Register Extractors by Space

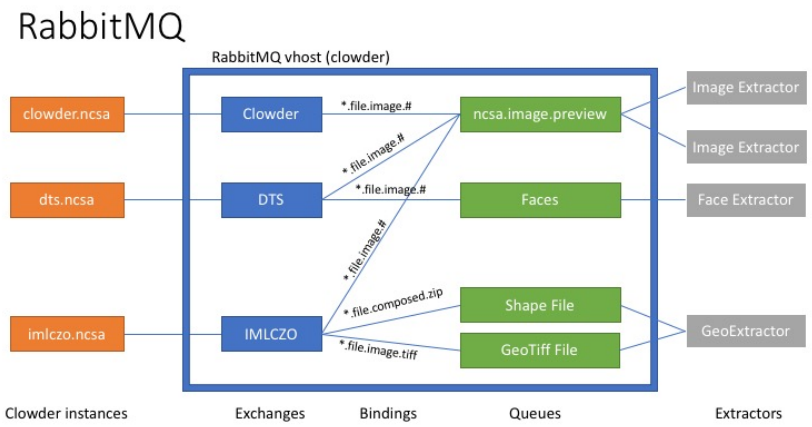
The ability to track extractors by space is already in master. Go to a project space and click on "Extractors."

Next we need to send the message over rabbitmq to that specific extractor. Since we have the ability to do this now by extractors this part should be easy.

Questions

- Should extractors now default to only bind themselves to extractor specific submission? And not on mime type?
- Should we have the ability to register extractors at the instance level if that's the case?
- Are list of extractors retrieved from RabbitMQ or from the list of registered extractors?
- Does any file get submitted to an extractor if submitted to a space?

Clowder and RabbitMQ



DISCUSSION DRAFT

Following are some steps that we can do to create a per space extractor. The thinking of this list is that this is path to implement.

1. List of extractors is kept per space / clowder instance
    - When a file is uploaded to clowder it will trigger the normal message to rabbitmq, however at the same time in case of a file it will do a lookup of what space(s) the file belongs to and will trigger a message to the normal exchange to each extractor explicitly listed.
    - quick implementation, only need to add logic to rabbitmq plugin to find the space and send additional messages, however can result in duplicate messages send to a queue, for example might send a message because binding image/* preview, as well as explicit space to that extractor
    - Assumption, all extractors register themselves with that specific clowder instance, as well as with the exchange (this is assumed now as well).
2. List of bindings is kept in clowder, removes exchanges
    - When a file is uploaded to clowder, it will look in the global bindings as well as the space bindings and makes a unique list of all extractors and fire a message for each extractor
    - all logic of which extractor is now controlled by clowder, no duplicate messages
    - Assumption, all extractors register themselves with that specific clowder instance (this is assumed now as well).
3. List of extractors is queried
    a. Extractors will bind themselves to a queue and have a command queue. The command queue is non-persistant. Each extractor will pick up messages from both queues, however give preference to the command queue. Clowder will send a message to the extractor exchange and send a command message (cmd) that is picked up by all command queues, allowing them to register them than with that specific clowder instance. All extractor.* messages will go the normal extractor queue.
    b. Can easily get a list of all extractors and refresh this list every 15 or so minutes.
4. More complex logic for extractors
    a. Extractors can now have more complex logic, such as (file added, mimetype=image/jpg, filesize>5MB). This is part of the extractor_info. json, clowder can use this more complex logic to see if an extractor should fire.

Bonus

1. Adding code so we use the key for the user that is responsible for the event instead of the global key.

UPDATE 2018-05-11

- Need to add global list for extractors, use the same mongo collection for global extractors
- Re-visit per space list of extractors

- RabbitMQ plugin needs to know what space a file belongs to, it will need to know what space
    - check list of extractors in space / global
    - Use mimetype to filter list (be ready for more complex rules in future)
- when we move dataset into new space, run all extactors on files in space
- private space for now only uses global
- Keep exchange + routing key for now mark as deprecated, remove in 2.0
- new extractors, removing bindings to exchange

WORK:

- rob does pyclowder 2 + simple python code for others
- mike does UI for extractor selection
- luigi/rob do clowder things