Generative AI Bootcamp

Telco Use-case

Telco use-case: Analyze logs across applications and systems

The Problem:

Customer reports: "My calls keep dropping when I'm in downtown area between 2-3 PM"

Traditional Log Analysis (Current Approach):

```
Manual scan across 20+ applications:
- Call Manager logs: "Call drop rate 15% above threshold"
- Network Probe: "Latency spike detected"
- Load Balancer: "High CPU utilization"
- Database: "Connection pool exhausted"
```

You'd manually try to connect these dots, but miss the deeper relationships.

Solutioning Approach

A traditional solutioning approach would be to take our learnings on embeddings, RAG, Fine tunning, LLM reasoning, KG etc. and do the reasoning ourselves to prepare solution document.

However, with modern AI tools now available, the approach needs to shift to **“AI assisted solutioning, with human oversight”**

We will give our problem statement (defined in detail) to ChatGPT, Google Gemini / deepresearch and Claude (or similar) tools and take their recommendations as inputs to prepare our final solution approach

Prompt used to generate the solution

You are an expert solution architect in AI. I am managing a telco datacenter where applications across machines create many log files. These log files contain telco domain specific messages such as
"Call drop rates are above average",
"network probe is taking more time" etc.
These messages have a timestamp as well. When there are issues such as customer complains of calls dropping/not connecting, I need to scan logs of many applications such as call application, network probe application, route application and find out where the delay or drops are happening.
I do that by manually looking at logs of 20+ applications and pull out what seem like "problematic" messages such as "call drop rate is high" and correlate them via chronology as well as functional aspects across log files of multiple applications.
I want to build a solution using AI such that using a prompt i can ask AI to give to me correlated messages from applications that could point to the issue? I am thinking of streaming the logs to a vector database and use RAG+LLM to do the similarity search and reasoning.
I see a few challenges here:
1) The messages are domain specific such as "call drop is above threshold" or "network probe delay is high". How to ensure the AI will be able to detect these as problematic
2) There is a chronology involved.
Your task is to recommend a solid architecture and recommend an AI stack and solution document for this use-case. Take into account chronology aspects and various cost-performance tradeoffs. Consider NER, traditional ML approaches along with RAG, fine tuning, knowledge graph and other potential AI componnents that can help. Recommend an ideal architecture for this use-case taking into account cost and accuracy/effectiveness.
Also include techniques that cen be used to evaluate effectiveness.

What did Gemini and Claude recommend?

The video below shows how Google Gemini Deep Research tool was used to assist with the solutioning

Link to the video showing how it was done:

https://github.com/vijay-agrawal/genaitraining/blob/main/training-materials/logs_analytics_with_google_deepresearch.mp4

Link to the solution document Google DeepResearch produced:

https://github.com/vijay-agrawal/genaitraining/blob/main/training-materials/Telco%20Log%20Analysis%20Architecture_Gemini.pdf

Link to the solution Claude came up with:

https://claude.ai/share/104ca78d-9093-4658-8726-0ac43982d562

You can try out your favorite AI tool as an exercise

Metadata with embeddings

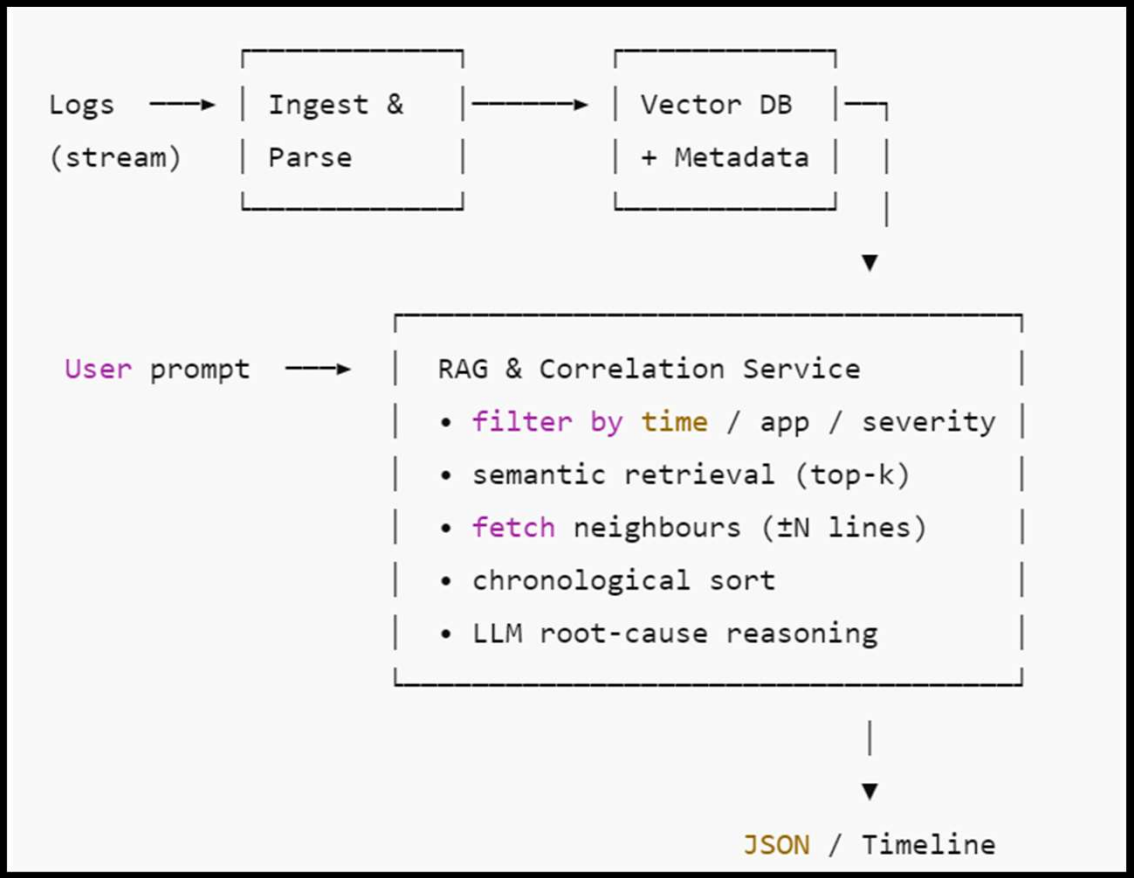
3. Storage Schema

```sql
-- Vector DB Schema
{
  "message_id": "uuid",
  "timestamp": "datetime",
  "application": "string",
  "severity": "enum",
  "embedding": "vector",
  "raw_message": "text",
  "classification": "array"
}
```

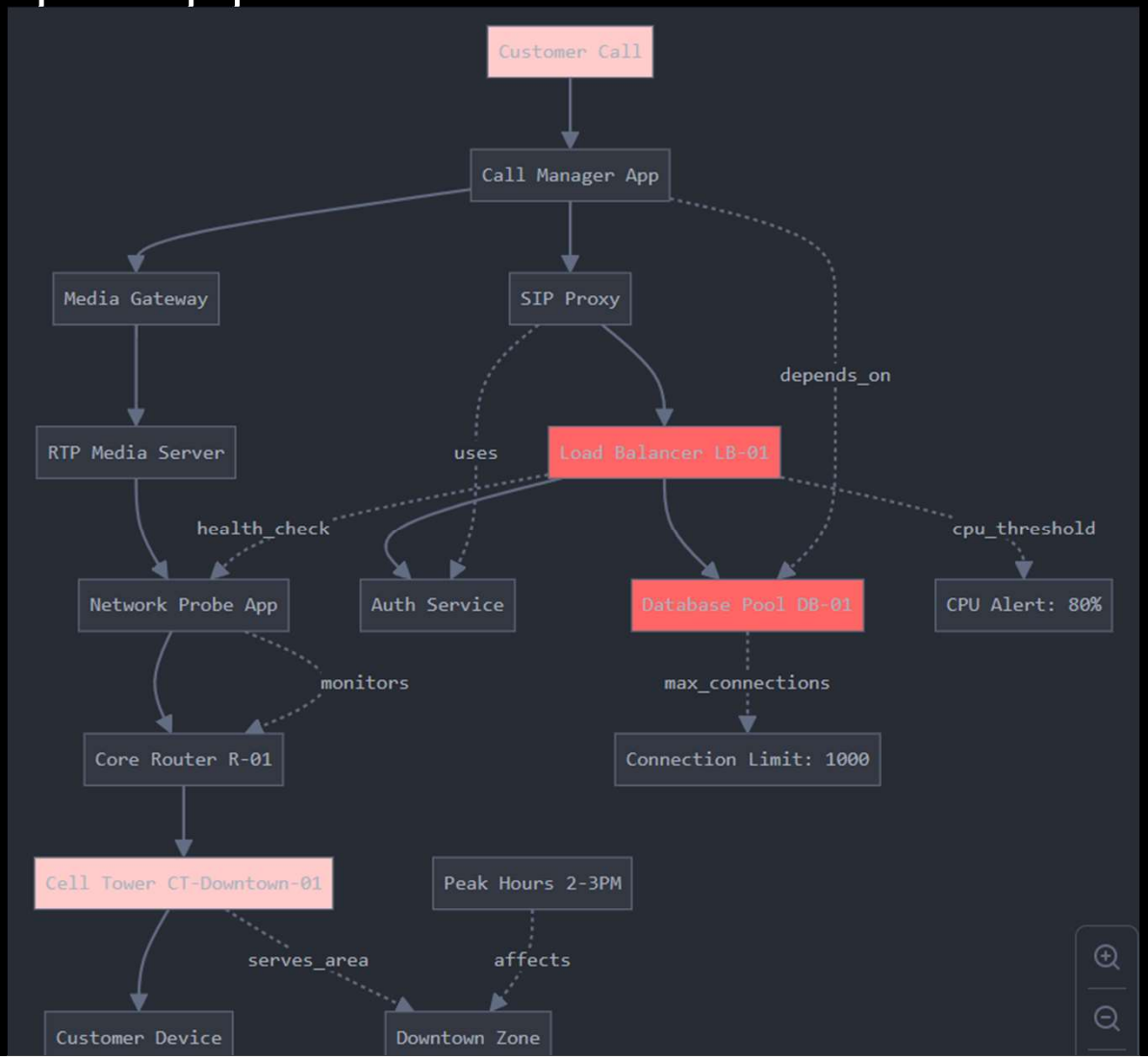
Prompt engineering for correlation

```
System: You are a telco network analyst. Analyze these
chronologically ordered log messages and identify the most
likely root cause sequence.

Context: Customer reported call drops at {timestamp}
Time Window: {start_time} to {end_time}
Applications: {app_list}

Messages:
{retrieved_messages}

Task: Identify the correlation chain and root cause.
```

Analyze logs across applications and systems



Knowledge Graph approach



AI for reasoning

AI Reasoning with Graph Context:

```
Query: "Customer calls dropping downtown 2-3 PM"

Graph tells AI:
- Downtown cell tower CT-Downtown-01 serves this area
- Peak hours 2-3PM correlates with high load
- Load Balancer LB-01 → Database Pool DB-01 dependency
- Database connection limit: 1000
- Current connections: 987 (approaching limit)

AI Correlation:
"High load during peak hours → Load balancer stress → Database
connection exhaustion → Call Manager can't establish new calls
→ Call drops"
```

2. Root Cause Path Discovery

AI for reasoning and cypher query generation

2. Root Cause Path Discovery

```cypher
// Find shortest path between problem symptoms and root causes
MATCH path = shortestPath(
  (symptom:Event {type: "call_drop", location: "downtown"})-[:CAU
)
WHERE root.status = "failing"
RETURN path, root.name, root.failure_mode
```

3. Temporal-Spatial Correlation

```cypher
// Find all components that could affect downtown calls during pe
MATCH (area:Location {name: "Downtown Zone"})<-[:SERVES]-(tower:C
MATCH (tower)-[:DEPENDS_ON*1..4]->(component:Component)
WHERE component.last_alert_time BETWEEN "14:00" AND "15:00"
RETURN component.name, component.application, component.alert_typ
```

AI for AI for generating the RCA

```
ROOT CAUSE ANALYSIS:
1. Database Pool DB-01 connection exhaustion (987/1000
connections)
2. Caused by: Load Balancer LB-01 high CPU (92%) during peak
hours
3. Impact chain:
   - Load balancer stress → Database connection buildup
   - Database slowdown → Call Manager timeout (5.2s avg)
   - Call Manager timeout → SIP proxy failures
   - SIP proxy failures → Call drops to downtown users

EVIDENCE TIMELINE:
14:15 - Load Balancer CPU spike (Network Probe App)
14:17 - Database connection pool warnings (DB Monitor)
14:20 - Call Manager timeout errors (Call Manager App)
14:22 - Customer call drops begin (Call Manager App)

RECOMMENDED ACTIONS:
1. Scale database connection pool to 1500
2. Add load balancer instance during peak hours
3. Implement circuit breaker pattern in Call Manager
```

AI for AI for generating the RCA

https://docs.google.com/document/d/1hzh-dj3ZirwuAqz82OrTg1XHxuNErXgSw2KSFEYnZLQ/edit?usp=sharing