BOINSO Documentation

Release 0.0.1

Marcel Cimander, Gregor Beyerle, Lars Mehnen

May 11, 2015

Contents

CHAPTER 1

Contents:

1.1 About

1.1.1 What is BOINSO?

BOINSO stands for "Berkley Open Infrastructure for Networking Satellite Operations". This name was chosen in honor to the BOINC and GENSO projects.

BOINSO is an attempt to create a Genso like structure from scratch. In contrast to GENSO this project will be open source and free to use in a less localized way so that every mission control center can tweak the system to its needs.

1.1.2 The current situation

A typical student space segment

A small satellite in Low-Earth Orbit, often Sun Synchronous, Low-power transmitters, Simple and standard communications protocols (such as AX25), Use of the Amateur Radio frequency bands: VHF, UHF and S-Band.

A typical student ground segment

A single, local, groundstation, usually at the host university, Capable of communication on one or two of the Amateur Radio frequency bands, A single rotator and a single elevator to track the spacecraft, A single PC controlling the groundstation hardware and the mission data.

Typical Limitations

From ~15 orbits there are around six passes a day, averaging perhaps five minutes each, Satellite is in communications range less than 3% of the mission time, For 97% of the time the groundstation is idle, The groundstation is not configured to communicate with other educational spacecraft, The spacecraft is only configured to communicate with the specific groundstation.

1.1.3 Our approach

Advantages of sharing resources

Provides near-global coverage for all participating missions, Allows for a dramatic increase in mission return, Many critical operations would benefit from having uninterrupted coverage for several hours, Powerful error-correction can be applied when using multiple downlink stations, Mutually beneficial at extremely low risk (original solution serves as backup).

1.2 License

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

1.2.1 Appendix

How to apply the Apache License to your work To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

1.3 How to contribute

As with every open source project on github you just have to fork the repository, implement your ideas and send a pull request. If you want to contribute on a regular basis just send an e-mail to gregor and you will be added as a contributor.

1.3.1 Code Conventions

We use the pyflakes and pep8 modules to enforce a tidy coding style. We also try to stick to the basic architecture of the Django project as it provides a very solid basis for maintanable and scalable web applications. We aren't by any means "professionals" and our work may break these conventions so if you find any error just let us know! If you use a pep8 linter you can exclude the following warnings:

> E501, C0301, W0142, W0402, R0201, E1101, E1102, C0103, R0901, R0903, R0904, C1001, W0223, W0232, W0201, E1103, R0801, C0111

1.3.2 Vagrant

We are using very different development setups in our team. As we don't want to spend our time with fixing "But-It-Works-On-My-Machine"-bugs we like to use Vagrant. If you have never tried Vagrant for one of your projects don't be afraid to test it. The base container is a basic Ubuntu 14.04LTS 32bit machin including a python2 interpreter, a python3 interpreter and a postgres database. Currently we only support virtual box but in the future we plan to also add VMWare Player and Hyper-V boxes.

Right now we only use a basic shell provisioner. If your worklow depends on a special setup you can customize *bootstrap.sh*.

To use our provided boxes you will need an Atlas account!

1.3.3 Basic setup

If you don't want to use Vagrant this is no problem. Just run the application as you would any other Django app (you will have to change the database backend in *Boinso.settings.py* though). If you use a normal Python installation make sure to use a virtual environment. If you favour using Anaconda stick to destinct conda environment. Naturally you can do whatever you want with your local setup but seperating application dependencies doesn't interfere with other project dependencies.

1.3.4 Testing

We are always eager to improve our test coverage. If you implement new functionallity please add automated tests!

1.4 Client Development

1.4.1 Why Clients?

The BOINSO Mission Control Center web application is designed to offer the end user (Mission Control Center) and the different participants (Ground Control Clients) as much flexibility in client design as possible. The core web application itself is exposed through different API endpoints (callable through the browsable API or other applications). To offer a convinient start using BOINSO we also provide a web application which can be used to administer a MCC or to register as a new GCC.

1.4.2 Where to find it?

The client web application resides at this Github repository location: BOINSO-MCC-Web-Client

1.5 Need Help?

Gregor, who is currently responsible for the implementation of the server side parts, can be reached here.

1.5.1 Further help

You can allways check gitter for currently active collaborators.

1.6 Models

class core.models.**Satellite**(**args*, ***kwargs*)
Satellite model represents an earht orbiter. Closely modelled after GPredict satellite representation.

class core.models.**Transponder**(**args*, ***kwargs*)
Transponder model. Pretty close to transponder representation in GPredicts trsp files.

class core.models.**UserProfile**(**args*, ***kwargs*)
User Profile model adds information to ground control clients. Right now nothing happens with that information as there are no active passes scheduled right now (just downlink).

1.7 Serializers

class api.serializers.**LoginSerializer**(*instance=None*, *data=<class rest_framework.fields.empty>*, ***kwargs*)
Used for initial log in (still http basic). Returns client_id and client_secret which in turn can be used to request OAuth2 token.

class api.serializers.**SatelliteSerializer**(*instance=None*, *data=<class rest_framework.fields.empty>*, ***kwargs*)
Serialzes Satellite objects. Clients should see data regardles of their login status. Read only.

class api.serializers.**SignUpSerializer**(*instance=None*, *data=<class rest_framework.fields.empty>*, ***kwargs*)
Serializer that takes/returns OAuth2 application client_id and client_secret. Used to sign up new users.

class api.serializers.**TransponderSerializer**(*instance=None*, *data=<class rest_framework.fields.empty>*, ***kwargs*)
Serializes transponder objects. One Satellite can have multible transponders. Transponder info is optional.

class api.serializers.**UserProfileSerializer**(*instance=None*, *data=<class rest_framework.fields.empty>*, ***kwargs*)
Serializes User Profiles, giving Clients the possiblility to see their profiles.

class api.serializers.**UserSerializer**(*instance=None*, *data=<class rest_framework.fields.empty>*, ***kwargs*)
Basic user serializer exposing Djangos core authentication user model.

1.8 Views

api.views.**api_root**(**args*, ***kwargs*)
API root endpoint. Gives information about all available Endpoint branches.

class api.views.**Login**(***kwargs*)
Login endpoint for existing users. Returns client_id and client_secrete for subsequent OAuth2 token requests. Only part of the application that still requires HTTP Basic Authentication.

serializer_class
alias of LoginSerializer

class api.views.**SatelliteDetail**(***kwargs*)
Detail endpoint related to SatelliteList.

serializer_class
alias of SatelliteSerializer

class api.views.**SatelliteList**(**kwargs)
Public endpoint to acess a list of mission control satellite data. All userers may see the data.

serializer_class
alias of SatelliteSerializer

class api.views.**SignUp**(**kwargs)
Endpoint for signing up new users. Returns client_id and client_secret for initial OAuth2 token request.

serializer_class
alias of SignUpSerializer

class api.views.**TransponderDetail**(**kwargs)
Detail endpoint for transponders.

serializer_class
alias of TransponderSerializer

class api.views.**UserDetail**(**kwargs)
Generic user detail endpoint. Authenticated users see the details of one distinct user.

serializer_class
alias of UserSerializer

class api.views.**UserProfileDetail**(**kwargs)
Detail view for UserProfile. Authenticated can update or destroy their Profiles.

serializer_class
alias of UserProfileSerializer

class api.views.**UserProfileProxy**(**kwargs)
Narrows down the search for a user via his authentication. Authenticated user sees his own profile and a link to his user endpoint. Is used to get userdata via oauth token authentication.

get_queryset()
This view should only get the profile of the authenticated user.

serializer_class
alias of UserProfileSerializer

1.9 Indices and tables

- genindex
- modindex

a

c

A

C

G

L

S

T

U