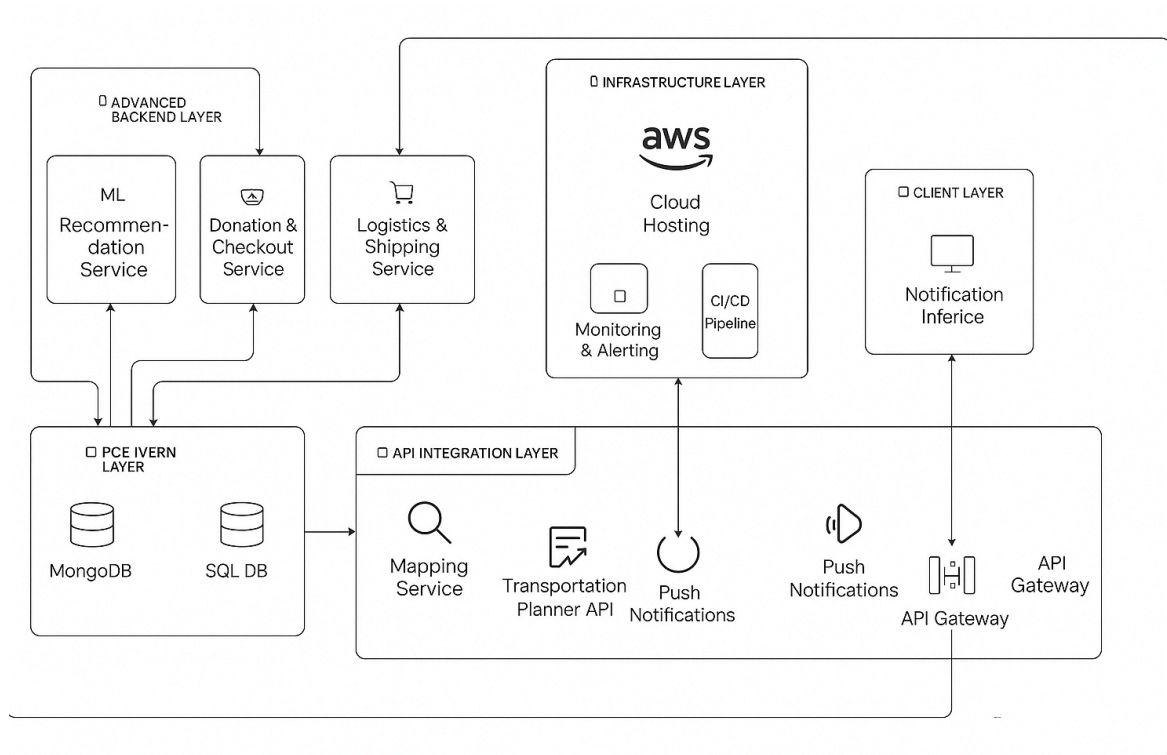


Architecture Document

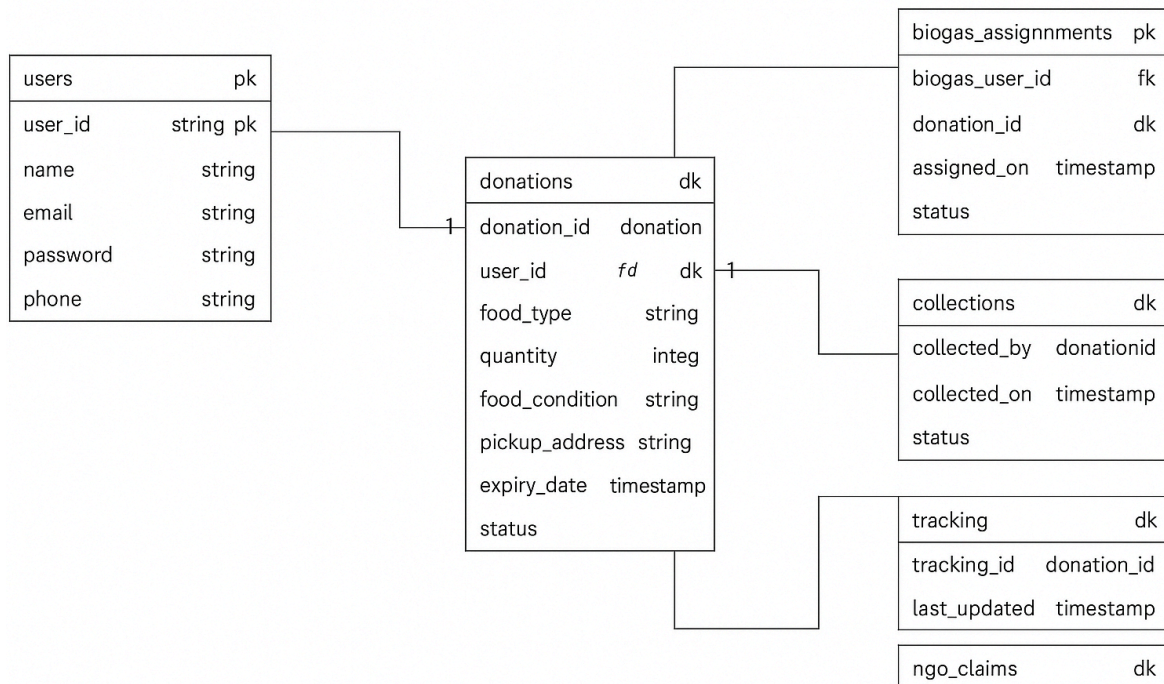
Event-Driven Architecture

Reason	Explanation
Loose Coupling	Services like Donor, NGO, Driver, and Biogas Plant interact indirectly through events. They don't need to know about each other.
Real-Time Notifications	When a donation is submitted, an event is triggered → NGOs notified → they can act fast.
Scalability	Easily scales. As donations grow, the event system (like Kafka, Firebase triggers, etc.) handles load without tightly linking services.
Flexibility	You can later add new consumers like analytics, SMS alerts, or fraud detection, without changing existing services.
Fault Tolerance	If an NGO doesn't act, no issue — the event is still stored, and fallback (biogas) happens automatically.
Reusability & Extensibility	You can plug in new services like “expiry detection,” “AI food classifier,” or “rewards for donors” later with minimal changes.

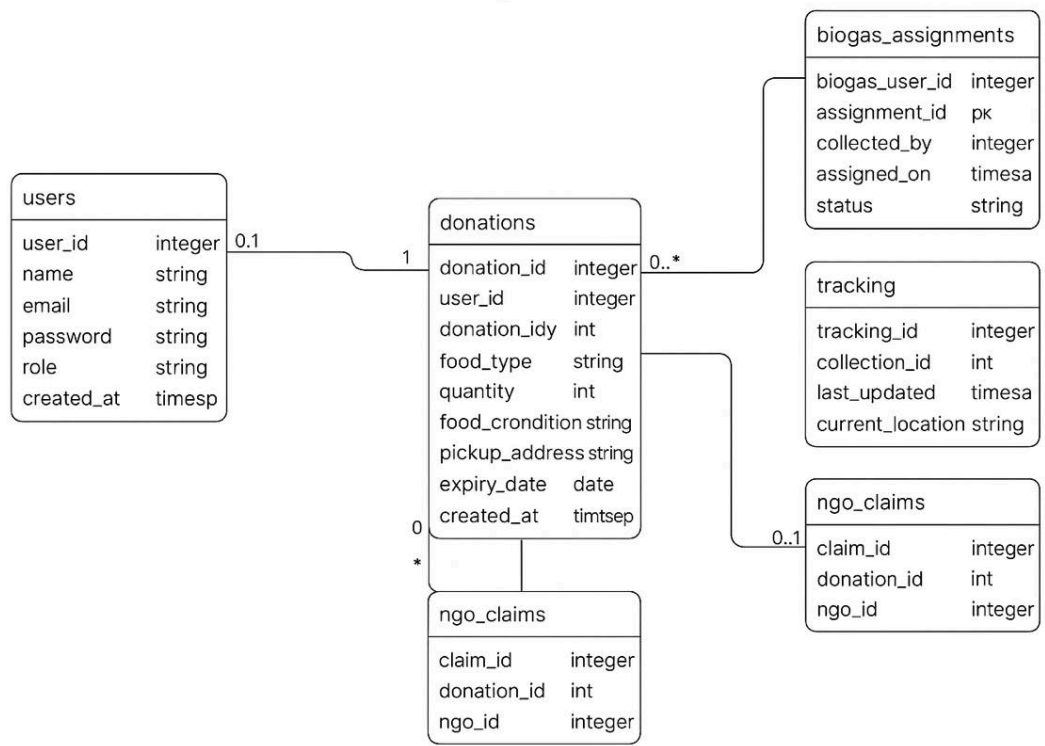
Architecture Diagram



ER Diagram



Schema Diagram



Data Flow Diagram

