

# Sécurité & Modélisation des menaces

Guillaume Bienkowski — Brincube

# Plan

- Introduction
- Présentation de la sécurité
- Gestion des risques de sécurité
- TD

# Introduction

Une petite histoire...

# La largeur du domaine de la sécurité

Des myriades de sous domaines:

- sécurité des personnes (physique et mentale)
- sécurité des biens (DAB, stocks, etc.)
- sécurité économique
- sécurité de l'environnement (chimie, nucléaire, etc.)
- sécurité de l'information (c'est là qu'on va travailler)

# **L'objectif de ce cours/TD: la sécurité de l'information**

La sécurité est un vaste sujet, nous nous concentrerons sur la sécurité de l'information.

Ce cours présente une méthodologie et un exemple concret de sécurisation d'un ensemble de logiciels de traitement de l'information.

# CIA / CID

Modèle qui sous-tend la grande majorité des méthodes de sécurisation. Ce sont les 3 choses considérées comme nécessaires à la sécurité de l'information.

- **Confidentialité**

La donnée doit être accessible uniquement aux destinataires légitimes.

- **Intégrité**

La donnée ne doit pouvoir être créée, modifiée ou supprimée seulement par les utilisateurs légitimes

- **Disponibilité (Availability)**

La donnée doit être consultable à tout moment par les utilisateurs.

# La sécurité parfaite n'existe pas

- facteur humain
- course aux armes -- intérêt financier, hackers motivés VS équipe de sécurité
- asymétrie du problème: 1 seule vulnérabilité compromet l'entièreté du système
- évolutions techniques / vétusté, facteurs économiques
- éléments naturels (*Disponibilité*)

→ insoluble, mais on peut réduire les *risques*

# La sécurité (informatique) comme une gestion du risque

Glossaire:

**Vulnérabilité:** une faiblesse du système étudié. Faiblesse matérielle, logicielle ou de procédure (*"il n'est pas écrit qu'il faut fermer le coffre fort en partant le soir"*).

**Menace:** un événement qui pourrait vous porter préjudice. Une catastrophe, un attaquant interne ou externe motivé, ...

**Risque:** conjonction entre les vulnérabilités et les menaces

**Impact:** les conséquences si le risque survient.



# La sécurité comme une gestion du risque

De chaque menace et vulnérabilité découle un risque:

**Vulnérabilité:** le béton de l'enceinte de confinement de ma centrale ne résiste pas à plus 40 tonnes de pression extérieure

**Menace:** Un attaquant motivé pourrait détourner un avion de 100T et le diriger sur mon enceinte de confinement

**Risque:** Il est possible que mon confinement soit inadapté à un avion de ligne qui s'écraserait sur mon bâtiment.

**Impact:** Le coeur de ma centrale pourrait entrer en fusion, >100 morts immédiats, >100 000 morts sur 10 ans

# La sécurité comme une gestion du risque

Autre exemple:

**Vulnérabilité:** mon logiciel possède une faille de sécurité exploitable pour exfiltrer les données de mes clients

**Menace:**

**Risque:**

**Impact:**

# La sécurité comme une gestion du risque

Autre exemple:

**Vulnérabilité:** mon logiciel possède une faille de sécurité exploitable pour exfiltrer les données de mes clients

**Menace:** ma société est assez grosse pour attirer des attaquants motivés et perfectionnés, disons 1 chance sur 100 dans l'année

**Risque:**

**Impact:**

# La sécurité comme une gestion du risque

Autre exemple:

**Vulnérabilité:** mon logiciel possède une faille de sécurité exploitable pour exfiltrer les données de mes clients

**Menace:** ma société est assez grosse pour attirer des attaquants motivés et perfectionnés, disons 1 chance sur 100 dans l'année

**Risque:** Mes données risquent d'être publiées ou utilisées par des attaquants motivés pour couler mon business.

**Impact:**

# La sécurité comme une gestion du risque

Autre exemple:

**Vulnérabilité:** mon logiciel possède une faille de sécurité exploitable pour exfiltrer les données de mes clients

**Menace:** ma société est assez grosse pour attirer des attaquants motivés et perfectionnés, disons 1 chance sur 100 dans l'année

**Risque:** Mes données risquent d'être publiées ou utilisées par des attaquants motivés pour couler mon business.

**Impact:** Perte de > 50% des clients, voire faillite

# Comment adresser ces risques

Il existe une myriade de méthodes: [EBIOS](#), [MEHARI](#), [OCTAVE](#), ...

Toutes se valent.

La plupart utilisent les mêmes prémisses:

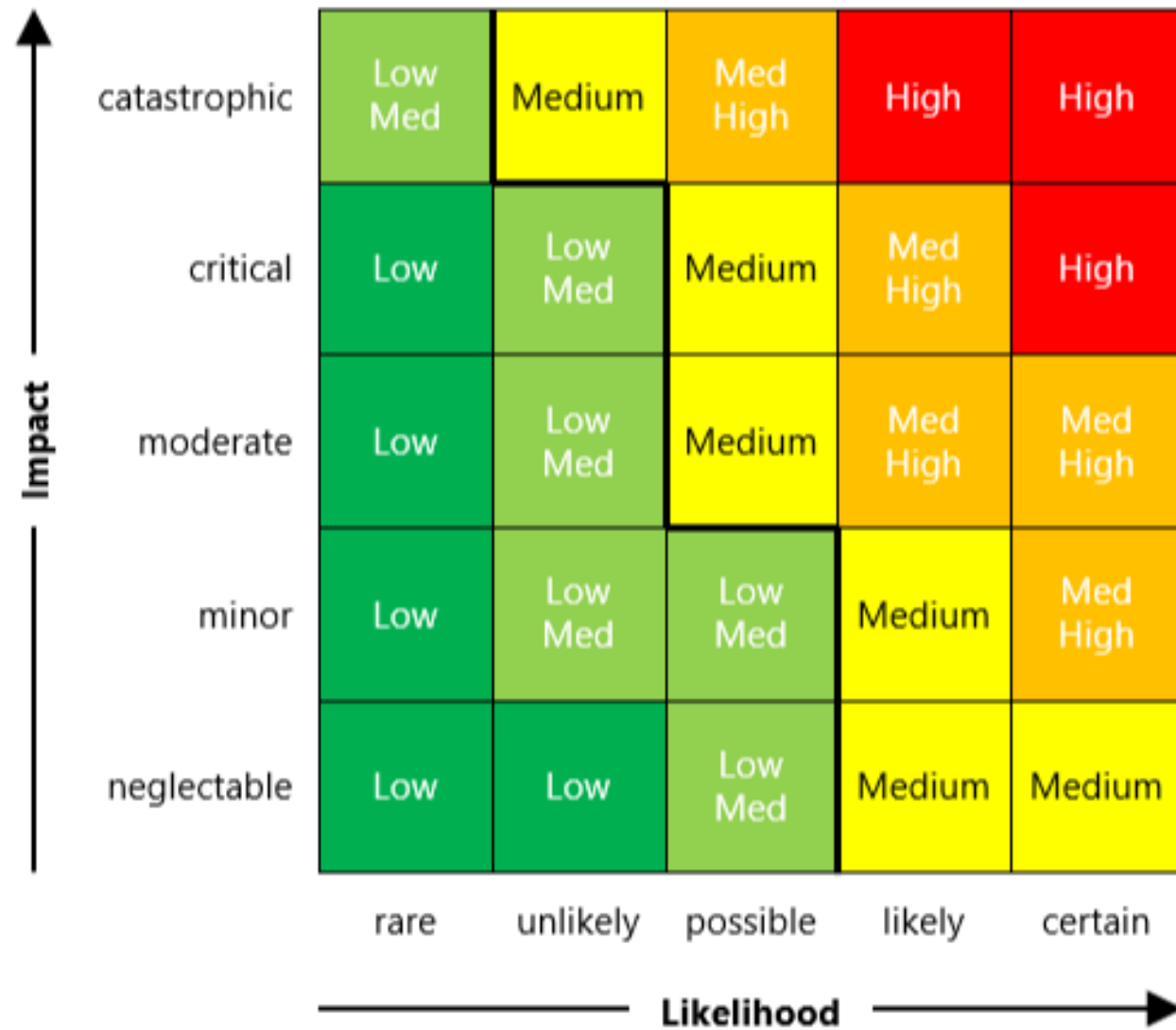
- Assurer le CIA/CID
- Échelonner les réponses en accord avec les risques
- Suivi long terme des actions, et ré-évaluation régulière

# Comment adresser ces risques

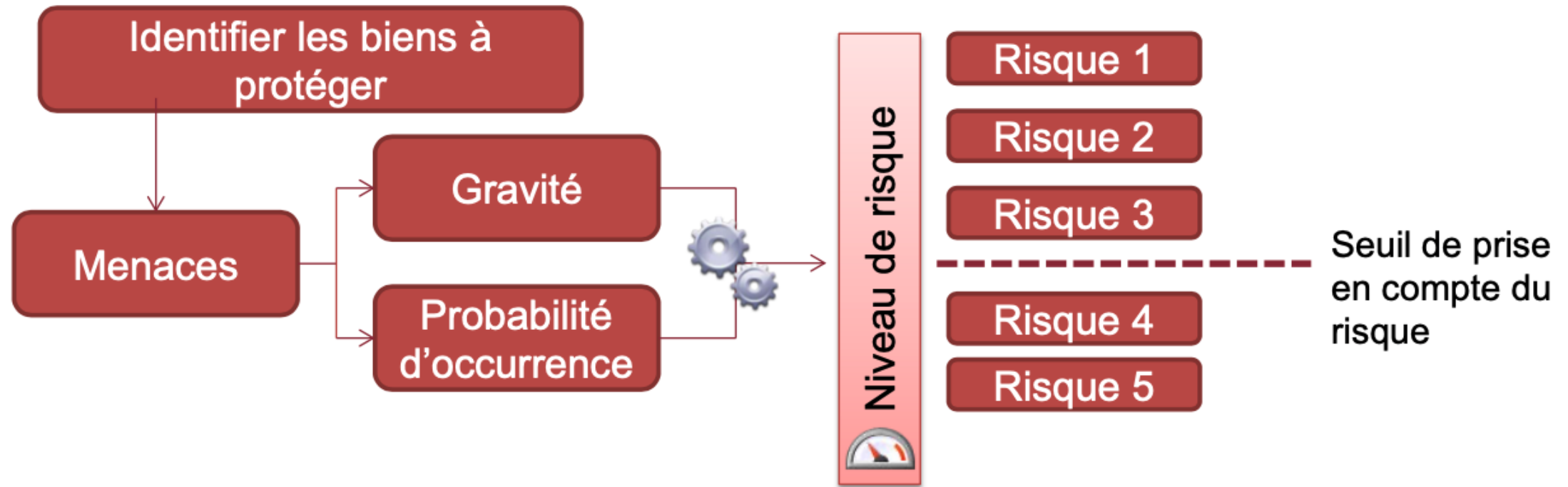
À partir de ce constat, on échelonne les mesures de sécurité selon 2 axes:

- la probabilité de la menace
- les conséquences, l'impact, la gravité de la menace

L'évaluation est subjective: elle va permettre d'ordonner les mesures selon leur impact.



# La méthode d'analyse



C'est celle qui est recommandée par l'ANSSI.



# Plusieurs types de solutions à une menace de sécurité

- **Techniques**

La conception du logiciel ou sa mise en oeuvre réduisent voire empêchent carrément le souci de sécurité. Poka Yoke.

*Exemples:*

- logiciel qui tourne avec un utilisateur dédié sur la machine, et non pas en *root*.
- Authentification déléguée OpenID ou SAML: plus besoin de stocker de mot de passe.
- Publication d'une nouvelle version impossible si des vulnérabilités sont détectées via scanning automatique

# Plusieurs types de solutions à une menace de sécurité

- **Organisationnelles ou humaines**

La structure de l'entreprise et du travail rend insignifiante ou inopérable la vulnérabilité

*Exemples:*

- Accès aux salles des machines via badge, autorisé seulement aux admin systèmes
- Impossible de faire entrer du code sans relecture
- Rituels de revue des risques (réunion de sécurité, ...)

# La méthode d'analyse

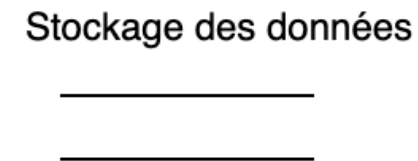
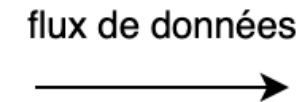
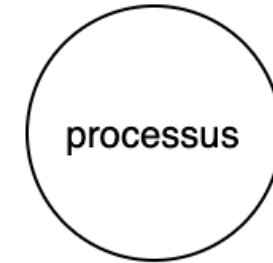
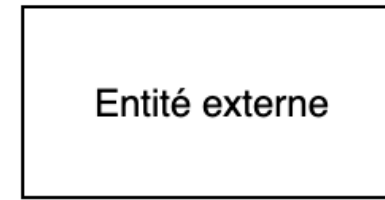
## 4 étapes:

1. identification des biens à protéger
2. énumération des manières d'enfreindre le *CID* sur ces biens
3. détermination de l'impact et la probabilité d'occurrence, hiérarchisation des risques
4. élaboration de contre mesures techniques ou organisationnelle pour réduire ces risques

# Identification des biens à protéger

On utilise la méthode du Data Flow Diagram.

Modélisation via un diagramme énumérant tous les flux de données, et les frontières de changement de droits.



# Identification des biens à protéger

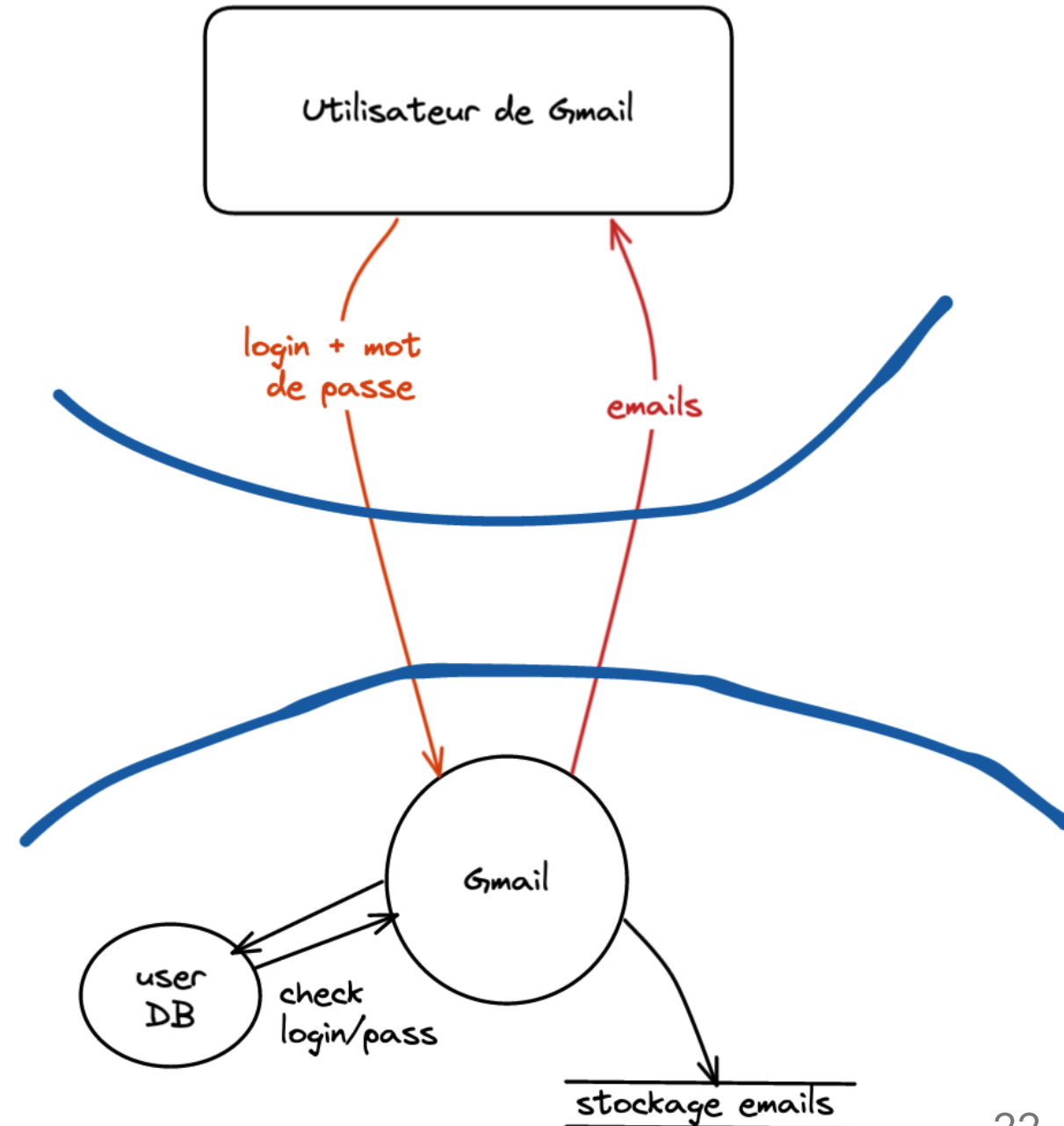
La liste des biens est constituée des choses en lesquelles un attaquant est intéressé.

Dans le diagramme de flow, les flux de données et les endroits de stockages sont généralement les endroits intéressants.

# Exemple de Data Flow Diagram

Exemple avec un flow simple d'utilisateur Gmail.

On explore pas ici les utilisateurs admin ou le processus de création d'utilisateur ou de changement de mot de passe.



# On démarre le TD

Comme la dernière fois, il y a un questionnaire à remplir ensuite dans Google Forms:

<https://forms.gle/vn1BgveSHWnK7sf1A>

# Le sujet

Nous allons réaliser une étude des menaces sur une entreprise dont vous êtes un des architectes logiciel.

Le composant que nous allons étudier est le système d'ingestion de données d'Assurance Qualité de vos prestataires.

Il faudra utiliser l'analyse de risques que l'on vient de voir en cours, afin d'identifier les vulnérabilités, échelonner les risques associés, proposer des mesures de réduction de ces risques, et implémenter les parties techniques.



# Fonctionnement théorique

Votre société fait appel à plusieurs sous traitants qui vous fournissent en pièces diverses pour votre production.

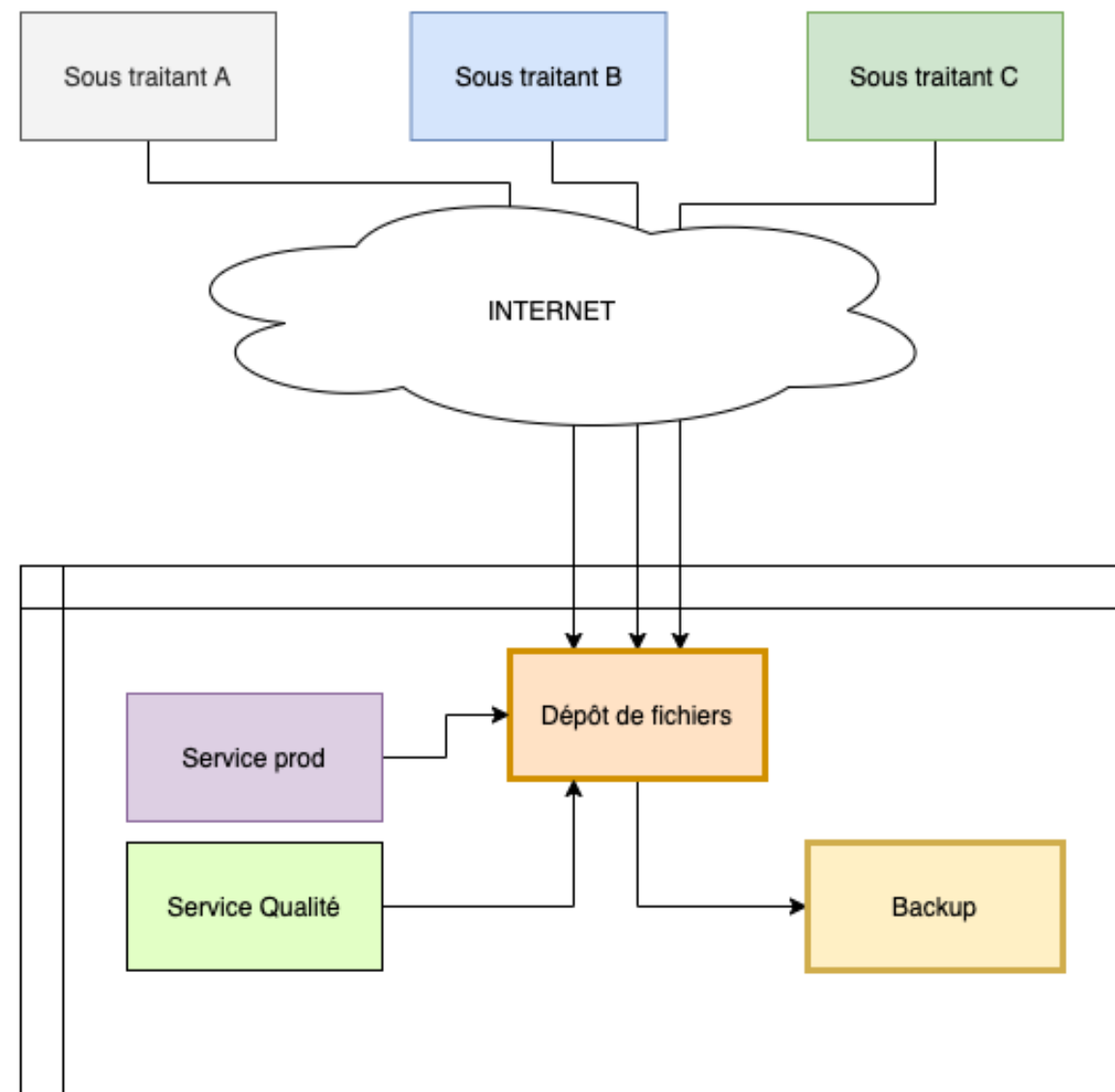
Leur contrat de fourniture stipule qu'ils doivent régulièrement vous fournir des rapports de contrôle qualité, ainsi que des rapports de production via un système d'échange de fichiers, de manière automatisée.

Une fois les fichiers déposés, vos collègues affectés à la qualité peuvent les consulter. *//s n'ont pas besoin de les modifier par contre.*

Côté production, vos équipes sont aussi responsables d'ajouter des fichiers dans les dossier.

Vous avez déjà anticipé des pertes éventuelles de fichiers et avez mis en place une sauvegarde régulière de ces fichiers.

# Schéma



# Comment démarrer

## 1. Télécharger le code du TD:

```
git clone https://github.com/masterind4/masterind4.github.io.git  
cd masterind4.github.io/td_securite
```

## 2. Ouvrir le document `Scoring.ods`

Il se trouve dans le dossier `td_securite` que vous venez de télécharger.

# Comment démarrer

Clonez le dépôt, et rendez-vous dans le dossier `td_securite` **dans un terminal**.

À cet endroit, lancez la commande `./docker-compose up` dans le terminal. Laissez le temps à docker-compose de faire son affaire, et ouvrez un navigateur sur <http://localhost:8080>

- `Ctrl + C` pour arrêter le serveur dans le terminal
- `./docker-compose up` pour lancer ou relancer.
- `./docker-compose build` pour recompiler filegator, si changé
- `./docker-compose down` pour complètement supprimer les logiciels.

Les fichiers manipulés par le site sont visibles dans le dossier

`./td_securite/run/{stockage,backup}`

# Fonctionnement pratique

Les fournisseurs ont un login/mot de passe pour envoyer leurs fichiers sur l'interface web <http://localhost:8080>

```
fournisseur:HarDt0guEss
```

L'admin a un login/mot de passe:

```
admin:An0tHerPassW0rd
```

Les utilisateurs internes ont un mot de passe dédié:

```
userinterne:SomeThinGh4sToGive
```

# Prise en main du TD

Amusez-vous quelques minutes avec le portail. Constatez que les fonctionnalités attendues sont présentes:

- les fournisseurs peuvent envoyer des fichiers (faites le pour tester)
- les fichiers sont stockés à la fois dans le dossier `stockage` ET le dossier `backup`
- Les utilisateurs internes peuvent consulter ces fichiers et en ajouter (peut être pas tous non??)

Nous allons procéder avec les 4 étapes de l'analyse de risques.

# Étape 1: Détermination des biens

On fait ça ensemble.

Lister les données manipulées par l'ensemble du système. Pour cela, on crée ensemble un schéma de flow de données. Ensuite, utiliser la première feuille du document ODS

Scoring.ods fourni.

A	B	C	
Numéro	Bien		
1			
2			
3			
4			
5			
6			
7			

+

☰

Liste des biens ▾

Score Vulnérabilités ▾

# Étape 2: Énumération des manières d'enfreindre le CID sur ces biens

## On fait ça ensemble

Laissez votre imagination parler. Pour cela utiliser la 2ème feuille du document `Scoring.ods` fourni, en remplissant les colonnes "Vulnérabilité" et "Risque".

Sélectionner le bien à gauche dans la liste déroulante.

A	B	C	D	S
Numéro	Bien	Vulnérabilité (texte)	Risque (texte)	Im
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				



## Étape 3: Notation des menaces

On applique un score à ces risques et on liste ceux qu'on souhaite corriger.

## Étape 4: Mesures correctives

Énumération des mesures correctives, et on passe à l'action.

# Mise à jour de Filegator

- Se rendre sur la page des releases de Filegator:  
<https://github.com/filegator/filegator/releases>
- Trouver la release actuelle (v7.8.5)
- Identifier une nouvelle version plus récente qui corrige des failles de sécurité  
**(attention aux changements breaking)**
- Modifier le Dockerfile dans `td_securite/depot/Dockerfile` pour utiliser cette version (+ modif docker compose si besoin)
- Reconstruire l'image puis relancer docker-compose:
  - `Ctrl + C` dans votre terminal pour arrêter les logiciels
  - `./docker-compose build` pour reconstruire filegator
  - `./docker-compose up` pour relancer les logiciels avec cette version

Constater que ça fonctionne encore.

# Utilisation d'un utilisateur dédié pour la base MariaDB

Modifier les variables d'environnement passées au container `database` pour utiliser un utilisateur dédié à la base d'authentification.

- [S'inspirer de la documentation ici](#) pour modifier la section `database` du fichier `td_securite/docker-compose.yml`
- Modifier la configuration du dépôt de fichier pour utiliser cet utilisateur:  
`td_securite/depot/configuration.php` (chercher `database` dans ce fichier et faire le nécessaire pour modifier)
- `Ctrl + C` pour stopper les logiciels
- Bien supprimer les données ( `rm -r run/db/*` ) et tout redémarrer:  
`./docker-compose up` (puis vérifier que tout refonctionne)

## Utilisateurs dédiés à chaque prestataire

Modifier le script `db/init.sql` pour créer des utilisateurs dédiés pour chaque cas: un par fournisseur, et chaque utilisateur interne (QA et Prod).

**Bien ajuster leur `homedir` et leurs droits pour qu'ils aient le strict minimum requis pour faire leur job.**

Si vous modifiez le fichier `db/init.sql`, il faudra supprimer la base de données et relancer les logiciels comme dans l'étape précédente.

## Autres améliorations possibles

- Implémenter un filtrage IP sur les prestataires
- Passer le frontal en HTTPS (très important, mais hors du scope ici)
- Rendre le frontal hautement disponible en ayant 2 frontaux
- Rendre le stockage hautement disponible via un stockage partagé
- Changer le frontal pour une authentification par certificats ou clés SSH
- Passer le backup à une authentification forte via clé SSH plutôt qu'un login/pass

# Questions?

