Assembling the Puzzle

High-Performance Entity Building streaming Beam pipeline using a Two-Tiered State Architecture

Israel Herraiz & Canburak Tümer

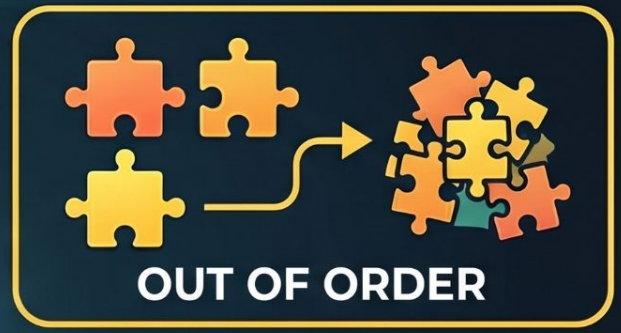
Problem Statement

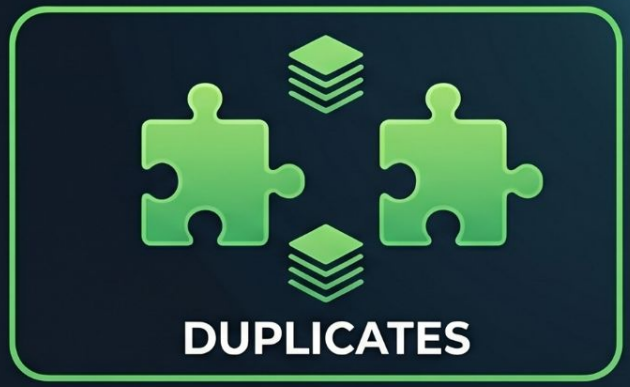
Information scattered into messages

Business Problem

- Events with complete entity data causes network load
- Events are sent with only an identifier and the updated field
- Each event should create a record in the database with all available information

KEY STREAMING DATA CHALLENGES





States could become

1

Costly

In a high traffic environment, keeping every ID in the state can be costly easily

2

Zombies

When business logic waits for an ID to become “complete”, it may never leave the state

3

Performance Bottleneck

High traffic combined with zombies can easily grow

Can a huge state cause OOM and kill your pipeline?

Solution Proposal

Bringing puzzle pieces together

Solution: Two-tiered State

Beam State

Locality is King: Sub-millisecond read/write latency.
Fragment Buffering: Handle high-velocity "puzzle pieces".
Transient Nature: Only "active" or incomplete data.

Bigtable

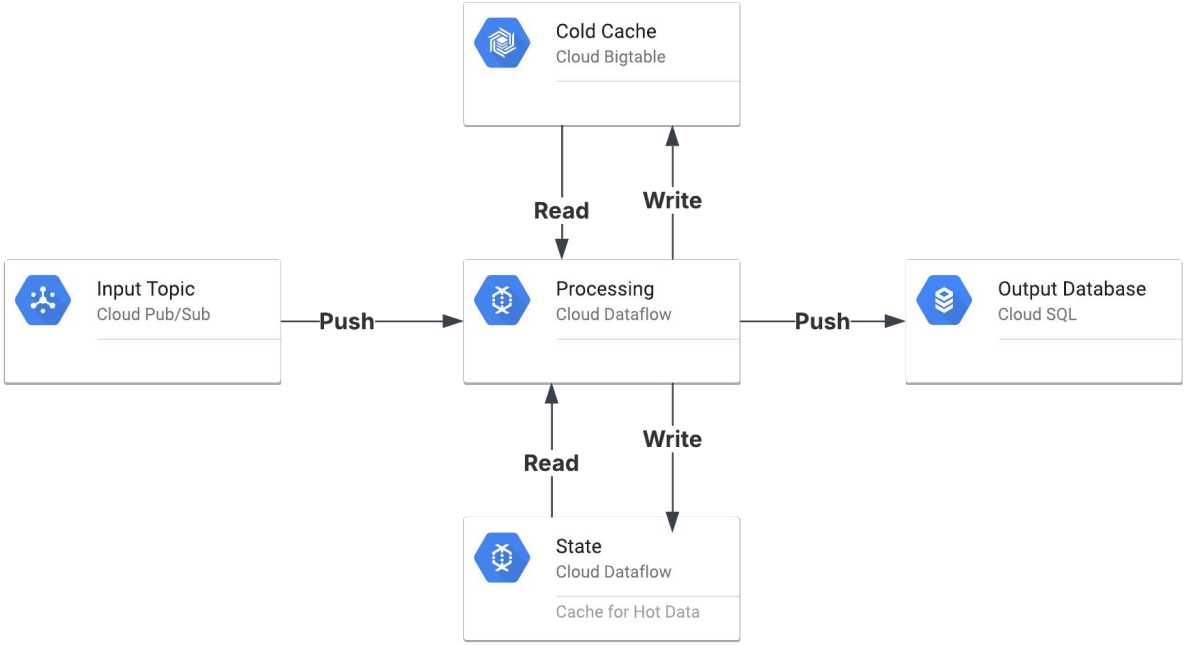
Global Persistence: Stores the entity beyond the lifecycle of a worker.
Breaking the Silo: Makes the data visible to external services.
Scale-Out Storage: Handles terabytes

Why

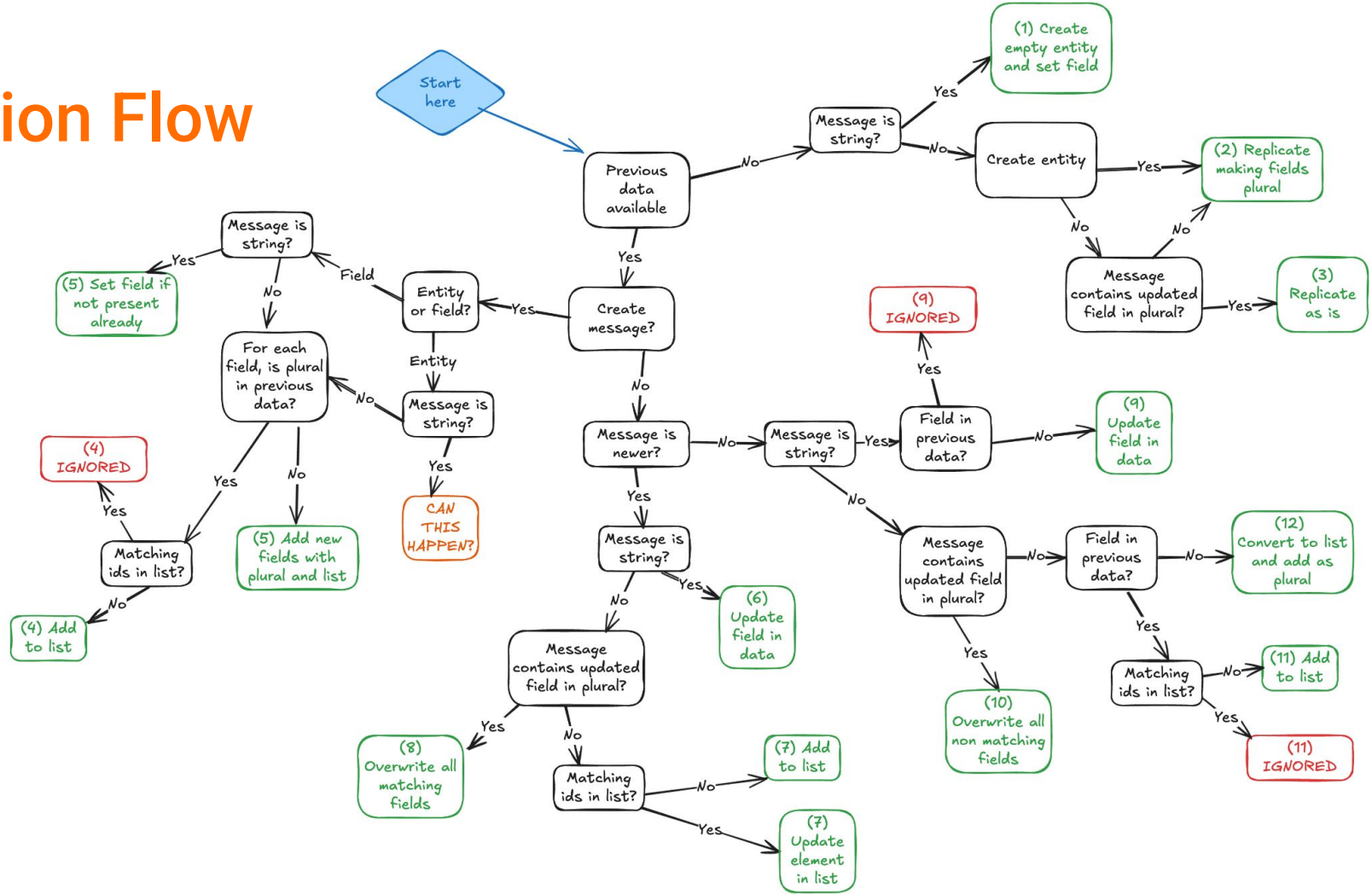
Cost Efficiency: Reduces Bigtable write volume.

System Stability: Prevents "Hot Key" issues.

Solution Architecture

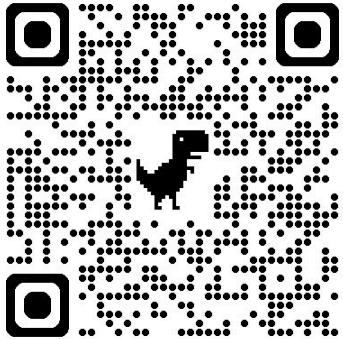


Solution Flow



Demo

See it in action!
Grab the repo from Github



github.com/iht/beam-college-2026

Thank you!

https://www.linkedin.com/in/herraiz/

https://www.linkedin.com/in/canburaktumer/