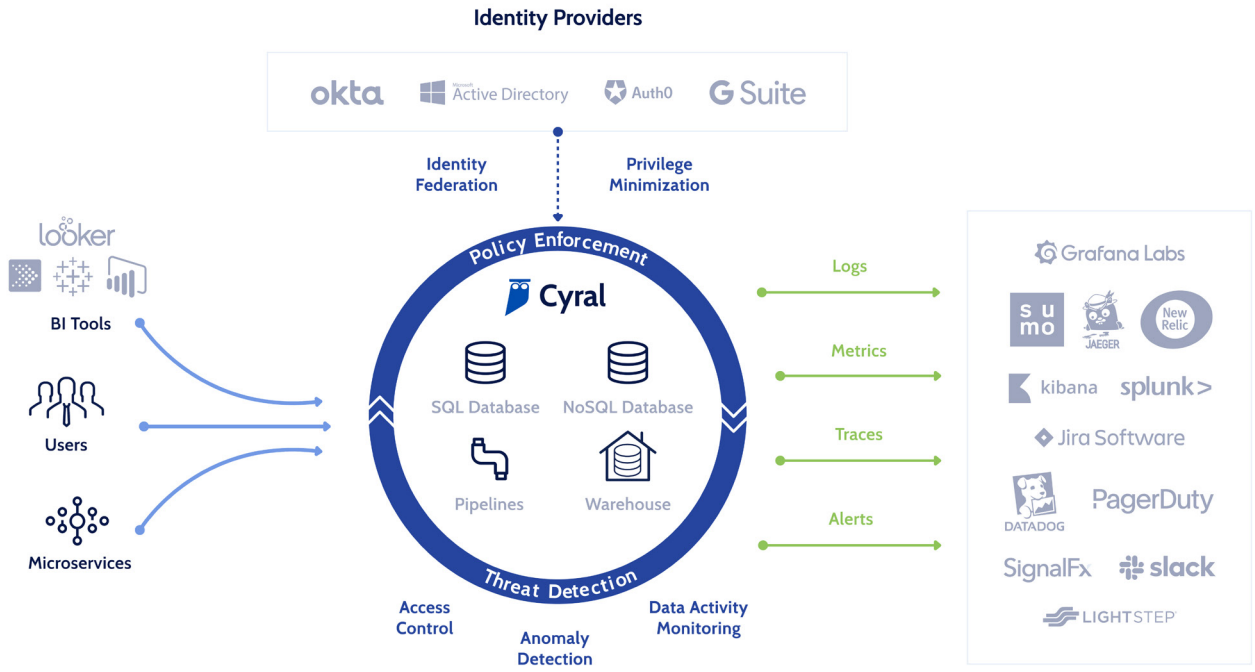
Cyral | WHITE PAPER

Cyral: Designed for High Availability

Contents

Security shouldn’t hurt reliability

The Cyral service—specifically the Cyral sidecar—operates in your data path, intercepting every request in order to monitor and protect your data repositories. This monitoring and protection is crucial, but it can’t come at the cost of interrupting your users’ access to data. To ensure you can protect your data while continuing to give users and applications uninterrupted access to it, we’ve designed Cyral as a highly available service. Cyral is built to sustain failure without interrupting the flow of data and without compromising the security the service provides. For sites where data availability is the top priority, Cyral can be run in a fail-open mode.

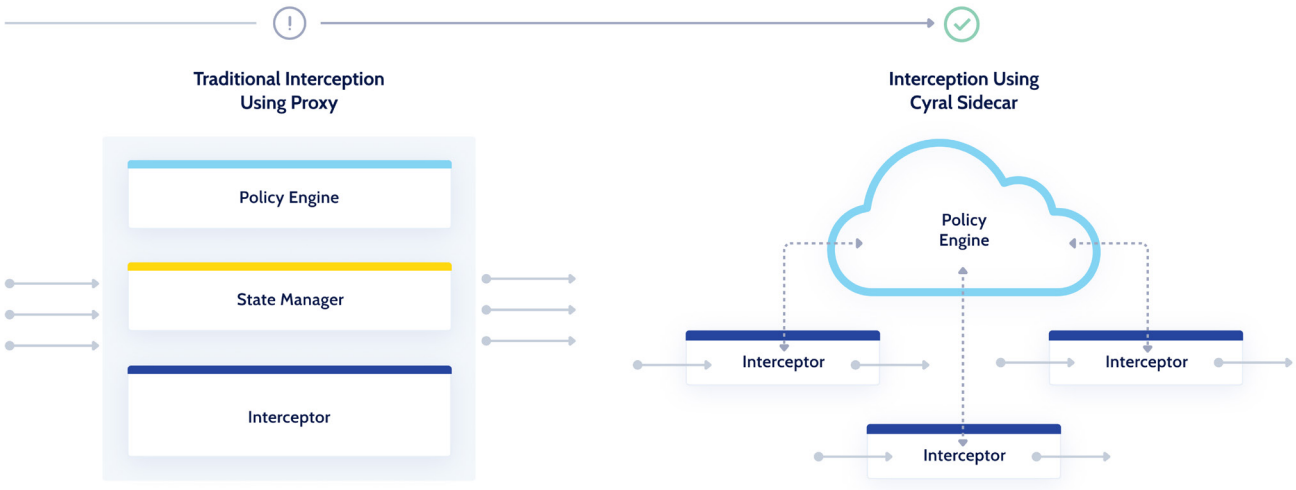


Designing for High Availability

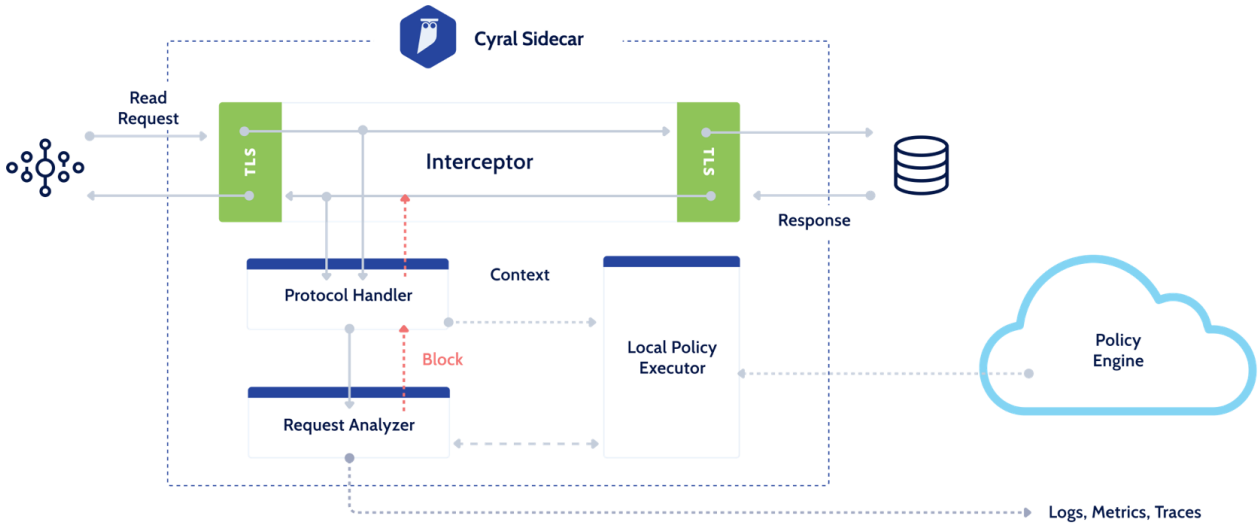
High availability refers to a design that avoids service loss by managing failures and minimizing planned downtime. A highly available service has no single point of failure, so that the failure of any single component within the service doesn’t render the entire service unavailable.

Since the Cyral sidecar sits in the datapath and intercepts all requests inline, Cyral was designed from the outset as a high availability service. The key to achieving this was building a featherweight, stateless proxy that your DevOps team can deploy easily in your cloud or data center environment. We call this a data layer sidecar, and it has the following characteristics.

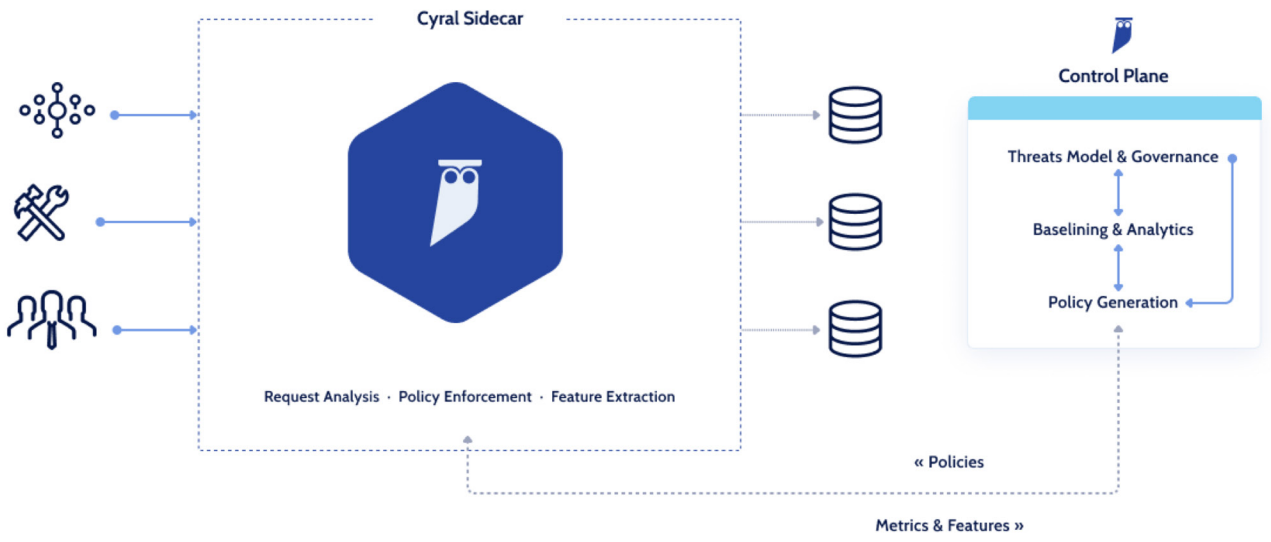
1. Stateless – Unlike traditional application proxies, our sidecar defers all session state management to the data layer connections themselves. This elegant design allows multiple sidecars to be deployed in a high-availability configuration and enables a true fail-open design.



2. Output Filtering – One key insight behind our sidecar is that it can pass read requests to the data layer without any delay, while blocking their corresponding results if the request is determined malicious or disallowed. This analysis of the request happens asynchronously, while the data layer is processing it in parallel, allowing the original read operation to happen without any extra delay.



3. SaaS-based Control Plane – Our customers can deploy sidecars in several different ways, and easily administer them using a SaaS based control plane. All integrations and provisioning can be managed centrally from here. It offers intuitive workflows to implement security policies and react to threats.



Cyral sidecars can be deployed in a customer's cloud or on-premises environment as a Kubernetes service, an autoscaling group, a cloud function or via a host-based install. All the data flows and sensitive information stays inside the customer's environment where the sidecar is deployed, creating no risk of spillage.

The sidecars can be deployed in a high availability configuration fashion. Because sidecars are stateless when one instance goes down another can take over for it instantaneously with no application disruption. Because of its lightweight nature, a new sidecar can be added back quickly since fewer instances of the sidecar are now handling the same request volume. The Cyral architecture ensures graceful failover. The points of failure to be considered are the Cyral management console and sidecars.

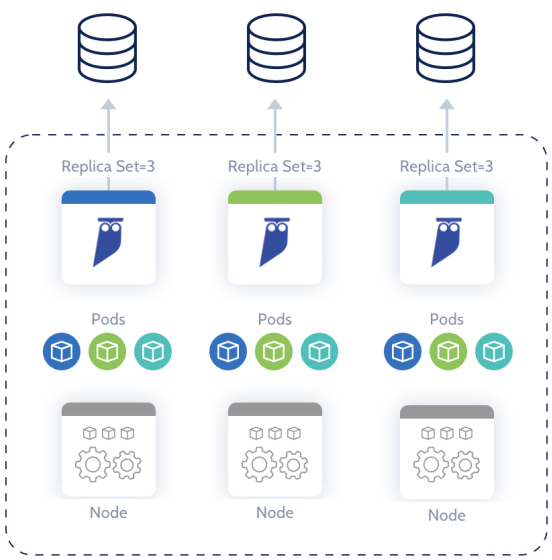
If the Cyral management console fails, all sidecar operations continue uninterrupted. Cyral platform APIs can be used to monitor activity and health. Administrative actions via the web UI or APIs cannot be performed until the management console is restored. Learn more about the Cyral platform APIs at www.cyral.com/docs.

If a Cyral sidecar fails, its responsibilities are normally assumed by another sidecar in its high availability group. The applications typically see no disruption. See the sections below for details.

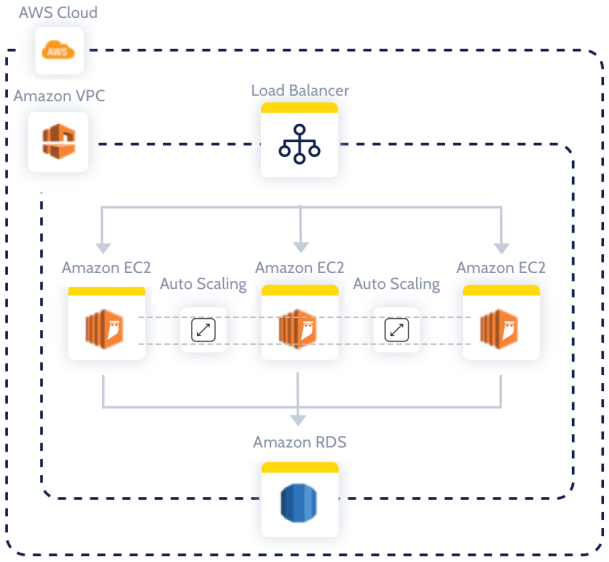
Cyral Deployment for

Highly available sidecar deployment

Cyral sidecars can be deployed in a highly available configuration both on premises and in the cloud. This can be done in multiple ways depending on the infrastructure used:



In a Kubernetes environment high availability is achieved via ReplicaSets. The diagram below shows Cyral sidecars deployed in unique namespaces with ReplicaSet = 3.



In a cloud or virtualized environment high availability is achieved via auto scaling. The diagram below depicts using AWS Auto Scaling to create multiple instances of a Cyral sidecar.

In such setups, if a sidecar instance incurs a denial-of-service (DoS) attack or fails for any reason, traffic will be routed across the remaining instances. A new sidecar instance will automatically start to take the place of the instance that went down. This new instance will immediately start handling traffic after coming up. In this manner, applications experience zero disruption.

Sidecars support passthrough mode

Cyral supports “passthrough” mode. In this mode, the sidecar acts as a layer 4 load balancer and forwards requests through to the data endpoint. All analysis, logging, policy evaluation stops. Passthrough mode can be configured via the platform APIs (Please refer to www.cyral.com/docs for details.) Passthrough mode is ideal for troubleshooting performance problems.

Sidecars support fail-open configuration

Cyral sidecars can be deployed in a fail-open configuration. This ensures that if the sidecar becomes unresponsive, the data repository remains available, though without the monitoring and/or protection provided by Cyral.

This configuration requires setting up active/passive DNS routing failover in your organization's DNS service. For example, in an AWS deployment, the Amazon Route 53 DNS service can be set up in active-passive failover mode with the Cyral Sidecar as the active member and the RDS instance itself as the passive member. During normal operation, Route 53 routes all traffic to the Cyral sidecar. If the sidecar fails, Route 53 routes traffic to the address listed as the passive member, namely the RDS instance. Since client applications are typically built to retry requests when a connection is dropped, the app will retry until a connection is made and will experience minimal to zero disruption. The diagram below illustrates how this will work.

On Azure, similar capability can be achieved by using Azure DNS and Traffic Manager.

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.

Want to learn more about how Cyral can help your organization? Sign up for a tech talk with one of our engineers!

cyral.com/tech-talk