PREPRINT

U-spaceChain: A Decentralized Approach to Unmanned Traffic Management Services Provision

Balita Rakotonarivo
ENAC-Airbus-Sopra Steria, Drones and UTM Research Chair
ENAC, University of Toulouse
Toulouse, France
0000-0001-9362-0468

Murat Bronz
Dynamic Systems, Optim Team, ENAC Lab.
ENAC, University of Toulouse
Toulouse, France
0000-0002-1098-5240

***Abstract*—Thousands of drone operations are expected over urban areas in different countries, as drones bring economic and societal benefits. However, it raises problems regarding unmanned traffic management, including safety, regulatory, security, automatization, communication, and scalability aspects. How can we provide decentralized services for drone traffic management that are compliant with the regulatory framework? We identified some requirements from the European U-space regulation and applied the Systems Development Methodology. We devised a permissioned blockchain-based architecture, Services as smart contracts, and a simple consensus mechanism named Appointed-by-Authority as conceptual contributions. The practical one is a working software named U-spaceChain. It is a proof-of-concept of a blockchain in which smart contracts provide Flight Authorization, Network Identification, Traffic Information, Conformance Monitoring, Geo Awareness, Weather Information, and Common Information services. It is compliant with regulations, secure by design, can support the development of new services, and the concepts can be reused in other domains. We also designed enablers that allow testing this proof-of-concept by creation and then simulation of drone traffic. U-spaceChain may be a reference for the research and application of blockchain to the drone traffic management domain.**

***Index Terms*—Blockchain, UTM, U-space, Drone, UAS, ATM, Air Traffic Management**

I. Introduction

Drones possess several advantages over traditional aircraft. Indeed, they can perform Dull, Dirty, or Dangerous missions, and as electric vehicles, they are environmentally friendly. They could create more jobs but come with challenges: safety, security, social acceptance, and traffic management - also called Unmanned Traffic Management (UTM). Worldwide, drone traffic is forecast to increase by such a large volume that it will outstrip the current traffic volume of manned traffic [1]. On the other hand, traditional Air Traffic Management (ATM) systems already show their limit, with problems of capacity and scalability, especially in Europe [2]. Researchers believe that cross-fertilization between ATM and UTM is one key to solving problems from each domain [3], and they have started initiatives to adapt and digitalize information management in aviation to face those challenges.

Blockchains suit aviation because of their security features and decentralized nature [3], [4]. Among existing architectures for UTM services, decentralized architectures improve resilience against malicious attacks and reduce the risk of single points of failure. Nevertheless, security issues may arise, and blockchain-based UTM systems may be one way of resolving them [5]. However, the drone ecosystem is also highly regulated; in contrast to casually connected objects, safety is paramount. Whereas Blockchain is a promising solution for UTM from a security point of view, it also needs to conform to the regulatory framework and its required services to ensure the safety of operations.

Given these facts, the research questions are: Can a practical blockchain-based system provide regulation-compliant services to support UTM? If so, how?

This paper covers UTM in Europe, with a concept called U-space, for which the regulation came into force in 2023 [6]. We chose this regulatory framework because we needed to reduce and control the scope of application to achieve concrete implementation. U-space is a set of new services and specific procedures designed to support safe, efficient, and secure access to airspace for large numbers of drones. These services rely on a high level of digitalization and automation of functions, whether on the drones or the ground. Among U-space key principles are minimizing deployment and operating costs, providing scalable and flexible systems, and adopting technologies and standards from other sectors [7]. These are in line with what blockchain technology can bring. For this paper, we focus on the architecture and concepts used to deliver the services, leaving the issues of confidentiality, trust, and performance metrics aside.

The rest of this paper is organized as follows: In Section II, the Related work is exposed. The Method is presented in Section III. The Results are described in Section IV. In Section V, the validation of the Proof-of-Concept is explained. The Discussion happens in Section VI before the Conclusion in Section VII.

II. Related work

Blockchains are potential solutions to the problems in the broader aviation domain. Still, many design options related to the type and configuration of Blockchain need to be carefully considered and compared [3].

Some blockchain-related work for drone operations has already been published. Still, they focused on one kind of use case, such as decentralized independent multi-drones to transport medical goods collaboratively [8] or environmental exploration with swarms [9]. It is desirable to be agnostic to the type of operations and compliant with regulation without implementing incentives like relying on cryptocurrencies, as these are unrealistic for sustainable ATM operations.

Security issues and requirements of UTM components, with an implementation based on Hyperledger Fabric, are presented by Allouch *et al.* [10]. Hamissi *et al.* identifies the security challenges of UTM systems and demonstrates how a blockchain system can solve them [5]. Yaguchi and Wakazono present a distributed autonomous flight plan management [11]: their approach is applied to blockchain technology for reservation systems, octree for airspace indexing, and probabilistic road maps with an A* algorithm for the replanning system.

To summarize, only a restricted amount of research has delved into developing UTM systems that integrate blockchain technology. Blockchain is predominantly utilized to bolster the security aspects of unmanned aircraft identification procedures and flight plan management. Moreover, no work was done to provide more advanced services within a defined regulatory framework in which each drone of the traffic has its mission.

III. Method

The Systems Development Methodology [12] is applied. It is rigorous and relevant, as we aim to develop not only a system that could serve as a proof-of-concept for fundamental research but also provide an artifact that becomes the focus of further research [13]. Figure 1 describes the methodology.

For the research described in this paper, we performed the following steps:

- identification of operational requirements based on European regulations and best practices in blockchain and software engineering,
- proposition of the general architecture and concepts to satisfy the chosen requirements. We mapped the architecture with the system actors to facilitate information exchange, based on the Conway-inspired "mirroring hypothesis" [13], and by leveraging the smart contracts to provide the services as we propose a classification of the possible services and deal with their inheritance, interfaces, and execution concerning required and later expected services,
- implementation of the system, not only the services but also some enablers to allow the testing and validation of the system,
- evaluation through simulation as it can achieve the validation of an approach and theoretical contributions [14].

As a proof-of-concept, the scope here does not include non-functional aspects, which we will study in future iterations of the work. Prescriptions in the European Implementing Regulation 2021/664 [6] are the source of the functional requirements for U-spaceChain. However, only a subset of these requirements are retained as hypotheses for the proof-of-concept. The following is a list of high-level functional requirements for the system:

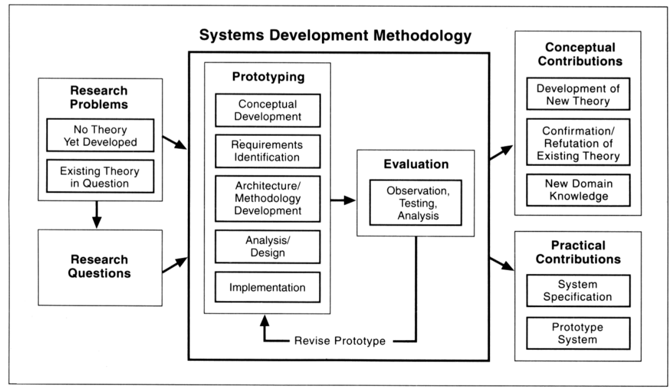

Fig. 1. The Systems Development Methodology, from Nunamaker, Chen, and Purdin [12]

- RQR1: The system shall provide static and dynamic data dissemination for the management of traffic of unmanned aircraft.
- RQR2: The system shall grant access to common information services to relevant authorities, air traffic service providers, U-space service providers, and UAS operators on a non-discriminatory basis, including with the same data quality, latency, and protection levels.
- RQR3: The system shall allow adequate records storage and reliable traceability of the activities of the U-space services providers (USSP) and Common Information service providers (CISP).
- RQR4: The system shall maintain data integrity.
- RQR5: The system shall provide Common Information Services (CI), Geo-awareness service (GO), Flight Authorization Service (FA), Network Identification Service (NI), and Traffic Information Service (TI).
- RQR6: Optionally, the system shall provide Weather Information Service (WE) and Conformance Monitoring Service (CM).
- RQR7: The system shall allow the NI to be accessed by the public for deemed public information, the other USSP, the air traffic services providers (ANSP) concerned, the single CI provider, and the relevant competent authorities.

IV. Results

A. A regulatory-compliant blockchain-based architecture

Literature states several possible system architectures for UTM, Figure 2. Centralized architectures include 1) a single entity providing services for all vehicles in the airspace or 2) a hub-and-spoke architecture in which multiple entities exist, but each vehicle gets services from a single entity. Decentralized architectures include 1) the federated model, where multiple providers exist for most services, where each aircraft can choose between entities; 2) the closest-peer model, where no entity provides services but aircraft communicate locally with their closest neighbors; 3) the distributed model, where vehicles communicate globally and directly relative to flight plan [15]; and 4) the blockchain-based architecture [10]. These

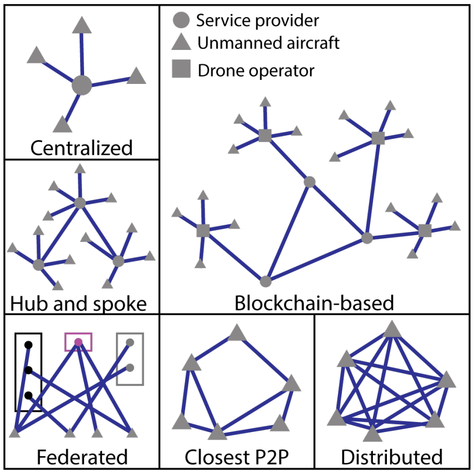

Fig. 2. Different architectures compared to the blockchain-based

cases do not consider the roles of drone operators and U-space service providers and isolate drone operations from manned traffic, thus conflicting with RQR2.

One basic architecture principle is that *Form follows function* [16]. A better model should consider the roles of the relevant authorities, air traffic service providers, U-space service providers (USSP), UAS operators, and the general public (RQR2, RQR7). By definition, USSP provides the services. However, they need to share information between them (RQR2), as several USSPs can manage U-space airspace, and data dissemination is mandatory (RQR1). Thus, they become a peer-to-peer network; only operators will connect to them to send information. The operators are responsible for the unmanned aircraft, so these latter do not require direct communication with the USSP. Next, RQR3 requires traceability of all the activities, so immutable and shared data storage can be appropriate. Some information from the rest of the world, such as Air Traffic data and operational information from other trusted information providers, must also be integrated [6]. They are injected into the shared data storage of the networks of peers or USSP, contributing to RQR4. All these elements can be associated with the components of a blockchain system. The proposed conceptual architecture, identified as *blockchain-based*, is shown in Figure 2.

There are different roles for different entities participating in the system. The USSP is a blockchain node that exposes an API to get and send data. The whole blockchain and the nodes constitute the *U-spaceChain*, Figure 3. The operator has an app to connect to the blockchain through a USSP, Figure 4. Likewise, users like the public and authority can connect to U-spaceChain through a USSP, Figure 5. The use case for each participant is modeled in Figure 6. In the final phase of U-space, the USSP actor and U-spaceChain Node in Figure 6 will be merged.

B. Smart contracts to provide U-space services

Smart contracts are computer scripts that allow autonomous computation guided by rules and policies while automating

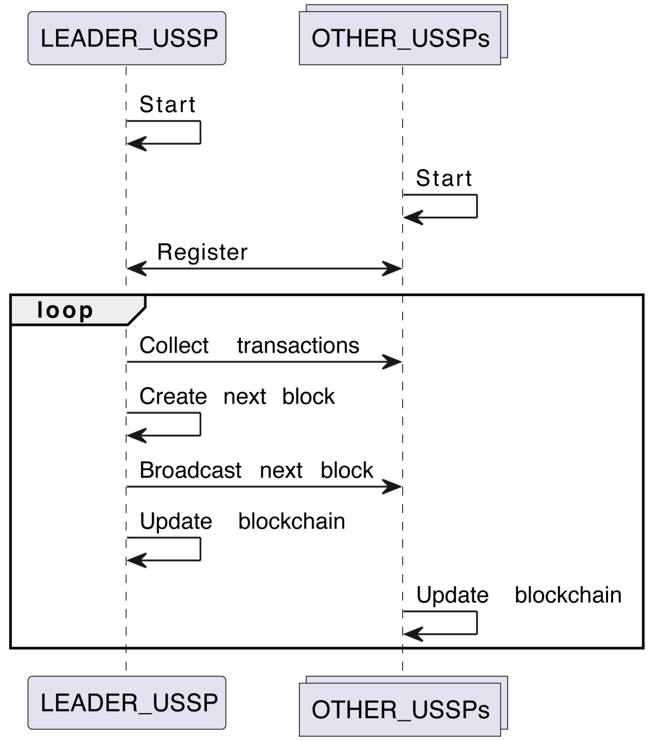

Fig. 3. The process between USSPs

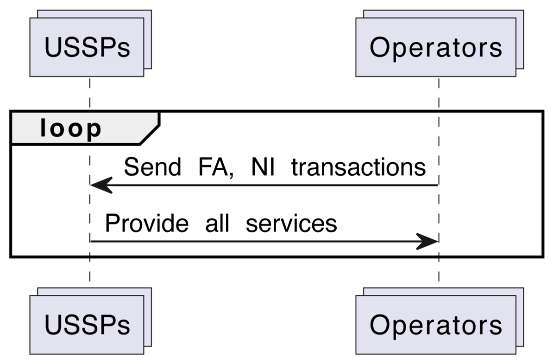

Fig. 4. The process between USSPs and operators

previously manual business logic. Some benefits of smart contracts over standard contracts are risk reduction, reduced service and administration costs, and enhanced business process efficiency [17]. We describe smart contract models for delivering the services required by the EU drone regulation in a way consistent with the ultimate goal of U-space, which is highly automated service delivery [7].

1) Structure of smart contracts for providing services: The regulation requires USSP to provide four services: GO, FA, NI, and TI (RQR5) and suggests optional services: CM and WE (RQR6) [6]. Figure 7 summarizes the different types of smart contracts used to provide the services.

Geo Awareness Service (GO) should provide operators with information about the latest airspace and defined UAS geographical zone information constraints [6]. Here, the information is updated according to events and does not concern an

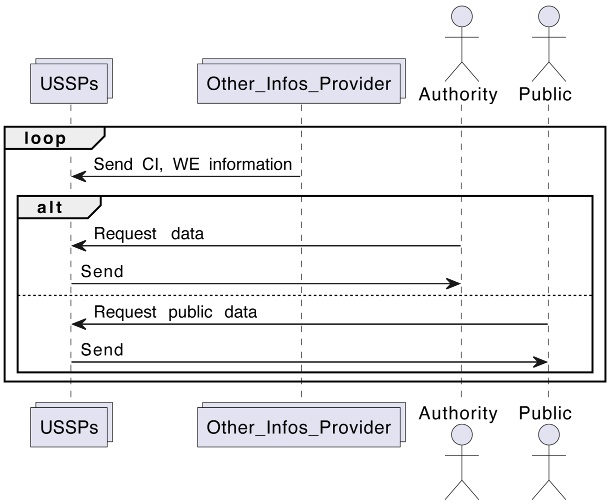

Fig. 5. The process between USSPs and other stakeholders

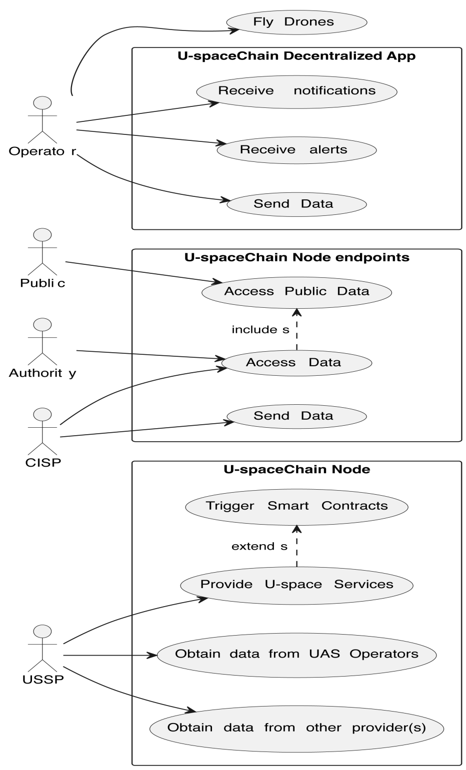

Fig. 6. Use cases and roles for each participant in U-spaceChain

identified flight. This case is the most straightforward service, so we identify it as a *static* service, Figure 7(a).

Common Information Service (CI) is a separate service that could also be provided by the same entity as a USSP. CI injects operational information from the outside world, like positions of close manned aircraft, and Weather Information Service (WE) injects weather operational messages like airports' METARs [6]. They share the same core principle: information from an oracle is updated periodically, and the blockchain must be updated. We identify it as an *imported* service, Figure 7(b).

U-space service providers deploy a Flight Authorization Service (FA) to grant individual flight authorization to UAS operators [6]. FA concerns each identified flight and can consist of several steps: either one request when the current blockchain data is enough to immediately produce the answer or forwarded to another entity for further coordination. Several loops between the operator and the safety officers take place if further coordination is required [18]. Sometimes, the request is waiting for answers from the safety officers, making it asynchronous. Moreover, iterations on the exact flight authorization may happen until the flight is accepted. At that final stage, the information regarding this operation will not be updated anymore. We identify it as an *asynchronous* service, Figure 7(c).

Network Identification Service (NI) shall allow the continuous processing of the remote identification of the drone throughout the flight and shall provide it to the authorized users [6]. NI is about data for each identified and authorized flight. Each transaction is atomic, having no relationship with the previous or the following ones for each flight. However, they are also repetitive, as the data from NI should be sent periodically as long as the flight plan is active. We identify it as a *repetitive* service, Figure 7(d).

Traffic Information Service (TI) shall contain information on any other conspicuous air traffic near the position or intended route of the UAS flight [6]. A Conformance Monitoring Service (CM) shall enable the UAS operators to verify whether they comply with the airspace requirements and the terms of their flight authorization [6]. The TI and the CM are sent to each flying aircraft, resembling NI. These services are not only repetitive but also dynamic. They are executed as long as the flight plan is activated, using relevant data in the blockchain to feed their algorithms and achieve the desired outputs. We identify them as *dynamic* services, Figure 7(e).

2) Executing the smart contracts: The smart contracts share the same payload structure, inheriting from the same abstract class, where a payload format is defined as an interface. They also execute a unique method each time they are invoked. Consequently, as long as the payload interface is respected, the execution of smart contracts should be automated and flawless. Figure 8 shows the relationships and structure between the desired services and smart contracts modeled in a UML Class Diagram.

C. The Appointed-by-Authority Consensus Mechanism

The context is a permissioned blockchain, where only authorized nodes are allowed. This approach abides by the regulation as the Authority must supervise the airspace and is responsible for its safety. Moreover, the assumption is that no participant will purposely send corrupted data, as each participant must go through a certification process before being accepted as USSP or CISP [6]. Thus, the Authority decides which USSP will create the next chain block, and the others must adhere to her decision. In this case, there is no need for competition, as there are no monetary incentives,

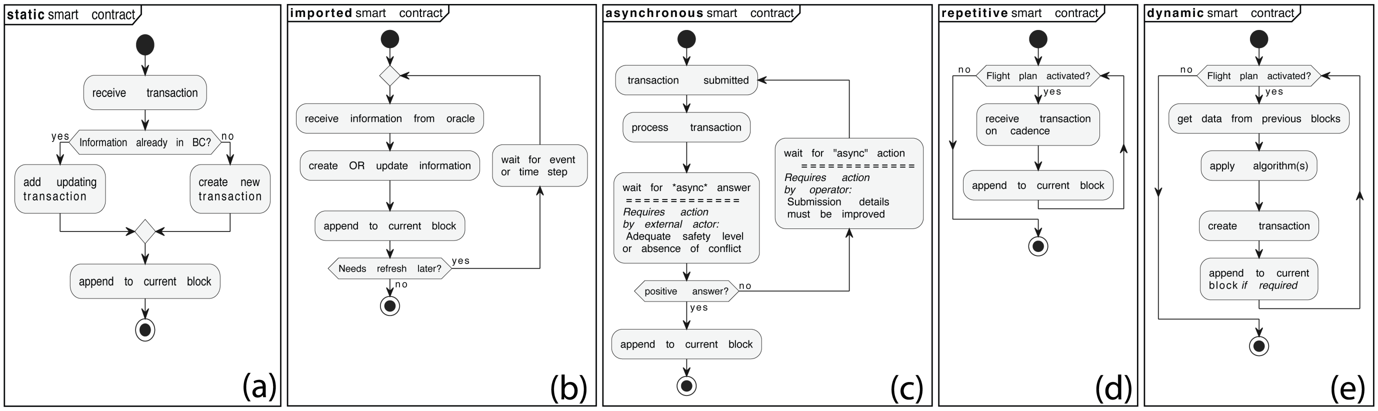

Fig. 7. The smart contracts classification for services provision, modeled in UML's Activity diagrams.

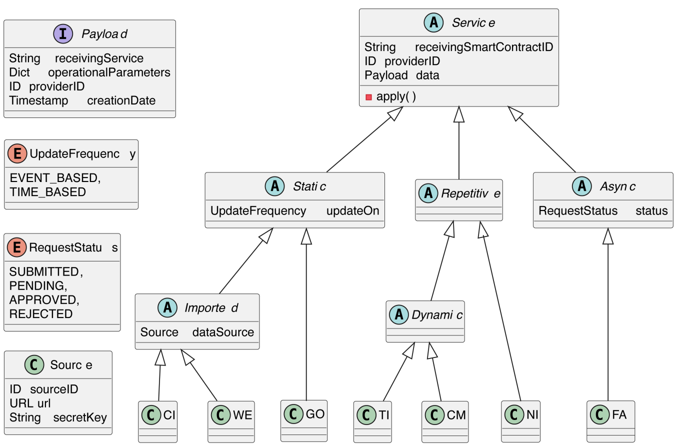

Fig. 8. Class diagram of the Services as Smart contracts

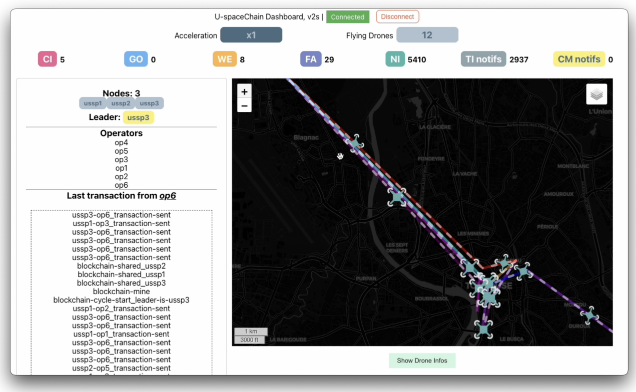

Fig. 9. The dashboard of U-spaceChain

and the stakes are the same for every participant. Safety and operational efficiency take priority.

We propose a consensus mechanism, identified here as *Appointed-by-Authority* (*AbA*), in which the Authority publishes a list of nodes that can create blocks. The attribution of the block creator's role, or *leader*, is at the Authority's discretion. She can update and publish that list anytime on the blockchain, whereas the Genesis block already contains an initial list. This leader node will retrieve the transactions from each registered node before creating the next block.

D. U-spaceChain enablers

U-spaceChain requires data to verify the system's functionality, with continuous and timely transactions feed. Thus, we have developed *enabler scripts* to support the system's testing by simulation.

The first enabler is a *trajectory generator* tool, which takes start and stop waypoints to generate a straight line with parameters such as minimum distance as well as minimum and maximum flight time. The start and stop waypoints are chosen in a library of known points of interest.

The second enabler is a *dispatcher* whose role is to randomly spread that traffic during a defined duration and generate all needed transactions for each service, with the correct time step and the node that will receive it. That node is also chosen randomly and never changes for a flight. Node numbers and identifiers are parameters of this enabler. In addition, one parameter allows allocating more or less traffic for each node. Moreover, it also creates the FA requests transactions. As *static* and *imported* services are less frequently solicited than repetitive services, one can manually add them to the final list of transactions.

The third enabler is an *orchestrator*, which reads all the transactions and, for each time step, then sends the correct transaction to its specified node. The acceleration factor is a parameter of this enabler, allowing the time to be multiplied in the simulation so it can be accelerated. This enabler is the entry point to the whole system, as it can start the nodes of U-spaceChain, register them, and then open other scripts to send the transactions steadily at each timestep.

The fourth enabler is a *dashboard*, which is a visual approach to understanding the state of U-spaceChain. It displays the number of transactions for each service, the current leader node, and the state of the blockchain, such as synchronized or idle. It also features a live map showing operations in real-time. Figure 9 is a screenshot of the dashboard.

TABLE I
SIMULATION RESULTS: THREE NODES, ONE HOUR OF TRAFFIC

Run Index	Node	Duration	CI	FA	GO	NI	WE	Total Trans-actions	TI exe-cuted	Total TI exe-cuted	TI notifi-cations sent	CM exe-cuted	Total CM exe-cuted	CM notifi-cations sent
Expected		3600	**5**	**30**	**4**	**8178**	**12**	**8229**		**8178**	**above 0**		**8178**	**10**
	Ussp1		5	33	4	8191	11		2011		2131	2011		10
1	Ussp2	3600	5	33	4	8191	11	8244	2580	8178	2867	2580	8178	0
	Ussp3		5	33	4	8191	11		3587		6830	3587		0
	Ussp1		5	30	4	8178	13		2011		2116	2011		10
2	Ussp2	3600	5	30	4	8178	13	8230	2580	8178	2857	2580	8178	0
	Ussp3		5	30	4	8178	13		3587		6830	3587		0
	Ussp1		5	29	4	8181	9		2011		2135	2011		10
3	Ussp2	3600	5	29	4	8181	9	8228	2580	8178	2858	2580	8178	0
	Ussp3		5	29	4	8181	9		3587		6827	3587		0
	Ussp1		5	30	4	8178	0		2011		2121	2011		10
4	Ussp2	3600	5	30	4	8178	0	8217	2580	8178	2853	2580	8178	0
	Ussp3		5	30	4	8178	0		3587		6828	3587		0
	Ussp1		5	30	4	8178	0		2011		2125	2011		10
5	Ussp2	3600	5	30	4	8178	0	8217	2580	8178	2850	2580	8178	0
	Ussp3		5	30	4	8178	0		3587		6832	3587		0
	Ussp1		5	30	4	8187	12		2011		2125	2011		10
6	Ussp2	3600	5	30	4	8187	12	8238	2580	8178	2866	2580	8178	0
	Ussp3		5	30	4	8187	12		3587		6815	3587		0

V. PROOF-OF-CONCEPT VALIDATION

Working software that behaves as expected for a determined scenario can validate the proof-of-concept [14]. With the explicit purpose of validating the proposed concepts, we built a prototype that effectively implements them. The simulation results were coherent with those expected, confirming the concepts' validity, Table I. This paper aims to prove their feasibility and provide a basis for further research.

The system was implemented from scratch in Javascript and locally tested on an M1 Mac Book Pro with 16 GB of RAM and MacOS Sonoma. It has three nodes, representing three USSP connected to six operators. We set the blockchain time step for synchronization to four seconds, meaning the system will add a new block every four seconds. We chose this value for the scenarios described below because we implemented a simple TI algorithm which verifies within a 500-meter radius. A speed of about 25m/s for a small drone is considered high. This speed allows 20 cycles of U-spaceChain synchronization, corresponding to 80 seconds before a collision occurs. This experimental value should be enough to avoid conflicts with actual flights, as this prototype does not focus on finding the optimal values.

We ran one scenario multiple times. The Table I shows the results of six runs. The scenario involved one hour of traffic, during which 30 drone flights were created and performed. Each drone's minimum flight time was 5 minutes, and the minimum distance flown was 900m. To test for CM, ten of the NI transactions of three of the flights were modified to trigger the CM notifications.

The counts of CI, GO, TI, and CM execution correspond precisely to the expected values. Only the flights from USSP2 were tampered with to force CM notifications. The FA, NI, and WE were mainly coherent with their expected value, with some variation. For two simulation runs, WE had 0 added transactions, as the nodes were intentionally disconnected from the internet, prohibiting the insertion of weather data into the blockchain. However, the number of invocations of *repetitive* and *dynamic* services could sometimes have been more accurate due to the implementation of *orchestrator*, which can still be improved. Screen recordings of the dashboard displaying this scenario are available online[1].

Algorithm 1 *Simple Traffic Information Algorithm*

Require: Drone A with latitude and longitude $(\text{lat}_A, \text{long}_A)$ and radius R
Set $B \leftarrow \emptyset$
for all other drones D **do**
  Calculate distance d between A and D using lat_A, long_A, and respective coordinates of D
  if $d \leq R$ **then**
    Add D to set B
  end if
end for
if $B \neq \emptyset$ **then**
  Send information regarding all B elements to drone A
end if

[1] https://shorturl.at/aiqBM

VI. Discussion

A. Security

Blockchain is a promising approach to providing UTM services [5], [11]. If designed and implemented correctly, it provides a robust solution for handling Confidentiality, Integrity, and Availability, thereby ensuring the security of UTM services [5]. Synchronization is also one of its concepts. Indeed, RQR1, RQR3, and RQR7 are about data access and dissemination, either static or dynamic, which relates to Availability. Blockchain has built-in Integrity because of its immutability, which fosters Traceability, which aligns with RQR4. However, this first iteration used no specific performance metrics.

On the other hand, safety and business-wise, complete access to the blockchain in reading and even more writing should be limited for Confidentiality. The requirements suggest more than a public blockchain. Therefore, the permissioned blockchain, which allows only selected participants to send data, emerges as the most viable option for UTM services. Moreover, the best practices and regulation mandate the drones, drone operators, and USSPs to be registered and trustable, according to the International Aviation Trust Framework [19]. In the future evolution of U-spaceChain, smart contracts can encrypt data before it is written to blocks to guarantee Confidentiality. As RQR7 expects access by different stakeholders, external consultations should be possible, using different endpoints to segregate the information viewers and close up the sensible data. Each U-spaceChain node exposes API endpoints and webpages from which external consultation is possible.

B. Extensibility and generalization

These properties are desirable for U-spaceChain to be helpful in its domain and reusable elsewhere.

Extensibility: The Services as Smart Contracts structure is designed to lower the difficulties of creating new services. The diverse types of smart contracts to inherit allow us to handle most of the cases. To illustrate this, we take three examples from the *U-space Concept of Operations*, a European project studying the concept of operations of U-space or CORUS[2], which is not part of regulation but serves as an inspiration for it and proposes more than 20 drone services across eight topics. Each proposed service can be matched to one of the Services as Smart Contracts abstract class. Example 1: the *Population density* map service should collect and present the relevant map for the operation from external providers, corresponding to the abstract *imported* smart contract. Example 2: the *Tactical conflict prediction*, an advanced service, requires that the positions of all aircraft be known and frequently updated in an airspace volume to try to predict whether two aircraft come too close together, corresponding to the abstract *dynamic* smart contract. Example 3: the *Collaborative interface with ATC*, an advanced service, is expected to be used while a drone is close to or in a controlled area. Its interface allows flights to receive instructions and clearances in a standard and efficient manner at the discretion of air traffic controllers, corresponding to the abstract *async* smart contract.

Generalization: Every domain where a permissioned blockchain is adequate could be a potential application. All domains where an authority wants to let the market thrive while maintaining light control may use the *AbA* consensus mechanism. Moreover, this is not limited to large-scale operations, such as forecasting urban area traffic management, but also to smaller initiatives. One example is retail and e-commerce, where the system can monitor stock levels. Most of the time, it would automatically proceed to trigger replenishment orders for some items - the *static* smart contract. But for other items, it can request or wait for manual inputs or validation - the *async* smart contract.

C. Limits and Future directions

We have presented a promising architecture as a prototype. However, further shortcomings have to be addressed.

a) Testbed for drone traffic management services: Research shows that many drone operations are expected in the future, especially in urban areas. For example, the traffic density within the urban airspace of Paris only is expected to be 174,521 drones per hour by 2035 [1]. Testbeds and simulation frameworks are required to design, implement, evaluate, and operate systems supporting such loads. This work contributes by providing another approach compared to the centralized, hub-and-spoke, distributed, federated, and closest peer-to-peer, as well as software from which other research can start.

b) Services as Smart Contracts: For service development, U-spaceChain provides essential building blocks that can be customized, opening the way to creating more advanced services, like Tactical Deconfliction or Emergency alerts, and evaluating the performance metrics needed for future loads. However, in this proposed implementation, services such as smart contracts cannot be loaded on top of blockchains like in Ethereum; they need to be part of the node's source code. Research regarding improvements in services, such as additions or updates of smart contracts without restarting the whole blockchain, can be beneficial for security and operational continuity, or implementation of the proposed concepts with other Blockchain frameworks is needed.

c) Performance metrics: Several properties can be studied regarding blockchain systems, including different abilities and performance metrics [20], [21]. Performance metrics remain open questions regarding U-space, as the blockchain's performance and traffic safety must be combined. The value of essential parameters has yet to be discovered. For example, the correct duration before U-spaceChain synchronization impacts the NI, TI, and CM services, calling for further study to find the correct number for future operational needs. Another example is the separation between unmanned air vehicles and the relationship between separation and capacity in U-space airspace[3]. The blockchain's performance can limit the

[2]https://www.sesarju.eu/projects/CORUSXUAM

[3]https://www.sesarju.eu/projects/SPATIO

capacity of the airspace. Moreover, no comparison can be made, with U-spaceChain being the first step in that direction. We hope this will foster research on performance metrics of blockchain applications for traffic management, taking into account operational objectives, communication overhead or delays, and other related metrics.

It's important to note that enablers, while crucial for generating required data and verifying system functionality, are not components of the U-spaceChain system and are not subject to its performance metrics. Although some variations in FA, NI, and WE occurred during execution, the implementation of the enablers can cause them. Therefore, complementary work is also needed to make the simulations more reliable and enablers more robust.

d) Consensus mechanism: Appointed-by-Authority is not yet a complete consensus mechanism. Although it allows U-spaceChain to work effectively within the prototype's test scenario, its performance and security aspects still need to be studied more. Knowing the *AbA*'s limits and ways to improve them will be a critical research perspective if U-spaceChain concepts are generalized to other fields.

e) Storage capability: Blockchain is not a data repository. Off-chaining or decentralized file systems could be envisioned for better scalability and performance to enhance U-spaceChain. Such concepts will add more storage capabilities while retaining all the system's qualities.

VII. Conclusion

This paper investigated how to provide UTM services compliant with the regulatory framework in a decentralized approach. Its conceptual contributions include the blockchain-based architecture associated with the Services as smart contracts and the *Appointed-by-Authority* consensus mechanism. The practical contributions are U-spaceChain, a proof-of-concept embodying the conceptual contributions, and four enablers allowing fast-time simulations on the prototype. U-spaceChain is based on the European U-space regulatory framework but can be translated into other drone regulations.

Although experimental, it can already serve as a basis for developing and testing more advanced services and establishing performance metrics for a future decentralized UTM framework. Thanks to its enablers, it allows simulation and monitoring of drone traffic for research purposes. U-spaceChain is positioned to serve as a valuable reference for researchers and professionals alike, providing guidance and insights for the application of blockchain in the UTM domain.

Acknowledgment

This work has been fully funded by the ENAC-Airbus-Sopra Steria *Drones and UTM* Research Chair. We would like to thank Yannick Seprey for his contributions during the discussion phase.

References

[1] M. Doole, J. Ellerbroek, and J. Hoekstra, "Drone delivery: Urban airspace traffic density estimation," 2018.

[2] EUROCONTROL, "Europe's atm capacity challenge," https://www.eurocontrol.int/sites/default/files/2019-05/skyway70.pdf, accessed: 2024-04-29.

[3] International Air Transport Association, "Blockchain in aviation white paper," IATA, techreport, 2018.

[4] M. Dehez Clementi, N. Larrieu, E. Lochin, M. A. Kaafar, and H. Asghar, "When air traffic management meets blockchain technology: a blockchain-based concept for securing the sharing of flight data," in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, 2019.

[5] A. Hamissi, A. Dhraief, and L. Sliman, "On safety of decentralized unmanned aircraft system traffic management using blockchain," in *2023 IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 2023, pp. 1–6.

[6] The European Commission, "Commission implementing regulation (eu) 2021/664 of 22 april 2021 on a regulatory framework for the u-space," 2021, http://data.europa.eu/eli/reg_impl/2021/664/oj.

[7] and Single European Sky ATM Research 3 Joint Undertaking, *U-space – Blueprint*. Publications Office, 2017.

[8] S. H. Alsamhi, B. Lee, M. Guizani, N. Kumar, Y. Qiao, and X. Liu, "Blockchain for decentralized multi-drone to combat covid-19 and future pandemics: Framework and proposed solutions," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 9, p. e4255, 2021, e4255 ETT-21-0077. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4255

[9] S. H. Alsamhi, A. V. Shvetsov, S. V. Shvetsova, A. Hawbani, M. Guizani, M. A. Alhartomi, and O. Ma, "Blockchain-empowered security and energy efficiency of drone swarm consensus for environment exploration," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 1, pp. 328–338, 2023.

[10] A. Allouch, O. Cheikhrouhou, A. Koubâa, K. Toumi, M. Khalgui, and T. Nguyen Gia, "Utm-chain: Blockchain-based secure unmanned traffic management for internet of drones," *Sensors*, vol. 21, no. 9, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/9/3049

[11] Y. Yaguchi and T. Wakazono, "Flight plan management system for unmanned aircraft vehicles using blockchain," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021.

[12] J. F. Nunamaker Jr., M. Chen, and T. D. Purdin, "Systems development in information systems research," *Journal of Management Information Systems*, vol. 7, no. 3, pp. 89–106, 1990. [Online]. Available: https://doi.org/10.1080/07421222.1990.11517898

[13] M. Conway, "How do committees invent?" http://melconway.com/Home/Conways_Law.html, accessed: 2024-04-29.

[14] E. Caillaud, B. Rose, and V. Goepp, "Research methodology for systems engineering: some recommendations." *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1567–1572, 2016, 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896316310850

[15] Acube by Airbus, "Airbus utm blueprint," Airbus, Tech. Rep., 2018.

[16] L. H. Sullivan, "The tall office building artistically considered," 1922.

[17] H. Taherdoost, "Smart contracts in blockchain technology: A critical review," *Information*, vol. 14, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2078-2489/14/2/117

[18] B. H. Rakotonarivo, N. Drougard, S. Conversy, and J. Garcia, "Cleared for safe take-off? improving the usability of mission preparation to mitigate the safety risks of drone operations," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3544548.3581003

[19] I. C. A. O. (ICAO), "International aviation trust framework." [Online]. Available: https://www.icao.int/airnavigation/IATF/Pages/default.aspx

[20] S. Smetanin, A. Ometov, M. Komarov, P. Masek, and Y. Koucheryavy, "Blockchain evaluation approaches: State-of-the-art and future perspective," *Sensors*, vol. 20, no. 12, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/12/3358

[21] S. Spencer-Hicken, C. S. L. Schutte, and P. J. Vlok, "Blockchain feasibility assessment: A quantitative approach," *Frontiers in Blockchain*, vol. 6, 2023.