EDIT Internet Platform for Cybertaxonomy Draft Design

Last Update: Tue Aug 21 13:18:08 2007

EDIT Internet Platform for Cybertaxonomy - Draft Design

This document serves as a draft for the design of the EDIT Internet Platform for Cybertaxonomy. EDIT, the European Distributed Institute of Taxonomy, is a Network of Excellence project within the 6th EU Framework Programme (Sub-Prioroty 1.1.6.3 "Global Change and Ecosystems"), Contract no. 018340

Version: 0.6

Date: 21 August 2007

Author: Markus Döring

Status of this document: Working draft under revision. This document may be updated, replaced or obsoleted by other documents at any time.

Comments about this document should be sent to Markus Döring <m.doering@bgbm.org> or the EDIT developer list: dev-edit@mnhn.fr. Subscription and archives are open to the public.

1. Introduction

EDIT's Internet Platform for Cybertaxonomy is a distributed computing platform that assists taxonomists to do revisionary taxonomy and taxonomic field work efficiently, expediently and via the web. Although the approach in the short and medium term will be strictly pragmatic and product oriented, room will be left to think about long term integration of taxonomy into a broader eScience environment. The character of EDIT as a joint effort of large taxonomic institutions with high level support in the institutional hierarchy provides a unique opportunity to build sustainable structures, if acceptance of the tools by the taxonomic researcher can be assured.

This document serves as a draft for the design of the EDIT Internet Platform for Cybertaxonomy. It is mainly targeted at the general overall architecture of the platform with recommendations for component developers to guarantee interoperability and to reduce development costs.

1.1. Requirements

The design of the platform is based on workflow modelling and basic use cases which are available on the EDIT developer wiki. Throughout the design of the platform some guiding objectives were kept in mind. The most important non-functional requirements are:

- increase productivity of taxonomical research
- integrate existing resources and applications rather than starting new developments
- integrate with existing or upcoming similar efforts
- compatibility with developing standards in the wider arena of Internet-based data exchange
- work with taxonomic concepts not names only
- potentially allow offline work
- restrict used ports to common web ports like http and https

2. Architecture

2.1. Networked Components

The Internet Platform for Cybertaxonomy consists of interoperable but independent platform components. Platform components can take the form of software applications (desktop or web-based) for human users or (web) services. The platform as envisioned will not have a single user interface or website; rather, it will be a collection of interacting components which may be combined and assembled according to the task in hand. A major aim of EDIT is the integration of existing software into the platform by means of establishing a common data model. To guide the user through common workflows, a high level abstraction of taxonomic revision and field work will be

formulated. It will show typical use cases and guide the user (a taxonomist) to appropriate software tools and resources.

2.2. Data Model

2.2.1. EDIT Common Data Model (CDM)

At the core of the platform lies a common data model to enable interoperability between the different components. In order to interact with the platform, a component should know how to interact with at least a subset of the CDM. The model describes all the commonly used data that is dealt with in the platform, and therefore covers at least taxonomic names and concepts; literature references; authors; (type) specimen; structured descriptive data; and species related content of any kind like economic use or conservation status. Nearly all this data has already been described by existing or upcoming TDWG standards. Unfortunately, there are still major gaps in compatibility, so a new integrated data model has to be developed in order to quickly yield results.

The model is being developed using the UML. Java classes are derived that have XML bindings and contain persistency annotations (JPA). An XML schema incarnation of the CDM will be used to validate data exchange and thus is the normative format that needs to be understood by the different components. As mentioned before, there is no such integrated TDWG schema yet, so the schema will try to incorporate as much of the existing standards as possible; especially TCS looks promising.

The latest TDWG approach using RDF and OWL ontologies for LSIDs (termed "biodiversity bus" by GBIF and EoL) has been considered but rejected as the foundation of the platform, because it is impossible to set cardinality constraints that are very important for applications. There will be services translating the EDIT CDM into TDWG RDF and vice versa, but TDWG RDF violating CDM cardinalities will be lost in this process, so no full round-tripping is possible. The envisioned CDM Sync Store (see below) will also expose CDM data as RDF.

A similar modelling effort has been undertaken by the CATE project. The platform development aims at creating a shared domain model library in Java (see developer tools below) that can be used as a foundation for many other java based biodiversity projects including CATE. Apart from the pure data model it will contain an XML serialisation (marshalling) and persistency layer as well as some basic business logic.

2.3. Webservices

Webservices in the broader sense (not SOAP only) allowing components to interact should be build using the common data model. The creation of new data formats should be limited as much as possible. Simple REST services with URL parameters instead of XML messages are the preferred solution, but SOAP services should be used for more complex services requiring state and transactions. If SOAP/WSDL is used, the literal binding (document/literal wrapped WSDL style) to the CDM XML Schema is recommended instead of the RPC style.

Services using a different format than the CDM should also provide at least a transformation service from/to their custom format into the CDM to guarantee

platform interoperability.

2.4. Security

Not all EDIT services or applications will be public and open. For data manipulating services and also for services which require a lot of computing power a secure authentication needs to be provided. In a distributed environment with many participating components and institutions and users, it would be desirable to have a common infrastructure for this problem. The EDIT platform embraces existing modern technology in this field which simplifies a users life while at the same time reducing development costs for service providers.

2.4.1. Introduction to Single Sign On (SSO)

Often a coherent authentication strategy or a solid authentication framework is missing. Over time this leads to a proliferation of applications, each of which comes with their own authentication needs and user repositories. At one time or another, everyone needs to remember multiple usernames and passwords to access different applications on a network. This poses a huge cost for the administration and support departments -- accounts must be set up in each application for each employee, users forget their passwords, and so on.

Authentication is a horizontal requirement across multiple applications and infrastructures. In general, there's no reason why user Mary should need multiple usernames. Ideally she should only need to identify herself once and then be provided with access to all authorized network resources. The objective of SSO is to allow users access to all applications from one logon. It provides a unified mechanism to manage the authentication of users and implement business rules determining user access to applications and data.

Benefits of SSO:

- Improved user productivity. Users are no longer bogged down by multiple logins and they are not required to remember multiple IDs and passwords. Also, support personnel answer fewer requests to reset forgotten passwords.
- Improved developer productivity. SSO provides developers with a common authentication framework. In fact, if the SSO mechanism is independent, then developers don't have to worry about authentication at all. They can assume that once a request for an application is accompanied by a username, then authentication has already taken place.
- Simplified administration. When applications participate in a single sign-on protocol, the administration burden of managing user accounts is simplified. The degree of simplification depends on the applications since SSO only deals with authentication. So, applications may still require user-specific attributes (such as access privileges) to be set up.

Problems with SSO:

- Single point of attack. With single sign-on, a single, central authentication service is used by all applications. This is an attractive target for hackers who may decide to carry out a denial of service attack.

2.4.2. Shibboleth

EDIT plans to make use of the existing open-source http://dev.e-taxonomy.eu/trac/wiki/SecurityComponents Shibboleth for federated identity-based authentication and authorization, single-sign-on and secure exchange of common user attributes, i.e. metadata like a persons name, email, taxonomic interest, geographic focus, etc. Shibboleth Identity providers (IdPs) supply user information, while service providers (SPs) consume this information and gate access to secure content.

It is an Internet2 Middleware Initiative project and has been under development since 2001. As a stable tool build on accepted security standards such as SAML, it has been promoted by the National Science Foundation's Middleware Initiative (NMI), the Middleware Architecture Committee for Education (MACE) and the Higher Education Funding Council for England together with JISC. Shibboleth is particularly popular within the higher education community which should facilitate the acceptance of it with many academic partners in EDIT. The setup of a true Shibboleth federation should be discussed with the ISTC as it provides a great potential of integrating IT resources.

2.4.2.1. EDIT Federation

A Shibboleth federation provides part of the underlying trust required for function of the Shibboleth architecture. A federation is a group of organizations (universities, corporations, content providers, etc.) who agree to exchange user attributes using the Shibboleth/SAML protocols and abide by a common set of policies and practices defined by the federation governing the exchange, use, and population of attributes (see User Attributes below) before and after transit. A https://spaces.internet2.edu/display/SHIB/ShibbolethFederations list of large, existing can be found on the Shibboleth website.

2.4.2.2. EDIT Identity Providers

At the heart of Shibboleth lies the Identity Provider. It knows how to authenticate a user and supplies service providers with attributes about a user through its Attribute Authority component. Initially the EDIT platform will work with a single IdP only. But as the institutional integration proceeds, a true distributed institute can be realised by setting up further Identity Providers in different institutions. Those IdPs could authenticate users and provide attributes about them through their existing local user management, i.e. they can safely expose their local user registry based on LDAP, Active Directory, Kerberos for example through a Shibboleth Identity Provider. The initial EDIT provided IdP uses an Apache based authentication accessing the Drupal based Taxonomic Experts Database that manages credentials and user attributes.

2.4.2.3. Shibbolethising EDIT Services

To control access to a web resource or service a shibboleth service provider needs to be installed, which currently runs best under the Apache web server, but also works with Microsoft Internet Information Server. As most services can at least run behind Apache somehow (e.g. Tomcat, Axis, CherryPy, PHP),

most services can easily be "shibbolethized" without writing Shibboleth or SAML specific software.

2.4.2.4. Integration with non EDIT services

Shibboleth is supported by a range of other mostly academic networks, especially in the library community. JSTOR, ArtSTOR and OCLC have support for Shibboleth already. Many applications https://wiki.internet2.edu/confluence/display/seas/Home] like https://mams.melcoe.mq.edu.au/zope/mams/pubs/Installation/dspace14 and Fedora support Shibboleth out of the box. A simple benefit for all EDIT users would be access to JSTOR, no matter where they are.

2.4.2.5. User Attributes

2.4.2.5.1. EduPerson

For resources that are being shared to a large community, it is also to the benefit of the resource provider to have a set of common attributes that can be easily categorized and distinguished. Among the Shibboleth participants, the most popular attribute scheme is EduPerson, which is the default scheme in Shibboleth. It defines a series of fields that are most relevant to the academic environment. Some of the fields that are relevant for authorization purpose in EDIT are:

eduPersonPrimaryAffiliation
: This field provides the name of the identity provider that the user is associated with.

eduPersonScopedAffiliation
: This field identifies the role of the user within the identity provider. Such role can be staff, student, or administrators, etc.

eduPersonEntitlement
: This field contains the access-control attributes. As described in EduPerson specification, this field accepts attributes with multiple values. Consequently, attributes to describe different levels of access control can be applied.

eduPersonTargetedId
: This field contains a unique ID that represents the user, instead of the normal login name. This is to satisfy the requirement of protecting the identity of the user, yet provide means for the service provider to backtrack and report to the identity provider in the case of malicious usage. Usually, this ID can be an encrypted combination of several attributes of the user.

2.4.2.5.2. EDIT Specific Extension

For EDIT it might be useful to consider the following additional attributes to describe people in the EDIT federation. These attributes could be used by applications and services to implement different default behaviour:

editNomenclaturalCode
: Primary nomenclatural code used: ICBN (botany), ICZN (zoology), ICNCP (cultivated plants), ICNB (bacteria), ICTV (viruses)

editTaxonomicFocus
: A higher taxonomic group the user's work is focussed on.

editGeographicFocus
: A geographic region the user's work is focussed on.

2.5. Annotations

It is envisioned that EDIT uses the W3C Annotea framework to annotate web resources including taxonomic data. Annotea is build around RDF and some OpenSource implementations like the Annozilla server which stores annotations exist already. This recommendation should be reviewed once such a component is being designed.

3. Platform Components

A list of components being developed, adapted or hosted to establish the initial EDIT platform. There maybe more components in the future, but most of the listed components here are already in development and required to establish a useful platform for the entire workflow.

3.1. Developer Tools

For all developers of the EDIT project some central development tools are provided. These tools should be used for all developments to increase communication and allow for a transparent development process keeping the virtual development team synchronised. All tools integrate with the Shibboleth SSO component too.

3.1.1. TRAC

TRAC is a lightweight project management tool targeted at software development that integrates with subversion (see below). It provides a "wiki" that is used in WP5 for development documentation, a flexible but simple "ticket" system that is used for general task management, as a bug tracker and for feature requests. Tickets itself are not linked to a specific date, but are instead linked to a "milestone" and an owner, i.e. the person responsible for this ticket. A milestone on the other hand has a planned date, so that a timeline can be generated with open/closed tickets or tasks related to a specific milestone. Official EDIT WP5 deliverables are entered as Trac milestones together with a task ticket linked to the person in EDIT responsible for the completion of the deliverable. The ticket and milestone subsystem of EDIT's Trac instance is not public and is only available to registered EDIT developers at http://dev.e-taxonomy.eu/trac.

Integration into the Eclipse IDE is provided through the Mylyn plug-in which is part of the standard distribution of Eclipse since version 3.3 Europa.

3.1.2. Subversion

The central subversion versioning system for source code and text based documents in EDIT is available at http://dev.e-taxonomy.eu/svn

All software source code, but also technical reports and documentation is expected to be kept in this versioning system which is open for read access to the general public, but requires developer authentication to modify or upload documents. The trunk of the single repository is split into subfolders corresponding to different EDIT components or tasks. Fixed releases should be copied to the tags section of this repository while keeping the first component/task related subfolder name. For example the cdm (library) component uses: http://dev.e-taxonomy.eu/svn/trunk/cdm. Once the first release version 0.1 is done, it will be copied to the tags section and named http://dev.e-taxonomy.eu/svn/tags/cdm/cdmlib_rel0.1. The structure below the first folder is entirely up to the responsible developers, but "pollution" of the first folder level shared between all developers should be avoided.

3.1.3. CDM Library

The CDM Library is meant to be a flexible shared java library to be used by different applications, i.e. J2EE having an application container, simple command line tools or swing or SWT based desktop applications. The library defines a persistent domain model, EDITs Common Data Model, that can be serialsed and read in XML. Business logic shared between applications are part of the library as much as possible, while application specific logic has to stay out. We hope to develop this library in collaboration with the CATE project initially.

The library will allow developers to quickly develop tools as it provides databasing, XML import/export, basic taxonomic logic, validation, data versioning and a synchronisation interface to talk to a CDM Sync Store in case the library is being used in an offline environment with a local database. We will use the Spring 2.0 framework to develop the library and keep the coupling of components low and use Hibernate to persist objects into a database. Access to the domain objects is through an exposed service layer API.

3.1.3.1. Synchronisation

The existing SyncML standard used mainly for mobile devices has some java libraries and an open source implementation of a bidirectional synchronisation server called Funambol. EDIT plans to make use of that within the CMD library and with the existing XML binding of CDM objects the main additional requirements are:

- track modified/deleted objects
- provide resolution mechanism for conflicts

3.1.3.2. Service API

The service API is a layer on top of the domain model that holds methods to interact with the persistent objects. It will provide search and other retrieval methods but also updating and deleting methods. Its business logic will prevent corrupted data being entered.

3.1.3.3. XML Binding

All CDM objects know how to serialise into XML. This marshalling and unmarshalling of objects will not only provide import and export mechanisms but also allow objects to be serialised into the SyncML protocol for example.

Initially there will be a native CDM XML format only as there is no integrated TDWG data standard yet. Transformation services will translate this CDM XML into other used TDWG standards such as TCS, ABCD or SDD.

3.1.3.4. Object Relational Mapping

The library will use JPA w/ Hibernate or alternatively TopLink for persistency of data. This allows the use of basically any relational database system with components using the library, e.g. the central CDM store or the taxonomic editor.

3.2. Central Facilities

Central components hosted by partners for the entire platform

3.2.1. Biodiversity Software Tracker

An application and service tracker has been established at http://www.bdtracker.net to document applications, services, and resources (e.g. authority files or maps that can be used as GIS layers). It will be used to review the most important and promising software tools for taxonomists.

The Biodiversity Service & Application Tracker is a collection of links to software, tools and resources useful to taxonomists. All applications and services are categorised into an expandable and potentially hierarchical system covering major biodiversity topics. The site is initially populated with the help of EDIT WP5 but in the long term after the project has finished (~2010) the site will have to be maintained by the user community.

Anonymous users can search for tools and suggest new software to be reviewed. They also have the option to file bug tickets and suggest new features for the system. If they register and create an account, they can write comments to any review or even write reviews themselves that then have to be released for publication by the BDTracker site administrator (currently Malte Ebach). Each review page is dedicated to a single version of an application or service. It provides details such as system requirements, interfaces and standards, information on licensing and cost and most important a review from editors as well as a comments section at the bottom that is open for everyone.

3.2.2. Taxonomic Experts Database

The objective of this information service is to provide an efficient online information service on European Taxonomic experts, their expertise and ongoing and planned taxonomic research projects, building on existing services and content from previous efforts. The information service is for internal as well as external use, but it will be designed (and content collected) with high priority on meeting the needs for information by EDIT partners regarding:

- Provision of a better knowledge base for an integrated recruitment strategy in EDIT
- Taxonomic information backbone and Pan-European checklist activities of WP3
- Coordination of research organised by WP4 (especially regarding methodology)
- Formation of networks, committees and task-forces for WP6 and WP7 activities
- WP8s assessment of training resources for taxonomy in Europe

Priorities for content: 1. EDIT partners expertise and projects 2. Exemplar data to test structural, sociological and legal constraints 3. Expertise and projects in demonstrator groups identified by WP6 and in ATBI-+M areas identified by WP7. Coverage of all major taxa (inside and beyond EDIT)

Taxonomists will be able to update their profile themselves if they like. The user information in this database could be used for authentication and user attributes of the Shibboleth Identity Provider (see below). The website will be Drupal CMS based and contain for every taxonomist metadata consisting of:

- Their taxonomic expertise in 7 kinds of taxonomic activities (Enghoff &Seberg, 2006 and see below), within taxa - searchable by all levels down to family (standard taxonomy needs to be decided)
- Geographic expertise (standard geography needs to be decided; ISO countries or TDWG regions?)
- Environment of their taxonomic group (see elaboration below)
- Their ongoing taxonomic research
- National / International taxonomic research projects they are involved in
- Country they reside in
- Institution they work in
- Methods used

3.2.3. Shibboleth Identity Provider for EDIT

For all EDIT users regardless of their institution a default Identity Provider will be installed that allows users to register and manage their personal data. Users of institutions with their own IdP do not need to register at the default EDIT IdP, but are part of the EDID federation through their own institution.

The authentication and single-sign-on session management will be provided by the mod_auth_mysql apache module. User credentials and other attributes come from a Drupal installation and which is aimed to be merged with the taxonomic experts database component of WP2.

3.2.4. ViTaL

Compiling a bibliography is a core component of all taxonomic work, whether published or online. Active research requires the tracing of pertinent references, actual sight of the content, retention of copies for further study (or retention of links to the source, either online or in an institutional library) and management of lists of references. When a researcher produces a taxonomic publication, a customised reference list has to be built, and formatted to fit the publisher's specifications. The task of gathering literature has been identified as one of the bottlenecks impeding taxonomic work in the Work Package 5.2 Draft functional model and bottleneck report for revisionary taxonomy.

EDIT will provide bibliographic and literature discovery tools which will help reduce literature related bottlenecks which can hinder the progress of day-to-day taxonomic research. In response to discussions and a broader requirements gathering exercise, we are planning a website supporting federated searching of taxonomically relevant data sources accessible via standard protocols (e.g. Z39.50). Data sources include EDIT partner and other library catalogues, the Biodiversity Heritage Library (BHL), and electronic journals. Users will be able to click through the results of their searches to see useful resources and metadata, and there will be a link to the original content where this is available, via an OpenURL service.

The virtual taxonomic library (ViTaL) will also provide a place for taxonomists to view and search aggregated bibliographic references harvested from a number of reference management services and web sites such as the ones created with EDIT's community web tools. References will benefit from the same linking technology used for the search results.

Both for specimen (GBIF) and literature (ViTaL), an OpenSearch RSS feed is envisaged to enable simple searches for specimen and literature that can be incorporated into personal mashup sites (for example, see www.netvibes.com), or at a later stage into the personal search and monitor page of the planned experts sites. It will allow users to keep track of new items for frequent searches, e.g. a listing of new specimen entries for a certain region or new species page annotations for a given family.

Currently metalib is being considered and tested for managing a virtual library catalogue and the SFX link server for storing and resolving direct links to digital documents.

3.2.5. Geo-Services

The general aim of this activity is to provide the resources and applications able to publish, visualise, and analyze the distributional information associated with taxonomic information. Geographical components will implement services and applications based on, but concealing, the complexity of the underlying OGC compliant services, e.g. visualisation of distribution maps, itineraries, biogeographical modelling. As with all EDIT services the data to be displayed should be submitted as CDM files to the service unless a special transformation service to a custom but simpler format is provided.

3.2.5.1. Visualize Distributional Information

Initially the following set of services will be created for visualisation of taxonomic data:

- Visualise (many) points (specimen & observation coordinates) as simple points.
- Visualise occurrence data a la native/extinct/invasive per region. use colour and symbols for different statuses. The input format used here could be TDWG's SpeciesProfileModel (SPM) or some custom new one.
- A simple calculation service that sums up single occurrences per region. A visualisation service for regions could then be used to display coloured regions instead of simple points.

3.2.5.2. Spatial Completeness Analysis

Statistically analyse distributional information with regard to completeness of surveys

- examine the degree of completeness of the taxonomic distribution information
- discriminate well surveyed localities from those do not have reliable inventories
- locate the localities in which is necessary to carry on additional surveys in order to recover the environmental and spatial variation of the area. The activity is collaboratively carried out by all the partners.

3.2.5.3. Maps / Layers

EDIT will provide maps and other environmental GIS layers covering the world and Europe in detail. In particular all 4 levels of the TDWG regions are provided in addition to current ISO countries layers.

All maps are provided as vector graphics or in different qualities that also support printing, i.e. at least 300 dpi. GIS layers in the EDIT geoplatform are in geodetic coordinates (longitude, latitude), datum WGS84.

3.2.6. GBIF OpenSearch

GBIF Specimen search proxy that translates specimen or general occurrence searches to GBIF and returns results as RSS feeds. This allows the subscription of often searched queries into RSS readers and tools like the taxonomic editor.

3.2.7. Transformation Services

Transformation services exist to convert data into different formats, do sorting or some other transformation. Initial services consist of:

- converter to and from the Common Data Model, e.g. into Word Documents, TaxonX, Endnote, NEXUS or Excel spreadsheets serving as taxonomic checklists.
- Sorting service for list of taxa. The sorting can be done alphabetically but also, most importantly, according to some taxonomic system. This was requested by taxonomists that want to sort their lists according to some herbarium classification for example.

3.3. Community Components

A community is used here to describe a team of taxonomists working together on a dataset. A community of taxonomists is most often identified through a set of taxa, often a single higher taxonomic group, but it can also be a regional interest group or any other commonality that brings together a team of people.

Community components should be installed once per community and might therefore exist many times in total across all communities.

For community based manipulation of data, a workflow that supports proposals and review needs to be implemented on a central level, probably already in the CDM Sync Store (see below).

3.3.1. Community Web Tools

The community web tools, formerly termed communication tools, are basically a Drupal site that provides a content management system for webpages, blogs, forums, mailing lists and other collaborative tools. Every community will have its own site, with or without a separate domain, where the basic tools are installed. The community website will allow users to host images and other documents like pdfs. A file sharing service for files that are too big to be send via email should be another basic community service for registered users. Especially in the case of images and videos this can add up to a substantial amount of storage space.

A hosting service is not yet planned, but should be brought to attention in the ISTC. Many taxonomic communities do not have their own server or a budget to rent a dedicated server.

3.3.2. CDM Sync Store

The CDM Store is a central database for a community. It can handle the entire range of CDM data and is expected to be the main component other components will talk to. The CDM store will be build on top of the shared CDM library explained in more detail below. It is planned to offer two main interfaces to the CDM Store, a bidirectional Sync API based on SyncML and a direct access of the service API of the CDM library exposed as webservices.

3.3.2.1. Synchronisation

The synchronisation API will work on single objects and thus bypass any logic such as validation. Clients using the sync API therefore are initially required to use the CDM library which guarantees integrity of the data. Clients not using this java library are initially required to work on the webservices directly and thus cannot implement an offline mode.

Automatic merging of objects modified by different agents is not a priority. In case the same object was modified multiple times, a conflict could be raised that needs to be resolved manually. A simple graphical user interface for conflict resolution is needed, but it seems that at least the Eclipse Rich Client Platform has some prebuild solutions for interfaces of that kind. It would be good to have a more or less standalone conflict resolver application that could be used in several places.

3.3.2.2. Webservices

The entire service API of the CDM library is expected to be exposed as webservices in a later stage, probably through SOAP. As the CDM lib API is rather fine grained based on many small but complex objects, webservices based directly on this API tend to be slow because many remote calls need to be made to accomplish a single overall task. After the fine grained CDM API has stabilised and if there still is the demand for a webservice based API, a much more coarse grained API based on broader transfer object classes should be developed that reduce the need of multiple remote calls to render for example a single webpage or GUI layout.

LSID resolution is part of the webservices that should be implemented in a pretty early stage of the development, allowing resolution of first class objects

such as names, taxa and publications into RDF documents based on the TDWG LSID vocabularies.

3.3.2.3. Archive

The CDM Store should provide archival functions, i.e. store CDM objects persistently and immutably. An LSID resolution will be provided to serve those archived objects as CDM objects as well as TDWG RDF.

3.3.3. CDM Data Portal

The public data portal allows browsing and searching of CDM data, focussed primarily around taxa. Because a portal is subject to many specific community needs, it should to be flexible and easy for communities to adapt. This is why we base it on PHP which is popular also among many non computer scientists. EDIT's preferred content management system Drupal is also written in PHP, so it might be an option to develop a Drupal portal module out of it too. The data to be published has to be a local CDM datastore which can be the primary CDM Store or alternatively a mirrored or synched read-only cdm store. Data displayed in the portal cannot be modified directly, but annotations to any taxon or other objects should be possible in a second stage. Requirements for the data portal are currently being gathered by the WP6 exemplar groups.

3.3.4. ATBI Site Database

A central ATBI site database the aggregates all data from the individual taxonomists that do field recordings. This Site Database is linked to GBIF to supply all it's occurrence records to the GBIF indexing system. It is the core of the ATBI platform similar to the CDM Sync Store for taxonomical data.

3.4. Individual Components

Software that is installed and used by individuals. This is usually desktop software or handheld devices as opposed to web based tools which are usable by many users per installation.

3.4.1. Taxonomic Editor

A local editor to manage taxonomic data. It will be a desktop application build on the Eclipse Rich Client Platform and the CDM library running on an embedded or shared network database. The application will be able to two-way synchronize its data with a CDM Sync Server to cater for teams working on the same dataset. Eclipse RCP seems to be very suitable and provides many needed but rich features like synchronisation and conflict resolving views, RSS readers, tree browser and much more.

The development starts with a technical prototype to proof the suitability of the chosen technology. The editor development will then start with botanical Taxon Names and grow from there into the breadth of the entire Common Data Model covering zoological names, taxonomic concepts, publications, type specimen and other related information with frequent releases in between.

3.4.2. Structured Description Editor

An application for managing structured descriptive data will be part of the initial platform as well. The ongoing reviews of existing descriptive tools in bdtracker suggest to extend the Xper2 application with CDM capabilities. Initially this will probably only be import/export, but we hope to integrate the CDM library with its synchronisation capabilities too. This way a single CDM store could be accessed with a specialised application for nomenclature and taxonomy, but also with a specialised descriptive tool. It is a Java based software, so the usage of the CDM library seems to be possible. We are currently seeking collaboration with Gregor Hagedorn, one of the developers of TDWG's SDD standard, in order to integrate a data model similar to SDD into the CDM.

3.4.3. ATBI Field Recorder

A simple MS Access database that stores observation records in the field. It can load a predefined set of taxonomic concepts to work with and can dump its data into the central ATBI Site Database.