Deliverable information

WP NO.	WP 6
DEL.	D6.1
DEL. NO.	D16
TITLE	Research and development environment
DESCRIPTION	The creation of a coding technical infrastructure that is scalable and extendable in a modular approach. The basis for the infrastructure will be open-source projects like R which gives access to developed machine learning projects like Tensorflow, PyTorch, MXNet and H2O. These research and development environments will be made available in a dedicated cloud server environment to manage the code, scripts, GUIs, models, users' access rights, software interaction and workflows.
NATURE	Websites, patents filling, etc.
EST. DEL. DATE	31 Mar 2019

Document information

DATE	28/03/2019
WRITTEN BY	Firamis GmbH
APPROVED BY	Arianna Agosto

Actions and achieved results

We have developed and built up a coding technical infrastructure that serves as the central platform for all coding related content of the FIN-TECH project. All project-participants, meaning consortium-partners as well as regulators and fintechs, are able to download or upload content (depending on their access rights). The platform is solely based on open-source software modules and is designed in a modular approach. Therefore the platform can be modified or expanded during the project life-time and afterwards, depending on the feedback and the future needs of the users.

The platform is designed in line with the EU General Data Protection Regulation (GDPR), as it can only be accessed by registered users. After the first registration personal data is processed.

The login-button on the project-website https://www.fintech-ho2020.eu serves as the entrance gate to the platform. All staff of the consortium-partners who are supposed to publish relevant content for the FIN-TECH project need access to the platform. The staff of regulators and fintechs who take part in suptech or regtech workshops, have access to the platform but not to a separate file management tool that is reserved to partners only. This is where the platform contents are prepared collaboratively by the partners before made available for the regulators and fintechs.

In order to register for a suptech or regtech workshop, a user has to reveal name, email address, affiliation, department, position and role in special forms that were designed for the project. This data is saved in an excel-list on the project-server at BadenIT GmbH. It is extended by user passwords that were generated with a password generator. There is also a script that runs automatically every minute in order to collect all data from all running forms and aggregates them on an overview list.

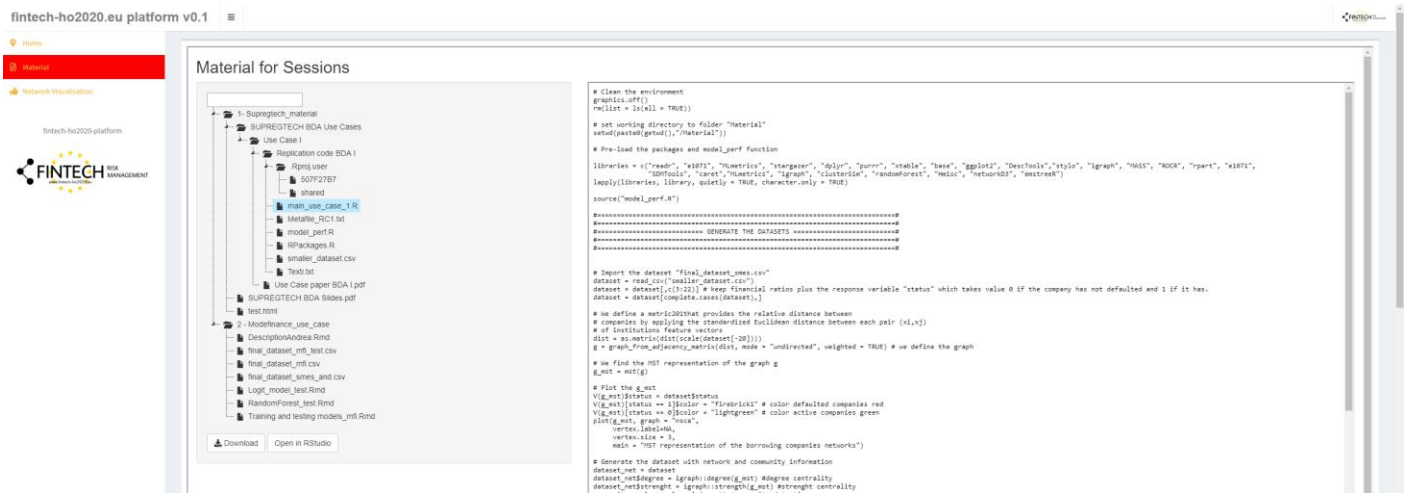
Using a self-written script, the relevant data is imported to LDAP, an application protocol for accessing and maintaining distributed directory information services over an Internet Protocol network. This program is necessary for allowing the sharing of the project-content with regard to different access rights of the users (only the staff of the consortium-partners has the right to upload content in the file management tool, whereas the staff of regulators and fintechs is only allowed to download/read content on the platform).

As the final step we have developed a fully automated process in order to send the relevant email address and the corresponding passwords (necessary for registering to the platform) to the single user. We use the open source based email-program Thunderbird for this step. The automatically generated (and personalized) email contains all information the user has to receive with regard to Art. 13 GDPR.

Detailed platform description:

The underlying technology of the platform is based on the open-source R packages “shiny” and the open-source IDE “Rstudio” that can we run in a browser. The main purpose of the platform is to provide and run code and to show corresponding papers, markdowns and other formats for users or for executing regtech and suptech workshops.

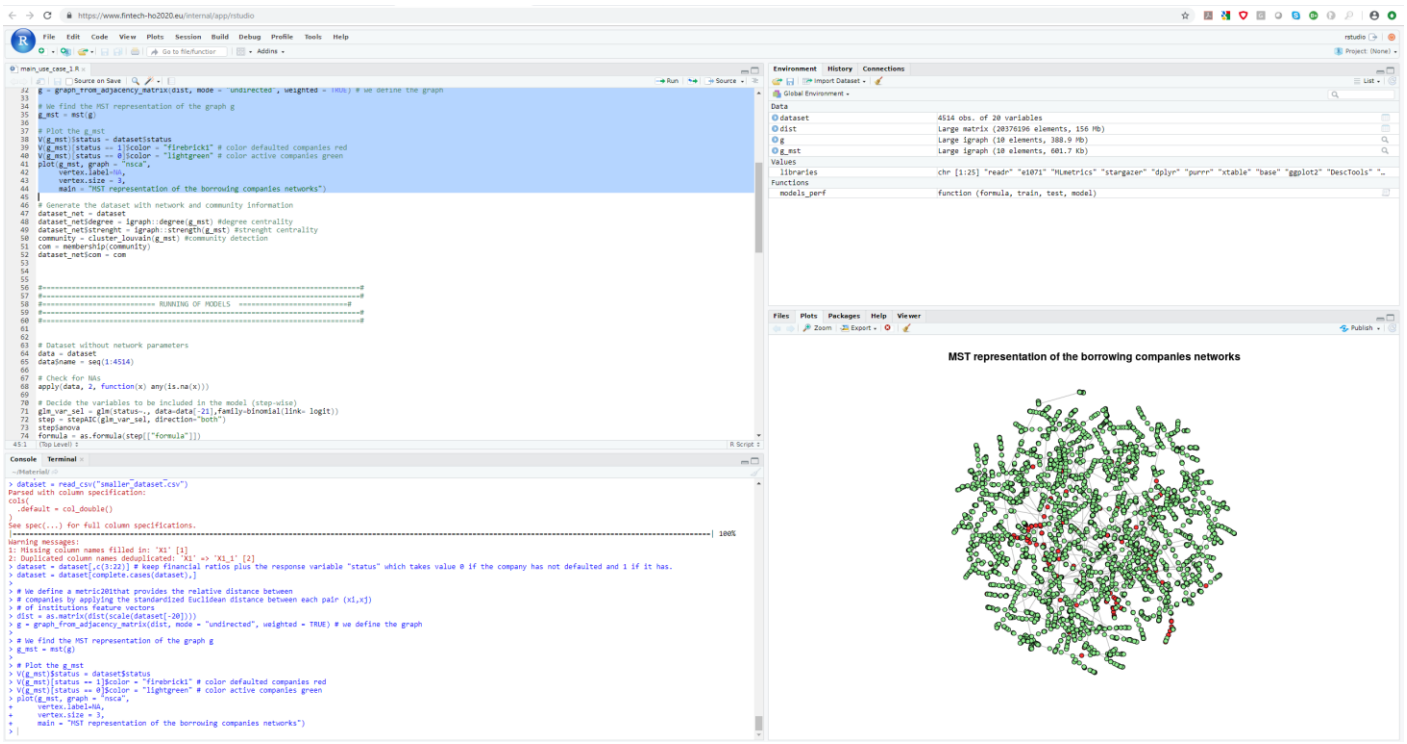
The following screenshot shows the first window that opens up after the login to the platform:



In the area "Material for Sessions" the platform user can select between different (session) materials that were published by consortium partners for their suptech or regtech workshop. This tree like structure is the same as the one in the file management where the partners work on this structure collectively. New versions of the file structure from that file management tool can be pushed to the platform where they are then available to the users.

The user has the possibility to display the content of the single files in the right area of the window or to download the file/material on his computer. If the material contains R or Rmd code, this code can also be shown in an Rstudio IDE dashboard (see next screenshot).

The following graphic is a screenshot of the R-Studio dashboard that is used during the first regtech workshop in Milan on March 2019, 29th:



In the top left area the relevant R-code is shown as a script:

The R code is compiled and can be run:

- on the whole (by selecting all the script)
- line by line (by selecting only the line or by placing the cursor in the correspondent line of the script)
- as a given part of the code inside a line (by selecting such part of the code).

The result of running the code is shown either in the R console (bottom left area) or in the graphical output window (bottom right area). If the code includes the creation of any object, it is listed in the R working environment window (top right area).

The R console (bottom left area) has the same functionalities as the R-Gui (graphical user interface). It has only one editing line and the results of the calculation are displayed in the same window. The error messages and other information related with the script running process are also shown in the R console. The R console is very useful for testing code before implementing it into the script.

The R environment window (top right area) shows all the objects that were created in R and that are active to be used. It provides a quick overview of not only the number and name of the objects in the working environment, but also about the information stored and the type of the object.

In the graphical output (bottom right area) the graphs are represented as default. The figures can also be represented in external devices by request. In this window the help about the core functions or the implemented in each library is displayed.

It is also possible to integrate web-links to code git repositories as a file with extension ".link" in the file structure. When clicked on those files in the tree structure the user gets a Rstudio browser tab with the linked git project opened and ready to be executed.

The required R libraries for the standardized code examples available in the file tree structure are prepared in a customized installer script before they are made available to the platform application. In this way, users need only a few clicks to have the code ready to be launched in a live coding or self-learning session.

The modular approach we chose for developing and building up the platform allows us to further modify or expand the platform during the project duration. The open-source and user community of our two main software building blocks (Rshiny and Rstudio) is already very large but still growing. This helps us to permanently improve and expand the project platform and the tools for dissemination and for support of the coding session.

Besides the event registration forms we have a number of evaluation forms for supervisors, regulators, fintechs and other evaluators of the syllabus, content and code material of the project.

Here are some potential next steps for implementation. The modular structure of the platform is helpful as most of the suggested tools should be able to be connected and combined with the platform

1. We are considering to use the R package reticulate which is able to run Python code besides the R code in Rstudio. If we are able to deploy customized R/Python-environments we could be able to also run Python codes in the same environments and even run mixed code examples with R and Python involved.
2. We could set up a community or communication tool like slack also for the non-partners. There are open-source packages that can be installed on the BadenIT server where the platform already runs. These can be connected to the same authentification (LDPA) that is already set-up for the platform.
3. For building the paper repositories we consider to use an R-package like "bibliometrix" as a tool for browsing and analyzing corresponding scientific paper collections which are collected and maintained by the work package leaders and the other partners.
4. Complex data and algorithmic visualization. The built-in access to javascript in Rshiny allows to get exposure to web-visualization technologies that can be connected to the R models and kernels by interactive dashboards. This supports advanced data analytics, teaching and dissemination of the project activities.