Grant Agreement No.: 823783
Call: H2020-FETPROACT-2018-2020

Topic: H2020-FETPROACT-2018-01
Type of action: RIA

D6.1 WENET PLATFORM ARCHITECTURE SPECIFICATIONS

Revision: v.1.0

Work package	WP 6
Task	Task 6.1
Due date	31/12/2019
Submission date	15/01/2020
Deliverable lead	UH
Version	1.0
Authors	Bruno Rosell i Gui (CSIC) Theofrastos Mantadelis (OUC) William Droz (IDIAP) Roberto Bona (UNITN) Avi Segal (BGU)

	Ronadl Chenu (UNITN) Sergio De Cristofaro (UNITN) Carlo Caprini (UH) Daniele Miorandi (UH)
Reviewers	Federico Facca (MARTEL)

Abstract	This deliverable describes the initial architecture of the WeNet platform, which will be used to support activities in the project pilot sites. Platform components are identified, and for each component functionality and APIs are detailed.
Keywords	platform, architecture, software

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	27/09/2019	Initial structure	Daniele Miorandi (UH)
v0.2	11/10/2019	First content in	Bruno Rosell i Gui (CSIC) Roberto Bona (UNITN) Carlo Caprini (UH) William Droz (IDIAP) Theofrastos Mantadelis (OUC)
v0.3	23/10/2019	Additional content added	Bruno Rosell i Gui (CSIC) Carlo Caprini (UH) Theofrastos Mantadelis (OUC) Avi Segal (BGU)
v0.4	9/11/2019	Additional content added and content aligned	Bruno Rosell i Gui (CSIC) Carlo Caprini (UH) William Droz (IDIAP) Theofrastos Mantadelis (OUC) Ronald Chenu (UNITN) Sergio De Cristofaro (UNITN)
v0.5	22/11/2019	Added missing sections, migrating to official WeNet template	Daniele Miorandi (UH)
v0.6	27/11/2019	Added executive summary and Section 12.	Daniele Miorandi (UH)
v0.7	4/12/2019	Content revised, sent for internal quality assessment	Daniele Miorandi (UH) Carlo Caprini (UH) Sergio De Cristofaro (UNITN) Bruno Rosell i Gui (CSIC) William Droz (IDIAP)
v0.9	12/1/2020	Content revised after reception of internal quality assessment	Daniele Miorandi (UH) Carlo Caprini (UH) Sergio De Cristofaro (UNITN) Bruno Rosell i Gui (CSIC) William Droz (IDIAP), Manos Seferis

Co-funded by the Horizon 2020 Framework Programme of the European Union

			(OUC), Avi Segal (BGU)
v1.0	15/01/2020	Content polishing	Daniele Miorandi (UH)

DISCLAIMER

COPYRIGHT NOTICE

Project co-funded by the European Commission in the H2020 Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	Public, fully open, e.g. web	✔
CL	Classified, information as referred to in Commission Decision 2001/844/EC	
CO	Confidential to WeNet project and Commission Services	

** R: Document, report (excluding the periodic and final reports)*

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

OTHER: Software, technical diagram, etc.

EXECUTIVE SUMMARY

The WeNet project is developing an IT platform able to support the rapid prototyping and deployment of applications able to effectively leverage user diversity; the primary role of the platform is to support the activities to be carried out with users in the project pilot sites.

This deliverable describes the structure and architecture of the WeNet platform. The platform has been designed as a set of modular, reusable and loosely coupled components. Each component exposes a well-defined set of functionality. For each component, a set of application program interfaces (APIs) are defined and presented.

The deliverable includes an example of how applications can leverage the platform functionality and APIs to provide value-added services to end users. Said example is based upon uHelp, a social network application developed by a WeNet consortium partner to facilitate people to ask for and receive help by the relevant community.

The deliverable also comprises the initial description of the architecture foreseen for the project's research infrastructure, which will be used to store and share research data collected throughout experimental activities on the pilot sites.

The specifications included in this deliverable represent the starting point for the actual implementation and integration of the WeNet platform. Version 1.0 of the platform will be released at M18 as D6.2

Co-funded by the Horizon 2020
Framework Programme of the European Union

TABLE OF CONTENTS

Co-funded by the Horizon 2020 Framework Programme of the European Union

Co-funded by the Horizon 2020
Framework Programme of the European Union

LIST OF FIGURES

Co-funded by the Horizon 2020
Framework Programme of the European Union

LIST OF TABLE

Co-funded by the Horizon 2020
Framework Programme of the European Union

ABBREVIATIONS

ML	Machine Learning
API	Application programming interface
GDPR	General Data Protection Regulation (EU - 2016/679)
DoA	Description of the Action - Annex 1 of the WeNet Grant Agreement
IT	Information Technology
AI	Artificial Intelligence
WP	Work-package

Co-funded by the Horizon 2020
Framework Programme of the European Union

1. INTRODUCTION

WeNet starts from the assumption that diversity is a key societal value for the future of Europe and that, if put at the centre of a platform design, can lead to sociotechnical systems that allow to connect people to achieve everyday life goals while respecting their differences and embodying fundamental features of transparency, fairness, and accountability.

Ultimately, in WeNet, this translates into the design, implementation and validation of a set of algorithms enabling diversity-awareness and diversity-centric applications. WeNet blends together fundamental work on science with an experimentally-driven approach, whereby solutions are co-designed with users on a large number of pilots, and then developed and tested.

In the WeNet project concept, a primary role is played by the so-called WeNet platform. The WeNet platform is a software artefact, which integrates the implementation of diversity-centric AI algorithms into a cohesive system, which can be leveraged by developers to prototype and test new applications and services. In particular, the WeNet platform will be used to support the activities on the project pilot sites, by powering the 'WeNet app' that will be used to empower the students that will take part in the pilots.

From the formal standpoint, in the DoA, the goal of WP6 in WeNet is described as "*to integrate and consolidate the methods and systems designed in Wp2-Wp5 into an online platform and a mobile app ready to support pilot trials*". WP6 acts therefore as 'system integrator' in the WeNet project. The two expected final results of WP6 are a software platform, allowing developers and innovators to rapidly prototype and deploy diversity-aware applications and services (the 'WeNet platform'), and a specific app (the 'WeNet app'), which will be used in the project pilot sites.

The WeNet platform will include a set of technological enablers, allowing developers to implement applications able to *"effectively empower interactions among individuals leveraging diversity"*.

This deliverable describes (as per DoA) the "(WeNet) *platform architecture and components interface specifications*". It targets a technical audience, both within and outside the Consortium. In particular, it will be used to drive the implementation of the first version of the platform, which will be released at M18 as D6.2.

The overall platform design was mainly informed by the work carried out within WP7 on the pilot scenarios; the work on component-level specifications was carried out together with the leaders of WP2/3/4/5.

The API specifications have been written according to the OpenApi v3.0 open standard[1]. They will be continuously updated and maintained on the project git repository, hosted by UH. Project partners can interact with the specifications using a purposeful Swagger UI[2]. Data models are not reported in the deliverable for the sake of space; they are available in the project git repository.

The remainder of the deliverable is organised as follows. In Sec. 2 we introduce the high-level platform requirements, and describe a sample application, based upon the uHelp app [1], which will be used throughout the deliverable to ground and better explain how the functionality

[1] https://www.openapis.org/
[2] https://swagger.io/tools/swagger-ui/

Co-funded by the Horizon 2020 Framework Programme of the European Union

provided by the platform can be used to power applications. In Sec. 3 we describe the high-level architecture of the platform, including its components and their role. We then detail, for each component, its functionality and a set of APIs used to expose said functionality (Sections 4-12). In Sec. 13, we describe how the platform can be used to enable the functionality provided by the uHelp app. We do also include in the deliverable (Section 14) some early-stage results on the design of the WeNet research infrastructure. Section 15 concludes the deliverable presenting an outline of the roadmap until the first platform release (M18, Deliverable D6.2).

Co-funded by the Horizon 2020
Framework Programme of the European Union

2. MOTIVATION

2.1. THE UHELP APP

uHelp[3] is a social network application, developed by WeNet partner CSIC to facilitate people to ask for and receive help [1]. It allows people in everyday life to source things such as a babysitter or someone to lend you chairs for a party. The added value of uHelp is that for each request, it is the user who determines the level of trust and whether that person has to be one of your uHelp contacts, a friend of a friend, or those at any connection level. This way, only those that match the requirements will receive the call for help.

The application is built on an intelligent volunteer search based on two principles: connection and trust. This allows users to create a community of mutual assistance with their friends, friends of friends and so on, building up a large social network that addresses people's needs.

uHelp represents a good starting point for checking how the WeNet platform can empower diversity-aware applications. While the actual WeNet pilots will not use directly WeNet (modulo some early trials for collecting valuable data on community dynamics), the scenarios that are currently being developed in the project [8] share a lot of key features, like the notion of asking for help/support, the notion of community (and related norms) and the importance of trust.

2.2. WENET PLATFORM HIGH-LEVEL REQUIREMENTS

Based upon work carried out in WP7 (partially reported in [8]), the following set of high-level requirements were elicited, which were used to guide the overall platform design and architecture specifications. We used the guidelines in [9] for the use of 'must', 'should', 'shall' and 'may'.

- Functional requirements:
  - The platform must support multiple applications.
  - A user should be able to use the same authentication credentials across multiple WeNet-powered apps.
  - A user should be able to define which profile components are public and which ones are private.
  - A user should be able to connect to the platform multiple data sources (including, but not limited to, mobile apps monitoring user location and using other phone sensors, calendar, social media account), which can be used to enrich the user profile.
  - The platform may use data from sources connected by the user to derive user habits, routines, preferences, and information about the user social network.
  - The platform shall include machine learning methods and algorithms for deriving user habits, routines, preferences, and information about the user social network.
  - The platform should support the definition of norms (at the individual, community and task level), which regulate how the interactions among the users take place and guide some aspects of the platform behaviour.
  - The platform must support the exchange of messages among users of the same app (possibly mediated by a set of norms).
  - The platform must allow users to create tasks.

[3] https://uhelpapp.com/

Co-funded by the Horizon 2020 Framework Programme of the European Union

- The platform must disseminate tasks created to potential volunteers (possibly mediated by a set of norms).
- The platform must allow users to volunteer for tasks.
- The platform should allow users who originally created a task to choose among volunteers.
- The platform must handle the whole task lifecycle.
- The user profile must include dimensions able to characterise the diversity of platform-powered apps.
- The platform shall provide methods for motivating users to contribute to a task.

- Non-functional requirement
  - Users must be in control of their personal data.
  - Users may be able to log in the platform by using their social media account.
  - Personal data must be protected from unauthorised access and usage.
  - The platform must be GDPR compliant.
  - The platform should be able to scale to at least tens of thousands of simultaneous users.
  - The platform may support a distributed deployment mode.
  - The platform should log interactions among users.
  - The platform should collect logs from components for monitoring and troubleshooting purposes.
  - Platform functionality must be exposed using open standards, and application developers should not be bound to a specific technology/framework/language for interacting with the platform.

3. THE WENET PLATFORM

3.1. FUNCTIONALITY AND INTENDED USAGE

The WeNet platform consists of *a set of modular, interoperable, open software components providing the required functionality to run the WeNet pilots*. Therefore, the platform functionality is driven by the requests coming from the pilot sites. The platform should therefore be seen as a tool (not by a goal in itself) for allowing the easy development, deployment and replication of applications across different pilots.

From the conceptual standpoint, as graphically represented in Figure 1 below, the WeNet platform will support multiple applications (be them mobile, web or chat apps), which will leverage the functionality exposed by the platform. The platform will ingest multiple data streams, which will be achieved by connecting the platform to third-party services and applications, including but not limited to, mobile phone sensors (through a purposeful mobile app), calendar applications, social media streams, open data portals etc.

Co-funded by the Horizon 2020 Framework Programme of the European Union

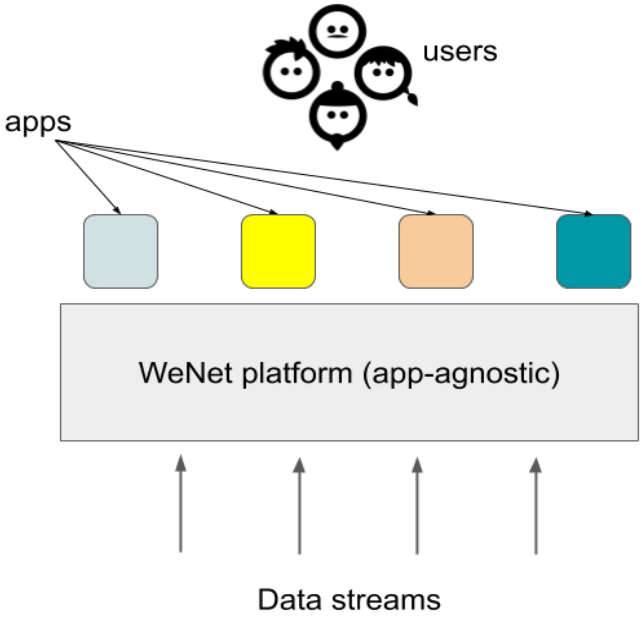


FIGURE 1: WENET PLATFORM AND APPLICATIONS

The platform will be online, and will expose a set of RESTful APIs. Calls to the platform need to be authenticated. The platform is based upon a number of loosely coupled components. Each single component will expose RESTful APIs; all components will communicate via HTTP(s). Also single components need to authenticate, for both allowing access control as well as the possibility of reconstructing an audit trail (provenance).

Ownership of components to WeNet partners is clearly defined. We have decided not to be prescriptive in terms of how single partners should develop their own components (in terms of programming languages, frameworks and libraries to be used), provided that decisions that may influence the exploitation of the platform as a whole (e.g., usage of copyleft-licensed software libraries) are coordinated with the WP6 leader. All components will be dockerized to ease deployment for the pilot sites.

At the moment, no decisions have been taken with respect to deployment, i.e., whether to go for a centralised vs. distributed deployment, and whether to go for single-tenant vs. multi-tenant deployment. We have taken careful design decisions to ensure that the resulting software is flexible enough to accommodate with modest effort all possible options. As the planning of pilot sites progresses, we will choose whatever fits best the pilot constraints (including those coming from non-European partners).

All in all, the platform design has been (and will be) driven by pragmatism: we need to have something that allows us (as whole project) to easily accommodate experiments; this is our highest priority as WP6 and is the `true North' of our activity in the project.

The expected users of the platform are application developers. Developers will need to register with the platform and get an *app key - app secret* pair, which will be used by the app for authenticating and authorizing calls to the platform APIs.

The WeNet platform provides single sign-on functionality. Users can register on the platform and obtain a WeNet user ID. They can then use such WeNet user ID to log into different apps that use the WeNet platform functionality for authenticating users. This is consistent with the long-term vision of creating an eco-system of applications and a

Co-funded by the Horizon 2020
Framework Programme of the European Union

community of developers that use the WeNet platform for creating diversity-aware, value-added applications for individuals and communities.

3.2. STRUCTURE AND ARCHITECTURE

Based on the high-level requirements in Sec. 2 and the general design guidelines in Sec. 3.1, we have identified eleven (11) different components for the WeNet platform, which are graphically represented in Figure 2. Here the list of the platform components together with the consortium partner responsible for its design, implementation and maintenance.

1. Service APIs - U-Hopper
2. Interactions Logger and Analytics - U-Hopper
3. Authentication Server - U-Hopper
4. Task Manager - U-Hopper
5. Interaction Protocol Engine - CSIC
6. Incentive Server - BGU
7. Social Hub / Profile Manager - CSIC
8. Personal Context Builder - Idiap Research Institute
9. Social Context Builder - OUC
10. ILogBase - University of Trento
11. Common APIs - University of Trento

The WeNet Service APIs are the ones application developers can leverage to build diversity-aware apps. In particular, the Service APIs allow to manage Applications, Users, Profiles, Messages and Tasks. Through them, a developer can register her own app on the platform, receiving the credentials for authenticating the requests and univocally associating them to her own app. Through them, users can create an account and login using a set of existing credentials. Through them, an application can access a user profile (provided that the user endorsed such request). Through them, users can exchange messages. Through them, users can create tasks, volunteers for open tasks and contribute to them.

The Commons APIs are used to collect data that can be used in order to enrich the user profile, by means of dedicated connectors. WeNet Commons can be used to connect, for example, to a third-party mobile application feeding measurements on the current location of the user, but also to third-party services such as Twitter or Google Calendar, allowing access to tweets and appointments.

The Task Manager is the component responsible for managing tasks and maintaining their status. A task collects all the information describing the help request posted by a user on the platform and tracks the various steps necessary for the requester to evaluate volunteers, choosing one to complete his request and verifying the successful completion of the task.

The Interactions Logger and Analytics plays a twofold role. First, it collects the messages exchanged within the users' conversations as well as the platform components' log messages. With its message logging functionality, this component has a key role in the

Co-funded by the Horizon 2020 Framework Programme of the European Union

project as it allows to analyse and audit past conversations, in particular for understanding the impact of the behaviour and configuration of the different platform components.
Second, this component also provides analytics both on a message and on a component log level. Such information can be used to monitor the intensity and content of the interactions, as well as to manage the status of the platform components.

The Authentication Server is responsible for authenticating and authorizing not only the requests coming from the applications built on top of the platform but also the communication between the various platform components. It is also responsible for authenticating the users connecting to the platform.

The Interaction Protocol Engine is the component responsible for managing the interaction between users while they are performing a task while guaranteeing that the users allowed to participate satisfy the norms that should guide the behaviour of the participants.

The Incentive Server is responsible for identifying those incentives that could trigger the user interest in volunteering and participating in a task and for maintaining users engaged over time.

The Social Hub (Profile Manager) is the component responsible for storing and maintaining the WeNet user profiles, while ensuring that access to the information is controlled and allowed only on an application basis. Also, we care a lot about user privacy, and for this, the profile manager allows any user to specify what and when it wants to share with other users and/or applications.

The Personal Context Builder is responsible for analysing the data collected by the various Common APIs connectors and identify relevant routines describing the users' habits.

The Social Context Builder is responsible for analysing the data collected by the various Common APIs connectors and learn information describing the social preferences, relations and interactions of users.

The iLogBase is a big data component, responsible for collecting in a scalable way data about the platform users and for feeding them to those components that have activated a feed subscription (or are periodically requesting for updates). The data may have been generated by another component within the platform or may be coming directly from one of the connectors of the Commons APIs.

Co-funded by the Horizon 2020
Framework Programme of the European Union

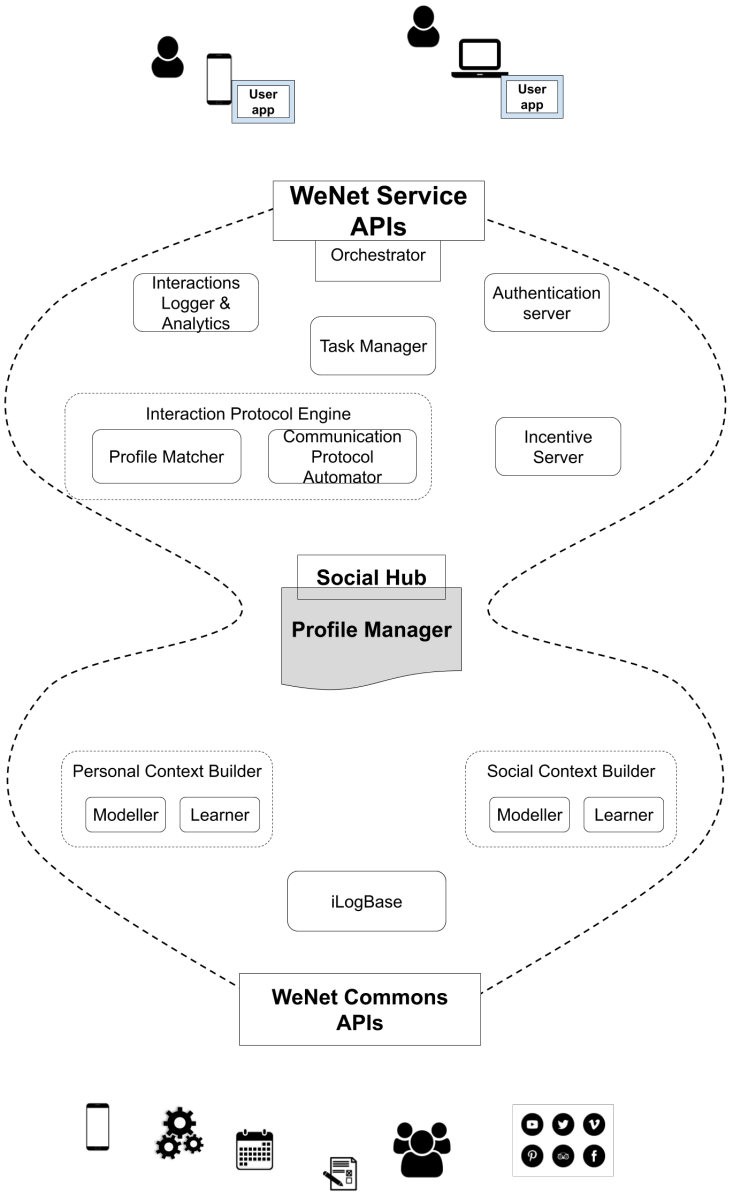


FIGURE 2: THE HIGH-LEVEL WENET ARCHITECTURE (LOGICAL VIEW)

3.3. SERVICE REST APIS

The Service REST APIs are responsible for exposing the platform functionality to the applications that are going to be built and developed during the four years of the project. In particular, the Service REST APIs allow to:

1. Manage WeNet applications;
2. Manage the authentication and creation of WeNet users;
3. Track incoming and outgoing messages;
4. Manage the message creation and forwarding while allowing users to discuss about an open task;
5. Manage tasks.

Co-funded by the Horizon 2020 Framework Programme of the European Union

The application endpoint allows a developer to register her own app, receiving the credentials for authenticating the app requests and univocally associating them to her own app

A WeNet user is defined at the platform level and exists cross-applications, making it possible to login into different WeNet applications, no matter their purpose, by using the same credentials. The platform is responsible for managing the user profile information in order to control which profile fields should be accessible to a given application.

The Service REST APIs allow to also track and monitor incoming and outgoing messages exchanged by users during their in-app conversations. Also, the platform will be responsible for creating new messages that will be of key importance for facilitating the conversation between users and the description of tasks.

A Task represents an activity that a user requires some assistance with. It may be something a user cannot complete autonomously or a task the requester is looking a substitute for.

The Service Rest APIs include an Orchestrator module that, for each incoming request, is responsible for forwarding and fetching the necessary information to and from the other components of the platform before producing a successful response.

Method	Path	Description
Application		
POST	*/app*	Create a new application
GET	*/app/{appId}*	Get the details of an application
GET	*/app/{appId}*	Update the details of an application
User		
POST	*/user/authenticate*	Authenticate a WeNet user
POST	*/user*	Create a new WeNet user
Profile		
GET	*/user/profile{wenetUserId}*	Get the profile of a user
PUT	*/user/profile{wenetUserId}*	Update the profile of a user
Message		
POST	*/messages*	Submit a list of new incoming messages
Task		
GET	*/task/{taskId}*	Get the details of a task

Co-funded by the Horizon 2020 Framework Programme of the European Union

POST	*/task/{taskId}*	Create a new task
PUT	*/task/{taskId}*	Update the details of a task

3.4. COMMON REST APIS

The Common REST APIs are used as the platform's southbound interfaces, as shown in Figure 2. These interfaces are wrappers of the iLogBase services (see below) for external clients. Their purpose is to allow the ingestion of low level data streams that will be later stored and processed within the WeNet platform.

The i-Log mobile application, developed by the University of Trento [7], is a first example of an external source of information that will be used in order to collect key data about the user habits. By taking advantage of the user's smartphone sensors, i-Log collects, among others, data about the user location, thus allowing WeNet to monitor the user movements and use such information to enrich the user profile.

The following APIs are used for acquiring information generated by third-party applications (such as i-Log, Twitter, Google Calendar, etc.). Support for NGSIv2 APIs[4] (NGSI-9/NGSI-10) will be added and NGSI-LD will be considered for inclusion in the next iteration by M18.

Method	Path	Description
Streams		
POST	*/streams*	Send a subscription request to receive data from
GET	*/streams/{sourceId}*	Get a list of active subscriptions for *sourceId*
GET	*/streams/{streamId}*	Get the details of an existing subscription
DELETE	*/streams/{streamId}*	Delete an existing subscription
Batch Data		
POST	*/data*	Upload a new batch of data
PUT	*/data*	Modify an existing set of data
GET	*/data/{userId}*	Retrieve data for a given user

[4] https://fiware.github.io/specifications/ngsiv2/stable/

Co-funded by the Horizon 2020 Framework Programme of the European Union

4. iLogBase

4.1. FUNCTIONALITY

The iLogBase is the main storage and repository for data streams within the WeNet platform. These streams include the support for:

- WeNet data streams captured directly from sensors and questionnaires enabled by the any WeNet consortium developed applications (e.g. the i-Log sensor and answer collection mobile application developed by the University of Trento [7])
- Third-party data streams generated by third-party services that provide key information about the WeNet users (e.g. social sources such as Twitter and Facebook streams): dedicated connectors will be developed for every external stream.

The iLogBase will also be responsible for storing the results of the data analysis processes applied on the enabled data streams.

Besides storing the information, the iLogBase is also responsible for making it available to the other components of the platform so that they can carry out their internal processes. For example:

- Machine Learning services (e.g., Personal Context Builder, Social Context Builder) shall use stream data from the iLogBase to feed their engines; in turn the results may be written back to iLogBase for other services to use;
- The Profile Manager shall use data from iLogBase, to generate and update the dynamic part of the user profile (e.g., current location).

To facilitate information access for the other components, the iLogBase offers two options. The first one is meant for real time data access and it is stream-based thanks to a publish/subscribe approach, allowing listeners to receive the data as soon as it is available. The second option is batch-based and it is meant for components which need a periodical access to the information (polling-based).

4.2. INTERFACES

Method	Path	Description
Streams		
POST	*/streams*	Make a subscription to receive info on properties every time some of the condition.properties change. The properties are specified in the body of the request.
GET	*/streams/{streamId}*	Get the details of an existing subscription

Co-funded by the Horizon 2020 Framework Programme of the European Union

GET	*/streams/{sourceId}*	Get a list of active subscriptions for a given source
DELETE	*/streams/{streamId}*	Delete an existing subscription
Batch Data		
POST	*/data*	Upload a batch of data
PUT	*/data*	Modify an existing set of data
GET	*/data/{userId}*	Retrieve data for a given user

5. Personal Context Builder

5.1. FUNCTIONALITY

The Personal Context Builder is responsible for building and maintaining users' routines. In WeNet, a routine describes user habits and is built by analysing the user data streams that are collected via the southbound Common REST APIs.

In particular, two different routine representations will be made available.

- The *Semantic routine* is a friendly routine representation that can be easily leveraged by the other WeNet components.
- The *Embedded routine* consists in the raw routine representation that matches the output of the machine learning algorithms applied for computing them.

The information generated on routines is pushed to the profile manager and maintains the profile constantly updated. A query interface is also foreseen for internal usage.

5.2. INTERFACES

Method	Path	Description
Embedded routines		
GET	*/routines*	Get the list all embedded routines for all users with all models
GET	*/routines/{user_id}*	Get specific user embedded routine

Co-funded by the Horizon 2020 Framework Programme of the European Union

GET	*/models*	Get the list of all available models for the embedded routine
Semantic routines		
GET	*/semantic_routines/{user_id}/{weekday}/{time}*	Get the semantic routine for a given user, weekday and time
GET	*/semantic_routines_transition/leaving/{user_id}/{weekday}/{label}*	Get the information about what when the user_id is leaving the label on the given weekday
GET	*/semantic_routines_transition/entering/{user_id}/{weekday}/{label}*	Get the information about what when the user_id is entering the label on the given weekday
Miscellaneous API		
GET	*/closest/{lat}/{lng}/{N}*	Get closest N users from a given point

6. Social Context Builder

6.1. FUNCTIONALITY

The Social Context Builder is responsible for building and maintaining the social details of a user profile, by leveraging the data collected by the various streams connected through the WeNet Commons APIs and by analysing them.

Having a model of the user's Social Context is crucial for assigning tasks to users. Because, after task experience (managed in the user profile), there are a lot of factors that affect a successful interaction of users through WeNet, such as friendship, common interests or views, personality types, etc. Summing up, user compatibility is highly important for ranking users to tasks, and this is what motivates us in creating this module.

The Social Context of a user will be represented as in the standard way of major social networks: There, a user follows some other users ("friends"), which can be either individuals, or groups (companies, interest groups, clubs, etc.). Similarly, the user is followed back by other users. Thus, the Social Context is represented as a directed graph between users, which is the standard practice in social network analysis [10]. Programmatic, there are many possible ways to implement such a data model.

The most important factor for us is being able to access the "strength" of a relationship, or the "compatibility" of two people. This is captured by the notion of "tie strength", [11]. Social relations are characterized as strong or weak ties, where strong ties indicate close relationships, and weak ties simpler acquaintances. Tie strength has been found to be dependent on various variables, such as time spent together, intimacy, reciprocal services, and others. An important study of these in the context of social

Co-funded by the Horizon 2020 Framework Programme of the European Union

networks can be found in [12], where the authors attempt to measure those variables from Facebook data, and then access their predictive power for tie strength.

Finding tie strength and user compatibility is an active area of research, and many ideas have been proposed in the literature since then. Recently, machine learning based methods have been employed for this task in all major social networks. We intend to investigate similar approaches in the context of WeNet's Social Context Builder. Additionally, another important factor is the ability to estimate the compatibility of two users who are not friends yet. Finally, balancing social ties and task experience is another challenging aspect, where further investigation is necessary. For example, social strength and trust is very important in the task of babysitting, while task experience might be more important in other cases.

Another idea, which we think is important in the context of WeNet, and which is not found in other Social Media is the ability to explain the decisions made by the builder to the user. For example, when the program ranks users for a task (say, baby sitting), we would like to be able to explain to the user why the program recommends that particular user for helping. We believe that such a system will be really helpful for the users, and promote the success of WeNet.

Having models able to explain their decisions is a very active area of research in Machine Learning and AI, and many approaches are being suggested. Some examples can be found in [13] for knowledge-based systems, or [14] for (statistical) ML models. A further investigation of the literature and research will be conducted in this aspect.

To summarize, the main objectives of the component will be the following:

- Identifying the user social relations (Alice is a friend of Bob). Such relations are going to be associated with a degree of confidence (e.g. how confident we can be about the existence of such a friendship) and with a degree of strength of that relation (e.g. how close friends Alice and Bob are), as mentioned previously.
- Identifying the user's social preferences in ranking users for a specific task (e.g., for a task that requires the use of a car, rank higher a volunteer with more driving experience than a person that lives closer to the user).
- Defining the social explanations about the inclusion of volunteers (e.g., for a task that requires help with moving, explain that a volunteer was included because they own a car and have helped others move in the past). Such information is meant for both the task creator and for the identified volunteers as it allows the motivate and explain the reasons why a match was identified.

6.2. INTERFACES

Method	Path	Description
Social Relations API		
GET	*/social/relations/{userID}/*	Return all social relations of the user

Co-funded by the Horizon 2020 Framework Programme of the European Union

Social Preferences API		
GET	*/social/preferences/{userID}/{taskID}/*	Return a ranked list of volunteers based on the user's social preferences.
Social Explanations API		
GET	*/social/explanations/{userID}/{taskID}/*	Return a social explanation of why a particular user is fit for the task.

7. SOCIAL HUB (PROFILE MANAGER)

7.1. FUNCTIONALITY

This component is responsible for storing and maintaining the WeNet user profile. A profile consists of a set of attributes, and their current values represent the state of the user. The main reasons why we cannot use any current user profile standard are the General Data Protection Regulation of the European Union (GDPR)[5] and the need to represent diversity. The GDPR forces to specify which personal data we store and what we are going to do with it, so we cannot store any data that we are not going to use. On the other hand, our project is focused on diversity and the current standards do not have sufficient expressivity on this field. For all this reason we define the profile with the following attributes:

- **id**: identifier of the WeNet user.
- **name**: includes the details about the user identity (first, middle and last name in addition to a prefix and a suffix).
- **dateOfBirth**: the date of birth of the user.
- **gender**: the gender of the user.
- **email**: the email of the user.
- **phoneNumber**: the phone number of the user, on the E.164 format (^+?[1-9]\d{1,14}$).
- **locale**: the locale used by the user.
- **avatar**: the URL to an image that can be used as an avatar of the user.
- **nationality**: the nationality of the user.
- **languages**: this is a set of languages that the user can understand.
- **occupation**: the occupation of the user.
- **norms**: the individual norms of the user.
- **plannedActivities**: the activities planned by the user.
- **relevantLocations**: the locations of interest for the user (may be the home or work location).

[5] https://eur-lex.europa.eu/eli/reg/2016/679/oj

Co-funded by the Horizon 2020 Framework Programme of the European Union

- **relationships**: the relations of this user with other users.
- **socialPractices**: the social practices of the user.
- **personalBehaviours**: the personal behaviours of the user.

7.1.1 Language

The languages that a user can understand are defined by a descriptive label, the ISO 639-1 code of the language and the linguistic ability level of the person. This level can be:

- **A0**: Beginner/False beginner. The person only knows a few words of the language, if that.
- **A1**: Elementary. The person can ask a few basic questions and ideas, but with a lot of mistakes.
- **A2**: Pre-intermediate. Limited vocabulary, but with some help the person can participate in basic conversations. She still makes a lot of big mistakes.
- **B1**: Intermediate. Limited vocabulary; the person can unerstand the main points on common topics at work, school or travelling.
- **B2**: Intermediate. Now the person can converse in many situations, with less serious errors.
- **C1**: Upper-intermediate. The person is comfortable in most situations, still some relevant mistakes are present.
- **C2**: Proficient. The person is fluent, pretty much mother tongue. Extremely comfortable, she has complete control over the language.

7.1.2 Gender

The gender of a user can be:

- **F**: A person whose gender identity matches the female sex.
- **M**: A person whose gender identity matches the male sex.
- **O**: Identifying a gender other than male or female.

7.1.3 Personality

The personality of a user is calculated following the Post-Jungian [2] theory of personality which is a reduced variant of the Myers–Briggs Type Indicator (MBTI) [3]. A user can obtain its personality filling in a questionnaire, that it can get from this component, and posting the answers to this component that returns the personality.

7.1.4 Intelligences

The intelligences of a user follow the theory of Gardner's multiple intelligences [4]. A user can obtain its intelligences getting the questionnaire from this component. After that it has to post the answers and this component will return the intelligence levels of the user.

7.2. INTERFACES

Method	Path	Description

Co-funded by the Horizon 2020 Framework Programme of the European Union

User profile		
GET	*/profiles/{profileId}/historic*	Get the status of the profile in specific time period
POST	*/profiles*	Create a profile
GET	*/profiles/{profileId}*	Get a user's profile
PUT	*/profiles/{profileId}*	Update the specified profile
DELETE	*/profiles/{profileId}*	Delete the profile of a user
Intelligence and Personality		
GET	*/intelligences*	Obtain the intelligence traits
POST	*/intelligences*	Calculate the intelligence traits of a user
GET	*/personalities*	Obtain the personality traits
POST	*/personalities*	Calculate the personality traits of a user

8. Interaction Protocol Engine

8.1. FUNCTIONALITY

The Interaction Protocol Engine is responsible for storing and maintaining the community description, along with their associated norms, to realize the user interactions between them, sending messages guaranteeing that they follow the norms, and to form teams to complete tasks. The functionalities provided by the Interaction Protocol Engine are organized around the concepts of community, message, norm, team and trust.

A Community consists of a set of users with the same goal.

Norms allow the specification of rules that can be applied on various levels: the individual one, the community one and the task one. An individual's norm might be "Suppress incoming messages at night". A task norm might be "Don't ask my ex". Finally, a community norm might be "If you don't volunteer, you cannot ask for help".

Messages exchanged during conversations are analysed by the Engine with the objective of verifying that they satisfy all the norms from the point of view of the individuals, the community and the task involved.

A Team consists of a group of users who decided to work together to complete a task. Its description can be detailed in order to define the importance and minimum level for each participant, and the proficiency degree of the team. It is up to the Interaction Protocol Engine to determine the best team distribution [5].

The Trust determines the level of trustworthiness of a user toward another one and its value is updated in real time during every step of the users' collaboration.

Co-funded by the Horizon 2020 Framework Programme of the European Union

This module will not be created from scratch because it will be developed from our previous work on uHelp[6] and team formation[7].
uHelp is a mobile application that allows users to provide or demand help in a community. From it, we are going to use the component to manage communities, and we will use the uHelp norm engine, it is similar to the If This Then That[8], meanwhile we work on the development of the norm interpreter.
From our previous work on team formation, we are going to provide the mechanism to calculate the personality or intelligence of a person, and the algorithms to build teams with diversity in gender, personality and intelligence.

8.2. INTERFACES

Method	Path	Description
Community		
GET	*/communities*	Return the communities that a used can join
POST	*/communities*	Register a community
GET	*/communities/{communityId}*	Return a community associated to the identifier
PUT	*/communities/{communityId}*	Update the values of a community
DELETE	*/communities/{communityId}*	Delete a community
GET	*/communities/{communityId}/norms*	Get some community norms
POST	*/communities/{communityId}/norms*	Add a norm into a community
GET	*/communities/{communityId}/norms/{communityNormId}*	Get a community norm
DELETE	*/communities/{communityId}/norms/{communityNormId}*	Remove a community norm
GET	*/communities/{communityId}/members*	Get the users in a community
POST	*/communities/{communityId}/members*	Add a user into a community
GET	*/communities/{communityId}/members/{userId}*	Get a user from a community
PUT	*/communities/{communityId}/members/{userId}*	Modify a user from a community

[6] https://uhelpapp.com/
[7] https://education.iiia.csic.es/#!/welcome/teams
[8] https://ifttt.com/

Co-funded by the Horizon 2020 Framework Programme of the European Union

DELETE	*/communities/{communityId}/members/{userId}*	Remove a user from a community
GET	*/communities/{communityId}/taskTypes*	Get the task types that can be done in a community
POST	*/communities/{communityId}/taskTypes*	Add a task type into a community
GET	*/communities/{communityId}/taskTypes/{communityTaskTypeId}*	Get a community task type
DELETE	*/communities/{communityId}/taskTypes/{communityTaskTypeId}*	Remove a community task type
POST	*/communities/{communityId}/taskInstances*	Associate a task to a community
Message		
POST	*/messages*	Send a message
Norm		
GET	*/norms/published*	Search for some published norms
POST	*/norms/published*	Publish a norm
GET	*/norms/published/{publishedNormId}*	Get a published norm
DELETE	*/norms/published/{publishedNormId}*	Delete a published norm
Task		
GET	*/tasks/types/published*	Search for a published task types
POST	*/tasks/types/published*	Publish a task type
GET	*/tasks/types/published/tasks/types/published/{publishedTaskTypeId}*	Get a published task type
DELETE	*/tasks/types/published/tasks/types/published/{publishedTaskTypeId}*	Delete a published task type
Team		
GET	*/teams*	Calculate the number of teams and each sizer that the composite can create
POST	*/teams*	Create teams
Trust		
GET	*/trusts/{userId}*	Get the friends that have a trust in a specific range

Co-funded by the Horizon 2020 Framework Programme of the European Union

POST	*/trusts/{userId}*	Set the trust of a user respect another
GET	*/trusts/{userId}/friends/{friendId}*	Get the trust over another user
PUT	*/trusts/{userId}/friends/{friendId}*	Update the trust over another user
DELETE	*/trusts/{userId}/friends/{friendId}*	Remove the trust over another user
GET	*/trusts/{userId}/communities/{communityId}*	Get the friends on a community that have a trust in a specific range
POST	*/trusts/{userId}/communities/{communityId}*	You can set the trust of a user respect another on a community
GET	*/trusts/{userId}/communities /{communityId}/friends/{friendId}*	Get the trust over another user on a community
PUT	*/trusts/{userId}/communities /{communityId}/friends/{friendId}*	Update the trust over another user on a community
DELETE	*/trusts/{userId}/communities /{communityId}/friends/{friendId}*	Remove the trust over another user on a community
GET	*/trusts/{userId}/communities /{communityId}/taskTypes/{taskTypeId}*	Return the friends to do a specific task type on a community that have a trust in a specific range
POST	*/trusts/{userId}/communities /{communityId}/taskTypes/{taskTypeId}*	You can set the trust of a user respect another to do a task type on a community
GET	*/trusts/{userId}/communities/{communityId}/taskTypes/{taskTypeId}/friends/{friendId}*	Get the trust over another user on a community to do a task type
PUT	*/trusts/{userId}/communities/{communityId}/taskTypes/{taskTypeId}/friends/{friendId}*	Update the trust over another user on a community to do a task type
DELETE	*/trusts/{userId}/communities/{communityId}/taskTypes/{taskTypeId}/friends/{friendId}*	Remove the trust over another user to do a task type on a community
GET	*/trusts/{userId}/publishedTaskTypes/{publishedTaskTypeId}*	Get the friends that have a trust in a specific range to do a published task type
POST	*/trusts/{userId}/publishedTaskTypes/{publishedTaskTypeId}*	You can set the trust of a user respect another to do a published task type
GET	*/trusts/{userId}/publishedTaskTypes/{publishedTaskTypeId}/friends/{friendId}*	Get the trust over another user to do a published task type
PUT	*/trusts/{userId}/publishedTaskTypes/{publishedTaskTypeId}/friends/{friendId}*	Update the trust over another user to do a published task type

Co-funded by the Horizon 2020 Framework Programme of the European Union

DELETE	*/trusts/{userId}/publishedTaskTypes/{publishedTaskTypeId}/friends/{friendId}*	Remove the trust over another user when do a published task type

9. Incentive server

9.1. FUNCTIONALITY

The Incentive Server is responsible for generating diversity-aware incentives, required during WeNet's individual and group interactions to maximize their probability of success. An interaction is understood as successful if it is beneficial to accomplishing the goals of the different participants in the WeNet system. The component leverages the information provided by the iLogBase for identifying the appropriate incentives to users while adhering to WeNet norms. The current APIs support message based and badge based incentives while enabling to set and change norms for the Incentive Server operation.

There are three mains API's families: Incentives, Norms and Badges.

- Incentive : The API's supports enabling and disabling incentives, getting incentives by users/communities and submitting a complaint about incentive.
- Norms: The API's supports adding, getting and editing norms for users, communities and general norms.
- Badges: The API's supports creating and editing issuers and BadgeTypes. The API's also allowing to issue an Assertion of a BadgeType to a specific user. This API is based on the Open Badges 2.0 (OBv2) API specification.

9.2. INTERFACES

Method	Path	Description
Incentives		
GET	/incentive/apps/{app_id}/community/{community_id}	searches for a community of users of a specific application and returns incentive information of the users.
GET	/incentive/apps/{app_id}/users/{user_id}	searches for a user of a specific application and returns incentive information of that user.

Co-funded by the Horizon 2020 Framework Programme of the European Union

POST	/incentive/disable_incentive	disables the incentive server
POST	/incentive/enable_incentive	Enables the incentive server
POST	/enquiry_incentives/app/{app_id}/users/{user_id}	submit complaint regarding an incentive
Norms		
POST	/norm/apps/{app_id}/users/{user_id}	adds a user norm
GET	/norm/apps/{app_id}/users/{user_id}	get the user norms
POST	/norm/apps/{app_id}community/{community_id}	adds a community norm
GET	/norm/apps/{app_id}community/{community_id}	Get the community norm
POST	/norm/apps/{app_id}	adds a general norm
GET	/norm/apps/{app_id}	get a general norm
PUT	/norms/apps/{app_id}/{norm_id}	Update a norm by id.
GET	/norms/apps/{app_id}/{norm_id}	get norm by id.
Badges		
POST	/badges/issuers	Create a new Issuer

Co-funded by the Horizon 2020 Framework Programme of the European Union

PUT	/badges/issuers/{entity_id}	Update a single Issuer
POST	/badges/badgeTypes	Create a new BadgeType
PUT	/badges/badgeTypes/{entity_id}	Update an existing BadgeType
POST	/badges/badgeTypes/{entity_id}/assertions	Issue an Assertion to a single recipient
GET	/badges/badgeTypes/{entity_id}/assertions	Get a list of Assertions for a single BadgeType

10. Interaction logger and analytics

10.1. Functionality

The Interaction Logger and Analytics component is responsible for collecting and storing all the messages, and associated metadata, that are exchanged during the conversations that take place during the organization and management of a task together with the low software logging events. In addition to this, this component is also responsible for computing key analytics describing the KPIs of each conversational application built on top of the WeNet platform.

This component is going to be built upon a logger and analytics platform developed internally by U-Hopper in the context of Memorable Experiences project (carried out within the ERDF Regional Operational Programme 2014-2020 of the Autonomous Province of Trento and co-financed by the European Union through the European Fund for Regional Development, for the Italian state and the Province of Trento).

Both Logger and Analytics information contain key information allowing to monitor the success of a conversational application. On the one hand, message data is going to be of great help to the applications' developers who will have the possibility to understand how conversations are evolving from a technical point of view, easily identifying potential issues and blocking factors. On the other hand, analytics data is going to offer key information on the performance and effectiveness of a service.

The Message model has been defined in order to provide a general structure that allows to log messages exchanged not only on the most common messaging platforms (e.g. Facebook, Telegram) but also on custom conversational services. In addition to the more common message details (such as text, sender and received), this model also includes the support for

Co-funded by the Horizon 2020 Framework Programme of the European Union

detailing the language and the outcome of an optionally applied NLP analysis together with a general metadata field.

The Log model has been structured in order to include the most common features of a code logging event. Among the others: component name, severity, timestamp, message and a general metadata field.

10.2. INTERFACES

Method	Path	Description
Message logging		
POST	*/logging/messages*	Register a batch of messages
GET	*/logging/message/{messageId}*	Get the details of the specified message
PUT	*/logging/message/{messageId}*	Update the details of a specific message
DELETE	*/logging/message/{messageId}*	Delete a message
Software event logging		
POST	*/logging/logs*	Register a batch of data logs
GET	*/logging/log/{logId}*	Get the details of the specified log event
PUT	*/logging/log/{logId}*	Update the details of a specified log event
DELETE	*/logging/log/{logId}*	Delete a log

11. TASK MANAGER

11.1. FUNCTIONALITY

The Task Manager is the component responsible for maintaining the state of tasks and providing task management functionality. A Task collects all the information regarding a user-generated help request. In addition, the Task contains all the information regarding the status of a task including the lists of identified users to be notified about the task, the list of users who volunteer for it, the details about the ongoing task discussion and the volunteer(s) chosen by the requester.

The Task manager support the application of task transactions defining specific operations that can be applied on a task depending on its current state (e.g. adding a new volunteer, selecting a participant, ...).

Co-funded by the Horizon 2020 Framework Programme of the European Union

11.2. INTERFACES

Method	Path	Description
POST	/task/{taskId}	Create a new Task
GET	/task/{taskId}	Get the details of an existing Task
POST	/task/{taskId}/transaction	Apply a transaction to the specified task

12. Authentication manager

12.1. FUNCTIONALITY

The Authentication Manager is responsible for handling the creation and authentication of applications, project components and users.
Every request performed to the platform (by taking advantage of the Service and Commons APIs) and within the platform (in the communication between the various platform components) must be authenticated. In order to do so, both applications and components must be registered in the Authentication Manager.
The Authentication Manager is also responsible for storing the identity of all the WeNet users and for validating their identity once they login into a WeNet application.
Authentication is surely a key aspect to be taken into consideration and to be included and guaranteed during the communication to and within the WeNet platform. No final decision has yet been made about the implementation framework. Various IdM standards, such as OpenID and OAuth2, are available and will be taken into consideration in order to avoid creating a custom WeNet identity management standard. For this reason, no interface specifications are reported for the authentication manager.

13. MAPPING UHELP TO WENET

We mapped all functionality of the uHelp app [1] to the WeNet platform. In the following diagrams we report how a set of core uHelp user stories can be mapped to the functionality exposed by the WeNet platform (and inside components). Later on we share some possible enhancements that WeNet can offer with respect to the current uHelp implementation.

We cover the following user stories, which cover most of the help request and user lifecycle in uHelp:

- A user creates a new account;
- A user logs in using an existing account;
- A user (issuer) creates a help request;
- A help request gets disseminated to potential helpers;

Co-funded by the Horizon 2020 Framework Programme of the European Union

- A user agrees to a help request (and becomes a helper);
- A helper chats with the issuer to sort out help request details;
- An issuer chooses a helper to carry out the task;
- A helper carries out a help request issued by an issuer;
- A helper marks a help request as completed;
- An issuer rates the helper for a completed task.

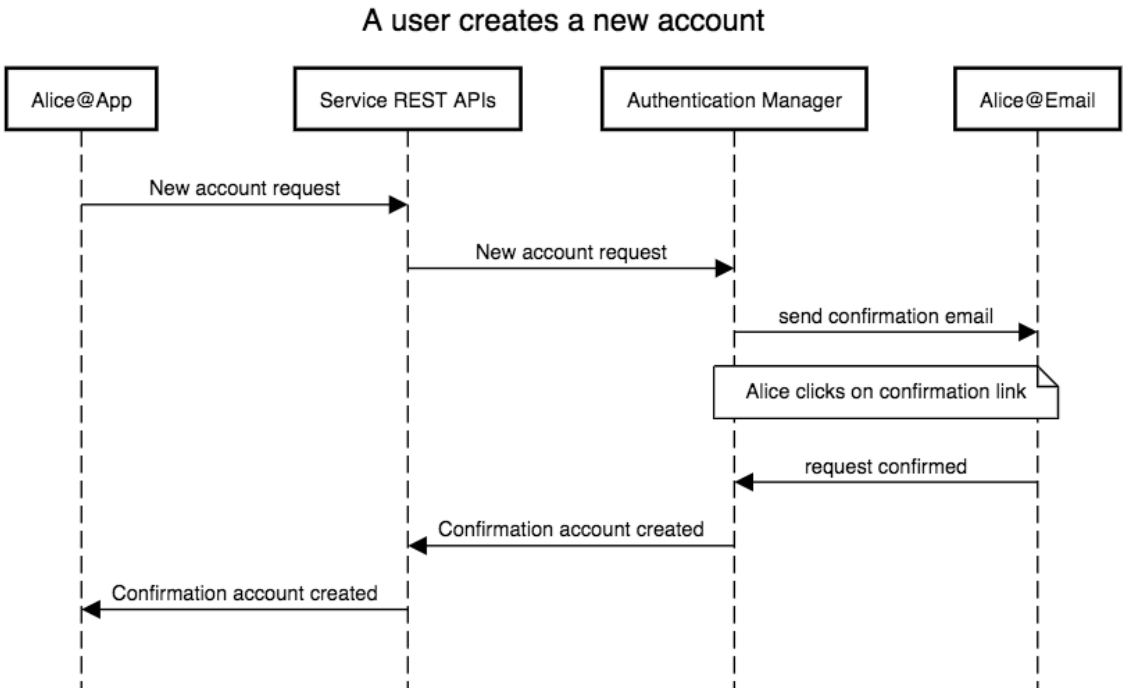


FIGURE 3: MAPPING UHELP: A USER CREATES A NEW ACCOUNT

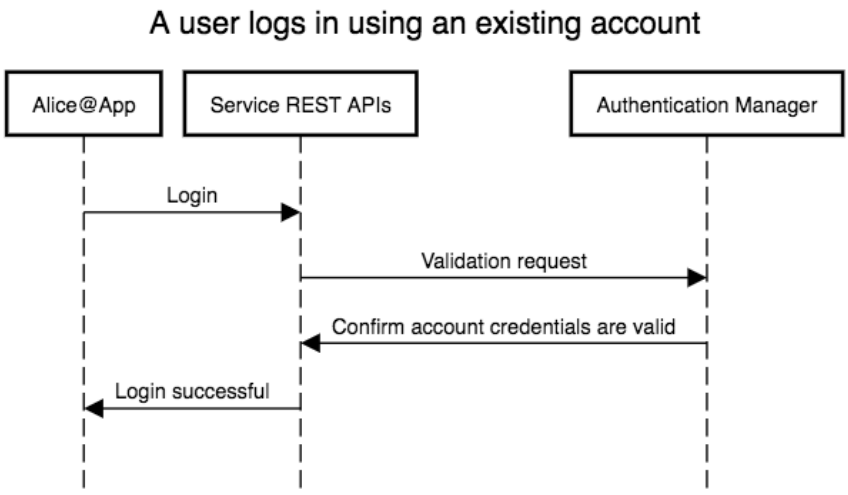


FIGURE 4: MAPPING UHELP: A USER LOGS IN USING AN EXISTING ACCOUNT

Co-funded by the Horizon 2020
Framework Programme of the European Union

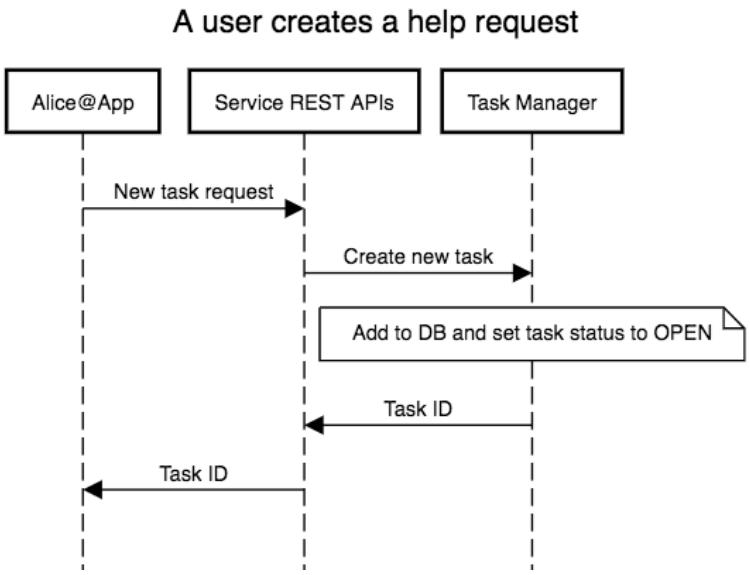


FIGURE 5: MAPPING UHELP: A USER CREATES A HELP REQUEST

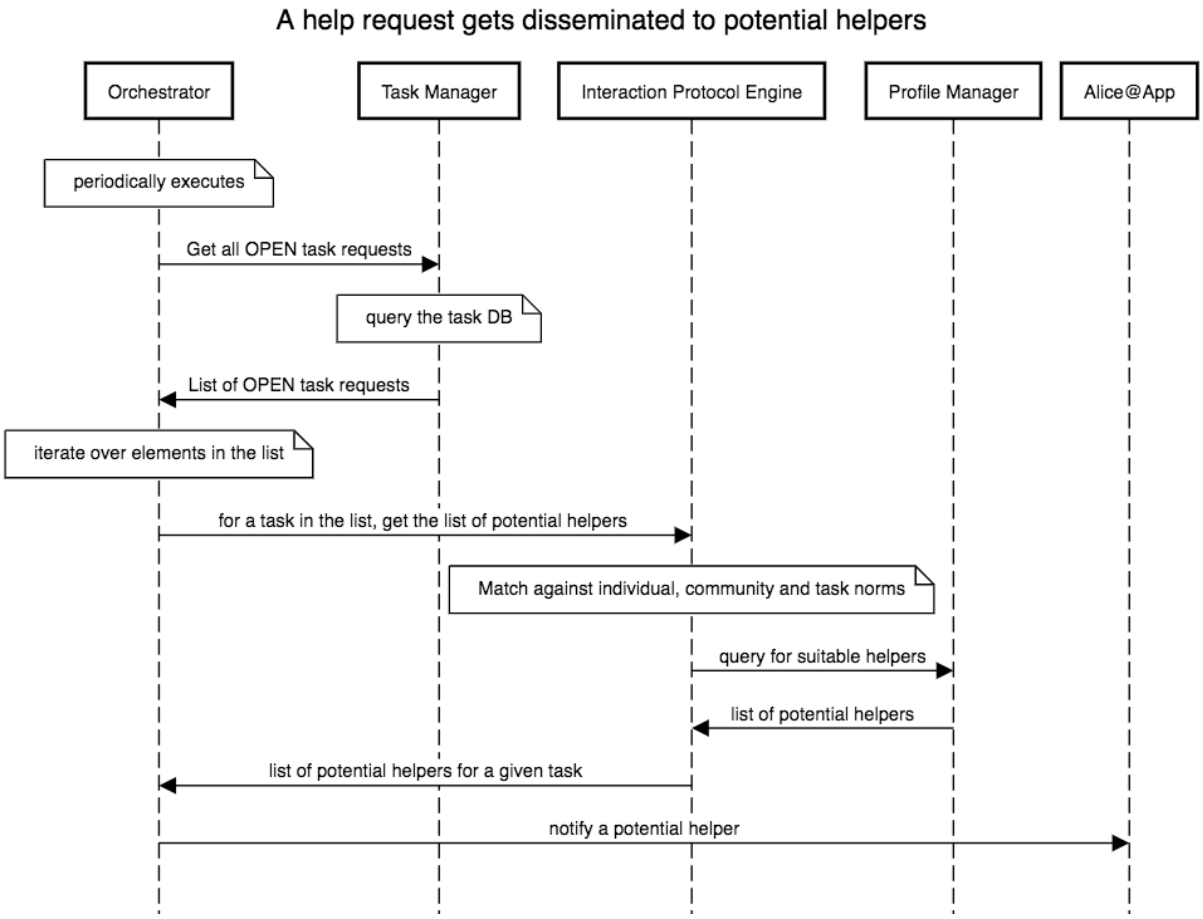


FIGURE 6: MAPPING UHELP: A HELP REQUEST GETS DISSEMINATED TO POTENTIAL HELPERS

Co-funded by the Horizon 2020
Framework Programme of the European Union

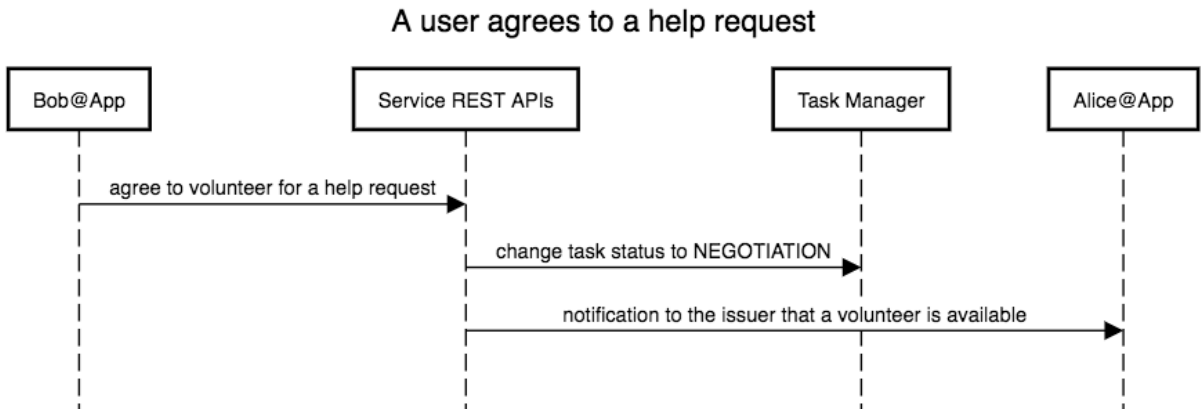


FIGURE 7: MAPPING UHELP: A USER AGREES TO A HELP REQUEST

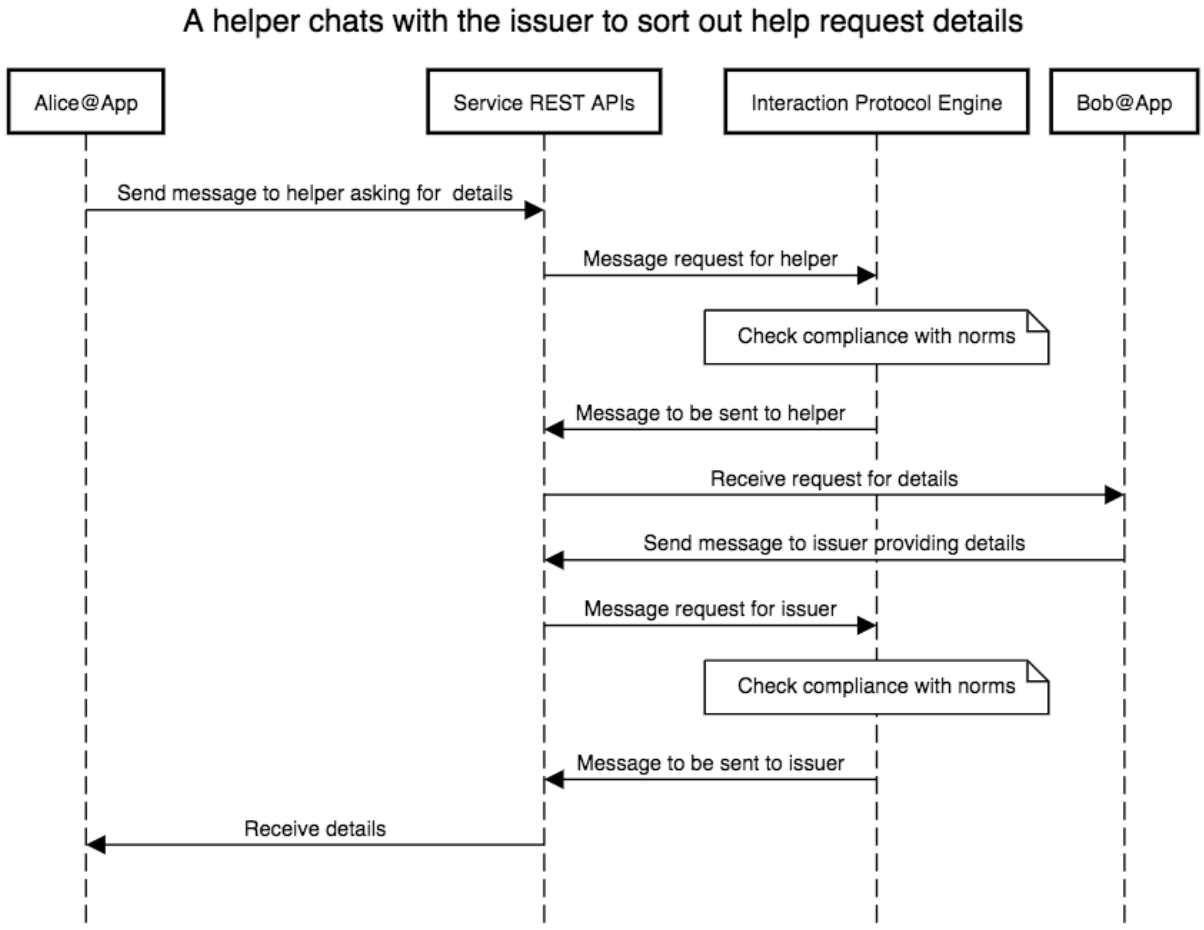


FIGURE 8: MAPPING UHELP: A HELPER CHATS WITH THE ISSUER TO SORT OUT HELP REQUEST DETAILS

Co-funded by the Horizon 2020
Framework Programme of the European Union

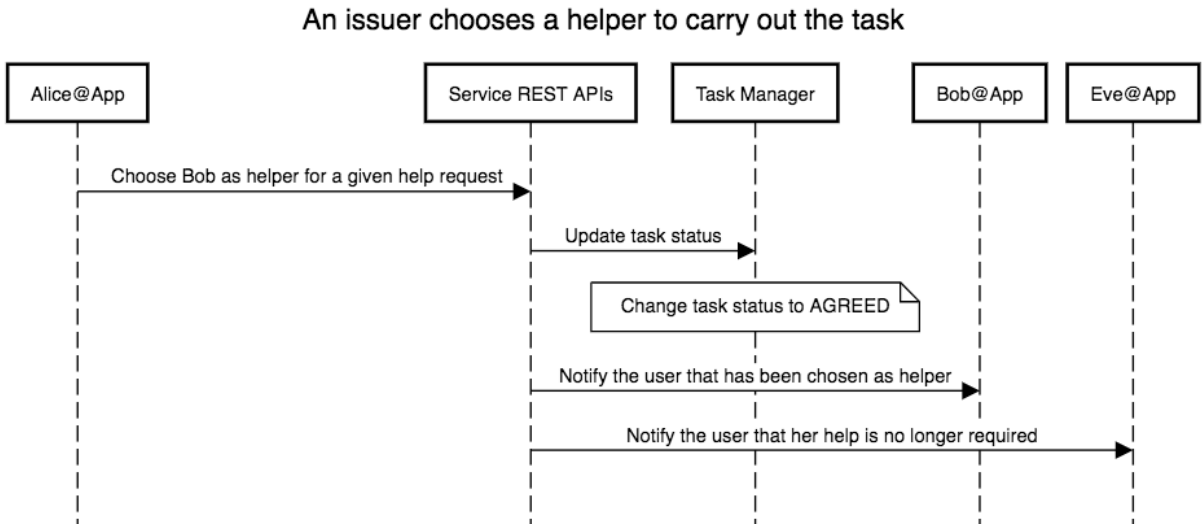


FIGURE 9: MAPPING UHELP: AN ISSUER CHOOSES A HELPER TO CARRY OUT THE TASK

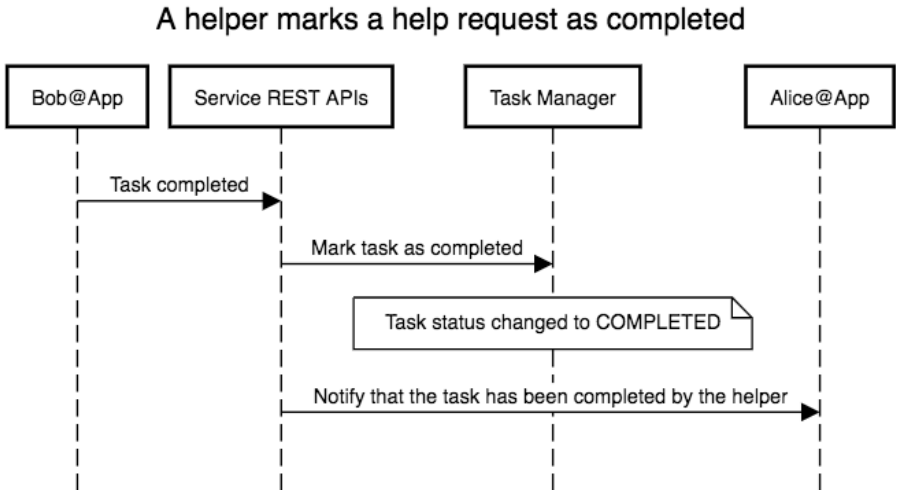


FIGURE 10: MAPPING UHELP: A HELPER MARKS A HELP REQUEST AS COMPLETED

Co-funded by the Horizon 2020
Framework Programme of the European Union

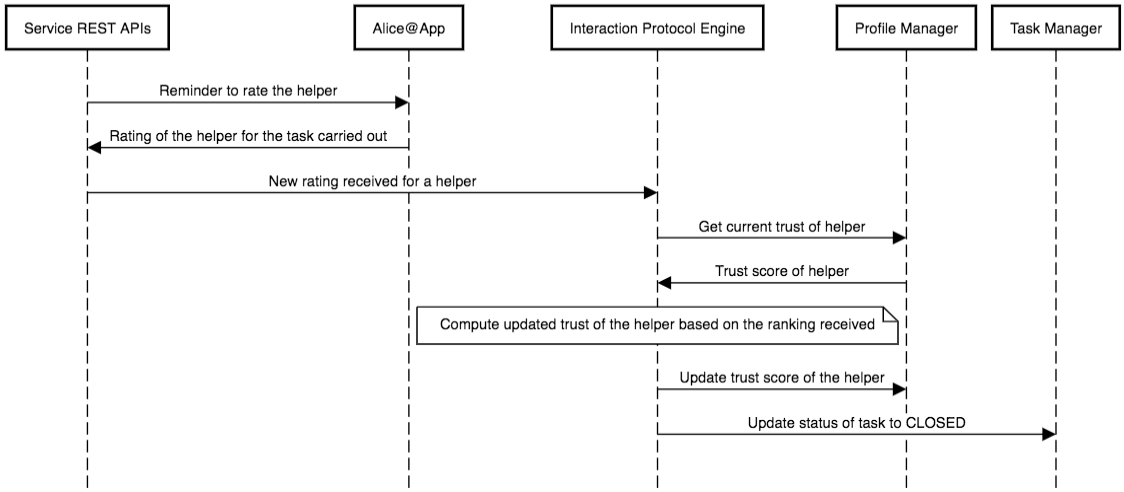


FIGURE 11: MAPPING UHELP: AN ISSUER RATES THE HELPER FOR A COMPLETED TASK

As it may be easily noticed, the mapping provided above does include only a subset of WeNet platform component. In particular, the chain WeNet Common APIs-Personal/Social Context Builder-Profile Manager is not used. And the same applies to the Incentive Server. This reflects the fact that such components do not have a direct counterpart in the current uHelp implementation. However, they could be used to enhance the functionality of a `WeNet-tised' version of uHelp as well as to improve the resulting user experience. This may include:

- Provide issuers with a ranked list of potential helpers (from Social context builder);
- Provide routine-aware notifications to maximise the chances of user engagement (Incentive server and Personal context builder);
- Provide incentives (badges etc.) to maximise user engagement and keep them in the platform for a longer time (Incentive server);
- Adapt the way messages are presented to different users based on their profile characteristics (Interaction protocol engine and Personal context builder).

14. WENET RESEARCH INFRASTRUCTURE

The focus of the WeNet Research Infrastructure is to provide access to high-quality human-centric socio-experiment data. This includes the setup of the technological infrastructure to store WeNet-relevant data and the services to enable scientists to access it. In our vision, a researcher should be able to collect and share data with little effort. At the same time, the Research Infrastructure needs to support for ethics and privacy compliance in relevant research and to facilitate the adoption of best practices and standards in research data sharing.

14.1. FUNCTIONALITY AND EXPECTED USAGE

The Research Architecture is designed to provide a single point of contact and to assist researchers in the study design, experiment reproducibility and data sharing in an ethical and privacy-compliant environment. This architecture is developed based on research and practical experience within the WeNet Consortium and is also enriched with a number of early adoption cases at WeNet partner institutions. The WeNet partners'

Co-funded by the Horizon 2020 Framework Programme of the European Union

expertise and reputation in the scientific sector, along with the Research Infrastructure technical and functional solution helps to establish a leading position as trusted providers of human-socio data and data services for research.

A wide range of human-centric socio-experiment data are made available through the Research Infrastructure. At the beginning, much of the data we will provision for research will come from the WeNet pilot experiments. The WeNet consortium holds and publishes the data, which can be re-used to support research studies. The Research Infrastructure Data Catalogue provides the full list of datasets held by the platform. The Data Catalogue enables the production, reuse and sharing of data sets, creates the marketplace that inventories data, catalogues meta-data and contextual information, and shares the knowledge needed to trust experiment data. Researchers trust analysis when the methods, context and data points are all well understood by decision makers. The Data Catalogue supports to achieve this with technology to:

- Inventory data and analytic assets;
- Catalogue the context of data and encourage social curation;
- Execute analysis in a replicable and repeatable way.

To have direct access to data, all researchers are required to demonstrate a research institution affiliation. Studies requiring access to experiment data must also ensure the policies and procedures governing individual privacy, data protection and freedom of information, as well as protection for breach of security or confidentiality of data. Studies using personal data must have appropriate approvals in place, and the researcher should be named in the application or data sharing agreement. The Research Data Infrastructure provides methods for accessing data, removes the barriers to scientific reproducibility of experiments with meta-data-aware collaboration tools, allows the execution of experiment analysis in a replicable and repeatable way, supports data for teaching usage and provides deep insight into how users create and share knowledge from human-centric socio-experiment data through reports that include top users and dataset popularity.

The WeNet partners have decades of experience in all aspects of human-centric socio researches and publications. As strong advocates of using data for research, the partners use their policies, procedures and background knowledge to help developing the Research Infrastructure services to manage research data more effectively. The Research Infrastructure platform includes software, methodology, technology and best practices and represents the culmination of the multidisciplinary research areas and expertise of the partners as well as an innovative tool set for computational social sciences to be used to provide answers for the needs of the scientific researchers.

14.2. Structure and Architecture

The Research Infrastructure design is based on a multi-stakeholder approach and sets a focus on data on human behaviour and social interactions, software tools and services for data collection and analysis, online training material, protocols for running experiments and trials, and compliance to ethics and European Privacy Law (GDPR), allowing interested parties from different research entities to join the WeNet Research Infrastructure community. The project enforces community building by reducing technical and legal boundaries across Europe. The expansion will be fuelled by the inception of the federated Research Infrastructure, which will include large amounts of data on human behaviour and social interactions, as well as components and tools provided by WeNet partners. The growth of the community will be supported by a coherent set of actions, including communication activities.

Co-funded by the Horizon 2020 Framework Programme of the European Union

The hearth of the Research Infrastructure marketplace is the Data Catalogue. The Data Catalogue is a single point of reference for all the data assets & data knowledge in the Research Data Infrastructure. The Data Catalogue enables the production, reuse and sharing of data sets, creates the marketplace that inventories data, catalogues meta-data and contextual information, and shares the knowledge needed to trust experiment data. The Data Catalogue makes finding and understanding the data easier and ensures that every user has equal access to data knowledge. It supports the novice researcher as well as the expert researcher in the task of data discovery and makes data, meta-data, and algorithms for data analysis easily discoverable. It encourages the Data asset reuse by lowering the barrier to the data consumers for “self-serving” the highest quality data assets.

The central component of the Research Infrastructure is a Centralized Data Repository. The purpose of this centralized approach is to provide one centralized storage facility, collecting all relevant data to be shared. The initial assumption is that the data would be available in a central storage location. Ideally, central location is a suitable solution; unfortunately, many real situations are far from ideal. During the shaping and the design phases we will consider real situations where different levels of federation could be potentially adopted in the final deployment strategy.

The research Infrastructure approach has been chosen to combine the experience of all partners of the WeNet project pilots, in order to bootstrap a huge repository of reusable data for an effective solution towards the increasing demand of human-centric socio data for scientific research. This approach allows researchers and academic institutions to get access to data and data services for their research, which is today unfeasible due to a lack of appropriate and sufficient data availability. Large corporations like Internet Service Provider own large human-related datasets, but academy does not have access to this information.

WeNet project joint effort and expertise enhance the possibility of a successful start up and adoption pattern. Before marketing the new Research Data Infrastructure within the academic field the WeNet partners will be the first to try the offers providing a degree of “thought leadership” for the potential adopters. The following scientific adopters will reasonably be risk averse; they will probably be in contact with thought leaders and use the opinions of these when making their first adoption decisions.

The overall software and service architecture of Research Infrastructure follows the idea of keeping the infrastructure for collaboration and data sharing as simple and flexible as possible. Next to the requirements for an easy accessible service platform, the architecture of the Research Infrastructure needs to provide an effective ethics and privacy compliance support, as the data reuse and data service runs into a target of human-centric socio data. The provided solution also needs to provide the participating researchers with a high level of trust, to share their “sensitive” and “proprietary” data with other stakeholders. As a matter of fact, the WeNet project needs to take ethics and privacy requirements into consideration for the overall solution design. Finally, the conceptual design needs to be future-oriented, as the Research Infrastructure is intended to become a sustainable solution after the official end of the WeNet project.

Co-funded by the Horizon 2020
Framework Programme of the European Union

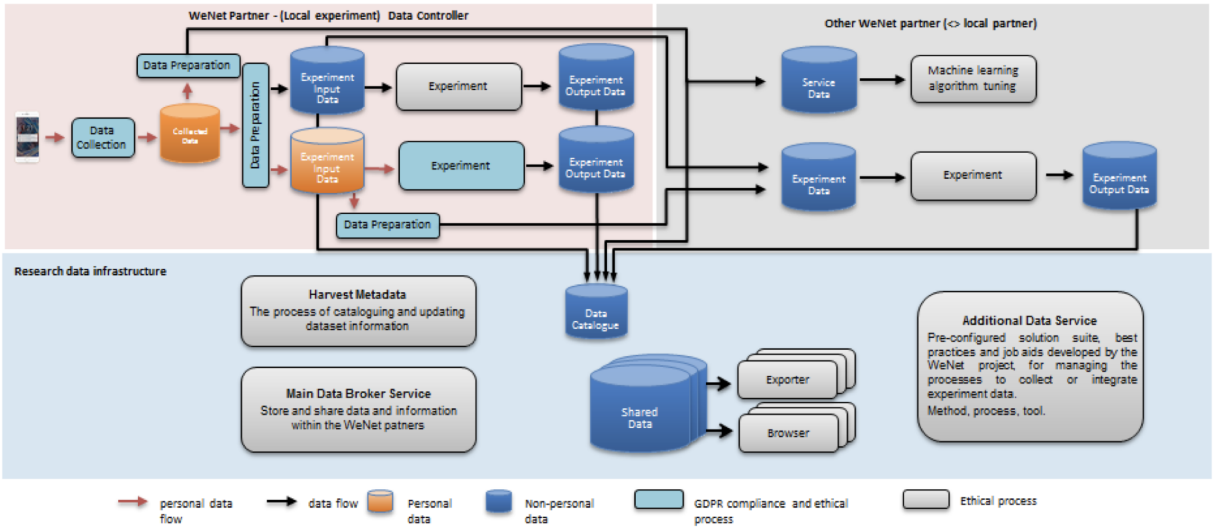


FIGURE 12: WENET RESEARCH INFRASTRUCTURE: HIGH-LEVEL ARCHITECTURE

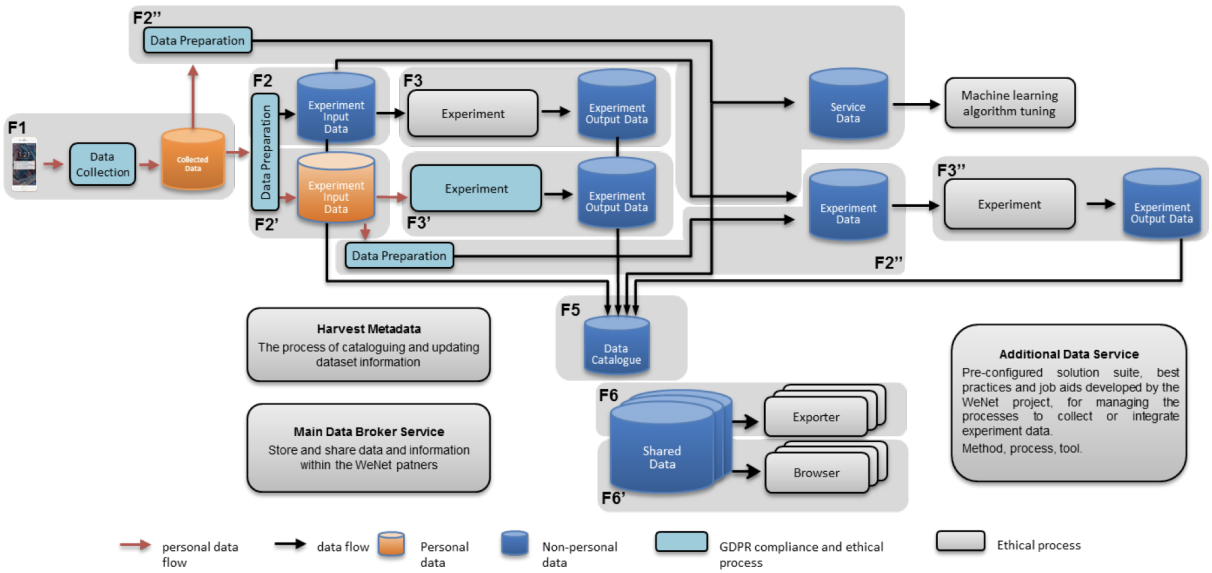
The locally collected datasets are stored in an IT backbone cloud infrastructure. These datasets are further processed for classification in a number of more focused categories and table schemas to be used by the Data Controller and/or requested by WeNet partners for performing experiments within the project. To satisfy these requests and comply with regulations experiment input data will be generated from the collected data.

All the available datasets will be listed in the Data Catalogue and detailed information on the collected and experiment data can be found in the WeNet Research Platform.

The data preparation operation to generate the experiment input data from the collected data will be carried out under the responsibility of the original data controller of the collected data. WeNet pilot trials are designed under a common generic methodology to be adapted and implemented under local conditions and the specific needs of the involved users' needs. Consortium partners of the WeNet project are initially the creator and the primary target audience for these datasets, but these datasets are also of interest in the relevant scientific communities. This leads to the federated Research Data Infrastructure, where the resulting datasets from the local pilot trials are used for research activities that may go beyond the original scope of those trials. The Research Infrastructure defines the data sharing methods to clean/anonymize/summarize data to enable it use by researchers in an ethic and privacy compliance approach.

The following figure describes the expected information flow within the research infrastructure.

Co-funded by the Horizon 2020 Framework Programme of the European Union



Legenda
F1: Data collection of personal data
F2: Preparation of experiment data
F2': Preparation of experiment data with personal information
F2'': Data syntesis of service reusable data
F2''': Preparation of experiment and service reusable data
F3: Experiment execution of anonymized data and preparation of experiment results
F3': Experiment execution of personal data and preparation of experiment results
F3'': Experiment reproduction of anonymized data and preparation of experiment results
F4: Preparation of Data catalog
Web portal development and content update
F6: Retrival of stored data, export and preparation log for reuse
F6' Retrival of stored data and browse

FIGURE 13: WENET RESEARCH INFRASTRUCTURE: FLOWS OF DATA AND OPERATIONS

15. OUTLOOK AND NEXT STEPS

This deliverable includes the architectural description of the WeNet platform. Eleven components are identified, and for each one of them the functionality and APIs are briefly described. An example (based upon the functioning of the uHelp app) is used to showcase how the various components interact. The deliverable also includes some preliminary results on the architecture of the WeNet research infrastructure.

All APIs and data models are maintained by the project on a git repository. D6.1 represents the basis for the initial implementation and integration. By M18 we will have a first version of the platform up and running, and a first version of the WeNet app, ready to be used on the first set of pilots.

The next steps include not only the implementation of the described and documented components but, most importantly, the integration of their APIs.

A first round of implementation will be dedicated to the creation of the identified APIs while leaving aside any logic that is going to be required for the platform for offering its functionalities. This means that all the endpoints will be created, they will provide correct responses but these responses will be static and hard coded. This is because the aim of this first step consists in the integration and alignment of the eleven identified components: a key requirement for obtaining a functioning platform. We plan to complete this step by M13.

Co-funded by the Horizon 2020 Framework Programme of the European Union

Once completed the integration step, the second round of implementation will be characterized by the inclusion of the logic behind the defined endpoints. This second step will ensure that real data will be collected and created thus allowing the first test applications to be run on top of the WeNet platform. We will proceed iteratively, with monthly internal releases of the platform from M15 onwards.

In parallel, we will also develop guidelines (with the support of WP11) for ethical and diversity-aware operations of the platform, along the lines of [6].

Co-funded by the Horizon 2020
Framework Programme of the European Union

REFERENCES

[1] Koster, Andrew, Jordi Madrenas, Nardine Osman, Marco Schorlemmer, Jordi Sabater-Mir, Carles Sierra, Angela Fabregues, Dave De Jonge, Josep Puyol-Gruart, and Pere Garcia-Calvés. "u-Help: supporting helpful communities with information technology." In Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems, pp. 1109-1110. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[2] I.B. Myers, P.B. Myers. “Post-Jungian Personality Theory for Individuals and Teams”, 2013

[3] I.B. Myers, P.B. Myers . “Gifts differing: Understanding personality type”, 2010

[4] H. Gardner. “Frames of mind: The theory of multiple intelligences”, 2011

[5] Ewa Andrejczu, Filippo Bistaffa, Christian Blum, Juan A.Rodríguez-Aguilar, Carles Sierra, “Synergistic team composition: A computational approach to foster diversity in teams”, 2019

[6] Ethics Guidelines for Trustworthy AI, High Level Expert Group on Artificial Intelligence, 2019. https://ec.europa.eu/futurium/en/ai-alliance-consultation/guidelines#Top …

[7] Giunchiglia, Fausto, Mattia Zeni, and Enrico Big. "Personal Context Recognition via Reliable Human-Machine Collaboration." In 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), pp. 379-384. IEEE, 2018.

[8] A. de Götzen, T. Bjørner and G. D’Ettolle (eds), “Critical Issues and Scenario Development,” WeNet project Deliverable D7.1, Dec. 2019.

[9] S. Bradner, “Key words for use in RFCs to Indicate Requirement Levels”, RFC2119, Mar. 1997.

[10] F. Vega-Redondo, “Complex Social Networks”. Cambridge University Press, MA, 2007.

[11] Granovetter, M. S., “The Strength of Weak Ties: A Network Theory Revisited”. Sociological Theory, 1, 201-233, 1983.

[12] E. Gilbert, K. Karahalios, “Predicting tie strength with social media”. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 211-220, 2009.

[13] L. Michael, “Machine Coaching”. IJCAI 2019, Workshop on Explainable Artificial Intelligence (XAI), 2019.

[14] H. Lakkaraju, E. Kamar, R. Coruana, J. Leskovec, “Faithful and Customizable Explanations of Black Box Models”. AIES, 2019.

Co-funded by the Horizon 2020 Framework Programme of the European Union