

Projet d'infrastructure de stockage mutualisée

(pour la communauté scientifique régionale)

Antoine Mahul - CRRRI

antoine.mahul@clermont-universite.fr



1. Contexte et objectifs

2. Panorama des technologies de stockage

Modèles de croissance : *scale up / scale out*

Modes d'accès : bloc, fichiers, objets

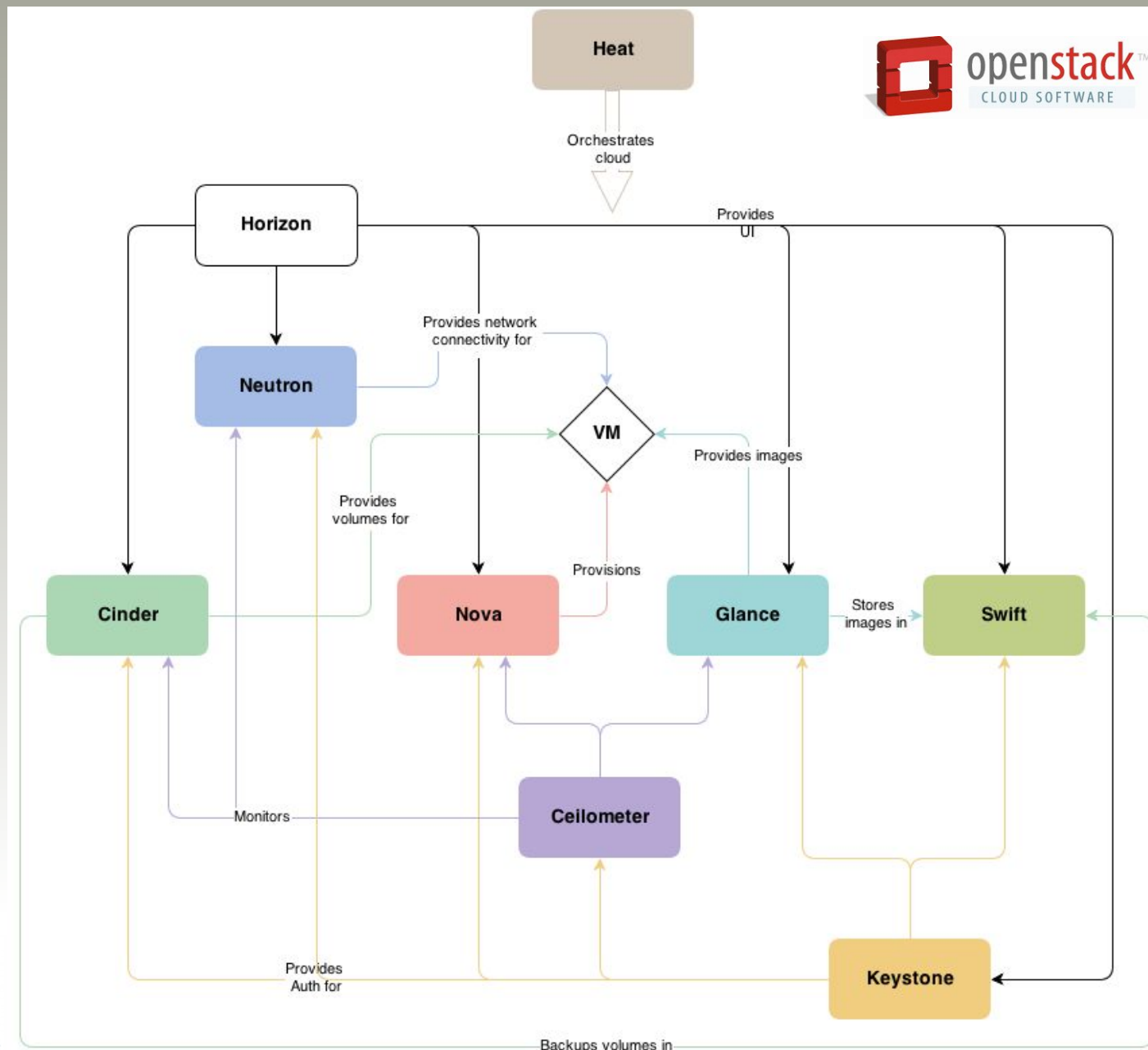
3. Vers un cloud de données

Cloud IaaS

API S3

4. Focus sur Ceph

Exemple de cloud IaaS : Openstack



Centre Régional de Ressources Informatiques

- Service de la CUE Clermont Université
- Pôle AMSI :
 - maîtrise d'œuvre et exploitation des applications nationales (Apogée, Harpège, SIFAC...) pour les universités
- Pôle Infrastructure
 - Opérateur réseau : réseaux campus et métropolitain
 - Opérateur datacentre : surface technique de 230 m²
 - Gestion de ressources informatiques :
 - Ressources mutualisées pour les DSI
 - Ressources mutualisées pour la recherche : calcul, stockage

Ressources mutualisées pour la communauté scientifique

- Ressources de la grille de calcul EGI
 - 240 coeurs, 480 Go de RAM
 - 48 To de stockage
- Ressources de calcul locales
 - 135 coeurs, 705 Go de RAM
 - 10 To de stockage partagé (NAS ZFS)
 - A court terme : 256 coeurs supplémentaires, 1024 Go RAM
- Services tiers
 - Hébergement de projets scientifiques
 - Plateforme galaxy
 - Forge pour le développement collaboratif

Projet de stockage mutualisé

- CPER 2014 :
“acquisition d’outils mutualisés pour le stockage, l’analyse de grandes masses de données, la modélisation et le calcul intensif”
- Projet de stockage capacitif
 - Stockage de grands volumes de données (>100 To)
 - Allocation d'espace de stockage à la demande
 - Extensibilité à moindre coût
 - Maîtrise des frais de fonctionnements
- Première brique d'un cloud IaaS

Modèle *Scale-up*

Croissance verticale : *Scale-up*

- 😊 *facilité d'exploitation*
- 😊 *performance maîtrisée*
- 😐 *dimensionnement pas toujours adapté*
- 😡 *passage à l'échelle (perf, taille)*
- 😡 *renouvellement*



Exemple : serveurs NAS, baies de stockage

Modèle *Scale-Out*

Croissance horizontale : *Scale-out*

- 😊 *passage à l'échelle linéaire (perf, taille)*
- 😊 *dimensionnement adapté à l'usage*
- 😊 *renouvellement progressif*
- 😐 *exploitation non triviale*
- 😞 *infra complexe*



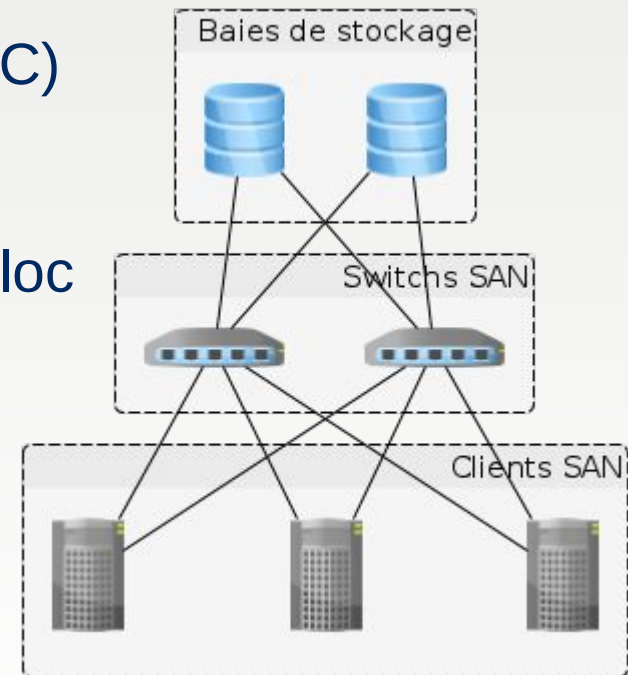
Exemples : ISILON, PANASAS, GlusterFS, Ceph...

Mode bloc

- Mode de stockage bas niveau
- Bloc :
 - Plus petite unité de stockage sur un disque physique
 - Une taille (fixe) + une position sur le disque
- Devices : disque dur, SSD, bande magnétique, DVD...
- Système de fichiers nécessaire pour associer les blocs aux fichiers
- RAID : apporte de la sécurisation au stockage en mode bloc
 - RAID matériel: carte RAID, baies de stockage
 - RAID logiciel: linux, ZFS
 - Différents niveaux de sécurisation : RAID 1 (2N), RAID 5 (N+1), RAID 6 (N+2)

Mode bloc : DAS, SAN

- DAS : Direct Attached Storage
 - Baie de disque externe
 - Attachement direct au serveur (SAS, FC)
- SAN : Storage Area Network
 - réseau de stockage (dédié) en mode bloc
 - Protocoles : FC, iSCSI (sur IP), FcoE
 - Sécurisation importante
 - Espace de stockage non partagé entre les clients



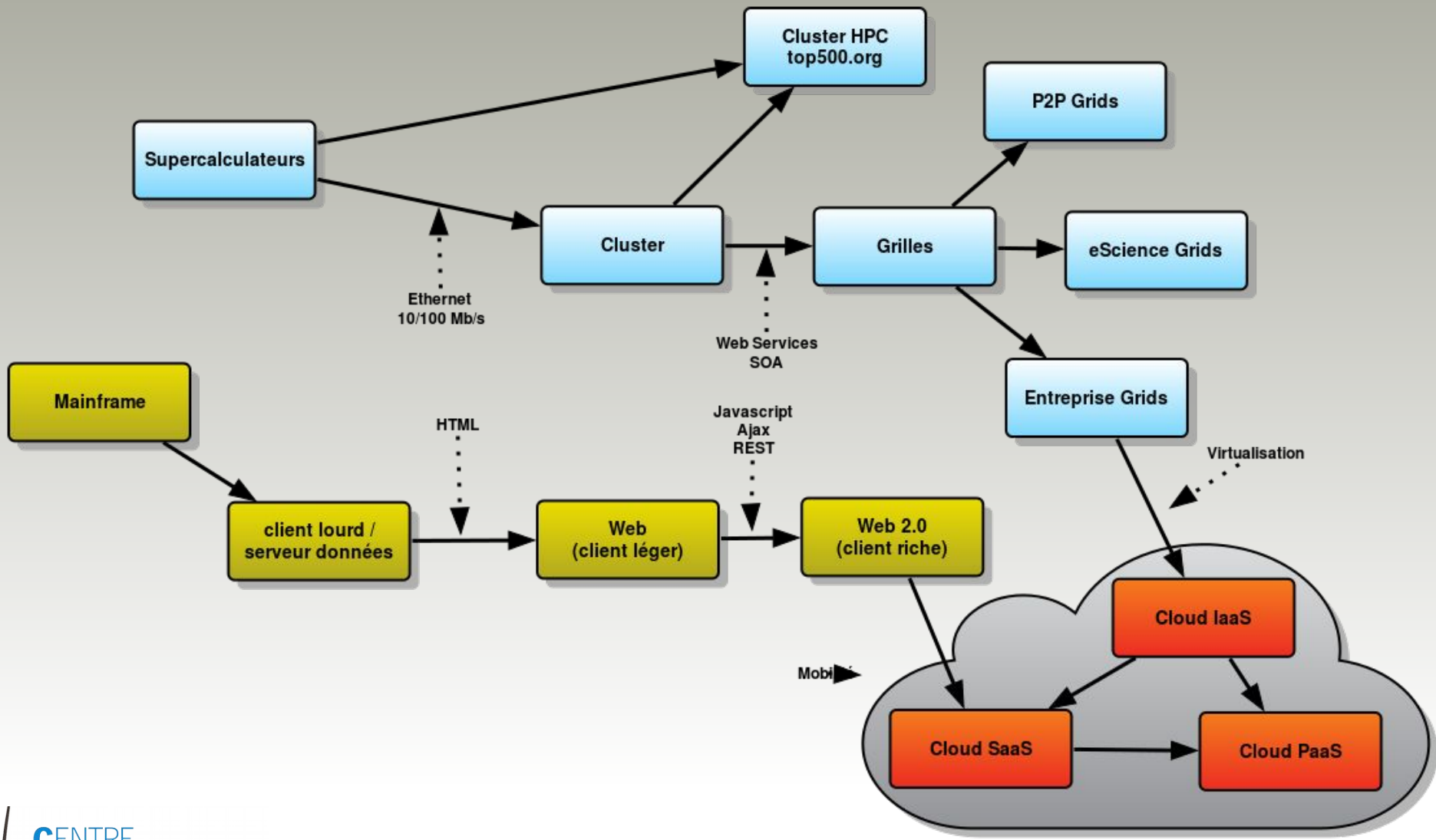
Mode fichiers

- Mode d'accès adapté à l'utilisateur final
 - Abstraction du stockage
 - Organisation hiérarchique : fichiers, répertoires,
 - Schéma de nommage “user friendly”...
- NAS : Network Attached Storage
 - Mode fichier sur le réseau
 - Pas de formatage de l'espace de stockage sur le client
 - Espace de stockage partagé entre plusieurs clients
 - Metadonnées centralisés sur le serveur
 - Protocoles (IP) : NFS, SMB/CIFS, AFP...

Mode objet

- Mode de stockage distribué :
 - Objet = données + métadonnées + identifiant unique
 - organisation plate de type clé/valeur
 - abstraction du stockage (bloc ou FS)
 - architecture de type cluster (*scale-out*)
- Stockage grands volumes de données non structurés
- Modes d'accès : API et/ou bloc, fichiers
- Exemples : Lustre, Amazon S3, GlusterFS, HDFS, EMC ViPR...

Vers un cloud IaaS



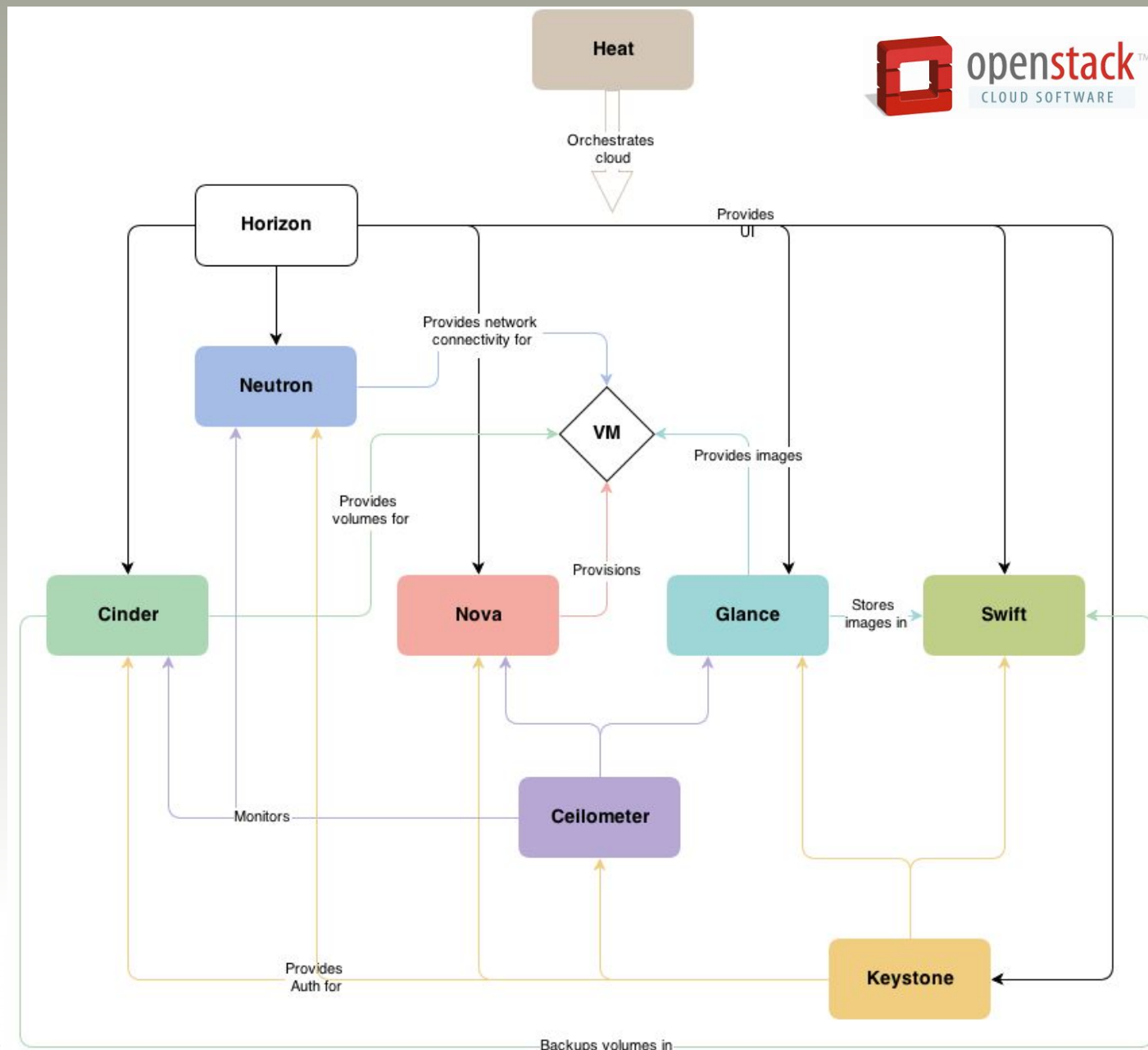
Définition du “cloud”

- **On-demand self-service** : un client peut s'allouer lui-même des ressources (temps cpu, stockage, bande passante)
- **Broad network access** : les ressources sont accessibles depuis un réseau via des mécanismes utilisables depuis des différents clients (pc, portable, smartphone, tablette...)
- **Resource pooling** : les ressources sont mutualisées, allouées dynamiquement selon la demande, sans localisation précise
- **Rapid elasticity** : les ressources peuvent s'adapter de façon élastique aux pics de charge
- **Measured service** : le système s'appuie sur des critères de mesures des différents services pour le contrôle, l'optimisation et la facturation

Différents modèles de cloud

- Infrastructure as a Service (IaaS)
 - Infrastructure informatique hébergée
 - Infrastructure physique transparente
 - Exemple : Amazon EC2, Amazon S3, Dropbox, Openstack
- Platform as a Service (PaaS)
 - Environnement & briques de développement prêt à l'emploi
 - Infrastructure transparente
 - Exemple : Google App Engine, Microsoft Azure
- Software as a Service (SaaS)
 - Application hébergée, accessible via un client léger
 - Plateforme et infrastructure transparente
 - Exemple : Google docs, Microsoft Office 365, BaseCamp...

Exemple de cloud IaaS : Openstack



Cloud de Données

- **AWS S3 :**
 - service de stockage cloud d'Amazon
 - API REST
 - de facto, API “standard”
- Exploitable via n'importe quel langage (PHP, Perl, Python, Java...)
- De plus en plus de clients S3 : CrossFTP, CyberDuck, S3CMD, Owncloud...

Exemple (en python)

```
import boto
import boto.s3.connection
access_key = 'put your access key here!'
secret_key = 'put your secret key here!'

# Creation de la connexion
conn = boto.connect_s3(
    aws_access_key_id = access_key,
    aws_secret_access_key = secret_key,
    host = 'objects.dreamhost.com',
    calling_format = boto.s3.connection.OrdinaryCallingFormat(),
)

# Creation d'un bucket
bucket = conn.create_bucket('my-new-bucket')

# Creation d'un objet
key = bucket.new_key('hello.txt')
key.set_contents_from_string('Hello World!')

# Download d'un objet
key = bucket.get_key('perl_poetry.pdf')
key.get_contents_to_filename('/home/larry/documents/perl_poetry.pdf')
```

Ceph

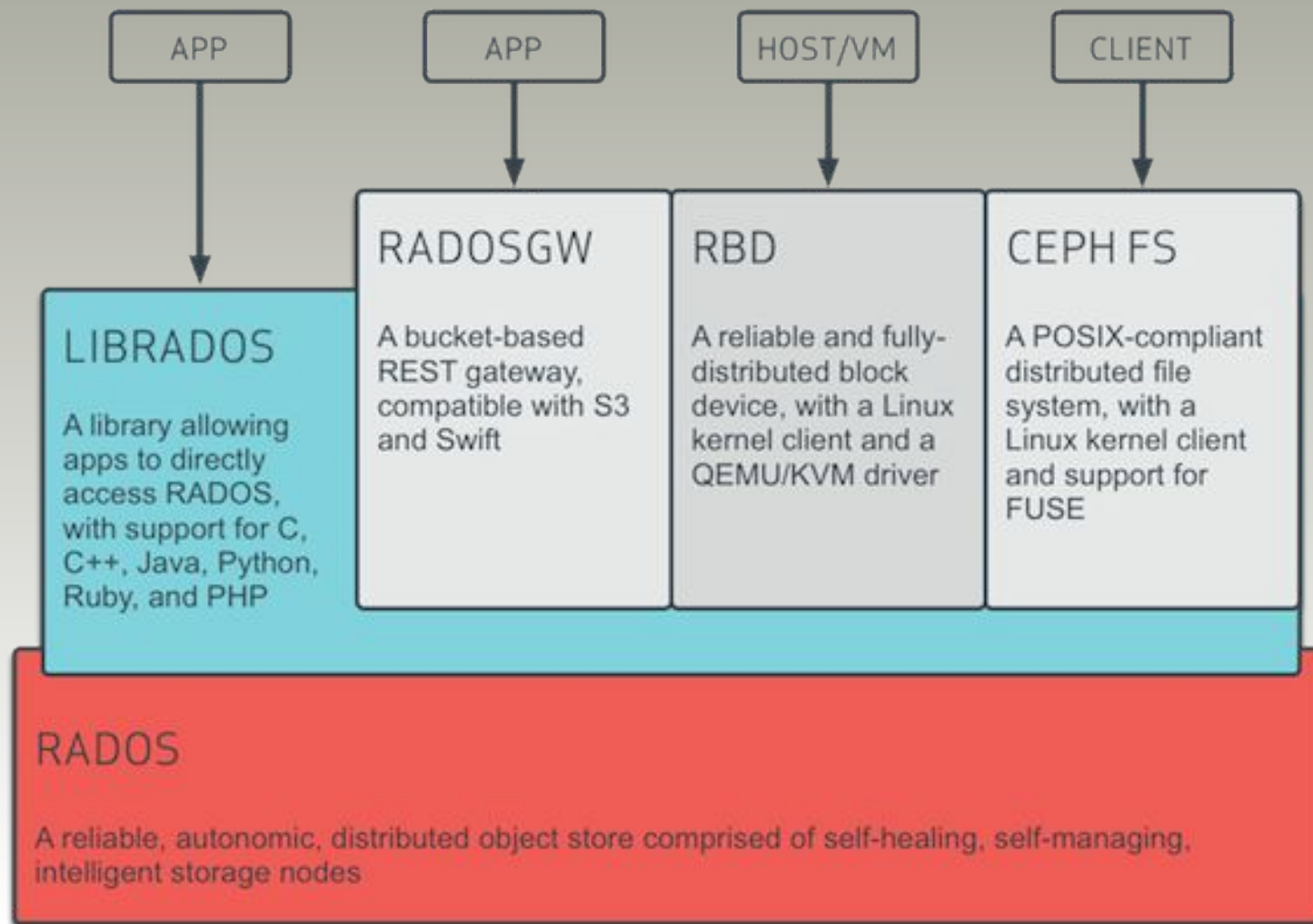


- Plateforme de stockage distribué
 - Stockage en mode objet
 - Architecture *Scale-out* adaptée au passage à l'échelle
 - S'appuie sur du matériel ordinaire (*commodity hardware*)
 - Sans SPOF
- Logiciel Libre
 - Technologie issue de l'université de Californie (Santa Cruz)
 - Thèse de Sage A. Weil (2007)
 - Développement et support commercial assuré par Inktank
 - Inktank racheté par Redhat le 30 avril 2014 !

Ceph : fonctionnalités

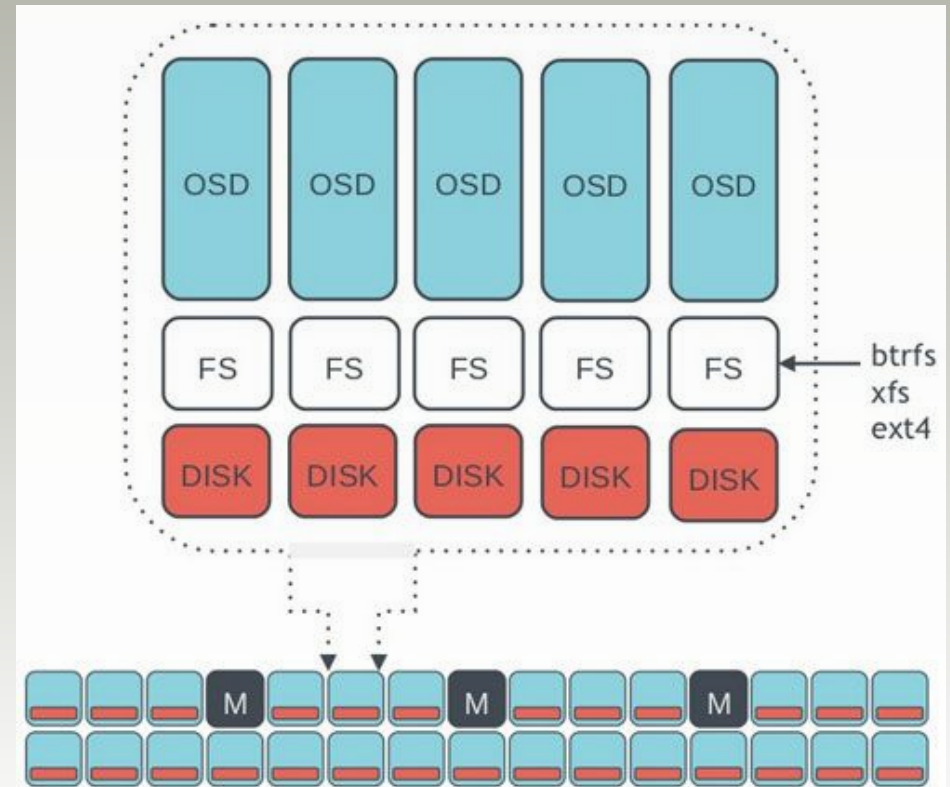
- Cluster de stockage RADOS
 - Réplication des données chaudes (surcout : x2, x3...)
 - Correction d'erreur pour les données froides (x1.5)
 - Equilibrage automatique
 - Algo de placement (CRUSH) assez évolué et modulable
- Client Ceph : interface vers les utilisateurs
 - RADOSGW : API compatible Swift et S3
 - RBD : mode bloc (intégré au kernel Linux)
 - CephFS : mode fichier (load balancing encore experimental)

Principes



Ceph : noeuds de stockage

- Pas de master
- Un cluster de moniteurs qui conserve la topologie
- Un cluster d'OSD
 - Un système de fichier par disque
 - Un démon OSD par disque
 - Pas de RAID !



Conclusion

- Objectifs :
 - Fournir une infra de stockage qui puisse passer à l'échelle pour les projets scientifique à venir (Big Data...)
 - Première brique d'un cloud plus large (compatibilité Openstack)
- Avancement
 - Veille techno & tests en cours au CRRI (Ceph, Openstack Swift)
 - Automne : financement d'une infra via CPER 2014
- Pilotage
 - Formalisation d'un mésocentre avec les établissements de tutelle
 - comité scientifique + comité d'utilisateur