Research Gateway Management System for ARSC Projects

Vincent Castro, Oralee Nudson and Don Bahls

Arctic Region Supercomputing Center - University of Alaska Fairbanks

vcastro5@alaska.edu

Motivation

The Research Gateway Management System (RGMS) is an account management system written in Python[1] geared towards creating and facilitating user authentication and management for Arctic Region Supercomputer Center (ARSC) users on the ARSC systems. The intent of the RGMS is to forge an easy to use interface for clients to manage project information and for system administrators to manage projects and users. Utilizing this account management system will catalyze a transition towards the simplification of user and project requests and creation procedures as well as streamline user access to ARSC systems.

Method

To begin the system development process, we first developed a criteria for identifying which tools to use for the RGMS implementation.The criteria consisted of being able to handle user or server requests efficiently, be invariable to various system environment changes or crashes, provide security to protect users and user sessions and to be able to provide a simplistic interface for user interaction. We found that the toolkit combination of the Python based web framework Django[2], Apache 2[3] hyper-text transfer protocol (HTTP) server and MySQL[4] relational database management system (RDMS) best fit this criteria.

Each tool fits a certain role in the management system which is broken down into flow of data from a user or an admin. Initially, when the user visits the RGMS website, they are connecting to the Apache 2 HTTP server which handles data transfer to and from the user and admin. The background functionality of the website is handled by the Django web interface and the RGMS. When a user makes a request for editing or creating a project the request goes through Django and the RGMS. Django is the RGMS' way to interface with the Apache 2 HTTP server. After the data is received through the Django web interface, the RGMS handles the parsing and management of the data. Depending on the request, the data is scheduled by the RGMS to be processed at the time of the request or addressed at a later time due to pending approvals. All requests which cannot be acted on at the time of the request are added to the process queue on the MySQL database for which Django also provides an interface. All requests that can be acted on at the time of the requests are automatically changed on the MySQL database by the RGMS. After requests are handled by the RGMS, it creates and sends results data back to the user (see figure 1).

Utilizing these tools enabled us to begin RGMS development by first creating a simplistic website where the user would begin a session by logging in, be shown some information about themselves, then allow the user to logout of the created session. To create this simple page we were able to employ the Lightweight Directory Access Protocol[5] (LDAP) provided by the University of Alaska (UA) Enterprise Directory (EDIR). The Python backend was then designed to use the Python LDAP module for user login, securely capture information about the user, and to insert the user information onto the MySQL database. This was achieved by the user providing their UAF or ARSC username and password over a Secure Sockets Layer[6] (SSL) connection. The Python backend uses this information to connect and create a secure gateway via a bind with the LDAP server. After this is successfully done, the user information on the LDAP server is requested and the resulting information is sent through the Django gateway by the RGMS and displayed to the user on the website (see figure 2).

Future Development

Development on the RGMS is progressing at this time. Currently, efforts are focused on the initial management and scheduling of requests and initial layout of database tables. In the near future, we plan for this system to manage all ARSC user information and data needs along with providing a secure and simple gateway to research on the ARSC systems.

The poster was created with LaTeX using the beamerposter plug-in. Original poster theme provided by http://bit.ly/1QFT9A

Figure 1: Information Transfer within the RGMS

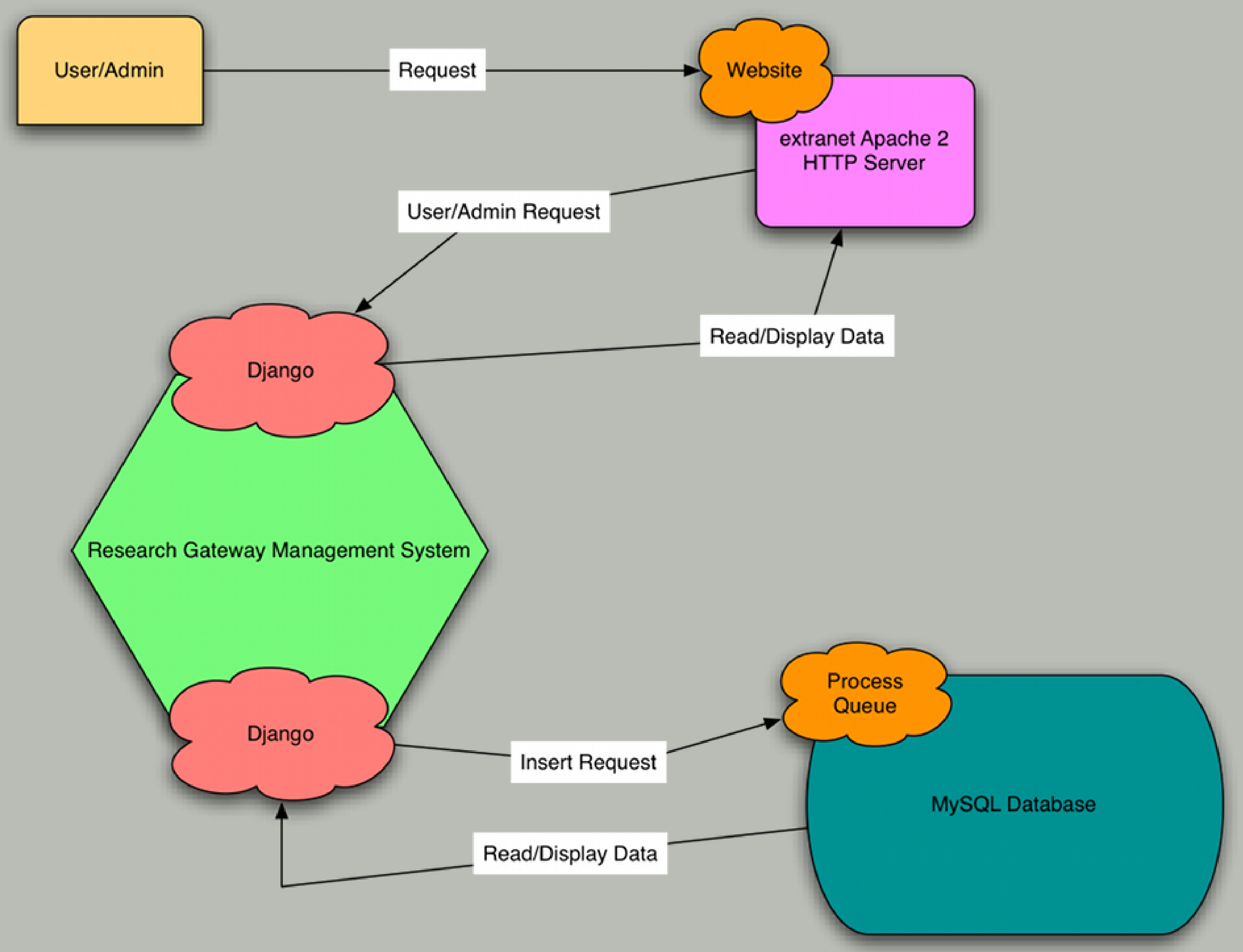
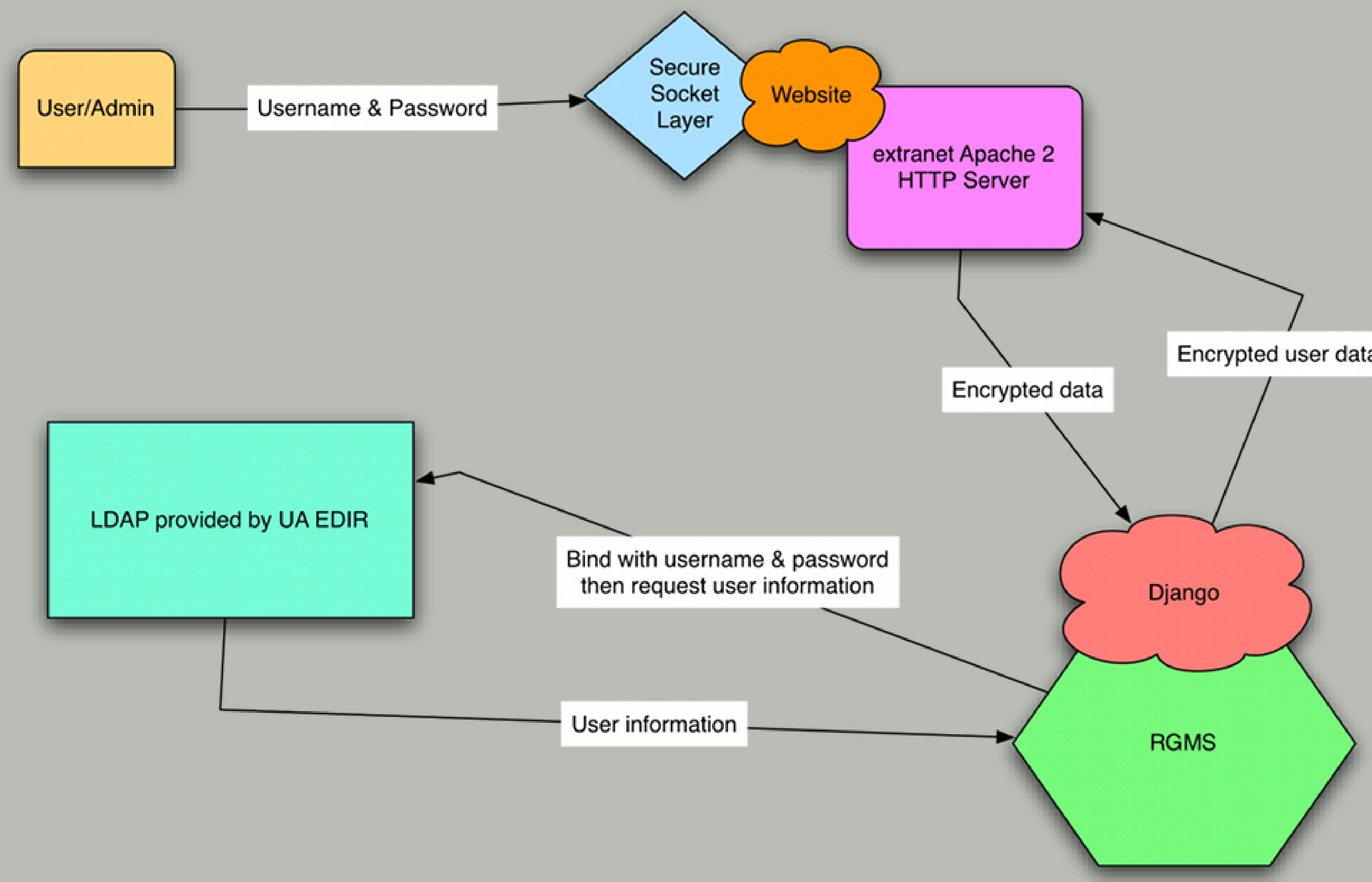


Figure 2: LDAP Server Compounding with the RGMS



References

1. "Python Programming Language" 2011. Python Software Foundation. http://www.python.org/
2. "The Django Web framework" 2012. Django Software Foundation. https://www.djangoproject.com/
3. "The Apache HTTP Server Project" 2012. Apache Software Foundation. http://httpd.apache.org/
4. "MySQL" 2012. Oracle Corporation. http://www.mysql.com/
5. "Lightweight Directory Access Protocol" 2006. The Internet Society. http://tools.ietf.org/html/rfc4510
6. "The Secure Sockets Layer (SSL) Protocol" 2011. Internet Engineering Task Force. http://tools.ietf.org/html/rfc6101