Case Study

"Bharat-Krishi Connect": National Precision Agriculture & Unified Data Platform

1. Detailed Business Scenario

The Organization: The **National Agricultural Research & Development Board (NARDB)**.

The Mission: To modernize India's agricultural backbone by launching "Bharat-Krishi Connect." This platform aims to provide **Real-time Crop Advisory** and **Automated Resource Management** (*water/pesticides*) to **100 million farmers across the country**.

The Mission: Build a unified, multi-tenant platform that serves **140 million landholdings**.

The Phased Roadmap:

The platform is designed to scale horizontally as more states and landholdings are onboarded.

- **Phase 1 (Pilot):** **Implementation** across **4 Southern States** (***Tamil Nadu, Kerala, Karnataka, and Telangana***).
  - **Target:** **25 million Farmers** and **30 million Landholdings**.
  - **Focus:** Managing **diverse topography** from **coastal belts** to the **Deccan Plateau**.
- **Phase 2 (Scale):** Expansion to **100 Million Farmers** (*reaching approximately* ***50% of the national footprint***).
- **Phase 3 (Full Rollout):** Final **unified nation**.

The **platform** must move from **"Manual Monitoring"** to **"Autonomous Farming".**

Business Goals & Key Results (KRs):

- **Yield Increase:** **Improve national crop productivity** by **18%** through AI-driven **sowing recommendations**.
- **Water Conservation:** **Reduce** agricultural **water consumption** by **25%** using **automated** IoT-gated **irrigation**. **Success** requires a **<500ms end-to-end "Sensor-to-Valve"** control loop.
  - ***Core Mechanism:*** **Rain-Preemption Logic.** The **system** must **automatically issue** a **"SKIP_IRRIGATION" command** to all relevant **farm gateways** if the **localized IMD** (*India Meteorological Department*) **rain probability exceeds 70%.**

- **Market Access:** Provide **real-time "Mandi" price** transparency to farmers within **2 seconds** of **price updates**.
- **Disaster Mitigation:** **Reduce crop loss** from **pests/weather** by **sending alerts** to **2.5 million+ concurrent users** within **60 seconds** of an event detection.

The Technical Challenge: Extreme Scale & Resilience

- **Astronomical Scale:** **Managing 150+ million hectares** requires a system that can **ingest 2 million+ messages per second** at **peak** national rollout. The **storage** layer must **handle petabytes** of **time-series data** while remaining searchable for historical yield trends.
- **Rural Connectivity Gaps:** **Despite 5G growth**, the **last mile** in **rural India** remains **unstable**. The **architecture** must **solve** the **Intermittent Connectivity**.
- **Real-Time Performance:** **"Actionable Insights"** (*e.g., detecting a locust swarm or a pump failure*) **lose value** if **delayed**.
- **Data Integrity (The RPO Challenge):** The system must survive regional disasters without losing critical "State" data. An **RPO of < 5 seconds** is **mandatory** to ensure that the **"Last Known State" of 30M+** valves is **consistent** across **primary** and **backup sites**.

2. Operational & Technical Constraints

A. Data Residency (The Legal Constraint)

- **Sovereign Boundary:** All **data** must be **hosted** on **servers** physically located **within India**. No data or metadata can be processed in overseas regions (*e.g., US-East or EU-West*).
- **Placement Flexibility: Data can** be **moved freely** between any Indian data centers (*e.g., Mumbai, Chennai, Delhi, or Hyderabad*) for load balancing or disaster recovery.

B. High Availability & Disaster Recovery

- **99.99% Uptime: Agriculture** is **time-sensitive**. A **system outage** during the **sowing season** could lead to **national food shortages**.
- **15-Minute RTO (Recovery Time Objective): Maximum time allowed** to **switch operations** from the Primary Hub to the DR site during a total outage.
- **< 5-Second RPO (Recovery Point Objective):** The system must guarantee that **no more than 5 seconds of sensor data or command logs are lost** during a failover. This prevents "Ghost Watering" (double-irrigation) caused by data gaps during a site switch.

- **Active-Active/Passive:** The system **must survive** a **total outage** of a **primary region** (*e.g., Mumbai*) by **failing over** to a **secondary region** (*e.g., Hyderabad*) within **15 minutes**.

C. Technical Friction Points

- **Legacy Integration:** The **platform** must **pull data** from **existing state-level weather stations** that use older SOAP APIs.
- **Peak Loads: During monsoon** or **harvest periods**, **traffic spikes 20x compared** to the **off-season**.

3. The Assignment: 12 Detailed Tasks

Students must act as the Lead Solution Architect to complete the following:

Section 1: Strategic Mapping

- **Assignment 1: Business Vision to Technical Vision.**
- **Assignment 2: Functional & Non-Functional Requirements.**

Section 2: Architectural Selection

- **Assignment 3: Select Paradigm.**
- **Assignment 4: Select Model.**
- **Assignment 5: Select Architecture Style.**
- **Assignment 6: Select Architecture Pattern.**

Section 3: Technical Design & Flow

- **Assignment 7: High-Level Design (HLD).**
- **Assignment 8: Low-Level Design (LLD).**
- **Assignment 9: Component & Service Selection.**
- **Assignment 10: Create 3 ADRs (Architectural Decision Records).**

Section 4: Visualizing the Flow

- **Assignment 11: Create System Flow.**
- **Assignment 12: Final Architecture Picture.**

Model Answer

1. Mapping Business Goals to Technical Pillars

This section translates the high-level KR (*Key Results*) into structural pillars that will guide the design of the **Bharat-Krishi Connect** platform.

Business Goal	Technical Pillar	Success Metrics & KPIs
National Productivity Growth	**Unified Analytical Intelligence**	• **Data Consolidation: Integrate telemetry** from **30M landholdings** (*Phase 1*) into a **standardized schema** for **cross-regional ML modeling**. • **Yield Optimization:** Deliver precise sowing windows to **achieve** an **18% productivity increase** by synthesizing soil, drone, and historical datasets.
National Water Stewardship	**Closed-Loop Autonomous Control**	• **Precision Actuation**: **Maintain < 500ms end-to-end latency** for **sensor-to-valve triggers**. • **Conservation Target**: **Drive a 25% reduction** in **water usage** by transitioning from manual/scheduled timers to real-time, sensor-driven automation.
Farmer Market Empowerment	**Real-Time Information Symmetry**	• **Low-Latency Propagation**: Utilize high-speed messaging patterns to **broadcast** national Mandi **price changes** in **< 2 seconds**.
Emergency Risk Mitigation	**High-Velocity Elastic Messaging**	• **Concurrent Notification:** Scale infrastructure to **broadcast critical pest** or **weather alerts** to **2.5M+ concurrent users** within **60 seconds** of detection.
National Mission Criticality	**Centralized Regional Resiliency**	• **Continuous Availability: Maintain 99.99% system uptime** through **primary hub** and **local DR** pairing (*e.g., Chennai → Madurai*). • **Disaster Recovery: Commit** to a **15-minute RTO** (*Recovery Time*) and **< 5s RPO** (*Data Loss window*) during regional outages.
Rural Infrastructure Resilience	**Decentralized Edge Autonomy**	• **Offline Continuity:** Empower farm gateways to maintain **100% irrigation functionality** for up to **48 hours** during a **total cloud-link blackout**.
Massive National Scale	**Hyper-Scale Horizontal Elasticity**	• **Linear Scalability**: **Architect** a **system** capable of onboarding **25M to 140M landholdings** and millions of IoT devices without latency degradation or data loss.

2. Functional (FR) & Non-Functional Requirements (NFR)

I. Functional Requirements (FR)

Functional requirements define **what** the **system** must **do** to support the **140 million farmers** and the **water conservation mission**.

ID	Requirement Category	Detailed Description
FR-1	**Autonomous Irrigation**	The **system must trigger "Open/Close" commands** to **field valves based** on **soil moisture** thresholds and **crop-specific hydration plans**.
FR-2	**Predictive Preemption**	The **system** must **ingest weather forecast data** and **automatically abort scheduled irrigation** if **rain probability exceeds 70%** in **a 6-hour window**.
FR-3	**Market Transparency**	The **platform** must **push "Mandi" price** updates to the **mobile app** for **crops registered** in the **farmer's** profile.
FR-4	**Legacy Integration**	The **system must pull data** from **state-level stations** via **SOAP/XML** and normalize it into a **unified national schema**.
FR-5	**Emergency Alerting**	The **system** must **detect pest outbreaks** via **drone imagery analysis** and **broadcast** regional **alerts** to all **farmers** in the **affected "Grid Cell".**

II. Non-Functional Requirements (NFR)

Non-functional requirements define **how** the system performs. These are the "Design Constraints" that ensure the **25% water saving** and **99.99% uptime**.

A. Performance & Scale (The "Krishi" Load)

- **NFR-1 (Latency):** The **end-to-end "Sensor-to-Valve**" control loop (*Actuation Latency*) must be **less than 500ms** to prevent over-watering or pipe bursts.
    - **The "Latency Budget" Breakdown**
        - To **achieve** the **< 500ms** target, the technical requirements for sub-components are as follows:

Component	Time	Technical Reason (The "Why")
Ingestion (Up)	**150ms**	**Signal Travel:** Covers the time for a sensor in a rural field to reach the nearest cell tower and travel via fiber to the Regional Hub.
Logic (Brain)	**100ms**	**Decision Making:** The time for the server to check the "Rain Rule," verify the farmer's ID, and decide whether to turn the water on or off.
Downlink (Down)	**150ms**	**Return Trip:** The time for the "Stop" command to travel back from the Hub through the 4G/NB-IoT network to the farm gate.

Actuation (Physical)	**100ms**	**Mechanical Move:** The physical time a relay switch or solenoid valve takes to physically flip from an "Open" to "Closed" position.
Total	**500ms**	**The Safety Limit:** The maximum delay allowed to ensure precision irrigation and prevent pipe damage (Water Hammer).

- **NFR-2 (Throughput):** The ingestion layer must support a sustained load of **150,000 messages per second (MPS)** and an **elastic peak** of **500,000 MPS** during monsoon surges.
    - **The Capacity Calculation (The "How Many")**
        - The **150,000 MPS** is the "Sustained Baseline" **required** to **keep** the **data fresh** enough for the **logic engine** to make **smart decisions**.
        - **Population: 30 Million Landholdings** (Phase 1).
        - **Freshness Goal:** We need an **update** every **200 seconds** (3.3 mins). The **200-second interval** tells the Hub **"What is happening"** often enough, so that when a command is triggered, it can be executed in **< 500ms**.
        - **The Math: 30,000,000 Devices \ 200 Seconds = 150,000 Messages Per Second (MPS)**
- **NFR-3 (Price Propagation): Market price updates** must be **fanned out** to **2 million concurrent users** in **less than 2 seconds**.

B. Availability & Resilience

- **NFR-4 (Uptime):** The **system** must achieve **99.99% availability** (*max 52 minutes of downtime per year*).
- **NFR-5 (Disaster Recovery):** In the event of a **regional failure** (*e.g., Mumbai DC*), the system must **failover** to the **secondary region** (*e.g., Hyderabad*) within an **RTO of 15 minutes**.
- **NFR-6 (Data Integrity):** The **Recovery Point Objective (RPO)** for **water consumption logs** must be **less than 5 seconds** to ensure accurate national water accounting.

C. Sovereignty & Security

- **NFR-7 (Data Residency): All PII**, geospatial data, and moisture logs must be physically **stored** within the **Indian Sovereign Boundary**.
- **NFR-8 (Edge Autonomy):** The **local farm gateway** must **retain** enough **"State"** to perform **autonomous irrigation** for **48 hours** during a total cloud-connectivity blackout.

3. Architectural Paradigm Selection

For the "Bharat-Krishi Connect" platform, we select the **Functional Reactive Paradigm**, where **Reactive** as the **"Logistics"** (*how we move data*) and **Functional** as the **"Decision"** (*how we process data*).

Paradigm	Role in the Platform	Simple Reason (The "Why")	Impact on NFRs
Reactive	**The High-Speed Pipeline**	**Non-Stop Movement:** It ensures the server never "waits" It can handle **150,000 messages** at once without crashing or getting stuck.	**Solves Scale:** Handles the 5M MPS peak and 2M concurrent users.
Functional	**The Instant Decision**	**Stateless Math:** It treats the **"70% Rain Rule"** as a **fast math equation**. Since **it doesn't have** to "look up" or **"change" data**, it is **incredibly fast**.	**Solves Latency:** Keeps the logic step under the **100ms budget**.

The "Winning Combo" Effect

- **Reactive** handles the **Quantity:** It manages the **massive crowd of 30 million sensors** trying to talk at the same time.
- **Functional handles** the **Quality:** It ensures that every decision (*like "Skip Irrigation"*) is **calculated** with **100% accuracy** in a **fraction of a second**.

4. Select Model

a) Domain Model (Logical Entities)

This model defines the core objects required to manage **30 million landholdings** and **25 million farmers**.

Entity	Core Attributes	Relationship
Farmer	CitizenID, Name, Contact, Language Preference	Owns one or more **Landholdings**.
Landholding	LandUID, Geo-Polygon (GIS), Soil Type, Primary Crop	Host for one **Gateway** and multiple **Sensors**.
Gateway	DeviceID, Firmware Version, Connectivity Status	Acts as the communication bridge for a **Landholding**.
Telemetry	ReadingID,	Linked to a specific **Gateway** or **Sensor**.

	Timestamp, Moisture, Temperature, pH	
Commodity	Name (e.g., Rice), Variety, Grade	Subject of **Mandi Price** updates.
Mandi Hub	HubID, Location Name, District	Source of real-time price fluctuations.
Weather Grid	GridID, Boundary Coordinates, Rain Probability	Overlays across multiple **Landholdings**.

b) Process Model

i. Autonomous Actuation Loop

Target: < 500ms (Ensuring Precision Irrigation)

Step	Action	Logic / Requirement
Ingestion	Sensor Data Uplink	Farm Gateway pushes current moisture status to the Regional Hub.
Contextualize	Fetch Environment State	System identifies the **Weather Grid** associated with the Landholding.
Evaluate	Apply "Rain Rule"	Logic Engine checks: *Is Rain Prob < 70% AND Moisture < 20%?*
Command	Issue Downlink	If condition is met, Hub generates an "Open Valve" instruction.
Execution	Physical Relay Flip	Gateway receives command and triggers the mechanical water valve.

ii. Market Access (Price Transparency)

Target: < 2 Seconds (Market-to-Mobile Delivery)

Step	Action	Logic / Requirement
Capture	Price Ingestion	Central eMandi system receives a price update for a specific **Commodity**.
Standardize	Schema Normalization	Data is converted into a standard unit (e.g., Price per Quintal in INR).
Targeting	User Segmentation	System finds all **Farmers** interested in that Commodity or region.
Broadcast	Reactive Push	Real-time push transmission to the active mobile app sessions.

Notify	Device Presentation	The farmer's app displays the updated price and trend analysis.

iii. Disaster Mitigation (Emergency Alerting)

Target: < 60 Seconds (Event-to-Masses Notification)

Step	Action	Logic / Requirement
Detect	Anomaly Identification	Drone AI or Weather Sensor detects a disaster event (e.g., Pest Swarm).
Scope	Geospatial Mapping	System identifies all **Landholdings** within the danger polygon.
Prioritize	Emergency Routing	Alert enters a high-priority queue, bypassing all routine sensor traffic.
Fan-out	Mass Notification	Simultaneous broadcast to **2.5M+ concurrent users** in the affected zone.
Action	Mitigation Guidance	Notification includes localized steps to minimize crop damage.

5. Select Architectural Style

The **Architectural Style** defines the "macro" view of the system.

We use **Microservices** to divide the project into independent, autonomous units that communicate over a network.

Component	Responsibility	Phase 1 Real-World Example	Rationale for this Style
Control Service	Manages the **500ms** sensor-to-valve loop.	A sensor in a **Telangana** cotton farm detects dry soil and triggers a pump.	**Resilience:** If the price service crashes, the water system stays alive.
Market Service	Manages the **2-second** price fan-out.	Pushing a sudden "Turmeric" price surge to 15 million mobile apps.	**Elasticity:** We can scale this service to 100 nodes during harvest peaks.
Alert Service	Manages the **60-second** emergency broadcast.	Detecting a locust swarm in **North Karnataka** and alerting nearby farms.	**Priority:** Emergency traffic is handled separately from routine soil data.
Identity Service	Manages **Aadhaar** and Land GIS records.	Verifying that a specific farmer owns a specific plot before acting.	**Security:** Isolates sensitive PII (Sensitive Data) from high-velocity IoT traffic.

6. Select Architectural Pattern

a) Hexagonal (Ports & Adapters)

- **Best for**: The **Control Service (*Sensor-to-Valve logic*).**
- This **pattern** is **used** where **Latency (*< 500ms*)** and **Logic Protection** are the highest priorities.

Pattern Layer	Role in Bharat-Krishi	Real-World Example (Control Service)	Rationale
Domain (The Core)	The "Pure" Business Logic.	The code calculating the **"70% Rain Rule"** using only moisture and forecast variables.	**Speed:** Runs in RAM with zero external dependencies. Keeps the logic step under **100ms**.
The Ports	Contractual Interfaces.	A standard "socket" for receiving soil data or sending "Stop Pump" commands.	**Independence:** The core logic doesn't care if the sensor is 4G, 5G, or Satellite.
Input Adapters	Technology-specific entry.	An **MQTT Adapter** that translates raw sensor signals from the farm into a "Moisture Event."	**Flexibility:** Allows the same core logic to work with different hardware types across Phase 1.
Output Adapters	Technology-specific exit.	A **Redis Adapter** that fetches the latest weather status from a fast cache instead of a slow DB.	**Latency Shield:** Prevents a slow database from "blocking" the decision to turn off a valve.

b) CQRS (Command Query Responsibility Segregation)

- **Best for**: The **Market Service** (***Mandi Price updates***).
- This **pattern** is **used** where **Read Volume** (***15M users***) is **vastly different** from **Write Volume** (***Mandi updates***).

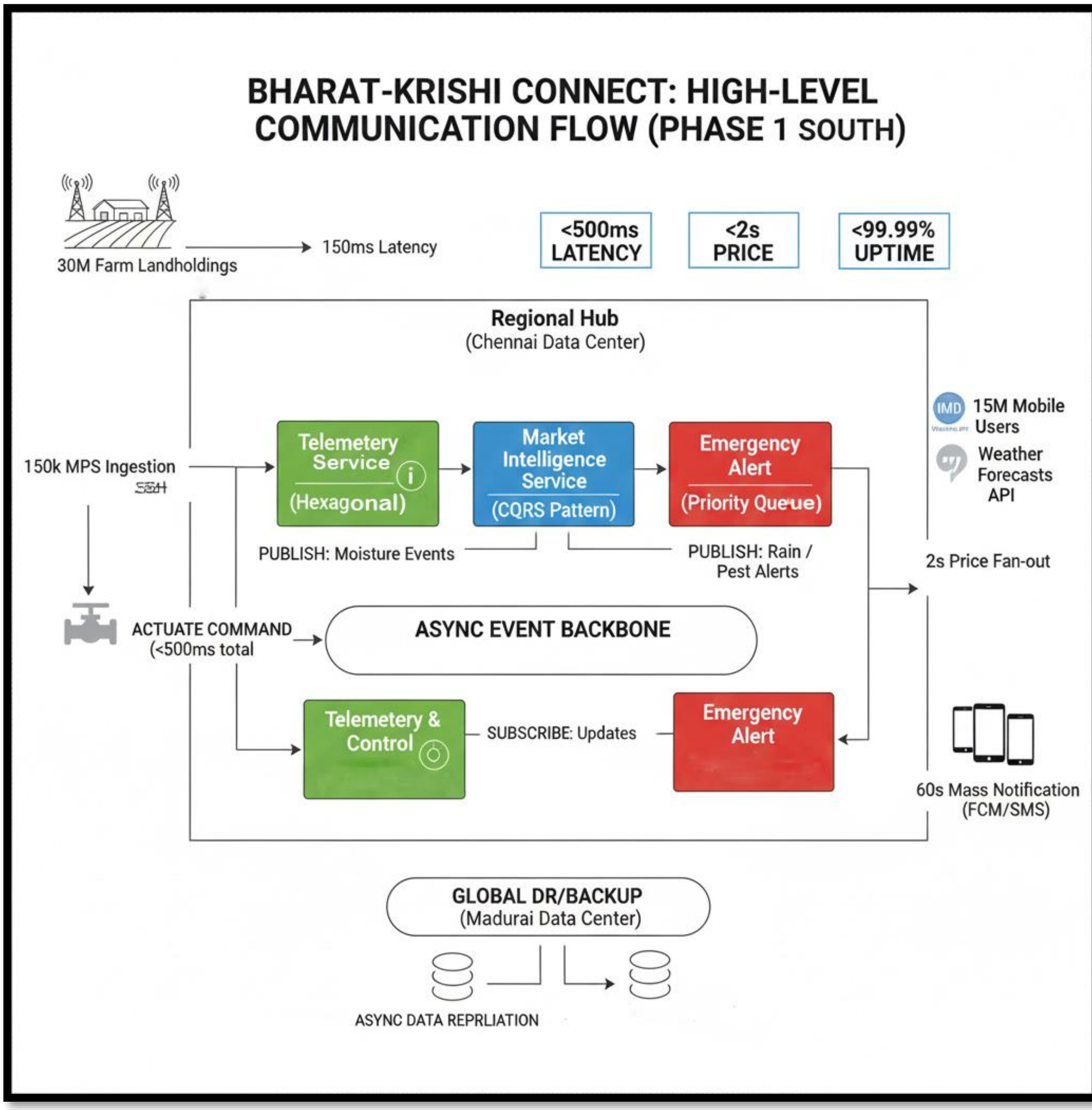
Pattern Layer	Role in Bharat-Krishi	Real-World Example (Market Service)	Rationale
Command Side (Write)	Handles data entry/updates.	Processing a price update for "Basmati Rice" from a specific Punjab Mandi hub.	**Integrity:** Optimized for secure, validated updates to the national price ledger.
Query Side (Read)	Handles data requests/display.	15 million farmers simultaneously checking the current price of Onions in their district.	**Massive Scale:** The "Read" side uses a high-speed replicated cache to prevent the system from crashing.
Event Bus	Synchronizes sides.	When a price is updated on the "Command" side, it triggers an event to update the "Query" cache.	**Performance:** Farmers read from a pre-calculated price list rather than a slow, complex database.

Projections	Pre-formatted data.	Pre-calculating a "Price Trend" (up/down) so the app displays it instantly without a search.	**NFR-3 Goal:** Ensures Mandi prices are visible to 15M users in **under 2 seconds**.

c) Summary: Which Pattern for Which Need?

Use Case	Recommended Pattern	Driving NFR
Irrigation & Water Control	**Hexagonal**	**Latency:** Must respond in < 500ms to prevent water waste.
Mandi Price Transparency	**CQRS**	**Price Propagation:** Must push 15M users in < 2 seconds.
Emergency Alerts	**Hexagonal + CQRS**	**Throughput:** High-speed detection (Hex) + Massive broadcast (CQRS).

7. Create HLD


BHARAT-KRISHI CONNECT: HIGH-LEVEL COMMUNICATION FLOW (PHASE 1 SOUTH)
30M Farm Landholdings
150ms Latency
<500ms LATENCY
<2s PRICE
<99.99% UPTIME
Regional Hub
(Chennai Data Center)
150k MPS Ingestion
Telemetery Service
(Hexagonal)
Market Intelligence Service
(CQRS Pattern)
Emergency Alert
(Priority Queue)
15M Mobile Users
Weather Forecasts API
PUBLISH: Moisture Events
PUBLISH: Rain / Pest Alerts
2s Price Fan-out
ACTUATE COMMAND
(<500ms total
ASYNC EVENT BACKBONE
Telemetery & Control
SUBSCRIBE: Updates
Emergency Alert
60s Mass Notification
(FCM/SMS)
GLOBAL DR/BACKUP
(Madurai Data Center)
ASYNC DATA REPRLIATION

8. Create LLD

a) The Sharding Strategy: District-Time Bucketing

- Since we are centralized in Chennai, we use **Internal Sharding**.
- We **divide** the **30 million landholdings** into **logical "Buckets" based** on their **District** and a **Time-Window**.
- This prevents "write-lock" during peak morning irrigation hours.

Component	Choice	Rationale
Database Engine	**Apache Cassandra**	Its "Masterless" architecture allows us to add nodes in the Chennai DC to scale linearly.
Partition Key	district_id + time_bucket	Distributes the 150k MPS across different physical "Racks" in the Chennai DC.
Clustering Key	recorded_at (DESC)	Ensures the **latest** moisture data is at the very top of the disk for instant retrieval.
Compaction	Time-Window Compaction	Optimized for "Delete-Old-Data" (TTL) once telemetry is no longer needed.

b) The Telemetry Table Schema

- To ensure the **100ms Logic Budget**, the **data must** be **stored** in a "**Query-First" format**.
- We do not use "JOINs" because they are too slow for real-time actuation.

SQL

```sql
-- LLD Table Schema for Telemetry Data
CREATE TABLE southern_pilot_telemetry (
    district_id TEXT,          -- Example: 'TN_COIMBATORE'
    time_bucket TEXT,          -- Example: '2025-12-15-10AM'
    land_id UUID,              -- Specific Farm ID
    recorded_at TIMESTAMP,     -- Exact time of sensor reading
    moisture_level FLOAT,
    rule_status TEXT,          -- Last decision made by the Hexagonal Core
    PRIMARY KEY ((district_id, time_bucket), recorded_at, land_id)
) WITH CLUSTERING ORDER BY (recorded_at DESC);
```

c) Why 150,000 MPS Needs This Sharding

Even with all servers in Chennai, the **Network Interface Cards (NICs)** and **Disk I/O** have limits.

i. **Shard Distribution:** By using time_bucket, we ensure that if 5 million sensors in Andhra Pradesh wake up at 6:00 AM, their data is spread across **different servers** in the DC rather than hitting one single machine.

ii. **Read-Optimized:** Because we sort by recorded_at DESC, the **Control Service** can fetch the most recent data in **< 2ms**, allowing plenty of time for the 150ms downlink back to the farm.

d) Summary Table for LLD

Design Concern	Technical Solution	Benefit for Bharat-Krishi
Write Throughput	NoSQL (LSM Tree storage)	Handles **150,000 writes** per second without locking tables.
Data Locality	Geo-Partitioning	Chennai farmers get **50ms faster response** because their data is physically close.
Query Speed	Time-Series Indexing	Allows the "Rain Rule" logic to fetch the last 5 readings in **< 2ms**.
High Availability	3-way Replication	If a data center in Chennai fails, the Madurai shard takes over in **< 1s**.

9. Component & Service Selection

Bharat-Krishi Master Architecture: Components & Product Selection

Layer	Core Service	Product	Key Feature
Edge Ingestion	Device Gateway	**EMQX**	Handles **1M+ concurrent MQTT v5 connections** per node; rule-engine for preprocessing.
Network Layer	5G Slicing	**Airtel / Jio 5G Private Slicing**	Dedicated **URLLC slice** (*Ultra-Reliable Low Latency*) for real-time drone video telemetry.
Event Backbone	Message Bus	**Confluent Kafka**	The "Nervous System"; handles **150k MPS** with Schema Registry for data consistency.

Fast-Path Storage	In-Memory Cache	**Redis**	**Active-Active Replication**; sub-millisecond lookups for the "Rain-Preemption" logic.
Massive Telemetry	Time-Series DB	**Apache Cassandra**	**NoSQL**; sharded by district to store 30M farm histories without downtime.
Market Push	Pub/Sub Broker	**Centrifugo**	Managed fan-out for the **15M user broadcast**; ensures < 2s price propagation.
Identity / Master	Land Records DB	**PostgreSQL**	Postgres for **Aadhaar-linked PII** with 100% ACID compliance.

Operational Logic: The "Product" Workflow

The integration of these products ensures that the **Hexagonal** and **CQRS** patterns are ready for production deployment.

1. **The Drone Detects Stress:** A **Marut Drone** using 5G sends a gRPC alert through the **EMQX Gateway**.
2. **The Backbone Responds: Kafka** triggers the **Emergency Service**.
3. **The Farmer is Notified: Centrifugo** pushes a "High Disease Risk" notification to the mobile app in under 60 seconds.
4. **The Soil Sensor Checks In:** An **IoT sensor** sends moisture data via **5G**. **Redis** provides the latest weather status to the **Go-based Hexagonal Core** in < 1ms to decide if irrigation should stop.

10.Create ADRs

ADR ID	Decision (Product Selection)	Architectural Context & Need	Key Rationale (The "Why")	Consequences & Trade-offs
001	**EMQX Enterprise**	**Edge Ingestion:** Handling 30M concurrent MQTT connections from NB-IoT/5G sensors.	Proven scale (100M+ connections); built-in **SQL Rule Engine** to filter noise at the edge.	Requires professional clustering and high-availability (HA) proxy setup.
002	**Confluent Platform**	**Event Backbone:** Buffering 150,000 MPS between the edge and the core logic.	**Schema Registry** ensures data contracts; MirrorMaker 2 handles DR replication to Madurai.	Significant operational overhead; requires dedicated infrastructure management.

003	**Redis Enterprise**	**Fast-Path Storage:** Powering the <500ms real-time irrigation actuation loop.	**Sub-millisecond latency** for "Hot" data like current valve state and weather pre-emption.	Data is volatile by nature; must be synced to Cassandra for permanent historical records.
004	**DataStax Enterprise**	**Massive Telemetry:** Storing years of history for 30M sensors (billions of rows).	**Wide-Column NoSQL** handles high write volume linearly; no single point of failure.	Complex data modeling; query patterns must be defined upfront (no ad-hoc joins).
005	**Centrifugo**	**Real-time Broadcaster:** Pushing price/alerts to 15M farmer apps simultaneously.	Self-hosted **Pub/Sub Broker** optimized for massive fan-out; manages persistent WebSockets.	Dependent on a stable Redis backplane to synchronize across multi-node clusters.
006	**EDB PostgreSQL**	**Identity/PII Vault:** Securing Aadhaar, PII, and sensitive land-ownership deeds.	**ACID Compliance** and Row-Level Security (RLS) ensure 100% data integrity for legal records.	Does not scale horizontally like NoSQL; reserved strictly for critical relational data.

11.Create System Flow

a) Flow Table 1: Autonomous Irrigation (The 500ms Loop)

Goal: *Precision water management using real-time soil and weather data.*

Sequence	Layer	Product	Action / Logic
1	**Sensing**	**Soil Sensor**	Detects Moisture () & Temperature.
2	**Ingestion**	**EMQX**	Receives MQTT packet and validates device certificate.
3	**Backbone**	**Kafka**	Buffers message in telemetry.moisture.south topic.
4	**Processing**	**Hexagonal Core**	**Logic:** Check **Redis** for "Rain-Preemption" flag.
5	**Decision**	**Hexagonal Core**	If Moisture Low + No Rain Predicted **Trigger Actuation**.
6	**Egress**	**Centrifugo**	Pushes "VALVE_OPEN" command to the field gateway.
7	**Actuation**	**Solenoid Valve**	Physical valve opens; starts irrigation.
8	**Storage**	**Cassandra**	Stores "Actuation Event" for long-term audit trail.

b) Flow Table 2: Mandi Price Distribution (The 2s Broadcast)

Goal: *Pushing localized market intelligence to 15 million farmers simultaneously.*

Sequence	Layer	Product	Action / Logic
1	**External**	**Mandi API**	Government prices updated for Basmati Rice/Cotton.
2	**Ingestion**	**Market Svc**	Fetches prices; stores metadata in **MongoDB**.
3	**Event**	**Kafka**	Publishes market.price.updated event.
4	**Mapping**	**PostgreSQL**	Queries PII Vault to find farmers subscribed to this crop.
5	**Broadcast**	**Centrifugo**	**Fan-out:** Sends price to 1M+ active app sessions.
6	**UX**	**Mobile App**	Displays instant pop-up notification with localized price.

c) Flow Table 3: Disease & Pest Control (The AI Drone Loop)

Goal: *Rapid response to crop threats using 5G and Computer Vision.*

Sequence	Layer	Product	Action / Logic
1	**Capture**	**Marut Drone**	Multi-spectral cameras scan the rice canopy.
2	**Ingestion**	**EMQX**	Receives high-bandwidth video/image stream.
3	**Streaming**	**Kafka**	Routes image frames to the vision.pest.analytics topic.
4	**Alerting**	**Alert Svc**	Creates an emergency "Geofenced Alert" for the area.
5	**Notify**	**Centrifugo**	Pushes "Locust Alert" to farmers within 5km of drone.

12. Final Architecture Picture

To wrap up the **Bharat-Krishi Connect** architecture, we are finalizing the system with a "Model Answer" approach.

This section outlines the high-level logic for **Disaster Recovery (DR)** and **Business Continuity (BCP)**, ensuring that the Chennai Data Center remains resilient against natural disasters or hardware failures.

Disaster Recovery & Business Continuity

Context: The Southern Pilot (30M farms) requires **99.99% Uptime (NFR-4)**. A failure in the Chennai DC would halt irrigation for millions and delay emergency pest alerts.

Strategy: We adopt a **"Warm Standby"** approach. While Chennai is the "Active" hub, a second DC in **Madurai** remains synchronized and ready to take over the 150,000MPS load within minutes.

1. Recovery Objectives (RTO & RPO)

- RPO (Recovery Point Objective): < 5seconds.
- RTO (Recovery Time Objective): < 15 minutes.

Final Integrated Architecture & DR Map

Component	Primary (Chennai)	Standby (Madurai)	Replication Strategy
IoT Ingestion	EMQX Cluster (Active)	EMQX Cluster (Standby)	DNS Failover (Anycast/GSLB)
Event Bus	Confluent Kafka	Confluent Kafka	**MirrorMaker 2** (Async)
Real-time Cache	Redis Enterprise	Redis Enterprise	**Active-Active CRDTs**
History DB	DataStax Cassandra	DataStax Cassandra	**Multi-DC Replication**
Identity/PII	EDB Postgres	EDB Postgres	**Streaming Replication**
Broadcaster	Centrifugo	Centrifugo	State Sync via Redis

The "Red Button" Failover Procedure

In the event of a total Chennai DC outage:

1. **Detection:** Prometheus/Grafana monitoring triggers a "Critical Outage" alert to the SRE team.
2. **Traffic Shift:** Global Server Load Balancing (GSLB) updates the **DNS record** from Chennai to Madurai.
3. **Promotion:** The Madurai **Postgres** and **Kafka** instances are promoted from "Follower" to "Leader."
4. **Resumption:** Centrifugo reconnects 15M mobile apps to the Madurai WebSocket pool.
5. **Validation:** The automated health check verifies that the 500ms irrigation loop is running from the new location.