Michelle Wetzler Follow
Chief Data Scientist @keen_io
Apr 4 · 5 min read

Photo by Ondrej Supitar

Architecture of Giants: Data Stacks at Facebook, Netflix, Airbnb, and Pinterest

Here at Keen IO, we believe that companies who learn to wield event data will have a competitive advantage. That certainly seems to be the case at the world's leading tech companies. We continue to be amazed by the data engineering teams at Facebook, Amazon, Airbnb, Pinterest, and Netflix. Their work sets new standards for what software and businesses can know.

Because their products have massive adoption, these teams must continuously redefine what it means to do analytics at scale. They've invested millions into their data architectures, and have data teams

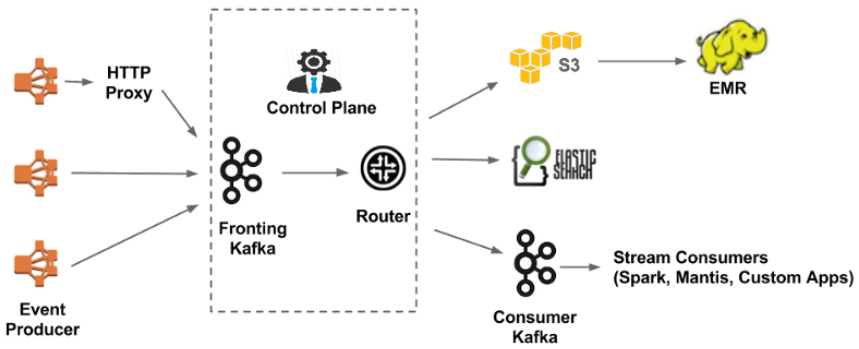
that outnumber the entire engineering departments at most companies.

We built Keen IO so that most software engineering teams could leverage the latest large-scale event data technologies without having to set up everything from scratch. But, if you're curious about what it would be like to be a giant, continue on for a collection of architectures from the best of them.

Netflix

With 93 million MAU, Netflix has no shortage of interactions to capture. As their engineering team describes in the Evolution of the Netflix Data Pipeline, they capture roughly **500 billion events per day,** which translates to roughly **1.3 PB per day**. At peak hours, they'll record **8 million events per second.** They employ **over 100 people** as data engineers or analysts.

Here's a simplified view of their data architecture from the aforementioned post, showing Apache Kafka, Elastic Search, AWS S3, Apache Spark, Apache Hadoop, and EMR as major components.

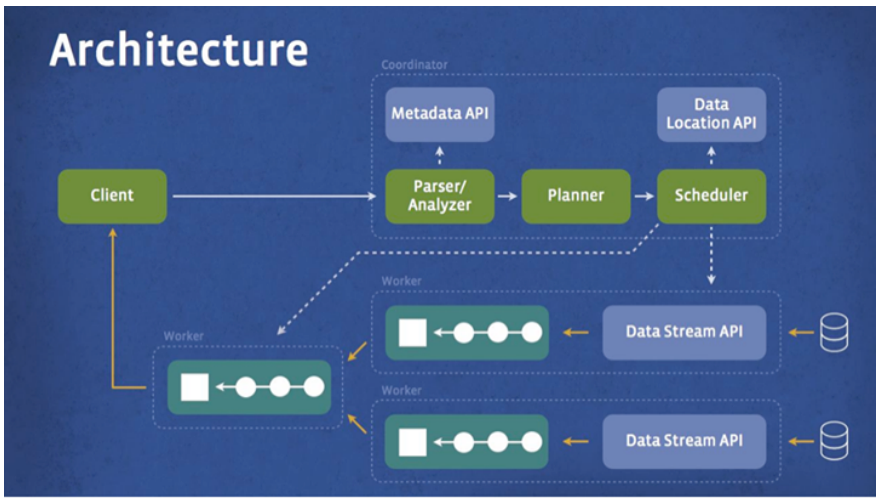


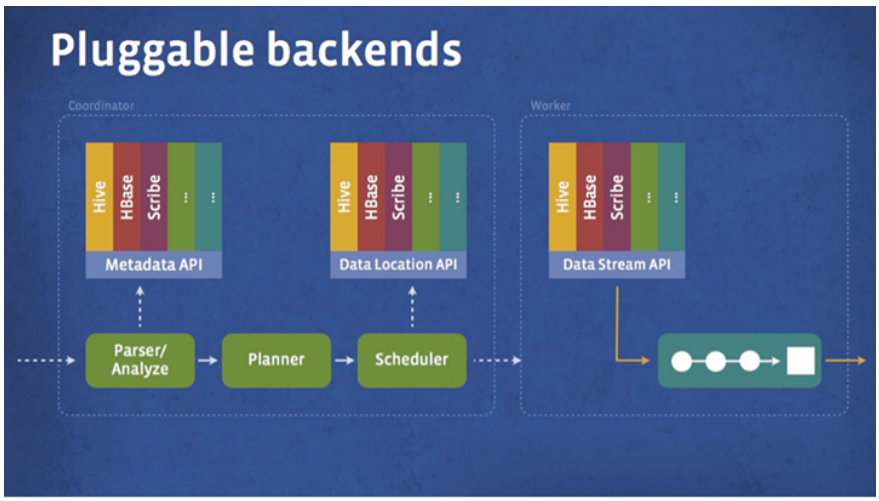
Source: Evolution of Netflix Data Pipeline

Facebook

With over 1B active users, Facebook has one of the largest data warehouses in the world, storing more than 300 petabytes. The data is used for a wide range of applications, from traditional batch processing to graph analytics, machine learning, and real-time interactive analytics.

In order to do interactive querying at scale, Facebook engineering invented Presto, a custom distributed SQL query engine optimized for ad-hoc analysis. It's used by over a thousand employees, who run more than 30,000 queries daily across a variety of pluggable backend data stores like Hive, HBase, and Scribe.

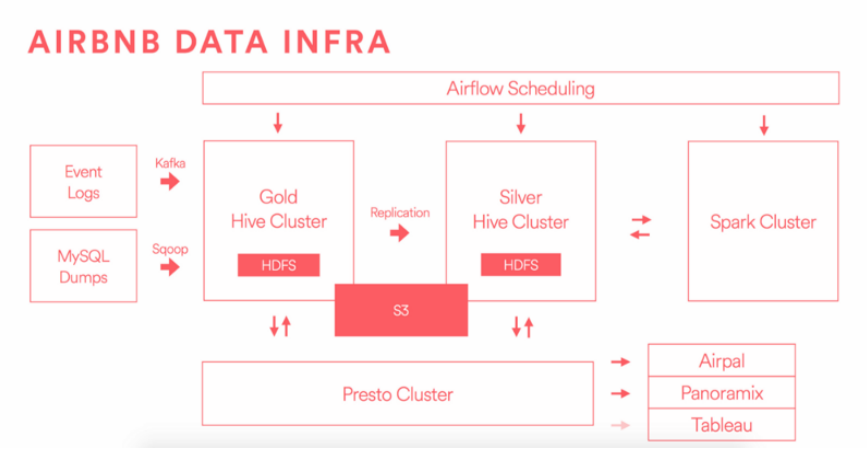




Airbnb

Airbnb supports over 100M users browsing over 2M listings, and their ability to intelligently make new travel suggestions to those users is critical to their growth.

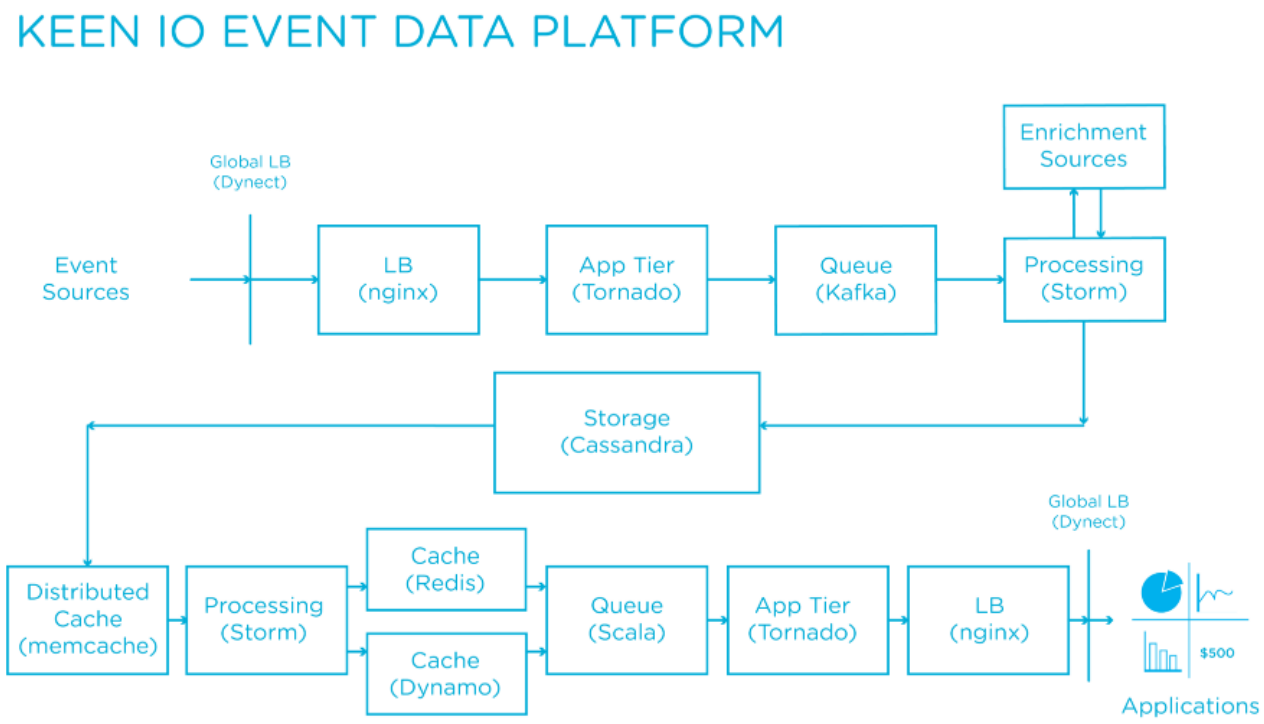
At a meetup last year "Building a World-Class Analytics Team", Elena Grewal, a Data Science Manager at Airbnb, mentioned that they had already scaled Airbnb's data team to 30+ engineers. That's a $5M+ annual investment on headcount alone.



Keen IO

Keen IO is an event data platform that my team built. It provides big data infrastructure as a service to thousands of companies. With APIs for capturing, analyzing, streaming, and embedding event data, we make it relatively easy for *any developer* to run world-class event data architecture, without having to staff a huge team and build a bunch of infrastructure. Our customers *capture billions* of events and *query trillions* of data points daily.

Although a typical developer using Keen would never need to know what's happening behind the scenes when they send an event or run a query, here's what the architecture looks like that processes their requests.



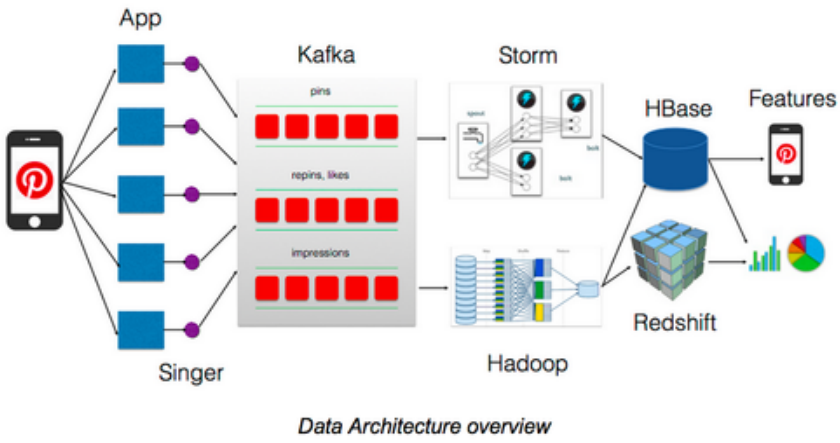
Keen IO Event Data Platform

On the top row (the ingestion side), load balancers handle billions of incoming post requests as events stream in from apps, web sites, connected devices, servers, billing systems, etc. Events are validated, queued, and optionally enriched with additional metadata like IP-to-geo lookups. This all happens within seconds.

Once safely stored in Apache Cassandra, event data is available for querying via a REST API. Our architecture (via technologies like Apache Storm, DynamoDB, Redis, and AWS lambda), supports various querying needs from real-time data exploration on the raw incoming data, to cached queries which can be instantly loaded in applications and customer-facing reports.

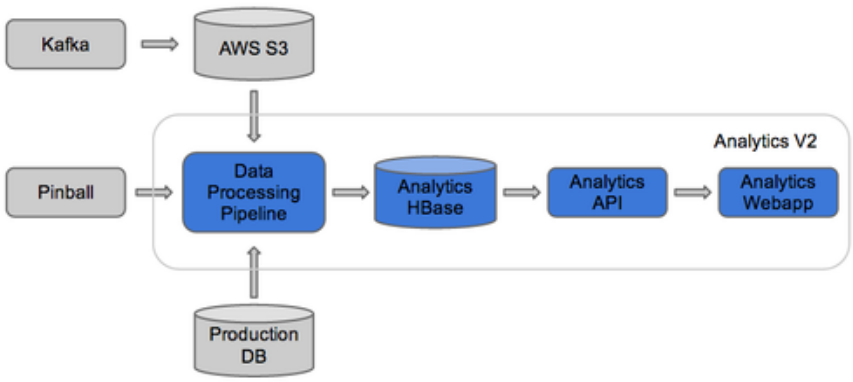
Pinterest

Pinterest serves over 100M MAU doing over 10B+ pageviews per month. As of 2015, they had scaled their data team to over 250 engineers. Their infrastructure relies heavily on Apache Kafka, Storm, Hadoop, HBase, and Redshift.

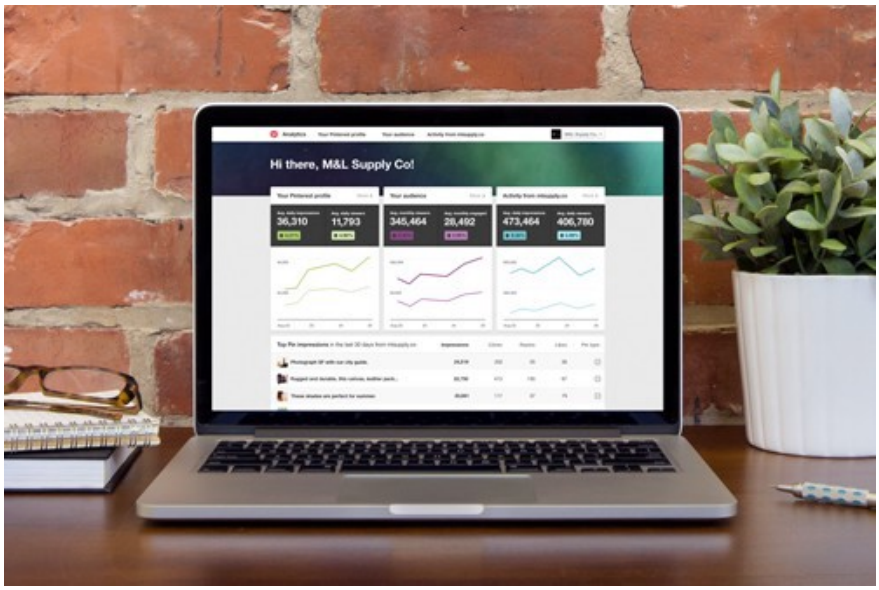


Pinterest Data Architecture Overview

Not only does the Pinterest team need to keep track of enormous amounts of data related to Pinterest's customer base. Like any social platform, they also need to provide detailed analytics to their ad buyers. Tongbo Huang wrote "Behind the Pins: Building Analytics at Pinterest" about their work revamping their analytics stack to meet that need. Here's how they used Apache Kafka, AWS S3, and HBase to do it:



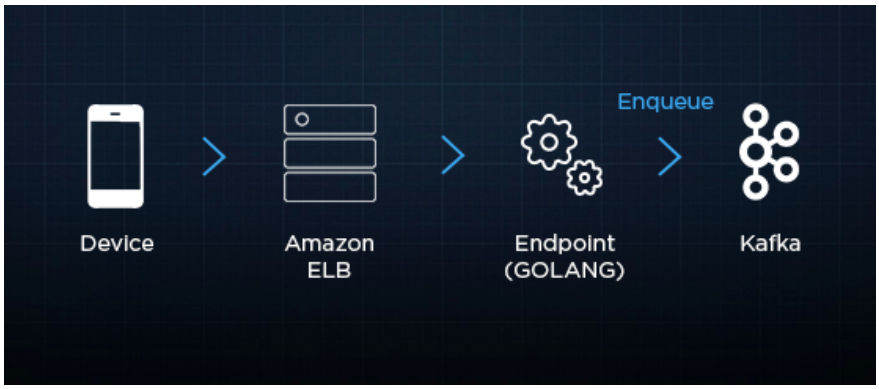
Data Architecture for Pinterest Analytics for Businesses



User View of Pinterest Analytics for Businesses

Twitter / Crashlytics

In Handling 5 Billions Sessions Per Day—in Real Time, Ed Solovey describes some of the architecture built by the Crashlytics Answers team to handle billions of daily mobile device events.

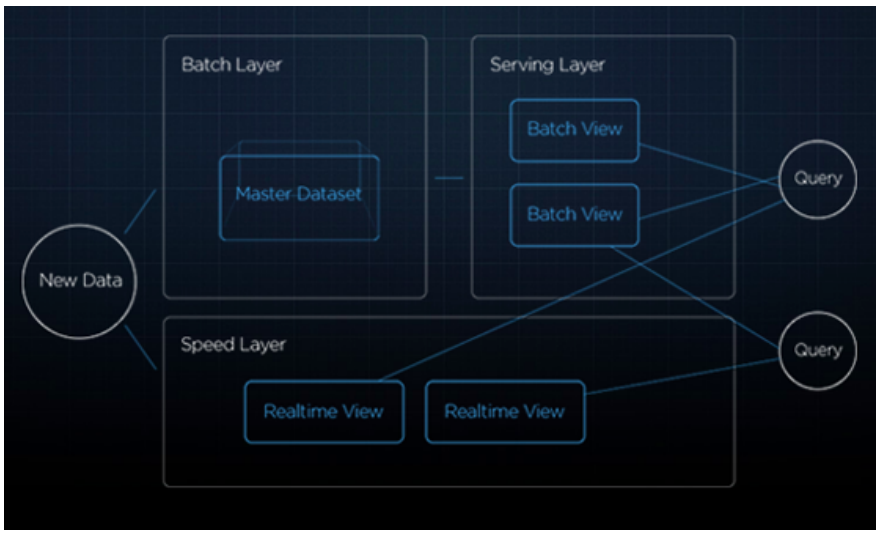


Event Reception

Archival

Batch Computation

Speed Computation



Combined View

Thank You

Thank you to the collaborative data engineering community who continue to not only *invent* new data technology, but to open source it and write about their learnings. Our work wouldn't be possible without the foundational work of so many engineering teams who

have come before us. Nor would it be possible without those who continue to collaborate with us day in and day out. Comments and feedback welcome on this post.

Special thanks to the authors and architects of the posts mentioned above: Steven Wu at Netflix, Martin Traverso at Facebook Presto, AirbnbEng, Pinterest Engineering, and Ed Solovey at Crashlytics Answers.

Thanks also to editors Terry Horner, Dan Kador, Manu Mahajan, and Ryan Spraetz .