

# Les architectures Microservices

## idée générale et avantages

Mohammed Khatiri<sup>21</sup>

<sup>1</sup>University Mohammed First  
Faculty of Sciences, LaRI, Morocco  
**Professeur El Mostafa DAOUDI**

<sup>2</sup>Univ. Grenoble Alpes  
CNRS, Inria, LIG, France  
**Professeur Denis trystram**

Séminaires - LaRI  
28-05-2018

# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 Avantage des Microservices
  - Mises à jour
  - Bugs décentralisée
  - Consommation de ressources et Scalabilité
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion

# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 Avantage des Microservices
  - Mises à jour
  - Bugs décentralisée
  - Consommation de ressources et Scalabilité
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion

# Architectures Monolithique

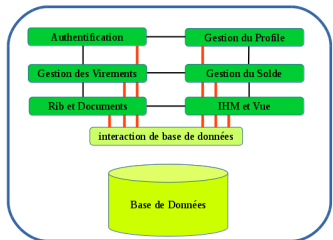


Figure – Exemple d'application de banque **Architectures Monolithique**

## Principe

- Un gros code contenant toutes les fonctionnalités et les différentes couches logicielles.
- Les services communiquent à travers des appels de fonctions
- Une seule grosse compilation et un seul livrable (un gros fichier WAR)
- Une seule pile logicielle (Linux, JVM, Tomcat Présentation et bibliothèques tierces)

# Architectures des Microservices

Premières discussions autour du terme microservice en 2011 lors d'un workshop sur les architectures logicielles

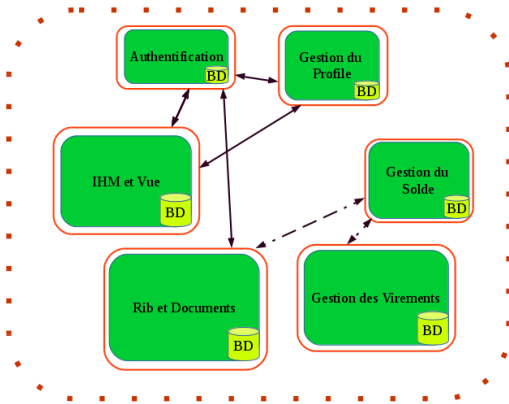


Figure – Exemple d'application de banque Architectures Microservices

## Principe :

- Un ensemble de Microservices autonomes.
- Chaque Microservices réalise une seule fonctionnalité de l'application globale.
- Un Microservice possède un contexte d'exécution séparé des autres.  
**isolation**
- Les Microservices communiquent à travers le réseau.
  - Communication synchrone : services web (**SOAP** ou **REST**)
  - Communication asynchrone : les bus d'événements

# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 Avantage des Microservices
  - Mises à jour
  - Bugs décentralisée
  - Consommation de ressources et Scalabilité
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion

- Un Microservices doit être séparé de l'application globale.
- Cette technique consiste à isoler l'utilisation des ressources de type processeur, mémoire et disque par application sur une même machine.
- Un microservice peut contenir toutes les couches logicielles (IHM, middleware et base de données)
- Comment on peut fait l'isolation ??????
  - Virtualisation.
  - Machines virtuelles.
  - Conteneurs.

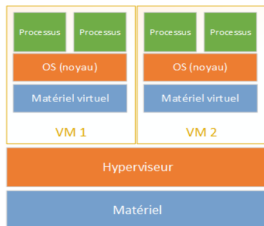


# Machines virtuelles VS conteneurs

## Mode de fonctionnement

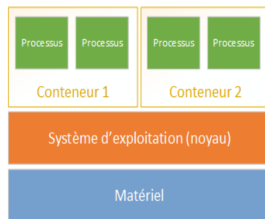
### Machine virtuelle :

- Représente un ordinateur complet
- Un système d'exploitation(SE) complet avec les pilotes et des des systèmes de fichiers binaires
- L'application



### Conteneurs

- Une image de base composée d'un SE. (docker image)
- L'application



# Machines virtuelles VS conteneurs

## Le différence entre les machines virtuelles et les conteneurs

### Machine virtuelle :

- S'exécute dans un environnement isolé de virtualisation matérielle grâce à un hyperviseur
- Tous les éléments exécutés dans une VM sont masqués du SE hôte
- Apparaît comme un ordinateur physique.
- Très lourde lors de démarrage.

### Conteneurs

- Ne nécessite pas d'hyperviseur pour assurer son
- utilise les fonctionnalités d'isolation des processus et du système de fichier du noyau linux.
- Le conteneur paraît comme une instance unique du système d'exploitation.
- Le temps de démarrage et la surcharge d'espace sont réduits.

# Machines virtuelles VS conteneurs

performances

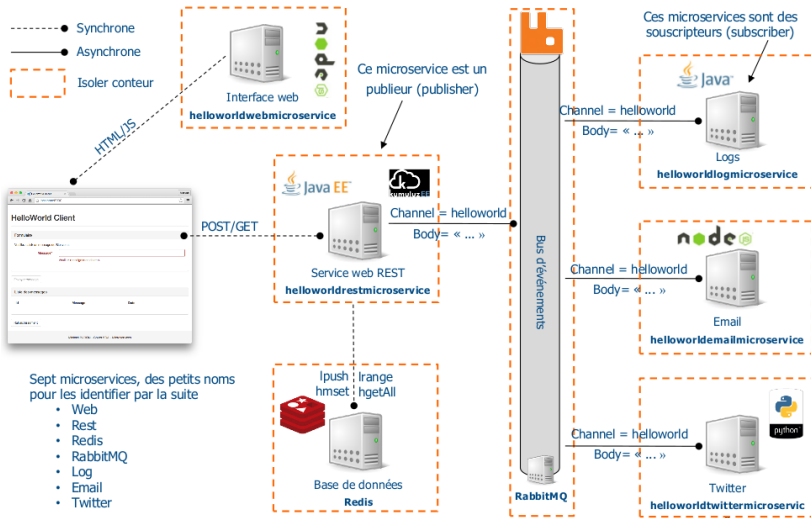
	<b>Machine Virtuelle (574 Mo)</b>	<b>Conteneur (100 Ko)</b>
<b>Démarrage</b>	~ 1 minutes	0.2 seconde
<b>Mémoire</b>	min 256 Mo	~ 0.3 Mo
<b>Espace Disque</b>	min 1 Go	~ 0.1 Mo

# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 Avantage des Microservices
  - Mises à jour
  - Bugs décentralisée
  - Consommation de ressources et Scalabilité
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion

- **les Microservices** : langage selon le besoin.
- **Isolation** : Docker
- **Image système** : [▶ ://hub.docker.com/](https://hub.docker.com/)
- **Communication**
  - **Synchrone** : HTML-JS, POST/GET, SOAP.
  - **Asynchrone** : RabbitMQ
- **Automatiser le déploiement sur clusters** : Kubernetes

## Exemple HELLO WORLD - Résumer



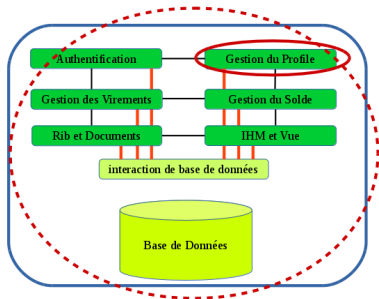
# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 **Avantage des Microservices**
  - **Mises à jour**
  - Bugs décentralisée
  - Consommation de ressources et Scalabilité
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion

# Mises à jour ou Maintenance

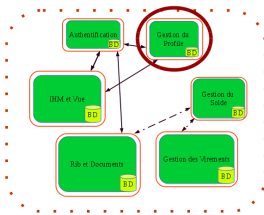
## Applications Monolithique :

- Mise à jours ou Maintenance d'un service nécessite la mise a jour de toute l'application.
- Redémarrer tout application.



## Applications avec Microservices :

- Mise à jours d'un Microservice ne nécessite pas la mise a jour de toute l'application.
- lancer juste la nouvelle 'instance du Microservice concerné.
- L'application ne s'arrête pas

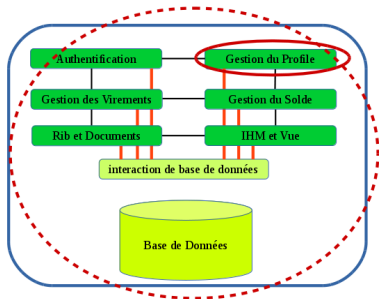




# Mises à jour ou Maintenance

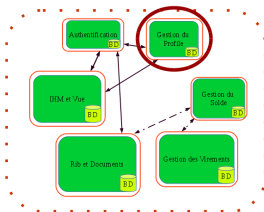
## Applications Monolithique :

- Mise à jours ou Maintenance d'un service nécessite la mise a jour de toute l'application.
- Redémarrer tout application.



## Applications avec Microservices :

- Mise à jours d'un Microservice ne nécessite pas la mise a jour de toute l'application.
- lancer juste la nouvelle 'instance du Microservice concerné.
- L'application ne s'arrête pas

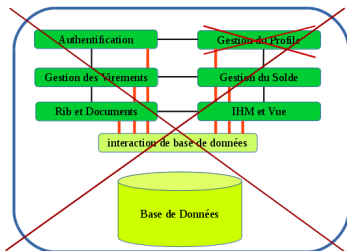


# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 **Avantage des Microservices**
  - Mises à jour
  - **Bugs décentralisée**
  - Consommation de ressources et Scalabilité
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion

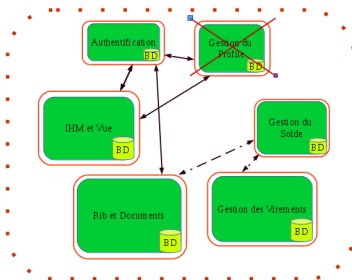
## Applications Monolithique :

- l'interruption d'un service implique l'interruption de toute l'application.



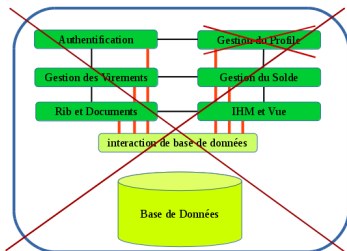
## Applications avec Microservices :

- l'interruption d'un service n'affecte pas toutes l'applications.



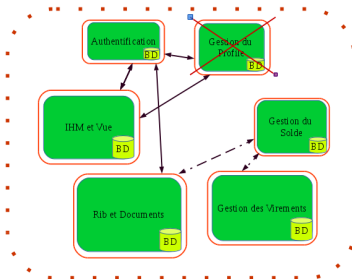
## Applications Monolithique :

- l'interruption d'un service implique l'interruption de toute l'application.



## Applications avec Microservices :

- l'interruption d'un service n'affecte pas toutes l'applications.



# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 **Avantage des Microservices**
  - Mises à jour
  - Bugs décentralisée
  - **Consommation de ressources et Scalabilité**
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion

## Applications Monolithique :

- Supposons que l'application est déployée sur 6 nœuds de calcul
- Même catégorie de clients pour l'application (quelle que soit la demande)
- Si le nombre de clients augmente :
  - Dupliquer l'application sur les machines et diviser le travail entre les deux instances
  - + 6 nœuds de calcul

## Applications avec Microservices :

- Supposons que l'application est déployée sur 6 nœuds de calcul (Chaque nœuds prend un Microservice)
- Les clients sont catégorisés en fonction des Microservices demandés
- Si le nombre d'utilisateurs augmente :
  - Besoin de savoir quel microservice est chargé?
  - Dupliquer Juste le microservice charger
  - + 1 ou 2 nœuds de calcul

## Applications Monolithique :

- Supposons que l'application est déployée sur **6 nœuds** de calcul
- Même catégorie de clients pour l'application (quelle que soit la demande)
- Si le nombre de clients augmente :
  - Dupliquer l'application sur les machines et diviser le travail entre les deux instances
  - + **6 nœuds** de calcul

## Applications avec Microservices :

- Supposons que l'application est déployée sur **6 nœuds** de calcul  
(**Chaque nœuds prend un Microservice**)
- Les clients sont catégorisés en fonction des Microservices demandés
- Si le nombre d'utilisateurs augmente :
  - Besoin de savoir quel microservict est chargé?
  - Dupliquer Juste le microservice charger
  - + **1 ou 2 nœuds** de calcul

# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 Avantage des Microservices
  - Mises à jour
  - Bugs décentralisée
  - Consommation de ressources et Scalabilité
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion



## Taille des services

Le plus grand principe des microservices concerne la taille de ces services.

- Pas de règles précises ou norme ou de spécification.
- Il est donc important de trouver un juste milieu entre des services imposants et des services trop petits

## la supervision

- La multiplicité des services entraîne une complexification de la supervision du système.
- le nombre de services croît régulièrement au fur et à mesure de la vie du système.
- Il devient vite impossible pour un humain de suivre tous les services manuellement
- Des règles d'alertes automatiques doivent être mises en place afin de savoir lorsqu'un service commence à ne plus fonctionner correctement
- Le rôle du kubernetes sur les clouds (mais la configuration est manuel)

# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 Avantage des Microservices
  - Mises à jour
  - Bugs décentralisée
  - Consommation de ressources et Scalabilité
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion

# Qui utilisent les microservices ?

- Uber : <https://eng.uber.com/soa/>
- Netflix : <https://eng.uber.com/soa/>
- Amazon : <http://fr.slideshare.net/apigee/i-love-apis-2015-microservices-at-amazon>
- Le prochain Eclipse Che IDE

# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 Avantage des Microservices
  - Mises à jour
  - Bugs décentralisée
  - Consommation de ressources et Scalabilité
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion

# Développer des architectures microservice?

- Savoir **Isoler** un microservice.
- Savoir **Coder** le contenu du microservice.
- Savoir faire **Communiquer** des microservices.
- Savoir **Composer** les microservices.
- Savoir **Répartir** les charges.

# Outline

- 1 Les architectures Logiciel
  - Les architectures Monolithique VS les architectures des Microservices
- 2 Architectures Microservices
  - Isolation des Microservices
  - Les outils
- 3 Exemple de HELLO WORLD
- 4 Avantage des Microservices
  - Mises à jour
  - Bugs décentralisée
  - Consommation de ressources et Scalabilité
- 5 Inconvénients (Limites) des architectures Microservices.
  - Inconvénients (Limites) des architectures Microservices
- 6 Qui utilisent les microservices ?
  - Qui utilisent les microservices ?
- 7 Conclusion
  - Développer des architectures microservice ?
  - Conclusion

- la différence entre les architectures Monolithique et Microservices.
- Principe d'isolation (Machine virtuelle ou conteneurs)
- Les outils nécessaires
- Les avantages et les limites des architectures Microservices.

Merci de votre présence

**Question ?**

Pour me contacter :

[khatiri.med@gmail.com](mailto:khatiri.med@gmail.com)