

RoboTractor

Jeremy Wright
Arizona State University
jlwrigh1@asu.edu

Arun Balaji Buduru
Arizona State University
abuduru@asu.edu

David Lucero
Arizona State University
dwlucero@asu.edu

Abstract—Here you need to describe the abstract of your project including (a) Problem statements, (b) project scope, (c) main tasks, and (d) schedule ***CRITICAL: Do Not Use Symbols, Special Characters, or Math in Paper Title or Abstract.** (Abstract)

Index Terms—put indexing key words here that (key words)

I. INTRODUCTION

In this section, describe:

- 1) The problems to be addressed in this project.
- 2) Why these problems are important?
- 3) Applied technologies and solutions to address these problems.
- 4) Expected outcomes of this projects.
- 5) Project management plan (timeline, and group members, etc.)

II. SYSTEM MODELS

A. System Model

RoboTractor will leverage the Django Web framework [1], to realize the required interfaces. Figure 1 describes the connection of these components. The combination of XMPP and REST in RoboTractor is a demonstration of how to leverage the existing HTTP development environment i.e. "the web of things" into a stateful protocol. HTTP is by design stateless. XMPP on the other hand is a stateful streaming connection between two clients. In this case the Tractor and a command server. To achieve this mesh we will leverage the Bidirectional-streams Over Synchronous HTTP standard protocol [2]. BOSH provides a standard mechanism to operate the streaming XMPP protocol efficiently over an HTTP connection. This is essential for a scalable webservice.

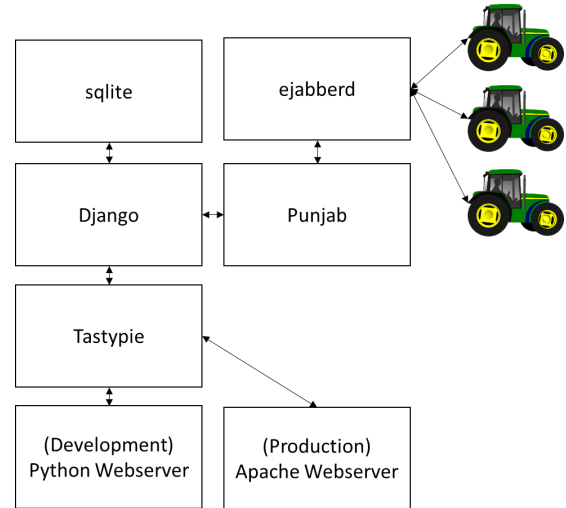


Fig. 1. Software Component Block Diagram

B. Software

1) *XMPP Server*: RoboTractor uses the ejabberd [3] XMPP server as it provides an existing Python interface, to integrate with the rest of our Python based ecosystem.

2) *REST Interface*: TastyPie provides REST [4] by extending the existing Django Models.

3) *BOSH Interchange*: Punjab is an Django plugin implementation of BOSH [5]. In addition to BOSH, this library also combines th ejabberd Users with Django Users to provide a single authentication and authorization framework. While this demo project will have a single user type, this combination is critical to maintain proper use management and least-privilege authorization.

Describe needed software (applications, tools, APIs) to develop this project.

C. Security Model (optional)

This section is needed if the project is focusing on security. Describe attack source, attack goal, attach

methods, and attack consequences.

III. PROJECT DESCRIPTION

Summarize the project description here.

A. Project Overview

Describe how many project tasks are proposed and what their relations (dependency) in the project. Use diagram is needed. Provide midterm and final goals of this project.

B. Task 1 : Development Environment

The development environment consists of configuring all the tools needed to portably work with the software package. This will involve installing project dependencies, and deployment scripts to allow all members of the team to work effectively together. The complete environment will be stored in git.

C. Task 2: XMPP Implementation

XMPP Implementation is primarily a configuration task to setup the ejabberd server and connect it into the HTTP Framework. Once this is in place the XMPP Clients may start working.

D. Task 3: Demo Tractor

The Demo tractor is the complementary component to the Frontend UI. This is the virtual machine, a piece of software which simulates a real tractor, or agricultural vehicle.

E. Task 4: Frontend UI

The "single-page" web application is the modern design methodology to web apps today. Leverging this design architecture the Frontend will query the REST api to draw the position of all tractors within a Google Map context. This task has 2 high level sub tasks:

- 1) Path generation
- 2) Map rendering

Path generation is the primary deliverable of this project. The user shall be able to input a path for a given tractor to drive. The Tractor will then receive this path over the REST to XMPP gateway. The UI may then periodically query the position of the tractor and update it on the Google Map.

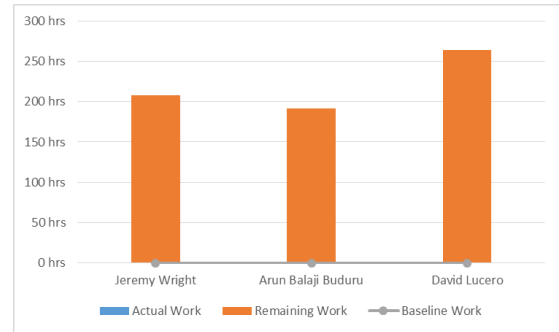


Fig. 2. Resource Allocation

F. Task 5: Server Backend

The Server Backend is the critical component which plumbs all the components together. Extensive knowledge of Django will make this task easier. As it will require linking multiple components together in a orchestrated fashion.

G. Task 6: REST Interface

The REST interface will server the primary means of interacting with the site. The Web interface will exist as a "single-page" app who leverages AJAX principles over this REST API.

H. Project Task Allocation

Jeremy Wright will serve as the project lead since he has past experience with Python based web application.

I. Deliverables

Describe the expected outcomes of the projects: e.g., software packages, tools, algorithms, system designs, publishable materials (manuscripts, white papers, surveys, etc.).

J. Project Timeline

The Attached project Timeline (generated from Microsoft Project) describes the overall tasks of the project. Please see the attached gantt for details.

IV. RISK MANAGEMENT OF THE PROJECT

Describe (a) what potential issues may prevent this project from being successful, (b) what mitigation strategies to prevent/mitigate the identified issues. A good project design should consider what if a task fails. How likely (low, medium, high), the proposed tasks may fail.

Are there alternates/makeup/get-around approaches available? Better use a table to highlight the risks and corresponding mitigation strategy.

V. CONCLUSION

(a) summarize the project proposal, (b) Describe potential future work (or applications) that can be built based on the proposed work.

VI. ACKNOWLEDGMENT

Put sponsor/mentor/assistance acknowledgments on developing this project proposal.

Reference is very important in your report. Please highlight where you have referred technical terms and solutions in the content. Following the IEEE citation format and provide a complete citation for each reference. The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]do not use Ref. [3] or reference [3] except at the beginning of a sentence: Reference [3] was the first ... Here are a few examples. [6]

REFERENCES

- [1] (). Django, GitHub, [Online]. Available: <https://github.com/django> (visited on 02/06/2014).
- [2] I. Paterson, D. Smith, P. Saint-Andre, and J. Moffitt. (Jul. 2, 2010). Bidirectional-streams over synchronous HTTP (BOSH). This specification defines a transport protocol that emulates the semantics of a long-lived, bidirectional TCP connection between two entities (such as a client and a server) by efficiently using multiple synchronous HTTP request/response pairs without requiring the use of frequent polling or chunked responses., [Online]. Available: <http://xmpp.org/extensions/xep-0124.html> (visited on 02/06/2014).
- [3] (). Ejabberd community site | the erlang Jabber/XMPP daemon, [Online]. Available: <http://www.ejabberd.im/> (visited on 02/06/2014).
- [4] (). Toastdriven/django-tastypie, GitHub, [Online]. Available: <https://github.com/toastdriven/django-tastypie> (visited on 02/06/2014).
- [5] (). Twonds/punjab, GitHub, [Online]. Available: <https://github.com/twonds/punjab> (visited on 02/06/2014).
- [6] D. Huang, "IPv6," Arizona State University, 2014.