HPSS Collaboration Disclaimer

Forward looking information including schedules and future software reflect current planning that may change and should not be taken as commitments by IBM or the other members of the HPSS collaboration.
Purpose

• This is a deep dive into the HPSS Treefrog Project’s use of open source to streamline the development effort and broaden functionality delivered within resource budget.
• Want to educate people on the depth and breath of our open source usage.
• Want to show that the open source that we have selected to use has a good pedigree and strong community support.
• We have compartmentalized functionality so we can upgrade components.
Use of Open Source Software (OSS) and COTS
Minimize in-house development and broaden function
SQLAlchemy

• SQLAlchemy is the Python SQL toolkit and Object Relational Mapper.
  o Abstracts and simplifies the SQL layer during development.
    • Python classes are mapped to the database.
    • We do not write SQL.
• This compartmentalization allows for possible future upgrades.
  o Supports MySQL, Oracle, SQLite, DB2, …
  o TreeFrog Project software uses DB2.
• Used by:
  o Yelp, reddit, DropBox, Openstack, and other major software and service deployments.
**SQLAlchemy**

**Example SQL Alchemy Model**

```python
class StgPolCopy(Model):
    __tablename__ = 'STG_POL_COPY'
    stg_pol_id = Column('STG_POL_ID', BigInteger)
    stg_pol_name = Column('STG_POL_NAME', String)
    ...
```

**Example python using the models**

```python
for policy in db.session.query(StgPolCopy):
    print 'Policy names: %s' % policy.stg_pol_name
```
Use of Open Source Software (OSS) and COTS
Flask

• Flask is a web framework.
  o It provides tools to build web applications.

• Flask supports extensions that add application features.
  o HPSS TreeFrog uses Flask extensions for form creation, table building, OpenID support, and Login functionality.
  o Other popular extensions allow Flask to send emails, perform two factor authentication, or SMS messaging.

• Who uses Flask?
  o Red Hat, Netflix, Lyft, Uber, Samsung, Airbnb, and other major software and service deployments.

• Flask is used to provide the web interface for the TreeFrog Administrative and REST API for User interface.
Administrative Interface

• Administrator interface is browser based.
  o Flask generates the screens based on the SLQ Alchemy models.
  o Accessible from web browsers.
• Flask model views, WTForms, and Tabulator are the extensions used to build the administrative pages.
• Handles all SQL Alchemy calls to displaying, add, update, or delete a row.

```python
class StgPolModelView(ModelView):
    route_base = '/v1/policies'
    datamodel = TreefrogSQLAInterface(StgPol)
    list_title = _('Storage Policies')
    list_columns = ['stg_pol_name', 'stg_pol_desc', ..]
    show_title = _('Show Storage Policy')
    show_columns = ['stg_pol_name', ..]
    add_title = _('New Storage Policy')
    add_columns = ['stg_pol_name', ..]
    edit_title = _('Edit Storage Policies')
    edit_columns = ['stg_pol_name', ..]
    search_columns = ['stg_pol_name', 'stg_pol_desc', ..]

    def post_add(self, item):
        ...

    def post_update(self, item):
        ...

    def post_delete(self, item):
        ...
```
User API

- Flask and Swagger are used to generate a REST API for the User API.
  - Accessible with a URL
  - `https://<host>[:<port>]/ + <location>`
  - Results are in JSON format.
- Simplifies the effort require to develop applications to interact with TreeFrog.
  - Not dependent on a programming language.
  - Not dependent on an operating system.
- Swagger is a machine readable format for describing an API.
  - Provides interactive documentation.
- Used by: Netflix, IBM, MS Azure, and other major software and service deployments.

Example Swagger definition

```
/v1/api/datasets/{project}:
  get:
    tags:
      - "dataset"
    description:
      "This API is used to retrieve a list of datasets for the given project."
    operationId: "list_datasets"
    Parameters:
      - name: "project"
        in: "path"
        description: "Project Name"
        type: "string"
    responses:
      200:
        description: "Success"
        schema: $ref: "#/DatasetResponse"
      403:
        description: "Forbidden"
      404:
        description: "Not Found"
    security:
      - SessionCookie: []
      x-swagger-router-controller: "treefrog.api.controllers.dataset_controller"
```
Use of Open Source Software (OSS) and COTS
Zookeeper

• Zookeeper is an open source coordination service.
• TreeFrog uses it to communicate across nodes, processes, and threads.
  o Synchronize the job queue.
  o Communicate server/copier/endpoint changes.
  o Manage server/copier/configurations configurations.
  o Communicate resource allocation.
  o Failover notification.
• ZooKeeper was a sub-project of Hadoop but is now an Apache Project in its own right.
  o Used in Hadoop and Apache Storm.
• Kazoo is the Python wrapper for Zookeeper.
Life of a Treefrog Ingest Request

Legend:
- NEW Code
- Extendable Functionality
- Existing Product

Diagram:
- Flask
- Scheduler
- Workers
- Copier
- Zookeeper
- DB2
- Kazoo
- Swagger
- SQL Alchemy
Zookeeper

• Looks like a directory tree
  o Tree elements are called znodes.
  o Can contain data and have children.
  o Znodes can be locked.
• Znodes can have callbacks called watchers.
  o Can watch for new nodes, deleted znodes, or data changes.
OSS and COTs in the Copier

- **Jclouds** - The Java Multi-Cloud Toolkit.
  - Provides TreeFrog with a put and get interface to storage.
  - Allows for extensions to support new storage types.
  - Used by: Twitter, Adobe, Red Hat, Salesforce, and other major software and service deployments.

- **Jerasure**
  - Provides an erasure coding library for TreeFrog.
  - Used by: Ceph
Open SIRF

• Self Contained Information Format
  o Defines a metadata catalog schema.

• The catalog contains the dataset manifest in JSON format. It provides:
  o The location and size of the data.

• Catalog acts as a preservation format.
  • The data can be recovered without TreeFrog with only the catalog.

• The catalog is self descriptive
  o Using the JSON format we can create datasets out of directories, tar files, Bagit data, buckets, …
Questions?