Engineer

Data Mesh

Domänenteams bieten ihre analytischen Daten als Produkte an

Was es ist

Ein Data Mesh ist eine Art, analytische Daten in einer großen Organisation so zu organisieren, dass die Teams jedes Geschäftsbereichs auch die Daten dieses Bereichs verantworten, veröffentlichen und für sie einstehen. Zhamak Dehghani, damals leitende Technologieberaterin bei Thoughtworks, stellte die Idee im Mai 2019 in mehreren Teilen auf Martin Fowlers Website vor. Im Dezember 2020 formulierte sie diese als vier Prinzipien neu: domänenorientierte Verantwortung für Daten und ihre Architektur, Daten als Produkt, Dateninfrastruktur als Self-Service-Plattform und föderierte, automatisiert durchgesetzte Governance. Die vier sollten gemeinsam notwendig und hinreichend sein. Dehghani erwartete, dass sich Praktiken und Werkzeuge ändern, die Prinzipien aber bestehen bleiben.

Beim Mesh geht es um analytische, nicht um operative Daten. Dehghani zieht die Grenze so: Operative Daten liegen in den Datenbanken hinter den Systemen, die den Geschäftsbetrieb abwickeln, sind transaktional und halten den aktuellen Zustand fest. Analytische Daten bilden die Geschichte des Geschäfts über die Zeit ab, aggregiert und modelliert. Auf ihnen beruhen Berichte, und mit ihnen werden Modelle trainiert. Die operativen Daten lässt sie in den jeweiligen Diensten. Das Mesh richtet sie auf die analytische Seite, auf der sie die Reibung bei wachsendem Maßstab sieht.

Domänenverantwortung. Das Mesh folgt den bereits vorhandenen Grenzen der Organisation. Dehghanis durchgängiges Beispiel ist ein digitales Medienunternehmen mit Domänen wie Podcasts und Künstlern. Das Team, das Podcasts betreibt und die Schnittstellen für ihre Veröffentlichung anbietet, soll auch die Historie veröffentlichter Podcasts und ihrer Hörerzahlen über die Zeit bereitstellen. Die Grenze stammt aus Domain-Driven Design. Der Bounded Context jeder Domäne ist dort der Bereich, in dem Änderungen lokal bleiben. Daher eignet er sich auch als Ort für die Datenverantwortung.

Daten als Produkt. Über viele Teams verteilte Daten sind schwerer zu finden, und es ist schwieriger, ihnen zu vertrauen. Jedes Team muss deshalb seine geteilten Daten als Produkt und deren Nutzer als Kunden behandeln. Der Beitrag von 2019 nennt die Anforderungen: Ein Datenprodukt soll auffindbar, adressierbar, vertrauenswürdig, selbstbeschreibend, nach globalen Standards interoperabel und durch globale Zugriffskontrolle gesichert sein. Ein Data Product Owner der Domäne verantwortet Messgrößen wie Qualität, die Wartezeit der Datennutzer und ihre Zufriedenheit. Die Qualitätsverantwortung wandert so nah wie möglich an die Quelle. In der Architektur ist ein Datenprodukt die kleinste unabhängig bereitstellbare Einheit. Es umfasst drei Teile: den Code, der Daten transformiert und bereitstellt und ihre Richtlinien durchsetzt; die Daten samt Metadaten; und die Infrastruktur, auf der es läuft.

Eine Self-Service-Plattform. Ein Datenprodukt aufzubauen, zu betreiben und zu überwachen verlangt spezialisierte Infrastrukturkenntnisse, die sich nicht in jede Domäne kopieren lassen. Ein Plattformteam bietet deshalb Werkzeuge auf einer höheren Abstraktionsebene an, etwa für Speicher, Produktschemas, deklarierte Pipelines und Datenherkunft. So kann ein Entwickler mit breitem fachlichem Profil in einer Domäne ein Datenprodukt ohne dieses Spezialwissen bauen.

Föderierte, automatisiert durchgesetzte Governance. Die Produkte müssen weiterhin zusammenpassen, sonst kann sie niemand kombinieren. Eine Gruppe aus Verantwortlichen für Domänenprodukte und Plattformen entscheidet, welche Regeln überall gelten und welche jede Domäne selbst bestimmt. Die Plattform setzt die globalen Regeln automatisch durch. Dehghanis Beispiel sind Hörerzahlen: Wie die Podcastdomäne sie modelliert, entscheidet sie selbst. Wie ein Hörer identifiziert wird, ist dagegen eine globale Frage, weil dieselbe Person auch in anderen Domänen als Nutzer auftritt.

Das Konzept für eine Entscheidung nutzen

Nutzen Sie „Was ist ein Data Mesh?“ für eine konkrete Entscheidung. Wir können helfen, das Konzept mit dem System in Verbindung zu bringen, das Sie entwickeln oder einkaufen. Vertriebs-, Betriebs- und Finanzdaten werden zusammengeführt, damit zwei Abteilungen nicht länger mit zwei Antworten in dieselbe Besprechung kommen.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Annahmen hinter dem Begriff prüfen

Nutzen Sie die Erklärung und die verlinkten Arbeiten, um Fragen für Ihr Projekt zu formulieren. Wir können die Annahmen gemeinsam mit Ihnen prüfen, bevor ein Werkzeug ausgewählt wird.
Geografische Datenpipeline 50-mal schneller — 50x schnellere Geodaten-Pipelines

Geografische Datenpipeline 50-mal schneller Kontakt aufnehmen →

Warum es wichtig ist

Dehghanis Argument beginnt bei den Problemen eines zentralen Data Lakes oder Warehouses in einem großen Unternehmen. Daten aller Domänen fließen in eine Plattform, die sie vollständig verantwortet. Ein Team von Spezialisten für Datenwerkzeuge betreibt sie, getrennt von den Menschen, die die Daten erzeugen, und denen, die sie nutzen. Sie nennt drei Fehlermuster. Die Plattform ist zentralisiert und monolithisch und kommt daher mit immer mehr Quellen und Datennutzern schwer zurecht. Sie ist in Pipelinestufen wie Aufnahme, Bereinigung, Aggregation und Bereitstellung gegliedert. Eine einzige neue Kennzahl, etwa wie oft Podcasts abgespielt werden, verlangt deshalb eine Änderung in jeder Stufe und eine teamübergreifend koordinierte Veröffentlichung. Und die Verantwortung liegt in hoch spezialisierten Silos: Die Entwickler, die die Daten verstehen müssen, wissen wenig über deren Herkunftsdomänen, während die erzeugenden Teams keinen Anreiz haben, sie richtig zu liefern.

Das Mesh ändert, wer bereinigt. Im alten Modell kommen Daten mit ihren Fehlern an, und ein zentrales Team bereinigt sie nach der Aufnahme. Im Mesh wird der Datenbestand einer Domäne erst zum Datenprodukt, nachdem die Domäne selbst ihn anhand der vereinbarten Qualitätsmaße und globalen Regeln geprüft hat. Wer weiß, was ein Feld bedeuten soll, steht dafür ein.

Auch die Governance verändert ihre Form. Der Beitrag von 2020 stellt beide Modelle in einer Tabelle gegenüber. Ein zentrales Team verantwortet Qualität, Sicherheit und Compliance selbst. Ein föderiertes Team definiert, wie Qualität modelliert wird und welche Sicherheits- und regulatorischen Regeln die Plattform einbauen und überwachen soll. Früher wurde Erfolg an der Zahl der verwalteten Tabellen gemessen. Im Mesh zählt, wie stark die Produkte tatsächlich genutzt werden. Manuelle Prüfungen, die Fehler verhindern sollen, weichen automatisierten Prozessen, die sie erkennen und sich davon erholen.

All das wird nicht als allgemeingültige Antwort angeboten. Im Beitrag von 2019 räumt Dehghani ein, dass das zentrale Modell für Organisationen mit einer einfacheren Domäne und weniger Arten von Datennutzern funktionieren kann. Sie beschreibt sein Scheitern bei Unternehmen mit vielfältigen Domänen, vielen Quellen und unterschiedlichen Nutzern. Ob ein Mesh seinen Zusatzaufwand lohnt, hängt davon ab, zu welcher Gruppe eine Organisation gehört.

Wie es in der Praxis aussieht

Ein echtes Mesh umfasst viele Teams und viele Speicher. Die folgende Skizze beschränkt sich auf eine PostgreSQL-Datenbank, damit sich jede Anweisung im Handbuch prüfen lässt. Sie zeigt die beiden kleinsten Schritte: Eine Domäne veröffentlicht ein Produkt, und die Plattform setzt eine globale Regel durch.

Das Bestellteam verantwortet die operativen Tabellen im Schema `orders`. Es veröffentlicht seine analytischen Daten in einem eigenen Schema als versionierte Sicht mit Beschreibung und ausdrücklicher Zugriffsfreigabe:

```
-- the orders team owns this schema and everything in it
CREATE SCHEMA order_data;

CREATE VIEW order_data.orders_by_day_v1 AS
SELECT order_date, currency, count(*) AS orders, sum(total) AS revenue
FROM orders.orders
WHERE status = 'paid'
GROUP BY order_date, currency;

COMMENT ON VIEW order_data.orders_by_day_v1 IS
  'Paid orders per day and currency. Owner: orders team.';

GRANT USAGE ON SCHEMA order_data TO analysts;
GRANT SELECT ON order_data.orders_by_day_v1 TO analysts;
```

Das PostgreSQL-Handbuch erklärt, warum das auf der Nutzerseite genügt: Wer eine Sicht verwendet, braucht das USAGE-Recht nur auf dem Schema, das die Sicht enthält, nicht auf den Schemas, aus denen sie liest. Die Analysten können das Produkt abfragen, während die dahinterliegenden operativen Tabellen für sie gesperrt bleiben. Evans zählt Datenbankschemas seinerseits zu den physischen Formen, die die Grenze eines Bounded Context annehmen kann.

Auf Plattformseite steht eine Regel, die für jede Domäne gilt, als Abfrage formuliert und bei jeder Veröffentlichung ausgeführt: Kein Produktschema darf eine Sicht ohne Beschreibung enthalten. `obj_description` liefert den gespeicherten Kommentar eines Objekts zurück. Ein `relkind` von `v` kennzeichnet in `pg_class` eine Sicht:

```
-- global rule: every published view carries a description
SELECT n.nspname, c.relname
FROM pg_class AS c
```

```
JOIN pg_namespace AS n ON n.oid = c.relnamespace
WHERE c.relkind = 'v'
  AND n.nspname ~ '_data$'
  AND obj_description(c.oid, 'pg_class') IS NULL;
-- any row returned stops the release
```

Diese Aufteilung zeigt den föderierten Teil im Kleinen. Was als bezahlte Bestellung gilt, entscheidet das Bestellteam. Dass sich jedes Produkt selbst beschreibt und Produktschemas auf `_data` enden, sind globale Konventionen, die die Plattform prüft, ohne dass sich jemand daran erinnern muss.

Die Leistungsseite Data Governance und Qualität verspricht, dass klar wird, wer jede Zahl verantwortet und woher sie stammt. Genau das muss ein Data Product Owner sagen können. Zu den dort aufgeführten umgesetzten Arbeiten gehört eine funktionsorientierte PostgreSQL-Datenschicht: Lese- und Schreibzugriffe laufen durch Funktionen, auf die die Datenbank Zugriff gewähren kann, nie direkt durch die Tabellen. Dahinter steht derselbe Gedanke wie bei der Sicht oben: Datennutzer erhalten eine Schnittstelle mit Zugriffsfreigabe, nicht die darunterliegenden Tabellen.

Die Fallstudie Ein Infrastrukturplan für 14 Abteilungen bezeichnet sich nicht als Data Mesh, trifft aber auf das Spannungsfeld seiner Governance-Gruppe. An einem Forschungsinstitut arbeiteten 14 Forschungsgruppen mit unterschiedlichen Datenquellen, Formaten, Maßstäben und Werkzeugen. Ihre technischen Fähigkeiten und IT-Ressourcen unterschieden sich stark: Einige hatten eigene Lösungen gebaut, viele hatten Schwierigkeiten. Der Plan sollte Forschern ermöglichen, sich auf die Forschung zu konzentrieren. Er balancierte Flexibilität und Standardisierung durch eine modulare Architektur, Zugangswege für sehr unterschiedliche Datenquellen, Schnittstellen für verschiedene Kenntnisstände, skalierbaren Speicher und skalierbare Verarbeitung sowie Strategien für Einführung, Unterstützung und Governance. Die Entscheidung, was jede Gruppe für sich behält und was alle gemeinsam nutzen, überträgt Dehghani der föderierten Governance. Von einer Self-Service-Plattform verlangt sie, dass eine Gruppe mit weniger Spezialwissen auskommt.

Wo es schiefgeht

Ein zentrales Team mit neuem Namen. Das Datenplattformteam Mesh zu nennen ändert nichts, wenn dasselbe Team weiterhin die Rohdaten jeder Domäne aufnimmt, bereinigt und ihre Qualität verantwortet. Dehghanis Governance-Tabelle liefert eine brauchbare Checkliste: Wer verantwortet

Qualität, wer entscheidet die Sicherheitsregeln und wird Erfolg noch in verwalteten Tabellen gezählt? Verlagern Sie die Verantwortung für jedes Datenprodukt in die erzeugende Domäne. Behalten Sie das zentrale Team für Plattform und Regeln bei.

Produkte, für die niemand einsteht. Eine Domäne, die Tabellen ohne Verantwortlichen, Beschreibung oder Zusage zur Aktualität exportiert, hat das alte Problem nur in ihre eigene Ecke verschoben. Dehghani verlangt, dass jedes Datenprodukt seine Qualität als Service Level Objectives angibt, etwa für rechtzeitige Bereitstellung und Fehlerraten. Eine Domäne kann demnach zwei Produkte derselben Ereignisse anbieten: eines nahezu in Echtzeit, das Ereignisse auslassen oder wiederholen kann, und ein später eintreffendes genaueres. Benennen Sie für jedes Produkt einen Verantwortlichen, veröffentlichen Sie seine Ziele und prüfen Sie Qualität dort, wo die Daten entstehen.

Keine gemeinsamen Identifikatoren. Identifiziert jede Domäne dieselben Dinge auf eigene Art, lassen sich keine zwei Produkte verbinden. Im Beispiel von 2019 sieht ein Künstler in der Domäne, die Wiedergaben erfasst, anders aus als in der, die Künstler bezahlt. Einigen Sie sich global darauf, wie solche gemeinsamen Entitäten identifiziert werden, die Dehghani Polyseme nennt. Überlassen Sie den Rest jedes Modells seiner Domäne. Feldformate, Adresskonventionen und gemeinsame Metadatenfelder gehören auf dieselbe globale Liste.

Jede Domäne baut den ganzen Stack neu. Ohne Self-Service-Plattform muss jede Domäne eigenen Speicher, eigene Pipelines und eigenes Monitoring bereitstellen. Das Spezialwissen, das in einem zentralen Team knapp war, wird nun in jedem Team benötigt. Bauen Sie die Plattform, bevor Sie die Domänen zur Veröffentlichung auffordern. Messen Sie sie daran, wie wenig Spezialwissen eine Domäne braucht, um ein Produkt auszuliefern.

Ein genutztes Produkt wird inkompatibel geändert. Ein Datenprodukt ist eine Schnittstelle. Es an Ort und Stelle zu ändern, beschädigt die darauf aufbauenden Berichte. In PostgreSQL verhindert `CREATE OR REPLACE VIEW` einen Teil dieses Fehlers: Die neue Abfrage muss dieselben Spalten mit denselben Namen, derselben Reihenfolge und denselben Datentypen erzeugen. Sie darf nur am Ende Spalten ergänzen. Eine Bedeutungsänderung erkennt das nicht, denn die Berechnung hinter einer Spalte kann völlig anders sein. Bringen Sie jede Änderung, die eine Spalte umbenennt, löscht oder neu definiert, in einer neuen Sicht `orders_by_day_v2` neben der alten unter. Nehmen Sie die alte außer Betrieb, sobald ihre Nutzer umgestiegen sind.

Ein Mesh, wo keines nötig ist. Vier Prinzipien, ein Plattformteam, eine föderierte Governance-Gruppe und ein Product Owner pro Domäne sind viel organisatorischer Aufwand. Gegen Ende räumt Dehghanis Beitrag von 2019 ein, dass „die Einführung im Unternehmensmaßstab noch einen weiten Weg vor sich hat“. Genügen wenige Quellen und ein Team, behalten Sie einen gut geführten zentralen Speicher mit einem benannten Verantwortlichen für jede Tabelle. Übernehmen Sie die Teile, die sich bei jeder Größe lohnen: Verantwortliche, beschriebene Datenbestände und maschinell geprüfte Regeln.

Weiterführende Lektüre

- Data-Mesh-Prinzipien und logische Architektur — Zhamak Dehghani, 2020: die vier Prinzipien, Bestandteile eines Datenprodukts und die Gegenüberstellung zentraler und föderierter Governance.
- Vom monolithischen Data Lake zum verteilten Data Mesh — der ursprüngliche Beitrag vom Mai 2019 mit drei Fehlermustern einer zentralen Datenplattform und den nötigen Eigenschaften eines Datenprodukts.
- Domain-Driven Design Reference — Eric Evans’ Zusammenfassung der Muster, einschließlich des Bounded Context, an dessen Grenzen ein Mesh Verantwortung aufteilt.
- PostgreSQL: CREATE VIEW — die Regeln zum Ersetzen einer Sicht und die benötigten Nutzerrechte, wie im Beispiel verwendet.
- PostgreSQL: COMMENT — wie eine Beschreibung an einem Datenbankobjekt gespeichert wird, die `obj_description` dann wieder ausliest.

Glossar

Für Ihre tatsächlichen Rahmenbedingungen auswählen

Bringen Sie einen repräsentativen Datensatz, die Fragen, die er beantworten muss, und den aktuellen Engpass mit. Wir können vereinbaren, wie Qualität, Durchsatz und Wiederherstellung gemessen werden.

Preise ansehen Kontakt aufnehmen →

Die Begriffe praktisch nutzbar machen

Fragen Sie, wie „Was ist ein Data Mesh?“ auf Ihre Organisation zutrifft. Ein konkretes Beispiel aus Ihrem bestehenden System reicht aus, um das Gespräch zu beginnen.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/glossary/data-mesh/