ALLSTAR: a blockchain based decentralized ecosystem for cloud and edge computing

Huan Zhou*, Xue Ouyang†, and Zhiming Zhao‡
*School of Computer Science, National University of Defense Technology, Changsha, China
†School of Electronic Science, National University of Defense Technology, Changsha, China
‡System and Networking Lab, University of Amsterdam, Amsterdam, the Netherlands
Email: {huanzhou, ouyangxue08}@nudt.edu.cn, z.zhao@uva.nl

***Abstract*—Last decades, Cloud computing has made significant impacts on traditional applications to change their development and operation methods. We witnessed ever more newly-built Clouds and data centers. However, the centralized management mechanism of current Clouds lacks the dispersion to satisfy the requirements of emerging collaborative applications, including AI, IoT, and autopilot. On the other hand, the Edge computing stays at the conceptual and experimental stage. Most organizations construct their own Edge nodes to operate applications. An efficient and incentive mechanism is missing to motivate the Edge and micro Cloud resource providers to join and constitute a more generalized and decentralized ecosystem. To address this issue, we propose ALLSTAR, a blockchain based architecture for equally combining all the Cloud and Edge resources to be seamlessly leveraged by the application in the DevOps (development and operations) lifecycle. The ALLSTAR architecture is a systematic solution to realize the "Cloud+Edge" management and contributes to constructing the corresponding ALLSTAR ecosystem. This paper describes the overall architecture of ALLSTAR, the related key techniques, and detailed application DevOps processes as well as the new business model.**

***Index Terms*—Edge computing, Blockchain, DevOps, Resource management.**

I. Introduction

As Web 3.0 rears into action, decentralization of core services across every facet of online activities is in the need for a powerful, open, data-driven, user-centric, interoperable platform or ecosystem to operate emerging applications [1]. In this context, with the wide adoption of novel paradigms such as Internet of Things (IoT), robotics, and crowdsourcing, ever more applications require operation at a larger scale in both involved users (e.g., cooperative crowd storytelling) and required infrastructure (e.g., mixing networked high-end servers and low-end client devices). Such Next Generation Internet (NGI) collaborative applications exhibiting high business values (e.g., precise service recommendation) and essential societal impacts (e.g., robotics in health care) require critical runtime time constraints (e.g., sensitive latency in immersive virtual realities (VR)), intensive artificial intelligence (AI) (e.g., processing natural languages during social interactions), and strict trust (e.g., using self-made media via social network).

Cloud computing has been a major disruptive technology in the last decade providing resources-as-a-service for diverse Internet applications. According to Gartner's report, it is poised to grow by around 27.5% and expected to reach $1,250 billion by 2025. While Cloud offers elastic capacity and customizable connectivity over a large-scale network, the resilience, sustainability and human-centric collaborative requirements of NGI applications demand an interoperable end-to-end ecosystem design that pushes the infrastructure services, traditionally bounded within big data centers, towards remote nodes closer to the data sources. These concerns are partially addressed by federating Cloud services with emerging Edge and Fog computing paradigms with reduced overheads of transferring distributed data into remote data centers. Since the data can be processed in the Edge or Fog nodes, the pressure of Cloud can be reduced and the application can also retrieve response faster. However, most of the Edge devices are maintained by a specific organization, or application developers and operators construct their own Edge nodes and devices. For example, in the paper [2], the Edge resources are provided by base stations and are owned by the Shanghai Telecom company. Teerapittayanon et al. [3] propose to deploy the neural network across multiple Edge devices, such as sensors and cameras, in a distributed manner. The issue is that these devices must be physically owned by the application developer to control and operate. Hence, missing an effective mechanism to motivate the Edge owner to share the resource hinders the the Edge application developer to further utilize resources distributed around the world.

This paper introduces the ALLSTAR ecosystem and the respective solution, a blockchain based architecture to provide the "Cloud+Edge" environment with a solution for achieving collaborations among Cloud/Edge resource providers and application developers/operators. The goal of this solution is to leverage the blockchain as the underlying support to enhance the trust among the resource providers and application developers when further motivating the Edge device owners to share the resources.

In the rest of this paper, we demonstrate the challenges of utilizing current Cloud and Edge resources to orchestrate NGI collaborative applications as well as related work in Section 2. To address the challenges, we propose the ALLSTAR approach in Section 3 and further detail technical components of the architecture in Section 4. Section 5 presents the business model and the new application DevOps lifecycle between resource providers and application developers/operators when adopting ALLSTAR approach. Finally, Section 6 concludes the paper.

II. Challenges and Related Work

Traditional Clouds are maintained by several well-known providers, such as Amazon, Google. Although their data centers are distributed around the world, the management method within one data center is still centralized. Especially, the distribution of the data centers is still limited. Zooming in to a small region, the resources within one data center are still too centralized to satisfy the application requirements, such as low latency, fast response, etc. Therefore, we illustrate challenges of developing and operating applications within such a complex computing environment as the underlying infrastructure in Figure 1. The green part of the figure represents the state of the art of orchestrating applications with the current Cloud computing infrastructure. When further considering to add the more decentralized Edge resources as the underlying infrastructure, we identified five critical barriers in the current Cloud environments that hinder the development, management and operation of NGI collaborative applications. We demonstrate these barriers by considering a crowd engaged live video and real-time storytelling use-case, the goal of which is for elastic collection and distribution of collaborative, trusted, high-quality live content and story opinions from a massive number of dynamically engaged crowd participants across hostile network channels. This use case is provided by an information broadcast and video encoding company called MOG Technologies SA from Portugal.

Barrier 1: Difficulty to develop and operate (DevOps) applications over heterogeneous infrastructures across distributed Cloud data centers and Edge devices. The MOG development team needs to keep their live event and storytelling service continuously online when incorporating dynamic customer requirements and feedback. Unfortunately, current Cloud DevOps solutions fail in automating continuous testing, integration, deployment and monitoring of distributed applications over heterogeneous resources, such as Clouds (e.g., for video processing and switching), Edge (e.g., for video and user contribution pre-checking), and mobile devices (e.g., media rewarding), due to diverse resource management models and interfaces.

Barrier 2: Difficulty to guarantee application and system level performance over heterogeneous infrastructures. For business-critical customers, MOG needs to ensure the performance of the application and deliver a guaranteed service quality (QoS). However, the diverse abstraction models on resource performance and data access constraints, such as security and privacy, result in inconsistent views from different providers. MOG cannot effectively plan its infrastructure capacity across provider boundaries to schedule and balance task loads among distributed Edge nodes and Cloud data centers, for effectively runtime handling of the dynamic data, energy, cost and security constraints.

Barrier 3: High complexity in controlling infrastructures consisting of heterogeneous distributed Cloud data centers and Edge resources. When system performance degrades (due to imbalanced load, device defects or security attacks), the high complexity of the heterogeneous infrastructure hampers MOG from making real-time decisions to adapt the infrastructure, especially when considering performance, energy, security, and cost, together with the lack of effective formal control verification mechanisms.

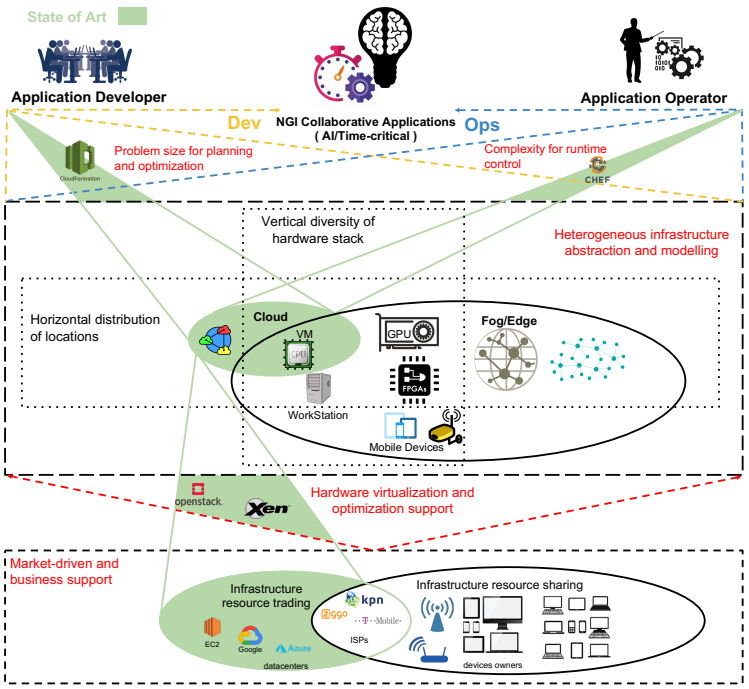

Fig. 1. The state of the art of the application DevOps in current Cloud computing environment and challenges when further considering Edge resources

Barrier 4: Difficulty to utilize advanced hardware accelerators in virtualized infrastructures. MOG developers heavily rely on natural language processing and deep neural networks to process the collected stories. Advanced specialized hardware like GPU and FPGA can largely accelerate the computation of such machine learning (ML) tasks but is only available in data centers of few public providers. MOG lacks a flexible and optimized commodity solution for accelerating its data processing, ML, and other resource-intensive tasks, capable of considering different hardware characteristics and virtual infrastructure constraints.

Barrier 5: Insufficient business support for providers in building on-demand trustworthy resource federations. MOG developers often need to consider not only services from different Cloud providers, but also special remote Edge devices deployed close to the event site where the live video platform operates. The current centralized quality control mechanisms available in the Cloud, built upon several layers of abstraction, do not suffice, hampering MOG from effectively federating heterogeneous resources from different providers. MOG strives for a trustworthy solution supported by a proper business model underneath that dynamically federates high quality and reliable resources on-demand in response to its real-time application demands.

In fact, there are plenty of tools and academic research working on how to utilize the Cloud resources. For example, OpenNebula[1] or OpenStack[2], provide different virtual infrastructure functions, known as APIs (Applications Programming Interfaces), to access physical resources through the hardware virtualization, while they are not targeting at managing Edge resources. On the other hand, academic works mainly focus on Cloud resource allocation and scheduling. Sreekrishnan et al. [4] optimize the application deployment process on hybrid Clouds and achieve the best-fit hosting combination. Xiaofeng et al. [5] enhance the makespan and the reliability of a workflow application through evaluating the Cloud resource reputation. Hassan et al. [6] develop the algorithm selecting a proper Cloud to run the task according to the data centre geolocation. All these works lack an efficient solution for the application to program and control the computing infrastructure in the DevOps lifecycle. Although our previous work CloudsStorm [7], [8] framework tackling the resource management issue from the application DevOps perspective, it still mainly focuses on working with Cloud resources, which is insufficient for a more complex Edge computing environment.

Besides, there are also some initial efforts paid in the Cloud environment using blockchain [9] and smart contract [10] techniques to enhance the trust between the provider and the customer. For instance, Hiroki et al. [11] automate the Cloud SLA (Service Level Agreement) enforcement using blockchain. We also previously proposed the witness model [12], [13] to improve the trustworthiness of demonstrating whether a Cloud service violation really happens. However, a comprehensive and systematic consideration for establishing collaborations among different roles of the ecosystem is still missing, especially when Edge resources are involved in the ecosystem.

[1]https://opennebula.org/

[2]https://www.openstack.org/

III. The ALLSTAR Approach

By analyzing these five current barriers, we firstly highlight a number of requirements when designing an approach to address these issues:

1) Resilient and seamless management across data center VMs (e.g., running heavy machine learning tasks), Edge computers (e.g., handling communication among mobile devices collecting live videos from users), and mobile devices (e.g., processing user inputs or lightweight learning processing), required to facilitate the application operation;

2) Effective infrastructure adaptation during the NGI collaborative application operation for manipulating the network topology and resources, and relocating tasks (e.g., when dealing with security attacks), alongside Cloud services scaling and load balancing;

3) Data privacy and security ensured when processing data (e.g., personal information), especially when coming from distributed sources (e.g., from crowd users or robots);

4) Critical time constraints required by NGI collaborative applications involving user interactions (e.g., in the immersive VR) or decision making (e.g., in the business recommendation);

5) SLAs required by all applications and guaranteed as QoS metrics by resource providers (even by multiple providers jointly hosting a single distributed NGI collaborative application);

6) Trustworthy service trading ecosystem based on the blockchain technology for sharing digital assets (e.g., individual story contributions) or developer media with use tracking;

7) Hardware characteristics, in particular accelerators, required in the virtualized environment to optimize the training of the machine learning tasks such as neural networks.

These requirements cover the key aspects in the development and operation lifecycle of NGI collaborative applications, their virtual hosting infrastructure, and their service management model. To solve these requirements, the current Internet, the Cloud, Fog, and Edge computing paradigms, blockchain, AI, ML, and the Cloud DevOps software engineering concept are important starting points. However, these technologies currently focus on different aspects and do not seamlessly work in a unified ecosystem. Therefore, we propose ALLSTAR architecture to unify them as a single integrated approach in support of NGI collaborative applications. The objective of "ALLSTAR" is to flatten the computing architecture of the infrastructure and make Edge resources equal to Cloud resources from the business perspective. Hence, more Edge resource will be motivated to adopt our approach and freely join the ecosystem. With their contributions in providing a more decentralized computing infrastructure, the requirements of the NGI collaborative applications can be satisfied.

Since the computing resources in the environment are organized in a decentralized manner, the traditional centralized

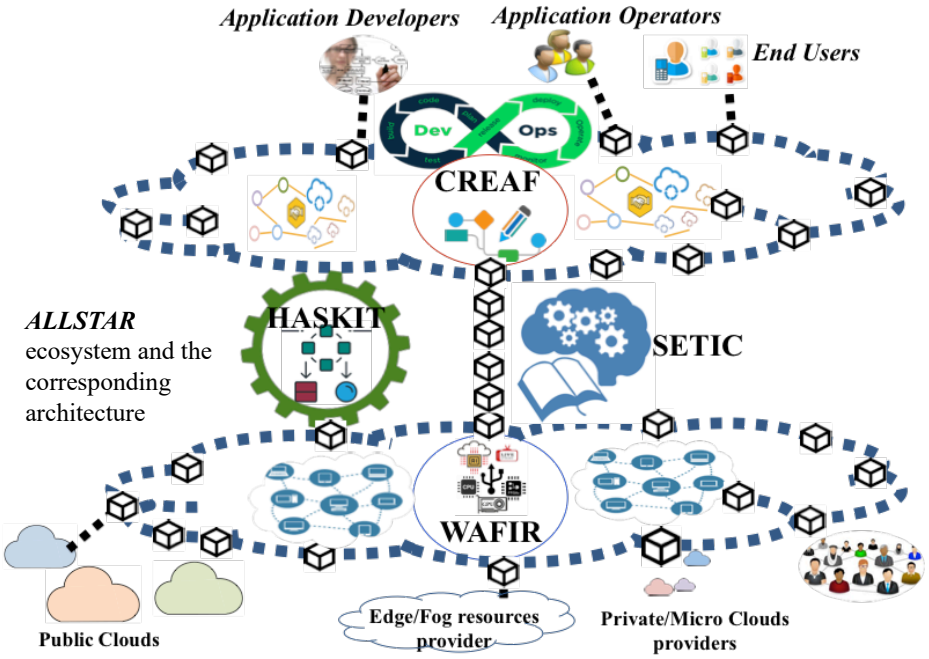

Fig. 2. The related roles of ALLSTAR ecosystem and the overview of the corresponding ALLSTAR architecture

management cannot operate so many heterogeneous resources from different providers, and also cannot provide the trust for all the providers and customers. On the contrary, the decentralized blockchain [14] technique naturally fits in this scenario, because it is an emerging technology to make every participant having trust in a decentralized ledger through the consensus algorithm, such as Proof of Work (PoW), Proof of Stake (PoS), and Practical Byzantine Fault Tolerance (PBFT).

A. Architecture Overview

As mentioned above, a more decentralized "Cloud+Edge" ecosystem requires new business models and application DevOps procedures, since the resources are complex and heterogenous. Therefore, the proposed architecture targets at solving following four challenging questions:

1) How to describe diverse resources and services in the ecosystem?
2) How to use a description model to build a seamless DevOps solution?
3) How to consider hardware features to optimize application function containerization and deployment?
4) How to provide a decentralized environment for sharing digital assets (e.g., infrastructure resources, software services, data)?

Hence, ALLSTAR designs four subsystems in response to above questions, as shown in Figure 2.

1) Heterogeneity-AS-code toolKIT (**HASKIT**) provides an open modeling language for describing heterogeneous resources and services in the ALLSTAR ecosystem, including both functional (e.g., application components and network topology) and non-functional properties (e.g., QoS, QoD, and data accessibility constraints) of an NGI collaborative application at a different stage of its lifecycle. Moreover, it provides tools to other subsystems to validate the specification composed by application developers, make control decisions, and analyze the service reputations during the operation of the application.
2) CRoss-Edge seamless infrAstructure orchestration Framework (**CREAF**) provides tools and APIs (Application Programming Interface) for application developers to plan the necessary capacity for an application, automate the provisioning of the underlying virtual infrastructure and the deployment of application components. The CREAF subsystem also provides tools to monitor, diagnose, and control the application at runtime.
3) HardWare chAracterized Function vIRtualization framework (**WAFIR**) provides a framework that focuses on hardware accelerators (e.g., GPU and FPGA) and OS-level (Operating System) virtualization (e.g., through containers) for meeting application QoS concerns. The framework also provides special care on security isolation and energy optimization.
4) Service fEderation defined Trustworthy Inter-Chain platform (**SETIC**) provides a blockchain-based decentralized fabric for the ALLSTAR ecosystem to record the service transactions among providers and consumers, enforces the SLAs using smart contracts, and provides secure-by-design services for naming, lookup, resource discovery, and querying acquired knowledge.

IV. Technical Details

The ALLSTAR software architecture consists of novel combination of state-of-the-art technologies in DevOps, Cloud,

Fog, and Edge computing, Blockchain. For each component mentioned in the last section, we present the detailed description of sub-components and related technologies in this section. The details are shown in different colored boxes, where each colored box represents the main component.

A. Heterogeneity-AS-code toolKIT (HASKIT)

The Heterogeneity-AS-code toolKIT (HASKIT) provides: 1) an open description language called ALLSTAR Modelling Language (ALLSTAR-ML); 2) a number of tools for validating the model; 3) decision making mechanisms for applications runtime control; and 4) reputation auditing for the participants in the ecosystem.

- *ALLSTAR Modeling Language* (ALLSTAR-ML) provides an open description language for modeling not only the key elements in both NGI applications (e.g., services, components, functions, and dependencies among them) and infrastructures (e.g., resource types, devices and network topologies), but also the properties of these elements at different security and access constraints levels (e.g., QoS, QoE, and QoD). The language examines the industrial modeling standards (e.g., TOSCA) and Cloud ontologies (e.g., INDL , mOSAIC) for describing Cloud-Edge computing infrastructures and extends them with flexible semantic linking to effectively capture and specify properties, which are derived from new characteristics of NGI collaborative applications.
- *Interactive ALLSTAR-ML verifier* provides verification support for both functional and non-functional requirements of applications, as specified in the NGI collaborative application model. It verifies the application logic, time, quality constraints, as well as security and privacy requirements in the context of requirement modeling, application composition, provisioning, and deployment.
- *Complex NGI-Cloud control decision validator* applies complex systems theories, such as (probabilistic) Boolean networks and dynamical systems, to validate the consequences a control decision may cause on the graph representation of the runtime Cloud infrastructure status. Efficient control algorithms detect the factors driving the infrastructure into a critical state, which may affect the overall performance and stability of the application.
- *Reputation auditor* for ecosystem participants provides a reputation model focused on QoS experiences and history to audit the behavior of the ecosystem participants. The results are achieved not only based on end users feedback, but also on violation detection reports during the crowd-based SLA enforcement. The SETIC blockchain-based fabric explained later provides this information.

B. CRoss-Edge seamless infrAstructure orchestration Framework (CREAF)

The CRoss-Edge seamless infrAstructure orchestration Framework (CREAF) provides a unified and robust interface for agile and programmatic seamless application-infrastructure orchestration considering heterogeneity across Cloud and Edge resources. CREAF also empowers ALLSTAR with smart, automated infrastructure capacity planning, and resource management. It employs predictive analysis of application-infrastructure driven requirements to simplify scaling and provisioning with improved operational efficiency, and also minimize management costs.

- *Interactive NGI application-infrastructure programming environment* creates the description, with privacy-by-design features in accordance to GDPR guidelines, using ALLSTAR-ML and the interactive verification tool provided by HASKIT for user-centric support approaches. CREAF further leverages real-time analytics based on intelligent optimization and deep learning heuristics, to measure against infrastructure and application-driven metrics. It provides proactive deployment insights of the application and the infrastructure management is based on ALLSTAR-ML constraints and specifications. It also considers performance issues, data privacy, security, application needs, unexpected traffic spikes or even to control costs between providers across Cloud and Edge boundaries.
- *Infrastructure planner* with automated integration capability enhances the continuous provisioning, deployment, testing, and intra-service orchestration DevOps processes across the software development lifecycle. Firstly, it is able to plan the infrastructure capacity according to the application requirements. Meanwhile, it simplifies and accelerates the transition from manual to automated continuous service delivery, ensuring full capability across the federation of heterogeneous physical and virtual infrastructures, services and applications. Precisely, CREAF utilizes the existing industrial DevOps tools (e.g., Kubernetes , Puppet , Chef) and provides AI-driven services to automate the data analysis and accelerate routine operations (e.g., continuous integration, provisioning, deployment, testing, and delivery) with effective infrastructure usage and collaboration.
- *Systematic NGI application-infrastructure diagnoser*, which is fed with covariate performance metrics and measurements across heterogeneous Cloud and Edge infrastructures, monitors and exploits the application and infrastructure orchestrated resource history as a baseline. It exploits the baseline performance against small performance deviations (e.g., network congestions, delays, and bandwidth allocations) and flags a possible heterogeneous resource anomalous condition in need of further proactive actions.
- *Seamless cross-Edge infrastructure orchestrator* considers the geographically dispersed and fragile networked infrastructures orchestration within and across Cloud data centers. It exploits open source orchestration tools tools (e.g., Kubernetes) and automates the operations not limited to application-driven development, production and deployment, but also incorporates infrastructure adjustments. Such adjustment owes to complex and

heterogeneous architecture design integration, including microservices, hybrid Cloud, Edge computing and IoT systems.

C. hardWare chAracterized Function vIRtualization framework (WAFIR)

The hardWare chAracterized Function vIRtualization framework (WAFIR) provides hardware accelerator aware virtualization and function containerization for quality-critical NGI applications. By adopting proper level of virtualization techniques (e.g., VM, container, and unikernel), WAFIR creates self-contain portable components for application functions according to the requirements and available hardware acceleration capacity (e.g., Intel, ARM, Nvidia GPU, FPGA), and publishes them to the knowledge mesh within the fabric of SETIC (see next section) based on the ALLSTAR-ML schema. WAFIR also provides CREAF with underlying hardware acceleration support for optimizing application function deployment.

- *Hardware portable function virtualizer* supports different types of hardware, able to expose to the correct acceleration depending on the application SLA. In the particular case of FPGAs, specific hardware extensions enable dynamic resource allocation, reconfiguration and runtime migration.
- *Quality-critical and energy-aware function composer* allows application developers iteratively select the ingredients of application functions and optimize them based on the energy and performance constraints, and the availability of the hardware accelerators choosing the suitable level of virtualization (e.g., container, VM, or unikernel) for different performance and security requirements.
- *Security and performance isolator*, for mixed criticality tasks in NGI applications, isolates functional safety critical workloads from untrusted connected applications to ensure security and performance. A secure enclave will leverage CPU processor extensions, such as ARM TrustZone and Intel SGX, to ensure a secure environment.

D. Service fEderation defined Trustworthy Inter-Chain platform (SETIC)

The Service fEderation defined Trustworthy Inter-Chain platform (SETIC) provides the underlying ledger and smart contract support for constructing the decentralized Cloud ecosystem, in which participants collaborate without relying on a centralized authority. Infrastructure providers can dynamically join and leave the ecosystem, and offer their resources as a service to the community of application developers and operators.

- *Service federation defined inter-chain fabric* mainly provides two aspects of functionalities: application DevOps management and the application execution environment. For the management perspective, the inter-chain fabric provides: 1) the underlying lEdgers for transactions and

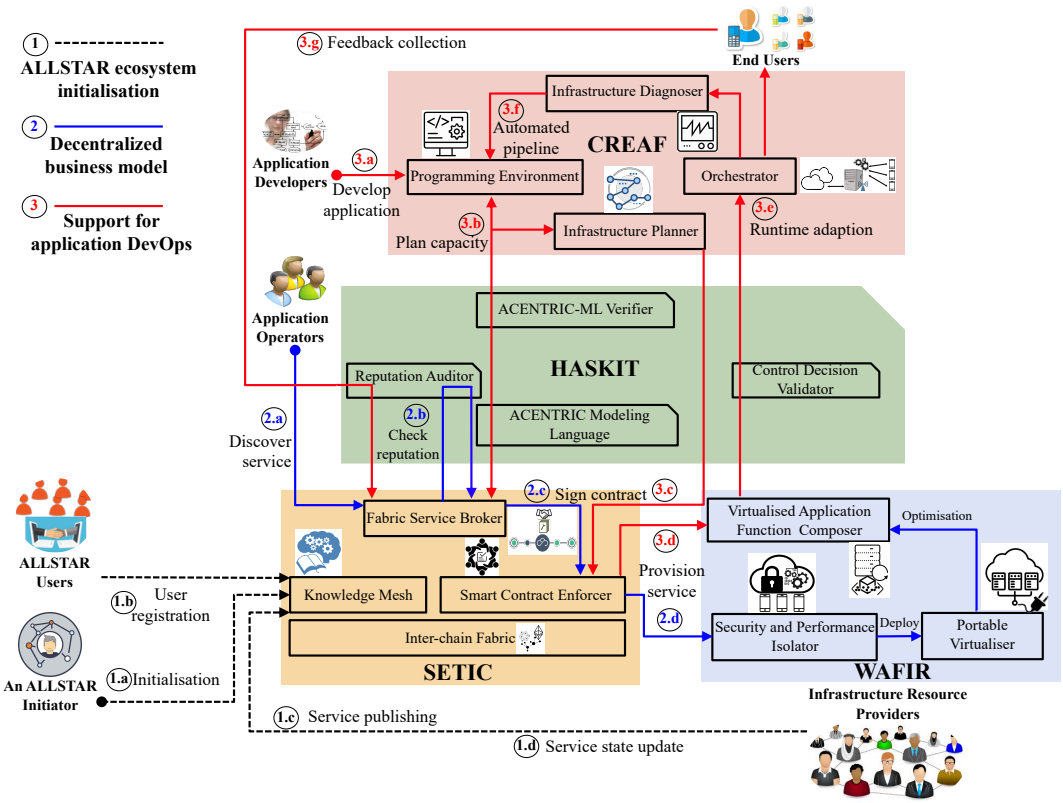

Fig. 3. Detailed sub-components description and key interaction lifecycles (business model and DevOps) of ALLSTAR approach

interactions among the different assets providers in the Cloud ecosystem, including infrastructure resources, services and models; and 2) an interoperable fabric for interconnecting different blockchains used in the ecosystem. For the execution perspective, the inter-chain fabric provides customizable blockchain-as-a-service for service federation defined applications, which can be executed as smart contracts on a new blockchain on demand. The content on the blockchain includes data, software, media content, etc. Through combining different chains, the fabric exploits the trade-off between the system performance and consensus mechanisms according to the transaction types: public blockchain with more trust and less efficiency, and permissioned blockchain with conversely less trust and more efficiency instead.

- *Evolutionary knowledge mesh* manages the information and the knowledge in the decentralized ecosystem using the underlying inter-chain fabric, including 1) infrastructure provider information (e.g., prices, capacity and special hardware feature), 2) a catalogue of virtualized application function repository for agile deployment, 3) application naming information for delivering services to end users, and 4) reputation from crowds and user feedbacks.
- *Crowd-based trustworthy smart contract enforcer* provides a trustworthy mechanism for service providers and consumers to automate specific service transactions, required by the applications in the ecosystem (e.g., negotiating prices, payment conditions, and violation conditions), with incentivized witnesses model to credibly feedback about the off-chain events. The results of this module also play an important role in generating the providers reputation.
- *Context-aware fabric service broker* enables users to interact with the inter-chain fabric and knowledge mesh, for example for publishing service on the mesh and semantically discovering resources or other types of assets from the ecosystem. It can also interact with executable smart contract code for negotiating SLA, generating new smart contracts, or detecting violation during enforcement.

V. The Business Model and DevOps lifecycle

ALLSTAR integrates the HASKIT, CREAF, WAFIR, and SETIC subsystems in a coherent ALLSTAR ecosystem shown in Figure 3, where each subsystem encapsulates the functionality and components as microservices, ensuring a high-level of modularity. The integration interfaces among subsystems and microservices define a high-level abstract and generic application-programming interface (API) to ensure portability and sustainability, so that new implementations is able to interoperate with the existing ones as the technology evolves. The ALLSTAR ecosystem integration has three main scenarios: 1) decentralized ALLSTAR ecosystem instantiation; 2) decentralized business model of resource providers; and 3) application development and operation in the ecosystem. We briefly explain each scenario in the following key steps.

Firstly, the decentralized ALLSTAR ecosystem instantiation sets up a working ecosystem in four steps:

a) *Initialization*. The first component needs to be initialized in the ALLSTAR ecosystem is the knowledge mesh of SETIC, which can initialized by creating specialized ALLSTAR smart contract on a public blockchain;

b) *User registration*. The knowledge mesh provides interfaces for a public blockchain participant to register his/her account (e.g., wallet address) in the ecosystem as a specific role, e.g., application developer, an application operator, resource provider, or end user;

c) *Service publishing*. A provider can announce his/her services by publishing service description (using the ALLSTAR-ML) into a catalogue in the knowledge mesh;

d) *Service state update*. A provider is also able to modify the statue (e.g., availability) of his/her service by setting availability state of the corresponding smart contract of the catalogue in the knowledge mesh.

A user can buy service from a provider by creating a smart contract of the SLA, using SETIC. The details will be in the next scenario.

Secondly, users buy services from providers by creating smart contracts of the SLA, using SETIC. The decentralized business model for buying and provisioning resources in the ecosystem has four steps:

a) *Discover service*. A user (e.g., application developer or operator) can discover services using the context-aware fabric service broker provided by SETIC based on the ALLSTAR-ML schema;

b) *Check reputation*. The user may invoke the HASKIT reputation auditor to check the resource provider reputation, performed without a centralized third party and with trustworthy results;

c) *Sign contract*. The user (e.g., application developer) can directly buy services from a provider invoking SETIC to negotiate and sign a smart contract with the provider based on quality constraints, and to initialize crowd-based trustworthy smart contract enforcer for detecting service violation;

d) *Provision service*. A service provider can utilize the hardware portable function virtualizer of WAFIR to provision virtual resources on top of specific hardware.

The resource provisioning and application function deployment is often embedded as part of the DevOps lifecycle. We will discuss them in more detail in the next scenario.

Thirdly, the DevOps application development lifecycle, embedding the resource provisioning and application function deployment, operates the application in seven steps:

a) *Develop application*. An application developer utilizes the interactive application-infrastructure programming environment from CREAF to customize the virtual infrastructure by invoking infrastructure information from the knowledge mesh. The developer also queries providers on their resources availability, prices, reputation, and so on;

b) *Plan capacity*. CREAF performs dynamic capacity planning to optimize the application and the required virtual infrastructure, and invokes the interactive ALLSTAR-ML verifier for

dynamically verifying whether the composed application and virtual infrastructure match the performance constraints;

c) *Sign contract*. The application developer uses SETIC to negotiate and sign smart contracts with the selected resource providers, after the application and infrastructure design;

d) *Provision service*. CREAF provisions the resources and invokes the quality-critical and energy-aware function composer of WAFIR to create a suitable application function for deployment based on application requirements and available components. CREAF deploys a blockchain-as-a-service and registers it to the knowledge mesh of SETIC, if the application needs its own blockchain services;

e) *Runtime adaptation*. CREAF enables the runtime application operators to diagnose the infrastructure failures according to the monitoring information and perform control decisions with support from the complex NGI-Cloud control decision validator (provided by HASKIT);

f) *Automated pipeline*. CREAF continuously considers changes in the application or its virtual infrastructure and continuously automates the pipeline during the lifecycle;

g) *Feedback collection*. SETIC provides decentralized naming services for delivering the application service to the end user. Furthermore, the end user is able to feedback with the QoS experiences.

VI. Conclusion

This paper describes a typical scenario in the "Cloud+Edge" environment when involving Edge resources for orchestrating dynamic applications. All the related roles of the scenario, including Cloud/Edge resource providers, application developers/operators and end users, etc, construct the ALLSTAR ecosystem. To tackle the issues of application DevOps lifecycle being faced with the complex environment, and the business model to motivate resource providers to join the ecosystem, we propose the ALLSTAR architecture, which adopts the blockchain technique as the underlying trust layer. Different from the existing work, this position paper considers the problem of the "Cloud+Edge" resource operation for the application orchestration and put forward the solution in a systematic view. However, since this is a position paper, there are still open questions to tackle in the future work, such as the performance of the underlying blockchain platform, or the specific reputation model for evaluating providers.

Acknowledgement

This work is funded by the EU Horizon 2020 research and innovation program under grant agreements 825134 (ARTICONF project), 824068 (ENVRI-FAIR project) and LifeWatch ERIC. It is also supported by the National Key Research and Development Program of China (2016YFB1000100). The authors would also like to thank Alexandre from MOG technologies providing the use case.

References

[1] K. Jeferry, G. Kousiouris, D. Kyriazis, J. Altmann, A. Ciuffoletti, I. Maglogiannis, P. Nesi, B. Suzic, and Z. Zhao, "Challenges emerging from future cloud application scenarios," *Procedia Computer Science*, vol. 68, pp. 227–237, 2015.

[2] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C.-H. Hsu, "User allocation-aware edge cloud placement in mobile edge computing," *Software: Practice and Experience*, 2019.

[3] S. Teerapittayanon, B. McDanel, and H.-T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 328–339.

[4] S. Venkateswaran and S. Sarkar, "Architectural partitioning and deployment modeling on hybrid clouds," *Software: Practice and Experience*, vol. 48, no. 2, pp. 345–365, 2018.

[5] X. Wang, C. S. Yeo, R. Buyya, and J. Su, "Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1124–1134, 2011.

[6] H. Ziafat and S. M. Babamir, "Optimal selection of vms for resource task scheduling in geographically distributed clouds using fuzzy c-mean and molp," *Software: Practice and Experience*, 2018.

[7] H. Zhou, Y. Hu, J. Su, C. de Laat, and Z. Zhao, "Cloudsstorm: An application-driven framework to enhance the programmability and controllability of cloud virtual infrastructures," in *International Conference on Cloud Computing*. Springer, 2018, pp. 265–280.

[8] H. Zhou, Y. Hu, X. Ouyang, J. Su, S. Koulouzis, C. de Laat, and Z. Zhao, "Cloudsstorm: A framework for seamlessly programming and controlling virtual infrastructure functions during the devops lifecycle of cloud applications," *Software: Practice and Experience*, 2019.

[9] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[10] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.

[11] H. Nakashima and M. Aoyama, "An automation method of sla contract of web apis and its platform based on blockchain concept," in *Cognitive Computing (ICCC), 2017 IEEE International Conference on*. IEEE, 2017.

[12] H. Zhou, X. Ouyang, Z. Ren, J. Su, C. de Laat, and Z. Zhao, "A blockchain based witness model for trustworthy cloud service level agreement enforcement," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1567–1575.

[13] H. Zhou, X. Ouyang, J. Su, C. de Laat, and Z. Zhao, "Enforcing trustworthy cloud sla with witnesses: A game theory–based model using smart contracts," *Concurrency and Computation: Practice and Experience*, 2019.

[14] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, "Blockchain challenges and opportunities: A survey," *International Journal of Web and Grid Services*, vol. 14, no. 4, pp. 352–375, 2018.