

CARE.COM

# Care.com Accelerates Time to Market and Improves Customer Experience with Confluent



## Headquarters

Austin, Texas

## Industry

Online Marketplace

## Challenge

Integrate a collection of large, monolithic systems into a single platform that accelerates time to market and improves the overall customer experience.

## Solution

Leverage the scalability and reliability of Confluent Cloud to reduce the operational burden of managing the core infrastructure.

## Results

- Increased operational agility and time to market
- Devops time-saving
- Better scalability
- Enhanced security

An online marketplace for a range of care services including childcare, senior care, tutoring, pet care, and housekeeping with millions of members across 20 countries, Care.com was in need of a much simpler, unified IT architecture to be able to streamline go-to-market initiatives and pivot quickly and effectively with the ever-changing needs of its customers and the marketplace in general.

To accomplish this architectural transformation, the company began building a new platform—the Bravo Platform—that could seamlessly scale with its changing data needs while taking a more agile, granular, and nuanced approach to data security. The company was already using Confluent Platform but chose to transition to Confluent Cloud to power the Bravo Platform and achieve its goal of unifying a series of monolithic architectures into a single, agile platform that could accelerate Care.com on its stated mission of being a one-stop shop for families of all types to meet all of their care needs.

## Challenges

Care.com had been stringing together the operation of various monolithic architectures that serve its markets across the globe. To bring these monoliths together, they created a centralized, microservices-based platform and began to use Apache Kafka® for messaging and data integration.

"In early 2020, one of our big initiatives was to take these disparate systems and combine them into a single platform that would allow us to serve the providers and seekers of care on the platform more consistently across all the different verticals that we manage," explained Matt Coddington, Senior Director of Devops Engineering at Care.com.

To accomplish this unification of disparate systems, Care.com began to build its Bravo Platform. At this point they were already using Confluent Platform in support of their core monolith, but chose the fully managed Confluent Cloud for Bravo because of its scalability, reliability, and the desire to offload essential but complex operational parts to experts.

"Confluent Platform was exclusively for the large domestic monolith and a little bit for the microservices historically built alongside of that," Coddington said. "It was being used a little bit for messaging but primarily for queuing work. For example, there might be microservices that have a certain function and when certain events happen, the monolith dumps that event into the topic and the microservice consumes it and does whatever work needs to be done. There weren't many topics that were subscribed to by multiple microservices or an idea of trying to share the events from the monolith side into the microservice side."

## Technical Solution

Switching from self-managed Confluent Platform to Confluent Cloud became a clear choice for Care.com in early 2021 when they realized they needed Confluent Cloud for scale and resiliency and also to free up Care.com developers to focus on higher-level projects and initiatives on their new Bravo Platform.

The team began to use Confluent Cloud with the central goal being to take apart these monoliths and put them back together again as a unified structure.

"There are a lot of events and messages streaming in, related to things that are happening on these monoliths. These messages are being consumed by microservices that will eventually, for example, become stores of truth for that aspect of the monolith as we start to take it apart," Coddington said.

Per Sean Schade, Platform Architect at Care.com, there were three central questions they were asking around their messaging architecture to get the platform unification to work:

1. Which messages or events are being generated?
2. What's the serialization format those messages use and can it be interoperable between all the different run times (i.e., Java, .NET, Go, etc.)?
3. How can they control and manage the schema evolution of those messages?

"That was the main thrust behind why we wanted to use Confluent Cloud," says Schade. "We wanted to use the Schema Registry and leverage Protobufs for our serialization format."

Accomplishing this required some customization on their part. First, they used protocol buffers they called "Care APIs" (inspired by Google APIs) to define all of their data and service contracts. Then, they created a custom Protobuf extension for their Kafka messages that could identify the topics events can be published to. This functionality is powered by CI/CD automation that does breaking change detection, ensures backwards compatibility, and publishes to the Schema Registry. Finally, they generated the client stubs for the various run times they support to control what's going in and out of Kafka.

## Business Results

**Increased operational agility and time to market.** "One of the main benefits Confluent Cloud has provided is being able to make the changes we want to make quickly and easily by creating a more seamless backend experience," Schade said. "With the monolith, it often took weeks to try out and test. Now we can iterate quickly with a small team in a controlled setting to be able to make those services better."

**Devops timesaving.** "Offloading that day-to-day burden of operations has been a huge help," Coddington said. "A lot of overall operations-type work gets offloaded when you move to Confluent Cloud. But you have the additional offload of expertise because you have Kafka experts at Confluent to assist in troubleshooting more complex issues as they come up. Where we're saving time now is on the devops side of maintenance of all those systems—patching underlying systems or upgrading Confluent Platform—those were big things to be able to offload."

**Better scalability.** "What we're trying to build with the Bravo Platform is the ability to be agile even as we scale to something pretty large," says Coddington. "Anytime you build out a new infrastructure or platform you're probably going to be able to move pretty quickly initially, but then as it scales it gets bogged down unless you design and implement things with some planning and standards in mind. With Confluent Cloud, you can programmatically get ahead of data-type issues and backwards compatibility-type issues. On the devops side, we've built out some automation to handle the creation and maintenance of topics, ACLs, and service accounts so that all lives in code as well. And when it's all in code, it can scale better and provide programmatic guardrails as the platform grows."

**Enhanced security.** "We have much finer-grained control now over security with service accounts per consumer or producer with the correct ACLs and better control over the schema messages being produced and their serialization format," explained Schade. "We define a lot of that with our schemas now so we can identify sensitive data and either not allow it through or ensure that it is masked or redacted, which is where we didn't have visibility before. We are also able to tag metadata about a topic, knowing and tagging PII via data lineage, and using it to tag topics within certain security or compliance profiles, which all ties into audit-related activities."

## The Future

"We plan to leverage Confluent really heavily on the analytics and data warehouse side," Coddington said. "Right now the ingestion is very simple where it can subscribe to a topic and publish via JSON to an S3 bucket but there are things we can do with single-message transforms and KSQL streams to really ramp that up. ETL is happening downstream and we're thinking of bringing it more upstream, maybe to KSQL."

A key aspect of this process, Coddington said, will be ensuring metrics consistency.

"When we're pulling apart all these different infrastructures and data stores, it's really important to ensure all the metrics stay consistent," he said. "Even today we struggle to have a consistent way to define across all of these different environments some of the core business metrics, and I think standardization is the key there. Being able to do these things in a place that's visible to engineering as a whole and not hidden in some SQL query implicit in some report that's being generated, the closer you can push that to a standard place using stranding tooling, the better off you're going to be. This becomes super important with the kind of migrations we're doing because you want to be certain you didn't break anything and in the end what you're trying to do is simplify all of this. Confluent Cloud will of course be a key part of this process."

## Learn More About Care.com

<https://care.com>