



Pr. Noura AHERRAHROU

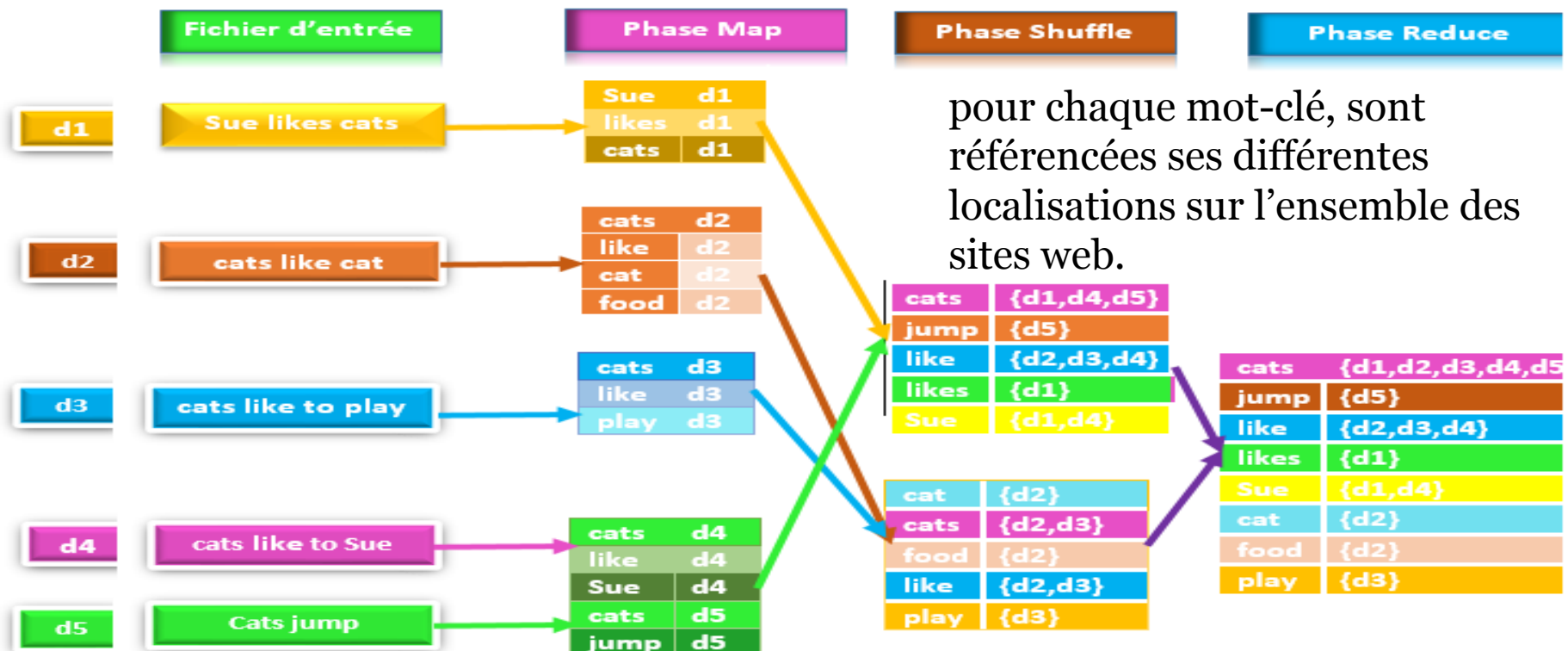
L'approche conceptuelle de Google

Pour répondre au défi des Big Data, l'idée de Google était de développer une approche conceptuelle consistant, d'une part, à **distribuer le stockage** des données et, d'autre part, à **paralléliser le traitement** de ces données sur plusieurs noeuds d'une grappe de calcul (un cluster d'ordinateurs).



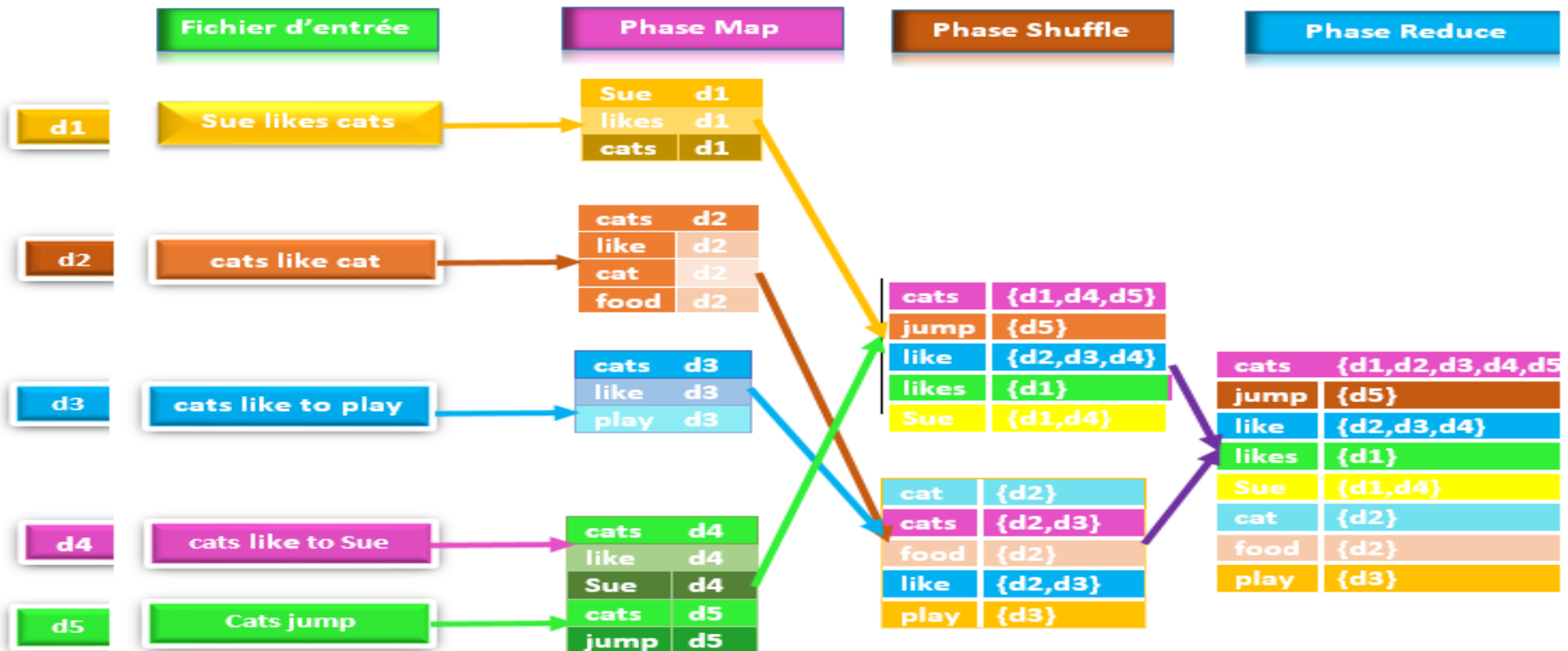
Au niveau de traitement

Google découpe le problème d'indexation des pages web en tâches ou sous-problèmes à distribuer dans le cluster pour exécution. Pour ce faire, il construit un index pour chaque mot-clé contenu sur chaque page web.



Au niveau de traitement

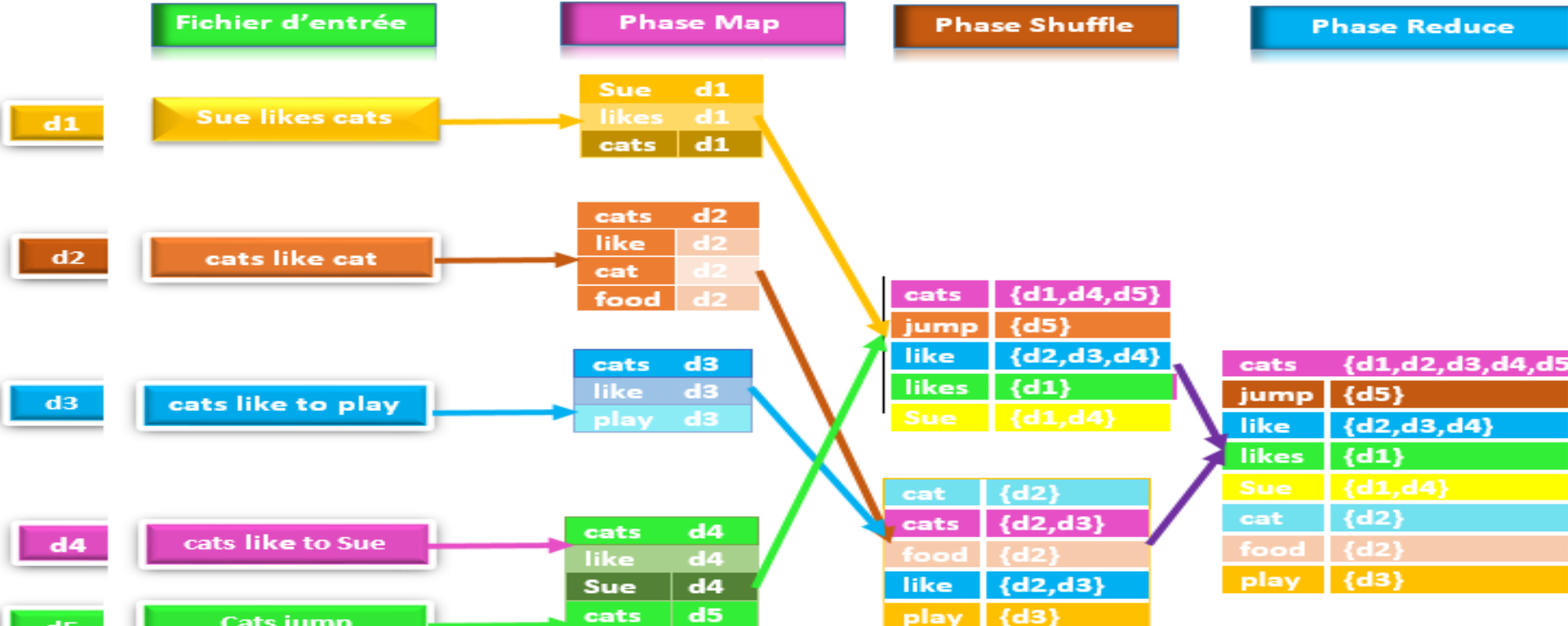
Ainsi, chaque fois que vous recherchez un mot, l'index fournit les pages où il se situe ; cela vous évite de fouiller tout le Web page par page pour retrouver le mot. Aujourd'hui, la recherche Internet fonctionne selon ce principe : le moteur de recherche constitue une base d'index pour chaque mot.



La phase Map

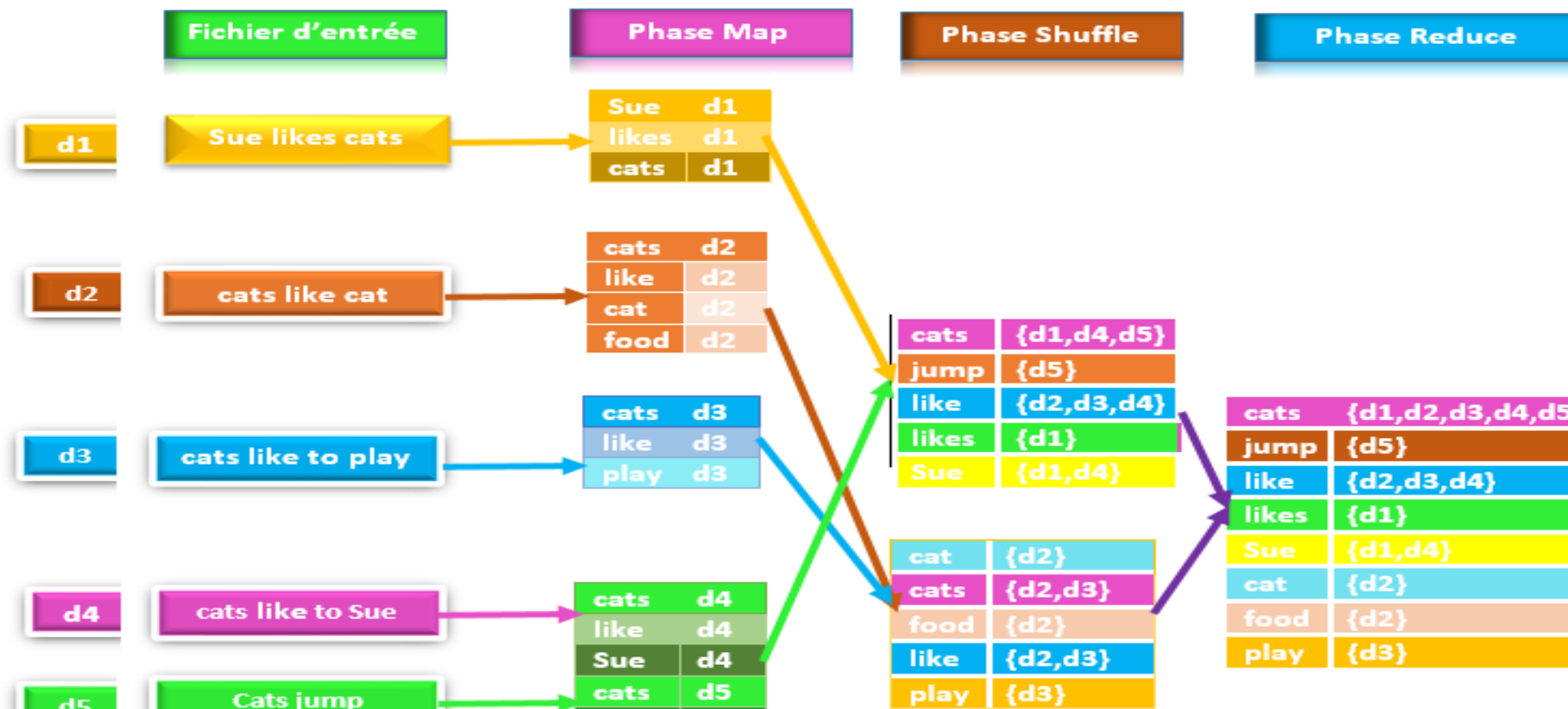


Consiste à assigner à chaque noeud du cluster la tâche d'attribuer à chaque mot de la page web un indice correspondant à la page dans laquelle il est. Cet indice peut être le titre de la page, son numéro ou n'importe quel élément qui permet de l'identifier de façon unique parmi toutes les pages qui constituent le site web.



La phase Shuffle: Tri et regroupement

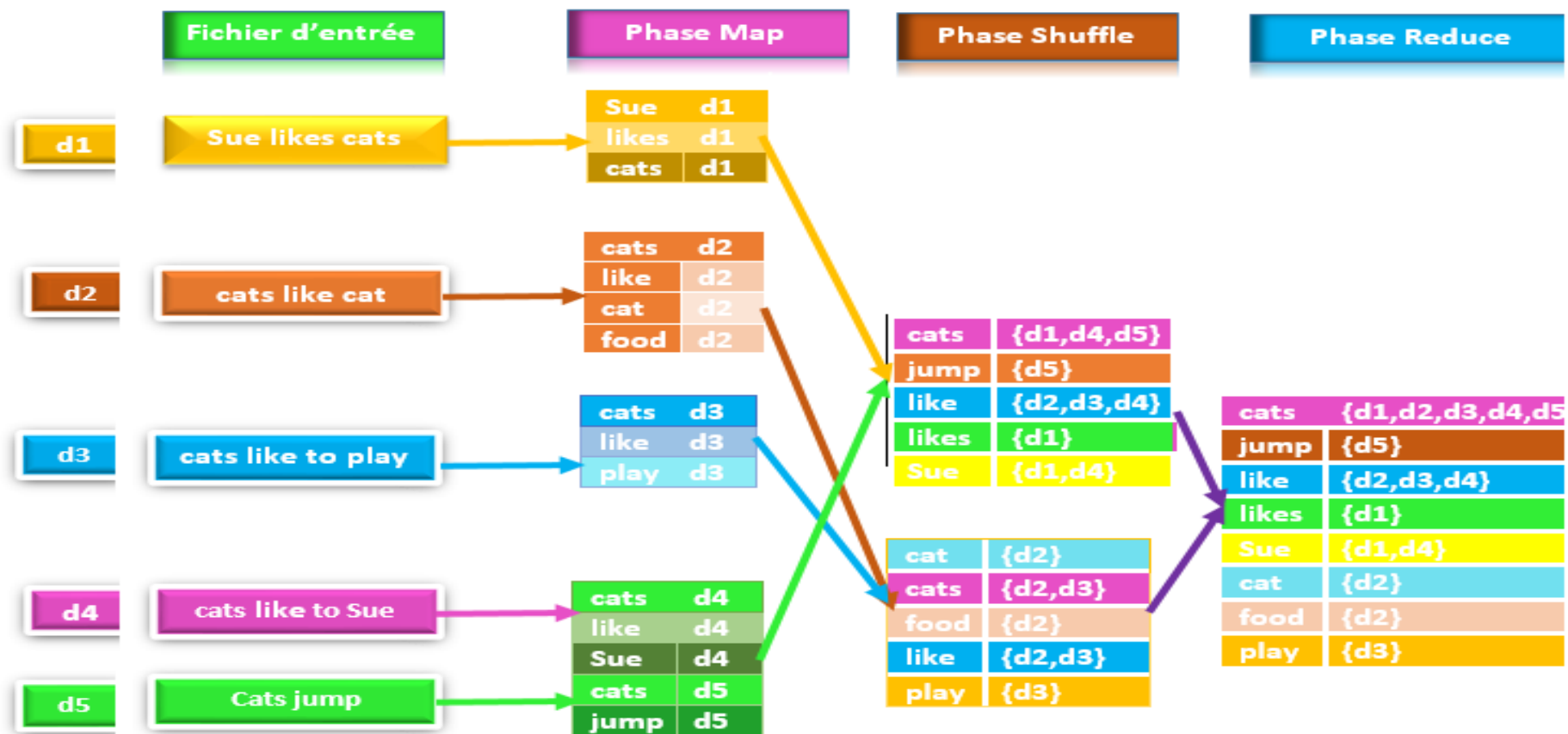
Consiste, pour chaque noeud, à trier par ordre alphabétique les mots auxquels il a affecté un index. Cette étape intermédiaire vise à faciliter le travail effectué par la troisième phase.



La phase Reduce

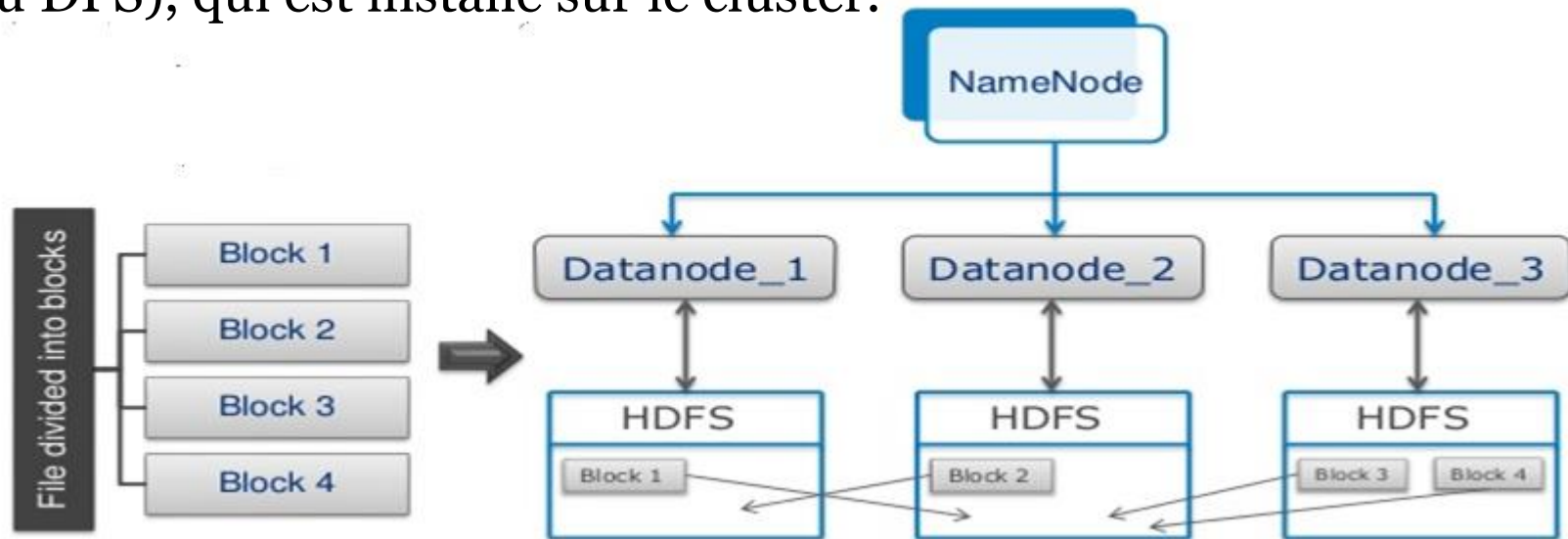


Consiste, pour chaque mot contenu dans l'ensemble des noeuds du cluster, à regrouper tous ces indices. Ainsi, on obtient l'index final.



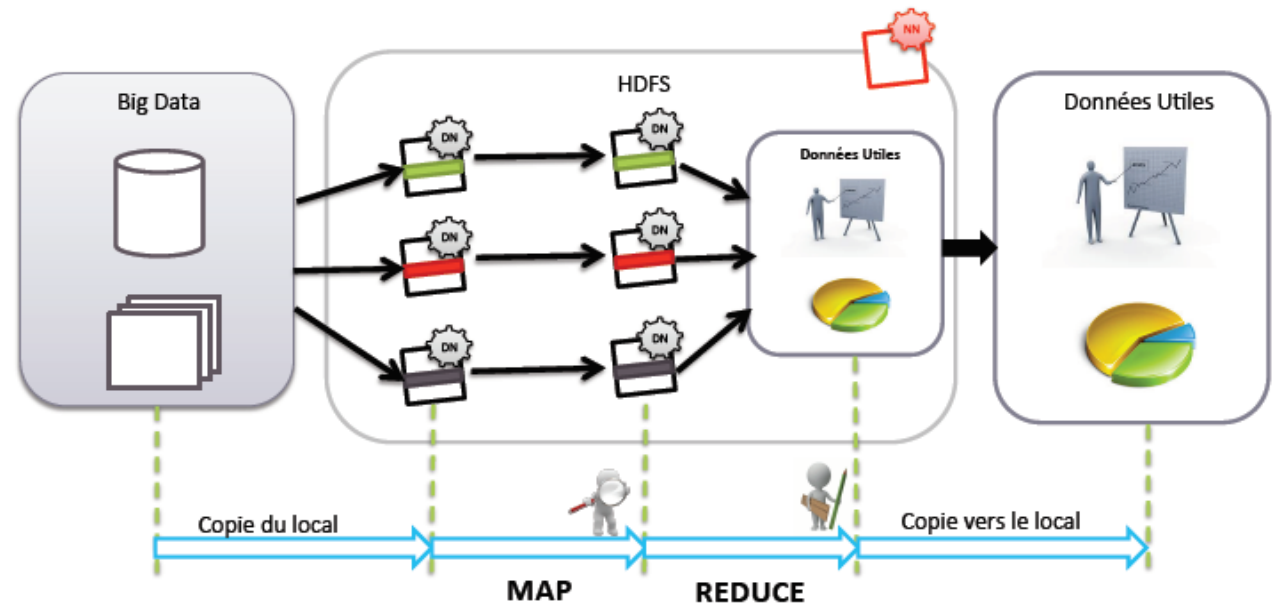
Au niveau de stockage

Comme la tâche de construction des index est partagée entre les noeuds du cluster, les fichiers contenant les mots (les pages web) doivent être divisés, et chaque morceau de fichier doit être stocké de façon redondante sur tous les disques durs du cluster. Ainsi, si un noeud tombe en panne au cours du traitement, cela n'affecte pas les autres tâches. Techniquement, le stockage de fichiers se fait à l'aide de ce que l'on appelle le système de fichiers distribués (Distributed File System ou DFS), qui est installé sur le cluster.



A l'origine d'Hadoop

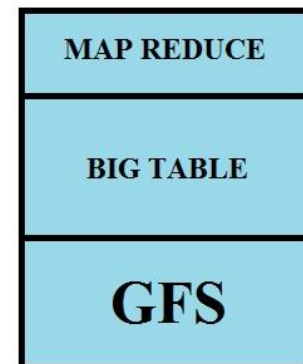
Le découpage des traitements en plusieurs tâches et l'exécution en parallèle de ces dernières sur un grand nombre de noeuds se font à l'aide du modèle en trois étapes présenté précédemment, nommé **MapReduce**. La distribution, le stockage et la redondance des fichiers sur le cluster sont pris en charge par le système de fichiers distribués mis au point par Google et qu'il a appelé **Google File System (GFS)**.



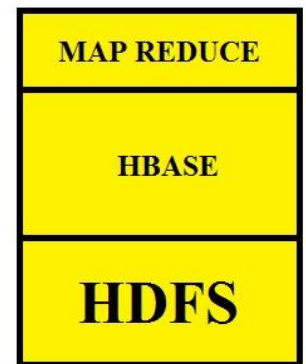
A l'origine d'Hadoop

À l'origine, ces deux éléments étaient utilisés en interne chez Google. Plus tard, un ingénieur de l'entreprise, Doug Cutting, implémentera en Java le MapReduce et le GFS et leur donnera le nom d'une des peluches de son fils : **Hadoop**. Le **HDFS (Hadoop Distributed File System)** constitue son système de fichiers distribués, l'équivalent du GFS.

Depuis 2009, le projet Hadoop a été repris par la fondation Apache.



GOOGLE

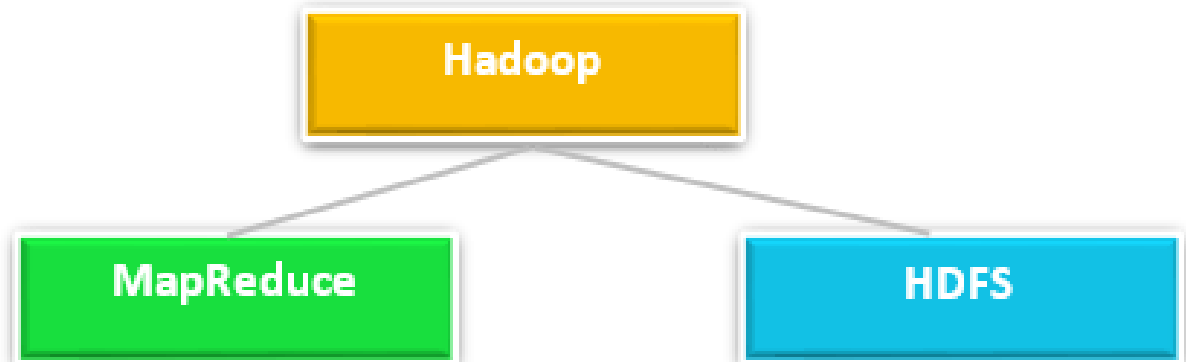


HADOOP

Aux origines d'Hadoop

le MapReduce est une approche conceptuelle qui a besoin d'être implémenté pour être utilisé. Hadoop répond à cette demande : c'est l'implémentation la plus populaire et la plus mature actuellement sur le marché.

En réalité, Hadoop est un ensemble de classes écrites en Java pour la programmation des tâches **MapReduce** et **HDFS**, et dont les implémentations sont disponibles dans d'autres langages de programmation : en Scala et C# notamment.



Terminologie d'Hadoop

Un **job MapReduce** est une unité de travail que le client veut exécuter. Il consiste en trois choses :

- le fichier des données à traiter (input file) ;
- le programme MapReduce ;
- les informations de configuration (métadonnées).

Le cluster exécute le job en divisant le programme MapReduce en deux : les tâches Map d'un côté et les tâches Reduce de l'autre. Dans le cluster Hadoop, il y a deux types de processus : le **JobTracker** et un ensemble de **TaskTrackers**.

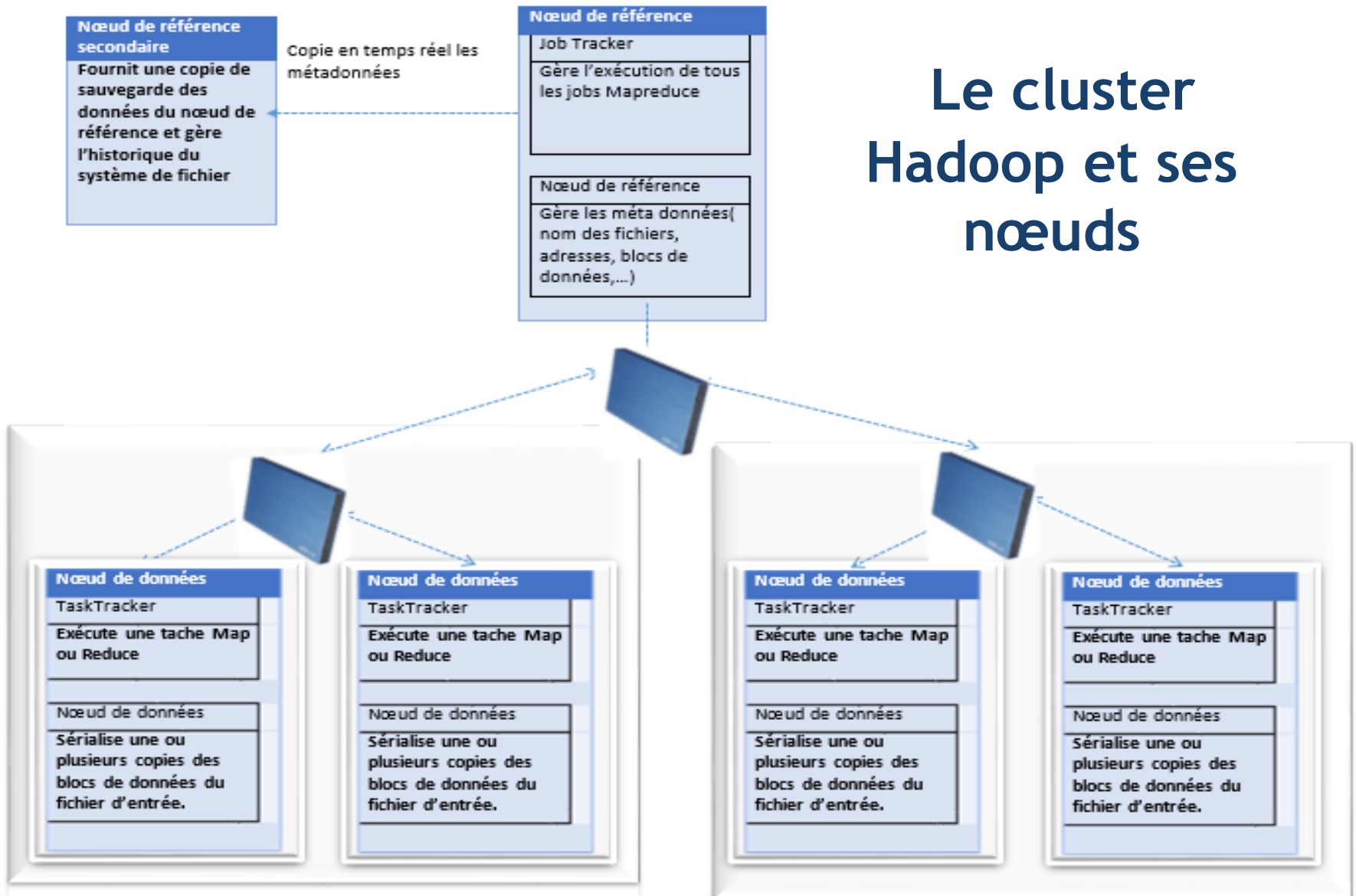
Terminologie d'Hadoop

- ❑ Le **JobTracker (Master)** est le processus central qui est démarré sur le noeud de référence (name node) ; il coordonne tous les jobs qui s'exécutent sur le cluster, gère les ressources de ce dernier et planifie les tâches à exécuter sur les **TaskTrackers**.
- ❑ **TaskTrackers (Workers)** sont les processus qui traitent le programme MapReduce ; exécutent les tâches Map ou Reduce et envoient des rapports d'avancement au JobTracker, qui garde une copie du progrès général de chaque job. Si une tâche échoue, le **JobTracker** peut la replanifier sur un **TaskTracker** différent.

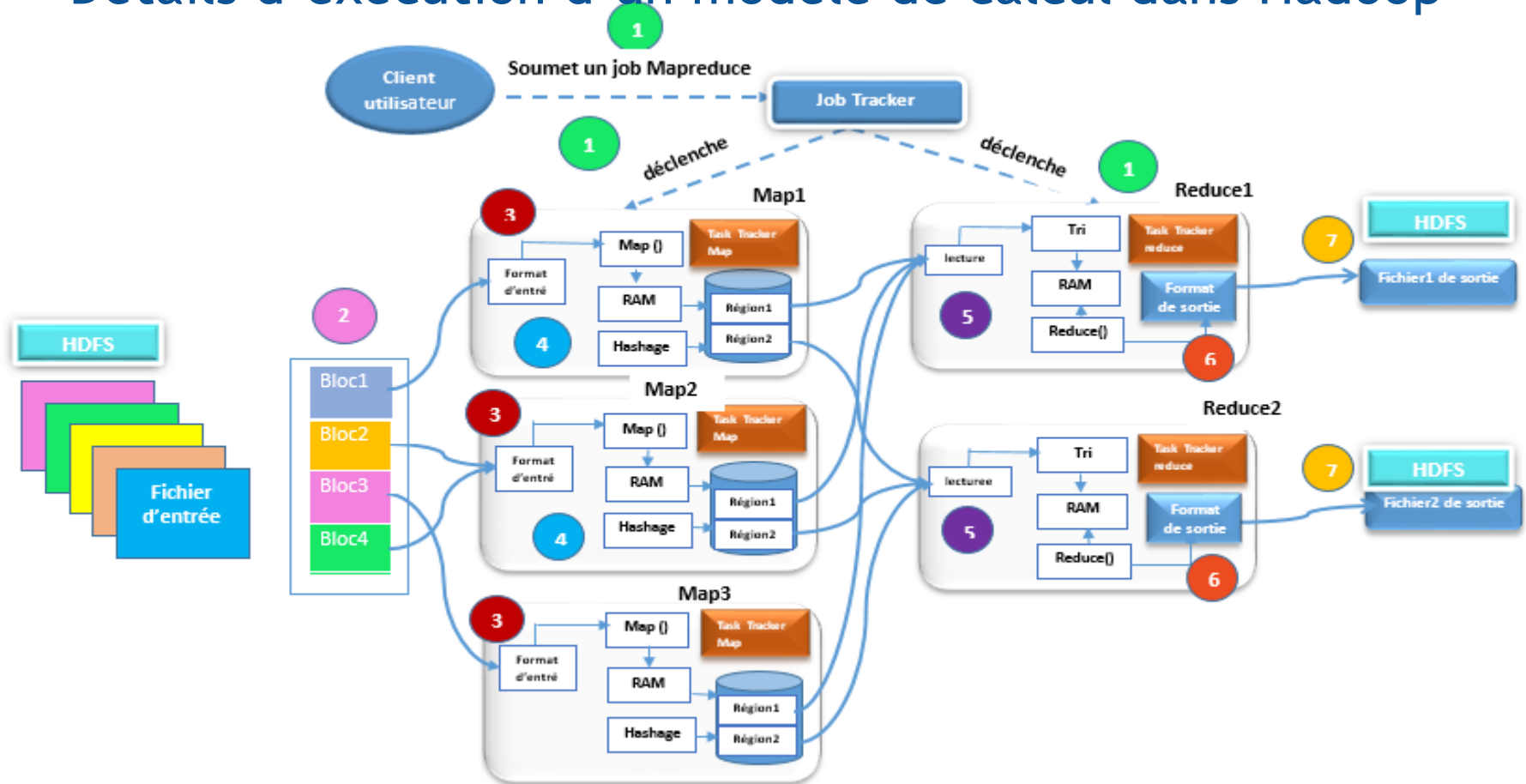
Terminologie d'Hadoop

- ❑ les **noeuds de données** sont des **TaskTrackers**, tandis que le **noeud de référence** est un **JobTracker**.
- ❑ Chaque **tracker** possède une **partie stockage**, assurée par le noeud, et **une partie traitement**, effectuée par le processus (Worker ou Master) démarré sur le noeud. Dans la terminologie Hadoop, le nom du processus est associé à la machine ; on parle de **JobTracker** et de **TaskTracker**.
- ❑ La figure ci-après illustre les relations entre le **JobTracker** et les **TaskTrackers** dans le cluster.

Le cluster Hadoop et ses nœuds

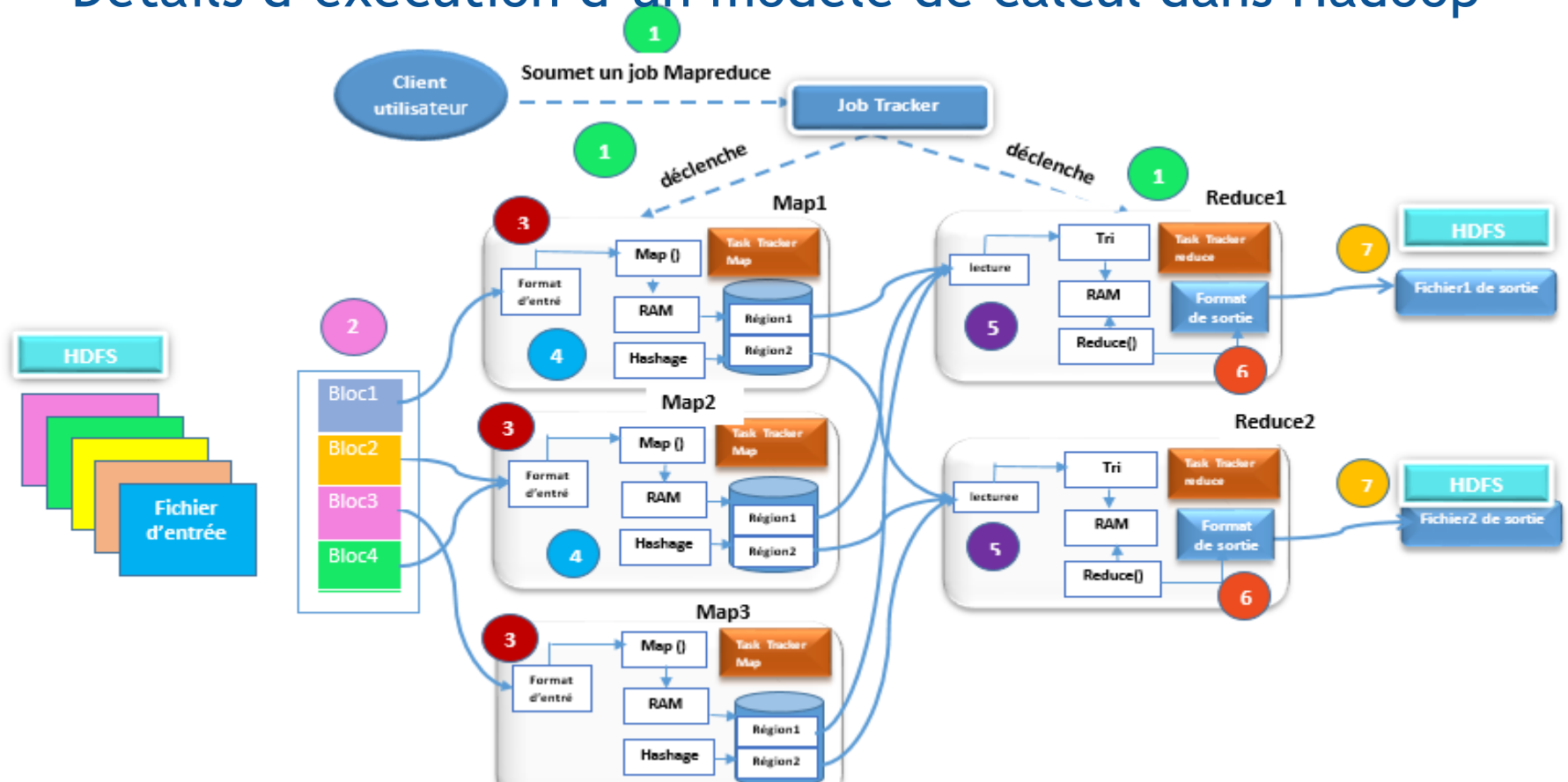


Détails d'exécution d'un modèle de calcul dans Hadoop



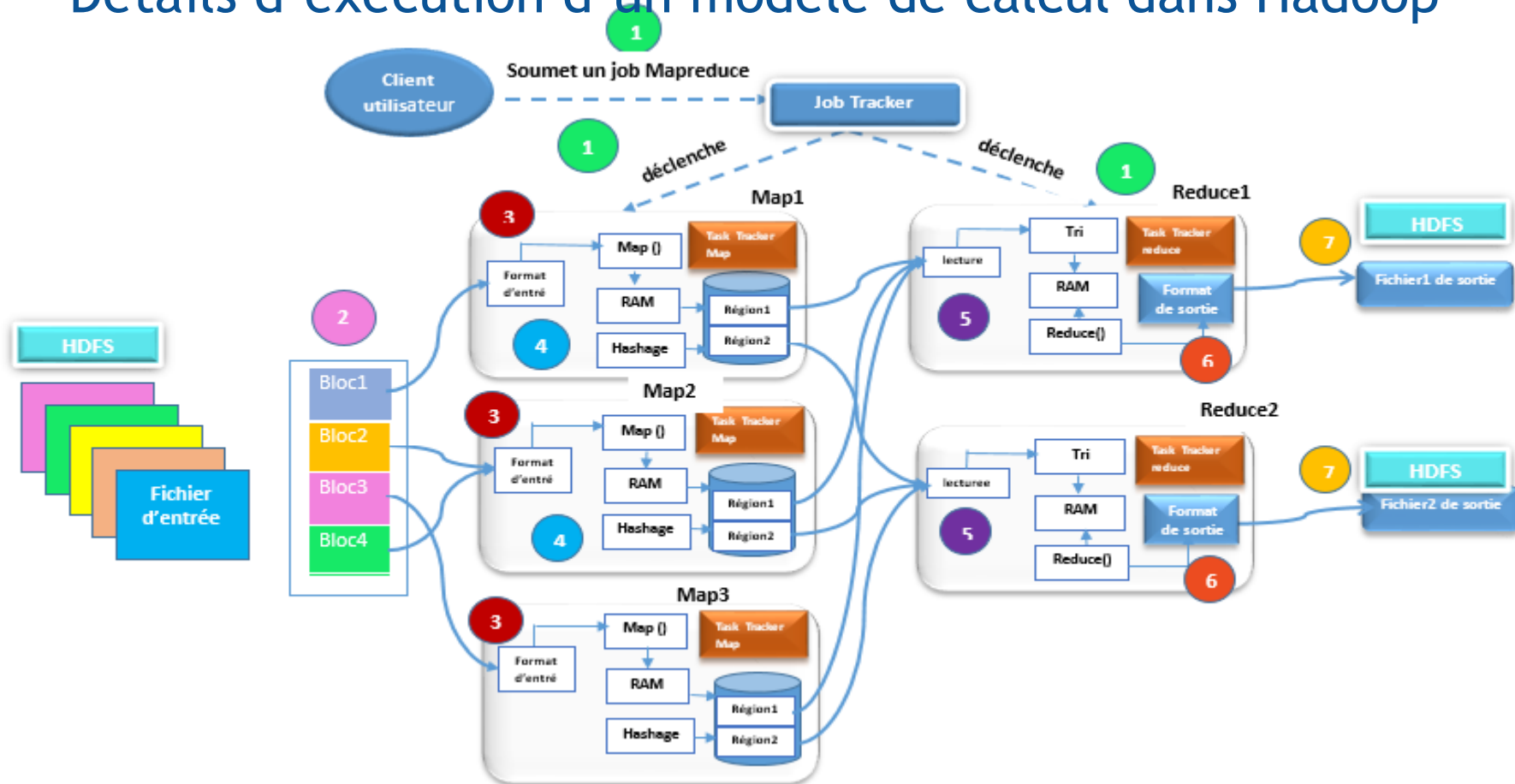
1. Au départ, l'utilisateur configure le job : il écrit la fonction Map, la fonction Reduce, spécifie le nombre r de tâches Reduce, le format de lecture du fichier d'entrée, le format des r fichiers de sortie, éventuellement la taille des blocs du fichier d'entrée et le facteur de réplication. Une fois que tout cela est fait, le JobTracker démarre les r TaskTrackers

Détails d'exécution d'un modèle de calcul dans Hadoop



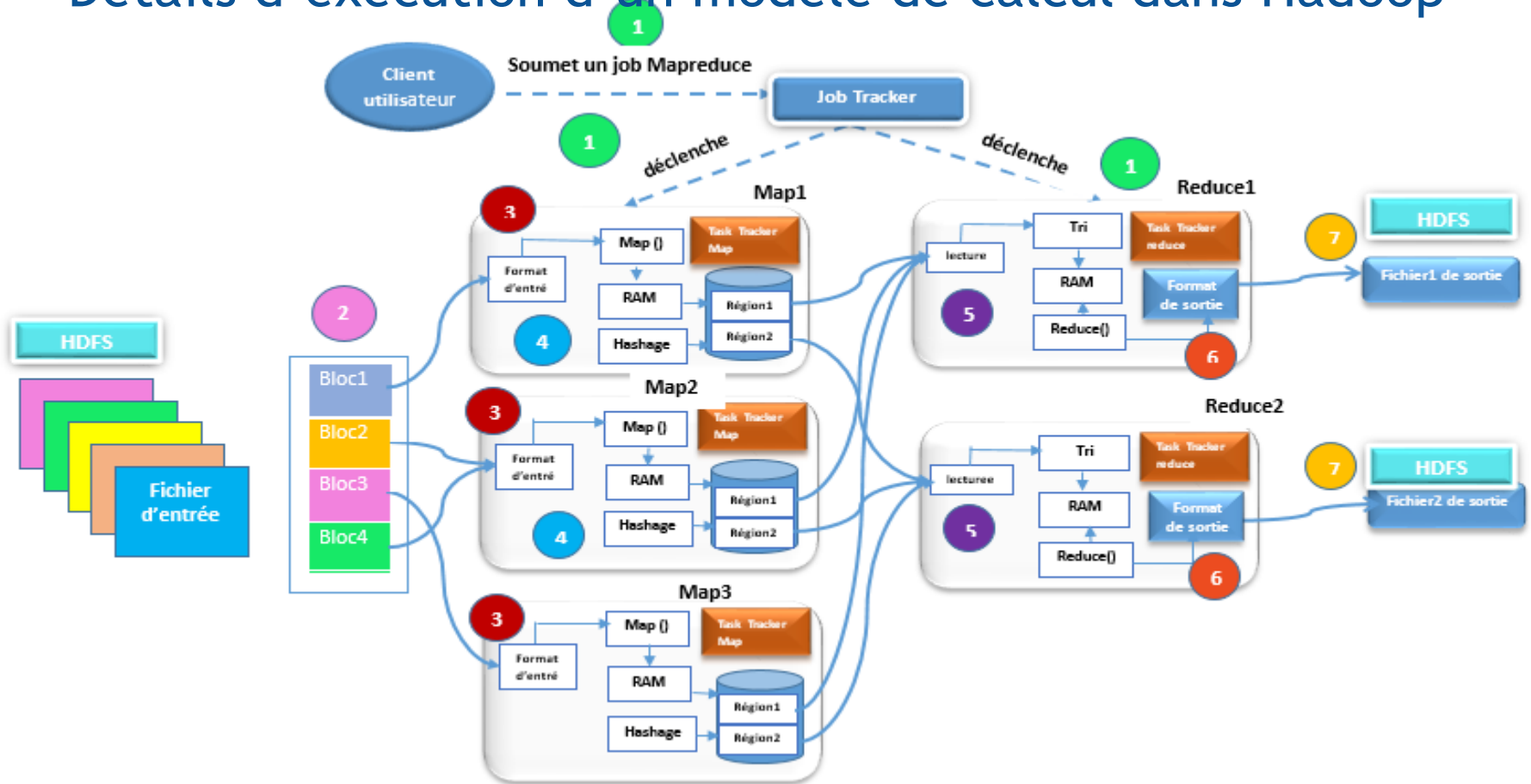
2. Le HDFS découpe le fichier d'entrée en M blocs de taille fixe, généralement 64 Mo. Ensuite, le HDFS réplique ces blocs selon le facteur défini par l'utilisateur (3, par défaut) et les distribue de façon redondante dans des noeuds différents du cluster. Diviser le fichier d'entrée en blocs de taille fixe permet de répartir de façon équilibrée la charge de traitement parallèle entre les noeuds du cluster.

Détails d'exécution d'un modèle de calcul dans Hadoop



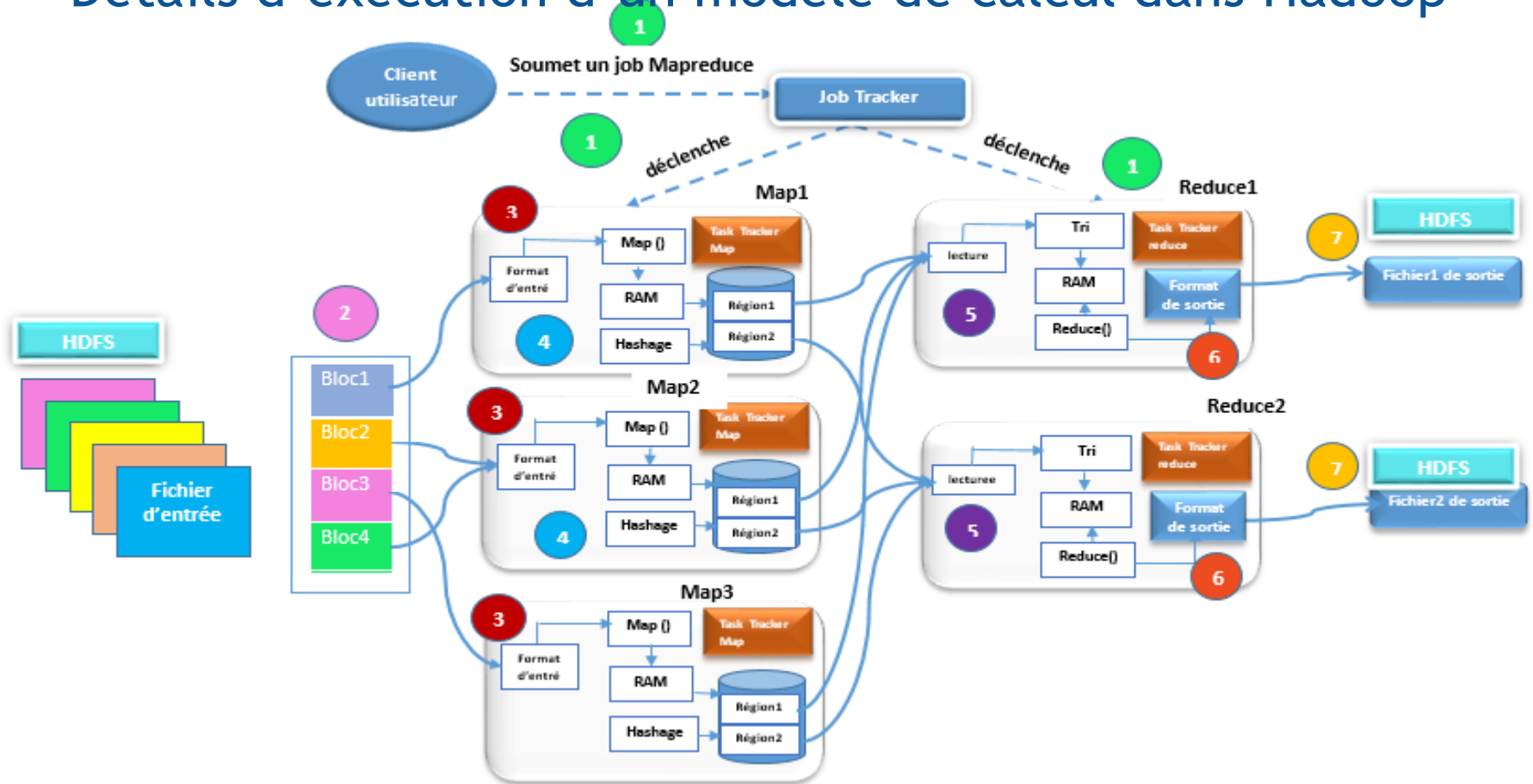
3. Par défaut, le JobTracker déclenche M TaskTrackers sur les M noeuds de données dans lesquels ont été répartis les M blocs du fichier d'entrée. Chaque TaskTracker lit le contenu du bloc de fichiers par rapport au format d'entrée spécifié par l'utilisateur, puis le transforme en paires clé/valeur par le processus de hachage défini dans la fonction Map.

Détails d'exécution d'un modèle de calcul dans Hadoop



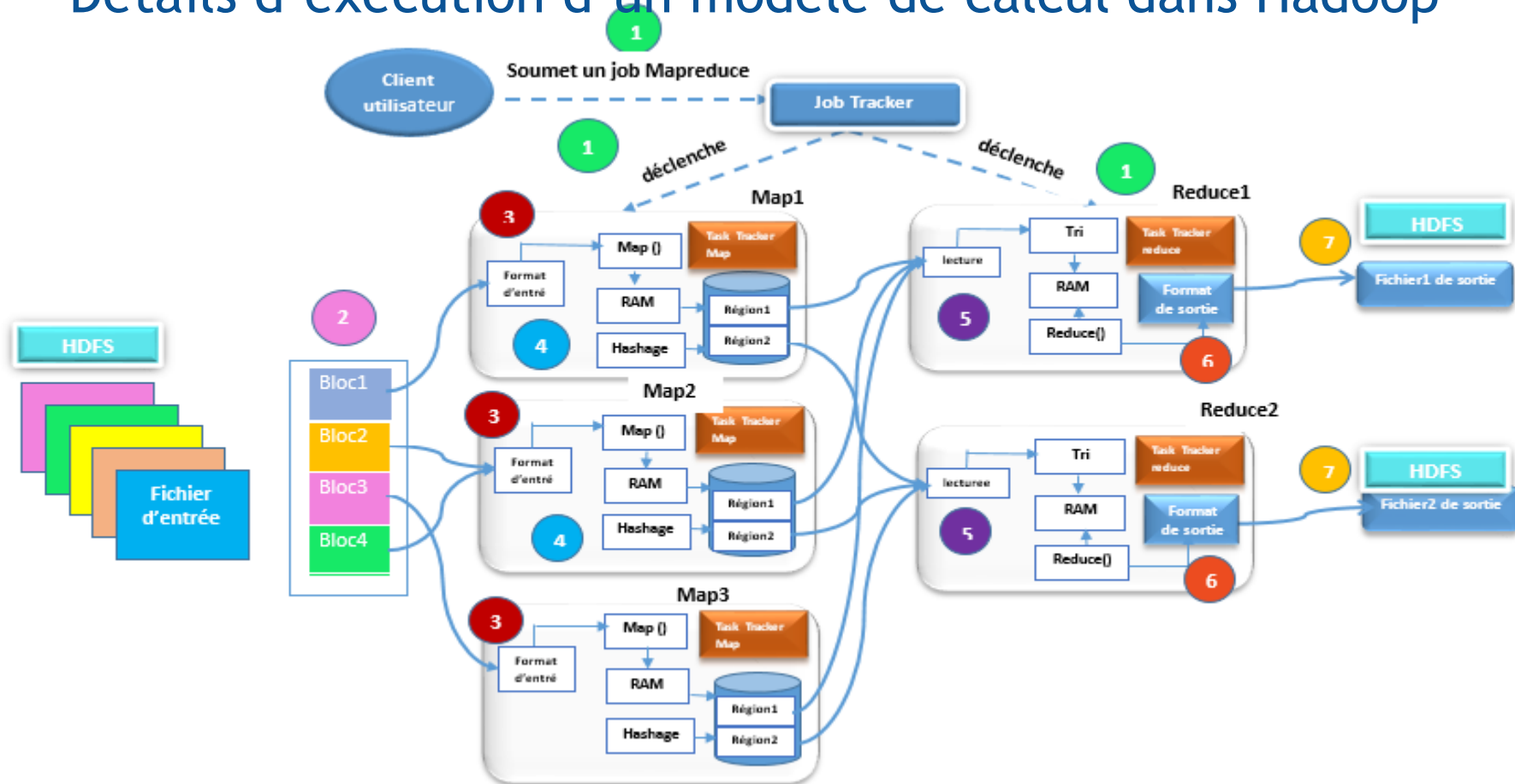
4. Périodiquement, dans chaque noeud, les paires clé/valeur sont sérialisées dans un fichier sur le disque dur local du noeud. Ensuite, ce fichier est partitionné en r régions (correspondant aux r tâches Reduce spécifiées) par une fonction de hachage qui va attribuer à chacune une clé correspondant à la tâche Reduce à laquelle elle a été assigné

Détails d'exécution d'un modèle de calcul dans Hadoop



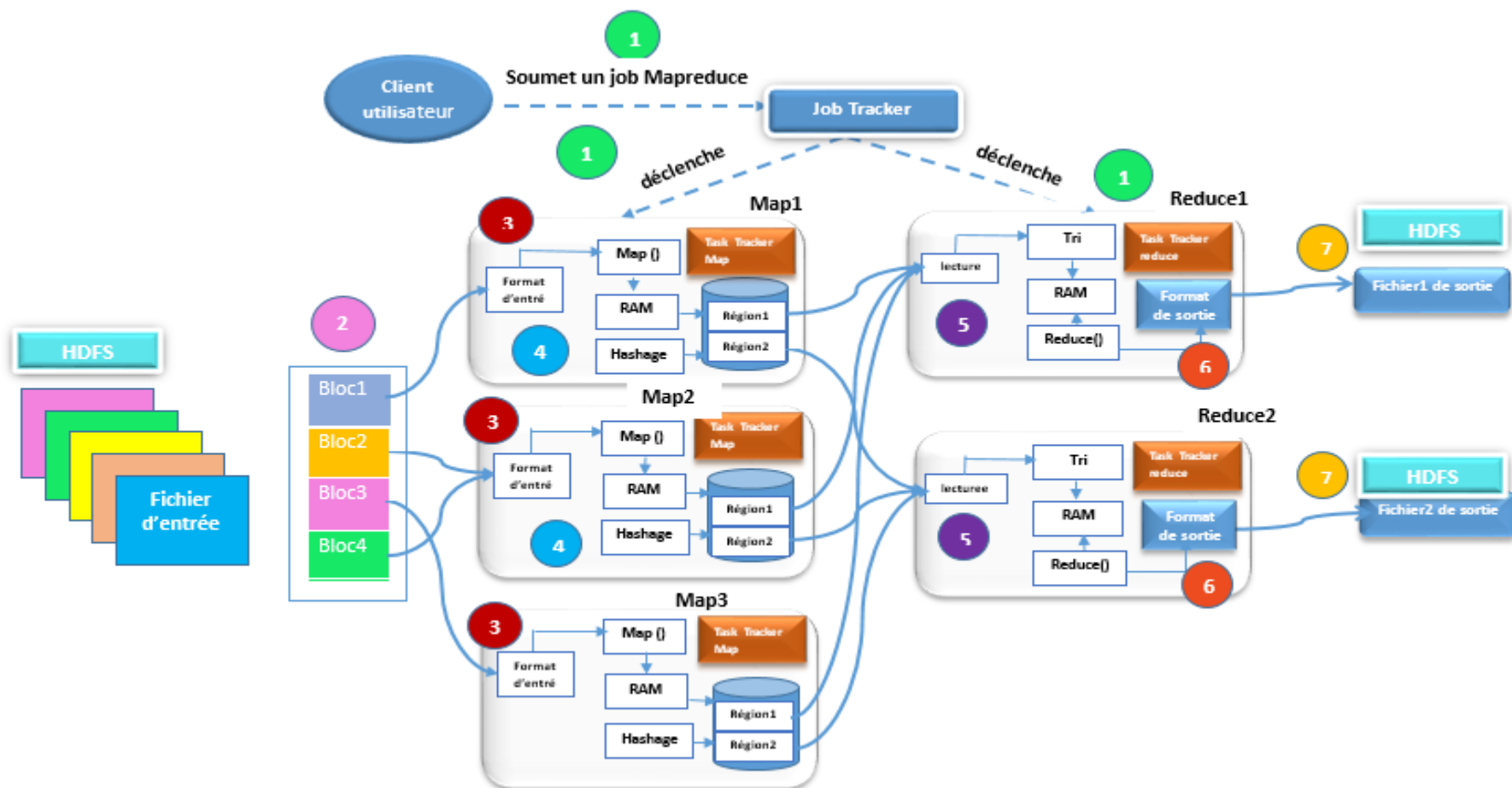
5. Lorsque les r TaskTrackers Reduce reçoivent les informations de localisation, ils utilisent des appels de procédures distantes (protocole RPC) pour lire – depuis le disque dur des noeuds sur lesquels les tâches Map se sont exécutées les régions des fichiers Map leur correspondant. Ensuite, ils les trient par clé.

Détails d'exécution d'un modèle de calcul dans Hadoop



6. Les TaskTrackers Reduce parcourent toutes les données triées, puis ils passent chaque clé unique rencontrée, avec sa valeur, à la fonction Reduce écrite par l'utilisateur. Les résultats du traitement de la fonction Reduce sont alors sérialisés dans un fichier selon le format de sortie spécifié par l'utilisateur.

Détails d'exécution d'un modèle de calcul dans Hadoop



7. Le job s'achève là ; à ce stade, les résultats sont disponibles et Hadoop applique, selon la demande de l'utilisateur, soit une capture Ecran, soit leur chargement dans un SGBD, soit encore leur passage comme fichiers d'entrée à un autre job MapReduce.

| | | | | |
|-----------------------------|------------------------|--|--|--|
| Introduction Au Big Data | L'écosystème Hadoop | | | |
|-----------------------------|------------------------|--|--|--|

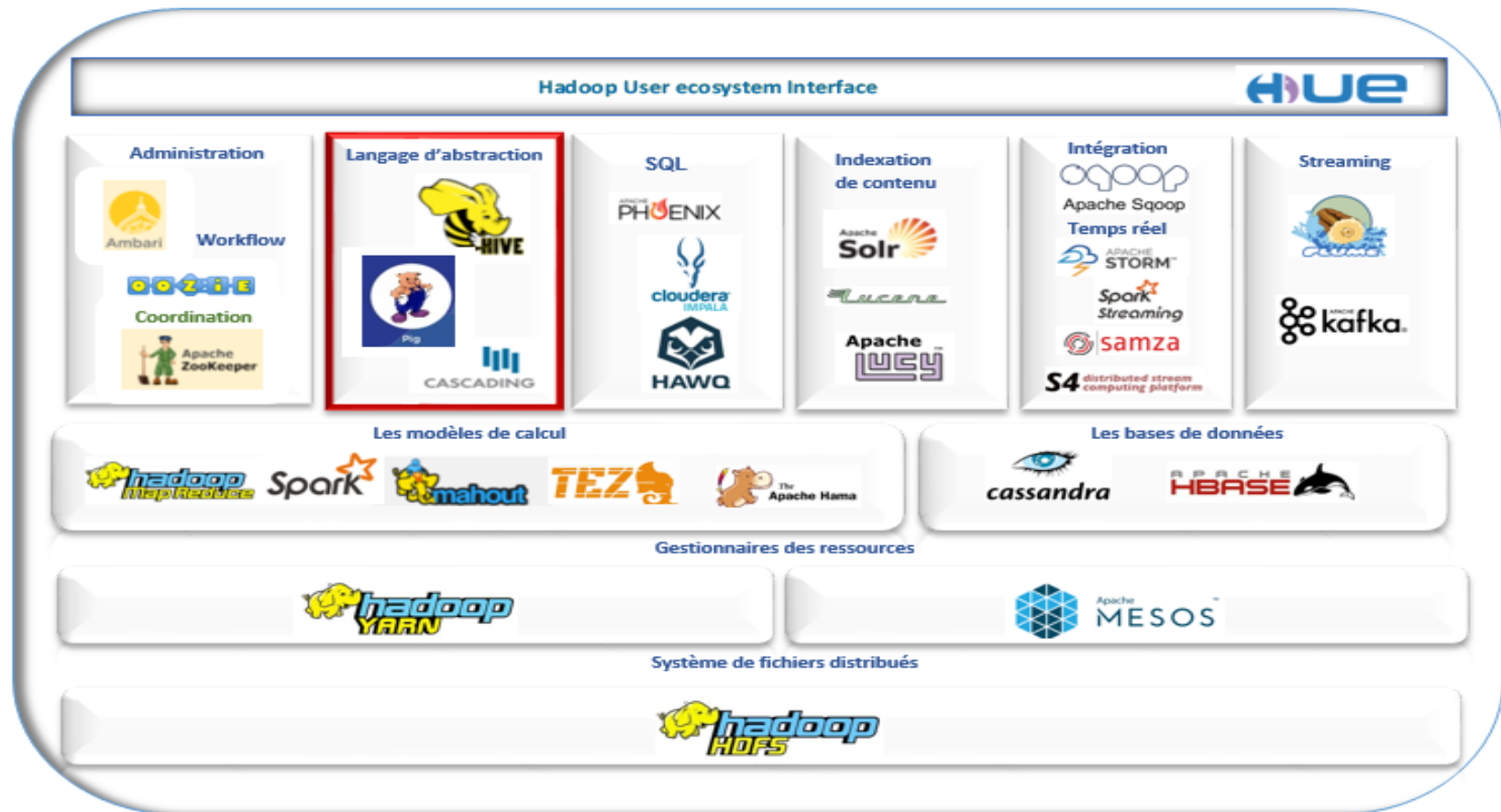
Taxonomie de l'écosystème Hadoop

Actuellement, de nombreux développeurs travaillent sur l'amélioration des outils et technologies de l'écosystème Hadoop : **Yahoo!**, par exemple, qui a développé **ZooKeeper** (service de coordination distribué), **Cloudera**, le créateur d'**Impala** (moteur de calcul SQL massivement parallèle), ou **LinkedIn**, le développeur de **Kafka** (système de messagerie publish/subscribe distribué).

L'écosystème Hadoop fournit une collection d'outils et de technologies spécialement conçue pour faciliter le développement, le déploiement et la prise en charge des solutions big data. La définition de cet écosystème est importante, car l'adoption d'Hadoop par les entreprises s'en trouve facilitée et leur permet de surmonter les défis.

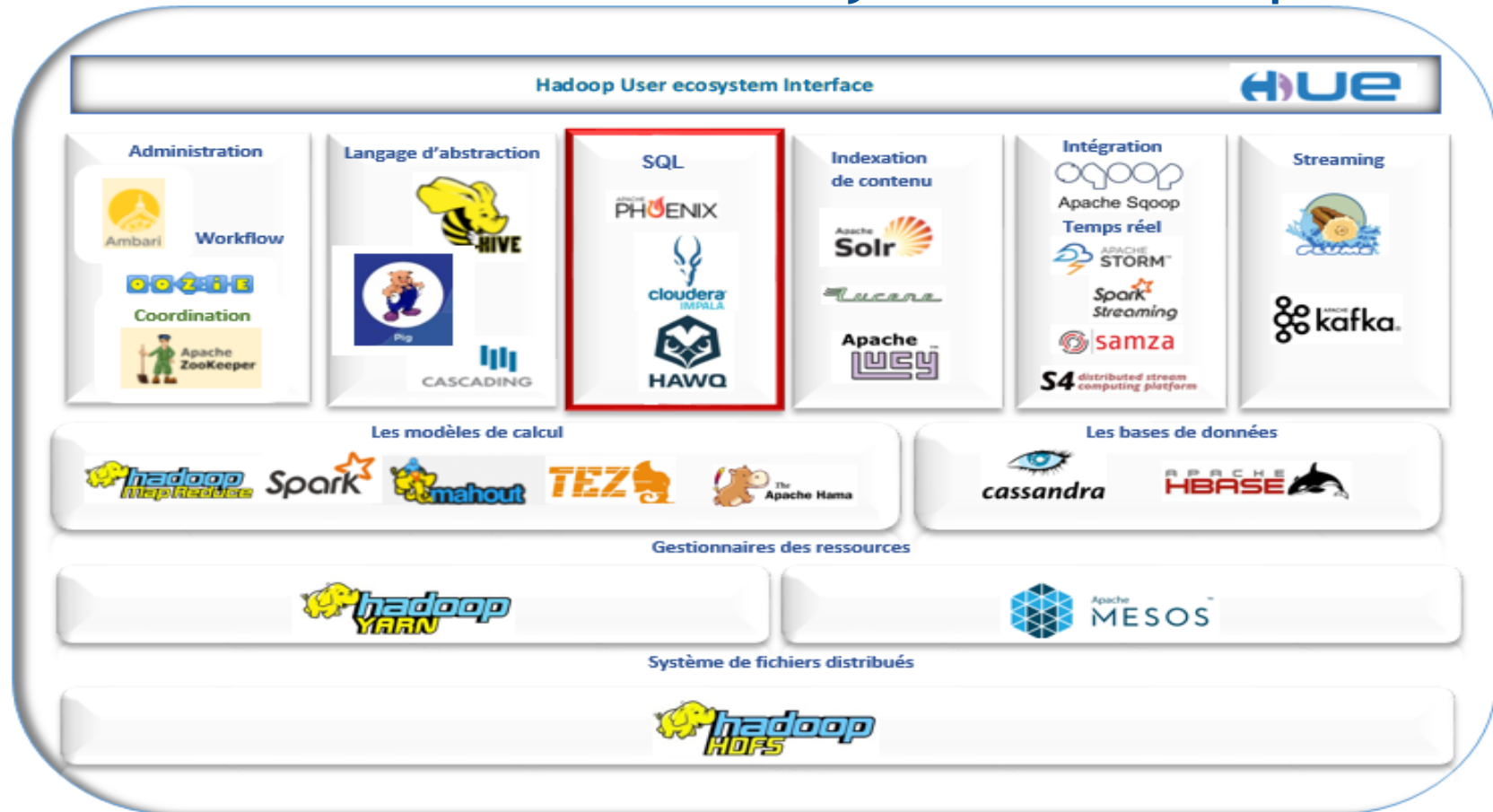
Tous ces outils peuvent être rangés selon 14 catégories :

Taxonomie de l'écosystème Hadoop



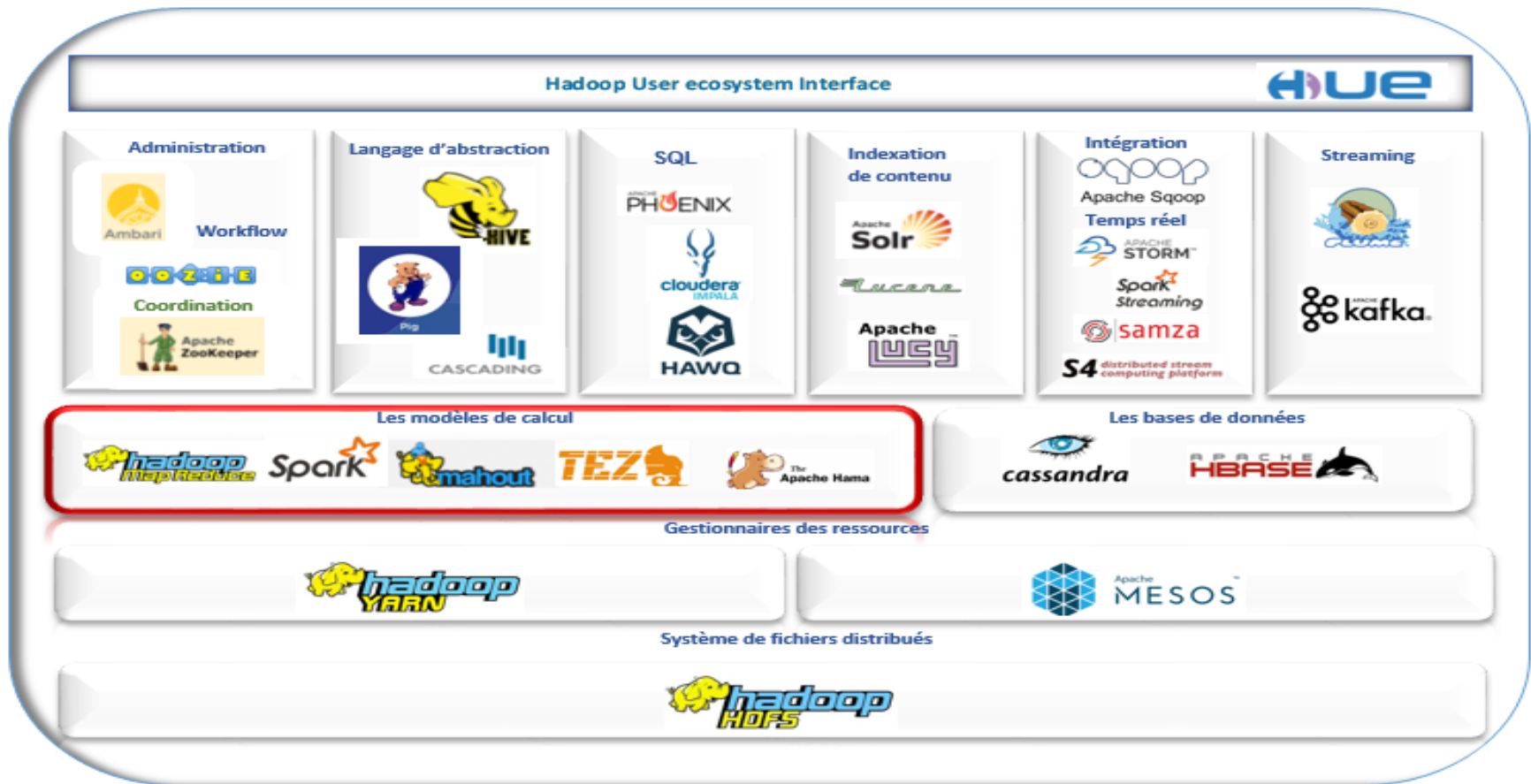
Langages d'abstraction. Ils servent à développer des jobs MapReduce à l'aide d'un langage similaire au SQL. Dans cette catégorie, on distingue principalement **Hive**, **Pig** et **Cascading**.

Taxonomie de l'écosystème Hadoop



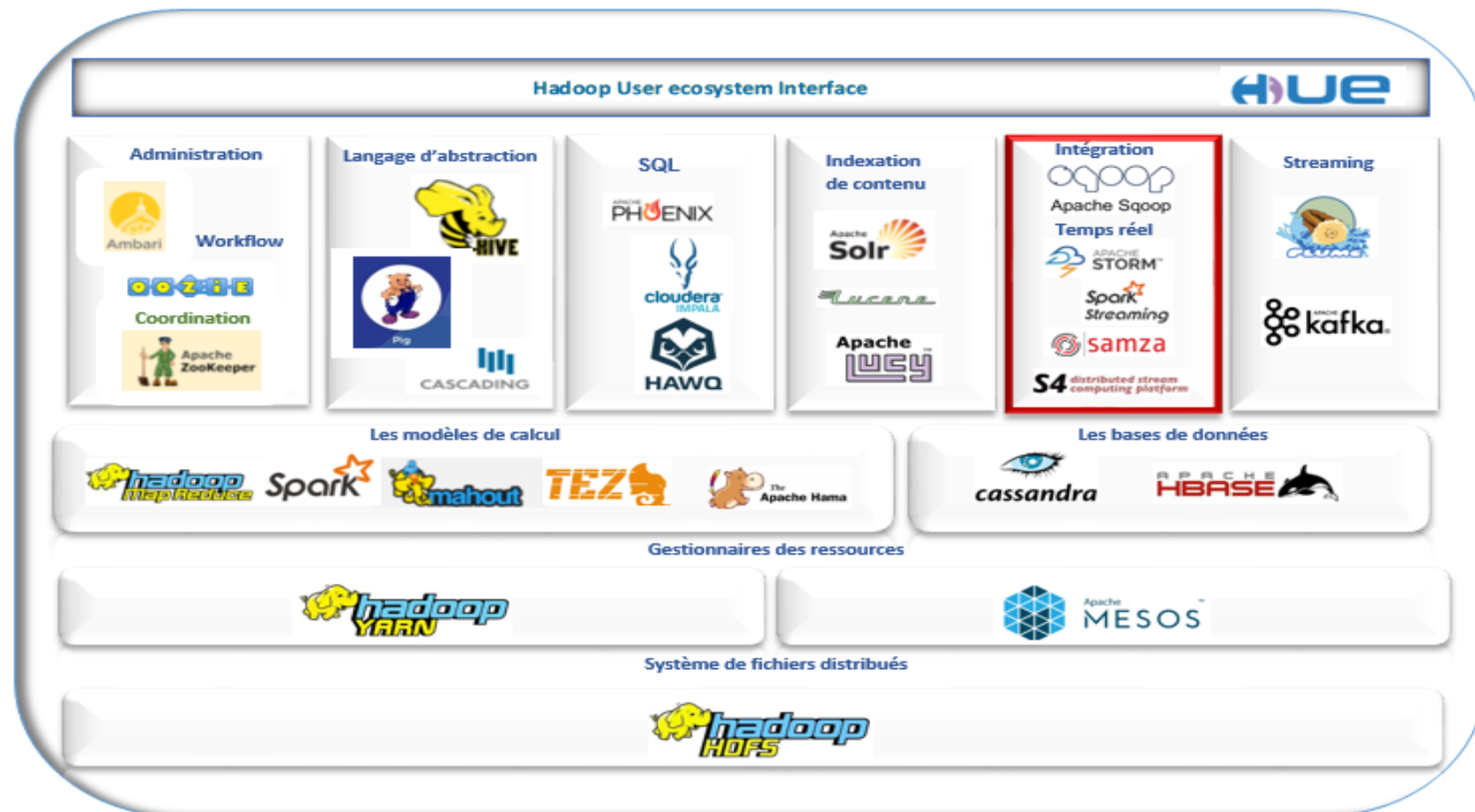
SQL sur Hadoop. Il s'agit de technologies permettant d'exécuter nativement du SQL sur un cluster Hadoop. À ce jour, trois projets sont en incubation à la fondation Apache, bien qu'opérationnels: **Impala**, **Phoenix** et **HAWQ**.

Taxonomie de l'écosystème Hadoop



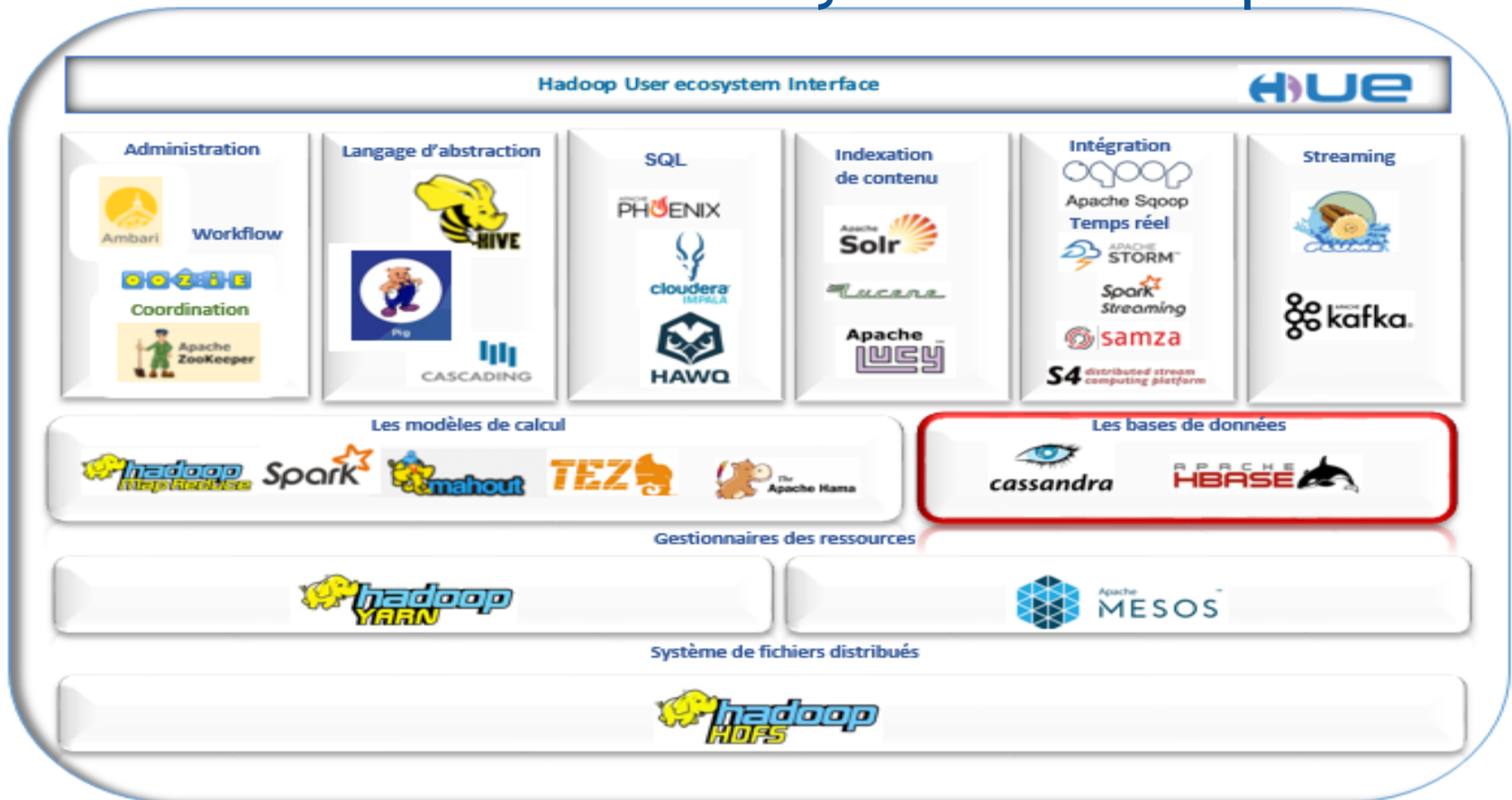
Modèles de calcul. Ce sont des moteurs d'exécution d'algorithmes en parallèle ou des modèles de programmation des tâches distribuées sur un cluster. Actuellement, Hadoop dispose de cinq modèles de calcul : **MapReduce**, le modèle original d'Hadoop, **Spark**, le moteur in-memory distribué créé par l'université de Berkeley, **Mahout**, **Hama** et **Tez**.

Taxonomie de l'écosystème Hadoop



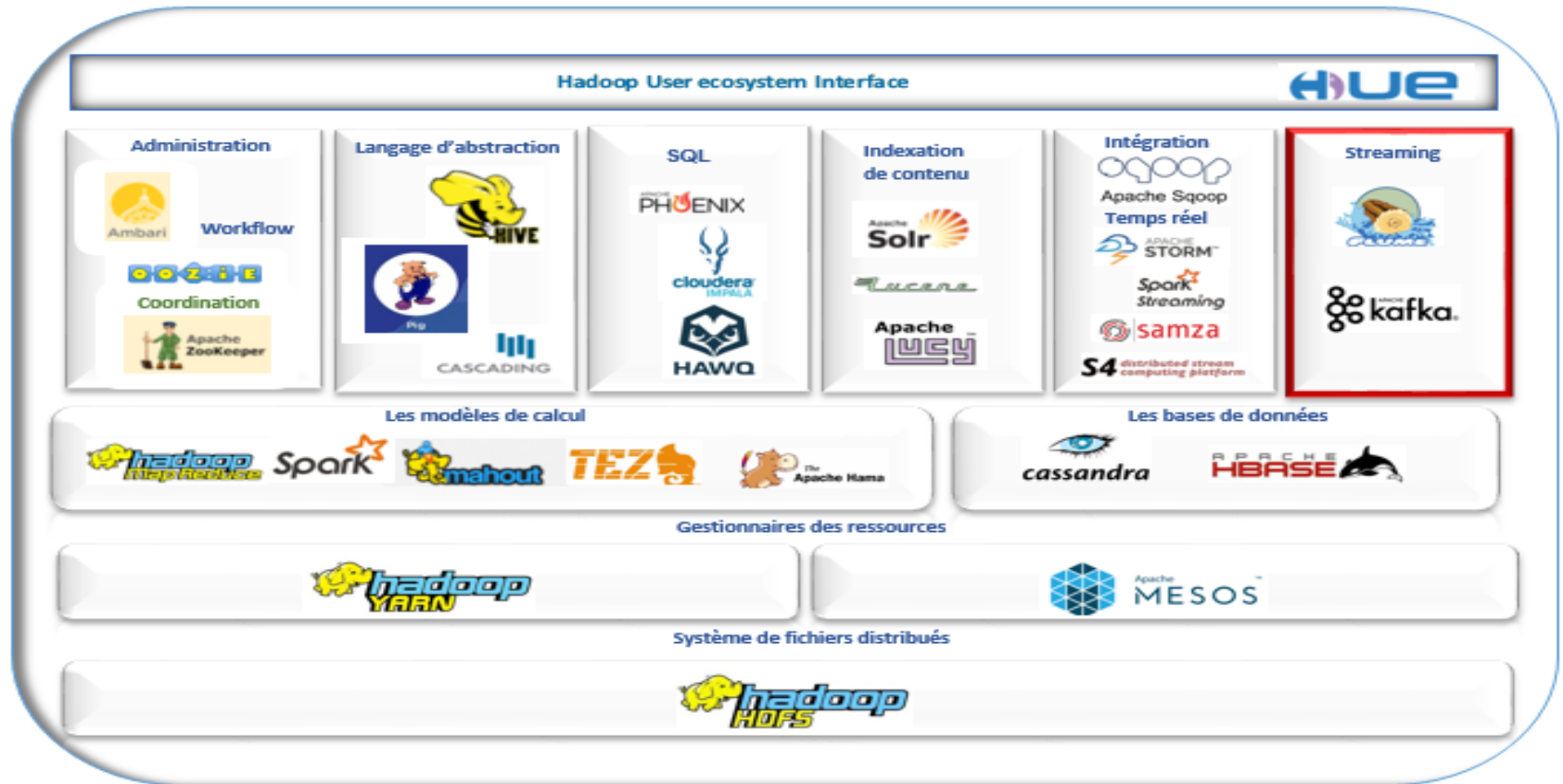
Outils de traitement en temps réel. Ils servent à traiter immédiatement des données générées en streaming(diffusées en continu). Quatre outils sont disponibles dans cette catégorie : **Storm**, **Samza**, **S4** et **Spark Streaming**.

Taxonomie de l'écosystème Hadoop



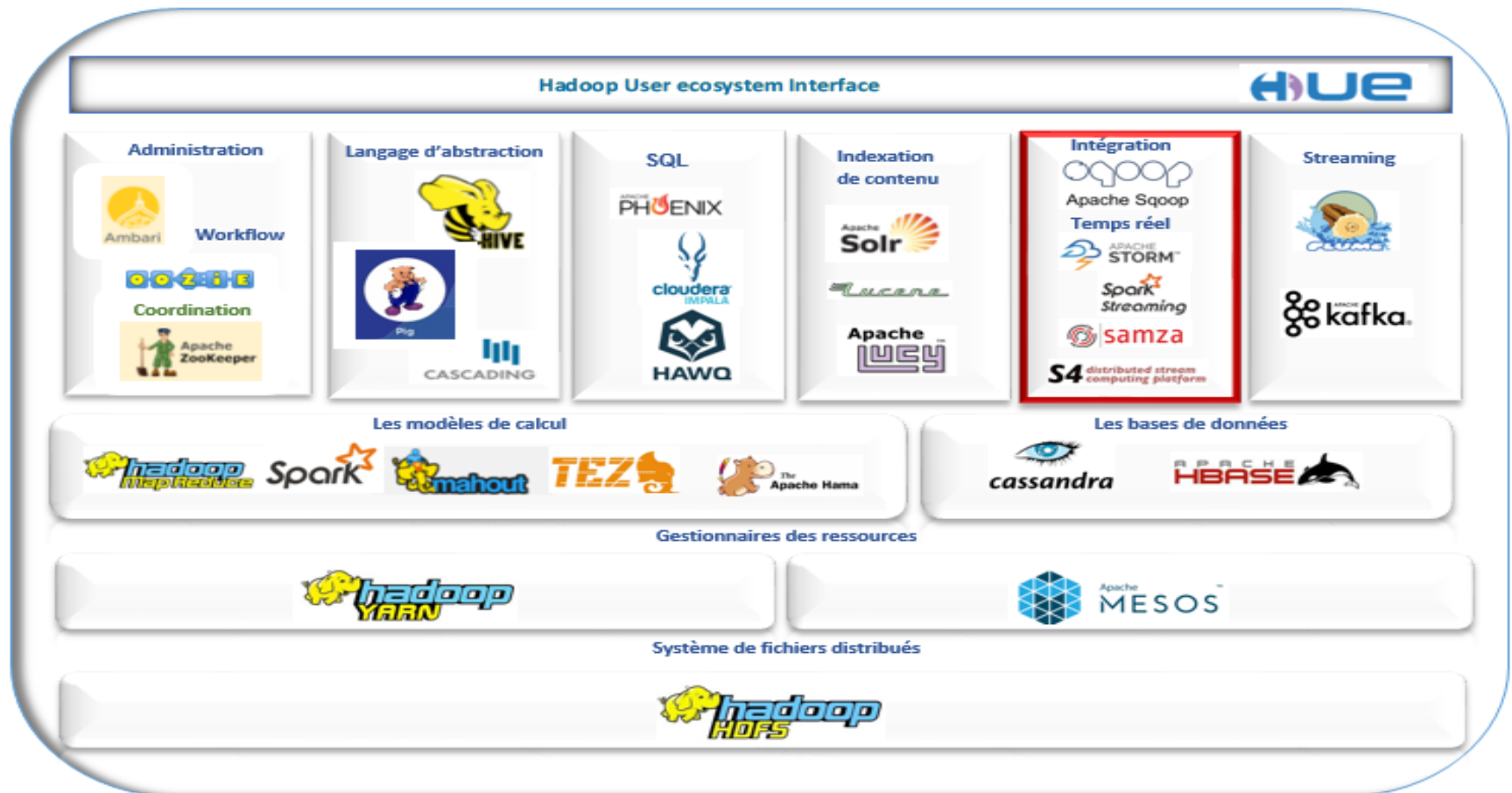
Bases de données. Ce sont des systèmes de gestion de bases de données distribuées (SGBDD), adaptés au fonctionnement sur un cluster. Plusieurs SGBDD open source existent, mais ils ne font pas partie de la fondation Apache. Les seuls systèmes attachés à la fondation sont, à notre connaissance, **Hbase et Cassandra**.

Taxonomie de l'écosystème Hadoop



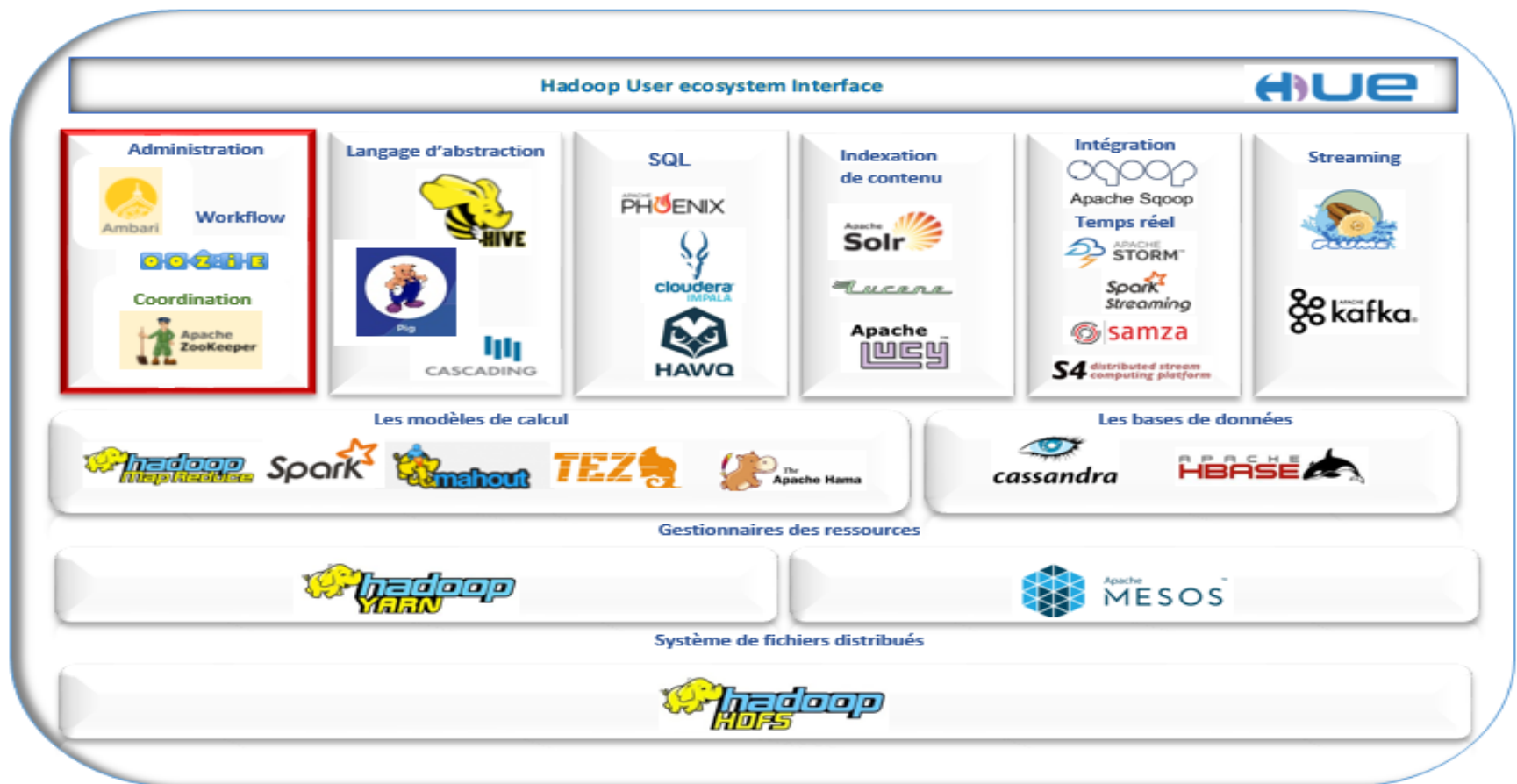
Outils d'ingestion & streaming. Permettent d'« ingérer » des données générées en streaming (diffusées en continu). Il s'agit d'outils capables de collecter et d'analyser les données en temps réel. La fondation Apache en propose deux : **Kafka** et **Flume**.

Taxonomie de l'écosystème Hadoop



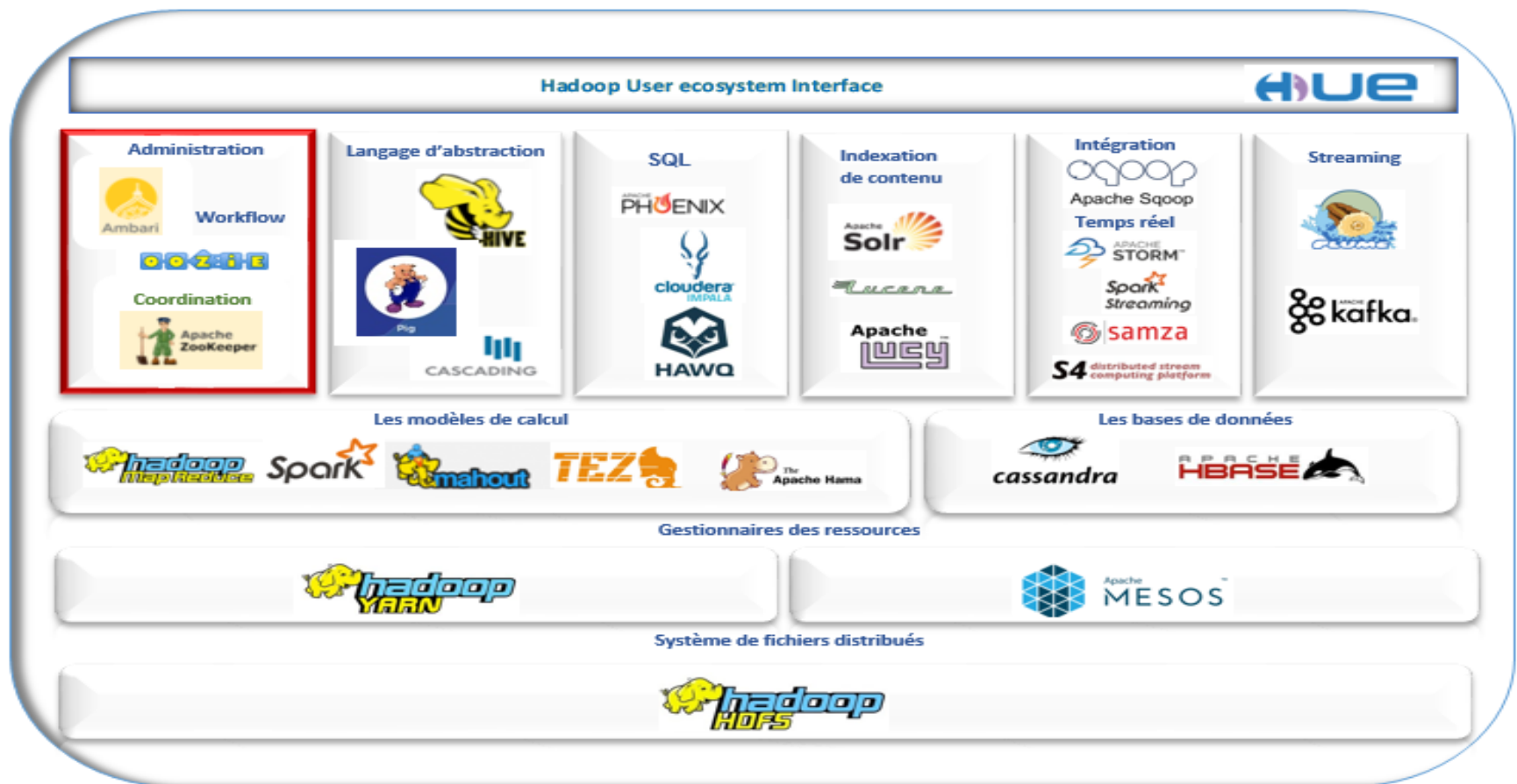
Outils d'intégration des données. Ils permettent de déplacer les données d'un SGBDR vers le HDFS et vice versa. Le seul outil disponible dans cette catégorie est **Sqoop**,

Taxonomie de l'écosystème Hadoop



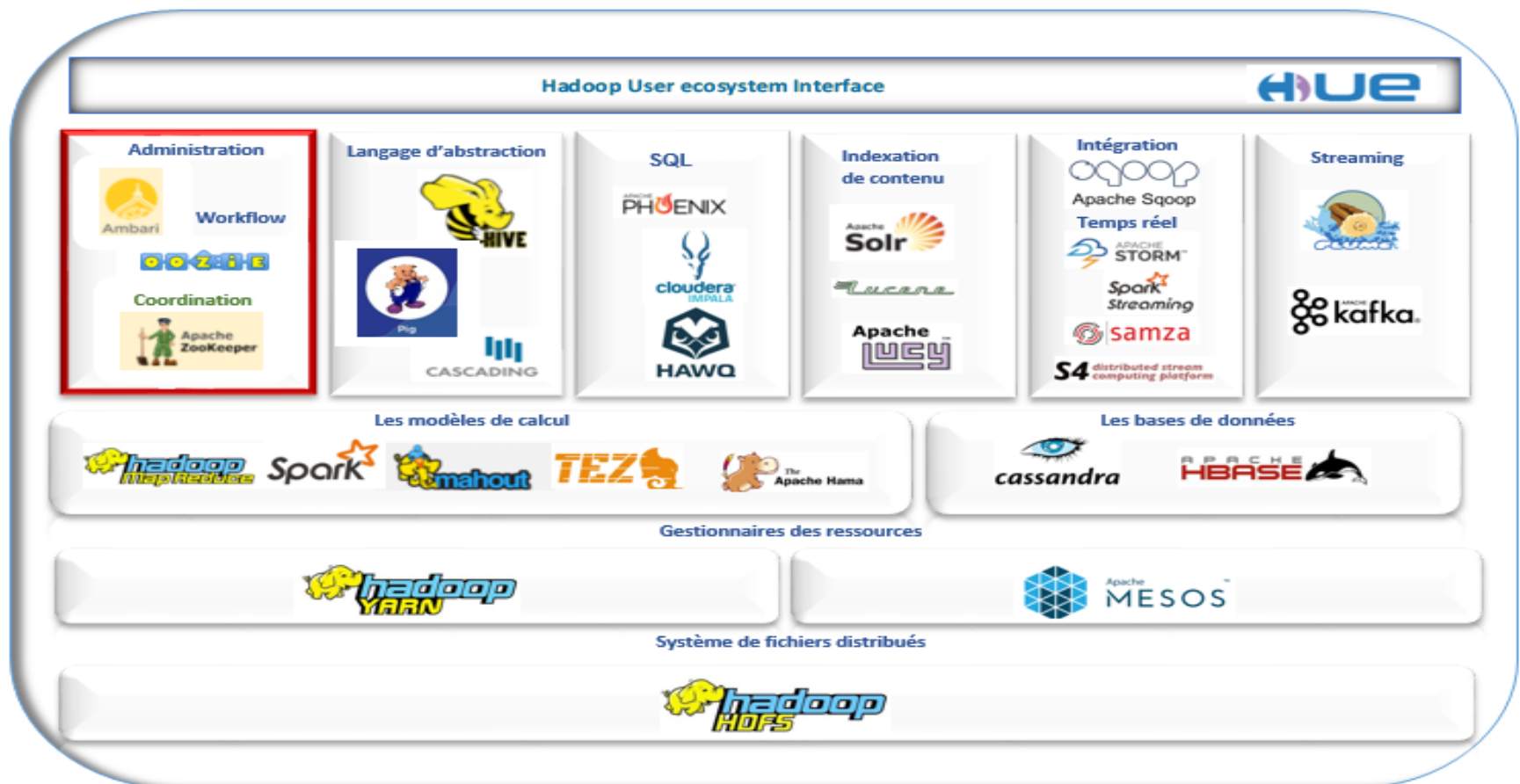
Outils de coordination de workflow. Ils servent à planifier et chaîner l'exécution de plusieurs jobs dans le cluster Hadoop. **Oozie** est le seul outil disponible dans cette catégorie.

Taxonomie de l'écosystème Hadoop



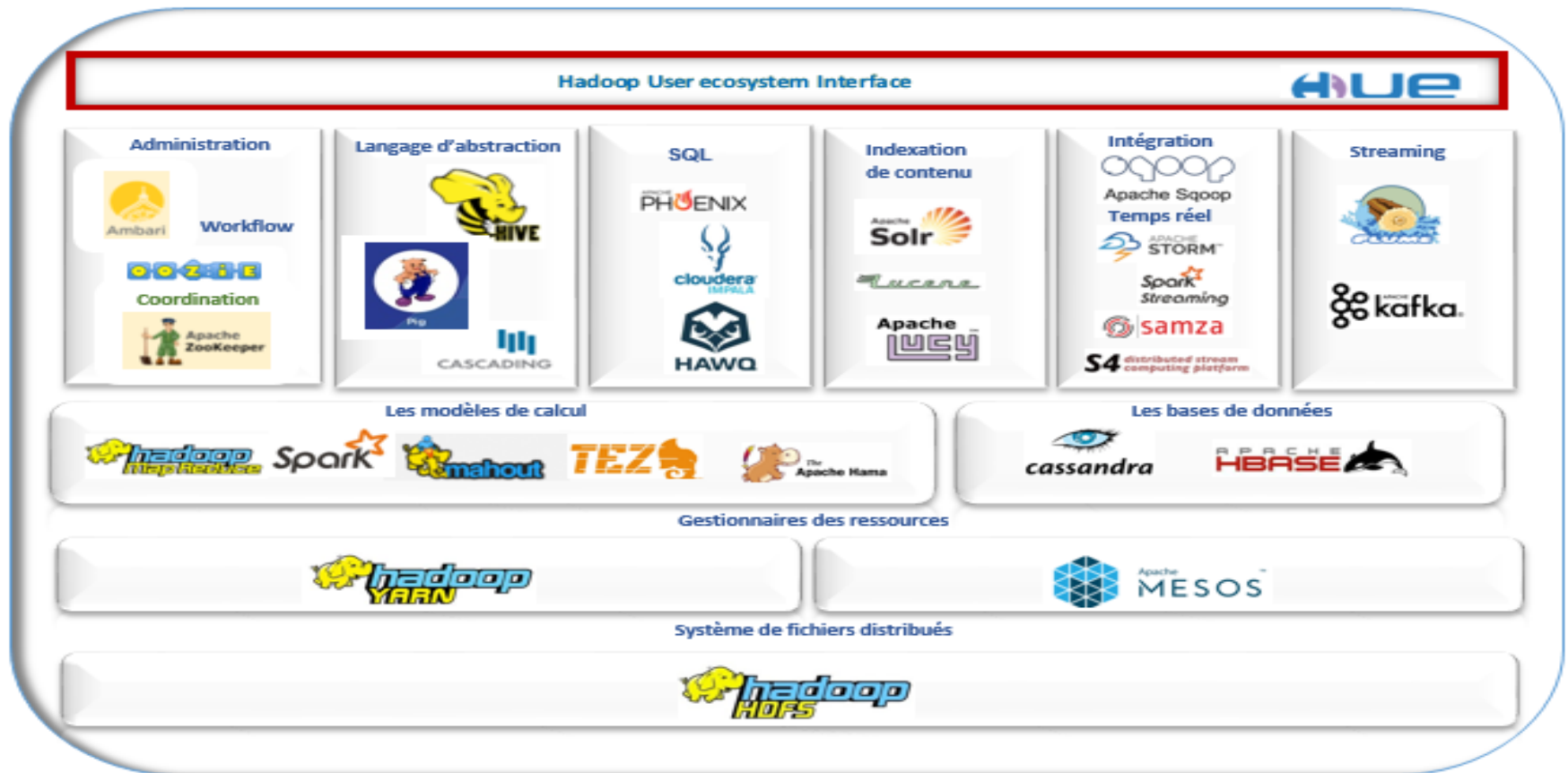
Outils de coordination de services distribués. Ils coordonnent les échanges entre les nœuds d'un cluster, ou entre plusieurs applications. **ZooKeeper** est l'outil de coordination de service distribué le plus populaire à ce jour.

Taxonomie de l'écosystème Hadoop



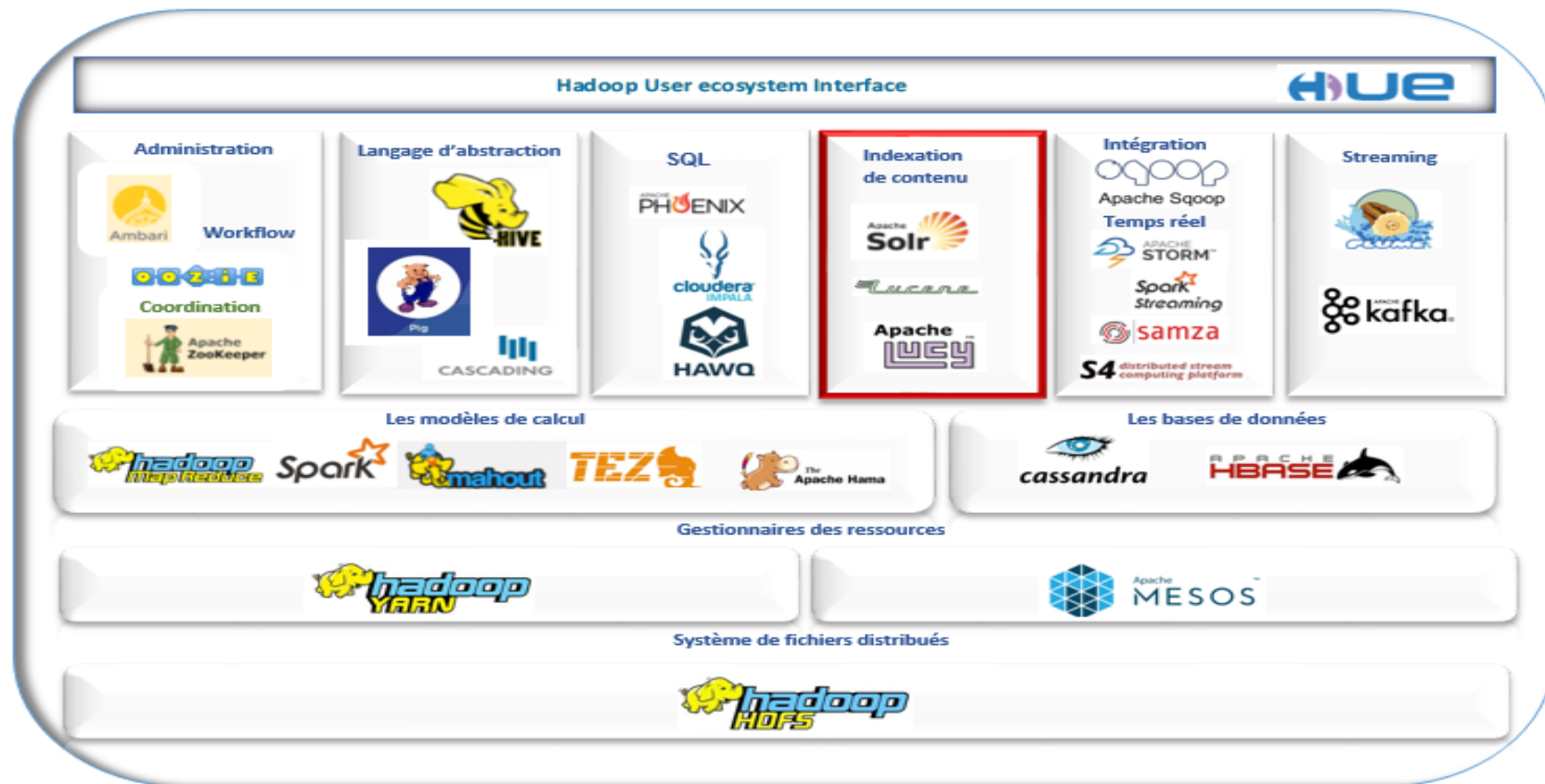
Outils d'administration de cluster. Ils servent notamment à gérer l'approvisionnement des ressources du cluster et les permissions des utilisateurs, à y suivre l'exécution des tâches, etc. Plusieurs outils indépendants ont été développés pour l'administration Hadoop, combinés dans **Ambari**.

Taxonomie de l'écosystème Hadoop



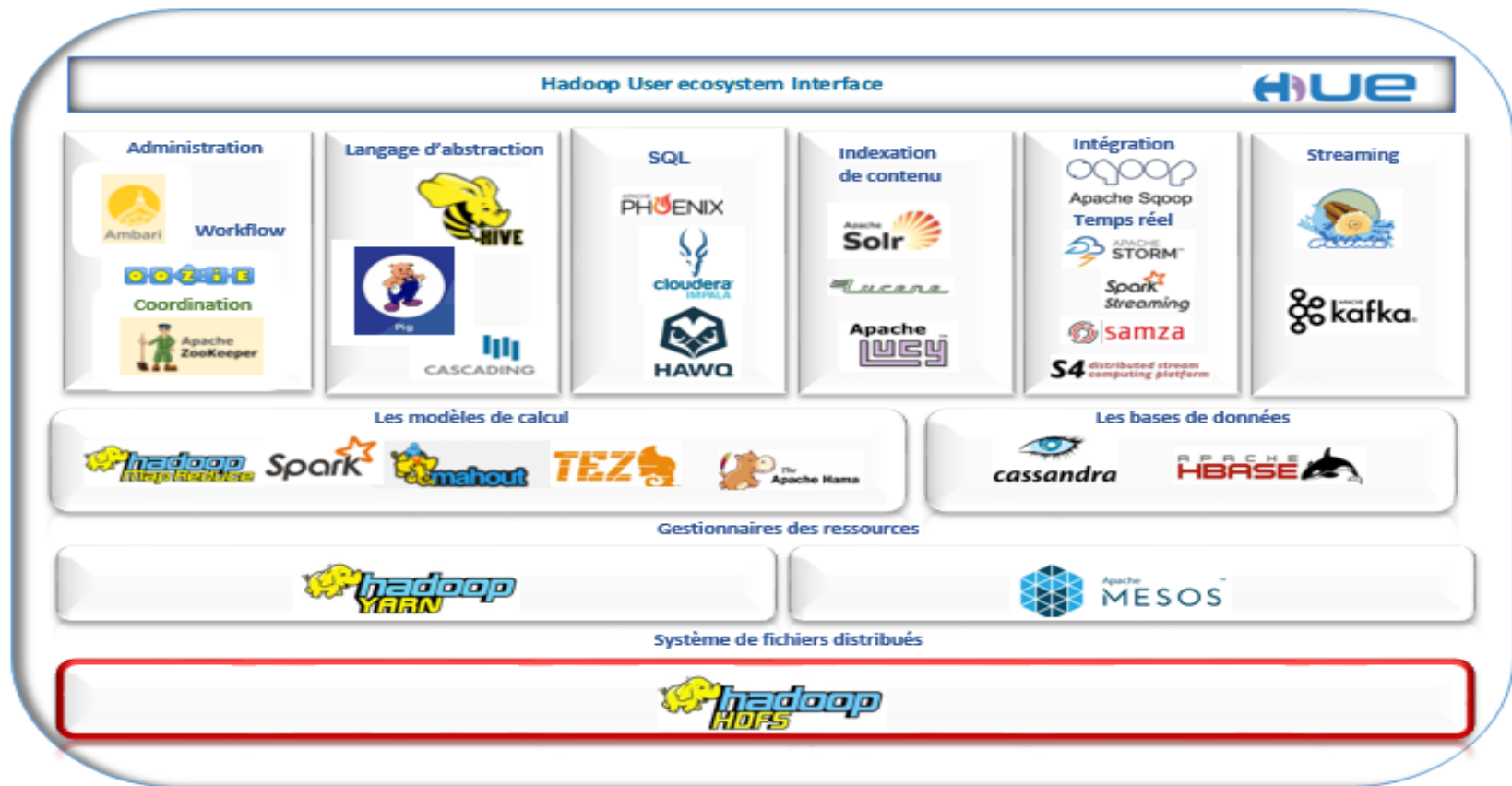
Outils d'interface utilisateur. C'est grâce à eux que l'utilisateur interagit avec le cluster et les outils d'Hadoop. Actuellement, la fondation Apache offre une interface web appelée **HUE**.

Taxonomie de l'écosystème Hadoop



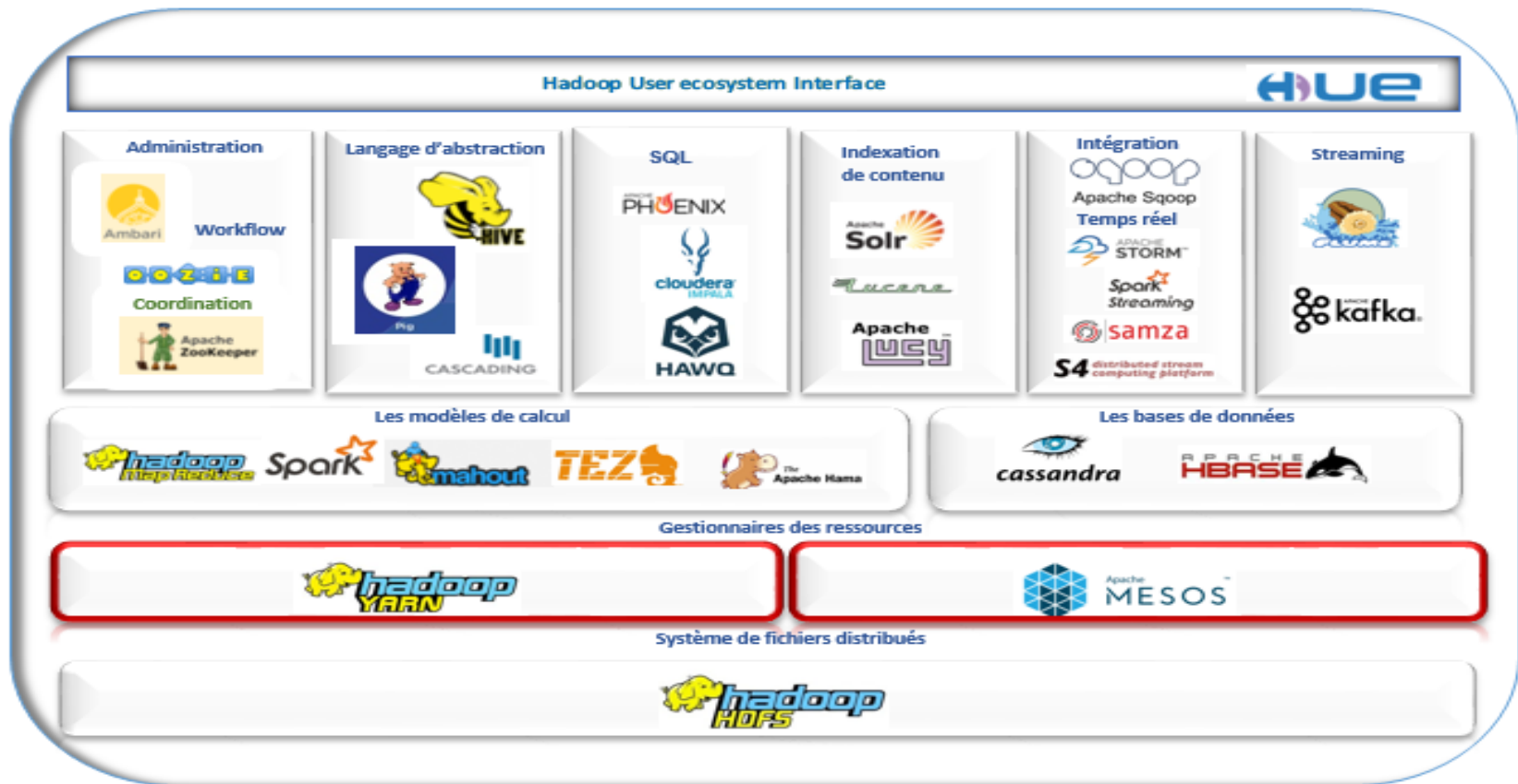
Outils d'indexation de contenu. Ils rendent possible le *full text search*, c'est-à-dire la recherche en texte intégral. Ils permettent d'effectuer des recherches de contenu, d'indexer le texte des pages Web et fournissent des fonctionnalités de recherche sur ces dernières. Dans cette catégorie, on distingue les moteurs **Lucene**, **Solr** et **Lucy**.

Taxonomie de l'écosystème Hadoop



Systèmes de fichiers distribués. Il s'agit de couches abstraites, installées sur le cluster pour gérer le stockage des données distribuées. Hadoop est fourni avec son système de fichiers distribués initial, le **HDFS**.

Taxonomie de l'écosystème Hadoop



Gestionnaires de ressources. Ce sont des outils qui exploitent au mieux les ressources d'un cluster. Ils permettent d'exécuter plusieurs modèles de calcul sur un cluster. Souvenez-vous qu'Hadoop est originellement l'implémentation du MapReduce ; Actuellement, la fondation Apache en fournit deux autres : **YARN** et **Mesos**.