

EUROPEAN LANGUAGE GRID

META-FORUM 2021, 15/16/17-11-2021

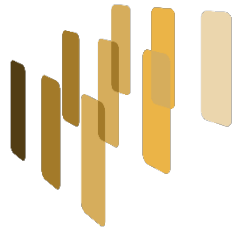
Integrating services into ELG

D. Galanis

<http://www.european-language-grid.eu>

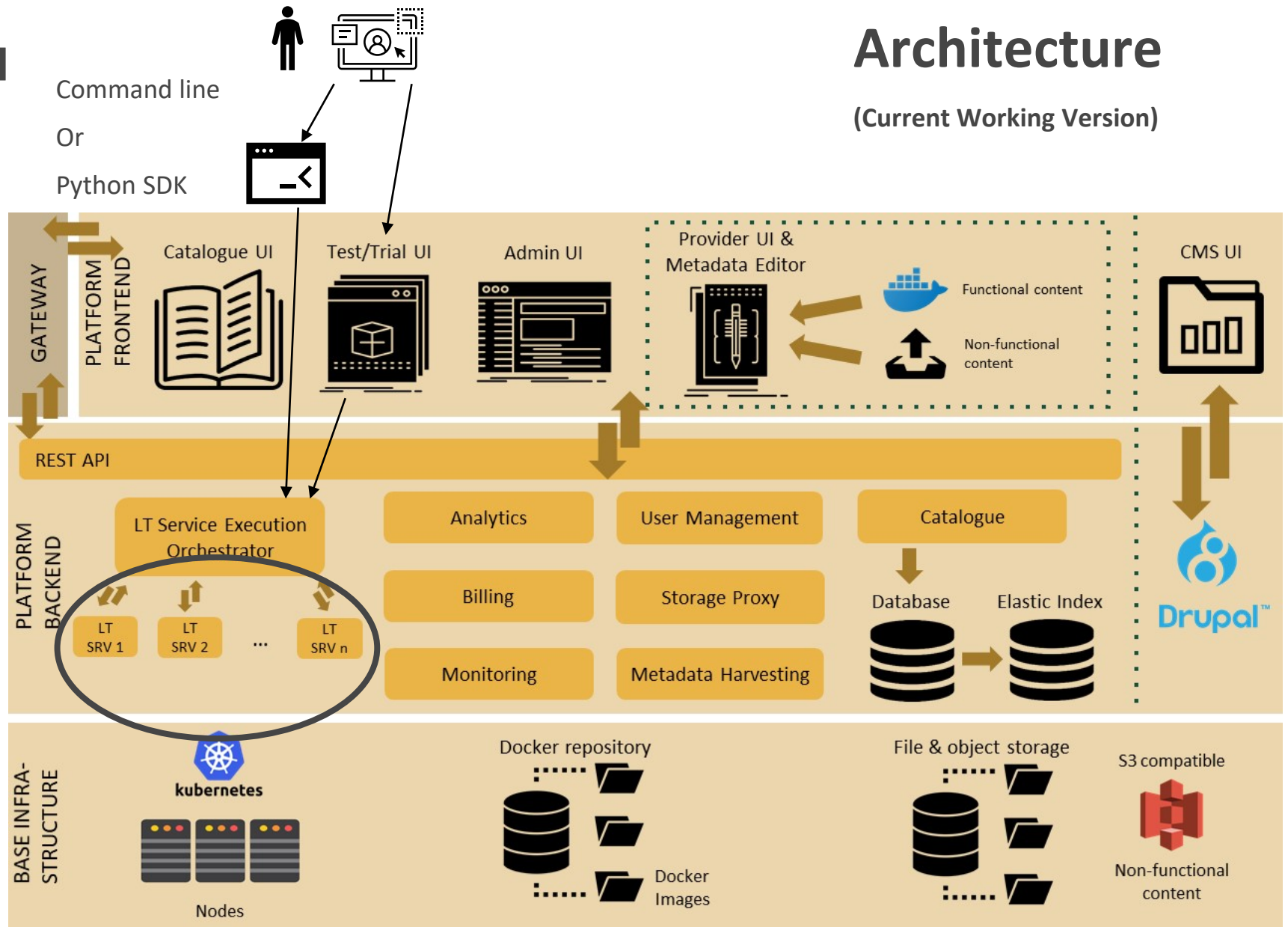
ELG platform

- It is web-based: <https://live.european-language-grid.eu>
- It is a catalogue for resources:
 - Each resource is documented with the required metadata
 - Datasets, **integrated LT services**, not integrated LT services
 - Machine learning models, lexica
 - “Published” resources are visible by anyone; i.e. log-in is not required
 - Log-in is required for using **integrated services**
 - Anyone can become a **provider** and contribute a resource
 - A validation process is required for publication

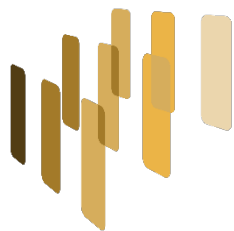


EUROPEAN LANGUAGE GRID

- All integrated LT services run as Docker containers in the Kubernetes cluster of ELG
- All integrated LT services can be called by a user via a common public API offered by the LT Service Execution server.
- Public API documentation: https://european-language-grid.readthedocs.io/en/stable/all/A3_API/LTPublicAPI.html



Architecture (Current Working Version)

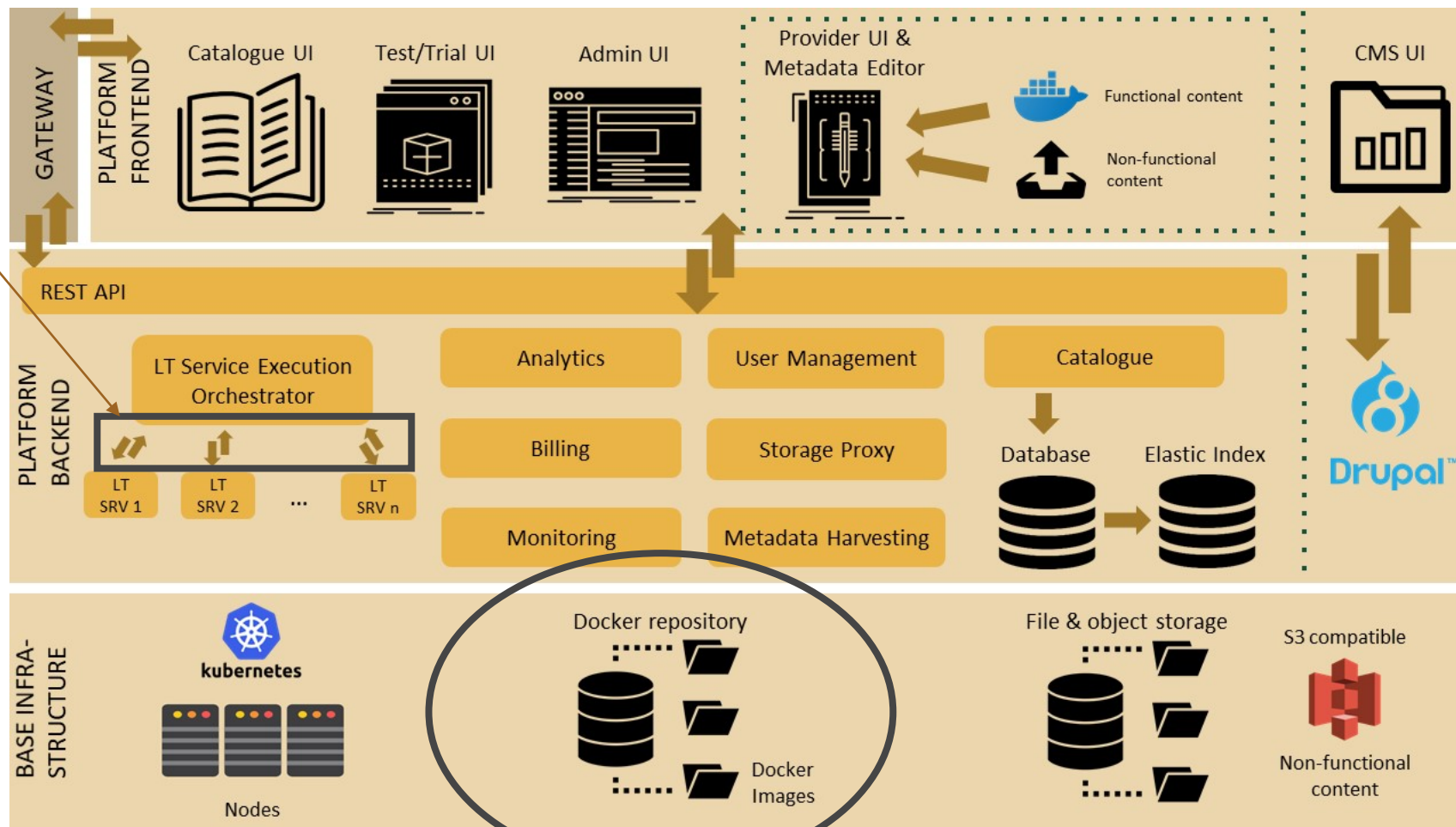


EUROPEAN LANGUAGE GRID

Architecture

(Current Working Version)

- An integrated LT service has to implement the internal LT Service API
- https://european-language-grid.readthedocs.io/en/stable/all/A3_API/LTInternalAPI.html
- Docker images are stored in Docker registries; DockerHub, Gitlab etc.



Provide an ELG-compatible LT service

- Currently, an LT service must fall into one of the following broad categories:
 - Information Extraction (IE)
 - Text Classification (TC)
 - Machine Translation (MT)
 - Automatic Speech Recognition (ASR)
 - Text to Speech Generation (TTS)
- Requirements:
 1. Implement the Internal LT API -> Expose an ELG compatible endpoint
 2. Dockerize

Otherwise it is a special case...and you should contact with the ELG tech. team.

(e.g. COREON SPARQL endpoint <https://live.european-language-grid.eu/catalogue/lcr/8099>)

Provide an ELG-compatible LT service: Technical Requirements

- (1) Expose an ELG compatible endpoint:
 - You MUST create an application that exposes an HTTP endpoint for the LT tool(s).
 - The application MUST consume requests that follow the ELG JSON format, call the underlying LT tool and produce responses again in the ELG JSON format (ELG LT Internal API)
 - E.g. for services that take as input plain text and return annotations (IE).

```
{
  "type": "text",
  "params": {...}, /* optional */
  "content": "The text of the request",
  // mimeType optional - this is the default if omitted
  "mimeType": "text/plain",
  "features": { /* arbitrary JSON metadata about this content, optional */ },
  "annotations": { /* optional */
    "<annotation type>": [
      {
        "start": number,
        "end": number,
        "features": { /* arbitrary JSON */ }
      }
    ]
  }
}
```



```
{
  "response": {
    "type": "annotations",
    "warnings": [...], /* optional */
    "features": {...}, /* optional */
    "annotations": {
      "<annotation type>": [
        {
          "start": number,
          "end": number,
          "features": { /* arbitrary JSON */ }
        }
      ]
    }
  }
}
```

Provide an ELG-compatible LT service: Technical Requirements

- **(2) Dockerization:** You MUST Dockerize the application and upload the respective image(s) in a Docker Registry, e.g. GitLab.

```
Dockerfile 885 Bytes
Edit Web IDE Replace Delete

# Base image.
FROM openjdk:8-jdk-alpine
#VOLUME /tmp

5 # SET TARGET DIR
6 ENV TARGETDIR /elg/
7
8 # Install tini and create unprivileged user
9 RUN apk add --no-cache tini && \
10     addgroup --gid 1001 "elg" && \
11     adduser --disabled-password --gecos "ELG User,,," \
12     --home /elg --ingroup elg --no-create-home --uid 1001 elg
13
14 # Create target dir.
15 RUN install -d -o elg -g elg $TARGETDIR
16 # Copy everything to target dir.
17 COPY --chown=elg:elg dockerCmd ${TARGETDIR}dockerCmd
18 # Copy/Rename server app jar.
19 #ADD --chown=elg:elg /elg-ilsp-lt-services-rest/target/elg-ilsp-services-rest-0.0.1-SNAPSHOT-exec.jar ${TARGETDIR}dockerCmd/app.jar
20 ADD --chown=elg:elg /elg-ilsp-services-rest-0.0.1-SNAPSHOT-exec.jar ${TARGETDIR}dockerCmd/app.jar
21
22 USER elg:elg
23
24 # Set working dir.
25 WORKDIR ${TARGETDIR}dockerCmd
26 #/elg/
27
28 # Start
29 ENTRYPOINT ["sh", "runInContainer.sh"]
```

A Docker **image** of an application contains the actual code of the tool and all required dependencies that are required to run it; e.g., the operating system, frameworks, settings, config files and libraries etc

Containers. are instantiations of images and can be thought of as lightweight virtual machines.

An image is created based on **Dockefile**

Command line:

Login: docker login registry.gitlab.com

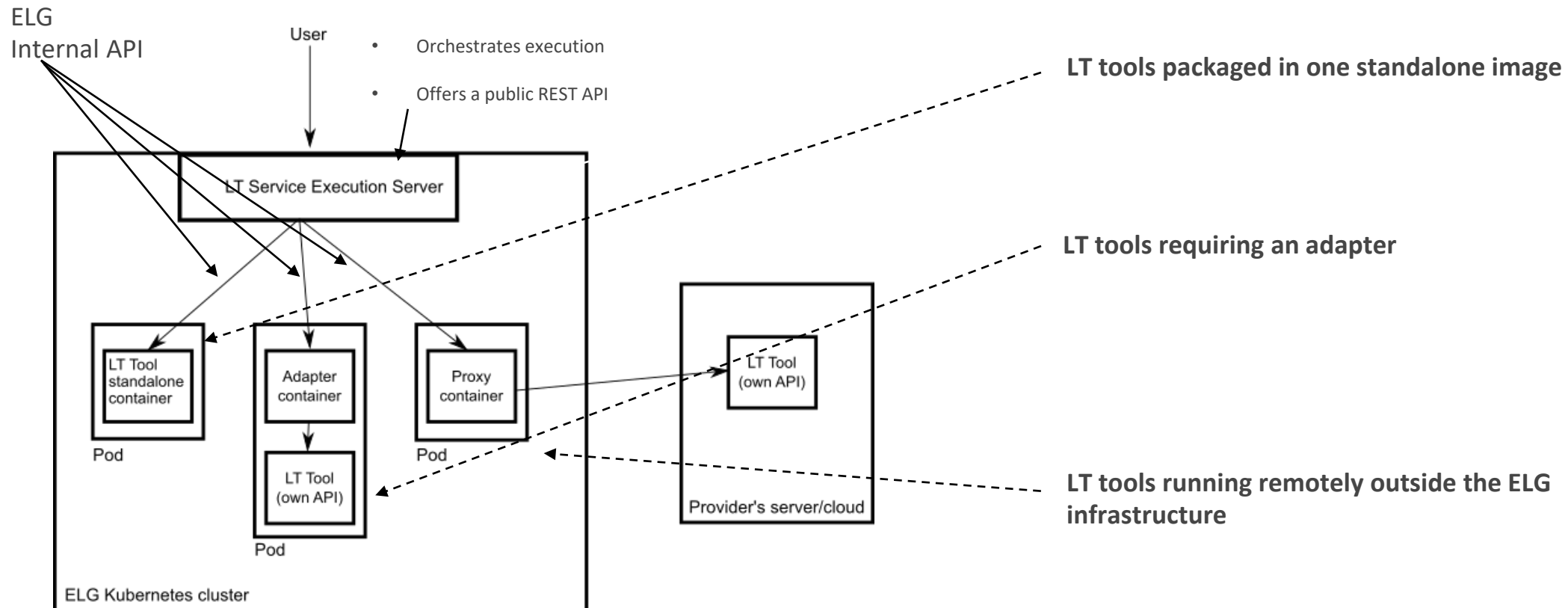
Create: docker build -t registry.gitlab.com/ilsp-nlpli-elg/elg-ilsp-lt-services .

Push: docker push registry.gitlab.com/ilsp-nlpli-elg/elg-ilsp-lt-services

Id/location of the image

Provide an ELG-compatible LT service: Technical Requirements

- Three integration options are available.



Provide an ELG-compatible LT service

- **Documentation:** https://european-language-grid.readthedocs.io/en/stable/all/3_Contributing/Service.html?highlight=boot#dockerize-your-service
- We provide examples on how to Dockerize
 - a Python-based tool
 - <https://gitlab.com/european-language-grid/usfd/simple-python-tokeniser>
 - a Java-based tool
 - <https://gitlab.com/european-language-grid/usfd/elg-spring-boot-starter>

Provide an ELG-compatible LT service

Provide Metadata

The screenshot shows the European Language Grid website interface. The top navigation bar includes the logo, 'RELEASE 2', and menu items: Technologies, Resources, Community, Events, Documentation, and About ELG. A user profile for 'Dimitrios Galanis' is visible. A dropdown menu is open under 'My grid', showing options: 'Interactive editor', 'Validate XML', and 'Upload XML'. Below the menu, there are six cards representing different resource types: 'Service or Tool', 'Corpus', 'Language description', 'Lexical/Conceptual resource', 'Project', and 'Organization'. Each card has a 'Go to form' button.

If a metadata record is syntactically valid then you can **submit it for publication**

Option A: interactive editor

The screenshot shows the 'Interactive editor' form. It includes a header with the European Language Grid logo and navigation links. A teal navigation bar contains 'Add items', 'My items', 'Feedback', 'Administration', and 'Go to catalogue'. An 'Info' box lists instructions for saving drafts and submitting records. The form has tabs for 'LANGUAGE RESOURCE/TECHNOLOGY', 'TOOL/SERVICE', and 'DISTRIBUTION'. A 'Work in progress' checkbox is present, along with 'Save draft' and 'Save' buttons. The 'IDENTITY' section contains fields for 'LRT name' (with value 'test tool'), 'language' (with value 'English'), and a 'Fill in' button. The 'CATEGORIES' section contains a field for 'LRT identifier' and another 'Fill in' button. The 'CONTACT' section contains a field for 'LRT short name' and another 'Fill in' button.

Option B: upload metadata

The screenshot shows the 'Upload metadata' form. It includes the same header and navigation as the previous screenshot. A teal navigation bar contains 'Add items', 'My items', 'Feedback', 'Administration', and 'Go to catalogue'. The form has tabs for 'VALIDATE XML FILES', 'UPLOAD SINGLE ITEM', and 'UPLOAD MULTIPLE ITEMS'. A message states: 'You can upload one xmi file each time. It is highly recommended that you validate your XML file against the ELG schema before you proceed.' There are two checkboxes: 'Work in progress' and 'ELG-compatible service'. A 'Drag & Drop your file or Browse' button is at the bottom.

Provide an ELG-compatible LT service

- The LT Service is assigned to two “validators”
 - A: Technical, metadata
 - B: Legal
- The LT tool is deployed to the ELG platform (@kubernetes) by validator **A**
- The LT Service is tested by **A** and the **LT provider**
 - Test via trial UIs
 - Check container logs, troubleshoot ...
- When both validators approve the tool then it is **automatically published** to the catalogue.

Provide an ELG-compatible LT service



EUROPEAN
LANGUAGE
GRID

RELEASE 2

My grid

Dimitrios Galanis

[Technologies](#) [Resources](#) [Community](#) [Events](#) [Documentation](#) [About ELG](#)

Add items

My items

Feedback

Administration

Go to catalogue

Clear all filters

Items

+ Tool/Service (7)

Status

+ published (7)

7 search results

	Resource name	Actions	Status
<input type="checkbox"/>	ILSP Machine Translation Service for modern Greek to English 0.4.0 Created: 28 February 2020 Updated: 07 December 2020 Tool/Service	Actions + Create new version	published
<input type="checkbox"/>	ILSP Machine Translation Service for English to modern Greek 0.4.0 Created: 28 February 2020 Updated: 07 December 2020 Tool/Service	+ Copy record Request to unpublish	published
<input type="checkbox"/>	ILSP opinion mining for customer reviews in Greek 1.0.0 Created: 28 February 2020 Updated: 30 November 2020 Tool/Service	Actions	published
	ILSP event extractor for physical attacks in Greek 1.0.0		