No More Flying Blind - Unlock Observability for Smooth Sailing with GoFr

Srijan Rastogi | github.com/srijan-27

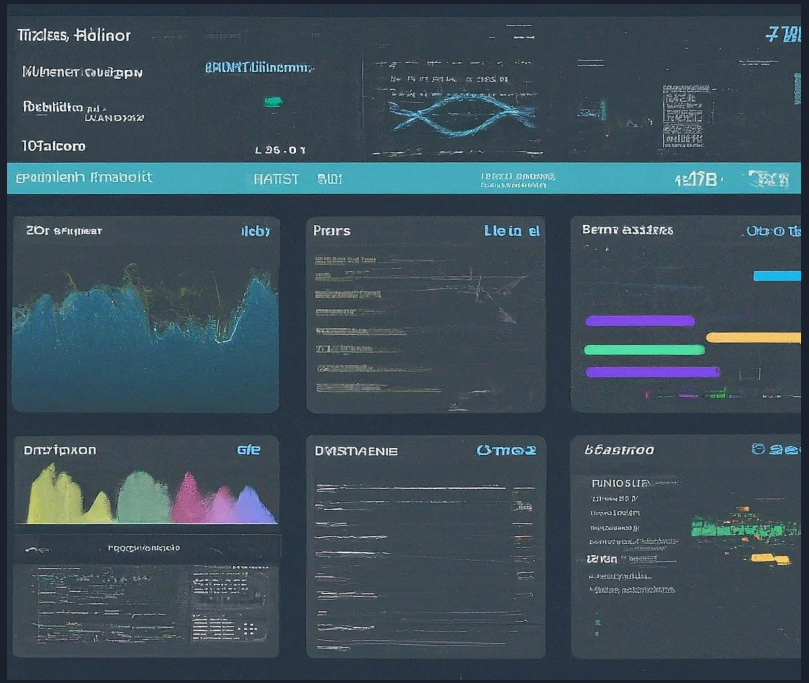
TOC

Overview

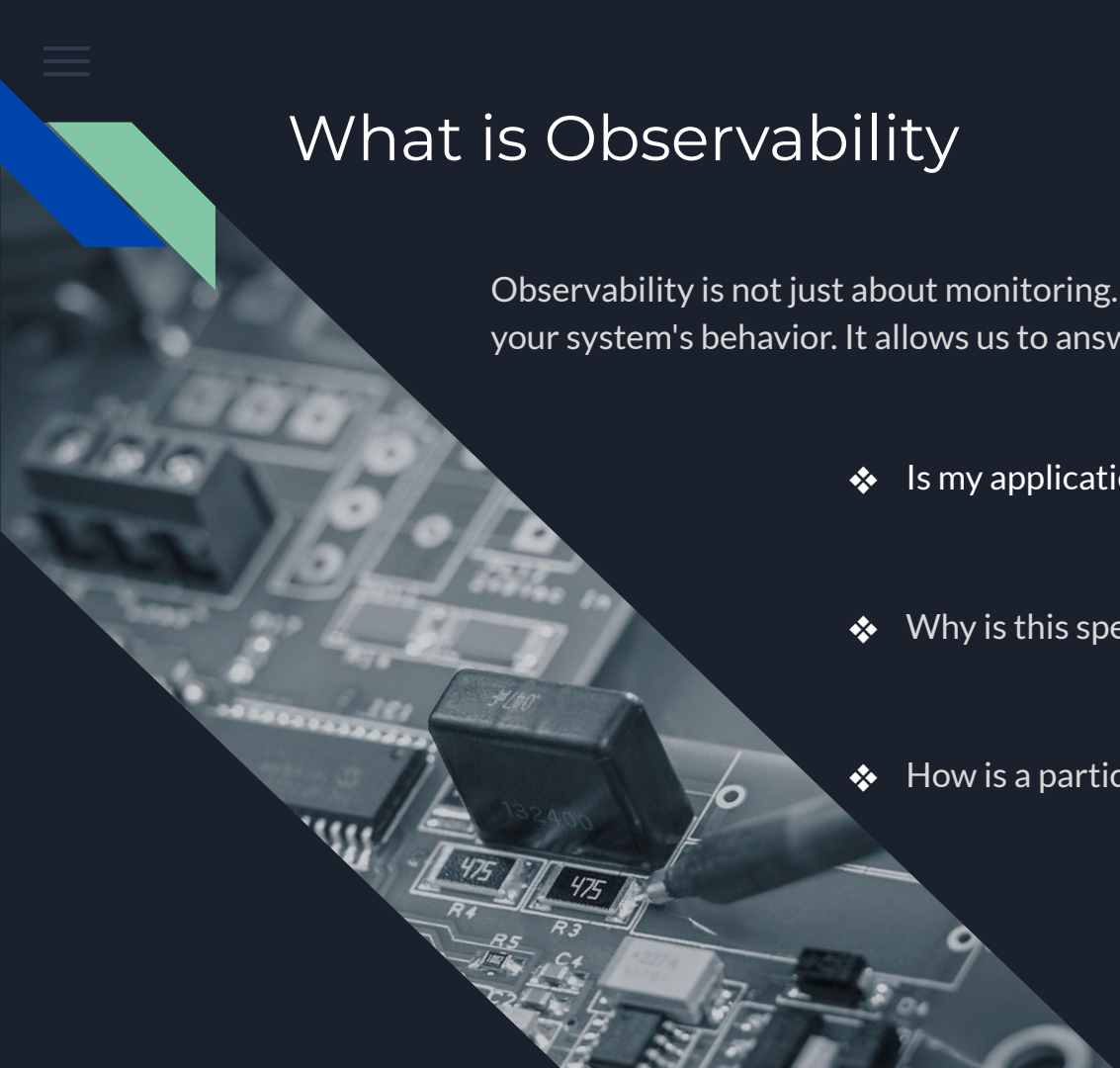
In today's complex software landscape, ensuring optimal application performance and user experience is crucial.

Here's where observability steps in as a game-changer. This presentation will explore the significant benefits observability.

We'll delve into key concepts like logging and tracing, showcasing how they empower developers to build robust and scalable applications with GoFr.



What is Observability



Observability is not just about monitoring. It's about gaining deep insights into your system's behavior. It allows us to answer questions like:

- Is my application performing as expected?
- Why is this specific user experiencing an error?
- How is a particular code change impacting the overall system?

The Building Blocks of Observability

Metrics: Quantitative data points that measure system performance, such as request latency, memory usage, and CPU utilization.

Logs: Textual records of events that occur within the system, providing context for understanding system behavior.

Traces: Distributed tracing tracks the flow of a request across different components in a microservice architecture.

Benefits of Observability

- Faster troubleshooting and problem resolution
- Improved application performance and stability
- Proactive identification and mitigation of potential issues
- Enhanced development and deployment processes

Getting Started with GoFr

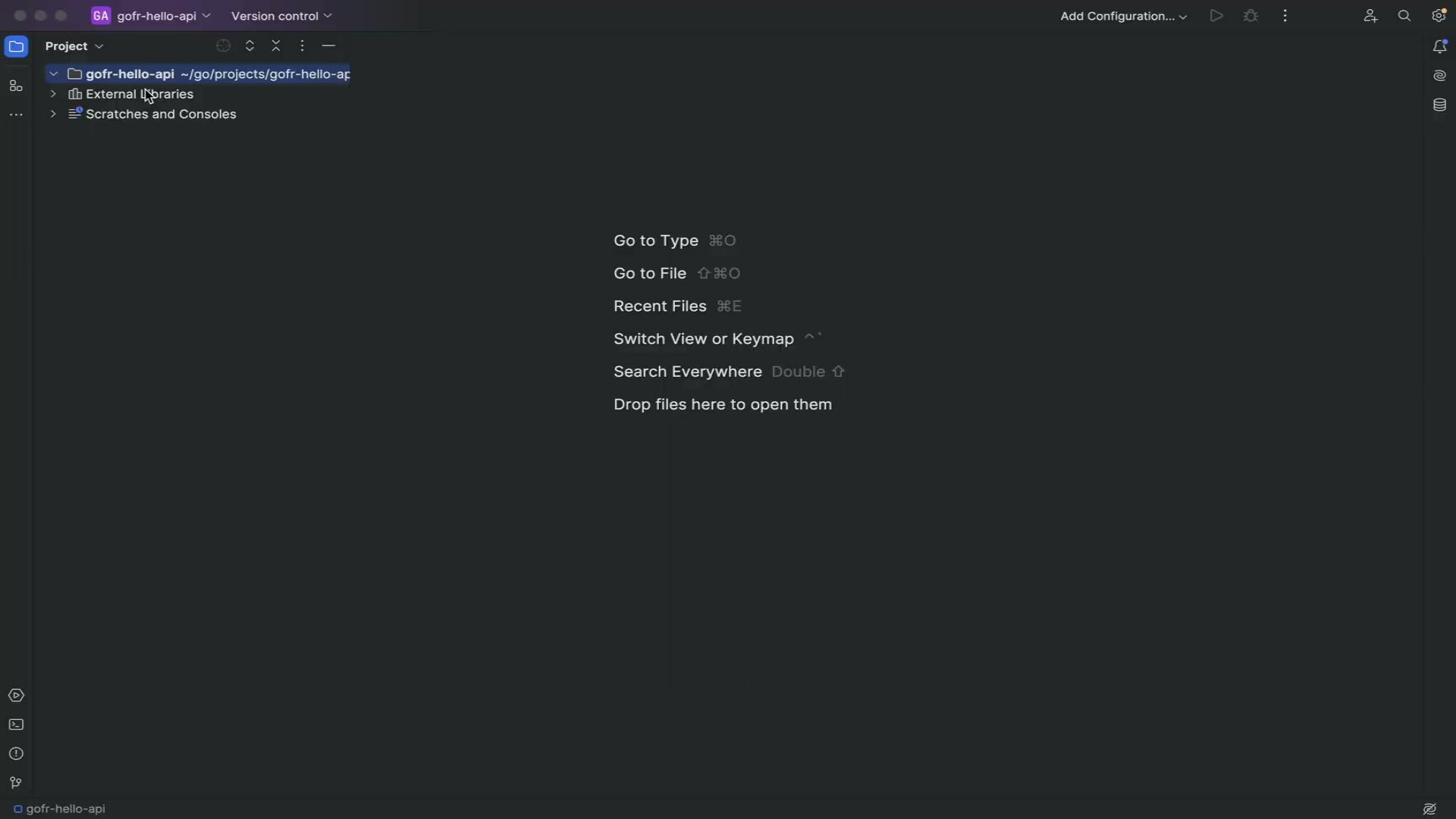
GoFr is an open-source platform with a large and active community.

Numerous resources are available to help you get started, including documentation, tutorials, and forums.

The GoFr community can provide valuable support and assistance.

Website: https://gofr.dev

GitHub: https://github.com/gofr-dev/gofr


GA
gofr-hello-api
Version control
Add Configuration...
Project
gofr-hello-api ~/go/projects/gofr-hello-ap
External Libraries
Scratches and Consoles
Go to Type ⌘O
Go to File ⇧⌘O
Recent Files ⌘E
Switch View or Keymap ^`
Search Everywhere Double ⇧
Drop files here to open them
gofr-hello-api

GoFr: Your Observability Powerhouse

GoFr offers features that are relevant to observability:

1. **Metrics collection**: GoFr can collect various metrics about your application's performance, such as response times and error rates.

2. **Logging**: GoFr provides built-in logging functionality, allowing you to record important events in your application's execution.

3. **Built-in tracing**: GoFr can trace requests as they travel through your microservices.

What are Metrics

Quantitative data points that measure the performance and health of your system.

Examples: Request latency, CPU utilization, memory usage, error rate, response times, throughput.

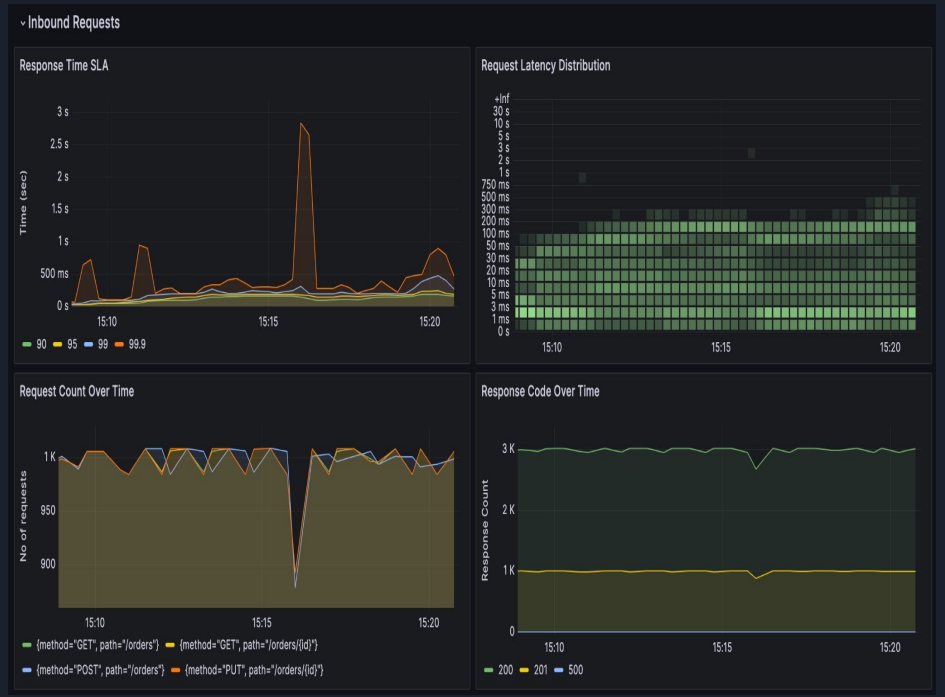
Benefits of Metrics:

1. Provide a high-level overview of system health.
2. Help identify trends and patterns in system behavior.
3. Allow for quick identification of potential issues.
4. Enable proactive monitoring and performance optimization.

GoFr collects metrics from various sources (applications, infrastructure, external tools).

GoFr also offers creation of custom metrics for specific use-cases.

Metrics are exported as per prometheus format.



Logging

Logs are textual records of events that occur within your application.

They provide valuable insights into application behavior, including:

1. Startup and shutdown events
2. User interactions and requests
3. Errors and exceptions
4. Application state changes

Logs offer real-time information, providing valuable insights and immediate visibility into the ongoing state and activities of the system.

It helps in identifying errors, debugging and troubleshooting, monitor performance, analyzing application usage, communications etc.

Benefits of GoFr logs:

1. Provide insights of application
2. Structured logging
3. Customizable log level
4. Support for remote configuration

```
INFO [13:21:46] Loaded config from file: ./configs/.env
INFO [13:21:46] connected to 'orders' database at 'localhost:2006'
INFO [13:21:46] Migration 1712568232 ran successfully
INFO [13:21:46] Starting server on port: 8081
INFO [13:21:46] Starting metrics server on port: 2121
INFO [13:21:54] 1e17f25a30219122637d342b15d193bc 201       23316µs POST /orders
INFO [13:21:59] 0d178f2241465c9e433630bbb864cb6a 201        3811µs POST /orders
INFO [13:22:06] 04ce01f71ae4456e57cfea3889206516 200       10161µs GET /orders/7a4288e6-2310-11ef-b06a-72525f9985f0
INFO [13:22:07] 47554c8ebbc4cb6350cf5e6b83e5a300 200        4166µs PUT /orders/7a4288e6-2310-11ef-b06a-72525f9985f0
INFO [13:22:11] b49c9a424b593fba7f291cc1f6e82a41 204        6835µs DELETE /orders/7a4288e6-2310-11ef-b06a-72525f9985f0
ERRO [13:22:26] 36c721cbec25f415e1753f7e9aacdb23 500        8316µs GET /orders/7a4288e6-2310-11ef-b06a-72525f9985f0
```

Traces

Traces are records of the path a request takes through a distributed system, like GoFr.

They capture information about each step of the request, including:

- The service involved (e.g., authentication service, product service)
- The time it took to process the request at each service
- Any errors that occurred

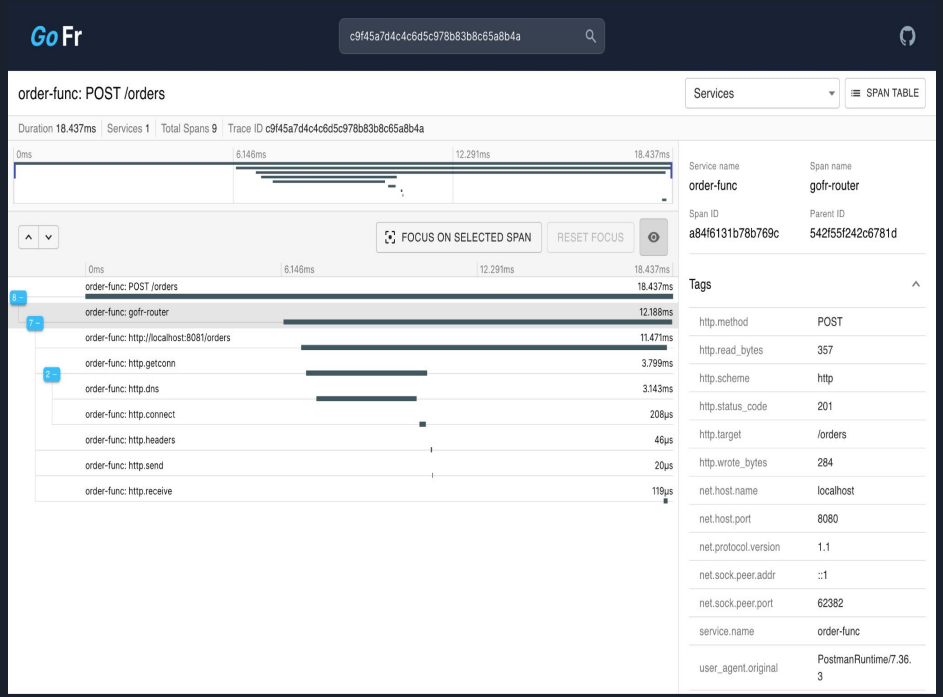
Benefits of Tracing:

1. Debugging
2. Performance Monitoring
3. Service health monitoring
4. Incident management

GoFr can integrate with popular distributed tracing tools like Zipkin, Jaeger.

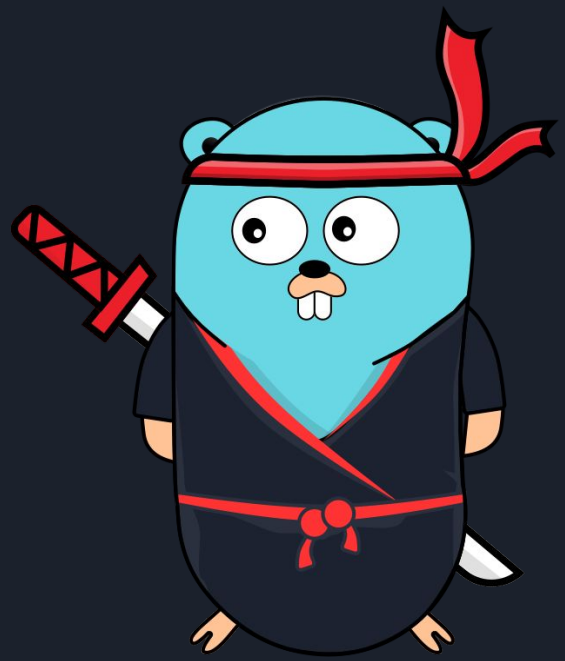
GoFr also has its own visualisation tool for traces.

Each request is assigned a unique trace ID, which is propagated across all services involved.



Benefits of GoFr

1. Simplified observability
2. Faster troubleshooting
3. Improved development efficiency
4. Reduced costs



Thank you!

You can find all the resources used in this presentation here: https://github.com/srijan-27/conf-42-observability