

# CESSDA Research Infrastructure Technical Framework



John Shepherdson  
IASSIST 2016, Bergen Norway

cessda

# Structure

Introduction to CESSDA

Technical Framework (Phase 1)

What's next?

# What is CESSDA?

Consortium of European Social Science Data  
Archives (<http://cessda.net>)

Currently 15 members, 1 observer

Aim: Provide a pan-European Research  
Infrastructure for the Social Sciences

# CESSDA Objectives

- coordination of the network of European data service providers and promotion of the results of social sciences
- facilitating researcher access to important resources of relevance to the European social science research agenda regardless of the location of either researcher or data
- work continuously to include further data sources from Europe and beyond, into the infrastructure

# CESSDA Objectives

- provide training within CESSDA and beyond on best practices surrounding operational processes and data management
- promote and facilitate wider participation in CESSDA
- the development and coordination of standards, protocols and professional best practice pertaining to the preservation and dissemination of data and associated digital objects

# CESSDA Current Activity

## Horizon 2020 Projects

- CESSDA [Strengthening and Widening \(SaW\)](#)
- [Synergies for Europe's Research Infrastructures in the Social Sciences \(SERISS\)](#)
- [Big Data Europe \(BDE\)](#)

## Work plan Projects

- Expert Seminars
- Metadata Harvester
- Metadata Management
- Data Access Policy
- **Technical Framework**

# Technical Infrastructure Required

In past, relied on members ('Service Providers') to develop and host standalone products

Now a company under Norwegian Law and in process of becoming an ERIC ([European Research Infrastructure Consortium](#))

**=> Greenfield site for technical development**

# Protection of Assets is Vital

Ensure CESSDA has access to

- source code
- configuration files
- technical documentation

that underpin its products and services



# CESSDA's common interoperability characteristics

1. **Loosely coupled but coordinated** - enable Service Providers to retain independence, yet fully interact in an integrated service
2. **Sustainable** - enable medium and long term investment and business change decisions to be made.
3. **Extensible** - enable additional services to be built on or around it, including adapting to changing functional requirements over time.
4. **Maintainable** - enable components to be updated when IT specifications change.
5. **Standards based** - enable the coordinated and planned change to all the coupled, but coordinated, services.

# What is the Technical Framework?

A guide for the development of the various (software) products and services that form part of the CESSDA Research Infrastructure

- promote good practice for software development
- meet common interoperability characteristics

Infrastructure for Development, Staging and Production

- harmonise development tool chain for SPs

# Partners and Activities

Partners DANS, GESIS, NSD, SND, UKDS (lead)

All partners

- Develop and publish Technical Architecture

UKDS and NSD

- Specify, install and configure development infrastructure

# Infrastructure for Development (to date)

- Email Identity c/w Avatar
- Collaborative document creation/management
- Source code management
- Issue Tracker
- Wiki
- File storage
- Chatroom
- Continuous Integration Server
- System Availability Monitor
- Micro-service registry

# Quality Control

## Software Maturity Levels

- ensure quality of the research infrastructure is maintained
- guidance on minimum, expected and excellent standards
- originally based on [NASA's RRLs](#)
- revised in light of 'Capability Development Model' from [CESSDA SaW](#) project

# Common Interoperability Characteristics

Achieved by:

- REST APIs c/w API design standards
- Architectural standards
- Common development environment
- Adoption of [12 Factor App](#) principles
  - methodology for building Software-as-a-Service apps

# 1. Loosely coupled but coordinated

Adopt (micro) services architecture based on RESTful web service APIs

- provides a mechanism for reusing and combining software artefacts

See also 12 factor app, number 7 ([Port binding - Export services via port binding](#))

## 2. Sustainable

The provision of common standards

- Technical Architecture document

Common development and test environment

- via the technical infrastructure

Deployment environment

- via extensions to the technical infrastructure

Central source-code repository

See also 12 factor app, number 1 ([Codebase - One codebase tracked in revision control, many deploys](#))



# 3. Extensible

Service API is key

- Integration point for new services
- Combination point for building new features

Version and support two versions simultaneously

- Allows services to evolve, without breaking contract provided to consumers.

See also 12 factor app, number 8 ([Concurrency - Scale out via the process model](#))

See also 12 factor app, number 9 ([Disposability - Maximize robustness with fast startup and graceful shutdown](#))

# 4. Maintainable

Again, service API is key

- implementation of a service can be changed as required, to take advantage of developments in software technology
- location of services can be changed as required, to take advantage of developments in hardware technology.

See also 12 factor app, number 2 ([Dependencies - Explicitly declare and isolate dependencies](#))

# 5. Standards Based

- Provision of common architectural standards (via Technical Architecture)
- A consistent (in both the calling and return structures and formats) and versioned API

See also 12 factor app, number 4 ([Backing services - Treat backing services as attached resources](#))

# Software Acceptance Criteria

- Documentation
- Intellectual property issues
- Extensibility
- Modularity
- Packaging
- Portability
- Standards compliance
- Support
- Verification and testing
- Internationalisation and Localization

# Software Maturity Levels

- 1 - Initial usability;** software use is not recommended.
- 2 - Use is feasible;** the software can be used by skilled personnel but with considerable effort, cost and risk.
- 3 - Use is possible by most users;** with some effort, cost, and risk. A risk assessment should be made before use.
- 4 - Software is usable;** with little effort, cost, and risk.
- 5 - Demonstrable usability;** there is clear evidence that the software is widely used by many users.



# Maturity Modelling - More Info

Mike Priddy, CESSDA SaW project

Wednesday 15:45

Session 1G - Data Services: Setting up and evaluating

‘Maturity Model For Assessing Data Infrastructures – Cessda As Example’

# What's Next?

## Proposed Phase 2

- more dissemination - understanding and acceptance
- Staging and Production environments
- extend 'how to' documentation for developers and admins





cessda

A pan-European Research Infrastructure for the  
Social Sciences