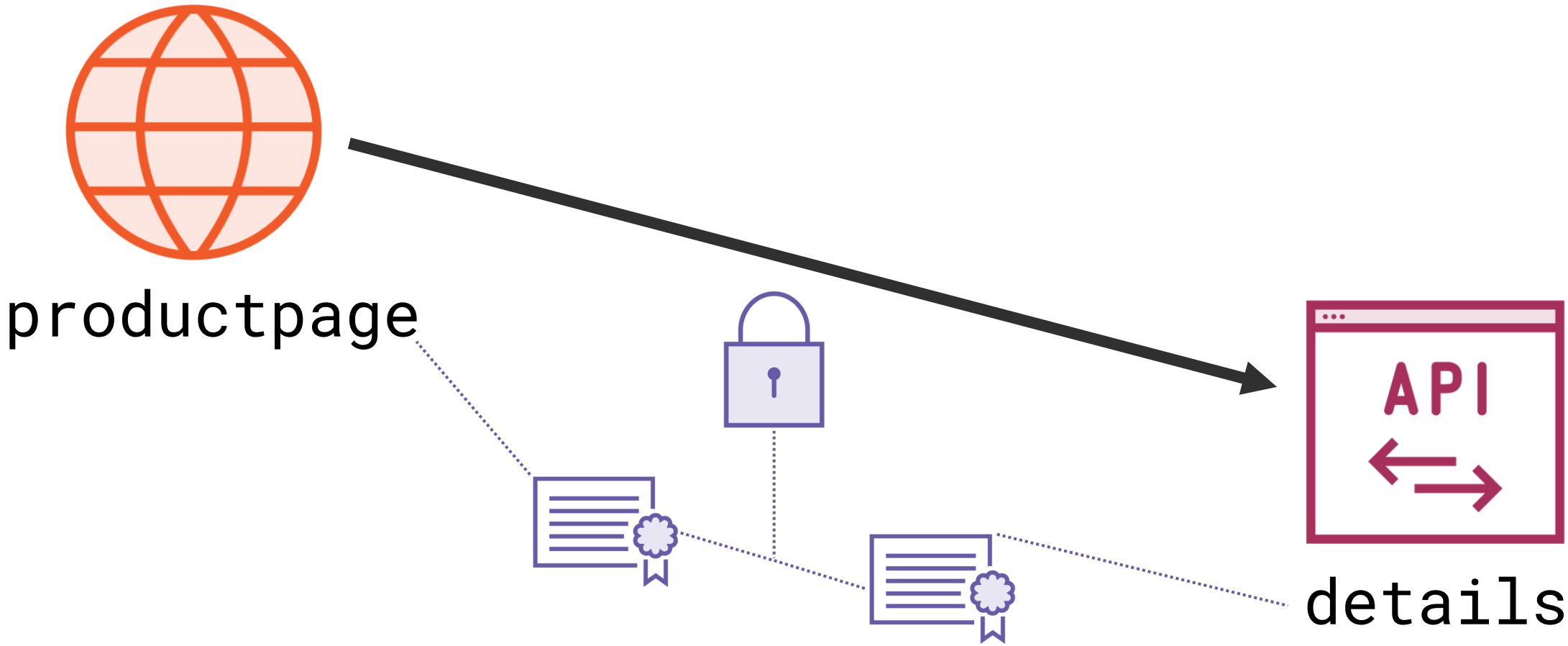
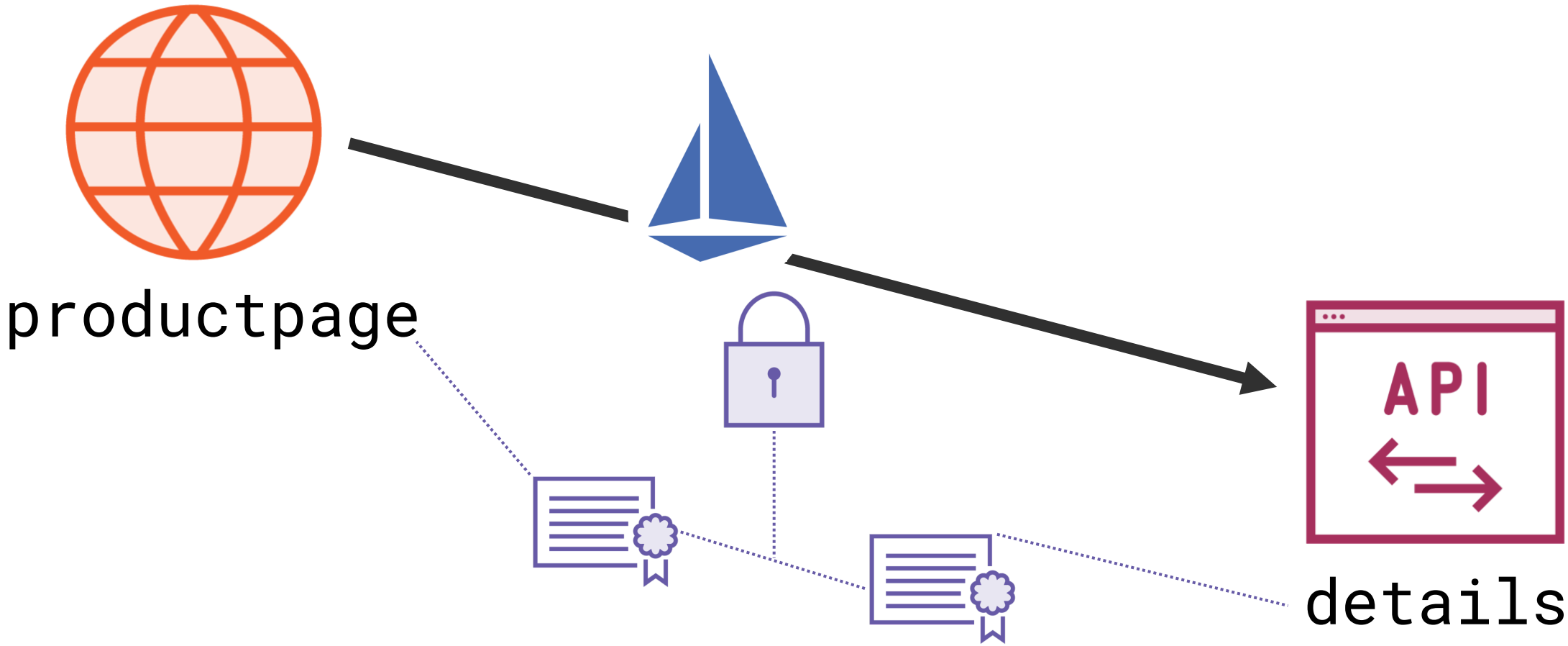
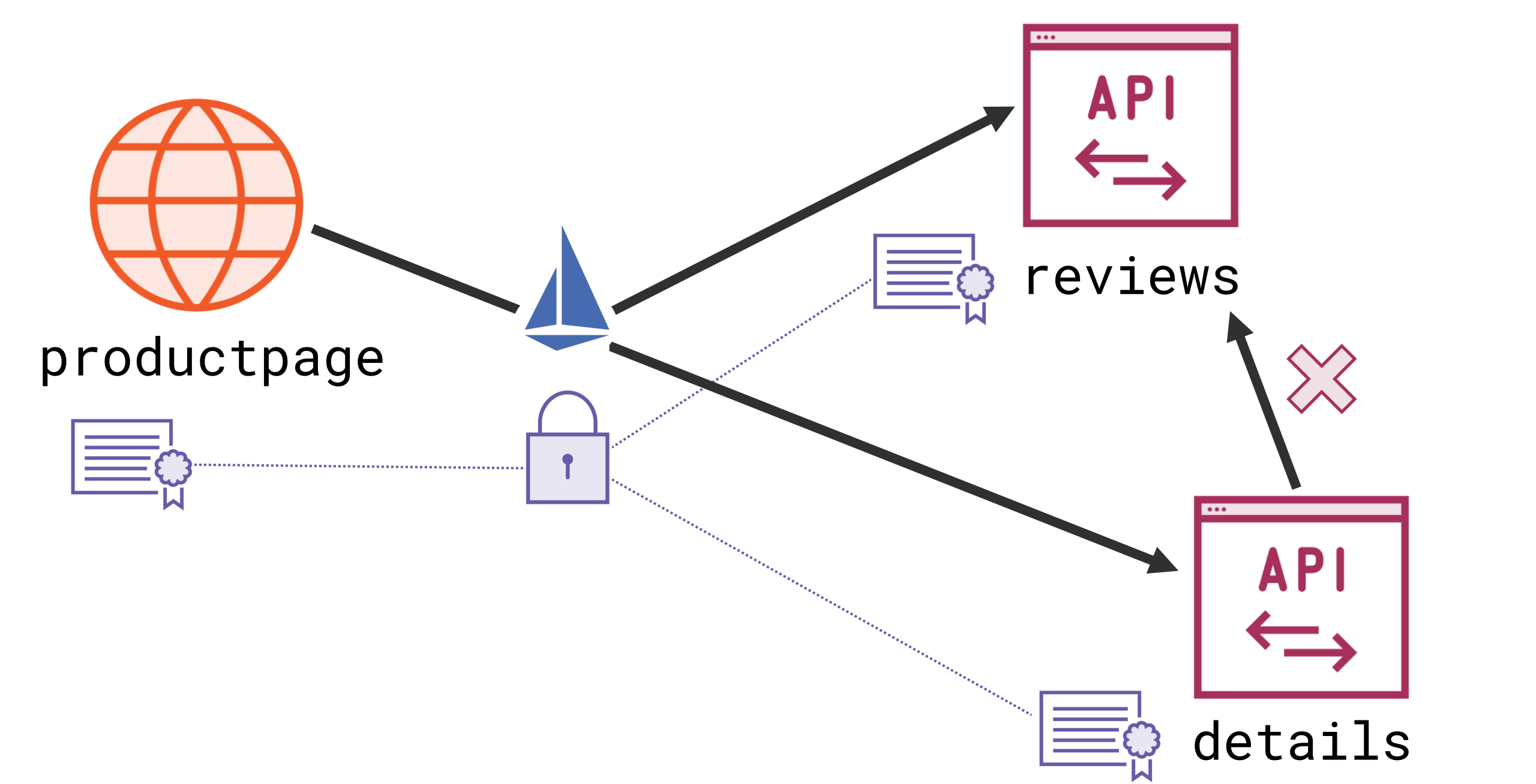
Securing Communication with Mutual TLS

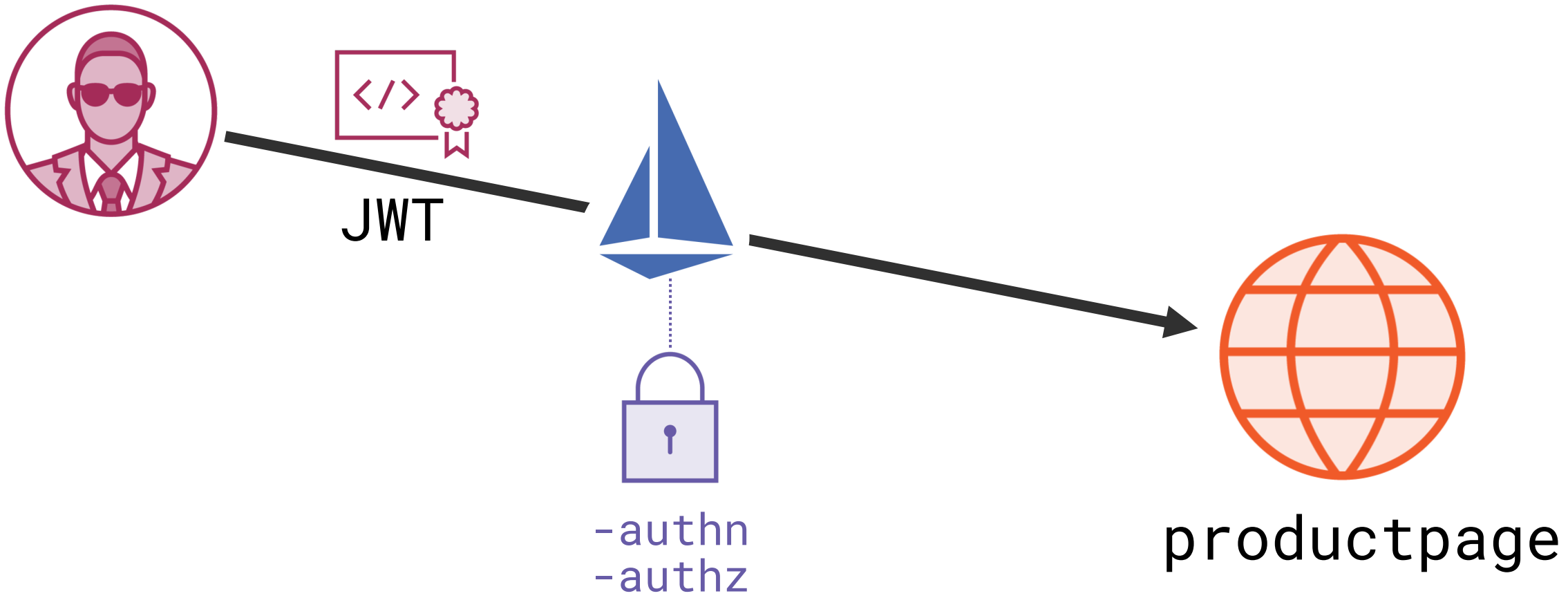
Elton Stoneman
CONSULTANT & TRAINER

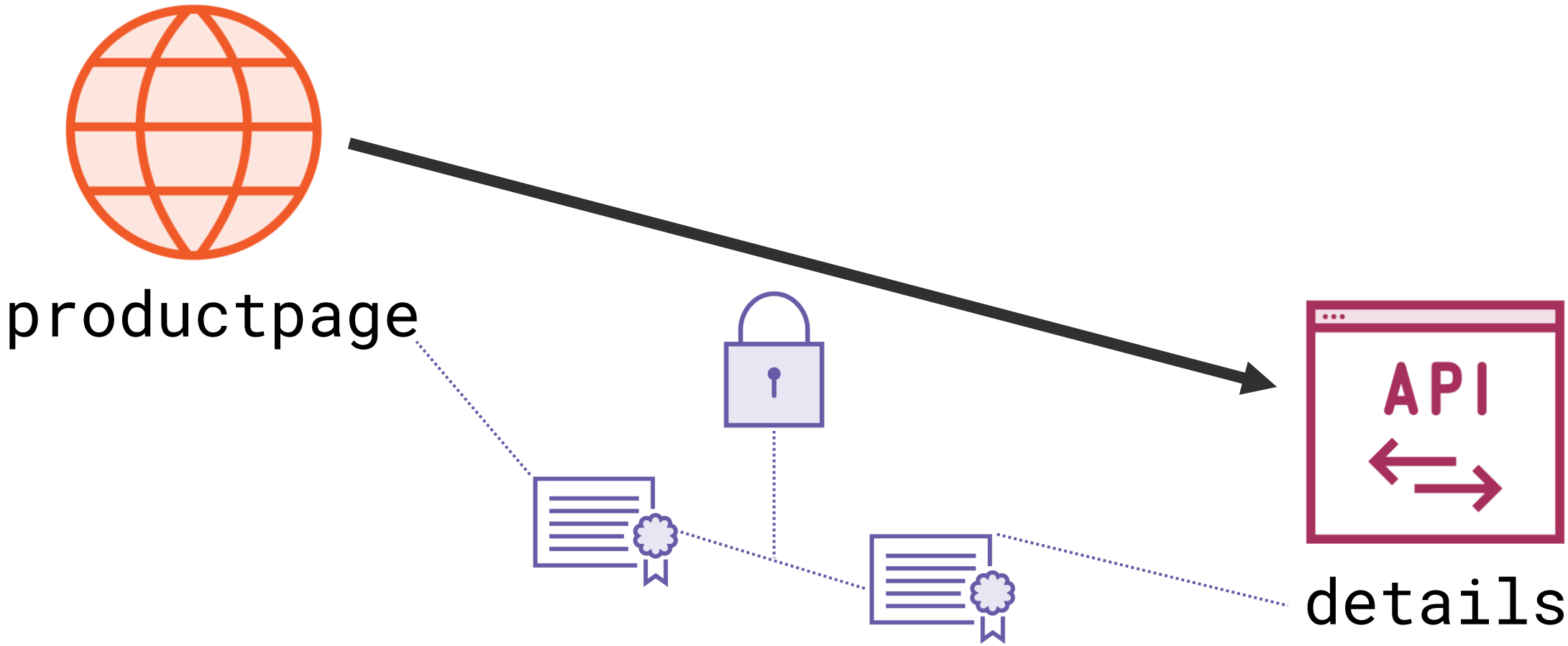
@EltonStoneman | blog.sixeyed.com

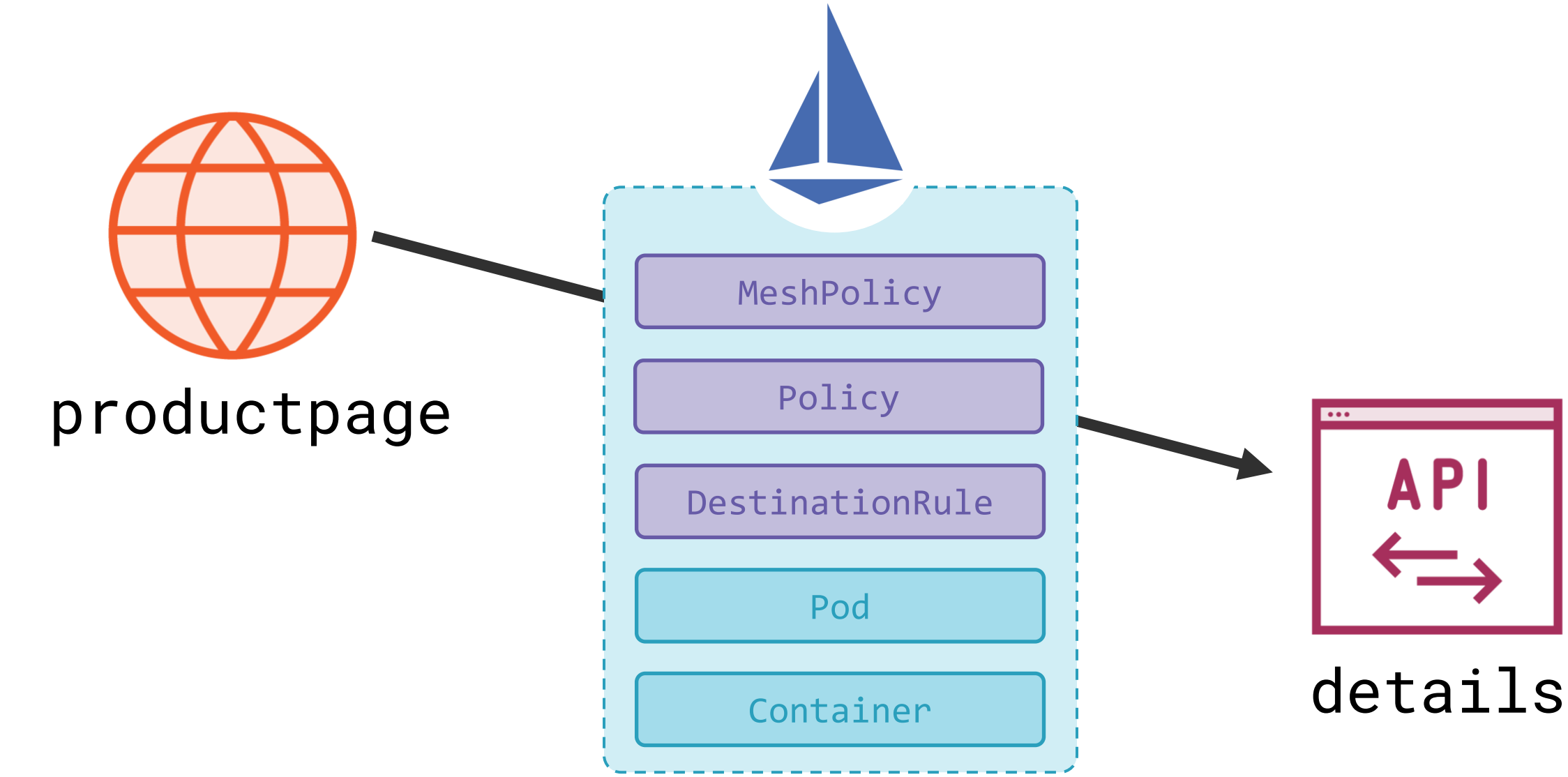

productpage
API
details


productpage
API
details


API
reviews
productpage
API
details


JWT
-authn
-authz
productpage


productpage
API
details


productpage
MeshPolicy
Policy
DestinationRule
Pod
Container
API
details

```
apiVersion: authentication.istio.io...
kind: MeshPolicy
metadata:
  name: default
spec:
  peers:
  - mtls: {}
```

- Default policy for all namespaces
- Required
- Authentication for calling service
- Require mutual TLS

```
apiVersion: authentication.istio.io...
kind: MeshPolicy
metadata:
  name: default
spec:
  peers:
  - mtls:
      mode: PERMISSIVE
```

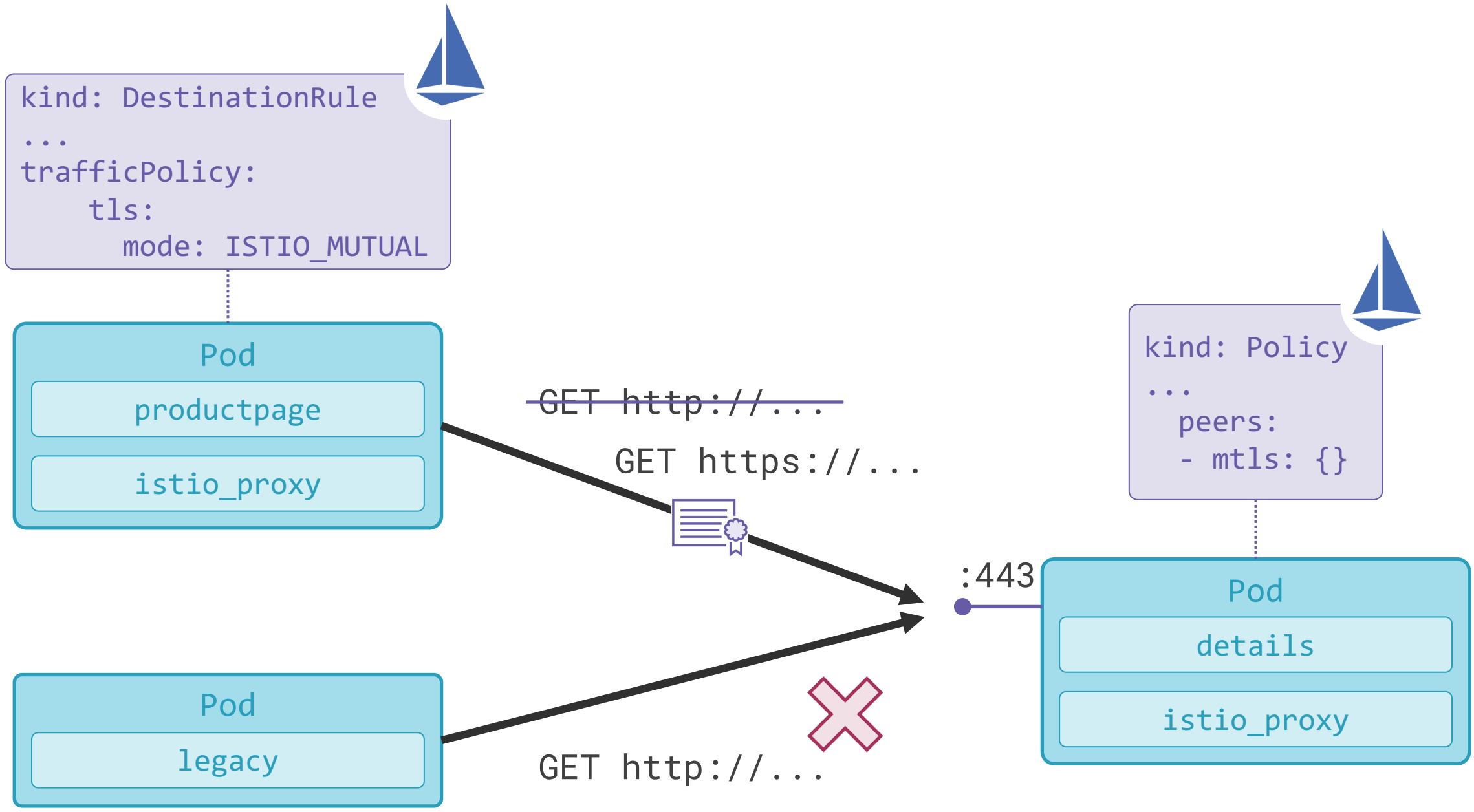
- Default policy for all namespaces
- Required
- Authentication for calling service
- Support mutual TLS

```
apiVersion: authentication.istio.io...
kind: Policy
metadata:
  name: default
  namespace: default
spec:
  peers:
  - mtls: {}
```

- Policy for specified targets
- Default for the `default` namespace
- Require mutual TLS

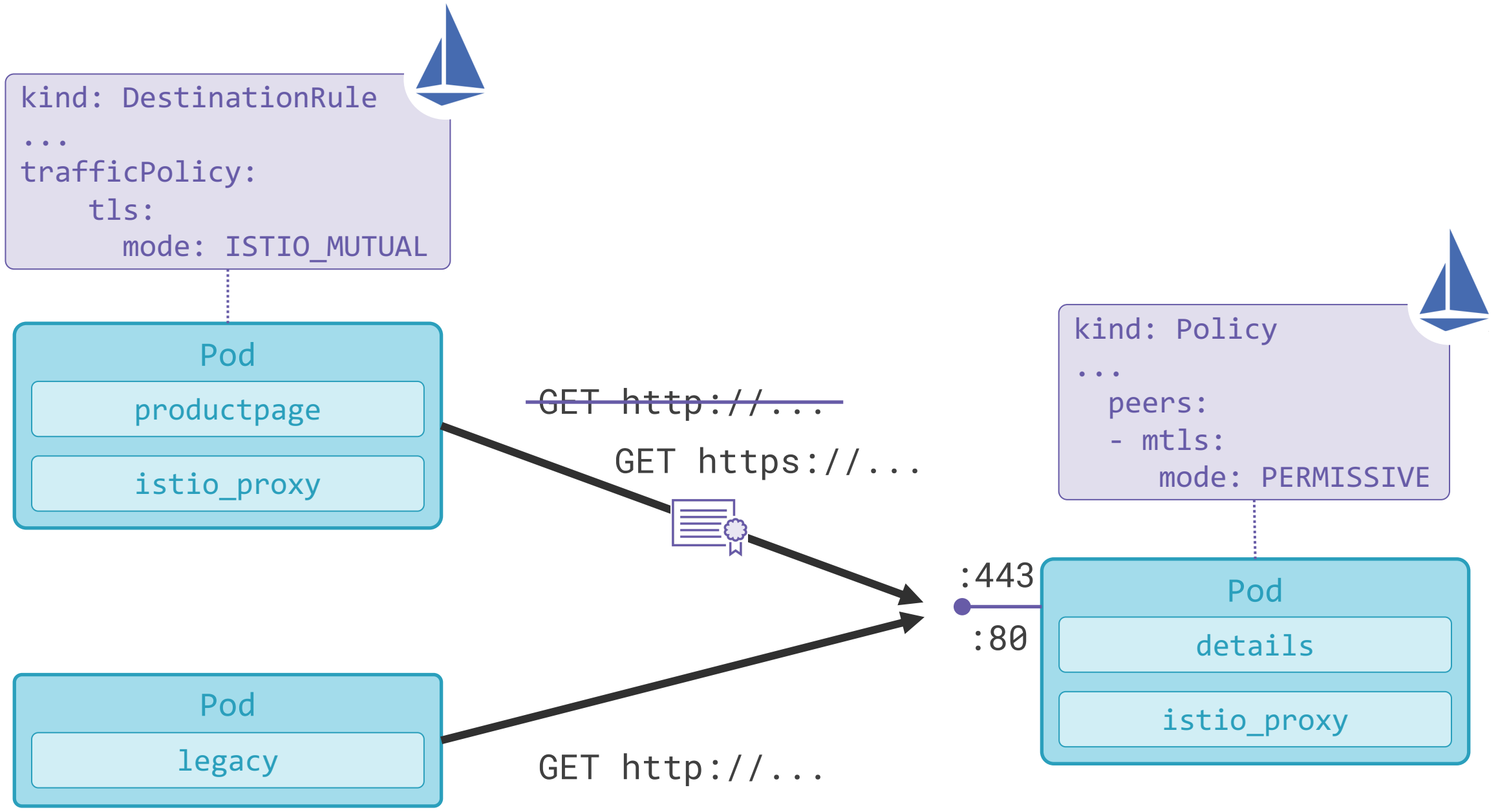
```
apiVersion: authentication.istio.io...
kind: Policy
metadata:
  name: default
  namespace: default
spec:
  peers:
  - mtls: {}
  targets:
  - name: productpage
```

- Policy for specified targets
- Default for the `default` namespace
- Require mutual TLS
- Apply to the `productpage` service

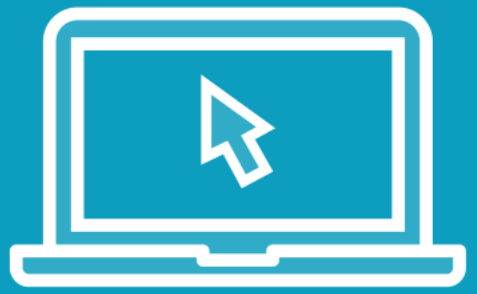

kind: DestinationRule
...
trafficPolicy:
tls:
mode: ISTIO_MUTUAL
Pod
productpage
istio_proxy
GET http://...
GET https://...
kind: Policy
...
peers:
- mtls: {}
:443
Pod
details
istio_proxy
Pod
legacy
GET http://...

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

- Default for the `default` namespace
- Any host in the same namespace
- Use mutual TLS
- With Istio-managed client certs

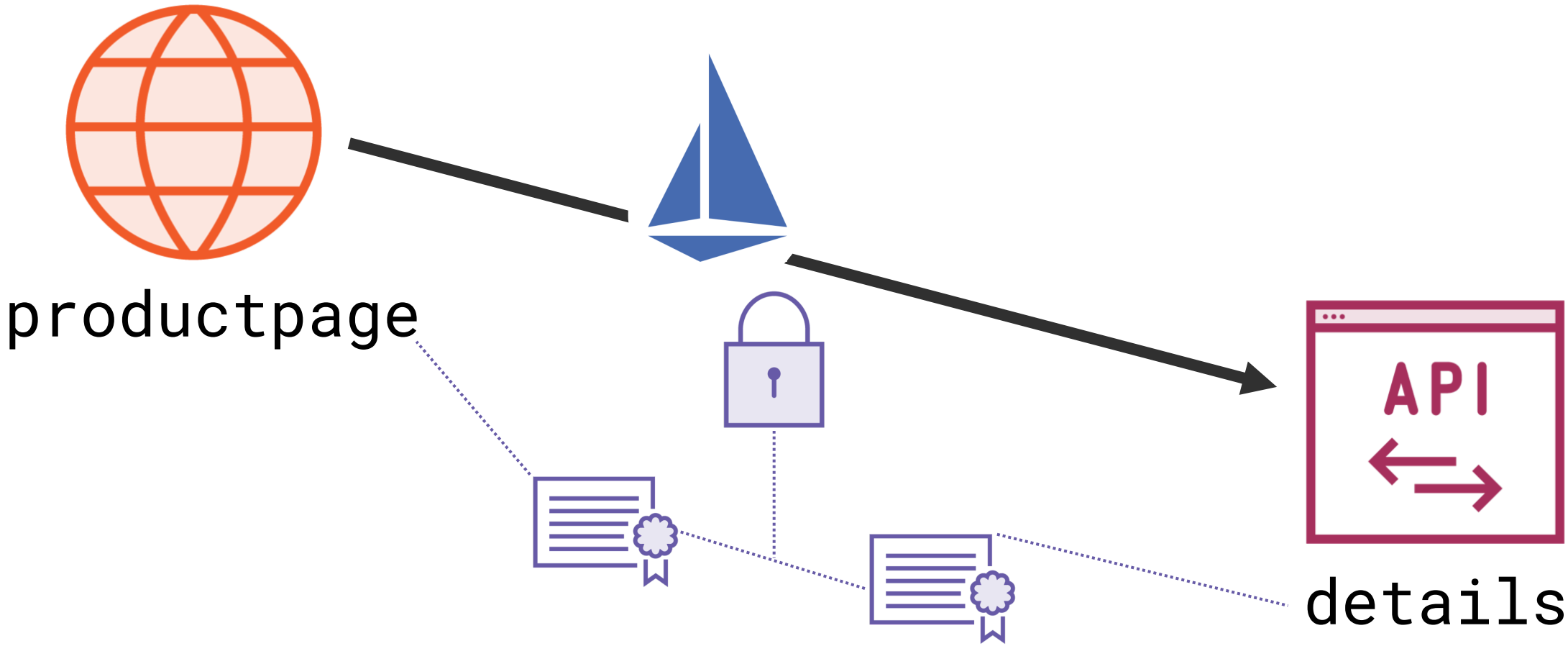

kind: DestinationRule
...
trafficPolicy:
tls:
mode: ISTIO_MUTUAL
Pod
productpage
istio_proxy
GET http://...
GET https://...
kind: Policy
...
peers:
- mtls:
mode: PERMISSIVE
:443
:80
Pod
details
istio_proxy
Pod
legacy
GET http://...

Demo



Securing Services with Mutual TLS

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


productpage
API
details

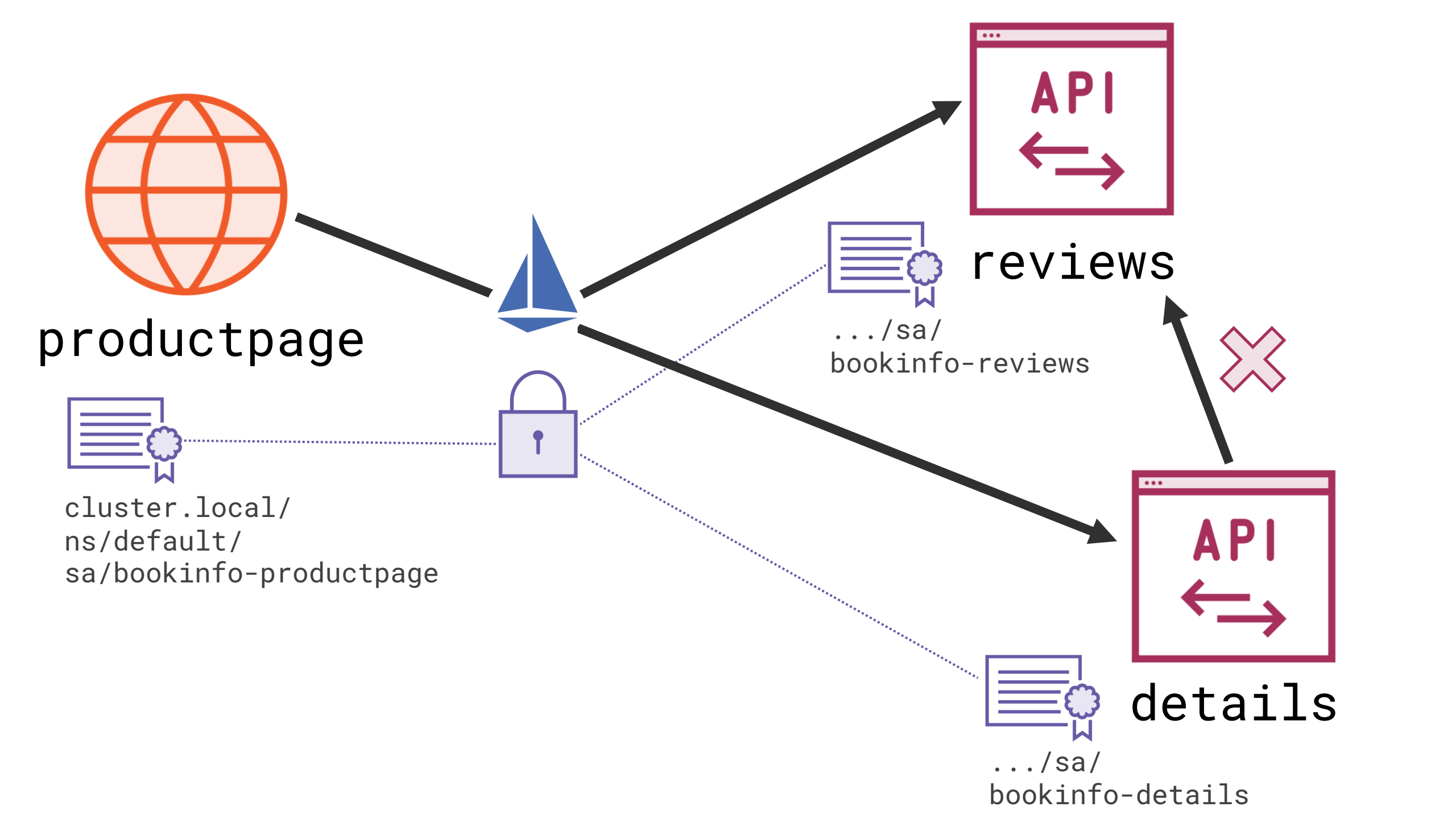
Enforcing Mutual TLS

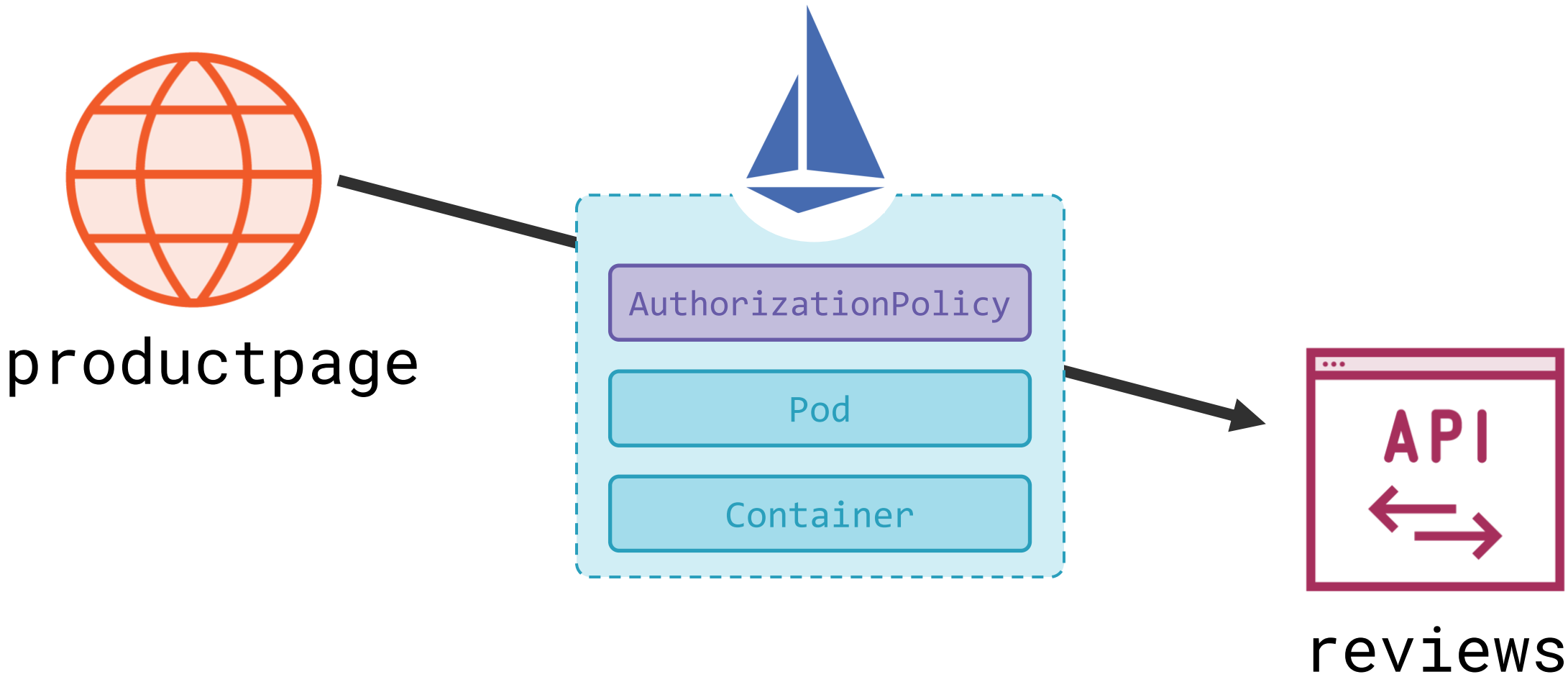
Policy.yaml

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

DestinationRule.yaml

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


API
reviews
productpage
.../sa/
bookinfo-reviews
cluster.local/
ns/default/
sa/bookinfo-productpage
API
details
.../sa/
bookinfo-details

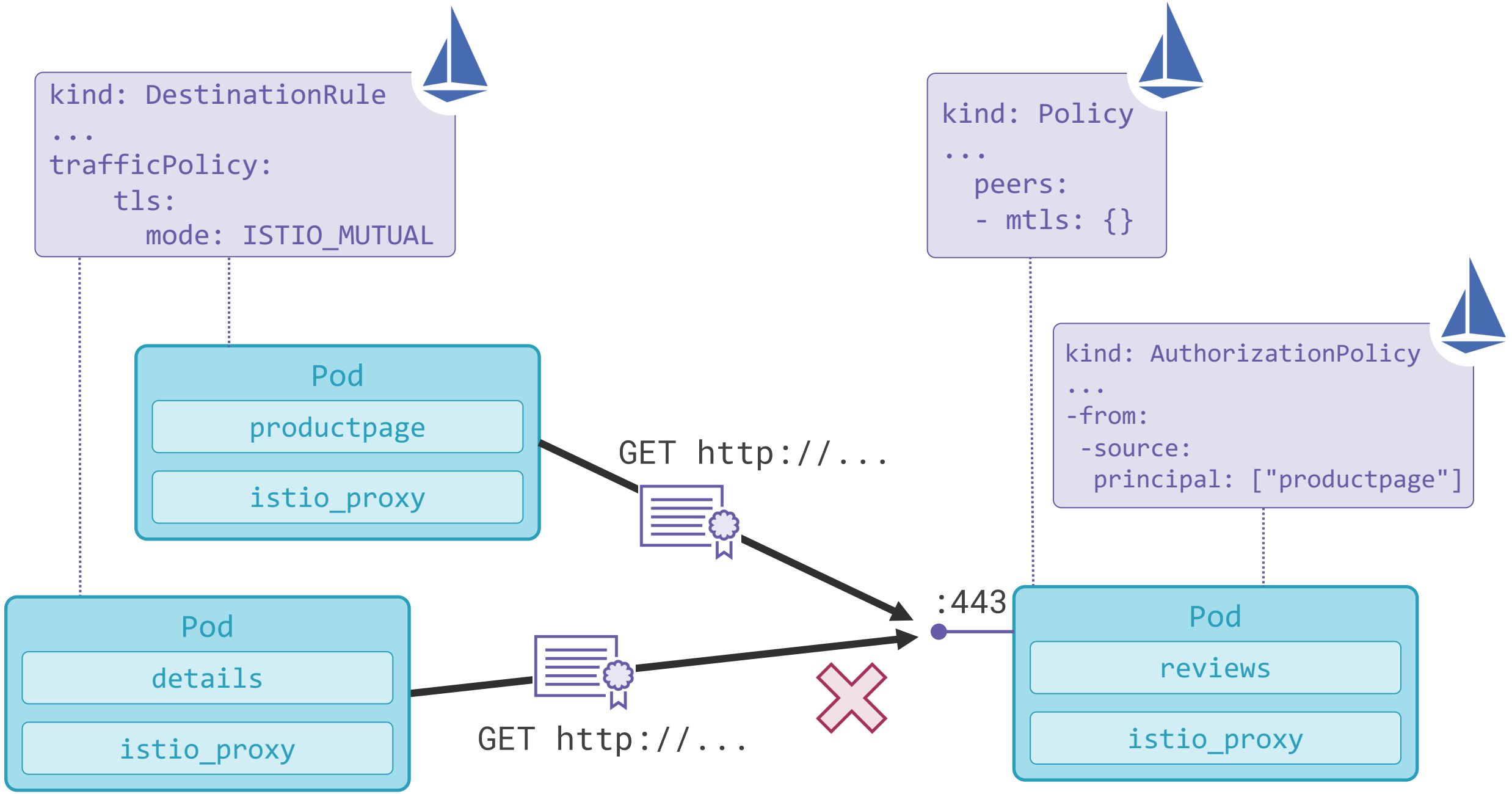

productpage
AuthorizationPolicy
Pod
Container
API
reviews

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

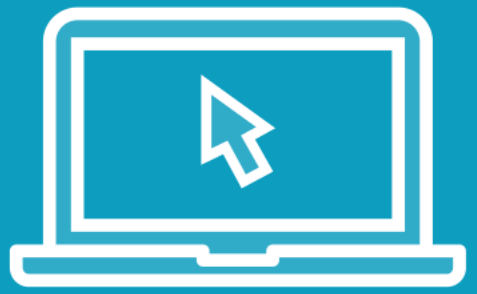
- Apply to `default` namespace
- Target selector identifies service(s)
- Default deny; no rules means deny all

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

- Rules specify allowed permissions
- Allow requests from specified principal
- To make GET calls to the service

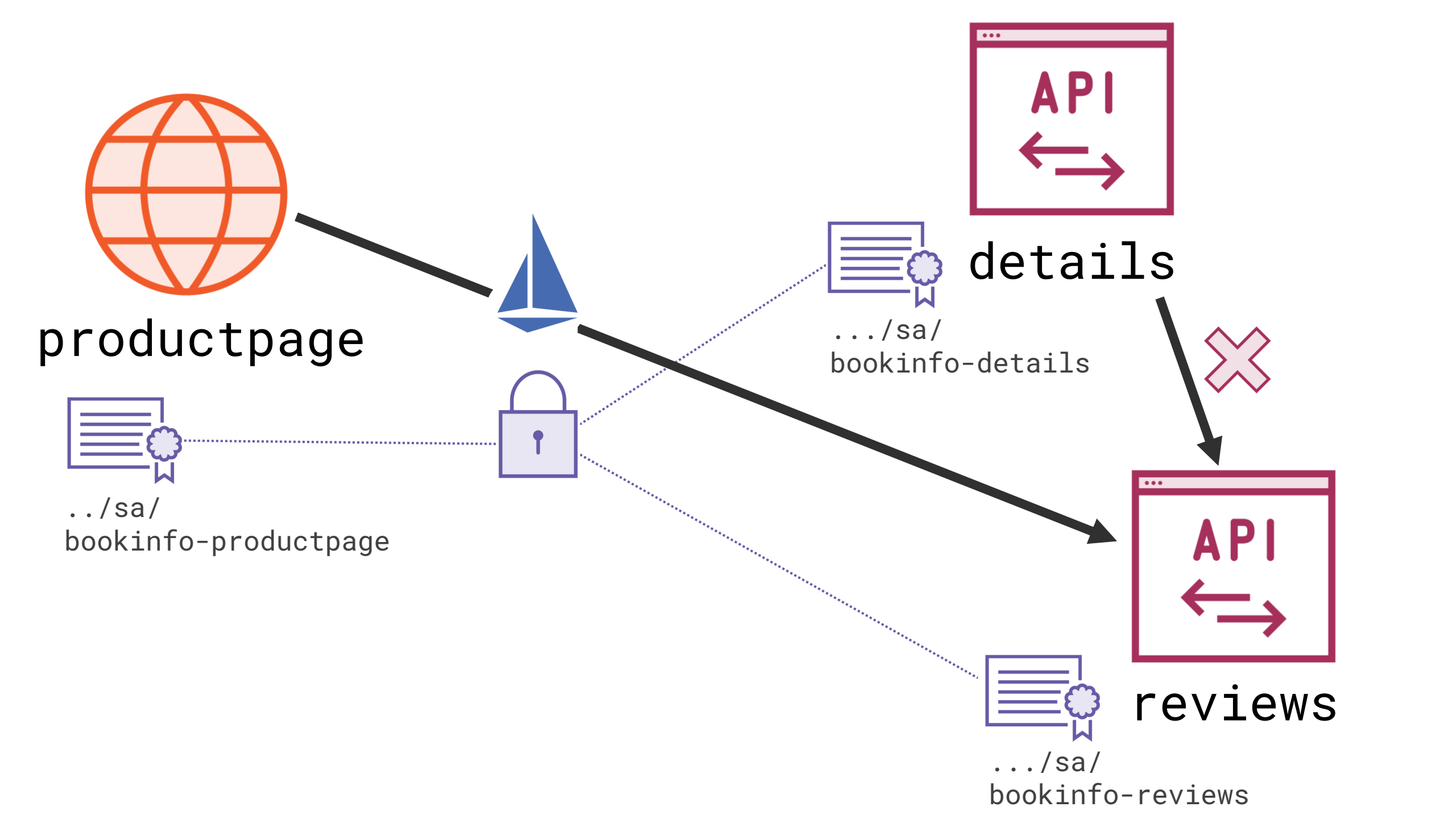

kind: DestinationRule
...
trafficPolicy:
tls:
mode: ISTIO_MUTUAL
kind: Policy
...
peers:
- mtls: {}
kind: AuthorizationPolicy
...
-from:
-source:
principal: ["productpage"]
Pod
productpage
istio_proxy
Pod
details
istio_proxy
GET http://...
GET http://...
:443
Pod
reviews
istio_proxy

Demo



Service Authorization with mTLS

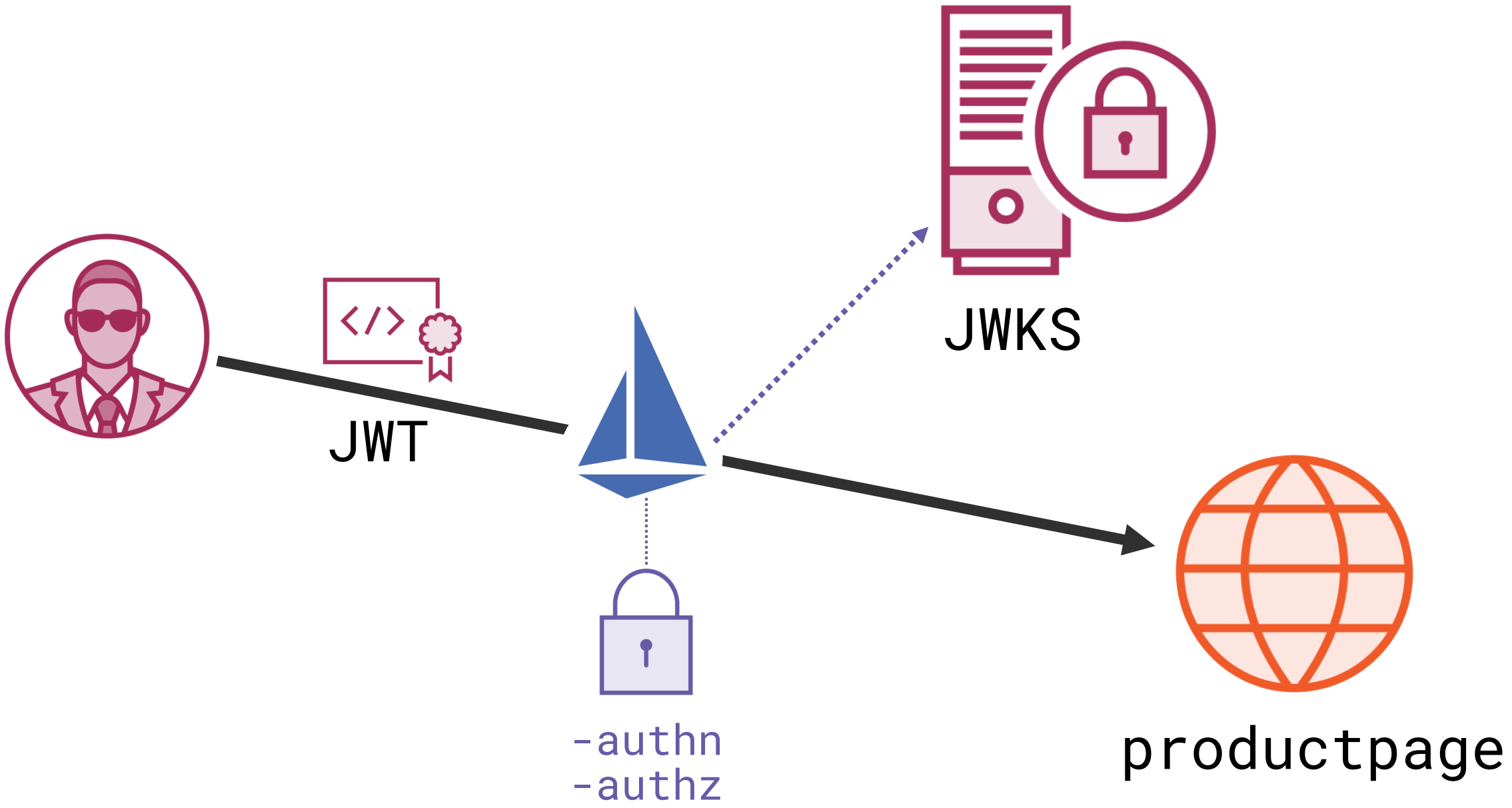
- Authorization with deny-all
- Allowing named principals
- Validate authentication identity

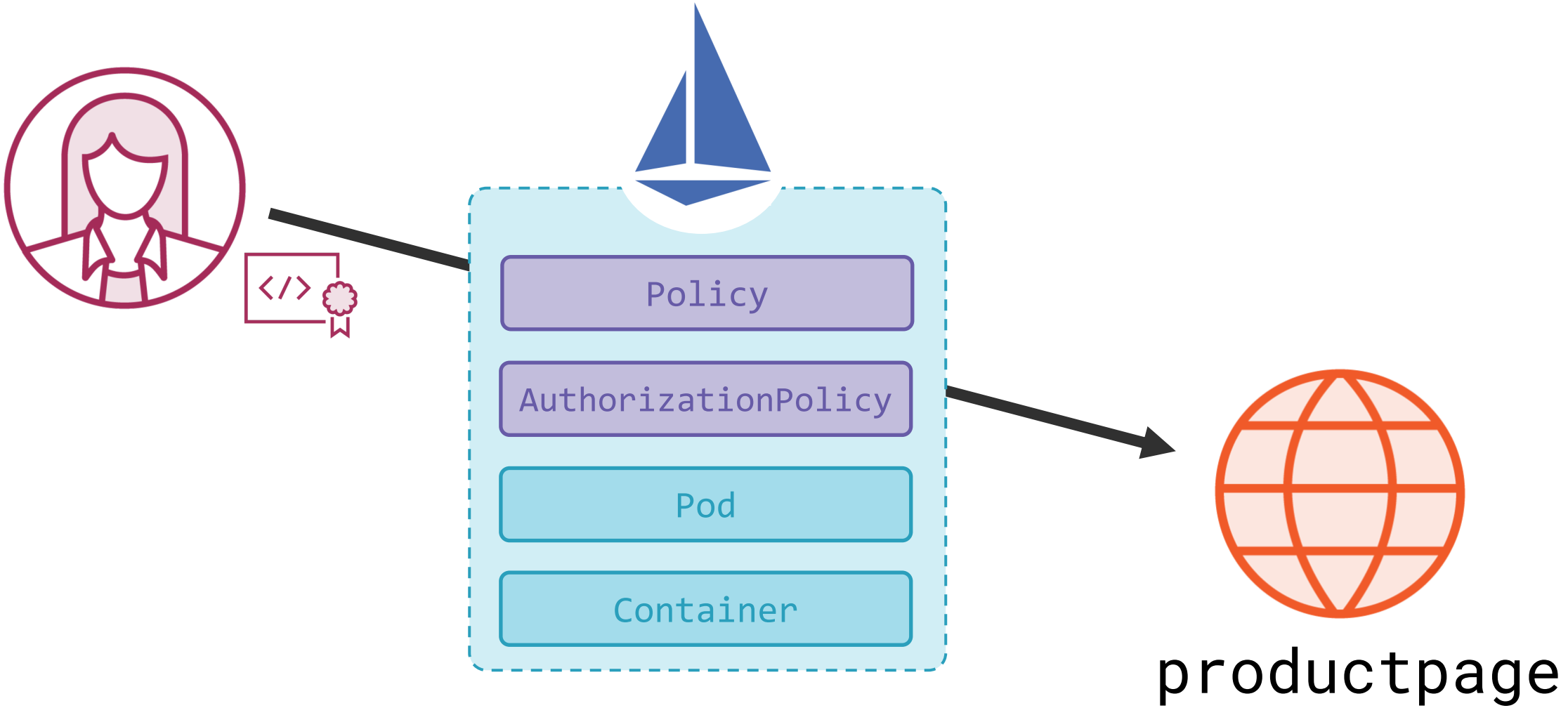

API
details
productpage
.../sa/
bookinfo-details
../sa/
bookinfo-productpage
API
reviews
.../sa/
bookinfo-reviews

Authorization with Service Principals

AuthorizationPolicy.yaml

```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"]
```


JWKS
JWT
-authn
-authz
productpage

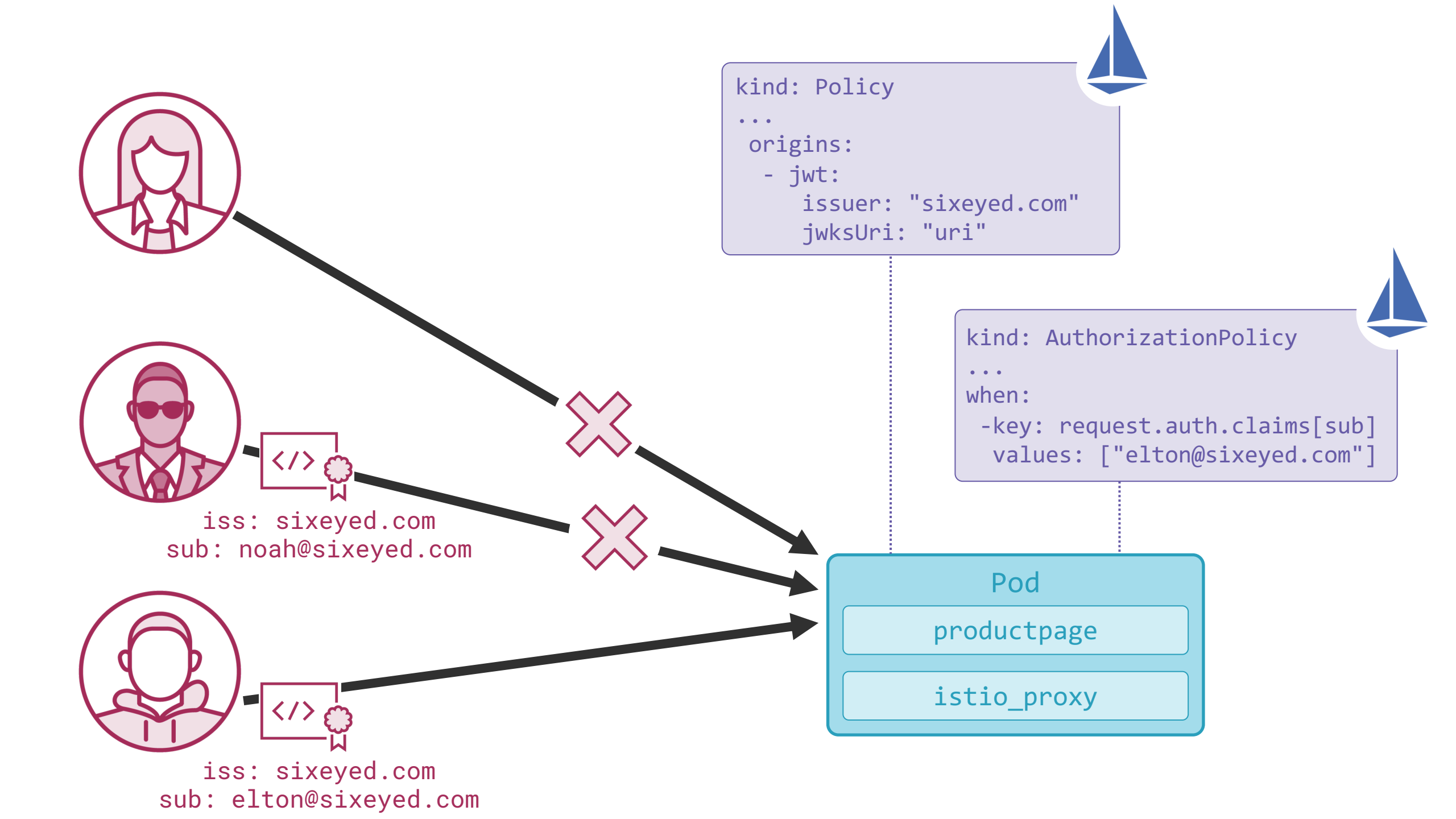

Policy
AuthorizationPolicy
Pod
Container
productpage

```
apiVersion: authentication.istio.io/...
kind: Policy
metadata:
  name: productpage-authn
spec:
  targets:
  - name: productpage
  origins:
  - jwt:
      issuer: "xyz"
      jwksUri: "uri"
  principalBinding: USE_ORIGIN
```

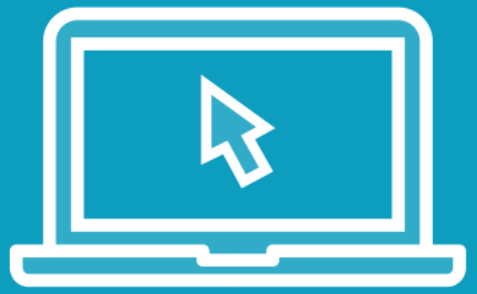
- Service authentication
- Target selector for service(s)
- Authentication for end user
- Require JWT
- JWT issuer and JWKS server address
- Sets principal from JWT

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
...
spec:
  rules:
  - to:
    - operation:
        methods: ["GET"]
    when:
    - key: request.auth.claims[sub]
      values: ["elton@sixeyed.com"]
```

- Service access
- Allow access to GET methods
- When the JWT has the specified claim

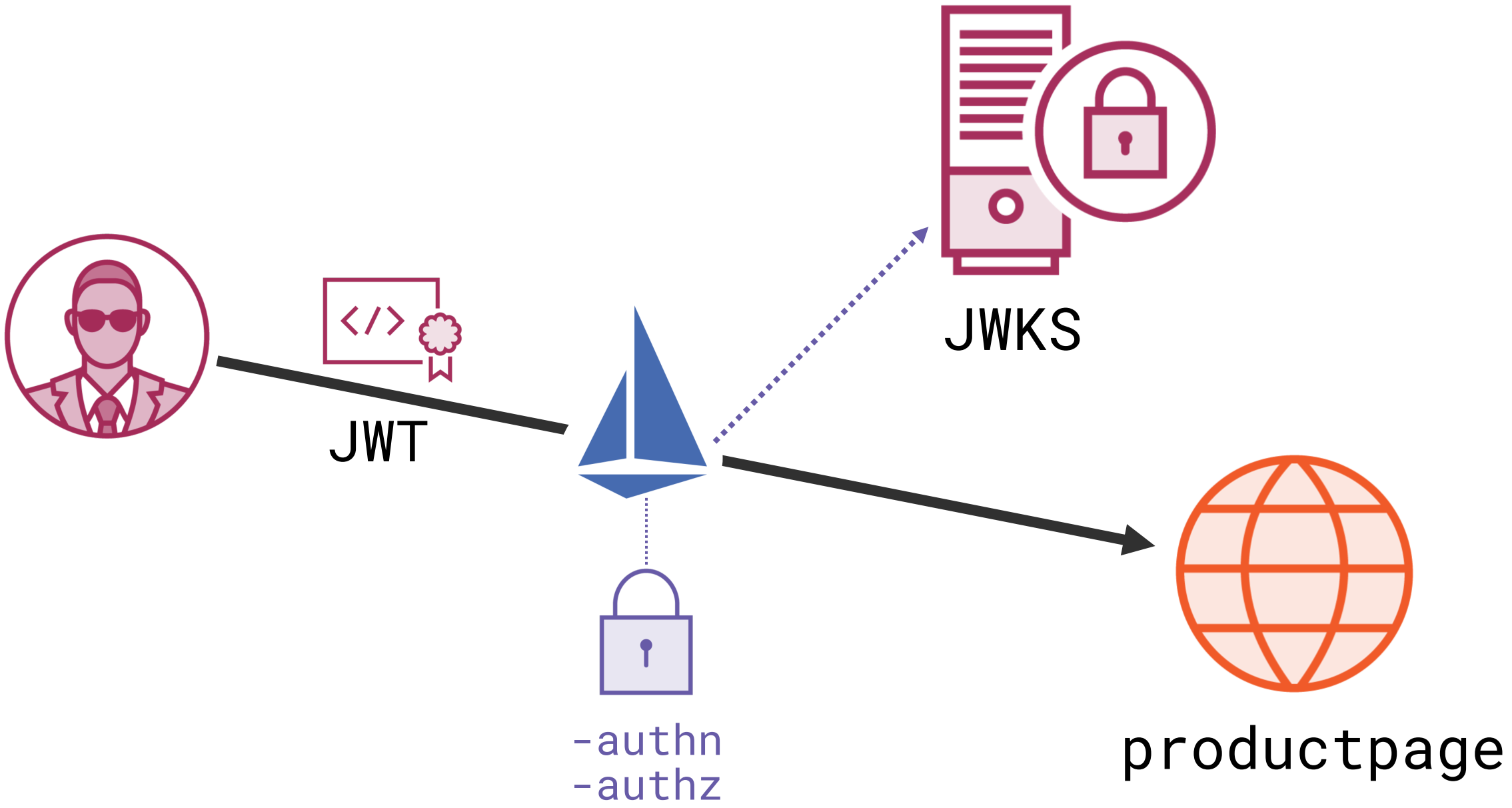

kind: Policy
...
origins:
- jwt:
issuer: "sixeyed.com"
jwksUri: "uri"
kind: AuthorizationPolicy
...
when:
-key: request.auth.claims[sub]
values: ["elton@sixeyed.com"]
iss: sixeyed.com
sub: noah@sixeyed.com
iss: sixeyed.com
sub: elton@sixeyed.com
Pod
productpage
istio_proxy

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

Policy.yaml

```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

```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