



TEN BEST PRACTICES

Sequoia's Microservices Summit Takeaways

MATT MILLER | @MCMILLER00



Matt Miller
@mcmiller00



Follow

Excited for today's [@sequoia](#) [#microservices](#) summit. 70 of the senior most thought leaders building microservices





Microservices Summit 2016

AMGEN
APCERA
CAA
CITI
COMMONWEALTH
BANK OF AUSTRALIA
CONFLUENT
CRATE IO
CUMULUS
DATAWIRE
DISNEY
DOCKER
DROPBOX
EDMUNDS
EVENTBRITE

EX-AMAZON
EX-GOOGLE
EX-GILT
GOLDMAN SACHS
HP
IBM
INSTACART
KISMATIC
LINKEDIN
MEDALLIA
METASWITCH
MICROSOFT
MINIO
MONGODB
MORGAN STANLEY

NASDAQ
NGNIX
OKTA
PAYPAL
QUALCOMM
REDLOCK
SALESFORCE
SARATOGA SPEED
SHIPPABLE
SPRINGPATH
SQUARE
STACKROX
SYSDIG
TALKDESK
THOUGHTWORKS

TWITTER
VERIZON
VMWARE
WELLS FARGO

INSIGHT 1

Determine if you really need microservices

Not every application is complicated enough to warrant being broken into microservices. Martin Fowler and Ryan Murray from Thoughtworks cite a “microservice premium” where in many use cases the complexity of microservices hampers the productivity of your team.

There comes a point when your application becomes very complex or your team begins to grow past 50-75 engineers that the benefits of this architecture begin to take off.

INSIGHT 2

Put your house in order

Continuous delivery and automation are more important than microservices. Small agile teams who can integrate their work frequently, at least daily, are an important precursor to microservices.

Being able to automate your systems and push code updates regularly are critical to deal with the complexity you will incur with this architecture.

INSIGHT 3

Appoint a master artisan

If you do not design and manage your evolution to microservices the result can be an uncontrollable sprawl. It is critical to have a person or a small team responsible for controlling architectural decisions and helping to ensure standards adoption. Google has a small team of artisans who understand how pieces fit together and help guide the creation of new services.

Alex Roetter from Twitter refers to this oversight as “applying the right amount of salt” You never want to overpower the great innovation that is coming from the bottom up.

INSIGHT 4

Microservices should map to business processes

Teams should have bounded context and systems should follow the ordinary flow of business. Melvin Conway first came up with this principle in 1967 and it holds true today. When your services are not directly mapped, it makes troubleshooting or re-architecting in the future far more difficult.

Organizations which design systems...are constrained to produce designs which are copies of the communication structures of these organizations.

Melvin G. Conway – 1967

INSIGHT 5

Start every new product as a monolith

It is impossible to know the best way to divide up a monolith until you can observe its usage. Once you have a sense for how your product will be used you can decompose from there.

Today there is a bit of trial and error on sizing your services. There may be an opportunity for someone to help with this process through software in the future.

INSIGHT 6

Move incrementally

Don't throw away the monolith. This can have disastrous results. Take one piece at a time and break it off. Once the piece is working, move to the next natural piece.

A few companies have found a more aggressive approach can lead to lost functionality and make diagnosing issues difficult.

INSIGHT 7

Create a shared repository

Consider creating one large shared repository of all services for teams to use in application development. You don't want to have two or three active versions of a common service in use.

Your artisan(s) should help in managing this repository.

INSIGHT 8

Apply more comprehensive monitoring

More pieces to manage telemetry very important. The microservices monitoring landscape is very fragmented. There is not a clear winner and some companies are building their own products.

Microservices require a much more comprehensive monitoring effort than you have had with your monolith.

INSIGHT 9

Improve security and governance

More surfaces and complexity increase the need for security and governance. Consider how you will authenticate who can speak to whom and identify illegitimate traffic.

Who has the authority to work on certain services? Can all services be used for all tasks in your company? How are shared services billed or managed?

INSIGHT 10

Reap the rewards

When successfully implemented, microservices delivers huge increases in the speed and agility in which companies can build and deploy software. The cost required to deliver an application is much less and your systems will become more resilient. Development time can go from months to weeks.

*Companies like Airbnb, Disney, Dropbox, GE, Goldman Sachs, and Twitter have seen development lead times cut by as much as **75%** when using microservices.*

FOR MORE ON MICROSERVICES AND OUR LATEST
MICROSERVICES ECOSYSTEM MAP VISIT:

seq.vc/microservices

