SDW update

- SDW re-charter
- GeoDCAT deliverable
  - potential for related WoT profiles
- Integration of OGC and application domains
  - FIWARE model examples
  - “Cross domain model”

Spatial Data on the Web WG

Joint W3C/OGC

- Purpose
- Original charter scope
- Re-charter
- Proposed mechanism

Fspl3brmfasPFw88s

Purpose

A means to allow W3C and OGC membership to collaborate in specification development.

Intended to act as a forum for considering common interests.

- but has not been active in this role in general

Original charter

- based on “Semantic Sensor Network” alignment with OGC/ISO “Observations and Measurements”
- Derived basic ontology: “**Sensor, Observation, Sample, and Actuator (SOSA)**”
- SSN retained axioms and details of system characteristics
- Some related work published as notes – e.g. use in RDF DataCube
- Expired mid 2024

Fspl3brmfasPFw88s

Re-charter

- About to go out for public comment
- https://w3c.github.io/charter-drafts/2024/sdw-wg.html
- “Work with OGC Standard Working Groups to jointly develop, maintain and promote geospatial Web standards and geospatial profiles of more general Web standards”
- SOSA/SSN
- GeoDCAT
- GeoSPARQL 1.3/2.x

Other..

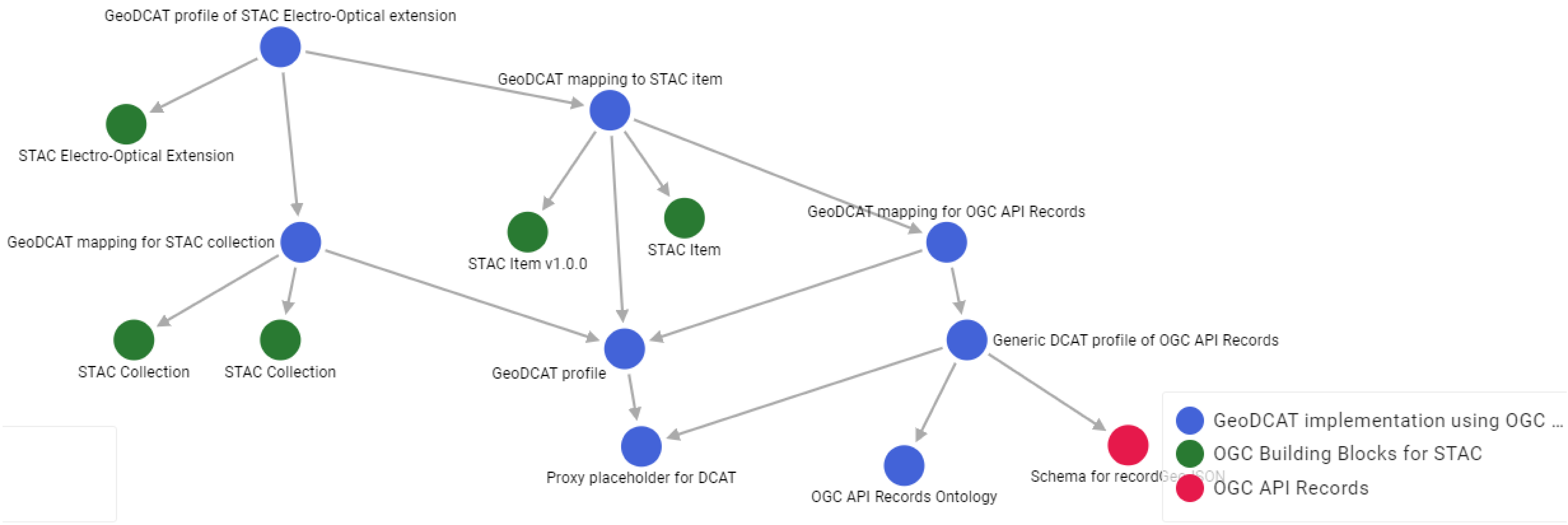
- How might WoT relate to..
  - “Connected Systems” – OpenAPI version of “Sensor Web Enablement” model?
  - https://www.ogc.org/requests/ogc-requests-public-comment-on-ogc-api-connected-systems-and-supporting-sensor-standards-before-adoption/
  - Note on alignment? JSON-LD contexts? Testing?
- Sensor Things API v2 planned… can we align better?
- CityGML -> OWL representation (and profiles)
- Digiital Twins WG – recharted – open to work items?
- (https://www.ogc.org/about-ogc/committees/dwg/urban-digital-twins-domain-working-group/)

GeoDCAT

- DCAT (W3C DXWG) does not define
  - Any specific profiles
  - Any profiling mechanism
- GeoDCAT is a WIP of a Geospatial Profile
- EU already published a GeoDCAT-AP
  - But Eurocentric and limited in scope
- Metadata needs to be far more extensible
- GeoDCAT will be a core and register of extension profiles of DCAT

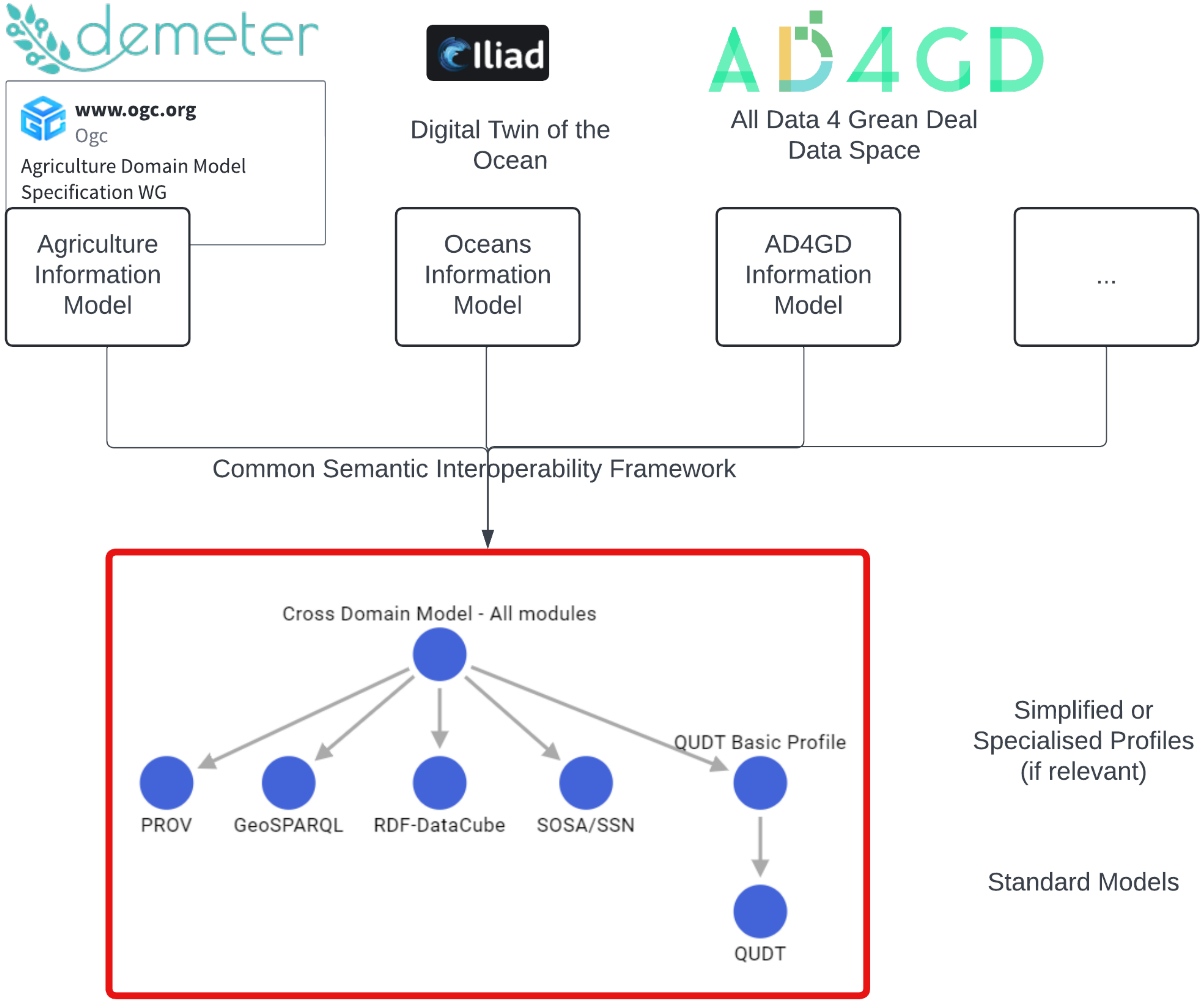
- => analogous to TD profiles – work to be done exploring this?

DCAT – a family of profiles


GeoDCAT profile of STAC Electro-Optical extension
STAC Electro-Optical Extension
GeoDCAT mapping to STAC item
GeoDCAT mapping for STAC collection
STAC Item v1.0.0
STAC Item
GeoDCAT mapping for OGC API Records
STAC Collection
STAC Collection
GeoDCAT profile
Generic DCAT profile of OGC API Records
Proxy placeholder for DCAT
OGC API Records Ontology
Schema for recordGeoJSON
GeoDCAT implementation using OGC ...
OGC Building Blocks for STAC
OGC API Records

- Modules for:
  - Ontologies
  - Profile constraints
  - Schemas
  - (APIs, Test cases, Transformations)

W3C foundations for “domain models”



demeter

www.ogc.org
Ogc
Agriculture Domain Model Specification WG

Iliad

Digital Twin of the Ocean

AD4GD

All Data 4 Grean Deal Data Space

Agriculture Information Model

Oceans Information Model

AD4GD Information Model

...

Common Semantic Interoperability Framework

Cross Domain Model - All modules

QUDT Basic Profile

PROV

GeoSPARQL

RDF-DataCube

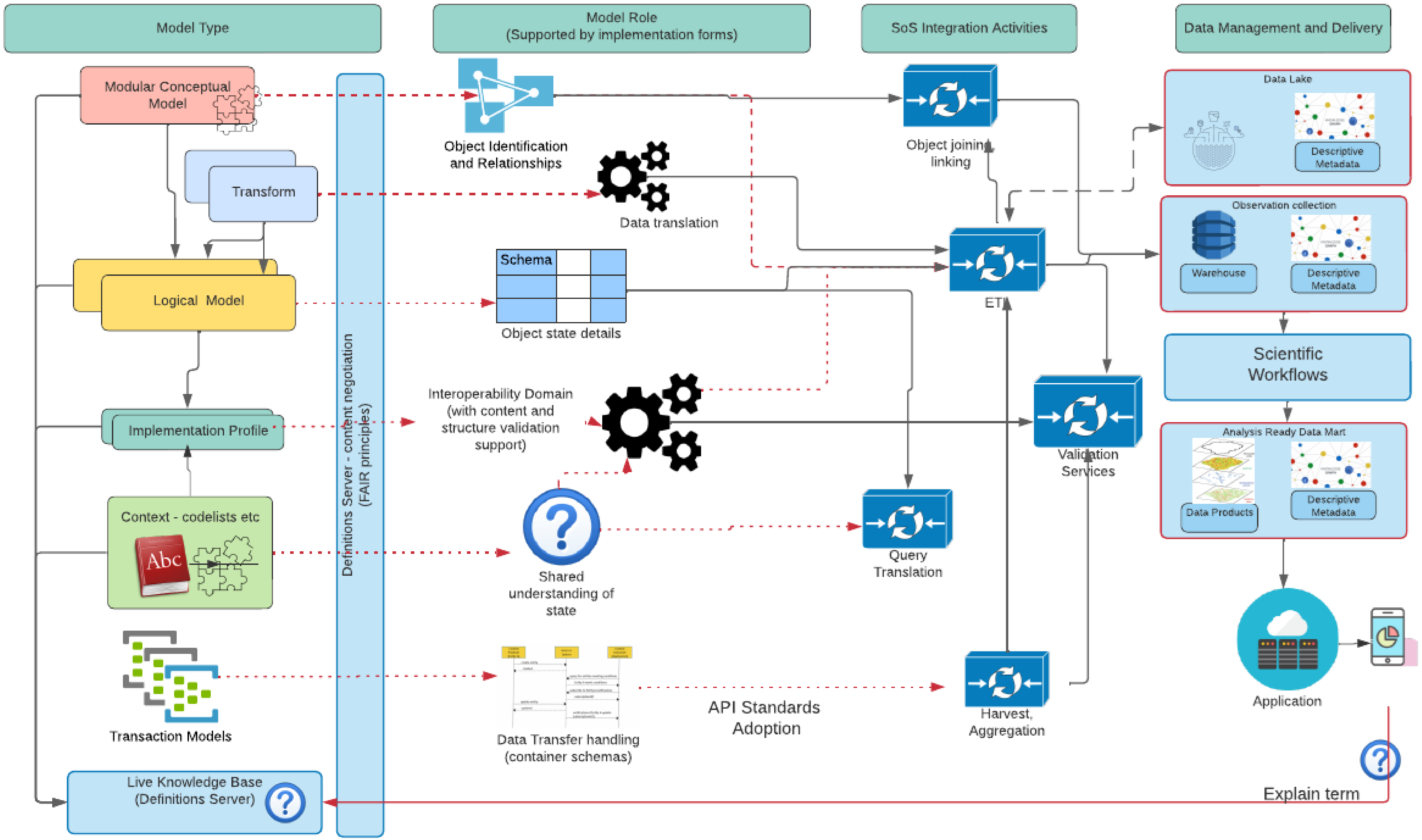
SOSA/SSN

QUDT

Simplified or Specialised Profiles (if relevant)

Standard Models

“Systems of Systems” View



Model Type

Model Role (Supported by implementation forms)

SoS Integration Activities

Data Management and Delivery

Modular Conceptual Model

Transform

Logical Model

Implementation Profile

Context - codelists etc

Transaction Models

Live Knowledge Base (Definitions Server)

Definitions Server - content negotiation (FAIR principles)

Object Identification and Relationships

Data translation

Schema

Object state details

Interoperability Domain (with content and structure validation support)

Shared understanding of state

Data Transfer handling (container schemas)

API Standards Adoption

Object joining, linking

ETL

Validation Services

Query Translation

Harvest, Aggregation

Data Lake

Descriptive Metadata

Observation collection

Warehouse

Descriptive Metadata

Scientific Workflows

Analysis Ready Data Mart

Data Products

Descriptive Metadata

Application

Explain term

Open Geospatial Consortium

Aligning and learning

- *”Thing Descriptions, by default, are encoded in a JSON format that also allows JSON-LD processing. The latter provides a powerful foundation to represent knowledge about Things in a machine-understandable way.”*
- (https://www.w3.org/TR/wot-thing-description11/#abstract)
- OGC has inherited a legacy where JSON has not been designed this way (IETF GeoJSON
- OGC has been evolving a “Building Blocks” model for collections of features – with an initial focus on mapping schema fragments to JSON-LD
- Complex, but possible, to aggregate domain models from fragments and retain semantic annotations
- Working in background with OpenAPI, JSON-schema on this – can we have a more inclusive forum?

Solution

BUILDING BLOCKS

A better data model and API Design Framework

SURROUND leverages state of the art data models and frameworks to create simplified building blocks that connect enterprise assets seamlessly, simplifying what is usually a rather complex challenge.

PROFILING

Making standards work for specific applications

SCALABILITY

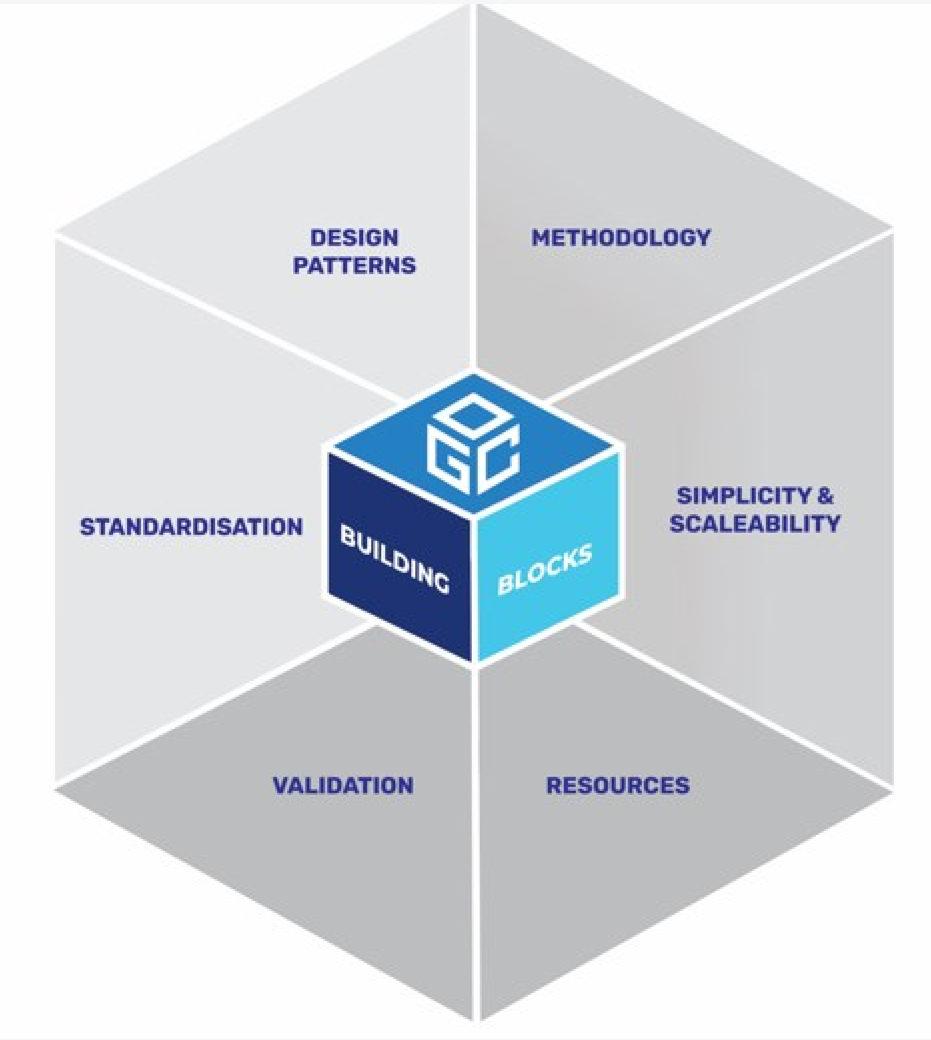
Cost effective reuse and power to keep on solving new problems

TRACEABILITY & TRANSPARENCY

Knowing what works together and what happens in complex systems

SEMANTIC INTEROPERABILITY

System of Systems - Complex data integration using standards, structure, meaning and content



COMPLEX APPLICATIONS

Simple parts, complete solutions. Bigger than the “sum of the parts”

RESOURCES

Open tooling and libraries of standards and solutions

FUTURE PROOFING

Support multiple technology versions from one tested design

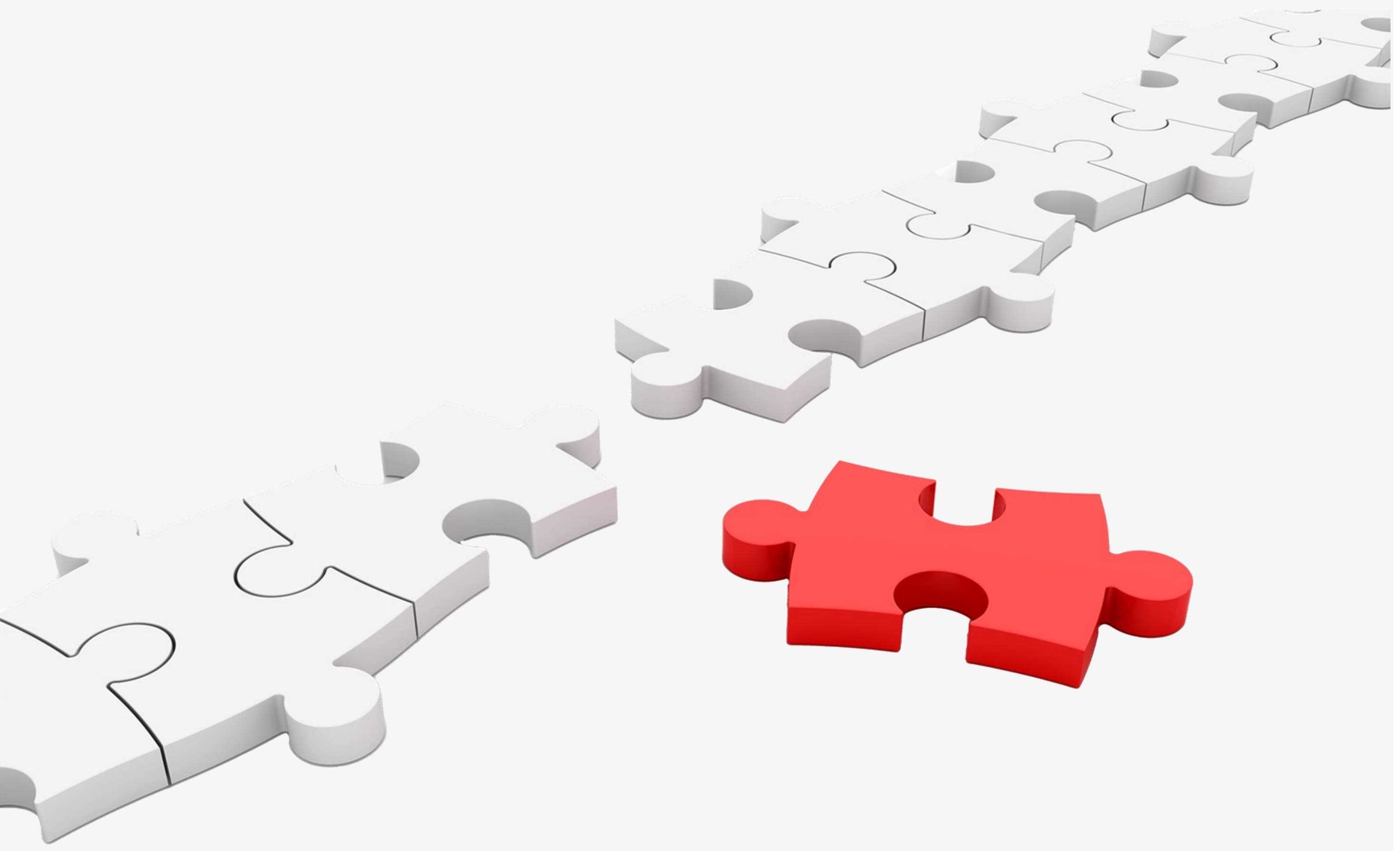
TESTING

Best practices applied at all levels provides robust, trustworthy and clearly documented solutions.

Proprietary model

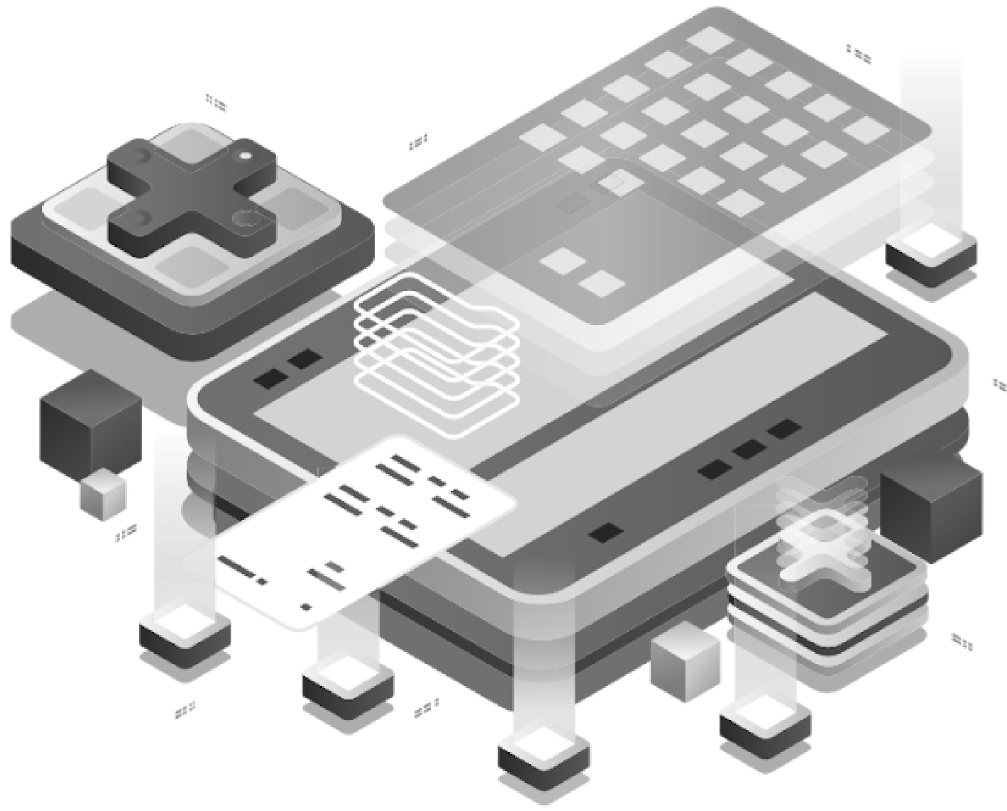
TRACEABILITY & TRANSPARENCY

Know what you know!

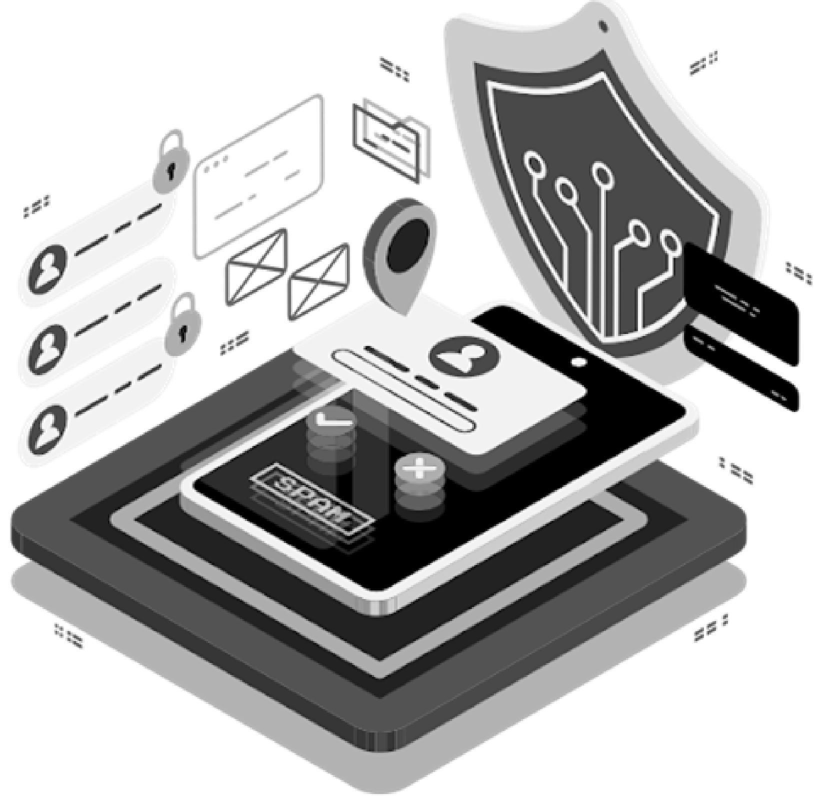


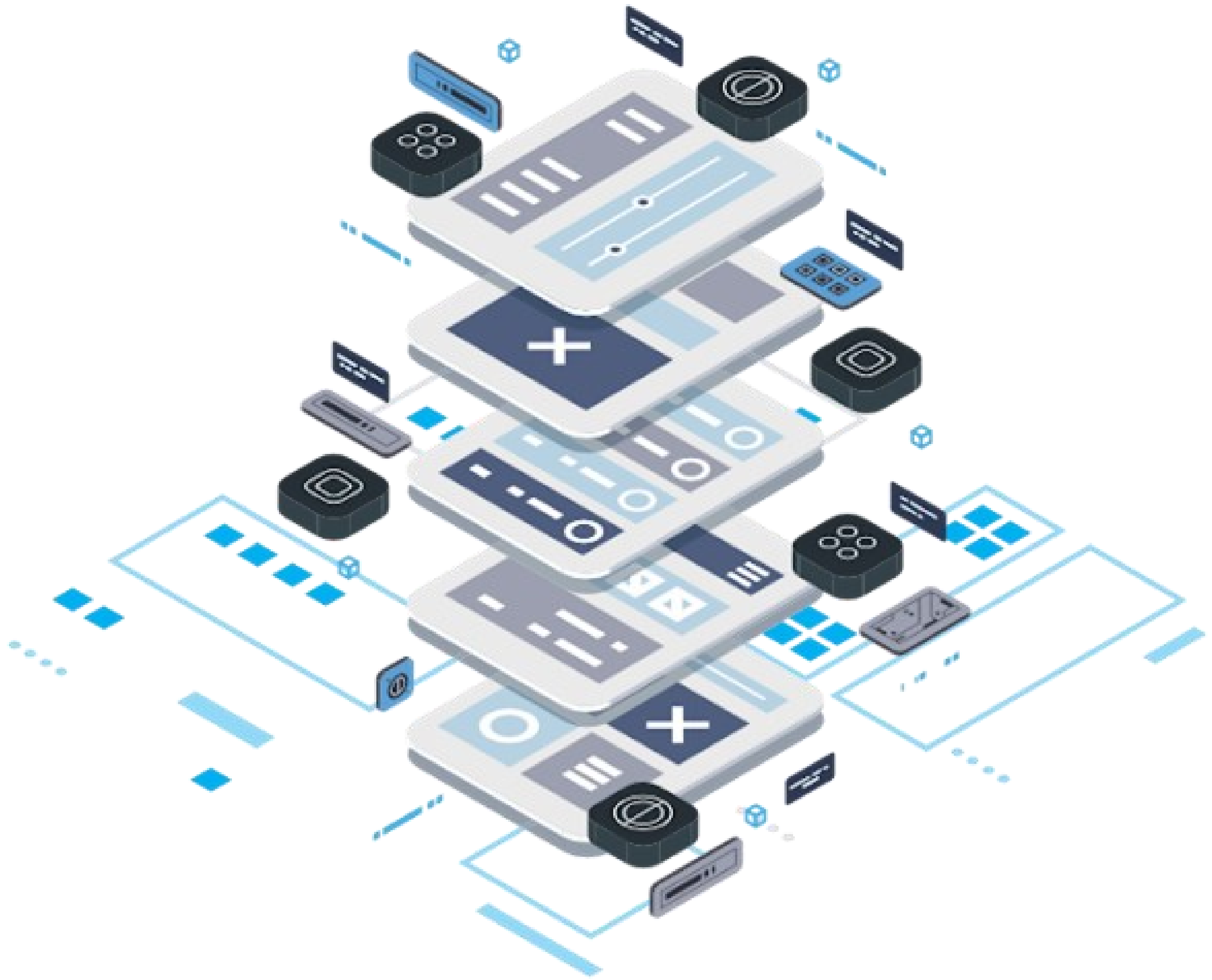
Knowledge graphs can identify how both standards and data is related. This traceability supports both efficiency and transparency, adding value and managing risks for data exploitation and reporting.

Copy. Paste. Modify.



Invent your own.





Link. Profile. Reuse.

No two applications are the same - but many share common problems.

Copy, paste, modify approaches hide this commonality. Inventing your own hurts you and the client in future.

Reuse via reference (linking) with profiling to explain how the re-use is done makes it all transparent. And is much cheaper to build and test!

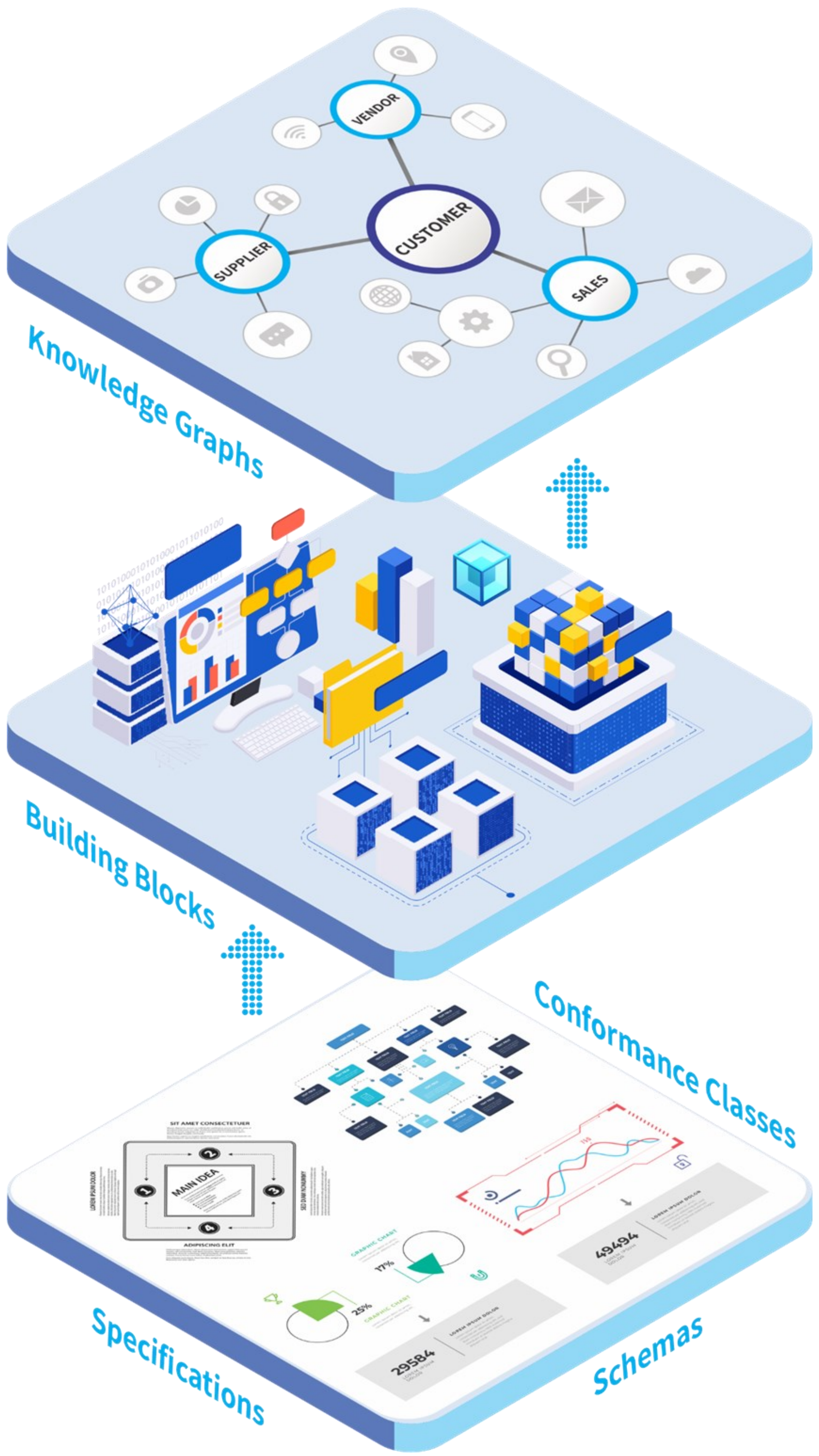
The dependency graph is the foundation of a semantic knowledge representation of how system components can interoperate.

Proprietary model

SCALABILITY

Many data sources means many challenges

Our approach allows us to identify common aspects and reuse effective solutions, and expand easily to meet specific needs of problem domains.



- Knowledge how:
  - standards relate
  - Data conforms to standards
  - Data relates
  - Services interact with data

- The bridge between sources and knowledge
- How pieces fit together
- Documentation and testing
- Tools to combine

- A large scattered body of complex documentation
- Many different forms
- Limited, if any, machine-readable information

01. HEALTHCARE

02. INDUSTRIAL & MANUFACTURING

03. AGRICULTURE

04. FINANCE

05. MOBILITY

06. GREENDEAL

07. ENERGY

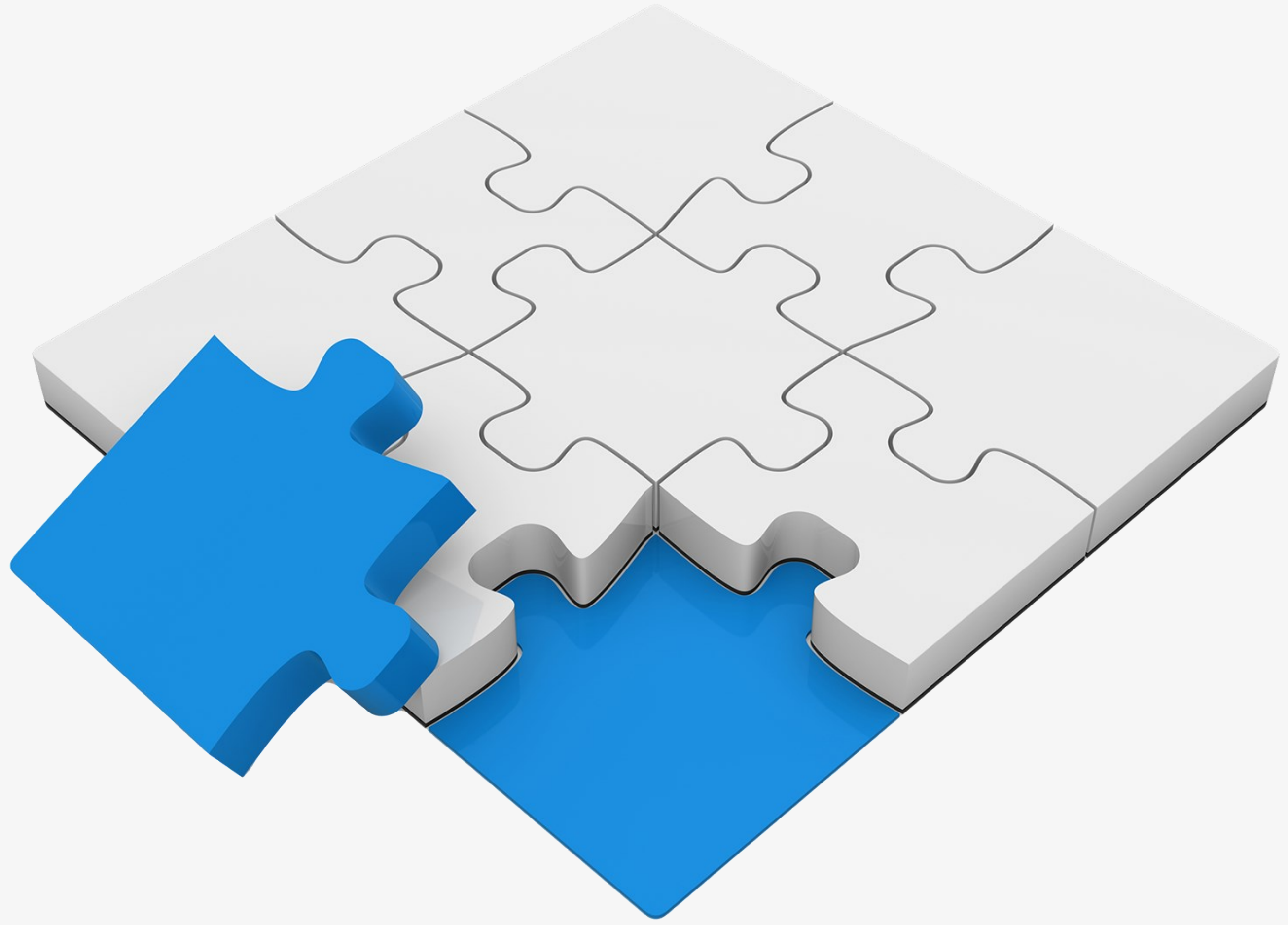
08. PUBLIC ADMINISTRATION

09. SKILLS

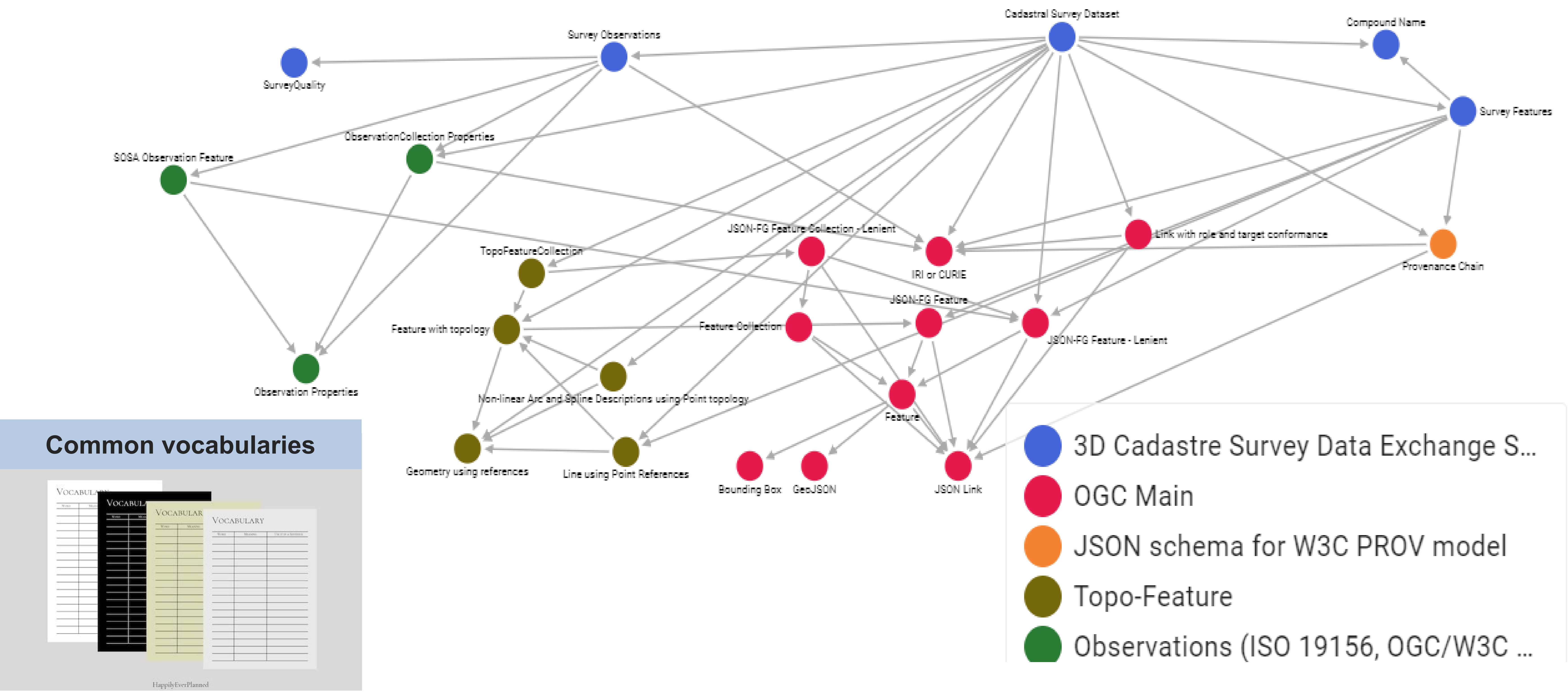
Proprietary model

COMPLEX APPLICATIONS

Manageable pieces, but unlimited power when combined.

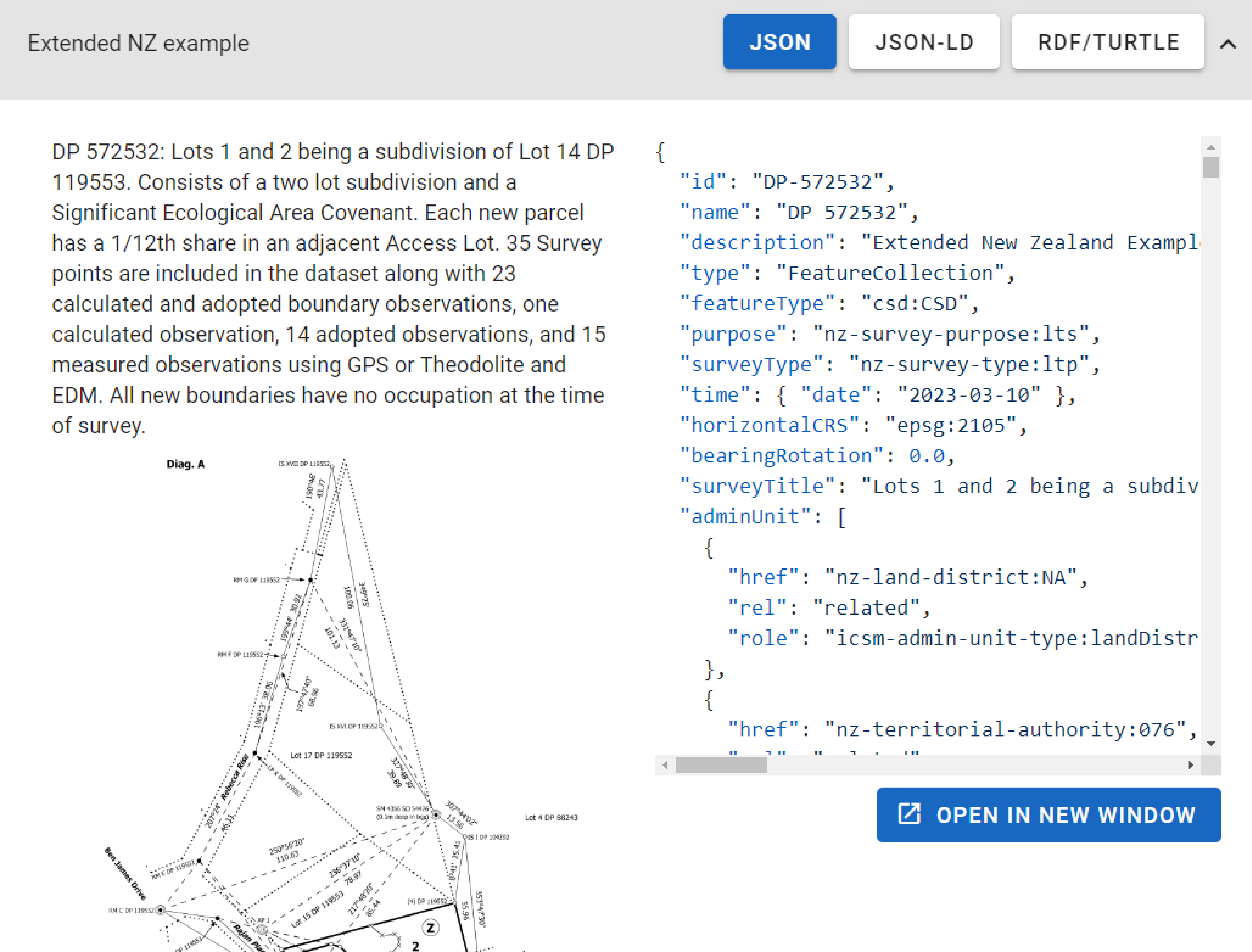


Reuse of well-tested components allows rapid assembly of solutions for arbitrary complex scenarios.



Cadastral Survey Data Exchange Model replaces an “one-size-fits-all” XML based solution that failed to fit anyone.

SURROUND, an OGC member, was able to demonstrate with hundreds of test cases that a wide range of special cases could all be handled using standardised Building Blocks and semantic profiling using controlled vocabulary resources.



DP 572532: Lots 1 and 2 being a subdivision of Lot 14 DP 119553. Consists of a two lot subdivision and a Significant Ecological Area Covenant. Each new parcel has a 1/12th share in an adjacent Access Lot. 35 Survey points are included in the dataset along with 23 calculated and adopted boundary observations, one calculated observation, 14 adopted observations, and 15 measured observations using GPS or Theodolite and EDM. All new boundaries have no occupation at the time of survey.

```
{
  "id": "DP-572532",
  "name": "DP 572532",
  "description": "Extended New Zealand Exampl
  "type": "FeatureCollection",
  "featureType": "csd:CSD",
  "purpose": "nz-survey-purpose:lts",
  "surveyType": "nz-survey-type:ltp",
  "time": { "date": "2023-03-10" },
  "horizontalCRS": "epsg:2105",
  "bearingRotation": 0.0,
  "surveyTitle": "Lots 1 and 2 being a subdiv
  "adminUnit": [
    {
      "href": "nz-land-district:NA",
      "rel": "related",
      "role": "icsm-admin-unit-type:landDistr
    },
    {
      "href": "nz-territorial-authority:076",
```

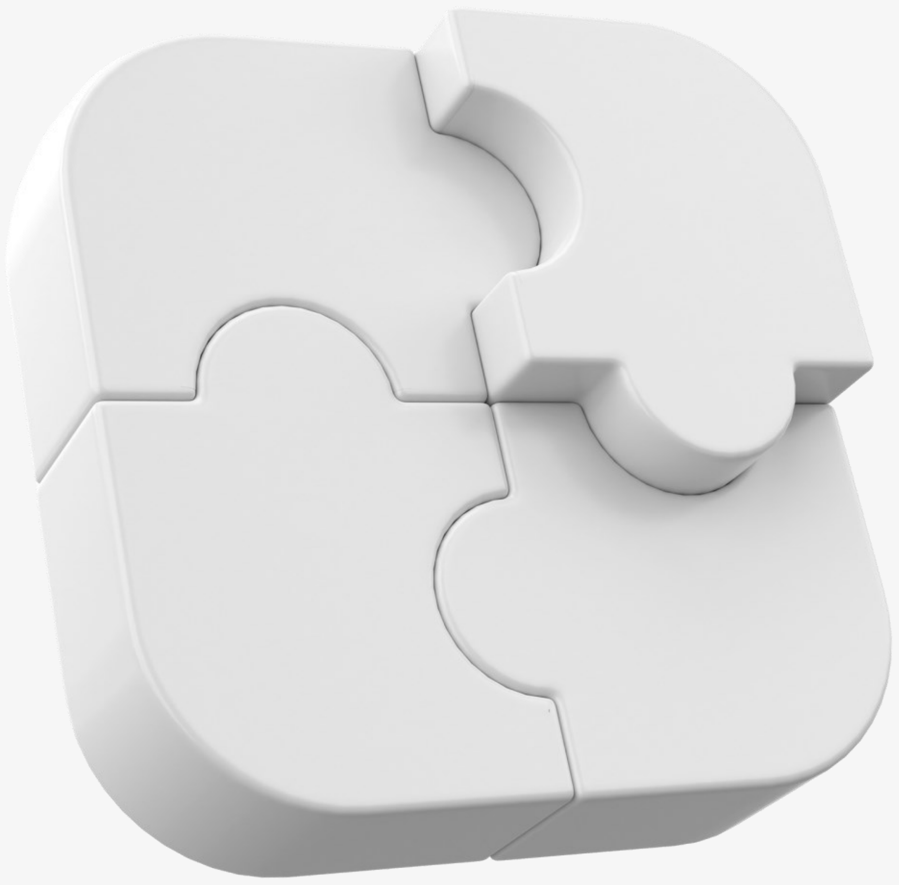
Many examples

- Comprehensive
- Single concepts
- Edge cases
- Failure testing

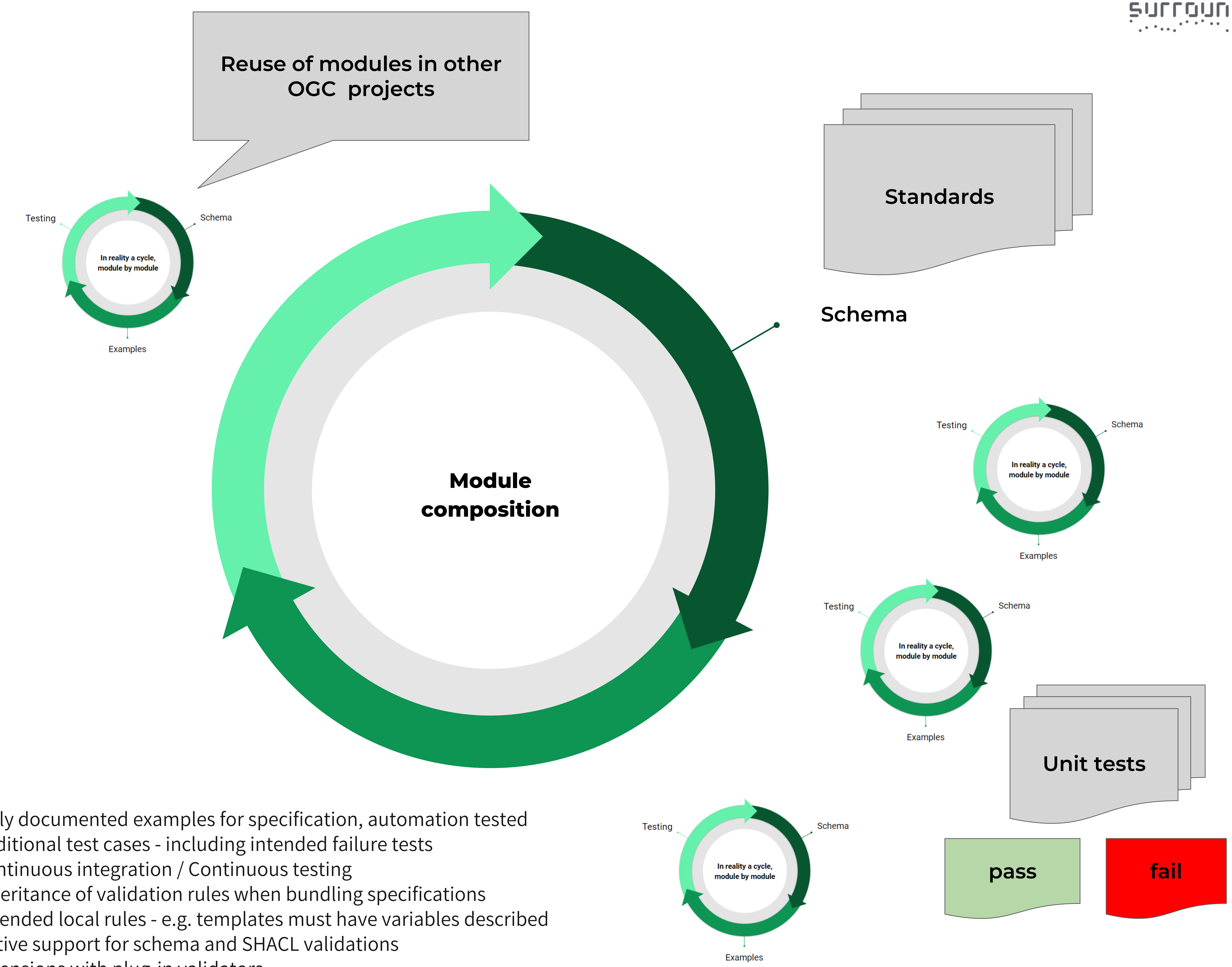
Proprietary model

TESTING

Confidence through testing



Automated testing procedures allows confidence in solutions. Rapid development is supported both by confidence in testing of reusable components as well the means to test new components and entire solutions.


Reuse of modules in other OGC projects
Testing
Schema
In reality a cycle, module by module
Examples
Module composition
Standards
Schema
Testing
Schema
In reality a cycle, module by module
Examples
Testing
Schema
In reality a cycle, module by module
Examples
Unit tests
Testing
Schema
In reality a cycle, module by module
Examples
pass
fail

- Fully documented examples for specification, automation tested
- Additional test cases - including intended failure tests
- Continuous integration / Continuous testing
- Inheritance of validation rules when bundling specifications
- Extended local rules - e.g. templates must have variables described
- Native support for schema and SHACL validations
- Extensions with plug-in validators
- Transformation testing to/from component to related forms

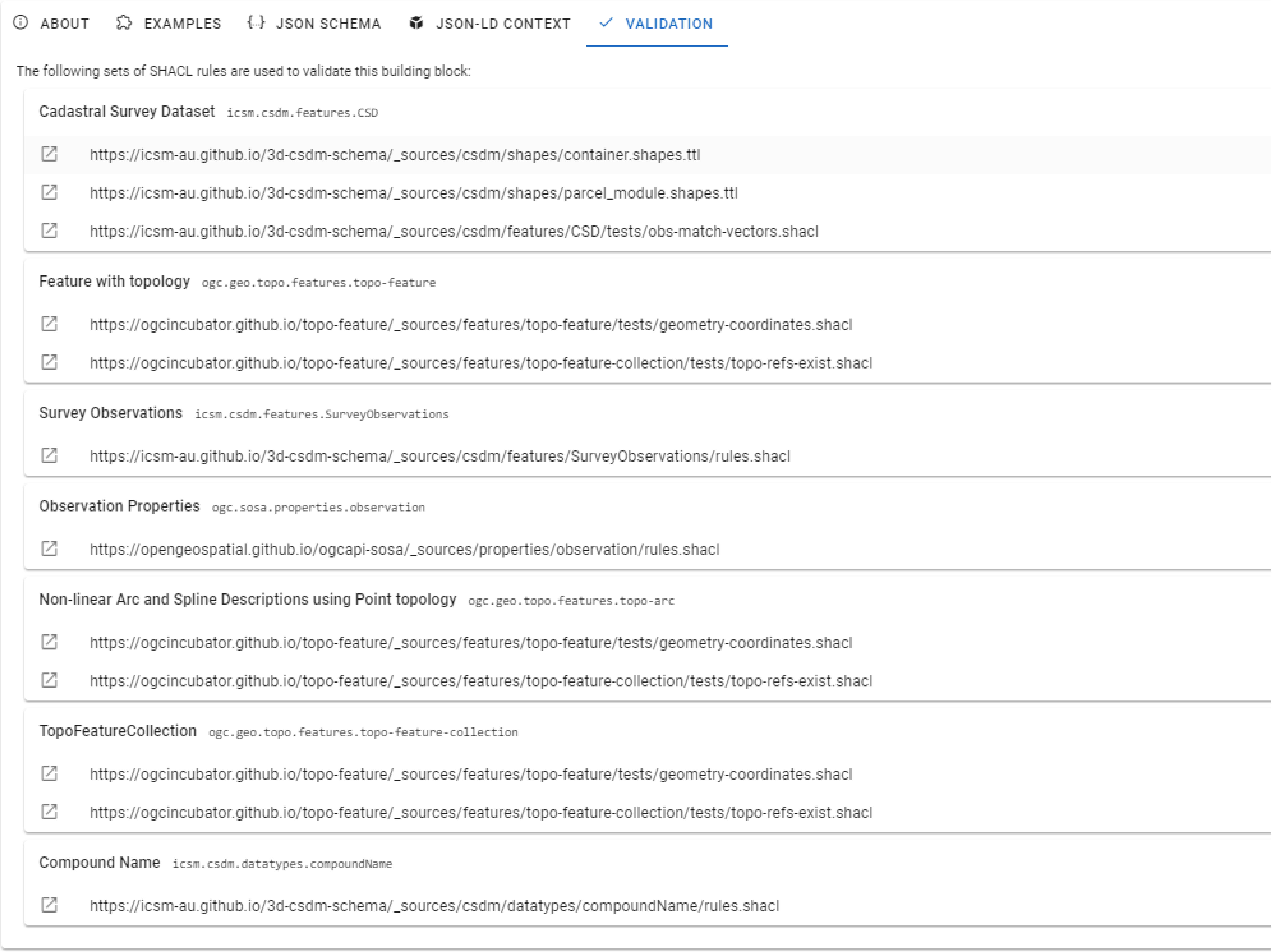
Rules

Complex

Reusable

Automated

(inherited)



The following sets of SHACL rules are used to validate this building block:

Cadastral Survey Dataset `icsm.csdm.features.CSD`

https://icsm-au.github.io/3d-csdm-schema/_sources/csdm/shapes/container.shapes.ttl

https://icsm-au.github.io/3d-csdm-schema/_sources/csdm/shapes/parcel_module.shapes.ttl

https://icsm-au.github.io/3d-csdm-schema/_sources/csdm/features/CSD/tests/obs-match-vectors.shacl

Feature with topology `ogc.geo.topo.features.topo-feature`

https://ogcincubator.github.io/topo-feature/_sources/features/topo-feature/tests/geometry-coordinates.shacl

https://ogcincubator.github.io/topo-feature/_sources/features/topo-feature-collection/tests/topo-refs-exist.shacl

Survey Observations `icsm.csdm.features.SurveyObservations`

https://icsm-au.github.io/3d-csdm-schema/_sources/csdm/features/SurveyObservations/rules.shacl

Observation Properties `ogc.sosa.properties.observation`

https://opengeospatial.github.io/ogcapi-sosa/_sources/properties/observation/rules.shacl

Non-linear Arc and Spline Descriptions using Point topology `ogc.geo.topo.features.topo-arc`

https://ogcincubator.github.io/topo-feature/_sources/features/topo-feature/tests/geometry-coordinates.shacl

https://ogcincubator.github.io/topo-feature/_sources/features/topo-feature-collection/tests/topo-refs-exist.shacl

TopoFeatureCollection `ogc.geo.topo.features.topo-feature-collection`

https://ogcincubator.github.io/topo-feature/_sources/features/topo-feature/tests/geometry-coordinates.shacl

https://ogcincubator.github.io/topo-feature/_sources/features/topo-feature-collection/tests/topo-refs-exist.shacl

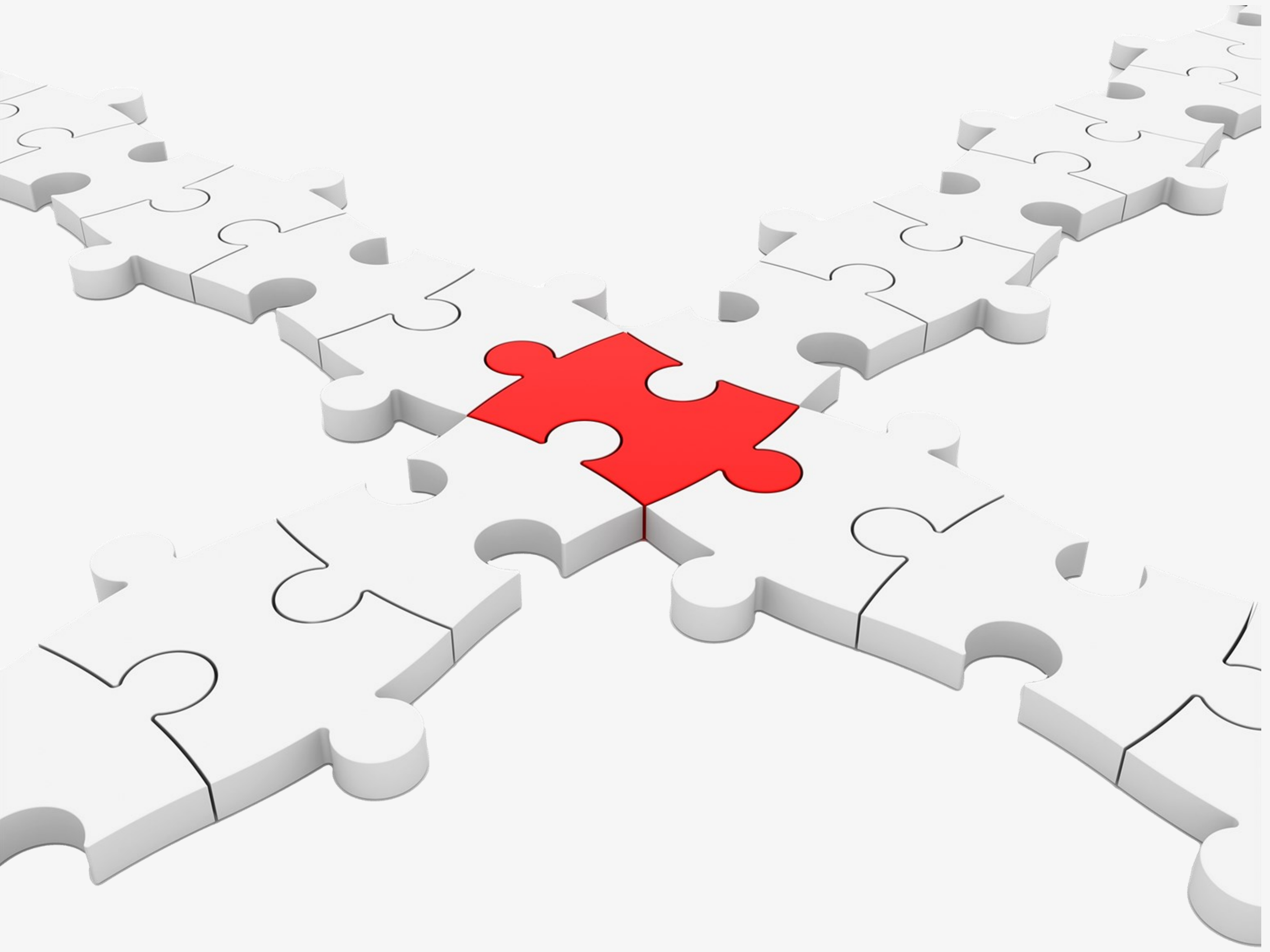
Compound Name `icsm.csdm.datatypes.compoundName`

https://icsm-au.github.io/3d-csdm-schema/_sources/csdm/datatypes/compoundName/rules.shacl

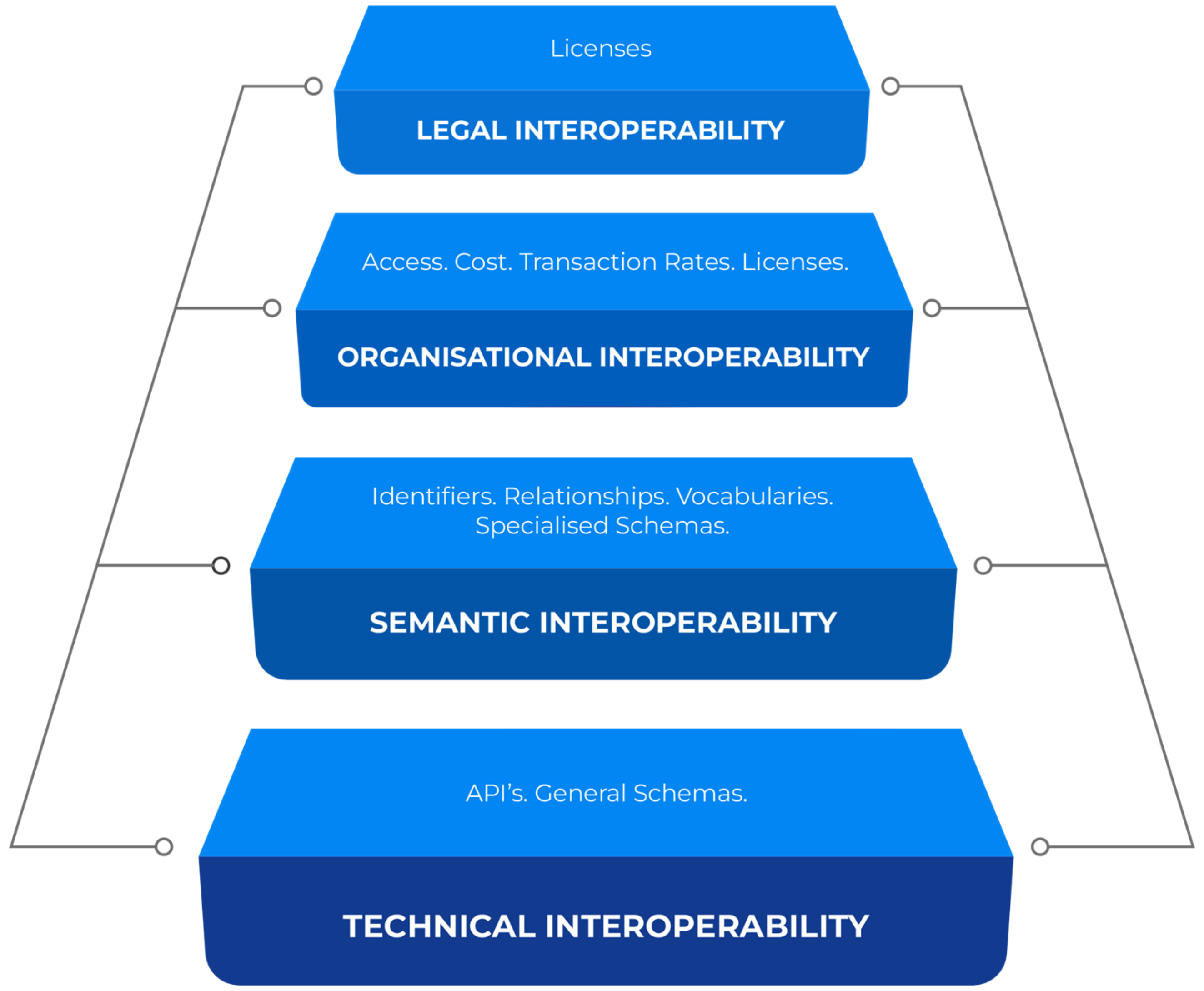
Proprietary model

FUTURE PROOFING

Technologies change, can you?



Building Blocks can be assembled and easily tested, and updated if necessary, to support new technologies, whilst retaining compatibility with legacy implementations.



Licenses

LEGAL INTEROPERABILITY

Access. Cost. Transaction Rates. Licenses.

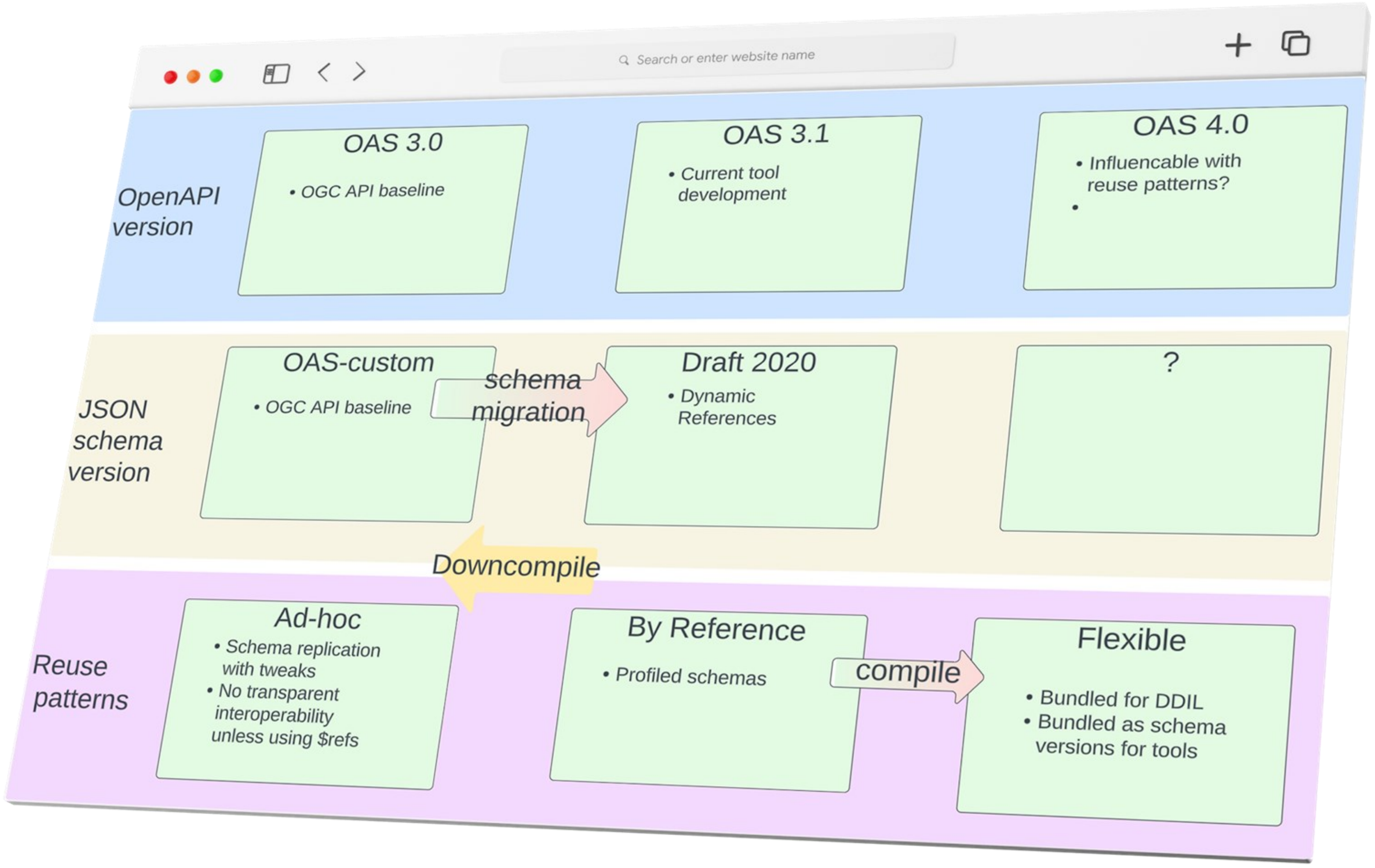
ORGANISATIONAL INTEROPERABILITY

Identifiers. Relationships. Vocabularies. Specialised Schemas.

SEMANTIC INTEROPERABILITY

API's. General Schemas.

TECHNICAL INTEROPERABILITY



OpenAPI version	OAS 3.0 • OGC API baseline	OAS 3.1 • Current tool development	OAS 4.0 • Influencable with reuse patterns? •
JSON schema version	OAS-custom • OGC API baseline	Draft 2020 • Dynamic References	?
Reuse patterns	Ad-hoc • Schema replication with tweaks • No transparent interoperability unless using $refs	By Reference • Profiled schemas	Flexible • Bundled for DDIL • Bundled as schema versions for tools

The technology that has allowed rapid uptake of API and JSON schemas - OpenAPI 3.0 is widely used - but a OpenAPI 3.1 is already being deployed to address several limitations.

Building Blocks support integration and "transpilation" of components using any or all versions of any technology.

The ability to rigorously test each component individually supports rapid development of support for emerging or future technologies.

Quick example

https://ogcincubator.github.io/bblocks-examples/bblock/ogc.bbr.examples.feature.externalSchema

Wraps a FIWARE schema (with JSON-LD context) in a GeoJSON feature, using a BuildingBlock that provides a context for GeoJSON and an context composition pattern