Blockchain-based Decentralized Applications meet Multi-Administrative Domain Networking

Raphael Vicente Rosa and Christian Esteve Rothenberg
School of Electrical and Computer Engineering (FEEC) - University of Campinas (UNICAMP), Brazil
{rvrosa,chesteve}@dca.fee.unicamp.br

ABSTRACT

Envisioned 5G use cases arise challenges around the realization of inter-domain relationships beyond traditional Border Gateway Protocol (BGP) peering, e.g. end-to-end service deployments tailored to specific business needs (a.k.a. slices). Blockchain technologies bring consensus among decentralized non-trusting counterparts as a shared ledger, offering potential approaches for multi-administrative domain networking seeking agile, transparent end-to-end sliceable Service Level Agreements (SLAs). This demo showcases an experimental prototype based on best of breed open source components (e.g., Ethereum, OVS, Neo4j, Ryu/OpenFlowv1.3, ARIA/TOSCA) illustrating blockchain Decentralized Application (DApp) functionalities for life cycle management of multi-administrative domain network services.

1 INTRODUCTION

5G services of diverse nature (e.g., augmented reality, vehicular communications, massive/dense IoT) are calling for advanced multi-administrative domain service deployments, an open challenges arising from vertical customers of communications service providers [2]. Delivered as so-called slices [8], diverse types of end-to-end services sharing common infrastructure resources pose automation challenges to carriers in order to continuously translate application requirements into partnership agreements while meeting customers demands for agility and transparency, altogether contributing to a complex distributed SLA-based orchestration hazard.

Administrative domains aim for shared revenue models from vertical businesses and roaming scenarios [6], whose realization for a given end-to-end network service would benefit from every point-to-point segment being able to distributively contribute with intermediary packet flow attribute assessments (e.g., throughput, latency, frame loss ratio) for a given service supply chain. To accommodate decentralized non-trusting counterparts (administrative domains) in the need of smart contracts (dynamic agreements) for consensus (composed SLAs), we advocate for the opportunities unlocked by a shared ledger of abstracted capabilities (end-to-end slice characteristics) based on blockchain-based DApps to fulfill the life cycle management of services in multi-administrative domain networking scenarios.

The main contributions of this demonstration is to showcase through an actual open-source-enabled prototype implementation the ability of blockchain DApp to implement smart contracts under an identity management scheme where only authorized entities are allow to realize specific tasks in remote administrative Multi-Domain Orchestrator (MdO). Moreover, using blockchain to store events of network service life cycle management workflows allows a DApp to keep track of traded network assets with regard to their deployment, configuration, and monitoring processes. The programmed logic inside a smart contract enables to trigger events based on a network service state (e.g., traffic overload) and apply contractual MdO actions (e.g., scaling, migration).

2 BLOCKCHAIN-BASED MULTI-DOMAIN EXCHANGE PLATFORM

Towards the feasibility claims of blockchain approaches for multi-domain networking, we built a prototype of an exemplar MdO (see Fig. 1) based on the following components:
Exchange: interconnects internal elements through event-oriented Application Programming Interfaces (APIs). *Peering* handles the interaction between administrative domains. *Slicing* uses the Aria-TOSCA [1] engine, jointly with blueprints to describe a Software Defined Networking (SDN) plugin interfacing applications and to realize lifecycle service orchestration in intra- and inter-administrative domains. *Notary* represents the DApp that manages and operates smart contracts implemented with Ethereum [3].

SIGCOMM Posters and Demos '18, August 20–25, 2018, Budapest, Hungary

ACM ISBN 978-1-4503-5915-3/18/08…$15.00
https://doi.org/10.1145/3234200.3234217

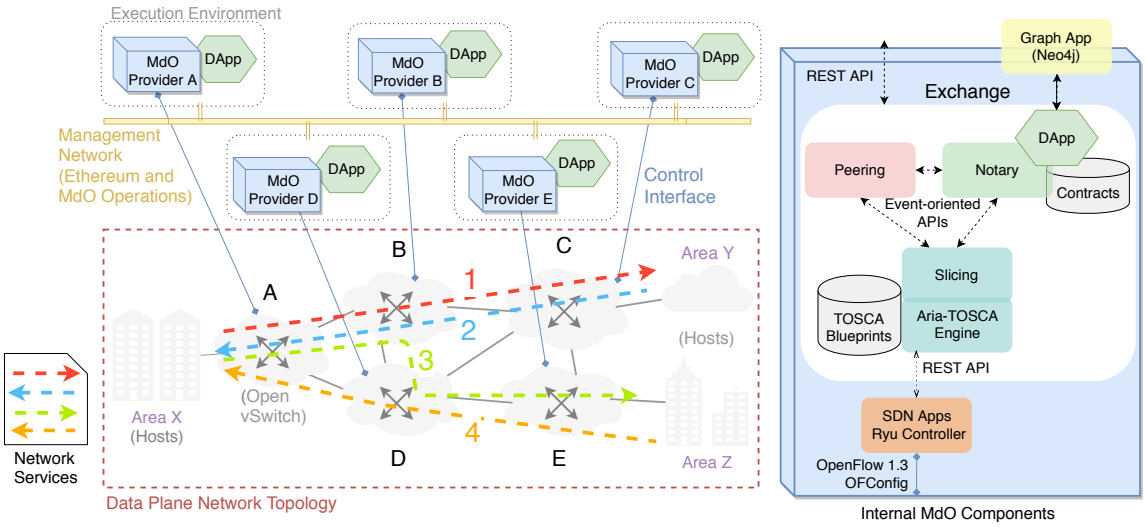

Figure 1: Demo scenario (left), and component details of the MdO/DApp prototype (right).

SDN App: based on the Ryu [7] controller, receives REST commands from the *Slicing* component for southbound programming of traffic forwarding rules in the OVS instances through flow entries (OpenFlow v1.3) and queues (OF-Config).
Graph App: interfaces *Notary* blockchain DApps where MdO smart contracts reside, enabling their information to be periodically pulled and pushed into the Neo4j [5] graph database model for network services auditing. Therefore, queries can be made into specific chained occurrences of particular contracts and events.

3 DEMONSTRATION

As shown in the companion video,[1] demo attendees will leard about the DApp implementation in a proof of concept experiment illustrating the registration of life cycle management events, from instantiation to decommission, of sliced network services (shown as dashed lines 1 to 4 in Fig. 1) deployed across multiple administrative domains.

In the demo scenario, a customer of provider A intends to deploy services using assets from providers A, B, C, D, and E. Accordingly, the customer issues a smart contract to log life cycle workflow events in each domain, wherein provider A deploys the smart contract in the blockchain network, requesting other domains to join it, and register each of them along their associated roles for the upcoming service deployment. When the customer deploys the smart contract, in each domain Exchange platform the programmable logic takes place to establish the semantic association of the smart contract with TOSCA service blueprint deployment workflows – i.e., in each *Slicing* workflow call, call-output events are handled by the *Notary* to create a transactions into the agreed smart contract, logging the requested life cycle management operations of the customer network service.

Through the deployment of each customer service, the demo highlights key aspects of the smart contract logic, including the event-based interconnection of *Exchange* components, data plane considerations, semantic relationships between smart contracts and TOSCA blueprints, and further platform design and prototype implementation aspects. A Neo4j application is used to instantiate the graph model of the blockchain and smart contracts presented as abstract logged life cycle management events along a live visualization of per-domain performance metrics (e.g., CPU consumption, transaction/event timing). Not described above but relevant to demo attendees will be a discussion to map life cycle operations and MdO considerations to three use cases: MEF SD-WAN, ETSI NFVIaaS, and 3GPP Network Slicing, in addition to observed trade-offs (e.g., performance, security, scalability) as well as open research issues, opportunities and related work (e.g., on DNS [4] and IoT [9]).

4 CONCLUDING REMARKS

This demo contributes with an exemplar implementation of open source blockchain and networking components to realize an approach that delivers potential solutions to life cycle management for network service consensus among administrative domains. Our proof of concept experiments speak in favor of the enabling open source software ecosystem but will certainly stimulate discussions on uncertain practical aspects of any multi-administrative technology in addition to ongoing research and development efforts.

[1] https://drive.google.com/open?id=14gsJuzS4PgOt_X5o1Lh0SOpwYRBcHrmB

ACKNOWLEDGMENTS

This research was partially supported by the Innovation Center, Ericsson S.A., Brazil, grant UNI.62, and by the European Union's Horizon 2020 grant agreement no. 777067 (NECOS - Novel Enablers for Cloud Slicing), as well as from the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC.

REFERENCES

[1] Apache Software Foundation. 2018. ARIA TOSCA. (March 2018). http://ariatosca.incubator.apache.org/ Accessed on 2018-05-01.

[2] Luis M. Contreras and Diego R. Lopez. 2018. A Network Service Provider Perspective on Network Slicing. (January 2018). https://sdn.ieee.org/newsletter/january-2018/a-network-service-provider-perspective-on-network-slicing

[3] Ethereum Foundation. 2018. Ethereum. (March 2018). https://www.ethereum.org/ Accessed on 2018-05-01.

[4] Adiseshu Hari and T. V. Lakshman. 2016. The Internet Blockchain: A Distributed, Tamper-Resistant Transaction Framework for the Internet. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks (HotNets '16)*. ACM, New York, NY, USA, 204–210. https://doi.org/10.1145/3005745.3005771

[5] Neo4j Company. 2018. Neo4j. (March 2018). https://neo4j.com/ Accessed on 2018-05-01.

[6] NGMN Alliance. 2017. 5G Network and Service Management including Orchestration. (March 2017). https://www.ngmn.org/uploads/media/170307_5G_Network_and_Service_Management__including_Orchestration_2.12.7.pdf

[7] Ryu SDN Framework Community. 2018. Ryu. (March 2018). http://osrg.github.io/ryu/ Accessed on 2018-05-01.

[8] K. Samdanis, X. Costa-Perez, and V. Sciancalepore. 2016. From network sharing to multi-tenancy: The 5G network slice broker. *IEEE Communications Magazine* 54, 7 (July 2016), 32–39. https://doi.org/10.1109/MCOM.2016.7514161

[9] P. K. Sharma, M. Y. Chen, and J. H. Park. 2018. A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT. *IEEE Access* 6 (2018), 115–124. https://doi.org/10.1109/ACCESS.2017.2757955