Implementation of the COAR Notify protocol in DSpace 8

- **Background**
- **High level proposal summary**
- **Questions to be addressed early in the development**
  - The feature will be turnable on/off, should we put the code in a separate maven module or in the main dspace-server-webapp maven module?
    - 1) Main dspace-server-webapp
    - 2) Separate maven module
  - Where the full LDN json messages should be stored? we want to keep also a copy of the original JSON message.
    - 1) in a new table with the full content as a detached bitstream (like the processes in/out files)
    - 2) in the database, the json file as a text column (the same used for the text_value in the metadatavalue)
    - 3) in a new dedicated SOLR core
  - How to deal with LDN Messages indexing?
    - 1) in discovery as a new IndexableObject
    - 2) with a dedicated API
  - How to name the endpoints specific of the COAR Notify feature?
    - Current proposals (4Science)

**Background**

The Arcadia-funded COAR Notify Project is developing and accelerating community adoption of a standard, interoperable, and decentralised approach to linking research outputs hosted in the distributed network of repositories with resources from external services such as overlay-journals and open peer review services, using linked data notifications. As part of this project, COAR is funding the development of platforms and systems to support the exchange of linked data notifications across partner organisations and the workflows to manage notifications in those platforms and systems. As the largest adopted repository platform in the World one of the first platforms to be addressed is DSpace the implementation of which has been entrusted to 4Science in continuation with a previous proof-of-concept project [2]. The implementation plan has been presented at OR2023 [1] and the discussed on the 13th July 2023 at the DSpace Developer meeting.

**High level proposal summary**

We will introduce support for Linked Data Notification in DSpace providing

- an embedded inbox implementation (i.e. the ability for DSpace to receive notification) with support for automatic discovery
- a service able to generate LDN message about DSpace items to make announcement, request services, acknowledge requests

Incoming and outgoing messages will be placed in a queue to allow asynchronous processing and future replacement of the InBox / LDN Sender by emerging open source implementation.

![Figure 1. Technical Architecture - Production ready but still "out of the box". Based on https://github.com/antleaf/notify-implementation/blob/main/handbook/content/architecture/figure_2.png](https://github.com/antleaf/notify-implementation/blob/main/handbook/content/architecture/figure_2.png)

External COAR Notify enabled services must be registered (TRUSTED) in advance in a local managed registry. A global COAR Notify Registry, currently under development in the context of the COAR Notify project, could be connected in future.
The whole feature can be turned off, by default we propose to have it enabled.

Processing of messages is configurable via a pipeline of reusable components, to easily add support to future new COAR Notify patterns and provide flexibility to meet different institution's needs and integration scenarios. Each component will take a specific actions such as send an email notification, suggest to change item's metadata, suggest to create new items, new versions, attach files, etc. A common interface will be defined so that the individual component will be registered and configured via Spring beans.

- the suggestions about performing changes to existing item will be based on the "Correction Service" proposed here (and in the other related PRs) https://github.com/DSpace/DSpace/pull/8184. Each COAR Notify external service will be treated as an individual provider of the correction service;

- the suggestions about creation of new content will be based on the "Publication Claim" service proposed here (and in the other related PRs) https: //github.com/DSpace/DSpace/pull/8280. Each COAR Notify external service will be treated as an individual provider of the correction service.
A new submission panel will be provided so that during the submission and / or the workflow it will be possible to select which COAR notify patterns LDN should be sent to selected external services. Moreover, it would be possible to configure some external services to be automatically notified about all the items matching specific criteria (using the Item Logical Filter system already in use by the DOI minting process).

![Submission panel to select COAR Notify patterns / services to interact with](image1)

**Figure 4.** Submission panel to select COAR Notify patterns / services to interact with

![Validation of selected services according to the item information](image2)

**Figure 5.** Validation of selected services according to the item information

On the administer item page it will be possible to check all the LDN related activities, moreover information box will provide status update about the COAR notify events related to the item to authorized users (Submitter, Administrators)
Figure 6. Administer item page, COAR Notify tab showing LDN messages related to the item

Figure 7. An item where a request for review / endorsement is still pending (not acknowledged nor fulfilled by the external service)

Figure 8. An item where a review / endorsement has been received and require manual approval
The result of a COAR Notify scenario is usually an enrichment of the item metadata such as

- related review
- overlay journals that have endorsed the item
- other resources that have been announced as related (mainly supplemented by or supplement of) to the item

As an end user the impact of the COAR Notify protocol in DSpace would be limited to the visualization of additional metadata that could have been filled manually or, indeed, automatically via the COAR Notify protocol.
The exact metadata that will be used to track these additional information will be configurable and the default configuration will suggest usage of appropriate metadata in existing standards. We are currently in touch with the Datacite metadata experts to get suggestions.

A global administrative dashboard will be also available to monitor the general usage of the COAR Notify protocol across the repository.

**Figure 11a.** Top of the Administrative dashboard, Statistics about usage of the COAR Notify protocol: focus on LDN message

**Figure 11b.** Bottom of the Administrative dashboard, Statistics about usage of the COAR Notify protocol: focus on COAR Notify patterns on the left the "result of pattern (received or generated)" on the right the status of request and their acknowledgement
Figure 12. Administrative dashboard LDN logs, possibility to search and check the individual status of each LDN message

Figure 13. Additional entries in the administrative menu

Questions to be addressed early in the development

We would like to get early feedback on the following aspects as changes on these could require significative work later and make harder to merge the contribution in the official DSpace code base. Ideally we would like to reach a consensus by the end of July 2023 at latest.

It stays confirmed that at any time of the development or during the code review if major fault or deficit would be noted due such early decision can be revisited

The feature will be turnable on/off, should we put the code in a separate maven module or in the main dspace-server-webapp maven module?
Options:

1) Main dspace-server-webapp

Pro: easier approach, lightweight, IDE friendly. Aligned with the other recent development of lightweight protocols such as OpenSearch and Signposting

Contro: Nothing identified yet

Vote: 4Science

2) Separate maven module

Pro: it would keep the library dependencies more isolated. It allows institution to potentially exclude the module from the build

Contro: the build process of DSpace is already more complicated than the average of opensource web project. At the end we build just a single web app forcing maven to make extra work to manage all our dependencies with the risk to introduce bug relative to transactional dependencies. This consider is not really specific of this project and would apply also for other dspace maven modules but in such case the benefit of eventually exclude a such lightweight implementation that don't rely on off stream libraries seems to be minimal

Vote:

- In meeting on July 13, 2023 several attendees (namely Tim Donohue & Art Lowell) favored this "separate maven module" approach as it aligns better with OAI-PMH and SWORD. The primary concern is around whether we should have generic LDN Java implementation code embedded in our "server webapp".
  - If this LDN implementation code is larger in scale (like the generic Java implementations of XOAI and SWORD protocol), then it'd make more sense to have a new "dspace-lrn" module (or similar) which contains that generic Java code separate from DSpace-specific implementation code. However, it's likely that this approach would still require some DSpace-specific implementation code to be in the "server webapp".
  - That said, if we find that the LDN implementation code is small in size, it would be reasonable to have it contained directly in the "server webapp" (as in option 1). This would be similar to Signposting (in org.dspace.app.rest.signposting), which has a much smaller implementation and most of the code is DSpace-specific.

Where the full LDN json messages should be stored? we want to keep also a copy of the original JSON message.

Please note that to manage the queue of messages to process we expect to use a separate table in the database because we need to guarantee atomic transaction and manage lock to avoid double processing.

Options

1) in a new table with the full content as a detached bitstream (like the processes in/out files)

Pro: It seems to be cleaner and future proof approach as the JSON file is currently limited in size but it could potentially become a bit larger in future especially for usage of LDN outside the Notify protocols (that by design promises to keep payload limited). It is similar to how we already deal with processes

Contro: it would be slower and slightly more costly than other approach

Vote: 4Science

- In the meeting on July 13, 2023, several attendees noted that either of the first two approaches seem reasonable. It'd be fine to store these messages in files... or, since they are small in size, they could be stored in the database (option 2). However, we recommend AGAINST storing in a dedicated Solr core (option 3).

2) in the database, the json file as a text column (the same used for the text_value in the metadatavalue)

Pro: Easier to implement, best performance

Contro: It couple a bit more the inbox with dspace as the LDN message is completely stored with the entity. It could be possible in future that advanced LDN InBox are developed out-side DSpace and it would be nice to support integration instead than maintain our limited implementation

Vote:
3) In a new dedicated SOLR core

**Pro:** As much as possible "External" to DSpace

**Contro:** Harder to implement. Harder to enable and maintain (another thing to backup) for institution during upgrade

Vote:

How to deal with LDN Messages indexing?

**Options**

1) In discovery as a new IndexableObject

**Pro:** Uniformity with the other search feature in DSpace. Availability of advanced features out-of-box (configuration of facets, sort options, etc.)

**Contro:** Nothing, as the discovery.xml already include default filter for all the configuration to specify which resources should be included in the result

**Vote:** 4Science

- *In the meeting on July 13, 2023, there were no objections to making this an IndexableObject.*

2) With a dedicated API

**Pro:** Nothing identified yet

**Contro:** Lot of duplication in the code base. Different APIs for similar purposes for developers that want to interact with DSpace

Vote:

How to name the endpoints specific of the COAR Notify feature?

No alternative proposals yet.

- *In the meeting on July 13, 2023, there were no objections to the below concept at a higher level.*

Current proposals (4Science)

LDN InBox outside of our API as it follow the LDN W3C specification not our API architecture/best practices

```
/server/ldn/inbox
```

LDN related resources under our API as they will be implemented according to our best practices

```
/server/api/ldn/services
```

will provide details about the external COAR Notify compliant services registered in the repository to interact with

```
/server/api/ldn/messages
```

will provide access to the all the received and sent/to be sent LDN messages

```
/server/api/ldn/qmessages
```

will provide access to the queue messages

```
/server/api/ldn/statistics
```

will provide access to different reports to monitor the COAR Notify protocol usage in the repository

An additional utility endpoint to retrieve all the configured option to filter items

```
/server/api/config/logicitemfilters
```
References
