GOTC

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE

OPEN SOURCE , OPEN WORLD

「CNCF云原生」专场

云原生服务网格安全解密

徐中虎 2021年07月10日

About me

GOTC

徐中虎(@hzxuzhonghu)

华为云云原生开源工程师

- Istio Steering Committee委员
- Istio社区资深Maintainer，核心贡献者
- Volcano项目维护者
- Kubernetes 核心贡献者
- 《云原生服务网格-Istio》联合作者

服务网格安全概述

GOTC

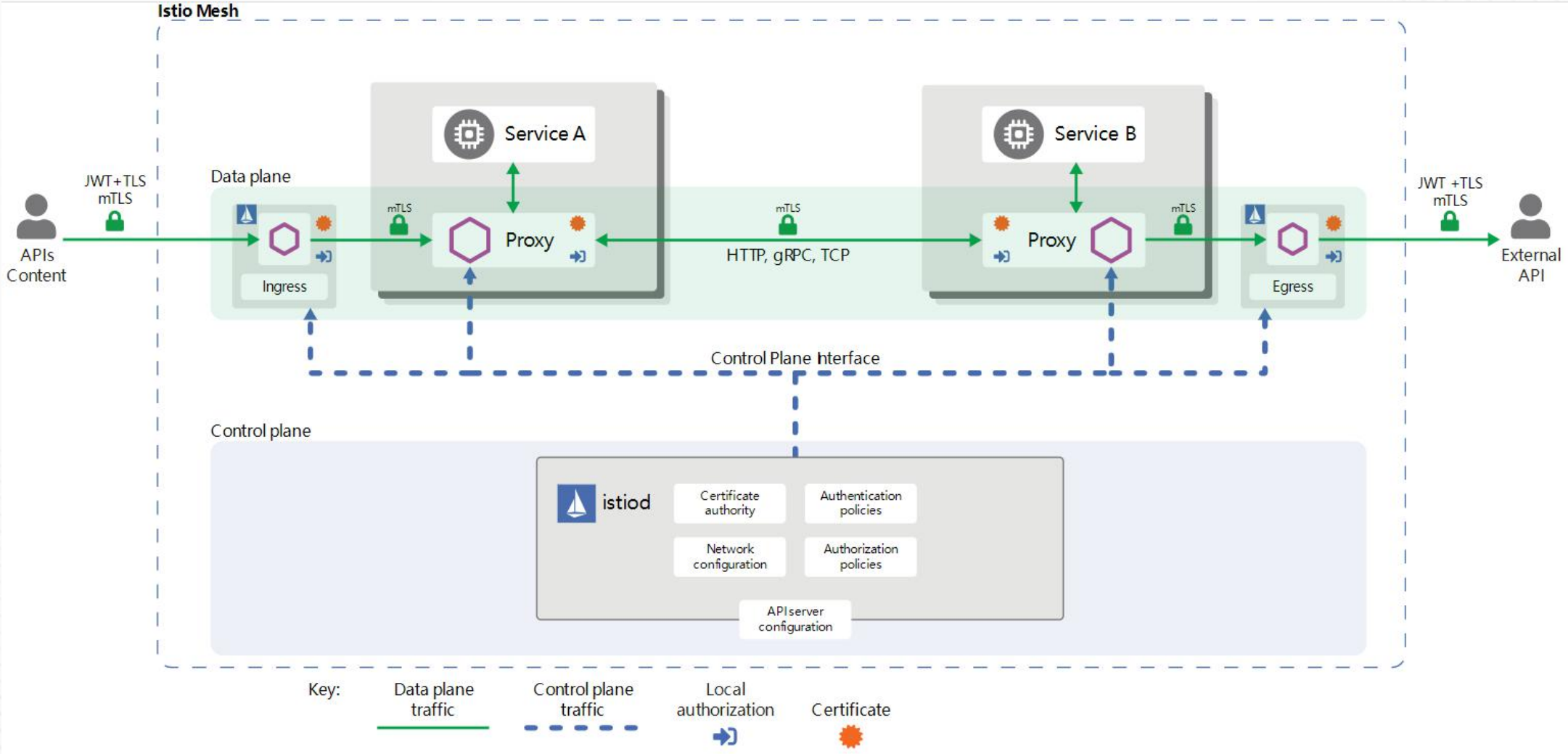
微服务安全需求:

- 为防御中间人攻击，流量需要加密
- 为了提供灵活的服务访问控制，需要双向 TLS 和细粒度的访问策略。
- 为了明确服务访问细节，需要Audit

Istio服务网格安全目标：

- 默认安全：无需更改应用程序代码和基础架构，零配置
- 深度防御：与现有安全系统集成，提供多层防御
- 零信任网络：在不受信任的网络上构建安全解决方案

Istio安全架构


Istio Mesh
Data plane
Service A
Service B
Proxy
Proxy
Ingress
Egress
mTLS
mTLS
mTLS
mTLS
JWT+TLS
mTLS
APIs
Content
HTTP, gRPC, TCP
JWT +TLS
mTLS
External
API
Control Plane Interface
Control plane
istiod
Certificate authority
Authentication policies
Network configuration
Authorization policies
APIserver configuration
Key:
Data plane traffic
Control plane traffic
Local authorization
Certificate

Istio安全模型

- Service-> Service数据面通信
  - 自动双向TLS认证，默认安全加密
  - 工作负载身份独立，符合SPIFFEE格式：*cluster.local/ns/default/sa/sleep*，身份基于Kubernetes ServiceAccount
- Authentication
  - x509认证， JWT认证
- Authorization
  - 内置授权，外部授权

证书管理

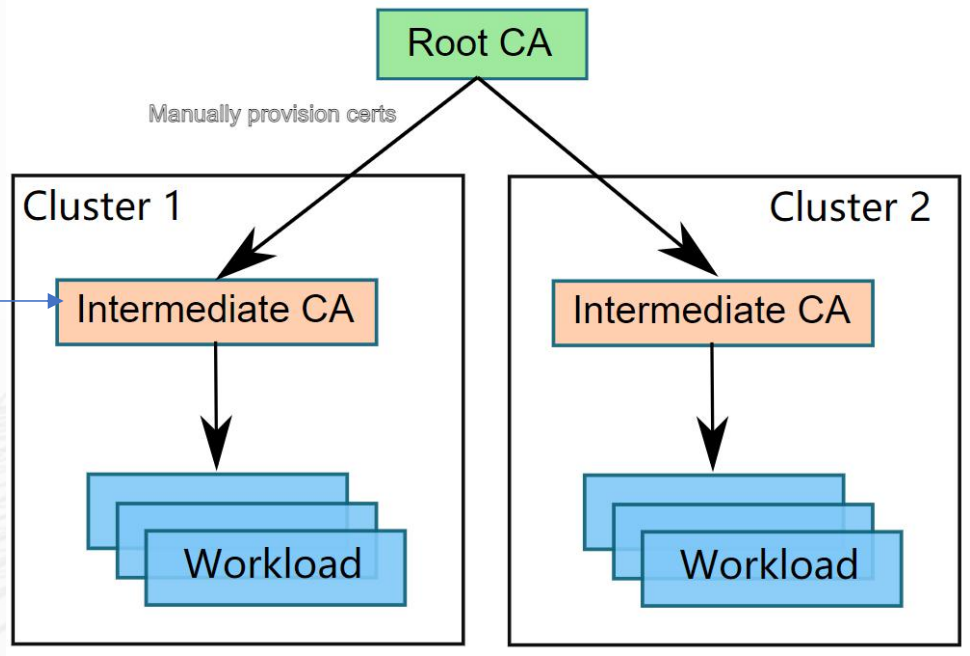
GOTC

证书Provision

- SDS：动态Provision，Istio负责证书轮换
- 自动挂载：静态Provision，可以通过K8s Secret挂载，或者VM手动提供

证书签发

- Plug in CA证书
- 个性化CA集成（Kubernetes CSR）



Authentication

GOTC

- 自动mTLS：
- 支持全局、namespace级、服务级严格mTLS配置

  优先级顺序：服务级 > namespace级 > 全局

```
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: "default“
  namespace: default
spec:
  selector:
    matchLabels:
      app: foo
  mtls:
    mode：STRICT
```

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

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

Authentication-JWT

GOTC

```yaml
apiVersion: security.istio.io/v1beta1
kind: RequestAuthentication
metadata:
  name: httpbin
  namespace: foo
spec:
  selector:
    matchLabels:
      app: httpbin
  jwtRules:
  - issuer: "issuer-foo"
    jwksUri: https://example.com/.well-known/jwks.json
```

```yaml
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: httpbin
  namespace: foo
spec:
  selector:
    matchLabels:
      app: httpbin
  action: ALLOW
  rules:
  - from:
    - source:
        requestPrincipals: ["*"]
```

JWT认证往往与Authz结合使用，限制认证的请求

Authorization

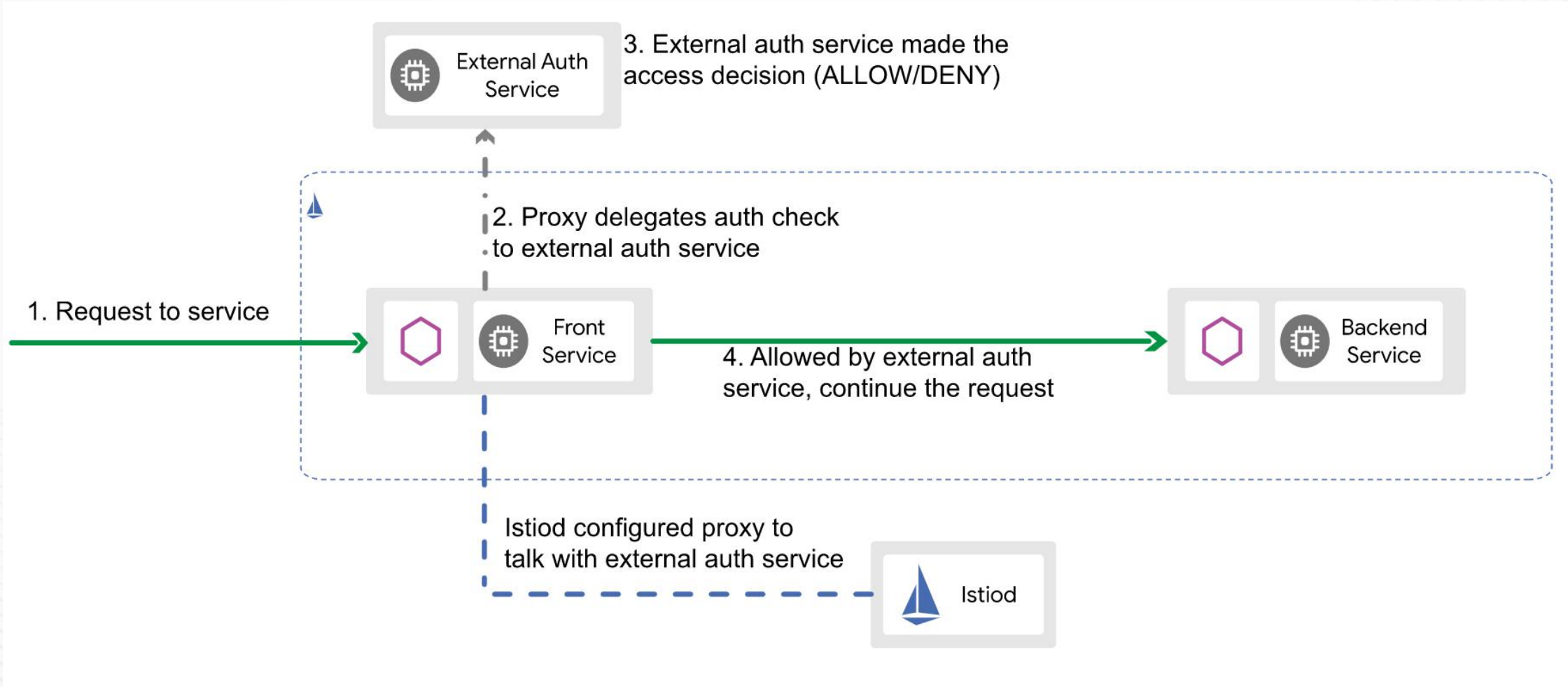
GOTC

- 支持直接ALLOW or DENY CUSTOM
- 支持TCP、HTTP鉴权，规则设置：
  - 支持源端：身份，namespace，IP地址
  - 支持目的端：Authority，端口，方法，URI路径
- JWT Token

  需要配合RequestAuthentication
- External Auth：CUSTOM

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
 name: httpbin
 namespace: foo
spec:
 selector:
   matchLabels:
     app: httpbin
     version: v1
 action: ALLOW
 rules:
 - from:
   - source:
       principals: ["cluster.local/ns/default/sa/sleep"]
   - source:
       namespaces: ["dev"]
   to:
   - operation:
       methods: ["GET"]
   when:
   - key: request.auth.claims[iss]
     values: ["https://accounts.google.com"]
```

External Auth



External Auth

GOTC

- 开发External Auth 服务，支持HTTP和gRPC两种协议
- AuthorizationPolicy ***Custom*** action

```
extensionProviders:
- name: "my-ext-authz-service"
    envoyExtAuthzGrpc:
      service: "ext-authz.istio-system.svc.cluster.local"
      port: 9000
```

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: ext-authz
  namespace: istio-system
spec:
selector:
    matchLabels:
      app: istio-ingressgateway
action: CUSTOM
provider:
    name: "my-ext-authz-service"
rules:
  - to:
    - operation:
        paths: ["/admin/*"]
```

GOTC

THANKS

全球开源技术峰会

THE GLOBAL OPENSOURCE TECHNOLOGY CONFERENCE