Cyral

WHITE PAPER

Data Access Governance for Securing the Modern Data Mesh

Contents

Overview 3

The Case for Data Access Governance in the Data Mesh 4

Access Management as the Root of the Data Mesh Security Challenge 6

Lack of Native SAML/OIDC Support 7

Service Accounts for BI Tools and Applications 8

The Data Mesh is Not Safe When Using Traditional Methods 9

An Access Management and Governance Model That Works 10

Identity Federation 10

Service Role Disambiguation 11

Privileged Access Management 12

Security as Code Model 13

Benefits of Identity-Based Access Management 14

Simplified Access Management 14

Centralized Compliance Controls and Audits 15

Reduced Attack Surface 16

Summary 17

Learn how to embrace data democratization and be data driven, while efficiently managing the security of your data mesh.

Overview

There is a rapidly growing ecosystem of cloud-based data services like Snowflake, MongoDB, Looker, Databricks, Kafka, and Tableau. These services store, process, and analyze operational intelligence, business intelligence, and other structured and semistructured data.

Referred to as the Data Mesh, this growing ecosystem of databases, data lakes and data services enables businesses to embrace data democratization and be data driven. It eliminates silos, unlocks innovation, and helps improve customer experience and company culture.

To truly harness these benefits, IT, DevOps, and security teams must learn how to properly govern access across their Data Mesh architecture for all their employees, partners, applications, and BI tools without impacting agility and user experience.

In this white paper, we outline a new model for Data Mesh security, and describe a solution that organizations can use to implement it.

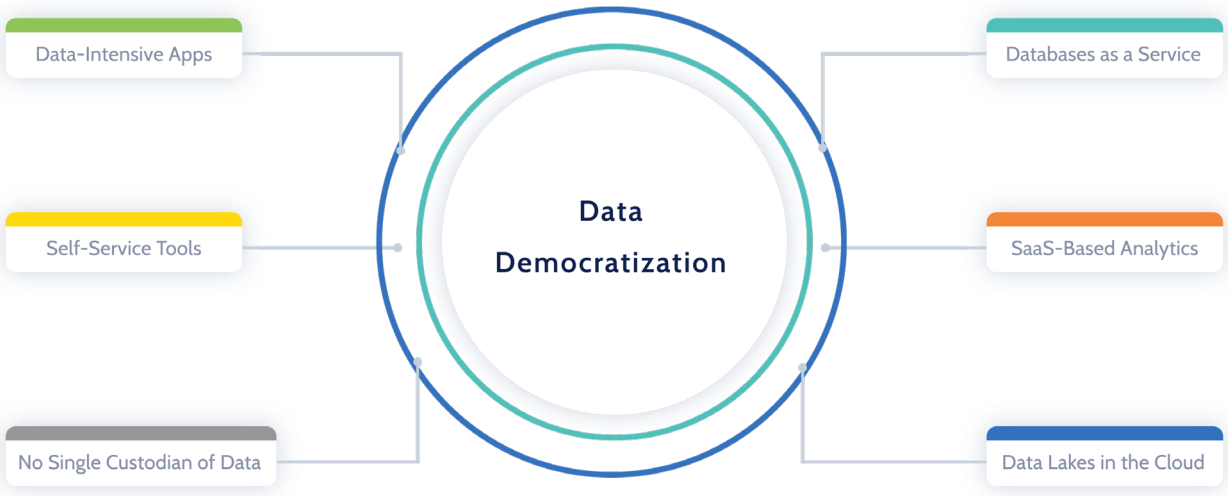
The Case for Data Access Governance in the Data Mesh

The last few years have triggered a once-in-a-generation shift in data engineering and processing technology. There are three core drivers behind this: growth in data collection, expansion in data access, and technologies that accelerate the first two.

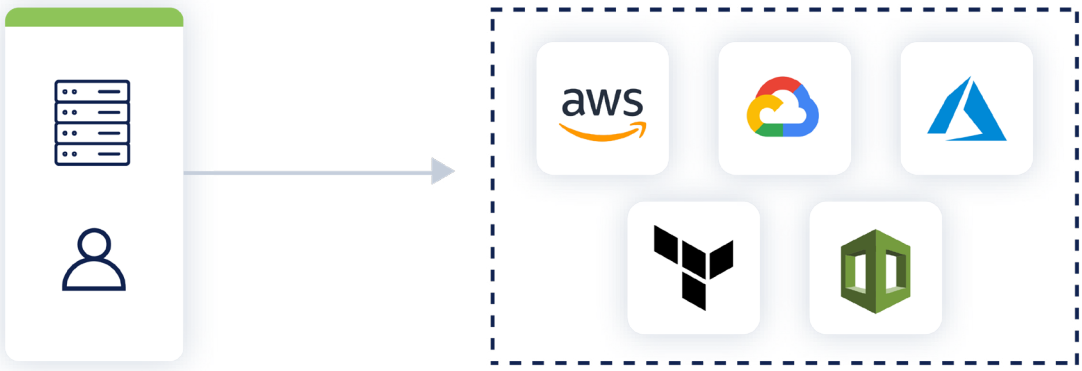
As companies have undertaken digital transformation initiatives, collecting data is at the core of these new services and offerings in order to drive engagement and innovation. One study in 2017 showed that data growth was occuring at a rate of 10x every 5 years.

As businesses plan to succeed in the “data economy”, democratizing access to data is recognized as the key to digital transformation. Data democratization means everyone in your organization has access to the right data in a highly usable way. It is the key to engaging customers, creating new opportunities, and unlocking value embedded inside organizations.

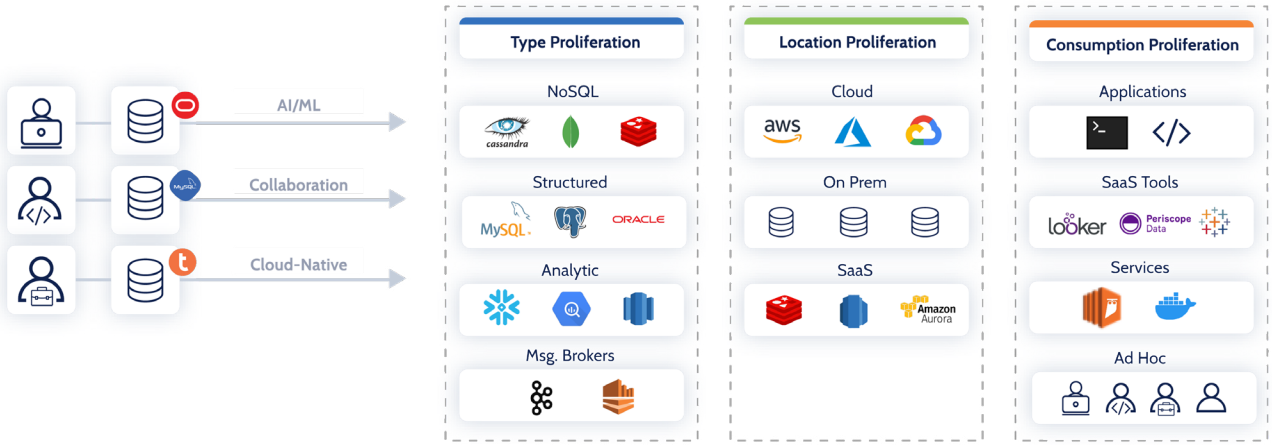
Boards are now asking management to break down silos and create a fluid ecosystem of data services that can be directly accessed by developers, data scientists, product managers, and analysts. This is what the Data Mesh is.



Simultaneously, as digital transformation drives IT teams to a multi-cloud architecture, these teams are increasingly adopting an infrastructure as code approach to industrialize their application delivery process. It enables them to rapidly deploy new services and specify the infrastructure on which they run, directly from their continuous integration/ continuous delivery (CI/CD) pipeline, without any human involvement.

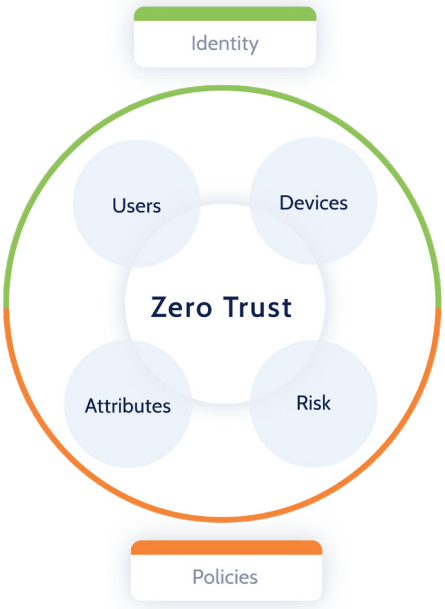


With the growth of the Data Mesh and infrastructure as code, businesses now have the opportunity to achieve massive gains in speed and efficiency, operate at virtually unlimited scale, and optimize price-for-performance. They can make their data and services readily accessible when and where it is needed, leaving legacy IT organizations far behind in agility. However, the implication of this new architecture is that the most critical data—which was earlier confined to a few trusted systems of record—now proliferates everywhere. This results in an urgency for organizations to securely govern user access to their Data Mesh, without impacting their data democratization or IT modernization efforts.



Access Management as the Root of the Data Mesh Security Challenge

As companies migrated to the cloud, traditional security models that relied on networkbased segmentation became increasingly obsolete. In their place, Zero Trust gained popularity as the desired security model for cloud. Popularized by Google and other thought leaders, it relies on knowing the identity of each user and device before authorizing access to any service or data.



As a result, the last few years have seen a dramatic growth of identity-based security services. These can be categorized as follows:

	Description	Examples
Identity Providers	An identity provider is any service that can be used to securely validate the identity of a human or a computer.	Active Directory, AWS IAM, Google IAM, Okta, Keybase, OneLogin, Auth0, JumpCloud
MFA Providers	A service that provides a second factor for authentication.	Okta, Yubikey, Duo, Google Authenticator, Authy, LastPass, OnePassword, JumpCloud
Federation Services	A federation service is an identity provider that can be used to delegate an identity to an outside service. OIDC and OAuth are examples of identity federation.	Google, Facebook, Okta, Auth0, Azure Active Directory, Gitlab, Keycloak, JumpCloud
Ephemeral Credentials Managers	This is any service that issues short-lived credentials to a human or computer.	AWS STS, Cyral, HashiCorp Vault

Today, most IT and DevOps teams rely on these services to centrally control access to their applications and infrastructure. However, for all these services to be effective, it is imperative for the target resources being managed to support protocols like SAML and OIDC. This is where the challenge lies for the Data Mesh.

Lack of Native SAML/OIDC Support

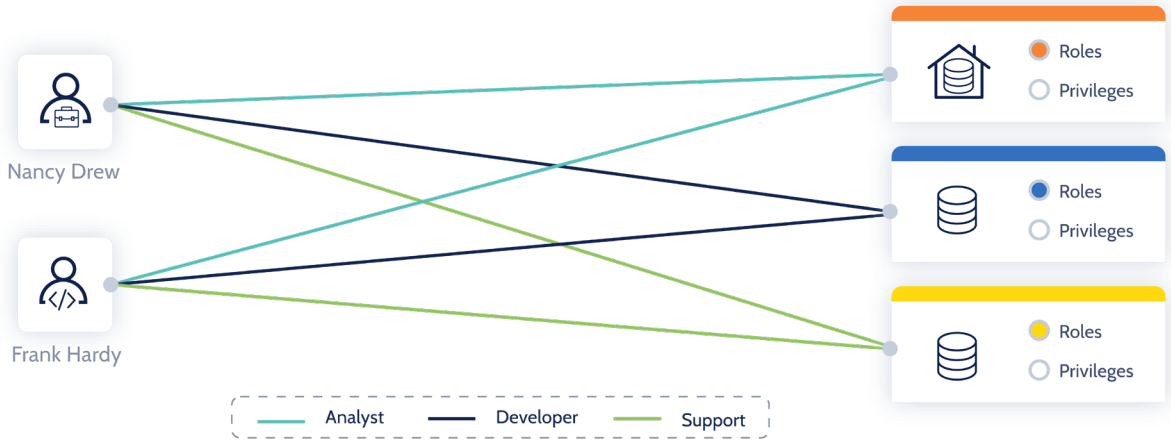
Security Assertion Markup Language (SAML) and OpenID Connect (OIDC) were developed to allow users and their permissions to be managed in one central identity provider, such as Active Directory. Then, users could log into other applications using their already-known identity and permissions.

This was not only a matter of convenience, but a matter of security. It prevented drift between the users and their known permissions across services. For example, if a user left a company not using SAML or OIDC, they would need to be removed from Active Directory, their MySQL user might need to be manually deleted, and their MongoDB user might need to be deleted. This obviously created a higher workload, and opportunities for error.

Instead, once identity and permissions were centralized, handling a person's departure became less risky. Under a centralized model, when a user leaves the company, deleting their account in one location has a cascading effect of removing them from every system to which they have been granted access. This was SAML and OIDC's appeal. Ideally, by using this approach, not only would companies centralize identity management, but they would be able to easily link the idea that a user like "noahcolley@example.com" was also the database user "noahcolley".

However, many databases continue to lack native SAML or OIDC support. That's because the two protocols are largely intended for web-based access patterns, and databases are designed to be accessed through queries, issued by various users and applications.

As a consequence, access management for the Data Mesh is still stuck in the stone age. Organizations usually give database and data warehouse access to users using shared accounts. Or they go through the pain of creating and maintaining individual user accounts for database access, which quickly becomes untenable.



Service Accounts for BI Tools and Applications

Commonly, data is analyzed through business intelligence tools and applications. These allow a somewhat non-technical user to transform data into digestible information through charts, graphs, statistical analysis, and pivot tables.

Usually, a BI tool is configured to provide connections to a database using a common database URL, username, and password—or essentially, a service account. Users, in turn, are configured to be allowed to use the BI tool. While most BI tools advertise their ability to relay the user's identity to the database, at the end of the day, these tools rely on many users sharing the same service account in the database. This introduces a gap in the audit trail, making it difficult to attribute actions to users. To deal with this, organizations adopt creative ways to relay the user identity. For a secure organization, the user's identity must be logged, so they typically add a SQL comment that includes the user's ID. To complete the audit trail, they ensure database logging is enabled (which often comes with its own performance challenges).

While this does accomplish the task of logging the user's ID as it's known to the tool, this still presents a challenge. To track a user's activity, their ID within the tool must be known. Also, if a company would like to introduce custom logic between the tool and the database, it's not possible because of the direct connection. For example, if a company wanted to use anomaly detection to interrupt a user's session when their activity looked like a data breach, it would not be possible.

Because a BI tool's service account is essentially shared across users, accounts are often over-permissioned. That is, when a single account is shared by any two users, it must have the sum of all the permissions that both those users need. When we scale this out to an account shared among many users, that account becomes over-permissioned, giving any single user more rights than he or she needs. This is at odds with the principle of least privilege, a key element of the security stance in many organizations.

Despite these challenges, BI tools are central to an organization's ability to turn data into a valuable asset. As a consequence, even for modern data repositories like Snowflake, BigQuery, and others that support SAML-like protocols, access will continue to be granted through service accounts, resulting in inability to centrally govern access to all users.

The Data Mesh is not safe when using traditional methods

In a traditional approach where a company runs its own on-premises database, it's common to configure a database to be protected by the same physical firewall as its users. If not, a jump host may be used to allow access to a more secure area. Deep packet inspection may be used to secure traffic. It may be possible to create a list of allowable traffic sources, such as allowed IP addresses or network CIDRs. To provide both security and disaster recovery, a nightly backup tape may be generated and sent to a separate location.

Another valuable aspect of typical on-premises database policies is that they include a database security review. While this can mean it takes much longer to create a new database (a new database must go through traditional channels that typically involve a company's security organization to perform checks and audits) it allows security to be a central consideration when delivering data.

This all changes when data is hosted on the cloud. Many of these traditional approaches are no longer possible in a dynamic cloud environment. A Virtual Private Cloud (VPC) is intellectually similar to a firewall, but access restrictions based on a user's physical location are more difficult to create. It's so trivial to create new data repositories that there can be a proliferation of data services, making it difficult to take appropriate security steps before or during a database's creation or lifecycle. And with the adoption of databases and data warehouses with an as-a-service model, none of the traditional rules apply!

An Access Management and Governance Model That Works

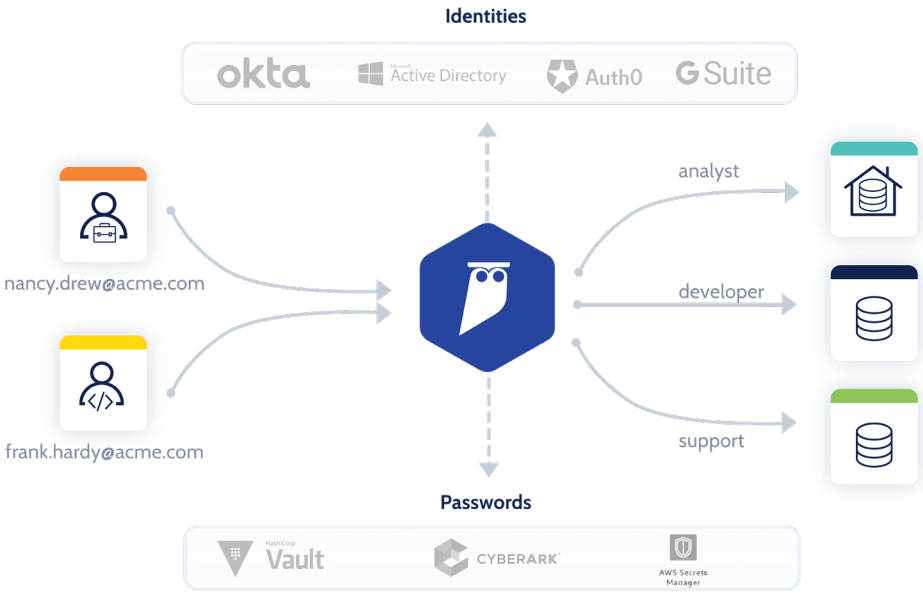
We believe the future of data is in its democratization, where any piece of information is readily available to those who are allowed to access it, and where IT organizations can replace tedious, manual handcrafting of infrastructure with code. In that world, access to all data comes instantaneously but with complete visibility and control. At Cyral, we created a technology that empowers organizations to see, control, and protect every piece of your data in the Data Mesh – all without impacting performance and agility. Cyral has built a featherweight, stateless interception service that can be easily deployed in the customer's environment, and which can intercept all requests to any structured or semi-structured data repository. We call this the Data Mesh Sidecar.

Identity Federation

Cyral provides identity federation by coordinating and managing user identities across an organization's data repositories and identity providers (IdPs). This allows users to authenticate with data endpoints using their IDP credentials. Rather than using a data endpoint's account name and password, federated authentication relies on the user's email address and a short-lived access token generated by their IdP.

The access token is generated after a successful SAML/OIDC authentication exchange between the user and the IDP. Subsequently, the user can pick a client (a CLI, UI or BI tool) of their choice to connect to the data endpoint via the Cyral Sidecar. When the sidecar receives the email address and token from the user, it verifies the authenticity of the email address and the validity of the token with the IDP. After these checks pass, the sidecar looks up the group that the user is a member of in the IDP, and maps the group name to a specific data endpoint account that is to be used for all members of that group. These identity mappings are created by an administrator and maintained in the Cyral control plane.

After determining the account to be used, the sidecar looks up its credentials in a secrets store running in the organization's environment (for example, an AWS Secrets Manager or Hashicorp Vault instance running in the organization's AWS account). Using these credentials, the sidecar connects to the data endpoint, thus allowing the user to access the data endpoint without needing dedicated credentials in the data endpoint itself.



Service Role Disambiguation

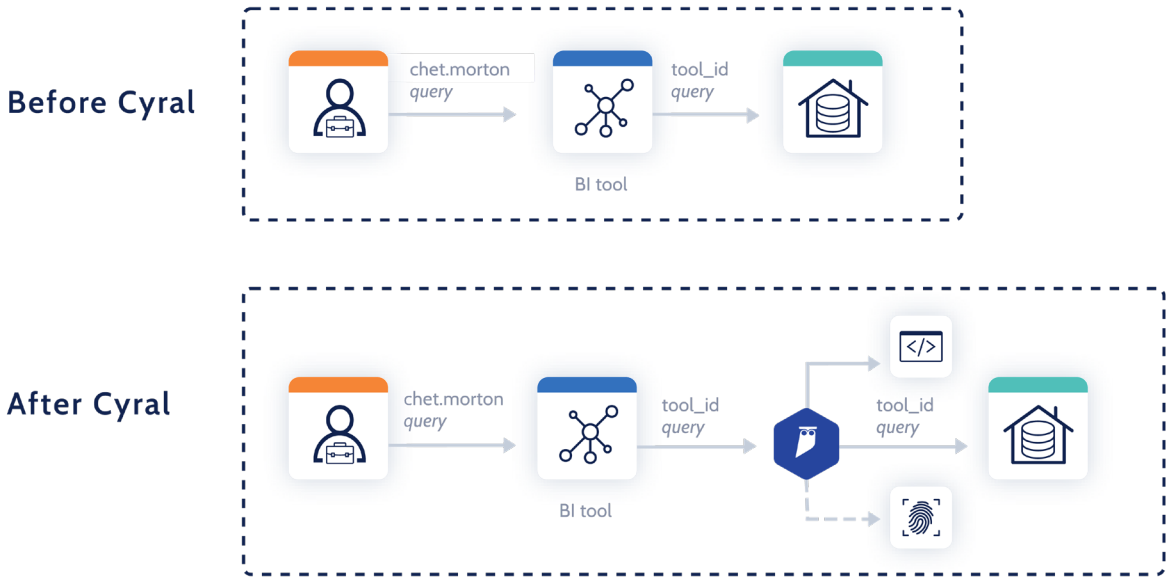
Applications and SaaS BI tools use various techniques to provide more visibility into the requests being executed. Request annotation is one such common technique which involves passing in comments that carry additional information about the end user's identity in the data endpoint's native language. Another common technique is setting custom session variables scoped within database transactions.

Both comments as well as custom session variables are ignored by data endpoints during request and transaction processing. However, they're useful for activity monitoring and performance debugging, and they help with tracing requests back to the end users that generated them.

Cyral understands the syntax and grammar of annotations and session variables used by popular SaaS tools such as Looker, Periscope Data, and Tableau, as well as data access API frameworks such as PostgREST, Hasura, and MongoDB Realm, which provide rich REST and GraphQL interfaces for database access.

Cyral uses this knowledge to extract the end user's identity and adds it to the generated data activity logs. Cyral can also determine the end user's group membership in an identity provider.

Together, the end user's identity and group membership are used for granular policy enforcement, which is otherwise not possible in the data endpoints.

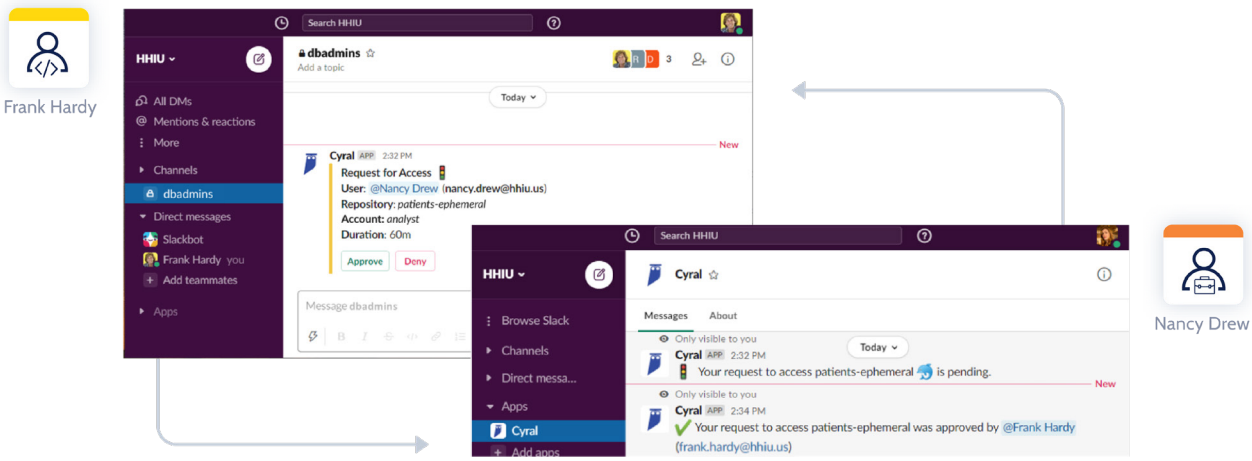


Privileged Access Management

To create a robust Data Mesh Security model, it's not sufficient to deploy just centralized account management and identity federation. Privileged accounts represent a frequent vector of attack in most organizations. Industry researchers and experts invariably point to compromise of privileged accounts as the main technique for access to the crown jewels of data in an organization.

Type	Description
Brute force	Using an automated script to rapidly try multiple potential passwords in succession until a correct one is found. Includes dictionary attacks.
Social engineering	Exploiting our trust of others through impersonation. For example, impersonating a CEO with a brute-forced password to ask the administrative assistant to wire money. Includes phishing.
Insider attacks	A person working for the company uses intentionally-granted credentials to misuse their privileges. Includes one employee using another's credentials that have been shared or leaked, and misuse of service accounts.

A benefit that Cyral provides customers is the ability to grant users only just-in-time and just-enough access to their mission critical and sensitive data repositories, with simple access requests and approvals in the organization's existing messaging apps. This eliminates the need for long-term privileged access for users, tightens up access control, and reduces the attack surface.

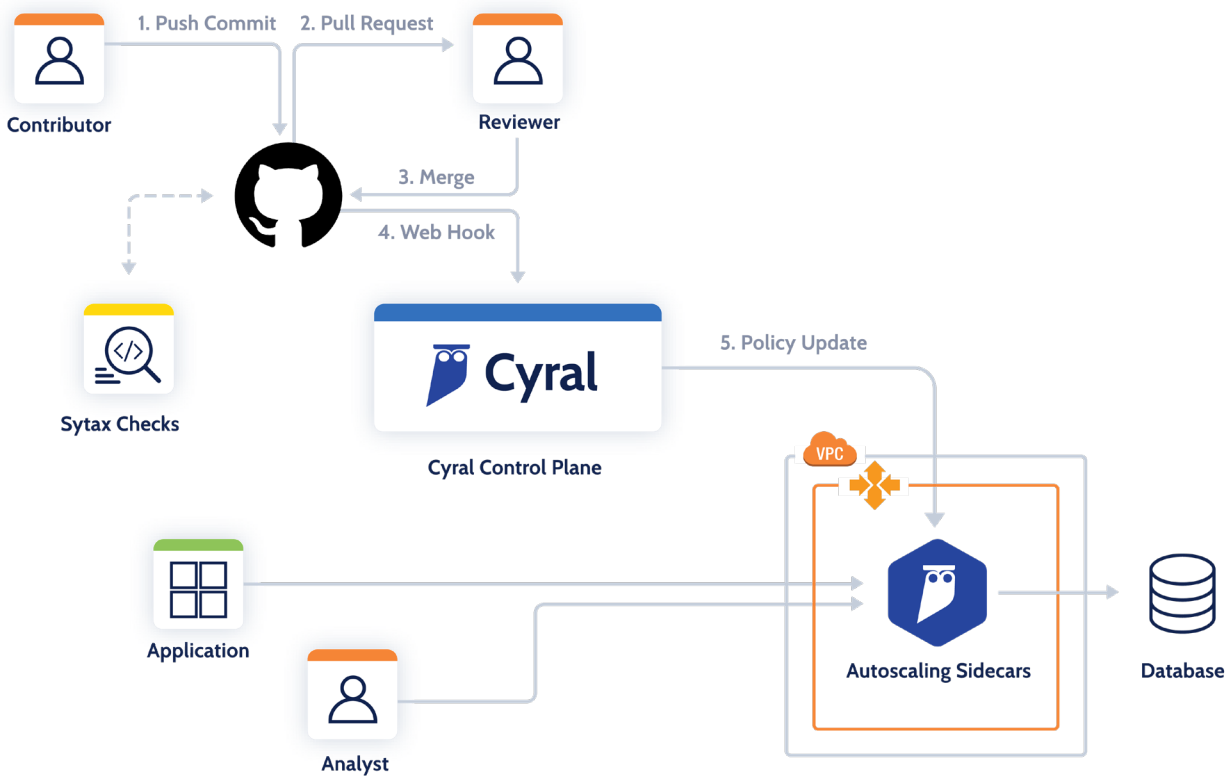


Security as Code Model

Just as Infrastructure as Code enables infrastructure to be created, managed and scaled using code, Security as Code allows organizations to codify their security and policy decisions. Embracing a Security as Code model allows developers, DevOps, and security teams to implement security testing and scans directly in their CI/CD pipelines and codify access policy decisions.

Cyral was designed with CI/CD integration in mind, to fully automate security for data repositories—all done directly in an organization's existing CI/CD pipeline. Cyral also integrates with code versioning services like GitHub, enabling customers to automate deployment of data security policies and keep an audit log of all policy contributions.

Cyral provides a simple YAML-based policy syntax for customers to specify their access control policies. Integration with services like GitHub means anyone in the company can review the policies, and there is a complete audit trail of who granted which privileges to whom.

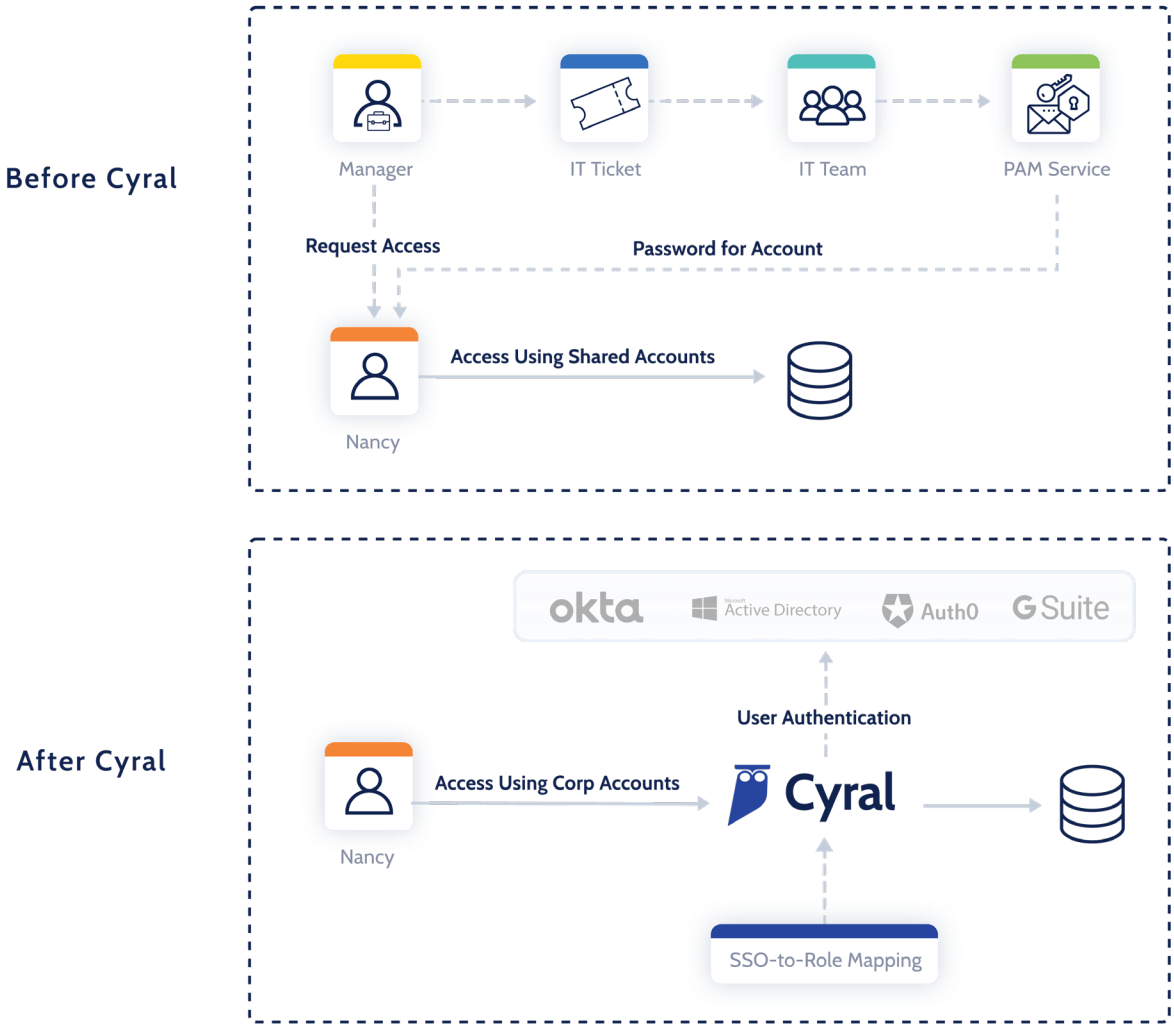


Benefits of Identity-Based Access Management

We have seen three main benefits for organizations that adopt an identity-based access management model for their Data Mesh: simplified access management; centralized compliance controls and audits; and a reduced attack surface. Below, we summarize each in turn.

Simplified Access Management

As new users need to be granted access to data repositories, administrators can just use their existing SSO information to govern access, without having to manage accounts in an ad hoc way.



By moving to Cyral, organizations can

- Use their identity provider as the primary authentication service
- Eliminate access to service account passwords and other shared credentials
- Use a single service to grant, revoke, and modify access privileges

Centralized Compliance Controls and Audits

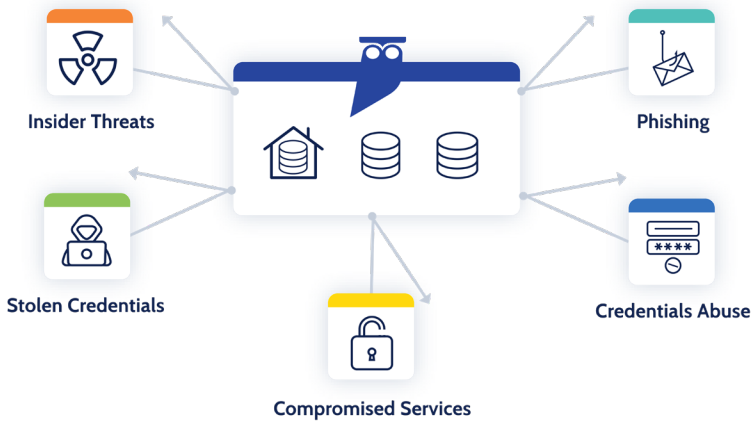
Cyral enables cloud-first security and IT teams to drive compliance by providing them with automated controls and system generated reports that simplify user access management and change management.

Encryption Benefits	Description
User access provisioning	Access to a user can be specified through properties in the identity provider Native data repository credentials can be secured in secrets managers
Ephemeral access management	Temporary access may be given to any user for fixed period of time Additional constraints may be applied on standard roles
Least-access enforcement	With access based on user groups, each group receives only the access it needs Users who don't need frequent access can quickly get a temporary token by requesting one in Slack or their usual messaging client
User access deprovisioning	All access can be revoked centrally from IDP Data users never have native data repository credentials
User access review	Ability to drill down on all access for each user, attributed to their enterprise user account Compare access to SSO group and past activity
Policy change authorization	Require approval for every change in data access policy Maintain a complete audit trail of all policy changes
Audit logging	Automatically detect and log suspicious activity directed at repositories Show the responsible enterprise user for each logged event Log activity on specific sensitive tables, collections, columns, or fields Use policies to determine which events are logged

Reduced Attack Surface

When an organization doesn't have simple ways to manage data access, the result is more administrative complexity: more manual steps that team members must take to request access, approve access, and manage access policies. This administrative complexity—and the workarounds that staff come up with just to get their work done on time—results in unnecessary privileges, unused accounts, and a lack of visibility into who has access to what and who can grant access to what. Together, these issues enlarge the attack surface of the organization, exposing the team and its data to greater risk.

Organizations reduce this risk when they adopt identity-based access management. By giving their team tools to easily regulate access to critical data services, the organization avoids elevated privileges and murky approval paths, minimizing the surface area available for attackers.



Summary

Data Mesh architecture and Infrastructure as Code are two inevitable shifts for companies of all shapes and sizes. The lack of support for IDP protocols like SAML and OIDC for data repositories, however, makes it very hard for organizations to simultaneously be agile, data-driven, and secure.

Cyral provides a Security as Code approach to easily govern all access to the Data Mesh. By statelessly intercepting requests to every data endpoint, Cyral is able to provide identity federation across all data repositories, BI tools, applications, and identity providers.

With inline, real-time identity federation and a Security as Code operating model, customers can enforce comprehensive security for their Data Mesh and fully embrace Zero Trust.

About Cyral

Cyral delivers enterprise data security and governance across all data services such as S3, Snowflake, Kafka, MongoDB, Oracle and more. The cloud-native service is built on a stateless interception technology that monitors all data endpoint activity in real-time and enables unified visibility, identity federation and granular access controls. Cyral automates workflows and enables collaboration between DevOps and Security teams to automate assurance and prevent data leakage. Cyral is venture-backed by Redpoint, A.Capital, Costanoa and SVCI. Follow the company on Twitter at @CyralInc.

cyral.com/tech-talk