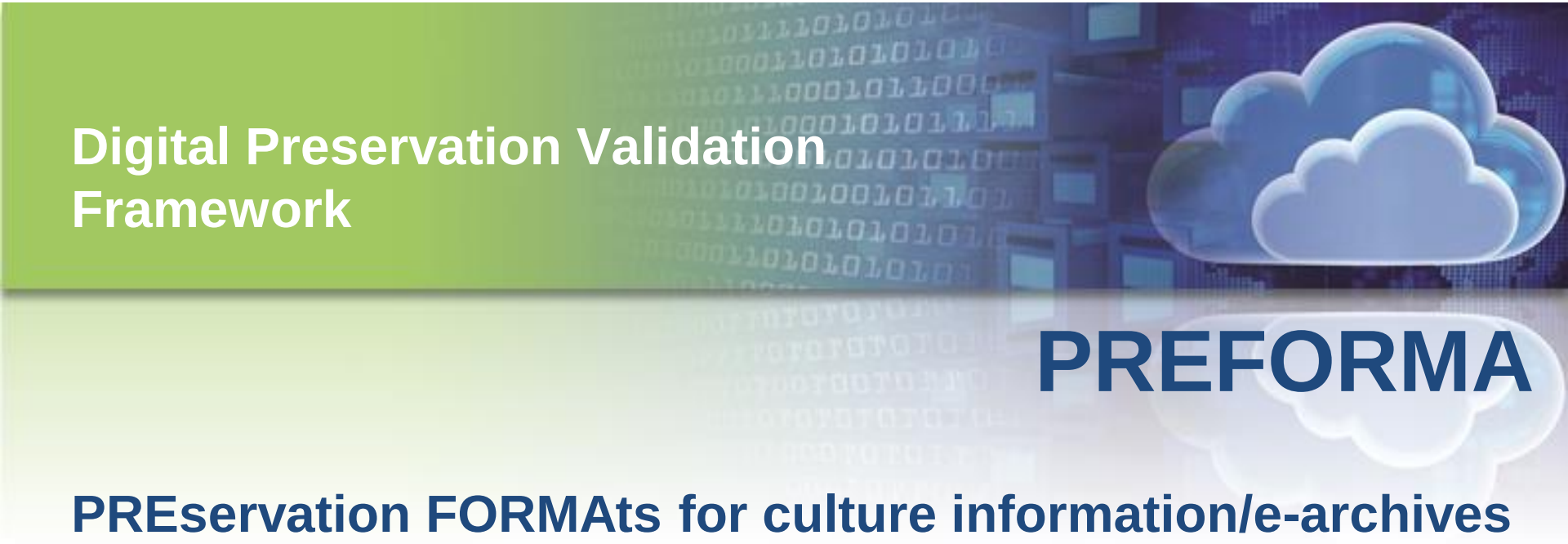
Digital Preservation Validation Framework



PREFORMA

PREservation FORMAts for culture information/e-archives

6th March 2015

Preservica
Digital Preservation

Introduction

James Carr Ph.D
Digital Preservation Architect

Ann Keen MSc.
Digital Preservation Account Manager

Preservica
Digital Preservation

Agenda

- The PREFORMA challenge
- What has been our approach?
- The technical solution
- The benefits and advantages of our solution
- How we will deliver the solution
- Future sustainability and our team

The PREFORMA challenge

The PREFORMA challenge brief:

“PREFORMA aims to establish a set of tools and procedures for gaining full control over the technical properties of digital content intended for long-term preservation by memory institutions.”

What is the PREFORMA challenge?

The PREFORMA challenge brief:

*"PREFORMA aims to establish a **set of tools** and **procedures** for gaining full control over the technical properties of digital content intended for long-term preservation by **memory institutions**."*

Success factors

We all want a PREFORMA project which:

- Brings together expert knowledge from all suppliers
- Gets the best out of complementary tools
- Is sustainable in the long term with community and commercial backing
- Provides real value to the digital preservation community
- Provides robust and quality software for users
- Provides powerful and intuitive APIs for developers

What has been our approach

To design a solution that:

- Provides significant value to the project
- Aligns with the entire preservation community
- Encourages close cooperation between all partners
- Limits the potential fragmentation of the entire project
- Leverages the strength of our organisation
- Re-uses our technology & experience

We aim to provide a solution that brings the whole project together

What is our Solution

The Digital Preservation Validation Framework is an extensible and modular system for:

- Format validation
- Conformance checking
- Metadata repair

It will form the central characterisation component of any OAIS system and allow *content producers, archivists and repository managers* to have confidence that the objects they are managing can be read and made accessible in the future

Technical Solution

As part of the design phase Preservica have provided:

Software Requirements Document (Functional Specification)

13.3 Open-Source Requirements

The open-source approach is fundamental for achieving the overall objectives of the PREFORMA challenge. The following table captures the requirements linked to developing an open-source project.

Label	Requirement	Necessity
S13.3.1	All software developed in the PREFORMA project will be released using established open source development practices with early and frequent releases of developed software and associated artefacts. Source: PREFORMA Challenge Brief v1.0	M
S13.3.2	All software developed in the PREFORMA project will be licensed under "GPL v3 or later" and "MPL v2 or later", enabling that anyone that has adopted such software has the right to freely read, use, improve and redistribute the source code for such software. Source: PREFORMA Challenge Brief v1.0	M
S13.3.3	All software developed in the PREFORMA project will be made available on an open platform, e.g. GitHub or equivalent. Source: PREFORMA Challenge Brief v1.0	M
S13.3.4	All file formats researched in the PREFORMA project will be available under licensing conditions that allow for implementation in open-source software, including allowing for implementation in open source software which is licensed under "GPLV3 or later" and "MPLV2 or later". Source: PREFORMA Challenge Brief v1.0	M
S13.3.5	All files produced in the PREFORMA project will be released under a CC-BY-SA license. Source: PREFORMA Challenge Brief v1.0	M

13.4 Speed and Capacity Requirements

13.4.1 Background

Speed and capacity requirements are crucial to any digital preservation tool being accepted by the wider digital preservation community outside of a niche research environment.
Preservica has a long history of designing systems and preservation components which meet stringent throughput requirements.

Sometimes throughput and speed can get confused but it is important to recognise that they are different things. Throughput is a measure of how much content the system can process in a given time period (e.g., it can ingest 50TB in a day). Speed is a measure of the elapsed time, start-to-finish, of an individual process or workflow (e.g., this particular SIP took 6 hours to ingest).

Speed is of direct interest only if there is some reason to want to get a particular task completed before some deadline. Otherwise, a better measure of a system's performance is its throughput.

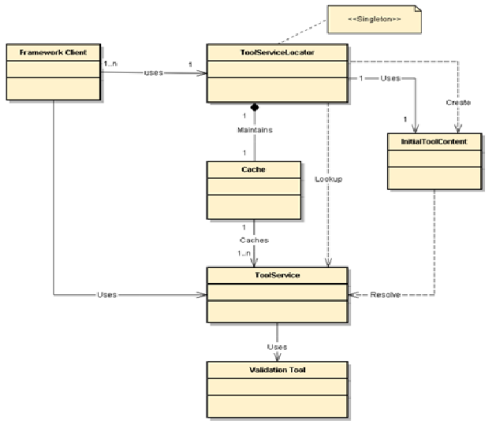


Figure 5 Tool Selection Classes

2.2.4 Translation Layer

The following describes how the Validation Framework will store the representation information from installed tools and provide a consistent data model on which further processing can be done.

2.2.4.1 The Representation Information Model Design

The following section describes how we might design a flexible data model for working with the output from various tools. The examples are given in Java since this is a functional specification requirement for the framework language.

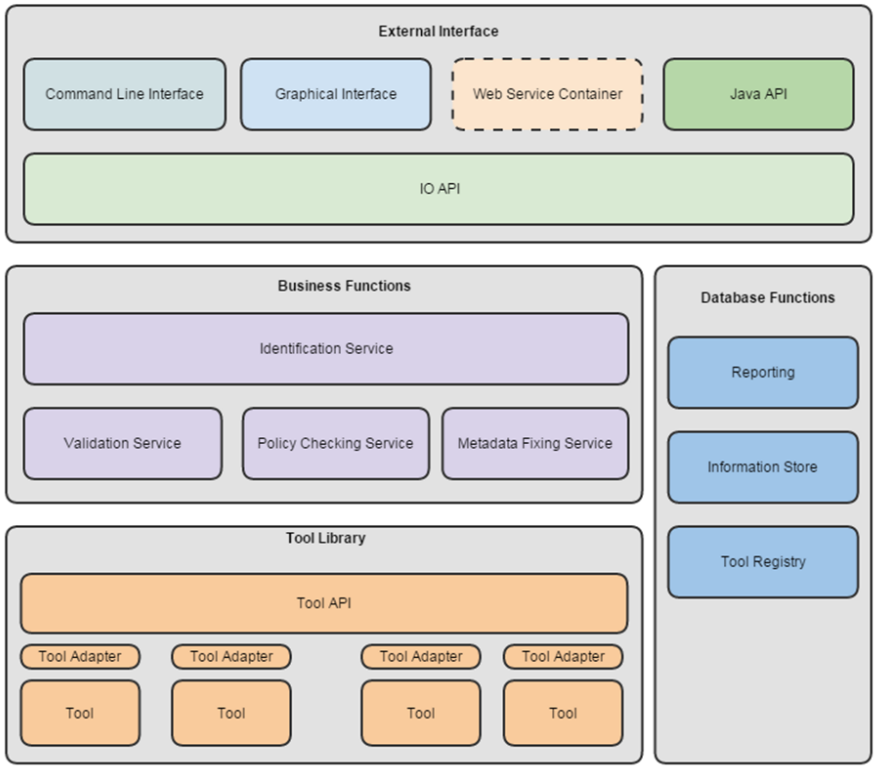
We will need an underlying PREFORMA model, let's call it Node1, which will be the interface to the data and which will hold tool and format independent representation information extracted from the source input streams.
An important requirement is that the model should be able to hold information on multiple files of different formats since the input source may be multiple byte streams.

To meet the requirements for reporting the model must be serializable so that it can be written out as XML or JSON for example.

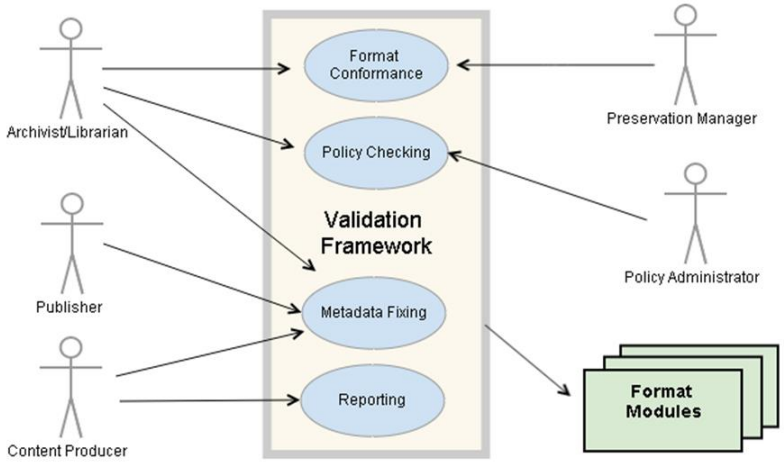
Architectural Design Document (Technical Specification)

Preservica
Digital Preservation

Solution Overview



The 5 minute overview of our proposed solution



Preservica
Digital Preservation

What the solution provides

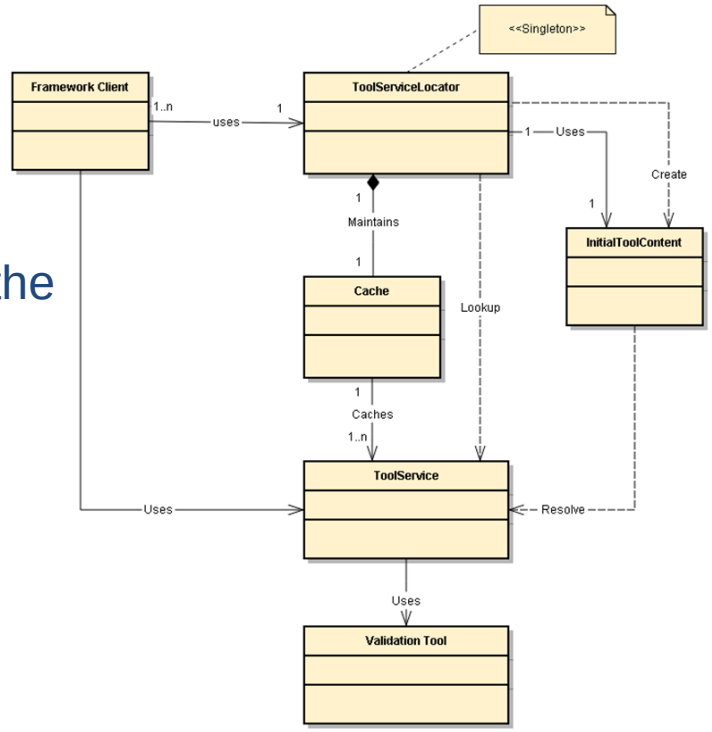
- Common interface for adding tools
- Powerful conformance and policy engine
- Flexible reporting and output
- Multiple APIs for different deployment environments
- Open source development environment on Github

How tools are wrapped

Look up tools to use based on the format (PUID)

The framework uses Java annotations to simplify the API

The tool wrapper just returns format metadata; the framework does the rest

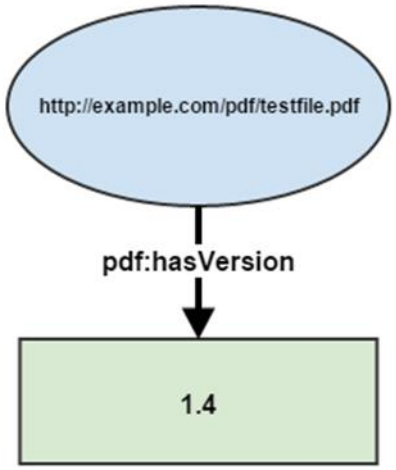



```
public interface AdapterValidate {

    Result validate(java.net.URI uri);

    List<Attribute>  getAttributes();

}
```

```
@CanValidate( format = {"fmt/14", "fmt/276", "fmt/95"} )
public class PdfValidationAdapter implements AdapterValidate {
       Result  validate(java.net.URI uri) {
          . . .
       }
}
```

Preservica
Digital Preservation

How we Assert Conformance



The framework stores the format metadata created by the validators in an RDF model.

We use a powerful policy engine which evaluates statements about the object properties using the policy rules.

The policy rules are simple statements which we wish to check

Example: Check for PDF version 1.4

The following is an example rule for checking if a PDF documentis version 1.4

```
@prefix rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
@prefix pf: http://eu.preforma-project/Text/PDF/
@prefix xs: http://www.w3.org/2001/XMLSchema#

[isPDF1.4: (?s pf:isPDF1.4 "true"^^xs:bool) <-(?s pf:version ?v) equal(?v, "1.4")]
```

This rule says the following:

Add a new statement to the model for each subject s, and Property :isPDF1.4 and value true
If we have an existing statement in the model for subject s, with Property :version and :version equals 1.4

Accessing the suite

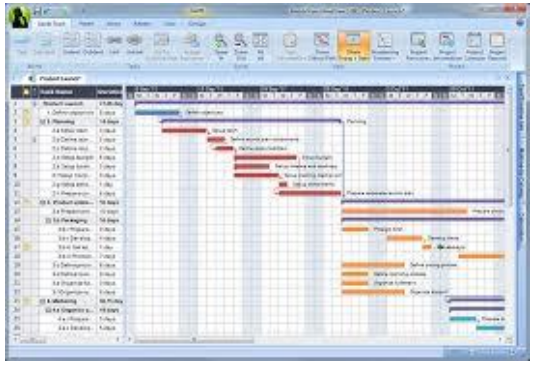
The benefits and advantages of our solution

- Ensure a consistent deployment of each format validator
- Ensure the future-proofing and longevity of such tools
- Encourage the addition of tools outside and after the life of the project
- Sustained investment by Preservica

How we will deliver the solution

Large qualified Project Management team

- Experienced in delivering large and complex projects
- Proven track record in delivering EU funded projects
- All delivery and software development are controlled by our ISO 9001 procedures and processes (since 1993)
- Regular audits by BSI, our Quality Department and our customers
- Flexible methodologies (Agile, Waterfall, etc)
- Configuration Management
- Large automated test suite



Preservica
Digital Preservation

Our Team

- Collaboration with the UK National Archives since 2001
- PRONOM and DROID (the latter is now open source).
- Many research projects such as PLANETS, KEEP, APARSEN, etc.
- Over 50 existing DP customers across the world
- Active in the community, supporting a range of DP initiatives
- Deliver papers and talks on Digital Preservation at conferences across the world

Sustainability of the project deliverables

- Release back any fixes or updates back into the open source community
- Incorporation of the outcomes into our Digital Preservation product
- Get practical feedback from Preservica's large user group
- Add to our existing GITHUB repositories providing communication and collaboration to the wider community

Conclusion

- ✓ Innovative technical solution
- ✓ Experienced project management
- ✓ Excellent long term sustainable model
- ✓ Strong commercial partner with proven track record

Thank You

"*Let me take this to thank you all for the excellent cooperation during the three years in KEEP. I can honestly say that with Preservica on board we were able to make the project a success. The created Emulation Framework is outstanding work and the most significant outcome of the project.*"

Koninklijke Bibliotheek | Nationale Bibliotheek van Nederland

"Investing in a professional, dedicated software team has resulted in significantly improved levels of user experience and service for the ISIS scientific community: Choosing to partner with the Tessella Group is one of the best things we have ever done!" -

Prof Robert McGreevy, STFC