



Project no. 610349

D-CENT

Decentralised Citizens ENgagement Technologies

Specific Targeted Research Project
Collective Awareness Platforms

D4.3 Technical Design of Open Social Web for Crowdsourced Democracy

Version Number: I

Lead beneficiary: OKF

Due Date: 31 October 2014

Author(s): Pablo Aragón, Francesca Bria, Primavera de Filippi, Harry Halpin, Jaakko Korhonen, David Laniado, Smári McCarthy, Javier Toret Medina, Sander van der Waal

Editors and reviewers: Robert Bjarnason, Joonas Pekkanen, Denis Roio, Guido Vilariño

Dissemination level:		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Approved by: Francesca Bria

Date: 31 October 2014

This report is currently awaiting approval from the EC and cannot be not considered to be a final version.

Contents

I Executive Summary	6
Description of the D-CENT Open Democracy pilots	8
Description of the lean development process	10
Hypotheses statements	10
Lean canvases	10
Active Pilot experiments and ongoing iteration	11
Finland	12
The Finnish Context	12
Open Ministry - Crowdsourcing Tools for Citizens Initiatives	13
Helsinki City - Grassroots Action to Helsinki City Decisions	14
Helsinki City Decisions Newsfeed	15
Helsinki City - Public Consultation Service	19
Spain	21
Collective decision-making of citizens movements	21
Collaboration experiments with Podemos and Guanyem	23
Collaboration experiment with Podemos CIENCIA	23
Collaboration experiment with Podemos I+D+i	25
Collaboration with Guanyem Barcelona	28
Specifications of the Spanish pilot for Guanyem Barcelona	28
Iceland	30
Bottom-up Municipal Democracy	30
Better Reykjavík	30
Better Neighborhoods	31
Better Iceland	32
The Lean Process	33
Open Source software used in the Icelandic pilots	36
D-CENT Front-End	37
Design Principles	37
Front-end functionalities	40
Login Page	40
Navigation menu	40

	VI
Context Help	41
Message Windows	41
Notifications.....	42
Open Decisions.....	43
ActivityStream Newsfeed	43
Mobile view	45
Visualized calendar timeline.....	45
Map Mode.....	45
Content.....	47
Articles	47
Buttons	48
Discussion.....	48
Social media sharing	48
Agreeing on Comments	49
Likes	49
Groups	50
Voting	50
Alternate Use Cases	52
Participatory budgeting.....	52
Collaborative bottom-up editing.....	54
Annotation.....	55
Tasks	56
Events	57
User Settings Page	60
Front-end State diagram.....	62
Architecture.....	64
Architectural principles.....	64
Modular, exchangeable building blocks.....	64
Technical feedback.....	64
Agile	64
Reusable and refactored.....	65
Value frameworks	65
Assure user control over personal and social data	65
D-CENT Back-end Features	66

	VI
D-CENT Core Features	67
Front-end	67
Back-end.....	67
D-CENT Applications	68
Docker Back-end and App Directory	68
D-CENT Core Features.....	70
1. Social Data Store	70
2. Cryptographic components.....	75
Introduction to cryptography	75
3. Strong authentication and Single-Sign On	78
4. Identity Management.....	81
5. Groups and Access Control.....	85
6. Notification Engine	89
7. Voting & Deliberation	94
1. Voting systems	94
2. Balloting methods.....	94
3. Tallying methods.....	97
4. Vote Delegation.....	101
5. Voting process and the block chain	104
6. Deliberation systems	106
7. Technical Specifications.....	108
8. Data Portability	113
9. Federation	115
10. Secure Messaging	117
11. Standardization plan	119
D-CENT Applications	123
App Directory	124
Open Data	125
Crowdsourcing	127
Background.....	127
Developer Community Engagement Methodology.....	130
Engaging Government Entities	131
Hackathons.....	132
User Groups	132

FP7 – CAPS - 2013 D-CENT D4.3 Technical Design of Open Social Web for Crowdsourced Democracy	VI
Training Materials and Master Course.....	132
Developer Support and Product Management	132
References	134

1 Executive Summary

The overall objective of Work Package 4 is to design the technical specifications for a standards-based, privacy-aware and decentralized D-CENT platform for open democracy. In this deliverable, we present here the technical architecture of the D-CENT nodes, each with its own social data store, in order to allow D-CENT to be used for the direct democratic decision-making pilots.

In particular we will describe: (i) **the federated and privacy-aware social networking architecture** for the core D-CENT platform that allows communities to own their social data (“data portability”), and crowd-source a data-driven “map” of their social relationships and environment (ii) **design of participatory democratic decision-making modules** (which include publishing articles and contents, collaborative text writing for instance in crowdsourced legislation, share annotations, debate and voting) for D-CENT that enables these communities to self-organise and make decisions, keeping a collective memory of their activity. The capabilities needed by each pilot will be mapped to a set of requirements that result in a gap-analysis of the core social-networking codebase of D-CENT. A number of core features, such as easy-to-use group access control, collaborative writing and voting mechanisms, and multi-media objects will be factored into features to be added to the D-CENT platform. Equally importantly, the integration of the crowd-sourcing PyBossa platform and the open social datastore (with export to RDF and the ability to use CKAN for human-produced metadata about data-stores) will be included as part of the work led by the OKFN as D-CENT enabled applications. Advanced capacities such as the ability for communities to discover resources from sensor data-streams and mapping data other and resources may also become requirements coming from the use cases.

Each of the direct democratic decision-making tools will be analysed for functionality. This functionality will include posting new Articles, comments and annotations on articles based on the Democracy OS codebase, discussion and deliberation; Voting on a text with series of options, including weighted voting and blockchain voting; Collaborative bottom-up editing; Notification engine with users preferences (Sending reminders out and decision results using Twitter). Core D-CENT features will be also included such as strong authentication; Single sign-on; Identity Management; Access Control for Groups, Secure Messaging, and the implementation of standards led by the W3C Social Web Working Group such as ActivityStreams, Federation, and Data Portability. Each pilot will run whatever tools, possibly different, necessary to solve their local problem from a combination of D-CENT core features and D-CENT application.

The output of this deliverable is the production a technical design for implementers that is strongly linked to the social requirements coming from the pilots (see D1.2 and D1.2 for a detailed description of our lean UX development methodology). This Deliverable will outline the technical details of the crowd-sourcing, open-data, and democratic decision-making tools capabilities of the federated social networking platform. eco-system that could then be generalized to address needs throughout Europe and beyond. The inputs for the technical design come from the ongoing and iterative social design that is happening with the D-CENT communities on the ground. This results in diverse pilots and experiments that constitute an open decentralised democracy ecosystem that will communicate via standards (as

outlined in D4.1 and D 4.2) and factor out, from each successful application, a common decentralized social data platform for democracy, the D-CENT platform, whose open-source components can then be shared and build future applications. This modular, open and standard-based characteristic of the D-CENT platform will make it easier to integrate in the future the digital social currency design based on the Bitcoin block-chain (D4.4) and its implementation (D 5.5) of the second pilots through JSON API.

Following the “lean” process outlined in D1.1 and D1.2, the ultimate target of the D-CENT development process to build software that users actually want, while taking into account what technical aspects of current applications are currently addressing real social needs, as well as a “gap analysis” of where existing solutions fall short. Furthermore, across all three pilots common needs will be taken into account, as well as fundamental design principles around data protection, security, open source, and decentralization. Thus, what is necessary is given the social requirements given by each of the pilots and explained in detail in D1.2, to essentially “map” these social requirements to technical features that we believe may *fulfil* the needs of users. Of course, through experimentation it may be possible that these needs are not actually fulfilled by the technical features, and thus further iteration is required.

Just like the technical recommendations in D4.1 and D 4.2, the recommendations in this deliverable are non-binding, but nonetheless provide a valuable map with dependencies and open decisions to help coders navigate the features needed for the decentralized social networking and direct democracy D-CENT applications. The features here described will be then prioritised within WP 5 (D5.1: D5.2: D5.3) and each application of the D-CENT platform will then be built integrating users feedback and requirements coming from the pilots during the testing phases.

Description of the D-CENT Open Democracy pilots

Across Europe, attempts to engage citizens and social movements in democratic decision-making and collaboration using digital platforms are still in their early stages. Most of these platforms lack features and have complex user-interfaces, which might leave many people unable to meaningfully participate in the democratic process via the Internet. A few existing platforms have been specifically designed to engage users into large-scale Internet-based democratic process that goes beyond the limits of traditional corporate social media. There are some bright spots, such as the large-scale usage in Iceland of Your Priorities for over four years. Yet, most of these initiatives did not succeed in supporting the process of large-scale collective action and participation by different communities. For more rich information about the different needs of the communities in the pilots, see Deliverables in Workpackage 1 and Workpackage 2.

D-CENT is conducting pilots across Europe to accelerate the development of distributed alternatives to online, democratic voting and deliberation. The goal is to develop a framework for the deployment of decentralized social networks for community-driven democracy which are both easy to use and properly aligned with citizen motivations and digital rights. Some of the features will be specifically designed to link into existing formal structures of democratic power; other will purport to build the capacity for the deployment of new democratic institutions, harnessing the network effects of digital tools and real-time collaboration to solve real citizens' problems.

On the theme of usability it may be mentioned the standing point that people with different abilities have when confronted with graphical user interfaces. Minorities are often excluded by the lack of accessibility of certain tools and we could generically address the need to think of graphical user interfaces that bridge the gap of usability in case of different abilities.

As of my perception of current design guidelines for GUI components in d-cent this is a problematic that can be well solved as the design is well accessible. Furthermore this could lead to add recommendations for developers to check already in an early stage the easy conversion to text-only format (for blind people) and such possible extensions.

I'm not sure the analysed use-cases ever included such conditions but I believe it wouldn't hurt to mention them at least as a possibility since they can be easily realized.

When the term “we” is used in the text, it refers to the D-CENT project members involved in lean design within these pilots.

The D-CENT project has been engaging with various grassroots civil society organisations, groups and communities in Spain, Iceland and Finland, in order to identify their needs with regard to the adoption of

new tools for democratic engagement (see Fig. 1). Specifically, the democracy pilots are meant to bring together leading European examples of collective deliberation and decision-making in Spain, Finland and Iceland, so that they can learn from each and, ideally, find possible synergies and paths for future collaboration and cooperation.

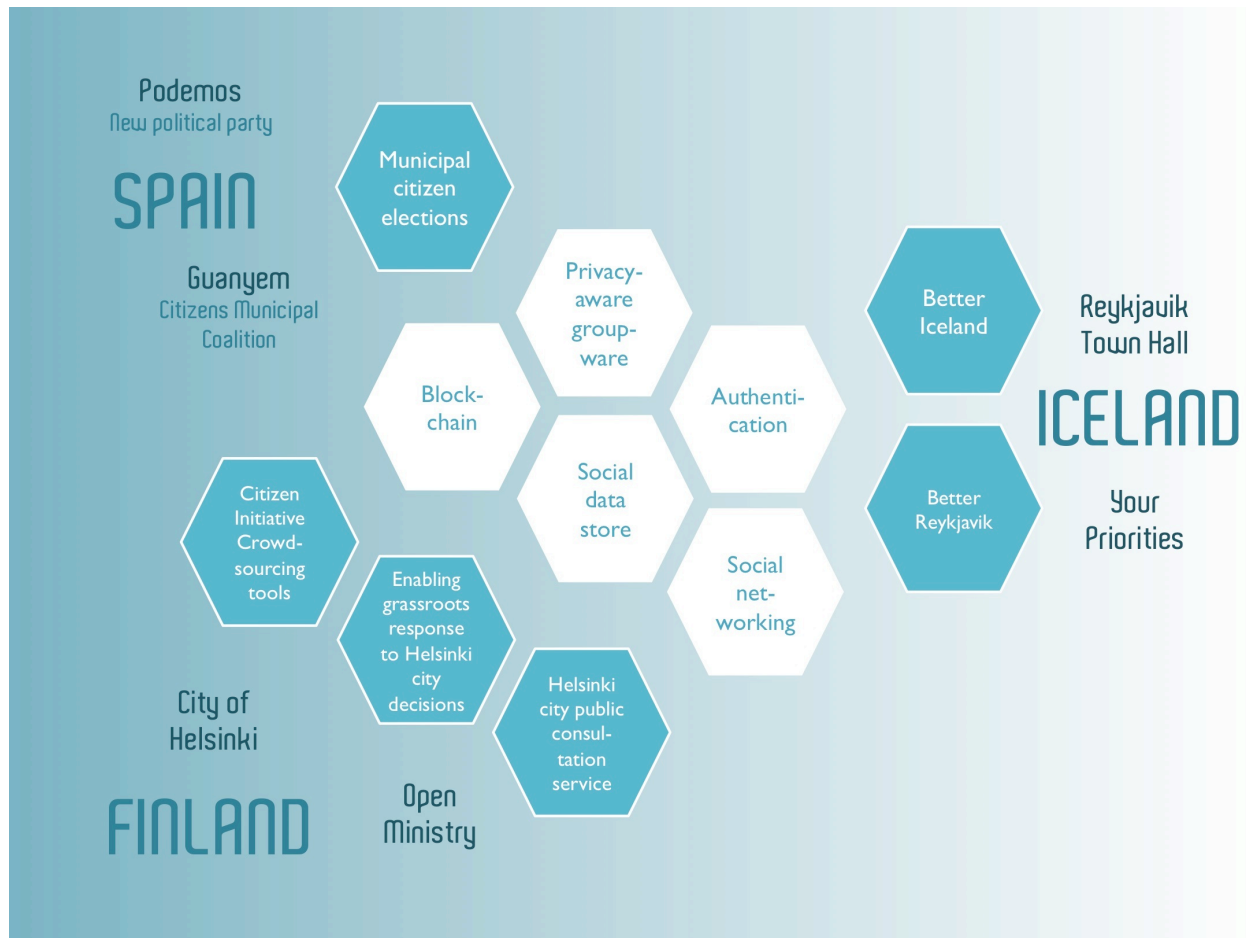


Figure 1 D-CENT Pilots Ecosystem

Description of the lean development process

As detailed in D1.1 and D1.2 the D-CENT project employs a combination of lean development and action research methodologies that engage users communities since the inception stage. From January to March 2014 we have been running lean inception workshops in the three pilot countries (Spain, Finland and Iceland) consisting of user interviews, persona development and hypothesis testing in order to identity the first D-CENT Minimal Viable Products (MVPs) to be developed and tested within a process of fast prototyping and iteration. Rather than focusing on features, we first focused on the main problems that users were encountering, and then to draft a series of hypothesis statements based on the interviews as a method to identify underlying assumptions about users' needs and solutions to be tested in the field. The hypotheses discovered in the lean inception workshops are considered the social requirements of D-CENT, to be developed by creating lean canvases for specific user-groups and needs, and then translated into concrete technical features and Minimal Viable Products (MVPs), to be tested (for a detailed description of the process followed see D2.1). The technical requirements that derived from this Lean UX process are the software features that will be detailed in this Deliverable.

Lean is a cyclical and on-going process. The technical requirements outlined in this document are based on the social requirements that are continuously gathered during the duration of the project in WP1, and they represent a first iteration of the overall requirements and features of the D-CENT platform.

Hypotheses statements

In the initial round of testing, hypotheses have been drafted for each user group in each country. Hypotheses are quickly drafted only to serve the purpose of turning the interviews with users into potential solutions and features, and should therefore be easily discarded if the assumptions are not verified when tested in the field. Through workshops sessions with local partners and stakeholders at during the lean inception workshop, and through follow-up work with bottom-up communities the D-CENT team arrived at a set of hypotheses to be developed into technical features and lean-canvases, outlined in the next pilot section of this document (See D1.2 for a comprehensive list of hypothesis tested during the first experiments).

Lean canvases

When a hypothesis has been selected for further development, a “product” is brainstormed and a set of features are noted down that are assumed to address the users stated needs. A lean canvas is then drafted for the “product” in order to tease out further assumptions underpinning the “product” and any potential feature related to its viability in the field as well as key metrics to measure its success. The lean canvas thus serves as the first step to begin outlining which tests need to be conducted via MVPs to validate assumptions and ensure that the “product” is addressing the concrete needs of users. As this is a cyclical process to take place on a regular basis during the development stage, the lean canvases are drafted online, allowing for distributed collaboration across pilot countries and between partners. Using

the LeanStack services, 2-4 canvases have been drafted for each pilot, with associated experiments being defined (Lean canvases and experiments are documented in D1.2).

Active Pilot experiments and ongoing iteration

A series of MVPs have been developed and tested within each D-CENT Pilot country. An MVP (Minimal Viable Product) can be any type of experiment that will feed back information needed in order to validate a hypothesis. In the first phase of the development of the D-CENT applications, this will take the shape of functional prototypes that can bring immediate value to users on the field. The three main questions that need to be asked are thus Is there a need for the solution I am designing? Is there values in the solution and features I am offering? Is my solution usable? (Gothelf and Seiden, 2013). Tests and experiments are taking place on an ongoing basis and are shared and monitored by the D-CENT consortium using a personalised LeanStack dashboard used as a tool for the D-CENT consortium to share experiment results across the three pilots and aggregate users feedback in real time. This section describes the first pilot tests in Spain and Finland, and reports some metrics recorded during the tests following an iterative process. We start the next section by summarising the main first pilots we are testing in the D-CENT project (see Fig. 2).

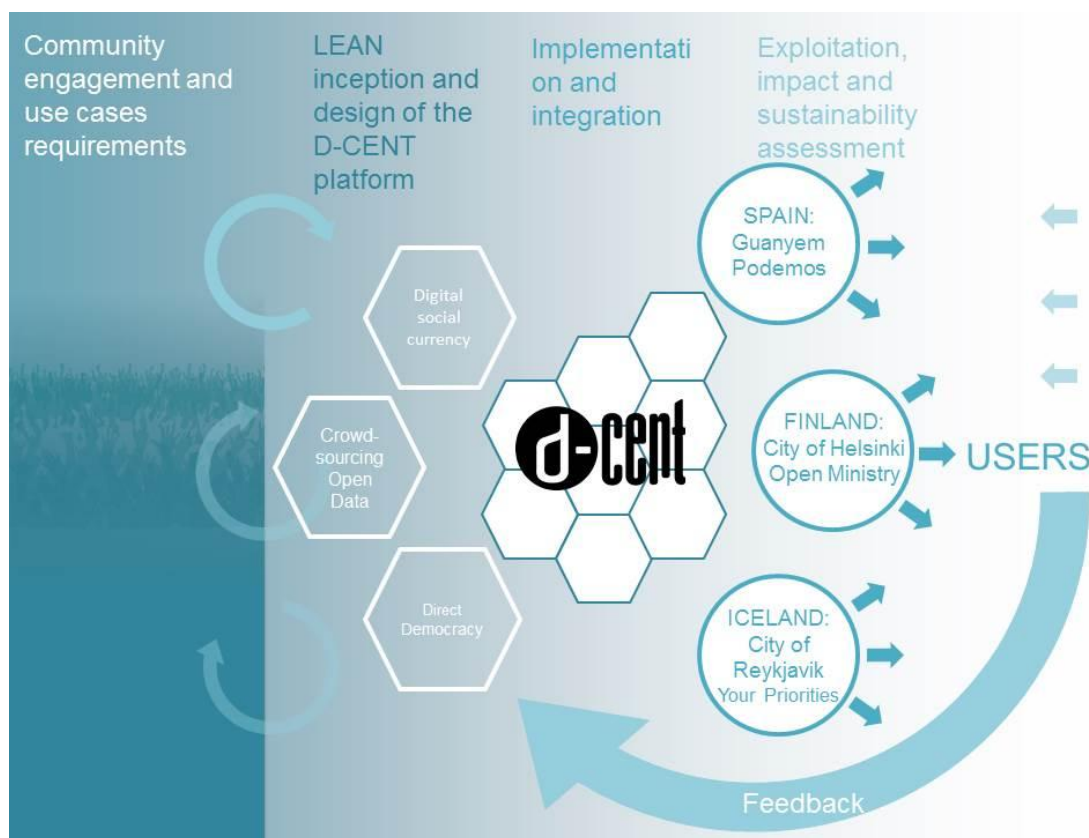


Figure 2 D-CENT lean development process

Finland

The Finnish pilots revolve around enabling citizens to engage into collective action with a grassroots, mostly bottom-up approach to decision-making. In this regard, the initiatives that have been selected as pilots are both intended to support policy change at either the municipal or national level.

The Finnish Context

The City of Helsinki has adopted a more open and citizen-centric approach under the leadership of the new Mayor, Mr Jussi Pajunen. The city has opened up its internal document management system (Ahjo) and has released all the agendas and decision items of the city council and the city's subcommittees as Open Data available through a JSON API (OpenAhjo).

Forum Virium Helsinki (FVH) is a company entirely owned by the City of Helsinki. FVH has been instrumental in facilitating the Open data and smart city development in Helsinki. The online Open data portal, [Helsinki Region Infoshare](#), started as a part of FVH, has now 1124 open datasets.

The open data and open participation agenda is now gaining momentum across the country. The six largest cities have started an 80 million euro project, the [Six City Strategy](#), jointly funded by the European Social and Structural Funds. The objective is to improve the services offered by these six cities so as to enable more widespread citizen participation. In this regard, the accessibility, effectiveness and productivity of services will be improved through the development of online service that will be deployed around three main focus areas: *open innovation*, *open data*, and *open participation*. The idea is to view cities as platforms, whose operations and services should be developed in ways that enable participation by third parties. Cooperation requires cities to open up their data, but also to produce tools that facilitate collaborative processes and joint development. In this regard, the Six City Strategy project offers important opportunities to leverage the D-Cent platform in a variety of situations oriented towards citizen participation and empowerment.

At the same time, at the national level, the government of Finland is also starting to embrace the principles of open data and open government. In 2012, Finland became the first country to give *full parliamentary process* to crowd-sourced law proposals proposed by citizens and civil society organisations, (CSO) as long as they would succeed in collecting at least 50 000 supporters online. "Full process" means that the parliament will process any crowd-sourced law proposals put forth by citizens through the exact same process as bills issued by the government. This process was recently enshrined within the Constitution, and it has now been translated into a right granted to every Finnish citizen.

Open Ministry is a civil society organization established to promote citizen's participation and civic empowerment. It supports citizens and other CSOs in setting up Citizens Initiative campaigns and advocate for the fair and transparent treatment of these campaigns within the Finnish parliament. Open Ministry has also been engaged for the past two years in educating citizens with the use of different ad

hoc online tools for collaborative work, co-editing tools for crowdsourcing law proposal, task management tools for community management, etc.

There is, today, a strong political buy-in for increasing citizen participation and open data. In September 2014, the Prime Minister's office organized a large two days event, *Open Finland 2014*, to underline the importance of Open Data and Open Government. It was organized together with ministries and civil society organizations like Open Ministry and Open Knowledge Finland. Today, many civil society actors are working closely together with the government and civil servants to implement the principles of open data and open government.

Open Ministry - Crowdsourcing Tools for Citizens Initiatives

Open Ministry will serve as a pilot to identify the minimum requirements to allow for an optimal workflow for the Citizens Initiative (CI) campaign it supports. Citizens involved in preparing and conducting CI campaigns need an effective way to organise their time and the time of others. They also need to understand the extent to which their efforts are being valued by the community.

This pilot will enable citizens to:

- Create projects and interest groups around certain thematic areas (e.g. ideas for national level law amendments in the Open Ministry context; or municipal suggestions on how to improve the neighborhood or the municipality in the Helka context).
- Invite friends and share the project in social media
- Create co-edited and co-annotated documents (such as press releases, FOI requests, law proposals)
- Create and assign tasks among group members (i.e. for the campaign core group to manage their internal work flow)

The pilot will result into a Minimum Viable Product (MVP) that will be jointly carried out by Citizens Initiative campaigners (coordinated through Open Ministry), neighborhood activists (coordinated through Helka), youth activists (coordinated through Nuorisoiäskeskus, the youth board in Helsinki), and a number of other user groups, including special interest groups.

The use case for these different user groups is very similar. They all need tools to establish a particular community of interest (i.e. find like-minded individual and gather them together within a group) and self-organize so as to take effective collective action. This include actions such as, *inter alia*, drafting a press release, sending emails to the city council, arranging a demonstration, planning a Citizen's Initiative at the national or municipal level. The pilot will consist mostly of functionalities that have already been planned to be the core functionalities of D-CENT. Figure 3 is a graphic from Open Ministry that illustrates the process.

The Helka's participation model (*paikallinen kehittämisspolku*) is designed to facilitate the way residents, local businesses and city representatives collaborate to improve things on the local neighborhood level. Local hubs are already active - to a varying degree - in various neighborhoods of Helsinki. So most of the

activities have been undertaken offline, through physical workshops and meetings. We believe these local hubs need an online tool to complement their offline activities.

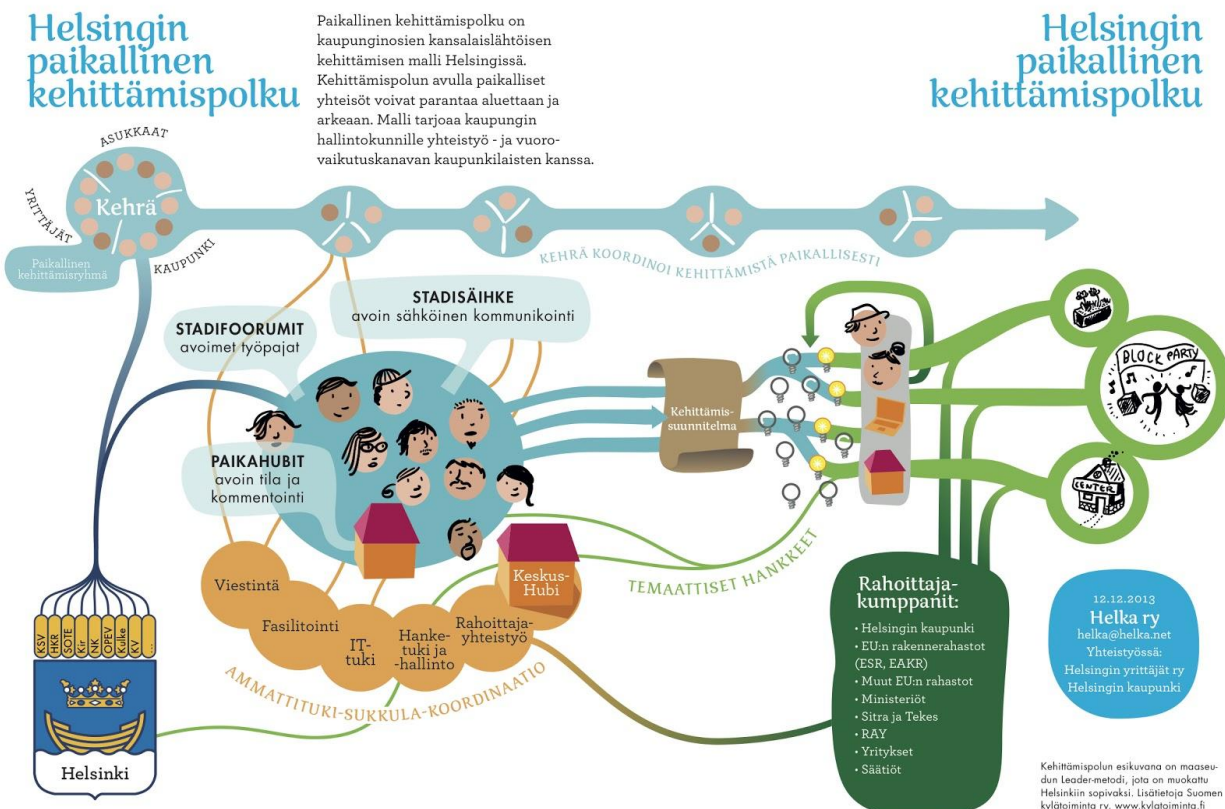


Figure 3 The Neighbourhood Collaboration Path of Open Ministry

Helsinki City - Grassroots Action to Helsinki City Decisions

This pilot is designed to support the inhabitants of Helsinki to quickly engage into effective collective action to respond to the issues that are being processed in the Helsinki City's decision making bodies, including the City Board, the City Council, the City Committees and their boards (see Figure 4 below). Already all items on the agenda of public bodies (as well as all the decisions of public officers since 2014) are available as Open Data in an award winning API provided by the City of Helsinki.

Committees and boards

The agendas and decisions of committee meetings are publicly available. The meetings, however, are closed to the general public.

Audit Committee
Board of the Commercial Services municipal corporation Palmia
Board of the Finnish-language Workers' College
Board of the Helsingin Energia municipal corporation
Board of the Helsinki Region Transport municipal corporation
Board of the Payment Management Services municipal corporation
Board of the Personnel Development Services municipal corporation (Oiva Academy)
Board of the Port of Helsinki municipal corporation
Board of the Swedish-language Workers' College
City Art Museum Board
City Museum Board
City Planning Committee
Committee of Early Education and Care
Construction Committee
Cultural and Library Committee
Education Committee
Environment Committee
Helsinki Philharmonic Orchestra Board
Housing Committee
Housing Production Committee
Metropolilab Board
Municipal Election Central Committee
Personnel Centre
Public Works Committee
Real Estate Committee
Rescue Committee
Social Services And Health Care Committee
Sports Committee
Technical Services Committee (Procurement Centre, Construction Services and Wholesale Food Market)
Youth Committee
Zoo Board

Figure 4 The City Decision-Making Bodies in OpenAhjo

Helsinki City Decisions Newsfeed¹

Helsinki City Decisions Newsfeed is a tool designed to help Helsinki residents to find out what decision are being made in the City Board, City Council and all the subcommittees.

The pilot will result into a Minimum Viable Product (MVP) that will enable citizens to take collective action based on the feed of the City's official decision making process. Users can subscribe to alerts on issues that they could be interested in (such as, for instance, those related to their neighborhood, or those concerning bicycling and dog parks). Anyone can "flag" important issues by creating a "project" on the issue, and then share the project on social media or invite people to collaborate. The issues that spurred the most interest can be shown with a higher priority on the activity stream for selected users. This allows for the continuous flow of decisions from different municipality's decision-making bodies to be effectively monitored through crowd-sourcing.

¹ <http://dev.hel.fi/paatokset/>

The D-CENT pilot will allow people to react to the relevant issues. Through this tool, citizens can request to be notified of various issues of interest whenever they appear on the agendas of the city council, or municipal subcommittees. Citizens can then alert others and/or form groups to react to those issues, as well as other issues which have been raised through other routes. The Open Ministry tool allow groups to co-edit documents (e.g. draft emails to council representatives, civil servants or media), to assign tasks and to organize actions like demonstrations when necessary. The key partner for this pilot on the local level is **Helka**, which coordinates the operations of several neighbourhood associations around Helsinki, but many other special interest groups (e.g. bicycle associations, parents associations, etc) also constitute important early adopters of the tool.

The key partner for this pilot is **Open Ministry**, a non-profit organisation with an online platform that supports individual citizens and organizations to gather together for co-designing Citizens Initiative campaigns and for crowd-sourcing law proposals. In Finland, Citizens Initiatives on the national level can be submitted to the Parliament if they reach 50, 000 supporters. Similarly, in Helsinki, the City Council is bound by law to consider Municipal Citizens Initiatives if they reach ca. 10,000 supporters. With 25,000 supporters, the municipal initiatives can arrange a local referendum.

User Story

Before and during their meetings, the citizens of Helsinki need tools to spread information about the municipal council agenda, so as to make it more understandable to all and eventually be able to take better-informed collective action.

The MVP should be able to support the following scenarios:

- (1) On “My Page”, users can subscribe to specific keywords concerning their own neighborhoods or topics of interests. These keywords are generated by the users themselves as hash-tags and not pregenerated. For example #bicycling #kallio, etc.
- (2) Users can follow the activity stream related to these keywords.
- (3) Users can create groups (or "projects") by defining specific activity stream filters and applying a series of action / algorithm to the stream. They can also invite people to the group, and set specific privacy settings for their group, e.g. "private" or "public" (although the version of the software will only support "public" projects)
- (4) Users can visualize the discussion surrounding the issues they subscribed to, and are given the possibility to comment on shared articles. There will also be a set of questions which can be answered on a form.

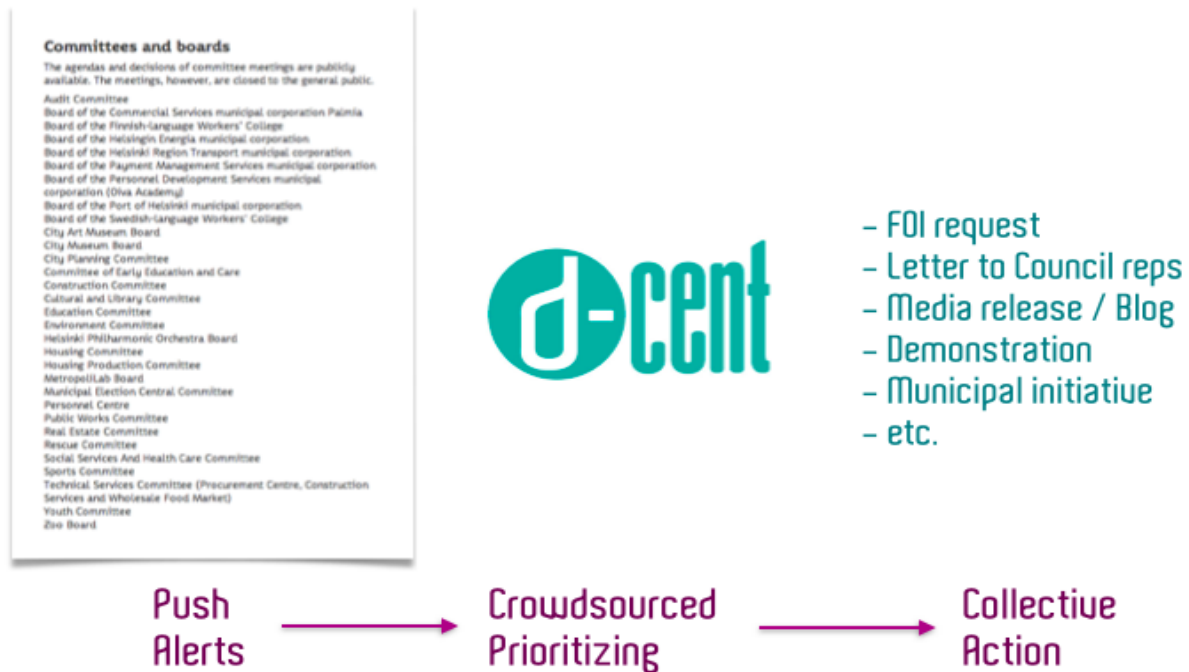


Figure 5 Crowdsourcing Collective Action based on City Decisions

Alternative use cases

- (1) Users are shown a discussion concerning a particular issue and are given the possibility to share links and/or collaboratively produce and edit articles concerning that issue. This can be implemented, for instance, through the use of an Etherpad or a more integrated functionality module within the D-CENT platform. This feature is necessary to enable citizens to collaborate in:
 - Co-editing an email to be sent to all council representatives regarding the issue at stake.
 - Co-editing a press release or a blog post on the issue
 - Coordinating a Citizens Initiative at the municipal or national level (including a law proposal)
 - Writing a plan for a demonstration on that particular issue
- (2) User can shares “projects” to Facebook & Twitter as providing by the sharing functionality of the respective social media applications.
- (3) Users are shown a discussion concerning a particular issue and are given the possibility to assign tasks to others. This can be useful for:
 - Assigning tasks to members of the group.
 - Drafting To-Do lists and specify tasks to be assigned or picked up by group members.
 - Set up a timeline view or a calendar view of different tasks.
- (4) Users are shown a discussion and are given the possibility to initiate polls.
- (5) User can share an article or link to the group and creates notifications for the other group members.

- (6) User can create tasks to mobilize people on addressing the issue.
- (7) Decisions API can be used to update the conversation with the decision result from the council.

Decisions API

OpenAhjo is an API used by the City of Helsinki that provides a common format (both XML and JSON-based) for decision-making data. This will be integrated tightly into D-CENT into the Helsinki pilot. See Deliverable I.2 for more information.

OpenAhjo API

API and UI for accessing the decision-making material of the City of Helsinki. The backend is RESTful and based on [Django](#) and [Tastypie](#). The HTML5 frontend utilizes the REST API and is written mostly in [CoffeeScript](#).

Background

The [City of Helsinki](#) has decided to open the data related to decision making inside the city. In the first phase, the focus is on political decision makers, e.g. committees, the city board and the city council. During the second phase (to be completed in 2013), data from all non-political civil servant decision makers will be opened, as well.

The original data is provided in zipped XML format. The data is available from the [OpenAhjo web server](#). The ZIP files are in subdirectories based on the committee name and ID. The ZIP file includes also other types of files such as the attachment documents (mostly PDF) to the agenda items.

This project automatically scans through the original repository periodically, imports the XML, cleans it up and augments the data with other information such as geospatial geometries. It then serves the content in linked data format over a REST API.

Figure 6 The OpenAhjo API²

² Source: <https://github.com/City-of-Helsinki/openahjo#openahjo-api>

issue

An issue being discussed and decided on at different committee meetings. One issue can be discussed in several different meetings. The resource `agenda_item` links issues to meetings.

`http://dev.hel.fi/openahjo/v1/issue/148/?format=json`

```
{
  "category": "/openahjo/v1/category/419/",
  "category_name": "Asemakaavoitus",
  "category_origin_id": "10 03 03",
  "geometries": [
    {
      "coordinates": [24.93393763640416, 60.16417504680238],
      "name": "Albertinkatu 27",
      "type": "Point"
    }
  ],
  "id": 148,
  "register_id": "HEL 2011-004632",
  "resource_uri": "/openahjo/v1/issue/148/",
  "slug": "hel-2011-004632",
  "subject": "Kampin tontin 72/2 asemakaavan muuttaminen (nro 12145; Albertinkatu 27",
  "summary": "Asemakaavan muutosehdotus mahdollistaa liikerakennusten korttelialueen"
}
```

Figure 7 An example open issue JSON data format from OpenAhjo³

Helsinki City - Public Consultation Service

This pilot represents a more traditional public consultation process in the online context. It is meant to create a Minimum Viable Product (MVP) to support the inhabitants of Helsinki in promoting cooperation with the City. To do so, the Helsinki City IT department will run a D-CENT node to facilitate the task for city officials to hear citizens and account for their comment with regard to the City's agenda and plans.

The main focus of this pilot relates to the *usability* of the service from the perspective of the civil servant. Namely, the goal is to support city officials in setting up online public consultations with as few clicks as possible. The main hypothesis is that, by making the process more intuitive, a greater number of civil servants will be inclined to set up such public consultations.

³ Source: <https://github.com/City-of-Helsinki/openahjo#openahjo-api>

Helsinki City will dedicate considerable effort in supporting the development of the tools and customise the User Experience to the specific needs of civil servants. The city will also ensure that all the requirements of the Finnish law are properly fulfilled with regard to citizen's involvement and participation in decisions concerning land use and city planning.

User Story

Civil servants want to get more citizens to participate in the drafting of the city's development plan. They also want to submit their questions for feedback.

Use Case

This pilot proposes to enable civil servants to set up a new online consultation in a few simple steps:

- Civil servants click on "Create new Article"
- Civil servants provide a diary of the various planning documents that the hearing is related to.
- Metadata about the documents and related attachments, including all city planning drawings, are fetched from the OpenAhjo Decision API.
- Civil servants edit the form; they can publish their own questions to the citizens.
- Civil servants save, publish and share the project to the public at large.

The customisation of this D-CENT node will be undertaken with the programming resources provided by the City IT department, with the D-CENT software team providing support and guidance. All pilot findings and results will be made available to the D-CENT project.

Spain

Collective decision-making of citizens movements

Context: Evolution of the 15M movement and emergence of new citizen options in the electoral and institutional arena

The historic event generated by the explosion of the 15M network movement, in May 2011, suddenly aged state institutions, forms of representative democracy, political parties and trade unions. The demand for real democracy crossed the whole Spanish society, with public support for the movement ranging from 65 to 75% of the population between 2011 and 2013. Three years later, a number of initiatives have emerged that try, in various ways, to bring many of the demands, practices, and spirit of the 15M network movement into the institutional and electoral arena.

One of 15M's main features is its ability to mutate and adapt to the needs of the moment. Initiatives in the 15M ecosystem have updated their forms of collective action at high speed; early ones stand as a contagious benchmark for those that follow them, such as the Valencian Spring, 12M15M (the first anniversary), 25S (a call to surround the Spanish Parliament) and many others in 2012. So the limits of previous stages are surpassed, and new ones are found, while practice remains the key method to overcome them. In the midst of this flux, the presence of common elements, replicated across stages, is noticeable.

The progressive drop in support for bipartisanship (together the two main national parties, PP and PSOE, would not reach 50% of the votes today, down from 80% before 15M), as well as in citizens' trust on institutions and existing parties, are the setting where a new social consensus has emerged. This consensus suggests that a citizen, social majority wants more and better democracy. In 2013, this desire jumped onto the institutional and electoral arena. Three years after the 15M "social tsunami", in the context of the 2014 European elections, this desire for a better, real democracy, entered the European Parliament with a new party: Podemos. This formation, launched four months before the elections (garnering many 15M demands) got 1,250,000 votes and broke the monopoly of institutional politics held by the major parties, PP and PSOE. In this context of Podemos' exponential growth (survey-polls position it as the third political force in Spain, near the second, PSOE) and of accelerated decay of the major parties — the ones sustaining the status quo in Spain since 1978 — the next milestone is the local election process coming in 2015. In their General Assembly this October 2014, Podemos has reached mass scale participation into their political process. Before the Assembly, debates and deliberations have been orchestrated in the party online platform (Plaza Podemos) built on Reddit with more than 600.000 monthly unique visits (<http://www.reddit.com/r/podemos>). More than 150.000 subscribed online to vote the key political documents, and over 150,000 were watching the online streaming of the Assembly.



Figure 8 Podemos Citizen Assembly

In May 2015, four years after the beginning of 15M, elections will be held in more than 8100 municipalities, and there already are a cadre of citizen convergence candidacies aiming to reach power at the local level all over Spain. Among them, Guanyem Barcelona is the most remarkable, although Municipalia in Madrid (Ganemos Madrid), Ganemos Sevilla in Seville, Ganemos Málaga in Malaga, Marea Atlántica Coruña (Ganemos Coruña) in La Coruña, Marea Atlántica Vigo (Ganemos Vigo) in Vigo and a long list are following suit. These citizen candidacies are becoming serious options for catalysing and canalising outrage and aspiration for better democracy into the 2015 local elections. This experience with citizen-lead municipalism improves the desire for democracy at the space nearest to the citizen: the local institutions. This is an excellent setting for combining innovations in direct and deliberative democracy essayed during the last 3 years, as well as for a citizen re-appropriation of public decision making on the city and city rights.

This is the context where the proposal for collaboration between D-CENT and Guanyem Barcelona (the reference project for municipalist candidacies) gains its meaning and full dimension. This collaboration may help to enable and adapt tools that those candidacies will need, to test new forms of citizen participation, deliberation, and cooperation in the coming wave of democratic bottom-up municipalism.

Those local processes need a toolkit that allows them to generate wide, transversal and transparent processes of social and digital participation. This toolkit should focus on processes that facilitate and validate the agreement between social and political forces.

Collaboration experiments with Podemos and Guanyem

D-CENT's collaboration with citizen collectives and initiatives has already started. So far we have worked with two thematic circles of Podemos, and started a collaboration with Guanyem Barcelona. In the following we explain both experiments.

Although the experience with the two circles of Podemos was positive, collaboration with Podemos at a larger scale is not possible for the moment and Podemos is using the commercial Reddit service along with Appgree, which is non-open source software, and Loomio.⁴ This is why the Spanish pilot will be focused mainly on the needs of Guanyem Barcelona.

Collaboration experiment with Podemos CIENCIA

Podemos Ciencia (Science) is one of the 42 circles listed in the official [web page](#) of Podemos. In its [Facebook fan page](#), created in May 2014, the group is defined as: “Podemos Ciencia is born to generate interest and participation about topics where science and politics converge (Twitter: @podemosciencia) If you want to participate in Podemos Ciencia circle, join our work groups in the following form: <http://goo.gl/V4y6Pc>.

In June 2014, Podemos Ciencia posted a [message](#) on the fan page to ask for virtual spaces where the participants of the circle would be able to hold an assembly. The message stated the following requirements: authentication process via Facebook or email, conversation threading and a voting system for the messages. After discussion, we agreed to build up a D-CENT [instance](#) for their assembly. Almost a hundred of citizens participated in, at least, one of the 19 discussion threads created by the administrators of the circle (see Figure 9). One of the threads was devised to provide feedback about the usage of the instance. 44 users took part of this thread through comments and/or votes (see Figure 10). 36 votes were counted: 33 positive votes (91.67%), 1 negative vote (2.78%) and 2 abstentions (5.56%). In the comments section, users spontaneously wrote suggestions with the hashtag #propuesta (#proposal). Some of them were the following:

- To create an index with the topics to be discussed.
- To build a section to publish the comments disapproved by the admins in order to enhance the transparency on the management of the platform.
- To set a link to the folder that contains the full documentation of the discussions.
- To enable email alerts to users when their comments are replied.
- Validate accounts through the electronic Spanish ID mechanism.

⁴ <http://www.newyorker.com/tech/elements/spain-politics-via-reddit>

Title of the discussion	Participants
¿Aceptas este Acuerdo de mínimos ?	88
¿Quieres un Grupo de Traducción y Divulgación ?	74
¿Quieres un Grupo de Informes y propuestas ?	73
¿Quieres un Grupo de Debates ?	72
¿Quieres un Grupo Redes Sociales ?	71
¿Quieres un Grupo de herramientas ?	67
¿Quieres un Grupo de Noticias ?	66
¿Quieres añadir al acuerdo de mínimos el punto Fomentar la formación de comisiones de expertos independientes para asesorar en la toma de decisiones políticas ?	62
¿Quieres cambiar el punto 6 del acuerdo de mínimos a: "Desarrollar una carrera científica organizada y meritocrática." ?	60
¿Quieres cambiar el nombre del Grupo de Traducción y Divulgación a Grupo para la Culturización científica y el pensamiento crítico ?	56
¿Quieres modificar el punto 3 del acuerdo de mínimos a: Promover la defensa de la actividad científica y proteger su diversidad ?	54
¿Quieres que los informes (o resúmenes) muestren qué posturas han sido mayoritarias y además separen las fuentes científicas del resto?	51
¿Quieres que un grupo (noticias o divulgación) contacte con divulgadores para enriquecer los debates?	46
Sugerencias	42
¿Qué tal tu experiencia con esta herramienta?	42
¿Quieres englobar los puntos del acuerdo de mínimos de la siguiente manera?	26
¿Quieres que la divulgación de la información sea previa a los debates ?	24
¿Quieres cambiar el nombre del Grupo de Traducción y Divulgación a grupo de Divulgación y Comunicación científica ?	21
Normas asamblea I	5

Figure 9 Discussions in D-CENT instance for Podemos CIENCIA



Figure 10 Feedback discussion thread in the D-CENT instance for Podemos CIENCIA

Collaboration experiment with Podemos I+D+i

After the successful experience with Podemos CIENCIA, Ernesto Caballero-Garrido, co-coordinator and spokesman of Podemos I+D+i (aka. Investigación+Desarrollo+innovación; Research+Development+innovation) contacted the team responsible of the Spanish pilot to build up a similar instance for that circle. The circle of PODEMOS I+D+i, also included in the official list of Podemos' circles, states in its Facebook fan page: "Welcome to Podemos Research, Development and Innovation (R+D+I), we encourage you to come in, be active and participate! PODEMOS (yes we can)!"

The D-CENT instance was a copy of the one implemented for Podemos CIENCIA. In this instance, about 25 citizens participated in, at least, one of the 44 discussion threads created by the administrators of the Podemos I+D+i (see Figure 11). The discussion thread about feedback on the platform (see Figure 12) received 16 votes: 14 positive ones (87.5%) and 2 negative ones (12.5%). Again, some participants commented their user experience:

- It is great, but a folder system on the left side would help users organize information visually. Also, receiving notifications of replies to comments by email would be desirable.
- Very good and easy to understand. It contains a space for commenting, supporting or denying proposals. Perfect, many people can work with this tool.
- I do like it. However, we should set a general tool for all the tasks of the circle and just use the online social networks to diffuse ads from the (political) party.
- It does not seem very practical. I don't know if it is my fault but it goes very slow and some days it didn't work. On the one hand, I highlight that one can argue every vote. On the other hand, I miss having a document with all the proposals or an online version which integrates the full text of the proposals.

Title of the discussion	Participants
II-6 No contabilizar del presupuesto de I+D el gasto en I+D militar.	23
II-9 Separar al profesorado universitario en dos perfiles, uno más docente y otro más investigador	22
II-15/13 Proponer un nuevo modelo de publicación científica basado en el acceso abierto	22
Crear un espacio donde la ciudadanía puede elegir y/o sugerir un número limitado de proyectos en I+D+i.	22
II-26/25 Modernización de los colegios profesionales	21
II - Recuperación de la masa critica de investigadores	21
II-5 Aumentar la inversión pública de la I+D+i a como mínimo lo que esta acordado en la UE: un 3% del PIB para 2020	21
II-22 Establecer una política Open Data para la administración pública	21
II-2 Aumentar la cuantía y la cantidad de las ayudas post-doctorales y de estancia.	21
II-28 Establecer convocatorias donde la evaluación del CV sea principal.	21
Creación del Consejo de la Ciencia estatal como órgano consultor	21
II-1 Aumento de los contratos laborales de doctorado.	21
II-20 Desarrollar un programa de incentivos de creación de campus tecnológicos	20
II-33 Promover la incorporación estable del personal de investigación después de terminar la subvención para su formación	20
II-29 Apoyo a la financiación de proyectos prácticos y que fomenten el bienestar social	19
II-8 Priorizar el gasto en I+D+i en los recursos humanos.	19
Creación de equipos de gestión para liberar al personal investigador de trabajos ajenos a la investigación	19
II-10 Realizar un manual de buenas prácticas con criterios transparentes y públicos	19
II-32 Fomentar la creación de planes de investigación en hospitales para facilitar la investigación traslacional	19
II Comisiones y grupos de trabajo del círculo I+D+i	19
II-12 Desarrollar nuevas revistas	19
II - Optimización y mejora del modelo Universitario	19
II-23/21 Fomento de tecnologías abiertas y libres en el sector informático, industrial,	18

educativo y gubernamental.	
II-16 Permitir la compaginación del tiempo laboral de nuestros investigadores con la participación en actividades profesionales de transferencia tecnológica en otras organizaciones	18
II-18 Desgravar fiscalmente a las empresas que financien I+D+i público.	18
Democratizar la elección de cargos del CSIC	18
II - Acuerdos y normativas para el desarrollo de la asamblea	18
II-4 Establecer grupos de investigación competentes con el fin de generar grupos estables de conocimiento.	18
II - Modelo de contratación en I+D+i	18
II - Aprobación del acta anterior	18
II - Regulación y condiciones en la investigación público-privada	18
II-31 Creación de campañas de concienciación sobre la importancia de la inversión en I+D+i	18
II-30 Replantear admisión de proyectos en las convocatorias y permitir la subsanación de errores	18
II - Tejido empresarial público	17
II - ¿Qué te ha parecido esta herramienta?	17
II-3 Mejorar las condiciones laborales de los técnicos de los institutos de investigación.	17
II-11 Cambiar la evaluación del profesorado universitario para fomentar la producción investigadora y la transferencia tecnológica.	17
II-19 Facilitar la creación de empresas de base tecnológica y reducir su carga fiscal.	17
Instauración en España de la figura del Later Career Fellowship	17
II - Aprobación de los principios del Círculo I+D+i	17
II-7 Incentivar a las comunidades autónomas para ser las principales financiadoras de la I+D+i y que lo hagan de forma directa a los grupos de investigación más competentes.	16
II-34 Creación de estructuras mixtas para velar y fomentar la transferencia de conocimiento universidad-sociedad	15
II-27 Recuperar a la universidad pública como templo del conocimiento y motor base del desarrollo social.	15
II-24 Revisar la regulación de las titulaciones universitarias para cubrir legalmente el ejercicio de su profesión	15

Figure 11 Discussions in D-CENT instance for Podemos I+D+i



Figure 12 Feedback discussion thread in the D-CENT instance for Podemod I+D+i

Collaboration with Guanyem Barcelona

In the current context of emergence of local citizen candidacies, we are currently collaborating with the "digital participation" group in Guanyem Barcelona, in order to test DemocracyOS (with D-CENT upgrades) with a look at its later use for public validation of documents elaborated by their working groups.

So far, the software development work is oriented to adapt the tool to the current needs of Guanyem Barcelona users. A test with DemocracyOS has been carried on during the process of public debate of Guanyem's ethical code. This ethical code embodies an agreement among different political forces that will run together in the 2015 city elections in Barcelona.

Municipal initiatives and processes may use this tool elsewhere in Spain, if the "Guanyem test" is successful. The work is ongoing, thereby, results and the subsequent analysis will be available only later on. In the next section we list the needs and specifications required by the democracy pilot in the Spanish case, mainly based on the Guanyem Barcelona use case.

Specifications of the Spanish pilot for Guanyem Barcelona

The collaboration with Guanyem Barcelona has required to probe the needs of this initiative regarding digital tools. We have found the need for a tool that allows validation of the content of their political program, in an open and transparent manner.

From the first use exercise with DemocracyOS by Guanyem members, several conclusions have been drawn. There is a need to improve several aspects from the viewpoint of both the user and the administrator, on the front-end as well as on the back-end side of the tool. We have separated urgent needs from those that may wait for a later moment. The development work and the upgrades in the tool are also documented here: <https://github.com/GuanyemBarcelona/democracyos>.

Iceland

In January 2014 the D-Cent team came to Iceland in order to perform some interviews with the users of Your Priorities eDemocracy software, and draw up an initial plan for Minimum Viable Products to be implemented. Your Priorities will be improved based on these MVPs and other, new MVPs that will come out of the pilot process.

Bottom-up Municipal Democracy

The Icelandic Pilot uses the open source Your Priorities eDemocracy software that has been in lean user centric development in Iceland since 2008 by the Citizens Foundation and others. Your Priorities was from the very start part of the original EU proposal under its former name Social Innovation. Over 500.000 people have used the software and outside Iceland it has been used to change national laws in Estonia and is a core part of the NHS Citizen project in England. The Icelandic pilot will be focused on improving and upgrading Your Priorities by adding functionality needed by the users of the Better Reykjavík and Better Iceland websites.

The plan is to package Your Priorities as a Docker application that can be installed in a D-CENT node for other governments, groups or citizens across Europe to easily deploy and use.

Your Priorities has gone through 5 major revisions since the open source project started in 2008. It is now being revised the 6th time and design work going on working on splitting the Ruby on Rails app into an API only backend and a web components based front-end using the Polymer web components solution.

Better Reykjavík

In 2010 Better Reykjavík was opened, it currently has over 12,000 registered users. Better Reykjavik is the most successful example of the use of Your Priorities platform. It enables citizens to voice, debate and prioritize ideas to improve their city, creating open discourse between community members and City Council and also giving the voters a direct influence on decision making. Since 2010 over 70,000 people have participated out of a population of 120,000. 12,000 registered users have submitted over 5000 ideas and 8000 points for and against whilst 257 ideas have been formally reviewed with 165 accepted since 2010. The 10-15 top priorities are being processed by Reykjavik City Council and voted upon at meetings every month. Better Reykjavík also holds a yearly participatory budget event called Better Neighbourhoods where citizens send in ideas, officials cost then and then finally citizens do a secure binding budget vote to decide how to spend it.



The screenshot shows the Betri Reykjavík website interface. At the top, there's a header with the logo and navigation links: Home, Ideas, Categories, People, a 'SUBMIT YOUR IDEA' button, a search bar, and language/About links. Below the header, a message states that Betri Reykjavík is a monthly idea competition where ideas are voted on and implemented. The main section features a large blue heading 'Betri hverfi 2015' followed by text about the voting period (Oct 8 - Nov 7), the prize (300 million ISK), and the goal (to improve neighborhoods). To the right is a map of Reykjavík divided into colorful districts. Below this is a section titled 'IN OFFICIAL STATUS' with counts for successful, failed, and in-progress proposals. The featured proposal is for 'Hverfisgötu', marked as 'Successful'. It includes a description in Icelandic about improving the street, a voting section showing 273 'UP' and 13 'DOWN' votes, and a 'Share' section with 0 tweets and 11 likes. At the bottom of the proposal card, it says '9 points for 1 against' and has a 'Join debate' link. On the right side of the page, there are login options for Facebook and email, and a language selector.

Figure 13 Screenshot of Better Reykjavik #1 - <https://www.betrireykjavik.is/>

Better Neighborhoods

Better Neighbourhoods is participatory budgeting in Reykjavik. 300 million ISK (1.9 million EUR) is allocated for ideas from citizens on how to improve 10 different neighbourhoods in the capital city of Iceland each year. Citizens submit their ideas for projects they think will improve their neighbourhoods and City of Reykjavik evaluates costs and feasibility of each project. Then citizens vote on the ideas. Each voter has the same budget amount as the total and has to choose which projects matter most to him. This voting interface helps citizens understand the realities of budgeting. After the voting City of Reykjavik executes voted ideas. Some ideas are realized in a few weeks but other ideas take over a year. And citizens use the results from benches to footpaths, dog parks, better lighting, playgrounds etc.

Better Neighborhoods uses Your Priorities for ideas collection and another open source software called Open Active Voting that has been designed in iterations over 3 years of one vote, one person.

Betri hverfi Þú getur valið annað hverfi | Hjálp | Reglur | Útskrá |  

1 AUDKENNA 2 VELJA HVERFI 3 **KJÓSA VERKEFNI** 4 ATKVÆÐI MÓTTEKID

Vesturbær

C	M
32,0 notað, 5,1 eftir	Þú þarft ekki að nota alla upphæðina
Veldu verkefni með því að smella á heiti þess	
Helti verkefnis	Upphæð í milljónum
A Koma fyrir mispungum lyftingasteinum við göngustíga á Ægisslóu	1,0
B Endurbæta 'Bláa róli' á horni Bræðraborgarstígs og Túngötu	7,0
C Setja upp gönguljós á gönguleið yfir Mýrargötu við Seljaveg	12,0
D Lagfæra núverandi körfuboltavelli í Vesturbænum	2,0
E Lagfæra göngustíg milli Neshaga og Fornhaga	10,0
F Setja bekk og laga gróður á reit milli Víðimels/Kaplasjólsvegjar	1,0
G Setja upp hreytivölli eða prautabraut á skólalóð Hagaskóla	5,0
H Setja upp bekk á gatnamótum Hagamels og Furumels	0,5
I Endurnýja leiksvæði milli Melhaga og Neshaga með leiktaekjum	8,0
J Setja upp skilti við Ægisslóu um góða umgengni hundaelgenda	0,5
K Laga göngustíg milli Ás- og Sólvallagötu, sunnan Bræðraborgarstígs	1,0
L Malbika stíg milli Tómasar- og Lynghaga við Lynghagaróli	5,0
M Endurgera gangstétt og leggja hjólaein vestan megin á Birkimel	20,0
N Setja upp hraðahindrun á Hjarðarhaga. Nánar staðsett síðar	2,0
O Setja upphitun í strætóskýli við Neshaga, á mótis við Melaskóla	3,0
Helti verkefnis	Upphæð í milljónum
P Setja upp jólaljóðaskreytingar í Vesturbænum	3,0
Q Setja upp sumarblóm í kerjum á völdum stöðum	0,8
R Setja upp hellulagða hraðahindrun á Ægisgötu sunnan Vesturgötu	3,0
S Endurnýja gangstétt við syðsta hluta Grandavegar, austan megin	5,0
T Endurnýja gangstétt norðan Kaplasjólsvegjar, austan Flyðugranda	10,0
U Setja leiktaeki og malbika hluta stígs milli Grana- og Frostaskjóls	10,0
V Gróðursetja, snyrta og setja borðbekk á leiksvæði við Lynghaga	3,0
W Bæta lýsingu á leiksvæði og göngustíg við enda Fjörugranda	2,0
X Setja upp gönguljós á gönguleið yfir Ánanaust, við Vesturgötu	12,0
Y Bæta lýsingu á leiksvæði við Bauganes	1,0
Z Endurbæta gróður á ýmsum stöðum á borgarlandi í Vesturbæ	4,0
0 Endurnýja gangstéttar við Suðurgötu, við Háskólasvæði	15,0
1 Fegra umhverfi grásleppuskúra á Ægisslóu	2,0
2 Setja upp hellulagða hraðahindrun á Furumel, sunnan Grenimels	3,0
3 Útbúa aðstöðu við stíga á Landakotstúni með bekkjum og gróðri	3,0

GREIÐA ATKVÆÐI

Þú hefur ekki kosið ennþá | betrireykjavik@ibuar.is | Um kosninguna | Um Betri Reykjavík | Um Betri hverfi | [Reykjavíkurborg](#)

Figure 14 Screenshot of Better Reykjavik #2 - <https://www.betrireykjavik.is/>

Better Iceland

Better Iceland first opened in 2009 under the name Shadow Parliament. In October 2011 it was relaunched as Better Iceland.

Better Iceland uses Your Priorities to create open discourse between Iceland's citizens and members of parliament. Citizens can voice, debate and prioritize their own ideas and parliament proposals to improve their community.

In April 2013 the Icelandic pirate party pledged to use take ideas from Better Iceland and submit in the Icelandic parliament.

Subsites of Better Iceland are used for various local sites such as Better Hafnarfjörður and Better Eskifjörður (For a detailed description of Better Reykjavik and Better Iceland requirements see D1.2).

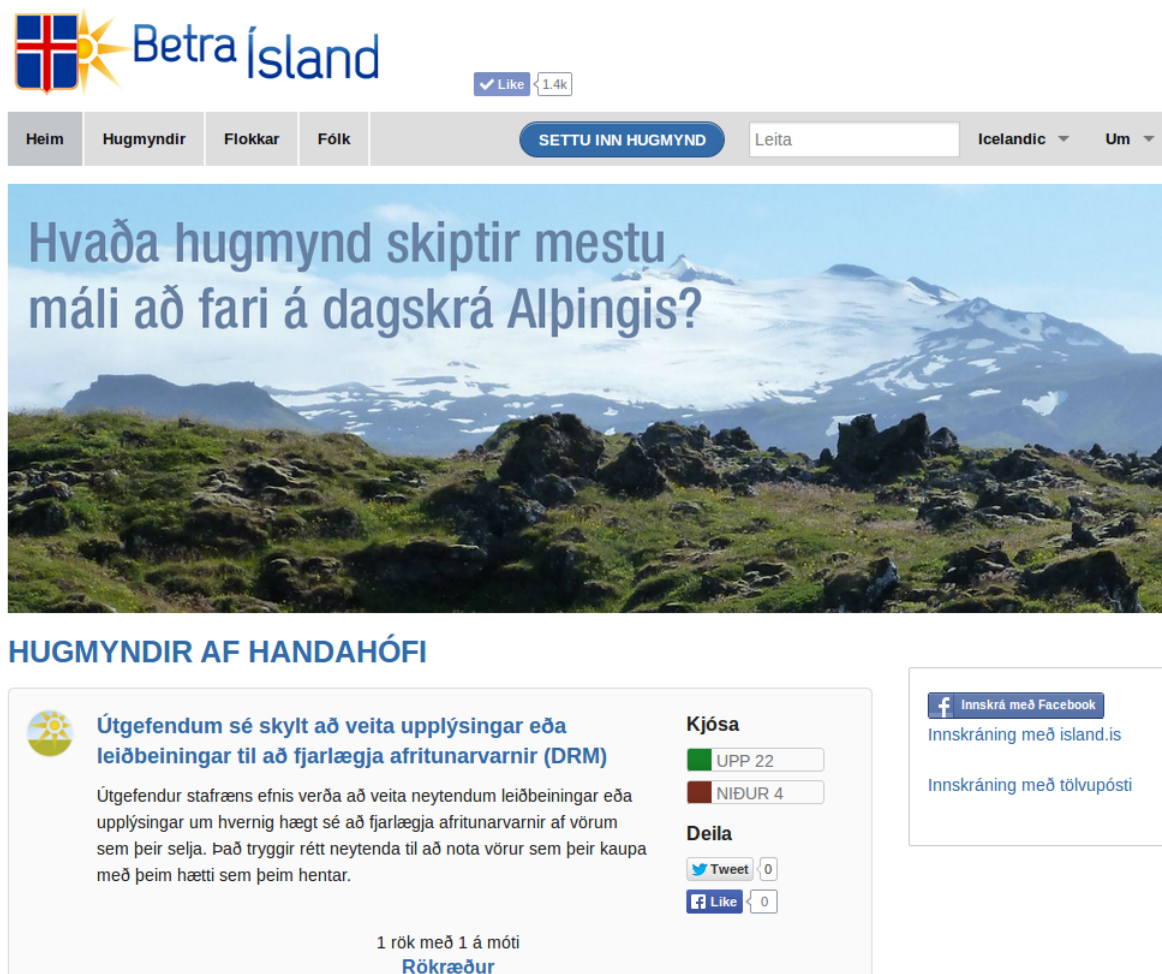


Figure 15 Screenshot of Better Reykjavik #3 - <https://www.betrireykjavik.is/>

The Lean Process

It started in January 2014 the D-CENT team came to Iceland for the Lean Inception Workshop (see D I.1 and D1.2 for full description of the lean development methodology adopted and its outcome) and did user interviews, co-design sessions and drew up a plan for some minimum viable products. Your Priorities will be improved based on those MVPs and new MVPs that will come out of the pilot process. Detailed below are some of the MVPs that came out of the original January D-CENT session and later iterative sessions with users groups.

Rating Town Hall responses to citizens proposals

One often repeated frustration expressed in the user interviews in Iceland was the inadequate argumentation received for why a given proposal was rejected, discouraging citizens from spending time

and energy engaging and developing new proposals. It was also found that if the explanation and response was detailed and generous, this would encourage citizens to engage further even in cases when a proposal was rejected.

The first MVP for Iceland will consist in the integration of a social rating system, for example a 1-5 star system, for users to rate responses from the Town Hall to Betri Reykjavik proposals, allowing citizens to collectively evaluate the quality of the Town Hall responses. The top five proposals from the Betri Reykjavik site are taken in and processed by the Town Hall at the end of each month. Rating systems can be an effective way of (i) increasing accountability to the public, (ii) improving quality of feedback, (iii) identifying and preventing failures, and (iv) providing reassurance to the public of the quality of a given service.

Quantitative metrics will be complemented with qualitative interviews with authors of rejected proposals as well as with Town Hall representatives. The interviews will be designed to measure satisfaction and engagement from the side of the citizens and responses from the Town Hall to positive and negative ratings. The ratings, in turn, can assist the Town Hall in improving its services, and increasing the communication and feedback to the citizens, opening up for another step in the conversation between citizens and Town Hall. Detecting trustworthiness of rating scores and responses is a challenging problem that will be tested through the Icelandic MVP.

Integrating distributed social networking functionalities

The second Icelandic MVP will consist in the Integration of Your Priority with a privacy-aware distributed open source social network, such as Diaspora. This integration will provide a privacy-aware deliberation space with a social and informal feel where people can be comfortable sharing unstructured information and undeveloped ideas with their peers. As an outcome new active citizen groups can form around issues that matter to them, and develop their ideas further before they can debate them and turn them into actions that the City can implement.

More images

Recently many people have complained about the lack of uploading images for ideas posted on Better Reykjavik and Better Iceland. We believe that by giving people the possibility to add images we will get people more interesting in interacting with ideas and be more interested in the debate. For Better Reykjavik we will integrate with city image database and use machine learning algorithms to automatically suggest images to use when people are typing in their ideas.

Better connection to city databases

Sometimes the quality of ideas in Better Reykjavik could be better, they are maybe based on old information. We believe we can increase the quality of ideas in Better Reykjavik by automatically pulling

in data from city databases while people are typing in their ideas, giving people access to relevant information. For example we can provide links to planning data for locations people are typing in and maps of areas. If people are talking about specific city institutions we can provide relevant links to information about those institutions.

Clustering of ideas

People sometimes submit very similar ideas that results in multiple ideas for proposals that divide the up and down votes. We believe by adding a feature that will show people similar proposals when they are entering their own ideas it will reduce the number of similar proposals and redirect them to new proposals.

Prototype of the AI technology that can accomplish this already running on the NHS Citizen Your Priorities website in the form of "recommended news stories" from the BBC and the Guardian that are automatically fetch for all ideas as seen in the screenshot below.

The screenshot shows the NHS Citizen website interface. At the top, the NHS Citizen logo is displayed with the tagline "The NHS belongs to all of us. Share your thoughts & make it better." Below the logo is a navigation bar with links for Home, Ideas, Categories, and People, along with a "SUBMIT YOUR IDEA" button and a search bar. The main content area features a proposal titled "All GP Patients automatically Opted Out of care.data" with a description: "Change the current NHS England proposals to automatically collect data from GP patient records unless patients 'opt out'." The proposal has 6 upvotes and 7 downvotes. To the right of the proposal is a sidebar with "Relevant news stories" including headlines like "Health data boom heralds new era of personalised medicine" and "Devon patients urged to opt out of data scheme".

NHScitizen The NHS belongs to all of us. Share your thoughts & make it better.

Home Ideas Categories People **SUBMIT YOUR IDEA** Search English About

Overview Next: Homeless community

All GP Patients automatically Opted Out of care.data

Change the current NHS England proposals to automatically collect data from GP patient records unless patients 'opt out'.

This is called automatically 'opted in'. The care.data project will then harvest GP data on a monthly basis and the data could be made available to private companies.

Add new point

Vote

UP 6 DOWN 7

Share

Tweet 16 Like 3 +1 0

Sign in with Facebook **Sign in with Twitter** [Sign in with email](#)

Relevant news stories

[Health data boom heralds new era of personalised medicine](#)

[Devon patients urged to opt out of data scheme](#)

[A&E: Some patients visit units 50 times a year](#)

[CQC 'to reveal NHS cover-up names' ministers say](#)

[GPs voice fears over giant patient records database](#)

[NHS waiting time data for elective surgery 'unreliable'](#)

[NHS targets review pledge by Carwyn Jones](#)

[Data-sharing 'good for patients'](#)

[Patients 'uninformed on data plan' | News | Health Service Journal](#)

Irrespective of opt-out vs opt-in, consent must be informed

Much clearer explanation of the benefits, and risks of the sharing of their data. Much more detail on when and how and why it will be shared, and the right to opt out again.

We should give consent to our data being collected

A basic principle of medical ethics is 'informed consent'. This means patients should always be aware of what is happening to them and their

Allow individual citizens online access to their own data

No data should be collected, unless the patient also has online access to all their own data. It seems totally unethical that data is collected about me and shared with 3rd parties, but I am not given access. Not only does this reduce safety for the patient, it also facilitates the collection of data which may be inaccurate (the patient would have the motivation to ensure that their data was checked and considered accurate)

Figure 16 Screenshot of NHS citizen

Open Source software used in the Icelandic pilots

Your Priorities

Coded in Ruby on Rails, Your Priorities is a Web based platform that enables groups of people to prioritize their democratic ideas and together discover which are the most important ideas to implement. <https://github.com/rbjarnason/your-priorities>. See more information in D4.2.

Open Active Voting

Secure electronic voting system. <https://github.com/rbjarnason/open-active-voting>

Active Citizen

Artificial intelligence empowering citizens democratically. <https://github.com/rbjarnason/active-citizen>

Contact / Your Priorities 3D

Multiuser 3D Virtual Reality frontend for Your Priorities based on three.js and WebGL. Is being used in a first of a kind live meeting on the 1. November 2014 in connection to the Better Neighborhoods project. <https://github.com/rbjarnason/contact>

KiwiIRC

Web based IRC client. <https://github.com/rbjarnason/KiwiIRC>

D-CENT Front-End

D-CENT is divided into a user-facing “front-end” that a user can access in a browser (or theoretically in a stand-alone program) on their client (such as smartphone or laptop) and a “back-end” that consists databases, blockchains, and other components. Front-end development is done through a system design methodology relying on exclusively on HTML, CSS and JavaScript with an responsive mobile user experience design. We rely on [Patternlab](#) to define the User Interface, so that mature HTML mock-ups can be created where the User Interface (UI) is disconnected from the backend. We also rely on [Mustache](#) modular description language to allow for design elements to be easily adapted to a variety of pilots and user-stories. (see e.g. <http://d-cent.github.io/patterns/?p=pages-blog>).

Patternlab fundamentally introduces a [systems design](#) approach to UI design. The smallest units of design are grouped as *atoms* that refers to basic elements such as fonts, colors, paragraph styles, lists, buttons etc. These atoms are then clustered together into *molecules* and then *organisms* – forming distinct but discrete design elements that can be easily reused. Finally, all these components can be employed to form templates for the dynamic generation of HTML pages that can be easily viewed and tested. While designing, a feedback loop is formed through the layers of user experience, to ensure a harmonized user experience.

The purpose is to develop an open and iterative style-guide for D-CENT pilots and - further down the line - for any D-CENT nodes to independently adopt or modify the design of *atoms*, *molecules* and *organisms* to their own template system. In this way, pilots have the freedom to design and run node-specific branches for user experience customisation that can then be incorporated to the D-CENT product main branch, based on the specific pilot experience.

Interviews done in Workpackage I will serve as the basis for establishing user story analysis and use case design. Each pilot will have their user story as requirement. The general feature set of the D-CENT platform will be built around the user stories identified through the pilots, and the final product will incorporate the requirements expressed by these pilots into a solid product.

Design Principles

The following principles will serve as the basic design principles for developing the platform:

API first

We build the application interface first, then subsequently analyse the user experience that should ensue from it. This enables us to develop a more synergic approach that combines user experience and machine-to-machine use cases. With regard to the development of the D-CENT platform, this is particularly important when it comes to the decentralisation and distribution of data between nodes.

- **Mobile first**
We develop responsive user interfaces, which allow users to use their preferred devices. This is important because more and more people primarily accesses the Internet through their mobile devices.
- **Minimalistic technical user experience**
We implement only one user story per user benefit, because we want to leave more space for user-created content.
- **Positive service**
We create a service that create positive experience to users, by greeting or rewarding them and inviting them to return to further use the service. The service also encourages users to experiment and explore new things by providing a secure user-experience.
- **Community-based**
The service is meant to serve as a community tool. It is designed to enable users to share content and communicate with one another through a versatile and interactive interface.
- **Equality and inclusiveness**
We recognize and support the intrinsic value of all human beings by creating and sustaining conditions that foster equity, empowerment, awareness and competence at the personal, group and organizational levels. Users are all equally welcome to use the service, and the service provides visual help and support to help users under complicated issues.
- **Ease of use**
We develop a minimal user experience that is simple and straightforward, so as to provide solutions that are clearer and more intuitive. See <https://www.gov.uk/designprinciples>
- **Technical accessibility**
We try to development a simple and efficient user navigation. We will follow the information given by the W3C Web Accessibility Group (see <http://www.w3.org/QA/Tips/> and <http://www.w3.org/TR/WCAG20/>). To allow for maximum accessibility, the User Interface should be as customizable as possible by the end-user, e.g. allowing the user control the colors, or the size of fonts.
- **Helpfulness**
Users shall be offered proper information, examples, and help concerning the objectives of the platform. Exact instructions should be provided as regards the use of the platform so as to steer the user towards constructive behavioural patterns. A proper explanation of the functionalities should be provided to explain clearly and concretely what are the benefits, consequences and ramifications of user actions.
- **Interactivity**

An interactive platform is necessary to encourage user participation. Simple types of interaction will be available to registered users, and after verifying an account with stronger authentication, there will be opportunity for further interaction with the functionalities of the platform (e.g. voting).

- **Simplicity**

When deciding upon functionalities, we will focus first on the simplest functionalities in order to create and validate the core benefit that the functionality is meant to produce. Additional functionalities should only be added based on metrics or concrete findings resulting from user experience study and observation.

- **Lean design**

In order to decide which new features to add, we rely on a constant feedback loop established between users and the software developers. The application is built through the pilots with a view to solve real and concrete problems. We will assess the success of the platforms by testing it against the actual needs recognized from the behaviour of the users in the pilots, rather than relying on work estimates and/or specialised or professional testers. The platform will be developed through a series of small iterations and the functionalities will be further refined until users are satisfied and able to use the functionality successfully.

- **Multilingual**

Any component should be tagged with a language code label for language versioning.

Front-end functionalities

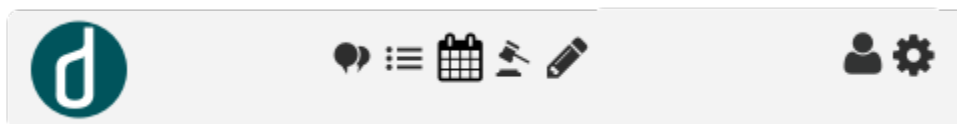
Login Page

A “Login Page” is presented when the user goes to the D-CENT to authenticate to a particular identity (See Section [Authentication](#)). Note that some D-CENT nodes may allow anonymous users who can view events and interact with the data, as to be determined in Workpackage 5. In this case, no login is necessary, but as anonymous selects an “Avatar icon” associated with a user identity or when they access functionality that requires authentication. The login page features “Login” and “Register” (create a new user) –button, with the latter links to a profile page and allows the user to establish their password for authentication (See Section [Strong Authentication](#)). Optional 3rd party login buttons like Twitter and Facebook are shown when applicable if groups allow users to login with third-party social networking. A final “Remember Login” button will save authentication cookie in browser for a user-configurable time. Unselecting “Remember Login” will remove the authentication cookie.

Navigation menu

All items related to navigation are responsive and possibly touchscreen-friendly. There are no drop-down menus, and all link are accessible by means of graphical logos in order to ease navigation and save screen space.

On a mobile device:



On a tablet:



On laptop, with a message-window and some unread notifications:

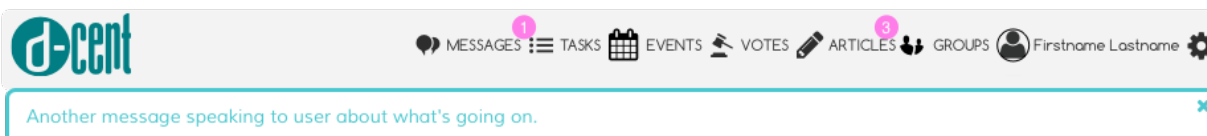


Figure 17 D-CENT sample visual layout on table

Context Help

Help text is openly displayed when the user might need to be informed as to the usage of a particular functionality. Help text is closed for functionalities that only certain users might need to be instructed about.

In general, helping elements and functionalities should be added to the user interface only on a user-need basis. For more details, see the design guidelines.

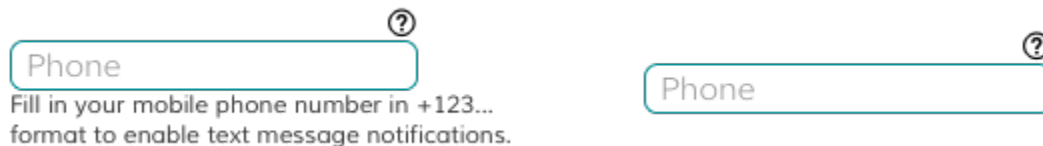


Figure 18 D-CENT Help text (Open and Closed)

Message Windows

Message windows are used when creating a dialogue with the user is necessary and/or beneficial. These windows answer frequently asked questions in the context of the user story and application functionality. They minimise the need for distracting the user with instruction manuals and help texts. Colors can be used to attach emotional emphasis on the message.

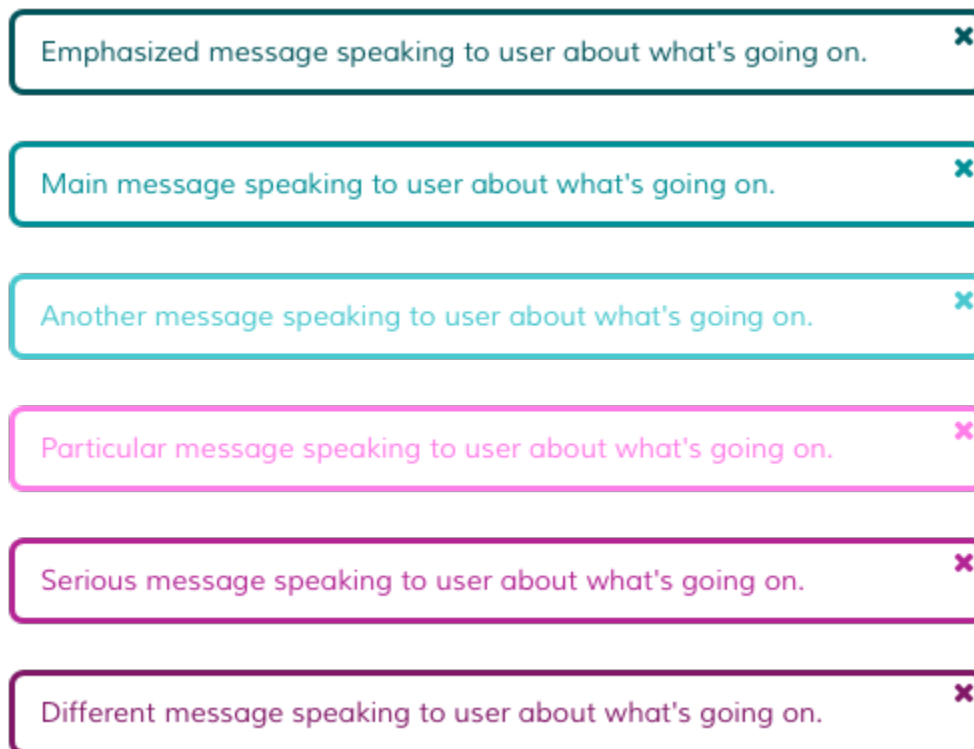


Figure 19 D-CENT Sample Message Windows

Notifications

Notifications are generated by the notification engine and stored in the core of the D-CENT platform: An activity stream is the collection of all messages sent by the users and groups within the D-CENT platform. Each individual then has their own stream, as do groups, and the D-CENT platform as a whole. Each individual activity is called an item and has a distinct owner (user or group), so an activity stream is just an ordered set of items. Normally, activity streams are ordered in a temporal order, from most recent till oldest.

The activity stream appears to be the ordinary user to be a rather simple newsfeed, similar to the status updates on Facebook or the stream of tweets coming in on Twitter. However, behind the visualization of the activity stream on D-CENT as a newsfeed is the W3C Activity Streams specification (when capitalization is used, we are referring not the D-CENT activity stream but the underlying W3C standard) that serves as the core of the D-CENT platform. This stream is displayed visually for users to read as a newsfeed. Note that the activity streams can be pushed to other media as well, such as SMS or e-mail. Lastly, they also serve as the basis for federation between different D-CENT instances (or “nodes”). Responsiveness Standard Twitter Bootstrap grid, mobile first -approach and simple rules for showing both content and navigation side-to-side are used. When enough space, Newsfeed is shown left and selected Article on the right.



Figure 20 D-CENT Sample Message Windows

Open Decisions

It should be reviewed based on pilot experiences, if and when view should change in between one-page and two-page views.

ActivityStream Newsfeed

Newsfeed

The Newsfeed contains the activitystream data, formatted for human readability. It is illustrated in Figure 20. Common functions include “liking” certain activities. The default ordering of activities will be by time-order, with most recent first.

Feed Filters

Feed filters implement filtering and sorting algorithms based on user stories. Unlike in major commercial browsers, these formats are under user control. For example, “Hot” activities can be detected.

Functionality buttons. Functionality buttons indicate the type and amount of user participation or activity that relates to the specific article in the newsfeed. When the user clicks on these buttons, he or she is brought to corresponding functionality on the content page (except for likes, where activity is immediate). On tablets and desktop computers, the messages button opens a message prompt. Examples of functionality include “Like,” “Comment,” and “Open for Co-editing” as detailed below.



Figure 21 D-CENT Mockup for Newsfeed

Mobile view

On mobile devices, a simple version of the activitystream based newsfeed is displayed as the default landing page. When the user clicks on an article, the newsfeed slides to the left (out of view) in order to display a simple UI where users can focus on the information they are looking for, without many distractions. This more simple mobile interface is illustrated in Figure 21.

Visualized calendar timeline

A User's activitystream will be read through the API with a client script to show the W3C Activitystream "embedded experience" (i.e. multimedia objects or Javascripts programs) and dependencies in a visual format. Note: there are ready-made libraries for this such as D3.js (d3js.org).

Map Mode

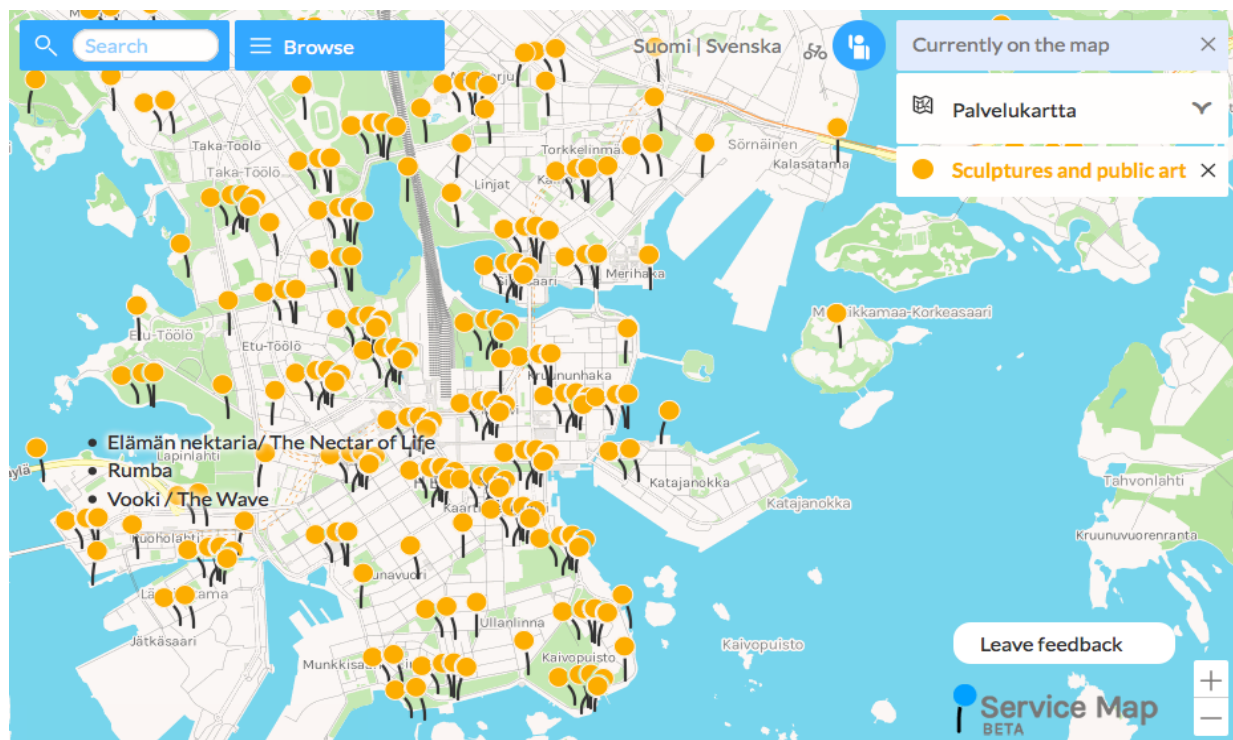


Figure 22 D-CENT Map Mode Mockup

Feature:

When opened in Map mode, the D-CENT platform will display a map of available activities or resources (Activitystreams) present in the local area, along with relevant information concerning these elements.

Use case:

Depending on the activity or typology of citizen interactions, there might be a need for geolocalisation technologies. In general, the possibility to geolocalise information will help strengthen the link between the type of information gathered and the territory to which such information refers. In D-CENT, the geolocalisation module will support citizens in their decision-making process at the neighborhood or municipality-level, thereby making the D-CENT platform more suitable for local tasks and activities with a strong territorial component.

In the “Bartering co-op” case, this feature has been implemented in order to show users each others’ locations. Other projects, such as Pumpipumpe, have shown how successful and efficient it can be to geolocalise the availability of tools in a neighbourhood (see for instance <http://www.pumpipumpe.ch>).

Description:

The geolocalisation feature will be implemented through OpenStreetMap as the backend. OpenStreetMap already gives users the ability to create custom maps showing only a particular type of resources. The D-CENT platform will cross-reference the information provided by OpenStreetMap with the additional information stemming from the local D-CENT database, in order to show a map of the various Activitystreams available in a particular area.

The map will include scroll-over icons displaying a short description of available resources. Clicking on the icon will take the user to a more detailed description of the resource, along with additional informations - if any.

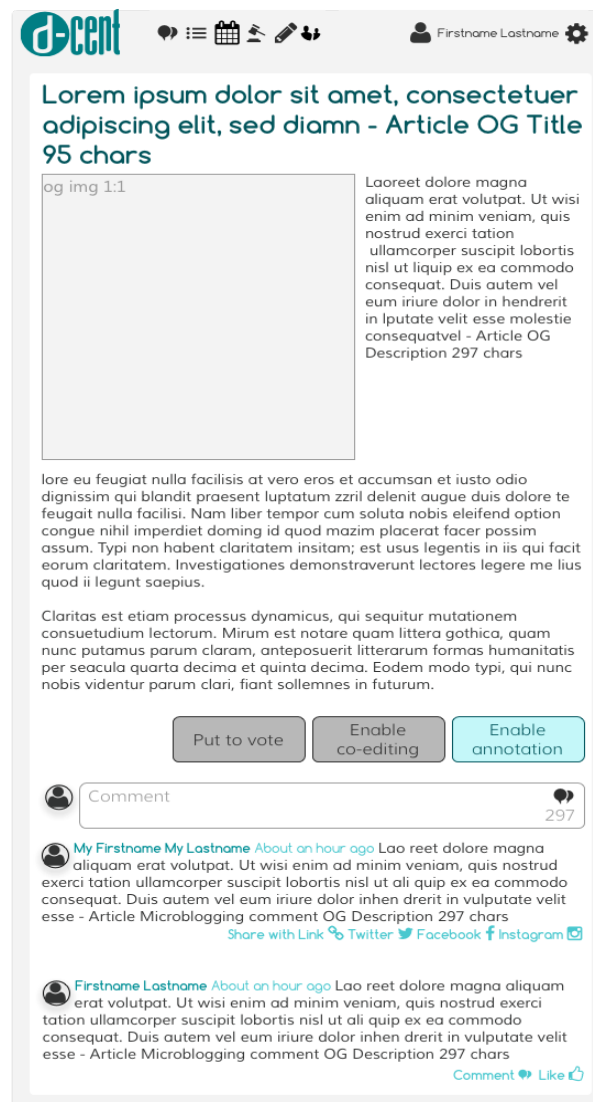
GeoJSON is a standard parameter of any ActivityStreams Object) in order to show Activitystreams items on a map. Some use-case specific client Javascript will be implemented to access the Activitystream data, whereas the map will be printed as a Leaflet layer with customized CSS. Besides, OpenStreetMap’s API allows users to view, edit and use geographical data in a collaborative way. Hence, after the initial bootstrapping, no specific technical skills will be required for users to annotate the maps with their own resources. Note that the GeoJSON format, as part of a RDF-based ActivityStreams 2.0 model, can support arbitrary data payloads, and thus is also an ideal data transport layer for sensor network data from a wide variety of sensors. This particular technological capability is not described in any more detail insofar as it is not mentioned by any of the pilots, but is possible within the D-CENT design if users want it.

It is important to note, however, that geolocalization information can be very sensitive to the extent that it qualifies as personal data. Thus, all personally identifiable information should be stored in the local client and the geolocalisation module should allow users to choose whether to make the data public or private, and implement a fine-grained access-control list for accessing the data based on the group interactions and social dynamics that have been recorded within the D-CENT platform.

Content

Articles

Articles are documents that can be written in plain text, html and [markdown](#). Articles have **owners**, users on D-CENT nodes that contributed them and then maintain control, although they may invite others to co-edit. Articles' owners can edit the text on-the-fly. In this mode, metadata (such as keywords and owners) are visible as form fields. The first 297 characters of the article are held into a separated data element for 3rd party social media sharing and search engine optimisation purposes.



d-cent [Icons] Firstname Lastname [Settings]

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam - Article OG Title
95 chars

og img 1:1

Laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in lputate velit esse molestie consequatvel - Article OG Description 297 chars

lore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudium lectorum. Mirum est notare quam littera gothica, quam nunc putamus parum claram, anteposuerit litterarum formas humanitatis per seacula quarta decima et quinta decima. Eodem modo typi, qui nunc nobis videntur parum clari, fiant sollemnes in futurum.

Comment 297

My Firstname My Lastname About an hour ago Lao reet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut ali quip ex ea commodo consequat. Duis autem vel eum iriure dolor inhen drerit in vulputate velit esse - Article Microblogging comment OG Description 297 chars
[Share with Link](#) [Twitter](#) [Facebook](#) [Instagram](#)

Firstname Lastname About an hour ago Lao reet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut ali quip ex ea commodo consequat. Duis autem vel eum iriure dolor inhen drerit in vulputate velit esse - Article Microblogging comment OG Description 297 chars
[Comment](#) [Like](#)

Figure 23 D-CENT Content Editing and Annotation

Buttons

Specific buttons for changing the state of the article and sharing it with other collaboration are only visible to article owners. For instance: Voting, Co-editing and Annotation. It may be possible to add images, and possibly also embedded video.

It is preferable not to rely too much on URLs linking to on external websites, but rather to manage document data as articles that are created within the D-CENT platform. Documents that are stored as structured data empower a more efficient search and allow for better content reuse capabilities. Each chapter, image, media clip, URI and html component contains a separate metadata table that contains the following information: language flag, version number, link to version history, automatically generated URI global for all D-CENT-nodes, and other user-defined metadata such as those necessary for tasks (see Section '[Tasks](#)' for more information about tasks). Note that version management for articles, chapters or other media components is dependent both on the selected data format and the selected co-editing solution. For instance, Pybossa crowd-sourcing platform could be utilized to import text strings from pdf files and convert them into Article or Chapters in text, and a file that is edited using a co-editing platform may later be saved for formatting using OpenOffice and then attached in the OpenOffice Data Format.

The preferred technical solution for media library and version management system is open until actual user needs will better inform the decision

Discussion

Discussions consist of comments, which essentially are microblogging articles owned by the user with a maximum length of 297 chars (the established length of [og:description](#)). Note that the comments should be signed by the user's public key in the D-CENT database. Discussion (also called "deliberation") provides the foundation for meaningful voting. After saving a comment, user is presented with the opportunity to copy a short URL (permalink) to the clipboard, and if they wish to send that comment to 3rd party social network such as Twitter. Discussion comments should be linked to all Activitystreams items that are stored within a D-Cent node. This is necessary to enable the organic improvement of user data and the use of social media links to be integrated directly within these items.

It is useful for users to give their personal opinions and specific arguments related to articles, as well as to provide further facts to enrich the article's content without losing ownership to their own writing.

Note that a user may serve as an **editor** of his article, which means the user may chose to share it only with a particular group as well as remove or edit the article.

Social media sharing

A "Share" functionality is available for microblogging comments, after the comment has been saved in the D-CENT platform. This functionality rely on third party platforms' sharing functionalities that can be expected to be available via Oauth and other client scripts. See the section on [Social Media Integration](#). A Permalink functionality is also available for facilitating social media sharing. Permalinks are in the

format of “https://d-cent-node.exampe.orgid” where *id* is the unique identifier of the microblogging comment that is linked to an article. This allows public articles to be linked to from other sites.

Agreeing on Comments

Note that comments themselves may be voted on, with an “Agree” link. This may be added without being directly added to the content page. For example, microblogging comments to other users’ update is possible. In such case, a link to both the original comment and the original article is created. Furthermore, the interface could be enriched by more actions. For example: coloring comments in a scale according to their popularity or agreement level. Even, some comments could be marked as hidden if they receive many negatives that might allow administrators to reduce trolling effects in a crowdsourcing methodology.

Likes

Likes are handled in a similar way to a comment. Liking an item adds the URI of the liked ActivityStreams Item to the User’s profile as a set of keywords. This enables the D-CENT platform to display an history of all liked items and to set up an advanced activity feed filters from the user client.

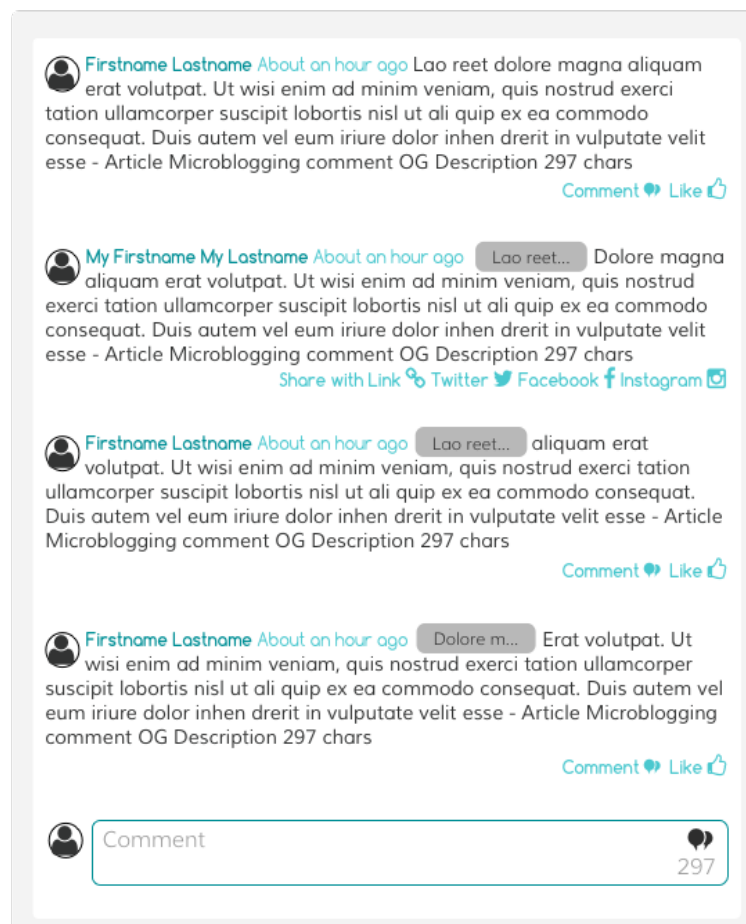


Figure 24 D-CENT Agreeing with Comments Mockup

Groups

For confidential organised collaboration, Groups with invitations and restricted access are required.



Figure 25 D-CENT Groups Mockup

Voting

Votes allow users with proper authorization to vote on an article using one of the methods described in Section [Voting](#). For example, a user may vote using “Yes/Abstain/No” in a simple case. In a more

complex case, a user may assign some numerical “weight” to their vote. This is especially important for use cases where the article is a proposed decision or law. Depending on the administrative set-up, article editors need to submit the for ratification by a group of users that have explicit voting rights. A user may also delegate their vote to other users (called “proxy voting”) as explain in Section ‘[Delegation algorithms](#)’.

When users click on their preferred option for voting, the choice can be saved in the open social datastore (with feedback how on how to be implemented in D4.4) and signed with their key. Users can freely change their mind before the deadline, and the new choice will be recorded into the append-only blockchain, thereby overriding the former vote. Voting on an article is implemented in a similar fashion to commenting on an article, yet the voting object is linked to the voting result. The voters serve as editors of their votes and can chose whether or not to show the vote result to others, depending on D-CENT node settings.

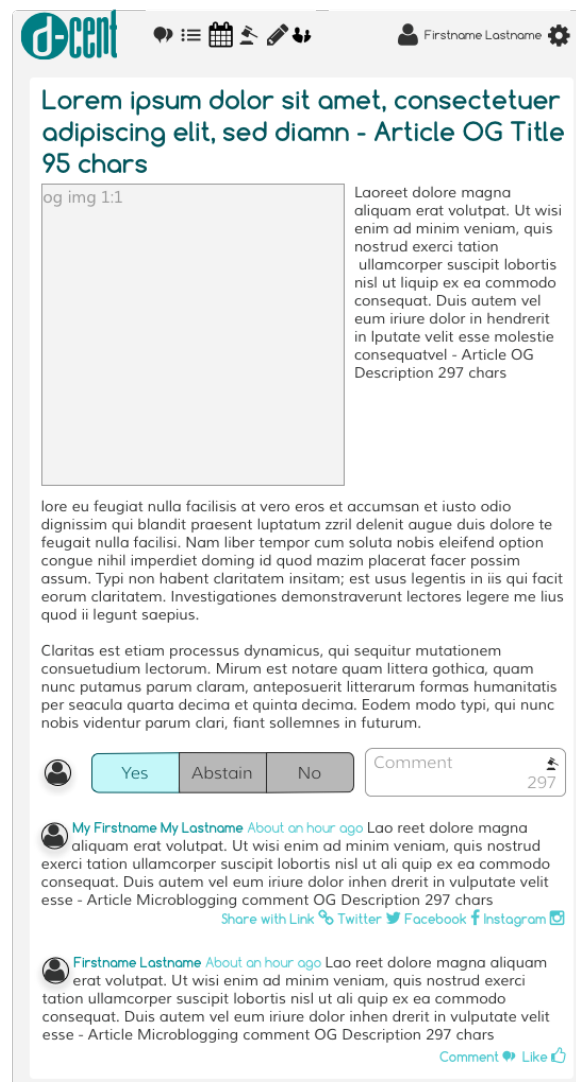


Figure 26 D-CENT voting mock-up

Alternate Use Cases

Voting results can be used as metadata for visualizing the deliberation process. One method which has been successful in Iceland was to group the arguments for and against a particular issue, in order to support and illustrate the deliberative process. Also, project coordinators need to be able to propose a project, which comes along a description and a list of required resources and costs. Users need to be able to “vote” on the projects they would like to see implemented, taking account of the actual budget limitations. This could be done through the “weighted cumulative voting” systems (detailed in the Voting section), whereby users are assigned a particular budget, which they can assign to different projects by order of preferences until the budget dries up.



Figure 27 D-CENT Deliberation Mockup

Participatory budgeting

Users are assigned a particular budget, which they can assign to different projects, by order of preference, until the budget dries up. Participatory budgeting (see e.g. McCarthy’s algorithm described later) allows the user to prioritize investment and have the resources allocated proportional to the project costs. Similar work has been done in terms of Secure Voting by Your Priorities that could be used for future development, as determined in WP5.

Crowdfunding Selection of Lorem Ipsum Dolor



Figure 28 D-CENT Participatory budgeting Mockup

In the D-CENT platform, Groups can be opened for managing participatory budget accounting. Users add ActivityStream objects with *Cost(Badge)* and *Url* of their *Product(Article)* to propose Projects, which are added as Articles (and can thus be updated, commented, Voted upon, etc.) like any other Articles.

Articles' OpenGraph data, like Title, Image and Description is linked to Project and Project objects are stored as ActivityStreams Products. If there are more than 140 chars in the Project description, they are visible when the respective Project is selected.

Users drag the proposed projects they would like to see implemented to the Chosen column, or clicks on Choose button, and the UID of the selected Project is stored in the blockchain. The progress bar on top is updated, according to the budget allocated to the projects. Users can keep selecting new Projects until the budget runs out. Any reselection will overwrite previous data as long as the voting procedure is open. The “weighted cumulative voting” system is detailed in the [Voting methods](#) section.

<i>Displayname</i>	<i>Property</i>	<i>Value</i>	<i>Description</i>
Currency unit, i.e. EUR	Badge	Activity Streams Object quantity	Projects have Badge value, and Badges are awarded to a user when Voting completes.
Project	Product	Activity Streams Object String	Has value, Image, Title and Description.
Article	Object	Activity Streams Object URI	Contains Product description.

Proxying

See Architecture - Voting

Technical options are to be tested against practical user experiences. The array of functionality some platforms provide, is clearly overwhelming for a majority of users.

Collaborative bottom-up editing

Articles can be opened into co-editing mode by their owner and shared to other document owners for real-time collaborative editing using functionality similar to the well-known Etherpad program (<http://etherpad.org>). This is useful as article owners might want to give other people access to collaborative edit their documents.

When Article Owners click on the button “Open for co-editing” on the Article Page, collaborative editing becomes available for anyone marked as Editor for the article. When Article Owners click on the

“publish” button on the Article Page, co-editing is revoked and a new article version is saved and published on the platform.

Co-edited versions of Articles can be saved as new versions on the D-CENT node, or they could be regarded as a separate functionality allowing to move an Article back and forth between editing and co-editing mode. The choice depends on the capacity and requirements of the version management system as they will be established through the pilots.

In terms of implementation, this is not a straightforward front-end feature. It may be implemented either using a branch of Etherpad (<http://etherpad.org>), a wiki-system such as MediaWiki (<https://www.mediawiki.org/>), or direct HTML editing via Content Editable by Mozilla Foundation. This will be decided in WP5. Different systems have different advantages. Etherpads allow live co-editing. Wikis do not allow live co-editing between multiple users, but provide a detailed versioning control mechanism. Lastly, Content Editable allows HTML to be edited directly, but unlike etherpads and wikis, it is not well-known to most users.

This functionality could be implemented through one of the following options:

1. Content Editable by Mozilla Foundation
2. Together by Mozilla Labs
3. Firepad by Firebase
4. ShareJS by Joseph Gentle
5. Foyt by Antti Leppä

Annotation

Annotations are implemented as ActivityStreams item linked to articles, but refer to only a subset of the text in an article and are so displayed “inline,” generally to the right of an article. This is illustrated in Figure 29.

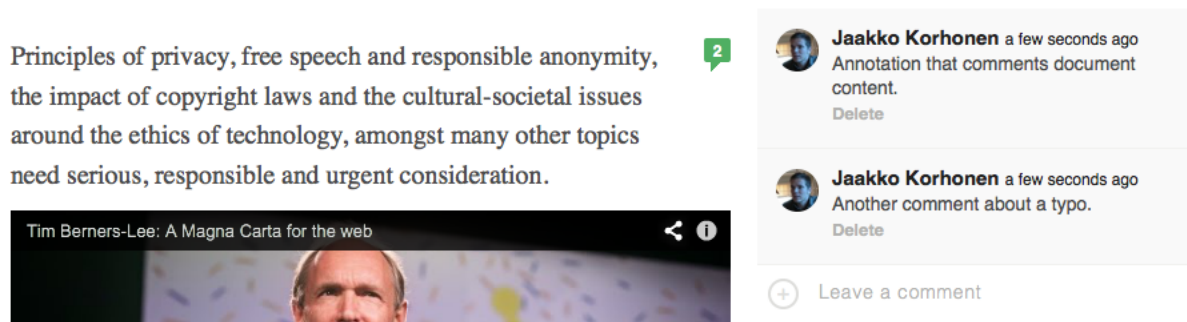


Figure 29 D-CENT Annotation Mockup

Annotations are needed article editors might want to allow third parties to annotate their text and possibly suggest changes that are connected to particular parts of text. This would be useful if, for example, in a proposed law (which is often a very large piece of text) there would be needed the ability to comment only on a single sentence.

Tasks

Tasks are implemented as Articles with specific metadata, which can be assigned to particular Users. This helps users keep track of articles that have to go for a particular workflow. For example, civic activists might need to assign tasks to organise and coordinate particular events and quickly gather volunteers capable of providing specific contributions to the project. In a typical legislative example, a user may first create a rough draft, then allow up to a week for co-editing with a trusted group of experts, and then bring the entire proposal to a vote within 30 days.

Tasks of Lorem Ipsum Dolor

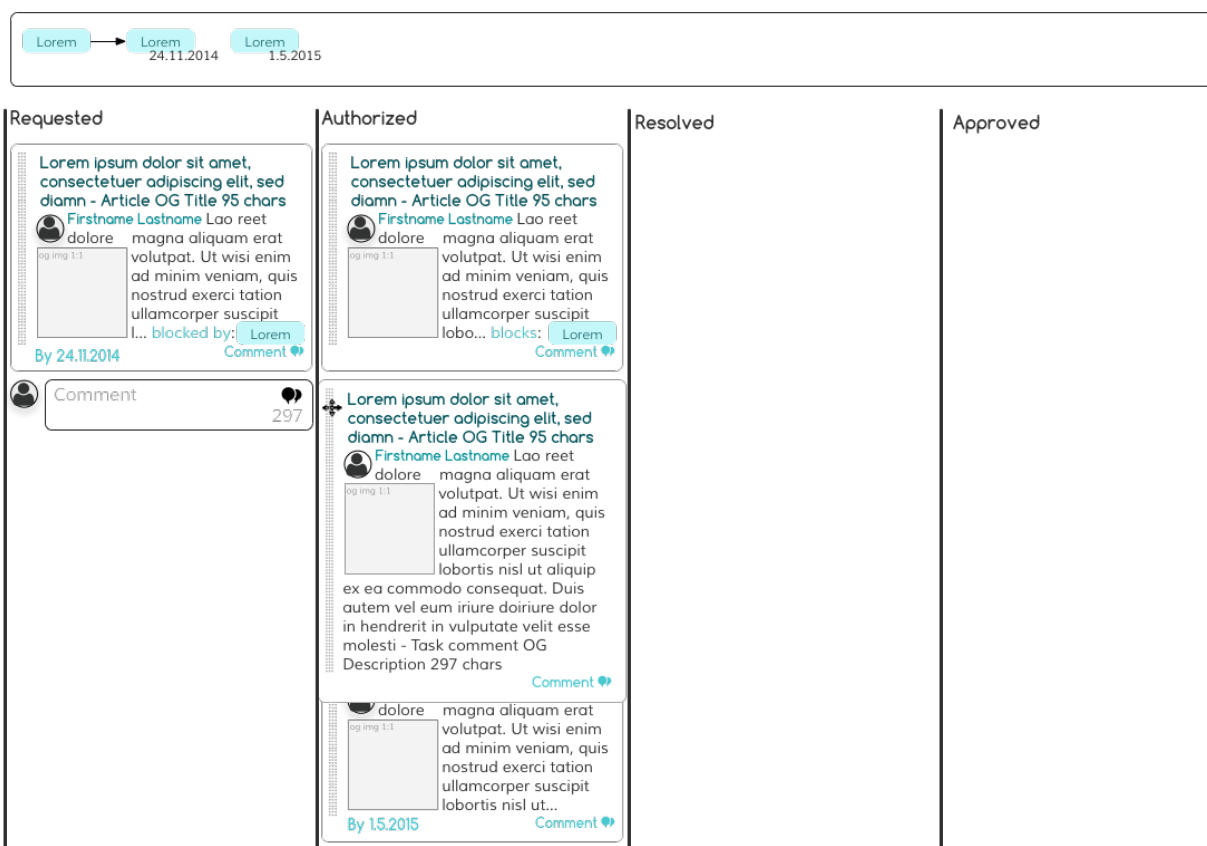


Figure 30 Task ActivityStream Schema

Given this is very advanced functionality; we have provided an initial table of functions that are necessary to define tasks using W3C ActivityStreams. Tasks are Articles that are created with specific metadata as described in the Activitystreams 2.0 Task Element description (more details at <https://github.com/activitystreams/activity-schema/blob/master/activity-schema.md#task>). Assigning a task is done by posting the task out as notification to the users involved in the group associated with the task, or all users of a D-CENT instance (for particularly voluntaristic tasks). Based on user feedback, the task can be configured to be exported to a 3rd party platform such as Trello (<http://trello.org>),

FP7 – CAPS - 2013 D-CENT D4.3 Technical Design of Open Social Web for Crowdsourced Democracy VI
 OpenERP (<https://www.odoo.com/>), Github (<http://github.com>) or Jira (<https://open.jira.com/>) if the 3rd-party platform is compliant with open standards for tasks at the time of WVP5.

<i>Displayname</i>	<i>Property</i>	<i>Value</i>	<i>Description</i>
Assignee	Actor	ActivityStreams Object	An Activity Streams Object that provides information about the actor that is expected to complete the task.
Deadline	By	String	A RFC 3339 date-time specifying the date and time by which the task is to be completed.
Title	Object	Activity Streams Object	An Activity Streams object describing the object of the task.
Is blocked by	prerequisites	Array of Task Objects	An Array of other Task objects that are to be completed before this task can be completed.
Blocks	supersedes	Array of Task Objects	An Array of other Task objects that are superseded by this task object.
Button	Verb	String	A string indicating the verb for this task as defined in W3C ActivityStreams.

Figure 31 Task ActivityStreams Specifications

Events

Events are articles that represent an upcoming event or an event in the past. Events are articles that have a certain number of attendees as divided into categories (given below) and are associated at least with a particular time, but also optionally with a particular place or group. Events may be public or restricted. For example, the Pirate Party may call for a rally at a certain time in front of the Parliament, or a private group of activists may call for a meeting to discuss internal matters. Compared to normal articles, have specific metadata and have a specific schema that can be used to trigger notifications, and must be linked to users that are participants. The broadcasting to groups is dealt with by the access control capabilities of D-CENT.

<i>Displayname</i>	<i>Property</i>	<i>Value</i>	<i>Description</i>
Firstname Lastname	attendedBy	Collection	A collection object as defined in the W3C ActivityStreams specification that provides information about entities that attended the event.

Firstname Lastname	attending	Collection	A collection object as defined in the W3C ActivityStreams specification that provides information about entities that intend to attend the event.
Time as defined in user settings.	endTime	String	The date and time that the event ends represented as a String conforming to the "date-time" production in [RFC3339].
Number of invited users	invited	Collection	A collection object as defined in Section 3.5 of the JSON Activity Streams specification that provides information about entities that have been invited to the event. Not needed if the event is public ("open to all")
Number of users "maybe attending"	maybeAttending	Collection	A collection object as defined in the W3C ActivityStreams specification that provides information about entities that possibly may attend the event.
Users not attending in reality.	notAttendedBy	Collection	A collection object as defined the W3C ActivityStreams specification that provides information about entities that did not attend the event.
Users who intend not to attend.	notAttending	Collection	A collection object as defined in the W3C ActivityStreams specification that provides information about entities that do not intend to attend the event.
Time the event begins. Also, an optiona "endTime" may be defined in the same manner.	startTime	String	The date and time that the event begins represented as a String confirming to the "date-time" production in RFC 3339.

Figure 32 Event ActivityStreams Specifications

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam - Article OG Title 95 chars

og img 1:1

Laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in lputate velit esse molestie consequatvel - Article OG Description 297 chars

Tag1

Tag2

Address

GPS

Tuesday 29th March 2015

lore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Typi non habent claritatem insitam; est usus legentis in iis qui facit eorum claritatem. Investigationes demonstraverunt lectores legere me lius quod ii legunt saepius.

Claritas est etiam processus dynamicus, qui sequitur mutationem consuetudium lectorum. Mirum est notare quam littera gothica, quam nunc putamus parum claram, anteposuerit litterarum formas humanitatis per seacula quarta decima et quinta decima. Eodem modo typi, qui nunc nobis videntur parum clari, fiant sollemnes in futurum.

Attendees

Discussion

Comment

297

My Firstname My Lastname About an hour ago

Lao reet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut ali quip ex ea commodo consequat. Duis autem vel eum iriure dolor inhen drerit in vulputate velit esse - Article Microblogging comment OG Description 297 chars

Share with Link
 [Twitter](#)
[Facebook](#)
[Instagram](#)

Firstname Lastname About an hour ago

Lao reet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut ali quip ex ea commodo consequat. Duis autem vel eum iriure dolor inhen drerit in vulputate velit esse - Article Microblogging comment OG Description 297 chars

Figure 33 Event ActivityStream Mockup

User Settings Page

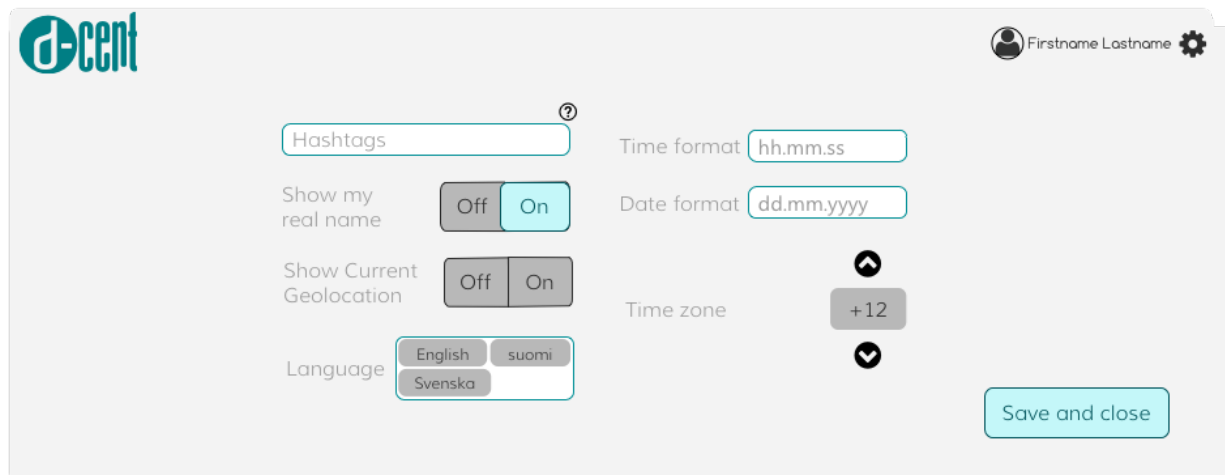
User settings is a slidedown that appears by clicking on the “Settings” button that lets them display a profile. The Settings Page is a form in a slidedown, and the profile and its elements may be marked private or public. Elements include at least (as given in Figure 34)

<i>Parameter</i>	<i>Value</i>	<i>Purpose</i>
First name	First name(s)	First name of user (Forename)
Last name	Last name(s)	Last name of user (Surname)
Email	email	Some contact info needs to be verified for Editor access. This can be phone, email or Strong Authentication. User can have open Discussions anonymously. Their account and IP can be blocked.
Phone	Numeric string	Some contact info needs to be verified for Editor access. This can be phone, email or Strong Authentication. User can have open Discussions anonymously. Their account and IP can be blocked.
Hashtags	Comma separated keywords.	Keywords are used for ActivityStreams sorting and filtering as well as Notification Engine.
Time zone	Time zone	Used for timestamp display.
Time format	Choice of time format options.	Used for timestamp display.
Language	Name of the language in that language. For example, EnglishSuomi, Espanol, etc.	Used for user interface, Message Windows and Help Texts.
Location of residence City, Zip, Street, etc vcard elements.	Strings	Used for calculating timestamps, for ActivityStreams sorting and filtering as well as Notification Engine messages.
Show Current Geolocation	On / Off	Granting access to GPS device or IP information required for mobile use. Used for real-time context aware use cases, i.e. maps of User’s current area.
Show my real name	On / Off	Some nodes may have a policy to require real names.

		User is prompted to enable where applicable.
Groups	List of group names as tags.	Visibility to own groups as tags and possibility to remove.
Allow roaming cost	On / Off	Required for mobile use.

Figure 34 User Profile Schema

User settings is a slidedown that appears by clicking on the settings button.



The mockup shows a user settings interface for 'd-cent'. At the top left is the 'd-cent' logo. At the top right is a user profile icon with the text 'Firstname Lastname' and a gear icon for settings. The settings are organized into two columns. The left column includes: a 'Hashtags' input field with a help icon; 'Show my real name' with 'Off' and 'On' toggle buttons; 'Show Current Geolocation' with 'Off' and 'On' toggle buttons; and 'Language' with buttons for 'English', 'suomi', and 'Svenska'. The right column includes: 'Time format' with an input field showing 'hh.mm.ss'; 'Date format' with an input field showing 'dd.mm.yyyy'; 'Time zone' with a dropdown menu showing '+12' and up/down arrows. A 'Save and close' button is located at the bottom right.

Figure 35 User Settings Mockup

User is also taken to Register page when they login with unknown 3rd party authentication and when they open “Do you want to register instead?”



The mockup shows a user registration interface for 'd-cent'. At the top left is the 'd-cent' logo. At the top right is a user profile icon with the text 'Firstname Lastname' and a gear icon for settings. The registration form consists of two columns of input fields. The left column includes: 'First name' (with a help icon), 'Last name', 'Email', and 'Phone' (with a help icon). The right column includes: 'City', 'Zip', 'Street', and '...etc vcard elements.'. A 'Save and close' button is located at the bottom right.

Figure 36 User Registration Mockup

Front-end State diagram

Figure 37 illustrates the typical flow of articles in a D-CENT node for a given user as a state diagram. This flow shows how articles may be shared, co-edited, and the like, and are distinguished from tasks and events, as well as how these kinds of activitystream items may also be pushed out to a notification engine. The schema of activitystream items that we have explained above is given in Figure 38.

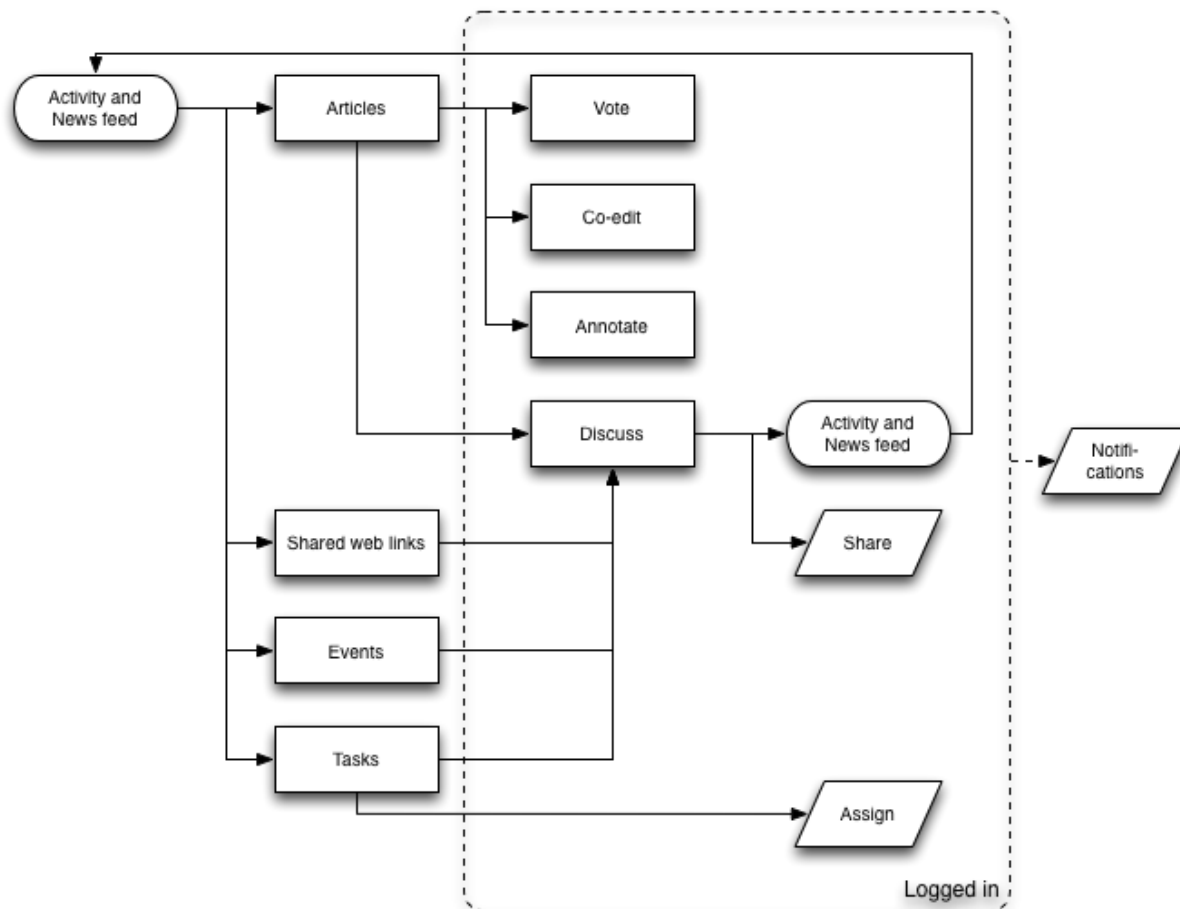


Figure 37 State Diagram of activitystreams in D-CENT

Item	Description
Activity feed	News feed is the main social media content feed that users follow and is distributed and aggregated across D-CENT nodes.
Articles	Articles are the main objects read and processed by users.
Shared web	Links are posted by users, and published for commenting as articles, but are not

links	available for annotation or editing.
Tasks, Assign	Tasks are shown as Articles, but assigned to Users.
Event	Tasks are shown as Articles, but have a specific schema.
Vote	Vote is a object like a comment, but is signed with a key in the blockchain.
Co-edit	Bottom-up co-editing is used when document has several owners that create content co-operatively. This enables civic activists to combine skills and effort in a heterogenic and improvised communities.
Annotate	Annotations are objects created by Users that contribute to Articles that they don't have ownership over. Annotations are owned by Article owners, not by their creators.
Discuss	Microblogging objects that are linked to Articles are owned by their creator and can be broadly linked to objects within D-CENT and external systems.
Share	Discussion comments can be given an unique url and be sent to other systems.
Notifications	Notifications are provided by the Notifications Engine that consumes activitiystreams data and generates notifications to a variety of Application Interfaces (API).

Figure 38 Schema of activitiystream item types and actions in D-CENT

Architecture

Architectural principles

Although we have already described in detail the front-end, the “backend” technical architecture needs to be described for the core features of the frontend. The “architecture” is a detailed description of how the various code components in D-CENT interact, with each feature being described in detail in Section Core Features. Also, D-CENT’s envisioned features go beyond the frontend described in the previous section, and include interactions with various D-CENT enabled applications such as CKAN open data and Pybossa for crowd-sourcing. The interaction of these components with D-CENT is detailed in Section Open Data.

In general, architectural development principles are used as basis for making design choices when developing a new web service such as D-CENT. Architectural principles have been adjusted to accommodate decentralized open source development, dispersed piloting and lean software development methodologies.

Modular, exchangeable building blocks

D-CENT will endorse modularity to enable decentralized, standards-based, and interoperable system design. The D-CENT platform will utilize open source components, libraries and pieces of existing software. The design should be kept modular so that individual functional parts can be replaced with other, similar solutions if e.g. the original source is no longer maintained. “Plugin”-styled interfaces allow the replacement of components. This enables varying component technology lifecycles as well as flexible implementation of varying environments and service solutions that are built to change, rather than build to last. Be aware of components’ potential replacing roadmap candidates before the need for replacement is urgent.

Technical feedback

We will build a rough consensus and start working on running code immediately. Our implementation will re-use standards when appropriate from the IETF/W3C, and will feed into new efforts once there are proven concept and user testimonials. We’ll build feedback loops from users and developer as well as to ensure standardization communities are participated in ratifying any standard proposals.

Agile

We will use open-source and free software to avoid vendor and technology lock-in. You can achieve this by API first approach and having varying technology options for different service modules. We will build module lifecycles to aim for standardized schemas and APIs.

Reusable and refactored

We will reuse data and rewrite code, but be ready to start software development work from scratch after any iteration. Usable code is tracked in D4.2 and “Open Decisions” of D4.3.

Value frameworks

Create software, data and participation experience that is open by default. We comply with the Open Definition, <http://opendefinition.org/od/>.

Assure user control over personal and social data

It is increasingly difficult to create trust between users and institutions, when institutions and commercial companies are very opaque about what they actually do with users ‘data. The decentralized, standards-based solution that can be envisaged in the context of D-CENT is instead one where citizens have control over their data, maximizing the value they can gain beyond monetary incentives, considering social data as commons. In D3.4 (WP3) we will carry out research on identity systems, taking into account philosophical, socio-economic, democratic, and ethical issues related to personal and social data in the digital economy.

D-CENT Back-end Features

D-CENT is a complex project with many interlocking features specified in this deliverable. We will deliver the parts in various deliverables as part of Workpackage 5, with the specific deliverables that embody each feature enumerated below. The features below are currently not prioritized for development, and thus we will prioritize based on feedback from users using the “LEAN” development process. Note due to the change in the partner leading the implementation, some work may be delayed. Also, some features may be dropped if the resources allocated in WP5 do not allow them to be implemented within the time allocated. A number of diagrams illustrate the various components of D-CENT and how they interrelate.

An overarching architecture diagram is given in Figure 39 that presents an easy-to-understand diagram of the relationships between the core components of D-CENT. Afterwards, each component is itself described briefly along with which projected deliverable in Workpackage 5 will present the deliverable.

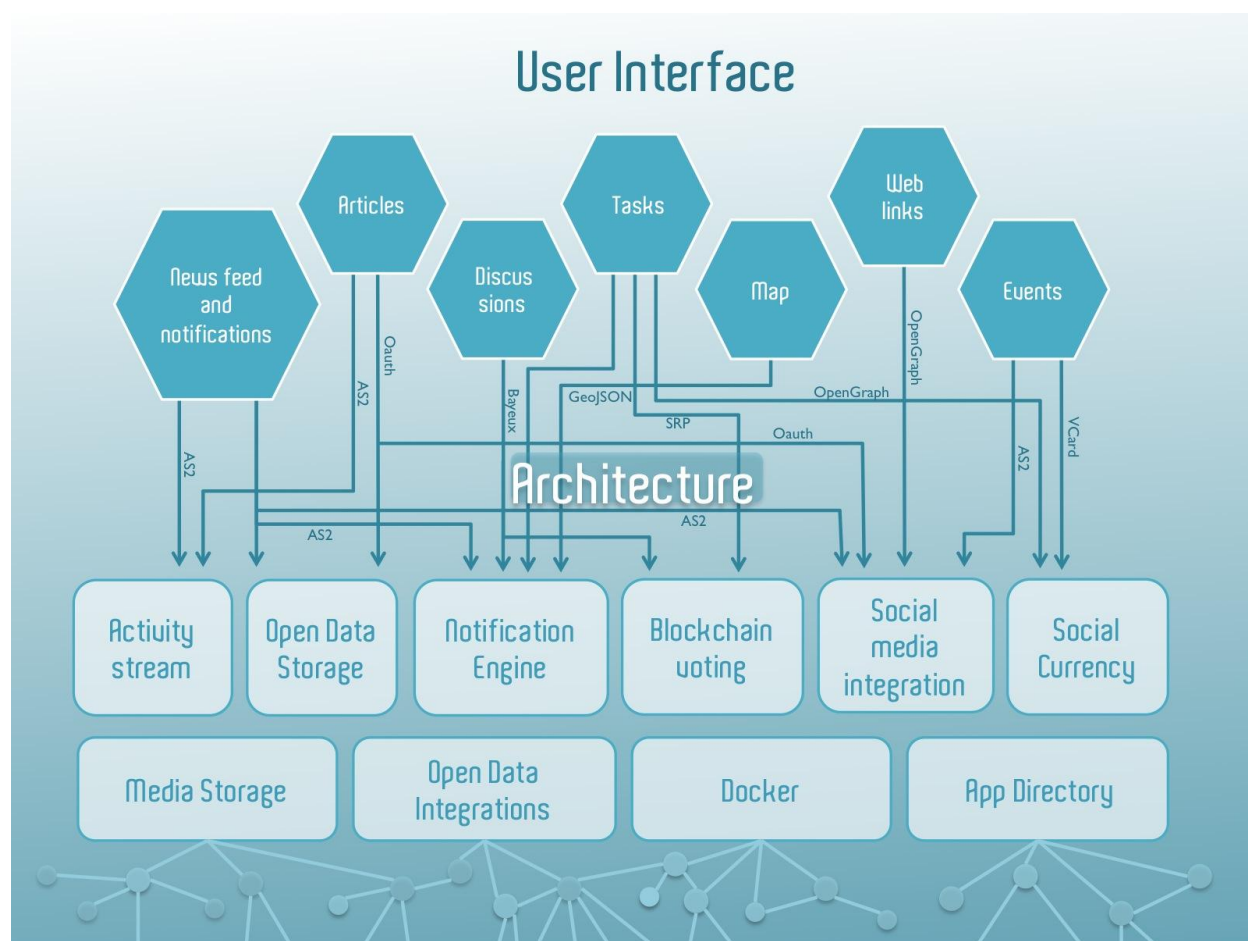


Figure 39 Overall Architecture of D-CENT. Core frontend features are listed on the top, with various software components that these features embody listed in the second row. The last row shows other applications that plan to interact with D-CENT.

The features are divided into two groups. The first is core features of the D-CENT platform. By D-CENT platform, we mean the databases, authentication, and previously described front-end that come as “standard” features of the default installation of a D-CENT node on a single server or virtual machine, although for a single node various features may be added or disabled. These features are all listed and described in Section ‘Core Features’ and will be available as a unitary Docker package. Note as regards D-CENT itself, a D-CENT node is simply an instance of the D-CENT platform. True to its name, multiple D-CENT nodes may federate in a decentralized manner to share information using activity streams. This forms a D-CENT network, which will be further described as implemented in WP5.3 and as standardized by the W3C Social Web Working Group. Preliminary details as regards how D-CENT will do decentralized social networking is available in Section ‘D-CENT Applications’ with the plan for standardization in Section ‘Standardization plan’.

By D-CENT applications (to be precise “D-CENT enabled applications”), we mean applications that take advantage of the core features of the D-CENT platform but require the installation of a new Docker and so may be located on a different server or virtual machine than the core D-CENT node that maintains the various features. However, to the end-user the experience between moving from the core platform to particular applications should be transparent, as the applications should be enabled with and OAuth for “single-sign on” and some applications could even be embedded using “iframes” within the D-CENT frontend.

D-CENT Core Features

The features on a high-level are listed here, and described in detail in Section [Features](#) and Section [Applications](#). Note that after each feature is the projected deliverable in WP5 is given in parentheses after the feature. Note that the only the first deliverable is listed, although the workplan has iteration-based deliverables D5.6, D5.7, and D5.8 that will refine these basic features as resources and time allow. These deliverables are tentative and may change due to the “lean” development process.

Front-end

- **Articles:** Post articles (D5.1)
- **Discussion:** Allow comments on articles (D5.1)
- **Voting:** Allow people to vote on an issue (D5.1)
- **Annotation:** Add annotations to specific parts of text to specific articles. (D5.1)
- **Notifications:** Sending reminders out such as those about decision results (D5.3)
- **Collaborative editing:** Integrate co-editing into the document. (D5.3)
- **Task Management:** Attach assignments of actions to users to articles. (D5.3)
- **Events:** Attach to articles time, place, and invitations. (D5.3)

Back-end

- **Open Social Data Store and Single Sign-On:** Underlying database and blockchain used by D-CENT (D5.3)
- **Strong authentication:** How to login and authenticate securely (D5.3)

- **Groups and Access control:** How to define groups with access control rights (D5.3)
- **Single sign-on:** Support for login across D-CENT nodes and with Twitter/Facebook/G+ (D5.4)
- **Identity Management:** Each user will have profile and a private key associated with them. (D5.3)
- **Secure Messaging:** Users can send encrypted messages to each other (D5.4)
- **Federation:** Sharing data across different D-CENT nodes (D5.4)
- **Data Portability:** Allow users to own and export their data (D5.4)

D-CENT Applications

- **Open Data:** Integration of CKAN into D-CENT for Open Data (D5.2)
- **Crowdmapping** (D5.2): How to map resources and structured data

Given that this is a large amount of features, their connections and interdependencies are given in Figure 40. For each feature, its ability to either rely on data (ingoing arrow) or output data (outgoing arrow) to another feature is given. The frontend (done in HTML and Javascript) is given in the top box and described in the previous Section [Front-end](#), with the Docker-based backend features described in Section [Docker](#). Connections to external services, including both D-CENT enabled applications such as CKAN as well as third-party service such as Twitter, are given as arrows to circles outside both the HTML frontend and the Docker-based backend.

Docker Back-end and App Directory

Given that the focus of D-CENT is to create a lightweight platform making it easy for activists and governments to launch web apps to empower citizens, the D-CENT platform should be easy-to-install on different kinds of servers, even if the groups that want to use it have a minimum amount of system administration expertise. While the D-CENT frontend is a straightforward mixture of HTML and Javascript, the D-CENT backend has many dependencies in terms of libraries and may also encompass apps written in many different programming languages, ranging from Ruby on Rails (Your Priorities) to DemocracyOS (Javascript Node.js). One of the biggest barrier to wider deployment of open source civic technology with activist groups, government and others is that they are hard to install usually requiring expert knowledge of programming language, although at the same time there is strong desire to use free open source software where possible with both many organizations and activists.

The solution best solves how to package and install in a uniform manner a wide variety of applications and libraries is Docker. Here is a description of Docker from the www.docker.io website: “Docker is an open-source engine that automates the deployment of any application as a lightweight, portable, self-sufficient container that will run virtually anywhere.” A container is a synonym for a specialized virtual machine that can run many different services. The description continues that “Docker makes it easy to build, modify, publish, search, and run containers. The diagram below should give you a good sense of the Docker basics. With Docker, a container comprises both an application and all of its dependencies. Containers can either be created manually or, if a source code repository contains a DockerFile, automatically.” As illustrated in Figure 40, Docker allows a file called a Dockfile that contains a list of libraries and runtime environment needed for particular applications. By installing a Docker, the Dockfile

is read automatically in order to build the correct container, saving time and expertise. Dockerfiles define some basic interfaces to be compatible with D-CENT. Interfaces include:

- Docker log interface for common logging of errors (done in standard Unix fashion)
- Single-Sign On for login (more detail on SSO in Section [Authentication](#))
- D-CENT node wide notification (more detail on the notification API in Section [Notifications](#))
- ActivityStreams for activity streams and notifications (See details in Section [Activity Stream](#))
- Common localization interface for multi-lingual usage of D-CENT done using common design patterns

The Docker-based solution also allows additional applications to be installed that build off of the user database and other core features of D-CENT.

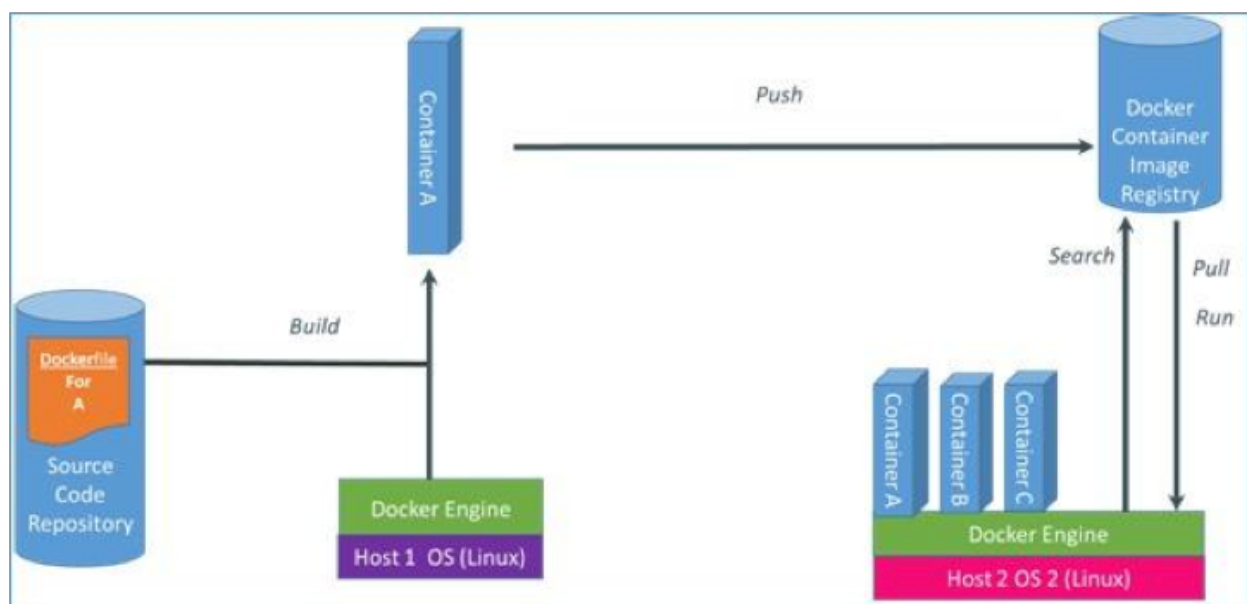


Figure 40 Illustration of how Docker works

D-CENT Core Features

For each D-CENT core feature, we will name the feature, provide if needed a more full feature definition, and explain which concrete user need it fulfills. Then we will give a description that includes an example flow, with if viewed as useful a user story in italics. Lastly, each feature will also have open decisions (based off of input from D4.1 and D4.2) that will be determined during the implementation phase in Work Package 5.

1. Social Data Store

Feature: Social data store that can act as collective memory of a D-CENT node

Feature Description: The social data store is the set of core databases that holds the data a given D-CENT node. The social data store is given as a transparent and decentralized database or cryptographic ledger that can be used for storing data in an append-only fashion, while user's personal data and other system data such as activitystreams is stored using more traditional SQL and NoSQL databases respectively.

User Need: As a citizen I want to make sure that privacy is granted for all my personal data but I also want to share my data with others in my group and make sure the recording of democratic decisions and actions can serve as a collective memory for my group. In detail, an append-only cryptographic ledger can fulfill the following user needs

- Secure authentication via the use of cryptographic identity
- Secure transactions in an open environment
- Reliable voting system for individual votes and group decisions, where the cryptographic ledger could be used to record votes in a secure and authenticated manner (i.e. via digital signature).

Description: Due to the nature of D-CENT, there are three distinct kinds of databases deployed by a given D-CENT node, each with their own distinctive advantages and drawbacks, as well as their own interface to open data such as CKAN (as explained in Section Open Data) and Linked Data in the W3C Resource Description Framework (RDF) (www.w3.org/RDF/). These databases are described below. Since the databases are used in almost every transaction, no user story is given.

- 1) **Cryptographic Ledger:** For some kinds of data, there needs to be high authenticity and a permanent record, i.e. data related voting, or recording group decision, and this kind of data will be stored on a cryptographic ledger (blockchain) that provides the highest level of security against fraud or tampering due to its keeping an append-only record of all transactions. Thus, the blockchain serves a permanent collective memory for D-CENT groups. For each D-CENT node, there will a cryptographic ledger for each group of users. Thus, a single D-CENT node could

contain many different blockchains or simply use a single blockchain with prefixing or coloring to “subdivide” a single blockchain between groups. This will be further explored in D4.4 and decided in WP5. For details on how a blockchain works, see Section Blockchain Mechanisms. Blockchains in general will have to be exposed using custom wrappers that export the data to a data-format that can be shared such as RDF. More background information on blockchain are provided below since many readers may not be familiar with it.

- 2) **SQL Database:** Data that is structured and dynamic, such as personal data related to user profiles, will be stored in a more traditional SQL database systems that provide a lower level of authentication in terms of cryptography but which are much more scalable and efficient. There are large efficiency costs for using a blockchain for dynamic data and the fact that common Web application development frameworks do not easily interact with a blockchain. Also, complex queries can be built and run as SQL queries. Since SQL databases are highly scalable and the schema of data that will be stored will be repeated (although with different value) across all users and groups for each D-CENT node, every D-CENT node will have a SQL database with multiple tables for storing values. It is using the SQL database for their personal data that user also has the right to be forgotten (to remove all personal data from a network or a node) and to import/export data. If needed for particular kinds of data in the SQL database, these kinds of databases are easily exportable to an open datastore such as CKAN (See Section Open Data for details). Also, for particular data the W3C R2RML standard (<http://www.w3.org/TR/2012/REC-r2rml-20120927/>) provides an excellent way to expose data as Linked Data. Given that SQL databases are quite standard and well-understood, no further details are given in this deliverable.
- 3) **NoSQL Database:** A “NoSQL” database will exist for each group in a D-CENT node that can store generic as key-values input from open-ended JSON datastructures such as data in the W3C ActivityStreams standard as well as other more ephemeral data generated by Javascript. Traditionally SQL databases require structured schema, and given the open-ended nature of D-CENT it is of too high of a cost for the database schema to changed for every modification for the schema. Much of the data stored in the NoSQL database will be in the W3C ActivityStream standard. Currently, there are a wide variety of incompatible “NoSQL” databases as detailed in D4.2. However, all that is required is for the underlying NoSQL database to store JSON documents without modification, as well as media: attachments, vide and images. Given that ActivityStreams is also based on a mapping of JSON-LD to RDF (See D4.1 and D4.2 for details), it can then be also imported to a native triplestore if necessary for exposure to RDF. Sample RDF schema for the NoSQL database is provided below.

Cryptographic Ledger Details

The “blockchain” is a term used to refer to a distributed append-only cryptographic ledger that is shared amongst all nodes participating in the network and that records information in a secure and irreversible manner. The blockchain differs from traditional database in that it is (1) decentralized, (2) append-only, (3) cryptographically secured. The blockchain resolves the [Byzantine fault tolerance](#)’s problem found in traditional distributed systems (see e.g. the Byzantine Generals Problem) by means of a probabilistic approach: while it cannot prevent a potential attacker from communicating false information to some

part of the network, the probability for such an attack to be discovered increases over time, until it becomes practically impossible for it not to be detected (Antonopoulos, 2014). The integrity of the network is achieved by means of a cryptographic mechanism based on the concept of Proof of Work (POW). All blocks in the blockchain incorporate a short string of meaningless data (a *nonce*) which is used to validate the block. For a transaction to be considered valid within the network, it needs to come with a particular nonce that satisfies an arbitrary set of conditions. In the case of Bitcoin, for instance, the condition is that the SHA-256 hash of the block (including the nonce) results into a string with a predefined number of leading zeros (whose quantity varies according to the difficulty level that is being sought). The hash of all valid transactions are aggregated into a series of 'blocks' that includes into their headers a particular string that represents a hash of the hashes of all the transactions in the block, the so-called *Merkle root* (Nakamoto, 2008). This is useful to ensure the integrity of all the transactions belonging to that particular tree branch, since adding, removing or altering any one of these transactions would always and necessarily generate a different hash to be included into the parent node (the *Merkle root*) -- which will, in turn, requires its the higher parent node's hash to be changing as well, as so on until the root of the tree. Although useful to secure the integrity of the network, such a mechanism obviously comes with a series of drawbacks, most notably in terms of efficiency and scalability. Yet, it is possible to deploy a more light-weight implementation that does not verify the whole blockchain, but relies instead on a series of 'trusted nodes' that will be in charge of verifying the integrity of the blockchain in place of the light-weight client. Instead of downloading the whole blockchain, the client only needs to download the blockchain headers from the trusted node, so as to subsequently be able to retrieve only those transactions that match their particular request (along with the Merkle tree branch that refers to the block in which the transactions actually appear). For D-CENT, we can assume that D-CENT nodes will store the blockchain in the server and that groups may be allowed to make, merge and share decisions in the same way blockchains are verified via trusted nodes. Yet nothing prevents any user to keep a copy of the blockchain at home, if capable of doing so by following some technical instructions on how to set that up. In general, D-CENT can use just the blockchain as a cryptographic ledger, but without the "proof-of-work" concept needed by Bitcoin, although we can continue to use the Merkle tree method for verifying transactions.

NoSQL database schemas

As many different kinds of data can be stored in a NoSQL database and the structure can be changed dynamically (unlike in a SQL database where the data schema is pre-defined). Thus, it is important for the various documents stored in the NoSQL database to have their own schemas defined. RDF Schema provide one efficient way to type these documents, where the names of the types are mapped to URIs using a modified JSON format called JSON-LD (json-ld.org/) that has an additional "@context" element that points to a URL that serves as the base "namespace" of the schema used, although unlike traditional database schemas these RDF (Resource Description Framework) schemas may be dynamically modified and changed without changing the underlying database. A logical diagram of some of the various RDF schemas and APIs are given below in Figure 41, as well in list form in Figure 42.

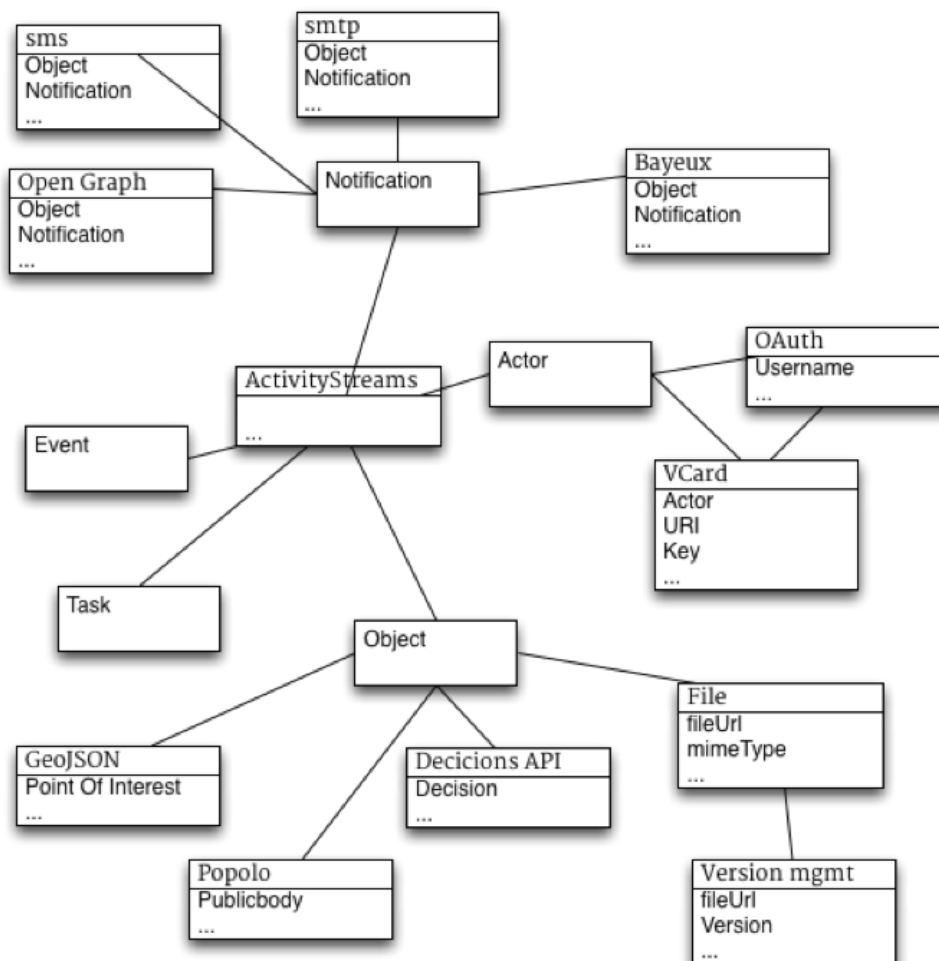


Figure 41 Logical Diagram of Schemas for a D-CENT node

Schema	Use	Source
Popolo	Organisation structure.	http://popoloproject.com/specs/organization.html
Decisions API	Organisation decisions.	http://dev.hel.fi/apis/openahjo
ActivityStreams 2.0	Objects, Events, Tasks, Notifications, Actors and their relations.	http://activitystrea.ms/
VCard 4.0[1]	Personal profile data.	http://www.w3.org/TR/vcard-rdf/
Bayeux	Push notifications.	http://svn.cometd.org/trunk/bayeux/bayeux.html#toc_9

GeoJSON	Geolocations.	http://activitystrea.ms/
Open Graph	Sharing “likes” to Twitter, Facebook and Instagram.	http://ogp.me/
http URI	Attachment file access.	Attachments are stored as Git Blob Objects with unique entry URI .
http URI	File version management.	File versions stored by their 40-digit SHA key as File URI .

Figure 42 RDF schemas to be used initially in a D-CENT node

Open Decisions

Currently, the main default SQL database for most WebApplications is PostgreSQL (www.postgresql.org), so we will continue to use this for any SQL-based work. This database easily integrates into all modern Web application frameworks.

For the NoSQL database, there is a wide variety of databases. DemocracyOS is currently using MongoDB (www.mongodb.org), which allows documents to be stored easily. Redis is considered similar and often faster as all data is stored in memory (redis.io). Very similar to MongoDB but programmed with a different set of user-cases in mind is CouchDB (couchdb.apache.org), which has some less features and is a bit less mature than MongoDB, but allows automatic synchronization of distributed document stores in an efficient manner. The precise database for single nodes will likely to continue to be MongoDB for the time being, but we can explore using native RDF triplestores such as 4Store (4store.org) as ActivityStreams 2.0 gets implemented in D5.3 and CouchDB as federation gets implemented later in Workpackage 5.

Ethereum and Bitcoin are currently two main candidates for the implementation of a cryptographic append-only public ledger. Both have been implemented as open source software, which can be freely forked and deployed as an alternative blockchain. In the case of Bitcoin, BitcoinD is a program that implements the Bitcoin protocol for command line and remote procedure call (RPC) use. Libbitcoin (<http://libbitcoin.dyne.org/>) is a sophisticated library that provides the necessary tools to create a scalable and configurable architecture, capable of interacting with the blockchain. An alternative is BTCD (<https://github.com/conformal/btcd>) which provides a full node bitcoin implementation written in Go (golang). An even simpler alternative is available through Bitcoinj, a library for working with Bitcoin protocol, which implements optimised lightweight simplified payment verification (SPV) mode. In this mode, only a small part of the block chain is downloaded, making bitcoinj suitable for usage on constrained devices like smartphones or cheap virtual private servers. In the case of Ethereum, interaction with the blockchain can also be easily achieved via the Node.js implementation, available at: <https://www.npmjs.org/package/ethereum>.

2. Cryptographic components

Introduction to cryptography

D-CENT, being a decentralized system, requires some kind of authentication and authorization protocol to exist technically between its various components. Otherwise, arbitrary groups or people could join a D-CENT enabled node and use it to spam the entire network (See definition of “Sybil” attack in D4.1). The essential building block of D-CENT’s approach to decentralization is to rely on cryptography. Cryptography relies on mathematical algorithms to encrypt and decrypt data in such a way that it can be securely transmitted through insecure channel without being read by unauthorised parties. At the most basic level, cryptography relies on a particular mathematical function - or *cipher* - that can be used to encrypt or decrypt a message with a key. The security of the encryption increases with the complexity of the cipher and the length or secrecy of the key. The main problem with conventional ciphers is that the key need to be communicated to begin with. If the communication channel is not secure, then some malicious third party could intercept the key, and then use it to decrypt the encrypted communications. Public-Private key encryption resolve this problem by means of an asymmetric encryption scheme. Instead of both parties relying on a single encryption key, each parties holds a pair of two keys: one public key, used to encrypt a message, and one private key, used to decrypt a message. This asymmetric scheme makes it possible for parties to communicate securely through an insecure channel, by simply sharing their public keys but without ever exchanging their private keys. In order to ensure the authenticity of online communications, the message can be also signed it with the private key of the party issuing it, so that the recipient can verify both the source and the integrity of the message. Below is an illustration of the basic fundamentals of Public-Private key cryptography.

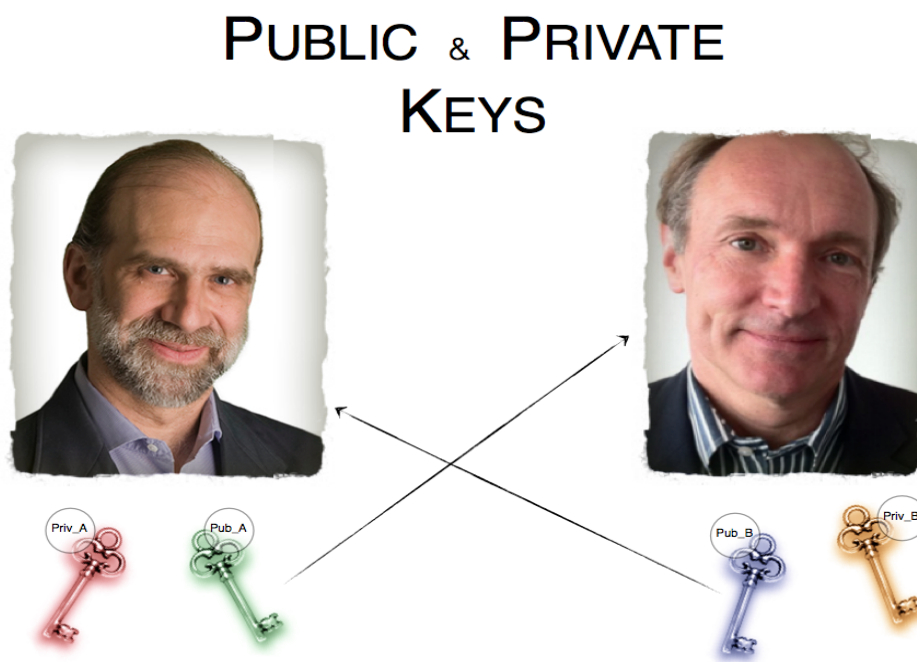


Figure 43 Private-Public Key Encryption, Step 1: Public Key exchange

Bruce loves cheese, but he does not want others to know. Yet, he needs to communicate to Tim that he would like Berners-Lee to bring him some Reblochon from France back to Cambridge. Bruce decides to rely on public-private key encryption, in order to send a secret message to Tim without anyone else from finding out about his cheese fetish. He retrieves Tim's public key (Pub_B) from a public key server, or perhaps he directly gets it from Tim – who can also get Bruce's public key in order to have two-way communication (Pub_A).



Figure 44 Private-Public Key Encryption, Step 2: Encryption & Decryption

Bruce encrypts his secret message with Tim's public key (Pub_B) and signs it with his private key (Priv_A) so that Tim can be sure that the message actually has been issued by Bruce. He sends the resulting encrypted message to Tim, who can then decrypt the message by using his own private key (Priv_B). In order to answer to Bruce, Tim then just needs to encrypt his own message with Bruce's public key (Pub_A) and sign it with his own private key (Priv_B) for the purposes of authentication.

Note that a message must be signed in order to be authenticated, but messages can be signed where the content of the message is not hidden. This allows authentication of the content of the message without hiding the message itself. D-CENT uses digital signatures with keys as the primary building block of its identity, authentication, and messaging systems. However, at this stage we do not specify many of the details, such as key revocation, key ratcheting, and even what particular cipher-suite should be used insofar as many of these aspects have implementation-dependent performance factors. For example, standard asymmetric public key cryptography is done with the RSA algorithm. However, in order to obtain higher performance on mobile devices, often a new kind of cryptography that is less widely studied and supported, Elliptic Curve Cryptography, could be used as it also supports a public and private part of a key. Yet implementation widely varies across platforms, with RSA being the de-facto standard and ECC still often unimplemented, and there has also recently been some discussion post-Snowden on new alternative ECC curves (<http://safecurves.cr.yp.to/>). Thus, at this point D-CENT will

not recommend any particular method, but will use whatever methods end up being standardized (to be completed by end of 2014) by the W3C Web Cryptography Working Group across all browsers, given the performance constraints discovered in implementation in Workpackage 5. The precise cryptographic algorithm used for the public key is still undecided officially, but strongly recommended to be a high-number RSA key, despite the computational complexity. Note that for a given system that the signature only has to be checked if the message comes from an external D-CENT node or if the internal system was worried to be compromised. For the time being, a 256-bit key may be appropriate, but given the increased in computing power envisioned over the next decade, 512-bit RSA keys may be needed.

3. Strong authentication and Single-Sign On

Feature: Authenticating to a D-CENT node

Feature Definition: The user of a D-CENT node needs to be able to associate their “real-world” identity with a username in a D-CENT node, in a way that lets these names be interoperable in a decentralized manner and utilize the highest security possible. A user can use various parts of a D-CENT node without having to re-authenticate and should be to sign-on using a well-known third-party service such as Twitter.

User Need: A user needs to verify their identity to a D-CENT node in a way that protects their privacy and uses the best possible security. This is especially important given the amount of “hacks” of systems where usernames and passwords are being stolen, as well as for high-value use-cases such as e-voting or for people whose account compromise may put them in danger, such as human rights activists. For low-security situations, users may just want to sign-in and authenticate using Facebook, Twitter, or Google logins.

Description: In almost all Web applications, a username must authenticate to a server. Authentication is just the process of connecting a username to a user, and then verifying that connection. Typically, this is done with a password. Today, many different methods of authentication can be used. For example, in some countries a SIM card can be used to authenticate directly, or in some others with an eID card, and these methods are generally more secure than a password as they rely on physical proof-of-possession of some sort of device. Sometimes, techniques like these combined or lower-security techniques such as the sending of a SMS code to a user can be used in combination with passwords, called “multi-factor authentication.” However, given that authenticating using these high-security methods is not yet standardized (although the W3C Web Cryptography Working Group is currently under a process of re-chartering to take them into account) we will focus on password-based authentication. In addition, D-CENT nodes can allow users to use 3rd party two-factor authentication using OATH standards compliant one-time passcode solutions, e.g. Google Authenticator (<http://code.google.com/p/google-authenticator/>). We assume that users don’t care about sign-in, they just want to access a service as quickly as possible. It is well-known from anecdotes from companies that forcing users to register a new account and sign-in leads to almost half of users being so discouraged that they do not even bother to register a new account. Thus, we must make sure our system is compatible with Facebook, Google, and Twitter accounts. However, this is done via re-direction and OAuth (See D4.1). Thus, we will not describe single sign-on using OAuth in detail, since it is now a common paradigm in all modern Web applications and using simply requires using techniques such as OmniAuth (<http://intridea.github.io/omniauth/>).

- Username: The identifier used by the user in the system. We are assuming this will take
- Client: The browser of the user.
- User: The human user that needs to prove they are associated with a username in the system.

- Password: A human memorable word or phrase that verifies the user owns the username.
- Ephemeral Key: A cryptographic keypair that lasts only one session.
- Server: The server that the user is verifying against.
- Verifier Database: The database that verifies the password stored on the server.

A user Joana wants to login into a D-CENT node without revealing his Facebook or Twitter account. She goes to the website www.d-cent.example.org to login with her username “Joana”. A username and password login box is displayed.

1. The user visits the domain of a D-CENT node. The user must enter their user-name and password. If the user has already registered, then skip to Step 3. Otherwise, the user must register a password with the D-CENT node. The user's password must not be stored directly on the server as the server database may be compromised. Therefore, from here we follow the SRP protocol.

A user Joana has not registered before, so she provides a password to her client in the browser-window.

2. The password is kept on the client side, but a cryptographic verifier is generated that can be stored on the server that can not be used to “guess” the password (unlike standard MD5-hashing of passwords in contemporary databases.) The client, using the W3C Cryptography API (<http://www.w3.org/TR/WebCryptoAPI/>), simply creates a “salt” (random data) that it then inputs with the password into a “one-way” hash function. At this time, we would recommend SHA-256. The password is then removed from the client and not sent to the server, although a copy of the salt is kept on the client. On the D-CENT server, a cryptographic verifier is taken by using a generator of a multiplicative group and then raising this the exponent specified by the result of the hash function. This verifier, from which the original password cannot be generated, is then stored in the verifier database of the server along with the original salt. This process is explained mathematically in the IETF standard (<https://www.ietf.org/rfc/rfc2945.txt>).

The user Joana types in their already-established password in the browser window to get authenticated to the D-CENT server.

3. The password is used again to generate a cryptographic verifier. In particular, it now establishes an ephemeral key between the client and the server. An ephemeral key is generated by a client and sent over with the username. The server then uses the verifier and sends an ephemeral key back to the client. Then using the user's password, the keys are re-established and strongly authenticated. This is considered to be equivalent to a zero-knowledge proof and requires no third-parties (<http://www.cs.rit.edu/~jjk8346/paper.pdf>).

Once authenticated, Joana can access D-CENT services such as decision-making capabilities, draft documents, and see other users in her groups.

4. See Section 4. In general, the ephemeral key established by SRP is quite useful and can be stored on the client-side using W3C Web Cryptography and marked as non-exportable. As long as the server is trusted and there is no successful attacks on the Javascript, the server can then use this

key in the same origin to generate a signing key necessary for the production of capabilities (See D4.2 and the Section on Groups and Access Control).

Dependencies

Currently, D-CENT nodes simply use user-name and passwords, so implementation work is required to update the spec. In particular, there are existing SRP protocol implementations in Javascript:

- srp-client SRP-6a implementation in Javascript (compatible with RFC 5054), open source, MPL licensed.
- The JavaScript Crypto Library includes a JavaScript implementation of the SRP protocol, open source, BSD licensed.

However, these implementations do not use the W3C Web Cryptography API for securing the pseudo-random number generation or cryptographic functions, and so these libraries will need to be reconverted to use them, which is a substantial amount of work. However, without these, it is unlikely that the key generation or cryptographic generations are correct.

In terms of single-sign on, there are many available libraries. We recommend at this time using OmniAuth, which is available for many programming languages: <http://intridea.github.io/omniauth/>.

Open Decisions

The storage of the private key in the long-term is difficult, as Web browser storage currently only supports per-session storage. Thus, recommendation is that when a user authenticates to a D-CENT store for the first time, the private key is generated and put on the server in a wrapped form. In future work, if the W3C standardizes it, we may have access to long-term key storage on the client via smartcards, SIMs, and other eID schemes (<http://www.w3.org/2012/webcrypto/webcrypto-next-workshop/report.html>).

We would eventually want to upgrade our protocol to use authenticators outside of username and passwords, such as eID cards and multi-factor authentication. Note that currently this is not compatible with the existing Web Cryptography API, but is currently part of the re-chartering effort. When this re-chartering is done, we expect there should be relatively straightforward access to strongly secured key storage and cryptographic functions in hardware tokens, which can be used rather than a password in combination with the salt and hash function to generate the cryptographic verifier needed by the server.

4. Identity Management

Feature: Defining identity via binding a username on D-CENT with cryptographic key material that they can use to authenticate messages in D-CENT.

Feature Definition: In digital information systems, digital signatures are necessary to prove that a message came from a given user or community and can't be repudiated at a later time. This is especially important for use-cases such as e-voting, but in general is needed in federated and decentralized systems to prevent impersonation of users, spammers, and sybil attacks (See D4.1).

User Need: A user would like to make sure that they know that a message they sent can be verified as sent by them and through their D-CENT node. This means that they have not been impersonated and a “chain of provenance” allows their messages to be traced authoritatively to them.

Description: In general, the only way to prove that a message came from a given user or D-CENT node is to use digital signatures over the message. We want to provide the following three cryptographic properties for every message.

Authentication: Can the receiver know that the message originates from the sender?

Integrity: Can the receiver know that the message has not been accidentally modified?

Non-repudiation (source authentication): If the receiver sends the message and the proof to some verifying service, the receiver can be verified that the message originated from the sender.

These properties are only provided by digital signatures provided by asymmetric keys (not MAC or simple hash functions). In order to do that, we need to attach some kind of asymmetric cryptographic key to the user. Since key management is typically outside of the technical proficiency of most users, we will instead focus on having the server manage their keys for them, but trying to make sure that a private key is kept on the user's client as can be generated by a key-derivation function from their login password (See Section on Strong Authentication and Single Sign-On). In this way, we consider the primary issue of identity management the attaching of a public key to a user (with a private key accessible via a password or other authenticator on their client) in a way that can be discovered and verified in a decentralized system. For discovery of a public key given a *user@domain*, we have chosen the WebFinger Protocol (<http://tools.ietf.org/html/rfc7033>).

A large problem facing any approach with asymmetric keys on the Web is that the Web browser does not have safe-long term key storage for the private key material that matches the public key material stored on the D-CENT server, although this may be addressed by the upcoming W3C Web Cryptography Working Group re-chartering. Thus, if a private-public keypair were created when a user registers with a D-CENT node, the private key material would be destroyed when the session ends. What we want is for the user to have a long-term private key for signing their messages and other data on D-CENT. Although there are some disadvantages to this approach (such as the loss of perfect forward secrecy), these are not strictly necessary as D-CENT messages are internally not to be considered confidential to a group, but merely verifiable. For uses of confidential messages between D-

CENT nodes, see the section on Secure Messaging. In this case, the long-term key can be “wrapped” (i.e. encrypted) with the key generated from their password locally using SRP. Since the server does not know their password but only a cryptographic verified and a salt (and thus a SRP authentication is equivalent to a zero-knowledge proof). Every time a user logs back in, the private key can be re-established and downloaded to their local client, and there unwrapped for using in digitally signing messages.

Currently, our use-case is based in terms of an individual user. However, a user’s key can easily be replaced by a group’s key, although for the group to be discoverable it would also require its own separate username. Thus, we believe identity management for groups should be done in the same manner as individual uses. The main difference would be that access to the group’s authentication mechanism would require binding individuals to groups. While technically this is easily done via capabilities that prove a user belongs to a group, the actual social process where a group determines criteria for membership and some process (such as voting or consensus) for membership is beyond the scope of this section. D-CENT will not provide instructions, but simply provide a number of methods, such as the voting techniques detailed later in this deliverable. .

Digital Signature: A digital signature given using asymmetric cryptography to a message.

Identity Provider: A D-CENT server that serves usernames and data associated with that username, such as a user public key.

Well-known URI: A place to store data about a user or group.

User public key: The long-term public key of a user. May be a “group” public key.

User private key: The long-term private key of a user. May be a “group” key (See Section on Access Control for constructing group keys from user keys).

A user on one D-CENT node, Alicia, wants to check to see if a message and a corresponding vote sent another user, Bob, on a D-CENT node is true. They may be on separate D-CENT nodes. Alicia searches for the username of the user Bob, as perhaps gathered from seeing Bob in one of their access-controlled groups or having posted a message, or from a voting-log.

1. Each D-CENT node must keep a directory of users (and groups) with their public key pairs. This makes the D-CENT node serve as an identity provider. When a username is inquired at identity provider by a relying party (either internally or externally to a given), the D-CENT identity provider should first check the access control capability of that relying property. For a given username of the form user@d-cent.example.org, a WebFinger query is committed.
2. The WebFinger query is simply a HTTP GET with a constructed URL of the following form:

```
GET /.well-known/webfinger?
resource=acct%3Abob%40d-cent.example.com&
rel=http%3A%2F%2Fwebfinger.d-cent.example%2Frel%2Fpublic-key
HTTP/1.1
Host: d-cent.example.com
```

The response should be:

HTTP/1.1 200 OK

Access-Control-Allow-Origin: *
Content-Type: application/json

```
{
  "subject": "acct:bob@d-cent.example.com",
  "properties": {
    "http://d-cent.example.com/rel/public-key": "-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: W3CWebCrypto 1.0 (Mozilla Firefox)

mQGIBEQUpzkrBADvBhJc8zkbwMVv87maMMZ5aR0ePqbAu//TawIHGdnEbXQYojpY
dYdnTqccLr6PhP3Uz3rsv3mP5YjDskSldxA0UmUjuPUwsK6uIh6oPx73APNkt40z
lIcU8X8m77IAkQ6pQm7nGlitVIFDo6iqsuxw5A2W0m2MhPpvzmWIXxM9NwCgipw/
S1kjfgp6lxLT4nCsJbdbl/8D/0sR4dsy54JkyvcbvR/mnJjsi8M+pMUAhNgYXiYK
jwgseX9voGzvC9zP25CFFFKpRwSFlg0LjQBsp3VXRtrzpqlr3GTyz5befMTRC3vu
R5Bmocek2DjYpFRGiXE2GshObsaGxZL8BeVHw+ffAaEPR+MfKDLgUZhlIqxMeNrH
ERbIBADLsEgJkXcPm0Rslx6zPzkEbK5d/nVKrOE2ZyhLHBwIROQcuPDqSPOeD8
JWYgGc2mTbotYuPN/cf97K86i8tFl3vyZ39G8czevdiezvhlwHDijnGl8p5lInYQ
PDREiU0ESTalO/gIaKLGpJla6rzOWT2EzD94fP7qNgN9BuL2SLQsSGFycnkgSGFs
cGlulChMZSBMb3VwKSA8aGhhbHBpbkBPYmliGivLm9yZz6lXgQTEQIAHGUcrBTO
mQIbAwYLCQgHAwIDFQIDAxYCAQIeAQIXgAAKCRBPylWlaIlkv84AJ0cjQvYnKVA
XBOb947bnUA7hdMGsQCfdmyPxf2r3VQNBxjp8yITiIvWKVbK5Ag0ERBTOvhAIANjV
xiZ4PfHqQwacsy64oAYRv4wtG5zhl7VzgiqMlBq7qyD4DAbly304mfEEKcRYD73P
7qk8uo4JVC6d3eQoFfw699c5Q0TctsRJRH6+A8iNnRISbPfyX2wUhubTMs3y8I8D
nYhLZf0qxN4Zpa7MHRd6weydl2tSqcpsQcoQofUxv7FhofxhS9PFjnz7EuuMStu2
ln3dGujQdIkIEf7rVJ03IH+873EXd/Gi30UZ9drndQda8u/e9J+IAVIMpCgkUEp
dQItlMzXcD8y6kaA8kUh9TRmQqFIMZ/y8+ZPU6flnRk2d0LJZttPfte7oeOxdnOE
FtiHnzze0LEbNpC5q8cAAwcH/AjeYChQrZzg+VOyVvU8wrXfRlu5kHOaA4nRX7j9
zDxA/oCvxhKI0540B+0xRRtoFQ9F0R3Zcx9uegkFG0iK5Hdo/K5AQv/emanJsaXV
gwGSmXWlQUIYraYDVB6jVFqgXOZXkklfhNzTltnHNeOASyi0sU2/ml0Intz2t3R
xPJADLCPdAc7b6o4fQOWhhZQvVo70+N3BIBGPh/bWI86J58Y0aIHjYObYFDS+i
9BfMeYSFH+I4uL2PEvG0FnVm/wW2WNI2Mgx7fmf6yCjv9fslrg8At3jGuCqzR9QF
O+o4ek8K7CQGx+F+VgMC6s/gdGUqyrgynqtLNjwMC4/P0c9iISQQYEQIACUcrBTO
yglbDAAKCRBPylWlaIlkv84AJ0cjQvYnKVA
-----END PGP PUBLIC KEY BLOCK-----" }

}
```

Alicia then wishes to check to see if the message belonged to Bob. She then asks her local D-CENT node to check the message. As all messages from Bob are signed, the D-CENT node then shows that Bob did indeed send the message.

- Any of user's messages that are stored in the social data store locally or sent to other D-CENT nodes require their signature. Using the public key discovered, a user should be able to verify that indeed, the user did indeed send that message. This can be done via normal straightforward digital signature verification.

The digital signature on a message to be verified could resemble:

```
{
  "subject": "acct:bob@d-cent.example.com",
  "hash": "SHA1",
  "message": "Yes",
  "decision": "genID:4567",
  "version": "W3CWebCrypto 1.0 (Mozilla Firefox)",
  "signature": "iQIcBAEBAGAGBQJUIhBUAAoJEPg..."
}
```

Dependencies

There exists a client Node.js library for WebFinger in order to do the discovery of user-names using the Web Finger protocol. This library would need to be updated to support public-key discovery (<https://github.com/el4n/webfinger>). Also, each D-CENT server would have to add server-side support for creating at least one (but possibly multiple) well-known URIs with public key material for each of their users with the proper access control.

Open Decisions

A WebFinger-enabled URI that can serve their public-key should be set-up for public use. For different groups, the user may wish to use different keys and access control done as in the Section on Group and Access Control. In this case, the user's public key should be hidden from everyone except groups they participate in using WebFinger-enabled URIs using access control.

For re-establishing a private key, a key wrapping/unwrapping function is needed that requires the SRP password necessary for authentication in order to unwrap the key. The Mozilla work around this is a good starting point: <https://wiki.mozilla.org/Identity/Cryptoldeas/02-Recoverable-Keywrapping>.

5. Groups and Access Control

Feature: Controlling Access to content.

Feature Definition: A group of users must be able to make sure only authorized members can access content from their group. Thus, we need to deal with the establishment of groups and their usage as well as how they establish permissions. Access to different kinds of content may need to be dynamically changed.

User Need: A user in a D-CENT node may wish to restrict access of their content to only members of their group.

Description: Access-control is a difficult problem for an open-ended system. As studied in D4.1 and D4.2, traditional access control lists require the entire system knowing the number of users on a system and their permissions on each file, which would not scale to open-ended and federated systems like D-CENT. Thus, we have proposed using in D4.1 a “capabilities” based approach based similarly to OAuth, except rather than using a short-lived randomly generated string as a “shared secret,” we would use tokens with digital signatures that convey the possession of key material rather than simply a short-lived shared secret, which would thus be unforgeable. This could then be used to prove membership in a group and so give access to the group (via access to the group's private key) where the server could then allow anyone with access to the group's private key to send messages and control access to content from a given group. When a user joins a group, the server would add that group's private key material to their profile. When a user leaves a group, the server would simply have to delete a copy of the private group token from their profile that gives them access to the group's private key.

User: Person who belongs to one or more groups

Identity Claims: Information about a user required by a group.

Group: A group of users with certain access control over resources and capabilities to do actions.

Relying Party: A service that requires identity claims of the user. Groups are kinds of relying parties where “proof of membership in a group” is a necessary identity claim.

Group Key: A private key material possessed or possible by members of the group.

Capability: “Can be thought of as a ‘handle’ representing some operation on some resource. Possession of the capability is all that is required to perform that operation. <http://www.links.org/files/capabilities.pdf>

User Key: A private key that connects the user to an identifier in the system.

In essence, our solution is that possession of a private key material, as proven by a digital signature given on some token or message, is enough to prove both identity (if a user key) or in a group (if a group key). This proof in of itself is then enough to let someone in an area that requires some form of access control, as the digital signature's verification can prove the possession.

Content is always owned by whoever sets up a group, manages group or a page. Nodes have a owner organisation who have systems administrator access to the server, and admin role who can override security setting to gain privileged access to functionalities and data for troubleshooting purposes.

Anonymous: Can read open data activitystreams when an anonymous user attempts to accesses Reader functionality, they are prompted to login or register.

Reader: Can participate (i.e. comment and vote). A reader sees functionality for contributor. All users who register are readers, and can create groups, for which they will be contributor. Is contributor for their personal activitystream and contributor for open groups.

Contributor: Can create content for their groups. They can also invite Contributors for their groups.

Moderator: Can moderate any content in the group (delete the content if needed).

Note that unlike BrowserID and like OpenID Connect (See D4.1), the capability approach allows for the user not to be online, although having the user online approach would be superior for privacy-respecting use-cases since the user could be manually prompted if needed for any further authentication. However, the information flow itself is the standard OAuth information flow (See D4.1) with the user in the loop. This approach then generalizes for access to any third-party service enabled using OAuth, which is the standard authorization protocol for large services such as Twitter as well as government services such as those in the United Kingdom. The flow is illustrated in Figure 45 below:

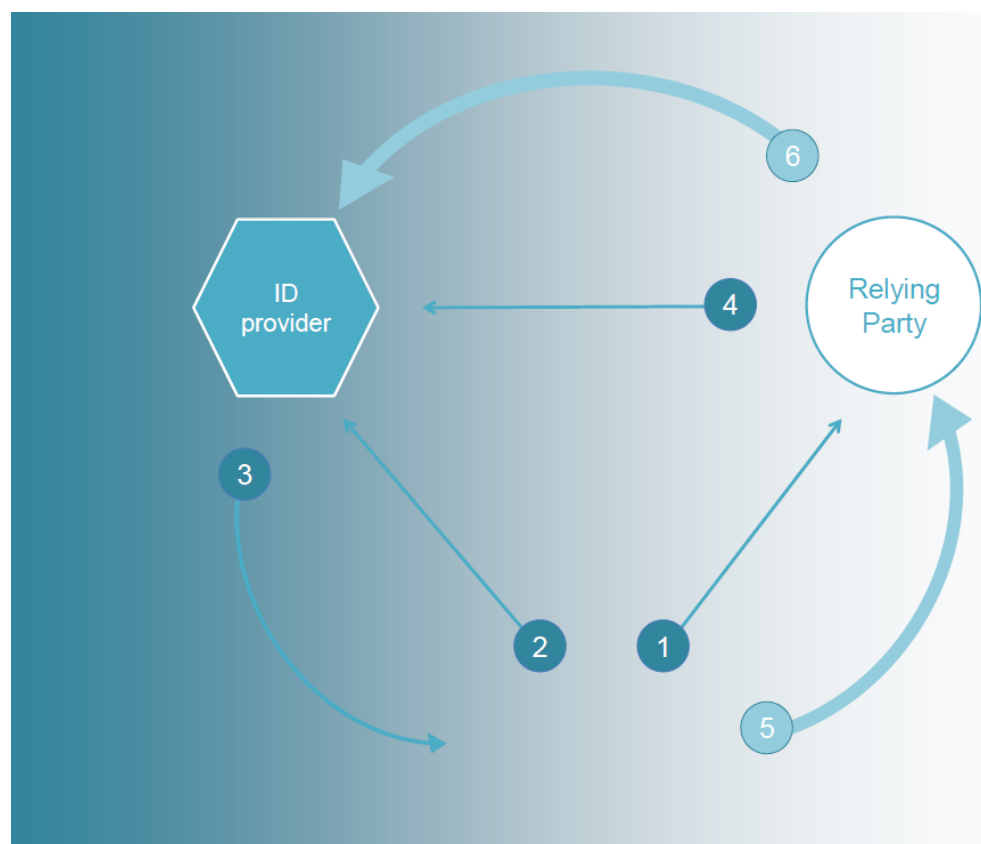


Figure 45 D-CENT Access Control Group Schema

Herbert wants to get involved in voting on a decision involving the construction of a dam in Iceland. However, he has to show he is a member of the activist group “Stop the Dam” before involving himself in group decisions like a consensus discussion on whether or not they should do a demonstration Sunday. Herbert just authenticates to the D-CENT node using his password and types in the name of the group he wants to join.

- A user attempts to authenticate to a group-controlled resource, and the D-CENT node requests a capability from the user.

Little known to Herbert, his browser in conjunction with the D-CENT server tries to figure out if there if he is authorized. If he isn't, his browser displays a messaging saying “Not Authorized. Please click here to apply to join the Stop the Dam Group.”

- The browser checks to see if that private key for the group already exists in the browser.
- If no private key is present, the browser requests a key for the group the identity provider using WebFinger. If the user is online and their level of privacy requirements, additional verification may be needed. If the key is not present, an error message is displayed to the user. The error message can be customized by the admin of the group.
- A signed token, signed with the private key of the group (and possibly, if needed also, the user key), is sent to D-CENT. With conventional signatures, the relying party (group) can check to the signature using the public key of the group or the public user key on the identity provider, which they can request if the group key and user-key if needed

If the request requires some kind of additional work (such as a manual double-check on Herbert's identity by visiting him in person or giving him a call) from the group administrator and Herbert goes offline, the access control can be fully handled by the server when Herbert is not there if the group allows this (i.e. for situations of low security).

- If the user is offline, the group (relying party) requests the capability from the identity provider. The identity provider sends the signed token to the relying party, which then the relying provider can verify using the public key material from the identity provider it has already received in the previous step.

If Herbert is online and has the right capabilities, he is let into the “Stop the Dam” group where he can participate in the discussions and the consensus vote, with others in the group being highly confident the person is him.

- If the user is online, the browser sends the signed token to the group (relying party) directly, which again the relying provider can verify using the public key material from the identity provider. For high security situations, the group may want to force the group key to come directly from the browser, requiring the user be online to use SRP-based authentication to retrieve the group key.

Dependencies

This flow again depends on the Web Cryptography API and the use of WebFinger to discover group keys. However, libraries for all of these currently exist, as detailed in the Authentication and Identity Management use-cases.

Open Decisions

There are many more complex ways to tackle this problem. In general, OAuth and capabilities are relatively straightforward, but one could imagine managing groups using a variety of cryptographic protocols to ensure that membership in a group with certain access-control capabilities came out of a proper decision-making process, including delegation of capabilities. For example, if a user wanted to delegate their capability to another user (as would be done in “proxy voting”), how could we cryptographically ensure that the delegation was authorized by the group? Shamir's Secret Sharing Scheme is one possibility for implementing this, as it allows (Shamir, 1979). Yet it is unclear for group keys if it Shamir's Secret Sharing Scheme is the best protocol to chose as there is much open cryptographic research in the field and multiple variations on group signatures and group authentication have been proposed in the literature (for example, Bellare et al. 2005). While we should re-visit these choices in implementation, starting simple with OAuth and capabilities and then scaling up makes the most sense.

6. Notification Engine

Feature: Sending Notifications

Feature Definition: Notifications allow a user to be sent a message via a number of different kinds of communication methods (e-mail, Android Push Notifications, and even possibly SMS and Twitter) when an event happens on a D-CENT node.

User Need: A user needs to know when an event happens relevant to their community's data. For example, when an edit is made to a document they are working on or when the community makes a decision about a proposal.

Description: Since we will have a wide variety of events coming in, we will need rich metadata format to trigger events and then a Notification API that will alert users via their preferred means of communication. The Notification API defines a standard API across D-CENT nodes for notifying users of events.

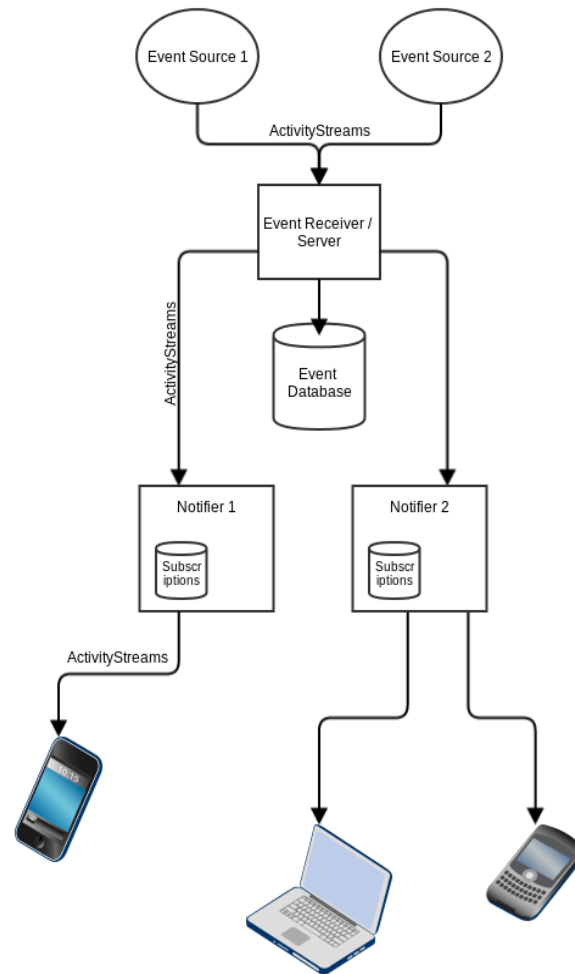


Figure 46 Illustration of D-CENT Notification Engine

A notification engine consists of the following components:

Server: External events are posted using the Notifier API to the server and are stored for later processing.

Actions: The actions that determine which users should be notified about what and how, and performs the actual messaging when an event has been resolved, but are built before the event is resolved.

Resolvers: Takes note of actions that need to have notifications sent and resolves whatever additional data or processing is needed for the notifications to be ready for sending. Every action that is resolved may not be executed if there is some external problem, such as Twitter being not working.

Executors: take actions that are ready to be sent as notifications, and for each one it build the final content for it and actually gets it sent using a particular transport (for example: email, Android push, iOS push, Twitter).

Essentially, the flow is as follows for notification inside D-CENT apps:

Jean clicks on “votes” on a D-CENT Node on particular bill around cycling lane going through the Finnish Parliament that he has affirmed. Jean in his user preferences sets up the fact that Jean would like a notification via email when the cycling lane bill is finalized and voted on, with the results of the vote.

- The API consists of triggers inside the application that are triggered when some action happens in the user interface. When an action happens in the user-interface, it calls the Notification API directly and posts an event using the Notifier API to the server, sending it as JSON structure to the server. The server receives this request and validates the sender (client app) through an access token (so the D-CENT nodes don't accept unauthenticated requests, i.e. “spam” or other malicious messaging).

```
{
  "event": "VotedAffirmed",
  "user": "jean@rocknroll.com",
  "document": "http://d-cent.example.org/CyclingPath"
}
```

The notification engine has recorded that Jean has voted affirming the cycling path, and so sets an action for this sending a message when there is a resolution to the bill,

- The server signals a signup event and then stores an action in the notifier. The notifier does not do real-time notifications as this would require people to be connected to the notifier server at all times, which would be unlikely. Since this is a valid event, the signup trigger comes into action. It creates and persists an `action` object.

This trigger for “voting yes and watch result” from the D-CENT node is stored in a notification engine. The resolver database looks up Jean’s profile and retrieves her first and last name, as well as the fact that her preferred language is English and her preferred method of message transport is e-mail (rather than a mobile push or a tweet).

- A resolver walks through the database of actions and attempts to resolve all data needed for execution (for example by retrieving additional data from the user profile) so that the resolver launches an executor based on the user preferences.

At some point, the Finnish parliament does the vote. This vote also triggers the notification engine and changes the state of the action from “ready” to “executed”.

- Each executor is transformation of the notification to a particular transport protocol to deliver the notification. A transport protocol could be email, Twitter “tweets”, or Facebook “status updates”. The executor checks the user's preference for preferred transport protocol or protocols to deliver the message. Once the executor is launched, the notification is sent to the user.

```
{
  id: 'send-welcome',
  data: {
    email: 'ozzy@blacksabbath.com',
    validateUrl: 'http://validation.url'
  }
  user: 'ozzy@blacksabbath.com',
  validateUrl: 'http://validation.url',
  resolvedAt: [timestamp]
}
```

An email is sent to Jean telling Jean that the cycling bill has passed Finnish Parliament and tells Jean that it gets 89 votes for “yes” and 13 votes against, it passes.

```
{
}
```

Dependencies

D-CENT application modules need to comply with these User roles. Modules may have application specific maintenance roles that are for local sysadmin use that do not apply here. Compatibility with data [permission roles in Activitystreams 2.0](#) needs to be maintained as possible, or the schema used in ActivityStreams will be extended.

Currently the notification engine is built in Javascript (Available here: <https://github.com/democracyos/notifier>) and sends out email as a transport protocol.

Open Decisions

There are some open questions. The first is whether or not each D-CENT node should have its own notification server. The second is whether or not it is more efficient to store the subscription database within the D-CENT node itself. Push protocol for Activitystreams clients should be implemented as a frontend Javascript. Other protocol such as Bayeux may help in the “push” activism.. Currently transport is provided for e-mail, Android Push, and iOS Push messages. Twitter and other transport protocols (including possibly even Facebook) need to be studied and if needed, added to the notification engine.

IBM Connections (probably most advanced product line, driving ActivityStreams and open data standards now inside W3C): https://www.youtube.com/watch?v=_brbqQulFxl

SAP Jam (absurd video but shows off some features):
<https://www.youtube.com/watch?v=he2dT7EF3Og>

Microsoft Yammer (more boring, but explains a lot):

<https://www.youtube.com/watch?v=b2X83XIsupA>

VMWare SocialCast (RoR, built on ActivityStreams)

<https://www.youtube.com/watch?v=aXRsbDRMtAM>

7. Voting & Deliberation

1. Voting systems

A voting system is any formal mechanism by which decisions are made by a group of people, wherein each person gets to provide information which is used for processing the collective result. Specifically, a voting system consists of four parts: a balloting method, a tallying method, a set of choices, and a set of external factors. Balloting methods are formal mechanisms by which each voter registers their vote. There are many types of balloting methods, but in the traditional literature on voting systems they are rarely treated separately from the rest of the system, which leads to much confusion. For instance, Single Transferable Vote and Borda Count are radically different methods, but for all intents and purposes their balloting method is identical. Tallying methods are frequently referred to as social welfare functions in the literature. They are mechanisms which take a set of ballots and calculate a result. Tallying methods have a large number of different properties, and often even small differences in implementation can have quite massive effects on results. In addition, there are a number of properties that cannot be held simultaneously, which is often a side effect of the balloting method. Each tallying method normally requires that ballots be collected by a particular balloting method, although some balloting methods are isomorphic. For instance, any ranked vote tallying method can accept ballots either from ranked ballots or a pair ballots.

The set of choices is the collection of options that are presented to the voter. These are normally determined in part by the nature of the vote, for instance whether it is a referendum, an election, or some other form of plebiscite. For some voting systems, information is included in the calculation of a result which does not originate from the ballots. These systems are rare, but it is necessary to keep track of all information inputs. These inputs are collectively referred to as external factors, the most common example being that in Single Transferable Vote systems, conditions can arise where resolution is only possible if a random selection is made. This randomness is an external factor. In the following pages, we shall look at balloting methods, tallying methods, and delegation, as well as briefly review other considerations, and make implementation recommendations.

2. Balloting methods

Balloting methods are how you collect information from the voters - in a way it's just a question of what the form looks like, that voters fill in. More generally, it specifies a data structure which is used as input to a tallying method. A ballot is any device used to cast votes in an election, and may be a piece of paper or a small ball used in secret voting. It was originally a small ball (see blackballing) used to record decisions made by voters. Each voter uses one ballot, and ballots are not shared. In the simplest elections, a ballot may be a simple scrap of paper on which each voter writes in the name of a candidate, but governmental elections use pre-printed to protect the secrecy of the votes. The voter casts his/her ballot in a box at a polling station. Electronic ballots are typically forms on a computer screen that show the options available and allow the voter to check boxes or arrange options, depending on the balloting method. Ballots methods have an impact both on what forms of tallying methods can be employed, and

how much information is collected from each voter. Balloting methods also have a psychological effect on the voter, in that different balloting forms can seem more or less complex, and make it varyingly easy to make informed decisions or vote strategically. For instance, it is difficult for a voter to get a sense of how strongly their preference for one option is being stated when using an approval ballot, and it is virtually impossible for a voter to calculate which option they have given strongest preference to on a pair ballot within any reasonable amount of time, without a computer.

Single-choice Balloting:

Choose one option

☐ A, B, C

☒ 1, 2, 3

☐ i, ii, iii

Also referred as the “Indiana” or “party-column ballot”. The voter can cast a vote for one of the choices made available on a ballot with a single mark. Only one choice is allowed per question. Frequently this form of balloting is used in elections to allow a voter to choose all candidates from a party list by crossing that party’s box. In those cases, sometimes modifications are allowed, such as striking out one of the candidates.

Multi-choice balloting

Role 1:

☐ A

☒ B

☐ C

Role 2:

☐ 1

☐ 2

☒ 3

Role 3:

☐ i

☒ ii

☐ iii

Also referred as the “Massachusetts” or “office-block-ballot”, multi-choice balloting is essentially multiple single-choice ballots. As such, a single-choice ballot can be considered a multi-choice ballot with only one question being asked. The voter is required to indicate each candidate that he wishes to support. This requires more effort on the part of the voter (e.g. if there are 20 offices to be filled, 20 marks are required). A common problem with this method is so-called “roll-off”, which is when voters become less likely to cast votes in the latter questions of the ballot. In addition, voter fatigue is more common, which results in higher propensity to abstain from voting at all. In some cases, where multi-choice ballots are used, ticket-splitting occurs, whereby choices in each question are made irrespective of the other questions. While this is ostensibly good behaviour, it can lead to political problems in situations where the election results in people with no common political ground are elected into roles which rely heavily on collaboration between them. This is similar to a hung parliament situation.

Ranked balloting

Rank the options 1-3:

[3] A

[1] B

[2] C

Ranked balloting is a system whereby voters rank choices on an ordinal scale. In different variations, voters may only be allowed to rank a certain number of choices (such as in Alternative Vote), or may rank however many or few as they see fit. In most cases ranked balloting goes from 1..n, with 1 being the primary choice, but in the case of Borda count, the highest number is the preference. For instance, in the Eurovision Song Contest, ranked balloting is used with a Borda count, and the countries assign points to candidates: 12, 10, 8, 7, 6, 5, 4, 3, 2, 1.

Pair balloting

Out of these pairs, which do you prefer (mark one per line):

[X] A [] B

[] B [X] C

[X] C [] A

Ranked list ballot is isomorphic to a pair voting ballot, but the two promote very different voter behavior. Pair selection gathers exactly the same information as ranked selection. However, pair selection requires more effort from the voter (as the number of pairs of candidates grows exponentially with the number of candidates). It can also be quite difficult for a voter to know, during the selection process, which options they are giving preference to. Therefore pair selection has a different effect on people, psychologically, than ranked selection.

Approval balloting (range-voting)

Give each option an approval rating between 1 and 10:

[4] A

[7] B

[2] C

Voters rate each option separately on a scale. All candidates should be rated. Votes give each candidate a score, the scores are added or averaged, and the candidate with the highest total is elected. Range voting allows voters to express preferences of varying strengths.

Range voting satisfies the monotonicity criterion, i.e. raising your vote's score for a candidate can never hurt their chances of winning, and lowering it can never help their chances. Also, range voting satisfies the participation criterion, i.e. casting a sincere vote can never result in a worse election winner (from your point of view) than if you had simply abstained from voting.

The range voting concept has been used in non-political contexts also. Sports such as gymnastics rate competitors on a numeric scale, although the fact that judges' ratings are public makes it less likely for them to engage in blatant tactical voting. Range voting is common for things where there is no single

winner: for instance on the Web, sites allow users to rate items such as movies (Internet Movie Database), comments, recipes, and many other things.

Advantages of Approval balloting, as per Brams (1993):

1. It gives voters more flexible options. They can do everything they can under plurality voting (PV)—vote for a single favorite—but if they have no strong preference for one candidate, they can express this fact by voting for all candidates they find acceptable. I
2. It helps elect the strongest candidate. Today the candidate supported by the largest minority often wins, or at least makes the runoff if there is one. Under AV, by contrast, the candidate with the greatest overall support will generally win
3. It reduces incentives for negative campaigning: likely to cut down on negative campaigning, because candidates will have an incentive to try to broaden their appeals by reaching out for approval to voters who might have a different first choice
4. It increase voter turnout. By being better able to express their preferences, voters are more likely to vote in the first place. Voters who think they might be wasting their votes, or who cannot decide which of several candidates best represents their views, will not have to despair about making a choice
5. It gives minority candidates their proper due: Because AV allows these supporters to vote for both candidates, they will not be tempted to desert the one who is weak in the polls, as under PV. Hence, minority candidates will receive their true level of support under AV, even if they cannot win.
6. It is eminently practicable. Unlike more complicated ranking systems, which suffer from a variety of theoretical as well as practical defects, AV is simple for voters to understand and use.

3. Tallying methods

A tallying method calculates a result from the vote given a set of ballots. Tallying methods normally only take into account the voter's input, but sometimes also includes external factors, as discussed above.

Tallying methods have a number of different properties that are important, and each method has behind it a wealth of research into relative pros and cons. Long story short, there is no “best” method, there are only different properties. At the end of the day, the choice of which method gets used comes down to the values of the people who choose them.

In this review, we will only cover a small sampling of possible methods organized into categories, and then we will make recommendations for implementation for each major category of methods, as different methods are useful for different situations. The categories we use depend on two properties: how many winners there are (single or multi), and whether a clear winner is required, or simply scores/orderings (order, score, or winner). In the case of single-winner systems, order and score are meaningless, so we end up with four categories: single winner methods, multi-winner methods, ordering methods and scoring methods.

Single-winner methods

The simplest single winner method is first-past-the-post. It is a very bad method for a lot of reasons, most importantly that in the case where there are more than two candidates, the winner is likely to have plurality (more votes than anybody else) but not majority (more than half of the total votes). This is a major contributor to voter disenfranchisement. It is not a stretch to claim that in any case where there are more than two candidates, first-past-the-post is the worst possible system to use.

Plurality Voting (PV) is a category of methods, of which first-past-the-post is an example, but also contains multi-round systems. The most common of these is the two-round system, whereby after a first round, if no clear majority winner has been found, all but the top two candidates are eliminated, and then a new vote is conducted. The extreme form of this is exhaustive ballot, whereby candidates are eliminated one by one and the votes are repeated until there is a majority winner.

As these plurality systems are tiresome in practice, ranking methods have become more popular. The most clear analogue of plurality voting is instant runoff, or IRV, in which a ranked ballot is used to discover ordered preference. It is equivalent to exhaustive ballot, except that each voter only has to enter one ballot, with the recount being conducted automatically after each elimination.

Condorcet methods are a set of ranked ballot methods based on the Condorcet criterion, which stipulates that the winner should be the candidate that would win against every other candidate in a pairwise election. As the Condorcet criterion is not guaranteed to be met, various strategies have been suggested to force a winning condition by eliminating the weaker winners from the Smith set. These include Copeland's method, the Schulze method and Nanson's method. Of these, the Schulze method is becoming a fast favourite due to its various properties, and its ability to be used for both single-winner and multi-winner competitions.

Most multi-winner methods can be used in single-winner polls, simply by only taking the first winner into account. In practice, many multi-winner methods reduce to known single-winner methods in the case where there can be only one winner. Single Transferable Vote, for instance, reduces to IRV when there is only one winner.

Multi-winner methods

There are four major categories of multi-winner methods: proportional party-list methods, proportional individual methods, semi-proportional methods, and majoritarian methods. These vary substantially, but it is safe to say that majoritarian methods are generally frowned upon as being undemocratic, and semi-proportional methods have emerged mostly to suit the needs of those in power at each time, rather than as part of a philosophical pursuit for fairness.

The most common majoritarian method is block voting, which is an extension to first-past-the-post to allow multiple members to be selected by way of multiple non-transferable votes. In this case, the single choice balloting method is extended to allow more than one unranked selection, typically a fixed number smaller than or equal to the number of contested seats.

Of the semi-proportional methods, the most typical is cumulative voting, in which a voter is given a fixed number of tokens or points that can be apportioned to any candidates, in whichever measure. Thus, a

voter with five tokens could choose to give four to candidate A and one to candidate B. This method is criticized mostly because it supports “plumping” candidates, whereby a minority candidate with a strong following in that minority may win out over a more generally accepted candidate with more moderate following. This behaviour causes it to violate the majority criterion, the later-no-harm criterion, and be Pareto-inefficient.

Due to the numerous drawbacks of these classes of methods, we are happier with the party-list and individual methods. In party-list methods, voters choose between sets of options, which typically have been pre-ordered. The winning seats are allocated to the lists proportionately in order in precedence, depending on some way of calculating how many seats each list obtained. Two popular seat-allocation methods are Sainte-Laguë and Droop.

The Sainte-Laguë method is a highest-averages method, whereby the first seat is allocated to the party with the largest vote count, and then that party’s votes are divided by 3, which is the next odd number for that party (1, 3, 5, 7...). When that is done, the process repeats itself. Generally, a party is allocated an n -th seat if that party’s quota $v/(2n-1)$ is higher than any other party’s quota. Usually this quota is represented as $v/(2s+1)$, with v being that party’s vote count and s the number of seats they have been allocated so far. The Sainte-Laguë method is similar to the d’Hondt method, but the latter has been criticized for its tendency to favour larger parties, whereas Sainte-Laguë is relatively neutral.

The Droop method is a largest-remainders method, where the seat value is calculated as a quota of the valid votes v and the number of seats being contested s , normally written as $(v/(s+1))+1$. This quota is called the seat value. The first seat gets allocated to the party with the highest number of votes, and then the seat value gets deducted from their total vote count. The Droop method is an improvement on the Hare quota, which is simply v/s . While the Hare quota is more neutral in terms of seat allocation than Droop, the Droop quota is considered more efficient when used as an allocation method with systems such as Single Transferable Vote. In individual methods, each candidate is independent of all others, and seats are allocated by some determination of their order.

Borda count is a ranked ballot method used for both multi-winner and single-winner elections. Ballots are counted by assigning a point value to each place in each voter's ranking of the candidates, and the choice with the largest number of points overall is elected. If there are n candidates, each receives $n-1$ points for every first place vote, $n-2$ points for every second place vote, and so on down to zero points for a last place vote. Variations exist where the points awarded are not simple increments, but have jumps, such as in the Eurovision song contest, where the top scores are worth 12 and 10 points, after which points 8 down to 1 are complete. This is done to increase the spread between winners and reduce the likelihood of ties. Borda count has been criticized for favoring candidates that are mediocre by consensus over better candidates that are disputed.

Single Transferable Vote (STV) is a popular method for calculating results based on ranked ballots. It is currently one of the preferential voting systems most used by countries and states. A quota is calculated, for instance by the Hare or Droop methods, and any candidates that achieve the number of votes required for election are elected. Their surplus votes are redistributed to the next choice candidate on each voter’s ballot. Once this is done, if not all places have been filled then the candidate

with the smallest amount of votes is eliminated, and their votes are also redistributed to the voter's next choice. This whole process is repeated until all seats are filled.

A situation can arise in STV where if there are two winners and only one seat left, or the two weakest candidates are equal in strength, then someone needs to be eliminated. This is generally done randomly, although non-random strategies exist. In the random case, an external factor is introduced: e.g. dice, coin toss, or pseudorandom number generator. For reasons of repeatability of calculation, it is preferable to use a pseudorandom number generator (PRNG), and publish both the generator function specification and its seed value. Otherwise, if a vote is contested, it cannot be guaranteed to yield the same result from the same ballot set on recount. There are numerous variants on STV, such as the Schulze-STV method, as described above.

Ordering methods

Ordering methods are concerned with the ordering of different decision alternatives by preferences. They are essentially a special case of multi-winner voting systems where there is no limit to the number of seats. As such, they can be used for issues such as prioritization of projects, or deciding a sequence of events. All multi-winner methods can be used for ordering, simply by setting the number of seats to the number of candidates. We highlight this as a special case though, because the desired properties of an ordering election may be different from those of an election with a limited number of seats. For instance, the fact that there is no loser in an ordering election may change voter behaviour.

For ordering, methods such as Borda count show renewed value, although Schulze is a preferred ordering method.

Scoring methods

Scoring methods can be implemented in many different ways, yet they all share a common general procedure: different decisions alternatives are scored according to a set of criteria. Decision criteria are important because they constitute the key factor through which the alternatives will be assessed. When several criteria are present, they that might be weighted according to their relative importance to the overall decision.

Scoring methods are commonly used in the context of qualitative assessments (e.g. to assess the relative quality or merit of a thing) as opposed to standard decision-making procedures (e.g. when the polity is asked to choose a particular set of alternatives). They are not generally guaranteed to yield a single winner, although there is in practice a high likelihood that they will.

The advantage of scoring methods is that they can be done quickly and they do not require many resources in terms of time, money, or expertise. They are the most useful in situations that require a formal value judgements, which can easily be made explicit through quantification. The drawback is that results are easy to manipulate through the selection of criteria.

The most typical scoring method is range voting, which can also be used in single- or multi-winner scenarios, with the aforementioned limitations.

Resource allocation methods

The aforementioned approaches are useful for situations where all the candidates are considered a priori to be equal in value. This is appropriate for most elections, but when candidates have an inherent

difference in value, a different approach may be required, whereby the relative values of the candidates are taken into account. These can be referred to as resource allocation methods, wherein a certain number of resources may be available, and a certain number of uses for those resources exist, but the different uses vary in how much of the resource is required.

The weighted cumulative method is a multi-winner system that is similar to the cumulative method, but adds weight (or cost) to every one of the alternatives that the polity can choose from. This makes citizens aware of the costs of each alternative, and invites them to think more thoroughly when deciding to which alternatives to allocate their votes (e.g. some alternatives might be incompatible with each other, because the cumulative weights of the two is higher than the actual availability of resources).

As a way of illustration, let's assume that the polity is asked to vote on a particular set of projects - each of which cost money - and that there is only a limited pool of money to draw from. Every citizen can allocate the budget to one or more of these projects, till the budget runs out. Once everyone has voted, the votes are aggregated and the projects that have the highest occurrence (or the highest frequency of occurrence) are selected as the winner, and so on until the budget has been completely allocated. However, in each step, any projects that exceed the remaining available budget are eliminated.

This method is particularly useful when it is necessary to decide between multiple alternatives, whose successful implementation is dependent upon a particular set of resource allocation. The drawback is that the system does not have a notion of priority that may have been given to each project by each voter. Which is to say, when a voter is choosing which projects to "fund", they may be more concerned about certain projects being funded than others.

The weighted ranking method solves the aforementioned drawback by taking into account the order of preferences (i.e. the priority) expressed by every citizen when asked to decide upon the best allocation of resources, and allocating the next valid option using STV. As such it is equivalent to STV except in that the number of seats is variable and unpredictable until the allocation is complete.

4. Vote Delegation

Any system which allows a voter to implicitly or explicitly allow another person to vote on their behalf can be called a delegated voting system. In practice, most legislation globally happens on the basis of implicit delegated voting, whereby general elections are used to proportionately choose delegates who each have equal voting power in a legislative assembly.

A more nuanced approach to this model is to assign voting power to delegates based on the number of delegations they received directly, rather than having an arbitrarily limited number of delegates who share equal voting power despite perhaps having radically different proportions of voter support behind them. This arrangement only makes sense in single-constituency systems, or systems where constituencies are guaranteed to be equal in size. If used appropriately, this approach strengthens political accountability and allows for merit-based appropriation of voting power to specialists in each field.

When implementing a delegated voting system of this sort, which is generally called a “liquid democracy system”, there are several assumptions which must hold in order to guarantee a well functioning system, and a number of optional features which may be implemented or not, depending on the wishes of the politics in question and the desired properties of the overall voting system.

Transitivity

Once a vote has been delegated, the question arises whether the delegate can further delegate the vote. If that is allowed, then the delegation system has transitivity. Without transitivity, the delegation system is quite limited. It is recommended that transitivity be implemented.

Acyclicity

The most important required property is acyclicity, meaning that no cycles of delegation may occur. If this assumption holds, the overall delegation graph is a directed acyclic graph (DAG). If Alice delegates to Bob, then Bob must not be able to delegate to Alice. This simple case is easy to detect, but if transitivity is allowed, then cycles of arbitrary length can occur. In the case of cycles emerging, it is necessary to first detect and then resolve the cycle. How this is done varies depending on implementation.

Detection is easiest if the entire delegation graph is public, in which case any person can traverse the graph and detect cycles. If the delegation graph is not public, there are two cases: the first is where the entire delegation graph is managed by a centralized system or trusted party who has full knowledge of the graph. In this case this trusted party can traverse the graph and detect cycles. The second is the case where the delegation graph is managed by a decentralized system where no party has full knowledge of delegations (but is otherwise assumed to be verifiable). In this case cycles can only be detected in the case where the number of votes counted differs from the number of votes cast plus the number of votes delegated. If that occurs, then each participant can choose to temporarily cancel their own delegation to see if it resolves the mismatch.

Once the cycle has been detected, it can be eliminated by anybody who is part of the cycle, simply by canceling their delegation. If the delegation graph is public, then any user can verify before creating a delegation whether the delegation will lead to a cycle being created. Their voting software can then disallow the creation of such a cycle. If the delegations are managed by a centralized system, it can resolve the cycle by either: a) cancelling one delegation at random from the cycle (and alerting the voter to the change, b) disallowing the delegation that created the cycle in the first place, or c) cancelling the most recently created delegation that contributes to the cycle. Which of these strategies is optimal depends on the computational constraints of the system, but generally strategy ‘b’ is the preferred strategy in this situation.

Vote to delegate vs information to voter

One choice that the implementer needs to make is between the “vote to delegate method” and the “information to voter method”. In “vote to delegate,” the system directly gives the delegate control over the vote, which the delegate uses whichever way they choose. This approach can maintain voter secrecy if voter does not get information about how the delegate voted, but this leads to the problem that a voter may not know which way their vote was ultimately used. In the “information to voter” approach, the delegate allows the voter to have information about her choice, which the voter can

choose to mimic. Does not maintain the delegate's voter secrecy. It is recommended that the "vote to delegate" method be implemented.

Total, categorical, and per-issue delegation

Vote delegation can be achieved at different degrees of granularity, depending on whether the vote is delegated to another individual for the whole realm of decisions (total delegation) or only with regard to a particular category of issues (categorical delegation) or to a predefined specific and individual issue (per-issue delegation).

Total delegation means that citizens can only delegate their vote to another citizen, which will act as a proxy for their decision-making process. A voter can generally only delegate their vote to one person, unless a fractional or weighted delegation system is used - something which has yet never been implemented, nor suggested.

Categorical delegation means that citizens could choose to delegate one particular set of issues related to, for instance, health care to one person (e.g. a doctor) and education to another person (e.g. a scholar), while retaining the right to vote on all the other types of issues. Because issues might sometimes refers to one or more categories, categorical delegation of votes requires a formal mechanisms for deciding which category an issue belongs to. Besides, specific mechanisms need to be put into place whenever one issue is declared to belong to multiple categories (these can be resolved by mathematical algorithms, similar to those employed for vote-splitting).

Per-issue delegation provides the highest degree of granularity. It allows citizens to retain control over who is going to decide on what, by delegating predefined issues to different people. The drawback is that it requires more effort on the part of the citizen delegating his or her vote.

It is possible to offer one or any combination of those types of delegation. For instance, a citizen might delegate all of his or her votes to Alice, except for the issues related to Agriculture, which have been categorically delegated to Bob, whereas the specific Farm Subsidy Reform Bill has been delegated to Charlie. In this case, there needs to be an order of preference (or a splitting rule) to address the situation where different delegation rules are conflicting with each others. It is recommended that the D-CENT platform implement all three forms of delegation, with the rule that more nuanced delegation takes priority.

Chain length limitations

Some have argued for adding an explicit limit to how much transitivity can be practiced, or causing transitive delegation to decay. The argument for this is that when a voter delegates to a delegate, they are entrusting their vote to that delegate, and not to anybody else. However, their delegate may choose to delegate her vote (and any delegated votes) onward to a third party. If an explicit limit is enforced, then votes can end up being voided if the chains get too long. For instance, if an explicit chain length limit of 3 is enforced, and Alice delegates to Bob, Bob to Charlie, and Charlie to Doug, then Alice's vote exceeds the limit and is ignored. The decay approach does not explicitly void the vote, but rather applies a decay function, such as $f(d) = 1/(d/3 + 2/3)$, which would make the vote count less and less as it proceeds further along a chain -- in this case, making a second order delegation be worth only 75% of the original vote.

The counterargument against such an approach is that the choice to delegate implies trust. When Alice delegates her vote to Bob, she is trusting Bob's judgement, and if Bob's judgement is that Charlie is better equipped to make an informed decision, then surely that doesn't alter Alice's opinion that Bob is trustworthy. If it does, however, then Alice is free to stop delegating to Bob. A middle ground here would be to be able to "tag" vote delegations as being either delegable or not, but this too fails the trust criterion.

Generally speaking, any chain length limitations or delegation rules add substantial amounts of complexity to the system, with insufficient justification. In centralized voting platforms this complexity is manageable, but in decentralized multi-authority voting systems, the complexity may spiral out of control quite fast. It is recommended that no limitations of this sort be implemented.

Split delegations

Once delegation has been accepted in principle, it is not hard to imagine the delegation being split, such that Alice can, rather than just delegating to Bob, also choose to delegate a portion of her vote to Eve. There are three main options in such a method: whether all delegates must receive equal "shares" in the delegation or they can be split arbitrarily, whether one delegate abstaining from the election causes that delegation to be void, and whether there are any limitations to how many delegates one voter can have.

The main argument in favor of split delegations is that it allows the voter to more comprehensively express trust in perhaps conflicting viewpoints. If Bob and Eve disagree strongly on a topic, and Alice believes both of them have a valid point, she may lend her voting strength to both parties. If arbitrary splits are allowed, she could decide that Bob can have 20% of her vote, and that Eve gets the other 80%. Similarly, if there are no limits to how many delegates she can have, she can also split her vote with several other delegates who she also holds faith in. If the abstention rule is implemented, and Eve decides not to vote, then Bob's share in the vote gets rebalanced such that the vote doesn't get wasted. All of these approaches are interesting, but in practice the likelihood of it being useful and meaningful to a voter is low. Most voters, most of the time, will not want to engage in vote splitting, and even providing the option means a greater cognitive burden for the voter, which reduces the voter's capacity to make informed or strategic decisions. The nuance which is added by allowing vote splitting is negligible compared to the amount of complexity it adds to the system, in particular whereas further delegations of split votes can lead to numerical instability and implementation-specific issues around floating point numbers. As a result, it is not recommended that split votes be implemented.

5. Voting process and the block chain

Another part of a voting system is the environment the process occurs in. Traditional voting occurs in a pipeline whereby people put anonymous votes into locked ballot boxes which are transported under supervision to highly guarded and monitored counting stations where multiple people are involved in each step of the count. This process is not foolproof, but it is an example of what has been called byzantine verifiability, whereby more people with differing opinions, allegiances and motives are added to the system until those in charge of the process agree that the likelihood of misdeed or error is negligible. This is analogous to byzantine fault tolerance in systems design. Electronic voting methods tend not to guarantee verifiability or anonymity (or, rather, the special case of anonymity where a ballot

cannot be linked to its voter; known as unlinkability). Centralized databases are easily manipulated and monitored in most cases, and there is no way to continuously audit the running code.

An alternative to centrally governed and centrally operated systems or inherently malleable and surveilable electronic systems, is to use a block chain mechanism as a distributed log of events. The block chain was originally developed for use with the Bitcoin digital currency, but is in effect a distributed, non-falsifiable, append-only data store. We can imagine a block chain being created for each election, where the genesis block contains details about the voting system that will be used, any external factors, and a formal statement about the election itself (such as the ballot question). Subsequently, tokens are pre-mined and given to voters or mined by legitimate voters by some mechanism which guarantees one vote per person. These tokens should only be divisible if split-delegations are allowed.

With this mechanism, it is possible to verify that every vote belongs to a legitimate person, but impossible to say who, unless the private part of that person's key is published. If a voter casts multiple votes, there will be multiple votes signed with the same key. In that case, the most recent vote can be considered valid, and older votes discarded.

This fulfills, in theory, all of the requirements for unlinkability and verifiability. It's possible to verify pretty easily that any balloting method and tallying method will work with this generic approach. However, in order to support vote delegation, or proxying, a further complication can be added in the form of having two separate keys: a delegation key, and a voting key.

The delegation key is the one that is arrived at through the token system described previously. A voter can generate a new key pair, which will be the voting key. The public part of the voting key is signed with the delegation key, and that signature published as a statement that this is the current valid voting key for this delegation key.

To delegate, the private part of the voting key is encrypted to the public delegation key of the delegate - this public key must be obtained somehow. The delegation can be "advertised" by publishing this encrypted blob, signed with the voter's delegation key, in the blockchain. This allows for cycle detection, by exposing the graph. If advertisement is not implemented, the delegation graph (which is anonymous, but could possibly be subjected to a correlation attack), is hidden. In this case there needs to be a non-blockchain (out of band) mechanism to deliver the voting key, and there is no simple way (beyond vote counting) to validate the existence of cycles.

The delegate then decrypts the voting key and adds it to their collection. If they want to delegate onwards they share it forward. This means that using this approach, transitivity cannot be practically avoided, and chain length limitations are unenforceable. However, split delegations can be allowed by allowing a voting key notification to specify multiple voting keys at once, and even with markup to define their relative vote share.

A vote is cast by signing the ballot with the voting key. If there are multiple ballots signed with a voting key, all but the most recent are ignored during tallying. If the voter wishes to revoke the proxying, they generate a new voting key and publish it the same way. This notification works similar to the votes, in that if there are multiple voting key notifications from the same delegation key, the most recent is valid,

all others not. Votes from invalid (revoked) voting keys are ignored. It is recommended that this method be implemented, as a secure, decentralized voting mechanism.

6. Deliberation systems

An essential part of the democratic process is based on dialogue and deliberation amongst the members of the polity (Barber, 1984; Elster, 1998; Amsler, 2004). This can be implemented through a variety of different deliberation systems, each with their own advantages and drawbacks.

Ideally, deliberation systems should be inclusive, they should encourage debate and discussion, facilitate compromise and consensus formation, so as to ultimately provide broader public support and greater legitimacy to the decision-making process. Yet, there are several practical difficulties in implementing a truly perfect deliberative system (Warren & March, 2006). These difficulties mostly relate to: (1) scale, i.e. given a large-scale polity, how to ensure that each citizen has an opportunity to participate equally in the deliberative process? (2) competency, i.e. given that citizens have diverse interests and levels of educations, how to ensure that the most competent ones will be recognized? and (3) time commitment, i.e. given that deliberation is a time-consuming activity, how to ensure that citizens participate to the deliberative process consciously and equally?

Deliberation systems can be subdivided into two categories depending on whether there exist a formal procedure that needs to be followed during deliberation (structured deliberation) or whether deliberation is done in the context of a free-form forum until a decision is taken (unstructured deliberation).

Unstructured deliberation

Unstructured deliberation consists of continuous informal exchanges of information in random or self-assembled groups. Unstructured deliberation systems suffer from a lot of inefficiencies. In terms of time management, beyond the need of ensuring that every citizen has equal opportunities to express him or herself, one must also ensure that all the issues at stake be properly discussed. In a unstructured context, people often get caught up on one particular issue, which might monopolize the debate for a long period of time, thus preventing the polity to address other (often more relevant) issues. Besides, in order to maintain public order, avoid useless debates or violent fights, deliberation systems should implement a mechanism guaranteeing that everyone act in a 'civil' way. This is discussed in the "Comments" section of the front-end analysis of D-CENT.

Unstructured deliberation generally gives disproportionate influence to the most vociferous, confident, and charismatic speakers, who are not necessarily the best qualified to formulate public policy (Pivato, 2009). Besides, the more unstructured deliberation is, the greater the possibility that more powerful parties can take advantage of it (Kahane & al., 2010). Another concern has been referred to as "the dictatorship of free cycles" whereby people who have greater (political) problems have less capacity to tend to structural solutions to their problems, while those who have smaller (political) problems are less aware of the realities of those problems (McCarthy, 2013). It is, therefore, frequently preferable to implement a more structured system of deliberation, even if it is procedurally imperfect.

Structured deliberation

The idea underlying structure deliberation systems is that, in order to fully realize the potential of deliberative democracy, there needs to be a structure allowing for polity to continually and substantively participate in the legislative process, and a particular deliberative format which ensures that everyone gets a voice during the deliberation process. Most of the major concerns found in unstructured deliberation systems were worked out about a hundred and forty years ago by Henry Martyn Robert and published in a book known as Robert's Rules of Order. There were many similar books both before and after Robert's Rules, with different focuses and different approaches, but Robert's Rules is the most universally known method. What Robert did was define what we would today call an algorithm for how to deal with any conceivable situation: how to determine the order in which people speak, how to limit people from going off topic, how to make sure nobody was excluded. It includes information about how assemblies should work, how subsidiarity should function, how debates and votes should be conducted, and much more.

Almost every parliament in the world has a set of rules, commonly referred to as rules of procedure, which are either structurally or functionally equivalent to Robert's Rules of Order. They may contain variations on the various themes, but they all hold in common the basic notions: one person at a time, deference to the speaker, etc.

Scale

If there are hundreds of people in an assembly, and only one can speak at once, there is a limit to how much can be said. Thus, scale is inherently connected to the notion of bandwidth. Bandwidth is a way of measuring how much information can be transmitted through a channel over a specified time interval: the greater the bandwidth is, the more information can be transmitted through a channel can per time interval.

Another important factor to account for, in addition to bandwidth, are the properties of the channel through which deliberation is made. For instance, one thing that differs significantly between speech and writing is a property called blocking. (If you've ever sat with lots of people at a dinner table and everybody is talking at the same time, you know what blocking is). While different people can read different things at the same time, when it comes to speech, people can only properly listen to one person at a time, and if many try to talk at the same time the whole process of communication is blocked (i.e. its increases the noise level and cause listeners to miss parts of what is being said - data loss).

Competency

If deliberation is to enhance the quality of legislation, then there must be meritocratic mechanisms which promote the most competent (i.e. intelligent, educated, informed, engaged, ethical, objective, pragmatic, and open-minded) participants, instead of favoring ideologues, extremists, and demagogues (Pivato, 2009).

Committees are an attempt to recognize the fact that not everybody has the time to deliberate on everything. By breaking the parliament into subgroups with specific mandates, these groups can operate independently under a special set of rules and then come back to the main assembly with its findings at regular intervals. This has worked for a while, but there have over the years been increasingly many

committees in virtually every parliamentary regime, often operating externally. There are just too many things to cover, and there often is not enough capacity to parallelize the parliament to the degree necessary to cover everything.

Time commitment

Time commitment is an important factor to be taken into account when assessing the viability and effectiveness of participatory democracy. First, most people have little time to get involved in democratic deliberations, and would rather not spend what little free time they have on getting involved with the deliberation process, with all the complexities it entails. As a cost assessment, the supply of time to self-govern is substantially less than the demand to time that self-governance requires. Therefore any effort and promoting participatory democracy must be focused on reducing the cost of participating. This can be done by reducing the amount of explicit deliberation required on issues, or increasing the impact of each individual's decision at each time.

Decentralized structured deliberation

For the purposes of decentralized or distributed systems, a number of assumptions that can be made in traditional parliamentary settings need not apply.

- Text comments, video comments, audio comments; compare and contrast
- Top-post vs bottom post
- Formal methods of deliberation vs informal methods
 - Robert's Rules of Order
 - Parliamentary Rules of Procedure
 - Other formal methods

7. Technical Specifications

Balloting methods

The balloting methods required are those which are needed by the voting methods suggested in the next section. They are therefore the following:

- Multi-choice balloting (and Single-choice balloting as a special case)
- Ranked balloting
- Approval balloting
- Weighted balloting

Voting methods

The following methods should be implemented:

- Consensus method
- Condorcet method for single-winner polls
- Schulze method for multi-winner polls and ordering

- Approval method for single- or multi winner polls and scoring
- Weighted ranking method for resource allocation

When to use each method

With all of this variety it can seem daunting to decide on a method to use, if many are available. Here are some rules of thumb that can help a person or polity decide which form of election to conduct.

- Use consensus only if necessary, when the gravity of the vote in question is such that no disagreement can be allowed.
- If it is reasonable for more than one to win, use a multi-winner method. However, be careful about choosing the number of seats, as any number you choose that is not 1 or infinity is arbitrary, and needs to be justified.
- Try to decide on issues rather than deciding on people. Pushing authority up a chain is a pathway to corruption and abuse of power.
- Uses resource allocation methods when there is a reasonable or natural weighting on the candidates, but be careful that the introduction of weighting is easily biased, and that process should be well laid out, documented, and scrutinized.
- If you don't need a clear winner, only a comparison of relative merits, use scoring methods. Don't try to make a top ten list if it isn't necessary. The long tail of options is long for a reason.

Vote delegation

It is recommended that the D-CENT platform implement all three forms of delegation - total delegation, categorical delegation, and per-issue delegation - with the rule that more nuanced delegation takes priority.

- It is recommended that the "vote to delegate" method be implemented.
- It is recommended that transitivity be implemented.
- In the case of vote delegation leading to a cycle (or loop), the system should resolve the cycle by either:
 - a) cancelling one delegation at random from the cycle (and alerting the voter to the change,
 - b) disallowing the delegation that created the cycle in the first place, or c) cancelling the most recently created delegation that contributes to the cycle. Which of these strategies is optimal depends on the computational constraints of the system, but generally strategy 'b' is the preferred strategy.
- It is recommended that no chain-length limitations be implemented.
- It is not recommended that split vote delegations be implemented.

Technical implementation

- It is recommended that the D-CENT platform implement, as a secure, decentralized voting mechanism relying on the blockchain technology
- It is recommended that zero-knowledge-proof mechanisms be incorporated by default into the system in order to ensure the verifiability and unlikability of the votes.
- It is recommended that vote delegation be supported in the scheme.

Block chain mechanism

Note that the actual cryptographic components will be decided in the course of implementation in WP5. The cryptographic primitives are covered in the “Introduction to Cryptography” section.

Delegation Algorithms

Alice wants to delegate her vote to Bob:

- Alice initiates an ECDH with Bob through the delegation message.
- Bob completes the ECDH with Alice through the delegation2 message.
- Alice calculates the shared key and publishes it as her valid voting key with a votekey message.
- Bob calculates the shared key and can use it to vote on Alice’s behalf once votekey has been done with that key.

Alice wants to revoke her delegation to Bob:

- Alice computes a new voting key and publishes it with votekey. This invalidates the previous shared voting key.

Bob wants to delegate his vote, to Charlie:

- Bob does the standard procedure for delegation, as described above.
- When issuing votekey, he adds signatures from all of the keys he has currently got delegated to him.

Somebody wants to count votes:

- Compile a list of currently valid voting keys (verify each one against the current election’s valid delegation keys)
- For each voting key that has more than one signature, check if each signature is from a currently valid voting key. If it is, increment the weight of that voting key.
- Compile a list of votes. Discard all votes that are not from currently valid voting keys, or are from voting keys which have signed another voting key as part of a delegation operation.
- Count remaining votes, weighting them by each voting key’s current weight.

Messages

Messages in the block chain mechanism can be derived from the Bitcoin block chain format although may be encoded with JSON for use as messages in Web APIs. The following message types should be supported:

- Main messages
 - election
 - token
 - votekey
 - vote
 - delegation

- candidate
- candidate_e
- candidate_s
- vote_start
- vote_end

election

This message should be the first message in a new election. It may not be repeated during the election. Election ID (eid) can be calculated as $\text{dhash}(\text{master_key})$.

Name	Description
master_key	Public part of the election master key
etype	The election type. See the voting systems list
dtype	The delegation type. See the delegation modes list
winners	The number of winners the election can have. 0 for infinite
ext_len	The number of bytes the external factors set will have
external	Any external factors required by the election, such as random seeds

candidate

This message is used to announce candidacy in an election. It can only be issued after candidate_s and before candidate_e in each election. Outside of that, it is ignored. It is also ignored if the delegation key is not a valid delegation key for the election.

An election master can preselect candidate options (such as in “yes/no”) elections by issuing candidate_s and candidate_e before announcing any legitimate delegation keys.

Name	Description
eid	The election ID.
signature	Signature of this candidacy, made with valid delegation key or with the election master key
name_len	The length of the candidate’s name (or option value)
name	The candidate’s name, UTF-8 encoded

candidate_s

Start accepting candidates

Name	Description
eid	The election ID.
signature	Signature made with the election master key

candidate_e

Stop accepting candidates

Name	Description
eid	The election ID.
signature	Signature made with the election master key

token

Advertise a new delegation key (token) as being valid in this election

Name	Description
eid	The election ID.
pubkey	Public part of the new delegation key
sign	Signature of the validity of the token by the election's master key.

votekey

Advertise a new voting key as being valid in this election. Only the newest voting key for any given delegation key is valid.

Name	Description
eid	The election ID.
pubkey	Public part of the new voting key
sign_num	The number of assigned signatures. Minimum 1 - a signature from the voter's delegation key. If greater than one, this is a transitive delegation. If the election's delegation rules forbid transitivity, then votekey messages with sign_num > 1 are rejected.
sign	A list of signatures.

vote

A balloting action. Contains the vote data.

Name	Description
eid	The election ID.
sign	Signature from the voting key that authorizes this vote.
ballot_len	The length of the ballot data
ballot	The ballot data, encoded correctly for the election type. If this is encoded incorrectly, the vote is discarded.

vote_start

Start accepting votes

Name	Description
eid	The election ID.
signature	Signature made with the election master key

vote_end

Stop accepting votes

Note: May need to do something to thwart replay attacks.

Name	Description
eid	The election ID.
signature	Signature made with the election master key

8. Data Portability

Feature: Porting user and community data between D-CENT nodes

Feature Definition: The various personal data accumulated about a user and a community should be seamlessly moved between D-CENT nodes when a user or community authorizes it.

User Need

Often, individuals or even entire groups may move physical location, interest, or even their trust relationship with the administrators of a D-CENT node. In any of these instances, the users should be

able to change their D-CENT node in a privacy-preserving manner without losing their data, which means having their “right to be forgotten” and right to “data portability” as part of their general rights around Data Protection enforced if desired.

Description: In this feature, there is generally an exporting system and an importing system, where the personal data of a user or social data of a community is being transferred from an exporting system to an importing system. In our case, we will consider both the importing system and exporting system to be D-CENT nodes. In general, we will consider the ability for a user to also “back-up” their data to a version of data portability where the user simply does not delete their personal data from the previous system. Right now, we consider personal data to be profile data and any copies of messages. For groups, it will also include the social data store that contains all group-based messages. However, if a user wishes to delete all messages sent to other users or in another group, because important decisions could depend on those messages, it would require the receiving user or other group to delete their messages as well for the messages to be truly deleted.

Personal Data: All data that is about an identified user. In terms of a community, we would substitute the term “social data” for personal data throughout this feature.

Profile Data: Data that contains information about an individual user, such as their profile photo and contact data.

Messages: Data between users to other users (including groups)

Exporting System: The system that currently contains the personal data before the exporting of the data.

Importing System: The system that will contain the personal data after the exporting of the data.

The steps for a typical export and import are outlined below:

James wants his data out of the D-CENT node from Helsinki, as he is moving to Barcelona and wants to have his data under local control in Barcelona. He authenticates to his previous D-CENT node in Helsinki and clicks on the “Export My Data” button.

For each D-CENT node, there should be a button that allows the profile data and all related personal data to be downloaded to the user's client. Since the D-CENT node keeps the user's profile (in VCard) and wrapped keys (both user keys and group keys), as well as any other data, in their WebFinger repository for the user, this data is automatically zipped on the server and given to the user to download. For the user's personal data, all this data should then be removed using a Secure Delete Function (<http://srm.sourceforge.net/>). For messages, as each message is indexed to a user, the D-CENT node searches for each message. If the message is located and the privacy controls of the group/user allows them to, the user should be able to download a copy of these messages. Packages are created on-demand and then immediately deleted.

James then re-enters his new Barcelona D-CENT node and finds his “Import My Data” button, and uploads his profile and messages. He is pleased that he doesn't have to re-enter his personal data and has a copy of his old important messages, in ActivityStreams format, at least where having a copy of the message doesn't violate the privacy of a third party.

A user can then enter a new D-CENT node and upload their data. The data needs to be re-stored in the data-store of the server and the profile of the user needs to be updated.

Dependencies

Note this requires a number of formats be sorted out for all sorts of personal data. For profile data, we can build off of vCard 4.0 as mentioned in D4.1 (<https://tools.ietf.org/html/rfc6350>). Custom extensions can be built for personal data that is specific to D-CENT use-cases by storing the VCard in JSON (<http://tools.ietf.org/html/draft-kewisch-vcard-in-json-01>) and building custom extensions in RDF serialized to JSON using JSON-LD (<http://www.w3.org/TR/json-ld/>), in a manner similar to ActivityStreams. Message formats can be kept in ActivityStreams. We can build custom vocabularies for groups such as SIOC (Semantically Interlinked Online Communities - see <http://rdfs.org/sioc/spec/>) and then inscribe these using JSON-LD (

Open Decisions

There are still some hard problems about how to export a person's messages when their messages and other data are part of a group. In some cases, the group may wish to disallow that for security and privacy reasons. For example, it would not make sense to have an extremely sensitive conversation that has been verified to happen internally only within trusted groups uploaded a non-trusted server just because a member of the group wishes to move servers (possibly after having been removed from their original group and server). This scenario should be explored and detailed, but will likely come down to group-specific rules for interoperability.

9. Federation

Feature: Activity in D-CENT nodes should be federated.

Feature Definition: The status updates from one user or group between D-CENT nodes should be capable of being shared between users on different D-CENT nodes.

User Need: A user wishes to receive updates from another D-CENT node other than the one they are on.

Description: Federation is a hard problem, as for decentralized systems to make sense, activity between the various nodes should be shared when relevant between different nodes. Although the problem is massively complex, we can reduce the complexity by reducing the problem to that of receiving status updates. As work continues in WP5, in particular D5.3, we should gain enough experience to determine how to do more complex forms of status updates between different D-CENT nodes. The components of a D-CENT enabled federation are listed below. Again, a user can be substituted for a group:

Publisher: A user on a D-CENT node that exposes their ActivityStreams to subscribers.

Subscriber: A user D-CENT that wishes to receive information from a publisher in the form of ActivityStreams.

Provider Key: A asymmetric key that lets a statement be authorized as coming from one particular D-CENT Node.

Blaine, who is on a D-CENT node in the United Kingdom with mostly British users, wants to follow Harry, who is on a D-CENT node in France, since they share interests in coding for the social good. When they meet, Harry gives Blaine his email address harry@france.example.org. Blaine then logs in and wants to follow status-updates from Harry, having them linked into his ActivityStream.

Subscriber sends a message to Publisher's service provider that they want to follow the Publisher via a WebFinger with a "follow" relationship with a HTTP PUT Command.

```
PUT /.well-known/webfinger?
resource=acct%3Aharry%40france.example.com&
rel=http%3A%2F%2Fwebfinger.d-cent.example%2Frel%2Ffollow
HTTP/1.1
Host: d-cent.example.com
```

Subscriber's identity provider gives a signed token to Publisher's identity provider signed token, signed by the D-CENT node's provider key, that says they have the capability to act on User's behalf. The Publisher's identity provider checks the Subscribers provider key against its local cache or by retrieving it again using WebFinger. The Publisher's identity provider sends a URI to to the Subscriber that allows the Publisher to verify they want to be followed.

Next time he authenticates to his D-CENT node, Harry notices a message that asks "Can Blaine follow you?" Harrys replies "Yes".

If the URI contains an authorization, every time a new ActivityStream is published with public access control, a copy of that ActivityStream can be accessible when the Since HTTP is a pull rather than push protocol, the D-CENT node of the subscriber must routinely poll the URI to find new ActivityStream content from the Publisher.

Blaine then finds that in his ActivityStream in D-CENT there are now new status updates from Harry.

Dependencies

In general, this work relies on all D-CENT nodes deploying ActivityStreams and WebFinger, with the general design outlined in Section on Identity Management, and then using these coherently to knit together a federated system.

As has been noted earlier, there are already Javascript libraries for ActivityStreams, although they need to be kept inline with the work of Social WG.

<https://github.com/el4n/pump.io>

There is also a library that lets Twitter, Facebook, and Google+ interoperate with ActivityStreams:

<https://github.com/snarfed/activitystreams-unofficial>

Open Decisions

This system of federation only allows broadcast out of messages to other D-CENT nodes, and does not deal with authenticated and time-stamped responses. This is a more challenging technical problem that was addressed originally by the Salmon Protocol (<http://www.salmon-protocol.org/>) which is also the sending of status updates, and their re-ordering due to timing differences. This is an important but harder problem to solve, and real require more work.

There is also the question of whether groups should be federated across different D-CENT nodes, or only internal. This would allow more complex scenarios of authentication rather than simple permissions. This will be tackled during the Deliverable D5.3 on federation.

Since HTTP is a pull rather than push protocol, it is unclear if federation will scale easily over HTTP since it requires the subscriber to check the publisher for new information on a regular basis.

10. Secure Messaging

Feature: Secure and Encrypted messaging.

Feature Definition: Secure Messaging is encrypted messages that can only be read by authenticated recipients.

User Need: A user does not want anyone else, including the D-CENT node itself, to be able to read the content of their messages. Thus, note this could not apply to public messages.

Description: Secure messaging is simply sending messages that are confidential, i.e. that can only be read by the sender and the receiver. This means that the content of the message must be encrypted. In general, this is a generalization of the properties discussed in Section on Identity Management, but with this additional property:

Confidentiality: The content of a message can only be read by the recipient.

In general, this is a well-known and solvable problem if we assume the messages are asynchronous, we do not include perfect forward secrecy, and handle groups as merely other users with long-term private and public-keys. In that case, we can reduce this problem to the familiar problem of encrypted email but with a JSON-based format (namely ActivityStreams) rather than SMTP-based email. There is the larger well-known problem that within the Web, the server is not necessarily trusted. However, we will assume the server trusted and so the server will not open the encrypted messages. We believe that, while imperfect, using SRP-based authentication to release wrapped user key material is the best way to do this until there are larger improvements in the Web Security Model. The components in this story are the same as in Identity Management. First, this technique was illustrated in the simple case of a single D-CENT node. However, encrypting the content of the message is most useful when messages are sent between nodes. This example should be able to generalize to those cases as well.

A user on one D-CENT node, Alicia, wants to send an encrypted message to Bob. She opens up her messaging window, types “Hello, are you there Bob?” and hits the “Send the encrypted message” button.

Using the same technique as given in Step 2 of Identity Management, a public key for the recipient is given by the D-CENT server via WebFinger.

The message from the sender is then encrypted using their private user key and embedded in an ActivityStream.

Alicia encrypts her message and sends it to Bob, with the usage of Alicia's private key to encrypted and sent over to Bob taken care of D-CENT. The message is then delivered to Bob, like this

```
{
  "To" : "acct:bob@d-cent.example.com",
  "From" : "acct:alice@d-cent.example.com",
  "alg" : "SHA1"
  "version": "W3CWebCrypto 1.0 (Mozilla Firefox)"
  "message": "yhfkkLj/kifGrEY/3rlmJSN7r7l...",
  "signature": "iswu7laBtmXQ/zQL0ex/AjXq..."
}
```

After receiving the message, Bob opens in. Seeing as the message is encrypted, his D-CENT node automatically tries to decrypt the message using his private key, resulting in him seeing the message “Hello, are you there Bob?”

After receiving the message, the message is stored until the user checks their messages. Then the user's private key is used to decrypt the message.

Dependencies

Currently, this depends on encrypting data using standard key operations using the W3C Web Crypto API.

Open Decisions

There is open work on how to best trust in a decentralized network the validity of keys retrieved using techniques like WebFinger. There is a broad array of possible techniques, ranging from a simple one based “trust on first use” to more complex mechanisms including “network perspectives” that are third-parties that validate keys from multiple points in the network.

If we wish to handle synchronous messages, we can move to a “chat” rather than “mail” protocol that can provide perfect forward secrecy, such as “Off-The-Record” messaging (Borislov, 2004). This is currently already implemented in a number of code-bases, such as Jabber (<http://wiki.xmpp.org/web/OTR>), although an amount of work would be required to upgrade such a pure JSON-based approach such as ActivityStreams to use OTR messaging.

In the area of cryptography, while OTF messaging is well-known and ideal in synchronous messaging, providing perfect forward secrecy with asynchronous messaging this is still a very much open area of research. While what we have discussed is very similar to encrypted email, with long-term private user keys being used to decrypt and encrypt messages, we lose perfect forward secrecy since the same key is used over and over again, so if the long-term private key is compromised adversary, all previous messages can be decrypted by an adversary. This can be ameliorated by some form of key rotation, but this still simply provides smaller time spans were an adversary can compromise the messages. Work on extending this to true forward secrecy generally involves pre-loading keys and there are a number of competing approaches, such as the key-ratcheting protocol by TextSecure (<https://whispersystems.org/blog/advanced-ratcheting/>) and the SCIMP protocol by Silent Circle (<https://silentcircle.com/scimp-protocol>).

11. Standardization plan

Since D4.1 has been written, with support of the D-CENT Project, the W3C has begun the Social Web Working Group and Social Interest Group to standardize the social web. This section outlines the decisions made in D4.1 and how they will influence the standardization effort, as well as next steps on the standardization of the various components of D-CENT that are still under standardization. To recapitulate D4.1:

Existing Standards:

- For retrieving personal data, we will begin with VCard 4.0. Extensions of this will happen in the Social Interest Group.
- For authorization, we will begin with ordinary OAuth 2.0 since at the time of writing User-Managed Access (UMA)'s OAuth variant is still immature and does not have good Javascript-facing library.
- For messaging, we will begin with HTTP calls delivering JSON-LD based ActivityStreams 2.0. Standardization will continue in the Social Web Working Group, as well as possible APIs for embedding ActivityStreams into content.

There are no known functional open standards around high-value authentication. Due to this reason, we have chosen to use Secure Remote Password (SRP) in conjunction with the W3C Web Cryptography API, as outlined in Section Strong Authentication. Although HTTPAuth does not yet feature such a scheme, the use of the Web Cryptography API Javascript calls on the client-side ameliorates these concerns if the server is trusted and the Javascript is delivered over a secure (TLS/HTTPS) origin. The W3C has begun work on assuring that the Javascript is unlikely to have suffered a Cross-Side Scripting attack by usage of Sub-resource Integrity, but has yet to begin to deal with work where the Javascript is independently verified by a third-party. In general, this work will happen across the W3C Web Cryptography Working Group and the Web Application Security Group, and we will actively track the developments in these Working Groups through the life of D-CENT to upgrade the security as needed.

There are also no known good standards for binding a user to a public key. This is important, as a public key is needed to authenticate a user. We have been discussed in Section Strong Authentication.

In D4.1, we outlined that our initial choices were to use a federated model where different D-CENT nodes would have different domains (such as example.org) email addresses with the name@domain form as generic user identifiers, with account identifiers taking the form of the acct URI scheme also of name@domain. These assumptions will be held through the rest of the document.

There are also no known good standards for federation. We will outline a simple message-passing of ActivityStreams in this document in Section Federation via HTTP, based on an updated model similar to the PubSubHubBub and Salmon Protocol. This work will continue in the Social Web Working Group.

Note that for this deliverable, we will only briefly deal with federation. While we will design our protocols with federation in mind, the user-stories here will deal with a single D-CENT node. We will scale these user-stories up out of a single-node, which may involve changes to the underlying protocols, in D5.3.

Standardizing the Social Web at W3C: The focus of the W3C Social Activity (<http://www.w3.org/Social/>) is on making “social” a first-class citizen of the Open Web Platform by enabling standardized protocols, APIs, and an architecture for standardized communication among Social Web applications. These technologies are crucial for both federated social networking and social business between and within the enterprise and can be built on top of Linked Data. This work will knit together via interoperable standards a number of industry platforms, including IBM Connections, SAP Jam, Jive, SugarCRM, and grassroots efforts such as IndieWeb and D-CENT.

The mission of the Social Web Working Group (<http://www.w3.org/Social/WG>), part of the Social Activity is to define the technical protocols, Semantic Web vocabularies, and APIs to facilitate access to social functionality as part of the Open Web Platform. These technologies should allow communication between independent systems, federation (also called “decentralization”) being part of the design. The Working Group is chaired by Tantek Celik (Mozilla), Evan Prodromou (E14N), and Arnaud Le Hors (IBM). Also part of the Social Activity is the Social Interest Group (<http://www.w3.org/Social/IG>) focuses on messaging and co-ordination in the larger space. This work will include a use-case document, including “social business” enterprise use-cases, as well as vocabularies. The Interest Group is chaired by Mark Crawford (SAP). More information is available in the charters of Social Interest Group and Social Web Working Group.

Context and Vision: The Social Activity has been a goal of many members of W3C for years. The Future of Social Networking Workshop was held in 2009 and attracted significant mobile and academic interest, and led to the creation of the Social Web Incubator Group that produced Towards a Standards-based, Open, and Privacy-Aware Social Web (<http://www.w3.org/2005/Incubator/socialweb/XGR-socialweb-20101206/>). Outcomes of this report included the more open W3C Community Group process, since much social web work was happening outside W3C as the W3C was at the time viewed as too exclusive of grass-roots efforts. This also led to further outreach, with the W3C sponsoring and helping organize the grass-roots Federated Social Web conference in 2011. However, at the time there was still not critical mass of W3C members interested in social.

More and more W3C members are embracing the concept of social standards, thank to the work of the Social Business Community Group, in particular the 2011 Social Business Jam (<http://www.w3.org/2011/socialbusiness-jam/>). The Social Standards: The Future of Business workshop (sponsored by IBM and the Open Mobile Alliance) developed the standards and ideas for decentralized social networking around industry use-cases. In particular, after the workshop the OpenSocial Foundation joined the W3C, and submitted (with other groups) the OpenSocial Activity Streams and Embedded Experience API as a Member Submission (<http://www.w3.org/2005/Incubator/socialweb/XGR-socialweb-20101206>).

Goals: The W3C Social Web Working Group will create Recommendation Track deliverables that standardize a common JSON-LD syntax for social data, a client-side API, and a Web protocol for federating social information such as status updates. This should allow Web application developers to embed and facilitate access to social communication on the Web. The client-side API produced by this Working Group should be capable of being deployed in a mobile environment and based on HTML5 and the Open Web Platform.

There are a number of use cases that the work of this Working Group will enable, including but not limited to:

User control of personal data: Some users would like to have autonomous control over their own social data, and share their data selectively across various systems. For an example (based on the IndieWeb initiative), a user could host their own blog and use federated status updates to both push and pull their social information across a number of different social networking sites.

Cross-Organization Ad-hoc Federation: If two organizations wish to co-operate jointly on a venture, they currently face the problem of securely interoperating two vastly different systems with different kinds of access control and messaging systems. An interoperable system that is based on the federation of decentralized status updates and private groups can help two organizations communicate in a decentralized manner.

Embedded Experiences: When a user is involved in a social process, often a particular action in a status update may need to cause the triggering of an application. For example, a decision that includes consensus from a community on a decision to post a Facebook status update may need to redirect a user to Facebook. Rather than re-direct the user to another page using HTTP, this interaction could be securely embedded within page itself.

Process-Oriented Human Computing: In any enterprise, different systems need to communicate with each other about the status of various well-defined social processes without having crucial information lost in e-mail. A system built on the federation of decentralized status updates with semantics can help replace email within any well-defined process, such as voting on a decision, for crucial well-defined institutional processes.

Scope and Deliverables: The Social Web Working Group, in conjunction with Social Interest Group, will determine the use cases that derive the requirements for the deliverables. Features that are not

implemented due to time constraints can be put in a non-normative “roadmap” document for future work. The scope will include:

Social Data Syntax: A JSON-LD based syntax to allow the transfer of social information, such as status updates, across differing social systems. One input to this deliverable is ActivityStreams 2.0. See the W3C ActivityStreams 2.0 Working Draft: <http://www.w3.org/TR/2014/WD-activitystreams-core-20141023/>.

Social API: A document that defines a specification for a client-side API that lets developers embed and format third party information such as social status updates inside Web applications. One input to this deliverable is the OpenSocial 2.5.1 Activity Streams and Embedded Experiences APIs Member Submission, but re-built on top of Linked Data with more secure Javascript sandboxing.

Federation Protocol: A Web protocol to allow the federation of activity- based status updates and other data (such as profile information) between heterogeneous Web-based social systems. Federation should include multiple servers sharing updates within a client-server architecture, and allow decentralized social systems to be built. One possible input to this task is WebMention and another possible input is the Linked Data Platform. Each of these technologies should not be tightly coupled but can allow general purpose use. Each specification must contain a section detailing any known security and privacy implications for implementers, Web authors, and end users. The Social Web WG will actively seek an open security and privacy review for every Recommendation-track deliverable.

Members of Social Web Working Group and Social Interest Group

At the time of writing (Sept 2014), the Social Web Working Group and Interest Group include:

- Mozilla
- IBM
- SAP
- Siemens
- Boeing
- University of Southampton
- University of Edinburgh
- OpenSocial Foundation
- INRIA
- Avaya Communications
- Apache Software Foundation
- Telecom Italia
- Nokia
- Ford Motors

Schedule: The production of the deliverables depends upon the resources available, and will change as new information and implementation experience is reported to the group. The most up-to-date timeline is available from the Social Web WG page.

<i>Social Data Syntax</i>	Q3 2014	Q3 2015	Q4 2015	Q2 2016	Q3 2016
<i>Social API</i>	Q3 2014	Q3 2015	Q4 2015	Q2 2016	Q3 2016
<i>Federation Protocol</i>	Q4 2014	Q4 2015	Q1 2016	Q3 2016	Q4 2016

Note that FPWD is “First Public Working Draft”, LC is “Last Call” (for public comments), CR is “Candidate Recommendation” (i.e. implementations), PR is “Proposed Recommendation,” (call for patent assignments) and Rec is “Recommendation” (full royalty-free standard). These are the formal levels within the W3C Process. As the standards progress, we expect D-CENT to keep track and use or provide open-source implementations of these standards in the final D-CENT platform. As the standards change over time, we expect the D-CENT codebase to keep up. As the standards are supposed to reach CR by end of 2015, the final D-CENT codebase should use stabilized W3C standards.

D-CENT Applications

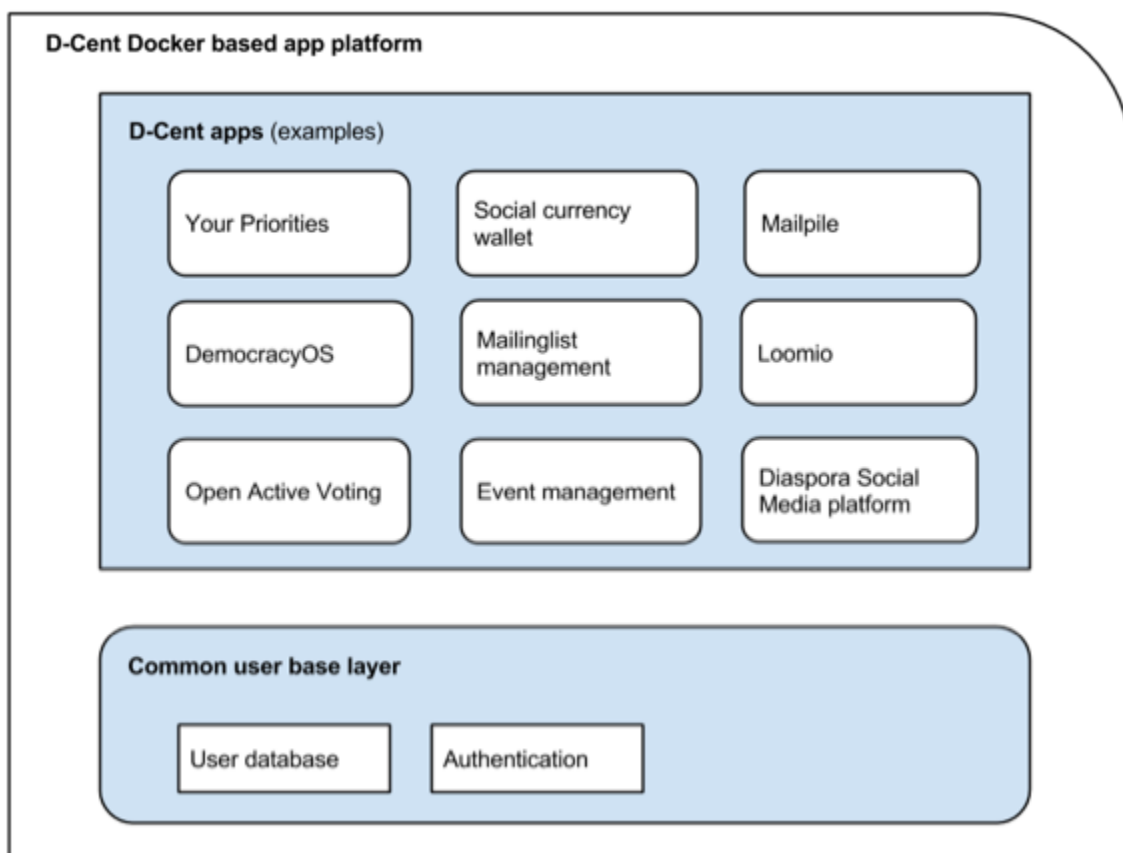


Figure 47 Illustration of how possible D-CENT Applications build on a common D-CENT platform

Although D-CENT will include a number of core features as outlined earlier, the basic idea for the D-CENT platform is to create a set of standard based APIs that enables activists apps to share the same user base on a given D-CENT node. Many types of apps could be D-CENT applications and communicate via these APIs, for example, the use of Your Priorities for debate and prioritisation. Rather

than build competing platforms, applications D-CENT users may find useful like Loomio, Mailpile, Diaspora, and possible social currency applications to be outlined in D4.4 could become D-CENT enabled by communicating via the API interface. This is illustrated in Figure 46. Importantly, it is likely that the social currency pilots will build applications that will use this method to integrate with the larger D-CENT platform. Social currency integration will be fully described in the social currency pilots design lead by Dyne as part of Deliverable D. 4.4

App Directory

Such applications are considered part of the D-CENT's App Directory which provides easy set-up for governments or organizations that want to deploy a D-CENT node with a custom selection of D-CENT apps. For the open source developers interested in D-CENT, there will be a set of APIs to implement and a Docker based app virtualization packaging process. Each D-CENT node can run a selection of apps running as Docker containers, organized and managed through a backend-frontend application running on a standard Linux installation, such as Ubuntu or CentOS. For apps that wants to be compatible with the D-CENT App Directory, they need to create a Dockerfile and support a basic set of APIs and publish the file somewhere publicly, for example on GitHub.

After installing the core system there will be an interface where the D-CENT system administrator can select the applications that they want to deploy in their D-CENT node. Once selected the software will be installed on a self-hosted server or in a cloud-based server. Note that D-CENT will provide an optional single login for all the different applications so that cusers only need to login once in order to use D-CNET apps, although some apps might choose not to use the SSO, for example when electronic id cards are needed for secure voting but still want to make it available on the D-CENT app store with a self-contained user base.

The interface to a database of D-CENT App Dockerfiles is given by a simple to use front-end after the initial installation of a new D-CENT node using Docker. This database of apps can be stored locally or remotely (such as from the D-CENT project website) using a standard JSON format. Each record in the App Directory database has the following columns:

- App name
- Description
- Homepage link
- Version
- Creator
- Date added
- Date updated
- External Dockerfile location

The backend has scripts to manage running Docker images, connect them to the main logger interface, start, restart and stop apps. It manages the Docker interfaces to different databases and other links

between Docker images. This functionality can be re-used from the open source project called Decking (<http://decking.io>) The Main functions of the App Directory will be:

- Browse, search for apps in the App Directory
- Installation and uninstallations of apps
- Start, restart and stop of apps
- System settings
- Links to Kibana-based open--source live system dashboard for monitoring (<http://www.elasticsearch.org/overview/>) so that apps log to the same logging interface that can deliver realtime logging information into the Kibana client through Logstash and Elasticsearch.

Although we plan to feature CKAN and Pybossa as D-CENT applications, the “lean” design process in Iceland have experimenting by creating Dockerfiles and images for running Your Priorities on Docker, and the Icelandic D-CENT team already tested Docker by creating two new Docker apps out of existing open source apps: An IRC server app written in C and and IRC user interface app written in Node.js. The current docker images are available at <https://index.docker.io/u/yrpri/>. Below are the Dockerfiles that were tested, including what programming language was used.

Your Priorities (Ruby on Rails)

- yrpri/base - <https://github.com/rbjarnason/docker-base>
- yrpri/rails - <https://github.com/rbjarnason/docker-rails>
- yrpri/postgresql - <https://github.com/rbjarnason/docker-postgresql> ngIRCd (C++)
- yrpri/ngircd - <https://github.com/rbjarnason/docker-ngircd> KiwiIRC (Node.js)
- yrpri/kiwiirc - <https://github.com/rbjarnason/docker-kiwiirc>

These experiments worked well and Docker proved easy to use different programming languages. A live example of the IRC apps working together is available here: <https://irc.yrpri.org:7778>.

Open Data

Feature: In order for D-CENT users to export, search, and find open data, we will recommend the use of CKAN as a D-CENT enabled application. CKAN (<http://ckan.org>) integration makes it possible to import datasets from any external open data portal which uses CKAN, as well as exporting datasets to any CKAN instance.

Feature Definition: CKAN integration will serve two main purposes. Firstly, it enables users of D-CENT to collaborate on analysing and viewing datasets from external open data portals. Datasets can be pulled in from open data portals which use CKAN, which is the software of choice for most national and many regional governments in Europe. Secondly, it enables communities collecting data to make that data available on a CKAN portal, by exporting it either to their own CKAN instance, or by pushing it to a community CKAN portal such as the DataHub. This could be any kind of data: either data that is relevant to the community because they have requested or collected it, or data about the community’s activity itself.

User Need: Governments are overwhelmingly using CKAN as the software of choice for their open data portal:

- In Finland, The Helsinki Region Infoshare aims to make regional information quickly and easily accessible to all. It is powered by CKAN and WordPress, and has over 900 datasets. The data is mainly statistical, giving a comprehensive and diverse outlook on a variety of urban phenomena, such as living conditions, economics and well-being, employment and transport.
- The Icelandic government are using CKAN for their national open data portal 'Opin Gögn', where key spending data is being published.
- In Spain, several CKAN portals exist which may be relevant. For example, the Markets and Competition Commission (CNMC) decided to release data openly to build transparency and participation, create economic value, and improve the efficiency of the organisation itself. They released a first version of their CKAN portal in October 2013.

Communities of users collaborate on issues that matter to them, and want to reference the datasets published by the authorities to accompany the documents they work on collaboratively. This can be any type of dataset, for example related to finances (budgeting or spending), healthcare data (eg. related to hospitals), or other relevant areas of interest. Data can also be integrated from other portals, to support for example a proposal or a petition from the community. Data-sets can be made available using a wide variety of formats as well to match local user requirements in terms of software.

For exporting data to an open data portal, we distinguish between two types of use cases. The first is to publish data that has been collected, either by collaboratively working on a spreadsheet or by combining data from various sources. The second type is the export of a dataset that contains the data of the community itself. This can be a representation of a feed of a user, which can be exported as described in Section Data Portability, or a representation of the collection of all interaction in a community. The resulting dataset can be exported to an external CKAN open data portal. There are various examples of communities running their own data portal, eg. the DataHub (<http://datahub.io>) or Offene Daten (<https://offenedaten.de/>).

Description: D-CENT communities are using open data portals to find the data that is relevant to their cause:

Peter and Sally are working on a petition to increase road safety in the UK. In order to strengthen their case, the published data from the Department of Transport in the UK is crucial to indicate why safer roads are needed. Peter uses the CKAN importer to link the published data directly from the UK open data portal into their petition. Peter finds the information directly on the site of the UK Government (data.gov.uk). When he has found the relevant datasource he goes to the D-CENT dashboard where he selects the relevant URL: <http://data.gov.uk/dataset/road-accidents-safety-data>. The relevant API call on data.gov.uk to import the data into the system is the following:

http://data.gov.uk/api/3/action/package_show?id=road-accidents-safety-data.

Dependences: Since the focus is on data that one wants available to the public (for non-public personal data, see Section on Data Portability), for importing data, the standard CKAN API can be used which does not require authentication. Therefore there are no additional dependencies with other components. However, when exporting data from CKAN, an authenticated user will need to have a valid CKAN API key for the CKAN portal it wants to export to. This key will need to be registered alongside the user preferences data in the D-CENT node from which the user exports the data. This will be stored as a CKAN URL-Key pair. The feature to export user or community data from a D-CENT node to a CKAN instance depends on the availability of the Data Portability feature (as described in section Data Portability). In order to make it easy for users of D-CENT to use CKAN, we will want to use standard OAuth-based single-sign on to establish new CKAN users.

Open Decisions: At the point of writing, none of the active pilot groups have yet identified the need to integrate data into their system. Therefore, the initial description here is fairly lightweight. With CKAN version 2.0, notifications have been introduced which allows users to be notified whenever a dataset, or a dataset in a certain category, gets updated. If there is a need from one of the pilot D-CENT communities to explicitly work with data from a CKAN open data portal, we can explore whether we can expand the notification mechanism of CKAN to work with the D-CENT notification system. This would make it possible for users to see an updated dataset, or a new dataset in a category relevant to their interests, to appear in their notification stream.

Crowdsourcing

Background

There are various different types of reasons why communities want to apply crowdsourcing techniques. In most cases, there is a large amount of information that needs to be analysed. The information cannot be easily analysed via automation but requires some manual and human step. Additionally, the analysis can be divided up in a large number of discrete steps, or tasks. In such cases, building a crowdsourcing application is an easy way of getting the information analysed by a group of people, who often volunteer. There are many examples of where crowdsourcing has been successfully applied, specifically in scientific projects where there are big projects such as Galaxy Zoo and PlanetHunters, which are now both part of the overarching Zooniverse project.

In D-CENT, we will build on the successful work of the PyBossa project. PyBossa was started in 2012 because the makers recognised that the existing crowdsourcing software platforms were closed source, and it was difficult for people to start their own project. PyBossa was started to make it easier for anyone to create a crowdsourcing project, initially focussing on the scientific community.

As a flagship demonstrator for what PyBossa can do, the platform Crowdcrafting.org has been created. This platform currently contains over 600 projects with tasks ranging from classifying images to transcribing PDF tables. It also contains templates that make it easy for users to create new projects based on them. An example of such a project is Héraðsdómar.

Example project “Héraðsdómar - sýknað eða sakfelld”

An Icelandic citizen group, Gögn, decided to analyse the conviction rates of judges and courts in their country as a consequence of a new published in the mass-media about a judge with a conviction rate of almost 99.4%.

The goal of the application, created on CrowdCrafting.org in spring 2013, was to classify all the judgments in criminal cases available on the website of the district courts. With more than 4,700 cases to review (from period 2006 - 2012), the volunteers completed all the work in one week, becoming one of the most used and popular applications in CrowdCrafting.org in that period. The creator of the application published the results online.

The developer has also created another popular application where the volunteers are invited to improve the bus stops in the country. The goal is to create an alternative application to the official one provided by the company that runs the service, as apparently there are missing stops and some of them are not very well located. With almost 1000 tasks to do, the application was completed in a few hours thanks to the volunteers.

For D-CENT, we will integrate connections with an external PyBossa server to allow communities to integrate their own crowdsourcing projects with their communities.

Feature: PyBossa integration to incorporate crowdsourcing initiatives into D-CENT.

Feature Definition: By integrating with PyBossa it will be possible for D-CENT user communities to apply crowdsourcing methods to how they collaborate. The integration will be light-weight via the PyBossa API – it will not require D-CENT communities to run their own PyBossa server because they can build a crowdsourcing application on an existing PyBossa service such as Crowdcrafting.org.

User Need: Much of the information that user communities work with is unstructured and does not allow for easy analysis. PyBossa, and the hosted service crowdcrafting.org, provides an easy-to-use templated platform that allows anyone to create a crowdsourcing application without needing much technical know-how. This makes it possible for a community to turn unusable documents, such as PDF files that contain information that is not machine-readable, into structured data that can be analysed and used to generate new insights.

Description: D-CENT communities collaborate on a variety of topics. Sometimes they need to analyse a bit task that is not easy to do by one person.

Christine wants to analyse the financial data from the political party because she suspects that there is corruption taking place. She has the data available as a collection of pdf documents that contain large tables with figures which need to be imported into a spreadsheet to enable easy analysis.

Christine uploads the PDF files to a server of her choice. She then creates a Google spreadsheet and adds links to the PDF files according to a predefined structure. She then goes to crowdcrafting.org to create a new project and chooses the template ‘Transcribing documents’ for importing the tasks and setting up the display template.

Christine now has a crowdsourcing project ready to go on crowdcrafting.org.

Transcribe financial figures of party XYZ
by Sander van der Waal

Transcribe the following page

Important This is just a demo project. You can re-use the code to create your own project.

You are working now on task: **761002**

You have completed: **12** tasks from **14**

Write here the transcription

SUBMIT TRANSCRIPTION!

	Loops	Trees	Traces	Aborts	Flashes	Trees/Loop	Traces/Tree	Traces/Loop	Speedup
3d-cube	25	27	29	3	0	1.1	1.1	1.2	2.20x
3d-morph	5	8	8	2	0	1.6	1.0	1.6	2.86x
3d-raytrace	10	25	100	10	1	2.5	4.0	10.0	1.18x
access-binary-trees	0	0	0	5	0	-	-	-	0.93x
access-fannkuch	10	34	57	24	0	3.4	1.7	5.7	2.20x
access-nbody	8	16	18	5	0	2.0	1.1	2.3	4.19x
access-n sieve	3	6	8	3	0	2.0	1.3	2.7	3.05x
bitops-3bit-bits-in-byte	2	2	2	0	0	1.0	1.0	1.0	25.47x
bitops-bits-in-byte	3	3	4	1	0	1.0	1.3	1.3	8.67x
bitops-bitwise-and	1	1	1	0	0	1.0	1.0	1.0	25.20x
bitops-n sieve bits	3	3	5	0	0	1.0	1.7	1.7	2.75x
controlflow-recursive	0	0	0	1	0	-	-	-	0.98x
crypto-aes	50	72	78	19	0	1.4	1.1	1.6	1.64x
crypto-md5	4	4	5	0	0	1.0	1.3	1.3	2.30x
crypto-sha1	5	5	10	0	0	1.0	2.0	2.0	5.95x
date-format-sqlite	3	3	4	7	0	1.0	1.3	1.3	1.07x
date-format-sqlite	3	3	11	3	0	1.0	3.7	3.7	0.98x
math-cordic	2	4	5	1	0	2.0	1.3	2.5	4.92x
math-partial-sums	2	4	4	1	0	2.0	1.0	2.0	5.90x
math-spectral-norm	15	20	20	0	0	1.3	1.0	1.3	7.12x
regex-dna	2	2	2	0	0	1.0	1.0	1.0	4.21x
string-base64	3	5	7	0	0	1.7	1.4	2.3	2.53x
string-flatten	5	11	15	6	0	2.2	1.4	3.0	1.09x
string-tagcloud	3	6	6	5	0	2.0	1.0	2.0	1.09x
string-unpack-code	4	4	37	0	0	1.0	9.3	9.3	1.20x
string-validate-input	6	10	13	1	0	1.7	1.3	2.2	1.86x

Figure 13. Detailed trace recording statistics for the SunSpider benchmark set.

mean). We exclude regex-dna from the following calculations, because most of its time is spent in the regular expression matcher, which has much different performance characteristics from the other programs. (Note that this only makes a difference of about 1.0% in the overall average.)

fastest available JavaScript inline threaded interpreter (SFX) on 9 of 26 benchmarks.

Figure 48 Illustration of PyBossa Integration with the D-CENT platform

Dependencies: The PyBossa integration requires a setting to describe the URL of the PyBossa service. It also requires the user to be able to store her PyBossa API key to integrate the task and taskrun information in their D-CENT activity stream.

Open Decisions: Because crowdsourcing has not been identified as a strong user need, the integration is lightweight and only based on what the external crowdsourcing service already offers. Once a need from a community has been identified to apply crowdsourcing within their community, we will re-evaluate whether we want to build deeper integration with PyBossa. One can image for example that crowdsourcing tasks can be executed without leaving the D-CENT node where the community resides. If we will decide to do that, tasks as defined in section Tasks will be created based on the tasks the PyBossa server has identified. This would also require changes to the PyBossa API, as external task completion is currently not supported by PyBossa.

Developer Community Engagement Methodology

The developer community engagement strategy consists of encouraging local hackers and developers to set-up local D-CENT nodes to answer to existing user needs and interfacing with local municipalities, regional and national authorities to encourage adoption of D-CENT as an open source alternative to commercial providers of citizen engagement (top-down) platforms.

The D-CENT project aims to create wide-scale adoption among developers wishing to set up new participation and collaboration instances. The dockerized decentralized architecture of D-CENT makes it easy for developers and hacktivists to set up new instances easily. At the same time municipalities and other governmental entities, who need to make sure the user content is stored securely and in many cases physically by the authority itself, may find setting up D-CENT instances a viable option when developing new citizen participation and online public consultation projects.

Building an ecosystem where developers can easily use the existing codebase and the shared core functionality of D-CENT is an ideal that can be kept in mind. The D-CENT project's approach is to build usable tools for actual users. If and when the tools gain popularity among early users and the pilots being carried out during the project, it will attract more early developers and hacktivists to set-up their own instances. This will provide more user feedback and also pull-requests to the actual core D-CENT project from a growing developer base. With enough developers projects linked to D-CENT there can, at some point, be enough developers (the open source projects can be active enough) to be considered a viable open source alternative to commercial software providers in the governmental context.

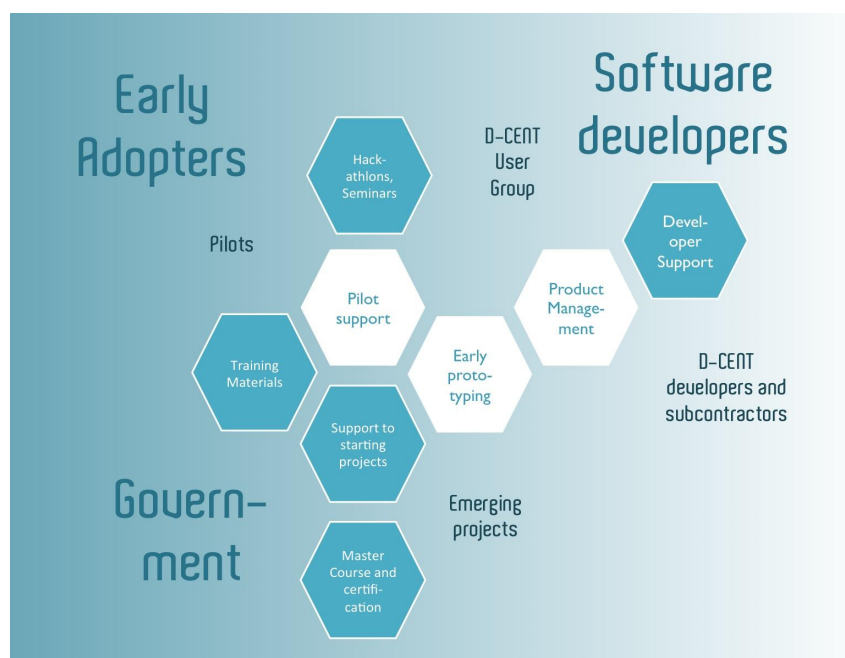


Figure 49 D-CENT Stakeholders Ecology

Engaging Government Entities

Already during the D-CENT project governmental entities are included in some of the pilots to gain experience from working with the public sector and to learn early about their mindset and specific needs. D-CENT wants to provide working examples for local, regional and national governments of how the tools can increase citizen participation and leverage crowdsourcing. The early prototyping with government entities will be showcased in Finland with the cooperation with the Helsinki City. The results will be leveraged to gain peer buy-in in other localities and countries.

The most viable route to gain traction in the public sector is through pilot organisation employees who want to serve their own community. Open-minded civil servants who support early technology-independent prototyping and Open Design should be able to easily pilot new services based on D-CENT. As early prototyping and hopes for open source alternatives becomes more wide-spread in the public sector, D-CENT may become the technology choice for the actual procurements of full-fledged systems.

D-CENT needs to provide support to starting projects, including new publicly funded research and development project that could leverage D-CENT as a software solution for new kinds of user engagement solutions. The D-CENT support team should provide support to civil servants and researchers preparing new projects to make sure D-CENT is as a potential or the actual software solution that will be expanded upon.

Hackathons

For large scale adoption it is crucial to first build momentum with early adopters in various localities. D-CENT should facilitate the arranging of hackathons in the pilot cities and other cities in Europe (and rest of the world) with active open source developer communities. The Open Knowledge network is an asset in this.

Arranging hackathons is not resourced in the current D-CENT project budget, but should be included in future plans. Some hackathons, in as far as they can be arranged with volunteer effort and local sponsors, should be arranged during 2015-2016.

User Groups

D-CENT will continue to adopt a grassroots, lean approach by engaging users groups throughout the development process, and gathering feedback from communities during the trials (to see all the D-CENT users communities involved in the three pilot areas see D2.1). As the D-CENT project evolves and attracts more developers and new groups of users, it should consider arranging itself into thematic development working groups with different approaches: Open data, Front-end, Crypto-currency, Decision-making for example.

Training Materials and Master Course

In order to be able to compete with commercial platform providers in the longer run, local IT solution providers need to have access to qualified software developers with experience and qualifications with the D-CENT software. To this end, D-CENT should produce open training materials, instruct local partners on how to arrange training activities and establish a framework for a D-CENT certified software developer accreditation. Developers who have added the D-CENT Master Course to their CV and are listed, if they want, on the D-CENT project site, would provide a pool of qualified developers for IT solution providers to tap into. A large enough pool of qualified resources is crucial for commercially scaling D-CENT-based solutions.

The training business could provide additional revenue streams to organizations working with the D-CENT software, but no monopolies will be created around it. The D-CENT philosophy of openness should apply to the teaching materials and the courses, with peer-reviews and social appraisal as key components of the accreditation and training processes, i.e. anyone with the necessary skills should be able to provide D-CENT Master Courses and listed on the D-CENT website.

Developer Support and Product Management

Open Source developers internationally are encouraged to branch the D-CENT software and to contribute to the main branch. As developers set up D-CENT nodes in different countries and contexts and develop their own use cases and user interfaces, the main software project is expected to receive increasing numbers of pull requests from developers who wish to contribute to the main project. This

means, that in the medium and longer run, D-CENT will need to tackle interoperability and integration issues.

References

- Antonopoulos, A. (2014). *Mastering Bitcoin*. Booktastic Distributor.
- Amsler, T. (2004). *Special issue on Deliberative Democracy*. Vol. 93 #4 of *National Civic Review*. Wiley
- Barber, B. R. (1984). *Strong Democracy: Participatory Politics for a New Age*. University of California Press
- Bellare, M; Shi, H; Zhang, C (2005). A Menezes, ed. "Foundations of Group Signatures: The Case of Dynamic Groups". *Topics in Cryptology - CT-RSA 2005 Proceedings. Lecture Notes in Computer Science (Springer-Verlag)* 3376.
- Brams, S. (1993). *Approval Voting and the Good Society in Political Economy of the Good Society Newsletter* 3, no. 1: 10-14.
- Brams, S. J. (1993). *Theory of moves*. *American Scientist*, 562-570
- Brams, S. J., & Fishburn, P. C. (2002). *Voting procedures*. *Handbook of social choice and welfare*, 1, 173-236.
- Brams, S. J., & Fishburn, P. C. (2007). *Approval voting (pp. I-XXI)*. New York: Springer.
- Elster, J. (1998). *Deliberative Democracy*. Cambridge UP
- Kahane, D., Weinstock, D., Leydet, D., Williams, M. (2010). *Deliberative Democracy in Practice*. UBC Press.
- McCarthy, S. (2013). *Demystifying Complexity: Why Worse is Better in Voting*.
- Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system*. Consulted, 1(2012), 28 .
- Pivato, M. J. (2009) *Pyramidal Democracy*, in *Journal of Public Deliberation*: Vol. 5: Iss. 1, Article 8.
- Shamir, Adi (1979), *How to share a secret*, *Communications of the ACM* 22 (11): 612–613
- Warren, M. E., March 1996. *Deliberative democracy and authority*. *American Political Science Review* 90 (1), 46–60.