Project #17 Open Telemetry Observability Dashboard, Team #1
Date: September 6, 2025
(Updated as of Novemeber 23, 2025 – highlights indicate new updates)

Project Overview

The goal of this project is to create and put into use a web-based Observability Dashboard that employs Open Telemetry (OTel). The dashboard will display metrics, traces, and logs from various apps, allowing you to view the system's performance in real-time. The dashboard remains vendor-neutral because it utilizes the Open Telemetry standard. This will make it work with a wide range of technologies and platforms.

Why It Is Important

Modern distribution systems are complex and often involve multiple services, making it challenging to identify and resolve problems quickly. A single outage or performance issue can result in significant downtime, lost sales, and unhappy customers. Teams can do the following by putting all their observability data onto one dashboard:

- Find problems fast with live monitoring and warnings.
- You can find the fundamental causes of problems faster by linking logs, traces, and measurements.
- Improve performance by looking at patterns and cutting down on the waste of resources.
- Make services more reliable, which is very important in today's competitive world.

Value for the Project Sponsor

The sponsor's value lies in offering a cost-effective, lightweight alternative to pricey enterprise monitoring tools. To be clear:

- Lower Costs: You don't have to rely on expensive vendor-locked tools anymore, but you still get good observability.
- Scalability: Good for both small and expanding teams that require their systems to be able to grow with them.
- Actionable Insights: Gives you real-time visibility that helps you make better decisions and run your business more efficiently.
- Competitive Edge: Faster issue detection and resolution equal less downtime and a better experience for users.

Project Participants (the Major responsibilities listed below are for examples only)

Roles	Name &	Major responsibilities	Contact (email, cell, etc.)

	Course Section		
Project owner sponsor	Taylor Cuffie	Define project goals, provide requirements, review deliverables.	Tcuffie@kennesaw.edu
Team leader	Kian Rezaeinejad - W02	Facilitate meetings, maintain MS Teams, GitHub repo management, track milestones, liaison with sponsor/instructor.	Phone: (678)780-6865
Hrezaein@students.kennesaw.edu			
Team members	Daniel Ekema - W01	Backend/Frontend Code Implementation, API integration and data visualization.	Phone: 404-200-0349
dekema@students.kennesaw.edu			
	Jaytin Bolling - W02	Ui/ Ux designer and framework	Phone: 470-493-4249
Jbollin5@students.kennesaw.edu			
	Dakobie Garcia – W02	Planning, producing programming deliverables, contribute expertise, testing and documentation processes.	Phone: 478-538-1717
dgarci64@students.kennesaw.edu			
Advisor / Instructor	Donald Privitera	Facilitate project progress; advise on project planning and management.	dprivit2@kennesaw.edu

Project Scope

In Scope (What we will do):

- Design and implement a web-based observability dashboard.
- Use telemetry data from open telemetry compliant endpoints
- Implement a lightweight data storage
- Provide a dashboard with real time visualization such as graphs, charts, and threshold breaches.
- Deliver documentation, recommendations and deployment guide.
- Set up an application which exposes open telemetry data with live traffic.
- Set up a linux server to host our application.

Out of Scope (What we will not do):

- Develop a mobile only based dashboard.
- Skip the framework design process.
- Build or maintain an application unrelated to observability such as unrelated backend logic.

Project Approach

- The project will follow a biweekly check-in approach.
- Biweekly is every week on Saturday or Sunday. Informal check-ins may occur midweek if needed to catch up on assignments.
- We plan to begin with a clear framework design. With a visualized dashboard, the team will simulate telemetry data before connecting to live endpoints.
- To further continue with a successful project approach, we will create a build vs integrate approach.
- Build the front end, alongside a lightweight backend api. Later on, we would integrate the open telemetry endpoints. Lastly, Testing and Q&A by Validating the system under different workloads using high frequency telemetry streams.

Initial sponsor and stake meetings were held:

- Sponsor emphasized vendor neutrality and containerized deployment. Gave each member a Linux account.
- Stake holders emphasized real time data visualization.
- **The framework and architecture design have only been discussed, not finalized. No documentation or diagrams have been created yet.** To sum up, our approach will consist of:
  - An Iterative approach
  - Architecture and framework design
  - Proof of concept

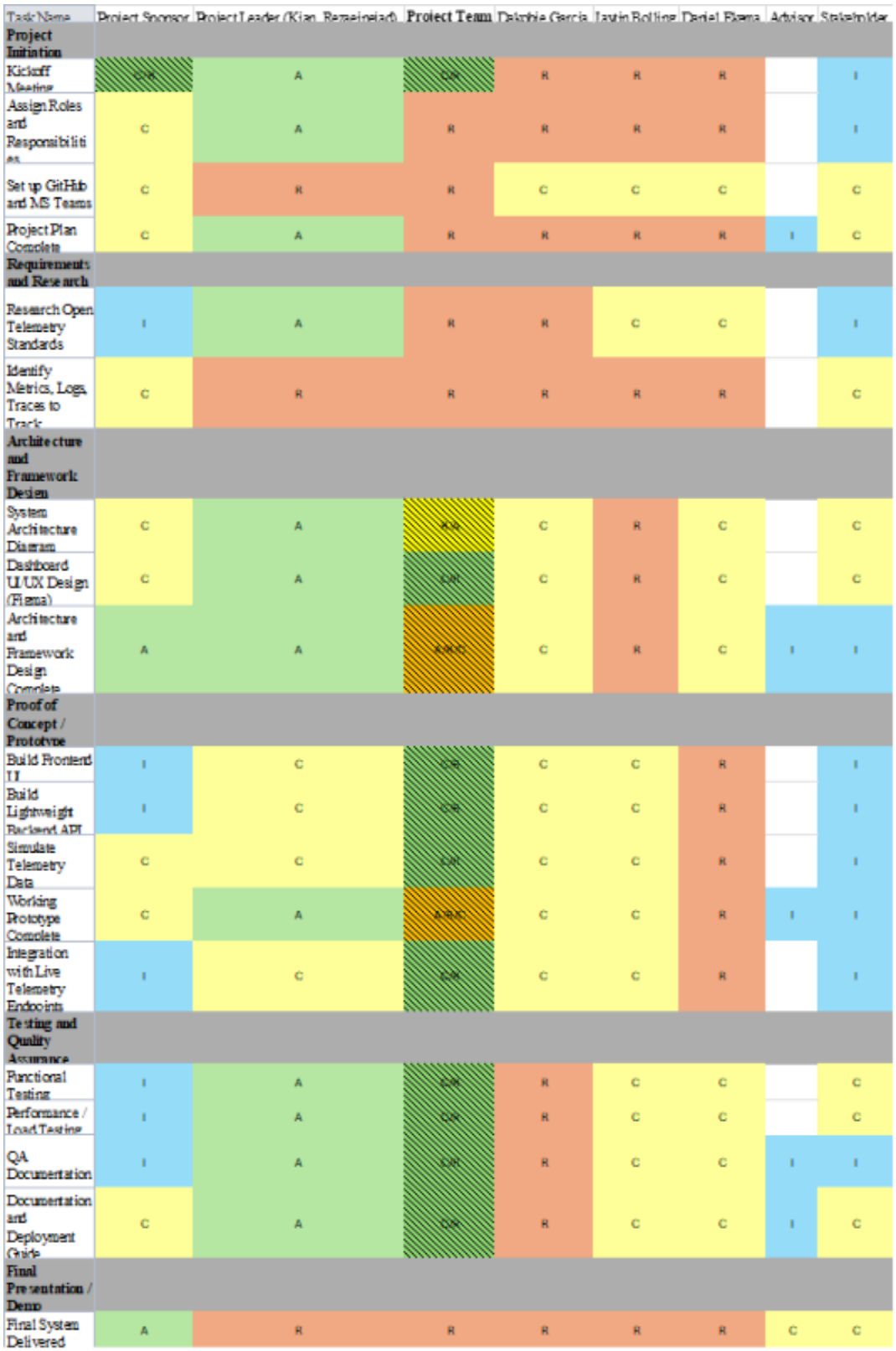
- Build & integrate
- Testing
- Deployment

RACI Chart

A RACI chart is a project management tool that defines roles and responsibilities for tasks or deliverables in a project.

- R = Responsible: The person(s) doing the work.
- A = Accountable: The person ultimately answerable; approves the work.
- C = Consulted: People who give input, expertise, or feedback.
- I = Informed: People who need updates, but don't actively contribute.

T17 RACI Chart



Task Name	Project Sponsor	Project Leader (Kian, Remeinelad)	Project Team	Daishie Garcia	Javin Rolling	Daniel Flores	Advisor	Stakeholder
Project Initiation								
Kickoff Meeting	C/R	A	C/R	R	R	R		I
Assign Roles and Responsibilities	C	A	R	R	R	R		I
Set up GitHub and MS Teams	C	R	R	C	C	C		C
Project Plan Complete	C	A	R	R	R	R	I	C
Requirements and Research								
Research Open Telemetry Standards	I	A	R	R	C	C		I
Identify Metrics, Logs, Traces to Track	C	R	R	R	R	R		C
Architecture and Framework Design								
System Architecture Diagram	C	A	A/R	C	R	C		C
Dashboard UI/UX Design (Figma)	C	A	C/R	C	R	C		C
Architecture and Framework Design Complete	A	A	A/R/C	C	R	C	I	I
Proof of Concept / Prototype								
Build Frontend UI	I	C	C/R	C	C	R		I
Build Lightweight Backend API	I	C	C/R	C	C	R		I
Simulate Telemetry Data	C	C	C/R	C	C	R		I
Working Prototype Complete	C	A	A/R/C	C	C	R	I	I
Integration with Live Telemetry Endpoints	I	C	C/R	C	C	R		I
Testing and Quality Assurance								
Functional Testing	I	A	C/R	R	C	C		C
Performance / Load Testing	I	A	C/R	R	C	C		C
QA Documentation	I	A	C/R	R	C	C	I	I
Documentation and Deployment Guide	C	A	C/R	R	C	C	I	C
Final Presentation / Demo								
Final System Delivered	A	R	R	R	R	R	C	C

Final Deliverables

We plan to deliver:

- System architecture. Figma design or framework with diagram
- Working web-based dashboard. Frontend and backend
- Telemetry data pipeline
- Alerting modules
- Evaluation Report
- Linux server
- User guide and deployment documentation, such as a planned-out GitHub ReadMe.md file
- Final presentation or demo to our sponsor and faculty

Milestones (Indicate what will be delivered and when)

Milestone	Date	Status
Project plan and team discussion	9/7/2025	Completed
Framework and Architecture design	**10/24/2025**	**Completed**
Sponsor and Stakeholder Meetings	**9/12, 9/26, 10/24**	**Held – Sponsor and stakeholder input recorded**
Working Prototype with simulated telemetry + frontend demo	11/14/2025	Completed
Final systems with live endpoints, testing, and documentation	12/3/2025	Completed

Deliverable Expectations

List the required and optional deliverables your sponsor expects to see.

Required:

- Functional web-based dashboard
- Documentation and deployment guide
- Presentation/demo

Optional:

- Containerized deployment such as docker

Future meetings dates/times

Weekly Team meetings: Every Saturday, 6-7pm.

Sponsor Meetings: Every Friday, 3-4pm

Collaboration Plan

How will the team collaborate?

- Use MS project to monitor team progress and document logs
- Use MS teams for communications, file sharing, and meeting notes.

Communication Plan & Policy

- Weekly meetings every Saturday/Sunday, 6-7 pm
- Response time expectation: within 24hours.
- All messages' Teams, emails or text should be within 24 hours.

Urgent and non-urgent
Urgent: Any issue that blocks or slows the team progress. Non-Urgent: Message the group and wait 24 hours for a response. If no response, escalate using the process below.

Escalation process:

- Message team chat (MS Teams) or the responsible team member: 24h response expected
- If no response is received within an hour, text message is sent
- If no response within 30 to 40 minutes a phone call is placed.
- If still no response, the team leader attempts a direct contact.

- If unresolved the same day, the team leader notifies the instructor
- If unresolved, the sponsor is notified.

See the Gantt chart, Resource Overview Report - .MPP file attached.
Gantt chart to be updated after rescheduling framework milestone

Project Change Management

- **Communicate with the Stakeholder on the changes that may arise and plan accordingly on how this could affect the deliverables and time expectations.**
- **Create an effective plan that is transparent and includes the changes and the effects of the current requirements and milestones.**
- **Adhere to the timeliness of the current deliverables and effectively incorporate the change into the plan while also maintaining transparent two-way communication with the stakeholder if questions arise.**
- **If changes could impact the project drastically, then communicate effectively with the stakeholder through a planned communication plan and escalation process if necessary.**

Quality Assurance Plan

QA Planning has not yet started. Testing strategy will begin once code development is underway after architecture finalization (expected October 31).

Our Q&A plan objective:

- Ensure all deliverables meet requirements. This includes the dashboard, API, and deployment guide.
- Minimize risks related to delays.
- Provide some assurance that our system is scalable and user-friendly.

Risk Management Plan

Risk	Description	Likelihood	Impact	Mitigation
1	Team member stops responding/withdraws	Med	High	Redistribute workload and notify instructor early

2	Telemetry endpoints unavailable	Low	Med	Use a simulates telemetry generators for callback
3	Missed deadlines	Med	High	Weekly progress reviews
4	Technical issues such as frameworks not working.	Low	High	Research alternatives or consult instructor early
5	**Framework & architecture delay**	**High**	**High**	**New target date set, team will prioritize and finalize framework by 10/31/2025**

APPENDIX - A

Kickoff Meeting Notes:

- Reviewed project document and sponsor expectations.
- Agreed on a weekly meeting
- Assigned Preliminary roles and responsibility
- Established a GitHub repo and MS Teams channel.
- Discussed milestones and project phases
- **Framework has only been discussed, not started.**
- **First sponsor and stakeholder meetings completed – priorities include lightweight design, Docker compatibility, real-time alerting.**