BEAM SUMMIT

NYC 2026

Streaming CDC at Scale

Spanner → BigQuery, and the Drain That Doesn't Drain

Jiufeng Liu

Senior SWE, Data Platform • Intuit Credit Karma

The business problem: Spanner → BigQuery

The Context

Many product teams at Credit Karma own their own Spanner DBs. They need data in BigQuery for analytics, ML, and reporting in real-time.

The Mandate

Decentralized teams self-serve via YAML with no platform team in the deploy loop. Over 10+ services onboarded.

Two Replication Shapes

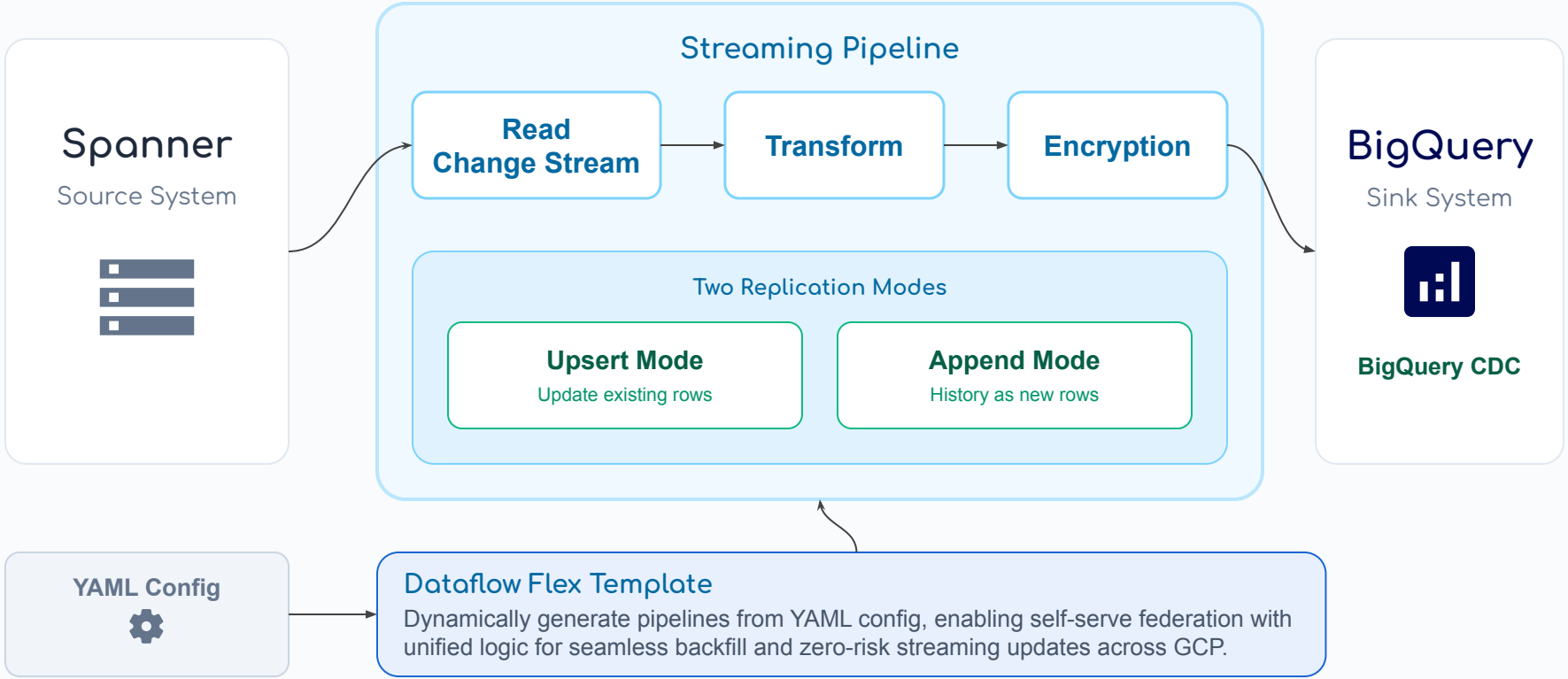
1. Upsert Mode

"What does this row look like right now?"

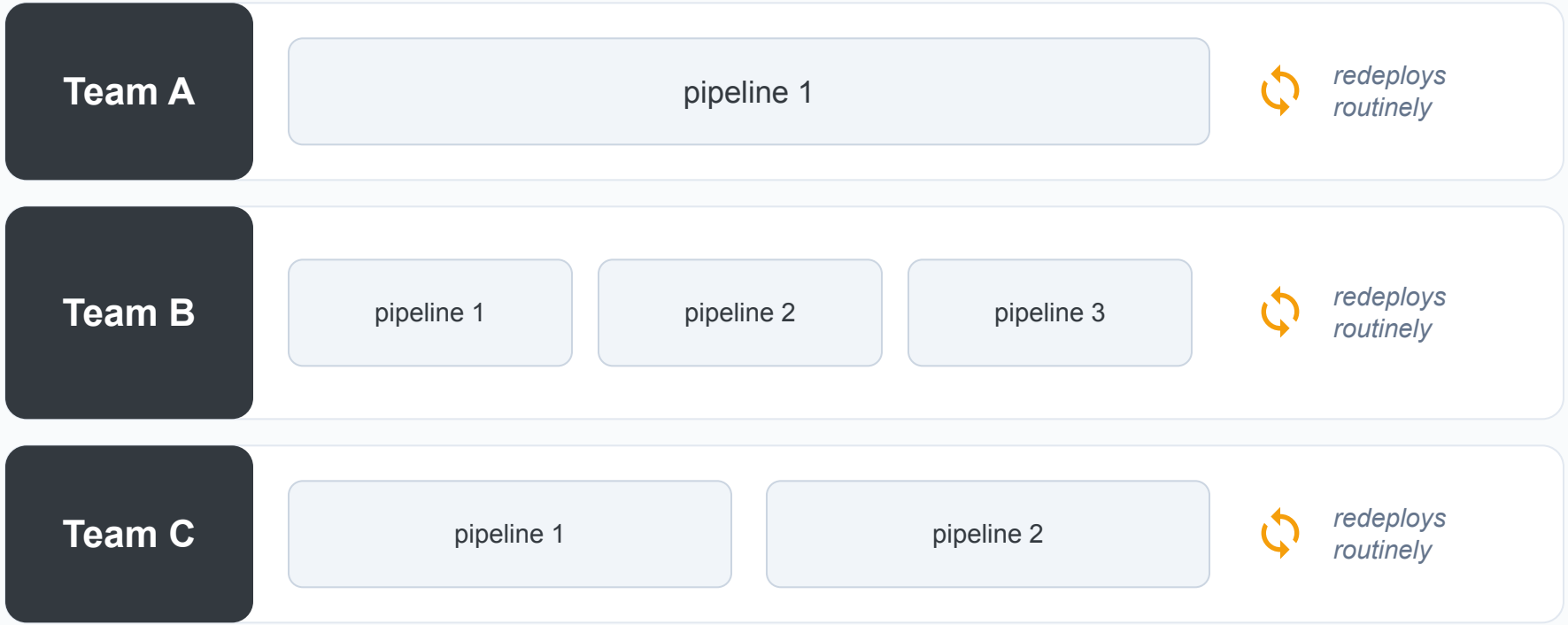
2. Append Mode

"Every change as a new row" — for compliance and historical analysis.

The Architecture: Federated Dataflow Path



Federated ownership → Many deployments



The "Drain" Problem for Dataflow SpannerIO

1. No Drain Support

Lacks logic for stopping unbounded sources at clean boundaries.

- UI "Drain" is a no-op
- Unclean stops only
- Cancel risks data loss

2. Lost Metadata

Internal partition metadata is ephemeral and not exposed.

- Offset lost on cancel
- Manually select restart timestamp

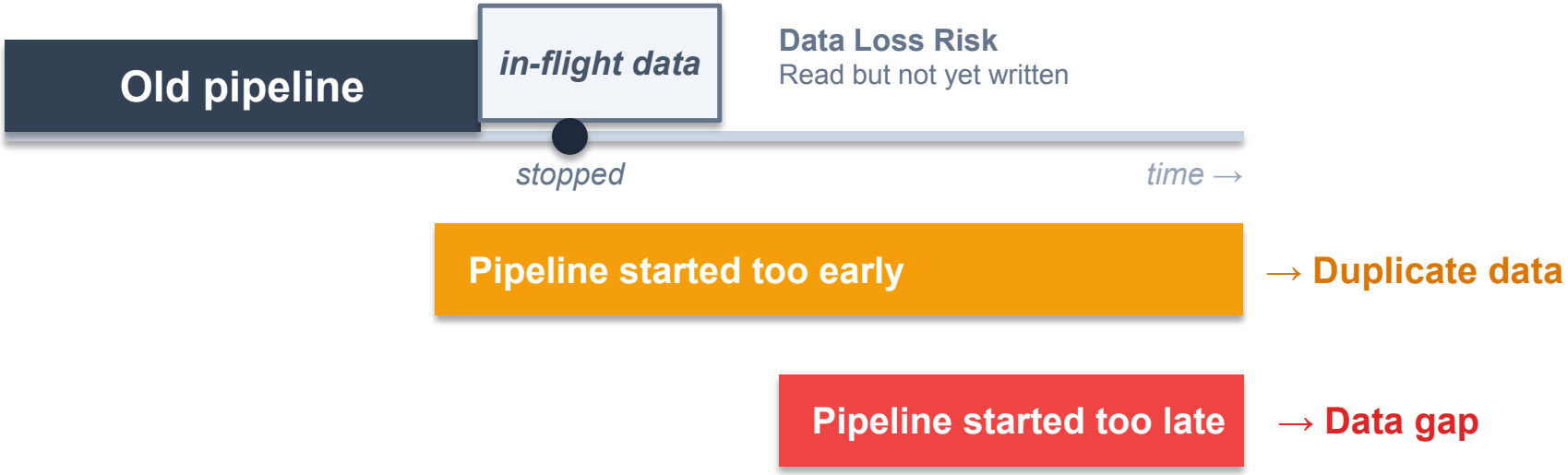
3. Fragile Update

Existing job update APIs are limited for flex templates.

- Snapshots not supported for flex template pipelines
- Pipeline graph shift breaks updates

No Beam-native public API exists for safe redeployment.

The Redeployment Risk



Routine deployment becomes a risky event for product teams !!!

The Initial Hacks

Hack #1: Hard-code Start

Set `startTimestamp` = now - 1h

Issues:

Data Risk: Loss possible if pipeline downtime exceeds 1h.

Hack #2: Replay All

Replay entire Spanner change stream(up to 30 days retention).

Issues:

Performance: Massive de-deduplication processing.

Latency: Significant lag on every restart.

Not a reliable or sustainable solution

Google's Suggestions(Confirmed gaps)

(a) PubSub Interim Layer

`Pipeline 1 [SpannerIO → PubSubIO] → Pipeline 2 [PubSubIO → BigQueryIO]`

- Redeploy 2nd pipeline; 1st pipeline runs forever.
- **Complexity and high cost.**

(b) Preflight Check

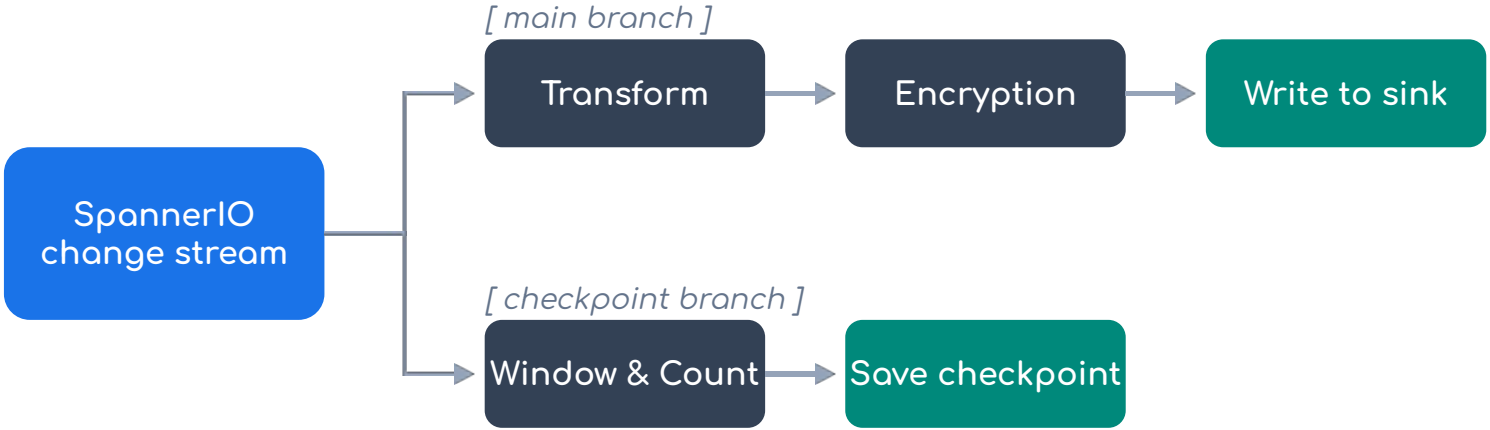
- Do preflight check for update API; fall back if check fails.
- **Fails on many YAML Config changes.**

(c) Datastream

- Less flexible because it lacks transformation and encryption.
- **High migration effort.**

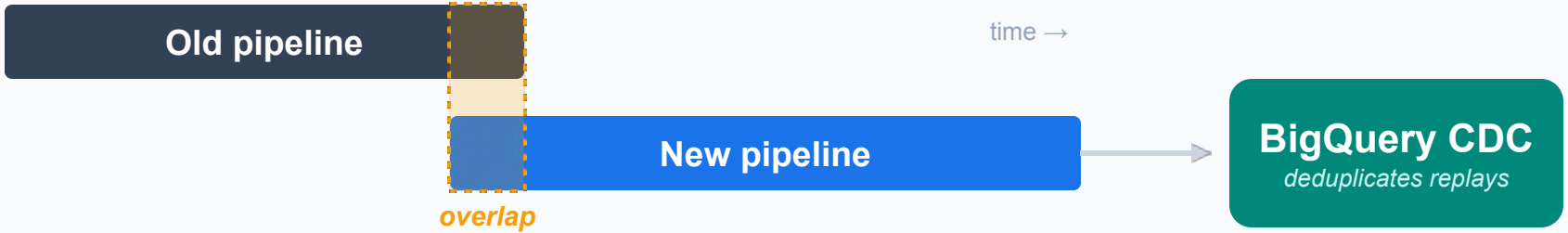
None are good enough for a platform solution.

Our Pick: Side Branch Checkpoint



The side branch is isolated from the main data stream

How It Works



1. Pipeline records its own progress as a checkpoint.
2. New pipeline starts a little before the checkpoint, **overlapping** the old one.
3. BigQuery CDC **deduplicates** the replayed records.

Redeployment Resiliency

First run ever

No checkpoint found. The pipeline starts processing from the current timestamp.

Explicit YAML Offset

YAML config override wins; existing checkpoints are ignored in favor of the YAML timestamp.

Retention Period

Stale checkpoints hit Spanner Change Stream retention limit. Policy dictates: USE / IGNORE / FAIL.

Table Missing

Create our own metadata table to store checkpoints for a zero-touch first deployment.

Write Failures

Mid-run failures are logged. Pipeline continues; redeployment falls back to last good checkpoint.

Shared Metadata DB

Each Spanner change stream has one checkpoint; no collisions.

*Covers scenarios from **unreachable Spanner** to **job cancellation** with guaranteed replay.*

Recovery Strategy

When to Recover: Used for bad data states requiring full rebuild or targeted replay.

Option A: Batch Backfill

Truncate + Rebuild

Best for significant corruption or sync issues.

- **Truncate:** Clear affected tables.
- **Batch Job:** Read Spanner snapshot to BigQuery.
- **Overlap:** Start streaming ~10m before batch end. BigQuery CDC will do deduplication.
- **Restart:** Pick up ongoing changes.

Option B: Streaming Backfill

Replay the Gap

Best for bounded data gaps within Spanner change stream retention period.

- **Gap Job:** Target specific timestamp ranges.
- **Deduplication:** BigQuery CDC handles replayed events.
- **Safety:** Skip checkpointing to avoid interference.

Future Improvement

Problem: Still Imperfect

- Deduplication still depends on the sink
- Adds an extra branch to every pipeline

Where the Real Fix Belongs

1. **Native resume across redeployments**
   SpannerIO + Beam track last job stopped offset
2. **Graceful drain for Spanner streams**
   Dataflow stops a streaming job cleanly

Google Cloud Support

Google Cloud

THANK YOU!

Jiufeng Liu

Senior Software Engineer
Data Platform, Intuit Credit Karma

jiufeng_liu@intuit.com

Special Thanks

Teams

- Spanner / Dataflow / Technical Support, Google
- Data Platform / Infra / Dev Efficiency, Credit Karma

Individuals

- **Naresh Kotha** — Sr. Manager, CK
- **Ushhud Khalid** — Staff Engineer, CK
- **Nick Gridinskiy** — Sr. Engineer II, CK

INTUIT creditkarma

Streaming CDC at Scale

QUESTIONS?

Jiufeng Liu, Senior Software Engineer
Data Platform, Intuit Credit Karma
jiufeng_liu@intuit.com