DELIVERABLE D3.1.1

Grant Agreement number : CIP-297300
Project acronym : InGeoCLOUDS
Project title : INspired GEOdata CLOUD Services

Funding Scheme : Pilot B

Analysis and monitoring of clouds for geo-data services

D3.1.1

Version 1.0

Reference D3.1.1-INGC

Project co-funded by the European Commission within the ICT Policy Support Programme		
Dissemination Level		
PU	Public	
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	X
CO	Confidential, only for members of the consortium (including the Commission Services)	

Contract Number :	CIP-297300
Document Title :	Analysis and monitoring of clouds for geo-data services
Document version :	1.0
Document status :	Approved
Date :	2012-07-13
WP contributing to the deliverable :	*WP3*
Availability :	*Confidential*
Authors :	CNR
Approved by :	InGeoCloudS Steering Committee

Abstract

This document is the first issue of a series of 3 documents. These documents survey existing cloud platform providers with the goal of choosing the best provider for the InGeoCloudS infrastructure. This first iteration integrates work performed in T3.1 and T3.5 by providing a first inventory of technical components, data, volumes and services usage profiles, infrastructure costs by every geodata provider of the consortium. The document also reviews the technical and commercial offer (spring 2012 status) of main actors of the cloud computing field. It then issues architectural guidelines for the definition of the basic InGeoCloudS cloud architecture.

Keywords List *Cloud Service Providers, Cost Estimate, Cloud Service Providers Monitoring.*

DOCUMENT CHANGE LOG

Document Issue.	Date	Reasons for change
Version 1-Draft 1	2012-04-12	Creation of the document
Version 1-FinalDraft1	2012-07-11	Integration of inputs and contributions from ALL partners after final changes.
Version 1-Approved	2012-07-13	Final editorial changes – minor corrections in section2 – approval through steering committee

APPLICABLE AND REFERENCE DOCUMENTS (A/R)

A/R and Document Reference	Title
[A1] ICT PSP Grant Agreement N° CIP 297300	InGEOCloudS Grant Agreement and its annex (including the description of work)

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1. ACRONYMS AND DEFINITIONS

Term	Definition
N/A	Not Applicable
CSP	Cloud Service Provider
GDAL	Geospatial Data Abstraction Library: a translator library for raster geospatial data formats
GIS	Geographic Information System
SEAM	An open source development platform for building rich Internet applications in Java
SOAP	Simple Object Access Protocol
SPARQL	Sparql Protocol And RDF Query Language
SQL	Structured Query Language
TBC	To be confirmed
TBD	To be defined
WSDL	Web Service Description Language

Table 1: Acronyms and Definitions

1.2. OBJECTIVES OF THE DOCUMENT

This document is the first issue of a series of 3 documents. These documents survey existing cloud platform providers with the goal of choosing the best provider for the InGeoCloudS infrastructure. The review of cloud platform provider will be based on the parallel and convergent analysis of:

- existing infrastructures and architectures of basic geodata services and tools published by the consortium members;
- use cases in the field of ground water management and geo hazards prevention/analysis;
- facilities offered by Cloud Service Providers for building InGeoCloudS infrastructure as a service model.

This first iteration integrates on-going work performed in T3.1 and T3.5 by providing a first inventory of technical components, data, volumes and services usage profiles, infrastructure costs by every geodata provider of the consortium. The document also reviews the technical and commercial offer (spring 2012 status) of main actors of the cloud computing field. It then issues architectural guidelines for the definition of the basic InGeoCloudS cloud architecture for Pilot1 (will be documented in details in D3.2 yet to be published, planned for M12). Subsequent iterations (M15 and M26) will potentially amend the followed strategy by taking into account the return of experience of Pilot1 architecture design but also the quickly evolving offers panel from the Cloud Service Provider (CSP) market.

1.3. OVERVIEW OF THE DOCUMENT

Section 2 reports on the analysis of the existing geodata services already running at the geodata providers sites. Services and datasets considered are mainly identified from the angle of those use cases addressed by the project (see notably D2.1 document, yet to be published, planned for M6). The goal is to understand the requirements of the cloud-based InGeoCloudS services.

Section 3 provides a logical view of the InGeoCloudS architecture, based on the requirements collected in Section 2.

A review of existing cloud platform providers is conducted in Section 4, by also estimating the monetary cost of a target cloud configuration. Section 5 draws some conclusions that will be used for the Pilot1 implementation.

2. ARCHITECTURAL REQUIREMENTS

2.1. APPROACH FOR GATHERING ARCHITECTURAL REQUIREMENTS

The goal of this section is to understand the requirements of a typical geodata service related to its deployment to a cloud computing platform. We thus collected a review of the existing services at the geodata providers sites of the consortium in order to estimate the typical software and hardware requirement of the InGeoCloudS platform to be developed. Such requirements are necessary to evaluate which cloud providers can support the InGeoCloudS services and the expected cost.

We asked each provider to describe the application, if present, currently used at their own site to serve the data at hand in terms of database management, file storage, map server support and web application environment. We also collected information about the data size, format and growth rate. We also requested resource requirement estimate in terms of number of accesses to be supported, storage volume, amount of memory and number of CPUs. This contributed to the definition of a reference set of requirement to be fulfilled by the InGeoCloudS platform. Finally, whenever possible we also asked to provide the cost of the current infrastructure running at the provider side.

In the following, we report the information gathered from each geodata provider of the consortium.

2.2. REQUIREMENT ANALYSIS FOR A GEO-SPATIAL INFORMATION SERVICE BY GEUS

Two kinds of groundwater-related datasets are contemplated in the frame of the project:

- **Pesticides in groundwater**. The idea is to make it possible for users to find areas where there are high concentrations of pesticides in the groundwater. It could be either pesticide in general or specific ones. It should also be possible to restrict the output to pesticides found at a certain depth interval and/or from certain geology (lithology or lithostratigraphy). Combining these data with the land use or information about the amounts of pesticides applied to the same areas and the surface geology (or even better 3D geological models) could give important information about the vulnerability of the aquifers. Potential users include NGOs, EEA, national environmental authorities, national or European environmental portals and researchers.

- **Groundwater levels.** The idea is to make it possible for users to extract measurements of groundwater level for input to hydrological flow modelling. These data needs to be harmonized across national boundaries as data in national databases very well can be related to different reference levels. Furthermore it will be important to be able to extract data originating from the same aquifer which means the measured levels should be combined with information about aquifer geometry or at least which aquifer is present at the depth where the water enters the wellbore. Potential users include water management authorities, consultants and researchers.

2.2.1. SOFTWARE REQUIREMENTS

The proposed services do not completely exist in a current implementation. A number of services that deliver the same kind of data exist on GEUS infrastructure, but the services for the cloud infrastructure must be targeted towards the specific use cases.

The infrastructure used at GEUS for the time being for serving data in the same manner as foreseen for the cloud services is shown below.

	Description
Operating System (OS)	*Windows Server 2008*
Web server	*JBoss*
Web Application Access	*WSDL/SOAP*
Database Server	*PostgreSQL (PostGIS) and Oracle.*
Geospatial components	*Mapserver, GDAL and ArcGIS Server.*
Development language	*Java*
Development framework	*SEAM*
Other components	*-*
Licensing issue	*Most components are Open Source except Oracle and ArcGIS Server. For that reason GEUS currently uses PostgreSQL for external access.*

2.2.2. DATA REQUIREMENTS

	Dataset name: Pesticides in groundwater
Nature of the data	*Regularly updated as new samples are taken.*
Data Maintenance	*Available for web service (Read) same day as data is received at GEUS.*
Format of raw data	*PostgreSQL database*
Format of geospatial data	*PostGis*
Constraints of access for data	*Public data.*
Data Model	*Basically 4 tables are needed:* *1: Borehole table holding borehole id, name and coordinates.* *2: Sample table holding borehole id, sample id, sample date, sample depth, laboratory, etc.* *3: Analysis table holding sample id, compound, measured value, unit of measurement, accuracy, etc.* *4: Borehole geology table holding borehole id, depth interval, lithology, lithostratigraphy, etc.*
Compliance with INSPIRE data model?	*Not currently.*
Involved by inspire (specify the annex)	*Yes, Theme Annex II.4, Geology (Borehole locations) and Theme Annex III.7, Environmental Monitoring Facilities (Borehole completion and groundwater samples).*

	Dataset name: ***Groundwater levels***
Nature of the data	*Regularly updated as new measurements are made.*
Data Maintenance	*Available for web service (Read) same day as data is received at GEUS.*
Format of raw data	*PostgreSQL database*
Format of geospatial data	*PostGis*

Constraints of access for data	*Public data.*
Data Model	*Basically 2 tables are needed:* *1: Borehole table holding borehole id, name and coordinates.* *2: Waterlevel table holding borehole id, measurement date, reference level, measured value, accuracy, etc.*
Compliance with INSPIRE data model?	*Not currently.*
Involved by inspire (specify the annex)	*Yes, Theme Annex II.4, Geology (Borehole locations) and Theme Annex III.7, Environmental Monitoring Facilities (Borehole completion and groundwater level measurements).*

2.2.3. RESOURCE REQUIREMENTS ESTIMATES

Expected hits number (if web)	*Not possible to estimate as we are currently not logging at such a detailed level.*
Expected users number	*Not possible to estimate as we are currently not logging at such a detailed level.*
Expected requests volume	*Not possible to estimate as we are currently not logging at such a detailed level.*
Expected data transfer volume	*Very little*
Expected data volume	*1.5 GB*
Expected growth rate	*0.2 GB of new data is added every year*
Expected memory requirement	*Unknown*
Expected computational requirements	*1-2 CPU's*
Quality of Services (QOS) required	*Currently we are obliged to have the services up and running 99% during working hours and 98% in the evening, so that could be a reasonable target.*

2.2.4. OUT-OF-CLOUD COSTS

It is unfortunately very difficult to make these estimates. The data and services of relevance to InGeoCloudS form a very small part of a very large infrastructure that includes internal applications and services and it will not make sense to try to divide the total costs based on size of tables or other measures.

2.3. REQUIREMENT ANALYSIS FOR A GEO-SPATIAL INFORMATION SERVICE BY GEOZS

GEOZS provides different types of services / application for geo-spatial use, candidate to be included in the cloud computing infrastructure.

- **Geo-hazards data dissemination** at 1/25.000 and 1/250.000 scale. The GeoZS provides information about geo-hazards induced by mass movement process. The dissemination enables effective access to the location of landslides and landslide susceptibility map. The service will be available in WMS and WFS.

⑥ **Dissemination of landslide triggering potential map due rainfall forecast (on daily basis).** GeoZS is building an early warning system that will be based on past landslide events, geology and weather (rainfall) forecast (ideally for the 52 hours in advance), prediction in a best possible way the areas/zones where the probability of triggering of landslides will be increased due to higher precipitation levels. The endangered zones will be determined/predicted using the combination of the landslide susceptibility model, precipitation and landslide triggering threshold values for every cell in the map at the scale 1:250.000 for the whole country.

2.3.1. SOFTWARE REQUIREMENTS

The proposed services do not exist in a current implementation. A number of services that deliver the same kind of data exist, but the services for the cloud infrastructure must be targeted towards the specific use cases.

The infrastructure used at GEOZS for the time being for serving data in the same manner as foreseen for the cloud services is shown below.

	Description
Operating System (OS)	*Windows Server 2008 R2*
Web server	*Apache, Tomcat,IIS*
Web Application Access	*REST*
Database Server	*Postgres, Mysql, MSSQL*
Geospatial components	*Mapserver X/GDAL/, ArcGIS Server 9.3 SP1 AND 10*
Development language	*ASP.NET, C#, Java*
Development framework	*.NET framework 2.0, 3.5, 4.0*
Other components	*Queue solution, Index search components, etc.*
Licensing issue	*Most components that we will be using in cloud are Opensource (Postgresql, and Mapserver) We are currently using Arcgis server Licence, MSSQL Licence and Windows server Licence.*

2.3.2. DATA REQUIREMENTS

	Dataset name: *Geohazard data - landslide susceptibility map 1/250.000 scale*
Nature of the data	*Static*
Data Maintenance	*Seldom updated as changes in a model are needed*
Format of raw data	*GRID*
Format of geospatial data	*PostGis*
Constraints of access for data	*Data only for authorities*
Data Model	*Value of susceptibility*
Compliance with INSPIRE data model?	*Not currently*
Involved by inspire	*Natural risk zones*

(specify the annex)	

	Dataset name: *Geohazard data - locations of landslides*
Nature of the data	*Dynamic, frequently updated*
Data Maintenance	*Add new/update data*
Format of raw data	*GRID*
Format of geospatial data	*PostGis*
Constraints of access for data	*Data only for authorities*
Data Model	*Localization of landslides, source, type...*
Compliance with INSPIRE data model ?	*Not currently*
Involved by inspire (specify the annex)	*Natural risk zones*

	Dataset name: *landslide triggering potential map due rainfall forecast*
Nature of the data	*Dynamic, streaming, Daily*
Data Maintenance	*Synchronization with web-services, ftp exchange files*
Format of raw data	*GRID*
Format of geospatial data	*PostGis*
Constraints of access for data	*Public data*
Data Model	*Model to be provided*
Compliance with INSPIRE data model ?	*Not currently*
Involved by inspire (specify the annex)	*Natural risk zones*

2.3.3. RESOURCE REQUIREMENTS ESTIMATES

Expected hits number (if web)	*Not possible to estimate*
Expected users number	*Not possible to estimate*
Expected requests volume	*Not possible to estimate*
Expected data transfer volume	*200MB per week (input data)*
Expected data volume	*1GB (approx.)*
Expected growth rate	*200MB per week (input data)*
Expected memory requirement	*4GB*
Expected computational requirements	*2 CPU*
Quality of Services (QOS) required	*99%*

2.4. REQUIREMENT ANALYSIS FOR A GEO-SPATIAL INFORMATION SERVICE BY IGMEM/EKBAA

IGMEM/EKBBA groundwater studies are currently not reflected on public accessible ITC infrastructures. Synthetic maps and scientific data are available through static web pages so far.

2.4.1. DATA REQUIREMENTS IGMEM

	Dataset name: Groundwater
Nature of the data	*static, read-only*
Data Maintenance	-
Format of raw data	*Database, Excel files*
Format of geospatial data	*Shapefiles*
Constraints of access for data	*Private data, public data,..*
Data Model	*Not available*
Compliance with INSPIRE data model?	*Not currently*
Involved by inspire (specify the annex)	*Yes, Annex II.4&5, Annex III.3, 4, 5, 12, 14*

2.4.2. RESOURCE REQUIREMENTS ESTIMATES

Expected hits number (if web)	**-**
Expected users number	-
Expected requests volume	-
Expected data transfer volume	-
Expected data volume	*2 GB*
Expected growth rate	-
Expected memory requirement	-
Expected computational requirements	-
Quality of Services (QOS) required	-

2.5. REQUIREMENT ANALYSIS FOR A GEO-SPATIAL INFORMATION SERVICE BY BRGM

BRGM provides different types of services / applications for geo-spatial use, candidate to be included in the cloud computing infrastructure. Two categories are identified:

Specific Data and tools to manage :

- **GroundWater** federation and dissemination (Ades, available with ades.eaufrance.fr). the BRGM manages and federates groundwater data acquisition in quality and quantity by partners in France. The global system manages the acquisition process and provides tools and services to integrate groundwater data in the federated database. The second system is on charge to publish public raw data in the field of groundwater. Only the second point will be studied on the InGeoClouds project.

- **Geology layer dissemination** at 1/1.000.000 scale. The data provided by the Brgm as the national geology survey, is published as a reference data in geology. The tool allows to interact with the attribute information associated to each polygon. The service is available in visualizing (WMS), interact (WFS) and download. see http://infoterre.brgm.fr or http://geoservices.brgm.fr/

- **Geo-hazards data dissemination** at 1/25.000 scale. The Brgm provides information about hazards in the field of landslides events, seismic events and underground cavities location. The dissemination includes geo-spatial access and data descriptive sheet. The service is available in visualizing (WMS), interact (WFS) and download. see http://infoterre.brgm.fr or http://geoservices.brgm.fr/

Generic tools available to answer INSPIRE requirements and uses cases :

- **CARMEN** (www.carmencarto.fr) is a web tool allowing any contributor of a public authority to push geospatial data, create a map with a online GIS tool and to publish the information as a web page, download access or like a web service (https://adullact.net/projects/carmen/). The web tool is based on open source solution and it from spatial database, OGC server to client browser. In the InGeoClouds project, Carmen could be analyzed as a tool to publish new data defined in the use cases. Current Data included (and described below) will not be pushed

- **GeoNetwork / +Géosource** (www.geosource.fr) is an open source tool to create geospatial catalogues compliant with Inspire and ISO metadata standards and to reference any information as well as allowing the dissemination of the environmental information between users, stakeholders, … (http://sourceforge.net/projects/geonetwork/). In the InGeoclouds projects, Geosource has to be analyzed as a tool which could be used to answer to the use cases.

- **GeoCache/exows service** is a web service allowing to access spatial data compliance with Inspire requirements while guarantying performance for end-users. These services are in particular useful for spatial reference data, like basemap provided by the open source Open Street Map (OSM) or official geospatial data (IGN in France for example). The solution adds compliance to INSPIRE requirements for services (OGC WMS and WFS) and data model specification. The solution is based on mapcache (http://mapserver.org/trunk/mapcache/) and exows projects (http://sourceforge.net/projects/exows/).

2.5.1. SOFTWARE REQUIREMENTS

2.5.1.1. Simple Web Mapping Architecture for ground water, geology and geo-hazards data access

On this platform, there are three different servers, as depicted in the following figure:

- Map Server in charge of the geo dataset dissemination as a OGC web services: geoservices.brgm.fr

- Database server in charge of the storage of geo-data : PostgreSQL server

⑥ [optional] Web server consumes the OGC services provided the first component or others externals OGC services. For example, a web server could provide a webmapping application, like infoterre.brgm.fr (named viewer.brgm.fr in the figure). The web server is not in charge of the direct data access, delegating this task to “mapserver” server.

All the three components have to be accessible by Internet, therefore have to be deployed in the DMZ area (or perimeter networking). Rather, all physical dataset (store as a file or as a database) should be deployed in a storage network not directly accessible by the Web (but through web services). The storage network is accessible for providers to put dataset in the dissemination infrastructure.

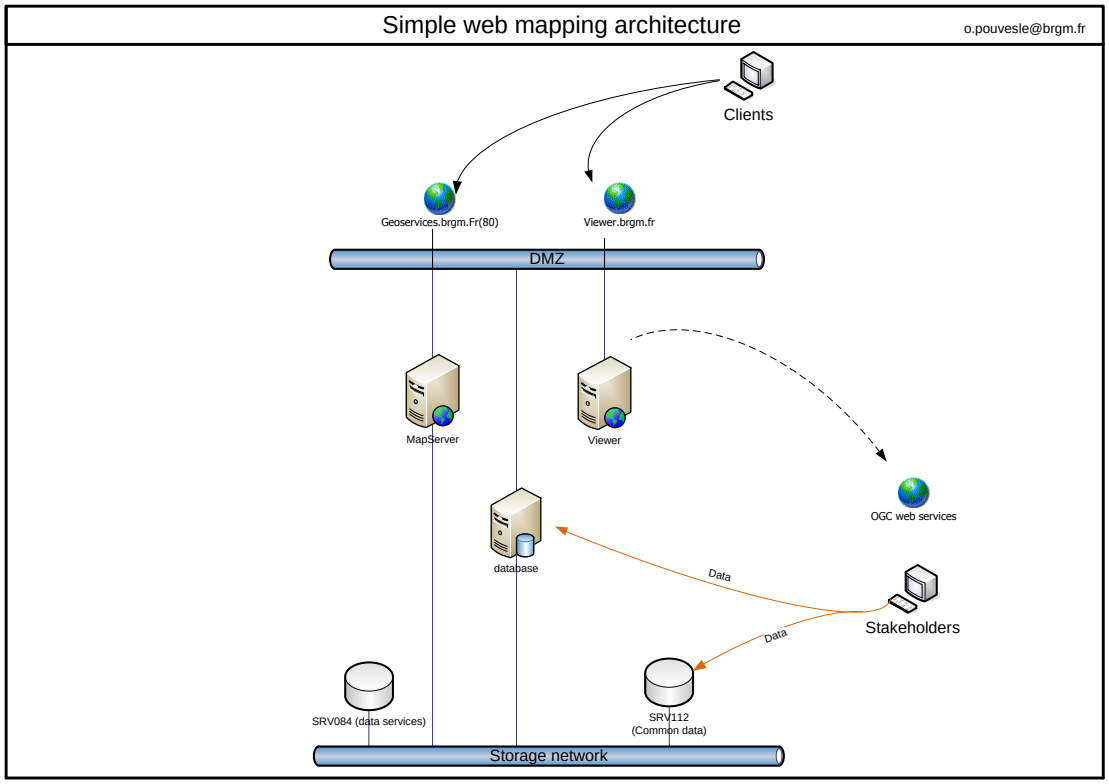

Figure 1: Technical architecture for the dissemination of geospatial dataset

Map Proxy cache server:

	Description
Operating System (OS)	*Linux CentOs Version 6.1*
Web server	*Apache 2.2.21*
Web Application Access	*REST*
Database Server	*/*
Geospatial components	*Mapserver 6.1*
Development language	*/*

Development framework	*C++*
Other components	

Database Server :

	Description
Operating System (OS)	*Linux CentOs Version 6.1*
Web server	*/*
Web Application Access	*/*
Database Server	*Postgres 9.0 + PostGIS 2.0*
Geospatial components	*/*
Development language	*/*
Development framework	*/*
Other components	*/*
Licensing issue	*OpenSource*

Java web server :

	Description
Operating System (OS)	*Linux CentOs Version 6.1*
Web server	*Apache 2.2.21 + JBoss*
Web Application Access	*REST*
Database Server	*/*
Geospatial components	*/*
Development language	*Java*
Development framework	*/*
Other components	*Infoterre Client*
Licensing issue	*OpenSource*

2.5.1.2. Carmen and GeoSource/GeoNetwork

The architecture described below is a typical technical architecture to provide a complete environment to publish geodataset in a production environment with scalabilty and QoS requirements. The environment includes platform to publish dataset as a service, create maps and metadata and to provide a webmapping platform for all citizens.

The platform refines the previous schema but some common aspects are always available: deployment in the DMZ environment and sharing network storage, differentiation of services per server. On this platform (named Carmen), there are:

- Several frontoffice servers (named *frontCarmenX*) in charge of the webmapping deployment. These servers are responding to the main URL of the platform (here, frtcarmen.ha.brgm.fr). To manage the scalabilty and performance, these servers are managed by a specific load balancer.

- One backoffice server (named *adminCarmen*) used only by providers to create new maps, OGC services,etc… The backoffice is available on a "private" URL (here admin.carmen…)

- One server dedicated to the download function (named *teleCarmen).* Indeed, this function is time-consuming and CPU-intense. To avoid freezing the others servers, the function is isolated in a specific server and can be only call as an asynchronous service.

- Several servers dedicated to the dissemination of data as a OGC service WMS/WFS (named *dataCarmenX).* The charge of the services could be very extensive; so, the number of dataCarmenX server is variable in function of the use with the use of a load balancer. Moreever, the OGC servers have a proxy as frontal (named *Geocache*) in charge to cache some useful dataset (images) and avoid calling too many times the same dataset. All the service are available with a OGC URL address (here datacarmen.ha.brgm.fr)

- A Database server (named *bdgeocarmen*) is used for Carmen to user geospatial data and geosource data

- For common dataset like background layer, the architecture use common servers dedicated to the WMS access for background *named mapsref.brgm.fr*

- The component *exows* (a java web server) is in charge of transform some OGC requests in INSPIRE compliance requests.

- A java web server dedicated to the metadata editor / publisher. The solution uses the Geosource software and are available on a specific URL address (metadata.carmencarto.fr)

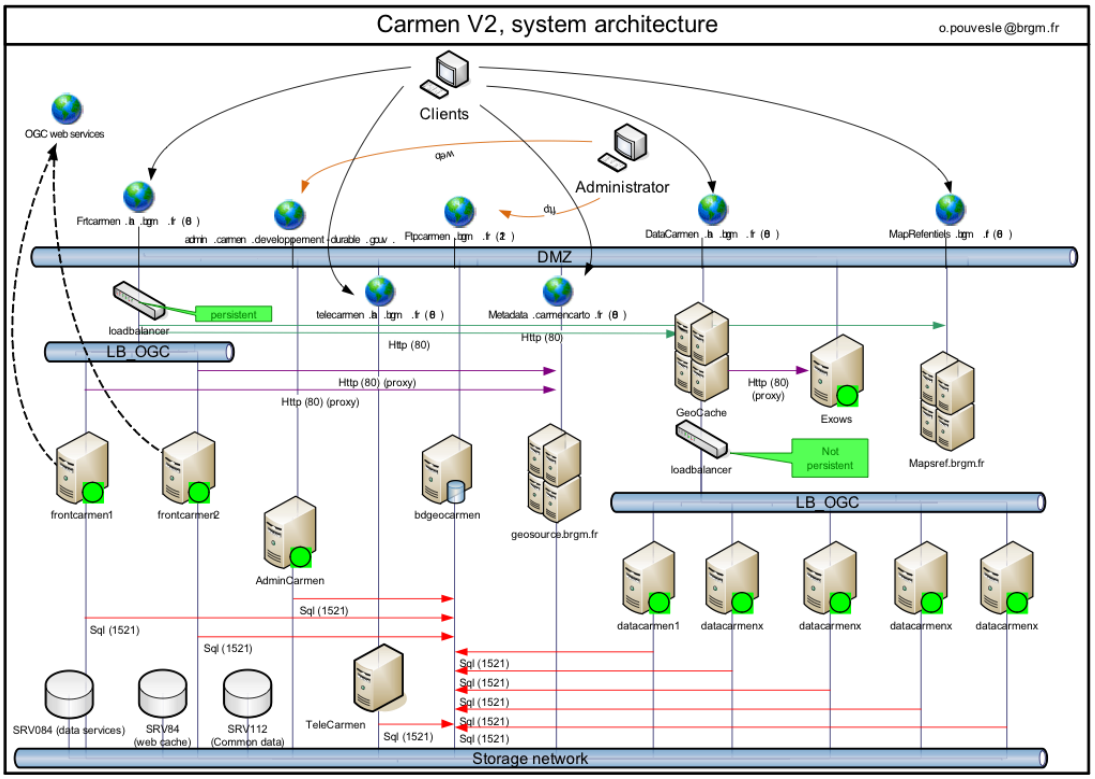

Figure 2: A typical complex architecture to publish geo-dataset in a production environment (BRGM)

Web Server:

	Description
Operating System (OS)	*Linux CentOs Version 6.1*
Web server	*Apache 2.2.21*
Web Application Access	*REST*
Database Server	*/*
Geospatial components	*Mapserver 5.7 /GDAL 1.6.0 / GEOS 3.3.1/ Proj 4.7*
Development language	*Php 5.2.9*
Development framework	*OpenLayer, phpmapscript*
Other components	*/*
Licensing issue	*OpenSource*

Database Server :

	Description
Operating System (OS)	*Linux CentOs Version 6.1*
Web server	*/*
Web Application	*/*

Access	
Database Server	*Postgres 9.0 + PostGIS 2.0*
Geospatial components	*/*
Development language	*/*
Development framework	*/*
Other components	*Vsftpd (FTP server)*
Licensing issue	*OpenSource*

Java web server :

	Description
Operating System (OS)	*Linux CentOs Version 6.1*
Web server	*Apache 2.2.21 + Tomcat 6.0*
Web Application Access	*REST*
Database Server	*/*
Geospatial components	*/*
Development language	*Java*
Development framework	*/*
Other components	*Geosource (GeoNetwork)*
Licensing issue	*OpenSource*

2.5.1.3. Geocache/exows solution

In Geocache service, two servers are used :

- Java web server (exows software)
- Map proxy cache server (Mapcache)

Java web server :

	Description
Operating System (OS)	*Linux CentOs Version 6.1*
Web server	*Apache 2.2.21 + Tomcat 6.0*
Web Application Access	*REST*
Database Server	*/*
Geospatial components	*Exows (http://sourceforge.net/projects/exows/)*
Development language	*Java*
Development framework	*/*
Other components	*/*
Licensing issue	*OpenSource*

Map Proxy cache server :

	Description
Operating System (OS)	*Linux CentOs Version 6.1*
Web server	*Apache 2.2.21*
Web Application Access	*REST*
Database Server	*/*
Geospatial components	*Mapserver 6.1 + mod_mapcache*
Development language	*/*
Development framework	*C++*
Other components	
Licensing issue	*OpenSource*

2.5.2. DATA REQUIREMENTS

The tables below describe examples of type of data which could be present in the InGeoCloudS infrastructure.

	Dataset name: *Groundwater data provided by partners in dissemination*
Nature of the data	*Dynamic*
Data Maintenance	*Provided by the production system with a automatic synchronization of the Oracle database to Postgres database*
Format of raw data	*Postgres database*
Format of geospatial data	*Postgres database*
Constraints of access for data	*Public data only*
Data Model	*model defined with the SANDRE conceptual model (sandre.eaufrance.fr)*
Compliance with INSPIRE data model ?	*partially.*
Involved by inspire (specify the annex)	*yes, Annex 3 - theme Environmental monitoring Facilities*

	Dataset name: *1/1.000.000 geology data provided by partners in dissemination*
Nature of the data	*static*
Data Maintenance	*Provided by the production system with a automatic synchronization of the Oracle database to Postgres database*
Format of raw data	*Postgres database*
Format of geospatial data	*Postgis database*

Constraints of access for data	*Public data only*
Data Model	*model defined by Brgm. (to provide later)*
Compliance with INSPIRE data model ?	*partially.*
Involved by inspire (specify the annex)	*yes, Annex 2 - Geology*

	Dataset name: *geo-hazards data, in particular landslide events (with localization), seismic (with localization) and underground cavities (with localization)*
Nature of the data	*Dynamic*
Data Maintenance	*Provided by the production system with a automatic synchronization of the Oracle database to Postgres database*
Format of raw data	*Postgres database*
Format of geospatial data	*Postgis database*
Constraints of access for data	*Public data only*
Data Model	*model to provide (later)*
Compliance with INSPIRE data model ?	*partially*
Involved by inspire (specify the annex)	*yes, with natural risk zones*

Carmen is going to publish all types of geospatial datasets.

	Dataset name: *Carmen dataset provided by partners*
Nature of the data	*Dynamic*
Data Maintenance	*ftp exchange.*
Format of raw data	*Filesystem defined below or Postgres/PostGis database*
Format of geospatial data	*Shapefile, MIF/MID, Postgres database*
Constraints of access for data	*Public data*
Data Model	*Flat dataset (geometry + attributes with [0.1] cardinality.*
Compliance with INSPIRE data model ?	*Some data published by the partner could be compliance with Inspire.*
Involved by inspire (specify the annex)	*Yes, in particular annex 2 and 3.*

Geonetwork/Geosource manages metadata compliance with ISO 19115 specification.

	Dataset name: *metadata created or imported in GeoNetwork*
Nature of the data	*Dynamic*
Data Maintenance	

Format of raw data	*Postgres database*
Format of geospatial data	*Not available. Geometry is stored in XML file as GML model*
Constraints of access for data	*Public or private metadata*
Data Model	*ISo 19115 model encoding in ISO 19139 schema XML*
Compliance with INSPIRE data model ?	*Yes. Compliance with metadata specification*
Involved by inspire (specify the annex)	*yes*

Geocache/exows is a web service allowing the transformation of an OGCWMS/WFS stream to a INSPIRE stream.

	Dataset name: *Transformation of geospatial dataset*
Nature of the data	*Dynamic*
Data Maintenance	
Format of raw data	*Configuration file to transform input stream to inspire stream*
Format of geospatial data	*GML*
Constraints of access for data	*Public data*
Data Model	*Transform "flat" gml to Inspire gml*
Compliance with INSPIRE data model ?	*Yes.*
Involved by inspire (specify the annex)	*yes*

2.5.3. RESOURCE REQUIREMENTS ESTIMATES

Note that the Carmen project requirements reported below result from a total of 150 data provider accounts. Even if a single account is likely to be "smaller" then an InGeoCLOUDS use case, these requirements can be used to estimate resources needed by a large adoption of the InGeoCLOUDS infrastructure.

Expected hits number (if web)	*696 954 hits per day*
Expected users number	*.*
Expected requests volume	*9 req/s (average), 1300 req/s (peak)*
Expected data transfer volume	*1275 GB per month*
Expected data volume	*380 GB*
Expected growth rate	*50 GB*
Expected memory requirement	*16 GB (Carmen) + 8GB (Geocache) + 8GB (Geosource)*
Expected computational requirements	*11 CPUs (Carmen)+ 12 CPUs (Geocache) + 6 CPUs (Geosource)*
Quality of Services (QOS) required	*98 % of available service*

GroundWater, geohazards and geology Platform:

Expected hits number (if web)	*104 055 hits per day*
Expected users number	*.*
Expected requests volume	*.*
Expected data transfer volume	*.*
Expected data volume	*Less 1 GB for geodata 44 GB for all groundwater observations*
Expected growth rate	*.*
Expected memory requirement	*4 GB (database) + 2 GB (mapserver) +2 GB (client)*
Expected computational requirements	*2 CPUs (database) + 2 CPUs (Mapserver) + 1 CPU (Client)*
Quality of Services (QOS) required	*98 % of available service*

2.5.4. OUT-OF-CLOUD COSTS

The estimated cost of platform is provided by year included the capital cost. The platforms equipments are amortized on a straight-line basis over a period of three to five years.

Carmen platform (with geosource solution)

Component	Estimation of capital cost	Estimation of operating cost / year	Comments of the methodology
Hardware acquisition and maintenance		*3 000 €/year*	*8 Virtual Machines with 11 CPU + 165 Go for system* *220 Go for high-performance storage for users.*
Infrastructural costs (power, air-conditioning, network access, security, …)		*6 000 €/year*	*To "divide" by applications hosted ?*
Technical personnel cost		*27 500 € / year*	*0,25 FTE*
Cost of supervision of the solution		*1 200 € / year*	
Cost of software licenses		*Opensource solution*	
Others costs			

Geosource Platform

Component	Estimation of capital cost	Estimation of operating cost / year	Comments of the methodology

Hardware acquisition and maintenance		*1200 € / year*	*4 Virtual Machines with 6 CPU + 60 Go for system* *5 Go for high-performance storage for users.*
Infrastructural costs (power, air-conditioning, network access, security, …)		*1 500 € / year*	
Technical personnel cost		*10 000 € / year*	
Cost of supervision of the solution		*1 200 €*	
Cost of software licenses		*OpenSource*	
Others costs			

GeoCache/Exows Platform

Component	Estimation of capital cost	Estimation of operating cost / year	Comments of the methodology
Hardware acquisition and maintenance		*8 000 € / year*	*4 VM – 12 CPU* *1To for storage geospatial dataset and cache tiles*
Infrastructural costs (power, air-conditioning, network access, security, …)		*2 000 € / year*	
Technical personnel cost		*10 000 € / year*	*0,1 FTE*
Cost of supervision of the solution		*1 200 € / year*	
Cost of software licenses		*Opensource*	
Others costs			

GroundWater, geohazards and geology Platform

Component	Estimation of capital cost	Estimation of operating cost / year	Comments of the methodology
Hardware acquisition and maintenance		*1008 € / year*	*1 VM – 2 CPU* *120 Go for storage geospatial dataset*
Infrastructural costs (power, air-conditioning, network access,		*1340 € / year*	

security, …)			
Technical personnel cost		*7500 € / year*	*0,1 FTE*
Cost of supervision of the solution		*1 200 € / year*	
Cost of software licenses		*Opensource*	
Others costs			

2.6. REQUIREMENT ANALYSIS FOR A GEO-SPATIAL INFORMATION SERVICE BY EPPO

Currently, there exists the *MONOGRAPHS* service (found in the URL http://monographs.itsak.gr), a web service that publishes Site Information on Accelerometer Stations of the Greater European Region. The service provides various geo-information about each site like morphological, geological, geotechnical and geophysical. The stations are visualized in a map using the Google Earth API. The information is presented in simple web pages. Geophysical and geological layers can be displayed over the map. The layers are ArcGIS shapefiles bundled into MXD files and are served to the application by ArcGIS Server. This service is currently used mostly by seismologists.

The *MONOGRAPHS* service will be extended to provide shake-maps for major earthquakes in Greek region. Shake-maps are maps showing ground movement and shaking intensity following major earthquakes. The shake-maps will be calculated using information about earthquakes that will be extracted automatically in near-real time (in a few minutes) from accelerogram records. The produced shake-maps come in several formats and will be provided to the public in two ways:

- as a layer over the earth map (the form of the layer remains to be decided)
- as a separate web page (see an example at http://earthquake.usgs.gov/earthquakes/shakemap/sc/shake/15115905)

The service will also send automatic notifications to interested actors.

The new service is expected to be used by public authorities, as an early warning system in case of a earthquake. It can also be used to propose the land usage (urban planning, type of structures), to increase preparedness in case of a strong earthquake event and to provide the first information in a disaster management system.

See also the following diagram.

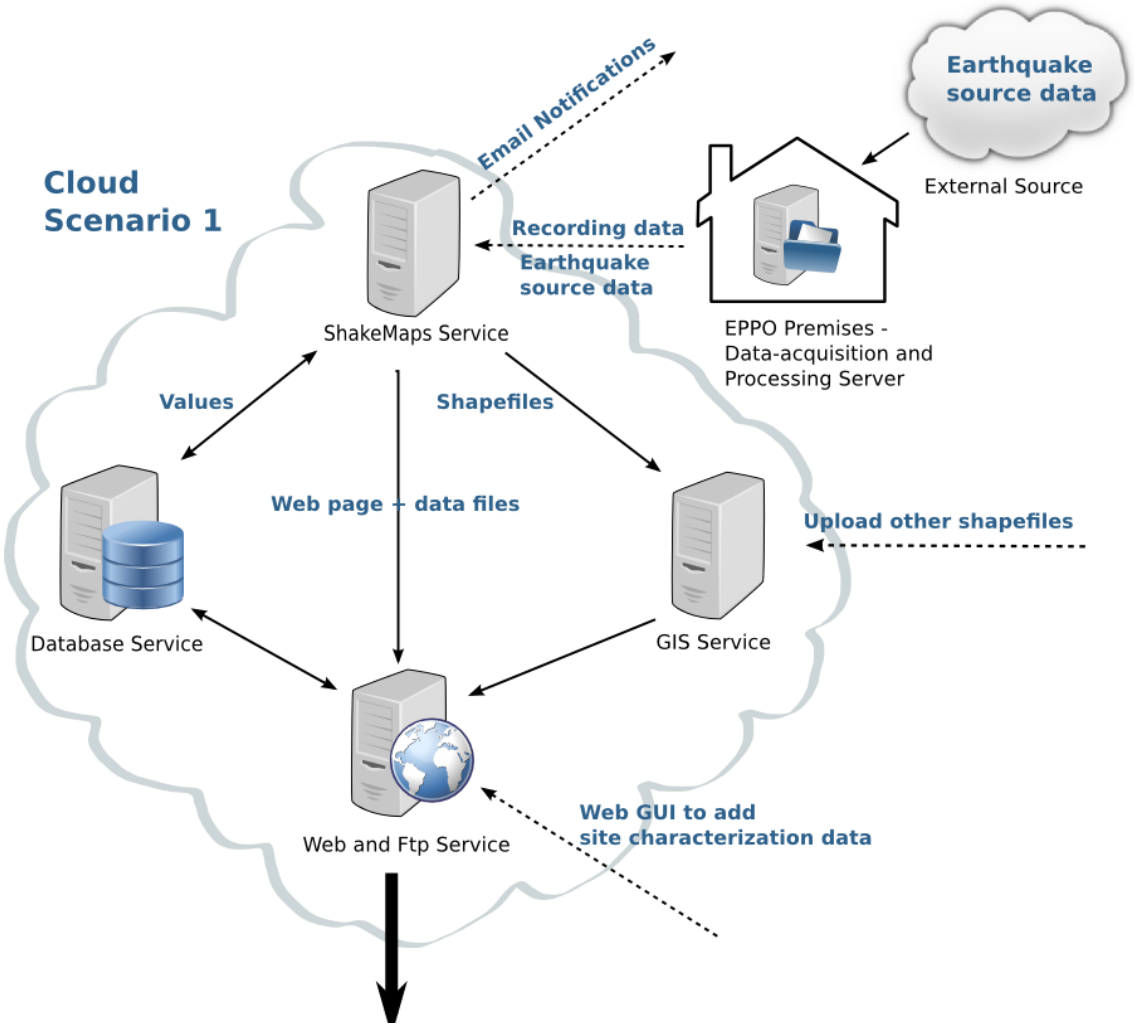

Figure 3. System architecture for publishing shake-map data

2.6.1. SOFTWARE REQUIREMENTS

Web and Ftp Server

	Description
Operating System (OS)	*Currently Fedora Linux*
Web server	*Apache 2.2*
Web Application Access	*No application server*
Database Server	
Geospatial components	
Development language	*PHP 5, Google Maps API*
Development framework	
Other components	*Ftp Server*
Licensing issue	*Open-source*

Database Server

Database Server currently runs in Windows 2003 Server but can be moved to a Linux system. It is used to store site characterization data and earthquake data.

	Description
Operating System (OS)	*Currently Windows 2003 (can be moved to a Linux system)*
Web server	
Web Application Access	
Database Server	*PostgreSQL 8.3 with PostGIS*
Geospatial components	
Development language	
Development framework	
Other components	
Licensing issue	*Open-source*

ArcGIS Server

The ArcGIS Server is used to serve shapefiles and mxd files to the web service. It is under investigation whether it can be replaced by an open GIS server.

	Description
Operating System (OS)	*Currently Windows 2003 (could be moved to a Linux system if replaced by an open GIS server)*
Web server	
Web Application Access	
Database Server	
Geospatial components	*ArcGIS Server 9.3.1*
Development language	
Development framework	
Other components	
Licensing issue	*Needs a license if ArcGIS*

ShakeMap Server

The ShakeMap Service will take earthquake recording data from an external source, process it and produce a set of shake-map files in various formats. These files will be made available to the public through the web and ftp service. The GIS service may also be involved to serve the produced shapefiles. The service will also send email notifications to interested actors.

	Description
Operating System (OS)	*Has been developed for Solaris & ported to FreeBSD. Probably runs on other unix-like systems with some modifications. Has to be tested.*
Web server	

Web Application Access	
Database Server	*MySQL 5.x used by the service to store the results of the processing*
Geospatial components	*None*
Development language	*Perl*
Development framework	*None*
Other components	
Licensing issue	*Open Source*

Data Acquisition and Processing Server (not to be moved to the cloud)

	Description
Operating System (OS)	*Windows 2003 and Linux Debian Squeeze*
Web server	*None*
Web Application Access	
Database Server	*None*
Geospatial components	*None*
Development language	
Development framework	*None*
Other components	
Licensing issue	Proprietary and open-source software

In the following we describe three possible integration scenario within a cloud computing platform.

Scenario 1: Full integration

According to this scenario all of the above services (except data-acquisition and processing) will be moved to the cloud. This is the best solution, as the service on the cloud will have no dependences on services on local data centres, thus fully exploiting the cloud potential and reducing the response times. This is the preferred scenario.

Scenario 2: ShakeMaps calculation local

The application that calculates the shake-maps runs locally and the results are uploaded to the cloud. Disadvantage: not taking advantage of the computational power that the cloud provides. Advantage: simpler implementation.

Scenario 3: ArcGIS and ShakeMaps Service run both locally

Only the web, ftp and database service are uploaded to the cloud. This means that the web Service has to contact the local GIS Service to get the shapefiles. This scenario is meaningful if it is not possible to replace ArcGIS Server with an open GIS Server. Disadvantage: see scenario 1 & 2. Advantage: no problems with ArcGIS license.

2.6.2. DATA REQUIREMENTS

	Dataset name: *Site Characterization Data*
Nature of the data	*Dynamic, read-only, regular update: daily at maximum*
Data Maintenance	*Web interface to add/update/delete*
Format of raw data	*Database – PostgreSQL, filesystem*
Format of geospatial data	*MXD files and shapefiles, PostGIS storage, txt, excel files, images*
Constraints of access for data	*Public data*
Data Model	*One table with site characterization data per site (parameter-value pairs for each site).*
Compliance with INSPIRE data model ?	*Not currently*
Involved by inspire (specify the annex)	*Partially Annex II (r2563, generated on 2011-07-22)*

	Dataset name: Earthquake Strong Motion data
Nature of the data	*Dynamic, read-only, updated after strong earthquakes, varies* between a few times per day to a few times per month
Data Maintenance	*Automatic command line shell script or program to add data.*
Format of raw data	*Database – PostgreSQL, filesystem*
Format of geospatial data	*Shapefile, PostGIS storage, txt, images, ps*
Constraints of access for data	*Public data*
Data Model	*Will be provided as soon as it is available.*
Compliance with INSPIRE data model ?	*Not currently*
Involved by inspire (specify the annex)	*Partially Annex III (r2563, generated on 2011-07-22)*

	Dataset name: Earthquake Source Data
Nature of the data	*Dynamic, read-only, updated after strong earthquakes, varies* between a few times per day to a few times per month
Data Maintenance	*Automatic command line shell script or program to add data.*
Format of raw data	*Database – PostgreSQL*
Format of geospatial data	*Attribute – value pairs*
Constraints of access for data	*Public data*
Data Model	*One table with earthquake information.*
Compliance with INSPIRE data model ?	*Not currently*
Involved by inspire (specify the annex)	*Partially Annex III (r2563, generated on 2011-07-22)*

2.6.3. RESOURCE REQUIREMENTS ESTIMATES

Expected hits number (if web)	*Normal: 1000/day* *Peak: 50.000/day*
Expected users number	*Normal: 20 users per day* *Peak: 1000 users per day*
Expected requests volume	*Normal: 300/day* *Peak: 15.000/day*
Expected data transfer volume	*Normal: 200MB per day* *Peak: 10GB per day*
Expected data volume	*8 GB*
Expected growth rate	*5 GB / year*
Expected memory requirement	*Not available at the moment*
Expected computational requirements	*Not available at the moment*
Quality of Services (QOS) required	*The service HAS to be available in ALL major earthquakes. This should be 99,9%.*

2.6.4. OUT-OF-CLOUD COSTS

The application is hosted at EPPO's Data Center.

The following table refers only to the available part of the service (the MONOGRAPHS application) which includes the web, database and GIS servers. The data-acquisition and processing server(s) as well as the high volume data storage are not included here. In general the maintenance costs of the service are very low.

Component	Estimation of capital cost	Estimation of operating cost / year	Comments of the methodology
Hardware acquisition and maintenance	*No new hardware was acquired*	*low*	*2 physical servers, 5GB RAM, 9 CPUs*
Infrastructural costs (power, air-conditioning, network access, security, …)		*low*	
Technical personnel cost	*10000 € (only for the MONOGRAPHS application – includes support and minor updates)*	*3000 € / year*	
Cost of supervision of the solution		*low*	
Cost of software licenses		*1000 € / year*	
Others costs			

3. DESIGNING THE INGEOCLOUDS ARCHITECTURE (T3.1: ALL)

3.1. INGEOCLOUDS ARCHITECTURE LOGICAL VIEW

On the basis of the requirements collected from the various data providers, we are able to define a high-level architecture for InGeoCLOUDS. This architecture will be further refined with MS11 "First pilot architecture definition", and it will drive the implementation of Pilot1. Deliverable D3.2 "Cloud architecture, configuration and data access implementation" will further improve the InGeoCLOUDS architecture. In Figure 4 we report a logical schema of the envisioned architecture.

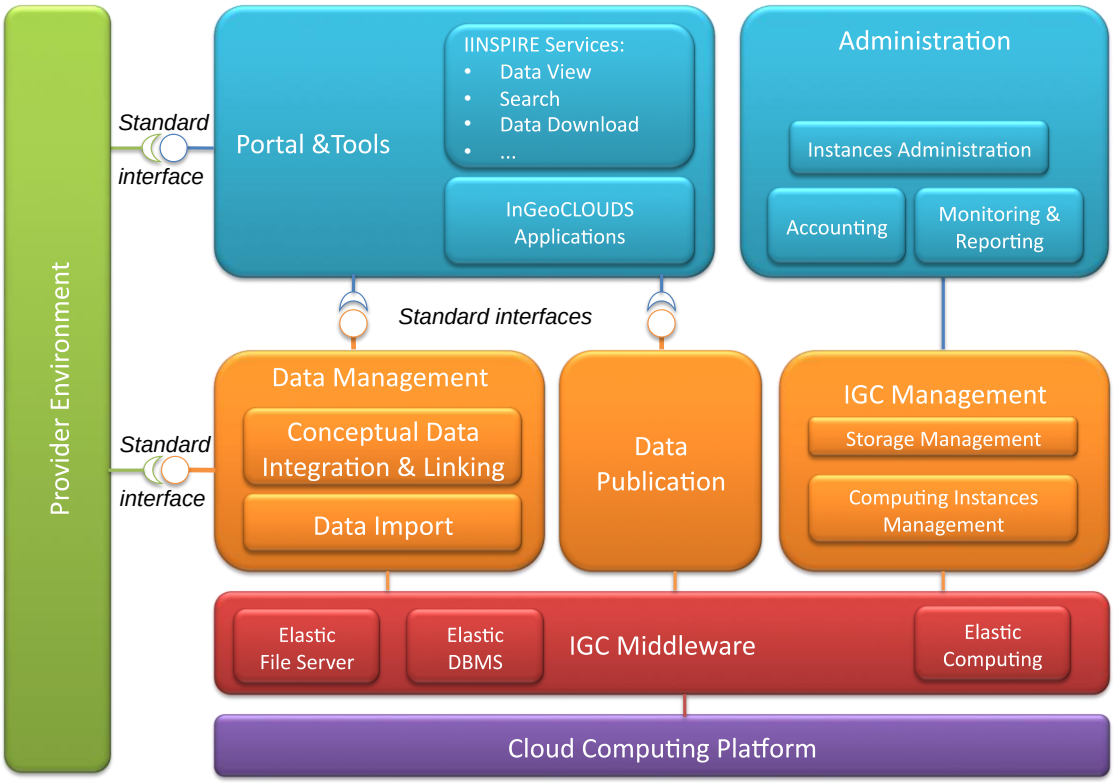

Figure 4: InGeoCLOUDS Architecture Logical Schema

We identified the following components:

⑥ **IGC Middleware**: this module provides storage and computing facilities to the other components of InGeoCLOUDS. Its goal is to make the IGC architecture independent from the tools and services offered by the specific cloud computing platforms. Among the services that will be implemented are: (a) "*Elastic File Server*" which should provide a file storage service transparently exploiting cloud elasticity in order to support large and varying volumes of data and users, (b) "*Elastic DBMS*" should similarly provide a spatially enabled database management system supporting replication to achieve reliability and scalability, and (c) "*Elastic Computing*" which should provide virtual instances on demand for all of the services running on the InGeoCLOUDS cloud-based architecture.

- **Data Management:** allows for seamless integration to the data published by different data providers. This module implements the services needed to push new geo-data into InGeoCLOUDS. It allows:
  - directly importing datasets without harmonized data storage (e.g. working / temporary data used by InGeoCloudS applications, specific geodata like shapefiles, etc.);
  - directly importing datasets compliant with IGC conceptual models through standard services (e.g. SOAP IGC Web Service);
  - implementing Export-Transform-Load processes for integration and linking of the different data sources in the infrastructure.
- **Data Publication**: this module makes it possible to access the InGeoCLOUDS data by means of standard geo-spatial services, e.g. OGC WMS, OGC WFS, OGC CSW, OGC SOS and semantic protocols, like SPARQL.
- **IGC Management**: this module provides the tools for the basic monitoring and management of the infrastructure, including the provisioning of virtual machines to all of the other services running on the InGeoCLOUDS infrastructure.
- **Portal & Tools**: this module is in charge of:
  - exposing INSPIRE compliant services, such as Data View, Search, Data Download, …
  - Integrating and giving access to specific applications (e.g. maps creation services, Web mapping interfaces, generation of Shakemaps, access to other applications made available by data providers …)
  - Technical facilities and tools (e.g. evaluating INSPIRE compliance from service regulation, data models, metadata perspectives…).
- **Administration**: this component of the architecture provides the tools for the business-level supervision of the infrastructure, of running applications. It also includes accounting facilities, authentication and authorization services for InGeoCloudS infrastructure.

Application and services on providers premises can interact with the platform for uploading or updating their data into the InGeoCLOUDS infrastructure, or for managing and interacting with other applications and services running in the cloud. The data is accessible only through services.

Existing applications and services will have to be reframed according to the InGeoCLOUDS architecture. On the other hand, the InGeoCLOUDS architecture will have to host and provide the services needed by these applications also exploiting the advantage of a cloud-based architecture. In the following section we revise the software packages currently in use at the geodata providers' sites.

3.2. OVERVIEW OF TECHNICAL COMPONENTS

This section compiles the information collected in Section 2 and identifies the technologies currently in use.

3.2.1. Operating Systems

		Windows		Linux			FreeBSD	Unspecified
		Server 2003	Server 2008	CentOS	Fedora	Debian		
GEUS			x					
GEOZS			x (R2)					
IGMEM								x
BRGM	Simple Web Mapping Architecture			x (6.1)				
	Carmen, GeoSource/GeoNetwork			x (6.1)				
	Geocache/exows			x (6.1)				
EPPO		x			x	x		

3.2.2. Web Servers

		Jboss	Apache	Tomcat	IIS	Unspecified
GEUS		x				
GEOZS			x	x	x	
IGMEM						x
BRGM	Simple Web Mapping Architecture	x	x (2.2.21)			
	Carmen, GeoSource/GeoNetwork		x (2.2.21)	x (6.0)		
	Geocache/exows		x (2.2.21)	x (6.0)		
EPPO			x (2.2)			

3.2.3. Web Application Access

		WSDL/SOAP	REST	Unspecified
GEUS		x		
GEOZS			x	
IGMEM				x
BRGM	Simple Web Mapping Architecture		x	
	Carmen, GeoSource/GeoNetwork		x	
	Geocache/exows		x	
EPPO			x	

3.2.4. Database Management Systems

		PostgreSQL	PostGIS	MySql	Oracle	MSSQL	Unspecified
GEUS		x	x		x		
GEOZS		x		x		x	
IGMEM							x
BRGM	Simple Web Mapping Architecture	x (9.0)	x (2.0)				
	Carmen, GeoSource/GeoNetwork	x (9.0)	x (2.0)				
	Geocache/exows	x (9.0)	x (2.0)				
EPPO		x	x	x			

3.2.5. GeoSpatial Components

		MapServer	mod_mapcache	GDAL	ArcGIS	GEOS	Proj	Exows	Unspecified
GEUS		x		x	x				
GEOZS		x		x	x (9.3)				
IGMEM									x
BRGM	Simple Web Mapping Architecture	x (6.1)							
	Carmen, GeoSource/GeoNetwork	x (5.7)		x (1.6.0)		x (1.3.3)	x (4.7)		
	Geocache/exows	x (6.1)	x					x	

EPPO				x (9.3.1)				

3.2.6. Programming Languages

		Java	ASP.Net, C#	C++	Php	Perl	Unspecified
GEUS		x					
GEOZS			x				
IGMEM							x
BRGM	**Simple Web Mapping Architecture**	x		x			
	Carmen, GeoSource/GeoNetwork	x			x (5.2.9)		
	Geocache/exows	x		x			
EPPO					x (5)	x	

3.2.7. Frameworks

		Seam	.Net	OpenLayer	phpmapscript	Google MAPS API	Unspecified
GEUS		x					
GEOZS			x				
IGMEM							x
BRGM	**Simple Web Mapping Architecture**						
	Carmen, GeoSource/GeoNetwork			x	x		
	Geocache/exows						
EPPO						x	

3.2.8. Others

		Queue solution	Index search solution	Infoterre client	FTP Server	Geosource	Unspecified
GEUS							
GEOZS		x	x				
IGMEM							x
BRGM	**Simple Web Mapping Architecture**			x			

	Carmen, GeoSource/GeoNetwork				x (vsftpd)	x	
	Geocache/exows						
EPPO					x		

3.2.9. Licences

		Open source	**Exception**	**Unspecified**
GEUS		x	Oracle, ArcGIS	
GEOZS		x	Windows, MSSQL, ArcGIS	
IGMEM				x
BRGM	**Simple Web Mapping Architecture**	x		
	Carmen, GeoSource/GeoNetwork	x		
	Geocache/exows	x		
EPPO		x	Windows, ArcGIS	

3.3. RDF TRIPLE STORAGE AND PROCESSING IN THE CLOUD

Along with the use of cloud computing as a data management platform, the recent years many parallel efforts have been started in order to support RDF described data storage and management on the cloud. Since RDF/S described data play a major role in the recent Linked Open Data initiative that is also of interest for the project we will try to review the existing efforts/offerings in the area in order to take this into consideration when a cloud platform will be evaluated and chosen. Since at the moment the process of requirements' elicitation from the data providers is not finalized we will refrain from reaching suitability conclusions; we will though try to provide a comprehensive reference for such platforms.

Dydra [2] is an RDF store relying on the Amazon Cloud infrastructure, which provides a SPARQL endpoint to query the data stored. Dydra is a commercial offering still in beta stage and many details are yet to be disclosed. Its authors claim through full SPARQL support and full exploitation of the cloud scaling capabilities. BigQuery [3] is a cloud computing offering by Google. It is deployed on Google App Engine (the Google offering for the cloud, reviewed earlier) and allows loading RDF/N-Triples content into Google Storage. They have also exposed an endpoint allowing querying the data but it does not support SPARQL queries as of now.

One system of interest is Stratustore [4], an RDF store that uses Amazon's SimpleDB as an RDF store back-end in combination with Jena's API. This project indexes all triples in SimpleDB using the subjects of the triples as items, the properties as attribute names and the objects as the values of the attributes, which causes SPARQL queries having a variable in the property position to remain unanswered; a proposed solution to insert one more entry per triple having as attribute names the objects with values the properties leads to a vast increase in storage space requirements.

A similar approach to Stratustore is proposed in [1], where is authors also use Amazon SimpleDB to support storage, indexing and querying of RDF triples on the cloud. The also use the full spectrum of Amazon Web Services (AWS) to offer to the user SPARQL based querying capabilities. They also propose different indexing strategies to overcome the limitations of the SimpleDB and promise to explore the newly introduced DynamoDB for the future.

Other efforts, like the CumulusRDF [7] system, use Apache Cassandra, a nested key-value store, as a triple store back-end. Cassandra is a clustered and thus scalable offering from Apache and can be used in a cloud environment, although this was not its primary target. The authors of [7] propose different indexing schemes to support the use of Cassandra for storing RDF triples and provide evidence for its feasibility.

Various works using MapReduce and related technologies are also available. Usually these works develop large-scale RDF stores using the MapReduce paradigm. For example, [8] introduces cloud computing in the area of Semantic Web early enough and gives some preliminary experimental results using Apache Hadoop (an implementation of MapReduce) and Pig, a tool that translates queries expressed in Pig Latin to MapReduce jobs. A similar approach is considered in [5] and [10]. Another effort [6] proposes partitioning of RDF files into smaller ones to be stored in HDFS, the file system of Hadoop. They also use summary statistics to determine the best plan to evaluate a SPARQL query. Evaluation of SPARQL basic graph pattern queries in a MapReduce framework is proposed in [9] where a multi-way join algorithm to process SPARQL queries efficiently is suggested, along with two methods to select the best query plan for executing the joins. Experiments were reported based on Cloudera's Hadoop distribution on the Amazon EC2. Finally, RDFgrid[1] is a framework that can be used for batch-processing RDF data with Hadoop, as well as Amazon's Elastic Map Reduce, which is in turn integrated with Amazon's DynamoDB.

3.4. CHOICE OF SOFTWARE PACKAGES AND TECHNICAL COMPONENTS

On the basis of the survey reported in Section 3.2, we are able to chose some of the software packages that can support most of the applications currently running on the provider environments, and that can be ported first to the InGeoCLOUDS architecture. Partially mapping bottom-up on the InGeoCLOUDS architecture depicted in Figure 4, we identified the following components:

- **PostgreSQL and PostGIS**. These are the preferred tools form managing the data and for implementing spatial functionalities on top of them. We will adopt them for the implementation of Pilot1. This choice must be reflected in the choice of the actual cloud platform provider: it must support PostgreSQL and PostGIS directly, in a Platform-as-a-Service fashion, or it must allow deploying them according to an Infrastructure-as-a-Service approach. This service will be implemented by the "IGC Middleware".

[1] http://rdfgrid.rubyforge.org/

- **MapServer**. This is the most used map server software in the consortium. Also other solutions, such as ArcGIS are adopted. Note that MapServer is open source, while ArcGIS poses some non-trivial licensing issues. Therefore, for the implementation of Pilot1 we plan to adopt MapServer software, and we will consider in inclusion of ArcGIS or similar at a later stage of the project. This service will by implemented within the "Data Publication" module.

- **Apache and Tomcat**. They are both used to host web applications. They are probably sufficient to support all of the applications that will be ported to the InGeoCLOUDS architecture. This kind of web application support will be provided by the "Portal & Tools" module.

- **Linux OS**. We notice equivalently popular use of Linux and Microsoft operating systems. Both operating systems are generally well supported by CSPs. However, for what regards Pilot1 we plan to support only Linux-based virtual instance, and to support other OSs at a later stage of the project. Note that direct access to a virtual instance, with the possibility of customizing them, might be needed in case of advanced applications, e.g. ShakeMap, or innovative services built on top of the InGeoCLOUDS platform. Such applications will be managed by the "Portal & Tools" module.

Given this preliminary analysis of the InGeoCLOUDS infrastructure and of its services, we can point out some relevant factors that are going to influence the choice of the underlying Cloud Service Provider. For instance, it is very clear, that many services are not going to be provided directly by a given cloud computing platform: the InGeoCLOUDS architecture must be able to deploy scalable and elastic services on top of some basic services and Infrastructure-as-a-Service approach made available by the given CSP. Some other criteria for the evaluation and choice of a CSP are discussed in the following section.

4. REVIEW OF EXISTING CLOUD COMPUTING PLATFORMS

4.1. EVALUATION CRITERIA

According to the requirements expressed by the data providers, we produced the following list of criteria to be evaluated for each Cloud Computing Platform:

1. ***Functional requirements***: whether or not the platform can support the management and publication of geospatial data (e.g. PostGIS).
2. ***Software requirements***: whether or not the platform is able to accommodate the software requirements (applications, software modules, licensing, development environments and tools, web server, etc.) collected from the data providers (e.g. Linux, Windows, etc.).
3. ***Elasticity model***: whether or not does the platform provides sufficiently large (storage/computation/bandwidth) "facilities" so as to support scalability. The fact that the provider has datacenter located in Europe should be considered as an important factor to guarantee reduced response time.
4. ***As-a-Service model***: Which of the three could computing service paradigms is provided: IaaS, PaaS or both, and which API of interest are provided.
5. ***Maturity and diffusion levels***: whether or not there is a lively developers community, an ecosystem of useful libraries and software components built around the platform that could make it easy to exploit the infrastructure. It is important to understand if the platform be available for the duration of the project and beyond.
6. ***Migration cost model***: whether or not the platform involves some lock-in effect, thus making it difficult to migrate to different cloud providers in the future. This is also related with the As-a-Service model, since PaaS paradigms usually induce larger migration costs.
7. ***Economic cost model***: Estimate of the cost of the InGeoCloudS project.

As the last criterion is particularly important, we used the information collected from the data providers to make a quantitative estimate of the incurred cost. At the current stage, the infrastructural constraints submitted by the data providers are as follows:

Table 2: Estimated Resource Requirements from data providers.

	GEUS	***GEOSZ***	***IGMEM/ EKBBA***	***BRGM small***	***BRGM large***	***EPPO***
Exp. requests volume (req/sec) (avg)	NA	NA	NA	NA	9	0.003
(max)					1300	0.17
Exp. transfer volume (GB/month) (avg)	NA	NA	NA	NA	1275	6

(max)						300
Exp. data volume (GB)	1.5	NA	2	45	380	8
Exp. growth rate (GB/year)	0.2	NA	NA	NA	50	5
Exp. memory req'd (GB)	NA	NA	NA	8	32	NA
Exp. comp. req'd (CPU)	2	2	NA	5	29	NA
Exp. QOS req'd	99%	99%	NA	98%	98%	99.9%

According to the provided resource requirements estimates, the following monthly infrastructure requirements have been identified, for two differently-sized configurations:

Table 3: Monthly Infrastructure requirements identified from use cases

Resource	***Typical***	***High Peak***
CPU	3	29
Memory (GB)	8	32
Initial Storage (GB)	45	380
Storage growth (GB/year)	5	50
Final Storage* (GB)	70	630
I/O Requests (million req.s/month)	34	3,400
Network Traffic (GB/month)	13	1,275
Exp. QOS	99%	99.9%

*: This value is computed with an horizon of 5 years.

In Table 3 we report two kinds of resource requirements. The first configuration, named "Typical", was derived by averaging the requirements collected when they were comparable (e.g. CPU numbers from GEUS, GEOSZ and BRGM "small"), or by taking the BRGM "small" as a reference. The value of 34 million I/O requests per month was obtained by considering the BRGM "large" peak of 1300 req/s and scaling it down by a factor of 100, which is reasonable according to the average vs. maximum requirements ratio reported by BRGM and EPPO. The second configuration, named "High Peak", tried to predict the peak requirement of the infrastructure, and it was derived from the BRGM "large" maximum requirements information.

In order to estimate the cost of a given cloud computing platform, we considered a combination of "Typical" and "High Peak" requirements aimed at hosting six use cases. We assumed that each use case has on average the same requirements of the "Typical" configuration, and that the peak of the overall InGeoCLOUDS infrastructure can be supported by the additional resources modeled by the "High Peak" configuration. This is reasonable since the BGRM "large" configuration hosts 150 data providers accounts, and it is not likely that all the six use cases will exhibit peaks simultaneously. Also, in order to provide a conservative estimate, we assume that the peak configuration is needed for 10% of the total uptime, which is a non trivial amount of time. In conclusion, we assume the "Typical" infrastructure requirements specified in Table 3 are adequate to support the activity of a single data provider application, while for the 10% of the uptime some additional resources are required as described by the "High Peak". Accordingly, The total cost is defined as follows:

$$\text{Total Cost} = 0.90 \cdot [6 \cdot \text{cost}(\textsc{Typical})] + 0.10 \cdot \text{cost}(\textsc{High-Peak})$$

4.2. EVALUATION OF CLOUD SERVICE PROVIDERS

The consortium decided on the list of CSPs to be evaluated as first iteration of the process. D3.1.2 and D3.1.3 are meant for providing updates in this evaluation in the course of the project. We do cannot provide here a comprehensive list of providers: the market is highly dynamic and moving on a daily basis, Moreover, the “cloud” and “<whatever>AsAService” labels are used by numerous companies and institutions for qualifying their service offers but this does not necessarily reflect their capacity of complying with main key characteristics of cloud computing offers:

- Elasticity (scale as you want)
- Pay per use (pay only real resources usage)
- Self-service (on-demand and quasi-real-time provisioning).

As such we contemplate here verified and major actors but also those emerging companies that have been recently gaining recognition from the market in Europe, often coming from the hosting / data center business (e.g. OVH).

4.2.1. AMAZON EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a Web service that provides resizable computing capacity in the cloud by presenting a true virtual computing environment. This allows exploiting Web service interfaces to launch instances with several operating systems, load them with a custom application context, manage the network's access permissions, and run the process image using the desired subset of features the system comes with.

To use Amazon EC2 you simply:

- Create an Amazon Machine Image (AMI) containing all your software, including your operating system and associated configuration settings, applications, libraries, etc. Think of this as zipping up the contents of your hard drive. Amazon EC2 provides all the necessary tools to create and package your AMI.
- Upload this AMI to the Amazon S3 (Amazon Simple Storage Service) service. This gives Amazon a reliable and secure access to your AMI.
- Register your AMI with Amazon EC2. This allows Amazon to verify that your AMI has been uploaded correctly and to allocate a unique identifier for it.

Use this AMI and the Amazon EC2 Web service APIs to run, monitor, and terminate as many instances of this AMI as required. Currently, Amazon comes with command line tools and Java libraries but you may also directly access the SOAP-based API programmatically. In order to do that, Amazon provides several SDKs (i.e., AWS SDK for Java, AWS SDK for .NET, AWS SDK for PHP, AWS SDK for Ruby, AWS SDK for Android, and AWS SDK for iOS).

While instances are running, you are billed for the computing and network resources that they consume.

Amazon has data centers in different areas of the world (e.g., North America, Europe, Asia, etc.). Correspondingly, EC2 is available to use in different Regions. By launching instances in separate Regions, you can design your application to be closer to specific customers or to meet legal or other requirements. Prices for Amazon EC2 usage vary by Region.

Criterion	Evaluation
Functional Requirements	**OK:** It guarantees 99.95% uptime and 1.7GB RAM, 160GB local storage, 1 EC2 Compute Unit.
Software Requirements	**OK:** OS: - RedHat Enterprise Linux, SUSE Linux, openSUSE Linux, Fedora, Debian, OpenSolaris, Cent OS 5.4, Gentoo Linux, Oracle Enterprise Linux, Ubuntu 10.04, and Ubuntu Linux - Windows Server 2003 and 2008 Language and Tools:

	- Java, PHP, Python, Ruby, WinDev
Elasticity Model	**OK**: Amazon EC2 enables you to increase or decrease capacity within minutes, not hours or days. You can commission one, hundreds or even thousands of server instances simultaneously. Of course, because this is all controlled with web service APIs, your application can automatically scale itself up and down depending on its needs (auto scaling, elastic load balancing). Several Instance Families: Standard, Micro, High-Memory, High-CPU, Cluster Compute, Cluster GPU each one coming with many combinations of main memory amount, CPUs, local secondary storage. Free: - Autoscaling and monitoring With charges: - Load balancing, virtual private servers, and file hosting
As-a-Service Model	IaaS – RESTful-based and SOAP-based API (Java, .NET, Python, PHP, Ruby). PaaS for some specific service, e.g. Dynamo DB.
Maturity and Diffusion	**OK:** No free support. Premium support available 24/7. Several Wikis, forums, FAQ sites, as well as articles, libraries, and code snippets highlight that the community is very active.
Migration Cost Model	**OK**: No vender-lock system, which makes the job of moving the code to another box quite easy, as long as IaaS model is used.
Economic Cost Model	Billing depends on the sizes of instances with hourly rates based on the actual or virtual hardware reserved for the instance. Typical: $ 104,75 / month Peak: $ 1542,92 / month Total Cost: 782,79$ / month

Amazon EC2 might be considered as a suitable choice due to its elastic, completely controlled, flexible, reliable (99.95% availability guaranteed for each Amazon EC2 Region), safe, and inexpensive environment.

4.2.2. SIGMACLOUD

CloudSigma is pure Infrastructure-as-a-Service (IaaS) provider offering a flexible web based and API driven platform based in Zurich, Switzerland, their servers are hosted in the Interxion data center near Zurich.

CloudSigma was founded in 2009, their main philosophy aim to provide unrestricted choice of OS, unlimited size of servers and no restrictions about software running on servers.

CloudSigma is really a full IaaS provider, they allow customer to customize everything they need on their servers, like CPU, RAM, storage, bandwidth. The only restriction is that software have to be abble to run standard AMD/Intel architecture. As it said earlier, you can run almost any existing OS on CloudSigma servers, indeed, CloudSigma provides a big amount of pre-installed server (like Centos or Fedora). Moreover you can upload with an FTPS access your own drive (you can also download a whole drive with FTPS).

About drives, CloudSigma has become the first established cloud provider to add solid-state-drive storage to its public cloud computing service. SSDs (flash memory) are known for their ability to significantly increase storage I/O performance and decrease power consumption when compared with HDDs. That's one of the things that make CloudSigma one of the most innovative cloud provider in Europe.

To manage servers, CloudSigma offers a nice looking web console but you can also use their script APIs access.

For the pricing, two approaches are possible:

- a per unit and per hour pricing (CPU, RAM, data storage, data transfer);
- burst pricing: dynamic on-demand pricing platform that varies the price of metrics in relation to the utilization rate of their cloud.

Criterion	Evaluation
Functional Requirements	**OK** : You can make the server as you wish, just have to take care of software licence
Software Requirements	**OK**
Elasticity Model	**OK**: • Scaling is (relatively) easy • Datacenters located in Europe
As-a-Service Model	**Pure IaaS**
Maturity and Diffusion	**OK** : Still young but very promising with strong will of innovation
Migration Cost Model	**OK** : You can download whole drives with FTPS access on ISO format
Economic Cost Model	Typical: € 176/ month Peak: € 888/ month Total Cost: € 1144.8 / month

4.2.3. ATLANTIC.NET

Atlantic.Net Hosting Solutions include an array of hosting services: Colocation, Cloud Server Hosting, Virtualization Hosting, Shared Web Hosting, Managed Server Hosting, and Dedicated Server Hosting Solutions. It owns and operates on its data center infrastructure and it is dedicated to implementing tailored hosting solutions that enable clients to enjoy the benefits of cost savings.

Criterion	Evaluation
Functional Requirements	**Not enough information available**
Software Requirements	OS: - Ubuntu, CentOS, Fedora, and Debian - Windows Server Language and Tools**: No information available**
Elasticity Model	U.S. data centers only
As-a-Service Model	IaaS – RESTful-based API.
Maturity and Diffusion	**KO:** RESTful-based API has been recently introduced to developers. No active communities.
Migration Cost Model	**OK**
Economic Cost Model	Billing computing power depends on the sizes of instances with hourly rates based on the actual or virtual hardware reserved for the instance. Typical: $ 352,22 / month Peak: $ 2642,25 / month Total Cost: 2377,54$ / month

Unfortunately, there is not a lot of additional information about the platform available on Atlantic.Net's Web site. However, it apparently does not meet the set of needed requirements.

4.2.4. FLEXIANT FLEXISCALE

Flexiscale is a PaaS offering Cent OS, Debian, Ubuntu, and Windows Server images to run on the service. Specific images created by customers are also supported. Each customer gets a private VLAN and the storage is virtualized and is kept in a separate Storage Array Network rather than in the physical hosting server. The separation of storage and the hosting server gives the capability to turn off a server and only pay for the storage for a period of time. The platform supports snapshots of existing servers enabling backup or starting of multiple servers with the same configuration. Flexiscale monitors physical servers and restarts instances on crashed servers automatically on other servers within fifteen minutes.

Criterion	Evaluation
Functional Requirements	**OK**: It guarantees 100% uptime and FlexiScale services are priced in units.

	Before you deploy any FlexiScale service, you will need to buy a package of units. 1 unit is 1.7 cents or less. The base plan includes: 0.5Gb RAM, the cost for storage and bandwidth is not included.
Software Requirements	**OK** OS: - Cent OS, Cent OS 5.3, Debian, Ubuntu Linux - Windows Server 2003 and 2008 Support for specific images created by customers Language and Tools: The cloud computing provider offers root access to the servers, all the programming languages are supported by the provider
Elasticity Model	**KO**: European data centers only No Autoscaling Free monitoring and virtual private servers File hosting service with charge
As-a-Service Model	PaaS – SOAP-based API.
Maturity and Diffusion	**KO:** phone, support ticket (24/7 critical errors, Mon-Thu 9:00-5:30, Fri 9:00-4:30 GMT). But community seems not so active (i.e., only few simple guides).
Migration Cost Model	**OK**
Economic Cost Model	Billing computing power depends on the sizes of instances with hourly rates based on the actual or virtual hardware reserved for the instance. Typical: $ 230 / month Peak: $ 12362 / month Total Cost: 8659,20$ / month

4.2.5. GoGrid

GoGrid is a pure IAAS solution; it builds more on the classical hosting approach than cloud paradigm. But, the company proposes now a cloud offer, with the respect of two important principles: pay-on-demand (pay-as-you-go) and an API to build and manage our infrastructure in the cloud.

GoGrid cloud hosting allows building scalable cloud infrastructure in data centers using dedicated and clouding servers. The company proposes load balancing, virtualization and cloud storage. The company is more specialized in the management of hybrid infrastructure: combines cloud computing with traditional hosting giving you the best of both worlds for hosting and scaling production web applications. The functionality is not the sandbox of the InGeoClouds project.

GoGrid allows creating different types of environments and uses a catalog of instances (cloud images): MyGSI. The catalog allows to create quickly instances of servers in regard of use and to share public images between communities. However, the community doesn't seem very active, in comparison with Amazon Web Services instances. No tools or easy solutions seem to be available to migrate MyGSI instances to another IAAS cloud or vice-versa.

GoGrid defines a API to programmatically access the functions that are available through our UI (create instance, delete, ...). The API is a proprietary REST-like service and little documentation is available on line (https://wiki.gogrid.com/wiki/index.php/API_Getting_Started_Guide).

GoGrid uses data centers only in America continent (no Europe datacenter). The network distance could reduce the performance for European project.

Criterion	Evaluation
Functional Requirements	**OK**: You install what you want. The cloud storage is a direct access in SCP or SAMBA. So geospatial database could be used.
Software Requirements	OS: **OK** (Linux based system, or Windows Server 2008) Language: **OK** Tools: **OK.**
Elasticity Model	**KO**: 3 components for the price (RAM use, transfer usage, storage usage) Cloud storage and Content delivery Network No available notion of automatic elasticity (auto-scaling of servers,)
As-a-Service Model	IaaS
Maturity and Diffusion	**KO:** the community does not seem very active.
Migration Cost Model	**KO:** No solution found
Economic Cost Model	Charges is defined by type of servers, storage size and data transfers Typical: 356,00 $/ month (with only 4 GB of RAM but 4 CPU) Peak: 2838 $ / month Total Cost: 2206 $ / month

Gogrid is an interesting solution for US market and for hybrid cloud solution, in particular for IT company. The company proposes a cloud solution but the price, the location and the technical solution - in particular the elasticity - are less efficient than others solutions.

4.2.6. GOOGLE APP ENGINE

Google App Engine (GAE) is a cloud computing service based on the PaaS (Platform as a Service) model founded in 2008. This platform is used for developing and hosting web application on Google data centers.

One interesting point with this solution is that GAE is free up to a certain level of used resources as reported in the following table:

Service	Free quota / day	Maximum quota charge / day
Request number	1300 000	43 000 000
Bandwidth in	1 GB	1046 GB
Bandwidth out	1 GB	1046 GB
CPU time	6.5 hours	1729 hours
Datastore call number	10 000 000	140 000 000
Data size	1 GB	No maximum

Moreover GAE offers automatic scaling for web application so if the numbers of requests increases for an application, there is an automatic allocation of resources to handle the demand and so on fees are charged for additional resources.

From a technical point of view there are some restrictions which are quite important like development languages: only python, java (JVM Langage too) and GO (no PHP) and GAE datastore for the storage which didn't use SQL but GQL (so you can't use DB like PostgreSQL).

A good point is, to develop an application, Google provides for each language a set of APIs for accessing various services :

- Memcache : a cache over the database
- URL Fetch : cam make HTTP / HTTPS requests to another server
- Email : can send or receive email
- Images : allow to manipulate images (rotate, scale, etc...)
- Google Accounts : allow Google accounts to login to an application
- XMPP : can send and receive messages formatted as XMPP
- Task queues : allows to put background tasks in the queue
- Con : can schedule tasks to run on a recurring way
- Channel API : can create a push communication
- Back-ends : can create permanent instances of an application with access to more memory

In term of reliability, applications have 99.95% uptime SLA (Service Level Agreement). Indeed GAE is designed to sustain multiple datacenter outages without any downtime.

Criterion	Evaluation
Functional Requirements	**OK**: There is some open source projects on GAE like GeoModel or Geodatastore to manage an publish geospatial data in Json or KML format.

Software Requirements	OS : **OK** (Linux based system, or Windows Server 2008) Language : **KO** (no PHP), also many important classes in the standard Java Development Kit are not available like javax.imageio Tools : **KO**, GAE provide a kind of sandbox where your application can run, but you can't add tools like Postgres database or anything like this, you have to deal with the environment GAE give to you. If you want a relational database, there is the project Google Cloud SQL which allows you to create, configure, and use relational databases with App Engine applications (a MySQL DB).
Elasticity Model	OK : • Generous free quota • Billed for actual CPU usage, not "live" hours • Scaling is (relatively) easy • Several datacenters located in Europe
As-a-Service Model	PaaS (but limited to Java / Python environment)
Maturity and Diffusion	**OK :** there is a big community around this platform and more and more open source project.
Migration Cost Model	**OK :** There is some open source project to make it easy to migrate your project to another cloud provider like AppScale, TyphoonAE or Web2py.
Economic Cost Model	**NA**

4.2.7. JOYENT

Joyent has developed their own cloud computing platform, based on the operating system Open Solaris they have created Joyent Smart OS that replaces the hypervisor and traditional operating systems used in other IaaS platforms. This platform is sold by Joyent to customers wanting to build a cloud of their own and is also used to provide a public cloud managed by Joyent. Joyent offers two kinds of products on their public cloud, SmartMachines and Virtual Machines. SmartMachines uses the Joyent Smart OS and comes with Apache, Python, Ruby on Rails, Java, and SVN preinstalled. There are also SmartMachines specialized for MySQL, Riak (a scalable open source key/value store database), and Zeus (a load balancer) available on the Joyent cloud. SmartMachines share a hardware resource pool and are capable of CPU bursting when load increases. Virtual machines run CentOS, Debian, Ubuntu or Windows Server operating systems. As opposite to most other vendors, Joyent charges by month instead of by hour for used instances.

Criterion	Evaluation
Functional Requirements	**OK**: It guarantees 100% uptime and Small 1GB RAM 1CPU SmartOS or

	Linux instance.
Software Requirements	OS: **OK** - Cent OS, Debian Linux, Fedora, Open Solaris, Ubuntu Linux - Windows Server 2008 Language and Tools: **OK** The cloud computing provider offers root access to the servers, all the programming languages are supported by the provider
Elasticity Model	**OK** U.S. data centers only Free: - Data protection persistency, Autoscaling, Virtual private servers With charge: - Backup storage, Load balancing, Monitoring, File hosting service
As-a-Service Model	IaaS – CloudAPI (RESTful-based).
Maturity and Diffusion	**KO:** No significant active communities, but 8/5 phone, mail, and chat tech free support is provided.
Migration Cost Model	**OK**
Economic Cost Model	Charges apply differently sized instances by month instead of by hour. Typical: $ 467,20 / month Peak: $ 4088 / month Total Cost: 3212$ / month

4.2.8. MICROSOFT AZURE

Windows Azure is a PaaS offering where developers can create applications using .NET, Java, Ruby, or PHP. In addition, Windows Azure AppFabric and SQL Azure and Windows Azure Marketplace compliment Windows Azure to constitute the whole Microsoft cloud platform. Windows Azure in turn is made up of several components, namely, compute, storage, connect, fabric controller, and CDN.

Compute is the component that runs the applications. Applications can be created using three kinds of roles: web roles, worker roles and VM roles. Applications can have just one role or several and each role can be run in one or more instances. Web roles are used for Web-based applications and are invoked by Web requests. Worker roles are intended to run background jobs and do not interact with the user directly. VM roles are a bit different: they run Windows Server images, which is more of an IaaS concept. The compute component also contains a load balancer distributing jobs between the available instances. In order for the application to be scalable, which is a big part of the incentive for creating an Azure application, the instances has to be stateless since there is no mechanism to ensure a client is handled by the same instance over many requests. Client specific information has to be stored in Azure storage or by other means be made available to all of the instances.

The storage component offers three kinds of storage: blob, table and queue. Blobs are held in containers, which can make up a hierarchy. They can store large data objects, up to a terabyte, and contain metadata about the data object. To optimize transmission of these potentially large blobs, they can be divided into blocks allowing retransmission of separate blocks in case of a failure. For data that require a more structured storage tables can be used. These are not relational tables, and consequently SQL is not used to access the data. Instead, the data is accessed through a REST API and the data objects, or entities, ha properties of types as int, string, and bool.

The use of a NoSQL storage model like this enables the data storage to scale and the data to be partitioned over several servers. The third part of storage, queue, is not intended for persistent storage in the same sense as blob and table. The queue structure provides a natural way for different roles to communicate.

A Web role that receives a request demanding heavy computations can hand it off to a worker role through a queue and the worker role can then hand it back through a different queue when the work is complete.

The data stored in Azure storage is replicated three times to mitigate any data loss.

The CDN, or content delivery network, is aimed at improving performance for data that is accessed frequently from places all around the world. By storing local copies at different sites the access speed can be improved.

The connect component enables an Azure application to connect to another computer at the IP level rather than HTTP, HTTPS, and TCP which instead are the canonical protocols for connecting to machines outside of the cloud. This can be useful if an application has to connect to a database on a local machine running SQL Server. This solution requires software on the local machine but enables the cloud application to connect as if the two machines were on the same IP network. The connect component can also be used to join a cloud application to a local Active Directory to enable single sign-on and use the Active Directory accounts for access control.

Criterion	Evaluation
Functional Requirements	**OK**: It guarantees 99.9% uptime and 1.6 GHz CPU, 1.75 GB RAM, 225 GB Instance Storage, Moderate I/O Performance
Software Requirements	**KO** OS: - Windows Server 2003 and 2008 only Language and Tools: **Not enough information available**
Elasticity Model	**OK** Microsoft datacenters (U.S., Asia, and Europe) Free critical data privacy Backup storage and File hosting service with charges
As-a-Service Model	PaaS – Windows Azure Managed Library (.NET), Windows Azure Native Library (native C), Windows Azure Storage Services API (RESTful-based), and Windows Azure Service Management API (RESTful-based).

Maturity and Diffusion	**OK**: Free support available. Online form 24/7. Several Wikis, forums, FAQ, as well as whitepapers and code sample galleries.
Migration Cost Model	**KO**
Economic Cost Model	Billing computing power depends on the sizes of instances with hourly rates based on the actual or virtual hardware reserved for the instance. Base plan cost: $12 cents per hour Typical: $ 362,40 / month Peak: $ 3033,60 / month Total Cost: 2477,76$ / month

4.2.9. OPSOURCE

The OpSource Cloud is an IaaS offering. In addition to Cloud Servers, Cloud Files and Cloud Networks are also parts of the platform. Cloud Servers support Windows Server, Ubuntu, CentOS and Red Hat Enterprise Linux. Cloud Servers and Cloud Files can both be accessed through a REST-based API. The OpSource Cloud Network refers to the physical network structure of OpSource. All cloud customers receive a VLAN separating their cloud servers from other OpSource servers, more VLANs can be created for an additional fee.

Cloud Networks offer features such as VPN connection to servers, load balancing, and layer two multicast. Cloud Files allows for creation of multiple storage accounts with separate administrative passwords per customer. Each account can store up to ten terabytes of data and there is no specific file size limit. Data Stored in Cloud files is encrypted with 256 bit AES.

Criterion	Evaluation
Functional Requirements	**OK**: It guarantees 100% uptime and Pay-as-you-go plan charges you only for the resources you use.
Software Requirements	OS: **OK** - Cent OS, Cent OS 5.1, Cloud Optimised Cent OS, Linux OSs, Red Hat, Red Hat Enterprise Linux, Ubuntu Linux - Windows Server 2003 and 2008 Language and Tools: **OK** Perl, PHP, Python, SQL
Elasticity Model	**OK** U.S. and European datacenters Free: - Critical data privacy, Data protection, Failover features, Persistency, Autoscaling, Load balancing, Monitoring, Virtual private servers With charges:

	- File and Web hosting service
As-a-Service Model	IaaS – RESTful-based API.
Maturity and Diffusion	**OK**: Free 24/7 phone and forum support. A few third-party forums and FAQ Web sites are available.
Migration Cost Model	**OK**
Economic Cost Model	A finer grained model where the user can choose the exact amount of RAM, storage and number of CPU cores to be . Typical: $ 251,27 / month Peak: $ 1923,10 / month Total Cost: $ 1699,93 / month

4.2.10. RACKSPACE

The Rackspace Cloud consists of two components, Cloud Files and Cloud Servers, as the names suggests Cloud Files is a storage service and Cloud Servers IaaS.

Cloud Files has a file size limit at 5 GB and files can be uploaded through the online control panel, the Cloud Files API, or via third party software. Files are replicated to three different data center zones with separate power and network connections. Rackspace provides a content delivery network in collaboration with Akamai that is integrated with Cloud Files. The CDN has locations all over the world and works automatically with no programming required. When a user downloads a file, the file is saved at an edge server close to the user reducing delivery time for the next download in the region.

Cloud Servers support Ubuntu, Debian, Gentoo, CentOS, Fedora, Arch, Red Hat Enterprise Linux, and Windows Server. Customer-created images are not supported. The platform is based on Xen hypervisor for Linux and XenServer for Windows. Each virtual server gains access to CPU cores and cycles according to its size. If resources are available, virtual servers can use CPU burst to temporarily increase the computing power. RAID-10 is used for the storage of Cloud Servers to preserve the data in case of a host failure. Snapshots of images can be created on demand or according to a schedule and used to backup the servers.

Criterion	Evaluation
Functional Requirements	**OK**: It guarantees 100% uptime and 256 MB RAM, 10GB local storage, 10 Mbps Network Throughput.
Software Requirements	OS: **OK** - Arch 2009.02, Cent OS, Debian, Fedora, Gentoo Linux, Red Hat Enterprise Linux, Ubuntu Linux - Windows Server 2003 and 2008 Language and Tools: **OK** The cloud computing provider offers root access

	to the servers, all the programming languages are supported by the provider
Elasticity Model	**OK** U.S. data centers only Free: - Bootable mode, Critical data privacy, Data protection, Persistency, Autoscaling, Load balancing, Virtual private servers With charges: - Backup storage, Snapshot backup, File/Web hosting services
As-a-Service Model	IaaS – RESTful-based API.
Maturity and Diffusion	**KO**: 24/7 phone and chat support, but no significant active communities.
Migration Cost Model	**OK**
Economic Cost Model	Billing computing power depends on the sizes of instances with hourly rates based on the actual or virtual hardware reserved for the instance. Typical: $ 352,74 / month Peak: $ 1543,50 / month Total Cost: $ 2270,79 / month

4.2.11. OVH PUBLIC CLOUD

Founded in 1999 by Octave Klaba, OVH.com is an independent French company based in Roubaix. It's number one for hosting services in Europe and is positioned 4th in the world for domain names.

For his first try in the public cloud, OVH chose simplicity, the offer is simply called PublicCloud. For now, it includes only part called Instances. Storage part and another CDN (Content Distributed Network) will be added soon.

These instances are all just virtual machines that you can set up, there is no need to worry about traffic information or the number of inputs / outputs. The customer selects the type of virtual machine that suits him and pays on time use. The price starts from € 0.01 tax per hour for a machine with 600 MB of RAM, a CPU to 0.8 GHz and a local hard drive (not persistent) 50 GB If for some machines.

An important point is if you need to keep the data from one instance to another, persistent storage of data is optional and have to be validated when creating the instance. It will then add € 0.00006 per gigabyte of storage per hour. Finally, only the active instances are billed. Instances extinguished, even related to an offer of permanent storage, are not charged.

With this offer, OVH aim to simplify billing of cloud for businesses and individuals.

Last but not least, all data are stored in data centers of the company in Roubaix (France).

Criterion	Evaluation
Functional Requirements	**OK** but be careful ton not forget to subscribe to the persistent storage
Software Requirements	OS : **OK** (Linux based system, or Windows Server 2008) Language : **OK**
Elasticity Model	**OK** : Instances are delivered per minute, and you can add 10, 100, 1000 instances if necessary. The API and the manager include features for cloning instances.
As-a-Service Model	IaaS
Maturity and Diffusion	**OK :** it's already a leader in network hosting, OVH is present all over the world
Migration Cost Model	**OK:** You can use your own image drive for your instances
Economic Cost Model	It's a pay-on-time model. Pricing list is very much changing at the moment and we could not simulate precisely a pricing corresponding to our needs. Commercial documentation of OVH though, states that their price are "very competitive" with regards to the market standards.

4.2.12. Cloud Providers Comparison Matrix

Table 4 summarizes the result of our survey on cloud computing providers. We observe that only a few of the providers considered fulfil the requirements of the InGeoCLOUDS infrastructure: Amazon, SigmaCloud, OpSource, and OVH Public Cloud. We also notice a large variation in the monthly cost of the service, ranging from about €600.00 for Amazon, to about €1300.00 for the OpSource platform. This result was somehow expected, since the large adoption of the Amazon platform allows for reduced costs. For the same reason, Amazon undoubtedly provides the most reliable and well-documented cloud computing platform.

The estimated cost of the Amazon platform is €624.58. This estimate is based on the information supplied by the data providers, and it is very heterogeneous, sometimes incomplete or a rough estimate itself. We believe this is actually an over-estimate, also because we considered a very demanding peak usage. As a comparison, the in-premises cost estimated by EPPO for a single application similar to an InGeoCLOUDS use case is €333.33 per month, while according to BRGM the estimate cost is of €920.60 for the ground-water, geo-hazards and geology platform and €3141.66 for the large Carmen infrastructure. This suggests that a cloud computing solution become more and more convenient as the amount of data and number of geo-services increases.

Of course, we cannot draw conclusive statements on the basis of the above estimates. After the deployment of the first pilot of InGeoCLOUDS, we will be able to monitor the actual requirements of the architecture and, thus to provide a much more accurate and realistic estimate of the cost of the various platforms. Also consider that evaluating the personnel cost for each provider in the InGeoCLOUDS cloud computing scenario is very difficult.

For the duration of the project, we will continue to monitor these and every other cloud providers with the goal of supporting the InGeoCLOUDS platform with the smallest possible cost. We will pay special attention also to potential platform stemming from the efforts of other European projects.

Table 4: Cloud Providers Comparison Matrix

	Functional Requirements	Software Requirements	Elasticity Model	As-a-Service Model	Maturity and Diffusion	Migration Cost	Economic Cost (Monthly)
Amazon	OK	OK	OK	IaaS/PaaS	OK	OK	€ 624.58
SigmaCloud	OK	OK	OK	IaaS	OK	OK	€ 1144.80
Atlantic.Net	KO	KO	KO	IaaS	OK	OK	€ 1897.02
Flexiant Flexiscale	OK	OK	KO	PaaS	KO	OK	€ 6909.11
GoGrid	OK	OK	KO	IaaS	KO	KO	€ 1798.32
Google App Engine	OK	KO	OK	PaaS	OK	OK	NA
Joyent	OK	OK	OK	IaaS	KO	OK	€ 2562.83
Microsoft Azure	OK	KO	OK	PaaS	OK	KO	€ 2477.76
OpSource	OK	OK	OK	IaaS	OK	OK	€ 1356.36
Rackspace	OK	OK	OK	IaaS	KO	OK	€ 1811.84
OVH Public Cloud	OK	OK	OK	IaaS	OK	OK	NA

5. CONCLUSIONS

In this document we first surveyed the existing tools, data and applications at the data providers side. The outcome of this analysis is twofold.

First, we identified the common software packages and services being used my most of geodata applications. Some of those services, e.g. the DBMS, will be casted into a cloud-based environment to achieve elastic scalability to large data sizes and large users requests volumes that cannot be achieved with the current in-premises infrastructures owned by the different data providers. As a result, we proposed a logical architecture for InGeoCLOUDS, that will be further improved with the definition of the InGeoCLOUDS cloud architecture for Pilot1, to be documented in D3.2 (planned for M12).

Second, we produced a preliminary resource requirements estimate, which was necessary for estimating the cost of a given Cloud Platform. After the deployment of Pilot1, we will be able to improve the accuracy of our estimate by taking into consideration the actual usage of the cloud platform. In fact, we reviewed several Cloud Service Providers by taking into account a number of criteria including migration costs, compliance with the InGeoCLOUDS infrastructure, etc. Given the information at hand, the Amazon platform is the one that best suits the requirement of InGeoCLOUDS and its Pilot 1 and appears to be also the cheapest one. The facilities for hosting data in an European data centre (Ireland) is also appreciated. Amazon services are also by far the best documented and best tooled solution.

At the same time, the consortium has a strong position of not being bound to a particular CSP. Technical choices and decisions during the implementation tasks will also be taken in the light of a continuous assessment on the reversibility and level of integration of CSP-specific tools and languages (scripting for example). As repeated above, other CSPs are not definitely excluded and pure European commercial actors such as OVH, SigmaCloud, as well as infrastructures designed by other EC projects will further be contemplated in Task 3.5 and may be also used for test purposes in parallel to the main InGeoCloudS infrastructure settlement.

6. REFERENCES

[1] F. Bugiotti, F. Goasdoué, Z. Kaoudi, and I. Manolescu. RDF Data Management in the Amazon Cloud. In Proceedings of DanaC 2012. March 30, 2012, Berlin, Germany

[2] Dydra. http://dydra.com/.

[3] Google. BigQuery. https://developers.google.com/bigquery/

[4] R. Stein and V. Zacharias. RDF On Cloud Number Nine. In 4th Workshop on New Forms of Reasoning for the Semantic Web: Scalable and Dynamic, May 2010.

[5] Y. Tanimura, A. Matono, S. Lynden, and I. Kojima. Extensions to the Pig data processing platform for scalable RDF data processing using Hadoop. In Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on, pages 251 –256.

[6] M. F. Husain, L. Khan, M. Kantarcioglu, and B. Thuraisingham. Data Intensive Query Processing for Large RDF Graphs Using Cloud Computing Tools. In 3rd International Conference on Cloud Computing, 2010.

[7] G. Ladwig and A. Harth. CumulusRDF: Linked Data Management on Nested Key-Value Stores. In SSWS, 2011.

[8] P. Mika and G. Tummarello. Web Semantics in the Clouds. IEEE Intelligent Systems, 23(5):82–87, 2008.

[9] J. Myung, J. Yeon, and S.-g. Lee. SPARQL Basic Graph Pattern Processing with Iterative MapReduce. In Workshop on Massive Data Analytics on the Cloud, 2010.

[10] A. Schatzle, M. Przyjaciel-Zablocki, and G. Lausen. PigSPARQL: Mapping SPARQL to Pig Latin. In SWIM, 2011.

**** End of the document ****