

JADN Version 2.0

Information Modeling and Conceptual Design

What is JSON Abstract Data Notation (JADN)?

- An Information Modeling language
 - Defines Information = essential content
 - Enables conceptual design: "What does this message/document need to communicate?" separately from "What does this message/document look like?"
 - Specifies equivalence across multiple schema formats and data formats
- A UML profile for messaging
 - Based on UML DataType classifiers: primitive, structured, multiplicity, association
- Composed of UML/XSD/RDF DataTypes that define:
 - Value Space (information content of a data item)
 - Lexical Space (literal sequence of bytes or characters in a data item)
 - Lexical to Value Mapping (encoding rules for a specific data format)

Where to use JADN?

Anywhere standardized messages are sent between systems.

All of these and more:

- ✓ Support for PACE
- ✓ Support for Indicators of Behavior Sharing (IoB)
- ✓ Support for OpenC2
- ✓ Support for STIX and/or TAXII
- ✓ Support for CSAF and/or VEX
- ✓ Support for NIEMOpen
- ✓ Support for Value Stream Management Interoperability (VSMI)
- ✓ Support for CACAO Playbooks
- ✓ Support for Threat Actor Context (TAC)
- ✓ Support for OASIS Heimdall Data Format (OHDF)
- ✓ Support for SPYDERISK
- ✓ Support for STIX Shifter
- ✓ Support for SARIF
- ✓ Support for OXA
- ✓ Support for SBOM

Electronic Court Filing

<https://docs.oasis-open.org/legalxml-courtfilling/ecf/v5.0/ecf-v5.0.pdf>

4 Information Model

The information model describes the data content exchanged between MDEs in each operation as a set of XML messages, case type **[NIEM]** augmentations, XML schema and **[Genericcode]** code lists and binary attachments.

4.1 Messages

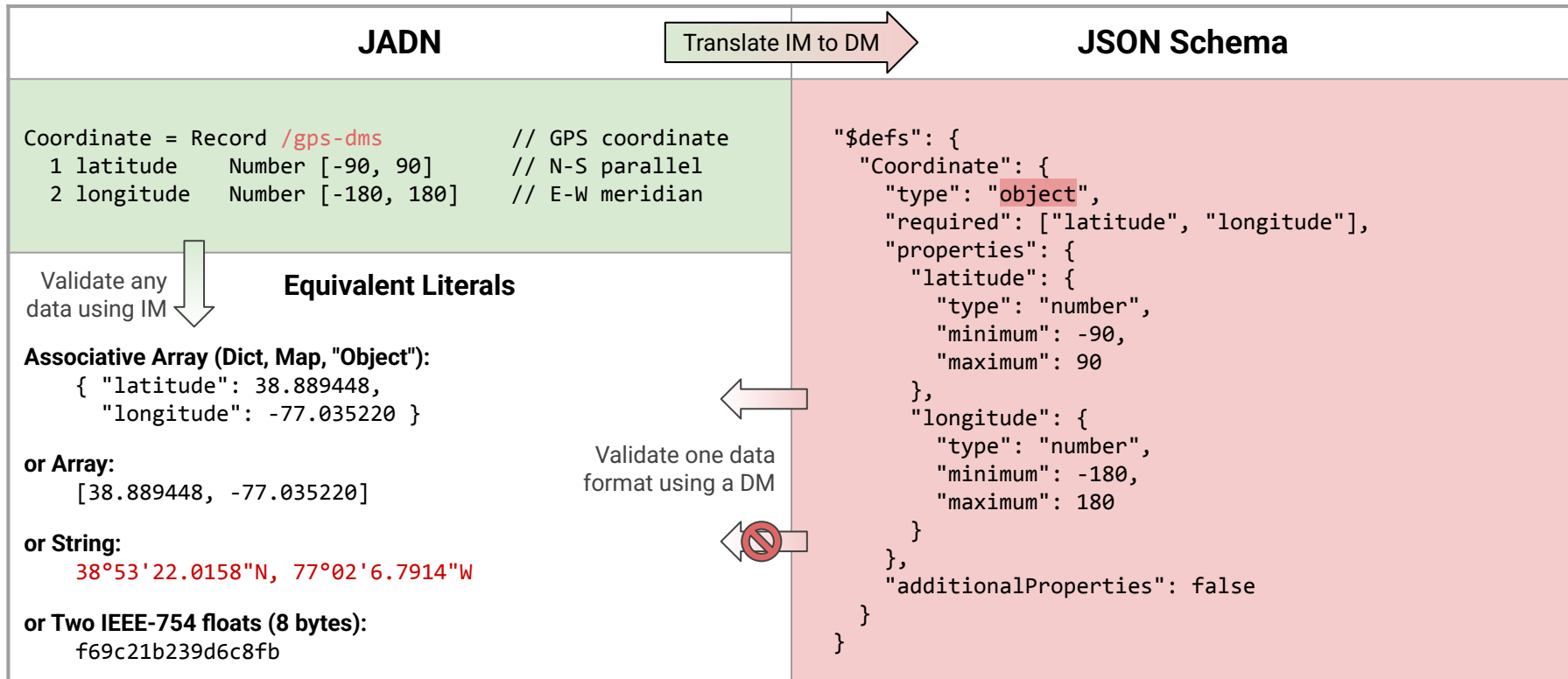
A message is an XML document that is a well-formed XML data structure with a root element that is valid as defined by a normative XML schema provided with the specification. Each message MAY reference one or more binary attachments. The transmission format of messages and attachments is defined in a service interaction profile

Table 1. Messages

Providing MDE	Consuming MDE	Operation	Input Message XML element(s)	Output Message XML element
Court Policy	Filing Assembly	GetPolicy	policyrequest:GetPolicyRequestMessage	policyresponse:GetPolicyResponseMessage
Court Record	Court Scheduling	AllocateCourtDate	allocatedate:AllocateCourtDateMessage	cbnr:MessageStatus
		GetCase	caserequest:GetCaseRequestMessage	caseresponse:GetCaseResponseMessage
		GetCaseList	caselistrequest:GetCaseListRequestMessage	caselistresponse:GetCaseListResponseMessage
		GetDocument	documentrequest:GetDocumentRequestMessage	documentresponse:GetDocumentResponseMessage

A defined message sent from A to B

Why use JADN? Abstract design => Simple yet precise specifications.



Information Model DataTypes

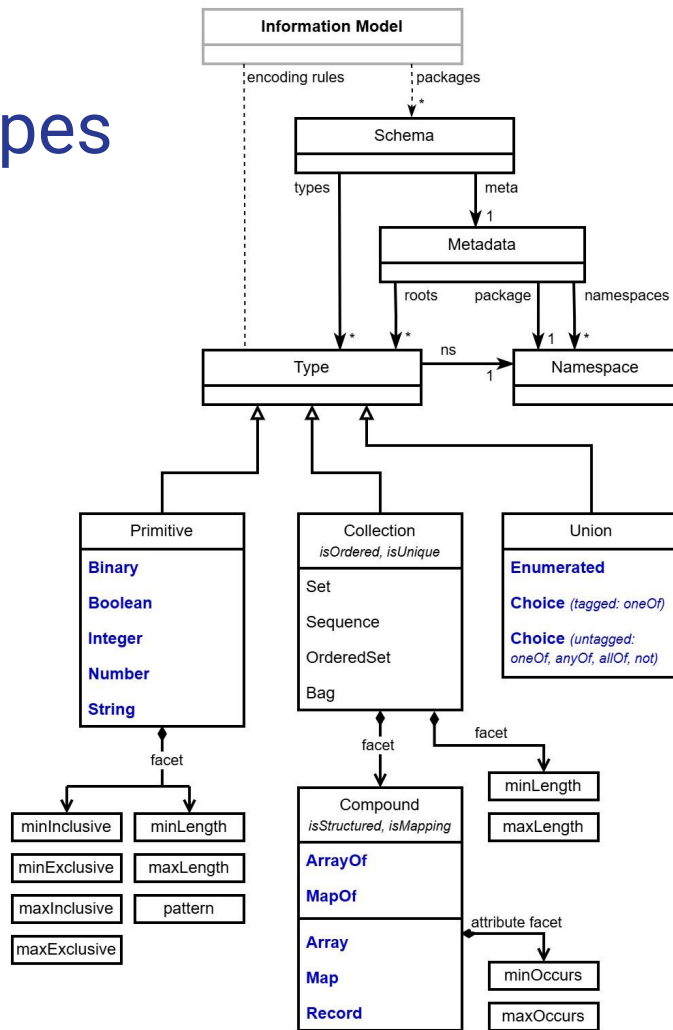
JADN Schema

- Package(s) identified by namespace

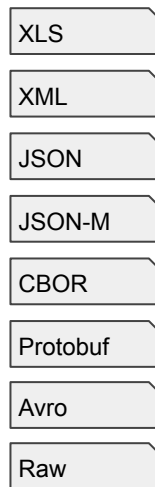
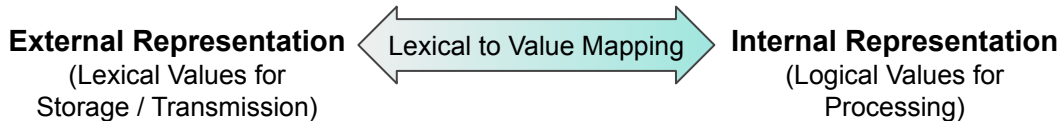
Minimal set of 12 core DataTypes:

- 5 Primitive
- 5 Compound
- 2 Union

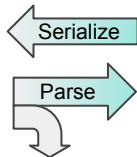
Single internal representation regardless of external data format



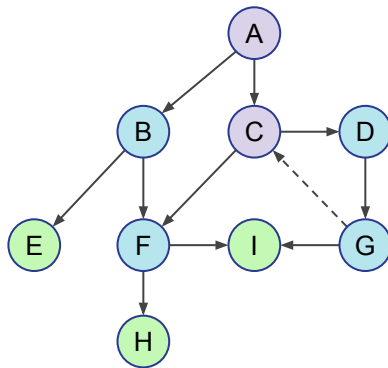
Information Equivalence



Translate:
Parse from one format
Serialize to another



Insignificant
Data



- = Compound Datatypes with ID / Primary Key
- = Compound Datatypes with no ID
- = Primitive Datatypes
- = Contain
- > = Reference (Foreign Key)

Information Model
DataTypes

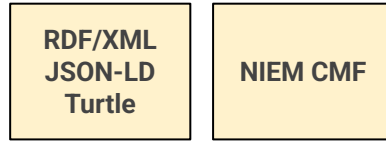
● **Primitive**
Binary
Boolean
Integer
Number
String

●● **Collection Type**
Set
Sequence
OrderedSet
Bag

Compound Type
ArrayOf
Array
MapOf
Map
Record

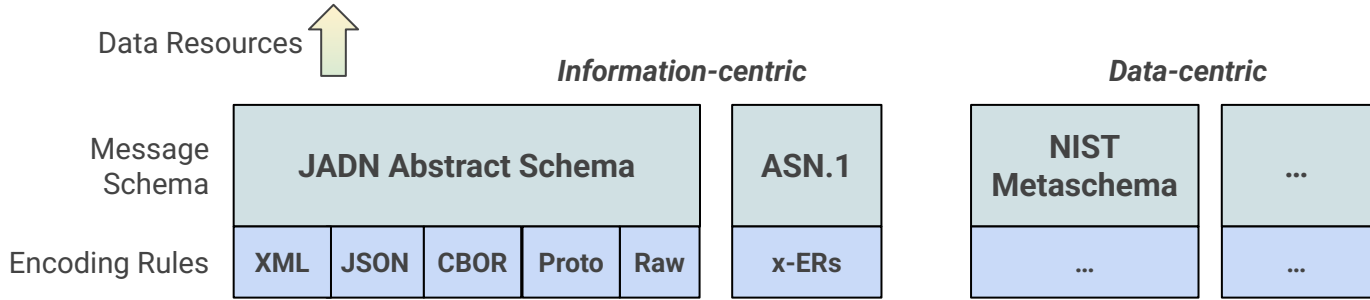
Union
Enumerated
Choice

Ontology /
Knowledge Graph



Data Resources

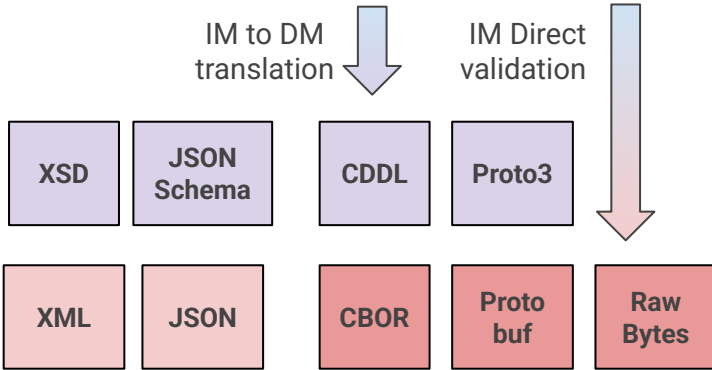
Information
Model



Information-centric

Data-centric

Data
Model

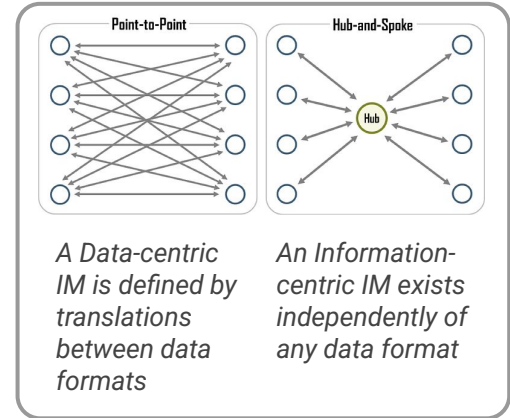


Data

Ontology
Formats

RDF/XML JSON-LD

No RDF Formats



Resources and Models

Physical Resources



Person, Organization



Building, Device



Processing HW/SW



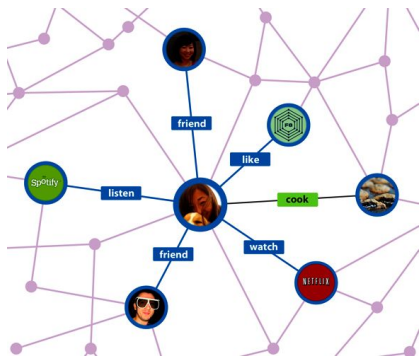
Event

Ontology / Knowledge Graph

RDF nodes *describe* physical and data resources. Resources exist independently of any graph.

"The node is not the territory"

Edges define relationships among nodes, enriching knowledge about them.



Data Resources

DataType = Model (a blank form)

Value = Data (fills in a form)

Form

Identity

Blueprint

Bill of materials

Sensor reading

Playbook

Report, Log

Value



Document



Message, Packet



Structure, PDU



Image, Media

Information Model

DataTypes *define* the *essential* content of data resources independently of data format, abstracting away insignificant detail.

Value = instance of an abstract DataType

Logical Value = essential content / meaning

Literal Value = sequence of bytes / characters

Data Model

DataTypes *define* the content of data resources in a fixed data format.

Value = instance of a concrete DataType

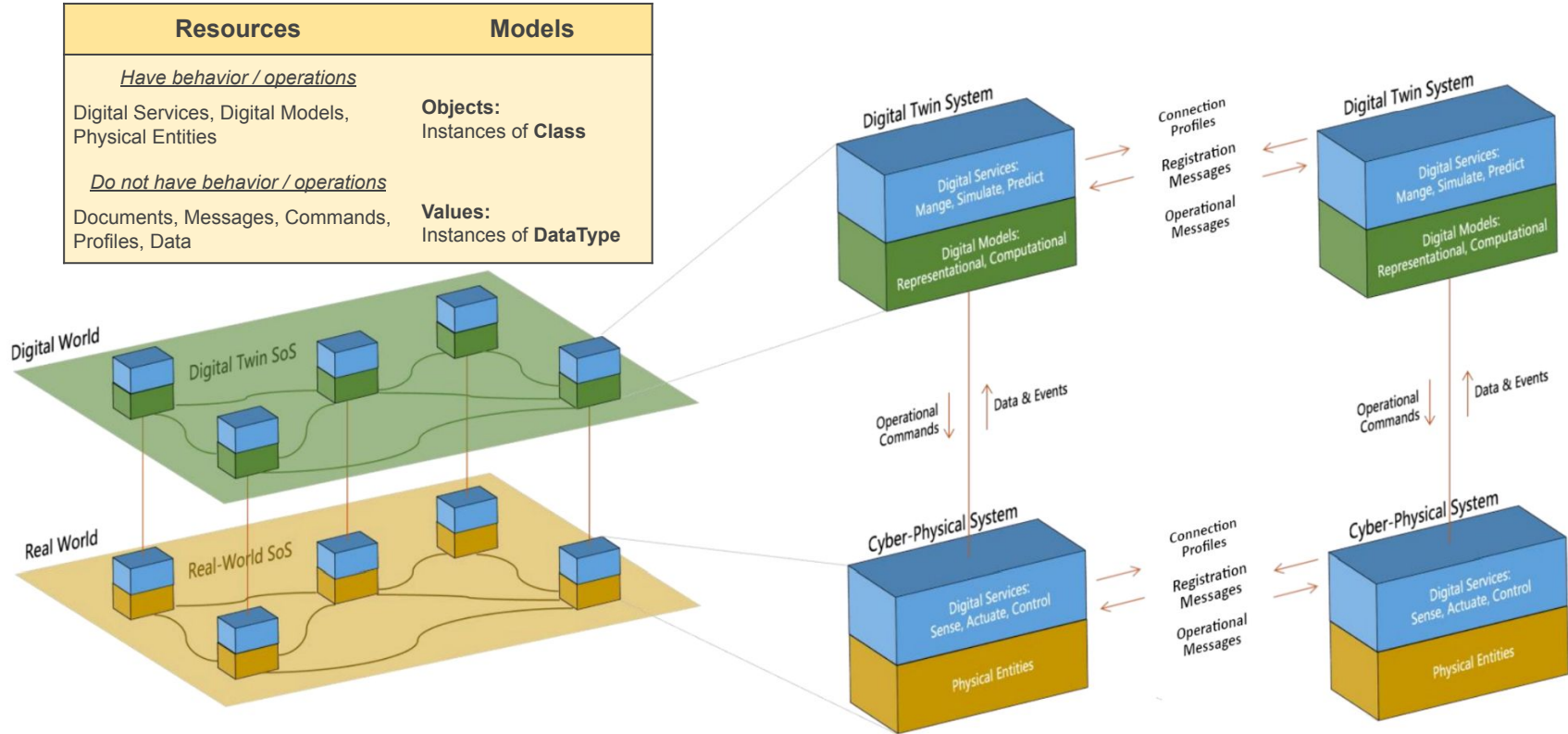
Value = sequence of bytes / characters

Object Model


Objects are instances of Class. Objects model processes / operations and are not Values that can be hashed or compared.

Values are instances of DataType. Information models are composed of DataTypes, not Classes.

Digital Twins: Resources and Models



Differences from JADN v1

- Shift emphasis from "Information Theory" to "Conceptual Design"
 - Move expository content from Specification into separate Committee Note
 - Add capabilities:
 - Type Inheritance
 - Untagged Unions (anyOf, oneOf, allOf)
 - Separate range, length, and occurrence count options (required new major version)
 - XML serialization rules
 - UML collection model
 - Updated type options
 - Additional semantic validation keywords
 - Package composition using namespaces
- 

Resources

JADN Specification

<https://docs.oasis-open.org/openc2/jadn/v2.0/jadn-v2.0.html>

Comments

<https://groups.oasis-open.org/discussion/invitation-to-comment-on-openc2-jadn-v20-csd01>

Information Modeling with JADN

<https://docs.oasis-open.org/openc2/imjadn/v1.0/imjadn-v1.0.html>

OpenC2 Technical Committee

<https://groups.oasis-open.org/communities/tc-community-home2?CommunityKey=a34c9baf-48b2-44c5-a567-018dc7d32296>

