Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

BlockchainBus: A Lightweight Framework for Secure Virtual Machine Migration in Cloud Federations using Blockchain

Joseph Doyle, Muhammed Golec and Sukhpal Singh Gill*

School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

Correspondence
*Muhammed Golec, School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Rd, Bethnal Green, London E1 4NS, UK.
Email: m.golec@qmul.ac.uk

Summary

Cloud Federations offer numerous advantages such as improved resilience, improved performance and the prevention of vendor lock-in. In order to function efficiently, however, they must be supported by automated solutions to deploy cloud resources securely and efficiently. In particular, a system which allows virtual machine (VM) migration from one cloud to another while recording this for charging and security purposes is particularly useful. This paper presents a lightweight framework called BlockchainBus which offers secure VM migration in cloud federations using blockchain. This framework is a VM migration Blockchain ledger which can be used to record these migrations. BlockchainBus framework is implemented using the HyperLedger solution and deployed on the Microsoft Azure cloud platform. The overhead of this system is then evaluated. The performance of the BlockchainBus framework was compared with the overall VM migration time and was determined as 2.36 seconds. This result is shorter than the overall VM migration time of 5 seconds and indicates that the BlockchainBus gives better performance.

KEYWORDS:
Blockchain, Cloud Federation, Virtual Machine Migration, BlockchainBus, Cloud Computing, Security

1 | INTRODUCTION

Recently there have been numerous proposals for the creation of cloud federations which are platforms that are comprised of several computing clouds. These federations have numerous advantages such as improved resilience, improved performance and the prevention of vendor lock-in[1]. Cloud federations, however, must be supported by automated solutions to deploy cloud resources securely and efficiently[2]. Cloud federations are particularly appealing to small/medium cloud providers as they can improve the utilisation levels of resources by offering them to external users at low prices when utilisation levels are low. They can also borrow additional resources from other cloud providers when utilisation levels are high thereby ensuring that the Quality of Service (QoS) users receive is sufficiently high. This positively affects the Quality of Experience (QoE), which is the measure of the customer's positive and negative experiences with a service[3].

Cloud federations achieve this through virtual machine migration. Virtual machines (VMs) stop utilising the physical resources of one cloud provider and start utilising the physical resources of another. This allows providers to deploy distributed services over a large geographical area, improve fault tolerance by creating backups in different clouds, improve QoS as well as reducing energy cost and the associated environmental effects through consolidation[4].

[0] **Abbreviations:** VM, Virtual Machine; QoS, Quality of Service; 5G, 5th Generation

These are significant advantages for cloud federations, but given that critical technologies such as Internet of Things (IoT) devices work with clouds today, security and privacy is too important to ignore[5]. A cloud which is part of a federation should be able to authenticate itself once and gain access to the resources of the federation which belong to the same trust context without further identification. This authentication can guaranteed by a third party[6]. Using a third party, however, requires a central controller which has several disadvantages[7]. Records of the migration of VMs maintained by a central authority are vulnerable to manipulation. Thus, multiple copies of these records should be maintained so that in the event of security breaches, vulnerabilities can be identified, re-mediated and mitigated. Similar problems are found in the management of 5G network slices[8].

In 5G networks, a network slice broker can lease networks resources on-demand[9]. Users can request a service level agreement (SLA) or access to Over the Top (OTT) providers for video, voice and other services using the Service Capability Exposure Function (SCEF) module[10]. To enable automatic agreements, a trustworthy platform is required. By utilising blockchain to record the network resource negotiations, process speeds can be increased and security can be enhanced[8]. The transactions recorded in the blockchain can also be used for charging or recording SLA violations.

Similarly in cloud federations one cloud can lease resources from another on-demand. To facilitate agreements, we consider using blockchain as a trustworthy platform for recording the migration of VMs. To the author's knowledge this is the first work which utilise blockchain in this manner. In this paper, we present a system for using blockchain to record virtual machine migration which can be used for detecting SLA violations and security vulnerabilities as well as for charging. We also examine the overhead introduced by the system to evaluate its effect on performance.

The main contributions of this article to the literature using HyperLedger technology, which is a blockchain-based system, can be listed as follows:

- Proposing a lightweight framework for secure VM migration in cloud federations using blockchain called BlockchainBus,
- Increasing processing speed and security in VM Migration,
- Charging and recording violations when there are SLA violations,
- Recording migration processes in cloud federations to facilitate agreements between providers,
- Finally, to be a source of inspiration for blockchain-based systems to be proposed in VM technology in the future.

The rest of the article is structured as follows: Section 2 discusses the related works and background technologies. The review of the Blockchain technology is given in Section 3. The implementation part of our study is presented in Section 5. The performance evaluation is given in Section 6. Section 7 concludes the paper and proposes promising future research directions.

2 | RELATED WORK AND BACKGROUND

In this section, we review previous works and then explain some concepts and terms for a better understanding of the article.

2.1 | Related Work

There have been several proposals for replacing central controllers with blockchain based solutions. A storage system for IoT data which uses blockchain to implement access control is proposed[7]. Samdanis *et al.* propose using a blockchain system to manage a network slice broker in a 5G network[9]. Ali *et al.* describe Blockstack which provides a decentralized and server-less version of the DNS protocol[11]. Jiao *et al.* detail a blockchain based system for resource management in cloud/fog environments[12]. Chen *et al.* propose using blockchain to implement a trusted social network to prevent malicious content[13]. With the increase in the number of IoT devices, problems such as bandwidth and transmission delay have occurred. To solve these problems, Software-Based Network (SDN) technology, which separates the data and control environment, comes into play. SDN technology is vulnerable to attacks on the central control plane such as Distributed Denial of Service (DDoS) attacks[14]. Meng *et al.* propose a trust-based filtering method using SDN and blockchain technology in their proposed study[15]. In this way, it has been shown that SDN is stronger against attacks such as flood attacking compared to systems where only SDN is used. In another study, Yin *et al.* propose a blockchain-based data storage system that supports a new data update in the data communication security of autonomous vehicles, due to its unalterable and transparency features. In the proposed study, advantages such as data reliability

TABLE 1 Comparison of BlockchainBus with related works

Work	Used Area	Type of Implementation	Value Stored
Shafagh et al.[7]	Internet of Things	bitcoin testnet	IoT Device Data
Samdanis et al.[9]	5G Network	Theoretical Analysis	5G Network Slice Data
Ali et al.[11]	Storage Systems	Custom Implementation	Data Path
Jiao et al.[12]	Resource Management in Cloud/Fog Environments	Go Ethereum	Sensor Data
Chen et al.[13]	Social and Information Networks	Simulation Based	Social Network Data
Meng et al.[15]	Software-Defined Networking	Custom Implementation	SND Data
Yin et al.[16]	Data Storage System for Intelligent Vehicles	Simulation Based	Sensor Data
Jing et al.[17]	Information Processing Management	Custom Implementation	Copyright code
Alverenga et al.[18]	Blockchain based Framework in Network Function Virtualization	Custom Implementation	VNF Data
Zhang et al.[19]	Blockchain-Based Framework in Edge Computing	Ethereum	Mobile User Data
Zhao et al.[20]	Blockchain based VM measurements	Custom Implementation	VM Commands
BlockchainBus (this work)	Blockchain based Security in Cloud Federations	Hyperledger	VM Migrations

were obtained by reducing the data size with smart contracts and addressing too much data on the chain[16].Jing *et al.* propose a new blockchain-based copyright management system for software projects in their study. In this system, blockchain ensures fasteness and storage efficiency, making copyright traceable and unchangeable[17]. Intelligence is added to the network core with the integration of network function visualization (NFV) and service function chaining (SFC). Thanks to this added intelligence, many network users are in danger as it will be programmable. In their proposed study, Alverenga *et al.* uses secure virtual network service functions (VNFs) and propose a Blockchain-based architecture for the migration and management of VNFs[18]. Zhang *et al.* propose a Blockchain-based architecture to ensure security in Mobile Edge Computing (MEC), which is used with 5G technology to control edge-to-device latency and energy efficiency in its proposed system[19]. Zhao *et al.* proposes to securely store Virtual Machine (VM) measurement data, which is very important in integrity assessment and decision-making, in the IaaS cloud, using the proposed two-layer Blockchain framework[20].

There are many studies that have taken advantage of Blockchain, one of the trending topics of recent years, to move from a central controller to a distributed architecture that does not require any authority. But as far as we know, there is no study that uses Blockchain technology to securely record Virtual Machine Migration. With this study, we aim to break new ground in the literature. Table 1 shows a comparison of BlockchainBus with related works. When comparing the systems in Table 1 we can see that many utilize custom implementations or simulation based evaluations which are not suitable for production based systems[18]. Some implementations utilize Ethereum which is suitable for production systems but it requires Ethereum cryptocurrency to function. Thus, we have chosen to implement our system with Hyperledger as it is suitable for production systems and does not require cryptocurrency to function.

2.2 | Background Technologies

In this section we introduce cloud federations (Section 2.2.1), virtual machine migration (Section 2.2.2) and blockchain technology (Section 2.2.3) concepts.

2.2.1 | Cloud Federations

One of the key technologies which supports the cloud computing paradigm is virtualisation. Virtualisation obscures the underlying infrastructure from computational processes by preventing them from interacting directly with physical resources. A logical layer is introduced and computational processes communicate with this to access the physical resources. This interaction is managed by Virtual Machine Monitors (VMMs) or “hypervisors". VMMs either emulate hardware or intercept hardware calls from VMs allowing software on the virtual machine to function as if it was installed on a stand-alone hardware platform.

In a cloud federation each cloud provider will possess a virtualisation infrastructure[21]. On this infrastructure virtual machines are hosted to provide services to the clients of the cloud provider. Each user specifies the resources which should be assigned to each VM. Thus, the total resources required by the provider will fluctuate depending on demands of the users. If the total demand for resources exceeds the capabilities of a cloud, further requests for virtual machine instantiation are forwarded to other clouds in the federation. Thus, the cloud can enlarge its own virtualization resources by hosting its own virtual machines on different virtualisation infrastructures.

Frequently, different clouds are based upon different underlying virtualisation and storage technologies. Thus, migrating a virtual machine can require a lengthy conversion process before it can be moved to another cloud[2]. Another important function is interoperable security. A cloud should be able to authenticate itself once to gain access to resources belonging to a trust context without further checks. This authentication can guaranteed by a third party[6]. This should be possible even if clouds use different security technologies to allow a wide variety of clouds to participate in the cloud federation. There are several solutions to this problem[6,2,1]. In general, these solutions utilise a cloud manager programme which has knowledge of the clouds connected to the federation and can match resource requests with the desired infrastructure.

2.2.2 | Virtual Machine Migration

A key technology in the cloud computing paradigm is the ability to boot a VM image on any physical node in a data centre. It enables many of the key concepts of the paradigm such as elastic scaling, computation migration and resource consolidation. While it is not impossible to migrate a VM from one data centre to another there are a number of challenges which make this difficult. Firstly, VM images are typically quite large (usually 1 - 30 GB[22]) and this data must transferred before the VM can be migrated. In addition, services which are deployed on the cloud such as three-tier web applications, business analytics solutions and virtual clusters can involve multiple VM images[23]. This exacerbates the challenge as multiple images must be transferred, thereby, increasing the overall volume of data required for transfer. Secondly, it is likely that this data will be transferred over a wide area network with limited bandwidth. While it is possible that a dedicated connection exists between different data centres, it is unlikely that all clouds in a federation will possess this as it would make membership in a federation prohibitively expensive for small/medium data centers. Thus, it possible to cause performance problems if a large number of VMs are migrated to different clouds in the federation. Thirdly, the image type of the VM cloud must be supported in the destination. These images are not standarised and are not designed to support efficient cross-datacentre VM image transfer. In addition, while it is possible to convert the image to an acceptable format in some cases, this is not always supported for strategic and technical reasons. Finally, it is likely that the migration process needs to be completed in a reasonable time. The conversion of the image and the throttling of the image transfer connection to prevent service interference are likely to delay the overall migration time which makes the challenge more complex.

There have been several proposals to solve these challenges[22,24,25]. In principle, they are centered around three areas. Firstly, the data required for transfer can be reduced as the VMs are frequently quite similar. This is particularly true if the virtual machines use the same OS distribution. Secondly, it is also possible to reduce the data volume transferred by comparing the image with a VM image that already exists in the destination data centre as it is likely that some of the VM image already exists there. Finally, traffic management techniques can be utilised to prioritise specific data for transfer to reduce the application migration time.

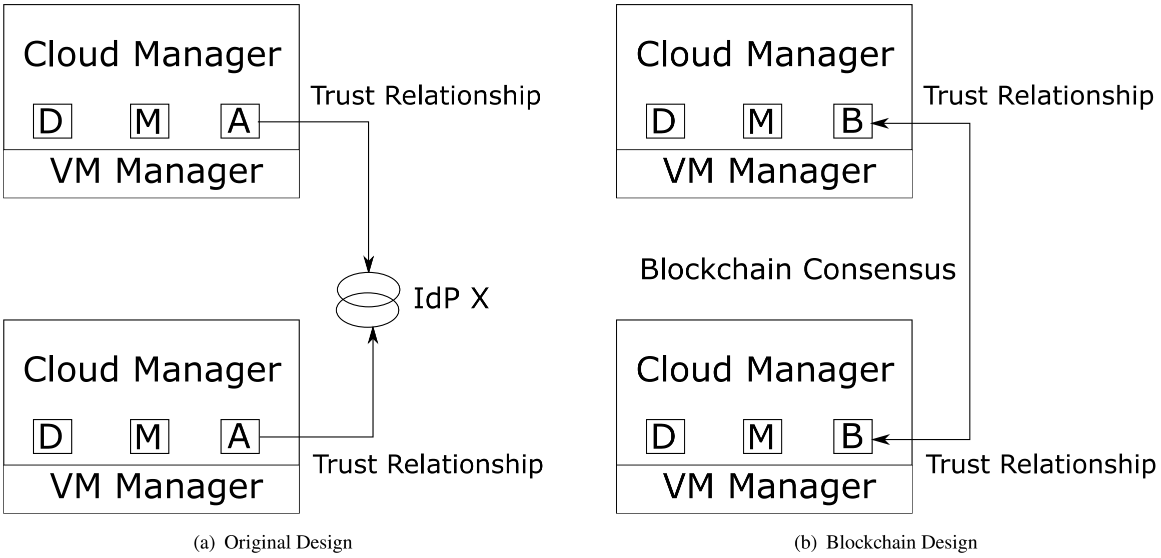
2.2.3 | Blockchain

Blockchain was first developed for the Bitcoin cryptocurrency[26] as a distributed database which maintains a list of data records. These records are confirmed by the nodes which participate in the blockchain network. The central part of a blockchain is a public ledger which records information on every transaction which is completed. The key difference between blockchain and other previous solutions is that it does not require any third party organizations to validate transactions. Information about every transaction is shared with all nodes in the blockchain network and is thus, available at each node. As a result the system is more transparent than other solutions[27].

A blockchain network functions as a Peer-2-Peer (P2P) network. In the case of the Bitcoin network each node collectively validates the information transactions which are submitted by other nodes. New transactions are sent to all the nodes. These transactions are collected into blocks and each node attempts to find a difficult proof-of-work for its block. Essentially, the proof-of-work is the search for a particular value which when combined with the block produces a particular hash. When a proof-of-work is found it is sent to all nodes[28]. Nodes will only accept the block if all transactions in it are valid. The nodes will then express their acceptance of the block by working on creating the next block in the chain using the hash of the accepted block.

Using this decentralised consensus mechanism participants in a P2P network can agree upon the contents of the distributed database. Part of the reason that blockchain was such as a success is that the system was previously considered impossible to implement[29] and as such it could be considered a system with emergent properties.

Blockchain networks can also use smart contracts to enable the automation of complex multi-step processes[30]. Smart contracts are defined as “a computerized transaction protocol that executes the terms of a contract"[31]. Smart contracts are scripts which



(a) Original Design (b) Blockchain Design

FIGURE 1 An overview of Cloud Federation design including the original design proposed in[6] and the adapted blockchain version.

are stored on the blockchain network. They function as autonomous actors whose behavior is fully predictable. Their code can be examined by every participant on the blockchain network as they reside on the chain and have a unique address. The contract's operations are also fully traceable as each transaction will be signed with the contract's key. The use of smart contracts in blockchain networks has enabled the use of this architecture in distributed applications such as voting, auctions, lottery, escrow systems, crowd funding and micropayments[32].

3 | BLOCKCHAIN VIRTUAL MACHINE MIGRATION LEDGER CONCEPT

As described in Section 2.2.1, VM migration in cloud federations facilitates the access of additional resources by computing clouds while ensuring security. This is contingent on the cloud being able to easily and automatically negotiate with external clouds based upon their current requirements. Previous solutions have relied upon a cloud manager program to implement this negotiation. In the next section, we briefly review how this is implemented by Celesti *et al.*[6] and show how blockchain could be used to enhance this functionality.

3.1 | Current Cloud Federation Resource Provisioning Manager

One solution for facilitating virtual machine migration in cloud federations was presented by Celesti *et al.*[6]. The cloud manager consists of three agents. The first agent is the discovery agent which publishes on-demand information about its supported features and resources state to other clouds which intend to be part of the federation. When a cloud wishes to discover which clouds are currently part of the federation it uses the discovery agent to check the shared area where this information is stored. The second agent is the match-making agent which selects the most convenient cloud to utilise when a particular cloud needs additional resources. This is achieved by examining the policies which are published by each cloud in the federation and the requirements of the cloud seeking resources. A best “fit" for the requirement is achieved via policy matching. Clouds which match the requirements of the resource seeking cloud are sorted according to an estimation of how closely the policy of the cloud matches the requirements of the resource seeking cloud. The resource seeking cloud can then iterate through this list to migrate its virtual machines to the most suitable cloud.

The third agent is the authentication agent which is responsible for creating a security context between the clouds using a third party trusted entity. This agent contacts the agent in the cloud with the desired resources and exchanges authentication information in the form of meta-data. A trusted third party is also involved in this process. This can be challenging as each cloud can support different authentication mechanisms. One way to achieve this is to use the Identify Provider/Service Provider (IdP/SP) model. In this model the cloud or asserting party hold at least one digital identity on an identity provider which provides access to the resources of service providers. Essentially, once a cloud has logged into the identity provider they are trusted with access to all resources associated with a particular authentication context. The identity provider acts as a third party asserting that the cloud which has logged on is trustworthy.

This can lead to several problems. Firstly, the logs recording the migrations of virtual machines are a central point of failure. If a participant in the federation gains access to the logs then they be manipulated maliciously for economic gain. For example, a participant could alter the time a virtual machine spent using its resources to increase the payment it would receive from another participant. Secondly, the logs are not tamper proof. One of the features of blockchain is that it is more difficult to alter a log in the distant past when compared with the recent past. This is achieved by using a hash of the previous block in the chain. The more hashes that need to be generated the more difficult it is to alter a record. Finally, VM escape where malicious code escapes from a virtual machine to the host or hypervisor which is executing the VM[33] and VM hopping where malicious code escapes from a VM to another VM being executed on the same physical machine[34] can cause issues in federated clouds. For example, a participant in the federation could execute a VM escape attack and gain access to the hypervisor. Once this has been achieved it could alter the resource constraints of its VMs or start new VMs on the physical machine to the detriment of other VMs utilising the same resources. While a blockchain solution would not prevent this it would make it more difficult to alter the logs to disguise these actions. Thus, the motivation for implementing these attacks is limited as they are likely to be discovered. The blockchain solution proposed in this paper essentially replaces the authentication agent with a blockchain solution to prevent these issues so that a third centralised party is not required to assert that a cloud which is part of the federation is trustworthy. This is depicted in Figure 1 and discussed in greater detail in the next section.

3.2 | Virtual Machine Migration Ledger

By replacing the trusted third party with blockchain the proposed system provides some features which are not viable using existing digital structures. The use of blockchain lowers the barrier for collaboration and enables a larger efficient ecosystem as all transactions are verified before being committed to the database and are visible to every cloud participating in the federation. Each cloud in the federation will possess unique digital keys which they use to sign and verify transactions. In this case the basic transaction can be defined as one cloud migrating a VM image to another cloud with specific resource requirements associated with the VM which will run on this cloud. For example “cloud X sends virtual machine Y with resource constraints R to cloud Z" where R is a vector of resource constraints relating to CPU, Memory, Hard Disk I/O and Network I/O[1]. This is similar to the standard Bitcoin Transaction "payer X sends Y bitcoins to payee Z" without the resource constraints. Additionally, as the Virtual Machine Migration Ledger is a permissioned blockchain system the identity of every cloud participating in the federation is tightly interconnected with their signature. This can be used as a foundation for charging and billing between clouds.

Moreover, smart contracts can be used to improve the automation of virtual machine migration. By using resource utilisation levels and prices as smart contract parameters negotiations for resource become more efficient. When two cloud have agreed on the service terms of the virtual machine migration this agreement can be timestamped with the signatures of both clouds and stored in the blockchain. The smart contracts are not able to access external data. Hence, an *oracle* is required to determine if the VMs which are running on different clouds in the federation receive the agreed upon SLA to determine if an alteration to the price or compensation is required due to SLA violations[35].

When designing blockchain networks it is important to minimise the data stored in the database as each transaction must be verified and stored at each blockchain node. Therefore, the proposed ledger only stores the following information:

- Timestamps recording when the virtual machine starts and finishes running in a cloud
- Agreed resource constraints for virtual machines
- Performance data for virtual machines
- Charging Evidence

This could be achieved using blockchain initiatives such as Hyperledger[36]. Additionally, the Hyperledger platform also has registration and identity management services which could be used to simplify the operation of the cloud federation. Finally,

[1] This list is non exhaustive and could be extended if deemed necessary.

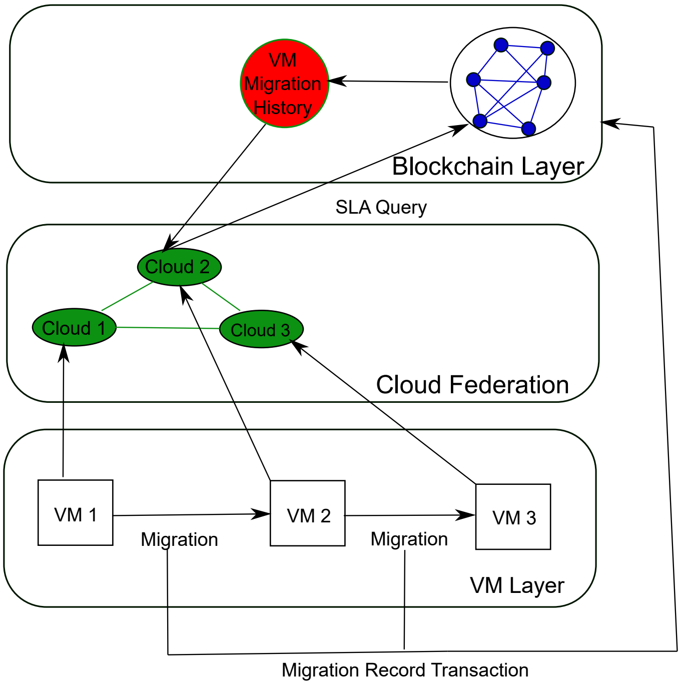


FIGURE 2 BlockchainBus System Design

it would be possible to implement a reputation system[37] in the proposal by assigning “reputation points" to clouds depending on how well they perform in line with SLAs. This could be valuable as it may allow small/medium clouds to participate more effectively in the federation.

4 | BLOCKCHAINBUS: A SYSTEM DESIGN

The BlockchainBus system is present in Figure 2. Each of the clouds in the cloud federation layer hosts VMs and also maintains a blockchain layer built which maintains the records of migrations of VMs between the different clouds. As new clouds join the federation they also join the blockchain network. When VMs migrate from one cloud to another a record of this is recorded in a transaction in the blockchain network. The blockchain network will groups these transactions into blocks and verify them to maintain a tamper proof record of these migrations. If there are SLA violations a cloud can query the blockchain network and receive a migration history which it can use to determine if which cloud is responsible for the SLA violation.

5 | IMPLEMENTATION OF BLOCKCHAIN SOLUTION FOR VIRTUAL MACHINE MIGRATION IN CLOUD FEDERATIONS

To support the implementation of blockchain solutions for recording virtual machine migration in cloud federations we have designed a Hyperledger composer solution which records virtual machines operating in a cloud, cloud who are participants in the federation and the migration of virtual machines from one cloud to another[2]. The code for composer model is depicted in Figure 3. A composer model is presented rather than pseudocode as it more accurately captures the transactional nature of the system which uses blockchain rather than an algorithmic nature which is better represented by pseudocode. The HyperLedger composer model is based around a business network concept which models:

- Goods and services that can be exchanged.
- How the exchange takes place

[2] The repository for this solution is available at https://github.com/Muhammed1616/BlockchainBus

```
namespace org.cloudfederationblockchain.network
participant Cloud identified by cloudId {
o String cloudId
o String Name
}
asset VirtualMachine identified by virtualMachineId {
o String virtualMachineId
–> Cloud operator
o Integer cpuCoreRequirements
o Integer memoryRequirements
o Double networkRequirements
o Double ioRequirements
}
transaction Migrate {
–> VirtualMachine virtualMachine
–> Cloud newOperator
}
```

FIGURE 3 Code for Cloud Federation HyperLedger composer model

- The groups that are allow to participate

This model adapts readily to the Cloud Federation paradigm as we can see from Figure 3. Virtual machines are modeled as “assets" which contain data on their operational requirements as well as an identifying string and data on the cloud on which they are currently operating. Clouds are modelled as “participants". In the current model, only basic data about the cloud is provided. In future work, we hope to extend the model so that resource availability in the cloud is also modelled so that this can be consulted during the migration process to determine if the transaction is successful and the VM can be migrated. Finally, the migration, of a virtual machine is modelled as a transaction. Further logic to control the transaction is placed in the javascript “logic.js" file. If the logic executes correctly then the transaction will be stored in the database with a timestamp and a transaction id. The model is deployed on HyperLedger Fabric which implements the blockchain framework. Transactions can be recorded in the database via a REST server interface.

6 | EVALUATION OF HYPERLEDGER IMPLEMENTATION

In order to evaluate the HyperLedger implementation we deployed it on the Microsoft Azure cloud platform. The implementation was deployed on a standard B2s instance with 2 virtual cpus and 4 GB of memory in the West Europe region. A critical performance metric for recording virtual machine migration is the overhead introduced as a result of recording the transactions. After deploying the solution on the installed HyperLedger platform we recorded the duration of a number of transactions recording virtual machine migration via the REST interface. Figure 3 shows the result of the operation. The average duration of these transactions was 2.36 seconds. It should be noted that no optimization was carried out in order to improve this value and it is likely that this figure can be significantly reduced.

In order to determine if this is an acceptable overhead it is important to determine the overall virtual machine migration time. This varies considerably depending on the type of virtual machine migration. An overall maximum time for virtual machine migration is set at 5 seconds in[38]. As the creation of the transaction could be done concurrently with other migration actions the overhead is sufficiently small that the system can be deemed viable. It should be noted that this target assumes that no machine image conversion is required and that a 1Gbps connection is available between the two clouds. Moreover, the system in[38] is based around the notion of live migration which may not be required for cloud federations. Clouds may offload batch jobs to other members of the federations in order to prioritise their own live virtual machines which are frequently considered more crucial. The duration of the conversion process will varying considerably depending on the resource which are assigned to the task. In Figure 4, the general virtual machine migration time and the migration time of our blockchain-based system recommended in our study are compared.

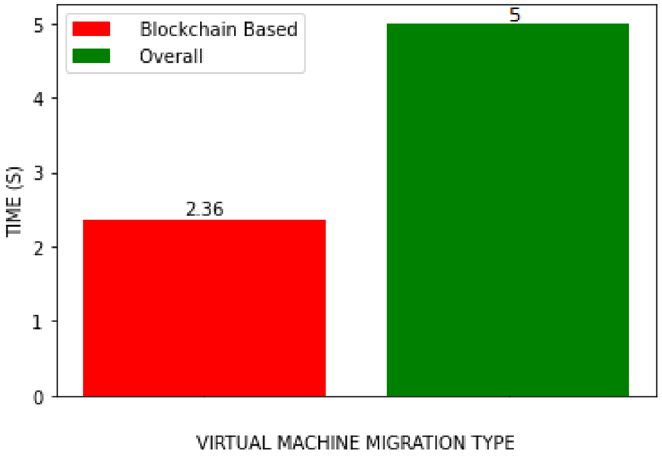


FIGURE 4 The Comparison of Virtual Machine Migration Times.

If virtual machine image conversion is not required the duration of the transfer from one cloud to another will affect the virtual machine migration process. This was also evaluated to provide insight into the overall cloud federation system. A large file from the B2s Microsoft Azure instance in West Europe region was transferred to an Amazon Web Service m3.medium instance which has 1 virtual CPU and 3.75GB of memory in US-East region numerous times. The average speed of this transfer was recorded as 124.8Mbs. Figure 4 shows the processing time. While the speed will vary considerably it can be seen as a general indicator of the difficulty in achieving a virtual machine migration time of less than 5 seconds as the speed achieved was nearly an order of magnitude smaller than the required 1Gbps. In addition, this is a transfer between two large commercial clouds who have considerable networking infrastructure. Speeds of this magnitude may not be possible for all clouds in a federation. Thus, the 5 second target in[38] may be somewhat ambitious and the transaction overhead may be an even smaller component in the overhead of the migration service. It should be noted, however, that containerisation technologies such as Docker are becoming more popular[39,40] and an average speed of 124.8Mbs would be sufficient to transfer a Docker image from one cloud to another in less than 5 seconds. Thus, we conclude that the overhead of a blockchain solution is viable and supporting technologies could make the overall system viable. In Figure 5, the comparison of the transfer speed and required transfer speed obtained in the study is shown. Thus, we demonstrate that the proposed system can be used to record VM migrations between different cloud providers as the time required to add a migration to the blockchain is less than the time required for a VM to migrate. The other contributions are secured by the nature of the blockchain solution. SLA violations can be detected by having a tamper proof record that proves that VMs were hosted by a provider at a given time and any performance issues can be deemed the responsibility of that provider. Security is improved by removing the need for an authentication authority which acts as a central point of failure.

7 | CONCLUSIONS AND FUTURE WORK

The lack of trust and the need for automated solutions for the deployment of cloud resources securely and efficiently make a decentralised solution more attractive to participants of cloud federations. This paper presented a BlockchainBus framework which implements a virtual machine migration ledger. Using this framework clouds would join the blockchain network and dynamically migrate virtual machines to different clouds as required. As the transactions are validated in the framework any transactions which are contradictory would not be recorded. It is therefore possible to develop a system where there is sufficient trust to utilise the transactions recorded in the blockchain for charging. It is also possible to utilise the transaction as a record of where the VMs where hosted in the event of security breaches so that vulnerabilities can be identified, remediated and mitigated.

The system was implemented using HyperLedger and deployed on the Microsoft Azure Cloud platform. This platform was evaluated with experiments to determine the duration of the submission of a transaction as well as experiments to determine the likely duration of a machine image transfer from the Microsoft Azure cloud platform to the Amazon Web Services cloud platform. From this the paper determines that blockchain can be used to support virtual machine migration in cloud federations.

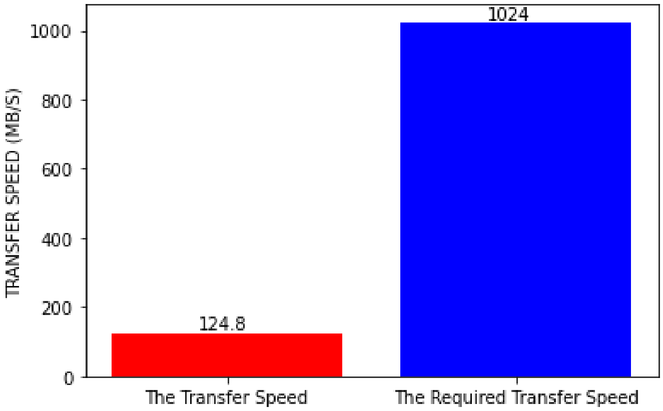


FIGURE 5 The Comparison of Transfer Speed.

In future work, we will expand the system so that joining a cloud federation platform is integrated with joining the blockchain network as well as incorporating resource requirements and availability in the HyperLedger model structure. Further, BlockchainBus framework can be integrated with iFaasBus[41] to improve the scalability in the future.

8 | DATA AVAILABILITY STATEMENT

We have released BlockchainBus as an open source. The data and code that support the findings of this study are openly available in GitHub at https://github.com/Muhammed1616/BlockchainBus

9 | FUNDING

Funding information is not applicable

10 | CONFLICTS OF INTEREST

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Kurze T, Klems M, Bermbach D, Lenk A, Tai S, Kunze M. Cloud federation. *Cloud Computing* 2011; 2011: 32–38.
2. Huedo E, Montero RS, Moreno R, Llorente IM, Levin A, Massonet P. Interoperable federated cloud networking. *IEEE Internet Computing* 2017; 21(5): 54–59.
3. Taha M, Canovas A, Lloret J, Ali A. A QoE adaptive management system for high definition video streaming over wireless networks. *Telecommunication Systems* 2021; 77(1): 63–81.
4. Giacobbe M, Celesti A, Fazio M, Villari M, Puliafito A. An approach to reduce carbon dioxide emissions through virtual machine migrations in a sustainable cloud federation. In: IEEE. ; 2015: 1–4.
5. Golec M, Gill SS, Bahsoon R, Rana O. BioSec: A Biometric Authentication Framework for Secure and Private Communication among Edge Devices in IoT and Industry 4.0. *IEEE Consumer Electronics Magazine* 2020: 1-1.
6. Celesti A, Tusa F, Villari M, Puliafito A. How to enhance cloud architectures to enable cross-federation. In: IEEE. ; 2010: 337–345.

7. Shafagh H, Burkhalter L, Hithnawi A, Duquennoy S. Towards blockchain-based auditable storage and sharing of IoT data. In: ACM. ; 2017: 45–50.

8. Backman J, Yrjölä S, Valtanen K, Mämmelä O. Blockchain network slice broker in 5G: Slice leasing in factory of the future use case. In: IEEE. ; 2017: 1–8.

9. Samdanis K, Costa-Perez X, Sciancalepore V. From network sharing to multi-tenancy: The 5G network slice broker. *IEEE Communications Magazine* 2016; 54(7): 32–39.

10. Matinmikko M, Latva-Aho M, Ahokangas P, Yrjölä S, Koivumäki T. Micro operators to boost local service delivery in 5G. *Wireless Personal Communications* 2017; 95(1): 69–82.

11. Ali M, Nelson J, Shea R, Freedman MJ. Blockstack: A global naming and storage system secured by blockchains. In: ; 2016: 181–194.

12. Jiao Y, Wang P, Niyato D, Suankaewmanee K. Auction mechanisms in cloud/fog computing resource allocation for public blockchain networks. *IEEE Transactions on Parallel and Distributed Systems* 2019.

13. Chen Y, Li Q, Wang H. Towards trusted social networks with blockchain technology. *arXiv preprint arXiv:1801.02796* 2018.

14. Gkountis C, Taha M, Lloret J, Kambourakis G. Lightweight algorithm for protecting SDN controller against DDoS attacks. In: ; 2017: 1-6.

15. Meng W, Li W, Zhou J. Enhancing the security of blockchain-based software defined networking through trust-based traffic fusion and filtration. *Information Fusion* 2021; 70: 60–71.

16. Yin Y, Li Y, Ye B, Liang T, Li Y. A Blockchain-based Incremental Update Supported Data Storage System for Intelligent Vehicles. *IEEE Transactions on Vehicular Technology* 2021: 1-1.

17. Jing N, Liu Q, Sugumaran V. A blockchain-based code copyright management system. *Information Processing Management* 2021; 58(3): 102518.

18. Alvarenga ID, Rebello GAF, Duarte OCMB. Securing configuration management and migration of virtual network functions using blockchain. In: ; 2018: 1-9.

19. Zhang X, Wu W, Yang S, Wang X. Falcon: A Blockchain-Based Edge Service Migration Framework in MEC. *Mobile Information Systems* 2020; 2020: 1–17.

20. Zhao B, Fan P, Ni M. Mchain: A Blockchain-Based VM Measurements Secure Storage Approach in IaaS Cloud With Enhanced Integrity and Controllability. *IEEE Access* 2018; 6: 43758-43769.

21. Yang M, Margheri A, Hu R, Sassone V. Differentially private data sharing in a cloud federation with blockchain. *IEEE Cloud Computing* 2018; 5(6): 69–79.

22. Al-Kiswany S, Subhraveti D, Sarkar P, Ripeanu M. VMFlock: virtual machine co-migration for the cloud. In: ACM. ; 2011: 159–170.

23. Chieu TC, Mohindra A, Karve A, Segal A. Solution-based deployment of complex application services on a cloud. In: IEEE. ; 2010: 282–287.

24. Liu H, He B. Vmbuddies: Coordinating live migration of multi-tier applications in cloud environments. *IEEE transactions on parallel and distributed systems* 2015; 26(4): 1192–1205.

25. Deshpande U, Keahey K. Traffic-sensitive live migration of virtual machines. *Future Generation Computer Systems* 2017; 72: 118–128.

26. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008.

27. Yli-Huumo J, Ko D, Choi S, Park S, Smolander K. Where is current research on blockchain technology?—a systematic review. *PloS one* 2016; 11(10): e0163477.

28. Gill SS, Tuli S, Xu M, et al. Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges. *Internet of Things* 2019; 8: 100118.

29. Fischer MJ, Lynch NA, Paterson MS. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)* 1985; 32(2): 374–382.

30. Christidis K, Devetsikiotis M. Blockchains and smart contracts for the internet of things. *IEEE Access* 2016; 4: 2292–2303.

31. Szabo N. Smart contracts. *Unpublished manuscript* 1994.

32. Baliga A. The Blockchain Landscape. *Persistent Systems* 2016.

33. Grobauer B, Walloschek T, Stocker E. Understanding Cloud Computing Vulnerabilities. *IEEE Security Privacy* 2011; 9(2): 50-57.

34. Ormandy T. An empirical study into the security exposure to hosts of hostile virtualized environments. 2007.

35. Xu X, Pautasso C, Zhu L, et al. The blockchain as a software connector. In: IEEE. ; 2016: 182–191.

36. Linux Foundation r. Hyperledger Project. 2016.

37. Tapscott D, Tapscott A. *Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world*. Penguin . 2016.

38. Mashtizadeh AJ, Cai M, Tarasuk-Levin G, Koller R, Garfinkel T, Setty S. XvMotion: Unified Virtual Machine Migration over Long Distance.. In: ; 2014: 97–108.

39. Chamberlain R, Schommer J. Using Docker to support reproducible research. *DOI: https://doi. org/10.6084/m9.figshare* 2014; 1101910: 44.

40. Moreews F, Sallou O, Ménager H, others . BioShaDock: a community driven bioinformatics shared Docker-based tools registry. *F1000Research* 2015; 4.

41. Golec M, Ozturac R, Pooranian Z, Gill SS, Buyya R. iFaaSBus: A Security and Privacy based Lightweight Framework for Serverless Computing using IoT and Machine Learning.. *IEEE Transactions on Industrial Informatics* 2021.

How to cite this article: J. Doyle, M. Golec and S. S. Gill (2021s), BlockchainBus: A Lightweight Framework for Secure Virtual Machine Migration in Cloud Federations using Blockchain , *Security and Privacy*, *2021;00:x–x*.