Session 4.1: CV Manager - Overview of how to design, build, test and deploy a CESSDA tool
Outline

» What is Controlled Vocabulary (CV) Manager
» Vocabulary Manager Architecture
» Vocabulary Manager in Google Cloud
» Deployment via Jenkins
» Dockerize Vocabulary Manager
» Continuous Integration
» Vocabulary Manager
CV Manager

CV Manager is a tool to manage controlled vocabularies (CVs).

It is a tool that provides the following:

» Create, Edit, Review and Publish CVs
» CVs Versioning
» CVs Language Translation
» Import and Export of CVs
» Rest endpoint to access published CVs and codes
Analysis unit

Describes the entity being analyzed in the study or in the variable.

Household family in time. If not known whether the analysis unit is "Family" or "Household family", use "Family".

Family Two or more people related by blood, marriage (including step-relations), adoption or fostering and who may or may not live together (National Community Services Data Dictionary, Vers 3, AIHW, 2004). For example, used when researching the extent to which people provide support and assistance for their relatives.
**DDI Controlled Vocabulary for Analysis unit**

- **Title**: Analysis unit
- **Definition**: Describes the entity being analyzed in the study or in the variable.

<table>
<thead>
<tr>
<th>Code</th>
<th>Descriptive term (en)</th>
<th>Definition (en)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual</td>
<td>Individual</td>
<td>Any individual person, irrespective of demographic characteristics, professional, social or legal status, or affiliation.</td>
</tr>
<tr>
<td>Organization</td>
<td>Organization</td>
<td>Any kind of formal administrative and functional structure - includes associations, institutions, agencies, businesses, political parties, schools, etc.</td>
</tr>
<tr>
<td>Family</td>
<td>Family</td>
<td>Two or more people related by blood, marriage (including step-relations), adoption or fostering and who may or may not live together (National Community Services Data Dictionary, Vers 3, AIHW, 2004). For example, used when researching the extent to which people provide support and assistance for their relatives.</td>
</tr>
<tr>
<td>Family, Household, Family</td>
<td>Household family</td>
<td>A more specific term, refers only to related people who live in the same household at a point in time. If not known whether the analysis unit is &quot;Family&quot; or &quot;Household family&quot;, use &quot;Family&quot;.</td>
</tr>
<tr>
<td>Household</td>
<td>Household</td>
<td>A person or a group of persons who share the same dwelling unit and common living arrangements. These terms may also be referred to as &quot;households&quot;.</td>
</tr>
</tbody>
</table>
CV Manager - Rest Endpoints

Get all CVs
../v1/vocabularies

Get a CV (parameters: vocabulary, language, version)
../v1/vocabulary/

Get all codes (parameters: vocabulary, language, version)
../v1/codes/

Get a code (parameters: vocabulary, language, version)
../v1/code/

Get a list of suggestions
/v1/suggest/vocabulary/{vocabulary}/version{version}/language/{language}/limit/{limit}/query/{query}
CV Manager Components

CV Manager is a web-based application, developed with Java using Spring Boot and Vaadin 8 Framework

Dependencies:

» DDI-FlatDB – a lightweight framework for heterogeneous DDI sources
» Elasticsearch
» Java EE Servlet container (Tomcat)
» GESIS’ usermanagement library - managing users, agencies, roles
» MySQL
Architecture

**Server-side**
- Vaadin Client-Side Engine (Widget Set)
- Spring Framework
- Vaadin Framework
- Vaadin Components
- Business Logic
- Java Code
- Servlet
- Spring-Boot Web Application
- HTTP(S)
- REST
- REST

**Client-side**
- Vaadin Client-Side Engine (Widget Set)
- Client-Side UI (HTML 5, CSS, Java Script)
- GWT
- HTTP(S)
- AJAX

**Data Management**
- DDI FlatDB
- RDBMS (My-SQL)
- Elasticsearch
- Analyzed CV Details Index
- CV Details
- XML
- User Details
- XML
- Agency Details
- SVOS-RDF
- CV-Workflow
- CV-Versioning
- XML Snippets

**Java**
- Java Code

**Web Technologies**
- Vaadin Client-Side Engine (Widget Set)
- Vaadin Components
- Client-Side UI (HTML 5, CSS, Java Script)
- GWT
- HTTP(S)
- AJAX

**Other Technologies**
- REST APIS
- REST APIS
CESSDA Technical Infrastructure

CESSDA Technical Infrastructure “requirements”
» Infrastructure as code (IaC)
» Docker containers
» Kubernetes @ Google cloud (docker orchestrator)
» Continuous Deployment pipeline (Jenkins)
» BitBucket Git

Benefits:
» Scalable cluster (horizontal and vertical)
» Docker containers are managed by K8S cluster manager
» Portable applications (IaC)
Deployment Dockerized

Spring-Boot Web Application

CV-Manager
  - Vaadin Framework
  - GESIS DDI FlatDB
  - GESIS User-Management
  - Dependencies

Maven Project

Fabric8.io docker-maven-plugin
  - Dockerfile
  - $ mvn docker:build

CV-Manager Docker Image

$ gcloud docker push

Google Container Registry (grc.io)

CV-Manager Docker Image
Architecture - Google Cloud (GC), Dockerized

Google Cloud Shell

$ docker build
$ gcloud docker push

$ kubectl
$ gcloud

CV Manager App
Docker Container

DDI-FlatDB
Docker Container

Elasticsearch
Docker Container

MySQL
Docker Container

Persistent volume for MySQL

Google Container Engine (GKE)

Google Container Registry (grc.io)

CV-Manager
Docker Image

DDI-FlatDB
Docker Image

Google Cloud (GC)

Dockerized
Deployment

<table>
<thead>
<tr>
<th>Checkout SCM Prepare Deployment</th>
<th>Build Project Sonar Scan</th>
<th>Sonar-Qube Get Gate Status</th>
<th>Push Docker Image (GC)</th>
<th>Check Requirement &amp; Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get Jenkins Source Control Management (SCM). Prepare the deployment environment, credential for GC Kubernetes.</td>
<td>Clean and compile the project. Apply SonarQube Scanner to the code.</td>
<td>Get the information of the application/code health and quality from the SonarCube Gate status.</td>
<td>Push docker image to Google Container Registry (gcr.io).</td>
<td>Load GC environment, Prepare the GC Container Cluster, Perform Kubernetes deployment.</td>
</tr>
</tbody>
</table>

**Jenkins-Pipeline**
Continuous Integration

CV-Manager Code
- GESIS DDI FlatDB
- GESIS User-Management
- Dependencies

GESIS User-Management

 GIT Version Control

Push Code

GESIS MAVEN REPO
- jar
- jar

Pull code, run test, build
deploy

JENKINS-PIPELINE

GESIS JENKINS

JENKINS-PIPELINE

Push Code

Bitbucket

Git Version Control

 pull code, run test, build

Google Container Engine (GKE)

Docker build, gcloud docker push

Deployment (via gcloud and kubectl)

Kubernetes Cluster

CV-Manager Dev
- Docker Container

CV Manager Staging
- Docker Container

CV Manager Production
- Docker Container

CV-Manager Docker Image

Google Container Registry (gcr.io)

Sonar scans code health & quality, checks SonarQube Gate status

Docker Container

CV Manager Dev

GKE

GESIS

JENKINS

GESIS

GITLAB

GESIS

GITLAB

GESIS

CV-Manager Code

Sonic qube

GESIS

GESIS DDI FlatDB

GESIS User-Management

Pull code, run test, build

Get Dependencies

CESSDA

JENKINS

CESSDA JENKINS-PIPELINE

GESIS MAVEN REPO

SonarQube

Deployment (via gcloud and kubectl)
### VM instances overview

A Kubernetes cluster is a managed group of uniform VM instances for running Kubernetes. Learn more

<table>
<thead>
<tr>
<th>Name</th>
<th>Location</th>
<th>Cluster size</th>
<th>Total cores</th>
<th>Total memory</th>
<th>Notifications</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>cessda-cvmanager-dev-cc</td>
<td>europe-west1-b</td>
<td>1</td>
<td>2 vCPUs</td>
<td>7.50 GB</td>
<td><img src="image" alt="Node upgrade available" /></td>
<td>env: dev, project: cessda-cvmanager_connect</td>
</tr>
<tr>
<td>cessda-cvmanager-live-cc</td>
<td>europe-west1-b</td>
<td>1</td>
<td>1 vCPU</td>
<td>3.75 GB</td>
<td></td>
<td>env: live, project: cessda-cvmanager_connect</td>
</tr>
<tr>
<td>cessda-cvmanager-staging-cc</td>
<td>europe-west1-b</td>
<td>2</td>
<td>3 vCPUs</td>
<td>11.25 GB</td>
<td><img src="image" alt="Node upgrade available" /></td>
<td>env: staging, project: cessda-cvmanager_connect</td>
</tr>
<tr>
<td>cessda-dataserver-ea-dev-cc</td>
<td>europe-west1-b</td>
<td>1</td>
<td>2 vCPUs</td>
<td>13.00 GB</td>
<td><img src="image" alt="Node upgrade available" /></td>
<td>env: dev, project: cessda-dataserver_connect</td>
</tr>
<tr>
<td>cessda-mgmt-live-cc</td>
<td>europe-west1-b</td>
<td>4</td>
<td>4 vCPUs</td>
<td>15.00 GB</td>
<td></td>
<td>env: live, project: mgmt_connect</td>
</tr>
<tr>
<td>cessda-pasc-dev-cc</td>
<td>europe-west1-b</td>
<td>5</td>
<td>8 vCPUs</td>
<td>30.00 GB</td>
<td></td>
<td>env: dev, project: cessda-pasc_connect</td>
</tr>
<tr>
<td>cessda-pasc-live-cc</td>
<td>europe-west1-b</td>
<td>5</td>
<td>8 vCPUs</td>
<td>30.00 GB</td>
<td></td>
<td>env: live, project: cessda-pasc_connect</td>
</tr>
<tr>
<td>cessda-pasc-staging-cc</td>
<td>europe-west1-b</td>
<td>5</td>
<td>8 vCPUs</td>
<td>30.00 GB</td>
<td></td>
<td>env: staging, project: cessda-pasc_connect</td>
</tr>
<tr>
<td>ovmanager-cc</td>
<td>europe-west1-b</td>
<td>1</td>
<td>2 vCPUs</td>
<td>7.50 GB</td>
<td><img src="image" alt="Node upgrade available" /></td>
<td>env: staging, project: ovmanager_connect</td>
</tr>
</tbody>
</table>
Lessons learned

» We can now
  ○ develop web-based, platform independent Java software
    ■ Stack:
      ● Storage: Database, DDI-FlatDB
      ● Service: Java Spring IO Framework
      ● User Interface: Vaadin 8/10
  ○ can create with maven docker container
  ○ can use the CESSDA pipeline to deploy docker in GC
Thanks for listening

Any Questions?