



Project Number 732223

D7.8 GitHub Integration

**Version 1.0
28 June 2019
Final**

Public Distribution

FrontEndART

Project Partners: Athens University of Economics & Business, Bitergia, Castalia Solutions, Centrum Wiskunde & Informatica, Eclipse Foundation Europe, Edge Hill University, FrontEndART, OW2, SOFTEAM, The Open Group, University of L'Aquila, University of York, Unparallel Innovation

Every effort has been made to ensure that all statements and information contained herein are accurate, however the CROSSMINER Project Partners accept no liability for any error or omission in the same.

© 2019 Copyright in this document remains vested in the CROSSMINER Project Partners.

Project Partner Contact Information

Athens University of Economics & Business Diomidis Spinellis Patision 76 104-34 Athens Greece Tel: +30 210 820 3621 E-mail: dds@aueb.gr	Bitergia José Manrique Lopez de la Fuente Calle Navarra 5, 4D 28921 Alcorcón Madrid Spain Tel: +34 6 999 279 58 E-mail: jsmanrique@bitergia.com
Castalia Solutions Boris Baldassari 10 Rue de Penthièvre 75008 Paris France Tel: +33 6 48 03 82 89 E-mail: boris.baldassari@castalia.solutions	Centrum Wiskunde & Informatica Jurgen J. Vinju Science Park 123 1098 XG Amsterdam Netherlands Tel: +31 20 592 4102 E-mail: jurgen.vinju@cwi.nl
Eclipse Foundation Europe Philippe Krief Annastrasse 46 64673 Zwingenberg Germany Tel: +33 62 101 0681 E-mail: philippe.krief@eclipse.org	Edge Hill University Yannis Korkontzelos St Helens Road Ormskirk L39 4QP United Kingdom Tel: +44 1695 654393 E-mail: yannis.korkontzelos@edgehill.ac.uk
FrontEndART Rudolf Ferenc Zászló u. 3 I./5 H-6722 Szeged Hungary Tel: +36 62 319 372 E-mail: ferenc@frontendart.com	OW2 Consortium Cedric Thomas 114 Boulevard Haussmann 75008 Paris France Tel: +33 6 45 81 62 02 E-mail: cedric.thomas@ow2.org
SOFTEAM Alessandra Bagnato 21 Avenue Victor Hugo 75016 Paris France Tel: +33 1 30 12 16 60 E-mail: alessandra.bagnato@softeam.fr	The Open Group Scott Hansen Rond Point Schuman 6, 5 th Floor 1040 Brussels Belgium Tel: +32 2 675 1136 E-mail: s.hansen@opengroup.org
University of L'Aquila Davide Di Ruscio Piazza Vincenzo Rivera 1 67100 L'Aquila Italy Tel: +39 0862 433735 E-mail: davide.diruscio@univaq.it	University of York Dimitris Kolovos Deramore Lane York YO10 5GH United Kingdom Tel: +44 1904 325167 E-mail: dimitris.kolovos@york.ac.uk
Unparallel Innovation Bruno Almeida Rua das Lendas Algarvias, Lote 123 8500-794 Portimão Portugal Tel: +351 282 485052 E-mail: bruno.almeida@unparallel.pt	

Table of Contents

Executive Summary	1
1 Introduction	2
2 Technical documentation	3
2.1 Overview	3
2.2 Implementation	4
3 Install the App	5
4 Requirements coverage	7

Document Control

Version	Status	Date
0.1	Document outline	14 June 2019
0.4	Figures added	25 June 2019
0.8	Ready for internal review	26 June 2019
1.0	Final version	28 June 2019

Executive Summary

The GitHub Integration component of the CROSSMINER platform provides the developers of open source GitHub projects the ability to be analysed by a CROSSMINER server in an easy way. The component is developed in T7.6 (GitHub integration) of work package WP7 (Advanced Integrated Development Environments). The component consists of two parts, which, relying on the CROSSMINER server features, enable the developers of open source projects on the GitHub platform to have their projects analyzed by a deployed CROSSMINER server. The first part is a GitHub App, the second is a server handling the GitHub events and performing actions.

This document presents deliverable D7.8 (GitHub Integration) and describes how the GitHub App server works, and how the App can be used.

1 Introduction

The goal of the CROSSMINER platform is to provide the software developers with relevant information that is useful in the development process. One of the main sources of these information are the open source projects. The more projects are analyzed the more accurate data can be extracted and synthesized. Thus, it is important to include more and more open source projects in the analysis.

On the other hand, such an analysis may provide visibility. The CROSSMINER platform can recommend e.g. the use of analyzed project, and can offer patterns how to use it. These may improve the usability and popularity of the project.

The GitHub Integration Services component of the CROSSMINER platform enables GitHub users to easily utilize the CROSSMINER platform. By installing the *CROSSMINER GitHub App* for their GitHub project, the project will be included in the set of analyzed projects of the given CROSSMINER server.

Figure 1 shows how the GitHub Integration Services are connected to the CROSSMINER platform. It is connected to the Logic Layer (the CROSSMINER server) through the common CROSSMINER Remote API. The implementation of the GitHub Integration Services are described in Section 2.

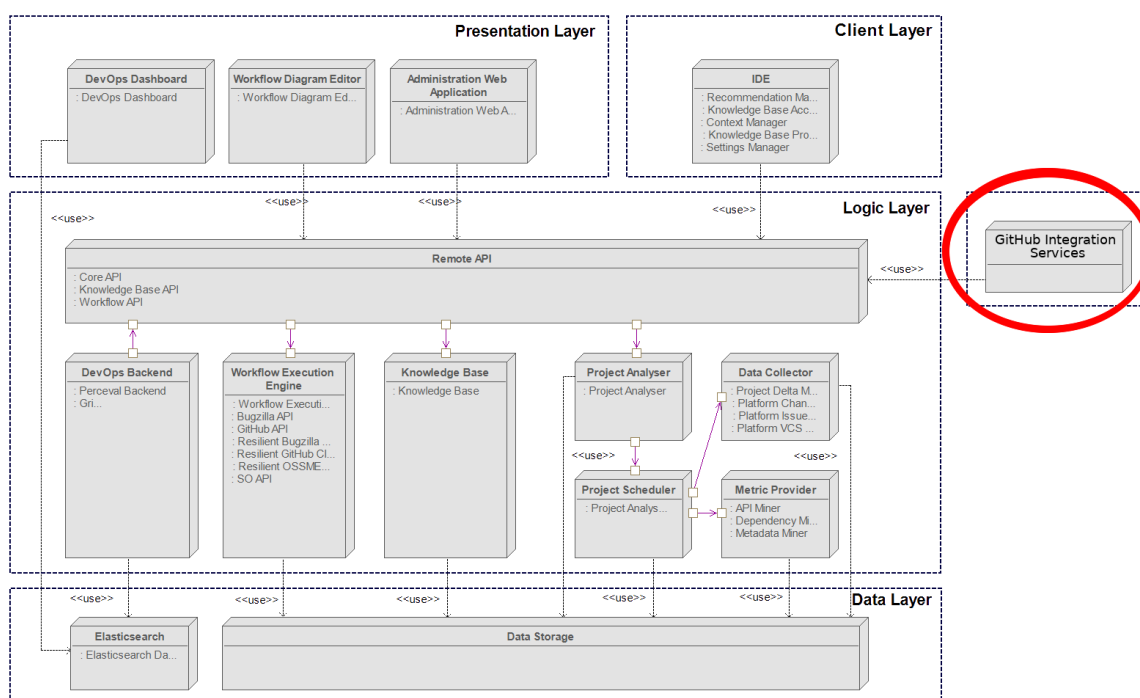


Figure 1: Location of the GitHub Integration Services in the CROSSMINER platform

2 Technical documentation

In this chapter we present how the CROSSMINER platform was integrated with the GitHub platform.

2.1 Overview

The main components of the integration are shown on Figure 2. Physically, there are two servers (blue boxes in the figure): a GitHub server and a CROSSMINER server. GitHub server hosts the git repo of open source projects, e.g. “MyRepo”, and provides GitHub API to access different events of the GitHub repo. We register a *CROSSMINER GitHub App* on the GitHub server that points to the *CROSSMINER GitHub Integration Server* as the handler of the registered events. The app requires access to the GitHub API *installation* and *push* events (will be granted when the app is installed for a GitHub repository).

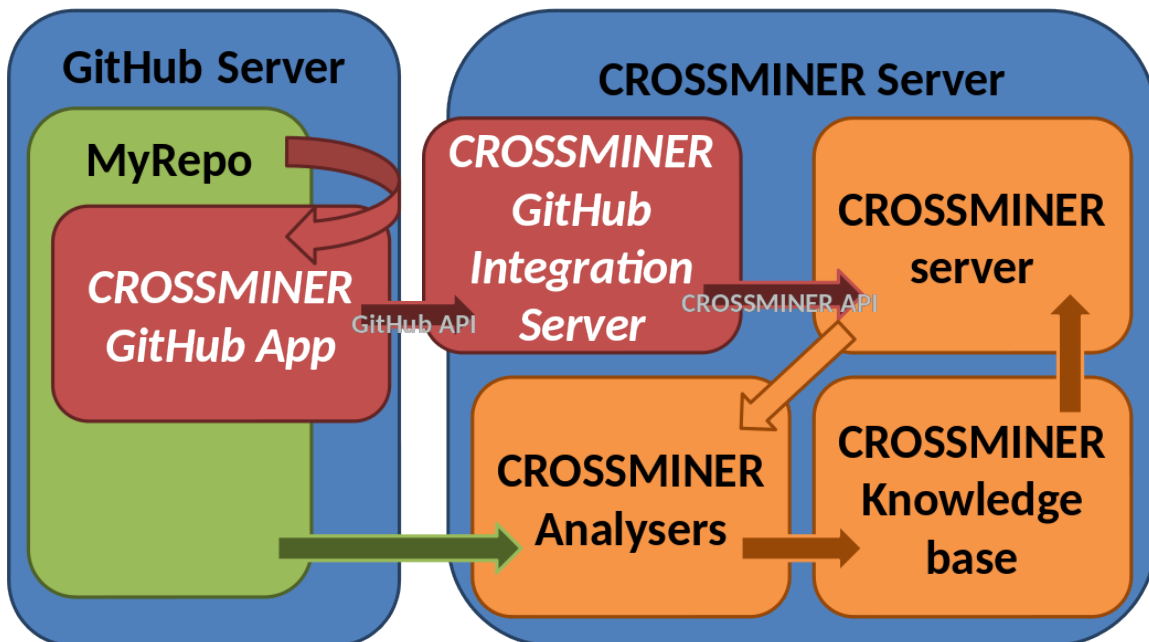


Figure 2: Components of the GitHub Integration Services in the CROSSMINER platform

After the *CROSSMINER GitHub App* is installed for a repository, it will notify the *CROSSMINER GitHub Integration Server* on all push events. In the case of such an event, the *CROSSMINER GitHub Integration Server* will check whether a `SCAVA.URL` file is added to the repository (in the root directory). If it is added, its single line content will be treated as a CROSSMINER API server URL, and the GitHub project will be added to the list of observed repositories of the given CROSSMINER server (by calling the corresponding CROSSMINER API endpoint).

The CROSSMINER server will then manage the daily analysis of the repository. It will checkout the head of the given repository, run the corresponding analyzers, and upload the extracted information to the CROSSMINER Knowledge Base.

2.2 Implementation

A GitHub App consists of a server functioning as a webhook for the GitHub REST API calls, and an “App” registered on GitHub.

The *CROSSMINER GitHub Integration Server* is implemented with Java Spring technology, and provides handlers for GitHub REST API *installation* and *push* events. It stores no information, simply reacts on GitHub API *push* events and calls the corresponding *CROSSMINER REST API* endpoint if necessary.

The “App” can be configured on the GitHub site, it needs the address of the running *CROSSMINER GitHub Integration Server*, and access to the *installation* and *push* events of the GitHub REST API.

3 Install the App

To install the CROSSMINER GitHub App, one should go to the app's page¹ on `github.com`, and press the *Install* button (see Figure 3). Then, the developer must select the repos they want the CROSSMINER GitHub App to be installed for. By pressing the *Install* button, the developers' repositories will be registered by the App and its *app server* will receive the corresponding GitHub events. On *push* events, the *app server* will check for the `SCAVA.URL` file and will register the project on the given CROSSMINER server.

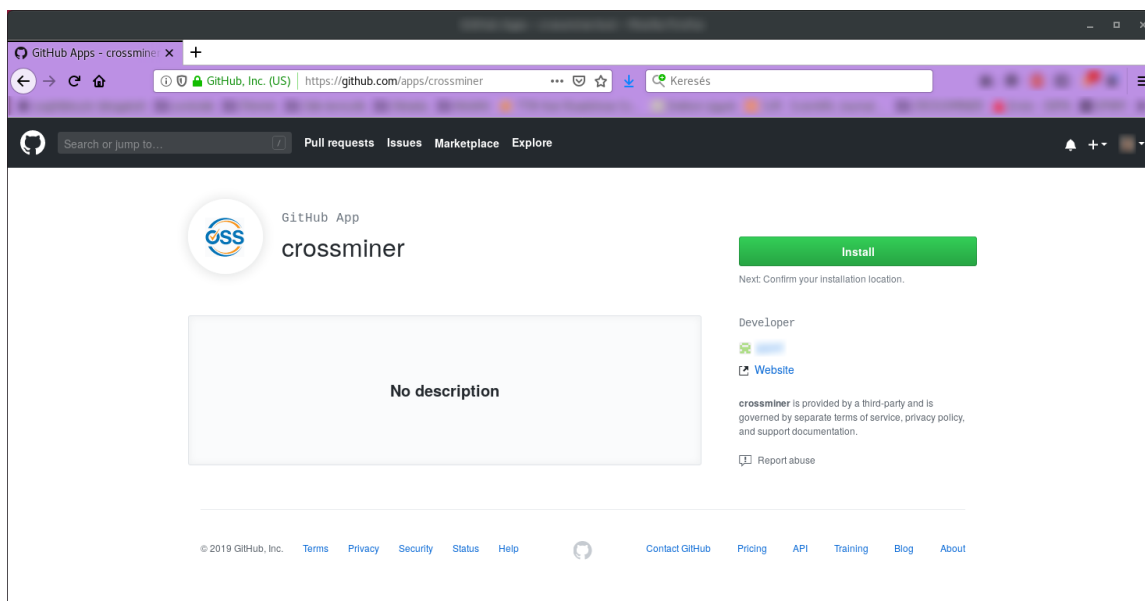


Figure 3: GitHub Integration Services install steps – App install page

¹Note, that the address `https://github.com/apps/crossminer` is currently a “fake” address used for testing during the development. As GitHub apps require a fixed webhook server that handles GitHub API events, the final App can be registered on GitHub after the CROSSMINER platform is integrated and a central server is set up to host the webhook server.

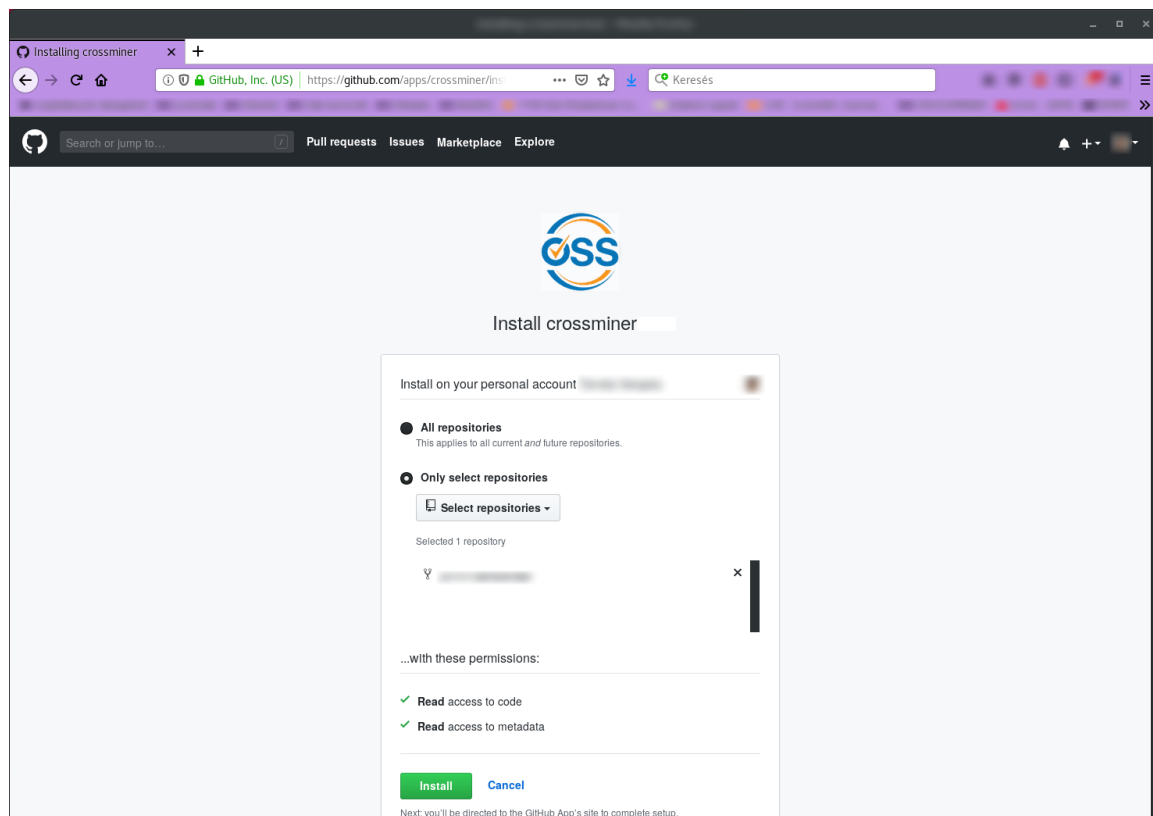


Figure 4: GitHub Integration Services install steps – Repo selection and permissions

4 Requirements coverage

In this deliverable, the GitHub Integration Services was prepared. In the following we show how the use case requirements related to the GitHub Integration Services and defined in the Project Requirements document (deliverable D1.1) are covered by the implementation. Note, that these requirements cover features that are utilized by the GitHub Integration, but were implemented in other workpackages (WP2, WP3, and WP4).

In the last column of the tables an empty circle (○) denotes that the requirement is minimally (or not) covered, a half-filled circle (◐) denotes that it is partially covered, and a filled circle (●) denotes that it is mostly (or fully) covered. Requirements marked with a cross (×) became invalid for some reasons during the development.

U184	Parsers use data dumps for all data sources	SHALL	●
U185	Communication channel parsers use mboxes	SHALL	●
U186	Code parsers use git clones	SHALL	●
U187	Bugzilla parsers use data dumps	SHALL	●
U188	Documentation parsers use data dumps	SHALL	●
U189	Able to analyse Docker files in GitHub repositories	SHALL	●
U190	Able to capture third-party API changes within 48 hours from time change is committed to the GitHub repo	SHOULD	●