

## Définition C.C.:

Système fournissant un accès à des ressources.

accès { à la demande : disponible.  
pratique : utile.  
ubiquitaire : accès de partout.

ressources { partagés : possibilité d'accès aux m. ressources par plusieurs persn  
toujours disponible : à la demande.  
accessible par un réseau : ex : internet.  
facilement { rapide → processus rapide.  
extensiblement { sans effort du client.  
sans investissement du fournisseur.

## Propriété:

→ tous clients est le bienvenu.

→ libre service (self-service)

→ ressources communes.

→ la location au lieu l'achat

→ abstraction de localisation

→ service mesuré.

⚠ quand on a le choix nous choisissons C.C car il est moins cher, donne la possibilité du travail collaboratif, ainsi qu'il ne consomme pas de ressource matériel.

→ l'infrastructure du cloud c'est l'ensemble du matériel permettant le déploiement.

## \* Modèles de déploiement \*

### → Modèle public:

utilisé par le grand public, gérée par une ou + entreprise.

\* on veut dire par l'utilisation de l'infrastructure du C.C

l'accès au service virtuelle mais sans utiliser le matériel.

\* gestion de l'infrastructure : gérer la sécurité, changer les



paramètres réseau ... (la gestion peut être technique comme elle peut être informatique).

### → Modèle privé :

utilisé uniquement par une entreprise donnée, gérée par l'entreprise utilisatrice ou une entreprise tierce (pas par le fournisseur pour l'infrastructure).

- \* le fournisseur gère le service

	Modèle public	Modèle privé
contrôle	minimal (inexistant)	personnalisable
<del>coût</del> <sup>coût</sup>	Réduit	important
Disponibilité	<u>Moins</u>	<u>Plus</u>
Maintenance	Nul besoin	Rôle de l'entreprise gérante
Sécurité	<u>Réduite</u>	<u>Maximale</u>
infrastructure	quasiment <u>illimité</u>	physiquement <u>limité</u>

### → Modèle hybride :

mélange de deux ou plusieurs modèles de déploiement diff.

#### \* Niveaux de déploiement \*

### → IaaS : infrastructure-as-a-service :

Accès à des ressources informatiques permettant de déployer des programmes (ex: sys d'exploitation, logiciel ... etc).

les ressources accessibles par les utilisateurs sont : RAM, CPU, processeur ...

app commune : machine distante, espace de stockage.

### → SaaS : Software-as-a-service :

accès à une application qui s'exécute sur un nuage info.

exemples : client Mail (outlook).

Réseau social (Facebook).



les ressources accessibles par user dans SaaS sont :

→ **Platform-as-a-Service: PaaS:**

Accès à un ensemble d'outils permettant au user la création de ses app à lui.

\* App habituelle :

→ Middleware, serveur web, outils de dev, Bdd.

les ressources accessible par user dans PaaS sont :

→ Comparaison par des exemples.

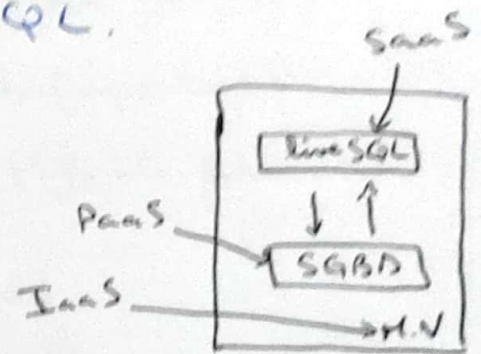
\* SAAS : interpréteur de commande SQL.

\* IAAS : serveur de base de données

\* PAAS : SGBD complet.

⚠ Heroku : PaaS.

google drive : SaaS



les service C.C de diff nu de déploiement peuvent être imbriqués, intérêt d'avoir une large échelle d'utilisation.

## **Cluster Computing**

- 1<sup>er</sup> ordinateur par Alan Turing: 1936 (recherche)
- 1<sup>er</sup> ordinateur purement électronique: 1948: Manchester Baby.
- ordinateur à programme stocké.

↳ Problème : ordinateurs peu performants, prix exorbitant.

↳ Solution : regrouper les ordinateurs pour former un seul ordinateur plus puissant appelé Groupe d'ordinateurs (cluster)

→ **Computer cluster:**

un groupe d'ordinateurs (slave nodes) relié par LAN

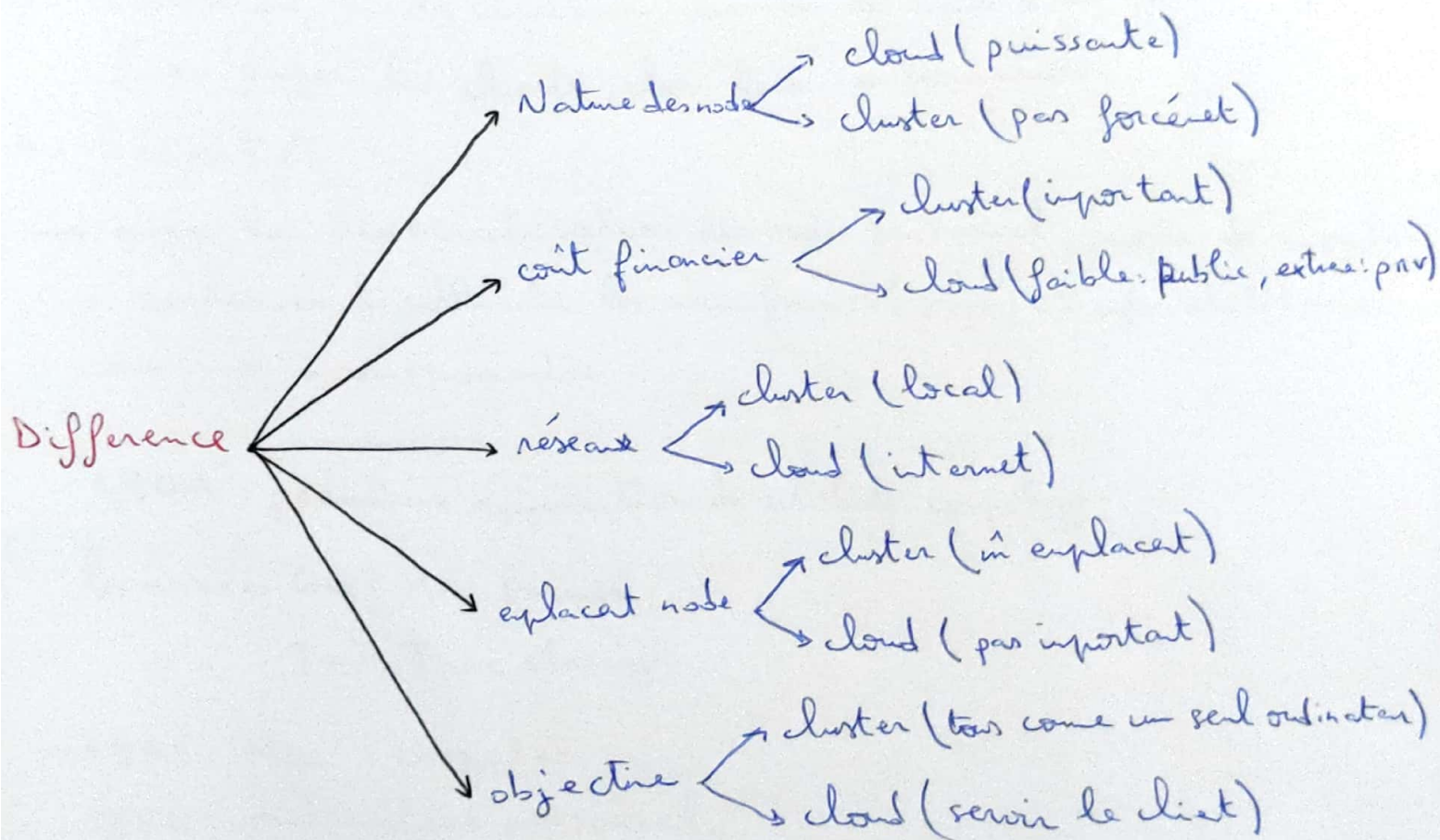
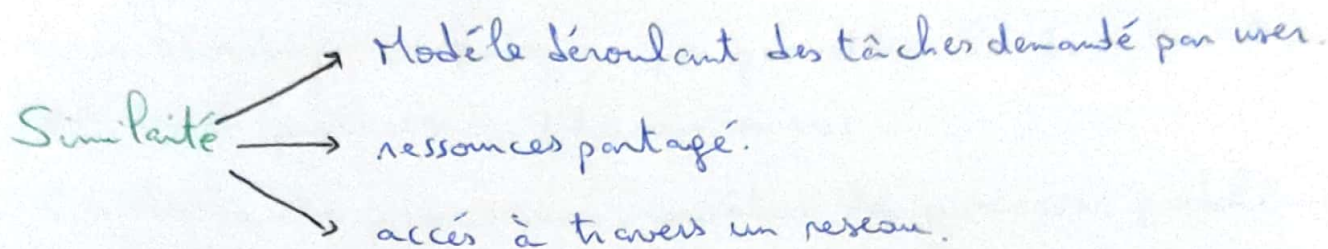
contrôlés par root node → grâce à un middleware.

↳ Role : orchestrer les tâches.



- \* middleware aide à segmenter les tâches.
- \* si un nœud slave tombe en panne les tâches seront interrompues (aucun résultats ne sera réalisé)

→ Cluster Vs cloud:



outil existant pour cluster: Rocks 7.0, version de Linux.

### Utility computing

utile linguistiquement: qui est profitable, avantageux, qui sert à qdq chose.

Rôle: c'est un modèle d'approvisionnement de service ou ressources, il donne accès à:

→ des ressources informatiques

→ gestion du matériel.



payé selon l'utilisation

Mainframe: c'est ordinateur dédié pour grandes entreprises utilisé pour les processus critiques, économise l'espace et l'effort de gestion (en remplaçant plusieurs serveurs). (UNIX à Chatouane)

- 1 - **Time sharing**: technique qui permet de faire tourner de + tâche sur un m. processeur. (les années 70)
- 2 - **Contrôle de processus**: arrêter le processus problématique.
- 3 - **Mesure de l'utilisation**: mesurer le taux d'utilisation afin de faire payer les clients selon leur consommation.
- 4 - **Sécurité**.

→ avec la démocratisation des ordi personnel, moins de monde a continué à solliciter les mainframes (payé) => info utilitaire commence à disparaître.

1961: première apparition de utility computing.

les années 60: Mainframes

" 70: Time sharing.

1963: Mini-computer

1973: ordinateur personnel.

1997: InsyQ lance des app à la demande.

1998: HP ouvre dep que pour utility info.

2000: Sun annonce Sun Cloud

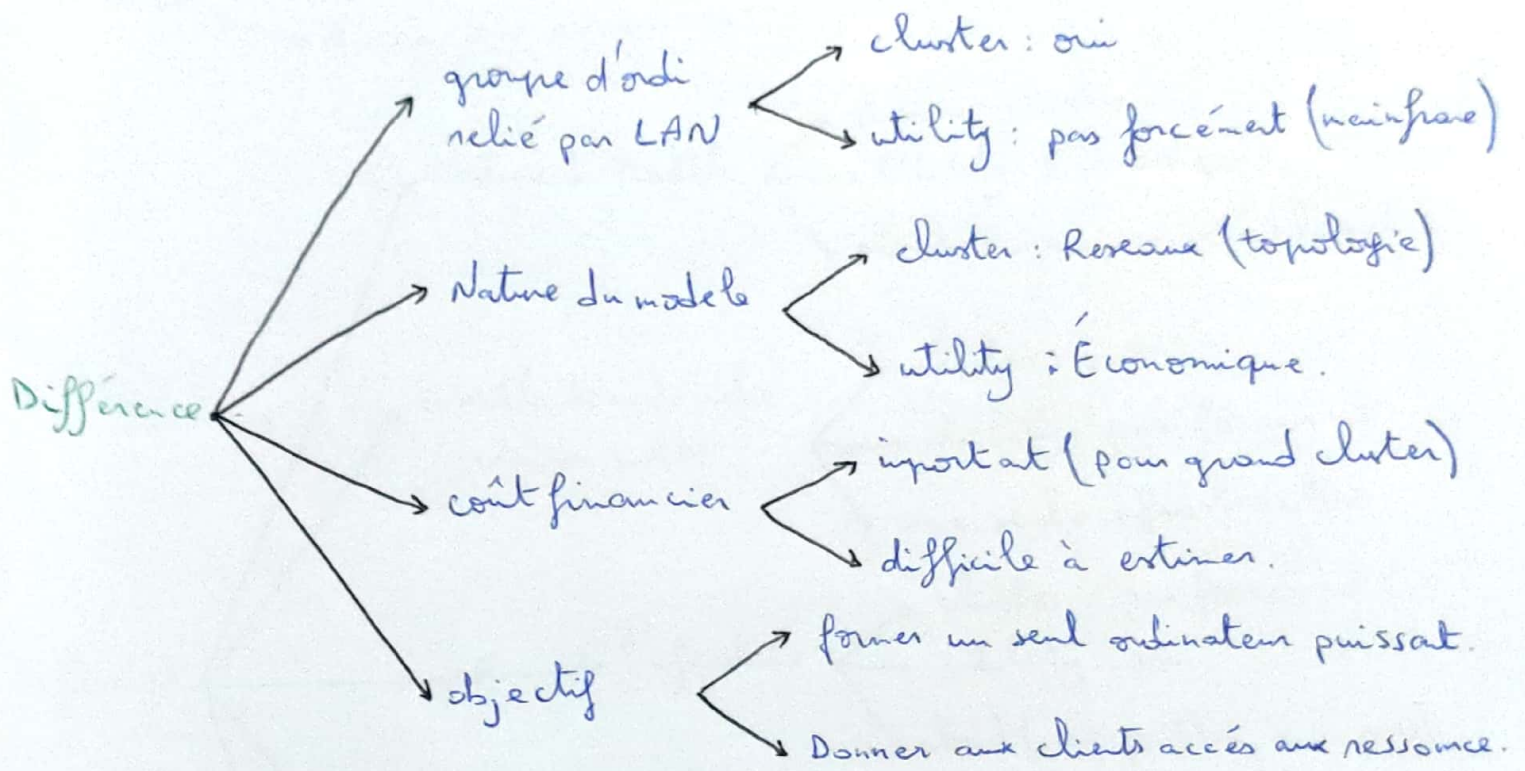
2001: HP introduit: centre de données utilitaire.

2005: Alexa Web Search platform.

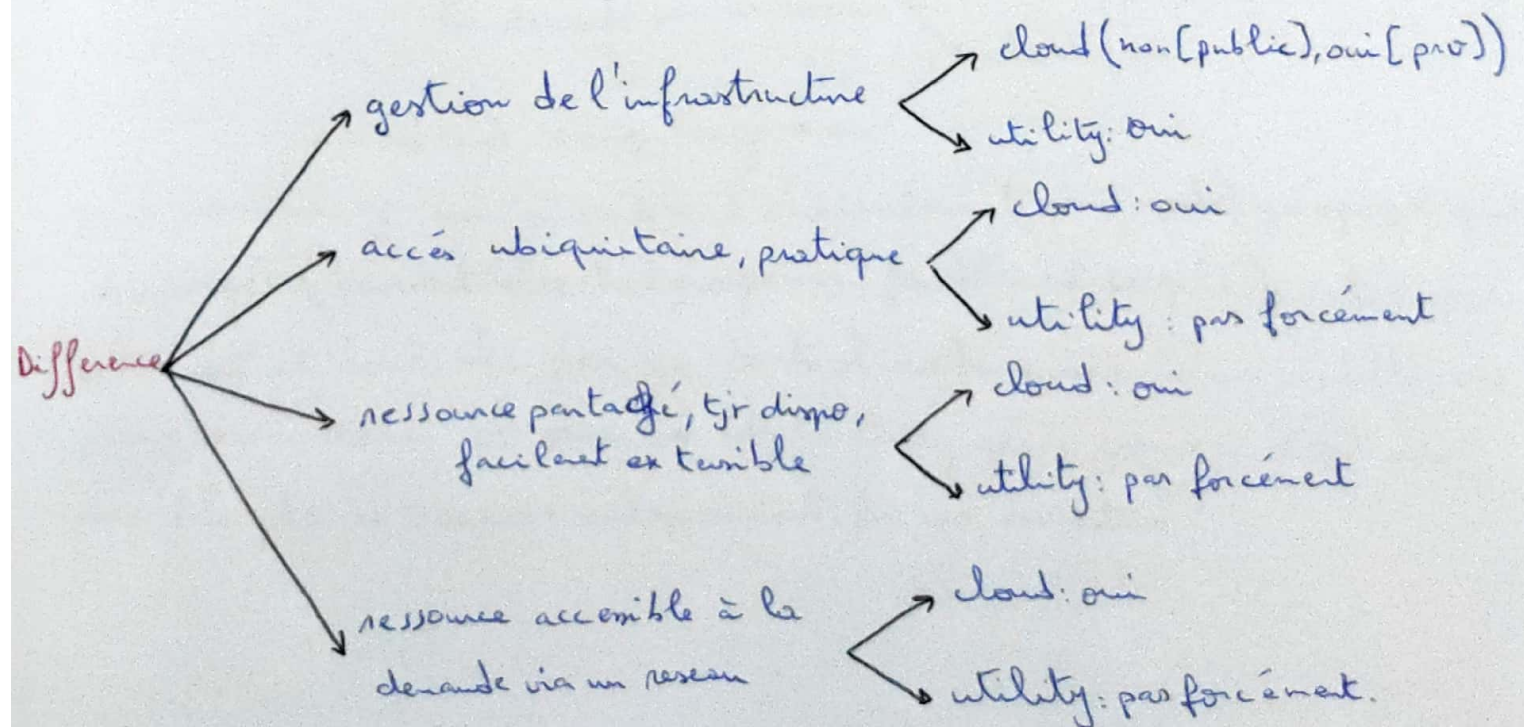
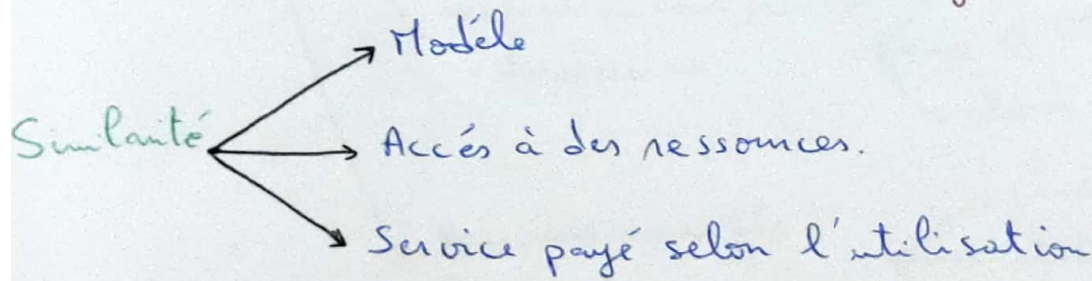
2006: App logic par 3tens, Amazon EC2 (Elastic Compute Cloud)



- Similitude avec cluster: ~~deux~~
- Modèle
- Accès à des ressources.

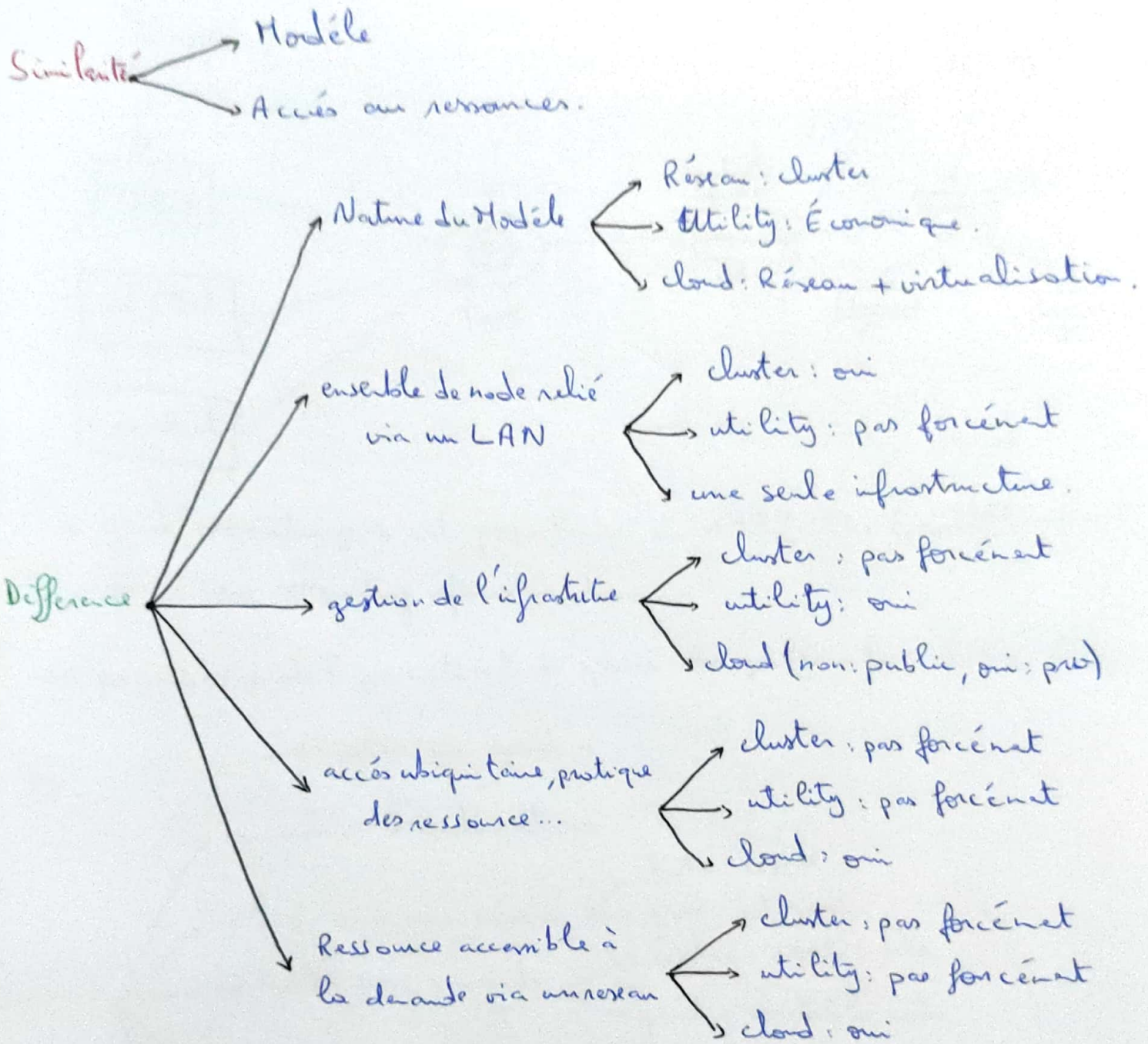


→ utility computing vs cloud computing





# utility vs cluster vs cloud

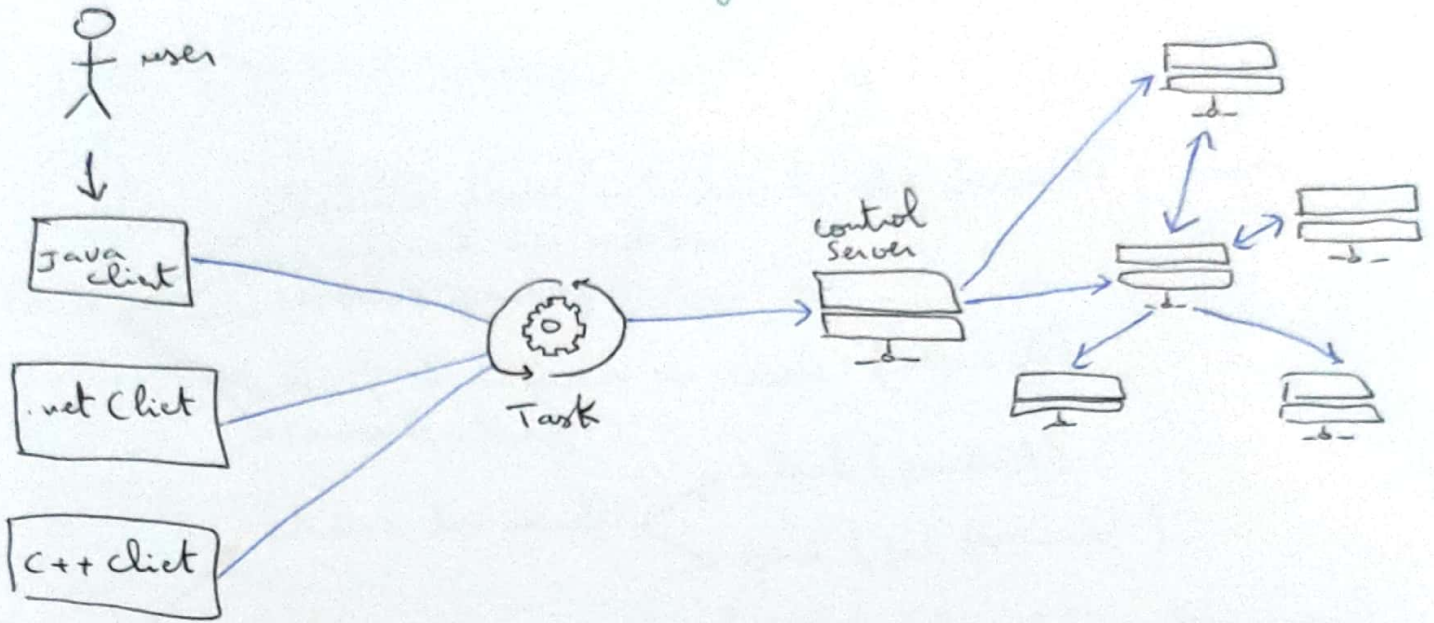


## \*\* Grid computing \*\*

grid computing: un ensemble d'ordinateurs (grid node) géographiquement dispersés (peuvent être hétérogènes, faiblement couplés), reliés par internet et contrôlés par un control node grâce à un middleware pour orchestrer et orienter des tâches, gérer comme étant un ensemble de ressources indépendantes les uns des autres.



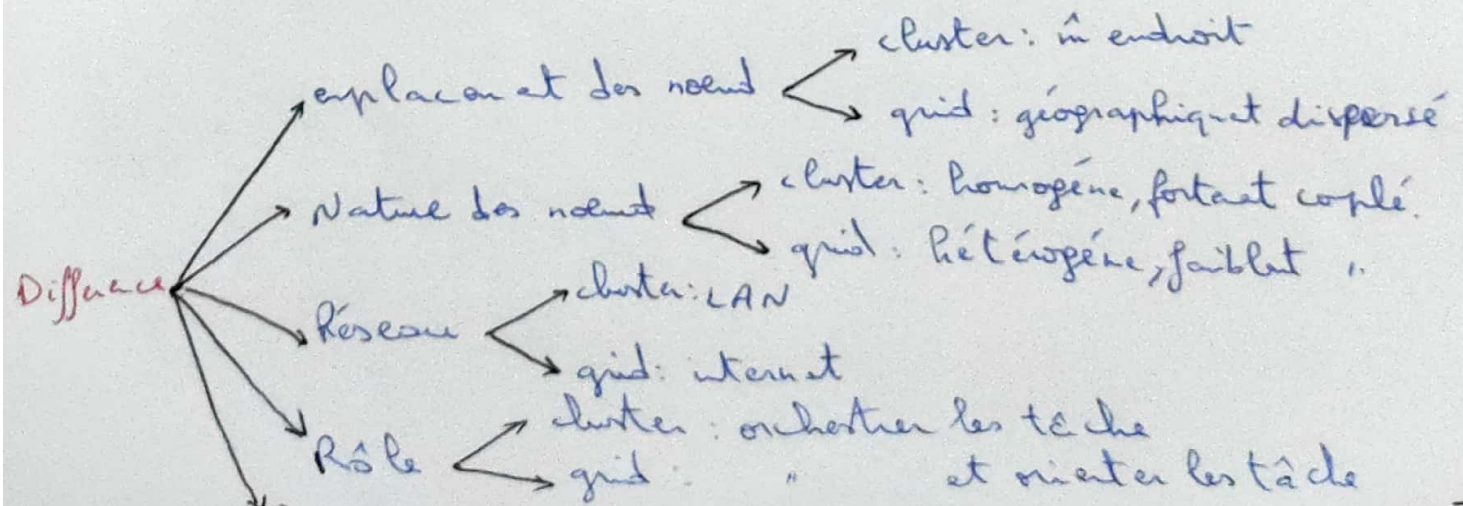
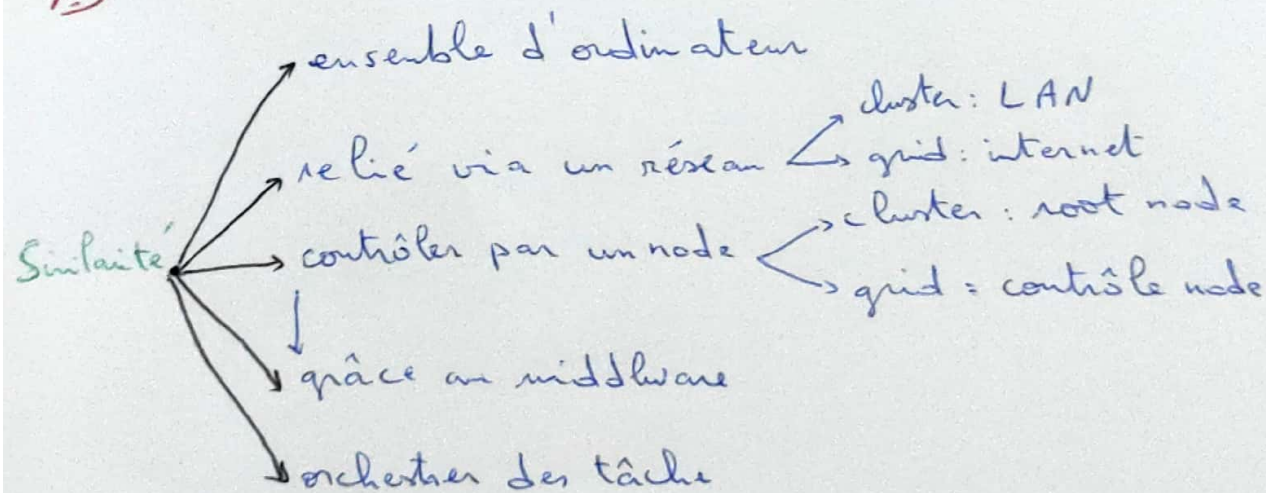
# Fonctionnement du grid computing:



→ grid computing s'est popularisé en 1993, The Grid (Blueprint for a new computing infrastructure).

→ premier outil permettant de gérer les grille informatique: BOINC

## - cluster vs grid -





Difference

gérer comme état

cluster: un seul ordinateur puissant

grid: ensemble de ressources indépendantes

- grid vs cloud -

Similarité

Modèle déroulant des tâches demandées par users

emplacement des nœuds

ressource partagée

accès à travers un réseau (internet)

servir le client

Nature des nœuds

cloud (puissant)

grid (pas forcément)

Difference

coût financier

cloud (faible: public, externe: privé)

grid (

- grid vs cloud vs cluster -



## \*\* La virtualisation dans le C.C \*\*

La virtualisation c'est la création d'une réplique virtuelle de quelque chose d'existant.

→ Problème: défi C.C.:

- \* Nombre important de serveur physique:

  - difficulté de gestion.

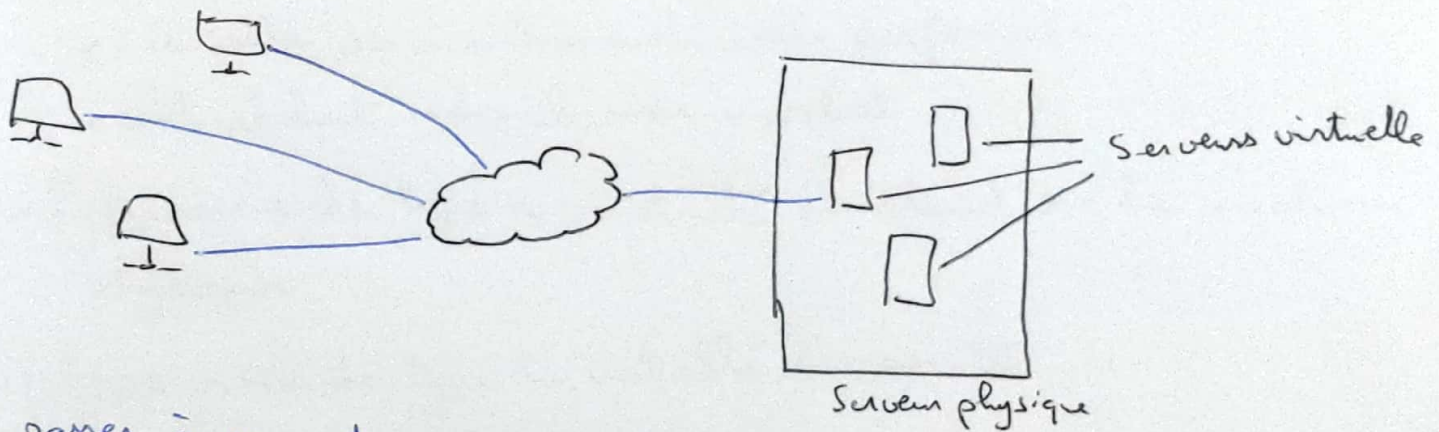
  - énergie importante consommée.

  - coût matériel important.

- \* Complexité.

- \* faible tolérance aux pannes.

→ Solution: Architecture virtualisée:



⚠ passer à un autre serveur virtuelle en cas de panne d'un serveur virtuelle donnée même à la vérification de la caractéristique: Toujours disponible (C.C.)  
Remarque





→ Définition de la virtualisation dans le C.C.

Mécanisme informatique permet l'exécution de + SE dans un seul serveur physique en  $n$  temps. (c'est un concept clé dans C.C.)

→ **Avantage de virtualisation:**

- \* tolérance aux pannes.
- \* Reprise plus rapide
- \* gestion des machine virtuel (plus facile, rapide)
- \* pour C.C : optimisation des ressources.

C.C. a besoin d'un outil utilisant hyperviseur pour faire mutualisation des ressources auto

→ **Inconvénients:**

- \* consommation des ressources supp
- \* nécessite des machine matérielle performante.
- \* coût réduit mais il reste important.

→ Hyperviseur de type 1: installé directement sur la machine physique

→ Hyperviseur de type 2: installé sur un SE.

⚠ on peut pas installer 2 hyperviseurs dans la même machine physique car cet hyperviseur prend le contrôle <sup>type 1</sup> totale de la machine physique.

**\*\* Architecture et gestion d'image info \*\***

→ **Service Level Agreement:**

Contrat entre fournisseur et client.

L> qualité	} les services doivent respecter le contrat.
L> disponibilité	
L> responsabilité	

\* exemple de responsabilité du client: paiement à jour, lire et accepter les droit.



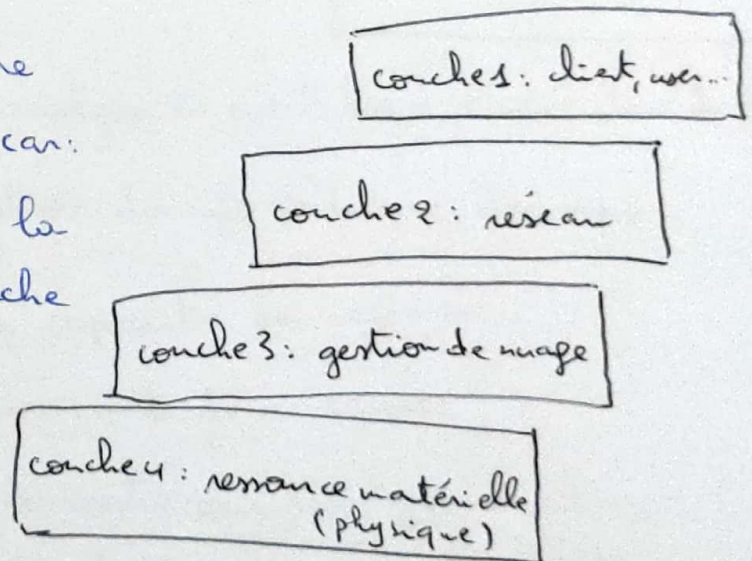
Cloud Sigma : exemple de contrat :

- définir les bases (désigner qui est implicite dans le contrat par ex)
- Décrire les garanties offertes (disponibilité du réseau, service)
- cas où les garanties ne sont plus d'actualité.
- responsabilité des clients.
- quoi faire en cas où les garanties ne sont pas fournies.

### \* SLA et brokers \*

- 1 seul fournisseur  $\Rightarrow$  plusieurs contrats S.L.A.
- Broker :
  - compétence : comprend plusieurs contrats.
  - Rôle : orienter un client vers le contrat le plus approprié.
- Client hétérogène : client léger, client lourd, les 2 peuvent être partie de couches (client, utilisateurs, consommateurs).

Le diagramme d'architecture n'est pas sous forme de pile car : dans une pile on veut montrer la dépendance, l'utilisation (couche supérieure peut utiliser les couche inf).



- architecture générique -

\* Runtime Cloud : intermédiaire entre les services et l'infrastructure (stockage).

→ Définition d'un nuage informatique (cloud) : c'est un service donnant pour des clients hétérogènes un accès à des ressources virtuelles, exécuté sur un ensemble d'ordinateurs (faiblement couplé, pas forcément regroupé).



→ Définition gestion du nuage: Son but est l'efficacité de la gestion des ressources.

\* Contraintes:

→ atteindre une qualité de service acceptable.

→ respecter les clauses des contrats S.L.A des clients.

\* Types:

→ gestion de l'infrastructure

→ " des application

⇒ Gestion de l'infrastructure: son rôle est la gestion des ressources matérielle.

L> objectif: \* Assurer une qualité de service.

\* Respecter les contrats S.L.A (assurant performance)

\* Réduire le coût.

L> extension: consommation d'énergie.

EULA: End user License Agreement  $\approx$  SLA.

⇒ Gestion des app:

L> objectif: \* collecter, analyser les events lié à l'exécution de l'app

\* identification des info critique des app.

L> Méthode: \* Ajuster la capacité des ressource.

\* Approvisionnement de nu service.

⚠ si une machine physique rencontre une très grande charge, l'outil de fournisseur automatiquement va migré certain VM vers d'autre machine physique libre.

IDLE: utilisation idéal d'une ressource (par exemple si on utilise une batterie de façon équilibré on aura une faible chance qu'elle va se defectuer rapidement).

les brokers: (contieurs) ne se catégorise ni du type SaaS, IaaS...

L> ce n'est pas un partner.

L> il ne fait pas partie de l'écosystème C.C.



## \* fourniture d'un Serveur As-a-Service \*

=> Virtualisation du matériel (Hardware)

virtualiser un serveur via un outil :

→ au début : Control program.

→ commun : Hyperviseur.

→ peu commun : Moniteur de VM.

3 façon : \* virtualisation complète du serveur.

\* " " à l'exception de qlq abstraction.

\* " " les fonctionnalités nécessaires.

Δ virtualBox inclue une API et un kit de dev logiciel pour la gestion des VM invité, un outil de ligne de commande pour la gestion locale des invité, un service web pour la gestion à distance des invité,

Virtual machine monitoring : CPU, memory, storage, network, and disk usage.

→ **Instantané** : état d'un système sauvegardé à un moment donné.

en hardware virtualization c'est un fichier incrémentel (état à  $t_0$  + modif faite entre  $t_0$  et  $t_1$ )

↳ un seul fichier contient tous ces sauvegardes (les actions faites dans VM).

→ outil utilisé par le fournisseur pour la gestion :

openstack : gérer un nuage informatique, à la base il était fait pour des cloud privé (mais il peut être utilisé pour public).

↳ openstack peut utiliser généralement tous les hyperviseurs

↳ compliqué à utiliser.

→ **Migration** : vient comme solution pour la surcharge de la machine

physique, blocage de la VM.

↳ changer de logiciel, SE ou machine.

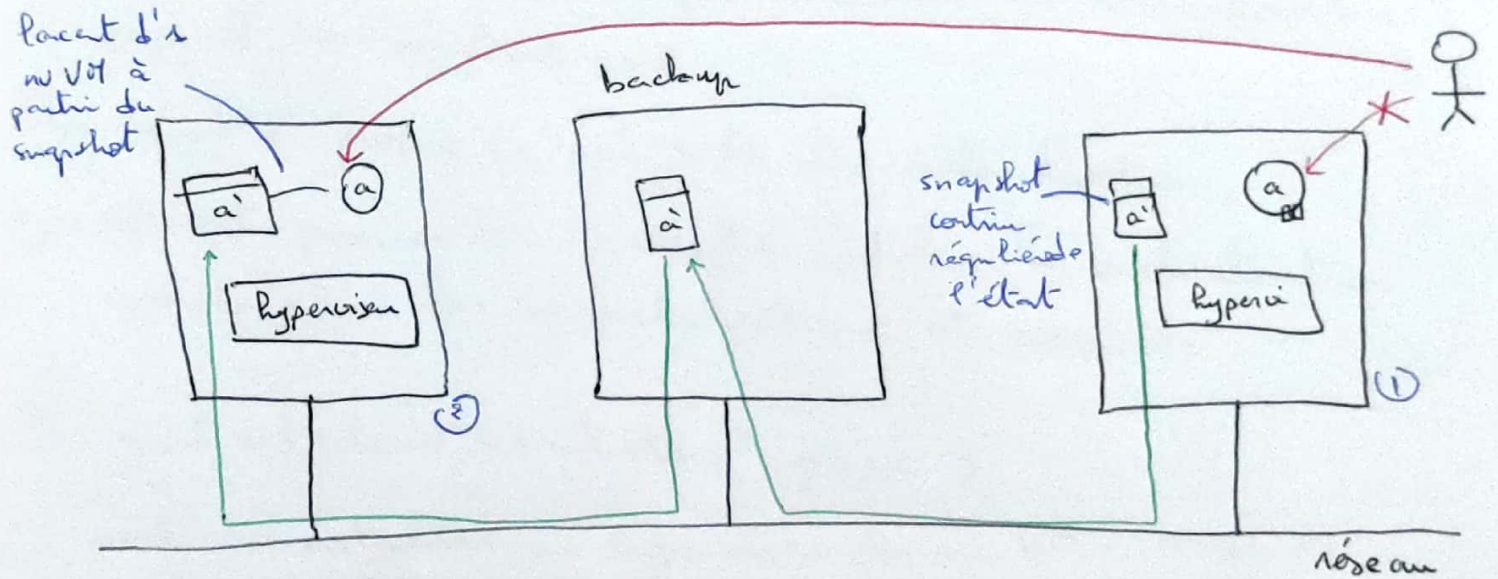


## ⇒ Migration d's VM.

- machine physique utilise un hyperviseur.
- " virtuelle en marche.
- hyperviseur simule VM

→ processus de déplacer machine virtuelle à une autre machine physique.

→ interruption devraient être prise en charge.



\* prise périodique de snapshot nécessairement ne prends pas bcp de temps, et si ça prend bcp de temps on doit réduire la période dont la prise des snapshot est faite.

⚠ si les dernières modif sur la VM sans perdu le image lance la nouvelle VM à partir du dernier snapshot capture.