The SDSS System

Overview

This chapter gives an overview of the software architecture of the RiskChanges SDSS system. One of the objectives within the project is the development of a SDSS system for probabilistic multi-hazard risk assessment taking into account changes in hazard scenarios related to climate change and exposed elements at risk. The aim of this section is to provide an overview of the design of the SDSS.

Regarding the quality attributes, the architecture of the system must be:

- Modular
- The various parts of the system must be loosely coupled
- Vendor independent
- Extensible
- Using standards for interoperability
- Flexible
- Web-based

Enterprise applications are usually designed using a Three-Tier Client-Server Architecture (see figure below). This architecture provides three layers in which each layer deals with a different level of responsibilities. The three-tier architecture is used in the web-based SDSS. The top tier or presentation layer constitutes the user interface, the middle tier is the core of the system for business logic and the bottom tier handles the data storage. One of the advantages of this architecture is easier to make changes in the layers without influencing the other layers.

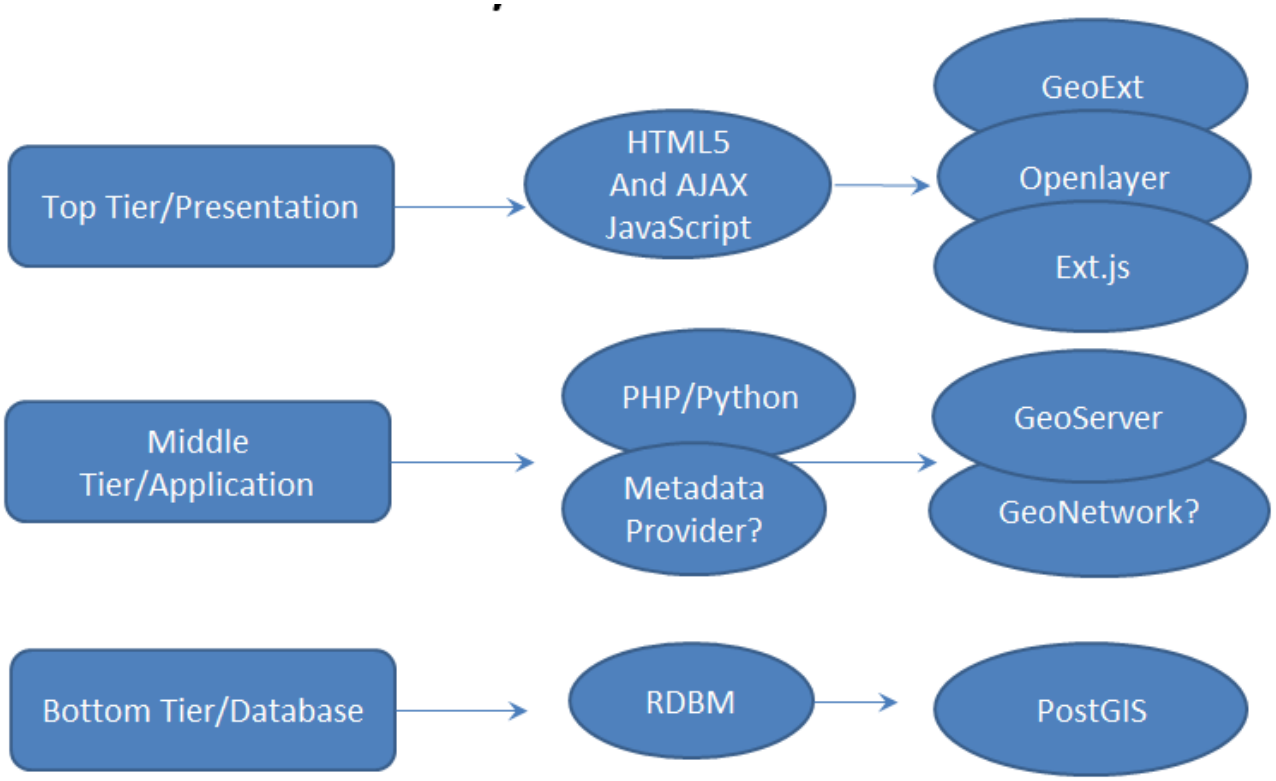


Figure 1 the three-tier architecture of the SDSS

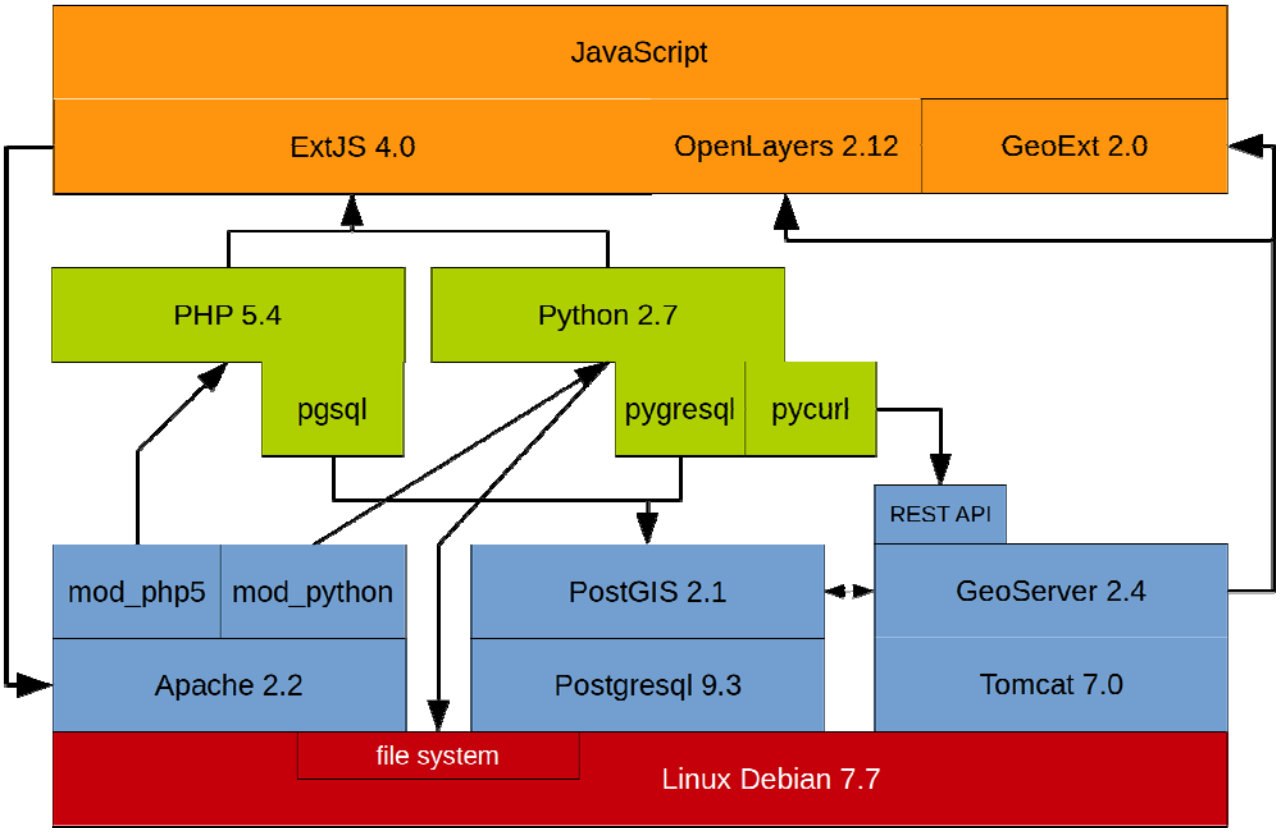
Furthermore, the SDSS is developed using open source and geospatial technologies. Open source software is freely available for public use and users are free to modify and customize the software. For instance, OSGeo (Open Source Geospatial foundation, http://www.osgeo.org/), which dates back to 2005, is a nonprofit foundation that supports development of the open source geospatial software.

As it is now, the SDSS is an open source platform that comprises of a Geoserver, PostGIS and Javascript client side (OpenLayers and GeoExt based on ExtJS). The SDSS employs HTML, JavaScript and CSS on the client side with a combination of Python/PHP scripting language and a PostGIS database on the server side. All the maps and all the data are stored in the PostGIS database. Analytical components for risk are implemented using SQL in the spatial database and decision analysis calculations are carried out in JavaScript on the client side. Finally, all the results are stored into the database.

Server-side components

The whole server-side of the SDSS is running on a single machine running the Linux operating system. For an overview of the used software components see the figure below. The Linux operating system is indicated in red. All the servers, the back-end data-tier and the middle-tier run on Linux. Linux was chosen as the operating system because it's easy to manage and maintain and because we already have

a lot of experience using Linux. In the figure, servers are indicated in blue, middle-ware is indicated in green. All these components together constitute the server-side of the SDSS. The user-interface, implemented using JavaScript technology, indicated in orange in the figure, constitutes the client-side of the SDSS.



OpenGIS

Figure 2 the software components of the SDSS

The Open Geospatial Consortium (OGC, http://www.opengeospatial.org/) is an international industry consortium of almost 500 companies, government agencies and universities participating in a consensus process to develop publicly available interface standards. OGC Standards support interoperable solutions that "geo-enable" the Web, wireless and location-based services and mainstream IT. The standards empower technology developers to make complex spatial information and services accessible and useful with all kinds of applications.

Some relevant standards are the Geography Markup Language (GML), Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS) and Web Processing Service (WPS).

WMS

The Web Map Service Interface Standard (WMS) provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc) that can be displayed in a browser application.

WFS

The Web Feature Service Interface Standard (WFS) defines an interface for specifying requests for retrieving geographic features across the Web using platform-independent calls. The WFS standard defines interfaces and operations for data access and manipulation on a set of geographic features.

The specified feature encoding for input and output is the Geography Markup Language (GML) although other encodings may be used.

Linux

Linux is a Unix-like and mostly POSIX-compliant computer operating system assembled under the model of free and open-source software development and distribution. The defining component of Linux is the Linux kernel, an operating system kernel which was first released on 5 October 1991 by Linus Torvalds.

Typically, Linux is packaged in a form known as Linux distribution, for both desktop and server use. Some popular mainstream Linux distributions include Debian, Ubuntu, Linux Mint, Fedora, openSUSE, Arch Linux, and the commercial Red Hat Enterprise Linux and SUSE Linux Enterprise Server. Linux distributions include the Linux kernel, supporting utilities and libraries and usually a large amount of application software to fulfill the distribution's intended use.

Due to the flexibility and the free and open-source nature of Linux, it becomes possible to highly tune Linux for a specific purpose. The distributions often used for this purpose include Debian, Ubuntu (which is itself based on Debian). The Debian Stable distribution is one of the most popular for personal computers and network servers, and has been used as a base for several other Linux distributions.

For the SDSS, the Debian distribution was selected. Debian is the most stable and popular non-commercial Linux distribution. Software is easy to install and upgrade, because Debian has the best packaging system in the world. It is fast and easy on memory. It has good system security. And remote

maintenance is easy.

The original SDSS ran on a quad core CPU running at 2.40GHz with a approximate speed of 6600 bogomips, a total memory of 8 GB and a two 1 TB disks configured as one Linux software RAID of 1 TB. At the time of writing the version of the operating system is Debian 7.7, which was released on 18 October 2014.

Communication with the Linux machine can be done using the SSL protocols, which are encrypted, secure protocols over the Internet. Of course, to be able to use these commands you have to know what you are doing.

The following command can be used to make a SSH (secure shell) login on the linux machine. Replace the user name by your own name. Please note that the SSH server does not listen to the default port 22 but is moved to the port 2222 where most hackers would not expect it to be.

```
$ ssh -p 2222 user@changes.itc.utwente.nl
```

When prompted for a password use your own password. This is the command you would use on another linux machine. On windows you can use, for instance, putty.exe which is a little client that can be used to talk to an SSH server.

Files can be transported to the Linux machine using the SFTP protocol which basically runs the FTP protocol on top of the SSH protocol for maing secure FTP connections and transfers. To make SFTP transfer use the following command. Replace the user name with your own account.

```
$ sftp -P 2222 user@changes.itc.utwente.nl
```

When prompted for a password enter your own password. Alternatively, you can also use GUI tools that are aware of the SFTP protocol. A nice program to use is FileZilla (filezilla-project.org), which runs on different platforms like Linux and Windows. Within FileZilla you should indicate the protocol to use together with the hostname. (host: sftp://changes.itc.utwente.nl)

In addition, the scp (secure copy) command can be used for file transfer. For example:

```
$ scp -P 2222 some.txt user@changes.itc.utwente.nl:/home/user
```

After prompting for the password, this command will copy a local file called some.txt into your home directory of the changes server. Please note the captital 'P' used for the port number, this is different from the ssh command. For other examples check the manual pages (man scp).

PostGIS is an open source software program that adds support for geographic objects to the PostgreSQL object-relational database. PostGIS follows the Simple Features for SQL specification from the Open Geospatial Consortium (OGC).

Maintenance on the PostGIS database on the SDSS system can be done using different tools. One tool that can be used everywhere is the web-interface through phppgadmin. The web-interface can be reached via the following url: http://changes.itc.utwente.nl/phppgadmin/

Alternatively, you can install pgadmin (www.pgadmin.org) and use the following credentials to make a

connection to the database.
```
hostname: changes.itc.utwente.nl
port: 5432
database: changes
```

In computing, the GeoServer, which is an open-source server written in Java, allows users to share, process and edit geospatial data. Designed for interoperability, it publishes data from any major spatial data source using open standards. The Geoserver of the SDSS runs on top of the Tomcat web server. It can be accessed via the following url:
`http://changes.itc.utwente.nl:8080/geoserver`
The Geoserver is used for publishing spatial data via WMS. Part of the JavaScript user-interface Uses this to present maps in the SDSS.

GIT

The software versions of SDSS system are maintained on a Git server on the Linux machine.

The Git server can be reached through the SSH protocol (on port 2222) on the following url:
`ssh://changes.itc.utwente.nl:2222/home/git/changes.git`
Git will ask for your username and password.

For instance, to get a copy of the most up to date code, retrieve the SDSS Master branch only. The following command can be used on a Linux box:
```
$ git clone -b SDSS-Master --single-branch
ssh://bakker@changes.itc.utwente.nl:2222/home/git/changes.git
```

The Web Servers, Apache2 and Tomcat7

The Apache HTTP Server, normally called just Apache, is the world's most widely-used Web server software.

Apache Tomcat, or simply Tomcat, is an open source web server and servlet container. Tomcat implements several Java specifications including Java Servlet, JavaServer Pages (JSP), Java EL, and WebSocket, and provides a "pure Java" HTTP web server environment for Java code to run in.

Scripting, Python and PHP

Python is a widely used general-purpose, high-level programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale.

Python can serve as a scripting language for web applications. The Python scripts on the SDSS run via mod_python on the Apache web server. Python scripts use the Python module pg to talk to the database. The required GIS processing for the SDSS is all done on the PostGIS spatial database. Python scripts construct SQL statements for this functionality.

Geoserver

GeoServer is an open-source server written in Java, which allows users to share, process and edit geospatial data. It is designed for interoperability, and it publishes data from any major spatial data source using open standards. In the SDSS we use it for publishing the data so that it can be visualized in the user-interface using JavaScript extension libraries such as OpenLayers and GeoExt.

In the SDSS the data is published using the WMS protocol. GeoServer uses Restlet as a framework for the REST services it provides. We use the GeoServer's REST API for configuring it via Python scripts. The configuration runs via the Python package PyCurl, which itself is based on the cURL library.

Spatial Database

A spatial database or geodatabase is a database that is optimized to store and query data that represents objects defined in a geometric space. Most spatial databases allow representing simple geometric objects such as points, lines and polygons. Some spatial databases can handle more complex structures such as 3D objects, topological coverages, linear networks, and TINs. While typical databases are designed to manage various numeric and character types of data, additional functionality needs to be added for databases to store and process spatial data types efficiently. These types are typically called geometry or feature. The Open Geospatial Consortium (OGC) created the Simple Features specification and sets standards for adding spatial data structures and spatial functionality on top of database management systems.

The spatial module on top of a normal DBMS basically turns it into a GIS system. The SDSS uses the PostGIS spatial extension on top of the Postgresql DBMS. All the vector and raster data is stored and processed using PostGIS. The data is organized per study area; every study area is represented as a spatial database inside PostGIS. All the spatial processing on the data is carried out inside the database.

The JavaScript user-interface of the SDSS typically calls a Python script that gathers all information and constructs an SQL statement which is thrown at the database which then carries out the necessary calculations. Resulting spatial layers are then posted to the GeoServer so the user-interface can visualize the results. Because the GeoServer currently cannot obtain raster data from the PostGIS database the SDSS also keeps a copy of GeoTIFFs on file. Vectors are visualized via the GeoServer from PostGIS. But rasters data are visualized via the GeoServer from the Linux file system.

Spatial SQL

A spatial DBMS adds additional functionality to the standard SQL for handling and processing spatial objects (or features). The statement below gives a (fictional) example of how this works. The following spatial SQL solves the following question: given a certain polygon, return a list of cities that are contained within it. Example:

```
SELECT *
FROM cities
WHERE ST_Within(points, ST_GeomFromText('POLYGON()...)', 28992)
```

The 'cities' is the table that contains the records of all the cities. This table has a special column that contains the spatial information (the geometry) of the city. In this case cities are represented as points. The function ST_GeomFromText constructs a spatial object that represents a polygon.

The number 28992 is the spatial reference system identifier (SRID) of the polygon, in this case the Dutch “Amersfoort New” grid. The SRID must be the same as the reference system of the cities data. The ST_Within function looks at the spatial relationship between the polygon and a city record. If the city is contained within the polygon then the result of ST_Within will be True, and False otherwise.

In this way the select-from-where statement only selects those cities that are contained within the constructed polygon. The SDSS uses far more complex spatial SQL statements for calculating loss maps from the input hazard maps, elements at risk maps and vulnerability curves.

Client-side components

Because the decision was made to make the SDSS web-based, the client-side of the SDSS, indicated in orange in the figure above, was implemented using JavaScript technology. JavaScript is a computer programming language most commonly used as part of web browsers, where it is used for making user-interfaces. The possibilities that JavaScript offers go far beyond the regular HTML documents. On the web, JavaScript is used for making dynamic pages. In the SDSS, JavaScript is used for constructing the user-interface, carrying calculations, communicating with the server, visualizing the results. In the SDSS the JavaScript extension library ExtJS is used for managing and handling the user-interface elements, such as combo boxes, tree controls, tab panels, and other controls or widgets.

The SDSS uses version 4 of ExtJS that provides a model-view-controller (MVC) style of code organization. MVC is a software architectural pattern for implementing user interfaces. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user.

In the SDSS the JavaScript extension libraries OpenLayers and GeoExt are used for visualizing spatial data. Both libraries are basically used for building client-side web mapping and webGIS applications.