Callisto
Software Transfer Document
Version 1.0.1

Project team
J.N. Adegeest | 0841357
G. Foks | 0847293
S.J. den Hertog | 0853459
K.R. Klaren | 0812587
S. Lovett | 0927993
K. Man | 0929573
K. Nassar | 0926380
B. Raducanu | 0870280
T. van Schijndel | 0857767
J.W. Sleijster | 0856823

Project managers
M. Brunings
L.D. Stooker

Project supervisor
N. Zannone

Customer
M.A. Weffers-Albu
Abstract

This document is the Software Transfer Document for The Callisto plug-in. Callisto is a plug-in for Jupyter which enables analysis of data from the MIMIC-III database [1]. The MIMIC-III database contains medical information of more than 40,000 critical care patients. With this information, researchers hope to obtain relevant information which can help critical care patients in the future. This document complies with the ESA software standard.
## Contents

1 Introduction ................................................................................. 6  
   1.1 Purpose ............................................................................. 6
   1.2 Scope ............................................................................... 6
   1.3 List of definitions and abbreviations .................................. 7  
      1.3.1 Definitions ................................................................. 7
      1.3.2 Abbreviations .............................................................. 8
   1.4 List of references ............................................................... 8
   1.5 Overview ........................................................................... 9

2 Build procedure ........................................................................... 10

3 Installation procedure ................................................................. 11  
   3.1 Development ................................................................. 11
   3.2 Deployment server with both MIMIC-III and Jupyter Hub .... 12
   3.3 MIMIC-III ..................................................................... 16

4 Configuration item list ................................................................. 17  
   4.1 Documentation ............................................................... 17
   4.2 Test plans ....................................................................... 17
   4.3 Software ......................................................................... 17

5 Acceptance test report summary .................................................. 18

6 Software Problem Reports .......................................................... 19  
   6.1 Matlab graph export ....................................................... 19
   6.2 Overlapping description tag with X-axis labels .................. 19
   6.3 Slow preset: Age vs. Disease Class vs. Average ICU Stay .... 19

7 Software Change Requests .......................................................... 20

8 Software Modification Reports .................................................... 21
DOCUMENT STATUS SHEET

GENERAL

Document title: Software Transfer Document v1.0.1
Identification: STD/1.0.1
Authors: K.W. Man, K. Nassar
Document status: Final version

DOCUMENT HISTORY

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author(s)</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.1</td>
<td>12-10-2016</td>
<td>K.W. Man</td>
<td>Initial document structure</td>
</tr>
<tr>
<td>0.0.2</td>
<td>08-11-2016</td>
<td>K. Nassar</td>
<td>Added development installation procedure</td>
</tr>
<tr>
<td>0.0.2</td>
<td>08-11-2016</td>
<td>K. Nassar</td>
<td>Added acceptance test report summary</td>
</tr>
<tr>
<td>0.0.2</td>
<td>08-11-2016</td>
<td>K. Nassar</td>
<td>Added MATLAB graph export problem report</td>
</tr>
<tr>
<td>0.0.3</td>
<td>09-11-2016</td>
<td>K. Nassar</td>
<td>Revised development installation procedure</td>
</tr>
<tr>
<td>0.0.4</td>
<td>10-11-2016</td>
<td>K. Nassar</td>
<td>Added overlapping description tag with X-axis labels problem report</td>
</tr>
<tr>
<td>0.0.4</td>
<td>10-11-2016</td>
<td>K. Nassar</td>
<td>Added overview</td>
</tr>
<tr>
<td>1.0.0</td>
<td>10-11-2016</td>
<td>K. Nassar</td>
<td>Added Deployment server with both MIMIC-III and Jupyter Hub</td>
</tr>
<tr>
<td>1.0.1</td>
<td>10-11-2016</td>
<td>K. Nassar</td>
<td>Added Slow preset: age vs. disease class vs. average ICU problem report</td>
</tr>
<tr>
<td>Section</td>
<td>Reason</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.x</td>
<td>Added abbreviations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Added installation procedure for development environment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Added Acceptance test report summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Added MATLAB graph export software problem report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.2</td>
<td>Added overlapping description tag with X-axis labels problem report</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td>Added document overview</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.2</td>
<td>Added Deployment server with both MIMIC-III and Jupyter Hub</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.3</td>
<td>Added Slow preset: age vs. disease class vs. average ICU problem report</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1 INTRODUCTION

1.1 PURPOSE

This document gives all the information needed to transfer the project to the customer. It describes how to install Jupyter Hub with the Callisto plug-in and how to setup the MIMIC-III database. Furthermore it the items to be transferred and how the product has been tested prior to transfer. Lastly the state of the items is compared to the requirements from the URD [2].

1.2 SCOPE

The Pyoneers is a Bachelor End Project group working for the TU/e and Royal Philips. The software product is a plug-in for Jupyter which is created by The Pyoneers for the collaborative initiative lead by the Research of Chronic Disease Management department at Royal Philips, and the department of Mathematics and Computer Science Flagship Data Science of Eindhoven University of Technology. The goal of the application is to develop a data science environment in which the researchers can use the data in the MIMIC-III database. The application will allow the user to generate graphs from the MIMIC-III data in an easy and user friendly manner. Text can be added to the graph to clarify the results that the graph represents. This will help researchers in gaining insights into relevant correlations. In the end this will all contribute to making it easier for the scientists to do research using the MIMIC-III database [1] to find insights in health care and treatment processes. When the researcher feels that the document is finished, they can export the document to Microsoft Word (.docx) format to easily share their findings. Furthermore, Callisto will allow scientists to save and resume their work.
## 1.3 LIST OF DEFINITIONS AND ABBREVIATIONS

### 1.3.1 DEFINITIONS

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Callisto Cell</td>
<td>A Jupyter cell which contains the graphical user interface for the creation of the graph and the graph itself. Furthermore it contains options for exporting to Microsoft Word and MATLAB.</td>
</tr>
<tr>
<td>Disease class</td>
<td>All diseases which would be in the same category, is called a disease class.</td>
</tr>
<tr>
<td>Jupyter Notebook</td>
<td>A web application that allows the creation of live code, visualisations and text.</td>
</tr>
<tr>
<td>Jupyter Project</td>
<td>A document in the Jupyter Notebook which can contain a set of cells.</td>
</tr>
<tr>
<td>Jupyter Hub</td>
<td>An application which enables an instance of Jupyter Notebook for every user which logs in.</td>
</tr>
<tr>
<td>MATLAB</td>
<td>A multi-paradigm computing environment for numerical data.</td>
</tr>
<tr>
<td>MIMIC Graph</td>
<td>A graph created with the data of the MIMIC-III database.</td>
</tr>
<tr>
<td>Philips</td>
<td>A company focused on improving people’s lifestyle with meaningful innovations in healthcare, consumers lifestyles and lighting.</td>
</tr>
<tr>
<td>Python</td>
<td>Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.</td>
</tr>
<tr>
<td>JavaScript</td>
<td>An untyped, dynamic and interpreted general purpose programming language.</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>An object-relational database management system which is used to store data and interact with it in a structured and secure manner.</td>
</tr>
<tr>
<td>Node.js</td>
<td>A cross-platform runtime environment for development and deployment of JavaScript code in various settings.</td>
</tr>
<tr>
<td>NPM</td>
<td>A package manager for managing JavaScript modules for Node.js.</td>
</tr>
<tr>
<td>psql</td>
<td>A tool for interacting and managed PostgreSQL database servers.</td>
</tr>
<tr>
<td>pip</td>
<td>A package manager for installing Python packages.</td>
</tr>
<tr>
<td>UNIX</td>
<td>A category of multi-user operating systems.</td>
</tr>
<tr>
<td>Linux</td>
<td>A Unix-like open source operating system.</td>
</tr>
<tr>
<td>Debian</td>
<td>A Linux operating system distribution.</td>
</tr>
</tbody>
</table>
1.3.2 ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>MIT</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>TU/e</td>
<td>Eindhoven University of Technology</td>
</tr>
<tr>
<td>URD</td>
<td>User Requirements Document</td>
</tr>
<tr>
<td>SRD</td>
<td>Software Requirements Document</td>
</tr>
<tr>
<td>ADD</td>
<td>Architecture Design Document</td>
</tr>
<tr>
<td>DDD</td>
<td>Detailed Design Document</td>
</tr>
<tr>
<td>SUM</td>
<td>Software User Manual</td>
</tr>
<tr>
<td>UPT</td>
<td>Unit Test Plan</td>
</tr>
<tr>
<td>ATP</td>
<td>Acceptance Test Plan</td>
</tr>
<tr>
<td>CSV</td>
<td>Comma Separated Values</td>
</tr>
<tr>
<td>ICU</td>
<td>Intensive Care Unit</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
</tbody>
</table>

1.4 LIST OF REFERENCES


[8] The Pyoneers (2016), Unit Test Plan version 1.0.0

1.5 OVERVIEW

The remainder of this document consists of seven chapters regarding the software transfer phase for Callisto.
Chapter 2 gives reasoning as to why there are no build procedures for Callisto.
Chapter 3 describes an example development environment setting and installation procedures for the Callisto plug-in and its immediate environment such as Jupyter Notebook and Jupyter-hub as well as the MIMIC-III database.
Chapter 4 gives an overview of all configuration items transferred.
Chapter 5 gives a summary of the performed acceptance tests.
Chapter 6 contains a list of issues regarding missing functionality or issues still in the software.
Chapters 7 and 8 are concerned with change requests and modification reports for the software, which are not applicable for this project.
2 BUILD PROCEDURE

No build procedure is necessary, because of the use of interpreted languages, namely Python and JavaScript.
3 INSTALLATION PROCEDURE

For all the commands outlined below, $ is used to denote that the command can be run under a non-privileged user, while # is used to denote that the command has to be run as a privileged user (for example, by using the sudo utility). Due to software compatibility problems, it is assumed that both development and deployment are done in a Unix-like environment.

3.1 DEVELOPMENT

Note that the MIMIC-III database needs to be setup either on a separate server or locally on the developer’s machine if results from the database are wanted. For setting up the MIMIC-III database on a server separately from jupyterhub and the Callisto plugin, please follow the instructions outlined in section 3.3.

Setting up the development environment includes the following steps:

1. Install Python v3.4.2
2. Install Pip v9.0.1
3. Install virtualenv v15.0.3
4. Install virtualenvwrapper v4.7.2
5. Install Git
6. Install IDE or text editor (for example, PyCharm)
7. Start a terminal and make a new virtualenv project with python3 using:

   $ mkproject --python=python3 <project_name_here>

   where <project_name_here> is the name of the directory where the Callisto project repository will reside.

8. Move the Callisto project to the directory created in the previous step

9. Set the current environment to be the one setup in the previous steps using:

   $ workon <project_name_here>

10. Install the Callisto project by going to the Callisto project directory and using pip as such:

    $ cd Callisto

    $ pip install . --upgrade --process-dependency-links

11. Navigate outside of the directory and back to the virtualenv project directory using:
12. Create and navigate to a new directory that will be used as a Jupyter Notebook start directory using:

```
$ mkdir -p notebooks
$ cd notebooks
```

13. Initialize the Callisto configuration and extensions using:

```
# callistoctl init
```

14. Modify the Callisto configuration file residing at /etc/callisto/callisto.conf with the database settings for the server hosting the MIMIC-III database.

15. Test the connection using:

```
# callistoctl test
```

16. Start Jupyter Notebook using:

```
$ jupyter notebook
```

17. Default browser starts with the application running on http://localhost:8888/

### 3.2 DEPLOYMENT SERVER WITH BOTH MIMIC-III AND JUPYTER HUB

For the next set of instructions, it is assumed that the operating system used on the server is Debian 8.6 Jessie (AMD64). The installation instructions for which can be found here https://www.debian.org/releases/stable/amd64/

1. Start a terminal and install all the dependencies for JupyterHub, Jupyter Notebook and Callisto using:

```
# apt-get install -y python3 python3-setuptools curl git
postgresql postgresql-server-dev-all pkg-config libfreetype6
libfreetype6-dev libpng++-dev libpng3 libffi6 libffi-dev
libblas-dev liblapack-dev libatlas-base-dev gfortran python3-
dev libxml2-dev libxslt-dev wget texlive texlive-latex-
recommended texlive-latex-extra texlive-fonts-extra
```

2. Create a new directory where the MIMIC-III database files will reside, for example /srv/csv as follows:

```
# mkdir -p /srv/csv
```
3. Navigate to the directory created in the previous step and download the MIMIC-III database files using:

   # cd /srv/csv
   # wget --user YOURUSERNAME --ask-password -A csv.gz -m -p -E -k
   -K -np -nd https://physionet.org/works/
   MIMICIII_ClinicalDatabase/files/

   where YOURUSERNAME is the username that you have used on PhysioNet. When prompted for your password, enter it.

4. Install Pip v9.0.1+ using:

   # easy_install3 -U pip

5. Install Node.js v6.x with NPM using:

   # curl -sL https://deb.nodesource.com/setup_6.x | bash -
   # apt-get install nodejs

6. Install an HTTP proxy for JupyterHub using:

   # npm install -g configurable-http-proxy

7. Navigate to the directory where the Callisto plug-in resides, for example /srv/callisto and install it using:

   # cd /srv/callisto
   # pip install . --upgrade --process-dependency-links

8. Switch to the postgres user and start the psql tool to login to the running PostgreSQL server using:

   # su - postgres
   $ psql postgres

9. Add a new user to be used for querying the MIMIC-III database by Callisto, using the following in the running psql session:

   CREATE USER YOURUSERNAME WITH PASSWORD ''YOURPASSWORD;

   where YOURUSERNAME is the username of your choice for the database user, and YOURPASSWORD is the password that will be used for that user.

10. Set your newly created user as superuser temporarily, using the following in the running psql session:

    ALTER USER YOURUSERNAME superuser;
11. Create a new database where the MIMIC-III data will reside, using the following in the running psql session:

```
CREATE DATABASE YOURDATABASE OWNER YOURUSERNAME;
```

where `YOURDATABASE` is the name of the database of your choosing, and `YOURUSERNAME` is the username that you’ve chosen.

12. Exit the psql session and from the postgres user login session by using:

```
\q
$ exit
```

13. Initialize the Callisto extensions, configuration and Jupyter Hub configuration using the callistoctl utility as follows:

```
# callistoctl init
```

the statement 'Initialization successful' should be present at the end of the output.

14. Modify the Callisto configuration file at `/etc/callisto/callisto.conf` using an editor (such as nano), similar to the following:

```
# nano /etc/callisto/callisto.conf
```

and modify the settings to match the selected PostgreSQL settings. The file should look like this:

```
{
  "database": {
    "database_name": "YOURDATABASE",
    "hostname": "localhost",
    "password": "YOURPASSWORD",
    "port": 5432,
    "user": "YOURUSERNAME"
  }
}
```

15. If SSL support is desired for Jupyter Hub, as is the case when Jupyter Hub is used alone without a reverse proxy such as nginx, the Jupyter Hub configuration file at `/etc/callisto/jupyterhub/jupyterhub_config.py` should be edited and the following lines should be added:

```
c.JupyterHub.ssl_key = '/path/to/my.key'
c.JupyterHub.ssl_cert = '/path/to/my.cert'
```

where `/path/to/my.key` is a path to the SSL certificate’s private key file and `/path/to/my.cert` is the path to the SSL certificate itself.
16. Migrate the downloaded MIMIC-III database files to the PostgreSQL database created using the callistoctl utility as follows:

   # callistoctl migrate /srv/csv

where /srv/csv is the directory where the files were downloaded. This should be changed to the directory of one’s choice. Note that this step might take a very long time due to the size of the MIMIC-III database, over 10 hours. The last statement in the output should be ‘Migration successful’.

17. Ensure that the MIMIC-III database has been migrated successfully and that the database settings at /etc/callisto/callisto.conf are correct by using:

   # callistoctl test

The last statement in the output should be ‘Testing successful’.

18. Get into an interactive psql session under the postgres user as follows:

   # su - postgres
   $ psql postgres

19. Revoke the superuser privileges from the chosen database user by running the following in the open psql session:

   ALTER USER YOURUSERNAME nosuperuser;
   \

20. Exit the postgres user login session:

   $ exit

21. Run the Jupyter Hub with Callisto instance using the callistoctl utility as follows:

   # callistoctl run --ip 0.0.0.0 --port 443

where 0.0.0.0 is the IPv4 address that the Jupyterhub instance will listen on, by default that is all interfaces and 443 is the port that the instance will listen on, by default that is the port used for HTTPS services. If SSL is not desired, you can add the --no-ssl flag to the end of the previous command.

22. Jupyter Hub with Callisto support should be running on the server at the IP and port number specified.

23. To stop the Jupyter Hub instance, the callistoctl utility may be used to do so by running:

   # callistoctl stop
24. The status of the Jupyterhub instance may be queried through the callistocctl utility as follows:

```
# callistocctl status
```

### 3.3 MIMIC-III

For installation of the MIMIC-III database on a separate server, please follow the instructions at: http://mimic.physionet.org/tutorials/install-mimic-locally-ubuntu/
4 CONFIGURATION ITEM LIST

In this chapter all the items that are delivered are listed. The documents are delivered in PDF format. For the software, the source code is delivered.

4.1 DOCUMENTATION
1. URD [2]
2. SRD [4]
3. ADD [3]
4. DDD [5]
5. SUM [6]
6. STD [7]

4.2 TEST PLANS
1. UTP [8]
2. ATP [9]

4.3 SOFTWARE
1. Callisto plug-in
5 ACCEPTANCE TEST REPORT SUMMARY

Description: All tests as described in the ATP version 1.0.1 [9] have been executed during the acceptance tests.

Activity and event entries: The first acceptance test was performed on October 31, 2016 from 8:30 to 10:30. During the first acceptance test, all tests except for ATB1, ATB2 and ATB3 passed. Attendees were the customer, the project supervisor, the whole project team and the project managers. A second acceptance test for tests ATB1, ATB2 and ATB3 was scheduled on November 3, 2016 from 8:30 to 9:00.

The second acceptance test was performed on November 3, 2016 from 8:30 to 9:30. Acceptance tests ATB1, ATB2 and ATB3 passed. Attendees were the customer, the project supervisor, the project managers and the project team members: J.W. Sleijster, K. Nassar and J.N. Adegeest.
6 SOFTWARE PROBLEM REPORTS

6.1 MATLAB GRAPH EXPORT
The MATLAB graph export bundlerextension was not implemented due to time constraints. There is partial code available for only exporting scatter plots. On the other hand, exporting all other graph types to MATLAB code is not implemented. The partial code is available in the code repository but is left out in production for consistency.

6.2 OVERLAPPING DESCRIPTION TAG WITH X-AXIS LABELS
In some bar charts, the X-axis labels overlap visually with the description tag of the chart. There was a suggested fix for this problem, however, it introduced another problem which manifested in X-axis labels wrapping around on other lines. Due to time constraints, it was not possible to find a solution that fixes the issue without unwanted side effects.

6.3 SLOW PRESET: AGE VS. DISEASE CLASS VS. AVERAGE ICU STAY
For one query preset, namely the preset of Age vs. Disease Class vs. Average ICU Stay, the execution time is around 2 and a half minutes. This is due to the size of the table of ICU stay records from the MIMIC-III database. An optimization could not be done in the time-span of this project, but future work could be done to optimize the basic query that is used for that preset.
7 SOFTWARE CHANGE REQUESTS

Not applicable
8 SOFTWARE MODIFICATION REPORTS

Not applicable