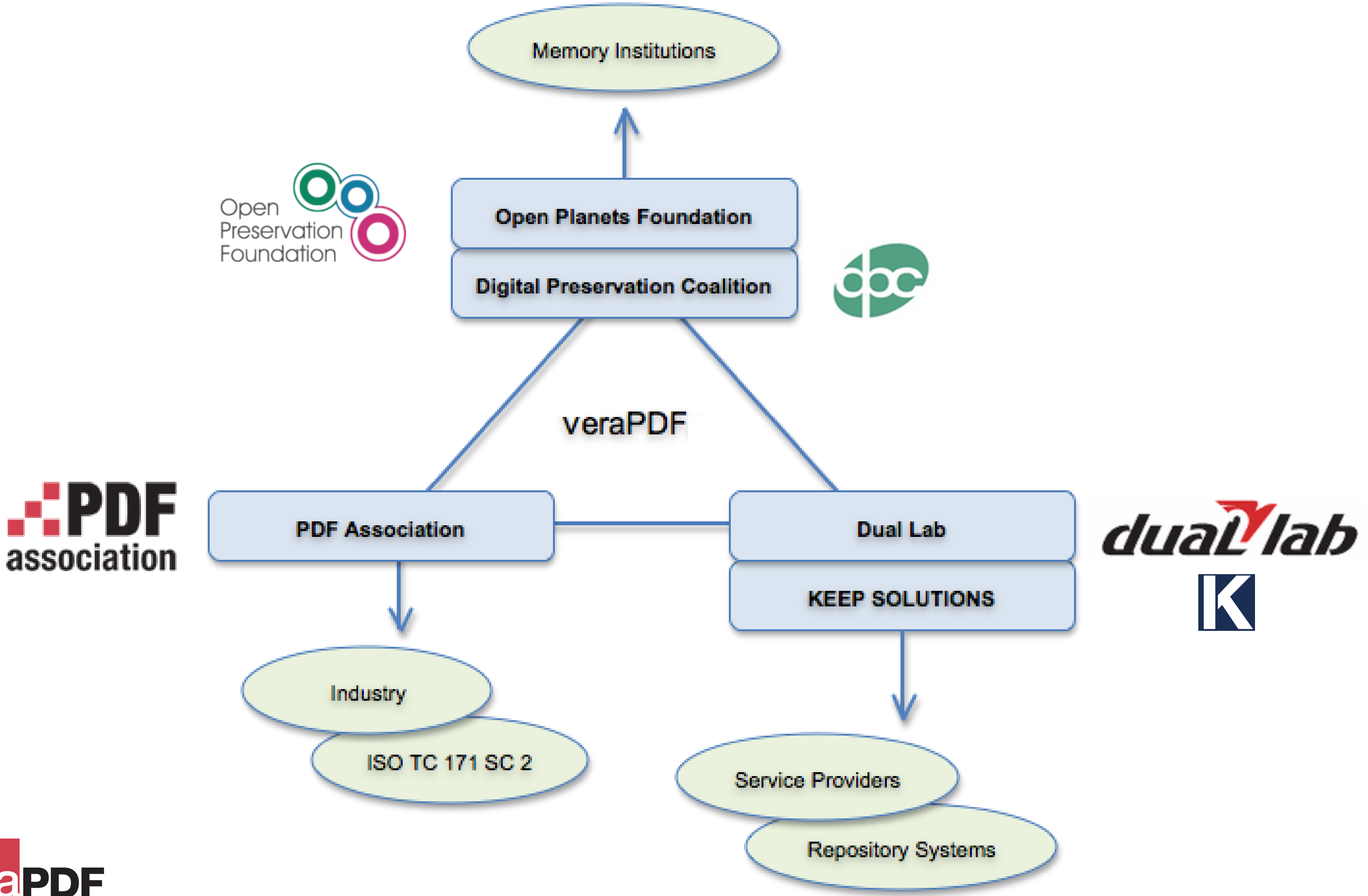
veraPDF

The “definitive” PDF/A validator

Overview

- The veraPDF consortium
  Ed Fay, Open Preservation Foundation
- Community engagement
  Duff Johnson, PDF Association & Ed Fay
- Functional specification
  Duff Johnson & Ed Fay
- Technical specification
  Carl Wilson, Open Preservation Foundation
  Boris Doubrov, Dual Lab

veraPDF consortium



Memory Institutions

Open Planets Foundation

Digital Preservation Coalition

veraPDF

PDF Association

Dual Lab

KEEP SOLUTIONS

Industry

ISO TC 171 SC 2

Service Providers

Repository Systems

veraPDF

Community Engagement

Becoming “definitive”

Community Engagement

- Stakeholders
- Engagement
- Adoption factors
- Activities

Stakeholders

Memory institutions		**Industry**			**3rd party comm-unities**	**Research organi-zations**	**Commercial Customers**
Developers	Users	PDF vendors	Other software vendors	ISO	ICC, fonts, others	Researchers	End users

Areas of Engagement

Awareness	Project visibility		
	Update on progress		
Recruitment	Identify collaborators		
Contribution	Functional requirements	**Evaluation**	Functional review
	Technical requirements		Technical review
	Corpora		Software testing
	Code	**Adoption**	Implementation
	Documentation		Support
	3rd party extensions		Sustainability

Industry

Memory institutions		**Industry**			**3rd party communities**	**Research organizations**	**Commercial Customers**
Developers	Users	PDF vendors	Other software vendors	ISO	ICC, fonts, others	Researchers	End users

PDF Validation TWG

The PDF Association’s PDF Validation Technical Working Group (TWG) builds on 9 years of experience in promoting ISO standards for PDF. The TWG provides:

- an international forum for PDF software developers to discuss ambiguities and establish industry consensus
- a formal “category A” liaison with responsible ISO Working Groups (ISO TC 171 SC 2 WG 5 and WG 8)
- a framework for coordinating activities with the PDF Association’s PDF and PDF/A TWGs, and with relevant 3rd party organisations
- a familiar and respected vehicle for driving information to and promoting adoption by PDF software developers

Adoption Drivers (industry)

- Involvement of industry leadership, including Adobe Systems, callas, iText and the leading members of the ISO's WG for PDF/A
- Industry awareness via communication with PDF Association members and implementers of PDF technology
- Technical clarity via a strict focus on validation
- Implementation diversity via a generic architecture that supports many use cases
- Transparency via open processes to select test files and address contentious questions

Means of Engagement

- veraPDF.org domain
  - The “official” free online validator for use by procurement agencies and end users
  - Static pages providing formal information and detailing industry involvement and support
  - Blogs engaging industry and end users with use cases and explanatory materials
- Mailing lists and social media
- Webinars, publications
- In-person briefings
- Advocacy at software industry events

Digital Preservationists

Memory institutions		**Industry**			**3rd party communities**	**Research organizations**	**Commercial Customers**
Developers	Users	PDF vendors	Other software vendors	ISO	ICC, fonts, others	Researchers	End users

Adoption Drivers (library/archive)

- Requirements workshops
- Policy Profile Registry
- Digital preservation tool integration
- Software evaluations
- Sustainability through the Open Preservation Foundation

Means of Engagement

- veraPDF.org domain
- Mailing lists and social media
- Webinars, publications
- In-person briefings
- Advocacy at memory institution events
- ‘Hack-a-thons’
- ‘Edit-a-thons’ (documentation sprints)
- Exemplar Policy Profiles

veraPDF

Functional Specification

Realising “definitive”

Functional Specification

- PDF/A validation in context
- Conformance Checker
  - Components
  - Extensions
  - Interfaces
  - Integrations

PDF/A Validation in Context

- ‘Shall’, ‘should’, and ‘may’
    - ‘Shall’ → normative requirements
    - ‘Should’ and ‘may’ → policy conformance
- Dependency on PDF 1.4 / ISO 32000
- 3rd party data structures
    - 80+ external normative references in PDF
    - images, fonts, colour profiles, attachments...
    - validated by veraPDF when explicitly required (“shall”) by the PDF/A specification
    - otherwise handled through extensions

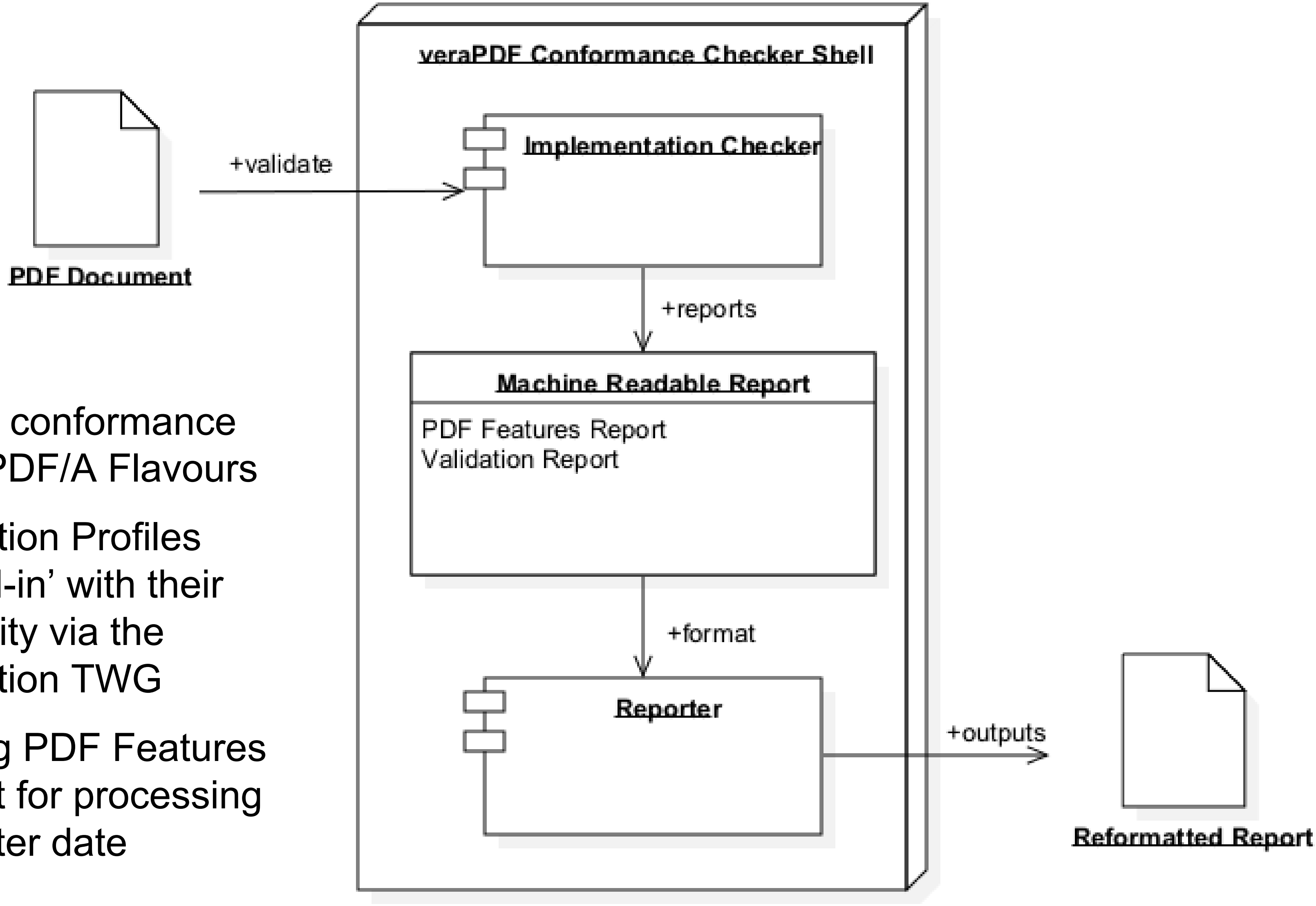
Beyond PDF/A: PDF Validation

- The vast majority (99+%) of PDF documents received by libraries and archives are “plain” PDF, not PDF/A
- In addition to meeting real-world archival needs, industry interest and involvement increases dramatically in the context of validating ISO 32000
- PREFORMA may consider extending the project to address all of ISO 32000 and required 3rd party data structures

The Conformance Checker

- Implementation Checker
- Metadata Fixer
- Policy Checker
- Reporter
- Shell(s)

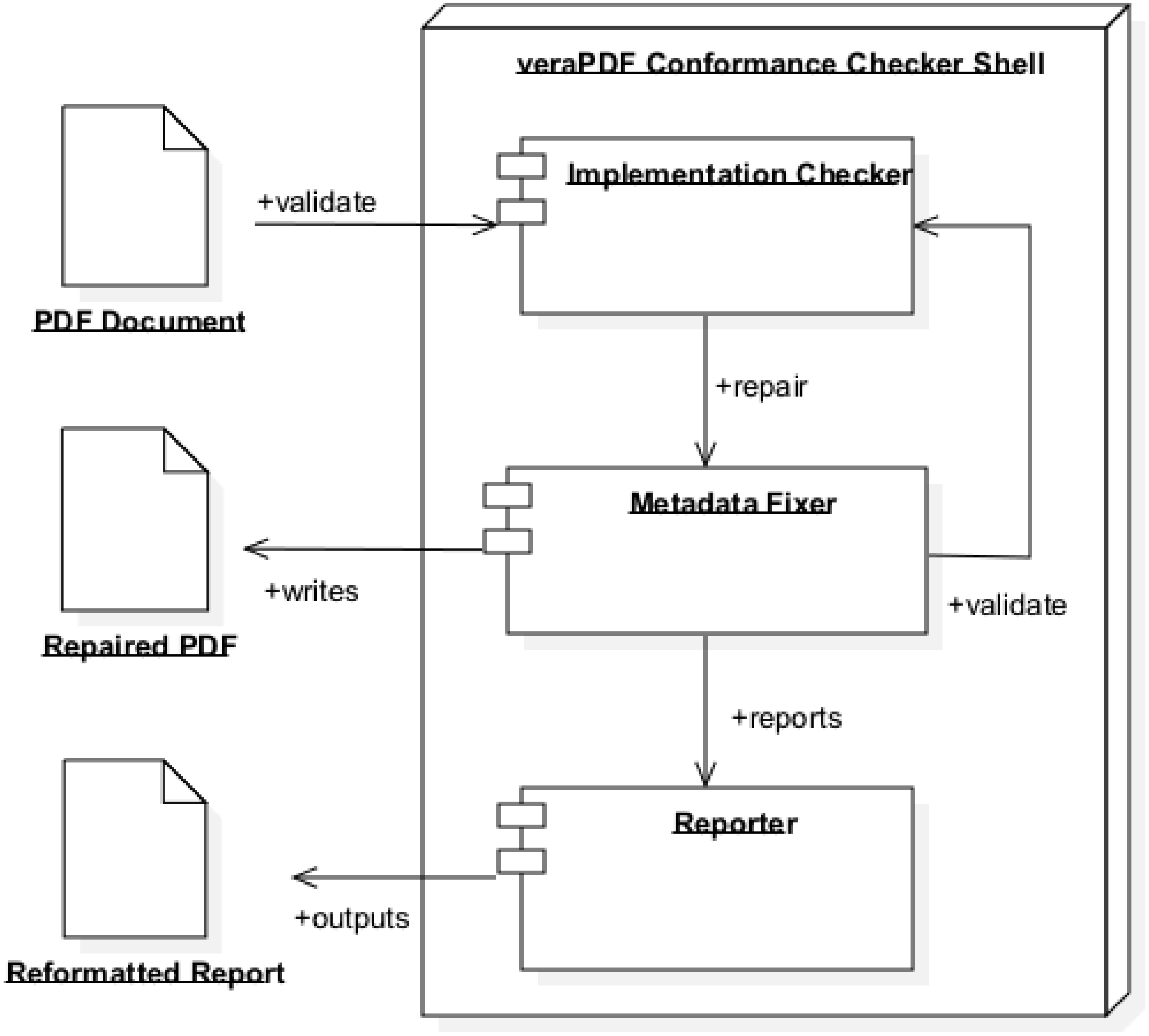
Implementation Checker



- Check conformance to all PDF/A Flavours
- Validation Profiles ‘baked-in’ with their authority via the Validation TWG
- Storing PDF Features Report for processing at a later date

Metadata Fixer

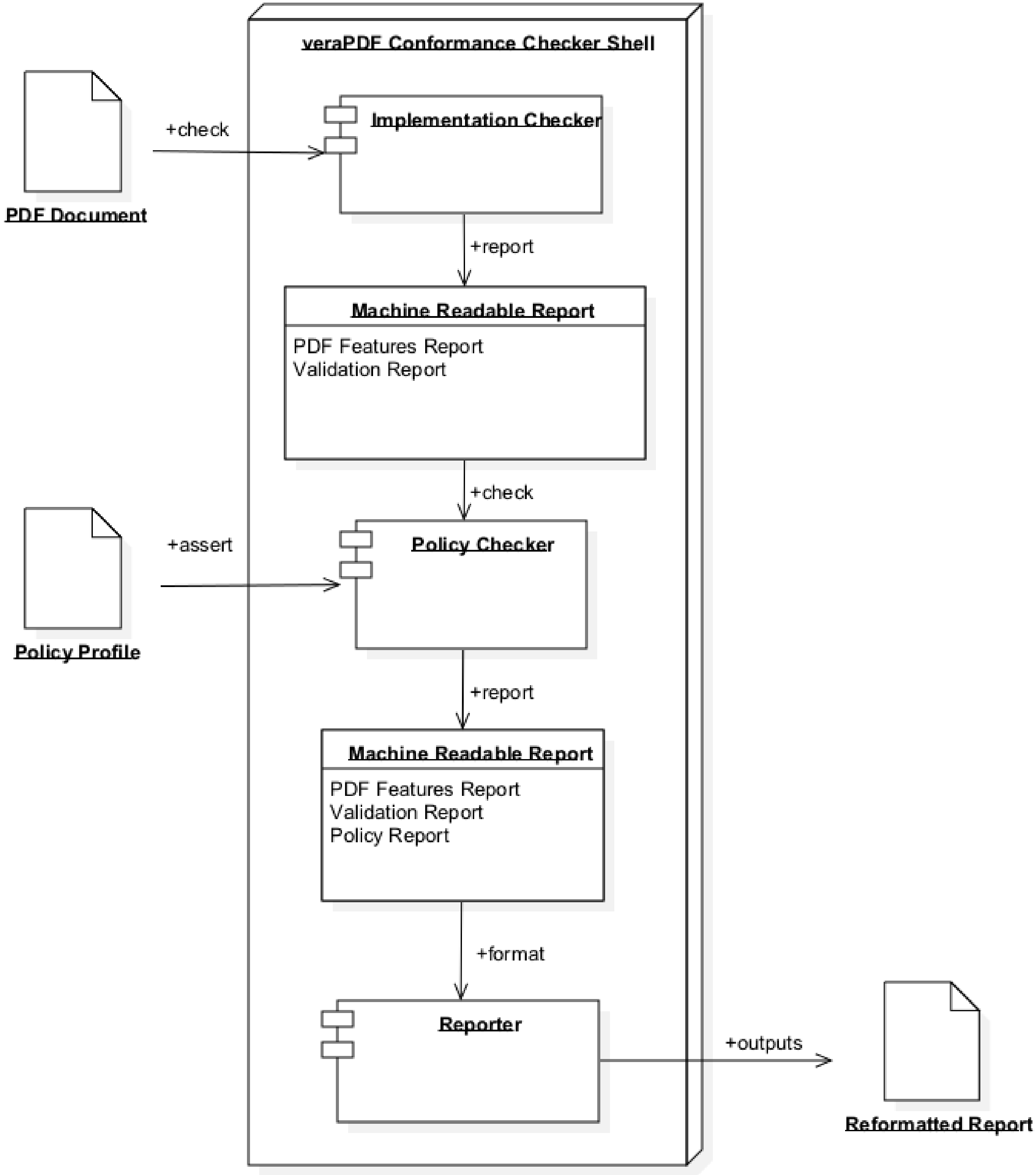
- Removes (from invalid file) or adds (to valid file) the PDF/A flag in PDF/A Documents
- Synchronizes Info dictionary with XMP Metadata
- Embeds a predefined XMP package if it is missing
- Allows third-party tools to modify XMP and validates it afterwards



veraPDF

Policy Checker

- Policy Checking is independent of PDF/A Validation
- ‘Should’ and ‘may’ statements can be enforced (normative specifications which are not requirements)
- Policy Profiles can be shared between institutions via the Policy Profile Registry



veraPDF Conformance Checker Shell

PDF Document

+check

Implementation Checker

+report

Machine Readable Report

PDF Features Report
Validation Report

+check

Policy Profile

+assert

Policy Checker

+report

Machine Readable Report

PDF Features Report
Validation Report
Policy Report

+format

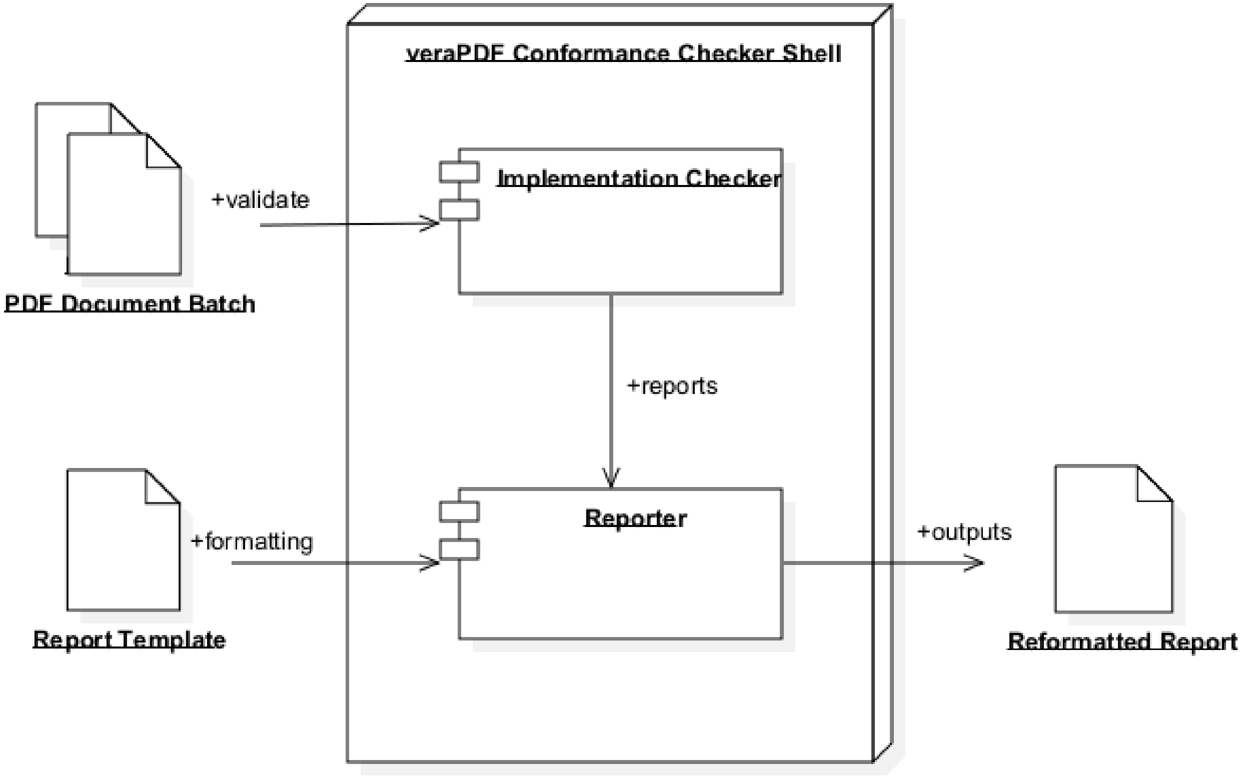
Reporter

+outputs

Reformatted Report

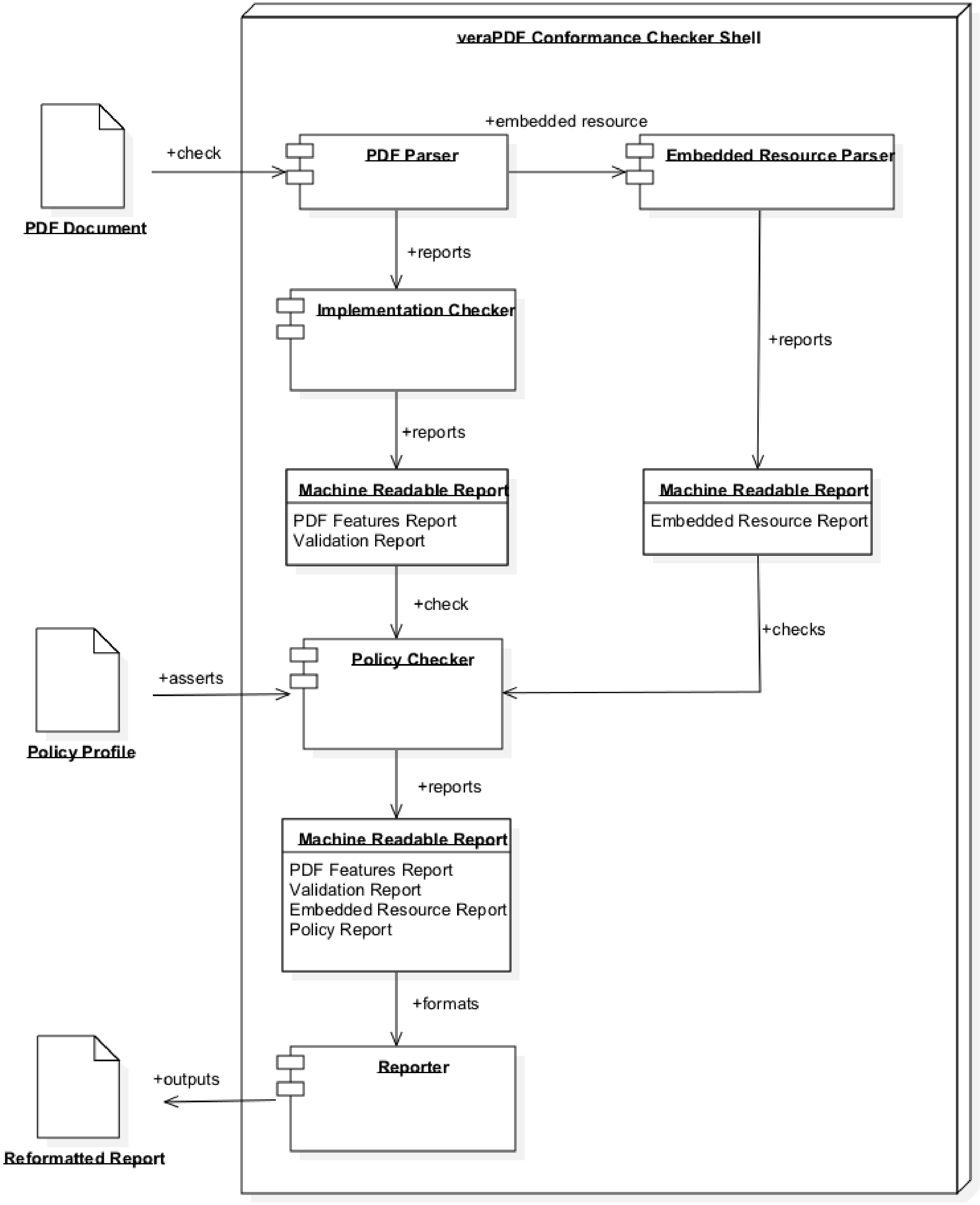
Reporter

- Transforms reports from all other components
- Report Templates control output (Machine-readable, Human-readable)
- HTML and PDF will be supplied, users can produce others
- Can also transform for compatibility with external systems (DIRECT, PREMIS, METS/MODS, etc.)



Extensions

- PDF Parser is independent of Validation and Policy Checking, however they depend on its outputs
- Embedded Resource Parsers handle third-party standards
- Policy Checker can use any extended information



PDF Parser

- Greenfield
  - Fully GPLv3+/MPv2+ (no dependencies)
  - But, limits information in PDF Features Report
- PDFBox (then greenfield)
  - Development and testing of Implementation and Policy Checkers begins immediately
  - Enables cross-testing between PDFBox and greenfield PDF Parser
  - Involves existing PDFBox community

Embedded Resources

- Implementation Checker will carry out the set of checks required by PDF/A
- Based on collaboration with relevant communities, we will provide options for developing extensions
  - Font validator
  - ICC profile validation
- This will improve reliability beyond the explicit requirements of PDF/A

Dependencies

- Implementation Checker, Fixer
  - No dependencies (greenfield Parser, Writer)
  - *Released under GPLv3+/MPv2+*
- Policy Checker, Reporter, Shell
  - Schematron
  - Format libraries and internationalization
  - Web services and layout frameworks
  - *Compatible with GPLv3+/MPv2+*
- High-level dependencies
  - Runtime, testing, standard libraries
  - *Compatible with GPLv3+/MPv2+*

Interfaces (Shells)

- Command Line Interface
- Desktop GUI
- Web GUI

- Batches
- Scheduling
- Integrations

Integrations

- Workflow systems
- Repository systems
- Digital preservation tools

- Existing committers doing the work

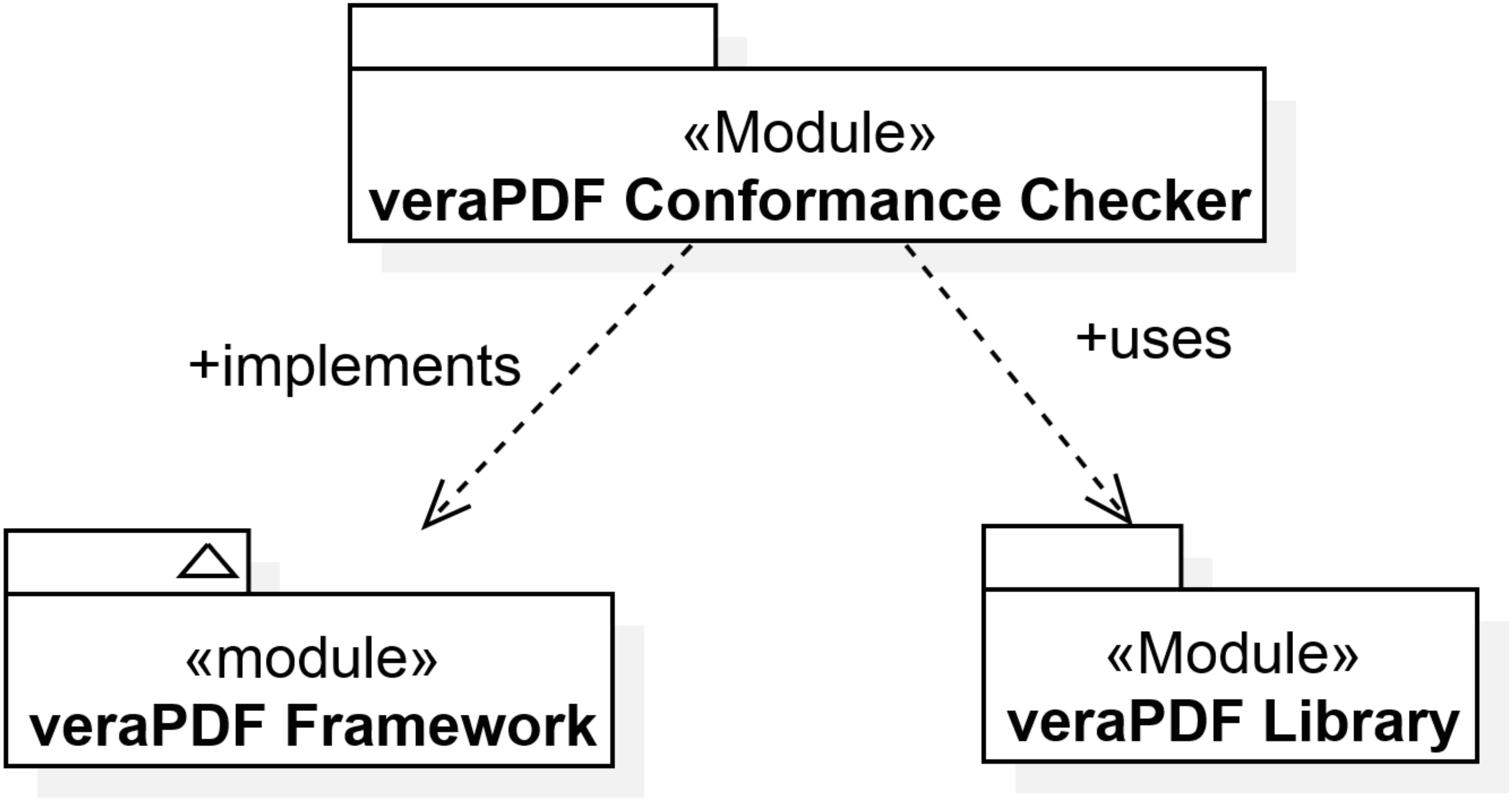
archivematica. DSPACE JHOVE

veraPDF

Technical Specification

Implementing “definitive”

Architectural Overview



«Module»
veraPDF Conformance Checker

+implements

+uses

«module»
veraPDF Framework

«Module»
veraPDF Library

Modularity

- **veraPDF Library**
  Java library that provides definitive Implementation Checking (PDF/A Validation and PDF Features Reporting) and Metadata Fixing for PDF Documents
- **veraPDF Framework**
  A light Java framework to support developers implementing a Conformance Checker
- **veraPDF Conformance Checker**
  Combines the library and framework and delivers a PDF/A Conformance Checker

Software Testability

- **Isolateability**

  The degree to which a component can be tested in isolation

- **Separation of concerns**

  The degree to which the component under test has a single, well defined responsibility

- **Understandability**

  The degree to which the component under test is documented or self-explaining

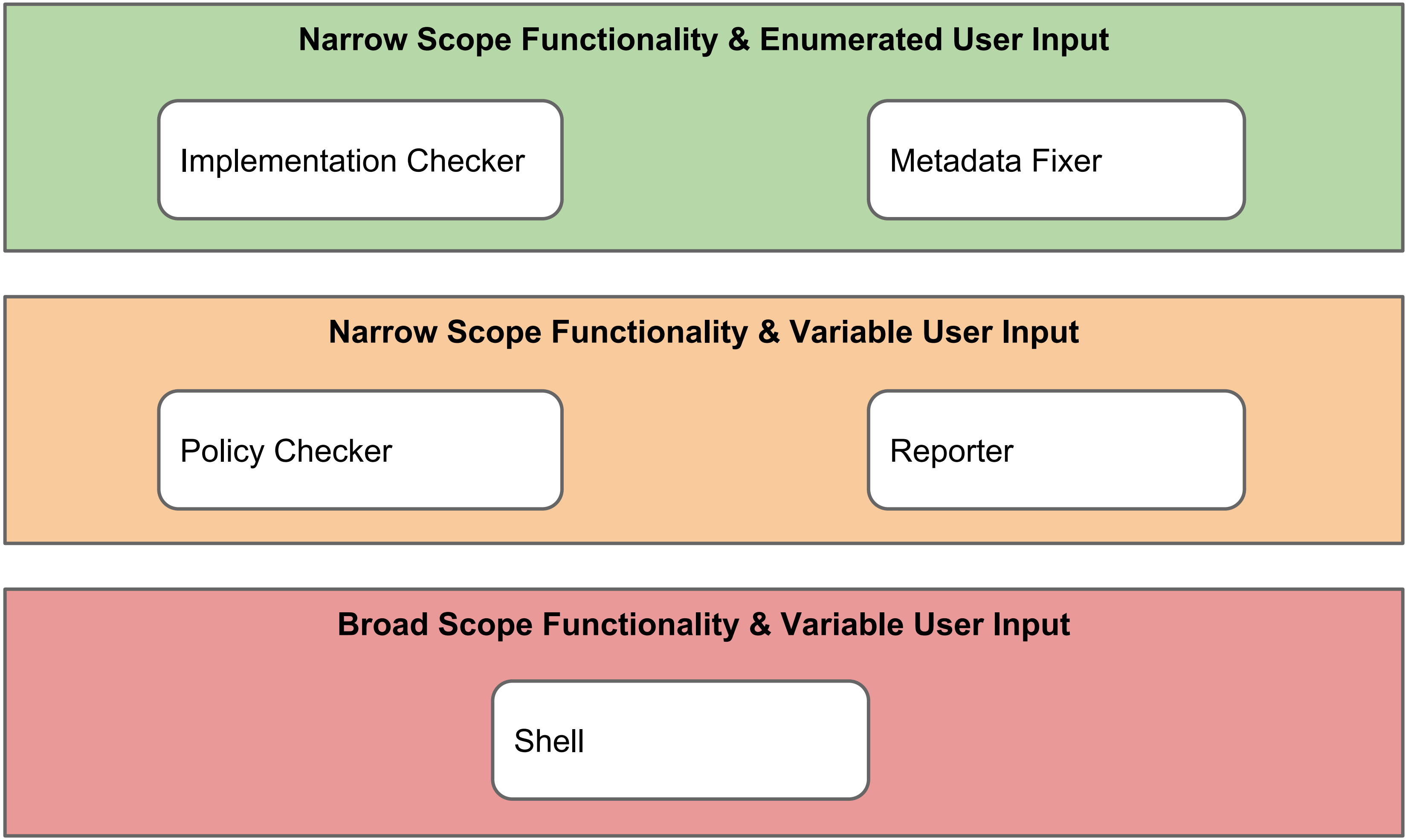
Testing and Traceability

- Providing a traceable path from requirements to test cases
- Requirements unambiguously represented as files in test corpora
- Visibly mapping the relationship between requirements and test cases
- Up to date reporting of test results and progress publically accessible

Engineered for Reliability

- Test driven development
- Immutable classes for built in failure atomicity and thread safety supporting scalability
- State and complexity kept outside of the Conformance Checker components, excepting the Shell
- Implementation Checker & Metadata Fixer offer enumerated, well tested execution paths

Engineered for Reliability



Narrow Scope Functionality & Enumerated User Input

Implementation Checker

Metadata Fixer

Narrow Scope Functionality & Variable User Input

Policy Checker

Reporter

Broad Scope Functionality & Variable User Input

Shell

veraPDF Shell

Manages state and complexity for the other Conformance Checker components:

- obtaining and parsing user input
- configuration of components
- storage and retrieval of user-defined Policy Profiles and Report Templates
- processing workflow
- automation and scheduling

veraPDF Framework

- Generic code for Shell functionality
  - managing system and user config
  - storage and retrieval of user-defined Policy Profiles and Report Templates
  - SHA-1 hash generation and validation
- “Vanilla” standards-based component implementations
  - XSLT-based Reporter
  - Schematron-based Policy Checker

veraPDF Framework

Open standards-based, provided as native Java functionality

- XML standards
  - XSD / XSLT
  - TMX
  - Schematron
- Web standards
  - URIs
  - Internet Media Types
  - JAX-WS REST services

Sustainability

- A community that brings together:
  - PDF Industry
  - Memory Institutions
- Software engineering standards
  - high unit test coverage (85% or greater)
  - code review for external contributions
- Automated unit and integration testing
- Nightly publication of:
  - Test results
  - Progress against requirements / corpora
  - Javadoc, style checks, and static analysis

PDF/A Validation Challenges

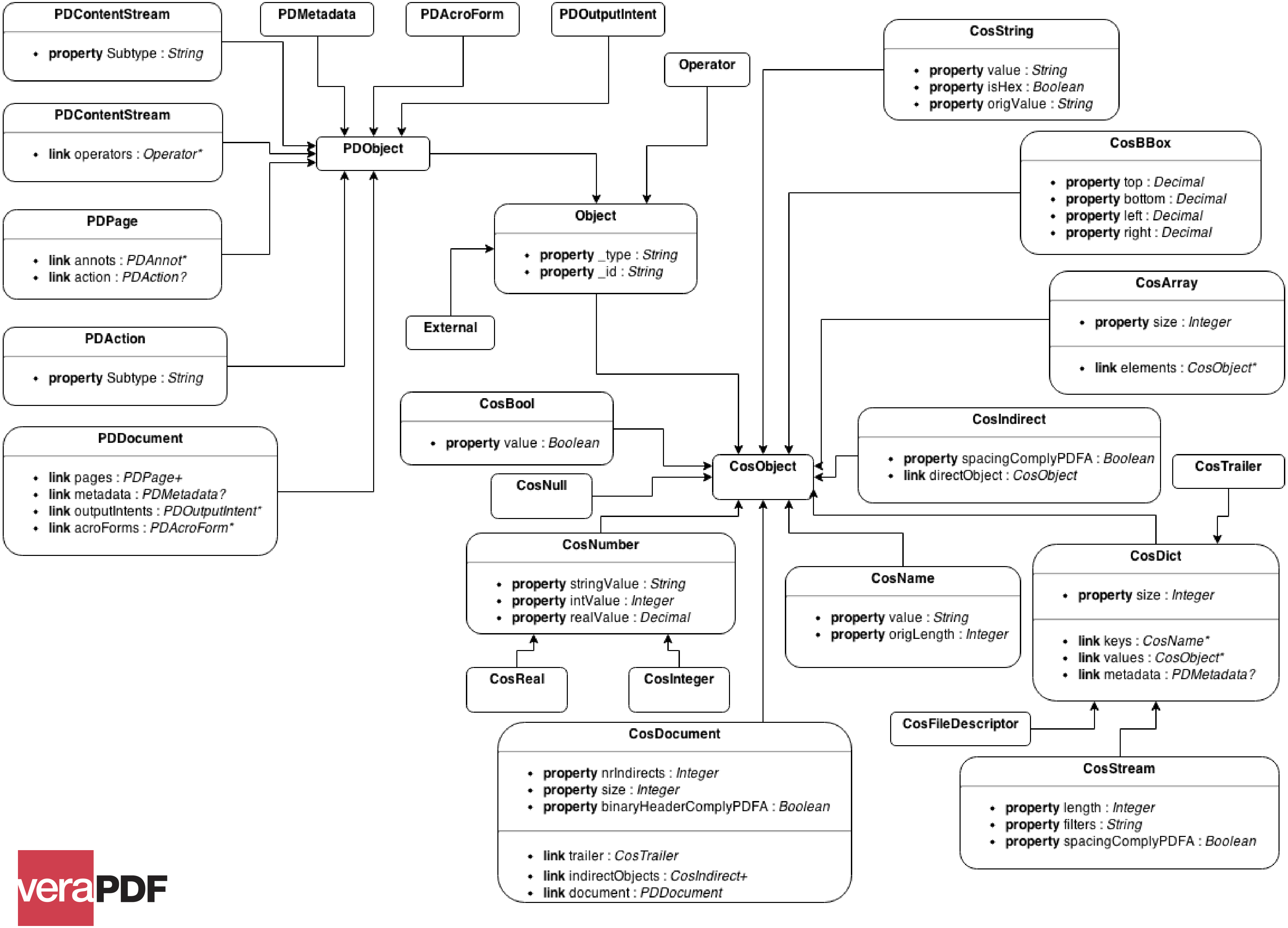
- The challenges
  - Hundreds of normative requirements
  - Specifications are often ambiguous
- The veraPDF solution
  - A platform-independent human-friendly language for formal description of all requirements
  - A common language for different communities
  - A model that may be extended to address the wide variety of applicable technologies

Test Corpora

- Identification of all normative requirements in all versions of PDF/A and relevant parts of ISO 32000-1 (or PDF 1.4 for PDF/A-1)
- Identification of possible test scenarios and their formal description (800+ test cases)
- Analysis of the existing PDF/A Corpora in order to incorporate them into the definitive corpora
- 200+ messages on “verapdf-tech” mailing list discussing the ambiguities

Abstract Validation Model

- Object Model
  - hierarchy of Object types
  - Objects have properties (inheritable) and associations with other types (links)
  - formal syntax for Object Model with automatic interface generation for the PDF Parser
- Validation Profile
  - a list of rules defined per each Object type
  - a rule is a boolean expression containing Object properties


PDContentStream
property Subtype : String
PDMetadata
PDAcroForm
PDOutputIntent
CosString
property value : String
property isHex : Boolean
property origValue : String
Operator
PDContentStream
link operators : Operator*
PDObject
CosBBox
property top : Decimal
property bottom : Decimal
property left : Decimal
property right : Decimal
PDPage
link annots : PDAnnot*
link action : PDAction?
Object
property _type : String
property _id : String
CosArray
property size : Integer
link elements : CosObject*
External
PDAction
property Subtype : String
CosBool
property value : Boolean
CosIndirect
property spacingComplyPDFA : Boolean
link directObject : CosObject
CosObject
CosTrailer
CosNull
PDDocument
link pages : PDPage+
link metadata : PDMetadata?
link outputIntents : PDOutputIntent*
link acroForms : PDAcroForm*
CosNumber
property stringValue : String
property intValue : Integer
property realValue : Decimal
CosDict
property size : Integer
link keys : CosName*
link values : CosObject*
link metadata : PDMetadata?
CosName
property value : String
property origLength : Integer
CosReal
CosInteger
CosFileDescriptor
CosDocument
property nrIndirects : Integer
property size : Integer
property binaryHeaderComplyPDFA : Boolean
link trailer : CosTrailer
link indirectObjects : CosIndirect+
link document : PDDocument
CosStream
property length : Integer
property filters : String
property spacingComplyPDFA : Boolean

veraPDF

Formal syntax for the Model

```
% low-level PDF Document object
type CosDocument extends CosObject
{
    % Byte size of the document
    property size: Integer;
    % link to the document trailer
    link trailer: CosTrailer;
    % link to all Indirect objects
    link indirectObjects: CosIndirect+;
}
```

Syntax for the Validation Profile

- XML-based:
  - metadata identifying the PDF/A Flavour
  - collection of rules
  - each rule has one or more normative references to the specifications
  - message template for errors
  - Metadata fixes
- Profiles are signed!

Benefits of Validation Model

- Technology agnostic
- Formalizes the language of normative references
- Extensible beyond PDF/A to include ISO 32000, images, fonts, ICC profiles, digital certificates
- Validation algorithm is predefined, so that different implementations shall generate identical reports

Policy Checks via PDF Features

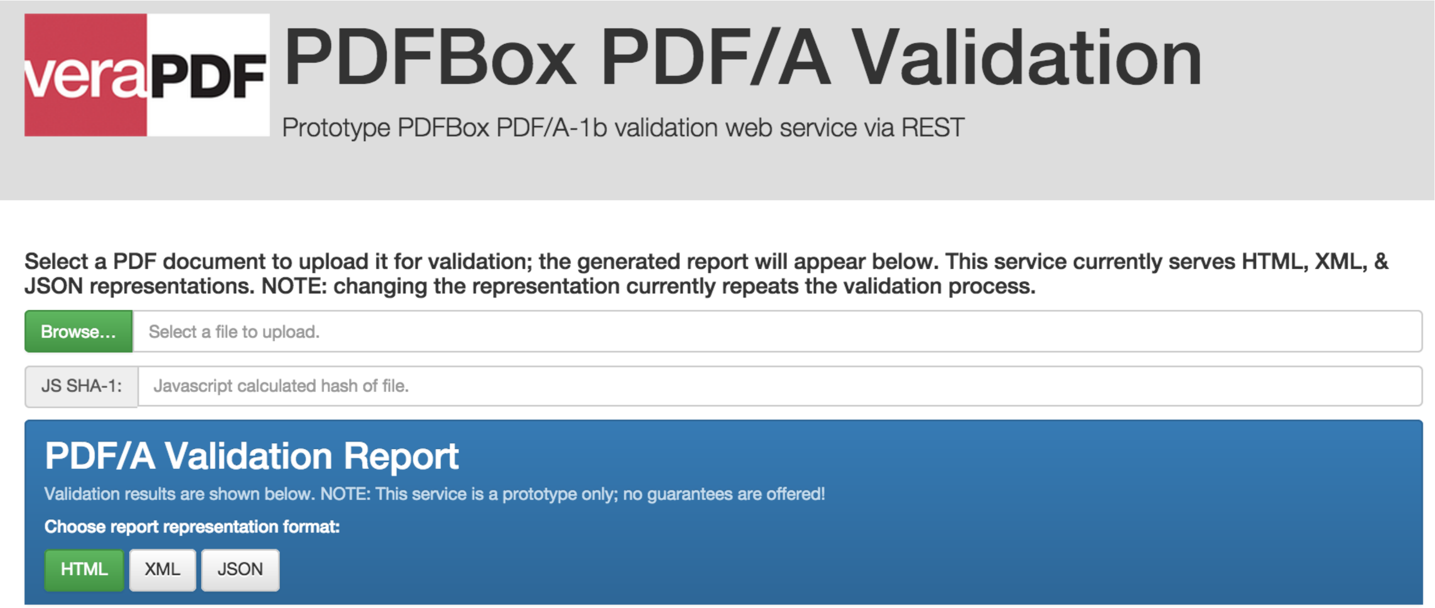
- Extract information from PDF into PDF Features Report (XML-based)
- Policy Profile uses XSLT-like syntax to verify content of PDF Features Report
- Schematron → proven technology for Policy Checks implementation
- No regeneration of PDF Features Report in case of Policy changes. Only Policy Profile needs to be updated

Human-readable Reports

- Generated from Machine-readable Report via XSLT technology
- Direct HTML5 Report generation
- PDF Report generation via XSL-FO
- Localization via Language Packs (TMX)
- Accessible (WCAG 2.0 Level AA)
- Easily adjustable design

Demonstration

- http://demo.verapdf.org



PDFBox PDF/A Validation

Prototype PDFBox PDF/A-1b validation web service via REST

Select a PDF document to upload it for validation; the generated report will appear below. This service currently serves HTML, XML, & JSON representations. NOTE: changing the representation currently repeats the validation process.

PDF/A Validation Report

Validation results are shown below. NOTE: This service is a prototype only; no guarantees are offered!

Choose report representation format:

Conclusion

How veraPDF is different

The definitive PDF/A validator

- A purpose-built PDF/A validator
- Formal liaison with ISO committees
- Industry and memory institution buy-in
- A generic, fully extensible framework
- Reuse of proven technologies
- Integrated with existing validation tools
- Open source best practices, including leveraging of existing communities