

CS461 Technology Review

Tools to Support Community Growth in Open Source Projects

Group 09

Justin Bruntmyer

Megan Goossens

Hai Nguyen

Client: Ross Gardler

November 11, 2015

1. Goal of Project

The goal of this project is to build connections, collaborations, and identifications between community members and leaders in open source projects. This entails contributing to Apache Software Foundation's existing software platform to create tools that identify and support community leaders and members. Better tools will help identify potential community leaders which will then allow community builders to engage with and support those leaders. Over the course of the project, consistent updates check what tools were used as intended and which ones could be improved and expanded. Evaluation of the tools implemented is done in order to measure its significance and effectiveness.

2. Components and Technologies

For this project we have broken down the technologies we could use into three different pieces which come together as part of the entire system: Web Application Framework, Backend Data Storage, User Interface, and User and Group Management System. Below you will find research done for the individual technologies available for these pieces along with a selection our team is using for the main system.

3. Web Application Framework

3.1 Introduction

As our project involves a website, we will need a web application framework to build it on. A web application framework is software that is designed to help with the development of websites, web applications, and more. A framework aims to help alleviate the overhead that comes with web development, with libraries that help with tasks such as database access and templating web pages. Possible frameworks are as follows:

- Django
- AngularJS
- Ruby on Rails

3.2 Django

Django is a free, open source, high-level Python web application framework which follows the model-view-controller (MVC) architectural pattern. Django aims to do most of the basic development for you. It emphasizes reusability and pluggability of components, rapid developments, and the "don't repeat yourself" principle. It provides an optional dynamic admin interface, which is controlled in a similar way as other portions of the framework, by models. Django also has a database migration system to make migrations easier, as well as a testing framework to make testing your web app easier. It can be used for both front-end and back-end applications.

3.3 AngularJS

AngularJS is a free, open source, JavaScript web application framework designed to help with developing single-page applications. It aims to simplify both the development and the testing of such applications by providing a framework for client-side MVC and model-view-viewmodel (MVVM) architectures, along with components commonly used in rich Internet applications. The AngularJS library works by first reading the HTML page, which has embedded into it additional custom tag attributes. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

3.4 Ruby On Rails

Ruby on Rails is a web application framework written in Ruby. Rails is an MVC framework, providing default structures for a database, a web service, and web pages. It encourages and facilitates the use of web standards such as JSON or XML for data transfer, and HTML, CSS and JavaScript for display and user interfacing. In addition to MVC, Rails emphasizes the use of other well-known software engineering patterns and paradigms, including convention over configuration (CoC), don't repeat yourself (DRY), and the active record pattern.

3.5 Selection

Our project already has a prototype, which uses Django. We will continue to use Django, because it is both already implemented, and a good, solid, easy to use framework that suits our needs. With the models, views, templates, administration, and database management, it will make our task much easier. Our other options require quite a bit more work

to do the same things in another framework, and even more work for us re-implementing all the features that are currently there.

4. Backend – Data Storage Piece

4.1 Introduction

For the backend section of our design project, we will need to store large sets of data for things such as profiles, people, groups, community leaders, events, and information of those events. These items require a data storage component that is already implemented into the prototype of the website. These data need to be accessed and stored quickly in order for data to be periodically transferred fast enough to handle consistent updates of events. Depending on the priority of the objective, relatively pertaining to process speeds and data storage types, requires different types of possible technologies that could be used. For our priority, we assume that we will be concerned most about storage capacity in managing events and new community leaders that start a new project. Individual project groups are also projected to contain a large amount of space. Possible technologies for our data storage component are listed below.

- Apache Derby
- MySQL
- PostgreSQL

4.2 Apache Derby

Apache Derby is a database management system developed by Apache. The engine is a full-functioned relational embedded database-engine that supports JDBC and SQL for programming APIs, with IBM DB2 SQL syntax. Experience with Derby indicates small liabilities in areas other than test environment. One issue has to deal with interrupted I/Os cause Derby to fail outright on Solaris. Building a shim becomes necessary to protect it from those failures. Apache Derby tends to have low performance for complicated queries and low performance on large datasets. Apache Derby as a result becomes more of a testing tool for testing environments and performance testing.

4.3 MySQL

MySQL is the most popular and commonly used relational database management system. MySQL can handle a lot of data and can work efficiently and “cuts corners” for runtime efficiency. A majority of websites can work with MySQL. There are tools scalable that are easy of use and easy to manage. Depending on the database-engine, MySQL can lack certain features such as the full-text search.

4.4 PostgreSQL

PostgreSQL is the advanced, open-source object relational database management system that has become standard and is the DBMS that is used for our prototype. PostgreSQL is different from other RDBMSs because of its highly object-oriented functionality. It becomes very efficient in handling many tasks very efficiently. PostgreSQL is open-source and free and supported by a large and experienced community. Some possible disadvantages to PostgreSQL includes any simple use database to be overly complicated and would be better used with MySQL.

4.5 Selection

As mentioned, the prototype currently uses PostgreSQL and will continued for the entirety of the project. Viewing in the advantages and disadvantages of other technologies, PostgreSQL shows the most stability and provides the most efficiency that our project has as an objective. It provides substantial data integrity and may be useful for integration of some potential complex designs with the websites. Any type of custom feature designed would have the least amount of trouble being implemented with PostgreSQL.

5. Front end – User Interface

5.1 Introduction

The majority of the website built will be dependent on its UI and how the user interacts and navigates through the site and its various options. This process requires and utilizes many different technologies including PHP, HTML, or JavaScript. The prototype utilizes the UI with the web framework of Django and its designing capabilities. With our front end piece, we will not have a specific technology that we will use entirely. Instead, we are utilizing part of each technologies to handle our data management, website navigation process, and the website’s graphical layout. Possible technologies comprised for handling the front end user interface are listed below.

- PHP
- HTML
- JavaScript

5.2 PHP

PHP is a server-side language designed for general purpose as well as web development. With the ability to access and connect to external databases, it would be utilized for accessing web servers and connecting to our PostgreSQL database and access all of the users and groups in it. PHP can be mixed easily with HTML code and vice versa. Any PHP code is interpreted and executed by the web server that sends its output to the client. PHP is integrated and utilized into a large majority of the websites on the internet.

5.3 HTML

HTML is the most basic building block of all websites. The technology allows images and objects to be embedded and used to create interactive forms. Our web framework technology uses HTML to build its websites through its own interface and becomes the front end of our piece. HTML user interface is more secure than most sites as there is less of a change that you will get hacked. Another advantage is that you have a lot of control over the UI. Another advantage is that other coding languages can be easily integrated into the websites user interface. Some disadvantages of HTML include the length of time it takes to construct a UI with HTML. Another disadvantage of HTML is that if one character is out of place it can mean the entire UI doesn't load properly and it is a much more tedious process. The final disadvantage is that simple changes to the UI can take much longer to implement than you are willing to spend since you may have to make those changes one page at a time.

5.4 JavaScript

JavaScript is one of the most simple, versatile and effective languages used to extend functionality in websites. advantages of JavaScript include execution on the client side which means that the code is executed on the user's processor instead of the web server thus saving bandwidth and strain on the web server. JavaScript is also easy to learn and comprises syntax that is close to English. It uses the DOM model that provides plenty of prewritten functionality to the various objects on pages making it easier to develop a script to solve a custom purpose. Some disadvantages would include security issues. JavaScript snippets, once appended onto web pages execute on client servers immediately and therefore can also be used to exploit the user's system. While a certain restriction is set by modern web standards on browsers, malicious code can still be executed complying with the restrictions set. Another disadvantage would be JavaScript rendering varies for the user interface. Different layout engines may render JavaScript differently resulting in inconsistency in terms of the user interface.

5.5 Selection

Based on the prototype that has been implemented already we have decided to select HTML as our user interface technology as it works great with Django. With the advantage of easily integrating other coding languages into the website means that we can add other language features as well. HTML also has a lot of UI features working with Django and with the security of HTML being very good we think this is the best technology for our webpage user interface. JavaScript and PHP both have their benefits but we feel that HTML is the best fit for our project.

6. User and Group Management System

6.1 Introduction

The user and group management technology in relation to this project is very important as one of the requirements is to enable user and group management. As we continue building the web page we will be implementing a user and group management system that will allow users to create accounts. This is also key in the admin section as admin accounts will be needed for the admin users of the webpage. The technologies looked into are listed below.

- LDAP
- AD
- Django Authentication System

6.2 LDAP

LDAP is abbreviation for lightweight directory access protocol which is a software protocol for enabling anyone to locate organization, individuals, and other resources such as files and devices in a network, whether on the public internet or on a corporate intranet. The main benefit of using LDAP is the consolidation of certain types of information within a system. An example that supports this is all of the different lists of users within the system can be merged into one LDAP directory. This

directory can be queried by any LDAP-enabled applications that need this information. The directory can also be used by users who need directory information. Other LDAP benefits include its ease of implementation, and its well defined Application Programming Interface (API). On the con side of LDAP is that if you want to use this user and group management technology you will need LDAP enabled applications or you will need to use LDAP gateways. There currently aren't a plethora of LDAP enabled applications available for Linux. While LDAP does support some access control, it does not support as many security features.

6.3 AD

For quite some time the standard in the user directory space has been Microsoft's Active Directory (AD), which is embedded in organizations large and small. Some of the pros of using AD is active directory is generally considered to be a significant improvement over Windows NT Server 4.0 and AD provides a centralized administration mechanism over the entire network. It also provides for redundancy and fault tolerance when two or more domain controllers are deployed with a domain. Active directory automatically manages the communications between domain controllers to ensure the network remains viable. Users can access all resources on the network for which they are authorized through a single sign-on. All resources in the network are protected by a robust security mechanism that verifies the identity of users and the authorizations of resources on each access. The cons of AD are that it is difficult to integrate into pre-existing network systems. AD offers no means to manage non-Windows clients (such as Macintosh or UNIX) or servers and supports very little management control over pre-Windows 2000 systems. Another con is that AD relies upon DNS to function, but not all DNS servers are capable of supporting AD. Existing DNS systems may need to be upgraded or replaced before they can support AD.

6.4 Django Authentication System

When looking into the Django authentication system I have found a lot of documentation that helps in implementing the system so that a web template can be created where users and groups can be created. This is also easily managed as it uses a database to store the user's information such as MySQL. Another pro to using the Django authentication system is that we are using Django as our framework for the web page that our tools will be utilized on. This will allow us to already have the tools necessary to implement the management of users and groups. Another pro to this system is that it is easy to set up the admin accounts so that accounts can be handled if needed. Another pro for using this system is that it lets you plug in other authentication sources as you can override Django's default database-based scheme, or you can use the default system in tandem with other systems which makes it more portable. Looking through the documentation I was able to find some disadvantages of using Django authentication system as there are some documentation gaps that need to be addressed. Another disadvantage is that it can get difficult to implement as there are small but time consuming hurdles as we are on a tight time frame for the next couple of terms.

6.5 Selection

We have decided to select the Django authentication system as our technology for user and group management as the benefits from doing so give our team greater advantages than choosing the AD or LDAP methods. Since we are already using the Django web framework it will be much easier to implement the authentication system as the tools necessary are already at our disposal. AD and LDAP are good systems however the complexity of their implementations and how they seem to have limitations on interacting with other systems means it is less portable. Django as our framework is the main reason for selection but the system also offers portability, good documentation, and simplicity.