QROWD - Because Big Data Integration is Humanly Possible

Innovation Action

Grant agreement no.: 732194

D3.1 – Participatory framework

Due Date	30 November 2017
Actual Delivery Date	15 December 2017
Document Author/s	Eddy Maddalena, Luis-Daniel Ibáñez, Tom Blount, Elena Simperl -- University of Southampton
Version	1.0
Dissemination level	PU
Status	Final
Document approved by	Nines Sanguino -- ATOS

Project co-funded under Horizon 2020 Research and Innovation EU programme

TABLE OF CONTENT

Page

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial intelligence
BDVC	Big Data Value Chain
CI	Collective Intelligence
CS	Crowdsourcing
GWAPs	Games with a purpose
HIT	Human Intelligence Task

ABSTRACT

This deliverable introduces the QROWD participatory framework, which aims to integrate crowdsourcing for the support of the project use cases described in D1.1 and D2.2. Crowdsourcing is a type of participative online activity in which an individual or organization proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the undertaking of a task, either voluntary or in exchange of a reward. Crowdsourcing is commonly used for complementing automatic computation.

The inclusion of human participation in the project benefits the first three phases of the Big Data Value Chain (BDVC): data generation and acquisition, data integration, and data analysis. In order to describe and compare the crowdsourcing solutions proposed for the use cases defined in D1.1 and D2.2, and inform the implementation of the crowdsourcing middleware prototype D3.2, we developed a participatory framework by combining the frameworks of Malone et al. **(Malone, 2010)**, Smart et al. **(Smart, 2014)**, and Quinn and Bederson **(Quinn, 2011)** already present in literature and then extending it with geospatial and temporal dimensions required for our mobility-related scenarios.

In this deliverable we first (1) introduce crowdsourcing definitions and fundamental concepts, then (2) we discuss QROWD participatory framework, focusing on novel elements, subsequently (3) we present some crowdsourcing based solutions to the given use cases, and finally (4) we describe the crowdsourcing middleware component architecture.

EXECUTIVE SUMMARY

This document introduces the QROWD participatory framework, that aims to integrate crowdsourcing solutions for supporting the use cases defined in deliverables D1.1 and D2.2. The framework helps organisations to design a task to be undertaken by a crowd.

This document is of interest for the QROWD consortium to be informed about the crowdsourcing solutions we will implement, easing the integration between other components of the QROWD platform and the future crowdsourcing middleware prototype (D3.2)

Outside of QROWD, the deliverable is useful for future projects and initiatives intending to make use of crowds for supporting Big Data Value Chain processes. Furthermore, city councils and public institutions interested in including citizens in their strategies for solving mobility problems.

The first part of this deliverable discusses the fundamental concepts and definitions of crowdsourcing. The second part discusses solutions to support the use cases discussed in deliverables D1.1 and D2.2.

1 INTRODUCTION

This deliverable discusses the QROWD participatory framework that integrates crowd to support project use cases and activities that require human supervision. The first part of the deliverable introduces crowdsourcing and related concepts, such as human computation, collective intelligence, microtask crowdsourcing and gamification. Then, the second part describes how we applied these concepts to help with the implementation of the use cases described in deliverables D1.1 and D2.2.

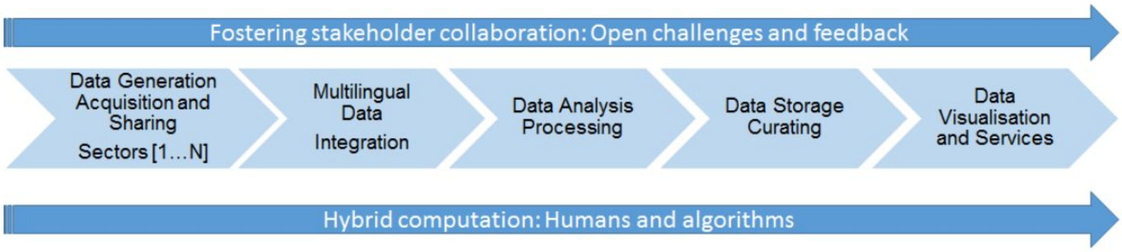

Figure 1: QROWD cross-sectoral Big Data Value Chain

Crowdsourcing supports the first three phases of the Big Data Value Chain (BDVC), shown in Fig. 1, as follow:

1. **Data generation and acquisition**: Crowd will be incentivised to install on their mobile devices an application (described in deliverable D2.2, Annex 2) to push traces of their daily trips to the QROWD infrastructure. Additionally, we consider the use of microtask crowdsourcing to support the Completing Mobility infrastructure use case BC2-UC#3 (described in deliverable D2.2, section 2.2.3).
2. **Data integration**: Data coming from multiple sources needs to be integrated to create more complete, consistent, and easy to query knowledge bases. Since in many cases automatic techniques are not able to perform data integration efficiently, crowdsourcing solutions may be used fruitfully for supervising data integration tasks **(Demartini, 2013)**.
3. **Data analysis**: We design ad-hoc tasks depending on the type of crowd, aimed to generate training sets for machine classifiers or to validate classifications computed by them.

2 CROWDSOURCING

In the last two decades, the rapid growth of web and communication technologies has facilitated the rise of crowdsourcing (CS), a strategy that allow companies and organizations to outsource activities to an undefined network of people **(Howe, 2006)**. Since CS does not have a unique and clear definition, in 2012, Estellés-Arolas et al. proposed a new definition that integrated more than 40 definitions already present in literature **(Estellés-Arolas, 2012)**:

“Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage what the user has brought to the venture, whose form will depend on the type of activity undertaken”

We clarify two details from this definition. First, recent developments in CS have extended the scope of the participation beyond “online activity”, to physical interactions, (e.g. ask the individual to move to a certain location to take a picture of it). Second, the concept of “voluntary undertaking” refers to the fact that individuals are not forced to undertake the task. As we will see, there are several types of rewards for individuals that decide to solve tasks.

From a business perspective, organizations, attracted by the possibilities offered by crowdsourcing, decided to include it in their business strategies **(Vukovic, 2009)**. A common way to do it is by relying on Human Computation **(Quinn, 2011; Von Ahn, 2008)**, a technique that enables the inclusion of human labour as part of automatic computation systems, and that turns out to be particularly useful when needed to perform tasks that machines cannot carry out effectively. Typical examples of these activities are video annotation **(Vondrick, 2010)**, text translation **(Zaidan, 2011)**, audio transcription **(Parent, 2010)**, and demographic surveys **(Huff, 2015)**.

The concept behind the CS success is the Collective Intelligence (CI): the capability of a group of individuals in reaching consensus for making a decision. The MIT Center for Collective Intelligence[1] defines Collective Intelligence as follows:

“Collective intelligence is shared or group intelligence that emerges from the collaboration, collective efforts, and competition of many individuals and appears in consensus decision making.”

Collective intelligence is at the base of the “wisdom of the crowds” concept, which affirms that crowds are not only intelligent, but also wise, and sometimes the collective opinion of a group of individual is considered better than the one of a single expert **(Suroviecky, 2004)**.

[1] http://cci.mit.edu/

2.1 *The four dimensions of crowdsourcing*

Malone et al. identified four dimensions to describe a crowdsourcing task: what, who, why, and how **(Malone, 2010)**.

2.1.1 "What" is going to be done?

This dimension refers to both what is required to the crowd and what goal the requester wants to achieve. In Malone's vision, the crowd can be asked to create new artefacts such as transcriptions or summarisations as well as evaluate existing materials in activities such as content reviewing or information validation.

Gadiraju et al. proposed six types of tasks that are commonly performed by users of CS platforms **(Gadiraju, 2014)**:

- **Information finding**: Contributors are required to search the web for information about some specific topic, such as people contacts, products, or events' details.
- **Verification and validation**: Contributors are asked to perform checks, validation, moderation, and/or spam detection on content, such as user-generated contributions on social media.
- **Interpretation and analysis**: Classification, categorisation, and tagging are typical examples of data interpretation and analysis activities that CS tasks can require.
- **Content creation**: Contributors are required to create new content, such as transcriptions of audio files, subtitles for videos, or drawing logos.
- **Surveys**: Contributors are asked to express opinions or give information in questionnaires or surveys. Typical examples are customer satisfaction analysis or demographic studies.
- **Content access**: Tasks can require contributors to access some online contents. For example, a service that needs to be tested.

2.1.2 "Who" is carrying out the task?

The "who" represents the type of crowd. Ideally, crowdsourcing contributors should belong to an undetermined group of people, and no assumptions should be made to their skills. However, contributors with particular skillsets are often required, such as multilingual users for translation tasks, or citizens of a particular city for location-dependent tasks. Online crowdsourcing platforms also often implement strategies aimed at identifying the best or most reliable contributors, who may receive access to special benefits and privileges.

2.1.3 "Why" is this task being performed? Which incentives exist?

This dimension concerns crowd motivation which is an aspect that plays a main role in CS **(Kaufmann, 2011; Chandler, 2013)**. According to the Oxford English Dictionary, motivation is "a reason or reasons for acting or behaving in a particular way", a desire or willingness to do something. Since this feeling of wanting drives behaviours of individuals and groups, it has been widely studied in several fields, such as psychology, economics, politics, or sociology.

Motivation can be of two different types: intrinsic and extrinsic **(Leimeister, 2009; Hossain, 2012)**:

- **Intrinsic** motivation concerns the internal benefits a task provide to the participants, eg. the participants enjoy or learn when performing a task.
- **Extrinsic** motivation arises from outside the individual and refers to behaviours driven by external rewards such as money or prizes.

Smart et al, identified six types of motivation specifically for crowdsourcing workers **(Ryan, 2000, Smart, 2014a, Smart, 2014b)**:

- **Economic (extrinsic)**: Contributors perform their tasks to receive a monetary payment.
- **Altruistic (intrinsic)**: Some tasks are developed for good purposes, e.g., for helping activities such as charity or medical research. Since these tasks do not offer any sort of explicit reward, their contributors are usually volunteers that perform the required work driven by selflessness.
- **Hedonic (intrinsic)**: Gamified systems engage users by adopting gamification strategy. The contributors perform this task to have a ludic experience.
- **Reputational (extrinsic)**: Contributors can pay particular attention to the quality and quantity of their work in order to increase their reputation. This can be driven either by a personal satisfaction, or by some goal the contributor want to reach, for example, some crowdsourcing platforms reserve the most profitable tasks to their best workers.
- **Instrumental (extrinsic)**: We talk about instrumental motivation when a contributor use a system because they wish to have some benefit from it. For example, to learn new things.
- **Other**: Any other type of motivation not included in the five previous ones.

Crowdsourcing tasks can offer rewards for their completion. Smart et al, list the four type of reward **(Smart, 2014)**:

- **None**: Not all CS tasks offer a reward to their contributors. Some of them are volunteers who are happy to participate in tasks that do not provide any sort of extrinsic recompense.
- **Monetary payment**: The monetary payment is a widely used reward provided for task resolution. Several CS platforms such as *CrowdFlower*[2] or *Amazon Mechanical Turk*[3] arose with the purpose to incentivises and support the paid crowdsourcing, by linking task requester and CS contributors. In paid CS, contributors are typically called “workers”.
- **Prize**: CS tasks can try to engage their contribution by giving them prizes for their volunteer participation. Common prizes for tasks participation are tickets for events, shopping coupons, or gadgets.
- **Other**: Any other type of reward not included in the three previous

[2] https://www.crowdflower.com
[3] https://www.mturk.com

categories.

Finally, the task reward can be fixed or variable **(Smart, 2014)**:

- **Fixed**: A priori decided, based on the amount of work or time required from the worker.
- **Variable**: The amount of reward depends on the quantity and/or quality of work performed by the contributor.
- **None**: If the task does not provide any payment.

2.1.4 "How" the task is going to be completed?

The "how" focuses on what way or manner a requester process a crowdsourcing exercise, that typically requires . Malone et al. identify three approaches for creating new content:

- **Collection:** A requestor has a collection of elements that need to be processed by humans. Instead of creating a single task covering all the items, the requestor divides the amount of work into small tasks that independent contributors can carry out separately, the results of which are collected (and, if necessary, aggregated). For example, given a collection of 1000 images that need to be annotated, 200 repetitions of a task requiring only five annotations might be a solution to save time (by parallelising the executions) and avoid overloading contributors.
- **Collaboration:** Multiple contributors can collaborate to the resolution of a task. Proper aggregation techniques turn the result of the group into a final task result, which is influenced by collaborators consensus. Collaboration strategies can also require tasks are executed sequentially (the output of a task is the input of the next one), this approach allows an incremental refinement of the overall result quality.
- **Contest**: The contest is a particularly appreciated solution when having tasks that need to be performed quickly (crowdsourcing for disaster relief) or when only a few excellent results are required. Contests often follow the "Prize" reward model.

2.1.4.1 Result aggregation

Human computation tasks require combining all crowd contributions to solve the global problem. Quinn and Bederson, identified seven class of problem of aggregation **(Quinn, 2011)**:

- **Collection:** Tasks in which crowd contributes to add or improve facts to a knowledge base.
- **Statistical processing of data:** Statistical functions can be applied to aggregate contributors results. For example, given a task that requires the crowd to judge an item by expressing a score on a specific measurement scale, the mean or median functions might be applied to

transform singular results into the final result.

- **Iterative improvement:** Some CS solutions can require providing a contributor with the answer of the previous one. This mechanism allows for an incremental improvement of the answer.
- **Active learning:** Machine classifiers can be trained by input large quantity of examples pattern along with annotation from which the classifier will learn. Therefore the classifier is the component who performs the aggregation.
- **Search:** Several tasks require contributors to search inside large collections of texts, images, or videos. The aggregation considers valuable only the contributions that contain the target of the search.
- **None:** Some CS tasks do not require aggregation at all.

2.1.5 Quality control

Since several reasons could interfere with the success of an experiment, quality control of the task results is an essential aspect of every CS initiative. Quinn and Bederson, identified nine types of quality control **(Quinn, 2011)**:

- **Output agreement:** Two or more contributors perform independently and simultaneously the same task. Only when all of them agree on the same answer, this is accepted. This strategy is widely adopted in ESP games **(Von Ahn, 2004)**.
- **Input agreement:** Two or more contributors perform independently and simultaneously the same task but, differently from the "output agreement", the task inputs could not be the same. By describing the own task each other, contributors are required to establish if their inputs are the same. If all the contributors agree, then the descriptions are considered correct.
- **Economic models:** Economic incentive structures can discourage cheating attempts and lead to a better task result, with higher overall quality.
- **Defensive task design:** This idea consists of designing tasks that are easier to perform correctly than find a way to cheat on.
- **Redundancy:** Multiple task repetitions through executions performed by different contributors allows the collection of redundant task results. Analyses of agreement among contributors permit to find out outliers and to validate the collected task results.
- **Statistical filtering:** By aggregating the results through appropriate statistical functions is possible to filter out outliers that can undermine the overall task quality.
- **Multilevel review:** A task is performed by a group of contributors and the produced results are subsequently evaluated by another group..
- **Automatic check:** Some problems are difficult to compute but easy to verify. Solutions for this class of problem can be automatically checked.
- **Reputation system:** Some CS platforms implement reputation scoring systems that motivate contributors to produce high-quality results. For

example, by submitting good quality results, contributors can access to more desirable tasks

2.1.5.1 *Task request cardinality*

Some tasks require to be performed by many contributors; others may need to be carried out by just a few or even only one participant. Quinn and Bederson defined a dimension named “task-request cardinality” that refers to the number of contributors that need to be involved in a task resolution **(Quinn, 2011)**. This can value:

- **One to one:** A specific contributor is asked to carry out a specific activity.
- **Many-to-many:** Collections of data need to be somehow operated by many contributors. For example, a shop has a collection of products and wants to make use of crowd to label them. It can be done recruiting a large number of contributors, asking each of them to label few products, and obtaining redundant tags for each item.
- **Many-to-one:** Many contributors are looking for a single result (for example an image in a collection, or the design of logo).
- **Few-to-one:** Few contributors give redundant results to each proposed task.

2.2 Types of crowdsourcing

There exist multiple types of CS. The most interesting for the QROWD purposes are microtask crowdsourcing, contests, hackathon, participatory sensing, and games with a purpose.

2.2.1 Microtask crowdsourcing

A largely widespread form of crowdsourcing is microtask CS. A microtask is a small task, also called HIT (Human Intelligence Task) or job, what needs to be performed by *contributors*. Microtasks are created by *requesters*, such as companies, researchers, or practitioners who require to include human labour in some parts of larger systems.

Once a requester designs a microtask based on their data and needs, they have also to specify the number of contributors required to perform it. Then, the requester uploads the microtask in an online platform specifically designed for this purpose, such as *Crowdflower* or *Amazon Mechanical Turk*. Contributors from all around the world can access the platform, consult a list of available tasks, perform one or more microtasks, and when finished, if expected, receive the due payment (in this case the platform withholds a percentage of the payment).

Microtask crowdsourcing offers several advantages. The fist of them is the wide availability of online contributors from all around the works, at any moment of the day. This feature guarantees tasks are performed quickly, especially when they are designed to be performed in a parallelizable way.

When running microtasks, requesters cannot choose specific contributors who can freely decide which task/s to perform. For this reason, requesters can implement quality checks in their microtasks to guarantee the quality of their results is satisfiable. Several types of quality check exist, some based on the agreement between contributors, others focus on individual microtasks results. For example, requesters can purposely include gold standards in their tasks. These are questions (typically multiple-choice) having answers known a priori, and then automatically checkable in task execution time. Requestors can reject the entire task results when contributors do not answer gold standards correctly.

There exist some alternatives to crowdsourcing markets. For example, *Pybossa*[4] is a crowdsourcing platform and framework, designed to support volunteer-driven projects, where, differently from *Amazon Mechanical Turk* and *Crowdflower*, no payment is expected for contributors who carry out tasks.

2.2.2 Contest and hackathon

Open challenges are a popular engagement method adopted by organisations that decide to use crowdsourcing to solve a specific problem. Participants, individually or together, look for solutions to the given problems to reach a final prize.

2.2.2.1 Idea competition

Idea competitions are initiatives aimed to include crowd for generating ideas and solutions to a given problem **(Ebner, 2009; Piller, 2006)**. Walcher defined ideas competition as:

An ideas competition is the invitation of a private or public organizer to a general public or a targeted group to submit contributions to a certain topic within a timeline. An idea-reviewers committee evaluates these contributions and selects the rewarded winner(s). **(Walcher, 2007)**

Ah example of idea competition is the “Netflix Prize” challenge, carried out by the *Netflix*[5] company that aimed to find the best collaborative filtering algorithm to predict user ratings for movies.

The *BellKor's Pragmatic Chaos* team won the competition, developing an algorithm that improved the predicting rating of the previous Netflix algorithm by 10.06%. The prize for the winner amounted to US$1,000,000.

4 http://pybossa.com/

5 https://www.netflix.com/

2.2.2.2 Hackathon

The word "hackathon" is a portmanteau of the words hack and marathon and refers to events where software developers collaboratively code in an intense manner over short periods of times. Typically hackathons last few days in which participants, individually or grouped, can choose to work freely on a given project **(Briscoe, 2014; Trainer, 2016)**.

2.2.3 Participatory sensing

Modern mobile devices such as smartphones, tablets, notebooks, and GPS navigators are equipped with accurate sensors, able to collect data of various nature **(Boulos, 2011; Sheth, 2009)**. Motion sensors such as accelerometers, gravity sensors, and gyroscopes measure acceleration and rotational forces. Environmental sensors, including barometers, photometers, and thermometers, can measure external parameters, such as ambient air temperature, pressure, illumination or humidity. Finally, position sensors, such as magnetometers and orientation sensors permit to measure the position of a device (linear, angular, or multi-axis).

Participatory sensing is a CS technique that aims to exploit the large amounts of data these sensors can collect to expand the scope of traditional CS beyond online participations. Smart cities projects adopt participatory sensing approaches to improve aspect such as city mobility, transportation systems, or community services **(Gabrys, 2014)**. For example, data collected from crowd devices can be analyzed to detect traffic jams or other related mobility events.

2.2.4 Gamification and 'Games with a Purpose'

Gamification is the use of game design elements in non-game contexts **(Deterding 2011)** . In CS, this technique is adopted to make tasks more enjoyable and fun to encourage more users to complete them. While gamification can, in some cases, be used as an alternative to other incentives (such as monetary payment), the user must still be provided with some reward (such as social status) to motivate them; however, the most appropriate reward will vary with the context of the task **(Deterding 2012)**. Gamification of CS tasks has been shown to increase both participation and quality of work **(Hamari, 2014; Feyisetan 2015)**.

Morschheuser (2016) et al. identify four common types of CS gamification:

- **Points/Scores:** keeping track of the number of tasks a user has completed, or some other metric (e.g. users may receive more points for completing more difficult tasks, or for completing tasks quickly).
- **Rankings/Leaderboards:** Publicly displaying the top players' scores on the platform. Short-term leaderboards (e.g. "This Week") are recommended over "All Time" leaderboards to avoid demotivating new or

casual users.
- **Level Systems:** systems where once a user earns a certain number of points (often referred to as “xp” for this purpose), they are reset to zero, and increment the user’s “level”. This may simply cosmetic, to give users a sense of progression (and/or social standing), or it may unlock new rewards and privileges for higher level users.
- **Manifold gamification:** systems which build gamification into the task itself by requiring contributors to, for example, utilise reaction time, or puzzle solving skills to complete the task. Fold.it[6], a crowdsourced science task about folding proteins, is a good example of this.

Games with a Purpose (GWAPs) are multiplayer online games designed to be fun and accomplish tasks that are easy for humans but beyond the capability of nowadays computers **(Von Ahn, 2006a)**. GWAPs does not rely on altruism or financial incentives, but only on the hedonic one. Results collected by GWAPs can be used for several purposes, such as annotate and classify collected data or crowdsource general knowledge. Some examples of GWAPs are:

- *Phetch*, a game for collecting explanatory descriptions of images on the web **(Von Ahn, 2006b)**.
- *Verbosity*, a game which tests participants common sense knowledge, to collect fact will be used for AI purposes **(Von Ahn, 2006c)**.
- *Tag a Tune*, a game in which players describe songs. Results are used to develop systems which allow for music search based not only on titles or lyrics but also on descriptions having, for example, emotional elements **(Law, 2007)**.

3 QROWD FRAMEWORK

The previous chapter introduced crowdsourcing and discusses some frameworks used to support the design and the deployment of crowdsourcing task. Nevertheless, those frameworks are not sufficient to support the requirements of our use cases adequately. We defined a new framework, that combines the Malone **(Malone, 2010)**, Smart **(Smart, 2014)**, and Quinn **(Quinn, 2011)** works, and extends them with a few additional dimensions particularly useful to describe the proposed use cases solutions discussed in the next chapter.

Table 1 shows the overview of the framework: For each considered dimension we show the original source and sample values. We have introduced some additional elements that we considered for modeling citizen sensing crowdsourcing tasks such as those in the given use cases. These elements are highlighted in bold.

[6] http://fold.it

Table 1: Overview of the QROWD Framework

Dimension	Based on	Sample Value
What	Malone et al.	Information finding Verification and validation Interpretation and analysis Content creation Surveys Content access **Passive sensing** **Active sensing**
Who	Malone et al.	**Expert** **Citizen** **Whoever** **Specific contributor**
Human skill	Quinn and Bederson	Visual recognition Language understanding Basic human communication **Physical**
Device required	Novel	**Mobile** **PC** **None**
Device limited resources	Novel	**Battery** **Storage** **Bandwidth** **CPU** **None**
Number of interactions	Novel	**[0 - N]**
Why - Motivation	Why (Malone et al.) - Motivation (Smart et al.)	Economic Altruistic Hedonic Reputational Instrumental Other
Why - Type of reward	Why (Malone et al.) - Type of reward (Smart et al.)	None Monetary payment Prize Gamification element Other

How	Malone et al.	Partition (Collection) Collaboration Contest **Sensor**
Acceptable question delay	Novel	**Immediate (seconds)** **Short (minutes)** **Medium (hours)** **Long (days)**
Acceptable resolution delay	Novel	**Immediate (seconds)** **Short (minutes)** **Medium (hours)** **Long (days)**
Quality control	Quinn and Bederson	Output agreement Input agreement Economic models Defensive task design Redundancy Statistical filtering Multilevel review Automatic check Reputation system
Aggregation	Quinn and Bederson	Collection Statistical processing of data Iterative improvement Active learning Statistical Search Iterative improvement None **Formula**
Task request cardinality	Quinn and Bederson	One-to-one Many-to-many Many-to-one Few-to-one

For Malone et al.'s "**what**" dimension, we added two values specifically tailored to sensing tasks:

- **Passive sensing** is when collaborators carry with them a sensor device that automatically collects measurements without intervention of the collaborator.
- **Active sensing** is when collaborators are asked to trigger the measurement or collection at regular intervals or following a request e.g., take a picture of a specific item whenever they see it.

For Malone et al.'s "**who**" dimension, we identify four possible actors:

- **Experts**: People who have expertise in the domain of the task.
- **Citizens**: A person that is located in a determined geographical area (a "city") for a determined time frame. Note that under this definition tourists are considered citizens.
- **Whoever**: Anyone can carry out the task - no particular constraints or requirements are given.
- **Specific contributor**: Only a specific individual can carry out a task. For example, when multiple interactions over the time with the same contributor are required.

We extended Quinn and Bederson's "human skill" dimension with the "Physical" value which refers to some tasks that can require physical movement or action to be undertaken, such as driving or cycling.

The **Device required** dimensions specify if/which device is required the contributor to perform a task:

- **Mobile**, such a smartphone or a table when the task requires people mobility.
- **Desktop**, probably mode adequate for task that need contributors in a comfortable workplace.
- **None**, if the task completion does not require any devices.

The **Device limited resources** dimension applies when the execution of a task can negatively affect devices functionalities. It regards:

- **Battery** needs to be safeguarded.
- **Storage** has to be prevent by running outs.
- **Bandwidth** can be lacking in some areas or lead to additional costs for the contributors**.**
- **CPU** can be particular limited in old devices.

A new dimension **Number of interactions** specify how many interactions with the crowd are required. For example, this applies when is needed to limit the number of interaction with contributors to avoid to bother them.

The Malone et al. dimension **how**, is now extended with the "Sensor" sample value, which means the crowd should complete the task by performing sensing activities.

We consider in our framework two temporal dimensions:

- **Acceptable question delay**: specifies the maximum latency allowed between task generation and its assignment.
- **Acceptable resolution delay**: defines the maximum delay allowed between task assignment and task completion.

Previous frameworks did not consider temporal dimensions. These are important since they allow us to define temporal constraints that guarantee result ‘freshness’.

We added the **Formula** sample value to the **Aggregation** dimension. This refers to the algorithm that aggregates the set of individual contributions into a final result. It differs from the statistical aggregation because it can involve more sophisticated aggregation techniques, such as clustering.

The last dimension **Impact on the device battery** applies to mobile devices typically affected by limited battery life. The framework allows limiting on impact in device battery life for tasks executed on mobile devices.

4 CS SUPPORT TO THE BDVC IN QROWD

We identified several CS based solutions for some of the project's use cases detailed in deliverables D1.1 and D2.1, belonging to three specific phases of the Big Data Value Chain (BDVC): (1) Data generation, acquisition and sharing, (2) data integration, and (3) data analysis process.

4.1 Data generation, acquisition, and sharing

CS is a valuable way to generate and acquire data. In QROWD we adopt crowd solutions to generate and exploit mobility related data like type and location of mobility infrastructure and movement patterns inside cities to inform the design of policies to improve traffic and reduce CO_2 emissions.

4.1.1 Participatory sensing through i-Log

The University of Trento is developing *i-Log* (see Annex 1 of D2.2 for details), a mobile application that once installed collects information from the device sensors, such as GPS, accelerometers, and gyroscopes. The campaigns organised by the Trento Municipality will invite Trento citizens to join a participatory sensing initiative by installing i-Log in their smartphones. The data gathered by i-Log is raw (just values generated by sensors - not post-processed) which is then collected and sent to the InfAi data centre to be stored and later processed.

4.1.2 The ideas competition

The “ideas competition”, described in detail in Deliverable 2.1, is a crowdsourcing contest. The event involves active citizens in activities aimed to collect from ideas and opinions about the city and their services. Table 2 shows the QROWD framework dimensions for the ideas competition:

Table 2: QROWD framework dimensions for the ideas competition

What	Who	Human skill	Device required
Ideas	Citizens (from Trento, full time resident)	Language understanding Basic human communication	None
Impact on the device battery	**Number of interactions**	**Why - Motivation**	**Why - Type of reward**
None	Many	Economic Altruistic	Prize
How	**Acceptable question delay**	**Acceptable resolution delay**	**Quality control**
Contest, survey	Long(days)	Long(days)	Multilevel review
Aggregation	**Task request cardinality**		
None	Many-to-many		

4.1.3 Completing mobility infrastructure

Participatory sensing techniques allows the collection of large amounts of information from mobile devices. Furthermore, city councils (such as the Municipality of Trento) may not have a complete and/or accurate account of all the relevant mobility infrastructure items in the city, e.g. bike racks, free parking slots, etc. CS based solutions can exploit the crowd's capabilities in identifying and classifying static city items not yet recorded, and detecting and correcting inaccurate records (see D7.1 for details).

The "Completing mobility infrastructure information through crowdsourcing" (BC2-UC#3) use case aims to create or extend city maps by including not-tracked static components, like bike racks or parking spots. It can be done on the spot, by asking people to collect information around in the city, or on the web, by using online microtasks.

We will compare experimentally both options to establish which is the most cost-effective, extensible and adaptable to different cities and infrastructure types. We plan to run crowdsourcing experiments aimed to study which are the best strategies to allow CS participants to contribute to the creation of accurate static maps.

At time of submission of this deliverable, we have developed a crowdsourcing microtask, based on Google Streetview[7], aimed at identifying bike racks in Trento. CS workers are asked to explore the city in a 3D virtual environment to find bike racks and, when one of them is identified, workers are required to take three pictures of it from three different angles. These allow us to identify the exact coordinates (latitude and longitude) of the bike rack, that are then stored for the subsequent processing. Figure 2 shows screenshot of the task interface, and Table 3 shows its QROWD framework dimensions.

Figure 2: Interface of the microtask used to explore the city to find bike racks

Table 3: QROWD framework dimensions for bike rack identification tasks

What	**Who**	**Human skill**	**Device required**
Information finding	Whoever	Visual recognition	PC
Impact on the device battery	**Number of interactions**	**Why - Motivation**	**Why - Type of reward**
Unconstrained	At least one	Economic	Monetary payment
How	**Acceptable question delay**	**Acceptable resolution delay**	**Quality control**
Collection	Long (days)	Long (days)	Output agreement Defensive task

[7] https://www.google.com/streetview

			design Statistical filtering Multilevel review
Aggregation	**Task request cardinality**		
Formula	Many-to-many		

We plan to improve the current task and design other microtasks aimed to validate the bike rack previously detected. In the validation tasks, workers will be shown with photos of the bike rack, asked to validate them, and invited to specify additional information about the bike rack, such as its model and capacity. Figure 3 shows a mockup of a graphical interface for the bike rack validation task. In D7.1, we demonstrate a general system architecture for realizing distributed crowdsourced streaming for disambiguating/deduplicating bike racks collected with this task with bike racks collected from other data sources (e.g. Open Street Maps, data from municipalities).

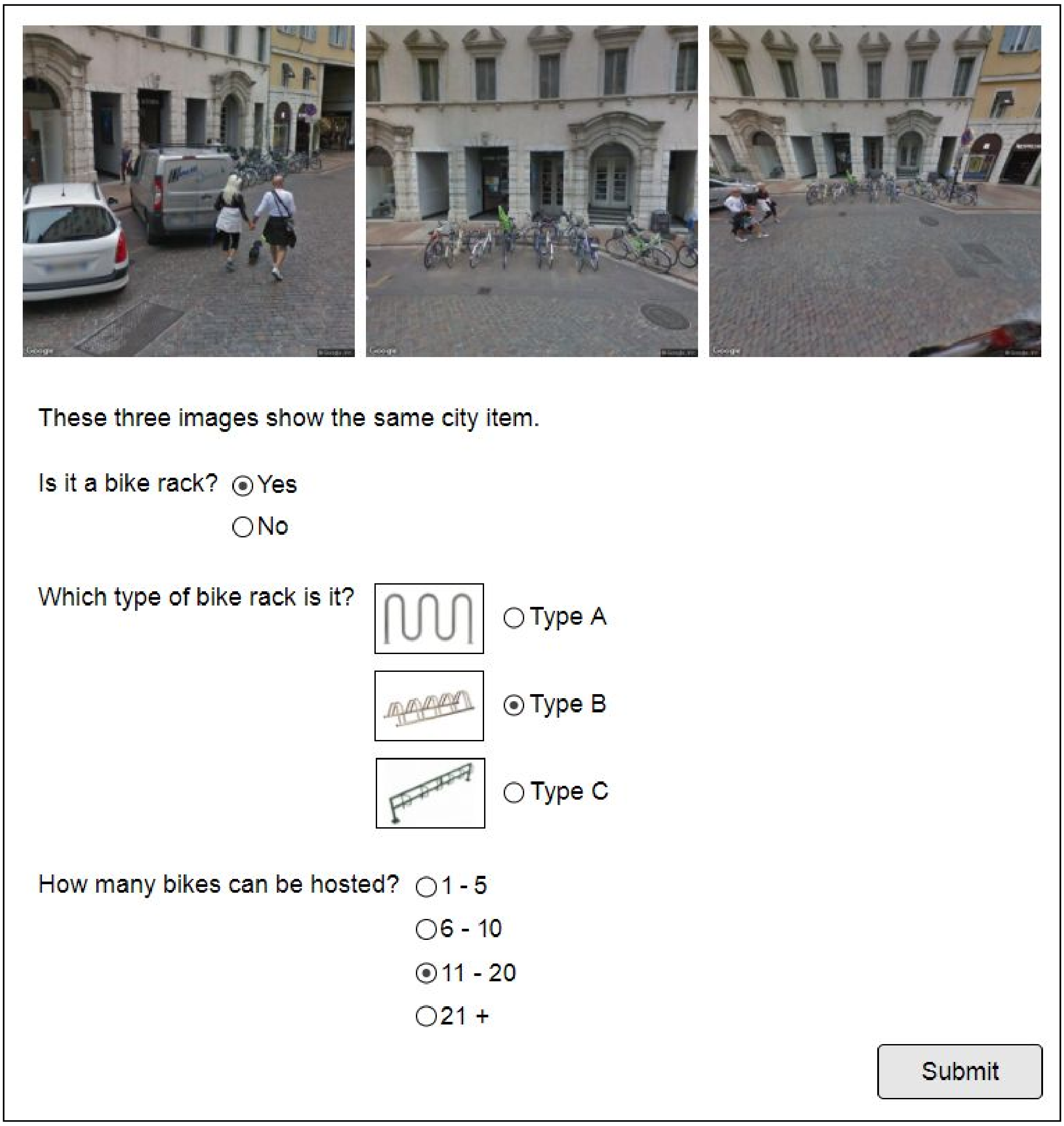

Figure 3: Mockup of a graphical interface for the bike rack validation task

Table 4 shows the QROWD framework dimensions for the bike rack evaluations tasks.

Table 4: QROWD framework dimensions for bike rack evaluation tasks

What	Who	Human skill	Device required
Verification and validation	Whoever	Visual recognition	PC

Impact on the device battery	Number of interactions	Why - Motivation	Why - Type of reward
Unconstrained	At least one	Economic	Monetary payment
How	**Acceptable question delay**	**Acceptable resolution delay**	**Quality control**
Collection	Long (days)	Long (days)	Output agreement
Aggregation	**Task request cardinality**		
Statistical	Many-to-many		

We plan to study spatial crowdsourcing techniques to identify strategies for guarantee overall city coverage.

The output of the work will be a catalogue of the Trento bike racks, that provides location, capacity, type, and at least one clear photo for each of the bike racks catalogued.

4.2 Data integration

An important part of the QROWD project concerns data integration. Data from different sources, either private or public, need to be integrated to obtain a more comprehensive, organised and easily-queryable knowledge base. Since different data sources can have different structures, nature and language, the integration process can encounter several difficulties, e.g. data inconsistencies, entity resolution and disambiguation, and missing data are common issues that need to be tackled when trying to merge together different data sources.

Integration can be carried out automatically when it is relatively simple and can be performed smoothly. However, since possible errors will depreciate the quality of the merged data, more challenging cases may require human supervision. As humans are usually better than machines in dealing with inconsistent data and in performing entity resolution, CS has been traditionally used to support these activities **(Wang, 2012; Demartini, 2012)**.

We will adopt hybrid data integration approaches, which involve both machine and human computation whenever automated techniques show limitations. The *Entity Name Service (ENS)* tool developed by *AI4BD* and described in deliverable D7.2 will provide unique URLs for given entities by relying on crowdsourcing when confidence scores of entity resolutions are unsatisfactory.

4.3 Data analysis process

Data analysis is an important process of the *BDVC*. When possible, it is performed by machines: algorithms process large amounts of data and produce meaningful outputs. Unfortunately, this process is not always possible. Artificial Intelligence (AI) has limitations that sometimes do not allow them to operate with a satisfactory degree of efficiency or accuracy when processing certain types of data, in particular unstructured data, such as texts, images, or video. Since humans can better interpret and operate on this type of data, we plan to involve crowdsourcing into the analysis process to help the machine-learning algorithm when the confidence of its analysis is below a desirable threshold.

4.3.1 Image annotation

Images or video coming from public cameras are a continuous source of valuable information. Machine learning algorithms can be trained to identify particular elements present in the photos, such as pedestrian, bikes or free parking places. This techniques allow automatic classification or labelling of images, in order to extract information from them at a later date. For the highest degree of accuracy, machine learning techniques need to be trained with sets of already classified items. For the training to be effective, the sets needs to be sufficiently large (in the order of thousands of classifications) and sufficiently varied (classifications should include examples of all possible categories).

A recurring problem of machine learning is the lack of training sets large and varied enough. When the classification can be performed as a HIT, CS can be an inexpensive, fast, and easy solution to label data for training sets **(Lease, 2011)**. The broad availability of crowd-workers allows the creation of training sets for machine learning in a short time (generally much less than that required to hire someone for labelling all the data) and within a relatively inexpensive way. Proper quality control techniques guarantee the training sets are correctly created. Also, if needed, CS approach easily allows the redundant data labelling (when multiple workers label the same data). In these cases, the agreement among workers can be stated as an index of quality the gathered annotations.

To support the **Completing information about mobility infrastructure through spatial crowdsourcing** use case **(BC2-UC#3)**, we plan to outsource images and videos from cameras installed in Trento to crowd workers, and require them (via microtask) to track occupancy of mobility infrastructures, such as free parking spots or bus stop areas. To carry out this task, we plan to reuse pre-existent components such as CCTV cameras made available by the Municipality of Trento. By using existing infrastructure, we reduce costs to the consortium, the only overhead being payment of the crowdworkers.

For example, to support the **Parking Probabilities** use case **(BC2-UC#2)** we could exploit images recorded by a camera oriented toward a parking area to automatically count the number of free places to analyse the daily parking availability. The datasets necessary to train machine learning algorithms can be obtained through CS microtasks that require workers to label the given images. Table 4, shows the QROWD framework dimensions for this task.

Table 5: QROWD framework dimensions for image annotation tasks

What	**Who**	**Human skill**	**Device required**
Interpretation and analysis	Whoever	Visual recognition	PC
Impact on the device battery	**Number of interactions**	**Why - Motivation**	**Why - Type of reward**
Unconstrained	Many	Economic	Monetary Payment
How	**Acceptable question delay**	**Acceptable resolution delay**	**Quality control**
Collection	Long (days)	Long (days)	Output agreement
Aggregation	**Task request cardinality**		
Formula	Many-to-many		

4.3.2 CS for data analysis

According to the General Architecture (Deliverable 8.1 - Chapter 3), the i-Log application collects data from citizens' mobile devices and stores it in its raw form (without any post-processing).

The **Modal Split** use case **(BC2-UC#1)** aims to train machines to automatically infer the transportation mode of a citizen’s trip by using data collected from their mobile devices.

This activity consist of two phases: first, the algorithm tries to segment the information collected in a day into journeys (the full travel, from beginning to end, for example commuting from home to work) and trips (individual parts of the journey, for example, from home to the bus stop); second, the algorithm tries to infer the correct transportation mode of each step. In some circumstances, it might be that machines are not be able to infer transportation mode with a satisfactory confidence interval.

In these cases, human intervention is necessary to help with both trip segmentation and classification process. We will design a component that given a trip that cannot be accurately classified by a machine will decide the best option between:

1. Asking for a trip classification directly the citizen who made it.
2. Running a microtask and asking the crowd for the trip classification.

Both alternatives have advantages and disadvantages: (1) should allow for the best classification, since the citizen who made the trip is the one who has more information about it. Nevertheless, two conditions need to be guaranteed:

- We hypothesize that the number of the interaction request with the citizen impact how cooperative they are. Too many interactions could annoy the citizen prompting them to ignore further request or even for uninstall the app from their smartphones.
- We consider a contributor may forget trip details if the delay between the trip occurrence and the classification request is too long. We plan to keep this delay under a certain reasonable threshold.

(1) can be implemented in the i-Log app, by asking citizens for fast and straightforward questions, which expect precise answers. Figure 4 displays two possible questions may be asked to citizens: The left one confirms a change of transportation mode when computing a trip segmentation. The right one requires citizens to specify the occupancy rate of a bike rack. The given information can help machine classifications.

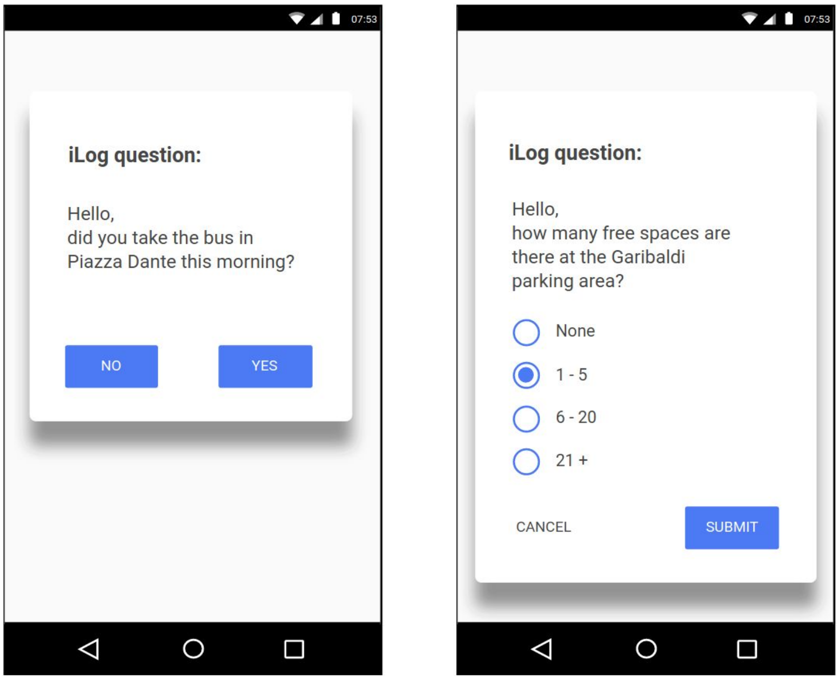

Figure 4: Examples of *i-Log* questions

Nevertheless, the number of times it can be adopted is quite limited: Even the most collaborative citizens may get annoyed by the continuous questions popping out on their mobile devices screens. Naturally, this can lead to negative consequences, such as unsatisfied *i-Log* users who provide wrong or random answers, and in the worst case, deliberately uninstalls *i-Log*. Therefore, by adopting the (1) solution, we need a way to ensure the citizens answer honestly. Techniques such as “gold standard” questions - questions with a known correct answer, used to verify the honesty and accuracy of CS workers - can be implemented to detect outliers or malicious *i-Log* users.

Table 6: QROWD framework dimensions for citizens trip classification tasks

What	**Who**	**Human skill**	**Device required**
Verification and validation	Citizens (from Trento, full time)	Language understanding, Physical	Mobile
Impact on the device battery	**Number of interactions**	**Why - Motivation**	**Why - Type of reward**
Constrained	Limited	Economic Altruistic	Prizes
How	**Acceptable question delay**	**Acceptable resolution delay**	**Quality control**
Collection	Medium (hours)	Medium (hours)	Defensive task design
Aggregation	**Task request cardinality**		
None	Few-to-one		

On the other hand, given the availability of crowdworkers, solution (2) places (theoretically) no limitations on the number of trip classification we can require to the crowd. Also, a microtask can be repeated multiple times or it can executed in parallel with others in order to measure the workers agreement as indicator of the trip classification quality.

Nevertheless, asking somebody to classify a trip made by another person could be challenging, and the correctness of the classifications depends on several factors:

1. The amount of data has to be large be enough to guarantee a proper trip classification. There needs to be enough data to provide input to a HIT, potentially including additional data to that needed by the machine task, due to the intrinsic difference between human and machine processing.
2. The quality of the data needs to be accurate and without noise.
3. The intrinsic uncertainty of the trip: some trips may be impossible to be correctly identified. For example, it could be complicated to determine if a trip was made by a car or by a taxi.

4. The user interface needs to provide workers with an intuitive representation of a moving objects spatial trajectory **(Yan, 2013)**, and a way to match the trip path with the surrounding layout.

Table 7: QROWD framework dimensions for crowd trip classification tasks

What	**Who**	**Human skill**	**Device required**
Interpretation and analysis	Whoever	Visual recognition Language understanding	PC
Impact on the device battery	**Number of interactions**	**Why - Motivation**	**Why - Type of reward**
Unconstrained	1	Economic Reputational	Monetary Payment
How	**Acceptable question delay**	**Acceptable resolution delay**	**Quality control**
Collection	Long (days)	Long (days)	Output agreement
Aggregation	**Task request cardinality**		
Formula	Many-to-many		

Solutions (1) and (2), and the automatic approaches can also be applied simultaneously to evaluate each other, and further build the training set of the automated classifier.

4.4 QROWDsmith

We are currently developing “QROWDsmith”, a CS platform that leverages the beneficial aspects of gamification, discussed in Section 2.4, as part of the participatory framework.

Importantly, we intend to directly integrate gamification elements, as well as to allow for real-time communication between both task-contributors and task-creators. This will allow for communication and community building, as well as the implementation of limited-time “contests”. Figure 5 shows the interface of the QROWDsmith platform.

QROWDsmith is being developed using Python/Flask. We are using Heroku, to enable easier (re)deployment, with scaling virtual resources. However, QROWDsmith can also be deployed independently. Recruitment to the platform - particularly for paid workers - will be handled through external channels such as CrowdFlower or Twitter. The progress will be described in deliverable D3.2 (Crowdsourcing services).

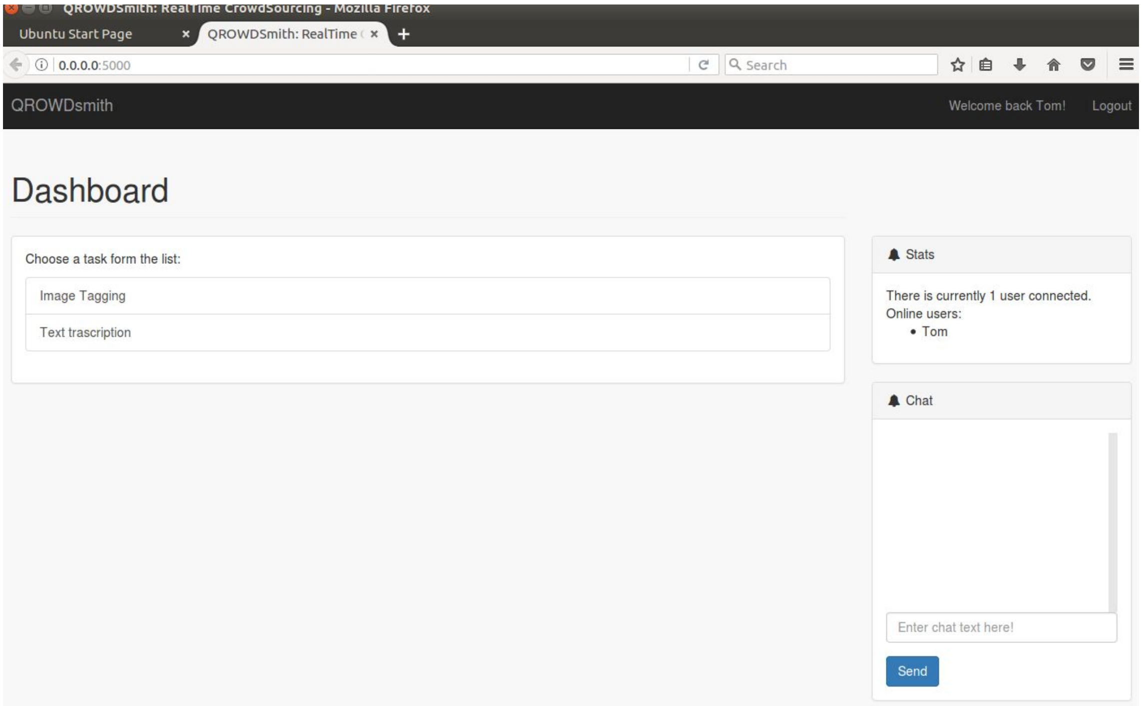

Figure 5: Interface of the QROWDsmith platform

5 CROWD COMPONENT ARCHITECTURE

Given an instance of the participatory framework describing the task to be undertaken by the crowd, plus any other required data (e.g., link to resources on which the crowd will operate), the *Crowd Component* proceeds as follows:

1. Support the decision (through the decision component) of which task and question templates (stored in the Question template and Task template libraries) is applicable to the task defined by the user with the participatory framework instance. The decision component also guides the setting of the parameters of the templates.
2. Deploy (through the deployment manager the task template instantiated in Step 1) in the appropriate platform.
3. Collect the contributions from the platforms/channels and combined them into a final result that is quality checked (through the *Aggregation and Quality Component)*
4. Push the results to the context broker for their delivery to the appropriate component or storage.

Figure 6 shows the scheme with the modules of the Crowd Component, and how they interact among them and between other components of the QROWD platform.

Task Templates: a task template is the set of source codes and libraries, and resources (such as texts or images), containing the logic of a specific task, the appropriate channel, and the logic for aggregation and quality assurance. We will develop a set of task templates based on our mobility related use cases and provide guidelines to develop new templates that can be added to the library. When needed, a template can be deployed to a specific crowdsourcing channel to be performed by the crowd through the deployment manager.

Question templates: a question template is a markup file that defines a closed question to be asked to one or more contributors, including type of question (single choice, multiple choice, click on a point, etc), text, and range of answers.

Decision Component: based on the participatory framework instance given and additional information on previous interactions with contributors (e.g. how many questions have been asked to them, if they have been responsive, etc), the component supports the process for deciding which task question template should be deployed. The component also guides the process (automating it when possible) of instantiating the task with the appropriate parameters. The instantiated task is passed to the deployment manager.

Deployment Manager: receives an instantiated task or question template and manages its deployment in the appropriate crowdsourcing channel and the collection of the results. Results are then outputted to the aggregation and quality component.

Crowdsourcing channel: the interface through which contributors will engage with the task, it can be one of:

- *Gamification platform*, such as, *QROWDsmith* (Described in Sec. 4.4), that allows for collaborative and real-time tasks, by leveraging on the gamification techniques.
- *Crowdsourcing platform*, such as *CrowdFlower*, which allows to involve a large amount of crowdworkers, in performing large-scale microtasks.
- *i-Log*, that consent to involve citizens, who are the contributors that have a better local knowledge of the city.

Aggregation and quality component: executes the aggregation and quality logic defined in the task template on the collected results. If the results are not of sufficient quality, it instructs the deployment manager to redeploy the task for a further iteration. The loop continues until the result quality is satisfactory. The final results are outputted to the context broker or QROWDDB.

The CROWD component interacts with three external components part of the QROWD platform

- *Cassandra db*: The database, administered by the University of Trento that contains the raw data collected by i-Log.
- *QROWD db*: The database containing input data for the instantiated task

templates, e.g. results of machine classification that need to be verified by humans; and the log of previous interactions with contributors.
- Context Broker: Handles communication with other platform components, guaranteeing the extensibility of the platform.

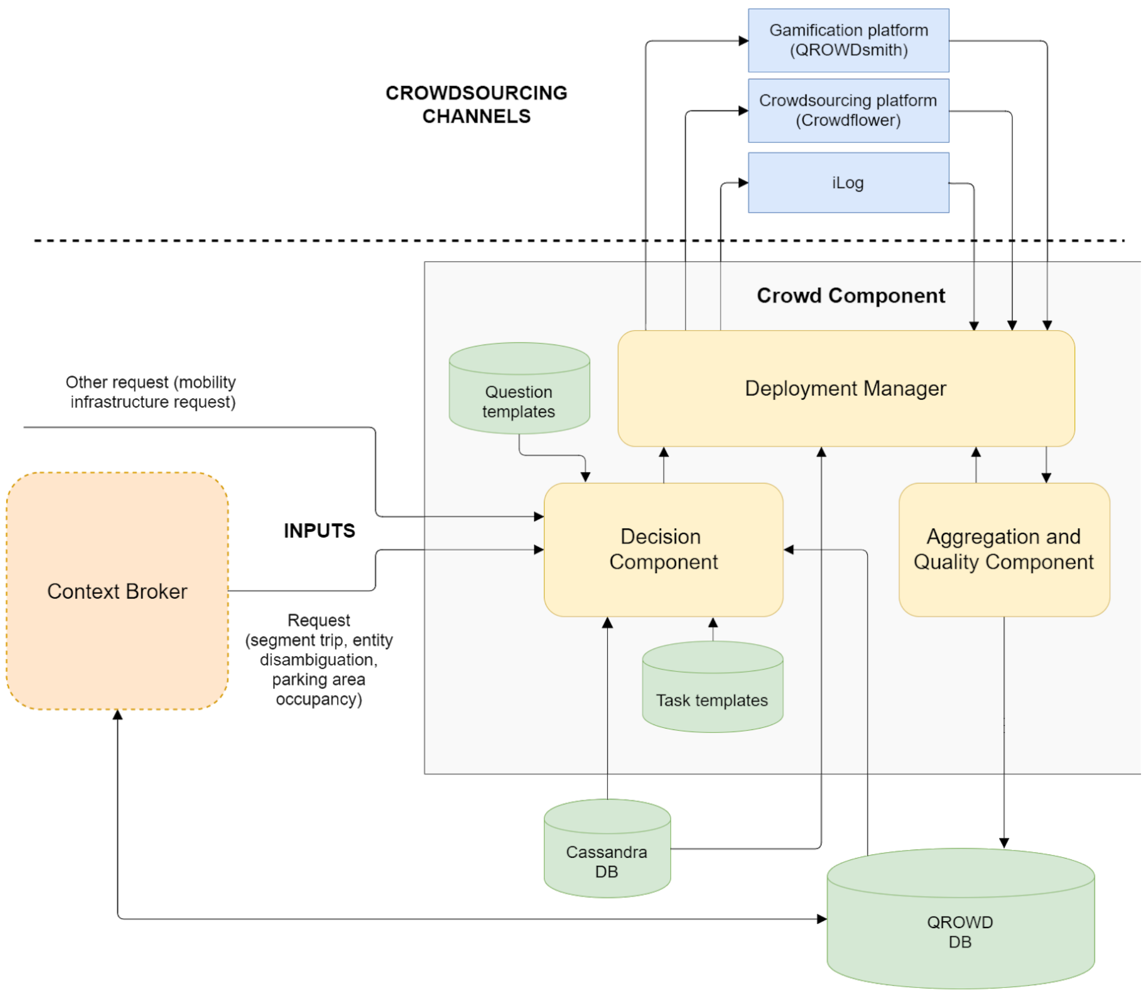

Figure 6: *Crowd Component* architecture

Figure 7 shows the sequence diagram of the *Crowd Component* for the trip segmentation activity, when citizen input is required. When the QROWD *Context Broker* issues a request for a new classification, the Crowd Component queries the QROWD and *Cassandra* databases to obtain the information necessary to decide the appropriate question to be asked to the citizen. After identifying the question type and the crowdsourcing channel (*i-Log* users in our example), the Crowd Component gets the proper question template from the QROWD database, and injects into it the parameters found on the request received, generating an instance of the current question. The *Crowd Component* sends the question instance to the *Context Broker*, which pushes it to the citizen mobile device through i-Log. The question instance remains pending until the citizen provides an answer. When the citizen provides a response, it is sent back to the *Context Broker*, and then to the QROWD database to be stored.

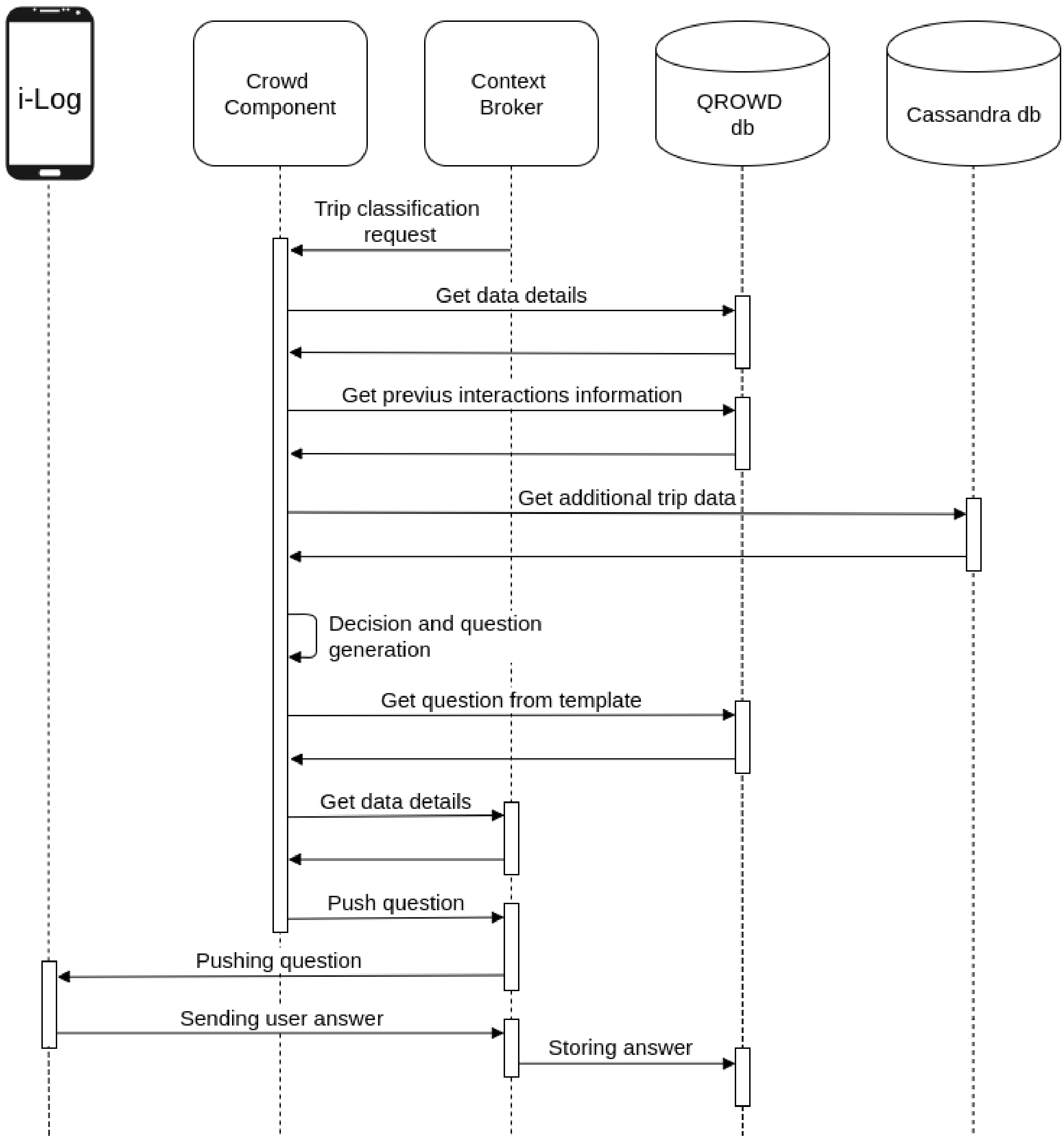

Figure 7: Participatory framework in trip-segmentation use case

6 CONCLUSIONS

This document discussed the QROWD participatory framework focusing on the role of citizens and crowd in supporting the Big Data Value Chain of the project. After the introductions, in which we provided crowdsourcing definitions and related key concepts, we discussed crowdsourcing-based solutions to the use cases detailed in deliverables D2.2 and D1.1. Some solutions are already clear and well defined and their implementation has already begun. Others are still in a preliminary phase, and since their implementation depends on outputs of other project components, these need to be explored further in the coming months.

REFERENCES

Boulos, 2011	Boulos, Maged N. Kamel, Bernd Resch, David N. Crowley, John G. Breslin, Gunho Sohn, Russ Burtner, William A. Pike, Eduardo Jezierski, and Kuo-Yu Slayer Chuang. "Crowdsourcing, citizen sensing and sensor web technologies for public and environmental health surveillance and crisis management: trends, OGC standards and application examples." *International journal of health geographics* 10, no. 1 (2011): 67.
Briscoe, 2014	Briscoe, Gerard. "Digital innovation: The hackathon phenomenon." (2014).
Chandler, 2013	Chandler, Dana, and Kapelner, Adam. "Breaking monotony with meaning: Motivation in crowdsourcing markets." *Journal of Economic Behavior & Organization* 90 (2013): 123-133.
Deterding, 2011	Deterding, Sebastian; Dixon, Dan; Khaled, Rilla and Nacke, Lennart, “From Game Design Elements to Gamefulness: Defining “Gamification”” In *MindTrek’11*, 2011
Deterding, 2012	Deterding, Sebastian, “Gamification: Designing for Motivation” In *ACM Interactions*, 19 (4), pp 14-17, 2012
Demartini, 2012	Demartini, Gianluca, Djellel Eddine Difallah, and Cudré-Mauroux, Philippe . "ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking." In *Proceedings of the 21st international conference on World Wide Web*, pp. 469-478. ACM, 2012.
Demartini, 2013	Demartini, Gianluca, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. "Large-scale linked data integration using probabilistic reasoning and crowdsourcing." *The VLDB Journal* 22, no. 5 (2013): 665-687.
Ebner, 2009	Ebner, Winfried, Jan Marco Leimeister, and Helmut Krcmar. "Community engineering for innovations: the ideas competition as a method to nurture a virtual community for innovations." *R&d Management* 39, no. 4 (2009): 342-356.
Estellés, 2012	Estellés-Arolas, Enrique, and Fernando González-Ladrón-de-Guevara. "Towards an integrated crowdsourcing definition." *Journal of Information science*

38, no. 2 (2012): 189-200.

Feyisetan, 2015 Feyisetan, Oluwaseyi, Simperl, Elena, Van Kleek, Max, and Shadbolt, Nigel. “Improving Paid Microtasks through Gamification and Adaptive Furtherance Incentives”. *Proceedings of the 24th International Conference on World Wide Web*, ACM, 2015

Gabrys, 2014 Gabrys, Jennifer. "Programming environments: environmentality and citizen sensing in the smart city." *Environment and Planning D: Society and Space* 32, no. 1 (2014): 30-48.

Gadiraju, 2014 Gadiraju, Ujwal, Ricardo Kawase, and Stefan Dietze. "A taxonomy of microtasks on the web." In *Proceedings of the 25th ACM conference on Hypertext and social media*, pp. 218-223. ACM, 2014.

Hamari, 2014 Hamari, Juho, Jonna Koivisto, and Harri Sarsa. "Does gamification work?--a literature review of empirical studies on gamification." In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pp. 3025-3034. IEEE, 2014.

Howe, 2006 Howe, Jeff. "The rise of crowdsourcing." *Wired magazine* 14, no. 6 (2006): 1-4.

Hossain, 2012 Hossain, Mokter. "Users' motivation to participate in online crowdsourcing platforms." In Innovation Management and Technology Research (ICIMTR), 2012 International Conference on, pp. 310-315. IEEE, 2012.

Huff, 2015 Huff, Connor, and Dustin Tingley. "“Who are these people?” Evaluating the demographic characteristics and political preferences of MTurk survey respondents." *Research & Politics* 2, no. 3 (2015): 2053168015604648.

Kaufmann, 2011 Kaufmann, Nicolas, Thimo Schulze, and Daniel Veit. "More than fun and money. Worker Motivation in Crowdsourcing-A Study on Mechanical Turk." In *AMCIS*, vol. 11, no. 2011, pp. 1-11. 2011.

Law, 2007 Law, Edith LM, Luis Von Ahn, Roger B. Dannenberg, and Mike Crawford. "TagATune: A Game for Music and Sound Annotation." In ISMIR, vol. 3, p. 2. 2007.

Lease, 2011 Lease, Matthew. "On quality control and machine learning in crowdsourcing." Human Computation 11, no. 11 (2011).

Leimeister, 2009	Leimeister, Jan Marco, Michael Huber, Ulrich Bretschneider, and Helmut Krcmar. "Leveraging crowdsourcing: activation-supporting components for IT-based ideas competition." *Journal of management information systems* 26, no. 1 (2009): 197-224.
Malone, 2010	Malone, Thomas W., Robert Laubacher, and Chrysanthos Dellarocas. "The collective intelligence genome." *MIT Sloan Management Review* 51, no. 3 (2010): 21.
Morschheuser, 2016	Morschheuser, Benedikt, Juho Hamari, and Jonna Koivisto. "Gamification in crowdsourcing: a review." In *System Sciences (HICSS), 2016 49th Hawaii International Conference on*, pp. 4375-4384. IEEE, 2016.
Parent, 2010	Parent, Gabriel, and Maxine Eskenazi. "Toward better crowdsourced transcription: Transcription of a year of the let's go bus information system data." In *Spoken Language Technology Workshop (SLT), 2010 IEEE*, pp. 312-317. IEEE, 2010.
Piller, 2006	Piller, Frank T., and Dominik Walcher. "Toolkits for idea competitions: a novel method to integrate users in new product development." *R&d Management* 36, no. 3 (2006): 307-318.
Quinn, 2011	Quinn, Alexander J., and Bederson B. Benjamin. "Human computation: a survey and taxonomy of a growing field." In Proceedings of the SIGCHI conference on human factors in computing systems, pp. 1403-1412. ACM, 2011.
Ryan, 2000	Ryan, Richard M., and Edward L. Deci. "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being." *American psychologist* 55, no. 1 (2000): 68.
Sheth, 2009	Sheth, Amit. "Citizen sensing, social signals, and enriching human experience." *IEEE Internet Computing* 13, no. 4 (2009).
Smart, 2014a	Smart, Paul, Elena Simperl, and Nigel Shadbolt. "A taxonomic framework for social machines." In *Social Collective Intelligence*, pp. 51-85. Springer International Publishing, 2014.
Smart, 2014b	Smart, Paul R., and Nigel R. Shadbolt. "Social machines." (2014): 6855-6862.

Surowiecki, 2004	Surowiecki, James. "The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business." *Economies, Societies and Nations* 296 (2004).
Trainer, 2016	Trainer, Erik H., Arun Kalyanasundaram, Chalalai Chaihirunkarn, and James D. Herbsleb. "How to hackathon: Socio-technical tradeoffs in brief, intensive collocation." In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pp. 1118-1130. ACM, 2016.
Walcher, 2007	Walcher, Dominik. *Der Ideenwettbewerb als Methode der aktiven Kundenintegration*. Springer Fachmedien, 2007.
Wang, 2012	*Wang, Jiannan, Tim Kraska, Michael J. Franklin, and Jianhua Feng. "Crowder: Crowdsourcing entity resolution." Proceedings of the VLDB Endowment 5, no. 11 (2012): 1483-1494.*
Von Ahn, 2004	*Von Ahn, Luis, and Laura Dabbish. "Labeling images with a computer game." In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 319-326. ACM, 2004.*
Von Ahn, 2006a	*Von Ahn, Luis. "Games with a purpose." Computer 39, no. 6 (2006): 92-94.*
Von Ahn, 2006b	*Von Ahn, Luis, Shiry Ginosar, Mihir Kedia, Ruoran Liu, and Manuel Blum. "Improving accessibility of the web with a computer game." In Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 79-82. ACM, 2006.*
Von Ahn, 2006c	*Von Ahn, Luis, Mihir Kedia, and Manuel Blum. "Verbosity: a game for collecting common-sense facts." In Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 75-78. ACM, 2006.*
Von Ahn, 2008	*Von Ahn, Luis. "Human computation." In Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, pp. 1-2. IEEE, 2008.*
Vondrick, 2010	*Vondrick, Carl, Deva Ramanan, and Donald Patterson. "Efficiently scaling up video annotation with crowdsourced marketplaces." Computer Vision–ECCV 2010 (2010): 610-623.*
Vukovic, 2009	Vukovic, Maja. "Crowdsourcing for enterprises." In

Services-I, 2009 World Conference on, pp. 686-692. IEEE, 2009.

Yan, 2013 Yan, Zhixian, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. "Semantic trajectories: Mobility data computation and annotation." *ACM Transactions on Intelligent Systems and Technology (TIST)* 4, no. 3 (2013): 49.

Zaidan, 2011 Zaidan, Omar F., and Chris Callison-Burch. "Crowdsourcing translation: Professional quality from non-professionals." In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 1220-1229. Association for Computational Linguistics, 2011.