ELIXIR Human Data Use Case

Mikael Borg, ELIXIR Sweden

www.elixir-europe.org/excelerate
What is the EGA?

The EGA is an Archive for human genomic data that should have controlled access because it allows identifying individuals

- Data is provided by research centers and health care institutions
- Access is controlled by Data Access Committees
- Data requesters are researchers from other research or health care institutions

Our goal is to foster responsible data reuse for the good of science and all humankind

https://www.ebi.ac.uk/ega
https://ega-archive.org
EGA: European Genome-phenome Archive

- Repository that promotes the distribution and sharing of genetic and phenotypic data consented for specific approved uses but not fully open, public distribution.
- All types of sequence and genotype experiments, including case-control, population, and family studies.
- Study & Metadata searchable
- Shares parts of the Metadata model with ENA
EGA

- Data Access Agreement
  - Defined by the dataset owner
- Data Access Committee – DAC
  - Decided by the dataset owner
Pre-EXCELERATE EGA

EGA future is
- Distributed
- Cloud based
- Accessible
Post-EXCELERATE EGA
WP9 Overview

- **Task 9.1**
  - Support for large-scale submissions of -omic data and sample metadata to the EGA
  - Portable submission toolkit ("EGA-in-a-box", "Local EGA")

- **Task 9.2**
  - Support secure integration of EGA data to downstream project client websites
  - Access management workflow support
  - ELIXIR and EGA access integration

- **Task 9.3**
  - Large-scale data mirroring support
  - EGA data access authorization integration
  - Data access APIs
Submission UI

UI on top of the Submission API

Available at https://ega-archive.org/submitterportal-test/

User/pwd: ega-box-148 / ELIXIR-Demo
Access management workflow support (9.2.2)

- Resource Entitlement Management System (REMS)
  - Electronic tool for the management of access rights to research data
  - Supports federated identity authentication for both parties
  - Developed at the CSC Finland

- Apply for an access to a resource
  - Login using their home organisation's user ID and password
  - Fill in an electronic data access application
  - Agree to the terms of use for this dataset

- Circulate data access application for approval
  - Provides reports on applications and approved access rights
Access management workflow support (9.2.2)

Requester

1. Apply

2. Grant

3. Update / Sync

EGA

4. Inform

DAC
Access management workflow support (9.3.2)

1. Apply
2. Grant
3. Update / Sync
4. Inform
5. Login / Request
6. Check Permissions
7. Stream Data
8. Mount via FUSE

Secure cloud access demo movie: https://www.youtube.com/watch?v=gwkZzz1e8bY
Task 9.1.2: Local EGA

EGA

ELSI

APIs

Authentication

Authorization

Sync

on metadata

Metadata

submission

Data submitter

Local EGA

ELSI

API

Data files

Others (services, users…)

Discover & Request Access
Local EGA Goals

• Establish a “portable” core code base, that supports
  1. **Submitter Registration** at CEGA, specifying LEGA instance to store data
  2. **Dataset Ingestion and Archiving**
  3. **Requester Registration & Dataset Access**

  https://github.com/NBISweden/LocalEGA

• Optional processes
  1. On-the-fly decryption data access in secure cloud instances (9.3.2)
  2. Mirroring/caching of CEGA hosted datasets at a LEGA instance (9.3.1)
  3. *Sync of metadata between CEGA and LEGA instance*

• Hardware agnostic

• Nordic partners working on establishing Local EGA instances
  – Coordinated in the Nordic *Tryggve project*
    • *Infrastructure for collaboration on sensitive personal data for biomedical research*
Data mirroring (9.3.1)

- Series of Python scripts in Luigi workflows

github.com/CSCfi/lega-mirroring
Data access APIs (9.3.3)

- Data discovery
  - Beacon API
  - Matchmaker API

- Data delivery
  - Reads and variants API
  - Streaming API

- Secure storage at rest
  - File formats encryption container API

- Security working group
  - Software security task team - AAI recommendations
Acknowledgements

**WP9 Partners**
ELIXIR Finland
ELIXIR Sweden
ELIXIR Norway
ELIXIR Denmark
ELIXIR Estonia
ELIXIR Netherlands
EMBL EBI
ELIXIR Spain

**Other nodes contributors**
ELIXIR Switzerland
ELIXIR Belgium
ELIXIR Luxembourg
ELIXIR Slovenia
ELIXIR Czech Republic
ELIXIR France

**The ELIXIR Hub**
ELIXIR - Excelerate

WP6: Marine metagenomic infrastructure services as driver for research and industrial innovation

WP8: Integrating ELIXIR infrastructure for Rare Disease research

WP1, Tools Interoperability & Registry

WP2, Community Benchmarking

WP4, Compute, Data access and exchange services

WP5, ELIXIR Interoperability Backbone

WP3, Data Resources

WP7: Integrating Genomic and Phenotypic Data for Crop and Forest Plants

WP9: An ELIXIR framework for secure archiving, dissemination and analysis of human access-controlled data

WP10, ELIXIR Node Capacity

WP11, Training

Efficient organisation

World-leading research infrastructure services

Bioinformatics capacity and competence across Europe
Submission API and Submission UI (9.2.1)

- Creating, editing and validating submissions prior to submission
- EGA submission
- OpenAPI documentation available at
  - https://ega-archive.org/openapi/currentspecs/submission-api.yaml
  - Supports JSON and XML
User registration & Data Archiving

Central EGA

Create INBOX for user X

INBOX created for user X

Archive FILES for user X

FILES archived for user X or ERROR

Make submission

Register Submitter

Submitter

Local EGA instance

VAULT

User X INBOX

Re-encrypt

Submitter
Local EGA Architecture

Host 1
- API
- Inbox service
- Public SFTP

Host 2
- Worker
- Message Queue
- Vault key

Host 3
- Vault

Inbox service
Public SFTP

Vault key

.mount

Inbox
Vault
ELIXIR and EGA access integration (9.2.3)

**Diagram Description:**

1. **OAuth2 authorisation Server**
   - EGA AAI
   - Authentication (e.g. ELIXIR AAI)

2. **Browser**
   - HTTP
     - "Let me access my files on a VM"
   - HTTPS OAuth2 authorisation request
     - "Please authenticate this user"
   - HTTPS OAuth2 authorisation code
     - "Reference number 12345"

3. **OAuth2 Client**
   - HTTPS OAuth2 Access token request
     - "Tell me about the user with reference number 12345"

4. **OAuth2**
   - HTTPS OAuth2 Access token and refresh token
     - "He is Bob"

5. **FUSE server** (managed and run by a trusted cloud provider)
   - HTTPS request Access token
     - "What are Bob’s permitted datasets?"
   - HTTPS response
     - "Bob’s permitted datasets are EGAD0123, EGAD0234.."
Data access API