

Séance 10

Cloud computing et *aaS



Ce(tte) œuvre est mise à disposition selon les termes de la Licence Creative Commons Attribution – Pas d'Utilisation Commerciale – Pas de Modification 4.0 International.

Rappels

- Comprendre les architectures pour **traiter des données**

Définition de composants et de l'orientation données

- Architectures orientées **flux de données**

- Batch séquentiel, pipe et filtre et pipeline
- Contrôle de processus et boucle de rétroaction

- Architecture **centrée sur les données**

- Données centralisées au sein d'un store
- Dépôt actif ou blackboard passif

Objectifs

- Services offerts pour le **cloud computing**
 - Définition et fonctionnement du cloud computing
 - Déploiement et distribution d'applications
- Analyse et comparaison de différents **modèles de services**
 - Infrastructure et plateforme comme services (IaaS, PaaS)
 - Software et backend pour applications (SaaS, BaaS)
 - Offre de fonctions comme service (FaaS)

Cloud computing



Cloud computing (1)

- Modèle/méthode de calcul **basée sur Internet**

Fournit ressources de calcul et de données à la demande

- Accès à des **ressources configurables** sur le réseau

- Ressources omniprésentes, pratiques et à la demande
- Mobilisées rapidement, effort minimal côté user
- Interaction minimale avec le fournisseur de service

- Évolution de technologies changeant **infrastructure IT**

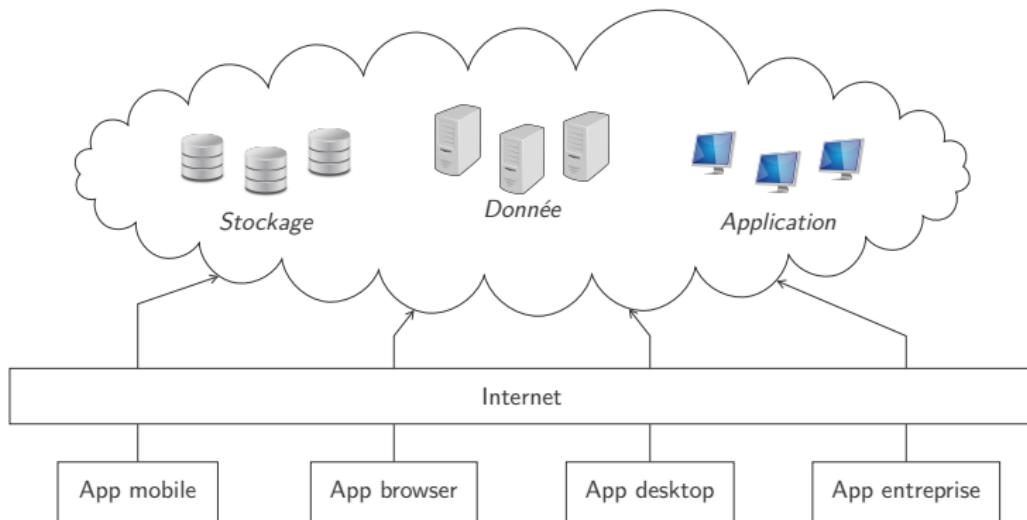
Du cloisonné et isolé au réparti accessible

Cloud computing (2)

- Technologies utilisées en cloud le sont **depuis longtemps**
Mais rendues accessibles à la masse sous le nom cloud computing
- **Basé sur internet**, mais beaucoup plus qu'internet
Cloud est l'endroit où l'on retrouve la technologie qu'on veut
- Rien à installer sur sa machine pour **exploiter le cloud**
Et pas besoin de payer pour la technologie lorsqu'utilisée

Définition (1)

“A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end-customer applications and billed by consumption.” — Forrester



Définition (2)

- Computer paradigm avec large pool de systèmes connectés
 - Connexions sous forme de réseaux publics ou privés
 - Fournit infrastructure dynamiquement adaptable
 - Propose applications, données et stockage
- Réduction des couts par capacités IT réutilisables

Calcul, hébergement d'app, stockage contenu, distribution
- Propose aussi bien du software que de l'infrastructure

Application comme Gmail ou infrastructure comme Heroku

Cloud Computing Model du NIST

- Modèle du National Institute of Standards and Technology
 - Cinq caractéristiques essentielles
 - Quatre modèles de déploiement
 - Trois modèles de service
 - Douze éléments/briques de base



Service

- Offre de **services cloud** software ou hardware

S'asseoir derrière n'importe quelle machine et accéder au service

- **Trois mesures** principales pour identifier un service cloud

- Service accessible par un browser web ou une API web service
- Aucun capital d'investissement initial pour démarrer
- On ne paie que ce qu'on utilise

- **Trois grandes catégories** de services fournis sur un cloud

Software/Platform/Infrastructure as a Service

Caractéristique (1)

- Le cloud offre un **self-service à la demande**

Accès direct aux capacités de calcul sans intervention humaine

- Accès très large et **via le réseau** aux services cloud

Hardware que software, par laptop, desktop, mobile, tablette...

- Mise en commun de ressources sous forme d'un **pool**

- Servir plusieurs clients avec modèle multi-locataire
- Affectation dynamique selon les besoins des clients
- Indépendantes de la localisation, transparence pour client

Caractéristique (2)

- Très grande élasticité et adaptation rapide

Scaling in/out très rapide, ressources semblent illimitées

- Mesure, contrôle et optimisation automatique des ressources

Paiement d'uniquement ce qui est consommé

Visibilité

- Trois niveaux de **visibilité** des services cloud

Public, privé ou hybride à déterminer par le cloud integrator

- **Cloud public** géré par des tiers, modèle “*pay-as-you-go*”

Même pool et config limitée, protection sécurité, dispo variable

- Meilleure sécurité dans une entreprise avec un **cloud privé**

- **Local** dans datacenter de l'entreprise

- Hébergé en **externe** chez un fournisseur de cloud

- Solution **hybride** privée et peut augmenter à l'extérieur

Avantage

- Réduction des couts par rapport à ressources inexploitées

Paie que ce qu'on utilise, pas d'infrastructure à maintenir

- Espace de stockage augmenté très fortement

Stockage, maintenance, backup et accès à quantité de données

- Maximum de flexibilité permettant adaptation

Blocs de base pour développement fournis directement par cloud

Challenge

- Protection des données contre compétiteurs et pour clients

Assurances, contrats bétonnés, mécanismes de protection

- Récupération et disponibilité des données

*Service level agreement, clustering, data replication, monitoring
disaster recovery, management capacité et performance*

- Capacité de gestion avec auto-scaling, load balancing...

- Restrictions de réglementations et conformité

Par exemple, données doivent être physiquement dans le pays

Aujourd'hui



amazon drive

Microsoft Azure



openstack.



Akamai


basho


阿里云


CLOUD FOUNDRY


DigitalOcean


salesforce


Cloudify


HEROKU


OPENSHIFT


OVH.com

IBM SmartCloud, Oracle Cloud, etc.

IaaS



Infrastructure as a Service (IaaS)

- Modèle **Infrastructure as a Service (IaaS)**

Fournit du stockage de base et de la capacité de calcul

- Mise en commun de **ressources hardware**

Serveurs, système stockage, équipement réseau, datacenter

- Client déploie **son propre software** sur l'infrastructure

Typiquement modèle de la forme “pay-per-use”



HEROKU

Microsoft Azure



Responsabilité

- Client doit installer et tenir à jour le **software**
Application, système d'exploitation, librairies, données...
- Fournisseur gère le **hardware et l'infrastructure**
Serveur, virtualisation, disques, stockage, réseau...
- IaaS+ offre quelques **services en plus** sur la virtualisation
Base de données, file de messages...

Architecture (1)

- Architecture composée de **quatre couches**

De la gestion des clients jusque celle des machines virtuelles

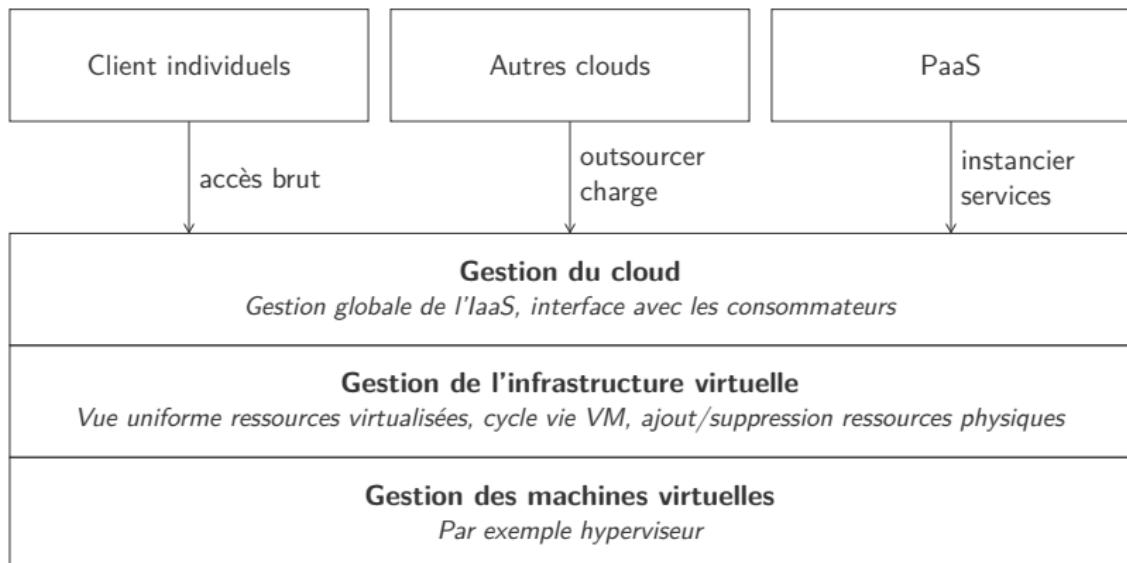
- Couches du haut gèrent **les clients** et lien avec clients

Gestion globale de l'IaaS et interface client

- Couches du bas gèrent les **machines virtuelles**

Vue uniforme des ressources virtualisées et hyperviseurs...

Architecture (2)



Cas d'utilisation

- **Augmentation temporaire** infrastructure datacenter

Par exemple pour faire face au trafic supplémentaire à Noël



PaaS

Platform as a Service (PaaS)



- Modèle **Platform as a Service** (PaaS)

Fournit une couche logicielle ou environnement de développement

- Client construit **son application** pour offrir services haut niveau

Application exécutée sur l'infrastructure du fournisseur

- Fourniture d'une combinaison d'**OS et applications serveur**

Typiquement modèle de la forme “pay-per-use”



Responsabilité

- Client doit développer, tester et maintenir l'**application**
From scratch, customiser app existante, configuration
- Fournisseur offre une **plateforme de développement**
Langage, framework, librairie, système d'exploitation
- De nombreux **outils et librairies** sont proposés
 - Outils de développement, test et déploiement
 - Héritage mise à l'échelle, haute disponibilité...

Exigence (1)

- Haute scalability et hardware **à la demande**

Automatique sans intervention IT, impression ressources infinies

- **Haute disponibilité** du runtime pour les applications

Basculement transparent grâce à du load balancing

- **Haute fiabilité** pour assurer l'exécution des applications

Service pas vampirisé par un autre qui prend toutes les ressources

- Garantir une **utilisation optimale** des ressources

Monitored la consommation des applications

Exigence (2)

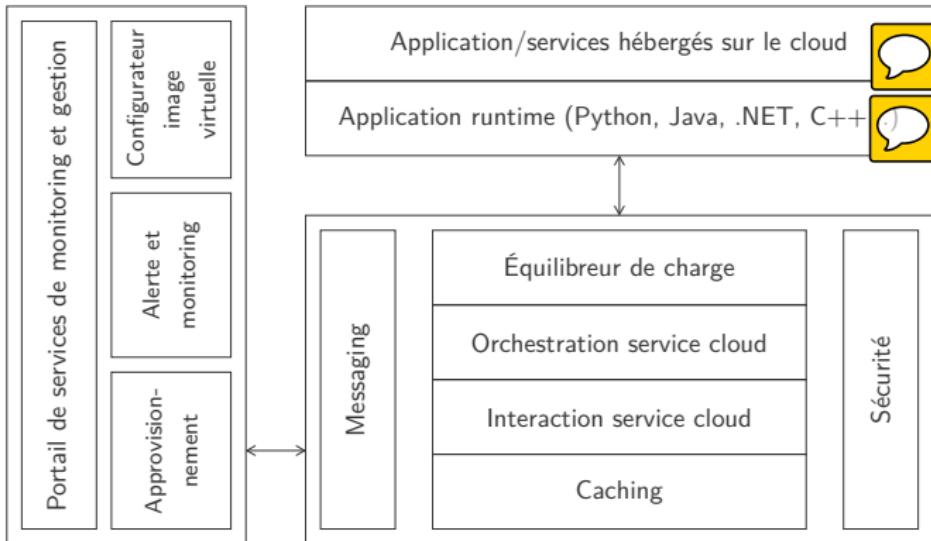
- Mise à l'échelle automatique selon la demande
Cloud élastique qui s'adapte tout seul à la charge
- Console admin/de gestion avec présence de rapports
Traçabilité, monitoring, suivi, reporting...
- Support pour plusieurs systèmes d'exploitation et langages
Windows, Linux... Python, Java, .NET, C++...

Architecture (1)

- Couche de **communication** relais entre runtime et hardware
 - **ÉquilibrEUR de charge** gère disponibilité, fiabilité, utilisation optimale des ressources
 - **Orchestration service** stocke résultats intermédiaires services lorsque business process besoin plusieurs
 - **Interaction service** retient où est exécuté chaque service
 - **Cache** commun pour tout le cloud
 - **Messaging** fiable entre composants et **sécurité**
- Couche de **gestion** du PaaS (overlap avec gestion IaaS)

Approvisionne en service, gère les alertes et monitore

Architecture (2)



Cas d'utilisation



- **Augmentation temporaire** infrastructure datacenter

Par exemple pour faire face au trafic supplémentaire à Noël

SaaS

Automap Universal

Name: Reaktor2Bas
Group: Inst.

Reaktor5 - Native Instruments

Preset 1/128: <empty>

Learn:
Page 1/



Software as a Service (SaaS)

- Modèle **Software as a Service** (SaaS)
Fournit une application complète au client
- Exécution d'une **unique instance** du service sur le cloud
Service de plusieurs clients sur la même instance
- Plus besoin d'investir en **serveur front-end et licences**
Accès direct pour le client une fois authentifié chez le fournisseur



G Suite



Responsabilité

- Client accès à l'application depuis un **web browser**

Après authentification et aucune installation particulière

- Fournisseur donne l'accès à des **applications via le web**

Gère la maintenance, mise à jour, la plateforme et l'infrastructure

- Permet de la **collaboration**, relation client...

Fournisseur doit parfois fournir des plugins

Avantage

- “Pay-as-you-go” en prenant une **souscription** par utilisateur

Diminution du cout pour l'utilisation faite d'un logiciel

- Meilleure **infrastructure réseau**

Fournisseur peut se permettre grosse infrastructure pour le client

- Meilleur contrôle centralisé contre le **piratage**

Désactivation ID customer frauduleux, pas de reverse engineering

- **Maintenance** software facilitée car environnement contrôlé

Une seule mise à jour lorsque serveur évolue

Désavantage

- Cout à long terme peut devenir très élevé
Surtout si on ajoute pleins de services additionnels
- Manque d'implication des équipes IT lors d'un achat
Fournisseurs vont directement contacter les business users
- Apparition de vendeurs éphémères dans marché maturité
- Problème d'intégration avec les systèmes et données existants
- Verrouillage des fournisseurs pour empêcher aller concurrence
- Plus (moins de) contrôle sur les changements software

Cas d'utilisation

- Remplacer software traditionnel à installer sur les machines

Accès direct via un web browser à une application complète

BaaS



Backend as a Service (BaaS)

- Modèle **Backend as a Service** (BaaS)
Fournit un lien avec stockage cloud, API backend
- Utilisation pour développement d'**applications web et mobile**
Aide pour gestion des users, notifications push, réseau sociaux
- Services complémentaires offerts par **SDKs et APIs** custom
BaaS est pont entre front-end et backend sur cloud



Firebase



Service

- Services de base offerts par la plupart des BaaS

Notification push, stockage/partage fichiers, base de données

- Intégration avec des réseaux sociaux

Facebook et Twitter, service de messagerie et chat

- Gestion d'utilisateurs et analyses de comportement

Localisation, logique business, outils d'analyse

FaaS

+ (z) o

Function as a Service (FaaS)

- Modèle **Function as a Service** (FaaS)

Gestion de fonctionnalités sans l'infrastructure associée

- Une manière de construire une **architecture “serverless”**

Notamment utilisé dans la conception d'applications microservice

- Service doit démarrer **dans la ms** pour gérer une requête

Contrairement à PaaS où un process tourne en permanence



AWS Lambda



Google Cloud Platform

GC Functions

Microsoft Azure

MA Functions

Exemple

- Définition d'une **fonction** dans faas.js

```
module.exports = function (ctx, cb) {  
    cb(null, 'Hello from a Function');  
}
```

- Transformation en un **service**

```
npm install -g wt-cli  
wt init  
wt create faas.js
```

- Déploiement comme **FaaS**

<https://webtask.it.auth0.com/api/run/tjanczuk/faas>

A photograph showing the backs of two security officers. They are wearing bright orange long-sleeved shirts with the word "SECURAIL" printed in white across the back. They also have black vests with shoulder straps and black belts with multiple pockets and attachments. The officer on the left is partially visible on the far left edge of the frame. The background shows an indoor setting with windows and a metal fence.

SECURAIL

SECURAIL

SECaS

Security as a Service (SECaaS)

- Modèle **Security as a Service** (SECaaS)

Services de sécurité offerts à travers le cloud

- Pas de **nécessité matérielles** évitant des dépenses

Authentification, anti-virus/malware, détection d'intrusion...

- Protections complémentaires pour **applications sur Internet**

Attaques comme DDoS, déploiement HTTPS...



Crédits

- Photos des logos depuis Wikipédia
- <https://www.flickr.com/photos/incredibleguy/6937250243>
- <https://openclipart.org/detail/94723/database-symbol>
- <https://openclipart.org/detail/155101/server>
- <https://openclipart.org/detail/6678/lcd-screen>
- <https://www.flickr.com/photos/viriyincy/7004881064>
- <https://www.flickr.com/photos/henriquev/6818926540>
- <https://www.flickr.com/photos/createdigitalmedia/2211729651>
- https://www.flickr.com/photos/chris_wilson/3625120346
- <https://www.flickr.com/photos/vestman/4908148942>
- <https://www.flickr.com/photos/followtheseinstructions/5428495287>