# Model-as-a-Resource

A paradigm for open model sharing

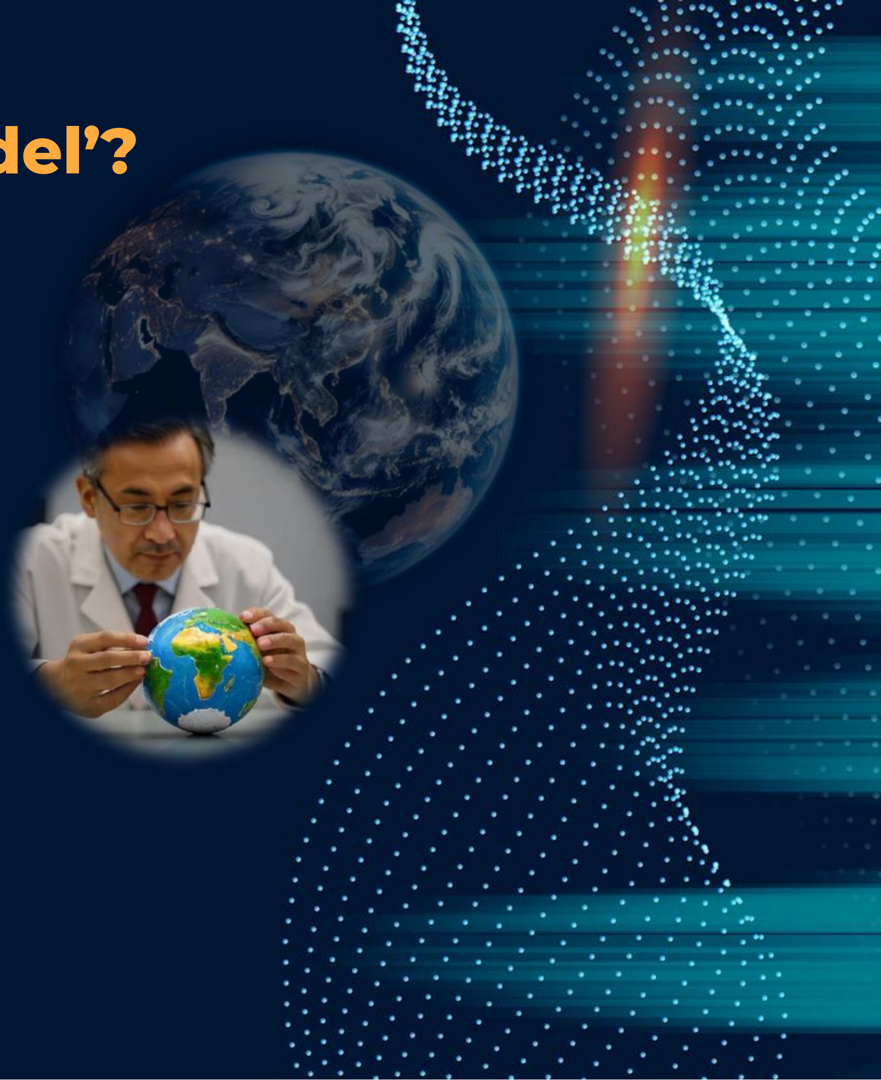
Paolo Mazzetti (CNR)

# Outline

1. What, Why and How of model sharing
2. The Model-as-a-Resource paradigm for the Model Web
3. Towards a MaaR framework
4. Conclusions

## What is a ‘scientific model’?

**An epistemic artifact supporting *surrogative reasoning***



*“…we use one sort of thing as a surrogate in our thinking about another, and so I shall call this surrogative reasoning”*

[C. Swoyer, 1991].

# Scientific model classification

Representational (e.g., simulation models) vs. non-representational (e.g., alternate reality models)

Mechanical vs. mathematical vs. computational

Analogue vs. digital

Input/output semantics vs. operational semantics

Physics-driven vs. data-driven

…

**For the Digital Earth we will focus on digital computational models**

# Why sharing models?

**For science:**

for validation by the scientific community

for documentation

for intercomparison

for re-using in similar or different contexts

…

**For science-based decision-making**

for making the decision transparent

for integrated modelling

…

Open Science

Open Knowledge

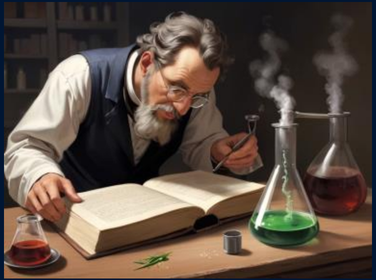
# Why sharing models?
# Open Science/Open Knowledge

## Transparency

Full documentation of the knowledge generation process

## Reproducibility

Possibility to repeat the scientific experiment



## Replicability

Possibility to adapt the knowledge process changing elements (e.g., input data)

## Reusability

Possibility to use the process in different contexts (e.g., integrated modelling)

# The Model Web

**“a dynamic network of computer models that, together, can answer more questions than the individual models operating alone”**

—G. N. Geller and W. Turner, 2007

Environmental Modelling & Software 39 (2012) 214–228

Contents lists available at SciVerse ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

ELSEVIER

Environmental model access and interoperability: The GEO Model Web initiative

Stefano Nativi [a,*], Paolo Mazzetti [a], Gary N. Geller [b]

[a] National Research Council of Italy – Institute of Atmospheric Pollution Research – (CNR-IIA), c/o University of Florence, Piazza Ciardi 25, 59100 Prato, Italy

[b] NASA Ecological Forecasting Program, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA

ARTICLE INFO

Article history:
Received 13 May 2011
Received in revised form
20 February 2012
Accepted 9 March 2012
Available online 7 April 2012

Keywords:
Model Web
Composition as a Service (CaaS)
Model as a Service (MaaS)
GEOSS
Environmental Modelling
Interoperability

ABSTRACT

The Group on Earth Observation (GEO) Model Web initiative utilizes a Model as a Service approach to increase model access and sharing. It relies on gradual, organic growth leading towards dynamic webs of interacting models, analogous to the World Wide Web. The long term vision is for a consultative infrastructure that can help address "what if" and other questions that decision makers and other users have. Four basic principles underlie the Model Web: open access, minimal barriers to entry, service-driven, and scalability; any implementation approach meeting these principles will be a step towards the long term vision. Implementing a Model Web encounters a number of technical challenges, including information modelling, minimizing interoperability agreements, performance, and long term access, each of which has its own implications. For example, a clear information model is essential for accommodating the different resources published in the Model Web (model engines, model services, etc.), and a flexible architecture, capable of integrating different existing distributed computing infrastructures, is required to address the performance requirements. Architectural solutions, in keeping with the Model Web principles, exist for each of these technical challenges. There are also a variety of other key challenges, including difficulties in making models interoperable; calibration and validation; and social, cultural, and institutional constraints. Although the long term vision of a consultative infrastructure is clearly an ambitious goal, even small steps towards that vision provide immediate benefits. A variety of activities are now in progress that are beginning to take those steps.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Climate change, land use change, and changes to the world's aquatic systems are rapid, massive departures from recent historical norms that directly impact many Earth system processes (e.g. Barriopedro et al., 2011; Millennium Ecosystem Assessment, 2005). Interdependencies between these complex processes further complicate the situation (Gaber et al., 2008). Despite the significance to humans in terms of infrastructure, societal operations, and the ecosystem services upon which society depends, our ability to model, predict, and understand the impacts of these changes is very limited (National Research Council, 2002; Service, 2011). This problem is compounded by the limited access that decision makers have to the models, predictions, and related knowledge that do exist. As a result, these decision makers do not have all the information needed to develop appropriate response plans or to evaluate mitigation or adaptation options, despite the urgent need for better approaches to understanding these pending impacts (Gaber et al., 2008).

While this unfortunate lack of predictive capability exists to varying degrees and for a variety of reasons in different Earth science disciplines, it is perhaps particularly true for ecology and related application areas (EEA, 2008). There are several contributing factors, including missing observations to drive the models, incomplete scientific understanding of the processes involved, and limited computing resources. One more factor—limited interaction between existing models—is also important: despite many excellent models, these often work in isolation, without communicating with other models, even those within the same discipline. To highlight this isolation, models and their outputs are often said to be in "silos", a problem frequently cited for data (e.g. Esty and Rushing, 2007; Hey et al., 2009). This results in part because model source code is typically not shared (Barnes, 2010), but also simply because most models were developed independently and were never designed to interoperate with other

☆ Thematic Issue on the Future of Integrated Modeling Science and Technology.
* Corresponding author.

1364-8152/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.envsoft.2012.03.007

https://doi.org/10.1016/j.envsoft.2012.03.007

**“a dynamic modelling infrastructure (Model Web) to serve researchers, managers, policy makers and the general public”**

—S. Nativi and G. N. Geller, GEO Model Web

# How to share models?

## Model-as-a-Tool

A local or networked system that makes possible running models through a GUI

**Transparency**
**Reproducibility**
**Replicability**
**Reusability**

## Model-as-a-Service

A network service that can be invoked to run a model

**Transparency**
**Reproducibility**
**Replicability**
**Reusability**

**MaaT and MaaS fit well for ‘running’ model, but reuse is more than running:**

**Inspection,**
**Comparison,**
**Move to another infrastructure, …**

**A Model is a general resource**

# Service vs. Resource

## Service

“***Work*** *done for others”**

**The provider decides what to offer as a ‘service’**

## Resource

“***Something*** *that can be used to help”**

**The user decides how to use a ‘resource’**

** From Cambridge Dictionary online*

# Architectural styles: SOA vs. ROA

## Service-Oriented Architectures

- Decomposition into distinct units of logic that machines (i.e., servers) expose to other machines (i.e., clients)
- Potentially complex actions offered as a service
- Ancillary services (e.g., registry)
- **Dedicated interfaces**

## Resource-Oriented Architectures

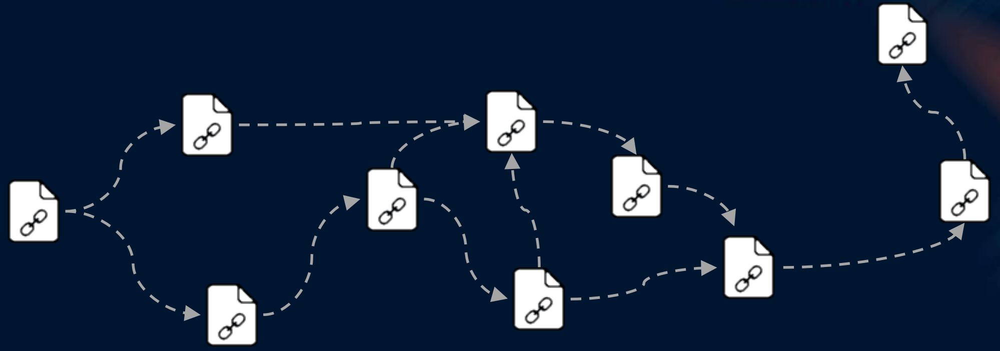
- Resource as items of the information space
- Basic set of actions on resources (Create-Retrieve-Update-Delete)
- Complex actions as a workflow of basic actions
- **Uniform interface**

*"An architectural style is a coordinated set of architectural constraints that restricts the roles/features of architectural elements and the allowed relationships among those elements within any architecture that conforms to that style"*
*[R. Fielding, 2000]*

# Representational State Transfer (REST): the ROA style of the World Wide Web

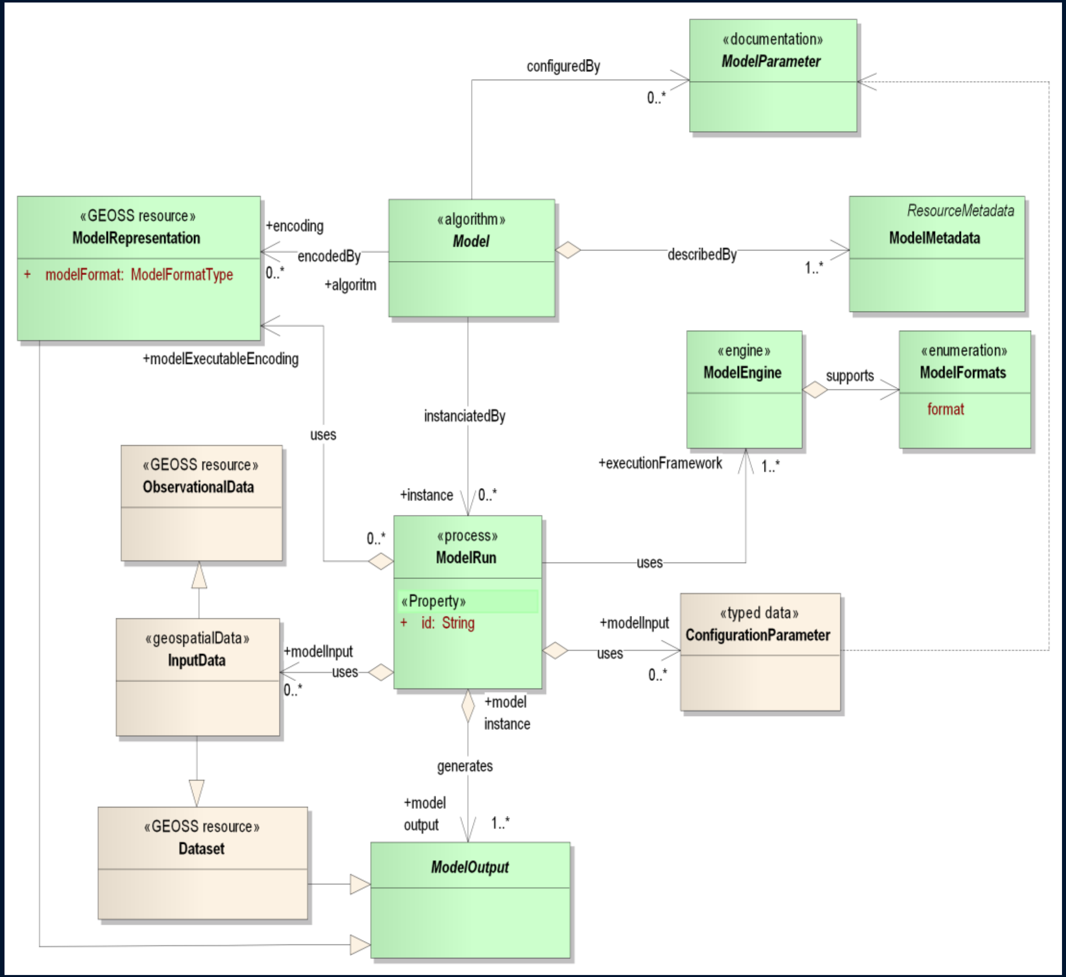
## CONSTRAINTS

- Client-server
- Stateless
- Cache
- Layered system
- Code-on-demand
- **Uniform interface**:
  - identification of resources (→ URI)
  - manipulation of resources through representations
  - self-descriptive messages (→ HTTP,...)
  - **hypermedia as the engine of application state (HATEOAS)** (→ HTML,...)



# How to design a RESTful MaaR framework?

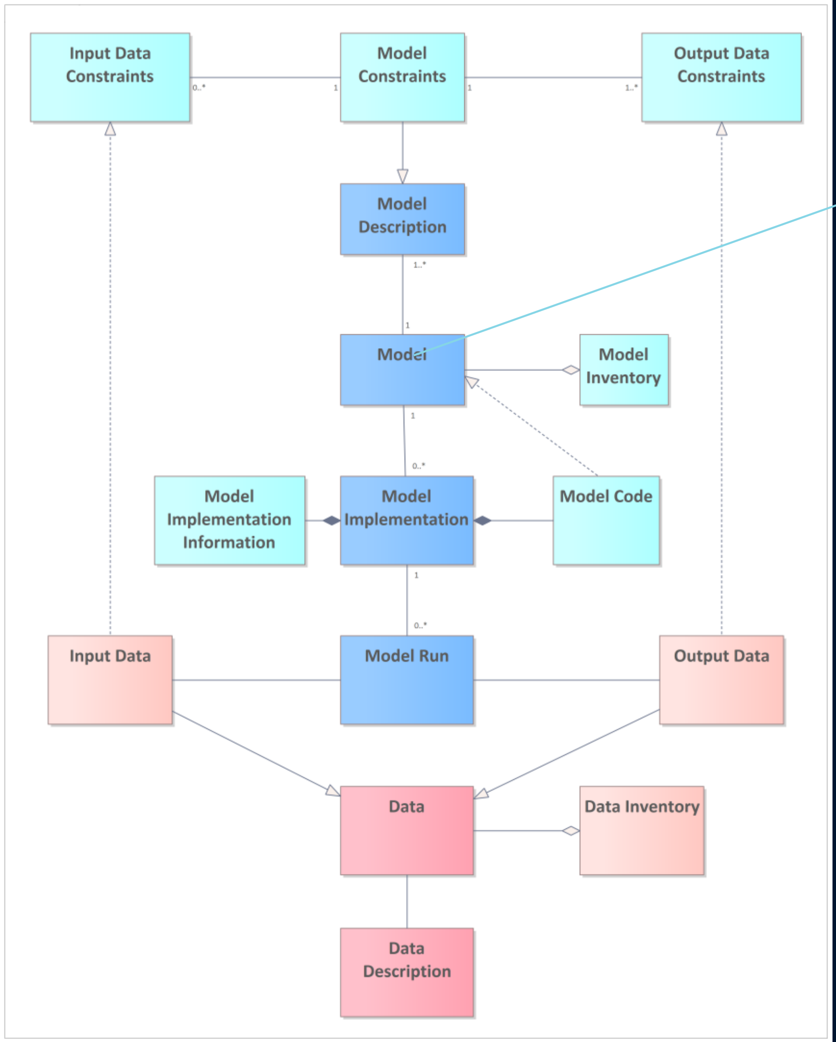
- From Open Knowledge requirements:
  - Identify the logical resources and their necessary representations (e.g., in HTML and/or other formats)
  - Design the functional components respecting the REST constraints (Client-server, Stateless, Cache, Layered system, Code-on-demand, Uniform interface)




# Logical resources:

# The Model Web

Set of resources identified in the original GEO Model Web




*"a Model resource is the algorithm that computationally describes a scientific representational model"*

Computational realization of a logical model
Implemented by a Model Code

# MaaR resources

Set of resources identified for the MaaR core.

Each resource will have its own hypermedia representation (plus others)

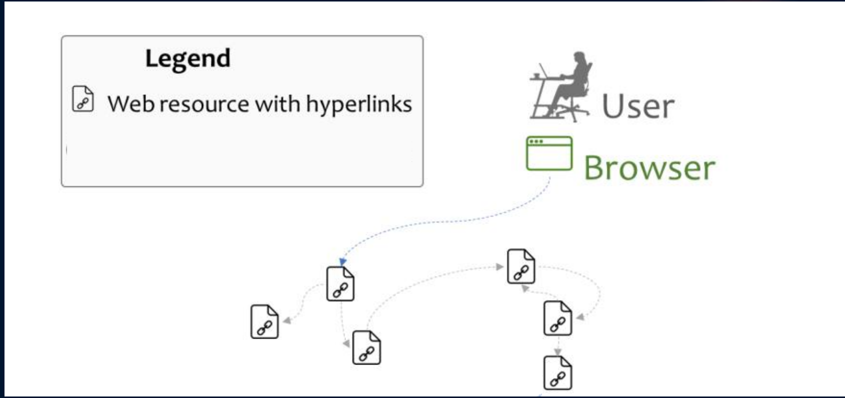
# Logical components

The REST constraints simplify the design of (high-level) logical components:

Clients accessing ‘resources’ through a uniform stateless interface

Clients able to visualize and navigate hypermedia

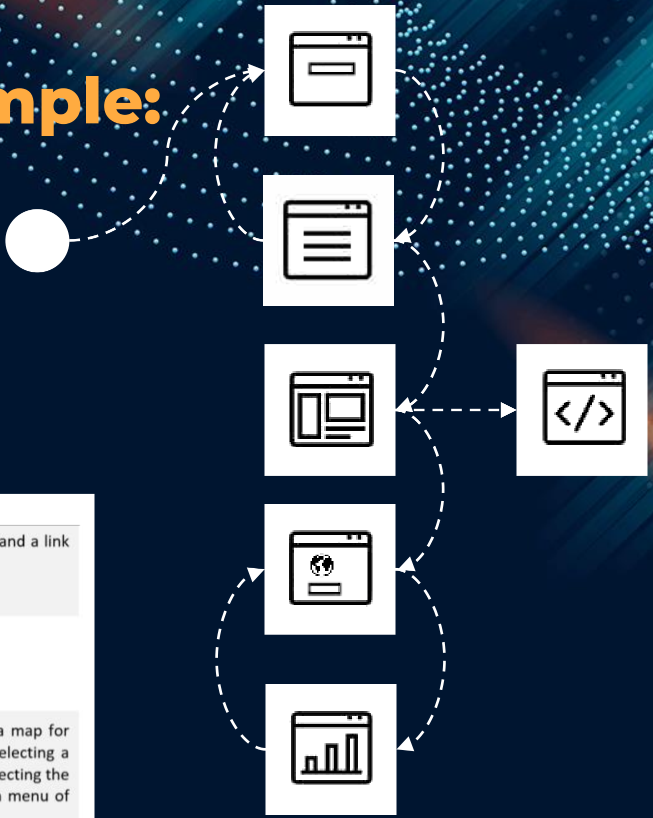
Clients able to run mobile code



# Building applications, an example: reproducibility/reusability

| STEP | USER | SYSTEM |
|---|---|---|
| 1 | The user accesses the *Model Inventory* resource. | The system answers presenting a Model Catalog form with fields for filtered search. |
| 2 | The user fills in the fields and launches the query. | The system answers presenting a form with fields for filtered search and the results of the previous query as a list of links to *Model* resources with a short description. |
| 3 | The user selects the link of interest. | The system answers presenting the basic representation of the selected *Model* resource with a backlink to the *Model Inventory* resource, links to *Model Description* resources, links to *Model Implementation* res res |
| 4 | The user selects one *Model Description* resource. | th |
| 5 | The user goes back and selects one *Model Implementation* resource. | im lar so fo |
| 6 | The user goes to the *Model Form* resource. | sel te in pr |

| STEP | USER | SYSTEM |
|---|---|---|
| 7 | The user selects one of the predefined scenarios (Reproducibility) cited in the *Model Description* previously read and start the model. | The system answers with some information and a link to the generated dataset. |
| 8 | The user downloads the generated datasets and locally verifies that it corresponds to what the *Model Description* says. | |
| 9 | The user goes back to the *Model Form* resource. | The system answers with a form including a map for selecting a geographical area, a calendar for selecting a temporal extent, and a drop-down menu for selecting the input datasets. The form also has a drop-down menu of predefined scenarios. |
| 10 | The user defines a new scenario (Replicability) selecting a new location, time and/or input data and runs the model. | The system answers with some information and a link to the generated dataset. |
| 11 | The user downloads the generated datasets. | |



# Building applications, an example: reproducibility/reusability

| STEP | USER | SYSTEM |
|---|---|---|
| 1 | The user accesses the *Model Inventory* resource. | The system answers presenting a Model Catalog form with fields for filtered search. |
| 2 | The user fills in the fields and launches the query. | The system answers presenting a form with fields for filtered search and the results of the previous query as a list of links to *Model* resources with a short description. |
| 3 | The user selects the link of interest. | The system answers presenting the basic representation of the selected *Model* resource with a backlink to the *Model Inventory* resource, links to *Model Description* resources, links to *Model Implementation* resources and links to previously generated *Model Run* resources. |
| 4 | The user selects one *Model Description* resource. | The system answers with a scientific paper informing that the model produced a significant scientific result. |
| 5 | The user goes back and selects one Model Implementation resource. | The system answers with information about an implementation of the model in Python programming language and links to a Git repository containing the source code, and to a Model Form resource for execution. |
| 6 | The user selects the Git link. | The system directs the user to the Git project landing page. |
| 7 | The user builds the model, prepares a Docker container, moves it on a cloud platform, and runs it. | |

**MaaR enables new reusability scenarios, but…**

***"the user builds the model, prepares a Docker container, moves it on a cloud platform and runs it"***

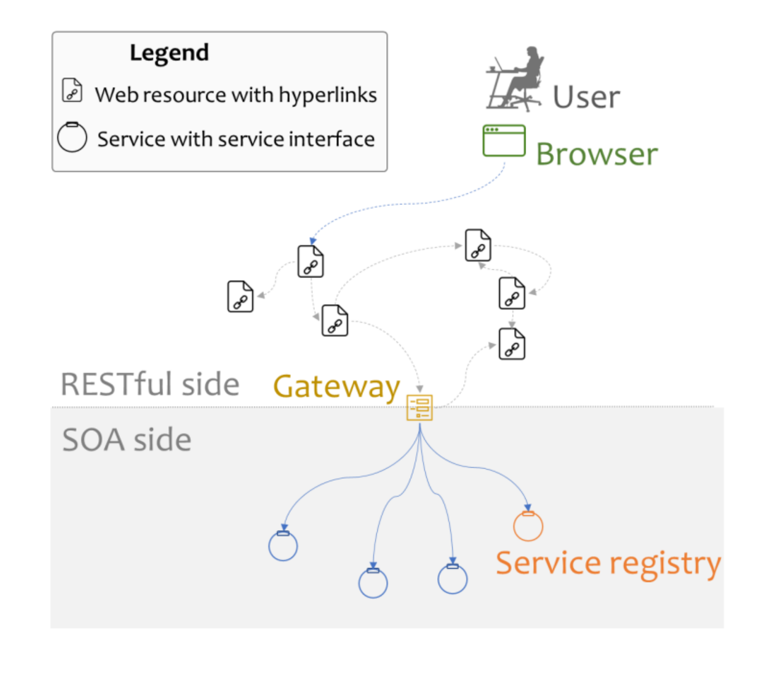
**is too a complex action! We need a *service* for it…**

# Mixed styles vs multi-style architectures

Architecture styles are defined by constraints that guarantee the system properties.

We cannot mix styles because this can violate constraints. (E.g. SOA dedicated interfaces vs. ROA uniform interface.)

But we can adopt different styles in different subsystems using *gateways* to make them dialogue.



# A minimal MaaR Framework

A RESTful subsystem for interaction with resources and building applications through hypermedia navigation

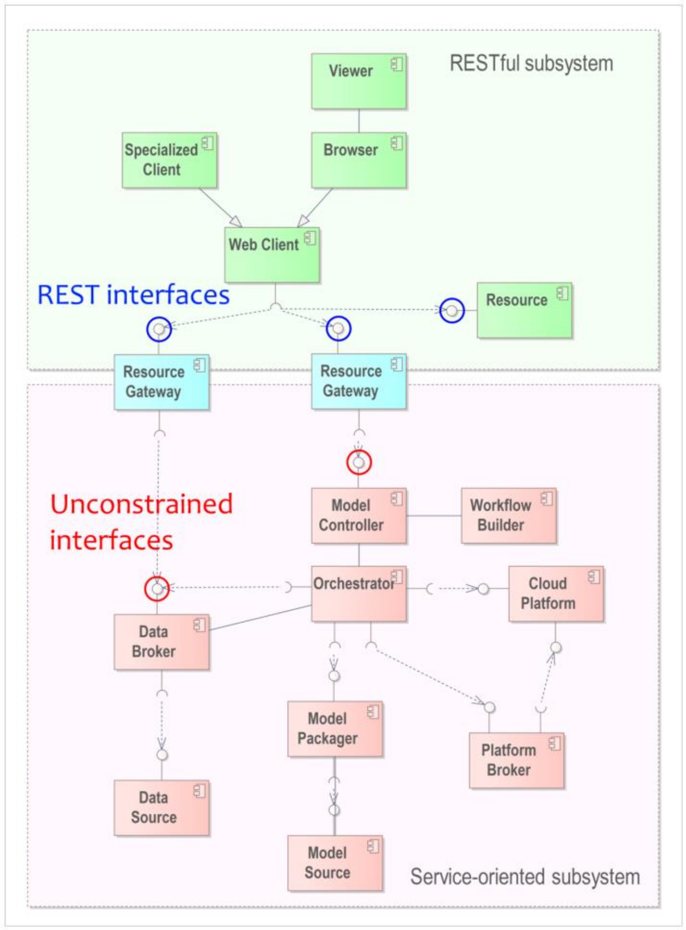
- REST (uniform) interface

A SOA subsystem for complex orchestration

- Uncostrained interfaces

Gateways to expose services through resources.

*"A RESTful API (done right) is just a website for clients with a limited vocabulary"*
*[R. Fielding]*



# Some thoughts on the MaaR framework

| —STRENGHTS | —WEAKNESSES |
| --- | --- |
| Separation of concerns<br>Low entry barrier<br>Extensibility<br>Viability | Multi-style architectures are fragile<br>Initial deployment complexity<br>Data and model availability |
| Multi-disciplinarity<br>Digital transformation | Governance |
| —OPPORTUNITIES | —THREATS |

# THANKS!

paolo.mazzetti@cnr.it

Consiglio Nazionale delle Ricerche

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.

Pictures generated with Copilot and Leonardo.ai

Environmental Modelling and Software 176 (2024) 106002

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft

ELSEVIER

## The model-as-a-resource paradigm for geoscience digital ecosystems

Paolo Mazzetti *, Stefano Nativi

*Istituto sull'Inquinamento Atmosferico, CNR, Via Madonna del Piano 10, 50019, Sesto Fiorentino (FI), Italy*

A R T I C L E I N F O

Handling Editor: Daniel P Ames

*Keywords:*
Scientific models
Interoperability
Digital ecosystems
MaaR
Earth observation

A B S T R A C T

A long-term goal of environmental science and Earth observation is to enable the creation of a "Model Web" of semantically interconnected data and models. Geospatial models are usually exposed on the Web as services accessible through heterogeneous interfaces. However, such services, which represent instances of the paradigm called Model-as-a-Service (MaaS), cannot be easily exploited beyond their original use as defined by the service provider. To overcome this important limitation and better support transparency, reproducibility, replicability and reusability of the model (following the Open Science paradigm), we investigated the adoption of a Model-as-a-Resource (MaaR) approach, in which a model is considered a generic digital resource that, as such, can play different roles in different potential use cases. The proposed MaaR framework can play an important enabling role in the realization of those digital ecosystems that generate environmental knowledge. The main challenges and opportunities are discussed in the manuscript.

**Software and data availability**

The paper presents an architectural framework for model sharing which does not refer specifically to any software solution for implementation. Section §4.5 mentions a set of technologies (software and standards) that could be adopted for the implementation of the proposed framework. All the cited technologies are available as open standards or open source software from their publishers and developers. A couple of software solutions (DAB and VLAB) are developed by authors' research unit and served to implement proofs-of-concept for the proposed architectural framework. More detailed information on these technologies is provided below.

| | |
|---|---|
| **Name of the software** | DAB (Discovery and Access Broker) |
| **Developer** | CNR-IIA |
| **Contact information** | enrico.boldrini@cnr.it |
| **Programming language** | Java |
| **Cost** | Free |
| **Software availability** | https://github.com/ESSI-Lab/DAB |
| **License** | GNU Affero General Public License v3.0 |
| **Name of the software** | VLAB (Virtual Earth Laboratory) |
| **Developer** | CNR-IIA |
| **Contact information** | mattia.santoro@cnr.it |
| **Programming language** | Java |
| **Cost** | Free |
| **Software availability** | https://github.com/ESSI-Lab/DAB |
| **License** | GNU Affero General Public License v3.0 |

### 1. Introduction

#### *1.1. Scientific computational models*

Modelling is an essential activity for modern science. In particular, the so-called *representational* models can emulate the behavior of a well-delimited system providing useful insights on the world that surrounds us. They can come in many different fashions: scale models, analogical models, idealized models, toy models, etc. (Frigg et al., 2020). In the current scientific practice, mathematical models, which aim at providing a mathematical representation of a real system, are the most important ones. In late XIX century and early XX century, mathematics formalization was a major step that allows expressing mathematical proofs as a mechanical procedure or an algorithm. This development suggested that, if physical processes can be represented by mathematical formulas, and mathematical formulas can be encoded as algorithms, then some mechanical instrument might emulate physical processes. This dream became a reality with computers development and the introduction of the computer science. The advent of computers has transformed science and engineering. Based either on physical theories or on big data processing, scientific procedures have been implemented as software code and executed to simulate and predict the behavior of physical systems (Imbert et al., 2017). Then, computational models have

* Corresponding author.
*E-mail address:* paolo.mazzetti@cnr.it (P. Mazzetti).

https://doi.org/10.1016/j.envsoft.2024.106002
Received 28 September 2023; Received in revised form 20 February 2024; Accepted 27 February 2024
Available online 4 March 2024


https://doi.org/10.1016/j.envsoft.2024.106002