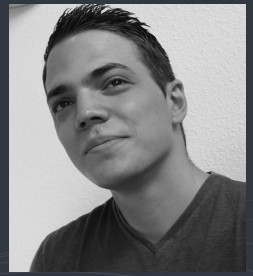
L’authentification et l’autorisation dans une architecture microservices ? Pas de soucis !



f.garcia@betclicgroup.com

@im_flog

- Backend lover
- Learner

- Tracing distribué
- Kafka
- Cloud

vmaleze@ippon.fr

@vmaleze

- Architecte
- Tech friendly

- Spring
- AWS
- DevOps

1h de sécurité ...

On y va pas à pas :

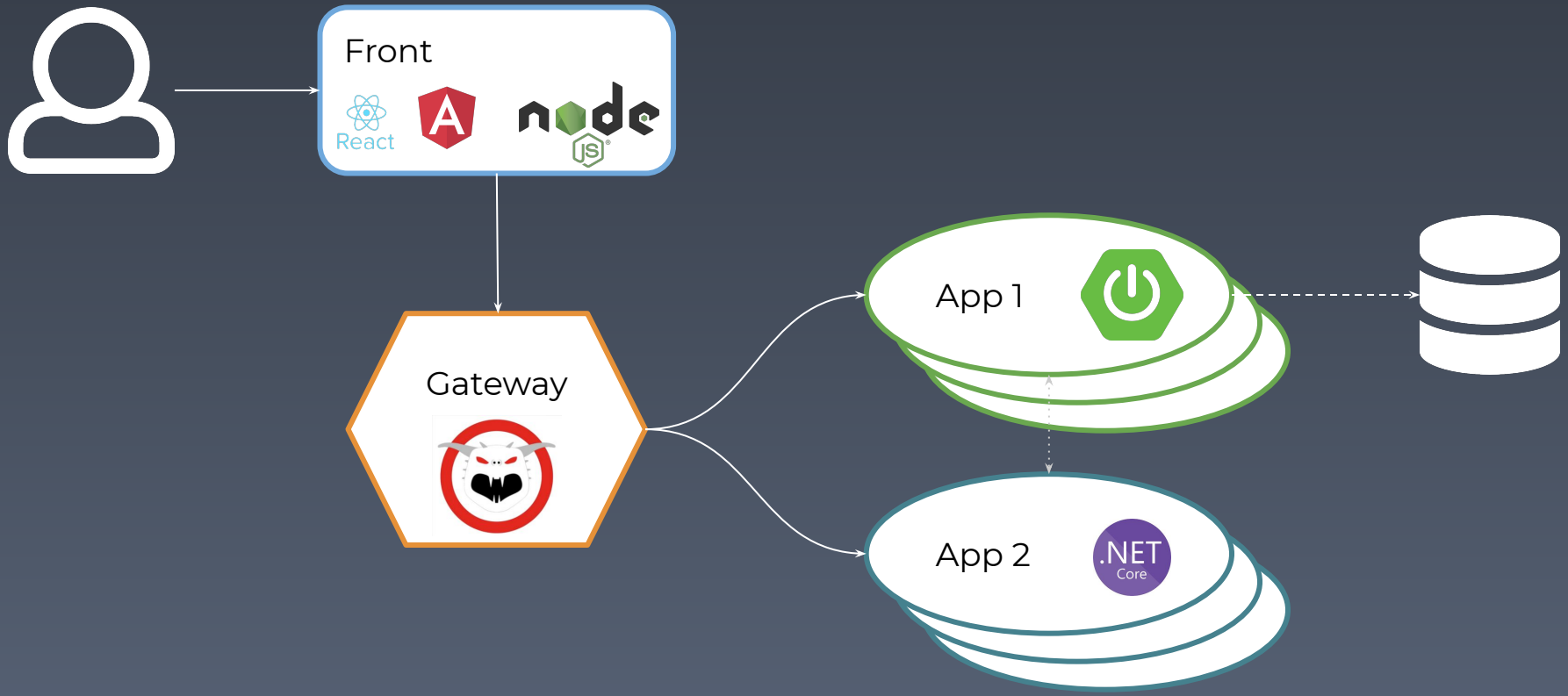
1. Les besoins du client
2. Les différentes solutions
3. Comment on met ça en place ?

Les thèmes :

- Authentification
- Autorisation
- Oauth2
- JWT
- Sessions
- Le code ...

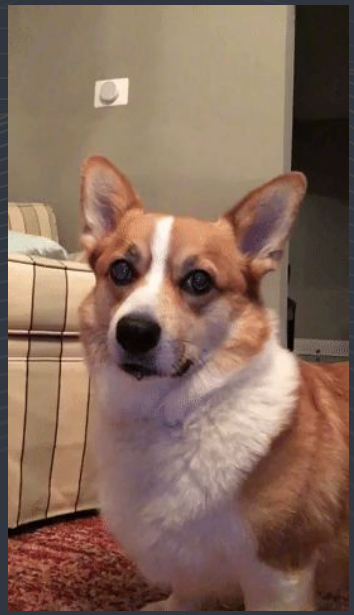
L'architecture

privalia



Le(s) problème(s)

- L’authentification ?
- L’autorisation ?
- Vraiment nécessaire ?
- C’est pas trop compliqué ?
  - LDAP / Services externes
  - SSO
  - Plusieurs langages de programmation



Serveur d'identité

Single Sign On

Fédération d'identité (LDAP / Custom DB)

Social login (Google)

Gestion de rôles

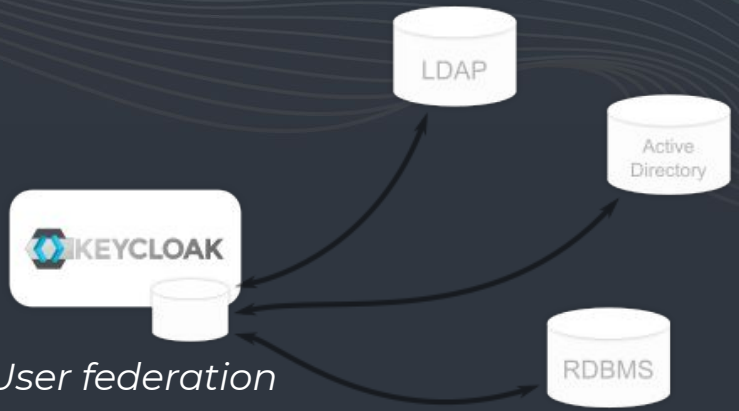
Facilité d'administration

Intégration Spring, Dotnet, NodeJs

Coûts (maintenance / Saas)

SSO

Identity Brokering



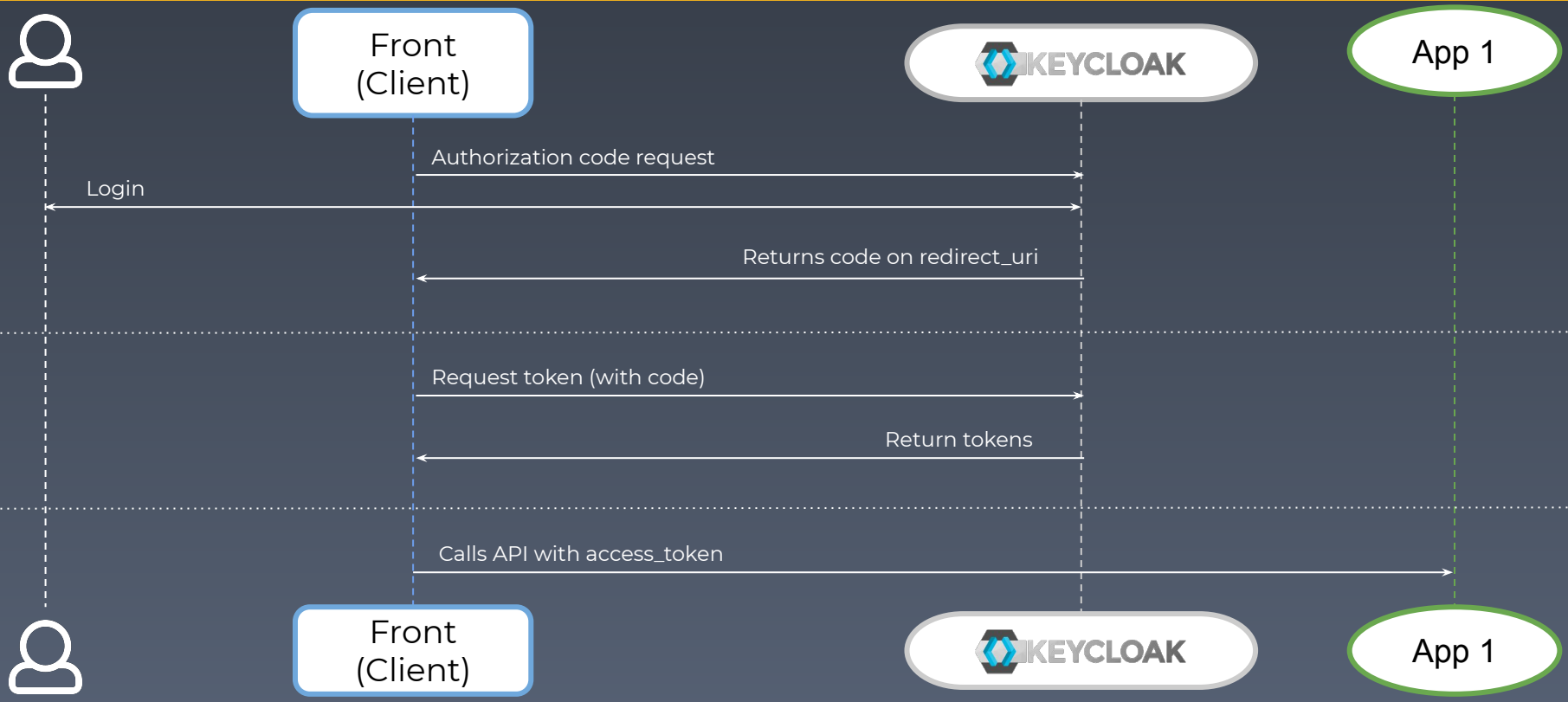
User federation

Standard protocols

L’authentification

Basic	OpenID	SAML
 Simplicité	 OAuth2	 Identity provider
 Base 64	 SSO	 SSO
 Intégration avec tiers	 JWTs	 Mature
	 “Nouveau” standard	 Verbeux
	 Complexe	 Complexe
	 HTTPS	 “Ancien” Standard

Oauth2


Front (Client)
KEYCLOAK
App 1
Authorization code request
Login
Returns code on redirect_uri
Request token (with code)
Return tokens
Calls API with access_token
Front (Client)
KEYCLOAK
App 1

OpenId en action

https://github.com/ImFlog/microservices-security

Les tokens

Les différents tokens

- id
- access
- refresh

Signés pour assurer l'intégrité

Validations

- Clé publique
- Endpoint Keycloak
- Secret

Gestion de sessions

Côté front
- Local storage
- Cookie

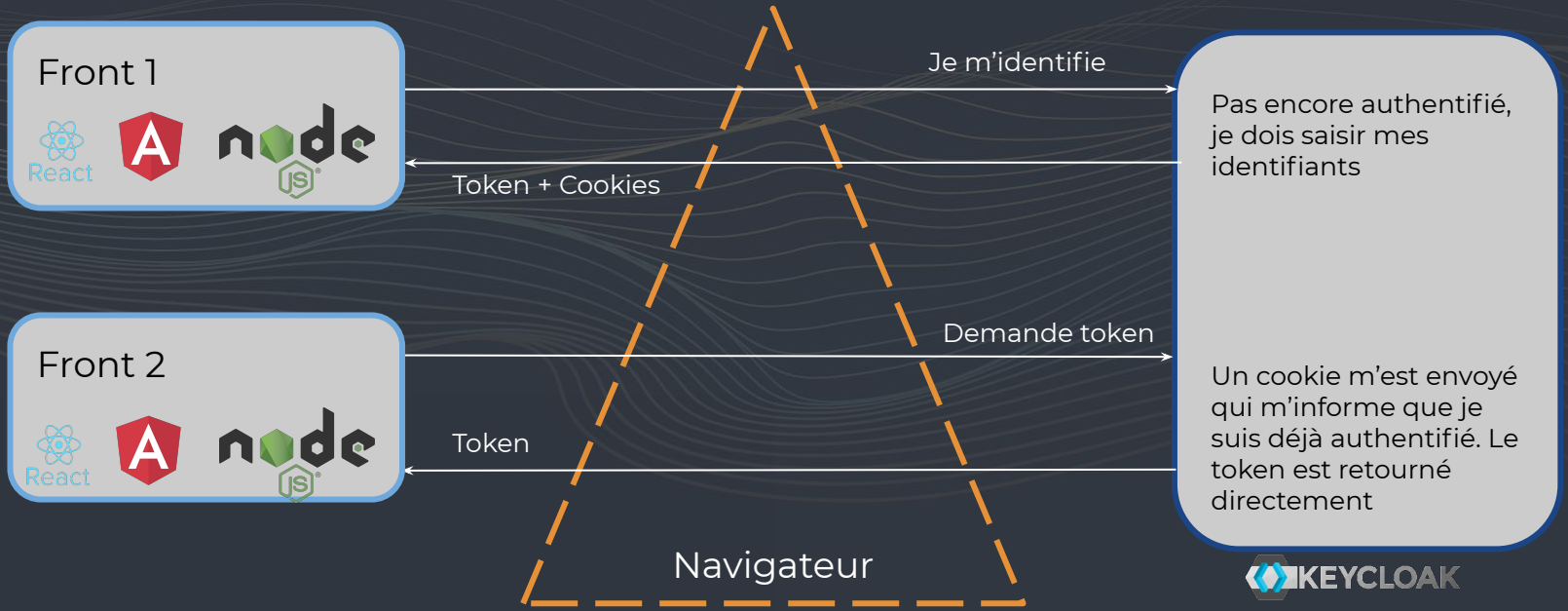
Stateless côté back !

Durée de vie des sessions
- *Short living token*
- *Long living refresh token*

Invalidation :
- Token introspection à chaque appel
- Révocation de sessions

Attention au contenu de vos sessions !

Web SSO


Front 1
React
node
Front 2
React
node
Je m'identifie
Token + Cookies
Demande token
Token
Navigateur
Pas encore authentifié, je dois saisir mes identifiants
Un cookie m'est envoyé qui m'informe que je suis déjà authentifié. Le token est retourné directement
KEYCLOAK

Frontend

https://github.com/ImFlog/microservices-security

Gestion des utilisateurs

Utilisateurs / Groupes / Rôles

Peuplé par LDAP / IDP / Services externes

Attribution rôles

- En fonction d'attributs de l'utilisateur
- Selon des règles (Javascript / Drools)
- Au niveau group ou user

Tout est lié à un royaume

L’autorisation côté back

On valide le token JWT

- Signature, expiration, issuer
- Avec la clé publique du serveur d’authentification (cachable)
- Techno agnostique

Rôle dans le JWT

Attribution d’un contexte de requête en fonction de l’appel

On sécurise route par route pour plus de granularité

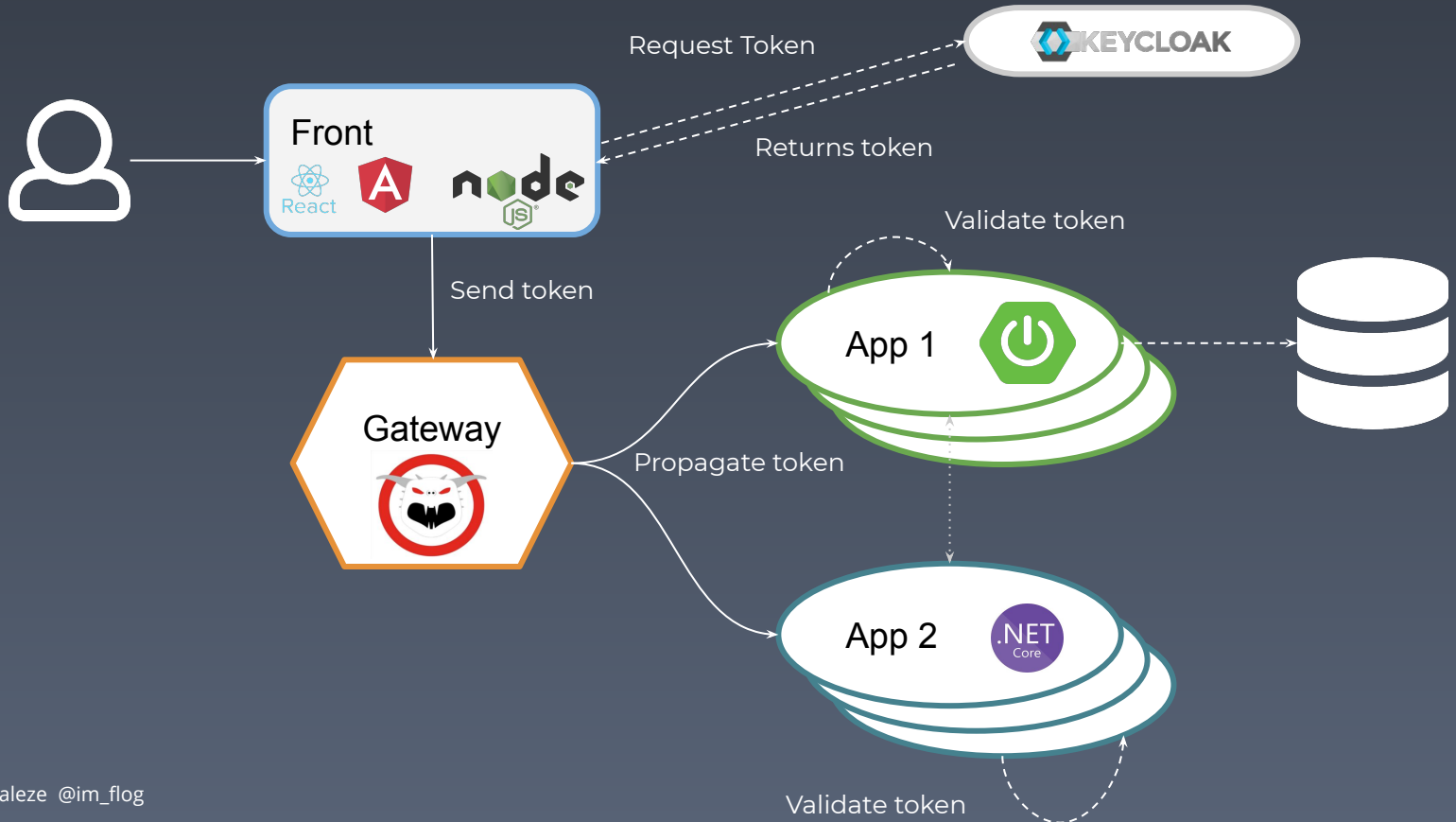
Passe le token a ton voisin

Pas besoin de s'identifier à chaque fois

- Une session est disponible
- Transfert du token dans les headers
- Validation à chaque "saut"

Client credentials pour les applications

L'architecture



Backend

https://github.com/ImFlog/microservices-security

Conclusion

- La sécurité c’est plus aussi compliqué
- On peut gérer assez facilement des droits
- Les standards sont bien ancrés
- Enfin des systèmes protégés !

- Il faut prendre le temps de définir son architecture
- La mise en place reste coûteuse (mais nécessaire)
- Il faut sensibiliser les développeurs
- Côté ops pas étudié


THANKS