NebulOuS: A Meta-Operating System with Cloud Continuum Brokerage Capabilities

Yiannis Verginadis[*,†], Christos-Alexandros Sarros[§], Mario Reyes de Los Mozos[‡], Simeon Veloudis[ǂ], Radosław Piliszek[*,#], Nicolas Kourtellis[⊥], Geir Horn[¤]

[*]*Athens University of Economics and Business,* Greece, [†]*Institute of Communications and Computer Systems,* Greece, [§]*Ubitech*, Greece, [‡]*Eurecat Technological Centre*, Spain, [ǂ]*South-East European Research Centre,* Greece, [*]*University of Białystok*, Poland, [#]7Bulls.com, Poland, [⊥]*Telefonica Research*, Spain, [¤]*University of Oslo*, Norway
Emails: [*,†]jverg@aueb.gr, [§]asarros@ubitech.eu, [‡]mario.reyes@eurecat.org, [ǂ]sveloudis@seerc.org, [*,#]r.piliszek@uwb.edu.pl, [⊥]nicolas.kourtellis@telefonica.com, [¤]Geir.Horn@mn.uio.no

***Abstract*— Cloud Continuum is the paradigm that unifies and exploits resources from far edge to public and private cloud offerings, as well as processing nodes with significant capacity in between. Nowadays, the combination of all these resources for augmenting modern hyper-distributed applications becomes a necessity, especially considering the vast volumes of data, their velocity, and their variety, which constitute well known challenges of Big Data processing. In this paper, we address the main research question on how a Cloud Continuum management platform should be structured to cope with the constantly increasing challenges and opportunities of the domain. We introduce the NebulOuS architecture vision towards accomplishing substantial research contributions in the realms of Cloud Continuum brokerage. We propose an advanced architecture that enables secure and optimal application provisioning, as well as reconfiguration over the Cloud Continuum. NebulOuS introduces a novel Meta-Operating System and platform, that is currently being developed, for enabling transient Cloud Continuum brokerage ecosystems that seamlessly exploit edge and fog nodes, in conjunction with multi-cloud resources, to cope with the requirements posed by low latency applications.**

Keywords—Cloud Continuum brokerage, cloud meta-OS,

I. INTRODUCTION

The exponential increase in volume and velocity of data produced by a constantly increasing number of devices (data sources), has led modern enterprises to urgently look for advanced multi-cloud offerings that can efficiently address the dynamically changing needs of their data- and processing-intensive distributed applications. Nevertheless, there are still several significant challenges. Even in advanced efforts [1, 2, 3] that pursue to combine hybrid cloud offerings and exploit the potential of the Cloud Continuum, significant issues remain unresolved. These are mainly related to the need of optimising network traffic and latency between processing jobs and data sources to respect Quality of Service (QoS) guarantees, as well as establishing security and trust. Furthermore, in today's digital era, where data is the main commodity for creating value in data-driven business ecosystems, multi- or cross-cloud solutions alone may be insufficient for satisfying the requirements associated with data- and processing-intensive applications. We witness a constant increase of connected devices that produce data at an astonishing rate of 2.5 exabytes per day [4] whilst, according to Cisco [5], the Internet of Things (IoT) will feature more than 14 billion connected devices worldwide by the end of 2023 [5], and by the end of 2025, more than 75% of enterprise-generated data will be created outside data centres [6]. Such unprecedented amounts of heterogeneous data and their propagation among diverse processing and storage architectures, amplifies the need for solutions that can cost-effectively host and process this data.

In recent years, edge computing has emerged as a panacea for coping with such challenges [7]. As low latency and location-sensitive applications continue to loom (e.g., precision agriculture, industry 4.0 maintenance applications, autonomous vehicles, AR-based remote patient treatment [8], etc.), the distant cloud will not be able to follow the dynamically changing requirements of these hyper-distributed applications, struggling to cope with the sheer magnitude of the data that they produce. In addition, certain application components are, by default, restricted from transmitting data to remote cloud resources due to organisation-wide security, legal and privacy concerns. To cope with such challenges, a flexible processing paradigm is needed to advocate data processing on adequate resources, hosted in proximity to data sources [2]. The importance of placing processing jobs as close as possible to the data sources is highlighted by the recent launch of Amazon's AWS Wavelength that allows developers to deploy, in a pre-defined way, application parts that require ultra-low latency to Wavelength Zones that embed AWS compute and storage services in telecommunication provider data centres (located in metropolitan aggregation sites of 5G networks). Moreover, telco operators have recently announced the Open Gateway API initiative [9], which aims to provide cloud developers with a universal, open means to access their networks and exploit the available computing capacity at the edge. Nevertheless, further research is required, that goes beyond such propositions, e.g., by investigating the secure enablement of computing paradigms (such as Function-as-a-Service / FaaS) over heterogeneous topologies, thereby providing generic solutions that are agnostic to any particular cloud provider or telco operator technologies and infrastructures. We argue that similar to how an Operating System (OS) efficiently interoperates with a wide range of heterogeneous hardware resources, and akin to how the rapid adoption of cloud computing led to the need of cloud service brokerage (CSB) [10], the true exploitation of the Cloud Continuum inevitably necessitates the introduction of holistic cloud and fog brokerage capabilities.

In this paper, we reflect on the core functionality that is further needed for adequately harnessing Cloud Continuum resources. Thus, we address the main research question *how a Cloud Continuum management platform should be structured in order to offer the capabilities outlined abov*e. To this end, we introduce NebulOuS: a novel platform that aspires to facilitate the formation of an adaptive application hosting environment, that spans the Cloud Continuum, it is driven by the need of ensuring proximity between processing and data sources, and it is optimised based on the data involved, the defined QoS requirements, and any applicable regulatory, security, or privacy constraints. In this respect, NebulOuS shall offer a meta-Operating System (meta-OS) that enables the emergence of ad-hoc fog brokerage ecosystems that exploit IoT/edge and fog nodes, in parallel to multi-cloud resources, in order to cope with the requirements of hyper distributed applications.

The rest of the paper is organised as follows. In Section II, we discuss relevant state-of-the-art. In Section III, we sketch the main aspects of the NebulOuS approach, while providing initial details on the core components of its architecture. In Section IV, we discuss the next steps of this work.

II. Related Work

We present three different, albeit closely inter-linked, strands of related work: i) cloud and fog brokerage solutions; ii) cross-cloud and fog application deployment; and iii) self-adaptive and proactive reconfiguration in the Cloud.

Cloud and fog brokerage. The works in [1,10] aim at identifying optimal cloud services, from a cloud consumer perspective, by considering user preferences, but limited to quantitative metrics. Nevertheless, they do not attempt to address the apparent multi-criteria optimisation problem that also involves qualitative aspects [10]. Furthermore, recent domain surveys [11, 12] identify the following key challenges and future research streams in cloud brokerage: i) adaptive and fluid application deployment; ii) intelligent decision-making; iii) proactive resource management; iv) governance and QoS; v) ability to solve high-dimensional problems; vi) reducing latency based on fog computing; vii) discovering and combining cloud services to meet complex needs; and viii) ability to deal with cloud provider failures. The proposed NebulOus architecture shall tackle these challenges by introducing cloud and fog brokerage capabilities analogous to the intermediation capabilities that OSs offer. Specifically, it will support Cloud Continuum providers (organisational units, telecom providers, etc.) and consumers in advertising available resources for ad-hoc formulation of local micro-continuums, and application provisioning requirements and preferences, in a way that retains their inherent vagueness (e.g., using linguistic terms). Furthermore, NebulOuS will be able to match, on demand, the above aspects using a multi-criteria decision making (MCDM) approach that handles precise and imprecise criteria to broker the ad-hoc generation of micro local cloud continuums.

Cross-cloud and fog application deployment. Main application orchestration challenges stem from the heterogeneity of resources, the dynamic resource offerings, and the dynamic workload patterns [13]. To address these challenges, the work by Wen et al. [14] proposes holistic cross-layer optimisation with an orchestrator maintaining an entire view of the system, whereas FogAtlas [7,15] offers a software platform and architectural framework for orchestrating microservice container workloads in fog environments using OpenStack and Kubernetes. NebulOuS will reinforce support for edge and fog infrastructures by introducing novel mechanisms for controlling edge devices and allowing aggregation of resources over Secure Shell (SSH), while enabling communication with devices regardless of their architecture and operating system. This will create a unified resource pool over which novel orchestration concepts for highly distributed applications, including workflow orchestration will be founded. In this respect, NebulOuS will attempt to leverage technologies such as FIWARE FogFlow [16] or Eclipse Zenoh [17] to extend the ways context management and data processing are addressed in a distributed fashion, from the IoT device to the datacenter. With respect to context management, NebulOuS will provide extensions in IoT discovery mechanisms and enable the efficient and dynamic propagation of context information.

Self-adaptive and proactive reconfiguration technologies. Existing work focuses on the analysis of application/workload behaviour, future workload predictions, and workload generation. Herbst et al. [8] conducted a survey of works using statistical analysis and machine learning methods in workload modelling and prediction. Ba, Lorenzi and Ploennigs [18] presented an approach, based on Graph Convolutional Neural Networks, that is capable of improving modelling and monitoring at the edge. Nevertheless, most relevant works fail to address self-adaptation capabilities in the Cloud Continuum as they are unable to cope with edge cases in which proactive reconfigurations potentially contradict reactive ones. Such challenges have been also recognized in the recent work of Dustdar Pujol and Donta [19]. The authors propose a methodology to generate proper representations for cloud continuum systems based on the Markov Blanket concept [19] and offer the concept of equilibrium, leaving behind the use of thresholds. This notion of the equilibrium aims at alerting the system in advance so that the adaptations enforced become more efficient. NebulOuS, on the other hand, shall keep the use of thresholds (not always predefined through the use of AI-driven anomaly detection engine) but it will limit the variability space of adaptation decisions through cloud and fog brokerage. NebulOuS will tackle complexity issues by providing tangible improvements in understanding and predicting workload behaviour of complex fog applications. The tools NebulOuS will develop for workload forecasting will support advanced application reconfiguration and remediation scenarios and will provide a sound basis for reasoning about the proactive management of workloads as well as about the optimization of application deployments. The abstract behavioural specification language for designing executable models of distributed systems will be used to create and maintain automatically a "Digital Twin" parallel simulation model of the running application supporting the proactive application adaptations and reconfigurations.

III. NebulOuS Approach

In this work, we introduce NebulOuS, a Cloud Continuum brokerage solution that aims to replace the monolithic public cloud paradigm with resources drawn from a transient mesh of interconnected, interoperable and heterogeneous micro local clouds, private enterprise clouds, public clouds as well as edge resources. The exact candidate resources that may consolidate this mesh will be filtered and aggregated according to each application's placement requirements (e.g., organizations involved, location of data sources, data velocity and volume aspect, physical/network distance among edge and cloud resources, legal and pricing issues, etc.). Therefore, one of our main directions of work is to develop a novel brokerage platform that will enable any enterprise, SME or public organization to become both a cloud and fog service provider, as well as a cloud and fog service consumer of resources that may span the limited organisational boundaries (Figure 1). This implies the ad-hoc generation of "virtual organisations" that involve partners willing to share and/or consume heterogeneous cloud resources according to

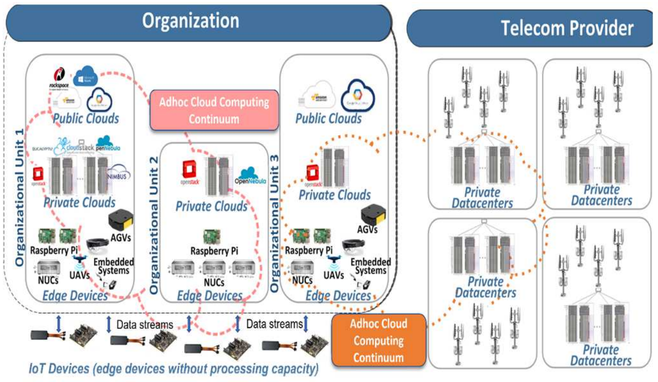

Fig. 1 Brokerage in Cloud Computing Continuum

blockchain-enabled SLAs and strict security enforcement guarantees for optimized usage of any available cloud or fog computing resources. From a business point of view this corresponds to the "virtualisation" of exploitable infrastructural assets that can be shared across departments of the same or different organisations.

The NebulOuS approach will offer capabilities along the following dimensions: i) facilitate the use of semantic technologies, as well as security policies and requirements stemming from pertinent legal and regulatory frameworks (i.e. providing models for: resource description, security overlay network and SLA aspects); ii) enable users to discover infrastructural resources among sets of heterogeneous and geographically-dispersed micro local clouds, private clouds and available edge resources, while augmenting them (if needed) with public resources in bursting scenarios; iii) enable the definition of appropriate SLAs and facilitating their interpretation into smart-contracts; iv) enable the allocation of application components, data and workflow tasks to the discovered resources by taking into account several factors such as the proximity of the resources to the data-generating IoT devices, their current health status, etc.; v) enable the effective orchestration of applications and workflows running on heterogeneous resources; vi) design, deploy and maintain a flexible monitoring and adaptation system of cloud applications to sustain their unobstructed operation according to the defined SLAs; vii) enable secure access to these resources by taking into account user security requirements, as well as any pertinent national and international legal and regulatory frameworks.

The NebulOuS approach will advocate a model-driven engineering (MDE) approach, whereby a user first describes an application and the required resources using a well-defined application model. This model can be expressed either using YAML constructs (as is the case in KubeVela - https://kubevela.io/), or specific modelling languages, e.g., the Topology and Orchestration Specification for Cloud Applications (TOSCA) [20], or the Cloud Application Modelling and Execution Language (CAMEL) [21, 22]. In the following section, we present in detail the main capabilities of the NebulOuS meta-OS.

A. *NebulOuS Architecure*

Before delving into the actual architecture, we first outline different kinds of NebulOuS users we envision

- Regular users who assume the role of Cloud Continuum consumers and may use NebulOuS to deploy workloads in the cloud-to-edge continuum (typically software developers or DevOps). Regular users provide NebulOuS with a service description, along with all related requirements and constraints (e.g., the hardware architecture that the service can be deployed on). They also declare their preferences regarding service brokerage in the form of requests handled by the NebulOuS meta-OS.
- Admin users, who deploy NebulOuS on their premises and advertise available resources to be brokered. Admin users may be telecom providers staff, who onboard compute resources from their edge locations, organizational units staff that make their private resources available for intra- or inter-organisational infrastructure sharing, etc. Admin users provide NebulOuS with policies (e.g., access control) to be enforced by the NebulOuS meta-OS.

Both kinds of actors use NebulOuS as an intermediary between computing resources and software, similar to a typical OS. They interact with the platform via the NebulOuS User Interface (UI), including a GUI providing information about the state of the deployed services and available resources. NebulOuS shall consider various tools for the observability, telemetry and monitoring aspects and provide visualizations of the relevant data to the users using a dashboard-like approach.

Figure 2 provides an overview of the different components comprising the NebulOuS platform, and the main communication flows between them. At the center of the NebulOuS meta-OS lies the Control Plane that comprises the following modules: the Optimizer, the Deployment Manager, the Overlay Network Manager and the Execution Adapter. The Control Plane is responsible for service deployment, orchestration, optimization and real-time reconfiguration across the entire cloud-to-edge continuum. Based on the application design time requirements, it deploys service graphs on available cloud/edge resources and continuously monitors execution (triggering runtime reconfigurations, if needed) to ensure that services operate as required. This involves:

- The evaluation of the severity of situations forwarded to it by the AI-driven anomaly detection mechanism, or the Event Management system (EMS), and triggering of reconfiguration action(s) in the form of application-level or platform-level adaptations.
- The dynamic optimization of the application placement, so that all applications meet their performance targets, and the infrastructure is not over-subscribed.
- The real-time scheduling of functions and workflow tasks, using traditional and AI-based scheduling approaches.

NebulOuS employs a layered approach to orchestration, assuming it is performed on two different levels: a local one and on a global one. Local orchestration (or L1) is performed by a service orchestrator which manages the deployment and execution of a distributed application within a single cluster, which will typically span a single resource provider (e.g., cloud provider). In the case of containerized workloads, Kubernetes may play the role of the local service orchestrator. However, NebulOuS as a Meta-OS aims to unify resource management spanning different resource providers under a single control plane, aiming to globally optimize application placement on a multi-cloud, multi-cluster setting from the cloud to the edge. Such a setting may consist of several local orchestrators being deployed in the different cloud providers. As a result, a global layer of orchestration is also needed (L2).

Layer-2 orchestration functionality is provided by the NebulOuS logically-centralized Control Plane that intermediates between the various infrastructure and application resources located in different clusters and optimizes their placement on a global level. As such, it is envisioned that although the two different levels will have a certain degree of autonomy in their operation, the global NebulOuS Control Plane will be able to override local orchestrator decisions, if needed. This approach allows L2 orchestration to achieve globally-optimal setups when available, while L1 can still operate locally even if communication to the global control plane is inhibited. Such an approach aims to realize the Fog Computing paradigm by combining the performance benefits of centralized approaches (i.e., the cloud) with the resiliency offered by distributed/decentralized ones (i.e., the edge) [23].

NebulOuS also offers agents that are deployed on compute resources managed by the NebulOuS Meta-OS. Agents enable information exchange with the Control Plane, including control signals, based on which the latter monitors, orchestrates, and continuously optimises application execution. Several other components intermediate between the NebulOuS agents and the control plane, implementing various functionalities such as Data Collection and Management, Event Management, Anomaly Detection, blockhain-based SLAs, and Cloud/Fog servive brokerage. These components are further discussed in Section III.B.

Communication in NebulOuS Meta-OS is mainly concerned with: connectivity establishment between infrastructure resources, and data management over the entire continuum. Two main components are introduced to realize these functionalities: the Overlay Network Manager and the Data Collection and Management component. The former is used for creating tunnels between the different physical and virtual devices. The latter is used for capturing raw IoT data streams and managing data flow details; it also handles the exchange of control plane information between the distributed agents and the orchestration components. The data plane is established using current virtual networking approaches, focusing on programmable and software-defined networks.

B. Core Functionalities

In this subsection, we take the reader through the basic details of the NebulOuS core functionalities.

1) User Interface (UI)

The User Interface component will be the main point of entry for end users and system administrators for interacting with NebulOuS. It will provide a user-friendly interface that allows the interaction with the system in an intuitive manner, reducing the learning curve and making it easier to understand and use the capabilities provided by the Meta-OS. As the process of deploying and managing applications can be complex and technical, a GUI will help simplify the experience for end-users, by providing tools that will enable the definition of their workloads and respective requirements, the management of the deployed services, as well as visualization of metrics and alerts that will be deducted through the monitoring process.

2) Cloud/Fog Service Broker

The Cloud & Fog Service Brokerage (CFSB) will be an important aspect of the NebulOuS meta-operating system for coping with the massive number of resources that can be considered in Cloud Continuum deployments. One of the aims of this mechanism is to reduce the variability space when considering Cloud Continuum resources for hosting application component instances. NebulOuS CFSB will introduce brokerage capabilities, beyond the state-of-the-art,

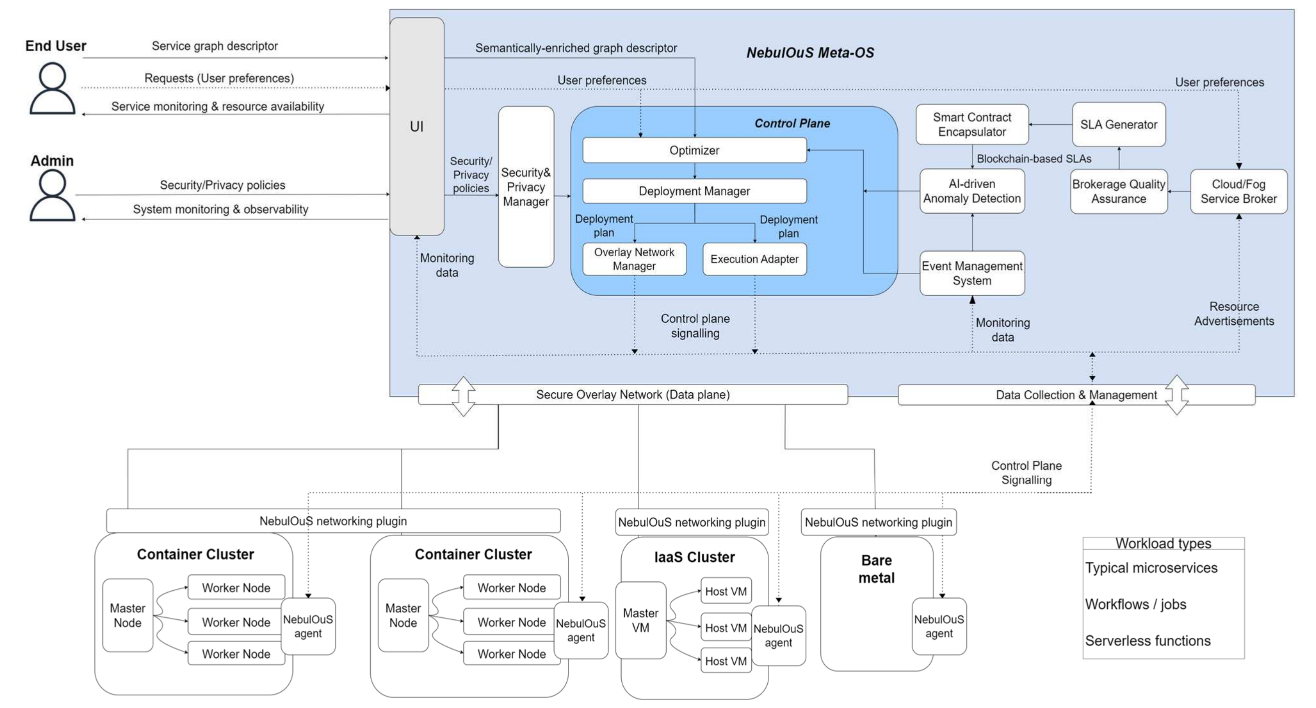

Fig. 2. NebulOuS Conceptual Architecture

in a similar manner that OS intermediates between computing resources and software. Specifically, it will support Cloud Continuum providers and owners (organisational units, telecom providers, etc.) to advertise the available hosting resources (Figure 1). These advertisements will combine quantitative (e.g., CPU cores) and qualitative characteristics (e.g., reputation) of their offerings (i.e., cloud, fog, edge resources that are willing to share across organisation units) that will be used for the ad-hoc formulation of local continuums according to the prerequisites of the applications. Without such brokerage capabilities a multicloud management platform's solvers would be difficult to continuously maintain an optimal application deployment solution, as the cloud continuum dynamically changes (i.e., new resources appear or become unavailable).

Furthermore, a graphical user interface (GUI) will be offered for application owners or DevOps to declare their preferences in a more loose and realistic way that retains the inherent vagueness in human preference expressions (e.g., using linguistic terms). Based on this information NebulOuS will be able to match, on demand, the preferences, and resources advertisements, based on a multi-criteria decision-making (MCDM) approach that copes with precise (related to quantitative resources characteristics) and imprecise (related to qualitative aspects) criteria to enable the ad-hoc generation of micro local cloud continuums.

Therefore, NebulOuS CFSB will research and develop the following functionalities:

- announcement of infrastructural resources and services that can be made available in intra or inter-organisational collaborative infrastructure sharing scenarios;
- declaration of application-related preferences in a way that permits the expression of quantitative and qualitative aspects;
- consideration of multiple precise and imprecise criteria for ranking cloud and fog services to recommend which subsets of them should formulate the required local computing continuums, involving cloud, flog and edge resources.

3) Brokerage Quality Assurance

An important characteristic of NebulOuS will be its ability to ensure the quality of the brokerage services that it will offer. To this end, the Brokerage Quality Assurance (BQA) mechanism is introduced.

The BQA mechanism assures brokerage quality by ensuring abidance of application provisioning requirements and preferences with organisational policies: higher-level requirements that reflect an organisation's business and/or security standpoint with respect to application provisioning. As an example, we may consider an application, call it application *A1*, that samples intermediate-frequency signals emitted by a radar. An organisational policy may articulate that application *A1* belongs to a class of applications that must always be provisioned over compute nodes that feature a base clock rate of at least 2GHz (e.g., to avoid aliasing effects); the policy may also maintain that these nodes should not be under the control of certain organisations (which are considered untrusted). Such an organisational policy clearly places constraints on the way application *A1* may be provisioned. It is the responsibility of the BQA mechanism to ensure, through adequate ontological reasoning, that such higher-level constraints are indeed respected by application's *A1* lower-level provisioning requirements expressed by instantiating the QoS facet of the NebulOuS meta-ontology.

4) Optimizer

A typical NebulOuS application consists fundamentally of a set of containers pre-grouped into microservices where each microservice implements a given transformation of input data to output data. The role of the optimizer is threefold:

1. To decide on the placement of the microservices ranging from the core cloud data centres to the edge devices.
2. To decide on the multiplicity of microservices and duplicate microservices that are bottlenecks in the application.
3. To provide the application with more resources with the right capabilities in the right location to execute the application's microservices.

A rational decision will maximise the decisions makers utility [24], and it is therefore mandatory that the optimizer has a way to assess the utility seen from each application owner of a particular application configuration prior to enacting this configuration. The best way to represent the application owner's utility is arguably through a representative utility function [25]. This will allow the optimiser to find the combined set of resources and application configuration that maximises the application's utility.

A workflow application has temporal constraints among its microservices. This makes the optimisation problem more difficult as the temporal constraints impose an ordering of the microservices, and consequently temporal bounds on the resources needed. Workflow applications are typically scheduled with a Critical Path (CP) heuristic [26], and the partial critical path (PCP) approach can be used to meet and distribute a given execution deadline over the microservices [27]. Serverless functions can be treated differently since they do not suffer from initialisation lags, and so they can be deployed almost as of need. If the same application workflow is repeatedly deployed, reinforcement learning can be used to further shorten the scheduling time, making the serverless function deployment almost instantaneous.

5) Deployment Manager

NebulOuS platform optimizes the deployment of the application based on an array of available infrastructures that range from bare-metal, public and private cloud virtual machines, container orchestration engines, edge and fog nodes, and distributed serverless platforms. This aggregation of different infrastructures requires NebulOuS to cope with the particular way to communicate decisions to each one of them. This design complexity can be eased through separation of concerns where the decision-making component selects the target infrastructure and the action and then passes it to a deployment controller that translates the decision according to the targeted infrastructure. In other words, the deployment manager constitutes a single point of contact in front of all the underlying infrastructure by receiving unified instructions to execute an action. Among others, it will support:

- Adding/deleting/updating an infrastructure;
- Scaling up or down the application resources;

- Reconfiguring the application using a completely new deployment strategy;
- Connecting bare-metal and edge nodes using SSH connections;
- Initializing the nodes with startup scripts;
- Creating a database of available node candidates that an application can use, including the candidate's hardware, image and location;
- Deploying serverless applications;
- Deploying overlay networks;
- Orchestrating containers.

6) Execution Adapter

The Execution Adapter is the component that is responsible for adapting the execution plan to the available infrastructure resources, i.e., managing the resource pool. It will be able to run workflow tasks in multiple languages and multiple environments. It will provide the capability of deploying heterogeneous resources on different public and private cloud, edge nodes, and bare metal. Furthermore, Execution Adapter will provide application components lifecycle management for all components of the application regardless of where the component is deployed (cloud, edge, IoT device). Is able to start components and stop components and react to monitoring alerts (if component requires restarting for example). In addition, it will provide the capability to deploy a serverless platform and call the serverless platform’s interfaces to deploy and invoke functions, following the FaaS paradigm.

7) Overlay Network Manager

The role of this component is to create a secure overlay network which interconnects compute resources to support the multi-cloud and cloud to edge vision of our approach. It receives a deployment plan from the Deployment Manager and, based on that, automatically creates the corresponding network connections between the endpoints.

The Overlay Network Manager implements three main functionalities: First, it establishes connections between the different resources (physical or virtual), enabling component-to-component and service-to-service communication. Second, it ensures that the traffic will be encrypted, and the information being exchanged in the channels is secure. Third, it is responsible for managing the various connections and all related connectivity aspects. Specifically, it will create the overlay network by leveraging the most recent advancements in VPN gateways, network programmability (SDN/NFV) and Linux networking (XPD/eBPF) solutions.

8) Data Collection & Management

To allow the communication between modules of the NebulOuS platform as well as modules of the applications running on top of the platform, the Data Collection and Management module will be employed. Due to the asynchronous and parallel nature of IoT data publishing and its processing, and considering the scalability, which is desired, employing a publish/subscribe mechanism at the core of the module is considered the most appropriate technology. The role of this component is to create a secure, efficient communications channels within the NebulOuS architecture that will utilize a publish/subscribe paradigm to allow:

- platform components to publish and subscribe to event streams;
- IoT devices or application components to publish event streams;
- application components to subscribe to these information streams.

To facilitate the integration of devices/systems that cannot connect directly to the publish/subscribe broker, a proxy to handle the communication with the platform will also be included. Data Collection & Management will also involve dedicated subcomponents that will handle:

- Inter-Component Communication: message-oriented software that will act as an intermediary layer between the sender and receiver of messages, allowing for decoupled and interoperable data transmission and forming a homogenous communication gateway.
- IoT time series data storage: a distributed time series database will be deployed as part of the NebulOuS platform to persist the time-series related data generated by the IoT devices. Using a well-defined query mechanism, other components of the application running on NebulOuS platform will be able to retrieve this data.
- IoT data processing pipelines orchestration: Interoperability aspects of the data collection and management can require the execution of transformation pipelines that include payload encoding conversion, mapping of data structures and conversion of values among others. To ease the development, deployment and management of these transformation pipelines, the NebulOuS platform will include a framework that allows users to declaratively express these data transformation pipelines by identifying their main components (data sources, transformation operations and data consumers) and interlace them to conform to data transformation flows.

9) SLA Generator

NebulOuS’s brokerage capability naturally rests upon its capacity to contract SLAs between resource providers and requesters. To this end the SLA Generator will be responsible for automatic creation of SLA templates. It will be based on a meta-ontology featuring two primary facets:

- The resources facet, that offers adequate support for describing entities in the Cloud Continuum (devices, observations, actuations, people, etc.) and the properties thereof.
- The QoS facet, that offers adequate support for describing application provisioning requirements and preferences.

The main functionalities of the SLA Generator are:

- Ontological modelling of cloud continuum resources and their properties of interest;
- Ontological modelling of QoS requirements and preferences regarding application provisioning;
- Semantic reasoning to determine ability of cloud continuum resources to satisfy QoS requirements;

- Generation of SLA templates specifying QoS requirements and violation penalties.

10) Smart Contract Encapsulator

The Smart Contract Encapsulator (SCE) encapsulates SLAs in a smart contract which features expressivity of semantic models with immutability aspects that blockchain can support. SCE's main functionality is to automatically transform SLAs into smart contract functions, introducing blockchain-enabled SLAs. SCE provides a mechanism for infusing into smart contracts both the SLA rules and the monitoring algorithms used to determine whether these rules are satisfied by application execution. SCE is also in charge of enabling smart contracts to invoke other services and third parties for performing arbitrage when SLAs are violated. It also encompasses a sensing and monitoring loop that regularly feeds the information into the smart contracts. This component has a key role in orchestrating the interactions between smart contracts running on blockchain, sensing loop, and off-chain external services for maintaining SLAs. Specifically, it will:

- enable the automatic transformation of SLAs into smart contract functions;
- infuse the SLA rules and the monitoring algorithms into smart contracts which are used to determine whether these rules are satisfied by application execution;
- offer new interfaces and mechanisms for obtaining feedback from the monitoring algorithms and interacting with off-chain services and other third-party services.

11) Event Management System

The Event Management System (EMS) deploys and maintains the monitoring functionality of NebulOuS on infrastructure nodes, which will observe the necessary QoS metrics both of the platform and of the deployed applications. Based on application description models, EMS will deploy and configure a number of monitoring agents that collect, filter, process and propagate the monitoring metrics values. The collected and computed monitoring metrics are then published in the pub/sub mechanism of the platform. Specifically, EMS will:

- Extract/Deduce monitoring requirements from application description models, as well as from system configuration;
- Deploy monitoring agents and configure them into a network of collaborating nodes that process and propagate, in a context-aware manner, monitoring metrics coming from candidate or already hosting nodes located in the Cloud Continuum;
- Detect SLO violations (specified in Application description models);
- Persist and publish metrics and SLO violation events to platform's pub/sub mechanism;
- Continuously monitor the health status of monitoring agents and allow them to autonomously recover when failures are detected.

12) AI-driven Anomaly Detection

The role of AI-driven anomaly detection engine is to ensure QoS, being able to identify deviations from "normal" behaviour (i.e., low probability events, anomaly) with respect to the network and the IoT devices. Such anomalies can be generated by lack of mobile network coverage, software or network misconfigurations, unexpected user behaviour, intruders, etc. It will consider the real-time processed monitoring data coming from EMS, so this will be accommodated as close to the edge as possible. To facilitate its operation, it will possibly leverage federated learning approaches to: i) decrease response time, ii) reduce network traffic, and iii) increase fault-tolerance in the network as a result of better decisions between autonomous monitoring nodes. To that end, different AI algorithms, from simple ML techniques to advanced immunological-inspired systems, will be considered. We will consider improvements over artificial immunological algorithms (e.g., dendritic cell algorithm – DCA) towards a semi-automatic approach (from the current manual one) for the feature selection and assignment of each selected variable to a type of signal.

13) Security & Privacy Manager

The Security and Privacy Manager is the NebulOuS component that manages the various security and privacy policies defined by admin users. NebulOuS will implement a policy engine for ad-hoc multi-cloud and fog environments. This component lies between the UI and Control Plane; it receives user-defined security and privacy policies by the UI and undertakes all necessary actions to ensure that the policies will be evaluated, deployed and enforced. Security and privacy by-design will be realized by offering a context-aware attribute-based access control (ABAC) mechanism (similar to [28]), to restrict access to deployed application components and data. To achieve the aforementioned security and confidentiality of the services and the provisioned resources, we are going to introduce a mechanism that will facilitate developers in defining access policies for application components that are deployed on cloud, fog and edge resources. Those fine-grained policies will reflect the security requirements that developers are interested in imposing on their applications. Using such an approach, the access rights of the platform and application will be bound to a number of dynamic policies.

14) NebulOuS agents

The NebulOuS agents are software agents that are used to realize a distributed control plane, implementing several functionalities when deployed side-by-side with the (end-user's) application components: First, they enable resource management, setting up the resources, the containers and communicating with the Control Plane components. Second, they are used to collect monitoring data from the services and compute resources, acting as data sources for the Data Collection and Management component and monitoring probes by providing health checks, thereby driving real-time reconfigurations. Third, they can act as processing agents to support complex event processing close to the edge, reducing the need to stream all monitoring data to a centralized monitoring mechanism. Last, they implement policy enforcement in a distributed environment, enabling fine-grained security.

IV. CONCLUSIONS

As part of this work, we considered unaddressed challenges in Cloud Continuum domain and reflected on core functionalities that are still missing. We believe that an adequate and brokered management of the Cloud Continuum power is the answer to the dynamically changing needs of the modern hyper-distributed applications. We introduced NebulOuS which is designed to transparently manage all the layers of the Cloud Continuum stack (in an equivalent manner to an OS managing the available computing capacity) in proximity to data sources or leveraging the virtually limitless scaling capacity of Cloud and Fog resources.

NebulOuS is currently being developed and it will be tested and validated in real pilots related to infrastructure maintenance, freights and resources management, precision agriculture, and crisis management. We focus on these domains for receiving the initial feedback on our novel system, due to the business domains involved, as they heavily rely on edge devices and sensors, and in parallel require intensive data processing capacity that spans different levels of the Cloud Continuum.

Acknowledgment

This work has received funding from the EU's Horizon research and innovation programme under grant agreement No. 101070516 NebulOuS project.

References

[1] V.R. Chintapalli, "A deadline and budget constrained cost and time optimisation algorithm for cloud computing," in Advances in Computing and Communications in Computer and Information Science 193. Berlin, Heidelberg: Springer; 2011. pp 455–62.

[2] S. Mazumdar, D. Seybold, K. Kritikos, Y. Verginadis, "A survey on data storage and placement methodologies for Cloud-Big Data ecosystem," Journal of Big Data, 6(1), 15, 2019, https://doi.org/10.1186/s40537-019-0178-3

[3] M. Różańska, P. Skrzypek, K. Materka, G. Horn, "An Architecture for Autonomous Proactive and Polymorphic Optimization of Cloud Applications," in Barolli, L., Hussain, F., Enokido, T. (eds) Advanced Information Networking and Applications. AINA 2022. Lecture Notes in Networks and Systems, vol 451. Springer, Cham. https://doi.org/10.1007/978-3-030-99619-2_53

[4] Z.U. Rehman, "Towards Multi-criteria Cloud Service Selection," in Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing. Seoul, South Korea: IEEE; 2011. pp 44–48.

[5] Cisco Annual Internet Report (2018–2023), available online at: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html last accessed 2023/02/25

[6] Gartner: What edge computing means for infrastructure and operations leaders, https://www.gartner.com/smarterwithgartner/what-edge-computing-means-for-infrastructure-and-operations-leaders Last accessed 29/01/2023.

[7] D. Santoro, D. Zozin, D. Pizzolli, F. D. Pellegrini and S. Cretti, "Foggy: A Platform for Workload Orchestration in a Fog Computing Environment," in IEEE International Conference on Cloud Computing Technology and Science (CloudCom), Hong Kong, 2017, pp. 231-234. doi: 10.1109/CloudCom.2017.62.

[8] N. Herbst, A. Amin, A. Andrzejak, L. Grunske, S. Kounev, O. J. Mengshoel, P. Sundararajan, "Online workload forecasting," in Self-Aware Computing Systems. Springer, 2017, pp. 529–553.

[9] GSMA Open Gateway APIs, https://www.gsma.com/futurenetworks/gsma-open-gateway-api-descriptions/ Last accessed 04/03/2023.

[10] I. Patiniotakis, Y. Verginadis, and G. Mentzas, "PuLSaR: preference-based cloud service selection for cloud service brokers," J Internet Serv Appl 6, 26, 2015, https://doi.org/10.1186/s13174-015-0042-4

[11] A. Elhabbash, F. Samreen, J. Hadley, and Y. Elkhatib, "Cloud brokerage: A systematic survey," ACM Computing Surveys (CSUR), vol. 51, no. 6, pp. 1–28, 2019.

[12] X. Li, L. Pan, and S. Liu, "A survey of resource provisioning problem in cloud brokers," Journal of Network and Computer Applications, p. 103384, 2022.

[13] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, J. Wilkes, "Borg, omega, and Kubernetes," Communications of the ACM, Vol. 59, N. 5, 2016, pp. 50-57, 10.1145/2890784

[14] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu and M. Rovatsos, "Fog Orchestration for Internet of Things Services," in IEEE Internet Computing, vol. 21, no. 2, pp. 16-24, Mar.-Apr. 2017. doi: 10.1109/MIC.2017.36.

[15] FogAtlas, Drives Applications through the Fog, Available online: https://fogatlas.fbk.eu/ last accessed 2023/02/25

[16] FIWARE FogFlow. Available online: https://fogflow.readthedocs.io/en/latest/ last accessed 2023/02/25

[17] Eclipse Foundation Zenoh, Zero Overhead Network Protocol. Available online: https://zenoh.io/ last accessed 2023/02/25

[18] A. Ba, F. Lorenzi and J. Ploennigs, "Monitoring of IoT Systems at the Edges with Transformer-based Graph Convolutional Neural Networks," 2022 IEEE International Conference on Edge Computing and Communications (EDGE), Barcelona, Spain, 2022, pp. 41-49, doi: 10.1109/EDGE55608.2022.00018.

[19] S. Dustdar, V. C. Pujol and P. K. Donta, "On Distributed Computing Continuum Systems," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 4092-4105, 1 April 2023, doi: 10.1109/TKDE.2022.3142856.

[20] D. Palma and T. Spatzier, "Topology and orchestration specification for cloud applications (tosca)," OASIS, Tech. Rep, 2013, https://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.pdf

[21] A. Rossini, K. Kritikos, N. Nikolov, J. Domaschka, F. Griesinger, D. Seybold, D. Romero, M. Orzechowski, G. Kapitsaki, and A. Achilleos, "The cloud application modelling and execution language (camel)," 2017, http://dx.doi.org/10.18725/OPARU-4339

[22] Y. Verginadis, K., Kritikos, I. Patiniotakis, "Data and Cloud Polymorphic Application Modelling in Multi-clouds and Fog Environments," in La Rosa, M., Sadiq, S., Teniente, E. (eds) Advanced Information Systems Engineering. CAiSE 2021. Lecture Notes in Computer Science(), vol 12751. Springer, Cham. https://doi.org/10.1007/978-3-030-79382-1_27

[23] C.-H. Hong and B. Varghese, 'Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms', ACM Computing Surveys (CSUR), vol. 52, no. 5, pp. 1–37, 2019.

[24] S.-I. T. Tosin and J. R. Gbenga, "Negative selection algorithm based intrusion detection model," in IEEE 20th Mediterranean Electrotechnical Conference (MELECON), 2020, pp. 202–206.

[25] X. Shen, X. Z. Gao, and R. Bie, "Artificial immune networks: Models and applications," in International Journal of Computational Intelligence Systems, vol. 1, no. 2, pp. 168–176, 2008, doi: 10.1109/ICCIAS.2006.294161.

[26] F. Gu, J. Greensmith, and U. Aickelin, "The dendritic cell algorithm for intrusion detection," in Biologically Inspired Networking and Sensing: Algorithms and Architectures, IGI Global, 2012, pp. 84–102, https://doi.org/10.4018/978-1-61350-092-7.ch00

[27] J. Timmis, T. Knight, L. N. de Castro, and E. Hart, "An overview of artificial immune systems," Computation in cells and tissues: Perspectives and tools of thought, pp. 51–91, 2004, doi: 10.1007/978-3-662-06369-9_4.

[28] Y. Verginadis, I. Patiniotakis, P. Gouvas, S. Mantzouratos, S. Veloudis, S. T. Schork, L. Seitz, I. Paraskakis, and G. Mentzas, "Context-Aware Policy Enforcement for PaaS-Enabled Access Control," IEEE Trans. Cloud Comput. 10(1): 276-291, 2022, doi: 10.1109/TCC.2019.2927341.