

# BattleVerse



## Extending Semantic Interoperability to Executable Workflows in Digital Twin Systems

Marco Amaro Oliveira

INESC TEC

[marco.a.oliveira@inesctec.pt](mailto:marco.a.oliveira@inesctec.pt)

AIOTI Workshop on Semantic Interoperability for Digital Twins

2026/03/24



# Motivation: Digital Twins of the Earth (DTEs)

**Digital Twin:** digital representation of a target entity with data connections that enable convergence between the physical and digital states at an appropriate rate of synchronisation

*ISO SC41 IoT and Digital Twin – ISO 30173 DIS*

**(Environmental) Digital Twins of the Ocean:** virtual representation of the real ocean and have a two-way connection with it. Observations from the real ocean change and refine the twin; manipulating the twin can highlight regions of the real ocean in need of better or different observations.

*UN DITTO*

**Digital Twins of the Earth:** digital representations of the whole or partial systems of the physical Earth, including, for example, Water/Oceans, Land, atmospheric systems, lifeforms, and human activities and their interactions. Digital Twins (DTs) can be used for monitoring, projecting and delivering actionable information supporting decision makers.

*P3501 - Recommended Practice for the Development of DTEs,  
Submitted to IEEE on 25 Apr 2024,  
PAR approved on 26 Sep 2024.*

... are increasingly capable of answering **core analytical questions:**

- **What past?** → Historical evolution
- **What now?** → Current State
- **What future?** → Prediction
- **What if?** → Scenario Simulations

... which are inherently **local to a given DT**



# Motivation: Beyond Isolated Digital Twins

A challenge in a  
**federated DT ecosystem** is  
enabling:

## What Else?

- *“What other Digital Twins exist that may influence or be influenced by my current scenario?”*

This requires:

- discovery of external models,
- understanding their semantics,
- evaluating their applicability,
- and integrating them dynamically.

→ fundamentally  
a **semantic interoperability**  
**problem,**

not just a technical one.



# From data to workflows

Semantic interoperability must extend from data models to executable workflows

We need:

- structured descriptions of tools and workflows
- machine-readable semantics
- standardised interfaces

Goal:

**Interoperable analytics services in Digital Twins**



# Challenge

Digital Twins must support both human-oriented and machine-oriented interactions.

A human can:

- read documentation;
- understand context;
- decide relevance.

A machines **must do the same autonomously**, which requires:

- **Discover** tools and workflows;
- **Understand** what they are and do:
  - Structure; Purpose;
- **Evaluate** usefulness:
  - Metadata queries;
- Check **usability constraints**:
  - Licensing: access rights: Consent;
- Take **action**:
  - execute, integrate, combine, or ignore

Only possible if tools and workflows (the models) are described in a machine-actionable way.



# Proposed approach

## Extended Best Practice for Application Packaging (APKG)

- Executable workflows (CWL + metadata + containerization)

## Registry

- Search, find and access layer

## APKG Metadata Application Profile (APKG-MAP)

- Semantic description layer

Together: Semantic interoperability

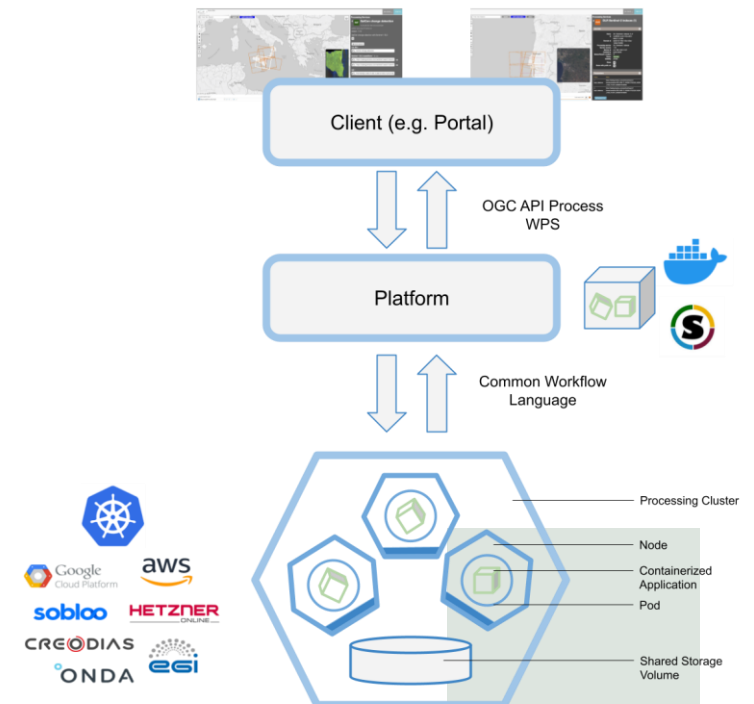


# Application Packaging Best Practice

An Application Package is a comprehensive collection of software, resources, and specifications bundled together to distribute and execute specific data processing workflows.

- a text document that describes the input and output interface of the Application and the orchestration of its command-line tools;
- an optional container image that bundles the tools and their dependencies.
- guarantees the automation, scalability, reusability, and portability of the Application while also being workflow-engine and vendor-neutral.

New OGC Best Practice for EO Application Packages  Open Geospatial Consortium



Key Properties:

Modularity – Portability – Reproducibility – Platform independence



# Why a Registry is Essential (1)

A **Registry** of *Application Packages* was introduced into the Iliad project's architecture as a **core component** of the service layer.

Not just a catalogue - it is a semantic interoperability enabler.

It allows:

- registration (of tools and workflows);
- global and unique persistent identifiers;
- metadata-based discovery;
- open and free protocol (OGC API Records);
- availability of the tools and workflows;
- versioning;
- provenance;
- compliance;
- machine-readable and machine-interpretable descriptions.

Prototype available at:



<https://iliad-registry.inesctec.pt/>



# Why a Registry is Essential (2)

The **Registry** and **Application Packages** approach is explicitly aligned with FAIR Principles:

- Findable
- Accessible
- Interoperable
- Reusable

But **FAIR** principles alignment is not only about (meta)data

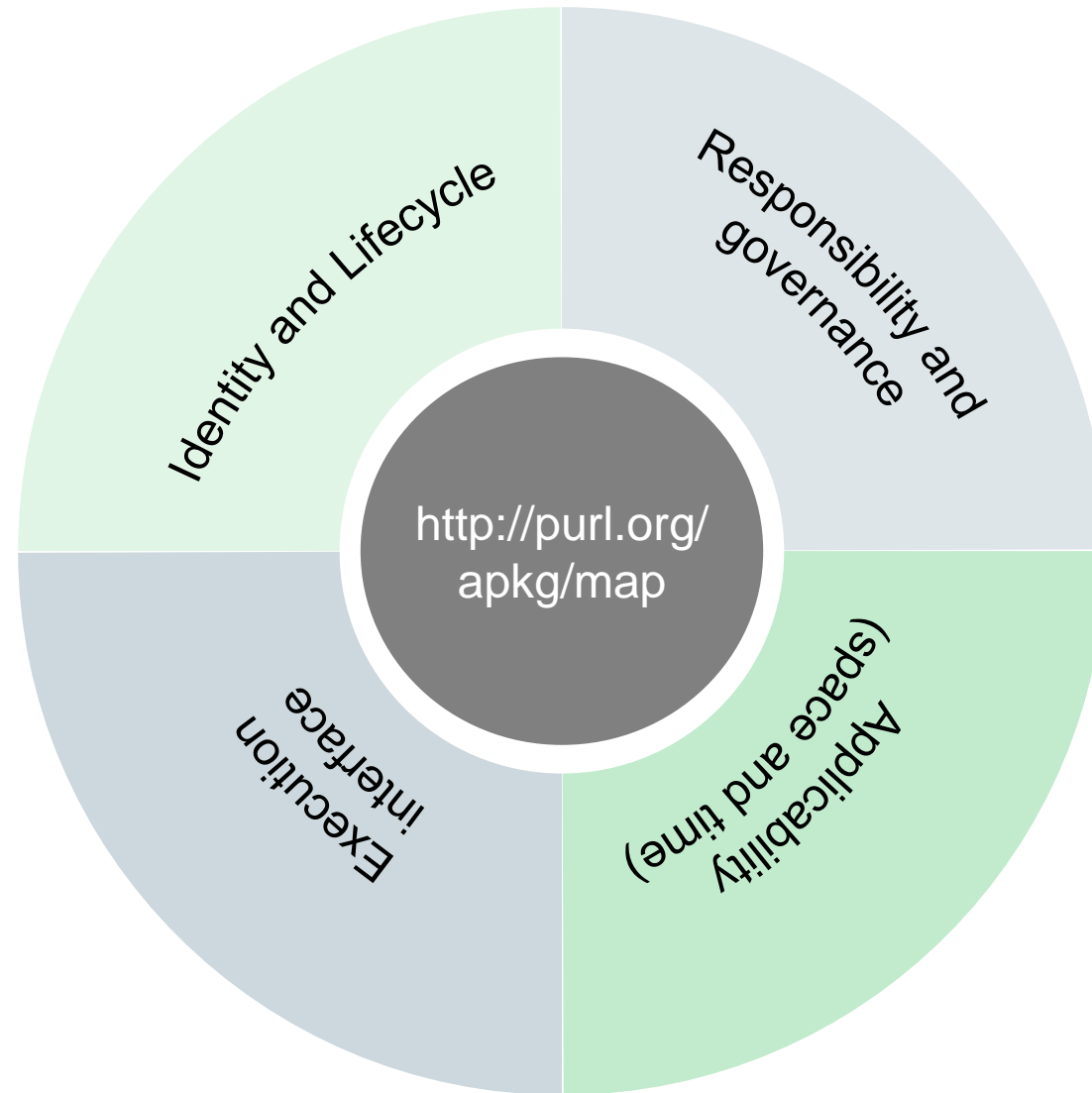
It is also about **machine-actionable digital objects**



# APKG-Metadata Application Profile

A semantic profile for executable applications.

Links the workflow description with software metadata, execution constraints, and domain applicability into one machine-actionable model.

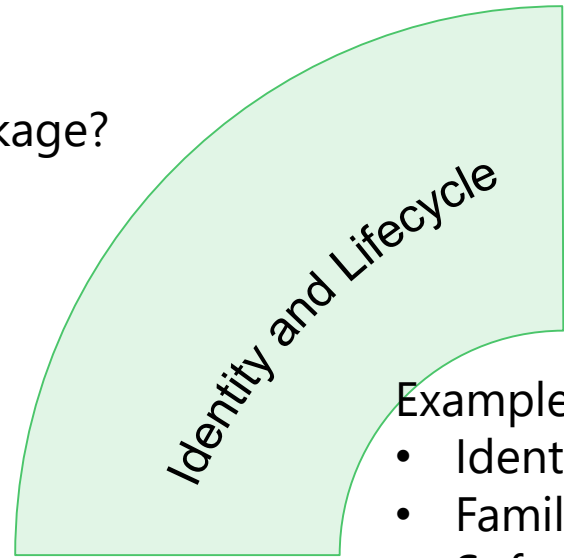


- A semantic model of an executable application
- A validation model that enforces quality and consistency
- A bridge between discovery and execution



# APKG-Metadata Application Profile

What is this application package?  
Which version is it?  
Where did it come from?



Provides stable identity, version lineage, and registry provenance.

Example fields:

- Identifier
- Family
- Software version
- Date created
- Registered by
- Registration date
- Original URL



# APKG-Metadata Application Profile

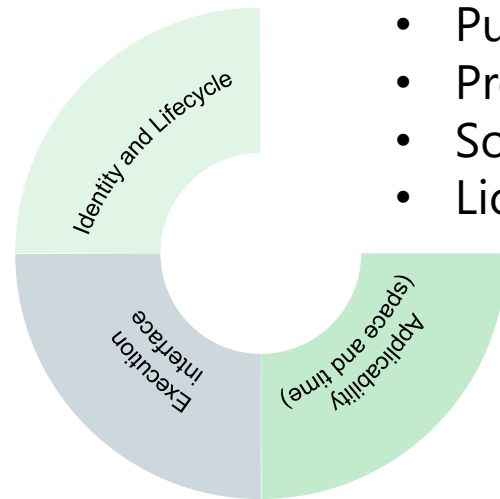
Provides governability: ownership, maintenance, organisational provenance, and licensing are explicit.

Responsibility and governance

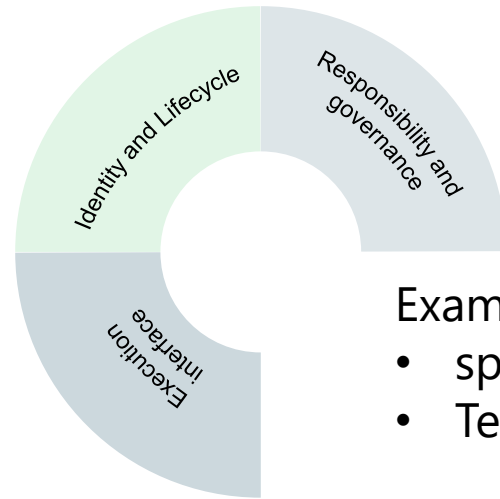
Which organisation is responsible?  
Who maintains it?  
Who made it?

Example fields:

- Author
- Contributor
- Maintainer
- Publisher
- Producer
- Source organization
- License



# APKG-Metadata Application Profile



Example fields:

- spatial coverage
- Temporal coverage

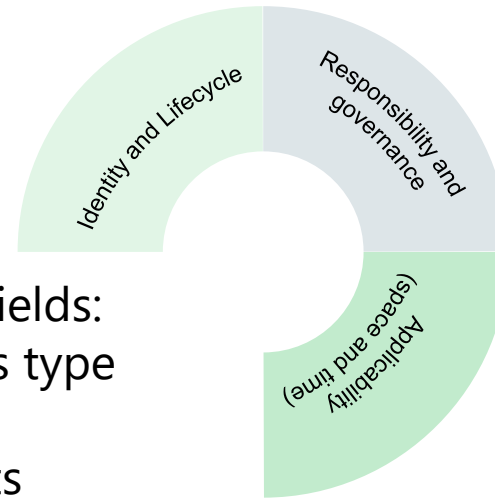
Provides meaningful context.

Applicability  
(space and time)

Where and when  
is this workflow relevant?

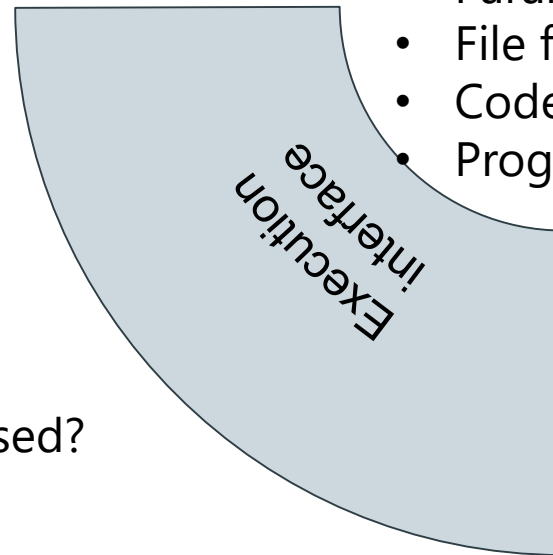


# APKG-Metadata Application Profile



Example fields:

- Process type
- Inputs
- Outputs
- Parameter types
- File format
- Code repository
- Programming language



How can this workflow be used?

Exposes executable interface of a workflow in a machine-actionable way.



## The APKG MAP models the application package itself

A MAP is a combination of already existent RDF vocabularies - Schema.org, Dublin Core Terms, CWL, GeoJSON vocabulary, Time ontology, and DCAT), and a specific created APKG RDF vocabulary.

MAP APKG uses e.g. schema:name, time:hasBeginning, apkg:Parameter, or dct:identifier.

It also encodes cardinalities and controlled values for some terms, such as dct:type.

It is a semantic contract for describing executable workflows so they can be: found, understood, validated, compared, and reused across infrastructures.

Workflows are not just described — they are **machine-actionable**

- Use of SHACL to validate the RDF graphs against the APKG MAP (Fig. 1);
- Every tool and workflow contains a machine-readable, semantically rich description in JSON-LD format (Fig. 2).

<http://purl.org/apkg/map>

```
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix apkg: <http://purl.org/apkg/terms/> .
@prefix schema: <https://schema.org/> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix cwlt: <http://commonwl.org/cwltool#> .
@prefix gj: <http://purl.org/geojson/vocab#> .
@prefix time: <http://www.w3.org/2006/time#> .

#####
# Shape: ApplicationPackage
#####
apkg:ApplicationPackageShape a sh:NodeShape ;
  sh:targetClass apkg:ApplicationPackage ;
  sh:property [
    sh:path schema:name ;
    sh:datatype xsd:string ;
    sh:minCount 1 ; sh:maxCount 1 ;
  ] ;
  sh:property [
    sh:path schema:description ;
    sh:datatype xsd:string ;
    sh:minCount 1 ; sh:maxCount 1 ;
  ] ;
  sh:property [
    sh:path schema:keywords ;
    sh:datatype xsd:string ;
  ] ;
  sh:property [
    sh:path dct:identifier ;
    sh:datatype xsd:string ;
    sh:minCount 1 ; sh:maxCount 1 ;
  ] ;
];
```



Fig. 1 - Shapes Graph

```
@context:
  > 0: { schema: "https://schema.org/", type: "@type", gsp:
  > 1: { xsd: "http://www.w3.org/2001/XMLSchema#", sh: "http://www.w3.org/ns/shacl#" }
  > 2:
    apkg: "http://purl.org/apkg/terms/"
    identifier: "dct:identifier"
    geometry: "apkg:hasGeometry"
    temporal_coverage: "apkg:hasTemporalCoverage,"
    inputs: "apkg:hasInputs"
    outputs: "apkg:hasOutputs"
    label: "apkg:label"
    family: "apkg:family"
    latest: "apkg:latest"
    links: "apkg:links"
    registration_url: "apkg:registrationURL"
  > 3: { schema: "https://schema.org/", keywords: "schema:"
  > 4: { owl: "http://www.w3.org/2002/07/owl#", cwlt: "http://commonwl.org/cwltool#" }
  > 5: { cwlt: "http://commonwl.org/cwltool#", Secrets: "http://purl.org/apkg/terms/" }
  > geometry: { type: "Polygon", coordinates: (1)[-] }
  > type: "apkg:ApplicationPackage"
  > id: "oilspill_pipeline_medslik_156"
  > outputs: [ (-) ]
  > programming_language: "python"
```

Fig. 2 - Application Package RDF Graph example



# Conclusion

This work extends semantic interoperability from data to executable workflows.

Required an extension of the best practice for application packaging.

Combined with registry-based discovery and standardised execution, this approach transforms DTs from isolated systems into

**federated, semantically interoperable DT ecosystems.**

Enables Digital Twins not only to answer:

*What... past, now,, future, if...*

but also:

→ **“What else exists that I should consider?”**



# Next steps

Not everything is solved!

- **Semantic alignment between workflows**
  - semantic matching of inputs/outputs, ontology-driven chaining, automatic pipeline generation
- **A distributed registry architecture**
  - federated discovery across registries (WIP)
- **Execution orchestration**
  - dynamic deployment across cloud/HPC (WIP)
- **Quality assessment**
  - reputation systems, certification/validation workflows, user feedback integration (done)
- **Application to other domains, such as defence.**



# TEAM



**Marco Amaro  
Oliveira**



**Miguel Correia**



**Mafalda Castro**



**Mariana Malta**



# Thank you for your attention!

Presenter	Marco Amaro Oliveira
Organisation	INESC TEC
Email	<a href="mailto:marco.a.oliveira@inesctec.pt">marco.a.oliveira@inesctec.pt</a>
Location	AIOTI Workshop on Semantic Interoperability for Digital Twins
Date	2026/03/24



# Example

**GerMachine**

**Iliad**

**Oil Spill Pilot**

Using MEDSLIK II software predicting the transport and weathering of an oil spill, using a lagrangian representation of the oil slick.

The underlying Digital Twin of the Ocean Core models run CMRL's (<https://cdl.iacm.forth.gr/en/>) models forecasting atmospheric state (WRF), hydrodynamics (NEMO) and waves (Wavewatch III). For further info and what-if scenarios contact [vasmeth@iacm.forth.gr](mailto:vasmeth@iacm.forth.gr) and [antonis@iacm.forth.gr](mailto:antonis@iacm.forth.gr). Relevant application packages will become available at <https://github.com/ILIAD-ocean-twin>.

[Upload Data](#) [Run with EDITO](#)

**Upload Oil Spill Data**

Your upload should be the netCDF file from the model output: \*.nc

Upload NetCDF File

Reference

Start Time (Hr)

End Time (Hr)

[Generate](#)

[Load Previous Oil Spill](#)

Machine search

Map interface showing Greece, Turkey, and Cyprus. A red location pin is placed on the island of Crete. The map includes a search bar and various map controls.