esrin

Via Galileo Galilei
Casella Postale 64
00044 Frascati
Italy
T +39 06 9418 01
F +39 06 9418 0280
www.esa.int

TECHNICAL NOTE

Exploitation Platform - Functional Model

Reference	**ESA-EOPSDP-TN-17-050**
Issue/Revision	**1.0**
Date of Issue	**30/11/2017**
Status	**For Information Only**

European Space Agency
Agence spatiale européenne

APPROVAL

Table of contents:

1 INTRODUCTION

1.1 Purpose

This document provides a high level concept, use case analysis and a platform orientated functional model for an Exploitation Platform.

1.2 Document Structure

This document is divided into the following sections:

- This section, Section 1, is a brief introduction and lists common references and acronyms;
- Section 2 describes the high level context required to understand the remainder of the document;
- Section 3 describes a user orientated view of an Exploitation Platform, with reference to a set of use cases, and then a set of functions which together could be combined to realise those use cases;
- Section 4 describes one possible platform orientated view, identifying a set of services which encapsulate related functionality from Section 3.

1.3 Reference Documents

The following table lists the Reference Documents that are referred to in this document.

Exploitation Platform - Functional Model	Title	Reference	Issue
EP-USE	EO Exploitation Platforms – Use Cases	ESA-EOPSDP-TN-17-049	1.0

Table 1-1: Reference Documents

1.4 Acronyms and Abbreviations

Acronym	Meaning
EO	Earth Observation
EP	Exploitation Platform
GUI	Graphical User Interface
HPC	High Performance Computing
IaaS	Infrastructure as a Service
M2M	Machine to Machine (Interface)
MEP	Mission Exploitation Platform
PaaS	Platform as a Service

Acronym	Meaning
REP	Regional Exploitation Platform
TEP	Thematic Exploitation Platform
VM	Virtual Machine
VNC	Virtual Network Computing

Table 1-2: Acronyms and Abbreviations

2 CONTEXT

In recent years there has been an explosion in the volume of Earth Observation data acquired, processed and stored. This increase in quantity has occurred at the same time as the wider IT industry has moved from deploying solutions which make use of many small instances of private, single tenant infrastructures, to deploying solutions which make full use of large, multi-tenant cloud infrastructures. These cloud infrastructures are run by specialist providers, and through the sharing of the necessary fixed capital investment and operating costs across the whole user community, enable users to access quantities of processing resource and data that would otherwise be unobtainable.

The concept of an Exploitation Platform is to make use of this kind of infrastructure to enable EO data users to focus their time and expertise on the tools and techniques required to make use of EO data at scale, while specialist providers deal with the procurement and management of IT resources to store, process and transfer that data. Many different types of portal/platform have been created over the last few years; servicing user groups who share thematic interests[1], who wish to access data from the same mission[2], or who are interested in particular regional areas[3].

In the future enabling interoperability between such platforms will allow further sharing of resources. For example, a user accessing a Thematic Portal will be able to make use of data and processing services hosted by a separate Mission Exploitation Platform, and vice versa as shown in Figure 2-1. Such interoperability will also reduce the system of system's dependence on any single vendor and allow providers to specialise at different levels of the infrastructure in order to develop new features and target different audiences.

[1] a so-called Thematic Exploitation Platform
[2] a so-called Mission Exploitation Platform
[3] a so-called Regional Exploitation Platform

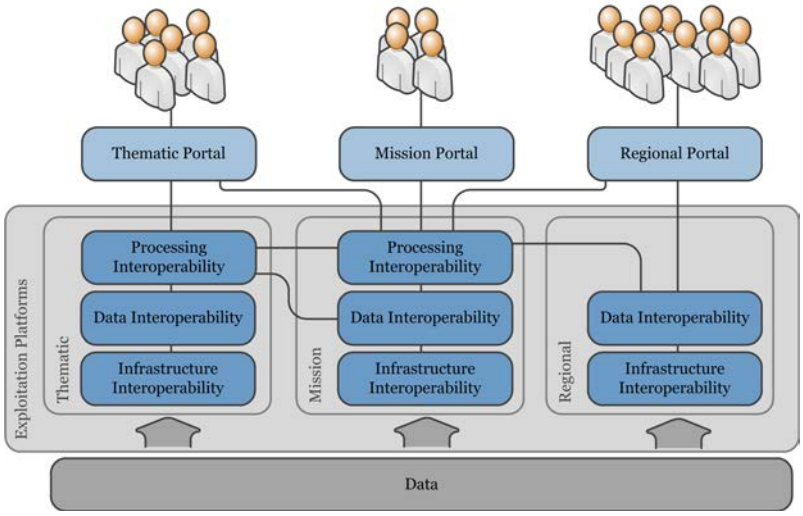

Figure 2-1: Portals and Platforms

In order to discuss this interoperability, and encapsulate the current state of the art, Figure 2-2 splits a deployment of an Exploitation Platform into a set of conceptual tiers.

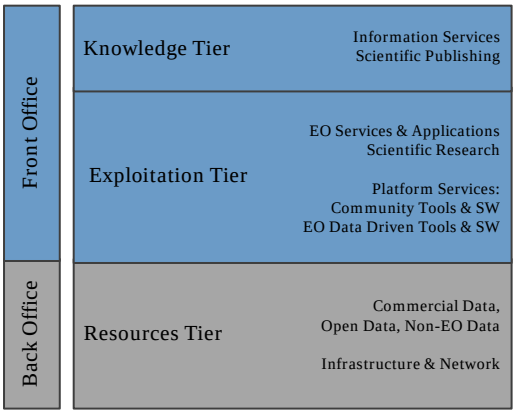

Figure 2-2: Conceptual Tiers

These tiers are further divided into a “back office” tier and a pair of “front office” tiers. The back office tier can be considered as an enabling element which provides common functionality and the basic large data storage and processing infrastructure, while the front office tiers are a stimulating element focused on helping specific communities of users to make use of EO data within their domain:

- [The] Resource Tier is composed of a set of Infrastructure & Network services providing IT data storage and processing resources, along with some level of resource management; and Data services which provide the Front Office tiers with access to data.

  The Resource Tier is responsible for connecting the set of platforms together, and linking platforms with data providers. The tier will host datasets relevant to the

platforms deployed on the infrastructure. Data is made available across this tier from data providers to the various infrastructure instances for storage. Where the front office requires archived data, the tier is responsible for facilitating the request.

The amount of customisation required to a standard IT cloud infrastructure depends on whether the underlying resource model is bare-metal, virtual machine, container based; whether standard big data technologies (grids, clusters) are mapped here or to the Platform/Data tiers; and the level of integration between this and the Knowledge/Exploitation tiers for the management of data.

It should be recognised that customisation here trades interoperability between infrastructures for additional capability/performance.

- [The] Exploitation Tier provides EO specific exploitation platform features, such as catalogue, tooling, data access, processing management, visualisation, etc. and additional features that support the building of communities around a particular theme, mission or region. An individual platform that targets a specific community may deploy algorithms and tools here to support the work of that community, for example to support Scientific Research.

  This tier also logically exposes EO Services and Applications hosted on the platform to other users of the platform, and externally to the Knowledge Tier. For services which are not fully automated, this includes any operator/analyst activity required to provide the service. Similarly, a community platform may provide information/support/outreach features to attract new users within this tier.

- [The] Knowledge Tier presents the EO services deployed on one or more platforms to a wide audience. This audience may only be interested in information derived from EO data which can be used to further their scientific research, or to support their public/private sector decision making.

Some common features (including user management, resource management, billing, monitoring) are relevant across multiple tiers.

Interoperability should exist within a tier to allow applications/services to be deployed on different platforms, but needs to at least exist at:

- The interfaces between the Infrastructure/Network Services and Data Services within the Resource Tier, in order to allow the platform instances to be populated with data from many different providers, and the data providers to provide data to many different platforms;

- The interface between the Resource Tier and Exploitation Tier, in order to allow the Exploitation Tier to be instantiated on multiple different infrastructures with different scaling and to avoid lock-in;

- The interface between the Platform specific parts of the Exploitation Tier and the EO Services, in order to allow a Community Platform to make effective use of the services/features exposed by multiple infrastructures;

- The Knowledge Tier should enable final users to discover services, applications and EO-derived information of interest to their use case. This therefore requires the providers of Exploitation Tier functionality to expose their services in a machine readable format.

The remainder of this document develops this conceptual model in order to identify which services and features the platforms need to provide, and hence provide a basis from which to start identifying these interfaces. Such services and interfaces could be deployed on a single Exploitation Platform, or across multiple platforms. Any individual platform may choose to implement only the relevant subset of the described services that are required by its community.

3 USER ORIENTATED VIEW

This section develops a user orientated view of the Exploitation Platform functionality by discussing an initial set of use cases, the interactions involved within those use cases, and then a set of logical functions which can be used to realise those interactions.

The set of use cases used to develop this description are taken from [EP-USE], which identifies a set of use cases covering the deployment and exploitation of user specified processing services across a distributed infrastructure. These should be understood as examples of the kinds of functionality required, and focus primarily on the Front Office tiers. Many other use cases may be relevant for particular communities and/or are required for a complete description of an Exploitation Platform.

3.1 Use Cases

The [EP-USE] use cases are described in terms of Consumers, Expert Users, External Applications and Platforms. In this subsection, we identify exemplar users for each of these categories then use these users to identify the common interactions in the use cases.

3.1.1 Exemplar Users

Name	User Category
Alice	An Expert User, developing and integrating services/products within the platform. Two broad groups of Expert User are identified. A scientific expert user may have limited IT knowledge, and be interested in a platform which minimises the amount of effort required to integrate their algorithm for use. A commercial expert user needs to create a stable/reliable service or product, and therefore may require lower level access to the platform for optimisation of their processing flows.
Bob	A consumer of services and data from the Exploitation Platform. A consumer may be scientific, commercial or a member of the general public; they may have a lot of knowledge of the Platform's domain or otherwise.
Charles	A consumer of services and data from the Exploitation Platform, using an external application. Exposure of Exploitation Platform interfaces to such an external application via an API enables specific applications to be written for different communities, or services/products to be shared across different Exploitation Platforms.
Platform T	An Exploitation Platform. In this set of use cases this platform supports functionality from all of the tiers identified in Section 2 above.
Platform U	An Exploitation Platform, hosting a commercial dataset. This platform provides both processing and dataset storage capabilities.

Table 3-1: Exemplar Users for the Use Cases

3.1.2 Use Case Interactions

A review of the use cases show that we can model the interactions of these users as shown in Figure 3-1:

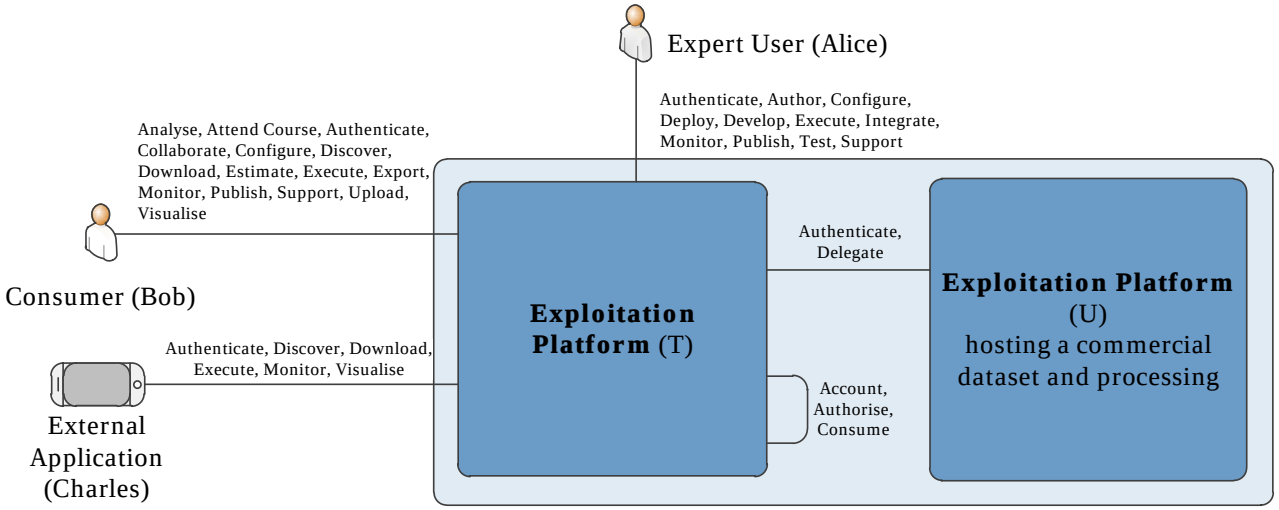

Figure 3-1: User Interactions with Platforms

Table 3-2 summarises the [EP-USE] use cases in terms of these interactions. Note that authentication, authorisation, internal resource consumption and usage accounting can be considered interactions that are involved in every use case.

#	Use Case Description	Interactions
UC2.1	*Consumer discovers and visualises value-added products* **Bob** accesses **Platform T** and uses the discovery interfaces to search for data and/or value added products. **Platform T** presents information about the data including access costs, and then enables **Bob** to visualise them. Depending on the specific constraints applied to the data, **Bob** may be able to authenticate and download selected data to their own machine.	Consumer: Discover, Download, Visualise
UC2.2	*Consumer uploads data into its own workspace* **Bob** uploads their own data into a workspace provided by **Platform T**. **Bob** is then able to visualise their data using the **Platform's** capabilities. Optionally, **Bob** is then able to publish their data to the **Platform's** catalogue so that other users can find and use it.	Consumer: Publish, Upload, Visualise

#	Use Case Description	Interactions
UC2.3	*Consumer discovers and executes processing services* **Bob** accesses **Platform T** and uses the discovery interfaces to search for appropriate processing services. **Bob** reviews the information presented for the processing service, and identifies input data from both within and outside the **Platform** which can be used with it. Prior to triggering execution, **Bob** is able to get an estimate of the execution costs and timescale. During execution, **Bob** is able to monitor their processing task, and at the end of the execution can visualise and potentially download the results.	Consumer: Discover, Download, Estimate, Execute, Monitor, Visualise
UC2.4	*Consumer discovers and executes interactive applications* **Bob** accesses **Platform T** and uses the discovery interfaces to search for appropriate interactive processing services. When executing, the **Platform** presents **Bob** with the application's graphical user interface via the web browser, and makes EO data and products available within the application's environment. **Bob** should be able to export the results from the application outside of the **Platform**, visualise and/or download the results.	Consumer: Discover, Download , Execute, Export, Visualise
UC2.5	*Consumer analyses value-added products* **Bob** uses tools provided by **Platform T** to analyse data, processing results or value-added products using an interactive online analysis tool.	Consumer: Analyse
UC2.6	*Consumer executes bulk processing* **Bob** configures a batch processing service provided by **Platform T** to process a large amount of data (e.g. daily acquisitions, all acquisitions over a given AOI).	Consumer: Configure, Execute, Monitor

#	Use Case Description	Interactions
UC2.7	*Consumer executes systematic processing* **Bob** configures a systematic processing service on **Platform T** to automatically process data at a given time (e.g. the first day of every week), on new data as it arrives, or following an external triggering event (e.g. an earthquake alert).	Consumer: Configure, Execute, Monitor
UC2.8	*Consumer performs open science* **Bob** shares the results of their processing on **Platform T,** including creation of a DOI to allow referencing in a scientific paper.	Consumer: Collaborate
UC2.9	*Consumer accesses Exploitation Platform with an External Application* **Charles** makes use of an External Application to access the services exposed by **Platform T**.	External Application: Discover, Download, Execute, Monitor, Visualise
UC2.10	*Expert User builds new processing services* **Alice** packages and deploys their own software onto **Platform T**, integrates it with services exposed by the **Platform** and then tests it in the platform environment. When happy, **Alice** publishes the software as a new processing service for herself or others to use.	Expert: Deploy, Execute, Integrate, Publish, Test
UC2.11	*Expert User builds new processing service chains* **Alice** uses functionality provided by **Platform T** to chain several processing services (potentially offered by different platforms) in parallel or sequentially. She defines the relations, inputs, outputs and parameters for each step, and then publishes the software as a new processing service for herself or others to use. During execution, **Platform T** is responsible for handling the chained processing services, delegating processing where necessary to other platforms (e.g. **Platform U**) and hence consuming resources on those platforms.	Expert: Execute, Integrate, Publish, Test Platform: Delegate

#	Use Case Description	Interactions
UC2.12	*Expert User builds new interactive applications* **Alice** packages and deploys their own interactive application onto **Platform T**, integrates it with services exposed by the **Platform** and then tests it in the platform environment. When happy, **Alice** publishes the software as a new interactive application for herself or others to use.	Expert: Deploy, Execute, Integrate, Publish
UC2.13	*Expert User builds new value-added products* **Alice** configures and executes a bulk (Use Case 2.5) or systematic (Use Case 2.6) processing activity, and chooses to publish the results automatically as a new value-added product collection on **Platform T**.	Expert: Configure, Execute, Monitor, Publish
UC2.14	*Expert User develops and executes code via a web development environment* **Alice** accesses an online interactive development environment on **Platform T** which enables her to upload her code and execute it as a whole, or step-by-step. **Platform T** provides a set of general libraries to facilitate the exploitation of EO data, and may also provide specific libraries which support different types of application (e.g. machine learning).	Expert: Develop, Execute
UC2.15	*Expert User performs training* **Alice** authors tutorials, documentation, lessons, webinars etc. to describe the use of her processing services/interactive applications and publishes them as online courses on **Platform T**; **Alice** may also answer questions raised by consumers on her software/value-added product. **Bob**, interested in making use of **Alice's** software, uses the functionality of **Platform T** to discover such training material, and attends the online courses. Bob may also use collaborative tools to raise questions to **Alice** on the use of her software/value-added product.	Expert: Author, Support Consumer: Attend Course, Discover, Support

Table 3-2: Use Cases Mapped To Interactions

3.2 Functions

The interactions identified can be mapped to set of logical functions. We start by identifying four high level groups of functionality in Figure 3-2, and then decompose each of these groups in turn, relating them to the tiers introduced in Section 2 above in order to identify how the functionality required to support the use cases maps to the conceptual model.

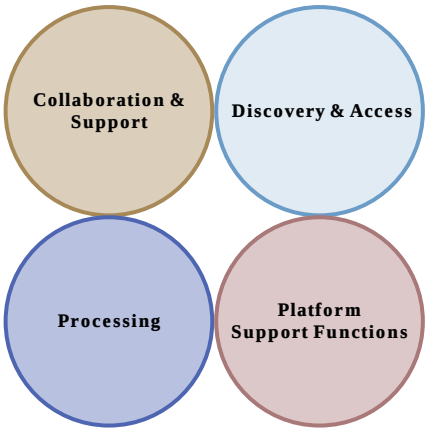

Figure 3-2: Functional Groups

3.2.1 Discovery & Access

A User will be able to discover services and data relevant to their domain of interest, and interact with the platform to access processing services/interactive applications and visualise their results. Certain of these features (e.g. data access, discovery) may be subject to access control constraints. Other platforms and external applications should be able to make use of the discovery and service/data access functions in order to support use cases such as UC2.9 and UC2.11.

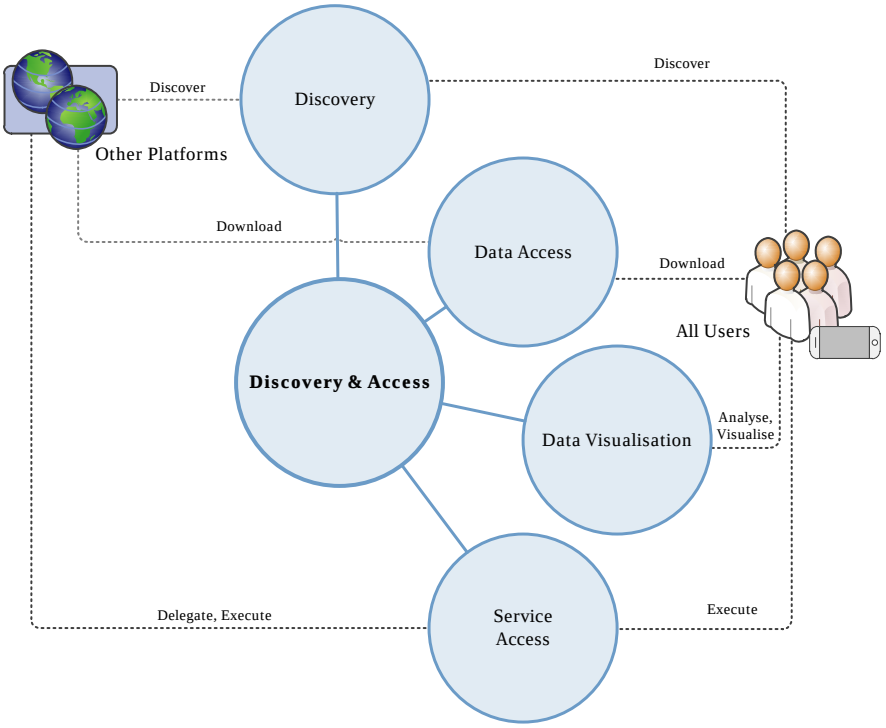

Figure 3-3: Discovery & Access Functions

These functions are primarily part of the Exploitation Tier, but aspects of them (e.g. discovery by final users of capabilities within the Exploitation Platform from within the Knowledge Tier, or discovery of data held remotely via interaction encapsulated within the Resource Tier) are part of other tiers. Visualisation may be used to help the user understand/browse the results of the services they consume. These functions are required to support all of the use cases identified in the previous section.

Within the conceptual model, we also consider Discovery and Data Access functions as supporting interaction between the Exploitation and Resource Tiers, i.e. between the exploitation platforms and the contributing data providers.

3.2.2 Collaboration & Support

A User will be able to find collaboration and support functionalities which aid them in making best use of the platform's services, and enable them to share their outputs, algorithms, results with other like-minded users. Some of these functions (e.g. wikis, forums, collaborative workspaces) may be provided as an inherent feature of the platform; other functions (e.g. training/outreach/support) are a supporting service built around the platform and intended to drive engagement with the platform within a particular community.

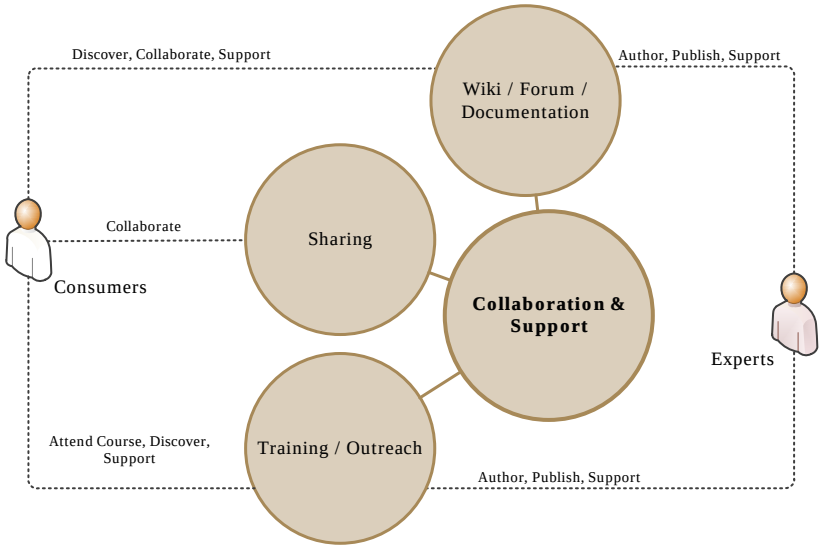

Figure 3-4: Collaboration & Support Functions

These functions are required to support use cases UC2.8 and UC2.15.

3.2.3 Processing

A Consumer will be able to access both interactive and non-interactive processing services via the platform, and to integrate their own processing services for use by others. Integration/development of new processing services will be limited to Expert Users. Systematic and Batch processing services may be accessible from other platforms in order to support the chaining use case UC2.11.

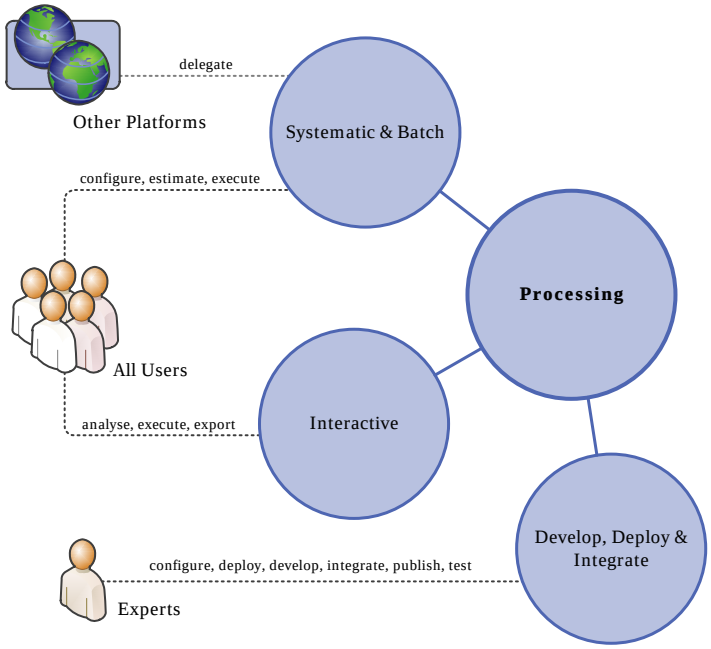

Figure 3-5: Processing Functions

These functions form part of the Exploitation Tier, and the set are required to implement all of the use cases identified in Section 3.1.

3.2.4 Platform Support Functions

These enable user-specific functionality and resource accounting. A user will be able to authenticate in order to gain access to platform functionality that requires significant compute/data resource, understand the amount of resource used in their activity, and the remaining quota of limited resources (e.g. data, credit) available to them. Expert Users will be able to publish their own services and data products, and configure the platform to control/account for access to these resources.

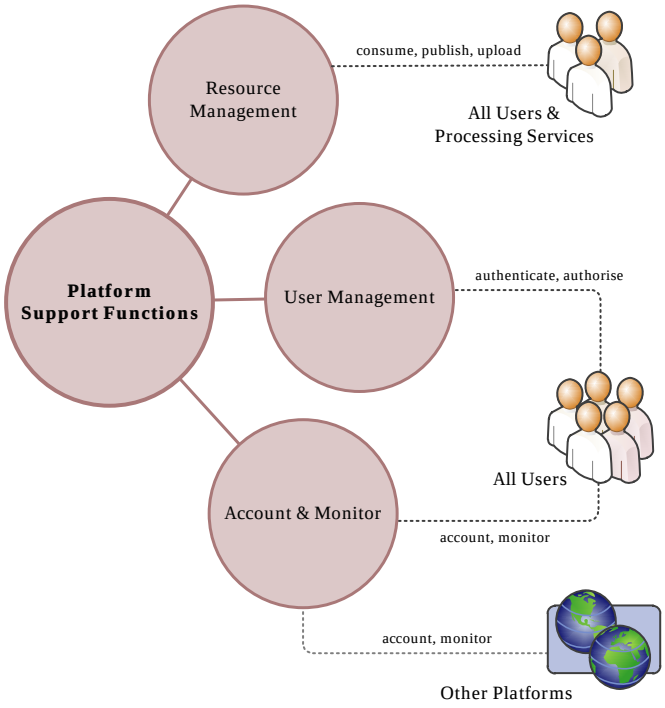

Figure 3-6: Platform Support Functions

These functions are split across the Exploitation and Resource Tiers, though in practice User Management and Accounting may need to be integrated into all Tiers.

These functions are required to implement all of the use cases identified in Section 3.1.

4 PLATFORM ORIENTATED VIEW

This section develops one possible platform orientated view in order to identify a set of services which encapsulate related functionality, and which together could support the necessary user functionality identified in Section 3 above.

4.1 Model

Figure 4-1 decomposes an Exploitation Platform into a “stack”.

At the bottom of the stack, shown in grey, is the underlying IT infrastructure, represented as a set of execution units (bare metal machines, virtual machines, containers), connectivity and storage. The software which supports the Exploitation Platform's services, along with any algorithms/services/software created and deployed by users, run across these execution units.
Above the infrastructure are groups of services, which each further abstract the underlying infrastructure away from the IT domain to the EO Data and Services domains.

The Cloud Layer is thus the first layer of abstraction, enabling the Exploitation Platform to manage execution units and data hosted within the infrastructure. Above this there is a set of Support Services which encapsulate resource management and business logic, and then a set of Exploitation Platform services which enable the common Platform functionality: the discovery and visualisation of resources, wrapping and management of batch processing, the development of new applications and services, and the collaboration and sharing features required to support a community. Finally, the presentation platform services have dependencies on all of the lower layers, and make the services and data of the platform accessible to the user base.

As shown on the figure, multiple Exploitation Platforms may coexist on the same underlying IT infrastructure, and any given platform may be deployed across more than one type of IT infrastructure. Exploitation Platforms may share services with each other, for example, multiple exploitation platforms may share the same User Management Service implementation.

The following sections further elaborate on each of these services and map them to other similar functional models.

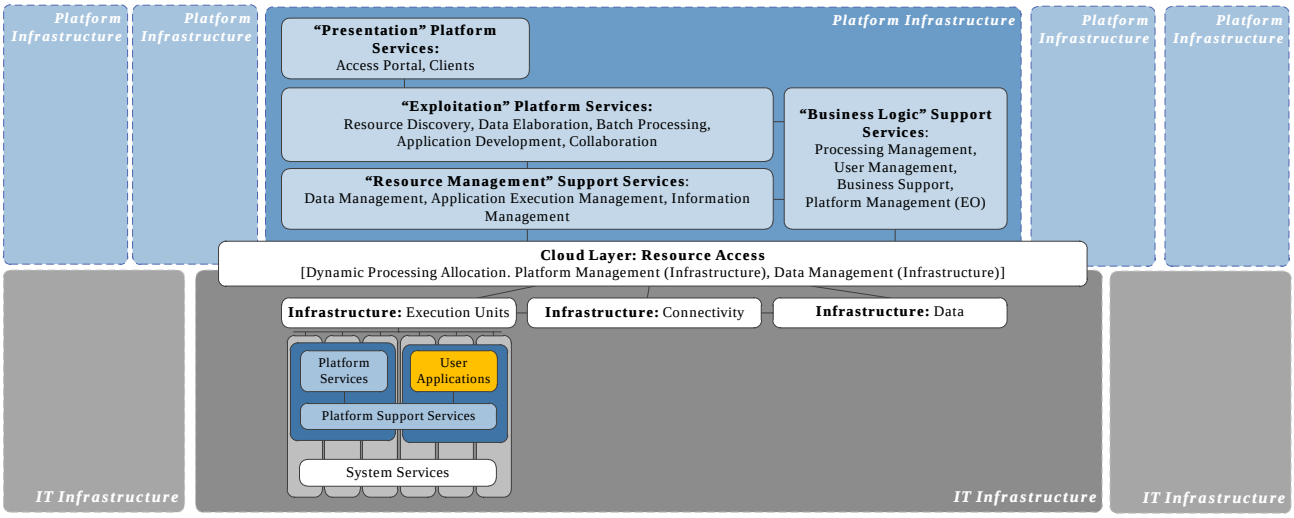

Figure 4-1: Platform Services and Infrastructure

4.2 Platform Services

4.2.1 Presentation

4.2.1.1 Access Portal

The Access Portal Service provides a single point of access for Users through which they can access the services published on, and/or accessible via, the Exploitation Platform. The structure and features of the access portal will vary depending on the targeted user community, but will usually expose discovery features. Within this decomposition, the Web GUI components that support these discovery features are described under the Resource Discovery Service in Section 4.2.2.4 below.

Specific Features:

- **Public Front End (Web):** Tailoring of the front end for the general public, promoting the services/applications accessible via the platform.
- **Authenticated User Front End (Web):** Tailoring of the front end for authenticated users, providing single point of access to the set of services/applications accessible by the user.
- **M2M Front End:** Front end for machine to machine users (e.g. delegated access via other platforms), providing single point of access to the set of services/applications accessible by the user.
- **Platform Presentation Framework:** Standard presentation elements for use on the platform.

Tiers (Section 2)	Knowledge; Exploitation.
Use Case Interactions (Section 3.1)	Discover; Consume.
Functions (Section 3.2)	Discovery & Access: Discovery [Information & Service], Discovery [Processing & Data], Service Access.

Table 4-1: Access Portal Traces

4.2.2 Exploitation

4.2.2.1 Application Development Service

The Application Development Service supports Expert Users who wish to develop, integrate, package and deploy their applications onto the platform.

Specific Features:

- **Application Development (opt.):** Enables a user to develop an Exploitation Platform hosted application based on their algorithm. The feature provides a development environment (e.g. code editor) to allow user to code their application directly on the Exploitation Platform. The platform should provide a framework on which application can be developed in order to ease coding and integration. It interacts with the Application Integration feature for application testing. A user has to be authorized to develop an application. This feature can be used to edit existing applications.

- **Workflow Development (opt.):** Allows a user to develop a workflow by chaining several applications together, across one or many Exploitation Platforms. Users have to be authorised to access an application in order to use it within a workflow. This feature can also be used to edit existing workflows. Allows a user to test the new/modified workflow and prepare it for being managed by the Application Management service.

- **Application Integration:** Allows a user to integrate their software application into an Exploitation Platform. It provides users with a sandbox within which they can integrate and test the application on the platform, and deploy it (via packaging) for being managed by the Application Management service. The application is developed on user premises and this feature is only used to perform the final integration and testing with the Exploitation Platform environment. This feature offers a web based interface to allow a user to:
    - Upload application executable;
    - Select execution environment;
    - Select test data;
    - Launch the test session;
    - Publish the application to the Application Management service.

Tiers (Section 2)	Exploitation.
Use Case Interactions (Section 3.1)	Configure; Deploy; Develop; Integrate; Test
Functions (Section 3.2)	Processing: Develop, Deploy & Integrate

Table 4-2: Application Development Service Traces

4.2.2.2 Collaboration Service

The Collaboration Service supports active collaboration between all types of user through the provision of Collaborative Workspaces within which a set of users can work collaboratively on the same data, algorithms and results and algorithms.

Specific Features:

- **Collaborative Workspace:** A shared environment in which data can be analysed and manipulated. The results of such analysis are available to all the users sharing the workspace.

- **Collaborative Workspace Management:** Allows authorised users and operators to manage Collaborative Workspaces, including creation, access control, deletion. Publishes metadata about the set of available collaborative workspaces to support discovery functions.

Tiers (Section 2)	Exploitation.
Use Case Interactions (Section 3.1)	Collaborate, Upload
Functions (Section 3.2)	Collaboration & Support: Sharing

Table 4-3: Collaboration Service Traces

4.2.2.3 Data Elaboration Service

The Data Elaboration Service provides common functionality used to visualise, combine and transform data which can be used by Expert Users to support the creation and publication of services (e.g. platform functionality which can be integrated into workflows, or display the results from a workflow), or by any user to interactively explore the data hosted on the platform.

Specific Features:

- **Visual Data Elaboration (opt.):** Allows users to display data content on a map and elaborate it using basic tools which do not require demanding processing. Can be used as the front end to an application.

- **Quicklook Generation and Provision (opt.):** This provides the ability to generate previews from the ingested data for provision as "browses" by the resource discovery service's catalogue.

- **EO Data Analysis Tools:** Provides common languages, libraries and tools which can be used to retrieve, manipulate and analyse EO data within a workflow or application.

- **Interactive Notebook (opt.):** Allows users to use interactive notebooks (e.g. Jupyter) to work directly with Exploitation Platform data and record the result in a document. This feature should be supported with the relevant libraries and examples required to enable a user to take advantage of the platform's functionality.

Tiers (Section 2)	Exploitation.
Use Case Interactions (Section 3.1)	Analyse; Visualise
Functions (Section 3.2)	Discovery & Access: Data Visualisation, Service Access; Processing: Interactive

Table 4-4: Data Elaboration Service Traces

4.2.2.4 Resource Discovery Service

The Resource Discovery Service provides a set of features which allow a User or Services/Applications to search for the following items within the Exploitation Platform:

- Data;
- Documents;
- Applications;
- Collaborative Workspaces.

The discovery service should provide a GUI interface and it should be possible to filter the results by access control constraints - i.e. whether the current user has access to the resource. The discovery services should also provide a M2M interface to enable integration between platforms and to allow community/marketplace portals to provide their own tailored GUI for their specific user domains.

Specific Features:

- **Data Catalogue:** Allow users to search for geo-temporal data, organised in datasets, matching a set of criteria, with results displayed projected onto a map where appropriate.

- **Documentation Library:** Allows users to search for documents, forum threads, wiki entries and issue items within the Exploitation Platform.

- **Application Marketplace:** Present the applications accessible via the Exploitation Platform, along with metadata/examples which identify what the applications can do for a user. These may be supplemented by user reviews, etc.

- **Collaborative Workspace Showcase:** Allows users to see which collaborative workspaces are available within the Exploitation Platform, and filter them according to a set of user specified criteria.

- **Integrated Search (opt.):** Allow users to search the documentation items, applications and collaborative workspaces from one location.

Tiers (Section 2)	Exploitation.
Use Case Interactions (Section 3.1)	Discover
Functions (Section 3.2)	Discovery & Access: Discovery [Information & Service], Discovery [Processing & Data]

Table 4-5: Resource Discovery Service Traces

4.3 Support Services

4.3.1 Resource Management

4.3.1.1 Application Management Service

The Application Management Service supports the management of, and access to, the applications hosted by the Exploitation Platform. It owns all aspects of User deployed "executable software" (including persistent storage of this software). It does not own any processing resources.
An application may be a single executable, or a chained service that executes a number of separate operations (perhaps calling other application services) in sequence. Applications are packaged in a standard packaging format which groups together the executable software and all of the information needed to deploy and run it. Within the features below, these “packages” are referred to as “application items”.

Specific Features:

- **Application Item Management:** Manages the entire life cycle (Create, Publish, Access, Update, Delete) of an application item within the platform. It does not manage the execution lifecycle of a specific instance of an application.

  An application item can be entirely managed by the Exploitation Platform such that the executable is instantiated on the platform, or can be referenced by the Exploitation Platform. In the referenced case the application entry describes the application and how it can be accessed, but the instantiation and execution is managed by another platform, allowing for cross platform integration.

  This feature provides a web based M2M interface to perform the following operations:
  - Create a new application item;
  - Publish an application item to other Platform users (all or list);
  - Download the application executable and/or all additional information required to instantiate it;
  - Update an application item;
  - Delete the application item.

This feature may interact with the Cost Specification and Estimation features to enable a user to specify a cost model / determine the cost of a processed use of the application prior to starting it.

Tiers (Section 2)	Exploitation.
Use Case Interactions (Section 3.1)	Consume; Delegate; Publish.
Functions (Section 3.2)	Platform Support Functions: Resource Management Processing: Develop, Deploy & Integrate; Systematic & Batch; Interactive

Table 4-6: Application Management Service Traces

4.3.1.2 Information Management Service

The Information Management Service supports the management of, and access to, the documentation and support tools hosted by the Exploitation Platform.

Specific Features:

- **Document Repository:** Enables the management / update / publication of documents and software tools related to the Exploitation Platform's functions.
- **Issue Tracking Support:** Provides an issue tracking system to allow users to track/share issues relating to the platform, and optionally, the services/datasets deployed on the platform.
- **Forum Support (opt.):** Provides a forum function for use by a community.
- **Q&A Support (opt):** Provides a question and answer forum function for use by the community.
- **Wiki Support (opt.):** Provides a Wiki which can be used by a community to share useful information.
- **Training Course Provision (opt.):** Provides an online repository of training courses which can be populated by Expert Users, and accessed by Consumers.

Tiers (Section 2)	Exploitation.
Use Case Interactions (Section 3.1)	Attend Course; Author; Collaborate; Discover; Publish; Support.

Functions (Section 3.2)	Collaboration & Support: Wiki / Forum / Documentation; Training/Outreach

Table 4-7: Information Management Service Traces

4.3.1.3 Data Management Service

The Data Management Services manages the data on the platform throughout the lifecycle - from ingestion to data access. Some parts of this service are infrastructure specific (e.g. managing the data on the infrastructure), whereas other aspects face the remainder of the platform (Inventory, Data Access) and yet other aspects interact with other parts.

Specific Features:

- **Data Ingestion:** This feature supports ingestion of data into the platform. This may include:
  - Extraction of metadata;
  - Conversion/packaging to make the data available within the infrastructure in a consistent way;
  - Interaction with the storage infrastructure;
  - Addition to the relevant collection as part of metadata publication.
  - Interaction with the preview / notification features where supported and required.

- **Data Collection Management:** This supports the management of data items within collections. Each data item is associated with one or more collections, and this requires the configuration of rules which govern:
  - Which data items are assigned to each collection;
  - How data items within a collection can be used, and by who: the collection access rights;
  - How data items are retained within the Platform (e.g. where, and for how long);
  - Ownership and licensing of the data items;
  - Where the data item metadata is published for use by the resource discovery service, and its contents;
  - Whether the data collection is local or remote, and if remote, how it can be accessed.

  This information is then used by the rest of the data management service.

- **Data Storage:** This is the feature responsible for the management of data physically within the infrastructure. It is written to by data ingestion, managed in bulk by data collection management and accessed via the data access feature. It is not necessarily specialised for the management of EO data. It is able to resolve a high level request containing an identifier to a given data object/item.

- **Local Data Access:** This provides one or more mechanisms to access the data held locally within the Exploitation Platform, and supports the necessary interactions

with the User Management function to provide accounting/access control. Nominally, this feature provides access to data only to services/applications located within the Exploitation Platform.

- **Remote Data Access (opt.):** This provides a mechanism for a local process within the Exploitation Platform to access the data held at a remote location (e.g. another Exploitation Platform, or from one of the data providers), and supports the necessary interactions with the User Management function to provide accounting/access control.

- **Notification (opt.):** This feature allows an application activity to register interest in a dataset such that it is notified of updates, the so-called Publish/Subscribe model. This can be used by the Application Execution Management Service to automatically trigger processing activities on ingestion of data.

Tiers (Section 2)	Knowledge; Exploitation; Resource.
Use Case Interactions (Section 3.1)	Consume; Download; Publish; Upload
Functions (Section 3.2)	Discovery & Access: Data Access Platform Support Functions: Resource Management

Table 4-8: Data Management Service Traces

4.3.2 Business Logic

4.3.2.1 Business Support Service

The Business Support Services provides usage accounting (summarising), and optionally estimation, cost specification and billing.

Specific Features:

- **Usage Accounting:** Summarises usage per resource / resource owner[4]. This is a complementary view to the user focused "Accounting" feature which captures use of resources by an individual user.

[4] Where the resource owner is the user responsible for publishing it.

- **Cost Specification (opt.):** Support for the creation of service cost estimates via the specification of cost/licensing models for data, applications or processing resources as an input to the cost estimation feature.

- **Cost Estimation (opt.):** Support for the calculation of an estimate of service cost based on resource use (processing and storage resources, software, data) and the cost parameters specified using the cost specification feature.

- **Billing (opt.):** Support for customer billing for services / resources consumed, reconciling use of resources by individual users with use of resources by the platform as a whole.

Tiers (Section 2)	Knowledge; Exploitation; Resource.
Use Case Interactions (Section 3.1)	Account; Estimate Optionally extended with “Billing”.
Functions (Section 3.2)	Platform Support Functions: Account & Monitor Supports “Estimation” for Systematic & Batch Processing.

Table 4-9: Business Support Service Traces

4.3.2.2 Platform Management Service

The Platform Management Service provides platform management control and monitoring features to Operators.

Specific Features:

- **Operator Interface:** Interface provided to the platform Operator to support monitoring/platform management activities.

- **Metrics Provision (opt.):** Provides information to the Exploitation Platform Operator (or other Exploitation Platform services), information on the Resource Discovery Service's activities (e.g. number of searches in a given time window, most popular search criteria etc.). Additional usage / metrics may be identified for other features of the platform.

- **Internal Authentication:** Interface provided to support the management of shared secrets across the platform where required to support authentication with infrastructure and internally within the platform.

Tiers (Section 2)	Exploitation; Resources

Use Case Interactions (Section 3.1)	Monitor; Authenticate
Functions (Section 3.2)	Platform Support Functions: User Management; Account & Monitor

Table 4-10: Platform Management Service Traces

4.3.2.3 Application Execution Management Service

The Application Execution Management Service supports the execution lifecycle of applications hosted via the Exploitation Platform.

These applications can be categorised as:

- Systematic or Batch non-interactive applications;
- Systematic or Batch non-interactive applications, chained in a workflow;
- Interactive applications, which require a GUI to be presented to a user. Such interactive applications may require provision of a VNC session to a virtual desktop, provision of a VNC session to an individual application, or even a full web-based notebook-style interface to the platform's functionality with additional libraries provided by an Expert User.

Specific Features:

- **Job Execution Management:** Manages the execution lifecycle (creation, scheduling, run, termination) of a batch application or an individual Workflow task. Interacts with the Dynamic Processing Allocation service to negotiate the resources on which the job will be deployed and run. Provides a web based interface to:
  - Create a job, including configuration where necessary;
  - Deploy a job on the execution environment;
  - Check a job status;
  - Start and Terminate a job;
  - Split a job among multiple nodes and multiple processing steps;
  - Pause and Resume a job (optional);
  - Submit input data in the job workspace;
  - Retrieve output data from the job workspace.

  A job can be created on user request, timer or another service, e.g. workflow task or notification (to support data driven processing). In case of time triggered applications (e.g. an application which is run every 12 hours to process data which has become available since the last execution) this feature ensures that a new execution for such application is scheduled with the appropriate frequency.

- **Workflow Execution Management (opt.):** This manages the execution of a workflow. It interacts with the Job Execution Management feature for the execution of each individual workflow task. It provides a web based interface to:
  - Instantiate a workflow;
  - Check a workflow status;

- Start and Terminate a workflow;
- Split the workflow among multiple processing centres and multiple workflow services invocation;
- Pause, Resume a workflow (optional);
- Submit input data in the workflow workspace;
- Retrieve output data from the workflow workspace.

- **Interactive Application Execution Management (opt.):** Manages the job execution lifecycle of an interactive application. It interacts with the Dynamic Processing Allocation service to negotiate the resources on which the job will be deployed and run. It provides a web based interface to:
  - Create a job;
  - Deploy a job on the execution environment;
  - Check a job status;
  - Start and Terminate a job;
  - Pause and Resume a job (optional);
  - Submit input data in the job workspace;
  - Retrieve output data from the job workspace;
  - Provide GUI access to the application.

- **Remote Deployment / Delegation:** Manages the interaction with a remote processing service, including any necessary transfer of the application to be executed and authentication if relevant.

Tiers (Section 2)	Exploitation.
Use Case Interactions (Section 3.1)	Configure; Consume; Delegate; Deploy; Execute
Functions (Section 3.2)	Discovery & Access: Service Access. Processing: Systematic & Batch; Interactive

Table 4-11: Application Execution Management Service Traces

4.3.2.4 User Management Service

The User Management Services supports authentication (registration, login, profile management), authorisation (policy definition and enforcement) and accounting. This functionality applies to all users.

Specific Features:

- **Authentication:** Allows a user to authenticate with the platform. Allows the platform to authenticate users belonging to different domains (external to the platform).

- **Delegated Authentication:** Allows the platform to authenticate with an external system on behalf of the user, or an external application to authenticate with the platform on behalf of the user via a M2M interface.

- **Policy Enforcement (opt.):** Supports access control to services through the definition of policies which can be evaluated against user attributes. These policies are defined at the relevant management functions for the given resources.

- **Profile Management:** Allows a user to manage their user information, including registration.

- **Accounting:** Provides accounting for resource access per user.

Tiers (Section 2)	All
Use Case Interactions (Section 3.1)	Authenticate; Authorise; Account.
Functions (Section 3.2)	Platform Support Functions: User Management

Table 4-12: User Management Service Traces

4.3.3 Infrastructure Resource Access

4.3.3.1 Infrastructure Access Service

The underlying infrastructure is expected to provide:

- **Infrastructure Balancing:** The Infrastructure Access includes any necessary load balancing/firewalls/routing required to enable a single point of access to the distributed service deployment: any user access is via the access portal, and not direct to the underlying virtual machines, containers etc.

4.3.3.2 Dynamic Processing Allocation Service

The Dynamic Processing Allocation Service supports the deployment, launch and release of computing elements on the available infrastructure. This service is accessed by the rest of the Platform via the Application Execution Management Service.

At least one of IaaS / PaaS / Bare Metal Resource Management has to be supported, in order to allow an application to be deployed on demand upon the platform in response to a user request. Support for more than one method, via the Dynamic Processing Allocation Management feature, is optional.

Specific Features:

- **Dynamic Processing Allocation Management (opt.):** Allows selection of the best execution environment for an application considering its requirements (e.g. executable type, performance requirements). The execution environment is selected from those available to the Exploitation Platform: they could be IaaS, PaaS or HPC environment. From amongst each category a specific provider is selected according to cost/performance/availability consideration. Interacts with the resource managers in the Dynamic Processing Allocation Service to physically deploy the application in the selected environment.

- **IaaS Cloud Resource Management (opt.):** Manages the provision of VMs on a IaaS cloud infrastructure according to the application execution requirements. Such requirements are described in a standardized format. It provides a web based M2M interface to:
  - Create a set of VMs;
  - Start a set of VMs;
  - Check VM status;
  - Shutdown a VM;
  - Hibernate a VM;
  - Destroy a VM.

- **PaaS Cloud Resource Management (opt.):** Manages the provision of containers in a PaaS cloud infrastructure according to the application execution requirements. Such requirements are described in a standardized format. Provides a web based interface to:
  - Instantiate a set of containers;
  - Check container status;
  - Save container status (to hibernate it to restart it in the future);
  - Destroy a container.

- **Bare Metal Resource Management (opt.):** Manages the provision of Bare Metal computing nodes according to the application execution requirements. Such requirements are described in a standardized format. Provides a web based interface to:
  - Request a set of computing nodes;
  - Check computing node status;
  - Deploy executable software on a computing node;
  - Release a set of computing nodes.

Tiers (Section 2)	Resource
Use Case Interactions (Section 3.1)	Delegate; Consume; Execute
Functions (Section 3.2)	Platform Support Functions: Resource Management.

Table 4-13: Dynamic Processing Allocation Service Traces