HYPERLEDGER FOUNDATION

ABOUT HYPERLEDGER

Hyperledger Foundation was founded in 2015 to bring transparency and efficiency to the enterprise market by fostering a thriving ecosystem around open source blockchain software technologies. As a project of the Linux Foundation, Hyperledger Foundation coordinates a community of member and non-member organizations, individual contributors and software developers building enterprise-grade platforms, libraries, tools and solutions for multi-party systems using blockchain, distributed ledger, and related technologies. To learn more, visit **www.hyperledger.org**.

LFNETWORKING

ABOUT LF NETWORKING

LF Networking (LFN) brings together eight top networking projects to increase harmonization across platforms, communities, and ecosystems. The LFN projects address major industry challenges—through collaboration between end users, vendors, and developers—and are transforming all aspects of the network and accelerating open source deployments.

Self-Assessing Service Level Agreements (SLAs) with Hyperledger Fabric

Purpose of this White Paper

This white paper describes a prototype for the self-assessment of Service Level Agreements (SLAs) in the Telecom industry, although this approach may be used in any industry. The proposed approach uses the Hyperledger Fabric framework for a permissioned blockchain network hosting Trusted Execution Environments.

Intended Audience

The intended audience includes executives, developers, and clients of any organization that does business using any type of SLA. The proposed generic approach aims to cover different types of SLAs with the roles of the various actors—service provider, service vendor, service seller, application provider, application client, customer, or end user—and their contract activity all defined and addressed within the proposed framework.

A Collaboration

This solution brief was driven by the **Hyperledger Telecom Special Interest Group**, a collaboration with LF Networking and its associated projects that explore use cases for blockchain technology in the Telecom industry.

V1.0 published December 2022.

1. Introduction

This part defines a Service Level Agreement, describes the drawbacks of conventional SLA assessments, and provides a quick overview of the new architecture that uses the Hyperledger Fabric blockchain framework[1] hosting Fabric Private Chaincodes.[2]

1.1 Defining Service Level Agreements (SLAs)

In many business relationships, one actor (the provider) promises to deliver a service to another actor (the client) in return for a certain fee. As part of their contract, the two actors sign a Service Level Agreement (SLA) that defines the service(s) to be delivered.

In the SLA, the provider promises the service will meet certain defined metrics such as availability, quality of service (QoS), reliability, responsiveness of technical support, and so on.

SLAs are used in many industries and defined in different ways. For instance, an SLA for supplying aeronautical data by the International Civil Aviation Organization (ICAO) focuses on metrics such as data accuracy, resolution, and timeliness.[3]

Most SLAs for the cloud services industry focus on metrics such as availability, reliability, and issue recurrence. This type of SLA often defines outage severity levels such as critical, urgent, serious, and so on, and prescribes target response times for each level of outage.

1.2 Drawbacks of Conventional SLA Assessments

When the actual metrics of a service received by a client do not meet the metrics promised by the provider, the agreement between the parties has been violated. Any kind of deviation from the agreement is called an SLA violation in this white paper. Any SLA violations are handled in a predefined way by the smart contract involved, as described in section 3. Depending on the severity level, SLA violations can lead to refunds for the client or even legal consequences for the provider.

This raises two immediate questions:

- Who determines whether the SLA has been violated?
- And how is that determined?

In conventional SLAs, the provider assesses the metrics of a service using their own tools and frameworks. Unfortunately, this increases the likelihood of biased results that may disregard the client's interests. This undermines trust and may interfere with sales efforts by the provider and purchasing decisions by the client.

When the provider assesses SLA metrics using their own methods, the client could well suffer from misunderstandings, lack of transparency, and insufficient refunds.

Misunderstandings: Client requirements for SLAs are becoming more advanced and more specific. This means more sophisticated tools and frameworks are required to compute these metrics. But the client could easily misunderstand how the provider's methods work and wonder if the assessments are done fairly.

Lack of transparency: For financial gain, the provider could deliberately hide the methods used to compute the SLA metrics from the client. For example, the provider's framework might not trigger every minor or temporary SLA violation as stated in the SLA. But the client would never know.

Insufficient refunds: The lack of clarity and transparency around the SLA metrics could result in under-reported violations, fewer refunds, and lower compensation to the client. This would match neither the spirit nor the letter of the SLA.

Many service providers are huge multinational enterprises such as Amazon (AWS), Google (GCP), or Microsoft (Azure) which completely control the SLA monitoring and computation

processes. Yet many clients are small local businesses a fraction the size of these providers. Any small client has a limited ability to question any billings or change any policies of a much larger provider.

1.3 A New Approach Using Blockchain

Many telecos are exploring a better way of handling SLAs with advanced software that can provide more automation, objectivity, and precision.

This white paper presents a novel approach to monitoring SLA agreements in clear view of both the provider and the client. This approach overcomes the drawbacks outlined above with a holistic ecosystem that provides computational transparency between each provider and client bound by an agreement, and preserves privacy for each particular SLA.

This approach uses a permissioned blockchain network to register each new SLA, along with:

- An immutable description of its metrics
- Specifics on how each metric is measured and computed
- The agreed-on refunds or penalties for any violations

Each SLA is automatically monitored and its metrics are measured by a specialized smart contract and the other components described in the rest of this white paper.

Any SLA violations trigger the agreed-on workflow. For example, a minor SLA violation could be considered as a warning for the provider. A moderate SLA violation could release a refund payment to the client. A more severe SLA violation could downgrade the reputation of the provider, cause the provider to be barred from the network, or begin a legal proceeding against them.

SLA self-assessment relies on blockchain technologies to tackle the gray areas of conventional SLA assessment. The solution presented in this white paper constitutes a unique architecture that addresses many of the privacy and trust issues in similar approaches.[11-17]

This solution provides a trusted and privacy-preserving monitoring network that can be used for SLAs in telecom and many other industries. Achieving effective SLA self-assessments constitutes an industry first that benefits everyone in the ecosystem by building trust, removing friction, streamlining processes, and saving costs.

2. Standard SLA Assessment Processing

This part describes the major elements of any SLA assessment process, including how to monitor and compute performance metrics, and the diverse products and services provided within a Trusted Execution Environment (TEE). This part also introduces the concept of SLA self-assessment as a desirable goal that can be achieved with the prototype system described.

2.1 SLA Monitoring and Computation

In order to monitor a Cloud Infrastructure-as-a-Service (IaaS) SLA, some specific questions must be answered:

- Which parameters are contained in the SLA?
- How are these SLA parameters computed?
- Do the computed parameters match the SLA's definitions of the IaaS?

The SLA must answer these questions by precisely defining the performance metrics, exactly how to compute each one, and the crucial points where a service can be considered not to meet the promised criteria and the SLA is violated.

Computing these metrics depends on factors such as the sampling rate, the period of evaluation, and the exact formula used to calculate each parameter. In the prototype system described in this white paper, these metrics are computed and any violations are processed through smart contracts, without any third-party intermediation.

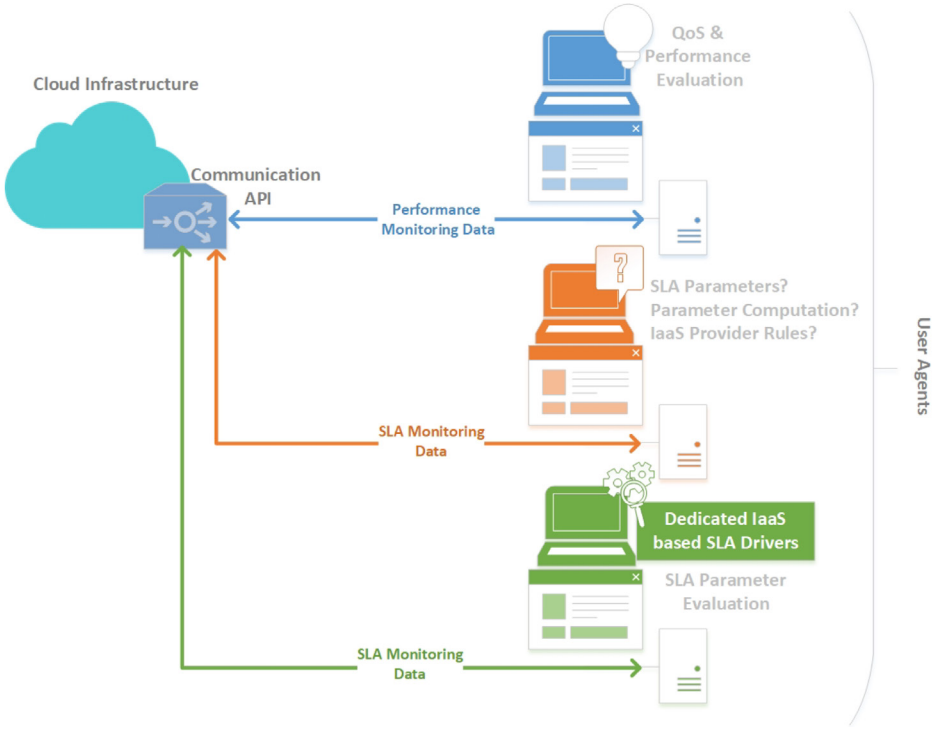

FIGURE 1: STANDARD SLA MONITORING PROCESS[4]

Figure 1 illustrates the standard SLA monitoring process from the perspective of the two actors in a Cloud IaaS SLA. In current cloud systems, the user agents receive performance and SLA monitoring data through the Communication API as this data is computed on the cloud provider infrastructure. The monitoring and computation of the parameter values is not transparent to the client, since all this processing occurs at the internal premises or private frameworks of the provider.

In this context, the provider not only delivers services, but plays a second role as a third-party intermediary between the assessment values and the data available to the client through the user agents endpoints. Although the calculated metrics are very valuable for assessing the performance of the cloud infrastructures, they cannot be used as metrics for the SLA assessment.

SLA assessment metrics describe explicit service guarantees in more qualified ways than a one-dimensional parameter value. And most of those metrics cannot be controlled directly by the IaaS. Therefore, the SLA parameter evaluation and computation are significantly influenced by the IaaS assessment procedures.

2.2 Telecom Ecosystem Perspective

In any telecom ecosystem, different providers offer a wide array of products and services and play different roles:

- Service vendors may offer consulting, outsourced, or managed services
- Software vendors may offer applications used in fixed and mobile telecom environments
- Equipment or device vendors may provide physical infrastructure and endpoints
- Systems integrators or value-added resellers may deliver a comprehensive fixed and mobile networking solution
- Network operators may provide wireline or wireless network services

With so many different types of providers, there can be many different practices for managing client relationships. For instance, the user experience may subscribe to different types of care, such as high-touch or self-service. What the client can expect from their type of promised care must be defined in the specific SLA.

2.3 Benefits of SLA Self-Assessment

The concept of SLA self-assessment differs from conventional SLA monitoring methods in certain important ways. As discussed above, conventional SLAs cannot be self-assessed, since the provider has far more control over the monitoring and computation procedures.

The prototype presented in this white paper provides a secure and private computing environment where an SLA can be monitored and all its metrics computed inside the immutable ecosystem.

With this solution, the provider can propose parameters and define all the specifics within the SLA. At the same time, the client can engage confidently in a secure ecosystem, trusting that their SLA metrics are being monitored and calculated with integrity and precision.

Computational transparency and data privacy are assured through the dedicated workflow of the Trusted Execution Environment (TEE). The entire SLA processing can unfold as determined by the smart contracts deployed by the TEE. This ensures that all SLA metric data is monitored and calculated properly, and kept isolated and private from all other parties.

This solution enables SLAs to be self-assessed following the rules agreed on when the initial contract was signed.

The main benefits of this approach are:

- Minimizing any misunderstandings over the contractual terms
- Delivering computational transparency for both provider and client
- Working around any conflict of interest in the SLA assessment
- Building trust between the client and the provider

3. Blockchain Architecture for SLA Self-Assessment

This part describes the architecture of the blockchain prototype for handling SLA self-assessments, including an overview of the system, how the Signed Parametric SLA is built, and the layered configuration of the algorithmic drivers.

3.1 System Overview

This prototype addresses the need for trust and confidentiality by storing all the SLA operational intelligence within the secure barriers of a permissioned network. Transparency of computation results from the on-chain agreement over the selected SLA evaluation methods used to monitor the SLA logs. Data privacy is established by the corresponding traits and capabilities of the on-chain TEE.[5]

Therefore, this approach achieves true and reliable SLA self-assessment. The client can now be assured that the SLA computations match the promises made at the outset of their contact with the provider.

The SLA Trusted Monitoring takes place within a permissioned blockchain network that enjoys the benefits of Distributed Ledger Technologies (DLTs) outfitted with on-chain TEEs.

The system builds on Hyperledger Fabric DLT software[6] which supports a dedicated on-chain TEE by using Hyperledger Fabric Private Chaincode (FPC)[2] as the dedicated enabler. This extends the on-chain framework to create and deploy protected, isolated environments for running smart contracts.

Figure 2 shows a high-level view of the blockchain-enabled SLA Self-Assessment system. Both the provider and the client use the same standardized process, which includes the on-chain processes plus the explicit off-chain contacts.

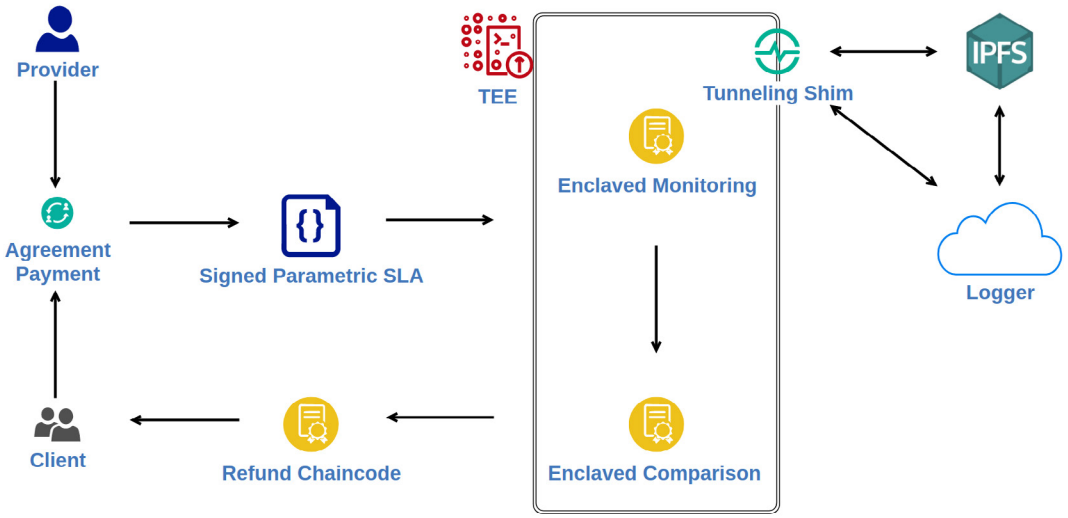

FIGURE 2: BLOCKCHAIN SLA SELF-ASSESSMENT ARCHITECTURE[4]

Here is a brief description of the components in this SLA process and how they interact.

The platform is based on a permissioned blockchain network. Both the two main actors, the provider and the client, are members of the network.

The **Agreement Payment** takes place when a client buys an SLA product offered by the provider. The Agreement Payment is a blockchain transaction validated by the signatures of both parties.

The Parametric SLA is a generic template for a digital document, with many empty fields. The **Signed Parametric SLA** populates this template with the required SLA data specified by the Agreement Payment plus the verified digital signatures of both parties.

After the Parametric SLA is signed, the **on-chain TEE** is notified. The TEE adds the Signed Parametric SLA as a new agreement to its registry of SLA self-assessments and begins to provide the enclaved operations.

The TEE is continuously supported by the **Enclaved Monitoring**, which gets the SLA metrics via the Tunneling Shim. The **Tunneling Shim** is a special integration component that enables the connectivity of the **Cloud Logger** and **InterPlanetary File System** (IPFS) while securely controlling all external integration points of the blockchain network.

The TEE also supports the **Enclaved Comparison**, which computes any SLA violations. If the necessary conditions for any violation are met, this component executes any required Refund Chaincode.

Any refunds are made as agreed in the Signed Parametric SLA.

3.2 Building the Signed Parametric SLA

The Signed Parametric SLA must include all the required information from the provider's SLA document in a specified data schema format. Regarding the schema, the Parametric SLA follows the ISO 19086-2 SLA standard[7] which provides the necessary instructions for creating the essential classes of the schema.

The SLA document is converted via this technique into a JSON document, which is then fed into the algorithmic drivers for SLA evaluation.

Certain key pieces of information must be specified to build the JSON schema that serves as the Signed Parametric SLA, including:

1. **Metrics:** This class represents all the measurable goals for the service guaranteed by the SLA. For instance, one popular metric in many SLAs is availability. This class also contains basic information about each metric to make it possible to monitor and measure it.
2. **Parameters:** This class completes each metric by listing all of its parameters. This class also describes the kinds of variables that express each metric and how to measure them.
3. **Rules:** This class defines the rules by which these metrics are measured. The most common application of a rule is to define what counts as failure or success for a certain metric. For example, the Amazon AWS SLA defines a service as unavailable when it is not reachable in two availability zones.[8]

Parameters also affect how the SLA metrics are calculated, and thus how the algorithmic drivers operate. These algorithmic drivers must account for all the data the Signed Parametric SLA provides.

The rules can have such a significant impact on how metrics are calculated that if two providers ever calculate the same data, they must use separate algorithmic drivers. In the Parametric SLA, the algorithmic drivers follow the standardized development approach as specified in the SLALOM cloud specification model.[9]

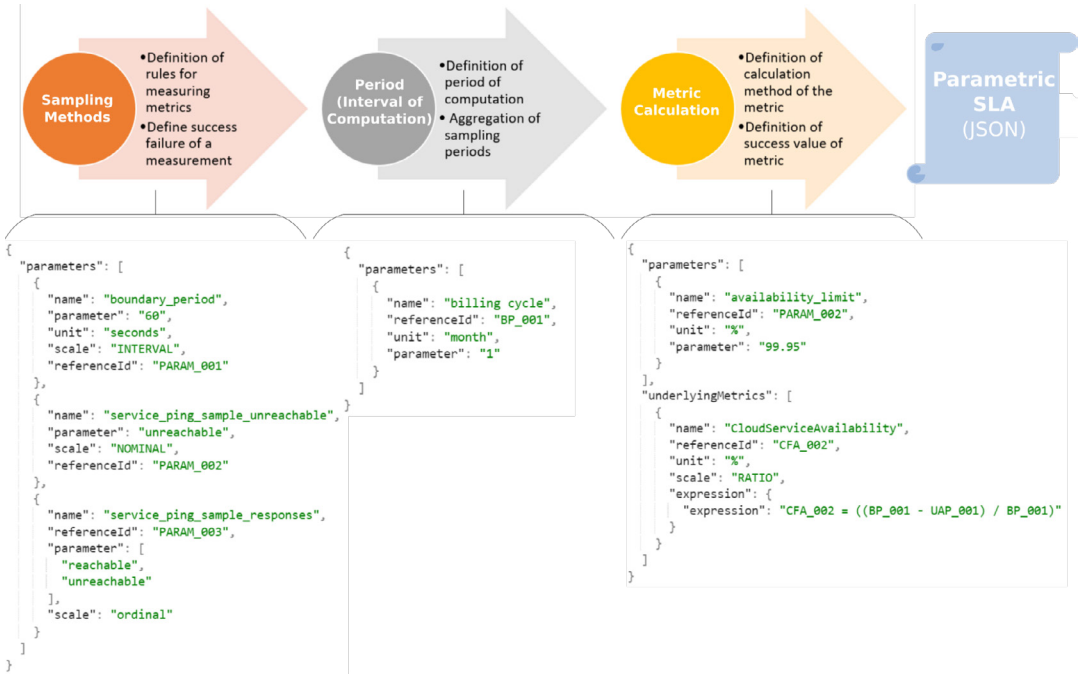

FIGURE 3: LAYERED CONFIGURATION OF ALGORITHMIC DRIVER[4]

3.3 The Layered Configuration of the Algorithmic Drivers

Figure 3 shows the layered configuration of the algorithmic drivers in the system. Here are some further notes:

The Sampling Methods layer develops the procedures for gathering sample data and assessing its reliability for calculating particular metrics. The constraints that result from the rules are represented in Boolean form. These constraints determine whether a particular sample will be used to compute the metric or rejected as invalid.

The Interval of Computation layer determines the data sampling rate used to calculate the metrics and the period in which the metrics are assessed. The billing cycle and the SLA evaluation cycle are provided by the SLA contract. Depending on the service and its defined parameters, the evaluation period can be anywhere from once a month to once every two years.

The Metric Calculation layer handles the actual computation of the metrics for any given period of time. The sampling data is processed to create the SLA metrics using specified functions based on the definition and rules for each metric.

After the configuration is successful, the Signed Parametric SLA becomes the on-chain proof of the agreement. This provides all the necessary information, including the wallet addresses of the parties, SLA metrics information, along with the algorithmic drivers used during SLA enclaved operations.

The TEE records this new agreement and the Signed Parametric SLA on the ledger and adds it to the portfolio of dedicated enclaves. From then on, that SLA benefits from the SLA Trusted Monitoring.

3.4 SLA Trusted Monitoring

The SLA Trusted Monitoring for a new agreement starts after the Signed Parametric SLA is communicated to the on-chain TEE. Since the entire scenario evolves on-chain, both the provider and the client can trust the associated SLA and its data.

The SLA monitoring and computation are transparent to the provider and the client, but secret from all other blockchain network entities outside the enclaved environment.

As soon as the Enclaved Monitoring component adds the new agreement to the dedicated enclave's portfolio, the most recent SLA logs are obtained and analyzed inside the TEE. The enveloping FPC ensures the privacy of all actions of the hosted components: the smart contract structures of Enclaved Monitoring and Enclaved Comparison.

The Enclaved Monitoring is deployed and operates automatically inside the FPC. Every other entity in the network that can read the common ledger is excluded from the FPC's blockchain activity.

The Enclaved Monitoring consists mainly of specialized smart contract functions that recognize any freshly signed agreements and use the Tunneling Shim to retrieve the most recent agreement logs from the IPFS network.

The entire workflow is completed inside the chaincode enclave using these dedicated privacy features.

The Cloud Logger component continuously broadcasts fresh log files of an agreement to the IPFS network during the life cycle of the workflow. The Tunneling Shim uses a unique Content Identifier (CID) to fetch an agreement's logs, since IPFS provides a content-addressed and versioned P2P file system.[10]

The Enclaved Monitoring connection with the IPFS nodes is achieved via the Tunneling Shim. The most recent SLA logs for a certain agreement are retrieved through the Tunneling Shim, sent to the Enclaved Monitoring component, and then forwarded to the Enclaved Comparison component.

The Enclaved Comparison is a special smart contract structure that executes automatically inside the installed FPC. The Enclaved Comparison is triggered only when the Enclaved Monitoring announces the logs. The SLA computations are carried out by the Enclaved Comparison smart contract which calculates any potential SLA violations.

To establish an SLA violation, the smart contract uses all the relevant information from the agreement, including the real metrics from the log files, the agreement's algorithmic driver, and the SLA agreement metrics.

This computation is done within the FPC's private, isolated environment using the relevant TEE features. This keeps all computations secure from all third parties using the blockchain network.

The final result of the calculations made by the Enclaved Comparison within this privacy-preserving environment is a fair conclusion that follows the original, clearly stated, and mutually accepted rules of the SLA.

The final result might or might not signal an SLA violation. In the event of an SLA violation, the Enclaved Comparison triggers the Refund Chaincode to be executed. The Refund Chaincode actively receives all the SLA agreement data required.

As a multi-purpose chaincode, the main role of the Refund Chaincode is to carry out the agreed-on compensations for violations. A penalty and reward scheme is used to define the refund mechanism used by the Refund Chaincode.

If an SLA violation occurs, this refund mechanism is triggered. This manages the system's financial balance and provides the client with compensation for any violated SLA terms. The client is rewarded for the violation and their wallet balance is increased, while the provider is penalized and their wallet balance is decreased by the same amount.

Other possible actions of the Refund Chaincode include altering an actor's reputation, altering the product score, and others. The Refund Chaincode forms the final component in the SLA violation process that completes the SLA Self-Assessment solution life cycle.

4. Conclusions

This white paper describes an innovative approach to SLA self-assessment using the Hyperledger Fabric blockchain framework. The DLT software that hosts TEEs with isolated computation serves as the foundation of the solution. All activity takes place within a permissioned blockchain, which creates a secure system that provides accuracy and fairness to both the provider and the client.

This system transparently monitors and evaluates every SLA for both the provider and the client, and preserves the privacy of both throughout the SLA life cycle.

Since SLA agreements can now be independently evaluated inside an honest and trusted ecosystem, this system achieves true SLA self-assessment. This delivers all the associated benefits: fewer misunderstandings, more transparency, fairer handling of any SLA violations, and ultimately more trust.

Also, this solution can scale for enterprise use cases.

Future work will focus on providing more actions in case of SLA violations, such as reputation management and product scoring.

Acknowledgements

The Hyperledger Telecom Special Interest Group would like to thank the following people who contributed to this white paper: Nima Afraz, David Boswell, Gordon Graham, Nikolaos Kapsoulis, Antonios Litke, Alexandros Psychas, Vipin Rathi, and Theodora Varvarigou.

Appendix A: Related Research

Table 1 shows some of the research literature that touches on the topic of SLA self-assessment. Much of this research aims to solve problems in neighboring areas without addressing the entire domain.

Note that the prototype system described in this white paper includes all the properties listed in the Neighboring Area column.

To see the specific papers noted in Table 1, please use the Notes section.

AUTHOR	TOPIC	NEIGHBORING AREA	APPROACH TO SLA SELF-ASSESSMENT
Nguyen et al[11]	SLA Assessment	SLA evaluation	Private from third parties
Ranchal et al[12]	Multi-Cloud SLA	SLA monitoring	Different rules within a single blockchain
Alowayed et al[13]	Cloud IaaS Evaluation	Privacy-preserving protocol	Distributed and enclaved operations
Uriarte et al[14]	Dynamic SLA	SLA provisioning	Dynamic SLA self-Assessment
Alzubaidi et al[15]	SLA Compliance	SLA dependability validation	Trustless of cloud IaaS
D'Angelo et al[16]	Cloud Accountability	Blockchain SLA monitoring	SLA Self-Assessment
Tan et al[17]	Performant SLA	Cloud IaaS supervision	No on-chain intermediaries

TABLE 1: A COMPARISON OF RESEARCH LITERATURE AND SLA SELF-ASSESSMENT

Notes

1 Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger Fabric: A distributed operating system for permissioned blockchains. In Proceedings of the thirteenth EuroSys conference (EuroSys'18), Porto, Portugal, 23–26 April 2018; pp. 1–15.

2 Hyperledger Fabric Private Chaincode. Available online: **https://github.com/hyperledger/fabric-private-chaincode**

3 Service Level Agreement Template. Available online: **https://www.icao.int/WACAF/Documents/APIRG/APIRG18/Docs/wp17_appendices_en.pdf**

4 Kapsoulis, N.; Psychas, A.; Litke, A.; Varvarigou, T. Reinforcing SLA Consensus on Blockchain. Computers 2021, 10, 159. **https://doi.org/10.3390/computers10120159**

5 Sabt, M.; Achemlal, M.; Bouabdallah, A. Trusted Execution Environment: What It is, and What It is Not. In Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, Finland, 20–22 August 2015; pp. 57–64.

6 Hyperledger Fabric. Available online: **https://www.hyperledger.org/use/fabric**

7 ISO/IEC 19086-2:2018 Cloud Computing—Service Level Agreement (SLA) Framework—Part 2: Metric Model. Available online: **https://www.iso.org/standard/67546.html**

8 Amazon Compute Service Level Agreement. Available online: **https://aws.amazon.com/compute/sla**

9 SLALOM SLA Specification and Reference Model. Available online: **https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5aa6eccf3&appId=PPGMS**

10 Benet, J. IPFS–Content Addressed, Versioned, P2P File System. 2014. Available online: **https://ipfs.tech**

11 Nguyen, T.-V.; Lê, T.-V.; Dao, B.; Nguyen-An, K. Leveraging Blockchain in Monitoring SLA-Oriented Tourism Service Provisioning. In Proceedings of the International Conference on Advanced Computing and Applications (ACOMP), Nha Trang, Vietnam, 26–28 November 2019; pp. 42–50.

12 Ranchal, R.; Choudhury, O. SLAM: A Framework for SLA Management in Multicloud ecosystem using Blockchain. In Proceedings of the IEEE Cloud Summit, Harrisburg, PA, USA, 21–22 October 2020; pp. 33–38.

13 Alowayed, Y.; Canini, M.; Marcos, P.; Chiesa, M.; Barcellos, M. Picking a Partner: A Fair Blockchain Based Scoring Protocol for Autonomous Systems. Proc. Appl. Netw. Res. Workshop 2018, 33–39.

14 Uriarte, RB, Zhou, H, Kritikos, K, Shi, Z, Zhao, Z, De Nicola, R. Distributed service-level agreement management with smart contracts and blockchain. Concurrency Computat Pract Exper. 2021; 33:e5800. **https://doi.org/10.1002/cpe.5800**

15 Alzubaidi, A.; Mitra, K.; Patel, P.; Solaiman, E. A Blockchain-based Approach for Assessing Compliance with SLA-guaranteed IoT Services. In Proceedings of the IEEE International Conference on Smart Internet of Things (SmartIoT), Beijing, China, 14–16 August 2020; pp. 213–220.

16 D'Angelo, G.; Ferretti, S.; Marzolla, M. A Blockchain-based Flight Data Recorder for Cloud Accountability. In Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems, Munich, Germany, 15 June 2018; pp. 93–98.

17 Tan, W.; Zhu, H.; Tan, J.; Zhao, Y.; Xu, L.D.; Guo, K. A novel service level agreement model using blockchain and smart contract for cloud manufacturing in industry 4.0. Enterp. Inf. Syst. 2021, 1–26.