Integrated Publishing Toolkit (IPT)

Architectual view

Primary Authors: Tim Robertson

Version history

Date	*Comment*	*Author*	*Document Version*	*IPT Version*
10/01/2011	Initial draft, detailing the functionality offered by 2.0GA and providing the document structure for future developments	Tim Robertson	0.9	2.0GA
18/04/2011	Enhancing the Registry Communication section	José Cuadra	1.0	2.0GA
08/07/2011	Modifying section 7.1.1.1.6 - RegistryAPI: updateIPT()	José Cuadra	1.1	2.0.2GA

Table of contents

1 Introduction 1

2 About this document 3

2.1 Intended audience 3

2.2 Document structure 3

2.3 Conventions 4

2.3.1 Significant actors 4

2.3.2 Reasoning 5

3 Use cases 6

3.1 Administrative use cases 6

3.1.1 Use case: Configure installation 6

3.1.2 Use case: REGISTER IPT 7

3.1.3 Use case: Associate organisation 8

3.1.4 Use case: Install extensions 9

3.1.5 Use case: Create user account 10

3.2 Managerial use cases 11

3.2.1 Use case: Create metadata RESOURCE 11

3.2.2 Use case: Create occurrence RESOURCE 12

3.2.3 Use case: Create checklist RESOURCE 13

3.2.4 Use case: Create RESOURCE from DwC-A 14

3.2.5 Use case: Make RESOURCE public 15

3.2.6 Use case: Publish RESOURCE 15

3.2.7 Use case: Add manager to RESOURCE 16

3.2.8 Use case: REGISTER RESOURCE 16

3.3 User use cases 17

3.3.1 Use case: Download DwC-A 17

3.3.2 Use case: Download Metadata 17

3.4 Scenario depicting a *typical* use of the IPT 18

4 Logical architecture 20

4.1 IPT Administration (Administrative user functionality) 21

4.1.1 Initial settings 21

4.1.1.1 Data directory 21

4.1.1.2 ADMIN account 21
4.1.1.3 Installation type 21
4.1.1.4 Base URL 22
4.1.2 REGISTRY Configuration 22
4.1.3 ORGANISATION association 24
4.1.4 Extension management 25
4.1.4.1 Vocabulary management 25
4.1.5 User management 25
4.2 RESOURCE management (Managerial user functionality) 26
4.2.1 RESOURCE creation 26
4.2.2 Metadata authoring 26
4.2.3 Source data configuration 30
4.2.3.1 File based sources 30
4.2.3.2 SQL Database sources 31
4.2.4 Mapping of source data to Darwin Core terms 32
4.2.4.1 Core source file mapping 32
4.2.4.1.1 Assigning identifiers 32
4.2.4.1.2 Filtering input data 33
4.2.4.2 Extension source files 33
4.2.5 RESOURCE visibility state (PUBLIC, PRIVATE, REGISTERED) 33
4.2.6 Publishing a RESOURCE 34
4.2.7 Deleting a RESOURCE 35
5 Discovery and access services (User level permissions) 35
5.1 EML Metadata 35
5.2 Darwin Core Archive 35
5.3 RSS Feed 35
6 Development architecture 36
6.1 Data directory 37
7 Process architecture 39
7.1 REGISTRY communication 39
7.1.1 Authentication 39
7.1.1.1.1 RegistryAPI: GetExtensions() 39
7.1.1.1.2 RegistryAPI: getOrganisations() 40
7.1.1.1.3 RegistryAPI: getVocabularies() 40

7.1.1.1.4 RegistryAPI: validateOrganisation()....41
7.1.1.1.5 RegistryAPI: registerIPT()....41
7.1.1.1.6 RegistryAPI: updateIPT() 43
7.1.1.1.7 RegistryAPI: registerResource()....44
7.1.1.1.8 RegistryAPI: updateResource() 46
7.1.1.1.9 RegistryAPI: deregister(Resource)....48
7.1.1.2 REGISTRY view of IPT....48
8 Physical architecture....50
8.1 Deployment scenarios....50

Index of figures

Figure 1: 4+1 Architectural view model 3

Figure 2: Administrative use cases 6

Figure 3: Managerial use cases 11

Figure 4: Managerial registration use case 16

Figure 5: User use cases 17

Figure 6: IPT software architecture 36

Figure 7: REGISTRY view of IPT with 2 resources 49

Figure 8: REGISTRY view of IPT in shared data hosting scenario 49

Figure 9: Deployment view of the IPT Application 50

Index of tables

Table 1: 4+1 View model description 4

Table 2: User roles and permissions 20

Table 3: Summary of metadata fields supported in the IPT 30

Table 4: Delimited text configuration options 31

Table 5: Relational database configuration options 32

Table 6: Core record definition locations 32

Table 7: Identifier assignment during mapping 33

Table 8: Supported filter predicate types 33

Table 9: RESOURCE state transitions 34

Glossary of terms

Term	*Definition*
Administrator (Admin)	A user of the IPT who is considered to have administrative permissions. An IPT will always have at least one user with this level of authority.
Checklist resource	A resource having information about one of many types of taxon-related lists.
Core extension	One of two types of Darwin Core extensions (Taxon and Occurrence) used as the basis of a resource. Additional extensions might be linked to these extensions when mapping data in the IPT.
CSV File	A file that contains data in the Comma-Separated Value format.
Data directory	The directory used by the IPT to keep all data and configuration.
Dublin Core	A standard consisting of generic metadata terms.
Darwin Core	A standard consisting of terms and classes of terms used to share biodiversity data.
Darwin Core Archive	A single zipped archive for a data set consisting of one or more text files of data, an XML file (meta.xml) describing the contents of the text files and how they relate to each other, and an XML file (eml.xml) containing the dataset metadata in EML format.
DOI	A Digital Object Identifier is a persistent, unique and actionable identifier that may be purchased and assigned for any digital resource that is available on the web.
EML	The Ecological Metadata Language is an XML-based profile used to encode metadata about a data set.
Extension	In this User Manual, an extension is a set of terms corresponding to a specific class of data. An extension should be thought of as an extension of the capabilities of the IPT rather than as an extension of any particular standard. For example, the Darwin Core Occurrence extension is set of terms from the Darwin Core describing Occurrences. It is not an extension to the Darwin Core.
GBIF Registry	The GBIF Registry is an application that manages the nodes, organisations, resources, and IPT installations registered with GBIF, making them discoverable and interoperable.
Manager	A role assigned to an IPT user account allowing permissions to create edit and delete a resource
Metadata	Metadata refers to the higher-level information about a dataset as opposed the primary data in the data set.
Metadata resource	A resource having information about a dataset, but without having the actual primary data. A metadata resource might give information about a collection that has not yet been

	digitized, for example. Over time a metadata resource might become an occurrence resource or checklist resource with the addition of data.
MVC	The Model View Controller is a software architecture pattern used in web applications to separate the concerns of data modelling, the rendering of the model for client consumption and the logic for accessing the content.
Occurrence resource	A resource having information about Occurrences as defined in the Darwin Core.
Private	A state of a resource in which only the creator, invited managers, and IPT administrators can view it.
Public	A state of a resource in which anyone can view it.
Published Resource	The latest version of the Darwin Core Archive produced for a resource in the IPT and registered in the GBIF Registry.
Registered	A state of a public resource or of an IPT instance in which anyone can discover it through the GBIF Registry.
Resource	In this document, resource refers to a dataset and the metadata about it.
RSS	Really Simple Syndication, a type of subscription format for tracking changes to a web site.
Manager	A role assigned to user accounts that allows permissions to create, change, and remove resources.
Shortname	A short unique name used for resource identification within the IPT and services that access the IPT.
Source data	In this document the source data are the data that are mapped to extensions within a resource and may consist of text files or a database.
UML	The Unified Modeling Language is a standardised general-purpose modelling language in the field of software engineering
Visibility	A term describing how a resource may be viewed (private, public, or registered).

1 Introduction

The Integrated Publishing Toolkit (IPT) is a deployable web-based tool that allows users to serve onto the Internet (or Intranet):

- Primary biodiversity occurrence data residing in databases or in text files, such as comma separated value files. Such data are typically individual specimen records, or observations of individuals of a species. This content type can be used where the occurrence of an individual of a species is considered the core conceptual unit of a dataset.
- Taxonomic checklist data residing in databases or in text files, and any associated information such as geographic coverage or vernacular names for example. This content type can be used to expose a dataset where the core conceptual object is considered a species.
- Higher-level dataset descriptive data (metadata), which is authored through the IPT web interface, to describe the dataset as a whole, including taxonomic, geographic, temporal scopes, how it was assembled and the citation requirements among others.

Occurrence and taxonomic data are published to the Darwin Core standard[1], which provides a glossary of well-defined terms to describe taxa, their occurrence in nature as documented by observations, specimens and samples and related information. The IPT uses the Darwin Core Archive format as the output format, which is a compressed text-file based data format.

The metadata format used by the IPT is in accord with the GBIF Metadata Profile, a profile built upon the Ecological Metadata Language[2] version 2.1.0.

To PUBLISH data, a RESOURCE is created, the source data are identified either through the uploading of files, or the configuration of a database connection, and a user defined *data mapping* is created to transform the input data into the terms found in the Darwin Core and the IPT extension definitions. This holds many similarities to *Extract Transform Load* tools[3].

The IPT communicates with the GBIF REGISTRY, to allow the easy publishing and sharing of data through the GBIF network, and the relating of datasets to associated Institutions. The IPT supports co-hosting scenarios, where one IPT installation may be used to register datasets on behalf of external institutions while still preserving the relationship of the dataset to the external institution.

The IPT communicates with the GBIF REGISTRY to discover IPT extensions, and any referenced controlled vocabularies, which may be used in the mapping of arbitrary data. The IPT extension mechanism provides the flexibility to define custom data formats that may be defined and registered

[1] http://rs.tdwg.org/dwc/

[2] http://knb.ecoinformatics.org/software/eml/

[3] http://en.wikipedia.org/wiki/Extract,_transform,_load

centrally, and then used to exchange information in a common format. An example could be the definition of a set of controlled terms to describe the invasive status of a species in a geographic area. Once created and registered, all IPT users would be able to map data to this well-defined extension, enabling communities to exchange data in common, yet flexible data formats.

The IPT supports multiple users, with a permission based authentication model. User management functionality is provided to create and edit users, and assign one of the following roles to the user:

- ADMINISTRATOR: Full permissions to configure the IPT settings, user accounts and all data RESOURCES configured in the IPT.
- MANAGERIAL: Permission to create, configure and delete data RESOURCES. A MANAGER may or may not be given permission to REGISTER the RESOURCE with the GBIF REGISTRY.
- USER: No permissions beyond an unauthenticated user. This role exists as a placeholder to allow a user account to remain in existence, but without any permission to configure content. This is particularly useful should a user no longer work at an INSTITUTION.

The tool is intended for Institutional deployment rather than deployment by an individual bench scientist, or for those with changing IP addresses such as a laptop user. However, it may be useful by those types of user for certain data transformation scenarios.

2 About this document

2.1 Intended audience

The intended audiences for this document are technical personal interested in understanding the IPT architecture and embedded business rules, and developers joining the project. In particular this document is intended for those who might consider building upon or extending the IPT platform. This document is expected to develop with the software, and provide the structure to which further IPT developments be documented.

The IPT is an open source project, and enhancements to the IPT codebase, documentation, functionality and project coordination are welcome.

2.2 Document structure

This document follows the 4+1 view model designed by Philippe Kruchten. The 4+1 is a view model for describing the architecture of software-intensive systems, based on the use of multiple, concurrent views. The views are used to describe the system from the viewpoint of different stakeholders, such as end-users, developers and project managers. The four views of the model are logical, development, process and physical view. In addition, selected use cases are utilised to illustrate the architecture serving as the 'plus one' view. Hence the model contains 4+1 views.

The document structure provides the framework by which engineering documentation for future IPT enhancements will be provided. Therefore this document will undergo distinct versioning, indicating the IPT version to which the document relates.

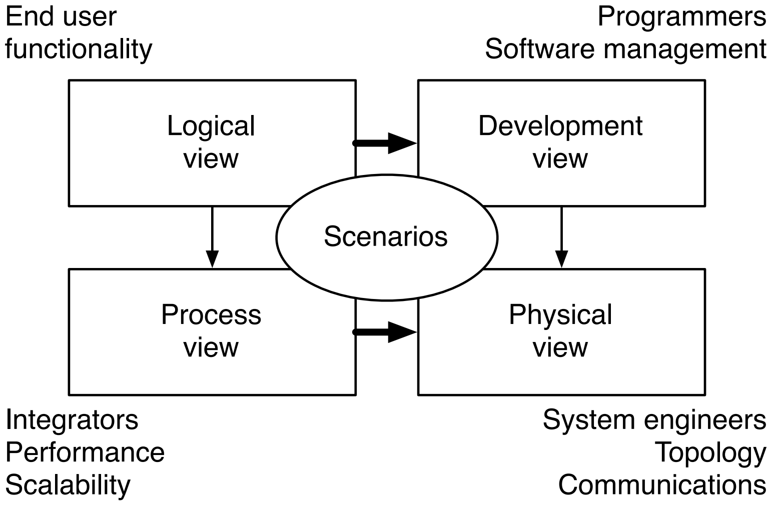

Figure 1: 4+1 Architectural view model

Logical view The logical view is concerned with the functionality that the system provides to end-users. UML Diagrams used to represent the logical view may include the class diagram,

	communication diagram and sequence diagram.
Development view	The development view illustrates a system from a programmer's perspective and is concerned with software management. This view is also known as the implementation view. It may use the UML component diagram to describe system components and may include a UML package diagram.
Process view	The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the runtime behaviour of the system. The process view addresses concurrency, distribution, integrators, performance, and scalability, etc.
Physical view	The physical view depicts the system from a system engineer's point-of-view. It is concerned with the topology of software components on the physical layer, as well as communication between these components. This view is also known as the deployment view. UML diagrams used to represent physical view may include the deployment diagram.
Use case view	The description of the architecture is illustrated using a small set of use cases, or scenarios, which become a fifth view. The scenarios describe sequences of interactions between objects, and between processes. They are used to identify architectural elements and to illustrate and validate the architecture design. They also serve as a starting point for tests of an architecture prototype. UML diagram(s) used to represent the scenario view will include the use case diagram.

Table 1: 4+1 View model description[4]

2.3 Conventions

2.3.1 Significant actors

Within this document, terms in CAPITAL CASE are considered significant actors, operations or states, and the vocabulary used is of importance. The terms ORGANISATION and INSTITUTION are used interchangeably and can be considered synonymous within the scope of this document; this means a GBIF PARTICIPANT NODE, a museum or a herbarium can be considered as an INSTITUTION or an ORGANISATION within the scope of this document.

[4] http://en.wikipedia.org/wiki/4%2B1_Architectural_View_Model

2.3.2 Reasoning

Reasoning behind certain functionality is captured so that others may benefit from previous discussion outcomes and better understand the rationale for the design. When developing this document, use of this is encouraged for areas where complex business rules apply.

The reasoning will be captured in boxes like this.

3 Use cases

[The architecture is illustrated using a small set of use cases, which describe sequences of interactions between objects, and between processes]

This section does not exhaustively document all known use cases for the IPT operation, but covers the primary scenarios necessary to understand the functionality and it's relationship with the external actors.

3.1 Administrative use cases

A user with ADMIN permissions is required to initiate the use cases in this section.

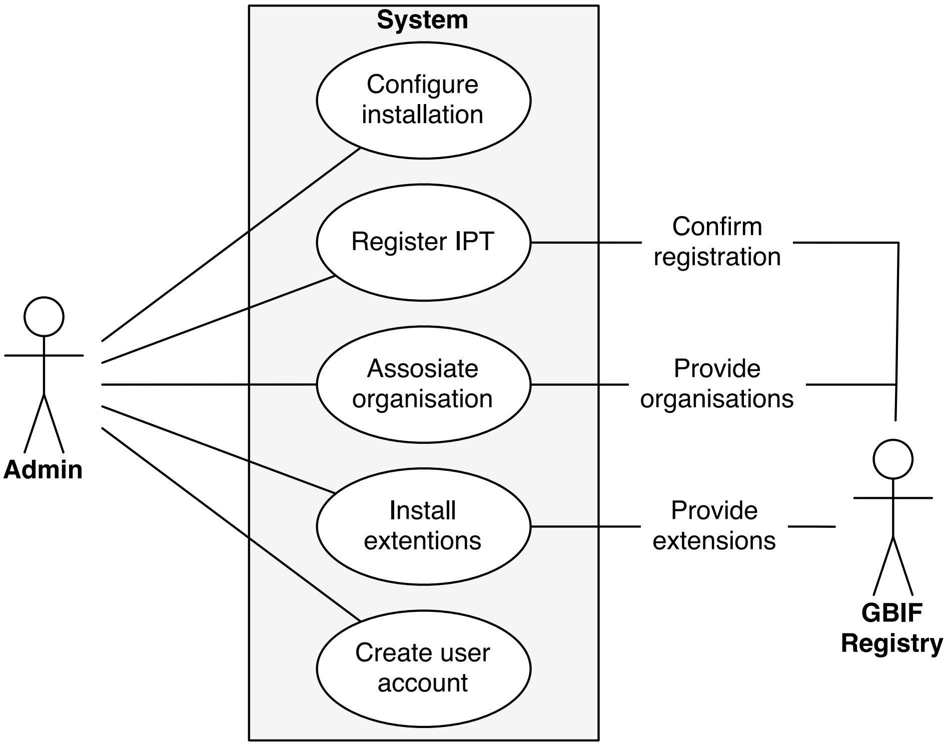

Figure 2: Administrative use cases

3.1.1 Use case: Configure installation

This use case captures the basic configuration of a newly installed IPT.

Prerequisites
The IPT has been successfully deployed in the application server (e.g. Tomcat or Jetty), with any necessary firewalls, ports etc configured along with DNS and any virtual host definitions, so that the IPT is addressable correctly on the desired URL. This use case covers 2 scenarios with the following prerequisites: - Basic flow: A directory has been created to hold the IPT content, and the

permissions on the directory allow read/write access to the IPT - Alternative flow 1: An existing IPT directory exists (e.g. from a previous installation)		
Basic flow		
Step	*Actor action*	*System action*
1	The user enters the server directory path where the IPT will store data	The IPT inspects the directory to determine if it holds existing configuration. If the directory does hold an IPT configuration, go to alternative flow 1, step 2. If the directory does not yet exist, create a new one. If it does exist, but is not empty, raise error and prompt for another path.
2	User enters the ADMIN account details, the URL that the IPT is addressable on, and any HTTP proxy information required for the IPT to open external connections to the Internet	The IPT validates the URL is accessible, and stores this configuration in the files and creates the ADMIN user account.
Alternative flow 1		
Step	*Actor action*	*System action*
2		The IPT reads and validates the configuration for the IPT setup (including users, extensions, installation type etc), and the RESOURCES (including source data, metadata, mapping configuration, user associations etc)

3.1.2 Use case: REGISTER IPT

This use case captures the registration of an IPT to the GBIF network.

Prerequisites
The IPT is installed and configured and the ADMIN user is logged in to the IPT. The URL of the IPT is addressable on the Public internet. This use case captures the following scenarios: - The INSTITUTION to which the IPT is associated is already known to GBIF and has been created in the GBIF REGISTRY

- The institution is not known to GBIF		
Basic flow		
Step	*Actor action*	*System action*
1	From the administration console, the user chooses to register the IPT.	The user is prompted to validate the URL at which the IPT is addressable on the Internet.
2	User initiates a validation of the URL.	The IPT instructs the GBIF REGISTRY to verify that the URL is addressable.
3	The GBIF REGISTRY performs a check that the URL is addressable, and reports to the IPT whether the test has succeeded or not.	If the test was unsuccessful, this use case terminates with a warning to the user. If the test was successful, the IPT requests the institution list from the GBIF REGISTRY
4	The GBIF REGISTRY provides the list of available institutions	The IPT provides this list to the user to choose from. If the desired institution is not available, the user is prompted to contact the GBIF Helpdesk. Otherwise the user is prompted to fill the necessary information (password, alias, description contact information etc) appropriate for the IPT
5	The user fills the form and submits	The IPT calls the GBIF REGISTRY with the new registration details
6	The GBIF REGISTRY validates the INSTITUTION password is correct and either stores the new information and returns successfully with the GBIF REGISTRY identifiers, or returns an error to indicate the password was not correct	The IPT stores the GBIF REGISTRY identifiers, or indicates to the user that the password was incorrect, whereby the user can attempt from step 5 again.

3.1.3 Use case: Associate organisation

This use case captures the association of an IPT to multiple institutions. This allows an IPT to be situated on a server belonging to one institution, but serving content on behalf of a different institution.

Prerequisites

The IPT is installed and configured and the ADMIN user is logged in to the IPT. The IPT has been registered to the GBIF network (this use case is only allowed following an IPT registration). This use case captures the following scenarios: - The INSTITUTION is already known to GBIF and has been created in the GBIF REGISTRY - The INSTITUTION is not known to GBIF		
Basic flow		
Step	*Actor action*	*System action*
1	From the administration console, the user chooses to configure the ORGANISATIONS to which the IPT is associated	The IPT presents the user with the current organisations
2	The user chooses to create a new association	The IPT requests the list of available organisations from the GBIF REGISTRY
3	The GBIF REGISTRY returns the list of ORGANISATIONS	The IPT displays the list to the user
4	The user can either - Select the organisation, and provide the password, alias and indicate if this organisation should be available for Managers to use when registering Resources - Contact GBIF helpdesk and request that the ORGANISATION be added, in which case the user must start this use case again	The IPT stores the chosen organisation, GBIF REGISTRY key, password, desired alias etc.

Note: A similar use case exists for the deletion of an organisation. Suffice to say, an ORGANISATION may only be deleted if there are no RESOURCES REGISTERED as associated with the ORGANISATION.

3.1.4 Use case: Install extensions

This use case captures the steps required to install extensions, and referenced vocabularies into the IPT, to enable them to be available to data MANAGERS during the RESOURCE creation stages.

Prerequisites		
The IPT is installed and configured and the ADMIN user is logged in to the IPT		
Basic flow		
Step	*Actor action*	*System action*
1	From the administration console, the user chooses to manage extensions	The IPT calls the GBIF REGISTRY for a list of all extensions
2	The GBIF REGISTRY returns the extensions available, and the URLs at which they are hosted	The IPT gets each extension, and any referenced vocabularies, and presents them to the user
3	The user chooses which extensions to install into the IPT	The IPT stores the installation status of each extension, to determine which are available to data managers during subsequent RESOURCE creation

Note: A similar use case exists for deletion of extensions. Suffice to say, an extension may only be deleted if it is not being used in a RESOURCE mapping.

3.1.5 Use case: Create user account

Prerequisites		
The IPT is installed and configured and the ADMIN user is logged in to the IPT		
Basic flow		
Step	*Actor action*	*System action*
1	From the administration console, the user chooses to manage users	The user is presented with a list of existing users
2	The user chooses to create a new account	The IPT displays the form necessary to create the user account
3	The user enters the account details, and chooses the ROLE to which the user is allowed (Admin, Managerial with/without registration permissions, User)	The IPT stores the information in the configuration.

Note: A similar use case exists for the deletion of a user. Suffice to say, a user may only be deleted if they are not connected to any data RESOURCE. A user may only be removed from a RESOURCE if they are not the creator of the RESOURCE.

3.2 Managerial use cases

A user may only initiate the use cases in this section if they have MANAGER or ADMIN level permissions. It should be noted that a MANAGER may or may not have permission to REGISTER a RESOURCE.

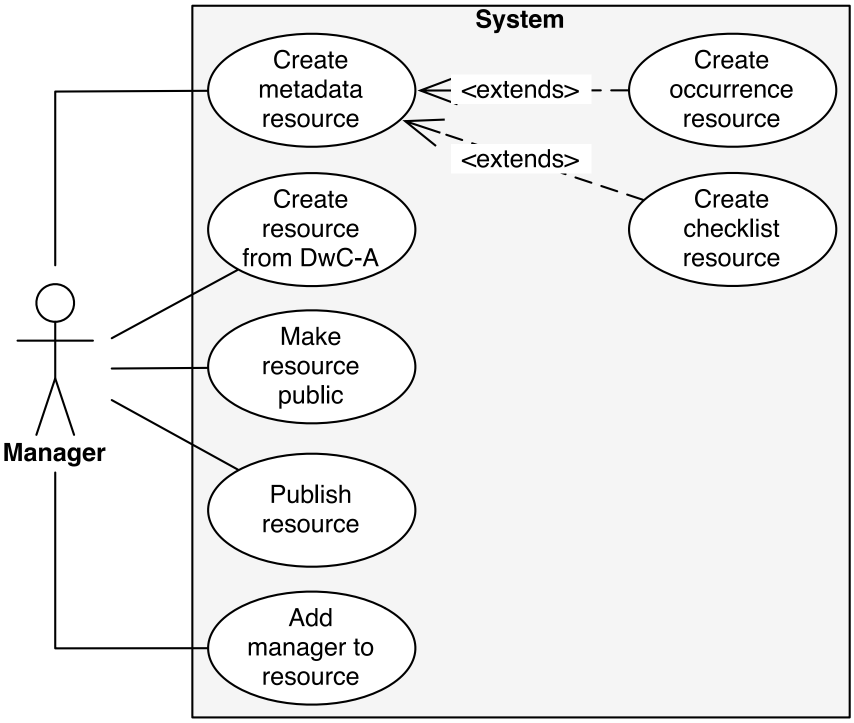

Figure 3: Managerial use cases

3.2.1 Use case: Create metadata RESOURCE

This use case captures a user wishing to create a RESOURCE in the IPT author dataset level metadata, but provide no source data. The metadata profile supported by the IPT conforms to the GBIF Metadata Profile[5], which is a profile based on the Ecological Metadata Language schema version 2.1.1[6][7]

Prerequisites
The IPT is installed, configured correctly and a user with MANAGER level permissions is logged in.

[5] http://rs.gbif.org/schema/eml-gbif-profile/

[6] http://knb.ecoinformatics.org/software/eml/

[7] http://rs.gbif.org/schema/eml-2.1.1/

Basic flow		
Step	*Actor action*	*System action*
1	From the manage resources console the user selects to create a new RESOURCE, providing a short unique name for the RESOURCE	The IPT checks the name is unique and configures the directory structure that will hold this RESOURCE content. The user is shown the RESOURCE overview console.
2	The user selects the metadata section from the RESOURCE overview console.	The user is presented the metadata forms to complete.
3	The user fills the forms of interest. On each form the user is required to save the progress	On saving progress, the IPT stores the saved information into the RESOURCE data directory, to ensure that no transient data is held.

3.2.2 Use case: Create occurrence RESOURCE

This use case captures a user wishing to create a RESOURCE in the IPT that represents an occurrence dataset, either by uploading a CSV file, or by connecting the IPT to a database. This use case extends the Use case: Create metadata RESOURCE.

Prerequisites		
The user has created a RESOURCE as per the Use case: Create metadata RESOURCE. The user has either a CSV file or a database holding the RESOURCE contents. The ADMIN has enabled the Darwin Core Occurrence extension.		
Basic flow		
Step	*Actor action*	*System action*
1	From the RESOURCE management console the user configures the source of the data by either: a) Uploading a CSV / Tab file b) Configuring a database connection, and SQL statement to the data	The IPT provides means to preview the data, and check the configuration is correct. These checks include the ability to verify that the line breaks, field delimiters etc are configured correctly and that the database resultset is readable.
2	From the RESOURCE	The IPT provides a mapping

	management console, the user configures a mapping for the source data configured in step 1 that will map to the Darwin Core Occurrence.	configuration view to the user. Features of the mapping view include: a) Fields in the source data that are named the same as DwC terms will be automatically recognised and mapped. b) The record ID (dwc:occurrenceID) may be mapped to a field in the source data, automatically incremented as a number, or a UUID[8] may be assigned. c) Terms may be mapped to a field in the source data d) Fixed values may be entered for terms that don't exist as fields in the source data e) Terms that the extension define as controlled by a vocabulary may be fixed to a vocabulary value, mapped to a field in the source data, or a translation may be provided to convert source values to the vocabulary preferred terms
3	The user completes the mapping and saves	The IPT stores the mapping configuration in the RESOURCE data directory.
4	From the RESOURCE management console, the user selects to PUBLISH the RESOURCE.	See Use case: Publish RESOURCE

3.2.3 Use case: Create checklist RESOURCE

This use case captures a user wishing to create a RESOURCE in the IPT that represents a checklist. This use case is identical to Use case: Create occurrence RESOURCE with the exception of the Actor action in Step 2.

Prerequisites
The user has created a RESOURCE as per the Use case: Create metadata

[8] http://en.wikipedia.org/wiki/Globally_unique_identifier

RESOURCE. The user has either a CSV file or a database holding the RESOURCE contents. The ADMIN has enabled the Darwin Core Taxon extension.		
Basic flow		
Step	*Actor action*	*System action*
2	From the RESOURCE management console, the user configures a mapping for the source data configured in step 1 that will map to the Darwin Core Taxon.	See Use case: Create occurrence RESOURCE

3.2.4 Use case: Create RESOURCE from DwC-A

This use case captures a user wishing to create a RESOURCE from an existing Darwin Core Archive.

Prerequisites		
The IPT is installed, configured correctly and a user with MANAGER level permissions is logged in.		
Basic flow		
Step	*Actor action*	*System action*
1	From the manage resources console the user selects to create a new RESOURCE, providing a short unique name for the RESOURCE, and selecting the DwC-A from their local computer to upload.	The IPT reads the uploaded Darwin Core Archive and performs the following steps: a) If a metadata file (e.g. eml.xml) is found in the DwC-A, it is read and the metadata section of the IPT RESOURCE is prefilled b) If the archive contains only a source file, and no meta.xml descriptor, then the header row of the source file is read and a mapping created c) If the archive contains a meta.xml descriptor, then this is translated into an IPT mapping d) Each DwC-A data file is configured to be a source file

		in the IPT RESOURCE The user is presented with the RESOURCE management console.
2	The user may choose to a) Edit the metadata b) Add additional source data files c) Add additional DwC mappings d) Modify existing DwC mappings	

3.2.5 Use case: Make RESOURCE public

This use case captures a user wishing to allow public access to a RESOURCE in the IPT.

Prerequisites		
A RESOURCE has been created, metadata filled, and any source data configured and mapped.		
Basic flow		
Step	*Actor action*	*System action*
1	From the RESOURCE management console, the user selects to make the RESOURCE public.	The IPT saves the configuration, and relaxes all security principles, so that the RESOURCE may be accessible by URL without authorisation

3.2.6 Use case: Publish RESOURCE

This use case captures a user wishing to PUBLISH a configured RESOURCE.

Prerequisites		
A RESOURCE has been created, metadata filled, and any source data configured and mapped.		
Basic flow		
Step	*Actor action*	*System action*
1	From the RESOURCE management console, the user selects to make the RESOURCE public.	The IPT performs the following: a) The metadata document is versioned and created b) If any source data has been

		mapped, a DwC Archive is created

3.2.7 Use case: Add manager to RESOURCE

This use case captures a user wishing to associate an additional user with MANAGER role to a RESOURCE, and thus enable them to configure the RESOURCE metadata and settings.

Prerequisites		
A RESOURCE has been created, and more than 1 user with managerial permission exist		
Basic flow		
Step	*Actor action*	*System action*
1	From the RESOURCE management console, the user chooses the additional manager to add to the RESOURCE.	The IPT stores the configuration, allowing the other user to manage the RESOURCE when they are authenticated with the IPT.

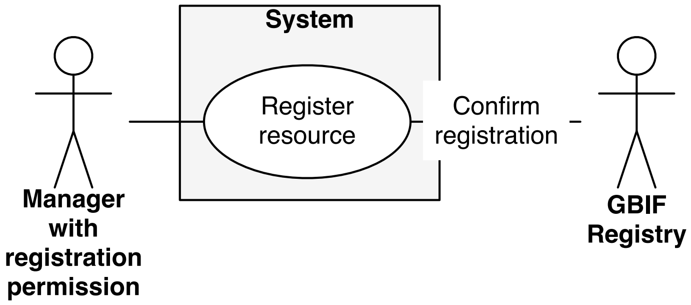

Figure 4: Managerial registration use case

3.2.8 Use case: REGISTER RESOURCE

This use case captures a user wishing to register a public RESOURCE with the GBIF REGISTRY.

Prerequisites
A RESOURCE has been created and configured, and made PUBLIC. The ADMIN has associated the necessary ORGANISATIONS with the IPT and all the ORGANISATION passwords are correct, and in synchronisation with those found in the GBIF REGISTRY; this will be the case unless changes are made through the GBIF Registry.
Basic flow

Step	*Actor action*	*System action*
1	From the RESOURCE management console, the user chooses the ORGANISATION to which the RESOURCE should be associated, and selects to register	The IPT communicates with the GBIF REGISTRY and creates the new registration. The RESOURCE state is moved to the registered state, after which it cannot be changed, other than by deletion.

3.3 User use cases

The IPT 2.0 provides limited RESOURCE interfaces. This section of the document is largely redundant for the IPT version 2.0, but serves as a placeholder for future expansion.

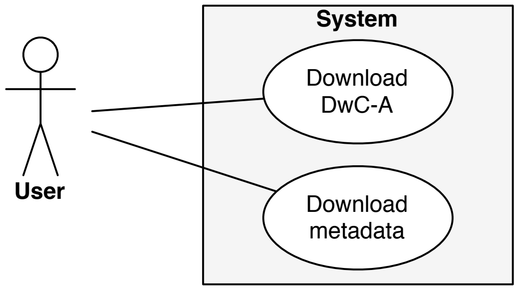

Figure 5: User use cases

3.3.1 Use case: Download DwC-A

This use case captures a user wishing to download a RESOURCE DwC-A.

Prerequisites		
The RESOURCE is PUBLISHED and PUBLIC		
Basic flow		
Step	*Actor action*	*System action*
1	The user calls the URL of the RESOURCE	The DwC-A is returned

3.3.2 Use case: Download Metadata

This use case captures a user wishing to download a RESOURCE metadata document.

Prerequisites		
The RESOURCE is PUBLISHED and PUBLIC		
Basic flow		
Step	*Actor action*	*System action*

1	The user calls the URL of the metadata document	The metadata is returned

3.4 Scenario depicting a *typical* use of the IPT

The following describes a typical use of the IPT:

a) An ADMIN installs the IPT onto a server with a public IP address.
b) The ADMIN dedicates a directory on the server that will act as the data directory for all resources published through the IPT. This directory will have sufficient privileges for the IPT application to write files to the directory. (The ADMIN might consider ensuring that this directory is included in their backup plan for disaster recovery).
c) The ADMIN configures the ADMIN email address and password and the location of the data directory.
d) The ADMIN enables the option to register RESOURCES with GBIF, and associates the IPT installation with the ORGANISATION hosting the IPT installation. During this phase the ADMIN will ensure that the URL at which the IPT is available at on the Internet is visible.
e) Because the INSTITUTION hosting the IPT will act as a virtual host for a second INSTITUTION, the ADMIN enables an additional INSTITUTION in the IPT, which will subsequently be available for data MANAGERS to associate their data resources with.
f) The ADMIN creates accounts for the users who will act as RESOURCE MANAGERS, and contacts them to issue their credentials for access.
g) A MANAGER logs into the IPT, and configures a new RESOURCE:
    i. A new RESOURCE of type "occurrence" is selected
    ii. The metadata about the RESOURCE (contact information, citation, sampling methods etc) are authored
    iii. A comma separated values (CSV) data file on the MANAGERS desktop computer is uploaded through the IPT web interface. This is considered the CORE SOURCE data file as it holds the occurrence details.
    iv. A comma separated values (CSV) file on the MANAGERS desktop computer holding image URLS of the specimens is uploaded. This is considered an EXTENSION SOURCE data file, as it holds information extending the records in the CORE file.
    v. The MAPPING is created to configure the fields from the CORE SOURCE file against the Darwin Core standard
    vi. The MAPPING is created to configure the fields from the EXTENSION SOURCE file against the multimedia EXTENSION

    Note: At this stage the MANAGER has now provided the core data and the means for the IPT to understand them

    vii. The MANAGER now PUBLISHES the data through the IPT. During this stage the IPT will create a Darwin Core Archive for the RESOURCE.

Note: Following a PUBLISH event, the RESOURCE is considered in a PRIVATE state. This means that the RESOURCE is only accessible to USERS logged with the privileges to view the RESOURCE. When satisfied the data is mapped correctly, the MANAGER selects to make the RESOURCE PUBLIC, where any user can view the RESOURCE on the IPT.

viii. With the RESOURCE in a PUBLIC state, the MANAGER selects to REGISTER the RESOURCE with GBIF. During this phase the MANAGER must choose the INSTITUTION to which the RESOURCE is associated. The ADMIN has already enabled 2 INSTITUTIONS from which the MANAGER may select. Should the MANAGER believe this to be insufficient, they may contact the IPT ADMIN to enable further INSTITUTIONS.

4 Logical architecture

[The logical view is concerned with the functionality that the system provides to end-users]

The IPT is a multi-user application, whereby users are granted permissions through the assignment of a role. These roles are described in Table 2 after which the logical architecture is separated into the functionality offered to each category of user.

Role	*Permissions*	*Included roles*
ADMINISTRATOR (ADMIN)	Configuration of all IPT settings. - User management - Registration options - Management of organisation relations - Management of extensions	MANAGER, USER
MANAGER (*WITH REGISTRATION PERMISSION*)	Ability to REGISTER a RESOURCE with the GBIF network	USER, MANAGER (*WITHOUT REGISTRATION PERMISSION*)
MANAGER (*WITHOUT REGISTRATION PERMISSION*)	Ability to create, edit and delete data resources to which they are associated. Ability to add or remove users with MANGER permission to a RESOURCE.	USER
USER	No permissions. An ADMINISTRATOR may demote a MANAGER to a USER. A user account may only be deleted if they are not associated with a RESOURCE. The user account that created a RESOURCE can never be removed from the RESOURCE, and therefore can not be deleted, but MANAGERIAL permissions may be removed by changing the role to USER.	

Table 2: User roles and permissions

Note: This document uses the term MANAGER in reference to both managerial roles. However, only MANAGERS explicitly allowed by an ADMIN may register resources with GBIF.

This dual functionality for the Manager permissions originated from the Finnish Participant Node Manager as a request to the IPT users mailing list. For their scenario, the Node Manager wishes to have many managers configuring and publishing data, but only a select few acting in a moderator capacity to verify that the mappings are indeed correct, prior to publishing through GBIF.

In other scenarios, IPT users do not want this moderation of data MANAGERS.

To support both uses, the Managerial role was split into 2 roles.

4.1 IPT Administration (Administrative user functionality)

4.1.1 Initial settings

When an ADMIN installs the IPT for the first time, they are required to set the mandatory fields for operation of the IPT. Until these settings are successfully configured, the IPT will not allow any further operation. These settings are listed in the following sections.

4.1.1.1 Data directory

On first installation, the IPT will request that the location of the data directory is specified. If the location supplied already contains an IPT configuration, then this will be read, tested and used, otherwise a new one will be created. The IPT will use this directory for all storage including source data files, user accounts, published resources and extension definitions.

The ADMIN should therefore ensure appropriate permissions to this folder; the IPT server must be able to write the directory, and it should be suitably restricted for access to other machine users. Additionally, the ADMIN should consider a routinely backup for this directory.

4.1.1.2 ADMIN account

Following the successful creation of a data directory, the IPT requires that the ADMIN account be created. Email addresses are used to identify IPT accounts and the ADMIN will be required to enter the email address, name and password associated with the ADMIN account.

4.1.1.3 Installation type

During installation, the administrator can select whether the installation is a live production installation or a test installation. The IPT will store this in the configuration so that options, such as REGISTRATION, operate against the GBIF test REGISTRY should a test installation be chosen. On selection the user will be informed that no further changes are allowed, and to move from TEST to LIVE would require a reinstallation and re-mapping of data. A TEST installation is suitable for training courses and for evaluation purposes.

This decision was deemed necessary due to potential assignment and storage of GBIF REGISTRY identifiers within the IPT that will differ between LIVE and TEST registries and potentially result in unmanageable complexity for subsequent synchronisation to the LIVE REGISTRY. It is anticipated users will install a TEST IPT and evaluate before deploying an IPT proper as LIVE.

4.1.1.4 Base URL

The administrator will be able to manually configure the BASE URL of the IPT, which is the URL at which the IPT is accessible by others. In the simplest deployment, the IPT could be installed onto a server with an IP address (e.g.10.20.30.40) and started using a chosen port option of (e.g. 8080). The base URL of this IPT installation will therefore be http://10.20.30.40:8080/ipt. A more complex installation of the IPT into an Apache Tomcat server might mean the BASE URL is http://10.20.30.40:8080/ipt but could be configured using virtual host definitions, and DNS records such that the application is also addressable using http://ipt.mybif.com/ and it is this address through which external users should access the machine. Because it is not possible for the IPT to detect the actual deployment and the preferences of the person deploying the IPT, the ADMIN is required to configure this URL in the IPT manually.

The IPT will default to detecting the server IP address, and the port in use, and initialise with http://<ipaddress>:<port>/ipt which will suffice many installations.

It is the responsibility of the ADMIN to ensure that external access to the IPT is possible and all required firewalls (etc) are configured to allow access on the Internet / Intranet.

4.1.2 REGISTRY Configuration

The IPT will communicate with the GBIF REGISTRY through the RegistryAPI[9], to allow for REGISTRATION of the IPT instance, and any RESOURCE that has been PUBLISHED through the IPT installation. By REGISTERING the IPT and its RESOURCES, GBIF will automatically index the content so that it may be found through the global discovery services offered through the GBIF Data Portal.

By default the REGISTRATION is disabled, and the ADMINISTRATOR is required to enable this option. When disabled, data MANAGERS will not be offered any option during RESOURCE configuration to register the RESOURCE with GBIF. Once enabled, the MANAGERS will be able to move a PUBLIC RESOURCE into a REGISTERED state, during which time the GBIF REGISTRY will be informed of the RESOURCE location. Once enabled, an ADMINISTRATOR cannot disable the registration option, and will be warned of this during the enabling of the option.

The decision to prevent an administrator from disabling registration is to avoid the situation whereby resources are deregistered at GBIF and thus seen as deleted. This has significant impacts with global indexes as it can result in many cascading deletions. Therefore should an administrator wish to delete resources they should explicitly do a delete operation on the RESOURCE.

[9] http://code.google.com/p/gbif-registry/wiki/OrganisationAPI

To enable the registration, the following sequence of events occurs:

a) The administrator will log in and from the administration menu, select the GBIF registration configuration option

b) The administrator will be presented with a message describing that this option will allow data managers the option of publishing the resources onto the GBIF network during the RESOURCE configuration stage.

c) The BASE URL will be shown, with an option to change, save and test this. During the test the following sequence will happen

   - The IPT will issue a RegistryAPI call, with a parameter that includes a *call-back* URL of the form http://<baseURL>/rss.do
   - The registry will then issue an HTTP GET to the *call-back* URL and confirm that an HTTP 200 is returned.
   - If anything other than an HTTP 200 is returned to the registry, the registry will indicate to the IPT that the test failed, and the IPT will inform the administrator that the IPT is not visible on the public Internet, and therefore not permissible for registration. Should it be successful, the ADMINISTRATOR will be allowed to continue.

   Note: This test is necessary to avoid registrations of Resources that are not accessible by others and to ensure that the IPT is configured correctly to avoid unnecessary manual registry data management.

d) The ADMINISTRATOR will be asked to select the ORGANISATION or INSTITUTION that is considered responsible for hosting the IPT installation, with the list provided by the GBIF registry through the RegistryAPI. Should the ADMINISTRATOR not find the ORGANISATION they seek, they will be prompted to Contact the GBIF Helpdesk to have this ORGANISATION created.

This model was chosen because during IPT testing, many duplicate ORGANISATIONS have been created, causing an unnecessary burden of Registry management. It is felt that on an organisation level the volume of new registrations are manageable (currently there are 300 ORGANISATIONS) and centralised management will prove to be cleaner and more manageable than allowing creation by all. This model will be reviewed in a later version of the IPT.

e) Once the organisation is selected, the ADMINISTRATOR will be required to enter the ORGANISATION password, or contact the ORGANISATION technical contact through the provided email address to obtain the password.

f) With the Base URL test complete, the ORGANISATION selected and the Password entered the ADMINISTRATOR can then choose to REGISTER the IPT with GBIF. During this registration, the following will happen

   - The IPT calls the RegistryAPI indicating that a new IPT installation is occurring. It will contain the organisation key and password, and the base URL for the IPT.
   - The registry will confirm the password supplied is correct and only continue if it is valid.

- The registry will re-perform the echo test to confirm the base URL is accessible and will register the IPT and the RSS service for the IPT.
- The IPT will receive confirmation of registration with the REGISTRY and will store the necessary REGISTRY keys in the IPT configuration.

4.1.3 ORGANISATION association

If the IPT is not registered with GBIF then this section is redundant. ORGANISATIONS may only be associated with an IPT after the registry options have been enabled.

The purpose of allowing multiple ORGANISATIONS within an IPT is to allow for shared data hosting capabilities within a single installation. An IPT hosted at one ORGANISATION may be used to create, configure and REGISTER data RESOURCES on behalf of secondary ORGANISATIONS. During REGISTRATION, the association of the RESOURCE to the secondary ORGANISATION is captured, to ensure that the relationship is preserved and visible on the GBIF network. This is a common use case within the GBIF network, where shared hosting is very cost effective.

If the ADMINISTRATOR has successfully registered the IPT with GBIF, then they may choose to configure the list of ORGANISATIONS that will be available for selection during the REGISTRATION phase.

> Restricting the list at the ADMINISTRATION level, rather than allowing all ORGANISATIONS to MANAGERS was considered to be a more user-friendly experience for the MANAGER who may find a large selection overwhelming and to enable fine-grained control by the ADMINISTRATORS

This is achieved through the following workflow:

a) The ADMINISTRATOR logs in and from the administration menu, select the ORGANISATION configuration option
b) The ADMINISTRATOR is shown a table of existing associated ORGANISATIONS, which will initially show only the ORGANISATION to which the IPT has been registered with
c) The ADMINISTRATOR can search for further ORGANISATIONS (as per the original installation) or contact GBIF Helpdesk to have them registered and made available.
d) When selecting an additional ORGANISATION, the ADMINISTRATOR will be required to supply the ORGANISATION password. Additionally the ADMINISTRATOR will supply an alias name for the ORGANISATION, which will be the title of the ORGANISATION shown in this installation of the IPT. At any time, the ADMINISTRATOR can modify the alias name. The purpose of this alias name is to allow usage of ORGANISATION names that are meaningful to users who will use the IPT. For example, an alias in the native language of the users might be chosen.

The ADMINISTRATOR will be able to delete from the list of associated organisations, only if there are no Resources configured against those ORGANISATIONS. The ADMINISTRATOR can disable the ORGANISATION from being used in further RESOURCE administration. If disabled, then Managers will no longer be able to select that ORGANISATION when creating a RESOURCE. The Managers will be prompted to contact the IPT ADMINISTRATOR if they believe the available ORGANISATIONS are incomplete.

4.1.4 Extension management

The IPT supports the Darwin Core Archive output format, which is comprised of a single core file, and a file per extension. The definitions of the terms available in each extension are governed by the profiles registered in the GBIF registry. On initial installation the IPT has no EXTENSIONS installed. The ADMIN is required to install EXTENSIONS, which will then be available for MANAGERS to use during RESOURCE configuration.

The IPT communicates with the GBIF REGISTRY to discover the existence of EXTENSIONS. This communication occurs through the Registry ExtensionAPI[10] interface. An EXTENSION is considered immutable, and once known to the IPT, is cached in the data directory.

An EXTENSION may only be uninstalled only if it is not in use by any RESOURCE mapping.

4.1.4.1 Vocabulary management

An EXTENSION may declare a recommended controlled VOCABULARY to use for a TERM being mapped. When an EXTENSION references a VOCABULARY, it is automatically installed into the IPT. This installation occurs through the Registry ExtensionAPI. VOCABULARIES are mutable, and therefore the ADMIN may periodically choose to update all vocabularies known the IPT installation.

4.1.5 User management

A user account may be administered by the account holder, or by any ADMIN. A user account is identified by an email address that cannot be changed, and has a name and password, which may be altered. In addition, an ADMIN may define the ROLE applicable (ADMIN, MANAGER (with or without publishing permission), USER), which determines the permissions for a user. Should a user's ROLE be demoted, any RESOURCES already created by the user will remain, but the user may no longer have permission to MANAGE that RESOURCE.

There will always be at least one ADMIN user. A user may only be deleted if they are not associated with any RESOURCE, and are not the only ADMIN user.

[10] http://code.google.com/p/gbif-registry/wiki/ExtensionAPI

4.2 RESOURCE management (Managerial user functionality)

The managerial functionality allows for the creation and configuration of a data RESOURCE. The typical lifecycle of RESOURCE configuration follows the sequence:

1. RESOURCE is created with a unique short name
2. Basic mandatory descriptive metadata is authored
3. Optionally, extensive metadata is authored

Items 4-5 may be skipped should no data be available for the RESOURCE

4. Source data for the RESOURCE are defined by either uploading text files, or configuring connections to the source databases. A combination of both might be applicable.
5. Each source defined is mapped to either the core extension (e.g. the Darwin Core Occurrence or Taxon extension) or to a community defined extension
6. The RESOURCE is published to create the Darwin Core Archive (DwC-A) and metadata
7. The RESOURCE is made PUBLIC to allow others to view and download the DwC-A and Metadata
8. Optionally, the RESOURCE is associated with an ORGANISATION and registered to the GBIF network
9. Optionally further MANAGERS are permitted to configure the RESOURCE

The following sections detail specific functionality in this sequence.

4.2.1 RESOURCE creation

When a RESOURCE is created, a MANAGER must supply a *short name* and optionally may upload a Darwin Core Archive (DwC-A), which will be read and used to configure the RESOURCE. The SHORTNAME is significant as it is used in URLs relating to the RESOURCE. An IPT with a base URL of http://ipt.mybif.org and RESOURCE of *short-name* mammals will therefore have its metadata addressable on http://ipt.mybif.org/eml.do?r=mammals.

If a DwC-A is supplied when the RESOURCE is created, then any EML or DublinCore metadata contained is read and used to populate the RESOURCE metadata, the core and extension files become source data for the RESOURCE, and the DwC-A meta.xml is read and a mapping created. Should the meta.xml contain references to extensions unknown to, or not installed in the IPT, then no mapping is created and the files remain available as source files.

4.2.2 Metadata authoring

The IPT provides means to author metadata in accord with the GBIF Metadata Profile which is documented in detail in <insert document reference>. A brief summary is provided in Table 3.

Basic metadata

Title	Full title for the RESOURCE
Description	The description of the dataset
Metadata language	The language in which the metadata is written in
Resource language	The language of the data in the dataset, to which the metadata relates
Sub type	A controlled vocabulary specifying the type of RESOURCE that the metadata describes (e.g. checklist, observation etc)
Resource contact	Details of the individual, or ORGANISATION who should be the primary contact for the RESOURCE. This may not necessarily be the owner, or creator of the RESOURCE.
Resource creator	Details of the individual, or ORGANISATION who is considered the primary party responsible for creation of the RESOURCE described by the metadata. This may not necessarily be the provider of the metadata.
Metadata provider	Details of the individual, or ORGANISATION who is considered the primary party for creation of the metadata being authored. This is typically, but not necessarily, the user logged in to the IPT
Geographic coverage	
Min. / max. latitude / longitude	The coordinates (in WGS84 datum) that represent the minimum bounding-rectangle of the RESOURCE data.
Description	A textual description of the geographic bounds of the RESOURCE data
Taxonomic coverages (all fields are repeatable as blocks)	
Description	A textual description of the taxonomic scope of the RESOURCE
Scientific name	The scientific name (Latin) of the taxon that is covered by the RESOURCE data
Common name	The vernacular / common name of the taxon that is covered by the RESOURCE data
Rank	The rank at which the scientific / common name apply (e.g. Family)
Temporal coverages (all fields repeatable)	
Single date	The single calendar date on which the RESOURCE data were collected or sampled
Living time period	Time period during which biological material was alive. Includes paleontological time periods or other text phrases

Formation period	Text description of the time period during which the collection was assembled e.g. "Victorian", or "1922 - 1932", or "c. 1750"
Date range	2 single calendar dates that represent the start and end days of the data collection or sampling
Other keywords (all fields are repeatable as blocks)	
Thesaurus	A name for the keyword thesaurus from which the keywords were derived. Keyword thesauri are usually discipline specific and can be custom or official
Keywords	Keywords that concisely describes the RESOURCE or are related to the RESOURCE
Associated Parties (all fields are repeatable as blocks)	
Multiple fields	Details of the individual, or ORGANISATION who is considered an associated party to the RESOURCE
Role	A controlled vocabulary of terms describing the nature of the parties association such as Author, custodian steward etc
Project Data	
Title	The title of the project
Personnel first / last name / role	The primary person and their associated role with the project
Funding	Description on the project funding
Study area description	Documents the physical area associated with the research project. It can include descriptions of the geographic, temporal, and taxonomic coverage of the research location. A project might have a larger scope than the dataset being described.
Design description	A general description in textual form describing some aspect of the study area
Sampling Methods	
Study extent	A textual description of the extent of study, which may be geographic, taxonomic or some other measure.
Sampling description	A textual description of the sampling employed
Quality control	A textual description of the quality control employed in the sampling
Step description (repeatable element)	A textual description of one stage of the sampling methods. Steps are sequential.

Citations	
Citation identifier	A persistent identifier that can be used as a citation. For example a DOI, or other persistent URI might be used as a citation identifier
Resource citation	A textual description that can be used verbatim to cite the RESOURCE
Bibliography (Bibliography sections are repeatable)	Allows citation identifiers, and textual descriptions to cite resources that were used to build an aggregate dataset. An example might be citing several taxonomic checklists, as the source for an aggregate checklist RESOURCE.
Collection Data	
Collection name	The name for the collection
Collection identifier	A persistent identifier that can be used as a reference to the collection. For example a DOI, or persistent URI might be used as a collection identifier
Parent collection identifier	If the collection is sub part of a bigger collection, a persistent identifier may be used as a reference to the parent collection.
Specimen preservation method	A controlled vocabulary describing how the specimens in the collection are preserved
Curatorial units (this block is repeatable)	Allows for the description of arbitrary ranges or counts with units related to the collection. Examples could be 7500-7600 specimens, or 7550 +/- 50 specimens
External links	
Resource homepage	A URL that points to the homepage for the RESOURCE being described
Downloadable items (repeatable block of elements)	Allows for the name, character set, URL, data format and version for downloadable data related to the RESOURCE
Additional metadata	
Date published	The date at which the dataset was considered published
Purpose	Summary of the intensions for which the dataset was developed. Includes objectives for creating the dataset and what the dataset is to support
IPR	A rights management statement for the RESOURCE, or reference a service providing such information
Additional	A textual block of extra relevant information pertinent to the

information	RESOURCE

Table 3: Summary of metadata fields supported in the IPT

4.2.3 Source data configuration

Source data represents the users' input data and can be defined by the result of an arbitrary SQL statement against a database, or can be provided by uploading a source data file. The source data is subsequently mapped as the core of the Darwin Core Archive (e.g. represents occurrence records or taxon records) or can be mapped to an installed IPT extension for the Darwin Core Archive. At the time of source definition, the IPT does not distinguish between core or extension types and is only concerned with ensuring that the source is accessible and can be read correctly.

4.2.3.1 File based sources

The IPT supports source data in the form of delimited text file. Delimited text refers to the format of files that have common characters that delimit fields and lines and use escape characters where necessary. Commonly used examples are comma separated value files and tab delimited value files, but the MANAGER may configure the IPT to recognize any delimiters through the RESOURCE configuration console. The IPT supports the user-defined values for the following properties.

Name	The name provides a handle to the file, and must be unique to the RESOURCE. To overwrite an existing source file, a new source must be created with the same name
Number of header rows	If the source contains header rows (e.g. the column names from an excel spreadsheet) then number of header rows can be set to instruct the IPT to skip those rows of data
Field delimiter	The field delimiter declares the separating character between fields in a single row of data. Commonly used delimiters are comma, tab and pipe (|)
Field quotes	Should the source data quote records to escape fields that use the delimiter, then the field quotes defines the character used to indicate this. Consider the sharing of *Copenhagen, Denmark* in a comma delimited file; since the field itself has a comma, then it must be quoted. Typically the " character is used to quote fields, and the example would be represented as *"Copenhagen, Denmark"*
Character encoding	Allows the configuration of the encoding (UTF-8, Latin1, Windows 1252 etc) used in the source data. The IPT will attempt to infer this, but it is not always failsafe
Date format	Should the source data contain date fields, the format may be declared to ensure correct handling. It is not failsafe to infer all date formats; E.g. 01/02/11 could be many

	possibilities including: - 1st February 2011 - 2nd January 1911 (or another century) - 11th February 2001

Table 4: Delimited text configuration options

4.2.3.2 SQL Database sources

The IPT supports source data defined by running a SQL statement against a relational database. The IPT supports the user-defined values for the following properties:

Name	The name provides a handle to the file, and must be unique to the RESOURCE. To overwrite an existing source definition, a new source must be created with the same name
Database system	The following database systems are supported - MySQL - Oracle - PostgreSQL - Sybase - Generic ODBC connections (Microsoft MS SQL, Microsoft Access etc) Note: Database connectivity is provided by Java's JDBC connection mechanism.
Host	The host database server name or IP address, and port if necessary. The route to the host must be available, with no firewalls blocking access. For ODBC connections this is not required
Database	The database name on the server or the ODBC name should ODBC be used
Database user	The username used in the database connection
Database password	The password used in the database connection
SQL Statement	The complete SQL statement to issue against the database to retrieve the source data. The IPT will issue the statement verbatim, allowing simple select statements as well as complex SQL queries including joins or functions as supported by the underlying database system
Character encoding	Allows the configuration of the encoding (UTF-8, Latin1, Windows 1252 etc) used in the source data. The IPT will attempt to infer this, but it is not always failsafe

Date format	Should the source data contain date fields, the format may be declared to ensure correct handling. It is not failsafe to infer all date formats; E.g. 01/02/11 could be many possibilities including: - 1st February 2011 - 2nd January 1911 (or another century) - 11th February 2001

Table 5: Relational database configuration options

4.2.4 Mapping of source data to Darwin Core terms

Once configured as source data, the IPT does not distinguish between SQL and file based sources with respect to the mapping phase. A source may be considered a core file, or an extension file, as per the Darwin Core Archive structure.

4.2.4.1 Core source file mapping

A core source file is considered the core of the star schema and will either be mapped such that each record represents an Occurrence record, or a Taxon concept record as chosen by the user. The core record definitions are treated by the IPT in the same manner as any community defined extension definition, but are modeled according to the Darwin Core definitions of the classes. The core extension definitions are controlled by the following profiles:

Occurrence record	http://rs.gbif.org/core/dwc_occurrence.xml
Taxon record	http://rs.gbif.org/core/dwc_taxon.xml

Table 6: Core extension definitions

It should be noted that only those extensions or core definitions made available by the ADMIN are available to the MANAGERS.

4.2.4.1.1 Assigning identifiers

Each record in the core file may be assigned with an identifier as follows:

Mapping to source data field	A user-selected field is chosen from the source file to use as a unique identifier for the record. This is the only identifier type that is applicable if extensions are to be used.
Line number	An automatically incrementing integer identifier is

	assigned to each record
UUID Generator	A UUID[11] is automatically minted and assigned to each record by the IPT
No ID	No identifier is assigned to the record

Table 7: Identifier assignment during mapping

Should a record identifier mechanism be mapped or chosen, then it will be mapped to the DwC:occurrenceID or DwC:taxonID property in the output file.

4.2.4.1.2 Filtering input data

Optionally, a filter may be supplied during the mapping to sub-select the input results. A filter consists of the source field, a predicate and optionally a value. The filter predicates types supported are:

Is Null	Only records with a NULL value for the chosen field are included. E.g. records where "year is null"
Is Not Null	Only records with a non-NULL value for the chosen field are included. E.g. records where "year is not null"
Equals	Only records with the value supplied for the chosen field are included. E.g. records where "year is 1999"
Not equals	Only records with anything but the value supplied for the chosen field are included. E.g. records where "year is not 1999"

Table 8: Supported filter predicate types

4.2.4.2 Extension source files

Extension file mappings are the same as the core file mapping, with the difference being that the join column must be selected. The join column represents the column in the extension source file that holds the foreign key to identifier in the core file.

4.2.5 RESOURCE visibility state (PUBLIC, PRIVATE, REGISTERED)

A RESOURCE always has a visible state, which is used to define the access policy to apply. A RESOURCE MANAGERIAL user or an ADMINISTRATIVE user is required to modify a RESOURCE state. The state transition table for resources is shown in Table 9.

State	Purpose	Possible subsequent States

[11] http://en.wikipedia.org/wiki/Universally_unique_identifier

PRIVATE	On first creation, a RESOURCE is considered PRIVATE, and can only be viewed by the ADMIN and MANAGERS with sufficient privileges for the RESOURCE.	PUBLIC
PUBLIC	A MANAGER of a PRIVATE RESOURCE can make a RESOURCE PUBLIC, meaning that the URLs to view and access the RESOURCE information are open to all with no access control applied. A PUBLIC RESOURCE may be moved to a PRIVATE state to re-enable the access control.	PRIVATE, REGISTERED
REGISTERED	Only PUBLIC resources may become REGISTERED, and only if the ADMIN has enabled the GBIF registration option. During the transition to REGISTERED, the associated ORGANISATION is selected by the user. The IPT will use the RegistryAPI to communicate the service access URLs for the resources. Once REGISTERED a RESOURCE may be deleted, but further state changes are disallowed.	No further changes allowed

Table 9: RESOURCE state transitions

- The rationale for disallowing further transitions from PUBLISHED, is to avoid excess change in the GBIF registry. A RESOURCE becoming un-PUBLISHED effectively is the same as a deletion, and could have significant impact on global indexes. Large amounts of data and derived metrics could need removed, or recalculated which may then be accessible shortly afterwards should the IPT MANAGER REGISTER the RESOURCE again. This situation is not desirable and the MANAGERS should be encouraged to take due consideration prior to PUBLISHING.

4.2.6 Publishing a RESOURCE

The publishing of a RESOURCE is initiated by a user with ADMINISTRATIVE or MANAGERIAL permissions, and instructs the IPT to build the RESOURCE Darwin Core Archive, by applying any data mappings to the source data. Should any subsequent changes be made to RESOURCE metadata, source data or data mappings, then the user must initiate a new publishing of the RESOURCE.

4.2.7 Deleting a RESOURCE

The deletion of a RESOURCE is initiated by a user with ADMINISTRATIVE or MANAGERIAL permissions, and results in the IPT permanently deleting the RESOURCE directory, including all source data, data mappings, metadata and Darwin Core archives. Additionally, if the RESOURCE has been registered, then the GBIF REGISTRY is informed of this through the GBIF RegistryAPI.

5 Discovery and access services (User level permissions)

The discovery and access services are accessible by users logged in with USER level permission, or any users who are not logged in. Only resources in a state of PUBLIC or REGISTERED (see Table 9: RESOURCE state transitions)

5.1 EML Metadata

The EML metadata is available of the URL http://<IPT_BASE_URL>/eml.do?<RESOURCE_SHORT_NAME> and is in the format defined by the GBIF Metadata Profile schema[12].

5.2 Darwin Core Archive

The RESOURCE Darwin Core Archive is available on the URL http://<IPT_BASE_URL>/archive.do?<RESOURCE_SHORT_NAME>.

5.3 RSS Feed

An RSS feed provides a list of the publicly available resources published through the IPT, and is available on the URL http://<IPT_BASE_URL>/rss.do. The feed contains links to the EML metadata and the RESOURCE overview.

[12] http://rs.gbif.org/schema/eml-gbif-profile/

6 Development architecture

[The development view illustrates a system from a programmer's perspective and is concerned with software management]

The IPT software architecture is shown in Figure 6.

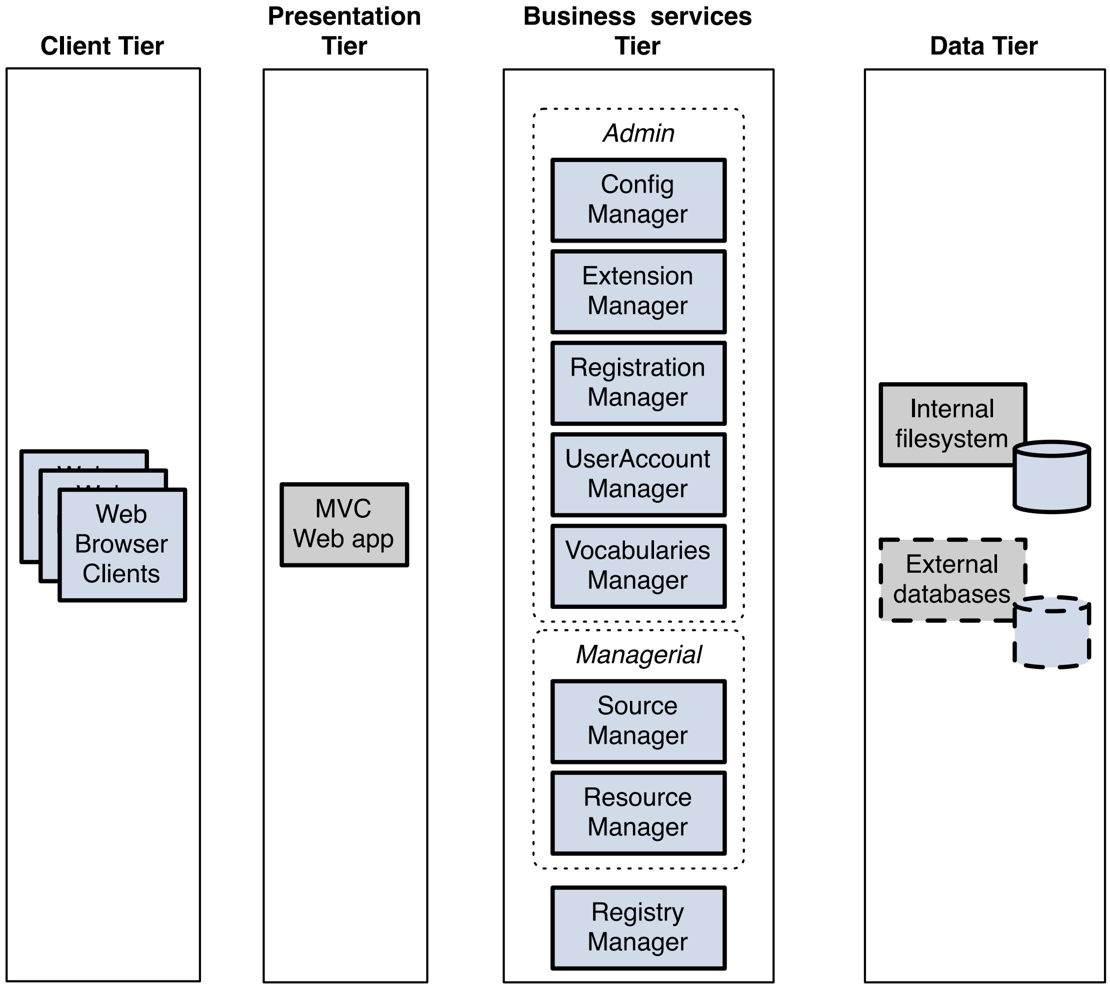

Figure 6: IPT Software Architecture

Note that the IPT Data Tier includes external databases, to which the IPT connects, allowing for import of data onto the internal file system.

- The ConfigManager is responsible for handling all IPT configurations, including the location of the data directory itself, and the persistence of IPT configuration information to file.
- The ExtensionManager provides means to inventory and install DwC extensions within the IPT to make them available for MANAGERS to use in RESOURCE configuration.
- The RegistrationManager provides means to register the IPT with the GBIF REGISTRY, and associate the IPT with one or more ORGANISATIONS connected to the GBIF network.

- The UserAccountManager is responsible for all operations on the accounts of users, including the assignment of roles.
- The VocabulariesManager provides means to update all vocabularies installed in the IPT. Vocabularies are referenced from DwC extensions and therefore may be automatically installed when an extension is installed.
- The SourceManager provides means to
  - Describe and import delimited Files to the data directory
  - Connect to databases, and allow an importing from a table or view through the definition of a SQL statement into a file in the data directory, and describe the contents of the result set
  - Delete source files
  - Preview lines of source files
- The ResourceManager provides:
  - Create, delete, update and get a RESOURCE in the IPT
  - List resources, optionally by providing filters such as user or status
  - Change the visibility status of a RESOURCE
  - Update the Registration information about a RESOURCE
- The RegistryManager provides a simple mechanism to interact with the GBIF REGISTRY web services API

6.1 Data directory

The data directory is the location on the server where all data and configuration information are stored in the IPT with the exception of the Data Directory location itself.

Once the administrator has configured the data directory, the IPT should perform tests to ensure that the directory exists and that the IPT has permissions to read and write to the directory. If the directory already exists and contains an IPT configuration already, the user should be displayed a message that the IPT was restored with an existing data directory and that all settings were preserved. Should an administrator wish to later change the location of a data directory, the administrator will be required to move the data directory manually, and change the directory location through the IPT configuration, whereby the IPT will inform them that the IPT was restored from an existing data directory.

The structure of the data directory is shown:

Directory	Content
/config	Holds all IPT configuration
/.extensions/	Holds the definition of installed extensions
http_…occurrence.xml	Extension definition
.gbifreg	Holds the chosen preference if the IPT is considered in development or live use
/.vocabularies	Holds the definitions of the vocabularies referenced by the extensions
http_...basis_of_record_xml.vocab	Vocabulary definition
about.ftl	The information shown to the user on the IPT "About" page. Can be customised using standard HTML
ipt.properties	Stores the IPT administrative settings

users.xml	The users and allocated permissions. Passwords are encoded using the MD5 algorithm.
/logs	Holds the application logs
admin.log	Log events triggered by administrative users
debug.log	Verbose log events
/resources	Holds each of the Manager created resources
/*resource_short_name*	Each RESOURCE resides in a directed, with a name equal to the user chosen *RESOURCE short name*
dwca.zip	The last DwC-A published by the IPT
eml-*version*.xml	Each publication generates a new *version* of the metadata.
eml.xml	The latest version is always copied (e.g. duplicated) to the unversioned file. Therefore after 2 publication events, eml-2.xml will be the same file as eml.xml
resource.xml	All RESOURCE configuration, including the source file definitions and data mappings
/sources	The uploaded source data files
source_name.extension	Files uploaded retain the original name
/tmp	Temporary files created by the IPT during DwC-A creation

Table 10: Data directory structure

7 Process architecture

[The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the runtime behaviour of the system]

The IPT version 2.0 is a single server application with limited processes. For version 2.0, this section is restricted to the runtime behaviour of the IPT in communication with the GBIF REGISTRY.

[Future versions of the IPT are expected to include communications with more external services (e.g. quality control, and annotation brokerage), which will be documented here.]

7.1 REGISTRY communication

The GBIF REGISTRY API is a RESTful[13] service returning JSON[14]responses. The RESOURCE definition in the API ensures that all operations are atomic; i.e. they can be completed in a single HTTP request/response. This ensures that no transactions span multiple requests and risk inconsistent state should communication fail mid-transaction. This is a significant deviation from the IPT 1.0, which was prone to network outages, particularly in areas of low bandwidth.

7.1.1 Authentication

The GBIF RegistryAPI enforces the use of HTTP Basic Authentication[15] for all operations that result in modification of REGISTRY content. This model of authentication dictates that a username and password be supplied in an HTTP header in the form of:

```
Authorisation: Basic <Base64Encoded username:password>
```

The username is the REGISTRY identifier for the entity (Agent) being *modified* and the password is the REGISTRY password for the same entity. A *modification* is considered as any change in property (e.g. a title, or email address) or the connection of a new entity to an existing one (e.g. a new IPT data RESOURCE being registered as coming from an ORGANISATION).

A summary of the REGISTRY operations and necessary authentication parameters follow.

7.1.1.1.1 RegistryAPI: GetExtensions()

A call to this service returns the details of all registered extensions, and the means to access them.

Example request: http://gbrdsdev.gbif.org/registry/extensions.json

[13] http://en.wikipedia.org/wiki/RESTful

[14] http://en.wikipedia.org/wiki/JSON

[15] http://en.wikipedia.org/wiki/Basic_access_authentication

Response:

```
{
extensions: [
{
  url: "http://rs.gbif.org/core/dwc_occurrence.xml"
  title: "Darwin Core Occurrence"
  subject: ""
  description: "The category of information pertaining to evidence of
an occurrence in nature, in a collection, or in a dataset (specimen,
observation, etc.)."
  identifier: "http://rs.tdwg.org/dwc/terms/Occurrence"
}
{
  url: "http://rs.gbif.org/core/dwc_taxon.xml"
  title: "Darwin Core Taxon"
  subject: ""
  description: "The category of information pertaining to taxonomic
names, taxon name usages, or taxon concepts."
  identifier: "http://rs.tdwg.org/dwc/terms/Taxon"
}
…truncated for example…
]
}
```

7.1.1.1.2 RegistryAPI: getOrganisations()

A call to this service returns the list of ORGANISATIONS that are registered in the GBIF network, including the title and REGISTRY identifier in the response.

Example request: http://gbrdsdev.gbif.org/registry/organisation.json

Response:

```
[
{
name: "Academy of Natural Sciences"
key: "f9b67ad0-9c9b-11d9-b9db-b8a03c50a862"
}
{
name: "ACOI - Coimbra Collection of Algae - University of Coimbra"
key: "e92e0710-24c4-11dc-a625-b8a03c50a862"
}
…truncated for example…
]
}
```

7.1.1.1.3 RegistryAPI: getVocabularies()

A call to this service returns the summary of all registered vocabularies, and the means to access them.

Example request: http://gbrdsdev.gbif.org/registry/thesauri.json

Response:

```
{
thesauri: [
{
url: "http://rs.gbif.org/vocabulary/dcterms/type.xml"
title: "Dublin Core Type Vocabulary"
subject: ""
description: "The DCMI Type Vocabulary provides a general, cross-
domain list of approved terms that may be used as values for the
Resource Type element to identify the genre of a resource. The terms
documented here are also included in the more comprehensive document
"DCMI Metadata Terms" at http://dublincore.org/documents/dcmi-
terms/."
identifier: "http://dublincore.org/documents/dcmi-type-vocabulary/"
}
{
url: "http://rs.gbif.org/vocabulary/dwc/basis_of_record.xml"
title: "Darwin Core Type Vocabulary"
subject: ""
description: ""
identifier: "http://rs.tdwg.org/dwc/dwctype/"
}
}
…truncated for example…
]
}
```

7.1.1.1.4 RegistryAPI: validateOrganisation()

This service is used to validate that the credentials for an organisation are correct. The authentication username is the REGISTRY identifier for the ORGANISATION entity, and the password is the REGISTRY password for the organisation.

Example request:
http://gbrdsdev.gbif.org/registry/organisation/_UUID_?op=login

where _UUID_ is an UUID for an existing organisation.

and

```
Authorisation: Basic <Base64Encoded
organisationIdentifier:organisationPassword>
```

Response:

200	Success

Errors:

401	Authorization header should be provided for this method
401	Authorization header is invalid

7.1.1.1.5 RegistryAPI: registerIPT()

This service modifies the GBIF REGISTRY content with the existence of a new IPT installation, which must be associated with an existing organisation. The authentication username and password provided must be correct for the associated organisation for this method to succeed.

Example request: http://gbrdsdev.gbif.org/registry/ipt/register [HTTP POST]

with

```
Authorisation: Basic <Base64Encoded
organisationIdentifier:organisationPassword>
```

Parameters:

Field	Mandat.?	Field value	Description
organisationKey	YES	valid UUID & should be an existing one	UUID of the hosting institution/organisation .
name	NO	free text	Name of the IPT instance
description	NO	free text	Description of the IPT instance
language	NO	free text	Language of the installation of the IPT instance
homepageURL	NO	free text	Homepage that contains more information on the IPT
logoURL	NO	free text	Logo URL for the IPT instance
wsPassword	NO	free text	Password used for updating the IPT instance's metadata
serviceTypes	NO	free text	Multiple types allowed, but must match the order specified in the **serviceURLs** field. Types should be separated by pipe |. Example: RSS|DWC-ARCHIVE-OCCURRENCE|TAPIR
serviceURLs	NO	free text	Mutiple types allowed, but must match the order specified in the **serviceTypes** field. Example: http://www.example.org/obj.rss|http://www.example.org/dwc-archive.zip|http://www.example.org/tapir.php
primaryContactName	NO	free text	Name of the IPT instance's primary contact
primaryContactType	YES	**technical** or **administrative**	Type of the IPT instance's primary contact
primaryContactFirstName	NO	free text	First name of the IPT instance's primary contact
primaryContactLastName	NO	free text	Last name of the IPT instance's primary contact
primaryContactAddress	NO	free text	Address of the IPT instance's primary contact
primaryContactDescription	NO	free text	Description of the IPT instance's primary contact
primaryContactEmail	YES	**valid e-mail format**	Email of the IPT instance's primary contact
primaryContactPhone	NO	free text	Phone of the IPT instance's primary phone

Response:

201	Success

Errors:

401	Authorization header should be provided for this method
401	Authorization header is invalid
400	Organisation key must be supplied
400	The organisation key does not correspond to any existing organisation
400	Primary contact email must be supplied
400	Primary contact type must be supplied
400	Primary contact email must be in valid format
400	Primary contact type must be either **administrative** or **technical**.

7.1.1.1.6 RegistryAPI: updateIPT()

This service modifies the GBIF REGISTRY content by updating the metadata (title, description etc) for the provided IPT instance. The username and password supplied for authentication must be correct for the IPT being updated.

Example request: http://gbrdsdev.gbif.org/registry/ipt/update/__UUID__

[HTTP POST]

where _UUID_ is an UUID for an existing IPT instance.

with

```
Authorisation: Basic <Base64Encoded iptIdentifier:iptPassword>
```

Parameters:

Field	Mandat.?	Field value	Description
organisationKey	-	valid UUID & should be an existing one	UUID of the new hosting institution/organisation. **Note:** This feature is disabled at the moment but might be used in the future if needed.
name	NO	free text	Name of the IPT instance
description	NO	free text	Description of the IPT instance
language	NO	free text	Language of the installation of the IPT instance
homepageURL	NO	free text	Homepage that contains more information on the IPT
logoURL	NO	free text	Logo URL for the IPT instance

primaryContactName	NO	free text	Name of the IPT instance's primary contact
primaryContactType	YES	**technical** or **administrative**	Type of the IPT instance's primary contact
primaryContactFirstName	NO	free text	First name of the IPT instance's primary contact
primaryContactLastName	NO	free text	Last name of the IPT instance's primary contact
primaryContactAddress	NO	free text	Address of the IPT instance's primary contact
primaryContactDescription	NO	free text	Description of the IPT instance's primary contact
primaryContactEmail	YES	**valid e-mail format**	Email of the IPT instance's primary contact
primaryContactPhone	NO	free text	Phone number of the IPT instance's primary contact
serviceTypes	NO	free text	Multiple types allowed, but must match the order specified in the **serviceURLs** field. Types should be separated by pipe |. Example: RSS|ATOM
serviceURLs	NO	free text	Mutiple types allowed, but must match the order specified in the **serviceTypes** field. Example: http://www.example.org/obj.rss|http://www.example.org/atom.xml

Response:

200	Success

Errors:

401	Authorization header should be provided for this method
401	Authorization header is invalid
400	IPT key provided (__UUID__) does not belong to any existing IPT instance.
400	Primary contact email must be in valid format
400	Primary contact type must be either **administrative** or **technical**.

7.1.1.1.7 RegistryAPI: registerResource()

This service modifies the GBIF REGISTRY content with the existence of a new RESOURCE. The RESOURCE will be related to the provided organisation, which may or may not be the same organisation that houses the IPT. The username and password supplied for authentication must be correct for the organisation to which the RESOURCE will be associated.

Example request: http://gbrdsdev.gbif.org/registry/ipt/resource [HTTP POST] with

```
Authorisation: Basic <Base64Encoded
organisationIdentifier:organisationPassword>
```

Parameters:

Field	Mandat.?	Field value	Description
iptKey	YES	valid UUID & should be an existing one	UUID of the IPT instance that is publishing the resource .
organisationKey	YES	valid UUID & should be an existing one	UUID of the organisation to whom the resource belongs to.
name	NO	free text	Name of the new resource
description	NO	free text	Description of the new resource
language	NO	free text	Language of the new resource's metadata
homepageURL	NO	free text	Homepage for the new resource
logoURL	NO	free text	Logo URL for the new resource
primaryContactName	NO	free text	Name of the new resource's primary contact
primaryContactType	YES	**technical** or **administrative**	Type of the new resource's primary contact
primaryContactFirstName	NO	free text	First name of new resource's primary contact
primaryContactLastName	NO	free text	Last name of the new resource's primary contact
primaryContactAddress	NO	free text	Address of the new resource's primary contact
primaryContactDescription	NO	free text	Description of the new resource's primary contact
primaryContactEmail	YES	**valid e-mail format**	Email of the new resource's primary contact
primaryContactPhone	NO	free text	Phone number of the new resource's primary contact
serviceTypes	NO	free text	Multiple types allowed, but must match the order specified in the **serviceURLs** field. Types should be separated by pipe |. Example: RSS|DWC-ARCHIVE-OCCURRENCE|TAPIR
serviceURLs	NO	free text	Mutiple types allowed, but must match the order specified in the **serviceTypes** field. Example: http://www.example.org/obj.rss|http://www.example.org/dwc-archive.zip|http://www.example.org/tapir.php

Response:

201	Success

Errors:

401	Authorization header should be provided for this method
401	Authorization header is invalid
400	Organistion key must be supplied
400	The organisation key does not correspond to any existing organisation
400	The IPT key must be supplied
400	The IPT key does not correspond to any existing IPTs.
400	Primary contact email must be supplied
400	Primary contact type must be supplied
400	Primary contact email must be in valid format
400	Primary contact type must be either **administrative** or **technical**.

7.1.1.1.8 RegistryAPI: updateResource()

This service modifies the GBIF REGISTRY content by updating the metadata (title, description etc) for the provided RESOURCE. The username and password supplied for authentication must be correct for the associated organisation.

Example request: http://gbrdsdev.gbif.org/registry/ipt/resource/_UUID_

[HTTP POST]

where _UUID_ is an UUID for an existing resource.

with

```
Authorisation: Basic <Base64Encoded
organisationIdentifier:organisationPassword>
```

Parameters:

Field	Mandat.?	Field value	Description
organisationKey	-	valid UUID & should be an existing one	UUID of the new hosting institution/organisation. **Note:** This feature is disabled at the moment but might be used in the future if needed.
name	NO	free text	Name of the resource

description	NO	free text	Description of the resource
language	NO	free text	Language of the resource's metadata
homepageURL	NO	free text	Any homepage referring to the resource
logoURL	NO	free text	Logo URL for the resource
primaryContactName	NO	free text	Name of the resource's primary contact
primaryContactType	YES	**technical** or **administrative**	Type of the resource's primary contact
primaryContactFirstName	NO	free text	First name of the resource's primary contact
primaryContactLastName	NO	free text	Last name of the resource's primary contact
primaryContactAddress	NO	free text	Address of resource's primary contact
primaryContactDescription	NO	free text	Description of resource's primary contact
primaryContactEmail	YES	**valid e-mail format**	Email of the resource's primary contact
primaryContactPhone	NO	free text	Phone number of the resource's primary contact
serviceTypes	NO	free text	Multiple types allowed, but must match the order specified in the **serviceURLs** field. Types should be separated by pipe |. Example: RSS|DWC-ARCHIVE-OCCURRENCE|TAPIR * If a serviceType already exists, its values will be updated. If it does not exist, a new service will be created.
serviceURLs	NO	free text	Mutiple types allowed, but must match the order specified in the **serviceTypes** field. Example: http://www.example.org/obj.rss|http://www.example.org/dwc-archive.zip|http://www.example.org/tapir.php

Response:

200	Success

Errors:

401	Authorization header should be provided for this method
401	Authorization header is invalid
400	Resource key provided (__UUID__) does not belong to any existing resource.
400	Primary contact email must be in valid format

400	Primary contact type must be either **administrative** or **technical**.

7.1.1.1.9 RegistryAPI: deregister(Resource)

This service modifies the GBIF REGISTRY content by deleting the RESOURCE and all associated services. The username and password supplied for authentication must be correct for the associated organisation.

Example request: http://gbrdsdev.gbif.org/registry/ipt/resource/_UUID_

[HTTP DELETE]

where _UUID_ is an UUID for an existing resource.

with

```
Authorisation: Basic <Base64Encoded
organisationIdentifier:organisationPassword>
```

Response:

200	Success

Errors:

401	Authorization header should be provided for this method
401	Authorization header is invalid
400	Resource key provided (__UUID__) does not belong to any existing resource.

7.1.1.2 REGISTRY view of IPT

The following illustrates the GBIF REGISTRY view of an installation configured to PUBLISH 2 data RESOURCES through the IPT.

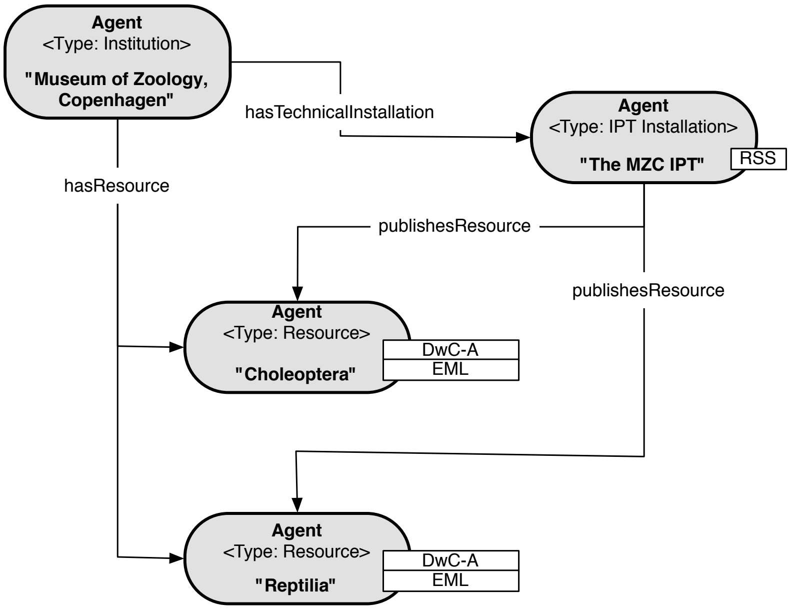

Figure 7: REGISTRY view of IPT with 2 resources

The following illustrates the REGISTRY view of an IPT in a multi-institution data-hosting scenario. In this scenario 2 resources are published through an IPT, but associated with a different ORGANISATION than the one hosting the IPT installation.

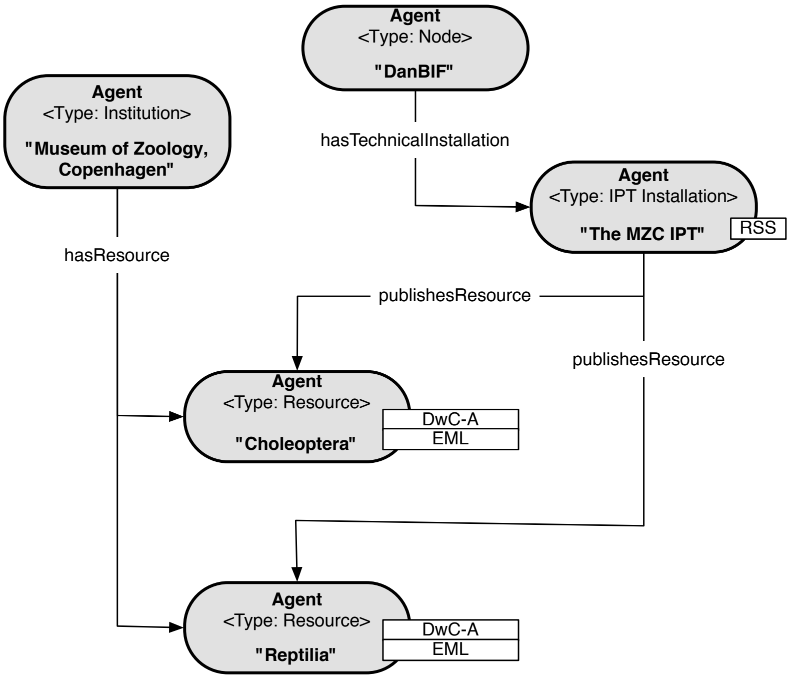

Figure 8: REGISTRY view of IPT in shared data hosting scenario

8 Physical architecture

[The physical view depicts the system from a system engineer's point-of-view. It is concerned with the topology of software components on the physical layer, as well as communication between these components]

8.1 Deployment scenarios

The tool is targeted at INSTITUTIONAL use, and requires that the IPT ADMINISTRATOR understand that they are deploying a tool designed to serve on the public Internet or a private intranet. This is similar to deploying any web server intended for widespread access. In the intended deployment the IPT should have a stable access point; e.g. a stable IP address, and potentially a DNS entry. The tool is not intended to be deployed by a bench scientist, or for those with changing IP addresses such as a laptop user, although it may be useful for certain scenarios. The reasoning behind this is that without a stable endpoint, others will not be able to discover and access the content being published. The endpoint can of course be periodically changed, but this is expected to be a rare event.

The IPT can PUBLISH data originating in either a SOURCE database or from uploading of a text file, such as a comma or tab delimited file. The following depicts the 2 intended deployment scenarios; connecting institutional databases to the Internet, and providing means for uploading content into an IPT. An IPT may be used in both scenarios simultaneously.

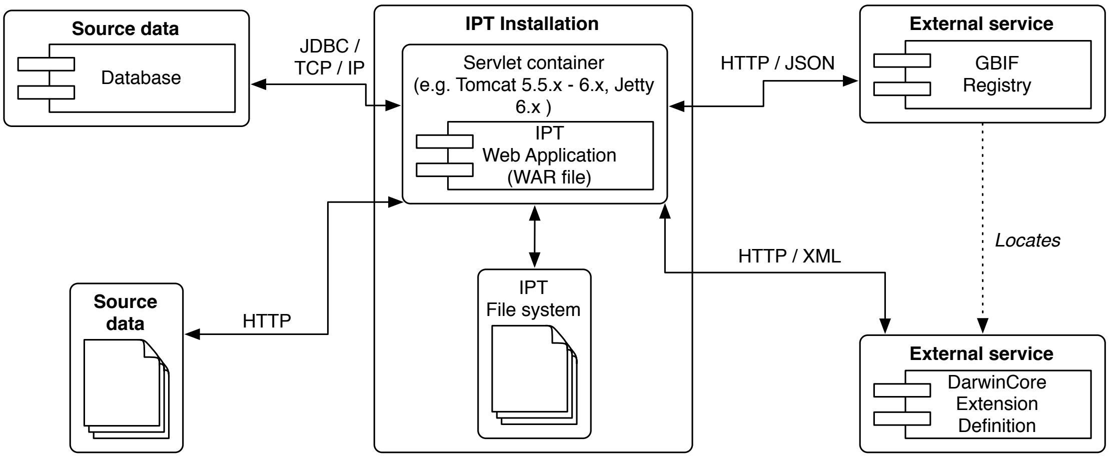

Figure 9: Deployment view of the IPT Application