



**MORPHEMIC**

# MORPHEMIC Crawler

Maria Di Girolamo – Engineering Ingegneria Informatica S.p.A.

H-Cloud (26/10/2021)

Virtual Meeting

## **Introduction: MORPHEMIC & the Web Crawler**

- Polymorphic Adaptation
- Application Profiling

## **The Crawler: a detailed description**

- Design
- Functionalities
- Next steps

# Outline

## Introduction: MORPHEMIC & the Web Crawler

- Polymorphic Adaptation
- Application Profiling

## The Crawler: a detailed description

- Design
- Functionalities
- Next steps

# The MORPHEMIC's Crawler

- Crawls the web looking for open source projects
- Configurable open source code repositories
- Gets and stores metadata and references to the source code

## Polymorphic Adaptation

Dynamic reconfiguration of polymorphic applications' deployment and form according to the context:

- apply alternative deployment models
- adopt different forms (e.g. function or micro-service) or configuration classes (container, VM, serverless).

# Polymorphic Adaptation Application Profiling

## Polymorphic Adaptation

Dynamic reconfiguration of polymorphic applications' deployment and form according to the context:

- apply alternative deployment models
- adopt different forms (e.g. function or micro-service) or configuration classes (container, VM, serverless).

### Application Profiling

- define an "application profile" based on application features
- potential deployment models retrieved from the application profile

# Polymorphic Adaptation Application Profiling Web Crawler

## Polymorphic Adaptation

Dynamic reconfiguration of polymorphic applications' deployment and form according to the context:

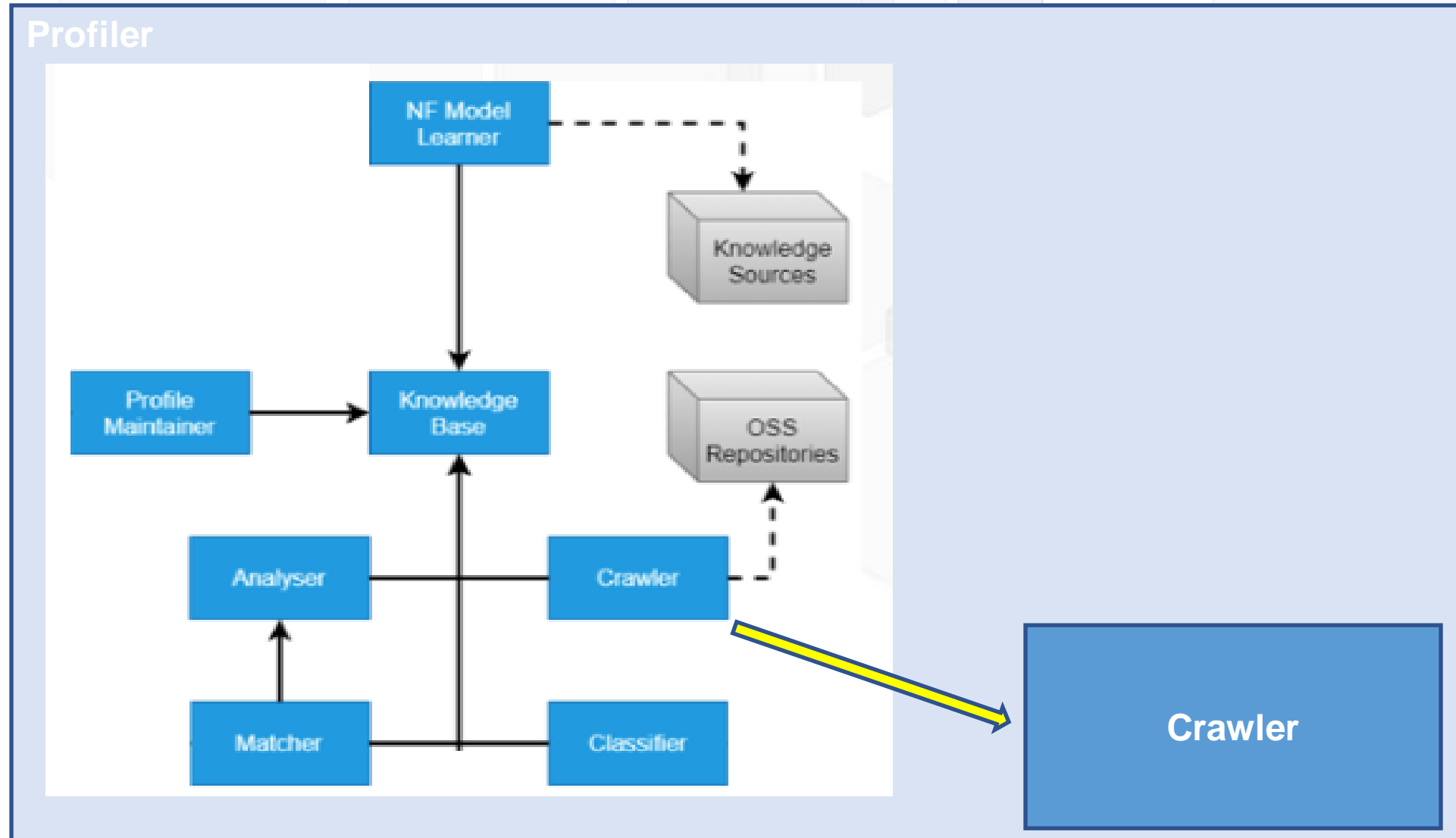
- apply alternative deployment models
- adopt different forms (e.g. function or micro-service) or configuration classes (container, VM, serveless).

### Application Profiling

- define an "application profile" based on application features
- potential deployment models retrieved from the application profile

**Crawler**

# Web Crawler: part of the Application Profiler Architecture





# Outline

## Introduction: MORPHEMIC & the Web Crawler

- Polymorphic Adaptation
- Application Profiling

## The Crawler: a detailed description

- Design
- Functionalities
- Next steps

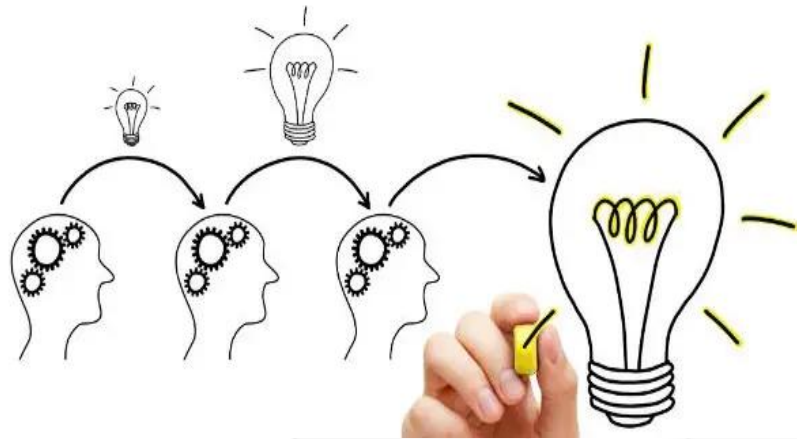
# MORPHEMIC' Crawler



MORPHEMIC

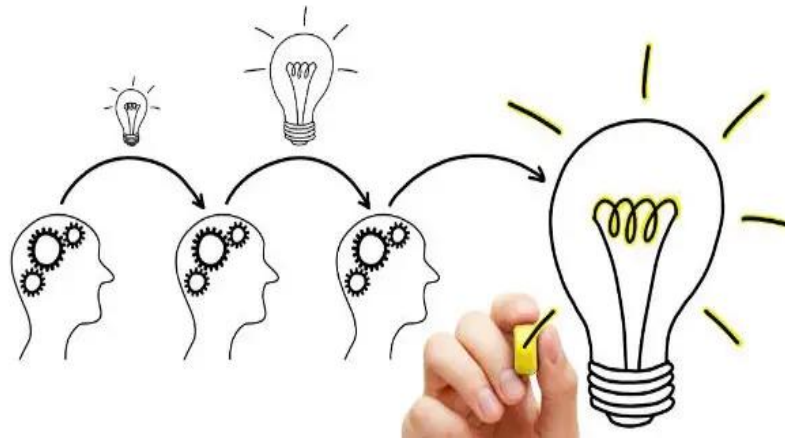
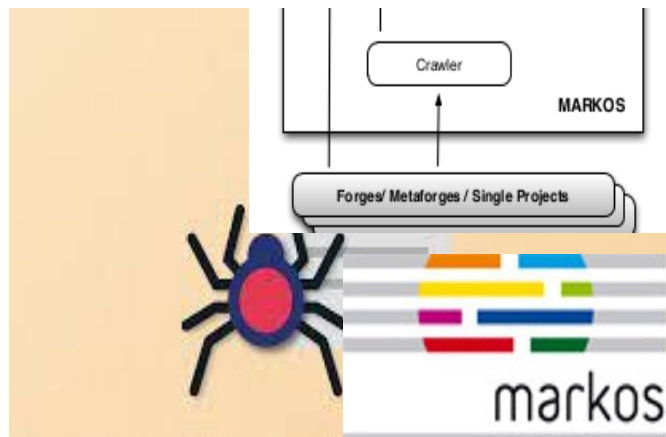
Crawler

# MORPHEMIC' Crawler



Crawler

# MORPHEMIC' Crawler



# Functionalities

- Download the source code and the metadata information of the projects using the “open source projects information”

*Open Source projects information: set of forge, metaforge, list of directory from which project metadata are downloaded. Specifically they are project-name, url, version, release date, creation date, reference link of the source code.*

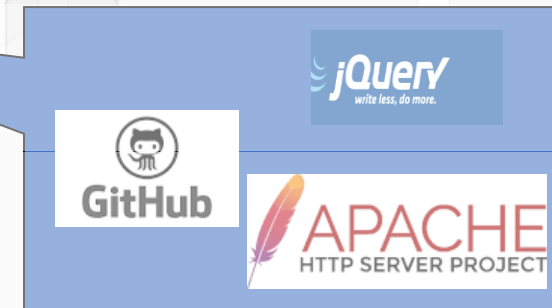
- Download the source code and the metadata information of the projects using the “open source projects information”
- Maintain a list of the configured open source projects repositories

*Open Source projects information: set of forge, metaforge, list of directory from which project metadata are downloaded. Specifically they are project-name, url, version, release date, creation date, reference link of the source code.*



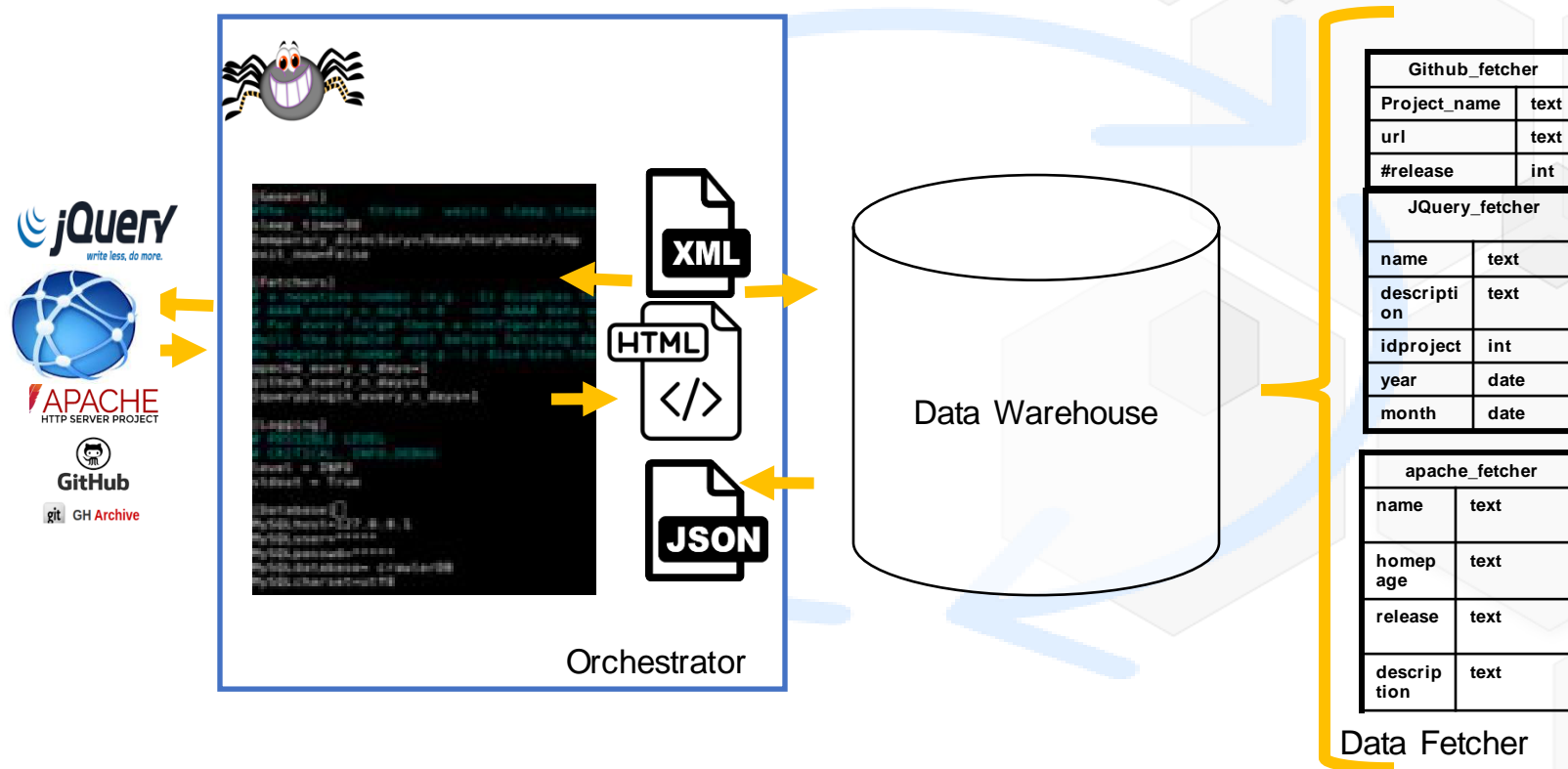
- Download the source code and the metadata information of the projects using the “open source projects information”
- Maintain a list of the configured open source projects repositories
- Store extracted data in a DataWarehouse
- Integrate and/or update the project metadata if needed(e.g. if a new release happens or change).
- Make the information on the projects available via REST API in JSON format for other MORPHEMIC components (as KnowledgeBase).

*Open Source projects information: set of forge, metaforge, list of directory from which project metadata are downloaded. Specifically they are project-name, url, version, release date, creation date, reference link of the source code.*



*For each open source project repository a dedicated table (data fetcher) is provided*

# Details of the Crawling process: Storing





# Details of the Crawling process: Reading



Orchestrator

```
#The main thread waits sleep_timesteps after each loop.
sleep_time=30
temporary_directory=/home/morphemic/tmp
exit_now=False

[Fetchers]
# a negative number (e.g. -1) disables the source
# AAAA_every_n_days < 0 ==> AAAA data won't be crawled
# For every forge there a configuration line which tells how many days
# will the crawler wait before fetching data again from that forge;
# a negative number (e.g. -1) disables the source.
apache_every_n_days=1
github_every_n_days=1
jqueryplugin_every_n_days=1

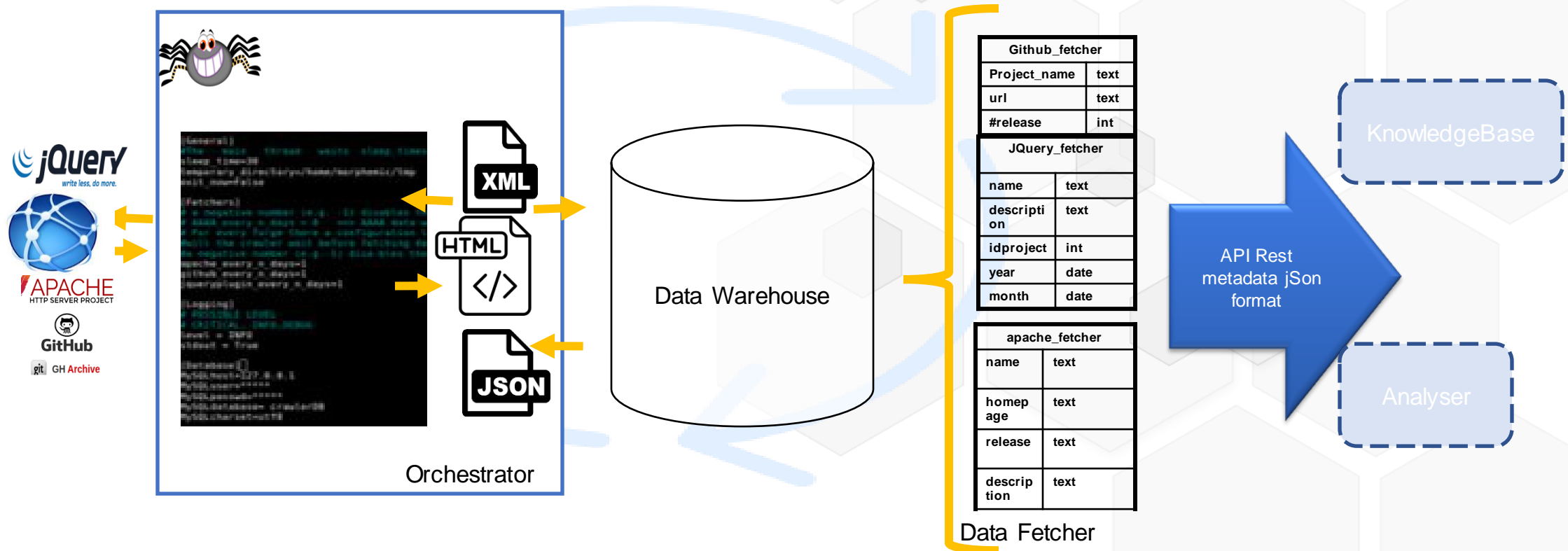
[Logging]
# POSSIBLE LEVEL
# CRITICAL, INFO, DEBUG
level = INFO
stdout = True

[Database]
MySQLhost=127.0.0.1
MySQLuser=*****
MySQLpasswd=*****
MySQLdatabase= crawlerDB
MySQLcharset=utf8
```

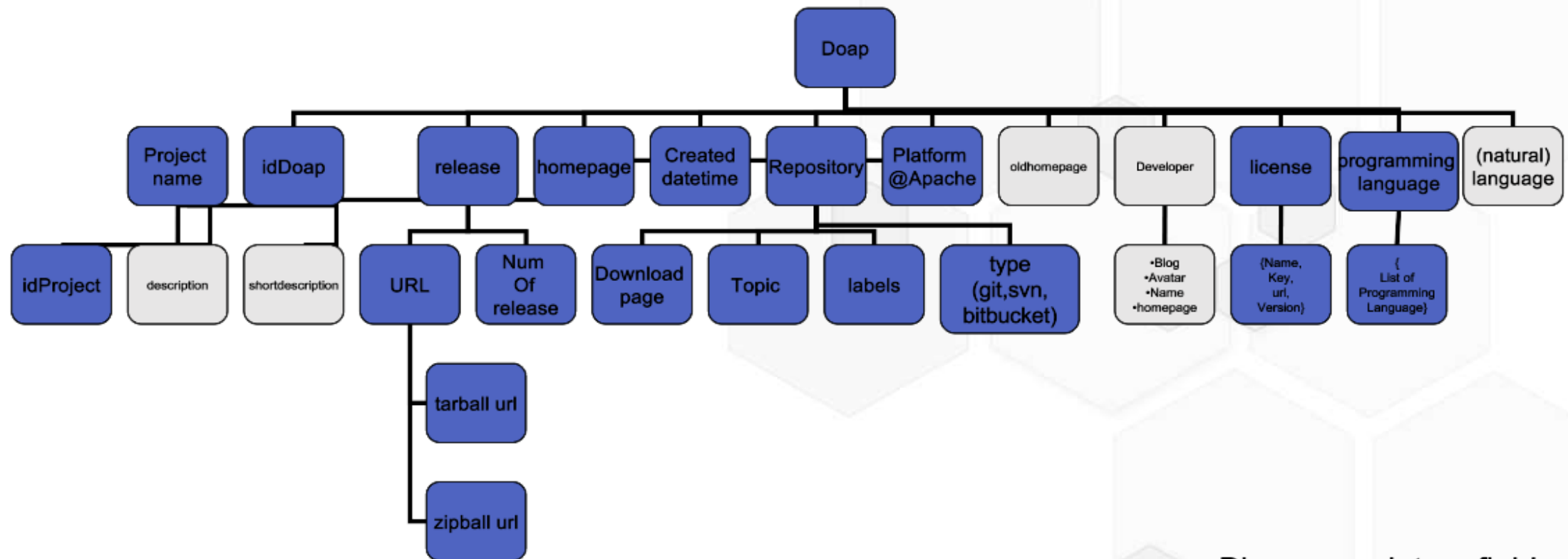
# Details of the Crawling process: Downloading



# Details of the Crawling process: Sending



# Details of the Crawling process: DOAP Model



Doap Model Schema

- Blue - mandatory field
- Light Grey - optional field

# Next steps

- Include more open source software repositories
  - e.g. *R-Forge*, *C-Tan*
  - Main criteria: metadata suitable for Polymorphic Adaptation
- Verify the compliance with Ubuntu 20.04 LTS
  - Currently the compliance with Ubuntu 18.01 is verified



**MORPHEMIC**

# Thank You



[www.morpheMIC.cloud](http://www.morpheMIC.cloud)



[facebook.com/MorpheMICProject](https://facebook.com/MorpheMICProject)



[www.linkedin.com/showcase/morpheMIC-project](https://www.linkedin.com/showcase/morpheMIC-project)



[twitter.com/\\_morpheMIC\\_](https://twitter.com/_morpheMIC_)