Engineer

Data mesh

equipos de dominio sirven sus datos analíticos como producto

Qué es

Un data mesh es una forma de organizar los datos analíticos de una organización grande para que los equipos que gestionan cada parte del negocio también se hagan responsables de los datos que esa parte produce, los publiquen y respondan por ellos. Zhamak Dehghani, entonces consultora principal de tecnología en Thoughtworks, expuso la idea en el sitio de Martin Fowler en una serie de entregas en mayo de 2019 y la reformuló en diciembre de 2020 como cuatro principios: responsabilidad orientada a dominios sobre los datos y su arquitectura, datos como producto, infraestructura de datos de autoservicio como plataforma y gobernanza computacional federada. Consideraba que los cuatro eran necesarios y suficientes en conjunto, y esperaba que las prácticas y las herramientas cambiaran mientras los principios permanecían.

El mesh trata de datos analíticos, no de datos operativos. Dehghani establece así la distinción. Los datos operativos están en las bases de datos de los sistemas que hacen funcionar el negocio, son transaccionales y contienen el estado actual. Los datos analíticos son el historial del negocio a lo largo del tiempo, agregado y modelado, y se utilizan para elaborar informes y entrenar modelos. Deja los datos operativos donde están, dentro de cada servicio, y dirige el mesh al ámbito analítico, donde sitúa las dificultades al aumentar la escala.

Responsabilidad por dominio. El mesh se divide siguiendo límites que la organización ya tiene. El ejemplo recurrente de Dehghani es una empresa de medios digitales con dominios como pódcast y artistas. El equipo que gestiona los pódcast y ofrece las interfaces para publicarlos también debería proporcionar el historial de pódcast publicados y el número de personas que los escucharon a lo largo del tiempo. El límite procede del diseño guiado por el dominio, donde el contexto delimitado de cada dominio es el ámbito dentro del cual los cambios permanecen locales, por lo que también es un buen lugar para situar la responsabilidad por los datos.

Datos como producto. Repartir los datos entre muchos equipos dificulta encontrarlos y confiar en ellos, así que cada equipo tiene que tratar los datos que comparte como un producto y a las personas que los utilizan como clientes. El artículo de 2019 enumera lo que eso exige: un producto de datos debe poder descubrirse y localizarse mediante una dirección, ser fiable, describirse a sí mismo, ser interoperable bajo estándares globales y estar protegido por un control de acceso global. Un responsable de producto de datos del dominio responde por medidas como la calidad, cuánto esperan los consumidores para disponer de los datos y su grado de satisfacción. La responsabilidad por la calidad se desplaza hacia el origen, lo más cerca posible de la fuente. En la arquitectura, un producto de datos es la unidad más pequeña que puede desplegarse de forma independiente e incorpora tres partes: el código que transforma y sirve los datos y aplica sus políticas, los datos junto con sus metadatos y la infraestructura que lo ejecuta.

Una plataforma de autoservicio. Construir, ejecutar y supervisar un producto de datos requiere conocimientos especializados de infraestructura que no pueden reproducirse en todos los dominios. Por eso, un equipo de plataforma ofrece herramientas de un nivel de abstracción mayor que abarcan cuestiones como almacenamiento, esquemas de producto, flujos de procesamiento declarados y linaje, para que un desarrollador generalista de un dominio pueda construir un producto de datos sin esos conocimientos especializados.

Gobernanza computacional federada. Los productos tienen que seguir encajando entre sí, o nadie podrá combinarlos. Un grupo de responsables de productos de dominio y de plataforma decide qué reglas se aplican en todas partes y cuáles se dejan a cada dominio, y la plataforma aplica automáticamente las globales. El ejemplo de Dehghani son las cifras de escucha: cómo las modela el dominio de pódcast es asunto suyo, pero cómo se identifica a un oyente es una cuestión global, porque la misma persona también aparece como usuario en otros dominios.

Utilice el concepto en una decisión

Aplique «¿Qué es un data mesh?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Cifras de ventas, operaciones y finanzas reunidas para que dos departamentos no lleven dos respuestas distintas a la misma reunión.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Flujo de datos geográficos 50 veces más rápido — 50x más rapidez en los flujos de procesamiento de datos geográficos

Flujo de datos geográficos 50 veces más rápido Póngase en contacto →

Por qué importa

El planteamiento de Dehghani parte de lo que falla cuando una empresa grande utiliza un lago o un almacén de datos central. Los datos de todos los dominios se llevan a una plataforma que se hace responsable de todos ellos, gestionada por un equipo de especialistas en herramientas de datos separado tanto de quienes los crean como de quienes los utilizan. Identifica tres formas de fallo. La plataforma es centralizada y monolítica, por lo que tiene dificultades cuando se multiplican las fuentes y los consumidores. Está dividida en etapas de procesamiento, como ingesta, limpieza, agregación y servicio de datos, por lo que una única cifra nueva, como la frecuencia con que se reproducen los pódcast, necesita un cambio en cada etapa y un lanzamiento coordinado entre equipos. Y su responsabilidad está compartimentada e hiperespecializada: los ingenieros que deben entender los datos saben poco de los dominios de los que proceden, mientras que los equipos que los producen no tienen incentivos para hacerlo bien.

El mesh cambia quién se encarga de la limpieza. En el modelo antiguo, los datos llegan con sus errores y un equipo central los limpia después de la ingesta. En un mesh, el conjunto de datos de un dominio solo se convierte en producto de datos después de que el propio dominio lo haya comprobado frente a las medidas de calidad acordadas y las reglas globales. Las personas que saben qué debería significar un campo son quienes responden por él.

La gobernanza también cambia de forma. El artículo de 2020 compara ambos modelos en una tabla. Un equipo central es responsable directo de la calidad, la seguridad y el cumplimiento; un equipo federado define cómo se modela la calidad y qué reglas de seguridad y normativas debe incorporar y supervisar la plataforma. Antes se medía el éxito por el número de tablas bajo gobernanza; en un mesh se mide por cuánto se consumen realmente los productos. La revisión manual que intenta prevenir errores da paso a procesos automatizados que los detectan y se recuperan de ellos.

Nada de esto se presenta como una respuesta universal. En el artículo de 2019, Dehghani reconoce que el modelo central puede funcionar en organizaciones con un dominio más sencillo y menos tipos de consumidor, y afirma que falla en

empresas con dominios complejos, muchas fuentes y consumidores diversos. Que la sobrecarga de un mesh compense depende de a cuál de esos dos tipos pertenezca la organización.

Cómo se aplica en la práctica

Un mesh real abarca muchos equipos y muchos almacenamientos. El siguiente esquema se limita a una base de datos PostgreSQL, para poder contrastar cada sentencia con el manual, y muestra los dos pasos más pequeños: un dominio que publica un producto y una plataforma que aplica una regla global.

El equipo de pedidos se hace responsable de las tablas operativas del esquema `orders`. Publica sus datos analíticos en un esquema propio, como una vista con versión, una descripción y una concesión explícita de permisos:

```
-- 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;
```

El manual de PostgreSQL explica por qué esto basta para quien consume los datos: para utilizar una vista, el privilegio USAGE solo es necesario sobre el esquema que la contiene, no sobre los esquemas de los que lee. Los analistas pueden consultar el producto mientras las tablas operativas que hay detrás siguen cerradas para ellos. Evans, por su parte, enumera los esquemas de base de datos entre las formas físicas que puede adoptar el límite de un contexto delimitado.

La parte de la plataforma es una regla aplicable a todos los dominios, escrita como consulta y ejecutada en cada lanzamiento: ningún esquema de producto puede contener una vista sin descripción. `obj_description` devuelve el

comentario almacenado sobre un objeto, y un `relkind` de `v` identifica una vista en `pg_class`:

```
-- 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
```

Esa separación representa la parte federada en miniatura. Qué cuenta como pedido pagado es decisión del equipo de pedidos. Que cada producto se describa a sí mismo y que los esquemas de producto terminen en `_data` son convenciones globales que la plataforma comprueba sin que nadie tenga que recordarlas.

La página del servicio de gobernanza y calidad de los datos promete que quedará claro quién responde por cada cifra y cuál es su fuente, justo lo que debe poder explicar un responsable de producto de datos. Entre los trabajos entregados enumera una capa de datos PostgreSQL basada en funciones, diseñada para que las lecturas y escrituras pasen por funciones a las que la base de datos puede conceder acceso, nunca por las propias tablas. Es el mismo criterio que hay detrás de la vista anterior: los consumidores reciben una interfaz con permisos concedidos, no las tablas que hay debajo.

El caso práctico de la cartera de trabajos Un plan de infraestructura para 14 departamentos no se presenta como un data mesh, pero se enfrenta a la tensión con la que convive el grupo de gobernanza de un mesh. En un instituto de investigación, 14 grupos trabajaban con distintas fuentes, formatos, escalas y herramientas de datos, y sus conocimientos técnicos y recursos informáticos diferían mucho: unos pocos habían construido sus propias soluciones, mientras que muchos tenían dificultades. El plan pretendía que los investigadores pudieran concentrarse en investigar y equilibraba flexibilidad y estandarización, con una arquitectura modular, vías de entrada para fuentes de datos muy distintas, interfaces adecuadas a diferentes niveles de conocimientos, almacenamiento y procesamiento escalables, y estrategias de incorporación, soporte y gobernanza. Decidir qué conserva cada grupo para sí y qué comparte

todo el mundo es el equilibrio que Dehghani asigna a la gobernanza federada, y reducir los conocimientos que necesita un grupo es lo que pide a una plataforma de autoservicio.

Dónde surgen los problemas

Un equipo central con otro nombre. Llamar mesh al equipo de plataforma de datos no cambia nada si ese mismo equipo sigue recibiendo los datos originales de todos los dominios, limpiándolos y respondiendo por su calidad. La tabla de gobernanza de Dehghani ofrece una lista de comprobación útil: quién responde por la calidad, quién decide las reglas de seguridad y si el éxito sigue midiéndose en tablas bajo gobernanza. Traslade la responsabilidad por cada producto de datos al dominio que produce los datos y reserve el equipo central para la plataforma y las reglas.

Productos por los que nadie responde. Un dominio que exporta tablas sin un responsable, una descripción ni un compromiso de actualización solo ha trasladado el problema antiguo a su propio rincón. Dehghani pide que cada producto de datos exprese su calidad como objetivos de nivel de servicio, por ejemplo puntualidad y tasas de error, y señala que un dominio puede ofrecer dos productos sobre los mismos eventos: uno casi en tiempo real que puede omitir o repetir eventos, y otro que llega más tarde y es más preciso. Asigne un responsable a cada producto, publique sus objetivos y compruebe la calidad donde se producen los datos.

Sin identificadores compartidos. Si cada dominio identifica las mismas cosas a su manera, no se puede cruzar ningún par de productos. En el ejemplo de 2019, un artista aparece de forma distinta en el dominio que registra reproducciones y en el que paga a los artistas. Acuerde globalmente cómo se identifican esas entidades compartidas, que Dehghani denomina polisemas, y deje el resto de cada modelo a su dominio. Los formatos de campo, las convenciones de direcciones y los campos de metadatos comunes pertenecen a la misma lista global.

Cada dominio reconstruye toda la infraestructura. Sin una plataforma de autoservicio, cada dominio tiene que aprovisionar su propio almacenamiento, flujos de procesamiento y monitorización, y los conocimientos especializados que escaseaban en un equipo central pasan a necesitarse en todos. Construya la

plataforma antes de pedir a los dominios que publiquen, y evalúela por lo pocos conocimientos especializados que necesita un dominio para entregar un producto.

Cambiar un producto mientras otros lo consumen. Un producto de datos es una interfaz y modificarla directamente rompe los informes construidos sobre ella. En PostgreSQL, `CREATE OR REPLACE VIEW` impide parte de este error: la nueva consulta debe producir las mismas columnas, con los mismos nombres, orden y tipos de datos, y solo puede añadir columnas al final. No detecta un cambio de significado, porque el cálculo que hay detrás de una columna puede ser completamente distinto. Lleve cualquier cambio que renombre, elimine o redefina una columna a una vista nueva, `orders_by_day_v2`, junto a la antigua, y retire esta una vez que sus consumidores hayan migrado.

Un mesh donde no hace falta. Cuatro principios, un equipo de plataforma, un grupo de gobernanza federada y un responsable de producto en cada dominio suponen mucha estructura. Hacia el final, el artículo de Dehghani de 2019 admite que «la adopción a escala empresarial todavía tiene un largo camino por recorrer». Si unas pocas fuentes y un equipo cubren las necesidades, mantenga un almacenamiento central bien gestionado con un responsable identificado para cada tabla y tome prestadas las partes que compensan a cualquier escala: responsables, conjuntos de datos descritos y reglas que comprueba la máquina.

Lecturas adicionales

- Principios y arquitectura lógica del data mesh — Zhamak Dehghani, 2020: los cuatro principios, las partes de un producto de datos y la tabla que compara la gobernanza central con la federada.
- Cómo pasar de un lago de datos monolítico a un data mesh distribuido — el artículo original de mayo de 2019, con las tres formas de fallo de una plataforma de datos central y las cualidades que necesita un producto de datos.
- Referencia de diseño guiado por el dominio — el resumen de patrones de Eric Evans, incluido el contexto delimitado cuyos límites dividen las responsabilidades en un mesh.
- PostgreSQL: CREATE VIEW — las reglas para sustituir una vista y los privilegios que necesitan sus usuarios, tal como se utilizan en el ejemplo.
- PostgreSQL: COMMENT — cómo se almacena una descripción sobre un objeto de base de datos, que `obj_description` recupera después.

Glosario

Elija según sus restricciones reales

Traiga un conjunto de datos representativo, las preguntas que debe responder y el cuello de botella actual. Podemos acordar cómo medir la calidad, la capacidad de procesamiento y la recuperación.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es un data mesh?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

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

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/data-mesh/