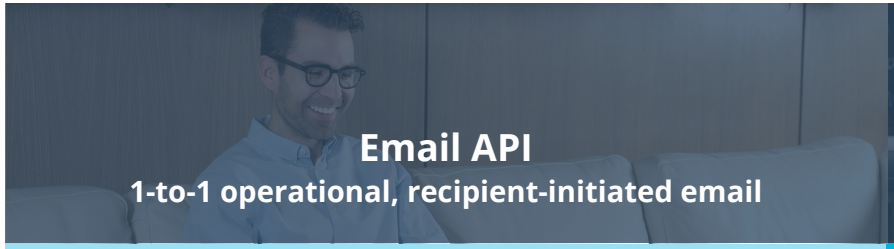
TWILIO SendGrid

Rearchitecting for Cloud Native
or...All We Changed Was Everything

J.R. Jasperson - CTO, Twilio SendGrid

About Twilio SendGrid

Built for Developers and Marketers



Email API

1-to-1 operational, recipient-initiated email

WAYS TO SEND

Marketing Campaigns

1-to-many promotional, sender-initiated email

WAYS TO SEND

EASY TO INTEGRATE

ANALYTICS AND REPORTING

SECURITY

SendGrid Email Platform

EASY-TO-USE USER INTERFACE

DISTRIBUTED CLOUD ARCHITECTURE

FLEXIBLE APIs

Expert Services and World Class Support

EMAIL DELIVERY SERVICES

ONBOARDING SERVICES

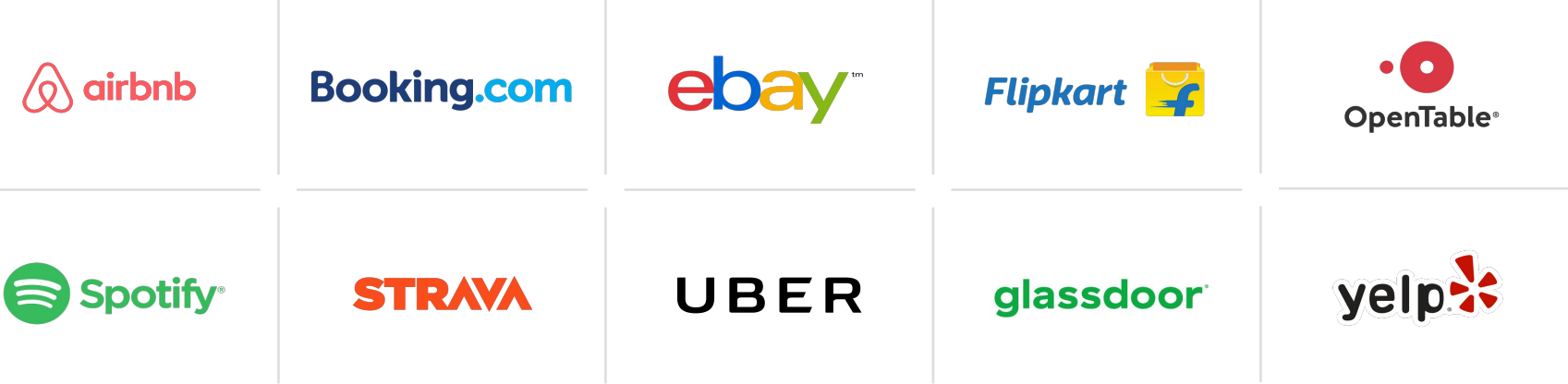
ONGOING EXPERT MANAGED SERVICES

Twilio SendGrid Snapshot

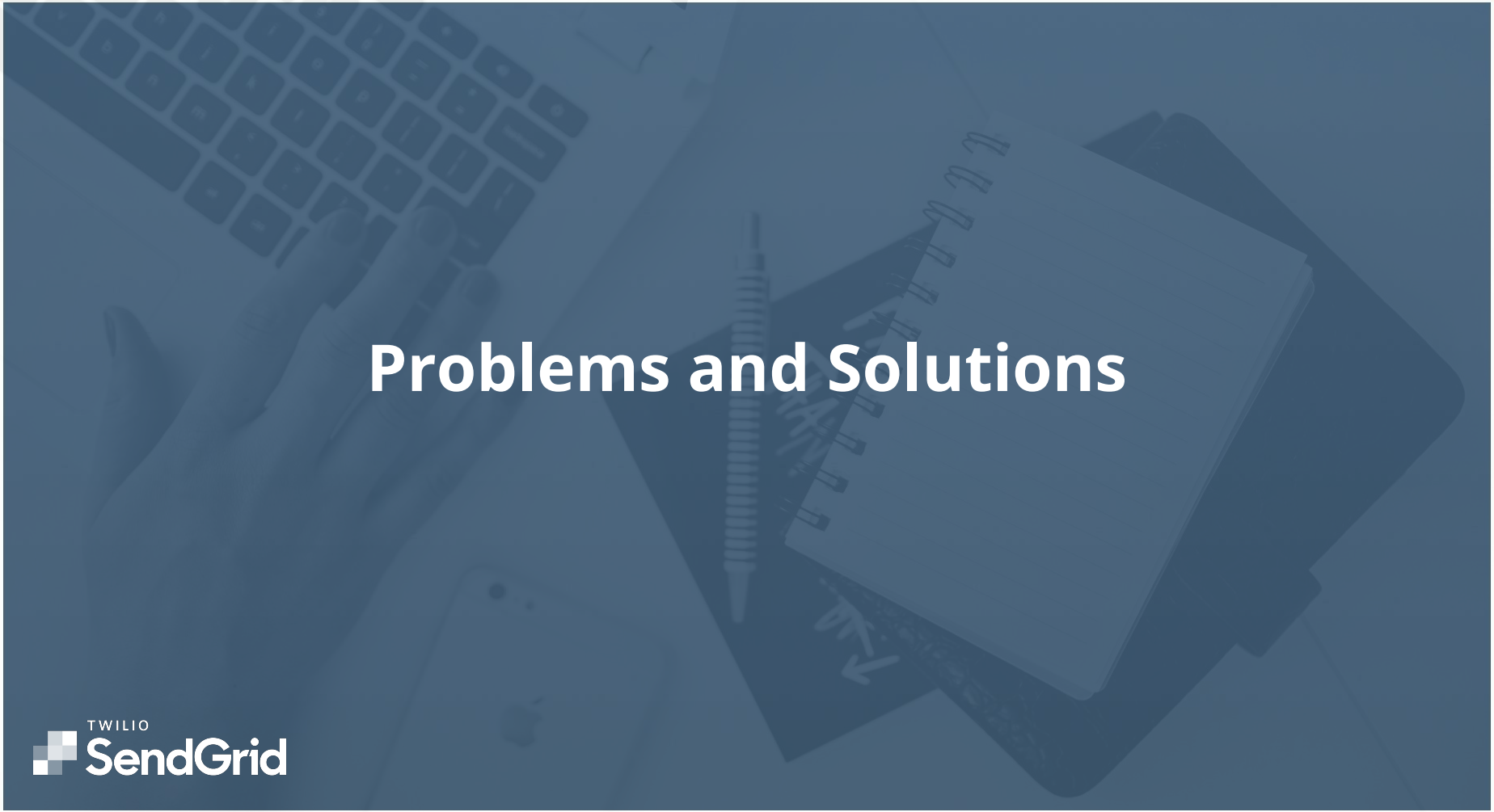
80K+ paying customers in 100+ countries

50B emails/month

99.999%+ uptime



Problems and Solutions



TWILIO SendGrid

Framing the Problem

- The initial architecture was beginning to show signs of strain
- Strategically shifting from self-managed colo’s to AWS
- Traditional email systems and software are based on legacy notions of infrastructure

Limitations and Solutions

Unbounded Failure Domains	→	Fault Isolation
Width Thrashing	→	Fixed Width
Fault Intolerant Storage	→	Durable Storage
Stateful Compute	→	Ephemeral Compute
Tight Coupling	→	Independent Scalability
Prone to Hotspots	→	Pull-Based Architecture

Systems Architecture: Scale Up to Scale Out

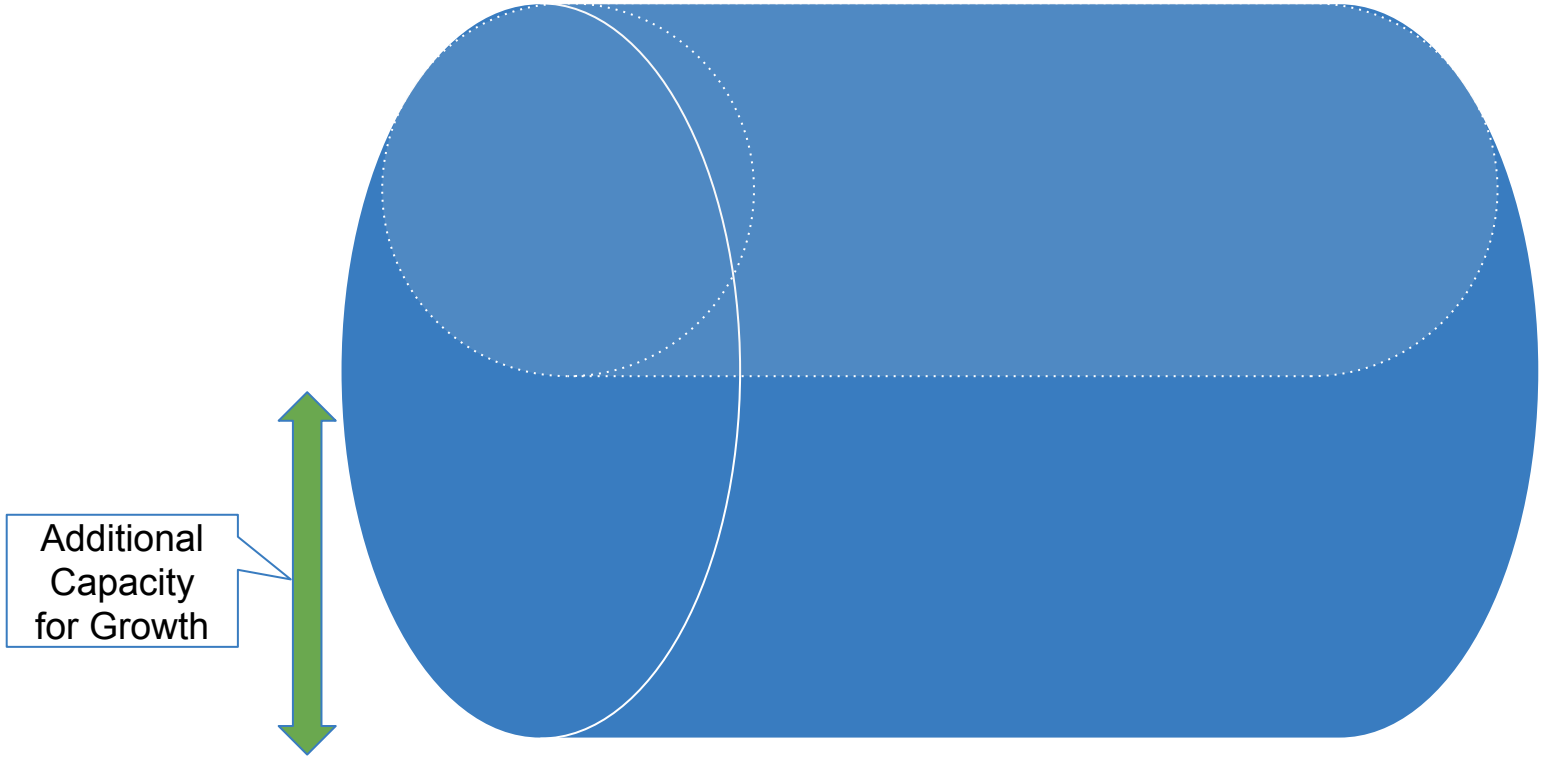
TWILIO SendGrid

Limitations and Solutions

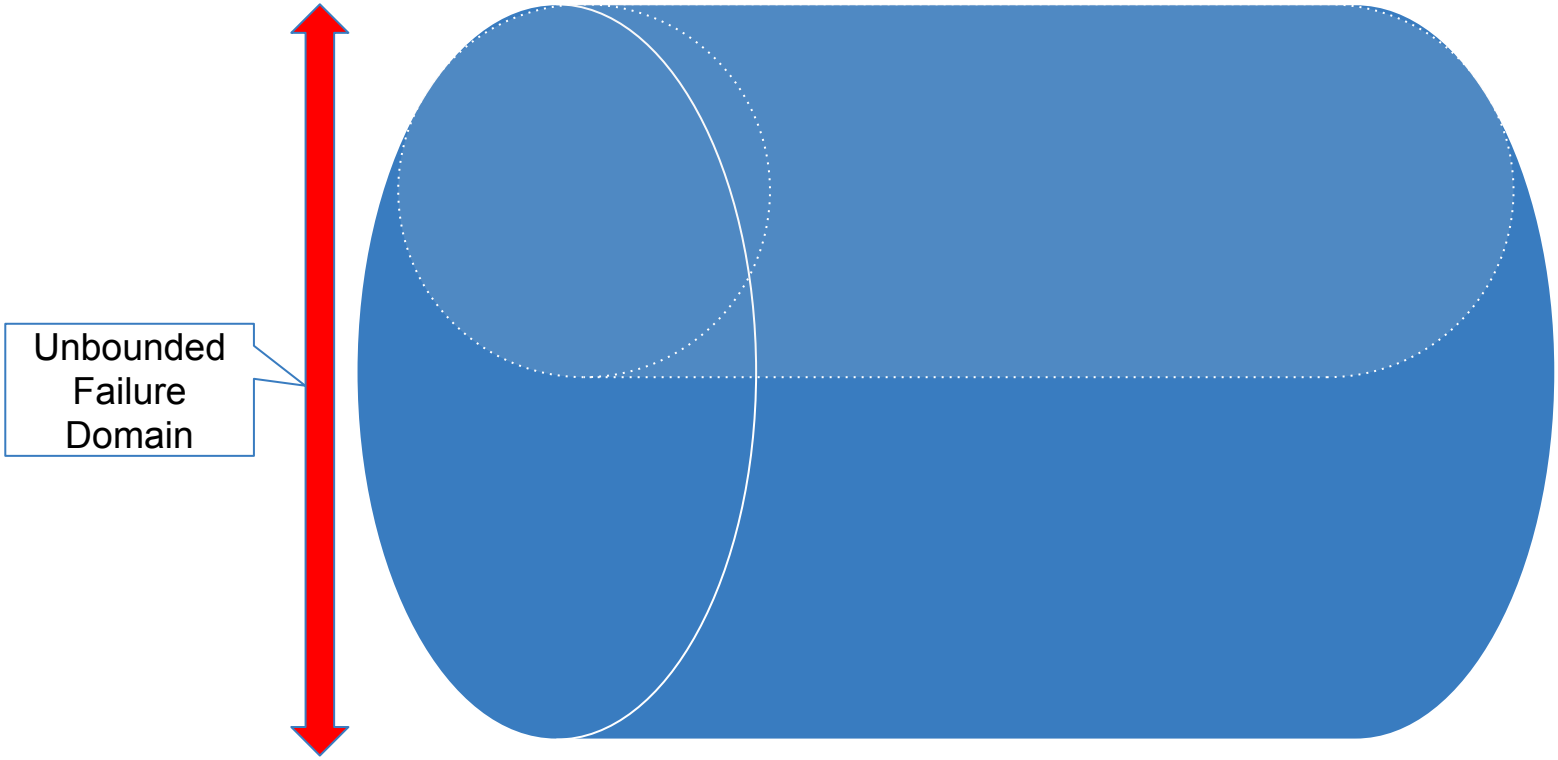
Unbounded Failure Domains →	Fault Isolation
Width Thrashing →	Fixed Width
Fault Intolerant Storage →	Durable Storage
Stateful Compute →	Ephemeral Compute
Tight Coupling →	Independent Scalability
Prone to Hotspots →	Pull-Based Architecture

Legacy Approach: Scale Up System

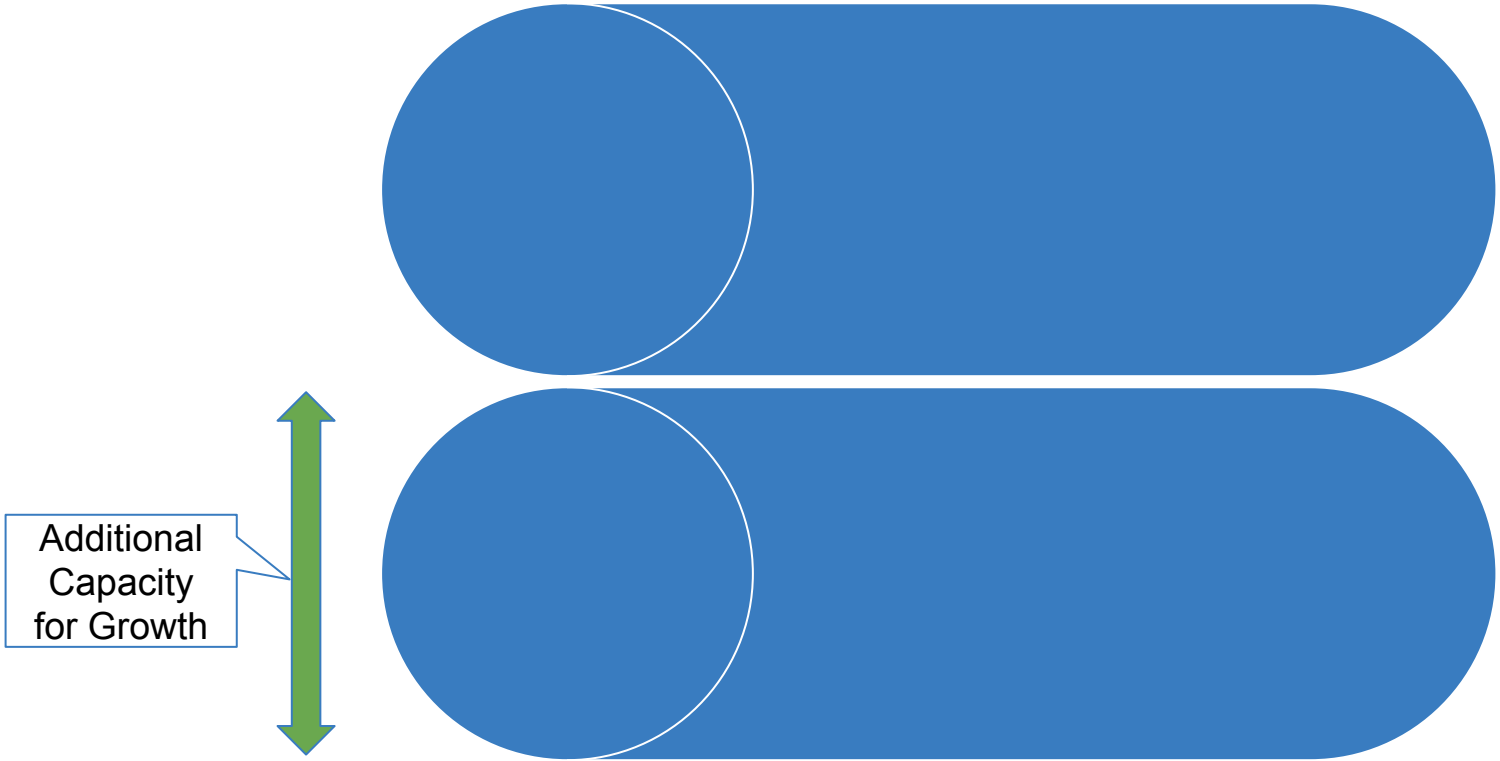
Legacy Approach: Scale Up System



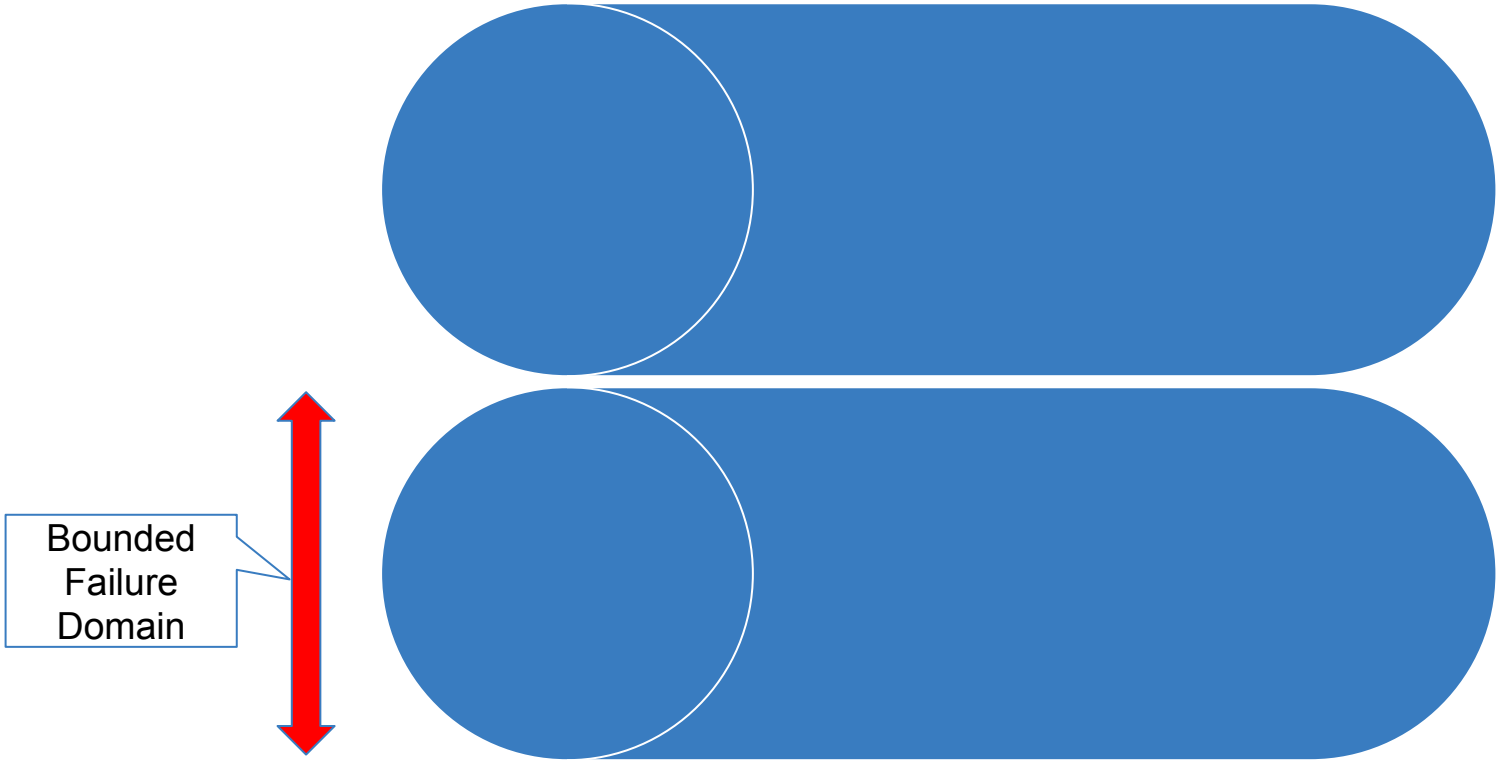
Legacy Approach: Scale Up System



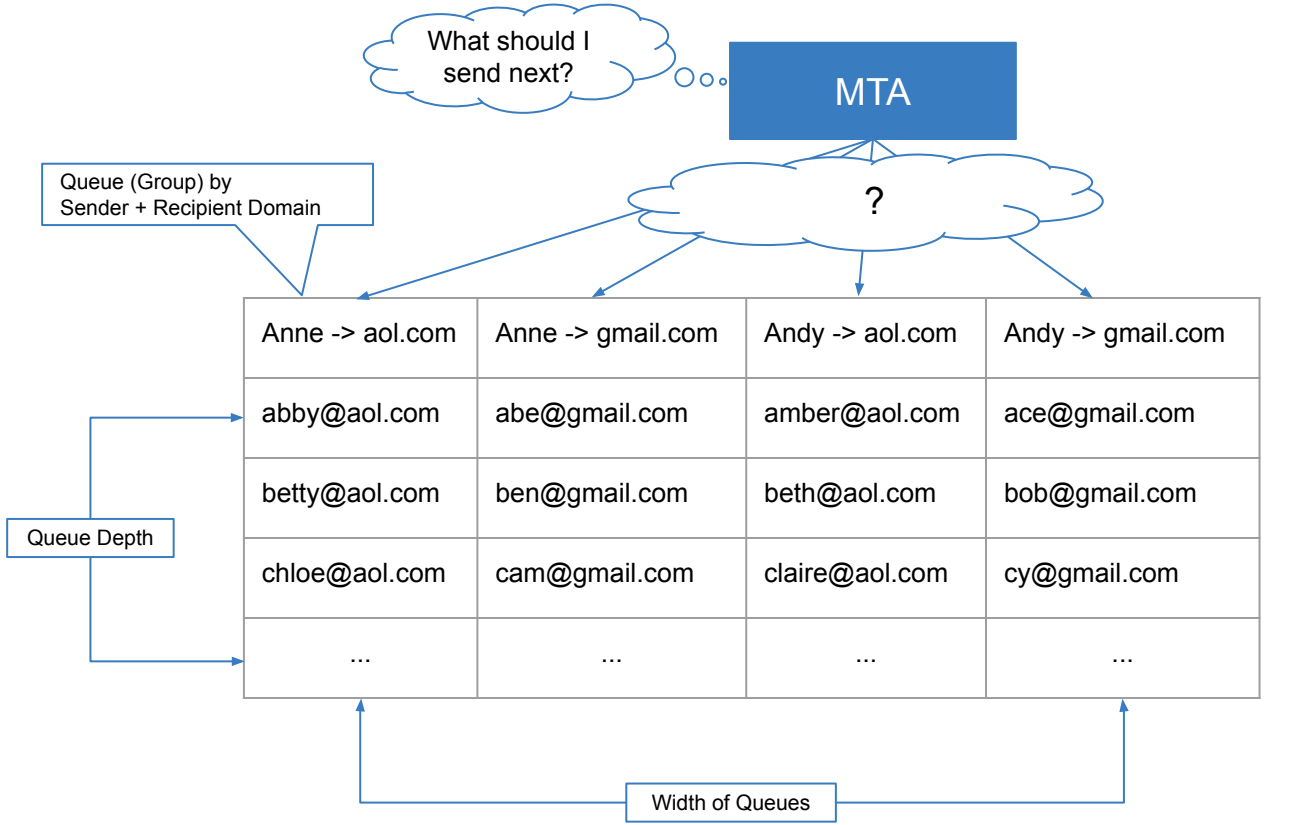
New Approach: Scale Out System



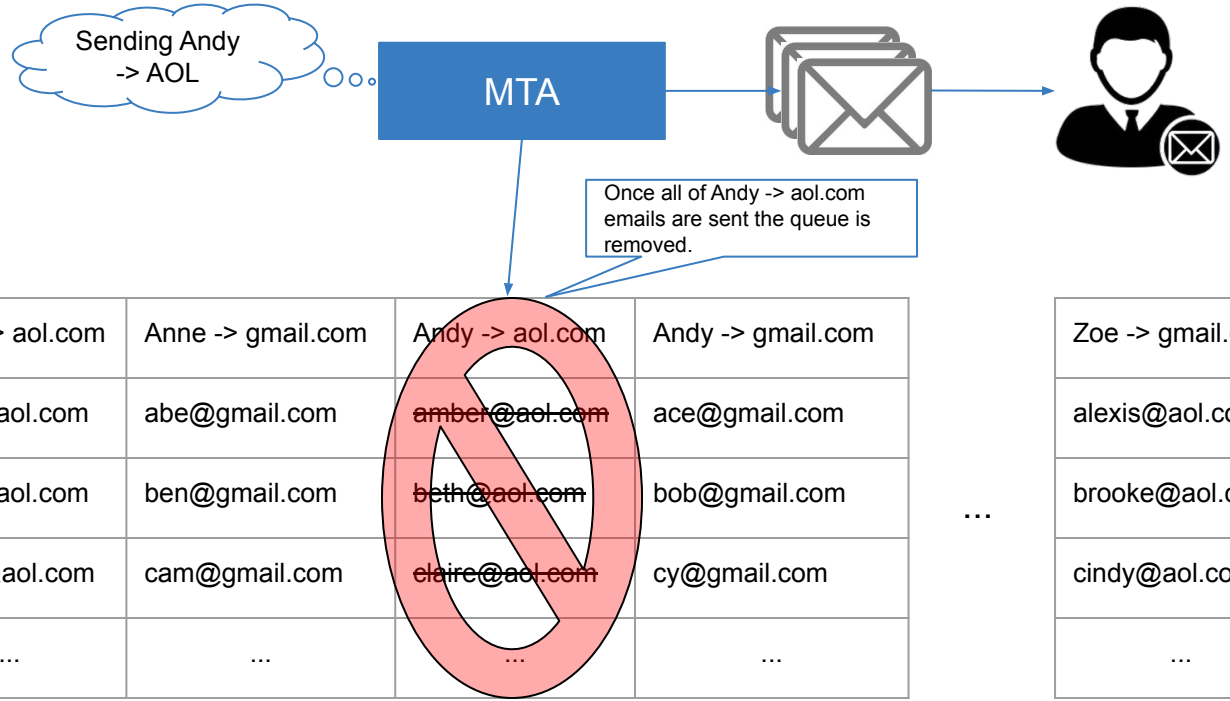
New Approach: Scale Out System



The Width Thrashing Problem

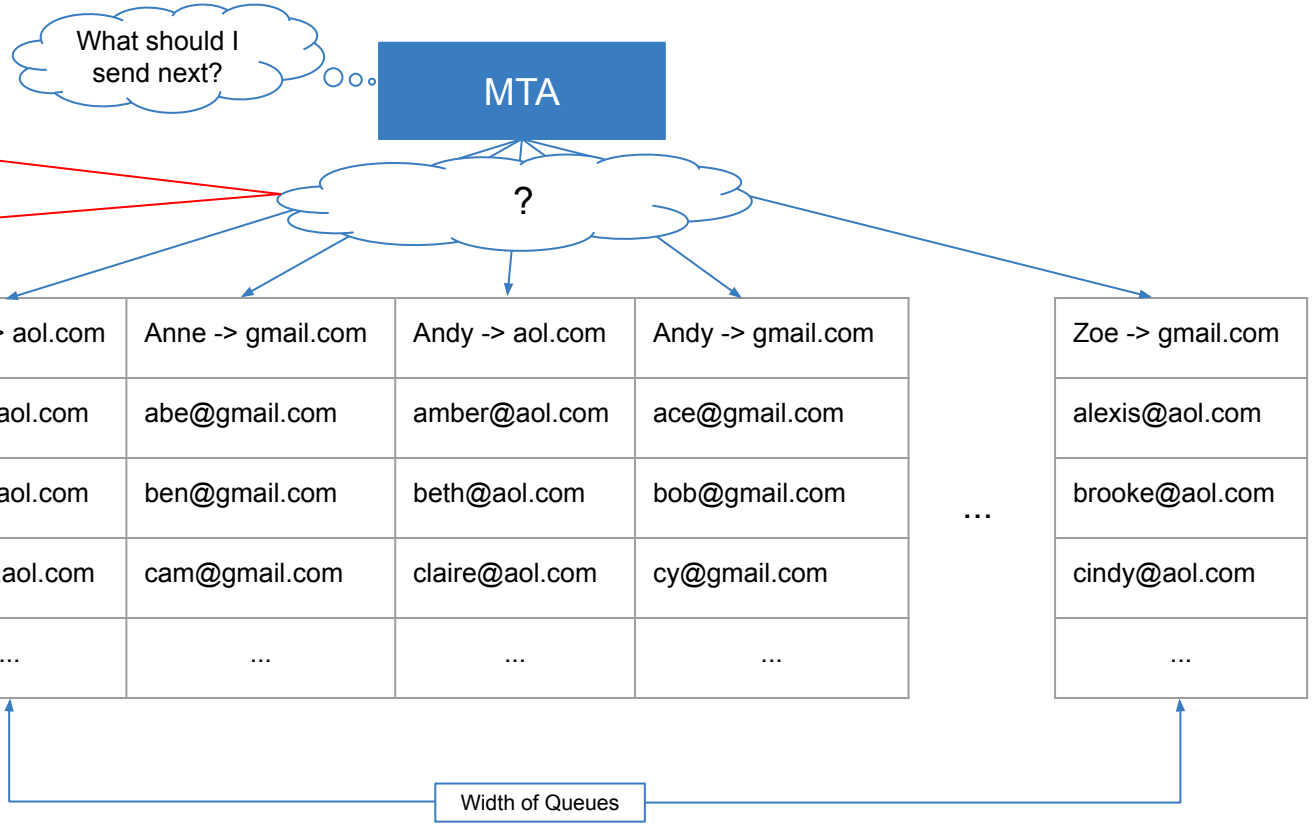


The Width Thrashing Problem (Cont.)

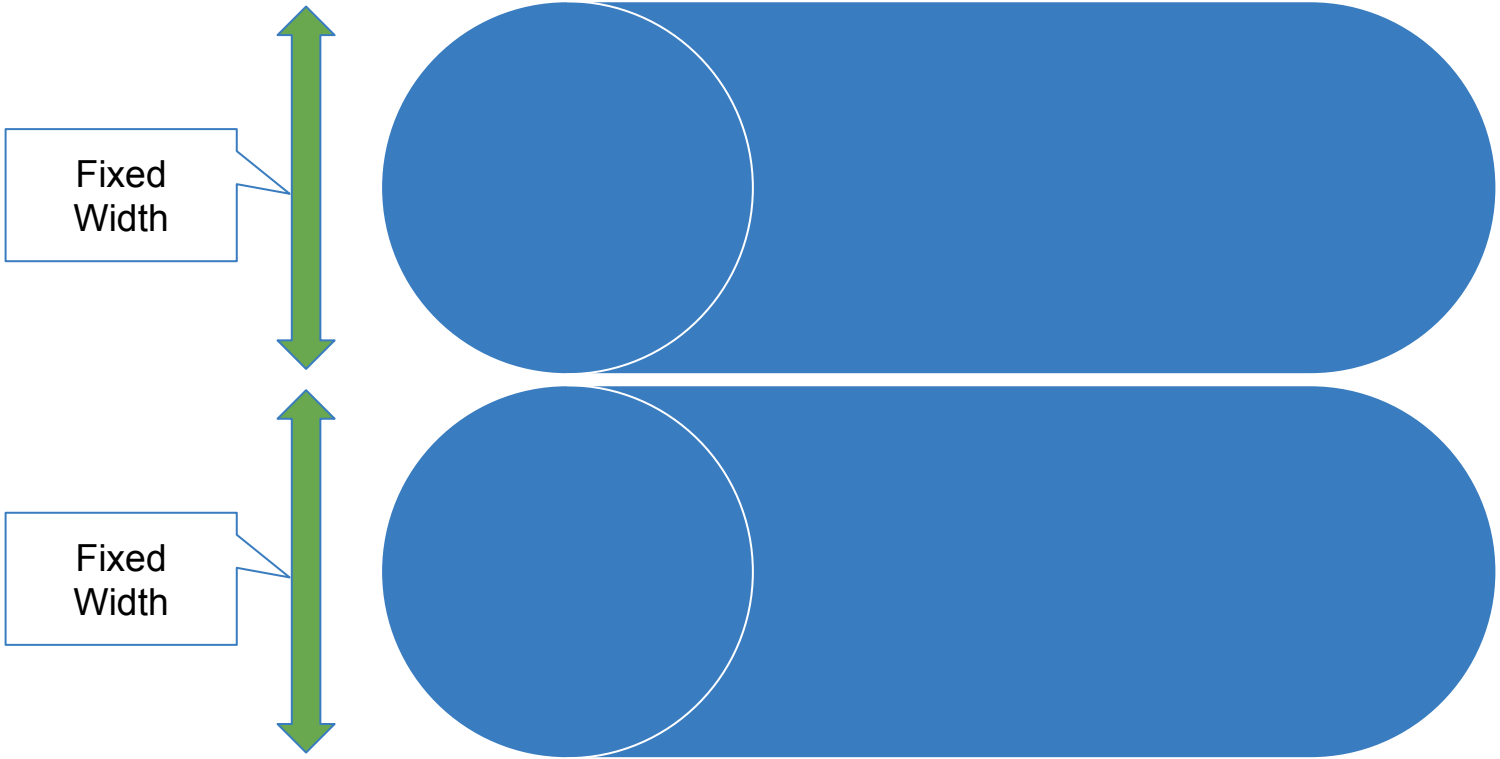


The Width Thrashing Problem (Cont.)

However, with unbounded customer growth the “width of queues” grows to a point of no return. The process spends increasing time determining what to send next and thus less time actually sending mail. This eventually puts it in an unrecoverable state.

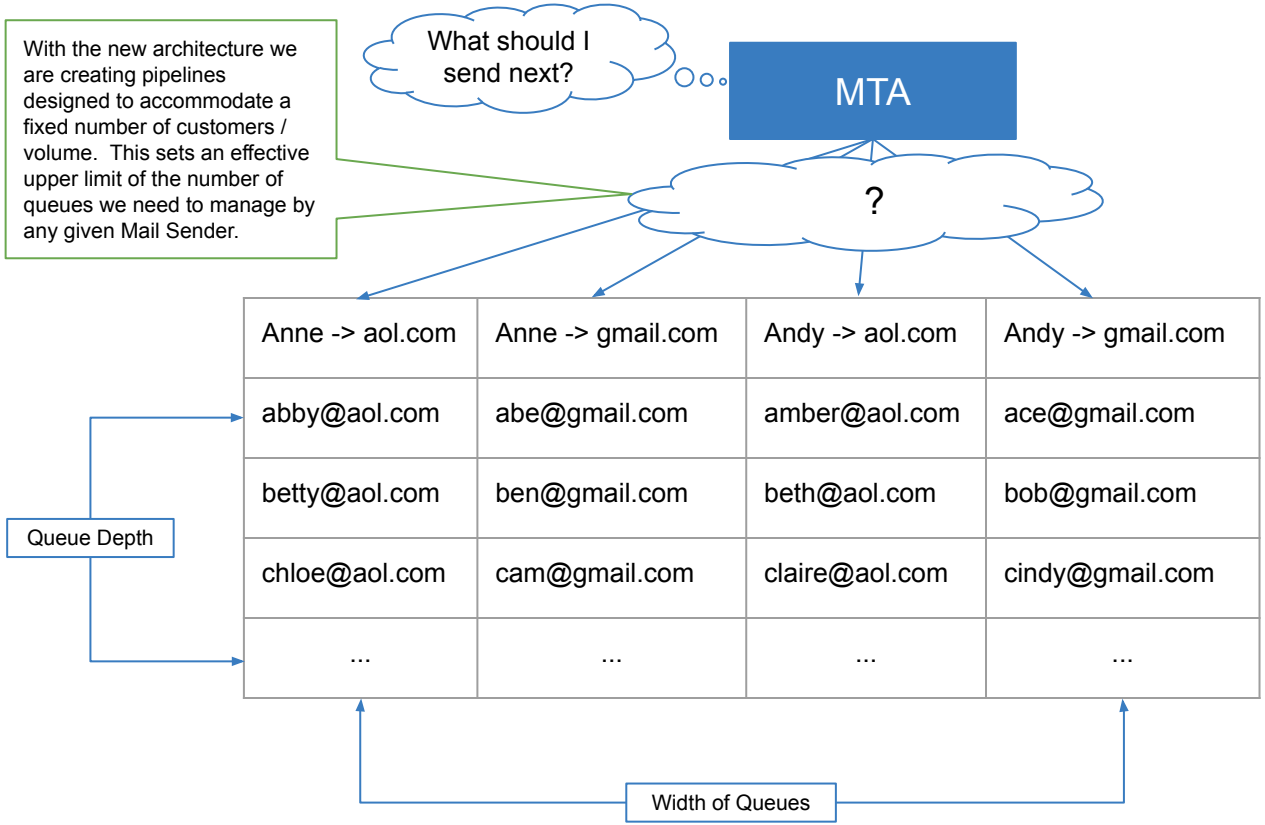


Scale Out System Fixes Egress Width



Fixed Width

With the new architecture we are creating pipelines designed to accommodate a fixed number of customers / volume. This sets an effective upper limit of the number of queues we need to manage by any given Mail Sender.



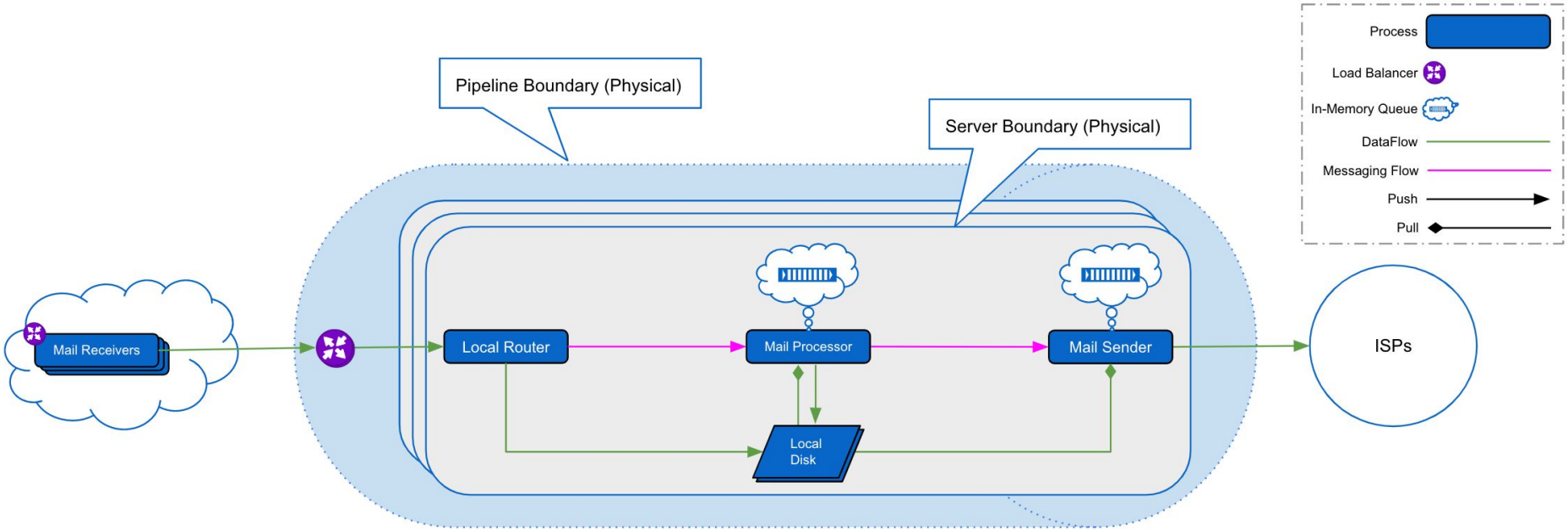
Software Architecture: Stateful to Ephemeral Compute

TWILIO SendGrid

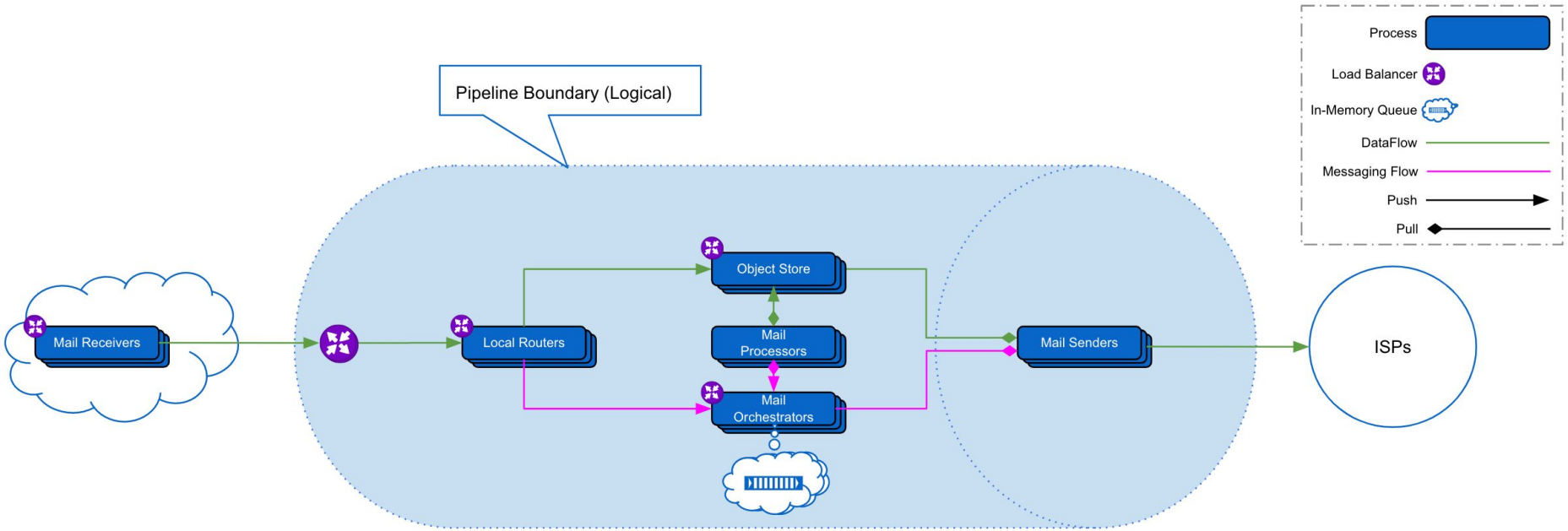
Limitations and Solutions

Unbounded Failure Domains	→	Fault Isolation
Width Thrashing	→	Fixed Width
Fault Intolerant Storage	→	Durable Storage
Stateful Compute	→	Ephemeral Compute
Tight Coupling	→	Independent Scalability
Prone to Hotspots	→	Pull-Based Architecture

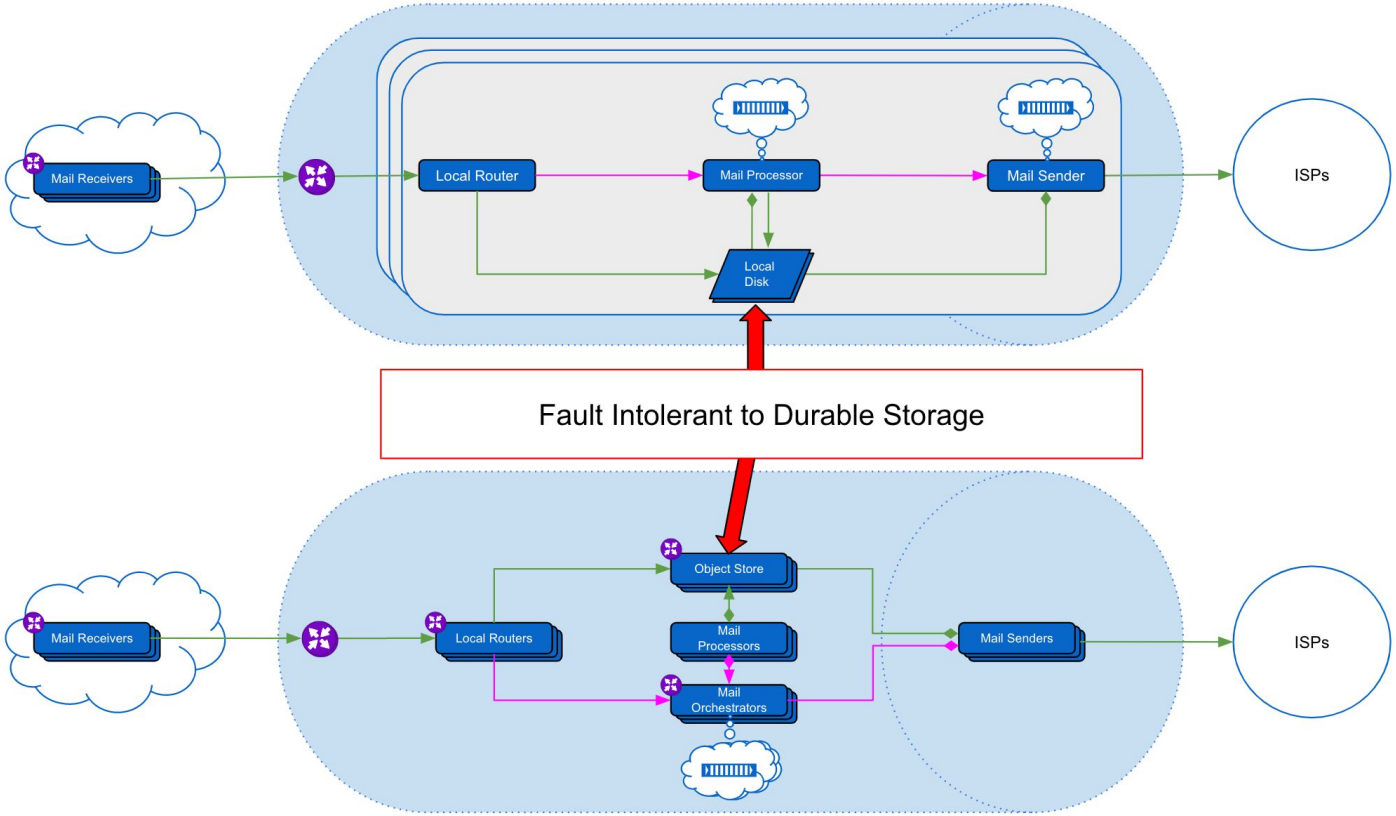
Legacy Approach



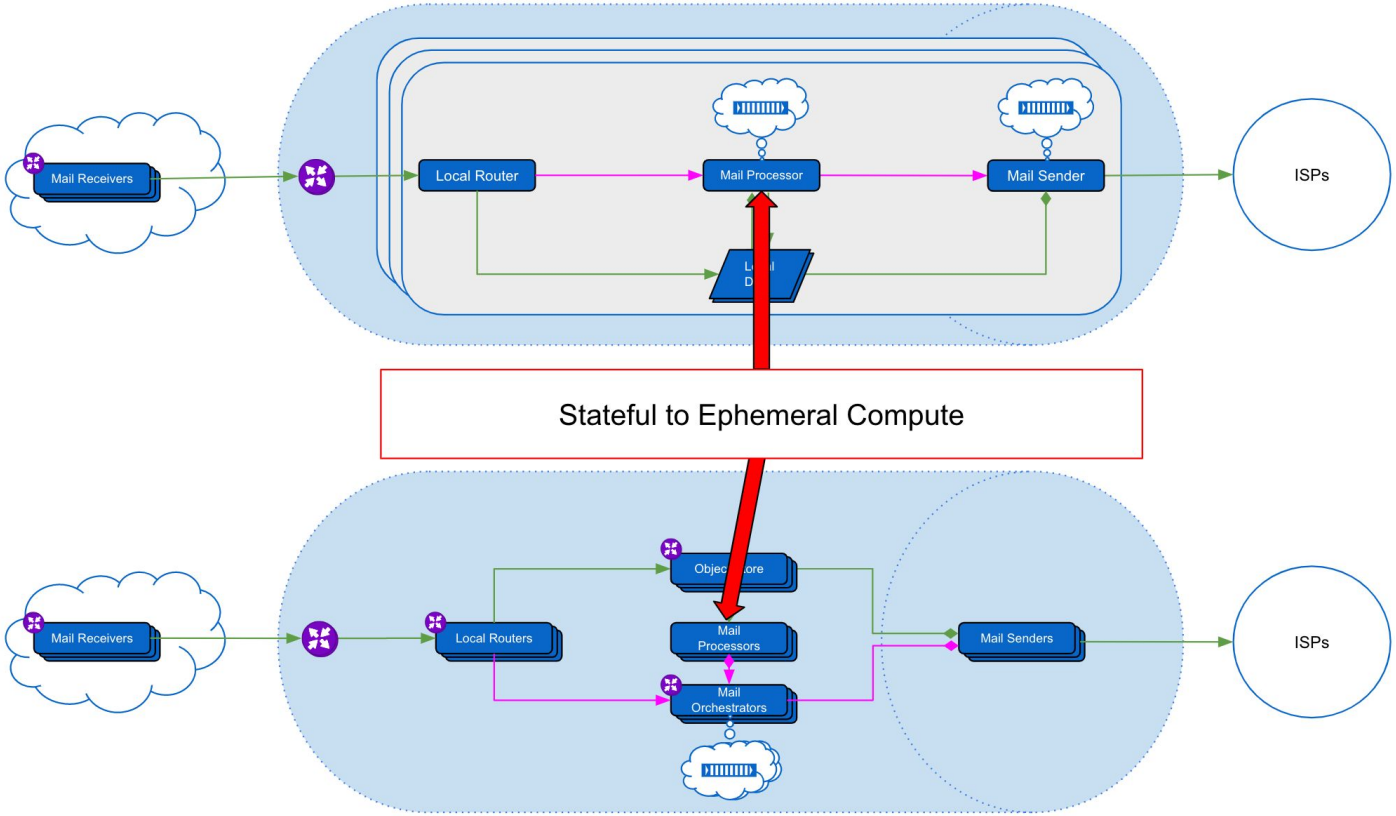
New Approach



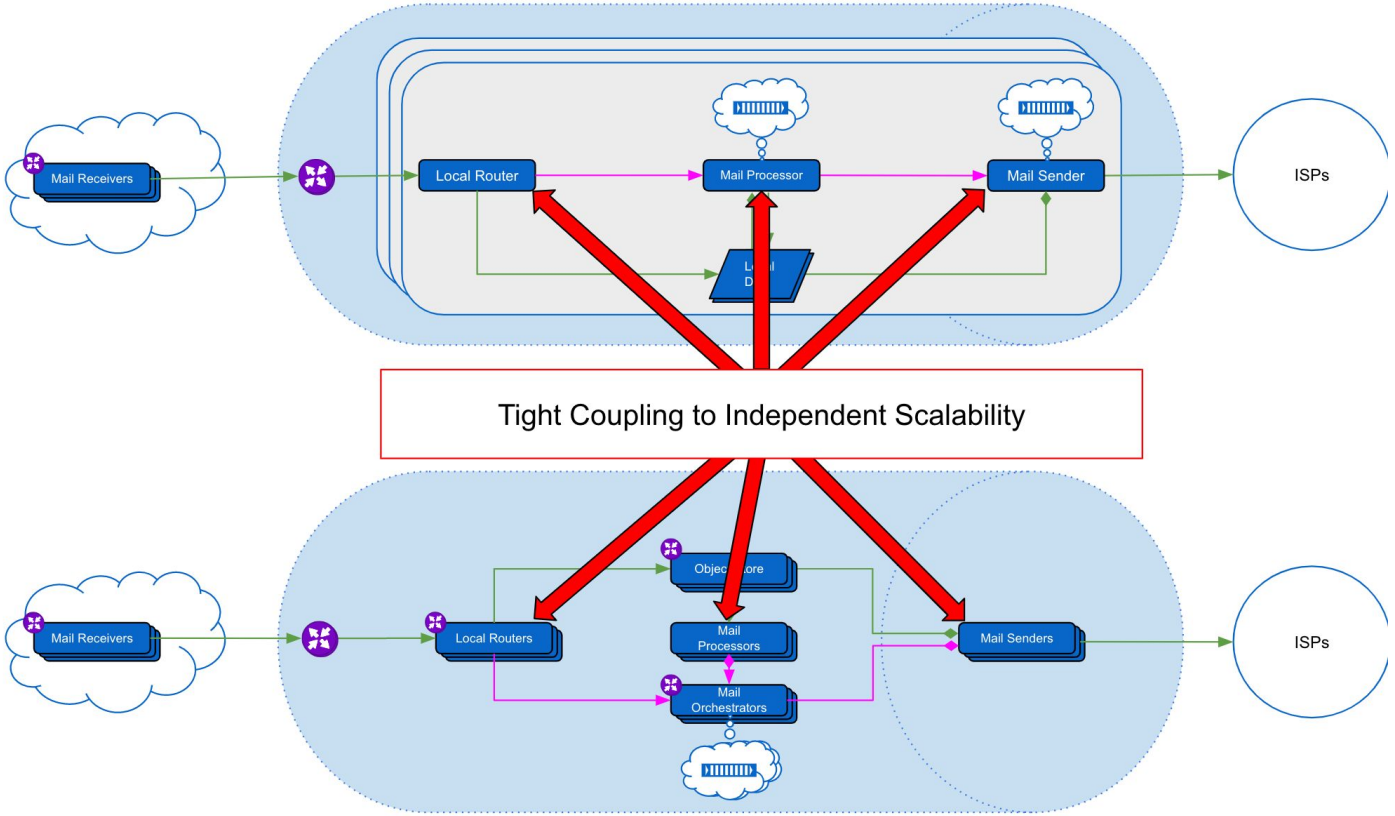
New Approach: Durable Storage



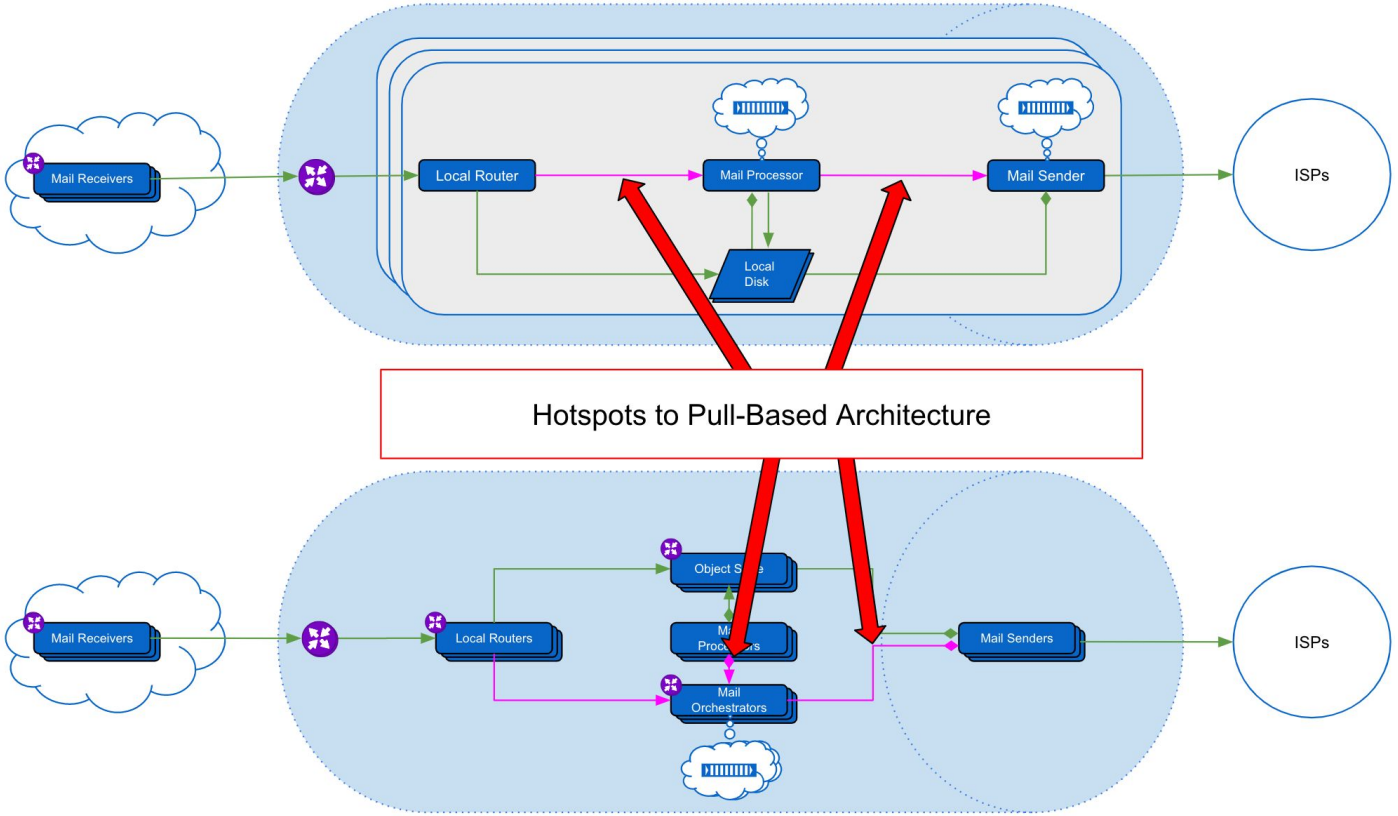
New Approach: Ephemeral Compute



New Approach: Independent Scalability



New Approach: Pull-Based Architecture



Conclusion

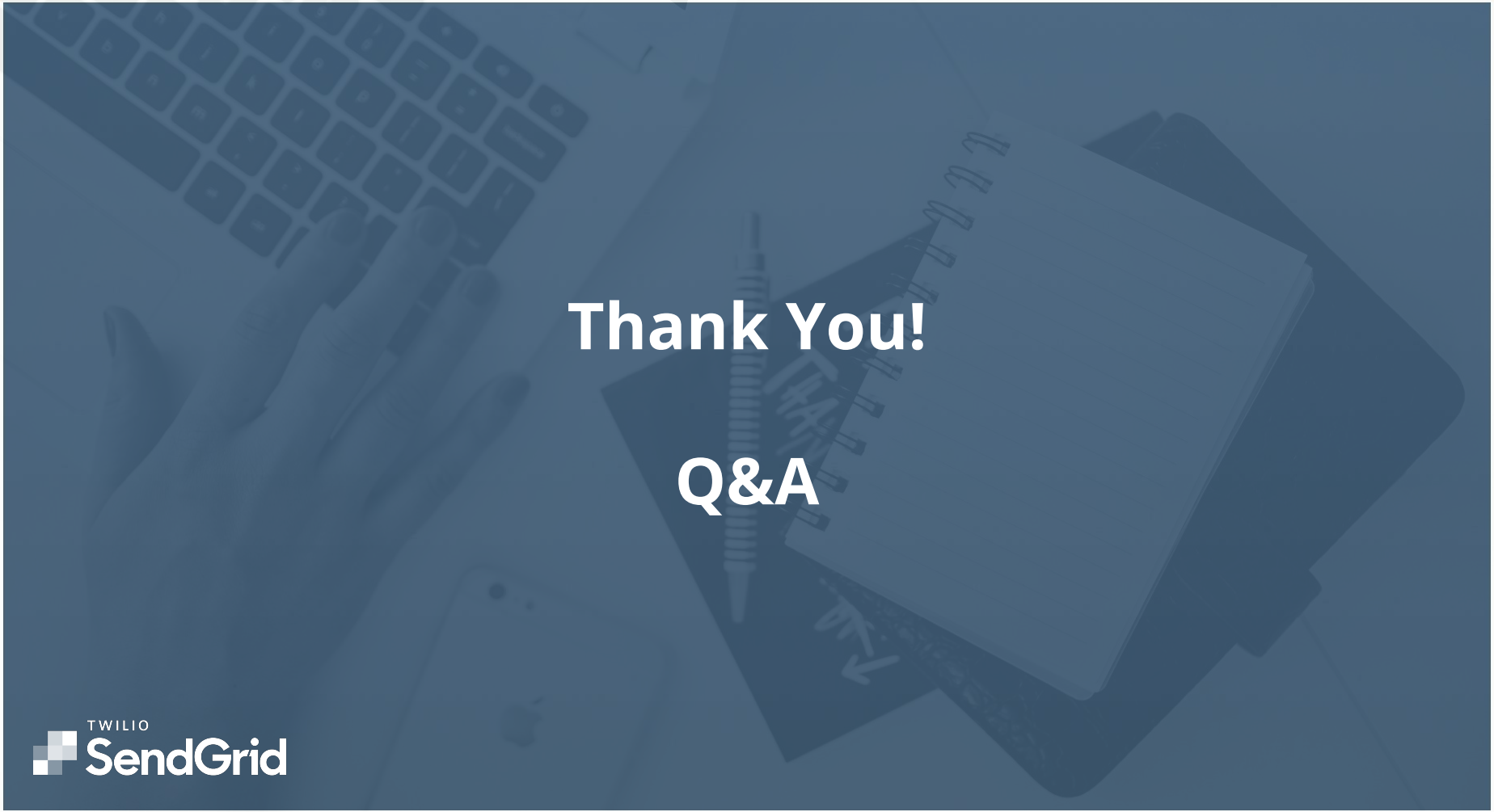
TWILIO SendGrid

Summary

- Twilio SendGrid’s infrastructure needed to be re-architected to manage increasing scale
- Simultaneously we needed to facilitate a migration to AWS
- This required sweeping changes to architecture at all layers: system/network, storage/data and compute/software
- We’ve developed and deployed this carefully and incrementally while continuing to serve 2B+ email per day
- Re-architecture complete, migration up next

Things We Learned Along the Way

- Establishing an Ideology to develop criteria saves a lot of time and fosters best-fit outcomes
  - Understand what success looks like
  - Create parameters to constrain architectural decisions
  - Reconcile opposing considerations
- More change may be required than you anticipate
  - Develop incremental transition plans
  - Lift and shift vs. re-architect cloud native is a false dichotomy



TWILIO SendGrid