Grey Matter Overview

Unstar(S)

What is a service mesh?

Microservices are a powerful and popular architectural pattern for modern cloud-based development and deployment. However, some activities are rendered *more* complex with microservices than under previous paradigms, such as announcement, discovery, instrumentation, logging, tracing, troubleshooting, encryption, and access control. The purpose of a “service mesh” is therefore to realize the promise of microservices by abstracting any extra complexity away from individual service developers, and into a “sidecar” proxy that can be paired with each service. The sidecar then provides the service with the common functionality required to behave as a good citizen of a microservice architecture, while the service performs the business-specific tasks for which it was written.

Grey Matter features overview

- Automatic service discovery and load balancing for ephemeral service instances
- Support for mTLS, OAuth/OIDC, and JWT-based user authentication and authorization, with support for impersonation by NPE whitelist
- Automatic mTLS between meshed services (with FIPS 140-2 compliance mode), point-to-point RBAC policy enforcement (with SPIFFE-compliant ephemeral and auto-rotating service certificates for complete “zero-trust” coming)
- Telemetry, monitoring, statistics, SLOs, and mesh state visualization, access logging, deep auditing, and observables
- HTTP, HTTP/2, and gRPC proxying, (TCP, UDP, QUIC, and Kafka support coming)
- Retries, timeouts, circuit-breakers, canary deployments, A/B testing, etc.
- Active and passive health checks
- Compatibility and interoperability with many third-party tools and platforms
  - OpenShift/Kubernetes, Consul, AWS, and flat-file for service discovery (with DC/OS Marathon in alpha)
  - Istio and Consul integration in the “other-service-mesh” category
  - Prometheus, Kafka/ElasticSearch/Kibana, CloudWatch, etc. for telemetry, observability, metrics aggregation, and event triggering
  - Proxying to external services such as legacy APIs, external services, off-cloud services, and Lambda services

- Multi-mesh self-interoperability with encryption, access control policy enforcement, data consistency, sharing, and communication between meshes
- Distributed, immutable, timestamped data storage with fine-grained access control, AES256 encryption for stored content, lineage tracking, file expiration, CDN features, and offline/rejoin/merge support with strong eventual consistency