

Cloud Computing

Introduction

Entreprise A veut développer et déployer ses applications.

Que lui faut-il ?

Introduction

Acheter, déployer et maintenir des serveurs

Des employés

Attendre le retour sur investissement

Installer et gérer les OS des serveurs

Installer et déployer la pile logicielle (git, java, ...)

Développer et déployer

Manuellement ou automatiquement gérer la montée en charge des instances

Cloud

Le cloud permet d'accéder à distance à une large variété de ressources.

Ces ressources peuvent être matérielles ou logicielles.

Location de machines virtuelles, bases de données, files de messages ...

Cloud

Disponibilité des ressources : Changement de la capacité de stockage, la puissance requise ...

Paieement à l'usage : Uniquement ce qu'on utilise !

Ouverture : Utilisable via Internet

Modèles



Software as a Service

Offre des logiciels ready-to-use

Platform as a Service

Offre l'environnement technique pour déployer des applications.

Infrastructure as a Service

Offre des machines virtuelles d'instances configurables.

IaaS

Des employés

Développer et déployer

Installer et déployer la pile logicielle (git, java, ...)

Installer et gérer les OS des serveurs

~~Acheter, déployer et maintenir des serveurs~~

~~Attendre le retour sur investissement~~

~~Manuellement ou automatiquement gérer la montée en charge des instances~~

PaaS

Des employés

Développer et déployer

~~Installer et déployer la pile logicielle (git, java, ...)~~

~~Installer et gérer les OS des serveurs~~

~~Acheter, déployer et maintenir des serveurs~~

~~Attendre le retour sur investissement~~

~~Manuellement ou automatiquement gérer la montée en charge des instances~~

SaaS

Des employés

~~Développer et déployer~~

~~Installer et déployer la pile logicielle (git, java, ...)~~

~~Installer et gérer les OS des serveurs~~

~~Acheter, déployer et maintenir des serveurs~~

~~Attendre le retour sur investissement~~

~~Manuellement ou automatiquement gérer la montée en charge des instances~~

Montée en charge

Deux types de scaling :

- Vertical : Améliorer les serveurs
- Horizontal : Ajouter des serveurs

Fournisseurs

Google Cloud Platform

Amazon Web Services

Microsoft Azure

Inconvénients

Le client devient dépendant du fournisseur

Que deviennent les données hébergées ?

Impact sur l'environnement !

Conclusion

Le Cloud est à la mode et se vend très bien !

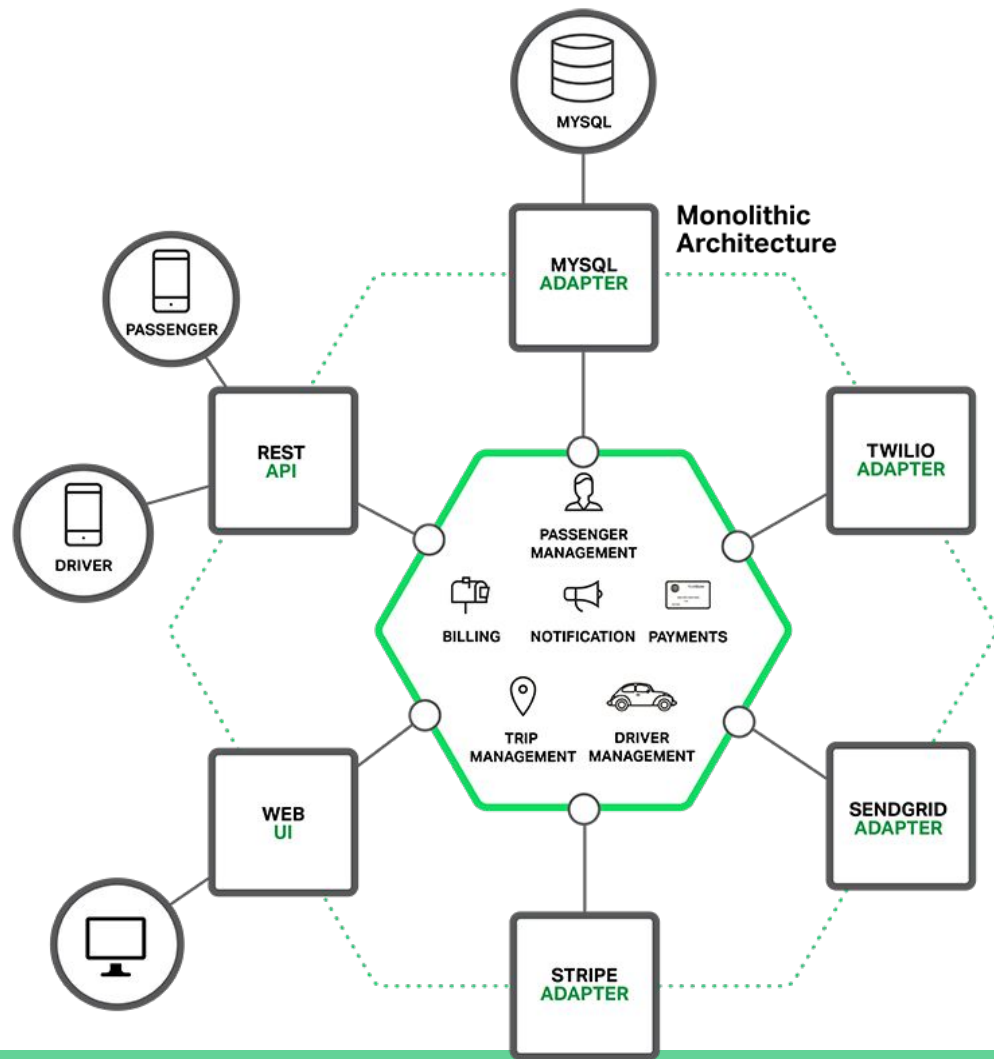
Pratique pour les PME / TPE

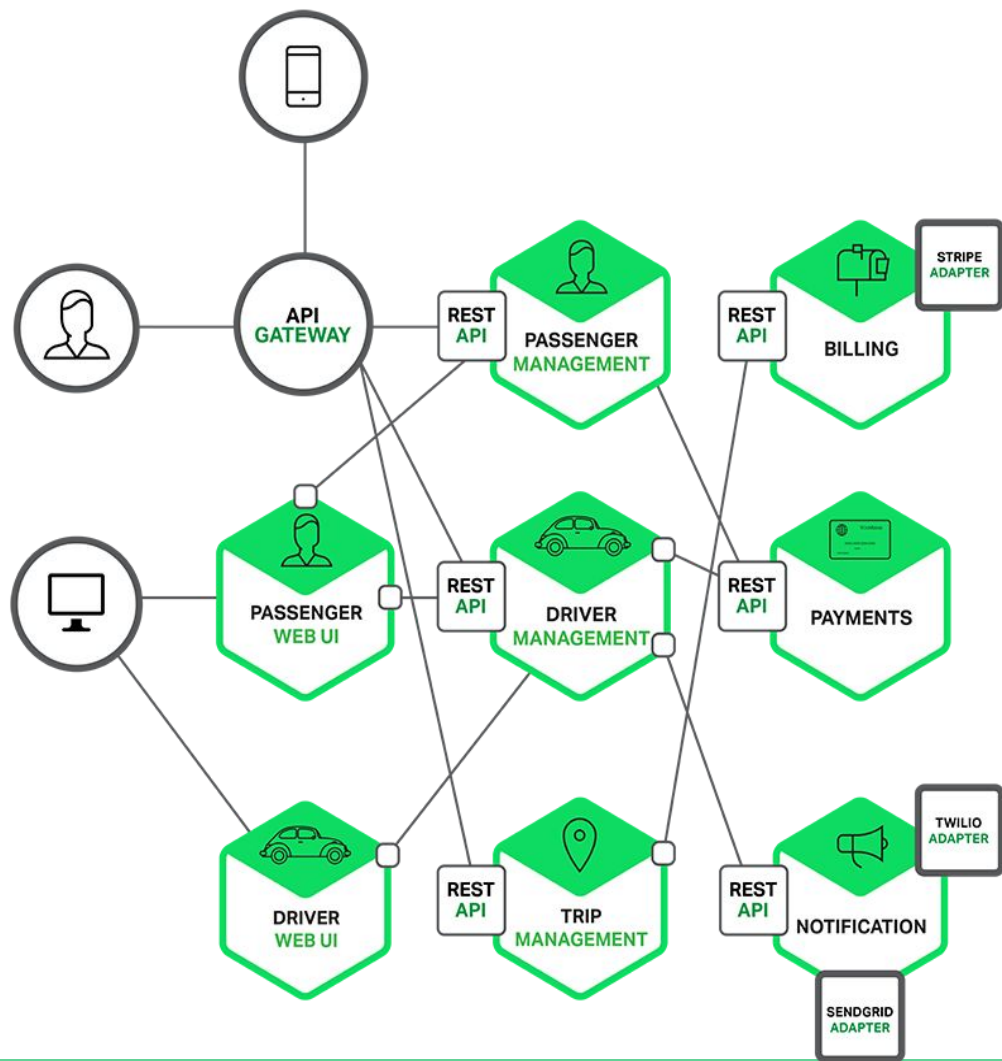
L'effet boîte noire est le principal inconvénient

Monolithe et Microservices

Introduction

Quelle serait l'architecture classique d'une société comme Uber ?







Inconvénients Monolithe

Plus le temps passe, plus le code grossit

- Maîtriser la totalité du code est impossible pour une personne
- Plus de code = start-up time plus long
- Déployer régulièrement fait peur
- Scaler est difficile (parties I/O - CPU bound)
- Un bug affecte tout le système (memory leak ...)

Inconvénients Microservices

Plus le temps passe, plus le nombre de service augmente

- Beaucoup de services à gérer (DevOps, ...)
- RPC entre services / failover
- Bases de données séparées
- Un changement sur plusieurs services

Communication

- Synchrone : Bloquant
- Pub / Sub : Non bloquant, sans réponse
- Pub / Async : Non bloquant, avec réponse

Communication synchrone - avantages

Utiliser REST !

- Simple, facile à tester, a les verbs / response status
- Supporte request/response
- Les firewalls le supporte
- Aucun intermédiaire (broker, ...)

Inconvénients ?

Communication synchrone - inconvénients

- Doit connaître l'URL
- Uniquement request/reponse (pas de notification)
- Timeout

Communication asynchrone

Un client envoie des messages à un channel <name>

1 consumer récupère le message : point-to-point

N consumers récupèrent le message : publish-subscribe

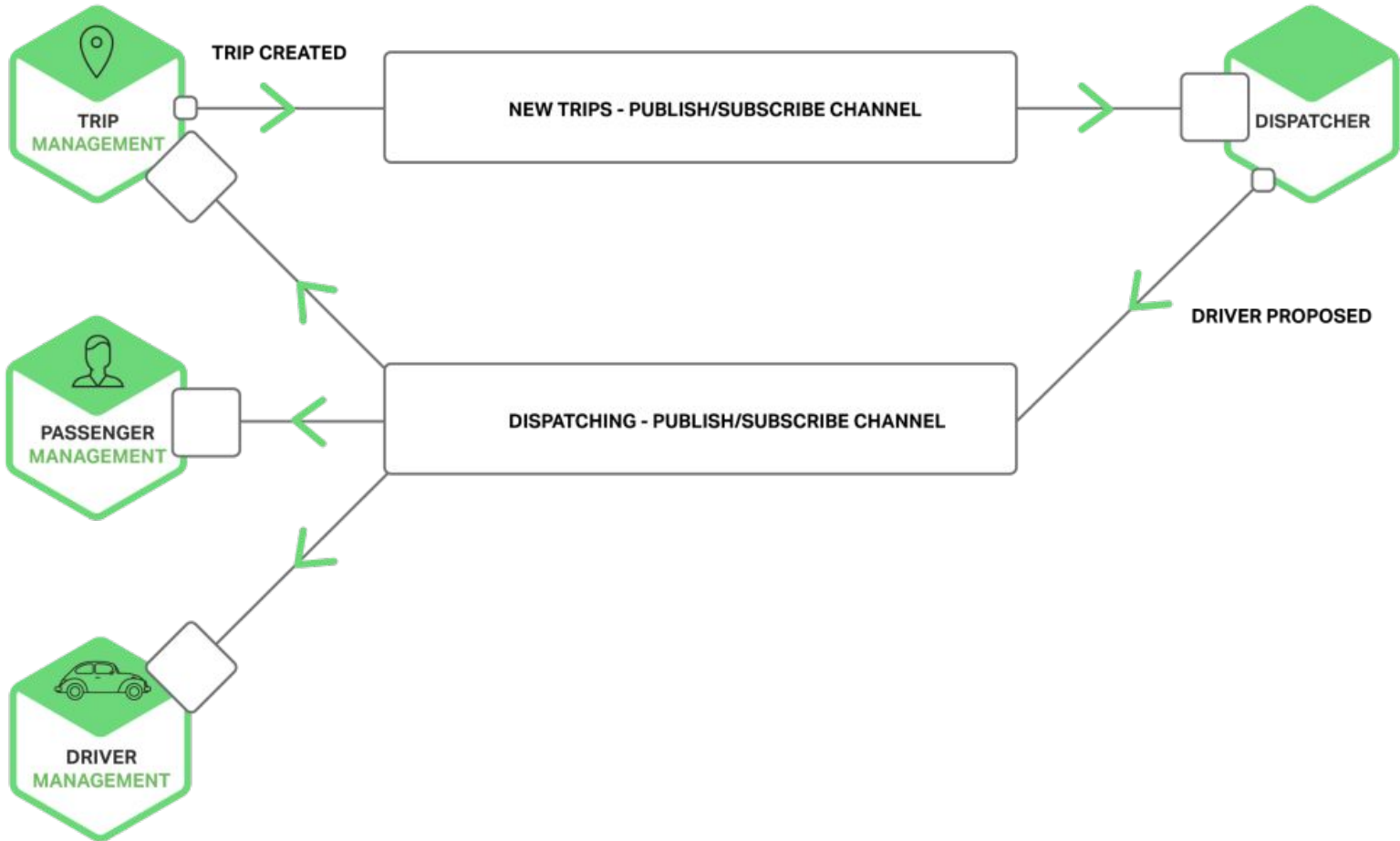
Communication asynchrone - avantages

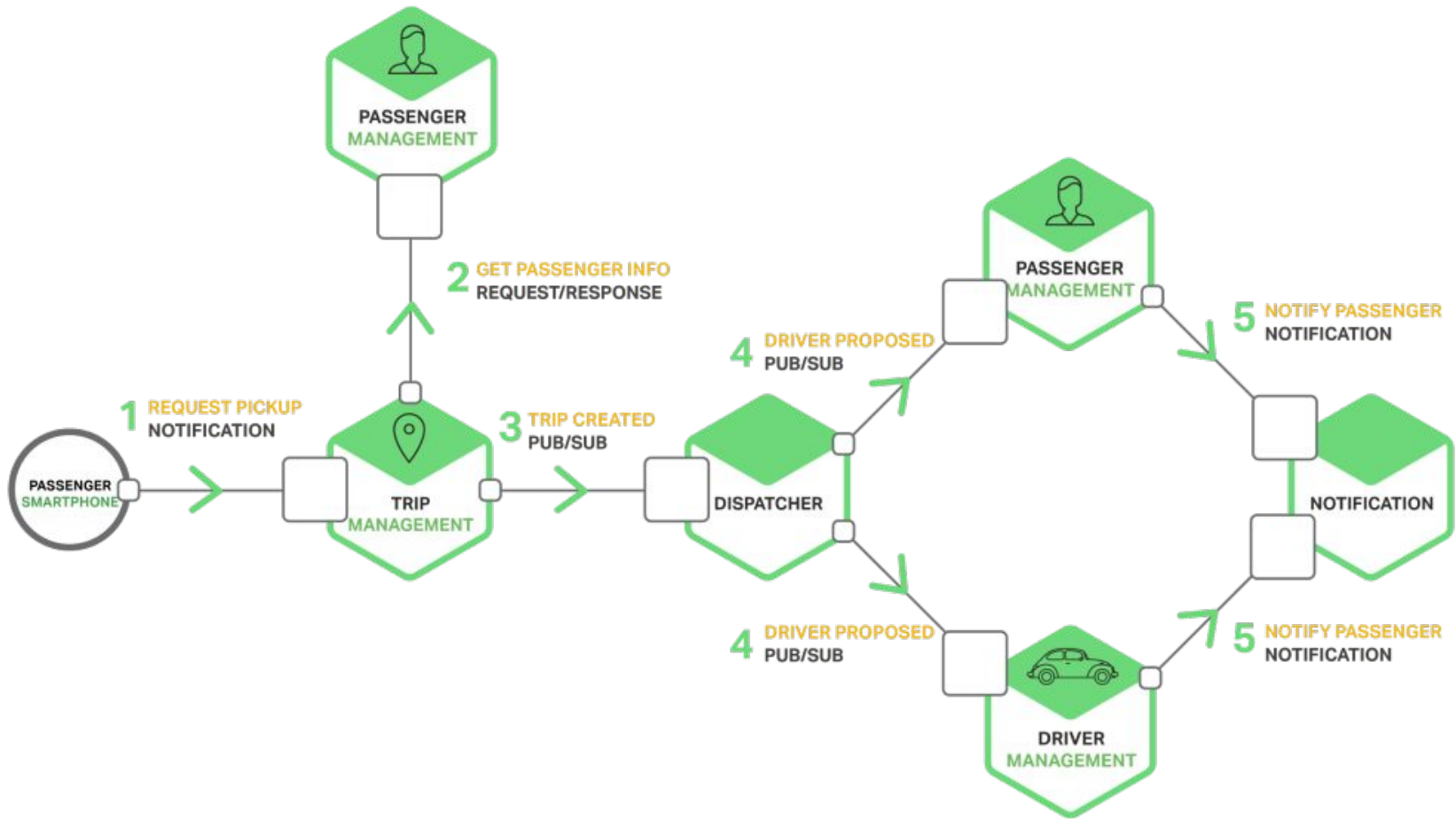
- Découpe client / service
- Client / server n'ont pas à être opérationnels au même moment
- Flexibilité des messages

Inconvénients ?

Communication asynchrone - inconvénients

- Nécessite un broker intermédiaire
- Complexité pour faire request / response

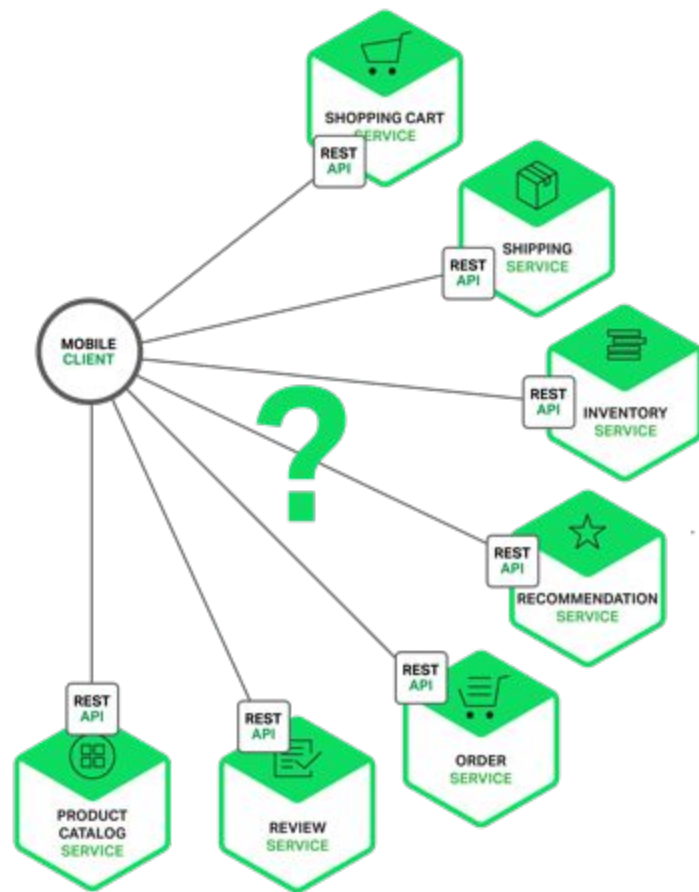




Gateway

Pourquoi utiliser une gateway ?

Quelles sont les autres possibilités ?



Gateway

- Agrégation de résultat
- Une seule entrée pour tous les clients

Quels sont les inconvénients ?

Gateway : Service Invocation

Quels sont les moyens de communication entre la gateway et les services ?

Gateway : Service Discovery

Comment déterminer où sont situés les services ?

Gateway : Handling failures

Que faire si un service n'est pas disponible ?

Gateway : Handling failures

- Network timeout : ne jamais bloquer indéfiniment. Utiliser un timeout.
- Rate limit : Imposer un nombre de requêtes / client
- Circuit breaker pattern : Après N fails, on considère le service down
- Fallbacks : Renvoyer des données en cache / valeur par défaut

Communication : Versioning

Les clients ne se mettent pas à jour.

Le versioning de services garde un état pour chaque version de l'appli

Comment s'y prendre ?