Technical specifications and primer on interoperability

Decentralised Citizens ENgagement Technologies

Specific Targeted Research Project Collective Awareness Platforms

FP7 - CAPS
Project no. 610349
D-CENT
Decentralised Citizens
ENgagement Technologies

Lead beneficiary: W3C

D5.8 Technical specifications and primer on interoperability

May 2016
Version Number: 6

Authors:
Irina Bolychevsky
Natalie Eskinazi
Amy Welch
Felicity Moon

Editors and reviewers:
Jaromil Rojo
Jaakko Korhonen
Francesca Bria
Orpa Haque

The work leading to this publication has received funding from the European Union's Seventh Framework Programme (FP7/2007 - 2013) under grant agreement n° 610349.

The content of this report reflects only the author's view and that the Commission is not responsible for any use that may be made of the information it contains.

Project no. 610349

D-CENT

Decentralised Citizens ENgagement Technologies

Specific Targeted Research Project

Collective Awareness Platforms

D5.8 Technical specifications and primer on interoperability

Version Number: V1
Lead beneficiary: W3C
Due Date: 31 May 2016
Author(s): Irina Bolychevsky, Natalie Eskinazi, Amy Welch, Felicity Moon
Editors and reviewers: Jaromil Rojo, Jaakko Korhonen, Francesca Bria, Orpa Haque

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 May 2016

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

Contents

1. Introduction

D-CENT (Decentralised Citizens ENgagement Technologies) is a Europe-wide project bringing together citizen-led organisations that have transformed democracy in recent years, and helping them to develop the next generation of open source, distributed, and privacy-aware tools for direct democracy and economic empowerment.

To realise this vision, we have been building open source tools (where the code is available and you have the right to use and modify it for your needs) that utilise open standards to grow an ecosystem of applications that work together.

Open standards are non-proprietary (you do not need a license or pay to use) specifications for a protocol (or vocabulary or application action) that allow different programs to talk to each other and work together automatically - simply by each of them conforming to the standard. This supports wide interoperability - allowing loosely coupled separate tools to work together, but decentralised.

Decentralisation is an incredibly powerful way to push the control of applications and data back towards the edges - to the users and the communities that maintain infrastructure and applications. It allows diversity to thrive and decentralised power structures to exist. This enables new voices to be heard and decisions to be made in an inclusive and a collaborative way - which is key for democracy and citizen engagement and a core part of our work.

As part of this project, ThoughtWorks, Open Knowledge, Open Knowledge Finland, Citizens Foundation Iceland and Dyne have built a set of open source tools with interoperability in mind. They have been designed in a modular way to allow an organisation to choose any combination of the tools that is useful for their campaign or organisation. These tools can not only fit well together, but in addition, be easily used with applications and tools outside the D-CENT ecosystem. To further facilitate maximum interoperability and best practices, we worked with the World Wide Web Consortium (W3C) to develop a new open standard, Activity Streams 2.0.

NESTA have been running the project, funded by the European Commission. For more information about D-CENT, visit the website: http://dcentproject.eu/

2 Standardisation work at the W3C

Most W3C work revolves around the standardization of Web technologies. To accomplish this work, W3C follows processes that promote the development of high-quality standards based on the consensus of the community. W3C processes promote fairness, responsiveness, and progress, all facets of the W3C mission: to lead the World Wide Web to its full potential by developing protocols and guidelines that ensure the long-term growth of the Web.

2.1 Overview of how W3C standardizes a Web technology

In many cases, the goal of the W3C process is a W3C Recommendation, the W3C equivalent of a Web standard.

1. People generate interest in a particular topic (e.g., Web services). For instance, Members express interest in the form of Member Submissions, and the Team monitors work inside and outside of W3C for signs of interest. Also, W3C is likely to organize a Workshop to bring people together to discuss topics that interest the W3C community. This was the case, for example, with Web services.
2. When there is enough interest in a topic (e.g., after a successful Workshop and/or discussion on an Advisory Committee mailing list), the Director announces the development of a proposal for a new Activity or Working Group charter, depending on the breadth of the topic of interest. An Activity Proposal describes the scope, duration, and other characteristics of the intended work, and includes the charters of one or more Working Groups, Interest Groups, and possibly Coordination Groups to carry out the work. W3C Members review each Activity Proposal and the associated Working Group charters. When there is support within W3C for investing resources in the topic of interest, the Director approves the new Activity and groups get down to work. For the Web Services Activity, the initial Activity Proposal called for one Working Group to work on Web Services Architecture and one to work on a language for Web Services Description. The Activity Proposal also incorporated an existing Working Group (from another Activity) working on XML Protocols.
3. There are three types of Working Group participants: Member representatives, Invited Experts, and Team representatives. Team representatives both contribute to the technical work and help ensure the group's proper integration with the rest of W3C. The Working Group charter sets expectations about each group's deliverables (e.g., technical reports, test suites, and tutorials).
4. Working Groups generally create specifications and guidelines that undergo cycles of revision and review as they advance to W3C Recommendation status. The W3C process for producing these technical reports includes significant review by the Members and public, and requirements

that the Working Group be able to show implementation and interoperability experience. At the end of the process, the Advisory Committee reviews the mature technical report, and if there is support, W3C publishes it as a Recommendation.

The Process Document promotes the goals of quality and fairness in technical decisions by encouraging consensus, requiring reviews (by both Members and public) as part of the technical report development process, and through an appeal process for the Advisory Committee.

2.2 Value of following the W3C process for standardisation

W3C standards:

- are created following a consensus-based decision process;
- consider aspects of accessibility, privacy, security, and internationalization;
- reflect the views of diverse industries and global stakeholders;
- balance speed, fairness, public accountability, and quality;
- benefit from Royalty-Free patent licensing commitments from participants;
- are stable (and W3C seeks to ensure their persistence at the published URI);
- benefit from wide review from groups inside and outside W3C;
- are downloadable at no cost;
- are maintained in a predictable fashion;
- are strengthened through interoperability testing.

2.3 General benefits of open standards

There has been extensive documentation, research and thinking on the benefits and importance of open standards:

- An Economic Basis for Open Standards
- ISO Studies on benefits of standards including Assessing Economic Benefits of Consensus-Based Standards - The ISO Methodology (2010)
- Long live the web (Scientific American, Nov 2010)
- The Economic Importance of Standards (slides), Tim Berners-Lee
- A Standards Quality Case Study: W3C, Arnaud Le Hors
- Business Case for Open Standards, Erik Silman
- The Business Value of Web Standards, Jeffrey Veen
- What are the advantages of using standards?, Web Standards Project
- Business Case for Web Standards Wiki
- Developing a Web Accessibility Business Case for Your Organization: Overview , W3C Web Accessibility Initiative
- Making the business case for web standards, Web Access Strategies
- CSS Talking Points: Selling Clients on Web Standards, Greg Kise

- Asia-Pacific Economic Cooperation (APEC)'s Standardization: Fundamentals, Impact, and Business Strategy
- Standards <Meta> Library

3 D-CENT and standards work

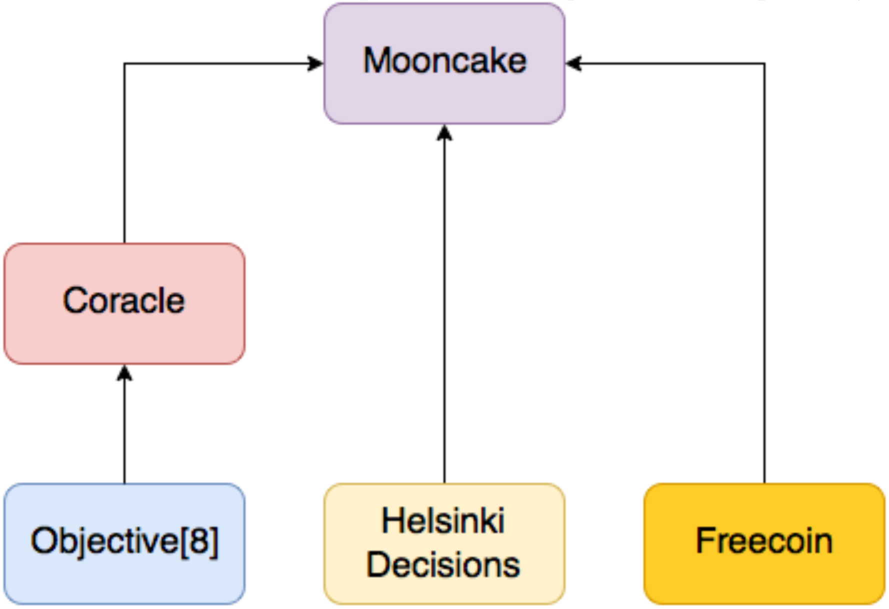
D-CENT tools use a number of standards that are documented in **section [?]** of this primer, two of these are W3C standards. One: Activity Streams have been supported and developed as part of this project as documented in D6.6, and the other: Web Components, is a new area of work within the W3C that has emerged during the course of the D-CENT project.

Standards take time and the key benefits of standardisation come about through adoption. So one of the key focuses of this work have been around building pilots and growing adoption of this work. Part of the W3C process also involves collecting implementations of standards before a fully-fledged W3C Recommendation.

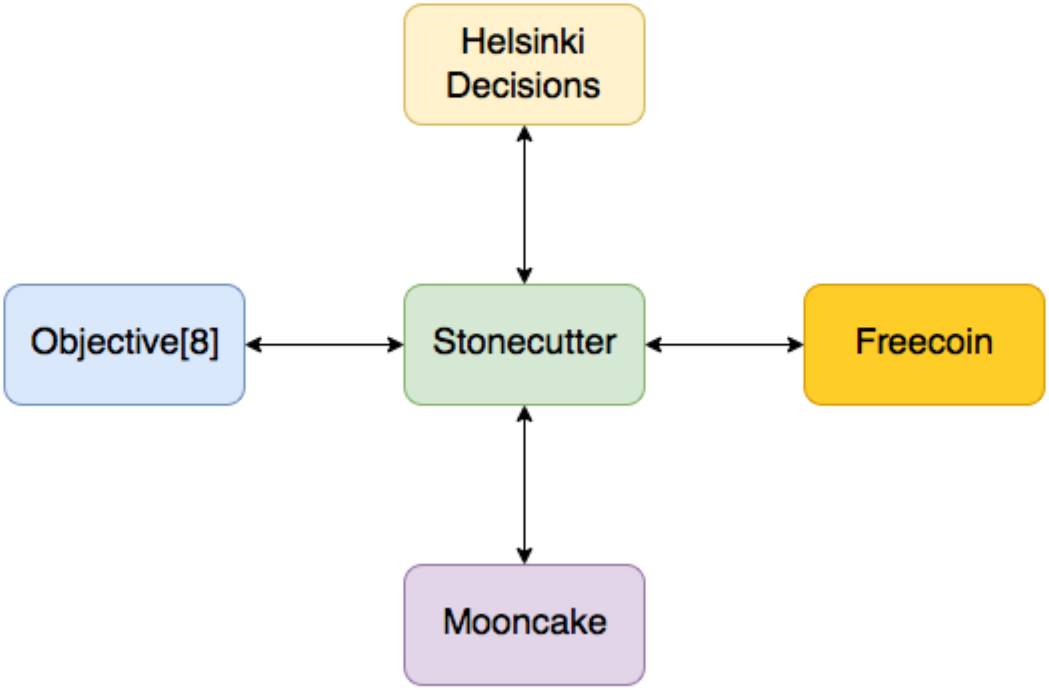
In this Primer, we present each of the open and interoperable tools we've developed with context, use cases and links to the code and documentation for installing, as well as the open standards to which they conform. We then discuss all the standards used, what they are, the specification and examples for each and then how the D-CENT tools interacts with the standard - giving you clear examples to follow for reusing these tools and incorporating them into your technical architecture.

4. D-CENT Tools

The D-CENT tools have been developed with a federated architecture based on open standards, open APIs and a shared identity system to allow for the growth of an eco-system of modular, interoperable and decentralised tools.



Communication using the Activity Streams 2.0 standard



Communication using the OpenID Connect standard

4.1 Objective8

Objective8 is a policy drafting tool that allows organisations to work with their members to produce crowd sourced policy/manifestos. Objective8 publishes notifications to Mooncake using the Activity Streams 2.0 and JWS standards.

Open Standards used: OpenID Connect, Activity Streams 2.0

Example use cases:

- A political party wants to produce a new edition of their manifesto. They choose to involve the entire community of members in drafting it.
- An organisation wants to update internal policy based on the values of their employees. They use Objective8 to gather opinion on existing policy and enable their employees to be directly involved in drafting new policy.

Used by:

- New Garden Cities Alliance - A platform to crowd source a definition of a Garden City
- West Midlands People's Plan - A platform to enable citizens to help draft a manifesto for the mayor of the West Midlands, UK
- ThoughtWorks - The LGBT+ group in ThoughtWorks are using the tool to collaboratively update their policy based on the results of the Stonewall Index

Live Demo: https://objective8.dcentproject.eu/

Open Source Code: https://github.com/d-cent/Objective8

4.2 Freecoin

A toolkit to build social wallets operated via a simple and customisable user interface as well a REST API. Freecoin is a middleware between multiple blockchain implementations, it is also designed to work offline, packaging and communicating transaction information to distributed ledgers of choice.

Open Standards used: OpenID Connect, Activity Streams 2.0, FXC (proposed standard)

Example use cases:

- Public Administrations, SMEs, NGOs, the financial services industry that want an internal audit management tool that can handle timesheet/accounting and payroll.

Used by:

- Organizations internal agile management, Freecoin is a timesheet/accounting and payroll and auditing tool that can serve Public Administrations, SMEs, NGOs, the financial services industry at large;

- Self-remuneration (D-CENT Finnish pilot);
- Micro-endorsement management (D-CENT Spanish pilot);
- Reward system (e.g. for political participation to the betterment of a common good or community) (D-CENT Icelandic pilot);
- Electronic Voucher Schemes (D-CENT use case Milan)
- Accounting system for web of devices and services in the Internet of Things

Open Source Code: https://github.com/d-cent/freecoin

4.3 Stonecutter

Federated and privacy aware user management for organisations. Mooncake integrates with Stonecutter to authenticate users, using OpenID Connect.

Open Standards used: OpenID Connect, JWS, VCards

Example use cases:

- An organisation wants to provide multiple websites to their community with a single account for all of them.
- An organisation wants to remove the added requirements of securely maintaining user passwords by outsourcing them to another application.

Used by: Garden Cities, People's Plan, ThoughtWorks

Live Demo: https://sso.dcentproject.eu/

Open Source Code: https://github.com/d-cent/stonecutter

4.4 Mooncake

Securely notify your members of events/activity on your D-CENT ecosystem.

Open Standards used: OpenID Connect, JWS, Activity Streams 2.0

Example use cases:

- A user is active in multiple decentralised websites and wants a single feed from all of them for updated content such as new posts, media, status updates or comments.
- An admin wants to get an idea of activity across their D-Cent ecosystem.

Live Demo: https://mooncake.dcentproject.eu/

Open Source Code: https://github.com/d-cent/mooncake

4.5 Your Priorities

Citizen social network application.

Open Standards used: Web Components, HTML, Ecmascript

Example use cases:

- Better Reykjavik. The city of Reykjavík has been using Your Priorities since 2010 both for getting citizens voices heard at city council meetings and to gather and priorities ideas for participatory budgeting.
- Rahvakogu People's Assembly in Estonia. After political scandals in Estonia in 2012, grassroots organisations with official ties lead a law reform project. Ideas were gathered through Your Priorities which was installed and modified locally. Over 50.000 people took part and submitted over 2000 proposals. The president of Estonia submitted the top 15 ideas to the parliament. Seven of the ideas have become Estonian law.
- Pirate Party annual meeting 2015. The Icelandic Pirate Party (5% in parliament, 30% in polls 6 months before next elections) called out to its members to find their most important common priorities.
- Better Pula participatory budgeting in Croatia. The environmental e-democracy and e-participation platform is a democratic participatory tool that enables the electronically literate citizens of Pula to be environmentally and politically active in a simple way. The platform is focused on environmental and nature protection issues, as well as on the issues of spatial planning and other policies and practices of local authorities which strongly influence the everyday environment and quality of life of local citizens.
- Better Left Green is a democracy forum where members of the Left-Green have the opportunity to put forward ideas about politics, discuss them with other members and bring them forward for implementation within the organization. The forum is open to all registered members of the movement to participate in. (Left Greens are currently polling 20%, 6 months before the next election)
- Better Maribor participatory budgeting in Slovenia. The project Ubrana skupnost, as carried out by the Association for support of Radio MARŠ, pursues as its primary goal the political activation of the population of Maribor. It aims to raise the awareness of citizens' rights and achieve an improvement in democratic culture in the city.
- Our Kopavogur 2016 in Iceland. Gathering and prioritizing ideas for participatory budgeting event

Live Website: https://yrpri.org/

Open Source Code: https://github.com/rbjarnason/your-priorities-app

4.6 Open Active Voting

Secure online budget voting. Encrypts the ballot on the client side.

Open Standards used: HTML, RSA, Ecmascript

Use cases:

- Has been used for the past 4 years in the annual city of Reykjavik participatory budgeting event where citizens submit proposals that are costed and Open Active Voting is used vote on the proposals in a secure and binding vote.
- Can be used for voting games for any sort of budgets.

Live Demo: https://ktest.betrireykjavik.is/votes/authentication_options

Open Source Code: https://github.com/rbjarnason/open-active-voting

4.7 Consul

Citizen participation platform to host discussions online and make/edit and vote on proposals, citizens surveys and collaborative writings

Open Standards Used: HTML5, CSS3, Javascript, PNG/SVG

Use cases:

- Has been developed and used by City of Madrid on several participation processes as Citizen proposals, Debates, participatory budgets, Urbanism participatory processes, collaborative legislation and accountability.
- Has been used, modified and adapted by the City of Barcelona to host the strategic planification of the city (2016-2019) combining digital proposals and physical meetings and debates.
- It has been used for other cities as Oviedo or Political parties as Podemos, to gather, vote and discuss citizen proposals.

Used by: https://decide.madrid.es/, https://decidim.barcelona/, http://www.consultaoviedo.es/, https://plaza.podemos.info/

Website: https://decide.madrid.es/

Open Source Code: https://github.com/consul/consul, https://github.com/AjuntamentdeBarcelona/decidim.barcelona

4.8 Helsinki Decisions

Helsinki decisions tools lets you get push notifications of municipal decisions, participate in decisions that involve their municipality, and crowdsource better content to the decision process.

Open Standards used: RSS, OpenID Connect, JWS, Activity Streams 2.0, OpenAhjo

Example use cases:
- Citizens want to follow, comment and take action on municipality or parliament decision-making real-time.
- Municipality or parliament wants to follow, comment and take action on citizen discussion on issues in the municipality decision-making real-time.

Live Demo: https://github.com/okffi/decisions
Open Source Code: http://decisions.okf.fi/
Used by: Open Knowledge Finland

5 Open Standards

5.1 Activity Streams 2.0

Activity Streams 2.0 is a standard format for describing an action performed by a user of some application. It allows other applications to consume this data from a variety of sources with the confidence that all the data exists in the same structures. Some examples of actions represented by this standard are: “Lucy added a post to her blog”, “Andrew viewed the article”, and “Rebecca has rejected the invitation from John”.

The Activity Streams 2.0 specification (http://www.w3.org/TR/activitystreams-core) is currently being defined by W3C as a standard for providing semantic descriptions of user actions on social media platforms. It incorporates the JSON-LD (JSON for Linked Data) format. Activity Streams 2.0 was chosen as the data format for notifications in the D-CENT platform. Adoption of this standard in this piece of software creates the potential for promoting and refining the standard to encourage further use in the open source community, and therefore facilitate integration with future components that provide other digital social functions.

5.1.1 Consuming Activity Streams 2.0

To consume another application’s Activity Stream, you only need to send a request to the endpoint which returns the Activity Streams Document. You can then parse and use this data as you wish.

One such example is a tool ThoughtWorks have built called Mooncake, described below.

There has also been a test suite set up to test Activity Streams 2.0 parsers https://github.com/w3c-social/activitystreams-test-documents.

5.1.1.1 Mooncake

Mooncake is a notifications tool that aggregates Activity Streams 2.0 data from different sources into a feed. This feed updates as new data is published so that users are aware of any activity in their network. This network can consist of multiple apps, including both those within the D-CENT ecosystem and additional custom tools, provided they publish data in the same format.

Mooncake expects activities in the following format:

```
{
 @context:      "http://www.w3.org/ns/activitystreams",
 published:     "2015-12-18T14:25:40.240Z",
 type:          "Create",  --- used to customise the feed
 name:          "Created content"
 object: {
   url:              "http://Objective8.dcentproject.eu/objectives/41/questions/28",  (optional)
   name:  "I am an objective",
   type:             "Objective"  --- used to set the action text
 },

 actor: {
   name:  "Jane Doe"
 },

 target:  (optional) {
   url:              "http://Objective8.dcentproject.eu/objectives/41",  (optional)
   name:  "This Objective"
 }
}
```

The published property uses the format YYYY-MM-DDThh:mm:ss.sZ as specified in the international standard ISO 8601.

Mooncake expects the activity source to support the to-and-from query parameters, returning activities with a published time less than or greater than the provided timestamp respectively. For example, https://Objective8.dcentproject.eu/as2/activities?from=2015-12-22T14:00:00.000Z should return all activities that occurred after 2pm on the 22nd December 2015.

5.1.2 Publishing Activity Streams 2.0

To publish the data from your application as an activity stream, you will need to expose an endpoint. This endpoint should return an Activity Streams Document containing the data in the Activity Streams 2.0 format, represented as JSON, and with a MIME media type of application/activity+json.

To convert the data from your application into the Activity Streams 2.0 format, you will need to choose an appropriate Activity Type and set values for the associated properties. You can adapt the specification to your specific use case (choosing a new activity type, leaving out or adding in properties), but interoperability may suffer as a result. You can find some examples here and a validator here.

5.1.2.1 Objective8

Objective8 uses another tool we built, Coracle, to publish its Activity Streams 2.0 data. It pushes all user activity to Coracle in the Activity Streams 2.0 format. Currently this comprises of any objectives which have been created, or any questions which have been asked. Coracle then publishes this activity data in an externally accessible API which can be read by Mooncake or any other consuming app.

You can access the API of the live demo here: https://Objective8.dcentproject.eu/as2/activities

Some example data from Objective8:

```
{"@context": "http://www.w3.org/ns/activitystreams",
 "type": "Collection",
 "name": "Activity stream",
 "totalItems": 3,
 "items":[{
   "object": {
          "url": "http://Objective8.dcentproject.eu/objectives/56",
          "content": "maybe there are too few things that aren't objectives",
          "name": "An Objective about not-Objectives",
          "type": "Objective"
   },
   "actor": {
          "name": "Felicity",
          "type": "Person"
   },
   "published": "2016-02-22T14:16:06.698Z",
   "type": "Create",
   "@context": "http://www.w3.org/ns/activitystreams"
}, {
   "object": {
          "url": "http://Objective8.dcentproject.eu/objectives/55",
          "content": "Very important objective.",
          "name": "Headline!",
          "type": "Objective"
   },
   "actor": {
          "displayName": "a",
          "type": "Person"
   },
   "published": "2016-02-22T14:07:52.112Z",
   "type": "Create",
   "@context": "http://www.w3.org/ns/activitystreams"
}, {
   "object": {
          "object": {
                 "name": "Make Hackney a safer and flourishing community.",
                 "type": "Objective"
          },
          "url": "http://Objective8.dcentproject.eu/objectives/51/questions/40",
          "name": "How would we do this?",
            "type": "Objective Question"
    },
    "actor": {
            "name": "natalie456",
            "type": "Person"
    },
    "published": "2016-02-15T14:51:37.454Z",
    "type": "Question",
           "name": "Questions",
    "@context": "http://www.w3.org/ns/activitystreams"
}]
```

5.2.2.2 Freecoin

Freecoin published Activity Streams 2.0 compliant collections directly from its /activities endpoint, accepting a query format that contains a starting and ending date for the range of activities requested, according to the way Mooncake works. Such collections are also formatted in JSON format as the rest of D-CENT tools and comprise of 3 objects contained in the main “Transaction” object:

- Actor: the sender of the transaction (Person or Project) identified by name
- Target: the recipient of the transaction (Person) identified by name
- Object: the amount and type (currency) of the transaction, with an url pointing to it

Here below is an example:

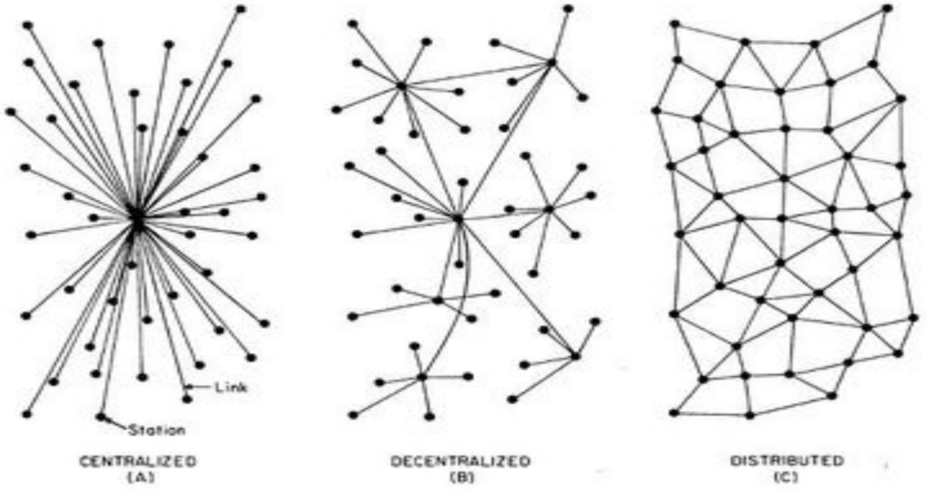
```
{
     "@context":"https:\/\/www.w3.org\/ns\/activitystreams",
     "type":"Transaction",
     "published":"2016-03-19T17:02:14.133Z",
     "actor":{
       "type":"Person",
       "name":"alice"
     },
     "target":{
       "type":"Person",
       "name":"gino"
     },
     "object":{
       "type":"NXT",
        "name":"50",            "url":"http:\/\/localhost:8000\/transactions\/2016-03-
19T17:02:14.133-FXC1_STUB_0e83388e-ccda-4fae-8a55-b23336163 1de_FXC_055ec14a-cb12-
43f8-b69f-be8068e7650e"
      }
  }
```

Please note that all slashes and HTML elements in the url are already escaped by Freecoin.

5.2.2.3 Open Data Crowdsourcing

Helsinki D-CENT Decisions publishes Activity Streams 2.0 as Open Data back to the city, with keywords and comments added to the data by the citizens. Data is read from multiple sources, indexed, refactored and harmonised. Users get to comment on the data, and add keywords i.e. hashtags. Then it is published for the municipality decision-makers and civil servants to participate in following the discussion. This gives city substantial visibility to which issues carry relevance to the civic community, and enables them to direct their efforts to a more meaningful creation of participation data.

Publishing Open Data through an Activity Stream API enables a decentralised architecture where there is a node to publish each city's data, and organisations representing the users join the network. When using Activity Streams with compliance to Open Data publishing policies, open participation can be built in a multi-channel, multimodal solution, where different organisations have different user stories for the same data. Regardless if they are civic organisations, supporting the political representatives, or public offices.



Network srtuctures

There is a project called 6aika in place by the municipalities of the five next biggest cities in Finland to reproduce the Helsinki setup with the same standard. Namely, Turku, Tampere, Oulu, Vantaa and Espoo.

Installation instructions for a node are in the respective Github repository.

5.3 JSON Web Signature (JWS)

JWS is a standard for signing content. The output is a base64 encoded message separated by periods into 3 parts: a header, the body, and the signature. The signature is generated by applying an asymmetric hash algorithm to the body of the message, combined with the private key of the message publisher. The message publisher’s corresponding public key is used by a consumer to verify the message signature. It is used to ensure the integrity of the message and the identity of the message sender.

5.3.1 An example signature

Header
{“typ”: “JWT",
 “alg”: “HS256”}

Base64Url encoded header
eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1NiJ9

Body
{“sub”: “1234567890”,
 “name”: “John Doe”,
 “admin”: true}

Base64Url encoded body
eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9

Signing input (encoded header + “.” + encoded body)
eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9

Signature
Run the algorithm specified in the header (in this case it is the HMAC SHA256 algorithm) on the signing input using the sender’s private key.

Base64Url encoded signature
TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

JWS representation (encoded header + “.” + encoded body + “.” encoded signature)
eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiYWRtaW4iOnRydWV9.TJVA95OrM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ

You can find more details about how JWS is implemented here.

5.3.2 Tools using JWS

5.3.2.1 Mooncake

Mooncake uses JWS to visualise the integrity of the notifications that it receives. It shows a warning symbol to the user next to every notification that isn't signed or that is signed incorrectly.

5.3.2.2 Coracle

The externally accessible API that Coracle uses to publish its data supports the parameter signed. If this parameter is set to true, the data is signed as a JWS using the public/private keypair generated on startup. The public key is published with the signed data.

5.3.2.3 Stonecutter

Stonecutter ensures the authenticity of the user's data by encoding the user information as a JWS. This way, the client applications know that the information returned to them was not tampered with. Client applications can decode the user information using Stonecutter's public key, which is published in its API.

5.4 Web Components

Web Components are a set of standards currently being produced as a W3C specification that allow for the creation of reusable widgets or components in web documents and web applications. The intention behind them is to bring component-based software engineering to the World Wide Web. The components model allows for encapsulation and interoperability of individual HTML elements.

There are 4 separate specs in this set of standards:

Custom Elements

This specification describes the method for enabling the author to define and use new types DOM elements in a document.

HTML Imports

HTML Imports are a way to include and reuse HTML documents in other HTML documents.

Templates

This specification describes a method for declaring inert DOM subtrees in HTML and manipulating them to instantiate document fragments with identical contents.

Shadow DOM

This specification describes a method of establishing and maintaining functional boundaries between DOM trees and how these trees interact with each other within a document, thus enabling better functional encapsulation within the DOM.

5.4.1 Tools implementing web components

Currently the only D-CENT tool that uses Web Components is Your Priorities, but Open Active Voting will be updated to use Web Components in the next few months.

Your Priorities choose to use Web Components to be able to support an app like experience on mobile phones. Increasingly citizens are using mobile phones for citizen participation and web components provide a simple way for enabling app like experience on mobile phones. The use of Web Components also greatly reduced development time.

Web Components are essentially HTML elements so anybody that knows HTML can use web components. These are not a new framework or library, these are standards that are baked into the web browsers and now and in the future will just be a normal part of developing HTML based web pages.

Other websites can use web components from Your Priorities, for example for embedding ideas. Here is a simple example that will embed an idea with the database ID of 1 from the Better Reykjavík website:

```
<yp-post id="1" host="www.betrireykjavik.is"></yp-post>
```

5.5 OpenID Connect

The OpenID Connect standard is an industry standard authentication protocol. It adds an identity layer to the OAuth2 authentication protocol. When successfully issuing an authorisation token to the client, this layer signs the user's details using the JWS standard discussed above. This provides a mechanism for identifying a user and providing basic information about them through an authorisation server (e.g. Stonecutter).

Benefits:

- OpenID Connect can be used as a single sign-on, allowing a user to be authenticated across multiple applications after only signing in once.
- Allows website developers to authenticate users without taking on the responsibility of storing and managing passwords securely.

A sample user flow would go as follows:

- The user visits your website and wants to perform an action that requires them to be signed in.
- They are redirected to the authentication server and logged in.
- The user consents to sharing information with your website.
- They are redirected back to your website with an authorisation code. In the background, your website communicates with the authentication server. It uses the authorisation code to make sure that the user really did sign into the authentication server. The user can then proceed to use your website.

You can find more details about how this is implemented on the OpenID website.

5.5.1 Connecting to your application using Stonecutter

Stonecutter acts as the authorisation server in the D-Cent ecosystem. To configure an existing application to use Stonecutter for authentication you will first need to deploy an instance of Stonecutter. You should then sign into Stonecutter as an admin and add your application as a client app. This will give you a client id and a client secret. You will need to add two endpoints to your application - a sign-in endpoint and a callback endpoint.

You will then need to implement the sign in flow as follows:

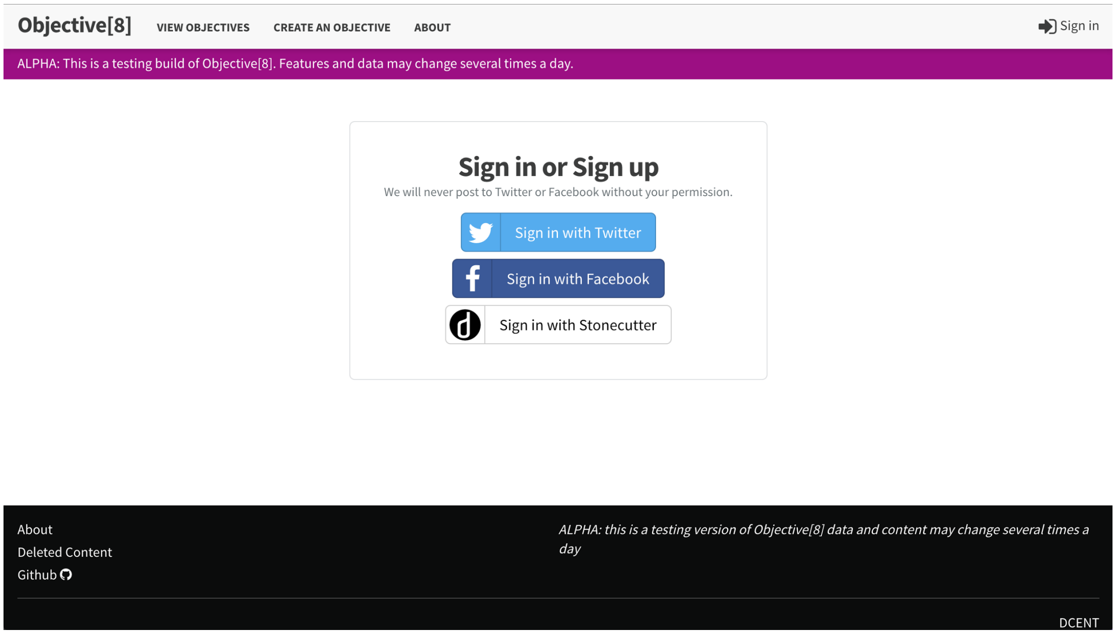
- When a user wants to sign into your application, you should redirect them to the Stonecutter instance at:

<stonecutter url>/authorisation?clientid=<client id> &response_type=code&redirect_uri=<callback url>&scope=openid

- The Stonecutter instance will make a request to <callback url> with either an authorisation code in the parameters or an error. If there is an error in the parameters, this could indicate that the user didn't give permission for your application to access their profile card or that there is an error in the Stonecutter configuration.
- After receiving the authorisation code, make a post request to <stonecutter url>/api/token with form parameters:
    - grant_type="authorization_code"
    - redirect_uri=<callback url>
    - code=<authorisation code>
    - client-id=<client id>
    - client-secret=<client secret>
- This request will return a response in JSON format. It will contain the keys token_type, access_token and id_token.
- You should then make a request to <stonecutter url>/api/jwk-set. This will contain Stonecutter's public key.
- Decode the id token using this public key. It will contain the user's email, user id, role and whether their email has been verified.

5.5.2 Objective8 & OpenID

When a user tries to log into Objective8 they are directed to this page



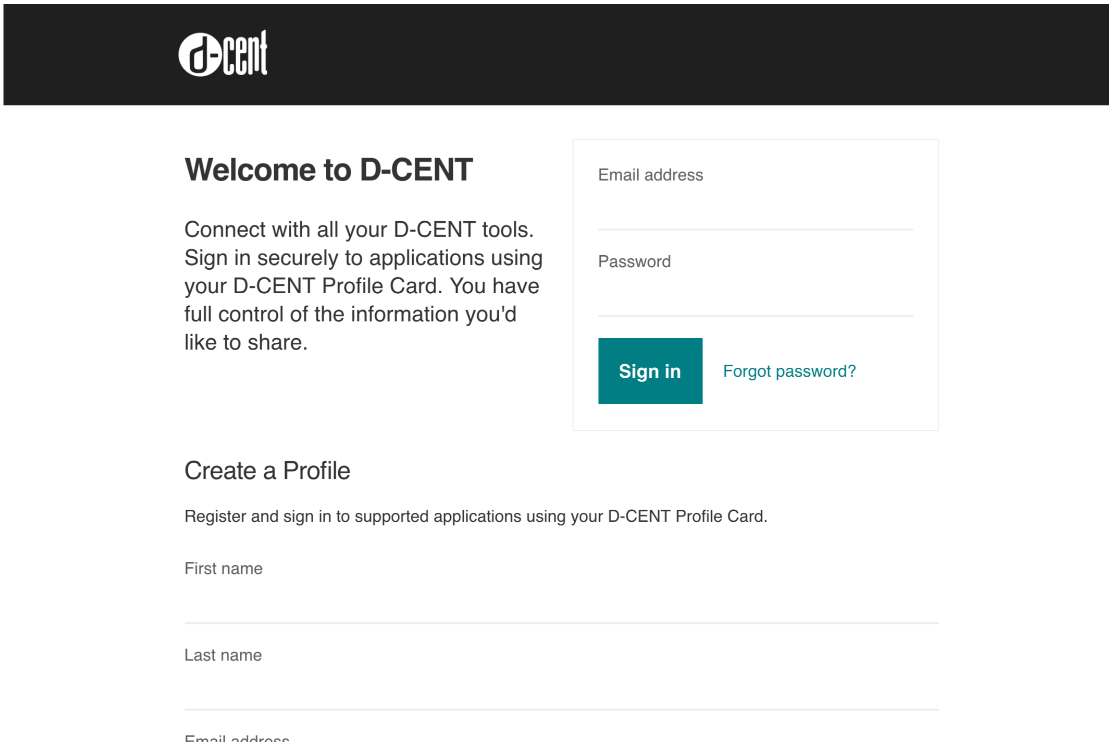
ALPHA: This is a testing build of Objective[8]. Features and data may change several times a day.

Sign in or Sign up

We will never post to Twitter or Facebook without your permission.

ALPHA: this is a testing version of Objective[8] data and content may change several times a day

The user then can choose which service they want to sign in with. If they choose to sign in with Stonecutter then they are redirected there, and prompted to sign in if they have not already done so.



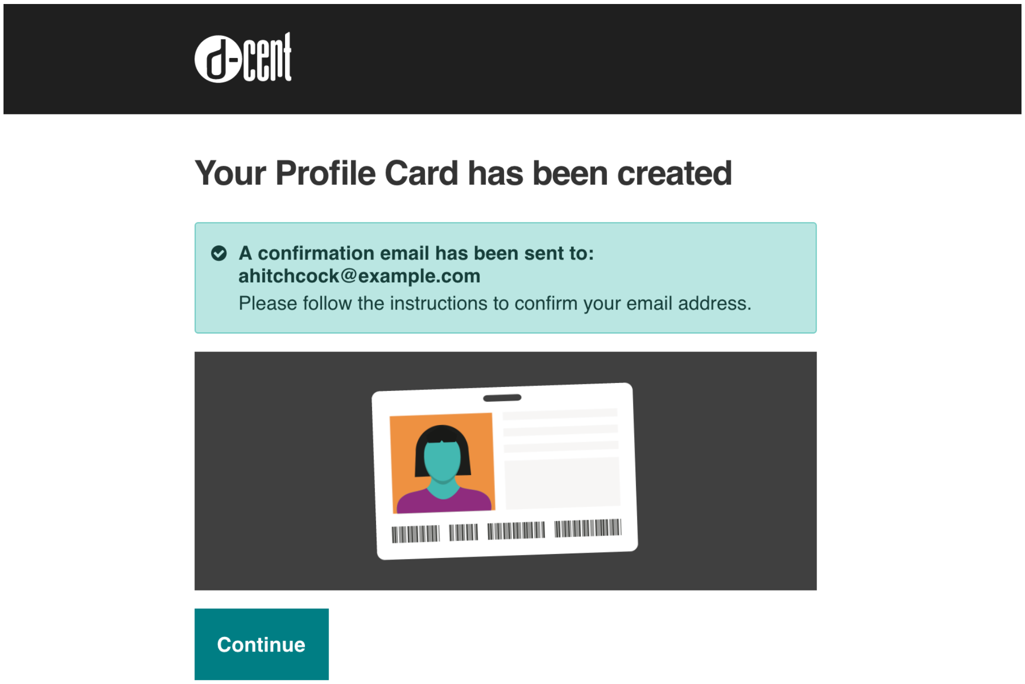
Welcome to D-CENT

Connect with all your D-CENT tools. Sign in securely to applications using your D-CENT Profile Card. You have full control of the information you'd like to share.

Create a Profile

Register and sign in to supported applications using your D-CENT Profile Card.

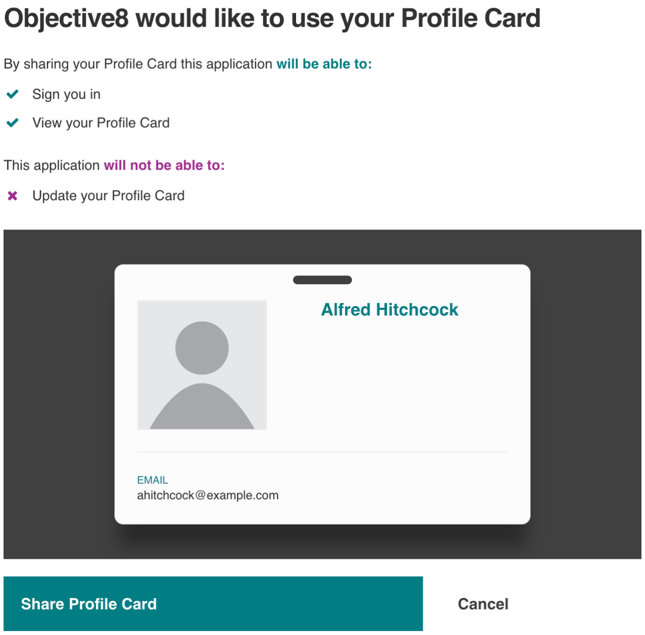
If they create a new profile then they are shown a confirmation page.



Your Profile Card has been created

A confirmation email has been sent to: ahitchcock@example.com
Please follow the instructions to confirm your email address.

After they have signed in, or if they were already signed in before they began, they are then directed to this page, prompting them to share their details with Objective8.



Objective8 would like to use your Profile Card

By sharing your Profile Card this application **will be able to:**

- ✓ Sign you in
- ✓ View your Profile Card

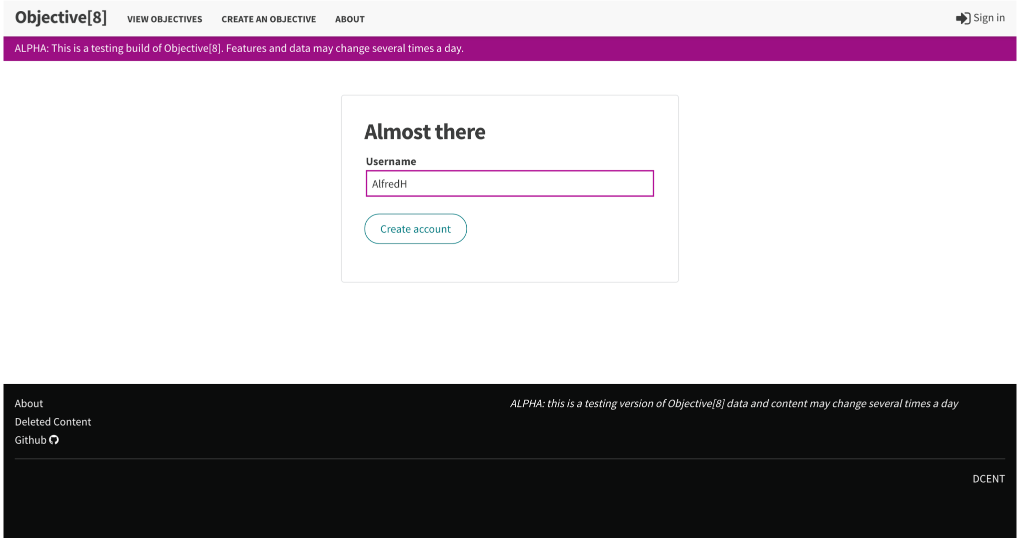
This application **will not be able to:**

- ✖ Update your Profile Card

Alfred Hitchcock

EMAIL
ahitchcock@example.com

If they decline they are provided with a link to Objective8, otherwise they are then forwarded back to Objective8 and asked to choose a username if this is their first time connecting to Objective8.


Objective[8] VIEW OBJECTIVES CREATE AN OBJECTIVE ABOUT Sign in

ALPHA: This is a testing build of Objective[8]. Features and data may change several times a day.

Almost there

Username

AlfredH

Create account

About

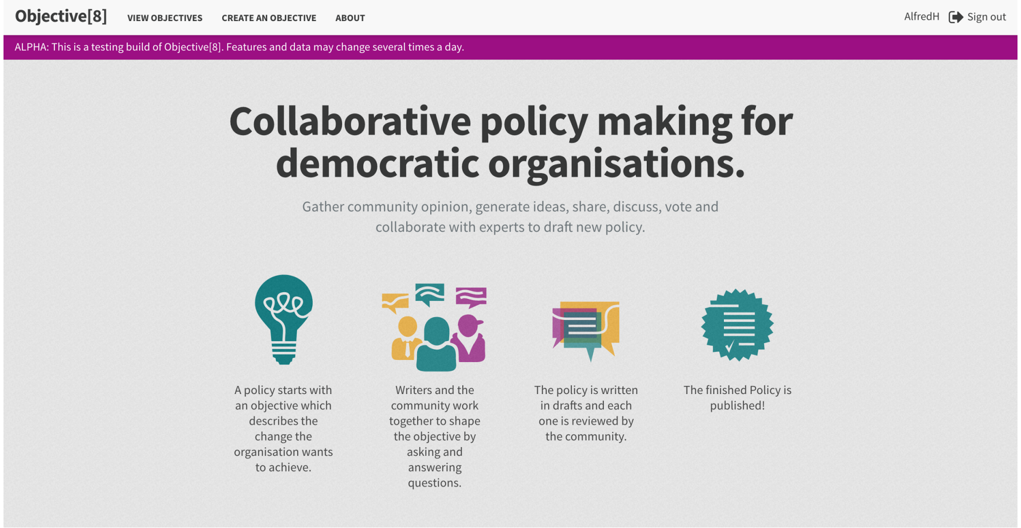
Deleted Content

Github

ALPHA: this is a testing version of Objective[8] data and content may change several times a day

DCENT

After choosing their username, they are then returned to wherever they were when they began signing in.


Objective[8] VIEW OBJECTIVES CREATE AN OBJECTIVE ABOUT AlfredH Sign out

ALPHA: This is a testing build of Objective[8]. Features and data may change several times a day.

Collaborative policy making for democratic organisations.

Gather community opinion, generate ideas, share, discuss, vote and collaborate with experts to draft new policy.

A policy starts with an objective which describes the change the organisation wants to achieve.

Writers and the community work together to shape the objective by asking and answering questions.

The policy is written in drafts and each one is reviewed by the community.

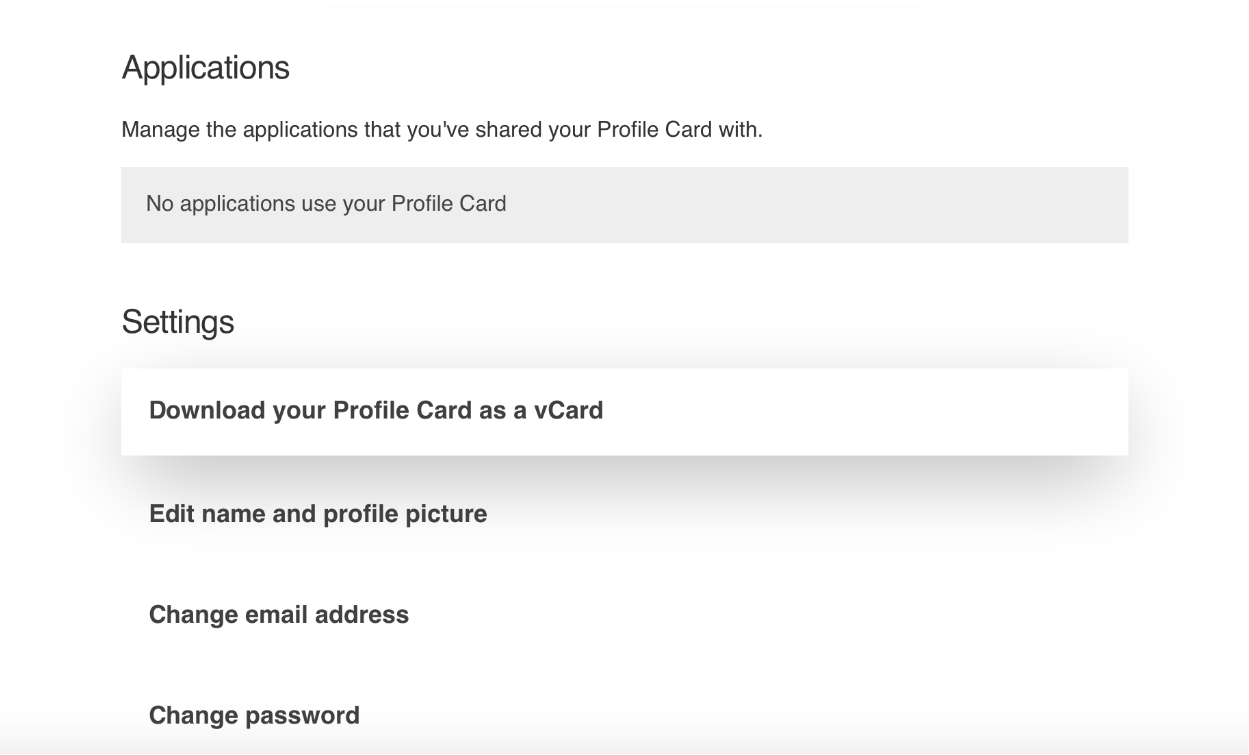
The finished Policy is published!

5.5 VCards

VCard is a standard file format for electronic business cards. It is used in the Single Sign On app, Stonecutter. Users can download their details in the format of a VCard, providing users with a .vcf file that can then be shared with contacts online. VCard is used by Gmail and Apple contacts for importing contacts.

Example:

```
BEGIN:VCARD
VERSION:3.0
N:Smith;Jane;;;
FN:Jane Smith
PHOTO;TYPE=JPEG;ENCODING=<encoded photo>
EMAIL:janesmith@example.com
REV:20160408T154654Z
END:VCARD
```



Applications

Manage the applications that you've shared your Profile Card with.

No applications use your Profile Card

Settings

Download your Profile Card as a vCard
Edit name and profile picture
Change email address
Change password

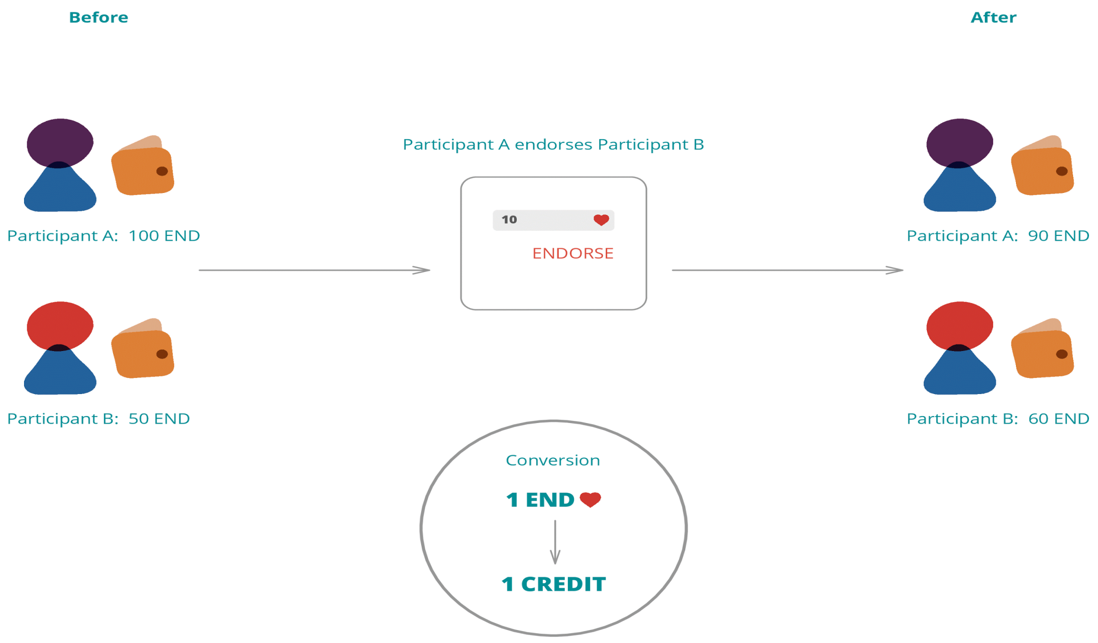
6 Freecoin implementation

Freecoin interoperates and links other software made in D-CENT (Stonecutter and Mooncake) via open protocols: OpenID Connect and Activity Streams.

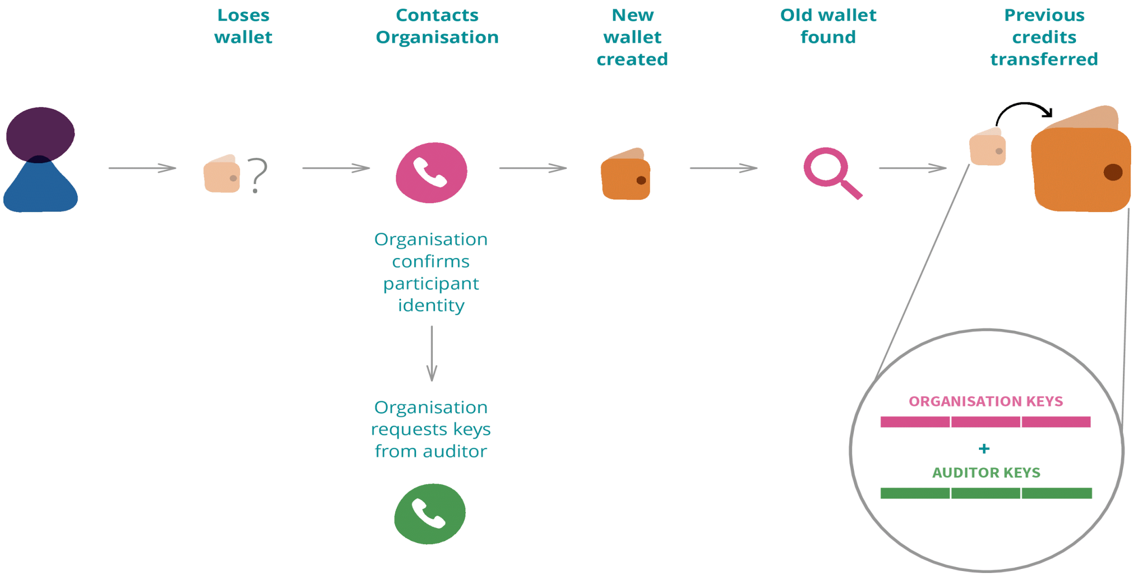
Freecoin presents its participants with a starting wallet and basic functionalities to send and claim value, as well list transactions. Most features can be re-programmed ad-hoc to match criteria established by the Social Proof of Work, for instance approved decisions in Objective8.

All Social Proofs of Work can be a variation of a similar, or at least compatible, Smart Contract, i.e. a piece of software that can encode actions to perform in order to execute a transaction and broadcast is on a distributed ledger. In this case, a transaction can be a virtual currency transfer (crypto-coin broadcasting), the exchange of tokens for voting purposes, or more in general the transparent and accessible storage of decisions made in a datum context be it social, political or economic/financial.

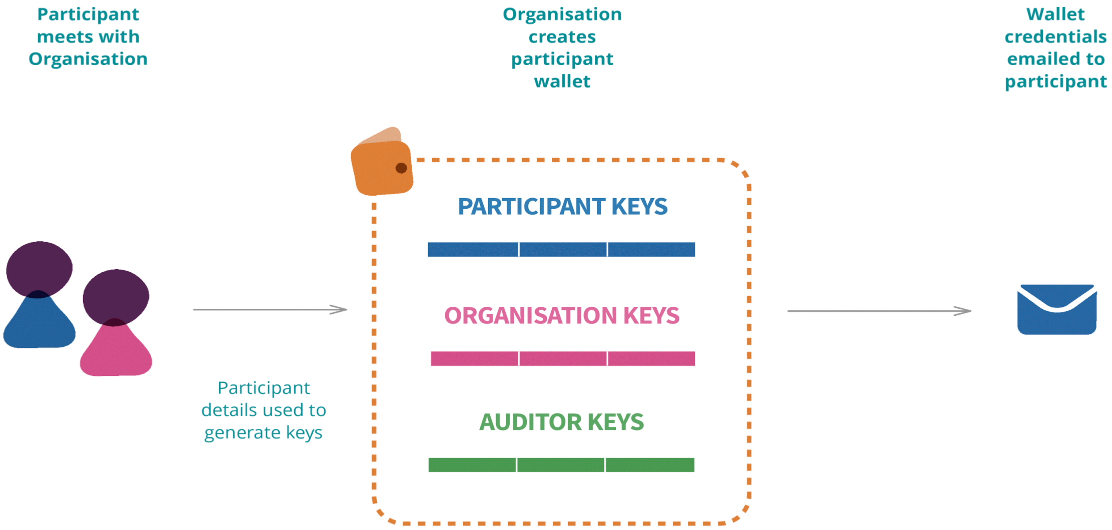
By expanding the possibilities of a Social Proof of Work, Freecoin is designed to balance private transparency and distributed ledger architectures.



Complete transparency on a community scale, as a private ledger, can validate certain transactions also on global distributed ledgers backends, or peer to peer networks running smart contracts.


Loses wallet
Contacts Organisation
New wallet created
Old wallet found
Previous credits transferred
?
Organisation confirms participant identity
Organisation requests keys from auditor
ORGANISATION KEYS
+
AUDITOR KEYS

The resilience and possibility to retrieve lost access on blockchain accounts in Freecoin is implemented as the FXC protocol, a proposal to scale the security of Shamir's secret sharing (SSS, key splitting technique) and standardize its parsing and rendering into portable ASCII encoded SSS entities.


Participant meets with Organisation
Organisation creates participant wallet
Wallet credentials emailed to participant
PARTICIPANT KEYS
ORGANISATION KEYS
AUDITOR KEYS
Participant details used to generate keys

Freecoin is written in Clojure, a dialect of LISP, and is designed to run on any JVM (including OpenJVM) as a website. Its security model is minimalism, it does not require any Javascript to navigate and all operations and interactions in Freecoin have test coverage. Freecoin is a readable implementation of a model-view-controller (MVC) architecture.

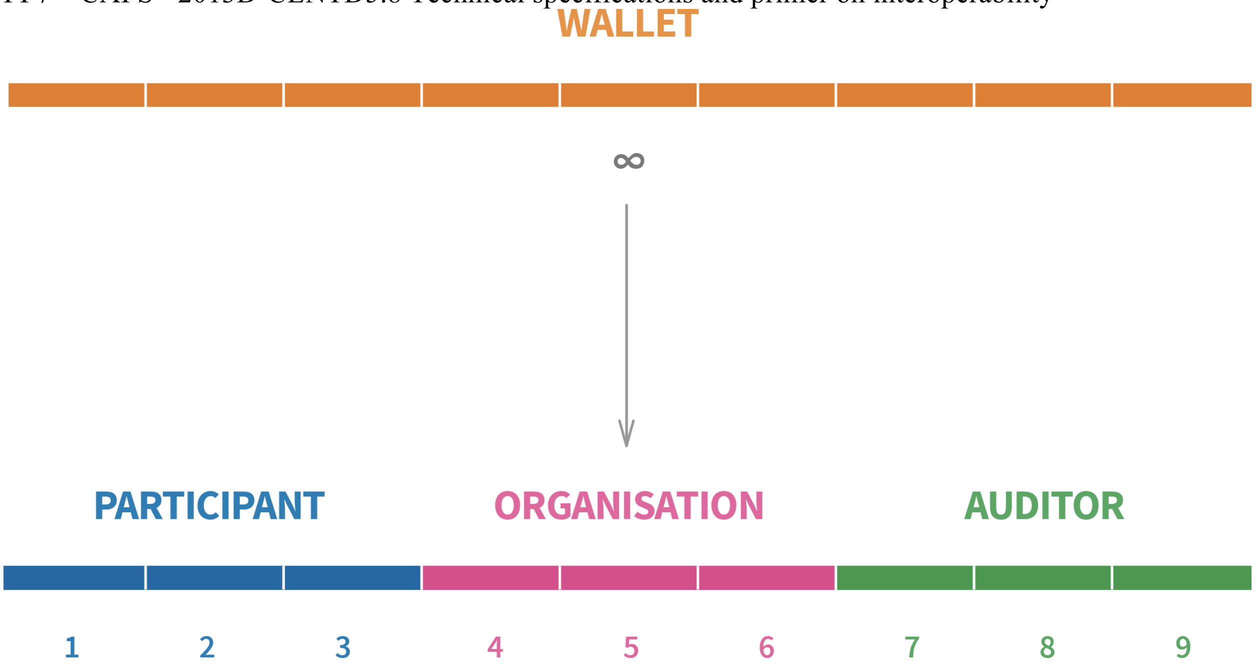
The blockchain interface API is a minimum common denominator to most existing blockchain implementations.

```
defprotocol Blockchain
 ;; account
  (import-account [account-id secret])
  (create-account [account-id])

  (get-address [account-id])
  (get-balance [account-id])

  ;; transactions
  (list-transactions [account-id])
  (get-transaction   [account-id txid])
  (make-transaction  [account-id amount recipient secret])

  ;; vouchers
  (create-voucher [account-id amount expiration secret])
  (redeem-voucher [account-id voucher])
  )
```

6.1 FXC (proposed standard)

The key to access an account on a blockchain platform is an extremely delicate secret to be kept by users and, once lost, cannot be retrieved. This represents one of the biggest challenges for the adoption and usability of blockchain solutions. Software wallet implementations so far have proposed different solutions to this same problem, with variable levels of success and reliability. The production of secret sequences that cannot be guessed (as shown by directory.io for the Bitcoin blockchain) is the main focus, while it is hardly possible to find decentralized implementations that can retrieve lost accesses.

The FXC protocol aims at marshalling fairly large secret sequences of numbers in a format that humans can easily note down on paper and communicate in voice.



Here below an example of wallet key generation in FXC shows how keys are basically strings with a prefix, an infix and a suffix containing informations on the version of the protocol, the name of the blockchain and the positioning of the slice, since the Shamir's secret sharing algorithm requires all slices to be placed in the same order as they were created.

The secret: "FXC1_NXT_77686450_FXC_79484894_0"

is split here below in 9 different parts, 5 of which are enough to retrieve it

```
 ["FXC1_NXT_218426873_FXC_230743203_1"
  "FXC1_NXT_699558074_FXC_1495463556_2"
  "FXC1_NXT_1829619775_FXC_6695186729_3"
  "FXC1_NXT_3919736402_FXC_20518480314_4"
  "FXC1_NXT_7283617085_FXC_49520938719_5"
  "FXC1_NXT_12237555658_FXC_102125183168_6"
  "FXC1_NXT_19100430659_FXC_188620861701_7"
  "FXC1_NXT_28193705330_FXC_321164649174_8"
  "FXC1_NXT_39841427617_FXC_513780247259_9"]
```

The columns of secret numbers are two, but can be more if desired, raising the cryptographic security of the system. While the limit for a SSS fast implementation is a long integer, multiplying the transformations can well be scalable.

Each slice is an underscore separates sequence of strings so composed:

- FXC1 is the name of the protocol and its version number
- NXT is the identifier of the blockchain used
- Number sequence
- FXC is the string separating each column of number sequences
- Number sequence

If a common format like the proposed FXC can be adopted in the future by wallet implementations, then there can be compatibility to port secret data across them.

Freecoin SSS implementation is based on code written in Java by Tim Tiemens, the project roadmap includes a verifiable implementation of the same algorithm and parser in C.

Shamir's Secret Sharing algorithm was invented by Adi Shamir. References:

- Shamir, Adi (1979), "How to share a secret", Communications of the ACM 22 (11): 612–613
- Knuth, D. E. (1997), The Art of Computer Programming, II: Seminumerical Algorithms: 505

7 Wider interoperability

Activity Streams

- Supported by a growing user base. Most up to date list can be found here: Activity Streams implementations

OpenID Connect

- Supported as a standard by Twitter and Google, and is currently under development to be supported by Okta. Facebook’s graph API is similar to OpenID Connect. Objective8 supports login with Facebook and Twitter, in addition to Stonecutter.

VCards

- Supported by Gmail, Outlook, Apple Contacts and other services for transferring personal details and backing up contacts. They’re capable of producing contacts in the .vcf format to be easily shared or sent by email, and adding contacts when given files of the same format. They are also used when producing backup copies of address books or contact lists in Apple products.

Web Components

- User interface components can be reused between different applications. You could for example use a component that represent an idea from citizen, originally submitted through Your Priorities, in Open Active Voting to use it in participatory budgeting context. Out of the Your Priorities web components you can easily create new domain problem specific apps that use only certain components. It’s a bit like assembling a new app using lego blocks.

7.1 Challenges in implementing open standards for D-CENT tools

The purpose, and advantage, of common standards for software is the ease of integration between multiple tools that were developed in isolation from each other. However, in practice these standards tend to be implemented slightly differently by different teams. This can provide additional complications when integrating multiple tools that need to be kept in mind.

In addition, working with standards that are still under development can necessitate last minute changes or extensive refactorings.

As part of the D-CENT work, since Activity Streams were not yet finalised - there were issues of redoing work. However, by the end of the D-CENT project (and before the end of 2016), Activity Streams 2.0 should be a finalised standard whose core will no longer change.

Sometimes there are competing standards. For example, with the Helsinki pilot, when combining different sources, there were a lot of different standards coming into play. When using municipality data there are national document metadata standards (SÄHKE2), previous technology standards (RSS), civic tech standards (Popolo), standards that come with AS2, (JSON-LD, GeoJSON) and the schemas the data producing organisations have been using previously (OpenAHJO). This was solved with a combination of small applications based on PostgreSQL and Python libraries that are handled as connectors, each refactoring and publishing the data as separate unit, based on each connections' user story. Data is then published as one harmonised, versioned data stream to be consumed by other nodes.

Within the Social Web Working Group, there were at times disagreement about the method by which to solve problems and which technical languages to use. Due to this, the group decided to allow for a multiplicity of implementations even if they overlapped. Their relationships and the different use cases they address are documented in the Social Web Protocols note.

8 Future

Standards and tools are only as effective as their level of adoption. The more that existing standards and tools are used, the more mature the user experience and functionality can be. It's important to overcome challenges and focus on the important benefits of decentralisation, mature open source tools and standardisation.

With the interoperable D-CENT tools and standardisation work we have carried out as part of this project we hope to have created a platform for citizen engagement communities and their tools. Sustainability strategies for the D-CENT ecosystem are further discussed in D1.3.

The work of the social web working group will continue until the end of 2016, by which point we expect Activity Streams 2.0 will become a full W3C Recommendation. Other related standards have also been developed and at least one or two look set to become official standards. More details and all the links can be found here: https://www.w3.org/Social/WG