.

Datenprodukte können unterschiedlich ausfallen. Wichtig ist, dass jedes Datenprodukt einen Mehrwert für die Konsumenten bietet und dass es überhaupt Konsumenten gibt. Einige Beispiele von Datenprodukten:

- Datenbanktabelle oder View
- ein Datensatz mit Bildern oder Video
- CSV-Dateien
- Dateien im optimierten Format, wie z. B. Parquet oder Delta

- REST API: leseoptimierte Daten aus der Applikation
- Features, die für das Trainieren von Machine-Learning-Modellen relevant sind
- Dashboards

Jedes Datenprodukt gehört zu einer Domäne bzw. einem Bounded Context. Das zuständige Team hat somit eine klare Verantwortung für die analytischen Daten, die es zur Verfügung stellt. Die Daten sind nur in diesem Kontext gültig. Wenn sie von einem anderen Team verwendet werden, muss der Kontext entsprechend berücksichtigt und verstanden werden. Dazu hilft eine semantische Beschreibung aller Felder. Eine gemeinsame Diskussion oder temporäre Kollaboration mit dem Domänenteam kann jedoch sinnvoll sein.

Entsprechend der Product-Thinking-Philosophie sollte jedes Datenprodukt dem Konsumenten einen Nutzen bieten. Ein Datenprodukt kann in Form einer klaren Anforderung von einem anderen Team entwickelt werden, oder man entwirft es für „den Markt“ und stellt dabei Hypothesen zu sinnvollen Anwendungsfällen auf. Nur wenn ein Datenprodukt langfristig genutzt wird und einen Mehrwert bietet, lohnt sich seine Entwicklung und sein Betrieb. Wird ein Datenprodukt nicht genutzt, muss es wieder eingestellt werden. Jedes Datenprodukt sollte also mindestens einem Use Case zugeordnet sein. Es muss auch klar sein, wer die Konsumenten für jedes Datenprodukt sind, welche Qualitätsanforderungen vorliegen und wie intensiv jedes Datenprodukt genutzt wird.

Datenprodukte sollten eine Reihe von Eigenschaften aufweisen, um als solche zu gelten. Die wichtigsten Eigenschaften sind:

- **Auffindbarkeit** (Discoverability) – Andere Teams sollten in der Lage sein, Datenprodukte zu finden und zu verstehen (Understandability), indem man die Datenprodukte ausreichend mit Metadaten anreichert.
- **Interoperabilität** – Datenprodukte müssen so gestaltet werden, dass sie mit anderen Datenprodukten kompatibel sind. Dies führt zur Standardisierung von Datenproduktkomponenten wie Typsystem, IDs und Metadata-Feldern.
- **Vertrauenswürdigkeit** – Diese Eigenschaft repräsentiert die Datenqualität und wird durch Definition von datenproduktspezifischen SLOs realisiert. Solche SLOs können etwa Veränderungsfrequenz der Daten, Vollständigkeit, Lianeage oder Freshness sein.

Um einen strukturierten Prozess des Entwurfs von Datenprodukten zu etablieren, empfehlen wir den von uns entwickelten Data Product Canvas (siehe Abbildung 5.3) zu nutzen. Dieser Canvas ermöglicht es, Datenprodukte im Rahmen eines standardisierten Vorgehens kollaborativ zu entwerfen. Im Folgenden möchten wir den Canvas vorstellen und somit die Struktur des Datenprodukts beleuchten.

Der Data Product Canvas besteht insgesamt aus zehn Blöcken, die ein Datenprodukt definieren: Domain, Data Product Name, Consumer and Use Case, Output Port, Metadata, Input Ports, Data Product Design, Observability, Ubiquitous Language und Classification. Wir beschreiben diese Blöcke in der Reihenfolge, in der wir empfehlen, sie in einem Workshop mit allen Stakeholdern kollaborativ auszufüllen.

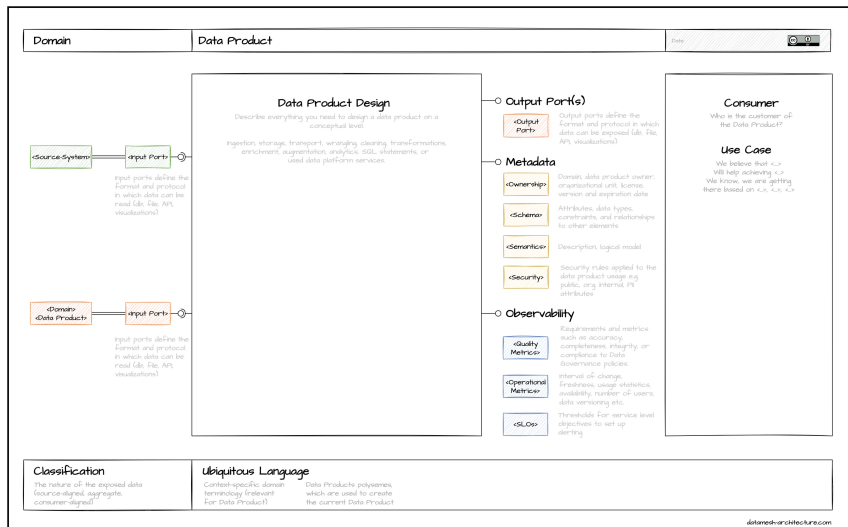


Abbildung 5.3: Data Product Canvas. [3]

Jedes Datenprodukt sollte von einem einzigen Domänenteam implementiert, weiterentwickelt und gepflegt werden. Daher gehört jedes Datenprodukt genau zu einer **Domain**. Die folgenden Fragen sind hierfür relevant:

- Wer ist für das Datenprodukt verantwortlich?
- Wer spezifiziert seine Anforderungen?
- Wer beantwortet Fragen zum Datenprodukt?
- Wer ist für die Wartung des Datenprodukts verantwortlich?

Jedes Datenprodukt hat einen eindeutigen **Namen**, um innerhalb einer Organisation identifiziert und konsumiert werden zu können. In der Regel sollte der Produktname der allgemeinen Benennungsstrategie im Data Mesh folgen.

Die **Consumer Cases** und **Use Cases** liefern die Gründe für die Existenz des Datenprodukts. Die Gestaltung von Datenprodukten folgt der Philosophie des „Product Thinking“. Wir beginnen immer mit den Bedürfnissen der Datenkonsumenten. In diesem Block beschreiben wir analytische Anwendungsfälle und organisatorische Ziele. Das Verständnis der Anwendungsfälle ist entscheidend für die Spezifikation der Daten, die zur Umsetzung der Anwendungsfälle erforderlich sind. Die Konsumenten des zukünftigen Datenprodukts gehören entweder zu unserem eigenen Team oder zu Teams aus anderen Fachdomänen.

Die **Output Ports** definieren das Format und das Nutzungsprotokoll, wie die Daten konsumiert werden können. Das kann etwa eine Datenbanktabelle, eine Datei, eine API oder eine Visualisierung sein.

Neben der Spezifikation des Output Ports sind die Metadaten eines Datenprodukts zu definieren. Neben den allgemeinen **Metadaten** wie Domänenname, Produkteigentümer, Abteilung, Lizenz, Version und voraussichtliches Verfallsdatum des Datenprodukts sind insbesondere das **Schema** mit seinen Datentypen und die **Semantik** der Daten wichtig. Da es einen großen Unterschied macht, ob PII-Attribute Teil der Daten sind und ob die Daten extern oder lediglich firmenintern veröffentlicht werden, sollte der Security Block immer beachtet werden.

Konsumenten eines Datenprodukts müssen den Daten des Datenprodukts vertrauen können. Anders formuliert: Konsumenten haben Anforderungen an die Daten eines Datenprodukts, damit diese einen Nutzen für sie haben. Solche

Qualitätsanforderungen sind Treiber für die Definition von Qualitätsmetriken, die wiederum Teil von Service Level Objectives und Data Contracts werden. Hier sind diese Metriken neben der Spezifikation von Metadaten unter dem Oberbegriff **Observability** zusammengefasst. Dabei geht es um Metriken wie Genauigkeit, Vollständigkeit, Integrität sowie die Einhaltung von Data-Governance-Richtlinien. Operative Metriken können Änderungsintervalle von Datenprodukten, Aktualität, Nutzungsstatistiken, Verfügbarkeit, Anzahl der Konsumenten, Datenversionierung usw. umfassen.

Der Canvas enthält jedoch keine vollständige Liste von Metadaten und Observability-Kriterien. Vergewissern Sie sich, dass Sie alle Informationen definieren, die in Ihrer Organisation erforderlich sind, um eine Data-Mesh-Architektur aufzubauen.

Nachdem das Ergebnis aus Datennutzersicht feststeht, muss festgestellt werden, woher die Daten dafür kommen. Dafür springt man zu den **Input Ports**, welche die Eingangsdaten für das zukünftige Datenprodukt beschreiben. Die Input Ports definieren das Format und das Protokoll, in dem die Eingangsdaten ausgelesen werden können. Wir unterscheiden hier zwischen operativen Quellsystemen und anderen Datenprodukten, die entweder intern gebaut sind oder aus anderen Domänen konsumiert werden.

Die Interna des Datenprodukts, also die Transformationen, werden im Block **Data Product Design** entworfen. Hier werden alle Schritte festgehalten, die aus Daten aus den Input Ports die Daten für die Output Ports transformieren. Man kann es sich wie eine kleine Datenpipeline vorstellen, inklusive Schritten wie Bereinigung von Duplikaten, Aggregation zu höherwertigen Daten, Integration mit anderen Daten durch Joins etc.

Damit ist der Entwurf des Datenprodukts fertig. Während des kollaborativen Entstehungsprozesses können jederzeit die **Classification** festgelegt oder zentrale Fachbegriffe der Domäne im Block **Ubiquitous Language** beschrieben werden.

Nachdem ein Datenprodukt entworfen wurde, muss es natürlich umgesetzt werden. Die Datenplattform bildet dafür die Basis. Mehr dazu im nächsten Kapitel.

5.3 Datenplattform

Die Self-Service-Datenplattform wird von einem Datenplattformteam erstellt und betreut, und zwar für die Domänenteams – und für diese Zielgruppe muss die User Experience passen. Spannend dabei: Die Zielgruppe besteht nicht mehr nur aus Analyst:innen und Data Scientists, sondern oft zu 90 % aus Anwendungsentwickler:innen.

Es gibt (derzeit) noch keine Data-Mesh-Datenplattform, die man einfach kaufen kann. Aber mit den bestehenden Datendiensten von Cloud-Anbietern und den Anbietern moderner Data Stacks kann eine erste Datenplattform für Data Mesh erstellt werden, wie in Abbildung 5.4 am Beispiel von AWS zu sehen ist.

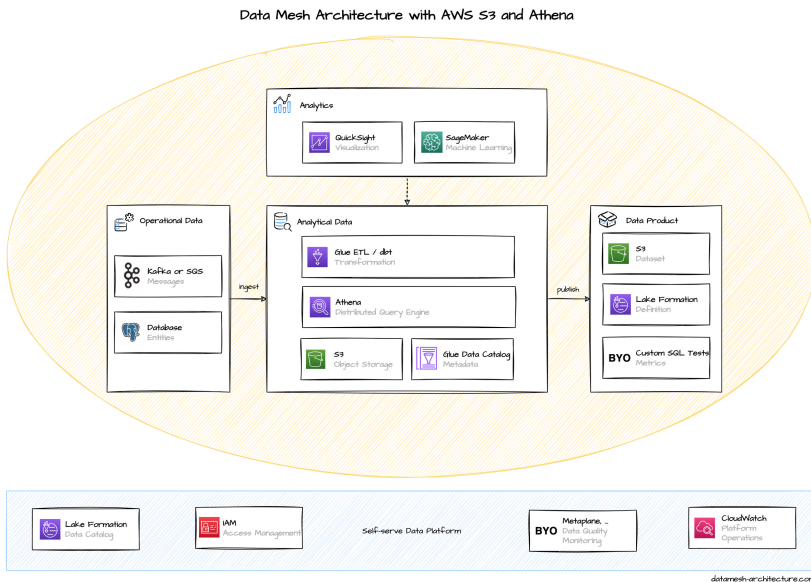


Abbildung 5.4: Die Datenplattform ist aus verschiedenen Komponenten zusammengesetzt; hier am Beispiel von AWS. [3]

Eine Datenplattform umfasst in der Regel mindestens folgende Komponenten:

- Storage & Query-Engine
- Visualisierung
- Datenkatalog
- Access Management
- Monitoring der Datenqualität
- Monitoring der Datenplattform
- Policy Automation

Bei der Storage-Technologie und der damit zusammenhängenden Query-Engine wird auf bestehende Datentechnologien zurückgegriffen.

Beispiele:

- Data Warehouse (Redshift, BigQuery, Snowflake)
- Data Lake (AWS S3, Google Cloud Storage, Azure Data Lake Storage)
- Spark-Engines (Databricks)
- Distributed Query Engines/Datenvirtualisierung (Presto/Trino/Starburst)
- Kafka (Confluent)

Die Storage-Technologie bildet die Grundlage, um operative Daten in einem analytisch optimierten Modell zu speichern. In der Regel wird dazu ein Cloud Data Warehouse oder ein Object Store als Data Lake verwendet. Je nach Software-Architektur kann zudem in Betracht gezogen werden, mit einer Distributed Query Engine virtuell auf operative Daten zuzugreifen, also keinen Import in die Datenplattform vorzunehmen (Datenvirtualisierung). Sobald die Daten in der Datenplattform vorhanden sind, müssen sie aufbereitet werden, also bereinigt, vervollständigt, anonymisiert und ggf. aggregiert, sodass daraus ein Datenprodukt angeboten werden kann. Diese Transformationen werden über SQL-Queries mithilfe von Frameworks wie dbt realisiert, oder als Python-Code, der beispielsweise in einer Spark-Engine ausgeführt wird.

Grundsätzlich kann man sich vorstellen, dass je nach Anforderungen und Vorlieben der Teams mehrere Datenplattformen zum Einsatz kommen. Trotzdem

empfehlte es sich in der Regel, dass sich die Domänenteams auf eine gemeinsame Datenplattform einigen, damit der Zugriff auf Datenprodukte anderer Domänen unkompliziert, effizient und kostengünstig vonstattengeht. Die Daten sollten möglichst im gleichen Cloud-Rechenzentrum liegen, um Egress-Kosten, VPC-Konfigurationen und lange Laufzeiten bei Join-Operationen zu minimieren.

Damit das Team Datenauswertungen vornehmen kann, sind Tools zur Visualisierung nützlich, die es Menschen ermöglichen, leichter Trends zu erkennen und in Form von Dashboards einen schnellen Überblick zu gewinnen. Kollaborative Notebooks werden gerne verwendet, um mit entsprechenden Algorithmen deskriptive und prädiktive Datenanalysen durchzuführen.

In einem gemeinsamen Datenkatalog werden Datenprodukte eingetragen und dokumentiert, sodass andere Teams diese finden, verstehen und nutzen können. Der Datenkatalog kann ein integrierter Bestandteil des Anbieters der Datenplattform sein; alternativ kann auf einen separaten Dienst zurückgegriffen werden.

Bei der Berechtigungsverwaltung greift man in der Regel auf das zugrundeliegende Berechtigungssystem der Storage-Anbieter zurück (IAM-Dienste von AWS, GCP und Azure), das teilweise ebenfalls über den Datenkatalog verwaltet und überwacht werden kann.

Datenprodukte geben Aussagen über eine bestimmte Dienstgüte und Datenqualität. Diese müssen überwacht werden, und das verantwortliche Team muss informiert werden, wenn ein Datenprodukt davon abweicht, wenn beispielsweise ein Datensatz seit mehreren Tagen nicht aktualisiert wurde oder die Anzahl der Daten plötzlich massiv abweicht. Ein entsprechendes Datenqualitäts-Monitoring- und Alerting-System sollte in das bestehende System für die operativen Systeme integriert werden. Auch die Dienste der Datenplattform selbst müssen vom Datenplattformteam ständig überwacht werden.

Es zeigt sich, dass Data Mesh keine neue Datentechnologie ist. Vielmehr steht die Organisation der Dienste im Mittelpunkt: Die Datentechnologie muss den Domänenteams so einfach wie möglich zugänglich gemacht werden. Jedes Team sollte einen eigenen Bereich für seine Daten bekommen. Bei der Gestaltung der Plattform steht der Self-Service-Gedanke im Vordergrund. Kein Team sollte auf

das Plattformteam warten müssen. Stattdessen sollten alle Aktivitäten zu Bereitstellung, Nutzung, Berechtigungen und Überwachung von Datenprodukten vom Team selbstständig vorgenommen werden können.

5.4 Governance

Data Mesh setzt auf ein föderales Governance-Modell: Die Governance-Gruppe besteht aus Vertreter:innen der einzelnen Domänenteams und des Plattformteams, also den Personen, die täglich mit Datenprodukten arbeiten. Sie ziehen Expert:innen hinzu, beispielsweise aus den Bereichen Datenschutz, Sicherheit und Recht (vgl. Abbildung 5.5).

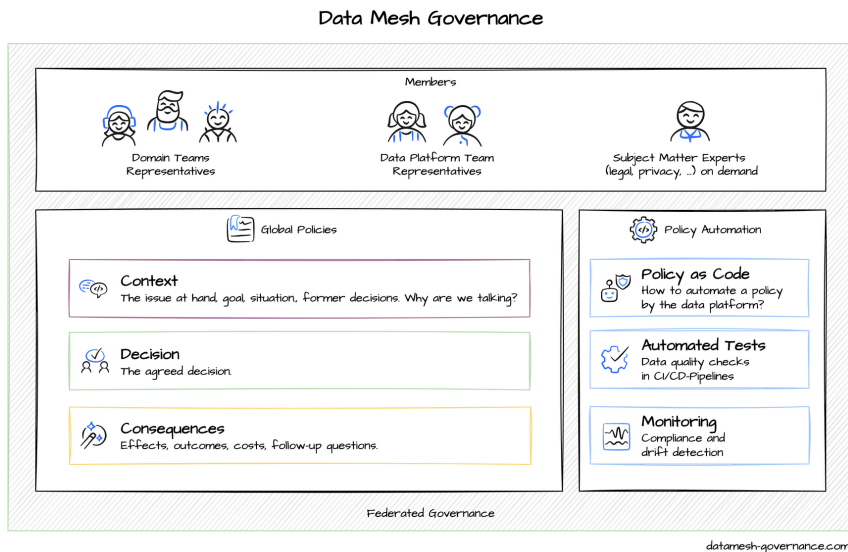


Abbildung 5.5: Struktur und Aufgaben der Governance-Gruppe. [5]

Die Governance-Gruppe beschließt, welche übergreifenden Policies (Regeln und Vereinbarungen) im Mesh gelten sollen, damit Datenprodukte interoperabel und verständlich sind. Außerdem regelt sie, wie die Aspekte rund um Sicherheit und

Datenschutz umgesetzt werden sollen. Die Governance sorgt also für einen Ausgleich zwischen hoher Autonomie der Domänenteams und den notwendigen gemeinsamen Standards, um das Zusammenspiel im Mesh so einfach wie möglich zu gestalten.

Beispiele für solche Policys sind:

- Verwendung des Dateiformats „Delta“ für Datenprodukte
- Definition eines Adressierungsschemas für Domänen und deren Datenprodukte
- Datenprodukte werden in einem bestimmten Datenkatalog registriert.
- Festlegung des Dokumentationsformats der Datenprodukte
- Umgang und Anonymisierungsstrategie von zu schützenden Daten
- Regeln, wie Datensätze nach DSGVO-Vorgaben ggf. gelöscht werden („Recht auf Vergessenwerden“)
- Führendes System für Einwilligungen und Definition der erlaubten Anwendungsfälle

Verantwortlich für die Umsetzung der vereinbarten Policys sind die Domänenteams. Dazu müssen die Teams die Regeln kennen, verstehen, technisch umsetzen und letztlich auch die Einhaltung überwachen.

Um den Aufwand für die Umsetzung und Überwachung der Policys zu minimieren, schlägt Zhamak Dehghani eine Automatisierung (engl: computation) durch die Plattform vor. Policys sollten möglichst als Code formuliert werden. So sollen etwa Berechtigungen anhand von gesetzten Tags automatisiert eingerichtet werden. Automatisierte Tests und Monitoring von Metriken sorgt für die Einhaltung der Datenqualität und Überwachung von Datengarantien. Zum aktuellen Zeitpunkt ist dieser Aspekt noch als experimentell zu betrachten, wird aber mit jedem Team und jedem Datenprodukt im Mesh wichtiger.

5.5 Enabling

Eine erfolgreiche Einführung von Data Mesh in ein Unternehmen ist auch mit einer hervorragenden Datenplattform nicht selbstverständlich. Die Nutzung der Datenplattform und der Data-Mesh Prinzipien muss bei den Domänenteams aktiv

beworben werden, damit diese die Plattform auch wirklich nutzen und hochwertige Datenprodukte bereitstellen. Dabei ist es wichtig, ihnen Hilfestellung geben, wie sie dies effizient tun können.

Hierfür ist ein Data-Mesh-Enabling-Team sinnvoll. Es besteht aus einer Gruppe von Spezialist:innen, die Domänenteams helfen, Wissen aufzubauen, und erklären, wie man für ein bestimmtes Problem eine technische Lösung einsetzt. Sie sind mit der Datenplattform gut vertraut und wissen, welche Datenprodukte existieren. Das Enabling-Team arbeitet in der Regel für einen bestimmten Zeitraum eng mit den Domänenteams zusammen. Es implementiert selbst aber keine technische Lösung, sondern berät lediglich. Aus den Erfahrungen dieser Beratung leitet das Enabling-Team Best Practices für die zukünftige Gestaltung von Datenprodukten ab, dokumentiert das optimale Vorgehen und stellt gegebenenfalls Templates für die Entwicklung und die Deployment-Pipelines von Datenprodukten zur Verfügung.

Ein Enabling-Team senkt die Hürde für Domänenteams, mit analytischen Daten und der Datenplattform zu arbeiten. Es zeigt typische Methoden bei der Datenanalyse, gibt Impulse, wie Domänenteams durch bessere Entscheidungen in ihrem Entwicklungsprozess einen Mehrwert erzielen und wie ihre Anwendungen durch innovative Dienste erweitert werden können.

Der Aufbau eines Enabling-Teams sollte parallel mit der Entwicklung der Datenplattform beginnen. Das Enabling-Team kann sich so frühzeitig mit der Datenplattform vertraut machen und als „Anwalt der Anwendungsentwickler:innen“ mit Anforderungen für eine optimale Gestaltung der Datenplattform sorgen. Ein Enabling-Team besteht aus erfahrenen Software-Entwickler:innen und Datenexpert:innen. Anfangs kann es auch durch externe Berater:innen unterstützt werden.

Zur Einführung vom Data Mesh sollte man mit einem oder zwei Domänenteams starten, die vom Enabling-Team unterstützt werden. Die Domänen sollten so ausgewählt sein, dass sie einen betriebswirtschaftlich relevanten Anwendungsfall für Datenanalysen haben. Das Entwicklungsteam sollte erfahren und bereit sein, sich auf neue Technologien einzulassen und Dinge auszuprobieren. Die erfolgreiche Umsetzung von fachlich relevanten, datenintensiven Anwendungsfällen und der

daraus generierte Mehrwert sind die besten Voraussetzungen, um Data Mesh auf weitere Domänen im Unternehmen auszuweiten.

6 Data Mesh im Vergleich

Data Mesh ist eine dezentralisierte Datenarchitektur für analytische Daten, die dient, um den Herausforderungen der Skalierung analytischer Use Cases zu begegnen. In diesem Umfeld existieren jedoch viele andere, sowohl recht bekannte als auch neuere, Datenarchitekturen. Um Data Mesh besser zu verstehen, hilft es, den Ansatz gegenüber anderen Datenarchitekturen abzugrenzen.

6.1 Data Warehouse

Data Warehouse (oft als DWH abgekürzt) gehört zur ersten Generation der Datenarchitekturen und ist zentralistisch aufgebaut. Die transaktionalen Daten der operativen Systeme (OLTP) fließen über ETL-Pipelines in das integrierte Datenschema des zentralen Data Warehouse und werden so zu analytischen Daten (OLAP). Nutzer:innen können dann auf Basis des integrierten Datenschemas ihre analytischen Fragen beantworten oder Reports erstellen. Dazu werden häufig Business-Intelligence-Systeme eingesetzt. Der Data Mart ist eine spezielle Form des Data Warehouse, die sich nur auf einen Geschäftsbereich oder ein Thema konzentriert, funktioniert aber ansonsten analog.

Data Mesh unterscheidet sich fundamental von Data-Warehouse-Architekturen, denn es setzt auf starke Dezentralisierung und nutzt ganz bewusst kein gemeinsames integriertes Datenschema.

6.2 Data Lake

Als Antwort auf immer schneller wachsende Datenmengen (Big Data) entstand das Konzept des Data Lake, die zweite Generation von Datenarchitekturen. Anstatt wie im Data Warehouse lediglich aufbereitete, integrierte Daten zu speichern, werden in einem Data Lake zunächst alle Rohdaten unabhängig voneinander als Dateien mit einer EL-Pipeline abgelegt. Im Data Lake selbst werden die Rohdaten dann mittels verteilter Datenverarbeitung aufbereitet und integriert. Auf technischer Seite werden die Daten typischerweise in einem optimierten Format wie Avro oder Parquet in Object Stores wie S3 auf AWS auf der Cloud

abgespeichert und dann mittels Map-Reduce-Technologien wie Hadoop verarbeitet. Erstmals sind ein Datenkatalog und ein explizites Schema-Management erforderlich, um die Vielzahl an unterschiedlichen Daten auffindbar und nutzbar zu halten. Der Hauptunterschied zum Data Warehouse ist eine Verlagerung der Transformation (das T im ETL-Prozess) und der Datenmodellierung in die späteren Phasen der Datenverarbeitung, da diese Operationen von den analytischen Use Cases abhängig sind.

Über die Zeit haben sich Data Warehouse und Data Lake einander angenähert, und es entstand das Data Lakehouse, das auf einem Data Lake basiert, jedoch dessen Nachteile mit Konzepten aus dem Data Warehouse ausgleicht. So werden Daten wie im Data Lake als Dateien in einem Object Store persistiert, aber die Datenmanagement- und Datenformatierungsfunktionalitäten, inklusive ACID-Transaktionen, werden dank technischer Innovationen vom Data Warehouse übernommen.

Sowohl bei Data Lake als auch bei Data Lakehouse handelt es sich immer noch um zentralisierte, monolithische und technische Ansätze, in denen ein zentrales Datenteam sich um die EL-Prozesse, die Transformationen und die Plattform kümmert. Mit der steigenden Anzahl von analytischen Anwendungsfällen kommen solche Ansätze an ihre Grenzen. Im Vergleich dazu verlagert Data Mesh ganz bewusst die Verantwortung für die Daten, die EL-Prozesse, die Transformationen und die Analysen in die entsprechenden Domänenteams, die von einer domänenagnostischen Datenplattform unterstützt werden.

6.3 Data Virtualization

Da Unternehmensdaten an unterschiedlichen Orten wie Data Warehouse, relationalen Datenbanken, Data Lake usw. und in unterschiedlichen Formaten wie XML, CSV, JSON, Parquet usw. gespeichert werden, besteht die Herausforderung, diese einheitlich für Analysen zu verwenden. Data Virtualization ist ein Ansatz für den einheitlichen Zugriff auf verteilte und heterogene Datenquellen. Anstatt Daten mit Pipelines physisch zu verarbeiten, führen Datenvirtualisierungssysteme Extraktion, Transformation und Integration von Daten virtuell durch. Während alle Daten in den Quellsystemen verbleiben, wird durch Data Virtualization eine

logische Schicht geschaffen, die den Datenzugriff in Echtzeit ermöglicht. Danach kann mit SQL oder REST API auf diese logische Schicht zugegriffen werden. Erst bei Ausführung der Abfragen wird so auf die Daten in den Quellsystemen zugegriffen, dass genau diese Abfragen bedient werden können. Das Konzept Logical Data Warehouse baut auf Data Virtualization auf und legt ein zentrales, integriertes, logisches Datenmodell darüber, um einer Vielzahl von analytischen Anwendungsfällen gerecht zu werden.

Das Konzept des Logical Data Warehouse steht klar im Widerspruch zum Data Mesh, da es wieder ein zentralisierter Ansatz ist. Die Technik der Data Virtualization könnte man sich aber als Möglichkeit vorstellen, Datenprodukte im Data Mesh zu definieren. Dies ist jedoch schwer mit den Eigenschaften eines Datenprodukts wie der Unveränderbarkeit vereinbar. Durch Data Virtualization gibt man Kontrolle über die Daten an die Quellsysteme ab, und wenn die Daten sich im Quellsystem einfach ändern, ändert sich auch das Datenprodukt.

6.4 Data Fabric

Data Fabric ist ein technischer Ansatz zur Vereinfachung des Datenzugriffs in einer Organisation durch Vereinheitlichung. Ähnlich wie bei Data Virtualization legt Data Fabric eine Abstraktionsschicht über die Daten, die verteilt in heterogenen Systemen liegen, und ermöglicht einen einheitlichen Zugriff, Discovery, Weiterverarbeitung etc. Im Gegensatz zu Data Virtualization stehen bei Data Fabric jedoch die Metadaten im Mittelpunkt. Auf deren Basis lässt sich ein sehr hoher Automatisierungsgrad erreichen, u. a. auch bei der Automatisierung von Data Discovery und Governance.

Data Mesh hingegen setzt stark auf den organisatorischen Aspekt der Dezentralisierung und löst somit ein organisatorisches Problem. Data Virtualization sowie der umfassendere Ansatz von Data Fabric können jedoch für die Umsetzung von den technischen Aspekten von Data Mesh, etwa Plattform oder Governance, verwendet werden. Genaues Augenmerk ist jedoch darauf zu richten, wie die Isolation der Datenprodukte voneinander realisiert wird.

7 Fazit

Data Mesh ist ein neuer Begriff, und dieser soziotechnische Ansatz zeichnet sich vorwiegend durch die organisatorischen Aspekte aus, insbesondere Domain Driven Design, Product Thinking, Team-Autonomie und föderale Governance-Strukturen. Diese sind alle per se nicht neu, stellen zusammen aber einen massiven Unterschied zur aktuellen Realität zentraler Datenteams dar. Daraus ergibt sich die eigentliche Herausforderung: das Change Management bei der Transformation von einer zentralen Datenarchitektur hin zu einer dezentralen Organisation, die skaliert. Dafür sind die volle Unterstützung durch Führungskräfte und ein Enabling-Team notwendig, das diese Transformation begleitet.

Data Mesh ist aber nicht völlig unabhängig von der Technik. Data-Mesh-Architekturen verwenden zurzeit bestehende Technologien wie Data Lake, Cloud und Datenvirtualisierung. In Zukunft werden wir sicherlich dedizierte Data-Mesh-Plattformen von Cloud-Anbietern sehen, die Datenprodukte nativ unterstützen. Ein Plattformteam wählt anforderungsgetrieben die passenden Lösungen aus und stellt diese Dienste als Self-Services für Domänenteams zur Verfügung.

Das Potenzial ist enorm. Dadurch, dass Datenanalysen näher an die Domänenteams gebracht werden, wird datengetriebenes Arbeiten in der Softwareentwicklung selbstverständlicher. Data Mesh verkürzt die Feedback-Loops von einer fachlichen Hypothese bis zur Umsetzung in der Produktion. Analytische Daten bilden zudem die Grundlage für innovative und intelligente Dienste.

Die Einführung von Data Mesh ist ein großes Unterfangen. Aber man kann und sollte klein starten, indem man mit einem oder zwei Teams erste fachliche Anwendungsfälle Ende-zu-Ende umsetzt, dabei Wissen aufbaut und die Plattform entstehen lässt.

8 Unser Angebot

Data Mesh Workshop¹ Gemeinsam klären wir mit Ihnen in diesem 1-Tages-Workshop folgende Fragen: Was ist Data Mesh wirklich? Ist Data Mesh sinnvoll für uns? Sind wir bereit? Wie führen wir Data Mesh ein?

Data Mesh Training² In diesem Training zeigen wir Ihnen, was die vier Prinzipien von Data Mesh bedeuten. Sie lernen die Herausforderungen der Einführung von Data Mesh kennen und bekommen Empfehlungen für ein schrittweises Vorgehen. Wir werden gemeinsam ein Datenprodukt, das zentrale Element in einem Data Mesh, mithilfe unseres Data Product Canvas entwerfen und die Implementierungsalternativen aufzeigen. Am Ende des Workshops können Sie die soziotechnischen Implikationen von Data Mesh bewerten und Datenprodukte entwerfen.

Entscheidung

www.innoq.com

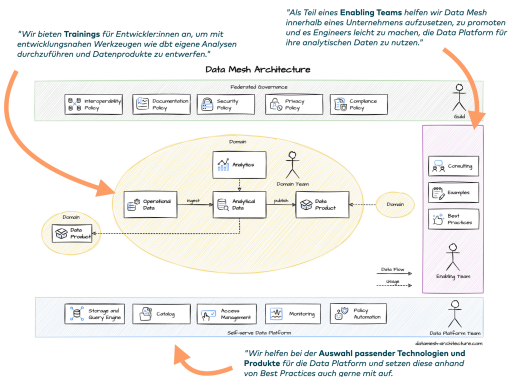
Data Mesh Workshop

INNOQ bietet einen 1-Tages-Workshop zu Data Mesh an, der für dein Unternehmen folgende Fragen klärt: Was ist Data Mesh wirklich? Ist Data Mesh sinnvoll für uns? Sind wir bereit? Und wie führe ich Data Mesh ein?

Wie läuft der Tag ab?

- 1000 Vorstellung und Erwartungen (30min)
- 1030 Was ist Data Mesh wirklich? (30min)
- 1200 Mittagspause (60 min)
- 1300 Ist Data Mesh sinnvoll für uns? Sind wir bereit? (30min)
- 1430 Kaffeepause (30min)
- 1500 Wie führe ich Data Mesh ein? (30min)
- 1630 Office Feedback und Feedback (30min)

Umsetzung



Data Mesh Beratung und Umsetzung Wir beraten Sie bei der Einführung und Umsetzung von Data Mesh ganz individuell. Sei es bei der Gestaltung der Datenplattform, als Teil des Enabling-Teams oder als Facilitators der Data Mesh Governance Group. Sprechen Sie uns an unter **info@innoc.com**.

¹<https://www.innoq.com/de/topics/data-mesh-workshop/>

²<https://www.socreatory.com/de/trainings/datamesh>



Wir beraten ehrlich, denken innovativ und entwickeln leidenschaftlich gern. Das Ergebnis: Erfolgreiche Softwarelösungen, Infrastrukturen und Geschäftsmodelle.

Als Technologieunternehmen fokussieren wir uns auf Strategie- und Technologieberatung, Softwarearchitektur und -entwicklung, Methoden- und Technologietraining sowie Plattform-Infrastrukturen.

Wir unterstützen mit über 180 Mitarbeiter*innen an Standorten in Deutschland und der Schweiz Unternehmen und Organisationen bei der Konzeption und Umsetzung komplexer Vorhaben und der Verbesserung bestehender Softwaresysteme.

Wir engagieren uns in Open-Source-Projekten und dem iSAQB e.V. Wir teilen unser Wissen und unsere Erfahrung auf Konferenzen und Meetups sowie in zahlreichen Büchern und Fachartikeln.

Literaturhinweise

- [1] J. Christ, L. Visengeriyeva, und S. Harrer, „TWDI eBook Data Mesh“. <https://www.tdwi.eu/wissen/studien-buecher/e-books/tdwi-e-book-data-mesh>, 2022.
- [2] Z. Dehghani, *Data Mesh: Data Mesh*. O'Reilly, 2023.
- [3] J. Christ, L. Visengeriyeva, und S. Harrer, „Data Mesh Architecture“. <https://www.datamesh-architecture.com/>, 2022.
- [4] „Welcome to Domain-Driven Design (DDD)“. <https://github.com/ddd-crew/welcome-to-ddd>, 2022.
- [5] J. Christ, L. Visengeriyeva, und S. Harrer, „Data Mesh Governance“. <https://www.datamesh-governance.com/>, 2022.

Über die Autor:innen



Jochen Christ

Jochen Christ ist Senior Consultant bei INNOQ, erfahrener Software-Entwickler und Spezialist für Self-contained Systems und Data Mesh. Er befähigt Teams, nachhaltig erfolgreich Software zu liefern. Jochen ist Maintainer von http-feeds.org, whichjdk.com und Co-Autor von remotemobprogramming.org sowie datamesh-architecture.com.

🐦 [@jochen_christ](https://twitter.com/jochen_christ)



Dr. Simon Harrer

Dr. Simon Harrer ist Senior Consultant bei INNOQ und am liebsten Teil eines kleinen, autonomen Teams, das gemeinsam Probleme löst und dabei voneinander lernt. Sei es als Java-Entwickler in einem Produktteam, als technischer Coach beim Kunden, als Mitautor bei dem Fachbuch „Java by Comparison“ oder als Mitübersetzer von „Data Mesh“.

🐦 [@simonharrer](https://twitter.com/simonharrer)



Dr. Larysa Visengeriyeva

Dr. Larysa Visengeriyeva ist Senior Consultant bei INNOQ und hat im Bereich Augmented Data Quality an der TU Berlin promoviert. Bei INNOQ beschäftigt sie sich mit Datenarchitekturen wie Data Mesh, der Operationalisierung von Machine Learning (MLOps) und Domain Driven Design (DDD).

🐦 [@visenger](https://twitter.com/visenger)

**Data Mesh ist ein sozio-technischer Ansatz,
der es Entwicklungsteams ermöglicht,
Datenanalysen autonom durchzuführen.**

Data Mesh setzt auf Dezentralisierung, die wir bereits aus der modernen Softwareentwicklung mit Domain-driven Design, autonomen, funktionsübergreifenden DevOps-Teams und Microservices kennen, und wendet diese konsequent auf die Welt der Daten und der Analyse an. Data Mesh basiert auf vier Prinzipien, die alle voneinander abhängen: Domain Ownership, Data as a Product, Self-serve Data Platform und Federated Computational Governance. In diesem Primer erklären wir Data Mesh aus der Engineering-Perspektive.

innoq.com