

# **MIXED**

## **Architecture**

Author: Kris Klykens  
[kris.klykens@dans.knaw.nl](mailto:kris.klykens@dans.knaw.nl)  
[kris.klykens@daidalos.nl](mailto:kris.klykens@daidalos.nl)

Version: 0.2

Date: 21-06-2007 15:04

MIXED is a project of DANS.

## **DANS – Data Archiving and Networked Services**

Postbus 93067  
2509 AB The Hague, Netherlands

Gebouw Borneo  
Anna van Saksenlaan 51  
2593 HW The Hague, Netherlands

<http://www.dans.knaw.nl>

### **History**

Version	Author	Distributed to	Date
v0.1	Kris L. Klykens	Henk Harmsen, Dirk Roorda, Henk van den Berg (DANS)	11 June 2007
V0.2	Kris L. Klykens	Henk Harmsen, Dirk Roorda, Rutger Kramer, Laurents Sesink, Henk van den Berg (DANS); Wolfgang Beer (SCCH)	21 June 2007

# Introduction

In order to avoid duplicate information, the real introduction to MIXED can be found by accessing the MIXED project page<sup>1</sup>.

In this document, the architecture for MIXED is documented on 3 levels:

- Enterprise architecture: high level description of the related objects within MIXED, without the effort of structuring the specific different aspects
- Information architecture: expression of the information-model for MIXED, such as interactions and data-flow
- Technical architecture: the design of MIXED, focused on interaction with software developers

This architecture must be seen in regard to the project type, being R&D, and the development process. The DANS-specific method combines XP and agile, and in this perspective, there will be a continual refinement of the architecture, rework when necessary, and a constant attention to the overall design of MIXED.

The design of the MIXED architecture is based on the DANS principles, the initiation document of MIXED, and many discussions with Rutger Kramer, Laurents Sesink, Dirk Roorda and the development team. Also, this design builds further on following documents:

- Reference Model for an Open Archival Information System (OAIS), CCSDS 650,0-B-1, January 2002
- Testbed Digitale Bewaring, ICTU 2003
- Nederlandse Overheid Referentie Architectuur (NORA 2.0), e-Kenniscentrum, 25 april 2007

These documents can also be found on the references page of MIXED<sup>2</sup>.

The goal of this document is to be as brief as possible, yet complete.

---

<sup>1</sup> The MIXED Homepage is available at <http://mixed.dans.knaw.nl>

<sup>2</sup> The References page can be found at <http://mixed.dans.knaw.nl/node/4>

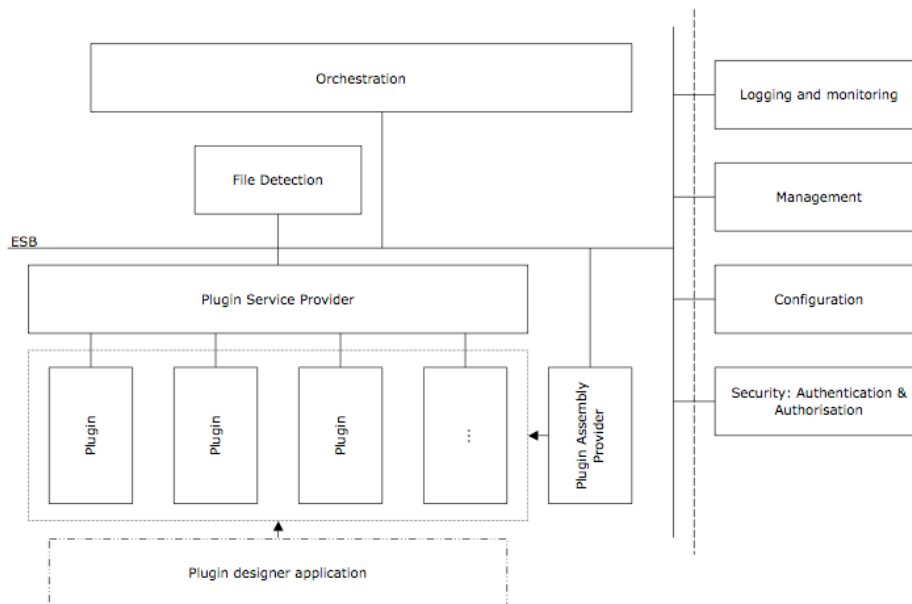
# Enterprise architecture

## *DANS Principles*

MIXED can not be seen as an island within a broad ocean of software products. The context of DANS imposes certain specific guidelines which should be incorporated in this design:

- DANS promotes long-term preservation of research-data, also enabling access to this data, and aims towards open access
- DANS works together with researchers, and promotes collaboration
- DANS promotes collaboration between institutes
- DANS adheres the idea of open source software and open standards
- DANS imposes strict guidelines towards quality of data

## *Schematic overview*



## *Functional components*

### **ESB**

The model used for the architecture is an Enterprise Service Bus. This refers to the use of a construction, which allows base functions to be held within their constituent parts, in contrary to the more classical enterprise application integration.

Furthermore, ESB provides the features (but does not implement) a service oriented architecture.

The ESB will provide a common layer, based on standards, providing the functionalities being used within loose binding.

## **Orchestration**

The orchestration layer allows MIXED to implement its 'business logic'. The collected information about the data-format used (provided by the File Detection component), and the specific information about the available plugins makes it possible to provide a specific choice about the plugin to use for a specific action.

The Orchestration component will enable 3 functions to the outside world:

- file detection: what file-type is a given file
- convert to: convert an xml-file to a specified format
- convert from: convert an xml-file to m-xml

## **File detection**

This component will enable to classify the specific vendor-type of dataset-files based on a given bitset.

Note: within this component, a specific business logic can be applied to the various sources of the specific functionalities.

## **Plugin designer application**

This application will allow the re-use of specific atomic plugins and provide a designer interface for publishing specific collections of these atomic plugins.

As the name clarifies, this is a designer application, and therefore holds a lower priority in the development process of MIXED. The impact of this application lies more into the management of MIXED over time, than within the project-plan 2007-2008.

## **Plugin Service provider**

The plugin service provider acts as an entry point to the plugins available at the system and locally managed. It will as well be possible for plugins to be available directly at the ESB, enabling coupling of different MIXED systems, while using all available plugins on the network.

## **Plugin**

The is the actual component, able to convert a specific file format (vendor) dataset-file to M-XML and an M-XML dataset-file to the specific file format (vendor) dataset-file.

The plugin is only aware of itself, and its capabilities.

## **Plugin Assembly provider**

Provides the Orchestration module the information needed to address the specific plugins. It collects this information from the specific plugins.

## ***Service components***

Vertically on the right of the scheme, the specific service components can be found.

These components are provided here without the specification of the precise functionalities, which will be determined and refined during the development process.

We distinguish:

- Logging (and monitoring) (note: by definition, this component provides application-wide logging, and does not supersede the logging done for development purposes of specific plugins)
- Management
- Configuration
- Security: authentication and authorization

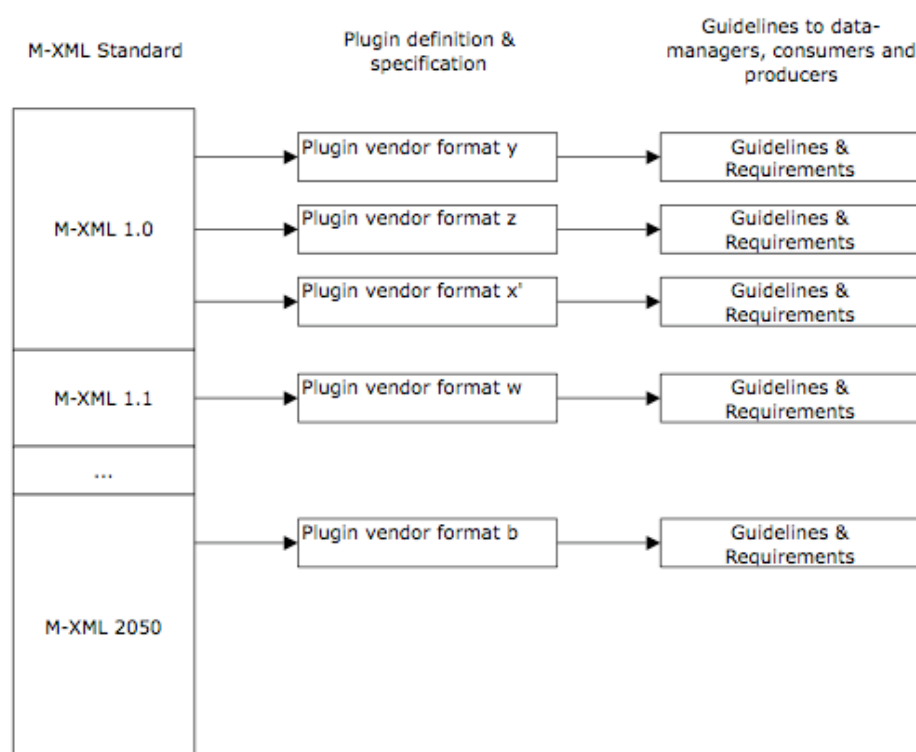
# Information architecture

## M-XML

The definition of the standard M-XML is not part of this architecture, and will be available as a separate document.

Still, it is interesting to specify the flow in regard to the actual users of the MIXED software.

In this regard, we can see the workflow as follows: using the M-XML standard in it's current version, a specific plugin will be assembled (and/or developed), which leads to a specific plugin-documentation comprising the specific data-aspects covered within this plugin. This documentation must lead to guidelines, enabling specific users of MIXED a clear view on the aspects for collaboration with MIXED.



{add specs according to OAIS on this matter}

## Use Cases

### Designed use-case for MIXED

The use-cases for MIXED can be described as follows:

- a request is handed over to convert a dataset from a specific vendor format to the intermediate M-XML-format
  - orchestration handles the incoming request
  - filedetection determines the file format of the given dataset
  - the plugin assembly provider provides information about the available plugins (internal and/or external)

- orchestration selects a specific plugin to use for the given dataset, based on business logic (e.g.: internal plugins may be favorable opposed to external, ...) and transmits the dataset to the plugin service provider
- the plugin service provider acts as a proxy to a specific plugin, and the converted dataset will be handed over to the orchestration
- the result set will be transferred to the originator of the request
- the reversed process (conversion from an M-XML dataset to a vendor specific dataset) differs only where the File Detection step is superfluous.

## ***Open Standards***

### **File types**

For file-types the mime-type specification will be used, see [rfc2822](#).

### **Communications**

The standard used for communication between components is SOAP and Attachments (see [here](#)).