GridAI

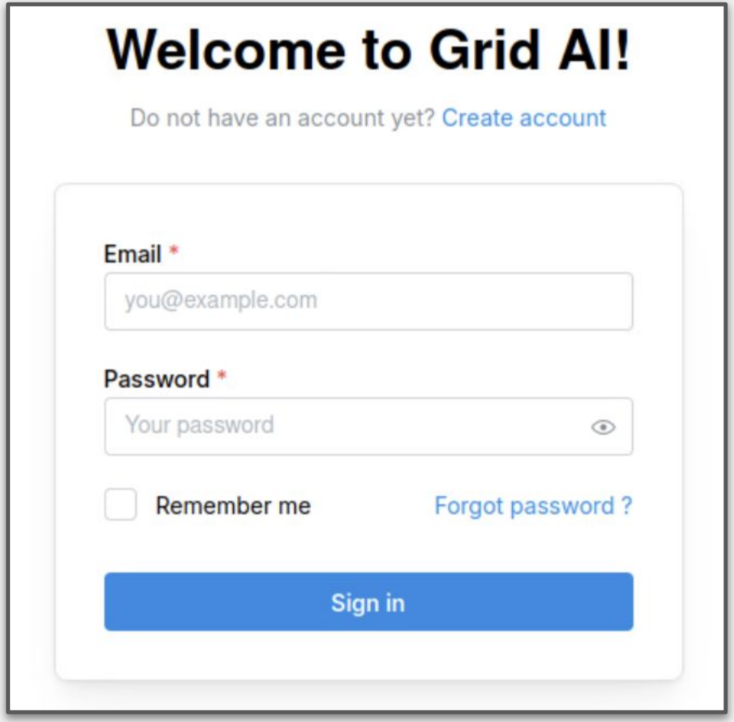
Front-end Team

IRP Presentation

Team ID: SD May 25-43

Client/Adviser: Dr. Gelli Ravikumar
Mentors: Peeyush Gupta, Rolf Anderson

- Skyler Kutsch
- Rangsimun Bargmann,
- Hang Thang
- Justin Soberano B.,
- Franck Biyoghe Bi Ndoutoume
- Jesus Soto Gonzalez



Welcome to Grid AI!

Do not have an account yet? Create account

Email *

you@example.com

Password *

Your password

□ Remember me

Forgot password ?

Project Overview

- ➔ **GridAI** is an advanced web-based platform for electric grid management, designed to simplify data analytics and provide seamless data interaction.

- ➔ It **empowers users** by streamlining decision-making in power grid operations through an intuitive interface and efficient data handling capabilities.

- ➔ **Intended Users:**
  - ◆ Distributed Energy Resources
  - ◆ Distributed and Independent System Operators

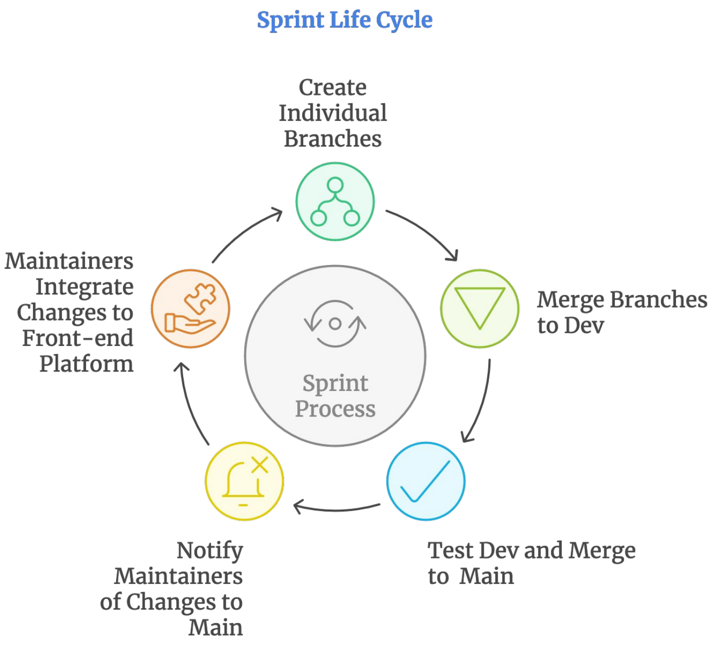
- ➔ With **over four years of continuous development**, GridAI has evolved significantly on the backend, though the frontend initially had limitations that opened the door for improvement.

- ➔ As **Team sdmay25-43**, we focused on improving the originally provided frontend by addressing its key limitations and introducing enhancements that made the interface more intuitive, interactive, and adaptable to user needs.

Project Management

➔ Followed an **Agile-Inspired Project Management**:

- Sprint-based development with weekly team meetings.
- Weekly client and mentor check-ins for feedback and alignment.
- Mid-semester and final showcases to present progress.
- GitLab issue board and milestones for task tracking and sprint planning.

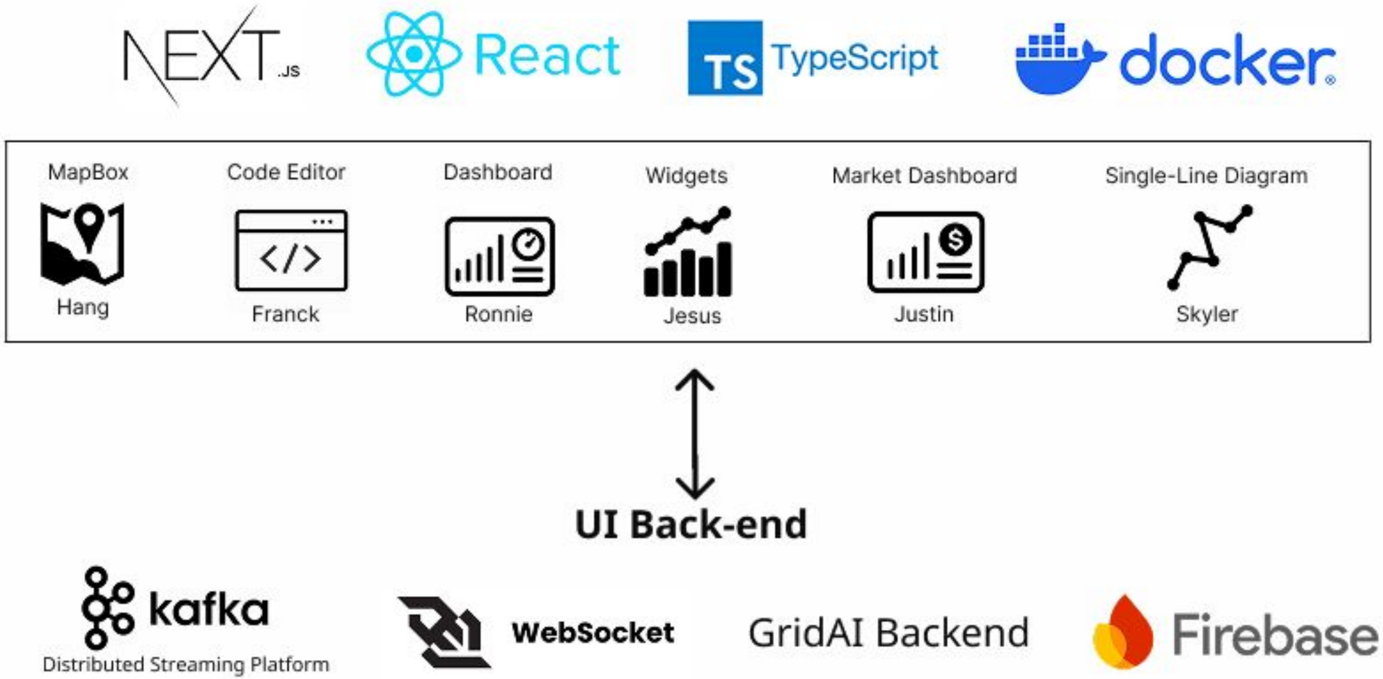

Sprint Life Cycle
Create Individual Branches
Merge Branches to Dev
Test Dev and Merge to Main
Notify Maintainers of Changes to Main
Maintainers Integrate Changes to Front-end Platform
Sprint Process

Sprint Life Cycle Visual

GridAI

Web Platform

Front-end



NEXT.js React TS TypeScript docker

MapBox — Hang

Code Editor — Franck

Dashboard — Ronnie

Widgets — Jesus

Market Dashboard — Justin

Single-Line Diagram — Skyler

UI Back-end

kafka Distributed Streaming Platform

WebSocket

GridAI Backend

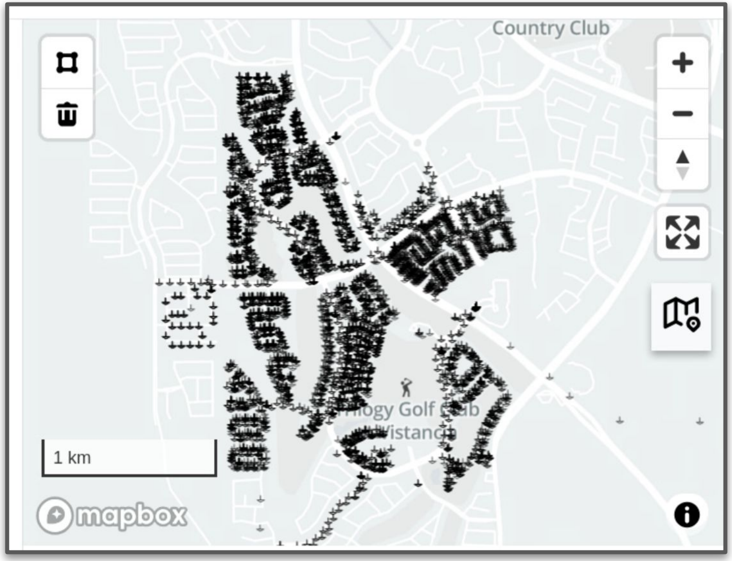
Firebase

MapBox

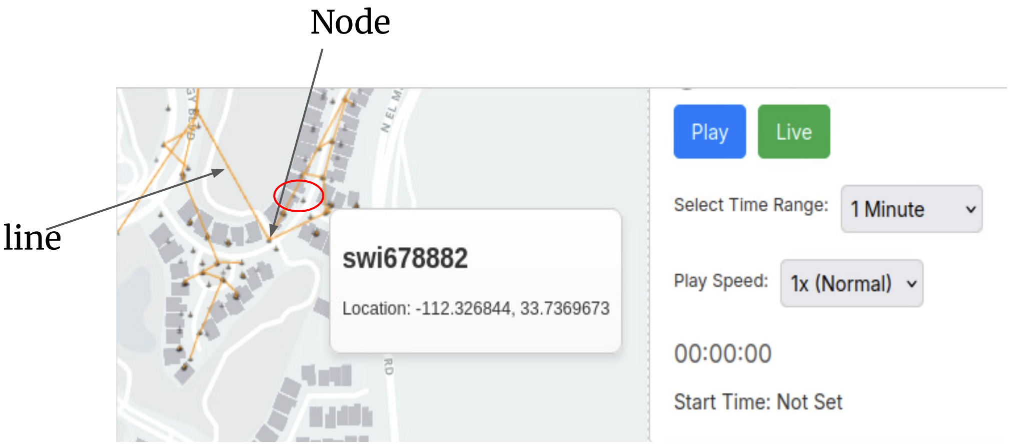
Hang

What is the MapBox Timeline?

- Allows user to replay the record data by location, select an specific event, filter date and time to see what happened during the period of the event all within the GridAI platform.
- Live view allows user to monitor the electric power system of a specific location in the map.



MapBox



swi678882

Location: -112.326844, 33.7369673

Play Live

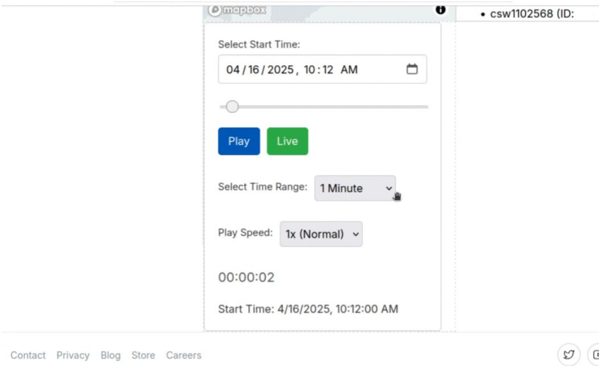
Select Time Range: 1 Minute

Play Speed: 1x (Normal)

00:00:00

Start Time: Not Set

Node and line with ID and location



• csw1102568 (ID:

Select Start Time:

04 / 16 / 2025 , 10 : 12 AM

Play Live

Select Time Range: 1 Minute

Play Speed: 1x (Normal)

00:00:02

Start Time: 4/16/2025, 10:12:00 AM

Selection of date and time

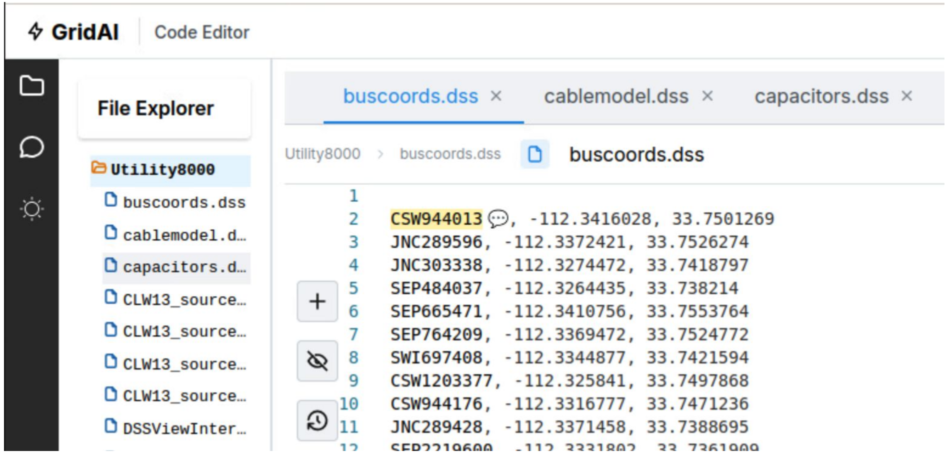
(Timeline)

Code Editor

Franck

What is it?

A built-in **code editor**—like Visual Studio Code—designed for writing, editing, and real-time collaboration on code, all within the **GridAI platform**.



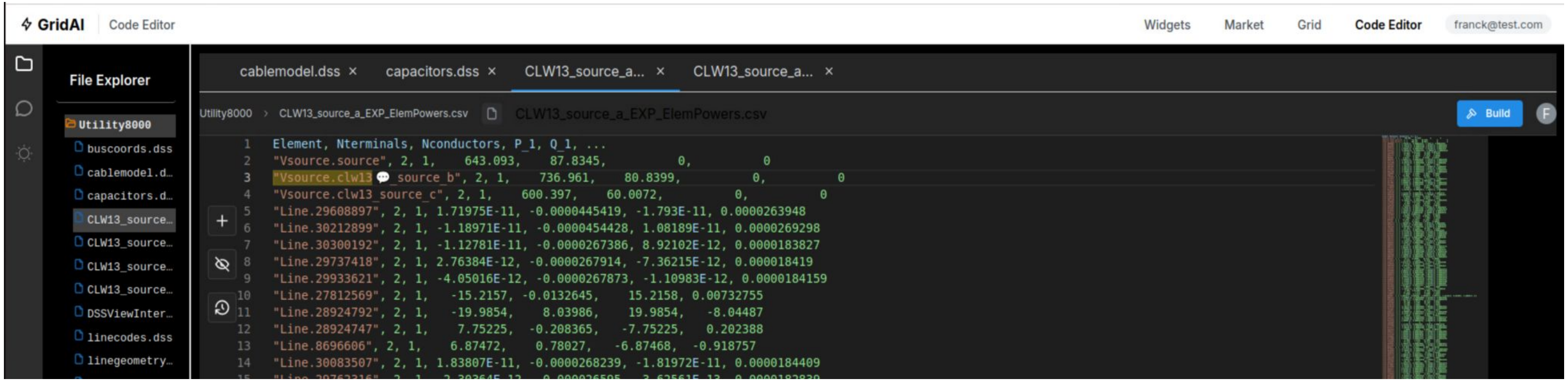
```

CSW944013, -112.3416028, 33.7501269
JNC289596, -112.3372421, 33.7526274
JNC303338, -112.3274472, 33.7418797
SEP484037, -112.3264435, 33.738214
SEP665471, -112.3410756, 33.7553764
SEP764209, -112.3369472, 33.7524772
SWI697408, -112.3344877, 33.7421594
CSW1203377, -112.325841, 33.7497868
CSW944176, -112.3316777, 33.7471236
JNC289428, -112.3371458, 33.7388695
SEP2219600, -112.3331802, 33.7361909
```

Code Editor

Franck

➔ Why It Matters

- ❏ **Quick Fixes:** Instantly resolve code issues without switching tools
- ❏ **Seamless Workflow:** Edit directly in GridAI during project uploads
- ❏ **Time-Saving:** No need to jump back and forth between IDEs



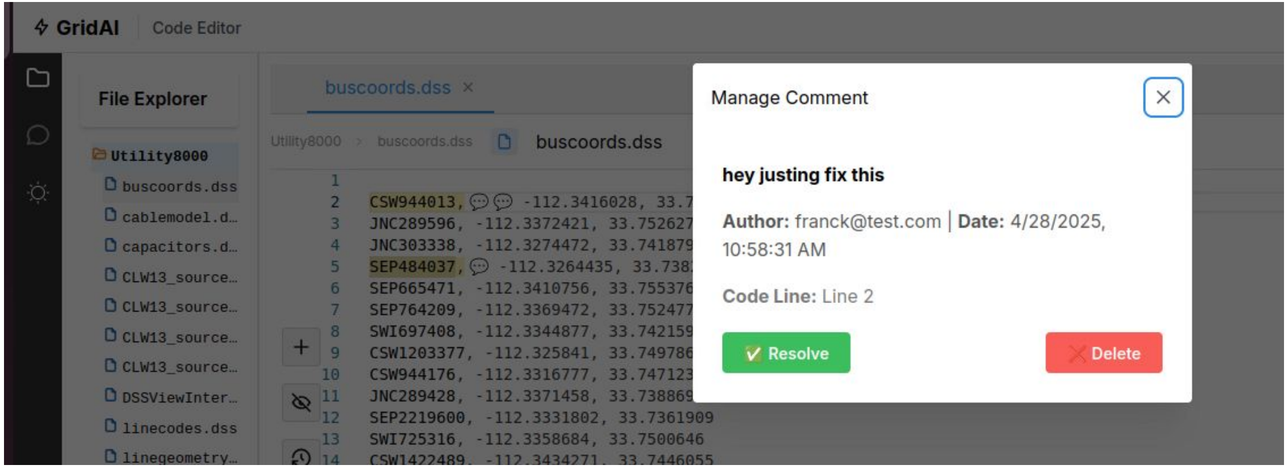
```
Element, Nterminals, Nconductors, P_1, Q_1, ...
"Vsource.source", 2, 1,    643.093,    87.8345,          0,          0
"Vsource.clw13_source_b", 2, 1,    736.961,    80.8399,          0,          0
"Vsource.clw13_source_c", 2, 1,    600.397,    60.0072,          0,          0
"Line.29608897", 2, 1, 1.71975E-11, -0.0000445419, -1.793E-11, 0.0000263948
"Line.30212899", 2, 1, -1.18971E-11, -0.0000454428, 1.08189E-11, 0.0000269298
"Line.30300192", 2, 1, -1.12781E-11, -0.0000267386, 8.92102E-12, 0.0000183827
"Line.29737418", 2, 1, 2.76384E-12, -0.0000267914, -7.36215E-12, 0.000018419
"Line.29933621", 2, 1, -4.05016E-12, -0.0000267873, -1.10983E-12, 0.0000184159
"Line.27812569", 2, 1,   -15.2157, -0.0132645,    15.2158, 0.00732755
"Line.28924792", 2, 1,   -19.9854,    8.03986,    19.9854,   -8.04487
"Line.28924747", 2, 1,    7.75225,  -0.208365,   -7.75225,   0.202388
"Line.8696606", 2, 1,    6.87472,    0.78027,   -6.87468,  -0.918757
"Line.30083507", 2, 1, 1.83807E-11, -0.0000268239, -1.81972E-11, 0.0000184409
```

Code Editor

Franck

➔ Collaboration Power

- ❏ Real-time teamwork
- ❏ Live troubleshooting
- ❏ Faster development cycles



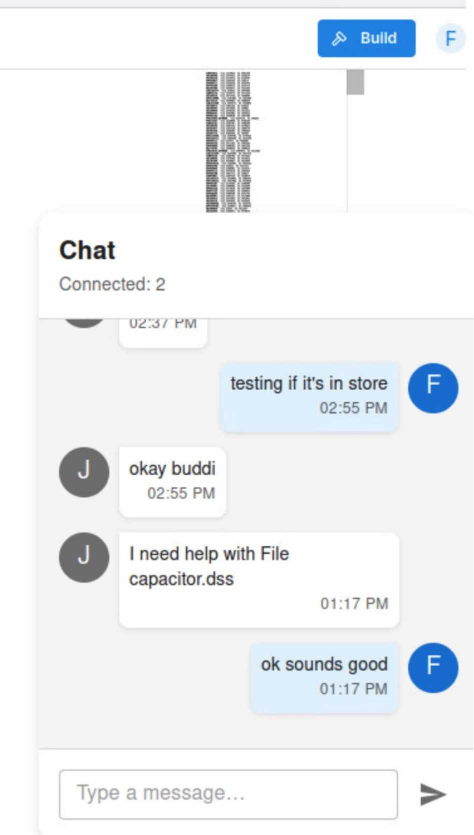
```
1
2   CSW944013, 💬💬 -112.3416028, 33.7
3   JNC289596, -112.3372421, 33.752627
4   JNC303338, -112.3274472, 33.741879
5   SEP484037, 💬 -112.3264435, 33.738
6   SEP665471, -112.3410756, 33.755376
7   SEP764209, -112.3369472, 33.752477
8   SWI697408, -112.3344877, 33.742159
9   CSW1203377, -112.325841, 33.749786
10  CSW944176, -112.3316777, 33.747123
11  JNC289428, -112.3371458, 33.738869
12  SEP2219600, -112.3331802, 33.7361909
13  SWI725316, -112.3358684, 33.7500646
14  CSW1422489, -112.3434271, 33.7446055
```

Manage Comment

hey justing fix this

Author: franck@test.com | **Date:** 4/28/2025, 10:58:31 AM

Code Line: Line 2



Chat

Connected: 2

02:37 PM

F: testing if it's in store 02:55 PM

J: okay buddi 02:55 PM

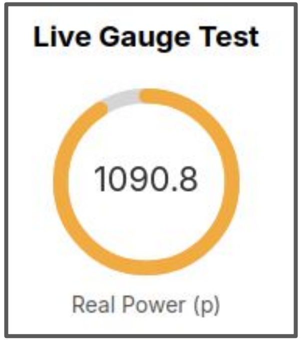
J: I need help with File capacitor.dss 01:17 PM

F: ok sounds good 01:17 PM

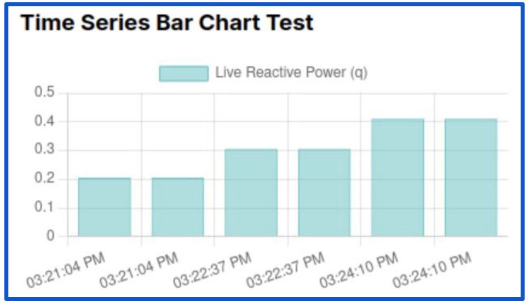
Widgets Component

Jesus

- A **widget** is a small, customizable component that visualizes real-time data, such as voltage or power.
  - Users can create and edit widgets for tailored data visualization.
  - Widgets integrate dynamically into user dashboards.
- The **task for this component** was to develop a flexible, real-time widget system for visualizing live grid data.
- The **main goal** was to enable users to create, customize, and deploy widgets with real-time updates directly into their dashboards.



Live Gauge Widget



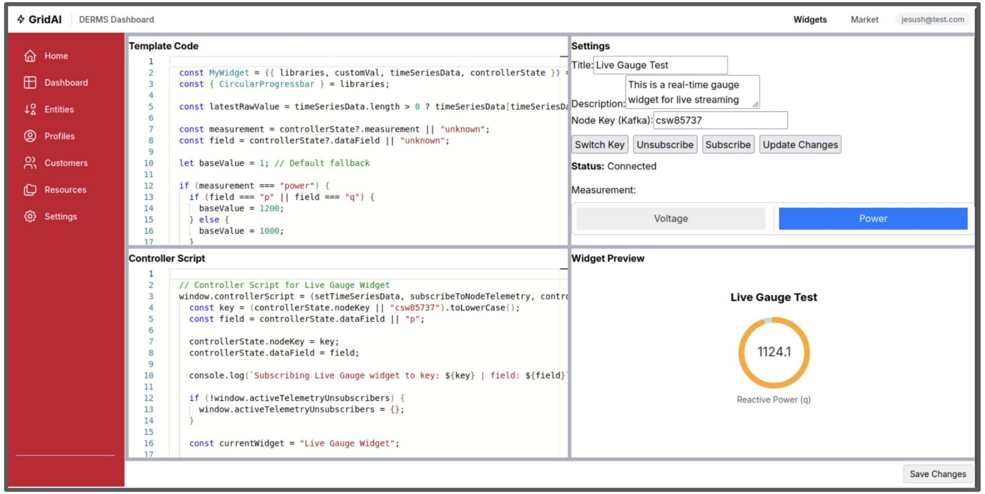
Time-Series Bar Chart Widget

Widgets Component

Jesus

➔ Implemented Action Items:

- Real-Time Updates: Widgets receive live telemetry data via Kafka event streaming platform and WebSockets.
- Full Customization: Users can edit templates, settings, and scripts to tailor widgets through widget editor.
- Diverse Widget Types: Time series charts, gauges, trends display, and more.
- Scalability: Designed to support easy creation and addition of new widget types.



Template Code

```
const MyWidget = ({ libraries, customVal, timeSeriesData, controllerState }) =
const { CircularProgressbar } = libraries;

const latestRawValue = timeSeriesData.length > 0 ? timeSeriesData[timeSeriesDa

const measurement = controllerState?.measurement || "unknown";
const field = controllerState?.dataField || "unknown";

let baseValue = 1; // Default fallback

if (measurement === "power") {
  if (field === "p" || field === "q") {
    baseValue = 1200;
  } else {
    baseValue = 1000;
  }
```

Settings

Title: Live Gauge Test

Description: This is a real-time gauge widget for live streaming

Node Key (Kafka): csw85737

Status: Connected

Measurement:

Controller Script

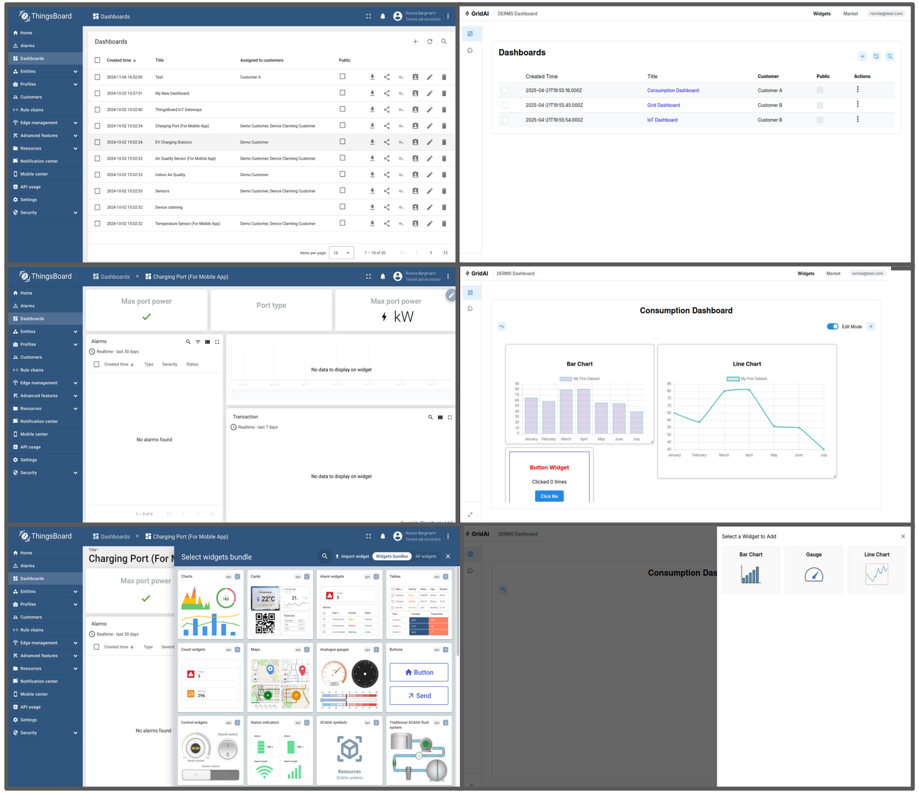
```
// Controller Script for Live Gauge Widget
window.controllerScript = (setTimeSeriesData, subscribeToNodeTelemetry, contro
  const key = (controllerState.nodeKey || "csw85737").toLowerCase();
  const field = controllerState.dataField || "p";

  controllerState.nodeKey = key;
  controllerState.dataField = field;

  console.log(`Subscribing Live Gauge widget to key: ${key} | field: ${field}`

  if (!window.activeTelemetryUnsubscribers) {
    window.activeTelemetryUnsubscribers = {};
  }

  const currentWidget = "Live Gauge Widget";
```

The widget editor allows widget customization.

Ronnie

Dashboard

- ➔ **What is a Dashboard?**
  - ◆ Allow users to create a customizable layout of widgets to view relevant data
- ➔ **Problem:**
  - ◆ Implement a working dashboard that resembles ThingsBoard
- ➔ **Approach**
  - ◆ Identify the key features that need to be implemented
  - ◆ Prototype with pre-existing libraries
  - ◆ Gradually implement changes, ensuring that components are loosely coupled
  - ◆ Continuous Integration with backend services



Dashboards

☐	Created time ↓	Title	Assigned to customers	Public
☐	2024-11-04 16:52:06	Test	Customer A	☐
☐	2024-10-02 15:07:51	My New Dashboard		☐
☐	2024-10-02 15:02:40	ThingsBoard IoT Gateways		☐
☐	2024-10-02 15:02:34	Charging Port (For Mobile App)	Demo Customer, Device Claiming Customer	☐
☐	2024-10-02 15:02:34	EV Charging Stations	Demo Customer	☐
☐	2024-10-02 15:02:33	Air Quality Sensor (For Mobile App)	Demo Customer, Device Claiming Customer	☐
☐	2024-10-02 15:02:33	Indoor Air Quality	Demo Customer	☐
☐	2024-10-02 15:02:33	Sensors	Demo Customer, Device Claiming Customer	☐
☐	2024-10-02 15:02:32	Device claiming		☐
☐	2024-10-02 15:02:32	Temperature Sensor (For Mobile App)	Demo Customer, Device Claiming Customer	☐

Items per page: 10 1 – 10 of 20

Dashboards

	Created Time	Title	Customer	Public	Actions
☐	2025-04-21T19:55:16.000Z	Consumption Dashboard	Customer A	☐	⋮
☐	2025-04-21T19:55:45.000Z	Grid Dashboard	Customer B	☐	⋮
☐	2025-04-21T19:55:54.000Z	IoT Dashboard	Customer B	☐	⋮

Comparison of ThingsBoard Dashboard and GridAI Dashboard

Market Dashboard

Justin

The market dashboard's purpose is to allow key stakeholders to participate in the wholesale electricity market, a relatively new business model introduced in 2020.

Imagine you have solar panels on your roof, a home battery, and an electric vehicle in your garage. Right now, most of that energy sits idle or just offsets your own usage.

You can now sell that excess energy into the wholesale electricity market, just like how a power plant would.

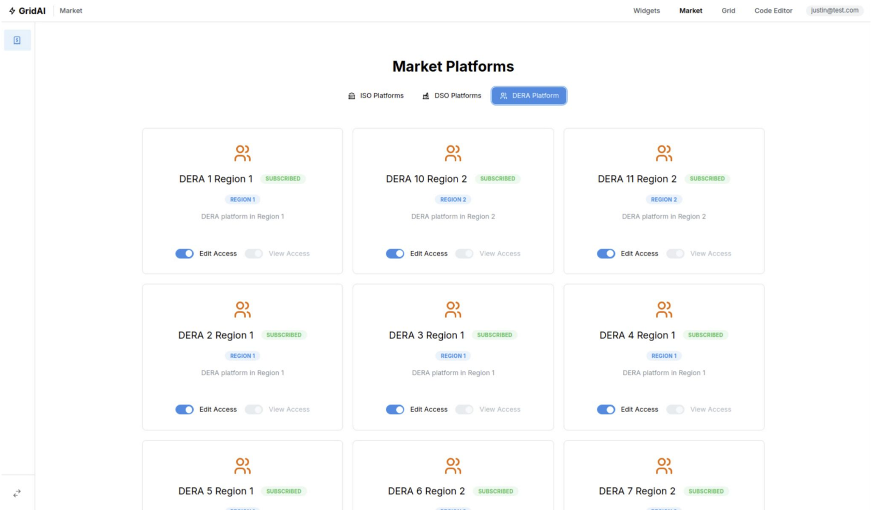
The Task

Justin

Develop a novel platform that allows key stakeholders to participate in the wholesale electricity market.

Two focus areas:

- ➔ Responsive and Intuitive Design
- ➔ Data Management and Transparency



Market Platforms

DERA 1 Region 1 SUBSCRIBED
REGION 1
DERA platform in Region 1
Edit Access View Access

DERA 10 Region 2 SUBSCRIBED
REGION 2
DERA platform in Region 2
Edit Access View Access

DERA 11 Region 2 SUBSCRIBED
REGION 2
DERA platform in Region 2
Edit Access View Access

DERA 2 Region 1 SUBSCRIBED
REGION 1
DERA platform in Region 1
Edit Access View Access

DERA 3 Region 1 SUBSCRIBED
REGION 1
DERA platform in Region 1
Edit Access View Access

DERA 4 Region 1 SUBSCRIBED
REGION 1
DERA platform in Region 1
Edit Access View Access

DERA 5 Region 1 SUBSCRIBED

DERA 6 Region 2 SUBSCRIBED

DERA 7 Region 2 SUBSCRIBED

View of the different stakeholders portfolio's

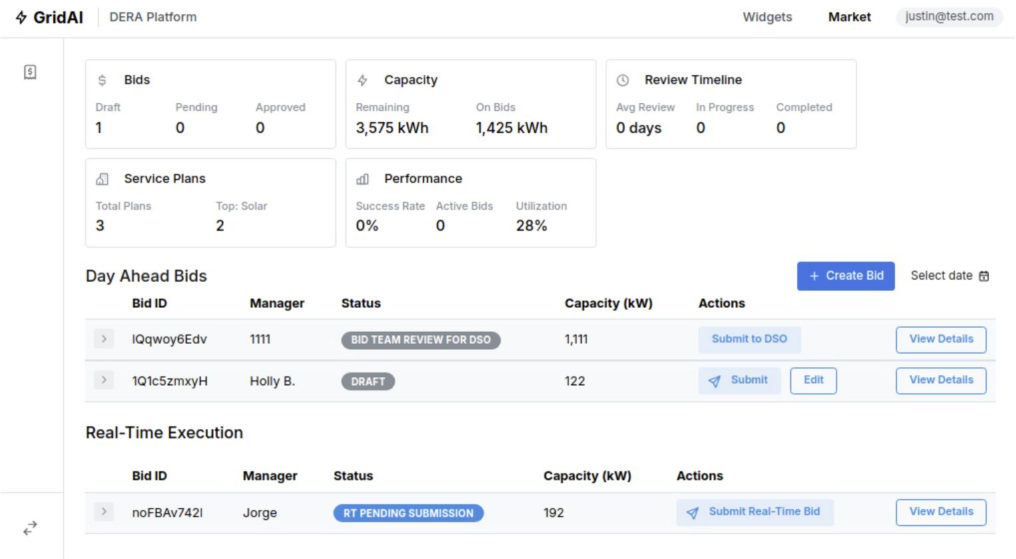
Responsive and Intuitive Design

Justin

Speed, clarity, and usability are critical when managing complex, time sensitive bid strategies.

- ➔ At-a-glance metric cards
- ➔ Table/List view for the bids
- ➔ Clean Minimal Layout

> **What are *bids*?**
> A structured offer describing a energy resource in terms of its energy quantity, price, location and delivery timeframe.



Bids

Draft	Pending	Approved
1	0	0

Capacity

Remaining	On Bids
3,575 kWh	1,425 kWh

Review Timeline

Avg Review	In Progress	Completed
0 days	0	0

Service Plans

Total Plans	Top: Solar
3	2

Performance

Success Rate	Active Bids	Utilization
0%	0	28%

Day Ahead Bids

Bid ID	Manager	Status	Capacity (kW)	Actions	
lQqwoy6Edv	1111	BID TEAM REVIEW FOR DSO	1,111	Submit to DSO	View Details
1Q1c5zmxyH	Holly B.	DRAFT	122	Submit Edit	View Details

Real-Time Execution

Bid ID	Manager	Status	Capacity (kW)	Actions	
noFBAv742l	Jorge	RT PENDING SUBMISSION	192	Submit Real-Time Bid	View Details

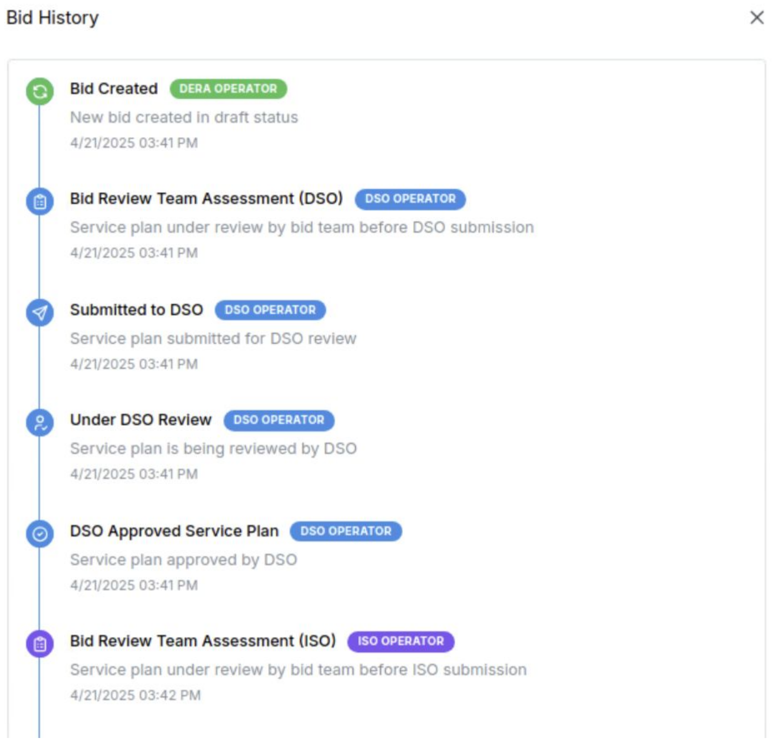
View of the DERA dashboard with metric cards and Table view

Data Management and Transparency

Justin

Transparent data flows and clear audit trails build trust and empower all stakeholders to operate with confidence.

- ➔ Detailed bid history
- ➔ Clear role attribution
- ➔ Integration and Compliance



Bid History ×

Bid Created DERA OPERATOR
New bid created in draft status
4/21/2025 03:41 PM

Bid Review Team Assessment (DSO) DSO OPERATOR
Service plan under review by bid team before DSO submission
4/21/2025 03:41 PM

Submitted to DSO DSO OPERATOR
Service plan submitted for DSO review
4/21/2025 03:41 PM

Under DSO Review DSO OPERATOR
Service plan is being reviewed by DSO
4/21/2025 03:41 PM

DSO Approved Service Plan DSO OPERATOR
Service plan approved by DSO
4/21/2025 03:41 PM

Bid Review Team Assessment (ISO) ISO OPERATOR
Service plan under review by bid team before ISO submission
4/21/2025 03:42 PM

View of the bid history modal that allows a clear view of what was done and edited

Single-Line Diagram - Introduction

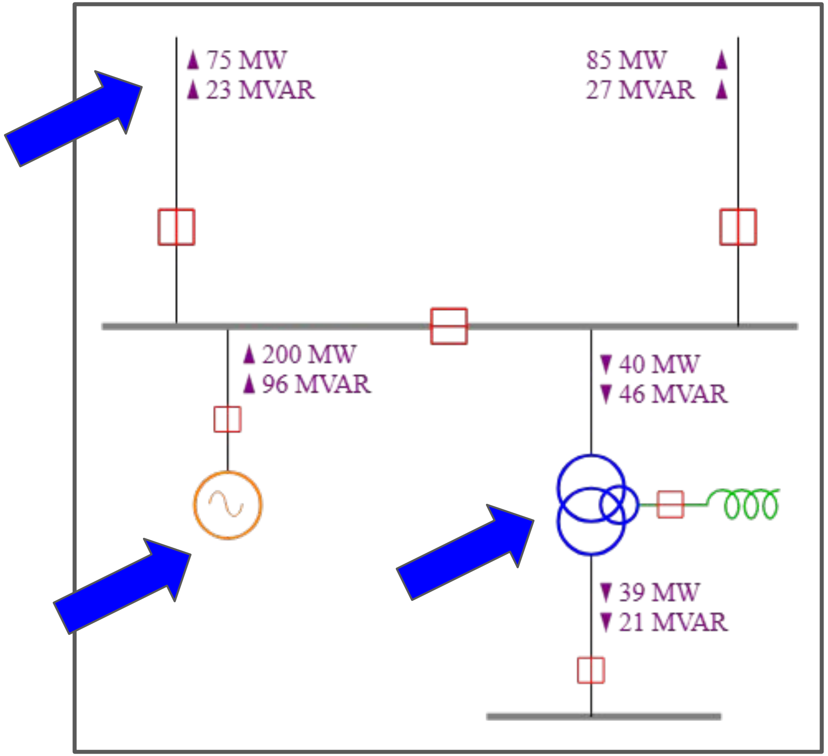
Skyler

What is a SLD?

- Common for electrical systems
- Expected elements
- High-level overview

Purpose/Goal

- Simple intuitive design
- Display relevant information
- Allow edits and movement



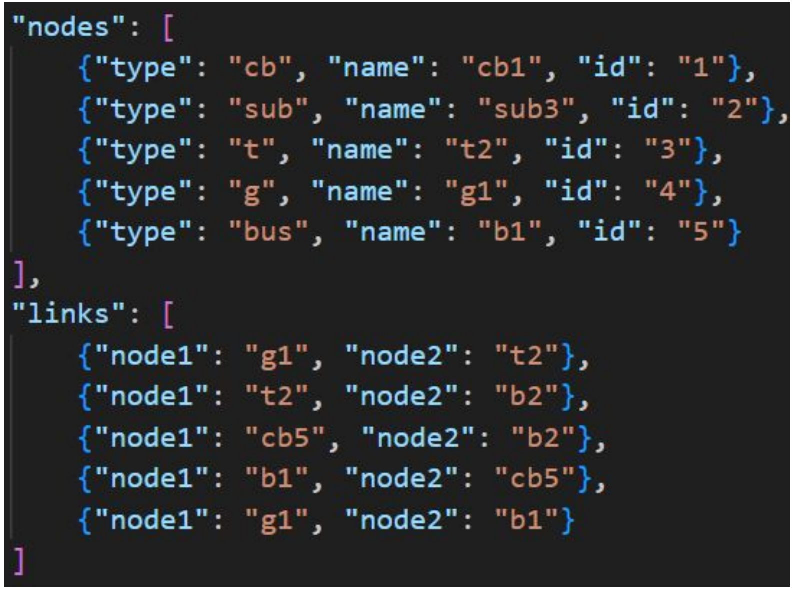
Single-Line Diagram - Challenges

Key Requirements:

- Translate given data to SLD
- Use SLD symbols and layout
- Display info in a way that's easy to read and manage

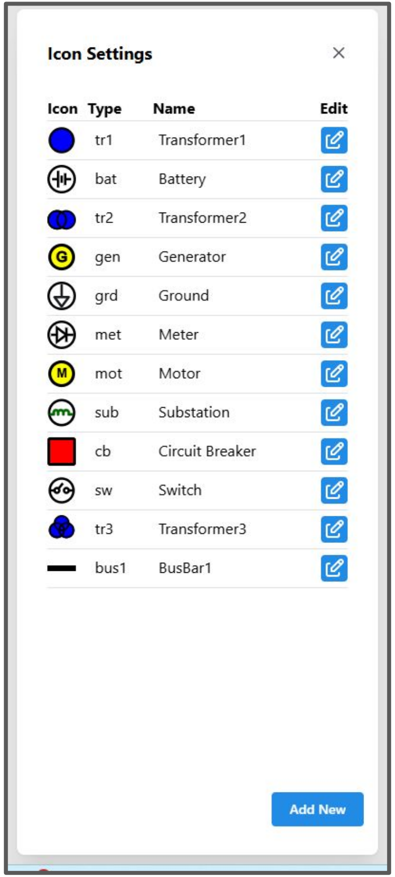
Challenges

- How can I create the layout?
- How many symbols should be included?



```
"nodes": [
    {"type": "cb", "name": "cb1", "id": "1"},
    {"type": "sub", "name": "sub3", "id": "2"},
    {"type": "t", "name": "t2", "id": "3"},
    {"type": "g", "name": "g1", "id": "4"},
    {"type": "bus", "name": "b1", "id": "5"}
],
"links": [
    {"node1": "g1", "node2": "t2"},
    {"node1": "t2", "node2": "b2"},
    {"node1": "cb5", "node2": "b2"},
    {"node1": "b1", "node2": "cb5"},
    {"node1": "g1", "node2": "b1"}
]
```

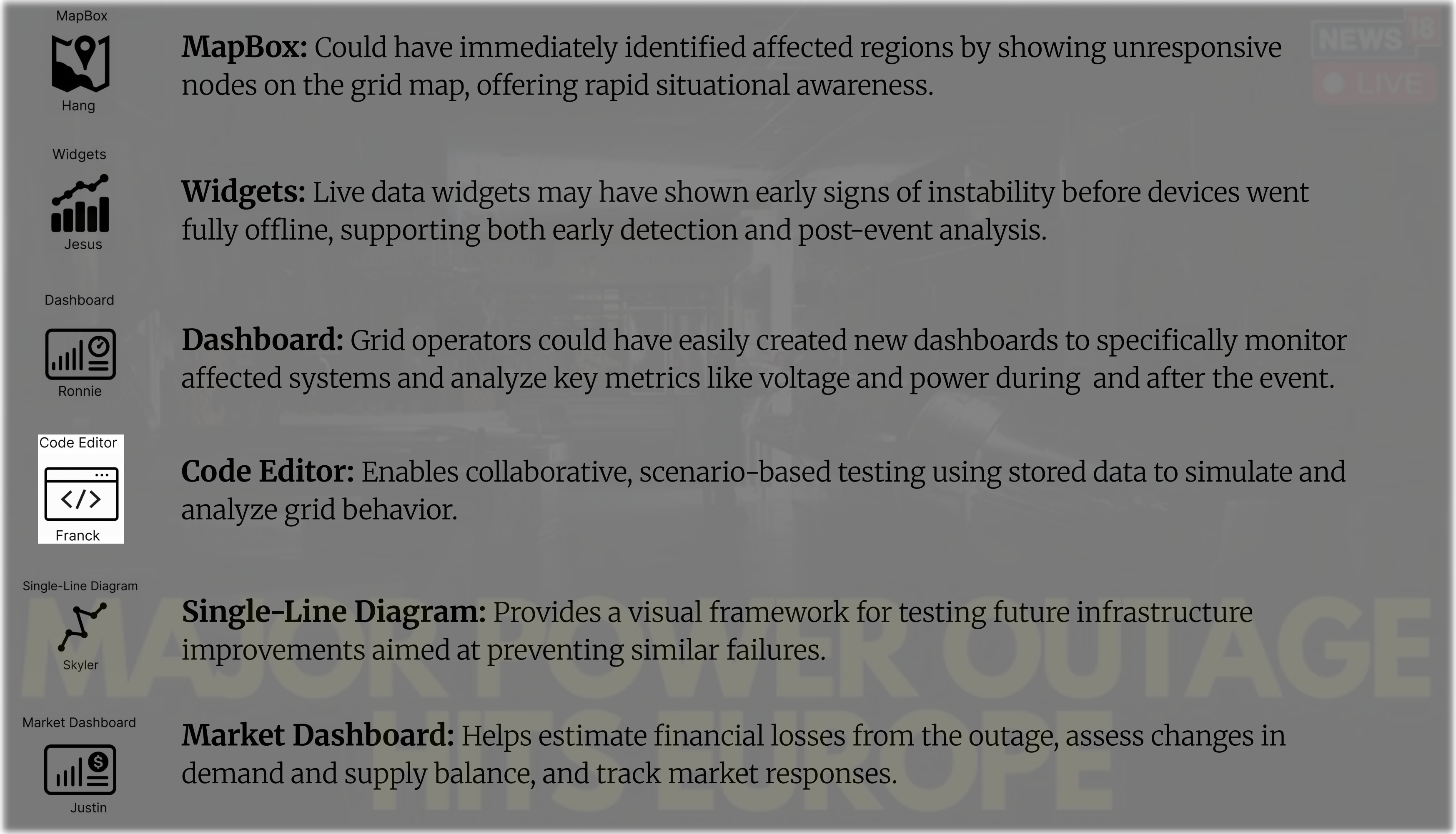
Single-Line Diagram - Solutions



Icon Settings ×

Icon	Type	Name	Edit
●	tr1	Transformer1	✎
⊕	bat	Battery	✎
●●	tr2	Transformer2	✎
G	gen	Generator	✎
⏚	grd	Ground	✎
⊛	met	Meter	✎
M	mot	Motor	✎
○	sub	Substation	✎
🟥	cb	Circuit Breaker	✎
⊘	sw	Switch	✎
●	tr3	Transformer3	✎
▬	bus1	BusBar1	✎

European Power Outage – How could GridAI have helped?



MapBox

Hang

MapBox: Could have immediately identified affected regions by showing unresponsive nodes on the grid map, offering rapid situational awareness.

Widgets

Jesus

Widgets: Live data widgets may have shown early signs of instability before devices went fully offline, supporting both early detection and post-event analysis.

Dashboard

Ronnie

Dashboard: Grid operators could have easily created new dashboards to specifically monitor affected systems and analyze key metrics like voltage and power during and after the event.

Code Editor

Franck

Code Editor: Enables collaborative, scenario-based testing using stored data to simulate and analyze grid behavior.

Single-Line Diagram

Skyler

Single-Line Diagram: Provides a visual framework for testing future infrastructure improvements aimed at preventing similar failures.

Market Dashboard

Justin

Market Dashboard: Helps estimate financial losses from the outage, assess changes in demand and supply balance, and track market responses.

Project Takeaways:

- **Widgets:** Working on the widget system taught me how to design flexible, real-time components that balance user customization with live data integration.
- **Dashboard:** Implementing the dashboard component taught me the importance of taking the time to learn and understand an existing code base to help with efficient development of the component.
- **Market Dashboard:** Implementing a novel platform for a new developing market showed me the importance of developing flexible and reusable front end components.
- **MapBox:** Expected design and the actual design are different. The more I spend time on the project and work on my component, I got new ideas and have solution to the problem I have from previou week.
- **Single-Line Diagram:** Creating the single-line diagram component showed me the importance of planning. The app could have been developed a lot faster if the proper planning had taken place.
- **Code Editor:** Working on this code editor taught me how to manage complex frontend-backend interactions and reinforced the importance of clean API design.

Conclusion

- GridAI provides a unified platform for users to observe, interact with, and manage grid data through dynamic, customizable components all integrated into a single web-based system.

- The platform brings together key capabilities:
  - Mapbox Visualization for geographic grid data mapping.
  - Code Editor for collaborative project and script management.
  - Dashboard and Widgets for real-time telemetry visualization and customization.
  - Single-Line Diagram for high-level system overviews.
  - Market Dashboard for supporting DER aggregators in energy market participation.

- For the last two semesters our team worked to enhanced GridAI’s frontend architecture, expanded real-time data capabilities, and improved overall user experience.

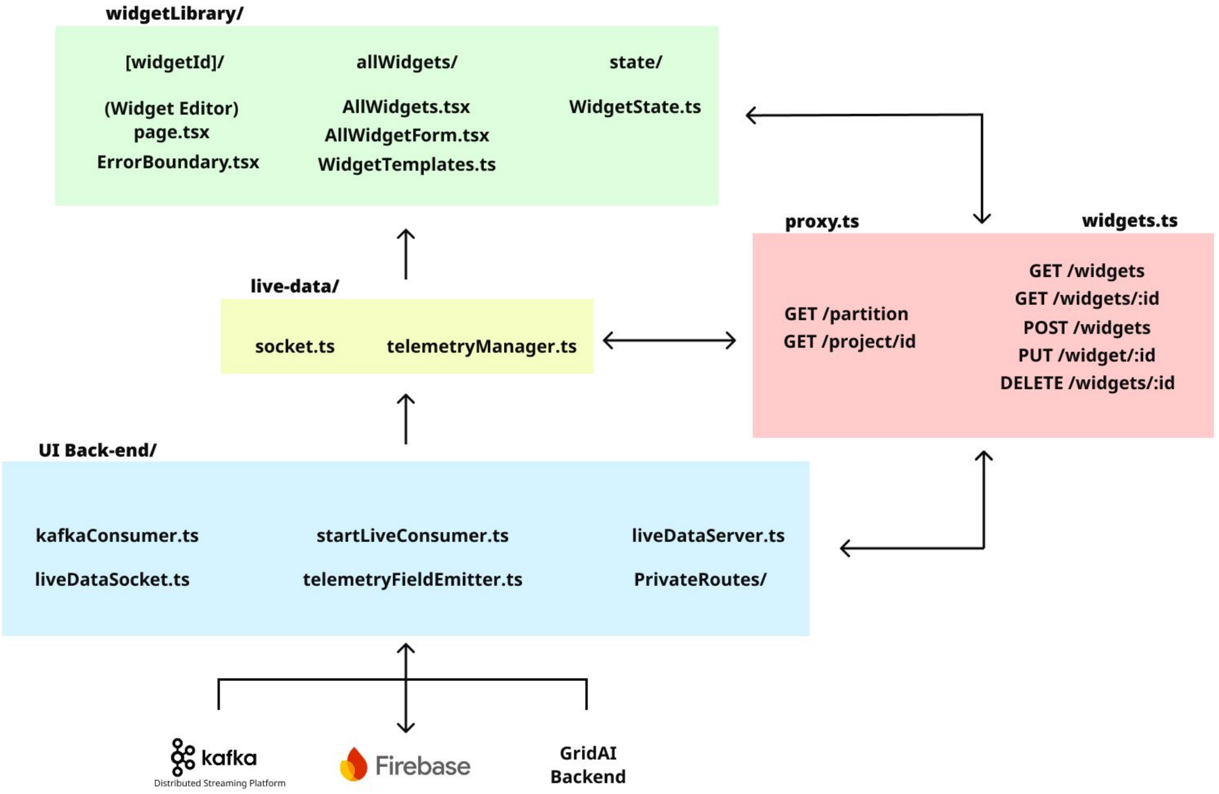
Acknowledgment

This research is partially supported by the U.S. NSF Grant # CNS-2105269, U.S. DOE CESER Grant DE-CR000016, and the Iowa Energy Center Grant #21-IEC-009.

Thank you

Backup Slides

Widgets Architecture Flow



widgetLibrary/

[widgetId]/

(Widget Editor) page.tsx

ErrorBoundary.tsx

allWidgets/

AllWidgets.tsx

AllWidgetForm.tsx

WidgetTemplates.ts

state/

WidgetState.ts

live-data/

socket.ts

telemetryManager.ts

proxy.ts

GET /partition

GET /project/id

widgets.ts

GET /widgets

GET /widgets/:id

POST /widgets

PUT /widget/:id

DELETE /widgets/:id

UI Back-end/

kafkaConsumer.ts

liveDataSocket.ts

startLiveConsumer.ts

telemetryFieldEmitter.ts

liveDataServer.ts

PrivateRoutes/

Widget Lifecycle and Data Flow

DERMS

→ Widget Lifecycle:

- Widgets are created from a template.
- Users can customize widgets in the Widget Editor.
- Users can subscribe to receive telemetry data providing a Node Key.
- Changes can be saved to Firebase.
- A live preview of the functional widget is provided to support customization.
- Saved widgets can be added to user dashboards for real-time data observation.

→ Live Data Flow:

- Real-time data is streamed from the GridAI backend using Kafka.
- Kafka messages are consumed and filtered by project ID, partition, and node name before being forwarded.
- WebSocket feeds live data to the frontend interface.
- Controller scripts dynamically process incoming telemetry updates for each widget instance.
- Widgets render updated visualizations based on real-time data streams.

Dashboards

Created time | Title | [illegible] | [illegible] | [illegible]

[illegible]

Title

[illegible]

Description

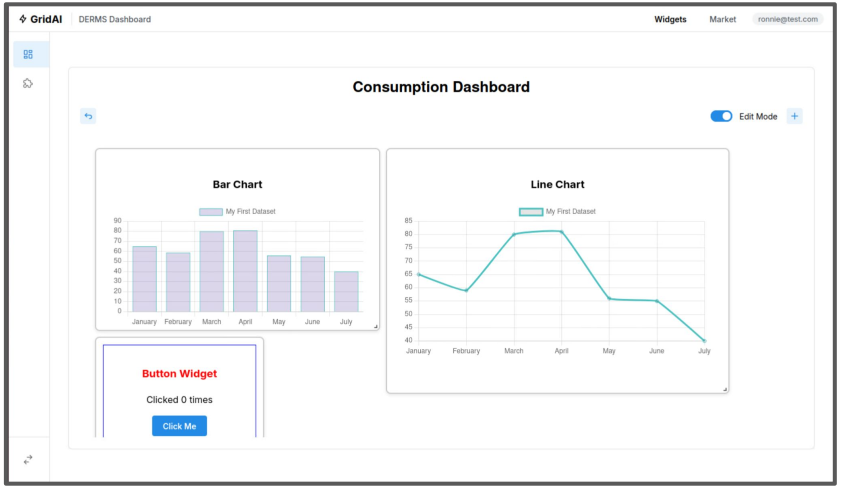
[illegible]

[illegible]

Dashboard

Ronnie

- **GridAI in the Real World**
  - Power outage in the Iberian Peninsula last week
  - Grid operators could view relevant dashboards to paint a picture of the outage
    - Create new dashboards, specifically monitoring affected systems
  - Consumers could get insight into the extent of the outage
  - All in an intuitive and user centric design

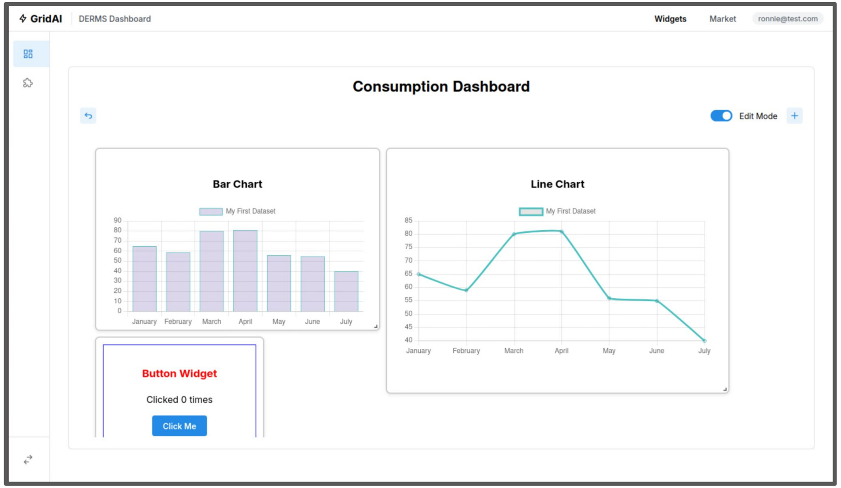


Example Dashboard with Widgets

Dashboard

Ronnie

- **What is a Dashboard?**
  - Allow users to create a customizable layout of widgets to view relevant data
- **Problem:**
  - Implement a working dashboard that resembles ThingsBoard
- **Key Features**
  - Widgets need to be draggable in the dashboard
  - Support of multiple dashboards
  - Add, Import, Export dashboard configurations
  - *View* mode and *Edit* mode
  - *Search* and *Sort* capabilities
  - Ability to display real-time data
  - Full integration with widget components



Example Dashboard with Widgets

Dashboard

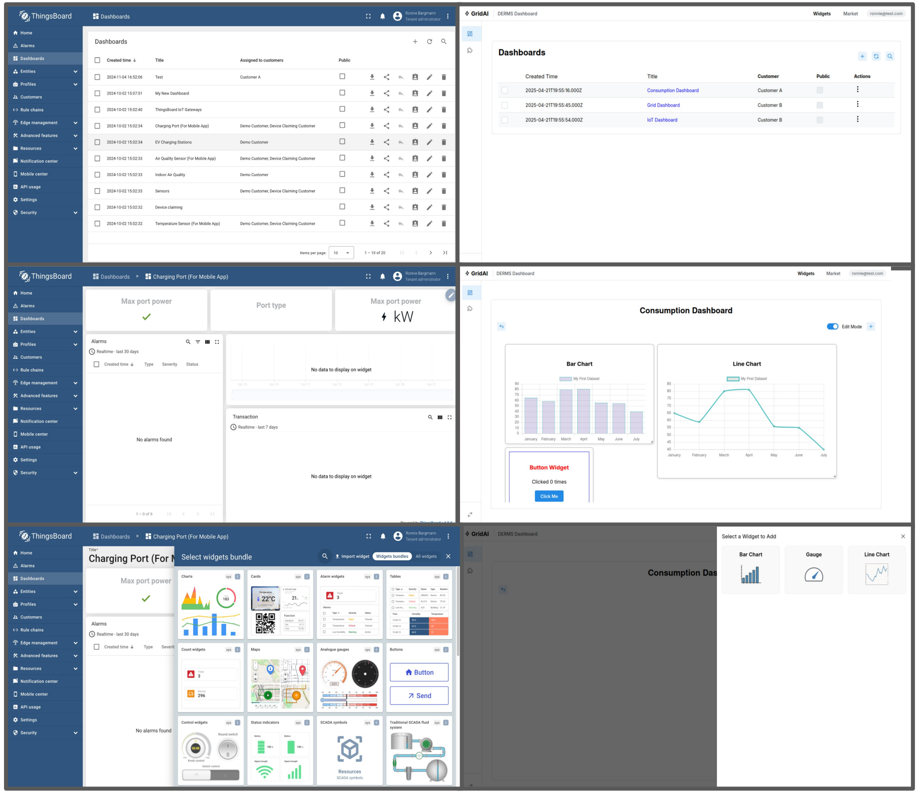
Ronnie

➔ **Key Requirements:**

- User friendly and responsive design
- Unified theme throughout the platform
- Intuitive user experience

➔ **Approach:**

- Identify the key features that need to be implemented
- Prototype with pre-existing libraries
- Gradually implement changes, ensuring that components are loosely coupled
- Continuous Integration with backend services



ThingsBoard — Dashboards — Ronnie Bargmann, Tenant administrator

Dashboards

□	Created time ↓	Title	Assigned to customers	Public
□	2024-11-04 16:52:06	Test	Customer A	□
□	2024-10-02 15:07:51	My New Dashboard		□
□	2024-10-02 15:02:40	ThingsBoard IoT Gateways		□
□	2024-10-02 15:02:34	Charging Port (For Mobile App)	Demo Customer, Device Claiming Customer	□
□	2024-10-02 15:02:34	EV Charging Stations	Demo Customer	□
□	2024-10-02 15:02:33	Air Quality Sensor (For Mobile App)	Demo Customer, Device Claiming Customer	□
□	2024-10-02 15:02:33	Indoor Air Quality	Demo Customer	□
□	2024-10-02 15:02:33	Sensors	Demo Customer, Device Claiming Customer	□
□	2024-10-02 15:02:32	Device claiming		□
□	2024-10-02 15:02:32	Temperature Sensor (For Mobile App)	Demo Customer, Device Claiming Customer	□

Items per page: 10 1 – 10 of 20

GridAI — DERMS Dashboard — Widgets Market ronnie@test.com

Dashboards

	Created Time	Title	Customer	Public	Actions
□	2025-04-21T19:55:16.000Z	Consumption Dashboard	Customer A	□	⋮
□	2025-04-21T19:55:45.000Z	Grid Dashboard	Customer B	□	⋮
□	2025-04-21T19:55:54.000Z	IoT Dashboard	Customer B	□	⋮

Comparison of ThingsBoard Dashboard and GridAI Dashboard