Fundamentals of Istio

What is Istio?

- An open platform to connect, manage, and secure microservices.
- Supports:
  - load balancing
  - service-to-service authentication
  - monitoring
  - more
- No changes to service code!

What is Istio?

- Traffic management
  - Control the flow of traffic and API calls between services, make calls more reliable, and make the network more robust in the face of adverse conditions.

- Observability
  - Gain understanding of the dependencies between services and the nature and flow of traffic between them, providing the ability to quickly identify issues.

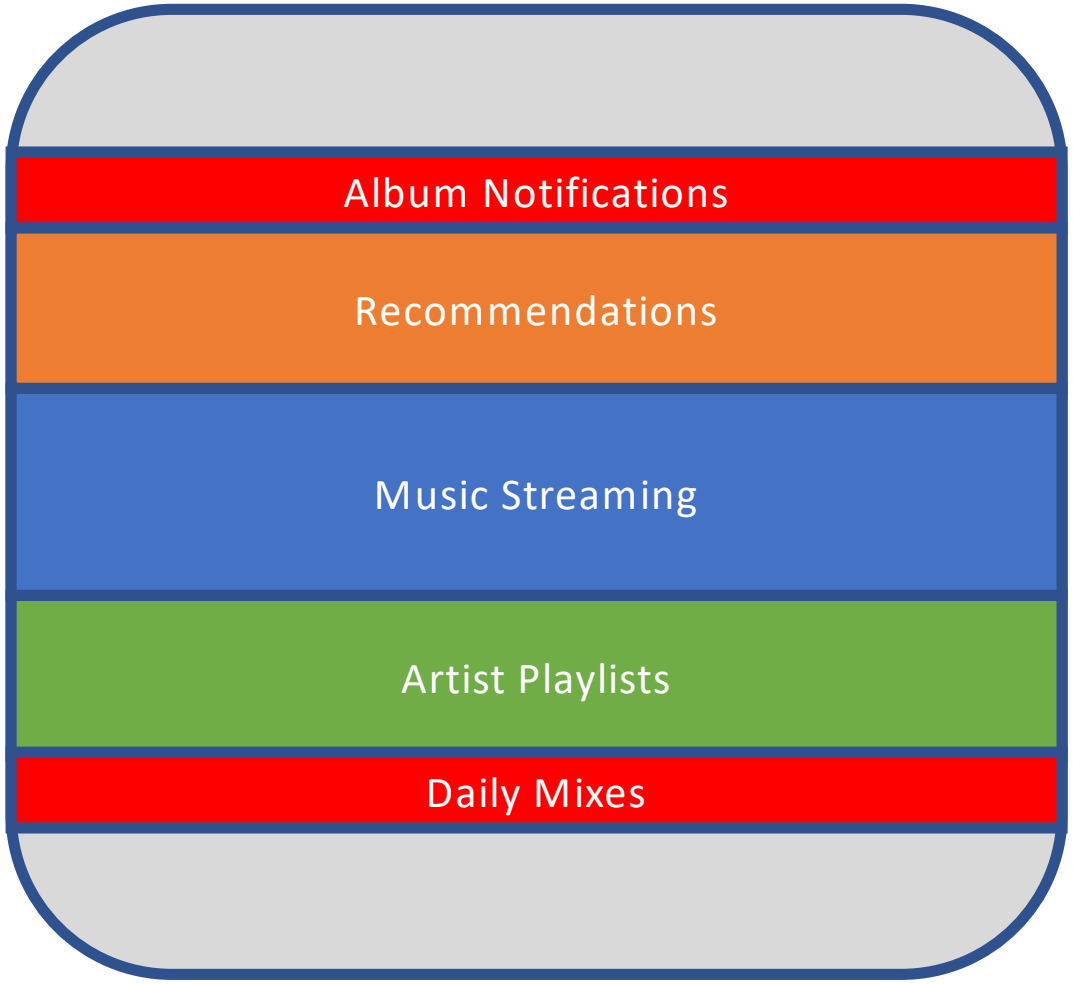
What is Istio?

- Policy Enforcement
  - Apply organizational policy to the interaction between services, ensure access policies are enforced and resources are fairly distributed among consumers. Policy changes are made by configuring the mesh, not by changing application code.
- Service Identity and Security
  - Provide services in the mesh with a verifiable identity and provide the ability to protect service traffic as it flows over networks of varying degrees of trustability.

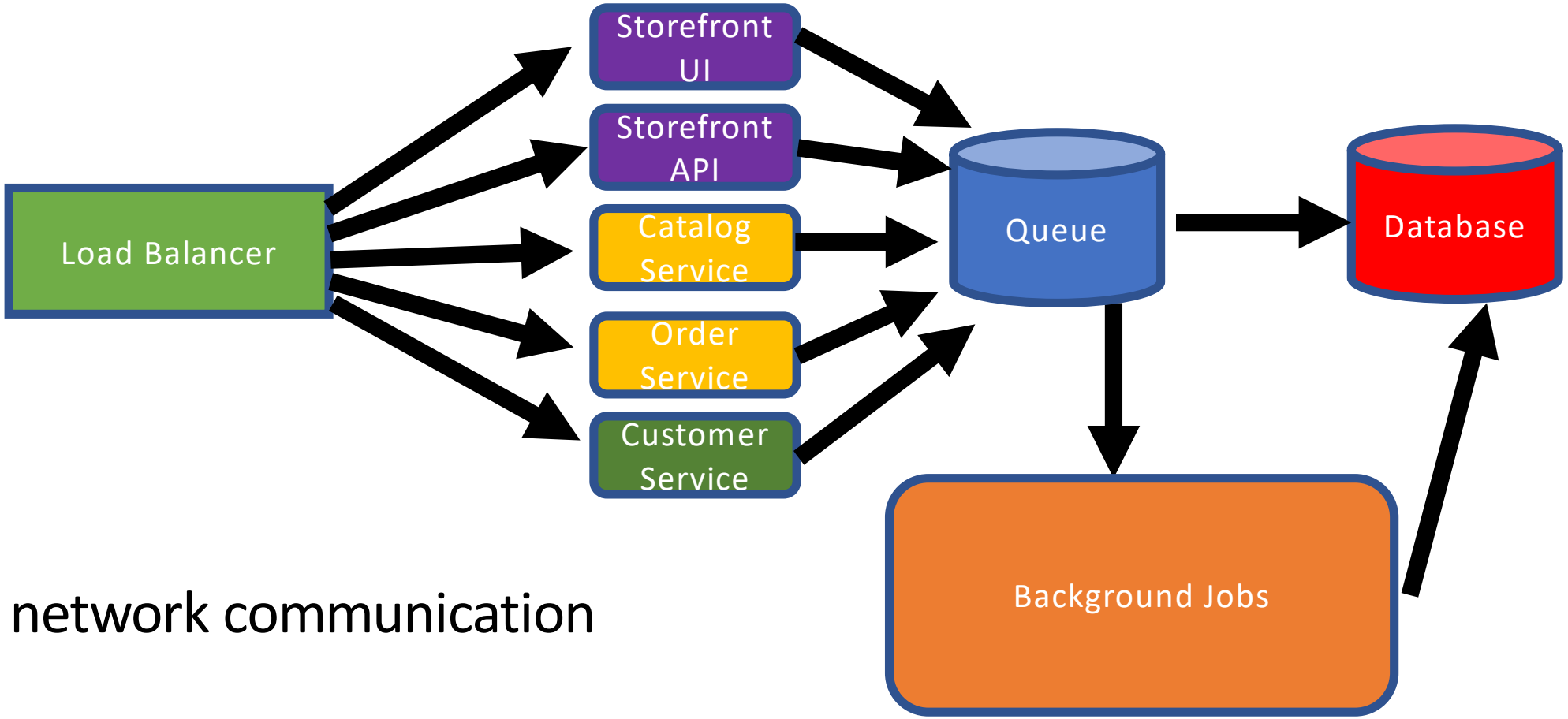
Microservices Challenges Review

Monolithic Architecture

- Components communicate seamlessly
- Easy to monitor
- Easy to determine where error is
- Communication inside monolith is secured



Microservices Architecture

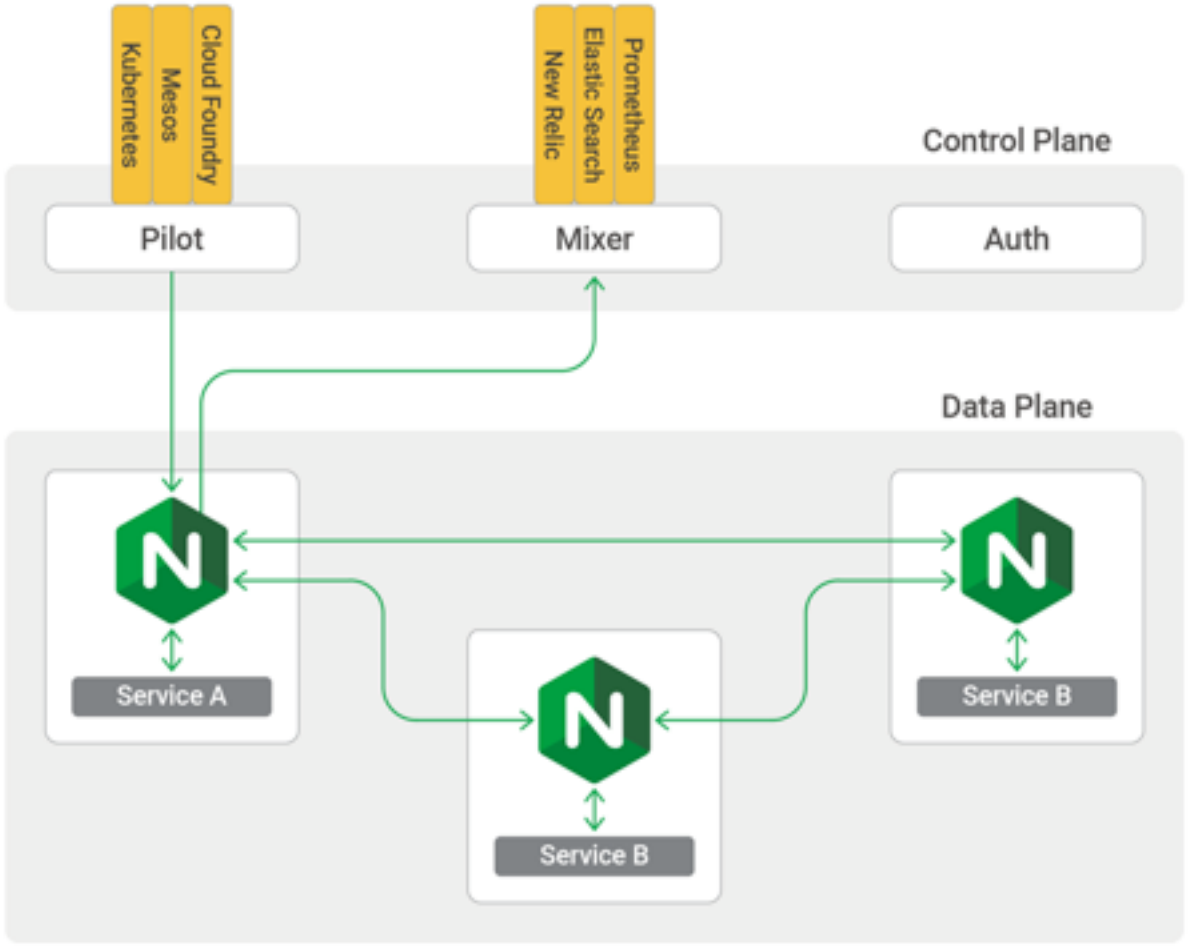


- Insecure network communication
- Visibility
- Policy enforcement

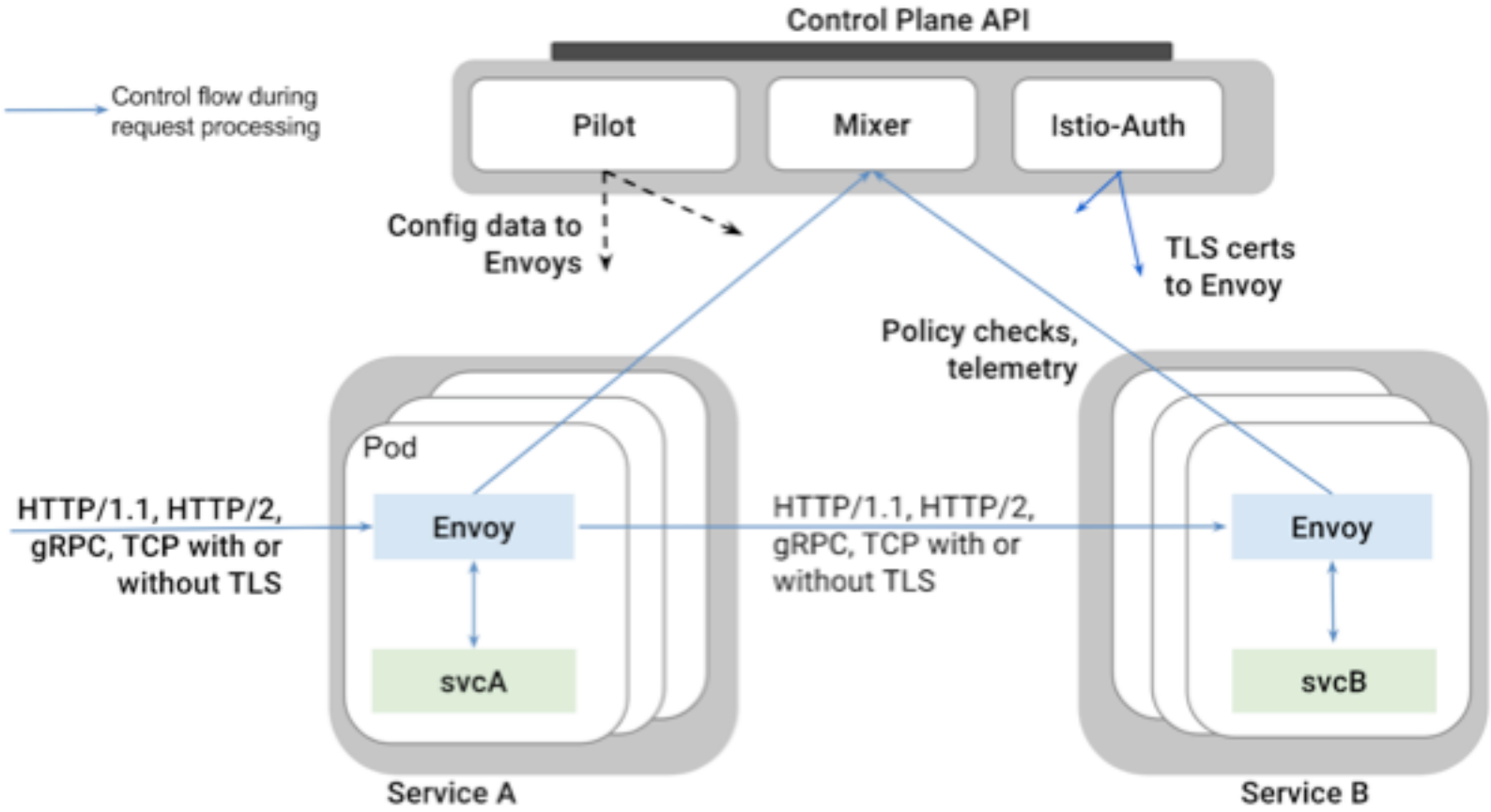
Service Mesh

- Dedicated infrastructure layer for making service-to-service communication safe, fast, and reliable.
- Array of lightweight network proxies
  - Deployed alongside application code
- Encryption
  - Developers no longer have to add encrypt/decrypt code to their applications
- Circuit breaker pattern support
  - Service signals it's "unhealthy", service mesh removes it, then when fixed adds it back in to pool.

Service Mesh

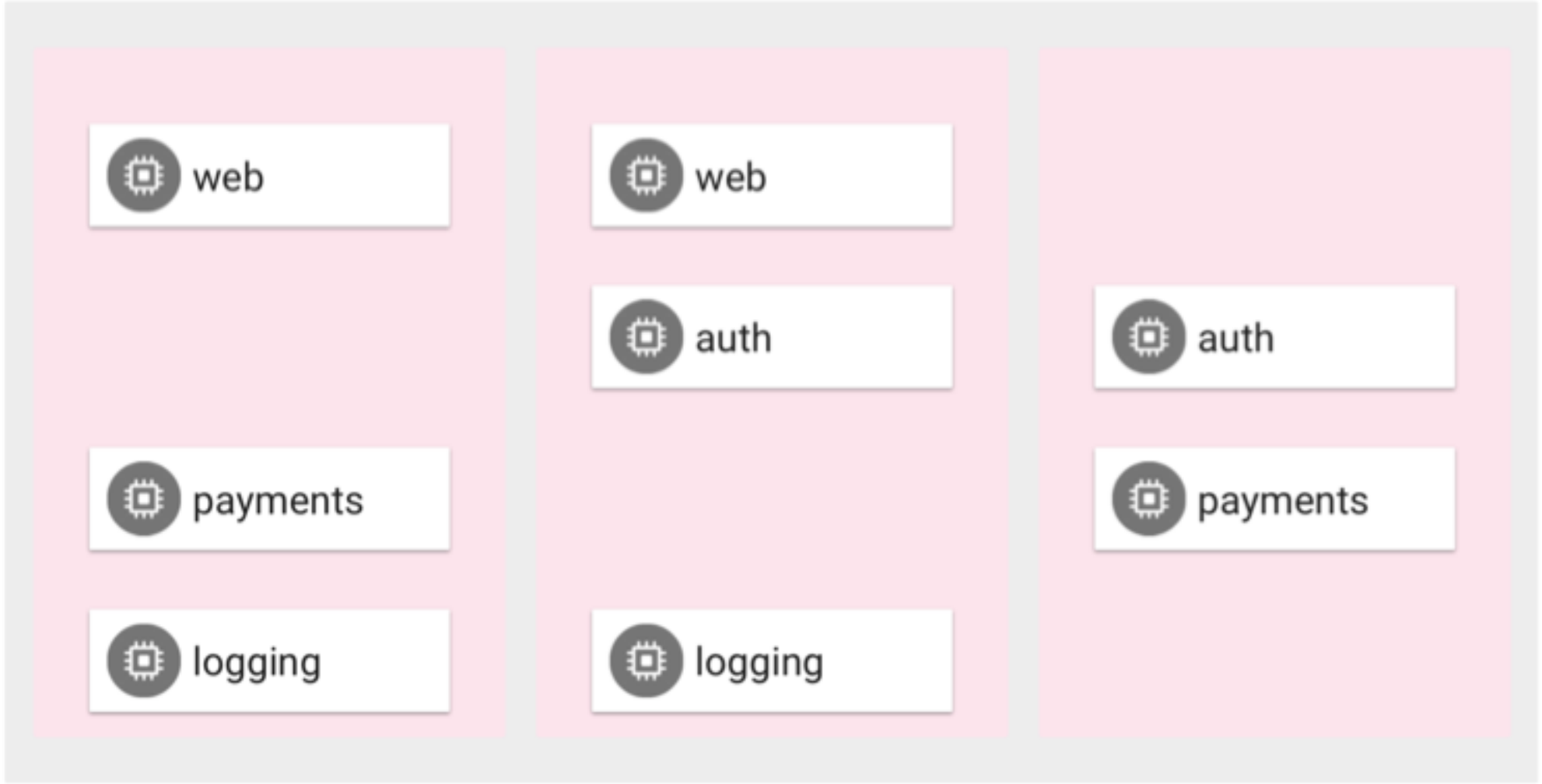


Service Mesh

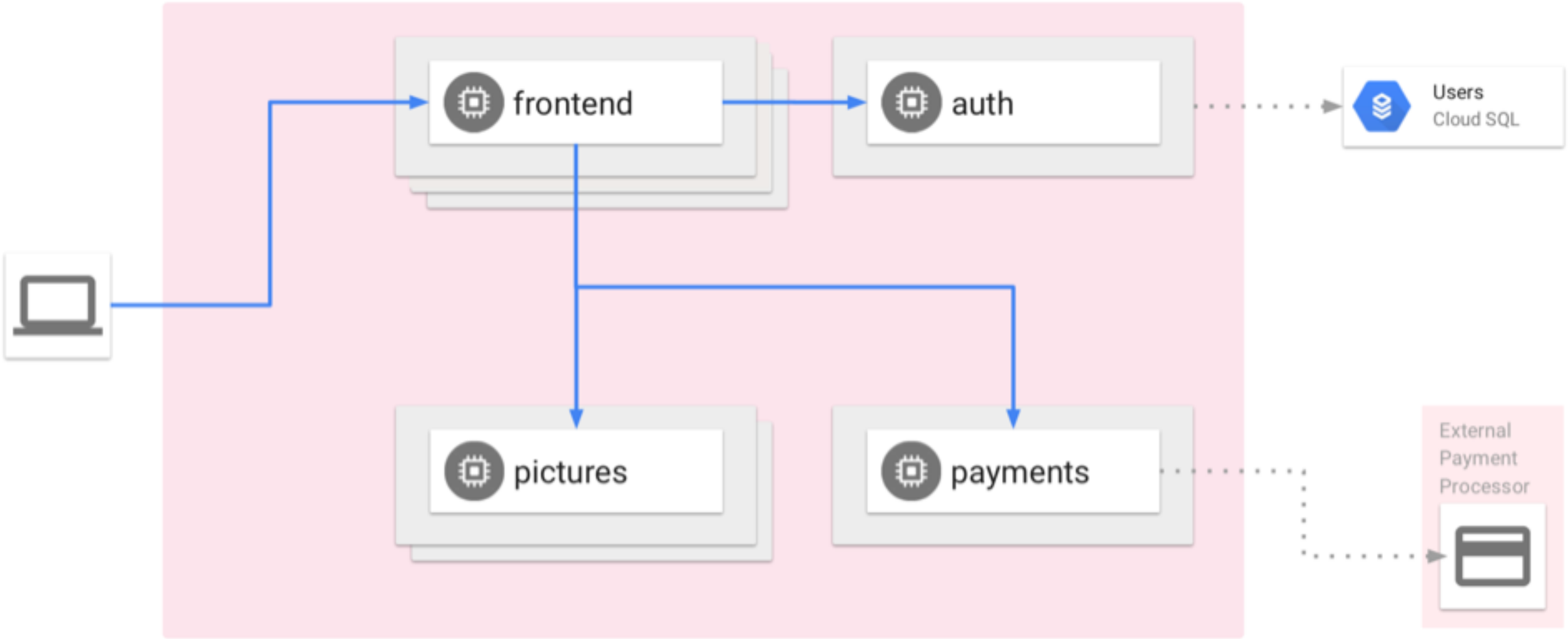


Istio Architecture

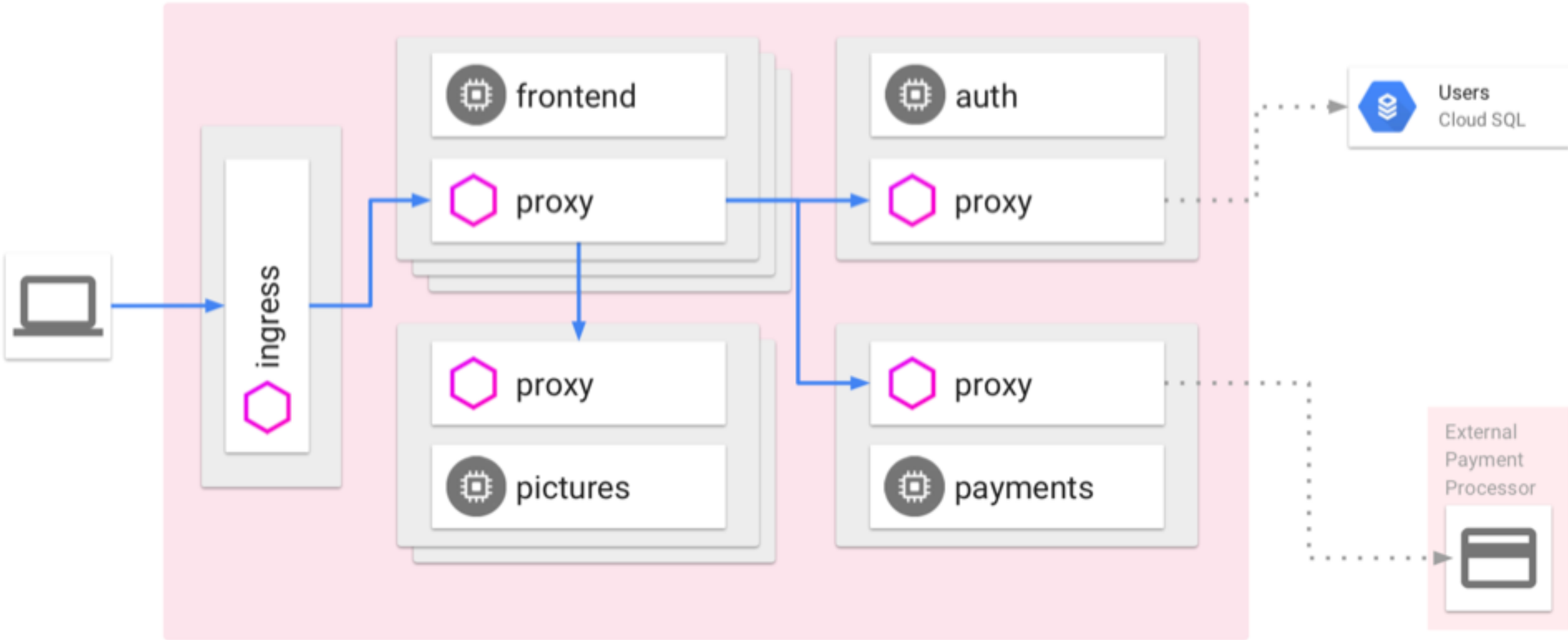
Service Mesh



Service Mesh



Service Mesh

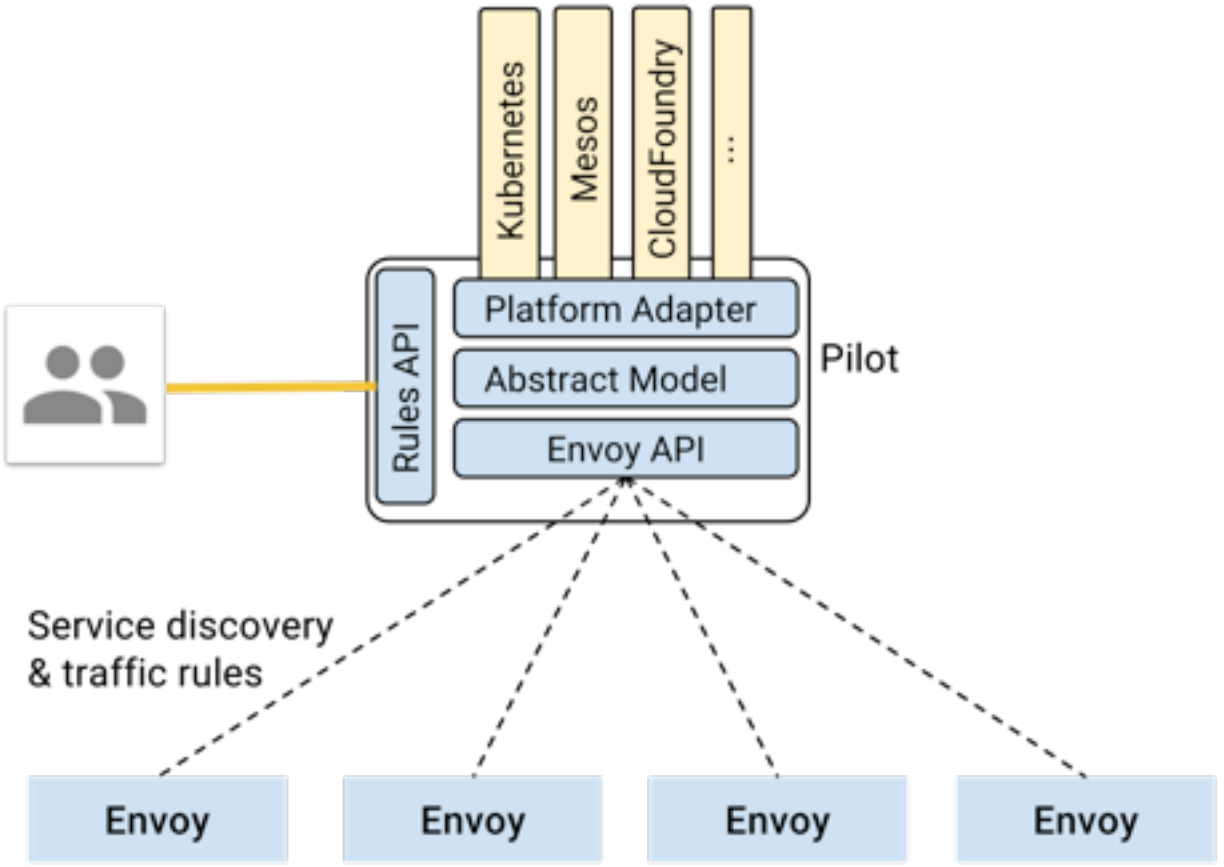


Istio Components

- Pilot
  - Manages Envoy
- Mixer
  - Input data collector
    - telemetry
    - policy
- Auth/Citadel (0.8)
  - Envoy security
- Envoy
  - Proxy

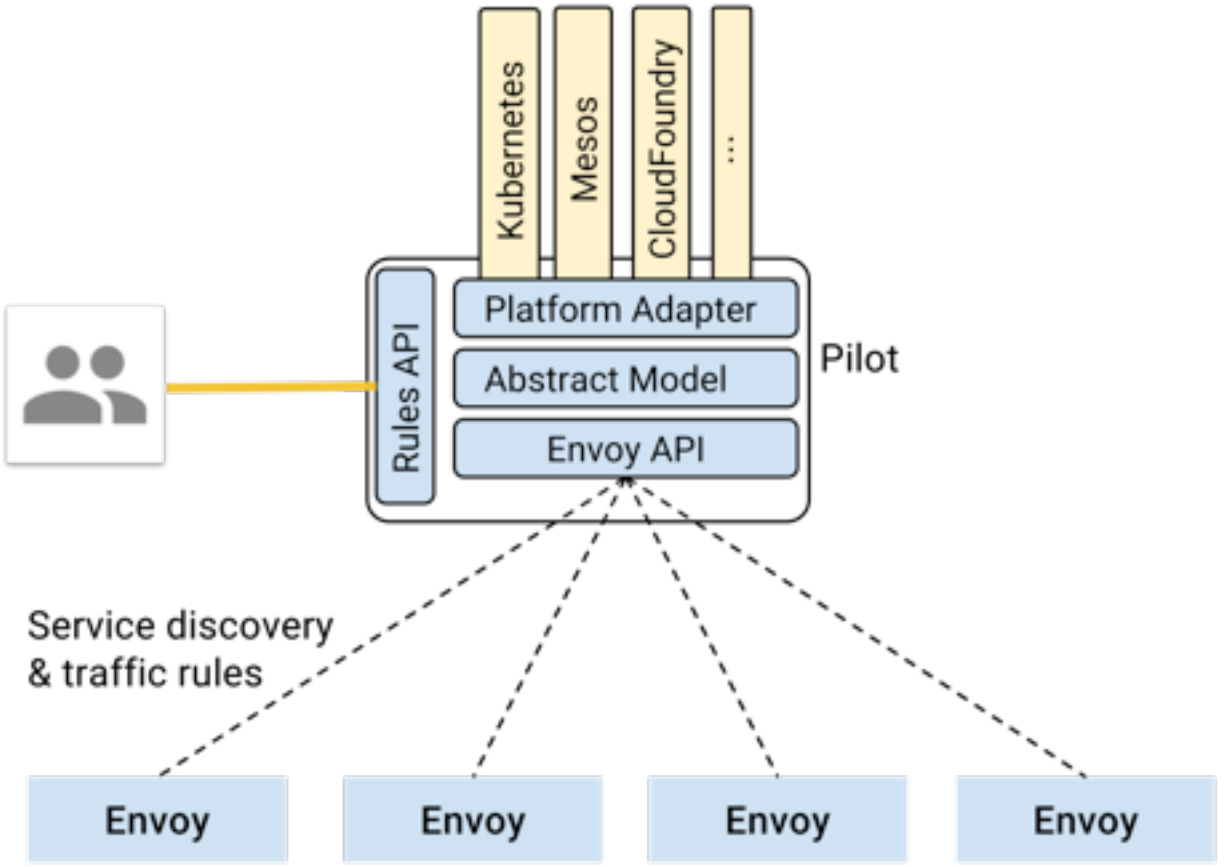
Pilot

- Manages Envoy proxies
- Exposes APIs
  - Service discovery
  - dynamic updates (LB pools, routing tables)
- Independent of the underlying platform
- Each platform has it's own adapters



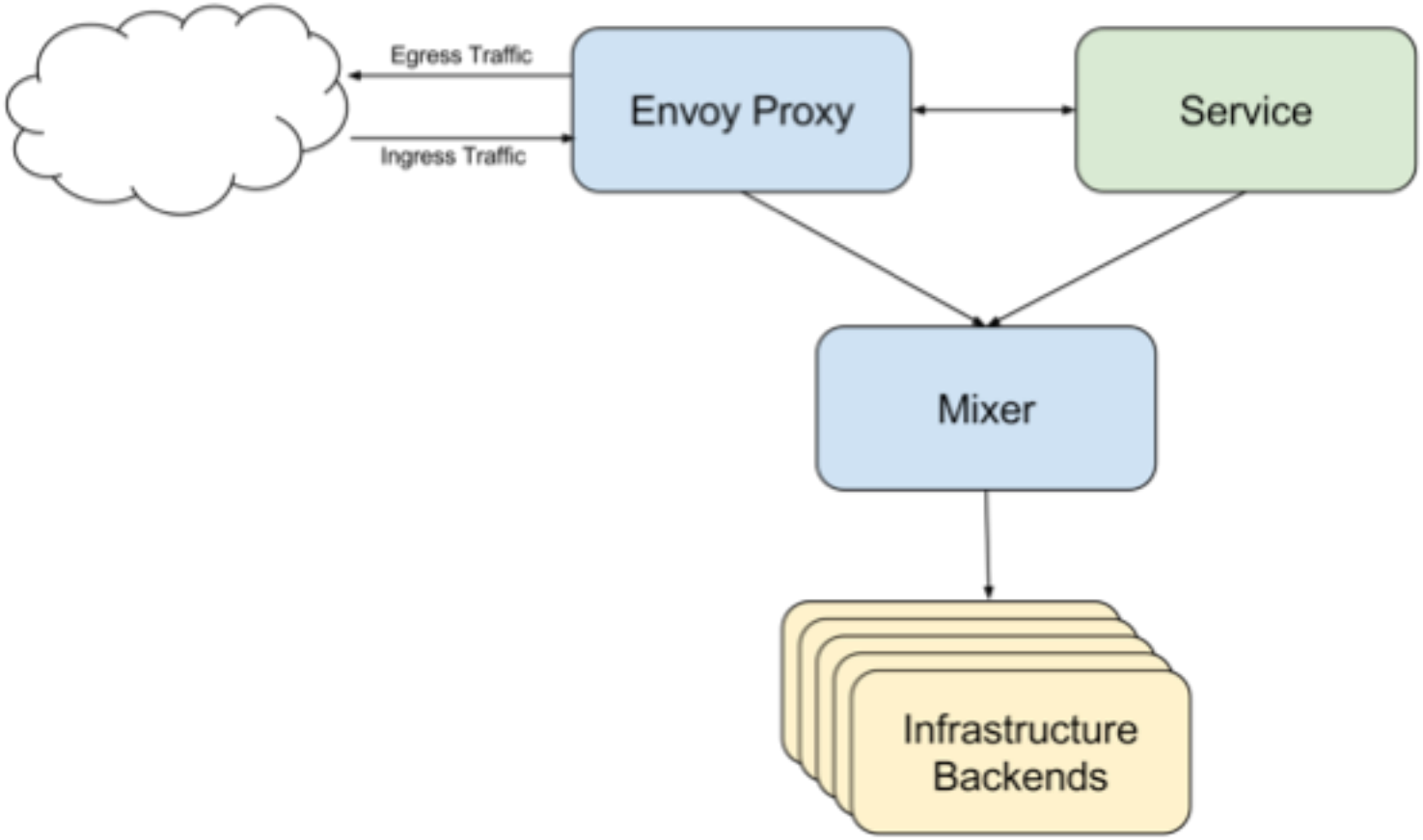
Pilot

- Kubernetes adapter watches Kubernetes API for changes to:
  - Pod
  - Service registration
  - ingress
  - traffic management rules



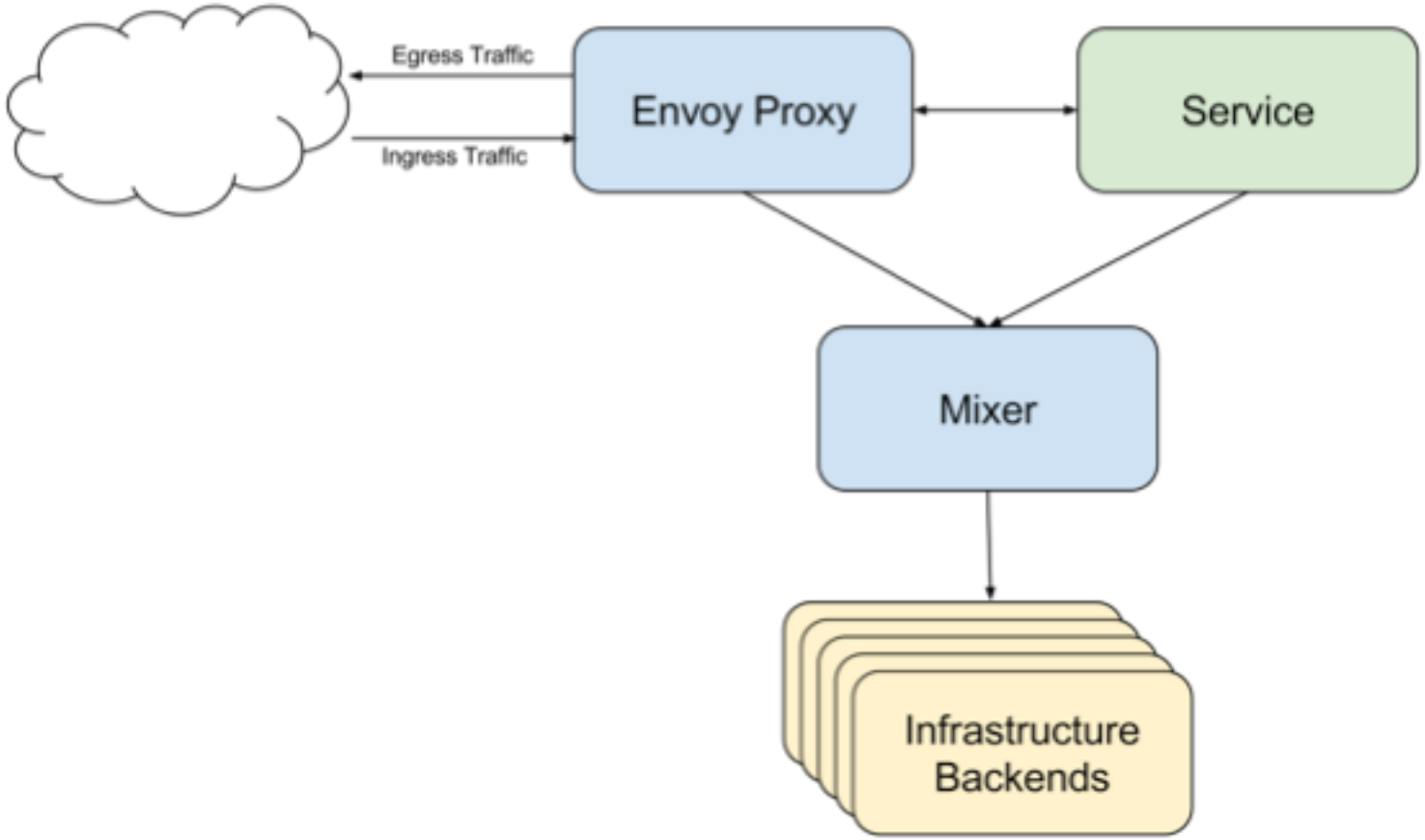
Mixer

- Abstract services from backends
  - access control
  - telemetry
  - quotas
  - etc…



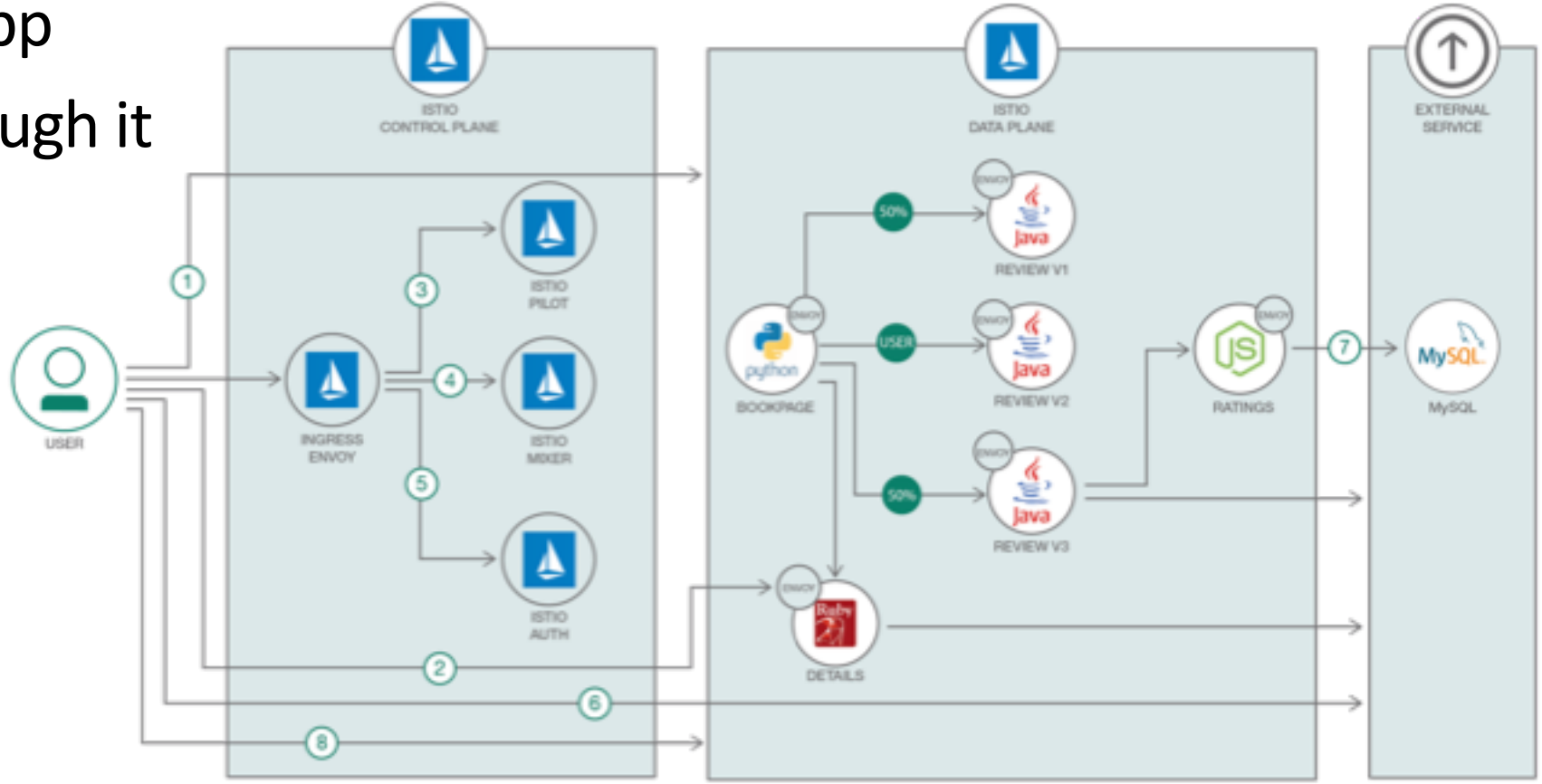
Mixer

- Precondition checking
  - Check to see if authenticated
  - Is caller on whitelist?
  - Authorization for requested service?
- Quota management
  - API rate limits
- Telemetry reporting
  - Services can report logging and monitoring (future: billing and tracing)



Envoy

- Deployed alongside app
- All traffic proxied through it
- Build by Lyft
  - 100+ services
  - 10,000+ VMs
  - 2M requests



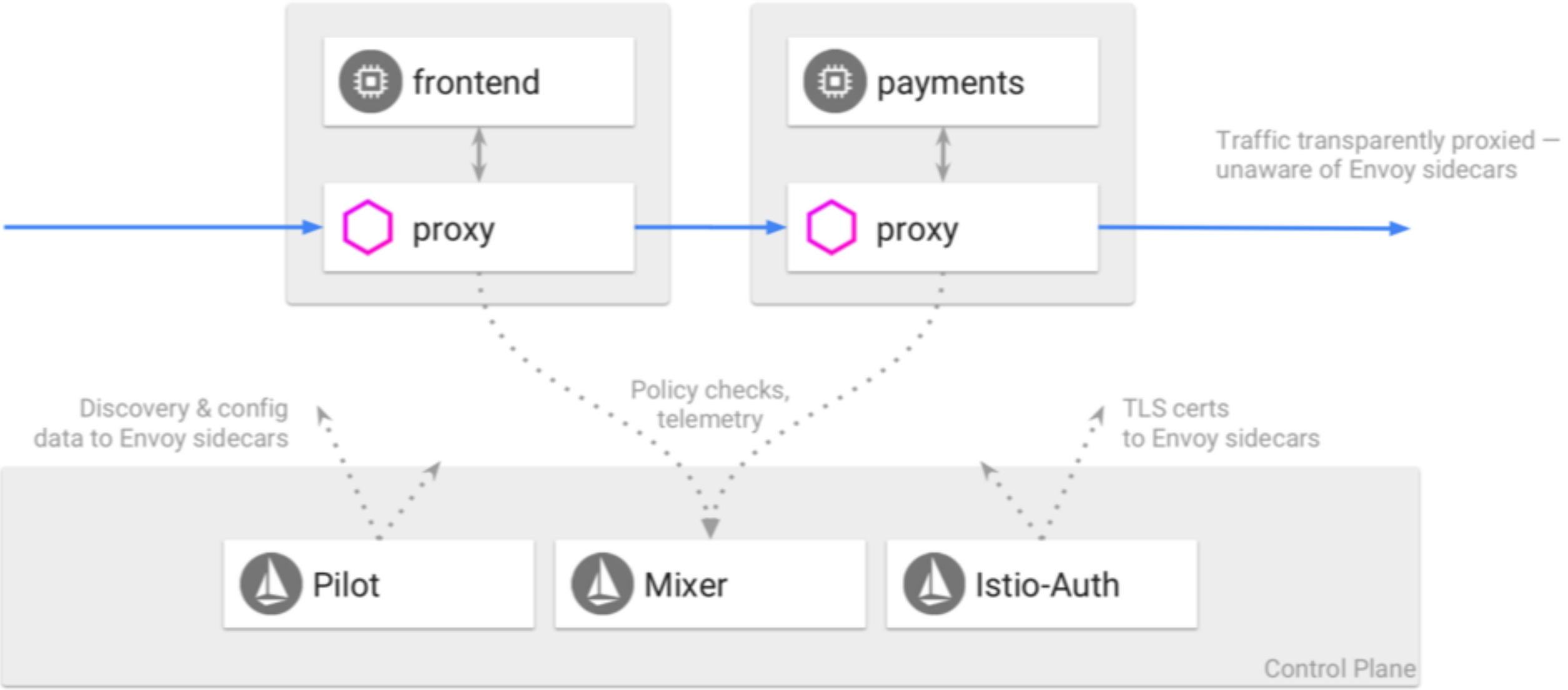
Envoy

```
spec:
  containers:
  - image: frontend:latest
```

```
spec:
  containers:
  - image: frontend:latest
  - image: istio/proxy
```


```
initImage: docker.io/istio/proxy_init
  proxyImage: docker.io/istio/proxy
```

Envoy


frontend
payments
proxy
proxy
Traffic transparently proxied —
unaware of Envoy sidecars
Discovery & config
data to Envoy sidecars
Policy checks,
telemetry
TLS certs
to Envoy sidecars
Pilot
Mixer
Istio-Auth
Control Plane

Lab: Install Istio
Lab: Deploy Bookinfo application

Traffic Management - Rules

Control how API calls & layer-4 traffic flows between services in application deployment

- Set service-level properties
  - timeouts
  - retries
  - circuit breakers
- Continuous deployment
  - canary
  - A/B
  - staged rollouts (% based traffic splits)

Traffic Management - VirtualService

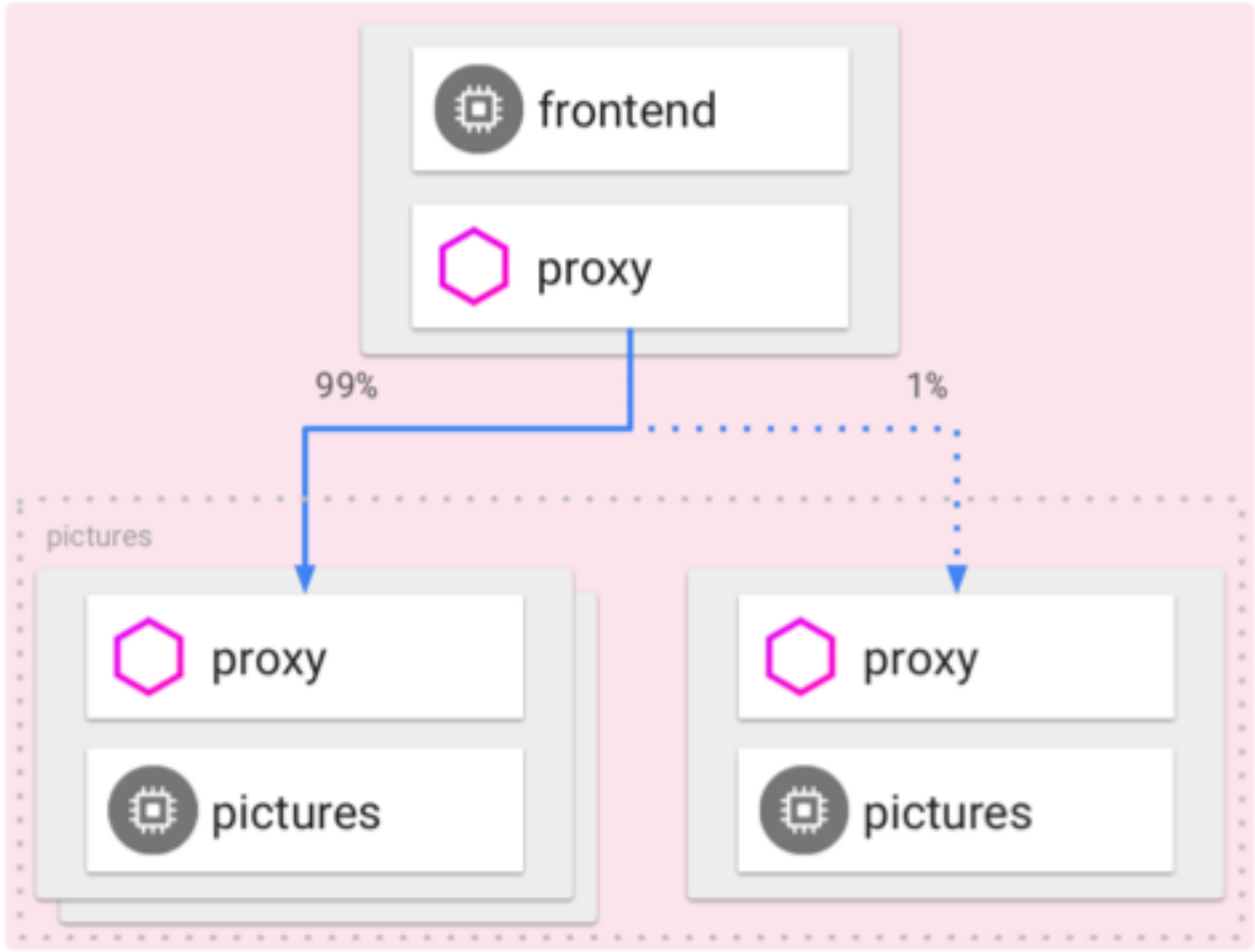
Defines rules that control how requests are routed within an Istio service mesh

- source
- destination
- HTTP paths
- header fields

VirtualService

// Split traffic

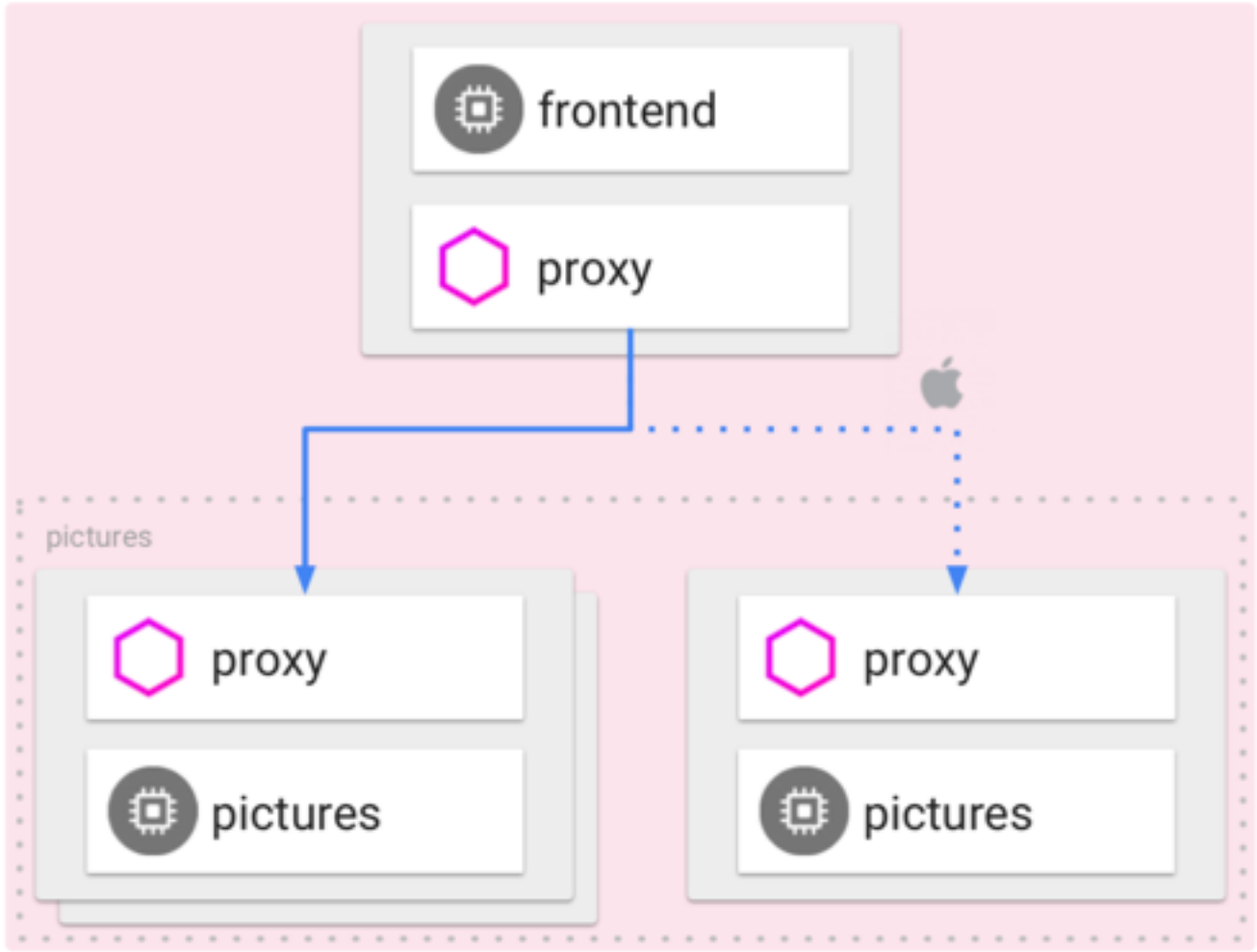
```
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
  - pictures
  http:
  - route:
    - destination:
        host: pictures
        subset: v1
      weight: 99
    - destination:
        host: pictures
        subset: v2
      weight: 1
```



VirtualService

// content based traffic steering

```
kind: VirtualService
metadata:
  name: pictures
spec:
  hosts:
  - pictures
  http:
  - match:
    - headers:
        cookie:
          regex: ^(.*?;)?( iPhone)(;.*)?$"
```



VirtualService- config

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
  - reviews
  http:
  - route:
    - destination:
        host: reviews
        subset: v1
```

VirtualService- config

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
  - reviews
  http:
  - route:
    - destination:
        host: reviews
        subset: v1
```

The route subset specifies the name of a defined subset in a corresponding destination rule configuration:

Traffic Management - DestinationRule

Correspond to one or more request destination hosts that are specified in a VirtualService configuration

- Hosts may or may not match actual destination workload
- Route to Service in mesh, or external
- hosts field must specify FQDN(s)

```
hosts:
  - reviews
  - bookinfo.com
```

Traffic Management - DestinationRule

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
```

Traffic Management - config

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
```

Corresponding DestinationRule with matching subset 'v1'

Qualify Rules - Specific caller

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - match:
      sourceLabels:
        app: reviews
    ...
```

Qualify Rules - Specific caller

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - match:
    - sourceLabels:
        app: reviews
    ...
```

- sourceLabels: matches
  Kubernetes service selector labels

Qualify Rules - Specific caller

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - match:
    - sourceLabels:
        app: reviews
        version: v2
    ...
```

- Can also specify a specific version

Qualify Rules - HTTP headers

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
    - reviews
  http:
  - match:
    - headers:
        cookie:
          regex: "^(.*?;)?(user=jason)(;.*)?$"
    ...
```

Qualify Rules - HTTP headers

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
    - reviews
  http:
  - match:
    - headers:
        cookie:
          regex: "^(.*?;)?(user=jason)(;.*)?$"
    ...
```

- Match user=jason HTTP header

Qualify Rules - HTTP headers

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - match:
    - sourceLabels:
        app: reviews
        version: v2
      headers:
        cookie:
          regex: "^(.*?;)?(user=jason)(;.*)?$"
    ...
```

Qualify Rules - HTTP headers

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - match:
    - sourceLabels:
        app: reviews
        version: v2
      headers:
        cookie:
          regex: "^(.*?;)?(user=jason)(;.*)?$"
    ...
```

- Multiple conditions in same 'match' (AND)

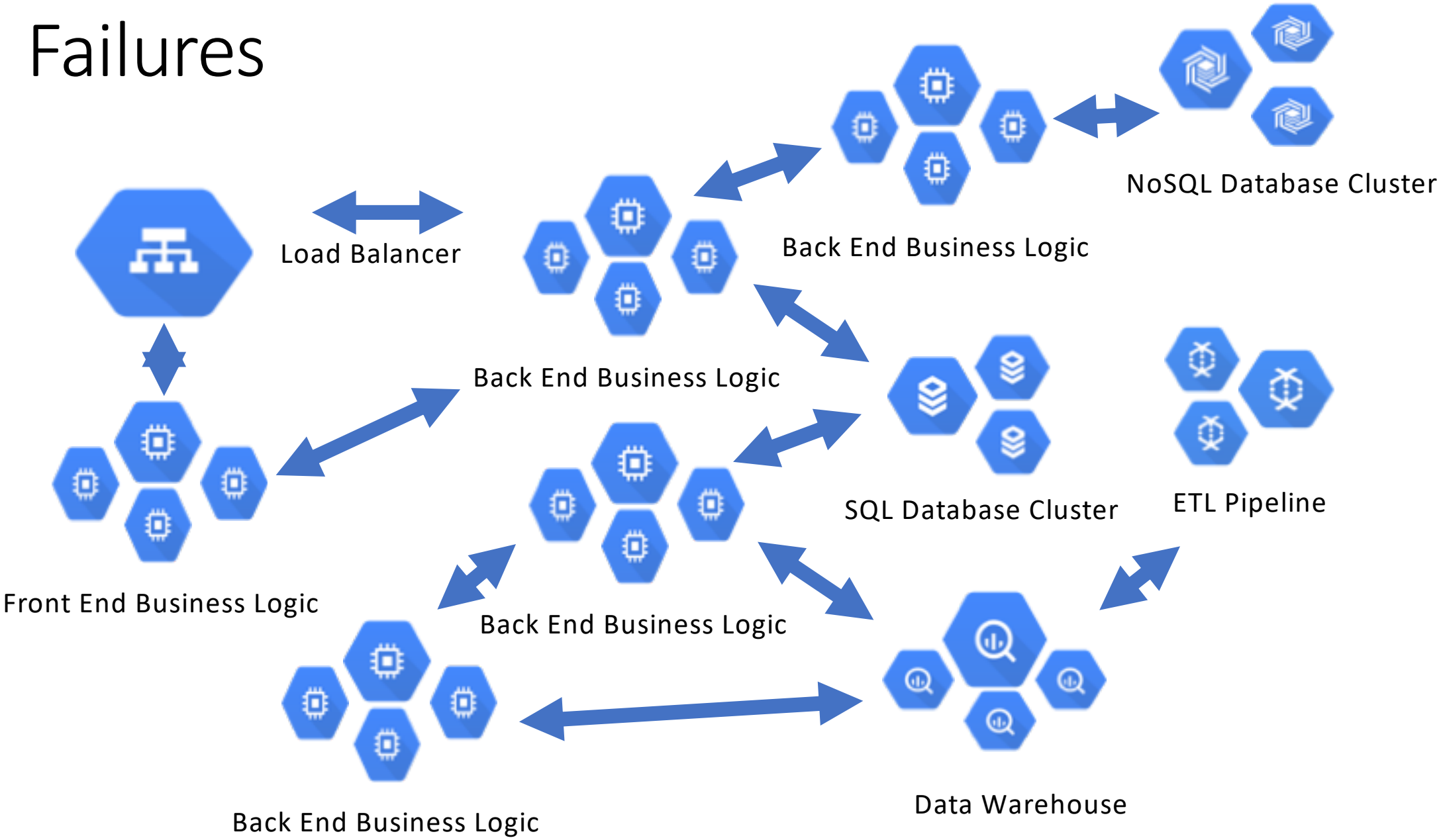
Qualify Rules - HTTP headers

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - match:
    - sourceLabels:
        app: reviews
        version: v2
  - match:
      headers:
        cookie:
          regex: "^(.*?;)?(user=jason)(;.*)?$"
    ...
```

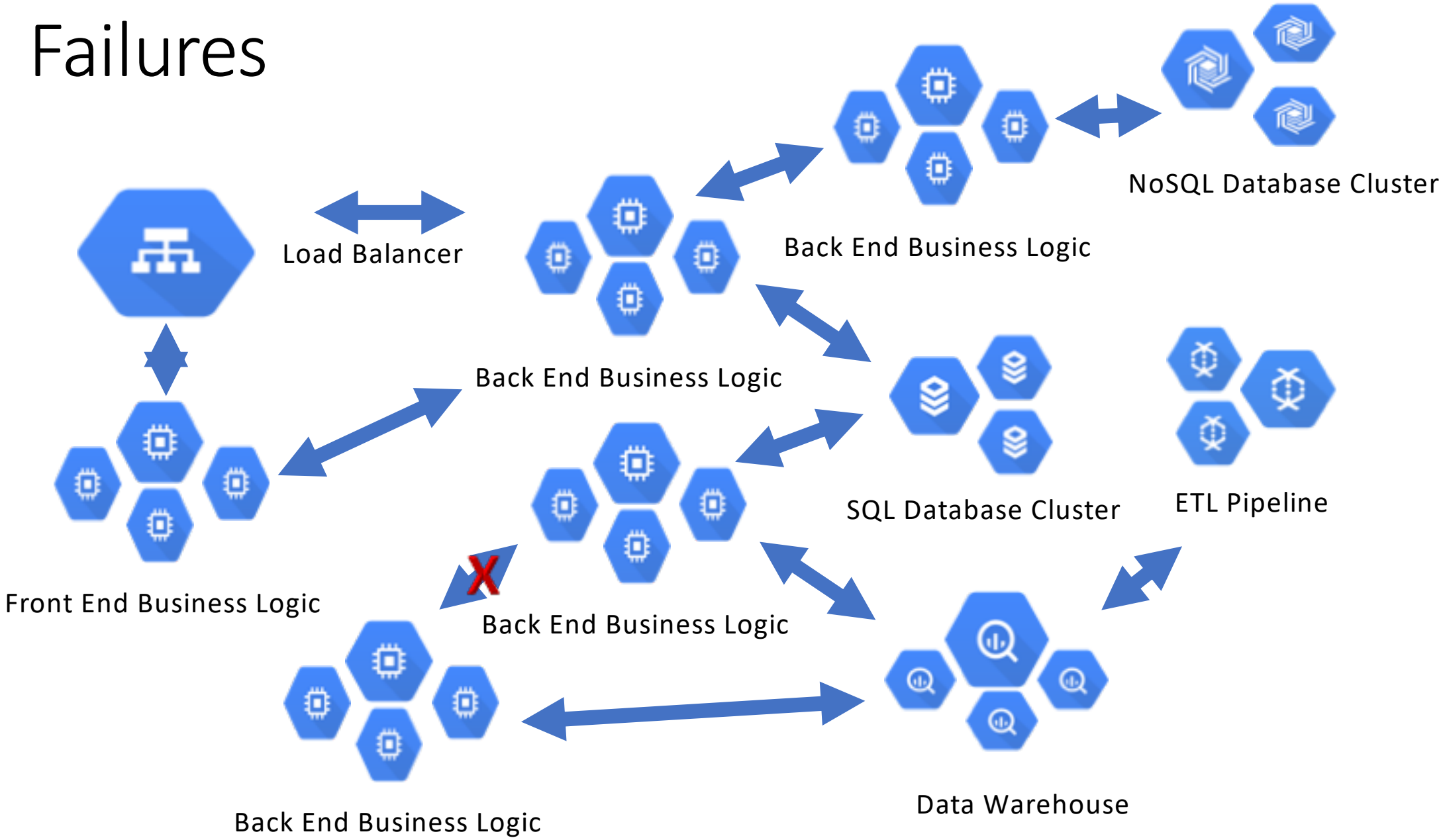
- Multiple 'match' (OR)

Istio: Handling failures

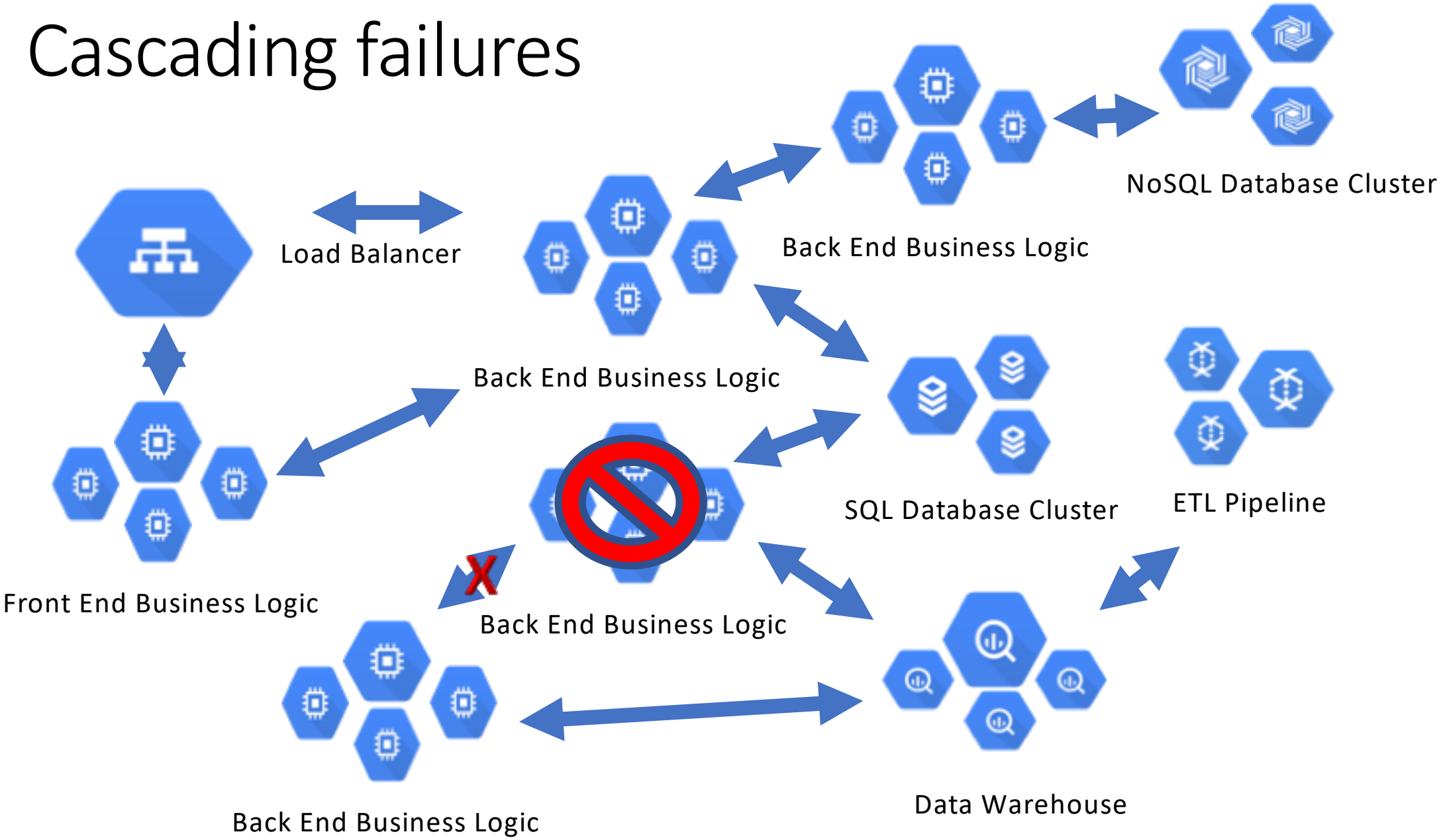
Failures


Load Balancer
Back End Business Logic
NoSQL Database Cluster
Back End Business Logic
SQL Database Cluster
ETL Pipeline
Front End Business Logic
Back End Business Logic
Back End Business Logic
Data Warehouse

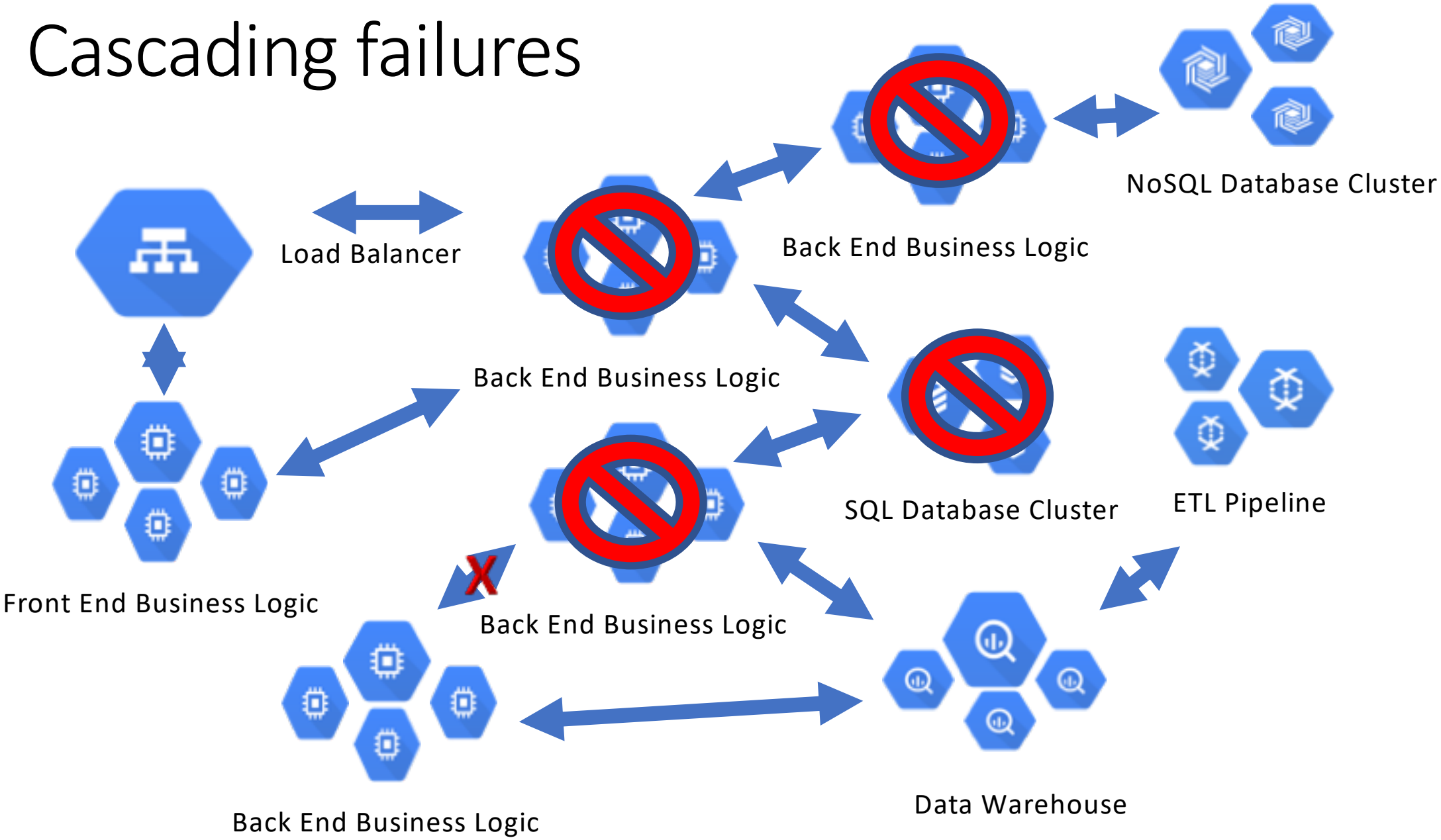
Failures


Load Balancer
Back End Business Logic
NoSQL Database Cluster
Back End Business Logic
SQL Database Cluster
ETL Pipeline
Front End Business Logic
Back End Business Logic
Back End Business Logic
Data Warehouse

Cascading failures



Cascading failures




Database Cluster
Back End Business Lo
Back End
peline
Front End Busine
ack End Business Logic
Back End Business L
Warehouse

Timeouts & Retries

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
    - ratings
  http:
  - route:
    - destination:
        host: ratings
        subset: v1
    timeout: 10s
```

Timeouts & Retries

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
    - ratings
  http:
  - route:
    - destination:
        host: ratings
        subset: v1
    timeout: 10s
```

- Change timeout from default 15 to 10 seconds

Timeouts & Retries

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
    - ratings
  http:
  - route:
    - destination:
        host: ratings
        subset: v1
    retries:
      attempts: 3
      perTryTimeout: 2s
```

Timeouts & Retries

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
    - ratings
  http:
  - route:
    - destination:
        host: ratings
        subset: v1
    retries:
      attempts: 3
      perTryTimeout: 2s
```

- Set retries

Testing Services

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - fault:
      delay:
        percent: 10
        fixedDelay: 5s
    route:
    - destination:
        host: ratings
        subset: v1
```

Testing Services

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - fault:
      delay:
        percent: 10
        fixedDelay: 5s
    route:
    - destination:
        host: ratings
        subset: v1
```

- Inject fault (delay 10% by 5 secs)

Testing Services

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - fault:
      abort:
        percent: 10
        httpStatus: 400
    route:
    - destination:
        host: ratings
        subset: v1
```

Testing Services

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - fault:
      abort:
        percent: 10
        httpStatus: 400
    route:
    - destination:
        host: ratings
        subset: v1
```

- Inject fault (400 error code 10%)

Testing Services

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - match:
    - sourceLabels:
        app: reviews
        version: v2
    fault:
      delay:
        fixedDelay: 5s
      abort:
        percent: 10
        httpStatus: 400
    route:
    - destination:
        host: ratings
        subset: v1
```

Testing Services

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - match:
    - sourceLabels:
        app: reviews
        version: v2
    fault:
      delay:
        fixedDelay: 5s
      abort:
        percent: 10
        httpStatus: 400
    route:
    - destination:
        host: ratings
        subset: v1
```

- Use delay and abort together to confirm services are resilient

Rule Priority

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
  - reviews
  http:
  - match:
    - headers:
        Foo:
          exact: bar
    route:
    - destination:
        host: reviews
        subset: v2
  - route:
    - destination:
        host: reviews
        subset: v1
```

- Read from top to bottom
- Once match found all other rules are ignored
- Best practice: Higher priority rules first

Rule Priority

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
  - reviews
  http:
  - match:
    - headers:
        Foo:
          exact: bar
    route:
    - destination:
        host: reviews
        subset: v2
  - route:
    - destination:
        host: reviews
        subset: v1
```

- Checks for Foo = bar, if match sends to v2
- If no match sends to v1

Traffic Management - DestinationRule

configures the set of policies to be applied to a request after VirtualService routing has occurred.

- authored by service owners
- circuit breaker
- load balancer settings
- TLS settings

DestinationRule

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  trafficPolicy:
    loadBalancer:
      simple: RANDOM
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
    trafficPolicy:
      loadBalancer:
        simple: ROUND_ROBIN
  - name: v3
    labels:
      version: v3
```

DestinationRule

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  trafficPolicy:
    loadBalancer:
      simple: RANDOM
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
  trafficPolicy:
    loadBalancer:
      simple: ROUND_ROBIN
  - name: v3
    labels:
      version: v3
```

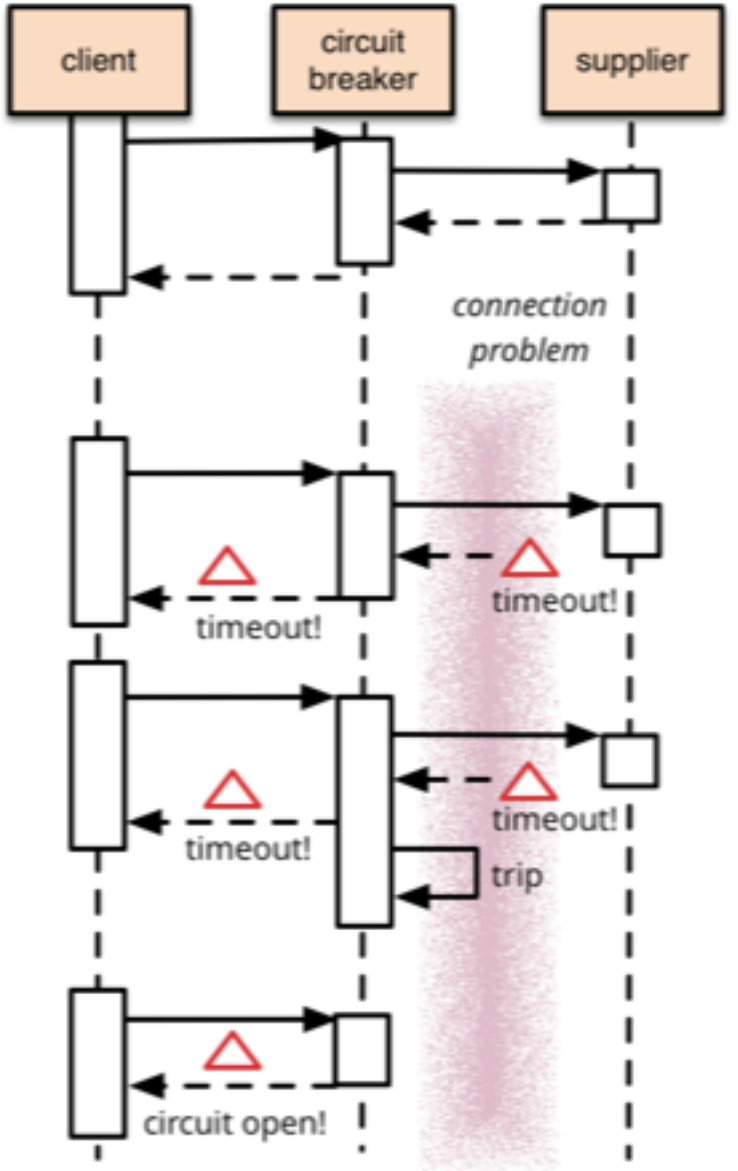
- trafficPolicy (LB RANDOM)
- trafficPolicy (LB Round_ROBIN

Circuit breakers

critical component of distributed systems, which failing quickly, applying back pressure to downstream services

Circuit breakers

- External service calls (HTTP timeout)
- Cascading failures



Circuit breakers

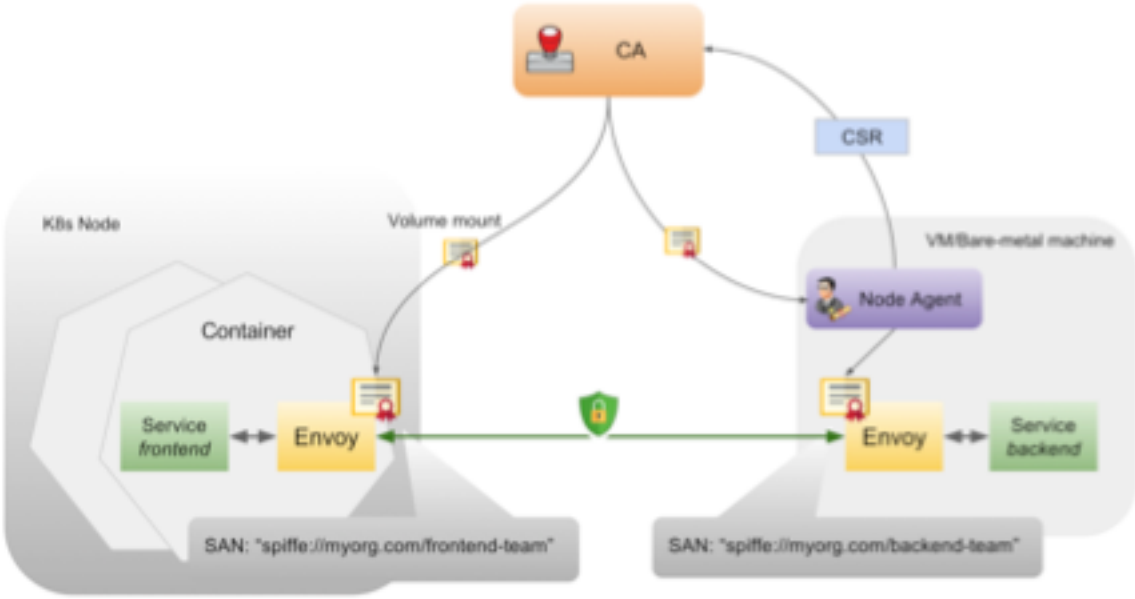
- External service calls (HTTP timeout)
- Cascading failures

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  subsets:
  - name: v1
    labels:
      version: v1
    trafficPolicy:
      connectionPool:
        tcp:
          maxConnections: 100
```

Lab: Dynamic traffic routing
Lab: Handling failures

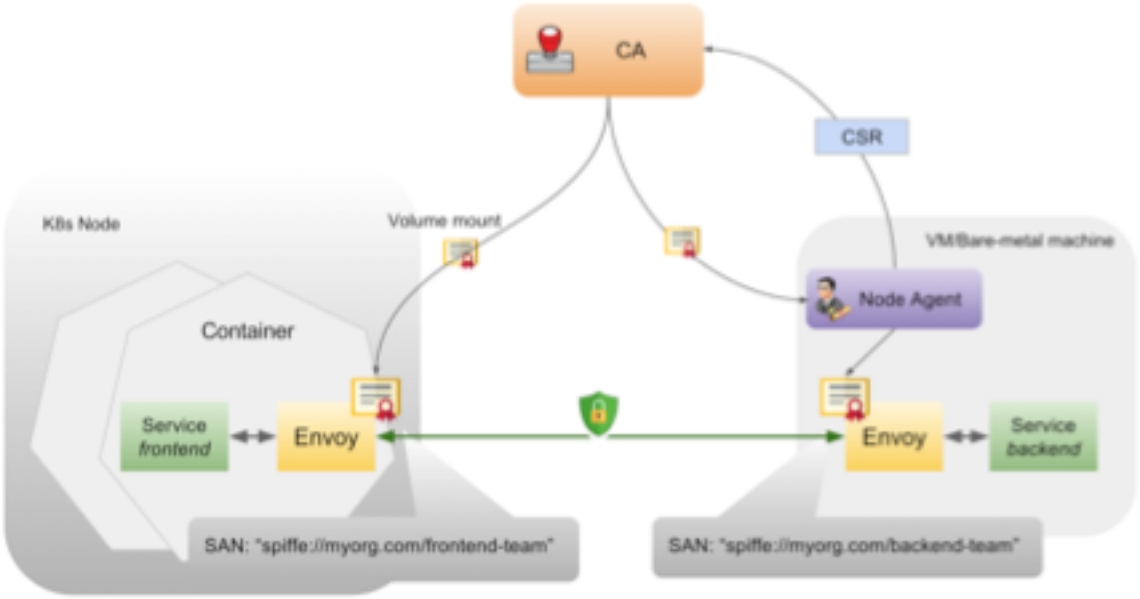
Istio-Auth/Citadel

- Identity
- Key management
- communication security
- Leverages secret volumes to deliver keys/certs from Istio CA to Kubernetes containers.
- Optional mutual TLS encryption



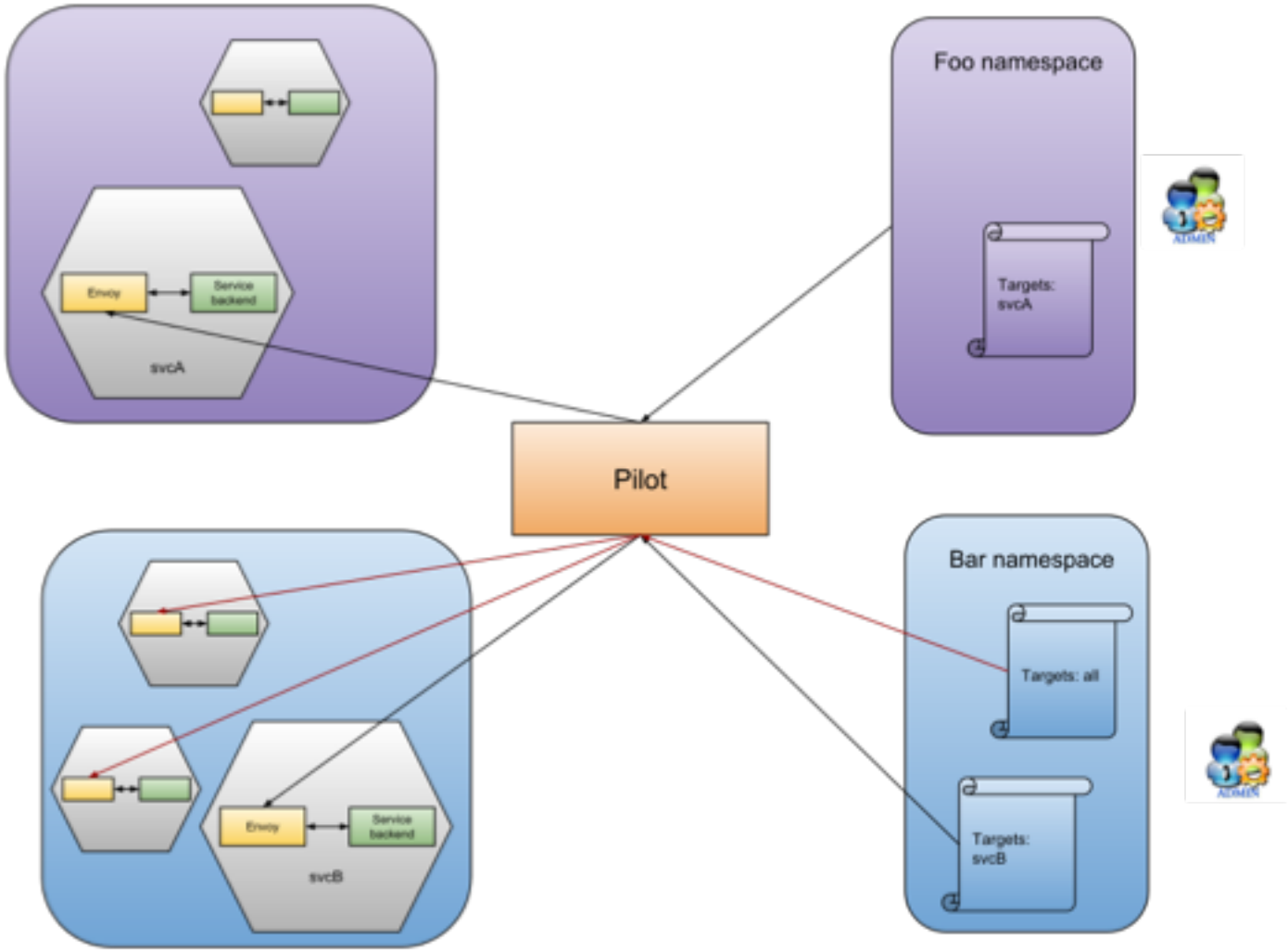
Istio-Auth/Citadel

- Automate key & cert management process:
  - generate key/cert pair for each service account
  - distribute key/cert pair to each pod
  - rotate keys/certs periodically
  - revoke key/cert pair when needed



Authentication Policy

- Istio authentication policy is composed of two parts:
- Peer: verifies the party, the direct client, that makes the connection. The common authentication mechanism for this is mutual TLS.
- Origin: verifies the party, the original client, that makes the request (e.g end-users, devices etc). JWT is the only supported mechanism for origin authentication at the moment.



Anatomy of policy

```
targets:
- name: product-page
- name: reviews
  ports:
  - number: 9000
```

- Target selectors
  - determine which hosts/ports policies applied to.

Enable mTLS

```yaml
apiVersion: "authentication.istio.io/v1alpha1"
kind: "Policy"
metadata:
  name: "example-2"
spec:
  targets:
  - name: httpbin
  peers:
  - mtls:
```

- Enable mTLS

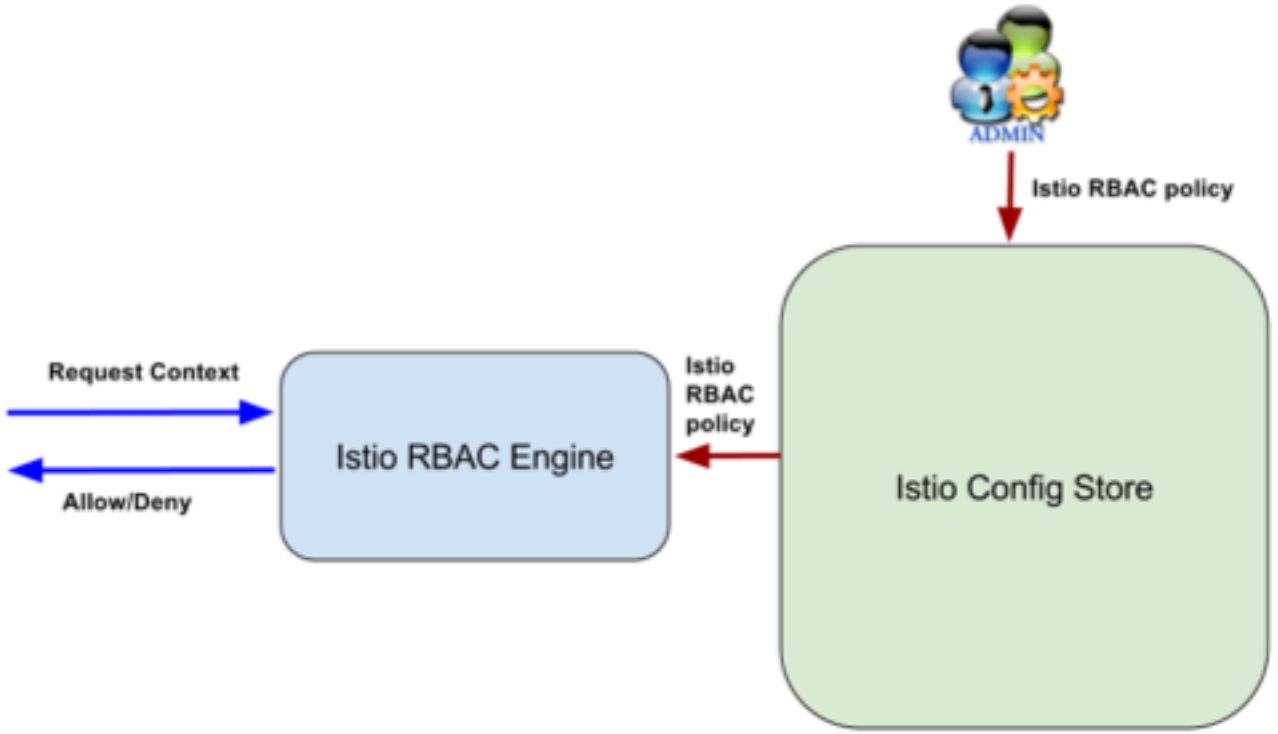
mTLS host

```yaml
apiVersion: "networking.istio.io/v1alpha3"
kind: "DestinationRule"
metadata:
  name: "example-2"
spec:
  host: "httpbin.bar.svc.cluster.local"
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
```

- Enable mTLS on httpbin host

Istio RBAC

- More granular
- Service to service
- end user to service



Istio RBAC

```
apiVersion: "config.istio.io/v1alpha2"
kind: authorization
metadata:
  name: requestcontext
  namespace: istio-system
spec:
  subject:
    user: source.user | ""
    groups: ""
    properties:
      service: source.service | ""
      namespace: source.namespace | ""
  action:
    namespace: destination.namespace | ""
    service: destination.service | ""
    method: request.method | ""
    path: request.path | ""
    properties:
      version: request.headers["version"] | ""
```

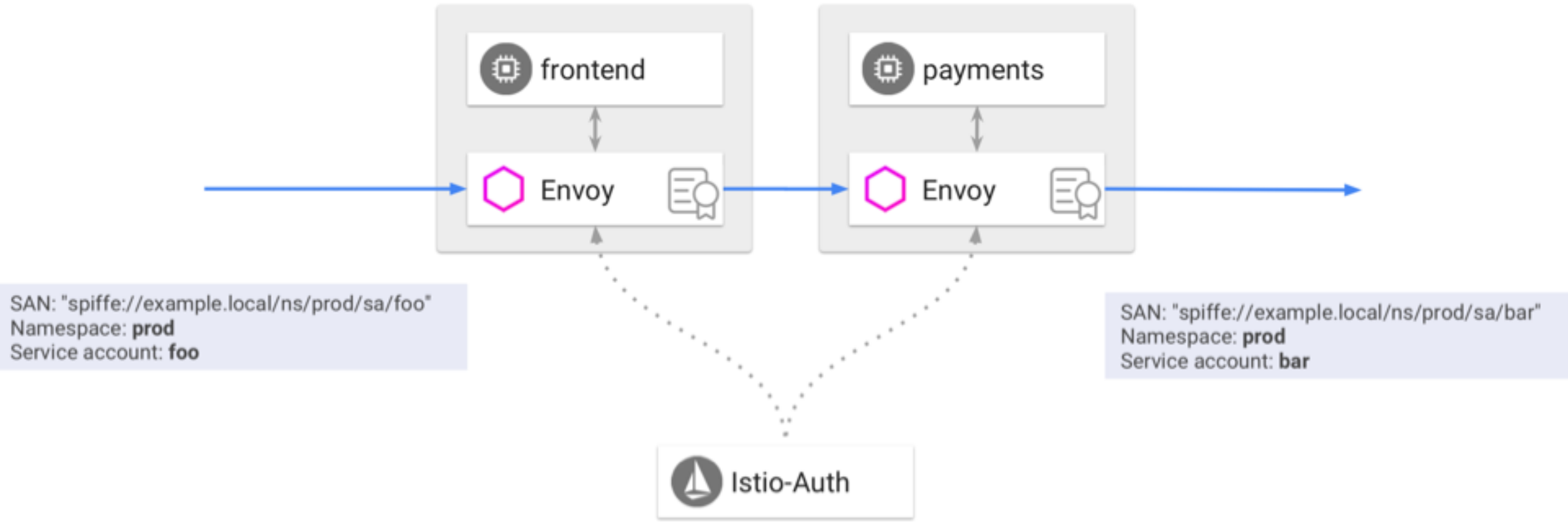
Istio RBAC

```yaml
apiVersion: "config.istio.io/v1alpha2"
kind: ServiceRole
metadata:
  name: products-viewer
  namespace: default
spec:
  rules:
  - services: ["products.default.svc.cluster.local"]
    methods: ["GET", "HEAD"]
```

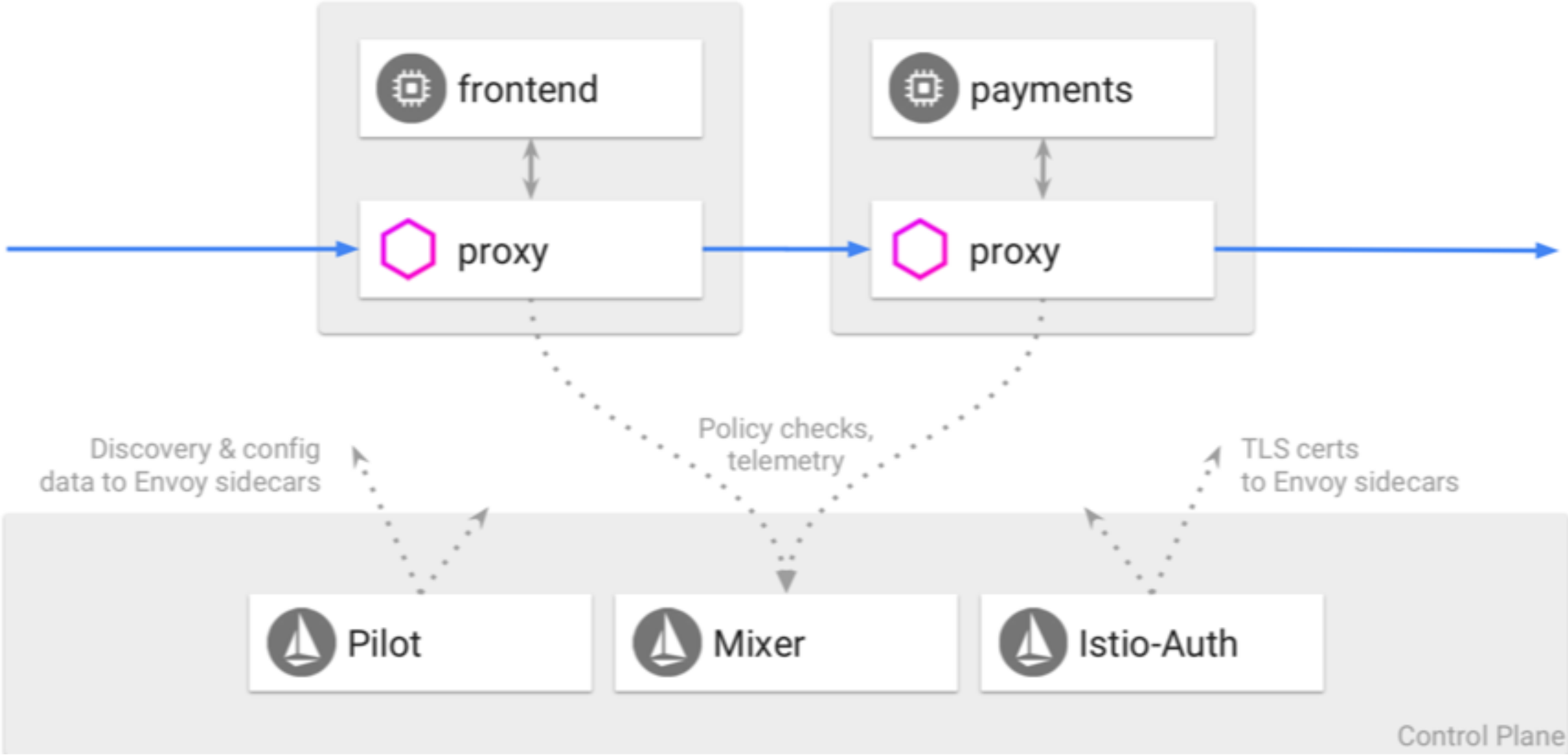
Istio RBAC

```yaml
apiVersion: "config.istio.io/v1alpha2"
kind: ServiceRole
metadata:
  name: products-viewer-version
  namespace: default
spec:
  rules:
  - services: ["products.default.svc.cluster.local"]
    methods: ["GET", "HEAD"]
    constraints:
    - key: "version"
      values: ["v1", "v2"]
```

Security


frontend
payments
Envoy
Envoy
SAN: "spiffe://example.local/ns/prod/sa/foo"
Namespace: prod
Service account: foo
SAN: "spiffe://example.local/ns/prod/sa/bar"
Namespace: prod
Service account: bar
Istio-Auth

Security


frontend
payments
proxy
proxy
Discovery & config
data to Envoy sidecars
Policy checks,
telemetry
TLS certs
to Envoy sidecars
Pilot
Mixer
Istio-Auth
Control Plane

Service Entries

- Istio blocks all outgoing external requests by default
- ServiceEntry required to allow connections to outside services
- mutal TLS authentication disabled
- Policy enforcement performed on client-side

ServiceEntry

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: foo-ext-svc
spec:
  hosts:
  - *.foo.com
  ports:
  - number: 80
    name: http
    protocol: HTTP
  - number: 443
    name: https
    protocol: HTTPS
```

- Allow outgoing on HTTP & HTTPS to *.foo.com

VirtualService & ServiceEntry

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: bar-foo-ext-svc
spec:
  hosts:
    - bar.foo.com
  http:
  - route:
    - destination:
        host: bar.foo.com
    timeout: 10s
```

- Work with ServiceEntry to set a 10s timeout on external service

Gateways

- Describes load balancer at the edge of mesh
  - exposed ports
  - protocol
  - TLS

Gateway

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: my-gateway
spec:
  selector:
    app: my-gatweway-controller
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
    hosts:
    - uk.bookinfo.com
    - eu.bookinfo.com
    tls:
      httpsRedirect: true # sends 302 redirect for http requests
  - port:
      number: 443
      name: https
      protocol: HTTPS
```

Gateway

```
  hosts:
  - uk.bookinfo.com
  - eu.bookinfo.com
  tls:
    mode: SIMPLE #enables HTTPS on this port
    serverCertificate: /etc/certs/servercert.pem
    privateKey: /etc/certs/privatekey.pem
- port:
    number: 9080
    name: http-wildcard
    protocol: HTTP
  hosts:
  - "*"
- port:
    number: 2379 # to expose internal service via external port 2379
    name: mongo
    protocol: MONGO
  hosts:
  - "*"
```

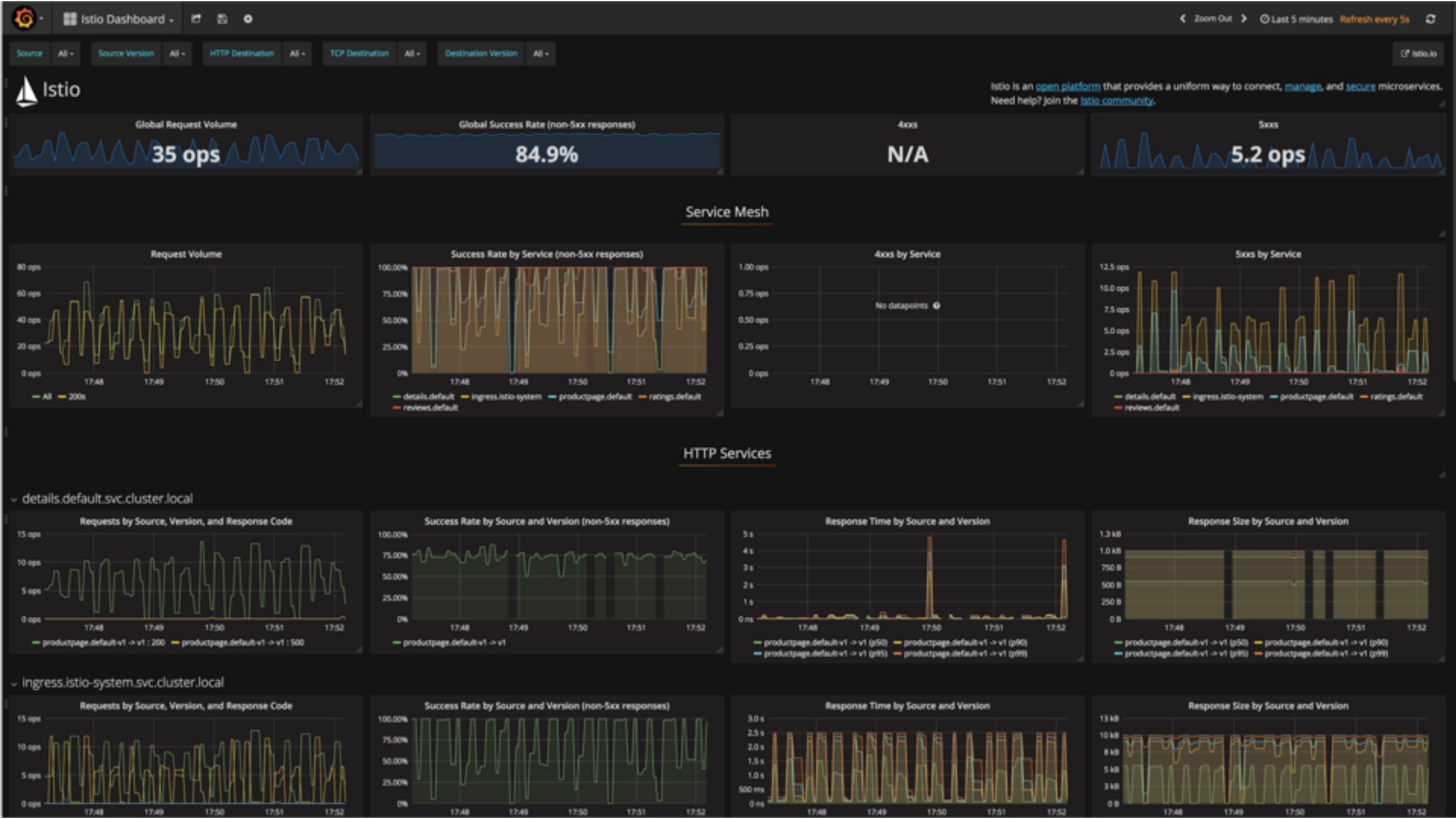
Gateway/VirtualService

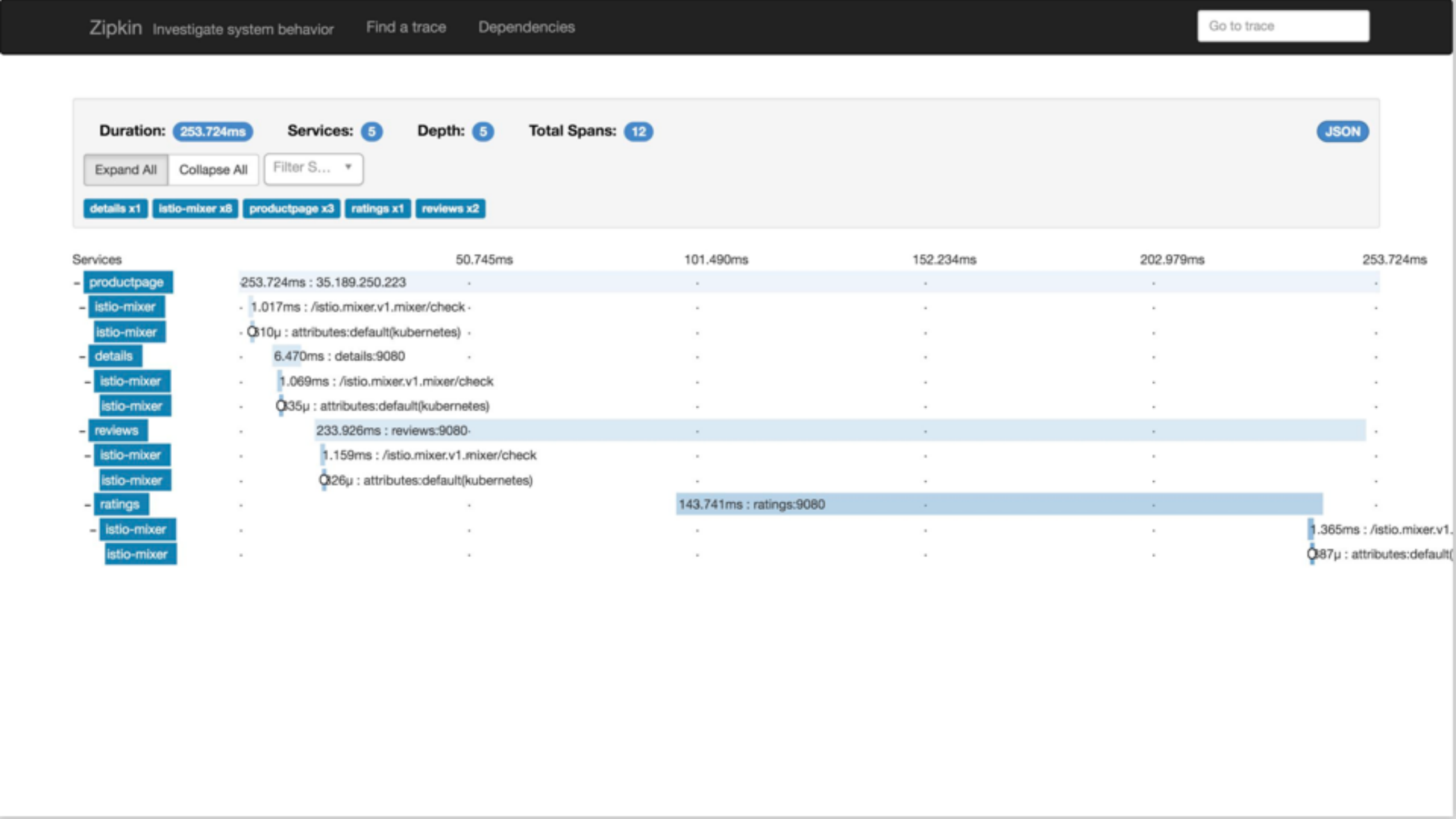
```yaml
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: bookinfo
spec:
  hosts:
    - bookinfo.com
  gateways:
  - bookinfo-gateway # <---- bind to gateway
  http:
  - match:
    - uri:
        prefix: /reviews
    route:
```

Lab: Security

Visibility

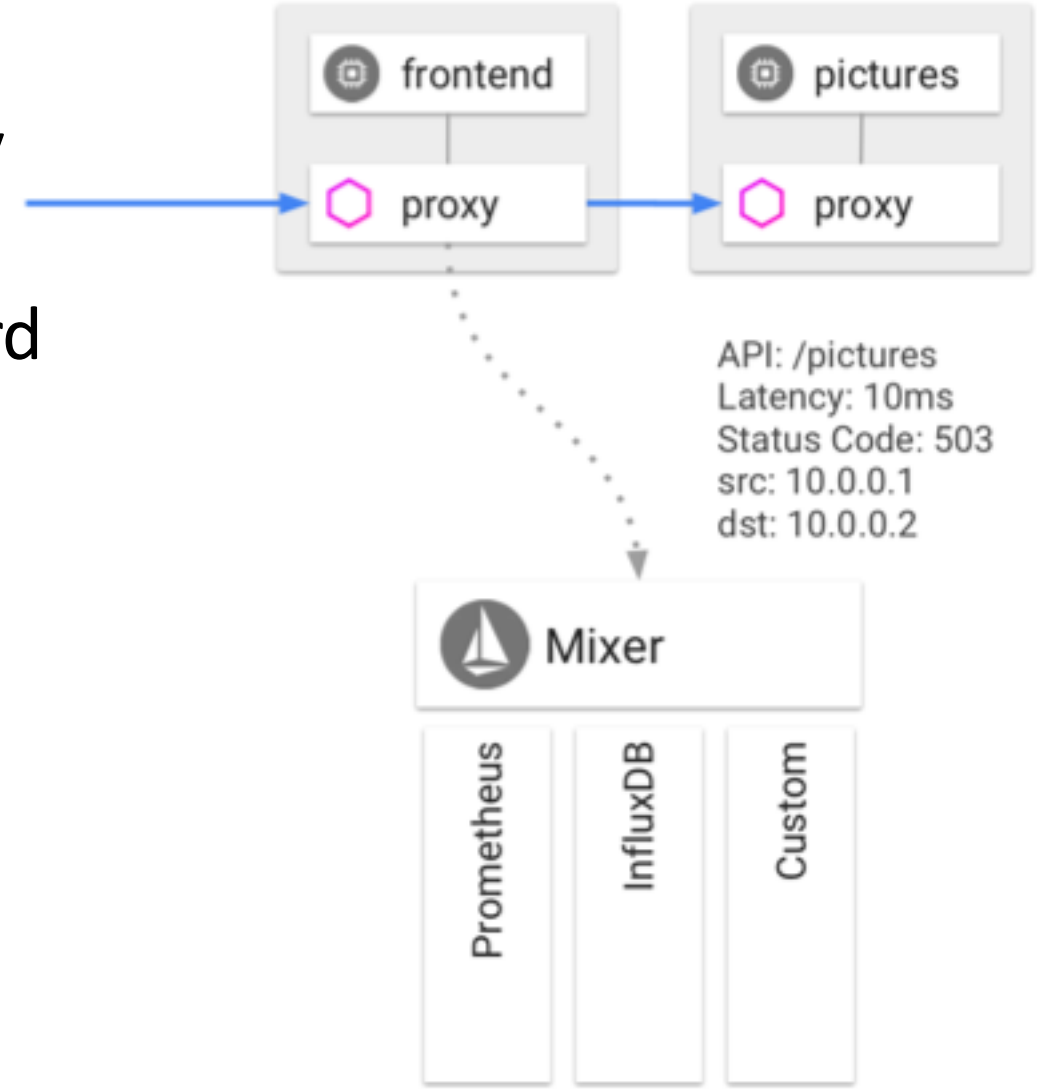
- Metrics without code changes
- Consistent metrics for every app deployed
- Trace flow of requests across services
- Portable across metric backend providers


Istio Dashboard
Zoom Out
Last 5 minutes
Refresh every 5s
Source
All
Source Version
All
HTTP Destination
All
TCP Destination
All
Destination Version
All
istio.io
Istio
Istio is an open platform that provides a uniform way to connect, manage, and secure microservices.
Need help? Join the istio community.
Global Request Volume
35 ops
Global Success Rate (non-5xx responses)
84.9%
4xxs
N/A
5xxs
5.2 ops
Service Mesh
Request Volume
80 ops
60 ops
40 ops
20 ops
0 ops
17:48
17:49
17:50
17:51
17:52
All
200s
Success Rate by Service (non-5xx responses)
100.00%
75.00%
50.00%
25.00%
0%
details.default
ingress.istio-system
productpage.default
ratings.default
reviews.default
4xxs by Service
1.00 ops
0.75 ops
0.50 ops
0.25 ops
0 ops
No datapoints
5xxs by Service
12.5 ops
10.0 ops
7.5 ops
5.0 ops
2.5 ops
0 ops
HTTP Services
details.default.svc.cluster.local
Requests by Source, Version, and Response Code
15 ops
10 ops
5 ops
0 ops
productpage.default-v1 -> v1 : 200
productpage.default-v1 -> v1 : 500
Success Rate by Source and Version (non-5xx responses)
productpage.default-v1 -> v1
Response Time by Source and Version
5 s
4 s
3 s
2 s
1 s
0 ns
productpage.default-v1 -> v1 (p50)
productpage.default-v1 -> v1 (p90)
productpage.default-v1 -> v1 (p95)
productpage.default-v1 -> v1 (p99)
Response Size by Source and Version
1.3 kB
1.0 kB
750 B
500 B
250 B
0 B
ingress.istio-system.svc.cluster.local
Requests by Source, Version, and Response Code
Success Rate by Source and Version (non-5xx responses)
Response Time by Source and Version
3.0 s
2.5 s
2.0 s
1.5 s
1.0 s
500 ms
0 ns
Response Size by Source and Version
13 kB
10 kB
8 kB
5 kB
3 kB
0 B


Zipkin
Investigate system behavior
Find a trace
Dependencies
Go to trace
Duration: 253.724ms
Services: 5
Depth: 5
Total Spans: 12
JSON
Expand All
Collapse All
Filter S...
details x1
istio-mixer x8
productpage x3
ratings x1
reviews x2
Services
50.745ms
101.490ms
152.234ms
202.979ms
253.724ms
productpage
253.724ms : 35.189.250.223
istio-mixer
1.017ms : /istio.mixer.v1.mixer/check
istio-mixer
810µ : attributes:default(kubernetes)
details
6.470ms : details:9080
istio-mixer
1.069ms : /istio.mixer.v1.mixer/check
istio-mixer
835µ : attributes:default(kubernetes)
reviews
233.926ms : reviews:9080
istio-mixer
1.159ms : /istio.mixer.v1.mixer/check
istio-mixer
826µ : attributes:default(kubernetes)
ratings
143.741ms : ratings:9080
istio-mixer
1.365ms : /istio.mixer.v1.
istio-mixer
887µ : attributes:default(

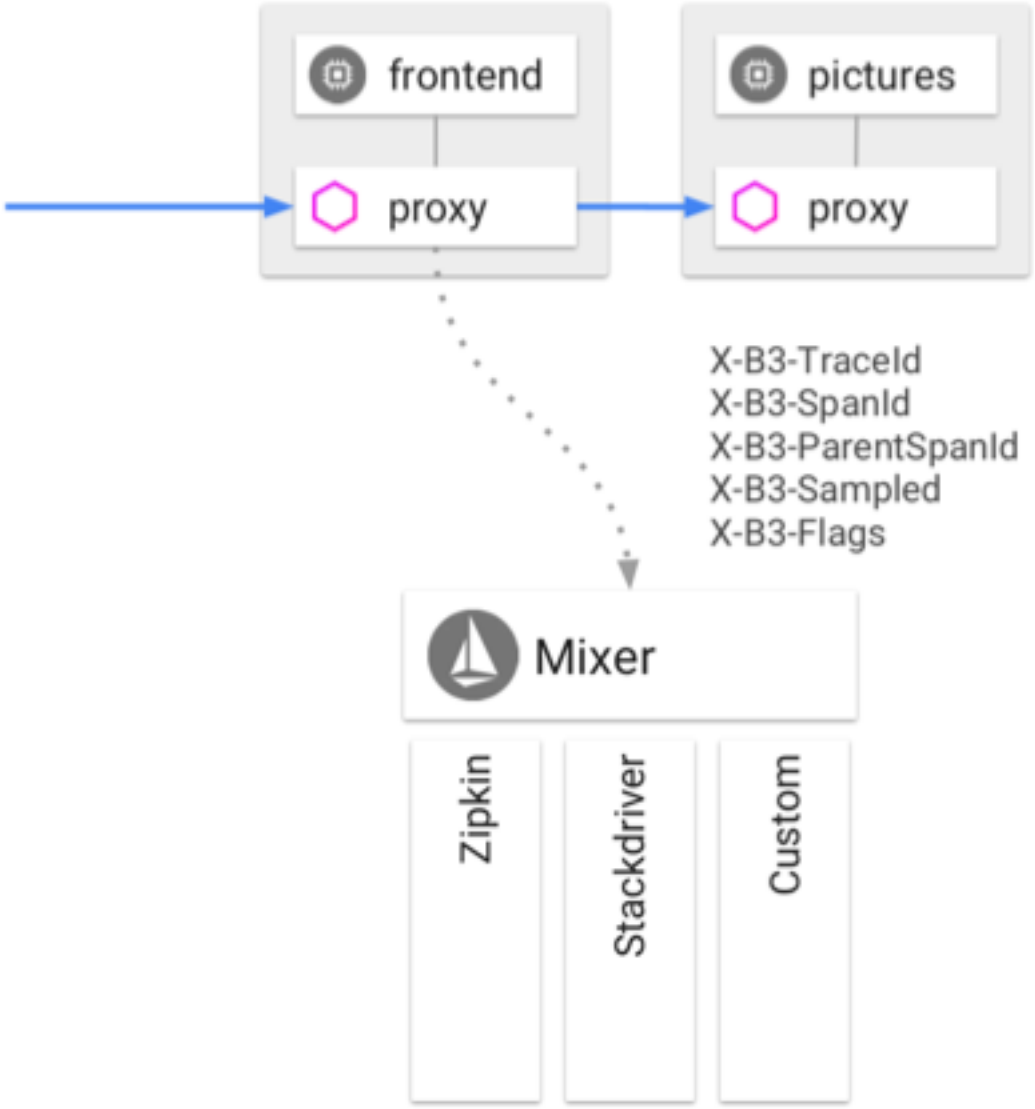
Metrics flow

- Mixer collects metrics emitted by Envoy proxies
- Adapters in Mixer normalize and forward to monitoring backends
- Metrics backend can be swapped at runtime



Tracing

- Apps do not have to deal with generating spans or correlating causality
- Envoy proxies generate spans
  - Apps do need to send some context heads on outbound calls
- Envoys send traces to Mixer
  - Adapters send to backends



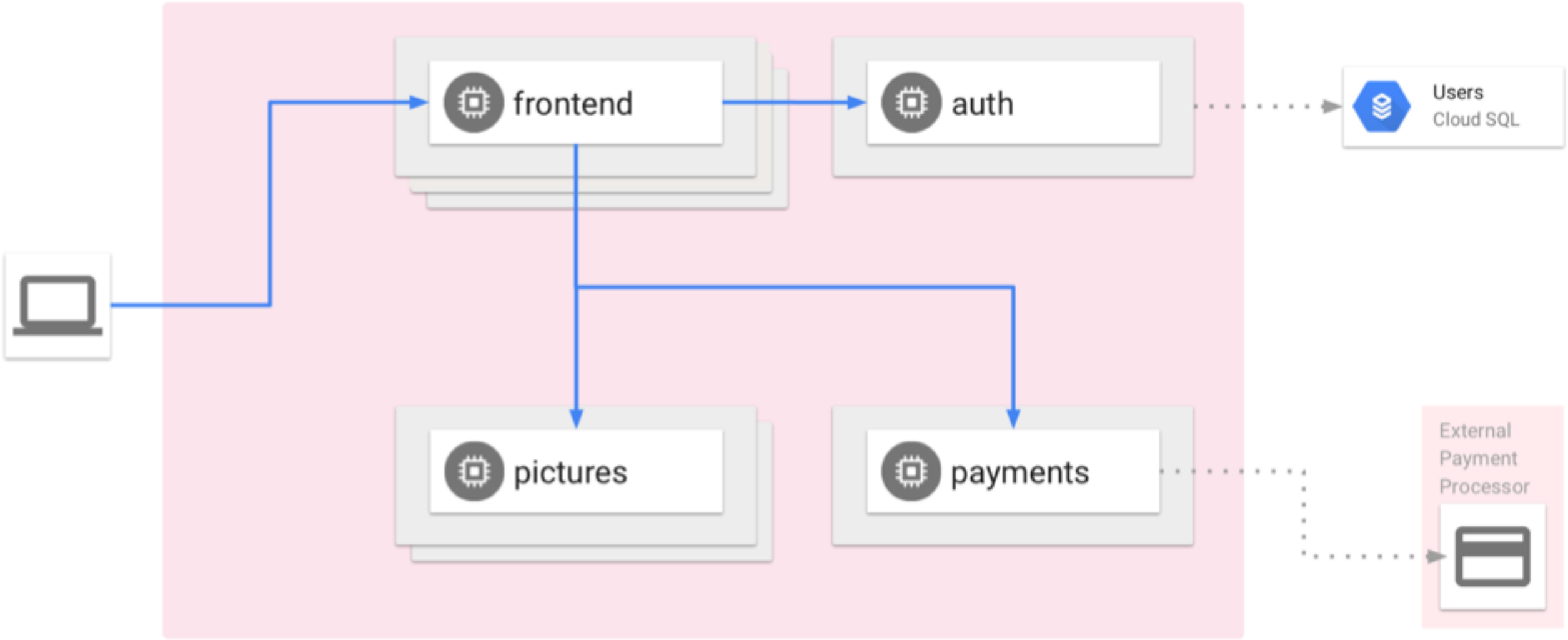
Tracing - headers

```
def getForwardHeaders(request):
    headers = {}

    user_cookie = request.cookies.get("user")
    if user_cookie:
        headers['Cookie'] = 'user=' + user_cookie

    incoming_headers = [ 'x-request-id',
                         'x-b3-traceid',
                         'x-b3-spanid',
                         'x-b3-parentspanid',
                         'x-b3-sampled',
                         'x-b3-flags',
                         'x-ot-span-context'
    ]

    for ihdr in incoming_headers:
        val = request.headers.get(ihdr)
        if val is not None:
            headers[ihdr] = val
```

Tracing - headers

```
@GET
@Path("/reviews")
public Response bookReviews(@CookieParam("user") Cookie user,
                            @HeaderParam("x-request-id") String xreq,
                            @HeaderParam("x-b3-traceid") String xtraceid,
                            @HeaderParam("x-b3-spanid") String xspanid,
                            @HeaderParam("x-b3-parentspanid") String xparentspanid,
                            @HeaderParam("x-b3-sampled") String xsampled,
                            @HeaderParam("x-b3-flags") String xflags,
                            @HeaderParam("x-ot-span-context") String xotspan) {
  String r1 = "";
  String r2 = "";

  if(ratings_enabled){
    JsonObject ratings = getRatings(user, xreq, xtraceid, xspanid, xparentspanid, xsampled, xflags, xotspan
```

Ultimately, it's just this



Lab: Monitoring


THANK
YOU