

ManGO Platform Updates

Paul Borgermans
and team FOZ RDM ICTS – KU Leuven



ManGO RDM Platform

 Portal

 Flow

 Lib

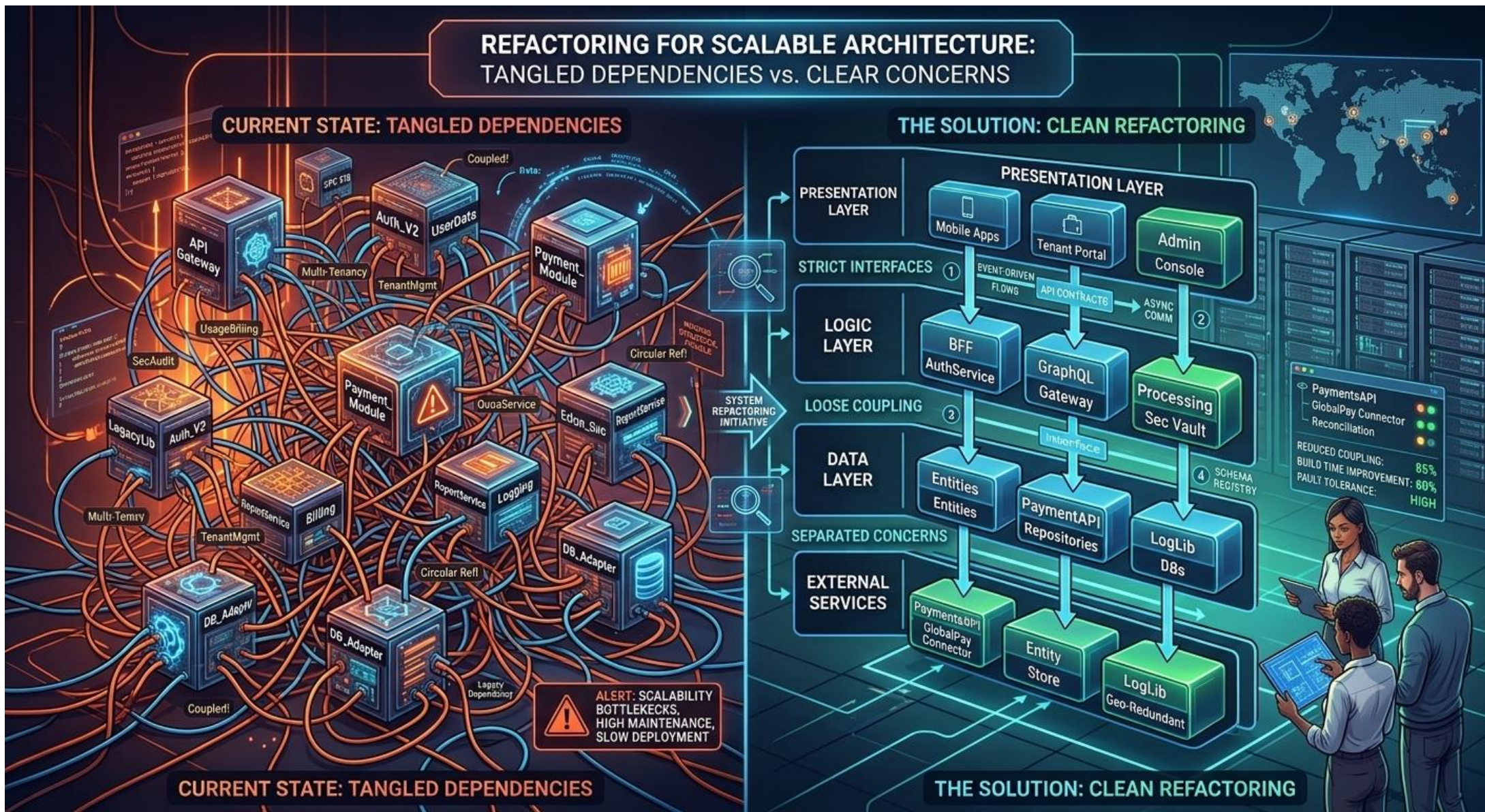


mango_audit

mango_open_search

mango_mdschema

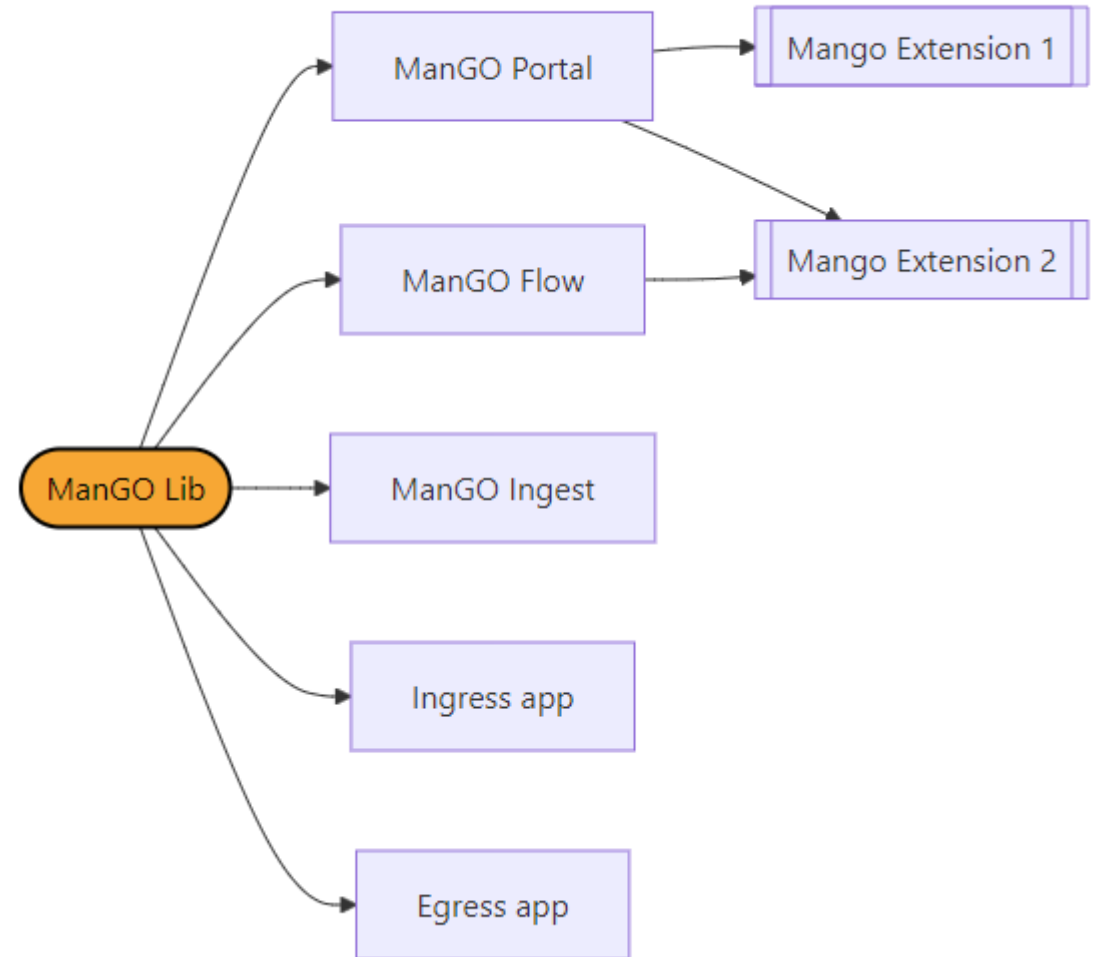
...



Generated by Google Gemini with lousy prompting

Refactoring goals

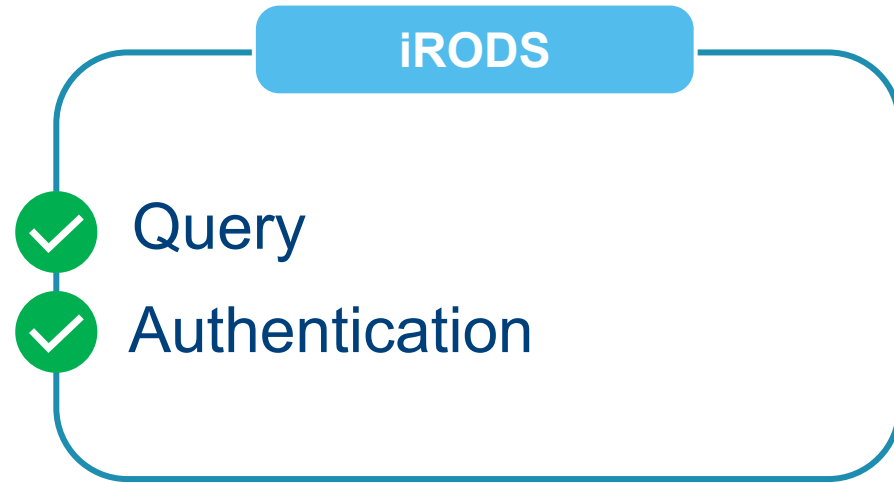
- Proper packaging (**uv**)
- **py.typed**
- Enhancing re-useability
- General <> KULeuven specifics
- Better testing
- Documentation





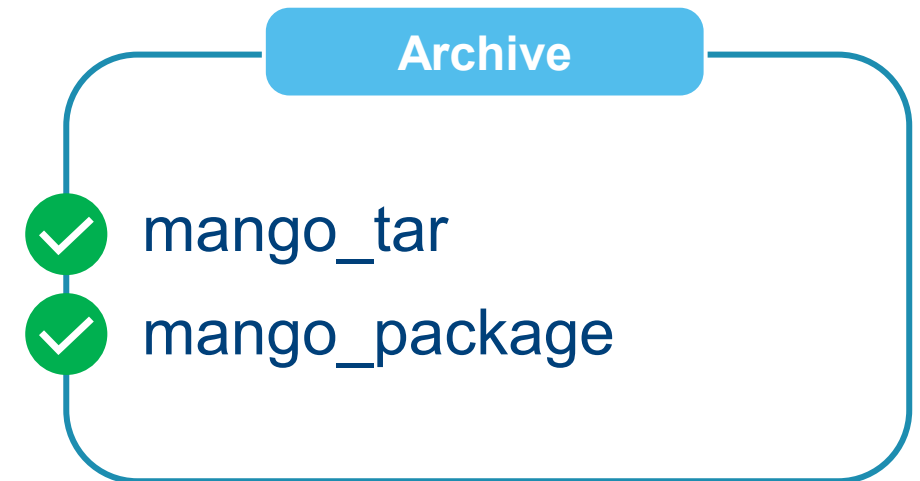
```
mango_lib
├── __init__.py
├── archive
│   ├── __init__.py
│   ├── mango_package.py
│   └── mango_tar.py
├── cli
│   ├── __init__.py
│   └── base.py
├── debug.py
├── irods
│   ├── __init__.py
│   ├── auth.py
│   └── query.py
├── jinja
│   └── __init__.py
├── metadata
│   └── __init__.py
└── patterns.py
```

Modules



```
result = query_module.genquery2_select_dict(  
    session,  
    {"id": int, "name": str.upper},  
    "WHERE DATA_NAME like '%.txt'",  
)
```

```
ZoneType = str  
ClientUserType = str  
ZoneOperatorSessionFunction = Callable[  
    [Optional[ZoneType], Optional[ClientUserType]], iRODSSession  
]
```



Modules

```
@MangoJinjaExtension.template_filter()
def get_one_irods_metadata(
    irods_item: iRODSDataObject | iRODSCollection,
    metadata_name: str,
    default_value=None,
    as_tuple=False,
):
    try:
        avu: iRODSMeta = irods_item.metadata.get_one(metadata_name)
        if as_tuple and avu:
            return (avu.value, avu.units)
        return avu.value if avu else default_value
    except Exception:
        return default_value # iRODSMeta(metadata_name, default_value)
```

Jinja

- ✓ Native Jinja Extension
- ✓ Template filters

Patterns

- ✓ alternate regex
- ✓ Path parameter extraction

Modules

Metadata

- ✓ Extraction from files
- ✓ Optional: Exiftool, CZI, ..

Debug

- ✓ true object size

CLI



```
• > mango --help
Usage: mango [OPTIONS] COMMAND [ARGS]...

ManGO command line interface.

Base CLI for the ManGO ecosystem. Other packages (e.g. mango_portal,
mango_flow) can register their own subcommands via the ``mango.plugins``
entry point group.

Options:
  --version  Show the version and exit.
  --help     Show this message and exit.
```

- ✓ mango
- ✓ extensible

ManGO Portal

Recap



- Multi-tenant/multizone web client to iRODS
- Flask framework
- Currently developed on project aka realm paradigm
 - `/<zone>/home/<project aka realm>`
 - user management via rodsadmin proxy

Changes

- package
 ➡ create your own "instance"
- move out KULeuven specific parts
- Move "hard coded" dependencies to optional plugins
 - metadata extraction (Tika)
 - Previews (Pillow, ...)

Changes (...)

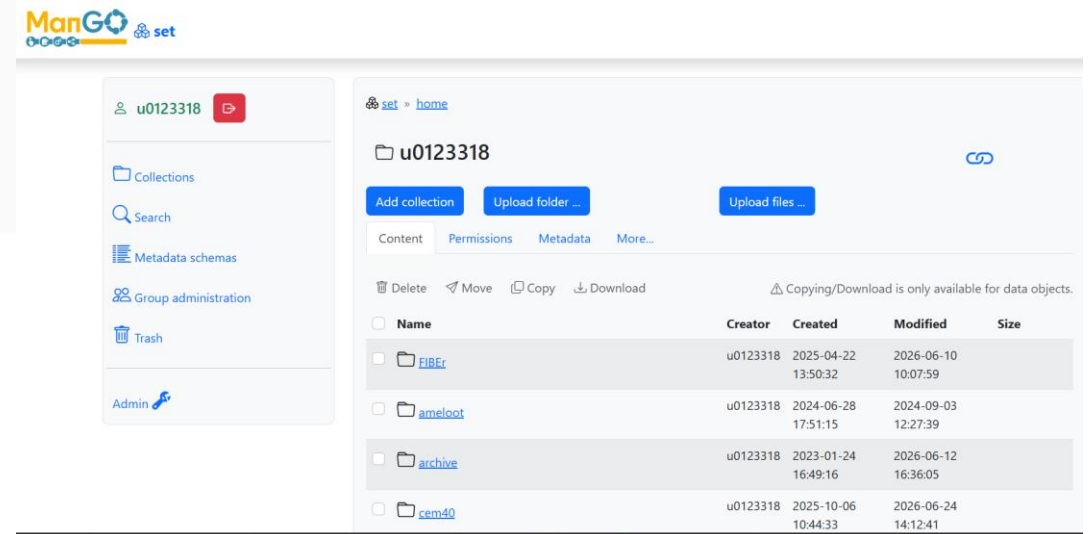
- templates moved to blueprint
 - Application instance have priority (by design, Flask, Jinja2)
- Configuration defaults
 - Overriden by instance defaults

Installation



```
# irods running on localhost 1247
uv init --app my-portal
cd my-portal
uv add mango-portal # or --editable <local>/mango-portal
uv run mango portal init
source .venv/bin/activate
source boot_mango.sh # development
```

Shell





Recap

- data flows and transformations
- not (yet) driven directly by iRODS PEP's
- driven by
 - periodic monitoring
 - direct actions
 - from users
 - outside actors
- Celery (thrusted, mature with active development)

Changes (internal)

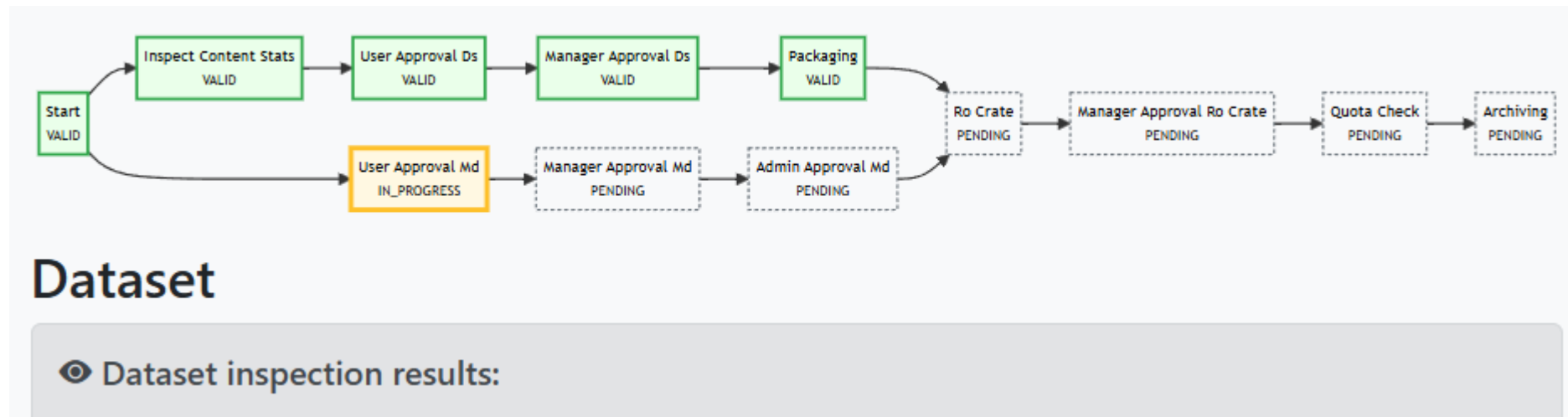
- Internal modules -> ManGO Lib
 - archiving
 - patterns
- tasks and extended canvas paradigms
- beat (scheduler)
 - extended monitoring options

Use case : cold storage

- Data driven
- Mix
 - interactions
 - asynchronous tasks
 - monitoring

ManGO Portal + ManGO Flow

= FriGO



Documentation

- **mkdocs + mkdocstrings**
 - Markdown + Docstrings (google style)
- One documentation repository
- Static site
 - Including powerful search
 - For now hosting on github pages
- Frequent updates

Documentation

 **ManGO (Development) Documentation**



ManGO Lib

- mango_lib.debug
- mango_lib.jinja
- mango_lib.patterns
- Archive
 - mango_lib.archive.mango_pa...
 - mango_lib.archive.mango_tar
- Irods
 - mango_lib.irods.auth
 - mango_lib.irods.query

mod mango_lib.jinja

mango_lib.jinja provides a custom Jinja2 Extension , template filters and utility functions

Usage examples:

General use:

```
from jinja2 import Environment
from mango_lib.jinja import MangoJinjaExtension

env = Environment(extensions=[MangoJinjaExtension])
template = env.from_string("{% 3661.5 | format_delta_time %}")
print(template.render()) # 01:01:01.50
```

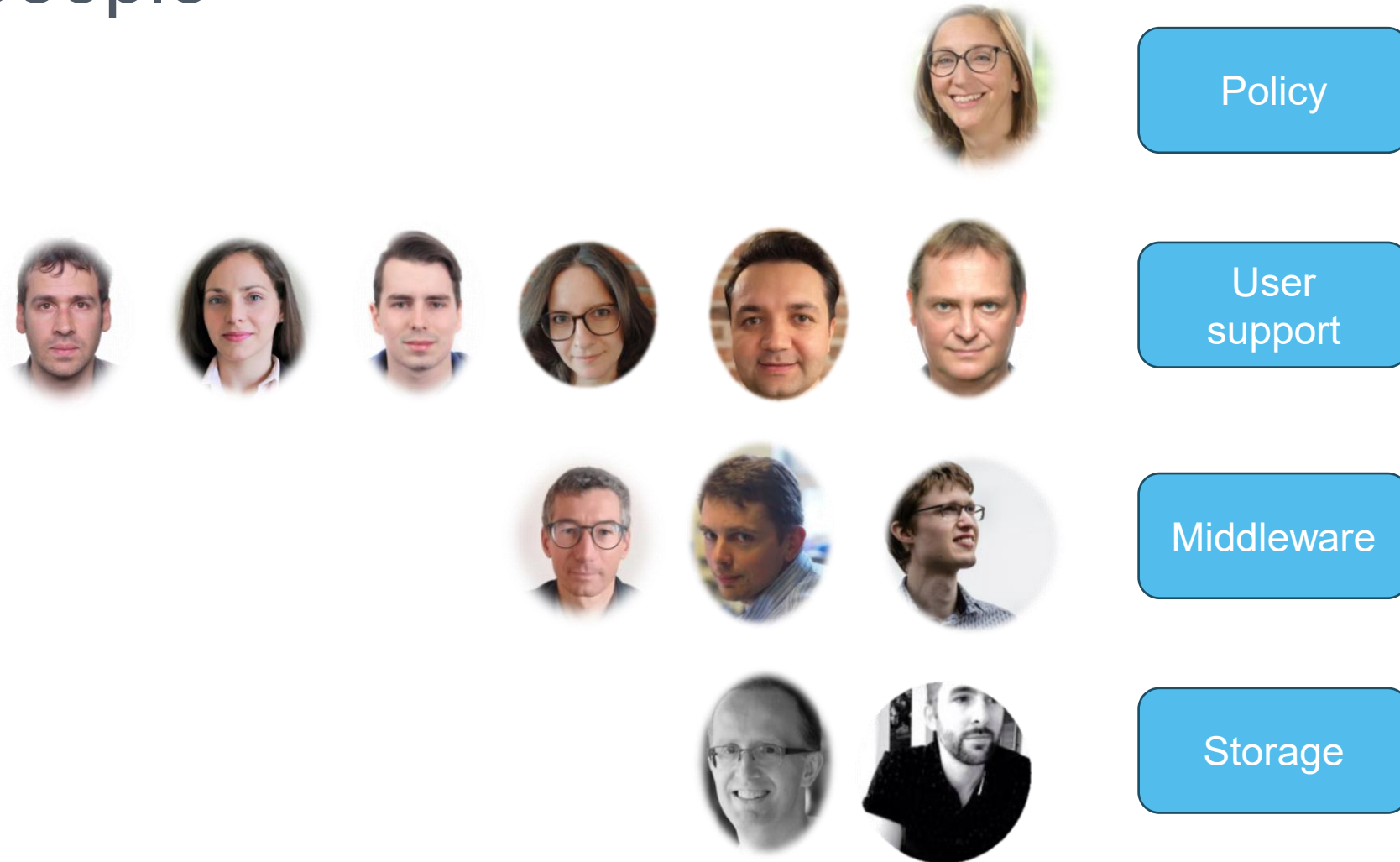
Use in a Flask app like ManGO portal:

```
from flask import Flask
from mango_lib.jinja import MangoJinjaExtension
```

Table of contents

- class** MangoJinjaExtension
 - meth** template_filter
- class** MangoJinjaUtils
 - attr** jinja_env
 - meth** apply_jinja_template
- func** b64encode
- func** bleach_clean
- func** convert_timestamp_date.
- func** format_datetime
- func** format_delta_time
- func** format_timestamp
- func** get_irods_metadata
- func** get_one_irods_metadata
- func** irods_to_sha256_checks.
- func** iter_intersection

The people





Thank you!

paul.borgermans@kuleuven.be

<https://github.com/kuleuven/RDM>