

CASE STUDY

How Storied Leveraged Event Sourcing to Power a Global Genealogy Platform.

From their beginnings as an online hub for digitized historical newspapers to the accessible and innovative genealogy site they are today, Storied strives to take family history beyond the records. Their platform allows users to collaborate and reveal the stories behind their ancestors, with access to historical records, tools for creating a family tree, plus AI assistance to string it all together and more.

Behind the scenes is an event-driven system that leverages event sourcing, backed by Kurrent Cloud (formerly Event Store Cloud). Kurrent Cloud is a fully managed, multi-cloud service that enables developers to provision and deploy secure, high-availability KurrentDB clusters across AWS, Azure, and GCP in minutes, eliminating infrastructure management overhead. CTO, Chris Bradford, who was the first developer for Storied when the project began, talked us through the evolution of the system and some use cases for which Kurrent has been central.

| Storied and their journey to Event Sourcing

For Chris, the benefits of an event-driven system for Storied were clear from the start: “It kind of gets you into that mindset of how the different pieces of your system operate, how you think about the way state is managed, but then also how state changes are communicated through a distributed system.”

Event sourcing is a “natural fit” with such a system and having used it in a previous role, Chris felt confident that it could provide a competitive advantage to Storied. Especially considering the need for a system that could scale quickly with surges in demand from annual conferences, as well as the value an immutable event log could add to a Family History Researcher.

“A complete audit trail would help answer questions of when any changes were made and who made them.”

— Chris Bradford, Storied CTO

The next question was how to go about it. In his previous role, Chris's team had built their own event store before migrating it to DynamoDB in AWS. However, he describes himself as a "big fan of leveraging managed services as much as you can rather than building them yourself, unless it's your core competency – and event stores are not the core competency of a family history business."

As such, Chris wanted to go with an "established provider." He considered Marten, and had also heard good things about Kurrent from peers, adding "it was a plus that it's open source."

“ I liked the fact that Kurrent was purpose built and the technology was more agnostic, so we had the ability to leverage it across different technology stacks. ”

— Chris Bradford, Storied CTO

At the time, Kurrent Cloud was in development and describing himself as a big cloud proponent, Chris – like many of our cloud customers – was keen to "have that managed for me rather than me having to figure it out or have a team member who needs to know all about clustering and backups."

| Adopting KurrentDB

We often talk of a learning curve when it comes to adopting an event sourced approach for the first time; for Chris, event sourcing wasn't new but KurrentDB was. However, he determined that "using Kurrent itself was not that hard; there are good libraries that we can use and they work well." He adds: "I like the simplicity of Kurrent. It's proven very stable for us and the mental model is quite nice."

Chris also found the Kurrent community spaces useful, particularly early in the project: "open source tends to be accompanied by a greater sense of community, helping each other solve problems. I will say that the community of developers who are active in the Event Store space tend to also be very aligned with the kinds of architectural approaches that we're taking more broadly, so I didn't just learn about Event Store but also the common patterns around it."

When Chris built his team, all members were new to event sourcing. This required his efforts to help illustrate the event based mindset. He found that once the team understood the patterns and saw them in practice, it "actually made patterns quite standardized so was proven to be productive." This helped make Kurrent "easy to adopt."

| Example use cases

With Kurent now a central part of Storied's architecture, Chris highlighted for us a couple of use cases to illustrate how the event sourcing database has helped solve key challenges. In particular, note how KurrentDB has seamlessly integrated with other technologies such as graph databases and analysis tools.

→ Use case 1: Handling vast amounts of data

Researchers often have existing family trees that they want to import into Storied using the standard genealogical data interchange format GEDCOM. The system therefore needs to be capable of handling an enormous amount of data; “we’re talking tens of thousands of people, each of whom may have many different data points that we also store as independent entities, relationships among those people and so forth.”

KurrentDB – which is praised for its ability to handle large volumes of events – has “no problem keeping up with that data.” Chris explains, “we can send 100,000 commands into our system for this file and the system handles it fine.”

However, the graph database – CosmosDB – that lies at the heart of the family tree system struggles to handle this volume of information in a short period of time. “We wanted all of the data for a family tree to live on a single partition, which would quickly get overwhelmed with the volume of data that we were writing from a single one of these files. We had to come up with a solution for how we import a file, while using the same sort of infrastructure we built for everything else.”

The infrastructure Chris is referring to involves command handlers and event handlers. “The current method we use writes events to Kurrent and we have an application that has a subscription; it’s a way of keeping up with the latest things coming into Kurrent and publishing them out to Azure Event Hubs. We then leverage another nice feature of the Orleans framework, which takes those messages published through Event Hubs, partitioning that out and processing them in parallel so actual Event Store subscribers are doing almost no work.”

Chris continues: “it’s only publishing these out quickly, maybe looking at some metadata, so the rest gets parallelized out. It’s that parallel processing that was overwhelming CosmosDB.”

The solution was in KurrentDB’s projections. Chris was able to assign a specific correlation ID to an import, that would in turn flow through the commands to all events.

He explains: “in our subscriber, we’d detect that one of the GEDCOM imports had started and could track the correlation ID for that event. We could then just look at every other event, ask if it has the same correlation ID, and ignore it until we get to the event that says the import is completed.

At that point, we have the correlation ID and can read that projected stream from beginning to end, create a CSV and store it. A background process then goes through, using a bulk import functionality to import it into CosmosDB at a rate that doesn’t overwhelm the database. From there, we can send another command to publish an event that says everything’s in the graph database and is now available in the family tree.”

He concludes “having this correlation ID projection was super helpful and it made it really easy for us to figure out how to solve this particular problem of very high volumes of data coming in at once that need to be processed in a more asynchronous way.”

→ Use case 2: Deep analytics

With Kurrent now a central part of Storied's architecture, Chris highlighted for us a couple of use cases to illustrate how the event sourcing database has helped solve key challenges. In particular, note how KurrentDB has seamlessly integrated with other technologies such as graph databases and analysis tools.

Data is a powerful asset for any forward-thinking business and by using a state transition database like KurrentDB, Storied can leverage their rich data for in-depth analysis, with benefits extending far beyond the software team.

"Business users and analysts know we have a really rich source of data for analysis; every day we read all of the last days events and ship them off to our data warehouse team, who then uses Snowflake to build all kinds of reports.

If we want to know how many users have created a tree or published a story, it's easy as we have an event that's named in a meaningful way, rather than getting a SQL table named Q underscore ID underscore F two, right or whatever it is that you so often so often get."

Chris notes how the use of events has allowed business users to grasp an understanding of the system, what's possible and how projects are actioned: "the company has gained an understanding of how we think about events and this is at the heart of how we implement things and talk about business."

| Summary

Storied's adoption of KurrentDB has not only revolutionized their genealogy platform's technical architecture but has also empowered their team to efficiently tackle key challenges. The seamless integration of KurrentDB into Storied's event-driven system has enabled Storied to enhance user experiences, from proactive historical content matching to efficiently handling large data imports. Moreover, the platform's commitment to an event-centric mindset has not only streamlined development but has also provided valuable insights for business users, fostering a deeper understanding of system operations and opening avenues for rich data analytics.