UNIVERSITÉ DE GENÈVE

MASTER THESIS
16INFOTRAV

Thesis Name

Teacher Supervisor: - -

External Supervisor: Florent Martin

Author: João Quinta

March 2023

UNIVERSITÉ
DE GENÈVE

FACULTÉ DES SCIENCES
Département d'informatique

Contents

1 Introduction

TODO: define city actors (CA)

2 Related Work and Challenges petite intro des technologies que que je vais introduire

In this section, we discuss the definition of Digital Twin as well as Urban Digital Twin and its applications. Moreover, we show how smart contracts can help manage access control and how key exchange protocols can be set up for group communication.

2.1 Digital Twin intro DT

A Digital Twin (DT) can be described as a virtual instance of a Physical Twin (PT), the PT can be an object, system or even process, enabling real-time monitoring, analysis and optimization through the PT's life cycle[1]. It is important to note that the definition of a DT may vary somewhat depending on the context of application. Nonetheless, the core concept of a virtual instance that mirrors its physical counterpart for context-dependent purposes remains true across these variations[1].

digital model vs digital shadow vs digital twin

In [1, 2] the authors mention common misconceptions and highlight the different between digital models, digital shadows, and digital twins. Figure 1 illustrates these distinctions:

- Digital Model or mirror model is a digital representation of a pre-existing physical object. However, if the PT were to evolve, the digital model won't change accordingly, this is due to the fact that there is no automatic data exchange between them.
- A Digital Shadow goes one step further, by adding a one-way data flow between the physical and digital object. This means that the digital object is capable of evolving with the physical object.
- A DT is characterized by a bidirectional data flow between the physical and virtual object. This two-way integrated data flow, allows both twins to evolve side-by-side.

In a later section, we will address the security issues specifically associated with the bi-directional link in DT, and discuss potential solutions to overcome these challenges.

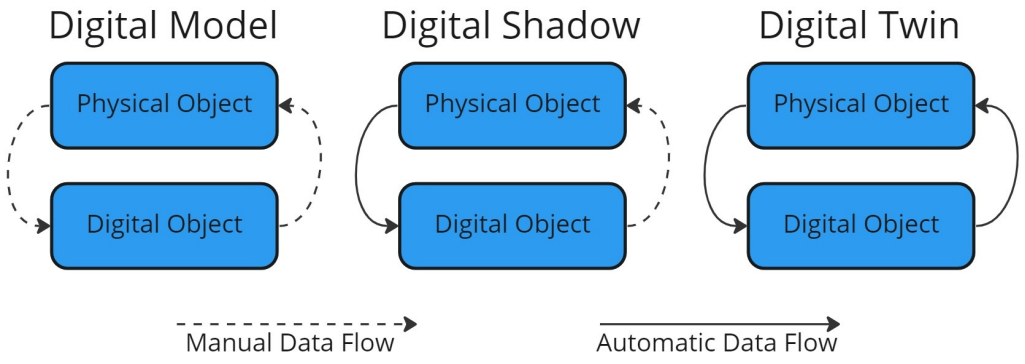

Figure 1: Difference between digital Model, Digital Shadow and Digital Twin

5D-DT architecture description

The models described above implement a three-dimension architecture, the physical entity its virtual model, and connection between them. In [3] the author extends it into a five-dimension digital twin (5D-DT) architecture, adding data and services dimensions. Figure 2 illustrates the five dimensions.

- Physical Entity: Consists of various functional subsystems that perform various tasks, as well as sensors that collect the states of the subsystems.
- Virtual Entity: Represents high fidelity integration of the data gathered from the sensors in the physical entity.
- Services: Refers to the functionalities the DT provides, such as simulation, optimization, and visualization.
- Data: Represents the contextual information of the DT, including real-time data, historical data, and predicted data.
- Connection: The bi-directional link between Physical and Virtual entity.

5D-DT framework x microservice architecture + microservice intro

The 5D-DT framework inherently aligns with the principals of microservice architecture [4], which has become more relevant when developing distributed software applications as it assures scalability and maintainability. The idea is to create applications by combining small independent software services, each focusing on a single functionality. These services are composed to form a cohesive and functional application [5], using either orchestration or choreography. Orchestration represents a centralized approach, while choreography is a decentralized. It is possible to use both compositions together, depending of the specific application.
Inter-service communication represents another crucial design decision in microservice architecture. Communication can be established using asynchronous or synchronous request-reply patterns, or event-driven asynchronous message exchange. In microservice architecture it is preferred to use event-driven communication as it promotes decoupling of the different services. A commonly used event-driven asynchronous communication solution is Apache Kafka, it offers fault-tolerant, salable, and stream-based messaging.
By leveraging the modular and flexible nature of microservice architecture, the 5D-DT framework can integrate various technologies and functionalities, making it suitable for designing and implementing DTs, or even Digital Twin Networks (DTN) which we'll discuss in the next section.

advantages of the combination

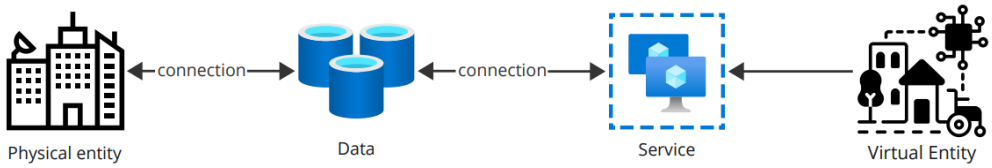

Figure 2: Five-dimension Digital Twin

2.1.1 Digital Twin Challenges

challenges chez DT

By looking at figure 1, the three different models progressively add layers of functionality and interactivity within the system. However, new functionalities usually introduce new security challenges, as discussed below:

challenges de digital model --> challenges digital shadow --> challenges DT

- Digital Model: The primary challenge is to ensure that all sensors used to capture the physical details of the PT are accurate and reliable. Data inaccuracy can lead to discrepancies between the digital model and its physical counterpart, which might impact any analysis and decisions made based on the digital model.

- Digital Shadow: In addition to the accuracy challenges faced in digital models, digital shadows require a secured and constant channel for continuous transmission of data from the sensors incorporated in the physical object. Ensuring that the data flow is accurate, timely and protected from potential attacks is critical for maintaining the integrity of the digital shadow. This secure channel will be referred as the Physical-to-Virtual (P2V), in a later section we will discuss the security requirements for this communication channel.

- Digital Twin: Since the flow of data is bi-directional, the virtual model will be able to modify the state of the PT, this increases the potential risks in the system if an attacker compromises the DT. In such a scenario, the attacker could manipulate the PT, potentially endangering human lives. The flow of data from the DT to the PT, will go through the same P2V channel.

example of dangers / why security is REQUIRED

In [6], the author describes how a computer system made a decision based on sensor data, the data was gathered by a single sensor. Because the sensor was faulty, the data it fed didn't accurately describe the current state of the physical object. The decision made based on the faulty data resulted in the death on multiple humans. In this system there were no backup sensors, and no verification was performed on the quality of the data.

gives an idea of surface of attack

Finally, Fuller et al. in [1], highlight additional challenges in the field of DT research. One challenged is the lack of a single objective within the research community, resulting in dispersed research efforts. Another challenge is the lack of standardization within the research of DT, which can delay the progress of DT technologies across different domains.

DT is a broad research field --> lack of single "focus point"

2.2 Urban Digital Twin

define Digital Twin Network --> UDT is DTN

DT establish a one-to-one mapping relationship between a physical and virtual space, whereas Digital Twin Network (DTN) is characterized by a many-to-many mapping relationship [2]. An Urban Digital Twin (UDT) is an extension of the DT concept, applied to entire urban environments or city-scale systems, figure 3 is an illustration of potentially involved City Actors (CA). This means than an UDT is essentially a DTN applied to a specific urban context. UDTs enable real-time monitoring, analysis and optimization of various functions and domains, thus playing a crucial role

define UDT as a n-to-n relationship of Digital Twins
each DT being controlled by City Actor

in the development of Smart Cities [7].

In recent years, researchers have investigated various UDT potential applications in various urban related domain. Here are some examples: various applications of UDT --> why its important to develop this technology

- Data Management: The data generated by the various CA present in a UDT will be both vast and of different formats, hence the ability to manage and process city data is important. To deal with heterogeneous data [8] proposes the utilization of the ontological approach, which was suggested to improve the semantic interoperability and secure future data expansions. Developing data standardization and data-sharing frameworks is fundamental to enhance data exchange among the various CA.

- Visualization: Enhancing the visualizations of the city can bring better understanding of the urban environment, which can lead to an overall better understanding of the urban environment and fewer design errors. UDTs can utilize and integrate various visualization tools the improve the current experience. For example in [9] they developed a DT of a city, and were able to visualize it almost seamlessly across all scales with the use of virtual and augmented reality.

- Situational awareness: Improve the visibility of the city as well as the understanding and analysis of urban events and operations. This will be achieved by harvesting multiple data points from the various CA. For instance, monitor noise pollution in the city [10], analyze energy consumption patterns [11].

- Planning and prediction: By analyzing past and present data, UDTs can provide valuable insights into future patterns and plans, enabling optimization of city functions. Some examples include optimizing electric vehicle charging locations [12] and even conducting flooding scenario analysis [13].

- Integration and collaboration: UDTs facilitate the integration of complex city elements and promote collaboration among CA, the goal being the co-development of the city. To that end, two avenues have been proposed. The first one argues that developing multiple DTs for the city may be more achievable than developing a single DT for the whole city [14]. The second one suggests the creation of a collaborative platform, used by the various CA, this platform would be used for data sharing among CA [15].

UDTs are bound to play a large role on how cities will evolve and be designed in the future. By leveraging the power of DT at a city scale, the various CA will be able to collaborate and share resources as they never were before, with the goal of making data-driven decisions dedicated to solve complex challenges of modern cities.
In a later section we will mention some of the key challenges that the development of UDT and all its applications face.

idea of collaboration among CA for city development

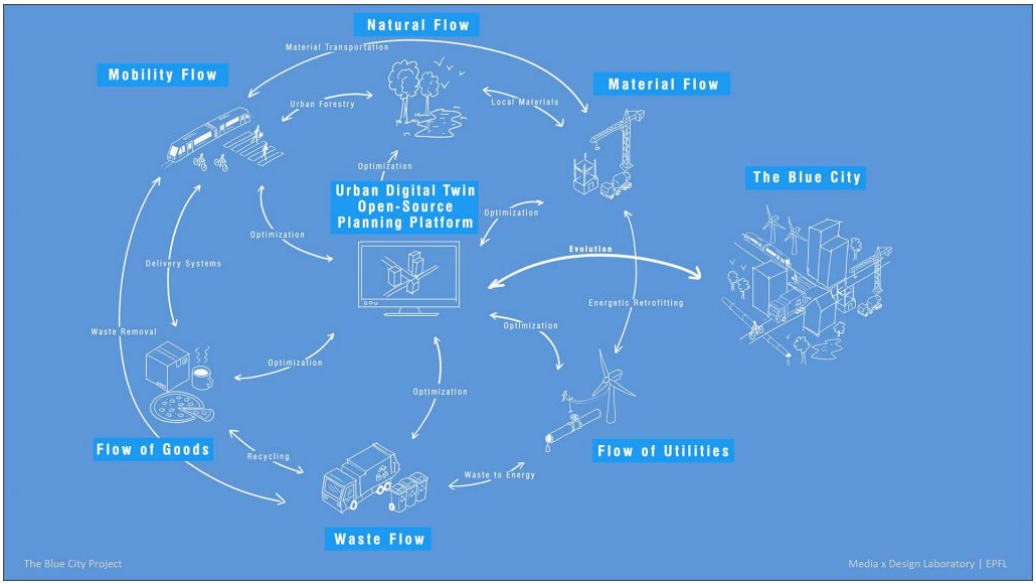

Figure 3: Example of city actors involved in Urban Digital Twin

2.2.1 Urban Digital Twin Challenges

Since an UDT is a DTN, the challenges seen in 2.1.1, that apply to DT, also apply the UDT.

challenges to UDT --> #TODO

2.3 Blockchain and Smart Contract

small intro to Smart contracts

In [7], the author mentions security of data in UDTs under the challenges section, implying that at the time of writing, no comprehensive solutions exist. As UDTs continue to evolve, and play increasingly important roles in the development of smart cities, addressing the challenges related to data management, access control and collaboration among various CA becomes crucial. One potential solution to these challenges is the use of Smart Contracts (SC). These decentralized, self-executing agreements can automate processes. They are implemented on top of Blockchain (BC) technology to promote transparency among the various CA. In the following section, we will explore the concept of SC and discuss how they can potentially be applied to benefit UDTs. Before going further into SC, we need to define BC.

2.3.1 Blockchain

what is blockchain ?

creation

Satoshi Nakamoto introduced the BC in 2008, its first application was in the world of finance, namely Bitcoin. He proposed to distribute electronic transactions rather than being dependant on a centralized entity. Since then the BC technology has been applied to other applications outside the financial world [16].

what is it?

A BC is a continuously-growing chain of blocks, where each block represents a single transaction, meaning that for each new transaction, a new block is added to the chain, hence the name. For a new block to be added into the chain, it needs to be validated by a consensus algorithm. The use of such consensus algorithms for new block validation, ensures that transactions are done without the participation of a Trusted Third Party (TTP), this reduces costs, and speeds up the overall transaction process [17]. Each block in the chain contains a cryptographic hash of the previous block, a timestamp of when it was created and any transaction information. Cryptographic hashes help ensure the integrity and immutability of the BC, figure 4 helps visualize this concept. The larger the chain is, the more protected it is [17, 18, 19]. BCs can be seen as a public ledgers where no transaction can be falsified.

how it ~works

how it achieves security

The consensus algorithm may change depending on which BC platform is being used, but the general idea is that each node that is participating in the validation of each new block, can vote on whether or not the new block is valid with its own view of the BC at a given time, the new block is only added if the majority of nodes accepts the new view. In [20] the author completes a review on the various existing consensus algorithms, some of the most used are Proof of Work (PoW) and Proof of Stake (PoS).

types of BC -- private vs public etc

There are three types of BCs, public, private and federated. Public BC allow any anonymous entity to join the BC and to become a node that has voting power. Whereas Private BC are controlled by a limited number of pre-authorized participants. Federated BC are a combination of the previous two, they assign a leader that verifies each transaction. The choice on which type to use is context dependant [21].

visual of BC

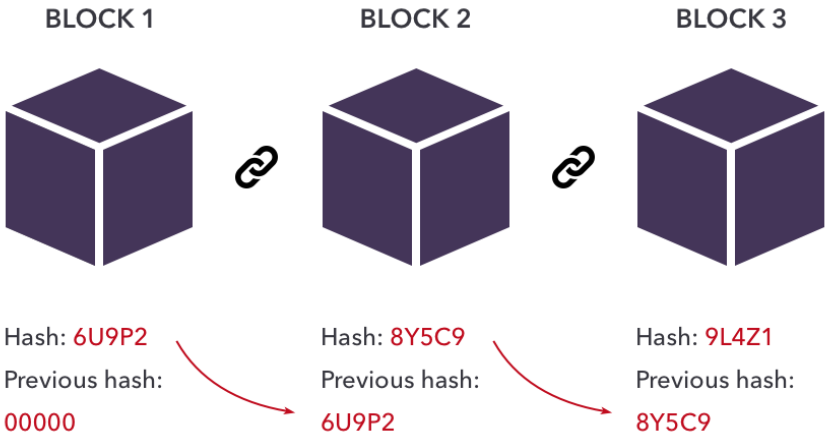

Figure 4: Visualization of simplified blockchain

2.3.2 Smart Contract

creation

Smart contracts were first introduced in 1997 by Szabo Nick. In his paper, he argues that the digital revolution has enabled the development of new institutions that can secure relationships in cyberspace. He describes them as a type of protocol that reduces transaction costs and facilitates all phases of the contract development and enforcement. They are capable of formalizing secure digital relationships, making them more functional than their paper-based counterparts [22].

BC technology enables the use of SC, the execution of each contractual transaction is recorded in the BC that, as we saw in the previous section, is immutable which will be a key factor in ensuring the integrity of SC transactions [17].

why BC enables SC

To better understand how SC function, we can examine the four stages of SC life cycle identified in [17]:

life cycle of SC

- Creation of SC: The creation process involves collaboration among various experts, and multiple steps. First, parties involved in the contract must agree to the terms, obligations and rights. Secondly an actual physical contract is drafted by actual lawyers. Afterwards, software developers must convert the agreement into a SC using programming languages, declarative languages and logic-based languages are preferred. Finally, the SC must be developed, which includes design, implementation and validation. This process can be quite lengthy and iterative, as multiple rounds of negotiation may occur.
- Deployment of SC: Once the SC is thoroughly validated, it can be deployed on top of a BC platform. After a contract is deployed on BC, it cannot be modified, any modifications will require a whole new SC. All parties have access to the SC through the BC. Moreover, digital assets of both parties are locked by freezing the corresponding digital wallets. Parties involved in the SC can be identified by their digital wallets.
- Execution of SC: After deployment, contractual clauses are monitored. Once all pre-conditions for execution are met, the contractual transactions are automatically executed. Each transaction that is executed is validated by the various nodes in the BC and will be committed to the BC as a new block.
- Completion of SC: Once execution of the contract is done, involved parties states are update in the BC. Digital assets are transferred and digital wallets are unfrozen.

These four stages complete the life cycle of SC, illustrating the process from creation to completion. It is worth noting that at all stages but the first, the SC writes to the BC. Challenges associated with each stage will be discussed in later sections.

SC have a broad field of applications, in [17] the author categorizes six different types of SC applications, we will go over them and defend the choices with various examples:

applications of SC / examples

- Internet of Things (IoT): Integration of IoT can support various applications such as supply chain management, inventory control systems, access control. For example, devices may obtain hashes from SC for swifter firmware update process, this will reduce the cost for maintaining central servers, in addition it enables automate Peer-to-Peer (P2P) business trading, which lowers TTP fees, and speed up the overall process [23].
- Distributed system security: Distributed Denial-of-Service (DDoS) attacks are often used to great success, these attacks try to overload systems to try and interrupt the services proposed by the system, in [24] a SC-based solution for DDoS attacks was proposed, the main idea was to use the BC to track attackers IP addresses. Cloud computing is a service where the provider sells access to computing and storage power, to verify the trustfulness of a Cloud Service Provider (CSP), Dong et al. [25] proposes a solution based on SC.
- Finance: SC can reduce financial risks, cut down on administration costs. In commercial and retail banking SC can bring benefits to the mortgage loan industry through automation of the mortgage process as shown in [26]. In the world of insurance, AXA has implemented solutions based on SC, their idea was to automate flight insurance reimbursements, when any passenger decides to buy a plane ticket with AXA insurance, a SC will automatically be signed and launched to the BC, this contract connects to the global air traffic database, if there is a flight delay of over two hours, the passenger will immediately be reimbursed via the SC [27], this improves efficiency, and reduces claim process costs.
- Data provenance: The ability of ensuring the authenticity of data is very important, some solutions propose to store meta-data records, current implementation from Trusted Platform Module assures data provenance at the cost of privacy [28]. SC might just be the solution to the problem of data provenance while preserving user privacy as proposed by Liang et al. in [29].
- Sharing economy: Represents the idea of reducing costs for consumers, by putting in place a platform for underutilized resource sharing such as cars, homes, objects. However, most sharing economy platforms are suffering from high transaction costs, privacy exposure and unreliable TTP. SC can revolutionize these platforms, by reducing transaction costs, enforcing agreed upon transactions, protecting user privacy and assuring trust between the parties without TTP, as shown by Bogner et al. in [30].
- Public sector: The use of BC can essentially prevent data fraudulence and provide transparency. Currently e-voting faces challenges in identity verification as well as preservation of voter anonymity. SC offers the solution for e-voting as shown by McCorry et al. in [31]. SC can also establish personal digital identity, users are able to protect their private information via SC that grant access permission to other users [32].

These examples showcase the potential of SC across various sectors. Their design and implementation are made possible by underlying platforms tailored to support their execution. In the following section we will go through the major platforms that enable SC, their characteristics, and their influence in the adoption of this technology.

2.3.3 Platforms

#TODO -- talk about the various platforms

2.3.4 Smart Contract in Urban Digital Twin

So far we've seen some possible applications of SC. In this section we will look at which of these applications are useful in the context of UDT.

#TODO -- examples of SC work in DT/UDT -- talk about where they are

Given the challenges seen in UDT we'll look at SC to solve some of them, namely access control, DDoS attacks .....

2.3.5 Smart Contract Challenges

#TODO challenges in each step of life cycle

2.4 Key exchange protocols

#TODO -- intro + challenges

3 Solution Model

4 Proof of Concept

4.1 Design

4.2 Implementation

5 Evaluation of Solution Model

6 Conclusion and Future Work

References

[1] Aidan Fuller, Zhong Fan, Charles Day, and Chris Barlow. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access*, 8:108952–108971, 2020. Conference Name: IEEE Access.

[2] Yiwen Wu, Ke Zhang, and Yan Zhang. Digital Twin Networks: A Survey. *IEEE Internet of Things Journal*, 8(18):13789–13804, September 2021. Conference Name: IEEE Internet of Things Journal.

[3] Fei Tao, Meng Zhang, Yushan Liu, and A. Y. C. Nee. Digital twin driven prognostics and health management for complex equipment. *CIRP Annals*, 67(1):169–172, January 2018.

[4] Gernot Steindl and Wolfgang Kastner. Semantic Microservice Framework for Digital Twins. *Applied Sciences*, 11(12):5633, January 2021. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

[5] Roberto Minerva. Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models.

[6] Bruno Cruz and Murillo Dias. CRASHED BOEING 737-MAX: FATALITIES OR MALPRACTICE? 8:2615–2624, January 2020.

[7] Ehab Shahat, Chang T. Hyun, and Chunho Yeom. City Digital Twin Potentials: A Review and Research Agenda. *Sustainability*, 13(6):3386, January 2021. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[8] Dessislava Petrova-Antonova and Sylvia Ilieva. Digital Twin Modeling of Smart Cities. In Tareq Ahram, Redha Taiar, Karine Langlois, and Arnaud Choplin, editors, *Human Interaction, Emerging Technologies and Future Applications III*, Advances in Intelligent Systems and Computing, pages 384–390, Cham, 2021. Springer International Publishing.

[9] Fabian Dembski, Uwe Wössner, Mike Letzgus, Michael Ruddat, and Claudia Yamu. Urban Digital Twins for Smart Cities and Citizens: The Case Study of Herrenberg, Germany. *Sustainability*, 12(6):2307, January 2020. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

[10] Božidar Radenković, Marijana Despotović-Zrakić, Zorica Bogdanović, Dušan Barać, Aleksandra Labus, and Tamara Naumović. A distributed IoT system for modelling dynamics in smart environments. In *2020 International Conference Engineering Technologies and Computer Science (EnT)*, pages 47–53, June 2020.

[11] Abigail Francisco, Neda Mohammadi, and John E. Taylor. Smart City Digital Twin–Enabled Energy Management: Toward Real-Time Urban Building Energy Benchmarking. *Journal of Management in Engineering*, 36(2):04019045, March 2020. Publisher: American Society of Civil Engineers.

[12] T. Nochta, L. Wan, J. M. Schooling, and A. K. Parlikad. A Socio-Technical Perspective on Urban Analytics: The Case of City-Scale Digital Twins. *Journal of Urban Technology*, 28(1-2):263–287, April 2021. Publisher: Routledge _eprint: https://doi.org/10.1080/10630732.2020.1798177.

[13] S. Q. Dou, H. H. Zhang, Y. Q. Zhao, A. M. Wang, Y. T. Xiong, and J. M. Zuo. RESEARCH ON CONSTRUCTION OF SPATIO-TEMPORAL DATA VISUALIZATION PLATFORM FOR GIS AND BIM FUSION. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-3/W10:555–563, February 2020.

[14] L. Wan, T. Nochta, and J. M. Schooling. Developing a City-Level Digital Twin ?Propositions and a Case Study. In *International Conference on Smart Infrastructure and Construction 2019 (ICSIC)*, Cambridge Centre for Smart Infrastructure & Construction, pages 187–194. ICE Publishing, January 2019.

[15] Siavash H. Khajavi, Naser Hossein Motlagh, Alireza Jaribion, Liss C. Werner, and Jan Holmström. Digital Twin: Vision, Benefits, Boundaries, and Creation for Buildings. *IEEE Access*, 7:147406–147419, 2019. Conference Name: IEEE Access.

[16] Tareq Ahram, Arman Sargolzaei, Saman Sargolzaei, Jeff Daniels, and Ben Amaba. Blockchain technology innovations. In *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*, pages 137–141, June 2017.

[17] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Weili Chen, Xiangping Chen, Jian Weng, and Muhammad Imran. An overview on smart contracts: Challenges, advances and platforms. *Future Generation Computer Systems*, 105:475–491, April 2020.

[18] Akanksha Kaushik, Archana Choudhary, Chinmay Ektare, Deepti Thomas, and Syed Akram. Blockchain — Literature survey. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 2145–2148, May 2017.

[19] Yuanyu Zhang, Shoji Kasahara, Yulong Shen, Xiaohong Jiang, and Jianxiong Wan. Smart Contract-Based Access Control for the Internet of Things. *IEEE Internet of Things Journal*, 6(2):1594–1605, April 2019. Conference Name: IEEE Internet of Things Journal.

[20] Du Mingxiao, Ma Xiaofeng, Zhang Zhe, Wang Xiangwei, and Chen Qijun. A review on consensus algorithm of blockchain. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2567–2572, October 2017.

[21] Rebecca Yang, Ron Wakefield, Sainan Lyu, Sajani Jayasuriya, Fengling Han, Xun Yi, Xuechao Yang, Gayashan Amarasinghe, and Shiping Chen. Public and private blockchain in construction business process and information integration. *Automation in Construction*, 118:103276, October 2020.

[22] Nick Szabo. Formalizing and Securing Relationships on Public Networks. *First Monday*, September 1997.

[23] Konstantinos Christidis and Michael Devetsikiotis. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access*, 4:2292–2303, 2016. Conference Name: IEEE Access.

[24] Bruno Rodrigues, Thomas Bocek, Andri Lareida, David Hausheer, Sina Rafati, and Burkhard Stiller. A Blockchain-Based Architecture for Collaborative DDoS Mitigation with Smart Contracts. In Daphne Tuncer, Robert Koch, Rémi Badonnel, and Burkhard Stiller, editors, *Security of Networks and Services in an All-Connected World*, volume 10356, pages 16–29. Springer International Publishing, Cham, 2017. Series Title: Lecture Notes in Computer Science.

[25] Changyu Dong, Yilei Wang, Amjad Aldweesh, Patrick McCorry, and Aad van Moorsel. Betrayal, Distrust, and Rationality: Smart Counter-Collusion Contracts for Verifiable Cloud Computing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 211–227, New York, NY, USA, October 2017. Association for Computing Machinery.

[26] Ye Guo and Chen Liang. Blockchain application and outlook in the banking industry. *Financial Innovation*, 2(1):24, December 2016.

[27] Remy Remigius Zgraggen. Cyber Security Supervision in the Insurance Sector: Smart Contracts and Chosen Issues. In *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, pages 1–4, June 2019.

[28] Mohammad M. Bany Taha, Sivadon Chaisiri, and Ryan K. L. Ko. Trusted Tamper-Evident Data Provenance. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 646–653, August 2015.

[29] Xueping Liang, Sachin Shetty, Deepak Tosh, Charles Kamhoua, Kevin Kwiat, and Laurent Njilla. ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 468–477, May 2017.

[30] Andreas Bogner, Mathieu Chanson, and Arne Meeuw. A Decentralised Sharing App running a Smart Contract on the Ethereum Blockchain. In *Proceedings of the 6th International Conference on the Internet of Things*, IoT'16, pages 177–178, New York, NY, USA, November 2016. Association for Computing Machinery.

[31] Patrick McCorry, Siamak F. Shahandashti, and Feng Hao. A Smart Contract for Boardroom Voting with Maximum Voter Privacy. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 357–375, Cham, 2017. Springer International Publishing.

[32] Affan Yasin and Lin Liu. An Online Identity and Smart Contract Management System. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 192–198, June 2016. ISSN: 0730-3157.

[33] Marietheres Dietz, Benedikt Putz, and Günther Pernul. A Distributed Ledger Approach to Digital Twin Secure Data Sharing. In Simon N. Foley, editor, *Data and Applications Security and Privacy XXXIII*, Lecture Notes in Computer Science, pages 281–300, Cham, 2019. Springer International Publishing.

[34] Akanksha Saini, Qingyi Zhu, Navneet Singh, Yong Xiang, Longxiang Gao, and Yushu Zhang. A Smart-Contract-Based Access Control Framework for Cloud Smart Healthcare System. *IEEE Internet of Things Journal*, 8(7):5914–5925, April 2021. Conference Name: IEEE Internet of Things Journal.

[35] Data Mesh Principles and Logical Architecture.

[36] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Provably secure authenticated group Diffie-Hellman key exchange. *ACM Transactions on Information and System Security*, 10(3):10, July 2007.

[37] Yong Wang, Byrav Ramamurthy, and Xukai Zou. The Performance of Elliptic Curve Based Group Diffie-Hellman Protocols for Secure Group Communication over Ad Hoc Networks. In *2006 IEEE International Conference on Communications*, volume 5, pages 2243–2248, June 2006. ISSN: 1938-1883.

[38] Sihan Huang, Guoxin Wang, Yan Yan, and Xiongbing Fang. Blockchain-based data management for digital twin of product. *Journal of Manufacturing Systems*, 54:361–371, January 2020.

[39] Yongdae Kim, Adrian Perrig, and Gene Tsudik. Tree-based group key agreement. *ACM Transactions on Information and System Security*, 7(1):60–96, February 2004.

[40] Jonathan Katz and Moti Yung. Scalable Protocols for Authenticated Group Key Exchange. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, Lecture Notes in Computer Science, pages 110–125, Berlin, Heidelberg, 2003. Springer.

[41] Emmanuel Bresson, Olivier Chevassut, and David Pointcheval. Provably Authenticated Group Diffie-Hellman Key Exchange — The Dynamic Case. In Colin Boyd, editor, *Advances in Cryptology — ASIACRYPT 2001*, Lecture Notes in Computer Science, pages 290–309, Berlin, Heidelberg, 2001. Springer.

[42] Jae Cha Choon and Jung Hee Cheon. An Identity-Based Signature from Gap Diffie-Hellman Groups. In Yvo G. Desmedt, editor, *Public Key Cryptography — PKC 2003*, Lecture Notes in Computer Science, pages 18–30, Berlin, Heidelberg, 2002. Springer.

[43] Massimo Bartoletti and Livio Pompianu. An Empirical Analysis of Smart Contracts: Platforms, Applications, and Design Patterns. In Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y.A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, and Markus Jakobsson, editors, *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 494–509, Cham, 2017. Springer International Publishing.