

Federated Data Interpretation Systems Reference Architecture (FDIS-RA)

[Go to General Topics](#)

Purpose of This Response

The Federated Data Interpretation Systems Reference Architecture (FDIS-RA) is a proposed response to the [Structured Information Processing Reference Architecture \(SIP-RA\) Request for Proposal](#).

FDIS-RA defines a platform-independent Reference Architecture for the governed interpretation, validation, comparison, and analysis of structured information artifacts in regulated and mission-critical environments. The architecture applies the separation of data, structure, semantics, and interpretation required by SIP-RA to federated systems that process reported information across organisational, jurisdictional, and technological boundaries.

The proposed architecture defines governed processing flows, explicit architectural boundaries, version-controlled interfaces, semantic and interpretive lifecycle controls, and traceability mechanisms that preserve evidentiary integrity and support reproducible interpretation. FDIS-RA provides the architectural response through which the SIP-RA requirements are realised for federated data interpretation systems without prescribing specific implementation technologies, products, transport mechanisms, or deployment topologies.

Table of Contents

* [1. Introduction](#)

- [1.1 Architectural Context and Motivation](#)
- [1.2 Purpose and Intent](#)
- [1.3 Reference Architecture Definition](#)
- [1.4 Scope](#)
 - [1.4.1 In Scope](#)
 - [1.4.1.1 Financial Data Interpretation Systems](#)
 - [1.4.1.2 Semantic Interpretation and Validation](#)
 - [1.4.1.3 Architecture and Interfaces](#)
 - [1.4.1.4 Standards Applicability and Coordination](#)
 - 2. [1.4.2 Out of Scope](#)
 - [1.4.2.1 Specific Implementations](#)
 - [1.4.2.2 Product Architectures](#)
 - [1.4.2.3 Enforcement Policy](#)
 - [1.4.2.4 Infrastructure and Platform Standards](#)
- 5. [1.5 Intended Audience](#)
 - [1.5.1 Standards Bodies](#)
 - [1.5.2 Regulators and Supervisory Agencies](#)
 - [1.5.3 System and Enterprise Architects](#)

- 1.5.4 Technology Providers and Integrators
- 1.5.5 Researchers and Practitioners in Semantic Governance and Regulatory Technology

* 2. Background and Motivation

- 2.1 International Context and Applicability
- 2.2 The FDTA Context
 - 2.2.1 FDTA Implementation Timeline
 - 2.2.2 Architectural Implications of the FDTA Timeline
- 3. 2.3 The Role of SBRM
 - 2.3.1 Reporting Semantics as a Governed Architectural Concern
 - 2.3.2 Separating Claims, Observations, and Interpretations
 - 2.3.3 Why SBRM Alone Is Not Enough
- 4. 2.4 Existing Functional Capabilities in Reporting Systems
 - 2.4.1 Source Artifact Persistence
 - 2.4.2 Information Extraction and Observation
 - 2.4.3 Observations with Provenance
 - 2.4.4 Semantic Representations
 - 2.4.5 Rules and Validation
 - 2.4.6 The Architectural Gap
- 5. 2.5 Why a Reference Architecture Is Needed
 - 2.5.1 Semantic Drift over Time
 - 2.5.2 Tool-Driven Interpretations
 - 2.5.3 AI Opacity
 - 2.5.4 Lack of Conformance Criteria
 - 2.5.5 Difficulty Proving Comparability Across Systems and Agencies

* 3. FDIS Reference Architecture

- 3.1 Definition
- 3.2 Design Principles of the FDIS Reference Architecture
 - 3.2.1 Governance-Centric Orientation
 - 3.2.2 Explicit Separation of Interpretive Concerns
 - 3.2.3 Pipeline-Oriented, Domain-Centric Composition
 - 3.2.4 Quality Attributes as Architectural Drivers
 - 3.2.5 Independence from Specific Technologies and Tools
 - 3.2.6 Architectural Style as a Foundation for Long-Term Regulatory Reliance
- 3. 3.3 Relationship to OMG Architectures and Standards
 - 3.3.1 Compatibility with Other OMG Reference Architectures
 - 3.3.2 Alignment with OMG Model-Driven Architecture (MDA)
 - 3.3.3 Non-Overlap with Product, Platform, or Implementation Architectures
- 4. 3.4 Normative Artefact Types
 - 3.4.1 Guidance Statements (G)
 - 3.4.2 Interface Definitions (I)
 - 3.4.3 Distinguishing Obligation from Structure
 - 3.4.4 Illustrative Examples
 - 3.4.5 Section-Scoped Normative Artefacts

* 4. Architectural Overview

- 4.1 Architectural Flow and Boundary Model
- 4.2 Interpretation Architecture Model
 - 4.2.1 Pipeline-Based Interpretation Architecture
 - 4.2.2 Signals, Indicators, Parameters, and Arguments
- 3. 4.3 Architectural Style
 - 4.3.1 Governance-Centric Orientation
 - 4.3.2 Explicit Separation of Interpretive Concerns
 - 4.3.3 Pipeline-Oriented, Domain-Centric Composition
 - 4.3.4 Quality Attributes as Architectural Drivers
 - 4.3.5 Independence from Specific Technologies and Tools
 - 4.3.6 Architectural Style as a Foundation for Long-Term Regulatory Reliance
- 4. 4.4 Architectural Boundaries and Responsibilities
- 5. 4.5 Non-Functional Characteristics and Quality Attributes
 - 4.5.1 Maintainability
 - 4.5.1.1 Modularity
 - 4.5.1.2 Reusability
 - 4.5.1.3 Analysability
 - 4.5.1.4 Modifiability
 - 4.5.1.5 Testability
 - 2. 4.5.2 Manageability
 - 3. 4.5.3 Traceability
 - 4. 4.5.4 Contractability
 - 5. 4.5.5 Auditability
 - 6. 4.5.6 Interoperability
 - 7. 4.5.7 Securability
 - 8. 4.5.8 Reliability
 - 9. 4.5.9 Portability
 - 10. 4.5.10 Scalability
 - 11. 4.5.11 Performance

* 5. Core Architectural Components

- 5.1 Data Plane Components
 - 5.1.1 Evidence Intake and Document Custody
 - 5.1.2 Extraction and Observation Generation
 - 5.1.3 Observation and Metadata Management
 - 5.1.4 Semantic Canonicalisation (SBRM-Aligned)
 - 5.1.5 Interpretation, Rules, and Calculation
 - 5.1.6 Validation, Conformance, and Assessment
 - 5.1.7 Analysis, Reporting, and User Interaction
- 2. 5.2 Boundary Realisation Components
 - 5.2.1 Ports
 - 5.2.2 Port Interfaces
 - 5.2.3 Adapters
 - 5.2.4 Adapter Realisation Pattern
- 3. 5.3 Control Plane Components
 - 5.3.1 Semantic Governance and Version Context Management
 - 5.3.2 Parameter and Configuration Control
 - 5.3.3 Contract Governance and Lifecycle Control

- 5.3.4 Monitoring, Signals, and Indicators

* 6. Interfaces and Contracts

- 6.1 Interfaces in a Reference Architecture
- 6.2 Core Interface Categories
- 6.3 Form-Scoped Semantic Interfaces (FSIs)
- 6.4 Adapter Modularity, Compatibility, and Reuse
- 6.5 Forking and Parallel Interpretation
- 6.6 Semantic Version and Context Management

* 7. Standards Alignment

- 7.1 Purpose and Scope of Standards Alignment
- 7.2 Relationship to FDTA and Federal Standards Policy
- 7.3 Classification of Standards Used in FDIS
 - 7.3.1 Technical (Governance-Authority) Standards
 - 7.3.2 Defacto (Market-Adopted) Specifications
- 4. 7.4 Use of Defacto Specifications within FDIS-Aligned Architectures
- 5. 7.5 When No Suitable Standard Exists
- 6. 7.6 Mapping Standards to Core Architectural Components
- 7. 7.7 Structure of the Standards Appendix
- 8. 7.8 Summary and Architectural Position

* 8. Who Should Use the FDIS Reference Architecture and When

- 8.1 Regulators and Supervisory Agencies
- 8.2 Standards, Architects, and Governance Bodies
- 8.3 System and Enterprise Architects
- 8.4 Technology Providers and System Integrators
- 8.5 Data Producers and Reporting Entities
- 8.6 Researchers, Policy Analysts, and Oversight Bodies
- 8.7 Summary of Intended Use

* 9. Benefits and Expected Outcomes

- 9.1 Improved Comparability
- 9.2 Reduced Semantic Ambiguity
- 9.3 Improved Auditability and Defensibility
- 9.4 AI Use with Bounded Authority
- 9.5 Long-Term Semantic Stability
- 10. Summary and Next Steps
- 11. References

Table of Contents

- [1. Introduction](#)
- [2. Background and Motivation](#)
- [3. FDIS Reference Architecture](#)

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

From:
<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:
<https://wiki.didosolutions.com/dido/01-fdis-ra/start?rev=1783781585>

Last update: **2026/07/11 07:53**

