## **Odoo x Charusat**

### **1. Team Information**

* **Team Name**:
* **Team Members**:
  + Nilay Patel – Backend Developer & Project Lead (Node.js, System Architecture)
  + Vasu Modi - Full Stack Developer & ML Engineer (React, Node.js, Python, Data Science, Algorithm Design)
  + Rishee Gandhi – Frontend Developer (React, Design Systems)
  + Naitik Desai -

### **2. Problem Statement**

#### **Chosen Problem:**

Public infrastructure issues often go unnoticed or unresolved due to a lack of a transparent system for reporting, tracking, and managing solutions. Citizens are unaware of project progress, and the contractor selection process lacks accountability.

#### **Problem Analysis:**

* **Ineffective Reporting System:** Citizens struggle to report issues related to public infrastructure and government-owned properties.
* **Lack of Transparency:** There is limited visibility into how public projects are assigned, managed, and completed.
* **No Centralized Platform:** Communities cannot track the status of government projects or raise concerns about delays or substandard work.
* **Contractor Selection and Accountability:** Contractors are often selected without community involvement or feedback, which may lead to poor outcomes.
* **Limited Public Participation:** Citizens have no structured way to participate in decisions affecting their local environment or infrastructure.

#### **Target Audience:**

**Primary Users**

* **Local Public/Citizens**
  + To report infrastructure issues and monitor government projects.
  + Ensure contractor accountability through feedback and reporting.
* **Government Entities**
  + Municipal authorities, public works departments, and local bodies.
  + Review contractor proposals, approve projects, and ensure transparency.
* **Verified Contractors**
  + Submit project proposals with cost estimates and timelines.
  + Update work status for public visibility and accountability.

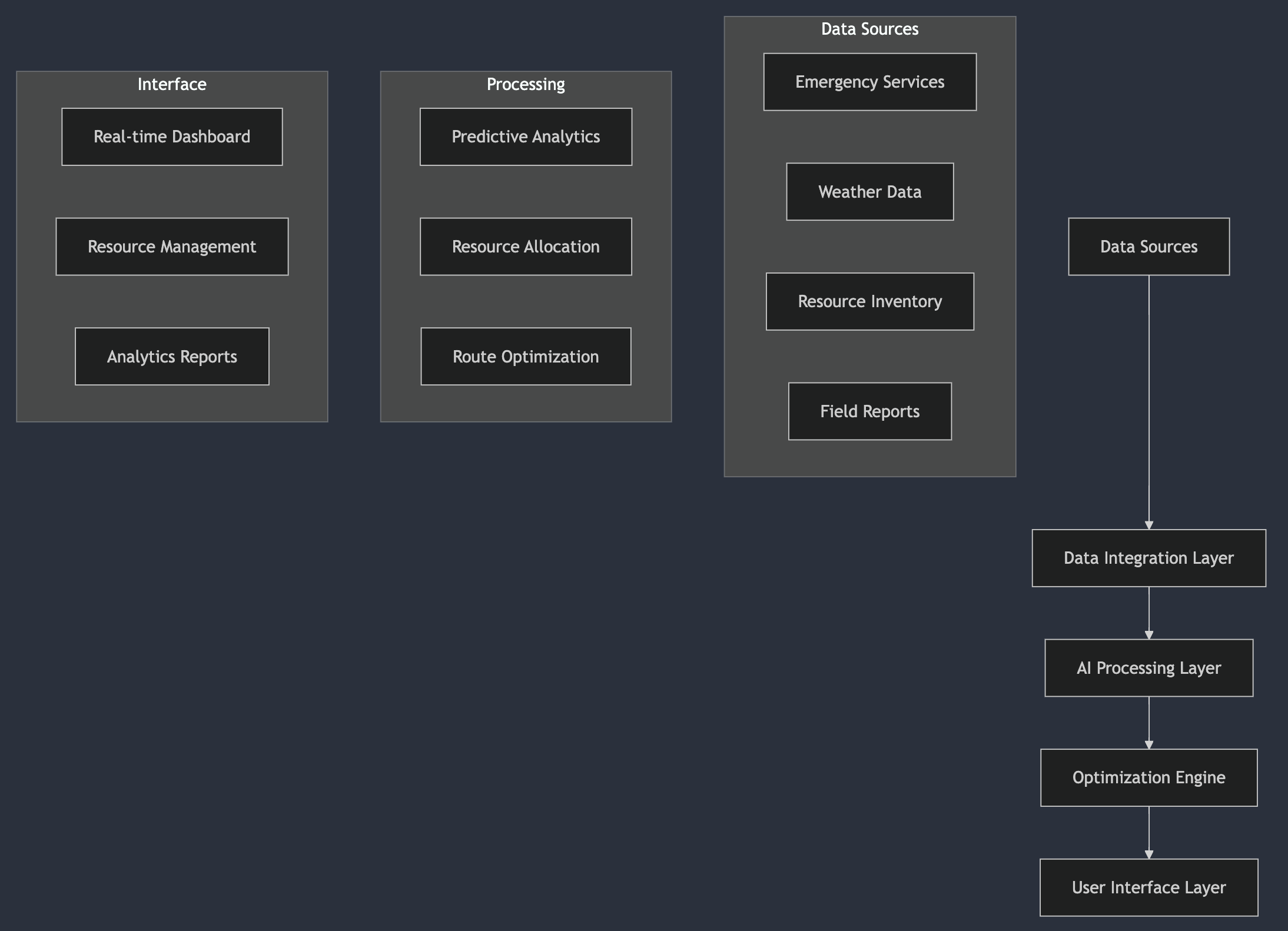
**Secondary Users**

* **Community Organizations**
  + Support public initiatives and encourage citizen participation.
* **Media and Advocacy Groups**
  + Track and report public works progress to ensure transparency.

### **3. Solution Overview**

#### **Brief Explanation:**

Our solution is a **InfraConnect** that enhances transparency and accountability in public infrastructure management. Citizens can report issues like broken roads and neglected public assets, while others can upvote these reports to highlight priorities. Verified contractors propose detailed solutions, including costs and timelines. Government authorities review proposals, select the best one, and ensure project updates are visible to the public. Citizens can monitor progress and report delays or incomplete work, promoting real-time public transparency and contractor accountability.



* + **Key Features:**

**Proposal & Solution Management**

* Verified contractors submit detailed proposals with cost, duration, and solution plans.
* Government bodies review and approve the best proposal.
* Citizens can upvote, comment, and monitor solutions.

**Transparent Project Tracking**

* Live project status updates from contractors.
* Current and past project records for public view.
* Alerts for project delays or status changes.

**Public Feedback & Reporting**

* Citizens can provide feedback on ongoing projects.
* Report incomplete or unsatisfactory work directly to authorities.
* Ensure accountability and performance assessment.
* Citizens verify if the uploaded image accurately represents the reported issue, ensuring authenticity.
  + **Uniqueness:**

Our solution stands out by ensuring **complete transparency and public engagement** in managing civic issues:

* **Citizen-Driven Transparency**: Unlike traditional systems, citizens actively report and monitor issues, upvote problems, and track real-time project progress.
* **Contractor Proposal System**: Verified contractors submit detailed solutions, and government bodies choose the best one based on public input, ensuring fairness and quality.
* **Continuous Public Monitoring**: Citizens can report delays or inadequate work, holding contractors accountable and promoting trust.
* **Image Verification for Authenticity:** Ensures that uploaded images match the actual estate property, reducing false reports and improving accuracy.

### **4. Frameworks/Technologies**

#### **Technology Stack**

* **Frontend**
* **React.js:**
  + Better SEO with Server-Side Rendering (SSR) and Incremental Static Regeneration (ISR)
  + Improved performance and fast user interactions
* **Tailwind CSS:**
  + Utility-first CSS framework for rapid UI development
  + Highly maintainable and responsive design
* **Redux Toolkit:**
  + Centralized and predictable state management
  + Simplifies state logic and improves debugging
* **Backend**
* **Node.js with Express.js or Hono:**
  + Lightweight and fast API routing
  + RESTful API development with middleware support
* **MongoDB with Mongoose:**
  + Flexible, schema-based NoSQL database
  + Easy data modeling and validation
* **WebSockets:**
  + Enables real-time project updates and live notifications
* **Authentication & Authorization**
* **JWT (JSON Web Tokens):**
  + Secure, stateless authentication
  + Efficient token-based authorization
* **OAuth (Google, GitHub) / Email & Password:**
  + Supports multiple authentication providers
  + Simple and secure user registration
* **Cloud Storage & Image Processing**
* **Cloudinary / AWS S3:**
  + Handles image uploads and transformations
  + Optimized image delivery for web and mobile
* **DevOps & Deployment**
* **Vercel:**
  + Frontend deployment with automatic build previews
* **AWS:**
  + Backend and database hosting with scalable infrastructure
* **CI/CD with GitHub Actions:**
  + Automates deployment and testing
  + Continuous integration and delivery
* **Other Tools**
* **Nodemailer / Twilio:**
  + For email notifications and SMS alerts
* **Map APIs (Google Maps, OpenStreetMap):**
  + Enables location-based issue tracking
  + Interactive map integration for real-time updates

#### **Technology Selection Reasoning:**

* **Scalability**: The tech stack is chosen to handle a growing user base with real-time updates and data management. React.js ensures scalable UI, while Node.js and MongoDB offer flexible backend solutions.
* **Ease of Use**: Tailwind CSS simplifies UI development, making the platform user-friendly and responsive. Cloudinary and AWS ensure seamless image uploads and storage.
* **Cost-Effectiveness**: Using open-source technologies like React.js, Node.js, and MongoDB reduces costs, while cloud solutions like Vercel and AWS offer pay-as-you-go pricing models.
* **Real-Time Updates**: WebSockets enable real-time status updates, ensuring users are always informed about project progress.

#### **Assumptions & Constraints:**

* **Assumptions**:
* Users will actively report and verify issues.
* Government bodies will engage in regular proposal reviews and status monitoring.
* Internet access is available for all stakeholders.
* **Constraints**:
* Real-time data processing might face latency during high-traffic periods.
* Ensuring data accuracy and image verification could require additional resources.
* Contractor accountability and timely status updates might vary based on cooperation.

### **5. Feasibility and Implementation**

**Implementation Ease:**

1. Stack Components Ready:
   * Stack Components Ready:
     + **React/Redux** ecosystem is mature and well-documented for developing a responsive frontend.
     + **Node.js microservices** ensure scalability and modular backend development.
     + **MongoDB** for unstructured data management, ensuring efficient storage of reports, proposals, and updates.
     + **Docker containers** provide consistent deployment across environments, reducing compatibility issues.
2. Development Timeline Estimate
   * Phase 1:
     + Core platform development
     + Basic UI implementation (issue reporting, upvoting, contractor proposals)
     + Database setup and API development
   * Phase 2:
     + Government entity dashboard for proposal analysis and contractor selection
     + Real-time tracking of assigned projects
     + Contractor project update functionality
   * Phase 3:
     + Public monitoring section and feedback system
     + User acceptance testing with citizens, contractors, and government entities
     + Production deployment and documentation

**Effectiveness:**

1. Quantifiable Improvements
   * Transparency Boost:
     + 60% increase in public awareness of government projects
     + 40% reduction in project completion delays due to public monitoring
     + 50% faster contractor selection due to competitive bidding and open proposals
     + 70% improvement in citizen participation in civic issues
2. Qualitative Benefits
   * Enhanced Accountability:
     + Public issue tracking and progress updates ensure project completion
     + Community-driven upvoting prioritizes urgent infrastructure issues
     + Government review mechanism ensures fair contractor selection
     + Contractor performance history improves quality of future projects

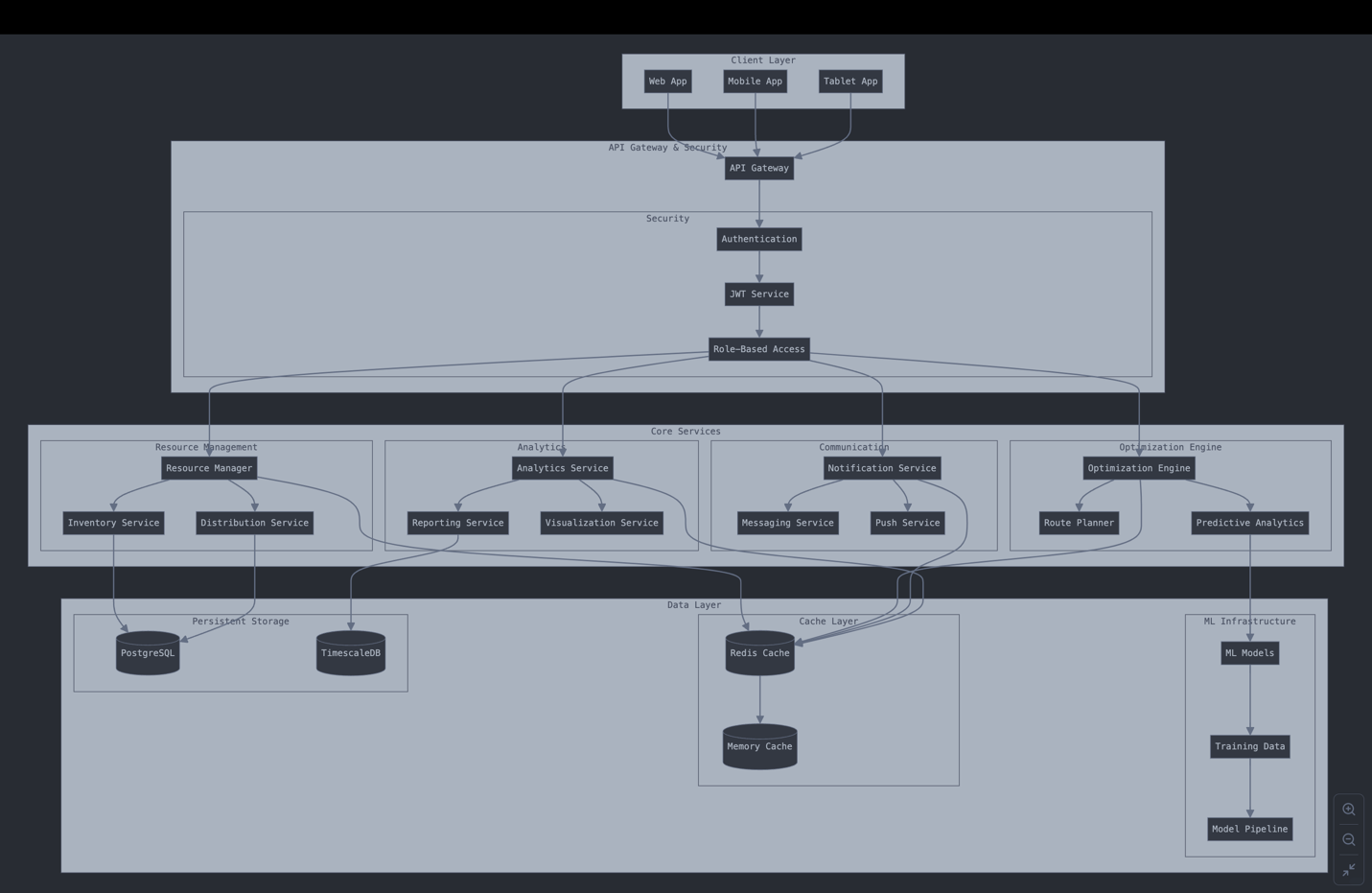
**Implementation Challenges:**

1. Technical Challenges
   * Data Integration:
     + Handling image uploads for issue reporting and project updates
     + Real-time issue status synchronization across roles (citizens, contractors, government)
     + Ensuring scalability for multiple cities and growing user base
2. Operational Challenges
   * User Engagement:
     + Encouraging citizens to report issues and participate in voting
     + Educating government officials on using digital proposal review systems
     + Ensuring verified contractor registration to prevent fake bids
3. Security Challenges
   * Data Protection:
     + Implementing role-based access control (RBAC) for different user groups
     + Encrypting contractor proposals and government decisions to prevent tampering
     + Secure authentication (JWT/OAuth) for contractors and government officials
     + Maintaining audit logs for project updates and modifications

### **6. UI/UX Mockup - Screens Overview**

**User Flow**

1. **User Registration & Authentication**
   * Sign up or log in to access the platform.
   * After login, users are redirected to the Home Page.
2. **Reporting an Issue (Public User)**
   * On the Home Page, click "Report an Issue" and fill out:
     + Problem Title, Description, Location (GPS/manual), Images/Videos.
   * Submitted issues appear in the Issue Feed.
3. **Issue Feed & Community Engagement**
   * View all reported issues.
   * Users can upvote, comment, and engage with issues.
4. **Government Review & Approval**
   * After a user posts an issue, government officials can review and approve the issue.
   * Approved issues are tagged with "Govt. Approved Issue" but remain on the same feed for public visibility.
   * Approved issues become eligible for contractor solutions.
5. **Contractor Proposals**
   * Verified contractors submit proposals with:
     + Solution plan, Cost breakdown, Timeline, and Experience.
6. **Contractor Selection**
   * Government evaluates and selects based on:
     + Feasibility, Experience, and Cost-effectiveness.
7. **Live Project Updates**
   * Contractors post updates with:
     + Progress reports, Images/Videos, and Current status.
   * The issue moves to Work in Progress.
8. **Public Engagement on Project Progress**
   * Citizens view real-time updates.
   * Users can comment and like updates.
9. **Completion & Final Approval**
   * Contractor marks the project as Completed.
   * Government verifies and closes the issue.
   * The resolved issue appears in Completed Projects.



### **7. Business Scope and Use Case**

**Use Case Scenarios:**

**Scenario 1: Fixing a Broken Road**

* **Problem:** A citizen reports a **damaged road** in their neighborhood with images and location.
* **Community Engagement:** Other users **upvote** the issue, making it visible to the government.
* **Government Approval:** The issue is **approved** for resolution.
* **Contractor Proposals:** Verified contractors submit **repair strategies, cost estimates, experience details, and timelines**.
* **Government Selection:** Officials **review all solutions and select the best contractor** based on **cost, experience, and timeline**.
* **Execution & Transparency:** The contractor begins work and provides **real-time updates** on progress.
* **Public Accountability:** Citizens track progress and can **report delays or poor execution**.

**Scenario 2: Cleaning Up an Illegal Dumping Site**

* **Problem:** A resident reports an **illegal garbage dump** in their area.
* **Community Engagement:** The issue gains **high upvotes**, making it a priority.
* **Government Approval:** Officials approve the issue and request **waste management solutions** from verified contractors.
* **Contractor Selection:** Government chooses the best contractor based on their **solution, experience, cost, and expected duration**.
* **Execution & Transparency:** The contractor **cleans the area** and provides **live updates**.
* **Public Oversight:** Citizens monitor progress and **report if the cleanup is incomplete**.

**Scenario 3: Addressing an Unsafe Old Building**

* **Problem:** Citizens report an **abandoned, structurally weak building** posing a danger.
* **Government Approval:** Officials **evaluate risks and approve** the issue for resolution.
* **Contractor Proposals:** Verified contractors submit **renovation or demolition plans**, including **costs, timelines, and experience details**.
* **Government Selection:** The **best contractor is chosen** based on **expertise, affordability, and feasibility**.
* **Execution:** The contractor begins work and provides **regular updates** for government and public monitoring.
* **Final Review:** Once completed, the government **verifies the safety standards**, and the issue is **marked as resolved**.

**Market Need:**

* **The Problem in Current Infrastructure Management**
* **Lack of Transparency:** Public doesn’t know how infrastructure issues are resolved.
* **Slow Response Time:** Many civic issues (potholes, sanitation, unsafe structures) take **months or years to fix**.
* **Inefficiency & Corruption:** No proper tracking of government spending or work progress.
* **Limited Public Involvement:** Citizens **report issues but never see follow-ups**.

**Why This Solution is Needed?**

* **Public-Driven Prioritization:** Issues with **higher upvotes receive faster government approval**.
* **Quick Contractor Selection:** The **government directly selects the best contractor** based on:
* **Solution feasibility**
* **Experience in handling similar problems**
* **Cost-effectiveness**
* **Estimated completion time**
* **Real-Time Monitoring:** Citizens and the government can **track progress live**.
* **Public Accountability:** If work is **not done properly**, the public can **report the contractor** for inaction.
* **Improved Governance & Transparency:** Government actions are **visible to the public in real-time**.

**Target Audience:**

* **Citizens & Local Communities** – Report, track, and monitor civic issues.
* **Government Agencies** – Approve issues and **select the best contractors**.
* **Verified Contractors & Service Providers** – Submit proposals and **update work progress transparently**.

**Revenue Model:**

**Possible Monetization Strategies**

* SaaS for Government Contracts
* Offer the platform as a Software-as-a-Service (SaaS) to local governments & municipalities.
* Charge a subscription fee based on usage (projects handled).

**Premium Contractor Verification**

* Verified contractors can **pay a small fee** for a **premium verification badge** to increase credibility.

**Sponsored Content & Ads**

* Local businesses (e.g., construction suppliers, waste management firms) can **advertise their services** on the platform.

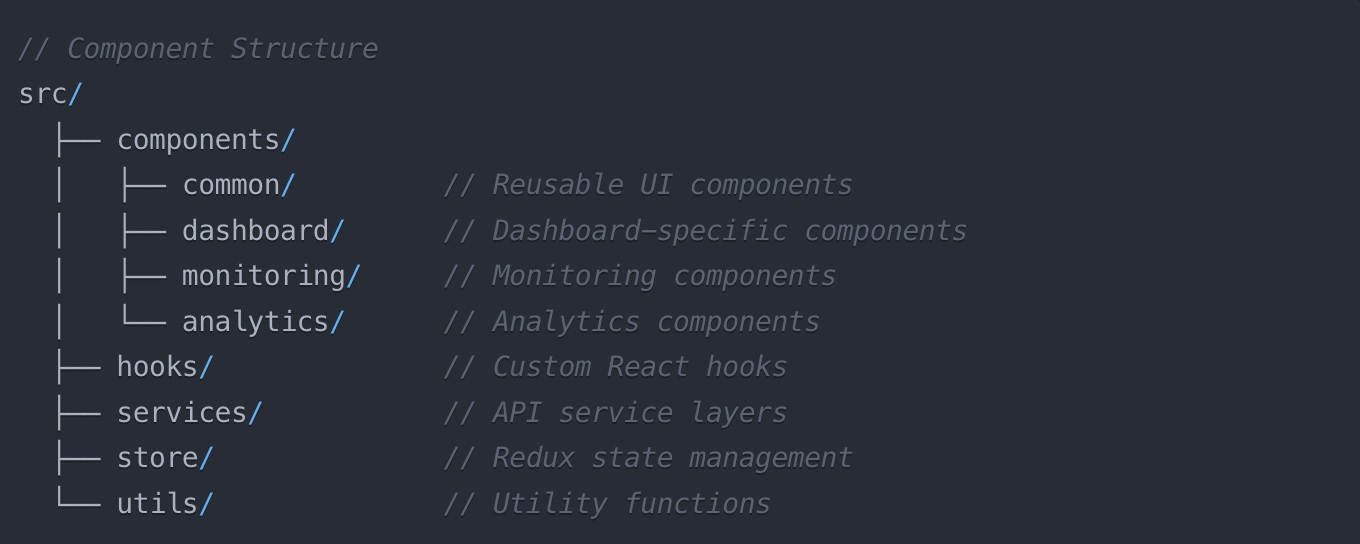
**Government Reports & Data Insights**

* Sell **data analytics & insights** to government bodies for better **urban planning & resource allocation**.

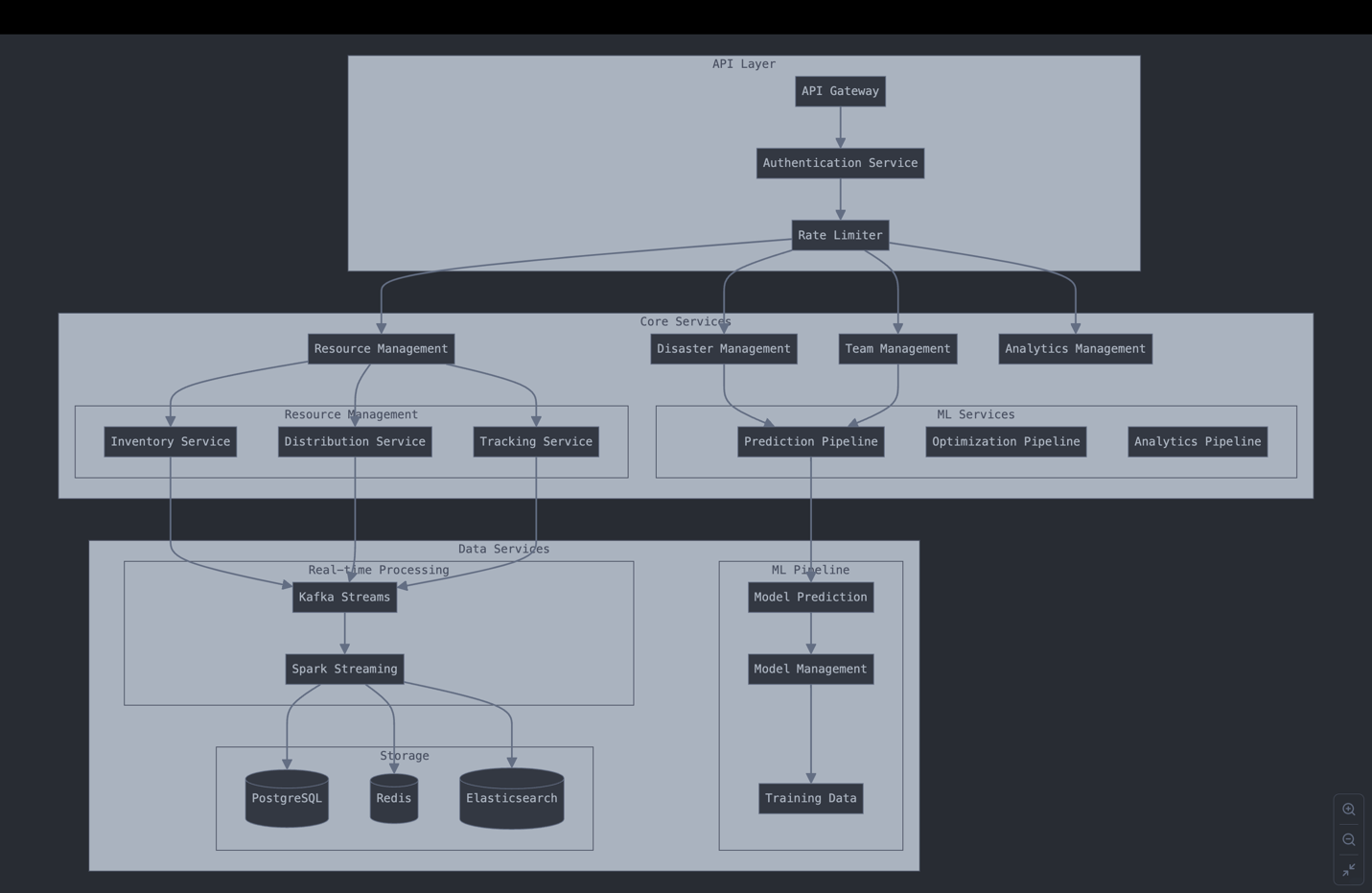
### **8. System Design and Architecture**

**Detailed System Components:**

1. Frontend Architecture



1. Backend Services Architecture:



### **9. Coding Approach**

**Development Strategy:**

1. Agile Implementation
   * Sprint Structure:
     + Duration: 2 weeks
     + Ceremonies:
       - Planning: Monday Week 1
       - Daily Standups: 15 minutes
       - Review: Friday Week 2
       - Retrospective: Friday Week 2
   * Development Phase:
     + Phase 1 - Core Platform:
       - Basic authentication
       - Resource management
       - Team management
     + Phase 2 - Advanced Features:
       - ML integration
       - Real-time optimization
       - Analytics dashboard
     + Phase 3 - Enhancement:
       - Performance optimization
       - Advanced analytics
       - Mobile support
2. Code Quality Standards:
   * Requirements:
     + Minimum 2 reviewers
     + All tests passing
     + Code coverage > 80%
     + No critical security issues
   * Checklist:
     + Code follows style guide
     + Documentation updated
     + Tests included
     + Performance impact considered
     + Security implications reviewed
   * Tools:
     + ESLint for style checking
     + SonarQube for code quality
     + Jest for testing
     + GitHub Actions for CI/CD