KU LEUVEN

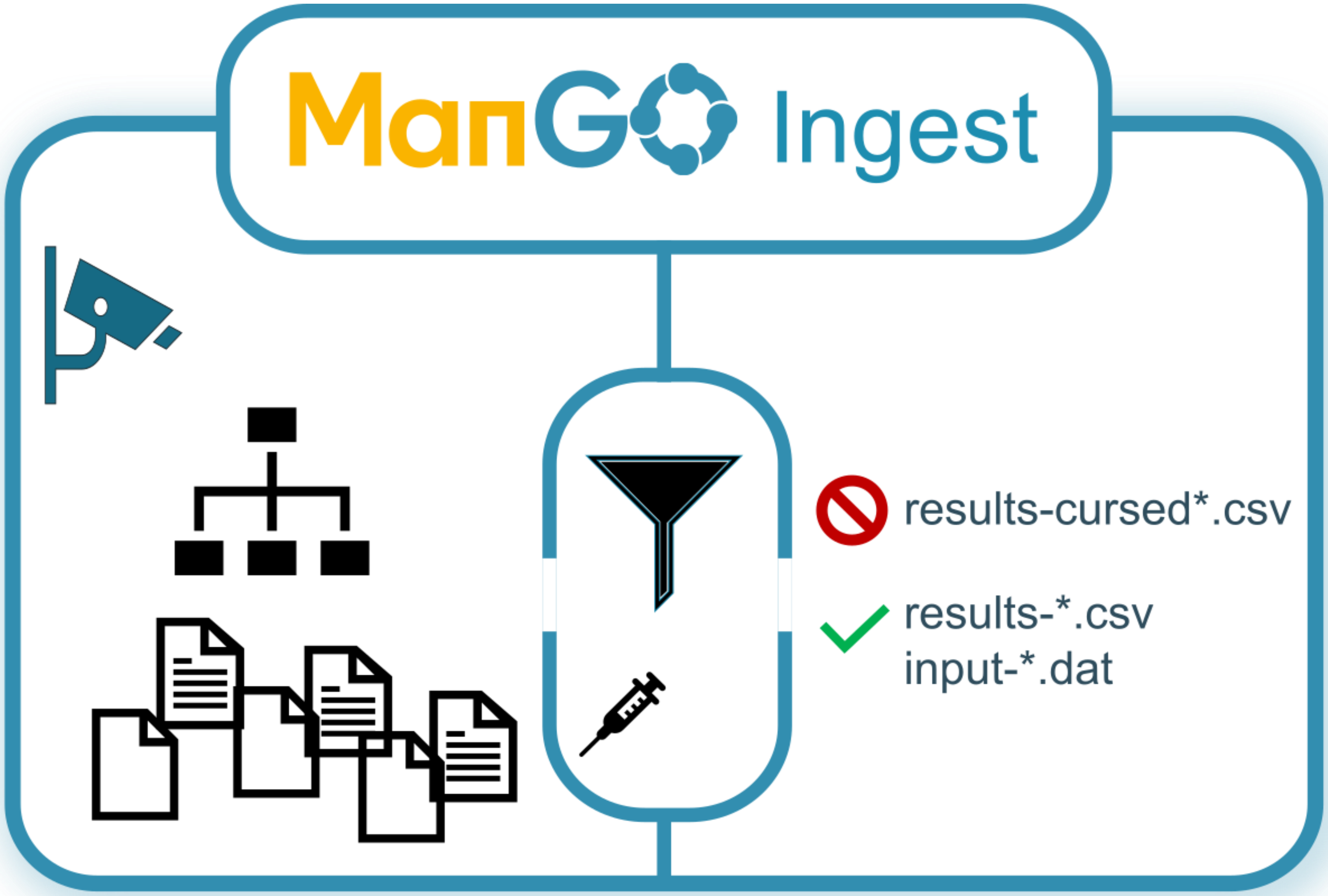
ManGO platform updates

ManGO Ingest
ManGO Flow
ManGO Portal

Paul Borgermans
and FOZ-RDM ICTS – KU Leuven

UGM 2025
iRODS®
DURHAM, NC

ManGO Ingest
ManGO
iRODS
ManGO Flow



ManGO Ingest

🚫 results-cursed*.csv

✓ results-*.csv
input-*.dat

ManGO Ingest

✓

Pure 🐍

No cronjob needed

Can be used to sync ad-hoc

Powerful path-based filtering

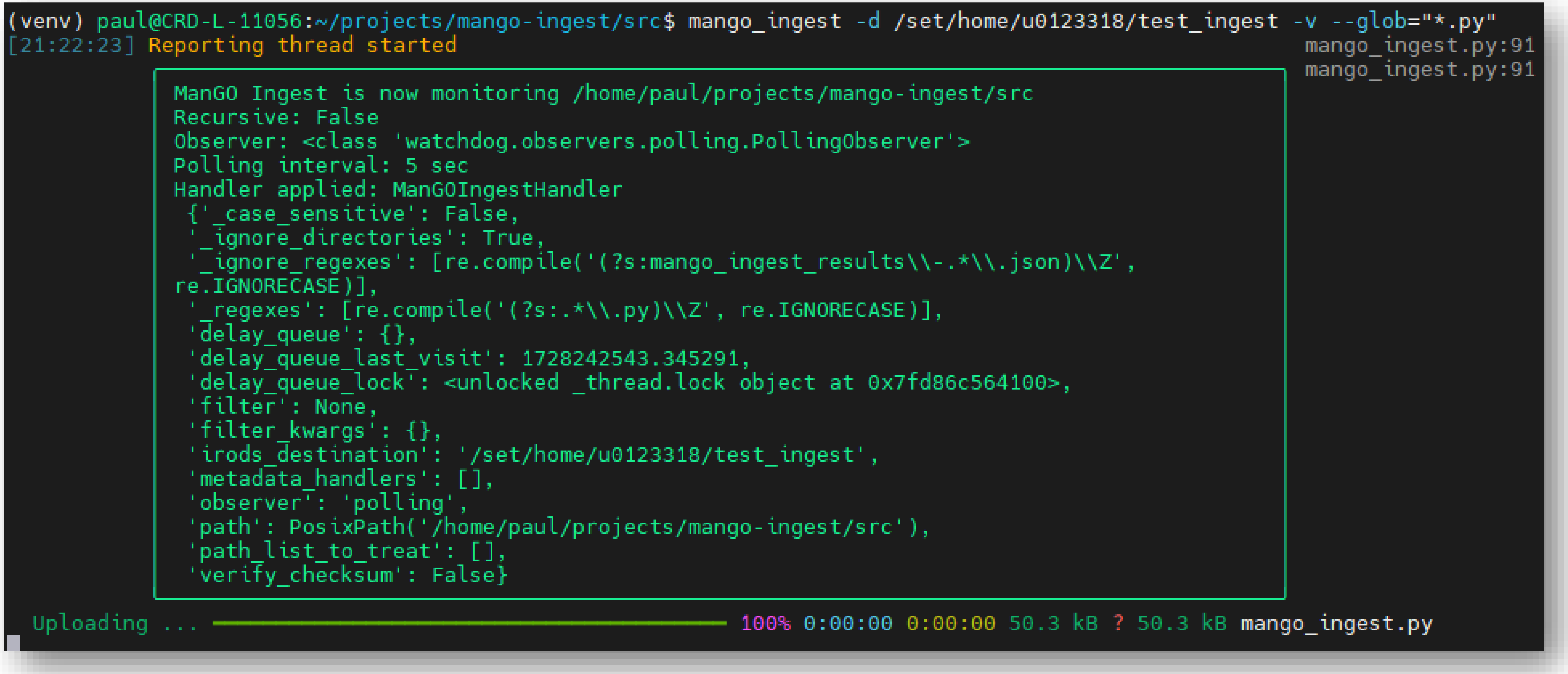
Automatic metadata extraction

Extensible with project specific filtering and metadata extraction (eg ExifTool included) +

!

Designed as a command line tool

ManGO Ingest

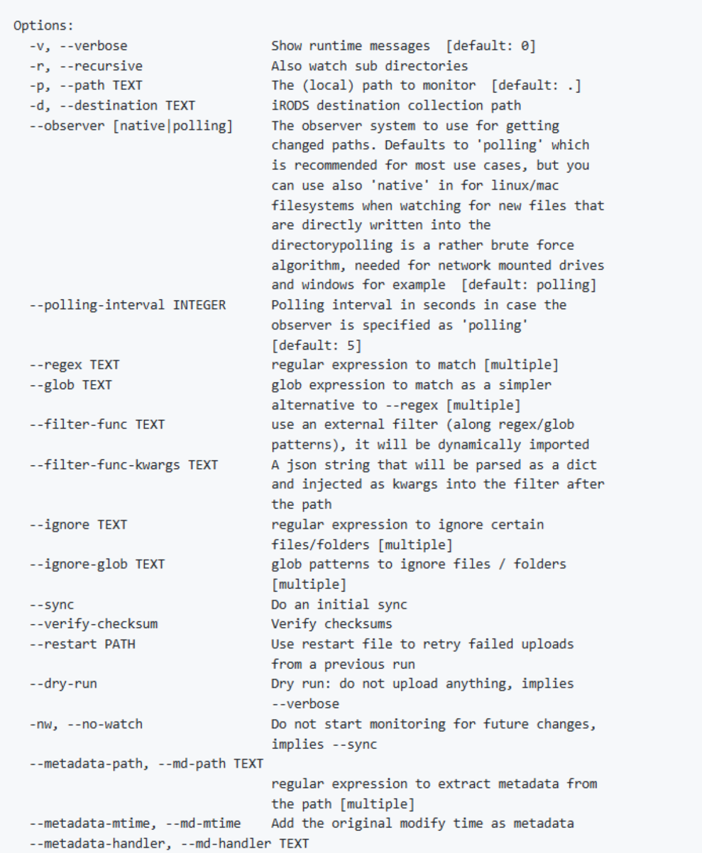


```
(venv) paul@CRD-L-11056:~/projects/mango-ingest/src$ mango_ingest -d /set/home/u0123318/test_ingest -v --glob="*.py"
[21:22:23] Reporting thread started                                                                   mango_ingest.py:91
                                                                                                      mango_ingest.py:91
           ┌────────────────────────────────────────────────────────────────────────────────────┐
           │ ManGO Ingest is now monitoring /home/paul/projects/mango-ingest/src                  │
           │ Recursive: False                                                                     │
           │ Observer: <class 'watchdog.observers.polling.PollingObserver'>                       │
           │ Polling interval: 5 sec                                                              │
           │ Handler applied: ManGOIngestHandler                                                  │
           │  {'_case_sensitive': False,                                                          │
           │  '_ignore_directories': True,                                                        │
           │  '_ignore_regexes': [re.compile('(?s:mango_ingest_results\\-.*\\.json)\\Z',          │
           │ re.IGNORECASE)],                                                                     │
           │  '_regexes': [re.compile('(?s:.*\\.py)\\Z', re.IGNORECASE)],                         │
           │  'delay_queue': {},                                                                  │
           │  'delay_queue_last_visit': 1728242543.345291,                                        │
           │  'delay_queue_lock': <unlocked _thread.lock object at 0x7fd86c564100>,               │
           │  'filter': None,                                                                     │
           │  'filter_kwargs': {},                                                                │
           │  'irods_destination': '/set/home/u0123318/test_ingest',                              │
           │  'metadata_handlers': [],                                                            │
           │  'observer': 'polling',                                                              │
           │  'path': PosixPath('/home/paul/projects/mango-ingest/src'),                          │
           │  'path_list_to_treat': [],                                                           │
           │  'verify_checksum': False}                                                           │
           └────────────────────────────────────────────────────────────────────────────────────┘
  Uploading ... ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% 0:00:00 0:00:00 50.3 kB ? 50.3 kB mango_ingest.py
```

https://github.com/kuleuven/mango-ingest

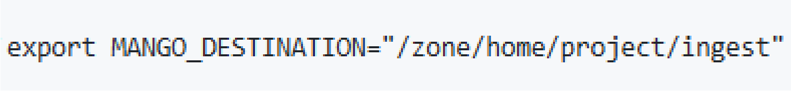
Controlling …

- Command line options



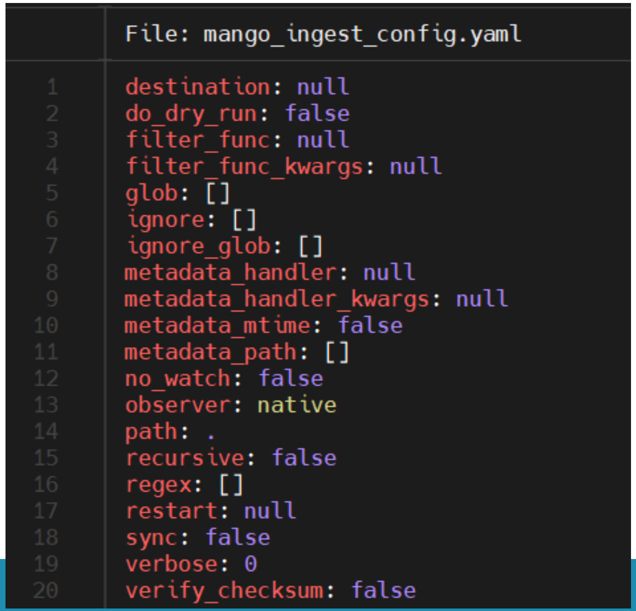
```
Options:
  -v, --verbose                  Show runtime messages  [default: 0]
  -r, --recursive                Also watch sub directories
  -p, --path TEXT                The (local) path to monitor  [default: .]
  -d, --destination TEXT         iRODS destination collection path
  --observer [native|polling]    The observer system to use for getting
                                 changed paths. Defaults to 'polling' which
                                 is recommended for most use cases, but you
                                 can use also 'native' in for linux/mac
                                 filesystems when watching for new files that
                                 are directly written into the
                                 directorypolling is a rather brute force
                                 algorithm, needed for network mounted drives
                                 and windows for example  [default: polling]
  --polling-interval INTEGER     Polling interval in seconds in case the
                                 observer is specified as 'polling'
                                 [default: 5]
  --regex TEXT                   regular expression to match [multiple]
  --glob TEXT                    glob expression to match as a simpler
                                 alternative to --regex [multiple]
  --filter-func TEXT             use an external filter (along regex/glob
                                 patterns), it will be dynamically imported
  --filter-func-kwargs TEXT      A json string that will be parsed as a dict
                                 and injected as kwargs into the filter after
                                 the path
  --ignore TEXT                  regular expression to ignore certain
                                 files/folders [multiple]
  --ignore-glob TEXT             glob patterns to ignore files / folders
                                 [multiple]
  --sync                         Do an initial sync
  --verify-checksum              Verify checksums
  --restart PATH                 Use restart file to retry failed uploads
                                 from a previous run
  --dry-run                      Dry run: do not upload anything, implies
                                 --verbose
  -nw, --no-watch                Do not start monitoring for future changes,
                                 implies --sync
  --metadata-path, --md-path TEXT
                                 regular expression to extract metadata from
                                 the path [multiple]
  --metadata-mtime, --md-mtime   Add the original modify time as metadata
  --metadata-handler, --md-handler TEXT
```

- Environment variables



```
export MANGO_DESTINATION="/zone/home/project/ingest"
```

- Configuration file



File: mango_ingest_config.yaml

```yaml
destination: null
do_dry_run: false
filter_func: null
filter_func_kwargs: null
glob: []
ignore: []
ignore_glob: []
metadata_handler: null
metadata_handler_kwargs: null
metadata_mtime: false
metadata_path: []
no_watch: false
observer: native
path: .
recursive: false
regex: []
restart: null
sync: false
verbose: 0
verify_checksum: false
```

ManGO Flow

Automation: rules or ManGO Flow?

Rules

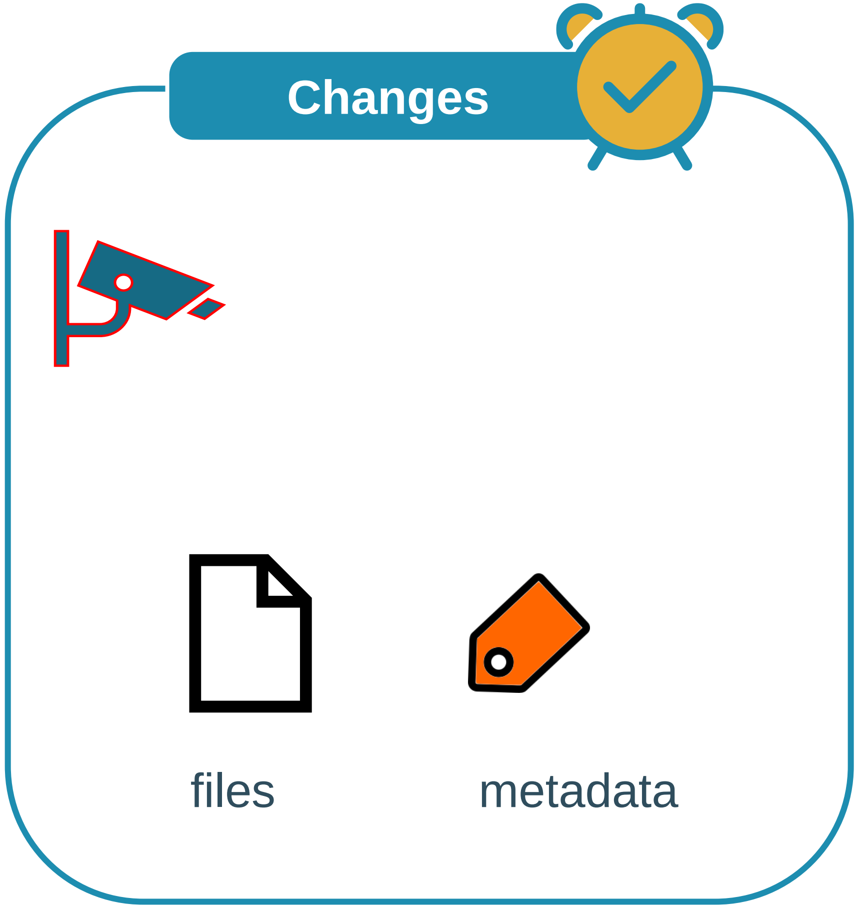
iRODS®

- ✓ Internal events
- ✓ “Pre” policy enforcement
- ! Not meant for heavy tasks

ManGO Flow

- ✓ Built on Celery
- ✓ Scalable
- ✓ Build workflows through composition
- ! No “pre” policy enforcement

Triggers: launching tasks



Changes

files

metadata

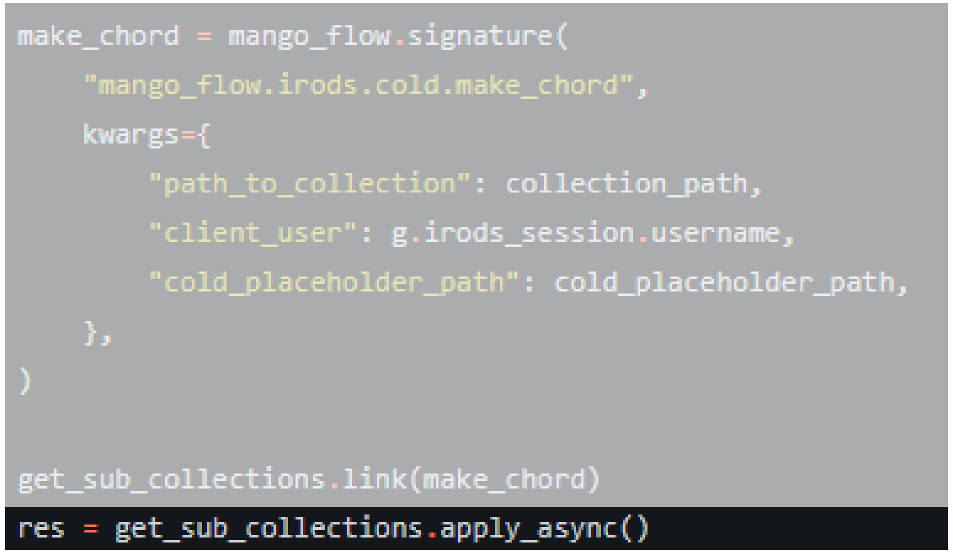
Direct actions

KU LEUVEN
ManGO
set
Submit

custom scripts, PEP, rules

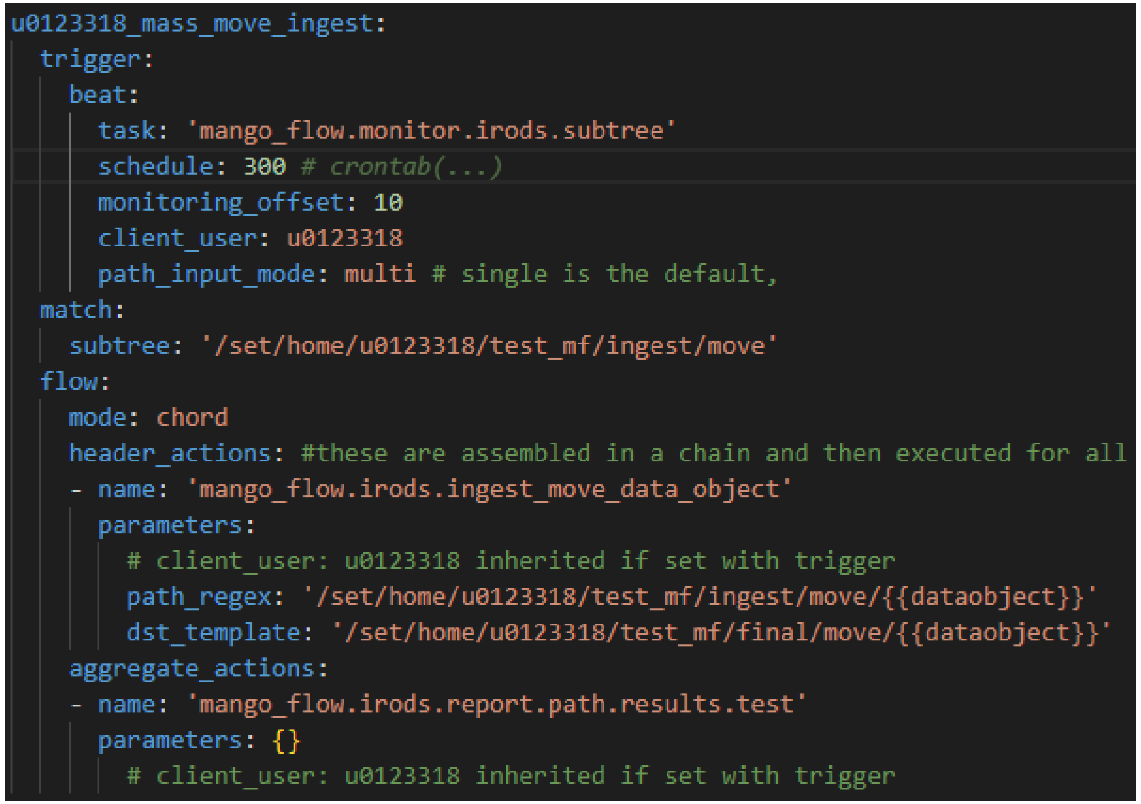
Creating workflows

- In code



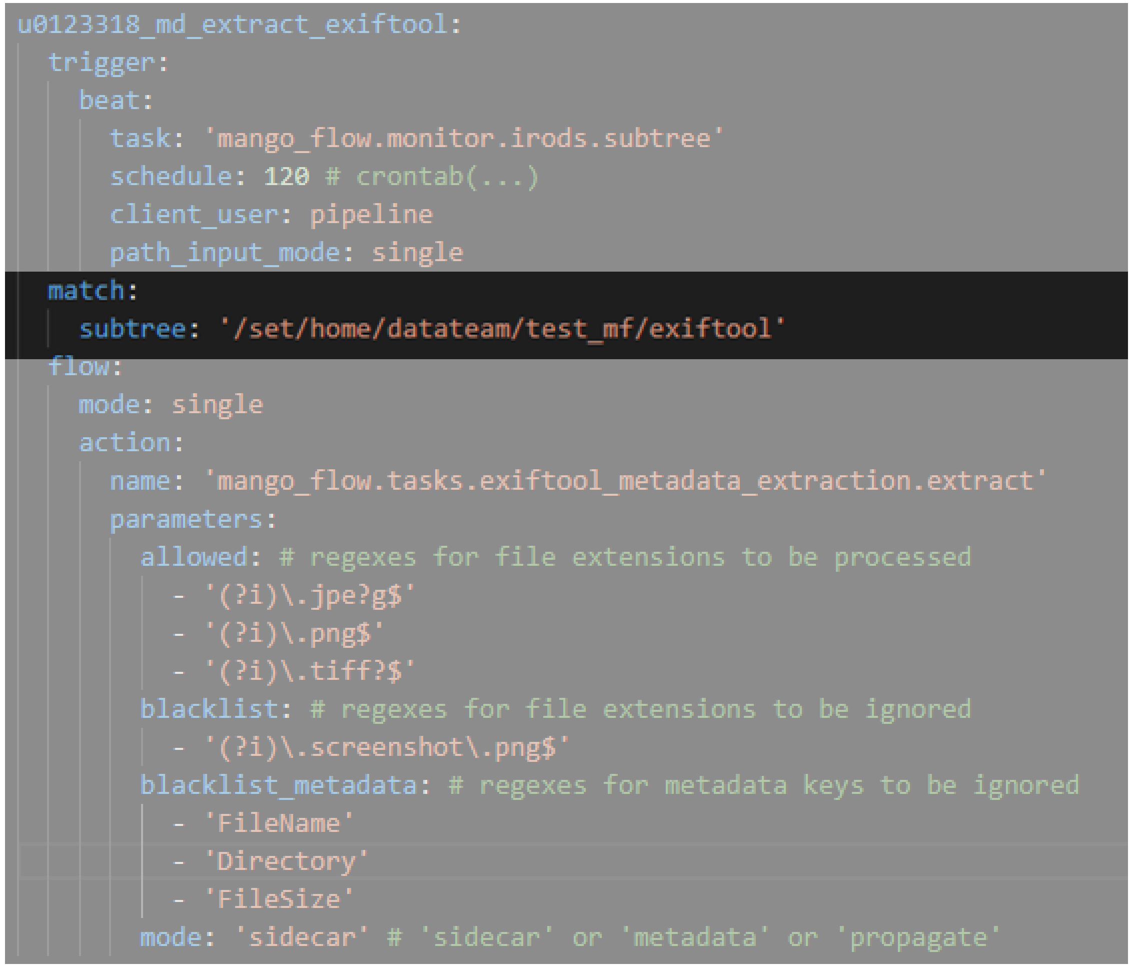
```
make_chord = mango_flow.signature(
    "mango_flow.irods.cold.make_chord",
    kwargs={
        "path_to_collection": collection_path,
        "client_user": g.irods_session.username,
        "cold_placeholder_path": cold_placeholder_path,
    },
)

get_sub_collections.link(make_chord)
res = get_sub_collections.apply_async()
```

- Declarative

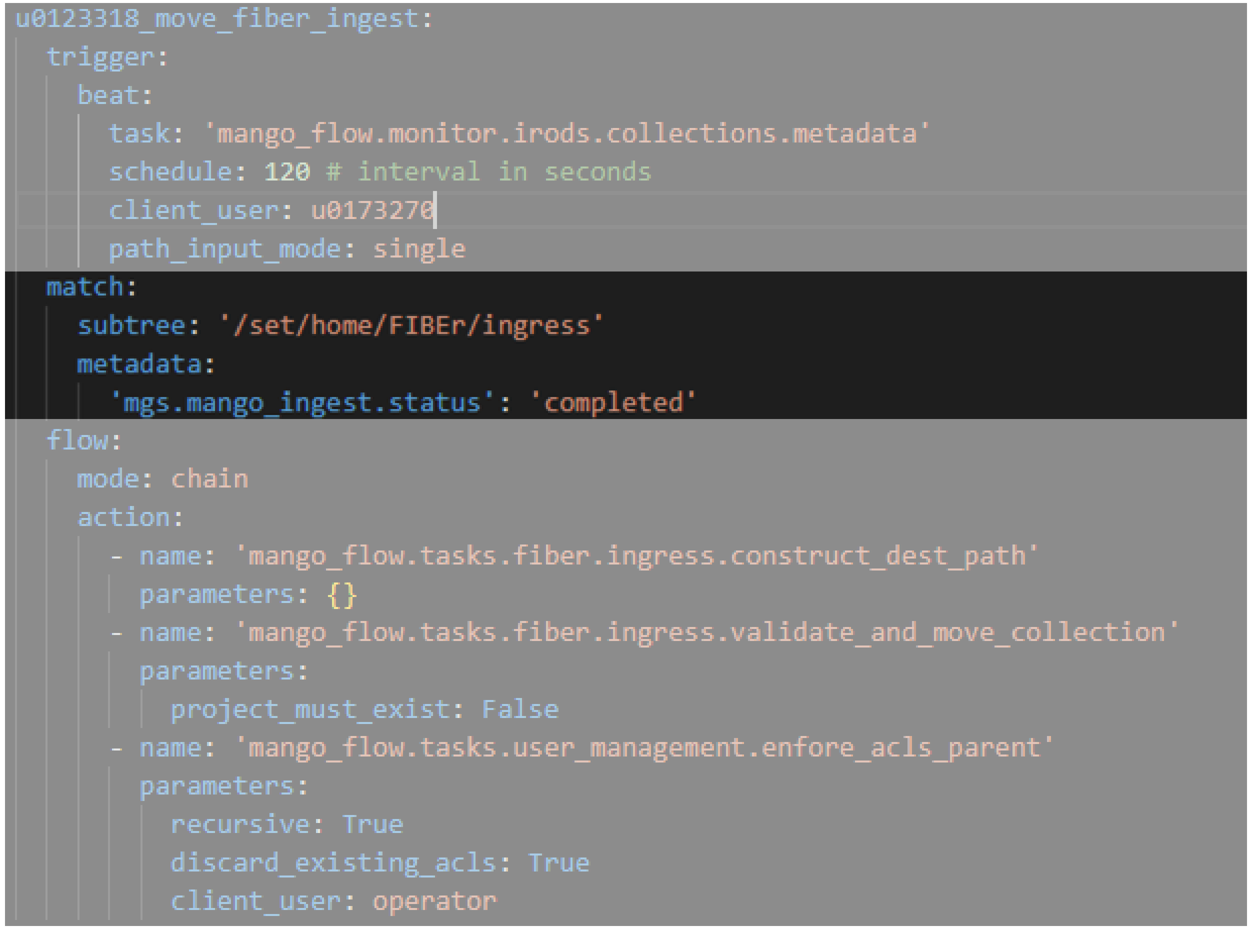


```
u0123318_mass_move_ingest:
  trigger:
    beat:
      task: 'mango_flow.monitor.irods.subtree'
      schedule: 300 # crontab(...)
      monitoring_offset: 10
      client_user: u0123318
      path_input_mode: multi # single is the default,
  match:
    subtree: '/set/home/u0123318/test_mf/ingest/move'
  flow:
    mode: chord
    header_actions: #these are assembled in a chain and then executed for all
    - name: 'mango_flow.irods.ingest_move_data_object'
      parameters:
        # client_user: u0123318 inherited if set with trigger
        path_regex: '/set/home/u0123318/test_mf/ingest/move/{{dataobject}}'
        dst_template: '/set/home/u0123318/test_mf/final/move/{{dataobject}}'
    aggregate_actions:
    - name: 'mango_flow.irods.report.path.results.test'
      parameters: {}
        # client_user: u0123318 inherited if set with trigger
```

Conditions

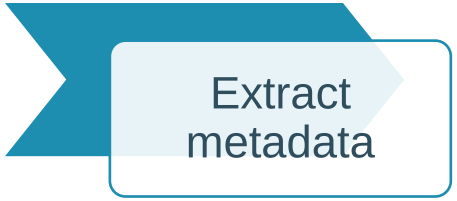


```yaml
u0123318_md_extract_exiftool:
  trigger:
    beat:
      task: 'mango_flow.monitor.irods.subtree'
      schedule: 120 # crontab(...)
      client_user: pipeline
      path_input_mode: single
  match:
    subtree: '/set/home/datateam/test_mf/exiftool'
  flow:
    mode: single
    action:
      name: 'mango_flow.tasks.exiftool_metadata_extraction.extract'
      parameters:
        allowed: # regexes for file extensions to be processed
          - '(?i)\.jpe?g$'
          - '(?i)\.png$'
          - '(?i)\.tiff?$'
        blacklist: # regexes for file extensions to be ignored
          - '(?i)\.screenshot\.png$'
        blacklist_metadata: # regexes for metadata keys to be ignored
          - 'FileName'
          - 'Directory'
          - 'FileSize'
        mode: 'sidecar' # 'sidecar' or 'metadata' or 'propagate'
```

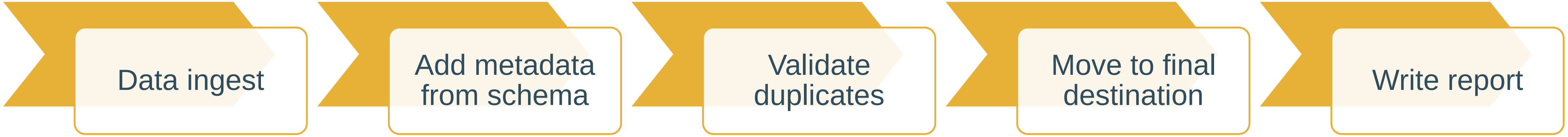


```yaml
u0123318_move_fiber_ingest:
  trigger:
    beat:
      task: 'mango_flow.monitor.irods.collections.metadata'
      schedule: 120 # interval in seconds
      client_user: u0173270
      path_input_mode: single
  match:
    subtree: '/set/home/FIBEr/ingress'
    metadata:
      'mgs.mango_ingest.status': 'completed'
  flow:
    mode: chain
    action:
      - name: 'mango_flow.tasks.fiber.ingress.construct_dest_path'
        parameters: {}
      - name: 'mango_flow.tasks.fiber.ingress.validate_and_move_collection'
        parameters:
          project_must_exist: False
      - name: 'mango_flow.tasks.user_management.enfore_acls_parent'
        parameters:
          recursive: True
          discard_existing_acls: True
          client_user: operator
```

Declarative: linear composition



Extract metadata



Data ingest

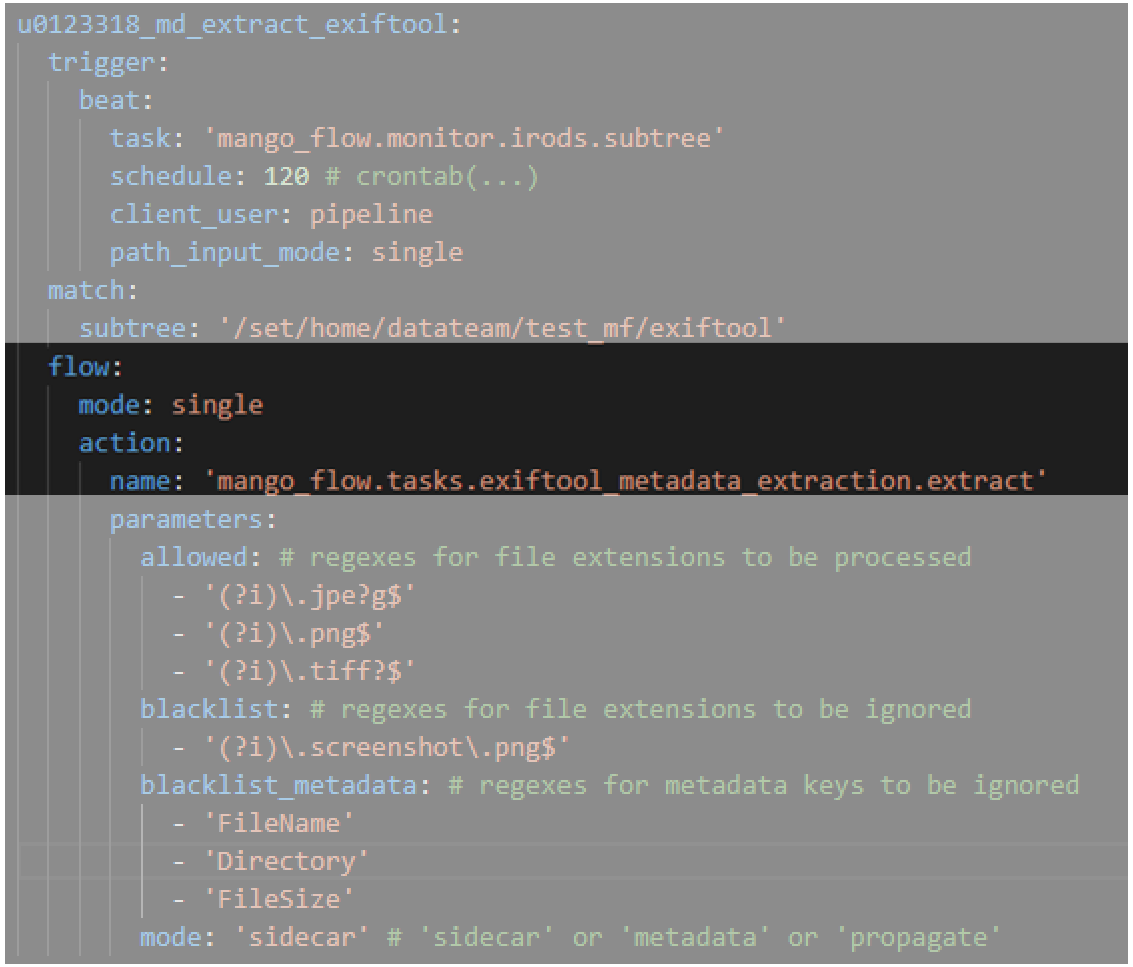
Add metadata from schema

Validate duplicates

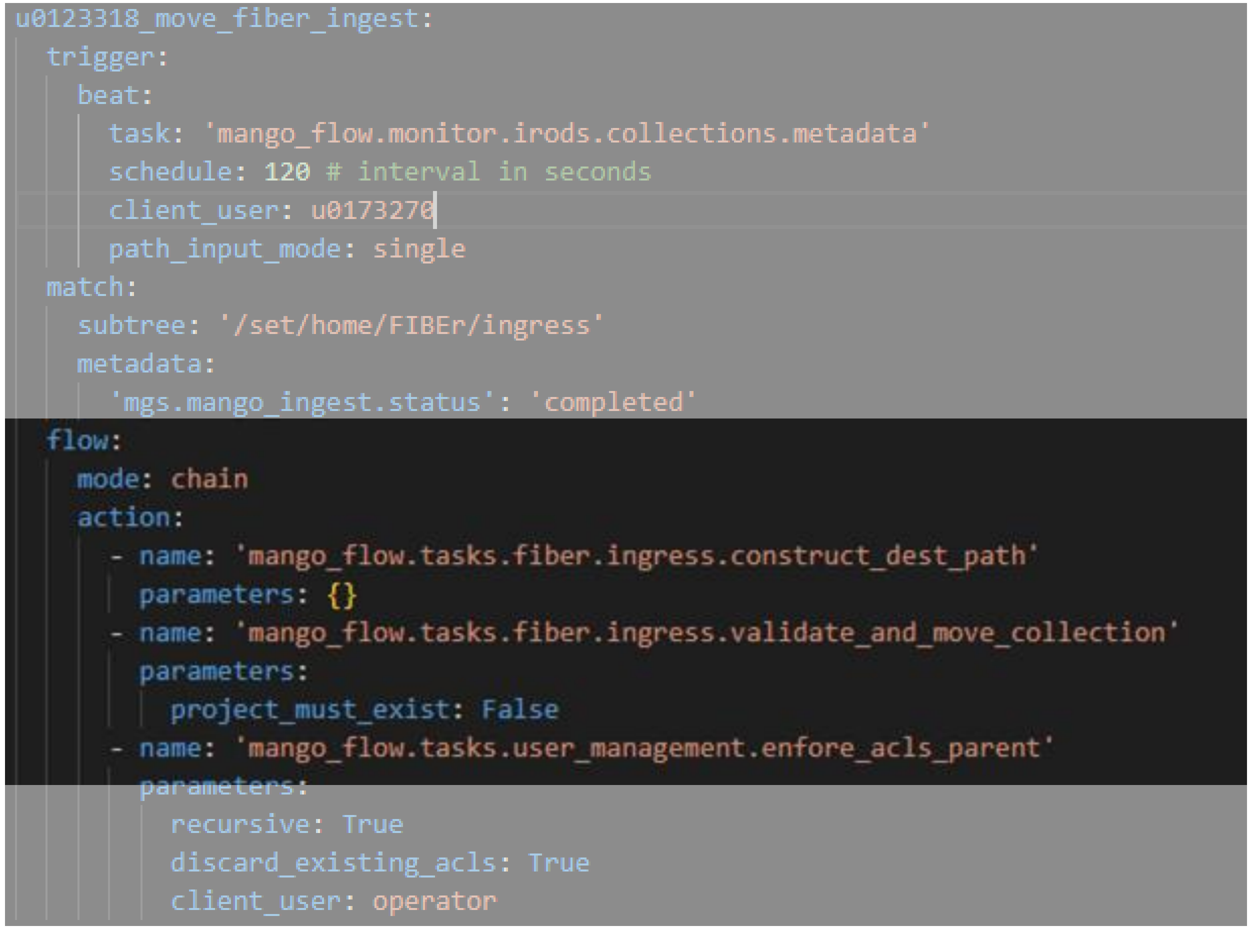
Move to final destination

Write report

Declarative syntax

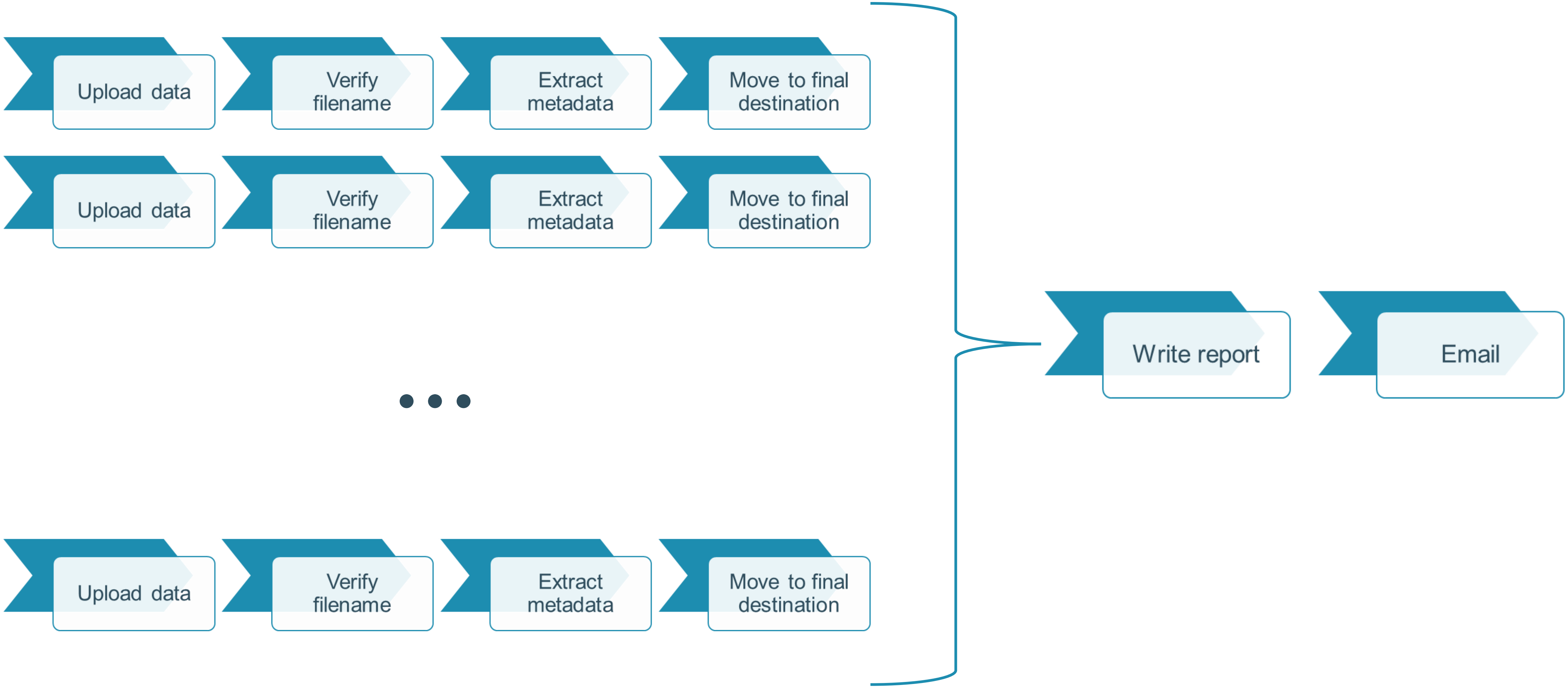


```
u0123318_md_extract_exiftool:
  trigger:
    beat:
      task: 'mango_flow.monitor.irods.subtree'
      schedule: 120 # crontab(...)
      client_user: pipeline
      path_input_mode: single
  match:
    subtree: '/set/home/datateam/test_mf/exiftool'
  flow:
    mode: single
    action:
      name: 'mango_flow.tasks.exiftool_metadata_extraction.extract'
      parameters:
        allowed: # regexes for file extensions to be processed
          - '(?i)\.jpe?g$'
          - '(?i)\.png$'
          - '(?i)\.tiff?$'
        blacklist: # regexes for file extensions to be ignored
          - '(?i)\.screenshot\.png$'
        blacklist_metadata: # regexes for metadata keys to be ignored
          - 'FileName'
          - 'Directory'
          - 'FileSize'
        mode: 'sidecar' # 'sidecar' or 'metadata' or 'propagate'
```



```
u0123318_move_fiber_ingest:
  trigger:
    beat:
      task: 'mango_flow.monitor.irods.collections.metadata'
      schedule: 120 # interval in seconds
      client_user: u0173270
      path_input_mode: single
  match:
    subtree: '/set/home/FIBEr/ingress'
    metadata:
      'mgs.mango_ingest.status': 'completed'
  flow:
    mode: chain
    action:
      - name: 'mango_flow.tasks.fiber.ingress.construct_dest_path'
        parameters: {}
      - name: 'mango_flow.tasks.fiber.ingress.validate_and_move_collection'
        parameters:
          project_must_exist: False
      - name: 'mango_flow.tasks.user_management.enfore_acls_parent'
        parameters:
          recursive: True
          discard_existing_acls: True
          client_user: operator
```

Parallel and aggregate



Upload data

Verify filename

Extract metadata

Move to final destination

Upload data

Verify filename

Extract metadata

Move to final destination

• • •

Upload data

Verify filename

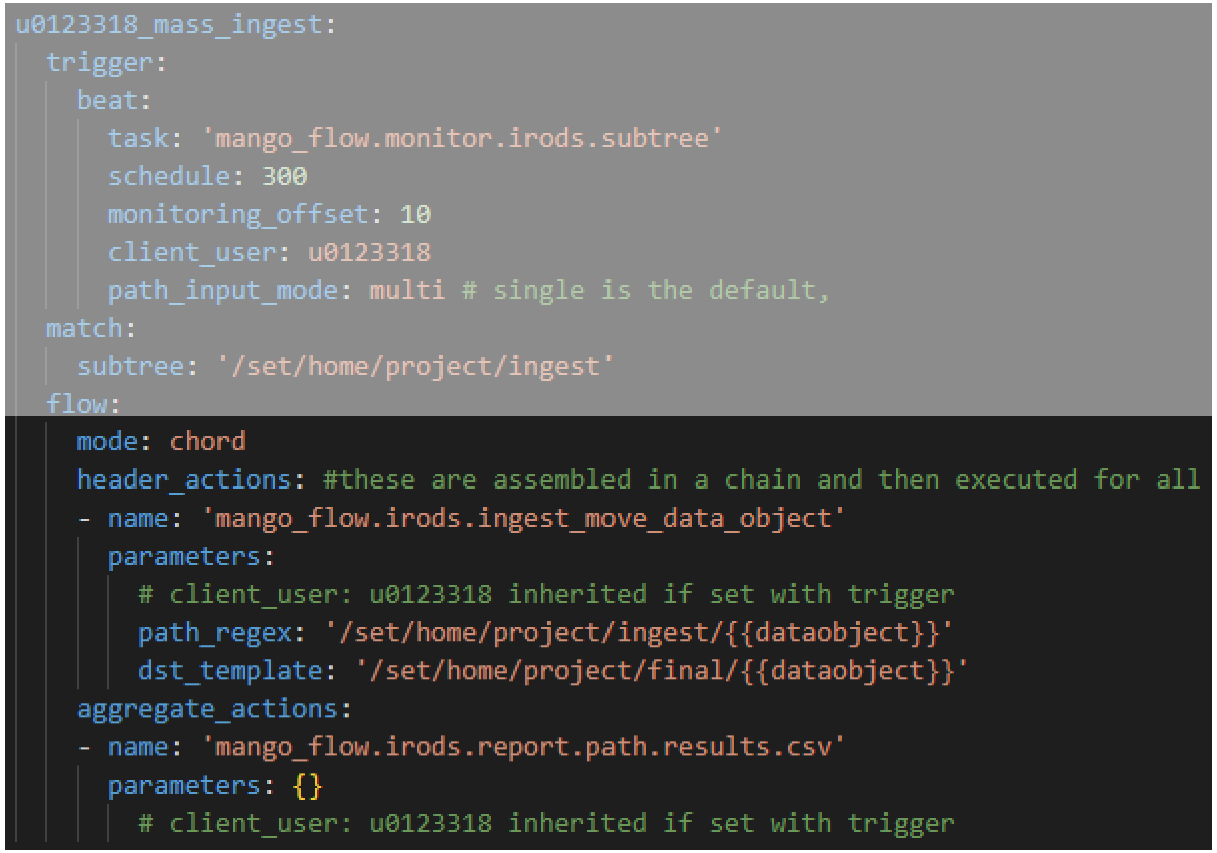
Extract metadata

Move to final destination

Write report

Email

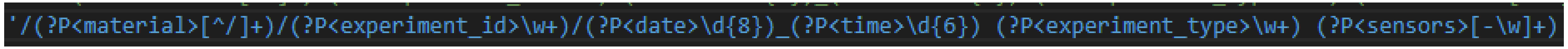
Parallel and aggregate



```yaml
u0123318_mass_ingest:
  trigger:
    beat:
      task: 'mango_flow.monitor.irods.subtree'
      schedule: 300
      monitoring_offset: 10
      client_user: u0123318
      path_input_mode: multi # single is the default,
  match:
    subtree: '/set/home/project/ingest'
  flow:
    mode: chord
    header_actions: #these are assembled in a chain and then executed for all
    - name: 'mango_flow.irods.ingest_move_data_object'
      parameters:
        # client_user: u0123318 inherited if set with trigger
        path_regex: '/set/home/project/ingest/{{dataobject}}'
        dst_template: '/set/home/project/final/{{dataobject}}'
    aggregate_actions:
    - name: 'mango_flow.irods.report.path.results.csv'
      parameters: {}
        # client_user: u0123318 inherited if set with trigger
```

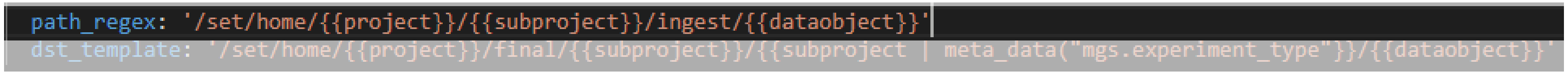
Pattern matching & composition

- Matching:
  - Full regex



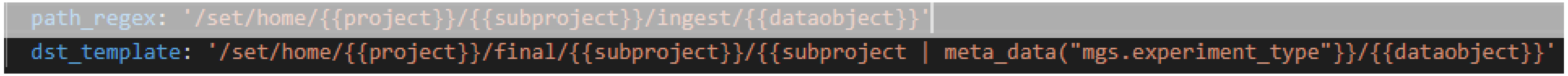
```
'/(?P<material>[^/]+)/(?P<experiment_id>\w+)/(?P<date>\d{8})_(?P<time>\d{6}) (?P<experiment_type>\w+) (?P<sensors>[-\w]+)
```

  - Mustache syntax



```
path_regex: '/set/home/{{project}}/{{subproject}}/ingest/{{dataobject}}'
dst_template: '/set/home/{{project}}/final/{{subproject}}/{{subproject | meta_data("mgs.experiment_type"}}/{{dataobject}}'
```

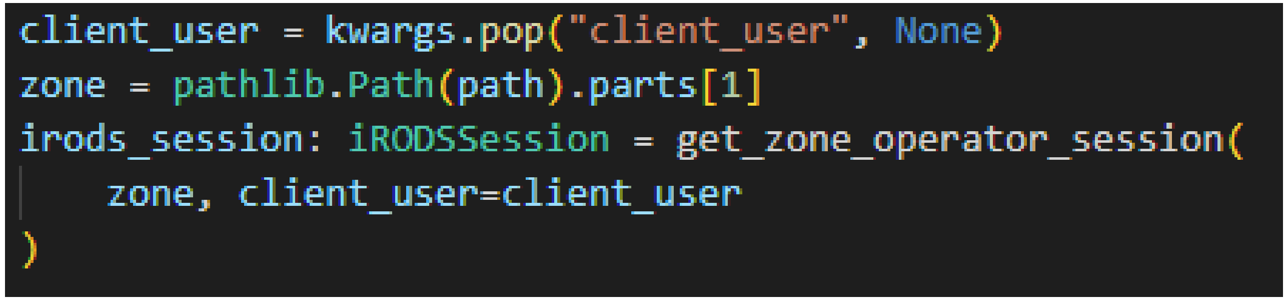
- Composition: Jinja2 template strings



```
path_regex: '/set/home/{{project}}/{{subproject}}/ingest/{{dataobject}}'
dst_template: '/set/home/{{project}}/final/{{subproject}}/{{subproject | meta_data("mgs.experiment_type"}}/{{dataobject}}'
```

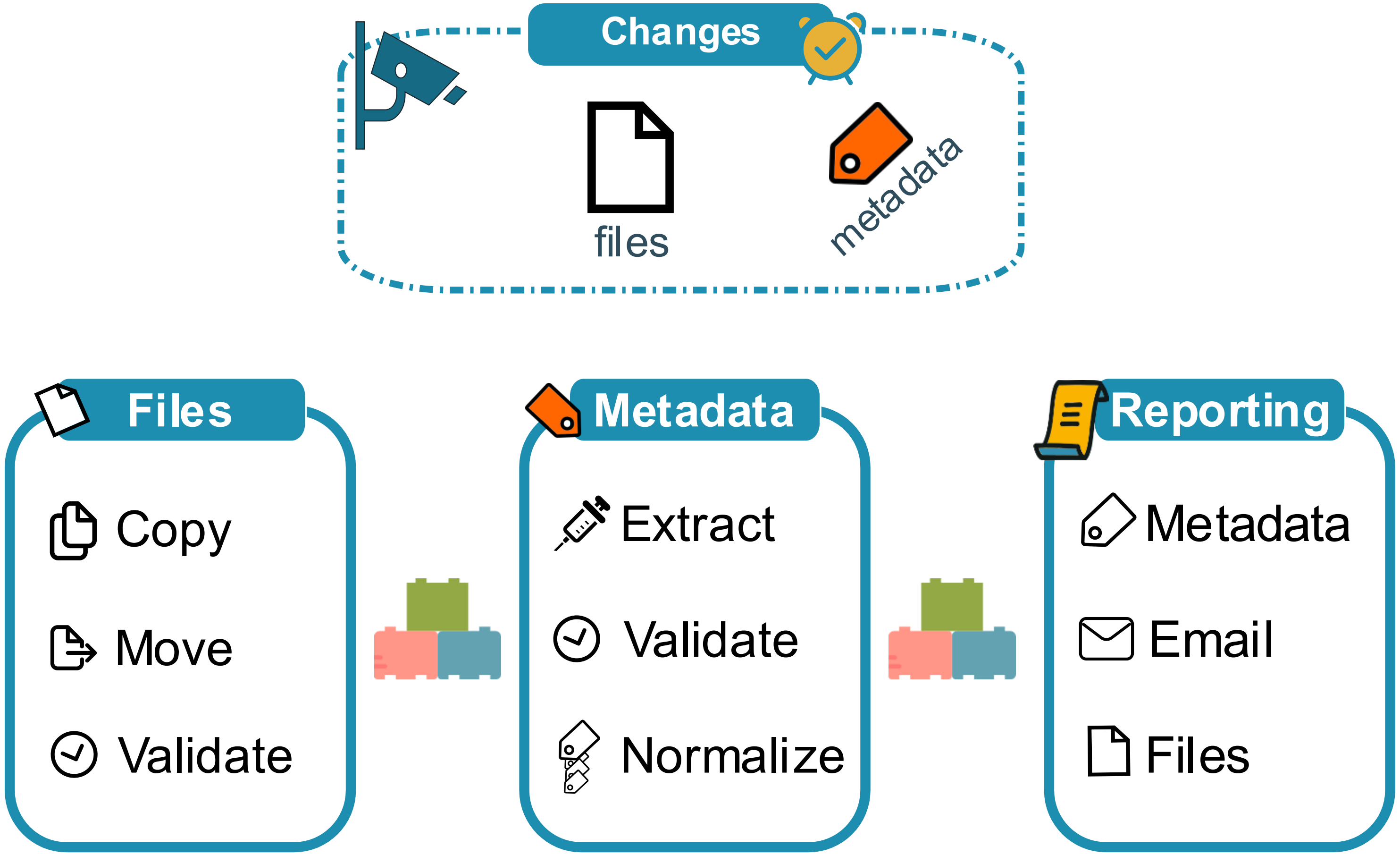
ManGO Flow: client_user?

- User or ‘operator’ (rodsadmin)
- With strict mode:
  - Depends on the operations (metadata, move, …)
  - ‘pipeline’ account
  - Regular account (‘owner’)



```
client_user = kwargs.pop("client_user", None)
zone = pathlib.Path(path).parts[1]
irods_session: iRODSSession = get_zone_operator_session(
    zone, client_user=client_user
)
```

ManGO Flow



Changes

files

metadata

Files

- Copy
- Move
- Validate

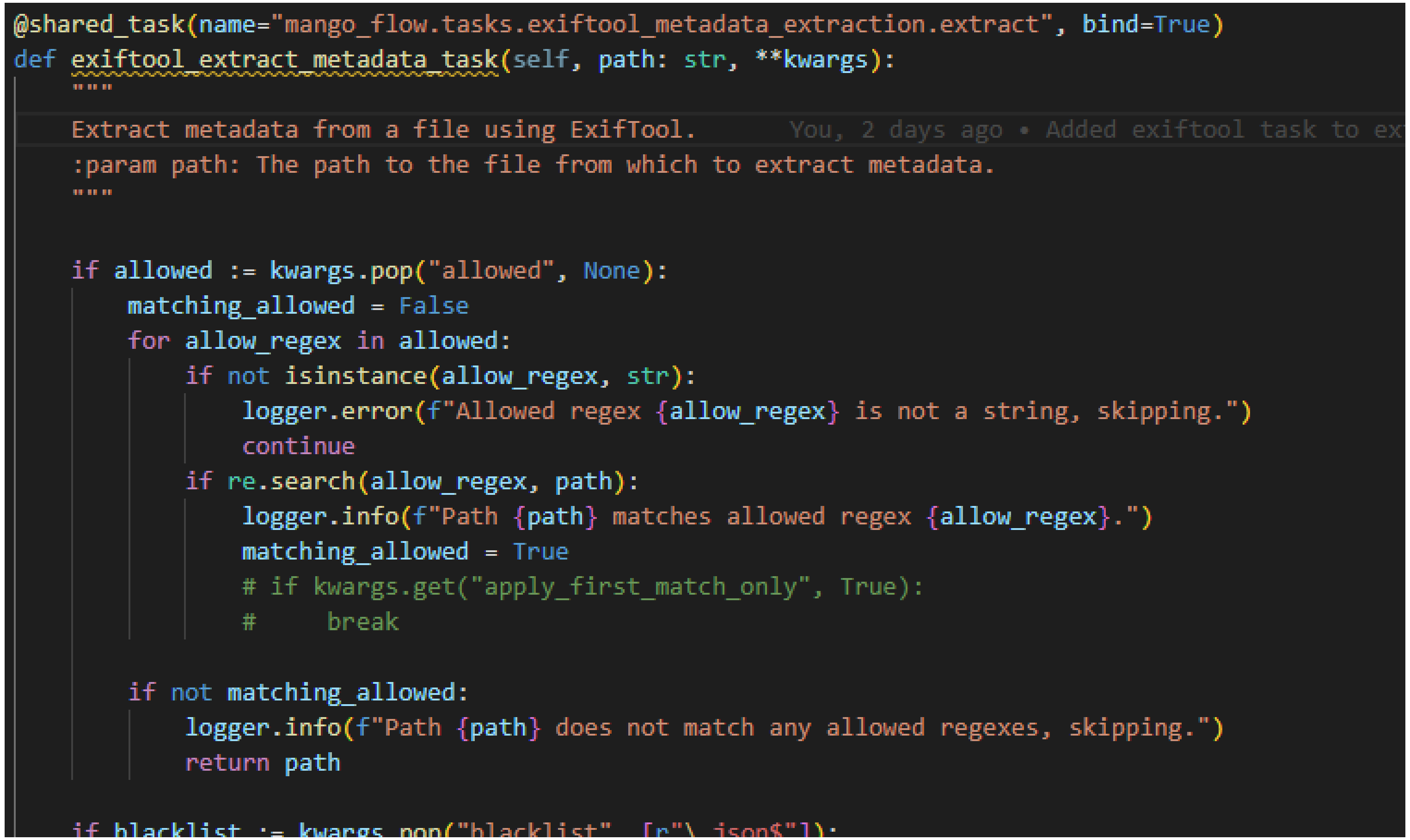
Metadata

- Extract
- Validate
- Normalize

Reporting

- Metadata
- Email
- Files

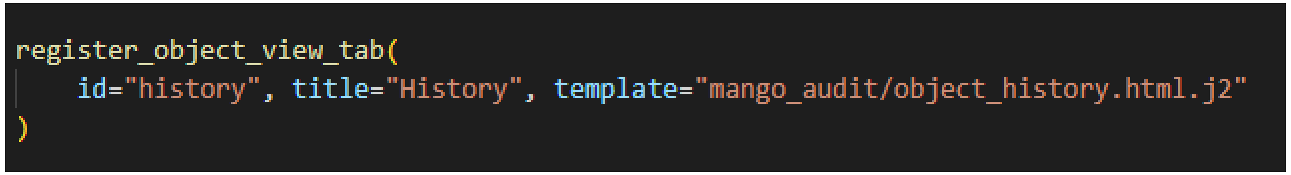
Create additional tasks



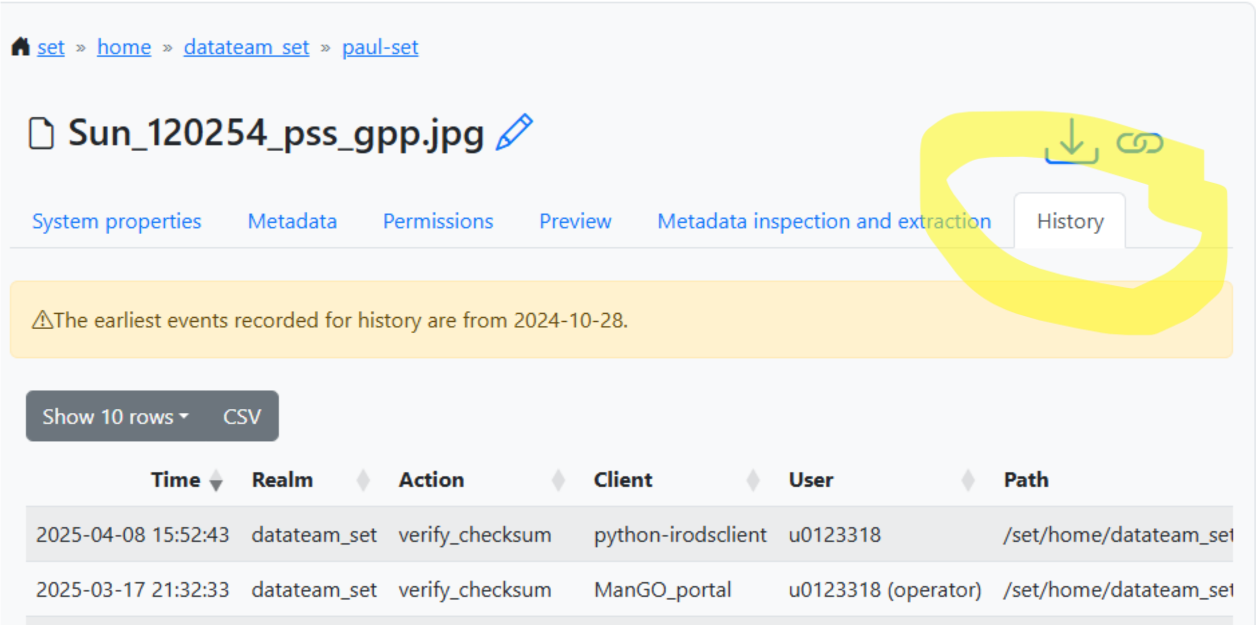
```
@shared_task(name="mango_flow.tasks.exiftool_metadata_extraction.extract", bind=True)
def exiftool_extract_metadata_task(self, path: str, **kwargs):
    """
    Extract metadata from a file using ExifTool.      You, 2 days ago • Added exiftool task to ex
    :param path: The path to the file from which to extract metadata.
    """

    if allowed := kwargs.pop("allowed", None):
        matching_allowed = False
        for allow_regex in allowed:
            if not isinstance(allow_regex, str):
                logger.error(f"Allowed regex {allow_regex} is not a string, skipping.")
                continue
            if re.search(allow_regex, path):
                logger.info(f"Path {path} matches allowed regex {allow_regex}.")
                matching_allowed = True
                # if kwargs.get("apply_first_match_only", True):
                #     break

        if not matching_allowed:
            logger.info(f"Path {path} does not match any allowed regexes, skipping.")
            return path

    if blacklist := kwargs.pop("blacklist", [r"\.json$"]):
```

ManGO Portal

- Much more extensible



```
register_object_view_tab(
    id="history", title="History", template="mango_audit/object_history.html.j2"
)
```



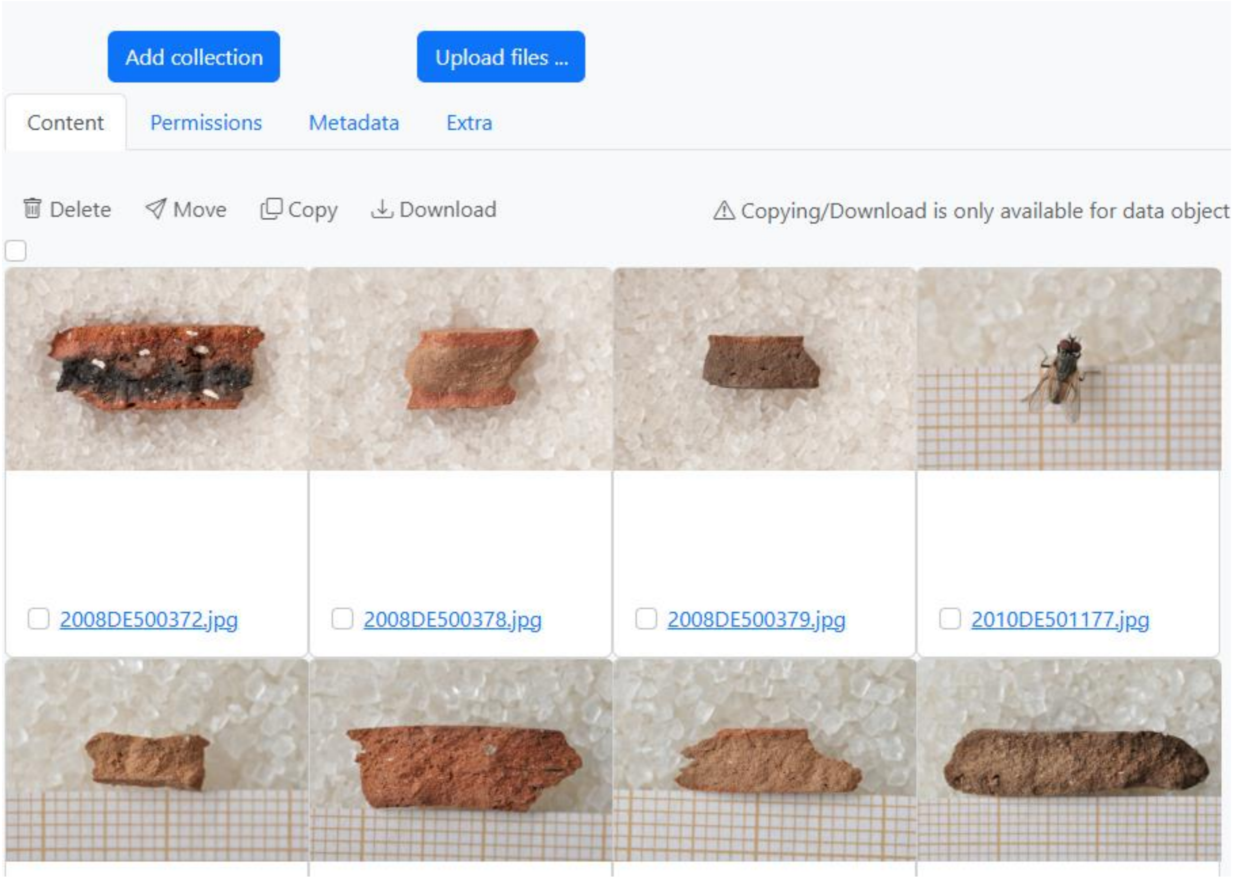
set » home » datateam_set » paul-set

Sun_120254_pss_gpp.jpg

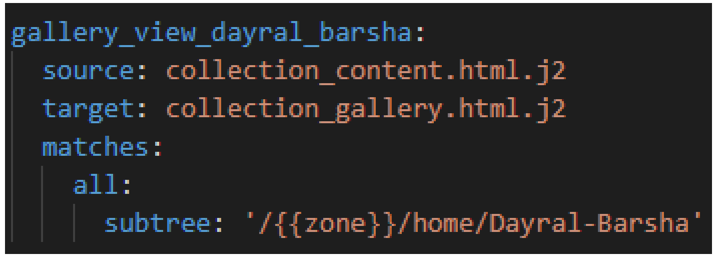
⚠The earliest events recorded for history are from 2024-10-28.

Time	Realm	Action	Client	User	Path
2025-04-08 15:52:43	datateam_set	verify_checksum	python-irodsclient	u0123318	/set/home/datateam_set
2025-03-17 21:32:33	datateam_set	verify_checksum	ManGO_portal	u0123318 (operator)	/set/home/datateam_set

Fine grained template overrides



⚠ Copying/Download is only available for data object



```
gallery_view_dayral_barsha:
  source: collection_content.html.j2
  target: collection_gallery.html.j2
  matches:
    all:
      subtree: '/{{zone}}/home/Dayral-Barsha'
```

Future

ManGO Ingest

- Modularity: use higher level functions in your own ingest scripts/framework
- Stability/robustness improvements
- Egress functions

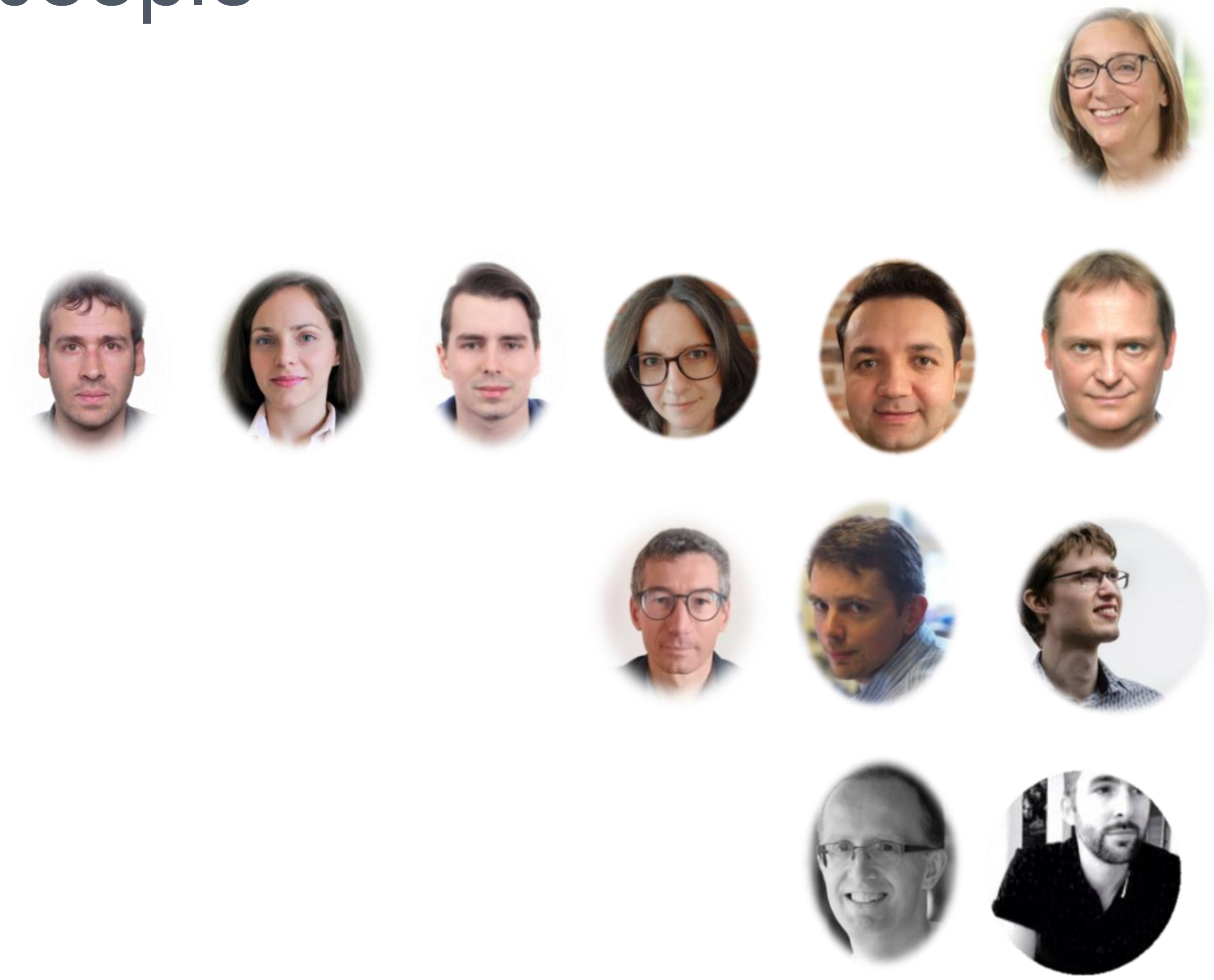
ManGO Flow

- GUI (Flask Blueprint)
  - CRUD on workflow definitions
  - Monitoring
- Many more

ManGO

- Refactoring
- More plugins

The people



Policy

User support

Middleware

Storage

Thank you!

paul.borgermans@kuleuven.be

CODE:
https://github.com/kuleuven