EBOOK

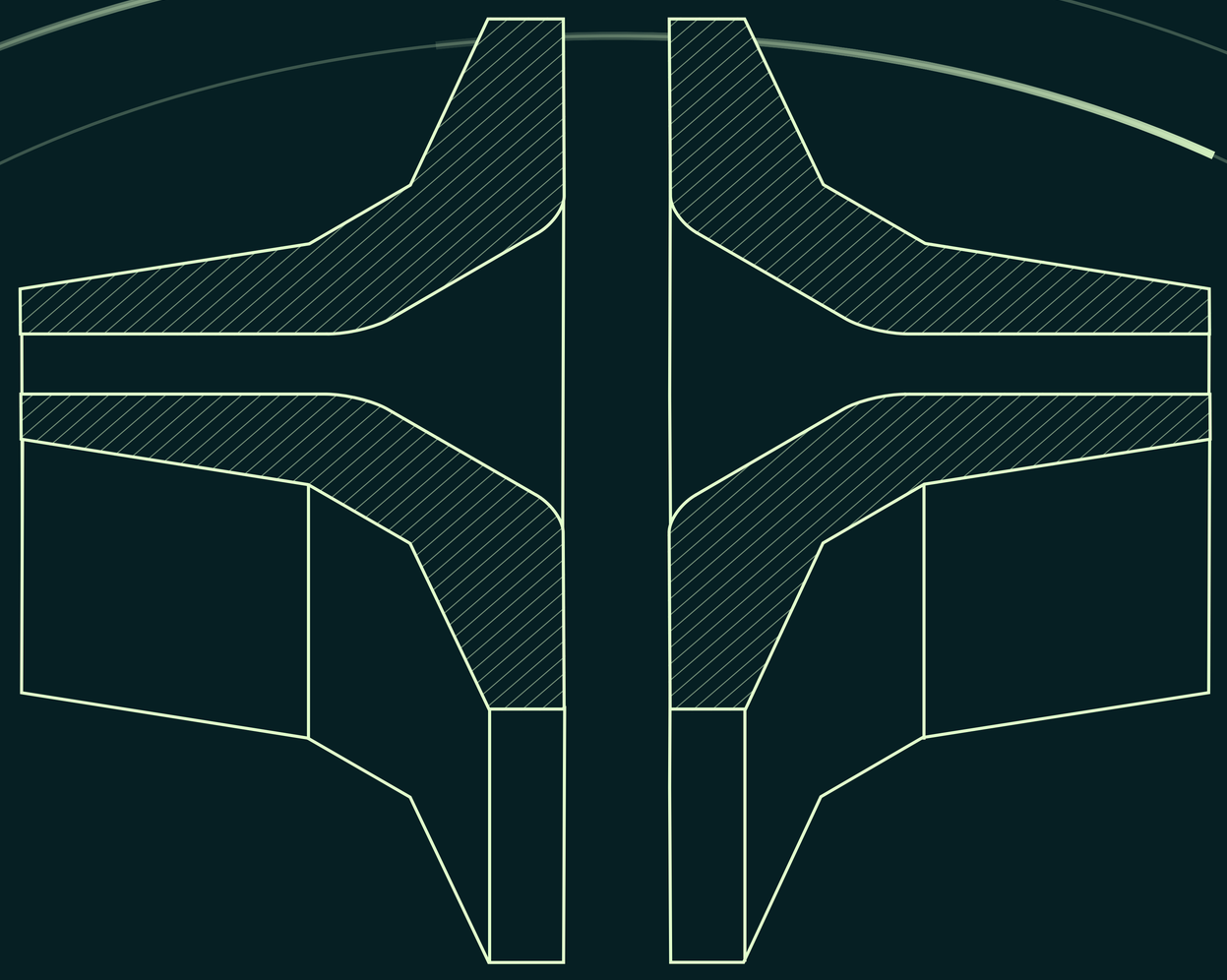
conduktor

Federation

Solve Data Governance at Scale

conduktor.io

2025



Contents

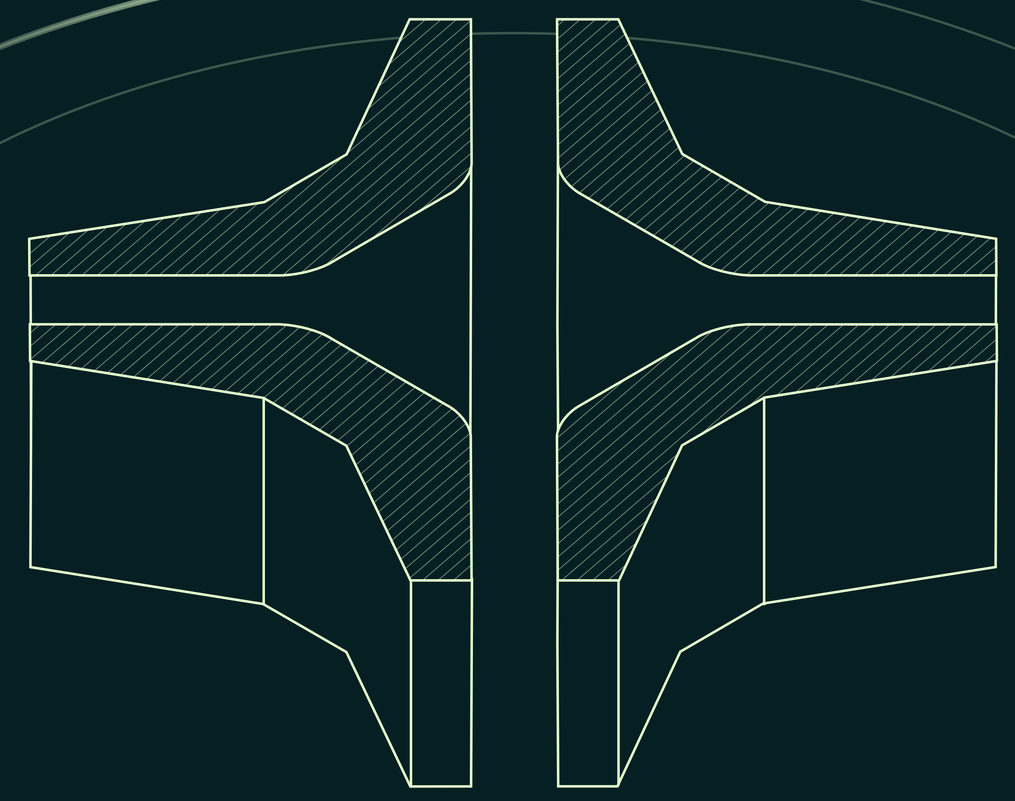
Abstract	01
What is federation—and how does it reconcile control vs. autonomy?	04
Where does federation fit into the data mesh paradigm?	06
How does federation look in practice?	09
How can federation supercharge innovation?	11
How Conduktor handles federation	13
Case study: Swiss Post	16
Conclusion	18

Abstract

Modern data environments face a difficult dilemma: centralized control or decentralized speed, each with their disadvantages, including throttled developer velocity or operational chaos.

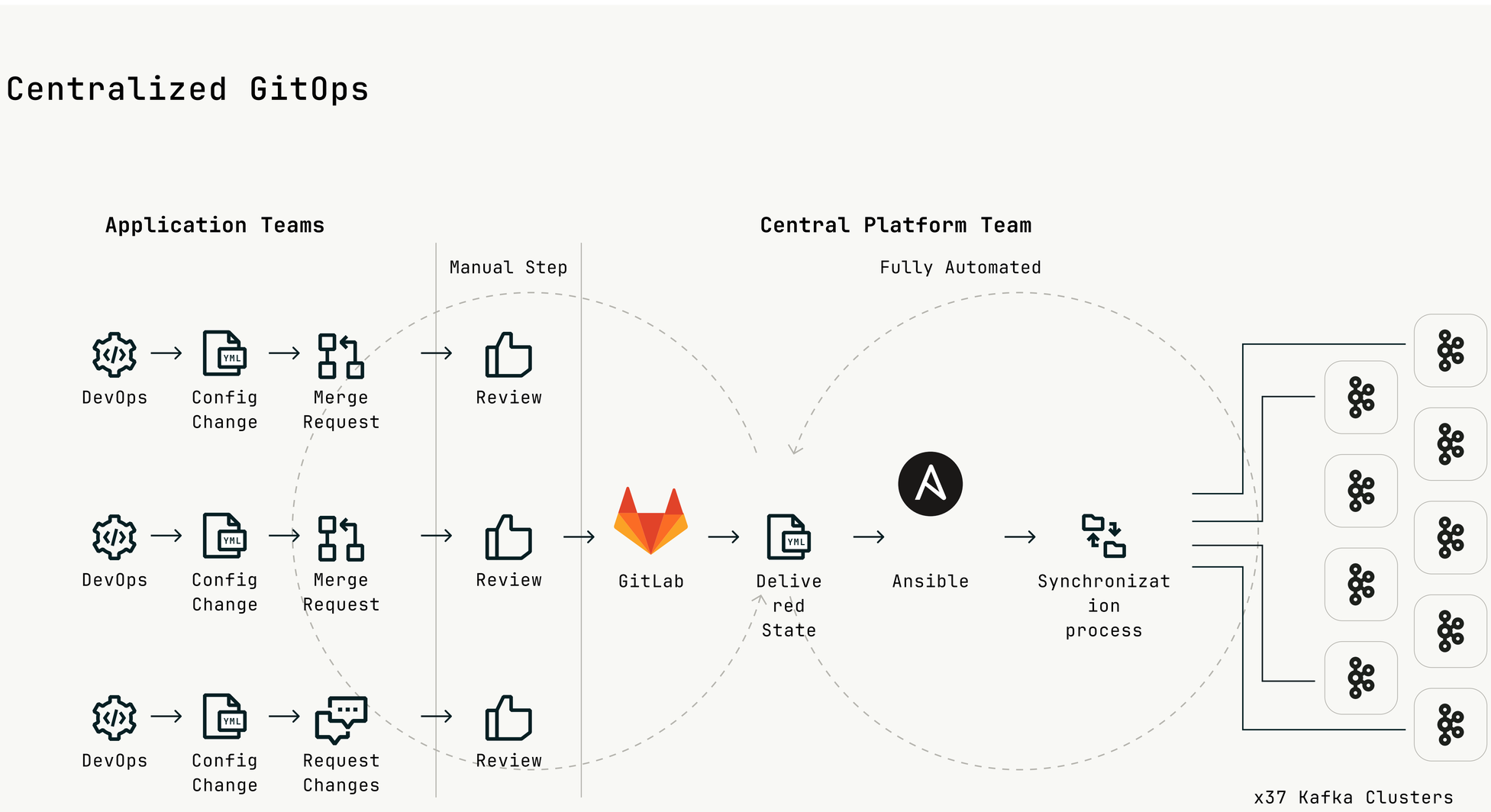
Instead, there is a third path: federated governance. Just as federal governments devolve some functions to local governments, so too can platform teams grant developers and data teams a measure of independence. This approach balances platform stability with developer autonomy, reducing friction and ensuring agility without compromising security, data quality, or observability.

Learn how federation can transform your data architecture—and how Conduktor empowers your teams to operationalize governance.



Modern data environments are brutal to manage—for both rapidly growing startups and established enterprises alike. Regardless of the size of their team or the scale of their infrastructure, companies will soon encounter the same choice: should they optimize for platform stability or developer speed?

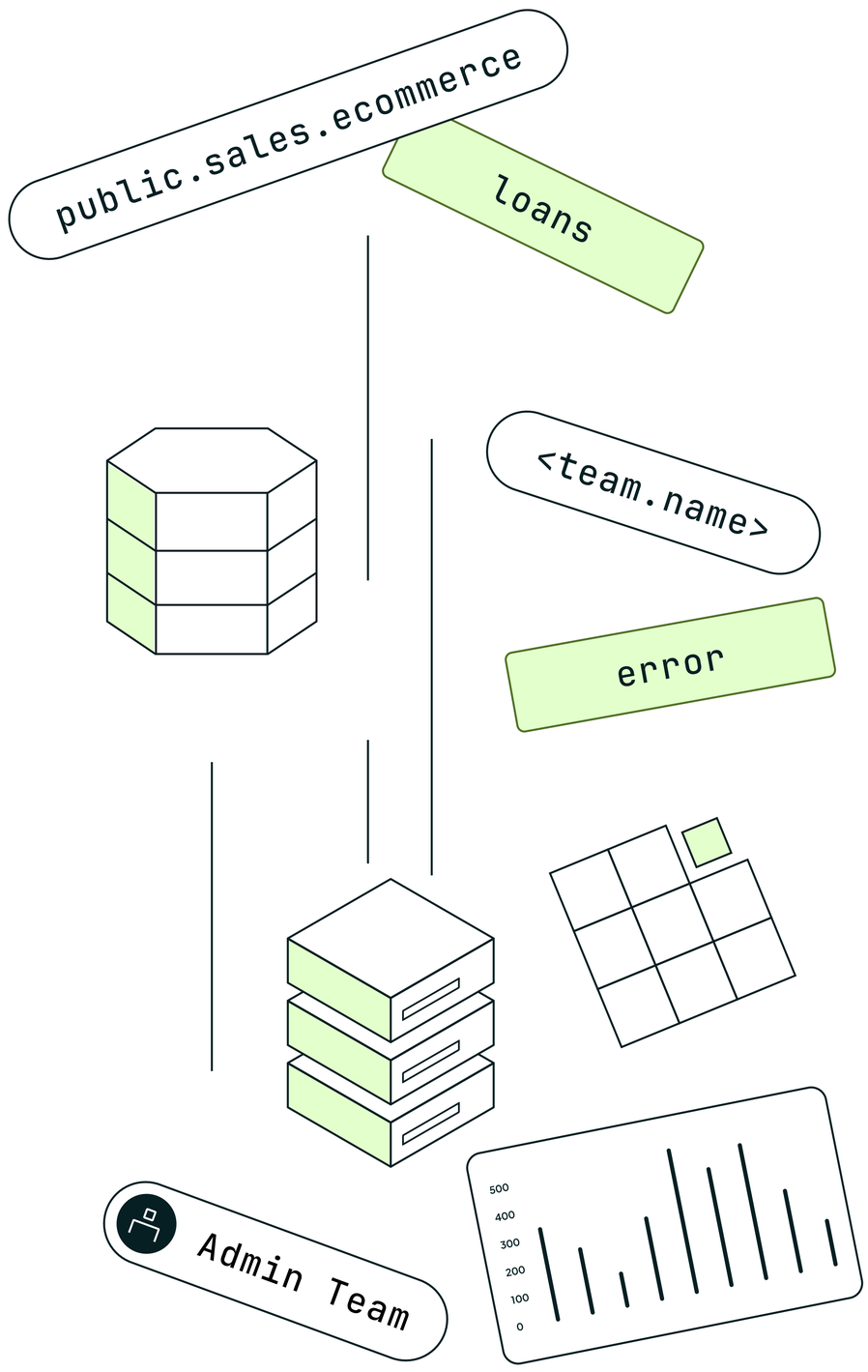
This is a difficult choice. If companies lean too far toward stability, routing every request through the platform team, then platform engineers will be quickly overwhelmed by tickets and troubleshooting. Their team becomes a bottleneck, slowing innovation, stalling growth, and eventually burning out their engineers. As a result, developers will be blocked from accessing data, creating a topic, or deploying a connector, unable to complete projects and deliver value.

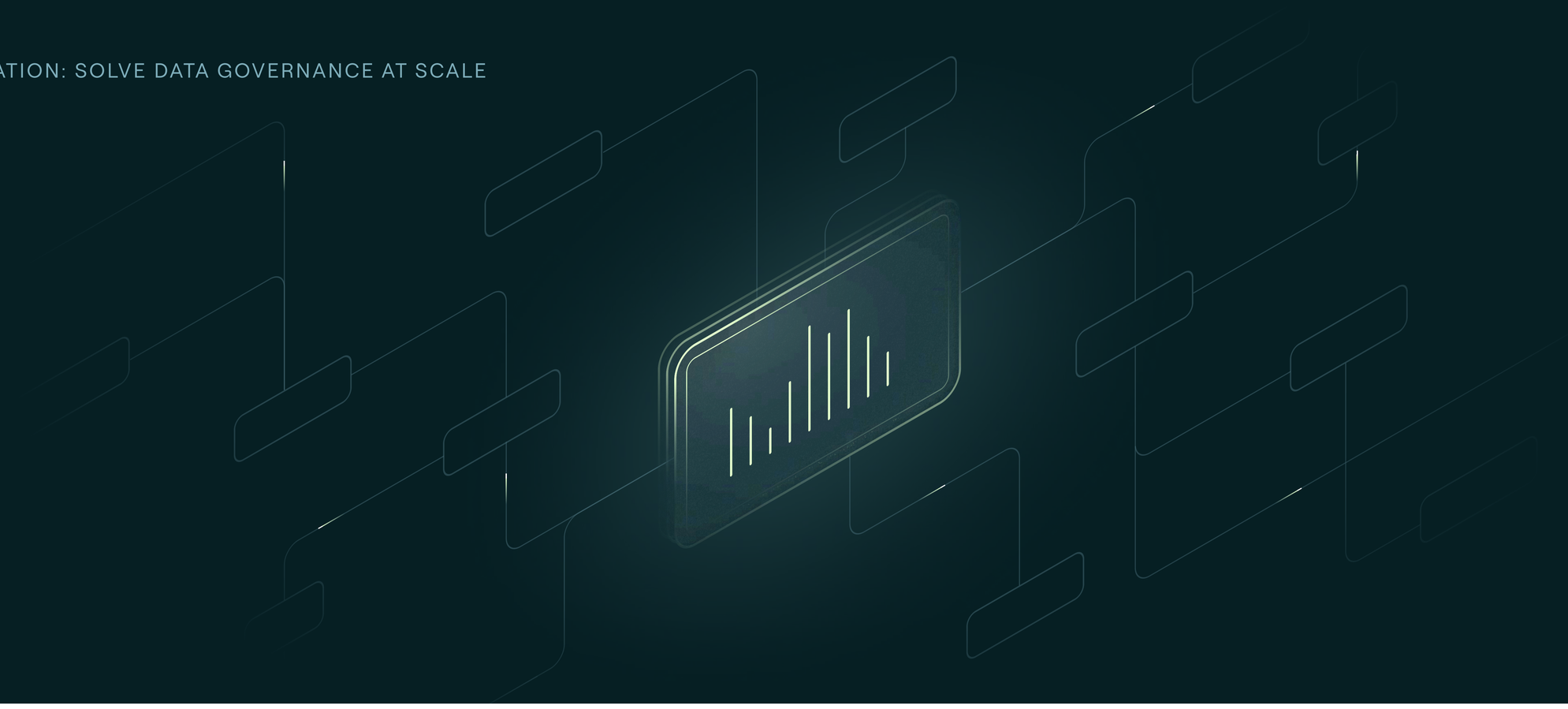


But if a company goes too far the other way, favoring developer autonomy over everything else, they will encounter operational chaos. Without standards for naming data infrastructure, updating schemas, or setting data retention periods, developers will invent their own, creating conflict and confusion. Simultaneously, inconsistent or malformed data spreads downstream, breaking applications, analytics, and AI, while security gaps appear as developers relax access controls for agility.

In both approaches, operational and analytical data estates (and teams) are disconnected and siloed, lacking a shared understanding of details such as schema, KPIs, data freshness, and ownership. Operational teams are unable to persist analytics, maintain or update schema, or to predict what effects their data will have on downstream applications; analytical teams can't find or access necessary data, or suddenly encounter breaking changes when their applications ingest operational data.

Ultimately, both extremes are unsuitable, and bring about the same result: your data environment becomes a liability, rather than an asset.





What is federation—and how does it reconcile control vs. autonomy?

But this dichotomy is false. Instead, companies can federate their data governance, allowing them to get the best of both worlds: centralized oversight alongside local autonomy. Without federation, organizations face spiraling costs, failed AI initiatives, and compliance gaps. With federation, they can build a model for teams to collaborate effectively, encouraging developers to move fast within safe boundaries, while platform teams maintain stability without becoming bottlenecks.

The concept of federation originates from political science. The federal model, in use in Brazil, the United States, and Germany, divides powers between the federal (national) and local (state and municipal) governments. The federal government sets broad rules for defense, finance, and foreign relations, while states and cities define local laws for sanitation, healthcare, or transportation. All of this is rounded off by a constitution, which serves as a shared legal and operational framework.

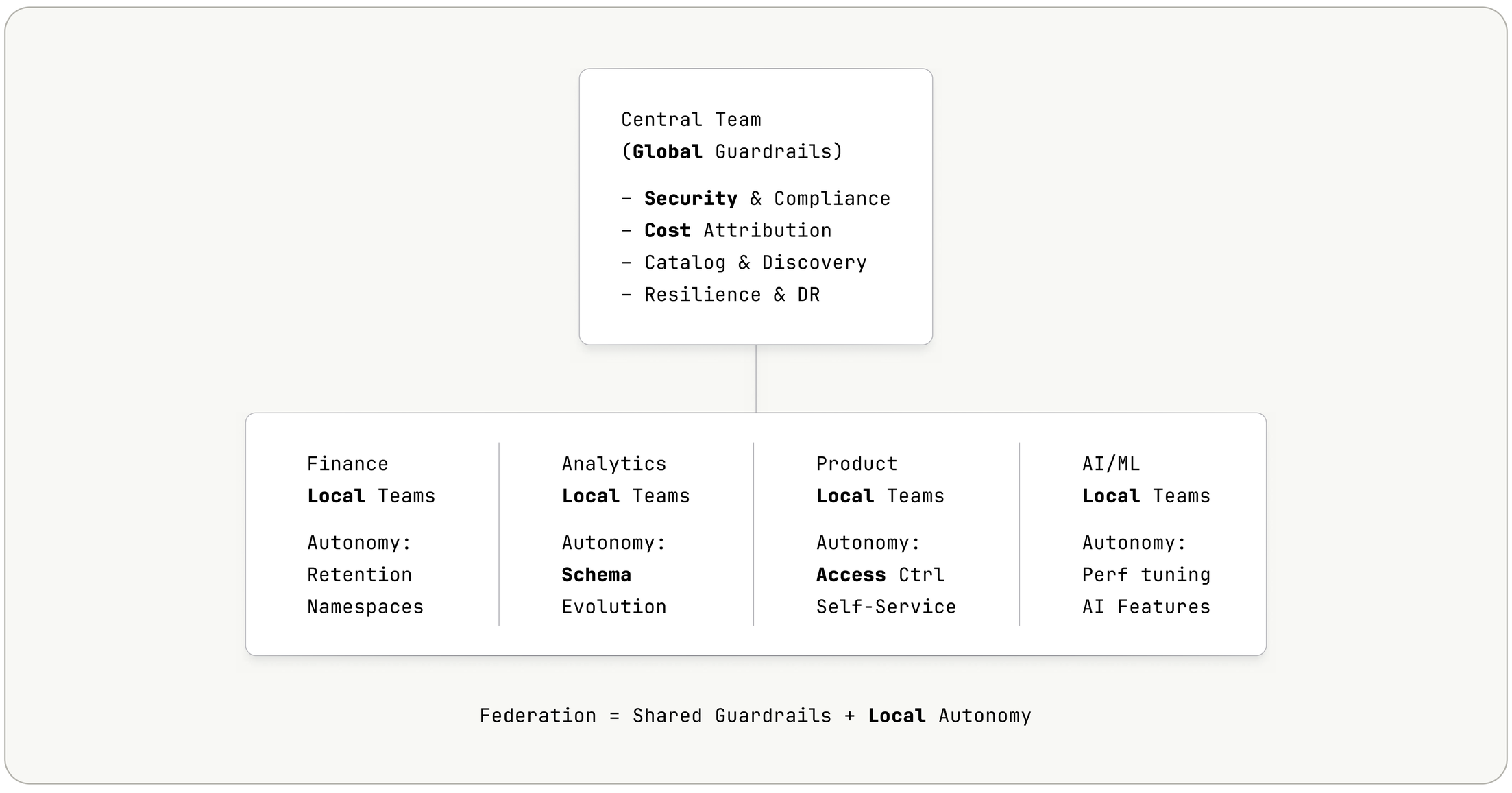
In this analogy, the platform team is the national government. Depending on the company, platform teams may govern the entire business, or be divided amongst subsidiaries in different industries or regional units. Either way, platform teams are charged with safeguarding the overall data environment and defining key policies for security and encryption, disaster recovery, and naming conventions.

Application teams and end users, which include data analysts, AI and machine learning engineers, application developers, and data engineers, operate like states. They have the independence to configure the details of their environments, experiment with retention policies, onboard temporary users like consultants, and monitor metrics.

Just as federal governments require a legal framework, such as courts and constitutions, to function, so too does federated data governance also require a technical framework.

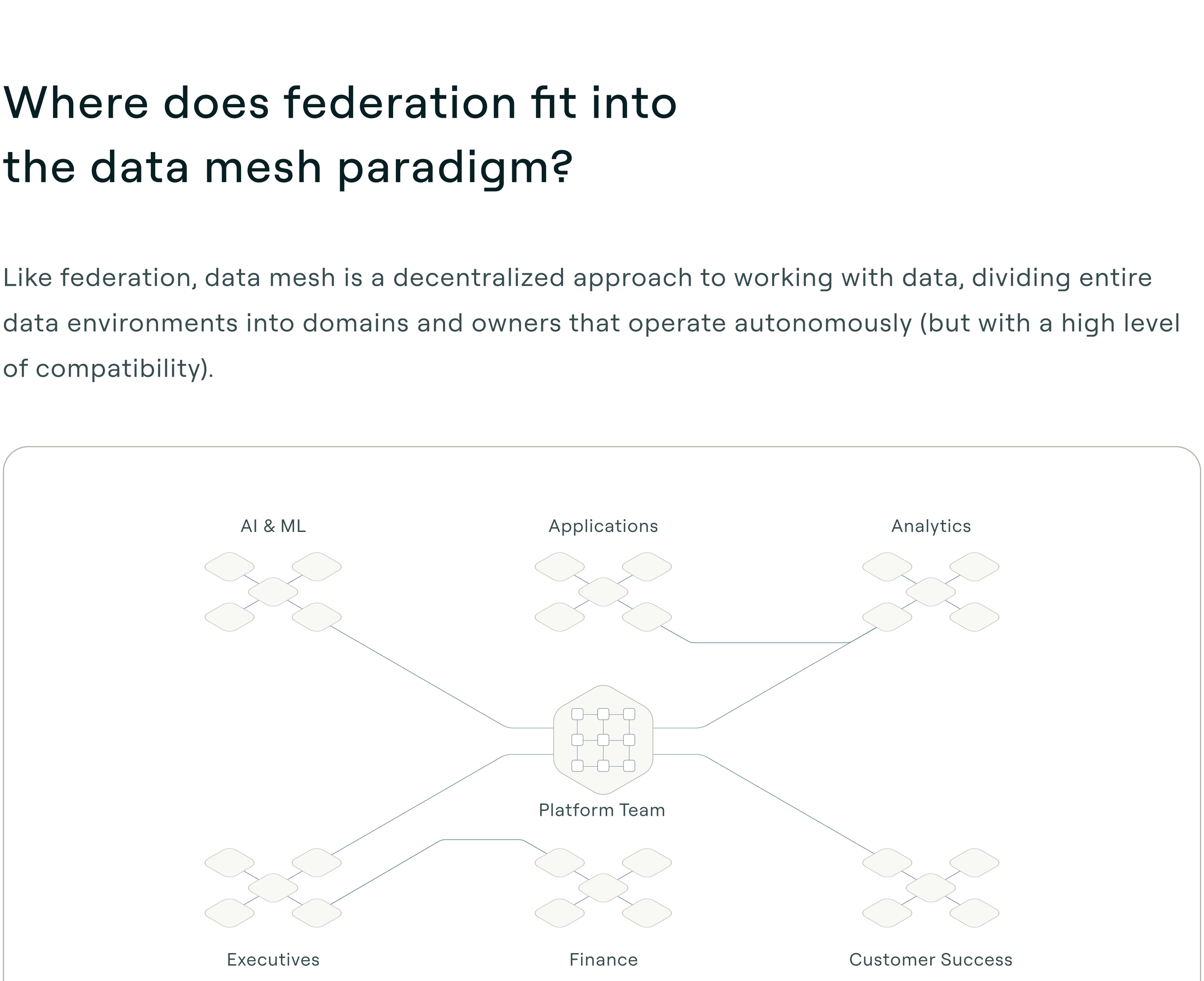
In this case, it’s a solution that will handle the technical details, by:

- Allowing platform teams to establish safe boundaries that data owners, developers, analysts, and engineers can work with.
- Decentralizing ownership by assigning owners to data domains, and empowering these individuals with the authority to approve or reject access requests and to manage controls for quality, schema evolution, and retention.
- Providing insights on Kafka usage and expenses, guardrail violations, expired or unused RBAC permissions, and performance and operational metrics.



Where does federation fit into the data mesh paradigm?

Like federation, data mesh is a decentralized approach to working with data, dividing entire data environments into domains and owners that operate autonomously (but with a high level of compatibility).



In fact, federation is indispensable to the concept of a data mesh. Without it, data mesh simply recreates all the problems associated with the excessive devolution of power to local teams, including insecure or non-compliant data (and a heightened risk of leaks or hacks) or operational chaos arising from a lack of uniform governance.

Executed correctly, data meshes include many benefits:

Parallelism.

Teams can scale, manage, and update their domain without impacting others. This also improves resilience, as issues in one domain will also no longer spill over to other domains.

Improved data quality.

By devolving key aspects of domain management to the ownership team, who are closer to (and have a better understanding of) the data, data meshes benefit from higher data quality. In contrast, if central teams pick up this responsibility, organizations risk encountering data quality issues, simply because central teams don’t understand the intricacies of every environment and domains within a company.

Interoperability.

Various domains can seamlessly collaborate together, exchanging, using, and understanding data regardless of where it lives or who owns it—and without having to worry about incompatibilities or other friction. Lacking interoperability, a data environment isn’t a mesh so much as it is a siloed mess. With interoperability, a data mesh can promote the creation of reusable data products, improve governance, and reduce the overhead associated with maintaining and integrating a patchwork of different technologies. This also creates…

...simplified data products.

With higher quality data and improved collaboration, teams can build and release reusable data products such as APIs, event streams, and curated data sets for consumption by other internal users. This reusability also removes redundancies and accelerates delivery, as teams can now use shared data products for new projects instead of building their own from scratch.

Speed.

Due to their decentralized nature, data meshes turbocharge innovation and insights. Because domains can now work independently, teams can work amongst themselves, moving products from concept to production without going through a central team.

Tighter integration between operational and analytical data estates.

Far too many companies silo operational and analytical systems (and teams). A properly-implemented data mesh can bridge this gap, allowing real-time operational data to flow seamlessly into analytics, AI, and applications. This creates more timely, trustworthy outputs, shortening decision cycles and helping organizations gain a competitive edge.

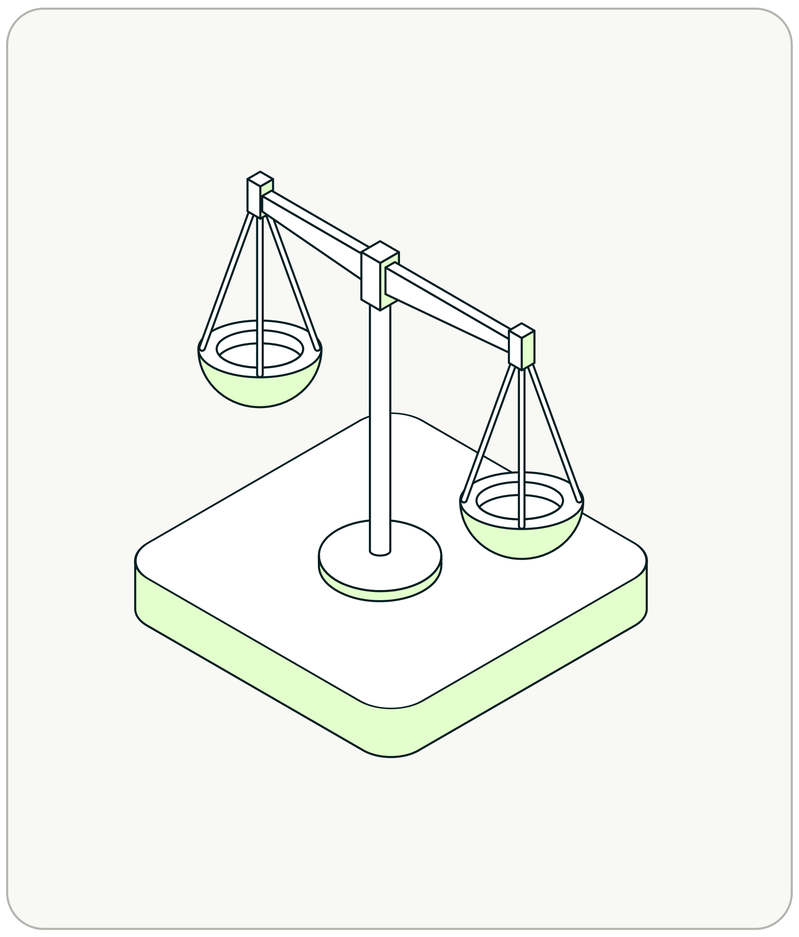
Accountability and transparency.

Clear guidelines, coupled with supporting technology such as a governance solution, make it easier to stay within safe boundaries for operations, compliance, and security. If an issue arises, it's easier to trace the root cause.



How does federation look in practice?

In order to successfully federate data governance, organizations must know where to draw the line between control and autonomy. Some operational concerns are too granular or business-specific for platform teams (and must be devolved to data owners and application teams), while others are too critical to leave to chance.



The key is to effectively balance these concerns. While the specifics of federated governance may vary by company, the basics remain similar across organizations and industries.

Generally, data owners will manage all the details that are closest to their work. For instance, they can name their topics whatever they like, so long as they follow approved naming conventions that prevent conflicts. Local teams will also define data retention policies, assuming that they don’t exceed specific restrictions set by central teams, and onboard users.

Because local teams also understand their workloads and environments best, they will also be responsible for performance management. Owners and their teams will determine which latency thresholds are acceptable, how they can tune consumer lag alerts, or when they can add or remove connectors. They’ll also manage schema versioning and upgrades —but likely leave larger concerns such as backwards compatibility for platform teams.

Meanwhile, platform teams will have authority over systemic concerns that affect the entire enterprise. They will oversee disaster recovery, cross-region replication, and failover; perform chaos testing and global performance monitoring; and define shared governance standards for naming, partitioning, and discovering topics. This way, platform teams can prevent confusion and silos while ensuring interoperability.

Importantly, federation acknowledges that not all environments are created equal. A team ingesting telemetry from a factory assembly line has different requirements and risk tolerances than a team analyzing event data from a software application. Each vertical needs the flexibility to operate on its own terms, but within a consistent enterprise framework.

With the right technology, the flexibility provided by federation can remove friction. For instance, if a product team needs to grant temporary Kafka access to an external consultant or contractor, they can use self-service controls to do so safely, quickly, and without involving the platform team. This creates faster collaboration, tighter control, and less operational overhead.



How can federation supercharge innovation?

Federation doesn't just improve data governance, it also boosts productivity and innovation. By balancing autonomy with oversight, federation ensures that data is a competitive advantage rather than a source of friction (and frustration).

At its core, federation takes a systematic approach to collaboration. Faced with clear, non-negotiable rules, teams understand how to operate and what to expect.

One vital concern is ensuring that new technologies or infrastructure remain compatible with the larger enterprise environment. For example, a simple policy requiring all data connectors to remain backwards compatible ensures that updates never break existing data flows. Teams can experiment freely, deploying and scaling new products as needed—so long as their solutions of choice do not conflict with existing infrastructure.

Some centralization does remove confusion around operations, and brings consistency and predictability. For instance, a platform team mandates that schema updates cannot drop fields, instead retaining outdated fields with values of NULL or 0. By implementing this rule, platform teams can reduce the frequency of application issues, saving application teams (and other downstream users) from having to waste hours on restoring service.

In addition, when data is more secure, compliant, and discoverable, it is also more usable. Federated governance allows owners to build catalogs, helping other users to quickly find the data they need (and to provision their own accounts) without having to wait on other teams. Data owners still have the final authority to allow or deny access, adding a layer of safeguards.

Templating workflows for common components, such as connectors, topic configurations, or integrations, not only makes it easier for users to work independently, but also takes the guesswork (and risk) out of the process. Less technical roles, such as data analysts or business leaders, can use these preset sequences to access data without involving platform teams or data owners—and thus move faster.



Being able to share data seamlessly and collaborate across teams without friction has saved countless hours. It's improved the quality of questions and insights we receive, which boosts overall productivity...

—Scot M, Platform Engineer at Virgin Australia

Because of their parallelism, federated data meshes are also far easier to scale. In some ways, this is similar to a microservices architecture, where multiple interlinked, yet independent nodes can scale as needed. In this analogy, each node is a different data domain, which can scale at its own pace, and more easily accommodate fluctuations in user traffic or demand.

Lastly, federation encourages autonomy, but not at the expense of accountability. As data owners, domain teams are tasked with managing access, quality, and compliance for their local environment. At the same time, owners will still operate within the boundaries set by platform teams, guaranteeing safety and reducing friction.



How Conduktor handles federation

As an operating model, federation is nearly impossible to execute manually, especially via a patchwork of technologies, scripts, and unwritten rules. An ad hoc approach simply can not scale, especially in large data environments: governance will break down, visibility will become nonexistent, and operations will devolve into chaos.

Instead, any team (or organization) that seeks to operationalize federation will require tooling to enforce boundaries, enable autonomy, and keep up with growth in users, traffic, and data domains.

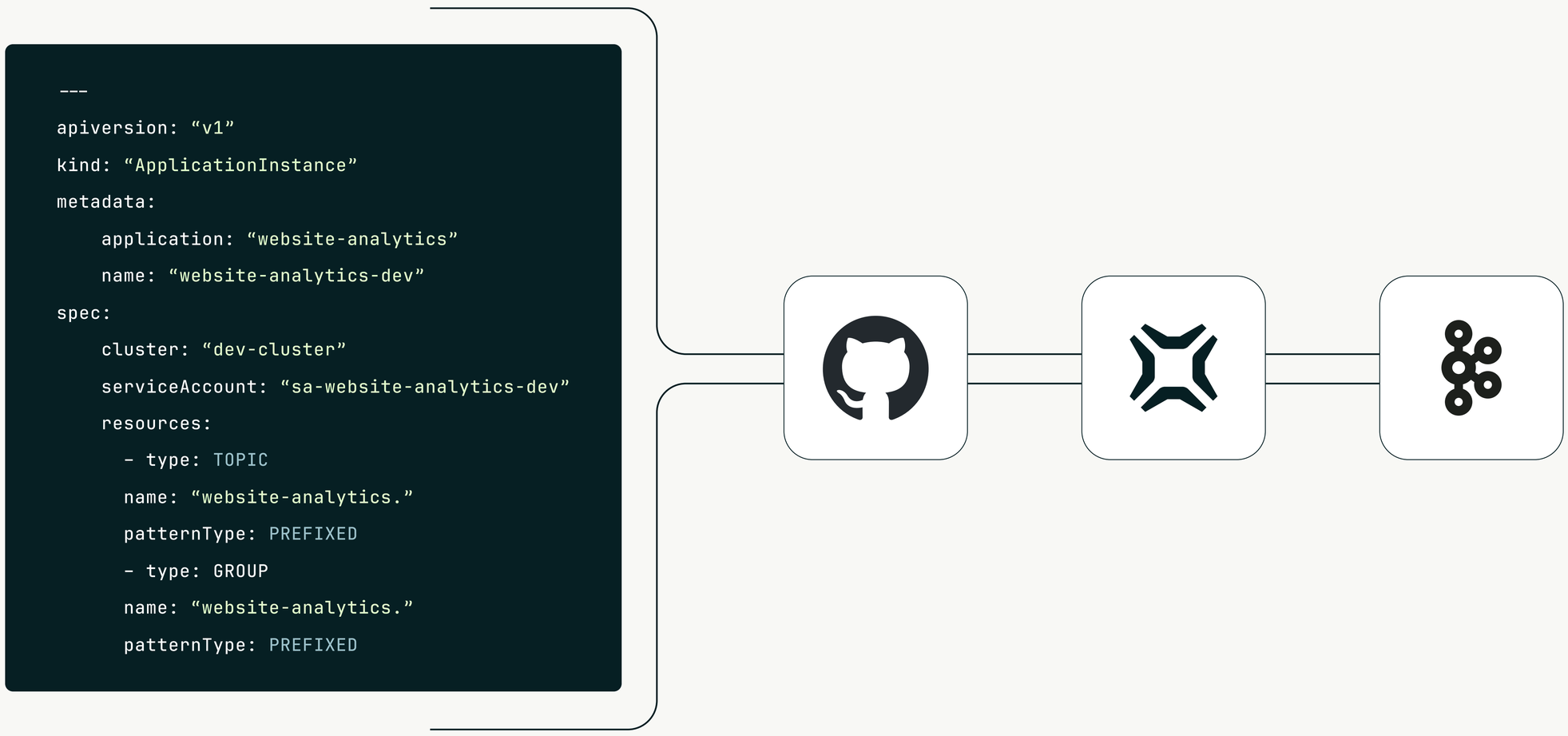
That's where Conduktor comes in.

Conduktor is a governed data hub that sits on top of data streaming infrastructure, between source systems and consumers such as applications, lakehouses, business intelligence tools, and AI agents. It gives platform and local teams exactly what they need: guardrails to work safely, self-service to access data quickly, and shared visibility for collaboration.

Governance tier	Conduktor enables…
Central platform team	Defines global guardrails for encryption, schema, partition limits, and naming conventions
Local domain teams	Self-service, access control, connector deployment, schema actions (within bounds)
Shared infrastructure	Unified observability, Insight Center, audit trails, disaster recovery, data catalog and discovery

Conduktor solves the core challenges of Kafka governance, including:

- **Kafka as a black box.** Many platform teams have no observability into their data infrastructure, and are unable to understand what's happening. Conduktor provides visibility into topics, clusters, and domains via dashboards and monitoring.
- **Security and compliance gaps.** Conduktor builds encryption, auditing, and resilience directly into the data streaming layer. Every interaction can be traced, every user action logged, and every environment complies with preset rules.
- **Governance impeding innovation.** With Conduktor, platform teams can use pre-defined policies and templates to automate repetitive tasks. Routine actions, such as onboarding users, deploying connectors, or setting retention policies, can be converted into self-service workflows.
- **Environments can't scale or integrate new technologies.** As a platform-agnostic solution, Conduktor prevents vendor lock-in and supports a wide range of on-prem, cloud-based, and hybrid deployments and products.

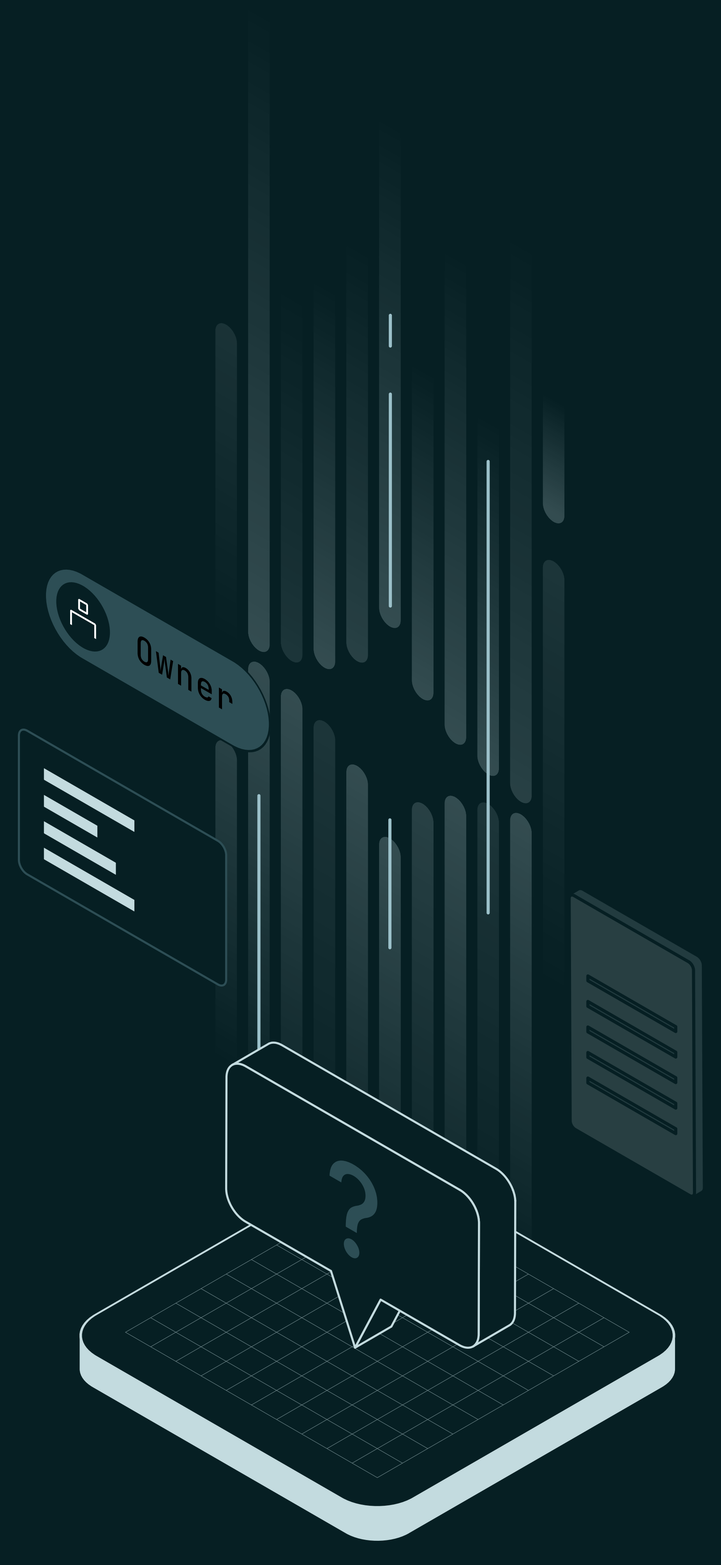


☑ Fragmented operational data.

In a federated model, every team needs seamless access to insights. But when data is scattered across dashboards and tools, insights are much harder to extract, and become a blocker. Platform teams cannot determine which policies to create, application teams cannot prioritize which components to optimize, and architects are unclear on the returns (or losses) of their governance efforts.

For this reason, Conduktor is planning **Insight Center**, an AI-powered feature that can help users quickly answer governance questions via natural language queries. For instance, if a platform team needs to find the five data domains (and their associated owners) with the highest monthly spend, they can use the Insight Center.

As another example, if a platform team needs to identify which topics are missing vital encryption for sensitive data or violating retention policies, they can simply query Insight Center, rather than sifting through SQL and aggregations. Ultimately, this feature helps central teams get granular visibility and accountability into how they operationalize governance.





Customer case study: Swiss Post

In 2024, Swiss Post handled 1.55 billion letters, 107 billion customer assets, and 180 million parcels. As real-time data became central to everything from sorting facility automation to customer analytics, Swiss Post faced a familiar challenge: scaling Kafka and modernizing their systems, all without ceding governance or stifling developer autonomy.

Under platform team lead Anke Raich, Swiss Post grew their Kafka infrastructure from 60 to 300+ applications, and to over 1,100 integrations, all without increasing the size of their platform team. Using Conduktor, Swiss Post implemented federated governance, balancing autonomy with oversight across their data environment.

> **Being able to share data seamlessly and collaborate across teams without friction has saved countless hours. It's improved the quality of questions and insights we receive, which boosts overall productivity...**
>
> —Anke Raich, Swiss Post

As Kafka became increasingly important to Swiss Post, the platform team struggled with visibility, access management, and compliance. Developers were unable to self-serve, analysts couldn’t discover or access data, and Kafka had effectively become a “black box,” in Anke’s words.

As a result, Swiss Post used Conduktor to implement centralized guardrails for compliance and reliability, enabling autonomy via self-service and role-based access controls (RBAC). Developers, data scientists, and analysts could now independently discover, explore, and validate data—without waiting for the help or approval of the platform team.

By federating governance, Swiss Post teams could now automate data access and quality management at scale, ensuring that users worked within clear guidelines. At the same time, Conduktor also facilitated monitoring and observability for the platform team, giving them a unified view of Kafka clusters and schema health.

Conduktor helped Swiss Post achieve the following results:

- A fivefold increase in Kafka applications (from 60 to over 300), and a threefold increase in integrations (from 300 to 1,100+).
- Empowered 800+ internal users across all departments with secured access to Kafka.
- Zero increase in platform headcount, achieved through automation and self-service.
- Ensured compliance with key legislation, such as the Swiss Federal Act on Data Protection, via strict RBAC controls.
- Higher data quality and AI readiness, as teams could now govern and reuse data with confidence.



Conclusion

By federating data governance, enterprises can opt out of the developer autonomy or platform stability dilemma, instead choosing to have both. By decentralizing ownership while standardizing shared safeguards, organizations can move faster, stay compliant with rules and policies, and take a systematic approach to innovation.

As a governance hub, Conduktor empowers organizations to take this third path, and to facilitate self-service, monitoring, and management, with the help of automation and AI-powered insights. Whether your organization is new to Kafka or an established enterprise with a massive data environment, Conduktor can help your teams operate with accountability, independence, and confidence.

If you’re ready to operationalize federation at scale, book a Conduktor demo. Learn how Conduktor helps platform teams turn Kafka from an expensive black box into a governed, valuable business asset.

conduktor

To learn more about Conduktor, sign up for a demo today.

Book a demo