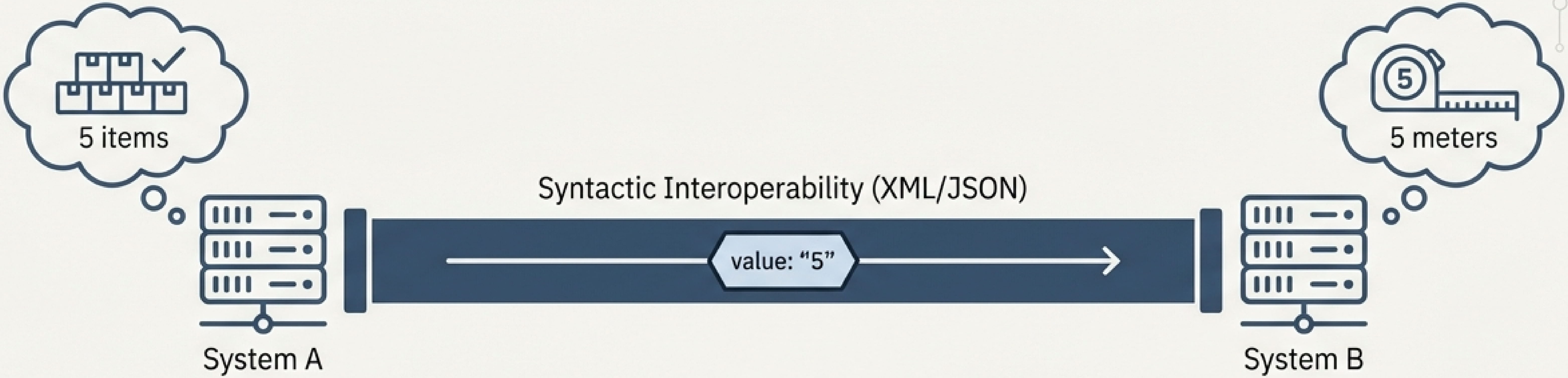
Axius SDC

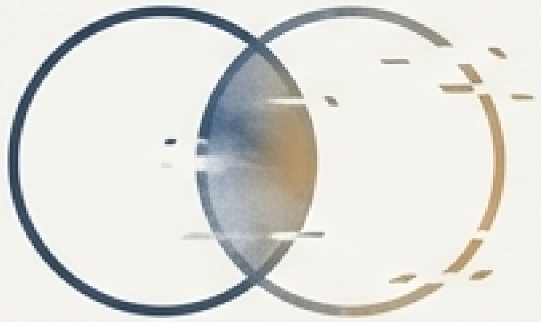
The Semantic Interoperability Crisis

Why Contemporary Standards Fail and How SDC4 Provides a New Blueprint for Trusted, AI-Ready Data

We Solved Syntax, But We Failed on Semantics.

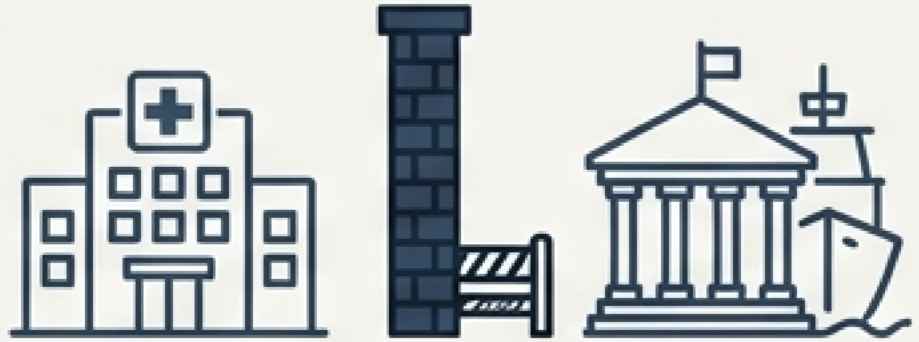
For decades, we’ve focused on ensuring systems can *parse* the same XML or JSON. But we face a deepening crisis: they don’t understand what the data *means*.





Semantic Drift: Critical context is lost in translation between systems.

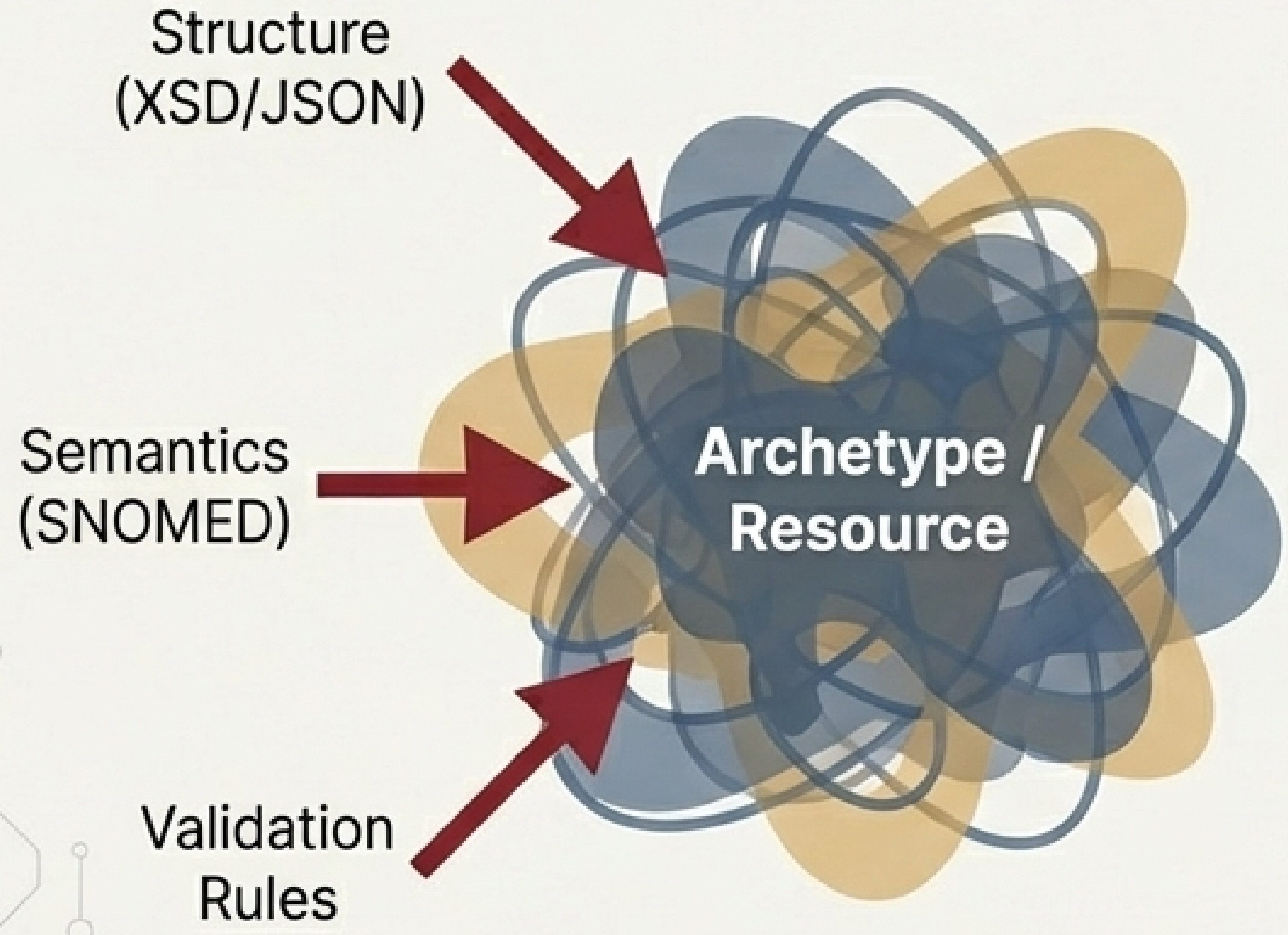
AI ‘Hallucinations”: Probabilistic models fail when fed ambiguous, context-poor data.



Domain Silos: A ‘Patient Record’ in healthcare is alien to a ‘Vessel Inspection’ system in maritime.

The “Original Sin” of Data Standards

Most standards conflate *how* data is encoded with *what* it means. This fundamental architectural flaw creates complexity, semantic ambiguity, and tooling overhead.



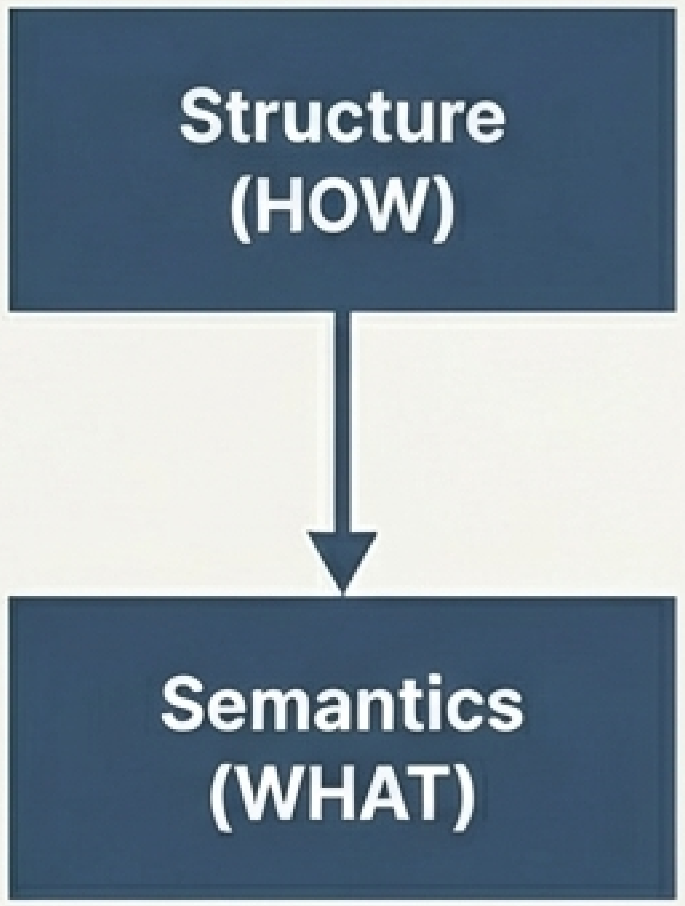
Structure (XSD/JSON)

Semantics (SNOMED)

Validation Rules

Archetype / Resource

The Conflated Approach (openEHR, FHIR, NIEM)



Structure (HOW)

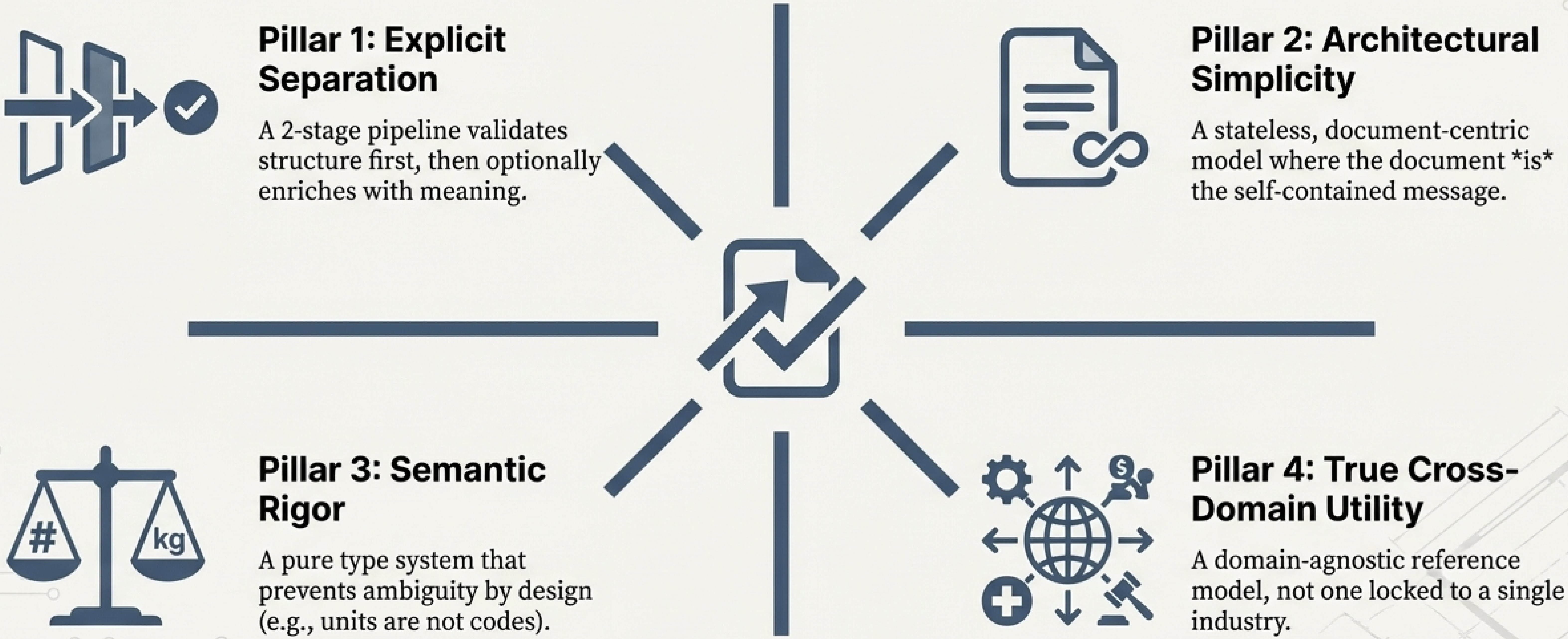
Semantics (WHAT)

The SDC4 Principle: Explicit Separation

“It is impossible to process structure independently from semantics in conflated models.”

The SDC4 Paradigm: An Architecture of Clarity

SDC4 is not another silo. It’s a new architectural paradigm derived from the first principle of separation.



Pillar 1: Explicit Separation

A 2-stage pipeline validates structure first, then optionally enriches with meaning.

Pillar 2: Architectural Simplicity

A stateless, document-centric model where the document *is* the self-contained message.

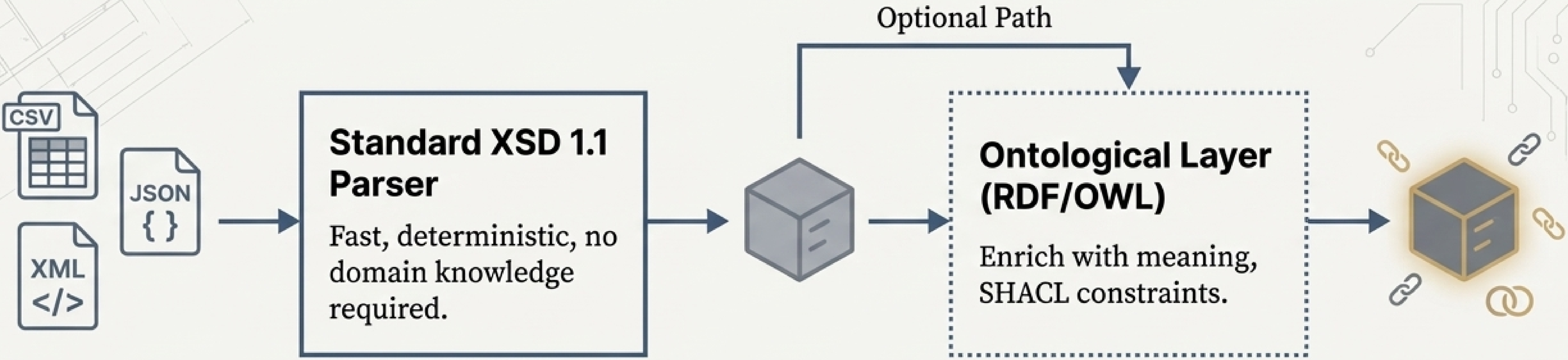
Pillar 3: Semantic Rigor

A pure type system that prevents ambiguity by design (e.g., units are not codes).

Pillar 4: True Cross-Domain Utility

A domain-agnostic reference model, not one locked to a single industry.

Pillar 1 In-Depth: How Separation Creates Simplicity


Optional Path
CSV
JSON
{ }
XML
</>
Standard XSD 1.1 Parser
Fast, deterministic, no domain knowledge required.
Ontological Layer (RDF/OWL)
Enrich with meaning, SHACL constraints.

SDC4

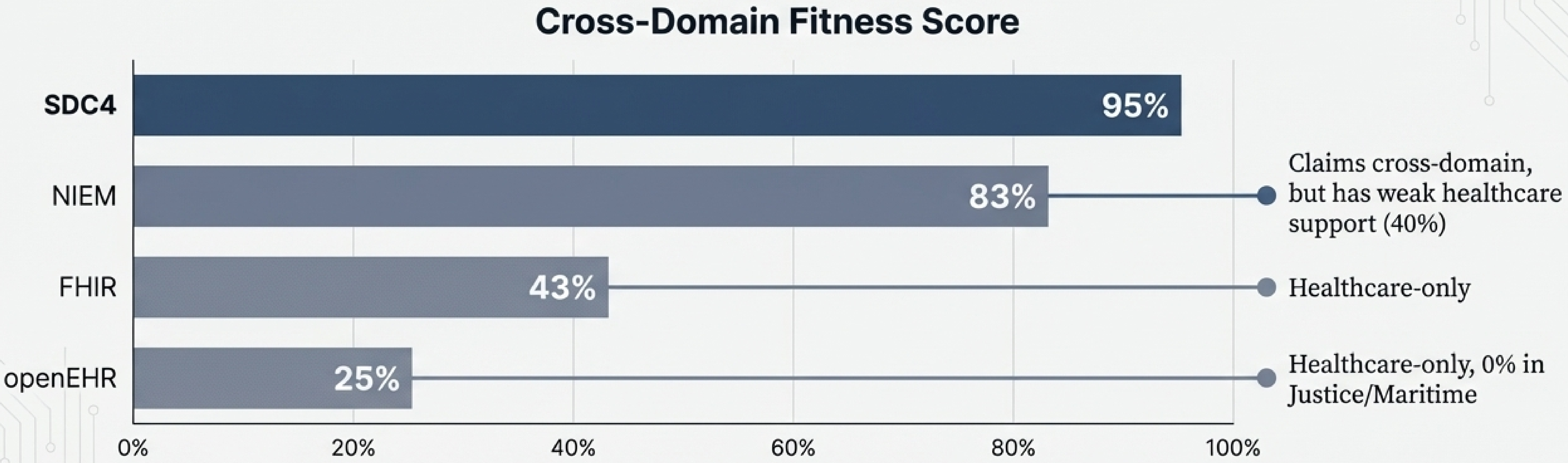
Leverages the massive existing XML tooling ecosystem. Uses standard XML Schema 1.1, validated by any compliant parser (lxml, Saxon, Xerces).

openEHR

Requires specialized ADL (Archetype Definition Language) parsers and tools, creating a high barrier to entry and tooling complexity.

Pillar 4 In-Depth: From a Single Silo to a Universal Blueprint

A truly ubiquitous standard must work across industries. SDC4 was designed for this; others were not.



Claims cross-domain, but has weak healthcare support (40%)

Healthcare-only

Healthcare-only, 0% in Justice/Maritime

Based on use case validation across Justice, Heallcare, Immigration, and Maritime domains (Ubiquitous Semantic Interoperability White Paper, Table 7.5).

SDC4 is the only standard with consistent high performance across all tested domains.

SDC4 vs. The Incumbents: A Head-to-Head Comparison

Feature	SDC4	openEHR	FHIR	NIEM
Separates Structure from Semantics	✓	✗	✗	✗
Mandatory Units for Quantities	✓	✓	✗ (Optional)	✗ (Text only)
Distinguishes Units vs. Codes	✓	~ (Partial)	✗	✗
Automatic RDF Triple Generation	✓	✗ / Limited	✗ / Limited	✗
Cross-Domain Utility Score	**95%**	25%	43%	83%
Tooling Requirement	Standard XML	Proprietary ADL	Custom Ecosystem	Custom Ecosystem
Semantic Rigor Score (Weighted)	**93%**	69%	59%	42%

The Difference in Practice: Why a ‘Number’ Is Not Enough

Modeling a Blood Glucose value of ‘5.5 mmol/L’.

NIEM

```
<nc:MeasureValue>5.5</nc:MeasureValue>
<nc:MeasureUnitText>mmol/L</nc:MeasureUnitText>
```

Ambiguous. Units are free text, prone to typos. No semantic link.

FHIR

```
{
  "valueQuantity": {
     "value": 5.5,
     "unit": "mmol/L",
     "system": "http://unitsofmeasure.org",
     "code": "mmol/L"
```

Optional. `unit` is not mandatory, allowing for values to exist without units, creating ambiguity.

SDC4

```
<BloodGlucose>
  <value>5.5</value>
  <units>mmol/L</units>
</BloodGlucose>
```

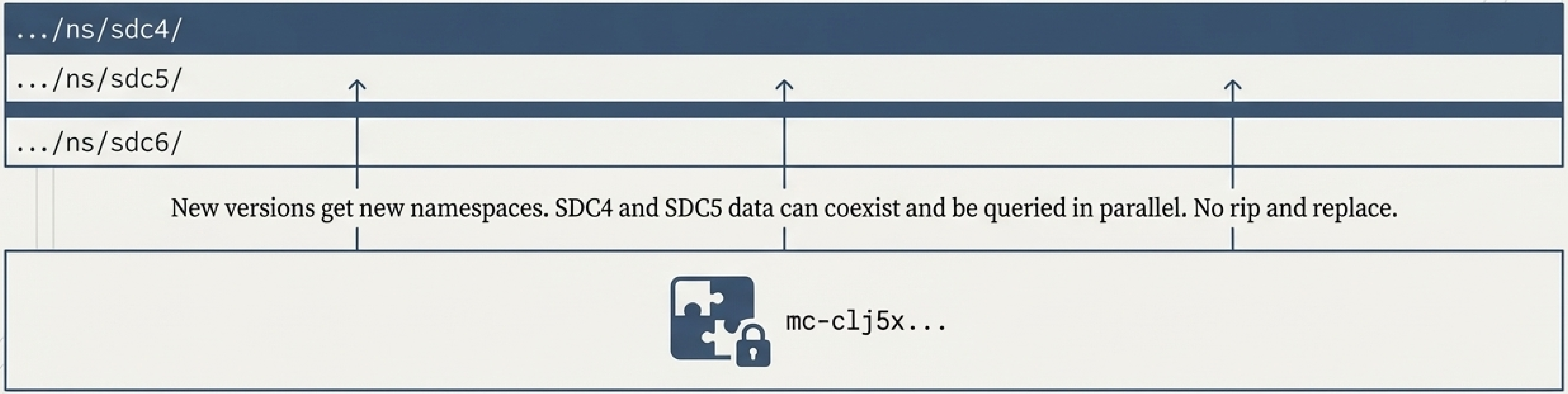
Enforced. `units` is a mandatory foreign key to a controlled, semantic-rich Units model. Ambiguity is architecturally prevented.

Designing for Decades, Not Just for Deployment

Software changes. Forced data migrations cost millions and introduce risk.

The SDC4 Solution: **A dual-layer Permanence Architecture.**

Layer 2: Namespace Versioning



```
.../ns/sdc4/
.../ns/sdc5/
.../ns/sdc6/
```

New versions get new namespaces. SDC4 and SDC5 data can coexist and be queried in parallel. No rip and replace.

Layer 1: A component’s structure is identified by a permanent CUID and locked forever. Create new models without touching old data.

Key Takeaway: Your 2025 SDC4 data is designed to validate against its original SDC4 models in 2045.

Is SDC4 ‘Missing’ an API? No. It’s a Deliberate Architectural Choice.

SDC4’s document-centric architecture prioritizes simplicity, determinism, and transport independence.

FHIR (API-Centric)	SDC4 (Document-Centric)
Paradigm: Resource-centric	**Paradigm:** The document is the self-contained contract
State: Server-managed (complex)	**State:** Stateless (simple)
Validation: Runtime at endpoint	**Validation:** Deterministic & offline capable
Transport: Tightly coupled to HTTP REST	**Transport:** Agnostic (HTTP, Kafka, AMQP, File)

The “Best of Both Worlds” Strategy

Like openEHR, use SDC4 for its superior semantic modeling, and expose data via a FHIR API when the ecosystem requires it.

We Didn't Reinvent the Wheel. We Assembled It with Precision.

SDC4 is an engineered assembly of the world's most trusted open standards.

OWL

SHACL

SPARQL

W3C

ISO

XSD 1.1

ISO 21090

ISO 8601

SDC4 Reference Model
Open Standard
Specification on GitHub.

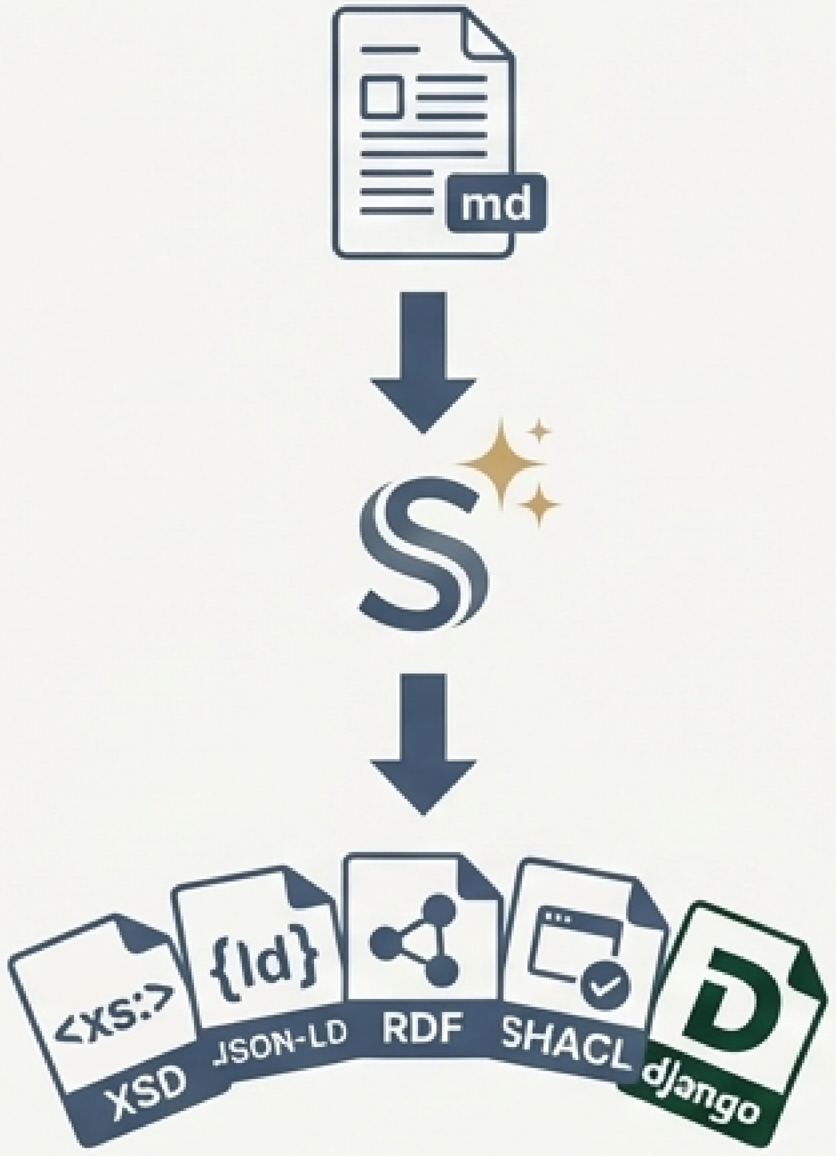
`sdcvalidator`
Production-ready
Python library on PyPI.

`sdcvalidatorJS`
TypeScript/Node.js
library on npm.

The foundation of data trust should be open, accessible, and community-driven.

From Model to Application: SDC Studio

SDC Studio bridges the gap between domain knowledge and production-ready, semantically-rich applications—no coding required.



DESIGN

Domain experts describe data in plain language using simple templates (Obsidian, CSV).

GENERATE

AI-powered engine generates schemas, knowledge graphs, and documentation.

DEPLOY

One model, nine output formats, including full-stack applications.

The Vibe Coding Revolution

Formal model + AI prompt = Production-quality app in one iteration.

The Strategic Advantage of an SDC4 Foundation

AI-Ready by Design

The semantic rigor and knowledge graphs required for next-gen AI and Graph RAG are built-in, not bolted on. Your data has “physics.”

Ubiquitous Interoperability

Finally achieve true cross-domain data data exchange—from healthcare to justice to maritime—without manual translation or semantic loss.

Radically Reduced Technical Debt

The Permanence Architecture and standards-based approach minimize costly data migrations, freeing investment for innovation.

“Make everything as simple as possible, but no simpler.”

— Albert Einstein

The Industry Doesn’t Need a Better API. It Needs a Better Blueprint.

1. Contemporary standards have hit a semantic ceiling because they conflate structure and meaning.

2. SDC4’s separation principle, learned from the strengths and weaknesses of openEHR, provides a superior foundation.

3. This results in provably better semantic precision, true cross-domain utility, and long-term data permanence.

For organizations requiring strictly governed, cross-domain data exchange
For organizations requiring strictly governed, cross-domain data exchange
with AI-readiness built in, SDC4 represents the only viable path forward.

Explore the Blueprint. Build the Future.

Read the White Paper

Link to the detailed "Ubiquitous Semantic Interoperability" white paper.

Explore the Open Source

Link to the GitHub organization to see the reference model and validation tools.

Schedule a Technical Deep-Dive

Link or email for a demo of SDC Studio.

contact@axius-sdc.com
semanticdatacharter.com
github.com/SemanticDataCharter