

## CASE STUDY

## Insights Gaming



*"We were using Google Cloud Datastore, which was fine to start, but as the project became more and more complex, we had to put in more effort for data management than our business logic."*

-Jim Ko, Chief Product Officer

# Esports coaching collaboration platform builds features faster with CockroachCloud

## When Manual Data Management Impedes Feature Development

Insights Gaming is the premier collaborative coaching platform in the rapidly growing esports industry.

Insights Gaming uses CockroachCloud, the Cockroach-DB-as-Service offering, as their general purpose database. In this article we'll cover the data management obstacles that caused Insights to migrate from Google Cloud Datastore, their decision process, and the reasons why CockroachCloud has enabled their engineers to build and iterate on features more quickly to deliver a better customer experience.



*"We had to either stop adding new features, or change the database."*

- Steve Lam, Lead Engineer

Insights Gaming built a game-agnostic collaborative platform on which esports players and coaches can share real-time video of their performances and leave feedback for each other. Building the collaborative platform on a NoSQL database was manageable at the start. But each new feature that required relational data management also required manual workarounds. The data management responsibilities were beginning to consume time that was needed for feature development.

Any database migration requires effort and planning, and the team avoided it for as long as possible. But the quantity of time required to manage the data had begun to limit feature production. The developers had to manually link entities together to create relational structures in their NoSQL database. But whenever they added a new type, they had to remember everywhere else it was used. Inevitably, this led to errors.

An example of a problem caused by all the manual linking was a situation in which a user deleted a comment on the coaching platform, but the comment was still visible. The developers didn't want to add new features because of the complex gymnastics they'd have to perform within the database. The additional time spent manually linking entities was also causing problems for product managers. They struggled to properly size new feature requests.

The database was preventing Insights from adding new features they wanted to add, and so they finally decided they needed to migrate to a relational database before it impacted user experience.

## Why Insights Gaming Chose CockroachCloud, a relational database-as-a-service

When Insights decided to migrate to a relational database, they created the following database requirements:

**Standard SQL:** their developers were all familiar with SQL and felt they would be more productive with it

**Cloud-Native**

**ACID transactions**

**Resilient**

**Easy to maintain:** Low data management overhead!

They evaluated a few options, including Vitess, and decided to go with CockroachDB since they wanted a modern database that would scale easily and distribute data automatically. And they chose to go with the database-as-a-service option, CockroachCloud, because they wanted to allocate all their resources towards building out features and improving the user experience.

## Application Architecture with CockroachCloud

As the general-purpose database, CockroachCloud stores all the latency sensitive data for Insights. That means CockroachCloud stores user data pertaining to account information, user access control, user generated content (comments and annotations), and payment processing.

Insights Gaming used to use a pub/sub system to send real-time updates to clients. But with CockroachCloud on board, they've been able to simplify their code using changefeeds connected to cloud storage sinks to broadcast live updates to clients.

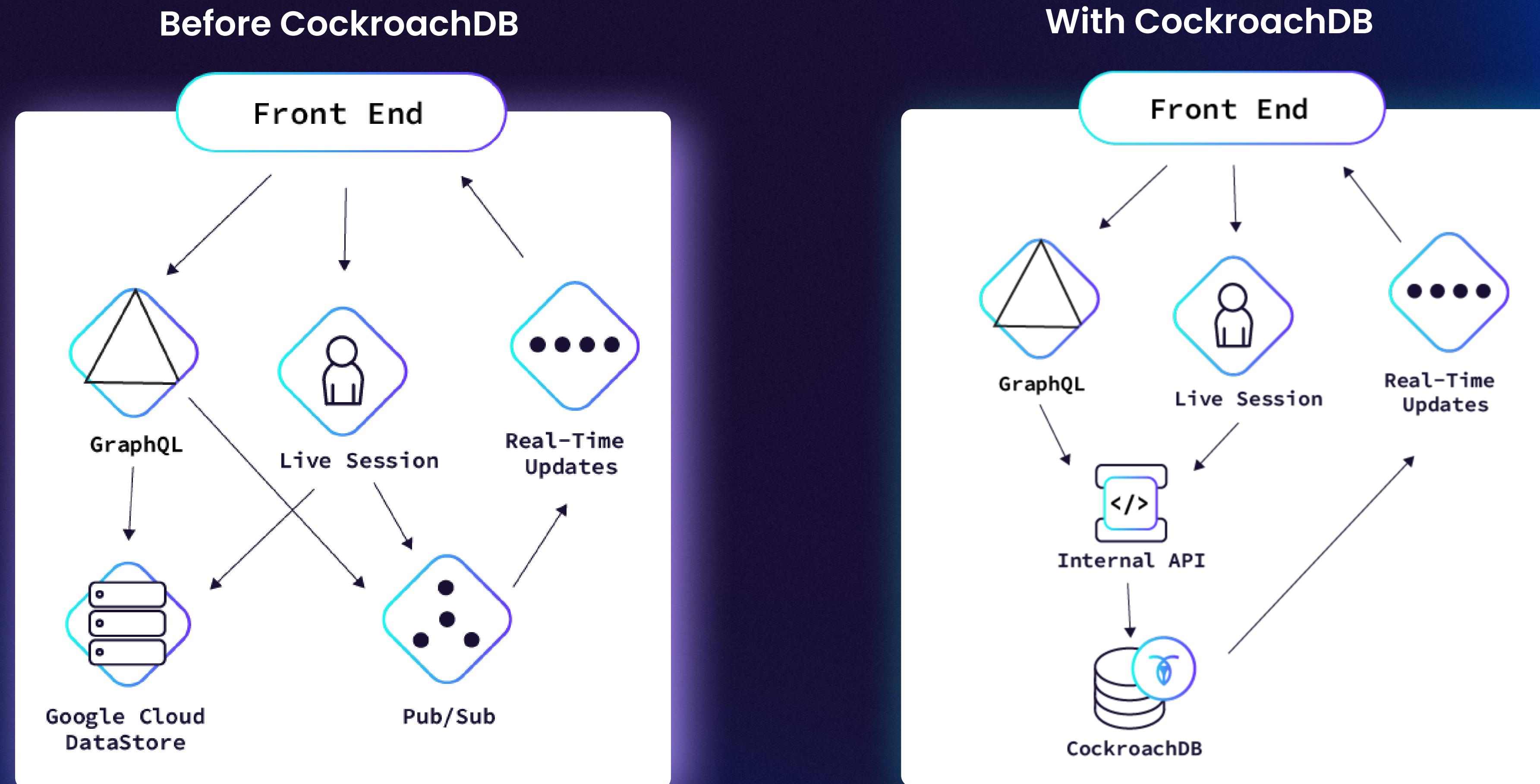


*“Previously we had to keep track of how everything interacted with everything else in the system. Which took up a lot of time and energy. Now we just define interactions within the schema. With all the extra time I’m finding that the team is more creative with our feature development.”*

- **Steve Lam, Lead Engineer**

## CASE STUDY

# Insights Gaming



## Time Travel Queries & other useful features

Some other features that the Insights development team finds helpful are cascading deletes, automatic garbage collection, recursive CTEs, as well as time travel queries which they use in tandem with changefeeds in their GraphQL subscriptions system to calculate differences to deduplicate the update messages that get sent to users (it has also saved them a few headaches when data was deleted accidentally).

Together, all of this creates more time for developers to focus on feature building. For example, right now Insights Gaming is building a new tool to help gamers record and evaluate their performances. This is a feature they simply would not have been able to build with their previous architecture.



*“We didn’t want a database where we’d have to use brute force solutions that use software that was never meant for global distribution.”*

**-Steve Lam, Lead Engineer**

## Growing the Business with CockroachCloud

Insights Gaming has future-proofed their business, enhanced feature production, and unlocked the creativity of their team with CockroachCloud. The combination of a database-as-a-service with the capabilities of elastic scale, bulletproof resilience, and familiar SQL has allowed Insights Gaming to build what they want, when they want. No other database could give Insights Gaming this combination of modern database capabilities and ease of use.

Currently Insights Gaming has customers in the US, Canada, UK, Germany, and Korea. They are going to begin using CockroachDB’s geo-partitioning feature to keep data close to users so that they can scale the business to Asia and keep latencies low.