COBACORE

Community Based Comprehensive Recovery

Grant Agreement Nº 313308

D3.2: Report on interface components of COBACORE workspace and functional behaviour of the COBACORE system

Version:	1.0
Due Date:	30-06-2015
Delivery Date:	30-07-2015
Nature:	R
Dissemination Level:	PU
Lead partner:	TNO
Authors:	Martijn Neef (TNO), Marijn Rijken (TNO)
Internal reviewers:	WP leaders WP1, WP2, WP4, WP5. Leo Galway (UU), Conor Woods (GEOP), Francisco Fornés (INT), Martin Haran (UU)

www.COBACORE.eu

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 313308.

Version Control:

Version	Date	Author	Author's Organization	Changes
0.1	08-06-2015	Martijn Neef, Marijn Rijken	TNO	Base draft
0.2	12-06-2015	Marijn Rijken	TNO	Review draft
0.3	19-06-2015	Martijn Neef	TNO	First complete draft, for final review
1.0	20-07-2015	Martijn Neef	TNO	Final version

Deliverable Title: Report on interface components of COBACORE workspace and functional behaviour of the COBACORE system

Deliverable Number 3.2

Keywords: Functional specification; platform design

Executive Summary:

Report on interface components of COBACORE workspace and functional behaviour of the COBACORE system: This report describes the functional behaviour of the platform and various interface concepts of the COBACORE platform, including the underlying production rules and support mechanisms.

Table of Contents

Table of tables and figures

Tables

Figures

1 Introduction

This report represents deliverable 3.2 (D3.2) of work package 3 (WP3), which is titled: Report on interface components of COBACORE workspace and functional behaviour of the COBACORE system.

1.1 WP3 and its contribution to the COBACORE project

The Community Based Comprehensive Recovery (COBACORE) project aims to support common needs assessment and recovery planning efforts in complex multi-sectorial, multi-stakeholder crisis environments by building upon the community as an important source of information and capabilities. COBACORE aims to help bridge the so-called collaboration gap: failure of collaboration through insufficient information sharing among partners, incompatible work practices and misaligned decision making processes. In the field of humanitarian needs assessment, this collaboration gap is ubiquitous and detrimental to the efficiency of many recent relief efforts. Closing this gap is the key to improve the efficiency of needs assessment, enhancing the robustness of needs monitoring, as well as providing an evidence base to inform planning and resource allocation decision-making.

WP3 helps to develop the COBACORE concept through its tasks and intermediary role in the project. It is the responsibility of WP3 to define the underlying tool behaviour concepts that determine how end-users interact with the COBACORE platform. Therefore WP3 will develop user interface concepts, functional mechanisms and use procedures for effective use of the COBACORE platform.

The role and tasks of WP3 are closely related to the other work packages in COBACORE. The functional concepts developed by WP3 are based on functional requirements specified by WP1, the data- and information models from WP2 and feedback from stakeholder interaction sessions organised by WP5. Furthermore, the functional concepts of WP3 have consequences for the experimentation and evaluation requirements in WP5 and provide guidance to platform development activities in WP4.

1.2 Deliverable 3.2

D3.2 describes the core concepts of the COBACORE platform, the underlying assumptions and information elements, and introduces a number of design patterns, drawing from the initial core set of features introduced in D3.1 and outcomes from interactions sessions. These features are used by WP2 (information model), WP4 (constraints of the COBACORE platform) and WP5 (experimentation and evaluation of the COBACORE concepts) to structure their work, especially in light of the intermediate evaluation session (WP5).

D3.2 serves as a capture of the state of the platform at the time of report-release, including features that may be developed in the remaining project time.

This document describes the COBACORE development process (Chapter 2), the core concepts that underlay the COBACORE platform (Chapter 3), the feature set to be used in the development of the COBACORE platform (Chapter 4), and provides an overview of

functionality per user group). This document concludes with an outlook on the remaining part of the project (Chapter 5).

1.3 Relationship to other deliverables

This deliverable builds on top of the core features and information objects defined in D3.1. Relevant sections of D3.1 are included and updated where needed, so that D3.2 provides a complete, independently readable and final set of specifications.

The concepts and features defined in D3.2 are guided by domain analysis and user requirements in D1.1 and D1.2, and are aligned with the information model in D2.3.

Furthermore, D3.2 aligns with the upcoming WP4 deliverables D4.3 (Final platform implementation) and D4.4 (COBACORE platform user and administration guides) and provides the basis for the final development phase of the COBACORE platform.

2 COBACORE development process[1]

The domain analysis, as carried out by the WP1 team and described in D1.1 and D1.2, has established the core domain issues that are abound in the disaster recovery domain, and affect recovery performance. Table 1 summarises the findings in three core issues:

Issue	Description
I1	Information provision issues between the professional community and the affected community
I2	Collaboration issues between the professional community and the responding community
I3	Inefficiencies in needs and capacities matching between the affected and responding community

Table 1: COBACORE core issues

From these three core issues, three core functions were derived. These functions assert the major functions that the COBACORE platform should provide. Table 2 lists the three core COBACORE functions.

Function	Description
U1	Enhance information exchange between the professional community and the affected community
U2	Facilitate collaboration between the professional community and the responding community
U3	Improve needs and capacities matching between the affected and responding community

Table 2: COBACORE core functions

The three core functions do not state how the platform should behave or look, but denote the value of the platform for the target user groups. The functions thus guide the development of platform core concepts and features.

Further elaboration on the functions yields desirable features: aspects of the platform that contribute to the fulfilment of the stated functions. Because of the general nature of the functions, there will be many ways to satisfy them. One could suggest features that focus more on improving collaborative work and information exchange (e.g. build a portal that makes it easier for user to find matching needs and capacities), or one could suggest features by which

[1] This chapter has largely been previously published as part of D3.1. Because of readability and continuity concerns, this chapter has been reiterated from D3.1 and adapted where necessary.

the platform itself becomes more autonomous (e.g. the platform does the needs and capacities matching). There is no single best answer.

Through domain and state-of-the-art tool analysis, development workshops with consortium members, and (partial) evaluation sessions with stakeholders, valuable platform functionalities have been uncovered, prototyped and evaluated to inform the next iteration of the platform development. We have thus adopted a simple incremental platform development process. The process derived major platform functions from identified domain issues. From asserted functions, features are derived in a number of iterations, starting with the definition of a core feature set. Features should not be regarded as fine-grained functional specifications, but rather as blueprints that need to be satisfied by the eventual implementations. The features give direction to the implementation by proposing logical structures, interaction patterns and suitable interface elements. The actual design and implementation choices result from considerations by the various COBACORE project teams, from their own perspective.

In subsequent iterations, new features have been defined that build upon the previously implemented core features, and thus add new capabilities to the platform. Not all proposed features might be realisable during the project due to time constraints or technical limitations, so certain features might need to be realised in follow-up projects. Figure 1 illustrates this feature development process.

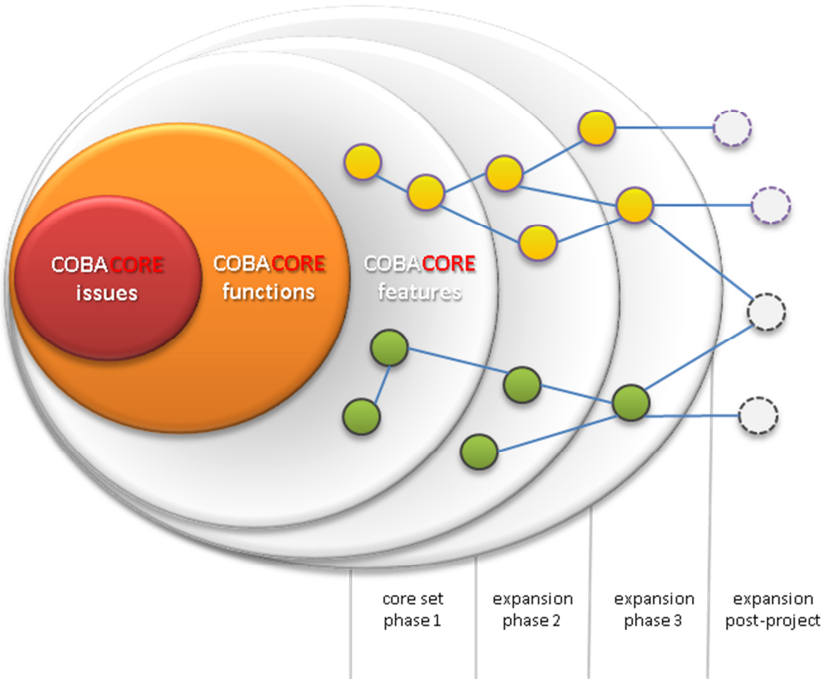


Figure 1: The COBACORE feature development process

The optimal set of features is inherently a result from a collaboration between stakeholders, project concept designers and technical project partners.

It is important to note that features do not equal implementations. Implementations need to satisfy the properties of the features, but might take on widely different shapes. For every suggested feature, there are multiple ways to implement them. Possible Implementations might differ in terms of process workflow, interface layout or technical design. The form in

which a feature is eventually implemented will result from a collaboration between project concept designers, technical engineering teams and stakeholders.

Features are developed in an incremental fashion. The table below describes the three main sets of features (as also shown in Figure 1).

Feature set	Description
Core set	The core set of core features represent the minimum functional requirements that the platform implementation should fulfil, and
Expansion set	Incremental set of features that build upon the core set of features.
Post-project set	Sets of features that have been recognised, but will not be implemented during the project.

Table 3: COBACORE feature development phases

The core set of features represents the foundation of the platform. When implemented, the core set of features represents the minimal form of the platform that satisfies the functions.

Chapter 4 describes features of the COBACORE platform in textual mock-ups, and thus gives an impression of the functional behaviour and form of the COBACORE platform. First, however the core concepts, main behavioural mechanisms that underlay the features, will be described in Chapter 3.

3 Core Concepts

3.1 Introduction

Before defining the platform features in Chapter 4, this chapter first presents the core concepts (mechanisms) that drive the COBACORE platform and its features. The core concepts include Actor, Need, Capacity and Activity. Features on the platform allow users to instantiate these concepts as information objects (in D3.1 called profiles). This leads to four distinctive types of core objects that are of use in the COBACORE platform:

- **Actor object**
- **Need object**
- **Capacity object**
- **Activity object**

Each object is characterised by a number of attributes. For each attribute we give a description and define the associated values. Attributes and values are likely to be changed and reconfigured depending on the deployment situation and based on progressive insights that result from evaluations. Furthermore, objects have relations with each other and may have dynamics in time that will be described in this chapter by state machines.

This deliverable defines the concepts and objects in a semi-structured way. Full specification using a suitable modelling language will be done in WP2. Interface design and implementation of the platform will be done in WP4.

This deliverable expands on the profiles defined in D3.1 Chapter 3.2 “Information structures and views”. To ease reading, this deliverable provided a complete and updated overview of all objects.

3.2 Overview of objects and their interrelations

The figure below shows how the different objects are interrelated. Actor objects can create Need objects and Capacity objects. Capacities can be matched directly to Needs on the marketplace. This is the primary matchmaking form for simple (singular) needs. More complex and compound needs require the further combination of capacities of different actors, and/or some planning and organisation. This is where the Activity object comes in. Activities allow an Actor to broker between Needs and Capacities, by proposing a work plan that addresses one or more Needs and mobilises Actors with the required Capacities. A single Activity may address several identical or similar needs at once. A more detailed description of activities and other objects and how they are related will be given in the next Sections.

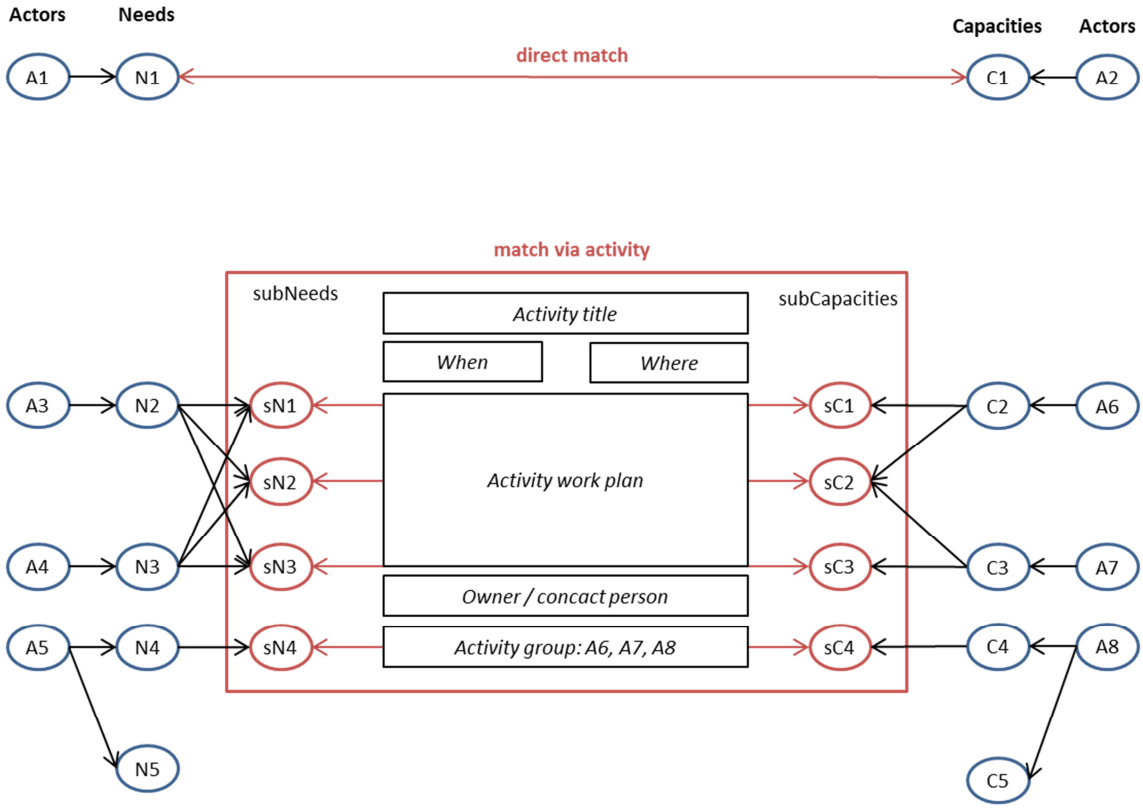


Figure 2: Overview of objects and their interrelations showing the different relations between a direct match between a need and a capacity and a match via an activity

3.3 Actor object

We start this section by giving some definitions:

- We define an *Actor* as an entity that acts on the platform. An actor may or may not have needs, capacities and register or join activities. An Actor can be an individual user or a *group* of users.
- A u*ser is a person* who directly or indirectly makes use of the services of the COBACORE platform.
    - A *registered user* has an account on the platform and can directly act on the platform.
    - A *non-registered* user does not have an account on the platform and indirectly makes use of platform services through a registered user who serves as proxy.
- *Groups* are containers of at least one registered user and may include non-registered users. Groups are actors in themselves but thus also contain at least one actor.

An Actor object is created when a new user registers with the platform or when an existing registered user creates a new Group. Currently, a user can select to be a members of a professional or affected/responding community. In due course, the platform may offer more communities types, such as digital volunteer communities or community liaison teams, but that depends highly on framing of the platform (i.e. what communities do the platform manager perceive as relevant?).

Groups

Groups are containers of at least one registered user and may include non-registered users. Groups are actors themselves but thus also contain at least one actor. See picture below. The actor who creates the group is the group owner and lead contact. The group size is specified as the total of registered and non-registered users. Groups can also be created to register organisations on the platform, with an extra set of attributes. Individual users, staff of these organisations, can than become a member of these groups.

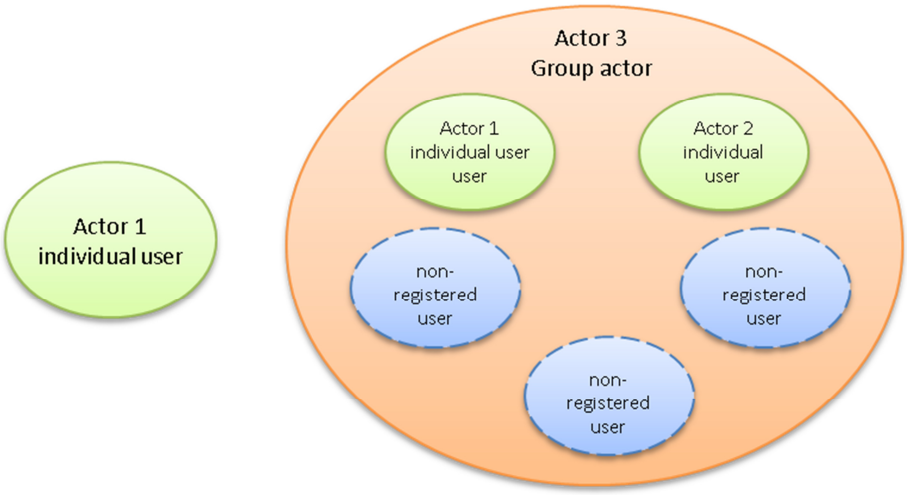


Figure 3: Diagram showing an individual user actor (Actor 1) and a group actor (Actor 3) that represents two individual registered users (Actor 1 and 2) and three non-registered users who are served by proxy.

In the current setup of the platform, groups cannot be nested (i.e. contain subgroups). Nested groups are technically possible, but might not provide any practical value to users. Additionally, nested groups add an additional level of complexity in object structures.

Groups are defined independently from activities. The rationale behind this is that, in mobilising community members, an actor (e.g. a community champion) may first search for people with common interest, drivers, objectives, or location, and once the group has been formed, discuss and start (a number of) activities with this group. Activities are created and closed, but the group can still remain in existence and undertake new activities later on. In the future, it may be interesting to link the group feature to groups already defined on popular platforms like Facebook, LinkedIn and WhatsApp.

To register, organize, join and monitor groups a new core feature (just like need, capacity and activity) is necessary for several frames and functionalities of COBACORE (e.g. community champion, community liaison teams, searching and matching and needs management and monitoring). For these groups a 4th button “Create Group” and new workspace should be created where new groups can be registered, people are invited or can join groups, groups can register activities, members can be informed, etc. The participants of the Group will be provided with a group chat and memo board, to facilitate internal organisation and design. Similar as to needs, capacities and activities there should be an option where you can search for groups that are relevant for you (based on your object) and are suggested to you. Furthermore, groups can invite people to join based on their object, and an invitation message is sent to all the objects that match a specific attribute.

As an example we provide three use cases on groups.

A city floods. When the water is gone the people return to their houses. Several families in a particular street register on the COBACORE platform as individual users and state their needs. They feel that together their voice will be stronger, and ask one family to set-up a group for them. This group does not have a group-need yet (it is a simple way in facilitating the communication between neighbours). During talks the on the COBACORE platform they agree to define a common group need: 'we need the vegetation in our street to return'. Based on this need they may later register an activity to mobilise resources in making this possible.

Martin is an affected community member who is tired of not knowing what is going on in his own street. He talks to his direct neighbours and hears they are dealing with the same problems and he really would like to know if his whole street feels that way. Getting more insight on and combining the similar needs and wishes of his street would make them stronger in their cry out for help. Therefore Martin joins the COBACORE platform and he registers a group with the name of his street: 'help and rebuild Johnson Street'. He sends out messages to all the registered accounts located in his street (maybe even to non-registered persons, just through their email addresses: they would then receive an email to join and register on the platform), asking them to join his group. On his group pages he describes the purpose of the group and starts blogs sharing his idea of combining the needs and questions.

Samantha is a responding community member and connected to the COBACORE platform to help out the affected community members from the flooding in the neighbouring city. Right now the recovery phase has started and she wants to contribute. However she can't find any activities or groups she would like to join. She has some contacts based on some previous activities and through social media they are still in contact. They've decided to focus on the green space in that neighbourhood, and call themselves 'turning Utrecht Green again'. To expand and create activities Samantha start a group within COBACORE. She registers the groups and invites specifically her friends, but also sends out a message to all responding community members if they would like to join her group, when interested in the subject. In the group she proposes new activities like planting trees and cleaning up parks.

3.3.1. Attributes

Object	Attribute	Description	Values
actor object	type	the type of actor	[professional, affected/responding community member]

	name	the name of the actor	Text
	contact details	contact information – relevant contact details	[address, phone number, email address, social media accounts]
	state	state of the object on the platform	[created, archived]
	group?	does the actor represent a group?	yes/no
	if group:		
	group size	size of the group, sum of registered and possible non-registered users	number
	group members	list of the names of the registered users. non-registered users are not named.	list of user names
	Activities	List of activities that the group is involved in, if any	List of activities
	organisation?	does the group represent an organisation?	yes/no
	if organisation:		
	type of organisation		public/government, NGO, company, association, etc.
	organisation mission / objective	Including beneficiary / target group	text
	type of services offered		text
	area served		Country/State/province
	no. of staff		Number categories

Table 4: Suggested properties for the actor object

The sign-up **forms on the current prototype platform differentiate** between 'Affected/Responding Community Member' and 'Professional Responder Organisation'. Doing so, we can filter out professional users and professional groups (organisations) and e.g. list them to create more organisational awareness, or perhaps plot their service to obtain a better sense of professional coverage**.** The separation in type of users may also be useful to provide different interfaces and service packages and pricings later on. We assume Professional Responders to represent an organisation. Affected or Responding Community members may be individuals or other (non-response) organisations, such as companies, knowledge institutes, and so on.

3.3.2. Relation with other objects

An actor may have a relation with needs, capacities and/or activities. See also figure 2 in Section 3.2.

3.3.3. Dynamics

Just like all objects, an actor object can be created, modified, archived and deleted. In addition an actor object may need to be validated, e.g. by confirming from a valid email address. Alternatively, a validated actor can be directly created when using an identity provided by a trusted third party (such as OpenID, Google, Facebook, Twitter, Microsoft, etc.) during the registration process. For the current demonstration platform we define only two states: created and archived.

3.3.4. User interface considerations and screenshots

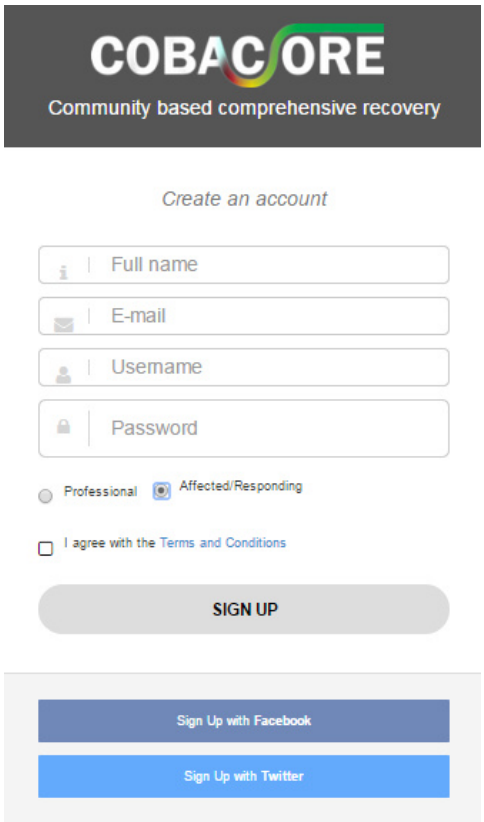


Figure 4: Screenshot of sign up form

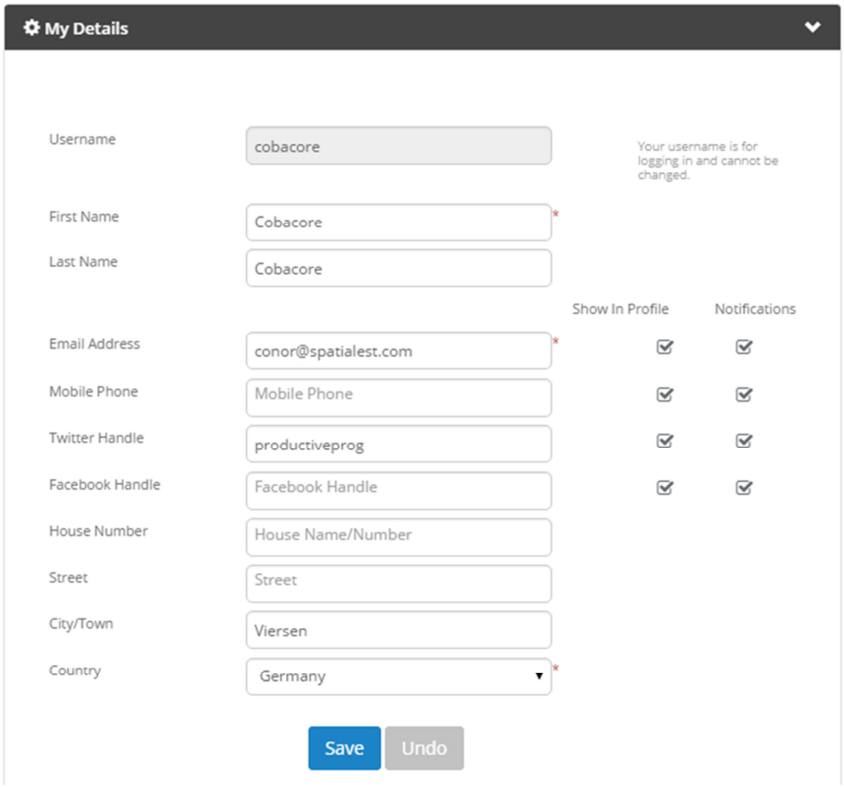


Figure 5: Screenshot showing part of the attributes of an Actor object

3.4 Needs object

An actor can register a need and ask for help. Subsequently, a need can be of various types, like physical needs, need for labour or service, and information needs. For reporting, assessment, and internal matching purposes each need will be assigned a type and category (recovery domain). Type and categories may be entered by the actor, may be fully automatically assigned based on the description entered by the actor by using semantics, or something in between. One of the proposed options is to present the actor with a range of category and type choices, which may be used for the explicit selection of one or more categories and types. These will be subsequently used to automatically determine the most appropriate categories for the Need/Capacity, which can be further corrected by the user at a later stage. Another option that is being explored is that of automated categorisation through a semantic engine. This approach is technically more complex, but seems promising and might make the content registration and matching features in the platform far friendlier to the user.

Several other attributes are recorded to allow others to make a need assessment and/or organise response. From experiments and discussions conducted during Partial Evaluations, we learned that the WHEN attribute (time and date, urgency) was not used appropriately. People typically assign their own need as urgent and/or leave it at its default option. At the time of writing, the urgency attribute has been removed from the user's view because of possible biased use. For now, an auto-archiving is being implemented that ensures that all presented needs are (more or less) current, and need attention.

3.4.1. Attributes

Object	Attribute	Description	Values
need object	type	the type of need	[service, information, supply]
	category	the recovery domain to which the need is related	[Institutional and governmental, Mobility and transport, Vital and critical infrastructure, Social, cultural and educational, Healthcare, Economic, Environmental, Safety and security]
	description	a concise description of the need	text
	registrant	the actor that has registered the need	<association with actor object>
	owner	the actor that has the need, can be the registrant, someone else, or a group.	<association with actor object>
	location	the location where the need exists. For a need this denotes the area where the need is present – not the location where the need is registered	a geographical coordinate or an area.
	state	Current state in the dynamic state machine	[open, addressed, resolved, archived]

Table 5: Suggested properties for the need object

3.4.2. Relation with other objects

A Need has a relation with an Actor and may have a relation with one or more Capacities and/or Activities. See figure 2 in Section 3.2

3.4.3. Dynamics

The dynamics (life cycle) of a Need is illustrated by the state diagram in the figure below.

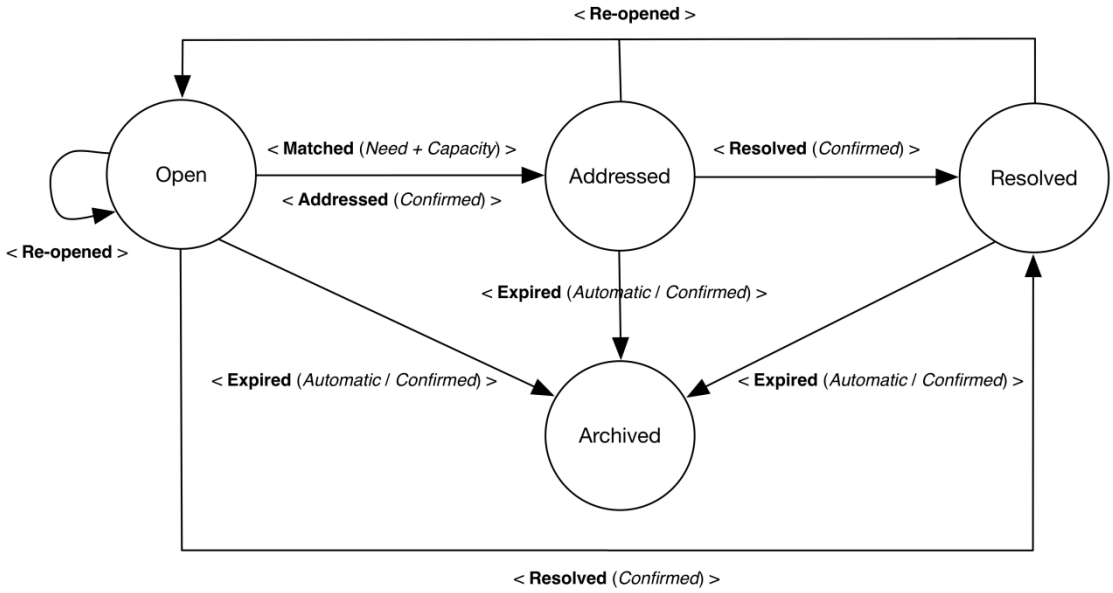


Figure 6: Need state diagram

The following state transitions may occur:

- An Actor registers a Need on the COBACORE platform, resulting in the Need being placed in an Open state.
- The Need will transition to the Addressed state if the Need is matched with a suitable Capacity. The Owner receives a message with the Capacity offered and will need to confirm that the offered Capacity suitably addresses the Need (e.g. is a match). Also, an Activity in progress will trigger associated Needs to transition to the Addressed state.
- The Need will transition from the Addressed state to the Resolved state once the Need is resolved. Needs may be addressed directly, or via an Activity that has been Accomplished. The owner of the Need is required to confirm (via expiry message or self-triggered in his MyNeeds management) that the Need has been resolved
- To keep the information on the platform actual, Needs will be automatically archived after a certain period (let's say 1 month, though the period may differ per deployment case). Regardless of the present state (Open, Addressed or Resolved), the Need will transition to the Archived state, unless the owner of the Need re-opens the Need and thereby resets the lifetime of the Need. One week before auto-archiving the owner will be informed of the upcoming auto-archiving date and has the option to: 1) re-open the need, 2) correct the current state by setting it to Addressed or Resolved, before being auto-archived. If the owner doesn't react, the Need will automatically transition to Archived.

3.4.4. User interface considerations and screenshots

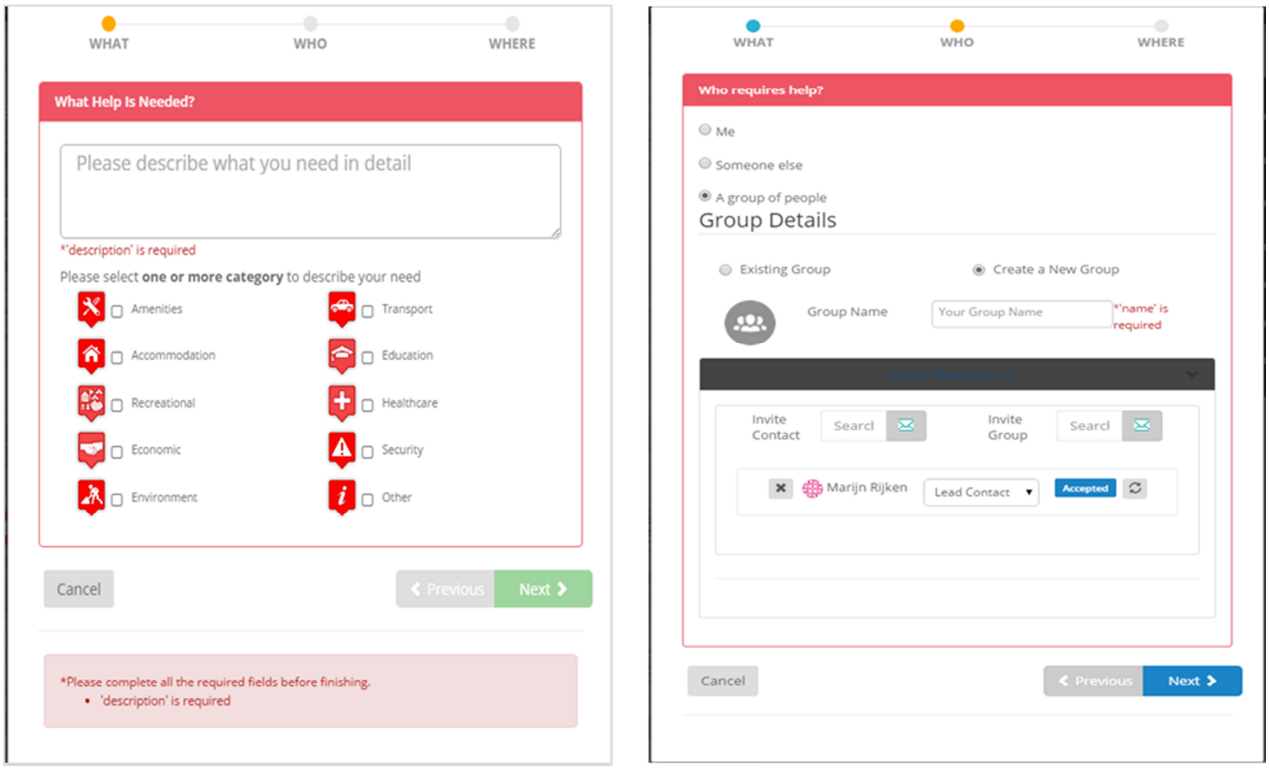


Figure 7: Screenshots of the Needs registration wizard

3.5 Capacity object

An actor can register a capacity to offer help. Capacities can be of various types, like physical capacities, services or information and apply to one or more categories (recovery domains). Type and categories may be entered by the actor, may be fully automatically assigned based on the description entered by the actor by using semantics, or something in between. One of the proposed options is to present the actor with a range of category and type choices, which may be used for the explicit selection of one or more categories and types. These will be subsequently used to automatically determine the most appropriate categories for the Need/Capacity, which can be further corrected by the user at a later stage. Another option that is being explored is that of automated categorisation through a semantic engine. This approach is technically more complex, but seems promising and might make the content registration and matching features in the platform far friendlier to the user.

Several other attributes are recorded to allow others to make a need assessment and/or organise response. From experiments and discussions conducted during Partial Evaluations, we learned that the WHEN attribute (time and date, urgency) was not used appropriately. People typically assign their own need as urgent and/or leave it at its default option. At the time of writing, the urgency attribute has been removed from the user's view because of possible biased use. For now, an auto-archiving is being implemented that ensures that all presented needs are (more or less) current, and need attention.

3.5.1. Attributes

Attributes Object	Attribute	Description	Values
capacity object	type	the type of capacity	[physical, labour, information, instruction, service, financial, expertise]
	category	the recovery domain to which the capacity is related	[Institutional and governmental, Mobility and transport, Vital and critical infrastructure, Social, cultural and educational, Healthcare, Economic, Environmental, Safety and security]
	description	a concise description of the need	text
	registrant	the actor that has registered the capacity	<association with actor object>
	owner	the actor that has the capacity, can be the registrant, someone else, or a group.	<association with actor object>
	location	the location where the capacity is available. For a capacity this denotes the deployment area – e.g. where an actor would be available to deploy his capacity.	A geographical coordinate or an area.
	state	Current state in the dynamic state machine	[offered, addressed, used, archived]

Table 6: Suggested properties for the capacity object

3.5.2. Relation with other objects

A Capacity has a relation with an Actor and may have a relation with one or more Needs and/or Activities. See the figure in Section 3.2

3.5.3. Dynamics

The life cycle of a capacity is very similar to the one of a need. The biggest difference is that, depending on the type of capacity, it may be used multiple times. So it can transition from used to offered or available again.

The capacity life cycle is illustrated by the state diagram in the figure below.

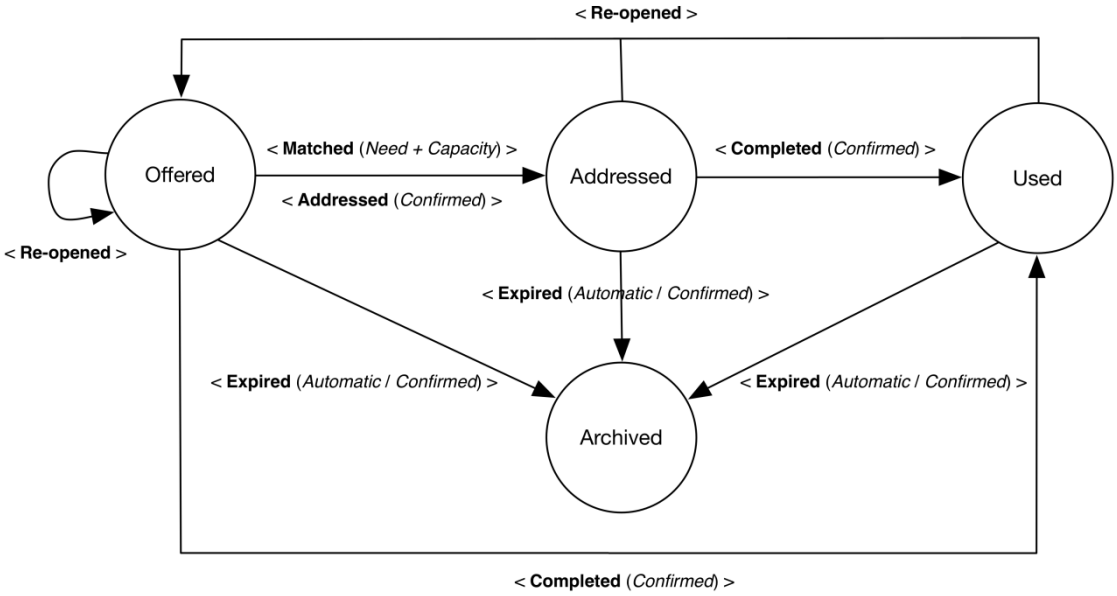


Figure 8: Capacity state diagram

The following state transitions may occur:

- An Actor registers a Capacity on the COBACORE platform, resulting in the Capacity being placed in an **Offered** state.
- The Capacity will transition to the **Addressed** state if the Capacity is *matched* with a suitable Need. The owner receives a message with a Need asked for and will need to confirm that his/her Capacity is suitable (i.e. is a match).
- The Capacity will transition from the Addressed state to the **Used** state once an exchange or an Activity has been *completed.* The owner of the Capacity will need to confirm (via expiry message or self-triggered in his MyCapacities management) that the Capacity is in use or has been used. The owner may also indicate that his Capacity is still available by resetting the state to Offered.
- To keep the information on the platform actual, Capacities will be automatically archived after a certain period (let's say 1 month, though may differ per deployment case). Regardless of the present state (**Open**, **Addressed** or **Used**), the Capacity will transition to the **Archived** state, unless the owner of the Need re-opens the Capacity and thereby resets the lifetime of the Capacity. One week before auto-archiving the owner will be informed of the upcoming auto-archiving date and has the option to: 1) re-open the Capacity, 2) correct the current state by setting it to Addressed or Used before being auto-archived. If the owner doesn't react, the Capacity will automatically transition to Archived.

3.5.4. User interface considerations and screenshots

The wizard is very similar to the Register Need wizard. Therefore it is not replicated here.

3.6 Activity object

An actor can register an Activity to propose a plan to address a need, and optionally combine other needs and capacities in this plan. An activity is registered by an actor (individual or on behalf of a group). The activity can be created by a need owner, but it is also possible for another actor, e.g. a responding community member (a community champion), to create the activity and mobilise volunteers. When an activity is registered it may be associated with one or more needs and one or more capacities (based on whether relevant capacities are already available). It can further be grouped by a common attribute/type (e.g. recovery domain, information, instruction, service, expertise, etc.) and has a stated date/time, location and status, just like needs and capacities.

To keep registration of activities as easy as possible, type and category could be automatically assigned based on the description entered by the actor by using semantics. To adjust wrong categorisations, the actor could be prompted with the chosen type and category and given the option to correct these. However, there are alternative paths that are being explored, such as basing the activity category on the categories of the associated needs and capacities. This approach might make more sense as it is unlikely that an activity would have a vastly different category than key associated needs. In further development and evaluations sessions, we will investigate the practical value of either approach.

Several other attributes are recorded to allow others to make a need assessment and/or organise response. An activity consists of:

- a title (WHAT)
- a place (WHERE)
- a date/time (WHEN)
- a description: a plan that mentions what is needed (free text) (HOW)
- an owner / contact person (WHO)
- participants (WHO) (optional)
- associated top level needs (WHY) (optional)
- associated top level capacities (WHY) (optional)
- Nice to have: sub-needs (WHY) (optional)
- Nice to have: sub-capacities (WHY) (optional)

See also the figure in Section 3.6.2 for an Activity and its relations with needs and capacities.

An activity may be created to address one or more needs in the platform. E.g. an activity is created to address the existing need “I need a house”. This activity proposes a work plan, and may break the need down into sub-needs like bricks, mortar, manpower, etc. Capacities can then be assigned to those sub-needs when these are available. The availability of capacities for these sub-needs has consequences for the lifecycle of an activity. Activities are useful to deal with more complex (compound) needs or a multiple of needs and/or capacities, and allow an actor to broker between these and propose work plans. More simple (singular) needs and capacities can be matched directly one on one without an activity. In future versions of the platform, and activity, may be improved by others (co-creation) by adding sub-needs and/or sub-capacities.

We call needs and capacities already available on the platform *top level* needs and capacities. Needs and capacities that are part of an activity are referred to as *sub* needs and capacities. They may be implemented with the same information object (to allow suggestion and matching algorithms to work on activities as well), but sub level needs and capacities do not appear in (top level) Needs lists, maps, needs assessments, etc. Sub-needs and sub-capacities

are “nice to have”, however noting the complexity both in implementation and usage, they will not be implemented in the prototype COBACORE platform. In this version, sub-needs and sub-capacities can just be mentioned in free text in the description field. The purpose of the Activity feature is to bring top level needs, top level capacities and actors together in combined effort. It is at the time of writing still undecided whether management of sub-needs and sub-capacities (like ‘are all sub-needs of my activity resolved?’) are managed within the platform, or regarded as the responsibility of the user.

Participants are invited to join the activity. In the current implementation of the platform, participants would be existing actors. In practice, it would be valuable to have the options to invite people who have not yet registered on the platform, for instance via email or social media notice. This type of invite is commonplace nowadays and would add to quickly building up a solid user-base.

To register, organise, join and monitor activities is a new base feature (just like needs and capacities) and is necessary for several frames and functionalities of COBACORE (e.g. community champion, community liaison teams, searching and matching and needs management and monitoring). For these activities a new workspace should be created where new activities can be registered, activities can be monitored, people can join activities and people who are linked to activities can be informed. These activities are tied in with needs and capacities. Similar as to needs and capacities there should be an option where you can search for activities that are relevant for you (based on your object info such as location, availability, profession, skills, tools, resources) and are suggested to you. Finally, professionals can monitor activities.

The participants of the Activity will be provided with a group chat, memo board and task list to facilitate internal organisation and design.

Important aspects:

- An activity may or may not be associated with needs and capacities.
- An activity cannot exist without a group. In other words: each activity has its own dedicated group. Existing groups may be invited, but members will become members of the new dedicated group, as groups cannot be nested.
- These activities pages should be somehow visible for the professionals to monitor and support if necessary. Professionals might have the option to add information that is not visible to the active members (e.g. for attending partner professionals on implications and monitoring prioritization)

We provide two use cases on activities:

Marc is a responding community member and currently individually active on the platform. He has been helping affected community members in the surrounding flooded area with transportation. He is offering his car, food and time and has helped over a dozen people already. Now one of the local communal parks is still littered with rubble. The playgrounds are unusable and the kids-farm is having a hard time getting back to its original luster. The government has given higher priority to other recovery activities and has delayed cleaning and restoration efforts. Together with some other responding friends he came up with the idea to do it themselves. He goes to the COBACORE platform and signs in. Signed in he goes to the option to register a new activity. For the registration he is asked for the following details: description of activity and category, contact lead of activity, members for the group, location where

your activity will take place, date when you'll start. He invites all his friends for the activity and fills in their COBACORE user names or email addresses in case they are not yet registered on the platform. The activity is registered and all his friends receive a notification, which they have to respond with yes to really join. The activity is visible for all COBACORE users. When clicking on the activity they can join and be informed about new information.

Iris is a responding community member who would like to help out the affected community members. She lives in a different country but has time and the ability to travel towards the affected area. However she does not yet know how she can be of any help. Therefore she visits the COBACORE platform and registers as an actor but doesn't register any capacities yet). When registered she goes to the list of activities and browses through them. She finds an activity that inspires her: help to restore a school. It asks for people to join, no specific expertise is needed, only hands. When interested please join. Iris can simply join the activity, which takes her to the activity page: Here a description, person of contact, extra information is available, in addition to a list of all the members of this activity. On this page Iris is informed about actions in this activity and can also join a chat group.

3.6.1. Attributes

Object	Attribute	Description	Values
activity object	type	the type of activity	[to be determined]
	category	the recovery domain to which the category is related	[Institutional and governmental, Mobility and transport, Vital and critical infrastructure, Social, cultural and educational, Healthcare, Economic, Environmental, Safety and security]
	title	the colloquial title of the activity.	text
	description	a concise description of the work plan	text
	status	the status of the activity	[scheduled, in progress, accomplished, Archived]
	needs	the needs that are associated with this activity, if any	<association with need objects>

	capacities	the capacities that are associated with this activity, if any	<association with capacity objects>
	sub-needs	Needs internal to the activity, if any	<association with need objects>
	sub-capacities	Capacities internal to the activity, if any	<association with capacity objects>
	group	the actors that take part in the activity	<association with actor objects>
	owner	the actor that is responsible for this activity and that can be contacted	<association with actor objects>
	time and date	the timeframe in which the capacity exists (might be open-ended, might be recurring)	start and stop time and date.
	place	The place where the activity takes place	location
	state	Current state in the dynamic state machine	[scheduled, in progress, accomplished, archived]

Table 7: Suggested properties for the activity object

3.6.2. Relation with other objects

An Activity has a relation with one or more Actors (who are contained in the dedicated activity group), and may have a relation with Needs and Capacities. The figure below shows how an Activity is used to:

1. combine the identical Needs N2 and N3 and similar N4 and address them with a single Activity
2. break down complex compound Needs N2 and N3 into subNeeds SN1..SN4 and define subCapacities sC1..sC4.
3. Map Capacities C2, C3 and C4 on the subCapacities sC1..sC4 and invite Actors A6, A7 and A8 to join the activity.
4. Make use of existing actors (groups and individual actors) to form a new dedicated activity group.

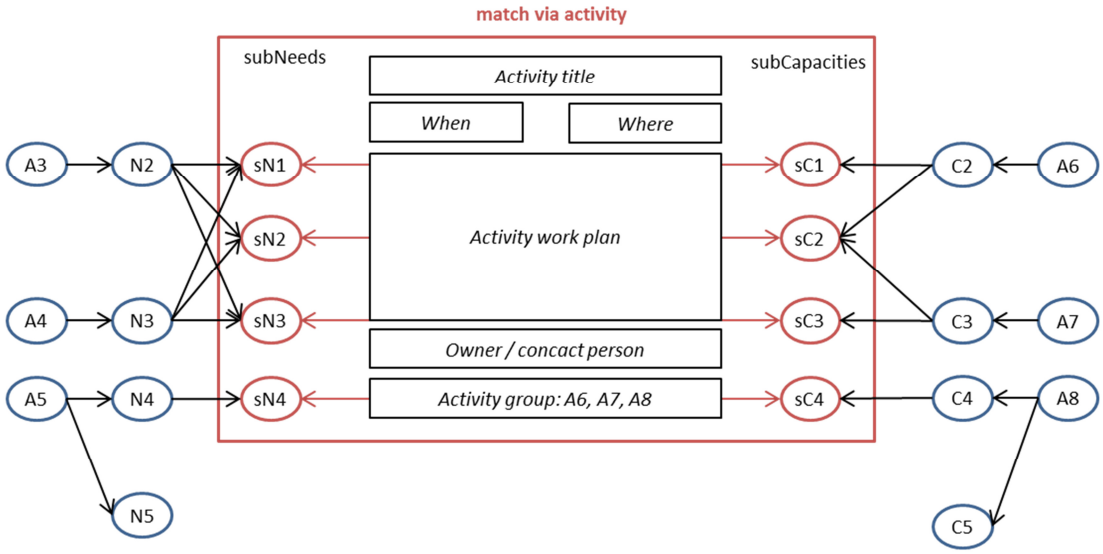


Figure 9: Overview of objects and their interrelations

3.6.3. Dynamics

The names of the states in the life cycle of an Activity are again slightly different than for a Need and Capacity. Also additional mechanisms for transitions between the **Scheduled** state and **In progress** state are used. See the state diagram below.

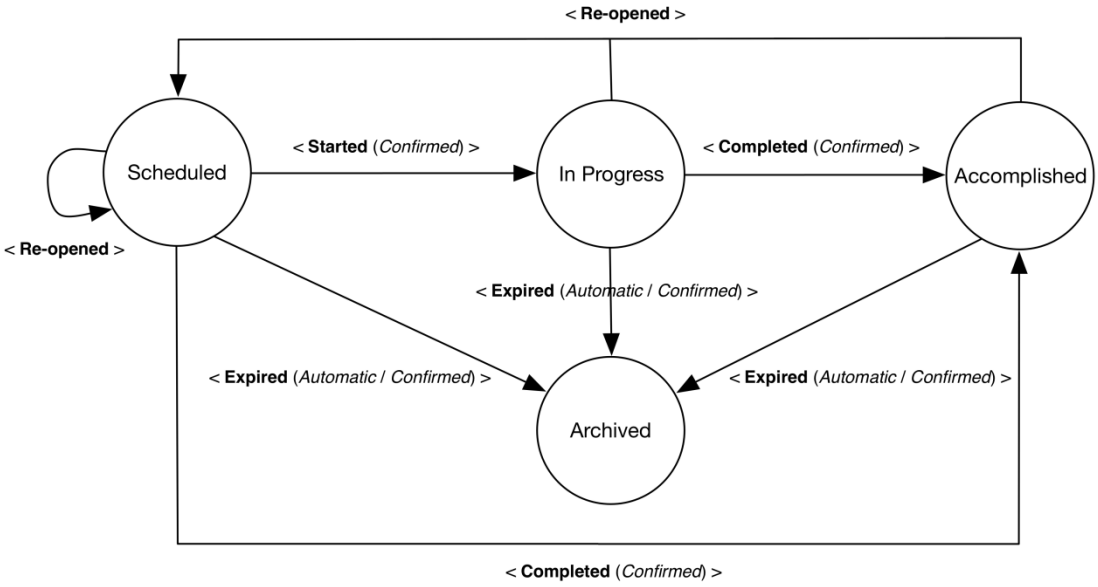


Figure 10: Activity state diagram

- An actor registers an Activity on the COBACORE platform, resulting in an Activity object in a **Scheduled** state. When an Activity is registered it may or may not directly be associated with one or more needs and capacities (based on whether these are already registered on the platform).

- An Activity transitions to the **In progress** state when an actor confirms the transition. This is either when the actor places the Activity In progress him-/herself, or when the actor confirms by answering a pushed message sent on the associated start date of the activity. The pushed message asks if the activity is in Progress, or maybe even accomplished (if so the Activity will be transitioned to the **Accomplished** state).
- The Activity will transition to the **Accomplished** state from the In progress state once the activity has been completed. The owner of the Activity needs to confirm, via a message sent on the associated end date of the activity or self-triggered in his My Activities management, that the activity has been completed.
- After a certain period, e.g. one month, the owner of the activity receives a message asking if the Activity is still open, is in progress, accomplished, or can be archived. The owner may reset the lifetime of the activity by confirming that the activity is still open. If no reply is given, the Activity will automatically be **Archived**.

Relation to state diagram of associated Needs and Capacities:

- If an Activity is In Progress, the Needs associated with that Activity (that are also registered on the COBACORE platform) would be placed in the Addressed state. The same applies for the transition to the Resolved and Archived states.
- If an Activity is In Progress, the Capacities associated with that Activity (that are also registered on the COBACORE platform) will be placed in the Addressed state. The same applies for the transition to the Used and Archived states.

3.6.4. User interface considerations and screenshots

- The main site on the platform has buttons to register a new need, and new capacity, a group, or a new activity.
- Activities should be shown on the map with a distinct pictogram and distinct colour.
- ‘Activities’ is another tab in “my profile”
- Each activity has its own page with information: POC, description, group (members), dates, group chat and memo features and the option to email all the members at once to ask questions.
- Ideally existing Needs and Capacities can be associated to an activity by drag and drop: drag them from the map into the activity wizard screen. If this is not feasible, a search/ dropdown list of existing needs and capacities will do. To improve usability the list should only include relevant Needs and Capacities based on e.g. the location of the activity and semantic matching of descriptive text.

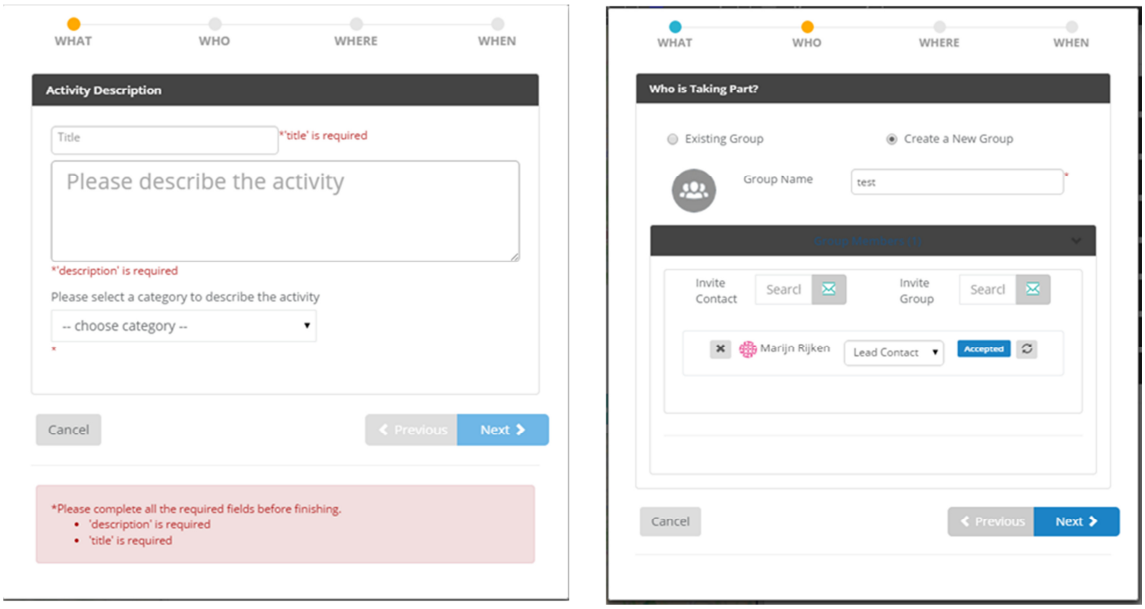


Figure 11: Screenshots of the Activity registration wizard

4 COBACORE feature descriptions

Chapter 4 gives an overview of the COBACORE platform features that have been designed after consultation with stakeholders and evaluation sessions. Section 4.1 describes the core features. Subsequent sections describe features of the expansion set and post-project set. Although it is to be expected that most of these features will be implanted in the prototype, the relevance of prototyping specific features (e.g. for evaluation and/or demonstration) as well as the implementation progress during the final year of the project will determine what features will finally be implemented within the scope of the project.

In the following sections, we describe each COBACORE core features on a number of characteristics. The table below lists the attributes we use to define each feature.

Attribute	Description
identifier	An identifying code
name	The colloquial name of the feature
description	A brief description of its purpose
fulfils function	A specification of the function it fulfils: • U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching • Generic system performance
supports frame	The frame that the feature provide support to: • Needs and Capacities Marketplace • Community Champion • Community Liaison Team • Intelligence and Insight • Learning Environment
primary user community	A selection of the primary target user group of this feature • Professional community • Affected community • Responding community • Platform managers • Trainers
related features	A listing of the core features that have a direct relationship

	with feature that is being described.
required information input	A description of the information that is required for this feature, and a description of the most likely source of that information.
information output	A description of the information that is produced through this feature, and a description of the nature of the information.
interaction patterns and interface elements	A description of the elemental interaction patterns and interface elements that could be used to instantiate the feature.
evaluation criteria	A description of the major evaluation criteria for the feature.
development status	A brief description of the status of development at the time of writing.

Table 8: Description of feature attributes used in this report

On user groups

From the early phases of the project, we have made use of three prototypical user communities: the professional community, the affected community and the responding community. For concept development purposes, we distinguish two further users communities: the 'platform managers' and 'trainers'.

Platform managers are users that have taken on a responsibility to manage the platform, including configuration and moderation tasks and other typical system management assignments. A platform manager can be professional or a non-professional (e.g. volunteers) depending on the deployment form of the COBACORE platform.

Trainers are users that employ the platform for training purposes, e.g. as a supporting instrument. To be able to properly use the platform in training, it needs to be configured according to training specifications. Additionally, the platform needs to have proper scenario and logging capabilities as to provide trainers with the tools to manage a training.

On frames

In the first half of the project, the project team has explored the domain of community-based disaster recovery, and laid out the foundation for the development of the COBACORE platform. The conceptual and technical foundations led to a solid start for platform development, evaluations sessions and a sound basis for interactions with stakeholders. However, as with many innovation projects, the creative process in the first half of the project led to many interesting narratives, and a wide range of options for further development of the COBACORE foundation. The lack of focus led to indecisiveness, and thus action has been taken

to improve project focus: clearer use cases, clearer stakeholder groups, and a clearer perspective on desirable platform functionality.

We have built up a strong set of 'frames' that are used to steer development and tell the 'COBACORE story'. A frame is essentially a vision on how the platform would be used by a group of users, and that can be used to identify preferable platform features.

We have established five distinct frames:

- Needs and Capacities Marketplace. In this frame, the 'needs' and 'capacity' matching by affected and responding communities take center stage, and the COBACORE platform functions primarily as a marketplace during disaster recovery.
- Community Champion. This frame emphasizes the use of the platform to initiate activities and mobilize community members. A 'community champion' from the responding community would use the platform to make plans known and build up a community-based activity.
- Community Liaison Team. In this frame, the focus is on a community liaison team: a team that consists of professionals and trained volunteers that act as intermediary between the various communities and that use the COBACORE platform as its primary information gathering and dissemination platform.
- Intelligence and Insight. The Intelligence and Insight frame accentuate the value of the COBACORE platform for professionals to gain a deeper understanding of the activities that take place among the communities in the affected area. Through metrics and analytical tools that work off the base information in the platform, professionals can make better-informed decision on where to deploy capacities.
- Learning Environment. The COBACORE platform can not only be used as an operational tool, but also as an instrument to train professionals and trained volunteers on interacting with civil communities and parties during disaster recovery and building up effective partnerships.

Throughout the remainder of the project all frames will be further explored and materialised through feature development. The proposed features in this chapter support one or more frames.

On feature categories

The features are subdivided into ten categories.

- **Core features.** The 'core feature set' is a collection of fundamental features that are required to satisfy the key COBACORE platform ambitions.
- **Teamwork support.** Features in this set are aimed at improving teamwork, and enhance team awareness, improve intra-team communication and collaborative work.
- **Content matching and management.** Features in this contribute to the quality of information contained in the platform by providing semantic matching functionalities and options for authorised users to adjust the state of content.
- **User feedback and reputation management.** This set of features is aimed at providing user management functionality to the platform, such as user reporting and rating functionalities and user management options.
- **User mobilisation.** Features under this heading provide options to mobilise users, for example via activity publishing tools.

- **Information provision.** Information provision features provide users with information from relevant sources and provide options to receive notifications.
- **Analytics and sensemaking.** Features in this set are aimed at providing professional users with insightful information about the state of the environment and the activities that take place among the various communities.
- **Instructions and guidance.** These features propose information sharing platforms that can be used by professionals to convey documents and instructions to other communities.
- **Training features.** These features facilitate the use of the COBACORE platform as a training facility, and provide essential functionalities such as scenario management and platform configuration.
- **Platform configuration and administration.** This set of features provides authorised users with options to administer and configure the platform.

The table below provides a reference summary and a note on the current state (at M27). Features F1 through F10 were previously proposed (see D3.1). Features F11 through F35 are newly proposed features, based on evaluation sessions and stakeholder interactions.

Type	id	Feature	Purpose	Status at M27
Core set	**F1/AR**	**Actor registration**	Provides means for users to register actors with the platform. An actor is the bearer of a need, a capacity or an activity.	Implemented in web and mobile clients
	F2/NR	**Need registration**	Provides means for users to register needs related to their recovery from a disaster. They may register needs for themselves, or as representative of a group.	Implemented in web and mobile clients
	F3/CR	**Capacity registration**	Provides means for actors to register capacities that they are willing to provide towards the disaster recovery.	Implemented in web and mobile clients
	F4/NCO	**Needs and capacities**	This feature provides means for users to	Implemented in web and mobile

		overview	view registered needs and capacities, and perform various filtering and sorting operations to establish an informative overview.	clients
	F5/NCM	**Needs and capacities matching**	Provides means for users to establish matches between registered need profiles and registered capacity profiles. After a match has been determined, it can be effectuated as an activity.	Implemented in web and mobile clients
	F6/AAO	**Actors and activities overview**	Provides means for users to view registered actors and activities, and perform various filtering and sorting operations to establish an informative projection.	Implemented in web and mobile clients
	F7/BSO	**Baseline situation overview**	Provides an overview of the situation in the affected area through projection of baseline data and information.	Implemented in web client.
	F8/BRV	**Basic recovery views**	Provides the user with options to build data views that are relevant to understanding the recovery process, and thus provide a basis for further analysis and plan-definition.	Implemented in web client.
	F9/BIE	**Basic information exchange**	Provides means for users to exchange information	In progress

			(documents, data, digital media), centred around mutual capacity profiles and activities.	
	F10/ACT	**Activity registration**	Provides means for users to register an activity with the platform. An activity is a recovery effort in which actors partake.	Implemented in web and mobile clients
Teamwork support	**F11**	**Group Presence Panel**	Show the availability of group members	Implemented
	F12	**Group communication channel**	Enable communication within a group	Implemented
	F13	**Collaboration canvas**	Enable group collaboration	Partly Implemented
Content matching and management	**F14**	**Semantic content matching**	Facilitate matching through semantic suggestions	Partly Implemented
	F15	**Lifecycle management**	Facilities to manage the state of content	Partly implemented
User feedback and reputation	**F16**	**User reporting**	Tools to report users to platform managers	Partly implemented
	F17	**User rating**	Tools to rate peer users	Implemented
	F18	**User management**	Tools to manage platform users	Partly Implemented
Group activity and user mobilisation	**(F10)**	**Activity initiation**	Start group activities	Implemented
	F19	**Activity publishing tools**	Attract relevant actors to an activity	Partly Implemented
Information provision	**F20**	**Information stream**	A panel that provides an ongoing stream on information	Partly implemented

	F21	**Notification panel**	A panel that provides notifications	Implemented
	F22	**Targeted communication**	Communicate to a specific groups of users	Partly implemented
Analytics and sense-making	**F23**	**Annotation of maps**	Build up awareness, share insights by annotating maps	Implemented
	F24	**Status report export**	Compilation of status reports	Partly implemented
	F25	**Gap analysis support**	Build up awareness on gaps in recovery	Rudimentary implementation
	F26	**Recovery analysis view**	Data-driven instruments for recovery analysis	Rudimentary Implementation
	F27	**Situation alerts**	Supportive tools that professionals can use to track events	Not Implemented
Instructions and guidance	**F28**	**Public instruction portal**	A public information repository	Not Implemented
	F29	**Dedicated instruction environment**	Ad-hoc closed environment in which professionals can instruct volunteers/groups	Not Implemented
Training features	**F30**	**Platform feature configuration**	Configuration for training	Not Implemented
	F31	**Scenario manager**	Prepare scenario to be using during training	Not Implemented
	F32	**Performance analysis support**	Post-action performance analysis	Partly Implemented
Platform configuration and administration	**F33**	**Typology configuration**	Configuration of the COBACORE platform typology	Not Implemented

	F34	**Dataset typing**	Configuration of datasets	Not Implemented
	F35	**Information source management**	Configuration of external information sources	Partly implemented

Table 9: A summary of all proposed features for the COBACORE platform

Many of the features that are currently 'partly implemented' will be matured over the remaining project time. Proposed features that are 'not implemented' at this time, might be selected for implementation in the final platform in due course.

The following sections detail the proposed features. Please note that the 'development status' refers to the state at the time of writing of this report. Many existing features will further matured towards the final evaluation, and several others will be added during the final development phase.

4.1 Core features

As mentioned in the introduction, the first phase of the feature development process consists of establishing a core set of features. This set is also referred to as the *'No-Regrets list'*, hinting at the critical quality of these features. These core features represent the minimum functional requirements that the platform implementation should fulfil and is the result from stakeholder consultations and project-internal concept design activities in which multiple options are assessed. Apart from the updated feature 10 'Activity Registration' and some terminology updates, this section is identical to the descriptions of core features in D3.1. It has been included to increase readability and for the sake of completeness.

In the following sections, we describe each COBACORE core features on a number of characteristics.

4.1.1. Feature 1: Actor registration

Attribute	Description
Identifier	F1/AR
Name	**Actor registration**
Description	This feature provides means for users to register actors with the platform. An actor is the bearer of a need, a capacity or an activity. Users can register themselves as actors, someone else, or a group, and will be asked to provide information about the actor.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching

primary user community	• Responding community • Professional community • Affected community
related features	• F2/Need registration • F3/Capacity registration • F6/Actors and activities overview • F10/Activity registration
required information input	• user input
information output	The output of this feature is an **actor object**. An actor object contains information about the actor, such as contact information, location, and other relevant characteristics. Note that **need** and **capacity objects** are not part of the actor object, but are intrinsically linked.
interaction patterns and interface elements	A **form** through which a user can provide relevant personal details and contact information. Input accelerators could be used where applicable (e.g. using map-based selections to obtain location information).
evaluation criteria	**Quality of the resulting actor object** – does the object contain sufficient information to capture the actor's characteristics and engage the actor in follow-up activities?

4.1.2. Feature 2: Need registration

Attribute	Description
Identifier	F2/NR
Name	**Need registration**
Description	This feature provides means for users to register needs related to their recovery from a disaster. They may register needs for themselves, or as representative of a group.
fulfils function	• U1/Enhance information exchange • U3/Improve needs and capacities matching
primary user community	• Affected community
related features	• F1/Actor registration • F4/Needs and capacities overview

	• F5/Needs and capacities matching
required information input	• associated registered **actor object** • user input
information output	The output of this feature is a **need object**. A need object includes information about the nature of the need, the location where the need exists, the actor associated with the need and other relevant information.
interaction patterns and interface elements	**Form-based input.** The user is taken through a number of entry-fields in a form that help to register the various characteristics of a need. Some entries might be constrained to predefined values, such as the recovery domain to which the need belongs. Other entries might be supported by input accelerators, such as map-based input to gather geospatial information. At the end, the user is presented with the final need object.
evaluation criteria	**Quality of the resulting need object** – does the object sufficiently capture the actor's need? Does the object capture enough information for follow-up activities?

4.1.3. Feature 3: Capacity registration

Attribute	Description
Identifier	F3/CR
Name	**Capacity registration**
Description	This feature provides means for users to register capacities that they are willing to provide towards the disaster recovery.
fulfils function	• U2/Facilitate collaboration support • U3/Improve needs and capacities matching
primary user community	• Responding community • Professional community
related features	• F1/Actor registration • F4/Needs and capacities overview • F5/Needs and capacities matching • F9/Basic information exchange
required information input	• associated registered **actor object**

	• user input
information output	The output of this feature is a **capacity object**. A capacity object includes information about the nature of the capacity, the location where the capacity exists, the actor associated with the capacity and other relevant information.
interaction patterns and interface elements	**Form-based input.** The user is taken through a number of entry-fields in a form that helps to register the various characteristics of a capacity. Some entries might be constrained to predefined values, such as the type of capacity that is being offered, or the recovery domain to which the capacity belongs to. Other entries might be supported by input accelerators, such as map-based input to gather geospatial information. At the end, the user is presented with the final capacity object for a check.
evaluation criteria	**Quality of the resulting capacity object** – does the object sufficiently capture the actor's capacity? Does it object enough information for follow-up activities?

4.1.4. Feature 4: Needs and capacities overview

Attribute	Description
Identifier	F4/NCO
Name	**Needs and capacities overview**
Description	This feature provides means for users to view registered needs and capacities, and perform various filtering and sorting operations to establish an informative overview.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching
primary user community	• Responding community • Professional community • Affected community
related features	• F1/Actor registration • F2/Need registration • F3/Capacity registration • F5/ Needs and capacities matching

required information input	• registered **actor objects** • registered **need objects** • registered **capacity objects**
information output	An output of this feature is a presentation view: a **needs and capacities view.** Another output of this feature could be a subset of the registered needs and capacities, based on a certain query. This subset could be used by other features for further processing (for instance: select a group registered needs and capacities based on a geographic location, and send associated actors supporting information through F10: basic recovery information exchange).
interaction patterns and interface elements	This feature presents registered needs and capacities in a **list form**, with **filtering options**, **sorting options** and **search options** to allow the user to define relevant selections. This feature might include additional views, such as basic **map projections** to plot registered needs and capacities on a map using their registered location. Users can also view further details of the registered needs and capacities, such as time- and date-stamps, associated actor characteristics, categories and so on.
evaluation criteria	**Information presentation quality** and **usability**. Does the feature present the available information in a suitable manner, and do the item management functions provide enough means to obtain the preferred view on the data.

4.1.5. Feature 5: Needs and capacities matching

Attribute	Description
Identifier	F5/NCM
Name	**Needs and capacities matching**
Description	This feature provides means for users to establish matches between registered need objects and registered capacity objects. After a match has been determined, it could be effectuated by sending out requests for contact, or some other form of establishing contact.
fulfils function	• U3/Improve needs and capacities matching
primary user community	• Responding community

	• Affected community
related features	• F1/Actor registration • F2/Need registration • F3/Capacity registration • F4/Needs and capacities overview
required information input	• Registered **need objects** • Registered **capacity objects**
information output	This feature produces established matches between needs and capacities. Such a match can be effectuated as an activity, and registered in the system as an **activity object.**
interaction patterns and interface elements	Based on the overview provided through Need & Capacity Overview (F4/NCO), this feature provides means to an authorised user to connect a need with a capacity. A user is authorised to make connections for needs and capacities that are associated to his actor object. Therefore, a user could self-establish a match between his registered need (e.g. the need for transport) with the capacity that another actor is providing (e.g. a car, or a transport service). A match can be made via a number of **selection steps**, or via a **wizard**-type interaction. Once a match has been established by the actor, an activity proposition sequence is started via a **wizard**, or a similar method. In this sequence, the user provides a justification for the match, and can affirm the activity by sending out a request for contact to the receiving actor.
evaluation criteria	**Information presentation quality** and **usability**. Does the feature present the available information in a suitable manner, and does the item management function provide enough means to obtain the preferred view on the data.

4.1.6. Feature 6: Actors and activities overview

attribute	Description
identifier	F6/AAO
name	**Actor and activity overview**
description	This feature provides means for users to view registered actors and activities, and perform various filtering and sorting operations to establish an informative projection.

fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching
primary user community	• Responding community • Professional community • Affected community
related features	• F1/Actor registration • F4/Needs and capacities overview • F5/Needs and capacities matching • F10/Activity registration
required information input	• registered **need objects** • registered **capacity objects** • registered **actor objects**
information output	Presentation view: **an actor and activity view.** Another output of this feature could be a subset of the registered actors and activities, based on a certain query This subset could be used by other features for further processing (for instance: select a group registered actors based on their geographic location, and send this supporting information through feature F9/Basic information exchange).
interaction patterns and interface elements	This feature presents registered actors and activities in a **list form**, with **filtering options**, **sorting options** and **search options** to allow the user to define relevant selections. This feature might include additional views, such as basic **map projections** to plot registered actors and activities on a map using their registered location. Users can also view further details of the registered actors and activities, such as time- and date-stamps, associated type or other relevant properties.
evaluation criteria	**Information presentation quality** and **usability**. Does the feature present the available information in a suitable manner, and do the item management functions provide enough means to obtain the preferred view on the data.

4.1.7. Feature 7: Baseline situation overview

attribute	Description
identifier	F7/BSO

name	**Baseline situation overview**
description	This feature provides an overview of the situation in the affected area through projection of baseline data and information.
fulfils function	• U1/Enhance information exchange
primary user community	• Professional community
related features	• F8/Basic recovery views
required information input	Baseline data-sources that contain data about the affected area. This may include data from the pre-crisis state and information about the current state.
information output	Presentation view: **Baseline situation overview**
interaction patterns and interface elements	This feature would be best-implemented using map-based and graph-based projections of situational data. Additionally, **textual representations** of relevant operational information could be added, and **selection means** to select a time-frame.
evaluation criteria	**Information presentation quality** and **usability**. Does the feature present the available information in a suitable manner, and do the item management functions provide enough means to obtain the preferred view on the data.

4.1.8. Feature 8: Basic recovery views

attribute	description
identifier	F8/BRV
name	**Basic recovery views**
description	This feature provides the user with options to build data views that are relevant to understanding the recovery process, and thus provide a basis for further analysis and plan-definition. The *Baseline situation overview* feature supports this feature.
fulfils function	• U1/Enhance information exchange
primary user community	• Professional community

related features	• F7/Baseline situation overview
required information input	• registered **need objects** • registered **capacity needs** • **baseline data sources**
information output	Presentation view: **Basic recovery view**: a view in which various information sources are projected in an integrated manner as to obtain a deeper understanding of the situation. For instance: a hotspot projection of needs onto the **baseline situation view.**
interaction patterns and interface elements	This feature would be best implemented using a **map-based and graph-based projection** of need and capacity registrations. Additionally, **textual representations** of relevant operational information could be added, and **selection means** to select a time-frame.
evaluation criteria	**Information presentation quality** and **usability**. Does the feature present the available information in a suitable manner, and do the item management functions provide enough means to obtain the preferred view on the data.

4.1.9. Feature 9: Basic information exchange

attribute	description
identifier	F9/BIE
name	**Basic information exchange**
description	This feature provides means for users to exchange information (documents, data, digital media), centred around mutual recovery interests. For instance, this feature could provide professionals with the means to convey information about capacity deployment (e.g. instructions, tutorials, suggestions for action) to selected other community members, or request specific information. Conversely, responding community members (e.g. volunteers) can request supporting information concerning their offered capacities from professional community members.
fulfils function	• U2/Facilitate collaboration
primary user community	• Affected community

	• Responding community • Professional community
related features	• F4/Needs and capacities overview • F6/Actors and activities overview
required information input	• registered **actor objects** • registered **capacity objects**
information output	This feature results in the exchange of digital content that is related to capacity deployment or recovery progress. These materials might include training instructions, general preparatory advice, and other relevant information.
interaction patterns and interface elements	This feature might take the shape of a direct information exchange between parties, or an indirect exchange that makes use of a document repository. For the direct information exchange, the feature would require a receiving **contact search option**, a **document upload, annotation,** and **transfer management** option on the sending side. On the receiving end, a **notification of reception** and **document management options** is required (e.g. open, store, annotate, delete). For indirect information exchange, the feature would require on the sending side: means to **categorise and annotate** and **upload** digital materials to a repository. On the receiving end, the user should have **filtering options** to select relevant materials, and **document download and management options** to obtain the materials.
evaluation criteria	**Quality and usability of exchange process.** Does the process lead to the right information with the right actor, and is the process organised in an effective and efficient manner?

4.1.10. Feature 10: Activity registration

Attribute	Description
identifier	F10/ACT (updated after D3.1)
name	**Activity registration**
description	This feature provides means for users to register an activity with the platform. An activity is a recovery effort in which actors partake. This feature might be materialised

	as an explicit functionality (in which the user is given the option to explicitly register an activity), or an implicit functionality (in which an activity results from the matching of a need with a capacity). Both varieties result in an activity object that includes essential properties such as time-specification, associated actors, needs and capacities, location and description.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching
supports frame	• Community champion • Community Liaison Team
primary user community	• Responding community • Professional community
related features	• F1/Actor registration • F2/Need registration • F3/Capacity registration • F5/Needs and capacities matching • F6/Actors and activities overview • F9/Notification and information exchange
required information input	• User input
information output	The output of this feature is an **activity object.** An activity object contains information about the activity, such as contact information, location, and other relevant characteristics. The resulting activity can be viewed through views such as the *actors and activities* view, and can be localised on the main map.
interaction patterns and interface elements	A **form** through which a platform-user can provide relevant activity details. Input accelerators could be used where applicable (e.g. using map-based selections to obtain location information, or a clock to select valid timeframes).
evaluation criteria	**Quality of the resulting actor object** – does the object contain sufficient information to capture the actor's characteristics and engage the actor in follow-up activities?
development status	Implemented in the IMEV2 state platform.

4.2 Teamwork support

4.2.1. Feature 11: Group presence panel

Attribute	Description
identifier	F11
name	**Group presence panel**
description	This feature provides a selected group of users with an overview of fellow group-members, their role in the group, their affiliation and online status. This feature aims to provide enhanced 'team awareness 'to group members, especially in cases where members are not co-located and the group arrangement differs over time.
fulfils function	• U2/Facilitate collaboration
supports frame	• Community Champion • Community Liaison Team
primary user community	• Professional community • Affected community • Responding community
related features	• Activity registration • Group communication channel
required information input	• Actor information such as name, affiliation • Group information, especially members and roles • A notice from a group member about his online status
information output	• A list of users that belong to a group • Per user, the online status as set forth by the user. • Per user, relevant information pertaining to his position in the group (e.g. role information, personal message, contact information)
interaction patterns and interface elements	This feature could be implemented as an **online indicator** such as present in most social applications (e.g. WhatsApp, Lync or Telegram). These applications typically use a panel that lists group members, a graphical indicator of their presence and relevant additional information, such as a personal message. Online presence can be set by either the user via an availability

	toggle, or via auto-detection (e.g. availability is set based on activity).
evaluation criteria	**Information presentation quality** and **usability**. Does the feature present the available information about the group in a suitable manner? Is the information accurate and does it add to group communication performance?

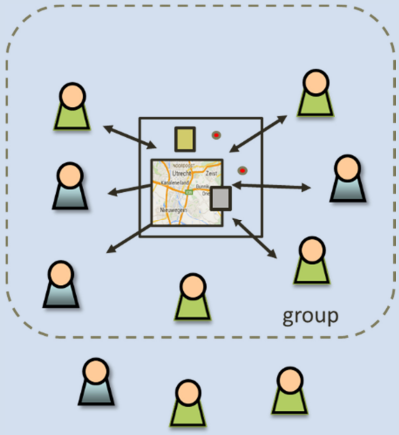
4.2.2. Feature 12: Group communication channel

Attribute	Description
identifier	F12
name	**Group communication channel**
description	This feature provides a selected group of users (i.e. a 'team') with a dedicated communication channel. The channel could be implemented in different ways, for example in the form of a text messaging service, voice communication channel or a multimedia group-chat facility. The important aspect here is that the dissemination of information does not spread beyond the team as to provide a safe and trusted information sharing environment. This feature should be available to existing groups when they come into existence. Practically, this implies that when a user joins a group, there will be a group communication option available.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community Champion • Community Liaison Team
primary user community	• Professional community • Affected community • Responding community
related features	• Group presence panel • Collaboration canvas
required information input	• Actor information such as name, affiliation • Group information such as members and roles

	• Information put forward by group members
information output	• Information put forward by group members via the communication channel
interaction patterns and interface elements	This feature could be implemented as a communication panel as present in social communication applications (e.g. WhatsApp, Lync or Telegram). These applications use a panel that provides text and media sharing functionality within a group (e.g. a chat window). Via this panel, messages are typically communicated with a timestamp and sender information. A similar setup would fit the group communication requirements of the COBACORE platform.
evaluation criteria	**Information presentation quality** and **usability**. Does the feature present the available information about the group in a suitable manner? Is the information accurate and does it add to group communication performance?
development plan	This feature has been implemented.

4.2.3. Feature 13: Collaboration canvas

Attribute	Description
identifier	F13
name	**Collaboration canvas**
description	This feature provides a group with a work-canvas on which members can collaborate in establishing proper situation awareness and determine the course of action. Group members can add text, drawings, photos, maps and other materials to the canvas and work together to build a common operational picture. This feature is particularly aimed at professional groups, but might be valuable to citizen-led activities too as to coordinate actions, align priorities and other group decision making activities.
fulfils function	• U2/Facilitate collaboration
supports frame	• Community Liaison Team
primary user	• Professional community

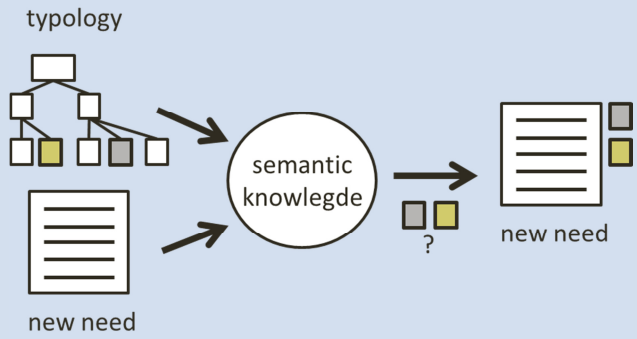
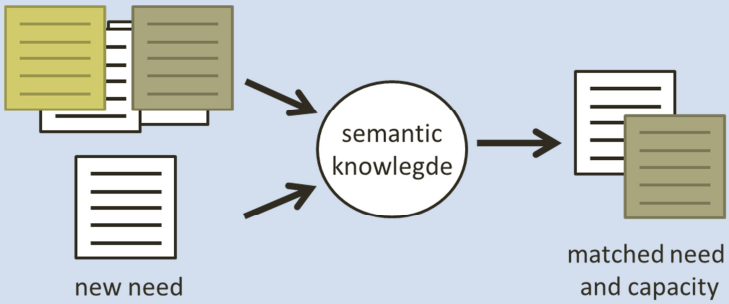
community	• Responding community (as part of a professional-led team)
related features	• Group presence panel • Group communication channel
required information input	• Information provided by group members, e.g. notes, documents, links.
information output	• Information provided by group members on the canvas
interaction patterns and interface elements	The collaboration canvas could be viewed as a group-accessible **pin board** or **whiteboard**. It serves as a joint working area where users can share relevant information. In its simplest form the canvas could be a joint text-sharing panel such as a basic wiki-page. In more advanced forms it could be more like a group work support system.  This canvas could be activated upon establishment of a group or on-demand by an authorised user such as professional user or a group or activity manager. Important design aspects are: • **structured versus unstructured information input**. When entering information onto the canvas, the user could be guided through pre-set information-structures (e.g. agreements on type and form of the information that can be put on the canvas), or be allowed to enter information in free-form (e.g. an unstructured information page). • **information management and moderation.** A canvas could function without moderation of information, but depending on the group structure and group type, it might be necessary to adopt information management and moderation strategies. In that case, only specific members of the group can modify or delete information, and can

	allow information to enter the canvas.
evaluation criteria	**Usability.** Does the canvas provide enough options for users to share relevant information, and does it do so in an accessible manner?
development status	Rudimentary implementation. The platform at IMEV2 state contains an activity information pane where activity group members can share notes.

4.3 Content matching and management

4.3.1. Feature 14: Semantic content matching

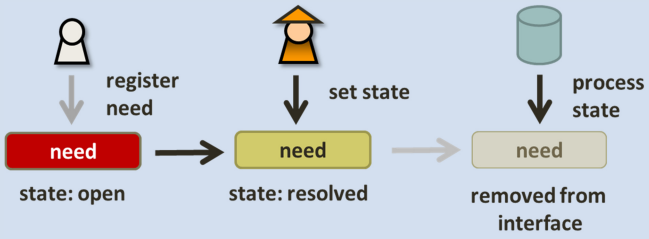
Attribute	Description
identifier	F14
name	**Semantic content matching**
description	This feature provides automatic, semantically-based matching capabilities to the platform. The platform offers various options to match needs, capacities and activities. This feature supports those options and provides suggestions for relevant matches, based on semantic relations. The user will experience this feature when registering needs, activities and capacities via system-generated suggestions that are basic in semantic matching.
fulfils function	• U3/Improve needs and capacities matching
supports frame	• Needs and Capacities Marketplace • Community Champion
primary user community	• Affected community • Responding community
related features	• F5/Needs and Capacity matching • Needs registration • Activity registration • Capacity registration
required information	• User input

input	• Registered content (needs, capacities, activities)
information output	• Suggestion for relevant matches (e.g. a set of capacities that would fit the requested need; a set of activities that would fit a offered capacity)
interaction patterns and interface elements	This involves a system feature. The visible output of this feature is sparse. There are two distinct instances when the user can experience this feature, both during the registration of content. When registering needs, capacities or activities, the system may suggest a suitable typing based on analysis of entered descriptions. Using its typology, the system may use semantic knowledge to infer a proper type.  Secondly, the semantic framework can provide matching suggestions based on semantic matching between the new content and registered content. 
evaluation criteria	**Operational value.** Semantic matching potentially provides a major speed-up of the registration and matching process, but only when it provides relevant and trustworthy results. If users distrust the process based on faulty suggestions, then that may undermine the registration process. Rigorous testing and evaluation needs to uncover whether the semantic matching system is robust enough to provide operational value.

development status	Implementation pending. This feature will be available for the final evaluation session.

4.3.2. Feature 15: Content lifecycle management

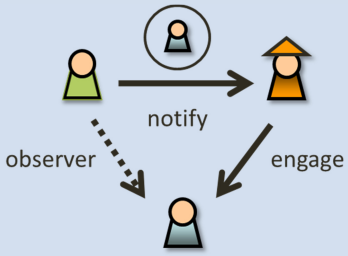
Attribute	Description
identifier	F15
name	**Content lifecycle management**
description	This feature provides options for authorised users to adjust the state of needs, capacities and activities and thus govern the lifecycle of registered content. Registered content can have different states, depending on their status (e.g. matched, resolved, cancelled, and so forth). The transition through different states constitutes the lifecycle of registered content, and is partly regulated In the platform through automated rules. See chapter 3 for an in-depth description of the lifecycle of platform content. This feature prescribes options for platform users to actively manage the state of content, and intervene in the core lifecycle process.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching
supports frame	• Needs and Capacities Marketplace • Community Champion • Community Liaison Team • Intelligence and Insight
primary user community	• Professional community • Affected community • Responding community • Platform managers
related features	• Need and capacity matching • Needs registration • Activity registration • Capacity registration
required information	• The state of registered content

input	
information output	• Updated states of registered content
interaction patterns and interface elements	This feature could be implemented as a simple state modification option alongside views of registered needs, capacities and activities, such as the activity list as is present in the user interface. Authorised users would have an option to change the state of content, for instance from `open` to `resolved`. The system would process the new state, and consequently remove the need from the user interface.  Most likely, the platform itself will contain rules for state changes. Chapter 2# details the various processing rules. Practically, this means that there are different actors that can exert state changes: the platform, authorised professionals and platform managers (system administrators that are not participating in an operational role).
evaluation criteria	**Usability** is an important evaluation aspect for this feature. With potentially hundreds of active needs, capacities and activities, platform managers and professionals may quickly lose track of the state of entries. Additionally, users may need to revert on their state changes decisions when new information arrives (e.g. undo a state change). Thirdly, there are different actors that can influence the state of content. Therefore, the implementation needs to be transparent and user-centred in order to prevent loss of awareness by users or unintended modifications. Additionally, with different actors exerting state changes, the system needs to be **robust** and prevent concurrency issues.
development status	Rudimentary manual lifecycle management available. See 3.6.3 for the final vision on lifecycle management of needs, capacities and activities

4.4 User feedback and reputation management

4.4.1. Feature 16: User reporting

Attribute	Description
identifier	F16
name	**User reporting**
description	This feature allows users to alert platform managers to fellow users. Users could use this feature in case of observations of abusive behaviour, or other circumstances that need direct user reporting. Examples of such circumstances would be commercial spamming, criminal activities, harassment, vandalism or similar undesirable behaviours. Platform managers can subsequently take action and engage with the reported user.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching
supports frame	• Community Liaison Team
primary user community	• Professional community • Affected community • Responding community • Platform managers
related features	• User rating • User management
required information input	• Details of the offending actor • Active platform manager
information output	• An alert to the active platform manager

interaction patterns and interface elements	This feature provides platform users with an option to signal the active platform manager on abusive fellow platform users. This feature could take the shape of an **alert button** (‘*report this user*’) in  every interface where users are listed (e.g. in marks on the map, in activity listings, in overview panels). Pressing this button would send a notice to the active platform manager, who can then take action via his own communication means. Additionally, the original observer could be informed about the result of his report.
evaluation criteria	**Practical consequences.** This reporting option is a powerful asset in preventing abuse of the system. However, it can also lead to deliberate false reporting and moderation overload on the side of the platform management. Evaluation sessions with a sufficiently large audience need to show the practical consequences.
development status	Not implemented yet.

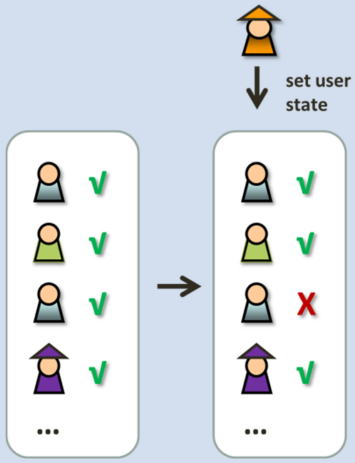
4.4.2. Feature 17: User rating

Attribute	Description
identifier	F17
name	**User rating**
description	This feature gives users the option to convey appreciation or disapproval of other users.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching

supports frame	• Needs and Capacities Marketplace • Community Champion • Community Liaison Team
primary user community	• Professional community • Affected community • Responding community • Platform managers
related features	• User reporting • User management
required information input	• A feedback report from the reporting user
information output	The output of this feature is an **increase or decrease of a user rating,** similar to the feedback score on Ebay. This rating may be stored in the associated actor profile, and can be shown as part of the actor's visible profile. Optionally, this feature could include an option to leave a **feedback message** that is stored alongside the score. The user's feedback score is made available in every instance where the user is mentioned (e.g. need and capacity offerings, or activities).
interaction patterns and interface elements	Users are given an option to rate fellow users. This could be instantiated via an **appreciation selector** such as a simple 'like or dislike' button. Optionally, this feature could be implemented in a more elaborate form such as a **feedback report.** Such a report could include a more detailed account including background information about the interaction between the reporting user and the receiving user, and explanatory notes.
evaluation criteria	**Practical consequences and operational value.** This reporting option is a powerful asset in helping users find good peers and to prevent abuse of the system. However, it can also lead to deliberate false reporting, especially when there are no options to retort. Evaluation sessions with a sufficiently large audience need to show the practical consequences.
development status	Not available at the IMEV2 platform status.

4.4.3. Feature 18: User management

Attribute	Description
identifier	F18
name	**User management**
description	This feature provides platform managers with option to modify user profiles and access rights. By providing these options, platform managers can facilitate platform users or intervene when necessary. For instance, this feature would allow a platform manager to revoke the access rights to the system in case of system abuse.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching
supports frame	• Community Liaison Team
primary user community	• Professional community • Platform managers
related features	• User rating • User reporting
required information input	• Actor profiles
information output	Depending on the type of action taken, this feature could result in an adjusted actor object (e.g. changed attributes). Subsequently, this might lead to a user being barred from entering the platform.
interaction patterns and interface elements	This feature requires a dedicated **user management panel.** In this panel, the platform manager can view the profiles of users and perform modifications or deletions.

	 The user management panel needs to provide access to all relevant information about users (contact details, activity history, associated content), and provide a clear overview of modifiable attributes, such as access rights. Additionally, to ensure transparency and accountability, there need to be options to inform the affected users of changes in their profile with facilities to respond to the platform administration.
evaluation criteria	**Practical consequences and operational value.** This feature is a potentially asset to professional users to prevent abuse of the system and uphold overall quality. However, it can also lead to false reporting or acting on incomplete information. Evaluation sessions with a sufficiently large audience need to show the practical consequences.
development status	Not available as a front-side-feature.

4.5 User mobilisation

4.5.1. Feature 19: Activity publishing tools

Attribute	Description
identifier	F19
name	**Activity publishing tools**
description	This feature provides an activity manager with options to promote the activity beyond the regular listing. After registration an activity

	is mapped onto the main map space based on its area of deployment. Furthermore, the activity becomes listed in overviews and can be offered as suggestion in matching or search efforts. This feature provides for additional publishing options that can be used to attract interest from platform users, such as advertisements on users` information stream or messages on external sources.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community champion • Community Liaison Team
primary user community	• Responding community • Professional community
related features	• Activity registration
required information input	• Available external information channels
information output	• Publishing messages to selected groups of users • Publishing messages to chosen external channels
interaction patterns and interface elements	This feature provides an activity manager with a set of options to publish the activity to a wider audience. This set of options could include pushing notifications to users of a certain type or in a certain location, publishing the activity to the general platform information stream or using external channels to publish the activity. The activity manager could be presented with a **dashboard of publishing options**, or guided through a **wizard-style selection process**. For each step, the activity manager is presented with a preferred message to send, a preferred set of recipients and other relevant information.
evaluation criteria	**Operational value.** Do the provided tools offer options to publish an activity in an effective manner? Additionally, does this feature elicit abusive behaviour or contribute to information overload? **Usability.** Are the provided options understandable enough for activity manager, and are they easy to use?
development status	Not available at the IMEV2 level platform

4.6 Information provision

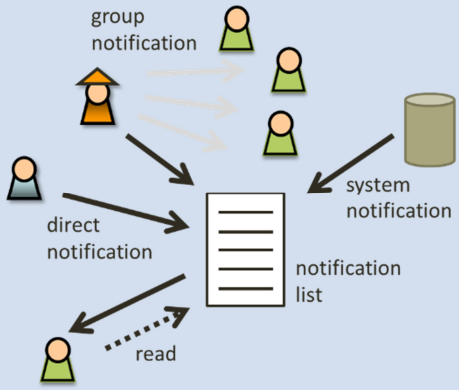
4.6.1. Feature 20: Information stream

Attribute	Description
Identifier	F20
Name	**Information stream**
Description	This feature allows the presentation of a stream of user-relevant information from a variety of sources. These sources could be external (public news sources such as Twitter or Facebook), or internal (information provided by professionals).
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community Champion • Community Liaison Team • Intelligence and Insight
primary user community	• Professional community • Affected community • Responding community • Platform managers
related features	• Notification panels
required information input	• Information sources available to the system • Search profiles
information output	• A running list of news items offered from selected sources
interaction patterns and interface elements	This feature provides a running stream of information on the main interface of the platform. The feature can be implemented as a dynamic, scrolling information panel.
evaluation criteria	**Information intensity and operational value.** An information stream that is populated through various sources can quickly lead to an information overload to the recipient or a loss of operational value. Evaluation sessions need to make clear how the information

	stream should be managed or filtered as to provide relevant information.
development status	This feature is available at the IMEV2 level platform and will be expanded upon with future platform releases.

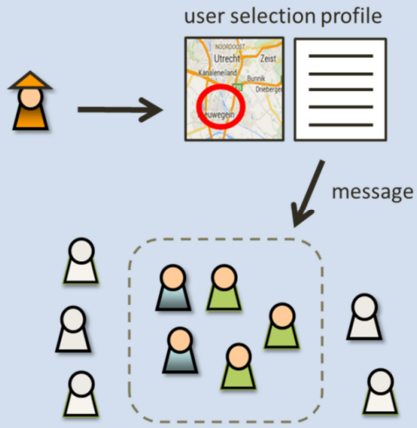
4.6.2. Feature 21: Notifications panel

Attribute	Description
identifier	F21
name	**Notifications panel**
description	This feature provides a panel that presents the user with relevant notifications. These notifications may stem from a variety of sources, and may be sent directly, or because the user is associated with a specific need, capacity, activity, group or environment.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community Champion • Community Liaison Team • Intelligence and Insight
primary user community	• Professional community • Affected community • Responding community • Platform managers
related features	• Information stream
required information input	• Incoming notifications from platform users
information output	• A running list of incoming notifications
interaction patterns and interface elements	This feature involves an information space in the users` interface where notifications from other parties and system management are placed,

	 Notifications are placed in a list form with optionally labels and icons for different types of notifications (peer-user contact requests, system generated messages, group notifications).
evaluation criteria	**Information intensity and operational value.** A notification panel that is populated with various types of notifications can quickly lead to an information overload to the recipient or a loss of operational value if it does not capture the user's attention properly. Evaluation sessions need to make clear how the notification panel should be configured and managed as to provide relevant information with the proper prioritisation.
development status	Implemented in the IMEV2 state platform.

4.6.3. Feature 22: Targeted communication

Attribute	Description
identifier	F22
name	**Targeted communication**
description	This feature offers professional users with the option to send messages to a group of users based on certain target parameters, such as location, need or capacities offered or actor type. This feature gives professional users the means to quickly contact a specific group of users.
fulfils function	• U1/Enhance information exchange

	• U2/Facilitate collaboration
supports frame	• Community Liaison Team • Intelligence and Insight
primary user community	• Professional community
related features	• Notifications panel
required information input	• None.
information output	• A message to a selected group of users
interaction patterns and interface elements	This feature works from the definition of a desired profile: a pattern that describes which users should be reached. The user could be taken through a **wizard-style** process that asks the users to specify the target audience based on questions of user-type, location, relevant characteristics (needing help, providing capacities, etc.), and so forth. At the end, the system has a communication profile that it can use to send the message to the selected group of users. 
evaluation criteria	**Usability**. Does the feature provide the user with the right functionalities to identify the intended group of recipients? And what is the **operational value** for this kind of group communication? Are there enough use cases to make this feature worthwhile pursuing?
development status	Not implemented at the IMEV2 level platform.

4.7 Analytics and sense making

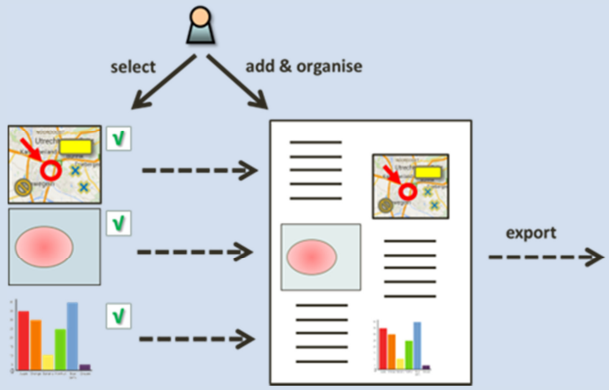
4.7.1. Feature 23: Annotation of maps

Attribute	Description
identifier	F23
name	**Annotation of maps**
description	The COBACORE platform provides data in various forms among which maps. Maps are important tools in creating shared situation awareness among professionals, and in operational environments, they form the backdrop for planning and coordination activities. This feature provides professional users with the option to annotate maps in the COBACORE platform with notes, graphics and cues.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community Liaison Team • Intelligence and Insight • Learning Environment
primary user community	• Professional community
related features	• Collaboration canvas
required information input	• None.
information output	• A map layer that contains notes, graphics and other annotation items added by the user.
interaction patterns and interface elements	This feature provides authorised professional users with a toolset to annotate maps with graphics. The options could take on the form of a traditional drawing application with icons for various tasks such as drawing polygons, lines, adding text boxes and inserting external media.

	 The annotations are stored in a separate layer that can be presented to peer-users, and kept hidden from others. This feature could materialise part of the **collaboration canvas** feature,
evaluation criteria	**Usability and flexibility.** Creating data views is a difficult process, and care should be taken that this features are implemented in a simple manner. If the process is too complex and involves too many steps, then the **operational value** will be low. Also, the **validity** of output needs to be assessed. If the feature constructs views that do not properly convey a proper view of the situation, then this feature could lead to biases decisions. Evaluation sessions need to assess potential pitfalls.
development status	Not available at the IMEV2 level platform

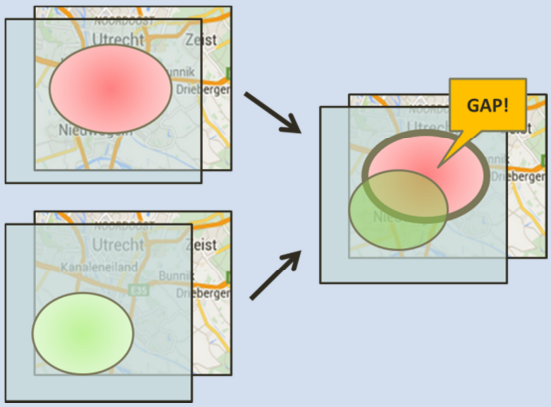
4.7.2. Feature 24: Status report export

Attribute	Description
identifier	F24
name	**Status report export**
description	Through the various data visualisation options, professional users can build up materials that underpin a need assessment. Through this feature, the user can build up a needs assessment report. This report consists of user selected visuals and added informative notes. The report is meant to capture the key status as perceived by the professional community.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community Liaison Team

	• Intelligence and Insight • Learning Environment
primary user community	• Professional community
related features	• Recovery analysis view • Gap analysis support • Annotation of maps
required information input	• A user-selected collection of graphics, notes, diagrams and graphs
information output	• A formatted document containing selected captures of dynamic data graphics and text segments
interaction patterns and interface elements	This feature gives the user an option to format a report, using selected captures of data and diagrams, and complemented with text. This report can then be exported as document e.g. in pdf form).  Selection of diagrams could be done via simple checkboxes (‘add to report’), or similar selection options.
evaluation criteria	**Usability.** Compiling a report should be a simple operation, not a full layout and design process. The feature should be implemented in such a way that the user can quickly select the data-views that need to be included, and that text can be easily added.
development status	Not implemented.

4.7.3. Feature 25: Gap analysis support

Attribute	Description
identifier	F25
name	**Gap analysis support**
description	This feature provides simple map-based gap analysis support. In its simplest interpretation, gap analysis aims to uncover whether needs have been properly met, and where gaps exist between what has been asked and what has been provided. The COBACORE platform provides various ways to list and visualize needs and capacities. This feature adds a new analytical capability: visualization of gaps. This feature takes the registered needs and capacities in a certain area, and provides means to visualize and analyse the gaps.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community Liaison Team • Intelligence and Insight • Learning Environment
primary user community	• Professional community
related features	• Features that manage content mapping.
required information input	• Registered needs, capacities and activities • User analysis preferences
information output	• A layer in the map space that depicts ‘gaps’
interaction patterns and interface elements	This feature could be implemented as an analysis-instrument to identify the disparity between data-sets (e.g. ‘needs’ vs. ‘capacities’). The outcome of this disparity could be said to point to a gap, and visualised on screen as the difference between two maps layers, for instance using heat map projections. The gaps would be where there are needs that have not been met.

	
evaluation criteria	**Operational value.** Gap analysis is a critical part of needs assessment. There is a danger for biased views if the gap determination is based on overlying simplistic data-representations. Elicitation sessions and evaluation sessions need to make clear what a gap exactly is, and how it should be determined.
development status	Not implemented in the IMEV2 level platform.

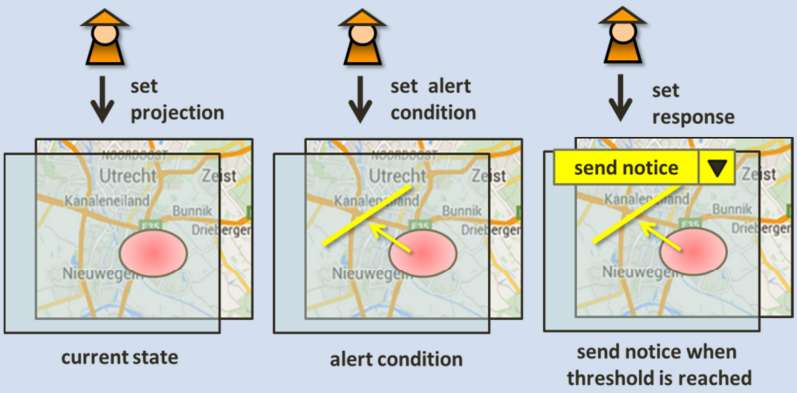
4.7.4. Feature 26: Recovery analysis view

Attribute	Description
identifier	F26
name	**Recovery analysis view**
description	Through this feature, the user can obtain various views on the recovery process that can help in progress assessment. The feature uses the information present in the platform and offers options to display data in different formats, such as line-, pie or bar-charts as to obtain insight in trends and events.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community Liaison Team • Intelligence and Insight • Learning Environment

primary user community	• Professional community
related features	• All features related to acquisition, management and representation of content. • Status report export
required information input	• Registered content (needs, capacities, activities) • Active typology (types, categories) • Baseline information (maps, key objects in the area)
information output	• User-defined graphs
interaction patterns and interface elements	The COBACORE platform contains a wealth of information, including baseline information (static information about the environment), dynamic information (current situation information) and registered content (needs, capacities, activities). This feature should give the user options to build meaningful views that can help in the assessment of the recovery process. The feature could use a **wizard-style** process to let the user define data-types, ranges and view preferences.
evaluation criteria	**Usability and flexibility.** Creating data views is a difficult process, and care should be taken that this feature is implemented in a simple manner. If the process is too complex and involves too many steps, then the **operational value** will be low. Also, the **validity** of output needs to be assessed. If the feature constructs views that do not properly convey a proper view of the situation, then this feature could lead to biases decisions. Evaluation sessions need to assess potential pitfalls.
development status	Rudimentary implementation in the IMEV2 status platform with non-modifiable graphs. A user-configurable version is planned for a next iteration.

4.7.5. Feature 27: Situation alerts

Attribute	Description
identifier	F27
name	**Situation alerts**

description	This feature provides professional users with an option to track the evolution of certain data, such as the number of needs met or the number of activities initiated, and setup warnings when thresholds have been reached.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community Liaison Team • Intelligence and Insight • Learning Environment
primary user community	• Professional community
related features	• All features related to acquisition, management and representation of content.
required information input	• Registered content (needs, capacities, activities) • Active typology (types, categories) • Baseline information (maps, key objects in the area)
information output	• An alert in the user interface
interaction patterns and interface elements	This feature consists of three steps: 1) defining the data-view, 2) specifying the alert condition and 3) defining the alert response. The first step, defining the data-view should lead to a data-view that is suitable for the purpose of this feature. This could be a map projection, a graph-projection, a list or any other projection that is suitable to define alert threshold.  The second step concerns setting the alert condition. This could be a

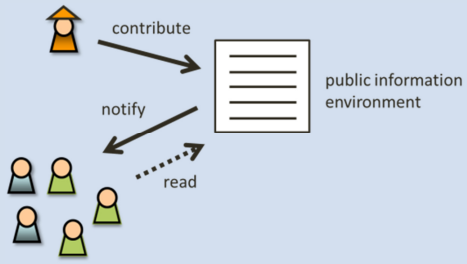
	simple value-setting, or something more elaborate or graphic-based. For example, if the alerting has a geographical basis ('alert me when there are new needs of this type in that area'), then conditions could be set in a graphical way (e.g. draw a line). The third step concerns the alert response once the condition has been met. The user could be presented with various options, such as a high-priority notice in the notification panel, or an overlaid alert.
evaluation criteria	**Usability and flexibility.** Creating alerts is a difficult process, and care should be taken that this feature is implemented in a simple manner. If the process is too complex and involves too many steps, then the **operational value** will be low. Also, the **validity** of output needs to be assessed. If the feature constructs alerts that do not properly convey a proper state of the situation, then this feature could lead to biases decisions. Evaluation sessions need to assess potential pitfalls.
development status	Not implemented.

4.8 Instructions and guidance

These features are aimed to support instruction and guidance activities by professional community to the responding community.

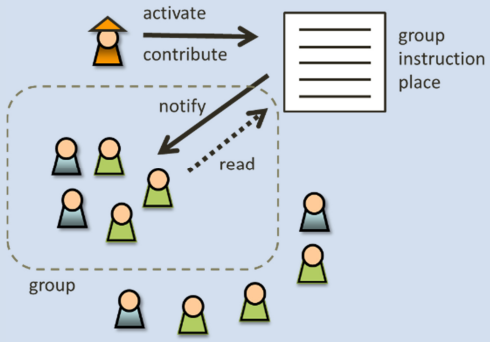
4.8.1. Feature 28: Public information environment

Attribute	Description
identifier	F28
name	**Public information environment**
description	The 'public instruction portal (PIE)' is a publically accessible information and documentation environment, aimed at instructing and guiding non-professional users. Authorised professional users can add information and documentation on relevant recovery topics. This feature needs to be distinguished from the feature 'information stream'. The information stream contains information from a variety of sources. The PIE is a dedicated instruction and information environment, populated and curated by a select group of professionals.

fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community Liaison Team • Intelligence and Insight • Learning Environment
primary user community	• Professional community
related features	• Dedicated instruction environment
required information input	• Public content, provided by trusted parties
information output	• A portal containing the public information
interaction patterns and interface elements	This feature provides an **information space** for publically accessible information; information that is of relevance given the circumstances and has an instructional or informational value.  Authorised users would provide content for the PIE, such as documents, video-links and other media. Changes to the PIE would give rise to alerts and incite other platform users to visit the PIE.
evaluation criteria	**Information intensity and operational value.** A public information environment that is not properly populated and managed can quickly lead to an information overload to the recipients or a loss of operational value. Evaluation sessions need to make clear how a PIE should be should be setup or managed as to provide relevant information.
development status	Not implemented yet. There are some plans for rudimentary implementation for the final platform release.

4.8.2. Feature 29: Group instruction place

Attribute	Description
identifier	F29
name	**Group instruction place**
description	The 'group instruction place' (GRIP) is an information sharing area in which group members can receive instructions and support from professional parties. It provides options for professionals to share relevant documents and insights to a selected group of recipients.
fulfils function	• U1/Enhance information exchange • U2/Facilitate collaboration
supports frame	• Community Liaison Team • Intelligence and Insight • Learning Environment
primary user community	• Professional community • Responding community
related features	• Public information environment • Group communication channel • Collaborative canvas
required information input	• Active activities and groups • Instructional content
information output	• A group-accessible information area
interaction patterns and interface elements	A GRIP is to be activated by a professional party as a means to provide professional support. Furthermore, a GRIP is always tied to an activity or group. Therefore an option to start a GRIP would most likely be available to professional users as part of their group or activity management facilities. In this way, a professional can quickly setup a dedicated GRIP when appropriate with links to all group members. When a GRIP has been activated by a professional, all related group or activity members receive a notification and an option to visit the GRIP.

	 The associated professional can contribute materials to the GRIP such as instructional videos or guideline documents. In a more elaborate setup, the GRIP could also contain group work facilities such as shared whiteboard or live video-feeds. In that perspective, this feature could align with the **group communication channel** feature and the **collaboration canvas feature.**
evaluation criteria	**Information intensity and operational value.** A group instruction environment that is not properly populated and managed can quickly lead to an information overload to the recipients or a loss of operational value. Evaluation sessions need to make clear how a PIE should be should be setup or managed as to provide relevant information.
development status	Not implemented.

4.9 Training features

4.9.1. Feature 30: Platform feature configuration

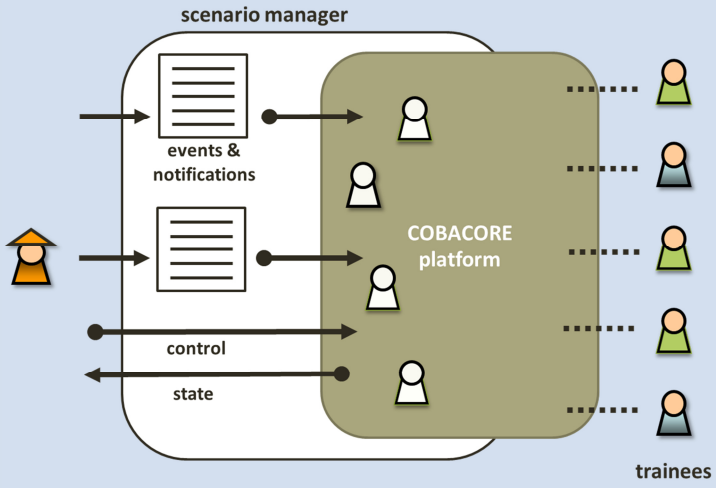
Attribute	Description
identifier	F30
name	**Platform feature configuration**
description	This feature allows platform managers to configure the platform by activating or disabling specific elements of the platform, such as information streams, communication facilities and analytical tools. Platform managers use this feature to provide trainees with a platform experience that fits their training or evaluation objectives.

fulfils function	• This feature fulfils the following functions in a training setting: • U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching
supports frame	• Learning Environment
primary user community	• Trainers • Platform managers • Professional community (in training) • Responding community (in training) • Affected community (in training)
related features	• Scenario manager • Performance analysis
required information input	• A system configuration profile
information output	• An updated configuration profile
interaction patterns and interface elements	This system could be implemented using a system configuration application, which should not be accessible for regular platform users. The application should present the user with a list of available system elements and options to change, remove or add elements. After modification of the typology, the user should be made aware of the consequences and have an option to revert modifications.
Evaluation criteria	**Usability for non-developers.** Modifying the platform configuration is typically a back-end process where system developers build the necessary coding structures to integrate resources into a platform. When seen as a feature for platform managers, it needs to be usable for non-technical users and thus be evaluated on usability and comprehensibility for the target user group.
Development status	Not available as a front-side feature to platform users.

4.9.2. Feature 31: Scenario manager

Attribute	Description

identifier	F31
name	**Scenario manager**
description	The scenario manager governs the events and inputs that are being presented to the trainees during the training, such as incoming information, registered content, baseline information and so on. Depending on the type of training, events and inputs can differ in nature, and can be time-planned. For example, the scenario manager can introduce new needs at specific times in the scenario in response to new events (e.g. new requests for water-proof boots in a certain area after a new flooding). Also, the scenario manager could disrupt communication between certain users as a response to scenario events (e.g. disrupt communication between actors in a certain area after a (fictive) power-failure).
fulfils function	• This feature fulfils the following functions in a training setting: • U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching
supports frame	• Learning Environment
primary user community	• Trainers • Platform managers • Professional community (in training) • Responding community (in training) • Affected community (in training)
related features	• Platform feature configuration • Performance analysis
required information input	• User input
information output	• A scenario manager configuration
interaction patterns and interface elements	The scenario manager feature provides options for platform managers to programmatically enter content and events into the system and to alter various aspect of the platform. By doing so, the scenario manager can materialise a training scenario. The process has three major steps: 1) compile content that needs to be registered in the system, 2) set system events and general

	notifications, and 3) manage the scenario. In the first step, the user needs to prepare the content that needs to be entered into the system. Each entry (need, capacity, activity) needs a simulated actor object and time- or condition statement when the scenario manager can enter the entry. In this way, the scenario manager creates simulated actors. In the second step, the user needs to do a similar action regarding system events and general notifications. A system event could involve the disruption of communication between certain groups of actors (based on type or location), or the insertion of goal-changing events via general notifications. In a third step, the user obtains control of the scenario and can start and stop the training at will and intervene in the execution when necessary.  Scenario management could be implemented as a separate application that has control over the inputs and behaviour of the system.
evaluation criteria	**Usability and control.** Does the proposed setup give the training management enough options to control the platform in a suitable way for training purposes?
development status	Not implemented as a front-side feature.

4.9.3. Feature 32: Performance analysis support

Attribute	Description
identifier	F32

name	**Performance analysis support**
description	This feature provides performance analysis functions. Through this feature, trainers that use the COBACORE platform during training can better assess the performance of trainees. The feature provides an overview of user activity during the training and key performance indicators.
fulfils function	• This feature fulfils the following functions in a training setting: • U1/Enhance information exchange • U2/Facilitate collaboration • U3/Improve needs and capacities matching
supports frame	• Learning Environment
primary user community	• Trainers • Platform managers • Professional community (in training) • Responding community (in training) • Affected community (in training)
related features	• Platform feature configuration • Scenario manager
required information input	• User activity • Platform state
information output	• A report per user of their activity throughout the training
interaction patterns and interface elements	The features should include a **configuration process, a monitoring process and an output process**. The trainer should be able to state his monitoring preferences in the first process. This step could be implemented as a simple selection form. In the monitoring process, the feature monitors a) the activity of all platform users during training, and b) the events as generated by the scenario manager. This step has explicit user interaction. At the output process, the feature generates a report that correlates user behaviour with the platform state (e.g. what actions did the user take, and what was the status of the environment). The resulting report could have the form of a document, or interactive after-action replays the users' behaviour. This feature could be implemented as part of the **scenario manager**, and turn it into an integrated training control system.

evaluation criteria	**Validity.** Does this feature produce results that are meaningful to assess a user's performance? Does the report accurately capture the state of the environment, and does it provide enough insights for the trainer so establish a proper image of the user's thought processes?
development status	Activity logging is available as a back-end feature, but not as a user-accessible function.

4.10 Platform configuration and administration

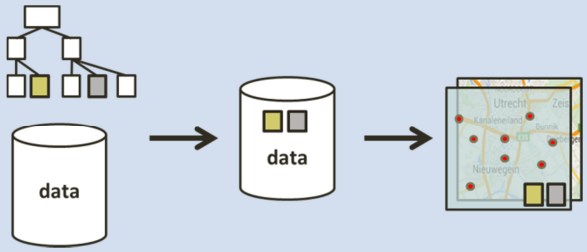
4.10.1. Feature 33: Typology configuration

Attribute	Description
identifier	F33
name	**Typology configuration**
description	This feature allows platform managers to alter the typologies that are used in the platform, such as the categories and types of need, capacities and activities. To cater to local preferences or adhere to interoperability standards, the typology that is used in the platform needs to be configurable. Platform managers need to be able to change core type-labels (e.g. recovery domains) and add or remove elements (attributes and descriptions). To the user, a change in the typology would result in different typing options in the user interface.
fulfils function	• Generic system performance
supports frame	• Needs and Capacities Marketplace • Community Champion • Community Liaison Team • Intelligence and Insight • Learning Environment
primary user community	• Platform managers
related features	• Dataset typing

	• Information source management
required information input	• The current typology structure (need, capacity and activity types, actor types, etc.)
information output	• An updated typology structure (need, capacity and activity types, actor types, etc.)
interaction patterns and interface elements	This system could be implemented using a system configuration application, which should not be accessible for regular platform users. The typology could be presented as a list structure with clear options to change, remove or add elements. After modification of the typology, the user should be made aware of the consequences and have an option to revert modifications.
evaluation criteria	**Operational value.** Changing the typology significantly impacts the user experience. Evaluation efforts need to uncover whether dynamic adaptation of the typology is necessary, or that effort should be invested to develop a fixed typology beforehand. **System stability.** Changing the typology might impact already registered content. Evaluation testing should uncover whether the system can successfully handle typology changes.
development status	Not implemented as a user feature at the IMEV2 platform state. Typing can be changed in back-end-code.

4.10.2. Feature 34: Dataset typing

Attribute	Description
Identifier	F34
Name	**Dataset typing**
Description	This feature gives platform managers the option to type available datasets using elements from the platform typology. In such an activity, the dataset would be annotated with selected types. Typing makes it possible for users to identify datasets that are relevant to their objectives based on a harmonized typology. For instance, a platform manager could opt to type a dataset of medical facility locations with a suitable label from the platform typology, such as 'Healthcare'. This would make it possible to establish logical links between elements from this datasets (e.g. medical facilities) and registered needs, capacities or activities with the same label (e.g.

	people that have registered a need with a 'Healthcare' specification). In practice, this would provide support for smart need and capacity matching, and linking resources to activities.
fulfils function	• Generic system performance
supports frame	• Needs and Capacities Marketplace • Community Champion • Community Liaison Team • Intelligence and Insight • Learning Environment
primary user community	• Platform managers
related features	• Typology configuration • Information source management
required information input	• The available datasets that are used in the platform to populate mapping layers. • The current typology
information output	• A dataset with associated typing
interaction patterns and interface elements	This feature is a system configuration feature, and should not be performed by regular users. Upon attaching a new information source to the system, the platform manager would be presented with an option to label the dataset. The labels come from the current typology, and one or more labels can be associated with the dataset at hand. This information is stored, and used in the presentation of information to platform users. 
evaluation criteria	**Operational value.** What is the operational value of such a dataset typing feature? Does it help to find relevant information earlier, or does it increase biased behaviour?

	Usability. How easy is it for a platform manager to associate a type with a dataset?
development status	Not implemented.

4.10.3. Feature 35: Information source management

Attribute	Description
Identifier	F35
Name	**Information source management**
Description	This feature allows platform managers to alter the information sources that feed into the platform, such as news feeds, environmental datasets, social media channels and custom information streams.
fulfils function	• Generic system performance
supports frame	• Needs and Capacities Marketplace • Community Champion • Community Liaison Team • Intelligence and Insight • Learning Environment
primary user community	• Platform managers
related features	• Typology configuration • Dataset typing
required information input	• New information sources (datasets, online feeds, channels)
information output	• Information sources available to users via platform features.
interaction patterns and interface elements	This feature enables platform managers to add new information sources to the platform, or remove existing sources. This feature is a system configuration feature, and should not be performed by regular users, and should not be confused with hiding information elements in the user interface. For this feature, platform managers would access a separate configuration application and modify the information sources available in the platform through a number of identification and integration steps. If all necessary technical requirements have been

	met, this could be visualised as a wizard-style process where the platform manager is taken through all the necessary selection and configuration inquiries to properly integrate the information source into the platform.
evaluation criteria	**Usability for non-developers.** Adding information sources is typically a back-end process where system developers build the necessary coding structures to integrate resources into a platform. When seen as a feature for platform managers, it needs to be usable for a non-developer.
development status	Not available as a front-side feature at IMEV2 platform state.

4.11 Features per target user group

The previous section describes the proposed features in great details. The following table shows a comparison of core platform features and views per target user group:

Identifier	Feature name	Category	Professional community	Affected community	Responding community	Platform managers	Trainers
F1	Actor registration	Core set	•	•	•		
F2	Need registration	Core set		•			
F3	Capacity registration	Core set			•		
F4	Needs and capacities overview	Core set	•	•	•		
F5	Needs and capacities matching	Core set		•	•		
F6	Actors and activities overview	Core set	•	•	•		
F7	Baseline situation overview	Core set	•				

F8	Basic recovery views	Core set	•				
F9	Basic information exchange	Core set	•	•	•		
F10	Activity registration	Core set	•		•		
F11	Group presence panel	Teamwork support	•		•		
F12	Group communication channel	Teamwork support	•		•		
F13	Collaboration canvas	Teamwork support	•		•		
F14	Semantic content matching	Content matching and management	•	•	•		
F15	Content lifecycle management	Content matching and management	•	•	•	•	
F16	User reporting	User feedback and reputation management	•	•	•	•	
F17	User rating	User feedback and reputation management	•	•	•	•	
F18	User management	User feedback and reputation management	•			•	
F19	Activity publishing tools	User mobilisation		•	•		
F20	Information stream	Information provision	•	•	•		
F21	Notifications panel	Information provision	•	•	•		
F22	Targeted communication	Information provision	•				
F23	Annotation of maps	Analytics and sensemaking	•				

F24	Status report export	Analytics and sensemaking	●				
F25	Gap analysis support	Analytics and sensemaking	●				
F26	Recovery analysis view	Analytics and sensemaking	●				
F27	Situation alerts	Analytics and sensemaking	●				
F28	Public information environment	Instructions and guidance	●	●	●		
F29	Group instruction place	Instructions and guidance	●	●	●		
F30	Platform feature configuration	Training features				●	●
F31	Scenario manager	Training features				●	●
F32	Performance analysis support	Training features				●	●
F33	Typology configuration	Platform configuration and administration				●	
F34	Dataset typing	Platform configuration and administration				●	
F35	Information source management	Platform configuration and administration				●	

Table 10: Comparison of core features and views per target user group

5 Outlook

The D3.2 report represents the current thinking about the features and core concepts of the COBACORE platform. This document is meant to guide WP2 and WP4 development towards the final state of the platform. It is fuelled by knowledge gained in evaluation sessions, which gave us insights on what stakeholders wanted to see implemented in the COBACORE platform. The features and structural mechanisms proposed in this report should help WP2 and WP4 build a platform that is conducive to stakeholders' needs and expectations and serve as a point of inspiration for discussions on further development and deployment.

This report has two major parts: the Core Concepts (chapter 3) and COBACORE feature descriptions (chapter 4). The Core Concepts section describes key structural decisions that underlie the COBACORE platform and details various functional mechanisms that drive the platform.

The COBACORE features section progresses from the work described in D3.1 and introduces 25 new features. These features result from observations of evaluation sessions and statements made by stakeholders. The proposed features provide guidance to WP2 and WP4 in the development of the final platform, and many have already found their way into intermediary versions of the platform.

In conclusion:

- The WP2 team will use the proposed features and the surrounding structures to finalise their data model and shape their semantic matching algorithms.
- The WP4 team will use the proposed features and concept structures to further refine the functionalities included in the final platform.
- The WP5 team will use the proposed features to establish valuable performance criteria for the final evaluation, and construct a practical assessment framework.