Déroulement du cours

Cette semaine
- Choix d'article pour évaluation
- CM sécurité + CTD sécurité

La semaine prochaine
- Cours serverless et énergie
- TP AWS serverless (?)

Fin du mois
- Cours sur le fog + présentations d'articles
- TDs consacrés aux présentations d'articles

Grenoble INP ENSIMAG

Cloud et sécurité

Plan

Sécurité dans le cloud : problématique
- Analyse des risques

Sécurité dans les data centers

4 études de cas
- Service d'authentification dans OpenStack
- Attaque DDoS (OVH)
- Rupture d'isolation physique
- Authentification transverse dans K8S

Sécurité ?

Confidentialité

Intégrité
et donc authentification

Disponibilité

Le cadre légal évolue pour imposer ces caractéristiques à tout système d'information

Analyse des risques

Pas possible de se protéger contre tout

Importance d'identifier

- Acteurs
- Attaquants
- Menaces
- Vulnérabilités

Acteurs du Cloud

Fournisseur
- Infrastructure physique (machines, stockage, connexion réseau)
- Logiciels de gestion de cette infrastructure
- Intergiciels (PaaS, SaaS)

Client
- Développeurs d'applications
- Utilisateurs et développeurs de services
- Produisent, gèrent et accèdent à des données

} payent pour externaliser (la majorité) des problèmes de sécurité

Utilisateur
- Du service d'un client
  - Données liées au service

Profil des attaquants

Fournisseurs : moyens++
- concurrents
- états

Clients : moyens+
- concurrents
- pirates privés
- crime organisé

Utilisateurs : moyens-
- cybercrime traditionnel

Points clé pour les fournisseurs

Isolation
et donc préservation de la confidentialité

Disponibilité

Efficacité (coût!)

Points clé pour les clients

Commodité d'utilisation
- Authentification, single Sign On

Disponibilité
- passage à l'échelle
- vitesse de réponse aux incidents
- résistance aux DDos

Coût

Facilité de mise à jour
- influe directement sur le temps de correction des failles
- container-isation, devops

Points clé pour les utilisateurs

Facilité d'utilisation

Disponibilité

Confidentialité des données
- isolation
- authentification dans le logiciel développé par les clients

Ne connaissent généralement pas le(s) fournisseur(s)

Relations contractuelles

Fournisseur <-> client
  typiquement B2B
  avec garantie sur la qualité de service et le temps de réponse

Client <-> utilisateur
  EUA
  contient plus les droits du client que ceux de l'utilisateur

Menaces

Physiques
- Vol
- Destruction

Logiques
- Altération des données
- Vol des données
- Vol de puissance de calcul

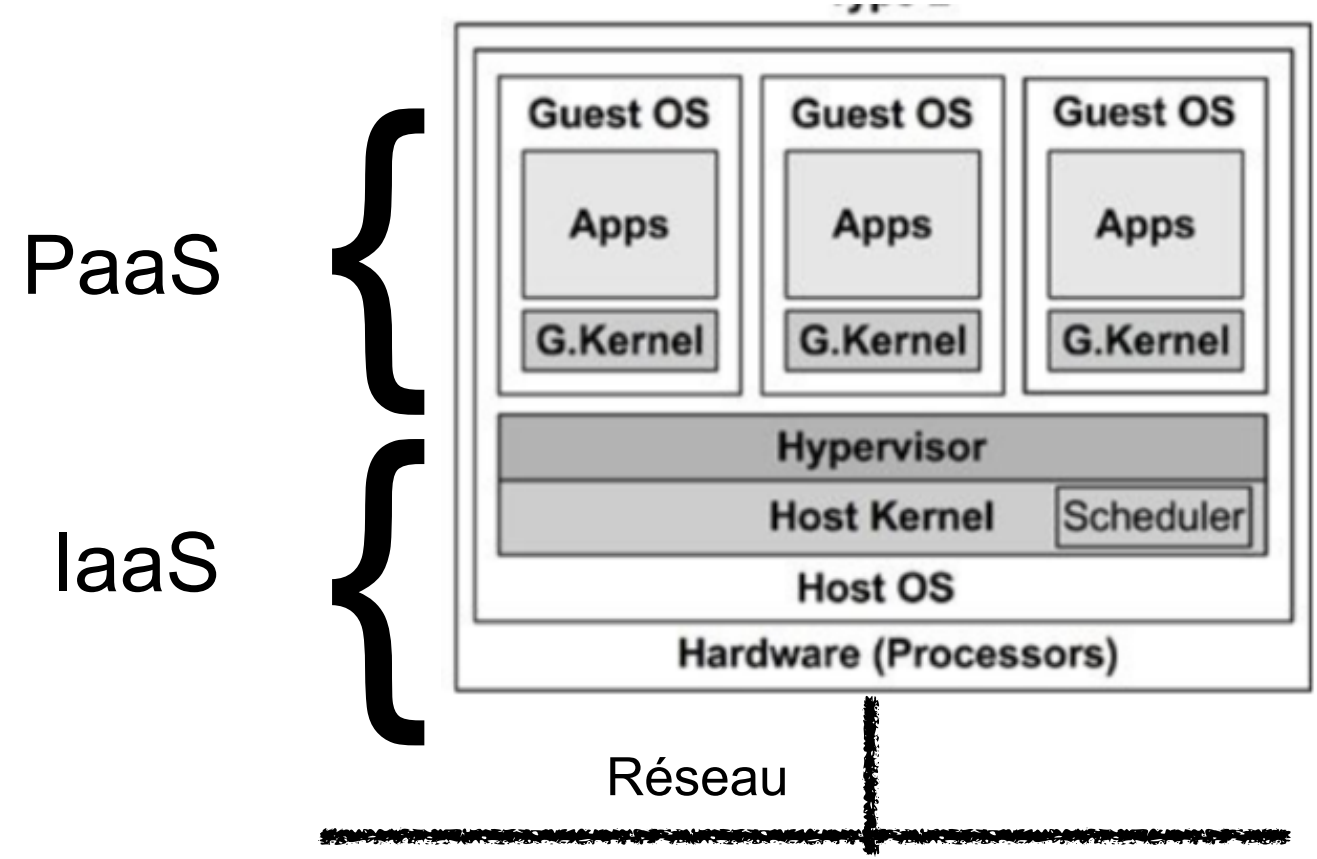
Particularités du cloud

attaques peuvent être locales
ou semi-locales

Importance cruciale de l'isolation

Aspects techniques pour l'isolation

Rappel architecture physique typique



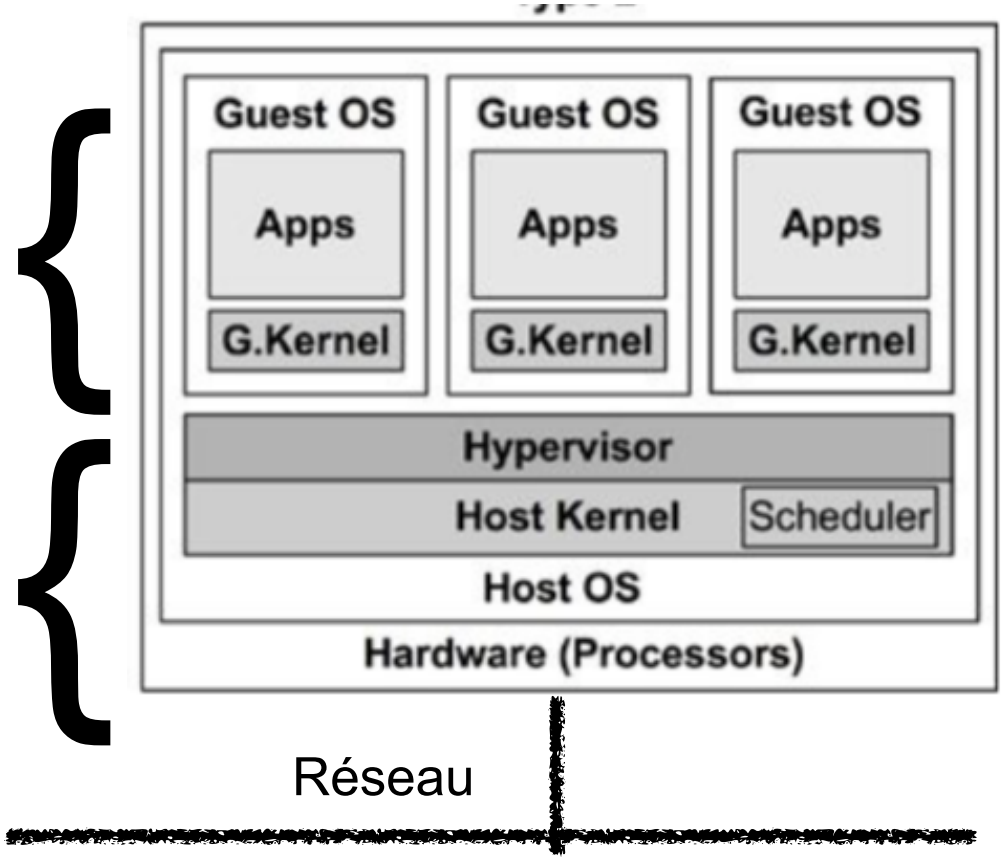
Aspects techniques pour l'isolation

élément clé : hyper viseur
assure séparation mémoire
et cache

La robustesse de
l'hyperviseur
est cruciale

PaaS

IaaS



Le stockage

Accès par une couche d'abstraction
- DBMS, K/V, FS

Interrogation par
- web service
- protocole dédié (NFS, Lustre,…)
- drivers locaux (File system)

Entités de stockage
- qualité de l'implémentation (résistance aux attaques type force brute)
- robustesse de l'authentification

Le stockage (2)

Chiffrement
  solution naturelle pour la confidentialité

Comment gérer les clés de chiffrement ?
  Automatique -> retrouvable par les attaquants
  Manuel -> intervention humaine nécessaire

Solution technique simple à mettre en place, organisationnelle compliquée

Risque de perte et de corruption
  copie = affaiblissement

Le réseau

Mise en place de réseaux privés virtuels
- locaux (VLAN) ou globaux (VPN)

Qualité de l'administration
- largement ou totalement automatisée
- importance de l'authentification sur les appareils

Robustesse du matériel
- dimensionné en conséquence
- mis à jour régulièrement

L’adversaire invincible

Celui qui peut contrôler le matériel

NSA ou autres agences gouvernementales

Il ne faut pas que ses connaissances soient diffusées

Services liés à la sécurité

Gestion des identités et des groupes
  et des certificats associés
  et des clés (publiques et/ou privés)

Détection automatique de menaces ou d'attaques

Tests de compliance automatisés

Protection contre
  attaques applicatives
  atteintes à la disponibilité

....

4 études de cas

Le service d'authentification d'OpenStack

La solution anti-DDoS de OVH

L'impact d'une faille sur les hyper viseurs

L'authentification transverse dans Kubernetes

Authentification

Les 3 dimensions de l'authentification

1/ secret partagé

2/ détention d'un objet

3/ caractéristiques physiques

Dans le DataCenter : typiquement 2 sur 3, parfois 3

À l'extérieur : 1 ou 2/3

Client utilise biométrie + secret partagé

Authentification

Élément clé pour toute la sécurité
  avec la gestion des identités au sens large et des droits d'accès

Dimension interne
  authentification pour l'administration

Dimension externe
  les clients depuis leur localisation
  les utilisateurs

Les éléments

Authentification : process de confirmation d'une identité
Credentials : données qui confirme l'identité

User : une personne, un système ou un service

Domaine : ensemble de projets et d'utilisateurs pour une gestion d'identité donnée (ex : entreprise) -> namespace

Endpoint : adresse d'un service

Les éléments

Groupe : ensemble d'utilisateurs pour un domaine
Projet : ensemble de ressources et d'identités

Region : un ensemble de ressources physiques et logiques OpenStack

Role : un ensemble de droits sur des ressources

Service : points d'accès à des ressources ou des traitements
Token : élément à présenter pour accéder aux services

Authentification : rôle des acteurs

Administrateur plate-forme
- crée les régions et les utilisateurs de base (domaine)
- crée les correspondances entre utilisateur, administrateur et ressources

Administrateur du client
- crée les projets
- crée les utilisateurs
- assignent les rôles aux utilisateurs

Administrateurs projet
- crée/assigne les droits aux utilisateurs associés aux projets
- crée les utilisateurs finaux (automatiquement)

Authentification dans OpenStack

1/ utilisateur se connecte

requête d'authentification : contient un mot de passe et/ou un token

si succès reçoit un token d'identification qui contient les credentials et une portée (projet, domaine, region, none)

Token a un ID unique et une date d'expiration

2/ utilise ce token pour se connecter aux services

requête REST avec le token en paramètre

le token contient un rôle qui définit des permissions

les services possibles pour ce token sont listables

Exemple de requête

```
{
    "auth": {
        "identity": {
            "methods": [
                "password"
            ],
            "password": {
                "user": {
                    "id": "ee4dfb6e5540447cb3741905149d9b6e",
                    "password": "devstacker"
                }
            }
        },
        "scope": {
            "system": {
                "all": true
            }
        }
    }
}
```

Exemple de token reçu

Grenoble INP ENSIMAG


```
{
    "token": {
        "audit_ids": [
            "3T2dc1CGQxyJsHdDu1xkcw"
        ],
```


...


```
    "expires_at": "2015-11-07T02:58:43.578887Z",
    "issued_at": "2015-11-07T01:58:43.578929Z",
    "methods": [
        "password"
    ],
    "roles": [
        {
            "id": "51cc68287d524c759f47c811e6463340",
            "name": "admin"
        }
    ],
    "system": {
        "all": true
```

Gestion des rôles

Utilisé pour définir les droits sur les services
Suit la norme RBAC
Rôles dépendent de l'utilisateur et du groupe


```
{
    "links": {
        "self": "http://example.com/identity/v3/projects/9e5a15e2c0dd42aab0990a463e839ac1/users/b9
64a9e51c0046a4a84d3f83a135a97c/roles",
        "previous": null,
        "next": null
    },
    "roles": [
        {
            "id": "3b5347fa7a144008ba57c0acea469cc3",
            "links": {
                "self": "http://example.com/identity/v3/roles/3b5347fa7a144008ba57c0acea469cc3"
            },
            "name": "admin"
        }
    ]
}
```

Exemple de services

```
{
    "service": {
        "description": "Keystone Identity Service",
        "enabled": true,
        "id": "686766",
        "links": {
            "self": "http://example.com/identity/v3/services/686766"
        },
        "name": "keystone",
        "type": "identity"
    }
}
```

Exemple de services (2)


```
"services": [
    {
        "description": "Nova Compute Service",
        "enabled": true,
        "id": "1999c3a858c7408fb586817620695098",
        "links": {
            "self": "http://example.com/identity/v3/services/1999c3a858c7408fb586817620695098"
        },
        "name": "nova",
        "type": "compute"
    },
    {
        "description": "Cinder Volume Service V2",
        "enabled": true,
        "id": "39216610e75547f1883037e11976fc0f",
        "links": {
            "self": "http://example.com/identity/v3/services/39216610e75547f1883037e11976fc0f"
        },
        "name": "cinderv2",
        "type": "volumev2"
    },
```

Application credentials

Permet à une application d'utiliser l'identité d'un utilisateur sans devoir connaître son mot de passe

C'est de la délégation d'identité

Étapes

1/ création par l'utilisateur d'une application credential avec une partie de ses droits -> nouvel objet

2/ stockage de l'ID du credential et secret associé dans la configuration de l'application

Interface avec d'autres systèmes

Construction d'un credential incluant les informations d'authentification sur l'autre sytème (AWS, Azure, …)

en clair dans le token!

Les différentes méthodes d'authentification

Login/mot de passe
  dans une base interne au système

Serveur LDAP externe

Kerberos

Oauth

One Time Password

Single Sign On dans Openstack

Nécessite de modifier la configuration par défaut
- en spécifiant quels sont les serveurs avec lesquels c'est possible
  - c'est-à-dire sans possibilité d'usurpation
  - pour éviter l'attaque man-in-the-middle

Vulnérabilités (potentielles)

Implémentation
- manque d'aléatoire
- vérification incorrecte

Structurelles
- vol de token ?
  - analogie avec les cookies
- permet l'usurpation d'identité
- unique niveau d'authentification
- possibilité de révocation

Suppose que l'architecture est sûre (cf. SSO)
- ne protège pas contre les attaques internes

et que le token n'en sorte pas
- Isolation entre les projets/domaines/régions

Sécurité est une chaine dont le point faible est le plus faible des maillons

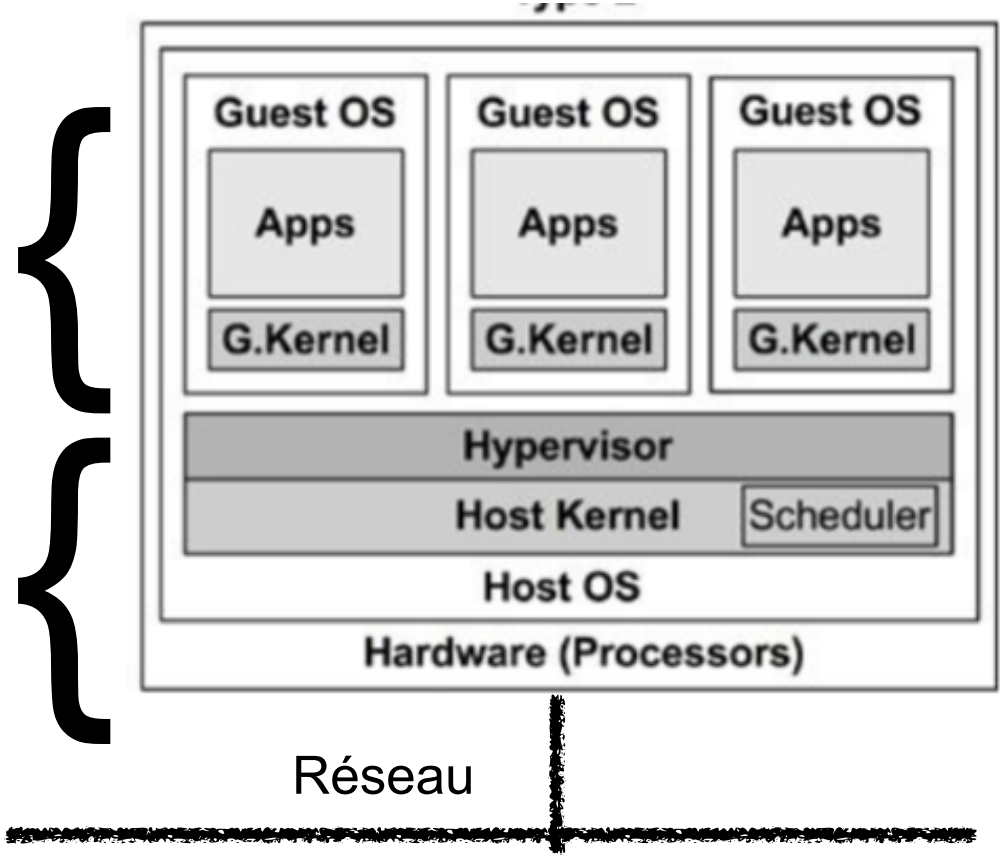
Les dénis de service

Attaque sur la disponibilité du système

Différents niveaux
On restreint aux attaques externes

PaaS

IaaS



Les dénis de service

Par logique

envoi de requêtes erronées provoquant un plantage

requête trop lourde -> DoS financier possible

Par force brute

saturation des liens de communication

interne (depuis l'intérieur du cloud)

externe

Quels sont les attaquants ?

Business model des attaques DDoS
- modèle de l'extorsion de fond

Outil de base : le botnet
- réseau de machines compromises (PC, frigo etc.)
- envoie du trafic
- parfois en utilisant un miroir (amplificateur)
- exploite l'absence de filtrage dans IP

Résultat
- suffisant pour saturer les coeurs d'internet
  - exemple : 125 Goctets/s (Mirai sur OVH)

Protection contre les dénis de service

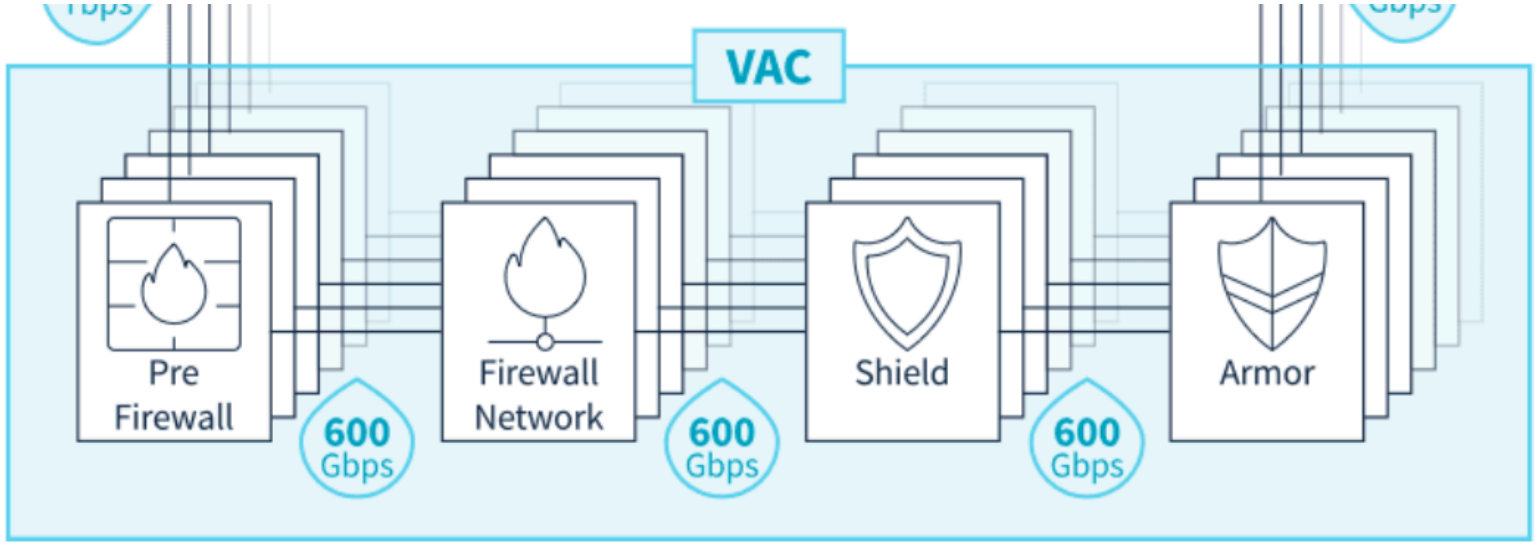
Logique
- firewall applicatif
- à dimensionner pour qu'il ne devienne pas un goulet

Par force brute
- dimensionnement d'architectures
- analyse du traffic en direct

Protection anti DDos de OVH

Développement d'une solution in-house
« les plus efficaces du marché »
Protège contre les DDoS force brute

Analyse en direct du traffic
  Activation en cas de traffic anormal
  infrastructure unique : le VAC
    Tri actif du traffic



Protection anti DDos de OVH

Pre-firewall
- en charge d'aiguiller les paquets suspects vers la suite de l'infrastructure
- règles fixées par OVH
- peut monter à 30Tbits/s

Firewall
- configurable par le client
- applique des règles de filtrage de traffic

Shield and Armor
- capacité de calcul plus avancé
- en charge de traiter les attaques par amplification

Conclusion sur les DDoS

Reste une menace très concrète

Un des grands avantages d'utiliser un service de cloud

Alternatives

1. acheter de la connectivité supplémentaire pour encaisser les attaques
2. dupliquer le service sur plusieurs sites
3. payer la rançon :-(

IAAS : Perte d'isolation

une faille sur Xen/KVM/intel

http://xenbits.xen.org/xsa/advisory-133.html

détails techniques : https://www.crowdstrike.com/blog/venom-vulnerability-details/

Lisez et analysez là

Quelles conséquences ?

Quelles gestions de la part
- du fournisseur
- du client
- de l'utilisateur

Conséquences

Rupture de l'isolation matérielle

- Un client peut accéder aux informations d'un autre

À peu près le pire scénario pour un fournisseur et un client

Réactions du fournisseur

Patcher!

importance du secret pendant un moment

mais pas trop longtemps!

ça va coûter cher….

Ce n’est pas toujours possible

1. fonctionnement dégradé : pas de partage de machine physique
2. placement choisi pour minimiser les interactions
3. ?

Réactions du client

Demande au fournisseur de patcher le plus vite possible

Exiger du non partagé

Demander un audit sur ce qui s'est passé précédemment

si exploitation de la faille

-> importance des logs de la plate-forme

et notamment avec qui « ses » machines ont été partagées

doit être prévu contractuellement

ce n'est pas encore largement le cas

Réaction de l'utilisateur

Encore faut-il qu'il soit au courant!
- évolution du cadre législatif vers plus de transparence

En pratique à peu près aucun droit de recours
- quelle marge de manoeuvre de toute façon ?
- mais ça évolue…

Quelles conclusions ?

Pour le fournisseur
- se baser sur des technologies les plus éprouvées et répandues possible
  - et que l'on peut modifier rapidement si besoin
  - une des raisons du succès de l'open source

Pour le client
- qualités et écoutes du fournisseur

Pour les utilisateurs
- transparence ?
- choix ?

Authentification transverse dans K8S

Exemple d'application : site web avec BD associé
- 2 containers (ou pods)
  - Bases de données
  - Front-end
- authentification sur la base de données
  - informations communes aux 2 pods -> comment la gérer ?

Informations dans le code directement
- peu flexible
- droits d'accès au code ?

Séparation entre le code et les informations de configuration
- comme les informations d'authentification

Comment faire ?

Dans Kubernetes, s est pour sécurité ?

Analyse des solutions proposées par K8S

configmap (https://kubernetes.io/docs/concepts/configuration/configmap/)

secrets (https://kubernetes.io/docs/concepts/configuration/secret/)

Analyser et critiquer ces solutions