

Plan

- 1 Introduction
- 2 Solutions existantes
- 3 Architecture
- 4 Technologies
- 5 Réalisation
- 6 Discussion
- 7 Conclusion

h e p i a

Haute école du paysage, d'ingénierie
et d'architecture de Genève

Problématiques

- Nombre de fichiers énorme.

Problématiques

- Nombre de fichiers énorme.
- Difficulté à retrouver des fichiers.

Problématiques

- Nombre de fichiers énorme.
- Difficulté à retrouver des fichiers.
- Plusieurs emplacements logiques pour un seul fichier.

Problématiques

- Nombre de fichiers énorme.
- Difficulté à retrouver des fichiers.
- Plusieurs emplacements logiques pour un seul fichier.

Système de "tagging" de fichiers et répertoires avec possibilité de recherche par tags.

Cahier des charges

- Répertorier les applications existantes permettant d'étiqueter les fichiers.

Cahier des charges

- Répertorier les applications existantes permettant d'étiqueter les fichiers.
- Étudier les attributs étendus (XATTR) lors des manipulation courantes sur les fichiers.

Cahier des charges

- Répertorier les applications existantes permettant d'étiqueter les fichiers.
- Étudier les attributs étendus (XATTR) lors des manipulation courantes sur les fichiers.
- Analyser les moyens d'indexer et de surveiller une arborescence de fichiers.

Cahier des charges

- Répertorier les applications existantes permettant d'étiqueter les fichiers.
- Étudier les attributs étendus (XATTR) lors des manipulation courantes sur les fichiers.
- Analyser les moyens d'indexer et de surveiller une arborescence de fichiers.
- Concevoir et implémenter le système (open source et sur Linux) et mesurer ses performances.

Cahier des charges

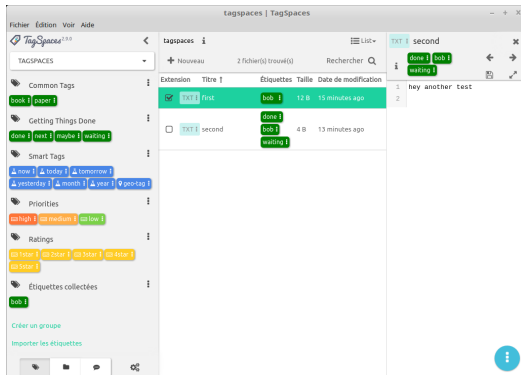
- Répertorier les applications existantes permettant d'étiqueter les fichiers.
- Étudier les attributs étendus (XATTR) lors des manipulation courantes sur les fichiers.
- Analyser les moyens d'indexer et de surveiller une arborescence de fichiers.
- Concevoir et implémenter le système (open source et sur Linux) et mesurer ses performances.
- Étudier et s'appropriier le langage Rust.

TMSU, Tagsistant et TagSpaces

- Gestion des tags.
- Liste de fichiers liés aux tags.
- CLI ou GUI.

TMSU, Tagsistant et TagSpaces

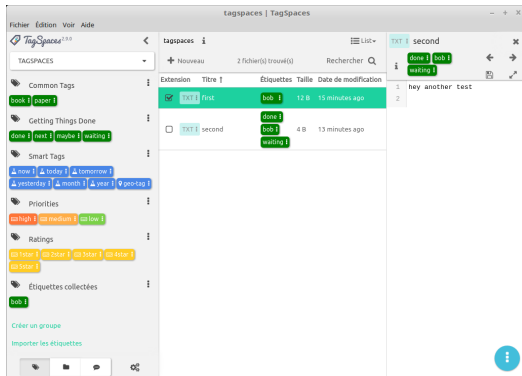
- Gestion des tags.
- Liste de fichiers liés aux tags.
- CLI ou GUI.



TMSU, Tagsistant et TagSpaces

- Gestion des tags.
- Liste de fichiers liés aux tags.
- CLI ou GUI.

Points positifs	Points négatifs
Simple	Dépendance à une BDD externe
Rapides et efficaces	Modification et accès uniquement par l'app
Open source	



macOS

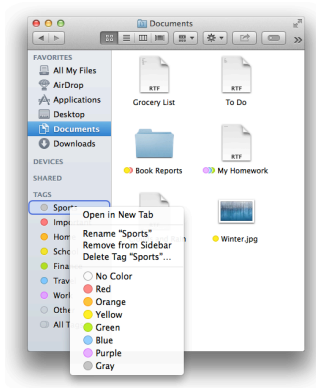
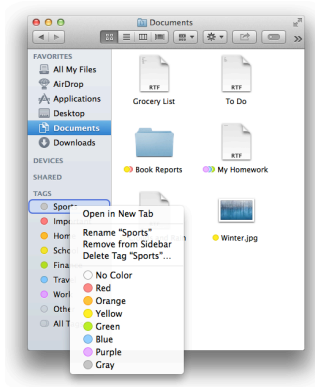


Figure – Gestion d'un tag dans le Finder [1]

h e p i a

Haute école du paysage, d'ingénierie
et d'architecture de Genève

macOS



Points positifs	Points négatifs
Système de tags intégré à l'explorateur de fichiers	Code propriétaire
Stocke les tags dans les XATTR	Seulement pour macOS
Performant	

Figure – Gestion d'un tag dans le Finder [1]

h e p i a

Haute école du paysage, d'ingénierie
et d'architecture de Genève

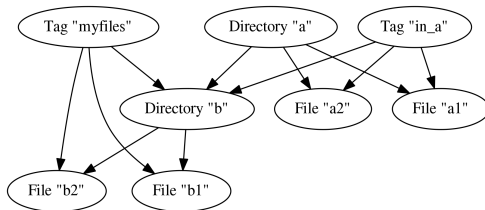
Gestion des tags

- Stockage des tags dans les attributs étendus (XATTR).

Gestion des tags

- Stockage des tags dans les attributs étendus (XATTR).
- Outil dédié plutôt que reprendre les commandes existantes
=> Confort d'utilisation.

Indexation des fichiers et des tags



Indexation des fichiers et des tags

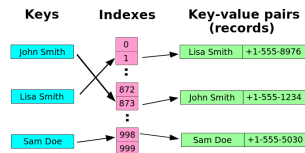
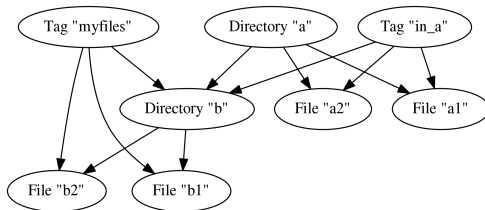


Figure – Un annuaire représenté comme une table de hachage - [2]

Surveillance du système de fichiers

Mise à jour du graphe lors des événements suivants :

Surveillance du système de fichiers

Mise à jour du graphe lors des événements suivants :

- Changement sur les tags.

Surveillance du système de fichiers

Mise à jour du graphe lors des événements suivants :

- Changement sur les tags.
- Création de fichiers/répertoires.

Surveillance du système de fichiers

Mise à jour du graphe lors des événements suivants :

- Changement sur les tags.
- Création de fichiers/répertoires.
- Suppression de fichiers/répertoires.

Surveillance du système de fichiers

Mise à jour du graphe lors des événements suivants :

- Changement sur les tags.
- Création de fichiers/répertoires.
- Suppression de fichiers/répertoires.
- Déplacement/renommage de fichiers/répertoires.

Requêtes de tags et fichiers

Requêtes de tags et fichiers

- Lister les fichiers et répertoires associés à des tags => requêtes sous forme d'expressions logiques.

Requêtes de tags et fichiers

- Lister les fichiers et répertoires associés à des tags => requêtes sous forme d'expressions logiques.
- Lister les tags existants.

Requêtes de tags et fichiers

- Lister les fichiers et répertoires associés à des tags => requêtes sous forme d'expressions logiques.
- Lister les tags existants.
- Renommer un tag.

Rust

Généralités

- Langage système moderne, performant, fiable (plus sécurisé qu'Ada), compilé, et fortement typé.

Rust

Généralités

- Langage système moderne, performant, fiable (plus sécurisé qu'Ada), compilé, et fortement typé.
- Disponible sur Linux, Windows et macOS.

Rust

Généralités

- Langage système moderne, performant, fiable (plus sécurisé qu'Ada), compilé, et fortement typé.
- Disponible sur Linux, Windows et macOS.
- Cargo : outil de compilation et d'exécution et gestionnaire de paquets intégré à Rust.

Rust

Généralités

- Langage système moderne, performant, fiable (plus sécurisé qu'Ada), compilé, et fortement typé.
- Disponible sur Linux, Windows et macOS.
- Cargo : outil de compilation et d'exécution et gestionnaire de paquets intégré à Rust.
- Structures, collections, généricité, immutabilité, énumérations et *pattern matching*.

Rust

Généralités

- Langage système moderne, performant, fiable (plus sécurisé qu'Ada), compilé, et fortement typé.
- Disponible sur Linux, Windows et macOS.
- Cargo : outil de compilation et d'exécution et gestionnaire de paquets intégré à Rust.
- Structures, collections, généricité, immutabilité, énumérations et *pattern matching*.
- Gestion des erreurs.

Rust

Généralités

- Langage système moderne, performant, fiable (plus sécurisé qu'Ada), compilé, et fortement typé.
- Disponible sur Linux, Windows et macOS.
- Cargo : outil de compilation et d'exécution et gestionnaire de paquets intégré à Rust.
- Structures, collections, généricité, immutabilité, énumérations et *pattern matching*.
- Gestion des erreurs.
- Tests.

Rust

Ownership, Borrowing

Ownership

Rust

Ownership, Borrowing

Ownership

- Chaque variable est dite le "possesseur" (*owner*) d'une valeur.

Rust

Ownership, Borrowing

Ownership

- Chaque variable est dite le "possesseur" (*owner*) d'une valeur.
- Il ne peut y avoir qu'un seul *owner* pour une valeur.

Rust

Ownership, Borrowing

Ownership

- Chaque variable est dite le "possesseur" (*owner*) d'une valeur.
- Il ne peut y avoir qu'un seul *owner* pour une valeur.
- Lorsque l'*owner* est détruit ou change de portée, la valeur est détruite.

Rust

Ownership, Borrowing

Ownership

- Chaque variable est dite le "possesseur" (*owner*) d'une valeur.
- Il ne peut y avoir qu'un seul *owner* pour une valeur.
- Lorsque l'*owner* est détruit ou change de portée, la valeur est détruite.

Borrowing

Rust

Ownership, Borrowing

Ownership

- Chaque variable est dite le "possesseur" (*owner*) d'une valeur.
- Il ne peut y avoir qu'un seul *owner* pour une valeur.
- Lorsque l'*owner* est détruit ou change de portée, la valeur est détruite.

Borrowing

- À tout moment, il ne peut exister qu'une seule référence mutable ou plusieurs références immutables, mais pas les deux en même temps.

Rust

Ownership, Borrowing

Ownership

- Chaque variable est dite le "possesseur" (*owner*) d'une valeur.
- Il ne peut y avoir qu'un seul *owner* pour une valeur.
- Lorsque l'*owner* est détruit ou change de portée, la valeur est détruite.

Borrowing

- À tout moment, il ne peut exister qu'une seule référence mutable ou plusieurs références immutables, mais pas les deux en même temps.
- Les références doivent toujours être valides.

Attributs étendus (XATTR)

- Métadonnée sous forme de paire nom:valeur.

Attributs étendus (XATTR)

- Métadonnée sous forme de paire `nom:valeur`.
- Nom = chaîne de caractères, valeur = chaîne de caractères ou données binaires.

Attributs étendus (XATTR)

- Métadonnée sous forme de paire `nom:valeur`.
- Nom = chaîne de caractères, valeur = chaîne de caractères ou données binaires.
- Existent sous ext2-3-4, XFS, Btrfs, UFS1-2, NTFS, HFS+, ZFS.

Attributs étendus (XATTR)

- Métadonnée sous forme de paire `nom:valeur`.
- Nom = chaîne de caractères, valeur = chaîne de caractères ou données binaires.
- Existent sous ext2-3-4, XFS, Btrfs, UFS1-2, NTFS, HFS+, ZFS.
- Outils CLI pour facilement les manipuler.

Inotify

- API de notifications d'événements sur le système de fichiers.

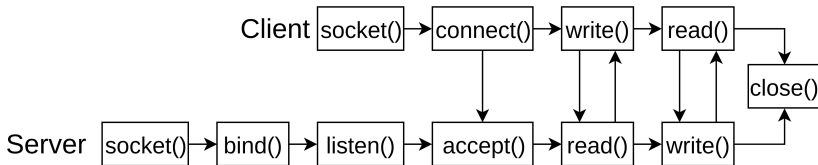
Inotify

- API de notifications d'événements sur le système de fichiers.
- Trois appels système : initialisation, ajout de surveillance sur un chemin de fichiers donné et suppression de cette surveillance.

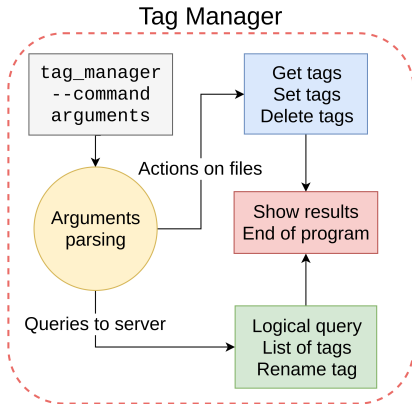
Inotify

- API de notifications d'événements sur le système de fichiers.
- Trois appels système : initialisation, ajout de surveillance sur un chemin de fichiers donné et suppression de cette surveillance.
- Lecture d'un événement avec `read()`.

Sockets

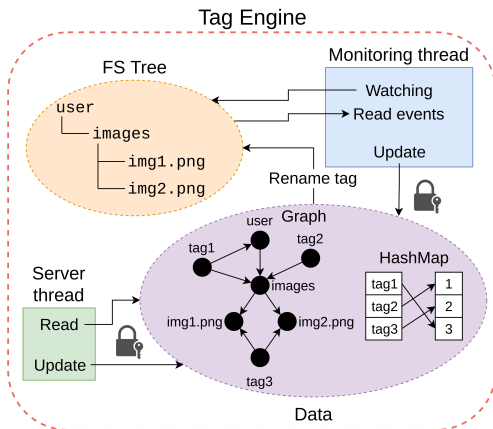


Tag Manager



- Programme "client".
- CLI.
- Gestion des tags.
- Manipule les XATTR des fichiers.
- Envoi de requêtes à Tag Engine.

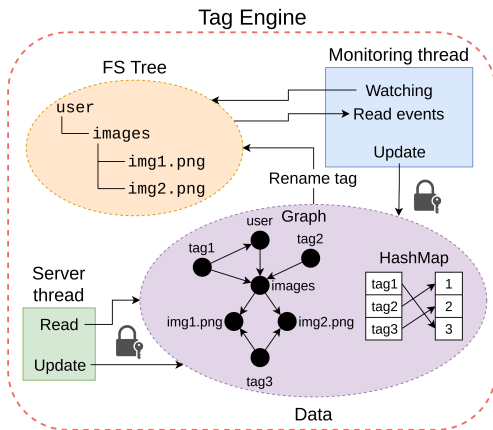
Tag Engine



h e p i a

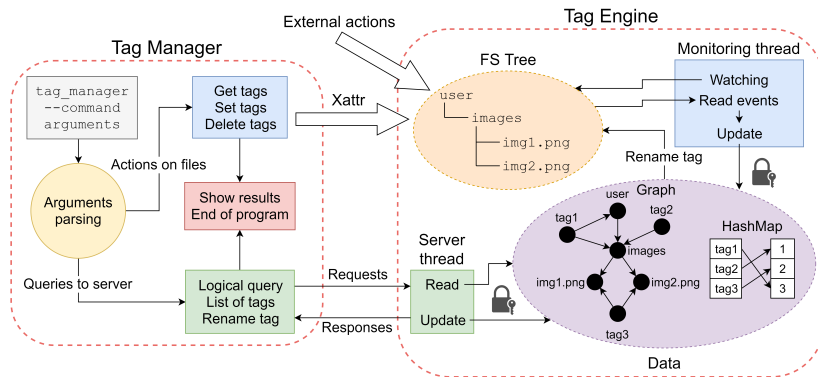
Haute école du paysage, d'ingénierie
et d'architecture de Genève

Tag Engine

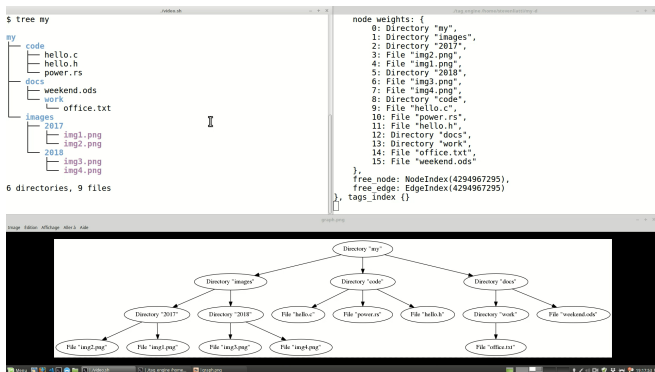


- Programme "serveur".
- Surveille l'arborescence des fichiers.
- Les changements sur le FS sont répercutés sur le graphe.
- Maintient la relation entre tags, fichiers et répertoires (graphe et hashmap).
- Écoute sur une socket les requêtes provenant de Tag Manager.
- Retourne la liste des fichiers correspondants à une requête logique (opérateurs OR et AND).

TagFS



Démo



Vidéo

h e p i a

Haute école du paysage, d'ingénierie
et d'architecture de Genève

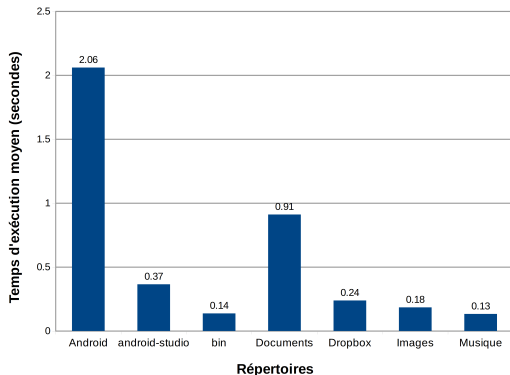
Hes·SO GENÈVE
Haute École Spécialisée
de Suisse occidentale

Mesures de performances

Répertoire	Sous-répertoires	Fichiers
Android	15'172	112'046
android-studio	3'331	13'287
bin	553	9'306
Documents	15'442	64'486
Dropbox	2'377	8'659
Images	5	863
Musique	135	1'352

Mesures de performances

Répertoire	Sous-répertoires	Fichiers
Android	15'172	112'046
android-studio	3'331	13'287
bin	553	9'306
Documents	15'442	64'486
Dropbox	2'377	8'659
Images	5	863
Musique	135	1'352



Améliorations

Améliorations

- GUI : environnement de bureau ou application web.

Améliorations

- GUI : environnement de bureau ou application web.
- Daemon pour Tag Engine.

Améliorations

- GUI : environnement de bureau ou application web.
- Daemon pour Tag Engine.
- Ajout de nouveaux répertoires de surveillance (partiel).

Améliorations

- GUI : environnement de bureau ou application web.
- Daemon pour Tag Engine.
- Ajout de nouveaux répertoires de surveillance (partiel).
- Gestion des périphériques amovibles (limitation inotify).

Améliorations

- GUI : environnement de bureau ou application web.
- Daemon pour Tag Engine.
- Ajout de nouveaux répertoires de surveillance (partiel).
- Gestion des périphériques amovibles (limitation inotify).
- Cache des dernières requêtes logiques adressées au serveur.

Améliorations

- GUI : environnement de bureau ou application web.
- Daemon pour Tag Engine.
- Ajout de nouveaux répertoires de surveillance (partiel).
- Gestion des périphériques amovibles (limitation inotify).
- Cache des dernières requêtes logiques adressées au serveur.
- Ajouter des opérateurs logiques (NOT).

Bilan personnel

Bilan personnel

- Conception d'un moteur de gestion de tags.

Bilan personnel

- Conception d'un moteur de gestion de tags.
- Étude du langage Rust.

Bilan personnel

- Conception d'un moteur de gestion de tags.
- Étude du langage Rust.
- Diverses technologies et approches.

Bilan personnel

- Conception d'un moteur de gestion de tags.
- Étude du langage Rust.
- Diverses technologies et approches.
- Cahier des charges rempli.

Bilan personnel

- Conception d'un moteur de gestion de tags.
- Étude du langage Rust.
- Diverses technologies et approches.
- Cahier des charges rempli.
- Progression personnelle : technologies, bonnes pratiques, démarche.

Remerciements

- Florent Glück
- Orestis Malaspinas
- Joël Cavat

Références I



Apple team.

Os x : Tags help you organize your files.

<https://support.apple.com/en-us/HT202754>, février 2015.

Consulté le 08.05.2018.



Wikipédia.

Un annuaire représenté comme une table de hachage.

https://fr.wikipedia.org/wiki/Table_de_hachage#/media/File:HASHTB08.svg, juin 2015.

Consulté le 23.06.2018.