# PA International Web presence

PA International produces Geo-location software and data logging units (*devices*). The department for data logging and Geo-location is fully developed, but the department for data presentation and unit monitoring is falling behind.  
  
The PA International has requested a project with a new website and a web-service.

The web-service should be used by the devices and the website, as well as anyone who wishes to write a custom client or (custom clients are naturally not a part of the project).

The system should be able to handle a large number of simultaneous devices (several thousand), as well as a large number of simultaneous web page and service users (several hundred).



## Devices

The devices upload status and readings to the web service.

The devices receive commands and configuration from the web service.

## Web-service

The web-service must ensure that only authorized access to data is permitted. The performance penalty of using HTTPS (SSL/TSL) as opposed to basic HTTP authentication should be considered, yet the actual implementation may be done without authentication or confidentiality concerns.

The web-service has three distinct types of requests:

1. Device reading or status requests.
2. Device configuration and command requests
3. Service administration requests



### Type 1 (Device readings and status)

The web service should maintain three pieces of information about a device:

1. Their location and previous locations,
2. The devices status and previous status.
3. The device readings.

and must expose this information to interested parties (web site, custom clients, etc.)

This device information is exchanged in the XML language PAGeoLoggingML, which informally can be described as:

* The root element is *geoLogCollection*, and contains attribute *deviceID*, attribute *timestamp*, and zero or more geoLog elements
* *deviceID* uniquely identify the device and is of type URI.
* *timestamp* is of type DateTime and indicate the time of the reading or status update.
* The *geoLog* element can contains element *location*, *status*, *readings* and *zone*
* *location* is of type KML (Keyhole Markup Language) and indicates the device’s location at time *timestamp*.
* *status* contains assorted status information for the device. Exact content will be defined later. Only present if this is a status update.
* *readings* contains a list of key-value-type pairs where the type is either a known MIME-type or defined in the Request header. Only present if this is a new reading.
* *zone* contains information about the zone that this device belongs to, if any. This information is generated by the web-service, and is not part of the device <-> web-service exchange.

An XML Schema formalization of this language must be created as part of the web-service API.

### Type 2 (Device commands and configuration)

Will be defined as needed.

### Type 3 (Service configuration)

The web-service should furthermore allow for the administration of devices as follow:

1. Which user/owner a device is associated with
2. Which zone a device is in, if any (only used for manual override).
3. Definition of zones as a collection of polygon points.

Write actual XML language setup.

The API for the web-service should be REST-like. E.g.

* A GET-request to /geolog/devices gives a list of all devices registered to the user
* A POST-request to /geolog/devices with a device ID and possible a timespan returns a collection of device data for that device.
* A detailed specification will be part of the web project

## Web site

The website should be written as Java servlets and JSP, with use of JavaScript (JQuery) and AJAX for client side performance.

There are two types of devices; Geo-stationary and dynamic. A Geo-stationary device has a fixed location and are generally lacking a GPS receiver (is not aware of its own location). A dynamic device is mobile and is always equipped with a GPS receiver. All devices communicate with the server. Each device has a unique ID (GUID, MACID, ...). Each device has a location, indicated by a GPS coordinate (Latitude, Longitude). The geographical datum is WGS84 (Earth). A device may measure one or several parameters each measurement having a time stamp. A device measurement is generalized into the concept of a *reading*.

## Server

The server refers to one or more physical servers equipped with load-balancing software and routers (bandwidth and load-balancing is not a part of this project, it is considered that sufficient bandwidth is available). Some considerations about the requirements for the server may be included for completeness.  
  
The devices communicate with the server, and the server stores the data received from the devices in a database and exposes the device data via a website and a web-service.  
  
The server furthermore forwards any commands or configuration changes to the devices (these commands or configuration changes are entered via the website or web-service).  
  
The server groups devices based on owner and administrator. This means that a given login has access to a subset of the total number of devices.

## Website

All website access must be authenticated. The performance penalty of using HTTPS (SSL/TSL) as opposed to basic HTTP authentication should be considered, yet the actual implementation may be done without authentication or confidentiality concerns.   
The website should be written as Java servlets and JSP, with use of JavaScript (JQuery) and AJAX for client side performance.

### Users

The website must be able to display information about the devices; their location (google maps), their status (operational, error, …), their history (status log), their data (graphical (Javascript) or tables (XSLT)), etc. The website must have an application look-and-feel (i.e. perceived performance for the client is an issue), and will therefore most likely need to rely on Javascript and AJAX. It is acceptable to have these technologies as a requirement for the client browsers (no mobile browser support). If mobile devices is to be supported it will be a new project. The user may also send commands to the devices (e.g. take a reading now) or update the device configuration (e.g. the frequency of readings) via the website.

### Administrators

The administrator website is used for administrating devices (registration of device/owner relations, device reconfiguration, ...) and creating new zones (see google maps).

## Google maps

A map must be used to display the devices, their position, status and last reading (where applicable). The map must be fluently zoom-able. Furthermore, for dynamic devices, Google maps must be used to track the movements of a device over a period of time. On-line updates of device status and location must be supported (AJAX).  
  
Zones may be created indicating a geographical area on Google maps (mathematically a polygon), which allows the aggregation of the status of all devices in a zone. The color property of the zone may then be used to display the aggregated status, e.g. translucent green signals no errors and translucent red signals that one or more devices has reported an error.  
  
Zones may be nested so a zone can contain other zones.  
If a device has a location within a given zone it belongs to this zone. Static devices generally belong to one zone, where dynamic devices can jump from zone to zone.  
A device’s zone relationship may be manually overridden (special circumstances may dictate that a device should not be part of any zone, or should be part of a specific zone regardless of its geographical location), which disables the location to zone relationship and simply sets this device to be part of the given zone no matter what location it has.  
  
Restrictions may be imposed to simplify the design, e.g. no overlapping zones, no mixed content (either a zone consists of devices or other zones, not both), limited number of zones or devices are permitted in a single zone (a maximum of N devices or zones are allowed in a zone. If this number is exceeded the system must create a new zone by intelligent splitting), etc.

## Database

The persistence layer is a GIS-aware relational database (PostgreSQL with PostGIS, Oracle 11g, Microsoft SQL Server 2008, ...) with a data access component. The database is considered installed, configured and maintained and is not a part of the project. Some considerations about the requirements for the database may be included for completeness.

## Web-service

A web-service shall be developed to support the required web clients.  
The web-service must expose the necessary functionality for the website to function, and also allow for client applications to be developed for easing the generation of zones and administrating the devices, etc.

An XML language for information exchange and an XML schema to validate these must be developed or adopted. Candidates for adoption are the Google [KML](http://en.wikipedia.org/wiki/Keyhole_Markup_Language) XML language or the OpenGIS [GML](http://en.wikipedia.org/wiki/Geography_Markup_Language) XML language in order to use current industry standards on geographic information.  
  
Technologies  
Relevant technologies (a subset of these will probably suffice):  
  
Java Servlets, JSP, XML, XML Schema, JDOM, XSLT, JavaScript, HTML, CSS, REST, JQuery, Java

## System design

A first iteration towards a graphical representation of the system is shown in the figure below