KubeCon | CloudNativeCon

North America 2019

KubeCon | CloudNativeCon

North America 2019

How Yelp Moved Security from the App to the Mesh with Envoy and OPA

Ben Plotnick & Daniel Popescu

Speakers

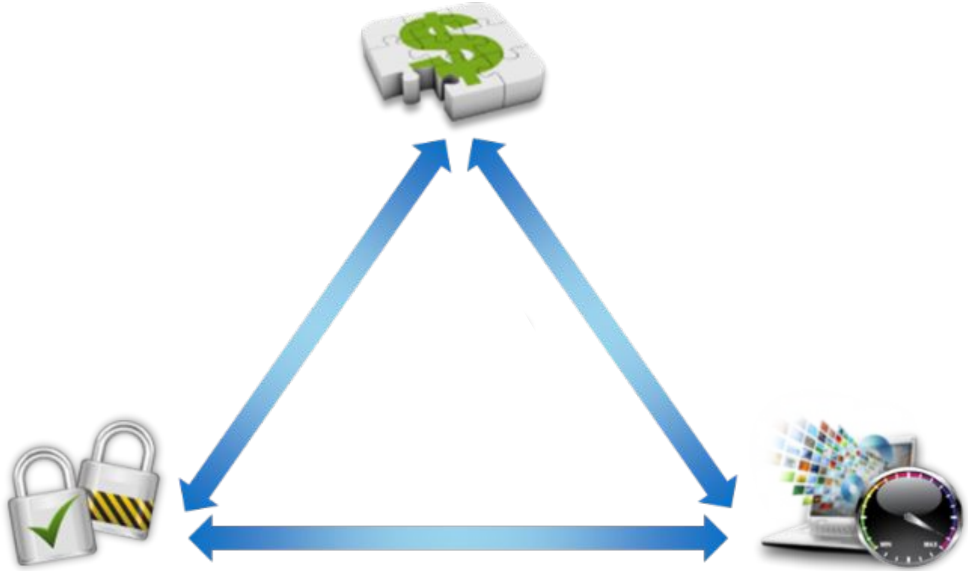
- Daniel Popescu
    - Infrastructure Security Engineer @ Yelp
    - Enjoys yoga, flow arts, and reading AWS IAM Policy docs

Speakers

- Ben Plotnick
  - Senior Platform Engineer @ Cruise Automation
  - Formerly Engineering Effectiveness @ Yelp

#Goals

Cost and Maintenance



Risk and Compliance

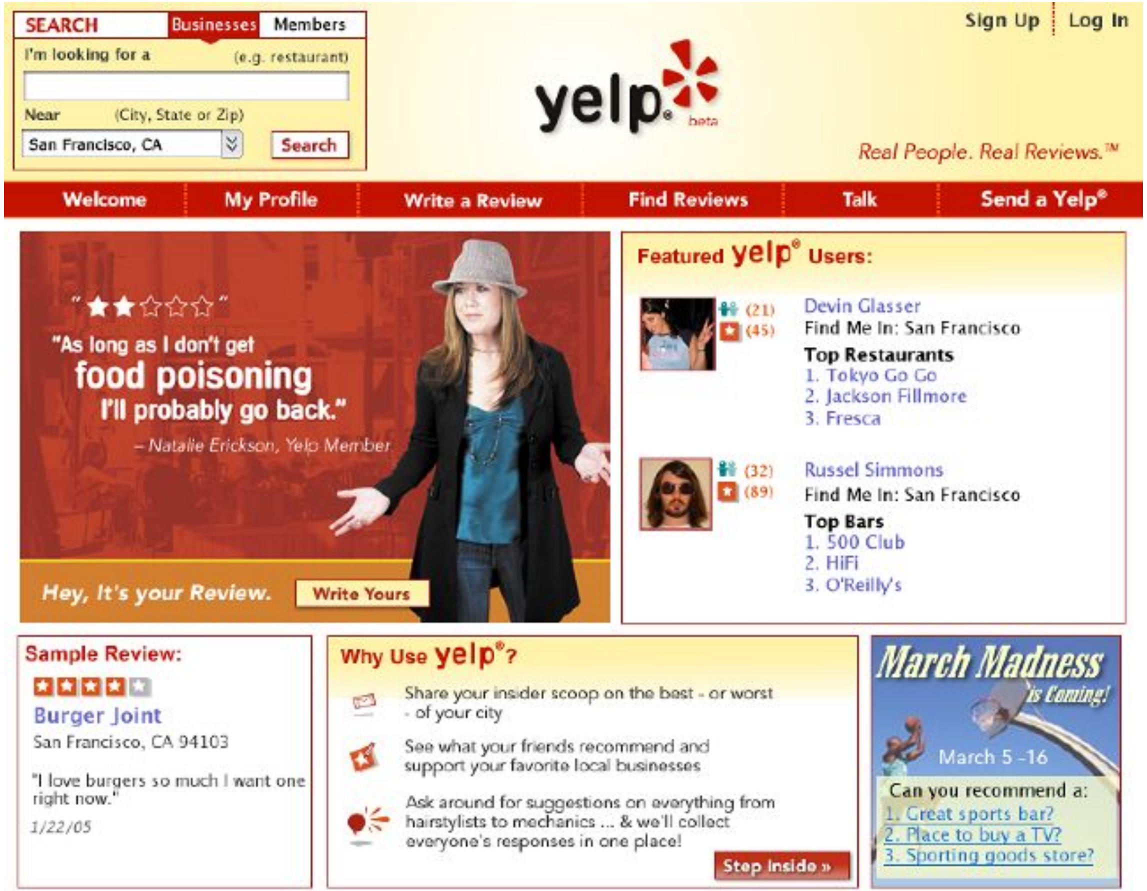
Productivity and User Experience

Yelp’s Mission

Yelp’s Mission

Connecting people with great local businesses

Yelp ca. 2005



SEARCH | Businesses | Members

I'm looking for a (e.g. restaurant)

Near (City, State or Zip)

San Francisco, CA

Search

Sign Up | Log In

yelp beta

Real People. Real Reviews.™

Welcome | My Profile | Write a Review | Find Reviews | Talk | Send a Yelp®

" ★★☆☆☆ "

"As long as I don't get **food poisoning** I'll probably go back."

– *Natalie Erickson, Yelp Member*

Hey, It's your Review. Write Yours

Featured yelp® Users:

(21) (45) Devin Glasser
Find Me In: San Francisco
Top Restaurants
1. Tokyo Go Go
2. Jackson Fillmore
3. Fresca

(32) (89) Russel Simmons
Find Me In: San Francisco
Top Bars
1. 500 Club
2. HiFi
3. O'Reilly's

Sample Review:

★★★★☆

Burger Joint
San Francisco, CA 94103

"I love burgers so much I want one right now."

1/22/05

Why Use yelp®?

Share your insider scoop on the best - or worst - of your city

See what your friends recommend and support your favorite local businesses

Ask around for suggestions on everything from hairstylists to mechanics ... & we'll collect everyone's responses in one place!

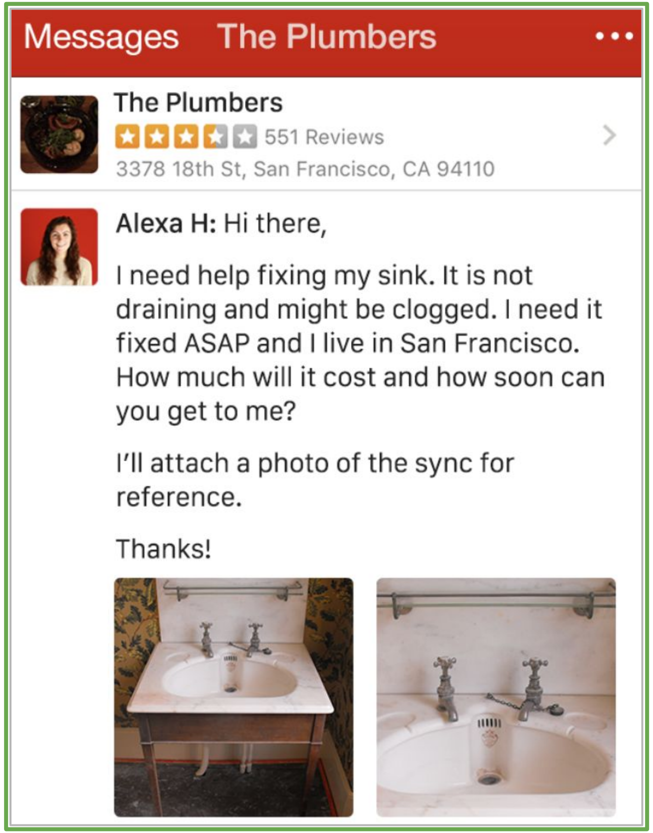
Step Inside »

March Madness is Coming!

March 5 -16

Can you recommend a:
1. Great sports bar?
2. Place to buy a TV?
3. Sporting goods store?

Yelp ca. 2019



Messages The Plumbers •••

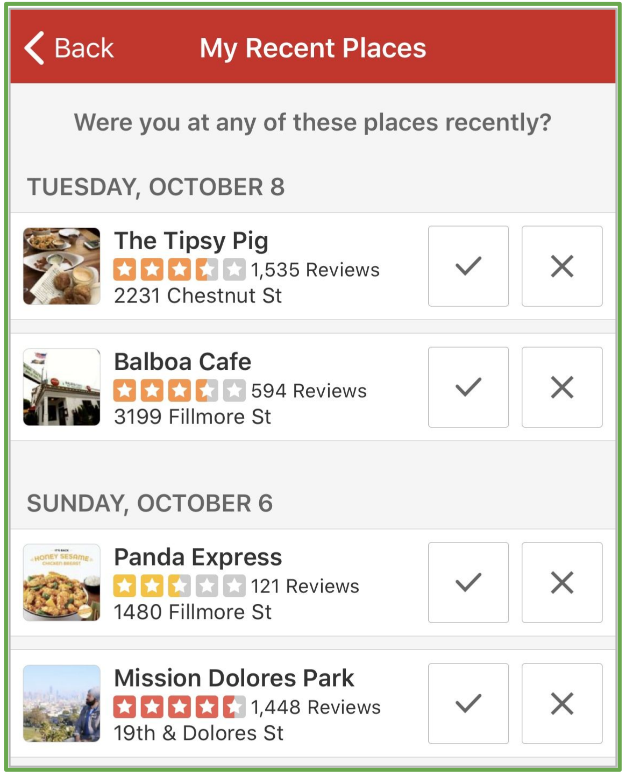
The Plumbers
★★★★☆ 551 Reviews
3378 18th St, San Francisco, CA 94110

Alexa H: Hi there,

I need help fixing my sink. It is not draining and might be clogged. I need it fixed ASAP and I live in San Francisco. How much will it cost and how soon can you get to me?

I'll attach a photo of the sync for reference.

Thanks!



< Back My Recent Places

Were you at any of these places recently?

TUESDAY, OCTOBER 8

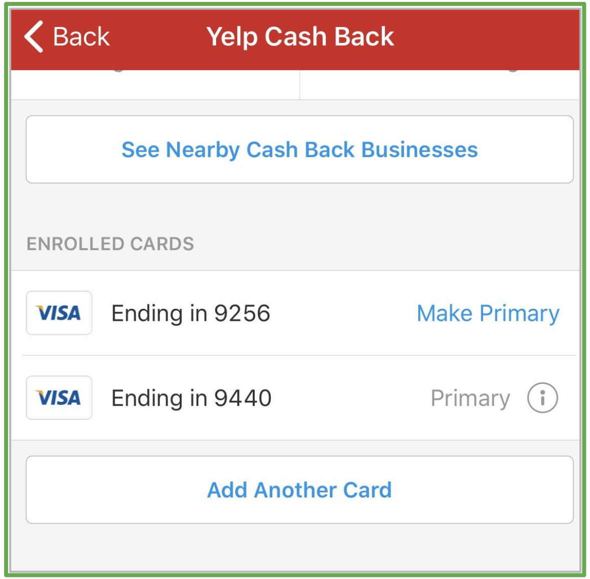
The Tipsy Pig
★★★★☆ 1,535 Reviews
2231 Chestnut St
✓ ✕

Balboa Cafe
★★★★☆ 594 Reviews
3199 Fillmore St
✓ ✕

SUNDAY, OCTOBER 6

Panda Express
★★★☆☆ 121 Reviews
1480 Fillmore St
✓ ✕

Mission Dolores Park
★★★★★ 1,448 Reviews
19th & Dolores St
✓ ✕



< Back Yelp Cash Back

See Nearby Cash Back Businesses

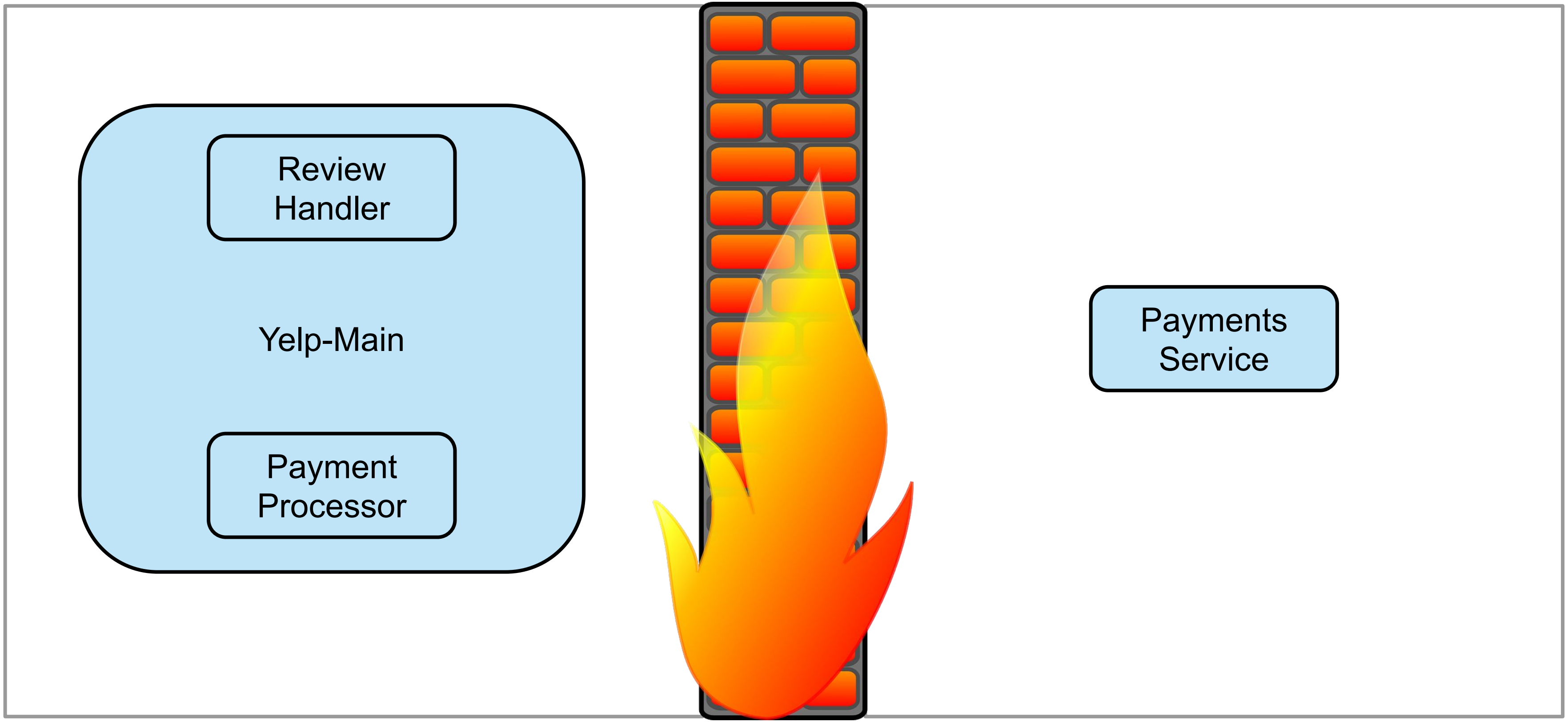
ENROLLED CARDS

VISA Ending in 9256 Make Primary

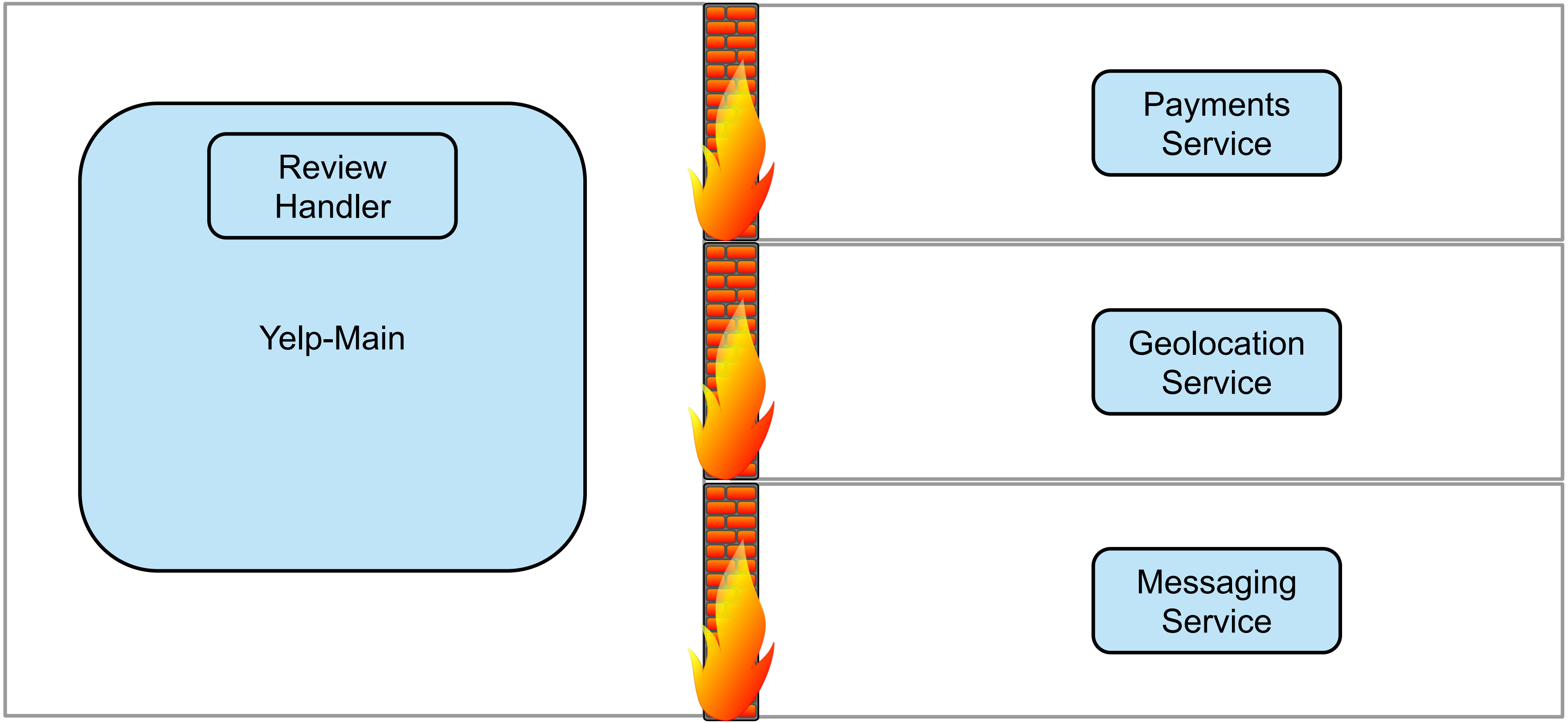
VISA Ending in 9440 Primary ⓘ

Add Another Card

Walled Garden Approach



Walled Garden Approach



Problem

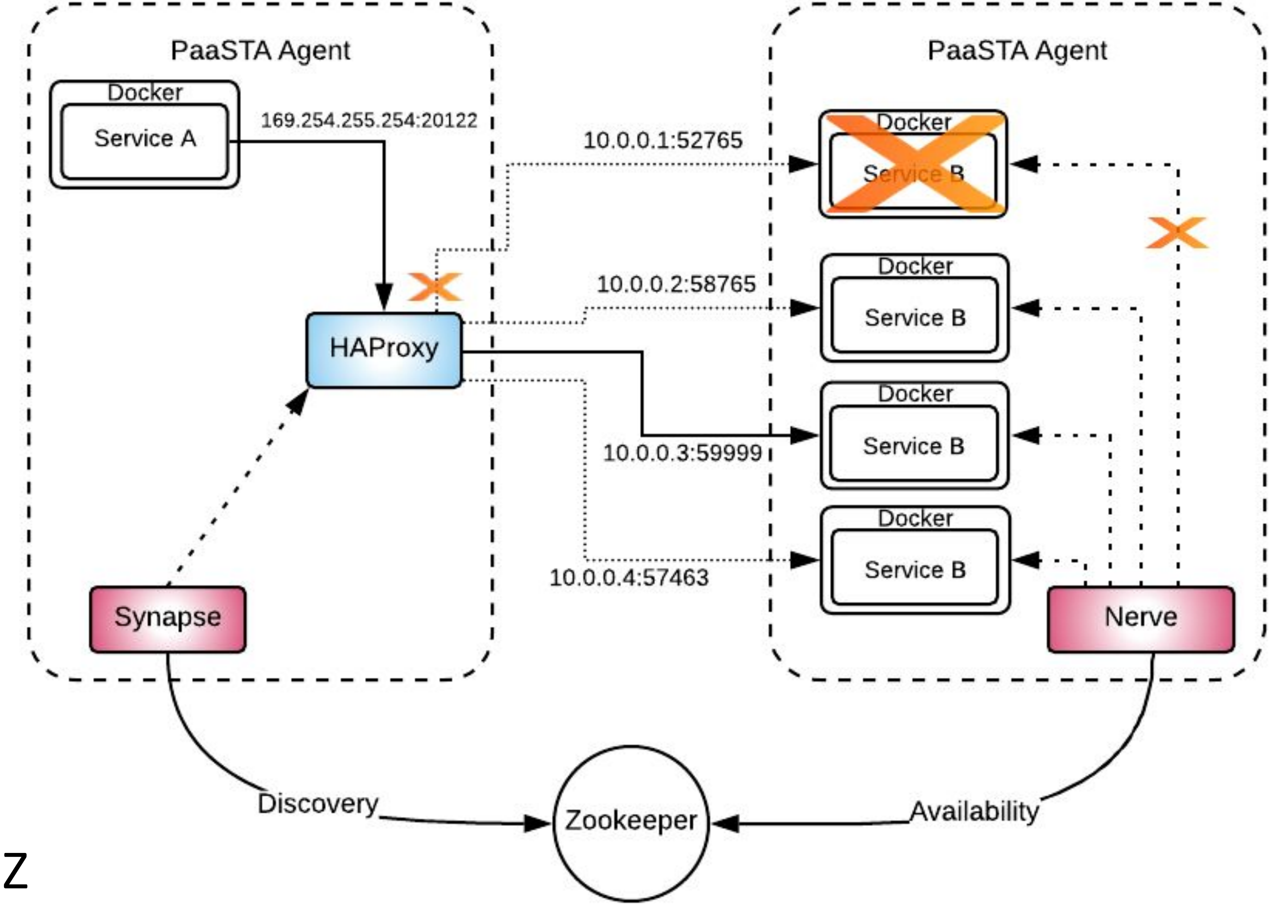
How do we build a generalized system for securing workloads?

Solution

<u>Istio</u>

Questions?

- PaaSTA
  - K8s + mesos
- Smartstack
  - Nerve
  - Synapse
  - Zookeeper
  - HAProxy
- Multi-Tenancy
- Egress proxy only
  - No sidecars
- Plain-text traffic
- Ad-hoc AuthN/AuthZ



Problem

Who are you?

AUTH
N
AUTH
Z

What can you do?

Authentication

- Who are you?
- Prove it!

Examples:

- username/password
- API keys
- x509 certificates

Authentication Requirements

- Handle identity resolution for computers and humans
- Identity must be unforgeable
- Avoid impacting developer velocity

Disclaimer: Yelp requirements not suitable for all orgs. Your mileage may vary. Only your security expert can tell you your authentication requirements

Authentication Requirements

- Handle identity resolution for **computers** and humans
- Identity must be unforgeable
- Avoid impacting developer velocity

Basic computer use-case:

```
requests.get(“happyhour.service.yelp/hours/123”)
```

Authentication Requirements

- Handle identity resolution for computers and **humans**
- Identity must be unforgeable
- Avoid impacting developer velocity

Basic human use-case:

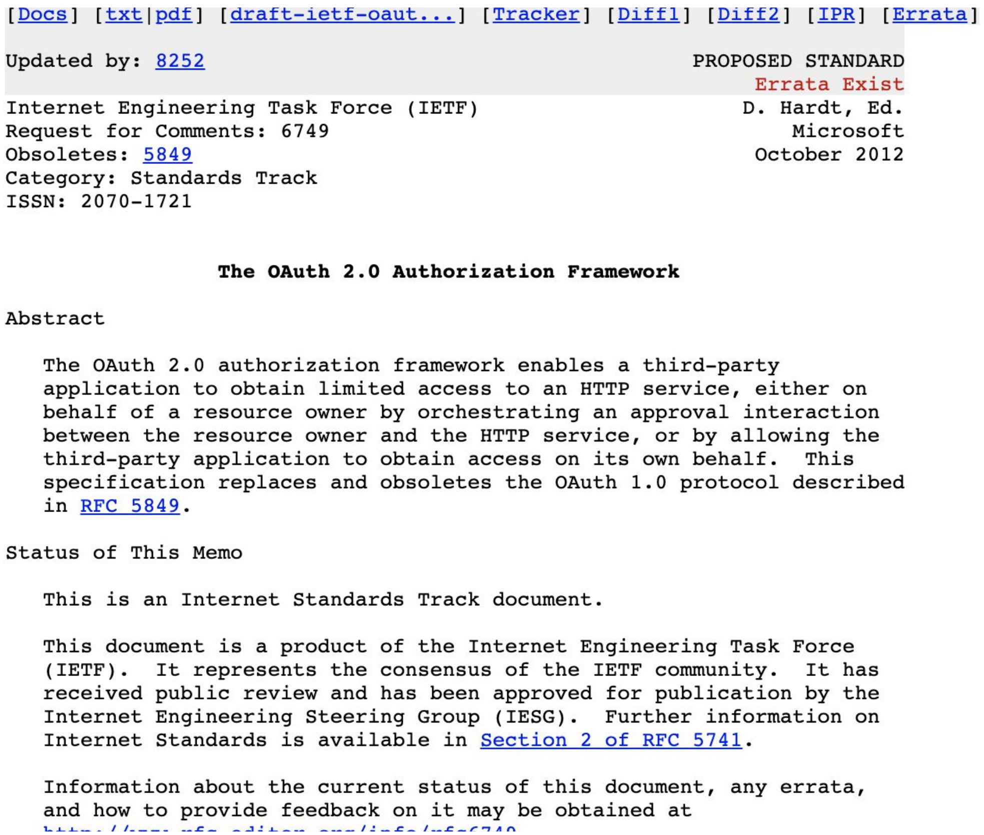
```
curl http://happyhour.service.yelp/hours/123
```

What not to do...

"Please read and understand steps 1-65 of the instructions and read the OAuth RFC"

"Also migrate all clients..."

"Also follow a separate process for workload authentication…"



```
[Docs] [txt|pdf] [draft-ietf-oaut...] [Tracker] [Diff1] [Diff2] [IPR] [Errata]

Updated by: 8252                                       PROPOSED STANDARD
                                                            Errata Exist
Internet Engineering Task Force (IETF)                     D. Hardt, Ed.
Request for Comments: 6749                                     Microsoft
Obsoletes: 5849                                             October 2012
Category: Standards Track
ISSN: 2070-1721


                  The OAuth 2.0 Authorization Framework

Abstract

   The OAuth 2.0 authorization framework enables a third-party
   application to obtain limited access to an HTTP service, either on
   behalf of a resource owner by orchestrating an approval interaction
   between the resource owner and the HTTP service, or by allowing the
   third-party application to obtain access on its own behalf.  This
   specification replaces and obsoletes the OAuth 1.0 protocol described
   in RFC 5849.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF).  It represents the consensus of the IETF community.  It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG).  Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6749
```

What not to do...

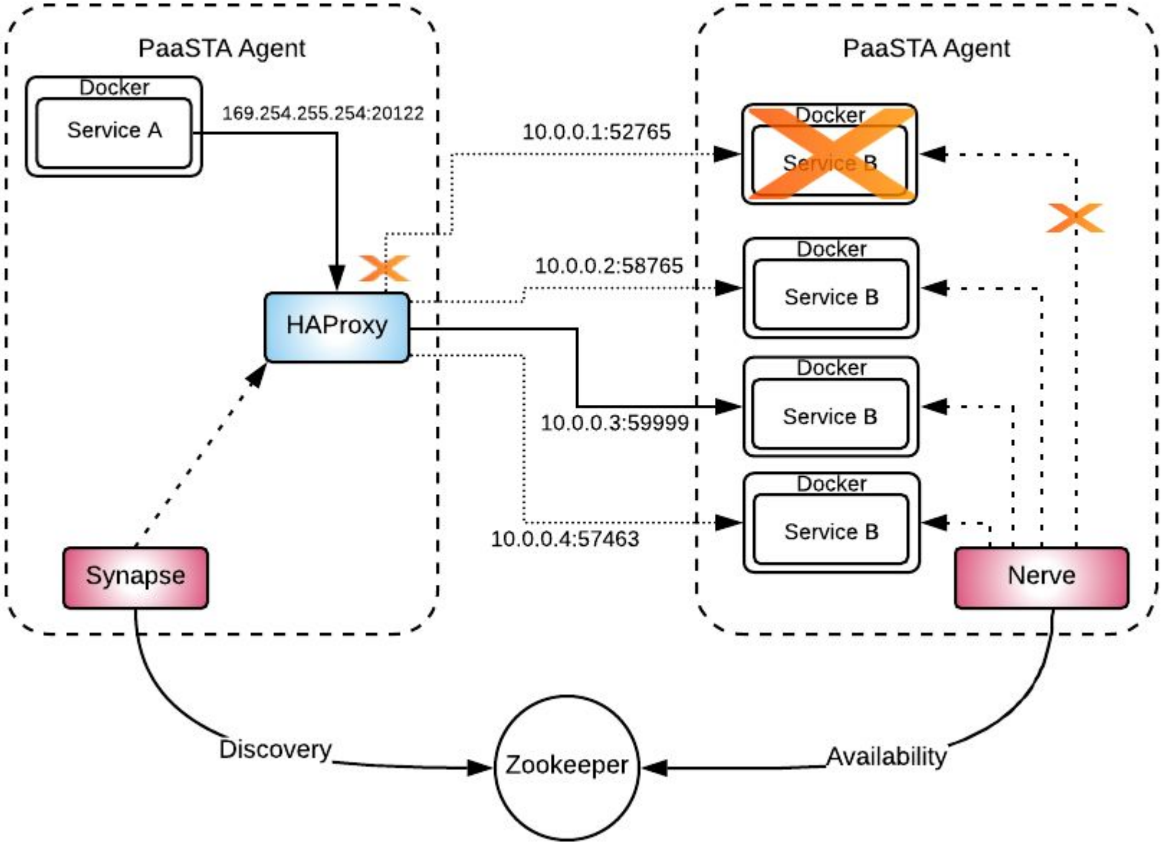
“Please read and 1-65 of the instru the OAuth RFC”

“Also migrate all

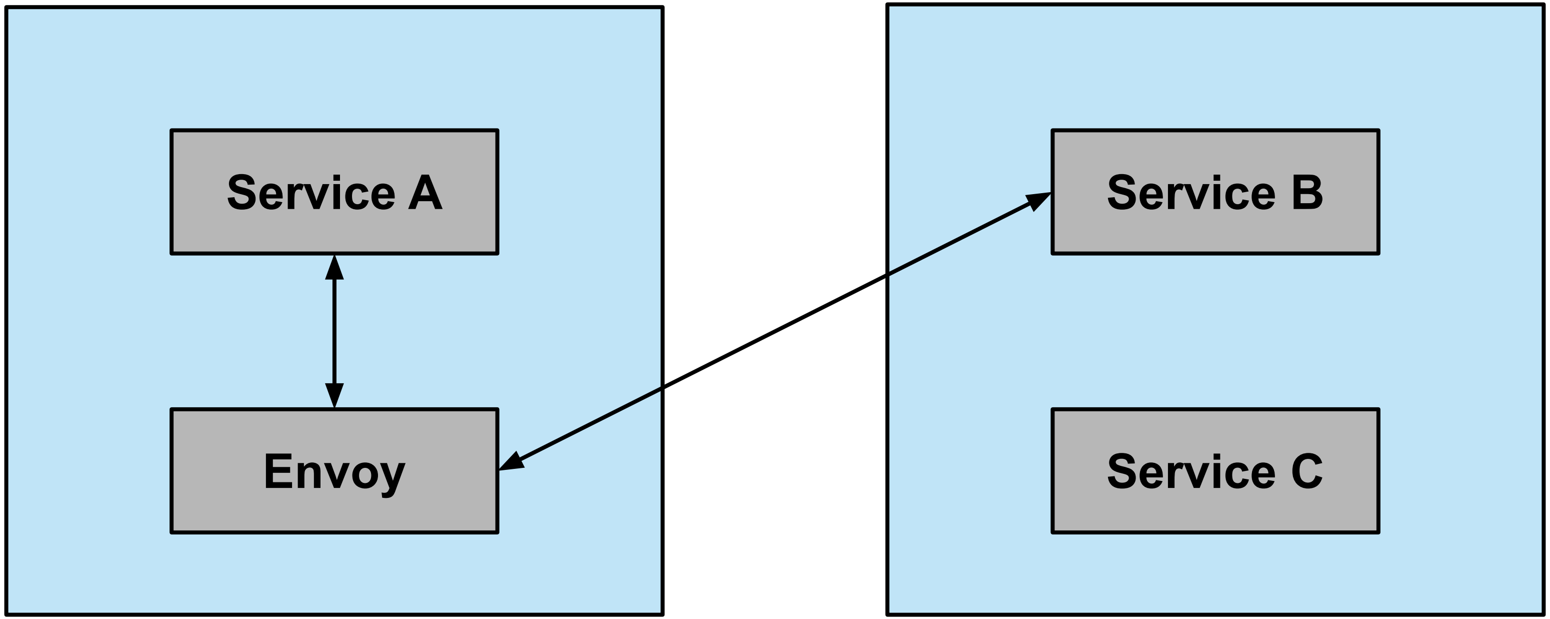
“Also follow a se workload authentication…

```
[Docs] [txt|pdf] [draft-ietf-oaut...] [Tracker] [Diff1] [Diff2] [IPR] [Errata]

Updated by: 8252                                           PROPOSED STANDARD
                                                                Errata Exist
                                                               D. Hardt, Ed.
                                                                   Microsoft
                                                                October 2012

                                    Framework

                                    bles a third-party
                                    n HTTP service, either on
                                    ng an approval interaction
                                    ervice, or by allowing the
                                    on its own behalf.  This
                                    OAuth 1.0 protocol described

                                    ent.

   This document is a product of the Internet Engineering Task Force
   (IETF).  It represents the consensus of the IETF community.  It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG).  Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6749
```

Yelp’s Service Mesh ca. 2017



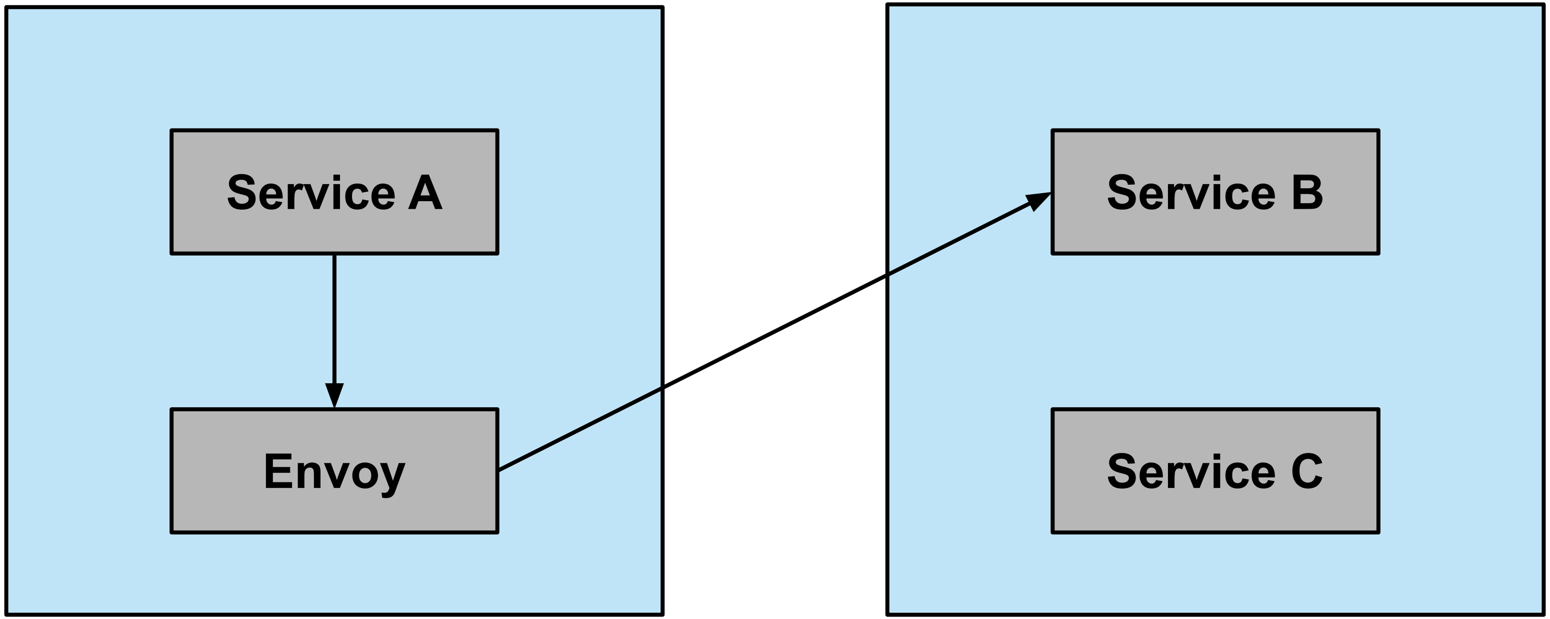
Yelp's Service Mesh ca. 2017



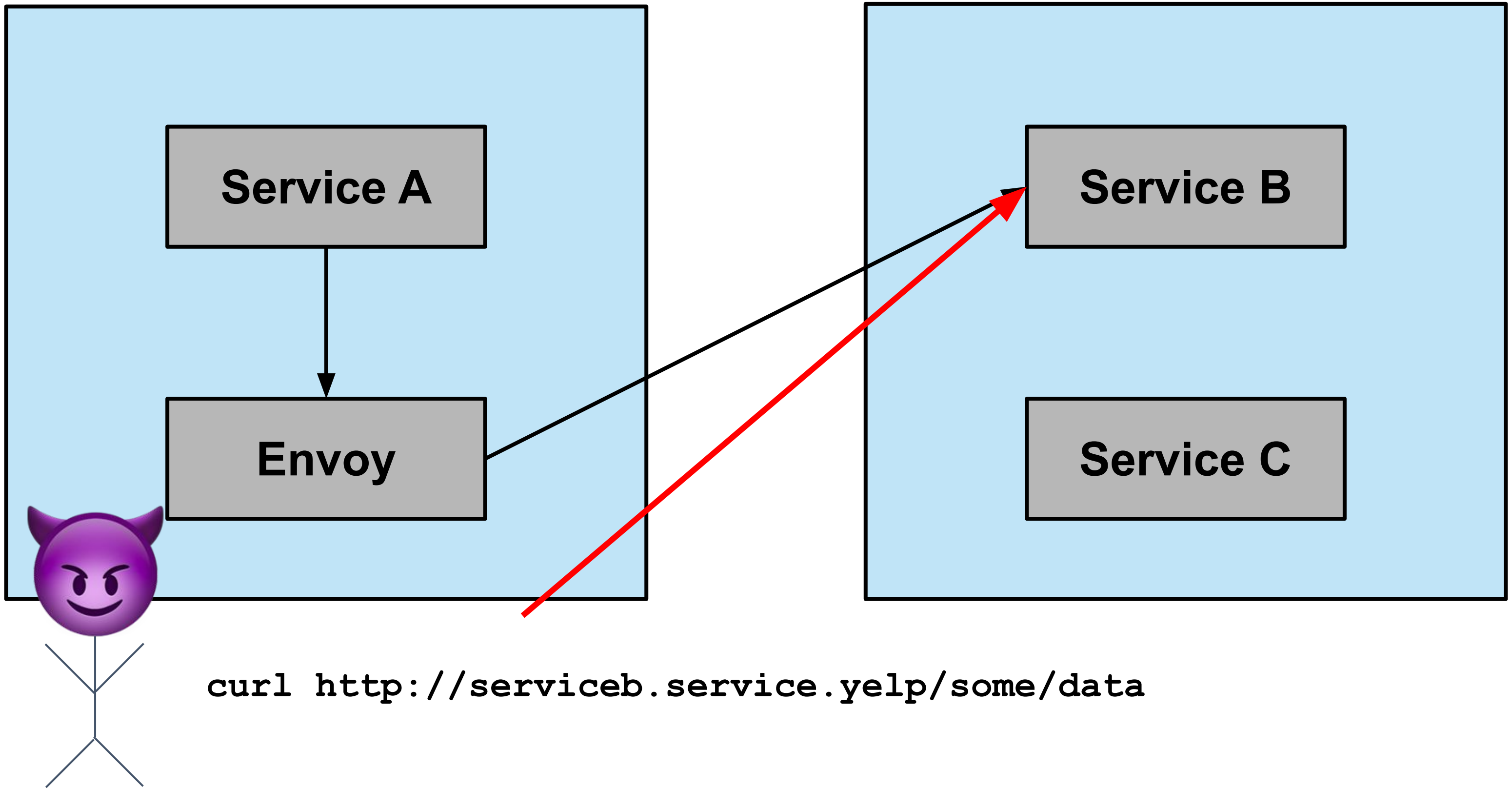
Service AuthN

Service Authentication Evolution

Service AuthN - None

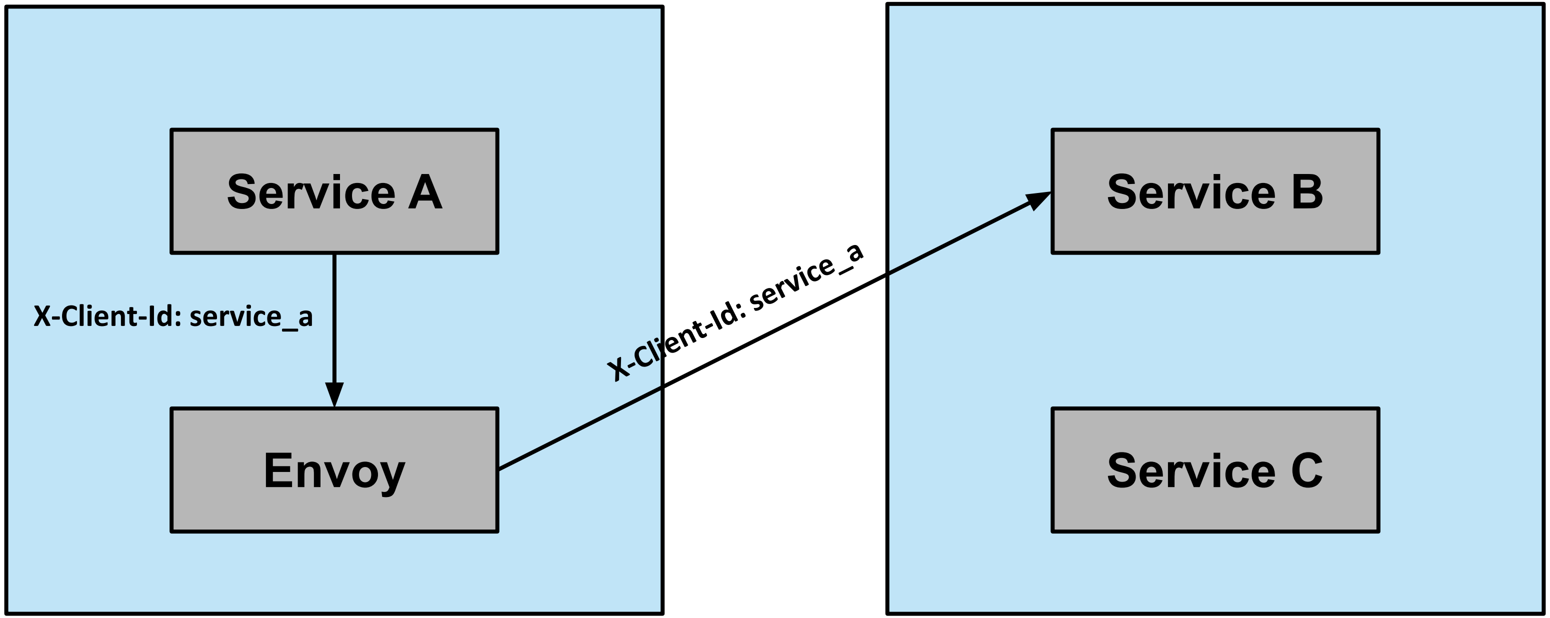


Service AuthN - None

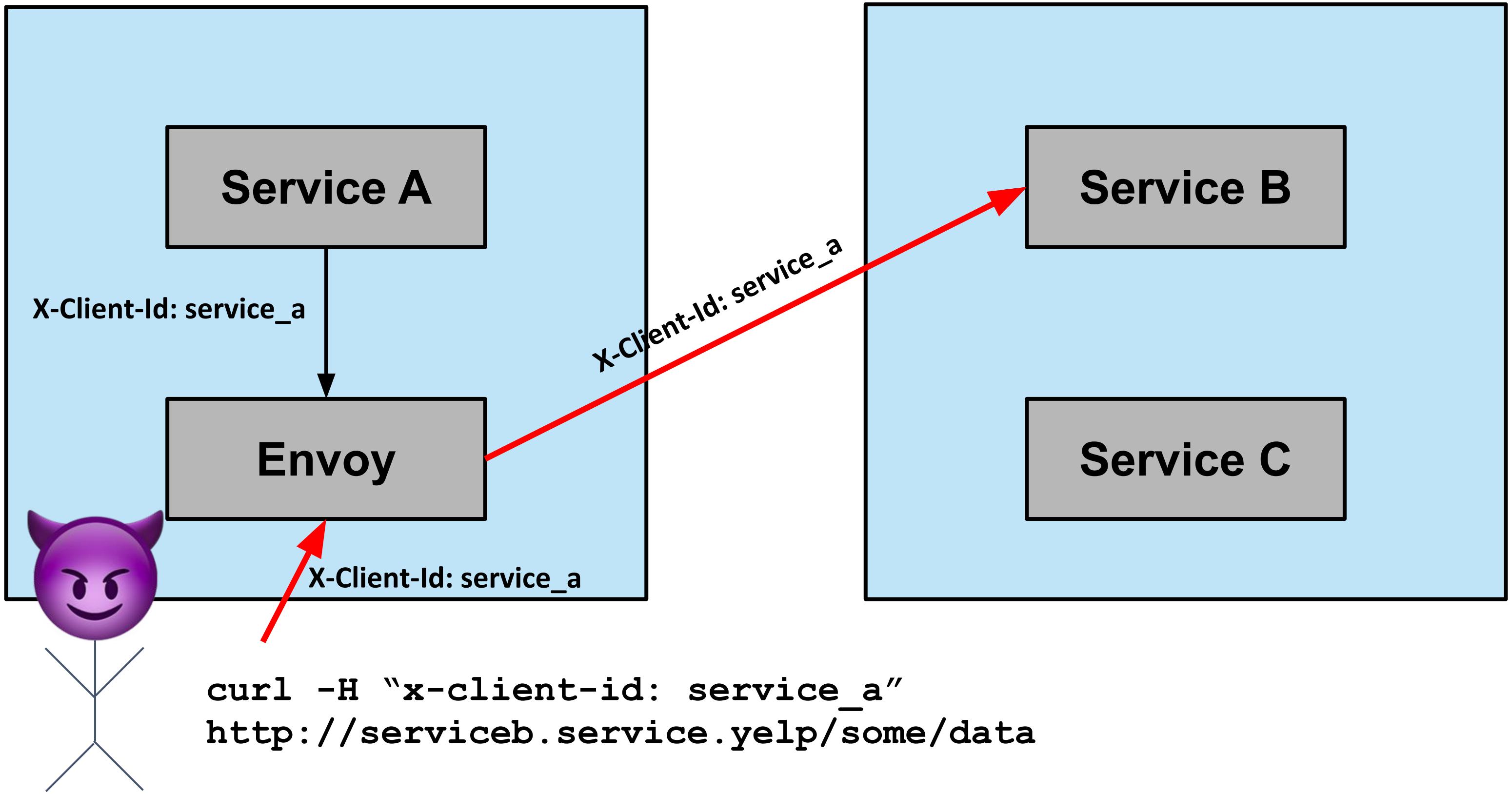


```
curl http://serviceb.service.yelp/some/data
```

Service AuthN - Application

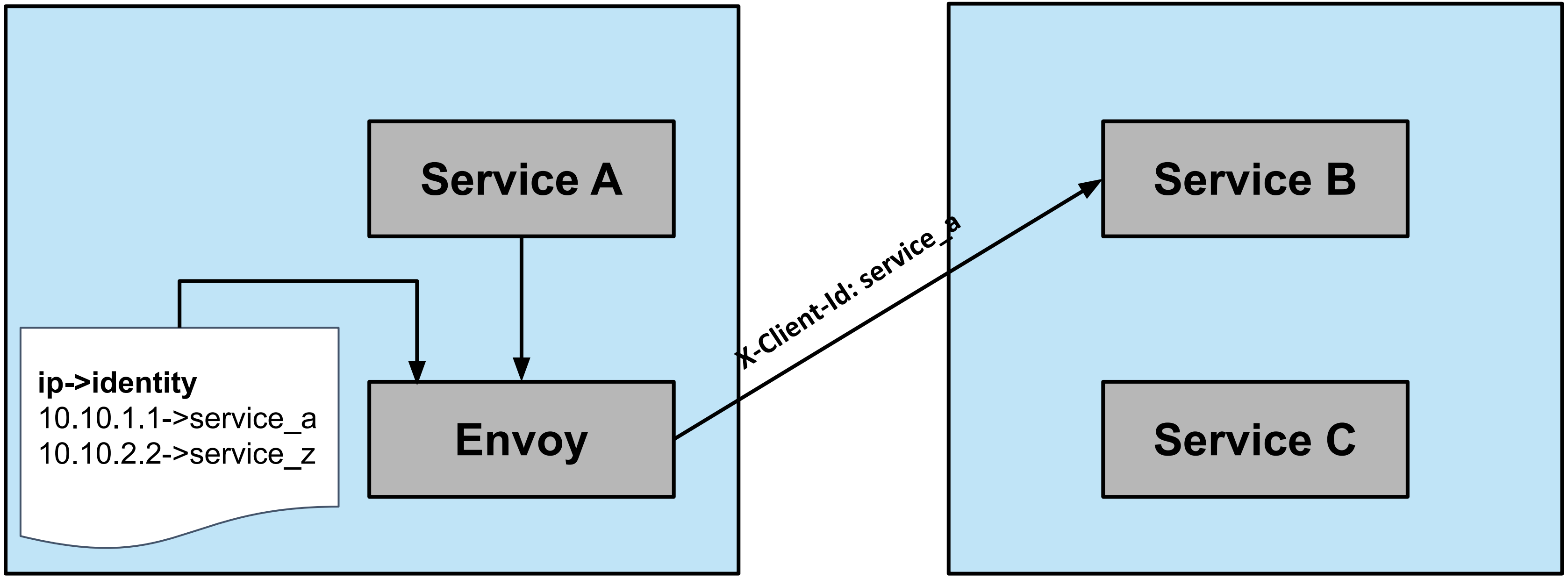


Service AuthN - Application



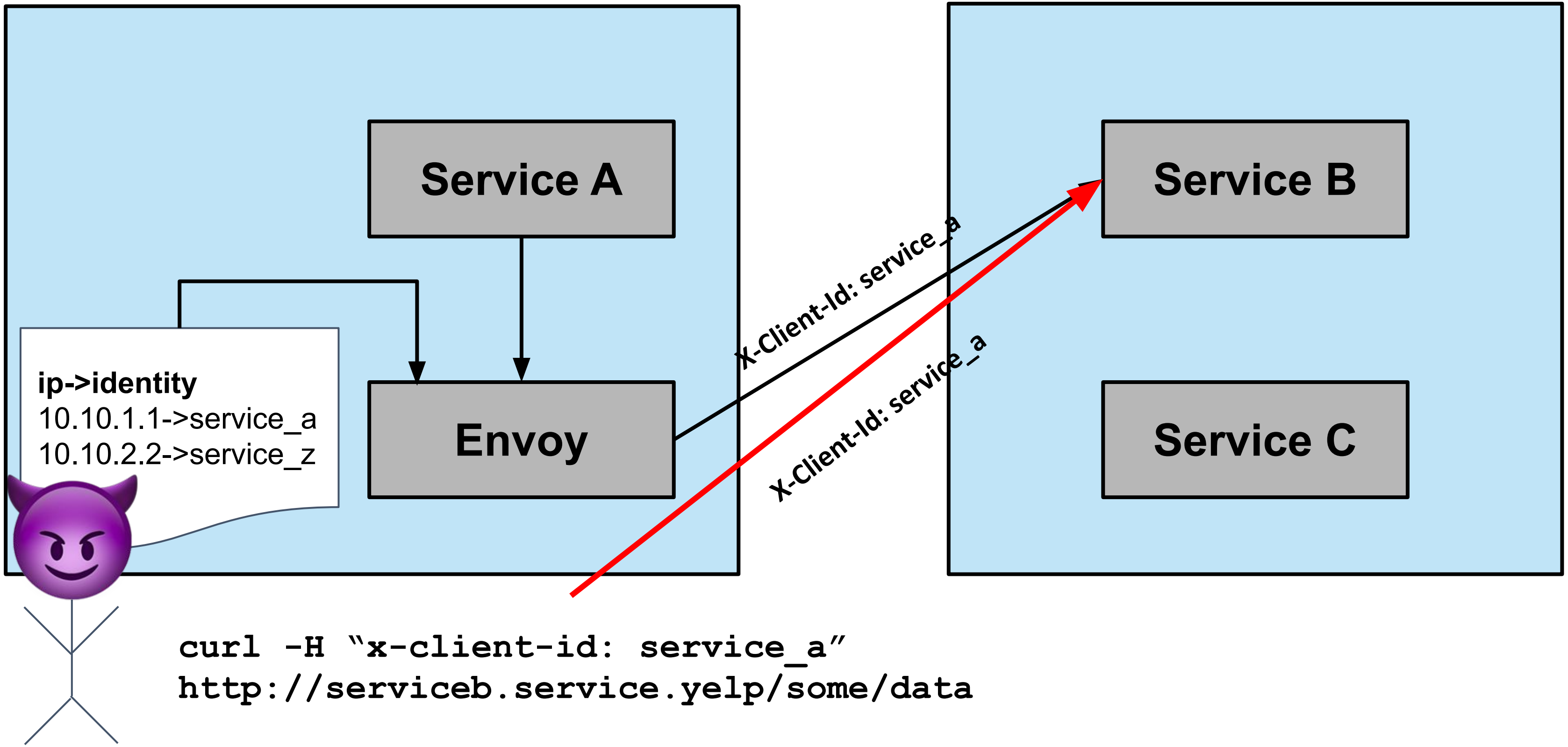
```
curl -H "x-client-id: service_a"
http://serviceb.service.yelp/some/data
```

Service AuthN - Service Mesh



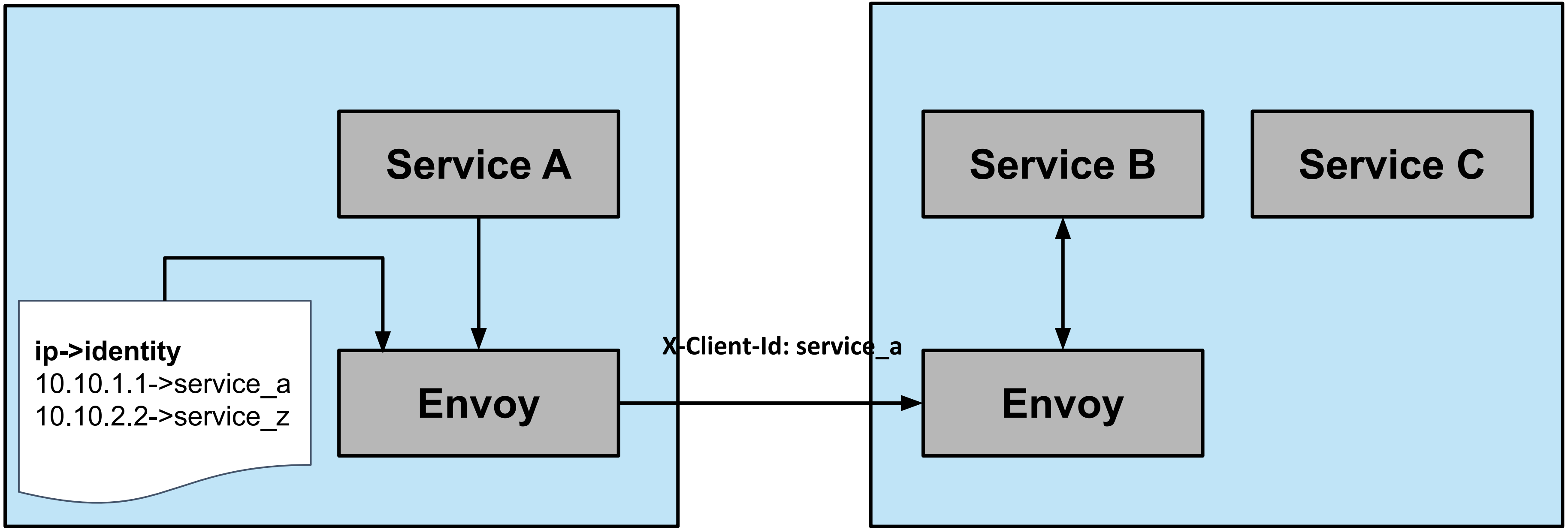
ip->identity
10.10.1.1->service_a
10.10.2.2->service_z

Service AuthN - Service Mesh



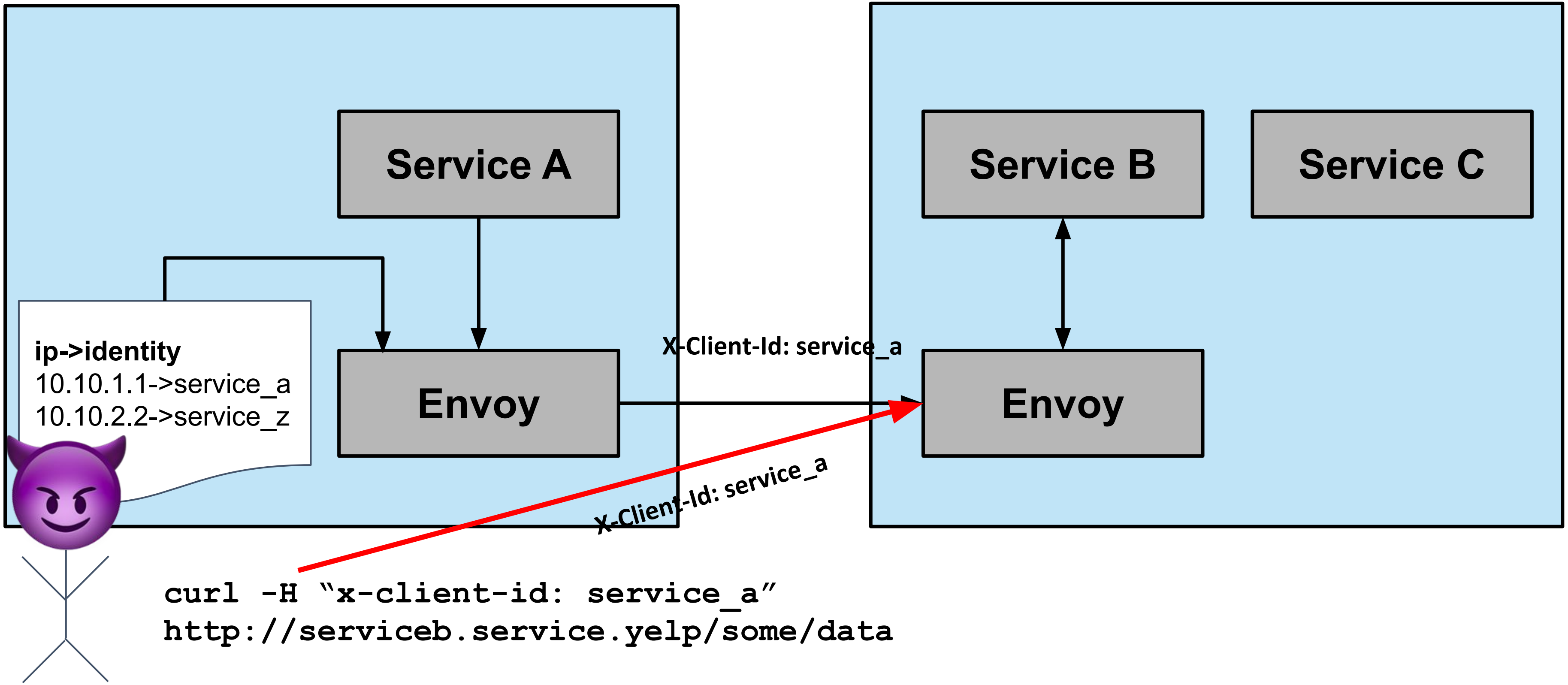
```
curl -H "x-client-id: service_a"
http://serviceb.service.yelp/some/data
```

Service AuthN - Service Mesh



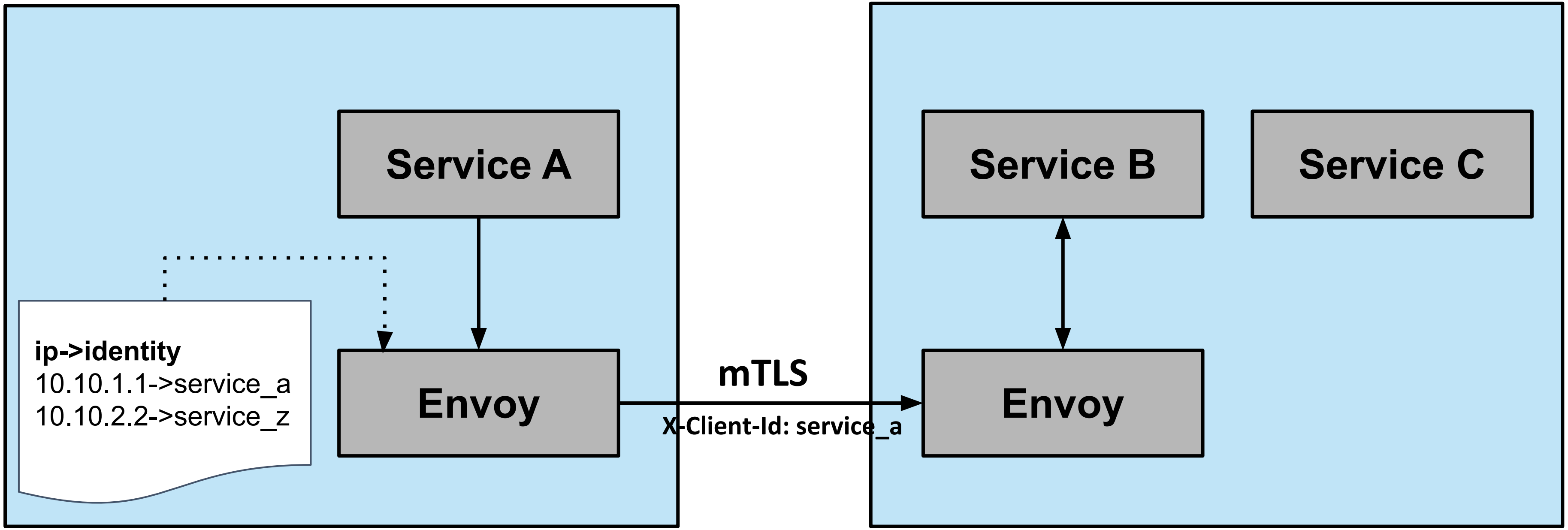
ip->identity
10.10.1.1->service_a
10.10.2.2->service_z

Service AuthN - Service Mesh



```
curl -H "x-client-id: service_a"
http://serviceb.service.yelp/some/data
```

Service AuthN - Service Mesh



ip->identity
10.10.1.1->service_a
10.10.2.2->service_z

Authentication Requirements

✓ Handle identity resolution for computers and humans

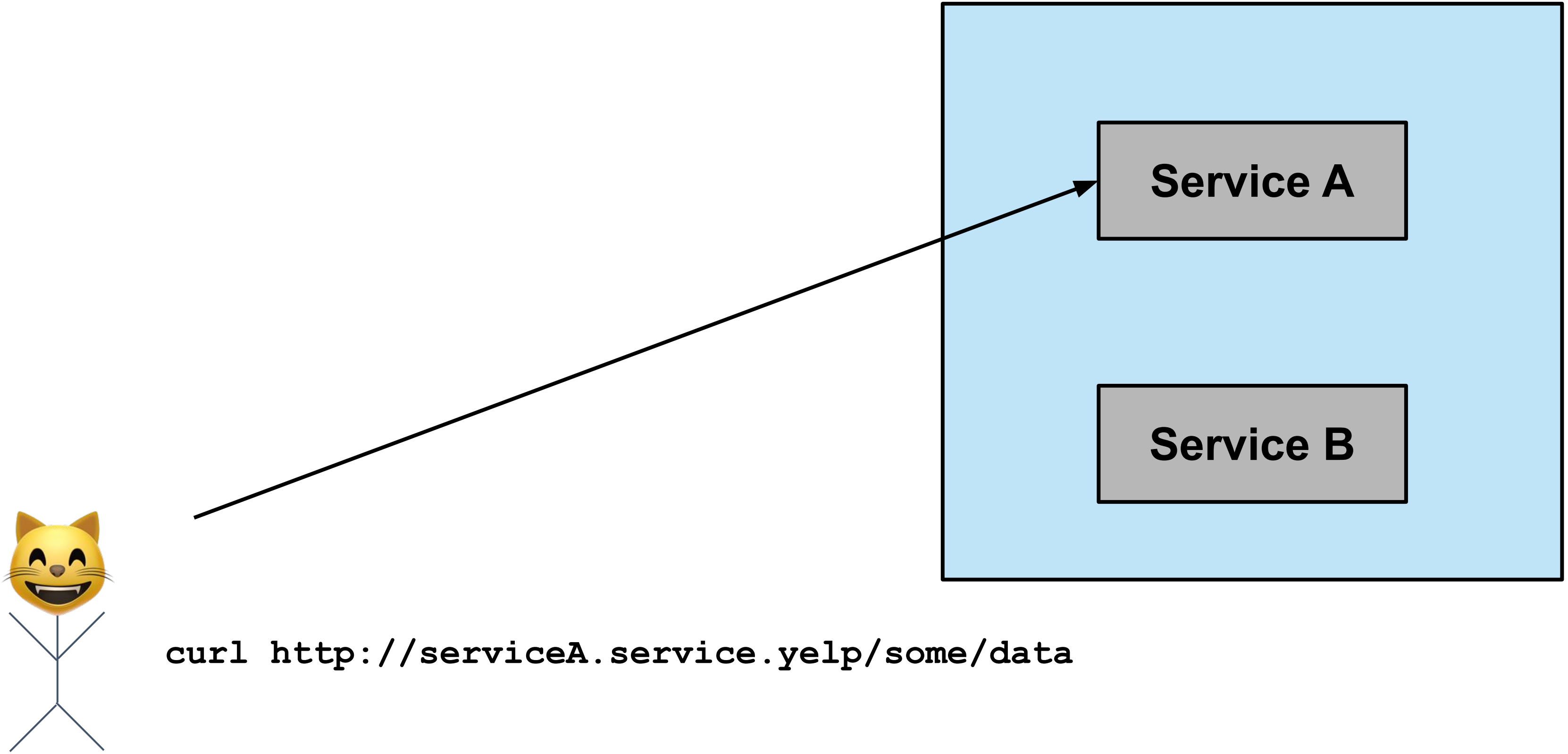
✓ Identity must be unforgeable

✓ Avoid impacting developer velocity

Human Authentication

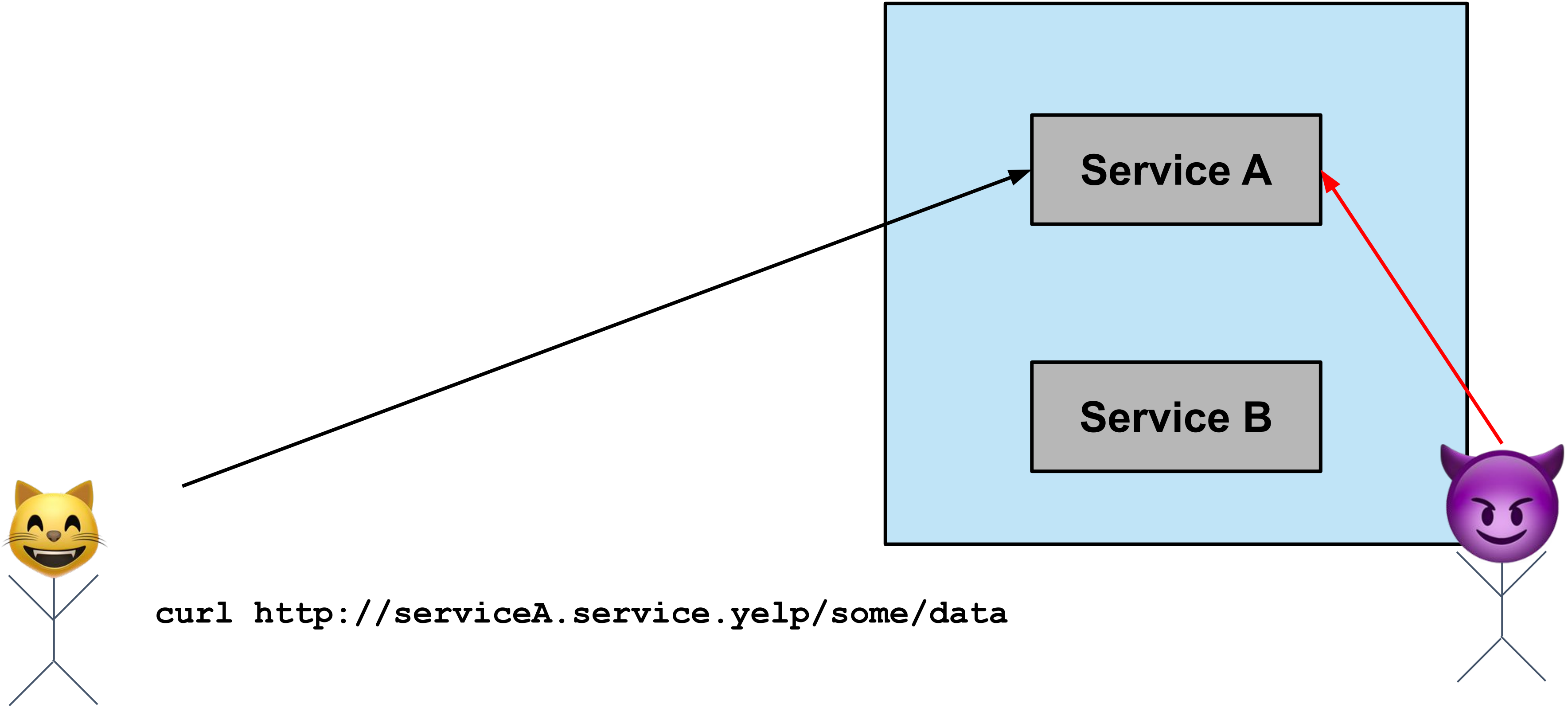
Human Authentication Evolution

Human Authentication - None



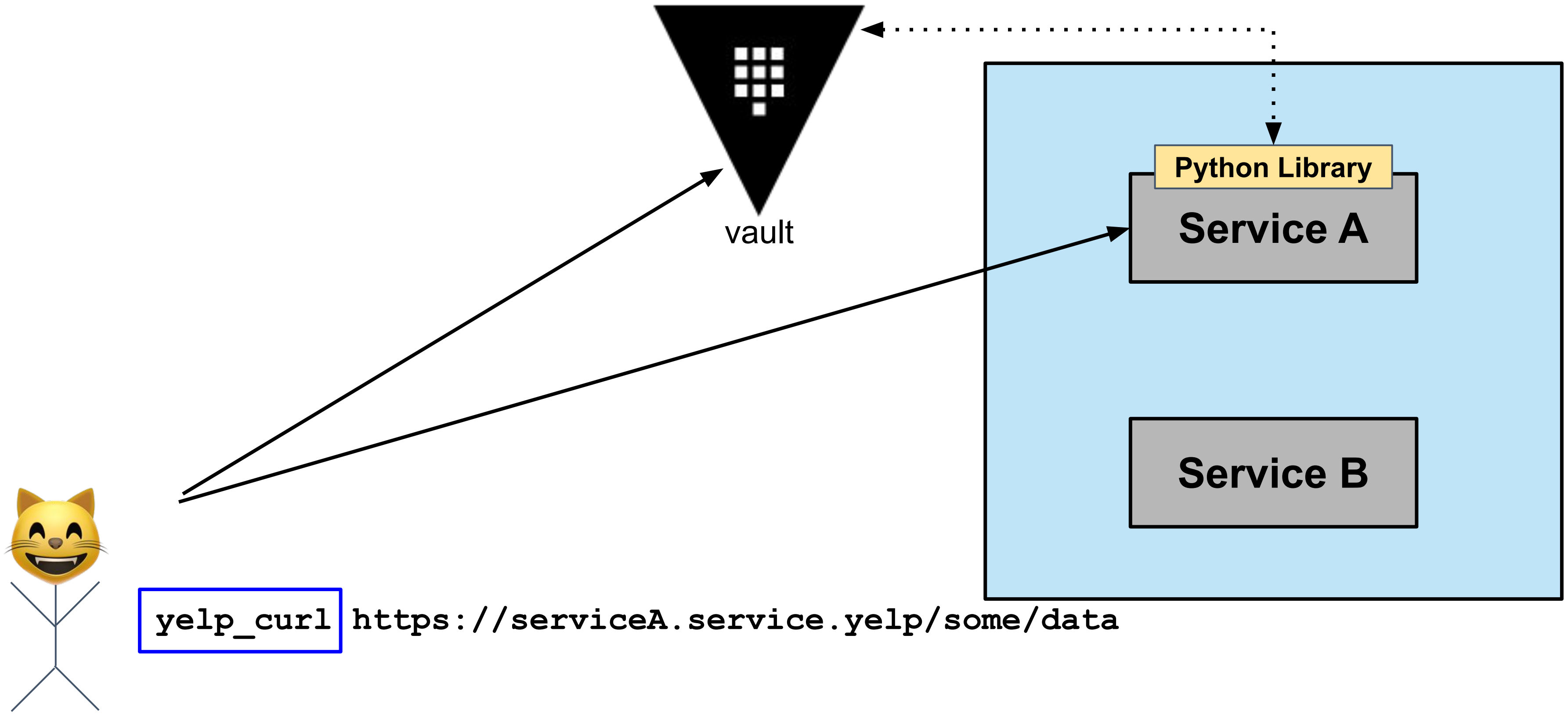
```
curl http://serviceA.service.yelp/some/data
```

Human Authentication - None



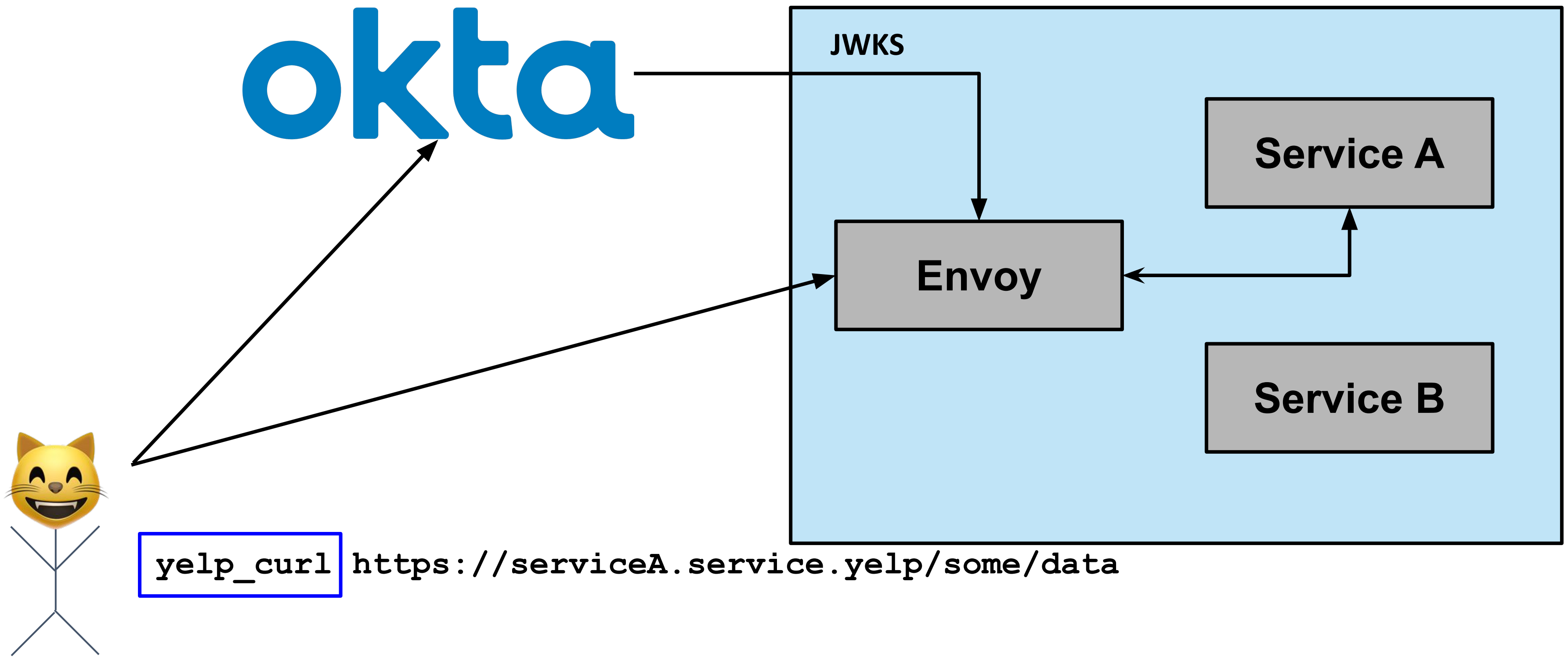
```
curl http://serviceA.service.yelp/some/data
```

Human Authentication - Library



```
yelp_curl https://serviceA.service.yelp/some/data
```

Human Authentication - Service Mesh



`yelp_curl` `https://serviceA.service.yelp/some/data`

Authentication Requirements

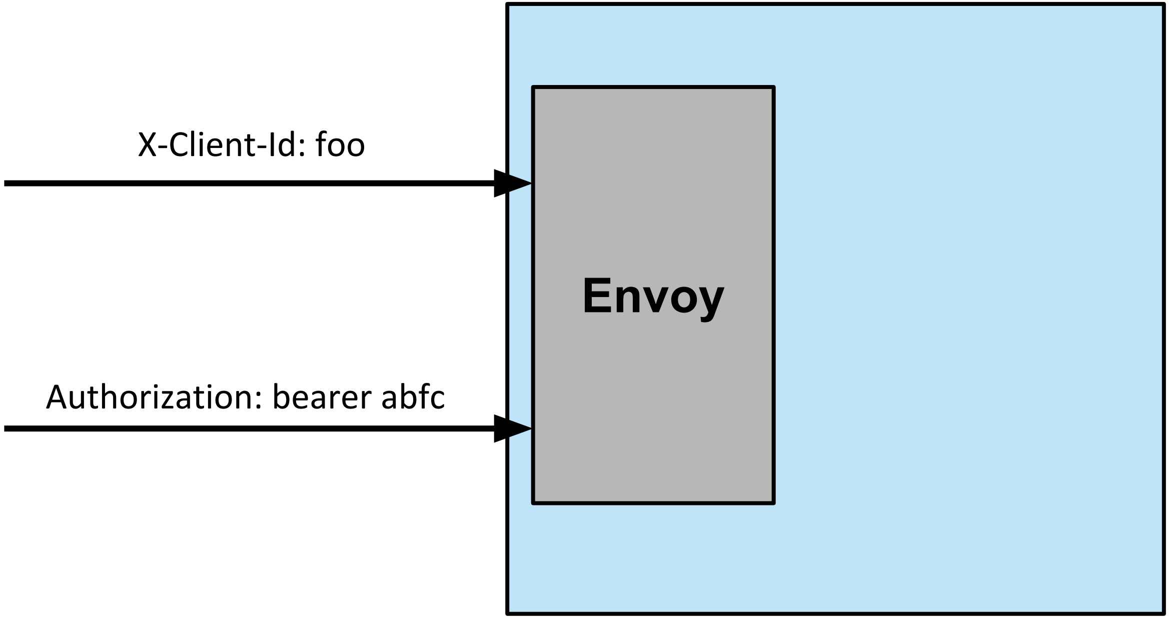
- ✓ Handle identity resolution for computers and humans
- ✓ Identity must be unforgeable
- ✓ Avoid impacting developer velocity

Authentication Summary

- Out of application code!
- mTLS
- Trusted identities

…

Now what?



Authorization

- Are you allowed to do a thing?

Examples:

- AWS IAM Policies
- RBAC
- ABAC
- XACML

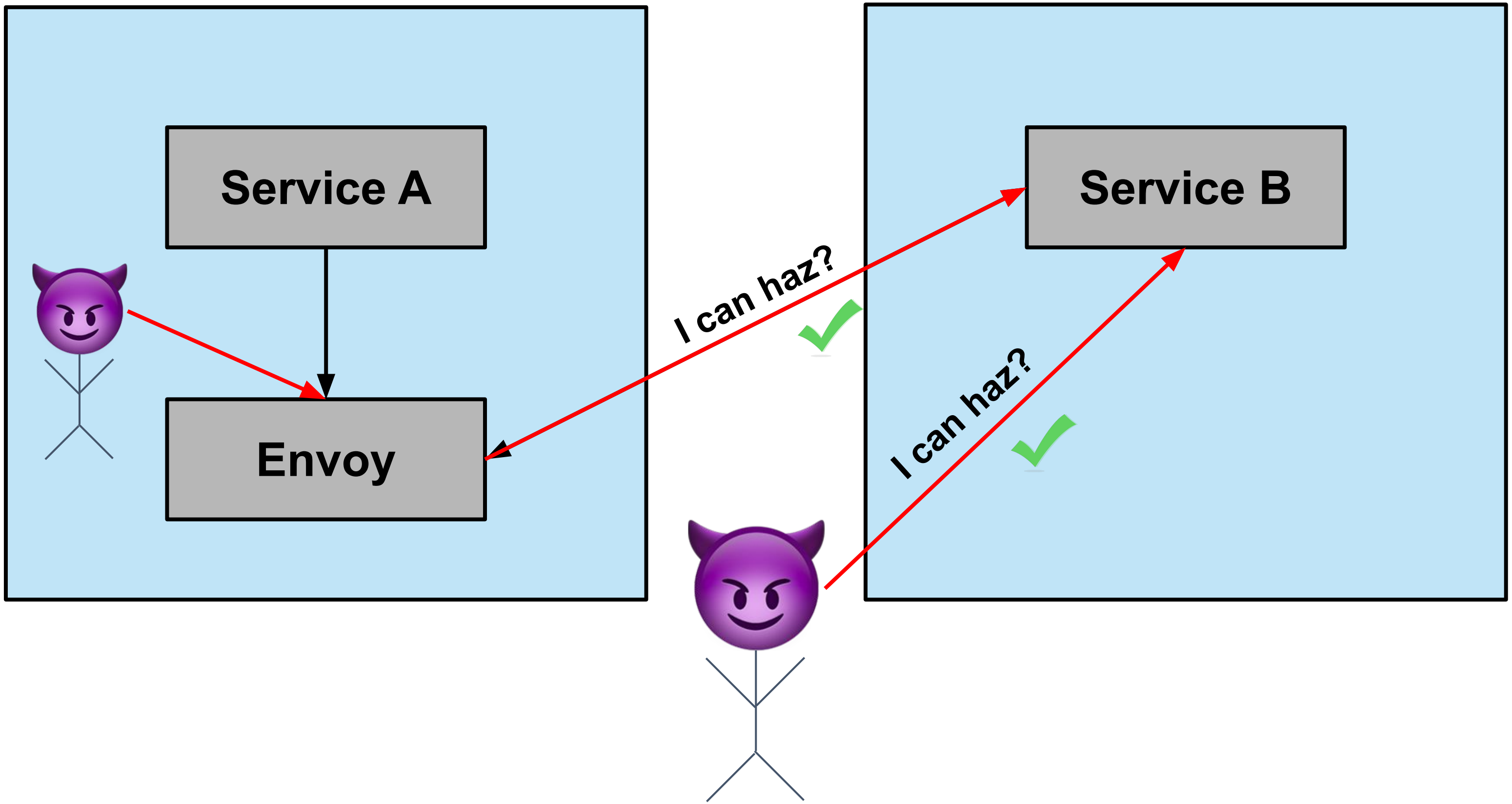
Authorization Requirements

- Prevent unauthorized access to services
- Principle of Least Privilege - **Deny by default**
- Available for any service in our service mesh
- Avoid impacting developer velocity
- Policies should be easy to use and easy to understand

Authorization

Authorization Evolution

Authorization - None

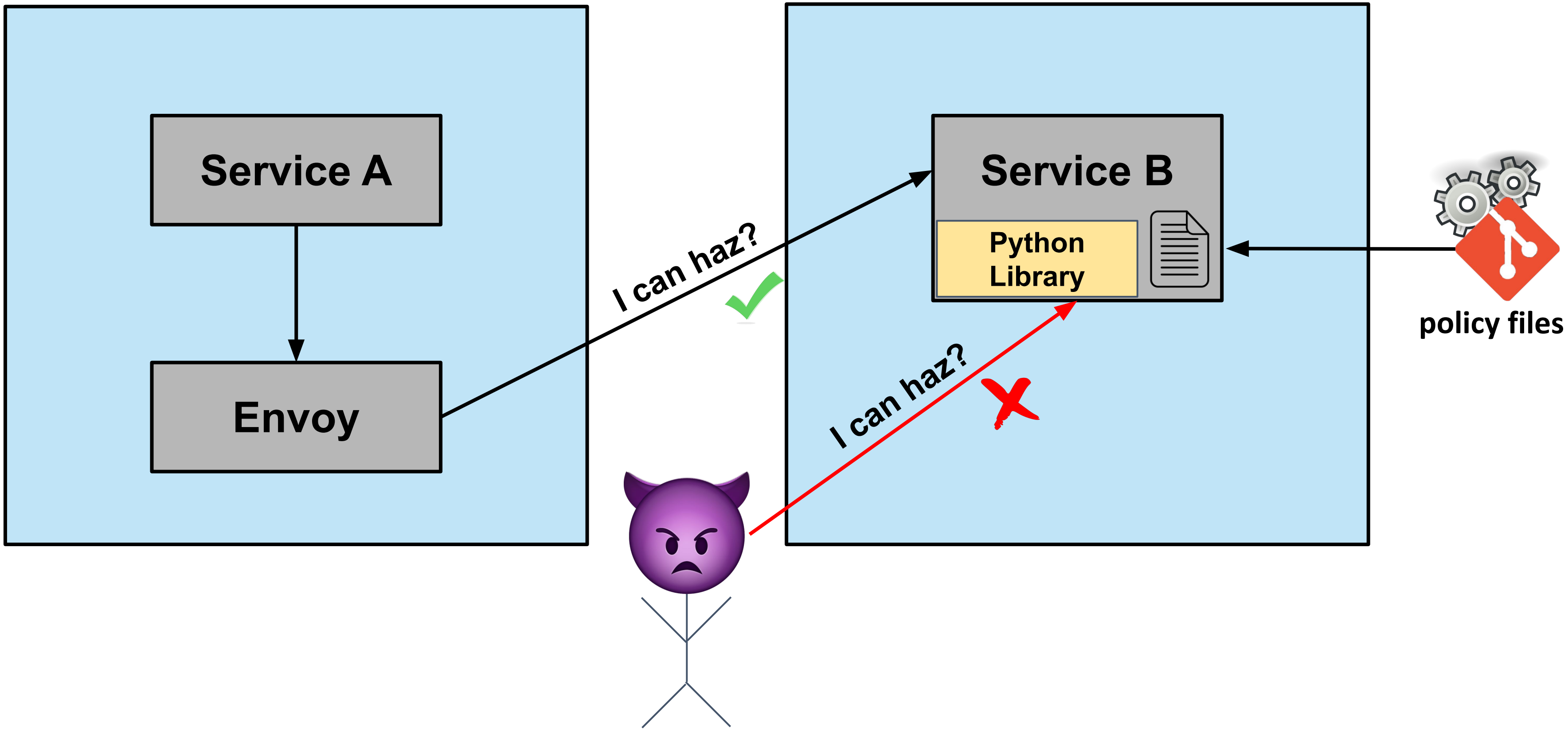


Authorization Requirements

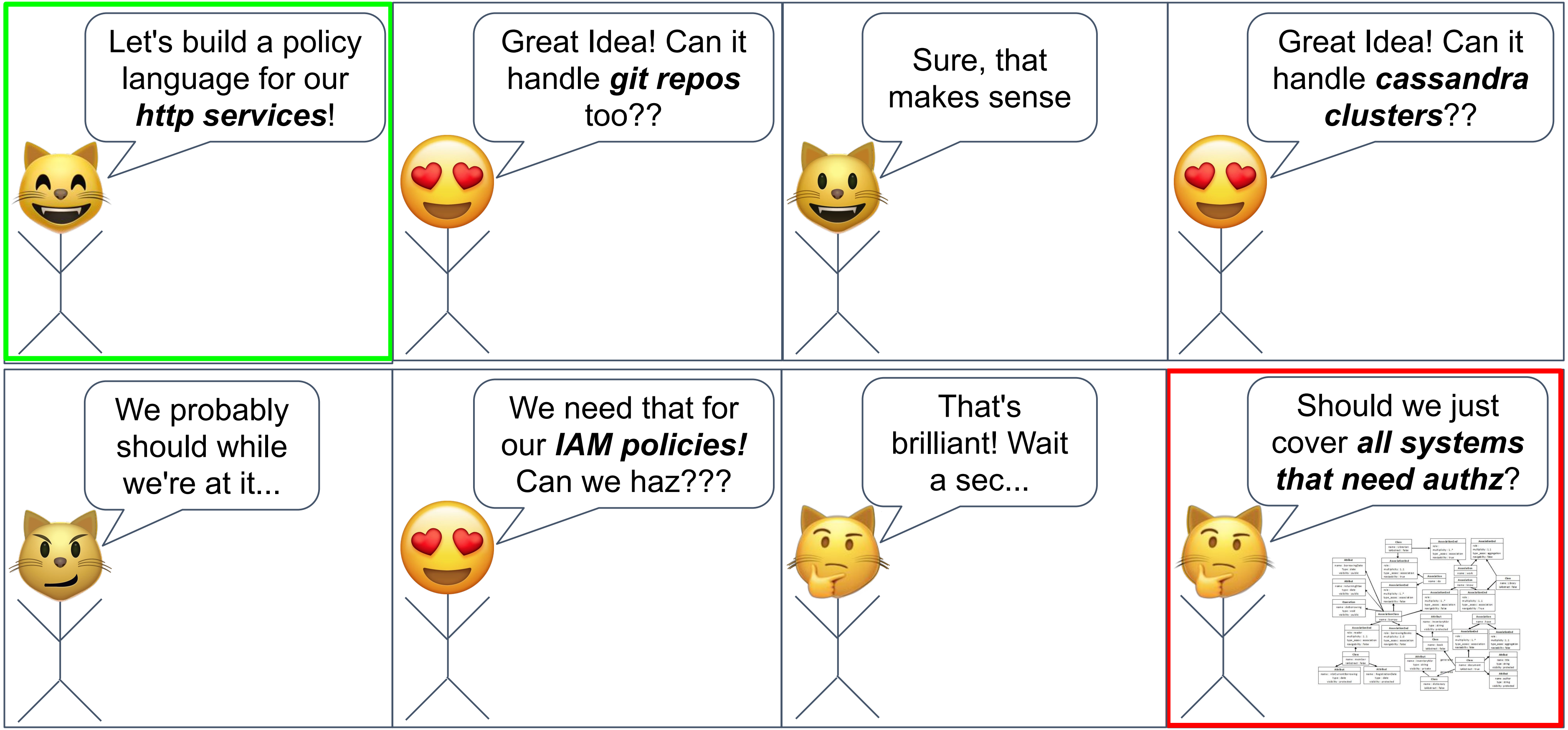
- ✗ Prevent unauthorized access to services
- ✗ Principle of Least Privilege - **Deny by default**
- ✗ Available for any service in our service mesh
- ✓ Avoid impacting developer velocity
- N/A Policies should be easy to use and easy to understand

FAIL

Authorization - Application

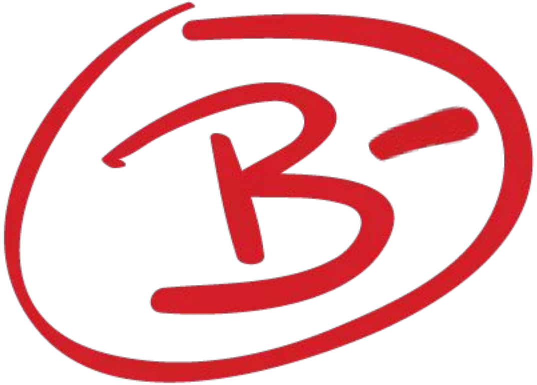


Custom Policy Language - Scope

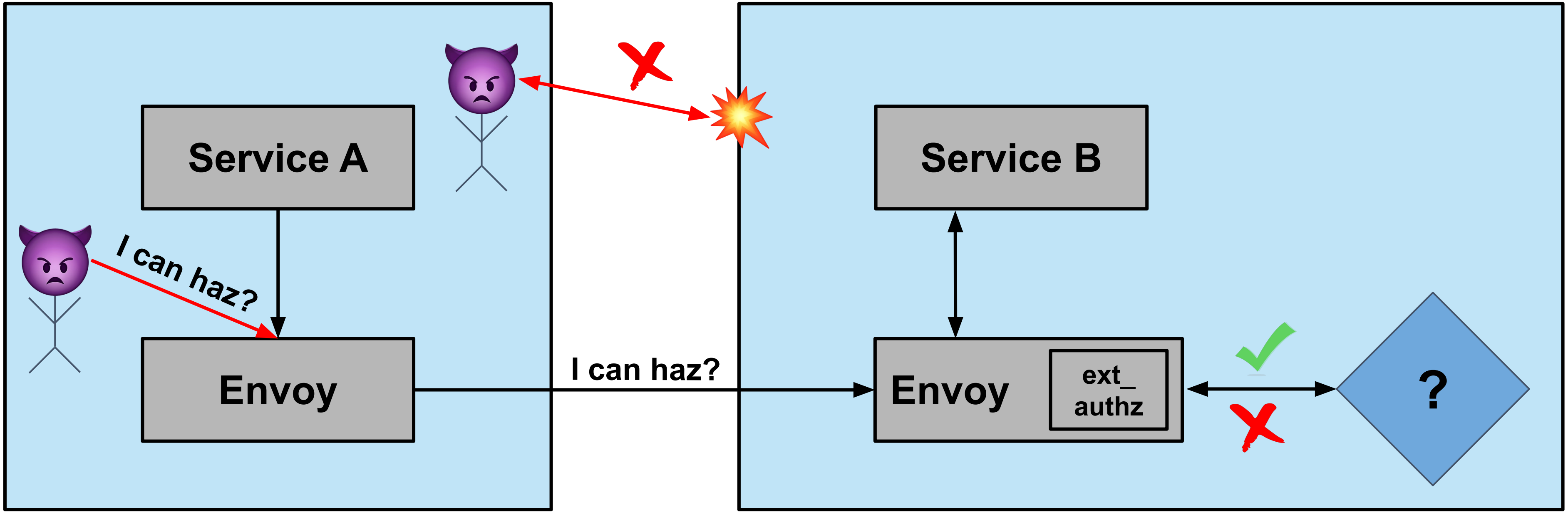

Let's build a policy language for our ***http services***!
Great Idea! Can it handle ***git repos*** too??
Sure, that makes sense
Great Idea! Can it handle ***cassandra clusters***??
We probably should while we're at it...
We need that for our ***IAM policies!*** Can we haz???
That's brilliant! Wait a sec...
Should we just cover ***all systems that need authz***?

Authorization Requirements

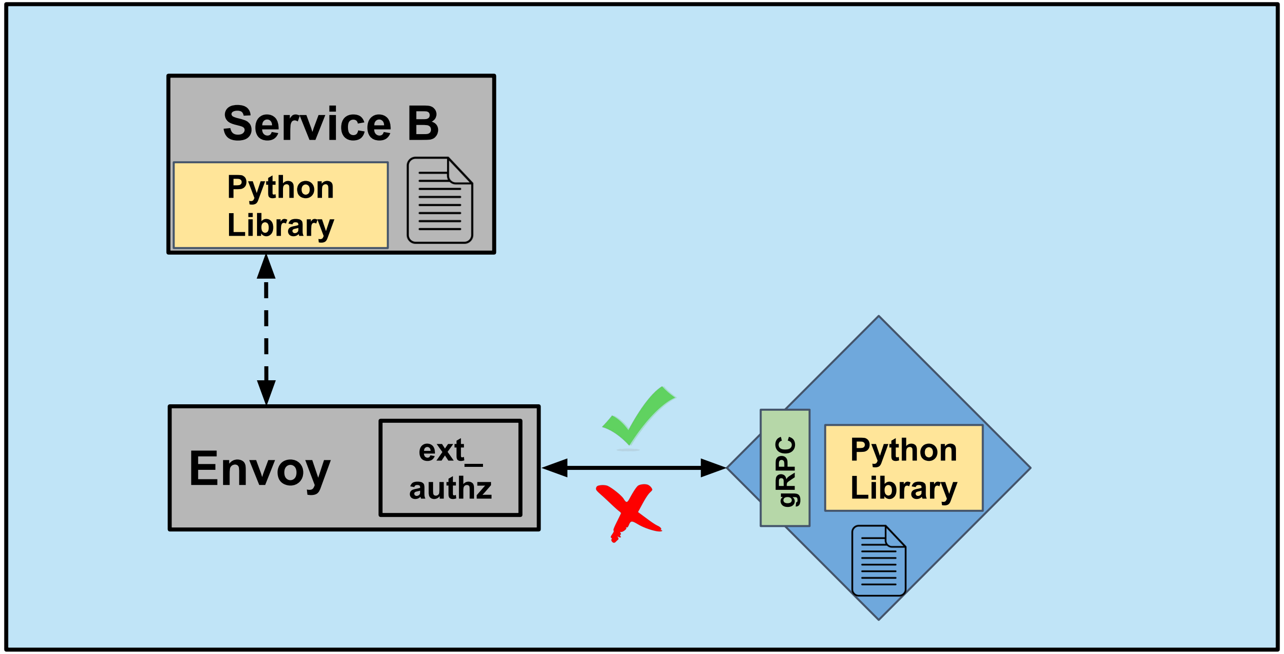
- ✓ Prevent unauthorized access to services
- ? Least Privilege - **Deny by default**
- ✗ Available for any service in our service mesh
- ✗ Avoid impacting developer velocity
- ✗ Easy-to-use and easy-to-understand policies



Authorization - Service Mesh



Extend Our Custom Solution?



Why OPA?

- General-purpose policy language (Rego)
- Comprehensive documentation
- Unit test support for policies
- Open Source Community
  - Active development with short release cycles
  - Quick feedback on github and slack
  - People solving similar problems

Why not OPA?

- We already have something
- Existing policies would need to be migrated
- Steep learning curve for Rego policy language

OPA Policy Manager

Problem

- Steep learning curve for Rego policy language

Solution

- Build an abstraction layer for expressing simple HTTP rules
- Transpile to data structures optimized for fast lookups in OPA
- *Most* engineers don't need to learn Rego
    - But they can if they need to

OPA Policy Manager

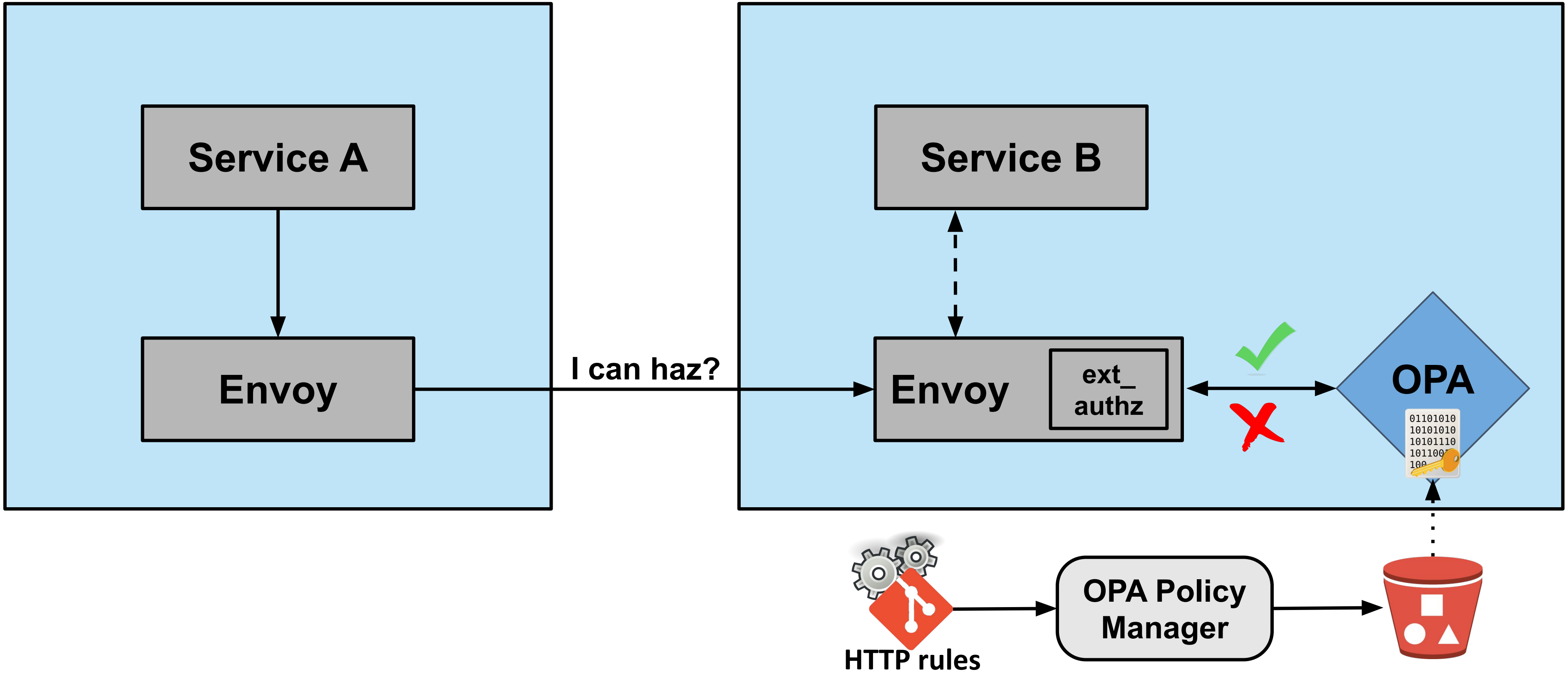
Problem

- Existing policies would need to be migrated

Solution

- Transpile legacy policies to something that OPA could understand

Authorization - Service Mesh



Authorization Requirements

✓ Prevent unauthorized access to services

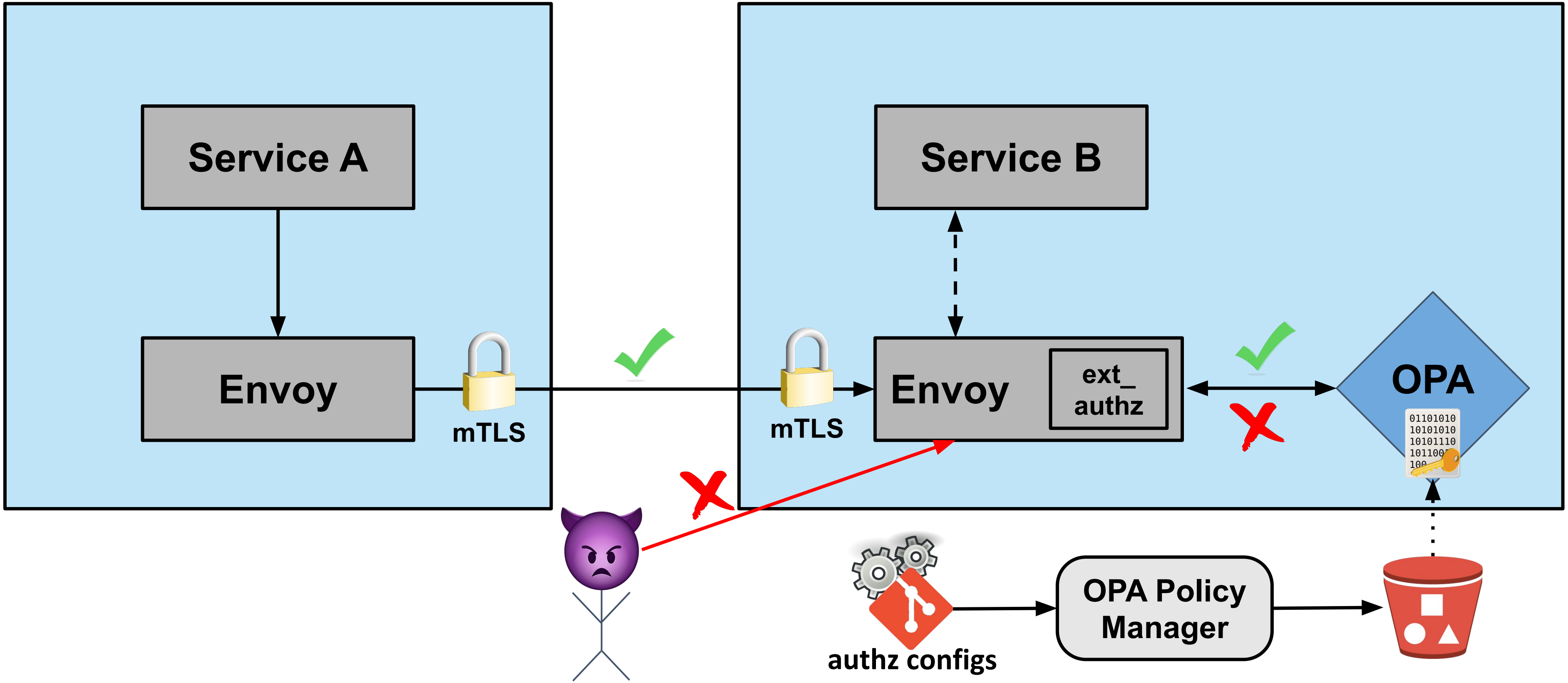
✓ Least Privilege - **Deny by default**

✓ Available for any service in our service mesh

✓ Avoid impacting developer velocity

✓ Easy-to-use and easy-to-understand policies

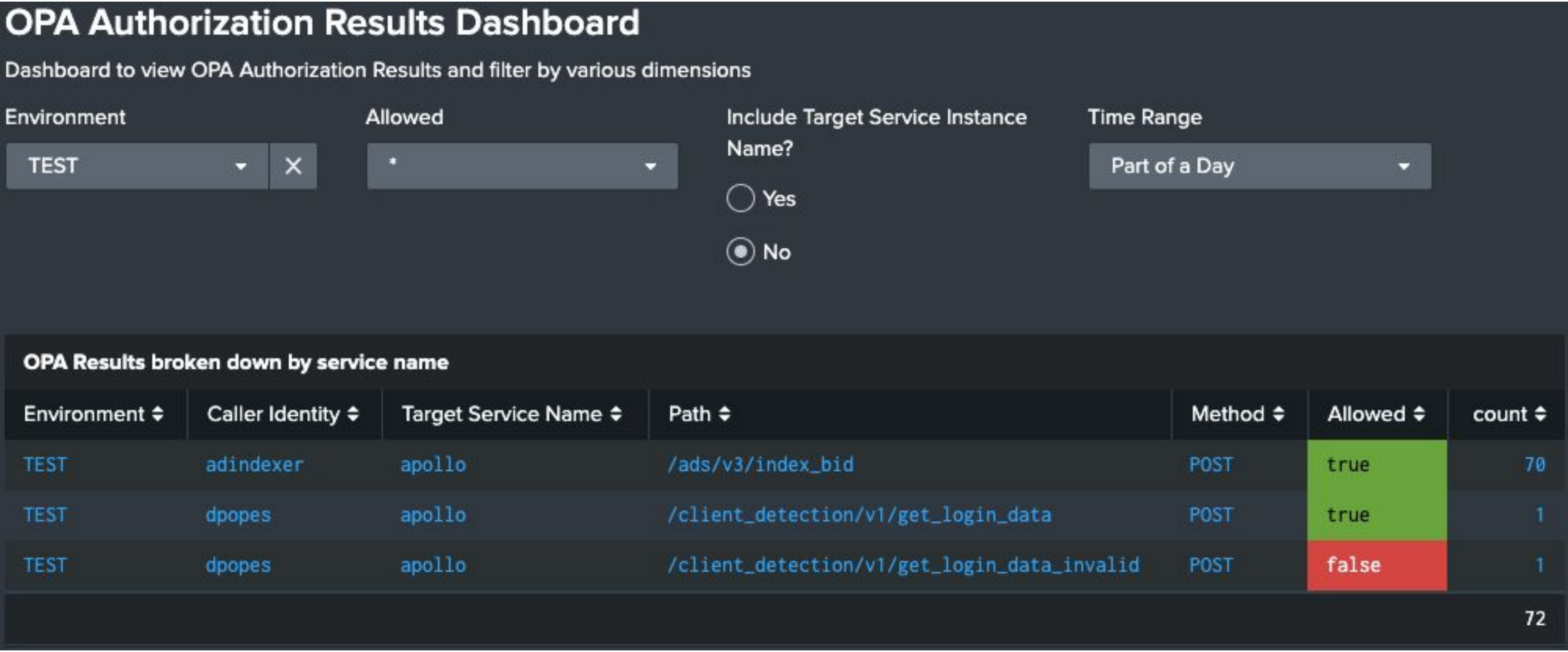
Authorization + Authentication



Where are we now?

- OPA deployed everywhere
- Requests to sensitive services are authorized
- Monitoring and Alerting
    - OPA Decision logs stream to SPLUNK
    - Alert for spikes in authorization failures
    - Dashboards to visualize authorization results

Authorization Results Dashboard



OPA Authorization Results Dashboard

Dashboard to view OPA Authorization Results and filter by various dimensions

Environment: TEST

Allowed: *

Include Target Service Instance Name?

○ Yes

● No

Time Range: Part of a Day

OPA Results broken down by service name

Environment	Caller Identity	Target Service Name	Path	Method	Allowed	count
TEST	adindexer	apollo	/ads/v3/index_bid	POST	true	70
TEST	dpopes	apollo	/client_detection/v1/get_login_data	POST	true	1
TEST	dpopes	apollo	/client_detection/v1/get_login_data_invalid	POST	false	1
						72

Future Ideas - Service Mesh

Authorization

- Phased Policy Rollouts
  - Service mesh config to roll out policy updates to “canary” OPA instances (~1% of traffic)
  - Monitor for deltas in authz distribution
  - Promote to “primary” OPA instances [automatically?]

Authentication

- Improvements to identity attestation
  - Client Certificates?
  - SPIRE?

Future Ideas - Other Use Cases

Deploy more OPA instances for more use cases

- K8s Admission Controller
- Docker Authz
- SSH and sudo
- Terraform
- Kafka

Key Takeaways

- Security in the service mesh makes Yelp services secure by default
- Incremental changes are necessary when making big tech leaps
- Automate migrations so your end users don't have to
- Start from the use case, and be mindful of scope creep
- OPA is a powerful building block

Thanks!

Questions?

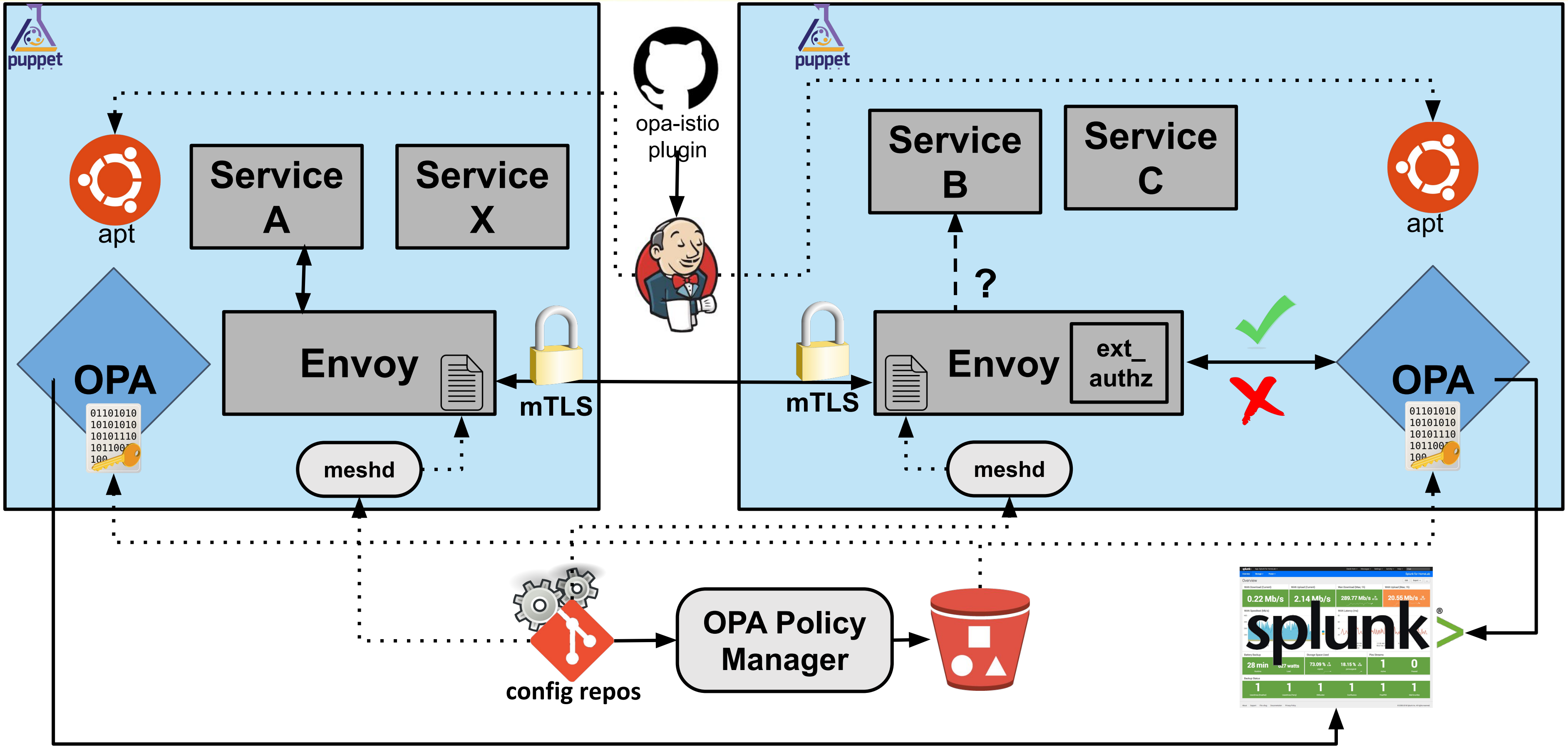
Appendix

FAQ

Q: Why not use client certificates for humans and services?
A: At the time of design, we did not have sufficient PKI infra in place

Q: Why not use SPIFFE/SPIRE?
A: Mostly complications from our multi-tenancy architecture

OPA at Yelp


puppet
apt
Service A
Service X
OPA
01101010
10101010
10101110
101100
100
Envoy
mTLS
meshd
opa-istio plugin
puppet
Service B
Service C
?
mTLS
Envoy
ext_ authz
OPA
01101010
10101010
10101110
101100
100
apt
meshd
config repos
OPA Policy Manager
0.22 Mb/s
2.14 Mb/s
289.77 Mb/s
20.55 Mb/s
splunk>
28 min
73.09 %
18.15 %