International Journal of Web Services Research
Volume 18 • Issue 1

Investigation on Blockchain Technology for Web Service Composition:
A Case Study

Sridevi S., PSG College of Technology, India*
Karpagam G. R., PSG College of Technology, India
Vinoth Kumar B., PSG College of Technology, India
Uma Maheswari J., PSG College of Technology, India

ABSTRACT

The blockchain is an incorruptible digital ledger of economic transactions that can be programmed to record not just financial transactions but virtually everything of value. Blockchain technology makes breakthroughs in business intelligence in many areas such as banking sector, finance, judiciary, commerce, and information technology. Web service compositions have a revolutionary impact on business intelligence by enabling loose coupling, data consolidation from diverse sources, consolidation of information under a single roof, easing ad-hoc querying and reporting. The objective of current work is to investigate the applicability of blockchain for the semantic web service composition process. The paper focuses on design of conceptual architecture and the algorithm for QoS-aware semantic web service composition (SWSC) using blockchain.

KEYWORDS

Blockchain, Quality of Service, Semantic Web, Service Consolidation, Service Level Agreement, Smart Contract, Smart Contract for SOA, Web Service Composition

1. INTRODUCTION

A Web service is a piece of software that is used to transmit information between two parties. It may be financial transactions, or text transactions, or data transactions or, media transactions, or Business-to-Business (B2B) interaction. But ordinarily, in the current world, a single service could not able to satisfy the heterogeneous B2B interaction process. This arises a concept called Semantic Web Service Composition (SWSC) process. Thus, there is need for a standard formats and organizing the

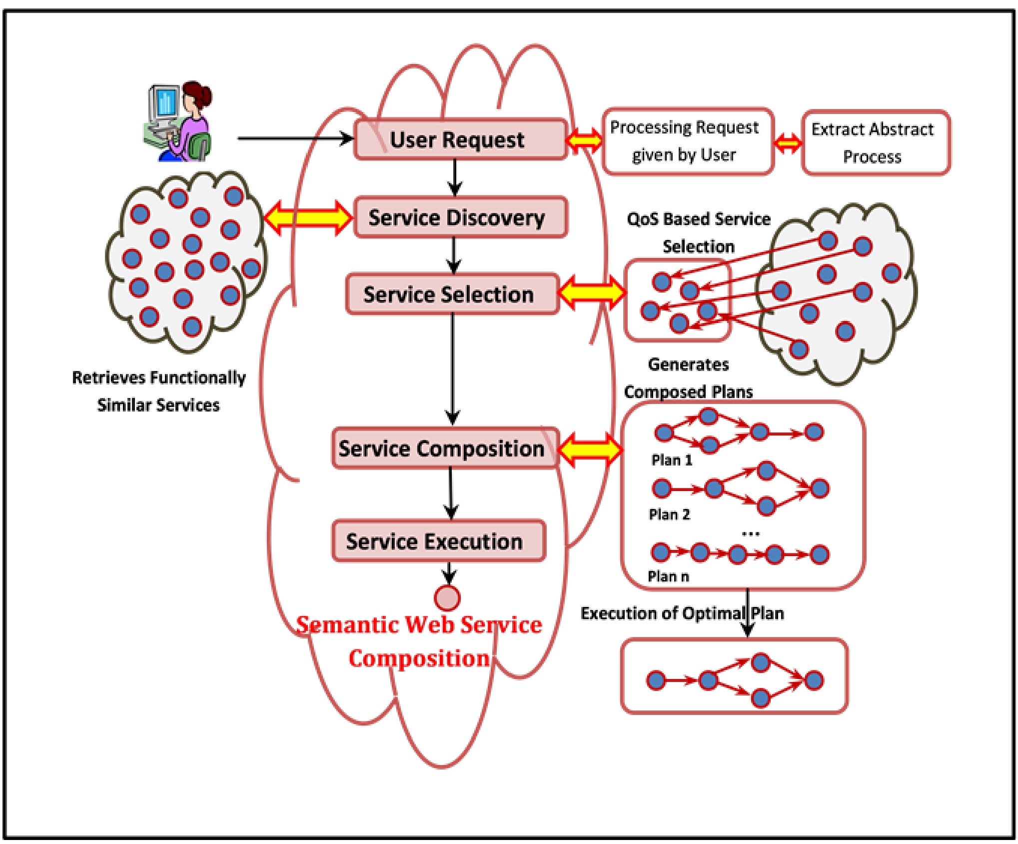
DOI: 10.4018/IJWSR.20210101.oa1 *Corresponding Author

heterogeneous components for enabling B2B interaction to compose the services in a semantic order for seamless execution of services, which brings about a concept called service-oriented-Architecture (SOA). According to (Anji Reddy et al 2012), the technical implementation of SOA is facilitated by a model termed Service-Oriented Computing (SOC). Real-world examples for the SWSC process are ticket booking, holiday-trip booking, e-book purchasing, e-shopping, and e-insurance system, etc. Usually, the SWSC process is attained by four stages: They are: i) Service Discovery – to retrieve functionally similar services, ii) Service Selection- to pick best services based on Quality of Service (QoS) values, iii) service composition – generate various possible plans in the ordering of execution of services, and iv) Service Execution – execute on optimal service.

In the foremost service discovery task, from the given user query request, the discovery process should semantically understand and extracts what are inputs (I) are given by the query, what are the outputs (O) expected by the user, what are the preconditions (P) of the user, and what is the effect (E) of a user (IOPE), then, using matchmaking algorithms like bipartite matching, etc, to retrieves the matched services which reside in the centralized web service repository (UDDI registry) (Amirthasaravanan et al., 2016). Then the best services are picked from the discovered list in terms of service-constraint QoS values and user-demanded QoS values (Meysam Ahmadi Oskooei & Salwani Mohd Daud, 2014), (Yaswanth, 2016), and (Maheswari & Karpagam, 2015). Then the top-ranked services based on the weightage given for specific QoS attributes are retrieved and considered for the composition process (Aram AlSedrani & Ameur Touir, 2016).

Accordingly, several possible composition plans are generated by combining various selected individual services using composition methods like colored Petri-nets, Sword, OWL-X plan planner,

Figure 1. Different phases of semantic web service composition



etc. In some cases, the service composition order may be random. The services are executed in parallel if there is no dependency between services. If services are dependent then the output of one service is given as the input for another service. After that, optimal plans are generated by considering dependencies. Working on different web service composition phases is illustrated in Figure 1.

However, there is a chance for service providers to provide counterfeit information about their service functionality or instant shutdown of services, or removing any mock service information from the registry. Meanwhile, the users do not have the ability to check the background and origin of web services. The commercial purpose web services are grievous could contain counterfeit provider's information that can even expose to data breaches. The vast majority of these platforms are proprietary in nature, giving no negotiating power for users to fully trust the platform. The objective of the paper is to propose a smart contract enabled services provider agreement (smart contract instead of SLA agreement) and QoS aware service composition in order to enhance trust, clear communication, efficiency, and secure execution of web services.

The Ethereum Blockchain is considered a second-generation cryptocurrency that uses the proof-of-stake (PoS) consensus mechanism (Schwab, 2017). Rather than leveraging the blockchain platform solely for monetary transactions, the Ethereum Blockchain platform views itself as a network for general purpose decentralized, distributed, and trustworthy platform for developing real-world E-applications scenarios, which is technically denied in the concept of current financial transaction blockchain platforms (Z Zheng et al., 2017).

A smart contract-enabled blockchain web service platform is built in order to radically change the traditional way of publishing and recommending the services by centralized parties to self-determining, and self-executing community platforms, which revolutionizes the internet services (Meysam Ahmadi Oskooei & Salwani Mohd Daud, 2014). As inspired by the current blockchain research works incorporation of Blockchain in IoT platform (Hamza Baqa et al., 2019), and law enforcement, etc, the objective of the study is to incorporating SWSC process in Blockchain platform in order to promise the trustworthiness of web services.

The main contributions of the paper are summarized as follows:

1. As the conventional centralized accessing of SLA agreements does not have the ability to check the background and origin of web services and also has a chance to compromise data breaches. Initially, a new novel trusted SLA-based agreement is created and stored in Blockchain instead of storing in a conventional centralized platform.
2. The governing contract monitors and verifies the functional and non-functional values which are maintained within bounds as per in published SLA contract. This avoids the retrieving of unpleasant services from the fundamental level itself. Here, the existing bi-partite matchmaking method, and Fuzzy-Topsis based selection method are implemented in the smart contract version in order to verify the correct execution of services.
3. Extensive experiments have been conducted on OWLS-TC datasets over the truffle ethereum framework to evaluate the effectiveness of the smart contract-based QoS-aware composition model for semantic web service composition in terms of precision, recall, F-measure, and accuracy. The experimental results demonstrate that the proposed smart contract-based QoS-aware composition model achieves substantial progress over conventional SLA-based composition models.

The rest of the paper is laid out as follows. Section 2 gives an overview of relevant works, while Section 3 represents the perceptual overview of creating and publishing SLA agreements for blockchain and a functional and QoS-aware service composition model. Whereas, section 4 demonstrates the proposed Semantic Web Service Composition in the Ethereum Blockchain platform. Section 5 depicts the successful execution of the recommended approach in the taken case study. The experimental analysis and discussion are explained in section 6. Finally, section 7 provides a conclusion which is followed by references.

2. RELATED WORKS

2.1 Semantic Web Service Composition Process

Web service technology plays a vital role in developing distributed applications that are accessed via the internet. OWL-S or WSDL standard is used to describe the functional aspects of web services and OWL-Q is used to describe non-functional aspects (QoS). The WSDL file is used to capture only the syntactical structure but, the OWL-S file facilitates to capture of the semantic relationship between the concepts easily. So, the OWL file allows semantic retrieval of all matched services easily. The discovery process retrieves the functionally similar services from the centralized registry like UDDI using matchmaking algorithms.

The Web service is a single piece of software that performs a particular task. These tasks are carried out through a business process. In the web service composition model, it can be noted that the overall process represents the operation of each web service, whereas each activity represents an operation on the web services of the provider. As a consequence, a trusted bond is required between the service user, service provider, and middle-man for executing the service. This necessitates needing a contract called SLA agreement (W3C RIF-WG, 2010). In recent years, the telecommunications industry shows a rapid growth to sort earth in hand, it would espouse new networking technologies at shorter time scales. For that, SLA standards are the only resources that support and enable consumers to adopt these emerging service provider's technologies reliably. The SLA Management Handbook of the TeleManagement (TM) Forum highlights the importance of SLAs for the telecommunications industry (John J. Lee & Ron Ben-Natan, 2012). As the SLA providing some trusted functionality, an SLA is developed for SOA (Philip Bianco et al., 2008). In SOA, the provider-side SLA, and administrator SLA's represent the functionality information and set of activities that they perform. Based on these SLA values web services are discovered and selected in SOA (Stephen et al., 2007) and (Andrea Zisma, 2011). The Web Service Level Agreement (WSLA) describes the functionality of web services and non-functional properties like availability, accessibility, reliability, performance, and accuracy of the web to ensure that the right information reaches the right person at the right place at the right time, in a safe and secured manner (Nagy Ramadan, 2015).

As per functional properties (OWL-S) and Non-functional properties (OWL-Q) in the SLA agreement, web services are discovered and selected for the specific consumer. In the discovery process, functionally similar services are retrieved using matchmaking algorithms and the best services are picked form the discovered service for service composition. However, a great challenge arises in any one of these stages like modifying functional values with counterfeit values in order to accept the bogus services and access the like of the user or counterfeit the service QoS values to make their service to be in the selection list. Thus, most of the researchers work in the area of trustworthiness of services in order to promise the credibility of the services.

2.2 Blockchain

Blockchain is a distributed, decentralized, and immutable public ledger technology that has gained surplus attention over the current years. Initially, the Blockchain concept is emerged for executing the financial transactions in a secured way and stored them in chronological order in a distributed, immutable and permanent fashion (Schwab, 2017). Bitcoin is the first electronic digital currency that allows participants to conduct secure payment transactions without the involvement of a trusted third party (Imran Bashir, 2018). As of its technical features, various advantages of Blockchain are security: homomorphic hash value is created for each transaction and stored in chronological order. Privacy: as there is no middle party to transact the data, the confidentiality and integrity of data are well-preserved. Availability: As the transactions are distributed storage the data is retrieved from a nearby cloud area which allows fast accessing of services. Consistency: the miners in the blockchain check wheather all the data in a distributed blockchain database is stockpiled in a consistent manner. Immutability: ensures no data is tempered by any centralized parties or middle-parties. Reliability: No single point of failure is happened due to the distribution of storage. Efficiency: all interactions between services are processed automatically according to pre-determined protocols at a low cost. Trust: assured by secure and consistent transactions (Shanti Bruyn, 2017).

The above-mentioned properties are achieved by two fundamental things namely miners and homomorphic hash values. The homomorphic hash value is a value generated once the transaction is executed and stored in the blockchain. The verification and validation of transactions are performed by miners. In the centralized system, only one middle party is verified and validates the transactions. But, in the blockchain for the execution of one transaction, more than one miner verifies and validates the transactions. As this process is executed in a distributed manner, the miners are not known to each other. Also, the validation of transactions they performed also in their own way. Thus no one can control the interactions. If any miners broadcasted falsify data, other miners easily find the counterfeit messages and triggers the counterfeit notification to the network. Thus, no counterfeit message is broadcasted in the network. It preserves the trustworthiness of the network. The miners can use the defined consensus mechanism to broadcast the transaction. A consensus mechanism is a rule, which should be satisfied by the miners, and based on that rule miners should add the transaction to the network (Wang et al., 2019). There are different types of consensus mechanisms is followed by the Blockchain. Some of them are Proof-of-Work, Proof-of-Stake, Byzantine-Fault-Tolerant (BFT), Distributed Byzantine-Fault-Tolerant (DBFT), Poof-of-Capacity, and Proof-of-Storage, etc.

For these properties, currently, the technology is attempt to incorporate in various fields namely land registry to historical record maintenance about the proprietor of the land (Arturo Castellanos & Raquel Benbunan-Fich, 2018), Financial applications for secure transactions (Fanning and DP centers, 2016), Business for trustworthy agreements, governance applications to maintain regulatory boards (Nomura Research Institute 2016), Medcare and healthcare for secure sharing of medical images, healthcare records and preserving it without compromising of privacy (Nomura Research Institute 2016), law and legal governance in order to avoid the legal standards are in the control hands of middle-parties (Robert Heria et al, 2018), E-voting (Nir Kshetri & Jeffrey Voas, 2018), IoT (Hamza Baqa et al., 2019), education for secure storing of certificates (Nomura Research Institute, 2016), cyber security (Guang Chen et al., 2018) and (Kosba et al., 2016), and Food processing organization (Khaled Salah et al., 2019), etc.

Some researchers tried to incorporate Blockchain technology for accessing web-based services. Hamza Baqa et al. (2019) attempt for incorporating Blockchain technology for discovering IoT devices. The smart contract automatically executes the discovery process by expanding the searching process using domain-specific keywords across distributed IoT services. It retrieves all the semantically matched services which are advertised across various public domains using the matching, indexing, annotation process. As all the services are automatically extracted by the given constraints, it ensures trust over retrieved services. Peter de Lange et al. (2019), proposed a Blockchain service registry, by which the user can explore the services in the registry, make use of it. It also retains the information about the provider's origin and history in a secured manner. Weihong Ca et al. (2019), proposed the Blockchain-based matrix factorization method to predict the trustworthiness of user feedback and predict the reliability of the user. Huan Zhou et al. (2018) proposed a witness paradigm as a service-level agreement for cloud services. Shubham Desai et al (2020) proposed Blockchain for the secure storing of cloud services. It enables the secure automatic allocation of cloud services in a distributed manner. Based on the related works and the lessons learned from the literature, this paper investigates smart contract-based QoS - aware semantic web service composition.

3. PROPOSED ARCHITECTURE OF TRUSTED BLOCKCHAIN-BASED QOS-AWARE WEB SERVICE COMPOSITION

3.1 Perceptual Overview of Smart Contract Agreement

A proposed tamper-proof service provider's Blockchain-based SLA agreement facilitates the trusted environment for service users to access commercial services securely. By combining the transparency, privacy preservation, and immutable property along with a trusted environment of ethereum Blockchain technology allows us to make use of these strengths to compensate for the current deficiencies namely counterfeit information, cost fraudulent and middle-man administration in web service interoperability. Primarily, the blockchain stockpiled the SLA agreement in four forms, which are shown in Table 1. Each form belongs to one block in the blockchain. Since data storage in the blockchain is costly, only essential fields of service contracts are stored in the blockchain, while the supplementary agreement fields like classified lists, other service information's and inessential reports are represented in conventional OWL-S files and published in the Blockchain registry. The Blockchain details are habitually validated via a consensus mechanism and securely referenced by their hash value.

Table 1. The SLA Agreement web service information stored in Blockchain

User Information in Blockchain	**Service Provider Information in Blockchain**		
User Registration username user ID public key ethereum address timestamp email address	**Service Profile OWL-S** Provider name provider ID ethereum address service category IOPE functional categories non-functional Values	**Service Model OWL-S** Access points WS business category Control flow Logical flow A hash value of a supplementary class	**Service Grounding OWL-S** ethereum Address node ID timestamp binding URL binding status tmodel

3.2 Perceptual Overview of Smart Contract-Based Web Service Composition

A proposed smart contract automatically acquires the user request and stimulates the web service composition process. The stimulated composition process discovers services based on functional (IOPE) and non-functional (QoS values) properties from the SLA agreement in Blockchain. When the service provider exemplifies counterfeit IOPE or QoS values in a conventional file rather than offered original IOPE or QoS values in order to make their service to be selected, then at the time of execution, the smart contract script easily found the counterfeit values beyond the bounded level. Here, the miner in the smart contract verifies the dynamic offered service values and agreement static values. When the smart contract identifies the counterfeit problem in specific service instances then it denies the participation of unpleasant services in the web service interaction process. These effective smart contract-based WSC algorithms enhance trust and lead to efficient service discovery, and selection to compose the heterogonous automatic services. Then it automatically generated different service plans and executes one optimal plan via the OWLS-XPlan tool.

4. THE PROPOSED ETHEREUM BLOCKCHAIN-BASED ARCHITECTURE FOR SEMANTIC WEB SERVICE COMPOSITION

Initially, the service provider requests the WSC network to register their service for public usage. Then, the proposed Blockchain network verifies and validates the identity of the service provider and creates the SLA agreement along with IOPE values (OWL-S file) and QoS values (OWL-Q file), and broadcasted the values in the public Blockchain service registry along with the provider's signature using Proof of Stake (PoS) consensus mechanism. Here, visibility of the credential information of the provider is encoded and public view values are only visible by the public user. The homomorphic hash value is created for the broadcasted service for direct B2B interaction. As all the broadcasted values and agreement information is read-only once it is created, then no one can able to change the static values. If required to change the static values, then the existing services should be deleted based on agreement and a new one is created but having a different hash value. When a request is given by the provider via the front end, the miners in the smart contract automatically executes the SWSC process and retrieves all the similar web services using matchmaking algorithms, and top-ranked services is selected by considering user-constraint QoS values using selection algorithms. The broadcasting of service instance files and ordering of SWSC execution process of proposed Ethereum Blockchain-based Semantic Web Service Composition Architecture is illustrated in Figure 2. Whereas, the automatic execution of a smart contract in the real world is illustrated in Figure 3. The fastest composed miner services are given to the user. If the user, uses the services, then miners have a responsibility to take care of the service execution process until a specific user leaves the network. Each information about the service offered functional values, dynamic QoS values, and feedback given by the user, which all are recorded by the miner along with their signature.

4.1 Problems in the Existing Web Service Composition (WSC) Process

A traditional WSC process possess major disputes namely,

Mutable service instance files: As the centralized agent controls the entire WSC process, if the service approaches the centralized agent, then the centralized agent can able to tamper with the SLA agreements for their own profit, which is unknown by the user.

Counterfeit Information: Provider can provide sophisticated QoS values instead of offered QoS in order to make the service selected by a user

Forcing of middle-man exposure to select the service: Every web service transaction is administrated by the middle party. They can access the confidential information of users and providers who participated in SOA.

Trustless Platform for service execution: As it is a mutable platform and controlled by a centralized agent, compromising of a user via backend process and extract their personal information's, credit card details, bank details, credential media files inaccessible devices which are not visible to the user easily. Thus, the trustworthiness of the existing WSC is entirely under question.

Figure 2. Proposed Ethereum blockchain-based semantic web service composition architecture

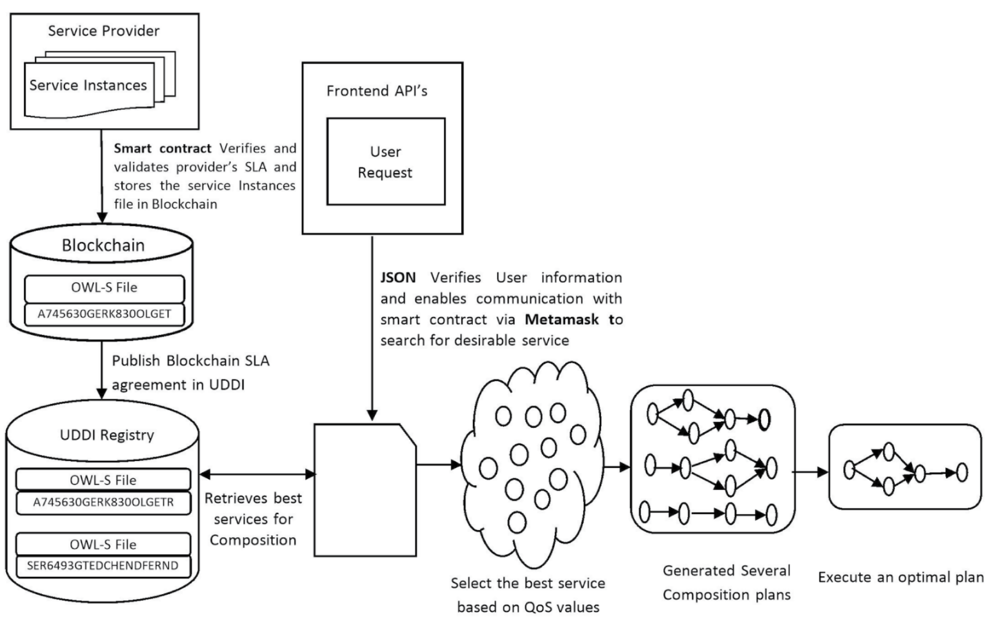
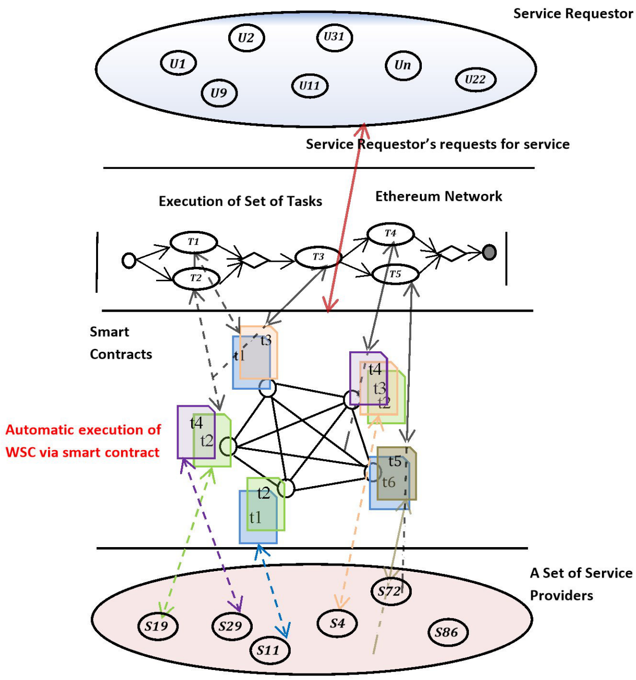


Figure 3. Automatic execution of Ethereum smart contract architecture for semantic web service composition process



4.2 The Blockchain-based solution for the proposed Blockchain-based Semantic Web Service Composition (SWSC) process

Immutable records: Once the static values are fixed then they cannot be changed. Only the allowable bounded limited values are added in dynamic stated along with the verification by a smart contract. The SLA agreement is once agreed upon, and then it cannot be changed. The service should follow the rules.

Clear Communication / Transparency: The blockchain network itself executes the WSC process, which clearly indicates the working of each script and its limitations. As there is no control of a centralized agent, the execution of each process is not be compromised.

Guarantee Outcomes: secure interaction among web services are promised while sharing transactional data, text files, and media files. Nobody can interrupt the interaction process.

Reduces counterfeit actions that may happen in cost transactions: As it is an automated interaction, the service cost is reserved as per the agreement.

Trusted environment: the proposed SWSC process is executed automatically, no one can interrupt beyond their authorization level. Thus the credential information about the user's data, and provider's data, is well-preserved. Each and every transaction is executed as per precise rules. So, there is no chance of triggering any counterfeit measures. If triggered, the WSC system clearly broadcasted the information along with the person who triggers the counterfeit measures. It enables the proposed Blockchain WSC network as a trusted network.

5. IMPLEMENTATION OF ETHEREUM SMART CONTRACT FOR SWSC PROCESS

The SWSC process experiments via Truffle framework which is one of the simulated platforms of the Ethereum blockchain network. Figure 4 depicts the interaction among the SWSC service provider and service requester. Figure 5 shows the API for the stimulated truffle framework. The smart contract (SC) is created using the solidity programming language. The developed solidity contracts are compiled and executed via a solidity compiler.

Figure 4. Interaction between the service requester and provider

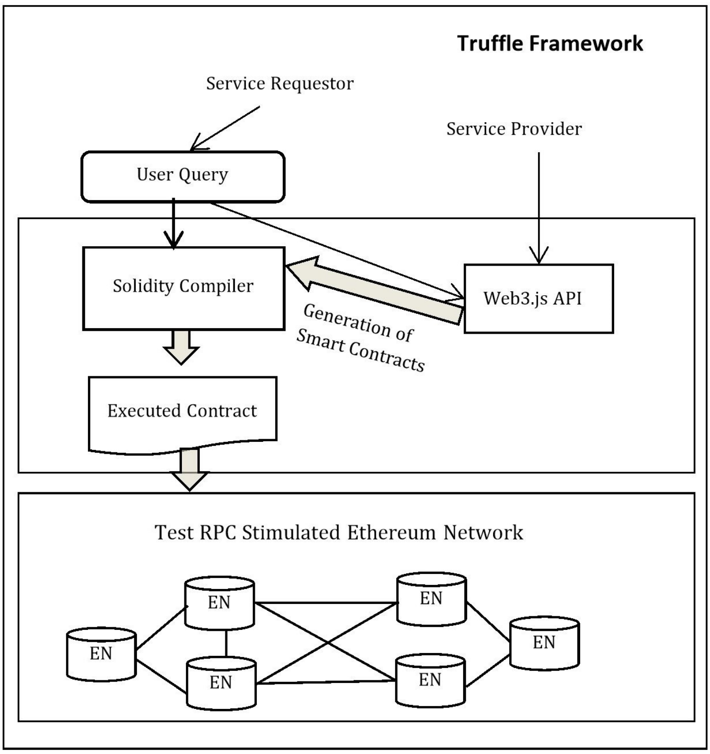
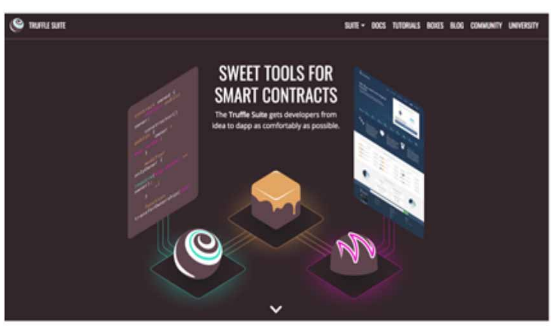


Figure 5. Truffle framework for creating a smart contract



Algorithm: Semantic Web Service Composition

```
TA – Service Assigned for Task, TUA - Service is Unassigned for
Task, S- Service
```

$T_A = \varnothing \; ; T_{UA} = T;$

```
if the service requester request for a service "S";
   Blockchain reads the request and initialize a smart contract
```

for each unassigned task $\forall t_i \epsilon T_{UA}$ and generates the smart contract on a node of Ethereum network;

```
    //Blockchain selects a service provider based on a contract
for each task Ti
while
```

$T_{UA} = \varnothing$, do

for $\forall t_i \epsilon T_{UA}$ do

```
the service requester the services and blockchain executes
discovery process to retrieve similar services as,
```

$\forall r.f_i \epsilon f.T_{i.s}$, r – requester, f-functional values, $T_{i.s}$ - functionally similar services in Blockchain repository,

```
      Evaluating non-functional values to pick the best services
as
```

$\forall s_j \epsilon s_i$ do,

```
select the best service and ranked the list
end
if the requests for the service can be satisfied and desired
services are listed by the provider then,
the smart contract picks a service and generates optimal plans
based on functionality and QoS values;
then execute one optimal plan
```

transfer task T_i and allocated services S_i along with functional and non-functional values in blockchain transaction storage database.

```
end end end
   //composition of service run over the Ethereum virtual network
```

for $\forall t_i \epsilon T_A$ do

```
executed service is given for user
end
```

5.1 Case Study: Applying of Blockchain Technologies for QoS Aware Semantic Web Service Composition

For example, assume a car collides with an accident, instantly, at that point, numerous steps must be taken to deal with the emergency scenario, which is made feasible through web services. The mandatory required web services are t1- ambulance service, t2- traffic official notification service, t3 - GPS location tracking service, t4- primary mobile first aid service, t5- nearby hospital searching service, t6- traffic info service, t7 - police notification service. The accident may be detected by sensor devices fixed in the car, or based on the information given by by-passers. These services are retrieved based on the functional information in the OWL-S file given by the providers like 24x7 multispecialty hospital, 24x7 emergency admission, GPS India location tracking system, 24x7 ambulance service, zonal e-government traffic notification system, emergency call, and zonal first aid primary care center, etc.

In the aforementioned services t1, t3, t5, and t4 are composed parallelly to deal with finding the right hospital, whereas, t1, t6, t5 should compose parallelly to give notification for ambulance driver in order search to reach the hospital within minimum time. t7 service and t2 service can be executed independently. There is no dependency between these two services. Thus, based on the provider's promised functional information, all the functionally similar services are retrieved from the Blockchain WS registry. So, the provider should provide all these promised instances, which are mentioned during the creation of services. So, from the provider side, they are aware of their functionality of service and should provide what they promised. Based on these all the SWSC are executed dependently or individually and discovers similar services.

The next stage is the service selection process. Habitually, the service provider exaggerates the QoS values in the OWL-Q file in order to make their service to be selected in the selection list. For example, for t1 service - medical transport service, the provider promised the sensitive QoS values like availability - 99.8%, reliability - 97.6%, service accessibility - 98.4%, trustworthiness – 99.6%, response time (rs) - 0.003 ms (millisecond). But in real-time, the for t1- service, S7 service provider only offers availability - 47%, reliability - 67.6%, service accessibility - 58.4%, trustworthiness – 48.6%, response time (rs) - 34 ms (millisecond). It does not offer the promised functional and non-functional values. Executing this S7 service leads to a life-critical circumstance that threaded the current SWSC process.

Later, the users notice and give the customer feedback about the service, the service can approach SWSC centralized agent to modify the provided QoS values and feedback values given by the customers by making a favor of the centralized agent in order to increase their own profit. It continuously creates an unfavorable scenario for future requests as well. It creates a pandemic SWSC platform as a completely trustless process. As a result, the service users are not having trust and reliability in this existing centralized SWSC platform. This motivates to incorporate the SWSC process in the Blockchain environment in order to see the enhancement over the trustworthiness of the SWSC process.

Figure 6. Implementation of smart contract

```
pragma solidity >=0.4.25 <0.6.0;

import "./ConvertLib.sol";


contract servicecompose {
    mapping (owl-s SC =>functionality) contract;

    function composeservice(compose) public returns(bool sufficient)
{

type webservice struct {
    discovery    *float64 `json:"serviceowl,omitempty"`
    }

type selection struct {
      selection      *functionalservice  `json:"discover,omitempty"`
 // qos value location
       QoS values    *float64      `json:"qosvalues,omitempty"`
// asset values
          carrier *string       `json:"carrier,omitempty"`
// the name of the carrier

type composition struct {

    compose      *selectedservice  `json:"selected,omitempty"`
 // owl-s XPlan location
       owl-s XPlan tool     `json:" owl-s XPlan tool,omitempty"`
// asset values
          carrier *string `json:"carrier,omitempty"`
```

Figure 7. Execution of a smart contract

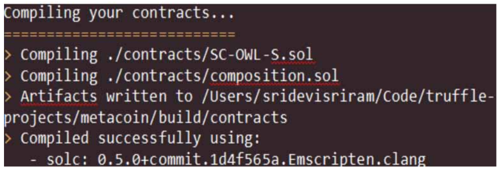


Figure 8. Generation of a hash value contract address for executed SWSC service

For a trial, a new SLA agreement along with the promised functional (OWL-S file) and non-functional values (OWL-Q) are created and broadcasted in the Blockchain registry. The homomorphic hash is created for the service. Each time this service is identified by its name and homomorphic hash value. The SWSC smart contract automatically retrieves the service as per existing SWSC algorithms. The newly developed smart contract for service composition in solidity language is illustrated in Figure 6. It automatically executes the SWSC stages and retrieves the pleasurable services for the given request which is shown in Figure 7. The service discovery is performed by a bi-partite matchmaking algorithm (Uma Maheswari and Karpagam et al 2014), where, the QoS-based selection process is executed by the Fuzzy Topsis method (Maheswari and Karpagam et al 2015). Whereas the retrieved service is composed and executed via the OWLS-XPlan tool this is accessed via metamask from the local system. Then the generated homomorphic hash value for the executed SWSC process is exemplified in Figure 8. As the SWSC composition process is created and executed in the Ethereum Blockchain Platform, it promises credibility, trustworthiness, and reliability for the SWSC composition process.

From the case study, a medical transport service is picked to epitomize the QoS-aware medical transport service selection using both centralized SLA agreements and Blockchain-based SLA agreements which are illustrated in Table 3, Table 4, Table 5, and Table 6. For the implementation of 72 services from Medical care, 28 services from education domains, and 21 services from the communication domain, a total of 122 services are selected. The decision tree C4 approach is used for finding functionally relevant services. The Fuzzy-Topsis method is used for the QoS-aware service selection process. The QoS values considered for the selection process are Availability (Avail), Reliability (Relia), Response Time (RT), cost, and user reputation (Reputation).

In the medical transport service, QoS like availability, response time, and reputation are the most important factors than cost. Based on this the higher weightage is assigned for Availability (Avail), Response Time (RT), and user reputation as 0.25, whereas the reliability weightage, is assigned as 0.15, and the cost is assigned as 0.1. Based on these QoS weightage values, the services are selected. The rank would be assigned using the following formulation

```
Final Rank (QoS) = Availability (w1× q1) + Response Time (w2 × q2)
+ Reputation (w3 × q3) + Reliability (w4 × q3) + Cost (w5 × q5),
Where, w - defines the weightage values and q - defines qos values
```

At different timestamps, the request is given to select the best service. All 122 functionally similar medical transport services are retrieved at different timestamps. But, the service selection varies at different timestamps when using centralized SLA-based – QoS values. During time t1, S82, S102, S14 services are not selected by the SWSC process which is shown in Table 2. Service provider considers the web service interaction and recognizes that their services are not used for business interaction. Therefore, S82, S102, S14 service providers make sophisticated changes in the OWL-S file to forcibly make the SWSC process access their unpleasant service. When the request is given at time t2, then the unpleasant services are selected by the SWSC process which is shown in Table 3. It makes the entire SWSC process a terrible circumstance.

Table 2. QoS aware selection of web services using existing centralized SLA file at time t1

Service	RT (ms) 0.25	Avail (100%) 0.25	Relia. (100%) 0.15	Cost (rs) 0.1	Reput (100%) 0.25	Rank
S10	107.00	95	88	8	4	1
S29	110.50	95	82	12	4.2	2
S32	91.81	92	78	10	3.5	3
S43	130.80	82	79	13	4.0	4
S56	138.50	79	68	17	3.8	7
S34	147.00	97	77	15	3.7	8
S76	154.00	72	76	10	4.1	6
S81	153.12	68	72	15	2.9	10
S96	111.10	78	58	12	2.5	9
S113	130.00	86	36	10	3.8	5

Table 3. QoS aware selection of web services using existing centralized SLA file at time t2

Service	RT (ms) 0.25	Avail (100%) 0.25	Relia. (100%) 0.15	Cost (rs) 0.1	Reput (100%) 0.25	Rank
S10	107.00	95	88	8	4	1
S102	**97.00**	**98**	**92**	**8**	**3.2**	**2**
S29	110.50	95	88	8	4.2	3
S32	91.81	92	78	10	3.5	4
S14	**98.80**	**82**	**79**	**12**	**3.2**	**5**
S82	**120.34**	**84**	**81**	**12**	**3.1**	**7**
S56	138.50	79	68	17	3.8	9
S76	154.00	72	86	10	4.1	8
S96	111.10	78	58	12	2.5	10
S113	130.00	86	36	10	3.8	6

The same selection process is repeated in the proposed smart contract-based SWSC process at different times t3 and t4 which is shown in Table 4 and Table 5. When a request is given for the SWSC process in a smart contract, it verifies and validates the QoS values via a consensus mechanism. So it does not allow anybody to make changes in the Blockchain-based service instance file. Thus, it correctly identifies the unpleasant services and rejects these hostile services. Thus, there are no changes in the generated service selection rank. The reason is the service provider or centralized agent wouldn't able to mutate the automatically generated and published values. Also, the smart contract manages the dynamically offered QoS values like availability, accessibility, reliability, response time, and user reputation in the background database, and makes them be updated. Thus the best services are selected and ranked appropriately. Nobody would steal the participant's peer's information during a web service interaction. Thus, the proposed smart contract-based SWSC process ensures a trusted and secured environment for executing the SWSC process.

Table 4. QoS aware selection of web services using proposed Blockchain-based SLA file executed via smart contract at time t1

Web Services	RT (ms) 0.25	Avail (100%) 0.25	Relia. (100%) 0.15	Cost (rs) 0.1	Reput (100%) 0.25	Rank
S10	107.00	95	88	8	4	1
S29	110.50	95	82	12	4.2	2
S32	91.81	92	78	10	3.5	3
S43	130.80	82	79	13	4.0	4
S56	138.50	79	68	17	3.8	7
S34	147.00	97	77	15	3.7	8
S76	154.00	72	76	10	4.1	6
S81	153.12	68	72	15	2.9	10
S96	111.10	78	58	12	2.5	9
S113	130.00	86	36	10	3.8	5

Table 5. QoS aware selection of web services using proposed Blockchain-based SLA file executed via smart contract at time t2

Service	RT (ms) 0.25	Avail (100%) 0.25	Relia. (100%) 0.15	Cost (rs) 0.1	Reput (100%) 0.25	Rank
S10	107.00	95	88	8	4	1
S29	110.50	95	82	12	4.2	2
S32	91.81	92	78	10	3.5	3
S43	130.80	82	79	13	4.0	4
S56	138.50	79	68	17	3.8	7
S34	147.00	97	77	15	3.7	8
S76	154.00	72	76	10	4.1	6
S81	153.12	68	72	15	2.9	10
S96	111.10	78	58	12	2.5	9
S113	130.00	86	36	10	3.8	5

Algorithm: Fuzzy Topsis Based QoS Selection

```
QoS based web service selection using Fuzzy TOPSIS
1. generate a normalized decision matrix for the given
QoS constraint value.
```

$$NM_{ij=}\sum x_{ij}^{2}\text{, for i=1,…,n and j = 1,..,m.}$$

```
where 'i' defines the QoS values of the discovered services
and 'j' defines QoS constraints given by user.
2. Evaluate the NDM using constraint weightage
given by user as
```

$$\nu_{ij} = \sum W_{ij} NM_{ij} = 1.$$

```
3. Find and evaluate positive ideal QoS attributes and
Negative ideal QoS attributes as,
```

$$A^{+} = \left\{ \left(pv_1^{+}, pv_2^{+}, .., pv_n^{+} \right) \right\} = \left\{ \left(\max pv_{ij} \mid I \epsilon\, I \right), \left(\min pv_{ij} \mid I \epsilon\, 0 \right) \right\}$$

$$A^{-} = \left\{ \left(pv_1^{-}, pv_2^{-}, .., pv_n^{-} \right) \right\} = \left\{ \left(\min pv_{ij} \mid I \epsilon\, 0 \right), \left(\min pv_{ij} \mid I \epsilon\, I \right) \right\}$$

```
4. Calculate the separation measures using ED
```

$$d_j^{+} = \{ \sum \ (v_{ij} + v_i^{+})^2 \}^{1/2}\ and\, d_j^{-} = \{ \sum \ (v_{ij} - v_i^{-})^2 \}^{1/2}$$

```
Where,  j = 1….n, ED - Euclidean distance
5. Evaluate the relativity of positive ideal values and negative
ideal values among retrieved services for the given request,
```

$$r_j = \frac{d_j^{+}}{d_j^{+} + d_j^{-}}, j = 1,..,n \quad , \quad r_j = \frac{d_j^{-}}{d_j^{+} + d_j^{-}}, j = 1,..,n$$

```
6.  In order to pick the user given QoS-constraint by fixing
weightages by
```

$$R\,qos\left(ws_i \right) = w_{qos} * r_{qos} = 1$$

```
7. Generates the ranked list in ascending order
8. Given top-ranked list service for composition
End
```

6. EXPERIMENTAL EVALUATION

The experimental dataset is taken from OWLS_TC data set from the URL - http://projects.semweb central.org /projects /owls-tc/, which consist of 1083 OWL-S service files from the following nine domains: education-286, medical care-73, food-34, travel-197, communication-59, economy-395, weapon-40, geography-60, simulation-16 and the semantic relationship between concepts can be got through the reasoning tool of Mindswap OWL-S API (http://www.mindswap.org/2004/owl-s/api/). During the experiment, 73 services from Medical care, 28 services from education domains, and 21 services from the communication domain, a total of 122 services are selected from OWL-TC for experimental analysis and comparison. The QoS values are created and experimental on our own.

A. Experiment Settings

During the experiment, the SLA agreement is created manually with certain conditions like a penalty, constraints values are set, along with OWL-S functional file which is validated and stored in Blockchain, then broadcasted the metadata in the proposed service Blockchain registry. The Experiment is evaluated for both OWL-S SLA (Python Skit) and Blockchain-based OWL-S (Truffle framework). The web services files are accessed from the working system (Intel CORE i5 8th Gen). The truffle framework access the service files from the working system via metamask. The smart contract executes the PoS miner to validate the credentials of 122 stimulated web service provider files and enable them to publish in the blockchain. The Ganache shows the status of the running framework namely: accounts address, mining details, and hash values. During service composition, the OWL-X-Planner and Smart contract act as baseline methods to evaluate the performance of service discovery.

B. Confusion Matrix

In general, a confusion matrix is a table used to predict the performance of the specific model. Here, the confusion matrix is used to predict the stimulation changes that existed over the services implemented over the existing SWSC environment and Ethereum Blockchain environment. Here, the task is to predict the correct service provider based on its functional and non-functional values for which true values are already known. The confusion matrix predicts the performance level of the model based on four factors: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The values over the four tables are inscribed based on two values Yes or No. Table 6 shows the way that the confusion matrix predicts the model.

TP – If the services are retrieved correctly based on the published IOPE and QoS constraints in the registry without an attempt to any modification, then predicted as TP.
FP – The services are selected based on changes made by centralized executing algorithms.
FN – The service has correct IOPE and QoS values, but has not been selected by making modifications in values by centralized execution algorithms.
TN – Number of services that have not been tampered with and have not been selected in the service list.

Table 6. Confusion Matrix

		Predicted	
		Positive	**Negative**
Actual Result	**Positive**	**a** True Positive (tp) Correct Inference	b False Negative (fn) (Error of the Second kind – a miss)
	Negative	c False Positive (FP) false alarms (Error of the First kind – false hit)	**d** True Negative (TN) Correct Rejection

C. Performance Measure

The performance of the existing stimulated model and proposed Blockchain-based SWSC model is evaluated over the confusion matrix and results are retrieved in terms of precision, recall, F-measure, False-Positive Rate (FPR), and Accuracy.

Precision, recall, f-measure, accuracy are the significant measures for selecting the best models. Precision is the ratio that defines the number of correct services that have been retrieved based on IOPE and QoS selection algorithms among incorrect services. Recall defines the ratio of retrieving correct services among a total number of correct services. F-measure evaluates the harmonic mean of precision and recall terms. FPR defines the ratio of selecting the incorrect services as correct web services. Accuracy defines the model that correctly predicts the correct services and incorrect correct for the given total web services from the service repository.

$$precision : \frac{TP}{predicted\ yes}$$

$$Recall : \frac{TP}{actual\ yes}$$

$$F - Measure\ : 2x \frac{precision\ x\ recall}{precision + recall}$$

$$False\ Positive\ Rate\left(FPR\right) : \frac{FP}{actual\ no}$$

$$Accuracy : \frac{TP + TN}{total}$$

D. Experimental Results and Discussion

This section compares the efficiency of the web services composition process in both the Python SKlearn and the smart contract environment using confusion matrices. The selection efficiency of the proposed smart contract-based QoS-selection is compared with the Python Sklearn based fuzzy Topsis model in terms of Precision, Recall, and Accuracy. The confusion matrices represent the enactment of the execution platform over-stimulated set of services for which true values are already known. Table 7 and Table 8 illustrate the confusion matrices of the SLA based-QoS aware service selection (Python SKlearn) and Table 9 and Table 10 illustrate the confusion matrices of the smart contract-based QoS aware service selection in the ethereum smart contract environment. The performances are evaluated for the generated confusion matrices which are represented in Table 11, Table 12, Table 13, and Table 14. Based on the performance evaluation, the graph is generated to pictorially illustrate the performance of the existing centralized SLA based SWSC execution environment and proposed Blockchain smart contract-based SWSC execution environment in terms of Precision, Recall, and overall performance measure graph which is illustrated in Figure 9, Figure 10, and Figure 11. Correspondingly it has been seen that smart contract-based QoS-aware selection at an average of 14.9% improvement in precision, 6.23% improvement over recall rate, 13.4 % improvement of accuracy, and 10.02% improvement in F-Measure. However, the major decrement of 73.06 % is seen in FP (False Positive) Rate. This ensures a decline in the selection of unpleasant services for the SWSC process. The higher the precision, recall, F-Measure, accuracy rate, and lower the false positive rate shows that our proposed ethereum platform provides a secure, trusted environment for web service interactions.

Confusion Matrix for Selected Service

The confusion matrix is generated for SLA based-QoS aware service selection at time t1 and time t2 is shown in Table 7 and Table 8 and smart contract-based QoS aware service selection at time t1 and time t2 is shown in Table 9 and Table 10. At time t1, the SLA-based QoS selection retrieves 67 correct services from the dataset. But at time t2, when a service provider improves their services, then unpleasant services are also selected, hence only 60 correct services are identified. The rate of true negative (TN) values is also increased in SLA-based selection time t2. Considering the smart contract-based selection, 68 correct services are identified based on agreement. Only small fluctuations are identified for selecting correct services.

Table 7. QoS aware selection of web services from the given dataset using existing centralized SLA file at time t1

	Predicted Services a (YES)	**Predicted Services b (NO)**	
Actual YES	TP = 67	FN = 2	69
Actual NO	FP = 3	TN = 50	53
	70	52	

Table 8. QoS aware selection of web services from the given dataset using existing centralized SLA file at time t2

	Predicted Services a (YES)	**Predicted Services b (NO)**	
Actual YES	TP = 60	FN = 5	65
Actual NO	FP = 12	TN = 45	57
	72	50	

Table 9. QoS aware selection of web services from the given dataset using proposed Blockchain-based SLA file at time t1

	Predicted Services a (YES)	**Predicted Services b (NO)**	
Actual YES	TP = 68	FN = 2	70
Actual NO	FP = 1	TN = 51	52
	69	53	

Table 10. QoS aware selection of web services from the given dataset using proposed Blockchain-based SLA file at time t1

	Predicted Services a (YES)	**Predicted Services b (NO)**	
Actual YES	TP = 68	FN = 1	69
Actual NO	FP = 3	TN = 50	53
	70	52	

Performance Measure

The performance is evaluated based on the generated confusion matrix. The performance measure of SLA based-QoS aware service selection at time t1 and time t2 is shown in Table 11 and Table 12 and smart contract-based QoS aware service selection at time t1 and time t2 is shown in Table 13 and Table 14. As per the confusion matrix values, it has been seen that the performance widely varies at time t1 and time t2 in SLA-based QoS selection. Considering the smart contract-based selection, the performance is always higher as compared with the SLA-based selection process.

Table 11. Performance calculation of confusion matrix for existing SLA based selection at time t1

Performance Measures	
Precision	0.9571
Recall	0.971
F-Measure	0.9639
FP Rate	0.0567
Accuracy	0.9590

Table 12. Performance calculation of confusion matrix for existing SLA based selection at time t2

Performance Measures	
Precision	0.833
Recall	0.923
F-Measure	0.8756
FP Rate	0.2105
Accuracy	0.860

Table 13. Performance measures of a confusion matrix for proposed Blockchain-based SLA selection at time t1

Performance Measures	
Precision	0.985
Recall	0.97
F-Measure	0.977
FP Rate	0.019
Accuracy	0.977

Table 14. Performance measures of a confusion matrix for proposed Blockchain-based SLA selection at time t2

Performance Measure	
Precision	0.985
Recall	0.97
F-Measure	0.977
FP Rate	0.056
Accuracy	0.967

Graph Critical Analysis

The graph is generated to pictorially represent the performance of SLA based-QoS aware service selection at time t1 and time t2 and smart contract-based QoS aware service selection at time t1 and time t2 which are illustrated in terms of precision in Figure 9, False Positive Rate in Figure 10, and Overall Performance measure (Precision, Recall, FP Rate, F-measure) in Figure 11. The precision graph, FP Rate graph is emphasized individually to illustrate the effectiveness of the prediction accuracy of the services in both SLA-based and smart contract-based selection. The precision represents how much the approach predicts the correct services as comparing with SLA based approach. Whereas, the FP Rate represents how much the approaches predict incorrect services as true services. Then, the overall performance graph is illustrated to exemplify overall performance.

The precision of the smart contract model has a sharp improvement of nearly 18.25 % at both times t1 and t2=1 as compared with SLA-based selection at time t2. The recall of the smart contract model has a sharp improvement of nearly 5.09 % at time t1 and 5% improvement at time t2 as compared with SLA-based selection at time t2. The F-measure values are also 11.58% improvement with SLA-T2 selection. The average accuracy of the smart contract model under the same category is higher than the other SLA -based selection. But, the FP Rate is of the smart contract model has a sharp decrement of nearly 85% at time t1 selection and 73.4% at time t2 selection. The reason for this effective selection of the smart contract-based model is by avoiding malevolent services. The smart contract-based service agreement and smart contract-based SWSC process execution are jointly computed together to achieves a trusted high-performance system. The comparisons show that our proposed smart contract-based approach considerably increases the selection accuracy in terms of recall, precision, F-measure, and accuracy, which proves that the proposed approach outperforms the SLA-based OWL-S baseline method in all cases.

Figure 9. Precision as per SLA and smart contract

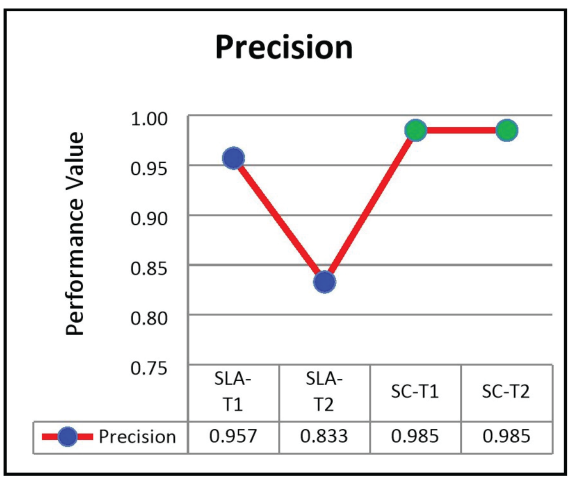


Figure 10. False positive rate as per SLA and smart contract

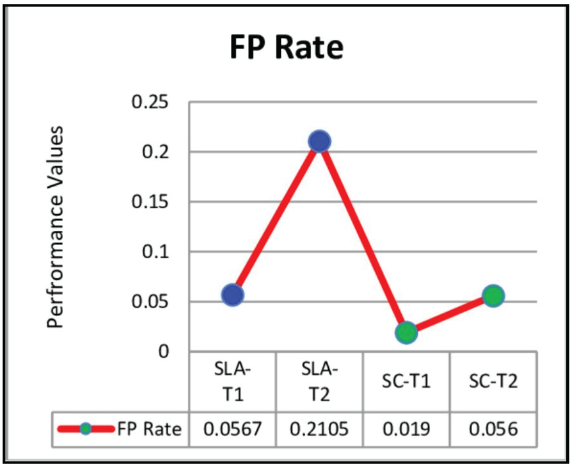
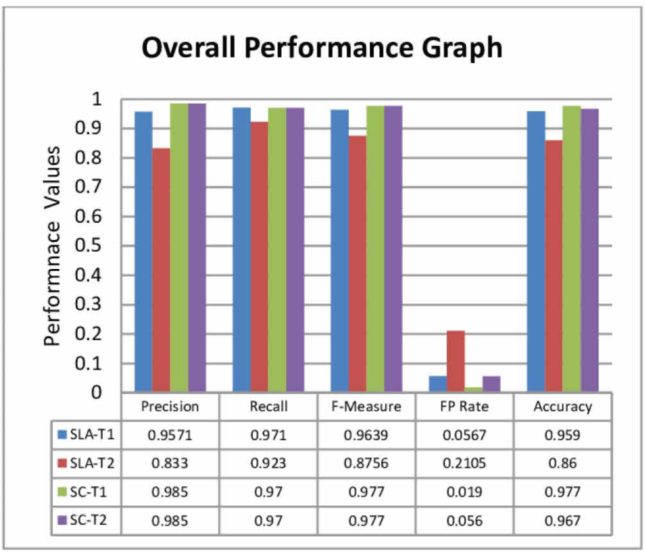


Figure 11. Overall performance measure between SLA T1, SLA T2, SC T1, and SC T2



7. CONCLUSION

Blockchain is a distributed ledger in which all the transactions are executed and homomorphic hashes are generated and stored in the public distributed ledger. Due to its properties, in this work, an attempt of the SWSC process is built in an Ethereum Blockchain to find the trustworthiness and credibility of the SWSC process. As the existing web service composition environment having untrustworthiness among participating parties, there are necessities to build a trusted environment for healthy interactions among service providers and consumers.

This conceptual web service composition architecture is set up using Blockchain and the existing centralized-based architecture is fully automated. In Blockchain, from service provider verification to broadcasting of service in the blockchain repository, and retrieving these services for the given customer request is enabled by Blockchain alone and registered every single transaction via homomorphic hash value. Even the centralized WSC architecture also enables the automation process but the hash value is generated by the centralized parties. It clearly brings the i) trust among participating WSC peers, ii) provide transparency while broadcasting of service and selecting of service.

An extensive experiment is conducted in the ethereum conceptual WSC model, whereas the functional data is collected from the OWL-TC dataset and QoS values are manually generated. The automation of the WSC process is stimulated and values have been tampered with in order to check the effectiveness of centralized SLA-based conceptual model and smart contract-based conceptual model. As the blockchain generated a homomorphic hash and is stored in the blockchain, no one can change the value, but the centralized agent can change the value in the existing architecture. Only the QoS values are changed in order to verify the effectiveness of the proposed model and performance is calculated in terms of precision, recall, F-measure, FPR, and accuracy.

Blockchain-based WSC model achieves higher performance accuracy as compared with the existing WSC process. The immutable property ensures no SLA agreement data, IOPE and QoS values have tampered which is resides in the registry. As a distributed network it supports the reliability of service. Traceability ensures the history and origin of transactions are well-preserved and anyone can view the transactional information based on their authorization levels. Trustworthiness and credibility of the WSC process are achieved by smart contract-based automation and homomorphic hash values which improve the trust over B2B interactions and business intelligence.

REFERENCES

AlSedrani, A., & Touir, A. (2016). Web service composition processes: a comparative study. International Journal on Web Service Computing, 7(1).

Amirthasaravanan, P. R., & Sudhesh, R. (2016). Analysis on Runtime Associated QOS-Based Proficient Web Services Discovery Optimization. *An International Journal of Engineering, Technology, and Techniques. Brazilian Archives of Biology and Technology*. Advance online publication. doi:10.1590/1678-4324-2016161012

Baqa, H., Nguyen, B., Truong, N., Crespi, G., Lee, M., & Le Gall, F. (2019). *Semantic Smart Contracts for Blockchain-based Services in the Internet of Things.* http://www.eglobalmark.com/wp-content/uploads/2019/11/NCA_2019_Camera- Ready _29 _ deposited.pdf

Bashir, I. (2018). *Mastering Blockchain*. Packt Press.

Bianco, P., Grace, A., & Merson, L. (2008). *Service Level Agreements in Service-Oriented Architecture Environments.* Technical Note at Software Architecture Technology Initiative Integration of Software-Intensive Systems Initiative.

Bruyn, S. (2017). *Blockchain an introduction*. Retrieved from https://beta.vu.nl/nl/Images/werkstuk-bruyn_tcm235-862258.pdf

Cai, W., Du, X., & Xu, J. (2019). *A Personalized QoS Prediction Method for Web Services via Blockchain-Based Matrix Factorization. MDPI.*

Castellanos, A. & Benbunan-Fich, R. (2018). Digitalization of Land Records: From Paper to Blockchain. *Thirty Ninth International Conference on Information Systems*.

Chen, G., Xu, B., Lu, M., & Chen, N.-S. (2018). *Exploring blockchain technology and its potential applications for education*. Smart Learning Environments. doi:10.1186/s40561-017-0050-x

de Lange, P., Janson, T., & Klamma, R. (2019). Decentralized Service Registry and Discovery in P2P Networks Using Blockchain Technology. *Web Engineering*, 296-311. DOI: 10.1007/978-3-030-19274-7_22

Desai, S., & Shelke, R. et al., (2020). Blockchain Based Secure Data Storage and Access Control System Using IPFS. *Journal of critical reviews*, 7(19).

Fanning, K., & Centers, D. P. (2016). Blockchain and its coming impact on financial services. *Journal of Corporate Accounting & Finance*, *27*(5), 53–57. doi:10.1002/jcaf.22179

Heria, R. (2018). Legal Recognition of Blockchain Registries and Smart Contracts. In *Proc.Blockchains & smart contracts legal and regulatory framework workshop*. Retrieved on July 23, 2019.

Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). The Blockchain Model of Cryptography and Privacy-Preserving Smart Contract. *IEEE Symposium on Security and Privacy (SP)*, 839–858. doi:10.1109/SP.2016.55

Kshetri, N., & Voas, J. (2018). Blockchain-Enabled E-Voting. *IEEE Software.* www.computer.org/software

Lee, J. & Ben-Natan, R. (2008). *Integrating Service Level Agreements: Optimizing Your OSS for SLA Delivery.* https://www.amazon.com/Integrating-Service-Level-Agreements-Optimizing/dp/0471210129

Maheswari, S., & Karpagam, G., R., (2015). Comparative Analysis of Semantic Web Service Selection Methods. *Indian Journal of Science and Technology*, *8*(S3), 159–169.

Maheswari, U., & Karpagam, G. (2014). I/O Match (I/O MAT) and Behavioural Match (BEH MAT) Based Semantic Web Service Discovery. *Journal of Theoretical and Applied Information Technology*.

Nomura Research Institute, (2016). *Survey on blockchain technologies and related services*. Retrieved from https://www.meti.go.jp/english/press/2016/pdf/0531_01f.pdf

Oskooei, M., & Daud, S. (2014). Quality of Service model for semantic Web service Selection. *IEEE 2014 International Conference on Computer, Communication, and Control Technology (I4CT 2014)*, 266-270.

Ramadan, N. (2015). A Proposed Approach for Monitoring Quality of Web Services Using Service Level Agreement. *International Journal of Computer Science and Information Security*, *13*(1), 29–36.

Roy, S., Ashaduzzaman, M., Hassan, M., & Chowdhury, A. R. (2019). BlockChain for IoT Security and Management: Current Prospects, Challenges and Future Directions. *5th International Conference on Networking, Systems and Security (NSysS- IEEE Xplore).*

Salah, K., Nizamuddin, N., Jayaraman, R., & Omar, M. (2019). Blockchain-Based Soybean Traceability in Agricultural Supply Chain. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 73295–73305. Advance online publication. doi:10.1109/ACCESS.2019.2918000

Schwab, K. (2017). *The Fourth Industrial Revolution*. The Crown Publishing Group.

Shanti Bruyn, A. (2017). *Blockchain an introduction.* Research paper from https://beta.vu.nl/nl/Images/werkstuk-bruyn_tcm235-862258.pdf

Stephen, J., Yang, H., Zhang, J., Blue, C., & Lan, W. (2007). Service-level agreement-based QoS analysis for web services discovery and composition. *International Journal of Internet and Enterprise Management*.

W3C RIF-WG. (2010). *Rule-based Service Level Agreements (SLA) and Web Services.* https://www.w3.org/2005/rules/wg/wiki/Rule-based_Service_Level_ Agree ments_ (SLA) _and_ Web_Services.html

Wang, W., Hoang, D., Hu, T., Xiong, P., Niyato, Z., Wang, P., Wen Y, & Kim, I. (2019). A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access*.

Yaswanth, C. (2016). A Hybrid QoE-QoS Web Service Selection for Efficient Product Recommendation Using Service Composition. Department of Computer Science. *Middle East Journal of Scientific Research*, *24*(3), 712–721.

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. *IEEE International Congress on Big Data,* 557–564.

Zhou, H., de Laat, C., & Zhao, Z. (2018). Trustworthy Cloud Service Level Agreement Enforcement with Blockchain Based Smart Contract. *IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. doi:10.1109/CloudCom2018.2018.00057

Zisman, A. (2011). Supporting Service Level Agreement with Service Discovery. *Engineering Reliable Service Oriented Architecture: Managing Complexity and Service Level Agreements*.

S. Sridevi is a Ph.D. candidate in the Department of Computer Science at the PSG College of Technology, Coimbatore. Her research interest includes Semantic Web and Incorporation of Blockchain platform in semantic web to improve the accuracy of service selection. She received B.E in computer science from Anna University, Tamilnadu, India in 2014 and ME in computer science from Anna University, Chennai in 2016.

G.R. Karpagam is a Professor and associate department head of computer science at PSG College of Technology. She directs the investigation of the blockchain platform for semantic web composition method. Her research interest lies in web service, semantic web, cloud computing, sentimental analysis, Data Mining. She has published extensively in these areas in refereed journals and conferences. She has been awarded more research grants from agencies such as the Department of Science & Technology, and DBT. She is a senior member of IEEE.

B. Vinoth Kumar (PhD) is working as an Associate Professor with 16 years of experience in the Department of Information Technology at PSG College of Technology. His research interests include Soft Computing, Blockchain and Digital Image Processing. He is author of more than 26 papers in refereed journals and international conferences. He has edited three books with reputed publishers such as Springer and CRC Press. He serves as a Guest Editor/Reviewer of many journals with leading publishers such as Inderscience, De Gruyter and Springer.

J. Uma Maheswari (PhD) is an Assistant Professor (SG) with 14 years of experience. She completed her BE Computer Science and Engineering in Sri Krishna College of Engineering and Technology, Coimbatore from Bharathiar University in 2003 and ME Computer Science and Engineering in JJ College of Engineering and Technology, Trichy from Anna University in 2006. Her areas of research include Cloud Computing and Semantic web services. She has published journal papers extensively in these areas in refereed journals and conferences.