

Securing Communication with Mutual TLS

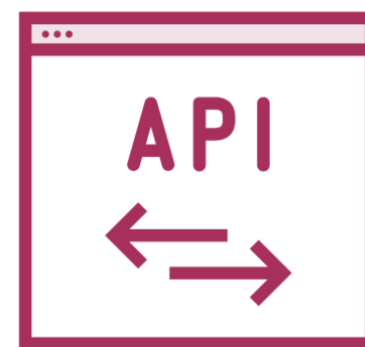


Kien Bui

DevOps & Platform Engineer



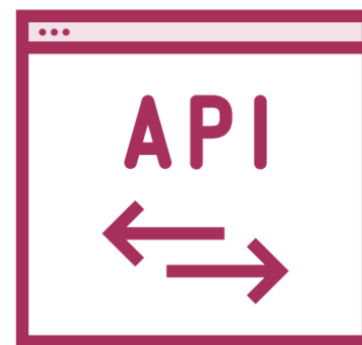
productpage



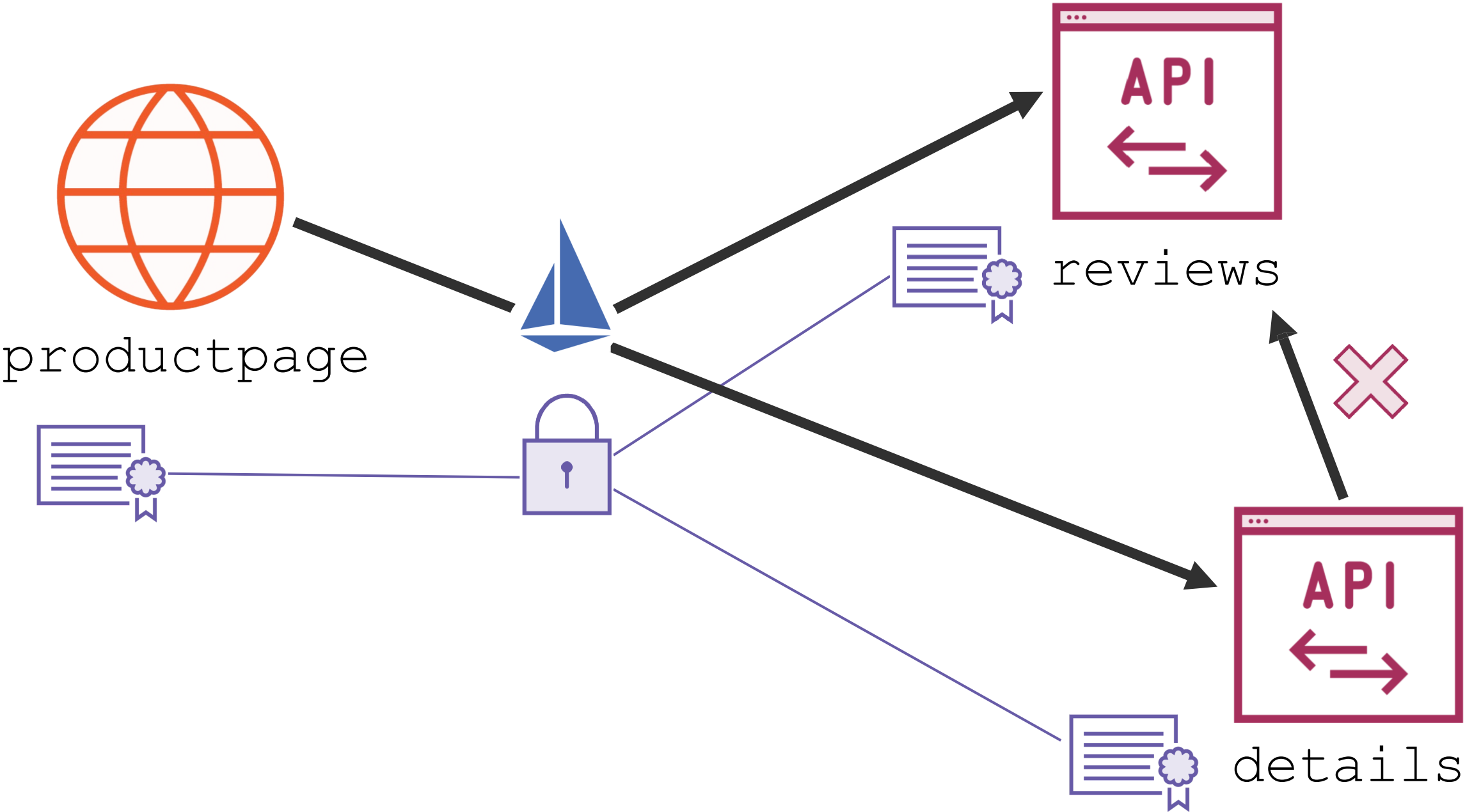
details

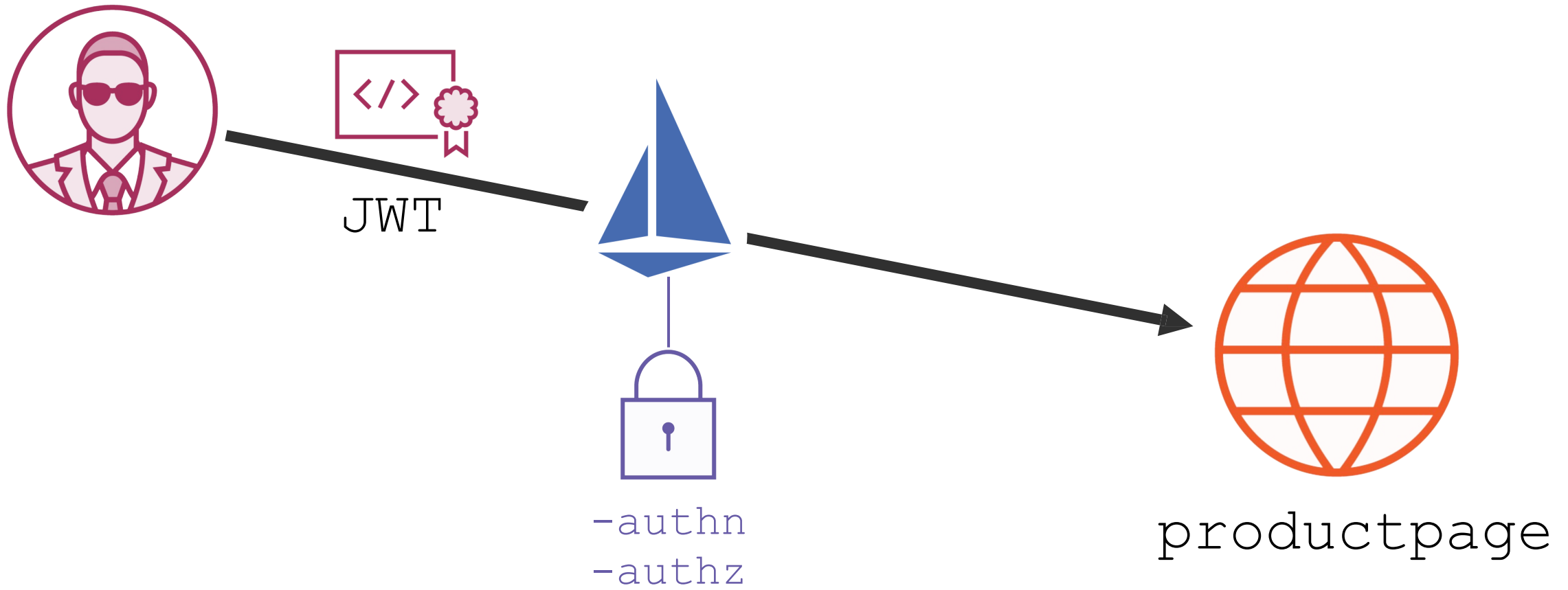


productpage



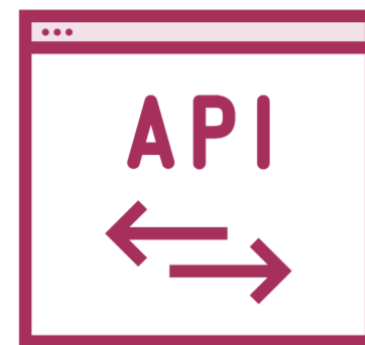
details







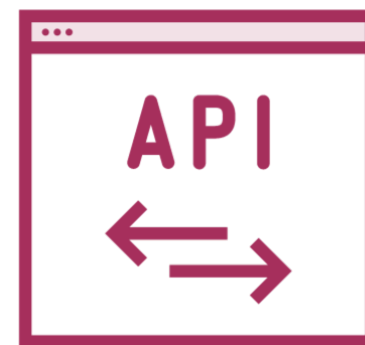
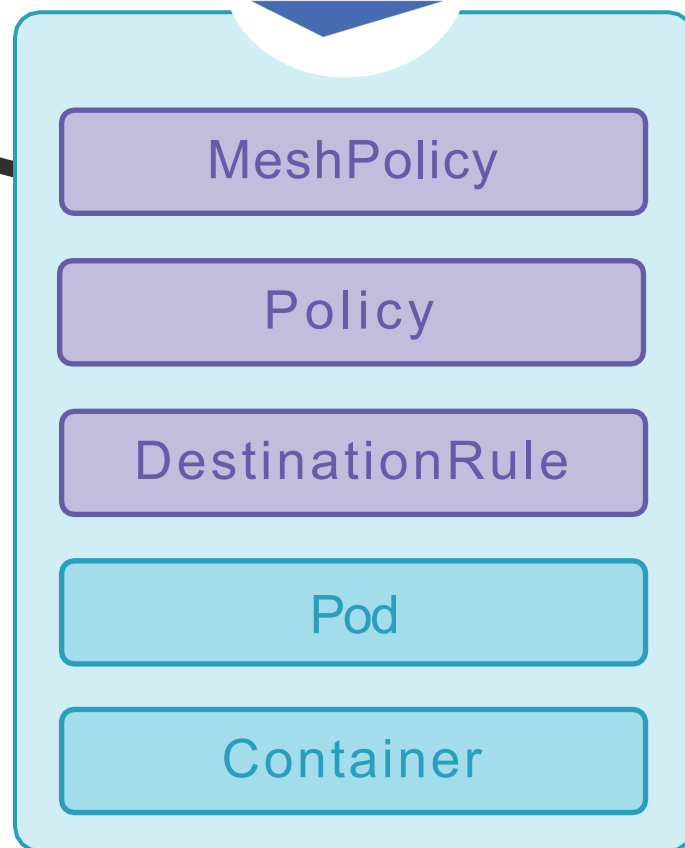
productpage



details



productpage



details

```
apiVersion: authentication.istio.io...
```

```
kind: MeshPolicy
```

```
metadata:
```

```
  name: default
```

```
spec:
```

```
  peers:
```

```
    - mtls: {}
```

t **Default policy for all namespaces**

t **Required**

t **Authentication for calling service**

t **Require mutual TLS**


```
apiVersion: authentication.istio.io...
```

```
kind: MeshPolicy
```

```
metadata:
```

```
  name: default
```

```
spec:
```

```
  peers:
```

```
    - mtls:
```

```
      mode: PERMISSIVE
```

t **Default policy for all namespaces**

t **Required**

t **Authentication for calling service**

t **Support mutual TLS**

```
apiVersion: authentication.istio.io...
```

```
kind: Policy
```

```
metadata:
```

```
  name: default
```

```
  namespace: default
```

```
spec:
```

```
  peers:
```

```
    - mtls: {}
```

t **Policy for specified targets**

t **Default for the** default **namespace**

t **Require mutual TLS**

```
apiVersion: authentication.istio.io...
```

```
kind: Policy
```

```
metadata:
```

```
  name: default
```

```
  namespace: default
```

```
spec:
```

```
  peers:
```

```
    -mtls: {}
```

```
  targets:
```

```
    - name: productpage
```

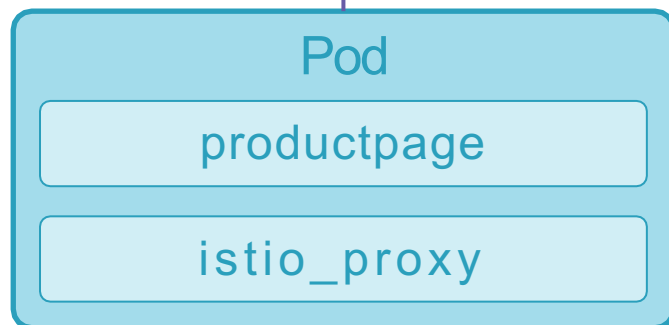
t **Policy for specified targets**

t **Default for the** default **namespace**

t **Require mutual TLS**

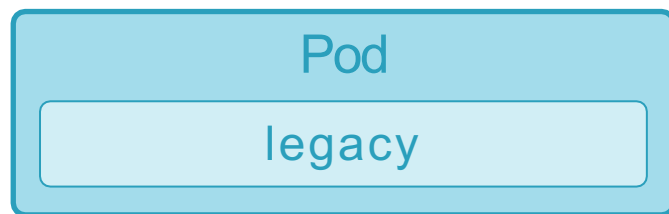
t **Apply to the** productpage **service**

kind: DestinationRule
...
trafficPolicy:
 tls:
 mode: ISTIO_MUTUAL



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

GET https://...

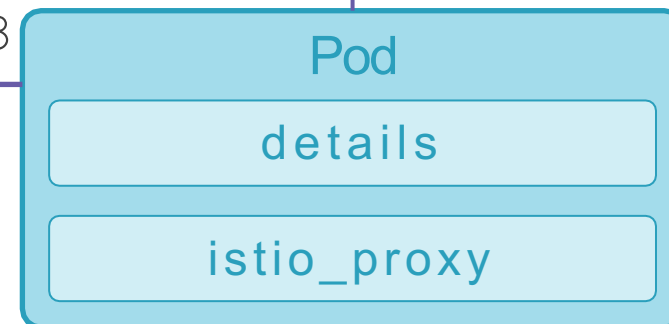


GET http://...



:443

kind: Policy
...
peers:
 - mtls: {}




```
apiVersion: networking.istio.io...  
kind: DestinationRule  
metadata:  
  name: default  
  namespace: default  
spec:  
  host: "*.default.svc.cluster.local"  
  trafficPolicy:  
    tls:  
      mode: ISTIO_MUTUAL
```

t **Default for the** default **namespace**

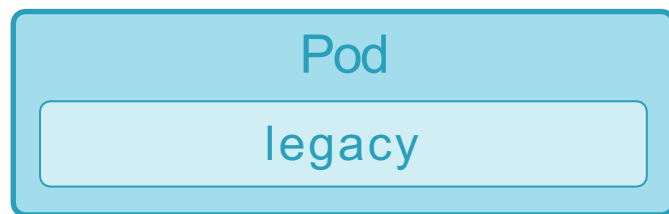
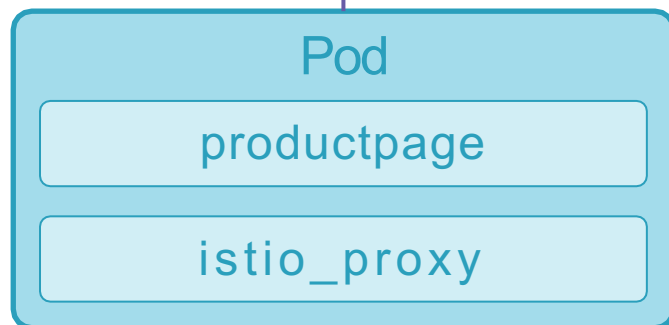
t **Any host in the same namespace**

t **Use mutual TLS**

t **With Istio-managed client certs**



```
kind: DestinationRule
...
trafficPolicy:
  tls:
    mode: ISTIO_MUTUAL
```




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

GET https://...

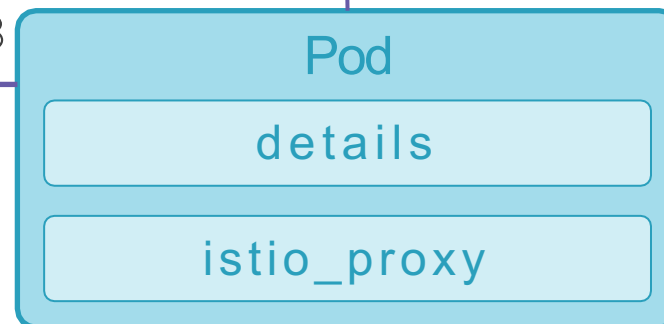


:443

:80



```
kind: Policy
...
peers:
  - mtls:
      mode: PERMISSIVE
```



Demo

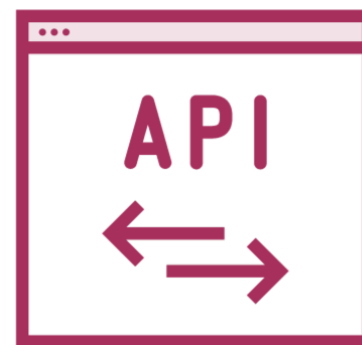


Securing Services with Mutual TLS

- Hack services over HTTP
- Apply mTLS policy
- Validate HTTPS setup



productpage



details

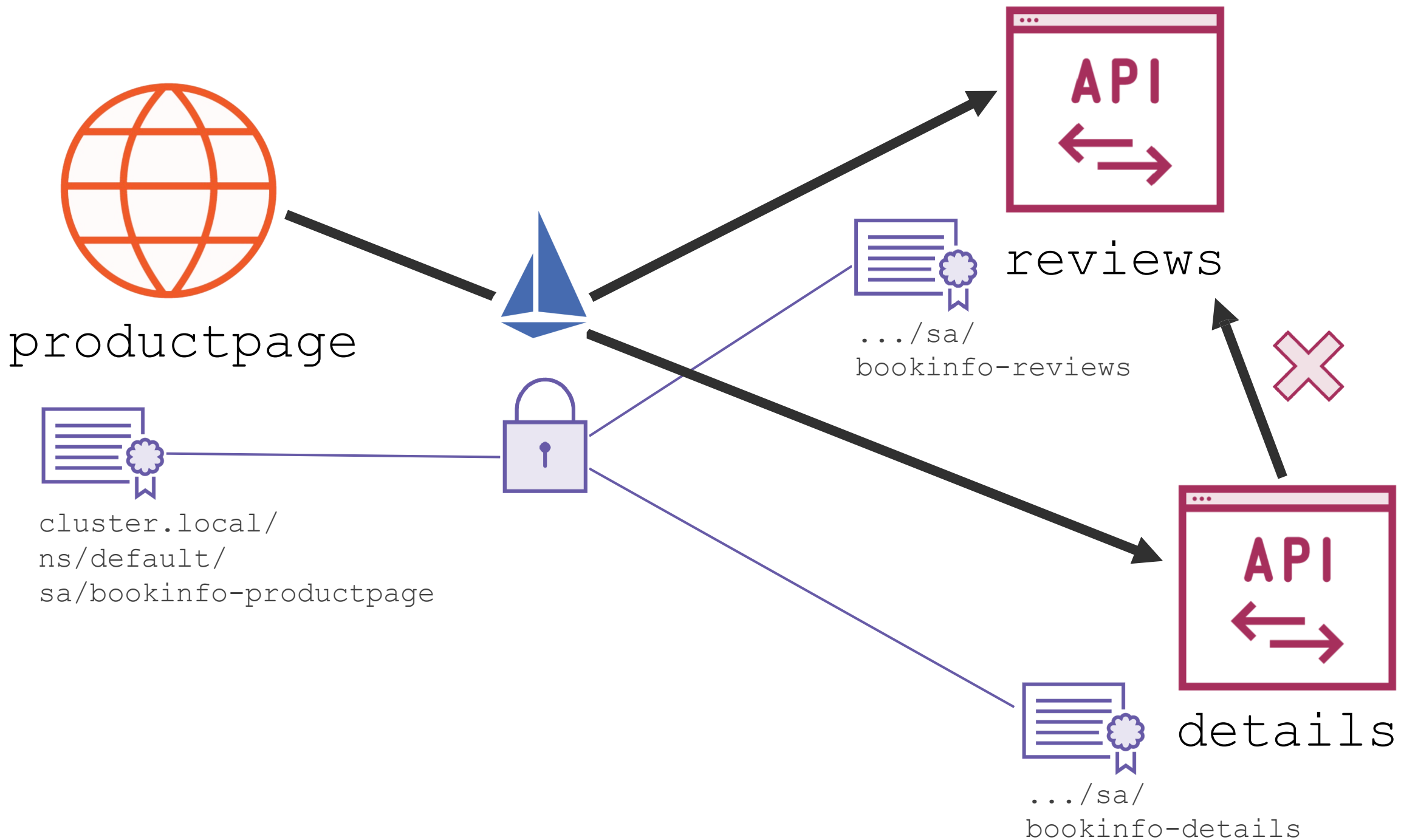
Enforcing Mutual TLS

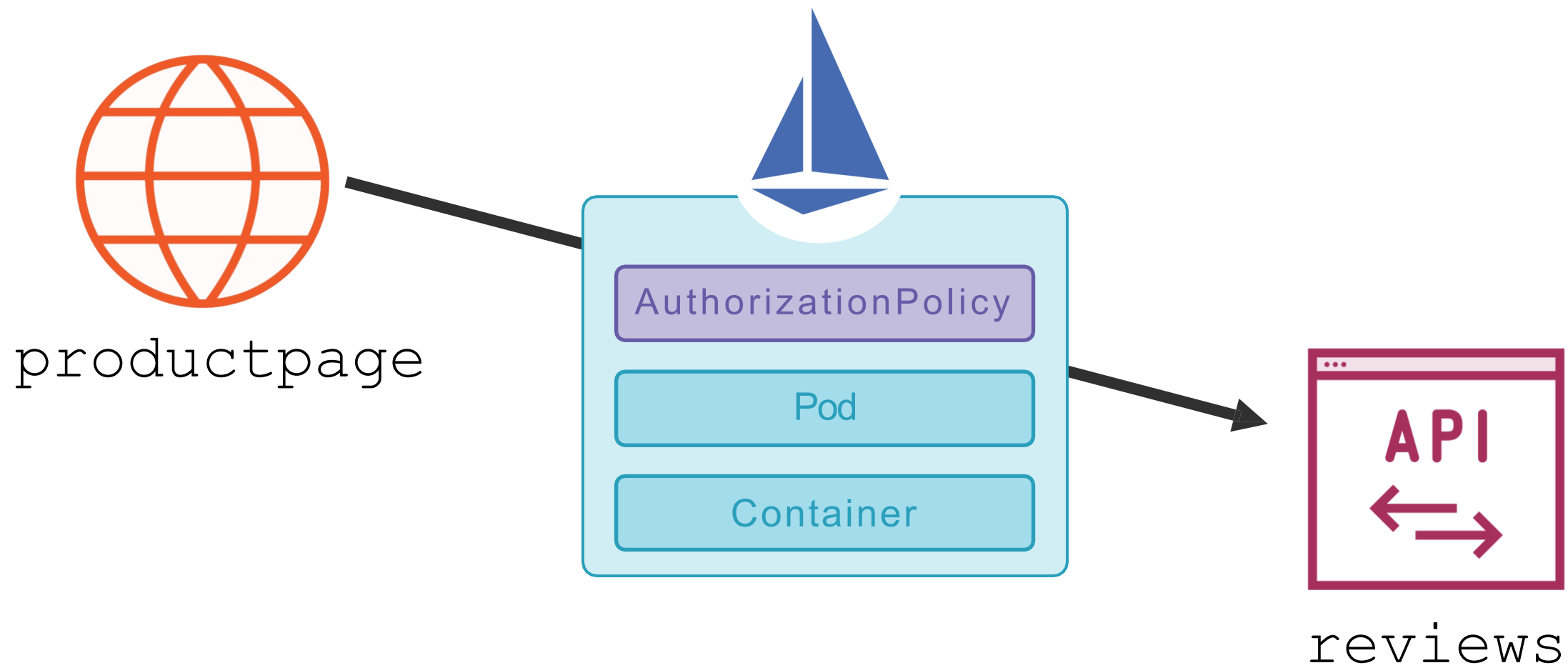
Policy.yaml

```
kind: Policy
metadata:
  name: default
  namespace: default
spec:
  peers:
  - mtls: {}
```

DestinationRule.yaml

```
kind: DestinationRule
metadata:
  name: default
  namespace: default
spec:
  host: "*.default.svc.cluster.local"
  trafficPolicy:
    tls:
      mode: ISTIO_MUTUAL
```





```
apiVersion: security.istio.io/v1beta1
```

```
kind: AuthorizationPolicy
```

```
metadata:
```

```
  name: reviews-authz
```

```
  namespace: default
```

```
spec:
```

```
  selector:
```

```
    matchLabels:
```

```
      app: reviews
```

t **Apply to** default **namespace**

t **Target selector identifies service(s)**

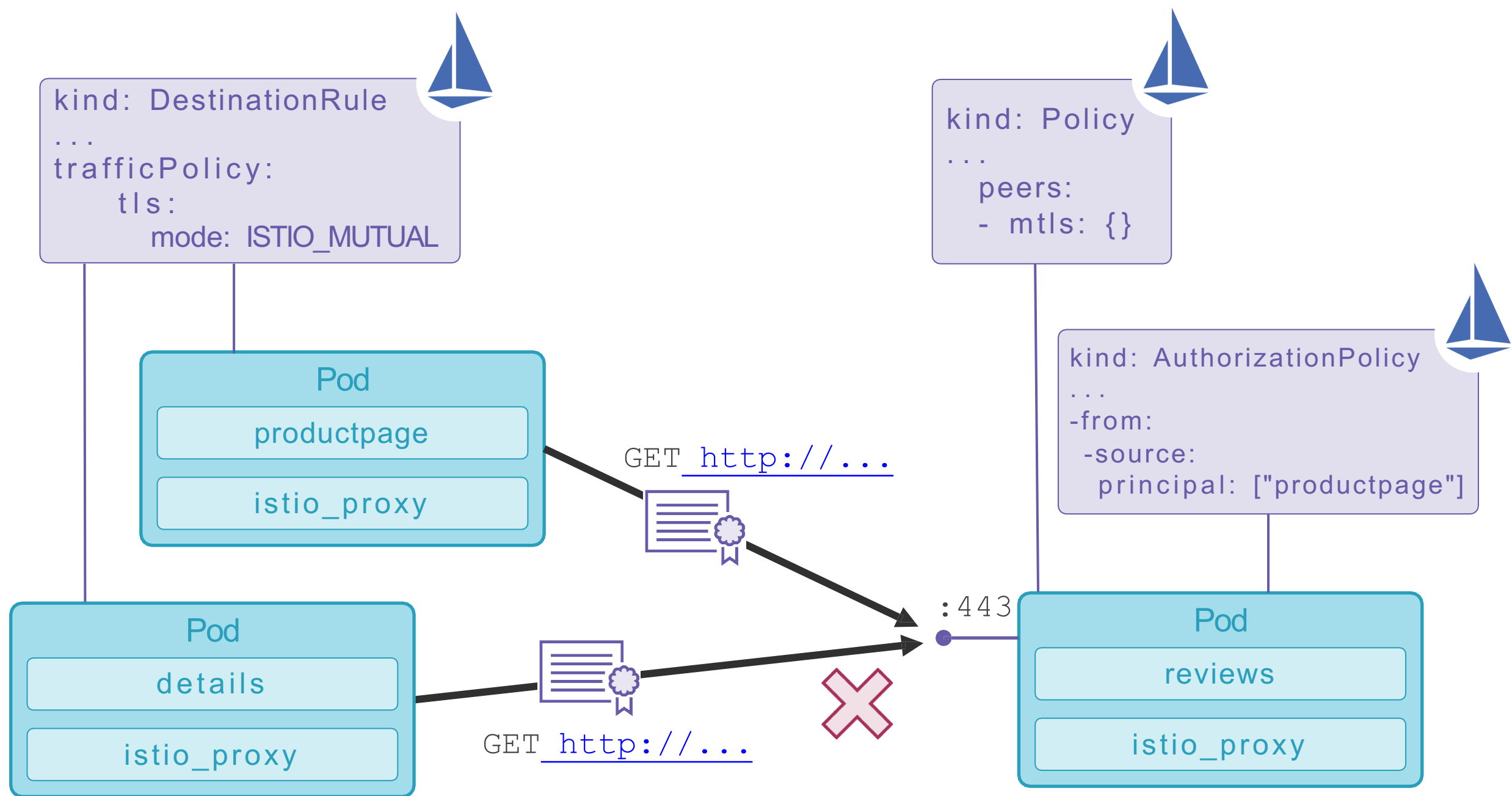
t **Default deny; no rules means deny all**

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

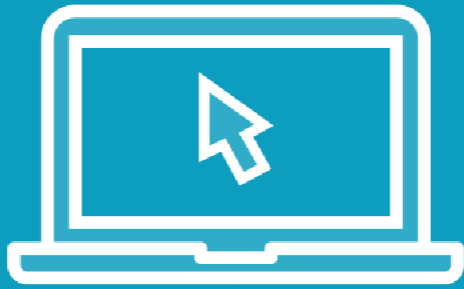
t **Rules specify allowed permissions**

t **Allow requests from specified principal**

t **To make GET calls to the service**

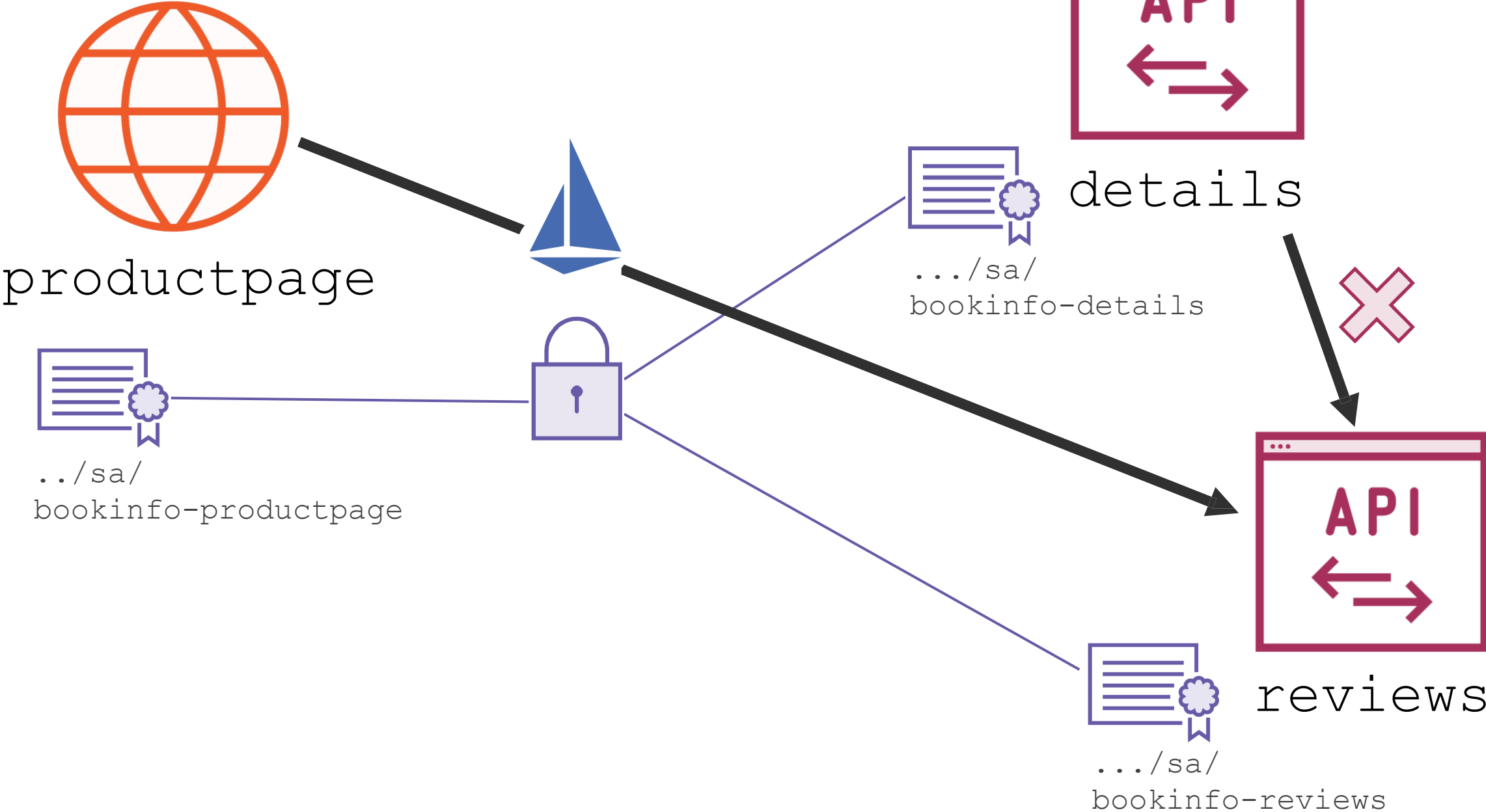


Demo



Service Authorization with mTLS

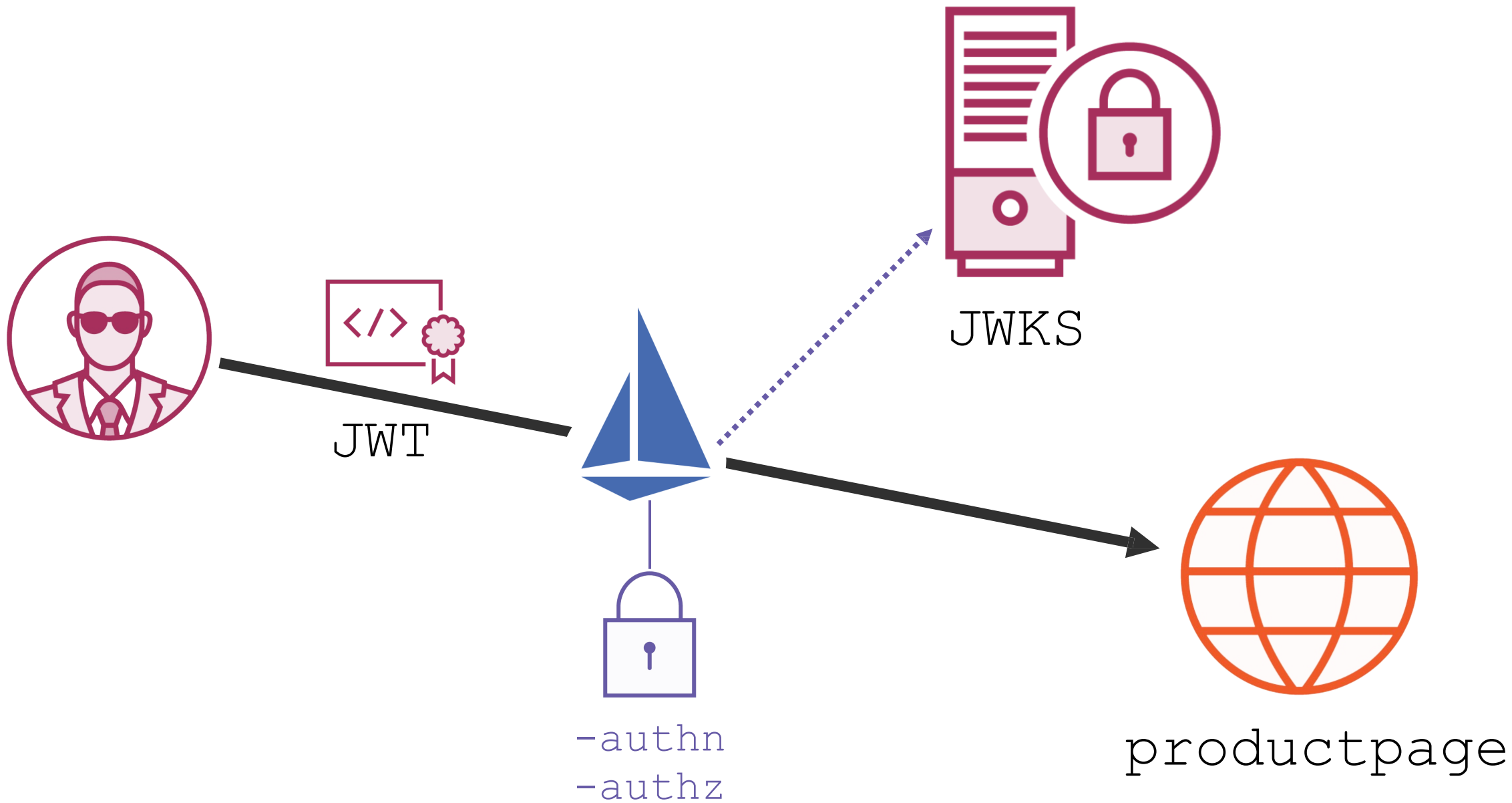
- Authorization with deny-all
- Allowing named principals
- Validate authentication identity

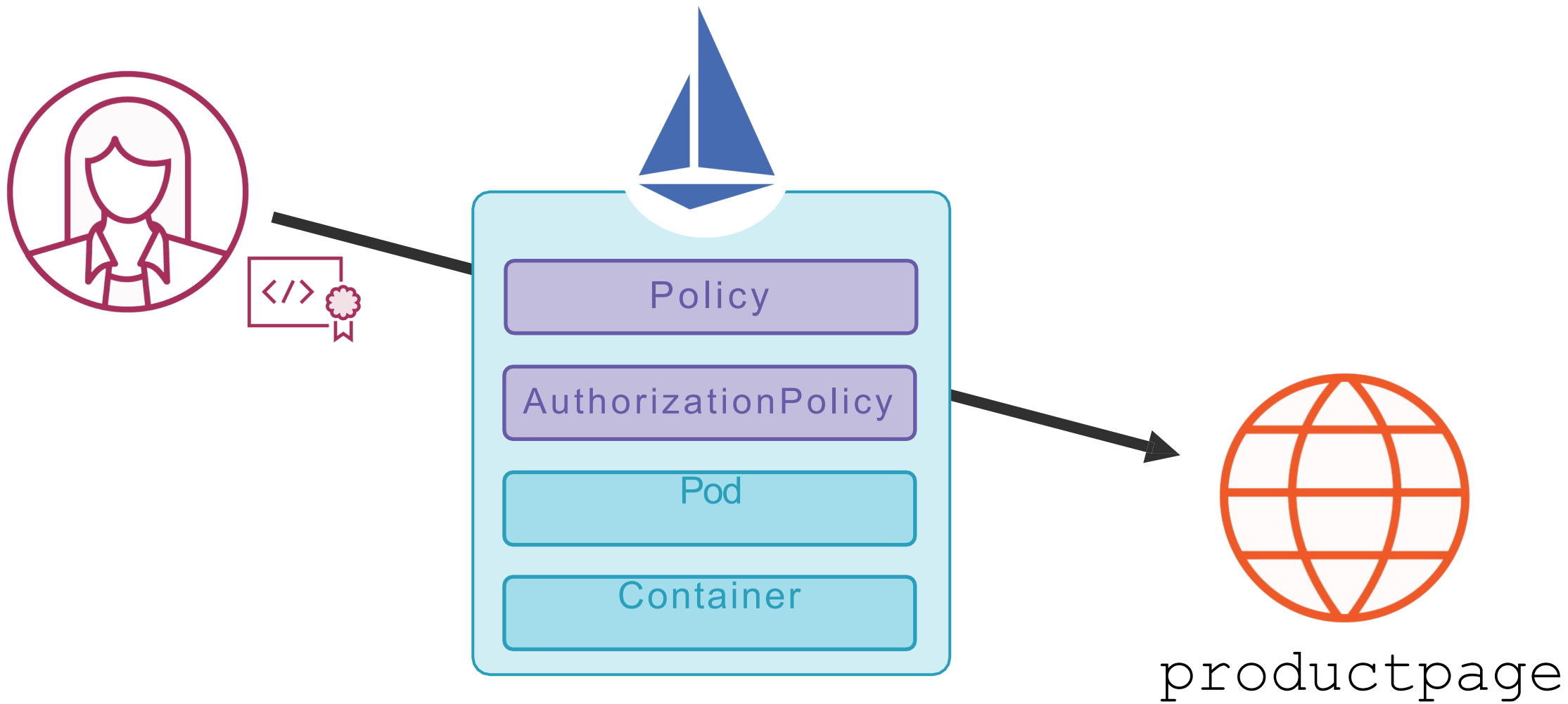


Authorization with Service Principals

AuthorizationPolicy.yaml

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: reviews-authz
  namespace: default
spec:
  selector:
    matchLabels:
      app: reviews
  rules:
    - from:
        - source:
            principals: ["cluster.local/ns/default/sa/bookinfo-productpage"]
      to:
        - operation:
            methods: ["GET"]
```





```
apiVersion: authentication.istio.io/...
```

```
kind: Policy
```

```
metadata:
```

```
  name: productpage-authn
```

```
spec:
```

```
  targets:
```

```
    -name: productpage
```

```
  origins:
```

```
    - jwt:
```

```
      issuer: "xyz"
```

```
      jwksUri: "uri"
```

```
  principalBinding: USE_ORIGIN
```

t **Service authentication**

t **Target selector for service(s)**

t **Authentication for end user**

t **Require JWT**

t **JWT issuer and JWKS server address**

t **Sets principal from JWT**

```
apiVersion: security.istio.io/v1beta1

kind: AuthorizationPolicy

...

spec:

  rules:

    - to:

      - operation:

          methods: ["GET"]

      when:

        - key: request.auth.claims[sub]

          values: \["elton@sixeyed.com"\]
```

t **Service access**

t **Allow access to GET methods**

t **When the JWT has the specified claim**

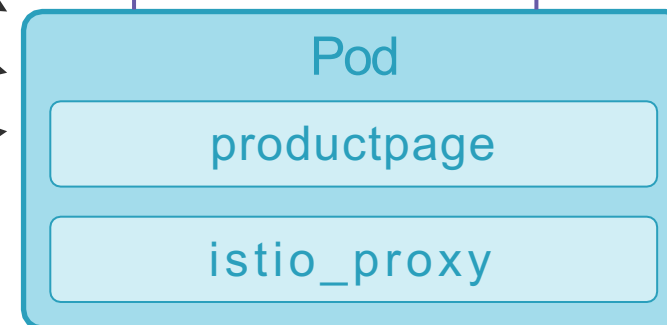


iss: sixeyed.com
sub: [noah@sixeyed.com](#)

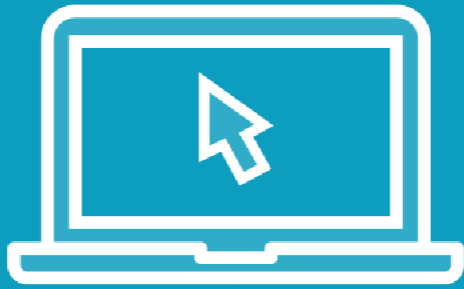
iss: sixeyed.com
sub: [elton@sixeyed.com](#)

```
kind: Policy
...
origins:
- jwt:
    issuer: "sixeyed.com"
    jwksUri: "uri"
```

```
kind: AuthorizationPolicy
...
when:
- key: request.auth.claims[sub]
  values: ["elton@sixeyed.com"]
```

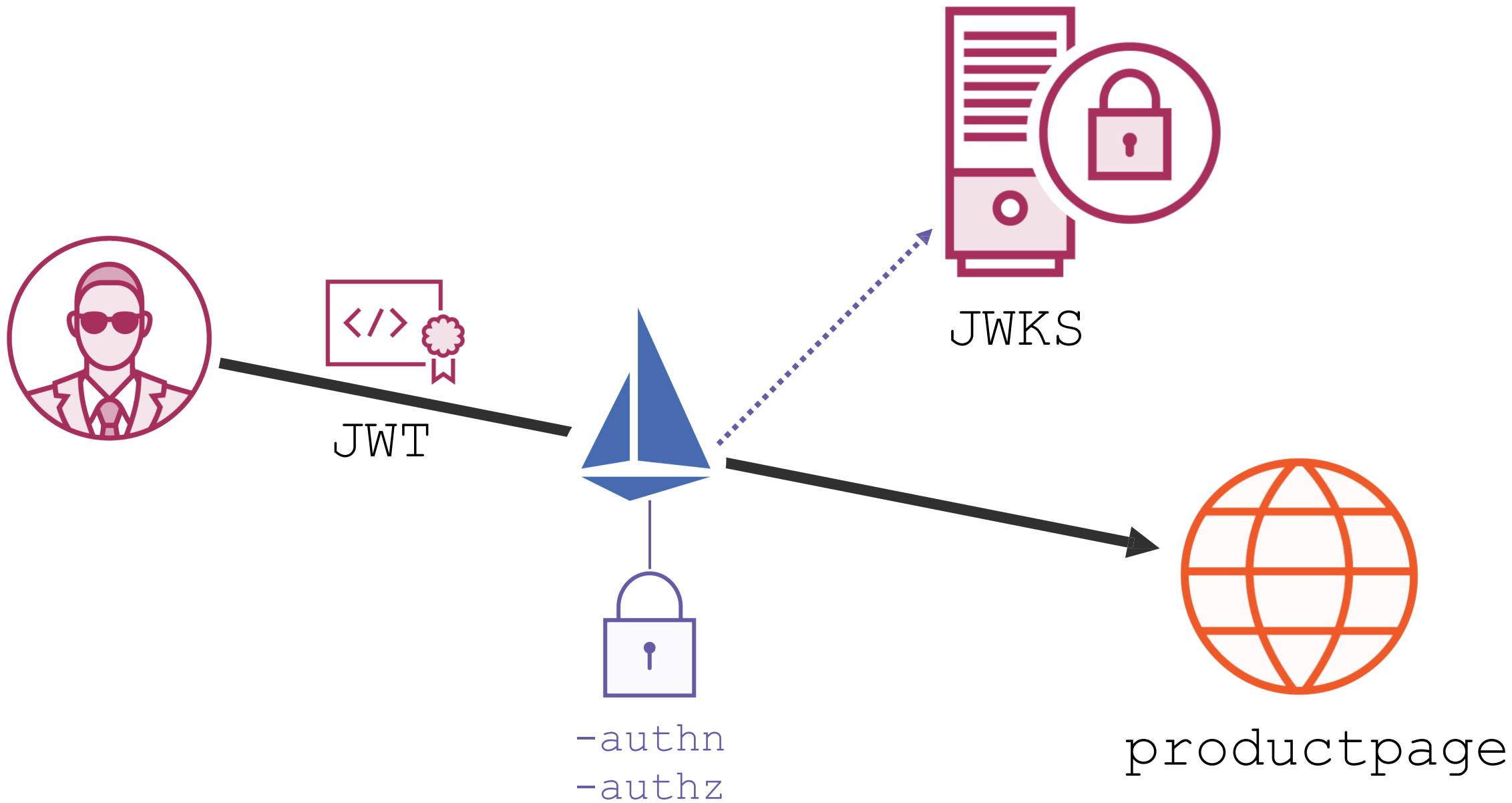


Demo



End-user Authorization with JWT

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



Authentication with JWT

Policy.yaml

```
apiVersion: authentication.istio.io/v1alpha1
kind: Policy
metadata:
  name: productpage-authn
spec:
  targets:
  - name: productpage
  peers:
  - mtls: {}
  origins:
  - jwt:
      issuer: "testing@secure.istio.io"
      jwksUri: "https://.../jwt/samples/jwks.json"
  principalBinding: USE_ORIGIN
```

Authorization On JWT Claims

AuthorizationPolicy.yaml

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: productpage-authz
  namespace: default
spec:
  selector:
    matchLabels:
      app: productpage
  rules:
    - 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

- Policy
- AuthorizationPolicy

Securing End-user Access

- Require JWT
- Authorize on claims

Up Next:

Observing the Service Network
