# Systèmes de métadonnées dans les lacs de données : modélisation et fonctionnalités

Étienne Scholly*,**, Pegdwendé N. Sawadogo*,
Cécile Favre*, Éric Ferey**, Sabine Loudcher*, Jérôme Darmont*

*Université de Lyon, Lyon 2, ERIC EA 3083
{etienne.scholly, pegdwende.sawadogo,
cecile.favre, sabine.loudcher, jerome.darmont}@univ-lyon2.fr
https://eric.ish-lyon.cnrs.fr/
**BIAL-X
{etienne.scholly, eric.ferey}@bial-x.com
https://www.bial-x.com/

**Résumé.** Au cours de la dernière décennie, le concept de lac de données a émergé comme une alternative aux entrepôts de données pour le stockage et l'analyse des mégadonnées. Le lac de données propose un stockage des données sans schéma prédéfini. En l'absence de schéma, l'interrogation et l'analyse des données dépendent alors d'un système de métadonnées qui se doit d'être efficace et complet. Cependant, la gestion des métadonnées dans les lacs de données demeure une problématique d'actualité et les critères d'évaluation de son efficacité sont peu ou prou inexistants.

Dans cet article, nous proposons MEDAL, un modèle générique pour la gestion des métadonnées d'un lac de données. MEDAL adopte une modélisation du système de métadonnées à base de graphes. Nous proposons aussi des critères d'évaluation du système de métadonnées d'un lac de données à travers une liste de fonctionnalités attendues et montrons que notre approche est plus complète que les systèmes de métadonnées existants.

## 1 Introduction

Depuis le début du 21e siècle, les usages des organisations dans les processus de prise de décision sont bouleversés par la disponibilité de grands volumes de données appelées *big data*. Ces mégadonnées, principalement issues des médias sociaux (Facebook, Twitter, Wikipédia, Youtube, etc.) et des objets connectés (*Internet of Things*), constituent une véritable opportunité pour les organisations. Cependant, elles s'accompagnent entre autres de problématiques de volume, de vélocité et de variété, qui surpassent les capacités des systèmes traditionnels de stockage et de traitement des données (Miloslavskaya et Tolstoy, 2016).

C'est dans ce contexte que Dixon (2010) introduit le concept de lac de données (*data lake*), en guise de solution aux problèmes induits par l'hétérogénéité des mégadonnées. Un lac de données propose un stockage intégré des données sans schéma prédéfini (Hai et al., 2016). En

l'absence de schéma de données, un système de métadonnées efficace devient alors essentiel pour rendre les données interrogeables et empêcher ainsi le lac de se transformer en marécage de données (*data swamp*), c'est-à-dire en un lac de données inutilisable (Alrehamy et Walker, 2015; Hai et al., 2016; Suriarachchi et Plale, 2016).

Si la littérature semble unanime sur l'importance du système de métadonnées dans un lac de données, des interrogations et des incertitudes subsistent toutefois sur la méthodologie à suivre pour le mettre en place. Plusieurs approches ont ainsi été proposées pour organiser les métadonnées dans un lac de données, mais la plupart d'entre elles concernent uniquement les données structurées et semi-structurées (Quix et al., 2016; Hai et al., 2016; Farid et al., 2016; Maccioni et Torlone, 2018). De plus, l'efficacité d'un système de métadonnées est difficile à mesurer dans le sens où il n'existe pas, à notre connaissance, de critères majoritairement partagés et acceptés pour l'évaluation d'un modèle de métadonnées.

Pour répondre à ces problématiques, nous identifions dans un premier temps un ensemble de fonctionnalités que devrait idéalement proposer le système de métadonnées d'un lac de données. En comparant plusieurs systèmes de métadonnées par rapport à ces fonctionnalités, il ressort qu'aucun d'entre eux ne propose l'ensemble des fonctionnalités attendues. C'est pourquoi nous proposons dans un second temps un modèle de métadonnées qui se veut plus générique et plus complet. Notre modèle de métadonnées, nommé *MEtadata model for DAta Lakes* (MEDAL), adopte une modélisation du système de métadonnées à base de graphes. Il s'appuie sur une typologie des métadonnées d'un lac de données en métadonnées intra-objets, inter-objets et globales.

Le reste de cet article est organisé comme suit. La Section 2 introduit le concept de lac de données. La Section 3 détaille les fonctionnalités attendues du système de métadonnées d'un lac de données, et compare plusieurs travaux sur l'organisation des métadonnées suivant ces fonctionnalités. La Section 4 présente une typologie des métadonnées sur laquelle s'appuie notre modèle. La Section 5 formalise notre modèle de métadonnées et en propose une représentation sous forme de graphe. Enfin, la Section 6 conclut l'article et présente nos perspectives de recherche.

# 2 Concept de lac de données

Le concept de lac de données étant relativement récent, il nous paraît primordial de le préciser en présentant dans la Section 2.1 plusieurs définitions de la littérature, puis en proposant dans la Section 2.2 notre propre définition en guise de synthèse.

## 2.1 Définitions de la littérature

Le concept de lac de données a été introduit par Dixon (2010) comme une alternative aux magasins de données (*data marts*), qui sont des sous-ensembles des entrepôts, auxquels il reproche de mettre les données en silos. Le lac de données tel que conçu par Dixon est un vaste dépôt de données brutes de structures hétérogènes, alimenté par des sources de données externes et à partir duquel des analyses diverses peuvent être réalisées.

Suite à la définition de Dixon, certains travaux ont rapidement associé le lac de données à la technologie Hadoop (O'Leary, 2014; Fang, 2015). Ainsi, Fang considère le lac de données

comme une méthodologie consistant à utiliser des technologies libres ou peu coûteuses, typiquement Hadoop, pour assurer le stockage, le traitement et l'exploration des données brutes au sein d'une entreprise. Cependant, cette vision est de plus en plus minoritaire dans la littérature, le concept de lac de données étant désormais également associé à des solutions propriétaires comme Azure ou IBM (Madera et Laurent, 2016; Sirosh, 2016).

Une définition plus consensuelle consiste à voir un lac de données comme un dépôt central où des données de tous formats sont stockées sans schéma strict (Laskowski, 2016; Khine et Wang, 2017; Mathis, 2017). Cette définition est basée sur deux caractéristiques clés du lac de données : la variété des données et l'approche *schema-on-read*. La variété des données désigne le fait d'intégrer des données de tous types, et donc hétérogènes. La propriété *schema-on-read* consiste à définir le schéma des données seulement lors de leur analyse (Miloslavskaya et Tolstoy, 2016). On parle aussi d'approche *late binding*, à l'inverse de l'approche *early binding* des systèmes d'information décisionnels traditionnels (Fang, 2015).

Toutefois, il convient de noter que la définition basée sur la propriété *schema-on-read* et la variété des données, bien que consensuelle, demeure incomplète dans le sens où elle donne peu de détails sur les caractéristiques d'un lac de données. C'est pourquoi Madera et Laurent (2016) introduisent une nouvelle définition plus complète. Un lac de données est alors considéré comme une vue logique de toutes les sources de données et de tous les ensembles de données dans leur format brut, accessible par des *data scientists* ou statisticiens pour l'extraction de connaissances. Cette définition est complétée par une liste de caractéristiques clés : 1) la qualité des données dans le lac est assurée par un ensemble de métadonnées ; 2) le lac est contrôlé par une politique et des outils de gouvernance des données ; 3) l'utilisation du lac est limitée aux statisticiens et aux *data scientists* ; 4) le lac intègre des données de tous types et formats ; 5) le lac de données possède une organisation logique et physique.

## 2.2 Proposition de définition

La définition la plus complète des lacs de données est celle de Madera et Laurent (2016), qui en plus de la variété des données et de l'approche *schema-on-read*, définit des caractéristiques supplémentaires. Cependant, certaines caractéristiques proposées dans cette définition sont discutables de notre point de vue. En effet, les auteures réservent l'utilisation du lac aux spécialistes des données et excluent *de facto* les experts métiers pour des raisons de sécurité. Pourtant, il est tout à fait envisageable, selon nous, de donner un accès contrôlé à ce type d'utilisateurs à travers une plateforme de navigation ou d'analyse.

De plus, nous ne partageons pas la vision du lac de données comme une vue logique des sources de données, dans le sens où les sources de données sont parfois extérieures à l'entreprise, et donc au lac de données. C'est d'ailleurs ce qui ressort de la définition initiale de Dixon (2010), qui précise que les données du lac proviennent de sources de données. L'inclusion des sources de données dans le lac peut donc être considérée comme contraire à l'esprit des lacs de données.

Enfin, bien que plus complète, la définition de Madera et Laurent (2016) omet une propriété essentielle des lacs de données, que nous retrouvons notamment dans les définitions de Miloslavskaya et Tolstoy (2016) et Haste (2017) : la capacité de passage à l'échelle. En effet, un lac de données étant destiné au stockage et au traitement des mégadonnées, il est indispensable de traiter la problématique liée au volume des données en assurant le passage à l'échelle, sans quoi le lac de données serait difficilement exploitable.

Au vu de ce qui précède, nous proposons une nouvelle définition des lacs de données qui vise à être aussi complète que celle de Madera et Laurent (2016), mais plus conforme à notre vision du concept de lac de données. Cette définition amende celle de Madera et Laurent (2016) et introduit la caractéristique de passage à l'échelle.

**Définition 1** *Un lac de données est un système évolutif (en termes de passage à l'échelle) de stockage et d'analyse de données de tous types, dans leur format natif, utilisé principalement par des spécialistes des données (statisticiens, data scientists, data analysts) pour l'extraction de connaissances. Les caractéristiques d'un lac de données incluent : 1) un catalogue de métadonnées qui assure la qualité des données ; 2) une politique et des outils de gouvernance des données ; 3) l'ouverture à tous types d'utilisateurs ; 4) l'intégration de données de tous types ; 5) une organisation logique et physique ; 6) le passage à l'échelle.*

# 3 Fonctionnalités de base d'un système de métadonnées

Le concept de lac de données prône un nouveau paradigme de stockage et d'analyse des données à travers une approche *schema-on-read*, c'est-à-dire une définition *a posteriori* du schéma des données. Sans schéma prédéfini, un système de métadonnées efficace est alors un élément indispensable pour rendre les données interrogeables, et empêcher ainsi le lac de se transformer en *data swamp*.

Cependant, il n'existe pas à notre connaissance de critères objectifs de mesure de l'efficacité d'un système de métadonnées dans le contexte des lacs de données. C'est pourquoi nous proposons, d'une part, une liste de fonctionnalités clés attendues idéalement du système de métadonnées d'un lac de données (Section 3.1) et, d'autre part, une comparaison de plusieurs systèmes de métadonnées sur la base de ces fonctionnalités (Section 3.2).

## 3.1 Fonctionnalités attendues du système de métadonnées

Nous identifions dans la littérature six fonctionnalités principales que devrait idéalement proposer le système de métadonnées d'un lac de données.

L'**enrichissement sémantique (ES)**, aussi appelé annotation sémantique (Hai et al., 2016) ou profilage sémantique (Ansari et al., 2018), consiste à générer une description du contexte des données (avec des *tags*, par exemple) pour les rendre plus interprétables et compréhensibles (Terrizzano et al., 2015). Il se fait à l'aide de bases de connaissances telles que des ontologies. L'annotation sémantique joue un rôle clé dans l'exploitation des données, dans le sens où elle permet de résumer les ensembles de données contenus dans le lac de sorte qu'ils soient plus compréhensibles par l'utilisateur. Elle peut également servir de base à l'identification de liaisons entre les données.

L'**indexation des données (ID)** consiste à mettre en place une structure de données permettant de retrouver des ensembles de données sur la base de caractéristiques précises (mots clés ou motifs). Cela passe par la construction d'index ou d'index inversés. L'indexation permet d'optimiser l'interrogation des données dans le lac à travers un filtrage par mots clés. Elle est particulièrement utile pour la gestion de données textuelles, mais peut également servir dans un contexte de données semi-structurées ou structurées (Singh et al., 2016; Haste, 2017).

La **génération et conservation de liaisons (GL)** consiste pour le système de métadonnées à détecter des relations de similarité, ou à intégrer des liaisons préexistantes entre des ensembles de données. L'intégration des liaisons entre les données peut servir à élargir la panoplie d'analyses possibles à partir du lac par la recommandation de données connexes à celles intéressant l'utilisateur (Maccioni et Torlone, 2018). Les liaisons entre les données peuvent également servir à identifier des *clusters* de données, c'est-à-dire des regroupements de données fortement liées entre elles et différentes des autres (Farrugia et al., 2016).

Nous définissons le **polymorphisme des données (PD)** comme le fait d'avoir dans le système de métadonnées plusieurs représentations de la même donnée. Chaque représentation correspond alors à la même donnée modifiée ou reformatée pour un besoin spécifique. Un document textuel peut par exemple être représenté sans *stopwords*, sous forme de sac de mots, etc. Il est indispensable dans le contexte des lacs de données de structurer au moins partiellement les données non structurées afin de permettre leur analyse automatisée (Diamantini et al., 2018). Ainsi, nous considérons comme Stefanowski et al. (2017) qu'il est important d'avoir des modèles de métadonnées permettant de conserver simultanément plusieurs représentations des mêmes données afin d'éviter la répétition de certains prétraitements, et donc d'optimiser les analyses depuis le lac.

La fonctionnalité de **versionnement des données (VD)** désigne la faculté du système de métadonnées à prendre en charge les évolutions des données tout en conservant leurs états précédents. Autrement dit, le versionnement consiste à conserver plusieurs états du même ensemble de données dans le temps. Cette faculté est primordiale dans le contexte des lacs de données, car elle permet d'assurer la reproductibilité des analyses et de supporter la détection et la correction d'éventuelles erreurs ou incohérences. Le versionnement permet également de prendre en charge une évolution ramifiée des données, notamment dans leur schéma (Hellerstein et al., 2017).

Le **suivi d'utilisation (SU)** trace les interactions entre les utilisateurs du lac et les données. Ces interactions sont généralement des opérations de création, de modification ou de lecture des données. L'intégration de ces informations dans le système de métadonnées permet de comprendre et d'expliquer d'éventuelles incohérences dans les données (Beheshti et al., 2017). Elles peuvent également servir à la gestion de données sensibles, par la détection d'intrusions (Suriarachchi et Plale, 2016).

Le suivi d'utilisation et le versionnement des données sont étroitement liés puisque les interactions induisent dans certains cas la création de nouvelles versions ou représentations des données. Cependant, ces fonctionnalités ne sont pas pour autant systématiquement proposées ensemble (Suriarachchi et Plale, 2016; Beheshti et al., 2017; Diamantini et al., 2018).

## 3.2 Comparaison de systèmes de métadonnées

Nous proposons dans cette partie une comparaison de plusieurs systèmes de métadonnées sur la base des fonctionnalités identifiées dans la Section 3.1.

Les systèmes de métadonnées considérés dans cette comparaison sont relatifs à deux types de travaux : les modèles de métadonnées et les implémentations de lacs de données.

L'intitulé « modèles de métadonnées » désigne des systèmes conceptuels d'organisation des métadonnées. Ils ont l'avantage d'être plus détaillés et plus facilement reproductibles que les implémentations de lacs de données, qui se situent elles à un niveau plus opérationnel.

Ces dernières sont des exemples de mise en œuvre de lacs de données pour lesquels le fonctionnement et les fonctionnalités résultants sont décrits, avec peu de détails sur l'organisation conceptuelle des métadonnées.

Nous incluons dans cette étude des systèmes (modèles ou implémentations) non explicitement associés au concept de lacs de données par leurs auteurs, mais dont les caractéristiques permettent de les y assimiler. C'est notamment le cas du modèle de métadonnées Ground (Hellerstein et al., 2017), qui peut très bien servir à organiser le système de métadonnées d'un lac de données.

La comparaison de 15 systèmes de métadonnées de lacs de données (et assimilés) est présentée dans le Tableau 1. Ce tableau montre que les systèmes les plus complets en termes de fonctionnalités sont les lacs de données GOODS et CoreKG, avec cinq fonctionnalités proposées sur six. Ces systèmes se distinguent des autres par la prise en charge du polymorphisme et du versionnement des données. Il convient toutefois de noter que ces deux systèmes de métadonnées sont des « boites noires » présentant peu de détails sur l'organisation conceptuelle des données. Le modèle de données Ground peut donc leur être préféré car il est beaucoup plus détaillé et presqu'aussi complet (4/6).

Sur le plan des fonctionnalités, nous constatons une quasi-unanimité sur la pertinence de l'enrichissement sémantique avec 12 systèmes sur 15 proposant cette fonctionnalité et, dans une moindre mesure, les fonctionnalités d'indexation des données (9/15) et de génération de liaisons entre les données (8/15). En revanche, d'autres fonctionnalités sont beaucoup moins partagées, en particulier le polymorphisme (4/15) et le versionnement (3/15) des données. À notre sens, cette rareté ne dénote pas pour autant un manque de pertinence, mais plutôt une complexité de mise en œuvre. En effet, ces fonctionnalités se trouvent principalement dans les systèmes les plus complets (GOODS, CoreKG et Ground), et peuvent donc être considérées comme avancées, à l'inverse des autres fonctionnalités qui sont plus usuelles.

# 4 Typologie des métadonnées

Les résultats de la comparaison réalisée dans la Section 3.2 entre différents systèmes de métadonnées de lacs de données montrent qu'aucun d'entre eux ne propose l'ensemble des fonctionnalités attendues. En effet, les systèmes les plus complets (GOODS et CoreKG) proposent seulement 5 des 6 fonctionnalités clés attendues. Pour répondre à ce défi, nous proposons dans la suite de cet article un modèle de métadonnées supportant l'ensemble des six fonctionnalités clés identifiées.

Pour ce faire, il est primordial de définir un concept générique représentant tout ensemble de données homogènes que le modèle doit traiter. Certains travaux sur les lacs de données ont proposé les concepts d'unité de données (Quix et al., 2016), d'entité (Beheshti et al., 2017), de jeu de données (*dataset*) (Maccioni et Torlone, 2018) et d'objet (Diamantini et al., 2018). Nous adoptons la notion d'objet, qui nous semble plus appropriée pour représenter de façon abstraite un ensemble de données. Plus concrètement, un objet peut se matérialiser par une table relationnelle ou un fichier physique (document de tableur, XML ou JSON, document textuel, collection de tweets, image, vidéo, etc.).

La définition d'un modèle de métadonnées pour les lacs de données passe également par l'identification des métadonnées à considérer. À cet effet, nous proposons dans la suite de cette

| Système | Type | ES | ID | GL | PD | VD | SU |
|---|---|---|---|---|---|---|---|
| SPAR (Fauduet et Peyrard, 2010) | ♦♯ | ✓ | ✓ | ✓ | | | ✓ |
| Alrehamy et Walker (2015) | ♦ | ✓ | | ✓ | | | |
| Terrizzano et al. (2015) | ♦ | ✓ | ✓ | | | ✓ | ✓ |
| Constance (Hai et al., 2016) | ♦ | ✓ | ✓ | | | | |
| GEMMS (Quix et al., 2016) | ♢ | ✓ | | | | | |
| CLAMS (Farid et al., 2016) | ♦ | ✓ | | | | | |
| Suriarachchi et Plale (2016) | ♢ | | | | ✓ | | ✓ |
| Singh et al. (2016) | ♦ | ✓ | ✓ | ✓ | ✓ | | |
| Farrugia et al. (2016) | ♦ | | | ✓ | | | |
| GOODS (Halevy et al., 2016) | ♦ | ✓ | ✓ | ✓ | | ✓ | ✓ |
| CoreDB (Beheshti et al., 2017) | ♦ | | ✓ | | | | ✓ |
| Ground (Hellerstein et al., 2017) | ♢♯ | ✓ | ✓ | | | ✓ | ✓ |
| KAYAK (Maccioni et Torlone, 2018) | ♦ | ✓ | ✓ | ✓ | | | |
| CoreKG (Beheshti et al., 2018) | ♦ | ✓ | ✓ | ✓ | ✓ | | ✓ |
| Diamantini et al. (2018) | ♢ | ✓ | | ✓ | ✓ | | |

♦ : Implémentation de lac de données ♢ : Modèle de métadonnées
♯ : Modèle ou implémentation assimilable à un lac de données

TAB. 1 – *Fonctionnalités proposées par les systèmes de métadonnées de lacs de données.*

section une typologie des métadonnées d'un lac de données. Cette typologie, qui catégorise les métadonnées en métadonnées intra-objet, inter-objets et globales, est une extension de celle proposée par Sawadogo et al. (2019), que nous complétons en prenant en compte de nouveaux types de métadonnées inter-objets (liaisons de parenté) et globales (index, journaux d'événements).

## 4.1 Métadonnées intra-objet

Cette catégorie désigne des métadonnées associées à un objet précis. Nous en distinguons plusieurs types (Sawadogo et al., 2019).

Les **propriétés** fournissent une description générale de l'objet, sous la forme de couples clé-valeur. Ces métadonnées sont généralement obtenues à partir du système de fichiers : titre de l'objet, taille, date de dernière modification, chemin d'accès, etc.

Les **résumés et prévisualisations** ont pour rôle de donner un aperçu du contenu ou de la structure d'un objet. Elles peuvent prendre la forme d'un schéma des données dans un contexte de données structurées ou semi-structurées, ou d'un nuage de mots pour des données textuelles.

Les données brutes dans le lac sont souvent amenées à être modifiées à travers des mises à jour. De telles opérations entraînent la création de nouvelles **versions** des données initiales, qui peuvent être considérées comme des métadonnées. De même, les données brutes (surtout non structurées) peuvent être reformatées pour un usage spécifique. Cette opération induit la création d'une nouvelle **représentation** de l'objet.

Les **métadonnées sémantiques** sont des annotations qui permettent de comprendre le sens des données. Il s'agit plus concrètement de *tags* descriptifs, de descriptions textuelles ou de catégorisations métier. Les métadonnées sémantiques servent souvent de base à la détection de relations entre les données du lac.

## 4.2 Métadonnées inter-objets

Les métadonnées inter-objets traduisent les liaisons entre les objets. Chacune de ces métadonnées est donc associée à au moins deux objets. Nous en distinguons trois types : les groupements d'objets, les liaisons de similarité et les relations de parenté.

Les **groupements d'objets** consistent à organiser les objets du lac en collections, chaque objet pouvant appartenir simultanément à plusieurs collections. Ces groupements peuvent être déduits automatiquement des métadonnées sémantiques telles que des *tags* et des catégories métier. Certaines propriétés peuvent également servir de base à la génération des groupements ; les objets peuvent ainsi être regroupés suivant leur format ou leur langue d'édition.

Les **relations de similarité** traduisent la force de la ressemblance entre deux objets. À l'inverse des groupements d'objets, les relations de similarité portent sur les propriétés intrinsèques des objets, notamment leur contenu ou leur structure. Par exemple, il peut s'agir du taux de mots en commun entre deux documents textuels, d'une mesure de la compatibilité des schémas de deux objets structurés ou semi-structurés (Maccioni et Torlone, 2018), ou d'autres mesures de similarité usuelles.

Les **relations de parenté**, que nous ajoutons à la typologie de Sawadogo et al. (2019), traduisent le fait qu'un objet peut être issu de la jointure de plusieurs autres. Dans un tel cas, il existe une relation de « parenté » entre les objets combinés et l'objet résultant, et une relation de « co-parenté » entre les objets fusionnés. Ce type de relation permet ainsi de tirer parti des traitements effectués dans le lac de données pour identifier des objets utilisables conjointement, en plus de conserver une traçabilité de la provenance des objets générés à l'intérieur du lac.

## 4.3 Métadonnées globales

Les métadonnées globales sont des structures de données destinées à donner une couche de contexte aux données du lac en vue de faciliter et d'optimiser leur analyse. Contrairement aux métadonnées intra et inter, les métadonnées globales concernent potentiellement l'ensemble du lac de données. En plus des ressources sémantiques identifiées par Sawadogo et al. (2019), nous proposons deux nouveaux types de métadonnées globales.

Les **ressources sémantiques** sont essentiellement des bases de connaissances (ontologies, taxonomies, thésauri, dictionnaires) utilisées à la fois pour générer d'autres métadonnées et améliorer les analyses. L'utilisation d'un thésaurus permet ainsi d'étendre une requête par mots-clés en associant des synonymes des termes saisis par l'utilisateur. De même, un thésaurus peut servir lors de la génération de groupements de données, de fusionner des collections issues de *tags* différents mais équivalents.

Les ressources sémantiques sont généralement issues de sources externes. C'est typiquement le cas des ontologies qui sont fournies par des bases de connaissances sur Internet. Notons toutefois que dans certains cas, les ressources sémantiques peuvent être constituées et personnalisées spécialement pour la gestion et l'analyse des données du lac. Une ontologie métier

peut ainsi servir à définir des *tags* abstraits permettant de regrouper lors de l'analyse plusieurs *tags* équivalents ou proches.

Les **index et index inversés** sont des structures de données permettant de retrouver rapidement un objet sur la base de caractéristiques précises. Elles établissent (ou mesurent) la correspondance entre ces caractéristiques (mots clés, motifs, couleurs) et les objets contenus dans le lac de données. Les index peuvent être simples (indexation textuelle) ou plus complexes (sur le contenu d'images, de sons, etc.). Ils servent principalement à la recherche de données dans le lac.

Couramment appelés *logs*, les **journaux d'événements** permettent de tracer les interactions entre les utilisateurs et le lac de données. Cela passe par l'enregistrement séquentiel d'événements comme la connexion d'un utilisateur, la modification ou la consultation d'un objet dans un fichier ou une base de données. Ces métadonnées permettent d'analyser l'utilisation du lac de données par l'identification des objets les plus consultés ou par l'étude des comportements des utilisateurs.

### 4.4 Définition formelle d'un lac de données

À partir de la typologie présentée ci-dessus, nous pouvons définir un lac de données.

**Définition 2** *Un lac de données est un doublet $DL = \langle \mathcal{D}, \mathcal{M} \rangle$, où $\mathcal{D}$ est un ensemble de données brutes (objets) et $\mathcal{M}$ un ensemble de métadonnées décrivant les objets de $\mathcal{D}$. Les objets de $\mathcal{D}$ peuvent prendre la forme de données structurées (tables de bases de données relationnelles, fichiers CSV, etc.), semi-structurées (documents JSON, XML, YAML, etc.) et non structurées (images, documents textuels, vidéos, etc.). Les métadonnées sont subdivisées en trois composantes : $\mathcal{M} = \langle \mathcal{M}_{intra}, \mathcal{M}_{inter}, \mathcal{M}_{glob} \rangle$, où $\mathcal{M}_{intra}$ est l'ensemble de métadonnées intra-objet, $\mathcal{M}_{inter}$ l'ensemble de métadonnées inter-objets et $\mathcal{M}_{glob}$ l'ensemble des métadonnées globales.*

## 5 Modélisation du système de métadonnées

Nous avons présenté dans la Section 4 une typologie des métadonnées qu'un lac de données doit prendre en charge de notre point de vue. Dans cette section, nous nous appuyons sur cette typologie ainsi que sur le concept d'« objet » développé précédemment pour introduire un nouveau modèle de métadonnées nommé MEDAL (*MEtadata model for DAta Lakes*).

D'un point de vue logique, MEDAL adopte une représentation des métadonnées à base de graphes. Les graphes présentent en effet le double avantage d'offrir un schéma flexible et de faciliter l'expression de relations. Ainsi, nous représentons un objet par un **hypernœud**, qui est un nœud pouvant contenir d'autres nœuds (graphe imbriqué). Ces hypernœuds contiennent divers éléments (versions et représentations, propriétés, etc.), et peuvent être liés entre eux (similarité, parenté, etc.).

### 5.1 Métadonnées intra-objet

Chaque hypernœud contient des **représentations**, qui traduisent le fait que les données associées à l'objet peuvent être présentées de différentes manières. Il existe *a minima* une représentation par hypernœud, qui est la donnée brute ingérée dans le lac de données. Les autres

représentations sont toutes issues de cette représentation brute initiale. Chaque représentation correspond à un nœud qui possède des attributs, simples ou complexes. Ceux-ci sont les propriétés de la représentation. Notons qu'une représentation peut être associée à un ensemble de données effectivement stocké dans le lac ou être une vue calculée à la demande.

Le passage d'une représentation à une autre se fait via une **transformation**. Elle prend la forme d'un arc dirigé reliant deux nœuds de représentation. Cet arc possède aussi des attributs, qui sont les propriétés décrivant le processus de transformation ayant permis de passer de la première représentation à la seconde. Dans l'idéal, la transformation conserve le script qui a servi à cette opération, bien que dans certains cas la transformation soit manuelle ; il faut alors que l'utilisateur ayant fait cette transformation saisisse une description pour assurer une meilleure compréhension de son travail par d'autres utilisateurs.

Un hypernœud peut aussi contenir des **versions**, qui servent à gérer les évolutions des données présentes dans le lac à travers le temps. Nous associons également les version à des nœuds, comme les représentations, possédant eux aussi des attributs pour y stocker leurs propriétés. La création d'un nouveau nœud de version n'est pas forcément systématique au moindre changement. Selon la nature et la fréquence d'évolution des données, il est possible de mettre en place diverses stratégies pour gérer ces évolutions, à l'image des dimensions à évolution lente des entrepôts de données (Kimball, 2008). Les versions sont les représentations des ensembles de données brutes. La première version est la représentation brute initiale. La création d'une nouvelle version se fait via une **mise à jour** semblable à une transformation, puisqu'elle est aussi traduite par un arc dirigé et possède des attributs.

Enfin, comme les nœuds de représentation et de version, l'hypernœud est porteur d'attributs, qui permettent de le décrire. Ces attributs peuvent être des propriétés comme la provenance de l'ensemble de données, ou bien des agrégats des attributs des représentations et versions qu'il contient, par exemple le nombre de versions, de représentations, la taille cumulée, etc.

Ainsi, un hypernœud contient un arbre dont les nœuds sont des représentations ou des versions et les arcs dirigés sont des transformations ou des mises à jour. Une représentation (resp. version) est issue d'une autre par une transformation (resp. mise à jour). Une version peut donner lieu à une représentation via une transformation, mais une version ne peut pas être issue d'une représentation. Ainsi, la racine de l'arbre est la représentation brute initiale de l'hypernœud et chaque version possède son propre sous-arbre de représentations.

**Définition 3** *Soit $\mathcal{N}$ un ensemble de nœuds. L'ensemble des* métadonnées intra $\mathcal{M}_{intra}$ *est l'ensemble des hypernœuds tel que* $\forall h \in \mathcal{M}_{intra}, h = \langle N, E \rangle$*, où :*

- $N \subset \mathcal{N}$ *est l'ensemble des nœuds (représentations et versions) porteurs d'attributs de* $h$*;*
- $E = \{r_{(transformation \mid mise\ à\ jour)} \in N \times N\}$ *est l'ensemble des arcs (transformations et mises à jour) porteurs d'attributs de* $h$*.*

Nous illustrons ces notions à travers un exemple (Figure 1). Imaginons une entreprise vendant divers produits. Les informations sur ces produits (nom, prix unitaire, description, etc.) sont stockées dans le lac sous la forme d'un fichier XML. Un hypernœud décrit cet ensemble de données et il possède un nœud de version qui correspond au fichier XML tel qu'initialement ingéré dans le lac. Afin d'assister le requêtage des informations sur les produits, un utilisateur

décide d'extraire le schéma du fichier XML. Ceci génère une nouvelle représentation issue de la version initiale. Supposons maintenant que les données évoluent, car le prix de certains produits a changé, et que de nouveaux produits ont été ajoutés au catalogue. Ce changement dans les données génère une nouvelle version, liée à la première version par une mise à jour. Enfin, si l'utilisateur souhaite obtenir le schéma des données plus récentes, cela crée alors une nouvelle représentation issue de la deuxième version.

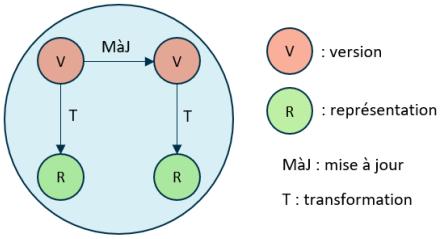


FIG. 1 – *Hypernœud et son arbre de représentations.*

## 5.2 Métadonnées inter-objets

Un regroupement d'objets est modélisé par un ensemble d'hyperarcs non orientés, c'est-à-dire des arcs pouvant lier plus de deux nœuds (en l'occurrence, des hypernœuds). Chaque hyperarc correspond à une collection d'objets. Si le regroupement est effectué sur un attribut d'hypernœud, un hypernœud appartient à l'hyperarc qui correspond à sa valeur pour l'attribut. Il existe donc autant d'hyperarcs que de valeurs distinctes pour l'attribut considéré. Notons que tous les attributs ne servent pas forcément à faire des regroupements et que des regroupements peuvent être effectués sur d'autres éléments que les attributs.

Une liaison de similarité entre deux hypernœuds est représentée par un arc non orienté porteur d'attributs : valeur de la mesure de similarité, type de mesure utilisée, date de la mesure, etc. Pour que deux hypernœuds soient connectés par une liaison de similarité, ils doivent être comparables, c'est-à-dire qu'ils doivent contenir chacun une représentation qui peut être comparée à l'autre grâce à une mesure de similarité.

Un hypernœud peut être issu d'autres hypernœuds à travers un lien de parenté. Pour traduire cette relation, nous avons recours à un hyperarc orienté : l'ensemble des hypernœuds « parents » et l'hypernœud « enfant » sont reliés par cet hyperarc orienté vers l'hypernœud enfant. Une fois encore, cet hyperarc possède des attributs descriptifs.

**Définition 4** *L'ensemble des* métadonnées inter $\mathcal{M}_{inter}$ *est défini par les trois couples* $\langle H, E_g \rangle$, $\langle H', E_s \rangle$ *et* $\langle H'', E_p \rangle$ *tels que :*

- $H \subset \mathcal{M}_{intra}$, $H' \subset \mathcal{M}_{intra}$ *et* $H'' \subset \mathcal{M}_{intra}$ *sont des ensembles d'hypernœuds porteurs d'attributs ;*
- $E_g = \{E_g^{param} \mid E_g^{param} : H \rightarrow \mathcal{P}(H)\}$ *est l'ensemble des fonctions regroupant les hypernœuds dans des collections selon un paramètre donné (souvent, un attribut) ;*
- $E_s = \{s \mid s \in H' \times H'\}$ *est l'ensemble des arcs (liaisons de similarité) porteurs d'attributs ;*

- $E_p = \{(h_1, ..., h_n, h_{enfant}) \mid (h_1, ..., h_n, h_{enfant}) \in (H'')^{n+1}\}$ *est l'ensemble des liaisons de parenté, où* $(h_1, ..., h_n)$ *sont les hypernœuds parents* $(n \geq 2)$ *et* $h_{enfant}$ *l'hypernœud enfant.*

Poursuivons l'exemple de la Section 5.1 en ajoutant d'autres hypernœuds : des tweets en rapport avec l'entreprise récoltés sur internet, ainsi qu'une vidéo commerciale des produits vendus. Dans un regroupement sur la provenance des données, l'hypernœud des tweets est seul dans la collection « source externe », tandis que les deux autres hypernœuds sont dans la collection « source interne ». Dans un second regroupement sur le format de la version initiale, c'est l'hypernœud de la vidéo qui est seul dans la collection « non structuré », alors que les deux autres hypernœuds sont dans la collection « semi-structuré ». Les collections sont représentées par des rectangles en pointillés dans la Figure 2 (les attributs des hypernœuds ne sont pas représentés).

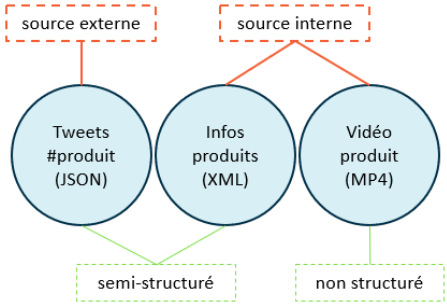


FIG. 2 – *Hypernœuds interconnectés.*

## 5.3 Métadonnées globales

Les métadonnées globales sont des éléments particuliers qui sont gérés différemment des autres données. Ils « gravitent » autour des hypernœuds et sont exploités dès que nécessaire, c'est-à-dire presque systématiquement, notamment pour les *logs* et les index. Nous pouvons considérer que les ressources sémantiques sont stockées dans des nœuds, tandis que les index et les journaux d'événements sont plutôt des structures physiques et sont grandement dépendants de la technologie employée pour implémenter le lac de données et le système de métadonnées.

# 6 Conclusion

Après un tour d'horizon des différentes définitions d'un lac de données de la littérature, nous proposons dans cet article notre propre définition de ce concept. Nous identifions ensuite ce que nous considérons être les fonctionnalités clés (au nombre de six) que le système de métadonnées d'un lac de données doit proposer pour être le plus robuste possible face aux problèmes inhérents aux mégadonnées et à l'approche *schema-on-read*. Une étude des systèmes de métadonnées existants montre que certains réussissent à proposer presque toutes les fonctionnalités, mais qu'aucun n'en propose l'intégralité.

C'est pourquoi nous proposons un nouveau modèle de métadonnées, MEDAL, qui s'appuie sur la notion d'objet et une typologie des métadonnées en trois grandes catégories : les métadonnées intra-objet, inter-objets et globales. Les métadonnées intra-objet sont associées à chaque ensemble de données homogènes sous la forme de prévisualisations, de versions, de représentations, de métadonnées sémantiques ou de propriétés. Les métadonnées inter-objets permettent elles de lier les objets entre eux, notamment au sein de collections ou à travers des liaisons de similarité ou de parenté. Enfin, les métadonnées globales servent à faciliter et améliorer les analyses des données ainsi que l'utilisation du lac de données de manière générale.

MEDAL organise les métadonnées sous forme de graphes. Un objet est représenté par un hypernœud contenant des nœuds qui correspondent aux versions et représentations de l'objet. Les opérations de transformation et de mise à jour sont modélisées par des arcs orientés reliant les nœuds. Les hypernœuds peuvent être liés de plusieurs manières : des arcs pour modéliser les liaisons de similarité et des hyperarcs pour traduire les groupements et liaisons de parenté. Enfin, les ressources globales sont aussi présentes, sous la forme de bases de connaissances, d'index ou encore de journaux d'événements ; nous nous intéressons peu à leur représentation dans notre modèle de métadonnées car elles se situent à un niveau plus opérationnel et dépendent fortement du support technologique utilisé.

Grâce à tous ces éléments, MEDAL supporte l'ensemble des six fonctionnalités clés que nous avons identifiées, faisant de lui le modèle de métadonnées le plus complet à notre connaissance. Toutefois, nous n'avons pas encore implémenté MEDAL. Cela fera l'objet de futurs travaux au cours desquels nous pourrons proposer une application de notre modèle de métadonnées dans un contexte de données structurées, semi-structurées et non structurées. Cette implémentation nous permettra d'évaluer MEDAL de manière plus détaillée, notamment en le comparant aux autres systèmes existants.

## Remerciements

Une partie des recherches présentées dans cet article est subventionnée par la Région Auvergne-Rhône-Alpes, dans le cadre du projet AURA-PMI qui finance la thèse de Pegdwendé N. Sawadogo.

## Références

Alrehamy, H. et C. Walker (2015). Personal Data Lake With Data Gravity Pull. In *IEEE 5th International Conference on Big Data and Cloud Computing(BDCloud 2015), Dalian, china*, Volume 88 of *IEEE Computer Society Washington*, pp. 160–167.

Ansari, J. W., N. Karim, S. Decker, M. Cochez, et O. Beyan (2018). Extending Data Lake Metadata Management by Semantic Profiling. In *2018 Extended Semantic Web Conference (ESWC 2018), Heraklion, Crete, Greece*, ESWC, pp. 1–15.

Beheshti, A., B. Benatallah, R. Nouri, V. M. Chhieng, H. Xiong, et X. Zhao (2017). CoreDB : a Data Lake Service. In *2017 ACM on Conference on Information and Knowledge Management (CIKM 2017), Singapore, Singapore*, ACM, pp. 2451–2454.

Beheshti, A., B. Benatallah, R. Nouri, et A. Tabebordbar (2018). CoreKG : A Knowledge Lake Service. *Proceedings of the VLDB Endowment 11*(12), 1942–1945.

Diamantini, C., P. L. Giudice, L. Musarella, D. Potena, E. Storti, et D. Ursino (2018). A New Metadata Model to Uniformly Handle Heterogeneous Data Lake Sources. In *New Trends in Databases and Information Systems - ADBIS 2018 Short Papers and Workshop, Budapest, Hungary*, pp. 165–177.

Dixon, J. (2010). Pentaho, Hadoop, and Data Lakes. https ://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/.

Fang, H. (2015). Managing Data Lakes in Big Data Era : What's a data lake and why has it became popular in data management ecosystem. In *5th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems (CYBER 2015), Shenyang, China*, IEEE, pp. 820–824.

Farid, M., A. Roatis, I. F. Ilyas, H.-F. Hoffmann, et X. Chu (2016). CLAMS : Bringing Quality to Data Lakes. In *2016 International Conference on Management of Data (SIGMOD 2016), San Francisco, CA, USA*, ACM, pp. 2089–2092.

Farrugia, A., R. Claxton, et S. Thompson (2016). Towards Social Network Analytics for Understanding and Managing Enterprise Data Lakes. In *Advances in Social Networks Analysis and Mining (ASONAM 2016), San Francisco, CA, USA*, IEEE, pp. 1213–1220.

Fauduet, L. et S. Peyrard (2010). A Data-First Preservation Strategy : Data Management In SPAR. In *7th International Conference on Preservation of Digital Objects (iPRES 2010), Vienna, Autria*, pp. 1–8.

Hai, R., S. Geisler, et C. Quix (2016). Constance : An Intelligent Data Lake System. In *2016 International Conference on Management of Data (SIGMOD 2016) ,San Francisco, CA, USA*, ACM Digital Library, pp. 2097–2100.

Halevy, A., F. Korn, N. F. Noy, C. Olston, N. Polyzotis, S. Roy, et S. E. Whang (2016). Managing Google's data lake : an overview of the GOODS system. In *2016 International Conference on Management of Data (SIGMOD 2016), San Francisco, CA, USA*, ACM, pp. 795–806.

Haste, J.-L. (2017). From the data lake to the agile data warehouse : decision-making in the big data era . https ://en.blog.businessdecision.com/bigdata-en/2017/02/data-lake-data-warehouse-big-data-era/.

Hellerstein, J. M., V. Sreekanti, J. E. Gonzalez, J. Dalton, A. Dey, S. Nag, K. Ramachandran, S. Arora, A. Bhattacharyya, S. Das, M. Donsky, G. Fierro, C. She, C. Steinbach, V. Subramanian, et E. Sun (2017). Ground : A Data Context Service. In *8th Biennial Conference on Innovative Data Systems Research (CIDR 2017), Chaminade, CA, USA*.

Khine, P. P. et Z. S. Wang (2017). Data Lake : A New Ideology in Big Data Era. In *4th International Conference on Wireless Communication and Sensor Network (WCSN 2017), Wuhan, China*, Volume 17 of *ITM Web of Conferences*, pp. 1–6.

Kimball, R. (2008). Slowly changing dimensions. *Information Management 18*(9), 29.

Laskowski, N. (2016). Data lake governance : A big data do or die. https ://searchcio.techtarget.com/feature/Data-lake-governance-A-big-data-do-or-die.

Maccioni, A. et R. Torlone (2018). KAYAK : A Framework for Just-in-Time Data Preparation

in a Data Lake. In *International Conference on Advanced Information Systems Engineering (CAiSE 2018),Tallin, Estonia*, pp. 474–489.

Madera, C. et A. Laurent (2016). The next information architecture evolution : the data lake wave. In *8th International Conference on Management of Digital EcoSystems (MEDES 2016), Biarritz, France*, pp. 174–180.

Mathis, C. (2017). Data Lakes. *Datenbank-Spektrum 17*(3), 289–293.

Miloslavskaya, N. et A. Tolstoy (2016). Big Data, Fast Data and Data Lake Concepts. In *7th Annual International Conference on Biologically Inspired Cognitive Architectures (BICA 2016), NY, USA*, Volume 88 of *Procedia Computer Science*, pp. 1–6.

O'Leary, D. E. (2014). Embedding AI and Crowdsourcing in the Big Data Lake. *IEEE Intelligent Systems 29*(5), 70–73.

Quix, C., R. Hai, et I. Vatov (2016). GEMMS : A Generic and Extensible Metadata Management System for Data Lakes. In *Forum at the 28th International Conference on Advanced Information Systems Engineering (CAiSE 2016), Ljubljana, Slovenia*, pp. 129–136.

Sawadogo, P. N., T. Kibata, et J. Darmont (2019). Metadata Management for Textual Documents in Data Lakes. In *21st International Conference on Enterprise Information Systems (ICEIS 2019), Heraklion, Crete, Greece*.

Singh, K., K. Paneri, A. Pandey, G. Gupta, G. Sharma, P. Agarwal, et G. Shroff (2016). Visual Bayesian Fusion to Navigate a Data Lake. In *19th International Conference on Information Fusion (FUSION 2016), Heidelberg, Germany*, IEEE, pp. 987–994.

Sirosh, J. (2016). The Intelligent Data Lake. https ://azure.microsoft.com/fr-fr/blog/the-intelligent-data-lake/.

Stefanowski, J., K. Krawiec, et R. Wrembel (2017). Exploring Complex and Big Data. *International Journal of Applied Mathematics and Computer Science 27*(4), 669–679.

Suriarachchi, I. et B. Plale (2016). Crossing Analytics Systems : A Case for Integrated Provenance in Data Lakes. In *12th IEEE International Conference on e-Science (e-Science 2016), Baltimore, MD, USA*, pp. 349–354.

Terrizzano, I., P. Schwarz, M. Roth, et J. E. Colino (2015). Data Wrangling : The Challenging Journey from the Wild to the Lake. In *7th Biennial Conference on Innovative Data Systems Research (CIDR 2015), Asilomar, CA, USA*, pp. 1–9.

## Summary

Over the past decade, the data lake concept has emerged as an alternative to data warehouses for storing and analyzing big data. A data lake allows storing data without any predefined schema. Therefore, data querying and analysis depends on a metadata system that must be efficient and comprehensive. However, metadata management in data lakes remains a current issue and the criteria for evaluating its effectiveness are more or less inexistent.

In this article, we propose MEDAL, a generic model for metadata management in data lakes. We adopt a graph-based model for MEDAL. We also propose evaluation criteria for data lake metadata systems through a list of expected features. Eventually, we show that our approach is more comprehensive than existing metadata systems.