Bazaar-Blockchain: A Blockchain for Bazaar-based Cloud Markets

Benedikt Pittl
University of Vienna
Faculty of Computer Science, Austria
benedikt.pittl@univie.ac.at

Werner Mach
University of Vienna
Faculty of Computer Science, Austria
werner.mach@univie.ac.at

Erich Schikuta
University of Vienna
Faculty of Computer Science, Austria
erich.schikuta@univie.ac.at

***Abstract*—Today, Cloud services are mainly traded on provider platforms such as on Amazon's EC2 On-Demand marketspace. Thereby, consumers and providers neither negotiate the price nor the characteristics of the services. The recent years underpin a trend to more dynamic Cloud markets. So e.g. the Cloud provider Virtustream released a revenue model where consumers are charged based on consumed μVMs while Amazon extended it's spot market with spot blocks and spot fleet management. Hence, multi-round bilateral negotiations are a promising approach for trading Cloud services on future Cloud markets. Such negotiations are based on an alternating exchange of offers and hence, they are termed Bazaar-negotiations. Specifications such as the WS-Agreement Negotiation - which is maintained by the Open Grid Forum - foster the development of Bazaar-based markets. To ensure integrity and transparency of negotiations with untrusted negotiation partners - which is a precondition for the adaption of Bazaar-based Cloud markets - blockchains are a promising approach. In this paper we introduce a concept of a blockchain for Bazaar-negotiations whereby we assume that the offers exchanged during negotiations follow the structure defined in the WS-Agreement Negotiation specification. We implemented it within a CloudSim based simulation environment which is able to simulate such Bazaar-based markets to show its technical feasibility.**

***Keywords*-SLA Negotiation; Cloud Market; Cloud SLA; Blockchain;**

I. Introduction

Today, Cloud markets are emerging where Cloud providers strive to develop and employ innovative vendor strategies to gain market shares [1]. The leading platform for purchasing infrastructure as a service (IaaS) in the form of virtual machines (VMs) is Amazon Web Services (AWS), see e.g. [2]. The traded virtual machines are preconfigured - called instance types on Amazon - and are sold on different EC2 marketspaces. Amazon distinguishes between (i) a reservation marketspace, (ii) an on-demand marketspace and, (iii) a spot marketspace. On the reservation marketspace, consumers have a long-term contract (e.g. three years) with Amazon. Such a long-term contract does not exist on the on-demand marketspace where virtual machines are charged per hour. On the spot marketspace consumers define the maximal price, which they want to pay per hour for a certain virtual machine. If the defined maximum price is higher than the so called *spot market price* then the consumer can use the virtual machine. If the spot market price exceeds the maximum price then the virtual machine cannot be used. The spot market price is a dynamic price reflecting Amazon's current demand and supply. There is an extension of the spot marketspace on Amazon - called spot block - where the consumer sets a maximum price for a virtual machine which runs 1 to 6 hours - it is not interrupted in cases in which the spot market price exceeds the maximum price of the consumer. In addition to the three marketspaces, Amazon hosts a platform where consumers are able to sell virtual machines with a long term contract - purchased on the reservation marketspace - to other consumers. Until now consumers usually purchase Cloud services directly from providers - neutral market platforms for trading Cloud resources such as envisioned in [3] failed in reality: The *Deutsche Boerse Cloud Exchange* was intended as such a platform which started in 2015 and closed - after a couple of months - in 2016. Using this platform, consumers were able to compare the prices of VMs from different providers and to purchase them. In the scientific community reasons for the failure of the platform - which was inter-alia founded by the German Stock Exchange - have not been discussed yet. In industry-related literature such as in [4] the low level of maturity of this platform was mentioned. The scientific community introduced different visions of future Cloud markets, ranging from centralized auctions [5] over decentralized auctions [6] to bilateral multi-round negotiations (Bazaar-negotiations) [7], [8]. The paper at hand focuses on autonomous Bazaar-negotiations where consumers and providers exchange offers until all offers are rejected or an agreement is formed such as shown in figure 1. During the negotiations consumers and providers use SLA negotiation strategies. Such strategies are e.g. described in [7] or [10].

A main research challenge is to develop and establish mechanisms to ensure transparency and integrity during negotiations with untrusted participants on Bazaar-based Cloud markets. The offers exchanged during negotiations could be e.g. subject of manipulations or participants could deny the receipt of certain offers. To overcome this issue trusted third parties could act as intermediaries which confirm the exchange of offers as well as the content of them. Robert Sams summarizes the power of trusted third parties by introducing *three sins* - representing three possible abuses of trusted third parties: *sin of commission*, *sin of deletion* and *sin of omission* [11]. Hence, in the last years the scientific community reverted to the blockchain technology for substituting such trusted third parties [12]. It is characterized by the distributed storage of data and the usage of cryptographical technologies which makes it tamper-safe. Unlike trusted third parties, no single participant of the blockchain network has the power to commit one of the previous mentioned sins - even if there are some weaknesses such as described in [13]. While the blockchain got popular with the Bitcoin hype it is further considered to be used e.g. for tracing modifications of knowledge stored in the form of conceptual models [14] or for business process monitoring and execution [15]. This paper is result of our research project which focuses (i) to develop and implement a blockchain for Bazaar-negotiations on Cloud markets and, (ii) to map the resulting agreements to Smart Contracts[1] which observe the execution of the agreements.

The paper at hand focuses on the first aspect. We developed a concept of a blockchain by considering the characteristics of Bazaar-negotiations. The approach was implemented in a simulation environment to test its technical feasibility. We see this work in line with the manifesto of future generation Cloud computing introduced by Buyya et.al. where the assistance of Cloud computing using blockchain technology is mentioned as a future field of research [17].

The remainder of the paper is structured as follows: In section II foundations of the blockchain as well as related work is presented. The concept of the envisioned Bazaar-blockchain is introduced in section III. The initial implementation is presented in section IV followed by the discussion in section V. The paper closes with the conclusion in section VI.

II. Background and Related Work

The related work section is structured along two parts. The first part summarizes foundations of the blockchain. In the second part related blockchain applications and approaches are presented.

With the introduction of Bitcoin the popularity of the underlying blockchain technology increased dramatically [18]. Narayanan et. al. describe the blockchain as linked list which use hash pointers as links [19]. Such hash pointers are generated using hash functions. A hash function maps any data to a string of a fixed length. This string is called hash value - $H : K \to S$, whereby the length of each $s \in S$ is fixed and $k \in K$ is any data which is hashed. For a given hash value $H(k)$ it is almost impossible to calculate the hashed data $k \in K$[2]. Each block of the blockchain uses the hash value of the previous block as hash pointer - an exception is the first block which is called *genisis* [20].

[1]We see Smart Contracts as [16]: software which is able to enacting legal contracts autonomously

[2]Other characteristics of hash functions are e.g. described in [19]

$$\begin{tikzcd}
 & \mathbf{Offer}_b \arrow[r] & \mathbf{Offer}_a \arrow[r] & \mathbf{Rejected}_b \\
\mathbf{Offer}_a \arrow[ur] \arrow[dr] & & \mathbf{Offer}_a \arrow[r] & \mathbf{Rejected}_b \\
 & \mathbf{Offer}_b \arrow[r] \arrow[ur] & \mathbf{Offer}_a \arrow[r] & \mathbf{Agreement}_b
\end{tikzcd}$$

Figure 1: Exchange of offers between the market participants a and b

This contributes to the stability of blockchains: if one of the previous blocks is changed its hash value changes and so it is unequal to the hash pointer used by the following block - it is obvious that the previous block has been modified. As each block contains a hash pointer to the previous block and is hashed again to be used as hash pointer in the following block a *tamper-evident log* occurs [19]. Hash values are not only used for hash pointers but also for the data stored in the blocks of a blockchain. Blockchain implementations such as Bitcoin make therefore use of so called Merkle trees [20]. They are binary trees whereby the leaves represent data which should be stored. All other nodes represent hash values created out of its child nodes. For example a node which connects the two nodes n_1 and n_2 generates a hash value of data stored in these nodes: $H(n_1||n_2)$ where $||$ represents the concatenation operator. The root of the Merkle tree is a hash value which represents all the data of the leaves. So if any data in the leaves changes, the hash of the root changes too. As the root of the Merkle tree is part of the block it is used for creating hash pointers. Hence, any modification of the data makes existing hash pointers to it's block invalid. The benefits of the pairwise hashing instead of creating a single hash out of all the data is that it can be e.g. efficiently proofed which data changed - for more information see [19]. The enabling technique of the blockchain is the public-private key technology - a precondition for creating signatures and hence to ensure integrity of data [21]. So blockchain users can sign data - such as transactions which represent the transfer of money in Bitcoin - with their private key. The generally available public key can be used by another blockchain user to validate the signature. These cryptographical approach ensures integrity without the need of encrypting the data. Hence, the stored data is readable and transparent but at the same time the integrity can be validated by everybody who has access to the public key of the signer. For example the data which should stored to the blockchain has to be singed by their creators. Miners which are responsible to add new blocks to the blockchain have to inter-alia validate the signature of the data which should be added to the block. After a miner created a block it publishes it to other blockchain nodes which validate the block and add it to their local blockchains if no inconsistency was detected.

In the scientific community different applications for the blockchain can be found. For example the authors of [22] identified challenges and opportunities of the blockchain in the filed of business process management. In [14] the authors introduced a so called knowledge blockchain which uses blockchain technology to keep track on knowledge expressed in conceptual models. In [15] the authors developed a blockchain approach for cross-enterprise processes where processes are mapped to Smart Contracts which are executed on the blockchain. Applications of the blockchain for the health community have been introduced, e.g. in [23]. In [24] the authors pursue the vision of using the transparent blockchain technology for SLAs established between Cloud providers and consumers. The paper presents a feasibility and a comparative study of blockchain in the Cloud domain - with a focus on identifying security and trust requirements the paper lacked an implementation of the approach. In [25] and [26] the authors presented a smart-contracting setup phase. A formalization of the setup-lifecycle for electronic community establishment was presented using graphical models with the underlying formal semantics. In [17] the authors mention that the blockchain can assist Cloud computing without mentioning a concrete approach. In [27] the authors envisioned to use a blockchain for power systems. For the management of Cloud services several approaches have been introduced such as the generic framework called *MonValley* [9] where blockhains were not considered. An overview of current existing blockchain applications is given in [28].

Our related work analysis shows that different domain-specific blockchains were introduced but most of them are described on a conceptual level without a concrete implementation. However, no approach for a blockchain based on bilateral SLA negotiations exists.

III. Bazaar-Blockchain

The WS-Agreement Negotiation standard [29] is maintained by the Open Grid Forum and aims on specifying Bazaar-negotiations. It is an extension to the WS-Agreement standard [30] and describes a XML based structure of offers as well as their possible states. In total the WS-Agreement Negotiation standard defines four states of offers. These four states and their transitions are illustrated in figure 2.

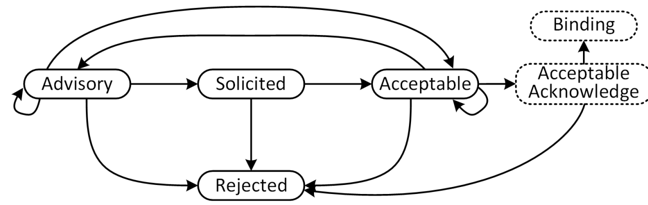

Figure 2: States of offers defined in the WS-Agreement Negotiation specification including two additional states

An offer in the advisory state requires further negotiation as it is e.g. not completely specified. The solicited state is used for offers which are completely specified. The receiver of such an offer is forced either to accept the offer so that the state of the offer becomes acceptable or reject it which leads to the state rejected. Acceptable offers might result into agreements. Agreements are offers to which consumers and providers agree. Offers in the acceptable state of the WS-Agreement Negotiation standard are not binding: *The ACCEPTABLE state indicates that a negotiation participant is willing to accept a negotiation offer as is.* But it is also described that *there is no guarantee that a subsequent agreement is created*. Hence, we introduced two further states in [31]: acceptable acknowledge and binding. The example depicted in figure 1 is a simplification - the agreement represents an offer in the binding state. The rejected state is used for offers which are rejected. Offers sent from participant a to b at time t can be formally described as a set of tuples $o_{a \rightarrow b}(t)$=$\{< s, p, m >\}$ where p represents the price for the service $s =< l_1, ..., l_i >$ and m represents the type of the offer defined in the WS-Agreement Negotiation standard. So m $\in$ $\{$acceptable, rejected, solicited, advisory, binding, acceptable acknowledge$\}$. s is a set of service characteristics. For example, in case of virtual machines $l_1...l_i$ represent inter-alia storage, RAM and processing power. A market participant can send several offers to another market participant at the same time. Hence, $o_{a \rightarrow b}(t)$ is a set.

To overcome the before-mentioned weaknesses of trusted third parties we reverted to the blockchain technology: Our aim is to use blockchain technology for such Bazaar-negotiations on Cloud markets to achieve the following main goals:

- **Documentation of the negotiation process.** A documentation of the whole negotiation process ensures transparency and helps to foster re-negotiation[3] processes such as envisioned in [29].
- **Integrity of offers.** The blockchain is a tamper-safe, highly reliable technology which allows to store the exchanged offers. Offers can not be changed after they are stored in the blockchain.
- **Ensuring conformance to negotiation conditions.** The usage of the blockchain can guarantee the conformance to negotiation conditions - they can be validated before an offer is added to the blockchain. For example, before an offer is added to the blockchain it can be checked if the negotiation to which the offer belongs is running. Hence, not only the agreements - all offers exchanged during negotiation are stored in the blockchain which helps to ensure that the negotiation process adhered to all conditions.

[3] Re-negotiation is a negotiation process where an existing agreement is re-negotiated

A further benefit of using the blockchain technology for Bazaar-negotiations is - due to the representation of blockchain participants using public keys - the guarantee of anonymity during the negotiation process. The negotiation partners store offers to the blockchain and read from it. So they are not directly communicating with each other. Such a blockchain is termed *Bazaar-blockchain* in this paper. Figure 3 shows an overview of the Bazaar-blockchain. The upper part of this figure shows an excerpt of a negotiation tree resulting from a Bazaar-negotiation between two participants a and b which conforms to the WS-Agreement Negotiation specification. Thereby, the first offer - the root of the negotiation tree - is termed template. The offers exchanged during negotiation are stored in the blockchain. This makes the negotiation tamper-safe - no entity could modify already submitted offers or deny their existence. In the given figure a one-to-one mapping between offers and blocks exists. The Merkle tree of each block stores this offer but only the root of the Merkle tree is - inter-alia - used for creating the hash pointer to this block. This helps to reduce the size of the blockchain as the subnodes of the Merkle tree need not to be stored - this technique is also used for the Bitcoin blockchain [32]. A WS-Agreement Negotiation conform offer consists of different sections. These sections are on the leaves of the Merkle tree denoted as $d_1...d_4$ in figure 3. Block 0 contains the template, block 1 contains an offer in the advisory state whereby the section represented by the black circle was modified from the template. Its modification leads to the modification of all parent nodes which contain hash values. Similarly, the second offer in the advisory state was created by modifying a section of the template which leads to a change of the parent nodes in the Merkle tree.

Before we describe the data of the leaves in more detail we discuss an alternative design of the Bazaar-blockchain: In Bitcoin a block contains hundreds of transactions[4]. The concrete number depends on the number of generated transactions and the time to *mine* a block - additionally a block is not allowed to be bigger than 1MB to avoid bottlenecks - an interesting analysis of the effect of the block size on the performance of the blockchain is given in [33]. The leaves of the Merkle tree contain these transactions [34]. These transactions are independent of each other. On the contrary, e.g. the knowledge blockchain introduced in [14] stores model elements and their attribute values on the leaves of the Merkle tree. Hence, the information of the leaves is related with each other - they contain model elements which belong to the same model. The Bazaar-blockchain depicted in figure 3 follows the same paradigm: the data of the Merkle tree contains data which belong to the same offer. In the same way as the Bitcoin blockchain, a block in the

[4]Approximately 1700 transactions are stored per block - see https://blockchain.info/de/charts/n-transactions-per-block for a live statistic

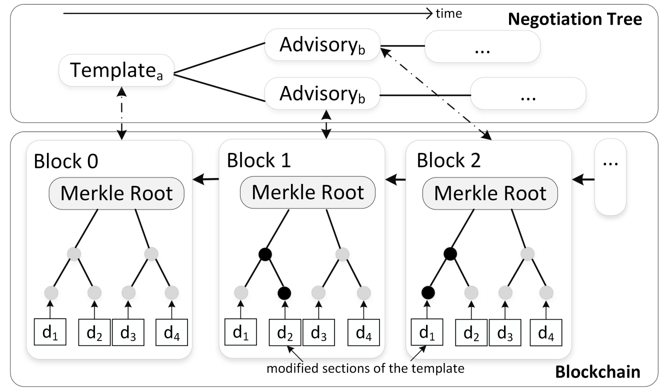

Figure 3: Conceptual negotiation tree and mapping to the Bazaar-blockchain

Bazaar-blockchain could store several offers instead of one. This would imply that the leaves of the Merkle tree would not represent the terms of a single offer. Instead each leaf would represent a whole offer and so the Merkle Root would comprise more information. We think that both approaches are feasible. Due to the complexity of certain SLAs its size could become big and consequently the size of the resulting block so that we prefer to use the variant where one block stores one offer.

A. Merkle Tree

An example of the Merkle tree and its data is illustrated in figure 4. The data are XML sections of the WS-Agreement Negotiation offers. If one of these sections is modified in order to create e.g. a new offer, it is sent to the Bazaar-blockchain which creates - after validating a set of rules - a block with the given data.

The Bazaar-blockchain introduced so far is sufficient to store a single bilateral negotiation. However, each negotiation would require its own blockchain which is hard to realize in reality. Hence we aim on designing the Bazaar-blockchain so that offers from different negotiations are able to be stored in a single blockchain. An offer is sent from a participant to another participant within a negotiation. Similar to transactions in the Bitcoin blockchain we have transactions of negotiation messages in the Bazaar-blockchain. If the transaction is added to the Merkle tree - as a separate leaf - the Bazaar-blockchain can be used for multiple negotiations as each block contains the offer - including a negotiation id - as well as the receiver and sender of the offer. In summary, the transaction contains: (i) The sender - represented by its public key, (ii) the receiver - represented by it public key, (iii) a hash of the offer, (iv) a negotiation id, (v) an offer id, (vi) the offer id to which the offer references, (vii) a timestamp representing the time when the offer was sent, and (viii) a timestamp representing the time when the offer expires.

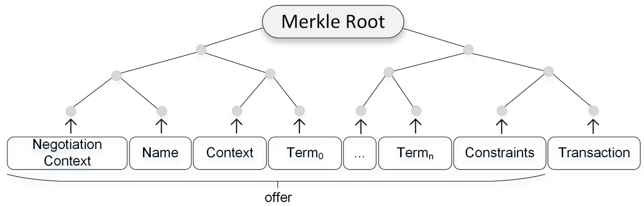

Figure 4: Merkle tree of an offer stored in the Bazaar-blockchain

Some of this data such as negotiation id are optional parts of the WS-Agreement offers which makes them redundant in the transaction. However, due to the transaction data the Bazaar-blockchain can be used even in cases in which the data is missing in the offer. The usage of the public keys for the identification of sender and receiver ensures anonymity during the negotiation process. The hash of the offer can be used for validation purposes. Similarly, the timestamps can be used by the blockchain network for validations - see section III-C.

Each block b of the Bazaar-blockchain B can be represented as a triple: $b =< o, w, ... >, b \in B$ where o is an offer and w is the transaction which is itself a tuple, $w =< sender, receiver, h, id_{negotiation}, id_{offer}, id_{ref\ offer}, t_1, t_2 >$ - the elements of the tuple represent the data described in the previous itemization. So each offer exchanged between two participants is stored to the Bazaar-blockchain so that the following condition holds true: $\forall o \in o_{a \rightarrow b}(t) \rightarrow b \in B, b.o = o \wedge b.w.sender = a \wedge b.w.receiver = b \wedge b.w.h = H(o), ...$. Other elements of a block $b \in B$ are introduced in the following section.

B. Structure of Block

A structure of a block in the Bazaar-blockchain is given in figure 5. The hash of the head of the blockchain is used as hashpointer in the following block. The elements which are not part of the head of the block - the Merkle tree which contains the offer as well as the transaction - are represented by the Merkle root which is part of the header. Hence, a modification of the Merkle tree leads to a modification of the Merkle root. The transaction data as well as the offer - termed *data* in figure 5 - is provided by the sender. It has to provide a signature so that the miners are unable to modify them. Also the knowledge blockchain [14] uses such a signature of the submitting entity which ensures that the identity of the submitter is tied to the block. Miners are responsible for finding a nonce in cases in which a cryptographical puzzle is used. Afterwards they send the block - including a timestamp which represents the time of the creation of the block and a signature of the header using their private key - to other participants which add the block to the blockchain in cases in which they are unable to identify inconsistences.

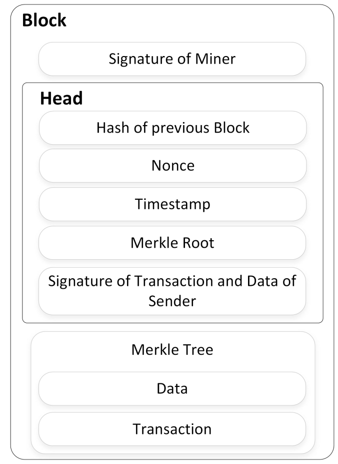

Figure 5: Structure of the block processed in the Bazaar-blockchain

C. Mining rules

Before a block is mined by miners, predefined rules have to be checked. Usually these rules are cryptographical validations such e.g. checks if the signatures are valid. Other rules are used to check the validity on a semantic level. So before a transaction is added to a block of the Bitcoin blockchain it is e.g. checked if the sender has enough bitcoins. For the Bazaar-blockchain the following rules have to be checked by miners:

- **Completeness Check.** Miners have to check if the offer and the transaction which should be added to the blockchain are complete. This means that they contain inter-alia a valid negotiation id as well as a valid receiver.
- **Negotiation Relationship.** If a participant wants to add an offer to the blockchain the miner has to check if the receiver of the offer is in a negotiation relationship with the sender or if the offer - to which the offer refers to - is already expired. So it is not allowed to create an offer in responds to a counteroffer which does not exist in the Bazaar-blockchain. Further it is disallowed to send an offer to a negotiation in which the sender of the offer does not participate.
- **Legal regulations and Inconsistencies.** The miner has further to valid the offers along legal regulations and other inconsistencies. So e.g. it could be checked if the terms of offers have negative values which have to be positive such as response time or price.

In cases in which two negotiation partners form an agreement with an offer in the binding state it is added to the blockchain and converted to a Smart Contract.

IV. Implementation

To test the technical feasibility of the Bazaar-blockchain we created an initial implementation. Figure 6 shows the

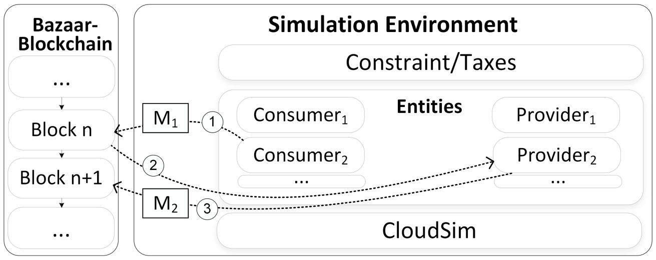

Figure 6: Conceptual overview of the implemented simulation environment and the Bazaar-blockchain

basic structure of it: the right side gives an overview of the simulation environment which we created. It is an extension of CloudSim and called Bazaar-extension [35] as it allows to simulate Bazaar-based Cloud markets where the participants trade virtual machines. During negotiation the market participants exchange offers. In the figure $Consumer_2$ negotiates with $Provider_2$. Therefore, they make use of the Bazaar-blockchain. $M_1...M_n$ are the miners which check the offers along the before mentioned rules. If the rules are validated, the block including the offer is added to the blockchain. The following numbers refer to the numbers in the figure which depicts an excerpt of a negotiation: (1) The consumers starts the negotiation by sending an offer to the provider using its public key. (2) The provider can retrieve the offer from the blockchain. (3) It responds with an offer by adding it to the blockchain from which the consumer can retrieve it.

Before running simulations with our simulation environment you have to define the market participants including their negotiation strategy. After running the simulation you can analyze the resulting resource allocation. Figure 7 shows a screenshot of the simulation environment. On the left side the market participants are listed which attend the simulation scenario (the broker represent consumers). After selecting one of the market participants you see its corresponding negotiations in the second column. In the shown screenshot, broker 2 negotiated with 15 providers. By selecting a negotiation, its exchanged offers - which contain a description of a virtual machine - are visualized as tree list as the right side of the figure shows. Each offer of this tree list is visualized as a dot on the utility-utility plot. The utility values are calculated using utility functions - market participants use them for ranking offers - see [10] for more information. The ordinate of the plot shows the utility of the virtual machines contained in offers for the provider (datacenter) while the abscissa shows the utility of the virtual machine contained in the offer for the consumer.

With the described simulation environment we are able to simulate negotiations Bazaar-based Cloud markets. During negotiations the participants read offers form and write offers to the blockchain. For the implementation of the Bazaar-blockchain we used $SHA1$ with DSA for creating signatures. For the creation of the private-public key pairs we reverted to Sun's Java *KeyPairGenerator*. Each participant gets such a key pair. Afterwards they are able to create signatures and submit offers including signatures. Miners validate them, create the block and publish them to other participants. An excerpt of a JSON serialization of a block from the Bazaar-blockchain is give in listing 1. The structure of the listing reflects the structure of the block depicted in figure 5.

LISTING 1: Excerpt of a JSON serialization of a block in the Bazaar-blockchain


```
{
    ”signatureMiner”: [ 43,... ],
    ”previousBlockHash”: ”93d...”,
    ”nonce”: ”b53...”,
    ”timeStamp”: 1517148618500,
    ”hashMerkleRoot”: ”aae...”,
    ”signatureOfSender”: [ 48,... ],
    ”offer”: {
        ”interactionIdentifier”: {
            ”source”: 2,
            ”negotiationId”: ”1bf0...”
        }
        ,
        ”msgIdentifier”: ”e7...”,
        ”vmCharacteristic”: {
            ”price”: 80.0,
            ”storage”: 886614.0,
            ”processingPower”: 250000.0,
            ”ram”: 6000.0
        }
    }
    ...
}
```


V. DISCUSSION

In this section we discuss research challenges towards the realization of the Bazaar-blockchain.

(i) While the WS-Agreement Negotiation specification [29] envisions that participants negotiate directly with each other, our approach foresees a negotiation process where the participants read and write from the Bazaar-blockchain. Here it has to be considered that the miners require some time before a block containing the submitted offer is added to the Bazaar-blockchain. Additionally, business models have to be developed for the participants of the Bazaar-blockchain. While blockchains for cryptocurrencies such as Bitcoin allow miner to transfer money to themselves, such a business model has to be developed for the participants of the Bazaar-blockchain. For its realization e.g. a fee for each offer stored in the Bazaar-blockchain could be charged.

(ii) While the anonymity of the negotiation partners during negotiation is achievable they need to reveal their identity - at least to a third party - after an agreement is formed in order to consume the service. Hence, a challenge is the identity management in case of anonymous negotiations.

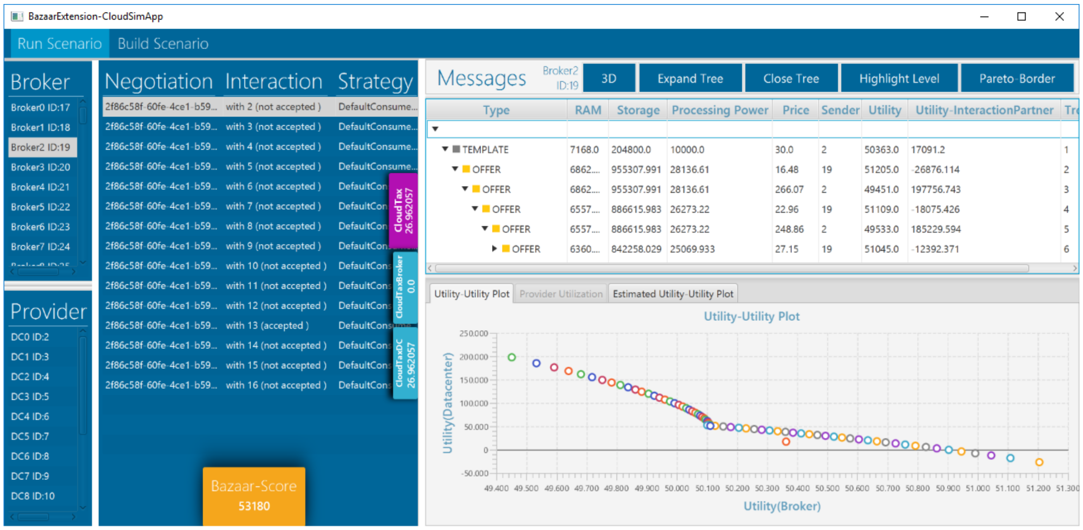

Figure 7: Simulation environment for simulating Bazaar-negotiations which access the Bazaar-Blockchain

(iii) As mentioned in the introduction the paper focuses on the blockchain - Smart Contracts are part of our further research. However, we want to discuss possible challenges of Smart Contracts which are intended to be used in the Bazaar-blockchain. The agreements formed between consumers and providers are stored in the Bazaar-blockchain. Hence, Smart Contract technology could be used for observing and executing the agreements. Therefore, the agreement has to be transformed to a Smart Contract. A main challenge is the enforcement of Smart Contracts where Clack et. al. distinguish between traditional methods and non-traditional methods [36]. Former includes methods such as the usage of courts which determine if the a delivered performance was appropriate. For latter Clack et. al. describe that there is *currently debate and experimentation on enforcing the actions of Smart Contract code at a network level without the need for dispute resolution*. However, if the network is unable to determine if a certain contract term has to be executed it has to call external functions [37] of a third party which determines if a certain condition is fulfilled or not [38]. Oracle middlewares such as ChainLink[5] help blockchain networks to validate contract-relevant conditions and consequently to execute Smart Contracts. For the network it is hard to identify if the Cloud service is delivered and if the payment is done - if no cryptocurrency is used. Hence, such oracles have to be used for Smart Contracts which observe if the service is delivered appropriately and if the payment is done.

VI. Conclusion

Bilateral, multi-round negotiations aka Bazaar-negotiations are a promising approach for trading Cloud services on future Cloud markets. Such negotiations are characterized by an alternating exchange of offers between untrusted market participants. Ensuring transparency and integrity during such negotiations is a main pillar towards the adaption of Bazaar-negotiations on the Cloud market. Therefore, we revert to the blockchain technology in the paper at hand. The introduced Bazaar-blockchain is used for reading and writing offers which are exchanged during negotiations as well as for agreements which result from such negotiations. Hence, the negotiation is stored in a tamper-safe log which allows the automatic execution and observation of the agreements with Smart Contracts. We implemented a prototype of the Bazaar-blockchain to show its technical feasibility. In our future work we focus on how to generate and design Smart Contracts to execute and observe the agreements.

[5] https://link.smartcontract.com/whitepaper

References

[1] F. Messina, R. Mikkilineni, G. Morana, and D. Rosaci, “Track chair’s report: Convergence of distributed clouds, grids and their management CDCGM 2017,” in *26th IEEE WETICE 2017, Poznan, Poland, 2017*, 2017, pp. 92–94.

[2] iDatalabs, “Companies using Amazon EC2,” https://idatalabs.com/tech/products/amazon-ec2, 2017, accessed: 2017-09-20.

[3] S. Chichin, Q. B. Vo, and R. Kowalczyk, “Towards efficient and truthful market mechanisms for double-sided cloud markets,” *IEEE Trans. Services Computing*, vol. 10, no. 1, pp. 37–51, 2017.

[4] W. Herrmann, “Deutsche boerse cloud exchange gibt auf,” in *Computerwoche*, 2016, https://www.computerwoche.de/a/deutsche-boerse-cloud-exchange-gibt -auf,3223201, Accessed: 2017-09-20.

[5] P. Samimi, Y. Teimouri, and M. Mukhtar, “A combinatorial double auction resource allocation model in cloud computing,” *Inf. Sci.*, vol. 357, pp. 201–216, 2016.

[6] P. Bonacquisto, G. D. Modica, G. Petralia, and O. Tomarchio, “A strategy to optimize resource allocation in auction-based cloud markets,” in *IEEE International Conference on Services*

Computing, SCC 2014, Anchorage, AK, USA, 2014, 2014, pp. 339–346.

[7] A. V. Dastjerdi and R. Buyya, "An autonomous time-dependent SLA negotiation strategy for cloud computing," *The Computer Journal*, p. 3202–3216, 2015.

[8] B. Pittl, I. U. Haq, W. Mach, and E. Schikuta, "Towards self-organizing cloud markets fostering intermediaries," in *26th IEEE WETICE 2017, Poznan, Poland, 2017*, 2017, pp. 131–136.

[9] X. Zhao, J. Yin, C. Zhi, and Z. Chen, "Monvalley: An unified monitoring and management framework for cloud services," in *IEEE International Conference on Web Services, ICWS 2016, San Francisco, CA, USA, 2016*, 2016, pp. 682–685.

[10] B. Pittl, W. Mach, and E. Schikuta, "A negotiation-based resource allocation model in iaas-markets," in *8th IEEE/ACM International Conference on Utility and Cloud Computing, UCC 2015, Limassol, Cyprus, 2015*, 2015, pp. 55–64.

[11] M. Mainelli, M. Smith *et al.*, "Sharing ledgers for sharing economies: an exploration of mutual distributed ledgers (aka blockchain technology)," *The Journal of Financial Perspectives*, vol. 3, no. 3, pp. 38–69, 2015.

[12] A. E. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, 2016*, 2016, pp. 839–858.

[13] T. I. Kiviat, "Beyond bitcoin: Issues in regulating blockchain tranactions," *Duke LJ*, vol. 65, p. 569, 2015.

[14] H.-G. Fill and F. Haerer, "Knowledge blockchains: Applying blockchain technologies to enterprise modeling," in *HICCS 2018*, 2018, pp. 4045–4054.

[15] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, "Untrusted business process monitoring and execution using blockchain," in *Business Process Management - 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, 2016. Proceedings*, 2016, pp. 329–347.

[16] Q. DuPont and B. Maurer, "Ledgers and law in the blockchain," *Kings Review (23 June 2015) http://kingsreview. co. uk/magazine/blog/2015/06/23/ledgers-and-law-in-the-blockchain*, 2015.

[17] R. Buyya et. al., "A manifesto for future generation cloud computing: Research directions for the next decade," *CoRR*, vol. abs/1711.09123, 2017.

[18] A. Walch, "The bitcoin blockchain as financial market infrastructure: a consideration of operational risk," 2015.

[19] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[20] S. Singh and N. Singh, "Blockchain: Future of financial and cyber security," in *Contemporary Computing and Informatics (IC3I), 2016 2nd International Conference on*. IEEE, 2016, pp. 463–467.

[21] M. Pilkington, "11 blockchain technology: principles and applications," *Research handbook on digital transformations*, p. 225, 2016.

[22] J. Mendling et. al., "Blockchains for business process management - challenges and opportunities," *CoRR*, vol. abs/1704.03610, 2017.

[23] C. Stagnaro, "White paper: Innovative blockchain uses in health care."

[24] I. Singh and S. Lee, "Comparative requirements analysis for the feasibility of blockchain for secure cloud," in *Requirements Engineering for Internet of Things - 4th Asia-Pacific Symposium, APRES 2017, Melaka, Malaysia, 2017, Proceedings*, 2017, pp. 57–72.

[25] A. Norta, "Creation of smart-contracting collaborations for decentralized autonomous organizations," in *Perspectives in Business Informatics Research - 14th International Conference, BIR 2015, Tartu, Estonia, 2015, Proceedings*, 2015, pp. 3–17.

[26] A. Norta, A. B. Othman, and K. Taveter, "Conflict-resolution lifecycles for governed decentralized autonomous organization collaboration," in *Proceedings of the 2015 2nd International Conference on Electronic Governance and Open Society: Challenges in Eurasia, EGOSE 2015, St. Petersburg, Russian Federation, November 24-25, 2015*, 2015, pp. 244–257.

[27] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Transactions on Smart Grid*, 2018.

[28] J. J. Karst and G. Brodar, "Connecting multiple devices with blockchain in the internet of things."

[29] O. Waeldrich, D. Battr, F. Brazier, K. Clark, M. Oey, A. Papaspyrou, P. Wieder, and W. Ziegler, "Ws-agreement negotiation version 1.0," in *Open Grid Forum.*, 2011.

[30] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, and M. Xu, "Web services agreement specification (WS-Agreement)," in *Open Grid Forum*, vol. 128, 2007.

[31] W. Mach, "A simulation environment for ws-agreement negotiation compliant strategies," in *iiWAS 2017, Salzburg*, 2017.

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

[33] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. E. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, D. Song, and R. Wattenhofer, "On scaling decentralized blockchains - (A position paper)," in *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, 2016, Revised Selected Papers*, 2016, pp. 106–125.

[34] E. Rosen, E. Wengrowski, G. D. Clark, and X. Gao, "Bitcoin: An empirical study of cryptocurrency," 2014.

[35] B. Pittl, W. Mach, and E. Schikuta, "Bazaar-extension: A cloudsim extension for simulating negotiation based resource allocations," in *IEEE International Conference on Services Computing, SCC 2016, San Francisco, CA, USA, 2016*, 2016, pp. 427–434.

[36] C. D. Clack, V. A. Bakshi, and L. Braine, "Smart contract templates: foundations, design landscape and research directions," *CoRR*, vol. abs/1608.00771, 2016.

[37] C. Frantz and M. Nowostawski, "From institutions to code: Towards automated generation of smart contracts," in *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W), Augsburg, Germany, 2016*, 2016, pp. 210–215.

[38] K. Delmolino, M. Arnett, A. E. Kosba, A. Miller, and E. Shi, "Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab," in *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, 2016, Revised Selected Papers*, 2016, pp. 79–94.