



# the Digital Object Gateway

Dieter Van Uytvanck & Michał Gawor  
CLARIN ERIC

Centre Meeting 2020-06-10

*on the internet, nobody knows you're a dog*

[https://commons.wikimedia.org/wiki/File:Nobody\\_knows\\_you\\_are\\_a\\_dog.jpg](https://commons.wikimedia.org/wiki/File:Nobody_knows_you_are_a_dog.jpg)

Creative Commons Attribution-Share Alike 4.0 International

# Origin (1)

- **CLARIN provides easy access to language resources.**
- One specific subgoal is to enable machine-processability of language data:
  - command-line/programmatic processing (for courses, etc)
  - services like the Switchboard (for demonstration/exploration)
- Wouldn't it be nice if you could ...
  - Iterate over the files in a virtual collection and process them with any python library?
  - Just put "any" URL or PID in the switchboard, even if there only is a landing page?

## Origin (2)

- Observations:
  - (neutral) Lots of relevant resources are available via general purpose repositories such as Zenodo or portals such as Europeana.
    - CLARIN tools should also be able to process these resources!
  - (negative) Lots of data (files) are only indirectly accessible via
    - a landing page (e.g. Zenodo) > not processable
    - a specific API (e.g. Europeana) > processable, but requires specific know-how
  - (positive) We have standardized metadata descriptions for all CLARIN repositories, including:
    - A persistent identifier (that points to a CMDI file)
    - A CMDI file that points in a standard and typed way to files

# Idea

- Let's make a **lightweight** solution that:
  - Provides access to general-purpose repositories
  - Solves the landing page / API problem
  - Leverages the standardized solution for CLARIN repositories
- Enter the Digital Object Gateway ... an abstraction layer for accessing Digital Objects that are referred to with a Persistent Identifier (PID) or URL.

# Philosophy

- acknowledging that
  - the PID and repository landscape is pretty **chaotic**
  - we need some working solutions right **now**
- not trying to impose a single technical framework to address this chaos, but rather to cross-connect and **leverage** the already **existing** (and **future**) access **methods**
- **not demanding repositories to make any changes**

# Implementation

- The DOG will be implemented on 3 levels:
  - As a REST (micro)service accessible over HTTP(S) [demo Michał]
  - As a Python library [demo Michał]
  - As a simple web application [wireframe demo]
    - based on the REST service or the Python library

# Digital Object Gateway

Input URL/PID:

<http://doi.org/10.23728/b2share.d64361c0a6384760a8a8f32e0dc4a481>

Analyze

## Analysis Result

Input type: **Digital Object Identifier (DOI)**  
Status: **Recognized**  
Repository name: **B2SHARE**  
Repository type: **Invenio**

## Deep Scan Result

### Metadata

Title: **Comparison of the usage of nouns by female and male members of the Polish parliament**  
License: **Creative Commons Attribution (CC-BY)**

### Files



[male\\_female\\_speeches.zip \(5.43MB\)](#)

Send to Switchboard



[output-descriptive\\_features.csv \(15KB\)](#)

Send to Switchboard



[output-descriptive\\_features.xlsx \(28KB\)](#)

Send to Switchboard

Go to Landing Page

View repository API output

# Future & Uptake

- The DOG is has a wider use than the CLARIN community
- Plans
  - First have a prototype/beta version out
  - Then promote it in wider circles
    - EOSC
    - PID fora
    - Europeana



## Recap: why is the DOG important?

- It will leverage the many repository-specific APIs.
- It increases the involvement and visibility of individual repositories, with no cost on their side.
- It will make the life of programming researchers easier.
- It will enhance reproducibility.
- It will showcase the practical benefits of Persistent Identifiers.
  - making them more actionable
- It allows for easy extension, to support [future technologies](#).
  - Once a plugin is created on the side of DOG, applications that use DOG as an abstraction layer to access digital objects can use it without the need to change code on their end.

# Thank you for your attention!

More information:

- the [Digital Object Gateway draft spec](#)
  - Feedback welcome!
- <https://github.com/clarin-eric/DOGLib>
- <https://fairdo.org/>