Preprint - to be presented at IEEE International Conference on Parallel and Distributed Systems (ICPADS)2-4 December 2020, Hong Kong

A Trustworthy Blockchain-based Decentralised Resource Management System in the Cloud

Zhiming Zhao*, Chunming Rong† and Martin Gilje Jaatun†
†University of Amsterdam, Netherlands z.zhao@uva.nl
*University of Stavanger, Norway {chunming.rong, martin.g.jaatun}@uis.no

***Abstract*—Quality Critical Decentralised Applications (QCDApp) have high requirements for system performance and service quality, involve heterogeneous infrastructures (Clouds, Fogs, Edges and IoT), and rely on the trustworthy collaborations among participants of data sources and infrastructure providers to deliver their business value. The development of the QCDApp has to tackle the low-performance challenge of the current blockchain technologies due to the low collaboration efficiency among distributed peers for consensus. On the other hand, the resilience of the Cloud has enabled significant advances in software-defined storage, networking, infrastructure, and every technology; however, those rich programmabilities of infrastructure (in particular, the advances of new hardware accelerators in the infrastructures) can still not be effectively utilised for QCDApp due to lack of suitable architecture and programming model.**

***Index Terms*—DevOps, Cloud Computing, Blockchain**

I. Introduction

The recent advances in Cloud computing, IoT, Artificial Intelligence and big data greatly accelerate the innovations in the digitalisation of business applications towards Next Generation Internet (NGI)[1], such as

- Seamlessly processing dynamic physical events when automating business processes (e.g., temperature sensitive cold supply chains);
- Realtime online cooperation (e.g., crowd story telling during content delivery (e.g., in live events); and
- Business critical operations (e.g., risk assessment) during complex decision making (e.g., for financial investment).

Those innovations promote a new paradigm of applications, which have strict requirements for system performance and service quality, involve heterogeneous infrastructures (Clouds, Fogs, Edges and IoT), and rely on the trustworthy collaborations among participants of data sources and infrastructure providers to deliver their business value, which cannot be fully supported by the classical centralised architecture. We thus call such new application paradigm as Quality Critical Decentralised Applications (QCDApp). Blockchain technologies have demonstrated their great potential for realising trustworthy (via immutable ledgers and consensus among peers) and fault tolerance (no single point failure among decentralised nodes) in business applications, and have become a basis for developing Decentralised Applications (DApp). However, the current blockchain technologies suffer from high storage cost of ledgers, low collaboration efficiency among distributed peers for consensus, and insecure off-chain data sources for blockchain transactions.

On the other hand, Cloud computing has been a major disruptive technology providing resources-as-a-service for diverse Internet applications. While Cloud environments provide not only elastic capacity, but also customisable connectivity, often called virtual infrastructure, over a large-scale network, the resilience of the cloud have enabled significant advances in software-defined storage, networking, infrastructure, and every technology, which promotes emergence of heterogeneous programmable infrastructuresIn this paper, the terms of programmable infrastructure and software-defined infrastructure are interchangeable. across different Clouds, and devices on the network edges (often called Edge or Fogs). However, this rich programmability of infrastructure, in particular, the advances of new hardware accelerators in the infrastructures, can still not be effectively included in the development and operations (DevOps) of Quality Critical Decentralised Applications.

II. Background

Quality Critical Decentralised Applications (QCDApp) focus on software architecture and DevOps tools, and across three typical domains: Decentralised Applications, quality critical systems and programmable heterogeneous infrastructures. In this section, we review the state of the art of 1) DApps and quality critical DApps, 2) trustworthiness and privacy in distributed applications, 3) software technologies for QCDApp development, and 4) infrastructure utilisation of heterogeneous hardware advances.

Decentralized Applications (DApps) employ blockchain technologies to realise decentralised trust among peers, and provide entire business logic at the backend. Blockchain maintains system states (e.g., transactions among participants) via a worldwide and decentralised ledger, and updates the ledger through consensus mechanisms among participants (e.g., Proof of Work as in Bitcoin [2]). All the transaction states stored on the blockchain are public, verifiable and immutable. Ethereum is a second-generation blockchain which allows a general-purpose program (called a smart contract) to be stored on the blockchain, and to be executed through the Ethereum VM

978-1-7281-9074-7/20/$31.00 ©2020 IEEE

concept. The business logic of a DApp is often governed by one or several smart contracts interacting with the underlying blockchain [3]. Since the data is decentralized, even if one point or node in the network is breached, other nodes in the network can securely reinstate the data. DApps have demonstrated their potential in a big spectrum of applications, including IoT [4], financial [5], logistics [6] and social networks [7]. Quality critical business applications consider quality of service (QoS) or user experience quality, e.g., decision time or delay for online user; any fails in those constraints may cause severe loses of the business value. Time-critical applications distinguish two classes: 1) speed critical (or latency sensitive) relying on continuously system optimisation to reach as fast as possible, and 2) timeliness relying on real-time task scheduling [8]. When using virtualised infrastructures, the uncertainty of the underlying infrastructure often makes it difficult to guarantee the time critical constraints.

Trustworthiness is considered a non-functional requirement for the consistency of software quality with subjective user assessments. Trust is commonly assessed through reputation systems [9]; however, existing systems rely on ratings provided by consumers, which can lead to non-objective evaluation results. Other approaches use computational models based on sociological and biological factors of reputation concepts, or formalism [10]. However, it is still very challenging to measure the general trustworthiness of software due to the complex social context of different software actors.

Privacy remains a fundamental research challenge in distributed applications, Cloud, Fog, or Edge computing [11], in particular, diverse privacy regulations involved due to states and regions involved in the same application. Other critical quality constraints include energy consumption, e.g., power limits of edge nodes [12], security, e.g., cyber attacks [13].

The development of Quality Critical DApps has to face challenges of not only decentralised nature of the application, but also performance critical requirements:

1) An effective incentive model is needed to credit participants to encourage them to contribute to maintain fairness of the system [14]. A typical way is through transaction fees, which have two sides: it can prevent Spam or malicious executions of smart contracts; however, it may become a barrier if there is a proportion difference between the monetary values and operational overheads. Currently, many DApp developers are struggling with the high transaction fees during deployment.
2) Decentralised consensus among participants establishes trust for new transactions; however, the long delay to achieve consensus (per transaction) has been a critical issue for many public blockchains, e.g. the average time for the Bitcoin nodes to mine a block is 10 minutes, the average transaction confirmation time is around an hour (as typical required 6 blocks). Even with significant response latency reduction in Ethereum, a sufficiently small latency to support interactions of general applications is yet to be achieved in public blockchains. Longer delays frustrate users, making current DApps less competitive with existing non-blockchain alternatives.
3) Sequential Performance of a DApp is determined by the response delays from all nodes in the network, since all transactions/operations should be executed and verified by all nodes to reach a consensus. However, dependencies among software components or logical steps often exist in an application and restrict it from parallel executions; there thus is a need to provide fast sequential performance in order to handle high volumes.
4) Blockchain evolution, e.g., hard fork, is the only current approach in order to enable a system wide upgrade, which may result in the loss of participating network nodes, due to the nature of P2P consensus in blockchain. Another potential issue for a hard fork is that there will be multiple similar tokens sharing a common origin, which will confuse users.
5) Security issues in smart contracts must be enforced by careful implementation and intensive tests. Nevertheless, it is hard to guarantee a bug-free non-trivial smart contract, and more so for the high complexity in many DApps. However, any bug patch delivery may run into conflict with the immutable nature of blockchain data. Hence, the related platform must provide flexibility in supporting bug patch approaches for developers, especially for critical issues that may have system wide impacts.
6) Identity management of users and transactions are important while anonymity may be needed in certain circumstances. There has also been recent work to add the ability for anonymity on top of existing blockchains, through smart contracts and regulatory bodies requiring the Know Your Customer (KYC) and Anti Money Laundering (AML) checks without giving up the identity of the contributors to the entire global network. On the other hand, there is a movement to create Distributed Identity (DID) that can be used across all DApps in a similar way how openID was used to create a common identity across web services.

We can see that DApps clearly demonstrates the potential for applications in the next generation internet; however, DApp development faces different challenges which hamper the realisation of quality critical requirements for application needs.

The development and operations (DevOps) engineering practices enable the continuous development, testing, integration, deployment and operation of the software products. It allows the development team to effectively respond to new software requirements and operational demands, and to incrementally develop and deliver new features for operational services by automating the integration, testing and deployment processes throughout the lifecycle [15]. There are a number of industrial DevOps solutions for DApp, such as DappBot and Truffle Suite , but most of the existing tools and technologies are only designed for specific one stage.The architecture of

DApp is often modelled differently, depending on the role of the Blockchain, and the type of decentralisation. Nevertheless, a number of components can be highlighted in the basis of a DApp: e.g., smart contract, interface to blockchain, backend function, and user interfaces; however, a reference architecture for the optimization, micro-services and runtime operation of functional components needed by quality critical DApps is still in the infant stage. The Coding of DApp smart contracts can be supported by several tools and frameworks. Solidity is the most popular and most commonly programming language in Ethereum; while several other languages have been proposed: e.g., a morphing language for smart contracts called Bamboo , Simplicity and Flint for more complete and secure functions, and functional programming language such as Liquidity . The smart contract Integrated Development Environment (IDE) often exists as web based like Remix , or desktop like Ethereum Studio that supports Metamask integration, transaction logger and other features. At the moment, DApp coding environments still lack of the effective views to integrate optimisation and infrastructure programming tools developed for centralised quality critical applications. Quality critical constraints have been well studied in virtualised infrastructure in the classical centralised application, via different aspects: real-time hypervisor, programming model, and runtime scheduling and control. RT-Xen is the first real-time hypervisor scheduling framework for Xen [16], which is the most widely used open-source VM monitor. It bridges the gap between real-time scheduling theory and Xen, and provides a platform for integrating a broad range of real-time and embedded systems. The performance guarantee at the software system level relies on the optimisation between application and infrastructure, e.g., CloudStorm [17] for service applications and PrEstoCloud [18] for real-time Big Data using an extension of the Fog computing paradigm to the extreme Edge of the network, real-time container scheduling [19]. However, the engineering of quality critical applications across heteronomous infrastructure for DApps is still very challenging, due to diverse programming interfaces, and the lack of effective programming methods.

A. Continuous testing and integration (CI) of DApps

Several DApp testing tools are designed for testing smart contracts: a) Coveralls and Solidity-coverage for checking the code coverage of Solidity smart contract, b) VeriSol for formal verification and analysis of Solidity smart contracts, c) Solidity Function Profiler and Sol-profiler are for profiling smart contract function, d) Solhint is a linter which provides security and best practice rules for smart contract verification, and e) Espresso and Eth tester are Solidity test frameworks for debugging smart contracts in Ethereum. By using a test network instead of the main network, users can test the operation of smart contracts on Ethereum without causing any loss of real Ethers. CI tools such as Jenkins and Travis CI can be used to build and manage code states. CI tools specifically designed for DApps, such as Truffle Teams , can also be used to prevent integration problems during the smart contract development. Those existing testing and integrating framework provide valuable starting point for DAppOps; however, the testing for the quality critical aspects, in particular, the extreme cases in large scale will be focus of the project. Deploying and operating DApp clients to interact with live blockchain networks. Several Ethereum clients available for DApps developed using different programming languages, such as Geth , Party , WebThree , and Nethermind . These clients support functions such as mining, networking, block and transaction processing. In addition to public deployment, these clients can also be used to deploy DApps in a private network. Major Cloud providers such as Google Cloud and Microsoft Azure also provided Blockchain-as-a-Service (BaaS) to help users to deploy a private or consortium Ethereum blockchain network in cloud environments with different consensus algorithms.Centralized DApps front-end content. Although the DApp's back-end logic (smart contract) runs on a decentralized network (e.g. Ethereum blockchain), the front-end content may come from a centralized database server. The Interplanetary File System (IPFS) [20] is a protocol initiated by the Protocol Labs and aims to create a decentralised storage and file referencing solution to store server information on the Web. A typical example is OrbitDB , a decentralized, point-to-point database that uses IPFS as a backbone. Infura and 3Box Storage project also used IPFS as a decentralized storage solution. When using IPFS, individual nodes can store data which they consider as important. However, there is no way to motivate others to join the network or to store specific data. Filecoin is a complementary solution to IPFS, which provides a persistent data storage system and an incentive mechanism to solve this key problem. In Filecoin, customers pay to store data, while miners earn payments and rewards by continuously storing data and proving passwords. Similar to IPFS and Filecoin, Swarm is a decentralized storage platform and works as a native base layer service of the Ethereum web3 stack. Storj and Sia are other decentralized storage solutions that still lag behind in development.Monitoring of DApps. The performance, users, and corresponding blockchain infrastructure can be monitored through different tools. Alethio is an Ethereum analysis platform that provides real-time monitoring and anomaly detection for smart contracts. Other tools like Scout , and Neufund can provide activities monitoring and keep event logs of smart contracts.

The development of quality critical business applications is difficult due to the often contradictory constraints, e.g., performance, trustworthy, energy efficiency, security and privacy, in particular, when the application is distributed over programmable infrastructure. The DevOps tools for current Cloud applications mainly focus on fast iteration as a best practice, which can be used as the basis for DevOps tool chains for DApps. However, the differences between DApps and traditional applications require DevOps tools with different features: immutability of smart contracts after being deployed. Meanwhile, it can be very expensive and take a long period of time to modify, update, or withdraw the deployed smart contracts, which makes DevOps for DApps more complicated.

Moreover, the testing and adaptation of quality critical aspects in DApps is another challenge currently not yet being studied.

The programmability of software-defined infrastructure can only be effectively included in the application when the infrastructure functions are well abstracted and presented via a usable model. Abstracting the function of networked infrastructure using techniques such as SDN [21] and NFV [22] have been extensively studied during the past years in the network and Cloud communities; but agile development across multiple domains requires greater automation of infrastructure configuration and adaptation based on platform-agnostic application specifications. For customised infrastructure planning, techniques such as multi-objective optimisation [23] or multi-deadline critical path optimisation [24] reference missing can be used to select optimal infrastructure resources for a given application. Zhao et al. [25] addresses the configuration of networked infrastructure across multiple sites, allowing data to flow freely over heterogeneous infrastructures.

1) Virtualisation consists in abstracting the system hardware resources to run multiple independent instances of an application. Different techniques exist today, where Hypervisors (e.g., Kernel-based Virtual Machine – KVM and XEN), Containers (Docker) and Unikernels (e.g., RumpRun) are the most important. They differ in the way they provide performance, security and deployability features. Hardware accelerators, considered as an important part of virtualized infrastructures (e.g., Cloud, Edge, High performance Computing and Network Functions Virtualisation), allow to execute some function faster and more efficiently than performing the same function on a general-purpose central processing unit (CPU) or on a networking device. Different types of specific hardware can be used to do acceleration such Application-Specific Integrated Circuits (ASICs), network processor, flow processors, Field-programmable gate array (FPGAs), multi-core processors, Graphics Processing Unit (GPU), etc. to offload the main CPU, and to accelerate workload performance. A field-programmable gate area (FPGA) is an integrated circuit designed to be configured through a hardware description language (HDL) similar to that used for an application-specific integrated circuit (ASIC). As a result of using FPGAs, developers can design custom hardware accelerators which are able to achieve higher performance and lower power consumption than general purpose CPU. FPGA acceleration can be applied to VNFs as well as to any other type of computing. This hardware acceleration is radically changing today's datacenter computing paradigm, as can be seen by Amazon F1 Elastic Cloud.
2) Refactoring of virtualized resources into smaller and reusable functional modules to reduce storage size and cost based on the identification of similarities in functionality [26] is one possibility. Cloud service load balancing and storage optimisation [27] can improve data fetching, but does not give sufficient consideration to dynamic storage optimisation.
3) For the deployment of services and data, researches such as [28] optimise transfer and migration of components onto Cloud via efficient network-aware scheduling; [29] consider context-aware background application scheduling in interactive mobile systems, though not with formal verification.
4) Cloud business transactions. Cloud computing is a business concept for renting affordable and cost-effective infrastructure on-demand for performing tasks with variable load. For example, the concept of SLA is one of the key business contractual terms to assure Cloud performance. Blockchain is naturally suitable in this scenario through smart contracts performing SLA negotiations [30], [31]. However, the smart contract model lacks an underlying execution engine with blockchain and is not clear whether eventual violation reports sent to the blockchain are trustworthy.

Fig. 1. Conceptual diagram of DAppOps: a blockchain-based Decentralised resource management, DevOps and trustworthy Cloud ecosystem

Today, hardware acceleration solutions are fragmented and do not enable flexibility, trusted computing and power consumption support. However, approaches like OpenCL propose a solution to decouple the hardware from the accelerated application, but fail in supporting trusted computing and power consumption decision making. In fact, applications are dependent on a specific accelerator, and do not have means to identify and authenticate the accelerator (e.g., a trained algorithm).

III. Proposed Approach

We propose a novel development and operations (DevOps) solution for such quality critical business use cases, built around six barriers we identified. DAppOps designs four subsystems in response to these questions, and crosses of three key domains: DApp, quality critical applications, and programmable infrastructures, as shown in Fig. 1.

1) Knowledge centred Application-Trust-Quality-Infrastructure cooperative DEVelopment toolKit (KATQIDevKit) provides a reference model, programming model, and tool kits for designing the application logic and underlying virtual infrastructure for a QCDApp, with effective optimisation on constraints like system performance, trustworthiness, security and energy efficiencies.
2) SErvice Federation defined Trustworthy Inter-Chains (SEFTIC) provides a blockchain-based decentralised inter-chain fabric and tools for the DAppOps developers, users and operators to effectively share digital assets in a trustworthy way.
3) Hardware characterised function virtualisation framework (HAFIR) provides a framework that effectively virtualise hardware accelerators (e.g., GPU and FPGA) and the architecture characteristics (e.g., CPU hardware extension) via the virtual infrastructure to optimise the QCDApp runtime performance, security and energy consumption.
4) Cross-Edge heterogeneous Infrastructure Decentralised Service Orchestrator (CEISOR) provides tools and APIs (Application Programming Interface) for application developers to automate the planning, provisioning, monitoring, and runtime adaptation of the virtual infrastructure during the DevOps lifecycle of a QCDApp.

The detailed explanations of those subsystems will be given in the next section.

IV. TECHNICAL CONSIDERATIONS

A. KNOWLEDGE CENTRED APPLICATION-TRUST-QUALITY-INFRASTRUCTURE COOPERATIVE DEVELOPMENT TOOLKIT (KATQIDEVKIT)

The Knowledge centred Application-Trust-Quality-Infrastructure cooperative DEVelopment toolKit (KATQIDevKit) provides: 1) an interoperable Quality Critical Decentralised Application Reference Model (QCDApp-RM) together with an open description language called DAppOps Modelling Language (DAppOps-ML); 2) a decentralised evolutionary knowledge base; 3) semantic search and recommendation tool; and 4) cooperative programming model for constraints across application logic-trust-quality-infrastructure.

The QCDApp Reference Model (QCDApp-RM) provides a multi-viewpoint based (by extending the ODP model) common vocabulary for modelling requirements, modelling architecture patterns, and for describing engineering choices, by different stakeholders involved in the development of QCDApp e.g., business operators, application component developers, DevOps managers, information specialists, and infrastructure providers. Microservices are used to model the functional components in a QCDApp for: application Functional Units (FUnits), service Quality and performance optimisation Units (QUnits), decentralised Trustworthiness Units (TUnits), Infrastructure function Units (IUnits), and semantic and Knowledge operation Unit (KUnits). A cognitive orchestrator to elastically orchestrate the functional units based on performance, energy and security constraints for heterogeneous infrastructures with self-learning capability on the runtime contexts.DAppOps Modelling Language (DAppOps-ML) provides an open description language for specifying not only the key services units in both QCDApps, execution model (e.g., decentralization, consensus and incentive models) 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., quality of services and user experiences). The language examines the industrial modelling standards (e.g., TOSCA) and Cloud ontologies (e.g., INDL and 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 QCDApps. Decentralised knowledge base manages the information and the knowledge in the decentralised ecosystem using the underlying inter-chain fabric (to be discussed in 1.3.4.2), including 1) infrastructure provider information (e.g., prices, capacity and special hardware feature), 2) a catalogue of DApp repositories for reusing code, and agile deployment, 3) application naming information for delivering services to end users, and 4) reputation of assets (e.g., services and infrastructure) providers (in 1.3.4.2).

The Semantic software component recommender can effectively search the software components from the Decentralised knowledge base, recommend the suitable software (often in open sources) for the application developers.

The Cooperative programming model for effectively programming Quality Critical Decentralised Applications, and customising the suitable virtual infrastructure for the application with consideration of not only application logic and the underlying hardware characteristics, but also the constraints for performance, security, trustworthiness, cost and energy. An interactive GUI programming interface for enabling different roles involved in the DevOps lifecycle to cooperatively program and verify the design will also be provided.

B. SERVICE FEDERATION DEFINED TRUSTWORTHY INTER-CHAINS (SEFTIC)

The SErvice Federation defined Trustworthy Inter-Chain (SEFTIC) platform provides the underlying ledger and smart contract support for constructing the trustworthy QCDApps ecosystem, in which participants collaborate without relying on a centralised 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.

The Service federation defined inter-chain fabric provides customised inter-chain environment for specific service federation required for developing and operating QCDApp. It provides: 1) the decentralised ledgers for transactions and interactions among the providers of different assets involved in DAppOps, including infrastructure resources, services and resource repositories; 2) customisable blockchain-as-a-service

for specific QCDApp to instantiate new blockchain or create new smart contracts on demand. Through combining different chains, the fabric effectively makes trade-off between the system performance and consensus mechanisms according to the transaction types, and application requirements.

The 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 incentivised 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.

The Reputation auditor for ecosystem participants provides a reputation model focused on QoS experiences and history to audit the behaviour 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.

C. HARDWARE CHARACTERISED FUNCTION VIRTUALISATION FRAMEWORK (HAFIR)

The Hardware characterised Function VIrtualisation Framework (HAFIR) provides hardware accelerator virtualisation and function containerisation for QCDApp applications. By adopting proper level of virtualisation techniques (e.g., VM, container, and unikernel), HAFIR creates self-contained 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 base based on the DAppOps-ML schema. HAFIR also provides CEISOR with underlying hardware acceleration support for optimising application function deployment. Hardware portable function virtualiser supports virtualisation of hardware accelerators (e.g., GPU and FPGA), able to expose to the correct acceleration depending on the application demands (e.g., application workloads).

The Quality-critical and energy-aware function composer allows application developers iteratively select the ingredients of application functions and optimise them based on the energy and performance constraints, and the availability of the hardware accelerators choosing the suitable level of virtualisation (e.g., container, VM, or unikernel) for different performance and security requirements.

The Security, trust and performance isolator, for mixed criticality tasks in QCDApp applications, isolates functional safety critical workloads from untrusted connected applications to ensure security and performance. Implements specific extensions to attest and verify the trustworthiness of the distributed/remote virtualized hardware infrastructure through Trusted Computing techniques and CPU extensions (e.g., ARM TrustZone and Intel SGX) allowing hardware authentication and software safety/security monitoring in real time.

D. CROSS-EDGE HETEROGENEOUS INFRASTRUCTURE DECENTRALISED SERVICE ORCHESTRATOR (CEISOR)

The CRoss-Edge heterogeneous Infrastructure Decentralised Service ORchestrator (CEISOR) provides a unified and robust interface for agile and programmatic seamless application-infrastructure orchestration considering heterogeneity across Cloud and Edge resources needed by a QCDApp. CEISOR also empowers DAppOps 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 minimise management costs.

The Quality critical DApp virtual infrastructure capacity planner enhances the continuous provisioning, deployment, testing, and intra-service orchestration DevOps processes across the software development lifecycle. The planner has to support different business models of the resource ecosystem: infrastructure might be provided by centralized big providers, e.g., Amazon or Azure, or provided by decentralised small providers. The planner has to effectively plan the infrastructure (including capacity, and topology) according to the application requirements for performance, security, trustworthiness and energy.

The Seamless cross-edge infrastructure orchestrator considers the geographically dispersed and fragile networked infrastructures orchestration within and across Cloud data centres, and Fog/Edge nodes. 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, CEISOR utilises the existing industrial DevOps automation tools (e.g., Kubernetes , Puppet and 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 QCDApp application-infrastructure diagnoser 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, failure, delays, and bandwidth allocations) and flags a possible heterogeneous resource anomalous condition in need of further proactive actions.

The Complex QCDApp application-infrastructure controller 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. Furthermore, it proposes solutions in case of identified problems.

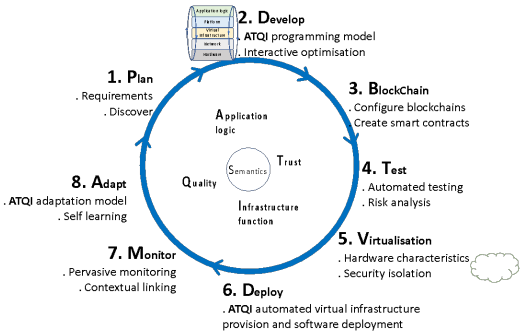

Fig. 2. DevOps lifecycle of DAppOps

V. SYSTEM INTEGRATION

The DAppOps framework integrates the KATQIDEVKIT, SEFTIC, HAFIR and CEISOR as a coherent DAppOps ecosystem shown in Fig. 2, 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 technology evolves.

In the application development phase, e.g., for a decentralised crowd story telling scenario:

1) A developer first describes the requirements of the application (e.g., locations, expected users and max video selection delay for story composition), using the QCDApp-RM vocabulary provided by the KATQIDevKit (step 1); the business representative and the application developers from MOG can specify the user stories and quality constraints using different viewpoints;
2) The developer will use the KATQIDevKit to search the knowledge base, and discover suitable source assets (e.g., mobile video processing for story telling), based on the constraints of performance (e.g., minimal quality of video, and max delay for the online story telling), operation models (incentive model for engaging participants), and privacy and security (e.g., ethic concerns of the story);
3) Based on the discovered assets (e.g., code, algorithms or services), the developers follow the reference model, and apply the cooperative programming model to optimize the application (components) design among logic, performance, trustworthy and infrastructures code;
4) During the application development, the infrastructure code will be developed cooperatively, including selecting providers, capacity planning, virtual infrastructure topology, and specific function to be deployed on the virtual infrastructure
5) Based on the infrastructure code, HAFIR will virtualize specific infrastructure, in particular, when specific hardware characteristics
6) and in the meantime, HAFIR will also encapsulate the application function and the components as mobile components
7) The SLA among infrastructure providers will be automated via the smart contracts by SEFTIC,
8) and the CEISOR will then automate the virtual infrastructure provisioning, software deployment, and the application execution
9) If the application requires provisioned blockchains for the internal usage, the SEFTIC will also dynamic customize the provisioned blockchain services, and deploy it on the virtual infrastructure initialized at the previous step
10) During the runtime of the application, CEISOR continuously monitor the execution of the application, diagnose the system status, identify the system bottlenecks or potential problems, and made decision on the system control, and finally the CEISOR should also conduct the control decision, adapt the system behavior with self-learning capability.

VI. CONCLUSION

Current blockchain technologies suffer from the low collaboration efficiency among distributed peers for consensus, but the resilience of the Cloud has enabled significant advances in software-defined storage, networking, and infrastructure. However, the advances of new hardware accelerators in the infrastructures can still not be effectively utilised for QCDApp due to lack of suitable architecture and programming model. In this paper we have outlined a system to be developed for quality critical management of decentralized resources with trust based on blockchain.

ACKNOWLEDGMENT

This research has received funding from the European Union's Horizon 2020 research and innovation program under grant agreements 860627(CLARIFY), 825134 (ARTICONF), 824068 (ENVRI-FAIR) and 862409 (BlueCloud) projects. The research is also supported by EU LifeWatch ERIC.

REFERENCES

[1] NGI4ALL, *Next generation internet white papers*, accessed Oct 18, 2020. [Online]. Available: https://www.ngi.eu/resources/white-papers-reports/
[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," http://bitcoin.org/bitcoin.pdf," 2008.
[3] X. Xu, I. Weber, and M. Staples, *Architecture for Blockchain Applications*, 2016.
[4] G. Dittmann and J. Jelitto, "A blockchain proxy for lightweight iot devices," in *2019 Crypto Valley Conference on Blockchain Technology (CVCBT)*, 2019, pp. 82–85.
[5] B. Chen, Z. Tan, and W. Fang, "Blockchain-based implementation for financial product management," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–3.
[6] Q. Betti, R. Khoury, S. Hallé, and B. Montreuil, "Improving hyperconnected logistics with blockchains and smart contracts," *IT Professional*, vol. 21, no. 4, pp. 25–32, 2019.

[7] Q. Xu, Z. Song, R. S. Mong Goh, and Y. Li, "Building an ethereum and ipfs-based decentralized social network system," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, 2018, pp. 1–6.

[8] S. Koulouzis, P. Martin, H. Zhou, Y. Hu, J. Wang, T. Carval, B. Grenier, J. Heikkinen, C. de Laat, and Z. Zhao, "Time-critical data management in clouds: Challenges and a dynamic real-time infrastructure planner (drip) solution," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 16, p. e5269, 2020, e5269 cpe.5269. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5269

[9] E. Ghazizadeh and B. Cusack, "Trust assessment for cloud identity providers using analytical hierarchy process," in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2017, pp. 1535–1541.

[10] L. Liu and W. Shi, "Trust and reputation management," *IEEE Internet Computing*, vol. 14, no. 5, pp. 10–13, 2010.

[11] A. B. Amor, M. Abid, and A. Meddeb, "A privacy-preserving authentication scheme in an edge-fog environment," in *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2017, pp. 1225–1231.

[12] Y. Li and S. Wang, "An energy-aware edge server placement algorithm in mobile edge computing," in *2018 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 2018, pp. 66–73.

[13] N. Mäkitalo, A. Ometov, J. Kannisto, S. Andreev, Y. Koucheryavy, and T. Mikkonen, "Safe, secure executions at the network edge: coordinating cloud, edge, and fog computing," *IEEE Software*, vol. 35, no. 1, pp. 30–37, 2017.

[14] Z. Yu, X. Liu, and G. Wang, "A survey of consensus and incentive mechanism in blockchain derived from P2P," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2018, pp. 1010–1015.

[15] L. Bass, "The software architect and devops," *IEEE Software*, vol. 35, no. 1, pp. 8–10, 2017.

[16] S. Xi, C. Li, C. Lu, C. D. Gill, M. Xu, L. T. Phan, I. Lee, and O. Sokolsky, "Rt-open stack: CPU resource management for real-time cloud computing," in *2015 IEEE 8th International Conference on Cloud Computing*. IEEE, 2015, pp. 179–186.

[17] 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*, vol. 49, no. 10, pp. 1421–1447, 2019.

[18] S. Taherizadeh, B. Novak, M. Komatar, and M. Grobelnik, "Real-time data-intensive telematics functionalities at the extreme edge of the network: Experience with the PrEstoCloud project," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2. IEEE, 2018, pp. 522–527.

[19] Y. Hu, C. T. de Laat, and Z. Zhao, "Multi-objective container deployment on heterogeneous clusters," in *CCGRID*, 2019, pp. 592–599.

[20] E. Nyaletey, R. M. Parizi, Q. Zhang, and K.-K. R. Choo, "Blockipfs-blockchain-enabled interplanetary file system for forensic and trusted data traceability," in *2019 IEEE International Conference on Blockchain (Blockchain)*. IEEE, 2019, pp. 18–25.

[21] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[22] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 236–262, 2015.

[23] A.-F. Antonescu and T. Braun, "Simulation of SLA-based VM-scaling algorithms for cloud-distributed applications," *Future Generation Computer Systems*, vol. 54, pp. 260–273, 2016.

[24] J. Wang, C. de Laat, and Z. Zhao, "QoS-aware virtual SDN network planning," pp. 644–647, May 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7987350/

[25] Z. Zhao, P. Grosso, J. van der Ham, R. Koning, and C. de Laat, "An agent based network resource planner for workflow applications," *Multiagent and Grid Systems*, vol. 7, no. 6, pp. 187–202, Dec. 2011. [Online]. Available: https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/MGS-2011-0180

[26] G. Kecskemeti, A. C. Marosi, and A. Kertesz, "The ENTICE approach to decompose monolithic services into microservices," in *2016 International Conference on High Performance Computing & Simulation (HPCS)*. Innsbruck, Austria: IEEE, Jul. 2016, pp. 591–596. [Online]. Available: http://ieeexplore.ieee.org/document/7568389/

[27] E. Al Nuaimi, H. Al Neyadi, N. Mohamed, and J. Al-Jaroodi, "Applications of big data to smart cities," *Journal of Internet Services and Applications*, vol. 6, no. 1, p. 25, Aug. 2015. [Online]. Available: http://www.jisajournal.com/content/6/1/25

[28] Y. Hu, H. Zhou, C. de Laat, and Z. Zhao, "ECSched: Efficient Container Scheduling on Heterogeneous Clusters," in *Euro-Par 2018: Parallel Processing*, M. Aldinucci, L. Padovani, and M. Torquati, Eds. Cham: Springer International Publishing, 2018, vol. 11014, pp. 365–377, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-319-96983-1_26

[29] J. Lee, K. Lee, E. Jeong, J. Jo, and N. B. Shroff, "CAS: Context-Aware Background Application Scheduling in Interactive Mobile Systems," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1013–1029, May 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7869347/

[30] 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*. Paris, France: IEEE, Apr. 2019, pp. 1567–1575. [Online]. Available: https://ieeexplore.ieee.org/document/8737580/

[31] R. B. Uriarte, H. Zhou, K. Kritikos, Z. Shi, Z. Zhao, and R. De Nicola, "Distributed service-level agreement management with smart contracts and blockchain," *Concurrency and Computation: Practice and Experience*, Apr. 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5800