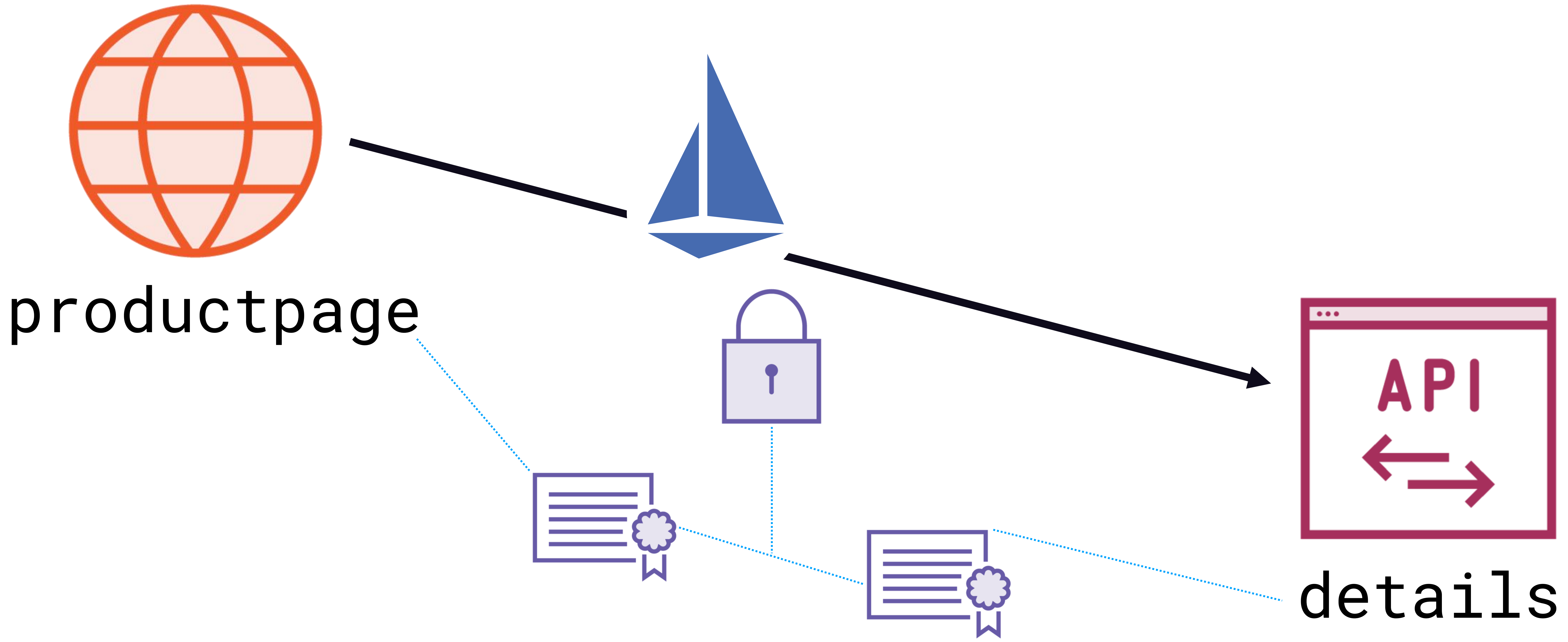
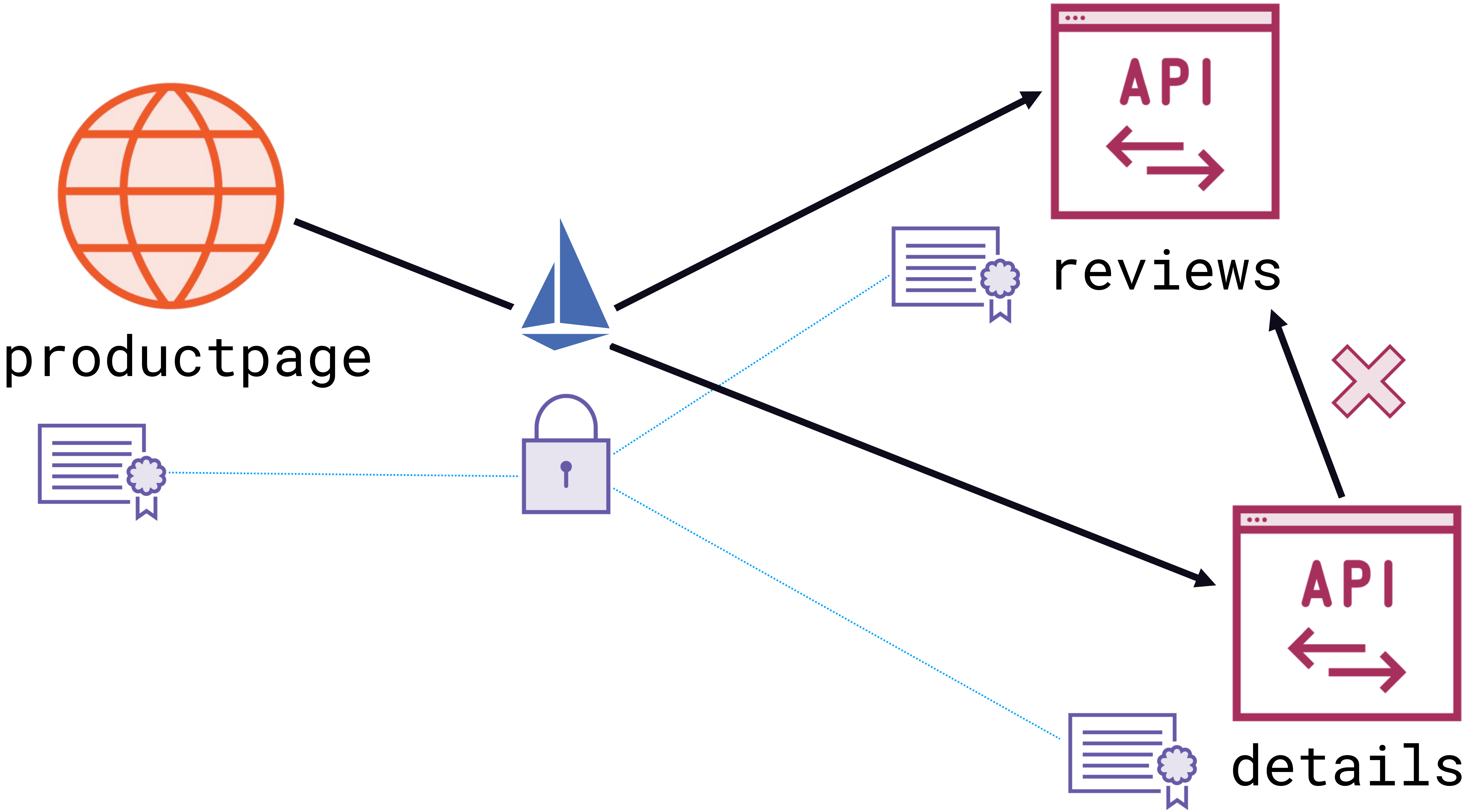
Layering on Security

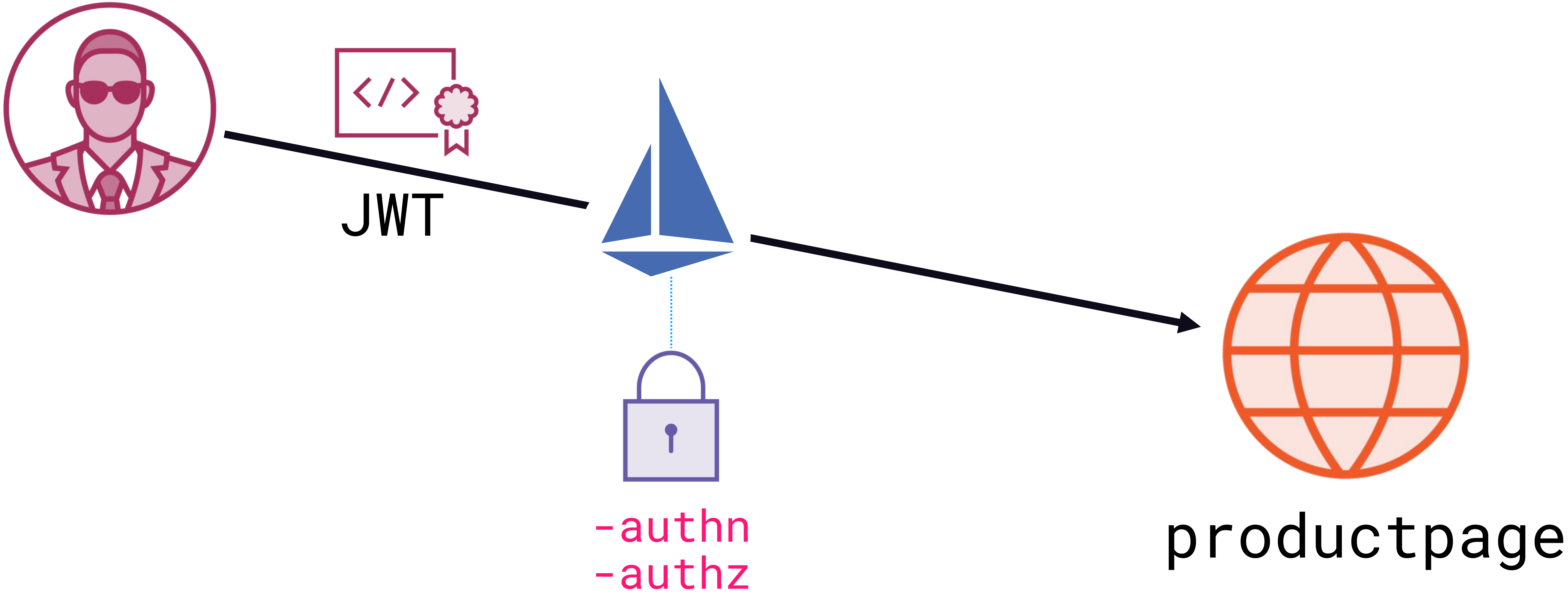
Elton Stoneman

Freelance Consultant and Trainer

@EltonStoneman | blog.sixeyed.com


productpage
API
details

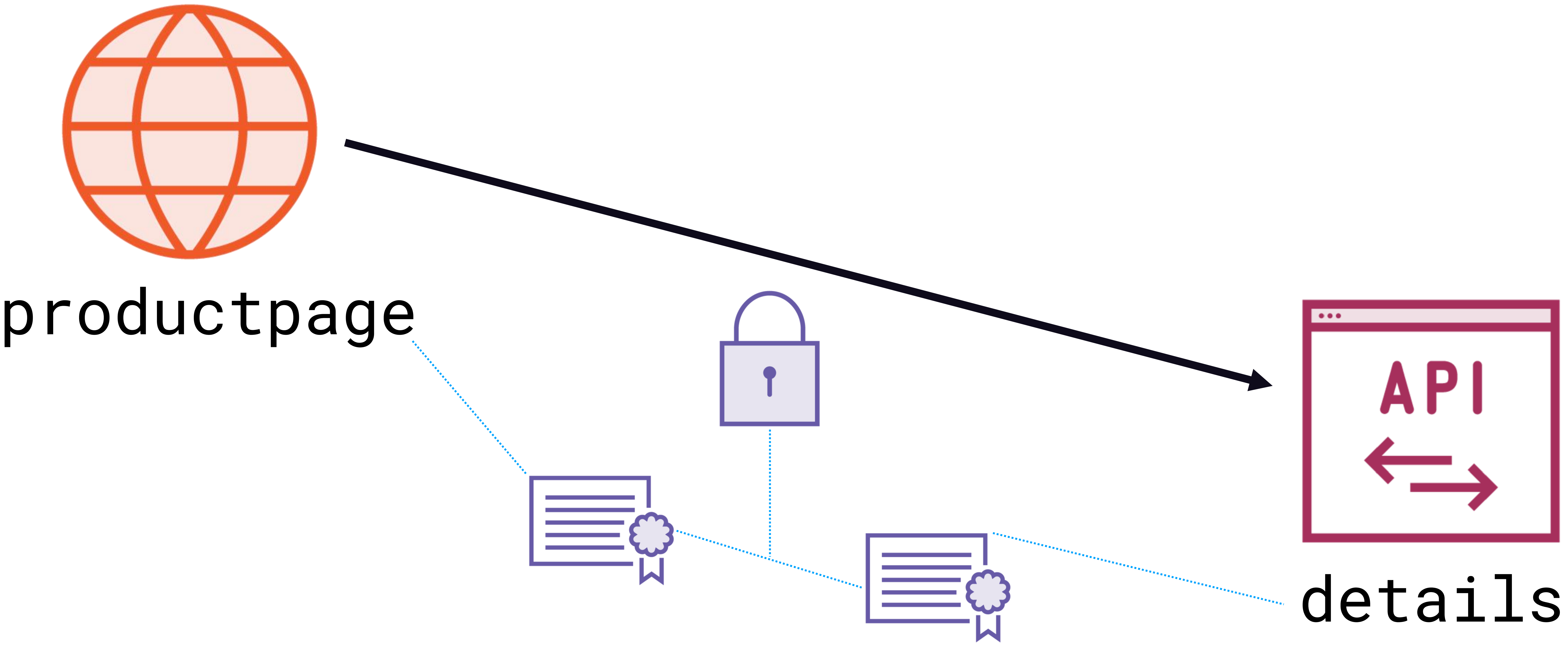

API
reviews
productpage
API
details

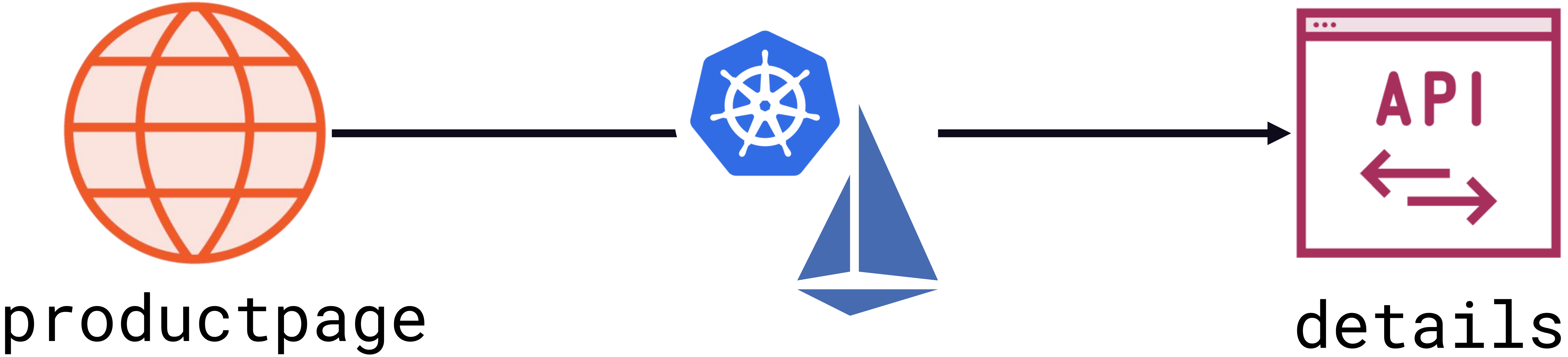


JWT

-authn
-authz

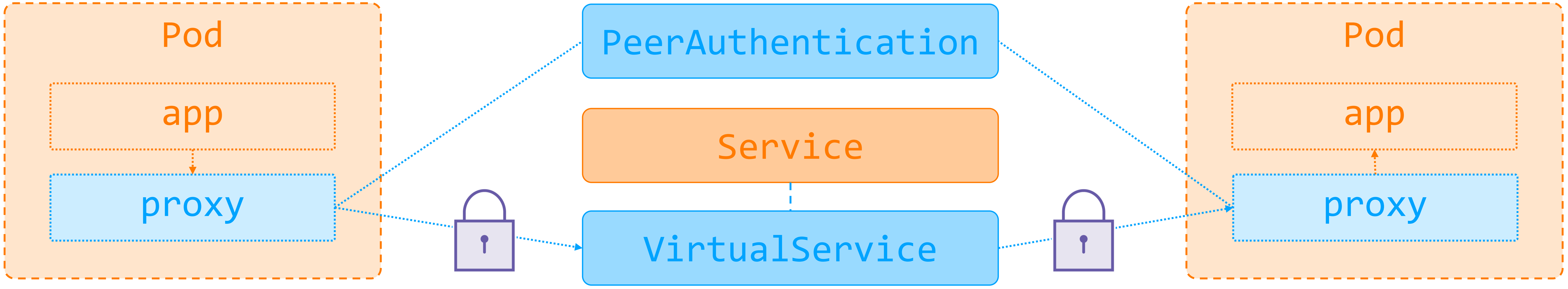
productpage


productpage
API
details



productpage

details



Pod

app

proxy

PeerAuthentication

Service

VirtualService

Pod

app

proxy

```yaml
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
  namespace: bookinfo    # istio-system
spec:
  mtls:
    mode: STRICT
```

◀ **Authentication between proxies**

◀ **Scope – namespace or whole-mesh**

◀ **Require mutual TLS**

Scoping Authentication

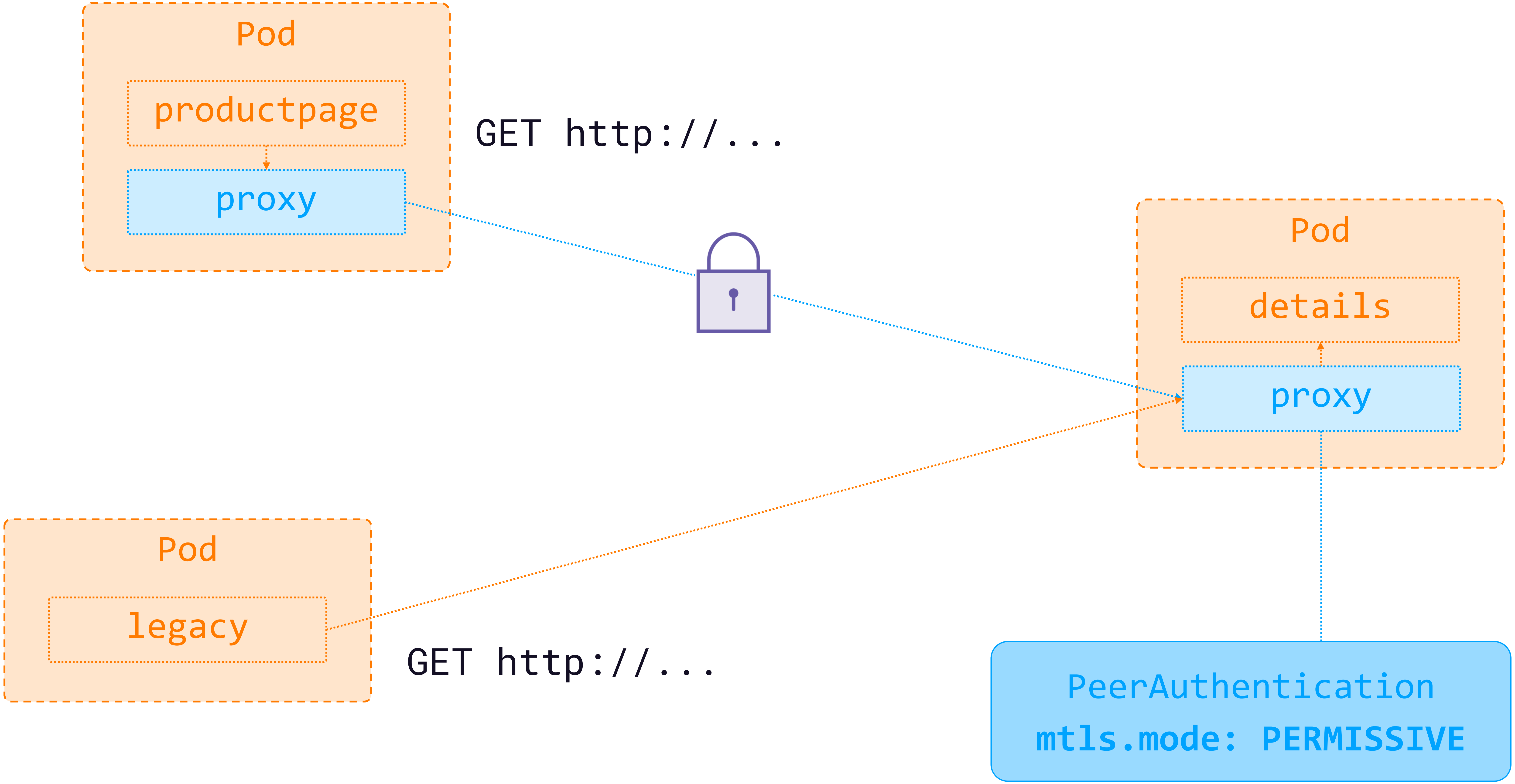
Extra security for specific components

authn-namespace.yaml

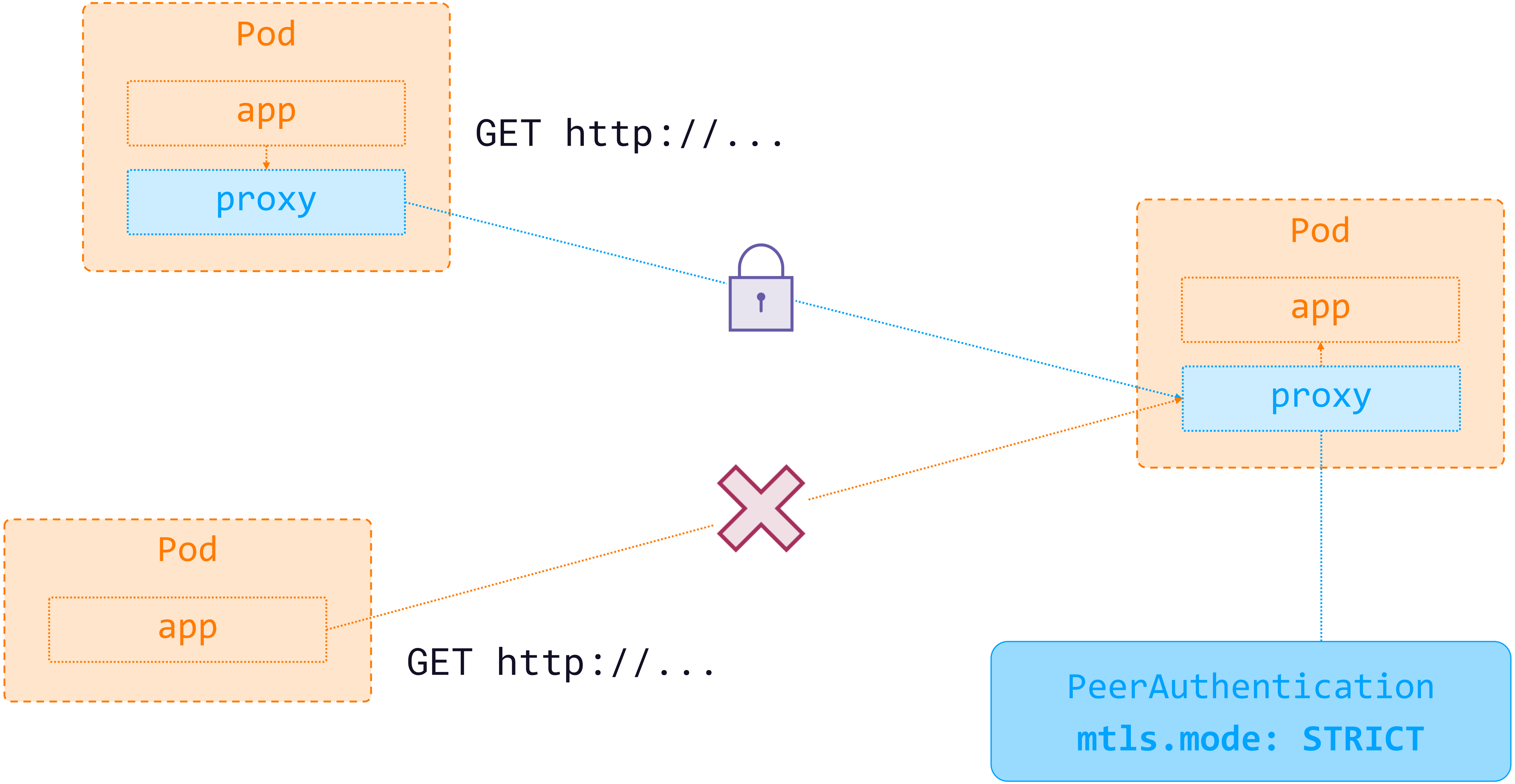
```yaml
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
  namespace: bookinfo
spec:
  mtls:
    mode: PERMISSIVE
```

authn-details.yaml

```yaml
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: details
  namespace: bookinfo
spec:
  selector:
    matchLabels:
      app: details
  mtls:
    mode: STRICT
```



Pod
productpage
proxy
GET http://...
Pod
details
proxy
Pod
legacy
GET http://...
PeerAuthentication
mtls.mode: PERMISSIVE



Pod

app

proxy

GET http://...

Pod

app

proxy

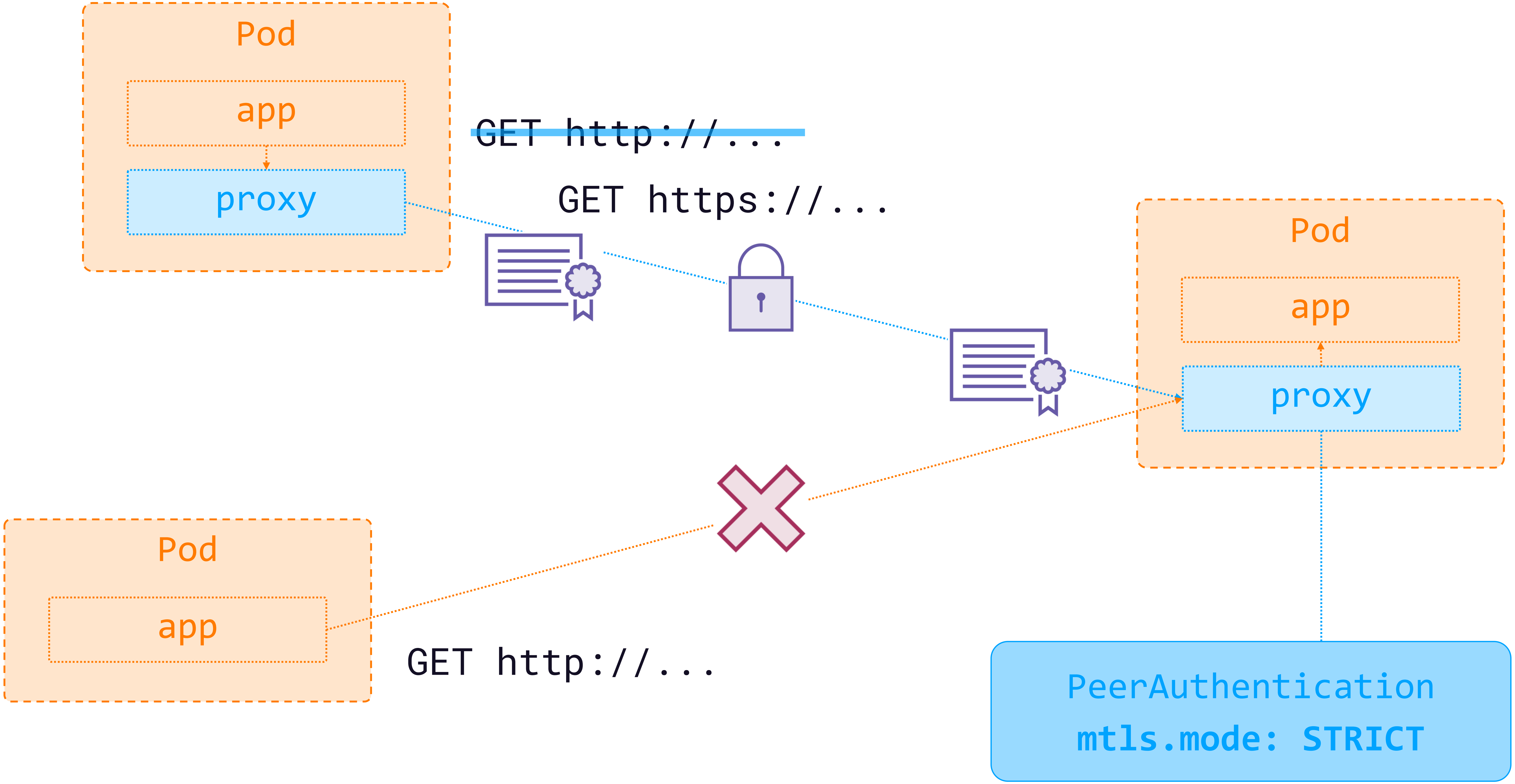
Pod

app

GET http://...

PeerAuthentication

mtls.mode: STRICT



Pod

app

proxy

~~GET http://...~~

GET https://...

Pod

app

proxy

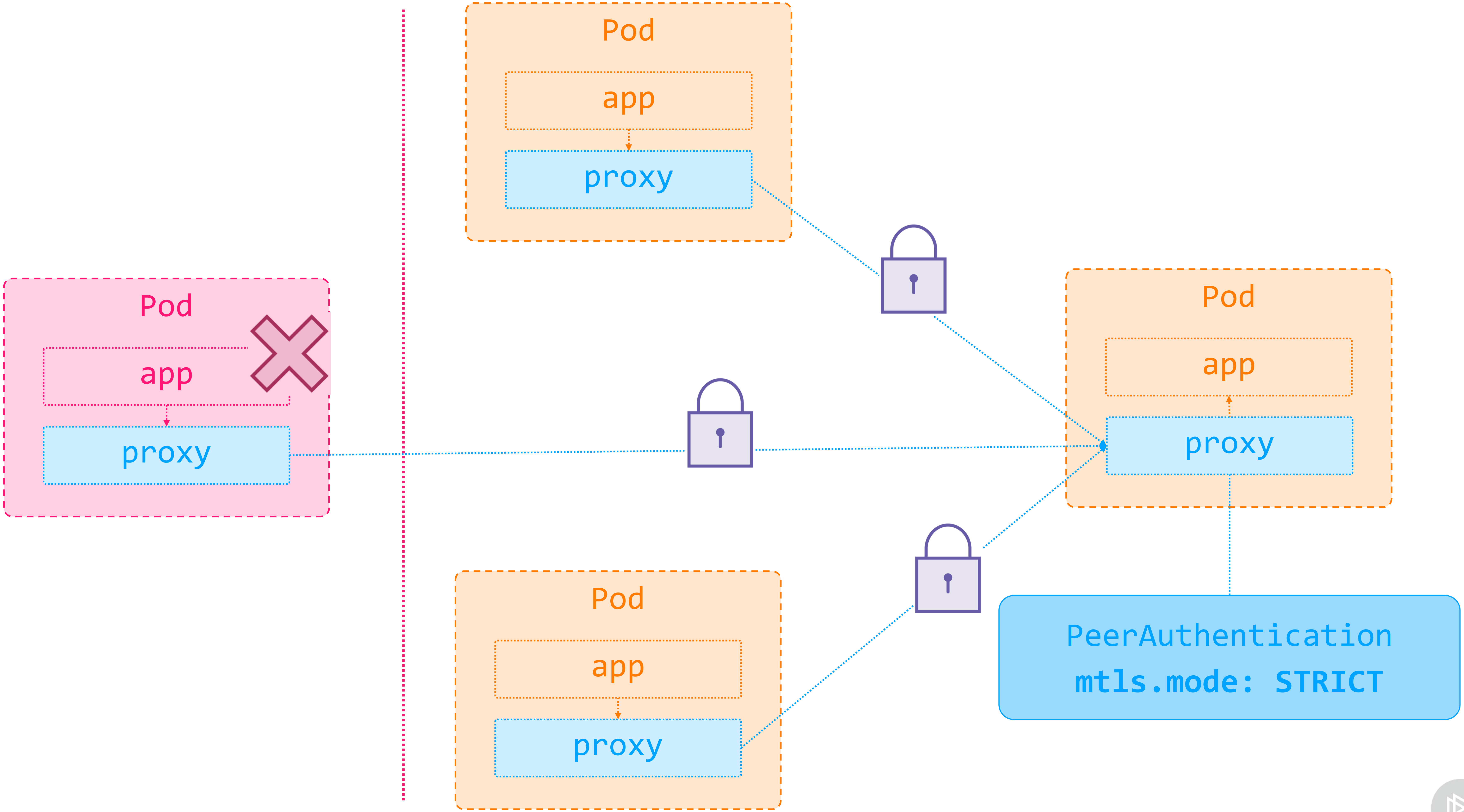
Pod

app

GET http://...

PeerAuthentication

mtls.mode: STRICT


Pod
app
proxy
Pod
app
proxy
Pod
app
proxy
Pod
app
proxy
PeerAuthentication
mtls.mode: STRICT

Opting Out

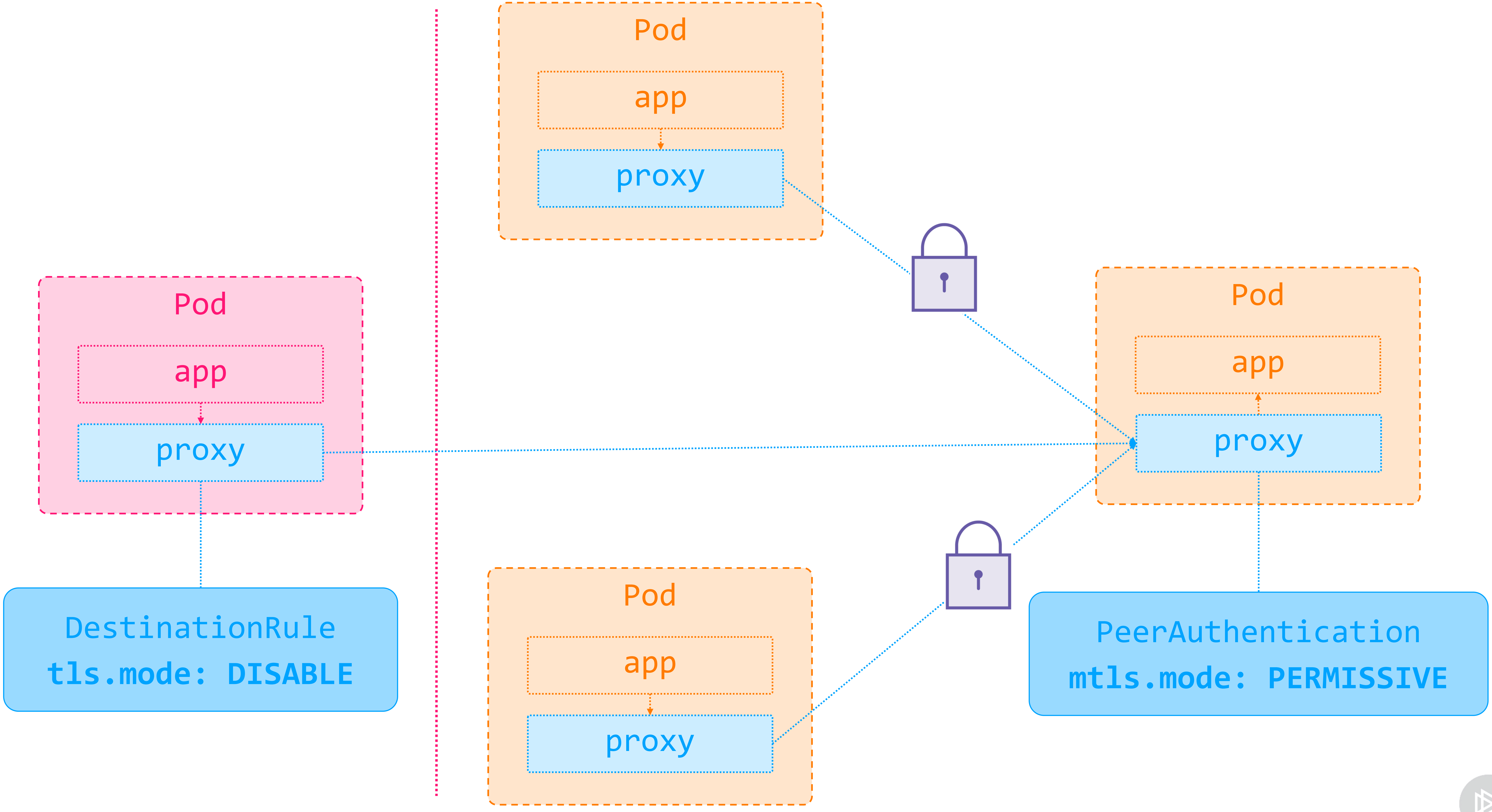
Clients unable to use mTLS

virtual-service.yaml

```yaml
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  namespace: legacy-with-istio
  name: details
spec:
  hosts:
  - details.bookinfo.svc...
  http:
  - route:
    - destination:
        host: details.bookinfo.svc...
```

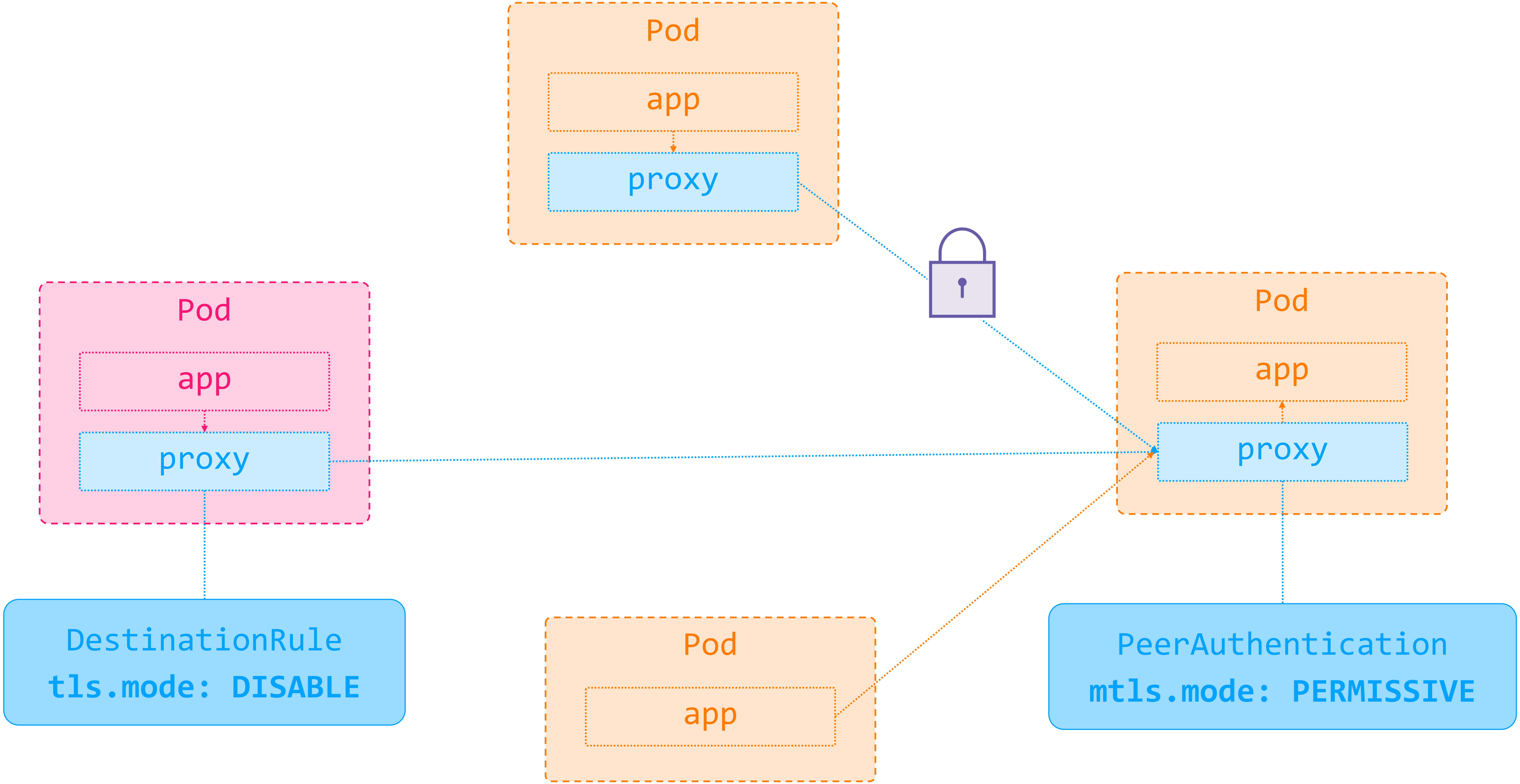
destination-rule.yaml

```yaml
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  namespace: legacy-with-istio
  name: details
spec:
  host: details.bookinfo.svc...
  trafficPolicy:
    tls:
      mode: DISABLE
```



DestinationRule
tls.mode: DISABLE

PeerAuthentication
mtls.mode: PERMISSIVE

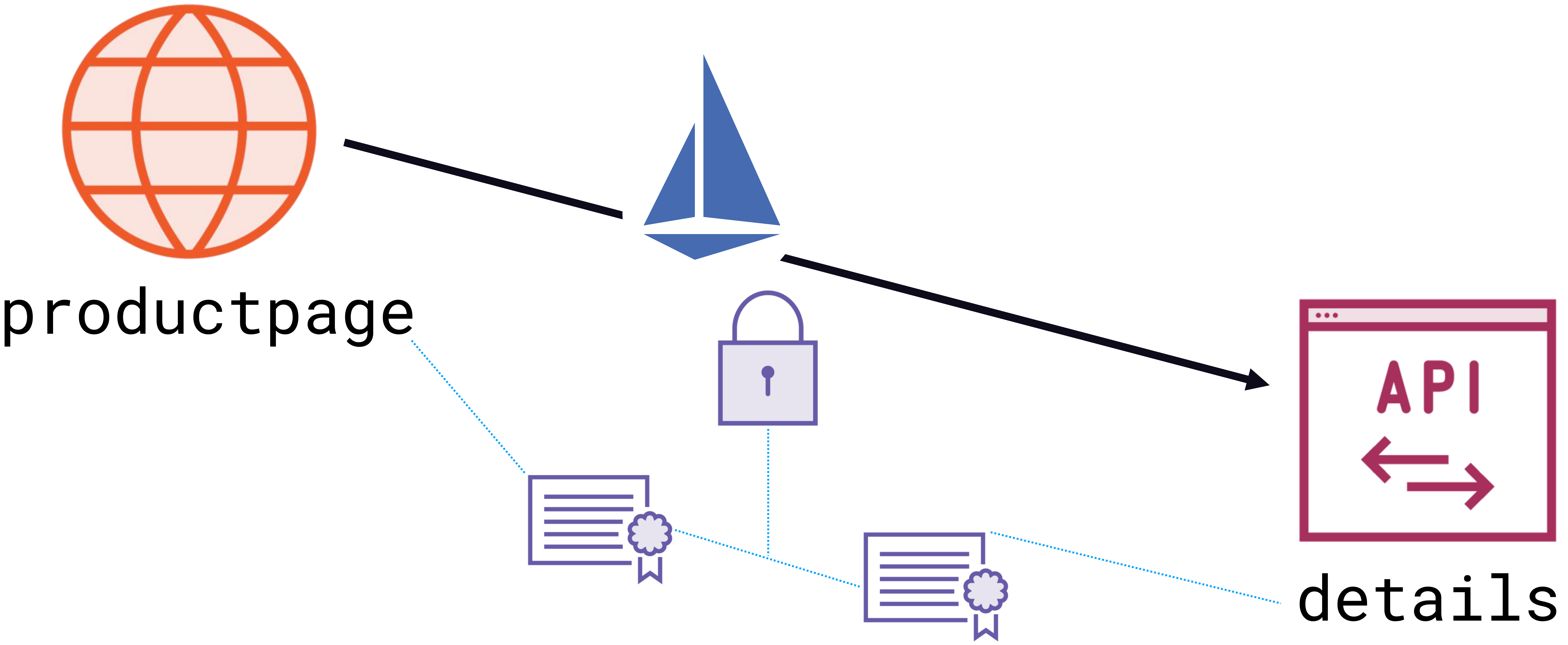


Pod
app
proxy
Pod
app
proxy
Pod
app
proxy
DestinationRule
tls.mode: DISABLE
Pod
app
PeerAuthentication
mtls.mode: PERMISSIVE

Demo

Securing services with mutual TLS

- Accessing services over HTTP
- Applying mTLS policy
- Understanding PeerAuthentication


productpage
API
details

Enforcing Mutual TLS

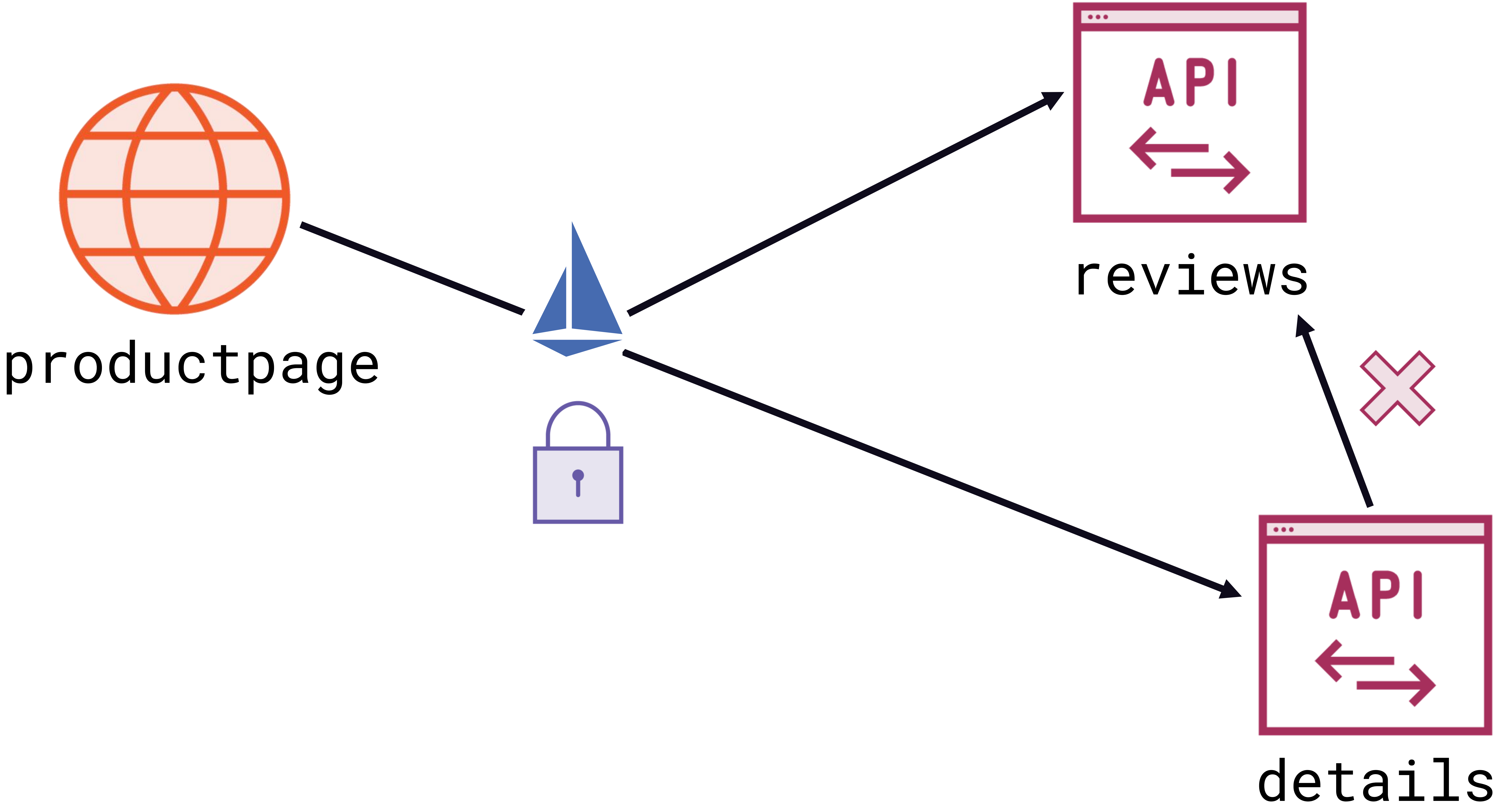
Namespace and component scope

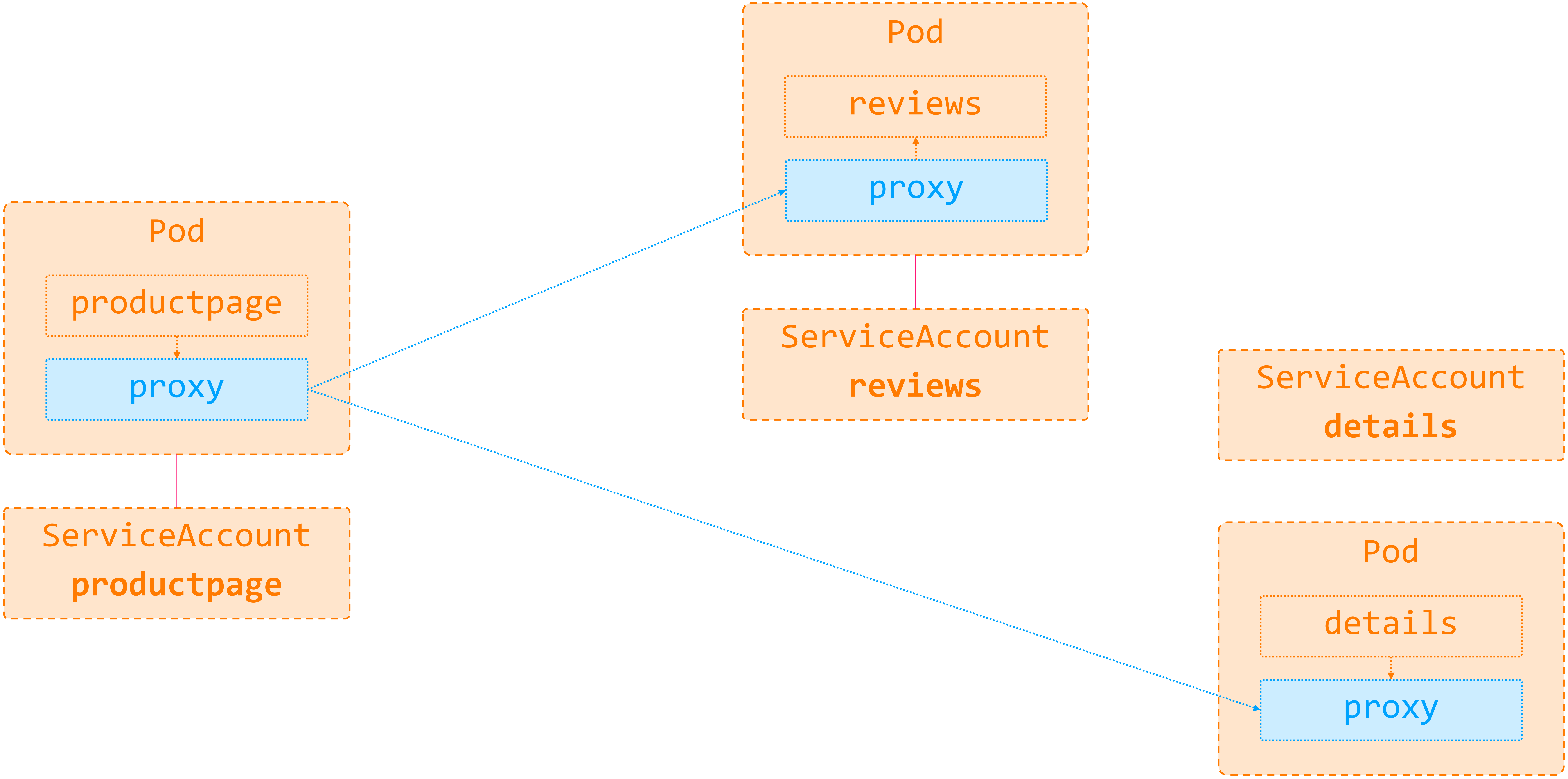
mtls-namespace.yaml

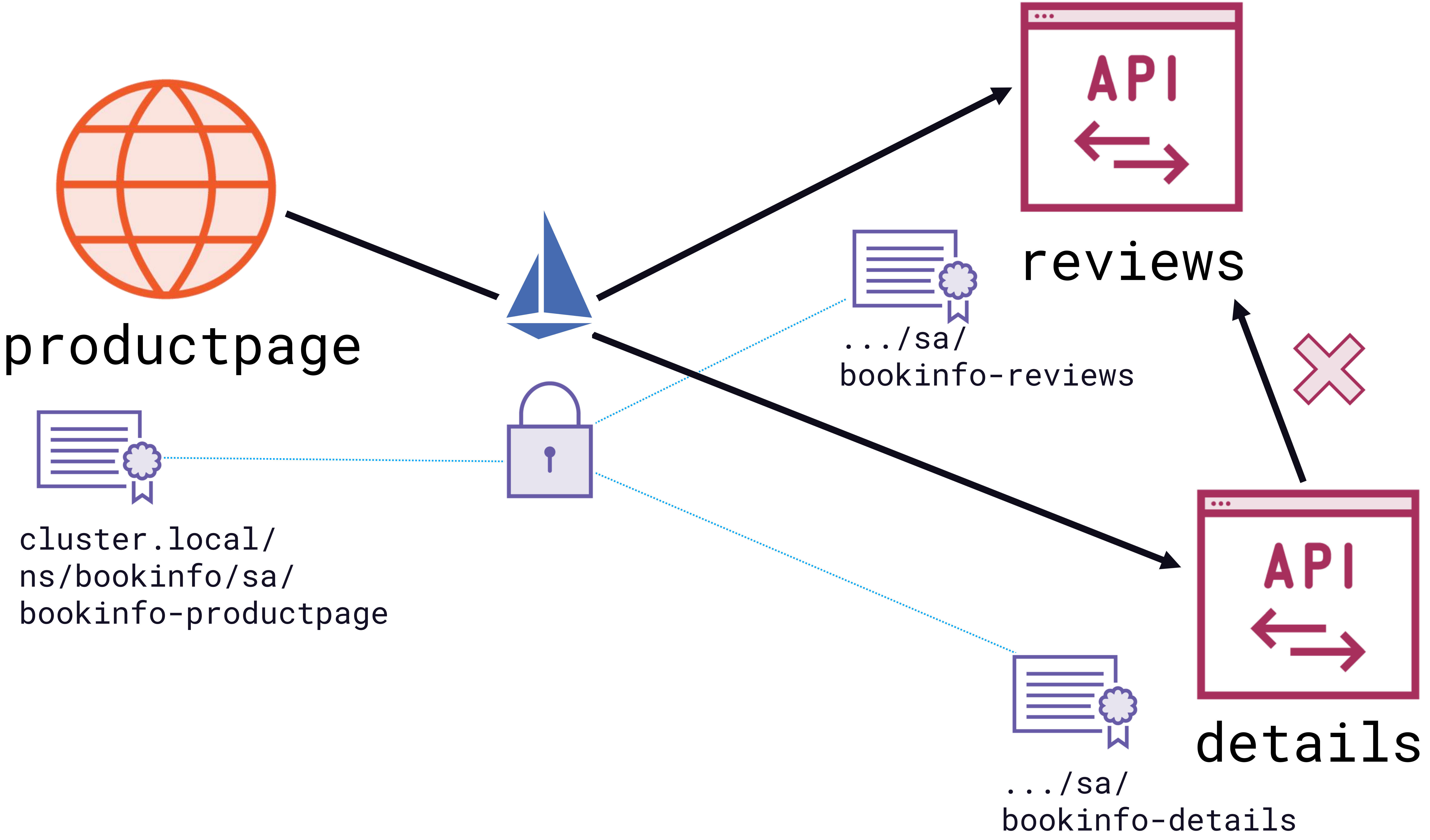
```yaml
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
  namespace: bookinfo
spec:
  mtls:
    mode: STRICT
```

mtls-component.yaml

```yaml
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: legacy
  namespace: bookinfo
spec:
  selector:
    matchLabels:
      authn: legacy
  mtls:
    mode: PERMISSIVE
```


API
reviews
productpage
API
details


Pod
reviews
proxy
ServiceAccount
reviews
Pod
productpage
proxy
ServiceAccount
productpage
ServiceAccount
details
Pod
details
proxy



productpage

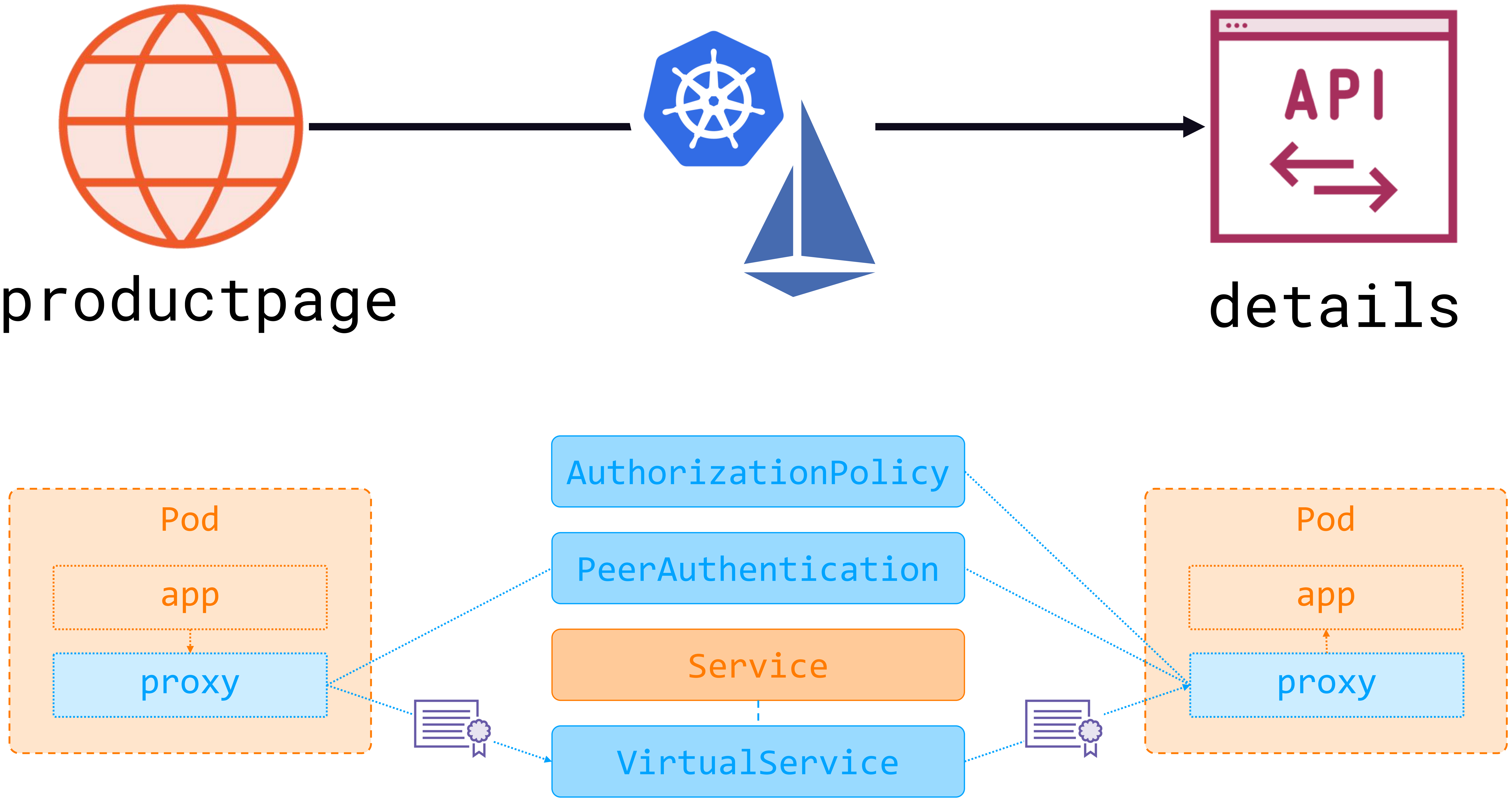
cluster.local/
ns/bookinfo/sa/
bookinfo-productpage

reviews

.../sa/
bookinfo-reviews

details

.../sa/
bookinfo-details



productpage

details

Pod

app

proxy

AuthorizationPolicy

PeerAuthentication

Service

VirtualService

Pod

app

proxy

```yaml
apiVersion: security.istio.io/v1
kind: AuthorizationPolicy
metadata:
  name: reviews-authz
  namespace: bookinfo
spec:
  selector:
    matchLabels:
      app: reviews
  action: DENY
  rules:
  - {}
```

◀ **Scoped to namespace**

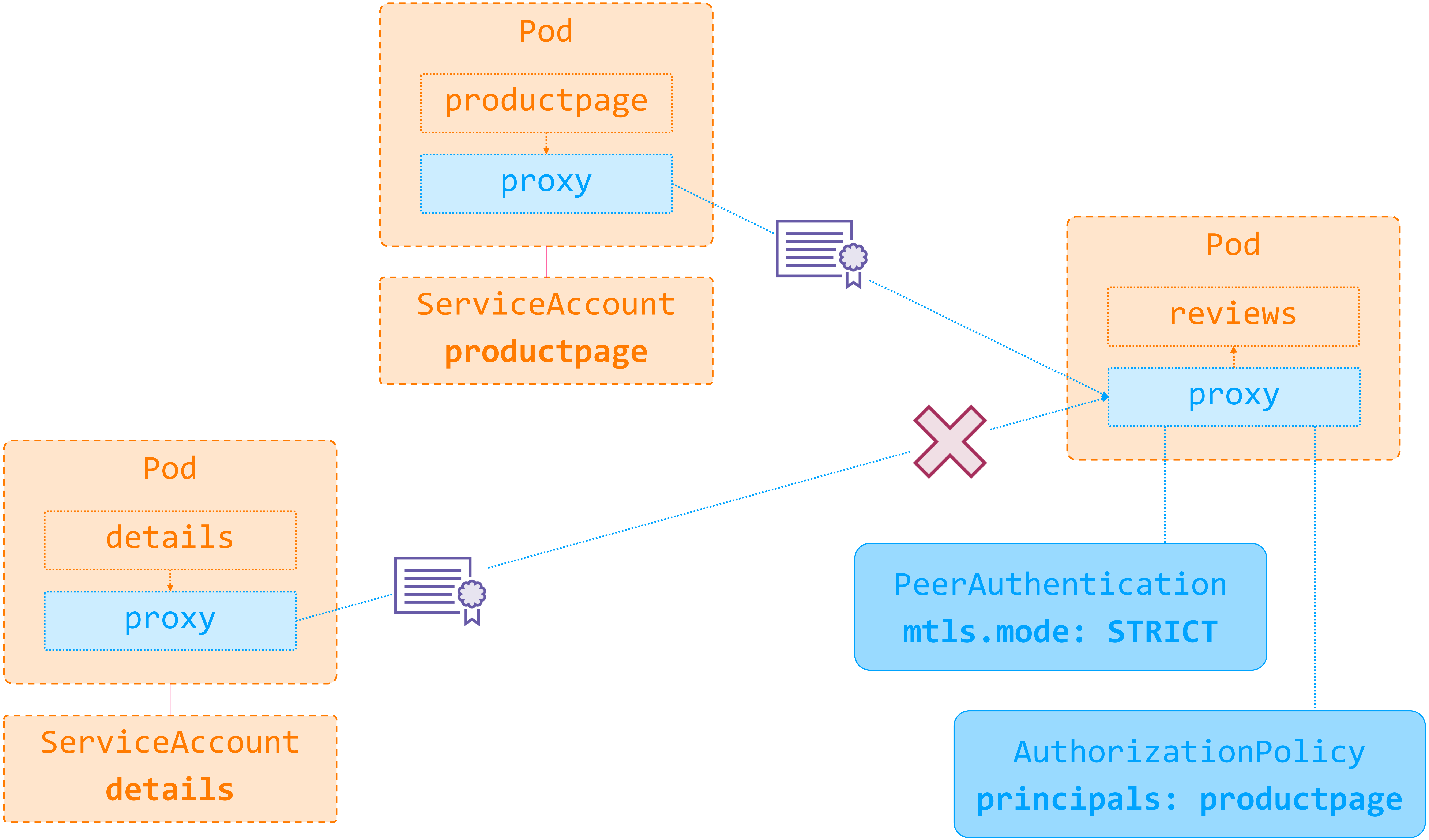
◀ **Scoped to component(s) within namespace**

◀ **Deny access**

◀ **No rules means deny all**

```
...
spec:
 selector:
  matchLabels:
   app: reviews
 action: ALLOW
 rules:
 - from:
  - source:
    principals:
    - .../sa/productpage
  to:
  - operation:
    methods: ["GET"]
```

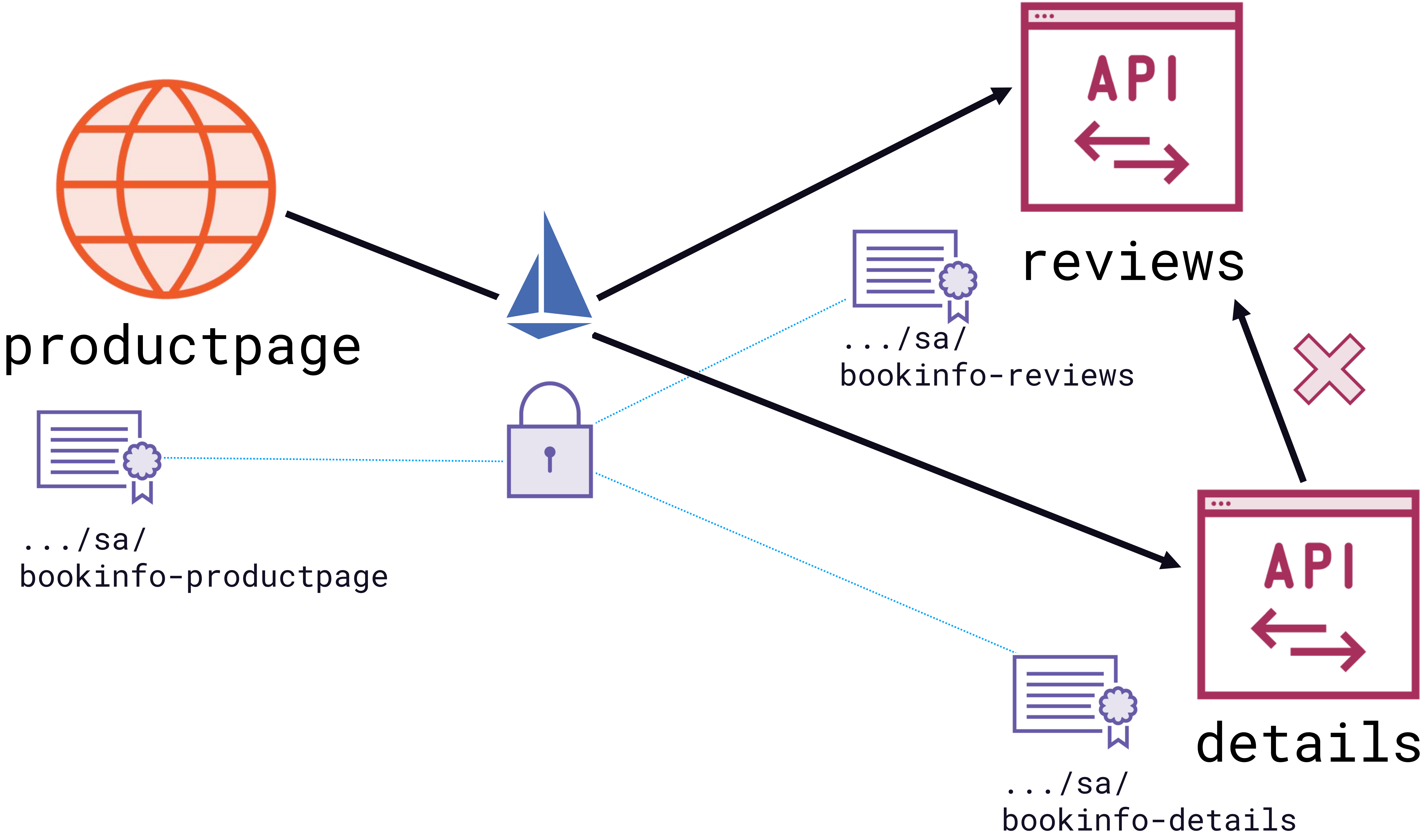
- **Rules specify allowed permissions**
- **Allow requests from proxy principal**
- **To make GET HTTP requests**


Pod
productpage
proxy
ServiceAccount
productpage
Pod
reviews
proxy
Pod
details
proxy
ServiceAccount
details
PeerAuthentication
mtls.mode: STRICT
AuthorizationPolicy
principals: productpage

Demo

Service authorization with mTLS

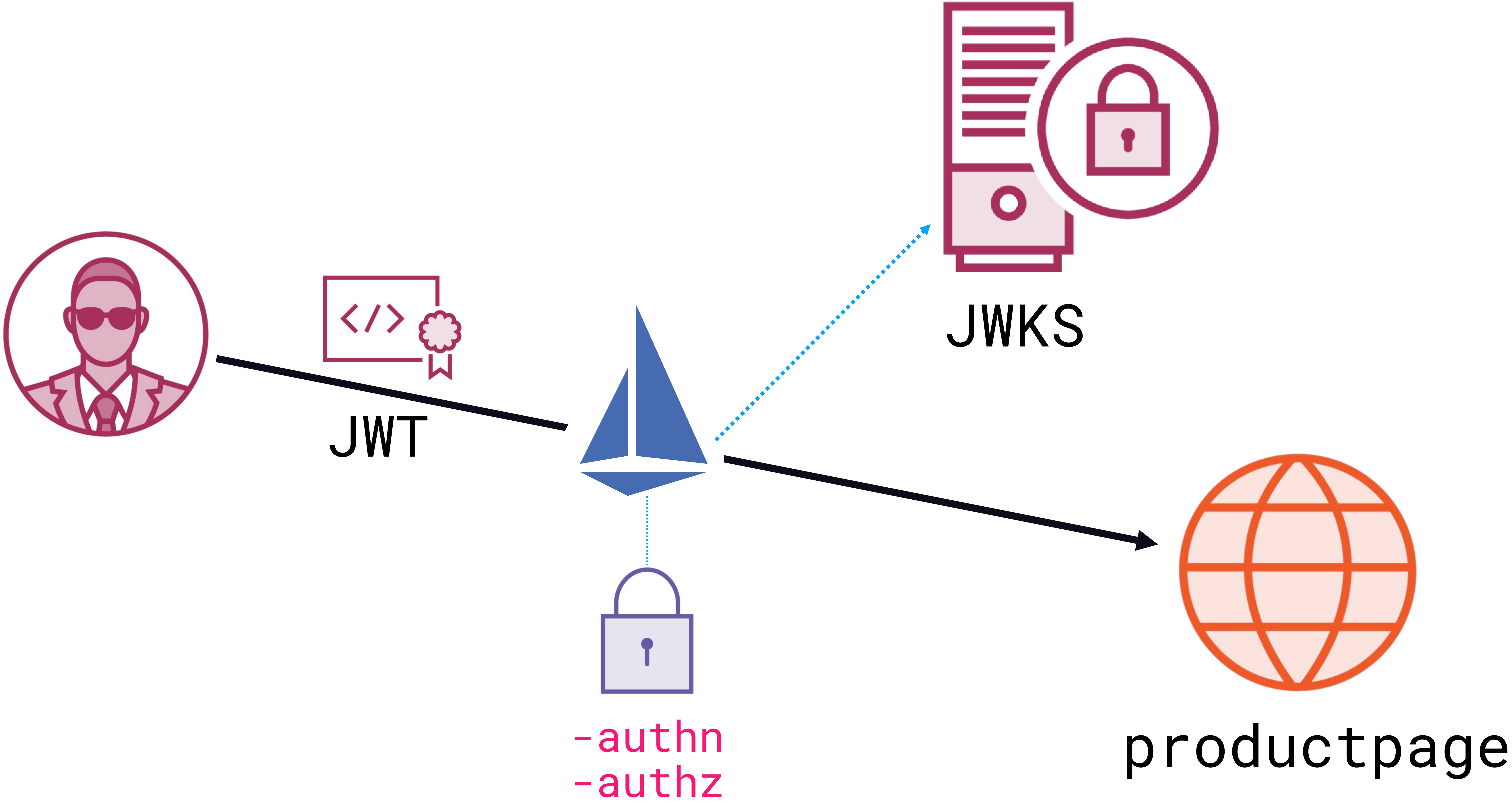
- Authorization with deny-all
- Allowing named principals
- Validate authentication identity

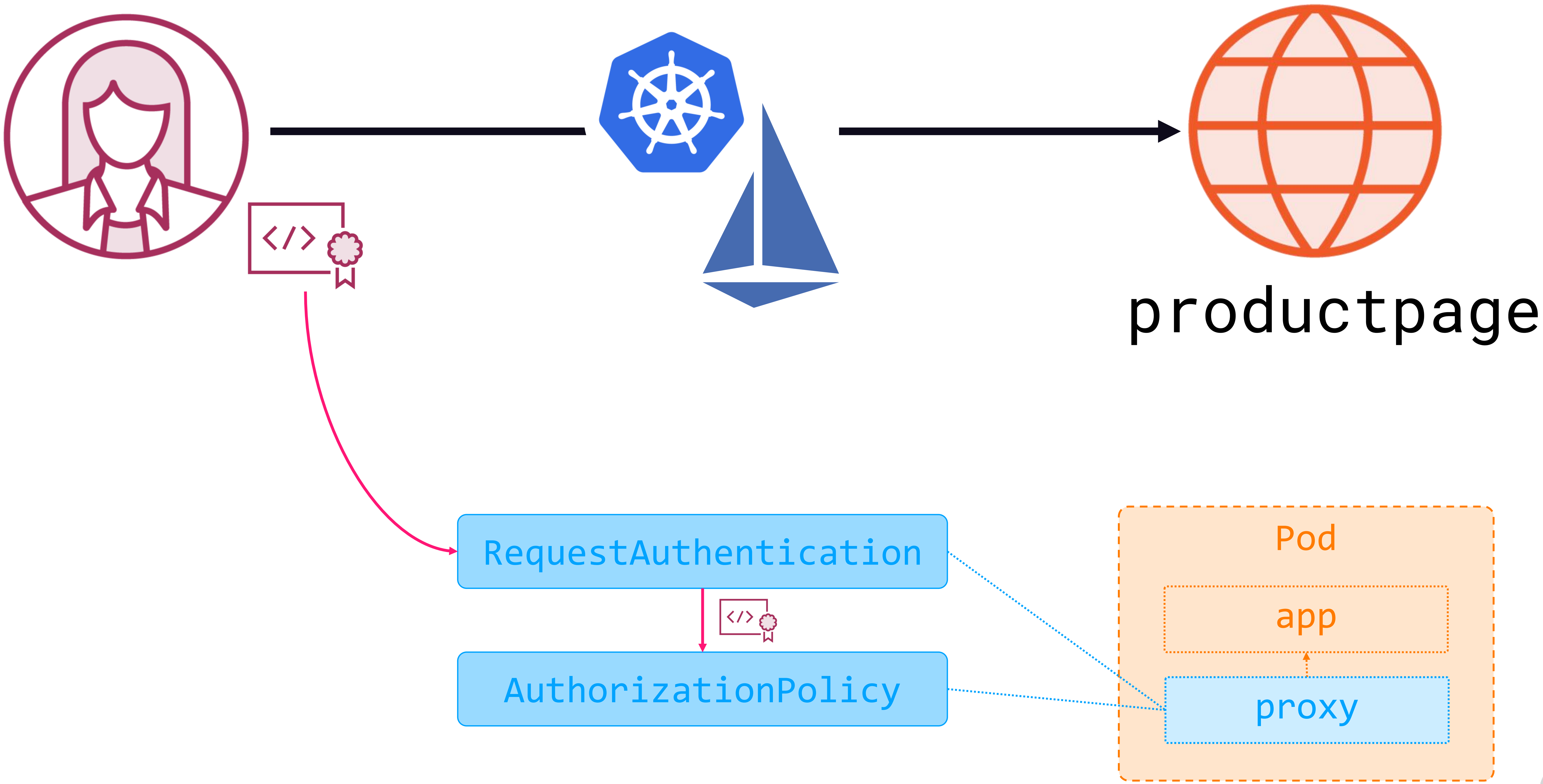

API
reviews
.../sa/
bookinfo-reviews
productpage
.../sa/
bookinfo-productpage
API
details
.../sa/
bookinfo-details

Authorization with Service Principals

AuthorizationPolicy.yaml

```yaml
apiVersion: security.istio.io/v1
kind: AuthorizationPolicy
metadata:
  name: reviews-authz
  namespace: bookinfo
spec:
  selector:
    matchLabels:
      app: reviews
  action: ALLOW
  rules:
  - from:
    - source:
        principals: ["cluster.local/ns/bookinfo/sa/bookinfo-productpage"]
    to:
    - operation:
        methods: ["GET"]
```


JWT
JWKS
productpage
-authn
-authz



productpage

RequestAuthentication

AuthorizationPolicy

Pod

app

proxy

```yaml
apiVersion: security.istio.io/v1
kind: RequestAuthentication
metadata:
  name: productpage-authn
  namespace: bookinfo
spec:
  selector:
    matchLabels:
      app: productpage
  jwtRules:
  - issuer: https://sts.windows.net/...
    jwksUri: https://login.microsoft.../keys
```

◂ **End-user authentication**

◂ **Target selector for service(s)**

◂ **Require JWT**

◂ **Issuer and JWKS server address for validation**

```yaml
apiVersion: security.istio.io/v1
kind: AuthorizationPolicy
metadata:
  name: productpage-authz
  namespace: bookinfo
spec:
  selector:
    matchLabels:
      app: productpage
  action: ALLOW
  rules:
  - from:
    - source:
        principals: [".../sa/istio-ingress"]
        requestPrincipals: ["*"]
    to:
    - operation:
        methods: ["GET"]
```

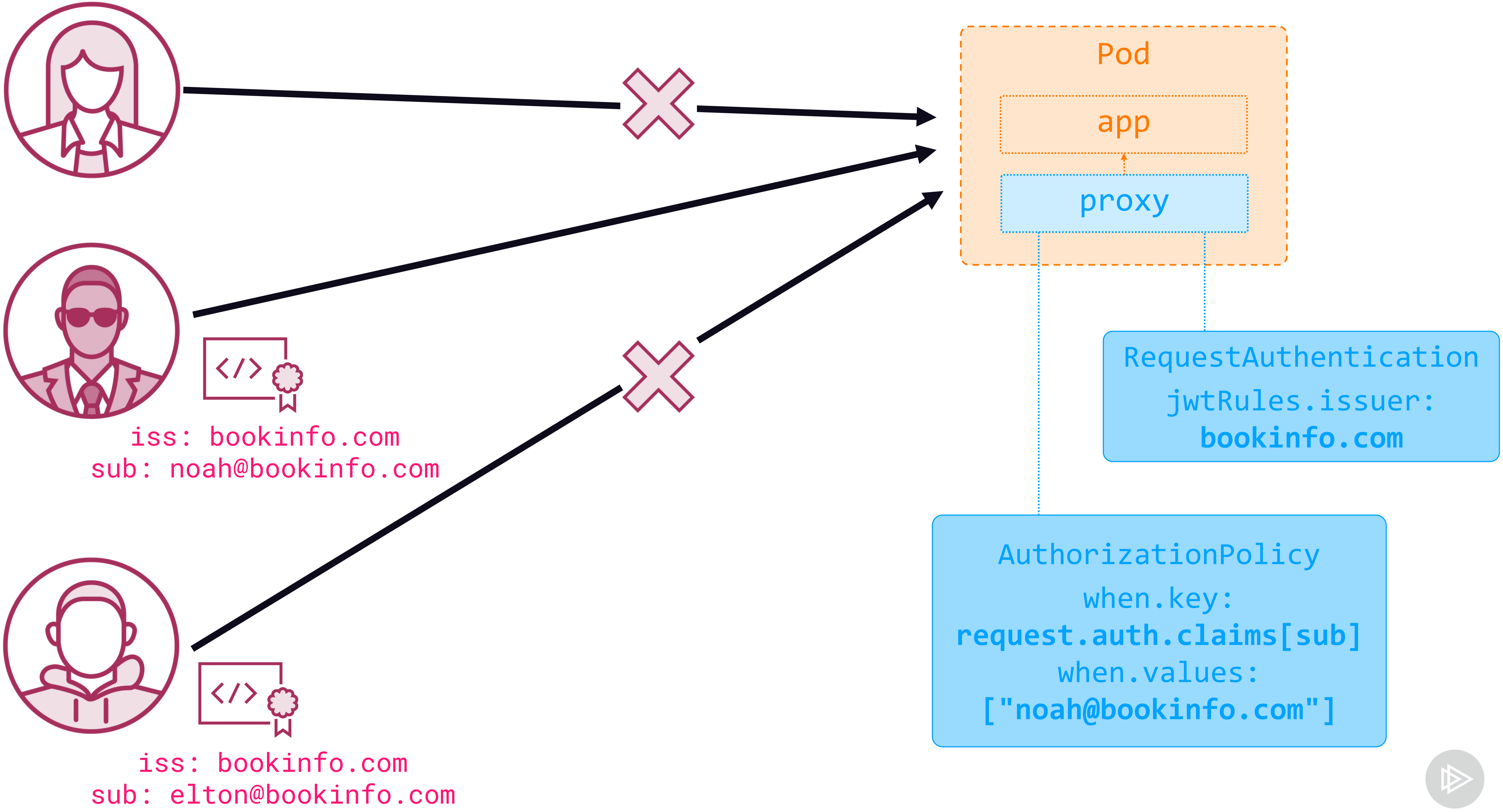
◀ **Service access**

◀ **Allow access to gateway Pod(s)**

◀ **And authenticated end-users**

```yaml
...
spec:
 selector:
  matchLabels:
   app: productpage
 action: ALLOW
 rules:
 - from:
  - source:
    principals: [".../sa/istio-ingress"]
    requestPrincipals: ["*"]
  to:
  - operation:
    methods: ["GET"]
  when:
  - key: request.auth.claims[sub]
   values: ["noah@bookinfo.com"]
```

◀ **Claims-based auth for end-users**



Pod

app

proxy

iss: bookinfo.com
sub: noah@bookinfo.com

iss: bookinfo.com
sub: elton@bookinfo.com

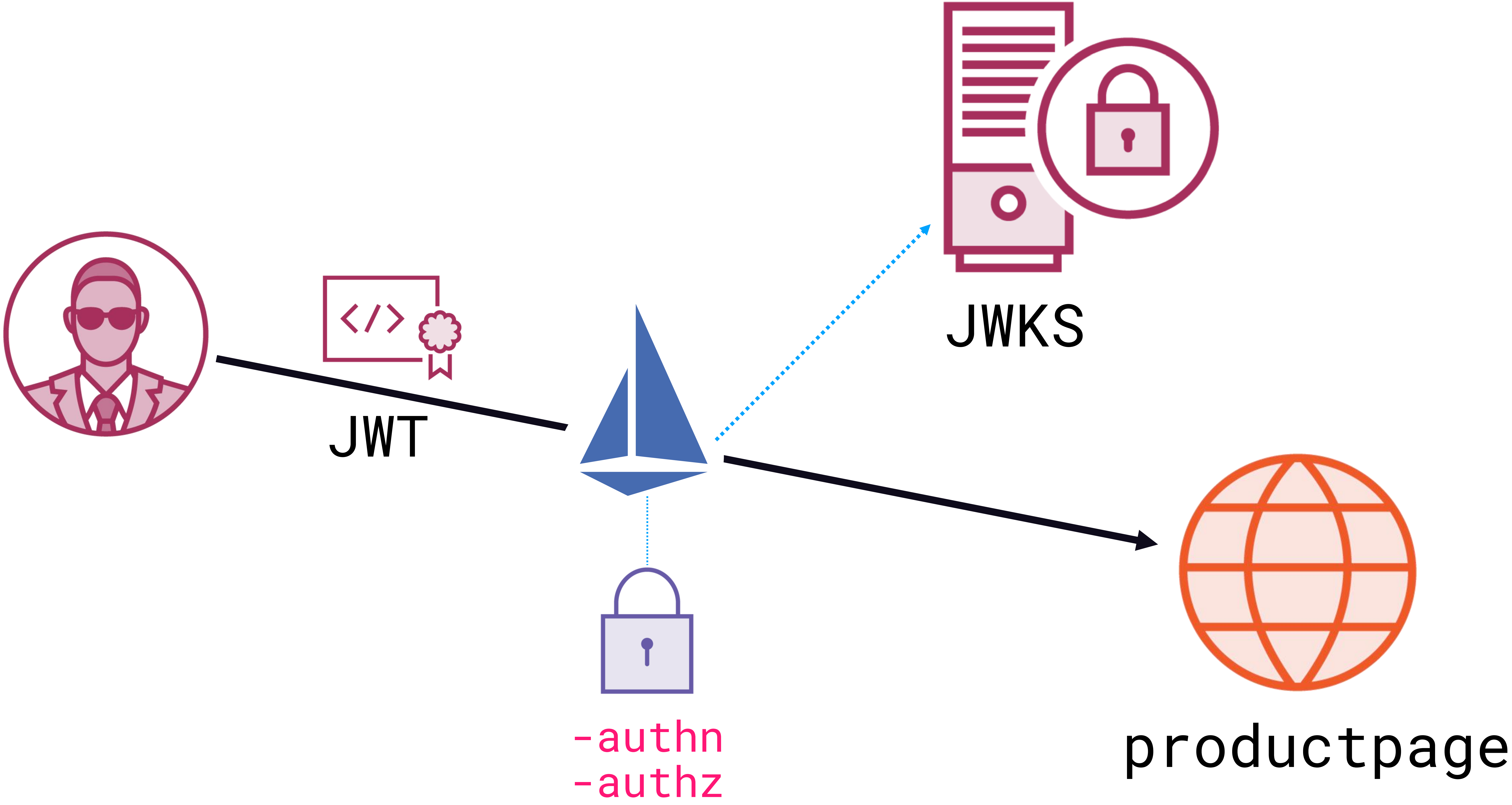
RequestAuthentication
jwtRules.issuer:
bookinfo.com

AuthorizationPolicy
when.key:
request.auth.claims[sub]
when.values:
["noah@bookinfo.com"]

Demo

End-user authorization with JWT

- Requiring JWT authentication
- Denying access to all
- Authorizing access by JWT claim


JWKS
JWT
-authn
-authz
productpage

Authentication with JWT

RequestAuthentication.yaml

```yaml
apiVersion: security.istio.io/v1
kind: RequestAuthentication
metadata:
  name: productpage-authn
  namespace: bookinfo
spec:
  selector:
    matchLabels:
      app: productpage
  jwtRules:
  - issuer: testing@secure.istio.io
    jwksUri: https://.../jwks.json
```

Authentication with JWT

RequestAuthentication-Azure.yaml

```yaml
apiVersion: security.istio.io/v1
kind: RequestAuthentication
metadata:
  name: azure-authn
  namespace: appx
spec:
  selector:
    matchLabels:
      authn: jwt
  jwtRules:
  - issuer: https://sts.windows.net/[tenantId]/
    jwksUri:
https://login.microsoftonline.com/common/discovery/v2.0/keys
```

Authorization On JWT Claims

AuthorizationPolicy.yaml

```yaml
apiVersion: security.istio.io/v1
kind: AuthorizationPolicy
metadata:
  name: productpage-authz
  namespace: bookinfo
spec:
  selector:
    matchLabels:
      app: productpage
  action: ALLOW
  rules:
  - from:
    - source:
        principals: ["cluster.../sa/istio-ingress"]
        requestPrincipals: ["testing@secure.istio.io/*"]
    to:
    - operation:
        methods: ["GET"]
    when:
    - key: request.auth.claims[foo]
      values: ["bar"]
```

Summary

Securing peer communication

- Mutual TLS
- Istio-managed certs
- Secure identity

Istio resources

- PeerAuthentication
- RequestAuthentication
- AuthorizationPolicy

Securing end-user access

- Require JWT
- Authorize on claims

Up Next:

Observing the Service Network