Notes and Photos from QCon SF

Author: Deep Singh

Claps: 36

Date: Oct 16

We went to QCon San Francisco again this year. Here are a few thoughts, a list of talks we liked, and photos!

Eleven of the 18 of us who attended. From left to right: Allen, Himnish, Patrick, Klaida, Katherine, Noah, Kenny, Ronnie, Jonathan, me, Mayur.

QCon is a place where tech leaders discuss ideas and industry practices spread. I was lucky to be at QCon for my second consecutive year. You have to go to a good conference to really appreciate how much you can absorb in a few days, not to mention getting to know the latest research, appreciating academic heroes, and engaging in fun debates that refine your thinking.

The biggest learning I took away from QCon this year is that engineers should take a step back from day-to-day work more often and look at the big picture. Big picture here means the vision for your career, your team, and maybe for the company. At Klaviyo, our job as software engineers is busy with lots of priorities and projects. So, as with engineers at most companies, it seems difficult to carve out time for the big picture and for research. But, this doesnâ€™t have to be a big, monolithic effort done once a year or once a quarter. Itâ€™s better to do it as often as once a week.

That was the idea behind one of the most inspiring talks for me: How to Get Tech-Debt on the Roadmap by Ben Hartshorne. He suggests documenting a list of ideas for improvement, sorted by impact and effort, and committing yourself to regularly adding, updating, and trimming the list. Each item can be subdivided into multiple steps. This approach provides a common ground for starting brainstorming, prioritizing work when there is available bandwidth, and making incremental improvements when feasible. In time, some of the ideas may become critical, more impactful, or an easier task. Quoting Kidlinâ€™s Law, â€œIf you write the problem down clearly, then the matter is half solved.â€

At first I was thinking that his advice was no different than maintaining tech debt stories in an issue tracking system. But as he walked through examples, I realized there was a profound difference. By committing yourself to revisiting an evolving document every week, you force your brain to take a step back and see the bigger picture, and when it is time to add tech debt tickets into your issue tracking system, you can do so more strategically. Benâ€™s advice was something I took away from QCon this year and will try to personally implement moving forward.

QCon was also a great opportunity to network and spell the virtues of Klaviyo to fellow professionals. Way more people knew our company name this year, no doubt from all our recent news and the IPO, and I felt a sense of pride when people recognized us. Beyond that it was also nice to spend downtime with my fellow Klaviyos.

Talks

Having a bunch of us around meant we could cover multiple tracks at the same time. We collaborated in slack during the talks which also gives us searchable notes for posterity. Here are a few of the talks we enjoyed with links to the abstracts and brief summaries from our internal notes.

From Mainframes to Microservices â€” the Journey of Building and Running Software **â€”** No one size fits all. Revisiting the long-term trend of moving from mainframes to serverless with the thesis that this is slowing down. Increasing tendency to find specialized hardware specifically suited for certain jobs.

LIquid: A Large-Scale Relational Graph Database â€” Showcase of LinkedInâ€™s in-house graph database powering all relations graphs there.

Coding With Cody: How AI Is Revolutionizing Software Development â€” Prediction from here is that within five years, 99% of code will be written by AI with developers as consultants. How can we leverage this?

Relational Data at the Edge â€” Case study of successful deployment of relational data using Postgres at the edge with disk level replication. Covered various challenges on the way.

Streaming Databases: Embracing the Convergence of Stream Processing and Databases â€” Streaming databases can be considered to be an OLAP alternative for streams. Provides real time analytics on the data in addition to stream processing. Also, having optimizer and query plan engines help with learning curves and leads away from costly mistakes.

Redesigning OLTP for a New Order of Magnitude â€” Traditional/prevalent OLTP solutions (mysql, postgres, sqlite) are all more than 30 years old and are more focused on general processing, with greater emphasis on reading/querying than transactions/writing. Talk was about the TigerBeetle project, a modern approach to OLTP databases. Ideas: 1 network hop === 10/100 or 1000s of DB operations; no memory allocation at the runtime; LSMs rather than B-trees:

LSMs a new concept only being researched in detail in last decade or so; compactions done as JIT compactions rather than [scary compactions from C*]; separate storage structure for mutable and immutable data if housed together.

Exciting Web Ahead with WebGL/WebGPU â€” Letâ€™s Build a 3D App w/ Three.js â€” Live, interactive session where we built a clone of 2048, in 3D, using three.js and accelerated using WebGPU.

Incremental Data Processing with Apache Hudi â€” A case study about Uberâ€™s usage of Hudiâ€™s stream processing to enable incremental processing rather than rewriting the whole data.

Sleeping at Scale â€” Delivering 10k Timers per Second per Node with Rust, Tokio, Kafka, and Scylla â€” A highly scalable scheduler written using a simple mechanism with timers pulled for a window and expired as needed. Sounded uncannily like Klaviyoâ€™s flows scheduler.

Designing Fault-Tolerant Software with Control System Transparency â€” If certain inconsistencies are acceptable, we donâ€™t need to over engineer around it. Only having backups is not enough (even in single DBs), restore strategy needs to be verified and practiced. (No kidding!)

How to Get Tech-Debt on the Roadmap â€” Always more work than can be done. From an engineer POV, true even if we get a nice continuous stream of groomed work. Will there be any break to work on tech debt? From product POV, there is so much pending work, when will it be picked by engineering?

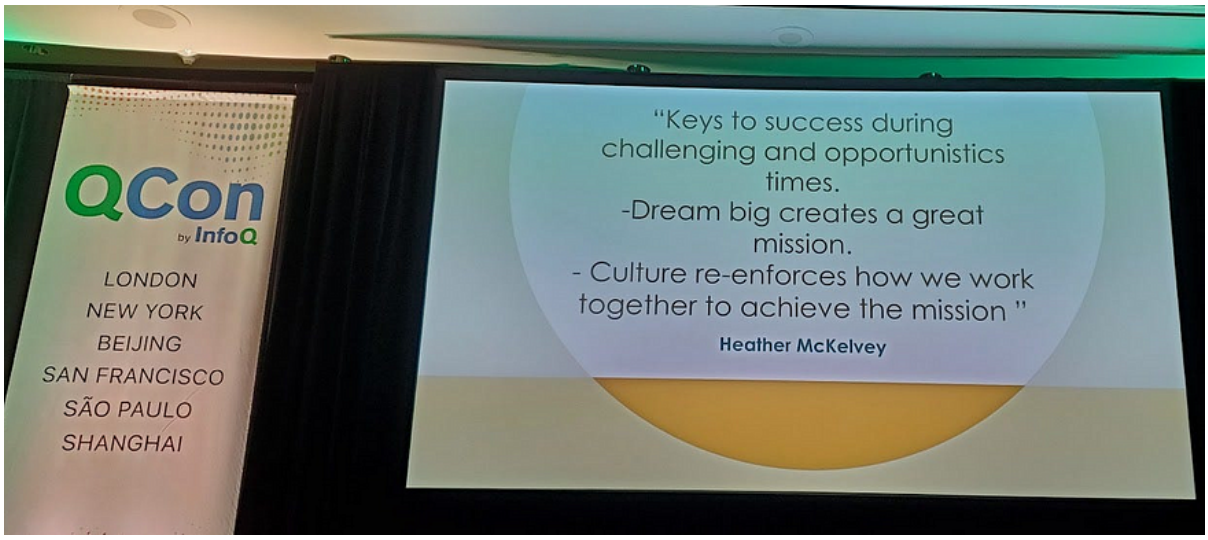
Streamlining Cloud Development with Deno â€” Ryan Dahl, creator of Node.js, talks about the future of technology, in particular the web and what heâ€™s been working on in Deno to solve those problems. Live coding where we globally deployed a simple deno server complete with KV database and an at-least-once queue â€” all in about 10 minutes. Very few things can be said definitely about the future of technology. Fair assumption though â€” web will still be here for at least next 5 or 10 years. Corollary: javascript will be here as well.

Hydration and Lazy-Loading Are Incompatible â€” Lazy loading only works for components NOT ALREADY in the render tree. If theyâ€™re in the render tree, lazy loading is useless. Trampoline functions are often bigger than what theyâ€™re loading and are often not worth it. Qwik is a framework that does lazy by default and skips hydration.

Combating AI-Generated Fake Images with JavaScript Libraries â€” The crypto library is useful out of the box for many cryptography applications for preventing fake images. Often we think about it as attackers vs defenders, where we need publishers to verify the photos they take using cryptography (e.g. private key signing using a trusted timestamping service) to be defensive.

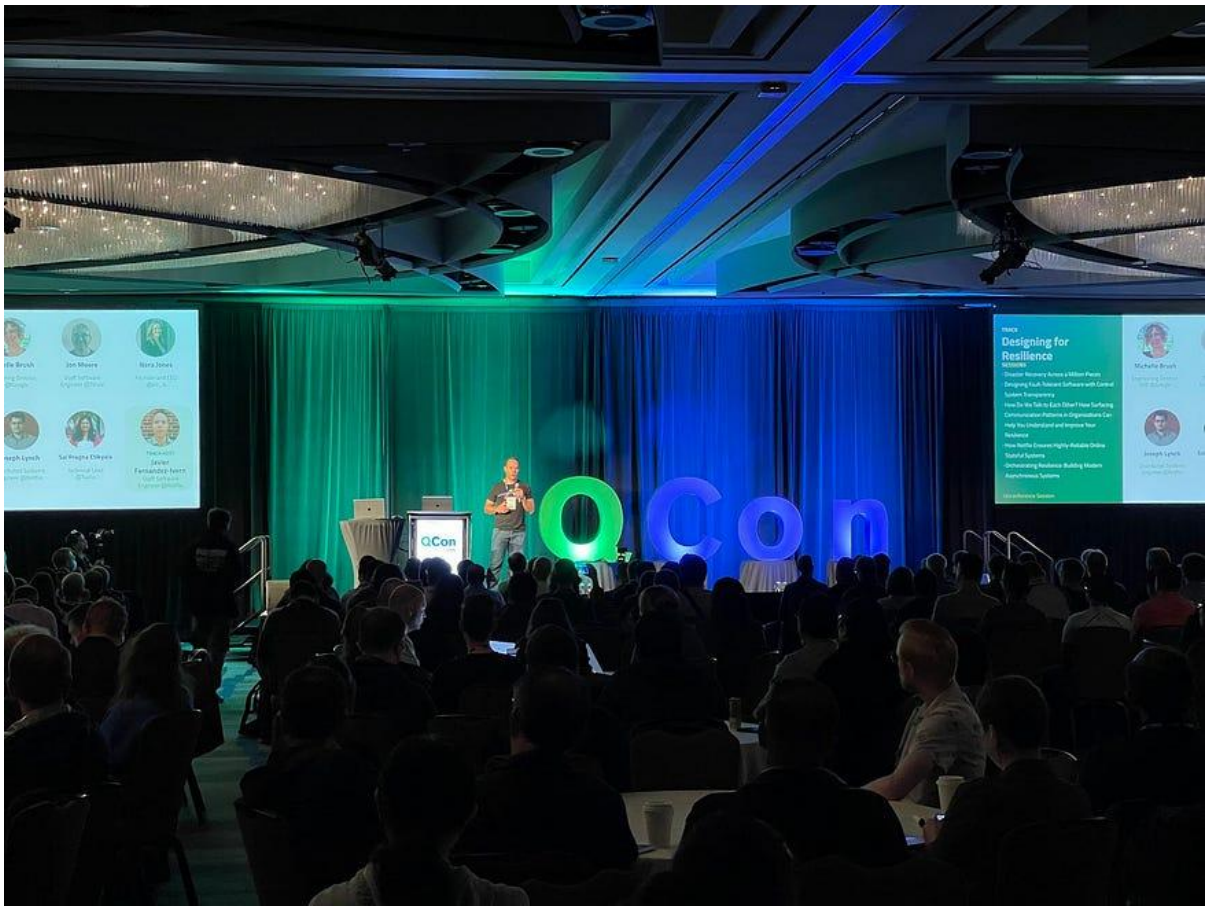
Building Organizational Resilience Through Documentation and InnerSource Practices â€” Ideals of documentation: useful, updated, relevant, available, discoverable. (We do pretty good!) â€œIf you write the problem down clearly, then the matter is half solved.â€ â€” Kidlinâ€™s Law (from a character in a James Clavell novel).

Mission, Culture, and Values: Using Them to Guide Your Company Through Good and Challenging Times â€” Powerful keynote to end the conference by Heather McKelvey from LinkedIn. She started with â€œI have been in this industry for longer than most of you have been alive.â€ Shared her experiences during major upheavals of the industry within the last 42 years, and how to weather the storm during tough times and make the best during happier times.



Slide from the closing keynote

Photos



Kickoff for talks in the morning

Networking during coffee breaks between sessions

Team â€œbondageâ€ event at an Escape from Alcatraz escape room. Thatâ€™s, front to back, Noah, Apra, Klaida, Katherine, Praveen.

Another photo from the escape room. Back row left to right: Kenny, Nithin, Noah, Jonathan, Praveen, Jonathan. Front row left to right: Himnish, Katherine, Klaida, Apra.