# GRAFANA
# TEMPO

# CONTENT

# INTRODUCTION TO TRACING

Distributed tracing used to monitor and track the interactions between components in a distributed system, particularly in a microservices architecture. It captures the flow of a single request as it traverses through different services, APIs, and databases, providing a detailed view of how the request is handled

Each request is assigned a unique identifier, which is passed along with the request as it moves through different parts of the system. This identifier is used to stitch together the different parts of the trace, allowing you to see the entire lifecycle of a request.

# IMPORTANCE IN MICROSERVICES ARCHITECTURE

## Visibility

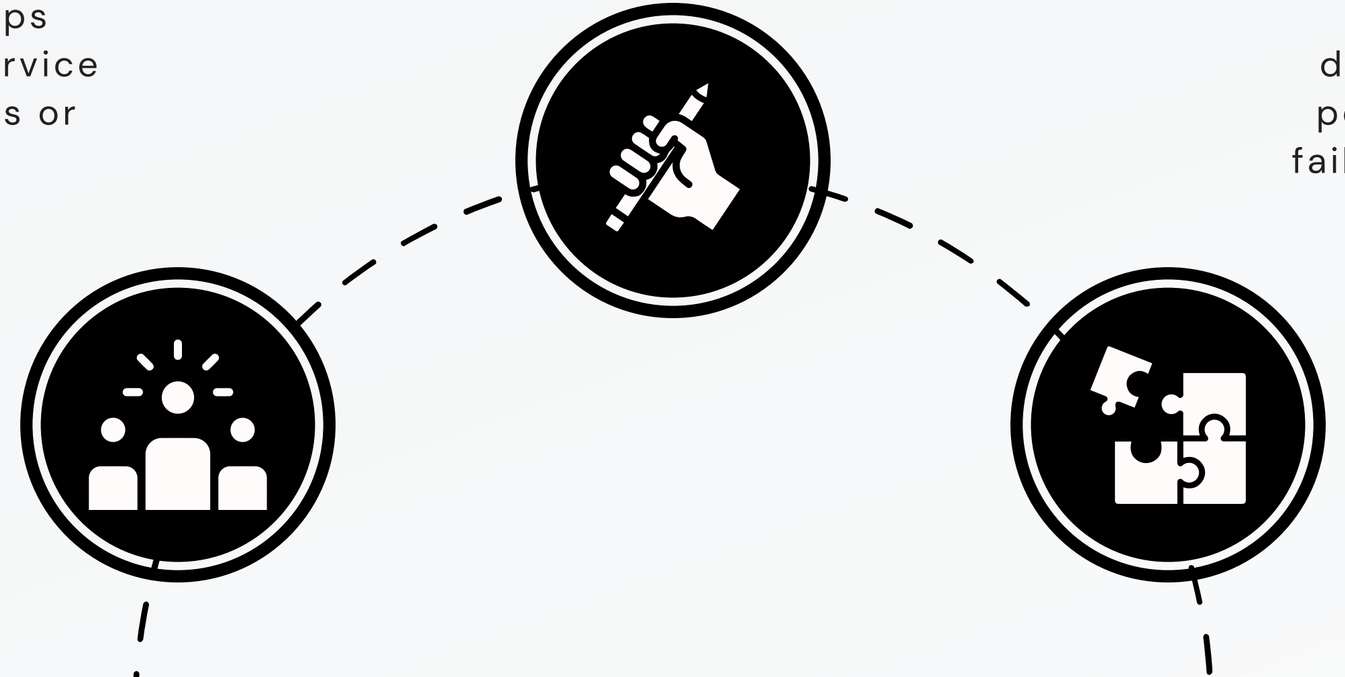
Distributed tracing provides visibility into the performance and behavior of these services. It helps identify which service is causing delays or failures.

## Performance Monitoring

Distributed tracing enables performance monitoring at a granular level, helping teams optimize their services and reduce latency.

## Dependency Mapping

By showing the relationships between services, distributed tracing helps in understanding dependencies and potential points of failure in the system.



# GRAFANA TEMPO

- High-scale, distributed tracing backend that is designed to ingest, store, and query traces from distributed systems. It is optimized for cost efficiency and scalability.
- Tempo only stores trace data in an object storage and uses a simple index to keep track of trace IDs. This approach reduces the need for a complex indexing system and significantly lowers operational costs.

- Tempo works seamlessly with other tools in the Grafana ecosystem, such as Loki for logs and Prometheus for metrics, to provide a full observability stack.
- Tempo supports open tracing protocols like OpenTelemetry, Jaeger, and Zipkin, making it easy to integrate with existing tracing setups and tools.

# FEATURES

**Scalability**

- Suitable for large-scale systems with numerous traces.

**Cost Efficiency**

- Storing only the trace IDs in an index, with full trace data stored in object storage.

**Backend Compatibility**

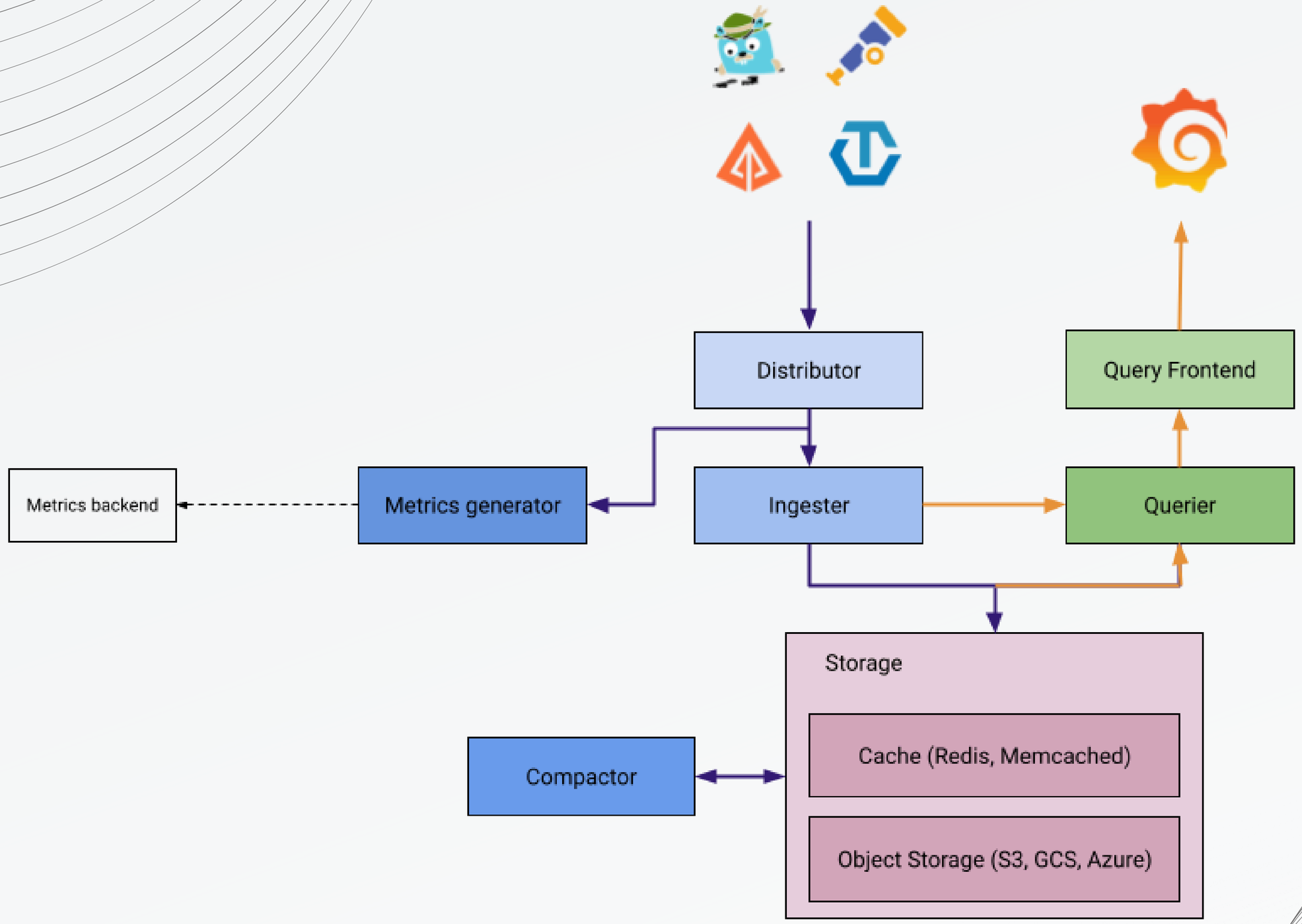
- Compatibility with multiple object storage systems (e.g., AWS S3, GCS).

**Integration with Other Tools**

- It integrates with other Grafana tools (Grafana, Loki, Prometheus).

**No Dependency on Elasticsearch**

- Unlike some other tracing tools, Tempo doesn't require Elasticsearch.


Distributor
Query Frontend
Metrics backend
Metrics generator
Ingester
Querier
Storage
Compactor
Cache (Redis, Memcached)
Object Storage (S3, GCS, Azure)

# DEPLOYMENT

- Deploying Grafana Tempo to Kubernetes or containers offers a scalable and flexible solution for managing distributed traces in microservices architectures. In Kubernetes, Tempo can be deployed using Helm charts, allowing for easy customization and orchestration. The deployment benefits from Kubernetes features like automatic scaling, high availability, and seamless updates. By running Tempo in containers, you also gain the advantages of containerization, such as portability and consistency across environments. This approach simplifies the integration with other services in the Grafana ecosystem, making it an ideal choice for cloud-native environments.

# WORKING WITH TRACES IN GRAFANA

- Grafana integrates seamlessly with Tempo to allow you to explore traces directly from its UI
- Grafana provides a detailed view that includes a trace timeline, span details, and metadata
- Use fields like trace IDs, service names, or time ranges to filter and locate specific traces
- Filter by trace duration, specific operations, or tags associated with spans
- Use aggregations and grouping to analyze trace data more effectively

# ADVANTAGES ON THE COMPETITORS

*No Dependency on Elasticsearch*

*Scalability*

*Lower Resource Consumption and Cost Efficiency*

*Integration with Grafana*

Grafana Tempo may lack advanced features found in Jaeger or Zipkin and has a less mature ecosystem with fewer integrations. Teams used to other tracing tools might face a learning curve adapting to Tempo's unique features.