

Thomas Koch  
thomas@koch.ro  
matriculation number 7250371

December 3, 2011  
Fernuniversität Hagen  
Faculty of mathematics and computer science

## Contents

<b>1. Motivation</b>	<b>3</b>
<b>2. Is webdav restful?</b>	<b>3</b>
<b>3. Kolab and its use of IMAP</b>	<b>4</b>
<b>4. Persistency for Groupware Data</b>	<b>5</b>
<b>5. Synchronizing a large collection</b>	<b>5</b>
<b>6. Mediatypes</b>	<b>5</b>
6.1. Updates with alternative Mediatypes . . . . .	5
6.2. Mediatype conversion . . . . .	5
<b>7. Hypermedia in RESTful applications</b>	<b>5</b>
7.1. Hypermedia in OpenSocial . . . . .	5
<b>8. Is OpenSocial restful?</b>	<b>6</b>
<b>9. Selection of components</b>	<b>7</b>
9.1. REST framework . . . . .	7
<b>10. Testing</b>	<b>7</b>
<b>A. Standards</b>	<b>7</b>
A.1. Contacts / Persons . . . . .	7
A.2. Calendaring . . . . .	8
A.3. Scheduling . . . . .	8
A.4. Relations and Links . . . . .	9
A.5. out of scope . . . . .	9
<b>B. People, Groups and Organizations</b>	<b>9</b>

<b>C. Implementations</b>	<b>10</b>
C.1. Servers . . . . .	10
C.2. Clients . . . . .	11
C.3. Web Services . . . . .	12
C.4. Portable Contacts . . . . .	12
<b>D. Links</b>	<b>12</b>
D.1. Apache Shindig . . . . .	12
D.2. Socialsite . . . . .	12
<b>E. TODO</b>	<b>12</b>

# 1. Motivation

Why vCard/CardDav: many clients

Why OpenSocial / Portable Contacts:

- used by Google, LinkedIn,
- used in Enterprise applications like Atlassian tools (Jira, Confluence, ...), Nuxeo CMS, ...
- OpenSocial can be used to implement inhouse portals and populate it with data from the companies GroupWare

## 2. Is webdav restful?

Roy Fielding says no: <http://tech.groups.yahoo.com/group/rest-discuss/message/5874>

PROP\* methods conflict with REST because they prevent important resources from having URIs and effectively double the number of methods for no good reason. Both Henrik and I argued against those methods at the time. It really doesn't matter how uniform they are because they break other aspects of the overall model, leading to further complications in versioning (WebDAV versioning is hopelessly complicated), access control (WebDAV ACLs are completely wrong for HTTP), and just about every other extension to WebDAV that has been proposed.

[...]

The problem with MOVE is that it is actually an operation on two independent namespaces (the source collection and destination collection). The user must have permission to remove from the source collection and add to the destination collection, which can be a bit of a problem if they are in different authentication realms. COPY has a similar problem, but at least in that case only one namespace is modified. I don't think either of them map very well to HTTP.

The discussion also continued on the microformats mailing list <http://microformats.org/discuss/mail/microformats-rest/2006-April/thread.html#217>.

see Amundsen2010 for a restful approach to properties.

Is ATOM an alternative to WebDAV?

AtomPub is different from DAV in two key respects:

- The client doesn't control where things go, the server does
- It is allowed and expected that an AtomPub server will look at the incoming information and change it (generate ID, timestamps, sanitize HTML, etc)

Tim Bray, <http://www.imc.org/atom-protocol/mail-archive/msg11271.html>

Examples of APIs that use REST instead of WebDAV:

- Amazon S3 <http://awsmedia.s3.amazonaws.com/pdf/RESTandS3.pdf> <http://docs.amazonwebservices.com/AmazonS3/latest/dev/RESTAPI.html>

### 3. Kolab and its use of IMAP

Kolab uses an IMAP server as the data store and synchronization protocol for calendar and contact informations. I want to compare this approach to a restful one.

Advantages of IMAP:

- already there, since Mail uses it
- can store blobs/files so no need to map the iCal/vCard files to a relational scheme
- out of the box support for offline work and later synchronization (How does it solve editing conflicts?)

Disadvantages of IMAP:

- Complicate, 38 RFCs according to [http://de.wikipedia.org/wiki/Internet\\_Message\\_Access\\_Protocol](http://de.wikipedia.org/wiki/Internet_Message_Access_Protocol) see also: <http://www.apps.ietf.org/rfc/ipoplist.html>
- All clients directly access the iCal/vCard files with no moderation layer in between. This means that no validation or normalization can be done. Schema updates can only be done if all clients cooperate.
- IMAP imposes a folder structure. Google's gmail is an example for another, tag based approach. Messages could have several tags. It is therefor hard to access Gmail via IMAP.
- Sam Varshavchik, the author of the courier Mail Transfer Agent explains the history of IMAP and claims that the IMAP standard is broken: <http://www.courier-mta.org/fud/>
- IMAP is so complicate that the IMAP wiki holds 10 pages of advises for IMAP client authors: <http://www.imapwiki.org/ClientImplementation> RFC 2683 "IMAP4 Implementation Recommendations" is a 23 pages document (cut 5 pages for verbosity) explaining how to implement another RFC standard. Is there any widely used standard that needs another RFC explaining how to implement it?
- [http://en.wikipedia.org/wiki/Internet\\_Message\\_Access\\_Protocol#Disadvantages](http://en.wikipedia.org/wiki/Internet_Message_Access_Protocol#Disadvantages)
- Some attempts to create a simpler alternative to IMAP:
  - <http://en.wikipedia.org/wiki/POP4>
  - [http://en.wikipedia.org/wiki/Simple\\_Mail\\_Access\\_Protocol](http://en.wikipedia.org/wiki/Simple_Mail_Access_Protocol) also here <http://www.courier-mta.org/cone/smap1.html>
  - [http://en.wikipedia.org/wiki/Internet\\_Mail\\_2000](http://en.wikipedia.org/wiki/Internet_Mail_2000)
  - HTTP restful: <http://tools.ietf.org/id/draft-dusseault-httpmail-00.txt> mailing list: <https://www.ietf.org/mailman/listinfo/httpmail>
  - BikINI is not IMAP <http://bikini.caterva.org>
  - Outlook uses HTTP to communicate with Hotmail
  - another rest mail proposal: <http://www.prescod.net/rest/restmail/>
- more rants: <http://blog.gaborcselle.com/2010/02/how-to-replace-imap.html>
- IMAP issues found by the chandler project <http://chandlerproject.org/bin/view/Jungle/IntrinsicI>

## 4. Persistency for Groupware Data

Relational Databases vs. NoSQL databases vs. plain files

Relational databases are not practical for contacts, events or todos. Common patterns in systems that use relational DBs for that purpose:

- artificial limits of entries, e.g. only 3 email addresses per contact, because there are only three columns email1, email2 and email3.
- Fields for custom data like custom1 to customX
- EAV pattern: tables like: id, foreign\_id, type, value

## 5. Synchronizing a large collection

How to efficiently synchronize a large collection of contacts with the server without checking each contact for changes?

Portable Contacts has a filter “updatedSince”.

How is synchronization done in CardDAV?

## 6. Mediatypes

### 6.1. Updates with alternative Mediatypes

How to handle updates, if the mediatypes are not isomorph?

How does Google handle PATCH in the calendar API?

### 6.2. Mediatype conversion

Which fields of portable contacts are derived from vCard: <http://wiki.portablecontacts.net/w/page/17776141/schema>

No single data representation is ideal for every client. This protocol defines representations for each resource in three widely supported formats, JSON [RFC4627], XML, and Atom [RFC4287] / AtomPub [RFC5023], using a set of generic mapping rules. The mapping rules allow a server to write to a single interface rather than implementing the protocol three times.

[?, Core API Server]

## 7. Hypermedia in RESTful applications

The model application is therefore an engine that moves from one state to the next by examining and choosing from among the alternative state transitions in the current set of representations.

[?, sec. 5.3, p.103]

### 7.1. Hypermedia in OpenSocial

Webfinger, e.g. get a profile picture from an email address

Danger: One can trigger na http request by sending an email.

## 8. Is OpenSocial restful?

Fielding says OpenSocial is not Restful <http://roy.gbiv.com/untangled/2008/rest-apis-must-be-discussion>: [http://groups.google.com/group/opensocial-and-gadgets-spec/browse\\_thread/thread/thread/aff4ba7373e21284/201a413efa67c26e](http://groups.google.com/group/opensocial-and-gadgets-spec/browse_thread/thread/thread/aff4ba7373e21284/201a413efa67c26e)

“OpenSocial defines several services for providing access to a container’s data.”

[?, Social API Server, sec 2, Services]

**Parameters in path** The REST-URI-Fragment is not a path but an encoding of parameters for a procedure or service call.

Each service type defines an associated partial URI format. The base URI for each service is found in the URI element associated with the service in the discovery document. Each service type accepts parameters via the URL path. Definitions are of the form:

{a}/{b}/{c}

[?, Core API Server, sec 2.1.1.2.2, REST-URI-Fragment]

Links structure in OpenSocial

URI fragment	Method	Description
/people/{User-Id}/@self	GET	profile for User-Id
/people/{User-Id}/@self	DELETE	remove User
/people/{User-Id}/{Group-Id}	GET	full profiles of group members
	POST	Create relationship, target specified by <entry><id> in body
	POST	Update Person
/people/{Initial-User-Id}/	GET	???
{Group-Id}/{Related-User-Id}		
/people/@supportedFields	GET	list of supported person profile fields
/groups/{User-Id}/[{Group-Id}]	GET	one or all groups of a user
	PUT	update group
	DELETE	delete group
/groups/{User-Id}	POST	create group

Table 1: URI fragments for peoples and groups in the OpenSocial REST API

- Collections are no entities of their own, but GET requests to entities returns lists of full objects.

The last URI is obviously missing an “s” behind mediaItem. This typo is present in the OpenSocial spec since Version 1.0, released in march 2010. This is of course not a big issue in itself, but rather a sign that the specification is too verbose and does specify things that should be self evident.

URI fragment	Method	Description
/albums/{User-Id}/@self	POST	create album
/albums/{User-Id}/{Group-Id}[/Album-Id]*	GET	one or multiple albums
/mediaItems/{User-Id}/{Group-Id}/{Album-Id}/{MediaItem-Id}	GET	one mediaitem
/mediaItem/{User-Id}/@self/{Album-Id} (sic!)	POST	create mediaitem

Table 2: URI fragments for albums and mediaitems in the OpenSocial REST API

**Conclusion** The so called REST API of OpenSocial is not restful but rather another RPC interface in another style. An indication of this can be seen in the frequent use of the word “Service” and the unnecessary elaboration of basic HTTP concepts. The Core API Server specification repeats the definition of HTTP methods (2.1.1.1 REST-HTTP-Method), requests (2.1.1 REST Request) and responses (2.1.2 REST Response). Even the structure of an HTTP URI is redefined (2.1.1.2 REST-Request-URI).

## 9. Selection of components

Apache Shindig for Open Social, includes client tests

<http://code.google.com/p/kolab-android/>

<https://evolis.org/projects/kolab-ws/>

<http://packages.ubuntu.com/source/maverick/dovecot-metadata-plugin> <https://launchpad.net/ubuntu/maverick/ubuntu/8-0ubuntu1>

### 9.1. REST framework

Jersey recommended by [?] above Restfulie and RESTeasy because of maturity and flexibility.

## 10. Testing

How to test the ReST/CardDAV interface?

Portable Contacts test client at plaxo <http://www.plaxo.com/pdata/testClient>

<http://code.google.com/p/rest-assured/> <http://restfuse.com/>

## A. Standards

### A.1. Contacts / Persons

#### RFC 6450 vCard Format Specification

This document defines the vCard data format for representing and exchanging a variety of information about individuals and other entities (e.g., formatted and structured name and delivery addresses, email address, multiple telephone numbers, photograph, logo, audio clips, etc.). This is the new version and obsoletes RFCs 2425, 2426, and 4770, and updates RFC 2739.

#### RFC 6351 xCard: vCard XML Representation

This document defines the XML schema of the vCard data format.

## **Portable Contacts, OpenSocial**

Portable Contacts defines contact data structures and a ReST API. It has been integrated in the OpenSocial standard.

## **Nepomuk Semantic Desktop Contact Ontology**

## **Friend of a friend (FOAF)**

FOAF is a

## **hCard**

## **A.2. Calendaring**

### **RFC 5545 Internet Calendaring and Scheduling Core Object Specification**

iCalendar is the core data schema for calendaring information. This is the new version and obsoletes RFC2445

### **RFC 6321 xCal: The XML format for iCalendar**

This specification defines a format for representing iCalendar data in XML. More specifically, is to define an XML format that allows iCalendar data to be converted to XML, and then back to iCalendar, without losing any semantic meaning in the data. Anyone creating XML calendar data according to this specification will know that their data can be converted to a valid iCalendar representation as well.

### **CalWS RESTful Web Services Protocol for Calendaring**

This document, developed by the XML Technical Committee, specifies a RESTful web services Protocol for calendaring operations. This protocol has been contributed to OASIS WS-CALENDAR as a component of the WS-CALENDAR Specification under development by OASIS.

### **Google Calendar API V3**

While not being a standard, the Google Calendar API is RESTful and will surely be implemented by many client applications. It's remarkable that the API supports partial GETs returning only specified fields and the HTTP PATCH verb to update only specified fields.

### **Open Services for Lifecycle Collaboration (OSLC)**

uses FOAF person [http://open-services.net/bin/view/Main/OSLCCoreSpecAppendixA?sortcol=table;up=#foaf\\_Person\\_Resource](http://open-services.net/bin/view/Main/OSLCCoreSpecAppendixA?sortcol=table;up=#foaf_Person_Resource)

provides change management, some overlapping to iCal TODOs <http://open-services.net/bin/view/Main/CmSpecificationV2>

reference implementation: <http://eclipse.org/lyo>



## **A.3. Scheduling**

### **RFC 5546 iCalendar Transport-Independent Interoperability Protocol (iTIP)**

Scheduling Events, BusyTime, To-dos and Journal Entries; Specifies the mechanisms for calendaring event interchange between calendar servers. This is the new version and obsoletes RFC2446

### **RFC 6047 iCalendar Message-Based Interoperability Protocol (iMIP)**

Specifies how to exchange calendaring data via e-mail. This is the new version and obsoletes RFC2447.

## **A.4. Relations and Links**

### **Xhtml Friends Network (XFN)**

One of the relations returned by Google's webfinger.

### **Webfinger**

Webfinger in Firefox Contacts Add-On <http://mozillalabs.com/blog/2010/03/contacts-in-the-1>

### **RFC 6415 Web Host Metadata**

### **Extensible Resource Descriptor (XRD)**

## **A.5. out of scope**

### **OMA Converged Address Book V1.0**

Standard by the Open Mobile Alliance defining data structures and synchronization of contact data. It references vCard.

### **W3C Contacts API**

A standard on how address books could be accessed on devices or from JavaScript inside a Web Browser. The standard references vCard, OMA Converged Address Book and Portable Contacts.

### **W3C vCard ontology**

### **W3C PIM ontology**

### **HR XML**

The HR-XML Consortium is the only independent, non-profit, volunteer-led organization dedicated to the development and promotion of a standard suite of XML specifications to enable e-business and the automation of human resources-related data exchanges.

## B. People, Groups and Organizations

### People

**Eliot Lear** `jlear@cisco.com`  
IETF Calsify WG chair

**Lisa Dusseault**

Lisa Dusseault is a development manager and standards architect at the Open Source Applications Foundation, where she's involved in the Chandler, Cosmo and Scooby projects. Previously, Lisa came from Xythos, an Internet startup where she was development manager for four years. She has also been an IETF contributor on various Internet applications protocols for eight years now, and continues to do this kind of work at OSAF. She co-chairs the IETF IMAP extensions and CALSIFY (Calendaring and Scheduling Standards Simplification) Working Groups. She is also the author of a book on WebDAV and co-author of CalDAV, an open and interoperable protocol for calendar access and sharing.

**Peter Saint-Andre** `jstpeter@stpeter.im`

IETF Calsify WG area director

**Joseph Smarr**

former Plaxo now Google presentation about portable contacts at vcarddav wg <http://tools.ietf.org/agenda/2004-09-08/2.pdf> <http://josephsmarr.com> <http://anyasq.com/79-im-a-technical-lead-on-the-google+-team>

**Mike Conley**

<http://mikeconley.ca/blog/> working on a new address book for Thunderbird: <https://wiki.mozilla.org/Thunderbird/tb-planning>

## C. Implementations

### C.1. Servers

**Cyn.in**  
Python, Open Core

**DAViCal**

PHP, SQL storage, CalDAV, CardDav

**eGroupWare**

**eXo Platform**  
Open Core, Java, AGPL, participates in OpenSocial?

**Group-Office**  
PHP, AGPL

**Horde**

**OBM Groupware**

PHP, GPL

**owncloud**

ownCloud supports syncing of calendar and contacts information via the CalDAV and CardDAV protocols.

**Scalix**

Open Core Scalix Public License (SPL) based on MPL, requires to show the Scalix Logo

**Simple Groupware**

PHP, GPL, SQL

**SOG**

CalDAV and CardDAV, written in Objective-C

**Tiki Wiki**

PHP, SQL Contacts <http://doc.tiki.org/Contacts>, Calendar <http://doc.tiki.org/Calendar> iCal export apparently no CardDAV/CalDAV many many features!

**Tine 2.0**

Tine is not eGroupWare

**Zarafa**

**Zimbra**

Open Core, Own license (Zimbra Public License), RFP since 2008 open: <http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=498316>

## **C.2. Clients**

**Spicebird**

built on top of Thunderbird with Calendar

**Thunderbird**

CardDAV via SoCO connector <http://www.sogo.nu/fr/downloads/frontends.html>

**Evolution, Evolution Data Server**

**KDE Kontact, Akonadi**

**more CardDAV**

<http://wiki.davical.org/w/CardDAV/Clients> <http://en.wikipedia.org/wiki/CardDAV#Implementations>

**more CalDAV**

[http://wiki.davical.org/w/CalDAV\\_Clients](http://wiki.davical.org/w/CalDAV_Clients) <http://en.wikipedia.org/wiki/CalDAV#Implementations>

## C.3. Web Services

## C.4. Portable Contacts

## D. Links

- <http://thesocialweb.tv>
- <http://www.vogella.de/articles/REST/article.html> REST with Java (JAX-RS) using Jersey - Tutorial
- <https://addons.mozilla.org/de/firefox/addon/restclient/>

### D.1. Apache Shindig

RPC vs. REST API for Shindig/OpenSocial: [http://groups.google.com/group/opensocial-and-gadgets-spec/browse\\_thread/thread/a4ddf7cd09f90237/5cfa1658e1c1d698?lnk=gst&q=rest#5cfa1658e1c1d698](http://groups.google.com/group/opensocial-and-gadgets-spec/browse_thread/thread/a4ddf7cd09f90237/5cfa1658e1c1d698?lnk=gst&q=rest#5cfa1658e1c1d698)  
[http://groups.google.com/group/opensocial-and-gadgets-spec/browse\\_thread/thread/d1a5627fb6e686ce/d27d47dee92a87b2](http://groups.google.com/group/opensocial-and-gadgets-spec/browse_thread/thread/d1a5627fb6e686ce/d27d47dee92a87b2) One argument was support for batching. A restful batching proposal didn't get consensus: [https://docs.google.com/View?docid=dc43mmng\\_23fdbpp7hd&pli=1](https://docs.google.com/View?docid=dc43mmng_23fdbpp7hd&pli=1)

Flow of REST requests in Shindig <https://sites.google.com/site/opensocialarticles/Home/shindig-rest-java>

Google+ is likely to become OpenSocial enabled: [http://groups.google.com/group/opensocial-and-gadgets-spec/browse\\_thread/thread/1187241df6759a9a](http://groups.google.com/group/opensocial-and-gadgets-spec/browse_thread/thread/1187241df6759a9a)

Shindig issues to implement OpenSocial 2.0 [https://docs.google.com/spreadsheet/ccc?key=0AihdZBncP3KzdGN3dVl3MFpIUlk2TXIyR3hfUDhHZUE&hl=en\\_US#gid=0](https://docs.google.com/spreadsheet/ccc?key=0AihdZBncP3KzdGN3dVl3MFpIUlk2TXIyR3hfUDhHZUE&hl=en_US#gid=0)

How Shindig supports extensions: <https://cwiki.apache.org/confluence/display/SHINDIG/Arbitrary+Extensions+to+Apache+Shindig%27s+Data+Model>

Videos about some 2.0 OS features [http://groups.google.com/group/opensocial-and-gadgets-spec/browse\\_thread/thread/7b911edfb1bb3b4d](http://groups.google.com/group/opensocial-and-gadgets-spec/browse_thread/thread/7b911edfb1bb3b4d)

OS and RDF [http://groups.google.com/group/opensocial-and-gadgets-spec/browse\\_thread/thread/20f62d627003509b](http://groups.google.com/group/opensocial-and-gadgets-spec/browse_thread/thread/20f62d627003509b)

OpenSocial Development Environment (OSDE, Eclipse Plugin) <https://sites.google.com/site/opensocialdevenv>

### D.2. Socialsite

Oracle's (former Sun's) extension to Apache Shindig. Blog <http://blogs.oracle.com/socialsite>

## E. TODO

- Does funambol.org has interesting implementations?