High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components of **AgroVision - Smart Agriculture Dashboard**. AgroVision is a futuristic, research-oriented dashboard for smart agriculture, providing real-time sensor data visualization, advanced crop management forms, and role-based access. The solution leverages React, TypeScript, Tailwind CSS, and Redux Toolkit to deliver a responsive, scalable, and secure user experience.

System Architecture Overview

Architecture Summary:
AgroVision is a single-page application (SPA) structured around modular React components, centralized state management, and responsive UI. It interfaces with backend APIs for real-time sensor data and user management.

Module/Component	Role/Responsibility
UI Layer (React + Tailwind)	Renders dashboard, forms, and visualizations
State Management (Redux Toolkit)	Manages global state, user roles, and sensor data
API Integration Layer	Handles communication with backend APIs
Auth & Access Control	Manages authentication and role-based authorization
Data Visualization	Displays real-time sensor data (charts, graphs, etc.)
Crop Management Forms	Advanced forms for crop and field data input
Responsive Design Engine	Ensures usability across devices

Component Interactions

Source Component	Target Component	Interaction Type / Purpose
UI Layer	State Management	Dispatches actions, subscribes to state
State Management	API Integration Layer	Triggers API calls for data fetch/update
API Integration Layer	State Management	Updates state with API responses
Auth & Access Control	UI Layer	Controls access to routes/components
Data Visualization	State Management	Receives sensor data for rendering
Crop Management Forms	API Integration Layer	Submits/updates crop data

Interaction Flow Example:

1. User logs in → Auth module validates → UI updates based on role

2. UI requests sensor data → Redux dispatches API call → Data returned, state updated → Visualization re-renders

---

Data Flow Overview

Data Source	Flow Path	Destination	Purpose
Sensor API	API Integration → State Management → UI Layer	Data Visualization	Real-time sensor display
User Input	UI Layer → State Management → API Integration	Backend API	Crop management, user actions
Auth Service	API Integration → State Management → UI Layer	Auth & Access Control	Role-based access, sessions

---

Technology Stack

Layer/Function	Technology/Framework
Frontend Framework	React, TypeScript
Styling/UI	Tailwind CSS, HTML, CSS
State Management	Redux Toolkit
Data Visualization	(e.g., Recharts, Chart.js)*
API Communication	Fetch/Axios (RESTful APIs)
Authentication	JWT/OAuth (via backend)*
Responsive Design	Tailwind CSS

*Specific libraries to be finalized during implementation.

Scalability & Reliability

- **Scalability:** Modular React components and Redux enable easy feature expansion. API-driven architecture supports horizontal scaling.
- **Reliability:** Centralized state management ensures data consistency. Role-based access and authentication enhance security.
- **Security:** All sensitive operations require authentication; role-based access restricts features per user type. Data is transmitted over secure channels (HTTPS).

End of Document