



# Portable, Universal, Single Sign-On for Your Clusters

Miguel Martinez @migmartri  
May 21st, Kubecon EU

# Hi, I am Miguel!

- Spaniard in San Francisco, obsessed with Mexican food
- Full-stack developer at Bitnami
- Core contributor of Kubeapps and Monocular
- Emeritus core maintainer in Helm



@migmartri

# Our problem

## Support Single Sign-on in Kubeapps

- Only supported service accounts
- Adoption barrier
- Best practices blocker, RBAC is hard



**Single sign-on, most requested feature**



Made with ❤️ by Bitnami and contributors.



v13.1

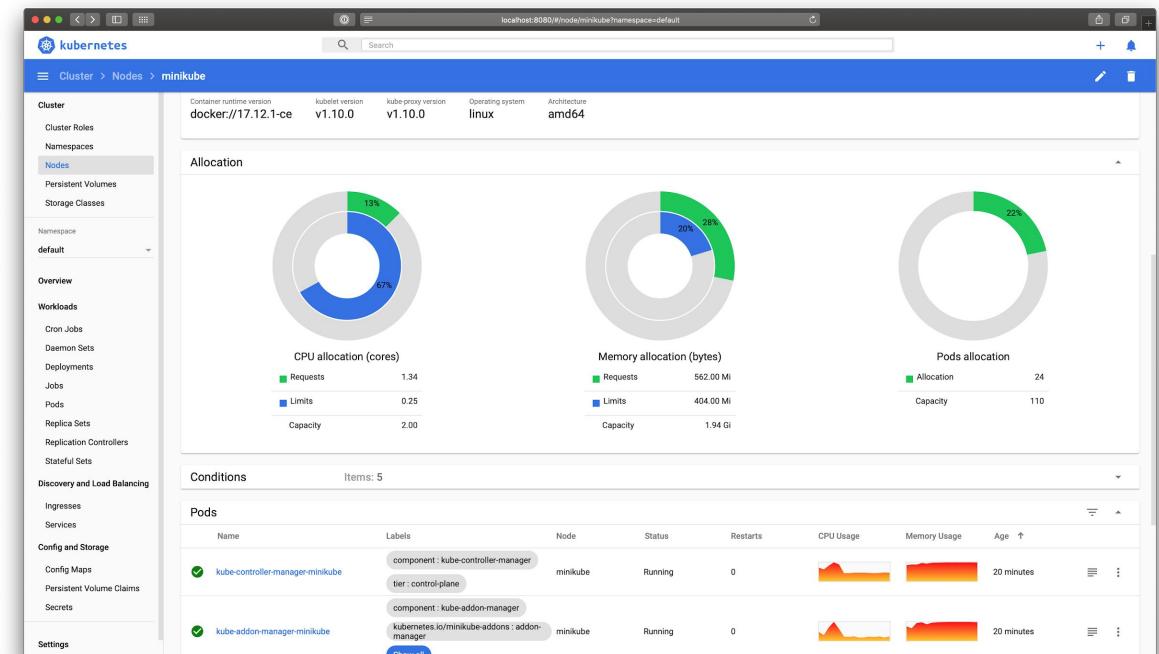
# Our problem

## General Statement

Application (YourApp) that

1. AuthN users via Single sign-on
2. **Talks to the k8s API server**

i.e kubectl, Kubernetes Dashboard, Kubeapps



```
migmartri ~ $ kubectl cluster-info
Kubernetes master is running at https://192.168.99.110:8443
KubeDNS is running at https://192.168.99.110:8443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

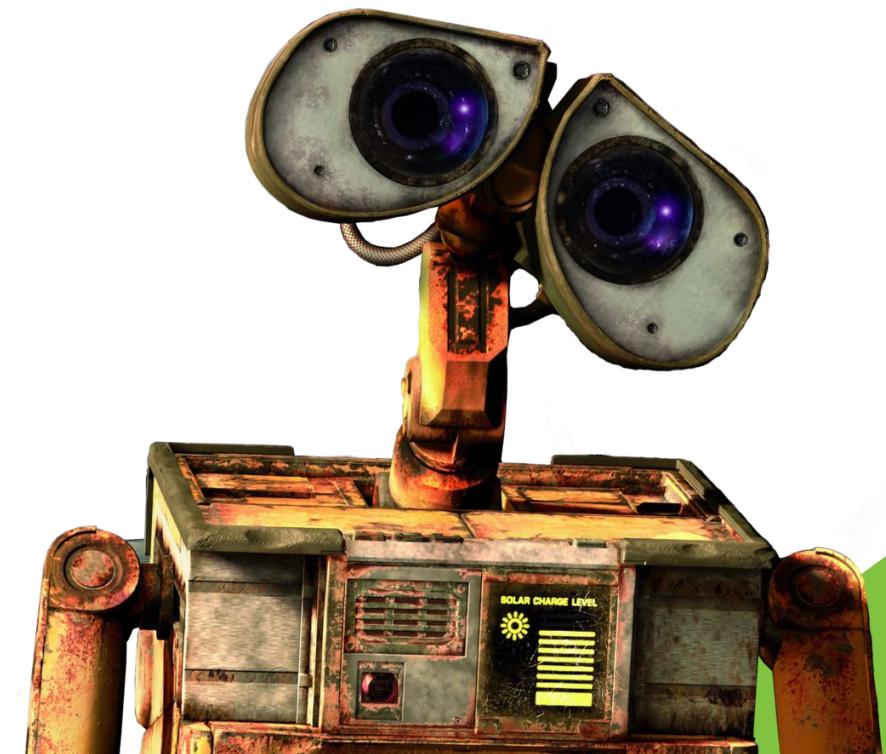
# Our problem

## Solution Scope and Caveats

AuthN vs ~~AuthZ~~

~~Robots~~ vs Humans

Platform ~~dependent~~ vs  
Independent



# AuthN in Kubernetes

## User Authentication Overview

	Self-serve	Rotation	Revocation	UX
X509 Client Certs				
Token (SA or Static)				
Basic Auth				
Single sign-on - OpenID Connect				

# SSO in Kubernetes

## Why?

### For Users

- Familiar AuthN mechanism
- No need to have additional set of credentials
- Self-serve

### For Cluster Operators

- No manual generation or transfer of credentials
- Built-in rotation and revocation methods
- AuthN delegation
- Support for groups and scopes

# SSO in Kubernetes

Kubernetes API understands  
OpenID Connect (OIDC)

OAuth2 != OIDC!

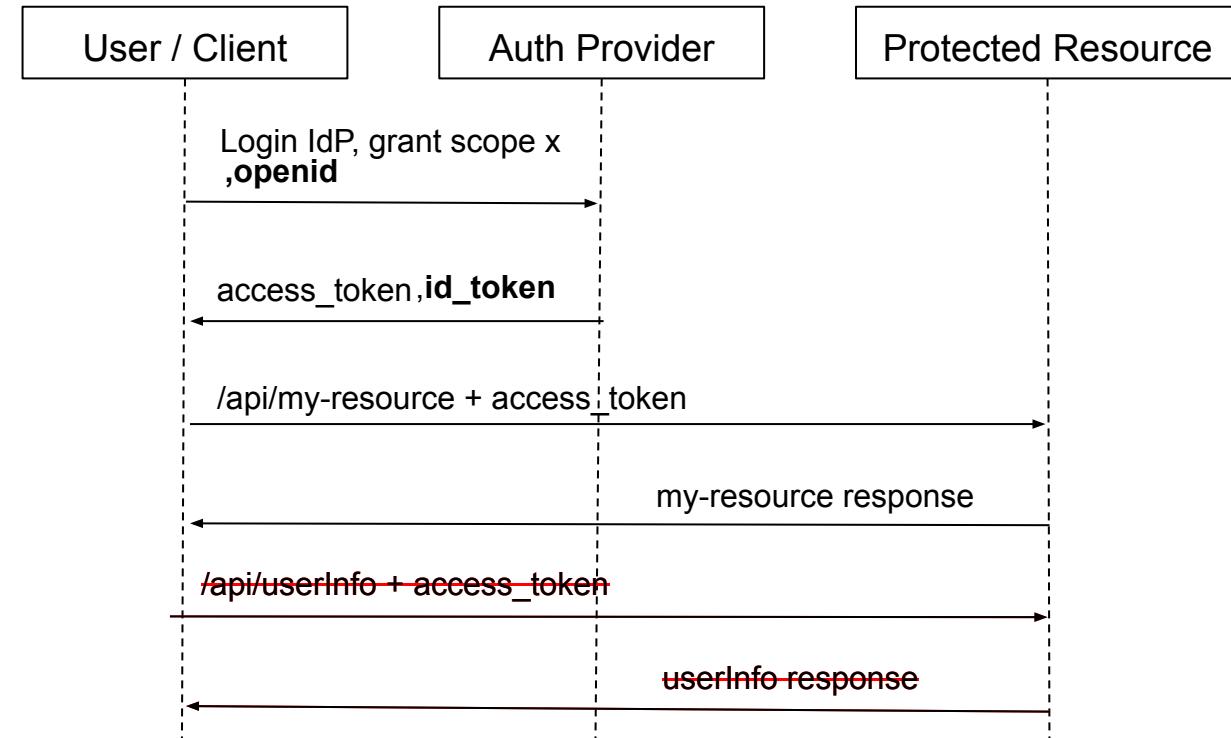


# SSO in Kubernetes

## OpenID Connect (OIDC)

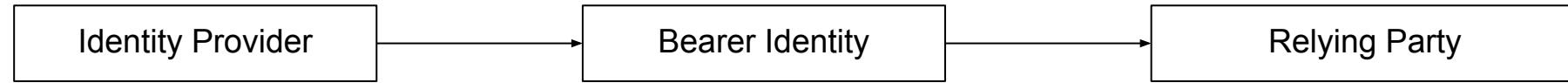
Identity layer **on top of** the OAuth 2.0 protocol

Authentication Info **standardized** in a **cryptographically signed JWT token** called id\_token



# SSO in Kubernetes

## OpenID Connect - Trust Chain In Real Life



- Verify Identity
- Craft Passport

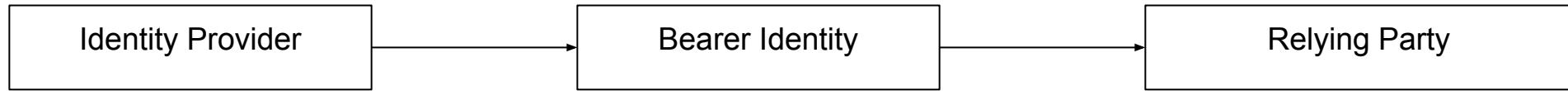


- It is from a trusted source
- It has not been tampered with (ePassport)
- It is not expired

**The relying party does not contact the identity provider**

# SSO in Kubernetes

## OpenID Connect - Trust Chain in Kubernetes



ID Token (JWT)

Relying Party



K8s API Server

3rd party app



- Header
- Payload
  - Identity (sub/email)
  - Who provisioned this token (**iss**)
  - Intended client audience (**aud**)
  - Expiration time (**exp**)
  - Claims (name, email, ...)
- Signature

- It has not been tampered with (**signature**)
- It is from a trusted source (**iss**)
- I am the receiver (**aud**)
- It is not expired (**exp**)
- **DOES NOT contact the IdP (except to retrieve the Public Key)**



# SSO in Kubernetes

## Integration

You need to configure the K8s API server  
to trust an OIDC Identity Provider

```
# API server flags

--oidc-issuer-url https://my-oidc-idP.com # .../.well-known/openid-configuration
--oidc-client-id my-client-id
--oidc-username-claim email
--oidc-groups-claim groups

# oidc-issuer.match(id_token.iss) && oidc-client-id.match(id_token.aud)
```

```
$ curl https://api-server -H "Authorization: Bearer ${id_token}"
$ kubectl --token ${id_token}
```

# Problem Statement

## Summary

Application (YourApp) that:

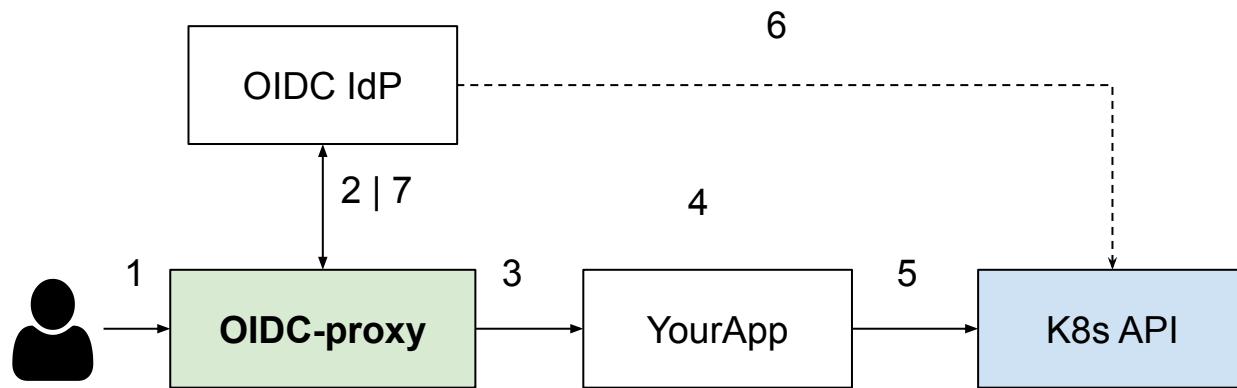
1. AuthN users via OIDC single sign-on
2. Talks directly to the k8s API server

i.e kubectl, Kubernetes Dashboard, Kubeapps

# Problem Statement

## Solution

Proxy configured with the **same OIDC IdP than the k8s API server** \*

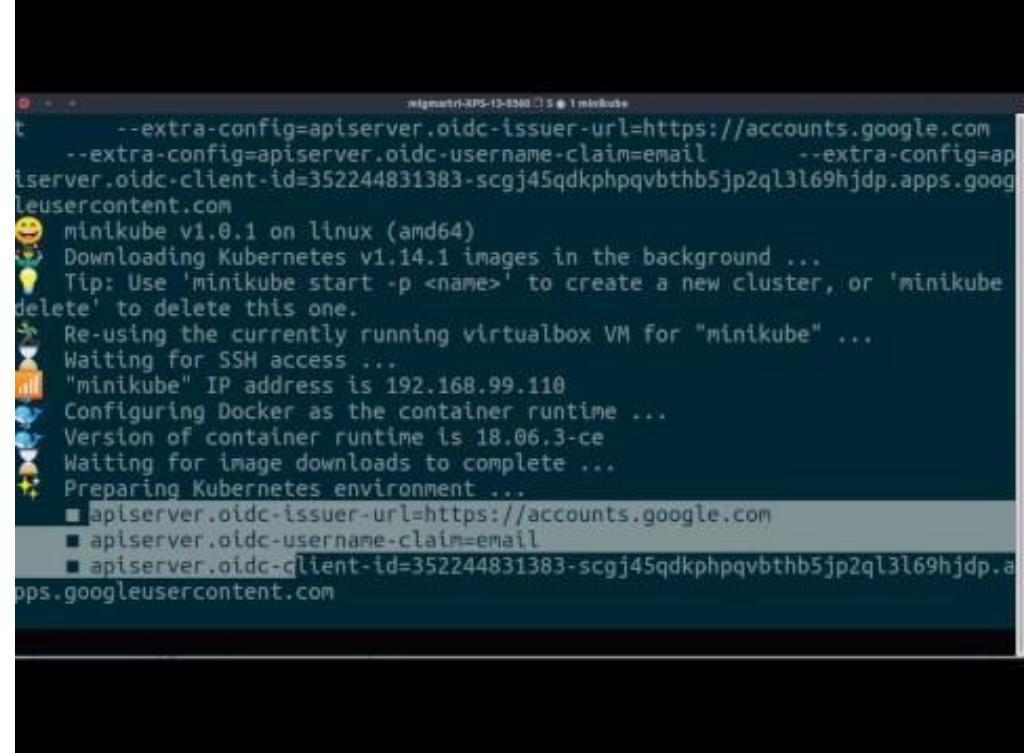


- Enforce AuthN with an external IdP
- Takes care of the OAuth2 dance, token exchange and refresh
- Inject ID Headers and forward them upstream

# Problem Statement

## Demo, Exclamation Mark

SSO-enabled Kubernetes Dashboard  
+ Google's IdP on Minikube



```
minikube start --extra-config=apiserver.oidc-issuer-url=https://accounts.google.com
--extra-config=apiserver.oidc-username-claim=email --extra-config=apiserver.oidc-client-id=352244831383-scgj45qdkphpqvbtb5jp2ql3l69hjdp.apps.googleusercontent.com
minikube v1.0.1 on linux (amd64)
Downloading Kubernetes v1.14.1 images in the background ...
Tip: Use 'minikube start -p <name>' to create a new cluster, or 'minikube delete' to delete this one.
Re-using the currently running virtualbox VM for "minikube" ...
Waiting for SSH access ...
"minikube" IP address is 192.168.99.110
Configuring Docker as the container runtime ...
Version of container runtime is 18.06.3-ce
Waiting for image downloads to complete ...
Preparing Kubernetes environment ...
apiserver.oidc-issuer-url=https://accounts.google.com
apiserver.oidc-username-claim=email
apiserver.oidc-client-id=352244831383-scgj45qdkphpqvbtb5jp2ql3l69hjdp.apps.googleusercontent.com
```

# Problem Statement

We Are Not Done Yet

**The solution does not work in all platforms**

- K8s provider API server lockdown
- Ops do not want OIDC enabled in k8s API
- IdP or authN requirements mismatch (LDAP)
- IdP groups/user claim support



# Problem Statement

## Challenge 1: K8s API Server OIDC Customisation

Kubernetes Distro	API Server OIDC Customization
GKE (Google)	No
EKS (AWS)	No
AKS (Azure)	Active Directory
OKE (Oracle)	Oracle Identity Cloud Service
Minikube	Any
Kops	Any
kubeadm	Any

# Problem Statement

## Challenge 2: Identity Providers and Group Claims

```
subjects:  
- kind: Group  
  name: "kubeapps:developer"  
  apiGroup: rbac.authorization.k8s.io
```

```
"session_state": "eedbf6d0-950a-40af-a14e-be840775285f",  
"acr": "1",  
"email_verified": false,  
"groups": [  
  "kubeapps:developer"  
],  
"preferred_username": "keycloak"  
}
```

```
--oidc-groups-claim "groups" # API flag
```

OIDC Identity Provider	Group Claims Support
Okta	
Dex	Depends on Upstream
Keycloak	
Active Directory	
Google Accounts	
...	...

“

Easy things should be easy, and hard  
things should be possible.

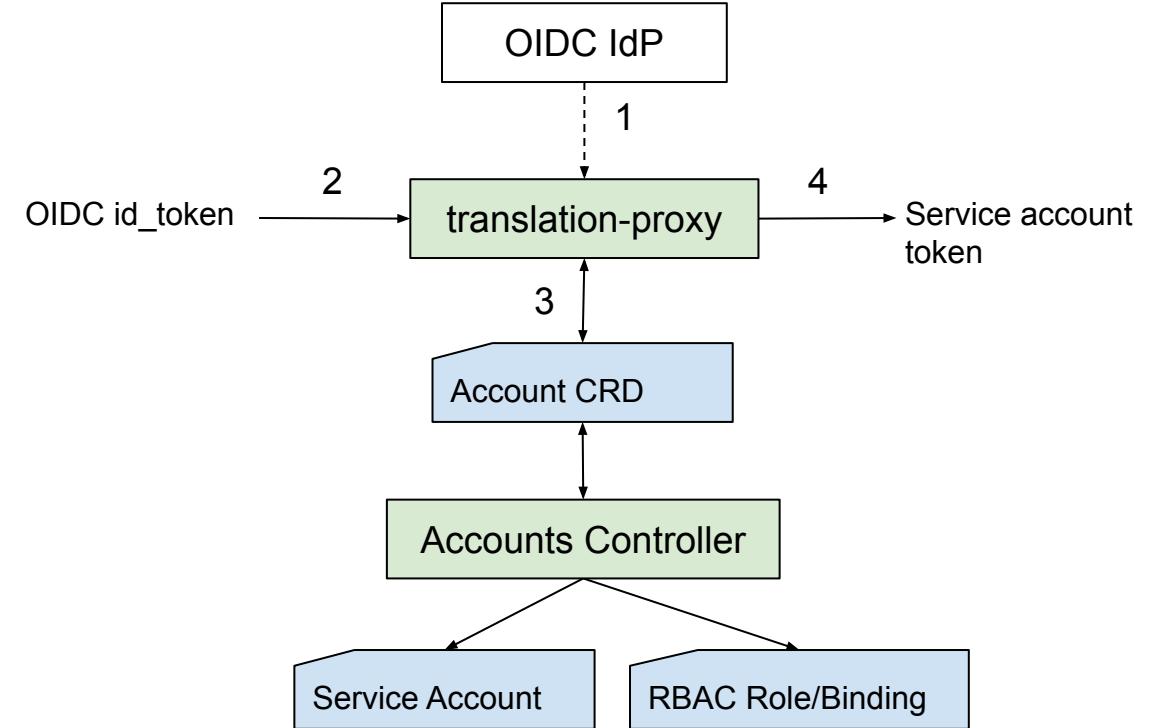
”

Larry Wall

# Workaround 1

## Translation to Service Accounts

**Translate OIDC `id_tokens` into service accounts via a translation proxy and a custom controller**



# Workaround 2

## Kubernetes Impersonation

```
$ kubectl get pods --as FooUser --as-group kubeapps-user  
  
$ curl https://api-server/api/v1/pods \  
-H "Authorization: Bearer ${impersonator token}" \  
-H "Impersonate-User: FooUser" \  
-H "Impersonate-Group: kubeapps-user"
```

Can ImpersonatorUser  
impersonate FooUser?

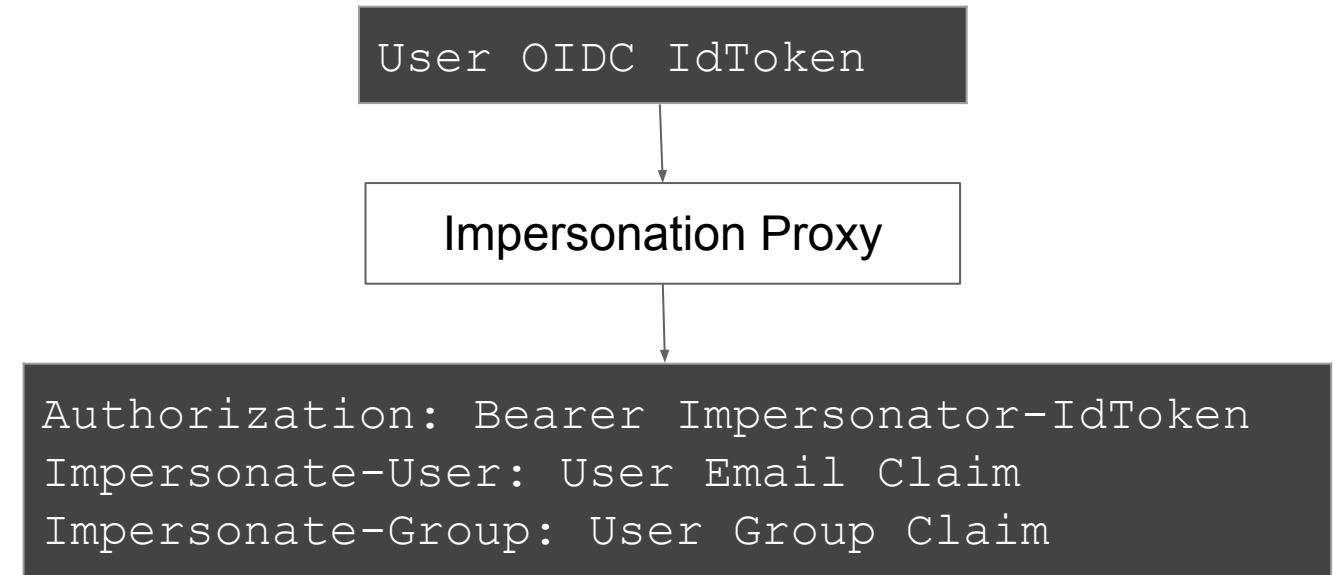
Can FooUser access  
pods in namespace x?

```
- apiGroups: [""]
resources: ["groups"]
verbs: ["impersonate"]
resourceNames:
["developers", "kubeapps-user"]
```

# Workaround 2

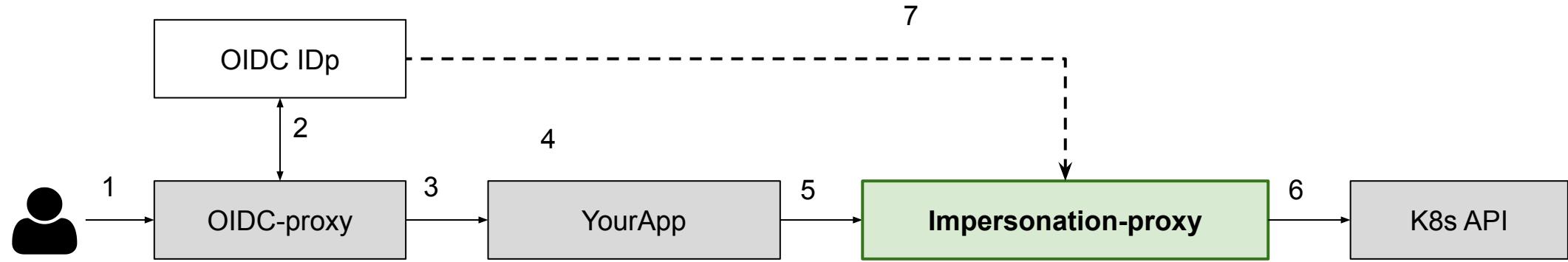
## Kubernetes Impersonation

Proxy in charge of impersonating users and groups based on OIDC id\_token claims



# Workaround 2

## Kubernetes Impersonation

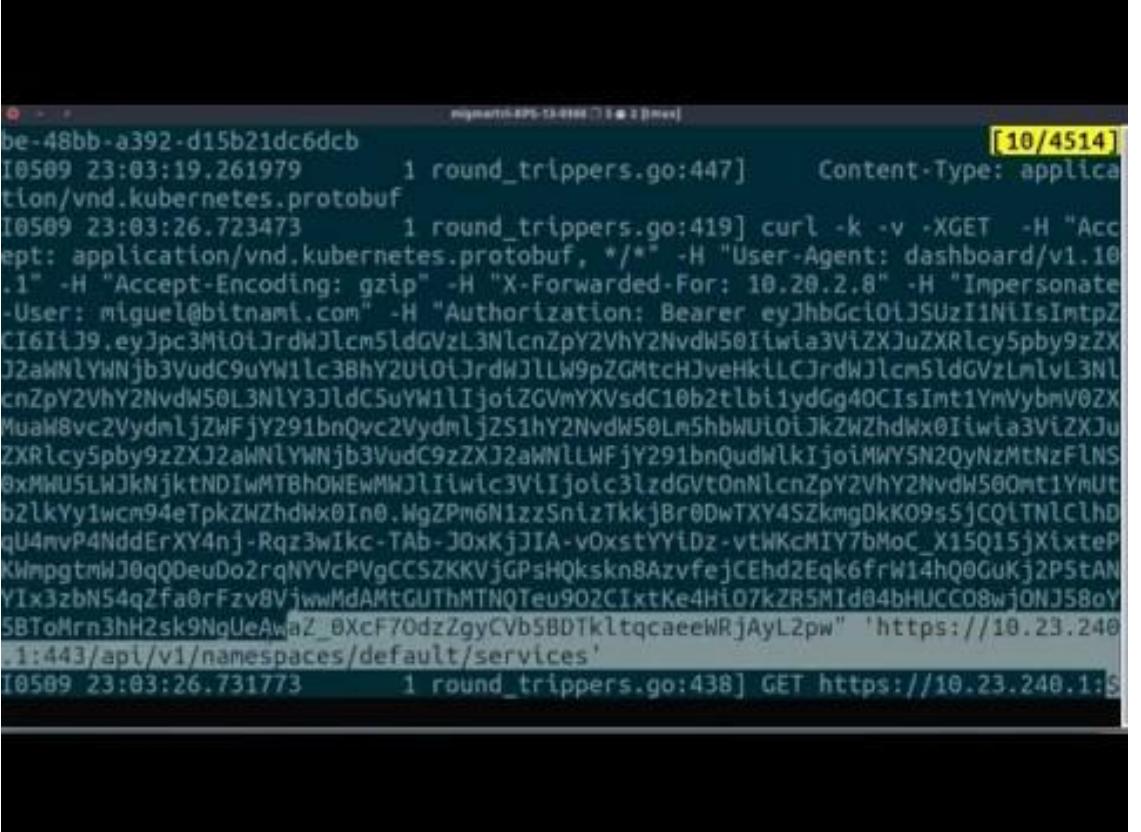


- Extracts OIDC verification logic from API server
- Prevent stale credentials
- Fewer moving pieces
- Single leak source

# Workarounds

## Kubernetes Impersonation

SSO-enabled Kubernetes Dashboard + Google's IdP on **GKE\*** using impersonation



```
be-48bb-a392-d15b21dc6dcb [10/4514]
I0509 23:03:19.261979 1 round_trippers.go:447] Content-Type: application/vnd.kubernetes.protobuf
I0509 23:03:26.723473 1 round_trippers.go:419] curl -k -v -XGET -H "Accept: application/vnd.kubernetes.protobuf, */*" -H "User-Agent: dashboard/v1.10.1" -H "Accept-Encoding: gzip" -H "X-Forwarded-For: 10.20.2.8" -H "Impersonate-User: miguel@bitnami.com" -H "Authorization: Bearer eyJhbGciOiJSUzIiNiIsImtpZC16ItJ9.eyJpc3MiOiJrdWJlcms5ldGVzL3NlcnZpY2VhY2NvdW50Iiwi3ViZXJuZXRlcyc5pbv9zZXJ2aWNlyWNjb3VudC9uYW1lc3BhY2UloJrdWJLLW9pZGMtcHJveHkIlCJrdWJlcms5ldGVzLmlvL3NLcnZpY2VhY2NvdW50L3NlY3JldC5uYW1lIjoizGVmYXVsdc10b2tlbi1ydGg4OCIsInt1YmVybmV0ZXMuam8vc2VydmljZWfjY291bnQvc2Vydmljzs1hY2NvdW50Lm5hbWUloJk2WZhdNx0Iiwlta3ViZXJuZXRlcyc5pbv9zZXJ2aWNlyWNjb3VudC9zZXJ2aWNllWFjY291bnQudwIkIjoimWY5N20yNzMtNzFlNS0xMmU5LWJKNjktNDIwMTBh0WEWMJlIiwiC3VLIjoic3lzdGVtOnNlcnZpY2VhY2NvdW500mt1YmUt b2lkYy1wcm94eTpkZmZhdNx0In0.HgZPm6N1zzSnizTkkjBr0DwTXY45ZkmgDkK09ssjCQLTNlClhd qU4mvP4NddErXY4nj-Rqz3wIkC-TAb-J0xKjJIA-v0xstYYiDz-vtWKcMIY7bMoC_X15Q15jXixtePKWmpgtmWJ0qQDeuDo2rqNYVcPVgCCSZKKVjGPshQkskn8AzvfejCEhd2Eqk6frW14hQ0GuKj2P5tAN YIx3zbN54qZfa0rFzv8VjwwMdAMtGUTHMTNQTeu902CIxtKe4HiD7kZRS5MId84bHUCC08wjONJ58oY SBT0Mrn3hH2sk9NgUeAwaZ_8Xcf70dzZgyCVb5BDTklcqaeewRjAyL2pw" 'https://10.23.240.1:443/api/v1/namespaces/default/services'
I0509 23:03:26.731773 1 round_trippers.go:438] GET https://10.23.240.1:443/api/v1/namespaces/default/services
```

\* API server not configurable

**SSO in Kubernetes can be `vendor locked`  
but we can `workaround` it and offer a  
Universal, Cross-Platform SSO  
experience**

# Resources

- Kubeapps SSO in-depth document - <https://bit.ly/30bi1zF>
- SSO for Kubernetes talk by Joel Speed - <https://bit.ly/2Hh6kQN>
- kube-oidc-proxy - @jetstack - <https://bit.ly/2Vip6uw>
- Demo files repository - <https://bit.ly/2HfV9GI>
- This slide deck - <https://bit.ly/2WD8YoT>



@migmartri

Thank You