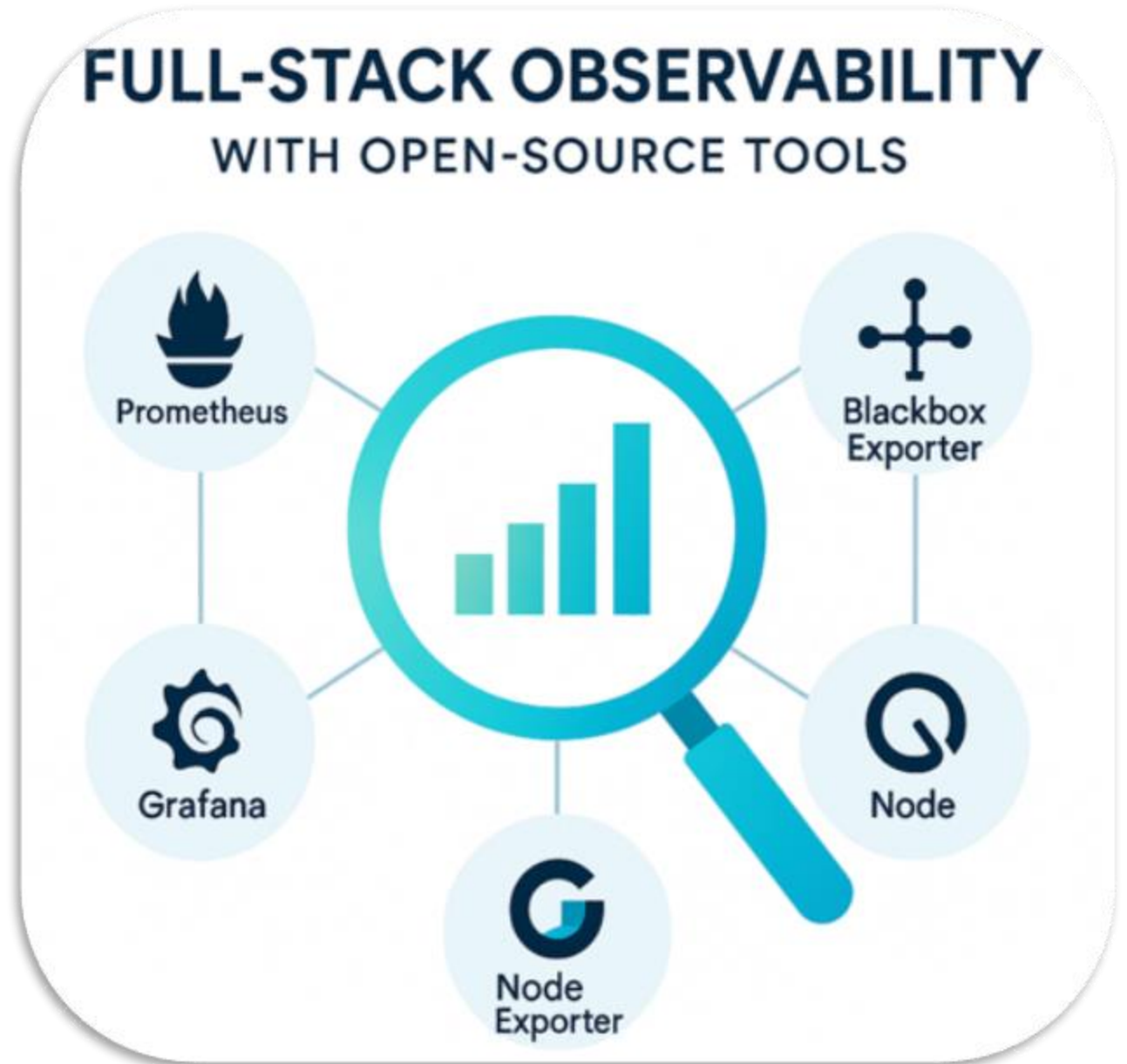


# Full stack Observability with OpenSource Tools



# Agenda

Project Overview & Goals

Architecture Diagram & Integrated  
Tools

One-Click Deployment- GitHub &  
Helm

Key Benefits

Helm Structure

Roadmap & Next Steps

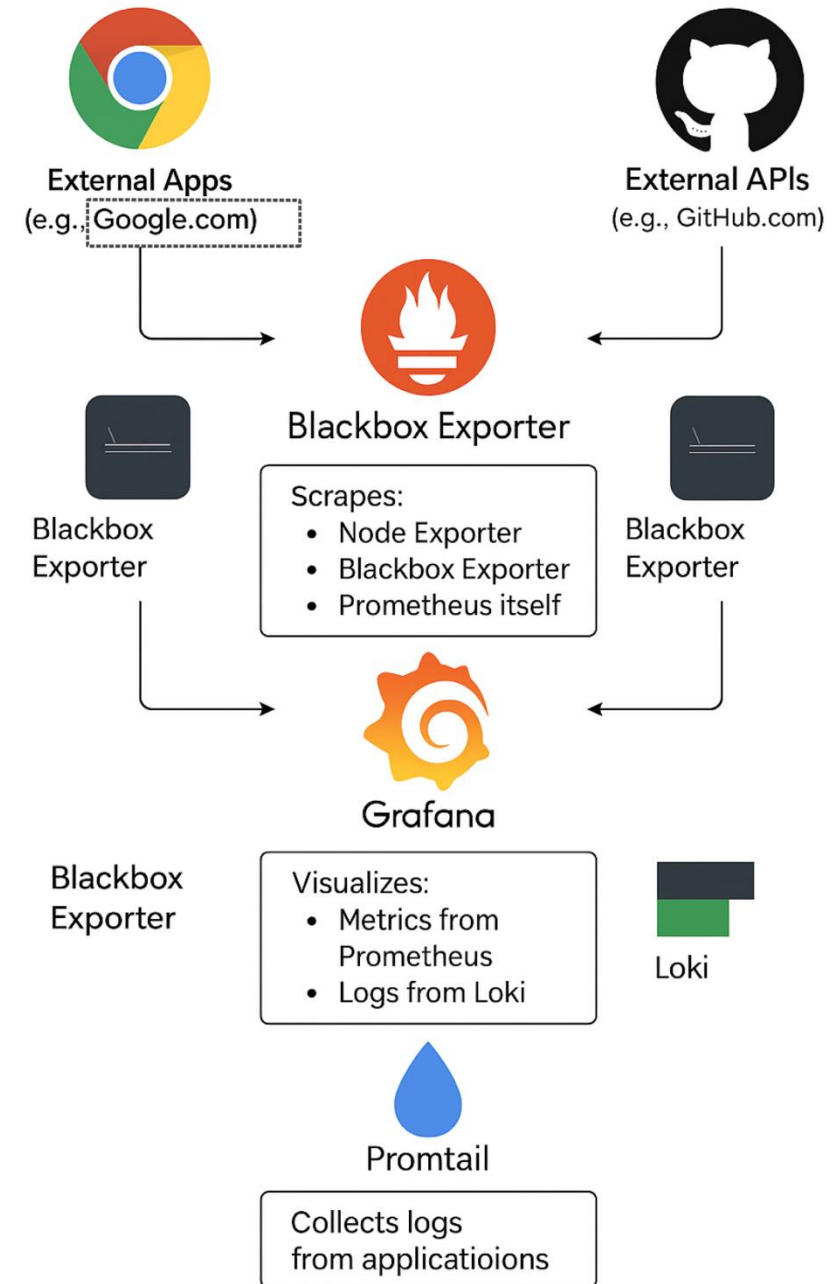


## Project Overview & Goals

Our primary objective with this initiative is to develop an integrated, enterprise-grade observability platform. By consolidating industry-leading open-source technologies, we are creating a solution that simplifies monitoring and provides immediate, actionable insights across our entire technology landscape, from infrastructure to application performance.



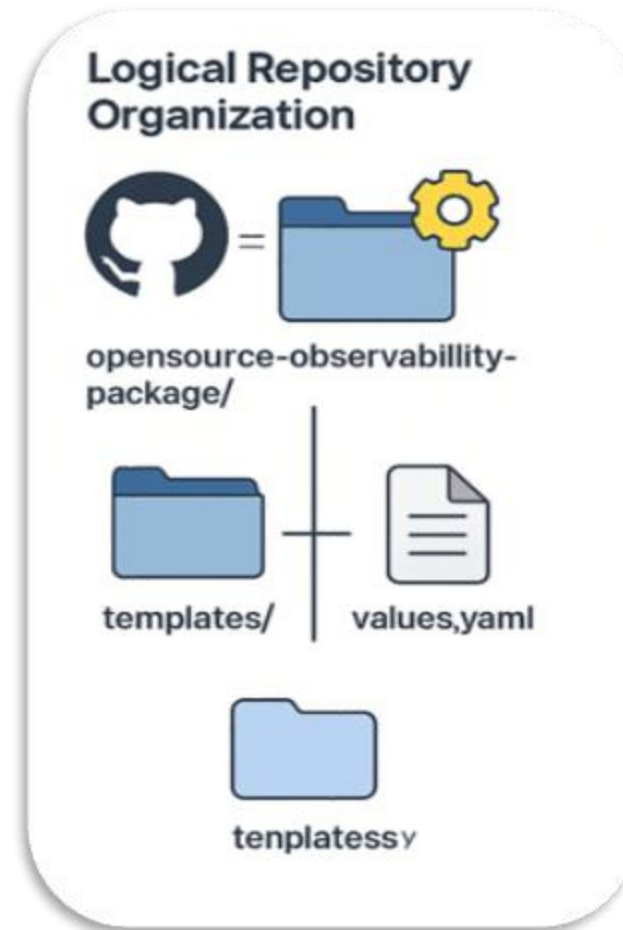
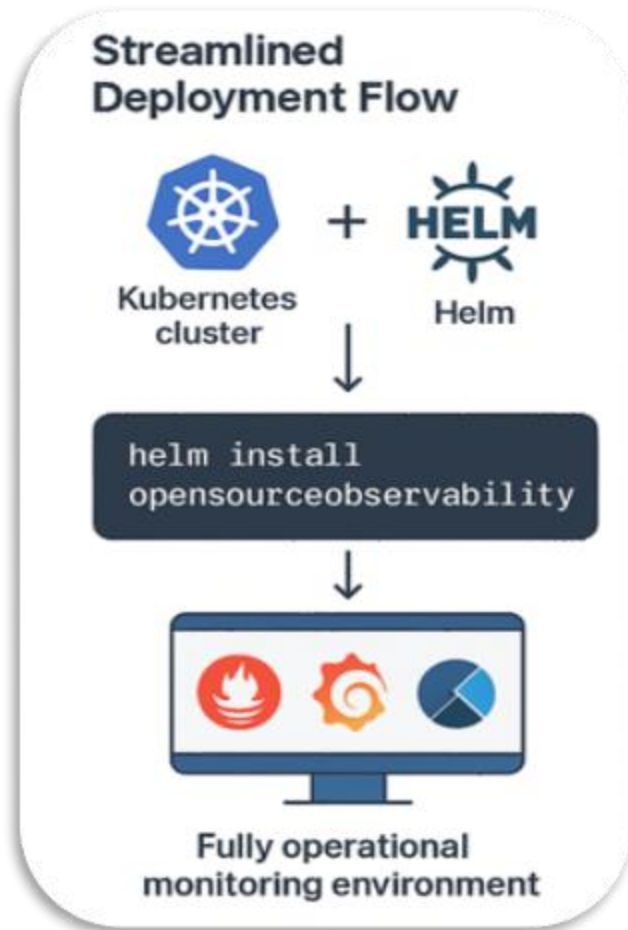
# Architecture Diagram & Integrated Tools





## One-Click Deployment- GitHub & Helm

We have focused on two critical areas: a highly simplified deployment process and a clean, logical code structure. This approach minimizes setup time and makes the project transparent and easy for our teams to contribute to.





# Key Benefits

This platform would deliver a comprehensive suite of monitoring capabilities, providing deep visibility across the entire technology stack which will have

## Metrics Monitoring

## Network Visibility

## Log Management

## Visualization & Alerting

### Metrics Monitoring:



CPU, memory, disk usage

- Custom app metrics

### Visualization:

- Pre-built & custom dashboards
- Alerting integration
- Historical analysis

### Network Probing:

- HTTP/S checks
- ICMP ping
- DNS lookups

### Log Management:

- Real-time log ingestion
- Centralized logging
- Filtering & streaming





# Helm Structure

This Helm chart bundles all observability components into a single, easy-to-deploy package. Here is the folder structure and a summary of its key files

helm-kube-observability-stack/	
— charts/	# (Optional) Subcharts if needed
— templates/	# Contains all Kubernetes manifest templates
— grafana-deployment.yaml	# Deploys Grafana Pod
— grafana-service.yaml	# Exposes Grafana service
— loki-deployment.yaml	# Loki for log aggregation
— loki-service.yaml	
— node-exporter-daemonset.yaml	# Node-level metrics collection
— node-exporter-service.yaml	
— prometheus-deployment.yaml	# Prometheus core service
— prometheus-service.yaml	
— prometheus-config.yaml	# Prometheus scrape config
— promtail-deployment.yaml	# Promtail for log shipping
— promtail-service.yaml	
— promtail-config.yaml	
— blackbox-exporter-deployment.yaml	# For probing HTTP, ICMP, DNS, etc.
— blackbox-exporter-service.yaml	
— blackbox-exporter-config.yaml	
— namespace.yaml	# Creates a dedicated namespace
— ingress.yaml	# Ingress for UI access
— NOTES.txt	# Helm notes shown after install
— values.yaml	# User-configurable settings for all components
— Chart.yaml	# Chart metadata (name, version, etc.)





## Roadmap & Next Steps

Open-source tools are generally good and budget-friendly way to start. But each tool by itself has some limitations. To make this a proper solution, we must plan for these limitations.

So, in the next phase of the project, our plan is to add some other tools. This will help us cover the more advanced monitoring requirements and make sure we have a complete solution.

### Immediate Priorities



**Add Traceroute  
Exporter**



**Improve Grafana  
Dashboards**



**Add Alerting Rules**



**Enable Backup/  
Persistence**

### Future Enhancements



**Multi-Cluster Support**



**Security Hardening  
(OAuth, TLS)**



**Optimized Resource  
Usage**



**More Exporters  
(MySQL, Redis, e.etc.)**



# Demo





# Questions & Inputs



The bridge to possible

Thank you