



M1 INFORMATIQUE AIGLE

HMIN201

M1 TER

TER Software Heritage

Rapport Final

Groupe BAJONIM

Bachar RIMA,

bachar.rima@etu.umontpellier.fr

Joseph SABA,

joseph.saba@etu.umontpellier.fr

Tasnim SHAQURA,

tasnim.shaqura@etu.umontpellier.fr

Encadrant :

Jessie CARBONNEL

Responsable de l'UE :

Mattieu LAFOURCADE

27 mai 2019

Table des matières

1	Introduction	2
1.1	Description de Software Heritage	2
1.2	Contexte du TER	2
1.3	Plan du rapport	2
2	Problématique	3
2.1	La diaspora du code source	3
2.2	La fragilité du code source	3
2.3	Software Heritage en tant que solution	3
2.4	Notre contribution	3
3	Analyse	4
3.1	Fonctionnement de Software Heritage	4
3.1.1	Modèle des données	4
3.1.2	Architecture et flot des données	4
3.1.3	L'archive	4
3.2	Méthodologie	5
3.3	Planning Prévisionnel	5
4	Conception	6
5	Implémentation	7
6	Résultats	8
7	Conclusion	9
7.1	Planning final	9
7.2	Difficultés rencontrées	9
7.3	Perspectives	9
7.4	Bilan et apports du TER	9

Chapitre 1

Introduction

1.1 Description de Software Heritage

1.2 Contexte du TER

1.3 Plan du rapport

Chapitre 2

Problématique

2.1 La diaspora du code source

2.2 La fragilité du code source

2.3 Software Heritage en tant que solution

Current status et roadmap de SWH

2.4 Notre contribution

Chapitre 3

Analyse

3.1 Fonctionnement de Software Heritage

3.1.1 Modèle des données

Plateformes d'hébergement

Artéfacts logiciels

Informations sur la provenance des données

Structure des données

3.1.2 Architecture et flot des données

Flot d'ingestion des données

Listing

Loading

Scheduling

3.1.3 L'archive

Stockage des noeuds BLOB de l'archive

les IDs, les fichiers

Stockage des autres noeuds des archives

chemins, répertoires, snapshot, revisions, releases
postgres, etc

Stockage haché des objets

Mise en miroir des noeuds

Politique de rétention

Récupération automatique des objects corrompus

3.2 Méthodologie

Sourceforge sitemap, api

Launchpad api, client

Analyzing the listers (bitbucket, gitlab, github, eclipse, LIRMM, OpenHub, Assembla, GNU savannah

heritage, injection de dependances

conclusion : on adoptera une strategie pour definir un lister, loader ou autre

3.3 Planning Prévisionnel

Chapitre 4

Conception

design de la solution proposée (diagrammes + explications)

Chapitre 5

Implémentation

les technos qu'on a utilisé
bibliotheques
Outils (e.g. XML parsers)
Launchpad client

Chapitre 6

Résultats

pull request ?

Chapitre 7

Conclusion

7.1 Planning final

7.2 Difficultés rencontrées

7.3 Perspectives

7.4 Bilan et apports du TER

annexesrésumés code

Bibliographie