Auth proxy pattern on k8s

Michał Wcisło

27.07.2019

Agenda

- OAuth2 and OpenID connect basics
- Introduction to Auth proxy on k8s
- Simple binary authorization scenario
- The way forward…

A few words about myself

- 8 years in Nokia
- Worked in telco research (VoIP, MIMO), QA, Technical Support and Development
- Currently working on development of Nokia AVA ecosystem, specifically k8s as a service

OAuth2.0

OAuth2.0

- Open standard for access delegation.
- OAuth1.0 2010, OAuth2 2012
- Should be used for **Authorization**
- Decoupling

Decoupling - components

Client
(app/service)

Client
Agent
(browser)

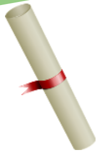
Resource
owner
(user)

Resource server
(service providing
information)

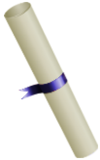
Authorization
Server

OAuth2.0 flows

The story...

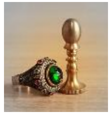


Auth code



Access token

Resource owner (user)



Secret (password)

User Agent (browser)

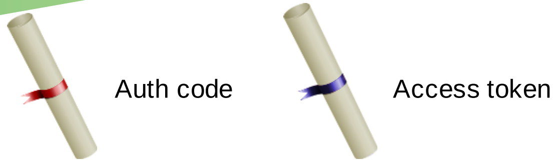
Authorization Server

Client

Secret (password)

Resource server

Authorization code flow



Auth code

Access token

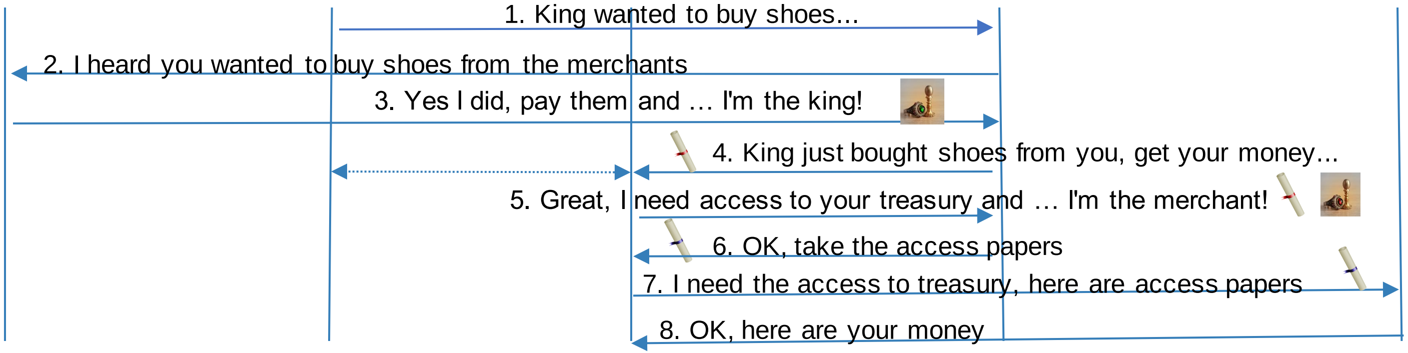
Resource owner (user)

User Agent (browser)

Client

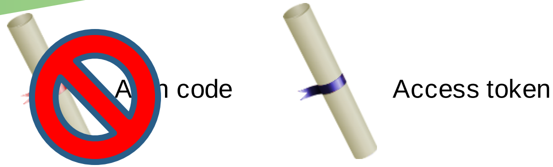
Authorization Server

Resource server



1. King wanted to buy shoes…
2. I heard you wanted to buy shoes from the merchants
3. Yes I did, pay them and … I'm the king!
4. King just bought shoes from you, get your money…
5. Great, I need access to your treasury and … I'm the merchant!
6. OK, take the access papers
7. I need the access to treasury, here are access papers
8. OK, here are your money

Implicit flow

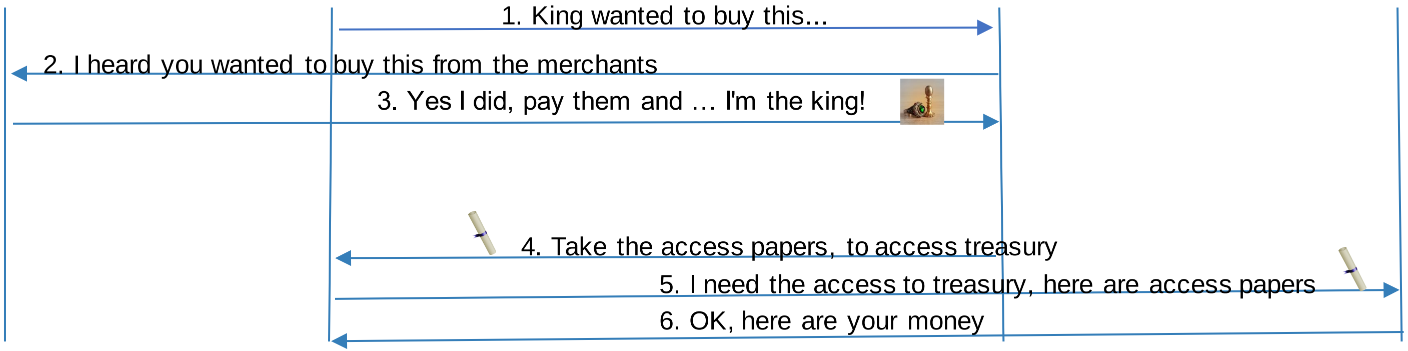


Resource owner (user)

Client (browser/js app)

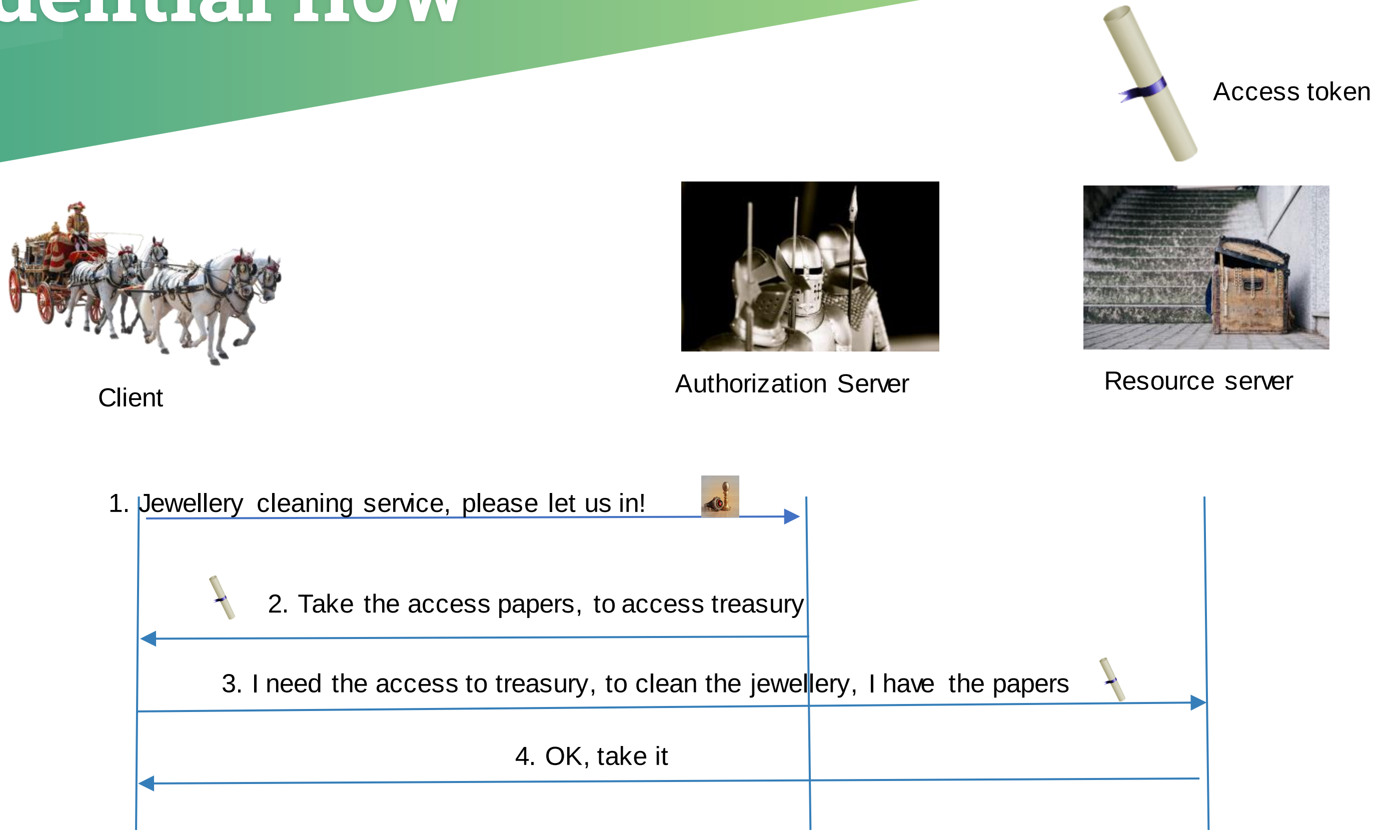
Authorization Server

Resource server



1. King wanted to buy this…
2. I heard you wanted to buy this from the merchants
3. Yes I did, pay them and … I'm the king!
4. Take the access papers, to access treasury
5. I need the access to treasury, here are access papers
6. OK, here are your money

Client credential flow



Access token

Client

Authorization Server

Resource server

1. Jewellery cleaning service, please let us in!

2. Take the access papers, to access treasury

3. I need the access to treasury, to clean the jewellery, I have the papers

4. OK, take it

OpenID connect

OpenID connect

- Build on top of OAuth2
- Released in 2014, as a standarization of different ways for using OAuth2 for AuthN
- Should be used for **Authentication**

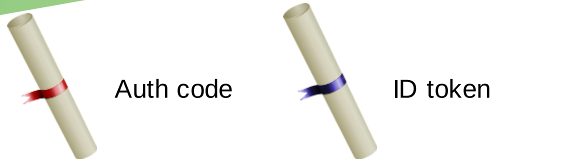
OpenID connect vs OAuth2

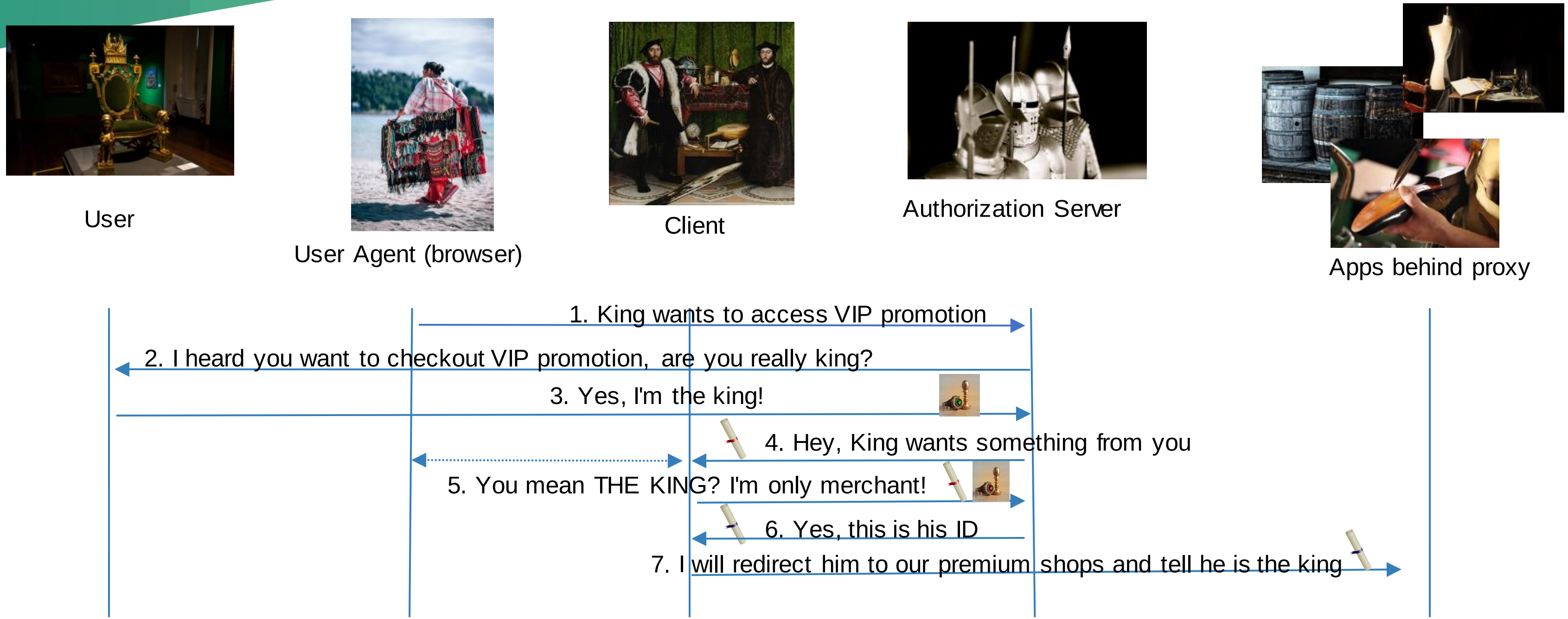
- User info becomes **resource**
- Authorization code, implicit and hybrid flows
- Additional parts for security – i.e. nonce
- Scopes and claims – openid scope
- ID token introduced

Client = relying party aka "who is scared the most"?

Auth proxy

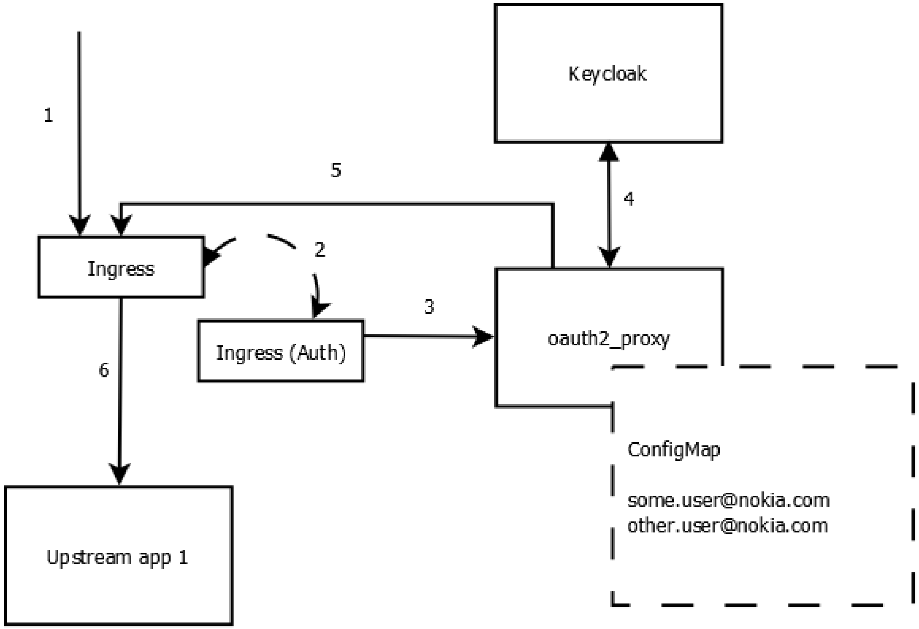
Authorization code flow





1. King wants to access VIP promotion
2. I heard you want to checkout VIP promotion, are you really king?
3. Yes, I'm the king!
4. Hey, King wants something from you
5. You mean THE KING? I'm only merchant!
6. Yes, this is his ID
7. I will redirect him to our premium shops and tell he is the king

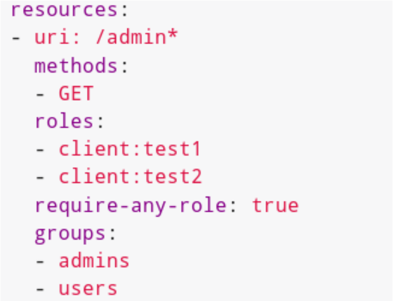
Running it on k8s + DEMO



Auth proxy beyond basics

Other Auth proxy implementations

- Keycloak-gatekeeper



```
resources:
- uri: /admin*
  methods:
  - GET
  roles:
  - client:test1
  - client:test2
  require-any-role: true
  groups:
  - admins
  - users
```

- Pomerium – zero-trust
- Buzzfeed/sso - double Auth proxy

K8s API access

- Configure kubectl to act as Auth proxy

```
kubectl config set-credentials USER_NAME \
  --auth-provider=oidc \
  --auth-provider-arg=idp-issuer-url=( issuer url ) \
  --auth-provider-arg=client-id=( your client id ) \
  --auth-provider-arg=client-secret=( your client secret ) \
  --auth-provider-arg=refresh-token=( your refresh token ) \
  --auth-provider-arg=idp-certificate-authority=( path to your ca certificate ) \
  --auth-provider-arg=id-token=( your id_token )
```

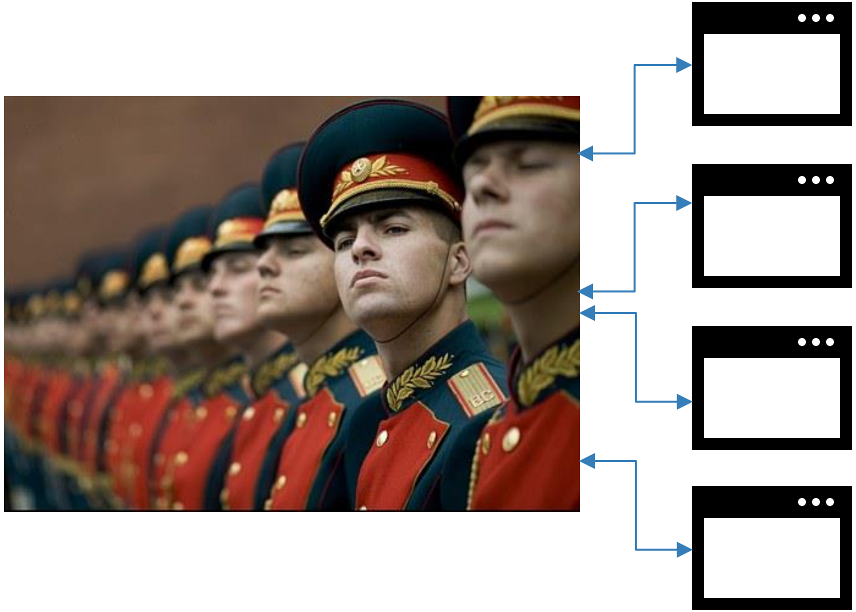
- Configure k8s API access with OpenID connect and Auth proxy

```
--oidc-issuer-url
--oidc-client-id
(--oidc-username-claim,  --oidc-groups-claim)
```

- Impersonation proxy: https://kccnceu19.sched.com/event/MPdT

Istio – access managment + DEMO

- Guards talk to each other in secret language (mTLS)
- Guards allow to talk to protected resource only when ordered (policy)
- Guards needs assistance to recognize (authenticate) people from outside world > Auth proxy



Summary

- OAuth2 and OpenID connect are modern standards for AuthZ and AuthN
- Auth proxy allows easily incorporating them with your applications
- K8s ingress controller can be used to dynamically define Auth proxy redirects
- Istio with Auth proxy enables fine-grained access managment

References

- OAuth 2.0 Threat Model and Security Considerations
- Security Best Current Practice
- Impersonation proxy talk KubeCon
- All configuration files from this presentation are published on https://github.com/m-wcislo/talks

Thank you!

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash