

# **Open Web Cooperation Schemes and Protocols**

**Respectful Friend,**

OMFICA (Open Market for Internet Content Accessibility) is a non-profit organization aimed at developing a competitive market for Internet companies. OMFICA has developed Open Web cooperation schemes and protocols which allow companies, carrying out internet services and statistic analysis, to establish mutually beneficial collaboration. This cooperation will radically change the trends in the internet field boosting evolution of internet technologies.

As an outcome of the integration of Open Web cooperation schemes and protocols, the entire internet content, brought to a semantic form which is interrelated with statistical data as an integral whole, becomes available for almost all Internet companies, as a result of which Internet becomes decentralized and controllable at the same time. The anticipated changes in the field will, without doubt, affect you and your company as well.

This document includes the descriptions of Open Web cooperation schemes and protocols which will help you to be aware of the future today and to get a deep insight about the possibilities of technologies which are beneficial to almost all existing companies in this field.

Welcome to the Future of World Wide Web!

Best Regards,  
OMFICA Team

<b>1. Overview</b> .....	<b>3</b>
<b>2. Open Web Cooperation Schemes</b> .....	<b>3</b>
<b>2.1. Crawling Synchronization and Data Exchange (CS DE) Based on Reciprocity</b> .....	<b>4</b>
2.1.1. CS DE abstract model .....	4
2.1.2. Crawling Synchronization (CS) .....	5
2.1.3. Data Exchange Based on Reciprocity .....	6
2.1.4. Reciprocity Score .....	7
<b>2.2. Open Web Statistics</b> .....	<b>8</b>
<b>2.3. Open Web Repository</b> .....	<b>9</b>
2.3.1. OWR Protocol .....	9
2.3.2. Open Web Repository Crawler Architecture .....	10
<b>2.4 Open Web Computer Resource Contribution (OW CRC)</b> .....	<b>12</b>
<b>2.5. Open Semantic Web (OSW)</b> .....	<b>13</b>
2.5.1. Entity Identification .....	13
2.5.2. Website Parse Template .....	14
<b>2.6. OWC Protocols Basics</b> .....	<b>20</b>
2.6.1. Packages .....	21
2.6.2. Security & Transport .....	21
2.6.3. Reciprocity Score .....	22
<b>3. OMFICA Open Source Products and Crawling Results</b> .....	<b>22</b>

## 1. Overview

OMFICA is a non-profit decentralized democratic system a member of which can become any professional. In order to become an OMFICA member, a professional individual or a company representative needs to fill in an application form providing contact information and detailed biography including educational background and professional experience. The application forms are viewed and approved by the OMFICA Trustees Committee after which the applicant becomes a valid member thus getting the opportunity to have influence on the company's activities, take part in project implementations and board elections both as a voter and a candidate.

OMFICA is working out and developing cooperating schemes, corresponding protocols and open source products that enable internet search providing companies:

- to collaborate for data exchange on the basis of reciprocity
- to increase the effectiveness of the expended resources
- to get the World Wide Web content archive from a single source easily
- to get the formal description of Internet content in RDF, UNDL/UNL, CWL and other applicable formats
- to favor the realization of the Semantic Web and web 3.0

OMFICA is an official member of W3C and actively collaborates with other companies that provide innovative technologies which promote the further advancement of Internet.

The cooperating schemes, protocols and open source products elaborated by OMFICA are open to everyone and can be utilized by OMFICA as well as third party companies for carrying out the cooperation coordination services.

## 2. Open Web Cooperation Schemes

Our recent research reveals that many Internet companies hold repositories of Internet website pages' crawled and parsed data (texts, images, video, etc), web page visit statistics and other publicly available documents/articles. Web pages crawled content is duplicated in several repositories and statistical data is incomplete per each repository. These repositories in most cases are proprietary and their access is reserved for owners' indexing or analyzing software tools.

OMFICA is developing cooperation schemes, hereafter referred to as Open Web Cooperation Schemes (OWC Schemes), for companies that provide Internet Services and conduct statistical analysis. Corresponding protocols, open-source products, and formats are elaborated by OMFICA to ensure the smooth operation of the OWC Scheme.

OWC Schemes are based on the analysis results of feedbacks, discussions and criticisms concerning the idea of OWC. It includes the overview of market's current situation, suggested architectural and synchronization models necessary for the implementation of OWC Schemes.

The main goal of OWC Schemes is to boost new technologies for analyzing of websites content and visit statistics, public documents, etc.

## **OWC Schemes defines the following general concepts:**

- Open Web Cooperation Coordinator (OWC Coordinator) is a protocol entity which at its realization becomes a server system securing the uninterrupted process of the cooperation.
- Open Web Cooperation Coordination Services (OWCC Services) ensure the coordination of OWC Scheme cooperating companies and can be provided by OMFICA, as well as any other commercial or non-profit company.
- Open Web Cooperation Coordination Service Provider (OWCC Service Provider) is the company providing OWCC Services. The same OWCC Service can be operated simultaneously by several OWCC Service Providers which will compete with each other.

## **OWC Schemes are the following:**

- Crawling Synchronization and Data Exchange (CSDE) Based on Reciprocity allows different companies that hold web content Repositories collaborate in the crawling process thus collecting a maximum amount of web content at the expense of minimum resources.
- Open Web Statistics (OWS) allows collaboration between organizations that collect statistics of web pages providing that maximum accurate results are gathered at the expense of minimum resources.
- Open Web Repository (OWR) allows organizations to have access to WWW entire content from a single source paying for the computing resources only, at the same time reducing this expenditure by means of collaboration with companies interested in the same content.
- Open Web Computer Resource Contribution (OW CRC) allows volunteers to provide their computer resources for carrying out public-oriented activities, research and investigation by different institutions and non-profit organizations.
- Open Semantic Web, (OSW,) permits to elaborate the Internet content into a semantically modified form of computer languages interrelating and integrating them into a comprehensive whole.

Benefits of OWC Schemes are efficiency of crawling resources utilization, website statistical data acquisition, and creation of open repositories, the integration of the volunteers' computer resources and their effective use, along with the opportunity to make the Semantic Web a reality.

Active discussions and conferences have been periodically organized over the past few years concerning the creation of Open Repositories and open network system for field development purposes. More information is available at [www.openrepositories.org](http://www.openrepositories.org).

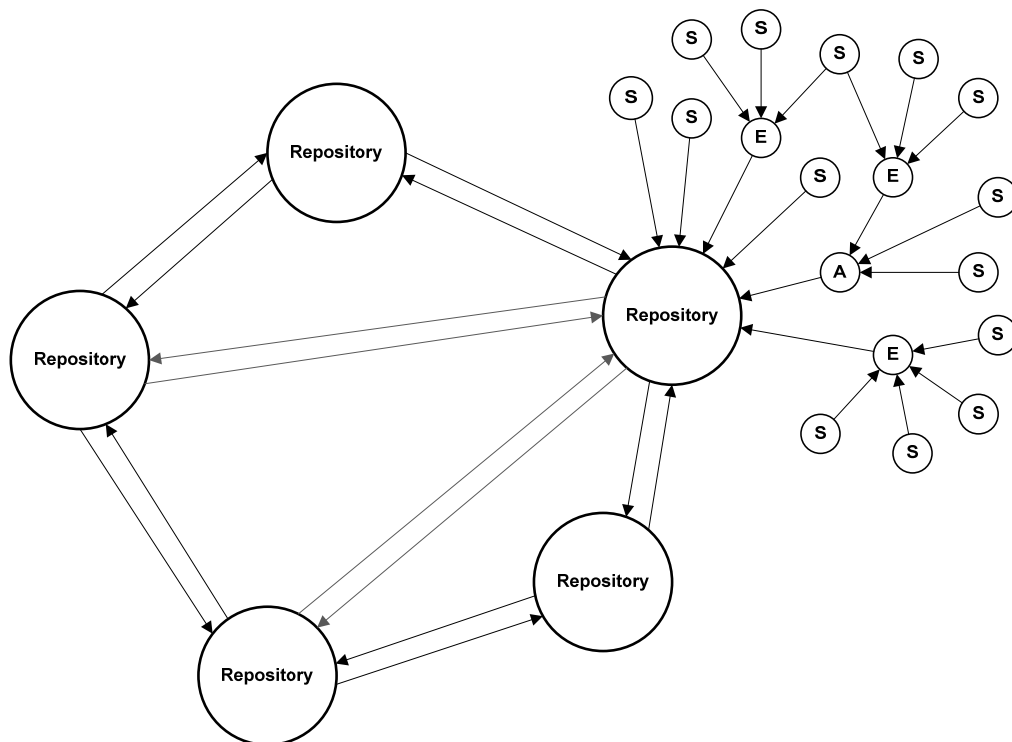
## ***2.1. Crawling Synchronization and Data Exchange (CS DE) Based on Reciprocity***

### **2.1.1. CS DE abstract model**

CS DE abstract model defines entities and describes the procedures necessary for Open Data collection, synchronization and exchange between multiple repositories. Open Data is a record (document, article, etc.) that does not contain any secure or private information.

OWC defines the following entities:

- **Source** - declares, distributes and publishes Open Data for commercial, legal and/or social purposes (e.g. website).
- **Extractor** - extracts data from multiple Sources (e.g. web crawler).
- **Aggregator** - collects data from Extractors and/or Sources and sends to Repositories.
- **Repository** - forms collected Open Data archives.
- **User** - retrieves and uses Open Data.
- **Coordinator** - coordinates Open Data sharing and crawling synchronization processes.
- **Registrar** - registers entities by assigning IDs to them and storing their profiles.
- **Dashboard** - directory available to all entities for read/write operations.



## 2.1.2. Crawling Synchronization (CS)

Crawling Synchronization (CS) is the business process sequence flow which enables to share crawling tasks between different Repositories. CS allows Repositories to crawl different Sources and exchange extracted content, thereby increases efficiency of resource sharing and data exchange. It declares the following concepts and defines corresponding package types (simple or hierarchical) for them: Source Profile, Source Group and Group Profile.

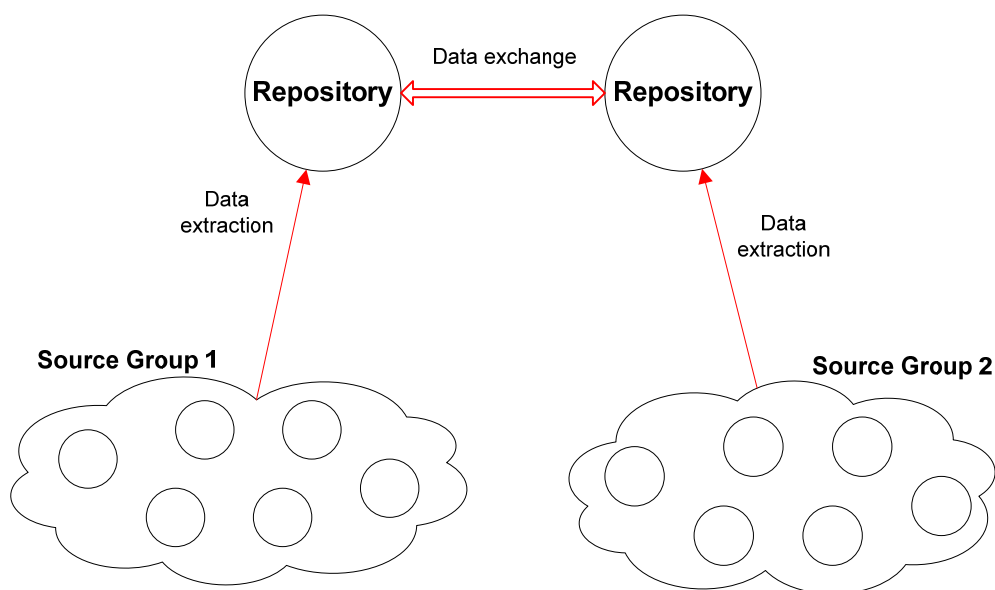
Source Profile stores Source content characteristics. For example, if Source is a website, Profile may contain the following fields:

- size of website in bytes
- number of pages
- average size of one page in bytes
- number of images and its proportion to number of pages
- number of PDF files and its proportion to number of pages
- number of SWF files and its proportion to number of pages

Coordinator receives Profiles for the same Source from several Repositories and generates integrated accurate Source Profile. Cluster analysis tools are being used to generate Sources Groups based on Source Profiles characteristics. Coordinator assigns unique ID to each Group, generates Group Profile and places those data on Dashboard.

CS defines package types which allow Repositories to publish the following information:

- list of available crawled Source Groups
- periodicity of content updates
- list of necessary Source Groups



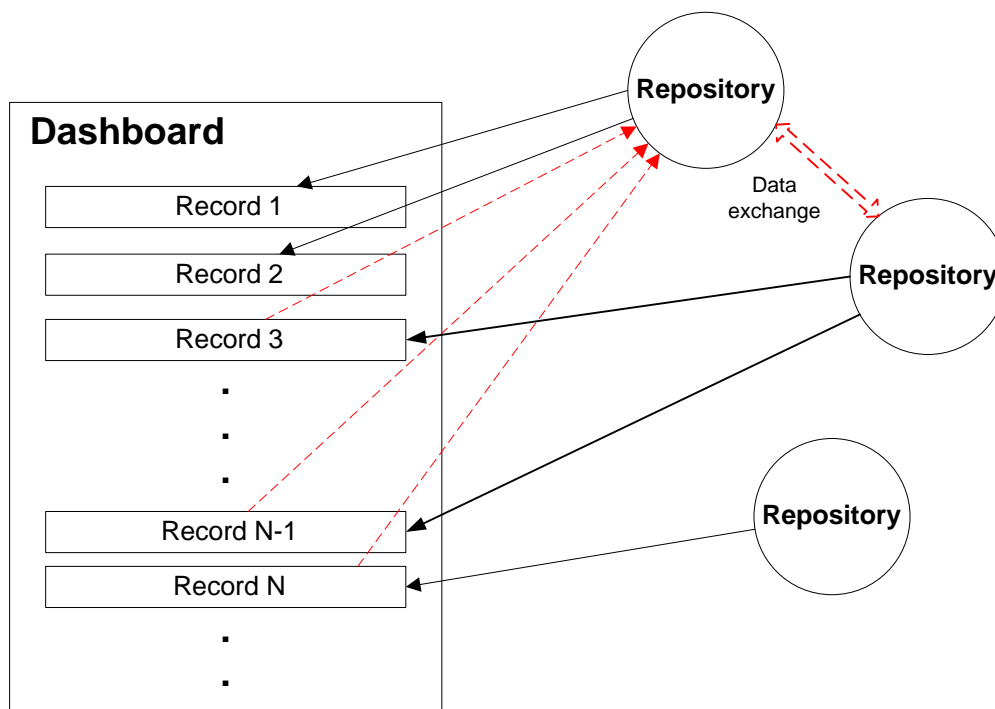
Data exchange between Repositories is implemented on the basis of reciprocity which is described in the following chapter.

### 2.1.3. Data Exchange Based on Reciprocity

DE based on reciprocity enables to organize available data exchange between Repositories. It relies on OWC Coordinator, Registrar and Repository entities. DE assumes data exchange procedure based on “publication” operation. Coordinator and Repositories may “publish” any record on Dashboard using corresponding packages.

DE defines package types which allow Repositories to publish the following information:

- list of available content’s Sources
- periodicity of content updates
- list of necessary Sources



Published information is available to other Repositories. Any Repository may find records of interest and send corresponding data exchange suggestion package to a Repository which holds necessary Sources' content. Responding Repository may accept/reject received suggestion or send another suggestion package to requestor. As soon as responding side accepts suggestion, Repositories exchange data and notify Coordinator about successful/failed communication. Coordinator logs communication history and recalculates Reciprocity Scores for those Repositories.

### **2.1.4. Reciprocity Score**

OWC does not imply direct regulation of the communication between entities. OWC only encourages but does not oblige entities to notify the Coordinator about received and sent packages. Coordinator logs received information details into the communications history directory of corresponding entity on Dashboard.

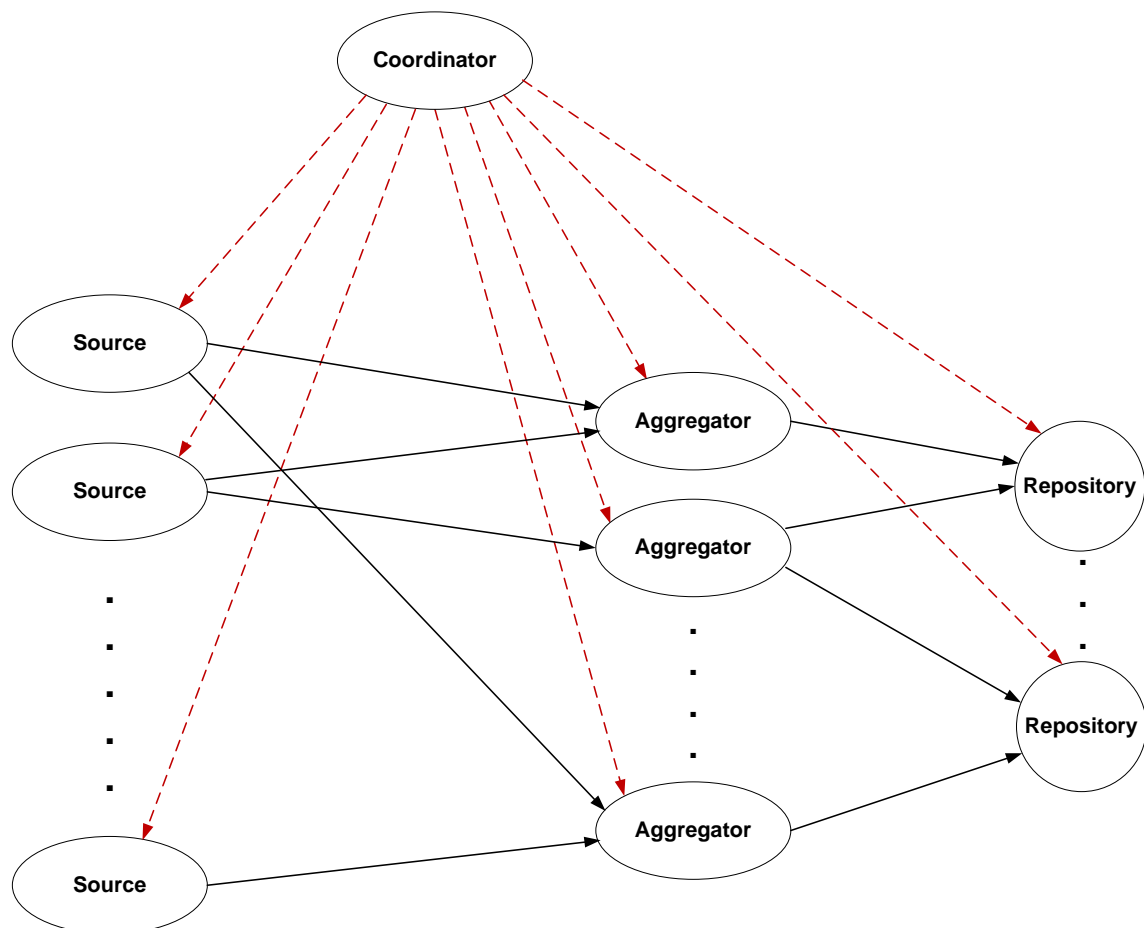
Each entity should notify Coordinator about readiness to provide action logs or not. Based on history logs, Coordinator periodically calculates Reciprocity Scores for all entities. If specific entity does not notify about readiness to provide any communication history or notifies but does not perform, that entity's Reciprocity score will be negative.

## 2.2. Open Web Statistics

Open Web Statistics (OWS) is the second scheme of OWC which enables Sources to publish their usage statistics to all interested entities. It is implemented by the following entities: Coordinator, Source, Aggregator and Repository, detailed descriptions of which are provided in Chapter 2.1.1.

Each Source requests Coordinator for the list of Aggregators to submit its visit statistics. Coordinator selects Repositories which are interested in that Source statistics and works out Aggregators' list to be sent as response. Coordinator may rearrange Aggregators based on requests received from Repositories.

Source submits visitors' action logs (referral URL, clicks, downloads, outcome URL, etc.) to assigned Aggregators which submit received information to corresponding Repositories.



Source visit statistics are provided by package with information on referral Source, visited Source and outcome Source, so that having visit statistics from all Sources it becomes easier to compare and validate provided information.

## **2.3. Open Web Repository**

Open Web Repository (OWR) is the third scheme of OWC which enables companies to provide their computing resources to Open Repositories for receiving necessary crawled web data in return. These companies in turn minimize their expenditure for receiving web data by means of collaboration with other companies interested in the same content.

OWR assumes two roles:

- Crawling Resource Contributor, which provides operational resources and Internet bandwidth.
- Open Repository Beneficiary, which gets access to Open Repositories content.

OWR undertakes financial reimbursement scheme for Crawling Resource Contributors. Open Repository Beneficiaries pay for computing resource utilization. The same entities may act in two roles at the same time. In this case, provided computing resources may be reimbursed by received data. The main principle of OWR is paying for servers' crawling resource utilization only.

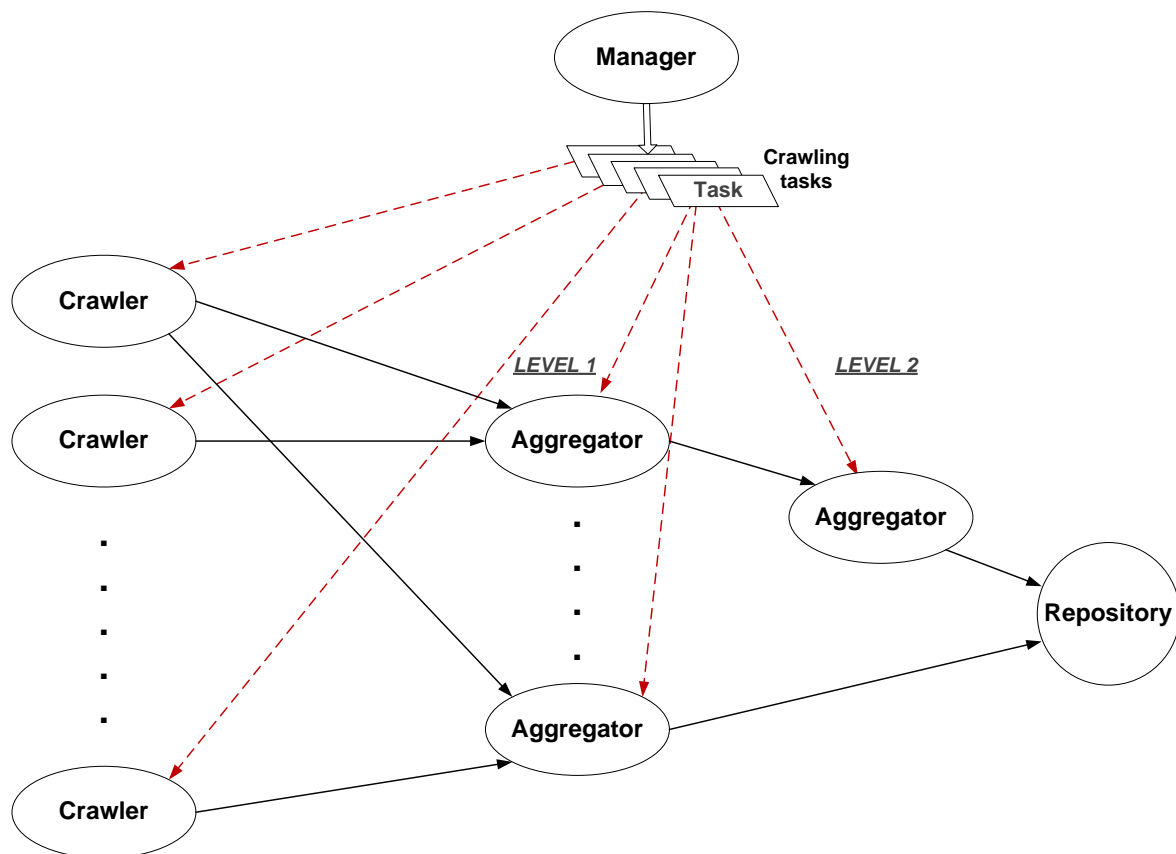
Contributed computing resources will be used for open crawling framework which can either be an open-source product or belong to all Crawling Resource Contributors and Open Repository Beneficiaries.

### **2.3.1. OWR Protocol**

One of the OWR protocol's basic concepts is a Crawling Task - a record which contains crawling details: targeted host name, URL patterns, crawling start and end schedule, list of aggregators to submit crawling results, crawling methods and plug-ins.

OWR protocol defines four types of entities which are:

- Crawler/Extractor
- Aggregator
- Manager
- Repository



Crawlers notify the Manager about available free computing resources such as network connection bandwidth, hard drive free space, RAM and CPU. Based on received information from Crawlers, Manager assigns them corresponding Crawling Tasks. Crawler executes Crawling Task and submits crawling results to specified Aggregators along with computing resources spent.

Aggregators form a hierarchical network to assure high level of extracted data accuracy. Aggregators get crawling results from Crawlers, verify received data and submit to the next level Aggregators or directly to Repository.

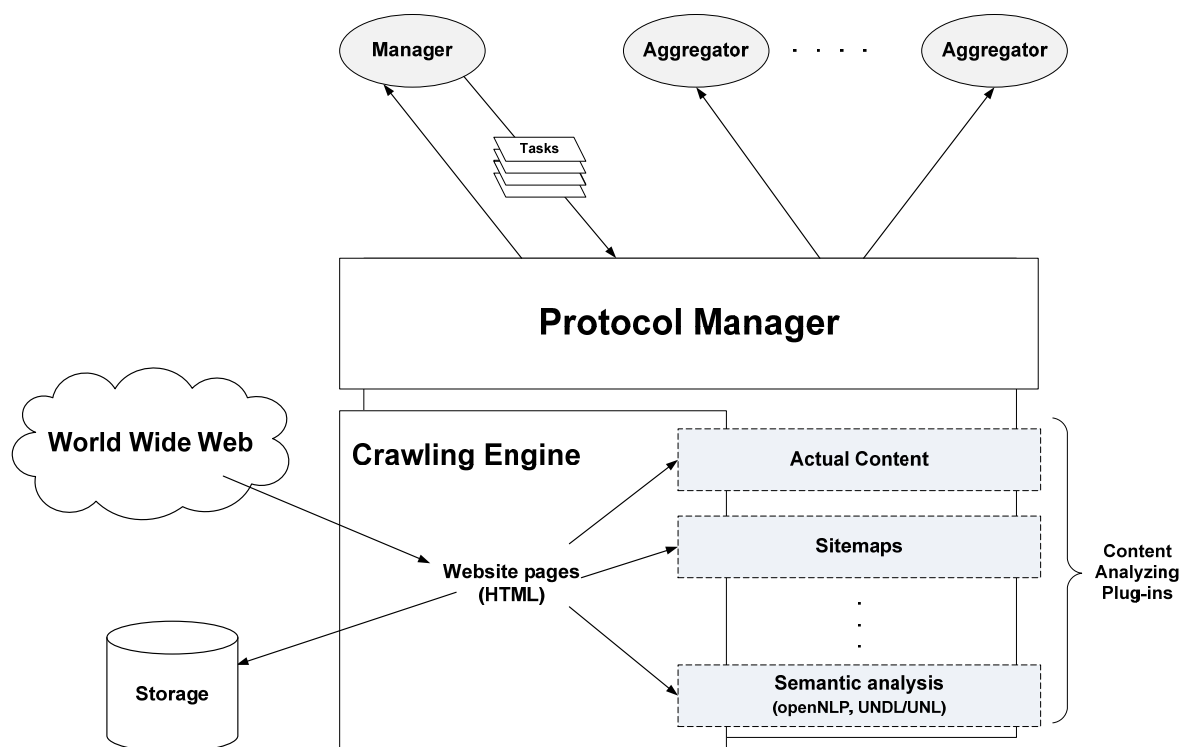
Manager's main purpose is to link and co-ordinate Crawlers and Aggregators (hereinafter "nodes"). Manager registers the nodes by assigning them unique IDs. Manager detects the optimal topological structure of the nodes, and sends corresponding Tasks to Crawlers and Aggregators.

### **2.3.2. Open Web Repository Crawler Architecture**

Open Web Repository Crawler Architecture (OWRCA) is pluggable framework architecture of web crawling client application.

OWRCA consists of four main parts:

- Protocol Manager
- Crawling Engine
- Analyzing Plug-ins
- Spent Resource Calculator



Protocol Manager makes communication possible between Crawler and other entities (e.g. Manager and Aggregators). Protocol Manager gets Crawling Tasks from OWR Manager and starts crawling processes in accordance with given schedule and mode.

Crawling Engine is a multi-thread system with parametric number of threads which is responsible for crawling algorithm realization. It downloads website pages, detects internal and external links, stores HTML content in the storage and activates content analyzing plug-ins if any.

Crawling Engine has two modes: simple and extended. The main difference between these modes is that in case of extended mode Crawling Engine embeds Mozilla Firefox web browser and Mozilla Firefox plug-ins to interpret and analyze JavaScript codes.

Content analyzing plug-ins perform additional analysis of downloaded content and submit results into a storage with certain format. For instance, OWC allows integrating open text analyzing tools into crawling client application, such as Open Natural Language Processing (OpenNLP) and UNDL/UNL generator.

Spent Resource Calculator is responsible for calculation of computing resources spent on each Crawling Task execution per each and all sub-modules. Calculated data is converted into a package with corresponding format to be sent to a Manager.

## 2.4 Open Web Computer Resource Contribution (OW CRC)

Open Web Computer Resource Contribution (OW CRC) is a cooperation model which allows organizations and research institutions to involve volunteers' computing resources for non-profit purposes,

OW CRC defines a basic scheme for implementing a special Computer Resource Contribution Protocol (CRCP), which Abstract Model includes the following concepts:

- **CRC Beneficiary Organization** – a company aimed at a non-profit cast, an independent professional, commercial structure or research institution that provides specific software solution, which is operated on volunteers' computing resources.
- **CRC Service** – a calculation process in a specific field of research, where a given CRC Beneficiary Organization is involved. Each service is utilized by a server and a proper software support.
- **CRC Service Coordinator** – a server, provided by a CRC Beneficiary Organization for coordinating the data exchange between its service and the computing resource provided by a CRC Volunteer.
- **CRC Component** - a software or another computing solution that allows a CRC Beneficiary Organization to implement and realize the given CRC Service. CRC Component can be considered as a specific plug-in, which may be either an open source product or a secure software.
- **Submission** - CRC Beneficiary Organization undertakes a submission process, which is performed through an application form.
- **Registrar** – a team based on moral and democratic principles, responsible for particular review and examination of all submitted application forms.
- **Archive** – besides standing in the Registrar queue, any application form submitted by an organization, automatically transfers to the main Archive, which in turn consists of two parts – Active List and Refuse List.
- **Active List** – a storage that serves as a data base of all the active CRC Services that are approved and can be operated on the volunteers' computing resources. *Active List* is being synchronized (a part of the CRC Synchronization), each time a new member joins or vacates it. Active List can be accessed both by all the current members, and the Volunteers.
- **Refuse List** – a storage that includes records that have been submitted by the users but were not approved by the Registrar. Access to *Refuse List* is strictly private and available only for the members of the Registrar itself.

- **CRC Volunteer** – an individual or an organization that provides computing resources, which may include both computing and networking resources as well as the proper memory storage for the above-mentioned resources. A volunteer grants his/its resource in order to make the latter a part of the common CRC Network, acquiring a promotion of his/its contribution in return.
- **CRC Network** - a network system based on integrated models of the standard client-server and P2P architectures. CRC Network connects the computing systems of the volunteers, CRC Coordinator services as well as the CRC Beneficiary Organizations' services, providing the necessary data exchange and synchronization process.
- **CRC Synchronization** – a process of synchronizing and updating the actual information. All changes that take place between different CRC entities within CRC Network, are being updated in real-time mode and shared between the CRC Network members.
- **CRC Framework** – a fundamental pluggable computing system, which performs CRC Component synchronization with the given CRC Service Coordinator. CRC Framework synchronization process is performed simultaneously on all the frameworks of the entire CRC Network.
- **CRC Coordinator** – a governing body that regulates and coordinates the whole process of the data exchange and cooperation between different CRC services within the entire CRC network.

## 2.5. Open Semantic Web (OSW)

Open Semantic Web Cooperation Scheme enables website publishers, providers of search services and texts' syntactic semantic analyzing services, to establish cooperation which will result in launching the Semantic Web.

Open Semantic Web includes technologies, open-source products, protocols, formats, and URI schemes which allow:

- to generate and analyze the formal descriptions of websites as a consequence of the human-machine dialogue
- to generate respective RDF, CWL, UNL format files
- to ensure the connection between the websites and generated files by means of WPT format defined in Chapter 2.5.2.

### 2.5.1. Entity Identification

The new URI Scheme defined by the OSW permits to identify any entity which can be an individual, organization, equipment, product, technology, concept, or object

Entities' profiles stored on Registrars are publicly available information. OSW defines the following URI scheme to access interested entities' profiles.

`oweid://registrar/entity`

where “oweid” (open web entity id) is the scheme name that refers to OSW specification; “registrar” is the authority component for the URI hierarchy that indicates certain Registrar storing necessary specific entity profile; and the remainder part - “entity” is an entity unique identifier assigned by Registrar which corresponds to URI path.

The new “ow:entity” XML tag of XML Domain defined by OSW allows providing a unique Identifier for each object described in the document, thus clarifying the exact object which is being referred to.

<ow:entity> has the following syntax:

```
<ow:entity id= “oweid://registrar/entity”>  
.....  
</ow:entity>
```

“id” refers to the entity/object which name is enclosed within <ow:entity> and </ow:entity> tags. It allows XML parser to detect specific entity even if it is described in an unusual form (e.g. instead of OMFICA it is indicated as OMFICA.org).

The new “ow:entity” XML tag defined by OSW permits to single out and clarify the official information provided by the entity.

<ow:copyright> has the following syntax:

```
<ow:copyright owner= “oweid://registrar/entity”>  
.....  
</ow:copyright>
```

“owner” refers to the entity which holds copyright for the content enclosed within <ow:copyright> and </ow:copyright> tags. This mechanism allows Copyright Office Repositories to register copyright automatically and track possible cheating.

## 2.5.2. Website Parse Template

**Website Parse Template (WPT)** is an XML based open format which provides web crawlers with additional proper information on web page HTML elements and content. WPT is compatible with XML schemas, such as RDF and OWL.

Website Parse Template uses XML tags of Open Web vocabulary which is being declared as an XML namespace: *xmlns:ow="http://www.omfica.org/schemas/ow/0.9"*.

Website Parse Template begins with opening <ow:wpt> tag and ends with closing </ow:wpt> tag. Single Website Parse Template is referred to the same host, while single host may have several Website Parse Templates describing its HTML structure. It is required to specify host name the Website Parse Template is for and declare the namespace within <ow:wpt> tag (see example below).

## Example 1. Website Parse Template frame

```
<?xml version="1.0" encoding="UTF-8"?>
<ow:wpt xmlns:ow="http://www.omfica.org/schemas/ow/0.9"
        ow:host="http://www.example.com">
.....
</ow:wpt>
```

Website Parse Template consists of following sections:

- **Templates** is a mandatory section, which contains web pages' HTML structure and content description.
- **URLs** is an optional section, which links URL Patterns for groups of web pages to specified Templates.
- **Ontology** is an optional section which defines concepts and relations used in website.

## Templates

Templates section describes web pages' structured content by referring to corresponding structural elements of specific pages.

Template starts with opening `<ow:template>` tag and ends with closing `</ow:template>` tag. It is required to specify unique template name within `<ow:template>` tag and define URL which complies with specific template.

Template consists of blocks corresponding to each structural element of specific web page. Each template must contain at least one block. Block makes reference to appropriate HTML element through one or any combination of following reference methods: TagID, XPath and Pattern. Each block must start with opening `<ow:block>` tag and correspondingly ends with closing `</ow:block>` tag. It is required to indicate specific HTML element reference(s) within block's opening tag.

## Example 2. Block reference methods

```
<ow:template ow:name="Template Example" ow:url="http://www.example.com/index.php">
.....
  <ow:block ow:tagid="ex1" ow:xpath="/html/body/div/div" ow:pattern="content (object[[a-z]*])">
..... //content description
  </ow:block>
  <ow:block ow:tagid="ex2">
..... //content description
  </ow:block>
  <ow:block ow:xpath="/html/body/div/div/table/tr[1]/td">
..... //content description
  </ow:block>
.....
</ow:template>
```

Each block contains specific HTML element's content description represented solely or within another block. Embedded blocks are used to describe specific HTML element ("*parent block*") which includes one or more elements ("*child block*").

### Example 3. Embedded blocks

```
<ow:template ow:name="Template Example" ow:url="http://www.example.com/index.php">
.....
  <ow:block ow:xpath="/html/body/div/div">
    <ow:block ow:xpath="/html/body/div/div/table/tbody/tr[1]/td">
      ..... //content description
    </ow:block>
  </ow:block>
.....
</ow:template>
```

Content description can be provided using concepts defined in Ontology section or any supported format/language: RDF, CWL, etc. It is required to declare namespaces of used XML schema(s) within `<ow:wpt>` tag and ontology name within `<ow:template>` tag.

### Example 4. Content description instances

```
<ow:wpt xmlns:ow="http://www.omfica.org/schemas/ow/0.9"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  ow:host="http://www.example.com">
.....
  <ow:template ow:name="Template Example" ow:url="http://www.example.com/index.php"
    ow:ontology="ontology_example">
    .....
    //Content description using defined concepts
    <ow:block ow:tagid="ex1" ow:xpath="/html/body/div/div" ow:pattern="Wellcome (user.name[[A-Za-z]*])">
      ontology_concept
    </ow:block>
    //Content description using RDF syntax
    <ow:block ow:tagid="ex2">
      <rdf:Description rdf:about="http://www.example.com/index.php">
        .....
      </rdf:Description>
    </ow:block>
    //Content description using CWL.unl
    <ow:block ow:xpath="/html/body/div/div/table/tr[1]/td">
      {cwl.unl}
      .....
      {/cwl.unl}
    </ow:block>
    .....
  </ow:template>
.....
</ow:wpt>
```

If the web page contains listed or structured repeatable content it can be represented as a single entry by specifying block type as repeatable. For example, if specific HTML element repeats several times, and the content category remains the same, it can be described as a repeatable block instead of specifying blocks for each element. In most cases repeatable blocks are child blocks embedded within parent block – another HTML element.

### Example 5. Repeatable content representation

```
<ow:template ow:name="Template Example" ow:url="http://www.example.com/index.php">
.....
<ow:block ow:xpath="/html/body/div/table/tbody/tr/td[2]" ow:type="repeatable">
..... //content description
</ow:block>
<ow:block ow:xpath="/html/body/div/div">
  <ow:block ow:xpath="/html/body/div/div/table/tbody/tr[1]/td" ow:type="repeatable">
    ..... //content description
  </ow:block>
</ow:block>
.....
</ow:template>
```

For Patterns' definition WPT uses Regular Expressions which identify HTML markup or textual content segments of specific web page. WPT defines the following scheme for regexp patterns: (*object[regexp pattern]*), where "object" specifies type/category of specific textual segment. Object must be declared in Ontology section.

### Example 6. Pattern references

```
<ow:template ow:name="Template Example" ow:url="http://www.example.com/index.php"
  ow:ontology="ontology_example">
.....
  <ow:block ow:pattern="<td bgcolor=\\"FFFFFF\\" class=\\"small\\"><a href=\\"/ar-(artist.id[[0-9]*])---
  (artist.name[[A-Za-z]*])\\" class=\\"small\\"><b>(artist.name[[A-Za-z]*])" ow:type="repeatable">
  ..... //content description
  </ow:block>
  <ow:block ow:pattern="The Best Music Blogs on the Web">
  ..... //content description
  </ow:block>
  <ow:block ow:pattern="Wellcome (member.name[[A-Za-z0-9]*])! Your member ID is (member.id[[0-9]*]). >
  ..... //content description
  </ow:block>
.....
</ow:template>
```

Single template may describe single web page or a group of similarly structured web pages. In the examples above template refers to a single web page (e.g. ow:url="http://www.example.com/index.php"). In case of describing group of similarly structured web pages via single template it is necessary to define URL pattern covering all of those pages (e.g. ow:url="http://www.example.com/(page.name[[a-z]\*]).php).

### Example 7. Template for a Single Artist Page on Yahoo! Music

```
<?xml version="1.0" encoding="UTF-8"?>
<ow:wpt xmlns:ow="http://www.omfica.org/schemas/ow/0.9"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  ow:host="http://music.yahoo.com">
.....
  <ow:template ow:name="Artist Page on Yahoo! Music"
    ow:url="http://music.yahoo.com/ar-(artist.id[[0-9]*])---(artist.name[[A-Z,a-z,-,0-9]*])"
    ow:ontology="general">
```

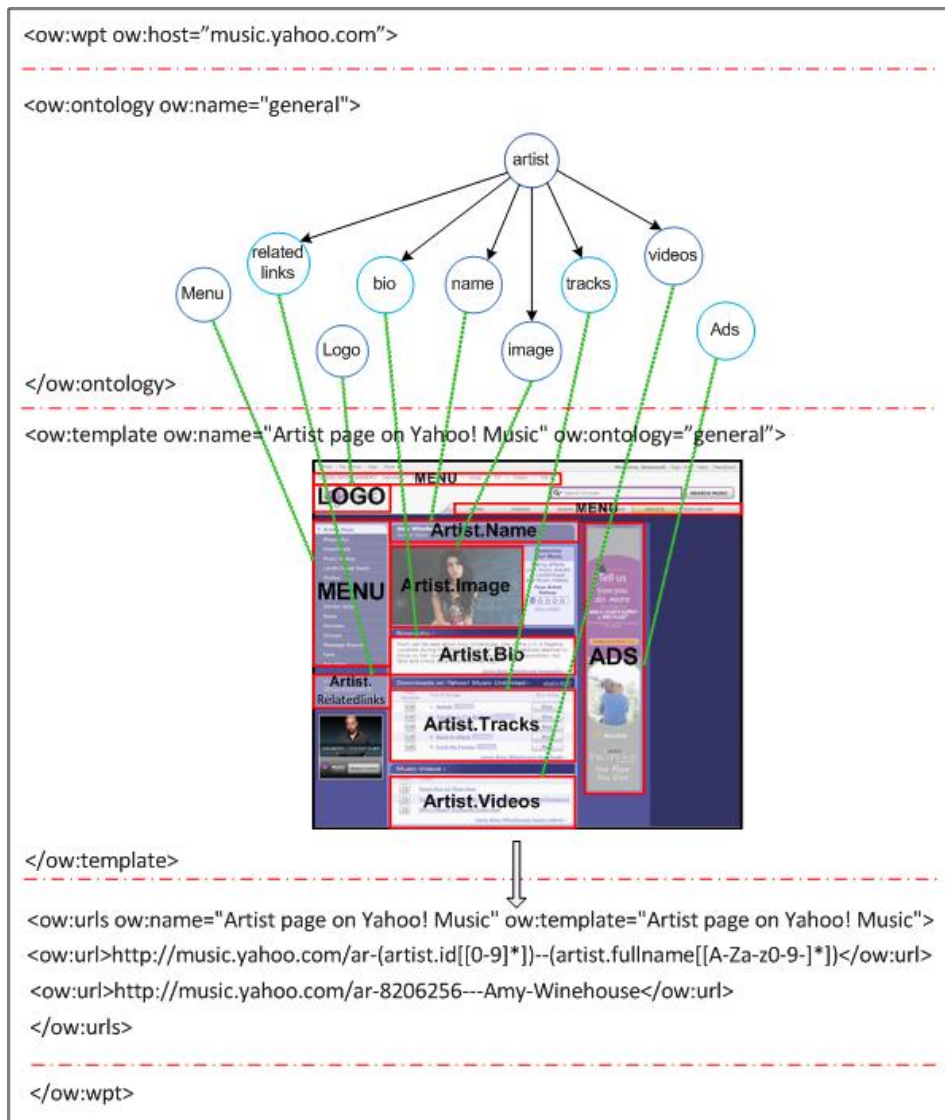
```

<ow:block ow:tagid="yent-uhdr">Menu</ow:block>
<ow:block ow:xpath="/html/body/div[2]/div/div/div[3]/div/a/span">Logo</ow:block>
<ow:block ow:xpath="/html/body/div/div">Advertisement</ow:block>
<ow:block ow:xpath="/html/body/div[3]/table/tbody/tr/td[2]/div/h1">artist.name</ow:block>
<ow:block ow:tagid="art_img">artist.image</ow:block>
<ow:block ow:tagid="biography">artist.bio</ow:block>
<ow:block ow:xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[22]">artist.album</ow:block>
<ow:block ow:xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[10]">artist.track</ow:block>
<ow:block ow:xpath="/html/body/div[3]/table/tbody/tr/td[2]/table/tbody/tr[13]">artist.video</ow:block>
</ow:template>
.....
</ow:wpt>

```

Visual representation of Website Parse Template (see figure 1 below) reveals ontology concepts connections with web page HTML elements.

**Figure 1. WPT visual representation**



## URLs section

This section defines the URLs/URL patterns of web pages described in Templates section. This section is mandatory if the templates do not define URLs/URL patterns of web pages.

In accordance with Templates section this section also may consist of several blocks/units. Either of those blocks starts with `<ow:urls>` tag and ends with `</ow:urls>` tag.

### Example 8. URL patterns

```
<ow:urls ow:name="Artist page on Yahoo! Music" ow:template="Artist page on Yahoo! Music">
  <ow:url>http://music.yahoo.com/ar-8206256---Amy-Winehouse</ow:url>
  <ow:url>http://music.yahoo.com/ar-([artist.id[0-9]*])---(artist.name[[A-Za-z0-9-]*])</ow:url>
</urls>
```

As a URL block's name can be chosen any string, but for the template it is necessary to indicate specific template name described in previous section.

RegExp specifications are used for URL patterns descriptions. The URL pattern provided in *Example 8* also includes the represented real URL. The concepts necessary for URL pattern definition (such as "id" and "name") are to be defined in Ontology section.

## WPT Ontology

Ontology section contains enumeration and definition of all concepts used in website. Listed concepts must be enclosed between `<ow:ontology>` and `</ow:ontology>` tags. It is required to specify the ontology name (any rational string) within `<ow:ontology>` tag. WPT allows using of either OWL or WPT Ontology language for concepts definition. The main difference between those languages is that WPT Ontology language provides simplified syntax for concepts and relations definition.

### Example 9. "artist" concept definition using WPT Ontology language

```
<ow:ontology ow:name="general">
  <ow:concept ow:name="artist">
    <ow:inherit>person</ow:inherit>
    <ow:has>name</ow:has>
    <ow:has>album</ow:has>
    <ow:has>track</ow:has>
    <ow:has>image</ow:has>
    <ow:has>bio</ow:has>
    <ow:has>video</ow:has>
    <ow:has>id</ow:has>
  </ow:concept>
  <ow:concept>logo</ow:concept>
  <ow:concept>menu</ow:concept>
  <ow:concept>advertisement</ow:concept>
</ow:ontology>
```

Each concept definition starts with `<ow:concept>` tag and ends with `</ow:concept>` tag. `<ow:inherit>` tag shows inheritance relations and `<ow:has>` tag shows attributable relations between two concepts.

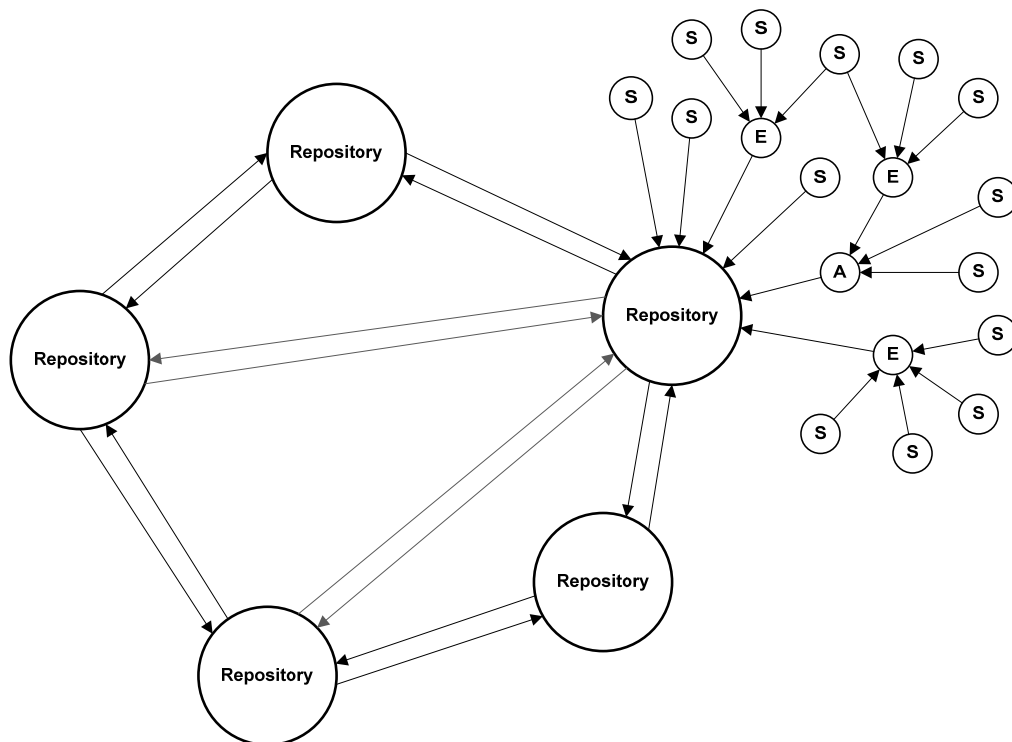
Either of defined concepts has default attribute - object identifier (**id**) to be used by web crawlers to coordinate the same object's attributes used in different pages of the same website.

## 2.6. OWC Protocols Basics

For OWC Schemes realizations, a couple of Protocol families are defined, which are based upon general approaches, concepts, package regulation, transportation and security systems.

OWC Protocol defines the following entities:

- **Open Data** - record (document, article, etc.) that does not contain any secure or private information.
- **Source** - declares, distributes and publishes Open Data for commercial, legal and/or social purposes (e.g. website).
- **Extractor** - extracts data from multiple Sources (e.g. web crawler).
- **Aggregator** - collects data from Extractors and/or Sources and sends to Repositories.
- **Repository** - forms collected Open Data archives.
- **User** - retrieves and uses Open Data.
- **Coordinator** - coordinates Open Data sharing and crawling synchronization processes.
- **Registrar** - registers entities by assigning IDs to them and storing their profiles.
- **Manager** - controls crawling procedure for single Repository.
- **Dashboard** - directory available to all entities for read/write operations.



OWC Protocol supposes four possible data collection chains:

- Sources send Open Data to Repository directly;
- Sources send Open Data to Repository through Aggregators;
- Extractor gets Open Data from Sources and sends to Repository directly;
- Extractor gets Open Data from Sources and sends to Repository through Aggregators.

OWC Protocol model implies verification procedure which is aimed to assure data extraction and aggregation correctness.

## **2.6.1. Packages**

OWC defines abstract model of Packages for data transmission between OWC entities. Package is an associative array which elements are either simple values or packages (hierarchical packages). OWC defines interface to ensure package processing. According to defined interface, packages are being serialized to be recorded or transmitted via network connection, and recipient de-serializes received stream to process data sent in the package. Packages abstraction and serialization increases efficiency of hierarchic data exchange for both binary and textual data.

Each package has at least three elements: package ID, sender ID and package type.

Package ID is a unique number that consists of following parts: creator ID, generator ID and sequential number. Creator ID is the identifier assigned to the entity which has created specific package. In most cases it matches with sender ID. Generator ID is the identification number of the module generated specific package. Sequential number is an automatically generated sequentially increasing number assigned to each package.

Sender ID is the identifier assigned to the entity which has sent specific package.

Package types are being defined by each OWC scheme separately in accordance with specified communications between entities. Depending on type, the package may have additional elements which number and type are not limited.

Serialized data stream consists of two fields: serialization type and body, where type indicates serialization mode and body contains the serialized package. Serialization type has the following structure: general-mode/sub-mode.

## **2.6.2. Security & Transport**

OWC uses asymmetric cryptography to encrypt serialized packages which assumes two approaches: digital signatures and public key encryption. To assure high level of privacy and security, OWC implements combination of asymmetric cryptography approaches. It helps to ensure authenticity of the sender and confidentiality of sent data simultaneously.

Any entity randomly generates two pairs of public and private keys: one pair for packages encryption and one for packages decryption. Encryption and decryption public keys are submitted to Dashboard for other entities usage.

OWC assumes two possible ways of package encryption depending on specified communication features:

- sender encrypts a package using its encryption private key only (digital signature);
- sender encrypts a package firstly by its digital signature, and secondly using recipient's decryption public key, so that recipient needs to decrypt the package using its decryption private key and sender's encryption public key.

OWRSA security level adds security head to encrypted serialized packages.

OWC does not define any specific protocol or approach for packages transmission. It can be implemented using TCP, UDP, HTTP or any other protocol.

### **2.6.3. Reciprocity Score**

OWC does not imply direct regulation of the communication between entities. OWC only encourages but does not oblige entities to notify the Coordinator about received and sent packages. Coordinator logs received information details into the communications history directory of corresponding entity on Dashboard.

Each entity should notify Coordinator about readiness to provide action logs or not. Based on history logs, Coordinator periodically calculates Reciprocity Scores for all entities. If specific entity does not notify about readiness to provide any communication history or notifies but does not perform, that entity's Reciprocity score will be negative.

## **3. OMFICA Open Source Products and Crawling Results**

The OMFICA development team has developed, tested, and deployed OMFICA Open-Source Web Crawler 1.7 version which is currently operating and is collecting the World Wide Web content.

Crawler binaries are available at:

[http://www.omfica.org/docs/opensources/omfica\\_crawler.1.7.bin.win32.zip](http://www.omfica.org/docs/opensources/omfica_crawler.1.7.bin.win32.zip).

Source codes are available at:

[http://www.omfica.org/docs/opensources/omfica\\_crawler.1.7.src.win32.zip](http://www.omfica.org/docs/opensources/omfica_crawler.1.7.src.win32.zip).

The 2.1 version has also been developed. Although not recommended yet, the source codes are available for downloading at:

[http://www.omfica.org/docs/opensources/omfica\\_crawler.2.1.src.win32.zip](http://www.omfica.org/docs/opensources/omfica_crawler.2.1.src.win32.zip).

OMFICA Crawler - v2.1 is going to be embedded with Firefox Mozilla browser, which will allow increasing the accuracy grade of crawling results.

Crawling results are being converted into specific packages before being transmitted to interested entities. You can download a sample of crawled data package here:

[http://www.omfica.org/docs/omfica\\_sample\\_package.zip](http://www.omfica.org/docs/omfica_sample_package.zip).

Crawling results are converted into packages before being transmitted or stored. The package is a TLV stream. TLV is an abbreviation for Type Length Value.

The first four bytes of TLV (**Type**) contain transmitted data type. There are assumed the following data types:

- 1 – **Hostname**
- 2 – **Website Sitemaps**
- 3 – **Website Parse Template (WPT)**
- 4 – **URL**
- 5 – **HTML Scripts**
- 6 – **Web Page Internal & External Links**
- 7 – **Web Page Extracted Content**
- 8 – **Web Page Parse Result**
- 9 – **External Hostnames**

The next four bytes (**Length**) contain size of transmitted data included in the remaining part of TLV (**Value**).

Each package consists of the following sequences of TLVs:

TLV 1 (type=1)	TLV 2 (type=2)	TLV 3 (type=3)	TLV 4 (type=4)	TLV 5 (type=7)	TLV 6 (type=5)	TLV 7 (type=8)	TLV 8 (type=6)	TLV 9 (type=9)
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Sequence	Data Type	Description
TLV 1	Hostname	Hostname the crawling has been performed for.
TLV 2	Website Sitemaps	Sitemap of corresponding host generated by OMFICA Crawler.
TLV 3	Website Parse Template (WPT)	WPT file describing specific web page.
TLV 4	URL	URL of specific web page.
TLV 5	Web Page Extracted Content	Actual content of specific web page detected by OMFICA Crawler.
TLV 6	HTML Scripts	HTML file of specific web page.
TLV 7	Web Page Parse Result	Structured data of specific web page extracted by OMFICA Crawler.
TLV 8	Web Page Internal & External Links	Internal and external links of specific web page.
TLV 9	External Hostnames	Available external hosts.