AgriSense

VERTICAL FARMING MANAGEMENT SYSTEM

Presented by Andreas Galanis & Konstantinos Kostakopoulos

Team 10

TODAY'S AGENDA

1 System Requirements

2 System Architecture

3 Technology and Tools

4 Timetable

5 Data Organization

6 FIWARE

7 Prototype’s Features

8 Demo / Drafts

SYSTEM REQUIREMENTS

- User Access Levels
  - User: Limited permissions, including viewing live metrics and sensor status.
  - Admin: Full access to all controls, settings, and system configurations.

- Frontend Technologies
  - User Accessibility and Ease of Use
  - Real-Time Data Display
  - Customization and Flexibility
  - Secure and Efficient Data Handling

- Backend Framework
  - Data Management and Processing
  - User Authentication and Authorization
  - API for Frontend Communication
  - Sensor Data Integration
  - Edge Computing Compatibility
  - AI and Machine Learning Integration

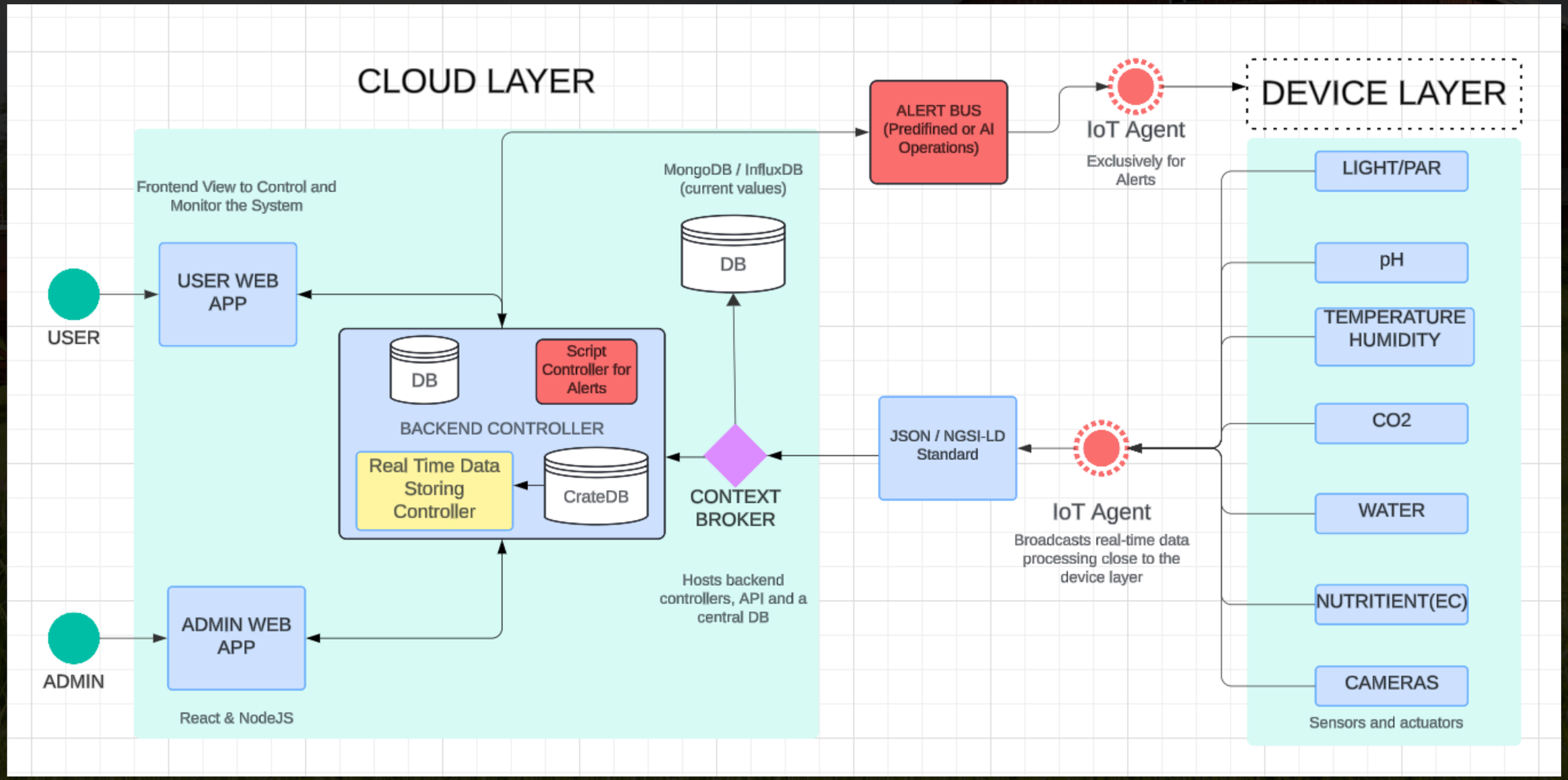
SYSTEM REQUIREMENTS

- Database
  - Sensor Data
  - User Data
  - AI Insights and Alerts
  - Automated Backup

- Sensors
  - pH Sensor
  - Temperature
  - Humidity
  - Light/PAR
  - Electrical Conductivity (EC)
  - CO_2
  - Water

- Data Library
  - Historical Data
  - AI-Enhanced Data Library
  - Presaved Configurations
  - Training Data

SYSTEM ARCHITECTURE



TECH AND TOOLS

- Frontend Stack
  - Vanilla JavaScript: Core scripting language for interactivity.
  - HTML & CSS: For structure and styling of the web interface.
  - React: Planned for better state management, improved UI scalability, and dynamic components.
- Backend Stack
  - Framework: Node.js with Express
    - Handles API requests for sensor data and user actions.
    - Supports CRUD operations.
  - Data Handling:
    - Efficiently processes data streams from sensors for real-time updates.
- Database Choices
  - MySQL/PostgreSQL:
    - Best for structured, relational data and complex querying.
    - Ensures high reliability and consistency for user data and sensor logs.
  - MongoDB / InfluxDB:
    - Handles varied and semi-structured sensor data

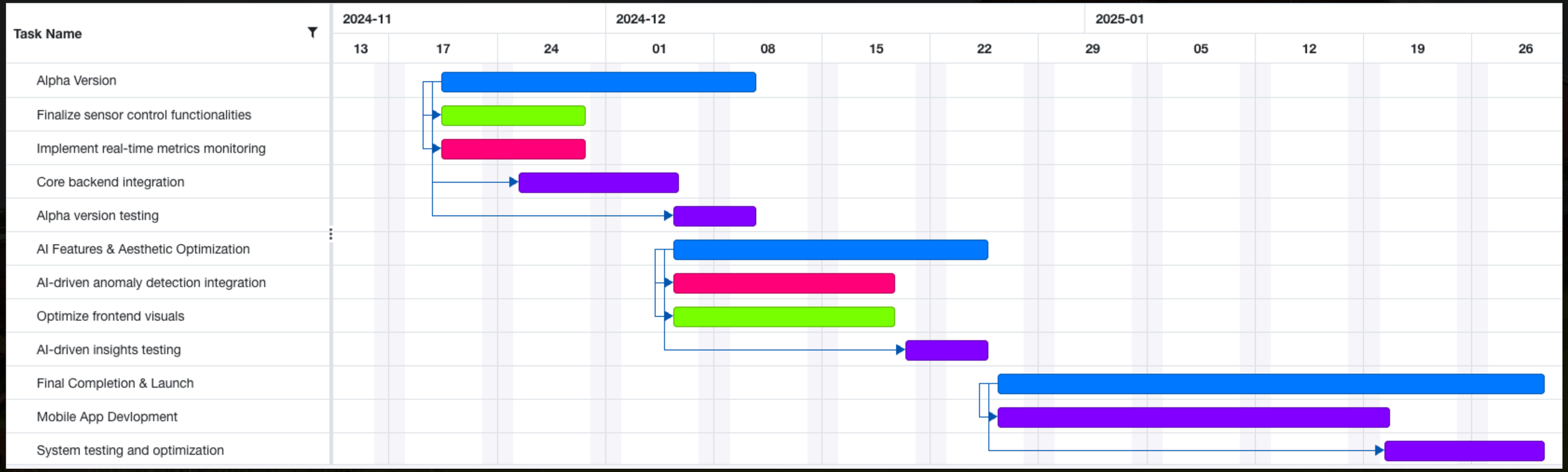
TECH AND TOOLS

- Monitoring Tools
  - Grafana:
    - Real-time visualization platform for sensor metrics.
    - Features customizable dashboards for different user roles (User/Admin).
    - Enables historical trend analysis and system health monitoring.

- AI Models
  - Yield Forecasting: Based on historical data and environmental conditions.(eg. Support Vector Regression)
  - Optimal Growth Suggestions: AI-driven recommendations for temperature, humidity, and nutrient adjustments. (Deep Reinforcement Learning Algorithms, eg.PPO)
  - Anomaly detection: Algorithms to detect sudden anomalies in sensors' conditions (eg. Isolation Forest)

- Sensors
  - DFRobot Analog pH Sensor Kit (SEN0161) ($20–$30)
  - DHT11 ($3–$5)
  - GY-302 BH1750 Light Sensor ($5–$8)
  - DFRobot Analog Electrical Conductivity Sensor (DIY Kit) ($35–$40)
  - Ultrasonic Sensor ($3–$5)
  - MH-Z19B CO_2 Sensor ($20–$30)

TIMETABLE



- Andreas
- Konstantinos
- Shared

DATA ORGANIZATION

DATA TYPES

SENSOR DATA

- Real-time measurements from sensors (e.g., temperature, humidity, pH, CO_2, light intensity).
- Historical data for trend analysis and anomaly detection.

USER DATA

- Profiles for User and Admin roles.
- System preferences and access logs.

AI GENERATED

- Predictive recommendations and alerts based on historical trends.
- Identified anomalies in environmental metrics.

CONFIGURATION DATA

- Preset thresholds for sensor operations.
- Automation rules for adjusting environmental parameters.

DATA MODELING

ENTITY-RELATIONSHIP DIAGRAM (ERD)

- Entities:
  - Sensors: Includes sensor types, configurations, and readings.
  - Users: Stores user roles, credentials, and preferences.
  - AI Insights: Logs recommendations, anomalies, and system-generated actions.

- Relationships:
  - Users ↔ Actions: Tracks user-initiated control actions.
  - Sensors ↔ Readings: Links sensors to their respective real-time and historical data.

SMART DATA MODELS

- Interoperability:
  - Common structure for data so that systems.
  - Based on widely accepted standards.
- Scalability:
  - Enable easy extension to include additional data fields or new types.
- Domain-Specific:
  - Predefined templates for specific use.
- Cloud and Edge Compatibility:
    - Designed to be compatible with cloud services.
- Ease of Integration:
- Smart Data Models are structured in common formats like JSON or XML.

DATA FLOW

- Input:
  - Sensor readings collected via microcontrollers/dummy data (e.g., Arduino UNO).

- Processing:
  - Real-time analytics through backend and cloud services.

- Output:
  - Visualized data on dashboards (Grafana).
  - Alerts and reports for users via the app interface.

FIWARE

KEY FEATURES

- Orion Context Broker:
  - Manages real-time contextual data.
  - Enables:
    - Collection of real-time sensor data.
    - Storage and querying of current system states.
    - Central hub for communication between system components.
- NGSI-LD Standard:
  - Ensures that data models are interoperable.
  - Aligns with the Smart Data Models.
- IoT Agent:
  - Bridges communication between IoT devices and the Orion Context Broker.
  - Supports multiple protocols like MQTT and HTTP.
- Data Processing and Analytics:
  - Supports integration with external tools for advanced analytics.
  - Enables predictive insights and anomaly detection.

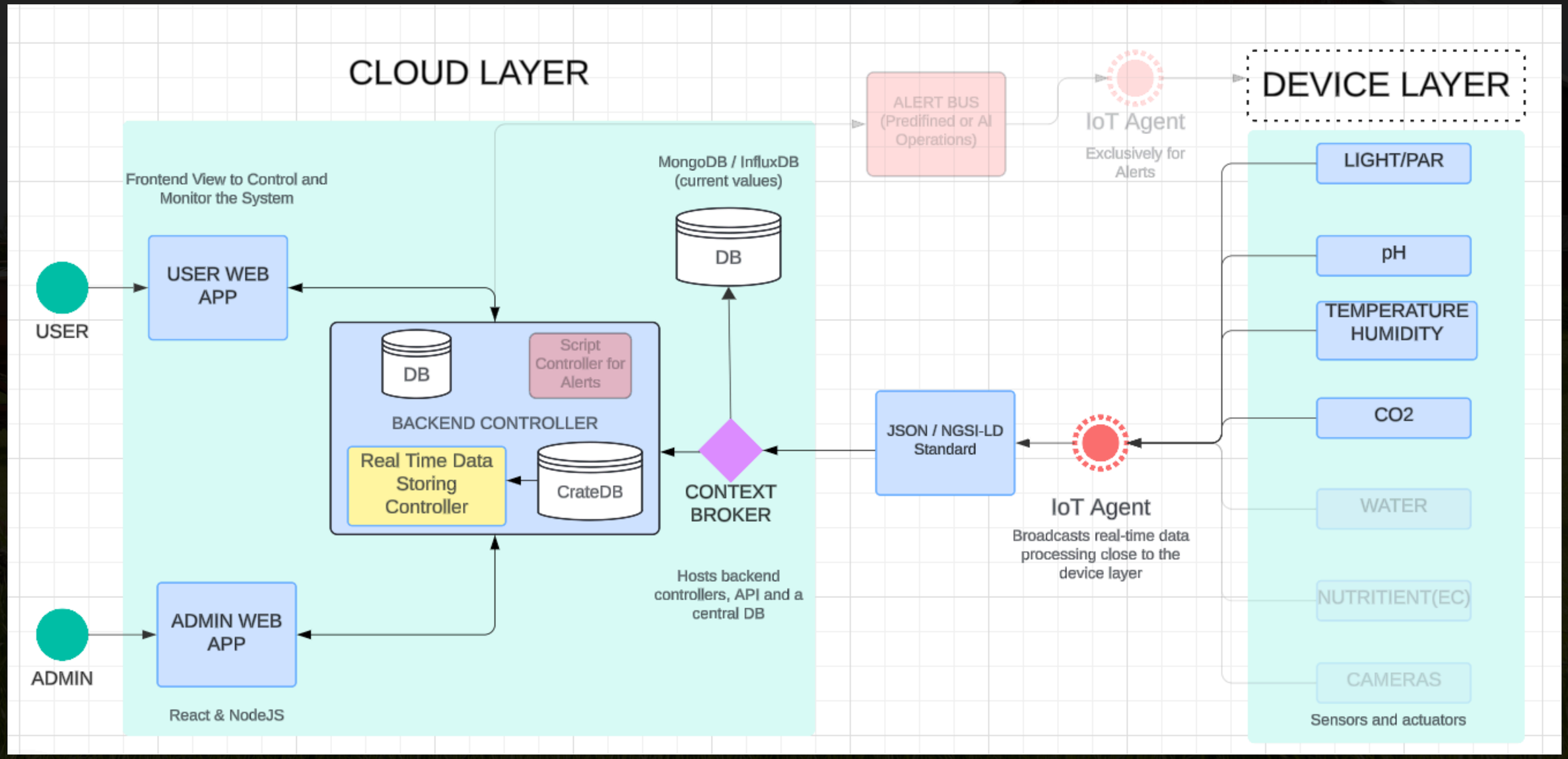
BENEFITS FOR VERTICAL FARMING

- Interoperability:
  - Ensures seamless communication between sensors, backend, and dashboards using NGSI-LD standards.
- Scalability:
  - Allows the system to scale as more sensors or features are added without major architectural changes.
- Real-Time Data Management:
  - Orion Context Broker ensures immediate access to sensor data, enabling real-time monitoring and decision-making.
- Integration with AI and IoT:
  - Simplifies the addition of AI models or external IoT tools for predictive analytics and automation.
- Open-Source Advantage:
  - Reduces costs by leveraging free, open-source components while ensuring high-quality performance.

IN ACTION

- Data Flow:
  - Sensors → IoT Agent → Orion Context Broker → Backend → Dashboard.
  - Real-time sensor readings are processed by FIWARE components and visualized through tools like Grafana or the web app.

PROTOTYPE'S FEATURES



PROTOTYPE'S FEATURES

REAL-TIME SENSOR OPERATION

- Integration with key sensors:
  - Temperature & Humidity (DHT11).
  - pH Sensor (DFRobot SEN0161).
  - Light Sensor (GY-302 BH1750).
  - CO_2 Sensor (MH-Z19B).

BASIC CONTROL FEATURES

- Turn sensors or associated devices (e.g., air, lights) on/off.
- Adjust sensor thresholds manually through the interface.

METRICS MONITORING

- Grafana Integration:
  - Real-time visualization of sensor data on customizable dashboards.
  - Historical data tracking for trend analysis.
- Support for dummy data during testing to simulate real-world operations.

BASIC USER ROLES

- Role-based access control:
  - Admin: Full control of the system, including sensor settings and configurations.
  - User: Limited access for viewing metrics and monitoring data.

DATA STORAGE

- Basic database setup to store:
  - Sensor readings.
  - User configurations and actions.
  - Support for retrieving historical data.

DEMO – DRAFTS

Grafana Dashboard through Gi

kostaskos.github.io/github-page-pwa/

HM&TY | UPnet mail | Progress | Eclass | YouTube

Vertical Farm's Sensors Data Dashboard

AgriSense

THANK YOU

for your time and attention

Presented by Team 10