Process for Approving SEED Platform Hosting Providers

The purpose of approving Standard Energy Efficiency Data Platform™ (SEED) hosting providers is to ensure that the SEED Platform trademark is being properly used, and end-users who use a SEED-branded service get the experience and core functionality they expect from the SEED Platform application. The operating principle is that modifications to the core open-source code are not allowed, other than minor modifications needed to register plug-ins or make cosmetic changes for purposes of branding the provider's service offering.

In order to offer a branded SEED Platform service, a hosting provider must follow these steps and execute a sublicense agreement with LBNL.

1. **Hosting Provider Commitment.**
    a. Provider reviews publicly available sample SEED Platform trademark license and logo usage guidelines.[1]
    b. Provider contacts LBNL via email indicating their intent to offer SEED Platform hosting, as well as their acceptance of the license terms and intent to proceed with the approval process.
2. **Set up and testing.**
    a. Provider installs SEED Platform instance on their test environment and provides LBNL a user account, including API credentials.
    b. Provider runs compliance tests listed below to identify any discrepancies with expected SEED Platform behavior and submits a test report to LBNL (see sample report attached). The test report can be an email with text file attachments.
        i. Identify base version of open-source SEED Platform being offered,
        ii. Text description of plug-ins/extensions added to open-source code,
        iii. Results of diff comparison between provider's code and open-source SEED Platform repo (using the "git diff --stat" command),
        iv. Results of API test run against provider's production SEED instance.[2]
        v. Note LBNL can require additional testing if results above are inconclusive.
    c. LBNL identifies any discrepancies in test report; Provider addresses these, and resubmits test reports as needed.
    d. LBNL runs the API tests on the provider's SEED Platform environment, using LBNL's login; provider addresses any issues raised during testing and submits a final test report to LBNL.
    e. LBNL gives provider access to the SEED Platform logo electronic files.
    f. Provider installs SEED Platform logo in their instance.
    g. LBNL does manual check of logo usage; Provider addresses any discrepancies in logo usage.

[1] Basic intent of guidelines is to prevent modification of the logo and restrict its use to only identifying installations of the SEED Platform application, not promoting companies or products.

[2] API test: use "test_seed_host_api.py" script in the /seed/tests/api/ folder in the seed github repository.

3. **Final Approval**
    a. Approval decisions will be made on a case-by-case basis, but generally only the following changes to the SEED Platform code will be allowed:
        i. Cosmetic changes to customize branding to provider,
        ii. Changes needed to register a plug-in (Django app),
        iii. The addition of files that comprise the Django app or a standalone app that accesses the APIs.
    b. LBNL sends provider signed license for countersignature.
    c. After provider signs and returns the license, LBNL will add the provider name to the publicly-available list of officially-approved SEED Platform hosting providers.

4. **Annual Renewal**
    a. 30 days before the annual anniversary of the license signature date, provider will send LBNL an annual report with the following information:
        i. statement of provider's intent to continue using the trademark,
        ii. updated test report,
        iii. the number of organizations and user accounts currently in their environment (for DOE's internal program evaluation purposes only),
        iv. description of ways the SEED Platform trademark has been used other than in the software (e.g., brochures, email campaigns, etc.).
    b. LBNL and provider execute new license every year.

Sample

SEED Hosting Provider Test Report

Provider Name:

Date:

SEED instance URL: *seed.example.com*

Base SEED version: v1.1 <e5e52ed> (*use the hash from the version in the SEED-platform GitHub repo*)

Description of additions or extensions added:
Text description of additions to code, both functionally from the users perspective, and how implemented in the application (e.g., as a Django app, with a new GUI page visible at a given URL, and a new API endpoint, etc.).

Diff report:
Attach text file
[example:]

```
git diff --stat 572871b master
 README.md                          |  24 ++++++++++++++----------
 docs/source/help.rst               |  26 +++++++++++++++++++++++++-
 howto/API_endpoints.pdf            | Bin 0 -> 236638 bytes
 howto/adding-new-BEDES-columns.txt |   4 ++--
 howto/resetting-the-database.txt   |   2 +-
 samples/demo_app/LICENSE           |  33 ---------------------------------
 6 files changed, 42 insertions(+), 47 deletions(-)
```

API test report:
Attach text file
[example with detail redacted:]

Hostname: Example
URL: http://www.example.com
Test Date: 2015-02-05 16:19:44
-------Accounts-------
API Function: get_user_profile
...
API Function: get_organizations
...
API Function: get_organization
...
API Function: update_user
...

```
API Function: get_organizations_users
...
API Function: get_query_threshold
...
API Function: get_shared_fields
...
-------Dataset-------
API Function: create_dataset
...
|---Covered Building File---|
API Function: upload_file
...
API Function: save_raw_data
...
API Function: save_column_mappings
...
API Function: remap_buildings
...
API Function: get_column_mapping_suggestions
...
API Function: start_system_matching
...
|---Portfolio Manager File---|
API Function: upload_file
...
API Function: save_raw_data
...
API Function: save_column_mappings
...
API Function: remap_buildings
...
API Function: get_column_mapping_suggestions
...
API Function: start_system_matching
...
API Function: get_PM_filter_by_counts
...
API Function: search_buildings
...
-------Project-------
API Function: create_project
...
API Function: get_project
```

```
...
API Function: add_buildings_to_project
...
-------Labels-------
API Function: add_label
...
API Function: get_labels
...
API Function: apply_label
...
-------Export-------
API Function: export_buildings
```