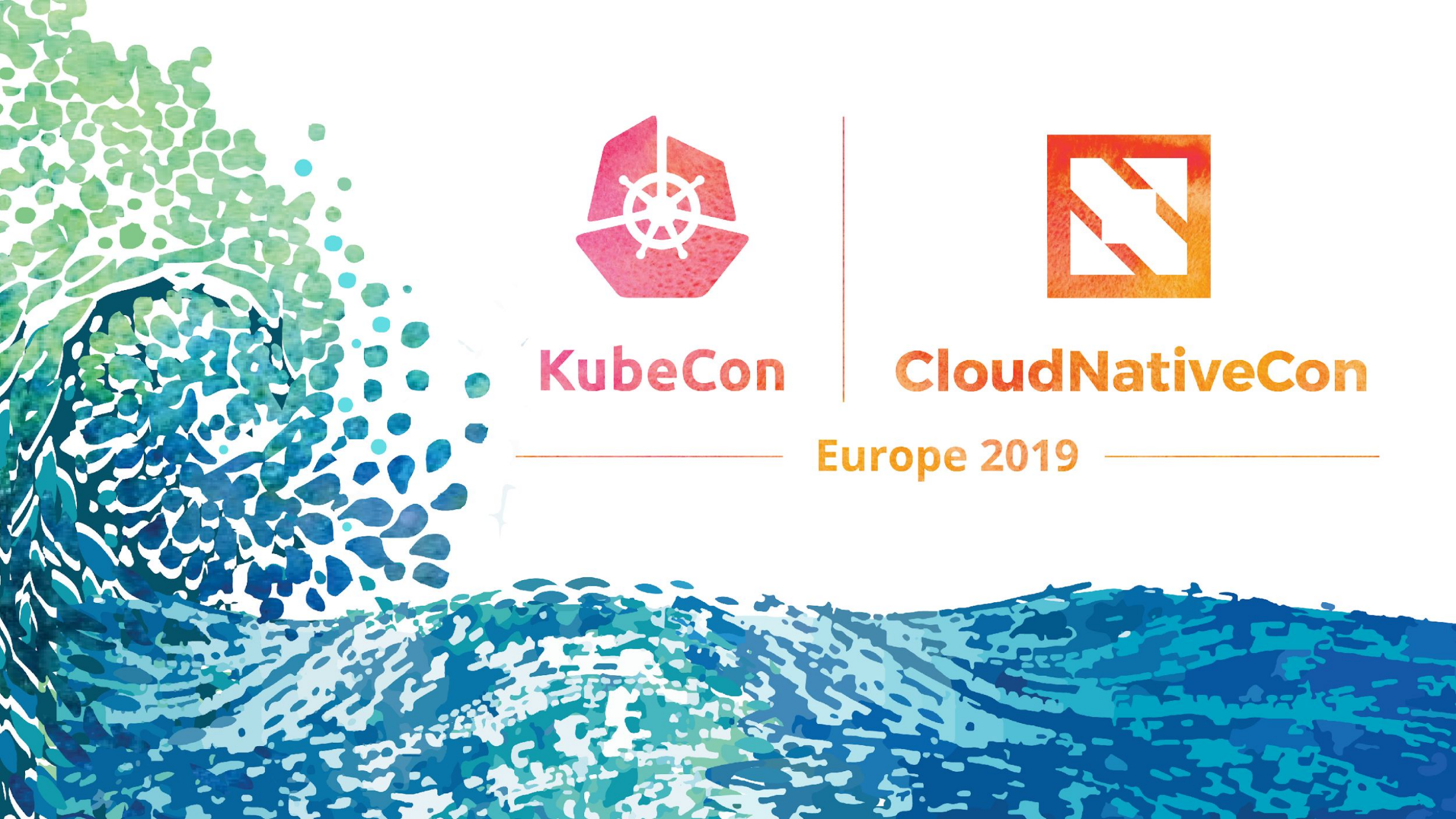

KubeCon
CloudNativeCon
Europe 2019

Caller ID in Kubernetes

Mike Danese, Google

Intros

Mike Danese

- Software Engineer at Google working on GKE Security
- Chair and Tech Lead of Kubernetes SIG-Auth
- Tech Lead on GKE Identity
- Seattleite

Objective

Provide value to customers!

The problem

No Bugs
+ No User Data
+ No Bad Actors
= No Problem!

The elephant

User Data

The problem

How do we create an environment that maintains a sufficiently high level of assurance on *user data*?

Refresh

Coarse Grained Authorization

Broad privilege independent of the object targeted by the request

Fine Grained Authorization

Narrow privileged specific of to the object targeted by the request.

Refresh

Channel-Bound Credential

Bound to the channel on which it arrived (e.g. TLS).

Forwardable Credential

Not bound to the channel on which it arrived (e.g. bearer tokens). Can be further forwarded through a service stack.

Refresh

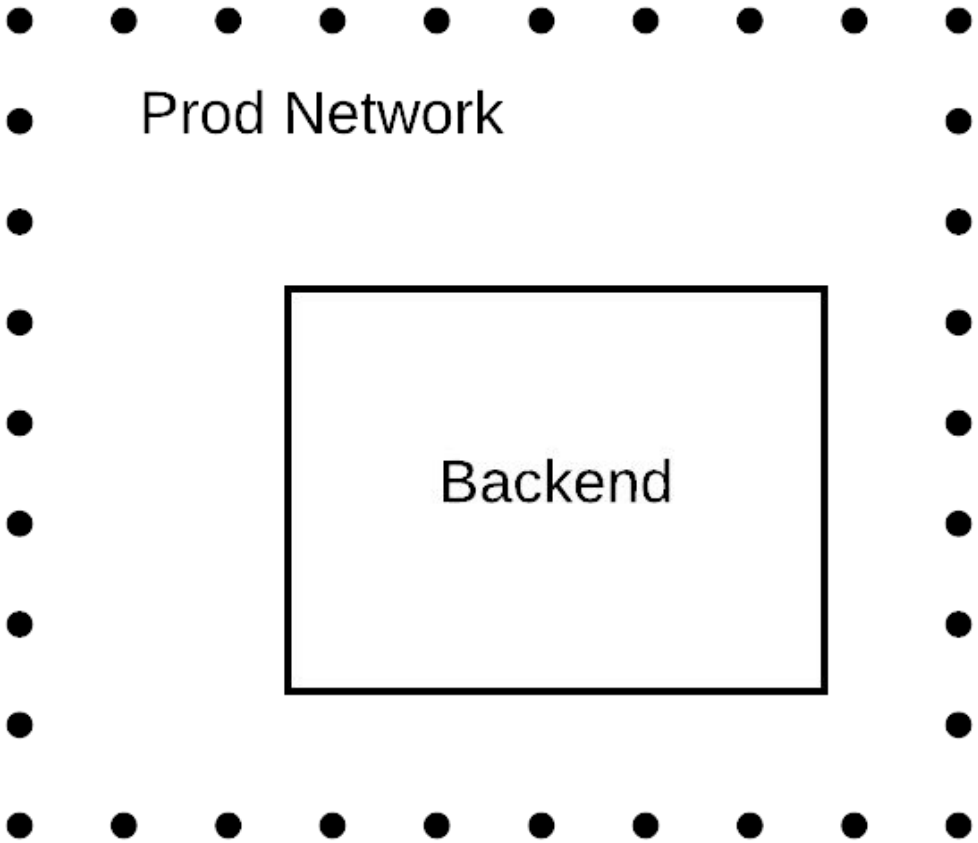
Direct Authentication

Authentication of the proximate requester or “peer”.

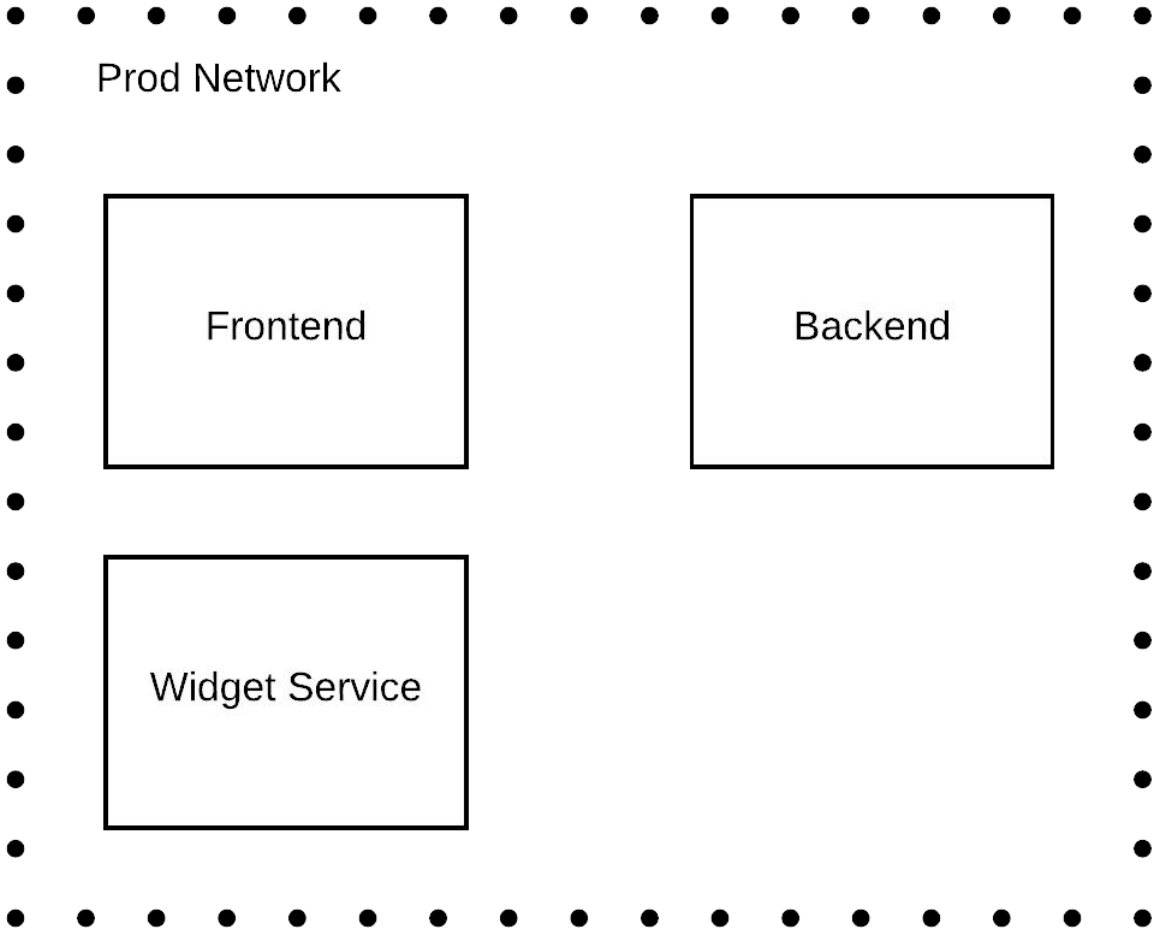
Delegated Authentication

Authentication of a requester somewhere up the call chain, not the direct requester.

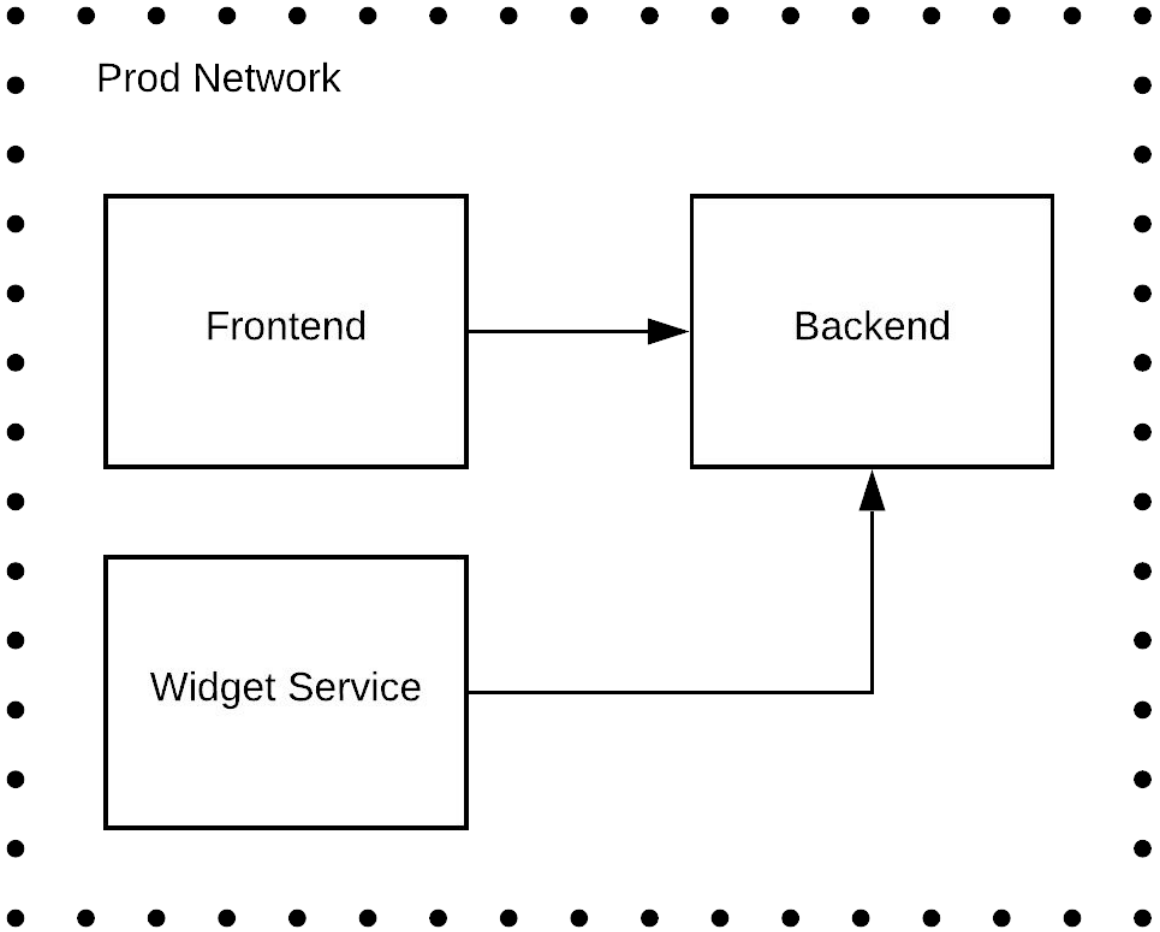
The game of telephone



The game of telephone



The game of telephone



Feature Creep

Grant access to call Backend

To Frontend

KubeCon CloudNativeCon Europe 2019

Native Service Identity

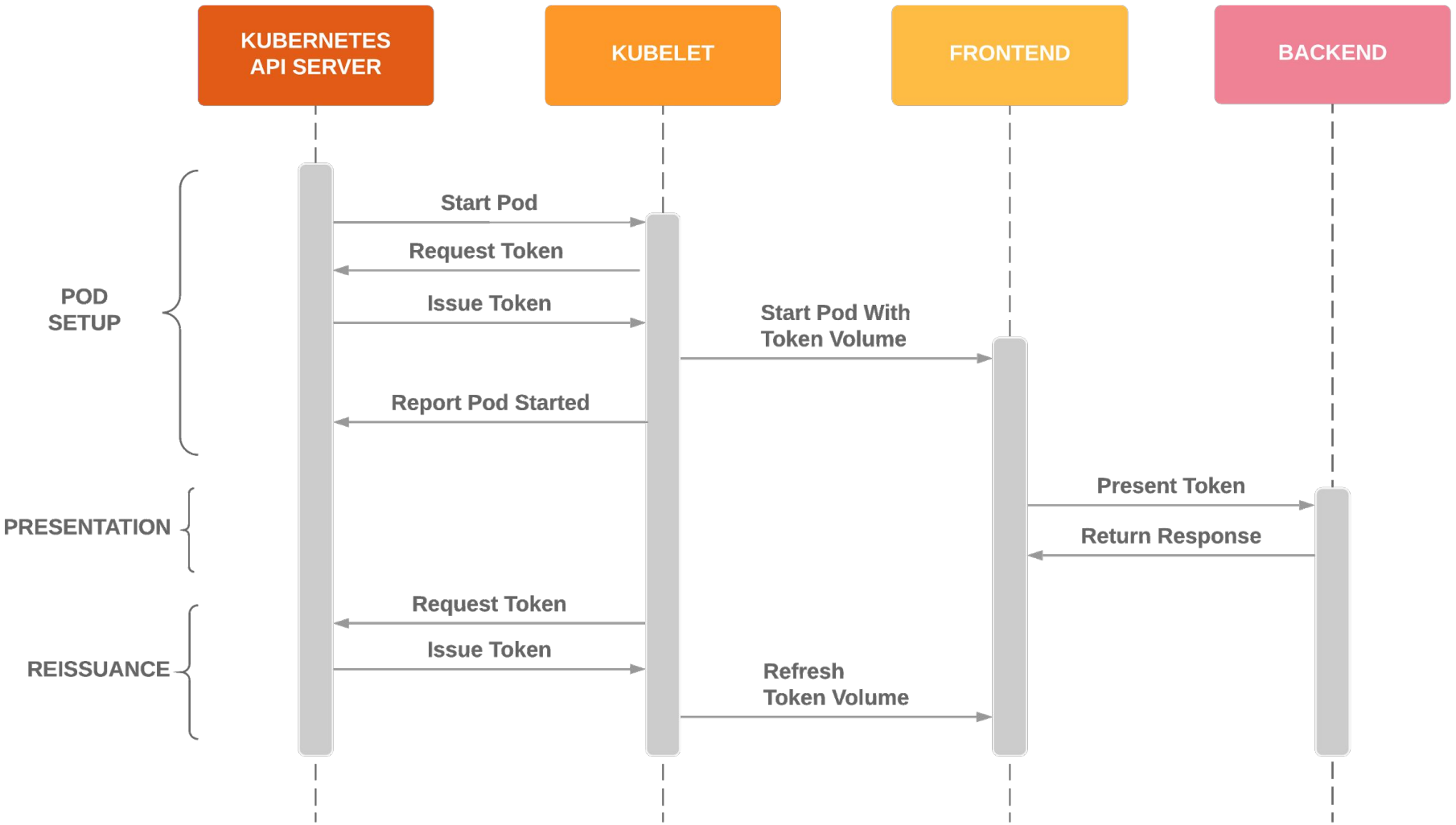
- All pods run as a service account
- Standard access control model
- Automatic credential management

Service Account Tokens

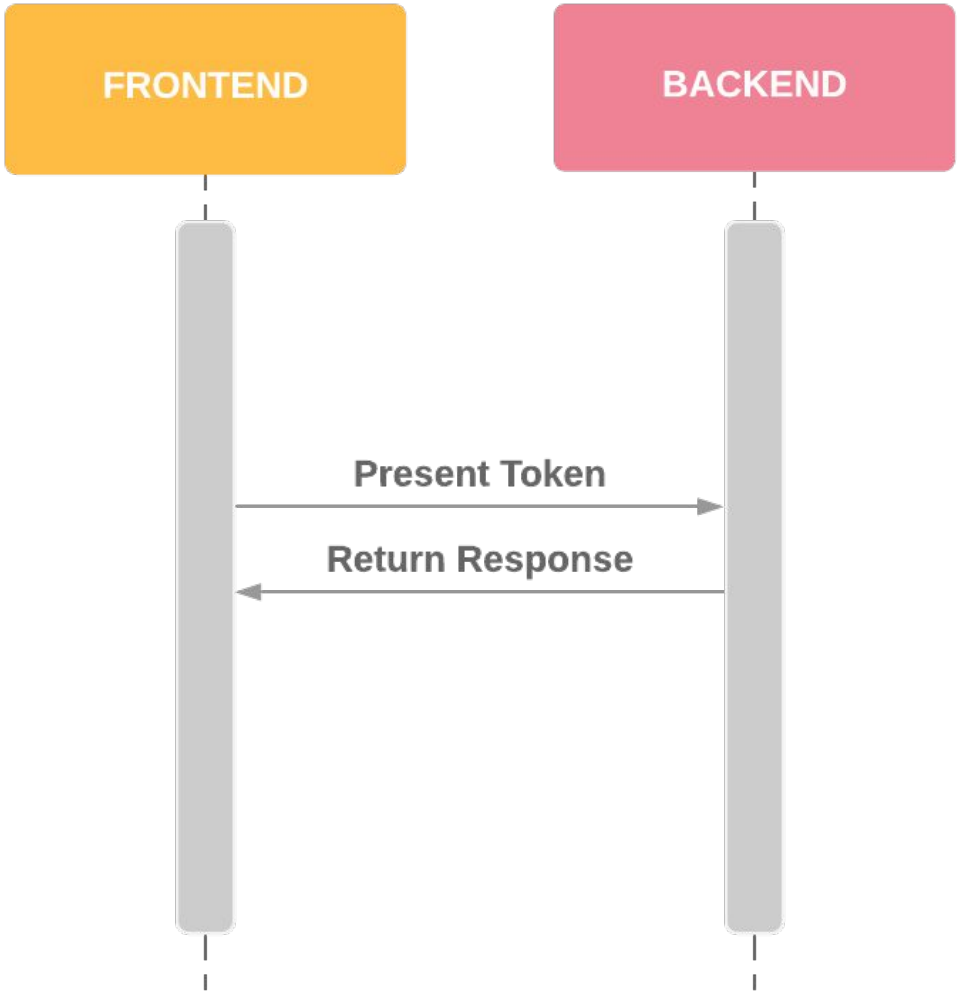
- Exposed to pods via a kubelet managed tmpfs
- Flexible verification
- Revocable via API
- Limited TTL*
- Audience binding*
- Automatic rotation*
- Never stored in etcd*

** True only for new style tokens. TokenRequest and TokenProjection features, Beta in 1.12*

Service Account Tokens



Service Account Tokens



Tokens have a major downside

- Forwardable so may be replayed
- Don’t solve server authentication

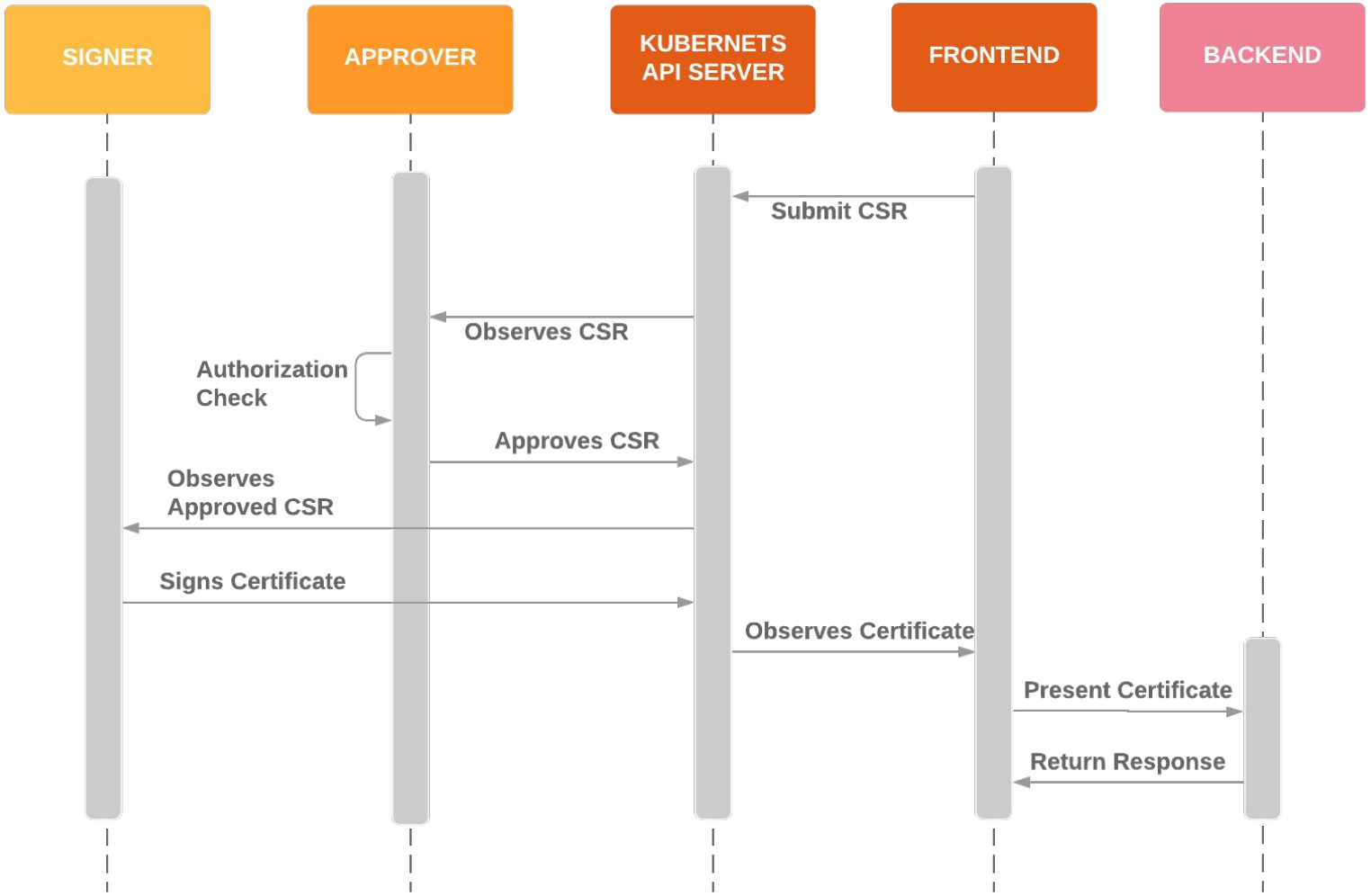
Certificates

Mutual TLS

- Provides server authentication
- Channel bound

Kubernetes Certificates API is flexible but requires some integration. But Istio can do all the heavy lifting for you.

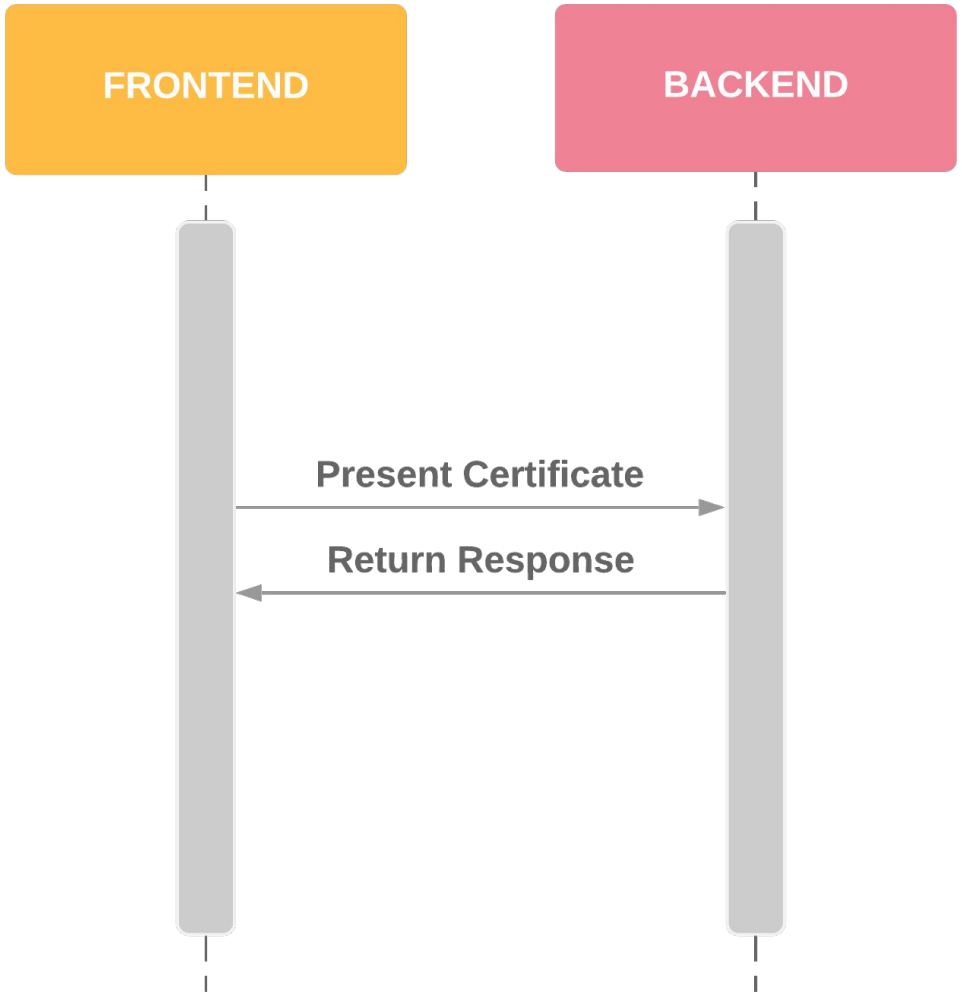
Kubernetes Certificates API



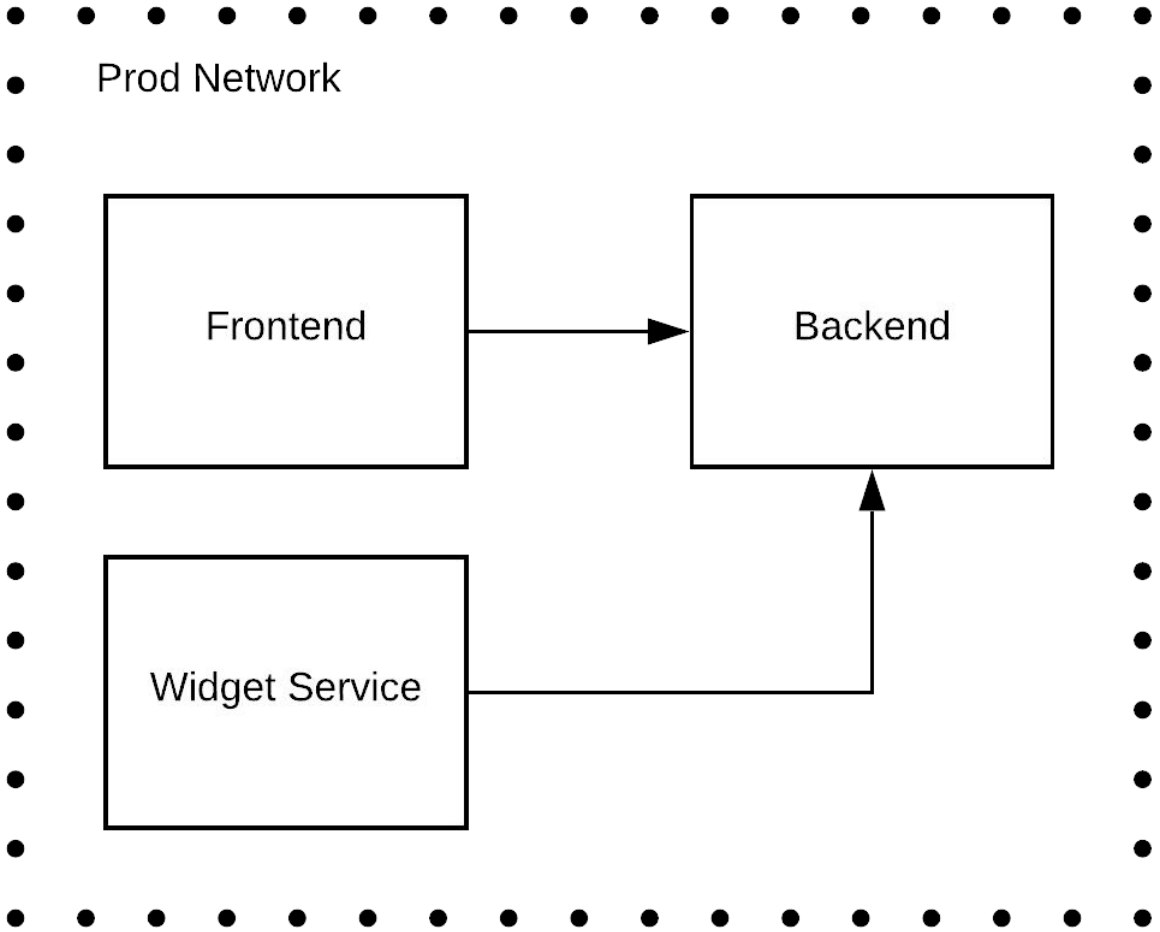
Istio mTLS

- Istio does all the heavy lifting for you
- Istio Citadel provides an API to exchange a service account token for an mTLS certificate
- Istio node agent does this automatically

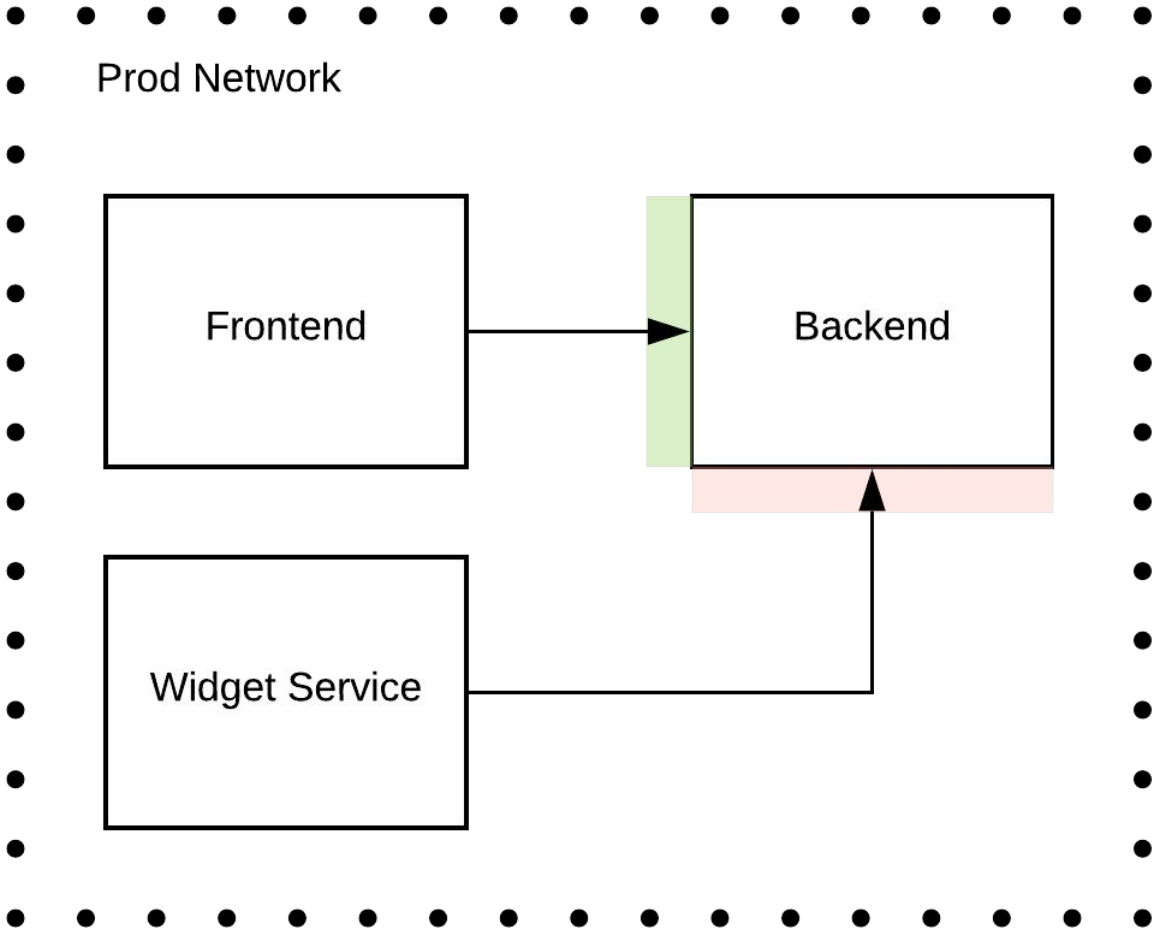
Istio



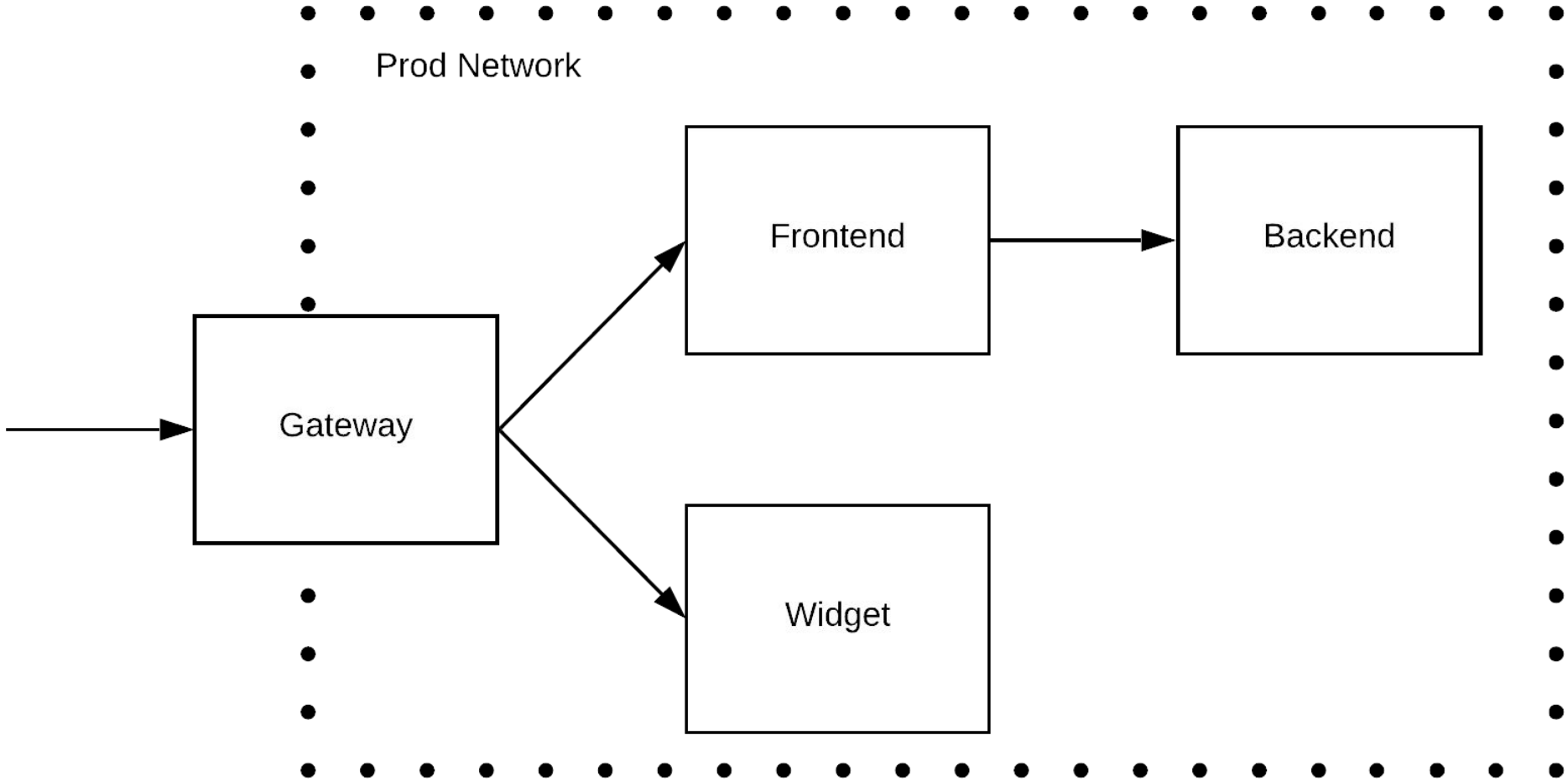
The game of telephone



The game of telephone



The game of telephone

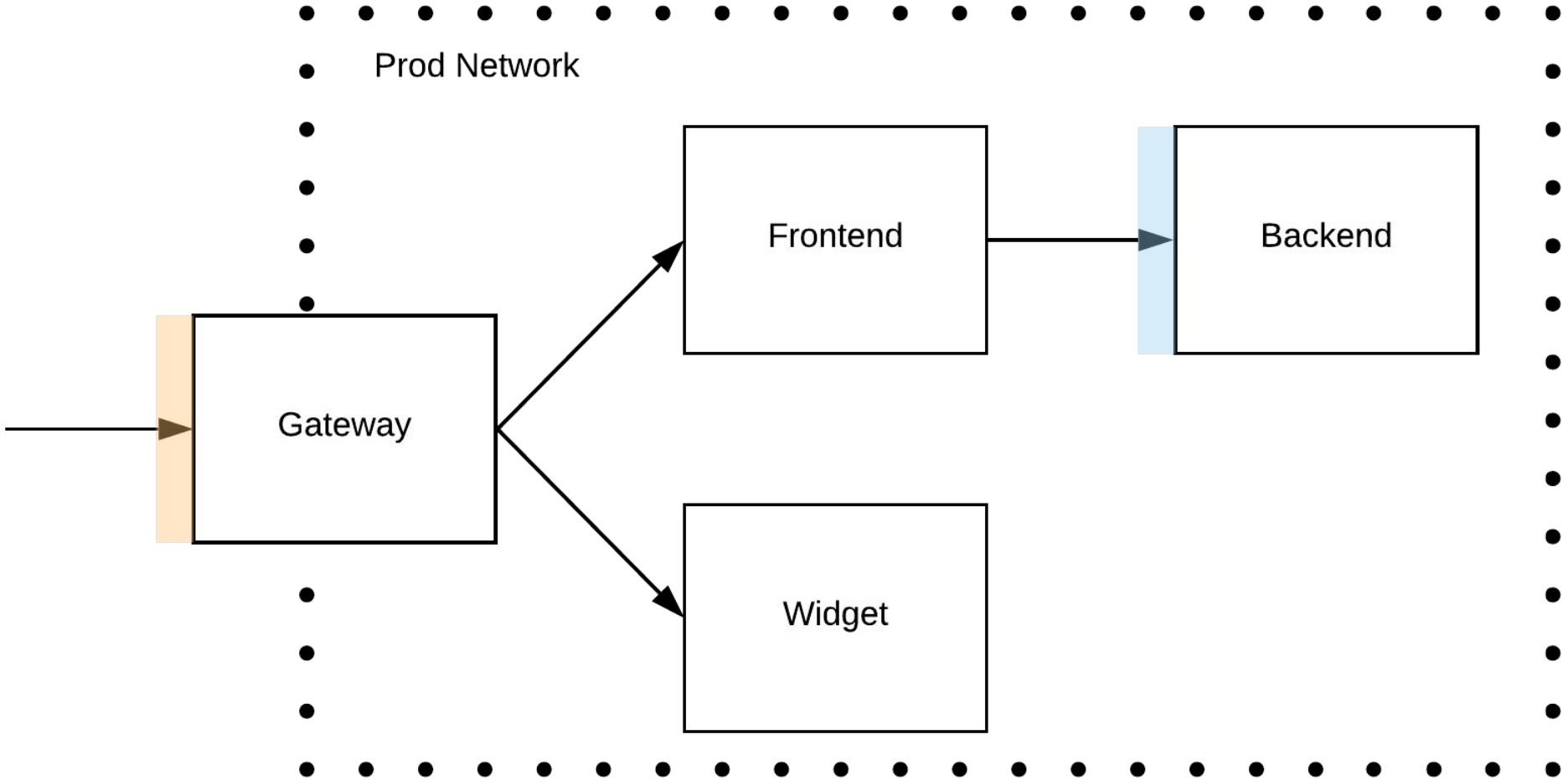


Feature Creep

Grant access to user A's data

To User A

The game of telephone

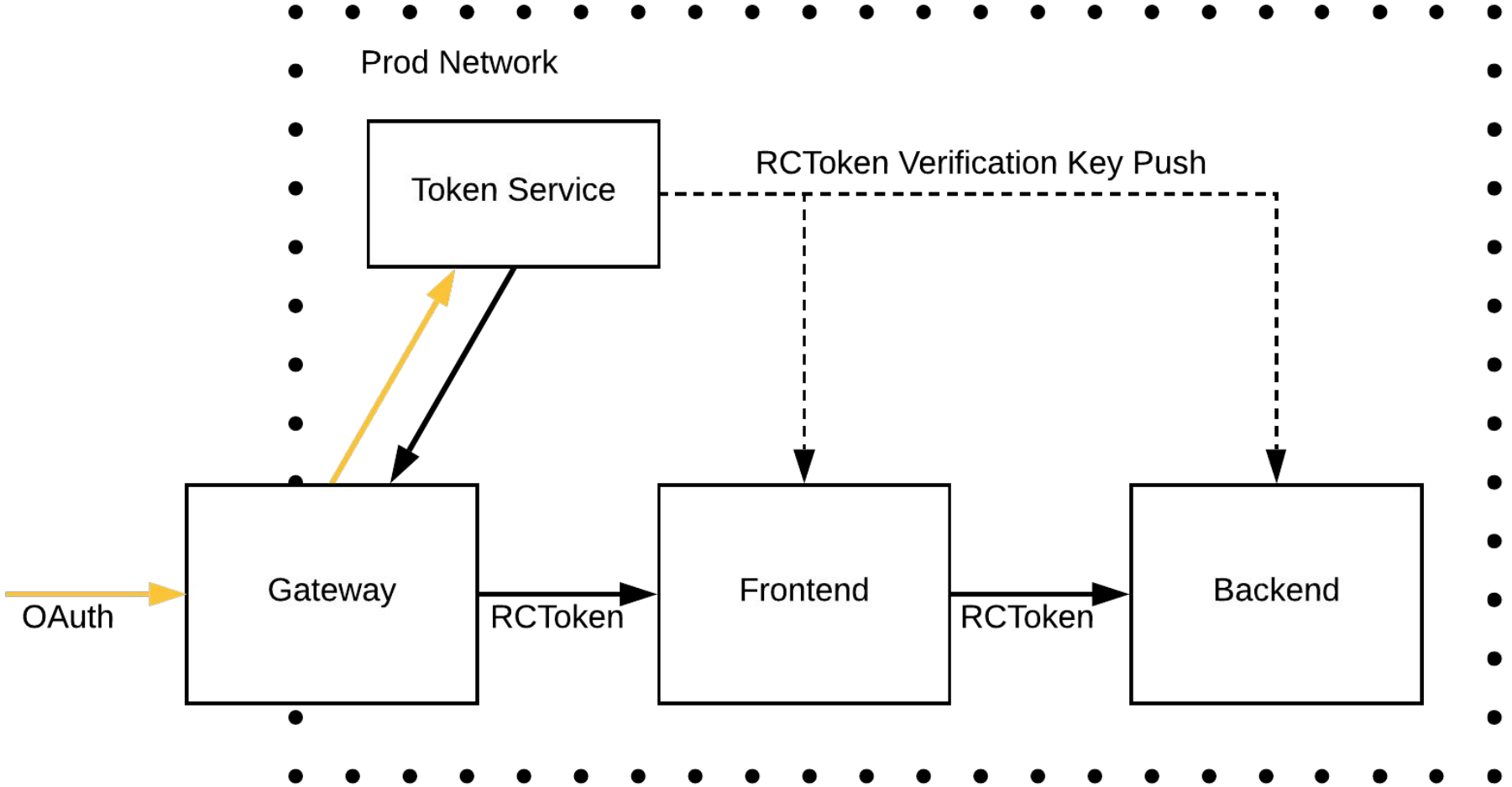


Request Context Token

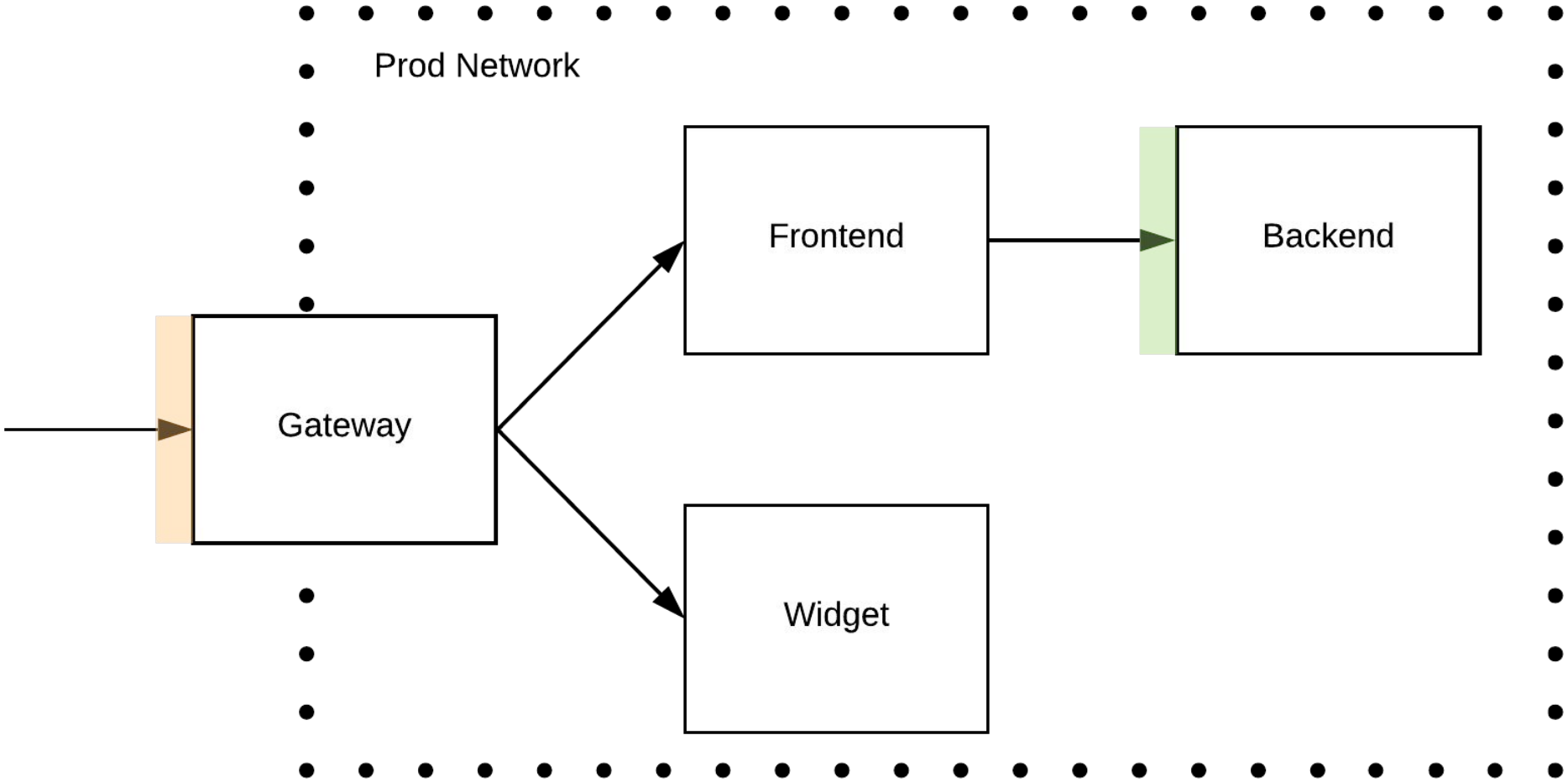
- Capture request context at ingress
- Packages attributes to handoff to upstreams
- Support for arbitrary attributes
  - Source IP
  - End user identity (i.e. request originator)

Warning: Early Development

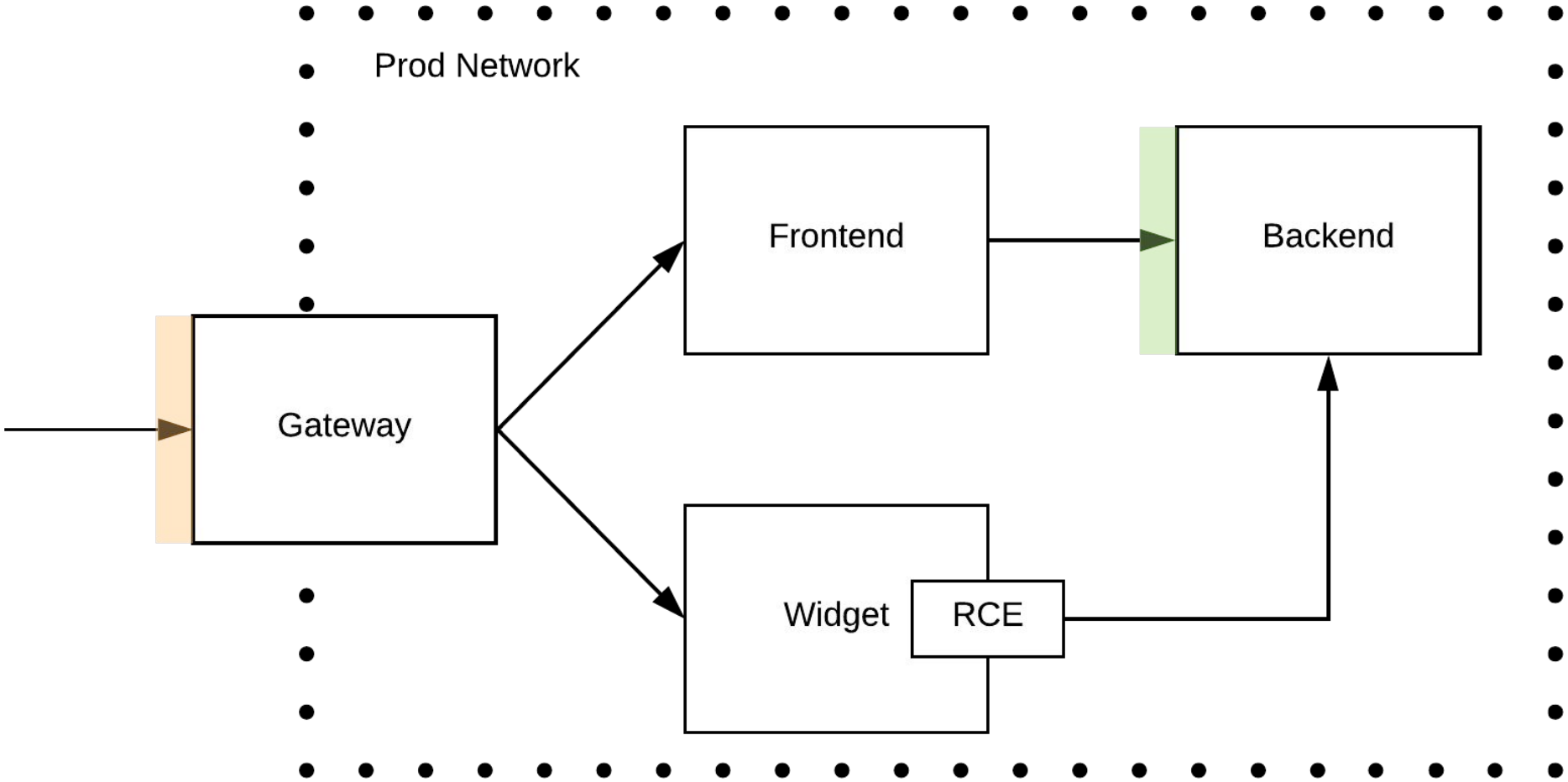
Istio RCToken



The game of telephone



The game of telephone



Feature Creep

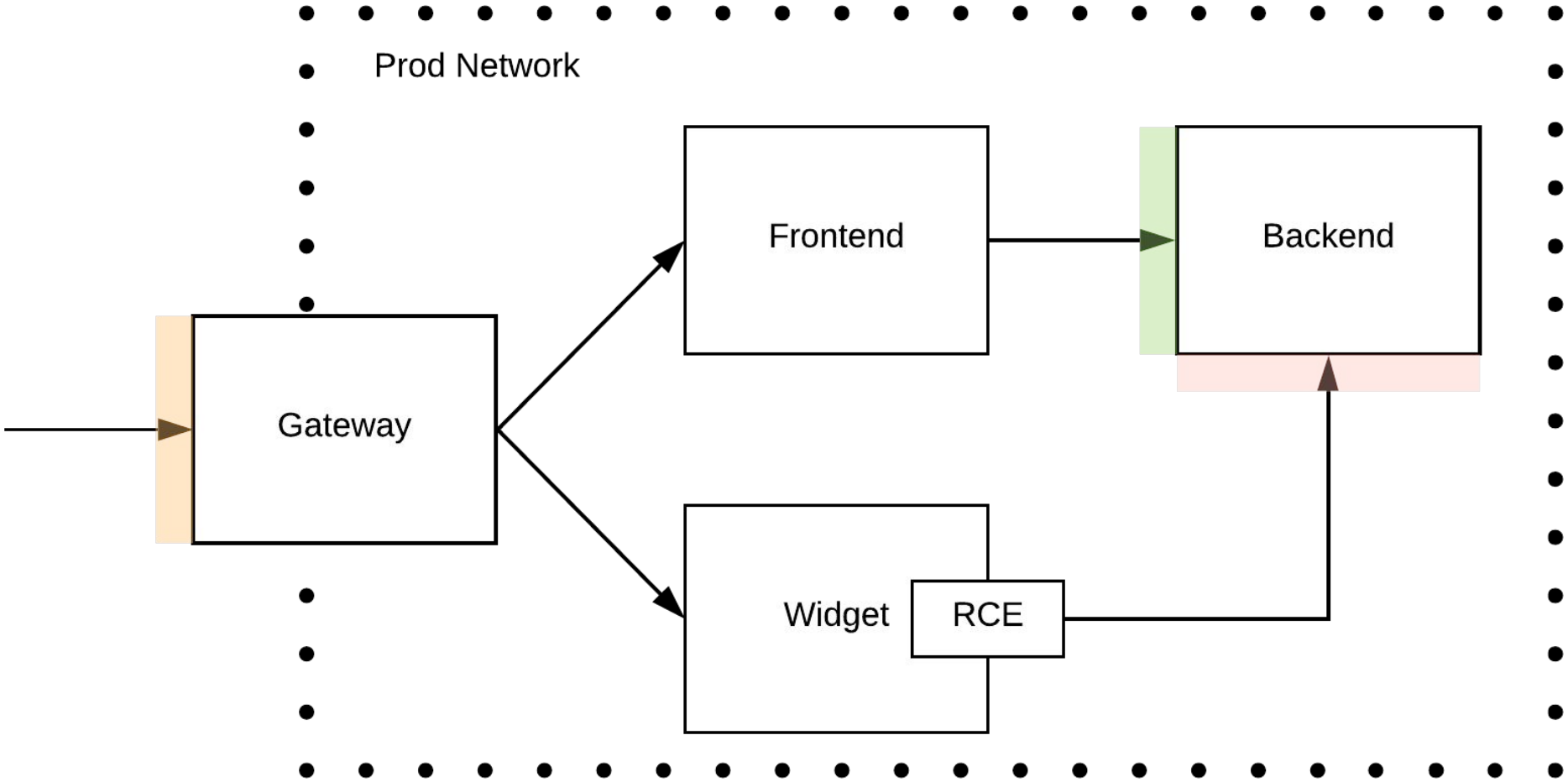
Grant access to call Backend

To Frontend

Grant access to user A's data

To User A

The game of telephone

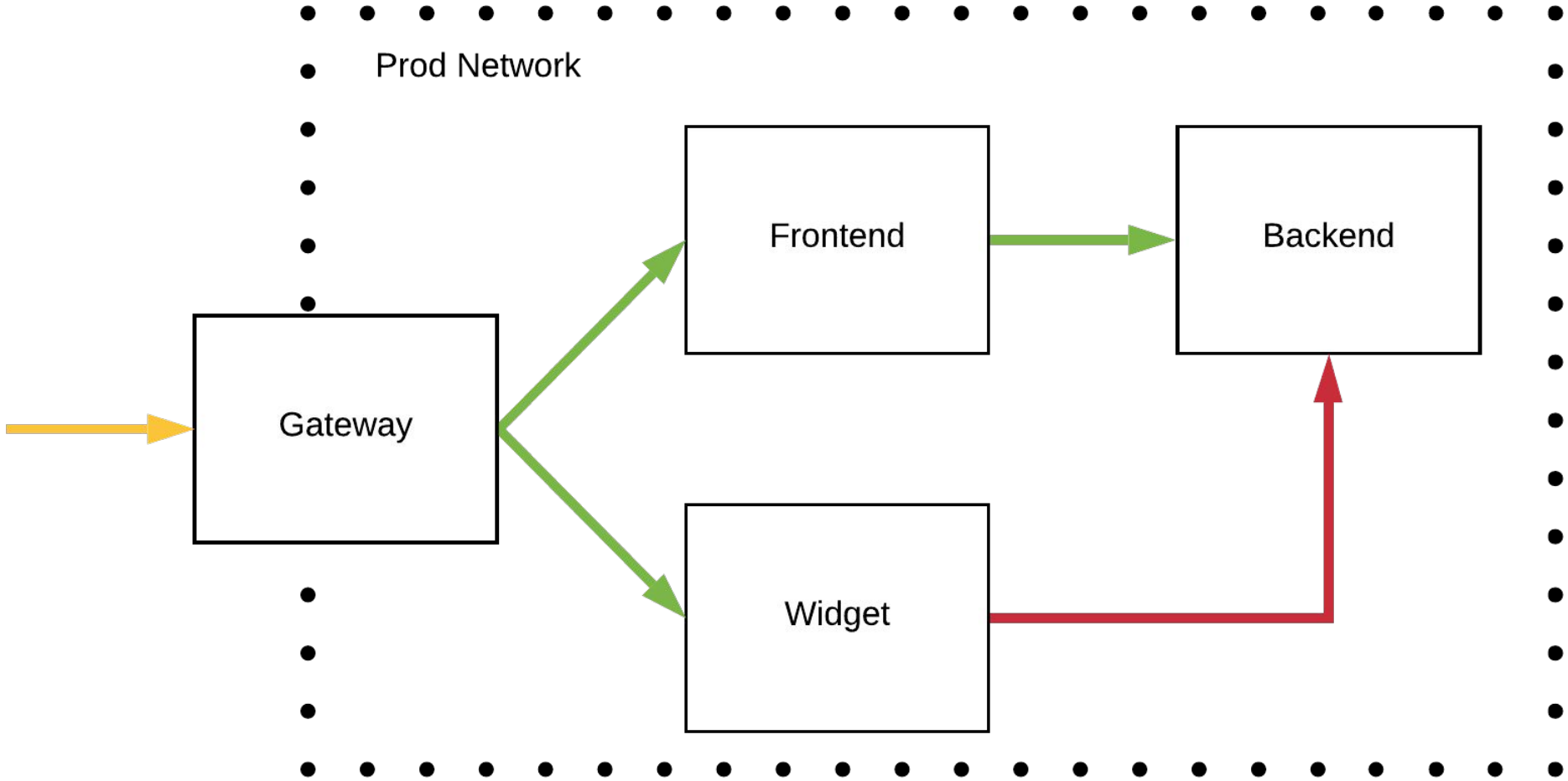


Feature Creep

Grant access to user A's data

To Frontend if it can prove recent interaction with user A

The game of telephone



Grant access to Backend

- To Service B
  - If request originated in my prod VPC
  - If service B was verifiably built by my CI system

Feature Creep

Grant access to user A's data

- To Employee C
  - With associated justification
    - e.g. support ticket, bug ID, page ID
  - If request originated on company issued device

Full audit history

Record the who, what, when and why of all accesses to sensitive data.

Accountability


```
$ cat /etc/sudoers.lecture
We trust you have received the usual lecture from the local
System Administrator. It usually boils down to these three
things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.
```

© Todd C. Miller <Todd.Miller@sudo.ws>
https://www.sudo.ws/

Back to the problem

How do we create an environment that maintains a sufficiently high level of assurance on *user data*?

Is authentication the answer?

Bad news:

No, not even close. The complexity of the problem requires:

- A holistic approach
- Sustained diligence

And nothing is perfect.

Is authentication the answer?

Good news:

However, it is foundational in a holistic approach. It enables:

- Granular, least-privileged authorization
- Complete audit history

Closing thoughts

What makes for a good solution?

- Easy to adopt
- Hard to use incorrectly, noisy to circumvent
- Applied consistently across all services
- Generally useful, built on open standards
- Easy to evolve and extend (in and out of core)

Closing thoughts

What belongs in Kubernetes?

- Extension points that allow experimentation in systems built on Kubernetes.
- Improvements that harden core infrastructure (but move cautiously)

Closing thoughts

Call to action

- Continue thinking about it
- From the basics, improve incrementally
- Give feedback:
  - What works well?
  - What didn't work?
  - What could we do better?

Closing thoughts

Shout out!

- SPIFFE and SPIRE
- SIG Auth
- Istio Security Working Group

Other Resources

- SPIFFE and SPIRE
- SIG Auth
- Istio Security WG