KubeCon | CloudNativeCon

North America 2018

Service Meshes:

The Production Readiness Checklist for the Rest of Us

Austin Adams & Zach Arnold

Who is this for?

KubeCon | CloudNativeCon
North America 2018

- You are probably:
  - At KubeCon/CloudNativeCon NA in 2018
  - Aware of what a *Service Mesh* is at a high level
  - Managing a production collection of services (micro or otherwise) that communicate over TCP with each other
  - Experiencing the pain that comes with running these services at some scale
  - Needing a solution to your TCP based problems and don't have time for a complete rework of your application's architecture
  - In a small to mid-size business

Well, as per usual I don't know which talk to choose...
There's a session on the future of web hosting which sounds good... or maybe the one about virtual assistants...
Don't overcomplicate it, I've found the perfect talk for us
HOW TO BUILD A CHATBOT FOR BLOCKCHAIN SMART CONTRACTS USING GRAPHQL WITH SERVERLESS APIS SUPPORTED BY A SERVICE MESH
Perfect!
CommitStrip.com

Who is this probably *not* for?

- You are probably not:
  - Able to dedicate an army of engineers to solving this problem specifically for your business
  - Currently holding a PhD in Computer Science in the specific field of networking
  - Already running a service mesh in production

EXCELLENT...
LET THE LOAD BALANCING BEGIN
memegenerator.net

Who/What is Ygrene?

- Financial Services Sector
- Privately Held, Publicly Good
- PACE = Property Assessed Clean Energy
- Ygrene = The word “Energy” spelled backwards
- Our mission is to make sure that Earth is still a thing in the future.

Ygrene
ENERGY FUND

HIRING IN SEATTLE, FLEXIBLE, REMOTE, WEWORK…

Service Meshes Distilled

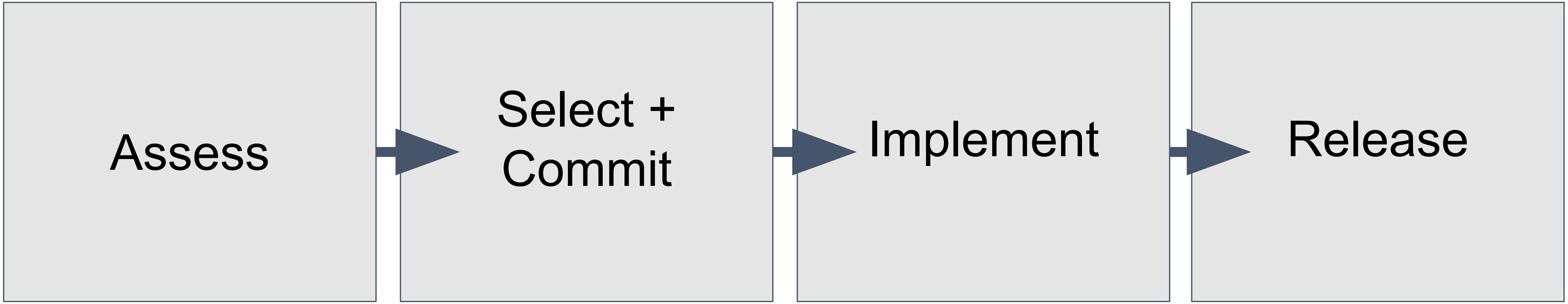
- With every “Mesh” worth using you’ll get:
  - TCP proxying
    - (HTTP1,1.1,2.0,gRPC…)
  - Traffic Flow Control:
    - DNS (or Service Discovery)
    - Load Balancing
    - Timeouts/Retries/Fault Injection/Circuit Breaking
    - Routing
  - Security
    - mTLS
    - Auth-n/Auth-z
  - Observability
    - Metrics
    - Distributed Tracing

SORRY
NO
MEME

The Landscape...more or less

- Linkerd 1 and 2 (CNCF project formerly Conduit)
- Istio (IBM + Google) with a proxy:
  - Envoy (CNCF Proxy Project)
  - Nginx
- Aspen Mesh (Managed Istio)
- AWS App Mesh
- Azure Service Fabric Mesh
- GKE Managed Istio (Is this a thing yet?)
- Nginx+…. Seriously you can pay for Nginx

The Roadmap to Production



Assess

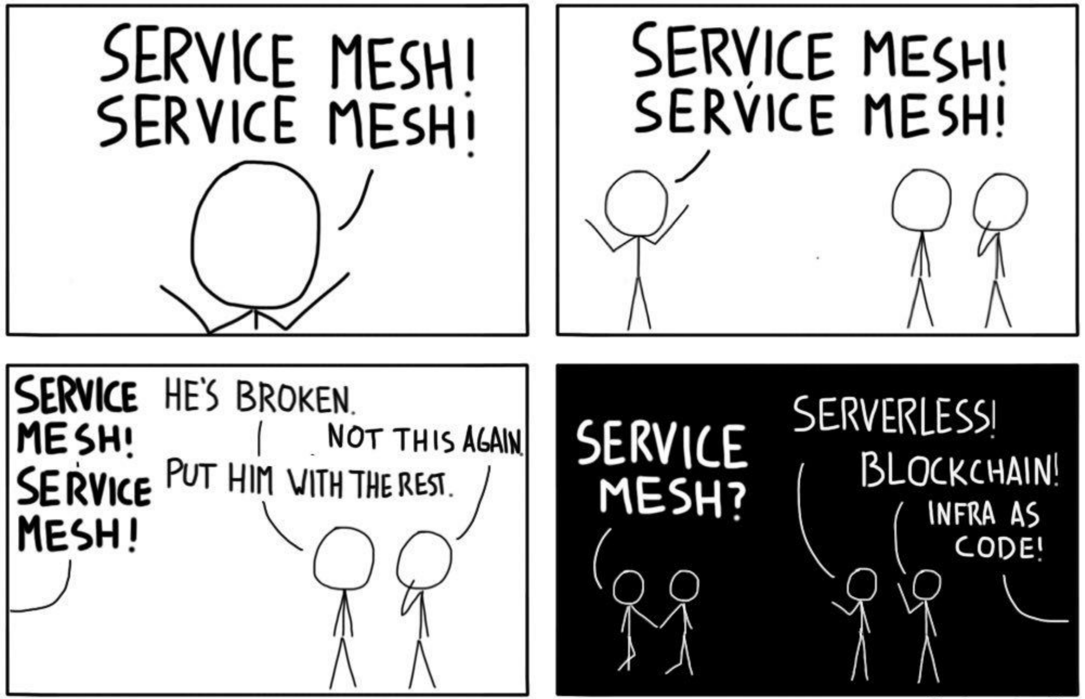
Select + Commit

Implement

Release

Assess

- Do we *need* a service mesh?, What problem does it solve?
  - Its okay if that’s a no!
- Questions to Ask yourself / POC
  - Can our team handle the added complexity ?
  - Can your application handle a service mesh ?



@sebiwicb

Assess

- Use Cases
  - Encryption between microservices without the cert management.
  - End user JWT authentication (Istio)
  - Service to service Authentication/Authorization
  - Tracing/Instrumenting your applications.
  - Intelligent routing, route by cookie, device, region, canary deployments, api version routing. Mirroring!!, Fault Injection!!

Assess - Our Use Case

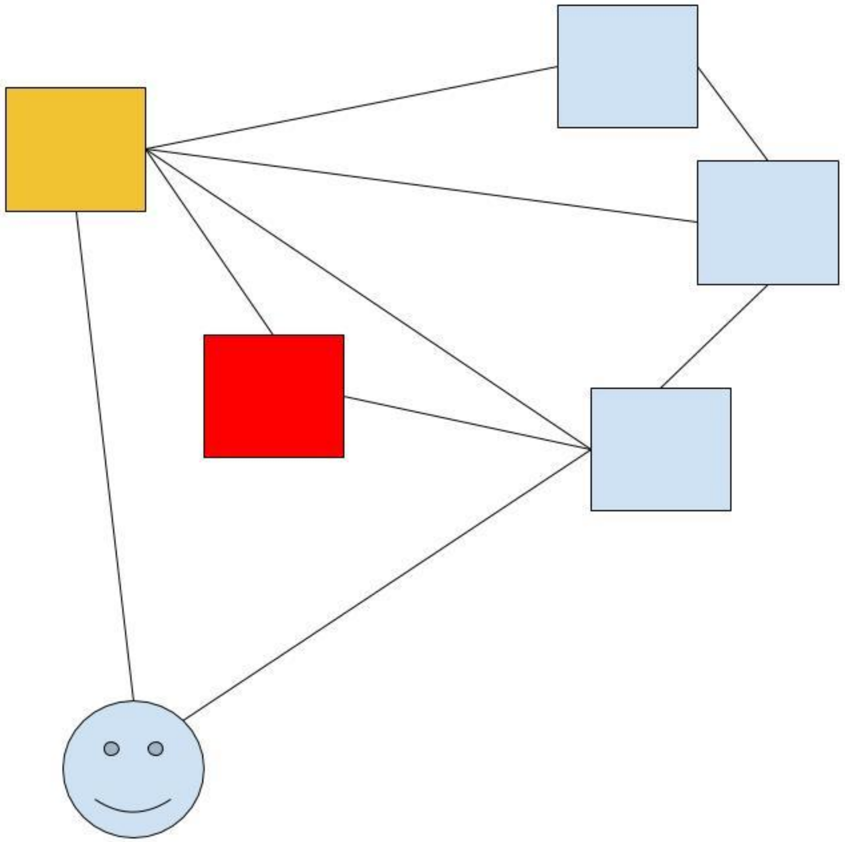
Traffic needed to be encrypted, everywhere

Traffic needed to be restricted.

JWT was being validated all the time.

Service to service permissions can be static.

We don’t want to refactor tons of code.



Assess

- Feature Mapping
- List our projects
- List our features
  - Including the Open Source Test
- Add Weights!
  - Add a number for every feature
- Try out the top contenders
  - Sum each project

Select + Commit

- Before Deciding, consider!
- The *open source* Litmus Test for choosing open-source Technology:
  - How long has it existed?
  - How popular is it in terms of contribution/usage?
  - How well sponsored is it and by whom?
  - How recently has it changed?
- Our general rule is to use a managed service where possible...it lets us concentrate on Ygrene stuff
  - But we couldn’t (EKS) so we went open source.

Select + Commit

- Our Advice!
- Linkerd 2 is simple, easy to install and will get you simple routing, metrics. Use it if you want to get a service mesh going quick.
- Istio for literally anything else if you depend on Kubernetes.
- Dont ignore Linkerd 1, especially if you have non Kubernetes services.

ISTIO!!!!!!!!

After talking to you this week...

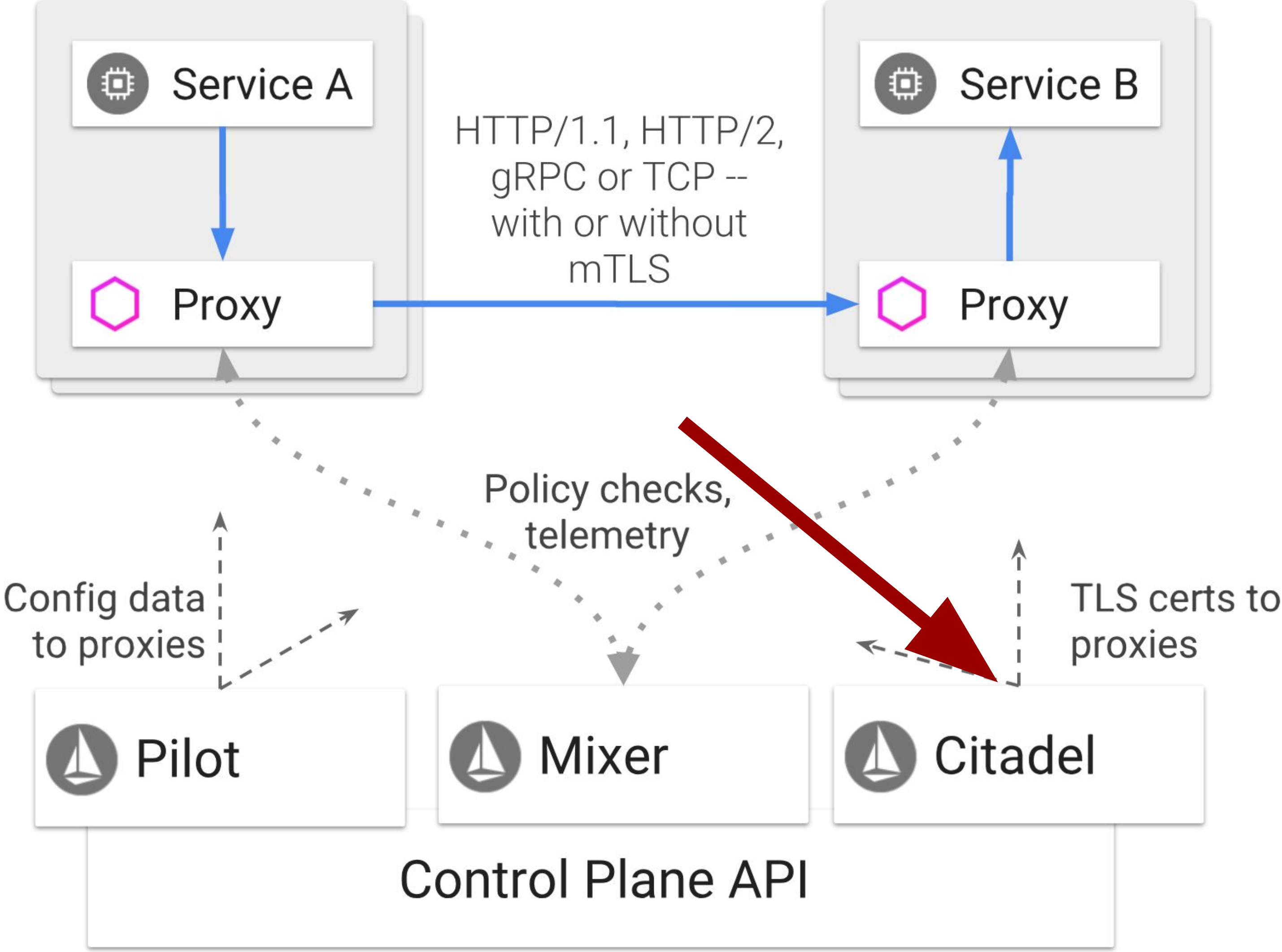
- I changed this portion of the talk
- We will focus on a few key areas
  - Shoehorning Istio (the Envoy sidecar) into your app/Engineer buy-in
  - The bits and pieces of Istio that don’t work well in EKS yet
  - How we got it into prod (for our use case, #security)

Implement...not just a demo app

- We did the opposite of what the textbook says for a good reason
  - Our first service in the mesh was the hardest to do and it handles almost 100% of our Ingress, which means we configured Ingress too!
    - It also has the most other peripherals (3rd party svcs, RDS, Redis...)
    - This portion of our app is the edge case factory...
  - The rest of our services inherit from a common base, so updating was a simple as pushing an upstream change and rolling out deploy plans
  - Dev buy in was simple, since it interrupted almost no one’s workflow

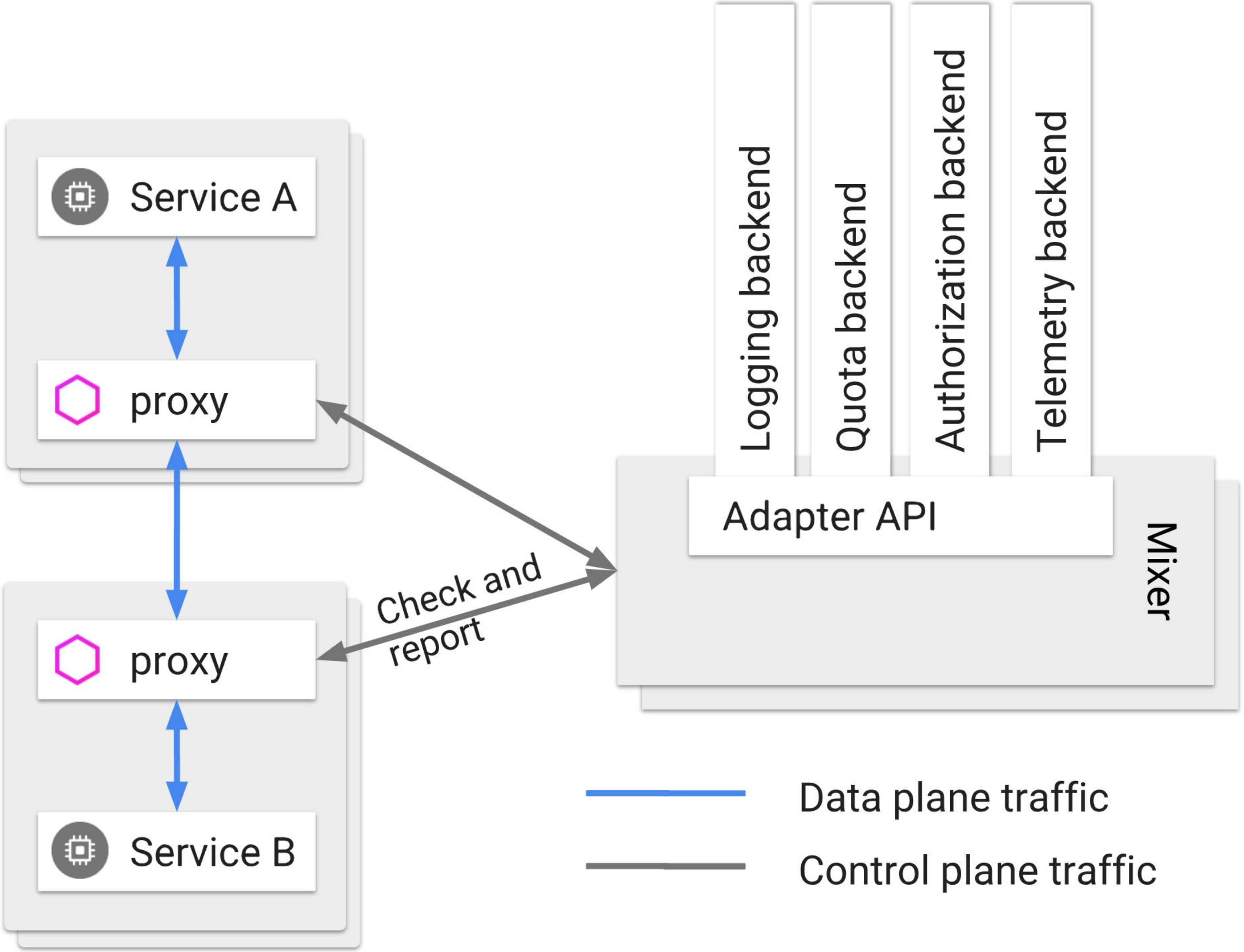
Implement...a suggested strategy

- If you can...use Helm
  - And turn off stuff for EKS
- Using your selected features, focus on the components you need
- For us, that was Citadel (mTLS)
- We *highly* recommend working on instrumenting metrics early, it will save you diagnosing problems in the long run



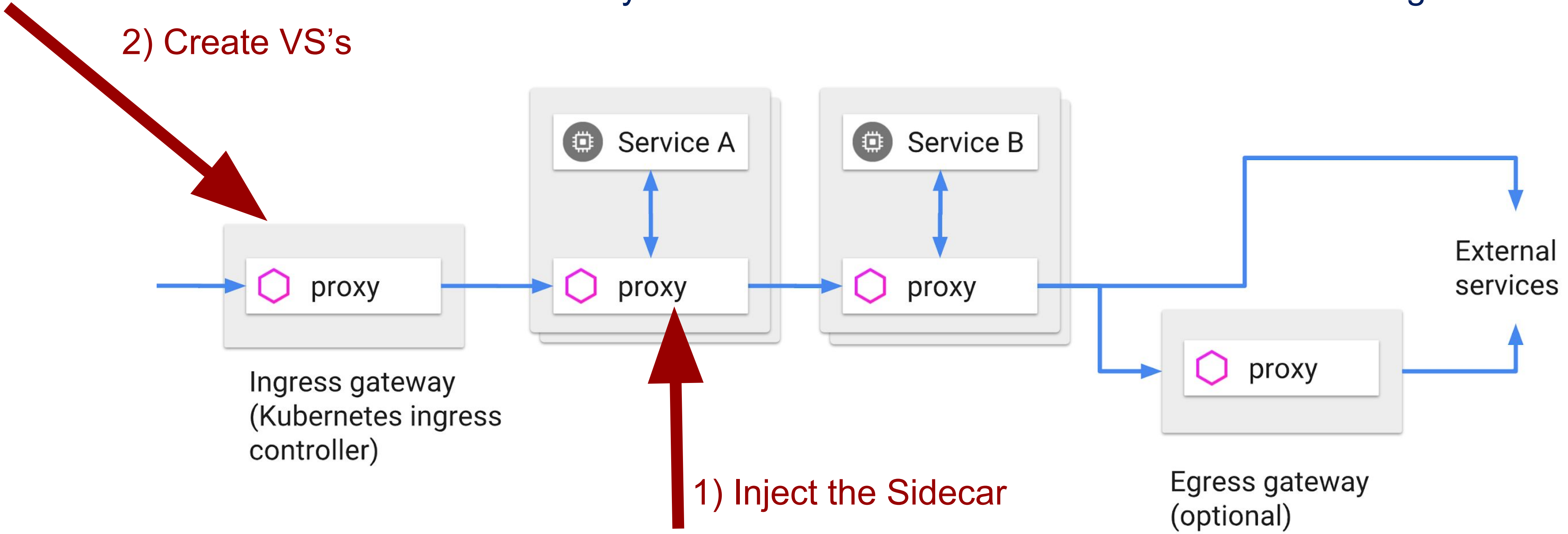
Implement...a suggested strategy

- Prometheus (our own)
- Fluentd (our own)
- Grafana (our own)
- Jaeger (SUPER USEFUL FOR DEBUGGING)



Implement...a suggested strategy

- Our hardest part would be our migration to using Service Mesh tools for authz/authn
  - Create VirtualServices for any service that would receive traffic from the Istio Ingress Gateway



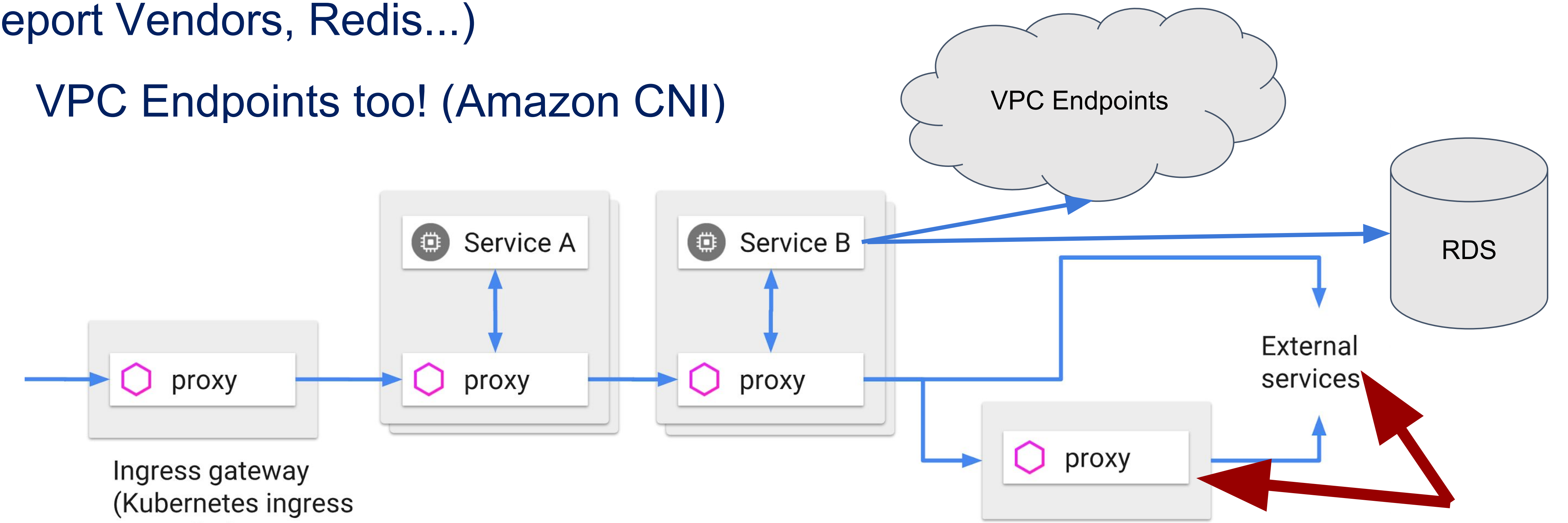
2) Create VS's

1) Inject the Sidecar

Service A
Service B
proxy
proxy
proxy
proxy
External services
Ingress gateway (Kubernetes ingress controller)
Egress gateway (optional)

Implement...a suggested strategy

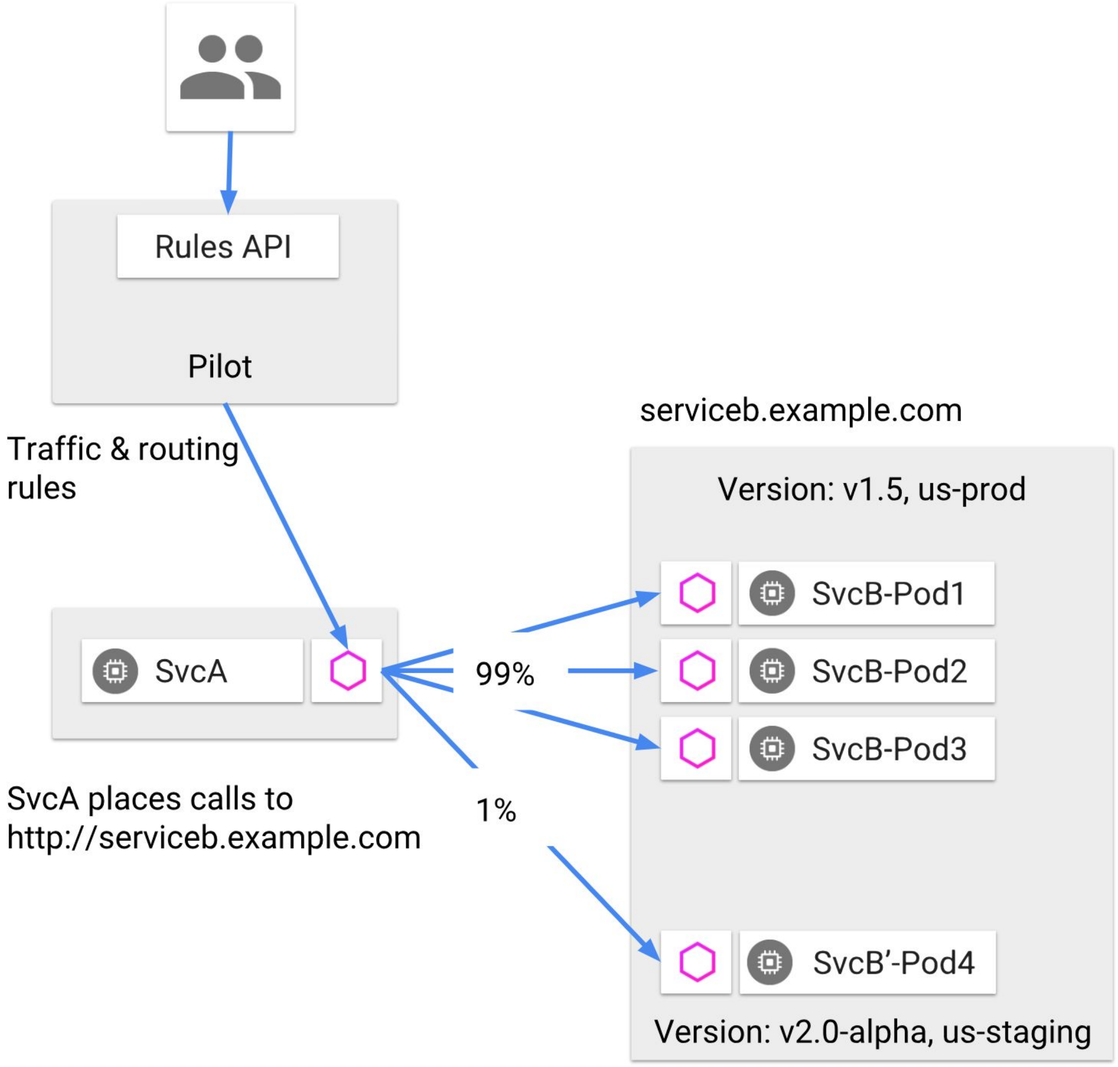
- Whitelist all outbound HTTPS/TCP traffic to 3rd party vendors (RDS, Credit Report Vendors, Redis...)
    - VPC Endpoints too! (Amazon CNI)


VPC Endpoints
RDS
Service A
Service B
proxy
proxy
proxy
proxy
External services
Ingress gateway (Kubernetes ingress controller)
Egress gateway (optional)

3) Whitelist ServiceEntries and VirtualServices

Implement...a suggested strategy

- Get ready for deployments, and leverage Istio DestinationRules or Kubernetes Services
  - (if you do Blue/Green or Canary deployments)
  - Otherwise Istio just works like Kubernetes services
- Load test, load test, load test



Rules API

Pilot

Traffic & routing rules

serviceb.example.com

Version: v1.5, us-prod

SvcB-Pod1

SvcB-Pod2

SvcB-Pod3

SvcA

99%

1%

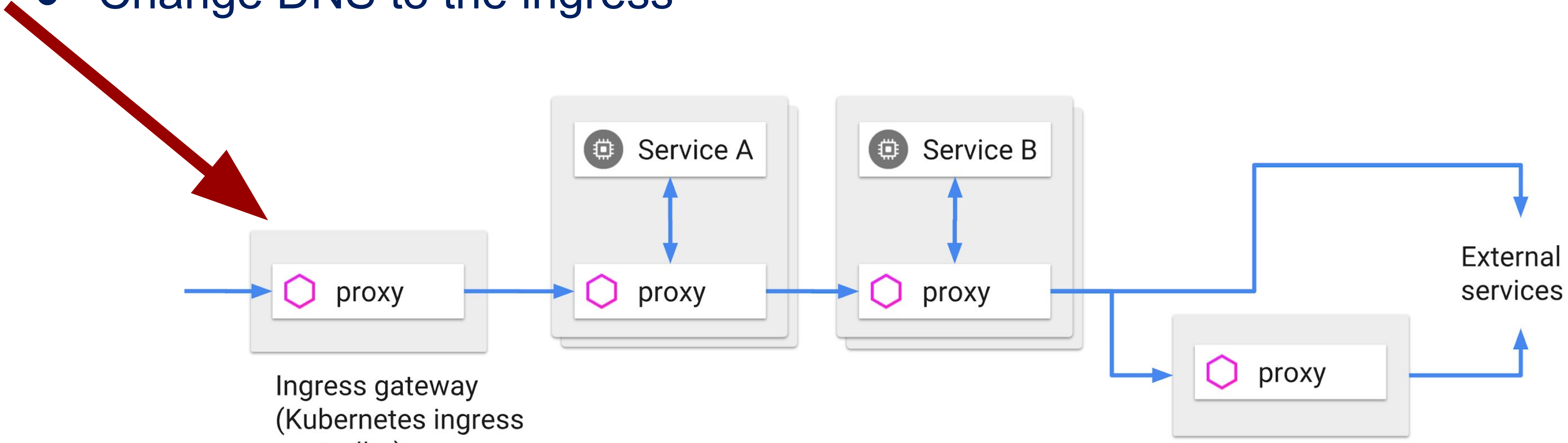
SvcA places calls to http://serviceb.example.com

SvcB’-Pod4

Version: v2.0-alpha, us-staging

Release...with no interruptions

- Provision SSL certs for public domains that you want routable in the mesh *early*
    - *We used the Jetstack Certmanager (Open Source)*
- Change DNS to the ingress



Service A

Service B

proxy

proxy

proxy

proxy

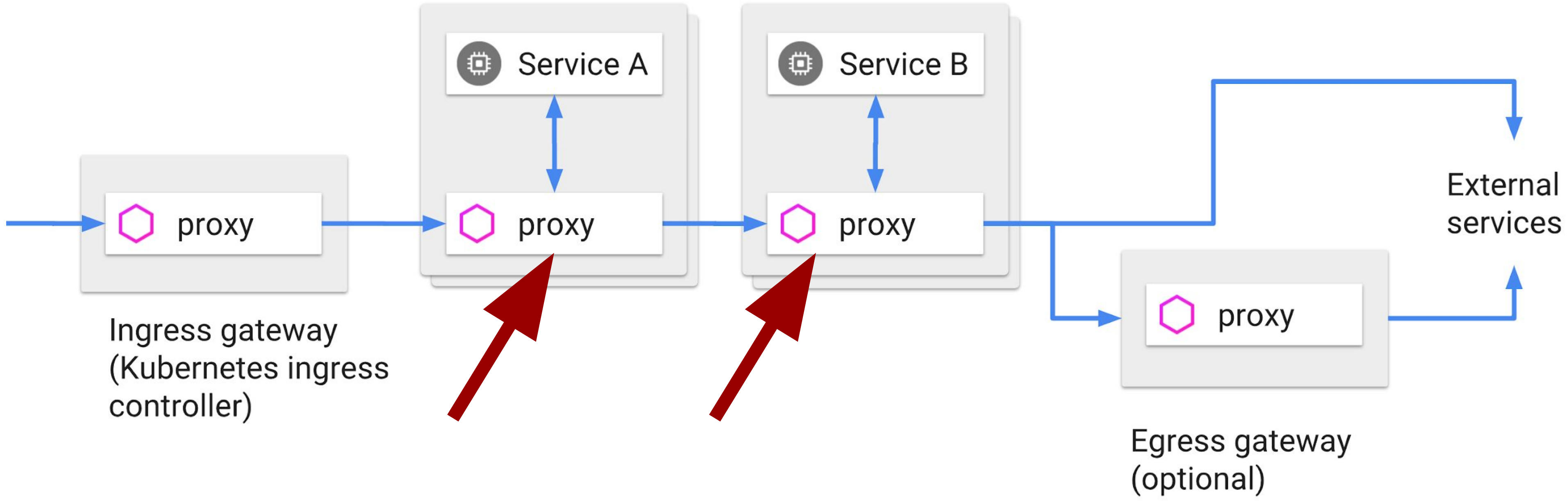
External services

Ingress gateway (Kubernetes ingress controller)

Egress gateway (optional)

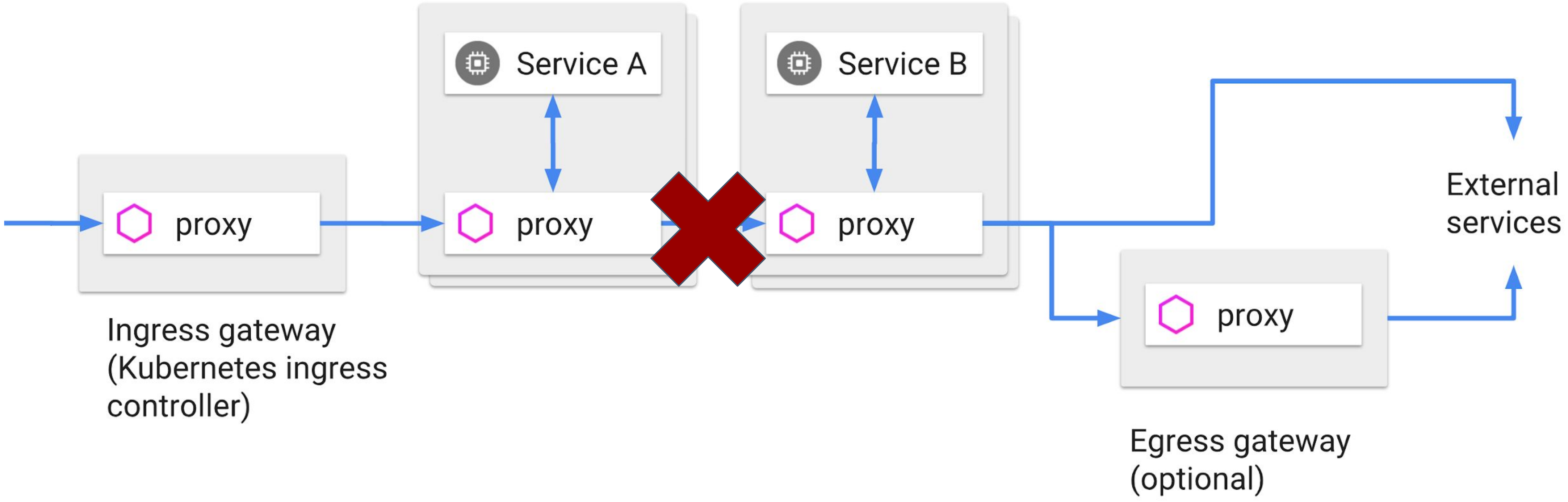
Release...with no interruptions

- Change the internal mesh policy to accept mixed auth traffic and change senders of traffic to use TLS
- Enforce TLS everywhere by policy


Service A
Service B
proxy
proxy
proxy
proxy
External services
Ingress gateway (Kubernetes ingress controller)
Egress gateway (optional)

Release...with no interruptions

- Enforce communication restrictions via RBAC (if necessary)


Service A
Service B
proxy
proxy
proxy
proxy
External services
Ingress gateway (Kubernetes ingress controller)
Egress gateway (optional)

KubeCon | CloudNativeCon

North America 2018