• ELG Platform Architecture
• ELG for LT consumers
• ELG for Providers of LT
• ELG in the wider LT and AI Ecosystem
ELG Platform Architecture

- Three-layer architecture
- All layers built with robust scalable, reliable, widely used technologies
- Ability to scale with the growing demand and supply of resources
- Laying the foundations for interoperable data and services spaces
ELG Platform Base Infrastructure

- Docker containers for all services and applications which comprise the ELG platform
- Kubernetes for container orchestration
- Storage solution, S3 compatible, for language data/resources and other related content
- Supporting tools for development and management of ELG software
**ELG Platform Backend**

- ELG platform repository and catalogue
  - Application for managing metadata records according to the ELG publication lifecycle policies
- Database and indexing mechanisms
- LT execution server
  - Invokes functional services
  - Flexible ways of integration
- User management
  - Different user categories and roles
  - Support and monitoring mechanisms
ELG Platform Frontend

- GUIs for catalogue application
  - User interactions, e.g., browsing, search
  - Providing and editing descriptions (metadata)
  - Admin interfaces, e.g., for validating submitted resources
- User interfaces for trying and testing functional services
- Code samples
- Content management system for ELG related content and information
On the data consumer side

- Consumers can **search and browse** the ELG catalogue
  - for **different types of data**, language processing services, related projects and organisations in Europe
  - using simple and advanced search: **facets for resource type, language, service function, license, related entities**
- Download data (depending on access conditions)
- Check number of views/downloads
- Check what is forthcoming in terms of data and services
On the services consumer side

- Consumers can **try out and test** language processing services
  - registration/authentication is required
  - daily quotas apply
- **Call a service** from the command line directly (via its common REST API) and integrate it in their own workflows
- View code samples
- Current APIs support
  - machine translation (MT)
  - information extraction (IE)
  - speech recognition (ASR)
  - speech synthesis (TTS)
  - text classification
- Consumers can use a Python-based API for accessing the ELG catalogue, searching and directly fetching datasets to feed them into, e.g., their model training pipeline
On the provider side

- Language data/resources/services providers can register metadata descriptions by
  - registering and getting authenticated as providers
  - using a **formal metadata schema**
  - uploading a **schema compliant XML file**
  - using an **interactive metadata editor**
- Get support through the **online documentation**
- View their resources in their **dashboard**
  - including their status according to the ELG publication lifecycle
  - lifecycle: draft → syntactically valid → submitted → published
- Upload data files
- Batch uploading also supported
- “Claim” metadata records and enrich them
On the service provider side

- LT service providers need to provide a **Docker image** with their LT tool or service
- Docker images have to be uploaded to a **Docker registry** (e.g., GitLab, DockerHub)

**Three different options:**

- LT tool packaged in one Docker image exposing an ELG-compliant endpoint
- LT tool running outside the ELG infrastructure – proxy image with one or more ELG-compliant endpoints
- LT tool requiring an adapter – adapter image exposes an ELG-compliant endpoint as proxy
ELG in the wider LT and AI ecosystem

• First step has been to **build bridges** to existing platforms/infrastructures
  • Mainly in terms of metadata-based descriptions
  • First bridges with
    • **ELRC-SHARE**, of the European Language Resource Coordination initiative
    • **LINDAT-CLARIAH**
  • Both repositories are automatically harvested once a week
    • Based on **open protocols** (OAI-PMH)
    • Respecting their own policies
• **Mapping** of the underlying ELG & AI4EU ontologies
• In order to support cross-platform search
Come and visit us at the ELG booth!