

REVEAL**FP7-610928****REVEALing hidden concepts in Social Media**

Deliverable D6.2.1**Integrated Platform**

| | |
|-----------------------------|---------------------|
| Editor(s): | Eftsratios Tzoannos |
| Responsible Partner: | ATC |
| Status-Version: | V0.4 |
| Date: | 01/09/2015 |
| EC Distribution: | PU |

| | |
|------------------------|------------|
| Project Number: | FP7-610928 |
| Project Title: | REVEAL |

Project Title: REVEAL
Project Coordinator: INTRASOFT International S.A.

Contract No. FP7-610928
www.revealproject.eu

| | |
|------------------------------------|---------------------|
| Title of Deliverable: | INTEGRATED PLATFORM |
| Date of Delivery to the EC: | 01/09/2015 |

| | |
|---|---|
| Workpackage responsible for the Deliverable: | WP6 – Framework Integration and REVEAL Applications |
| Editor(s): | Giorgos Giotis, Efstratios Tzoannos, Nikos Sarris |
| Contributor(s): | NCSR'D', CERTH, UKOB, ITINNO, ALBLF |
| Reviewer(s): | INTRASOFT |
| Approved by: | All Partners |

| | |
|----------------------|--|
| Abstract: | This document is a short report accompanying the prototype deliverable of the REVEAL platform. In the course of the project, as the specification of the functionality of the individual modules progresses, this document will be revised and further documented in the deliverable – Integrated Platform (D6.2.2). |
| Keyword List: | integrated platform, integration |

DOCUMENT DESCRIPTION

Document Revision History

| <i>Version</i> | <i>Date</i> | <i>Modifications Introduced</i> | |
|----------------|-------------|--|--------------------|
| | | <i>Modification Reason</i> | <i>Modified by</i> |
| v0.1 | 24/08/2015 | Table of content | ATC |
| v0.2 | 31/08/2015 | Final draft | ATC |
| v0.3 | 01/09/2015 | Review of content & provision of comments | INTRASOFT |
| v0.4 | 01/09/2015 | Address of internal review comments and final formatting | ATC |

CONTENTS

| | |
|---|-----------|
| EXECUTIVE SUMMARY | 7 |
| 1 PHYSICAL SERVER TOPOLOGY | 8 |
| 2 KEY COMPONENT ENDPOINTS | 11 |
| 2.1 MONGO..... | 11 |
| 2.2 APACHE SOLR..... | 11 |
| 2.3 RABBITMQ | 11 |
| 2.4 THE PERSONALISATION SERVER..... | 12 |
| 3 CONTINUOUS INTEGRATION | 14 |

LIST OF FIGURES

| | |
|--|----|
| FIGURE 1: RABBITMQ MANAGEMENT CONSOLE. | 12 |
| FIGURE 2: RABBITMQ EXCHANGE FOR SHARING ITEMS ACROSS MODULES AND ACROSS WPs..... | 12 |
| FIGURE 3: THE PERSONALISATION SERVER | 13 |
| FIGURE 4: JENKINS UI..... | 14 |
| FIGURE 5: REVEAL PROJECTS IN GITLAB | 15 |

DEFINITIONS, ACRONYMS AND ABBREVIATIONS

| Acronym | Title |
|---------|-----------------------------------|
| API | Application Programming Interface |
| UI | User Interface |
| CI | Continuous integration |

Executive Summary

Deliverable D6.2.1 represents the REVEAL integrated platform (deliverable of Prototype nature), which is deployed in a distributed fashion on a large number of servers in the premises of the integrator (ATC). We accompany its delivery with this report which acts as a reference describing the main components of the prototype explaining where these are installed.

1 Introduction

In D6.1.2 “Architecture and Specifications”, the architecture of the REVEAL platform was derived according to established design principles and in direct correspondence with the identified system requirements. The architecture specified the main functional components of the envisioned system and the interworking between these components. This deliverable is structured in the following way:

- Section 1 maps the modules installed on the physical topology of the deployed servers
- Section 2 describes the key components shared between most modules
- Section 3 provides an update on the tools used for continuous integration

Modifications will continue in the platform including more mature modules from the research components that will be later described in D6.2.2.

2 Physical server topology

The table that follows contains the listing of the REVEAL modules installed and their mapping on the physical topology of the servers hosting them. Some of the modules have been deployed on the Docker platform inside the servers constituting thus the logical topology of the whole architecture.

| | | | |
|---|--|---|---|
| <p><i>Reveal.atc.gr</i></p> <ul style="list-style-type: none"> • Social Media Client Framework • Influencers module • Orchestrator central • Orchestrator controller • RabbitMQ broker | <p><i>Social1.atc.gr</i></p> <ul style="list-style-type: none"> • MongoDB • Orchestrator controller | <p><i>Social2.atc.gr</i></p> <ul style="list-style-type: none"> • Apache Solr • Orchestrator controller | <p><i>Storm-1.atc.gr</i></p> <ul style="list-style-type: none"> • Sentiment analysis • Orchestrator controller |
| <p><i>Storm-2.atc.gr</i></p> | <p><i>Storm-3.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • OpenStreetMap database • Zookeeper-Nimbus-Controller • Storm Supervisor • Geospatial context extraction (pre-) processing • Situation assessment framework | <p><i>Storm-4.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • Storm Supervisor • OpenStreetMap database • Geospatial context extraction (pre-) processing • Situation assessment framework | <p><i>Storm-5.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • Storm Supervisor • OpenStreetMap database • Geospatial context extraction (pre-) processing • Situation assessment framework |
| <p><i>Storm-6.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • Storm Supervisor | <p><i>Storm-7.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • Web Crawler | <p><i>Storm-8.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • Source finding | <p><i>Storm-9.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • Geospatial topic |

| | | | |
|---|--|---|--|
| <ul style="list-style-type: none"> • OpenStreetMap database • Geospatial context extraction (pre-) processing • Situation assessment framework | | <ul style="list-style-type: none"> • Influence calculation | <p>model</p> <ul style="list-style-type: none"> • Focused Geospatial Exploration • Social context model |
| <p><i>Storm-10.atc.gr</i></p> | <p><i>Squall-1.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • PServer | <p><i>Squall-2.atc.gr</i></p> | <p><i>Squall-3.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • Presence • Multi-partite community detection • Influence similarity • Relation discovery • Semantic segmentation • Linguistic analysis (NER) • Complex Event Recognition |
| <p><i>Squall-4.atc.gr</i></p> | <p><i>Squall-5.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • User network profile classifier • Multimedia indexing • Multimedia similarity retriever • Multimedia concept detector • Multimedia clusterer | <p><i>Squall-6.atc.gr</i></p> <ul style="list-style-type: none"> • Orchestrator controller • Multimedia summariser • User network profile classifier • SN structure collector • Multimedia geolocator • Multimedia social context builder and modality extractor | <p><i>Squall-7.atc.gr</i></p> |

3 Key component endpoints

The following components have been identified as key components in the sense that they are shared between most of the REVEAL modules. They are important for integrating the various REVEAL modules together by assuring the smooth flow of the data as well as the efficient indexing and storing of data in the various databases/repositories.

3.1 Mongo

MongoDB is a powerful NoSQL object that will act as the permanent archive for the crawled objects. This for the moment hosts one collection for storing the WP2 Twitter REST API crawler (TwitterRestUsersDB) and several dynamically created collections based on ITINNO crawler. The idea is that for every object inserted into RabbitMQ, a new collection with the assessment id will be created populated with the mirrored content of RabbitMQ. The contained data is subsequently going to be used by the offline analysis modules for the batch processing. In some cases, mongodb is going to be used also as intermediate buffer, keeping the various temporary collections with the enriched metadata of each analyzed item. We have installed mongodb at social1.atc.gr and its admin console is found at: <http://social1.atc.gr:27017/>

3.2 Apache Solr

Apache Solr is going to be considered as a central repository, which is going to be used for the full-text-search purposes of the user interface. The various modules (mainly) of WP3 are going to feed content into it. The current installation is inside social2.atc.gr (v.4.7.2) and it is listening on port 80. The user management UI is found at: <http://social2.atc.gr/solr/#/>

We expect feedback by the various WP owners to create the collections needed together with the schema of each.

3.3 RabbitMQ

RabbitMQ is the message broker, which is going to be used for the communication of the near-real time modules of the storm topologies. Inside the topology it will play the role of a spout. It will play the role of buffer for the ITINNO crawler pushing items inside it.

The version installed is 3.4.0 and it is hosted on reveal server (reveal.atc.gr) in the container “ITINNO crawler”. We have enabled the RabbitMQ Management Console, accessible at: <http://reveal.atc.gr:42225/>

This URL endpoint corresponds to the RabbitMQ message bus, the management console of RabbitMQ broker is running at: <http://reveal.atc.gr:42224/>, as shown in Figure 1.

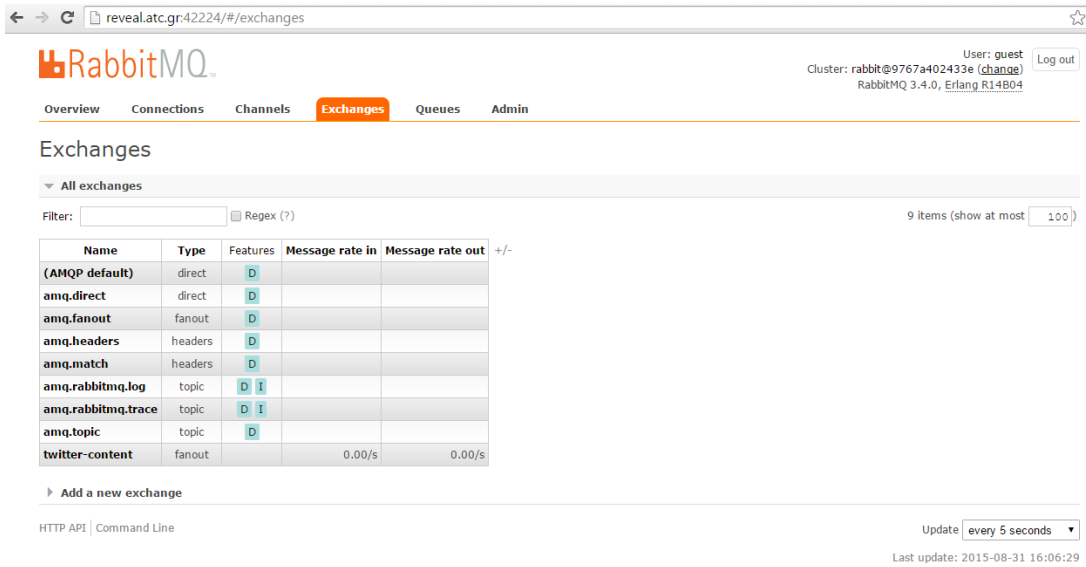


Figure 1: RabbitMQ management console.

The main exchange where the modules register their queues at in order to consume the items is named “twitter-content”, as shown in Figure 2.

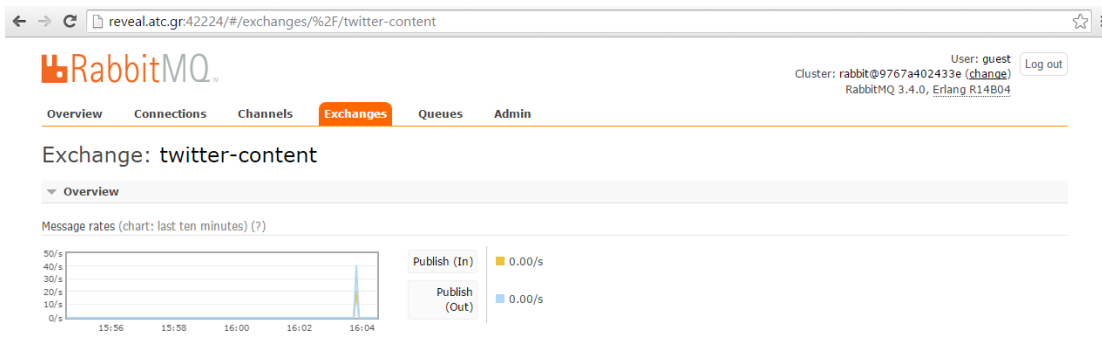


Figure 2: RabbitMQ exchange for sharing items across modules and across WPs.

3.4 The Personalisation Server

PServer which manages all user profiles and preferences has been installed under:

<http://squall-1.atc.gr:1111/admin>

← → | squal-1.atc.gr:1111/admin

PServer Administration panel

| | |
|---|---|
| Index page | A general-puprose Personalization Server called PServer . |
| PServer clients | Manages user profiles and preferences, in the frame of an application. Helps the application to tailor itself to its users. |
| Change PServer properties | Extends YServer, a simple but expandable web server. YServer supports multiple users, requests for file resources under a public directory root. GET and POST HTTP methods, MIME types, and loading application settings from an initialization file. Presently PServer uses RDBMS to store its data and had been tested with MS Access and MySQL Databases but it is created to work with any database supports the standar SQL syntax. |
| Pers mode help | An extended documentation of the PServer can be found by running the server and connecting through a web browser to: <code>http://persserver:port/</code> Three main modes are offered: Personal, Stereotypes and Communities. Each mode supports a number of operations that can be performed by issuing suitable HTTP requests to the Personalization Server. The two modes are independent from each other. They share the same database, however they are supported by a separate set of DB tables. The syntax of those special requests are as follows: <code>http://server:port/<clnt=client_name client_pass><mode_id><query_string></code> The clnt part contains the name and the password of the client that wants to be serviced by the PServer and can be excluded if the server sunt in anonymous mode. This can be aranged by PServer properties For Personal mode the mode ID is "pers", while for Stereotypes mode the mode ID is "ster" and for communities mode the mode ID is "commu". The query string identifies the operation and its parameters, and is described separately for each operation. Reserved words in the query string are case independent. All names in the query string must conform to the syntax defined on URLs, that is, they must not contain spaces, +, =, &, ?, etc. In addition, to avoid confusion with XML syntax, the characters < and > are also not allowed. The answer is formatted as XML. The XML answer must have a single element called <code>encompassing</code> , encompassing a number of elements, each containing the data elements corresponding to the specific request. By having such a standard format it becomes easier for the applications to parse all XML answers in a uniform way. For error handling, a number of HTTP error codes are used to denote success, client error (wrong request), or server error. |
| Ster mode help | |
| Commu mode help | |

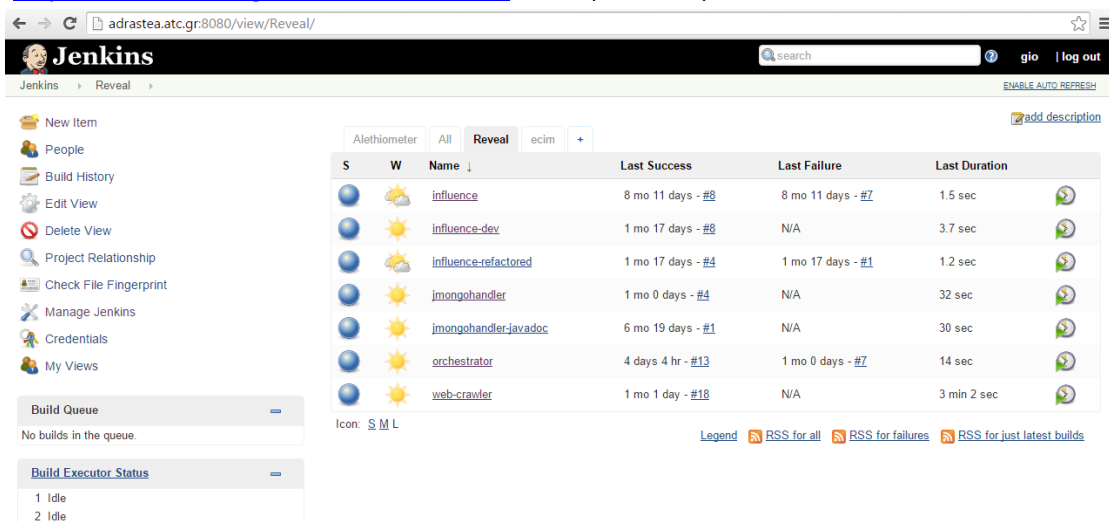
Figure 3: The Personalisation Server

4 Continuous Integration

The integration approach with Docker is described in detail inside D6.1.2. We have installed Docker platform v.1.3 at reveal.atc.gr server and we have released the necessary guidelines (documented in D6.1.2) to the Reveal Development Community.

The Continuous Integration Workflow followed is described in detail in D6.1.2. A brief reference to the tools with any updates is provided in this section

Jenkins¹: Jenkins is an open source tool for CI processes. It monitors the execution of repeated jobs, such as building a software project, focusing on building and testing software projects continuously and providing an easy-to-use so-called continuous integration system, making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. The automated, continuous build increases the productivity. Jenkins can, also, monitor the execution of externally-run jobs, such as cron jobs and procmail jobs, even those that are run on a remote machine. It is hosted at: <http://adrastea.atc.gr:8080/view/Reveal/> and is password protected.



| S | W | Name ↓ | Last Success | Last Failure | Last Duration |
|---|----|---------------------------------------|-------------------|-------------------|---------------|
| 🌍 | ☀️ | influence | 8 mo 11 days - #8 | 8 mo 11 days - #Z | 1.5 sec |
| 🌍 | ☀️ | influence-dev | 1 mo 17 days - #8 | N/A | 3.7 sec |
| 🌍 | ☀️ | influence-refactored | 1 mo 17 days - #4 | 1 mo 17 days - #1 | 1.2 sec |
| 🌍 | ☀️ | jmongohandler | 1 mo 0 days - #4 | N/A | 32 sec |
| 🌍 | ☀️ | jmongohandlercjavadoc | 6 mo 19 days - #1 | N/A | 30 sec |
| 🌍 | ☀️ | orchestrator | 4 days 4 hr - #13 | 1 mo 0 days - #7 | 14 sec |
| 🌍 | ☀️ | web-crawler | 1 mo 1 day - #18 | N/A | 3 min 2 sec |

Figure 4: Jenkins UI

Apache Maven²: Apache Maven is a software project management and comprehension tool. Based on the concept of a Project Object Model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information. It is important that all modules stay compatible with maven rules, sharing the same .pom file. In this way, the integration will be easily controlled through maven and moreover, the project will be independent of an IDE; as a maven project can be opened in Eclipse, Netbeans, IDEA etc.

¹ <http://jenkins-ci.org/>

² <http://maven.apache.org/>

Redmine³: A flexible project management open source web application for issue tracking and ticketing. The public REVEAL Redmine instance (v.2.1.2.stable.10669) is hosted at: <http://prometheus.atc.gr/projects/reveal/issues>

Gitlab⁴: A software versioning and a revision control system distributed under a free license. We have installed v.7.2.1 of Gitlab at a dedicated server: gitlab.atc.gr. We have also purchased and installed a signed certificate in order to secure the communication and avoid any malicious actions against it. The gitlab web interface is exposed at: <https://gitlab.atc.gr/> and the REVEAL central repository is at: <https://gitlab.atc.gr/groups/reveal> and is password protected, accessible only by members of REVEAL community.

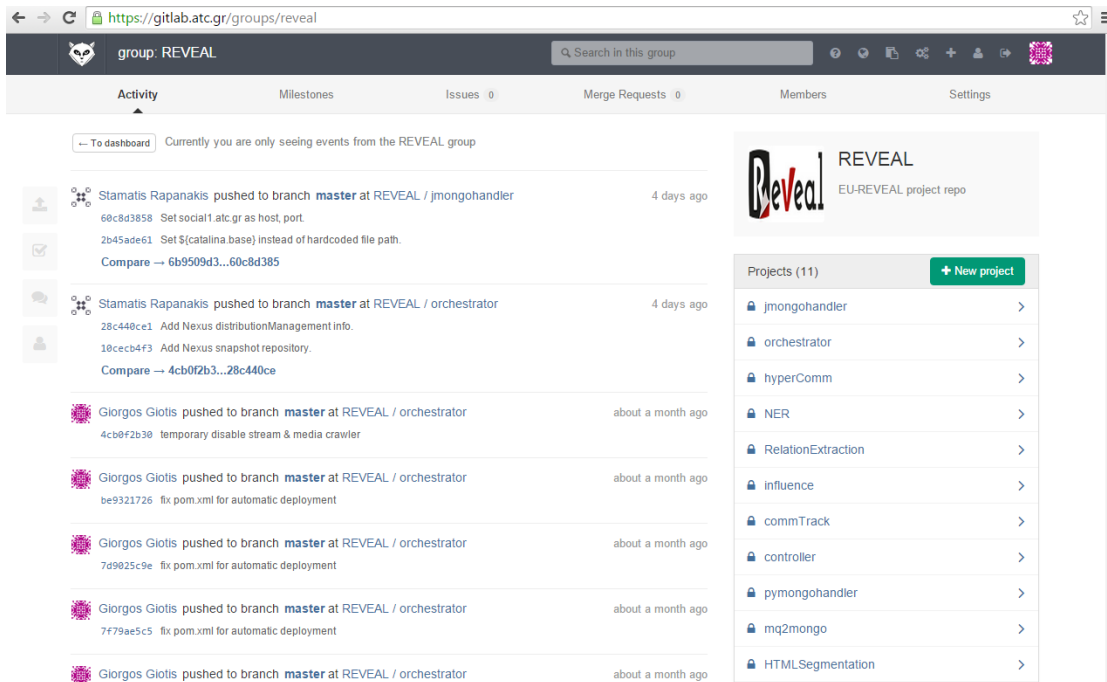


Figure 5: REVEAL projects in Gitlab

SonarQube⁵: SonarQube is an open source software quality platform. Sonar uses various static code analysis tools such as Checkstyle, FindBugs, Clover to extract software metrics, which then can be used to improve software quality. Version 4.5.1 of SonarQube is installed at: <http://adrastea.atc.gr:9000/dashboard>

Sonatype Nexus⁶: Nexus is an advanced repository management tool for controlling all the build and third party dependencies of a project. It can be used among a team of developers for sharing common libraries. We have installed nexus sonatype at: <http://prometheus.atc.gr:8080/nexus/>

For security purposes it is password protected and can be accessed only by REVEAL members. The reveal release repo is found at: <http://prometheus.atc.gr:8080/nexus/content/repositories/releases/eu/reveal/>

³ <http://www.redmine.org/>

⁴ <https://subversion.apache.org/>

⁵ <http://www.sonarqube.org/>

⁶ <http://www.sonatype.org/nexus/>

5 Conclusions

This report provided a quick reference to all parts of the Reveal integrated framework, which connects all software modules under a common orchestration, as designed in the system architecture documented in D6.1.2. This framework will continuously evolve until the end of the project with new modules being added and new functionalities deployed. This framework will be used to produce the two foreseen applications servicing the news and the enterprise industries. These applications will be delivered with D6.3.1 and D6.4.1.