

# **MANU MATHEW JISS**

---

PRESNTATION

# ARCHITECTURE & ETL DATA FLOW (REDIS → MONGODB)

---

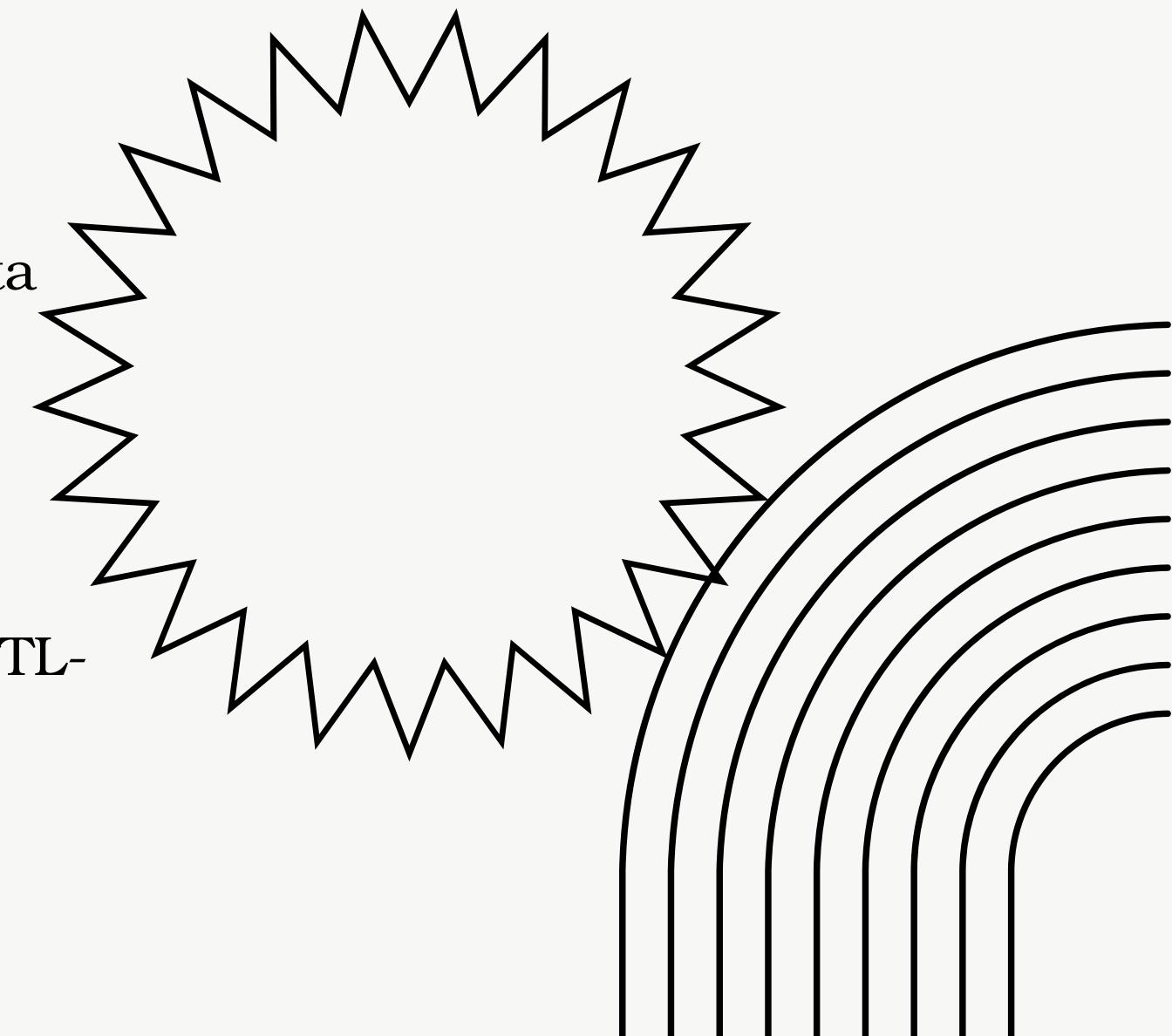
- Node.js Express application connects to both Redis (cache) and MongoDB Atlas (database)

- **Data Flow:**

1. Client sends request to Node.js server
2. Server checks Redis cache first
3. If cache hit: Return data immediately (0-2ms)
4. If cache miss: Query MongoDB (80-90ms) → Store in Redis → Return data

- **Components:**

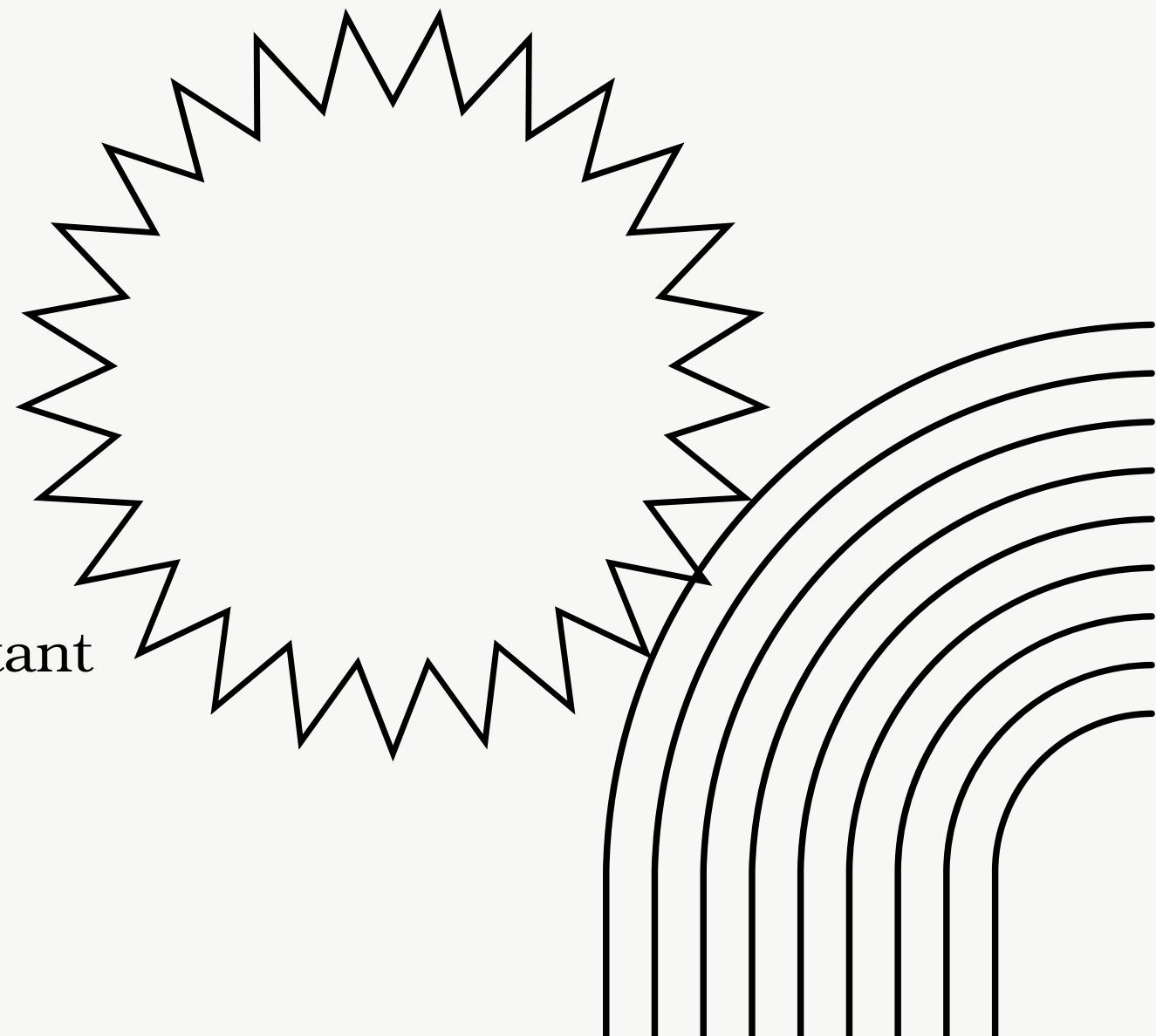
- MongoDB Atlas: Stores 2,000 sensor readings in AgriDB database
- Redis: In-memory cache for fast data retrieval
- Node.js: Handles requests and implements caching strategies
- Three caching patterns implemented: Cache-Aside, Read-Through, and TTL-based expiration



# KEY TAKEAWAYS FROM CACHED DATABASE PIPELINE

---

- Performance Impact: Caching reduces response time by 40-90x (from 80-90ms to 0-2ms)
- Database Load Reduction: Repeated queries hit cache instead of database, reducing MongoDB load significantly
- Different strategies for different needs:
- Cache-Aside: Best for full control over caching logic
- Read-Through: Simplifies code by abstracting cache layer
- TTL: Ensures data freshness by auto-expiring after 30 seconds
- Real-world application: Caching is essential for high-traffic applications to improve speed and reduce costs
- Redis is extremely fast: In-memory storage makes data retrieval almost instant



# CHALLENGES FACED & SOLUTIONS

---

- **What Worked Well:**

- MongoDB Atlas connection and data seeding (2,000 records)
- Redis integration was straightforward and fast
- TTL automatic expiration worked perfectly
- Performance improvements were immediately visible

- **What Didn't Work Well (Challenges):**

- IP Whitelisting Issue: MongoDB Atlas blocked connection initially - had to whitelist IP address
- Database Naming: Initially data went to "test" database instead of "AgriDB" - fixed by specifying dbName in connection
- Module System Confusion: Had to use CommonJS (require) instead of ES modules (import) for compatibility
- TL Testing: Had to be quick to capture cache hit before 30-second expiration - required multiple attempts

