
Architect
SACC
2022 中国系统架构师大会
SYSTEM ARCHITECT CONFERENCE CHINA 2022
激发架构性能 点亮业务活力
云上会议 网络直播 | 2022年10月27-29日
ITPUB

微服务下的身份认证和令牌管理

软件架构师 刘勇智

日程

背景

系统自身鉴权

API 网关鉴权

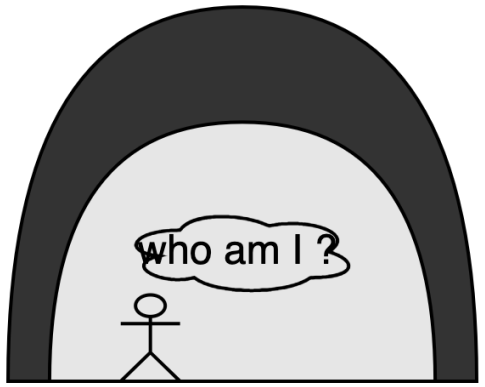
Authentication Sidecar 模式

扩展和总结

演进

Q & A

背景：服务间认证、授权和凭证

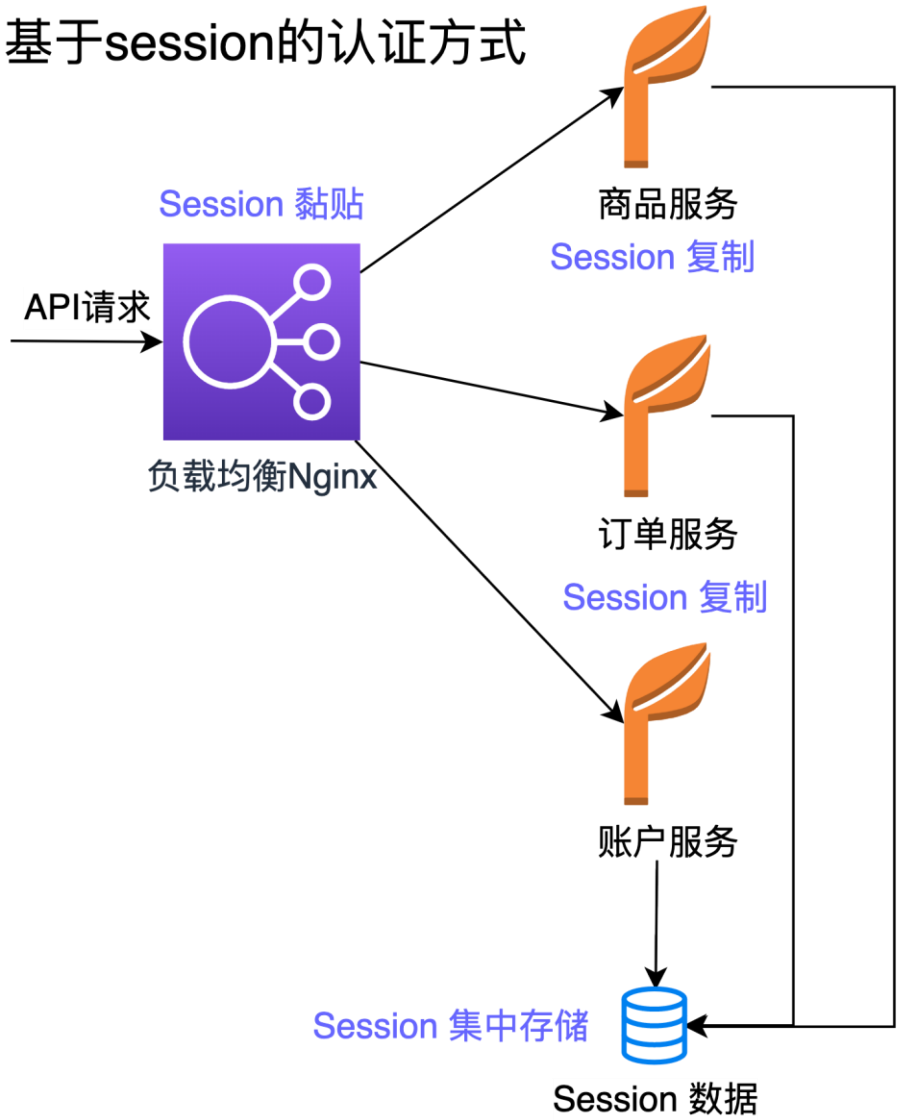


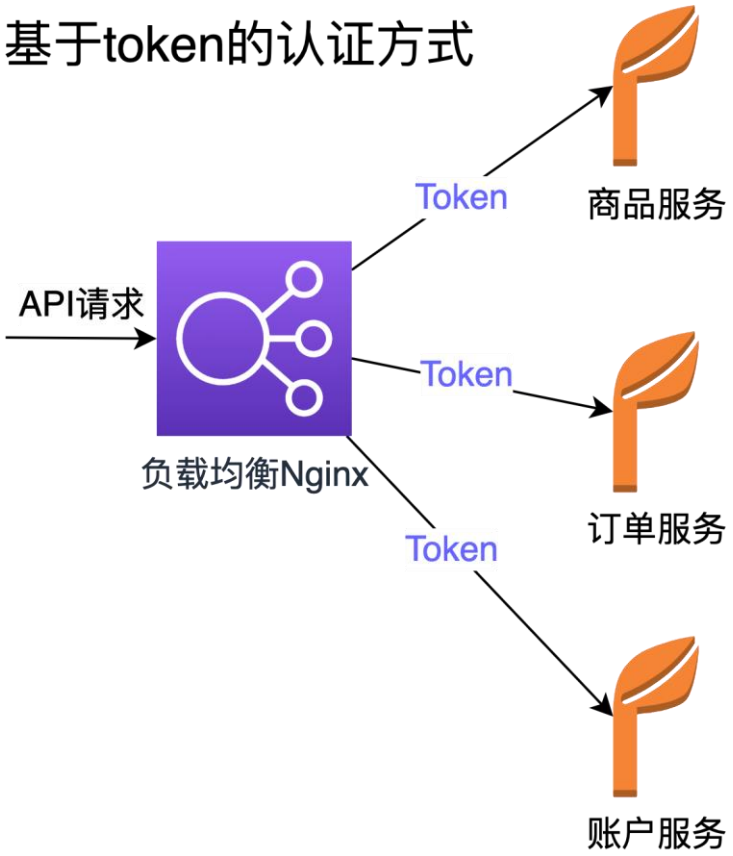
我是谁?
认证 / authentication

我能做什么?
授权 / authorization

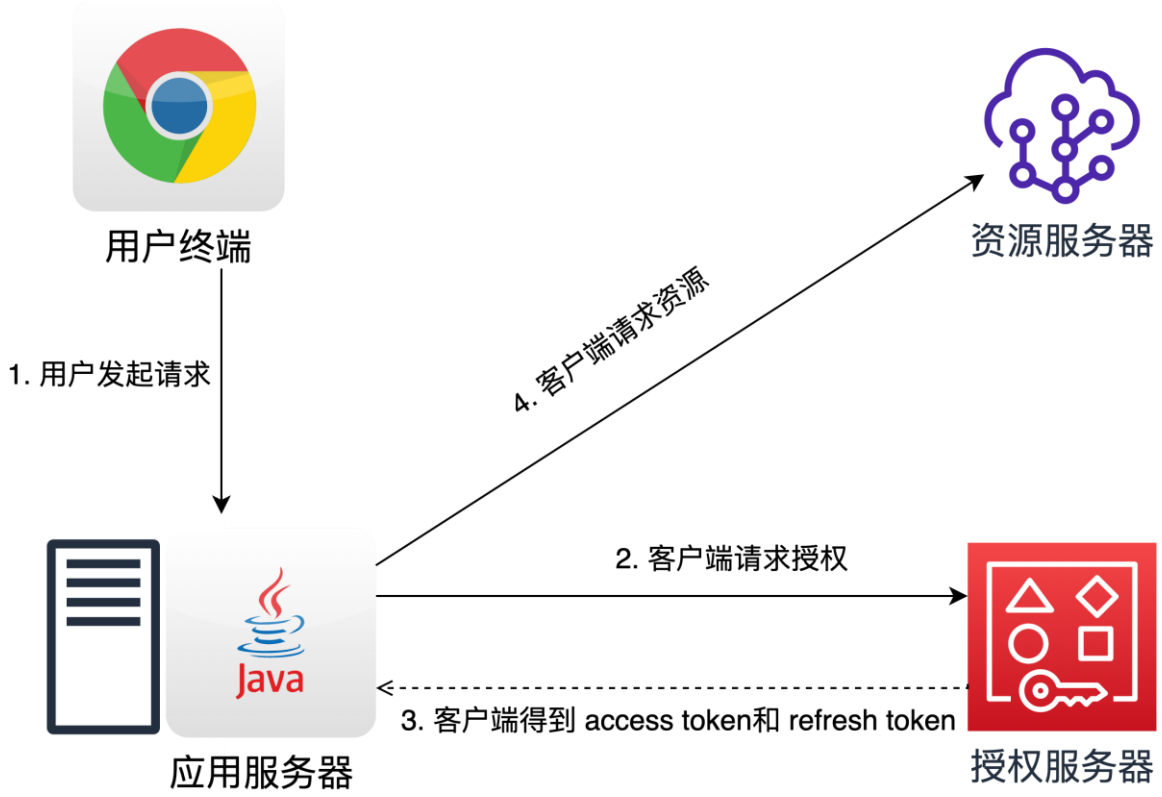
证据是什么?
凭证 / credentials

背景：常见的认证方式


基于session的认证方式
Session 黏贴
API请求
负载均衡Nginx
商品服务
Session 复制
订单服务
Session 复制
账户服务
Session 集中存储
Session 数据


基于token的认证方式
Token
商品服务
API请求
负载均衡Nginx
Token
订单服务
Token
账户服务

背景：OAuth2 和 JWT Token



Encoded

```
eyJhbGciOiJIUzI1NiIsInR5cCI6
IkpXVCJ9.eyJzdWIiOiIxMjM0NTY
3ODkwIiwibmFtZSI6IkpvaG4gRG9
lIiwiYWRtaW4iOnRydWV9.TJVA95
OrM7E2cBab30RMHrHDcEfxjoYZge
FONFh7HgQ
```

Decoded

HEADER:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD:

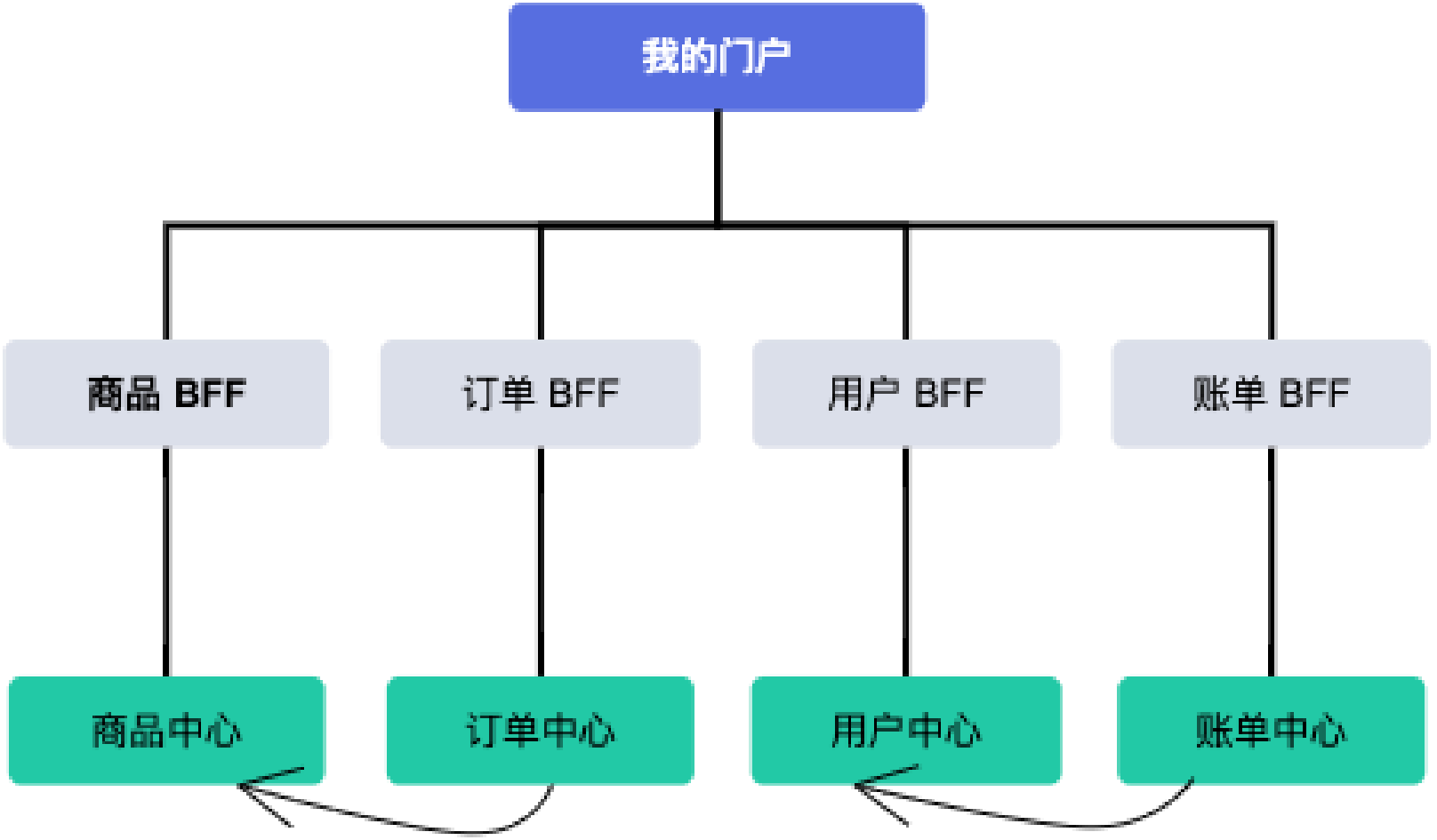
```
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

VERIFY SIGNATURE

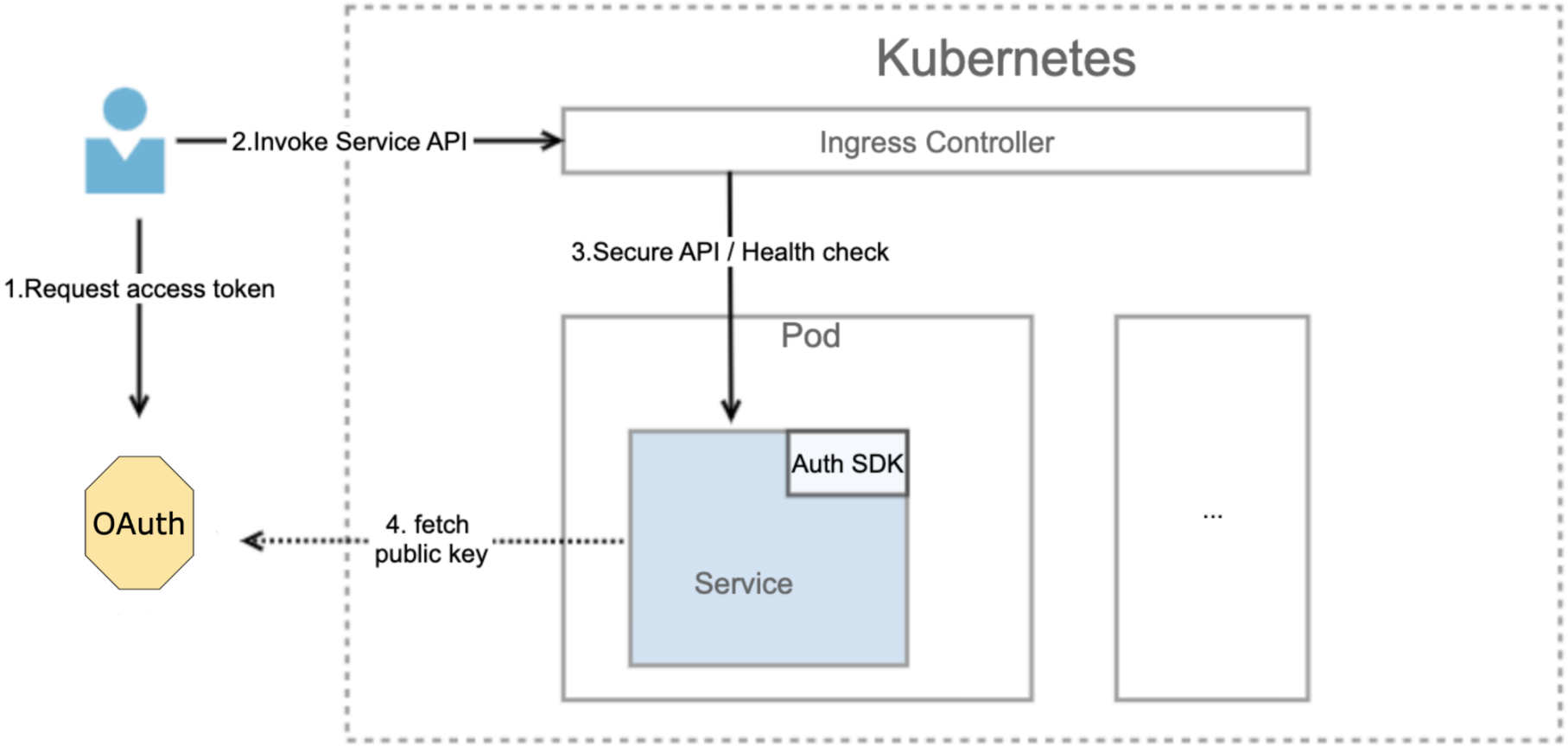
```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  secret
) ☐secret base64 encoded
```

Signature Verified

背景：项目



系统自身鉴权：入站身份验证



痛点：

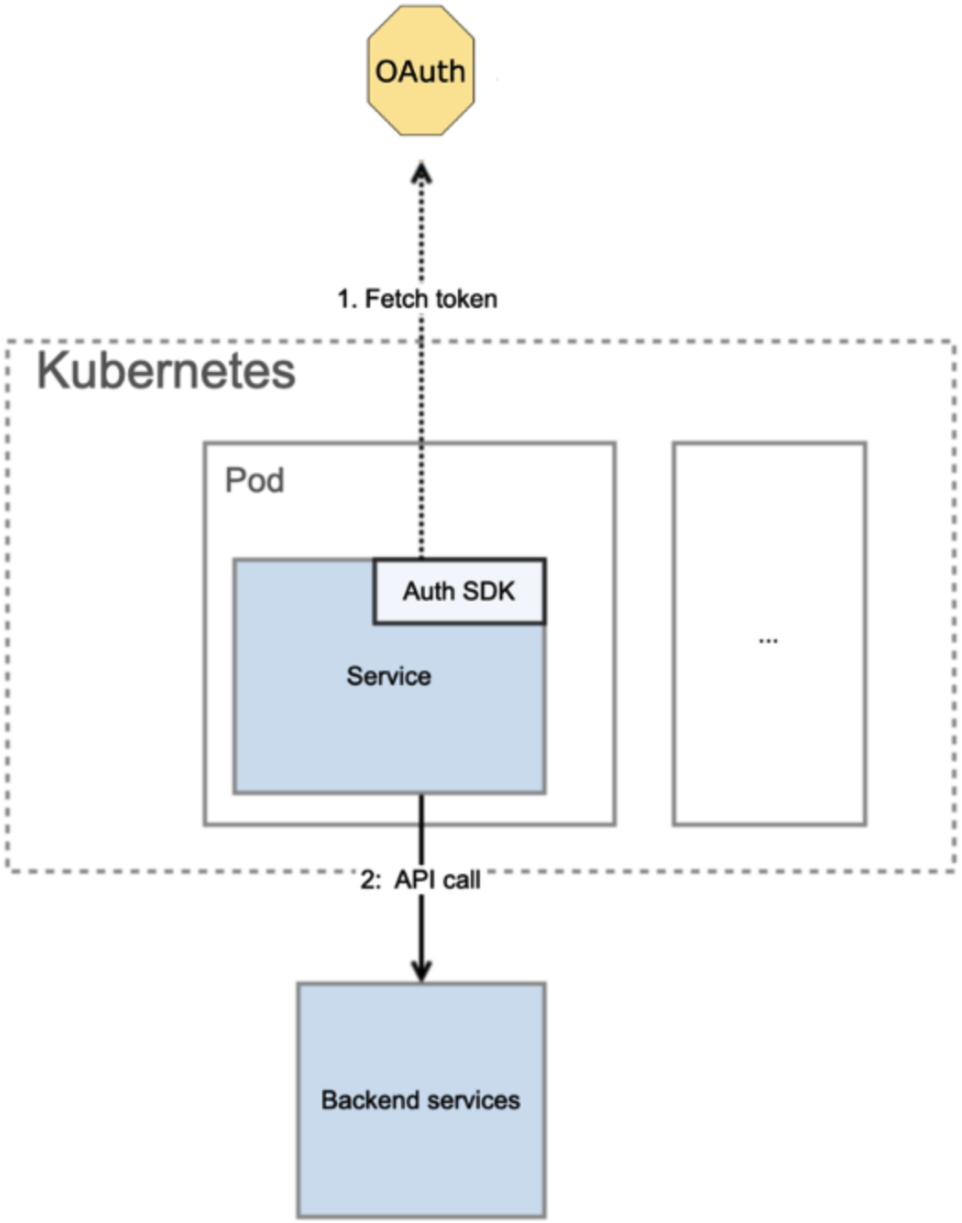
1. 耦合性：Service高度依赖于authentication SDK
2. 复杂性：Service还需要关注鉴权和token管理
3. 复用性：Auth SDK异构语言治理
4. 可维护性：升级成本高

系统自身鉴权

- 本地出站请求

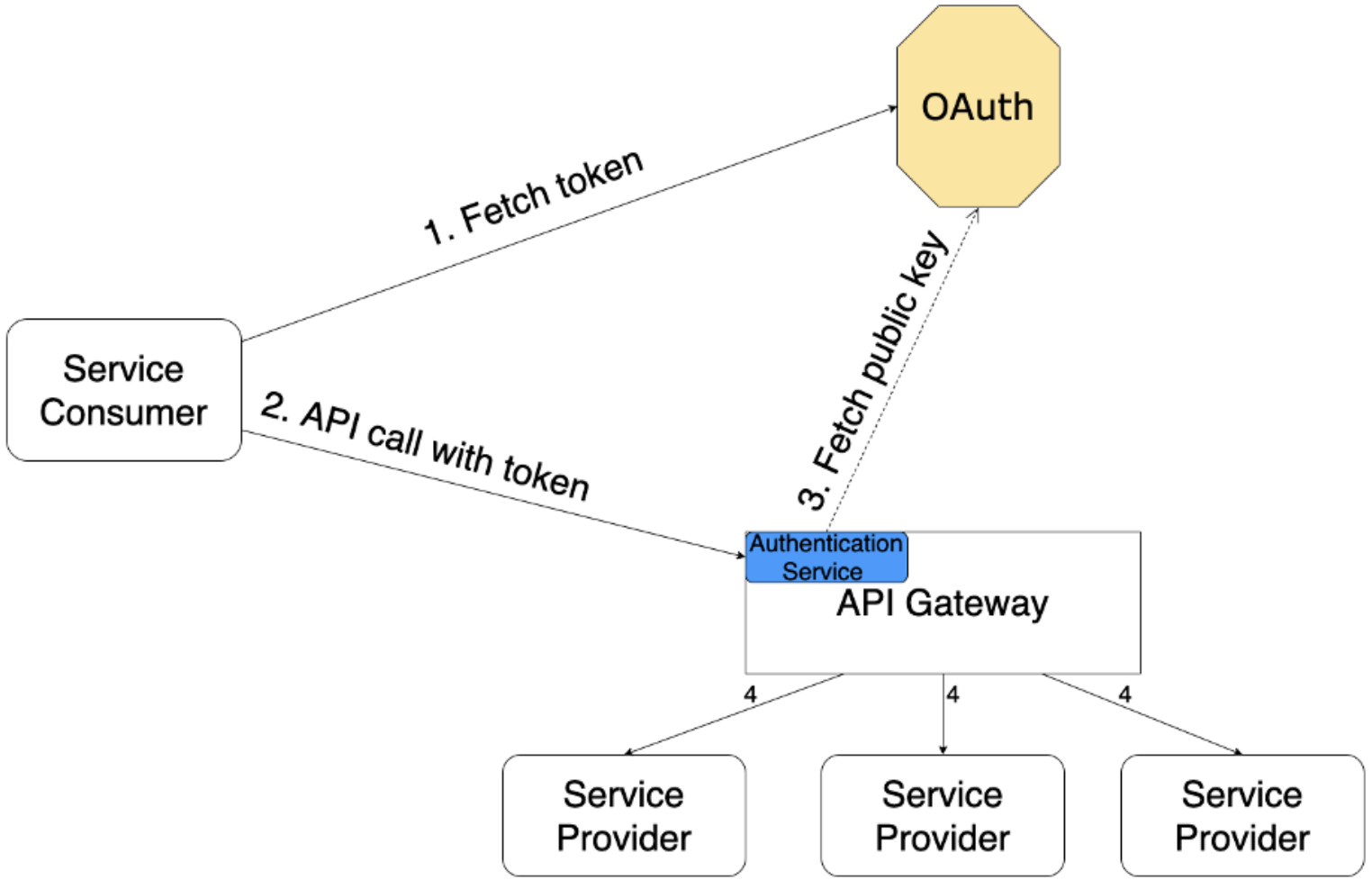
痛点：

1. 耦合性：Service高度依赖于authentication SDK
2. 复杂性：Service还需要关注鉴权和token管理
3. 复用性：Auth SDK异构语言治理
4. 可维护性：升级成本高



API网关鉴权

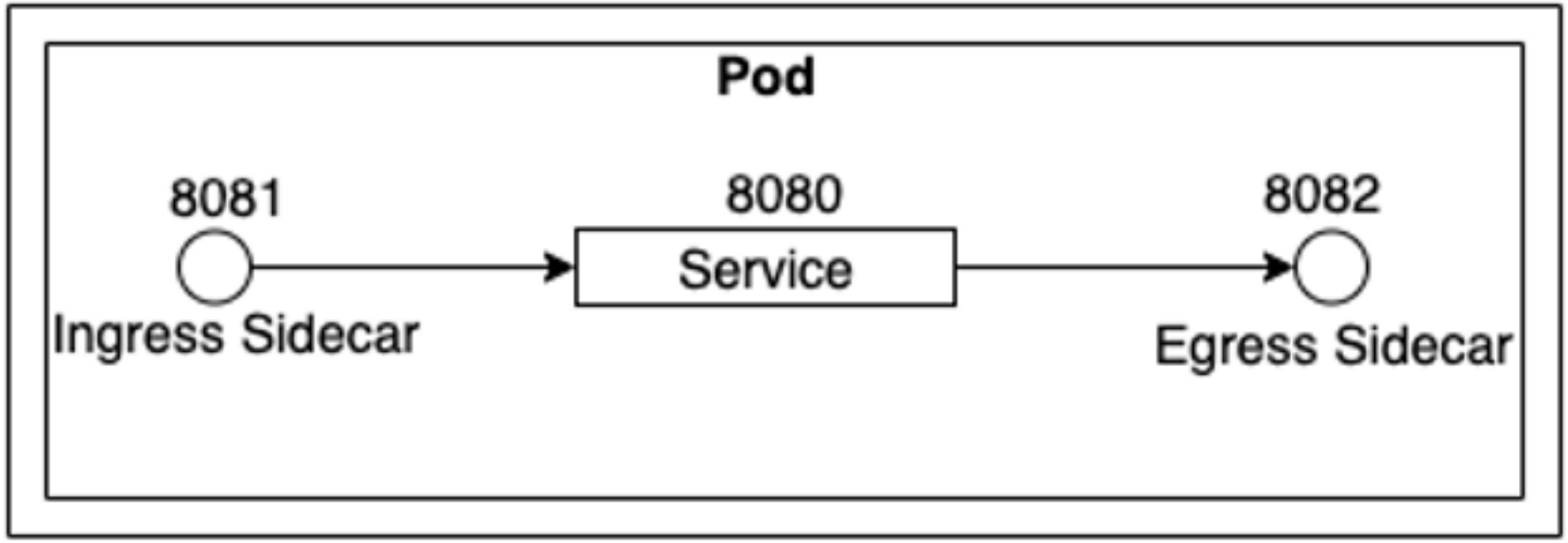
- Inbound Authentication
  （入站认证：校验令牌）



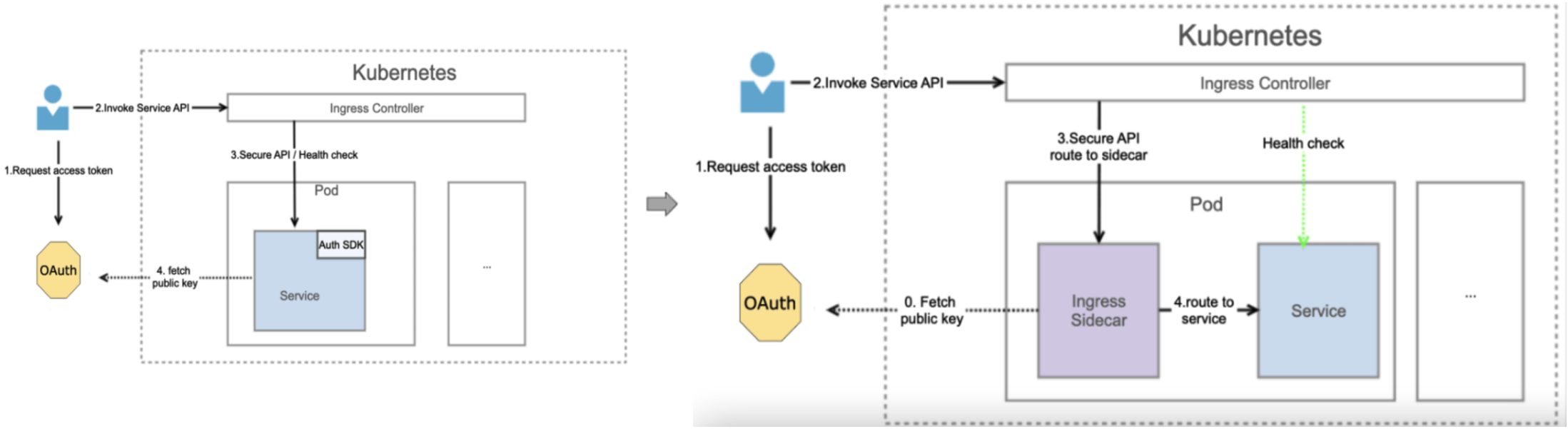
痛点：

1. API网关没有处理Outbound Authentication(出站获取令牌)
2. 零信任网络，永远不要信任网络并始终进行验证

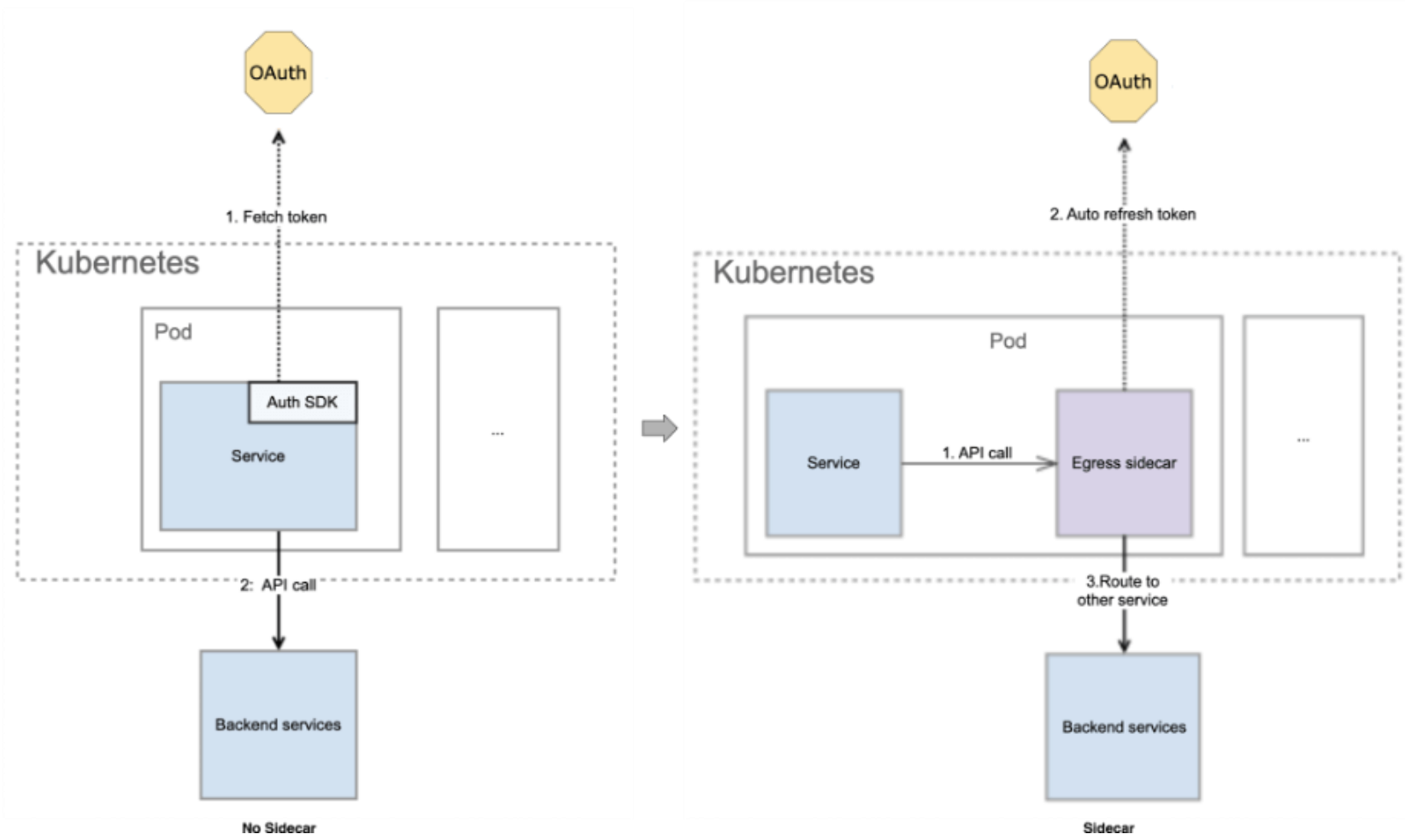
Authentication Sidecar 模式



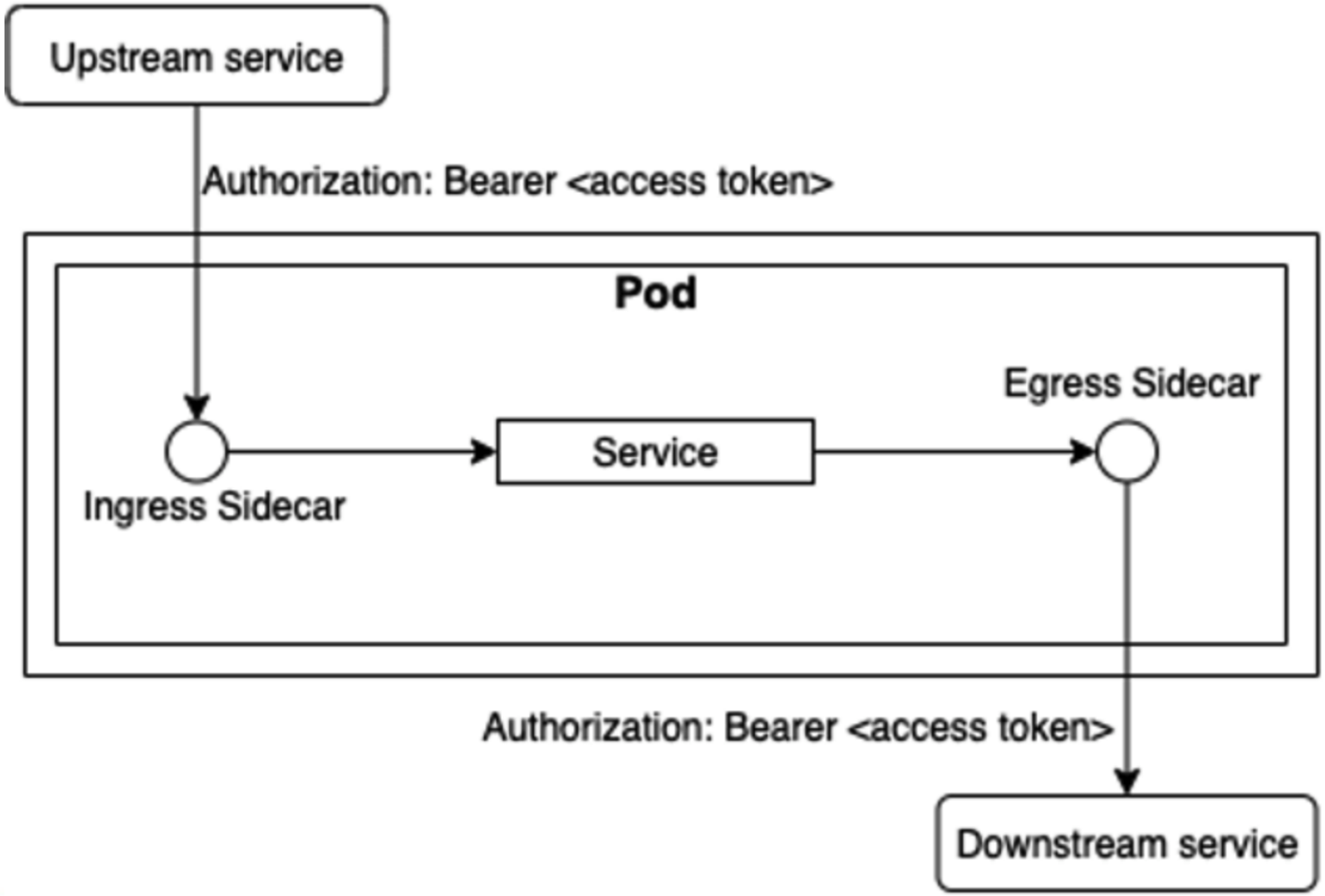
Inbound Authentication Sidecar



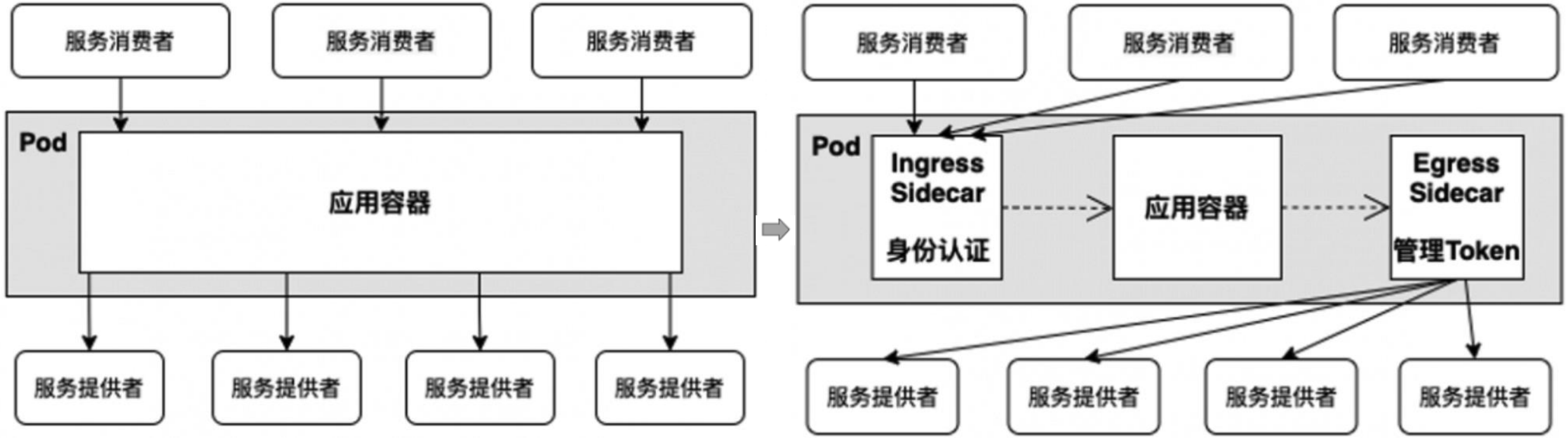
Outbound Authentication Sidecar



全貌


Upstream service
Authorization: Bearer <access token>
Pod
Egress Sidecar
Service
Ingress Sidecar
Authorization: Bearer <access token>
Downstream service

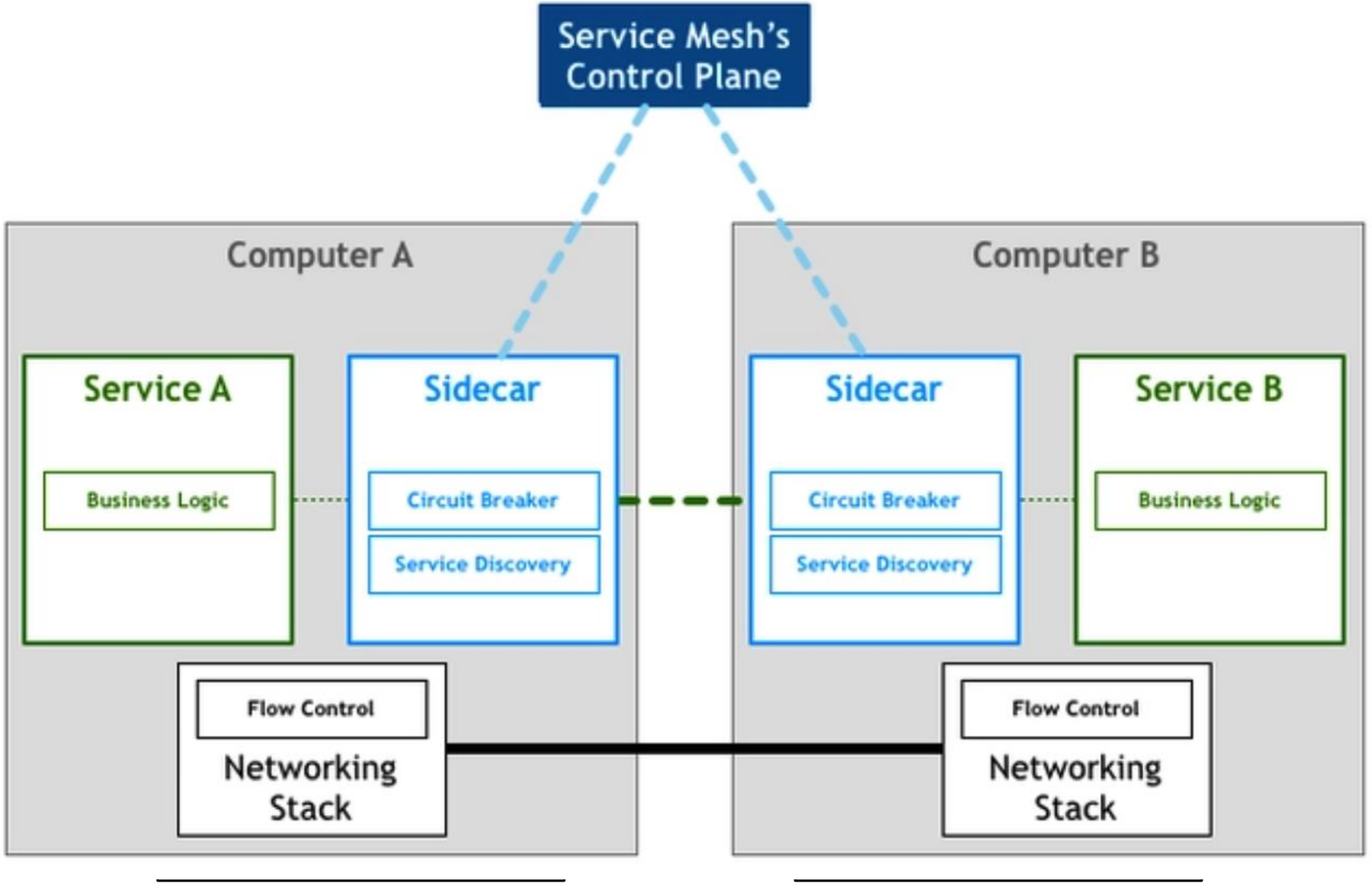
Authentication Sidecar的好处



1. 耦合性：解除了业务系统和authentication的耦合
2. 复杂性：降低应用系统的复杂性
3. 复用性：认证标准化，与编程语言无关
4. 可维护性：只有一个code base，易于升级

扩展

- 服务网格


Service Mesh's Control Plane
Computer A
Service A
Business Logic
Sidecar
Circuit Breaker
Service Discovery
Flow Control
Networking Stack
Computer B
Sidecar
Circuit Breaker
Service Discovery
Service B
Business Logic
Flow Control
Networking Stack

总结

方案	适用场景
系统自身鉴权	单体应用或各系统间没有太多服务通讯；需要快速完成的系统
API网关鉴权	所有的客户端和消费端都通过统一的网关接入微服务，只能进行Inbound Authentication
Authentication sidecar	企业内有很多分布式微服务并有容器化的部署；Service Mesh架构

演进：微服务实践支持

测试覆盖率

Infrastructure as code

Q & A

THANKS