

# 3rd IAA Latin American Symposium on Small Satellites

IAA-LA 2022

## Blockchain Applied in the Update the Firmware of Educational Nanosatellites

José Edilson Silva Filho<sup>(1)</sup>, Igor Braga Palhano<sup>(1)</sup>, Nicolas de Araújo Moreira<sup>(1)</sup>,  
Jarbas Aryel Nunes da Silveira<sup>(1)</sup>

<sup>(1)</sup>Federal University of Ceara, edilsonfilho@lesc.ufc.br

**Keywords:** Educational Nanosatellites, University, Firmware, Blockchain, Ethereum, Hyperledger Besu.

Space has increasingly attracted the attention of governments, large industries, and universities. One of the most popular strategies in recent years has been the adoption of nanosatellites to fulfill different missions, which can work alone or in constellations. Universities appear in the spotlight among the nanosatellite launch agents, with more than 600 launches until 2022. Updating the firmware of satellites or nanosatellites in orbit is always challenging, as the device needs to be visible to ground stations to receive data packets at an average of 10 minutes per pass. The firmware image is shredded and shipped until all the firmware can be mounted and the system completes the update. In this process, the data packet can be corrupted by natural phenomena or intentionally by intruders, and a new request will have to be redone. In general, universities have low financial resources to access Earth Stations frequently, and their projects are only protected if they can correct a firmware defect in the orbiting nanosatellite. This work presents a proposal to decentralize the use of ground stations to update the firmware of university nanosatellites. We proposed a consortium between universities and institutions to create a decentralized structure for providing firmware updates. For this, we propose using Blockchain technology and the concept of smart contracts to govern the process and diffusion of the new firmware throughout the consortium's network of satellites. We use the Hyperledger Besu Blockchain to create an Ethereum client and allow CubeSats to access the servers to make firmware update requests. In practice, the nanosatellite does not request only to its home stations but to any station on the network and relies on encrypted transactions to bring security to the procedure. The result is a faster propagation speed of firmware for each CubeSat. Our proposal still involves the creation of a token called GS-BC with a value of 10 percent of ETH to govern the monetization of the service. The result shows the feasibility for universities and small and medium-sized companies to access the service without incurring huge expenses. The system also rewards institutions with earth stations, generating a new incoming source.

## 1. Introduction

One of the satellite construction standards that has become popular in recent years is the CubeSat pattern [1], cubic-shaped nanosatellites measuring 10 x 10 x 10 cm. They are divided into units, with 1 U corresponding to the smallest unit of the standard. The format is popular because of its low development and launch costs, usually uses COTS (Commercial-Off-The-Shelf) parts, and is lightweight. According to

the Nanosats Database, universities represent a significant part of the market share of nanosatellite builders launched in 2022. Around 640 new nanosatellites are expected by the end of 2022, representing an increase of almost 98 percent over the previous year. In an environment where more and more governments restrict education budgets, financial resources for universities and educational projects are limited. Some universities do not have ground stations for tracking, telemetry, and control of their CubeSats and need public and private partnerships to rent or use ground stations from other entities. In this scenario, solutions to reduce the costs of research and space missions can increase the participation of universities in the nanosatellites ecosystem.

As with any electronic device, CubeSats run over embedded software called firmware, which may need to be updated periodically [2] [3], either because of a vulnerability that needs to be fixed or because of new functions.

The firmware update process is quite delicate. First, one of the bits may accidentally be reversed while transmitting and exchanging data, which can occur, for example, due to environmental reasons such as a solar storm or effects on the Earth's atmosphere. Second, there is another challenge regarding communication between the ground station and the nanosatellites. CubeSats operating in Low Earth Orbit (LEO) take between 8 min to 10 min [4] to overpass the communication window with the ground station. Within such a short time window, the amount of data transmitted is minimal, and the communication must be correctly done. The nanosatellite could take hours to overpass the station on Earth again. In the following sections, we present strategies to mitigate the challenges.

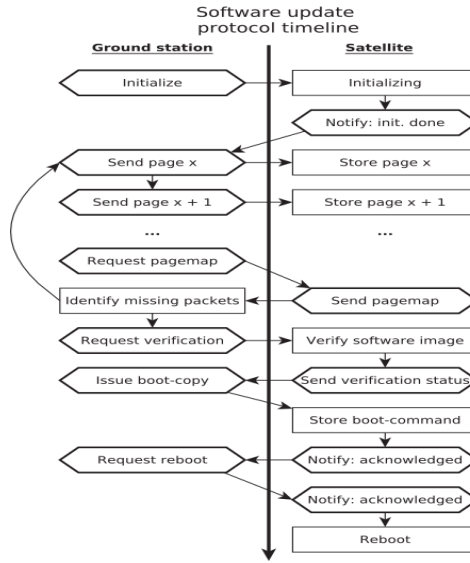
The remainder of this paper is as follows: Section 2 discusses CubeSats' state-of-the-art and firmware upgrade process. Section 3 explains blockchain and how this decentralized infrastructure of ground stations is built. Section 4 introduces the experiments performed and their results, and finally, Section 5 presents the conclusion of our work and future projects.

## **2. Nanosatellite Firmware Upgrade**

In general, the remote firmware update procedure on space devices encompasses four well-known parts [5]: the first part is dedicated to preparing the firmware image, fragmenting [6] [7] it into small blocks and delta-compressed [8] to be sent by the earth station; the second is the transfer protocol; the third part is related to storage in the nanosatellite memory and post-processing, that is, checking if the image is complete if it has been corrupted or altered. After the verification process, the routine points to the onboard computer, the new memory address that the firmware has, and initializes the system. A new request for this data packet is performed if any errors are found. Error Correcting Codes (ECCs) are commonly used in this step [9] [10] [11]. Figure 1 represents the Software Update Protocol Timeline [5]. In our timeline, we add the satellite identification and the last captured packet. Table 1 shows some examples of missions with updates.

Figure 1 represents the mission of ESTCube-1, a CubeSat focused on solar sailing experiments: it was designed to be reprogrammed in orbit.

Now, we can understand the importance and the need for a safe procedure for updating the firmware of nanosatellites and other equipment and spacecraft.



**Figure 1: Software Update Protocol Timeline.**

**Table 1: Examples Cubesats that underwent an in-orbit firmware update**

Name	Mission Time (Month)	Number of Updates
AeroCube4 [12] [13]	18	150
ESTCube-1 [14]	Uninformed	35
STRaND-1 [14]	24 months, with many flaws	Uninformed
UWE-3 [15]	Uninformed	3

### 3. Blockchain and Decentralized Infrastructure of Ground Stations

The present section is composed of two subsections. The first one is an additional exposition of the context of what has already been presented in the introduction regarding the relevance and demand of university nanosatellites. Then, some blockchain concepts are introduced, and an explanation of how it can be a powerful tool for building consortia to create a decentralized infrastructure of ground stations is shown.

#### *Contextualization*

As already presented in the introduction to this work, the costs and expenses to carry out projects in the space sector are enormous. Universities worldwide are looking for solutions to give continuity to their initiatives, and one example of cost-sharing and research is the BIRDS Ground Station Network [16].

BIRDS is a satellite constellation project involving students from 15 countries. The constellation usually encompasses 1 U CubeSat equipped with cameras, using UHF/VHF for communication and enabling to perform experiments.

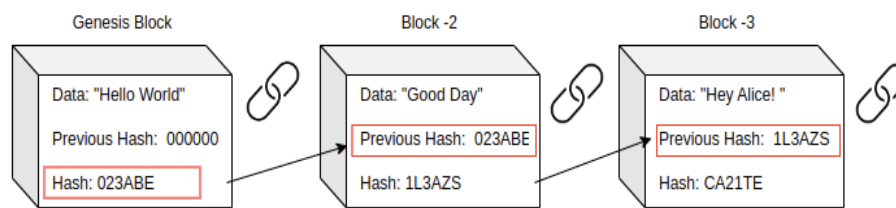
The limitation is that it needs to be clarified whether privacy and individuality exist, for example, in the firmware update process. Our proposal goes beyond generating a shared antenna network but in an individual, confidential, and secure firmware update service. We can foresee that some CubeSat developers, engineers, and researchers in the network wanted to keep their firmwares secret but wanted to share resources like

airtime and others. Another point is seen where it shows that the application is centralized; we can intuit that if there is an availability problem, the service may go down. Another exciting project is Amazon's AWS Ground Station, which rents terrestrial antennas to companies and startups. However, the prices are sometimes unacceptable for educational projects or those just starting. In the next segment, we will present an essential concept for our project.

### *Blockchain*

Blockchain is a list chained and a growing number of records called blocks with encryption that protects them. Blockchain has some components: data, hash, previous hash, and metadata (time/date stamp and block number); refer to Figure 2.

- Data: can be a simple string or a list of transactions;
- Hash: is a unique identifier for a block and is analogous to a fingerprint;
- Previous hash: is the hash value of the previous block in the list;
- Metadata: Information about the data, such as the block number, date, and time.



**Figure 2: High-level representation of a blockchain**

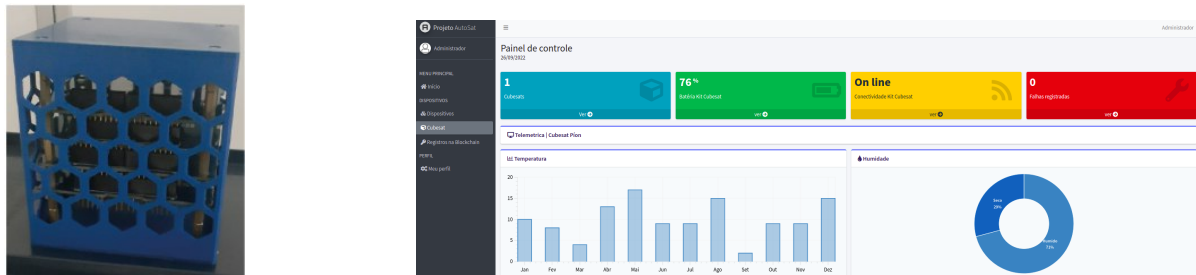
There are two main types of Blockchain: Permissionless and Permissioned [17]. In a Permissioned Blockchain, the distributed ledger can only be accessed by a few people or nodes who have been allowed to do so by the administrator. In the Permissionless model, also known as a public blockchain, there are no restrictions, and an administrator does not control the participation. Another widely used concept is the Smart Contract [18]. The smart contract concept is not new, but the Ethereum Blockchain allowed him to could be implemented reliably. They are self-executing contracts, blocks of codes automatically performed when reaching specific markers. It is the basis of transactions we will carry out. Another essential concept is Web3, a new iteration of the internet based on blockchain technology, which incorporates concepts such as decentralization and a token-based economy. We must mention that ESA [19], NASA [20] have already expressed interest in using Blockchain. Based on the fact that we have already laid the necessary foundations in contextualization and Blockchain, this topic will deal with practical points of the suggested infrastructure. The process of a decentralized ground station infrastructure can be understood as follows: Each ground station is a node in the decentralized network. Each ground station has the power to write data to the Blockchain and to read the data. In Section 4, this will be further detailed. Table 2 complements the reasons for our proposal

**Table 2: Motivation for Decentralization Infrastructure GS with Blockchain**

Motivation	Potential Benefit
Automation with smart contracts	-Automating smart contracts running on the blockchain is interesting because routines and algorithms, for example, sensor calibration or orbit corrections, can be executed without human intervention.
Privacy with cryptographic technology	-Decentralized infrastructure with blockchain can be combined with modern cryptographic techniques to preserve the privacy of codes, binaries and transactions, keeping each manufacturer's firmware code confidential.
Tokenization	-The possibility of tokenizing routines and some processes can generate an additional source of income for the institutions that maintain the service and even attract private initiatives to invest in projects.
Service availability	-If one of the antennas stops operating, the others continue to offer the service, and the firmware update process continues without stopping.
Cost Sharing	-University and other projects that will have the opportunity to share communication costs between ground stations and nanosatellites, generating savings and cheaper space missions.

#### 4. Experiments and Results

We use the PION Educational Cubesat from PIONS Labs [21], which has nine sensors for space mission data collection: brightness, temperature, pressure, humidity, CO<sub>2</sub>, battery level, gyroscope, magnetometer, and accelerometer. The CubeSat processor is 32-bit (ESP32), and we developed a platform for real-time data collection (refer to Figure 3). The platform assigns a blockchain identity to CubeSat and governs other operations, such as sending and requesting data.

**Figure 3: PION Educational CubeSat and Dashboard Web**

#### *Choosing Blockchain and Fragmentation Algorithm*

We chose the Hyperledger Besu blockchain [22] to implement smart contracts. The reasons for this choice are:

1. The Besu framework allows the creation of public and private networks;
2. It has several plugins and APIs for querying transactions efficiently;
3. It is an Ethereum client and brings together the features of the Hyperledger family;
4. It has several types of consensus algorithms.

We implement a fragmentation algorithm in C that takes firmware images of any size and breaks them into 32 bytes fragments - the payload size (data) that we place in each writing transaction on the blockchain. Several fragmentation algorithms exist

in the literature, such as the "LoRaWAN Fragmented Data Block" [23], with technical explanations and example code in Matlab.

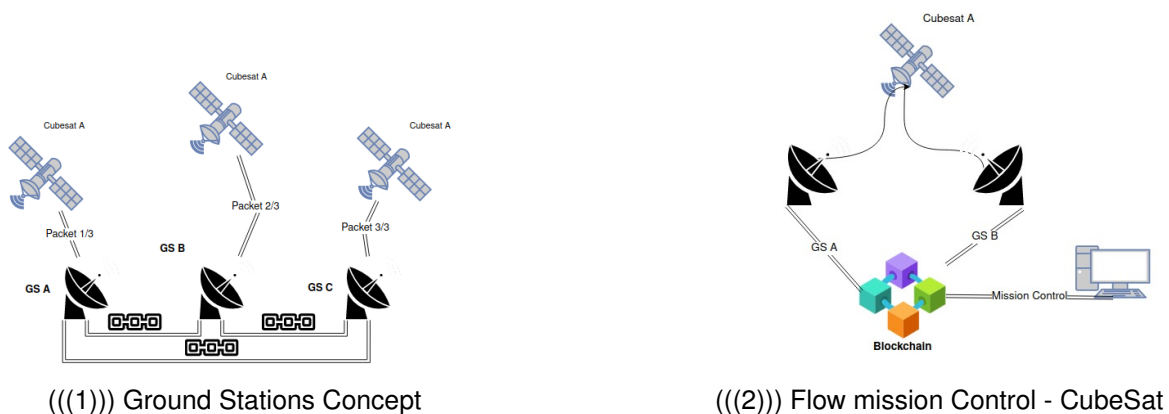
### Simulation

Table 3 presents the specifications used in the experiment, and Figure 4 represents the decentralization of sending the firmware image. For this, each Earth Station runs a private blockchain node in the consortium format. In our business model, the partner mission controls access to a web platform that communicates with the blockchain via web3.

The application backend fragments the firmware image respecting the maximum size of each packet that can be sent depending on the communication bandwidth used. For our tests, we simulated UHF. Important: Each institution has public and private keys to sign the transaction. The fragment is encrypted and stored. Thus, only the CubeSat of the developing institution can access the firmware fragment intended for it. The next step occurs when the CubeSat passes within the station's range. The flow in Figure 1 happens each time it enters a new range of partner stations.

**Table 3: Specifications**

Item	Description
Blockchain	-Hyperledger BESU 20.10.0
Used languages	-Solidity (Smart contract) and NodeJS (API)
Consensus algorithm	-IBFT
Frameworks	-Remix-ethereum (online) [24], Metamask (wallet), VS Code (IDE) and Docker Composer
Devices	-PION (CubeSat, with ESP32), Notebook DELL 8 GB RAM, 2.1Ghz I3 (Blockchain locally with 3 nodes and 1 rpc)



**Figure 4: Decentralized Infrastructure of GS in Blockchain**

Although the servers of the ground stations hosted the blockchain, we present below some helpful information about the cost and time relationship of the main smart contract methods. Note that the only transaction that generates costs is writing to the

blockchain; the other methods do not generate costs, making it cheaper for institutions to carry out the maintenance process.

Our experiment simulated the earth station server through a notebook, as referenced in Table 4, and requests replaced the communication via antenna via HTTP (using the internet). In order to simplify our procedures, we simulated an earth station server using a notebook (see Table 4), and CubeSat PION requests were made via the web using a REST API. As each transaction on the blockchain stores in its data field 32 bytes, the fragments created obey this size. When CubeSat makes requests, the backend on the server asks for the desired index. It queries the blockchain and creates a new data packet by joining all small 32bytes fragments up to a specific limit to comply with the maximum size the channel supports sending. For our project, we created a web platform that can be accessed at [www.autosat.iochip.com.br](http://www.autosat.iochip.com.br). The platform allows the user to register a device, assign a cryptographic identity, and query telemetry data.

## Results

The first result found is the cost in dollars to store the firmware on the blockchain. The Gas is the amount that will be charged for each transaction on Ethereum. Gas prices are indicated in Gwei, a proprietary denomination of ETH in which each Gwei equals 0.00000001 ETH. This value varies according to several factors, such as the network demand and the operation type. At the same time, each fragment is 32bytes in size (refer to Table 4 for some related costs - note the matching of firmware size and transaction cost in US Dollars). Although there is variation in the price of a cryptocurrency like ETH, institutions can predict how much they will spend on firmware update processes in their projects. The price of ETH against the dollar was from September 2022.

**Table 4: Cost to Add Firmware Fragments on Blockchain**

Size Kb	Firmware in	Nº Fragments	Gas	ETH	USD
1		31.25	9017687.5	0.017206	22.99
2		62.50	18035375.0	0.034412	45.98
3		93.75	27053062.5	0.051618	68.97
4		125.00	36070750.0	0.068823	91.96

Figure 5 represents the log of a storage transaction of a 32 bytes firmware fragment. Note that when analyzing Figure 5, there is a tag called "creator" - the same identifier shown in Figure 6 (Account 2). We call this the wallet address to give privacy to each institution and prevent the firmware from being sent to the wrong CubeSats. This information goes in the metadata.

An essential point that our architecture allows is the creation of tokens, i.e., digital assets, that can replace ETH, which is interesting because the transaction prices would not be tied to the fluctuations of ETH. At the same time, it allows a new way to monetize the project and create businesses on top of the solution. In Figure 8, this portfolio has 200 GS-BC (An ERC-20 token worth 10 percent of ETH); it is an example of a token that could replace ETH within the consortium of partner institutions.

### Figure 5: Log of transaction createFirmwareFragment()

**Table 5: Cost Ground Station VHF/UHF**

A traditional VHF/UHF station used for CubeSat missions costs around USD 69,7K. If we depreciate this value into the number of minutes in an 18-month mission, we will determine how much that station cost per minute for the project. This calculation does not consider technical operators and energy costs, which can make some educational and even small business projects unfeasible. There are commercial alternatives



for leasing earth stations; a highlight is Amazon (AWS Ground Station), which has an exciting quality standard. Although at first, its final cost in a 12-month package is USD 19,800.00 for an 18-month mission, it was estimated at USD 29,700.00 with only 248 minutes per month. In our study, in addition to proposing the decentralized use of earth stations, we created a GS-BC token and offered the GS usage service at USD 1.79 per minute. Our calculations show that in an 18-month mission, the cost of shared service can represent a savings of 66.7 percent of the purchase price of a VHF/UHF station. It becomes an option, as it allows better management of the earth station rental resource; it is cheaper than other alternatives and can still become a source of revenue for institutions that have earth stations for rent in the consortium.

## 5. Conclusion

Firmware updates are an essential component of successful space missions. Considering the budgetary constraints that several university projects suffer from the dynamic and unstable world market, solutions that allow cost-sharing and missions are always welcome. Our project does not just offer a solution to the problem of cheaper university space missions but also for industrial and commercial initiatives. Our project provides a robust and secure solution and a reliable infrastructure for uploading and firmware-update for space devices, in the studied case, for CubeSats. Our project proposed costs in Eth and a token we created called GS- BC with an initial fixed value of 10 percent of ETH, which can change depending on the consortium.

As a future project, we want to expand the study and implementation of technology for these other processes, such as data acquisition and sensor calibration, using centralized ground station infrastructure and blockchain. We also concluded that the costs of registering firmware on the blockchain network are acceptable as they can be paid in cash, facilitating operators' strategic planning.

## References

- [1] I.Nason, J. Puig-Suari, and R. Twiggs, Development of a family of picosatellite deployers based on the CubeSat standard , IEEE Aerospace Conference Proceedings (2002).
- [2] Garrido, B. García, A. Alfaro, N and Miguel, J, On-board software maintenance, Proceedings of the DASIA '98 Conference on Data Systems in Aerospace (1998).
- [3] Greco, M .E and Snyder, J. F, Operational modification of the Mars explorations rover's flight software, IEEE Systems, Man and Cybernetics, International Conference (2005).
- [4] R. A. Carvalho, Optimizing the communication capacity of a ground station network, Journal Aerospace Technology and Management (2019).
- [5] I. Sünter, A. Slavinskis, U. Kvell, A. Vahter, H. Kuuste, M. Noorma, J. Kutt, R. Vendt, K. Tarbe, M. Pajusalu, M. Veske, T. Ilves, Firmware updating systems for nanosatellites, IEEE Aerospace and Electronic Systems Magazine 31 (2016) 36–44.
- [6] S. M. O. R. Leiner, B., B. Huber, A comparison of partitioning operating systems for integrated systems, Proceedings of 26th International Conference on Computer Safety, Reliability, and Security (SAFECOMP) (2007) 342–355.
- [7] J. J. Rosa, J. Rufino, Exploiting air composability towards spacecraft onboard software update., Actas do INForum-Simpósio de Informática (2010).
- [8] B. Bing, A fast secure framework for over-the-air wireless software download using reconfigurable mobile devices., IEEE Communications Magazine (2006) 58–63.
- [9] L. Wood, W. Eddy, W. Ivancic, J. Mckim, C. Jackson, Saratoga: a delay-tolerant networking convergence layer with efficient link utilization, pp. 168 – 172.
- [10] F. Silva, A. Muniz, J. Silveira, C. Marcon, Clc-a: An adaptive implementation of the column line code (clc) ecc, in: 2020 33rd Symposium on Integrated Circuits and Systems Design (SBCCI), pp. 1–6.

- [11] R. Hamming, Error detecting and error correcting codes, The Bell System Technical Journal 29 (1950) 147–160.
- [12] J. W. Gangestad, B. S. Hardy, D. Hinkley, Operations, orbit determination, and formation control of the aerocube-4 cubesats.
- [13] J. W. Gangestad, D. W. Rowen, B. S. Hardy, Forest fires, sunglint, and a solar eclipse: Responsive remote sensing with aerocube-4, in: 2014 IEEE Geoscience and Remote Sensing Symposium, pp. 3622–3625.
- [14] C. P. Bridges, B. Yeomans, C. Iacopino, T. E. Frame, A. Schofield, S. Kenyon, M. N. Sweeting, Smartphone qualification linux-based tools for cubesat computing payloads, in: 2013 IEEE Aerospace Conference, pp. 1–10.
- [15] "UWE-3 Software Update in Space" 2014, [On Line], Accessed: 10-Nov-2022. <https://www.informatik.uni-wuerzburg.de/space/forschung/space-exploration/projects/uwe-3/uwe-3-news/single-news/news/uwe-3-software-update-in-space-1/>.
- [16] T. Tumenjargal, J. Members, J. Partners, M. Cho, Development status of joint global multi-nation birds cubesat constellation project.
- [17] A. B. Melo, Tecnologia blockchain: Fundamentos, tecnologias de segurança e desenvolvimento de software, 2017. <https://www.cpqd.com.br>.
- [18] Ethereum blockchain app platform, 2013. <https://www.ethereum.org/>.
- [19] Blockchain and earth observation white paper, April 2019. <https://eo4society.esa.int>.
- [20] J. d. La Beaujardiere, R. Mital, R. Mital, Blockchain application within a multi-sensor satellite architecture, in: IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, pp. 5293–5296.
- [21] Pion educational cubesat, 2022. <https://www.pionlabs.com.br/cubesat>.
- [22] Hyperledger besu blockchain, 2015. <https://www.hyperledger.org/use/besu/>.
- [23] Lorawan fragmented data block, 2018. <https://lora-alliance.org>.
- [24] Remix ethereum framework, Accessed in setembro 2022. <https://remix.ethereum.org/>.