European Commission

JRC TECHNICAL REPORT

Destination Earth

Ecosystem Architecture Description

Stefano NATIVI and Max CRAGLIA

2021

Joint Research Centre

EUR 30646 EN

Contact information
Name: Stefano NATIVI
Address: Via E. Fermi 749, 21027 Ispra VA (Italy)
Email: stefano.NATIVI@ec.europa.eu
Tel.: +39 0332 785075

EU Science Hub
https://ec.europa.eu/jrc

JRC124168

EUR 30646 EN

PDF	ISBN 978-92-76-32434-8	ISSN 1831-9424	doi:10.2760/08093

Luxembourg: Publications Office of the European Union, 2021

How to cite this report: Stefano NATIVI and Max CRAGLIA, *Destination Earth: Ecosystem Architecture Description*, EUR 30646 EN, Publications Office of the European Union, Luxembourg, 2021, ISBN 978-92-76-32434-8, doi:10.2760/08093, JRC124168.

Contents

Acknowledgements and Disclaimer

The authors are grateful to their colleagues in DG CNECT for their guidance and to those in the JRC and the policy DGs as well as the stakeholder organizations interviewed, for their help and support. The Authors would like to thank in particular Nicholas Spadaro (at the JRC) and the colleagues at the CNR for their contribution on discussing and testing the multi-cloud configurations and the orchestration framework. Ultimately however the views expressed are those of the authors alone and do not represent the official position of the JRC or the European Commission.

The Authors thank ECMWF, ESA, and EUMETSAT for the valuable comments that contributed to improve the present version of the document.

Scope

The architecture of a system is commonly documented with an architectural description. This document provides an architectural description of the Destination Earth ecosystem to be utilized for the following activities:

a) Express the system and its evolution –including, main services, their decomposition (initial Service Declaration), and service interfaces;
b) Express the persistent characteristics and supporting principles that are foundational to the Destination Earth system and that must be applied to guide acceptable changes;
c) Communicate among the Destination Earth stakeholders;
d) Evaluate and compare existing enterprise system architectures, in a consistent manner;
e) Plan, manage, and execute the future development activities;
f) Verify the implementation compliance with the Destination Earth architectural description;
g) Conceiving, defining, documenting, maintaining, and improving the Destination Earth architecture contributes to the development, operation, and maintenance of the Destination Earth ecosystem from its initial concept until its retirement from use.

Abstract

The concept of Digital Twins of the Earth is at the heart of the recent and ambitious European Commission's initiative, named Destination Earth –also known as DestinE. This initiative was introduced by the EU data strategy, as a concrete action contributing to realize the Common European Green Deal data space, with the aim of using the major potential of data in support of the Green Deal priority actions on climate change, circular economy, zero pollution, biodiversity, deforestation and compliance assurance. DestinE will "bring together European scientific and industrial excellence to develop a very high precision digital model of the Earth". This groundbreaking initiative will offer a digital modelling platform to visualize, monitor and forecast natural and human activity on the planet in support of sustainable development thus supporting Europe's efforts for a better environment as set out in the Green Deal. Destination Earth will contribute to Shaping Europe digital future, according to the EU digital strategy and noticeably its principles on ethics, democracy, fairness and open autonomy.

The present document introduces a set of principles and patterns to be applied by the Destination Earth ecosystem in order to match its stakeholders' and users' needs and requirements. They have been captured and articulated in a set of significant use cases, analysed in a previous study of the JRC. The presented well-defined principles are important to realize a flexible, evolvable, and viable Destination Earth ecosystem. Then, the document provides an architectural description of the ecosystem, applying a set of international standards for the definition of complex system architectures. Specific sections are dedicated to present and discuss the different concerns that characterize the Destination Earth system: the business, information, functional, engineering, and technological viewpoints.

Then, a technological architecture, based on a virtual cloud platform enabled by a multi-cloud environment, is discussed. Finally, the results of a proof-of-concept implementation of such technology architecture are reported and discussed.

1 ARCHITECTURE SPECIFICATION METHODOLOGY

1.1 ISO/IEC/IEEE Architecture Description Model

According to ISO/IEC/IEEE 42010[1], an architecture description (AD) expresses the architecture of a system of interest. A software-intensive system, in keeping with ISO 15288[2], is defined as a system that is man-made and is configured with: hardware, software, data, humans, processes, procedures, and facilities.

We adopted the ISO/IEC/IEEE 42010 conceptual model to express the Destination Earth architecture description –see Figure 1. In particular:

- An architectural description is organized into one or more constituents called architectural views. Each architectural view (or simply, view) addresses some of the architectural concerns held by the stakeholders of that system. An architectural view is an expression of a particular system's architecture with respect to a particular architectural viewpoint.
- An architectural view is composed of one or more architectural models. Architectural models are used to construct architectural views in a modular fashion. Each model can use a different language, notation or model type. Each such architectural model is sanctioned by its associated architectural viewpoint. An architectural model may participate in more than one view to enable sharing of concern-related details across views, without repetition.

Architecting is best understood in a life cycle context, not simply as a single activity at one point in that life cycle. Noticeably, in the case of an evolutionary system (like Destination Earth), the architectural description is used for system development and evaluation, but its uses and development are concurrent.

Given the complex distributed and evolutionary nature of Destination Earth, the different concerns characterizing the diverse stakeholders, and in keeping with ISO/IEC/IEEE/42010, we recognized the need to utilize good practices for specifying the Destination Earth architecture.

We recognized the Open Group's Architecture Framework TOGAF[3] along with the architectural practices and architecture description language specified by ISO RM-ODP[4]. The architectural framework identifies the set of architectural concerns, the generic stakeholders holding these concerns, and the predefined architectural viewpoints that frame these concerns. The architectural practices and description language are used to carry out the architectural models characterizing the views.

1.1.1 TOGAF architecture development Process

TOGAF® Standard is a proven enterprise architecture methodology and framework. It is a standard of The Open Group. The TOGAF architecture development method and cycle, consisting of 6+1 phases, is depicted in Figure 2, along with the description of each phase.

---

[1] ISO/IEC/IEEE 42010 is an International Standard entitled, Systems and software engineering — Architecture description.

[2] ISO 15288 is an International Standard entitled, Systems and software engineering — System life cycle processes.

[3] https://www.opengroup.org/togaf

[4] ISO/IEC 10746-1 Information technology — Open Distributed Processing — Reference model; and ISO/IEC 19793 Information technology — Open Distributed Processing — Use of UML for ODP system specifications.

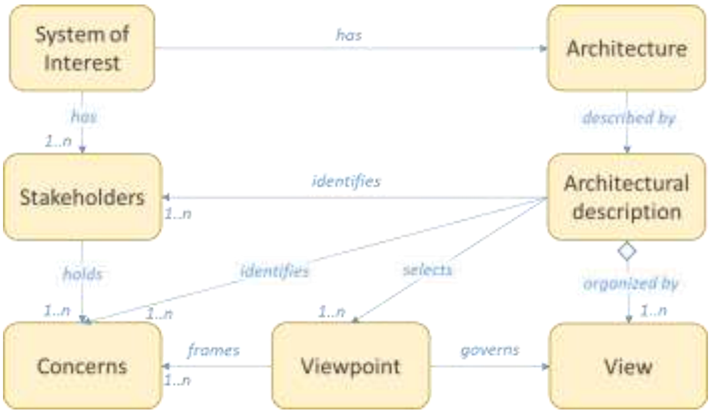


Figure 1. Simplified version of ISO/IEC/IEEE 42010 conceptual model of the Architecture Description realm –the schema makes use of the Unified Modelling Language (UML); for example, the diamond (at the end of an association line) denotes a "part-of relationship".

The iterative application of the TOGAF method allows to define the architectural building blocks (i.e. the packages of functionality meeting the business goals and objectives) which constitute the system architecture description –this happens mainly in the phases: A, B, C, and D. The iterative process of building block definition can be divided into four stages: (a) Business Process level, (b) Technical Functionality and Constraints level, (c) Architectural Model level, and (d) Solution Model level –see Figure 2.

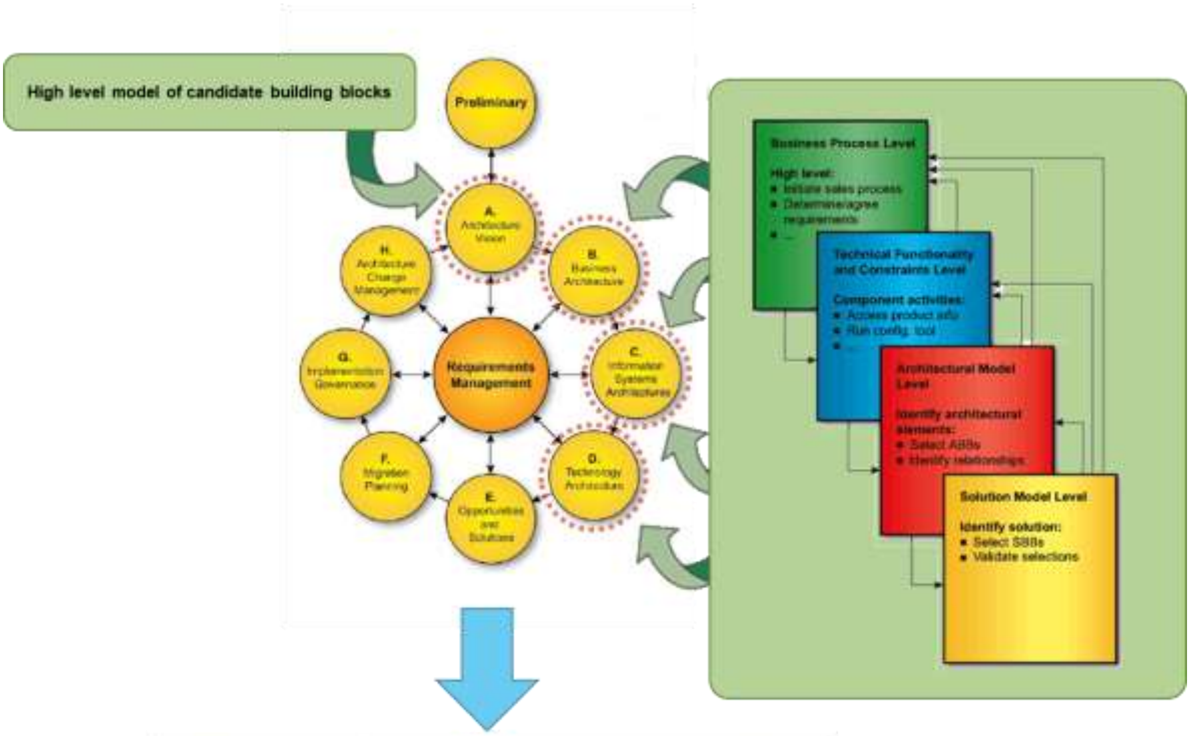
1.1.2 ISO/IEC/ITU Reference Model for Open Distributed Processing (RM-ODP)

The ISO Reference Model of Open Distributed Processing (RM-ODP[5]) is a reference model that provides a co-ordinating framework for the standardization of open distributed processing (ODP). This standard framework supports distribution, interworking, platform and technology independence, and portability. RM-ODP introduces five architectural viewpoints resulting in the specification of five views –showed in Figure 3**Error! Reference source not found.**:

1. **Enterprise** viewpoint deals with the purpose, scope and policies for the system. It describes the business requirements and how to meet them.
2. **Information** viewpoint deals with the semantics of the information and the information processing performed. It describes the information managed by the system and the structure and content type of the supporting data.
3. **Computational** viewpoint deals with the system distribution through functional decomposition into computational objects that interact at the interface level. It describes the functionality provided by the system and its functional decomposition.
4. **Engineering** viewpoint deals with the mechanisms and functions utilized in the system to support distributed interactions between the computational objects. It describes the distribution of processing performed by the system to manage the information and provide the functionality.
5. **Technology** viewpoint deals with the choice of technology of the system. It describes the technologies chosen to provide the processing, functionality and presentation of information.

[5] RM-ODP is also known as ITU-T Rec. X.901-X.904 and ISO/IEC 10746, being a joint effort of the ISO, IEC (International Electrotechnical Commission) and ITU-T (International Telecommunication Union-Telecommunication).

We started implementing the TOGAF process with the Architecture Vision phase –addressing the concerns of different Destination Earth stakeholders (including: end-users, developers, system engineer, and project managers) collected by means of proposed scenarios. To express the architecture building blocks, we utilized the concurrent RM-ODP viewpoints covering three diverse levels of the TOGAF process model, as showed in Figure 4:



Preliminary Phase	Prepare the organization for successful TOGAF architecture projects. Undertake the preparation and initiation activities required to create an Architecture Capability, including the customization of the TOGAF framework, selection of tools, and the definition of Architecture Principles.
Requirements Management	Ensure that every stage of a TOGAF project is based on and validates business requirements. Requirements are identified, stored, and fed into and out of the relevant ADM phases, which dispose of, address, and prioritize requirements.
Phase A: Architecture Vision	Set the scope, constraints, and expectations for a TOGAF project. Create the Architecture Vision. Identify stakeholders. Validate the business context and create the Statement of Architecture Work. Obtain approvals.
Phase B: **Business Architecture** **Phase C:** **Information Systems Architectures** **Phase D:** **Technology Architecture**	Develop architectures in four domains: 1. Business 2. Information Systems – Application 3. Information Systems – Data 4. Technology In each case, develop the Baseline and Target Architecture and analyze gaps.
Phase E: **Opportunities and Solutions**	Perform initial implementation planning and the identification of delivery vehicles for the building blocks identified in the previous phases. Determine whether an incremental approach is required, and if so identify Transition Architectures.
Phase F: **Migration Planning**	Develop detailed Implementation and Migration Plan that addresses how to move from the Baseline to the Target Architecture.
Phase G: **Implementation Governance**	Provide architectural oversight for the implementation. Prepare and issue Architecture Contracts. Ensure that the implementation project conforms to the architecture.
Phase H: **Architecture Change Management**	Provide continual monitoring and a change management process to ensure that the architecture responds to the needs of the enterprise, and maximizes the business value.

Figure 2. the TOGAF architecture development method and the four modeling levels characterizing the TOGAF architecture definition. Copyright © 2020 The Open Group. All Rights Reserved. TOGAF® is a registered trademark of The Open Group. [source: https://www.opengroup.org/togaf]

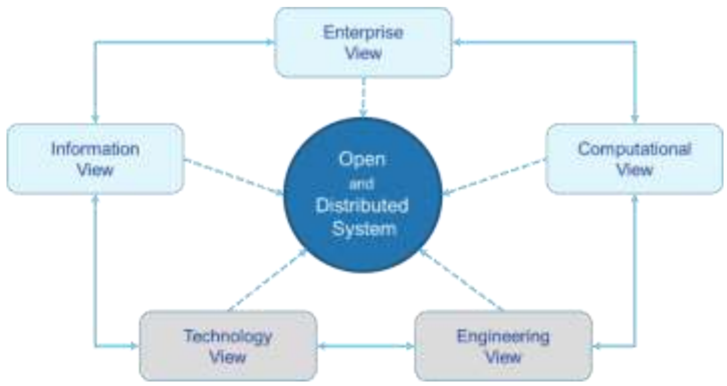


Figure 3. The RM-ODP, ISO/IEC 10746, ITU-T X.901-X.904 viewpoints model

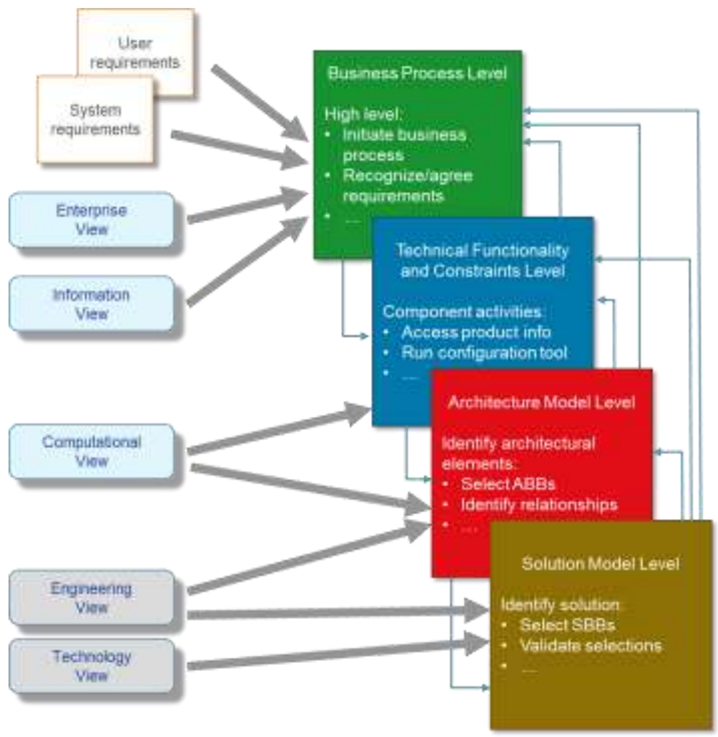


Figure 4. RM-ODP viewpoints utilization for expressing three TOGAF modeling levels: Business Process, Architecture, and Solution (source: https://www.opengroup.org/togaf).

2 BUSINESS PROCESS LEVEL BUILDING BLOCK

2.1 Scope definition

The objective of Destination Earth initiative is to develop a dynamic, interactive, computing and data intensive **"Digital Twin of the Earth"**: a digital, multi-dimensional replica of a physical entity, the Earth (system), which would enable different user groups (public, scientific, private) to interact with vast amounts of natural and socio-economic information.

2.2 Use Scenarios

About 30 use cases were collected and grouped in four Digital Twin policy areas:

1. "Extreme Earth" (disaster risk management from extreme weather-induced natural disasters).
2. Climate adaptation (food and water supply security).
3. Digital oceans around food and energy.
4. Less mature areas (including waste management and health & urban areas nexus).

They are described in the document "Destination Earth: Use Cases Analysis ver. 4.8" (Nativi & Craglia, 2020). Moreover, it was carried out a general survey on the existing international and national projects and initiatives dealing with Digital Twins (Nativi, Delipetrev, & Craglia, Destination Earth: Survey on "Digital Twins" technologies and activities, in the Green Deal area, 2020).

2.3 User requirements

In keeping with its scope and to address the requirements expressed by the collected use cases, Destination Earth must:

i) Offer a **cloud-based core platform**, providing users with access to data (characterized by a highly distributed nature) and infrastructure (consisting of a mixture of purpose-oriented centralize and also distributed components), enabling them to build applications on top of it and to integrate their own data.
ii) Form the **enabling core of a European earth observation, geospatial data and earth systems' applications ecosystem** (Green Deal Data Space).
iii) Develop a number of **thematic Digital Twins (DTs) in priority EU policy areas** (e.g. environment, climate, urban areas, civil protection) giving users easy access to thematic information, services, models, scenarios, forecasts, visualisations etc.
iv) Engage the **scientific community**, the **digital industry**, **governments** and **citizens** in the **modelling of the planet they live on**, based on the latest technological developments.
v) Allow **different types of users at different levels of complexity** to interact and query DTs.
vi) Allow **easy access to observations, models, forecasts,** and **scenarios**. DTs should also support the **visualisation of abstract concepts** and **data types**.

2.4 System requirements

In accordance with the stakeholder's recommendations and needs, to be implemented in an effective and efficient way, Destination Earth must:

i) Act as both **an (horizontal) enabler of an ecosystem/data space** as well as an **advanced (vertical) tool for elaborating and monitoring thematic public policy needs**.

a) If needed, it must be possible to disaggregate the vertical components (i.e. DTs) from the underlying infrastructure and focus on policy use cases (Nativi & Craglia, 2020), only.

ii) Build on the **flexible and convergent use of data**, **infrastructures** (e.g. HPC infrastructures, and federated cloud infrastructures for data distribution/access and hosted processing), **software and AI applications/analytics** supported by a strong horizontal framework.

a) Consider the following main **Data building blocks** –noteworthy, the diverse data considered are available on very different timescales ranging from near-real-time to reprocessed datasets:
   - Copernicus and Earth Observation data (e.g. the full Sentinel archive, Landsat time series, and EMODnet datasets), covering both initial observations as well as derived geo-bio-physical quantities.
   - Environmental data (e.g. climate, land cover, marine/oceans, geological, atmosphere, air quality, winds etc.) from a number of public sources (e.g. the European Environmental Agency, EMODnet, EuroGeoSurveys, meteorological agencies etc.).
   - The basic data layer recognized by the Green Data Space –e.g. social and economic data.
   - Data shared by IoT, social, and economic platforms.
   - Users' data for the development of specific, use-case driven applications.

b) Consider the following main **Infrastructure building blocks**:
   - An open federated system of digital infrastructures, which provide scalable on-demand processing capacity based on a hybrid cloud-HPC model – including HPC centres and EuroHPC, cloud capacities at organisations such as ESA, EUMETSAT and ECMWF, relevant e-Infrastructures (e.g. EPOS) and commercial providers (e.g. Copernicus DIASs).
   - The network capacity provided through GEANT –which already links a number of research and innovation centres across the EU.

c) Consider the following main **Core** (Cloud-based) **Platform services building blocks**:
   - The necessary components enabling federation of distributed cloud systems and services.
   - The necessary (middleware) software elements for extreme-scale computing, data handling and AI as well as workflow management and interfacing with different user communities.
   - The necessary software to enable users to create and manage their own work environments (i.e. self-provisioning services)
   - The necessary software to enable introducing additional data and build higher value-added applications.

d) Consider the following main **User-service-provisioning** (via individual thematic DTs) **building block**
   - The necessary software solutions offering on-demand, compute-intensive workflows for AI and Deep Neural Networks to develop AI-enabled applications for public and private sector (including academia) users, in a progressive way.

A strong user support component will provide guidance and advice both in the horizontal and vertical/thematic aspects.

2.5 Destination Earth Enterprise View

2.5.1 Ecosystem paradigm appliance

Considering the scope, the use scenarios, and the User and System Requirements, the best paradigm to adopt is the Ecosystem one. The ecosystem perspective (stemming from biology) was already successfully used to model complex collaborative and competing social domains –e.g. business and software realms.

The ecosystem holistic paradigm is able to capture the need for diversity of actors and expertise, direct the attention to synergies and connections, and put the focus on the capacity to produce common overall good outcomes, over time. The ecosystem perspective allows to capture the evolutionary systemic process of DTs, ensuring Destination Earth sustainability and thrive.

2.5.2 A thriving Destination Earth ecosystem

To implement an ecosystem that thrives in time, the design and implementation of Destination Earth framework architecture must apply a set of principles, which come along with a set of system and software design patterns.

2.5.2.1 (Digital) Ecosystem principles

Evolvability and Resilience: Destination Earth must be a flexible and dynamic framework, because of the dynamicity of the context where it will operate. Destination Earth ecosystem must be able to evolve and continue operating, in an effectively and efficiently way, in spite of the framework and context changes. Destination Earth must be resilient and survive changes (and even disruptions) that will likely occur in the coming years, due to technological, scientific, organizational, economic, political and societal changes. To achieve that, the architecture must implement the following patterns:

- ***High Flexibility and Modularity*** level to effectively de-couple the enabling infrastructure and platform system (i.e. a system-of-systems) from the thematic digital twin ones. The aim is to implement a high scalable system that implements the elasticity features required by the collected business goals and objectives.
- ***Independence*** from a specific provider, technology, or license –to defend the perceived European values and secure industrial competitiveness, in keeping with the European data and technology openness and autonomy.
- ***Preserve and facilitate the co-evolution of the "digital species"*** populating the digital environment in which the Destination Earth will operate ecosystem. This is important to maximize the Destination Earth resilience.
- ***Equal opportunities*** of access to the infrastructure and affordability for small organizations and across the ICT value chain.
- ***Meta-systemic governance*** of the ecosystem to govern its evolutions, adaptations, mutations, and strains. This will also control the observance of the EU Cybersecurity and Privacy rules.

Emergent-behavior, Exploitation, and Self-sustainability: Destination Earth ecosystem must generate an emergent behavior, i.e. a common value that is more than (or different from) the sum of the single components value. In natural ecosystems, for example, the common value is the survival of the ecosystem itself along with that of all the species. While, for digital/data ecosystems, it is the intelligence/knowledge sharing, enabling its constituents to thrive and generate more intelligence. Destination Earth self-sustainability largely depends on the value of its emergent behavior. To generate an emergent behavior, and sustain it, the Destination Earth architecture must consider the following system patterns:

- ***Belonging versus Autonomy***: the constitutive ecosystem collaborative mechanism is based on the paradox of autonomous and yet cooperative systems: enterprise systems reduce their autonomy and optimization to precise levels and acceptable risks and thus collaborate. In Destination Earth, the enterprise systems must decide/accept about diverse reductions in their degree of autonomy and optimization, which can vary in time –for this reason Destination Earth ecosystem must be highly dynamic –see the *Evolvability and Resilience* principles.
- ***Common versus Enterprise value***: to be effective, the Destination Earth ecosystem is called to preserve the enterprise systems utilities (i.e. allow them to maintain a good degree of autonomy and diversity) enabling a significant ecosystem common value –i.e. emergence value. This, in turn, may influence the degree of autonomy and heterogeneity of the enterprise systems to elicit collaboration and minimize unintended consequences and advance the emergent capabilities valued by the Community. Therefore, the Destination Earth constitutive mechanism must deliver a common value that provides choices for the enterprise systems to fulfill their expected utility in a holistic interaction. In other terns, the ecosystem value complements (e.g. augmenting, optimizing, or diversifying) the value aimed by each enterprise system.
- ***Critical mass*** of services and of users to guarantee a return-of-investments and facilitate the ecosystem resilience.
- ***Cost-effectiveness*** to avoid duplication, through use of shared components (built on infrastructure of organizations that can leverage economies of scale) and use of standard technology –e.g. Internet and Web ones.
- ***Public-Private-Partnership:*** maximizing the value of public investments for the entire Society (e.g. GNSS, Copernicus, GEANT, INSPIRE, Member States facilities etc.), enabling more investment by increasing project financing options, harnessing private sector innovation, facilitate the long-term sustainability, stimulating growth and development in the entire Society.

Minimizing data movement, data replication, and energy consumption: Destination Earth will commonly deal with "Big Data" challenges (Jain, 2016) (Sivarajah, Mustafa Kamal, Irani, & Weerakkody, 2017) requiring smart strategies to manage unusual volume, variety, and quality of data in a fast way. In the past, in the Earth Observation domain, significant platform development initiatives have been carried out (Nativi, et al., 2015), (Nativi, Spadaro, Pogorzelska, & Craglia, under publication), outlining the importance of moving analytical software where data are and not vice versa. Considering the requirements on analytical velocity and the needed computing and storage scalability, it is essential to optimize the nexus of four (often conflicting) minimization factors: data movement minimization, data replication minimization, analytics time lapse minimization, and energy consumption minimization, preserving the distributed data ownership..

Computing continuum (osmotic computing) paradigm: the digital revolution of our society, enabled by technologies such as mobile, edge, fog, and cloud computing, introduced the concept of computing continuum, which supports innovative and computationally demanding applications. By applying such a paradigm, at runtime, applications can choose to execute parts of their logic on different infrastructures that constitute the continuum, with the goal of minimizing latency and energy consumption while maximizing availability (Baresi, Mendonça, Garriga, Guinea, & Quattrocchi, 2019). The same holistic paradigm to support the revolution of Internet of Things

(IoT) services and applications, is also called osmotic computing. This paradigm entails a holistic distributed system abstraction enabling the deployment of lightweight microservices on resource-constrained IoT platforms at the network edge, coupled with more complex microservices running on large-scale datacenters (Villari, Fazio, Dustdar, Rana, & Ranjan, 2016).

Most effective Governance style: Destination Earth will build on existing building blocks (stimulating the effective creation of those elements that are missing), realizing a complex system-of-systems (or super-system). These blocks are characterized by heterogeneous traits and are managed by different organizations. Therefore, the Destination Earth success mostly depends on an appropriate governance of the ecosystem as a whole. In particular, the governance style significantly determines the cybernetic mechanisms that underpin the ecosystem evolution, viability, and self-organization. Finally, the governance style is bound to the ecosystem common value to be implemented to reach the self-sustainability. For Destination Earth DT ecosystem, different governance styles are possible encompassing diverse management and control configurations, from fully distributed to fully-centralized ones. The styles to be considered are, respectively:

a) **Virtual governance**: there is no central management authority and no centrally agreed-upon purpose for the system-of-systems.
b) **Acknowledged governance**: there is a central management organization without coercive power to run the system-of-systems, but constituent systems interact more or less voluntarily to fulfill agreed-upon central purposes.
c) **Collaborative governance**: (like in Directed system-of-systems) there are recognized objectives, a designated system-of-systems manager, and resources allocated for the system-of-systems; however, (like in Acknowledged system-of-systems) the normal operational mode of the constituent systems is not subordinated to the central managed purpose and they retain their independent ownership, objectives, funding, and development and sustainment approaches.
d) **Directed governance**: an integrated system-of-systems is built and managed to fulfill specific purposes by a central management organization. It is centrally managed during long-term operation to continue to fulfill those purposes, as well as any new ones the system owners might wish to address.

In the case of Destination Earth, Collaborative and Acknowledged governance styles seem to be the most appropriate to satisfy the business goals and objectives.

Destination Earth stakeholders outlined the importance of designing a multi-level governance.

In particular, the Destination Earth governance must define and apply the set of invariants that will steer the ecosystem evolution and effectiveness through the many changes occurring in the political, social, cultural and scientific environment where it operates.

2.5.2.2 Ecosystem invariants (i.e. Metasystem and cybernetic mechanisms)

Due to its nature, the Destination Earth DT ecosystem is more subject to changes for both internal reasons (e.g. enterprise system changes, new system addition) and external reasons (e.g. changes in the societal and technological environment where Destination Earth operates). Without any control, those changes could be (in principle) disruptive, modifying Destination Earth in an uncontrolled way that could make impossible for it to pursue its intended objectives. Therefore, Destination Earth ecosystem needs the capability to detect changes and respond to them. Figure 5 shows the dynamic and evolvable environment that will characterize the Destination Earth DT ecosystem.

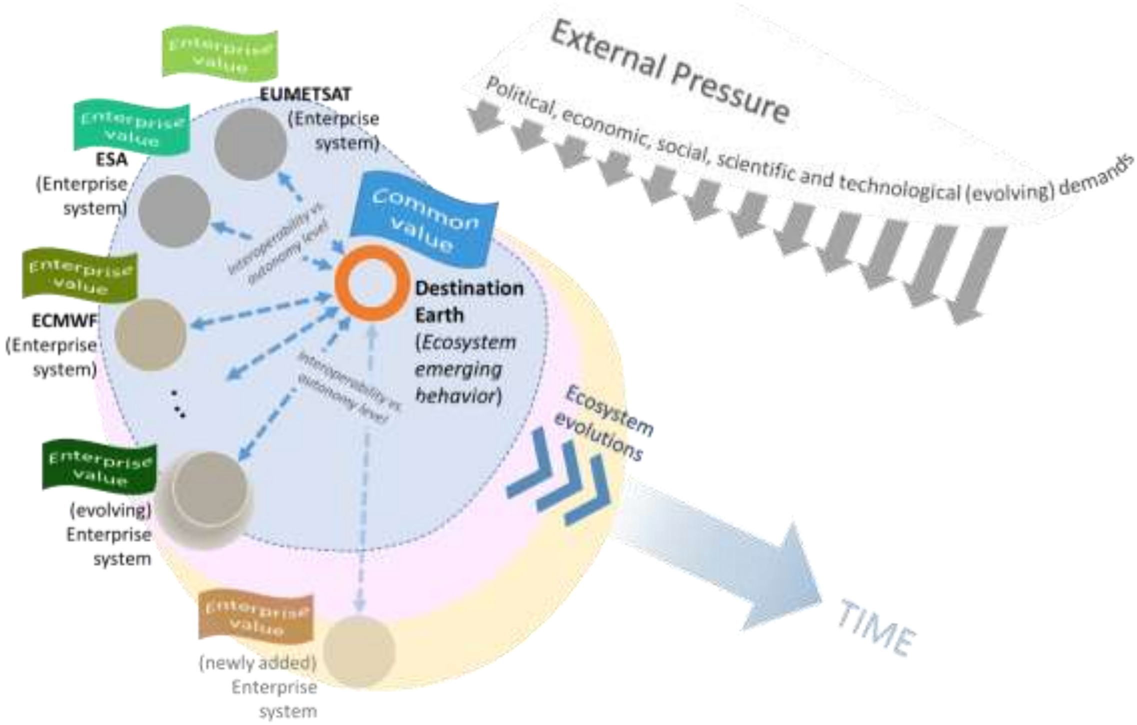


Figure 5. The evolutionary nature of the Destination Earth DT ecosystem, its components, and the context in which they operate

The strategy to make Destination Earth system stable consists in recognizing as invariant the process of change, rather than the things that change –see *viable* systems. Therefore, Destination Earth must be seen as a collection of parts that are dynamically related and such dynamicity must be formalized and governed. This is achieved through communication and control functions. Therefore, to survive, Destination Earth must implement a cybernetic[6] system, including components that implement the required communication and control functionalities. The structure of the controlling components varies according to the selected system typology, ranging from the mere expression of emerging properties, as in fully distributed systems (e.g. Virtual system-of-systems), to a coercive governing body, as in fully centralized systems (e.g. Directed system-of-systems).

The communication and control structures constitute a system on their own, which stands above the controlled Destination Earth system, i.e. a metasystem implemented through a cybernetic mechanism –as depicted in Figure 6. The metasystem works on a different level (second order level) of the controlled Destination Earth system (first order level), dealing with meta elements and items (e.g. providers, consumers, harmonizers, orchestrators, aggregators, governing boards, interoperability standards, etc.) and communicating through a governance language (i.e. a metalanguage). Noticeably, the metasystem definition creates a higher level of organization (i.e. the metalevel) that is in relation

[6] Cybernetics, from the Greek word κυβερνητική (kybernetike), meaning "governance" referred to steering ships, which is etymologically related to “govern” through Latin *gubernare*. (T. F. Hoad (ed.), “The Concise Oxford Dictionary of English Etymology” , 1996)

to the level of the organizations of the enterprise systems being integrated. In summary, the Destination Earth metasystem is the set of mechanisms that creates, controls, and enables the evolution of the ecosystem to reach its given goal –making it an actual cybernetic system. The metasystem assures the system-of-systems invariance that is considered essential for its sustainability and evolution.

Figure 6 shows the main architectural artefacts, as well as their relationships, which must be specified for designing the enterprise view of the Destination Earth ecosystem. These artefacts are:

a) **Metasystem/metalevel schema** –including the ecosystem common utilities.
b) **Governance style and Cybernetic mechanisms** –to apply the metasystem rules.
c) **System-of-systems architecture topology** –largely depending from the governance style, the cybernetic mechanisms, and the common utilities.

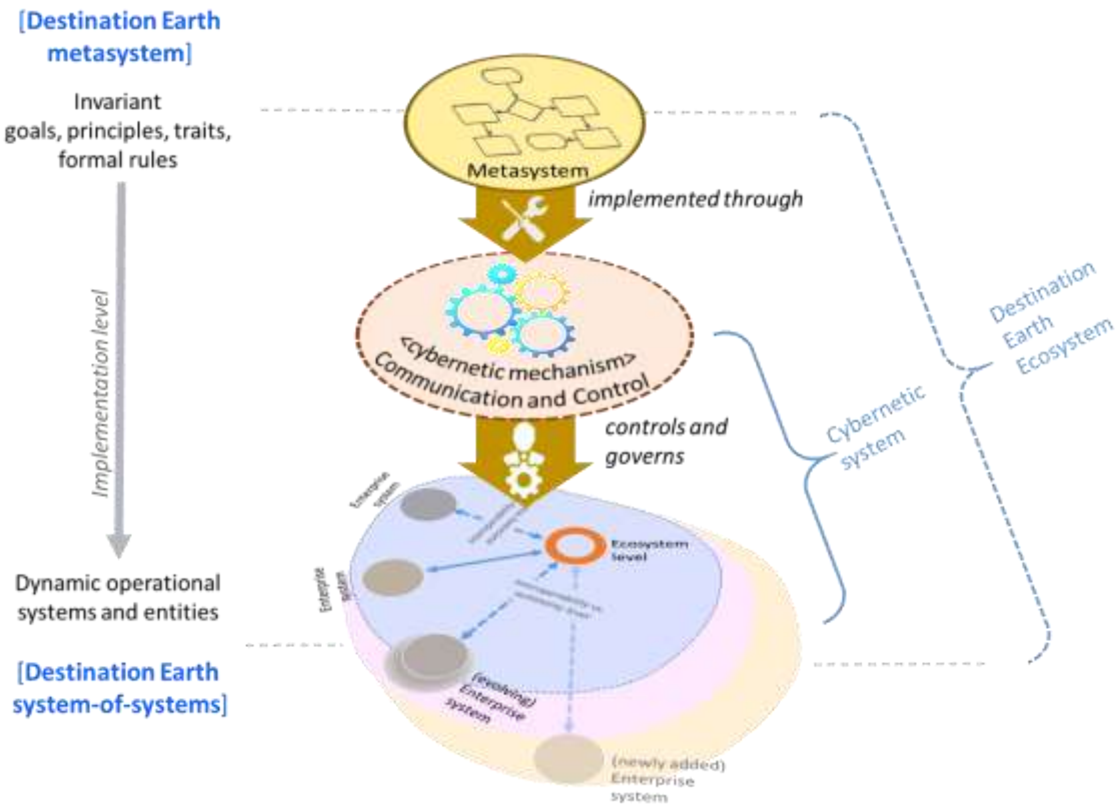


Figure 6. The main design artifacts to be realized for implementing the Destination Earth ecosystem

2.6 Destination Earth Information View

2.6.1 Shared digital entities

Destination Earth ecosystem must share a set of digital entities and related services. These entities and their relationships may be grouped in three strands (as depicted in Figure 7):

a) *Policy/Business* entities;
b) *Platform/software* entities;
c) *Infrastructure* entities.

The following conceptual digital entities will be managed and shared in the framework of the Destination Earth ecosystem:

- **Digital Twins (DTs)**, which are the DTs introduced by the many Destination Earth use cases, contributing to the recognized policy areas –i.e. "Extreme Earth" (disaster risk management from extreme weather-induced natural disasters), Climate adaptation (food and water supply security), Digital oceans around food and energy, and other less mature areas (including waste management and health & urban areas nexus) (Nativi & Craglia, 2020). These abstract entities belong to the *policy/business strand*.
- **Digital Threads,** which are the "*the flow of data fueling the digital insights behind customer-centric experiences*" (Accenture Consulting, 2017). A DT can be defined as a representation of an entity/system, mimicking a given (natural or artificial) process; in this context, a digital thread can be defined as a record of the entity/system lifetime, from its design to its cancelation (Miskinis, 2018). In Destination Earth, a digital thread is crucial to conceive, implement, (re-)use and assess a DT. DT and Digital Thread concepts have been around for decades, but only with the advent of recent digital technologies (e.g. IoT, AI, Big Data) they are effectively implemented, accessible, and usable in the diverse sectors of society (White, 2020). DTs and digital threads play a fundamental role for the digital transformation of our society. A digital thread abstract model and some important aspects covered by a digital thread, are showed in Figure 8. Digital thread conceptual entities (commonly implemented by artifacts like workflows and services orchestration processes) may belong either to the *policy/business strand* or to the *platform/software strand*.
- **Digital/Virtual Resources**, including:
  - **datasets**, along with relevant metadata schemas and the associated data streams;
  - **process-based analytical software**, along with relevant scientific algorithms, models, and workflows;
  - **Learning-based AI analytical software**, along with relevant neural network models and configurations, and workflows.
  - In a future and evolved Destination Earth system, it will be possible that complex DTs may result in a composition of two or more elementary DT instances. In this case, **DT** will be **managed as another kind of digital resources**. These Digital/Virtual Resources belong to the *platform/software strand*.
- **Virtual Network Services**, which provide functions deployed on cloud-based virtual machines (VMs) in the hosted network services environment, in the public cloud or premise-based VMs, subject to availability. An important role of them is to simplify the ecosystem network by bringing together distant and disparate resources in a more efficient way. In the Destination Earth computational view, (in addition to the traditional virtual network functionalities: routing, firewalling, and WAN acceleration) they are used to implement multiple combinations of network functions and/or multiple vendor services at multiple remote and cloud locations, for the automation and orchestration optimizing service provisioning times, and finally for implementing rapid service scaling. Virtual Network Services may belong either to the *platform/software strand* or to the *infrastructure strand*.
- **HPC and Cloud nodes**, which are physical or virtual machines providing compute, storage and networking services. They are the key entities to implement a multi-cloud approach. These entities belong to the *infrastructure strand*.

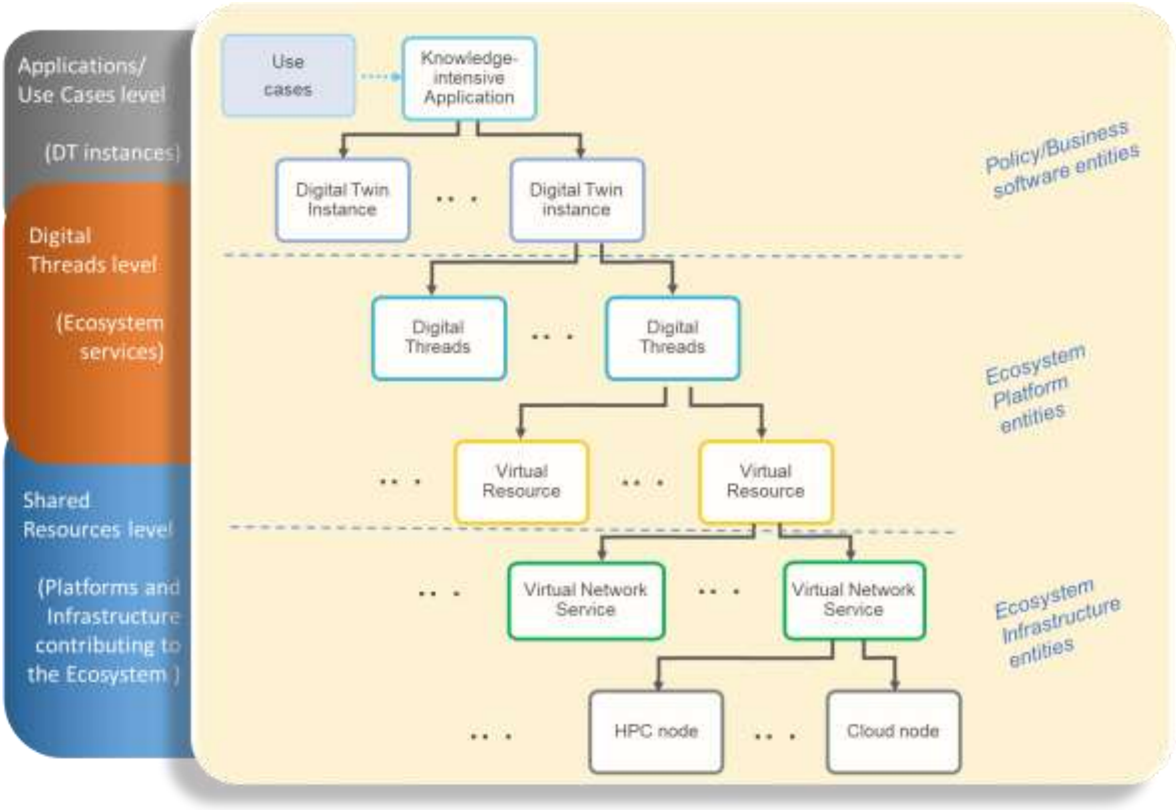


Figure 7. Destination Earth DT ecosystem high-level and abstract entities along with their simplified relationships.

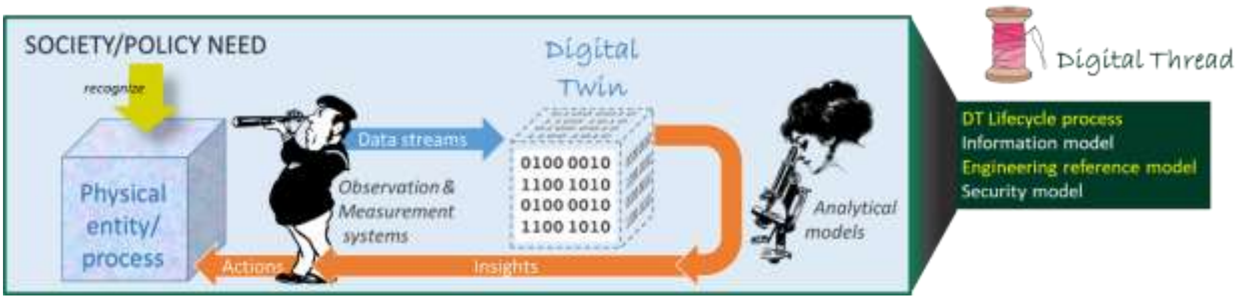


Figure 8. Digital Thread conceptual model and the aspects to be covered

3 ARCHITECTURAL MODEL LEVEL BUILDING BLOCK

3.1 Destination Earth Computational View

3.1.1 General architecture

To address some important requirements, previously defined, the Destination Earth ecosystem is composed of three different computational tiers –see Figure 9 from the bottom up.

- **The horizontal infrastructure and enabling platform**: a common support environment (i.e. platform) and an enabling infrastructure, including the shared digital resource –i.e. data, analytical software and workflow resources along with compute, storage and networking services.
- **The DT layer**: the DT components that make use of the resources and services provided by the enabling platform and the horizontal infrastructure.
- **The (smart) specific use case application**, defined by a specific knowledge community –see the recognized use cases (Nativi & Craglia, 2020).

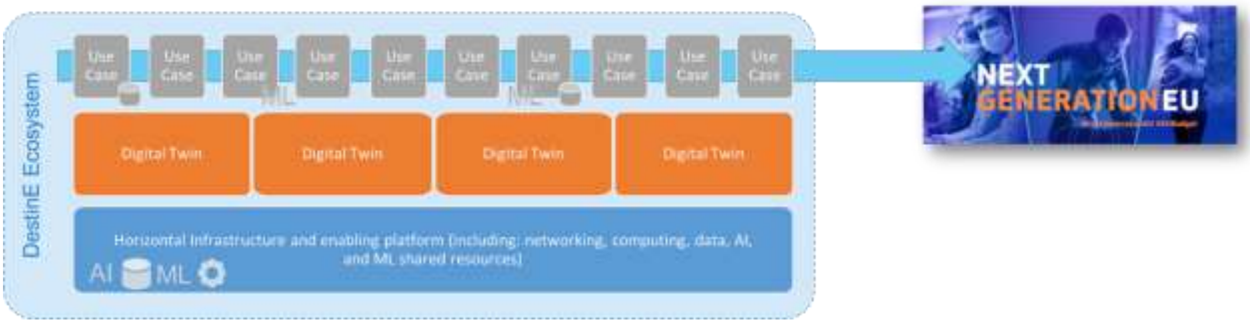


Figure 9. General architecture of the Destination Earth ecosystem

3.1.2 Computational Layers

Destination Earth ecosystem framework will provide the necessary digital services and tools, as well as the infrastructural capacities, to manage (i.e. create, update, access, and utilize) and share the digital objects showed in Figure 7.

To do that, Destination Earth ecosystem framework will implement three main digital layers (digital environments) –as represented in Figure 10:

- Policy/business software layer
- Ecosystem analytics platform layer
- Ecosystem infrastructure layer

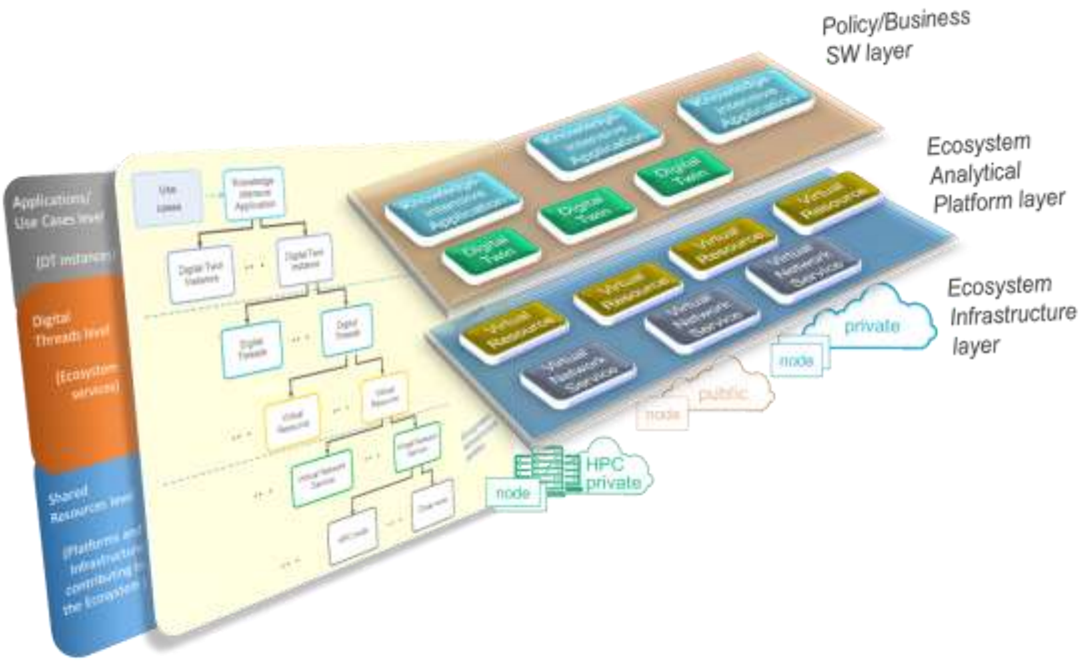


Figure 10. Destination Earth ecosystem digital layers

3.1.3 Functional Systems

Building on the architecture layers and applying the recognized ecosystem and software patterns, Destination Earth must implement five complex functional systems –three of them stem from the specific scope of Destination Earth, while the other two implement the horizontal utilities required by a distributed digital ecosystem. The five functional systems, depicted in Figure 11, are:

- **Knowledge-intensive application development system** to develop applications that address policy needs, using Digital Twins instances. This system builds on the
- **Distributed analytics system** that generates intelligence from the appliance of analytical models (e.g. ML/DL) to data streams. They make use of shared virtual resources (e.g. data, models, tools). This system is enabled by the
- **Multi HPC & Cloud system** that provides the storage/access, network and computational scalable resources.

- **Multi-level security system** that applies to all the three introduced systems and implements the Destination Earth security, privacy, and trust requirements.
- **Cybernetic mechanisms system** that governs and controls the correct behavior of the whole Destination Earth ecosystem –i.e. of the other systems and of their interactions.

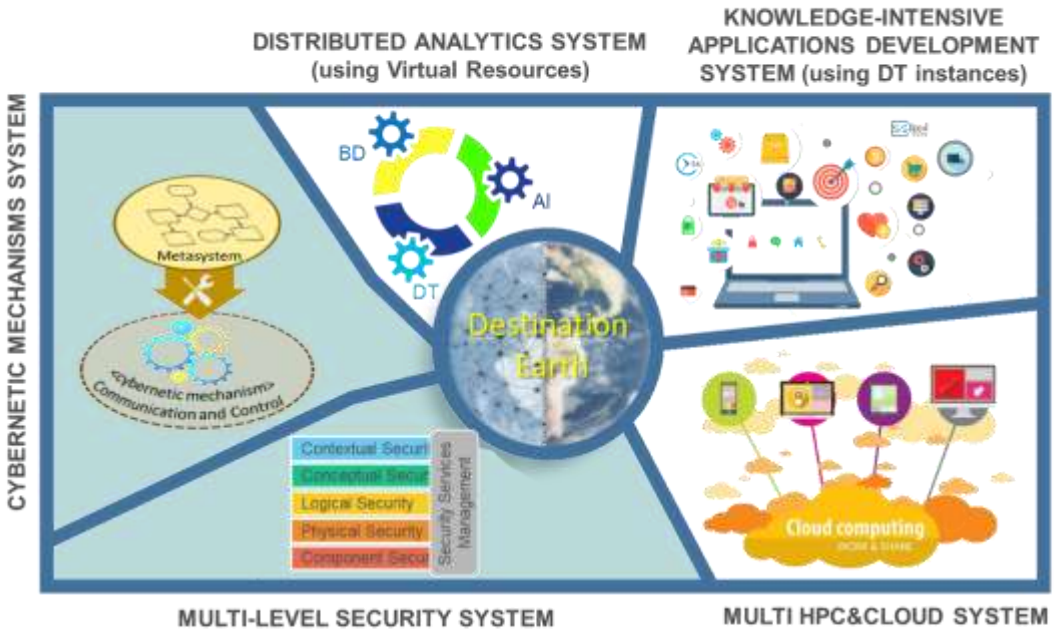


Figure 11. The Destination Earth functional systems to address its objectives (white background) and thrive as an ecosystem (pale blue background)

4 SOLUTION MODEL LEVEL BUILDING BLOCK

4.1 Destination Earth Engineering View

4.1.1 Enterprise system services, shared resources, and distribution mechanisms

Enterprise systems, belonging and contributing to the Destination Earth ecosystem, can be modeled as showed in Figure 12 (a). The model distinguishes between the enterprise system service capacities (i.e. Software-as-a-Service, Platform-as-a-Service, Infrastructure-as-a-Service), and the actual shared resources (i.e. Twin resources, Data Analytics resources, and Infrastructure resources) along with the implemented interoperability mechanisms (i.e. protocols/URIs, APIs, Containers, etc.) to interact with the other components of the distributed environment, which is implemented by the ecosystem. All service layers and resource interoperability mechanisms must implement the security level specified by the ecosystem as a whole.

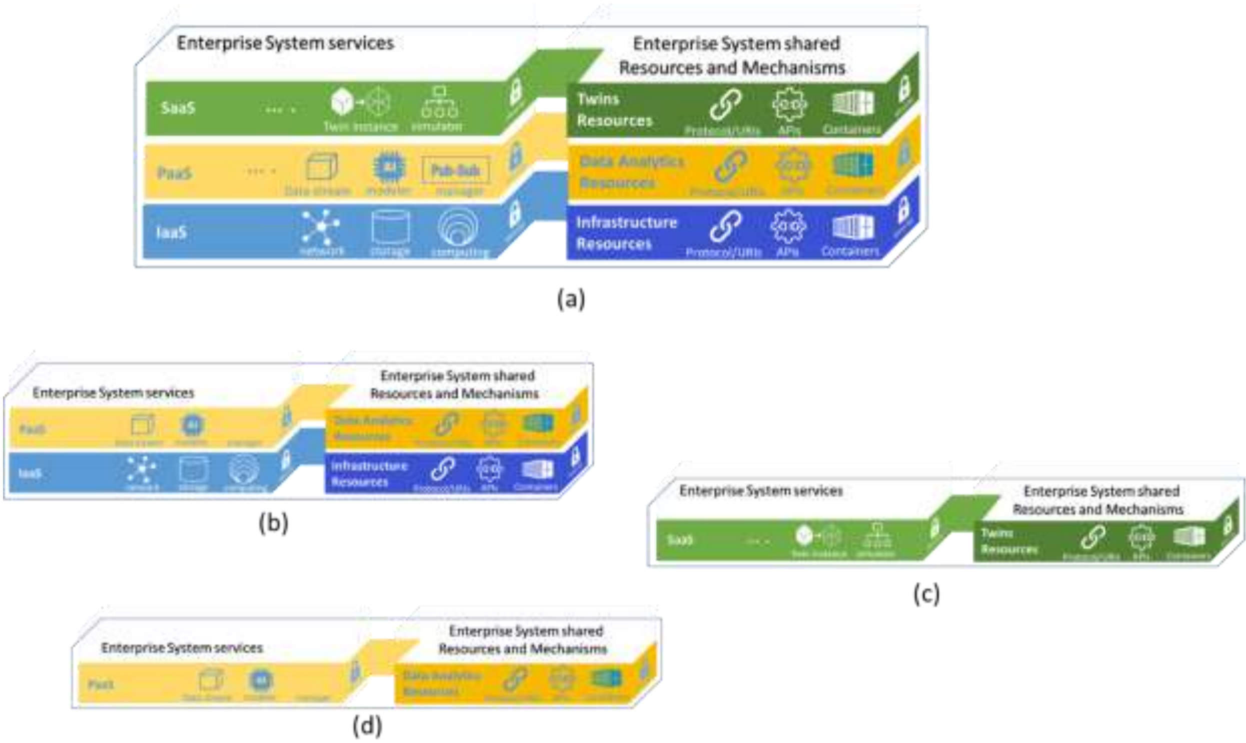


Figure 12. Modeling of enterprise system that belongs to the Destination Earth Ecosystem. An enterprise system contributing with all the three resource types is represented in (a); while, systems sharing Data Analytics and Infrastructure resources are showed in (b). Finally, enterprise systems contributing only either Twins or Data Analytics resources are depicted in (c) and (d), respectively.

Naturally, different enterprise systems can implement only one or more service layer, expose one or more resource type, and implement one or more interoperability mechanism –as showed in Figure 12 (b), (c), and (d).

Applying the ecosystem paradigm depicted in Figure 6 and characterizing the emerging ecosystem node with the digital layers showed in Figure 10, the engineering schema of Destination Earth results as depicted in Figure 13. The schema includes:

- A set of enterprise systems –i.e. the actual Destination Earth ecosystem components.
- The emerging (virtual) platform of the Destination Earth ecosystem.
- The Destination Earth metasystem –i.e. the governance and cybernetic framework.

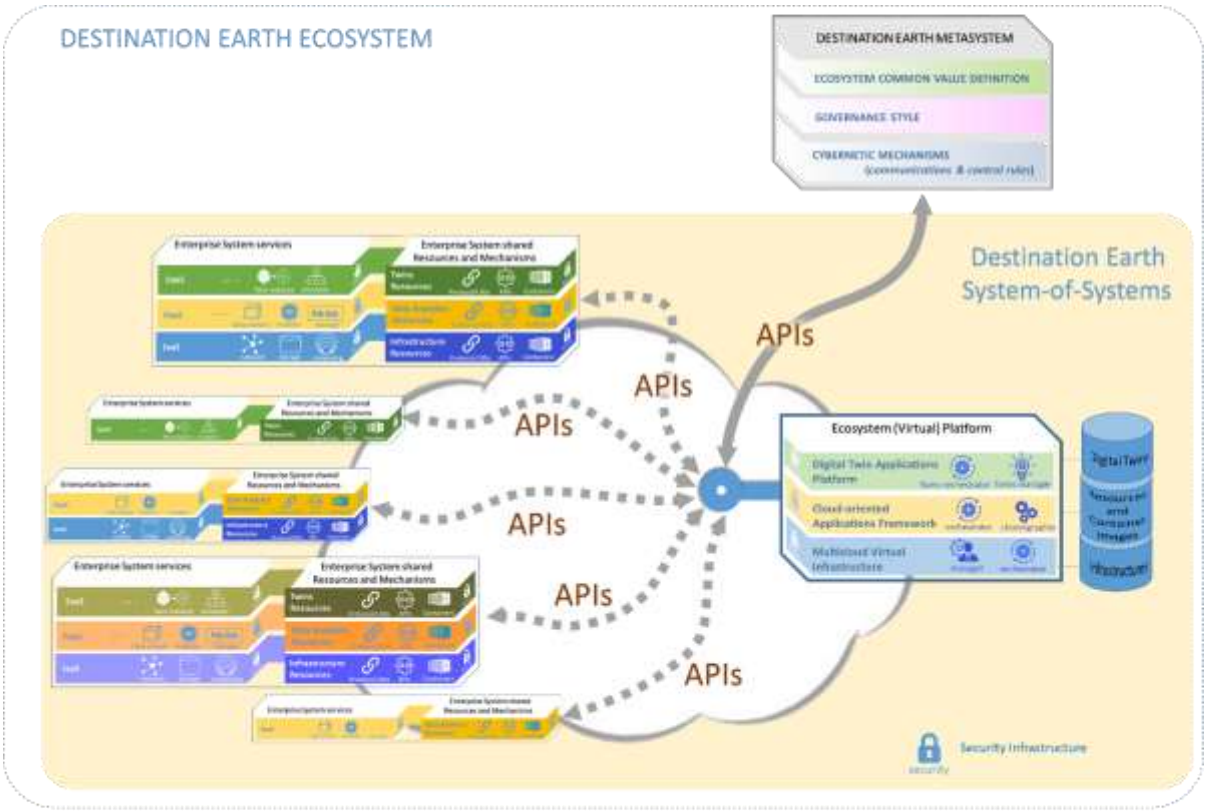


Figure 13. Destination Earth ecosystem engineering model

4.2 Data systems interoperability and brokering services

In the era of Big Data, data movements must be minimized as much as possible (see the "technological paradigms and models" chapter), building distributed solutions and moving analytical software around the ecosystem. To implement an effective and usable distributed data system, metadata sharing and data system microservices must be particularly curated. Considering the multi-disciplinary domain charactering Destination Earth, and the consequent high heterogeneity of datasets to be processed, data mediation and brokering services (along with their APIs) must be considered and developed (Nativi, et al., 2015) (Nativi & Bigagli, Discovery, mediation, and access services for earth observation data, 2009).

Use of APIs and microservices implement an architectural style where application is structured as a collection of services that are: highly maintainable and testable, loosely coupled, independently deployable, and organized around business capabilities.

4.3 Analytical software interoperability

In a distributed system, where analytics software is the main resource to be moved around, it is crucial to fully understand the level of interoperability implemented by that software. It is possible to distinguish among three diverse levels of interoperability, according to the openness, digital portability, and client-interaction style that characterize a given analytics tool –i.e. a data/process-driven analytical model provided as a digital software or service:

- **Model-as-a-Tool (MaaT)**: interoperability consists of user's interaction with a software tool (developed to utilize a processing/analytical model) and not with the model itself or a service API. A given implementation of the analytical model runs on a specific server, and a user interface is exposed to interact with the software. It is not possible to move the model and make it run on a different platform. Benefits include a strong control on the model use and execution. Limitations are on usability and flexibility of the model, as well as its scalability due to the limitation of the specific server. Machine-to-machine interoperability (chaining capabilities) is not allowed.
- **Model-as-a-Service (MaaS)**: as for the previous case, a given implementation of the analytical model runs on a specific server, but this time APIs are exposed to interact with the model. Therefore, interoperability consists of machine-to-machine interaction through a published API –e.g. for a run configuration and execution. Still, it is not possible to move the model and make it run on a different machine. Also in this case, it is not possible to move the analytical software that run on the model server. Concerns deal with a still limited flexibility, possible scalability issues (depending on the server capacities). To note, this time, the existence of possible concerns for less control on the model (re-)use.
- **Model-as-a-Resource (MaaR)**: the interoperability level resamples the same patterns used for any other shared digital resource –like a dataset. This time, the analytical model itself (and not a given implementation) is accessed through a resource-oriented interface –i.e. API. That allows to effectively move the model and make it run on the machine that best performs for a specific use case. There exist clear benefits in terms of flexibility, scalability, and interoperability. Main concerns are about the model sound utilization.

The three different interoperability levels and their interfaces are showed in Figure 14. Destination Earth will be likely required to support all the three levels of analytical software interoperability. Metadata describing analytical models is an important challenge (Nativi, Mazzetti, & Geller, Environmental model access and interoperability: The GEO Model Web initiative, 2013) and must be carefully considered by Destination Earth.

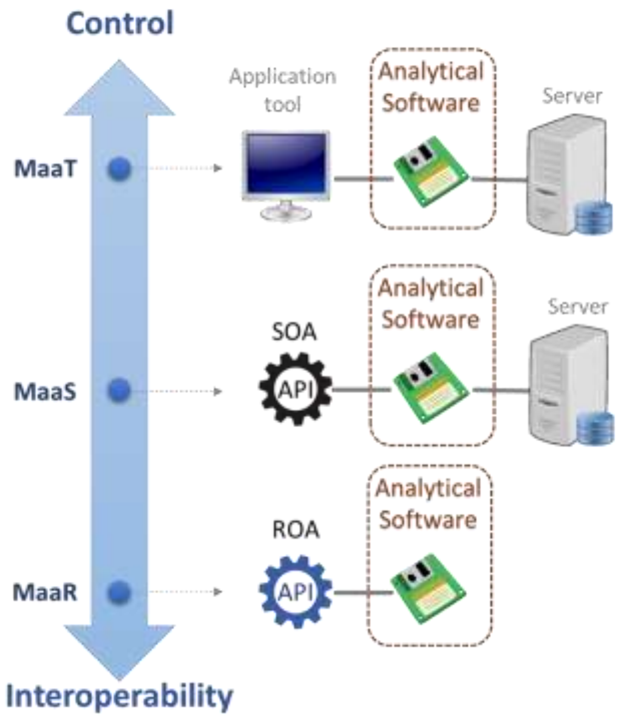


Figure 14. Different Interoperability levels of analytical software

4.4 Destination Earth Technology View

This viewpoint covers the technological implementation of the multi HPC and Cloud system that enable the Destination Earth ecosystem. In particular, the orchestration functionality is addressed by using the containerization technology.

4.4.1 Multi HPC & Cloud System orchestration

The main objectives of a multi HPC & cloud orchestration architecture are:

- To manage, in a simplified way, the ecosystem infrastructure, independently from cloud providers.
- To support distributed resources (e.g. data and models) access and orchestration.
- To deliver highly available applications and services across multiple geographic zones.
- To run models or applications where data are, taking even advantage of serverless functions
- Run different version of an application based on where it runs (e.g. to leverage cloud specific features, or data availability)
- Common workflow and semantics
- Avoid provider lock-in
- Connect IoT sensors data as resources
- Take advantage of edge computing for:
    - IoT sensor data cleaning and normalization;
    - directly running simple models -commonly those with real time or near real time needs, close to sensors;
    - creating simple small private clouds that interconnect with other clouds.

The main challenges to face are:

- Low interoperability between diverse cloud providers –some useful tools exist but still some development is necessary.
- Security.
- Creation of a simple install procedure.
- Meaningful and clear logging strategy.
- Network latency.
- Replications of databases and user storage –in some case handled by cloud providers.
- Proliferation of machines (in particular container machines) to be managed.

4.4.2 Orchestration model using containerization

For multi HPC & Cloud orchestration, the main conceptual actors and their collaboration model are showed in Figure 15. This conceptual model utilizes the **containerization technology**.

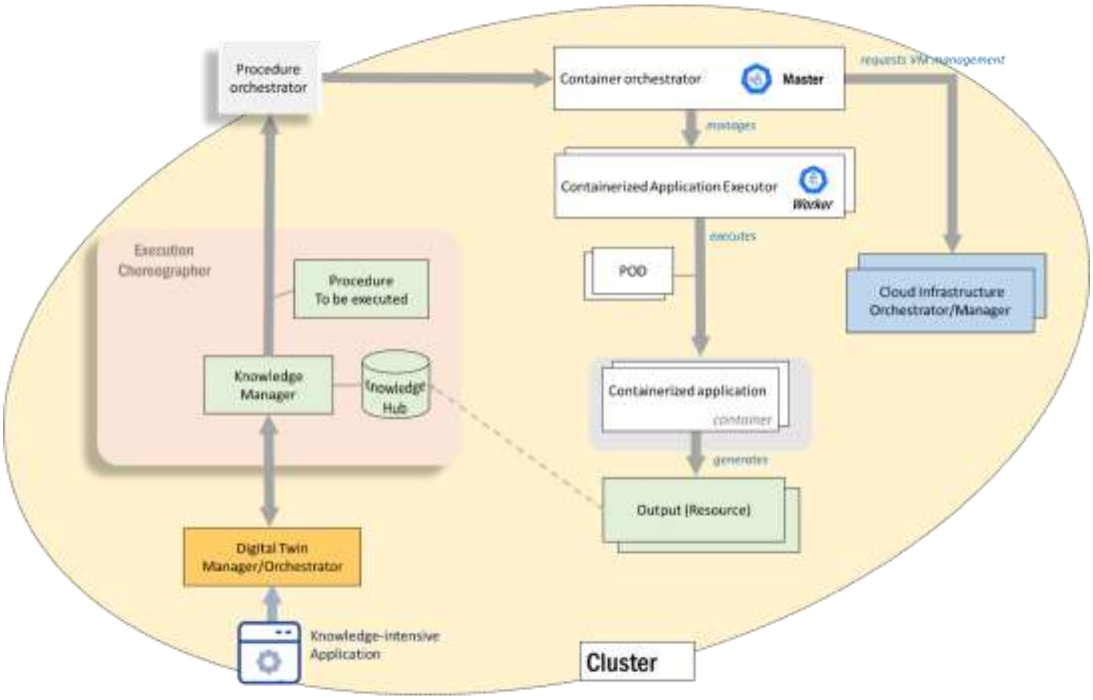


Figure 15. Collaboration model representing multi HPC & cloud orchestration, making use of the containerization technology

4.4.3 Orchestration architectural styles

In agreement with the technology government styles, the following orchestration architectural approaches are possible:

(a) **Centralized**.
(b) **Meshed**.
(c) **Opportunistic**.

4.4.3.1 Styles main benefits and concerns

For the scope of Destination Earth ecosystem, Table 1 summarizes important benefits and concerns characterizing the three different architectural styles.

Table 1. Main benefits and concerns of the possible orchestration architectural styles

Architectural Style	Benefit	Concern
Centralized	• Simpler orchestration management; • Security	• Fault tolerance/point of failure • Tight coupling
Meshed	• Fault tolerance; • Security	• More complex orchestration management (need to align the Master nodes) • Tight coupling
Opportunistic	• Loose coupling	• Security • More complex management by the procedure orchestrator

In addition, **multiple clusters federation** can be utilized to ease different clusters management. In this case, we also distinguish between interconnected clusters (i.e. they are connected using a secure VPN) and isolated clusters (i.e. they can only communicate using the public network). In particular, the last case is a hybrid solution of meshed and opportunistic architectures.

Figure 16 depicts the implementation architecture of the Centralized approach; while Figure 17 shows the Meshed architectural style and, finally, Figure 18 represents the opportunistic architecture.

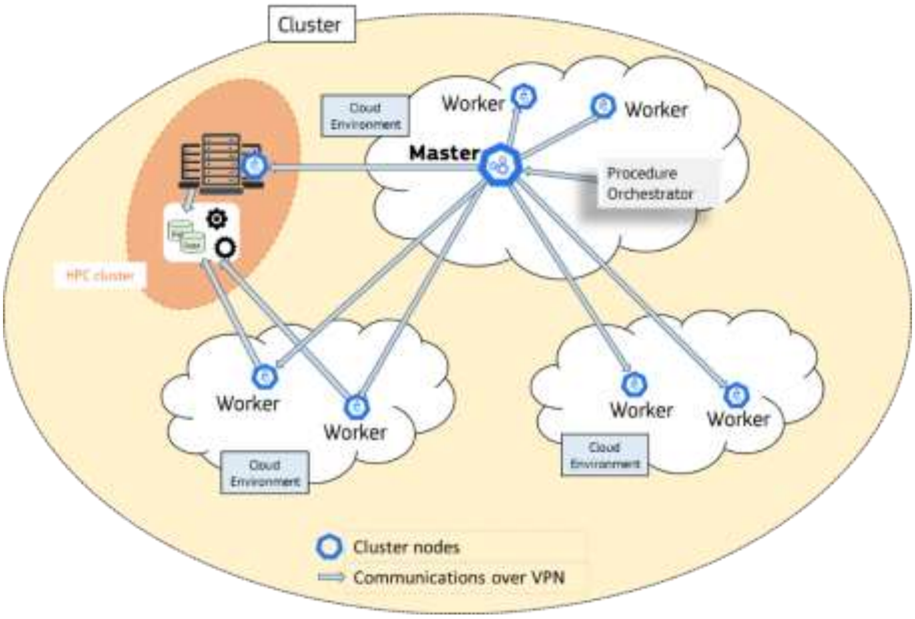


Figure 16. Centralized orchestration architecture

In respect to the centralized solution, an important benefit of the meshed architectural style consists in improving the fault tolerance. On the other hand, an additional task required to the cluster is the alignment of the master nodes.

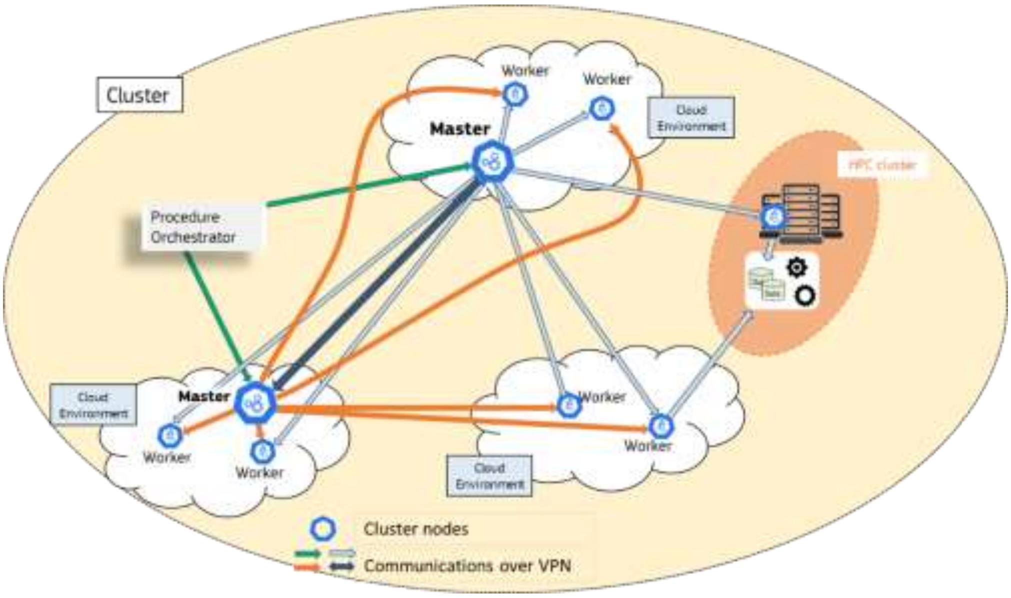


Figure 17. Meshed orchestration architecture

In the opportunistic architectural style, the clusters are fully isolated and do not communicate each other. Therefore, the procedure orchestrator must communicate with them using the public network –this affects the security of the overall ecosystem.

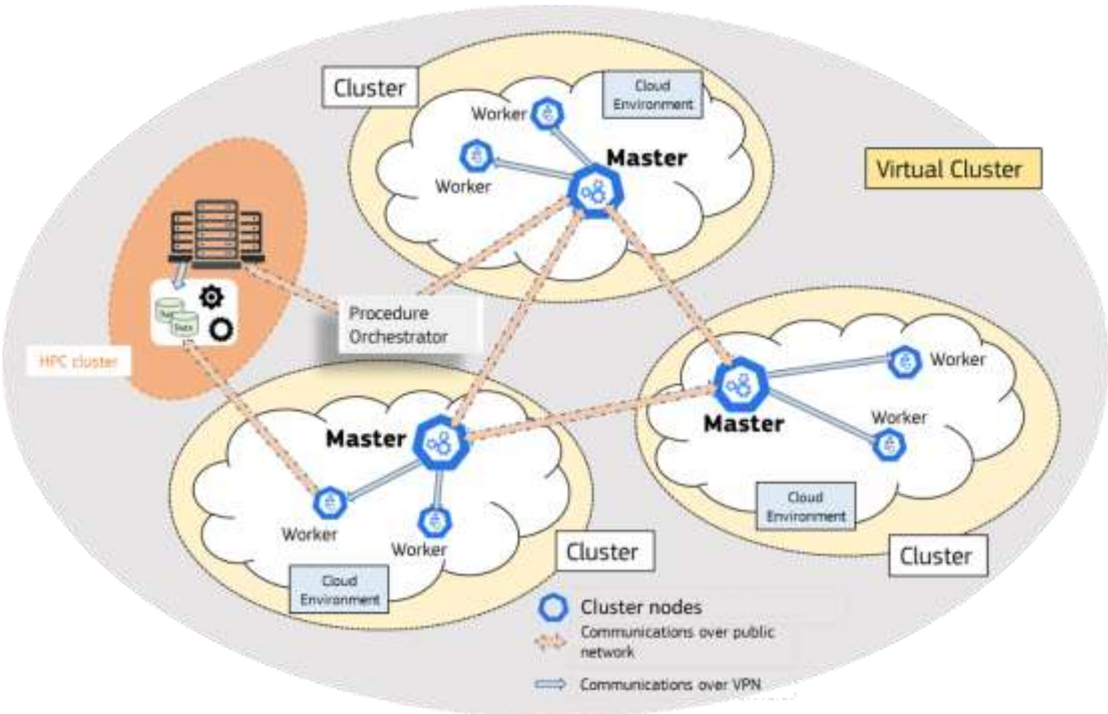


Figure 18. Opportunistic orchestration architecture

The opportunistic architecture can be simplified by introducing a virtual cloud that allows the procedure orchestrator to interact with one master node, only. However, also in this case, the isolated clusters must use the public network to communicate –thus not resolving the security issues. Figure 19 shows this implementation variation of the opportunistic style.

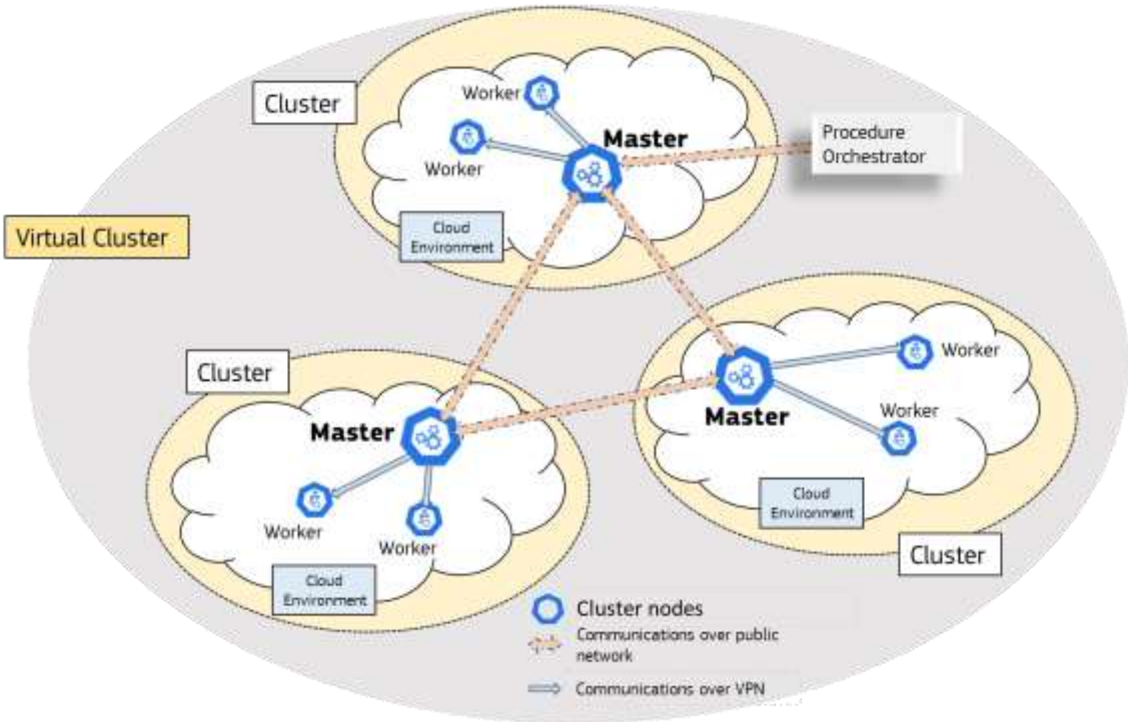


Figure 19. Opportunistic orchestration architecture, defining a virtual cluster in order to make the isolated clusters communicate directly

Cluster Federation variants

The federation architectural variant introduces a federation mechanism and its related components to simplify the clusters synchronization and improve security, allowing the nodes to communicate on a secure VPN. Figure 20 and Figure 21 depict the federation variant for the meshed and opportunistic architectures, respectively.

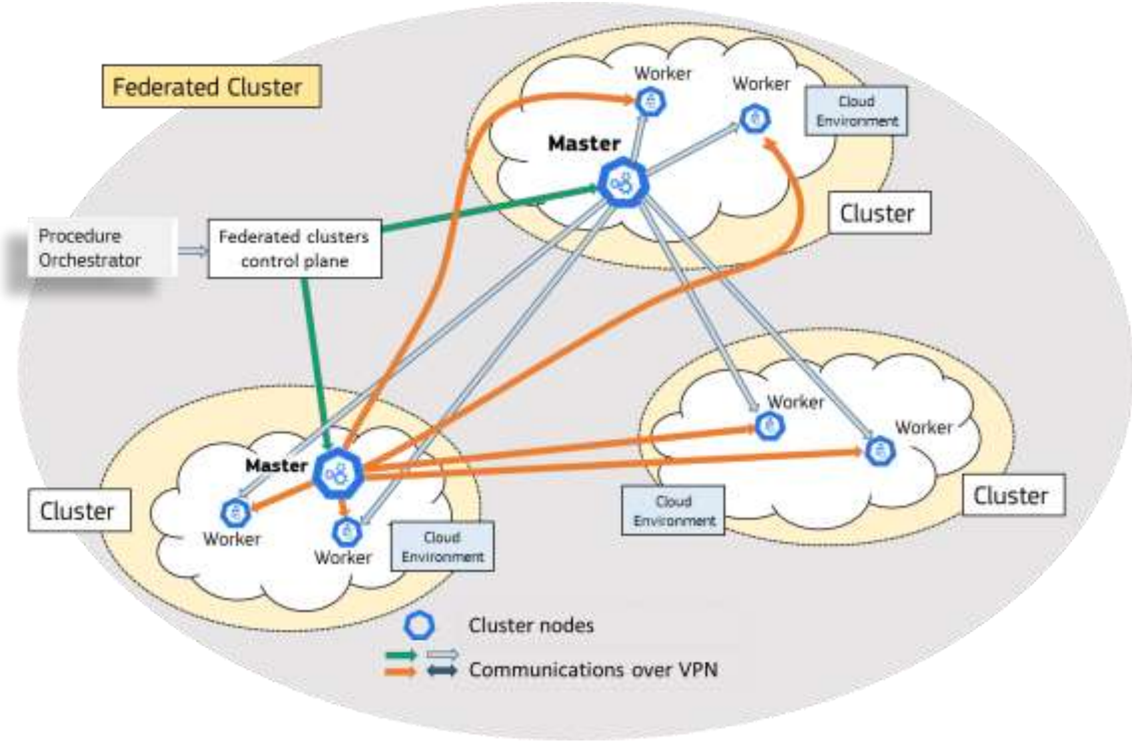


Figure 20. Meshed orchestration architecture, using multiple inter-connected clusters federation.

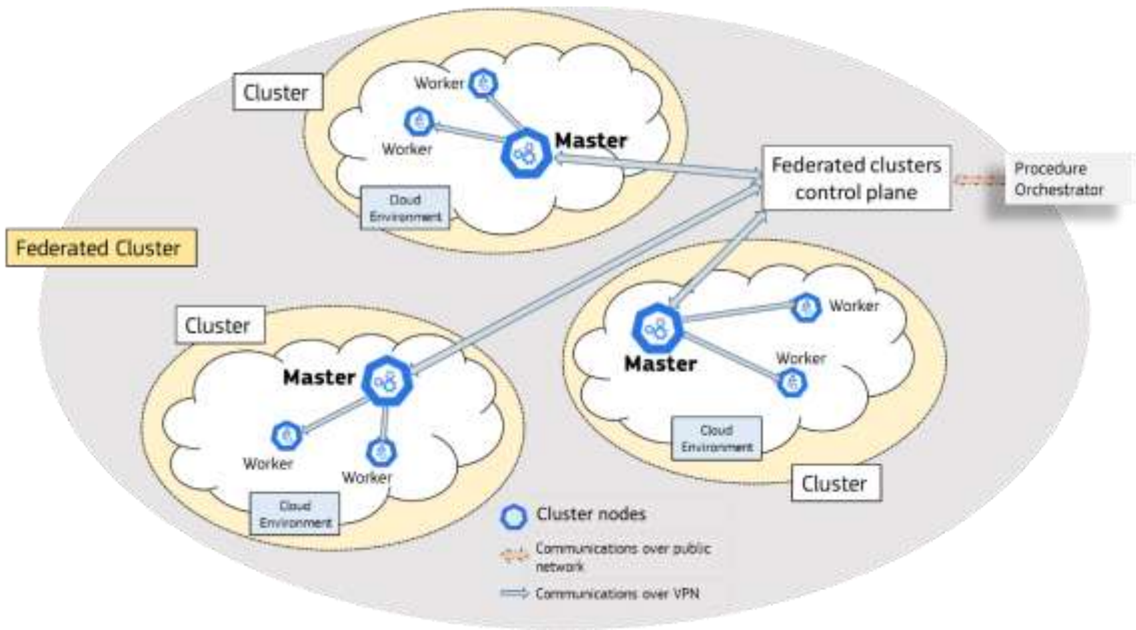


Figure 21. Opportunistic orchestration architecture, using clusters federation.

5 TECHNOLOGICAL IMPLEMENTATION FRAMEWORK

5.1 Technological paradigms and models

The technological implementation of Destination Earth ecosystem must consider the following technological paradigms and models:

Multi-cloud approach and the virtual cloud paradigm: IBM defines multi-cloud approach as the use of two or more clouds from different cloud providers (IBM, 2019). This can be any mix of Infrastructure, Platform, or Software as a Service (IaaS, PaaS, or SaaS). Differently from cloud federations, in the multi-cloud approach, cloud services providers do not have to agree to share resources for a given application (Hong, Dreibholz, Schenkel, & Hu, 2019). Multi-cloud also includes using private clouds and hybrid clouds with multiple public cloud components (TechRepublic staff, 2020). Adopting a multi-cloud approach requires, to provide a rich user experience, the implementation of the virtual cloud paradigm (McKee, Clement, Almutairi, & Xu, 2018): i.e. the development of a virtual layer between the heterogeneous cloud systems and the common DTs/applications development platform. According to the global provider of market intelligence IDC[7] (in its survey on multi-cloud management)[8], 93.2% of respondents (about three hundred US-based enterprise IT decision-makers) reported that their organization is currently using more than one infrastructure cloud. These multi-cloud users reported that optimizing cost, maintaining performance, and ensuring interoperability across clouds are critical to keep their business competitive (Turner, 2019). A similar result was highlighted by the Flexera's 2020 report on the "State of the Cloud" based on the inputs received from 750 technical professionals from around the globe and across a broad cross-section of organizations (Flexera, 2020).

By applying a multi-cloud strategy, organizations and applications are able to increase their efficiency and effectiveness, choosing the right service and provider for each different use case. The factors that are pushing the large adoption of multi-cloud solutions by worldwide enterprises include: (Faction, 2020)

- Avoiding vendor lock-in;
- Overcoming data gravity;
- Optimizing workloads in the cloud;
- Elevating application performance;
- Curbing shadow IT;
- Enhancing disaster recovery capability;
- Meeting regulatory compliance requirements;
- Flexibility;
- Proximity;
- Resilience;
- Interoperability.

On the other hand, it is important to understand and manage some security and performance challenges in multi-cloud architectures, such as: networking between clouds, scalability limits, and

[7] https://www.idc.com/about
[8] https://go.cloudhealthtech.com/rs/933-ZUR-080/images/IDC-whitepaper-multicloud-management.pdf

multiple clouds monitoring (Tozzi, 2019). Users and clients interact with the DE multicloud-based platform as with other virtual (private) cloud.

Multi-cloud paradigm implements a distributed cloud model delivering those traits required by Destination Earth –i.e. flexibility, scalability, evolvability, and viability. In particular, it allows to:

i) Act as both **an (horizontal) enabler of an ecosystem/data space** as well as an **advanced (vertical) tool for elaborating and monitoring thematic public policy needs**. If needed, it must be possible to disaggregate the vertical components (i.e. DTs) from the underlying infrastructure and focus on policy use cases, only.
ii) Build on the **flexible and convergent use of data**, **infrastructures** (e.g. HPC infrastructures, and federated cloud infrastructures for data distribution/access and hosted processing), **software and AI applications/analytics** supported by a strong horizontal framework.
iii) A strong user support component will provide guidance and advice both in the horizontal and vertical/thematic aspects.

IoT-Edge/Fog-Cloud: while cloud data centers are large facilities deployed in a limited number of locations (due to special infrastructure and management requirements), cloud users are spread everywhere in a digitally transformed society –IoT and 5G enabled applications are significant examples. Therefore, clients and users are commonly distant from cloud data centers managed by their preferred cloud provider. Edge and/or fog computing infrastructure can be closer to those devices and applications to bring computing capacity with lower response time (Bittencourt, et al., 2018).

Serveless computing (towards real-time services): Due to the vast heterogeneity of client/user devices (noticeably, their computational capabilities) the network systems diversity, (in particular their latency times), understanding which services should execute on a cloud data center and which on the edge devices remains a challenge (Bittencourt, et al., 2018). The serverless model focuses on the provision of computational functions, with limited resource requirements, that can be deployed closer to user devices; commonly, serveless functions are triggered by user-defined events –existing and well used technologies include: AWS Lambda, Google Cloud functions, and Microsoft Azure functions. This model enables real-time data streams processing and hence services –the ongoing IoT revolution is pushing serveless functionalities. Serverless computing fully support multi-cloud pradigm by allowing organizations to build services as necessary across different clouds with different approaches (Harper, 2020)

HPC-as-a-Service: High-Performance Computing (HPC) is important for enabling extreme digital twin applications. However, the access to supercomputers is out of range from the majority. Recently, cloud computing introduced the concept of providing on-demand resources that can be services (i.e. IaaS and PaaS), resolving existing issues such as hardware maintenance and networking expertise. HPC community is making a great effort to apply these HPC computing concepts in order to get benefits of cloud (Imran, et al., 2020). The main target is to create HPC system that can provide computing power as a service –i.e. HPC-as-a-Service (HPCaaS). Therefore, HPCaaS is the provision of high-level processing capacity to customers through the cloud (Rouse, 2020). This paradigm provides the resources required to process complex calculations, avoiding the investments for skilled staff and demanding software refactoring. An example is the award-winning prototype framework that transforms the IBM Blue Gene/P system into an elastic cloud of multiple joined clouds supporting dynamic provisioning, efficient utilization and maximum accessibility of HPC resources.

5.2 Virtual-cloud architectural solution

The utilization of a virtual cloud services layer (i.e. IaaS, PaaS, and SaaS) permits to utilize a distributed multi-cloud solution, in a dynamic fashion and in a transparent way for users. This, ultimately, obeys to the flexibility and scalability requirements recognized for the Destination Earth digital ecosystem.

As previously introduced, there exist different possible orchestration approaches for the infrastructures contributing to the Destination Earth ecosystems. In keeping with the philosophy of a progressive development, the starting configuration is the Centralized orchestration approach. This is a propaedeutic development for realizing the more complex and distributed ones (e.g. the "opportunistic" federation approach), in the next future.

The centralized orchestration approach is based on a Virtual Cloud (implemented through a multi-cloud solution) that harmonizes a set of heterogeneous, distributed, and evolving infrastructures –e.g. clouds and HPC infrastructures exposing cloud services/interfaces. The Virtual Cloud technology is managed by applying a collaborative governance style –see the already introduced diverse styles. The Virtual Cloud engineering view is depicted in Figure 16.

The Virtual Cloud implements the required interoperability arrangements to execute jobs on different computing infrastructures, making it transparent for the orchestrators. It also manages infrastructure resources scalability. To do that, the Virtual Cloud makes use of a set of services that know the resources (i.e. infrastructure, data, and modeling resources) available on the various clusters and can decide the best way to operate them. Referring to Figure 16, the main components and services of the multi-cloud architecture are introduced in Table 2.

Table 2 Main components and services characterizing the Destination Earth *virtual-cloud solution*

Virtual-Cloud element		Description
Virtual Cloud Services		Set of services offered by the virtual cloud to interconnect multiple clusters, residing on multiple Cloud Environments, logically as a single unit. The *Virtual Cloud* leverages these services to have knowledge of physical resources available on the different clusters and to interact virtually with them.
	Infrastructure resources Broker	a proxy of the cluster resources –i.e. data, computing, networking.
	DT components Broker	a proxy of the components building a DT –e.g. data streams and analytical models.
	Task Execution Optimizer	When an Orchestrator requests the execution of a *container*, the Virtual Cloud, using the previous services, decides where to execute the job sending the request to the most appropriate Kubernetes Cluster.
Cloud Environment		Physical computing infrastructures –e.g. Openstack, AWS, GCP, VMware, Morpheus, etc.
Cluster		Kubernetes cluster implementing asset of basic services to join the virtual cloud (e.g. *Cluster API*, *Cluster Autoscaler*, *Istio*)
	Master cluster	Core Kubernetes cluster where the *Virtual Cloud Services* reside
	Worker cluster	Kubernetes cluster that (if allowed by its Cloud Environment) can utilize *Cluster Autoscaler* and *Cluster API* to manage the underlaying physical infrastructure.
(Procedure) Orchestrator		Software component that receives execution requests from a *Client* and communicates with the *Virtual Cloud Services* to finalize the needed jobs.

	The *Virtual Cloud Services* decide where jobs execution happen.

5.3 Virtual Cloud software components and services

The framework makes use of microservices and API technologies to implement elastic interoperability between software components. This framework fully supports the Destination Earth flexibility, modularity, and evolvability traits. That is achieved by de-coupling DTs and software applications from the virtual cloud layer: i.e. the Destination Earth horizontal common infrastructure. In turn, the virtual cloud layer virtualizes a set of operational infrastructural environments (i.e. the multi-cloud environment) by implementing the necessary mediation and brokering services and publishing a set of open, common, and consistent network services and related APIs to clients. The virtual cloud services allow the multi-cloud infrastructures (i.e. the ecosystem enterprise components) to remain substantially autonomous and freely evolve in time, avoiding the propagation of changes on the DTs and the client applications, making use of them. Moreover, new operational infrastructures can be added enriching the ecosystem, in a transparent way. It is noticeable how the introduced layers de-coupling beautifully supports the separation and consequently the simpler addressing of the different concerns, expressed by the three principal DE stakeholders: the ecosystem users and clients, the organization(s) steering and funding the ecosystem as a whole, and the enterprises managing the infrastructures that contribute to the ecosystem system-of-systems (i.e. the enterprise systems). As depicted in Figure 22, their needs and constraints clearly influence the different services layers of the framework; thus, the more these layers are decoupled the more the stakeholders' concerns are separated –applying the "separation of concerns" computer science pattern (Dijkstra, 1982). Finally, the introduced technological framework supports both a Service-Oriented Approach (SOA) and a Resource-Oriented Approach (ROA), satisfying another important DE principle.

The engineering and technology framework characterizing the virtual-cloud solution is shown in Figure 22. The framework makes use of microservices and API technologies to implement elastic interoperability between software components. This framework fully supports the Destination Earth flexibility, modularity, and evolvability traits. This is achieved by de-coupling the software applications and DT software layers from the horizontal common infrastructure. In turn, the virtual cloud virtualizes a set of different operational infrastructures (i.e. the multi-cloud environment) by implementing the necessary mediation and brokering services and publishing a set of open, common, and consistent network services and related APIs to clients. The virtual cloud services allow the multi-cloud infrastructures (i.e. the ecosystem enterprise components) to remain substantially autonomous and freely evolve in time, avoiding the propagation of changes on the DTs and the client applications, making use of them. Moreover, new operational infrastructures can be added enriching the ecosystem, in a transparent way.

Referring to Figure 22, it is noteworthy how the introduced layers de-coupling beautifully supports the separation and consequently the simpler addressing of the different concerns, expressed by the three principal Destination Earth stakeholders:

(a) The ecosystem Users and Clients (i.e. machine-to-machine usage interaction);
(b) The European Commission that envisions and funds the Destination Earth ecosystem as a whole;
(c) The enterprise systems and infrastructures that contribute to the ecosystem system-of-systems.

Stakeholders' needs and constraints clearly influence the different services layers of the framework; thus, the more these layers are decoupled the more the diverse concerns are separated –applying the "separation of concerns" software engineering pattern.

Finally, the defined technological framework supports both Service-Oriented and Resource-Oriented approaches, one of the Destination Earth main requirements. The first approach is important to enable complex scientific business/domain logic; while, the second approach is essential to discover and benchmark shared data and software resources (e.g. AI-based analytical software) and chain them to generate Digital Twins and/or applications. Application-enabling components of Figure 22 are described in the next paragraphs.

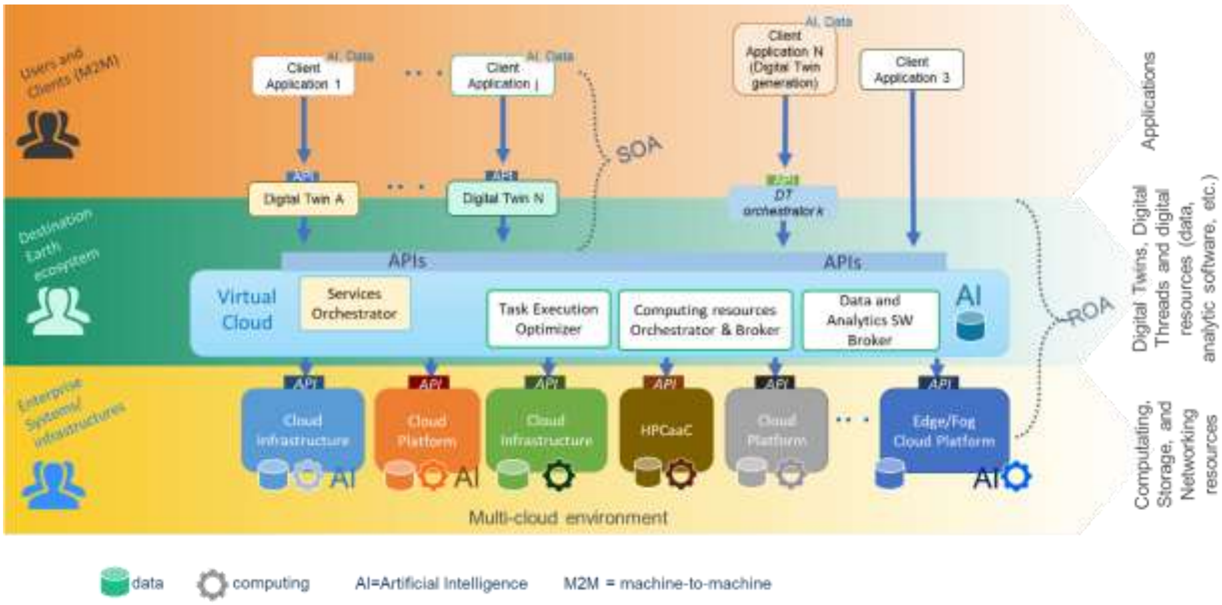


Figure 22. Destination Earth technological framework based on the Virtual-Cloud paradigm

5.3.1 Client software application

It is a software application (M2M or GUI-enabled) that makes use of the Destination Earth DE services, by accessing macro- and micro-service APIs. In particular, the schema shows applications utilizing the DT instances to develop a business logic, as well as applications using the virtual cloud services to discover, benchmark, and orchestrate shared digital resources –e.g. datasets and analytical software.

5.3.2 DT (Service) orchestrator

It is a software component that implements the business logic for orchestrating the components and services that are necessary to implement a DT instance –e.g. a sound operational workflow made up of data streams and analytical software. This component can be provided either by a third party (i.e. a client application) or by Destination Earth horizontal platform; in this case, it consists of a more general Internet services orchestrator, it is deployed in the virtual cloud, and is named "*Service Orchestrator*" –see Figure 22. In this case, *Service Orchestrator* orchestrates shared components to finalize the necessary tasks and generate a DT, including:

- Analytical software (processing models) retrieval/access/provision (by applying a MaaT/MaaS/MaaR approaches);

- Container(s) instantiation;
- Data ingestion request management;
- Container(s) execution request management;
- Output(s) storage request management;
- Security aspects implementation.

This component exposes a well-known API to be invoked by client applications.

5.3.3 Virtual Cloud middleware

DE virtual cloud is a middleware framework that allows the execution of tasks and applications (noticeably, workflows orchestration for DT instance generation) on a multi-cloud environment (including HPC-as-a-Cloud), exposing that as a unique and consistent virtual capacity. This middleware implements the necessary interoperability arrangements to finalize application execution and infrastructural scalability need. In addition, the middleware is capable of querying the underlying multi-cloud infrastructures for accessing and using digital resources they provide –e.g. computing, data, and analytical software.

When an orchestrator requests a workflow execution, the Virtual Cloud, on the basis of a set of predefined criteria (e.g. data and analytical models availability, computing capacities, and energy saving) decides where to execute the job and satisfy a set of agreed policies (e.g. to minimize data movement and time lapse), sending the request to the appropriate computing infrastructure(s). The virtual cloud exposes a set of APIs to discover, access, and benchmark the available virtual resources (e.g. data, analytical models, computing resources, etc.) that are provided by the connected infrastructure, supporting a resource-oriented approach (ROA).

The Virtual Cloud middleware must support a number of heterogeneous cloud infrastructures, either public or private –i.e. a multi-cloud environment. To accomplish this, the middleware utilizes the following open source technologies:

- Kubernetes (docker orchestrator)
- Cluster API (computing infrastructure orchestrator)
- Cluster Autoscaler (autoscaling service)
- ISTIO (service mesh and mutual TLS)

5.3.3.1 Virtual Cloud middleware components and services

Virtual Cloud middleware includes a set of open-source technologies (which are cloud-platform-neutral) to implement some ancillary services commonly provided by a cloud platform. In addition, the middleware implements some cloud-based components/services that enables disparate infrastructures cooperation:

- Infrastructure resources Orchestrator and Broker: Virtual Cloud implements the required interoperability arrangements to execute jobs on different computing infrastructures, making it transparent for virtual cloud clients. It also manages infrastructure resources orchestration and scalability.
- Data and Analytics SW Broker: The Virtual Cloud must be able to discover & access the DT components (data & modeling resources) that are available on the federated infrastructures.
- Task execution Optimizer: When an orchestrator requests the execution of a workflow (e.g. a containerized application), the Virtual Cloud, using the previous functionalities, decides where to execute the job sending the request to the "most appropriate" infrastructure(s).

- Internet Services orchestrator: This software technology orchestrates all the necessary (Internet) services chaining to implement a workflow.

Those components/services (represented in Figure 23) run in a cloud environment –depicted as a dark green box. On the same environment, a set of ancillary services (e.g. object storage (Min.IO), queue messaging service (KubeMQ), etc.) could be deployed, configured, exposed either directly on Internet or available on the private service mesh only.

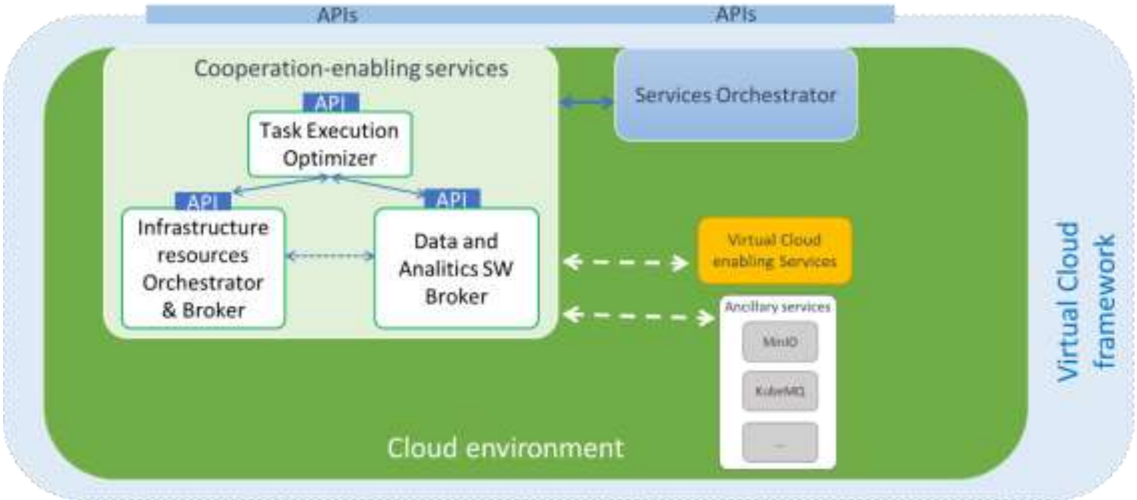


Figure 23. Virtual-Cloud middleware components/services

5.3.3.2 Infrastructure resources orchestrator and broker

The *Infrastructure resources broker* takes care of implementing the required interoperability arrangements to execute jobs on different computing infrastructures, making it transparent to the orchestrators. Besides, it (centrally) manages infrastructure resources scalability. The technological solution to implement the *Infrastructure resources broker* is showed in Figure 24.

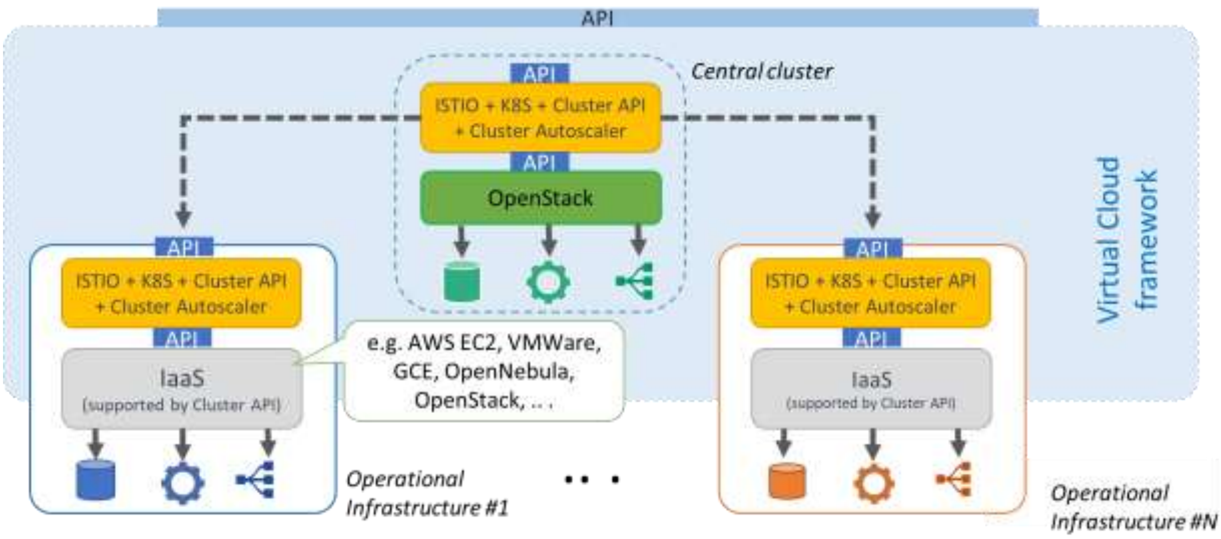


Figure 24. Technological solution of the "Infrastructure resources broker" component of the Virtual Cloud

5.3.3.3 Data and analytic SW Broker

The data and analytic SW broker consists of an analytical software broker and a data broker; the latter, in turn, is composed of two software technologies for data sharing and data harmonization/mediation. The recognized data sharing and access technological solution is shown in Figure 25.

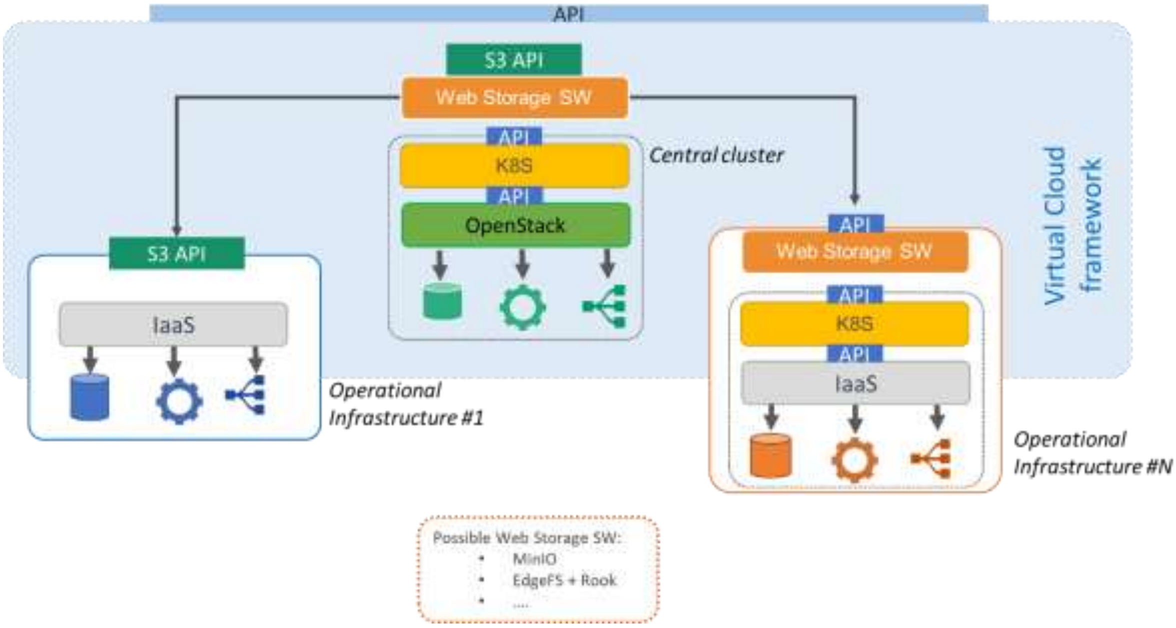


Figure 25. Data sharing and access technology for the DT Components broker of the Virtual Cloud.

5.3.4 Supporting special applications

According to the flexibility requirement for Destination Earth, it is reasonable to assume that for some application and DT workflows, the containerization approach and related technology cannot be utilized with success –e.g. lack of analytical software openness and need of a very low latency time. In those cases, the Virtual Cloud should be designed to support proxy services, avoiding to break the general architectural model.

The default and *special* procedures are described by the sequential and collaboration diagrams depicted in Figure 26 and Figure 27.

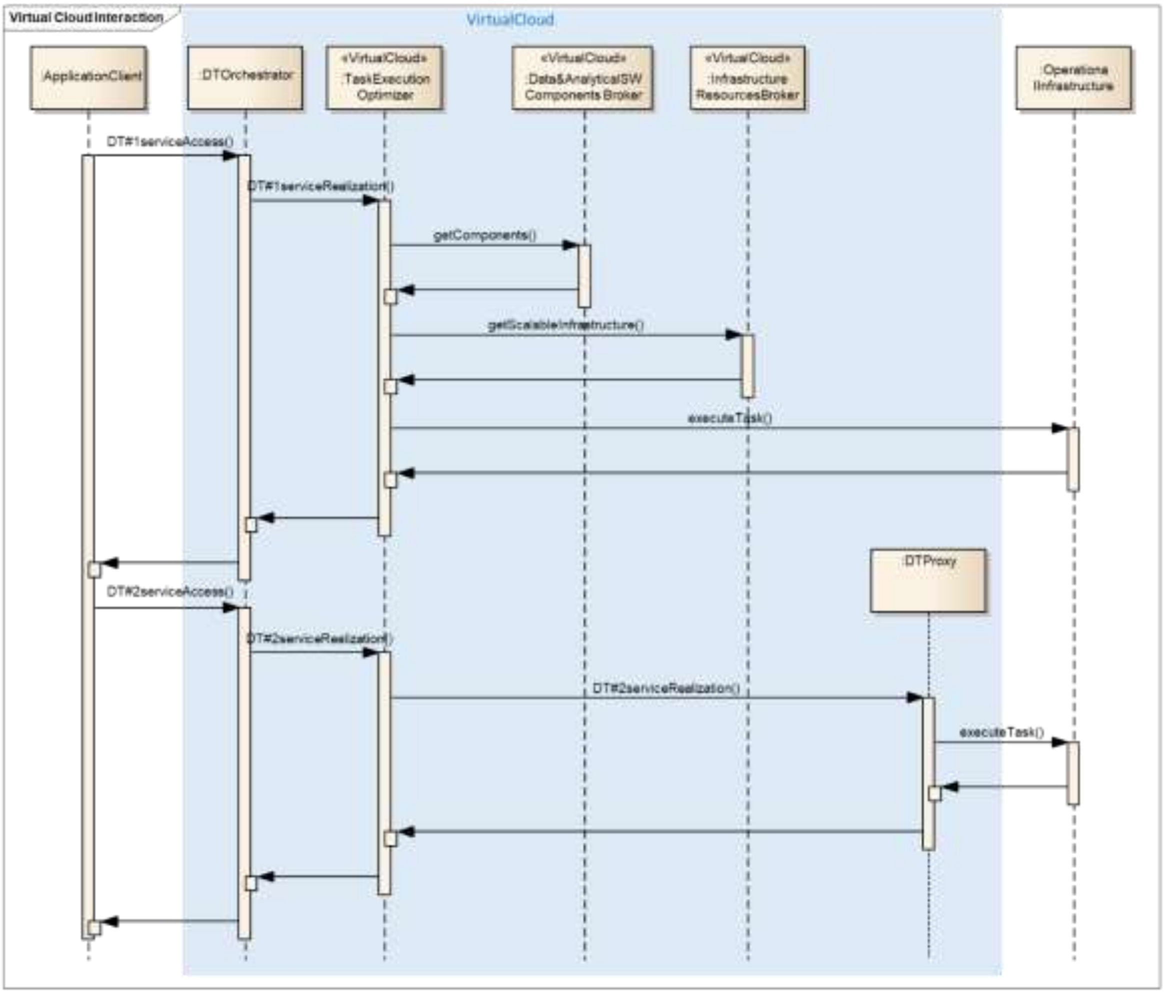


Figure 26. DT workflow execution sequential diagram

A Client application (DT Use Case) asks the *Orchestrator* for available DT workflow(s) and the related data. While a Client can reside ideally everywhere, the O*rchestrator* should live inside the Virtual Cloud as k8s applications, exposing the necessary public APIs. This permits to leverage *Istio* security services and hide the virtual cloud topology.

The *Orchestrator*, once received the Client request, talks to the *Virtual Cloud* (via the exposed APIs) to know where are the necessary resources and where to execute the job(s). In this phase, constraints are commonly applied by the *Virtual Cloud* to take the best scalability decision –e.g. the physical resources needed (e.g. CPU and RAM), the input data availability, the resources costs or the exclusivity of the infrastructure that operates a given Digital Twin (e.g. a workflow that cannot be containerized or that requires dedicated hardware or has license dependency). This is the case of the first request (*DT#1serviceAccess*), as depicted in Figure 4.

The execution of a workflow that cannot be containerized, might be handled by exposing a proxy service (on a machine belonging to the same infrastructure and network) that implements a basic and minimal Kubernetes cluster (i.e. single node like microk8s or k3s). In this way, security is still granted by *Istio* services and the DT workflow can be hidden being not directly accessible from outside of the *Virtual Cloud*.

This solution, which represents the second request showed in Figure 26 (*DT#2serviceAccess*), allows a given Digital Twin workflow to be executed by accessing an internal entry point –that exposes internal APIs to the proxy cluster. The corresponding collaboration diagram is depicted in Figure 27.

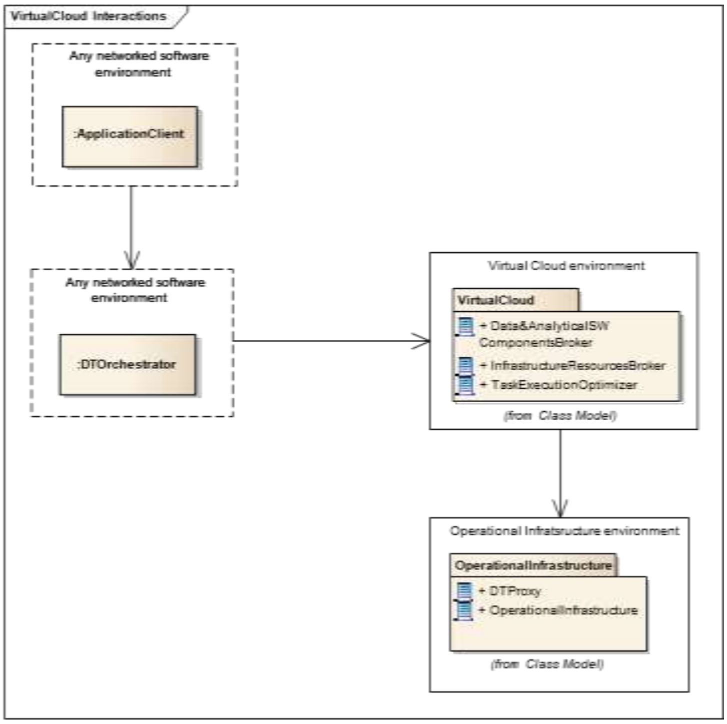


Figure 27. DT workflow execution collaboration diagram

The deployment schema along with the software technology configurations are showed in Figure 28 and Figure 7. In Figure 6 three scalable computational environments are represented:

(1) The "core Kubernetes cluster" contains the Virtual Cloud APIs and is the main entry point to the ecosystem. In this environment, the *Cluster API* and *Cluster Autoscaler* services could be deployed; however, they are less essential than in the federated clusters –i.e. "worker nodes". When an infrastructure joins the Virtual Cloud a DNS, handled by *Istio*, is updated with the available service entry point. In this way, even if a service is available on multiple clusters, it is easy to distinguish and load balance them.

(2) A "worker nodes", in addition to *Kubernetes* and *Istio*, are preferably backed by *Cluster API* and *Cluster Autoscaler* services, enabling them to optimally manage used resources. When the Virtual Cloud sends a job request, if *Kubernetes* doesn't find a node to execute it, and, if it has still scaling capabilities, the *Cluster Autoscaler* (utilizing the *ClusterAPI* knowledge of the federated infrastructure) will spawn a new node to permit its execution. Once the execution is ended, and a "*cooldown*" period is passed, the node will be drained by the *ClusterAPI* responding to a *Cluster Autoscaler* request.

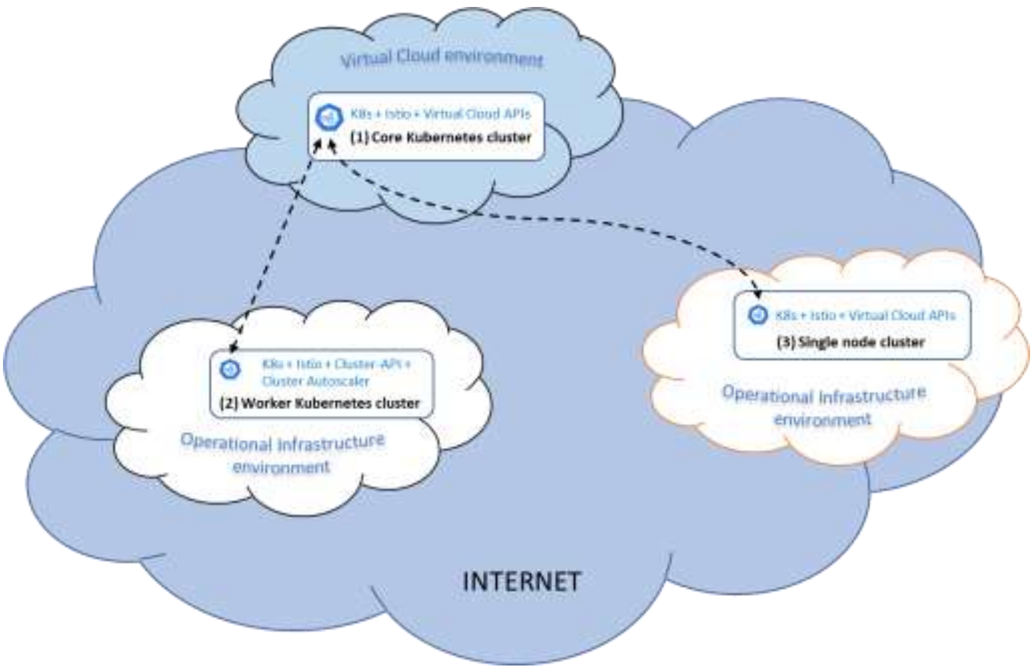


Figure 28. DT workflow execution deployment diagram

(3) In the case of a Digital Twin that cannot be containerized, a solution would be to install a "minimal Kubernetes cluster" with *Istio* that can directly communicate with the Digital Twin application via a proxy service, which can be called by the Virtual Cloud. In this way, the general DT workflow archetype would be respected, in keeping with the Destination Earth architectural requirements and system interoperability constraints.

6 PROOF-OF-CONCEPT

6.1 Virtual Cloud Proof-of-Concept

A proof-of-concept (PoC) of the Virtual Cloud technological architecture was developed and demonstrated in October 2020 by JRC, in collaboration with ESA, ECMWF, and EUMETSAT, and with the support of CNR for some multi-cloud interoperability aspects. The PoC demonstrated the Virtual Cloud distributed computing, the data sharing functionalities, and the analytical SW discoverability and access according to the MaaR paradigm. Noticeably, the CoP showed the architecture effectiveness to support the evolution in real time of an actual ecosystem, enabled by a virtual cloud layer. The demonstrated setup is represented in Figure 29 and Figure 30. In particular, the PoC demonstrated:

- A GUI-based application launching (in parallel) multiple runs of two use cases/DTs, enabled by the Virtual Cloud.
- The overall ecosystem evolution at runtime: passing from 2 different working clusters (i.e. infrastructures) connected to the central cluster (time indicated as T1) to 5 heterogeneous infrastructures, by connecting three more diverse infrastructures in real time (indicated as T2).
- Distributed Resources (data and analytical software) discovery, browsing, and access on-the-fly (applying the resource-orientated approach).
- The flexibility and effectiveness of the Task Execution Optimizer, which decided different execution platforms (according to the number and types of infrastructures connect to the central cluster) to optimize the data moving and computing times.
- High transparency for both Clients and Developers, by exposing APIs and microservices.
- High interaction simplicity for Administrators and Managers, via a set of simple open administration APIs and IT automation engines –e.g. ANSIBLE scripts.

Figure 31 depicts a snapshot of the ecosystem cooperating infrastructures dashboard demonstrating the evolution of the Destination Earth ecosystem, at runtime, based on the Virtual Cloud technological solution. Figure 32 shows a couple of snapshots for data and resource common discovery and access. Finally, Figure 33 depicts the exposed open APIs and IT automation engines (ANSIBLE scripts) used for enabling the PoC.

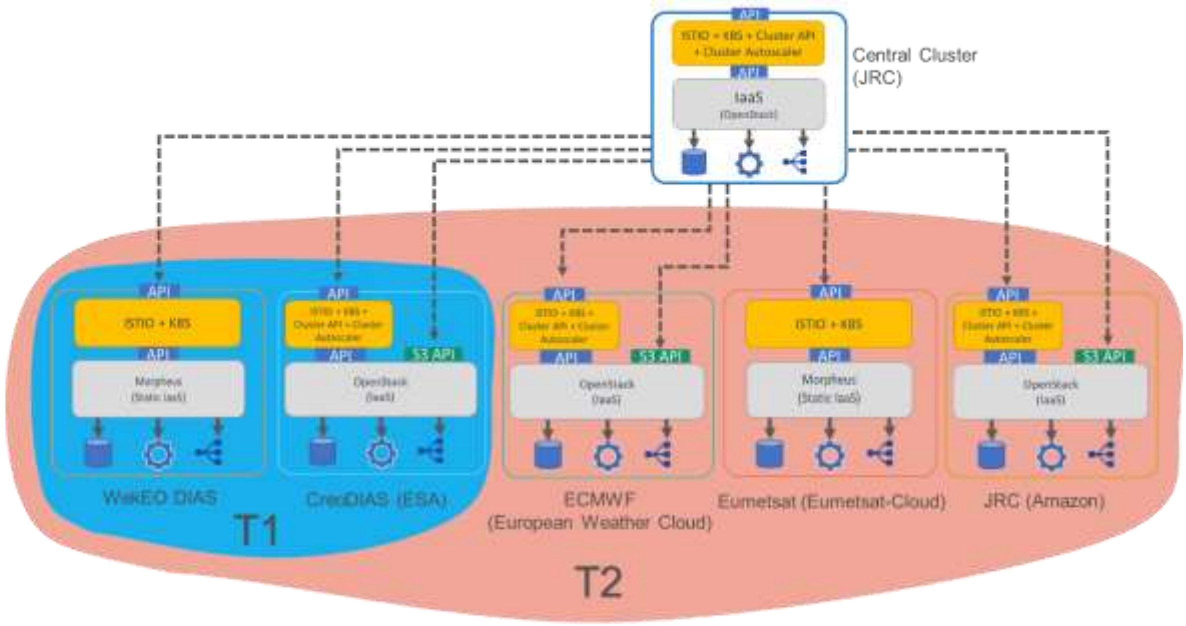


Figure 29. Setup of the Proof-of-concept demonstrating ecosystem functionalities and evolution in time (T1=starting ecosystem configuration; T2=ecosystem evolution at runtime).

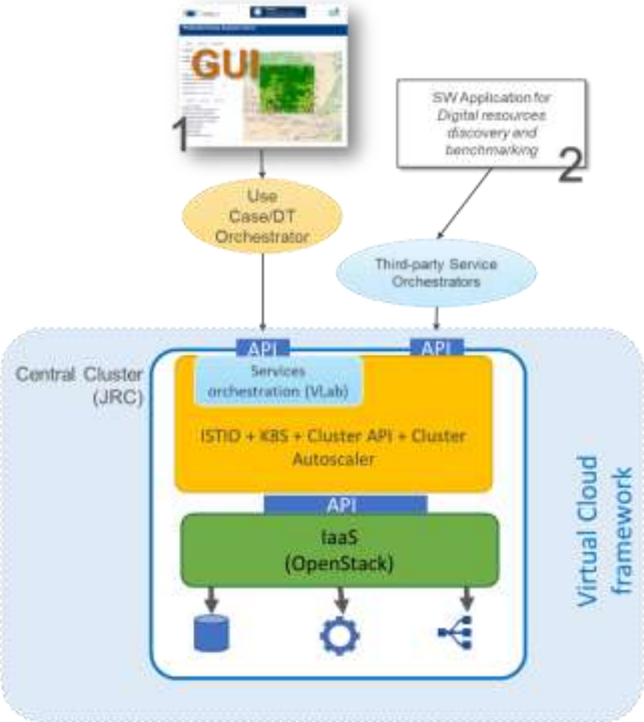


Figure 30. Setup of the Virtual Cloud central Cluster for the Proof-of-concept demonstration

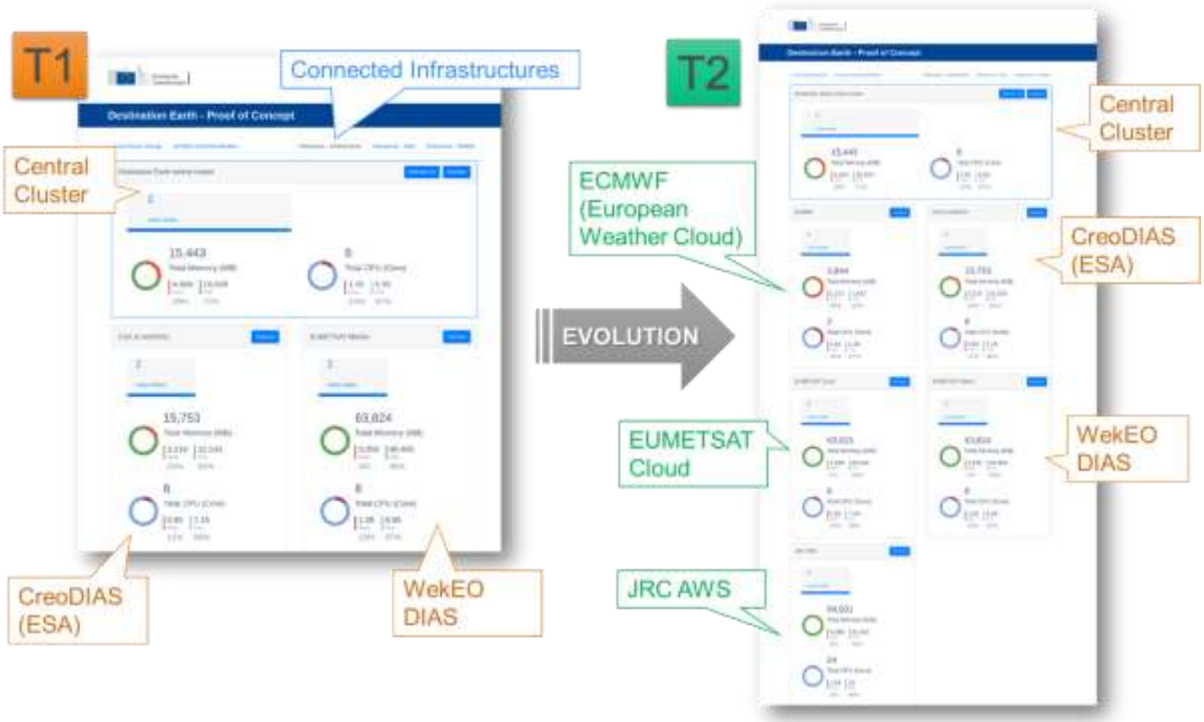


Figure 31. View of the Ecosystem evolution using the connected and cooperating infrastructure dashboard.

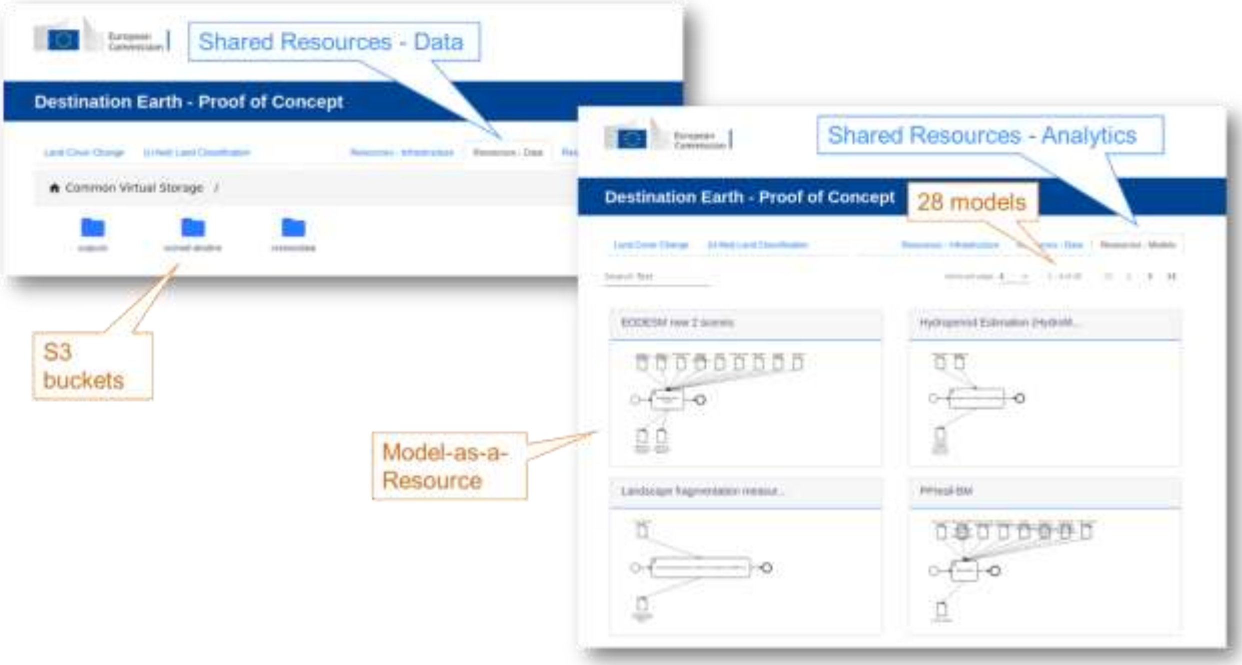


Figure 32. Data and Analytical software sharing as demonstrated by the PoC.

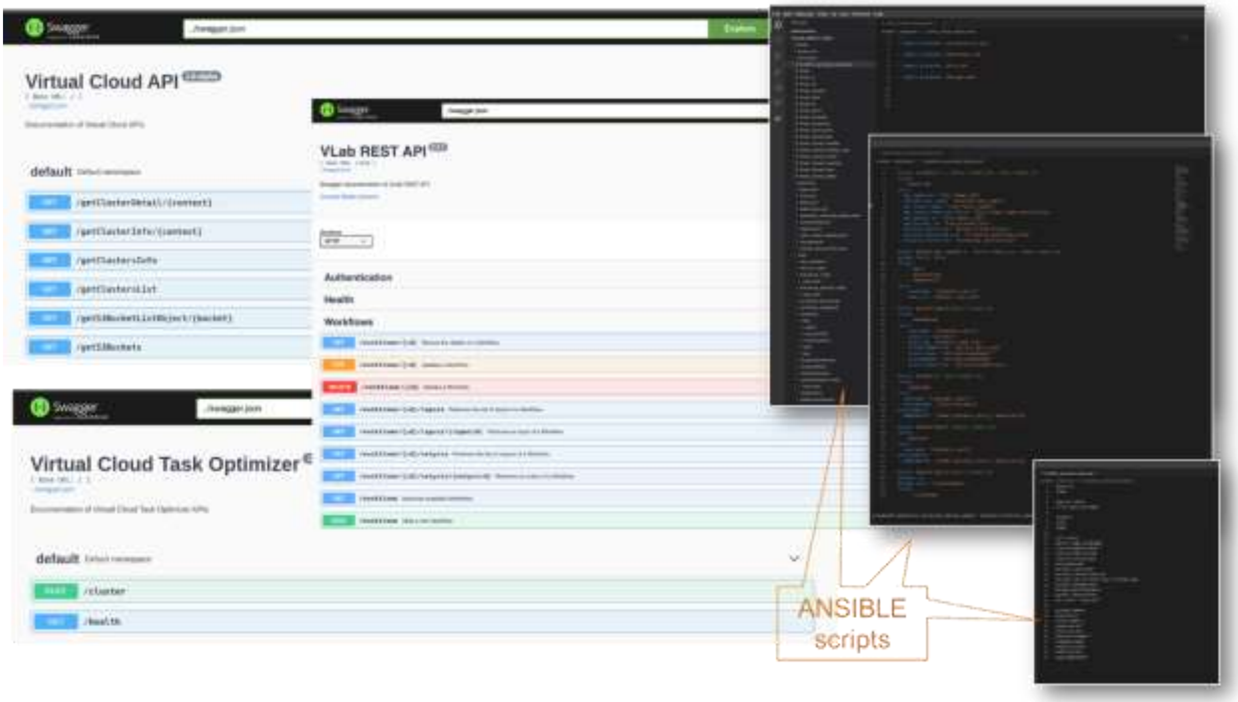


Figure 33. Exposed open APIs and IT automation engines (ANSIBLE scripts) used for enabling the PoC

6.1.1 Main outcomes

The Proof-of-Concept successfully demonstrated the feasibility of the proposed Virtual-Cloud-based technological architecture, and its functional effectiveness –within the demonstration scope. Therefore, the experimentation confirmed:

- The development of an effective Destination Earth ecosystem –consisting of relevant, current, and future infrastructures and for sharing resources that are relevant for the Green Deal Data Space.
- The implementation of a system-of-systems architecture that enable the evolvability and viability of the Destination Earth ecosystem, in time.
- The possibility to sustain the Destination Earth ecosystem architecture, considering operations and maintenance challenges among others.
- The need to define, as a key success factor, the ecosystem technological government style, and what must be defined as the Common Value characterizing an Ecosystem.

In particular, the PoC showed that a virtual cloud platform, building on a multi-cloud approach, can:

- implement the Destination Earth core platform and support heterogeneous use cases/DTs (adopting a service-orientated approach);
- be flexible enough to support rapid Ecosystem evolutions;
- support a resource-orientated approach, to dynamically share distributed digital resources and allow their benchmarking;
- provide different Interoperability levels to Clients and Users, through APIs and microservices;
- provide simple ecosystem administration and management tools, through IT automation engines.

Moreover, the PoC was useful to outline some existing challenges and developing opportunities, dealing with Destination Earth and the Green Data Space, more generally:

- to investigate static versus scalable infrastructure tenants and their relation with security issues;
- to define an efficient data sharing and replication strategy –in keeping with the optimization criteria taken by the task optimizer;
- to define a strategy to move from the present MaaS (Model-as-a-Service) approach to the MaaR (Model-as-a-Resource) one –this is important to move the code where data is and enable open science;
- to further decouple application business logic from the Digital Twin software components –pushing a “true” data-driven approach;
- to continue connecting other existing EO infrastructures (e.g. the other DIAS platforms and HPC-as-a-Cloud) along with their resources (data and analytics);
- to connect non-EO infrastructures and, noticeably, IoT based platforms leveraging the edge and fog computing patterns.

6.2 Future development

The PoC was also instrumental to recognize the need for further investigating the intelligence that must be implemented at the Virtual Cloud level. This originated the development of a second proof-of-concept, which is still under development.

6.2.1 Digital thread/services orchestration

As discussed in the previous chapters, a DT implementation requires to address important interoperability challenges. It is essential to orchestrate and operate a set of different types of digital resources, noticeably, datasets and analytical software, which must be accessed and used by means of APIs and microservices. To enable that, digital resources must be formally described (i.e. metadata) and referenced via online services and protocols. The heterogeneity, characterizing metadata schemas and interface protocols, is commonly addressed by using mediation and brokering services (Guo, et al., 2020), (Nativi, et al., 2015), (Nativi & Bigagli, Discovery, mediation, and access services for earth observation data, 2009). To be effective and enable efficient orchestration services, the mediation and brokering services must cover several interoperability levels (see Figure 34.) up to the "pragmatic" one (Nativi, Santoro, Giuliani,, & Mazzetti, 2019), (Vaccari, Craglia, Fugazza, Nativi, & Santoro, 2012) –as showed in Figure 34.

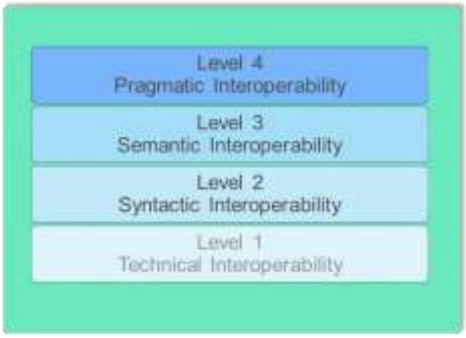


Figure 34. Interoperability levels to be covered by Destination Earth mediation and brokering services

Therefore, to implement a digital thread and its associated DT, it is important to test the effectiveness and usability of the deployed Internet services orchestration, along with its enabling mediation and brokering functionalities. This is the objective of a second proof-of-concept (under development) that aims at understanding the orchestration challenges, which the virtual cloud will be asked to address, and define a set of best practices for realizing a digital thread framework. The use case to be demonstrated consists of developing a DT mimicking the behavior of a given soil parcel. Its constituent digital resources are depicted in Figure 35, while, the different tasks constituting its development and use life-cycle are showed in Figure 36.

For the proof-of-concept, the Destination Earth middleware must implement a set of digital components and services, namely:

- **Knowledge Base**: it manages the formalization and the relationships of a set of data and model (analytical software) concepts/entities (Nativi, Santoro, Giuliani,, & Mazzetti, 2019).
- **Internet services orchestrator**: it orchestrates online accessible services, implementing workflows made up of data and analytical software resources –making use of brokering and mediation services where required (Santoro, Mazzetti, & Nativi, 2020), (Santoro, Nativi, &

Mazzetti, Contributing to the GEO Model Web implementation: A brokering service for business processes, 2016).

- **DT builder**: to work out a DT *Template* from a DT *Blueprint* that is already defined in the Knowledge Base. In the proof-of-concept case, it will describe: a ML architecture, a set of input streams, and the associated ground truth. Moreover, the DT builder is in charge of training the DT *template* and generating the DT instance. In doing that, it makes use of the services provided by the orchestrator. Finally, the DT builder manages the DT instance, finalizing any users' simulations.

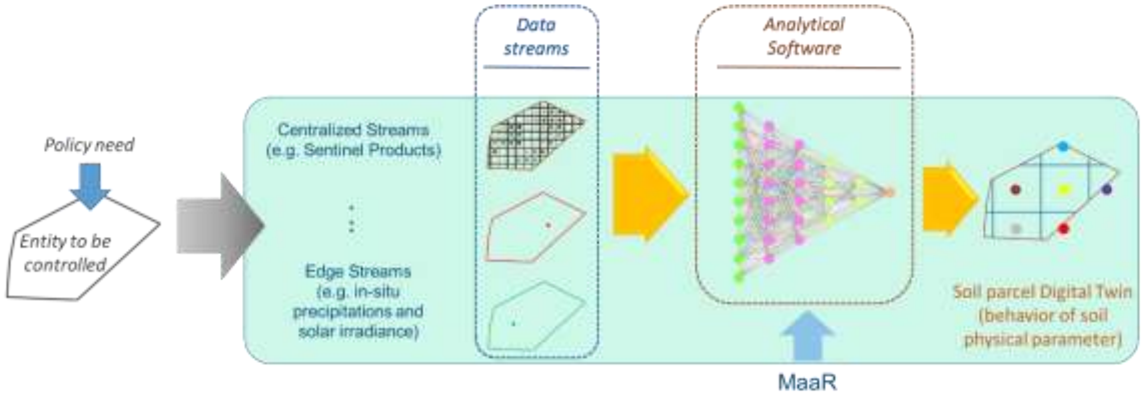


Figure 35. DT mimicking the behavior of a given soil parcel

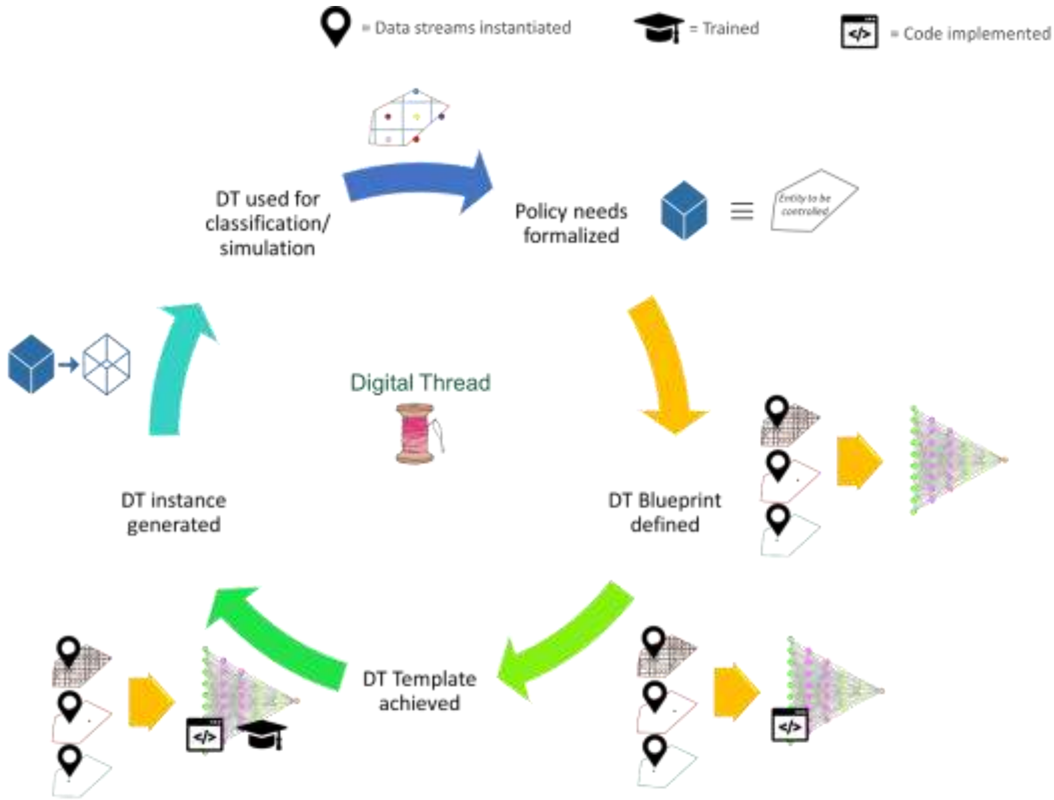


Figure 36. The digital thread framework implemented by the Destination Earth proof-of-concept

7 OPPORTUNITIES AND SOLUTIONS

The starting point for Destination Earth is a number of existing assets at both EU-level, in organisations such as ESA, EUMETSAT, ECMWF, the EEA, Mercator Ocean, HPC Centres, European cloud platforms (e.g. DIASs), GEANT and at the national level (holding geospatial, earth observation and environmental, meteorological information to be opened up through the implementing act under the Open Data Directive). There are also close links with the Copernicus Services, which are working on products in relevant thematic areas (land, marine, atmosphere, climate, emergency, security). For the preparation of this document, we engaged with some of these organizations starting to recognize opportunities and existing technological solutions.

Destination Earth promises to be a future core European contribution to GEOSS, complementing the already strong contribution made by Copernicus and INSPIRE, and consider the enabling role that Destination Earth common platform could play for EuroGEO initiative.

The Destination Earth Stakeholders, engaged in this initial phase, provided 30 scenarios (i.e. use cases), which were introduced and analysed in a separate document (Nativi & Craglia, 2020). Moreover, it was carried out a first survey of the existing and next coming projects and initiatives dealing with Earth Digital Twins (Nativi, Delipetrev, & Craglia, Destination Earth: Survey on “Digital Twins” technologies and activities, in the Green Deal area, 2020).

References

Accenture Consulting. (2017). *The Digital Thread Imperative.* Retrieved from Accenture Consulting: https://www.accenture.com/_acnmedia/PDF-67/Accenture-Digital-Thread-Aerospace-And-Defense.pdf

Ashby, R. W. (1957). *An Introduction to Cybernetics.* London: Chapman & Hall Ltd.

Baresi, L., Mendonça, D. F., Garriga, M., Guinea, S., & Quattrocchi, G. (2019). A Unified Model for the Mobile-Edge-Cloud Continuum. *ACM Trans. Internet Technol.*, 1-21.

Beer, S. (1959). *Cybernetics and Management.* London: The English Universities Press Ltd.

Beer, S. (1981). *Brain of the Firm - Second Edition.* John Wiley & Sons.

Bittencourt, L., Immich, R., Sakellariou, R., Fonseca, N., Madeira, E., Curado, M., . . . Rana, O. (2018). The Internet of Things, Fog and Cloud continuum: Integration and challenges. *Internet of Things*, 134-155.

DiMario, M. J., Boardman, J. T., & and Sauser, B. J. (2009). System of Systems Collaborative Formation. *IEEE SYSTEMS JOURNAL, 3*(3), 360-368.

Djavanshir, G. R., Khorramshahgol, R., & Novitzki, J. (2009). Characteristics of Metasystems: Toward Defining Metasystems' Governance Mechanism. *IEEE IT Professional, May/June*, 46-49.

Faction. (2020, Dec 05). *WHAT IS MULTI-CLOUD? EVERYTHING YOU NEED TO KNOW.* Retrieved from Faction: https://www.factioninc.com/blog/what-is-multi-cloud/

Flexera. (2020). *2020 State of the Cloud Report.* Retrieved from Flexera: https://resources.flexera.com/web/pdf/report-state-of-the-cloud-2020.pdf?elqTrackId=8e710f56b1c44fe39130b0d6168944ed&elqaid=5770&elqat=2

Flood, R. L., & Carson, E. (2013). *Dealing with Complexity: An Introduction to the Theory and Application of Systems Science.* Springer Science and Business Media LLC.

Guo, H., Nativi, S., Liang, D., Craglia, M., Wang, L., Schade, S., . . . Annoni, A. (2020). Big Earth Data science: an information framework for a sustainable planet. *International Journal of Digital Earth*.

Harper, J. (2020, Oct 19). *2021 Trends in Cloud Computing: The Omni Present Multi-Cloud Phenomenon.* Retrieved from Inside BigData: https://insidebigdata.com/2020/10/19/2021-trends-in-cloud-computing-the-omni-present-multi-cloud-phenomenon/

Hong, J., Dreibholz, T., Schenkel, J., & Hu, J. (2019). An Overview of Multi-Cloud Computing. In L. Barolli, M. Takizawa, F. Xhafa, & T. Enokido, *Web, Artificial Intelligence, and Networks applications* (pp. 1055-1068). Springer.

IBM. (2019, Sep 05). *Multicloud.* Retrieved from IBM: https://www.ibm.com/cloud/learn/multicloud#:~:text=Multicloud%20is%20the%20use%20of,%2C%20PaaS%2C%20or%20SaaS).&text=With%20multicloud%2C%20you%20can%20decide,based%20on%20your%20unique%20requirements.

Imran, H., Wazir, S., Ikram, A., Ikram, A., Ullah, H., & Ehsan, M. (2020, Mar 01). *HPC as a Service: A naive model.* Retrieved from Cornell University: https://arxiv.org/abs/2003.00465

Jain, A. (2016, Sep 17). *The 5 V's of big data.* Retrieved from IBM Watson Health Perspectives: https://www.ibm.com/blogs/watson-health/the-5-vs-of-big-data/

Klir, G. J., & Elias, D. (2003). *Architecture of Systems Problem Solving.* New York: Springer Science+Business Media.

Miskinis, C. (2018, Nov). *What does a digital thread mean and how it differs from digital twin.* Retrieved from Challenge Advisory: https://www.challenge.org/insights/digital-twin-and-digital-thread/

Nativi, S., & Bigagli, L. (2009). Discovery, mediation, and access services for earth observation data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 233-240.

Nativi, S., & Craglia, M. (2020). *Destination Earth: Use Cases Analysis.* Luxemburg: European Commission - JRC.

Nativi, S., Delipetrev, B., & Craglia, M. (2020). *Destination Earth: Survey on "Digital Twins" technologies and activities, in the Green Deal area.* Luxembourgh: European Commission.

Nativi, S., Mazzetti, P., & Craglia, M. (2020). Data Ecosystems for our changing Planet: a system-of-systems approach. *Submitted to ISDE journal*.

Nativi, S., Mazzetti, P., & Geller, G. (2013). Environmental model access and interoperability: The GEO Model Web initiative. *Environmental Modelling & Software*, 214-228.

Nativi, S., Mazzetti, P., Santoro, M., Papeschi, F., Craglia, M., & Ochiai, O. (2015). Big Data challenges in building the Global Earth Observation System of Systems. *Environmental Modelling & Software*, 1-26.

Nativi, S., Santoro, M., G. G., & Mazzetti, P. (2019). Towards a knowledge base to support global change policy goals. *International Journal of Digital Earth*, 1-29.

Nativi, S., Spadaro, N., Pogorzelska, K., & Craglia, M. (under publication). *DIAS assemment.* Luxembourgh: European Commission.

Rouse, M. (2020, Dec 28). *HPCaaS (High-Performance Computing as a Service).* Retrieved from WhatIs.com: https://whatis.techtarget.com/definition/HPCaaS-high-performance-computing-as-a-service

Santoro, M., Mazzetti, P., & Nativi, S. (2020). The VLab Framework: An Orchestrator Component to Support Data to Knowledge Transition. *Remote Sensing*, 1-23.

Santoro, M., Nativi, S., & Mazzetti, P. (2016). Contributing to the GEO Model Web implementation: A brokering service for business processes. *Environmental Modelling & Software*, 18-34.

Sivarajah, U., Mustafa Kamal, M., Irani, Z., & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research*, 263-286.

TechRepublic staff. (2020, Dec 24). *Multicloud: A cheat sheet.* Retrieved from TechRepublic: https://www.techrepublic.com/article/multicloud-the-smart-persons-guide/

Tozzi, C. (2019, Nov 25). *Multicloud Architecture: 3 Common Performance Challenges and Solutions.* Retrieved from ITPro Today: https://www.itprotoday.com/performance-management/multicloud-architecture-3-common-performance-challenges-and-solutions

Turchin, V. F. (1995). A dialogue on Metasystem transition. *World Futures: The Journal of Global Education, 45*(1-4), 5-57. doi:10.1080/02604027.1995.9972553

Turner, M. J. (2019, Aug). *IDC White Paper.* Retrieved from Automation, Analytics, and Governance Power: file:///C:/Quarantena/Destination%20Earth/Technical%20WG%20(Architecture)/Overall%20Architecture/IDC-whitepaper-multicloud-management.pdf

Vaccari, L., Craglia, M., Fugazza, C., Nativi, S., & Santoro, M. (2012). Integrative Research: The EuroGEOSS Experience. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5*(6), 1603-1611.

Villari, M., Fazio, M., Dustdar, S., Rana, O., & Ranjan, R. (2016). Osmotic Computing: A New Paradigm for Edge/Cloud Integration. *IEEE Cloud Computing*, 76-83.

White, N. (2020, Jan 29). *Digital Twins vs. Digital Threads: defining the concepts.* Retrieved from PTC: https://www.ptc.com/en/blogs/corporate/digital-twin-digital-thread

Yolles, M. (2004). Implications for Beer's system/metasystem dichotomy. *Kybernetes, 33*(3/4), 726-764. doi:10.1108/03684920410523670

List of Figures

Glossary

Architecting	The activities of conceiving, defining, documenting, maintaining, improving, and certifying proper implementation of an architecture [ISO/IEC 42010].
Architectural Description	A collection of information products used to document an architecture [ISO/IEC 42010].
Architecture Description Language (ADL)	An architecture description language (ADL) is any form of expression for use in architecture descriptions. Examples of ADLs include: Rapide, Wright, SysML, AADL, ArchiMate and the viewpoint languages of RM-ODP [ISO 10746].
Architecture Framework	A set of architectural concerns, generic stakeholders, predefined viewpoints, and viewpoint correspondence rules, that have been established to capture a common practice for architectural descriptions in a specific domain or stakeholder community [ISO/IEC 42010].
Architectural Model	A module of an architectural view [ISO/IEC 42010].
Architectural View	A representation of a whole system from the perspective of a related set of architectural concerns [ISO/IEC 42010].
Architectural Viewpoint	The conventions for constructing, interpreting and using an architectural view to frame a prescribed set of concerns for a set of stakeholders [ISO/IEC 42010].
API	Application Programing Interface.
Auto-scaling	In cloud computing environment, it is a method that permits to adjust the allocated resources based on load; usually two of the observed parameter are CPU and network load.
Beta software	Version of software still under development and testing, which is generally not available to all users.
Big Data	Extensive datasets — primarily in the characteristics of volume, variety, velocity, and/or variability — that require a scalable architecture for efficient storage, manipulation, and analysis [ISO/IEC 20546].
Client	The role adopted by an application when it is retrieving and/or rendering resources or resource manifestations [W3C].
Cloud	A global network of servers, each with a unique function. The cloud is not a physical entity, but instead is a vast network of remote servers around the globe, which are hooked together and meant to operate as a single ecosystem [Microsoft Azure].
Cloud Computing	A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned

	and released with minimal management effort or service provider interaction [NIST].
COA	Cloud-Oriented Architecture is another way to call the Internet of Transformation, which derives from the IoT 2.0 revolution.
(Architecture) Concern	An architectural concern is held by one or more stakeholders in the system of interest, and is addressed by an architecture view.
Container	A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another [Docker].
Data slice	A sub set of available data by selecting intervals along one or more dimensions characterizing data itself –e.g. space and time dimensions.
DCA	Distributed Computing Architecture.
Digital Resource	An entity or capability on a network. Anything characterized by a Uniform Resource Indicator (URI) can be a (Web) resource.
Digital system	a group of devices or artificial objects or an organization forming a network especially for distributing something or serving a common purpose [Merrier-Webster vocabulary].
Digital Twin	Digital twin is a digital replica of a living or non-living physical entity. It is a virtual representation of a connected real thing or a set of things representing a complex domain environment. It can be used to represent real-world things/systems that may not be continuously online, or to run simulations of new constructions, applications and services, before they are deployed to the real world [W3C 2019].
GUI	Graphical User Interface
Infrastructure	Basic support services for computing [FOLDOC].
Infrastructure-as-a-Service (IaaS)	A platform supporting the resources needed by other layers. IaaS can be "programmed" (see infrastructure-as-code) by utilizing provisioning tools. Because of this programming interface, even if IaaS is often (but not only) made of "physical" resources, IaaS can be considered as a component [IEEE Software Defined Networks].
Insight	Useful aspect of observed data, made possible by data processing and analytics capabilities. Data can be analyzed in real time, or can be stored for a later batch processing.
Internet of Transformation	It is the next Internet evolution enabled by IoT 2.0 generation. It underpins the Digital Transformation of the Society.
Interoperability	Interoperability is a characteristic of a product or system, whose interfaces are completely understood, to work with other products or systems,

	present or future, in either implementation or access, without any restrictions [AFUL].
IoT	Internet of Things.
Latency time	A measurement of delay in a system, especially the length of time it takes computer information to get from one place to another [Macmillan dictionary].
MaaR	Model-as-a-Resource
MaaS	Model-as-a-Service
MaaT	Model-as-a-Tool
Metasystem	In informatics, a metasystem can be considered as a synonymous of control or management
M2M	Machine-to-Machine
Micro-services	As an architectural framework, micro-services are distributed and loosely coupled, so one team's changes won't break the entire app [Red Hat Software].
Multi-cloud environment	Technological environment where multiple cloud computing and storage services are utilized in a single heterogeneous architecture.
Pattern	An architectural pattern is a general, reusable solution to a commonly occurring problem in software architecture within a given context [Taylor, Medvidović and Dashofy, Software architecture: Foundations, Theory and Practice. Wiley, 2009].
Platform	A specific combination of hardware and operating system and/or other software systems (e.g. databases) under which various smaller application programs can be designed to run. [FOLDOC and Dictionary.com].
Platform-as-a-Service (PaaS)	Systems offering rich environments where to build, deploy, and run applications. PaaS provides infrastructure, storage, database, information, and process as a service, along with well-defined APIs, and services for the management of the running applications, such as dashboards for monitoring and service composition [IEEE Software Defined Networks].
PoC	Proof-of-Concept
Resource	A central element in a model, which represents a physical construct or a logical service, and is further defined by other model entities.
ROA	Resource-Oriented Architecture is a software architecture style.

Scenario	A use case, change case or other interaction between a system stakeholder and the system of interest reflecting one or more architectural concerns [ISO/IEC 42010] –Cf. ISO 15288: Scenarios are used to analyze the operation of the system in its intended environment.
Server	The role adopted by an application when it is supplying resources or resource manifestations [W3C].
SOA	Service-Oriented Architecture is a software architecture style.
Stakeholder (of a system)	An individual, team, or organization (or classes thereof) with interests in, or concerns relative to, a system [ISO/IEC 42010] –Cf. ISO 15288: an individual or organization having an interest in a system or in its possession of characteristics that meet their needs and expectations.
System	A regularly interacting or interdependent group of items forming a unified whole [Merrier-Webster vocabulary].
System architecture	The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them. [Bass, Clements & Kazman, "Software Architecture in Practice", 1998].
System Architectural style	An architectural style defines: a family of systems in terms of a pattern of structural organization; a vocabulary of components and connectors, with constraints on how they can be combined [Shaw & Garlan, "Software Architecture: Perspectives on an Emerging Discipline", 1996].
Thing	A physical entity (e.g. a device or a sensor) to be connected to an infrastructure (e.g. a cloud). A thing can be an intelligent edge device with processors capable of running processes and functions or a low-power device that simply collects information.
(Architecture) View	An architecture view addresses one or more concerns held by the stakeholders, and is composed of one or more architecture models [ISO/IEC/IEEE 42010 Systems and software engineering — Architecture description]
(Architecture) Viewpoint	An architecture viewpoint is in effect a specification for an architecture view [ISO/IEC/IEEE 42010 Systems and software engineering — Architecture description].
Virtual Cloud	The resource abstraction and control layer build virtual cloud resources on top of the underlying physical resource layer and support the service layer where cloud services interfaces are exposed [NIST].
Virtual Machine	The simulated environment resulting from a virtualization process is called a virtual machine (VM). A computer image that behaves like an actual computer [Microsoft Azure].

Virtualization (process)	Virtualization is the simulation of the software and/or hardware upon which other software runs. There are many forms of virtualization, distinguished primarily by computing architecture layer [NIST].
WOA	Web-oriented Architecture style.

KJ-NA-30646-EN-N

The European Commission's science and knowledge service

Joint Research Centre

JRC Mission

As the science and knowledge service of the European Commission, the Joint Research Centre's mission is to support EU policies with independent evidence throughout the whole policy cycle.

EU Science Hub
ec.europa.eu/jrc

@EU_ScienceHub

EU Science Hub - Joint Research Centre

EU Science, Research and Innovation

EU Science Hub

Publications Office of the European Union

doi:10.2760/08093

ISBN 978-92-76-32434-8