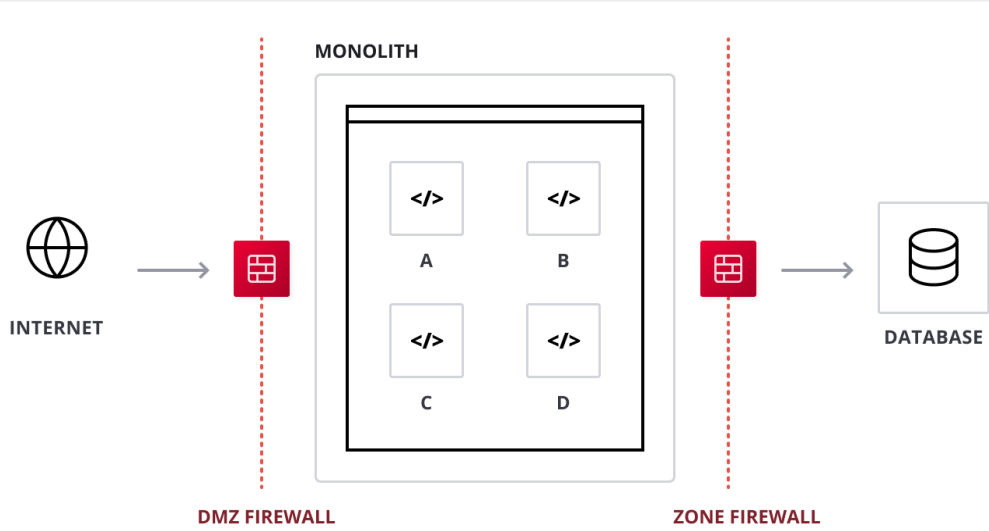

HashiCorp
Consul

Traditional application security

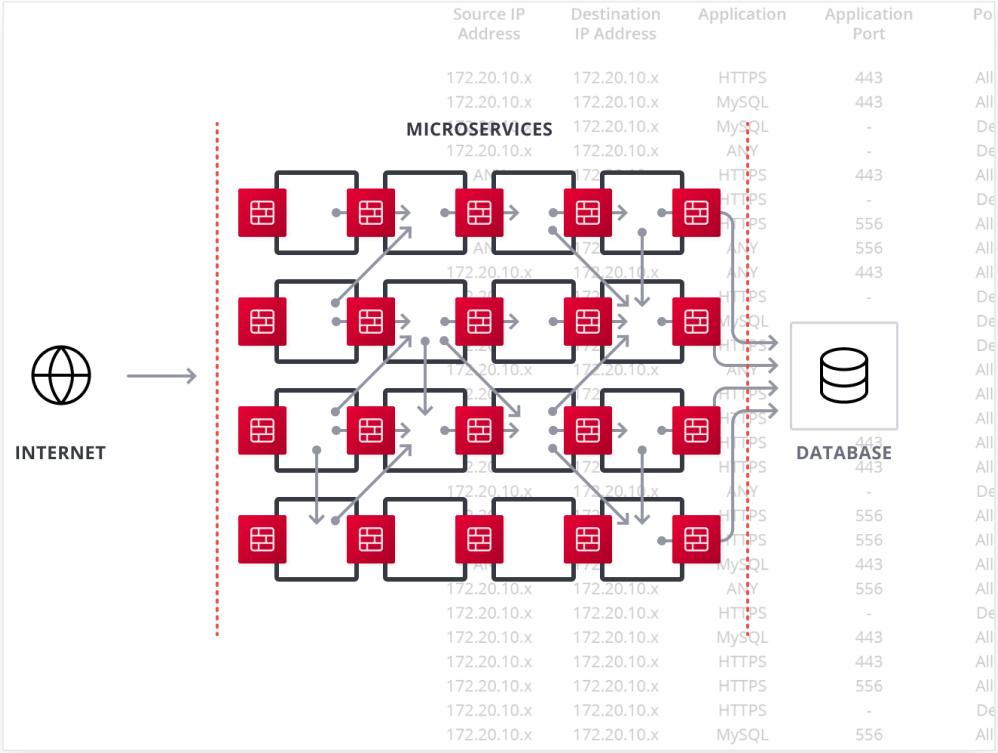
- Firewalls secure north-south traffic at well-defined perimeters.
- East-west traffic within the datacenter is assumed to be trusted.



ZONE	ZONE	SOURCE IP	DESTINATION IP	APPLICATION	APPLICATION PORT	POLICY
Untrust	DMZ	Any	172.20.10.x	HTTPS	443	Allow
DMZ	Trust	172.20.10.x	172.20.20.x	mSQL	443	Allow

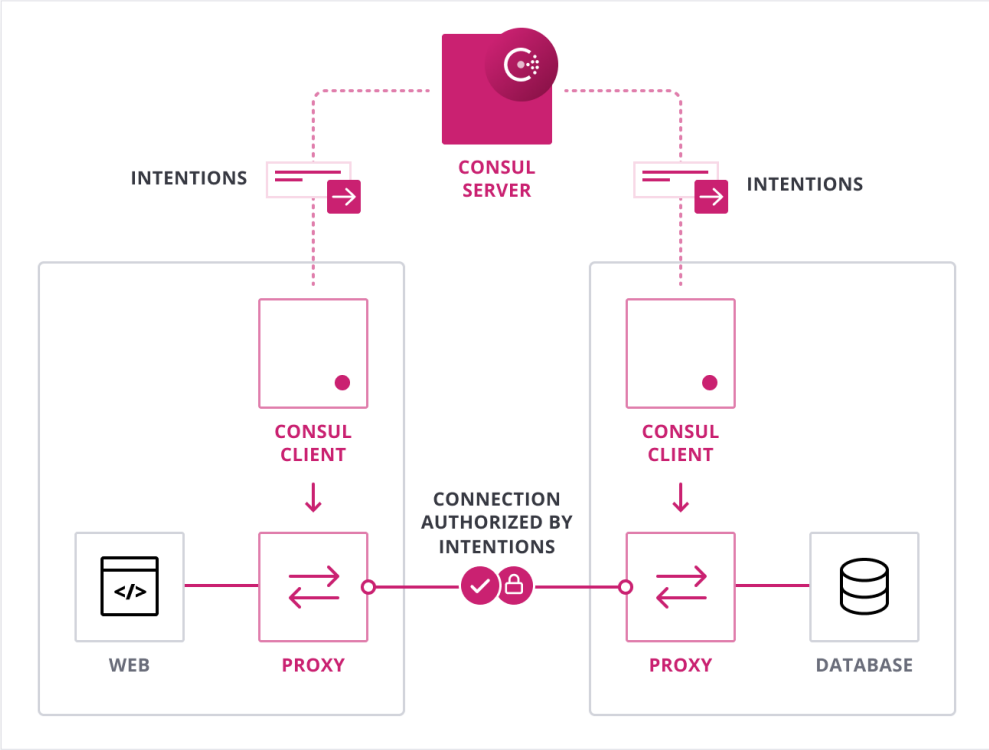
Complexity at scale

- Microservices and multi-cloud introduce complex network topologies and short-lived IPs.
- This dynamic nature results in security rules sprawl which is difficult to manage and increases the risk of misconfiguration.

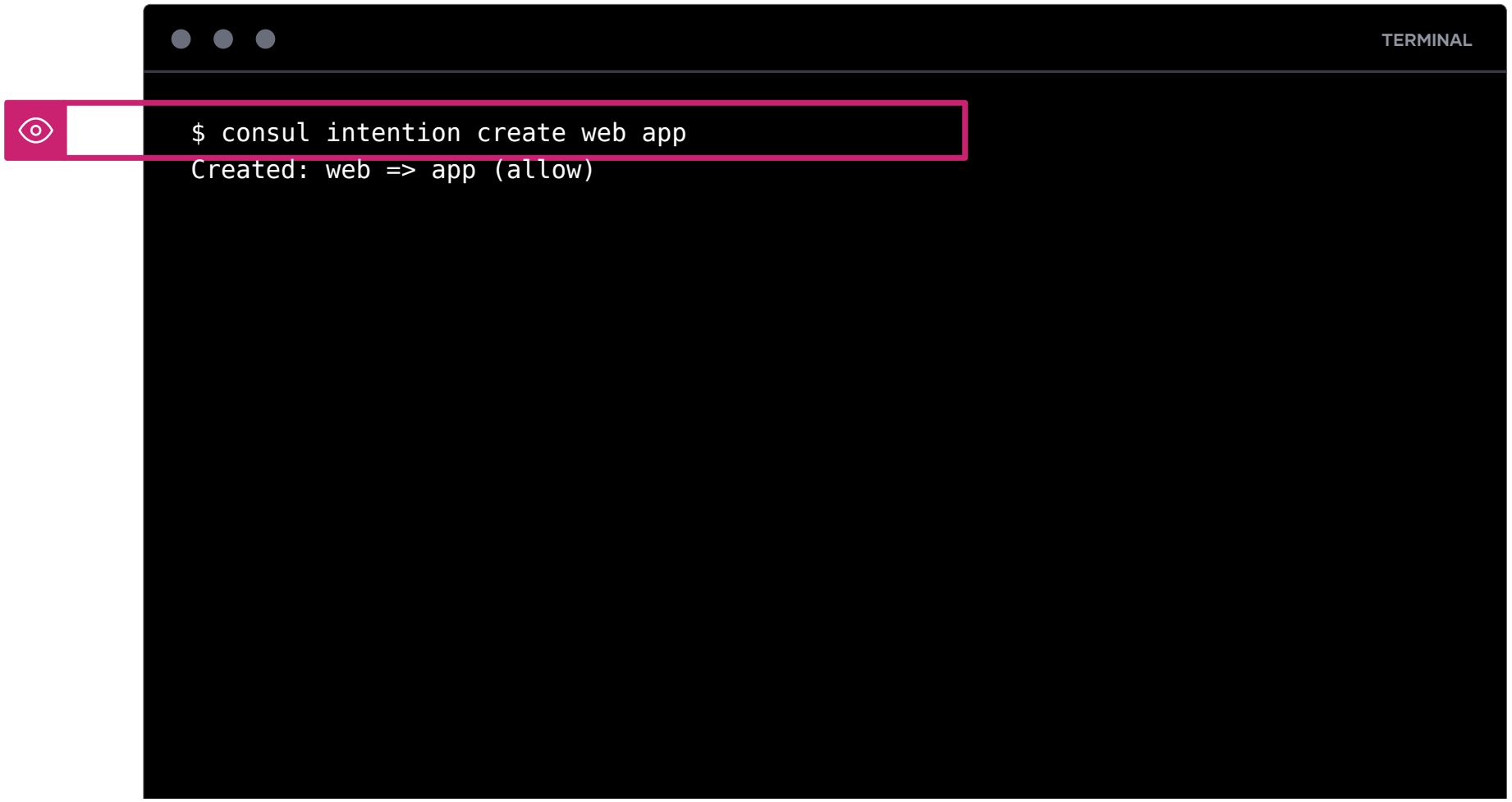


Sevice segmentation

- Services authenticate using identities.
- Mutual TLS secures the connection.
- Destination service authorizes connection by intentions.

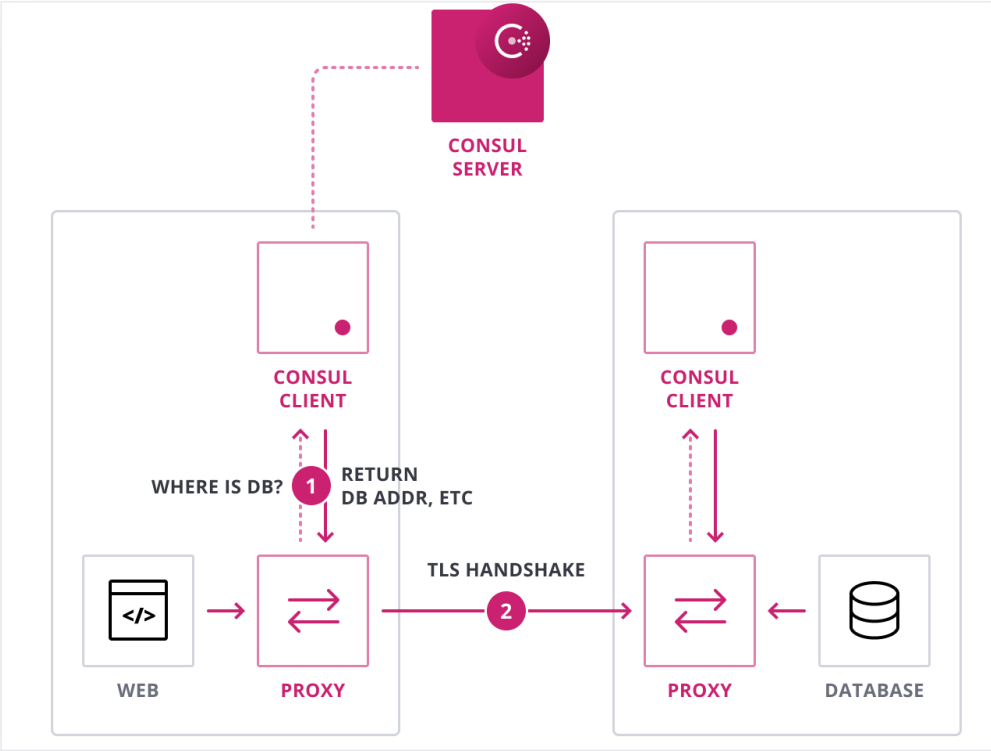


Allow web to talk to DB



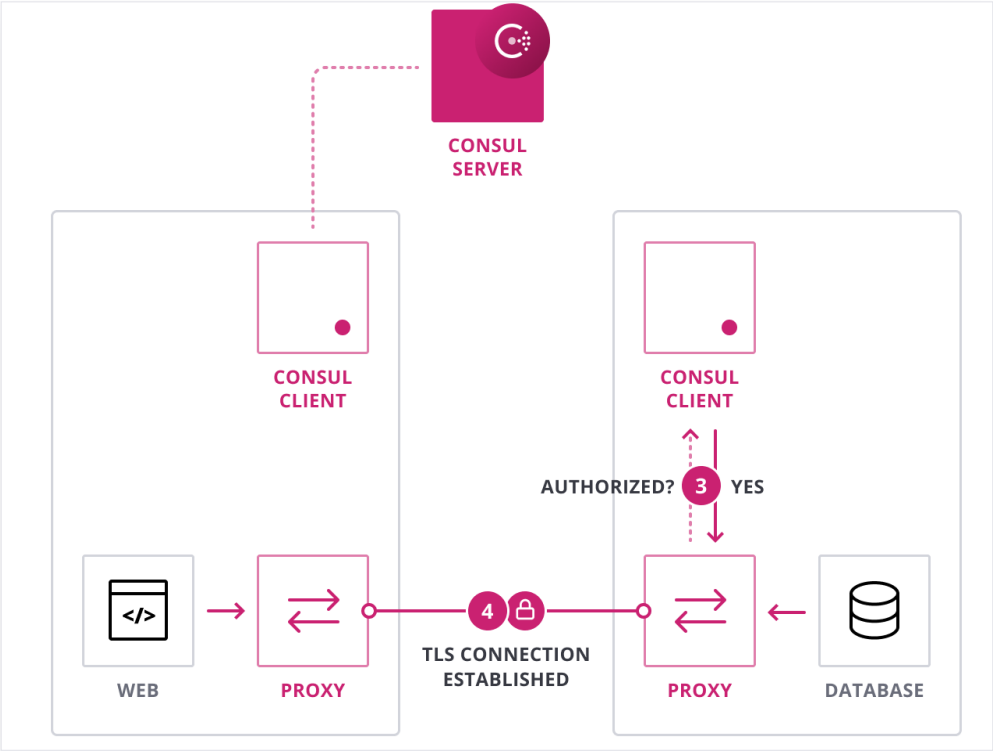
Connection establishment - 1 / 2

- Web proxy queries Consul for location of DB service.
- Web proxy initiates TLS session to DB proxy.
- Proxies mutually authenticate identity.



Connection establishment - 2 / 2

- Mutual TLS is established.
- The DB proxy sends the authorization request to Consul.
- Consul authorizes the connection based on intentions.



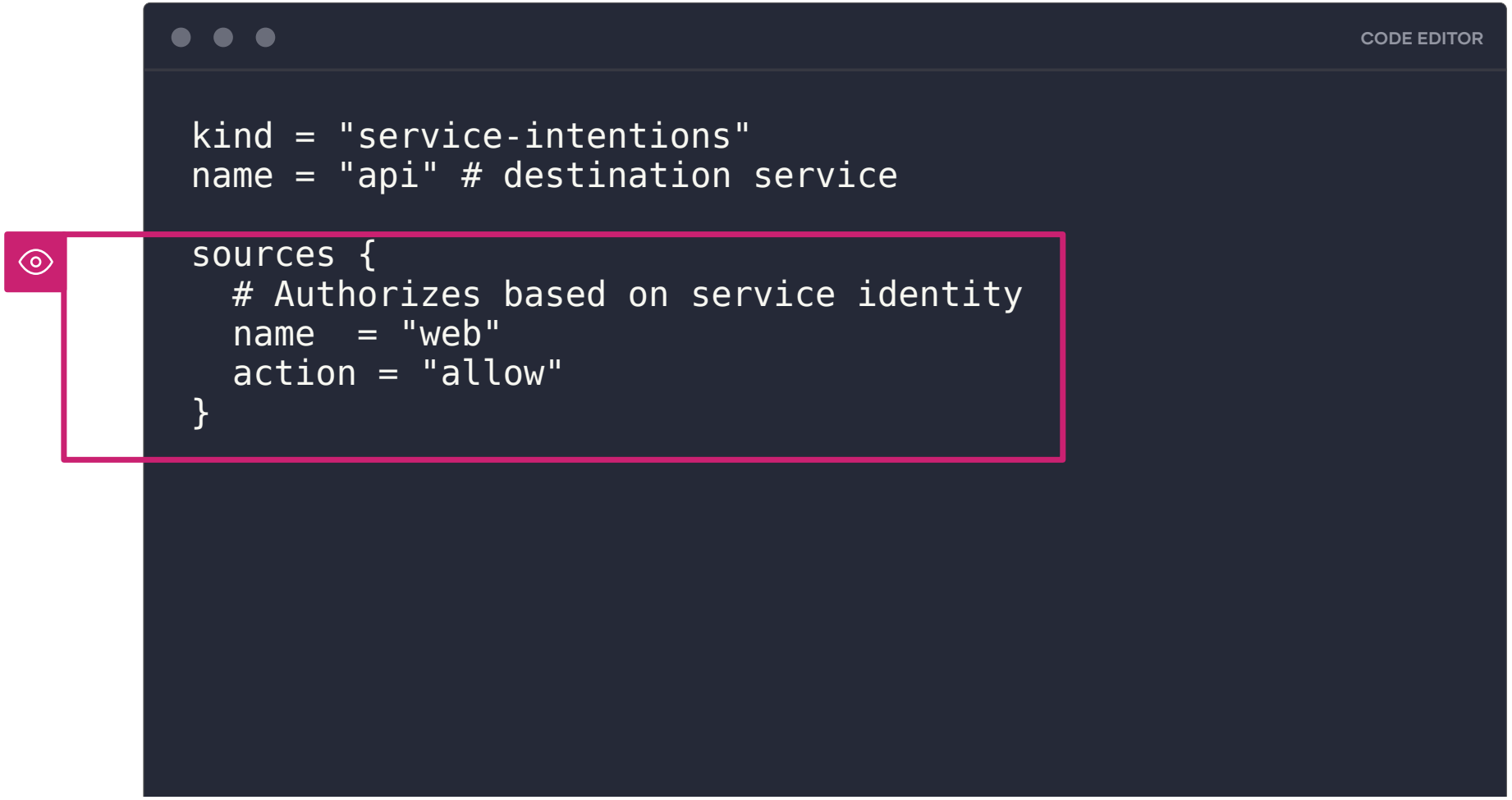
Challenges

1. Expensive, per-connection authorization callback to Consul agent.
2. Unable to express application-level security policies.
3. Inconsistent policy enforcement (e.g., mesh and in-app).

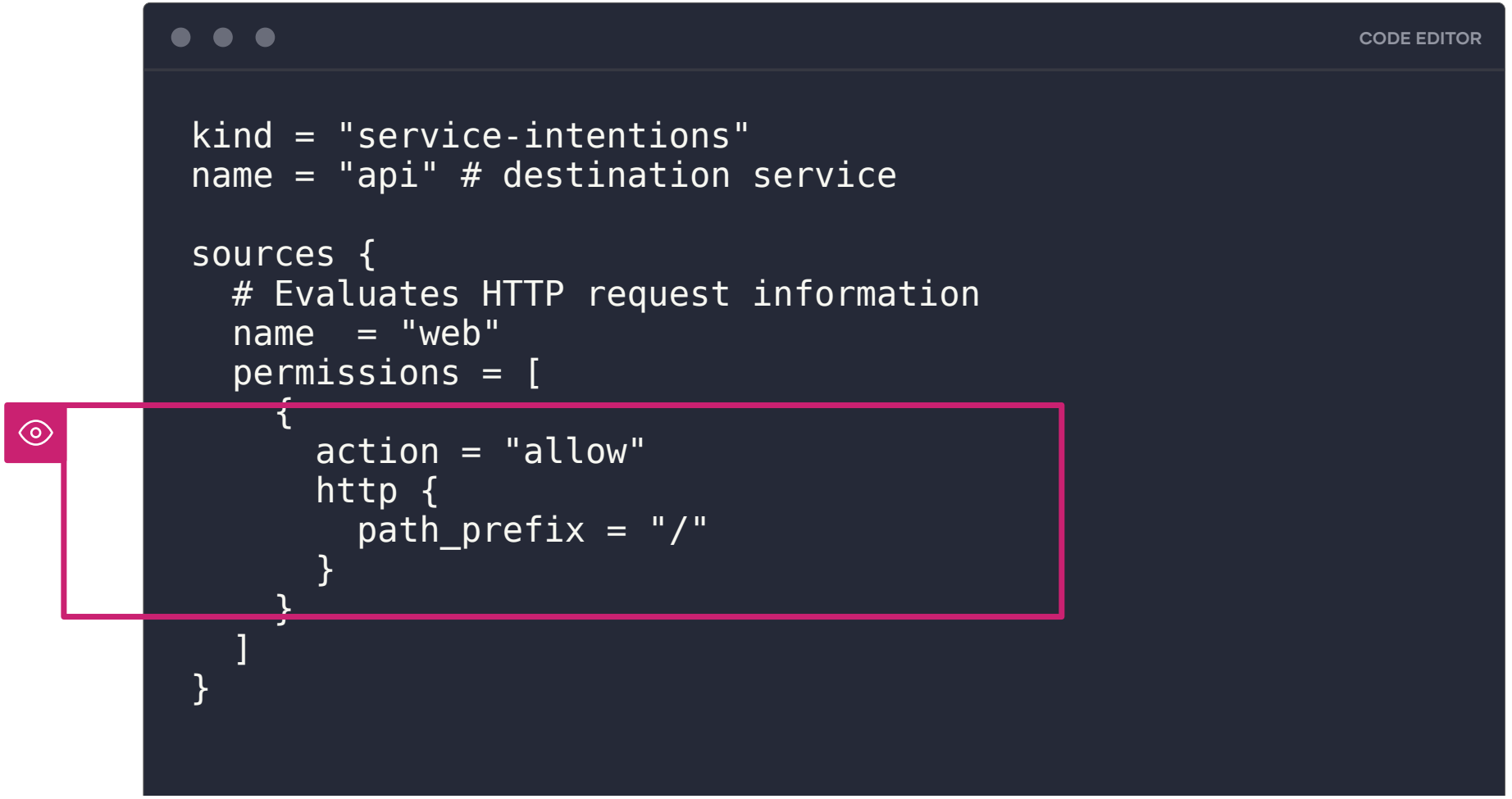
Intentions in Consul 1.9

1. Enforced in data plane proxy (Envoy).
2. Intentions as Consul Configuration Entries.
3. Application-aware (HTTP).

Service Intention



Service Intention



HashiCorp

Live Demo

Summary

Application-aware Intentions

Single workflow for defining security policies

Policy enforced in mesh for TCP and HTTP apps.

Higher request rates

Avoids per-connection auth callback to Consul.

Granular access control

Define least-privilege service security policies.


HashiCorp
Consul