WELCOME

# MongoDB SI Certification Presentation

**Name: Deepak Parmar**

Role: Senior Software Engineer

Company: Capgemini India

Date: 02-06-2025

📍ONLY FOR REFERENCE; DO NOT COPY THE CONTENT

# Agenda

Customer Overview

Use-case Description

Problem Statement

3 Whys

Current Architecture Landscape

Proposed Architecture Landscape ---

below is pick choose apply ---

Modernization Scorecard

Competition Comparison

📍ONLY FOR REFERENCE; DO NOT COPY THE CONTENT

# Customer Overview

The customer is a technology-driven company that provides a customer audience platform focused on automating and optimizing marketing communications. Operating withing MarTech (Marketing Technology) vertical, their platform enables businesses to target and engage audiences across various digital channels.

# Use Case Description

The system ingests data from multiple sources via Pub/Sub and Cloud Run (Node.js), performs DML operations on a Cloud SQL (PostgreSQL) database, and exposes APIs for downstream consumption. The customer is now integrating AI capabilities and requires a more scalable and resilient data platform

# Problem Statement

- **Data Loss**: Intermittent connectivity issues during high-volume ingestion led to missing records.
- **Performance Bottlenecks**: PostgreSQL CPU spikes during bulk operations, causing unresponsiveness.
- **API Latency**: Slow response times due to database contention and lack of horizontal scalability.
- **AI Readiness**: Current architecture is not optimized for AI/ML workloads requiring flexible, scalable data access.

# 3 Whys

**Why do anything?**

The current architecture is unreliable for real-time data ingestion and struggles to meet the performance and scalability needs.

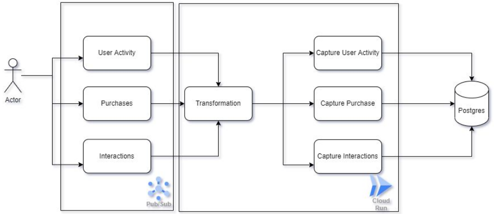
**Why Now?**

Delaying the solution increases the risk of compounding data integrity issues, customer dissatisfaction and missed business

**Why MongoDB?**

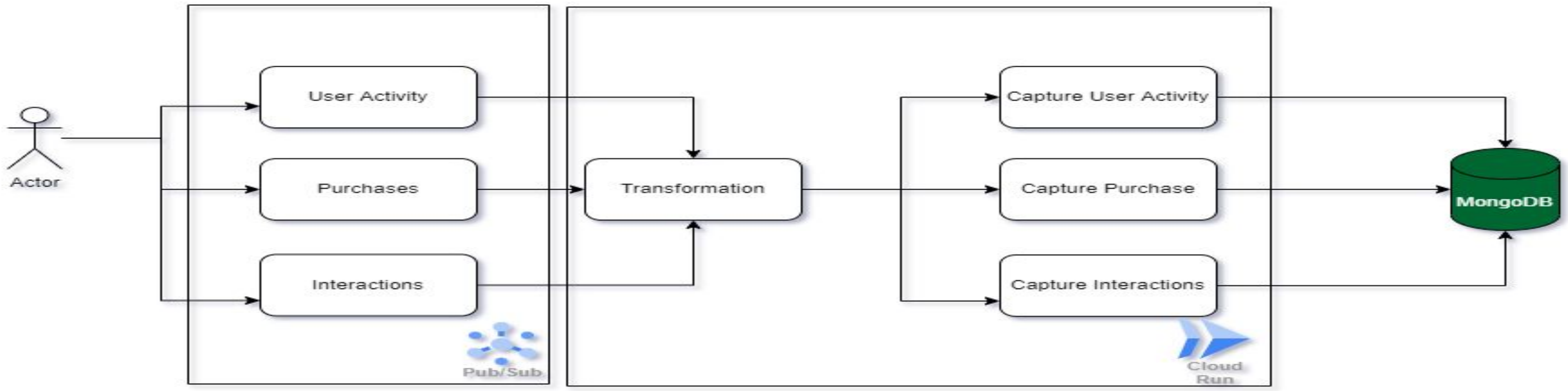
MongoDB provides a flexible schema model ideal for handling diverse and rapidly changing data in real-time. If supports horizontal scaling, low latency APIs and real-time analytics.

# Current Architecture



The current data ingestion pipeline is built using Google Cloud Platform (GCP) components to support real-time processing and analytics. In tech stack, we have Pub/Sub for reliable and asynchronous event-driven communication, Cloud Run for stateless, scalable execution environment to handle events and route them to downstream services. Cloud SQL (PostgreSQL) for transactional database but facing scalability and ingestion issues. BigQuery used for logging and analytics, supports dashboards and reporting use cases.

# Proposed Architecture



Here replacing CloudSQL with MongoDB for schema-less storage, high write throughput, horizontal scaling and real-time ingestion.

# Competition Comparison

| Criteria | Cloud SQL (PostgreSQL) | MongoDB Atlas |
|---|---|---|
| Scalability | Vertical scaling, limited horizontal scalability | Built-in horizontal scaling and global replication |
| Real-time Ingestion | CPU spikes and locks during bulk writes, poor high-volume support | Handles high write throughput, ideal for real-time data streams |
| Schema Flexibility | Rigid relational schema | Flexible, dynamic schema (NoSQL) |
| Latency | Higher latency due to locking and I/O contention | Lower latency with memory first, document oriented design |
| Cost Efficiency | Costlier for high compute/storage workloads | Pay-as-you-go, ore efficient for unstructured data |
| API Integration | Requires ORMs or connectors for REST/GraphQL APIs | Native support for JSON like documents, API ready |
| AI/ML Readiness | Requires ETL pipelines to BigQuery | Can serve as both operational for unstructured data |
| Availability | Manual HA setup, regional | Built-in multi region HA with automated failover |
| Maintenance Overhead | Requires tuning and vacuuming for performance | Managed service with auto-indexing and scaling |