Project no. 610349

D-CENT

Decentralised Citizens ENgagement Technologies

Specific Targeted Research Project
Collective Awareness Platforms

D6.5 Charting "open specifications" for standardisation activities on federated social networking

Version Number: 1
Lead beneficiary: W3C/ERCIM
Due Date: 31 May 2015
Author(s): Harry Halpin
Editors and reviewers: Gareth Rogers, Jaakko Korhonen

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: 29 May 2015

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

Contents

1. Executive Summary

The overall objective of this deliverable (D6.5) given by Work Package 6 is to present a unified suite of standards for the development of a decentralized (federated) social web. This document discusses the standardization efforts funded as part of the D-CENT project so far and presents their current state of play. The standardization efforts of D-CENT are happening at the World Wide Web Consortium, as managed by their European Headquarters in France (W3C/ERCIM). Harry Halpin (W3C) has been leading this effort, under a half-time contract with ERCIM. The World Wide Web Consortium (W3C), founded by the inventor of the Web, Tim Berners-Lee, is the world's leading standards body for Web standards. After the inception of the D-CENT project, the W3C started a new standardization effort, the Social Activity, in order to bring the social standards into reality.

In particular we will describe: (i) **the structure of the Social Activity** at the W3C, including both the charters that include deliverables of the Social Web Working Group and the broader Social Interest Group, as well as its relationship to other standards bodies like the IETF and the composition of each group (ii) the in-progress **federated social networking standards,** that will be implemented in the D-CENT platform, including ActivityStreams 2.0 and the Activity Vocabulary, which share their own data in a decentralized manner, and (iii) **next steps and open problems for decentralization** which includes the plan for how other partners of D-CENT (Iceland, Finland, Spain, and the primary implementer Thoughtworks) will implement decentralized social networking by the end of the project (D5.8) as well as a brief overview of open research problems in security and privacy that are necessary in order to achieve a decentralized architecture beyond the lifetime of the D-CENT project.

The output of this deliverable will be described for the most part by the charters and draft standards produced by the Social Activity at the W3C, which represent not only the work of W3C but the collective work of the Working Group and Interest Group, which is composed of W3C members. Indeed, the primary editors of the document are not W3C employees but volunteers from member organizations such as IBM and invited experts from open-source (such as the IndieWeb, Linked Data, and pump.io communities). However, the collective intelligence of the entire group has authored the document via e-mail discussions, github requests, and implementation experience. The D-CENT project participates in the standards process primarily via the work of ERCIM, as no other D-CENT members attend the regular meetings of the Working Group (although Forum Virium has joined the Working Group, and Thoughtworks has been invited but is still processing the royalty-free patent agreement). While these standards are currently "in development" and may change by the end of the D-CENT project and even afterwards. The important aspect is that *for the first time at a major standards body* there has been an explicit effort to develop the technical infrastructure for a decentralized social web, and that this effort has produced a *suite of standards* by bringing together an international community of interested companies (both SMEs and major enterprises), academic researchers, and open-source hackers.

2. W3C Social Activity

As a global standards body, the W3C has a well-defined process for creating new standards. In particular, it starts both Working Groups for new standards as well as Interest Groups to gather interest/use-cases in new topics. Working Groups and Interest Groups are attached to what are called *Activities* at the W3C, which are in turn attached to higher-level organizational entities called *Domains*. Each Activity is given an *Activity Statement* defining the scope of the activity, and then what Interest Groups and Working Groups fall beneath it.

The primary difference is that a *Working Group* typically is highly technical and tightly focused on a finite number of deliverables in terms of standards, while *an Interest Group* may often be open ended discussion with a number of non-standard deliverables. Standards, called W3C Recommendations by the W3C (but comparable roughly to ISO standards, IETF RFCs, and other forms of well-known standards, albeit specialized for the Web), are distinguished from non-standards by the fact that they guide the implementation of code and must demonstrate interoperability as well as maintain patent commitments. Legally in terms of patent commitments, the W3C is committed to a royalty-free licensing of all standards, which both protects the rights of open-source developers and SMEs who wish to use W3C standards, but also requires that each individual or organization license their relevant patents to the W3C on a royalty-free basis. Note that simply using open source licensing, even strict licensing such as the GPL, does not protect one from software patents, which the W3C Royalty-Free licensing does. Unlike in some more lightweight agreements such as the Open Web Foundation agreement, the W3C also applies licensing to the entire organization. For this reason, the W3C is a member-organization, and non-members must sign an Invited Expert agreement in order to join the relevant Working Group or Interest Group. The W3C leaves the number of Invited Experts to the discretion of the chairs.

In terms of D-CENT, the W3C started a new Activity, called the *W3C Social Activity* (as part of the Technology and Society Domain), in order to create the standards necessary for an open and federated social web. In this Activity were grouped two groups, a *Social Interest Group* and a *Social Web Working Group*. The suite of open and royalty-free standards is to be created by the Social Web Working Group, and so from D-CENT technically inclined participants such as OKFN and Thoughtworks have been invited to join as joining the Working Group as an Invited Expert requires proof of implementation experience in the federated social web space or a commitment to implement in a product or open-source platform (such as D-CENT). However, all members of D-CENT have been invited to join the Social Interest Group, which maintains the strategic membership direction.

In this section, we include Activity Statement for the W3C Social Web, and then the charters of the W3C Social Web Working Group (also given in D4.2, but given here for sake of completeness, with a revised schedule) and the W3C Social Interest Group charter. These charters have been edited to remove redundancy, online specific links, and process-oriented text. Note that all charters have been approved by the W3C membership, although large social networking sites (Facebook, Twitter, and Google) declined to participate. Yet the agreement to participate of large enterprise members such as IBM, Siemens, and SAP along with open-source hackers such as the IndieWeb movement and Tim

Berners-Lee's CrossCloud work demonstrates that the work has wide applicably, and large enterprise has committed their sizable relevant patents to royalty-free licensing, making this work legally much more safe to be implemented.

2.1 Structure of the W3C Social Activity

Note that this version is the version of May 31 2015 for D-CENT. The latest version is available here: http://www.w3.org/Social/

The W3C Social Activity (slogan: *Standardizing the Social Web*) is related to the Data Activity and ongoing Security Activity.

For definitions of terms such as "social" and "activity", please see the W3C Social XG report A Standards-based, Open and Privacy-aware Social Web.

The focus of the Social Activity 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 Socal Web applications. These technologies are crucial for both federated social networking and the success of social business between and within the enterprise.

The Social Web Working Group is committed to the technical standardization work of the following deliverables:

- An extensible transfer syntax for activities like status updates, with the ActivityStreams 2.0 data-format being an input.
- An API for third-party social content embedding, with the OpenSocial 2.5.1 Activity Streams and Embedded Experiences APIs Member Submission being an input.
- Possibly a protocol for federation, with the Web Mention protocol and the Linked Data Platform being possible inputs.

The Social Interest Group focuses on messaging and co-ordination in the larger space. This work will include a use-case document, including "social business" enterprise use-cases. Their work may include the following Interest Group Note deliverables:

- An unified architecture document.
- Creation and possibly maintenance of vocabularies for data related to social not already adequately covered by existing work.

A significant number of existing W3C members from industry have expressed interest in this work, ranging from large enterprises to start-ups as well as non-profit and academic institutions. Interest includes existing Community Groups such as the Federated Social Web Community Group and Social Business Community Group and these groups may continue to provide important input to the new Working Group and Interest Group in the Social Activity.

The Social Activity continues liaison work with the ActivityStream community, the OpenSocial Foundation, IETF, _ and IndieWeb community, and well as possibly OpenID Foundation and other communities.

2.2 Context & Vision

Interoperability around social should be standardized in order to allow communication between heterogeneous Web applications that feature explicitly social features such as status updates and user profiles. Currently, APIs and protocols in this space do not allow easy transfer of social data between existing systems, as is required by many "social business" systems for both business-to-business and business-to-customer relationships. Second, the lack of a standard API prevents Web application developers from embedding social functionality from third-party sites into their Web applications easily. Lastly, many users and organizations wish to have autonomous control over their own social data while sharing it in a decentralized manner, which requires a Web-based protocol for federation.

The Social Activity has been a goal of many members of W3C. 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 that produced Towards a Standards-based, Open, and Privacy-Aware Social Web. Outcomes of this report included the more open 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 like ActivityStreams and PortableContacts. This also led to further outreach, with the W3C sponsoring and helping organize the grass-roots Federated Social Web conference. However, at the time there was still not critical mass of W3C members interested in social. The W3C "Headlights" Social Taskforce was then started to discuss the topic with members and create a "block-diagram" of the space.

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 and the 2012 CTO Guide to Social Business. The Social Standards: The Future of Business workshop (August 7-8th, sponsored by IBM and the Open Mobile Alliance) developed the standards and ideas in this activity statement, see the Final Report of the workshop for more details. 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.

This work is funded in part by the European Commission through the D-CENT Project, which creates privacy-aware tools and applications for direct democracy and economic empowerment.

2.3 Social Interest Group Charter

Harry Halpin, <hhalpin@w3.org>, Social Activity Lead

Note that the current version of the charter for the Social Interest Group is available here: http://www.w3.org/2013/socialweb/social-ig-charter.

The mission of the Social Interest Group, part of the Social Activity, is to co-ordinate messaging around social at the W3C and to formulate a broad strategy to enable social business and federation.

End date	31 December 2016
Confidentiality	Proceedings are Public
Co-chairs	Mark Crawford (SAP)

Team Contacts	Harry Halpin (FTE %: 10)
Usual Meeting Schedule	Teleconferences: Bi-weekly, with additional topic-specific calls may be held on as needed basis. Face-to-face: Once a year at minimum, three times a year maximum. The Interest Group may meet during the W3C's annual Technical Plenary week; other additional F2F meetings may be scheduled as needed.

2.3.1. Goals

The Social Interest Group is a forum that can formulate and steer standardization in social across the W3C (both in Working Groups and Community Groups) and work in co-ordination with other standards bodies, as well as community-led efforts. For definitions of terms such as "social" and "activity", please see the W3C Social XG report A Standards-based, Open and Privacy-aware Social Web.

2.3.2. Scope

The interest group should harvest use-cases that produce concrete actions for social standards and review standards in light of those use-cases. In particular, there will be a focus on the discussion of requirements of social technologies that enable both a federated social web and social business. Federation includes the creation of distributed and decentralized social software. Social business is the use of such federated social software to help the development and delivery of competitive products and services. It is intended that the forum not only focus on the use of social technologies within the enterprise and other organizations, but also identify interoperability with user-facing social software.

The forum is intended to include businesses, users, designers, developers, equipment manufacturers, social platform vendors, browser vendors, network operators, advertising, and other relevant participants in the value chains that require social software.

2.3.3. Deliverables

Mandatory Deliverables

The group will deliver the following to fulfill its goals as an Interest Group Note, subject to discussion in the Interest Group:

- Use-case and Requirements Report
  This report will document the use cases to drive social standards for both businesses and consumers. These use-cases can include personalization based on context, including location, activities, and connected devices.

Possible Deliverables

The group may propose additional standards work to the W3C, and may publish Interest Group Notes as needed. Examples of possible Interest Group Notes are:

- *Social Architecture Report*: This document takes a broad look at all the social technologies and specifications under development, including summarizing their contribution and tracking their adoption. The document will then demonstrate how a subset of these can be used as a unified architecture. An overview report has been produced by the Social Web XG in 2010.

- *Social Vocabularies*: Various standards such as ActivityStreams and RDF allow various items of shared interest, such as products and actions ("likes"), to be named with a URI for reasons of interoperability. Vocabularies are sets of these related URIs around particular activities (business processes, sharing, shopping). The Interest Group may maintain a list of shared URIs relevant for the use of social standards.

The production of any Interest Group deliverables depends upon the resources available, and will change as new information, adoption, and implementation experience is reported to the group. As the deliverables are in general non-normative, they can be changed and maintained throughout the lifetime of the Interest Group to reflect the current landscape of social technologies. The group should work with other groups of experts to make sure issues of security, privacy, accessibility, and data protection are taken into consideration.

2.3.4. Dependencies and Liaisons

Dependencies

Social Web Working Group

To co-ordinate use-cases and vocabularies for the technical standardization work on social data, formats, and APIs, as well as on overall architecture.

HTML Working Group

To co-ordinate with any use-cases that may need to be addressed by HTML.

WebAppSec Working Group and Web Security Interest Group

To co-ordinate with any security requirements for use-cases or the social architecture.

Privacy Interest Group

To co-ordinate with any privacy and data protection requirements on the social web.

Protocols and Formats Working Group

To co-ordinate with any accessibility requirements for use-cases or the social architecture.

Web and Mobile Interest Group

Having the social web be easily usable by mobile is a priority for the Social Interest Group.

Federated Social Web Community Group

This work of this Community Group on the Federated Social Web provides a community-driven implementation experience and use-cases.

Social Business Community Group

This work of this Community Group provides social business experience and use-cases.

External Groups

The following is a tentative list of external bodies the Working Group should coordinate with: ActivityStreams CommunityThe work of this grass-roots community effort created Activity Streams, an important social data format of the open social web.Cloud Standards Customer Council

The Cloud Standards Customer Council is an end user advocacy group dedicated to accelerating cloud's successful adoption, and drilling down into the standards, security and interoperability issues surrounding the transition to the cloud.

IndieWebCamp Community

The IndieWebCamp grass-roots community has deployed and helped to develop a number of protocols and formats for federated social web related use-cases such as IndieAuth,

Webmention, h-entry, h-card, h-cite, as well as design conventions for replies, web actions and other cross-site social user interfaces.

Internet Engineering Task Force

The IETF is responsible for defining robust and secure protocols for Internet functionality, and their work on the protocol layer is exceedingly important for social, as is their input on security aspects of the overall social architecture.

Open Mobile Alliance

OMA was formed by mobile operators, device and network suppliers, information technology companies and content and service providers to deliver open specifications for creating interoperable services like social networking on any bearer network.

OpenSocial Foundation

The OpenSocial API is an open API for the social web, and is particularly important for social business.

XMPP Foundation

The XMPP Foundation has done relevant work in the area of federation using XMPP for social.

2.3.5. Communication

This group conducts its substantive work exclusively on the public mailing list public-social-interest@w3.org (archive) for Interest Group discussion and encourages non-member contributions on that list. Administrative tasks may be conducted Member-only communications. Task forces may use separate public mailing lists to conduct their work.

Information about the group (deliverables, participants, face-to-face meetings, teleconferences, etc.) is available from the Social Interest Group home page.

Harry Halpin, <hhalpin@w3.org>, Team Contact
Mark Crawford (SAP), Chair

2.4. Social Web Working Group Charter

The latest version of the charter is available at this link: http://www.w3.org/2013/socialweb/social-wg-charter. This is a draft revision that takes into account the current progress of the Working Group, in particular new timings as regards the delivery of the Social API and possible Federation deliverables.

The mission of the Social Web Working Group, part of the Social Activity, is to define the techical protocols, vocabularies, and APIs to facilitate access to social functionality as part of the Open Web Platform. These technologies should allow communication between indepedent systems, federation (also called "decentralization") being part of the design.

Join the Social Web Working Group.

End date 31 December 2017

Confidentiality Proceedings are Public

	Tantek Çelik (Mozilla)
Co-chairs	Evan Prodromou (E14N) Arnaud Le Hors (IBM)
Team Contacts	Harry Halpin (FTE %: 20)
Usual Meeting Schedule	Teleconferences: Weekly, although the chair may call for topic-specific calls in addition when needed and may change working mode as work progresses. Face-to-face: Once a year at minimum, three times a year maximum. The Working Group will meet during the W3C's annual Technical Plenary week; other additional F2F meetings may be scheduled as needed.

2.4.1. Goals

The Social Web Working Group will create **Recommendation Track** deliverables that standardize a common JSON-based 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. For definitions of terms such as "social" and "activity", please see the W3C Social XG report A Standards-based, Open and Privacy-aware Social Web.

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 travel request may need to redirect a user to the company's travel agent. Rather than re-direct the user, this interaction could be securely embedded within page itself.
- *Enterprise Social Business:* In any enterprise, different systems need to communicate with each other about the status of various well-defined business 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 an enterprise for crucial business processes.

2.4.2. Scope

The 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:

A transfer syntax for social data such as activities (such as status updates) should include at least the ability to describe the data using URIs in an extensible manner, time-stamping, and should include a serialization compatible with Javascript (JSON) and possibly JSON-LD. Formats based on XML or other data serializations are out-of-scope.

A social API should include the ability to embed third-party information and share social data between web applications. The API should re-use the social data transfer syntax and may allow some interaction with the federation protocol. The API should also be extensible in terms of the items of interest expressible by the data format.

A Web protocol for federating social data should include at least the ability to share status updates using the JSON-based syntax developed by the Working Group. This protocol may allow the capture of new data, the verification of data using techniques such as as digital signatures, and the use of groups with some form of access control or capabilities.

Other components necessary for building federated/decentralized social Web systems are in scope but will not lead to Recommendation-track work without re-chartering, and should be discussed in the Social Interest Group.

Success Criteria

In order to advance to Proposed Recommendation, the specification is expected to have two independent implementations of each feature defined in the specification.

2.4.3. Deliverables

Recommendation-Track Deliverables

The working group will deliver the following to fulfill its goals, subject to discussion in the Working Group:

- **Social Data Syntax**
  A JSON-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.
- **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.
- **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. Note that the Working Group may not develop the Federation Protocol into a Recommendation and it may become a Working Group Note.

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.

Other Deliverables

Other non-normative documents may be delivered as:

- A use case document defining how features in each specification relate to concrete use-case. This document may also be maintained by another group such as an Interest Group. The preferred method of adding features to specifications should be explaining how they fulfill a particular use-case that the Working Group has explicitly agreed to address.
- Test suite for Social Web API (and possibly federation if the Working Group sends the Federation Protocol to the Recommendation-track.);
- Primer or Best Practice documents to support Web developers when designing Social Web applications.

Milestones

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.

The milestones table below has the stages of W3C process: First Public Working Draft (FPWD), Last Call (LC), Candidate Recommendation (CR), Proposed Recommendation (PR), and Recommendation (Rec), with the schedule on a quarterly basis.

Milestones

Note: The group will document significant changes from this initial schedule on the group home page.

Specification	FPWD	LC	CR	PR	Rec
Social Data Syntax	Q3 2014	Q3 2015	Q4 2015	Q2 2016	Q3 2016
Social API	Q3 2015	Q3 2016	Q4 2016	Q2 2017	Q3 2017
Federation Protocol	Q4 2015	Q4 2015	Q1 2016	Q3 2017	Q4 2017

2.4.4. Dependencies and Liaisons

Dependencies

Social Web Interest Group

To co-ordinate around use-cases, architecture, and vocabularies for social as well as general messaging.

HTML Working Group

To co-ordinate with any features under development or that may need to be added to HTML.

Web Applications Working Group

To co-ordinate with APIs and features of the Social Web with other existing and planned Web application APIs, in particular work with Web Components that may be used by the Social API.

WebAppSec Working Group and Web Security Interest Group

To co-ordinate with any security requirements and specifications arising from the security needs of socially-enabled Web applications. In particular, the Social API may wish to build on top of the WebAppSec Working Group's chartered work around secure cross-domain framing, secure mixed content, and lightweight isolated / safe content.

Linked Data Platform Working Group

The Linked Data Platform Working Group is producing specifications for sharing Linked Data that could be useful to the federation protocol.

Privacy Interest Group

All specifications produced by the Working Group will have a review to help addresses privacy concerns.

Protocols and Formats Working Group

All specifications produced by the Working Group will have a review to help addresses accessibility concerns.

Web and Mobile Interest Group

Having the social web be "mobile first" and so easily usable by mobile is a priority.

Federated Social Web Community Group

This work of this Community Group on the Federated Social Web provides a community-driven implementation experience and use-cases.

Social Business Community Group

This work of this Community Group provides social business experience and use-cases.

Furthermore, the Social Web Working Group expects to follow the following W3C Recommendations, Guidelines and Notes and, if necessary, to liaise with the communities behind the following documents:

- Architecture of the World Wide Web, Volume I
- Internationalization Technical Reports and Notes
- QA Framework: Specification Guidelines

External Groups

The following is a tentative list of external bodies the Working Group should coordinate with:

IndieWebCamp Community

The IndieWebCamp grass-roots community has deployed and helped to develop a number of protocols and formats for federated social web related use-cases such as IndieAuth, Webmention, h-entry, h-card, h-cite, as well as design conventions for replies, web actions and other cross-site social user interfaces.

Internet Engineering Task Force

The IETF is responsible for defining robust and secure protocols for Internet functionality. A clear relationship with IETF is vital to ensure the security and success of elements of the Social Web that supervenes upon protocol-level work. Security reviews should involve participants from the IETF Security Area.

ActivityStreams Community

The work of this grass-roots community effort created ActivityStreams, a key part of the open social web. The Working Group will continue to communicate with the community.

Open Mobile Alliance

OMA was formed by mobile operators, device and network suppliers, information technology companies and content and service providers to deliver open specifications for creating interoperable services like social networking on any bearer network.

OpenSocial Foundation

This work of this Foundation will provide input for the API in the form of OpenSocial 2.5. There will also be active liasing with this Foundation over the course of the Working Group, as well as co-ordination over outreach and strategy.

2.4.5. Communication

Most Social Web Working Group Teleconferences will be conducted on an as-needed basis. Normally, at least one teleconference will be held per week.

Most of the technical work of the group will be done through discussions on one of the group's public mailing list and a list for public comments allows posts by anyone:

- public-socialweb@w3.org (archive) for Working Group discussion
- public-socialweb-comments@w3.org (archive) for general discussion and commentary

The group can use a Member-confidential mailing list for administrative purposes and, at the discretion of the Chairs and members of the group, for member-only discussions in special cases when a particular member requests such a discussion.

Information about the group (for example, details about deliverables, issues, actions, status, and participants) will be available from the Social Web Working Group home page.

Harry Halpin, <hhalpin@w3.org>, Team Contact
Tantek Çelik (Mozilla), Evan Prodromou (E14N), and Arnaud Le Hors (IBM), Chairs

3. W3C Social Web Working Group Specifications

As detailed in the charters, the Social Web Working Group is tasked to produce a number of open specifications that will enable federated social networking, and so allow the D-CENT platform to interoperate with other systems as well as to function itself in a decentralized manner. Although this will not be implemented and tested with the D-CENT pilot projects until later in Work Package 5, the drafting of the specifications is already underway. The Social Web Working Group has come to consensus on a number of use-cases of basic social networking functionality and these specifications are meant to satisfy them. The use-cases are included in this deliverable. We do recognize that these use-cases do not include voting, consensus, any alternative or crypto-currency usage, and the like. However, by creating the extensible core necessary for decentralized social networking for the general use-cases, we have created the basic specifications that can then be extended to deal with particular D-CENT use-cases.

In detail, the first specification included in this deliverable that was produced by the Social Web Working Group is *Activity Streams 2.0*. Loosely modelled on the successful RSS and Atom standards, but based on a profile of Activity Streams 1.0. This specification is the basic building block of syntax ("Social Data Syntax") that will allow D-CENT nodes to communicate to each other. On a high-level, each D-CENT node can produce a stream of activities, such as a vote, the purchase or exchange of commodities, or a release of open civic data. Then other nodes in D-CENT can then receive this 'stream' of activities and parse them, allowing decentralized communication between nodes as any node can both publish and subscribe to any stream of Activity Streams 2.0-based data. Note that the main difference between Activity Streams 1.0 and Activity Streams 2.0 is that URI-based extensibility, rather than the centralized schema built in Activity Streams 1.0, is part of the specification from the beginning, as is compatibility with RDF (and thus the Semantic Web).

The second specification included in this deliverable is the *Activity Vocabulary*. This vocabulary, otherwise known as a 'schema' or 'ontology', is a RDF-compatible vocabulary where each key term is given a URI and the relationships between the terms are clearly defined. This vocabulary is considered to be the 'common core' of various decentralize social networking systems. It is to be expected that in Workpackage 5, new vocabularies for the use-cases of D-CENT will be founded by extending this vocabulary.

Finally, note that the Social API, which defines an API that makes it easy to consume, update, and find ActivityStreams-enabled sites has yet to be defined. However, a number of strong candidates from different open source communities, including the IndieWeb "micropub" work, the ActivityPump specification as implemented currently by MediaGoblin and Status.Net (formerly identi.ca), and the W3C Linked Data Platform all provide relevant work. Lastly, the optional "federation" deliverable that details the larger architecture that includes features such as re-integration of messages, (near) real-time polling, and even authentication is currently still optional. This is because, as shown in the last section of this

deliverable, there are difficult research problems involving security and privacy at the core of federation that are likely out of scope of the D-CENT project and the W3C Social Activity. Also, many members inside the Social Web Working Group have argued that a properly designed Social API should make the need for a separate Federation protocol unnecessary.

An edited version of both the Activity Streams 2.0 specification and the Activity Vocabulary (with excessive non-JSON syntax and other 'boilerplate' text removed) are included below for review. These specifications will be implemented by the D-CENT project but are still unstable (and so subject to change). To find the latest version approved by the W3C Social Web Working Group, see the "Latest published version" link next to each specification. For the latest draft, see the "Latest editor's draft", although that draft may contain text that the Working Group does has not yet reached consensus on.

3.1 Federated Social Networking Use-cases

These use-cases represent only those that the Social Web Working Group has come to consensus on. These do not represent D-CENT use-cases and they may be expanded at any time by the Social Web Working Group, and new use-cases can be suggested by the Social Interest Group to the Social Web. The latest version of the accepted use-cases and suggested use-cases is maintained on a wiki by the Social Web Working Group: http://www.w3.org/wiki/Socialwg/Social_API/User_stories.

3.1.1. User posts a note

1. Eric writes a short note to be shared with his followers.
2. After posting the note, he notices a spelling error. He edits the note and re-posts it.
3. Later, Eric decides that the information in the note is incorrect. He deletes the note.

3.1.2. Reading a user's recent posts

1. Iris finds a comment by Sam on one of her photos funny. She'd like to read more posts by Sam.
2. Iris reads the latest notes by Sam. She also reviews his latest photos.

3.1.3. Following a person

1. Delano meets Beth at a company meeting. They are both user interface designers. He finds her ideas interesting.
2. Delano follows Beth on their company social network.
3. Beth posts a photo from a whiteboarding session at a company retreat.
4. Delano sees the photo in his inbox stream.
5. Ted, Delano's coworker, wants to find new people to follow. He looks at the list of people that Delano follows. He finds Beth in the list, reads her stream, enjoys it, and decides to follow her, too.
6. Beth posts frequently. Delano is having a hard time reading his inbox stream because Beth's activities drown out everyone else's. He stops following Beth.

3.1.4. Inbox

1. Jake is bored at work. He checks his social inbox stream to see what his friends, family, and coworkers are up to.
2. Jake sees in his social stream a note by Tammy about her new apartment. Tammy is his friend.
3. Jake sees in his social stream a photo by Edith from her concert last night. Jake follows Edith but Edith doesn't know Jake. Edith has thousands of followers.
4. Jake sees in his social stream a video from Damon. Damon and Jake are both in the "Boxing Fans" group. Damon posted the video to the group.
5. Jake sees in his social stream a sound file from Carol. Carol is Jake's wife. The sound file is a reminder to stop for groceries after work. Carol posted the sound file only for Jake.
6. Jake sees in his social stream that his friend Tammy has added a new friend, Denise. Jake remembers Denise from high school. Jake requests to add Denise as a friend, too.

3.1.5. Integration : Adding recommendations to bespoke software

1. James maintains an application for managing architectural designs
2. Maggie, a senior architect would like to recommend many of the better designs
3. James uses an existing liking service which allows him to post any recommendations, to provide this
4. This service also allows James to present existing likes for the design in question
5. Maggie gets to like specific designs, and her followers see these as do viewers of these designs
6. James achieves this with a simple inclusion on the associated web page, but could have chosen a more detailed integration if greater control was needed over the user interface

3.1.6. Integration : Adding comments to bespoke software

1. Maria, an IT Architect, has been tasked with encouraging better collaboration on the development of her companies Industrial Processes
2. As these processes are tightly controlled (though generally visible) an associated discussion and evangelisation capability is required
3. Maria integrates with an existing comment capability to store and retrieve comments rather than redeveloping
4. May-Ling sees the comment area with the Processes and suggests changes, as she herself does not have rights to update
5. The process owner gets a notification that someone has commented on this Process
6. Followers of both the Process owner and May-Ling will see this comment event
7. Maria achieves this with a simple inclusion on the associated web page, but could have chosen a more detailed integration if greater control was needed over the user interfac

3.1.7. Direct Messaging

1. Kyle wants to tell Lisa something privately.
2. Kyle sends her a message that no one else can view.
3. Lisa is notified she has a message.
4. Lisa reads the message and responds privately.

3.2 Activity Streams 2.0

W3C Editor's Draft 24 April 2015

This version:

http://jasnell.github.io/w3c-socialwg-activitystreams/activitystreams2.html

Latest published version:

http://www.w3.org/TR/activitystreams-core/

Latest editor's draft:

http://jasnell.github.io/w3c-socialwg-activitystreams/activitystreams2.html

Editor:

James M Snell, IBM

3.2.1. Abstract

This specification details a model for representing potential and completed activities using the JSON format.

Table of Contents

1. Introduction

In the most basic sense, an "Activity" is a semantic description of a completed or ongoing action. It is the goal of this specification to provide a JSON-based syntax that is sufficient to express metadata about activities in a rich, human-friendly but machine-processable and extensible manner. This can include constructing natural-language descriptions or visual representations about the activity, associating actionable information with various types of objects, communicating or recording activity logs, or delegation of potential actions to other applications.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.1 Relationship to JSON Activity Streams 1.0

This section is non-normative.

The JSON Activity Streams 1.0 [AS1] specification was published in May of 2011 and provided a baseline extensible syntax for the expression of completed activities. This specification builds upon that initial foundation by incorporating lessons learned through extensive implementation, community feedback and related ongoing work from a variety of other communities.

While the syntax defined by this specification diverges from that defined by JSON Activity Streams 1.0, the fundamental model defined by that original specification remains intact. Specific processing rules are defined by this specification that allow existing Activity Streams 1.0 documents to be mapped to and processed as an Activity Streams 2.0 document.

This specification incorporates several existing extensions to the 1.0 syntax directly into the 2.0 model. These include portions of the Activity Streams 1.0 Base Schema, Audience Targeting, Responses, and Priority extensions.

1.2 Serialization Notes

This specification describes a JSON-based [RFC7159] serialization syntax for the Activity Vocabulary that follows the conventions defined by the [JSON-LD] specification. While serialization forms other than JSON-LD are possible, alternatives are not discussed by this document.

When serialized, absent properties are represented by either (a) setting the property value to null, or (b) by omitting the property declaration altogether at the option of the publisher; these representations are semantically equivalent. If a property has an array value, the absence of any items in that array MUST be represented by omitting the property entirely or by setting the value to null. The appropriate interpretation of an omitted or explicitly null value is that no value has been assigned as opposed to the view that the given value is empty or nil.

This specification uses IRIs [RFC3987]. Every URI [RFC3986] is also an IRI, so a URI may be used wherever an IRI is named. There are two special considerations: (1) when an IRI that is not also a URI is given for dereferencing, it MUST be mapped to a URI using the steps in Section 3.1 of [RFC3987] and (2) when an IRI is serving as an "id" value, it MUST NOT be so mapped.

Unless otherwise specified, all properties with date and time values MUST conform to the "date-time" production in [RFC3339], with an uppercase "T" character used to separate date and time, and an uppercase "Z" character in the absence of a numeric time zone offset. All such timestamps SHOULD be represented relative to Coordinated Universal Time (UTC).

Activity Streams 2.0 documents MUST be serialized using the UTF-8 character encoding.

The serialized JSON form of an Activity Streams 2.0 document MUST be consistent with what would be produced by the standard JSON-LD 1.0 Processing Algorithms and API [JSON-LD-API] Compaction Algorithm using, at least, the normative JSON-LD @context definition provided here. Implementations MAY augment the provided @context with additional @context definitions but MUST NOT override or change the normative context. Implementations MAY also include in the serialized JSON document additional properties and values not defined in the JSON-LD @context with the understanding that any such properties will likely be unsupported and ignored by consuming implementations that use the standard JSON-LD algorithms. See the Extensibility section for more information on handling extensions within Activity Streams 2.0 documents.

2. Examples

This section is non-normative.

Following are three examples of activities with varying degrees of detail. Each of the examples uses an implied JSON-LD @context equal to that provided here.

Note that the Activity Streams JSON-LD @context maps the prefix "as:" to the base URI "http://www.w3.org/ns/activitystreams#". This means that terms such as "as:Activity" are equivalent to the expanded form "http://www.w3.org/ns/activitystreams#Activity". In order to best illustrate that implementors MUST support both forms, both the compact and expanded forms are used interchangeably by the examples in this document.

2.1 Minimal Activity

Fig. 1 Expresses the statement 'urn:example:person:martin' posted 'http://example.org/foo.jpg'. No additional detail is given.

- JSON-LD

  EXAMPLE 1

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Post",
  "actor": "urn:example:person:martin",
  "object": "http://example.org/foo.jpg"
}
```

2.2 Basic activity with some additional detail

Fig. 2 Expresses the statement "Martin Smith posted an article to the blog 'Martin's Blog' at 3:04 PM UTC on February 2, 2015." Some additional details about the article, actor and target blog are given using properties defined by the Activity Streams 2.0 Vocabulary.

- JSON-LD

EXAMPLE 6

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Post",
  "published": "2015-02-10T15:04:55Z",
  "actor": {
   "@type": "Person",
   "@id": "urn:example:person:martin",
   "displayName": "Martin Smith",
   "url": "http://example.org/martin",
   "image": {
     "@type": "Link",
     "href": "http://example.org/martin/image.jpg",
     "mediaType": "image/jpeg"
   }
  },
  "object" : {
   "@id": "urn:example:blog:abc123/xyz",
   "@type": "Article",
   "url": "http://example.org/blog/2011/02/entry",
   "displayName": "Why I love Activity Streams"
  },
  "target" : {
   "@id": "http://example.org/blog/",
   "@type": "urn:example:types:blog",
   "displayName": "Martin's Blog"
  }
}
```

2.3 An extended activity

Fig. 3 A more extensive, single-entry "Activity Stream" follows.

- JSON-LD

EXAMPLE 11

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Collection",
  "totalItems": 1,
  "items" : [
    {
      "@type": "Post",
      "published": "2011-02-10T15:04:55Z",
      "generator": "http://example.org/activities-app",
      "displayNameMap": {
        "en": "Martin posted a new video to his album.",
        "ga": "Martin phost le fisean nua a albam."
      },
      "actor": {
        "@type": "Person",
        "@id": "urn:example:person:martin",
        "displayName": "Martin Smith",
        "url": "http://example.org/martin",
        "image": {
```

```
            "@type": "Link",
            "href": "http://example.org/martin/image",
            "mediaType": "image/jpeg",
            "width": 250,
            "height": 250
          }
        },
        "object" : {
          "@type": "Image",
          "@id": "http://example.org/album/my_fluffy_cat",
          "preview": {
            "@type": "Link",
            "href": "http://example.org/album/my_fluffy_cat_thumb.jpg",
            "mediaType": "image/jpeg"
          },
          "url": [
            {
              "@type": "Link",
              "href": "http://example.org/album/my_fluffy_cat.jpg",
              "mediaType": "image/jpeg"
            },
            {
              "@type": "Link",
              "href": "http://example.org/album/my_fluffy_cat.png",
              "mediaType": "image/png"
            }
          ]
        },
        "target": {
          "@type": "Album",
          "@id": "http://example.org/album/",
          "displayNameMap": {
            "en": "Martin's Photo Album",
            "ga": "Grianghraif Mairtin"
          },
          "image": {
            "@type": "Link",
            "href": "http://example.org/album/thumbnail.jpg",
            "mediaType": "image/jpeg"
          }
        }
      }
    ]
}
```

3. Model

The Activity Vocabulary defines the abstract model for Activity Streams 2.0. The vocabulary is segmented into a one set of seven core class and an extended set of Activity and Object types common to most "Social" Web applications. The core classes include: Object, Link, Actor, Activity, IntransitiveActivity, Collection and OrderedCollection. Each of these classes is described and illustrated below. The extended Activity and Object types are defined normatively in the Activity Vocabulary specification.

3.1 Object

The Object class is the primary base class for the Activity Streams vocabulary.

In addition to having a global identifier (expressed as an absolute IRI using the JSON-LD @id keyword) and an "object type" (expressed using the JSON-LD @type keyword), all instances of the Object class share a common set of properties normatively defined by the Activity Vocabulary. These include: alias | attachment | attributedTo | content | context | contentMap | displayName | displayNameMap | endTime | generator | icon | image | inReplyTo | location | preview | published | rating | replies | scope | startTime | summary | summaryMap | tag | title | titleMap | updated | url

Previous versions of the Activity Streams format used the objectType property to identify the action type. The objectType property MUST NOT be used within an Activity Streams 2.0 document to represent object type.

While all properties are optional, all Object instances SHOULD at least contain either the displayName or displayNameMap. An object MAY contain both.

Fig. 4 Following is an example Object that uses the JSON-LD @id and @type keywords to express the global identifier and object type:

- JSON-LD

EXAMPLE 16

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@id": "http://example.org/foo",
  "@type": "Note",
  "displayName": "This is a note",
  "attributedTo": {
    "@id": "urn:example:person:joe",
    "@type": "Person",
    "displayName": "Joe Smith"
  },
  "published": "2014-08-21T12:34:56Z"
}
```

Implementations MUST treat all object types in an Activity Streams document as subclasses of Object unless the object is a Link.

Fig. 5 For instance, in the following example, the value of actor is considered to be an instance of Object while the value of object is not:

- JSON-LD

EXAMPLE 21

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@id": "http://example.org/foo",
  "@type": "Share",
  "displayName": "A Test",
  "actor": {
```

```
      "@type": "Person",
      "@id": "http://example.org/~sally",
      "displayName": "Sally"
    },
    "object": {
      "@type": "Link",
      "href": "http://example.org/posts/1"
    }
}
```

The Activity Vocabulary defines a broad range of Object types that are common to many "Social" Web applications. This specification stops short of defining semantically specific properties for most of these objects. External vocabularies can be used to express additional detail not covered by the Activity Vocabulary.

Furthermore, while implementations are free to introduce new types of Objects beyond those defined by the Activity Vocabulary, interoperability issues can arise when applications rely too much on extension types that are not recognized by other implementations. Care should be taken to not unduly overlap with or duplicate the existing Object types. For instance, some vocabularies (e.g. The Good Relations Vocabulary) define their own classes for describing locations. An implementation that wishes, for example, to use a http://purl.org/goodrelations/v1#Location as an object type SHOULD also identify the object as being both a Place and an http://purl.org/goodrelations/v1#Location, as illustrated in the following:

Fig. 6 An Object that is both a Place and a gr:Location:

- JSON-LD

EXAMPLE 26

```
{
  "@context": [
    "http://www.w3.org/ns/activitystreams",
    {
      "gr": "http://purl.org/goodrelations/v1#"
    }
  ],
  "@type": ["Place", "gr:Location"],
  "displayName": "Sally's Restaurant",
  "longitude": 12.34,
  "latitude": 56.78,
  "gr:category": "restaurants/french_restaurants"
}
```

Certain properties defined by some External Vocabularies can overlap or duplicate those defined by the Activity Vocabulary. Where such overlap exists, for the sake of consistent interoperability, implementers MUST favor the use of properties defined by the Activity Vocabulary.

3.2 Natural Language Values

Several properties defined by the Vocabulary are defined as having natural language values. These are representations of human-readable character sequences using one or more languages. Within the JSON-LD serialization, they are expressed as either (1) a single JSON string or (2) a JSON object mapping

[RFC5646] Language-Tags to localized, equivalent translations of the same string value. In [JSON-LD], such constructs are referred to as "Language Maps". In the serialized JSON-LD, these two forms are differentiated using a simple property naming convention, for instance: "displayName" identifies the JSON string form for the displayNameproperty while "displayNameMap" represents the Language Map form.

Fig. 7 A single displayName String value without language information:

- JSON-LD

EXAMPLE 31

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Object",
  "displayName": "This is the title"
}
```

Fig. 8 Multiple, language-specific values:

- JSON-LD

EXAMPLE 36

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Object",
  "displayNameMap": {
    "en": "This is the title",
    "fr": "C'est le titre",
    "sp": "Este es el titulo"
  }
}
```

Every key in the Language Map form MUST be a valid [RFC5646] Language-Tag. The associated values MUST be Strings.

The Activity Vocabulary defines four specific natural language values: displayName, title, summary, and content. Accordingly, the Activity Streams JSON-LD @context definition respectively maps the terms "displayName", "title", "summary", and "content" for representing the JSON string forms and the terms "displayNameMap", "titleMap", "summaryMap", and "contentMap" for representing the Language Map forms.

The default language for document or an individual object can be established using the JSON-LD @language keyword within a @context definition. For instance:

Fig. 9 Establishing a default language context:

- JSON-LD

EXAMPLE 41

```
{
  "@context": [
    "http://www.w3.org/ns/activitystreams",
    {"@language": "en"}
  ],
  "@type": "Object",
```

```
  "displayName": "This is the title"
}
```

The JSON-LD format generally supports one additional way of associating language tag information with a literal string value using what JSON-LD calls a "value object", as illustrated below:

Fig. 10 Specifying language in a JSON-LD value object:

- JSON-LD

EXAMPLE 46

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Object",
  "displayName": {
    "@value": "This is the title",
    "@language": "en"
  }
}
```

While Activity Streams 2.0 implementations MAY use JSON-LD value objects in this manner, doing so is strongly discouraged. Implementations SHOULD, instead, use the Language Map form. The one situation where use of the value object cannot be avoided is when a default language context has been established and a particular language-sensitive field needs to be explicitly excluded from that context, as in the following example:

Fig. 11 Excluding a natural language property from the language context:

- JSON-LD

EXAMPLE 51

```
{
  "@context": [
    "http://www.w3.org/ns/activitystreams",
    { "@language": "en" }
  ],
  "@type": "Object",
  "displayName": {
    "@value": "This is the title"
  }
}
```

By explicitly omitting the @language from the value of displayName in the JSON-LD example, the displayName is excluded from the default language context. However, because this mechanism requires specific understanding of JSON-LD algorithms, and makes publishers intention less obvious and visible, implementations SHOULD avoid such cases as much as possible.

3.3 Link

A Link describes a qualified, indirect reference to another resource that is closely related to the conceptual model of Links as established in [RFC5988].

The target URI of the Link is expressed using the required href keyword. In addition, all Link instances share the following common set of optional properties as normatively defined by the Activity Vocabulary: displayName | displayNameMap | hreflang | mediaType | rel | title | titleMap | height | width | duration

For example, all Objects can contain an image property whose value describes a graphical representation of the containing object. This property will typically be used to provide the URL to an image (e.g. JPEG, GIF or PNG) resource that can be displayed to the user. Any given object might have multiple such visual representations -- multiple screenshots, for instance, or the same image at different resolutions. In Activity Streams 2.0, there are essentially three ways of describing such references.

Fig. 12 To reference a single image without any additional metadata, a direct association can be expressed as a JSON string containing an absolute or relative IRI.

- JSON-LD

EXAMPLE 56

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Application",
  "@id": "http://example.org/application/123",
  "displayName": "My Application",
  "image": "http://example.org/application/123.png"
}
```

Fig. 13 Alternatively, if additional metadata is required (such as the MIME content type of the referenced resource) a Link can be used:

- JSON-LD

EXAMPLE 61

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Application",
  "@id": "http://example.org/application/123",
  "displayName": "My Application",
  "image": {
    "@type": "Link",
    "href": "http://example.org/application/123.png",
    "mediaType": "image/png"
  }
}
```

Formally, the former example establishes an unqualified direct relationship with the image resource while the latter creates a qualified, indirect relationship that allows additional properties about the relationship to be specified. Such properties (e.g. mediaType, hreflang, rel, etc) describe the Link itself as opposed to describing the referenced resource. For many practical applications, this distinction will likely be fairly insignificant but it is still worth keeping in mind.

Fig. 14 If more than one value is to be expressed, A JSON Array with a mix of strings and Links can be used:

- JSON-LD

EXAMPLE 66

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Application",
  "@id": "http://example.org/application/123",
  "displayName": "My Application",
  "image": [
    "http://example.org/application/abc.gif",
    {
      "@type": "Link",
      "href": "http://example.org/application/123.png",
      "mediaType": "image/png"
    }
  ]
}
```

Individual items contained in such an array are independent of the others and no significance is given to the ordering.

RFC 5988 defines that all Links have a "link relation" that describes the contextual purpose of the link. Within a Link, the rel property provides the link relation value. If no rel property is specified, the link relation is considered to be unspecified. Any given Link can have multiple link relation values. In the JSON-LD serialization, A single link relation is expressed as a single JSON string. Multiple link relations are expressed as an array of JSON strings.

In the following example, two separate references are provided. The link relation of the first is unspecified, while the link relation of the second is "thumbnail".

Fig. 15

- JSON-LD

EXAMPLE 71

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Application",
  "@id": "http://example.org/application/123",
  "displayName": "My Application",
  "image": [
    "http://example.org/application/abc.gif",
    {
      "@type": "Link",
      "href": "http://example.org/application/123.png",
      "mediaType": "image/png",
      "rel": "thumbnail"
    }
  ]
}
```

It ought to be noted that the [HTML5] specification provides it's own alternative definition of a "link relation" that differs slightly from the [RFC5988] definition. In the HTML5 definition, any string that does not contain the "space" U+0020, "tab" (U+0009), "LF" (U+000A), "FF" (U+000C), "CR" (U+000D) or "," (U+002C) characters can be used as a valid link relation. To promote interoperability, Activity Streams

2.0 implementations MUST only use link relations that are valid in terms of both the [RFC5988] and [HTML5] definitions.

Note that the Link and Object classes are disjoint from one another. That is, any given Object cannot also be a Link.

3.4. Actor

Actor objects are specializations of the base Object type that represent entities capable of carrying out an Activity. The Actor class is the base class for all Actor objects. The Activity Vocabulary provides the normative definition of five specific types of Actors: Application | Group | Person | Process | Service.

This specification intentionally defines Actors in only the most generalized way, stopping short of defining semantically specific properties for each. All Actor objects are specializations of Object and inherit all of the core properties common to all Object's. External vocabularies (VCard for instance) can be used to express additional detail not covered by the Activity Vocabulary.

Fig. 16 An Activity with a Person actor:

- JSON-LD

EXAMPLE 76

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Share",
  "actor": {
    "@type": "Person",
    "@id": "acct:sally@example.org",
    "displayName": "Sally Smith"
  },
  "object": {
    "@type": "Note",
    "content": "This is a simple note"
  }
}
```

Fig. 17 An Activity with a Person actor extended with VCard properties:

- JSON-LD

EXAMPLE 81

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Share",
  "actor": {
    "@type": "Person",
    "@id": "acct:sally@example.org",
    "displayName": "Sally Smith",
    "vcard:given-name": "Sally",
    "vcard:family-name": "Smith"
  },
  "object": {
    "@type": "Note",
    "content": "This is a simple note"
```

```
    }
}
```

While implementations are free to introduce new types of Actors beyond those defined by the Activity Vocabulary, interoperability issues can arise when applications rely too much on extension types that are not recognized by other implementations. Care should be taken to not unduly overlap with or duplicate the existing Actor types. For instance, some vocabularies (e.g. VCard) define their own classes for describing people. An implementation that wishes, for example, to use a vcard:Individual as an Actor SHOULD identify that Actor as being *both* a Person and an vcard:Individual, as illustrated in the following:

Fig. 18 An Actor that is both a Person and a vcard:Individual:

- JSON-LD

EXAMPLE 86

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Share",
  "actor": {
    "@type": ["Person", "vcard:Individual"],
    "@id": "acct:sally@example.org",
    "displayName": "Sally Smith",
    "vcard:given-name": "Sally",
    "vcard:family-name": "Smith"
  },
  "object": {
    "@type": "Note",
    "content": "This is a simple note"
  }
}
```

3.5. Activity

Activity objects are specializations of the base Object type that provide information about pending, ongoing or completed actions.

In addition to common properties supported by all Object instances, Activity objects support the following additional properties defined by the Vocabulary: actor | object | target | origin | result | priority | to | bto | cc | bcc

The JSON-LD @type keyword is used to identify the type of action the Activity Statement represents. Previous versions of the Activity Streams format used the verb property to identify the action type. The verb MUST NOT be used within an Activity Streams 2.0 Activity to represent the action type.

Fig. 19 The following example illustrates a simple Activity:

- JSON-LD

EXAMPLE 91

```
{
```

```
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Like",
  "@id": "urn:example:activity:1",
  "actor": "http://example.org/profiles/joe",
  "object": "http://example.com/notes/1",
  "published": "2014-09-30T12:34:56Z"
}
```

The Activity Vocabulary defines a broad range of Activity types that are common to many "Social" Web applications. This specification stops short of defining semantically specific properties for most of these activities. External vocabularies can be used to express additional detail not covered by the Activity Vocabulary.

Furthermore, while implementations are free to introduce new types of Activites beyond those defined by the Activity Vocabulary, interoperability issues can arise when applications rely too much on extension types that are not recognized by other implementations. Care should be taken to not unduly overlap with or duplicate the existing Activity types. For instance, some vocabularies (e.g. Schema.org) define their own classes for describing actions. An implementation that wishes, for example, to use http://schema.org/LikeAction as an Activity SHOULD identify that Object as being both a Like and an http://schema.org/LikeAction, as illustrated in the following:

Fig. 20 An Activity that is both a Like and a http://schema.org/LikeAction:

- JSON-LD

EXAMPLE 96

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": ["Like", "http://schema.org/LikeAction"],
  "@id": "urn:example:activity:1",
  "actor": "http://example.org/profiles/joe",
  "object": "http://example.com/notes/1",
  "published": "2014-09-30T12:34:56Z"
}
```

3.1 Audience Targeting

Conceptually, every Activity has both a Primary and Secondary audience. The Primary audience consists of those entities either directly involved in the performance of the activity or who "own" the objects involved. The Secondary audience consists of the collection of entities sharing an interest in the activity but who might not be directly involved (e.g. "followers").

For instance, suppose a social network of three individuals: Bob, Joe and Jane. Bob and Joe are each friends with Jane but are not friends with one another. Bob has chosen to "follow" activities for which Jane is directly involved. Jane shares a file with Joe.

In this example, Jane and Joe are each directly involved in the file sharing activity and together make up the Primary Audience for that event. Bob, having an interest in activities involving Jane, is the Secondary Audience. Knowing this, a system that produces or consumes the activity can intelligently notify each person of the event.

While there are means (based on the action type, actor, object and target of the activity) to infer the primary audience for many types of activities, heuristics do not work in every case and do not provide a means of identifying the secondary audience. The to, cc, bto and bcc properties MAY be used within an Activity to explicitly identify the Primary and Secondary audiences.

The prototypical use case for an Activity containing these properties is the publication and redistribution of Activities through an intermediary. That is, an event source generates the activity and publishes it to the intermediary which determines a subset of events to display to specific individual users or groups. Such a determination can be made, in part, by identifying the Primary and Secondary Audiences for each activity.

When the event source generates the activity and specifies values for the `to` and `cc` fields, the intermediary SHOULD redistribute that event with the values of those fields intact, allowing any processor to see who the activity has been targeted to. This is precisely the same model used by the `to` and `cc` fields in email systems.

There are situations, however, in which disclosing the identity of specific members of the audience may be inappropriate. For instance, a user may not wish to let other users know that they are interested in various topics, individuals or types of events. To support this option, an event source generating an activity MAY use the bto and bcc properties to list entities to whom the activity should be privately targeted. When an intermediary receives an activity containing these properties, it MUST remove those values prior to redistributing the activity. The intent is that systems MUST consider entities listed within the `bto` and `bcc` properties as part of the Primary and Secondary audience but MUST NOT disclose that fact to any other party.

Audience targeting information included within an Activity only describes the intent of the activity creator. With clear exception given to the appropriate handling of `bto` and `bcc`, this specification leaves it up to implementations to determine how the audience targeting information is used.

3.6. Collection

Collection objects are a specialization of the base Object that contain a listing of other Objects or Links. The Collection object is used primarily as the root of an Activity Streams Document, but can also be used as the value of object properties.

The model for collections within the Activity Vocabulary is designed largely around the abstract model of "logical feeds" and "pages" discussed in [RFC5005], Section 1.2. Specifically, collections have both a logical model and a physical serialization. While the logical model of a collection might contain a large number of member objects, any single serialized representation might include only a subset of those objects, with "paging" Links used to reference additional serialized representations that include other subsets. Such representations are known as "multi-page collections", with each serialized subset representing a single "page".

In addition to the common properties shared by all Object instances, `Collection` objects set of properties defined by the Vocabulary. These include: items | totalItems | itemsPerPage | startIndex | current | next | prev | first | last | self

Collections can be ordered or unordered. In the JSON-LD serialization, the unordered members of a Collection are represented using the items property while the ordered members are represented using the orderedItems property. The normative JSON-LD @context definition maps both the items and

orderedItems properties to the Activity Vocabulary items term. However, the orderedItems property is defined in the JSON-LD @context as @container = @list, indicating that the items are strictly ordered.

Note that when using orderedItems within a Collection object, the order imposed on the items applies only to the subset of items directly contained within the one Collection object. Implementations MAY use the ***OrderedCollection*** class to identify collections whose entire logical set of members are strictly ordered.

Fig. 21 The following is a simple unordered collection with paging:

- JSON-LD

EXAMPLE 101

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Collection",
  "totalItems": 10,
  "itemsPerPage": 1,
  "next": "http://example.org/foo?page=2",
  "self": "http://example.org/foo?page=1",
  "items": [
    {
      "@type": "Post",
      "actor": "urn:example:person:sally",
      "object": "http://example.org/foo"
    }
  ]
}
```

Fig. 22 The following is a simple ordered collection with paging:

- JSON-LD

EXAMPLE 106

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "OrderedCollection",
  "totalItems": 10,
  "itemsPerPage": 1,
  "next": "http://example.org/foo?page=2",
  "self": "http://example.org/foo?page=1",
  "startIndex": 0,
  "orderedItems": [
    {
      "@type": "Post",
      "actor": "urn:example:person:sally",
      "object": "http://example.org/foo"
    }
  ]
}
```

In ordered collections that use paging, the startIndex property is used to indicate the relative index position within the logical set of items of the first item contained in the orderedItems list. A startIndex of 0 indicates the first item in the collection.

Using paging with an OrderedCollection can be tricky because there is no guarantee that implementations will process the distinct pages in any predictable order. Therefore, it is strongly recommended that OrderedCollection instances that use paging always include the first, next, prev and last paging properties. Implementations that wish to reconstruct the appropriate complete ordering of member instances in the logical collection would navigate to the OrderedCollection instance identified by the first (or last) link, then recursively follow the next (or prev) link until all pages have been processed.

4. Activity Streams Document

An ***Activity Streams Document*** is a JSON-LD document whose root value is a Collection and whose MIME media type is "application/activity+json".

5. Extensibility

In Activity Streams 2.0, an "extension" is any property not defined by the Activity Vocabulary. Consuming implementations that encounter unfamiliar extensions MUST NOT stop processing or signal an error and MUST continue processing the items as if those properties were not present. Support for specific extensions can vary across implementations and no normative processing model for extensions is provided.

While consuming implementations are not required to use the standard JSON-LD Processing Algorithms [JSON-LD-API], it is important to note that the algorithms, as currently defined, will silently ignore any property that is not defined in a JSON-LD @context. Implementations that publish Activity Streams 2.0 documents that contain extension properties SHOULD provide a @context definition of those extensions.

It is also important to note that there are valid JSON constructs which cannot be used within a JSON-LD document. For instance, JSON-LD forbids "arrays of arrays" as used, for example, by the popular GeoJSON specification. While implementations are free to use such constructs as extensions within an Activity Streams 2.0 document, consumers that use the standard JSON-LD Processing Algorithms will be required to either ignore such extensions or map those to alternative compatible constructs prior to applying the JSON-LD algorithms. Simple GeoJSON Points, for instance, can be mapped to Place objects, while more complex geometries can be converted to GeoSparql "Well-Known Text" representations as illustrated in the non-normative examples below:

Fig. 23 GeoJSON Point Coordinates:

EXAMPLE 111

```
{
  "type": "Point",
  "coordinates": [36.74, -119.77]
}
```

Fig. 24 The Equivalent `Place` alternative:

- JSON-LD

EXAMPLE 112

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Place",
```

```
  "latitude": 36.74,
  "longitude": -119.77
}
```

Fig. 25 GeoJSON Polygon Coordinates:

EXAMPLE 117

```
{
  "type": "Polygon",
  "coordinates": [
    [
      [100.0, 0.0],
      [101.0, 0.0],
      [101.0, 1.0],
      [100.0, 1.0],
      [100.0, 0.0]
    ]
  ]
}
```

Fig. 26 The Equivalent GeoSparql Well-Known-Text alternative:

- JSON-LD

EXAMPLE 118

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "gsp:Geometry",
  "gsp:asWKT": "Polygon((100.0, 0.0, 101.0, 0.0, 101.0, 1.0, 100.0, 1.0,
100.0, 0.0))"
}
```

5.1 Handling of JSON-LD Compact IRIs

In JSON-LD, a "Compact IRI" is a type of shorthand notation that allows absolute IRI values to be split into an absolute base IRI and a relative token. For instance, in the following example:

Fig. 27

EXAMPLE 123

```
{
  "@context": {
    "foaf": "http://xmlns.com/foaf/0.1/"
  },
  "foaf:givenName": "Sam"
}
```

The property name foaf:givenName is a Compact IRI that expands to "http://xmlns.com/foaf/0.1/givenName" when processed by a JSON-LD implementation.

While there are differences of opinion on how extensively Compact IRIs ought to be used within JSON-LD document instances, JSON-LD implementations are required to support them.

In order to simplify implementation and encourage interoperable reuse, the following predefined prefixes are defined in the normative Activity Streams 2.0 JSON-LD @context. Implementations MUST NOT redefine any of the provided prefix mappings.

Prefix	Expanded	Name
as:	http://www.w3.org/ns/activitystreams#	Activity Streams Vocabulary
dc:	http://purl.org/dc/elements/1.1/	Dublin Core
dct:	http://purl.org/dc/terms/	Dublin Core Terms
dctypes:	http://purl.org/dc/dcmitype/	Dublin Core Types
foaf:	http://xmlns.com/foaf/0.1/	Friend of a Friend
vcard:	http://www.w3.org/2006/vcard/ns#	VCard
org:	http://www.w3.org/ns/org#	W3C Org Ontology
prov:	http://www.w3.org/ns/prov#	W3C Provenance Ontology
geo:	http://www.w3.org/2003/01/geo/wgs84_pos#	W3C Basic Geo Ontology
gsp:	http://www.opengis.net/ont/geosparql#	Geo-Sparql
ical:	http://www.w3.org/2002/12/cal/ical#	iCalendar Ontology
oa:	http://www.w3.org/ns/oa#	Web Annotation Data Model

For instance, the property name foaf:givenName would expand to "http://xmlns.com/foaf/0.1/givenName", while the property prov:actingOnBehalfOf would expand to "http://w3.org/ns/prov#actingOnBehalfOf".

6. Mentions, Tags and Other Common Social Microsyntaxes

Many social software systems use special text-based microsyntaxes that allow users to define special addressing for notifications, linking, or categorization within objects. For example, including text such as "@username" within an object's content will often route the object to a special "mentions" or "inbox" stream for a particular user. Likewise, including text such as "#topic" within the object's content will often mark the object as being related to the topic "topic". Such mechanisms are commonly referred to as "mentions" and "hashtags", respectively.

While such microsyntaxes MAY be used within the values of the content, displayName, summary, and title properties on an Activity Streams Object, implementations SHOULD NOT be required to parse the values of those properties in order to determine the appropriate routing of notifications, categorization or linking between objects. Instead, publishers SHOULD make appropriate use of the Activity Streams Vocabulary terms provided specifically for these purposes.

For example, suppose that an author wishes to send a note of thanks to another user named "sally" with a hashtag of "#givingthanks". A typical way this message would appear within the content of a note is shown below:

Fig. 28 A simple note with a mention an a hashtag:

"Thank you @sally for all your hard work! #givingthanks"

A typical social software implementation would typically render such a content such that "@sally" is replaced with a hyperlink to "sally"'s social profile page and "#givingthanks" is replaced with a hyperlink to a listing of other notes that have been "tagged" with the same topic. Most implementations would also send a special notification to sally letting her know that a note mentioning her has been posted.

The following illustrates an equivalent Activity Streams Note object:

Fig. 29 Mentions and Tags within an Activity Streams Note

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Note",
  "content": "Thank you @sally for all your hard work! #givingthanks",
  "to": {
    "@type": "Person",
    "@id": "http://example.org/people/sally",
    "alias": "@sally"
  },
  "tag": {
    "@id": "http://example.org/tags/givingthanks",
    "alias": "#givingthanks"
  }
}
```

The `to` property indicates that the user "@sally" is to be considered part of the primary audience of the note and should therefore receive notification. The tag property associates the Note with a reference to "http://example.org/tags/givingthanks". Note that the note's content still includes the "@sally" and "#givingthanks" microsyntaxes but that consuming implementations are not required to parse those in order to make the appropriate associations.

In the case a publisher wishes to indicate a mention without an associated notification, the publisher can use the Mention object type as a value of the tag property. The Mention object is a subclass of Link.

Fig. 30 Mentions and Tags within an Activity Streams Note

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Note",
  "content": "Thank you @sally for all your hard work! #givingthanks",
  "tag": [
    {
      "@id": "http://example.org/tags/givingthanks",
      "alias": "#givingthanks"
    },
    {
      "@type": "Mention",
      "href": "http://example.org/people/sally",
      "displayName": "@sally"
    }
  ]
}
```

7. Security Considerations

Publishers or Consumers implementing Activity Streams as a stream of public data may also want to consider the potential for unsolicited commercial or malicious content and should take preventative measures to recognize such content and either identify it or not include it in their implementations.

Publishers should take reasonable measures to ensure potentially malicious user input such as cross-site scripting attacks are not included in the Activity Streams data they publish.

Consumers that re-emit ingested content to end-users MUST take reasonable measures if emitting ingested content to make sure potentially malicious ingested input is not re-emitted.

Consumers that re-emit ingested content for crawling by search engines should take reasonable measures to limit any use of their site as a Search Engine Optimization loophole. This may include converting untrusted hyperlinks to text or including a rel="nofollow" attribute.

Consumers should be aware of the potential for spoofing attacks where the attacker publishes activities or objects with falsified property values with the intent of injecting malicious content, hiding or corrupting legitimate content, or misleading users.

Activity Streams are JSON Documents and are subject to the same security considerations described in [RFC7159].

Activity Streams implementations handle URIs. See Section 7 of [RFC3986].

Activity Streams implementations handle IRIs. See Section 8 of [RFC3987].

8. IANA Considerations

8.1. The `application/activity+json` Media Type

This specification registers the ***application/activity+json*** MIME Media Type specifically for identifying documents conforming to the Activity Streams 2.0 format.

Type name:	**application**
Subtype name:	activity+json
Required parameters:	None
Optional parameters:	profile: The profile parameter for the application/activity+json media type allows one or more profile URIs to be specified. These profile URIs have the identifier semantics defined in [RFC6906]. The "profile" media type parameter MUST be quoted. It contains a non-empty list of space-separated URIs (the profile URIs). profile-param = "profile=" profile-value profile-value = <"> profile-URI 0*(1*SP profile-URI) <"> profile-URI = URI The "URI" in the above grammar refers to the "URI" as defined in Section 3 of [RFC3986].
Encoding considerations:	Resources that use the "application/activity+json" Media Type are required to conform to all of the requirements for the "application/json" Media Type and are therefore subject to the same encoding considerations specified in Section 11 of [RFC7159].
Security considerations:	As defined in this specification.
Contact:	James M Snell <jasnell@gmail.com>

9.2 The `application/stream+json` Media Type

This specification registers the ***application/stream+json*** MIME Media Type specifically for identifying documents conforming to the JSON Activity Streams 1.0 [AS1] format.

Type name:	**application**
Subtype name:	stream+json

Required parameters:	None
Optional parameters:	profile: The profile parameter for the application/stream+json media type allows one or more profile URIs to be specified. These profile URIs have the identifier semantics defined in [RFC6906]. The "profile" media type parameter MUST be quoted. It contains a non-empty list of space-separated URIs (the profile URIs). profile-param = "profile=" profile-value profile-value = <"> profile-URI 0*(1*SP profile-URI) <"> profile-URI = URI The "URI" in the above grammar refers to the "URI" as defined in Section 3 of [RFC3986].
Encoding considerations:	Resources that use the "application/stream+json" Media Type are required to conform to all of the requirements for the "application/json" Media Type and are therefore subject to the same encoding considerations specified in Section 11 of [RFC7159].
Security considerations:	As defined in [AS1]
Contact:	James M Snell <jasnell@gmail.com>

3.3 Activity Vocabulary

W3C Editors Draft 24 April 2015

This version:

http://jasnell.github.io/w3c-socialwg-activitystreams/activitystreams2-vocabulary.html

Latest published version:

http://www.w3.org/TR/activitystreams-vocabulary/

Latest editor's draft:

http://jasnell.github.io/w3c-socialwg-activitystreams/activitystreams2-vocabulary.html

Editor:

James M Snell, IBM

3.3.1 Abstract

This specification describes the Activity vocabulary.

3.3.2. Introduction

The Activity Streams 2.0 Core Syntax defines the JSON syntax for Activity Streams. This document defines the vocabulary properties.

The Activity Streams 2.0 Vocabulary defines a set of abstract classes and properties that describe past, present and future Activities. The vocabulary is defined in two parts:

1. A Core set of properties describing the generalized structure of an Activity; and
2. An Extended set of properties that cover specific types of Activities and Artifacts common to many "Social" Web application systems.

While not all Activity Streams 2.0 implementations are expected to implement support for the Extended properties, all implementations MUST at least be capable of serializing and deserializing the Extended properties in accordance with the Activity Streams 2.0 Core Syntax.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3.3.3. Conventions

This specification uses IRIs [RFC3987]. Every URI [RFC3986] is also an IRI, so a URI may be used wherever an IRI is named. There are two special considerations: (1) when an IRI that is not also a URI is given for dereferencing, it MUST be mapped to a URI using the steps in Section 3.1 of [RFC3987] and (2) when an IRI is serving as an "id" value, it MUST NOT be so mapped.

Unless otherwise specified, all properties defined as xsd:dateTime values MUST conform to the "date-time" production in [RFC3339], with an uppercase "T" character used to separate date and time, and an uppercase "Z" character in the absence of a numeric time zone offset. All such timestamps SHOULD be represented relative to Coordinated Universal Time (UTC).

Note that this document uses illustrative examples using JSON-LD. Only the JSON-LD examples are normative. The non-JSON-LD examples are strictly informational.

3.3.4. Core Classes

The Activity Vocabulary Core Classes provide the basis for the rest of the vocabulary.

Base URI: http://www.w3.org/ns/activitystreams#.

The Activity Streams 2.0 Core Classes include: Object | Link | Activity | IntransitiveActivity | Actor | Collection | OrderedCollection

Class	Description		Example
Object	URI:	http://www.w3.org/ns/activitystreams#Object	• JSON-LD EXAMPLE 1 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Object", "@id": "urn:example:object:1",
	Notes:	Describes an object of any kind. The Object class serves as the base class for most of the other kinds of objects defined in the Activity Vocabulary, include other Core classes such as Activity, IntransitiveActivity, Actor, Collection and OrderedCollection.	
	Disjoint With:	Link	

	Properties:	**alias** | **attachment** | **attributedTo** | **content** | **context** | **displayName** | **endTime** | **generator** | **icon** | **image** | **inReplyTo** | **location** | **preview** | **published** | **rating** | **replies** | **scope** | **startTime** | **summary** | **tag** | **title** | **updated** | **url**	"displayName": "A Simple, non-specific object" }

Link	URI:	http://www.w3.org/ns/activitystreams#Link	• JSON-LD
			
			EXAMPLE 6
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Link",
			"href": "http://example.org/abc",
			"hreflang": "en",
			"mediaType": "text/html",
			"displayName": "An example link"
			}

	Notes:	A Link is an indirect, qualified reference to a resource identified by a URL. The fundamental model for links is established by [RFC5988]. Many of the properties defined by the Activity Vocabulary allow values that are either instances of Object or Link. When a Link is used, it establishes a qualified relation connecting the subject (the containing object) to the resource identified by the href.	
	Disjoint With:	Object	
	Properties:	**href** | **rel** | **mediaType** | **displayName** | **title** | **hreflang** | **height** | **width** | **duration**	

Activity	URI:	http://www.w3.org/ns/activitystreams#Activity	• JSON-LD
			
			EXAMPLE 11
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Activity",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"

	Notes:	An Activity is a subclass of Object that describes some form of action that may happen, is currently happening, or that has already happened. The Activity class itself serves as an abstract base class for all types of activities. It is important to note that the Activity class itself does not carry any specific semantics about the kind of action being taken.	
	Extends:	Object	

	Properties:	**actor** | **object** | **target** | **result** | **origin** | **priority** | **to** | **bto** | **cc** | **bcc**
		
		Inherits all properties from Object.
			},
			"object": {
			"@type": "Note",
			"displayName": "A Note"
			}
			}

IntransitiveActivity	URI:	http://www.w3.org/ns/activitystreams#IntransitiveActivity	• JSON-LD
			
			EXAMPLE 16
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Travel",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"target": {
			"@type": "Place",
			"displayName": "Work"
			}
			}

	Notes:	Instances of IntransitiveActivity are a subclass of Activity whose actor property identifies the direct object of the action as opposed to using the object property.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity except object.	

Actor	URI:	http://www.w3.org/ns/activitystreams#Actor	• JSON-LD
			
			EXAMPLE 21
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Actor",
			"displayName": "Sally"
			}

	Notes:	An Actor is any entity that is capable of being the primary actor for an Activity.	
	Extends:	Object	
	Properties:	Inherits all properties from Object.	

Collection	URI:	http://www.w3.org/ns/activitystreams#Collection	• JSON-LD
			
			EXAMPLE 26
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 2,
			"itemsPerPage": 2,
			"items": [
			{
			"@type": "Note",
			"displayName": "A Simple Note"
			},
			{
			"@type": "Note",
			"displayName": "Another Simple Note"
			}
			]
			}

	Notes:	A Collection is a subclass of Object that represents ordered or unordered sets of Object or Link instances. The model for collections within the Activity Vocabulary is designed largely around the abstract model of "logical feeds" and "pages" discussed in [RFC5005], Section 1.2. Specifically, a "logical collection" is a complete set of Object or Link instances while an individual Collection object may contain only a specific subset of those members. Paging properties such as next, prev, first, and last are used to connect multiple Collection objects containing different subsets of member items of a single logical collection together. When multiple, separate Collection instances share the same identity (represented in the JSON-LD serialization using the @id property) and each specifies distinct, non-overlapping subsets of member items (using the items property), the membership of the logical collection is the intersection of each of the subsets. However, when there is overlap in the items contained by multiple Collection instances (e.g. when Object or Link intances with the same identity appear in more than one Collection), member instances with the most recently updated timestamps -- as determined by examining first the updated and then published properties -- *MUST* be considered to be the most current representations. If those members do not have either an updated or published property specified, then the containing Collection object's updated and published properties are used. If no timestamps are available to use for comparison, implementations *MAY* use other means to determine the precedence of member instance representations.	

		When the members of a Collection are ordered, the ordering imposed applies only to the subset of members contained within that one specific Collection instance.	
	Extends:	Object	
	Properties:	totalItems | itemsPerPage | current | next | prev | first | last | self | items Inherits all properties from Object.	

OrderedCollection	URI:	http://www.w3.org/ns/activitystreams#OrderedCollection	• JSON-LD
			
			EXAMPLE 31
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "OrderedCollection",
			"totalItems": 2,
			"itemsPerPage": 2,
			"startIndex": 0,
			"orderedItems": [
			{
			"@type": "Note",
			"displayName": "A Simple Note"
			},
			{
			"@type": "Note",
			"displayName": "Another Simple Note"
			}
			]
			}

	Notes:	A subclass of Collection in which members of the logical collection are assumed to always be strictly ordered. Using paging with an OrderedCollection can be tricky because there is no guarantee that implementations will process the distinct pages in any predictable order. Therefore, it is strongly recommended that OrderedCollection instances that require paging always include the first, next, prev and last paging properties. Implementations that wish to reconstruct the appropriate complete ordering of member instances in the logical collection would navigate to the OrderedCollection instance identified by the first (or last) link, then recursively follow the next (or prev) link until all pages have been processed.	
	Extends:	Collection	
	Properties:	**startIndex** **Inherits all properties from Collection.**	

3.3.5. Extended Classes

Base URI: *http://www.w3.org/ns/activitystreams#.*

The Activity Streams 2.0 Extended Classes include Activity and Object subclasses that are common to many "Social" Web applications. They are divided into three sets:

- Activity Types

- Actor Types
- Object Types

Support for specific extended vocabulary classes is expected to vary, with implementations only selecting the extended classes and properties that make sense within the specific context and requirements of those applications. However, to avoid possible interoperability issues, implementations MUST avoid using extension classes or properties that unduly overlap with or duplicate the extended vocabulary defined here.

3.3.6. Activity Types

All Activity Types inherit the properties of the base Activity class. Some specific Activity Types are subclasses or specializations of more generalized Activity Types (for instance, the Like Activity Type is a more specific form of the Respond Activity Type).

The Activity Types include: Accept | Add | Announce | Arrive | Assign | Block | Complete | Confirm | Create | Delete | Dislike | Experience | Favorite | Flag | Follow | Give | Ignore | Invite | Join | Leave | Like | Listen | Move | Offer | Post | Question | Reject | Read | Remove | Respond | Review | Save | Share | TentativeReject | TentativeAccept | Travel | Undo | Update | View | Watch

Class	Description		Example

Accept	URI:	http://www.w3.org/ns/activitystreams#Accept	• JSON-LD
			
			EXAMPLE 36
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Accept",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type": "Invite",
			"actor": "acct:john@example.org",
			"object": {
			"@type": "Event",
			"displayName": "A Party!"
			}
			}
			}

	Notes:	Indicates that the actor accepts the object. The target property can be used in certain circumstances to indicate the context into which the object has been accepted. For instance, when expressing the activity, "Sally accepted Joe into the Club", the "target" would identify the "Club".	
	Extends:	Respond	
	Properties:	Inherits all properties from Respond.	
TentativeAccept	URI:	http://www.w3.org/ns/activitystreams#TentativeAccept	• JSON-LD EXAMPLE 41 {
	Notes:	A specialization of Accept indicating that the acceptance is tentative.	
	Extends:	Accept	

	Properties:	Inherits all properties from Accept.	<pre>"@context": "http://www.w3.org/ns/activitystreams", "@type": "TentativeAccept", "actor": { "@type": "Person", "displayName": "Sally" }, "object": { "@type": "Invite", "actor": "acct:john@example.org", "object": { "@type": "Event", "displayName": "A Party!" } } }</pre>

Add	URI:	http://www.w3.org/ns/activitystreams#Add	• JSON-LD
			
			EXAMPLE 46
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Add",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": "http://example.org/abc"
			}
			
			JSON-LD
			
			EXAMPLE 51
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Add",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type": "Image",
			"displayName": "A picture of my cat",
			"url": "http://example.org/img/cat.png"
			},
			"target": {
			"@type": "Album",
			"displayName": "My Cat Pictures"
			}
			}

	Notes:	Indicates that the actor has added the object to the target. If the target property is not explicitly specified, the target would need to be determined implicitly by context. The origin can be used to identify the context from which the object originated.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	

Arrive	URI:	http://www.w3.org/ns/activitystreams#Arrive	• JSON-LD
			
			EXAMPLE 56
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Arrive",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"location": {
			"@type": "Place",
			"displayName": "Work"
			},
			"origin": {
			"@type": "Place",
			"displayName": "Home"
			}
			}

	Notes:	An IntransitiveActivity that indicates that the actor has arrived at the location. The origin can be used to identify the context from which the actor originated. The target typically has no defined meaning.	
	Extends:	IntransitiveActivity	
	Properties:	Inherits all properties fom IntransitiveActivity.	

Create	URI:	http://www.w3.org/ns/activitystreams#Create	• JSON-LD
			
			EXAMPLE 61
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Create",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type": "Note",
			"displayName": "A Simple Note",
			"content": "This is a simple note"
			}

	Notes:	**Indicates that the actor has created the object.** **Create is generally equivalent to Post with the exception that the Post Activity has a defined meaning for "target" while Create does not.**	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	

			}

Delete	URI:	http://www.w3.org/ns/activitystreams#Delete	• JSON-LD
			
			EXAMPLE 66
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Delete",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": "http://example.org/notes/1"
			}

	Notes:	Indicates that the actor has deleted the object. If specified, the origin indicates the context from which the object was deleted.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	

Favorite	URI:	http://www.w3.org/ns/activitystreams#Favorite	• JSON-LD
			
			EXAMPLE 71
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Favorite",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": "http://example.org/notes/1"
			}

	Notes:	**Indicates that the actor likes, recommends or endorses the object. The target and origin typically have no defined meaning.** **The Favorite and Like activity types *MAY* be used as equivalent synonyms if an implementation chooses.**	
	Extends:	Respond	
	Properties:	Inherits all properties from Respond.	
Follow	URI:	http://www.w3.org/ns/activitystreams#Follow	• JSON-LD EXAMPLE 76 { "@context":
	Notes:	Indicates that the actor is "following" the object. Following is defined in the sense typically used within Social systems in which the actor is interested in any	

		activity performed by or on the object. The target and origin typically have no defined meaning.	"http://www.w3.org/ns/activitystreams", "@type": "Follow", "actor": { "@type": "Person", "displayName": "Sally" }, "object": { "@type": "Person", "displayName": "John" } }
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	
Ignore	URI:	http://www.w3.org/ns/activitystreams#Ignore	• JSON-LD EXAMPLE 81 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Ignore", "actor": { "@type": "Person", "displayName": "Sally" }, "object": "http://example.org/notes/1" }
	Notes:	Indicates that the actor is ignoring the object.The target and origin typically have no defined meaning.	
	Extends:	Respond	
	Properties:	Inherits all properties from Respond.	
Join	URI:	http://www.w3.org/ns/activitystreams#Join	• JSON-LD EXAMPLE 86 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Join", "actor": { "@type": "Person", "displayName": "Sally" }, "object": { "@type": "Group", "displayName": "A Simple Group" } }
	Notes:	Indicates that the actor has joined the object. The target and origin typically have no defined meaning.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	

Leave	URI:	http://www.w3.org/ns/activitystreams#Leave	• JSON-LD
			
			EXAMPLE 91
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Leave",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type": "Place",
			"displayName": "Work"
			}
			}

	Notes:	Indicates that the actor has left the object. The target and origin typically have no meaning.	• JSON-LD
			
			EXAMPLE 96
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Leave",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type": "Group",
			"displayName": "A Simple Group"
			}
			}

	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	

Like	URI:	http://www.w3.org/ns/activitystreams#Like	• JSON-LD
			
			EXAMPLE 101
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Like",
			"actor": {
			"@type": "Person",

	Notes:	**Indicates that the actor likes, recommends or endorses the object. The target and origin typically have no defined meaning.** **The Favorite and Like activity types *MAY* be used as equivalent synonyms if an implementation chooses.**	
	Extends:	Respond	

	Properties:	Inherits all properties from Respond.	"displayName": "Sally" }, "object": "http://example.org/notes/1" }

Offer	URI:	http://www.w3.org/ns/activitystreams#Offer	• JSON-LD
			
			EXAMPLE 106
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Offer",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type":
			"urn:examples:types:ProductOffer",
			"displayName": "50% Off!"
			}
			}

	Notes:	Indicates that the actor is offering the object. If specified, the target indicates the entity to which the object is being offered.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	

Give	URI:	http://www.w3.org/ns/activitystreams#Give	• JSON-LD
			
			EXAMPLE 111
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Give",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type":
			"urn:example:types:Present",
			"displayName": "A Present"
			},
			"target": {
			"@type": "Person",

	Notes:	A specialization of Offer in which the actor is giving the object to the target. If the target is not specified it can be determined by context.	
	Extends:	Offer	
	Properties:	Inherits all properties from Offer.	

			"displayName": "John" } }

Invite	URI:	http://www.w3.org/ns/activitystreams#Invite	• JSON-LD
			
			EXAMPLE 116
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Invite",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type": "urn:example:types:Present",
			"displayName": "A Present"
			},
			"target": {
			"@type": "Person",
			"displayName": "John"
			}
			}

	Notes:	A specialization of Offer in which the actor is extending an invitation for the object to the target.	
	Extends:	Offer	
	Properties:	Inherits all properties from Offer.	

Post	URI:	http://www.w3.org/ns/activitystreams#Post	• JSON-LD
			
			EXAMPLE 121
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Post",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type": "Note",
			"displayName": "A Simple Note",
			"content": "This is a simple note"
			}
			}

	Notes:	**Indicates that the actor is posting the object. If specified, the target indicates to entity to which the object is being posted.** **Implementations can treat Post as being generally equivalent to Create** .	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	
Reject	URI:	http://www.w3.org/ns/activitystreams#Reject	• JSON-LD

	Notes:	Indicates that the actor is rejecting the object. The target and origin typically have no defined meaning.	EXAMPLE 126
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Reject",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type": "Invite",
			"actor": "acct:john@example.org",
			"object": {
			"@type": "Event",
			"displayName": "A Party!"
			}
			}
			}

	Extends:	Respond	
	Properties:	Inherits all properties from Respond.	

TentativeReject	URI:	http://www.w3.org/ns/activitystreams#TentativeReject	• JSON-LD
			
			EXAMPLE 131
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "TentativeReject",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": {
			"@type": "Invite",
			"actor": "acct:john@example.org",
			"object": {
			"@type": "Event",
			"displayName": "A Party!"
			}
			}
			}

	Notes:	A specialization of Reject in which the rejection is considered tentative.	
	Extends:	Reject	
	Properties:	Inherits all properties from Reject.	
Remove	URI:	http://www.w3.org/ns/activitystreams#Remove	• JSON-LD EXAMPLE 136 { "@context":
	Notes:	Indicates that the actor is removing the object. If specified, the origin indicates the context from which the object is being removed (but not destroyed, as in	

		the 'delete' verb).	"http://www.w3.org/ns/activity streams", "@type": "Remove", "actor": { "@type": "Person", "displayName": "Sally" }, "object": "http://example.org/notes/1", "target": { "@type": "Folder", "displayName": "Notes Folder" } }
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	• JSON-LD EXAMPLE 141 { "@context": "http://www.w3.org/ns/activity streams", "@type": "Remove", "actor": { "@type": "http://example.org/Role", "displayName": "The Moderator" }, "object": { "@type": "Person", "displayName": "Sally" }, "origin": { "@type": "Group", "displayName": "A Simple Group" } }
Review	URI:	http://www.w3.org/ns/activitystreams#Review	• JSON-LD EXAMPLE 146 { "@context": "http://www.w3.org/ns/activity streams", "@type": "Review",
	Notes:	Indicates that the actor has reviewed the object. The target typically has no defined meaning.	
	Extends:	Respond	
	Properties:	Inherits all properties from Respond.	

			"actor": "acct:sally@example.org", "object": "http://example.org/posts/1", "rating": 3.5, }
Save	URI:	http://www.w3.org/ns/activitystreams#Save	• JSON-LD EXAMPLE 151 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Save", "actor": { "@type": "Person", "displayName": "Sally" }, "object": "http://example.org/posts/1", "target": { "@type": "OrderedCollection", "displayName": "Sally's Reading List" } }
	Notes:	Indicates that the actor is saving the object. If specified, the target indicates the context into which the object is being saved. The origin typically has no defined meaning.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	
Share	URI:	http://www.w3.org/ns/activitystreams#Share	• JSON-LD EXAMPLE 156 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Share", "actor": "acct:sally@example.org", "object": "http://example.org/posts/1", "target": "acct:john@example.org"
	Notes:	Indicates that the actor is sharing the object. If specified, the target indicates to context or entity to which the object is being shared. The origin typically has no defined meaning.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	

			}

Undo	URI:	http://www.w3.org/ns/activitystreams#Undo	• JSON-LD
			
			EXAMPLE 161
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Undo",
			"actor": "acct:sally@example.org",
			"object": {
			"@type": "Share",
			"actor": "acct:sally@example.org",
			"object": "http://example.org/posts/1",
			"target": "acct:john@example.org"
			}
			}

	Notes:	Indicates that the actor is undoing the object. The target and origin typically have no defined meaning.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	

Update	URI:	http://www.w3.org/ns/activitystreams#Update	• JSON-LD
			
			EXAMPLE 166
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Update",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": "http://example.org/notes/1"
			}

	Notes:	Indicates that the actor has updated the object. The target and origin typically have no defined meaning.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	
Experience	URI:	http://www.w3.org/ns/activitystreams#Experience	• JSON-LD EXAMPLE 171 {
	Notes:	Indicates that the actor has experienced the object. The type of experience is not specified.	
	Extends:	Activity	
	Properti	Inherits all properties from Activity.	

	es:		"@context": "http://www.w3.org/ns/activitystreams", "@type": "Experience", "actor": { "@type": "Person", "displayName": "Sally" }, "object": { "@type": "Article", "displayName": "An article about Activity Streams" } }
View	URI:	http://www.w3.org/ns/activitystreams#View	• JSON-LD EXAMPLE 176 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "View", "actor": { "@type": "Person", "displayName": "Sally" }, "object": { "@type": "Article", "displayName": "An article about Activity Streams" } }
	Notes:	Indicates that the actor has viewed the object. Viewing is a specialization of Experience.	
	Extends:	Experience	
	Properties:	Inherits all properties from Experience.	
Watch	URI:	http://www.w3.org/ns/activitystreams#Watch	• JSON-LD EXAMPLE 181 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Watch", "actor": { "@type": "Person", "displayName": "Sally" }, "object":
	Notes:	Indicates that the actor has watched the object. Watching is a specialization of View.	
	Extends:	View	
	Properties:	Inherits all properties from View.	

			"http://example.org/dog.mkv" }

Listen	URI:	http://www.w3.org/ns/activitystreams#Listen	• JSON-LD
			
			EXAMPLE 186
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Listen",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": "http://example.org/music.mp3"
			}

	Notes:	Indicates that the actor has listened to the object. Viewing is a specialization of Experience.	
	Extends:	Experience	
	Properties:	Inherits all properties from Experience.	

Read	URI:	http://www.w3.org/ns/activitystreams#Read	• JSON-LD
			
			EXAMPLE 191
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Read",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"object": "http://example.org/posts/1"
			}

	Notes:	Indicates that the actor has read the object. Reading is a specialization of View.	
	Extends:	View	
	Properties:	Inherits all properties from View.	

Respond	URI:	http://www.w3.org/ns/activitystreams#Respond	• JSON-LD
			
			EXAMPLE 196
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Respond",
			"actor": {
			"@type": "Person",
			"displayName": "Sally"

	Notes:	Indicates that the actor has reponded to the object.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	

			}, "object": "http://example.org/posts/1", "result": { "@type": "Note", "content": "This is a good article", "inReplyTo": "http://example.org/posts/1" } }
Move	URI:	http://www.w3.org/ns/activitystreams#Move	• JSON-LD EXAMPLE 201 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Move", "actor": { "@type": "Person", "displayName": "Sally" }, "object": "http://example.org/posts/1", "target": { "@type": "Collection", "displayName": "List B" }, "origin": { "@type": "Collection", "displayName": "List A" } }
	Notes:	Indicates that the actor has moved object from origin to target. If the origin or target are not specified, either can be determined by context.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	
Travel	URI:	http://www.w3.org/ns/activitystreams#Travel	• JSON-LD EXAMPLE 206 { "@context":
	Notes:	Indicates that the actor is traveling to target from origin. Travel is an IntransitiveObject whose actor specifies the direct object. If the target or origin	

		are not specified, either can be determined by context.	"http://www.w3.org/ns/activitystreams", "@type": "Travel", "actor": { "@type": "Person", "displayName": "Sally" }, "target": { "@type": "Place", "displayName": "Home" }, "origin": { "@type": "Place", "displayName": "Work" } }
	Extends:	IntransitiveActivity	
	Properties:	Inherits all properties from IntransitiveActivity.	
Announce	URI:	http://www.w3.org/ns/activitystreams#Announce	• JSON-LD EXAMPLE 211 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Announce", "actor": { "@type": "Person", "@id": "acct:sally@example.org", "displayName": "Sally" }, "object": { "@type": "Arrive", "actor": "acct:sally@example.org", "location": { "@type": "Place", "displayName": "Work" } } }
	Notes:	Indicates that the actor is announcing the object to target. The origin typically has no defined meaning.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	
Block	URI:	http://www.w3.org/ns/activitystreams#Block	• JSON-LD EXAMPLE 216 { "@context": "http://www.w3.org/ns/activitystreams",
	Notes:	Indicates that the actor is blocking the object. Blocking is a stronger form of Ignore. The typical use is to support social systems that allow one user to block activities or content of other users. The target and origin typically have no defined meaning.	

	Extends:	Ignore	"@type": "Block", "actor": "acct:sally@example.org", "object": "acct:joe@example.org" }
	Properties:	Inherits all properties from Ignore.	
Flag	URI:	http://www.w3.org/ns/activitystreams#Flag	• JSON-LD EXAMPLE 221 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Flag", "actor": "acct:sally@example.org", "object": { "@type": "Note", "content": "An inappropriate note" } }
	Notes:	Indicates that the actor is "flagging" the object. Flagging is defined in the sense common to many social platforms as reporting content as being inappropriate for any number of reasons.	
	Extends:	Respond	
	Properties:	Inherits all properties from Respond.	
Dislike	URI:	http://www.w3.org/ns/activitystreams#Dislike	• JSON-LD EXAMPLE 226 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Dislike", "actor": "acct:sally@example.org", "object": "http://example.org/posts/1" }
	Notes:	Indicates that the actor dislikes the object.	
	Extends:	Respond	
	Properties:	Inherits all properties from Respond.	
Confirm	URI:	http://www.w3.org/ns/activitystreams#Confirm	• JSON-LD EXAMPLE 231 { "@context": "http://www.w3.org/ns/activitystreams",
	Notes:	Indicates that the actor is confirming the object.	
	Extends:	Respond	
	Properties:	Inherits all properties from Respond.	

			"@type": "Confirm", "actor": "acct:sally@example.org", "object": { "@type": "http://example.org/Reservation", "actor": "acct:sally@example.org", "object": "http://example.org/events/1" } }
Assign	URI:	http://www.w3.org/ns/activitystreams#Assign	• JSON-LD • EXAMPLE 236 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Assign", "actor": "acct:sally@example.org", "object": { "@type": "http://example.org/Role", "displayName": "Moderator" }, "target": "acct:joe@example.org" }
	Notes:	Indicates that the actor is assigning the object to the target.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	
Complete	URI:	http://www.w3.org/ns/activitystreams#Complete	• JSON-LD EXAMPLE 241 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Complete", "actor": "acct:sally@example.org", "object": "http://example.org/tasks/1" }
	Notes:	Indicates that the actor has completed the object.	
	Extends:	Activity	
	Properties:	Inherits all properties from Activity.	

3.3.7. Actor Types

All Actor Types inherit the properties of the base Actor class. Actor's are objects that are capable of performing activities. The value of the actor property MUST be a type of Actor.

The core Actor Types include: `Application` | `Group` | `Person` | `Process` | `Service`

3.3.8. Object Types

All Object Types inherit the properties of the base Object class. Some specific Object Types are subclasses or specializations of more generalized Object Types (for instance, the `Person` Object Type is a more specific form of the `Actor` class).

The Object Types include: `Album` | `Article` | `Audio` | `Connection` | `Content` | `Document` | `Event` | `Folder` | `Image` | `Mention` | `Note` | `Page` | `Place` | `Question` | `Story` | `Video`

Connection	URI:	http://www.w3.org/ns/activitystreams#Connection	• JSON-LD
			•
			•
			•
			•
			
			EXAMPLE 271
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Connection",
			"a": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"relationship": "http://purl.org/vocab/relationship/closeFriendOf",
			"b": {
			"@type": "Person",
			"displayName": "John"
			}
			}

	Notes:	Describes a connection between two individuals. The a and b properties are used to identify the connected individuals.	
	Extends:	Object	
	Properties:	a | b | relationship Inherits all properties from Object.	
Content	URI:	http://www.w3.org/ns/activitystreams#Content	• JSON-LD EXAMPLE 276 { "@context":
	Notes:	Describes an entity representing any form of content. Examples include documents, images, etc. Content objects typically are not able to perform activities on their own, yet rather are usually the object or	

		target of activities.	"http://www.w3.org/ns/activitystreams" , "@type": "Content", "displayName": "Some generic content", "content": " This can be any kind of content ", "height": 100, "width": 100 }
	Extends:	Object	
	Properties:	duration | height | width Inherits all properties from Object.	
Article	URI:	http://www.w3.org/ns/activitystreams#Article	• JSON-LD EXAMPLE 281 { "@context": "http://www.w3.org/ns/activitystreams" , "@type": "Article", "displayName": "A Blog Post", "content": " ... a long blog post ", "attributedTo": "acct:sally@example.org" }
	Notes:	Represents any kind of multi-paragraph written work.	
	Extends:	Content	
	Properties:	Inherits all properties from Content.	
Album	URI:	http://www.w3.org/ns/activitystreams#Album	• JSON-LD EXAMPLE 286 { "@context": "http://www.w3.org/ns/activitystreams" , "@type": "Album", "displayName": "A Photo Album", "items": [{ "@type": "Image", "displayName": "My Dog", "url": { "@type": "Link", "href": "http://example.org/dog.jpeg", "mediaType": "image/jpeg" } }, { "@type": "Image", "displayName": "My Cat", "url": { "@type": "Link", "href": "http://example.org/cat.jpeg", "mediaType": "image/jpeg" }
	Notes:	A type of Collection typically used to organize Image, Video or Audio objects.	
	Extends:	Collection	
	Properties:	Inherits all properties from Collection.	

<table>
<tr><td></td><td></td><td></td><td>
<pre>
    }
  ]
}
</pre>
JSON-LD
EXAMPLE 291
<pre>
{
  "@context":
"http://www.w3.org/ns/activitystreams"
,
  "@type": "Album",
  "displayName": "A Music Playlist",
  "orderedItems": [
    {
      "@type": "Audio",
      "displayName": "Song 1",
      "url": {
        "@type": "Link",
        "href":
"http://example.org/song1.mp3",
        "mediaType": "audio/mp3"
      }
    },
    {
      "@type": "Audio",
      "displayName": "Song 2",
      "url": {
        "@type": "Link",
        "href":
"http://example.org/song2.mpg",
        "mediaType": "audio/mp3"
      }
    }
  ]
}
</pre>
</td></tr>
<tr><td rowspan="4">Folder</td><td>URI:</td><td>http://www.w3.org/ns/activitystreams#Folder</td><td rowspan="4">
JSON-LD
EXAMPLE 296
<pre>
{
  "@context":
"http://www.w3.org/ns/activitystreams"
,
  "@type": "Folder",
  "displayName": "Some Documents",
  "items": [
    {
      "@type": "Document",
      "displayName": "4Q Sales
Forecast",
      "url": "http://example.org/4q-sales-
forecast.pdf"
    }
  ]
}
</pre>
</td></tr>
<tr><td>Notes:</td><td>A type of Collection typically used to organize objects such as Documents.</td></tr>
<tr><td>Extends:</td><td>Collection</td></tr>
<tr><td>Properties:</td><td>Inherits all properties from Collection.</td></tr>
</table>

Story	URI:	http://www.w3.org/ns/activitystreams#Story	• JSON-LD
			
			EXAMPLE 301
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Story",
			"displayName": "My Vacation",
			"startIndex": 5,
			"orderedItems": [
			{
			"@type": "Image",
			"displayName": "Visiting the Vatican",
			"url": "http://example.org/photo5.jpeg",
			"location": {
			"@type": "Place",
			"displayName": "The Vatican"
			}
			},
			{
			"@type": "Image",
			"displayName": "Visiting the Eiffel Tower",
			"url": "http://example.org/photo6.jpeg",
			"location": {
			"@type": "Place",
			"displayName": "The Eiffel Tower"
			}
			}
			]
			}

	Notes:	A type of Ordered Collection usually containing Content Items organized to "tell a story".	
	Extends:	OrderedCollection	
	Properties:	Inherits all properties from OrderedCollection.	

Document	URI:	http://www.w3.org/ns/activitystreams#Document	• JSON-LD
			
			EXAMPLE 306
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Document",
			"displayName": "4Q Sales Forecast",
			"url": "http://example.org/4q-sales-forecast.pdf"
			}

	Notes:	Represents a document of any kind.	
	Extends:	Content	
	Properties:	Inherits all properties from Content.	

Audio	URI:	http://www.w3.org/ns/activitystreams#Audio	• JSON-LD
			
			EXAMPLE 311
			
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Audio",
			"displayName": "A Simple Podcast",
			"url": {
			"@type": "Link",
			"href": "http://example.org/podcast.mp3",
			"mediaType": "audio/mp3"
			}
			}

	Notes:	Represents an audio document of any kind.	
	Extends:	Document	
	Properties:	Inherits all properties from Document.	

Image	URI:	http://www.w3.org/ns/activitystreams#Image	• JSON-LD
			
			EXAMPLE 316
			
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Image",
			"displayName": "A Simple Image",
			"url": [
			{
			"@type": "Link",
			"href": "http://example.org/image.jpeg",
			"mediaType": "image/jpeg"
			},
			{
			"@type": "Link",
			"href": "http://example.org/image.png",
			"mediaType": "image/png"
			}
			]
			}

	Notes:	An image document of any kind	
	Extends:	Document	
	Properties:	Inherits all properties from Document.	
Video	URI:	http://www.w3.org/ns/activitystreams#Video	• JSON-LD EXAMPLE 321 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Video", "displayName": "A Simple Video",
	Notes:		
	Extends:	Document	
	Properties:	Inherits all properties from Document.	

			"url": "http://example.org/video.mkv", "duration": "PT2H" }

Note	URI:	http://www.w3.org/ns/activitystreams#Note	• JSON-LD
			
			EXAMPLE 326
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"displayName": "A Short Note",
			"content": "This is a short note"
			}

	Notes:	Represents a short written work typically less than a single paragraph in length.	
	Extends:	Content	
	Properties:	Inherits all properties from Content.	

Page	URI:	http://www.w3.org/ns/activitystreams#Page	• JSON-LD
			
			EXAMPLE 331
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Page",
			"displayName": "A Webpage",
			"url": "http://example.org/page.html"
			}

	Notes:	Represents a Web Page.	
	Extends:	Document	
	Properties:	Inherits all properties from Document.	

Question	URI:	http://www.w3.org/ns/activitystreams#Question	• JSON-LD
			
			EXAMPLE 336
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Question",
			"displayName": "What is the answer?",
			"oneOf": [
			{
			"@type": "Note",
			"displayName": "Option A"
			},
			{
			"@type": "Note",
			"displayName": "Option B"
			}
			]
			}

	Notes:	Represents a question being asked. Question objects are unique in that they are an extension of both Content and IntransitiveActivity. That is, the Question object is an Activity but the direct object is the question itself.	
	Extends:	Content AND IntransitiveActivity.	
	Properties:	oneOf | anyOf Inherits all properties from Content and IntransitiveActivity.	
Event	URI:	http://www.w3.org/ns/activitystreams#Event	• JSON-LD
	Notes:	Represents any kind of event.	

	Extends:	Object	

	Properties:	Inherits all properties from Object.	EXAMPLE 341
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Event",
			"displayName": "A Party!",
			"startTime": "2014-12-31T23:00:00-08:00",
			"endTime": "2015-01-01T06:00:00-08:00"
			}

Place	URI:	http://www.w3.org/ns/activitystreams#Place	• JSON-LD
			
			EXAMPLE 346
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Place",
			"displayName": "Work"
			}
			
			• JSON-LD
			
			EXAMPLE 351
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Place",
			"displayName": "Fresno Area",
			"latitude": 36.75,
			"longitude": 119.7667,
			"radius": 15,
			"units": "miles"
			}

	Notes:	Represents a logical or physical location.	
	Extends:	Object	
	Properties:	accuracy | altitude | latitude | longitude | radius | units Inherits all properties from Obct.	
Mention	URI:	http://www.w3.org/ns/activitystreams#Mention	• JSON-LD EXAMPLE 356 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Mention", "href": "http://example.org/joe",
	Notes:	A specialized Link that represents an @mention.	
	Extends:	Link	
	Properties:	Inherits all properties from Link.	

			"displayName": "Joe" }

3.3.9. Representing Connections Between Entities

The Connection object is used to represent relationships between individuals. It can be used, for instance, to describe that one person is a friend of another, or that one person is a member of a particular organization. The intent of modeling Connection in this way is to allow descriptions of activities that operate on the relationships in general, and to allow representation of Collections of relationships.

For instance, many social sytems have a notion of a "friends list". These are the collection of individuals that are directly connected within a person's social graph. Suppose we have a user, Sally, with direct connections to users Joe and Jane. Sally considers Joe to be a close friend while Jane is just an acquaintance.

Using the Connection object, we can model these relationships as:

EXAMPLE 361

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Collection",
  "items": [
    {
      "@type": "Connection",
      "a": {
        "@type": "Person",
        "displayName": "Sally"
      },
      "relationship": "http://purl.org/vocab/relationship/closeFriendOf",
      "b": {
        "@type": "Person",
        "displayName": "Joe"
      }
    },
    {
      "@type": "Connection",
      "a": {
        "@type": "Person",
        "displayName": "Sally"
      },
      "relationship":
"http://purl.org/vocab/relationship/acquaintanceOf",
      "b": {
        "@type": "Person",
        "displayName": "Jane"
      }
    }
  ]
}
```

The relationship property specifies the kind of connection that exists between the two individuals identified by the a and b properties. Used together, these three properties form what is commonly

known as a "reified statement" where a identifies the subject, relationship identifies the predicate, and b identifies the object.

While use of reified statements can be problematic and confusing in certain situations, their use within the Activity Streams vocabulary to describe connections provides a straightforward mechanism of describing changes to an individual's social graph. For instance, to indicate that Sally has created a new connection to user Matt, an implementer can use the Connection object together with the Create activity:

EXAMPLE 362

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Create",
  "actor": "acct:sally@example.org",
  "object": {
    "@type": "Connection",
    "a": "acct:sally@example.org",
    "relationship": "http://purl.org/vocab/relationship/closeFriendOf",
    "b": "acct:matt@example.org",
    "startTime": "2015-04-21T12:34:56"
  }
}
```

Additionally, modeling the relationship in this way allows implementers to articulate additional properties of the relationship itself. For instance, the date and time at which the relationship began or ended.

The Activity Streams vocabulary does not define normative values for use with the relationship property. It is expected that implementations will make use of several existing vocabularies that have been developed for the purpose of describing relationships. Examples of such vocabularies include the "Friend of a Friend" and "Relationship" vocabularies.

3.3.10. Representing Places

The Place object is used to represent both physical and logical locations. While numerous existing vocabularies exist for describing locations in a variety of ways, inconsistencies and incompatibilities between those vocabularies make it difficult to achieve appropriate interoperability between implementations. The Place object is included within the Activity vocabulary to provide a minimal, interoperable starting point for describing locations consistently across Activity Streams 2.0 implementations.

The Place object is intentionally flexible. It can, for instance, be used to identify a location simply by name:

EXAMPLE 363

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Place",
  "displayName": "San Francisco, CA"
}
```

Or, by `longitude` and `latitude`:

EXAMPLE 364

```
{
  "@context": "http://www.w3.org/ns/activitystreams",
  "@type": "Place",
  "displayName": "San Francisco, CA",
  "longitude": "122.4167",
  "latitude": "37.7833"
}
```

The Place object can also describe an area around a given point using the radius property, the altitude of the location, and a degree of accuracy.

While publishers are not required to use these specific properties and MAY make use of other mechanisms for describing locations, consuming implementations that support the Place object MUST support the use of these properties.

3.3.11. Properties

Base URI: http://www.w3.org/ns/activitystreams#.

The common properties include: actor | attachment | attributedTo | bcc | bto | cc | context | current | first | generator | icon | image | inReplyTo | last | location | items | oneOf | anyOf | origin | next | object | prev | preview | result | replies | scope | self | tag | target | to | url | accuracy | alias | altitude | content | displayName | duration | height | href | hreflang | itemsPerPage | latitude | longitude | mediaType | priority | endTime | published | startTime | radius | rating | rel | startIndex | summary | title | totalItems | units | updated | width | a | b | relationship

The "Domain" indicates the type of Object the property term applies to. The "Range" indicates the type of value the property term can have. Certain properties are marked as a "Subproperty Of" another term, meaning that the term is a specialization of the referenced term. For instance, actor is a subproperty of attributedTo. Properties marked as being "Functional" can have only one value. Items not marked as "Functional" can have multiple values.

<table>
<tr><th>Term</th><th colspan="2">Description</th><th>Example</th></tr>
<tr><td rowspan="5">actor</td><td>URI:</td><td>http://www.w3.org/ns/activitystreams#actor</td><td rowspan="5">• JSON-LD

EXAMPLE 365
{
"@context": "http://www.w3.org/ns/activitystreams",
"@type": "Share",
"actor": "acct:sally@example.org",
"object": "http://example.org/foo"
}

• JSON-LD</td></tr>
<tr><td>Notes:</td><td>Describes one or more entities that either performed or are expected to perform the activity. Any single activity can have multiple Actors. The Actor MAY be specified using an indirect Link.</td></tr>
<tr><td>Domain:</td><td>Activity</td></tr>
<tr><td>Range:</td><td>Actor | Link</td></tr>
<tr><td>Subproperty Of:</td><td>attributedTo</td></tr>
</table>

			EXAMPLE 370
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Share",
			"actor": {
			"@type": "Person",
			"@id": "acct:sally@example.org",
			"displayName": "Sally"
			},
			"object": "http://example.org/foo"
			}
			• JSON-LD
			• EXAMPLE 375
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Share",
			"actor": [
			"acct:joe@example.org",
			{
			"@type": "Person",
			"@id": "acct:sally@example.org",
			"displayName": "Sally"
			}
			],
			"object": "http://example.org/foo"
			}

attachment	URI:	http://www.w3.org/ns/activitystreams#attachment	• JSON-LD
			EXAMPLE 380
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"displayName": "A Simple Note",
			"attachment": [
			{
			"@type": "Image",
			"content": "A simple Image",
			"url":
			"http://example.org/cat.jpeg"
			}
			]
			}

	Notes:	Identifies an entity that is directly or indirected attached to this object	
	Domain:	Object	
	Range:	Object | Link	
attributed	URI:	http://www.w3.org/ns/activitystreams#	• JSON-LD

To		attributedTo	EXAMPLE 385
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Image",
			"displayName": "A Simple Image",
			"url": "http://example.org/cat.jpeg",
			"attributedTo": [
			{
			"@type": "Person",
			"displayName": "Sally"
			}
			]
			}
			• JSON-LD
			EXAMPLE 390
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Image",
			"displayName": "A Simple Image",
			"url": "http://example.org/cat.jpeg",
			"attributedTo": [
			"acct:joe@example.org",
			{
			"@type": "Person",
			"displayName": "Sally"
			}
			]
			}

	Notes:	Identifies one or more entities to which this object is attributed. The attributed entities might not be Actors. For instance, an object might be attributed to the completion of another activity.	
	Domain:	Link | Object	
	Range:	Link | Object	

bcc	URI:	http://www.w3.org/ns/activitystreams#bcc	• JSON-LD
			EXAMPLE 395
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Share",
			"actor": "acct:sally@example.org",
			"object": "http://example.org/posts/1",
			"target": "acct:john@example.org",
			"bcc": ["acct:joe@example.org"]
			}

	Notes:	Identifies one or more Actors that are part of the private secondary audience of this Activity.	
	Domain:	Activity	
	Range:	Actor | Link	
bto	URI:	http://www.w3.org/ns/activitystreams#	• JSON-LD

		bto	EXAMPLE 400
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Share",
			"actor": "acct:sally@example.org",
			"object": "http://example.org/posts/1",
			"target": "acct:john@example.org",
			"bto": ["acct:joe@example.org"]
			}

	Notes:	Identifies an Actor that is part of the private primary audience of this Activity.	
	Domain:	Activity	
	Range:	Actor | Link	

cc	URI:	http://www.w3.org/ns/activitystreams#cc	• JSON-LD
			
			EXAMPLE 405
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Share",
			"actor": "acct:sally@example.org",
			"object": "http://example.org/posts/1",
			"target": "acct:john@example.org",
			"cc": ["acct:joe@example.org"]
			}

	Notes:	Identifies an Actor that is part of the public secondary audience of this Activity.	
	Domain:	Activity	
	Range:	Actor | Link	

context	URI:	http://www.w3.org/ns/activitystreams#context	• JSON-LD
			
			EXAMPLE 410
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"items": [
			{
			"@type": "Share",
			"actor": "acct:sally@example.org",
			"object": "http://example.org/posts/1",
			"target": "acct:john@example.org",
			"context": "http://example.org/contexts/1"
			},
			{
			"@type": "Like",
			"actor": "acct:joe@example.org",
			"object": "http://example.org/posts/2",
			"context": "http://example.org/contexts/1"
			}

	Notes:	Identifies the context within which the object exists or an activity was performed. The notion of "context" used is intentionally vague. The intended function is to serve as a means of grouping objects and activities that share a common originating context or purpose. An example could be all activities relating to a common project or event.	
	Domain:	Object	
	Range:	Object | Link	

			] }

current	URI:	http://www.w3.org/ns/activitystreams#current	• JSON-LD
			
			EXAMPLE 415
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 5,
			"itemsPerPage": 3,
			"current": "http://example.org/collection",
			"items": [
			"http://example.org/posts/1",
			"http://example.org/posts/2",
			"http://example.org/posts/3",
			]
			}
			
			• JSON-LD
			
			EXAMPLE 420
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 5,
			"itemsPerPage": 3,
			"current": {
			"@type": "Link",
			"displayName": "Most Recent Items",
			"href": "http://example.org/collection"
			},
			"items": [
			"http://example.org/posts/1",
			"http://example.org/posts/2",
			"http://example.org/posts/3",
			]
			}

	Notes:	In a paged Collection, indicates the page that contains the most recently updated member items.	
	Domain:	Collection	
	Range:	Collection | Link	
	Functional:	True	
first	URI:	http://www.w3.org/ns/activitystreams#first	• JSON-LD EXAMPLE 425 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Collection", "totalItems": 5,
	Notes:	In a paged Collection, indicates the furthest preceeding page of items in the collection.	
	Domain:	Collection	
	Range:	Collection | Link	
	Functional:	True	

			"itemsPerPage": 3,
			"first":
			"http://example.org/collection",
			"items": [
			"http://example.org/posts/1",
			"http://example.org/posts/2",
			"http://example.org/posts/3",
			]
			}
			
			• JSON-LD
			
			EXAMPLE 430
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 5,
			"itemsPerPage": 3,
			"first": {
			"@type": "Link",
			"displayName": "First Page",
			"href":
			"http://example.org/collection"
			},
			"items": [
			"http://example.org/posts/1",
			"http://example.org/posts/2",
			"http://example.org/posts/3",
			]
			}

generator	URI:	http://www.w3.org/ns/activitystreams#generator	• JSON-LD
			
			EXAMPLE 435
			
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"content": "A simple note",
			"generator": {
			"@type": "Application",
			"displayName": "My Note Application"
			}
			}

	Notes:	Identifies the entity (e.g. an application) that generated the object.	
	Domain:	Object	
	Range:	Object | Link	
icon	URI:	http://www.w3.org/ns/activitystreams#icon	• JSON-LD • EXAMPLE 440 {
	Notes:	Indicates an entity that describes an icon for this object. The image should have an aspect ratio of one	

		(horizontal) to one (vertical) and should be suitable for presentation at a small size.	"@context": "http://www.w3.org/ns/activitystreams", "@type": "Note", "content": "A Simple note", "icon": { "@type": "Image", "displayName": "Note", "url": "http://example.org/note.png", "width": 16, "height": 16 } }
	Domain:	Object	
	Range:	Image | Link	• JSON-LD EXAMPLE 445 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Note", "content": "A Simple note", "icon": [{ "@type": "Image", "displayName": "Note (16x16)", "url": "http://example.org/note1.png", "width": 16, "height": 16 }, { "@type": "Image", "displayName": "Note (32x32)", "url": "http://example.org/note2.png", "width": 32, "height": 32 }] }
image	URI:	http://www.w3.org/ns/activitystreams#image	• JSON-LD EXAMPLE 450 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Note", "content": "A Simple note", "image": { "@type": "Image",
	Notes:	Indicates an entity that describes an image for this object. Unlike the icon property, there are no aspect ratio or display size limitations assumed.	
	Domain:	Object	
	Range:	Image | Link	

<table>
<tr><td></td><td></td><td></td><td>"displayName": "A Cat",
"url": "http://example.org/cat.png"
}
}

• JSON-LD

EXAMPLE 455
{
"@context":
"http://www.w3.org/ns/activitystreams",
"@type": "Note",
"content": "A Simple note",
"image": [
{
"@type": "Image",
"displayName": "Cat 1",
"url":
"http://example.org/cat1.png"
},
{
"@type": "Image",
"displayName": "Cat 2",
"url":
"http://example.org/cat2.png"
}
]
}</td></tr>
<tr><td rowspan="4">inReplyto</td><td>URI:</td><td>http://www.w3.org/ns/activitystreams#inReplyTo</td><td rowspan="4">• JSON-LD

EXAMPLE 460
{
"@context":
"http://www.w3.org/ns/activitystreams",
"@type": "Note",
"content": "A simple note",
"inReplyTo": {
"@type": "Note",
"content": "Another note"
}
}

• JSON-LD

EXAMPLE 465
{
"@context":
"http://www.w3.org/ns/activitystreams",
"@type": "Note",</td></tr>
<tr><td>Notes:</td><td>Indicates one or more entities for which this object is considered a response.</td></tr>
<tr><td>Domain:</td><td>Object</td></tr>
<tr><td>Range:</td><td>Object | Link</td></tr>
</table>

			"content": "A simple note", "inReplyTo": "http://example.org/posts/1" }

last	URI:	http://www.w3.org/ns/activitystreams#last	• JSON-LD
			
			EXAMPLE 470
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 5,
			"itemsPerPage": 3,
			"last":
			"http://example.org/collection",
			"items": [
			"http://example.org/posts/1",
			"http://example.org/posts/2",
			"http://example.org/posts/3",
			]
			}
			
			• JSON-LD
			•
			•
			•
			•
			
			EXAMPLE 475
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 5,
			"itemsPerPage": 3,
			"last": {
			"@type": "Link",
			"displayName": "Last Page",
			"href":
			"http://example.org/collection"
			},
			"items": [
			"http://example.org/posts/1",
			"http://example.org/posts/2",
			"http://example.org/posts/3",
			]
			}

	Notes:	In a paged Collection, indicates the furthest proceeding page of the collection.	
	Domain:	Collection	
	Range:	Collection | Link	
	Functional:	True	
location	URI:	http://www.w3.org/ns/activitystreams#location	• JSON-LD EXAMPLE 480
	Notes:	Indicates one or more physical or logical locations associated with	

		the object.	{ "@context": "http://www.w3.org/ns/activitystreams", "@type": "Person", "displayName": "Sally", "location": { "@type": "Place", "longitude": 12.34, "latitude": 56.78, "altitude": 90, "units": "m" } }
	Domain:	Object	
	Range:	Object | Link	
items	URI:	http://www.w3.org/ns/activitystreams#items	• JSON-LD EXAMPLE 485 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Collection", "totalItems": 2, "itemsPerPage": 2, "items": [{ "@type": "Note", "displayName": "A Simple Note" }, { "@type": "Note", "displayName": "Another Simple Note" }] } • JSON-LD EXAMPLE 490 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "OrderedCollection", "totalItems": 2, "itemsPerPage": 2, "startIndex": 0, "orderedItems": [{ "@type": "Note", "displayName": "A Simple Note"
	Notes:	Identifies the items contained in a collection. The items might be ordered or unordered.	
	Domain:	Collection	
	Range:	Object | Link | Ordered List of [Object | Link]	

			}, { "@type": "Note", "displayName": "Another Simple Note" }] }
oneOf	URI:	http://www.w3.org/ns/activitystreams#oneOf	• JSON-LD EXAMPLE 495 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Question", "displayName": "What is the answer?", "oneOf": [{ "@type": "Note", "displayName": "Option A" }, { "@type": "Note", "displayName": "Option B" }] }
	Notes:	Identifies an exclusive option for a Question. Use of oneOf implies that the Question can have only a single answer. To indicate that a Question can have multiple answers, use anyOf.	
	Domain:	Question	
	Range:	Object | Link	
anyOf	URI:	http://www.w3.org/ns/activitystreams#anyOf	• JSON-LD EXAMPLE 500 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Question", "displayName": "What is the answer?", "anyOf": [{ "@type": "Note", "displayName": "Option A", }, { "@type": "Note", "displayName": "Option B" }] }
	Notes:	Identifies an inclusive option for a Question. Use of anyOf implies that the Question can have multiple answers. To indicate that a Question can have only one answer, use oneOf.	
	Domain:	Question	
	Range:	Object | Link	
origin	URI:	http://www.w3.org/ns/activitystreams#origin	• JSON-LD
	Notes:	Describes an indirect object of the	

		activity *from* which the activity is directed. The precise meaning of the origin is the object of the English preposition "from". For instance, in the activity "John moved an item to List A from List B", the origin of the activity is "List B".	EXAMPLE 505
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Move",
			"actor": "acct:sally@example.org",
			"object": "http://example.org/posts/1",
			"target": {
			"@type": "Collection",
			"displayName": "List B"
			},
			"origin": {
			"@type": "Collection",
			"displayName": "List A"
			}
			}

	Domain:	Activity	
	Range:	Object | Link	

next	URI:	http://www.w3.org/ns/activitystreams#next	• JSON-LD
			
			EXAMPLE 510
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 5,
			"itemsPerPage": 3,
			"next": "http://example.org/collection",
			"items": [
			"http://example.org/posts/1",
			"http://example.org/posts/2",
			"http://example.org/posts/3",
			]
			}
			
			• JSON-LD
			
			EXAMPLE 515
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 5,
			"itemsPerPage": 3,
			"next": {
			"@type": "Link",
			"displayName": "Next Page",
			"href": "http://example.org/collection"
			},
			"items": [
			"http://example.org/posts/1",

	Notes:	In a paged Collection, indicates the next page of items.	
	Domain:	Collection	
	Range:	Object | Link	
	Functional:	True	

			"http://example.org/posts/2", "http://example.org/posts/3",] }

object	URI:	http://www.w3.org/ns/activitystreams#object	• JSON-LD
			
			EXAMPLE 520
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Like",
			"actor": "acct:sally@example.org",
			"object":
			"http://example.org/posts/1"
			}
			
			• JSON-LD
			
			EXAMPLE 525
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Like",
			"actor": "acct:sally@example.org",
			"object": {
			"@type": "Note",
			"content": "A simple note"
			}
			}
			
			• JSON-LD
			
			EXAMPLE 530
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Like",
			"actor": "acct:sally@example.org",
			"object": [
			"http://example.org/posts/1",
			{
			"@type": "Note",
			"content": "A simple note"
			}
			]
			}

	Notes:	Describes the direct object of the activity. For instance, in the activity "John saved a movie to his wishlist", the object of the activity is the movie saved.	
	Domain:	Activity	
	Range:	Object | Link	
prev	URI:	http://www.w3.org/ns/activitystreams#prev	• JSON-LD EXAMPLE 535
	Notes:	In a paged Collection, identifies the previous page of items.	

	Domain:	Collection	{ "@context": "http://www.w3.org/ns/activitystreams", "@type": "Collection", "totalItems": 5, "itemsPerPage": 3, "prev": "http://example.org/collection", "items": ["http://example.org/posts/1", "http://example.org/posts/2", "http://example.org/posts/3",] }
	Range:	Collection | Link	
	Functional:	True	• JSON-LD EXAMPLE 540 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Collection", "totalItems": 5, "itemsPerPage": 3, "prev": { "@type": "Link", "displayName": "Previous Page", "href": "http://example.org/collection" }, "items": ["http://example.org/posts/1", "http://example.org/posts/2", "http://example.org/posts/3",] }
preview	URI:	http://www.w3.org/ns/activitystreams#preview	• JSON-LD EXAMPLE 545 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Video", "displayName": "Cool New Movie", "duration": "PT2H30M", "preview": { "@type": "Link", "displayName": "Trailer", "href": "http://example.org/trailer.mkv", "mediaType": "video/mkv", "duration": "PT1M"
	Notes:	Identifies an entity that provides a preview of this object.	
	Domain:	Link | Object	
	Range:	Link | Object	

			} }
result	URI:	http://www.w3.org/ns/activitystreams#result	• JSON-LD EXAMPLE 550 { "@context": "http://www.w3.org/ns/activitystreams", "@type": ["Activity", "urn:example:verbs:Check"], "actor": "acct:sally@example.org", "object": "http://example.org/flights/1", "result": { "@type": "urn:example:types:flightstatus", "displayName": "On Time" } }
	Notes:	Describes the result of the activity. For instance, if a particular action results in the creation of a new resource, the result property can be used to describe that new resource.	
	Domain:	Activity	
	Range:	Object | Link	
replies	URI:	http://www.w3.org/ns/activitystreams#replies	• JSON-LD EXAMPLE 555 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Note", "@id": "urn:example:notes:1", "content": "A simple note", "replies": { "@type": "Collection", "totalItems": 1, "itemsPerPage": 1, "items": [{ "@type": "Note", "content": "A response to the note", "inReplyTo": "urn:example:notes:1" }] } }
	Notes:	Identifies a Collection containing objects considered to be responses to this object.	
	Domain:	Object	
	Range:	Collection	
	Functional:	True	
scope	URI:	http://www.w3.org/ns/activitystreams#scope	• JSON-LD EXAMPLE 560 { "@context": "http://www.w3.org/ns/activitystreams",
	Notes:	Identifies one or more entities that represent the total population of entities for which the object can considered to be relevant.	
	Domain:	Object	

	Range:	Object | Link	"@type": "Note",
			"content": "A simple note",
			"scope": {
			"@type":
			"http://example.org/Organization",
			"displayName": "My Organization"
			}
			}

self	URI:	http://www.w3.org/ns/activitystreams#self	• JSON-LD
			
			EXAMPLE 565
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 5,
			"itemsPerPage": 3,
			"self":
			"http://example.org/collection",
			"items": [
			"http://example.org/posts/1",
			"http://example.org/posts/2",
			"http://example.org/posts/3",
			]
			}
			
			• JSON-LD
			
			EXAMPLE 570
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 5,
			"itemsPerPage": 3,
			"self": {
			"@type": "Link",
			"displayName": "This Page",
			"href":
			"http://example.org/collection"
			},
			"items": [
			"http://example.org/posts/1",
			"http://example.org/posts/2",
			"http://example.org/posts/3",
			]
			}

	Notes:	In a paged Collection, identifies this page of entries.	
	Domain:	Collection	
	Range:	Collection | Link	
	Functional:	True	
tag	URI:	http://www.w3.org/ns/activitystreams#tag	• JSON-LD EXAMPLE 575
	Notes:	One or more "tags" that have been associated with an objects. A tag	

		can be any kind of Object. The key difference between attachment and tag is that the former implies association by inclusion, while the latter implies associated by reference.	{ "@context": "http://www.w3.org/ns/activitystreams", "@type": "Image", "displayName": "Picture of Sally", "url": "http://example.org/sally.jpg", "tag": [{ "@type": "Person", "@id": "acct:sally@example.org" }] }
	Domain:	Object	
	Range:	Object | Link	
target	URI:	http://www.w3.org/ns/activitystreams#target	• JSON-LD EXAMPLE 580 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Share", "actor": "acct:sally@example.org", "object": "http://example.org/posts/1", "target": "acct:john@example.org" } • JSON-LD EXAMPLE 585 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Share", "actor": "acct:sally@example.org", "object": "http://example.org/posts/1", "target": { "@type": "Person", "displayName": "John" } }
	Notes:	Describes the indirect object, or target, of the activity. The precise meaning of the target is largely dependent on the type of action being described but will often be the object of the English preposition "to". For instance, in the activity "John saved a movie to his wishlist", the target of the activity is John's wishlist. An activity can have more than one target.	
	Domain:	Activity	
	Range:	Object | Link	
to	URI:	http://www.w3.org/ns/activitystreams#to	• JSON-LD EXAMPLE 590 { "@context": "http://www.w3.org/ns/activitystreams",
	Notes:	Identifies an entity considered to be part of the public primary audience of an Activity	
	Domain:	Activity	
	Range:	Object | Link	

			"@type": "Share", "actor": "acct:sally@example.org", "object": "http://example.org/posts/1", "target": "acct:john@example.org", "to": ["acct:joe@example.org"] }

url	URI:	http://www.w3.org/ns/activitystreams#url	• JSON-LD
			
			EXAMPLE 595
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Document",
			"displayName": "4Q Sales Forecast",
			"url": "http://example.org/4q-sales-forecast.pdf"
			}
			
			• JSON-LD
			
			EXAMPLE 600
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Document",
			"displayName": "4Q Sales Forecast",
			"url": {
			"@type": "Link",
			"href": "http://example.org/4q-sales-forecast.pdf"
			}
			}
			
			• JSON-LD
			
			EXAMPLE 605
			{
			"@context":
			"http://www.w3.org/ns/activitystreams",
			"@type": "Document",
			"displayName": "4Q Sales Forecast",
			"url": [
			{
			"@type": "Link",
			"href": "http://example.org/4q-sales-forecast.pdf",
			"mediaType": "application/pdf"
			},
			{
			"@type": "Link",

	Notes:	Identifies one or more links to representations of the object	
	Domain:	Object	
	Range:	xsd:anyURI | Link	

			"href": "http://example.org/4q-sales-forecase.html", "mediaType": "text/html" }] }

accuracy	URI:	http://www.w3.org/ns/activitystreams#accuracy	• JSON-LD
			
			EXAMPLE 610
			
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Place",
			"latitude": 36.75,
			"longitude": 119.7667,
			"accuracy": 94.5
			}

	Notes:	Indicates the accuracy of position coordinates on a Place objects. Expressed in properties of percentage. e.g. "94.0" means "94.0% accurate".	
	Domain:	Place	
	Range:	xsd:float [>= 0.0f, <= 100.0f]	
	Functional:	True	

alias	URI:	http://www.w3.org/ns/activitystreams#alias	• JSON-LD
			
			EXAMPLE 615
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Place",
			"displayName": "Home",
			"alias": "@home"
			}

	Notes:	Provides a contextually meaningful alternative label for the object in addition to the id. For instance, within some systems, groups can be identified by both a unique global identifier and a more "human-friendly" label such as "@friends" or "@network". The value of the alias property MUST match either the isegment-nz-nc or IRI productions in [RFC3987]. The use of a relative reference other than a simple name is not allowed. It is important to note that the meaning of the alias value is implementation and context dependent.	
	Domain:	Object	
	Range:	xsd:anyURI	
	Functional:	True	
altitude	URI:	http://www.w3.org/ns/activitystreams#altitude	• JSON-LD EXAMPLE 620 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Place", "displayName": "Fresno Area",
	Notes:	Indicates the altitude of a place. The measurement units is indicated using the units property. If units is not specified, the default is assumed to be "m" indicating meters.	
	Domain:	Object	

	Range:	xsd:float	"altitude": 15.0, "latitude": 36.75, "longitude": 119.7667, "units": "miles" }
	Functional:	True	

content	URI:	http://www.w3.org/ns/activitystreams#content	• JSON-LD
			• EXAMPLE 625
			
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"content": "A simple note"
			}
			
			• JSON-LD
			
			EXAMPLE 630
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"contentMap": [
			"en": "A simple note",
			"sp": "Una simple nota"
			}
			}

	Notes:	A natural language description of the object content. HTML markup, including visual elements such as images, MAY be included. The content MAY be expressed using multiple language-tagged values.	
	Domain:	Object	
	Range:	xsd:string | rdf:langString	

displayName	URI:	http://www.w3.org/ns/activitystreams#displayName	• JSON-LD
			
			EXAMPLE 635
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"displayName": "A simple note"
			}
			
			• JSON-LD
			
			EXAMPLE 640
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"displayNameMap": [
			"en": "A simple note",
			"sp": "Una simple nota"
			}
			}

	Notes:	A simple, human-readable, plain-text name for the object. HTML markup MUST NOT be included. The displayName MAY be expressed using multiple language-tagged values.	
	Domain:	Object | Link	
	Range:	xsd:string | rdf:langString	

duration	URI:	http://www.w3.org/ns/activitystreams#duration	• JSON-LD
			
			EXAMPLE 645
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Video",
			"displayName": "A Simple Video",
			"url": "http://example.org/video.mkv",
			"duration": "PT2H"
			}

	Notes:	When the object describes a time-bound resource, such as an audio or video, a meeting, etc, the duration property indicates the object's approximate duration. The value SHOULD be expressed as an [RFC3339] duration (e.g. a period of 5 seconds is represented as "PT5S") but MAY be specified as a non-negative integer specifying the duration as a number of non-fractional seconds.	

	Domain:	Content | Link	• JSON-LD
			•
			•
			•
			•
			
			EXAMPLE 650
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Video",
			"displayName": "A Simple Video",
			"url": "http://example.org/video.mkv",
			"duration": 3600
			}

	Range:	xsd:integer | xsd:duration	
	Functional:	True	

height	URI:	http://www.w3.org/ns/activitystreams#height	• JSON-LD
			
			EXAMPLE 655
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Content",
			"displayName": "Some generic content",
			"content": " This can be any kind of content ",
			"height": 100,
			"width": 100
			}

	Notes:	When the object describes a visual resource, such as an image, video or embeddable HTML, the height property indicates the recommended display height in properties of device-independent pixels.	
	Domain:	Content | Link	
	Range:	xsd:nonNegativeInteger	
	Functional:	True	
href	URI:	http://www.w3.org/ns/activitystreams#href	• JSON-LD
	Notes:	The target resource pointed to by	

		a Link.	EXAMPLE 660
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Link",
			"href": "http://example.org/abc",
			"hreflang": "en",
			"mediaType": "text/html",
			"displayName": "An example link"
			}

	Domain:	Link	
	Range:	xsd:anyURI	
	Functional:	True	

hreflang	URI:	http://www.w3.org/ns/activitystreams#hreflang	• JSON-LD
			
			EXAMPLE 665
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Link",
			"href": "http://example.org/abc",
			"hreflang": "en",
			"mediaType": "text/html",
			"displayName": "An example link"
			}

	Notes:	Hints as to the language used by the target resource. Value MUST be a [RFC5646] Language-Tag.	
	Domain:	Link	
	Range:	[RFC5646] Language Tag	
	Functional:	True	

itemsPerPage	URI:	http://www.w3.org/ns/activitystreams#itemsPerPage	• JSON-LD
			
			EXAMPLE 670
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 2,
			"itemsPerPage": 2,
			"items": [
			{
			"@type": "Note",
			"displayName": "A Simple Note"
			},
			{
			"@type": "Note",
			"displayName": "Another Simple Note"
			}
			]
			}

	Notes:	A non-negative integer specifying the maximum number of items that will be included in the value of the items array.	
	Domain:	Collection	
	Range:	xsd:nonNegativeInteger	
	Functional:	True	
latitude	URI:	http://www.w3.org/ns/activitystreams#latitude	• JSON-LD EXAMPLE 675 { "@context":
	Notes:	The latitude of a place	
	Domain:	Place	
	Range:	xsd:float	
	Functional:	True	

			"http://www.w3.org/ns/activitystreams", "@type": "Place", "displayName": "Fresno Area", "latitude": 36.75, "longitude": 119.7667, "radius": 15, "units": "miles" }

longitude	URI:	http://www.w3.org/ns/activitystreams#longitude	• JSON-LD
			
			EXAMPLE 680
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Place",
			"displayName": "Fresno Area",
			"latitude": 36.75,
			"longitude": 119.7667,
			"radius": 15,
			"units": "miles"
			}

	Notes:	The longitude of a place	
	Domain:	Place	
	Range:	xsd:nonNegativeInteger	
	Functional:	True	

mediaType	URI:	http://www.w3.org/ns/activitystreams#mediaType	• JSON-LD
			
			EXAMPLE 685
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Link",
			"href": "http://example.org/abc",
			"hreflang": "en",
			"mediaType": "text/html",
			"displayName": "An example link"
			}

	Notes:	When used on a Link, identifies the MIME media type of the referenced resource.	
	Domain:	Link	
	Range:	MIME Media Type	
	Functional:	True	

priority	URI:	http://www.w3.org/ns/activitystreams#priority	• JSON-LD
			
			EXAMPLE 690
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Share",
			"actor": "acct:sally@example.org",
			"object": "http://example.org/posts/1",
			"target": "acct:john@example.org",
			"priority": 0.80
			}

	Notes:	An optional indicator of the relative priority, or importance, that the creator of the activity considers it to have. Represented as a non-negative, numeric decimal between 0.00 and 1.00 (inclusive), with two decimal places of precision. If the property is omitted or set to null, the assumption is that a default priority can be assumed by the implementation. The value 0.00 represents the lowest possible	

		priority while 1.00 represents the highest. The use of the priority property does not impose any specific processing or display requirements on the part of any consuming implementation. Expressing the value as a range of numeric decimal values is intended to provide the greatest level of flexibility in the expression and consumption of prioritization detail. It is expected that implementors consuming activity objects containing priority will utilize and expose the additional information in a number of different ways depending on the unique requirements of each application use case. Many existing systems do not represent priority values as numeric ranges. Such systems might use fixed, labeled brackets such as "low", "normal" and "high" or "urgent". Similar mechanisms can be established, by convention, when using the priority property. In typical use, it is RECOMMENDED that implementations wishing to work with such defined categories treat priority property values in the range 0.00 to 0.25 as "low" priority; values greater than 0.25 to 0.75 as "normal" priority; and values greater than 0.75 to 1.00 as "high" priority. Specific implementations are free to establish alternative conventions for the grouping of priority values with the caveat that such conventions likely will not be understood by all implementations.	
	Domain:	Activity	
	Range:	xsd:float [>= 0.00, <= 1.00]	
	Functional:	True	
endTime	URI:	http://www.w3.org/ns/activitystreams#endTime	• JSON-LD
	Notes:	The date and time describing the	

		actual or expected ending time of the object. When used with an Activity object, for instance, the endTime property specifies the moment the activity concluded or is expected to conclude.	EXAMPLE 695
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Event",
			"displayName": "A Party!",
			"startTime": "2014-12-31T23:00:00-08:00",
			"endTime": "2015-01-01T06:00:00-08:00"
			}

	Domain:	Object	
	Range:	xsd:dateTime	
	Functional:	True	

published	URI:	http://www.w3.org/ns/activitystreams#published	• JSON-LD
			EXAMPLE 700
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"content": "A simple note",
			"published": "2014-12-12T12:12:12Z"
			}

	Notes:	The date and time at which the object was published	
	Domain:	Object	
	Range:	xsd:dateTime	
	Functional:	True	

startTime	URI:	http://www.w3.org/ns/activitystreams#startTime	• JSON-LD
			EXAMPLE 705
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Event",
			"displayName": "A Party!",
			"startTime": "2014-12-31T23:00:00-08:00",
			"endTime": "2015-01-01T06:00:00-08:00"
			}

	Notes:	The date and time describing the actual or expected starting time of the object. When used with an Activity object, for instance, the startTime property specifies the moment the activity began or is scheduled to begin.	
	Domain:	Object	
	Range:	xsd:dateTime	
	Functional:	True	
radius	URI:	http://www.w3.org/ns/activitystreams#radius	• JSON-LD EXAMPLE 710 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "Place", "displayName": "Fresno Area", "latitude": 36.75,
	Notes:	The radius from the given latitude and longitude for a Place. The units is expressed by the units property. If units is not specified, the default is assumed to be "m" indicating "meters".	
	Domain:	Place	
	Range:	xsd:float [>= 0.0f]	

	Functional:	True	"longitude": 119.7667, "radius": 15, "units": "miles" }

rating	URI:	http://www.w3.org/ns/activitystreams#rating	• JSON-LD
			
			EXAMPLE 715
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Review",
			"actor": "acct:sally@example.org",
			"object": "http://example.org/posts/1",
			"rating": 3.5,
			}

	Notes:	A quality rating expressed as a non-negative decimal number between 0.0 and 5.0 (inclusive) with one decimal place of precision.	
	Domain:	Object	
	Range:	xsd:decimal [>= 0.0f, <=5.0f]	
	Functional:	True	

rel	URI:	http://www.w3.org/ns/activitystreams#rel	• JSON-LD
			
			EXAMPLE 720
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Link",
			"href": "http://example.org/abc",
			"hreflang": "en",
			"mediaType": "text/html",
			"displayName": "An example link",
			"rel": ["canonical", "preview"]
			}

	Notes:	A link relation associated with a Link. The value MUST conform to both the [HTML5] and [RFC5988] "link relation" definitions. In the [HTML5], any string not containing the "space" U+0020, "tab" (U+0009), "LF" (U+000A), "FF" (U+000C), "CR" (U+000D) or "," (U+002C) characters can be used as a valid link relation.	
	Domain:	Link	
	Range:	[RFC5988] or [HTML5] Link Relation	
startIndex	URI:	http://www.w3.org/ns/activitystreams#startIndex	• JSON-LD EXAMPLE 725 { "@context": "http://www.w3.org/ns/activitystreams", "@type": "OrderedCollection", "totalItems": 2, "itemsPerPage": 2, "startIndex": 0, "orderedItems": [{ "@type": "Note", "displayName": "A Simple Note" }, { "@type": "Note", "displayName": "Another Simple
	Notes:	A non-negative integer value identifying the relative position within the logical view of a strictly ordered collection.	
	Domain:	OrderedCollection	
	Range:	xsd:nonNegativeInteger	
	Functional:	True	

			Note" }] }

summary	URI:	http://www.w3.org/ns/activitystreams#summary	• JSON-LD
			
			EXAMPLE 730
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"summary": "A simple note"
			}
			
			• JSON-LD
			
			EXAMPLE 735
			
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"summaryMap": [
			"en": "A simple note",
			"sp": "Una simple nota"
			}
			}

	Notes:	A natural language summarization of the object. HTML markup, including visual images such as images, MAY be included. Multiple language tagged summaries MAY be provided.	
	Domain:	Object	
	Range:	xsd:string | rdf:langString	

title	URI:	http://www.w3.org/ns/activitystreams#title	• JSON-LD
			
			EXAMPLE 740
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"title": "A simple note"
			}
			
			• JSON-LD
			
			EXAMPLE 745
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"titleMap": [
			"en": "A simple note",
			"sp": "Una simple nota"
			}

	Notes:	A natural language title of the object. HTML markup, including visual images such as images, MAY be included. The title and displayName properties are closely related and overlap in function with the key difference being that title is permitted to contain HTML markup while displayName is not.	
	Domain:	Object | Link	
	Range:	xsd:string | rdf:langString	

			}

totalItems	URI:	http://www.w3.org/ns/activitystreams#totalItems	• JSON-LD
			
			EXAMPLE 750
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Collection",
			"totalItems": 2,
			"itemsPerPage": 2,
			"items": [
			{
			"@type": "Note",
			"displayName": "A Simple Note"
			},
			{
			"@type": "Note",
			"displayName": "Another Simple Note"
			}
			]
			}

	Notes:	A non-negative integer specifying the total number of objects contained by the logical view of the collection. This number might not reflect the actual number of items serialized within the Collection object instance.	
	Domain:	Collection	
	Range:	xsd:nonNegativeInteger	
	Functional:	True	

units	URI:	http://www.w3.org/ns/activitystreams#units	• JSON-LD
			
			EXAMPLE 755
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Place",
			"displayName": "Fresno Area",
			"latitude": 36.75,
			"longitude": 119.7667,
			"radius": 15,
			"units": "miles"
			}

	Notes:	Specifies the measurement units for the radius and altitude properties on a Place object. If not specified, the default is assumed to be "m" for "meters".	
	Domain:	Place	
	Range:	"cm" | "feet" | "inches" | "km" | "m" | "miles" | xsd:anyURI	
	Functional:	True	

updated	URI:	http://www.w3.org/ns/activitystreams#updated	• JSON-LD
			
			EXAMPLE 760
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Note",
			"content": "A simple note",
			"updated": "2014-12-12T12:12:12Z"
			}

	Notes:	The date and time at which the object was updated	
	Domain:	Object	
	Range:	xsd:dateTime	
	Functional:	True	
width	URI:	http://www.w3.org/ns/activitystreams#width	• JSON-LD
	Notes:	When the object describes a visual	

		resource, such as an image, video or embeddable HTML, the width property indicates the recommended display width in properties of device-independent pixels.	EXAMPLE 765
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Content",
			"displayName": "Some generic content",
			"content": " This can be any kind of content ",
			"height": 100,
			"width": 100
			}

	Domain:	Content | Link	
	Range:	xsd:nonNegativeInteger	
	Functional:	True	

a	URI:	http://www.w3.org/ns/activitystreams#a	• JSON-LD
			
			EXAMPLE 770
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Connection",
			"a": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"relationship": "http://purl.org/vocab/relationship/closeFriendOf",
			"b": {
			"@type": "Person",
			"displayName": "John"
			}
			}

	Notes:	On a Connection object, the a property identifies one of the connected individuals. For instance, for a Connection object describing "John is connected to Sally", a would refer to John.	
	Domain:	Connection	
	Range:	Link | Object	
	Functional:	True	

b	URI:	http://www.w3.org/ns/activitystreams#b	• JSON-LD
			
			EXAMPLE 775
			{
			"@context": "http://www.w3.org/ns/activitystreams",
			"@type": "Connection",
			"a": {
			"@type": "Person",
			"displayName": "Sally"
			},
			"relationship": "http://purl.org/vocab/relationship/closeFriendOf",
			"b": {
			"@type": "Person",
			"displayName": "John"
			}
			}

	Notes:	On a Connection object, the b property identifies one of the connected individuals. For instance, for a Connection object describing "John is connected to Sally", b would refer to Sally.	
	Domain:	Connection	
	Range:	Link | Object	
	Functional:	True	

<table>
<tr><td rowspan="4">relationship</td><td>URI:</td><td>http://www.w3.org/ns/activitystreams#relationship</td><td rowspan="4">• JSON-LD

EXAMPLE 780
{
"@context": "http://www.w3.org/ns/activitystreams",
"@type": "Connection",
"a": {
"@type": "Person",
"displayName": "Sally"
},
"relationship": "http://purl.org/vocab/relationship/closeFriendOf",
"b": {
"@type": "Person",
"displayName": "John"
}
}</td></tr>
<tr><td>Notes:</td><td>On a Connection object, the relationship property identifies the kind of relationship that exists between a and b.</td></tr>
<tr><td>Domain:</td><td>Connection</td></tr>
<tr><td>Range:</td><td>Object</td></tr>
</table>

4. Conclusion: Open Problems and Next Steps for Standardisation

The standardization strategy of D-CENT to enable federated social networking is being enabled by the W3C's Social Web Working Group and Social Interest Group. Into the future, we expect these groups to continue and for D-CENT project partners to become more highly active in them. However, already there are a number of fundamental research questions that must be addressed for the long-term success of decentralized social networking standardization. This section is dealt with first, and then an outline of the next steps for the D-CENT project and platform in terms of standardization. The author would like to thank and acknowledge the considerable help of George Danezis (UCL) and Nadim Koebessi (Inria) in working through these problems and contributing to the text in terms of the open research problems, and like to thank D-CENT partner Jaakko Korhonen (OKFN) for ideas on next steps.

4.1. Open Problems in Decentralisation

Key distribution and management have been the subject of considerable study: the seminal work of Diffie and Hellman introduced public key cryptography[i] as a means to do away with secret key distribution. However, the need to securely associate public keys with the correct communication partners persists. Approaches involving Public Key Infrastructures (PKI) were deployed to distribute keys of services, and more recently the shortcomings of PKI[ii] are addressed by the Certificate Transparency[iii] solution. These are difficult to extend and scale to associating keys with people rather than services (e.g. there are hundreds of million people using Google services, but modern PKI handles at most 5 million public key certificates at any time). Advanced "trust management" systems combine authentication and authorization, and allow for distributed naming, such as SDSI[iv] and TAOS[v]. However, the architecture of these designs predates modern Internet services, and presupposes a strong and hierarchical relationship of trust between end-users and well known institutions (their employer, school or government) rather than a decentralized model as needed by D-CENT. The Pretty Good Privacy (PGP) email encryption software relies on a "web-of-trust" instead of a centralized PKI, where users cross sign and validate each others' keys[vi]. This approach has well known problems, starting with the fact that trust is not transitive, limiting the reach of cross validation[vii], and a large number of usability problems[viii] resulting from exposing users to the intimate details in key management.

A number of approaches untie identity management from key management, and simply attempt to provide a service that authenticates users, in either a centralized or decentralized manner. Such mainstream centralized services have included Microsoft Passport[ix] and CardSpace[x] that have failed to be adopted. De-centralized single sign on systems, such as Liberty[xi], have also seen only limited deployment, and instead centralized systems associated with large service providers such as Facebook connect[xii] and Google sign-on[xiii] are now widely used, locking-in users to proprietary eco-systems. Open standards such as SAML[xiv], OpenID[xv] and Shiboleth[xvi] replicate in the authentication space the assumption of a strong relationship between a user and a hierarchical service provider, school / university or employer, and as a result do not provide strong de-centralization, user mobility, flexibility or privacy to support

user centric authentication and key distribution. More cryptographic privacy-friendly authentication / authorization mechanism such as UProve[xvii] and Idemix[xviii] involve selective disclosure of credentials. How to deploy such technologies within de-centralized settings, without fixed identity providers, has never fully happened due to the inability of these solutions to be easily installed and understood by users.

Bootstrapping strong identity in pure peer-to-peer settings is an open research problem. A key challenge is thwarting "Sybil attacks"[xix] which see a malicious entity creating a number of "fake" users to foil abuse detection and prevention mechanisms, manipulate reputation mechanisms, or win elections. Although researchers have proposed the first decentralized sybil defence[xx] for Distributed Hash Tables[xxi], which are a new field of defences based on using social relations such as SybilGuard[xxii], SybilLimit[xxiii] and our own SybilInfer.[xxiv] These form a key protection mechanism for attempts at building decentralized name systems, such as the relatively new GNU name system.[xxv] Other de-centralized defences against sybil attacks involve proof-of-work[xxvi] that most notably forms the sybil protection mechanism behind the Bitcoin digital currency.[xxvii] These kinds of defenses point to the fact that D-CENT has made the right design choice in terms of the blockchain in order to defend the 'collective memory' of decision-making and even digital social currency transactions, but more research in privacy and security is needed to see if this scales.

Future standards will be needed that **leverage peer relationships to establish identity**, and allow users to privately measure the degree of confidence or reputation in other users. Traditional identity mechanisms see authority to designate users flow hierarchically from a higher identity provider. Future research and standards could use both use the social graph of peer relationships to combat large-scale sybil attacks, but will also **apply ideas from digital currencies to identity** where possible to make it very expensive to create multiple fake accounts. In particular proofs-of-work from Bitcoin could be reused to establish a presence or links in a social graph. Conversely as explored in the design of the e-voting and digital social currency in D-CENT, **social capital itself may be used as a proof of work**, and be used to bootstrap a virtual currency representing social trust -- that may be transferred or traded to support a federated identity system. **Thus block chain technology may be used in a unified way to support both identity and value transfer**.

Another objective of future work is to **provide privacy to the users' social graph** while leveraging it to strength security. Therefore, operations on the social graph, such as adding or removing friends, presence, or establishing reputation metrics will be implemented using privacy-preserving protocols. Yet currently the research in this area is rather immature and thus cannot be standardized as the protocols in particular have not been applied to this space of problems invoked by decentralized systems.

The vast majority of email correspondence today is not encrypted end-to-end. The de-facto standards for email confidentiality and integrity is PGP[xxviii], which has been standardized as OpenPGP[xxix]. The S/MIME[xxx] alternative has seen adoption within corporate environments, but little across organizations or between peer users, mostly due to the need for user certificates. Poor client support throughout the 2000' and the subsequent rise in popularity of webmail clients (Gmail, Hotmail, …) which do not support either PGP or S/MIME, has hampered the deployment of those protocols. Instead server-to-server encryption has seen some deployment in the form of STARTTLS[xxxi]. A key challenge of asynchronous messaging is ensuring Forward Secrecy, namely preventing subsequent key compromise from affecting past communications. PGP separates encryption and signing keys to facilitate key rotation,

but this is neither automated nor a perfect defence. If we expect that federated social systems will replace email (as some predict that Facebook Messenger, GChat, Snapchat, etc. will do), then the total failure of e-mail to provide authentication and encryption needs to be taken seriously. We do not want federated social networking repeating the mistakes of the past as given by e-mail and so leading to federated social networking systems being a haven for spam.

Similarly to asynchronous email messaging, the vast majority of synchronous chat messaging is currently not end-to-end encrypted. Worse, unlike e-mail it is not even decentralized except for XMPP. In the case of XMPP, Off-the-Record messaging[xxxii] has existed as a de-facto standard for encrypted chat since 2004. However, OTR implementation efforts have largely been restricted to certain low-usage clients. These problems are compounded by the fact that unlike email, chat-based messaging currently exists in dozens of "silos" that are completely unable to inter-operate with each other: WhatsApp users cannot chat with Google Talk users, and so on. This offers a stark contrast to the email model, where any email service can openly communicate with another (Gmail to Outlook, etc.). In this regard, following the federated model of email makes sense for federated social networking, and with a 'push' protocol could be used for chat.

Confidentiality of content is not sufficient to protect users and their communications against traffic analysis that allows network observers to extract information from the patterns of who is talking to whom. To prevent traffic analysis a number of systems including Mixmaster[xxxiii] and our own Mixminion[xxxiv] were designed, that anonymize who is sending mail to whom. These high-latency anonymity systems have lost in popularity compared with the real-time Tor[xxxv] system. However, they promise, if widely adopted[xxxvi], to provide stronger privacy guarantees than Tor which we have shown to exhibit a number of covert channel[xxxvii] and other[xxxviii] attacks. While researchers have proposed more modern mix systems like Minx[xxxix], Sphinx[xl] and Drac[xli] but those have not had the opportunity to be deployed beyond the lab. For use in high-security situations and in order to preserve the rights of data protection, these kinds of techniques need to be studied more.

While the standardization of the W3C Web Crypto API[xlii] to extend support for such privacy systems to browser based applications may help, there are still larger problems around XSS (Cross-side scripting) attacks and other long-standing issues that need to be addressed before browsers can be used for high-security applications. Given that D-CENT primarily functions as a Web application rather than a 'native' application, it makes sense that addressing these security and privacy concerns as part of a larger standardization strategy at the W3C be of concern to D-CENT and any future effort in collective awareness platforms.

Any future successor platform to the D-CENT platform and W3C Social Web standards will need to implement **native secure asynchronous messaging**, both in terms of providing strong **content confidentiality and authenticity**, but also in terms of **hiding messaging meta-data**. Furthermore, it must to provide secure messaging through de-centralized mechanisms, that allow used to **choose and switch service providers**; **providers remain oblivious to the content** of messages, and the **social contacts of their users remain private**; and users may use messaging to **bootstrap trust relations, form groups and partake in collective processes** such as voting, polling, decision making and resource allocation (collective intelligence) while keeping their individual preferences private and secure. There is currently no system that supports these features, let alone with the degree of **de-**

centralization or privacy with respect to meta-data the D-CENT project aims to enable in the long-term.

4.2. Next Steps for D-CENT and Standardisation

The next steps have been outlined in the document, but summarizing them should provide a clear roadmap.

- The W3C Social Web Working Group will continue to mature the Activity Streams 2.0 and Activity Vocabulary specifications into full standards.

- The W3C Social Web Working Group will create a Social API (and if needed, a Federation Protocol) to finish the 'suite' of open standards needed for federated social networking.

- New use-cases will be discussed with the W3C Social Interest Group, including the use-cases particular to the D-CENT project that are not covered by generic federated social networking use-cases.

- An architecture diagram and easy-to-understand developer primer will be created by the D-CENT project in conjunction with the Social Interest Group, and this primer should guide implementation by Citizens Foundation, OKF, and Thoughtworks of federated social networking.

- A vocabulary detailing the exact kinds of transactions needed by the decision-making use-cases as well as possibly the digital social currency use-cases should be created as an extension to the Activity Vocabulary. This vocabulary can be submitted to the Social Interest Group.

- Any open-source work done in terms of creating ActivityStreams 2.0 compliant code by any of the pilots should

- The pilots should document their interoperability and share their experiences, including their experiences dealing with relatively hard open problems, with the W3C in order to determine what new Working Groups at IETF or W3C need to be chartered to solve these open problems once the research community has investigated them and reached a rough consensus on the solution.

Although it is a large amount of work, this work should mostly be tested via implementation in D5.6 and D5.8, and then finalized by the end of the D-CENT project in D5.8. We expect that will be a crucial part of helping a federated social web reach its full potential.

References

[i] Whitfield Diffie, Martin E. Hellman: New directions in cryptography. IEEE Transactions on Information Theory 22(6): 644-654 (1976)
[ii] Carl M. Ellison, Bruce Schneier: Risks of PKI: Secure Email. Commun. ACM 43(1): 160 (2000)
[iii] Ben Laurie: Certificate transparency. Commun. ACM 57(10): 40-46 (2014)
[iv] Dwaine E. Clarke, Jean-Emile Elien, Carl M. Ellison, Matt Fredette, Alexander Morcos, Ronald L. Rivest: Certificate Chain Discovery in SPKI/SDSI. Journal of Computer Security 9(4): 285-322 (2001)
[v] Edward Wobber, Martín Abadi, Michael Burrows, Butler W. Lampson: Authentication in the Taos Operating System. SOSP 1993: 256-269
[vi] Caronni, Germano. "Walking the web of trust." *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.(WET ICE 2000). Proeedings. IEEE 9th International Workshops on*. IEEE, 2000.
[vii] Christianson, Bruce, and William S. Harbison. "Why isn't trust transitive?." *Security protocols*. Springer Berlin Heidelberg, 1997.
[viii] Whitten, Alma, and J. Doug Tygar. "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0." *Usenix Security*. Vol. 1999. 1999.
[ix] Raymond Choo, Kim-Kwang. "Issue report on business adoption of Microsoft Passport." *Information management & computer security* 14.3 (2006): 218-234.

[x] Alrodhan, Waleed A., and Chris J. Mitchell. "Addressing privacy issues in CardSpace." *Information Assurance and Security, 2007. IAS 2007. Third International Symposium on*. IEEE, 2007.
[xi] Pfitzmann, Birgit, and Michael Waidner. "Analysis of liberty single-sign-on with enabled clients." *Internet Computing, IEEE* 7.6 (2003): 38-44.
[xii] Miculan, Marino, and Caterina Urban. "Formal analysis of Facebook Connect single sign-on authentication protocol." *SOFSEM*. Vol. 11. 2011.
[xiii] Wang, Rui, Shuo Chen, and XiaoFeng Wang. "Signing me onto your accounts through facebook and google: A traffic-guided security study of commercially deployed single-sign-on web services." *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 2012.
[xiv] Armando, Alessandro, et al. "Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps." *Proceedings of the 6th ACM workshop on Formal methods in security engineering*. ACM, 2008.
[xv] Recordon, David, and Drummond Reed. "OpenID 2.0: a platform for user-centric identity management." *Proceedings of the second ACM workshop on Digital identity management*. ACM, 2006.
[xvi] Needleman, Mark. "The Shibboleth authentication/authorization system." *Serials Review* 30.3 (2004): 252-253.
[xvii] Mostowski, Wojciech, and Pim Vullers. "Efficient U-Prove implementation for anonymous credentials on smart cards." *Security and Privacy in Communication Networks*. Springer Berlin Heidelberg, 2012. 243-260.

[xviii] Camenisch, Jan, and Els Van Herreweghen. "Design and implementation of the idemix anonymous credential system." *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002.
[xix] Douceur, John R. "The sybil attack." *Peer-to-peer Systems*. Springer Berlin Heidelberg, 2002. 251-260.
[xx] Danezis, G., Lesniewski-Laas, C., Kaashoek, M. F., & Anderson, R. (2005). Sybil-resistant DHT routing. In *Computer Security–ESORICS 2005* (pp. 305-318). Springer Berlin Heidelberg.
[xxi] Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., & Balakrishnan, H. (2001). Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, *31*(4), 149-160.
[xxii] Yu, H., Kaminsky, M., Gibbons, P. B., & Flaxman, A. (2006, September). Sybilguard: defending against sybil attacks via social networks. In *ACM SIGCOMM Computer Communication Review* (Vol. 36, No. 4, pp. 267-278). ACM.
[xxiii] Yu, H., Gibbons, P. B., Kaminsky, M., & Xiao, F. (2008, May). Sybillimit: A near-optimal social network defense against sybil attacks. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on* (pp. 3-17). IEEE.
[xxiv] Danezis, George, and Prateek Mittal. "SybilInfer: Detecting Sybil Nodes using Social Networks." *NDSS*. 2009.
[xxv] Wachs, Matthias, Martin Schanzenbach, and Christian Grothoff. "A Censorship-Resistant, Privacy-Enhancing and Fully Decentralized Name System." *Cryptology and Network Security*. Springer International Publishing, 2014. 127-142.
[xxvi] Back, Adam. "Hashcash-a denial of service counter-measure." (2002).
[xxvii] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Consulted*, *1*(2012), 28.
[xxviii] Zimmermann, Philip R., and Philip R. Zimmermann. *The official PGP user's guide*. Vol. 265. Cambridge: MIT press, 1995.
[xxix] Elkins, M., Del Torto, D., Levien, R., & Roessler, T. (2001). MIME security with OpenPGP. *RFC3156*.
[xxx] Ramsdell, Blake. "Secure/multipurpose internet mail extensions (S/MIME) version 3.1 message specification." (2004).
[xxxi] Hoffman, P. (2002). SMTP service extension for secure SMTP over Transport Layer Security.
[xxxii] Borisov, Nikita, Ian Goldberg, and Eric Brewer. "Off-the-Record Communication, or, Why Not To Use PGP." *Workshop on Privacy in the Electronic Society*. WPES 2004.
[xxxiii] Möller, U., Cottrell, L., Palfrader, P., & Sassaman, L. (2003). Mixmaster protocol—version 2. *Draft, July*.
[xxxiv] Danezis, George, Roger Dingledine, and Nick Mathewson. "Mixminion: Design of a type III anonymous remailer protocol." *Security and Privacy, 2003. Proceedings. 2003 Symposium on*. IEEE, 2003.
[xxxv] Dingledine, Roger, Nick Mathewson, and Paul F. Syverson. "Tor: The Second-Generation Onion Router." *USENIX Security Symposium*. 2004.
[xxxvi] Dingledine, Roger, and Nick Mathewson. "Anonymity Loves Company: Usability and the Network Effect." In *WEIS*. 2006.
[xxxvii] Murdoch, S. J., & Danezis, G. (2005, May). Low-cost traffic analysis of Tor. In *Security and Privacy, 2005 IEEE Symposium on* (pp. 183-195). IEEE.
[xxxviii] Borisov, N., Danezis, G., Mittal, P., & Tabriz, P. (2007, October). Denial of service or denial of security?. In *Proceedings of the 14th ACM conference on Computer and communications security* (pp. 92-102). ACM.

[xxxix] George Danezis, Ben Laurie: Minx: a simple and efficient anonymous packet format. WPES 2004: 59-65
[xl] George Danezis, Ian Goldberg: Sphinx: A Compact and Provably Secure Mix Format. IEEE Symposium on Security and Privacy 2009: 269-282
[xli] George Danezis, Claudia Díaz, Carmela Troncoso, Ben Laurie: Drac: An Architecture for Anonymous Low-Volume Communications. Privacy Enhancing Technologies 2010: 202-219
[xlii] http://www.w3.org/TR/WebCryptoAPI/