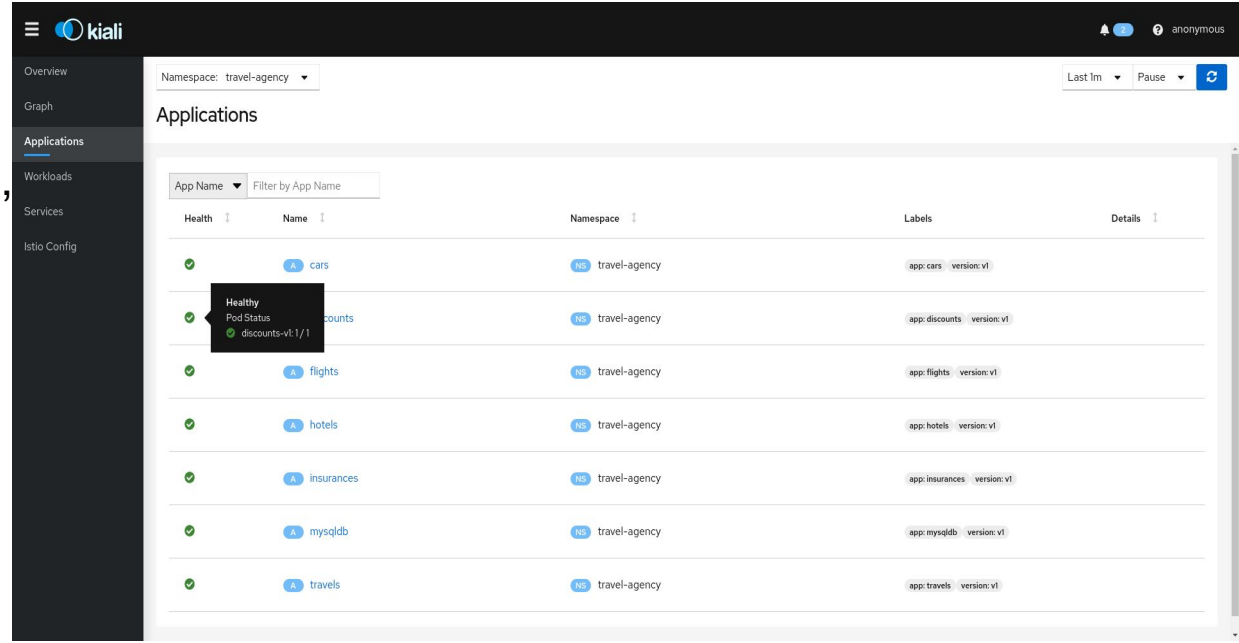
Kiali and Jaeger

Kiali helps to define, validate, and observe the connections and microservices of Istio service mesh.

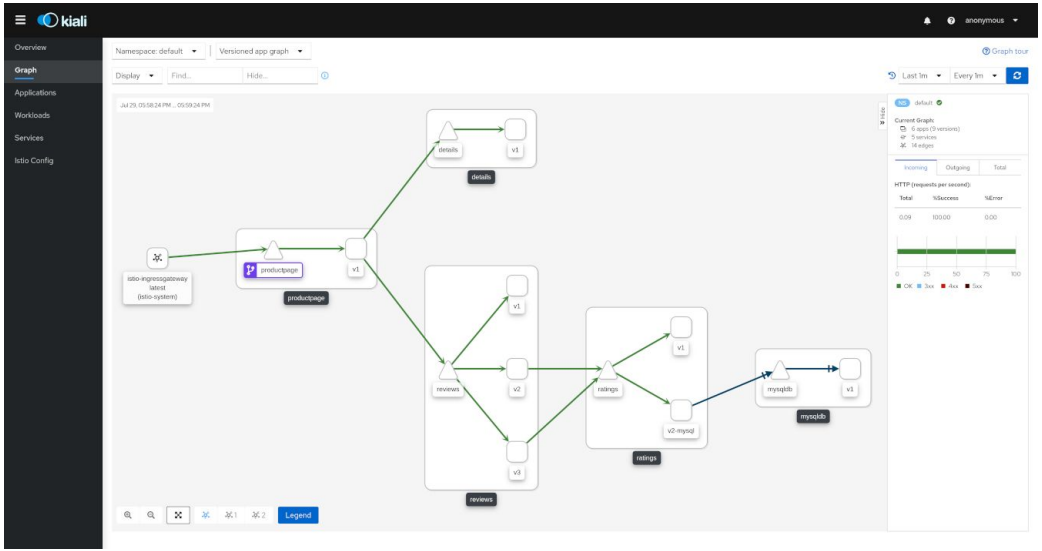
Kiali works with Istio in Kubernetes distributions. It visualizes the service mesh topology and provides visibility into features like request routing, circuit breakers, request rates, latency and more.

Kiali provides a summary page of the Service Mesh namespaces and the status of their Applications, Workloads and Services.



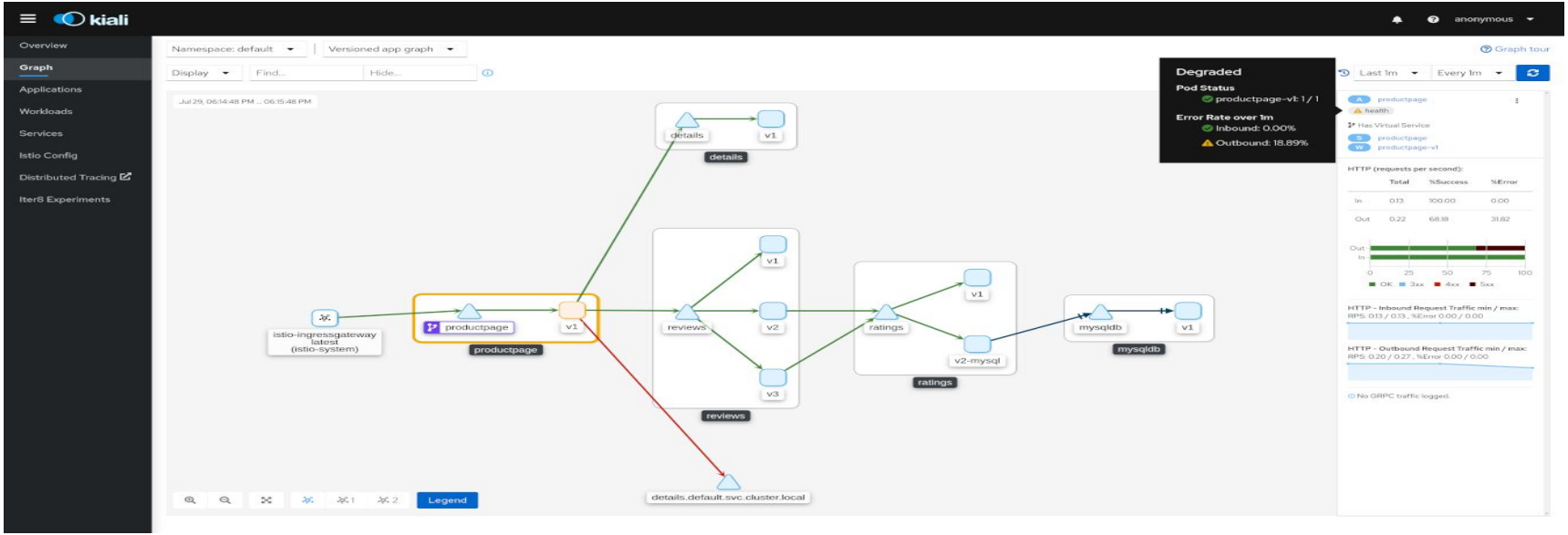
Observability Features

The graph provides a powerful way to visualize the topology of your service mesh. **It shows you which services communicate with each other and the traffic rates and latencies between them**, which helps you visually identify problem areas and quickly pinpoint issues. Kiali provides graphs that show a high-level view of service interactions, a low level view of workloads, or a logical view of applications.

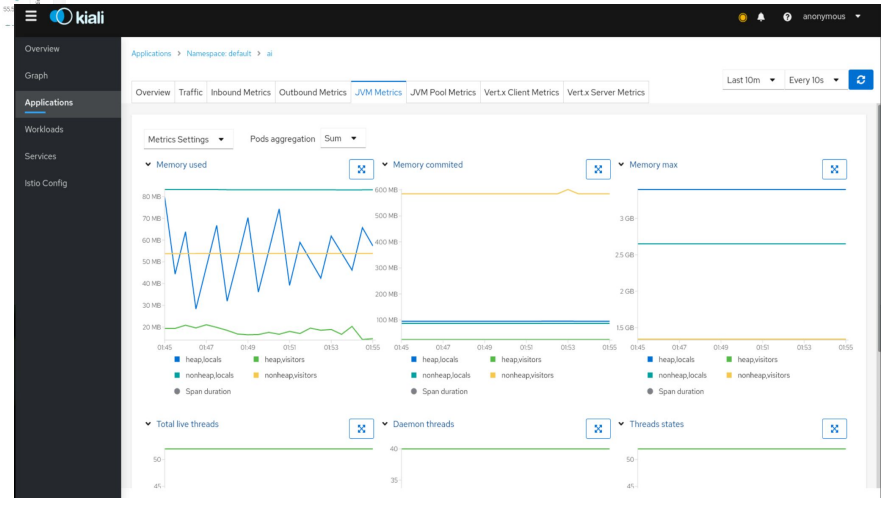


GRAPH: Health

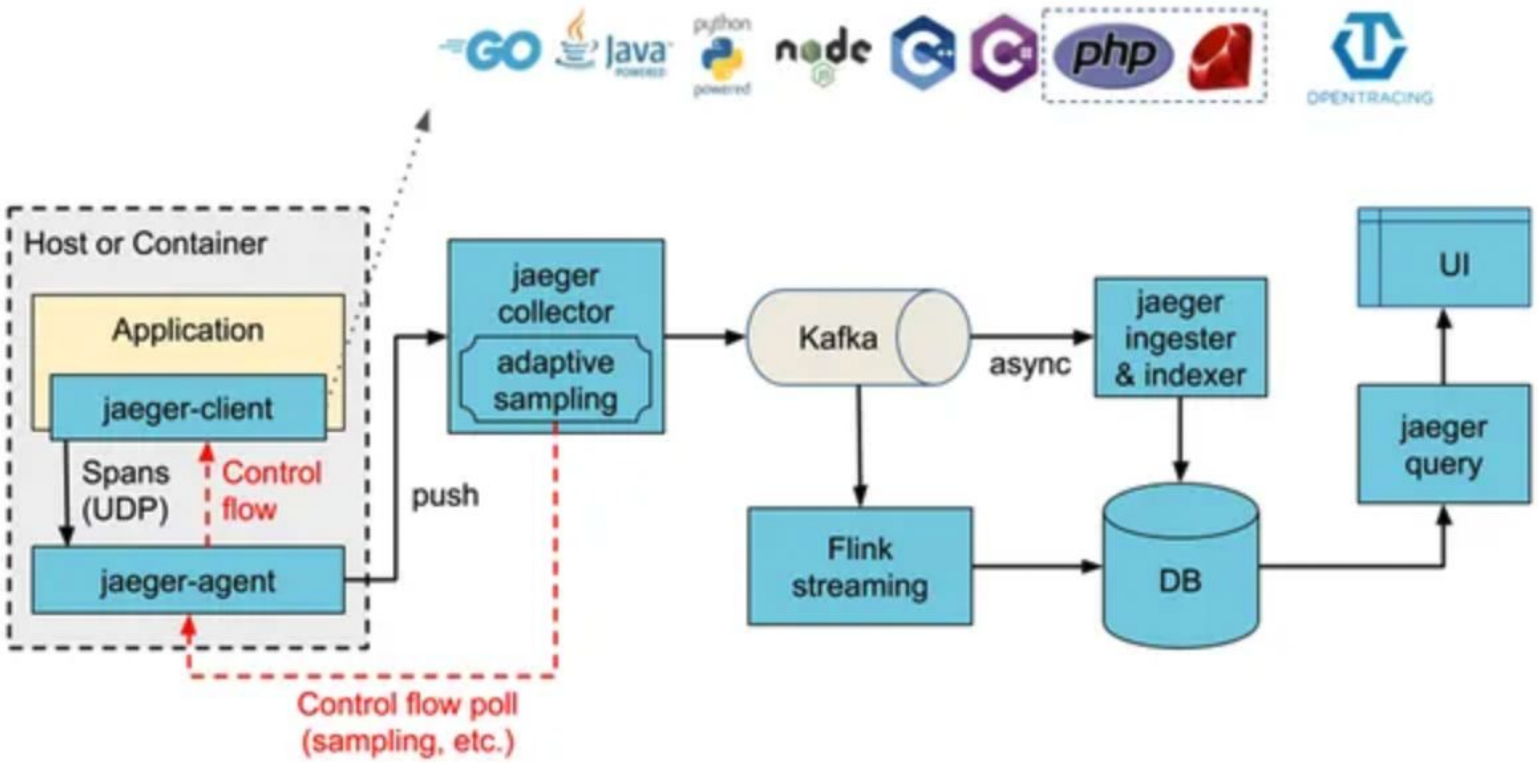
Colors in the graph represent the health of your service mesh. A node colored red or orange might need attention. The color of an edge between components represents the health of the requests between those components. The node shape indicates the type of component such as services, workloads, or apps.




kiali
anonymous
Overview
Graph
Applications
Workloads
Services
Istio Config
Distributed Tracing
Iter8 Experiments
Services > Namespace: travel-agency > travels
Overview
Traffic
Inbound Metrics
Traces
Last 10m
Every 1m
Metrics Settings
Reported from
Destination
Request volume
Request duration
Request throughput
Request size
Response throughput
Response size


kiali
anonymous
Overview
Graph
Applications
Workloads
Services
Istio Config
Applications > Namespace: default > ai
Overview
Traffic
Inbound Metrics
Outbound Metrics
JVM Metrics
JVM Pool Metrics
Vert.x Client Metrics
Vert.x Server Metrics
Last 10m
Every 10s
Metrics Settings
Pods aggregation
Sum
Memory used
Memory commited
Memory max
heap,locals
heap,visitors
nonheap,locals
nonheap,visitors
Span duration
Total live threads
Daemon threads
Threads states

Architecture of Jaeger:



JAEGER

Jaeger is open source software for tracing transactions between distributed services. It's used for monitoring and troubleshooting complex microservices environments.

Problems that Jaeger addresses

distributed transaction monitoring

performance and latency optimization

root cause analysis

service dependency analysis

distributed context propagation