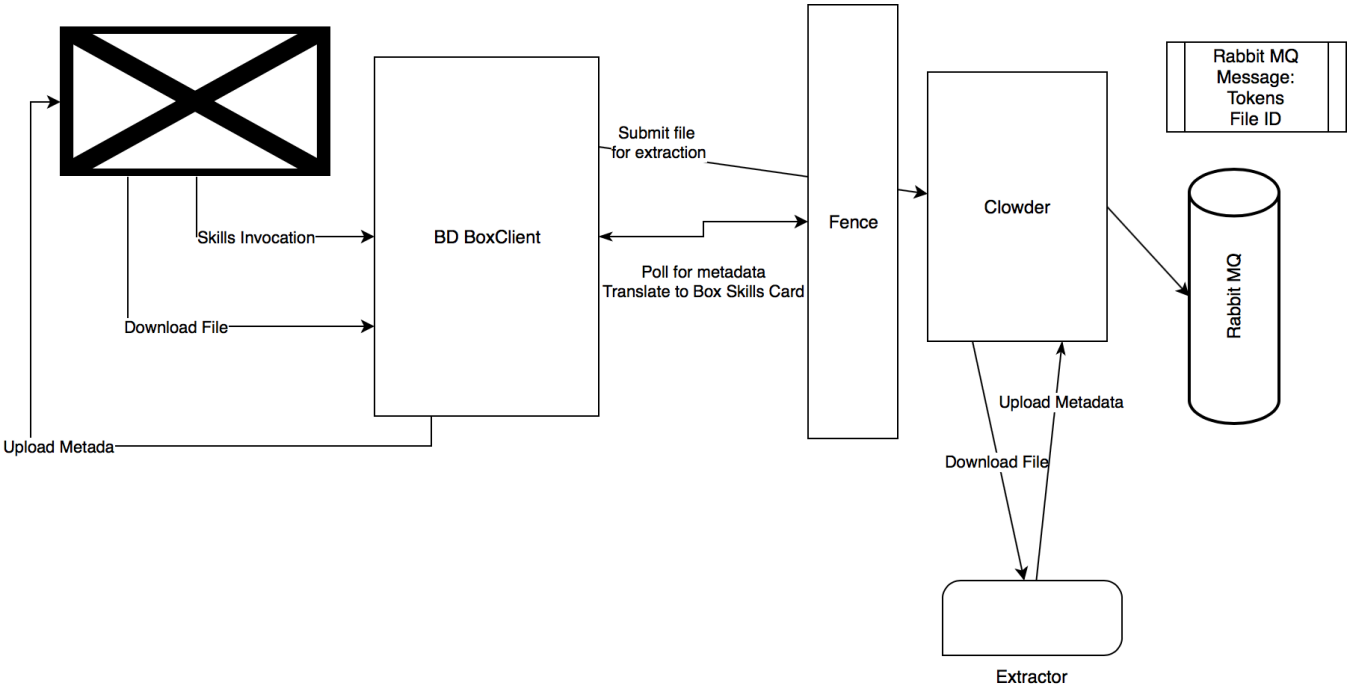
Box Skills Support

This page outlines an approach for Brown Dog to become the definitive source for scientific community curated box skills.

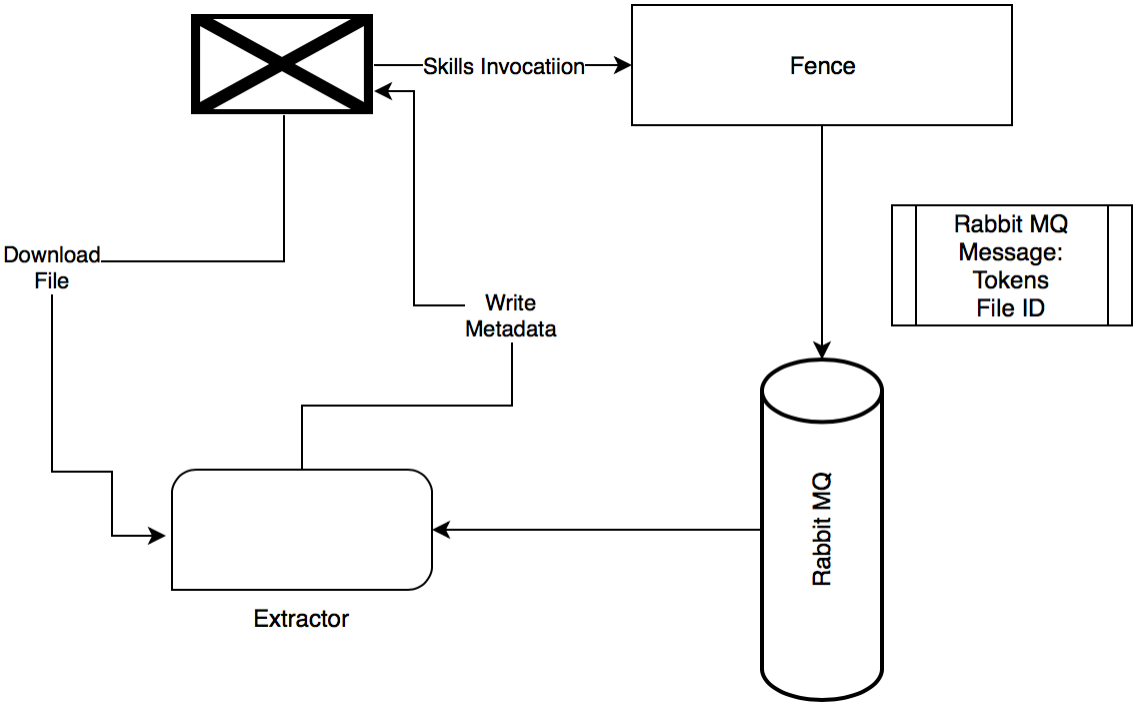
Architecture

There are two possible architectures:

Current "Standard" Approach: Create a Box client application that will handle the additional functionality not supported by Box Skills (e.g. polling, metadata mapping) involved with interacting with Fence, likely using bd.py:



Box Tuned/Potentially Streamlined: Enhance pyclowder to enable it to download files and upload metadata to Box in addition to clowder (eliminating data movement that isn't required). Eventually support would be added to Google Drive and Dataverse, etc.



When a Skill is registered with a Box account, the invocation URL is provided. This URL will resolve to an endpoint in Fence.

Comparison of Approaches

Current "Standard" Approach	Box Tuned/Potentially Streamlined
If using the /extractions/file endpoint the file is transferred three times: 1. From Box to BoxClient 2. From BoxClient to Clowder (via fence) 3. From Clowder to extractor container If using the /extractions/url endpoint the file is transferred ?? times: (not sure we could use the URL endpoint since the Box file won't be publicly accessible, Has to be downloaded via the Box API. 1. ?? File lives in clowder until the cleanup script is run	File is downloaded once from box to the extractor container File lives in extractor and deleted at end of process_message by PyClowder (is this correct, i.e. by PyClowder?)
Box SDK only lives in the BoxClient service. No changes are required elsewhere, however, this client will need to be maintained (by us or external party)	Box SDK has to be introduced into Pyclowder library. Any other repos we want to support would also have to be included in the future Burden of maintenance/adding new supported external services is squarely on our end (vs on the client's end) **Question:** Is Pyclowder the right place for this? PyClowder is just a convenient wrapper mechanism for creating "some" extractors that happen to be written /wrapped in python. This will make it more heavy weight and leave out other languages.
An automated translation of clowder metadata to box skills cards would have to be developed, which may be difficult, or may not (sounds like there are only 4 or 5 types)	Custom metadata structure for box would be implemented in the extractor What happens when an extractor doesn't support a specific service (e.g. Box, Dataverse)?
Potential Bottlenecks for massive scaling: 1. Fence 2. RabbitMQ 3. Extractors 4. BoxClient 5. Clowder 6. MongoDB **Notes:** We can't rely on threading in the BoxClient to do the polling since we would risk running out of threads. *We can run a small experiment to get some numbers here (e.g. average time per request, cpu hours/memory utilized, network I/O)*	Potential Bottlenecks for massive scaling: 1. Fence 2. RabbitMQ 3. Extractors **Notes:** Everything apart from Rabbit is stateless and can be horizontally scaled. *We can run a small experiment to get some numbers here (e.g. average time per request, cpu hours/memory utilized, network I/O)*
Would need to deploy a new service and proxy it behind Apache BoxClient would need to be allocated a service account and handle Brown Dog tokens	Would need to add some endpoints to fence
Errors can be reported in Clowder (not visible to the Box user) The BoxClient could potentially retry	Limited error logging and reporting Unsure about retry if an extraction fails Eventually we could create an app where user logs in via their Box credentials to see the history of extractions

Skills Invocation

When a file is uploaded to a Box folder that has an attached skill, the following payload is POSTed to the fence endpoint:

Sample Box Skill Invocation


```
{
  "type" : "skill_invocation",
  "skill" : {
    "id" : "469",
    "type" : "skill",
    "name" : "NCSA Brown Dog Demo",
    "api_key" : "m0rhky5ah812g7w52tsbrz2qzt2l1sqn"
  },
  "token" : {
    "write" : {
      "access_token" : "1!
G_elWx4TWjF1brZsCzoKF1BEA6OCP4DjGZZBGNfLI1J57hScZ2DbDp8IUGS2dN0Cjp0tWU5e_JybL41AYvxF8gpFcxil83kZa2kdiNKJbZm5m8Qz
_xzLO3Merb52nv4kXYHKLMKD2YNB-k6tEZiBRKvVCNDGdElW89aesULGbS4j4k0AQlMe9N93e3H7sbMfIHTDV-
zD06HqDEnaxZ71BQWQgEBhXUYDR0PiMIwkI4knC6DrbsackYVLZE47gYqEPkOSSJef81LgQLEp5vf2YXzxkZWIYT080WgamZOR8Wo4r28n0sLrB5
NAvrghKOWcFxOLoBa4dnltI8cEjzm2dYRwLVBo7SbnkwBUgZ6ATYA2vHZSYLHPdORSwPQLYu9iwgIGQbFa1_BZzkHkceC02-
oMXd4Cb37E5Z7sjyYt",
      "expires_in" : 1532043110,
      "restricted_to" : "[{\"scope\":\"gcm\"},{\"scope\":\"item_upload\",\"object_id\":305925334659,\"
object_type\":\"file\"},{\"scope\":\"manage_skill_invocations\"}]",
      "token_type" : "bearer"
    },
    "read" : {
      "access_token" : "1!bBZfVF3mUxemyCpxnbYfb-
uxE4IB8LWHfdK64WKOLC8pUGgoDUaxKWEDgv80_m4TffRBMISPEmIyfww_Ue5AslyiPHbNRt_E5vQqoHiFdkKDit1I0jVZ9p60NVcsCBOOe7-
YQSXlmiWiGlE-wUyfsQkdJ21MajxzSt7uPFQwHcKu_scJhOsi-
JkPlMi7jaWBD9R1CwOnGMmbyAh29ZwfqVUFnXX0Rzj5xSdSK_Ky051Wtsv2ndx2tr_Irn8o0p_P5qbStMQTN86CbNokEnHLtdsEPYLQ1QUDLMPTj
I0PRh2m3eEXlurSeJQbIJL5vO-DTzyApnVu4O8N4d429K8mOqlUI5g23w2c7nyGEt-M_M-g6Wo.",
      "expires_in" : 1532043110,
      "restricted_to" : "[{\"scope\":\"gcm\"},{\"scope\":\"item_read\",\"object_id\":305925334659,\"
object_type\":\"file\"}]",
      "token_type" : "bearer"
    }
  },
  "status" : {
    "state" : "invoked",
    "message" : "",
    "error_code" : "",
    "additional_info" : ""
  },
  "id" : "8e5373d5-3793-4627-8467-47c9f5554c12_819878470",
  "created_at" : "2018-07-18T16:31:49-07:00",
  "trigger" : "FILE_CONTENT",
  "enterprise" : {
    "type" : "enterprise",
    "id" : "61647055",
    "name" : "Pear Tree Studio"
  },
  "source" : {
    "type" : "file",
    "id" : "311371068932",
    "name" : "DESJ033735.8-402739.1.tif.png",
    "sequence_id" : "0",
    "file_version" : {
      "id" : "322195121235"
    },
    "owner_enterprise_id" : "61647055",
    "parent" : {
      "id" : "51605357386"
    },
    "old_parent_id" : "",
    "collab_accessible_by" : {
      "type" : "",
      "id" : "",
      "name" : "",
      "login" : ""
    },
    "size" : 24540
  },
```

```
  "event" : {
    "event_id" : "8e5373d5-3793-4627-8467-47c9f5554c12",
    "event_type" : "ITEM_UPLOAD",
    "created_at" : "2018-07-18T16:31:49-07:00",
    "created_by" : {
      "type" : "user",
      "id" : "3751070911",
      "name" : "Ben Galewsky",
      "login" : "box@peartreestudio.net"
    },
    "source" : {
      "type" : "file",
      "id" : "310871585151",
      "name" : "DESJ033735.8-402739.1.tif.png",
      "sequence_id" : "0",
      "file_version" : {
        "id" : "322195121235"
      },
      "owner_enterprise_id" : "61647055",
      "parent" : {
        "id" : "51605357386"
      },
      "old_parent_id" : "",
      "collab_accessible_by" : {
        "type" : "",
        "id" : "",
        "name" : "",
        "login" : ""
      }
    }
  },
  "parameters" : { }
}
```


Extractor Invocation

The RabbitMQ message for invoking an extractor will be changed to add a new property *source* which can be set to:

- clowder
- box
- dataverse
- etc...

The pyclowder library will use this source property to determine how to download the file and to update metadata. The extractor may use the source property to determine how to format the metadata.

Tools Catalog

The Brown Dog Tools Catalog will be the source of scientific community curated extractors. Extractors will be categorized into *Organizations* and *Repositories* within an organization. This pattern matches GitHub and DockerHub. The organizations will reflect scientific communities which will be responsible for curating the extractors in their repos.

Different versions and configurations of extractors can be specified by the use of *tags*.

Leverage ideas from binder to facilitate community extractor development/registering. Allow tools catalog to utilize an underlying Git repo for storing extractor_info documents and keep track of versions, issues, branches, and pull requests. It will download the `extractor_info.json` file to populate information on the page. This will additionally furnish Box enterprise admins with URLs that expose the tool as a skill. Initially they will have to copy and paste the URL. Once Box exposes management of Skills through an API this can be further automated.

Stories

Here are some initial stories to help us implement this vision:

Skills Workflow

Endpoints for Box Skills

Add unsecured endpoints to Fence for skills notifications in the form of /skills/*repo*/*extractor?*tag=*tag*

Source property for rabbitMq message

Add a new property to the rabbitMQ message that contains the source for the invocation (clowder, box, dataverse...)

Route Box Sills Invocations to correct Extractor

Translate the repo and extractor name to a queue name. Translate the tag into a routing key. Implement bindings to enforce routing messages by tools catalog tag.

Pyclowder downloads file from Box

Extend Pyclowder to observe the source property in the message and use the Box SDK to download the file locally to the docker container. Retain the existing functionality for Clowder sourced files.

Pyclowder uploads metadata to Box

Extend Pyclowder to make the files.upload_metadata method respect the source property and use the Box SDK to upload metadata to a box file. Re tain the existing functionality for Clowder sourced files.

Tools Catalog

Hello, Tools Catalog

Create a new Play 2.6 app based on the Clusterman code base.

GitHub Social Auth

Configure Silhouette with GitHub Social Auth

Organization Page

Configure a MongoDB Collection with basic organization data (basically the name and the GitHub URL)

Display a basic organization page that includes the list of all of the repos owned by that organiztion

Repository Page

User can click on a repository link from the Organization page and see information about that repository

Link to BDFiddle to try out tool

Repo page shows a link to BDFiddle where the user can try out a tool

Tags

Repo page interrogates GitHub for list of tags and displays them

Initial Skills

Prioritise these, port to the new Pyclowder and deploy to Tools Catalog

- Langid
- DBPedia
- Census From Cell
- Handwritten Decimals
- Killed Photos
- Mean Grey
- Faces
- Eyes
- Profiles
- Closeups
- NLTK Summary
- Stanford CoreNLP
- Tesseract
- Tika
- Versus
- VLFeat
- Generalized exif/image metadata extractor

Scientific Communities to be Seeded in Tools Catalog

- OpenCV - Grad student to curate?
- Critical Zone Observatory (via ESIP?)
- Data Driven Ag
- Bisque - Counting Cells in a microscope image??
- Cosmology