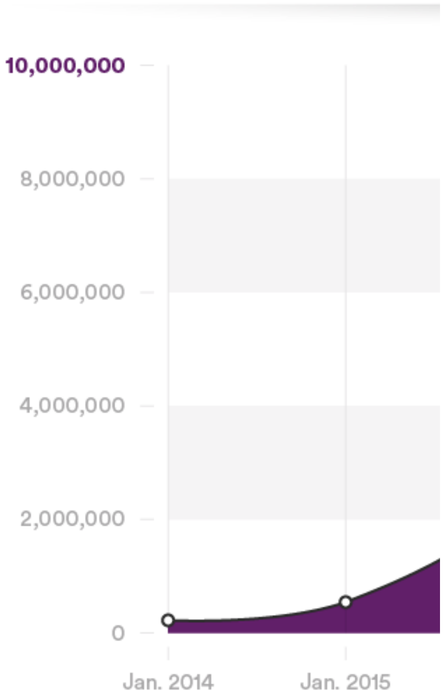
Scaling Slack Infrastructure

🚀

Julia Grace
Senior Director of Engineering
@jewelia

Phase 0: 2015

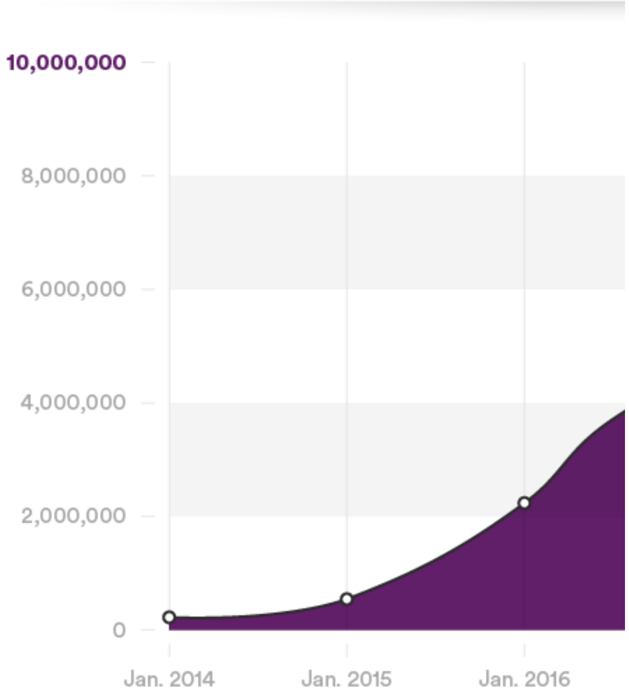
Daily Active Users



~2.5M Daily Active Users

Phase 1: 2016

Daily Active Users



~4M Daily Active Users

Phase 1: 2016

Slack was originally designed for teams < 150ppl.

You make very different architectural decisions when you're building for a team of 100 people vs 500,000.

Before August 2016 we had no Infra team.

Original infrastructure built for Glitch worked very well in 2014/2015.

~150 Engineers total.

Infrastructure investments would come secondary to feature work.

Things were starting to break in strange, unusual
ways.

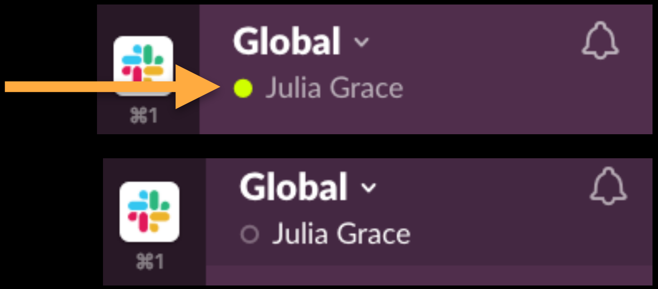
Phase 1: 2016

Example: User Presence

Green dot indicating online/away/offline.

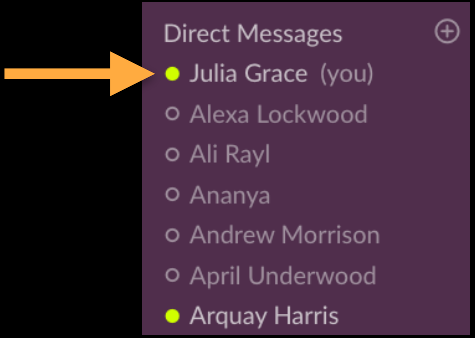
Very few people notice it, unless it's broken (people expect it to "just work").

Apps and bots are always online.



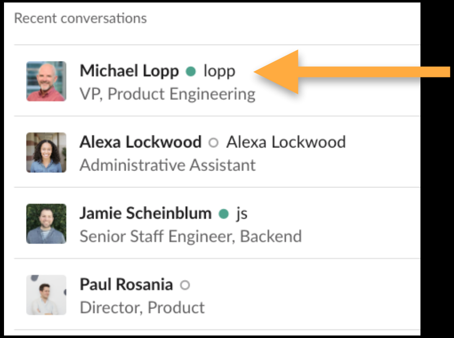
Global ⌄
⌘1
● Julia Grace

Global ⌄
⌘1
○ Julia Grace



Direct Messages ⊕

● Julia Grace (you)
○ Alexa Lockwood
○ Ali Rayl
○ Ananya
○ Andrew Morrison
○ April Underwood
● Arquay Harris



Recent conversations

Michael Lopp ● lopp
VP, Product Engineering

Alexa Lockwood ○ Alexa Lockwood
Administrative Assistant

Jamie Scheinblum ● js
Senior Staff Engineer, Backend

Paul Rosania ○
Director, Product

slack

@jewelia

Phase 1: 2016

User Presence

Initially broadcast all changes to all users (e.g. "Julia Grace is away") to the whole workspace: O(n^2).

Presence was ~80% of all web socket traffic. 😰

Peak volume in late 2016: 16 million messages/minute over web socket.

Presence messages: 13 million/minute.

Rapidly transition from broadcast to publish/subscribe.

There were many organizational challenges as well.

💥

Phase 1: 2016

How to build engineering-led org in a product-led company?

Would we be able to get headcount, budget?

How to communicate the value of we are doing to non-technical audiences?

How do we interface with sales? 💥

Infrastructure as a competitive advantage.

https://www.flickr.com/photos/pocheco/14833391966

Phase 1: 2016

Start internal evangelism on day #1.

I went on an internal PR campaign: *Why our work was important, why we needed to continually invest in infrastructure. Make work very visible to execs in other functions.*

Followed existing company process.

We did planning, status reporting, etc. at the same cadence and in the same meetings as product engineering. Don't try to start a new group and invent new process.

Identify executive sponsor.

Phase 2: 2017

Phase 2: 2017

Technology landscape.

Hack/PHP monolith on backend, JavaScript with no libraries on frontend.

1 service: presence and real-time messaging.

Building a second service: Go caching service.

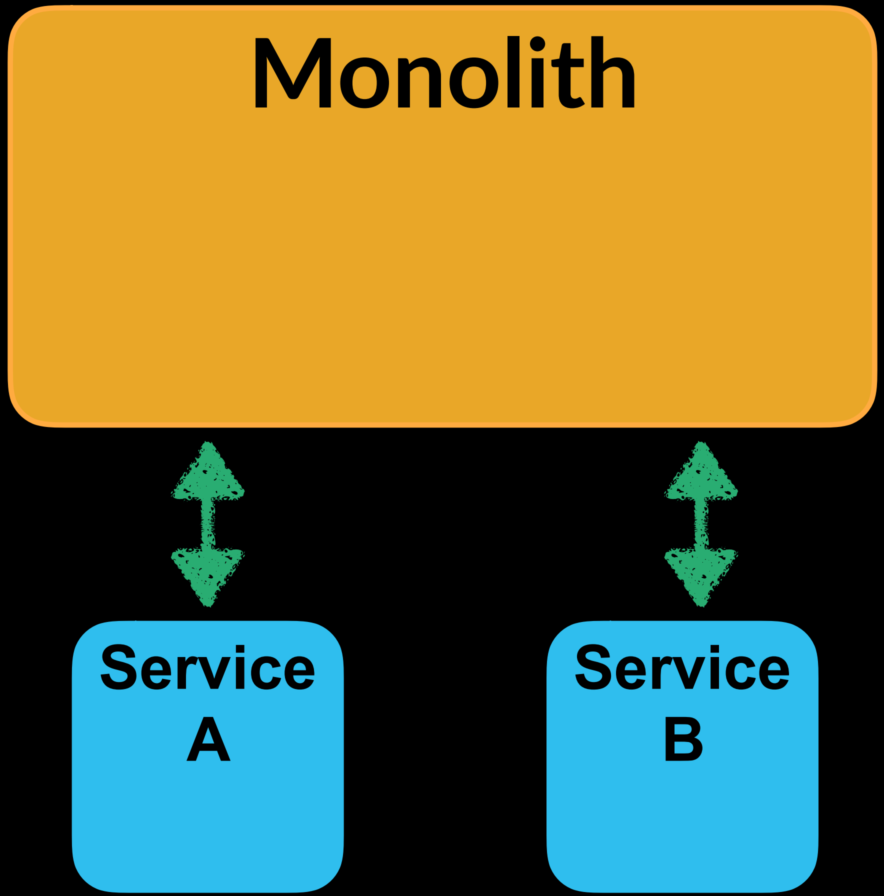
These bespoke services each had to handle rate limiting, traffic management, deployment.

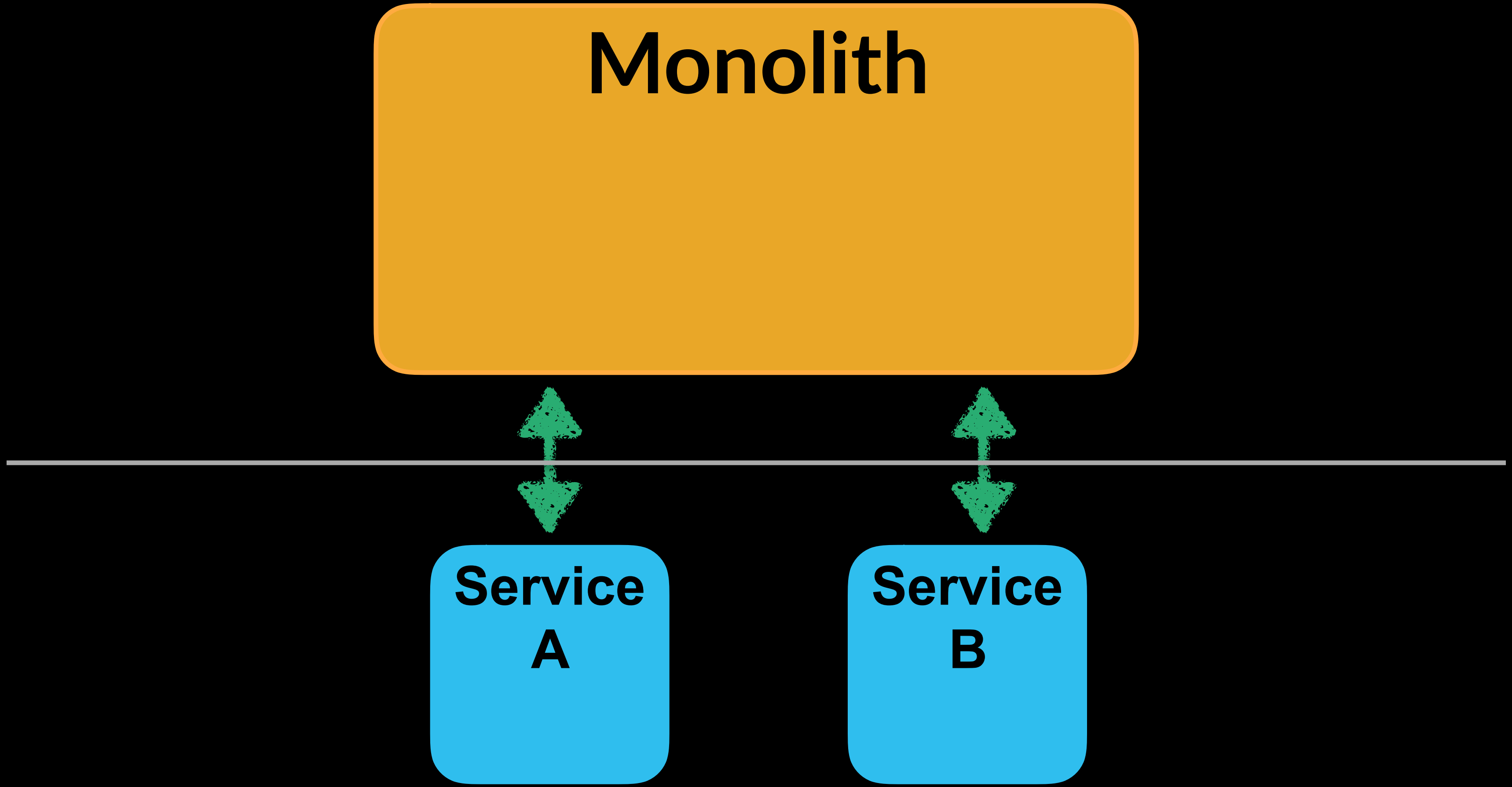
Phase 2: 2017

It was time to change our DB sharding strategy.

MySQL sharded by team/workspace to Vitess sharded by various keys.

Worked great! Until we hit scaling limits, significant hotspots.

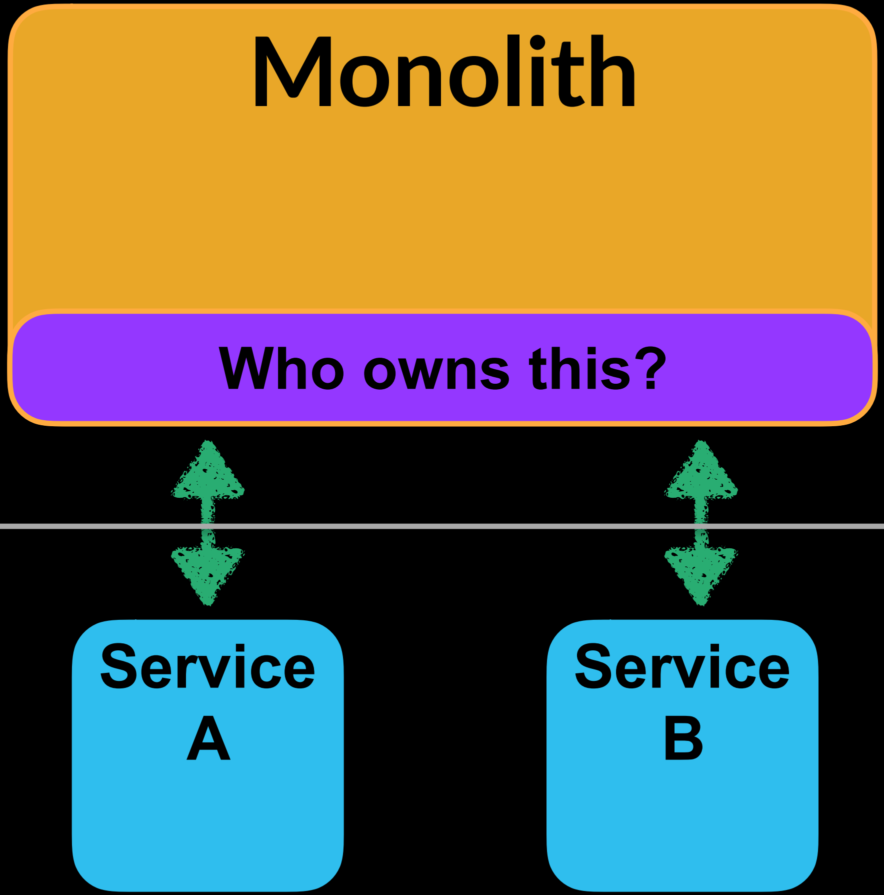

Monolith
Service A
Service B



Monolith

Service A

Service B


Monolith
Who owns this?
Service A
Service B

Communication Risk

The more technically complex, nuanced a problem is… 😺

Communication Risk

The more technically complex, nuanced a problem is…

The higher the communication risk.

Phase 2: 2017

Immense pressure to hire engineers.

Many human SPOFs (single points of failure) because team was so small.

Everyone was overextended and overcommitted.

We had to figure out how to hire Infra engineers. 💡

All our hiring processes were optimized around hiring generalists: frontend backend, iOS, Android, Ops.

We skills do we need and value? How do we test for those skills?

Phase 2: 2017

Decided to hire Infra engineering generalists.

Created a take home coding exercise designed to test:

1. *An understanding of servers, networking, and protocols.*
2. *An understanding of concurrency, performance, and resource constraints, and an ability to anticipate future issues and implement solutions.*
3. *An ability to write clear, easy to understand code, communicate your approach, and reason about tradeoffs that you have made.*

Phase 2: 2017

I wore so many hats. Too many hats.

Similar to my days as a startup CTO!

I was the Engineering Director and

Forming strategy, hiring managers and ICs, evangelizing the org.

…Product Manager and

Internal interface to Product Engineering/PMs building features, externally to customers with questions about the integrity of our infrastructure.

…Program Manager.

Running cross functional initiatives.

Phase 3: 2018

Phase 3: 2018

“0 to 1” was over. Now time for “1 to ∞”.

Reactive to Proactive.

Transition from few *teams* to an *org* in 3 offices.

Team nearly 100 engineers by end of year.

Now included Data, Machine Learning, Search Infrastructure
Many orders of magnitude better performance
Things were not breaking all the time.

Phase 3: 2018 😎

Services model matured significantly.

SLAs for services, consistent deployment processes, etc.
Mature incident response process.

Dividing into sub-teams made sense.

Data Stores & Cache Infra, Service Mesh & Web Serving, Distributed Messaging.

Phase 3: 2018

Hired Director Specialists…

Had to quickly learn how to hire senior leaders whose jobs you haven't done before. How to do this well: talk to a lot people who currently do the job you're trying to hire for, deeply understand the talent market.

and Product Managers… 📈

and did an acquisition.

Phase 3: 2018

Challenge: coherency across a large organization.

Example: overlap between Machine Learning and Frontend Infra was NULL.

Difficult to have a unified vision.

Stakeholders were each org were different for each part of the org; Data Infra organization worked closely with G&A (finance), Search Infra did not.

I should have done more re-orgs!

2016: 🙀

2016: 🙀

2017: 😅

2016: 🙀

2017: 😅

2018: 🚀

Today

Infra has been around for ~3 years

400M async jobs processed/day to 2.5B

3M DAU *(daily active users)* to 10M DAU

1M simultaneously connected users to 7.5M

10 to ~100 engineers in SF, NYC, YVR

Generalist (ICs, Managers) to specialists

1 amazing team

Thank You!