A BDI-based Multi-Agent System to Smart Parking Environment with Blockchain Technology

Gabriel L. M. Oliveira[1], **Gleifer Vaz Alves**[1], **Nilson M. Lazarin**[2]

[1]Universidade Tecnológica Federal do Paraná (UTFPR) – Ponta Grossa – PR – Brazil

[2]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (Cefet/RJ)
Nova Friburgo, RJ – Brazil

`gabrieloliveira.2020@alunos.utfpr.edu.br`

`gleifer@utfpr.edu.br, nilson.lazarin@cefet-rj.br`

Abstract. *This paper presents the development of a BDI (Belief-Desire-Intention) multi-agent system integrated with blockchain technology for decentralized digital asset negotiation in a Smart Parking environment. Using the Velluscinum middleware, the system enables the use of NFT to represent parking spots. Tests demonstrated its feasibility and functionality, though limitations such as partial centralization and network dependency were noted. The work contributes to the integration of BDI agents and blockchain in Smart City contexts, with future improvements focused on decentralization and broader integration.*

1. Introduction

The increasing reliance on decentralized intelligent systems is transforming daily life through connected devices such as smartphones, wearables, and smart vehicles [Calvaresi et al. 2019]. In this context, Multi-Agent Systems (MAS) emerge as a promising solution to complex problems, particularly those requiring collaboration among autonomous agents.

Within the field of Artificial Intelligence, MAS aim to build adaptive entities that operate in dynamic and distributed environments [Papi et al. 2022]. The BDI (Belief-Desire-Intention) model is a widely adopted cognitive architecture that structures agent behavior through mental states—beliefs about the world, desires as objectives, and intentions guiding actions. This approach has proven effective in both human cognition modeling and the implementation of distributed MAS.

With the expansion of digital assets such as cryptocurrencies and non-fungible tokens (NFT), new opportunities arise for autonomous agent negotiation. These assets offer secure and decentralized mechanisms for value exchange, reducing the need for intermediaries. Nonetheless, the integration of digital assets into MAS remains an emerging field with considerable potential [Oliveira et al. 2023].

To support this integration, middleware platforms like Velluscinum [Mori Lazarin et al. 2023] enable secure communication between MAS and blockchain infrastructures such as BigchainDB [McConaghy et al. 2016]. Additionally, protocols like the Contract Net Protocol [Smith 1980] and trust degree mechanisms [Souza De Castro et al. 2017] have been employed to support negotiation and decision-making processes in Smart Parking contexts [Alves et al. 2019].

This work proposes the development of a BDI-based multi-agent system integrated with blockchain to enable secure and decentralized negotiation of digital assets in a Smart Parking environment. The system leverages the JaCaMo framework [Boissier et al. 2016] for agent programming and Velluscinum, that provides a ready-to-use package compatible with JaCaMo, for blockchain communication. Our major contributions are:

1. Providing an integration of digital assets and a BDI agent architecture;
2. Enhancing the use of Velluscinum middleware;
3. Presenting an application example using a smart parking scenario representing a multi-agent system with a hybrid negotiation mechanism.

2. Background

This section presents fundamental concepts for understanding the architecture and integration of the proposed system. It is structured into three parts: the concept of intelligent agents and the BDI model, the fundamentals of blockchain technology, and the role of middleware in integrating MAS with distributed ledgers. These elements serve as the foundation for the system's design and implementation.

2.1. Intelligent Agents and BDI Model

An intelligent agent is an autonomous computational system capable of acting independently to achieve its goals, without requiring explicit instructions [Wooldridge 2009]. In a Multi-Agent System, multiple such agents interact within a network, promoting decentralized decision-making and enabling cooperation or competition in dynamic environments [Papi et al. 2022].

The Belief-Desire-Intention model, inspired by cognitive psychology [Bratman et al. 1988], is widely used in MAS to represent the mental states of agents. Beliefs represent the agent's perception of the environment, desires reflect its objectives, and intentions correspond to the actions it commits to in order to achieve those goals [Rao 1996]. BDI agents adapt their behavior by updating their beliefs and intentions in response to environmental changes.

Communication is crucial in BDI-based MAS, enabling agents to share beliefs, goals, and intentions to coordinate actions and resolve conflicts [Wooldridge 2009]. It supports the dynamic adaptation of agents and enhances cooperation through mechanisms such as argumentation [Sichman et al. 1992].

Negotiation protocols define structured rules that guide interactions between autonomous agents aiming to reach agreements. These interactions are influenced by factors such as goal alignment, task dependencies, and incomplete information [Alves et al. 2019]. Effective negotiation strategies are vital for achieving consensus in competitive or cooperative multi-agent environments.

2.2. Blockchain and Middleware Integration

Blockchain is a decentralized, tamper-resistant system that ensures secure transactions through Distributed Ledger Technology (DLT) [Kumar et al. 2023]. It employs cryptographic techniques, such as hash functions and digital signatures, to guarantee data integrity. Consensus mechanisms and smart contracts further enhance trust and automation in blockchain-based systems.

Digital assets such as cryptocurrencies and non-fungible tokens (NFT) allow secure and decentralized value exchange between agents [Minarsch et al. 2021]. These assets, which can be divisible or indivisible, broaden the scope of economic interactions in distributed systems.

DLTs can be classified as permissioned or permissionless [Rajasekaran et al. 2022]. Permissioned DLTs restrict access and validation rights to authorized participants, offering greater control and privacy. In contrast, permissionless DLTs, such as Bitcoin, allow open participation and promote decentralization and transparency.

Middleware plays a key role in facilitating communication and interoperability between MAS and blockchain technologies. It acts as an intermediary layer that abstracts the complexities of each system, enabling agents to interact with distributed ledgers seamlessly. This integration enhances trust, traceability, and automation in decentralized applications.

The combination of BDI agents with blockchain through middleware like Velluscinum [Mori Lazarin et al. 2023] has demonstrated potential for Smart City contexts, particularly in applications like Smart Parking. Prior works, such as by Alves et al. (2019), have successfully used agent-based approaches with negotiation protocols in parking scenarios, highlighting the viability of this integration.

3. Related Work

Recent studies aim to combine the advantages of BDI architecture, blockchain, and MAS into practical applications. Oliveira et al. (2023) conducted a systematic mapping to identify research gaps and consolidate literature on agent-based negotiation involving digital assets. Their analysis included key aspects such as:

- [Papi et al. 2022]: blockchain integration in MAS asset transactions;
- [Calvaresi et al. 2018a, Calvaresi et al. 2019]: trust and ethical implications of MAS-blockchain integration;
- [Luo et al. 2019, Liu et al. 2021, Mhamdi et al. 2022]: blockchain-supported agent cooperation in various domains;
- [Song et al. 2022]: BDI frameworks for resilient multi-domain SDN;
- [Calvaresi et al. 2018b]: permissioned DLT for reputation in MAS;
- [Mori Lazarin et al. 2023]: Middleware for digital asset negotiation in MAS.

This systematic mapping reveals that most approaches rely on smart contracts and emphasize decentralization. However, only a few combine permissioned DLTs with BDI agents, such as Calvaresi et al. (2018a) and Papi et al. (2022). Lazarin et al. (2023) stands out by proposing a middleware tailored to BDI agents negotiating through DLTs. Nonetheless, we can notice the lack of works towards smart parking applications based on BDI-MAS and permissioned DLTs, which motivates the system presented here.

4. System Development

The BDI Multi-Agent System was developed using the JaCaMo framework, with specific adaptations to agents and scenarios to enhance performance and achieve more effective outcomes. The framework was selected due to its native integration with the Velluscinum middleware, enabling seamless communication and coordination among agents.

The system encompasses scenarios that illustrate message exchanges between agents and their interaction with the blockchain through the Velluscinum, which is available in `https://github.com/GabrielOliveira23/smartParking`.

To implement and develop the proposed MAS, a set of tools was required, enabling the programming of agents and the integration with external technologies such as the Velluscinum middleware. These tools played essential roles in different stages of development, providing support for simulation, digital transaction management, and agent communication.

The development of the multi-agent system was carried out entirely using the JaCaMo framework [Boissier et al. 2016] through the JaCaMo CLI distribution, version 1.2. This version includes Jason 3.2, Moise 1.0, SAI 0.5.3, and Cartago 3.1. Additionally, Java 17 is required to use this framework.

Figure 1 provides an overview of the system's operation, highlighting how it manages negotiations securely and efficiently through Velluscinum requests, operating in a decentralized manner. It also illustrates the system's main agents and indicates the types of requests each one can access to communicate with the blockchain.

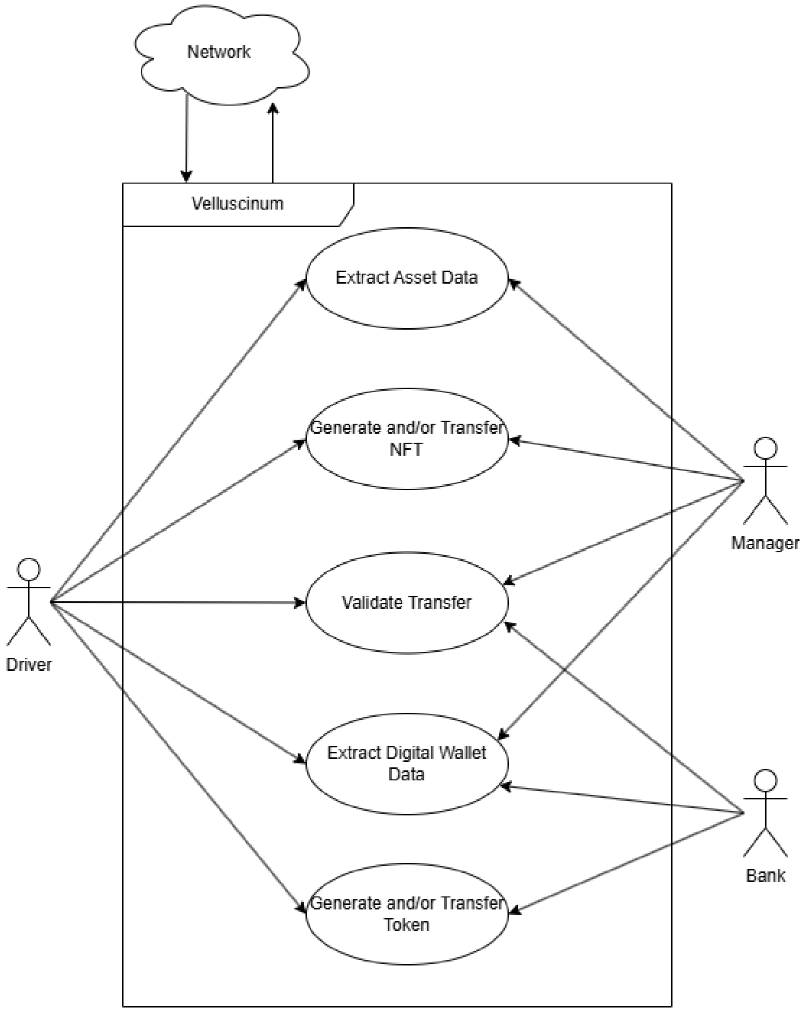


Figura 1. General System Diagram.

Agents: The multi-agent system consists of four main types of agents: Bank, Manager, Driver, and Creator. Each agent plays a specific role in the simulation, contributing to the operation of the decentralized Smart Parking. At system startup, the Creator generates the participating agents, and the Bank agent creates the currency—a divisible token—used in transactions, distributing it via loans to the Driver agents. Meanwhile, Drivers and Managers create or load their digital wallets. The Manager is also responsible for organizing parking spots, which are represented as NFT.

Bank: The Bank agent plays a crucial role by creating the divisible token used as currency for transactions within the system. It provides loans to the Driver agents, enabling them to engage in negotiations involving Smart Parking spaces and peer-to-peer transactions. In addition, the Bank maintains a belief regarding the server route, which is essential for Velluscinum middleware requests to function correctly [Mori Lazarin et al. 2023]. This route is shared with other agents to make blockchain integration work.

Manager: The Manager agent is in charge of creating the NFT that represent the parking spots. Each spot has a unique ID and a specific type, such as short, covered short, long, or covered long. Beyond this, the Manager also performs Smart Parking management operations, including the sale of spots, validation of transfers, and creation of reservations. It also validates transactions and communicates directly with Driver agents to carry out these operations.

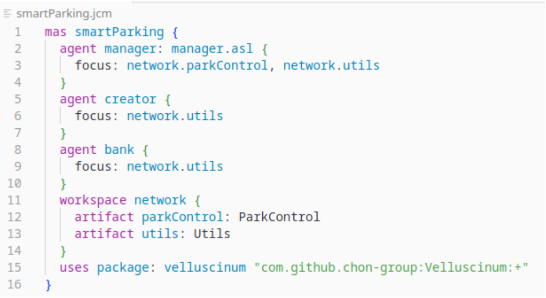
Driver: The Driver agent simulates an individual interested in using the Smart Parking. It is capable of executing various actions, such as purchasing a spot for immediate use, reserving a spot with the Manager, buying a reservation from another Driver, selling a reservation, and using a stored reservation. Each Driver maintains specific beliefs and manages its own digital wallet, allowing it to conduct transactions and interact with other Drivers and the Manager.

Creator: The Creator agent serves as an auxiliary entity responsible for automating the creation of Driver agents. Each Driver created is unique, with its respective digital wallet and predefined beliefs, ensuring diversity and individuality among the agents. This set of agents and their interactions form the foundation for operating the Smart Parking system, enabling the simulation of a decentralized system integrated with blockchain, as detailed in the following scenarios.

4.1. System/Velluscinum Integration

The Velluscinum middleware provides intelligent agents with the ability to create and transfer digital assets, validate transactions, and manage their wallets directly within the agent dimension [Mori Lazarin et al. 2023]. In the context of Smart Parking, the Bank, Driver, and Manager agents leverage the functionalities provided by Velluscinum, ensuring the decentralization of the system's negotiations. This approach allows all agents to maintain full awareness of the transactions performed, the values involved, and the validations executed on the blockchain, promoting transparency and trust in the operations.

To configure and integrate Velluscinum into the developed multi-agent system, the package was imported into the SMA configuration file (the .jcm file), as shown in Figure 2. A belief was then created in the Bank agent to store the server port and path. The Figure 3 show the use of a public server provided by ChonNet [Lazarin et al. 2023].



```
mas smartParking {
  agent manager: manager.asl {
    focus: network.parkControl, network.utils
  }
  agent creator {
    focus: network.utils
  }
  agent bank {
    focus: network.utils
  }
  workspace network {
    artifact parkControl: ParkControl
    artifact utils: Utils
  }
  uses package: velluscinum "com.github.chon-group:Velluscinum:+"
}
```

Figura 2. Import Velluscinum.

```
{ include("$jacamo/templates/common-cartago.asl") }
{ include("$jacamo/templates/common-moise.asl") }

/* Initial beliefs and rules */
chainServer("http://testchain.chon.group:9984/").
mensagensEnviadas(0).

/* Initial goals */
!carregarCarteira.

/* Plans */
+!incMensagensEnviadas : mensagensEnviadas(Num) <-
    -+mensagensEnviadas(Num+1).
```

Figura 3. Server Configuration.

This server key is subsequently broadcast to all system agents, enabling their access to the middleware.

Once the initial setup is complete, the agents connected to the server can perform the requests provided by the middleware. The Bank, Driver, and Manager agents take advantage of the features described in the following subsections.

4.2. Artifacts

The developed system incorporates several artifacts that support the execution of the Smart Parking system, providing auxiliary functionalities that enhance the overall operation of the multi-agent system. These artifacts serve as utility components, such as tracking the number of messages exchanged between agents during system runtime.

4.2.1. ParkControl

The ParkControl artifact is designed to manage parking spots and their corresponding reservations within the MAS. It handles essential operations such as availability checks, reservation record and lookup, and usage cost calculations. These functionalities are implemented directly within the artifact due to the convenience of handling the structured data returned by Velluscinum regarding digital assets. Key features include:

- **Validation of Purchase and Reservation:** Ensures that a parking spot is available for a specific type and checks for potential conflicts with existing reservations.
- **Reservation Record and Lookup:** Enables the creation of new reservations and retrieval of existing ones, associating relevant metadata such as identifiers and usage periods.
- **Parking Spot Information Retrieval:** Provides access to a spot's status and type.
- **Usage Cost Calculation:** Computes the proportional cost of a reservation based on usage duration and pricing rules.

Through integration with supporting methods and structured data handling, ParkControl enables efficient and flexible parking spot management.

4.3. Utils

The Utils artifact offers general-purpose utilities to aid the operation of the MAS. These include functionalities for message tracking, data conversion, and budget-aware parking duration estimation. Main functionalities include: Message Counter, to monitor the number of inter-agent messages; Estimated Parking Duration, that generates a random usage duration and ensures the projected cost does not exceed predefined budget.

These artifacts collectively enhance the MAS's ability to handle intelligent decision-making and autonomous coordination in a smart parking environment.

4.4. Scenarios

The system addresses scenarios that reflect typical communications and transactions within a Smart Parking environment, simulating interactions between Drivers, who seek to park their vehicles, and Managers, who oversee the sale and reservation of parking spots.

1. **Immediate Use – Purchase:** This scenario depicts a situation in which a Driver seeks to park immediately. This triggers a sequence of communications and transactions designed to facilitate the parking process swiftly and efficiently.
2. **Reservation Purchase – Driver/Manager:** In the second scenario, the Driver agent expresses the intention to reserve a parking spot for future use. The process involves purchasing the reservation, followed by the issuance and blockchain-based transfer of a NFT that serves as a digital proof of ownership and validation of the reservation.
3. **Reservation Usage:** The third scenario addresses one of the two possible actions a Driver can take after acquiring a parking reservation within the MAS. In this case, the Driver uses the previously obtained NFT as proof of reservation to access the parking spot.
4. **Reservation Sale and Purchase – Driver/Driver:** The final scenario explores the second use case of a reservation NFT: resale between Drivers. This scenario simulates a peer-to-peer transaction in which one Driver sells their reservation to another interested Driver. This use case clearly demonstrates the advantages of blockchain, particularly in enabling secure and transparent transactions without the need for intermediaries.

4.5. Negotiation

Negotiations within the developed MAS occur in a hybrid manner, divided into two distinct modalities according to the context. The first modality encompasses centralized scenarios in which the Driver intends to either use a parking spot immediately, reserve one through the Manager, or utilize an existing reservation. In these cases, negotiations are centralized, as the Manager holds exclusive authority to create and transfer NFTs that represent parking rights. Therefore, the Manager acts as an indispensable intermediary in the transaction process.

The second modality is decentralized and emphasizes direct interactions between Driver agents, specifically in the resale of previously acquired reservations. Although only the Manager can issue NFTs directly associated with the parking system, once a Driver obtains such an NFT, they gain the ability to engage in peer-to-peer transactions within the secondary market.

This is particularly evident in the reservation resale scenario (Figure 4), which illustrates how the use of blockchain technology and NFTs enables straightforward, secure, and transparent negotiations. In this process, a Driver holding a valid reservation

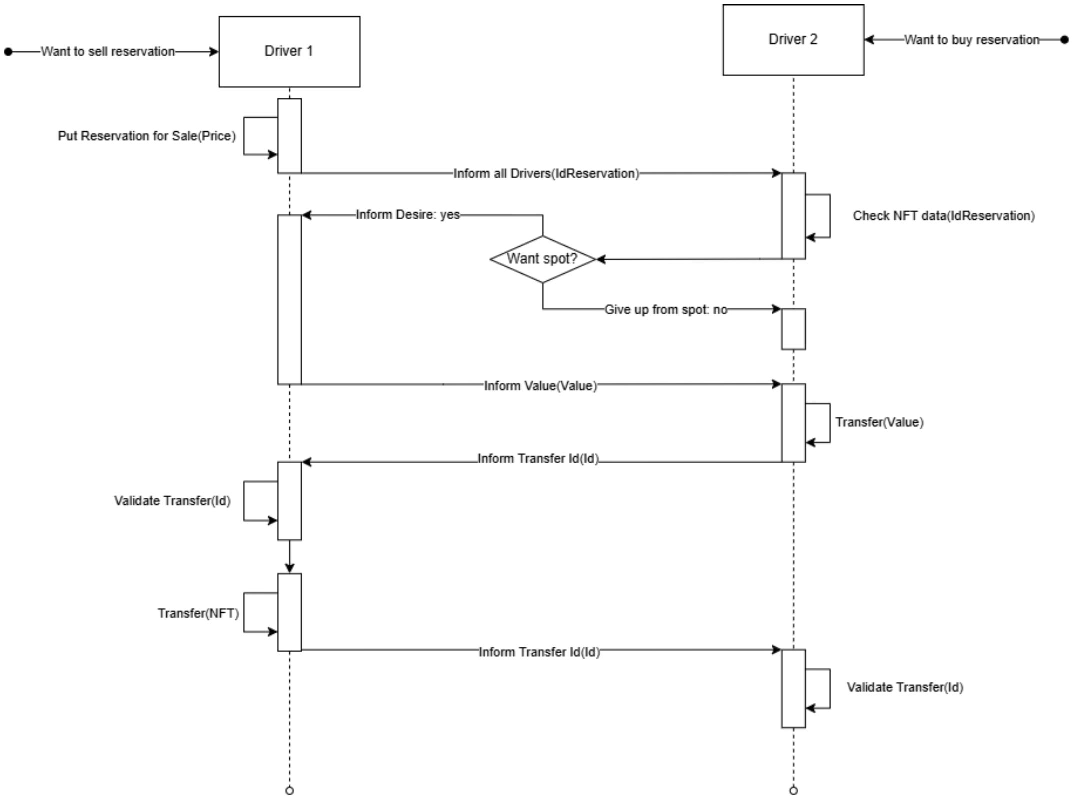


Figura 4. Reservation Resale Diagram.

NFT initiates a sale, while another Driver interested in purchasing it engages directly in the transaction, with no need for intermediaries. The exchange is completed through the validation and transfer of the associated NFT, which serves as a digital certificate of ownership and guarantees both traceability and trustworthiness in the transaction.

5. Results

This section presents the experimental scenarios used to validate the proposed multi-agent system. It includes the configuration parameters and results, focusing on the interactions among BDI agents in a Smart Parking environment, utilizing the Velluscinum middleware and digital assets.

All experiments were conducted within the MAS, monitoring communications and transactions performed by the agents. The execution time was measured from the start of the MAS. Preliminary steps like wallet creation, token generation, and parking slot registration were performed by an external system, available at `https://github.com/GabrielOliveira23/SmartWallets`.

The evaluation considered the total number of messages exchanged between agents, the number of blockchain transactions processed by Velluscinum, the success rate of completed negotiations, and the system's execution time.

Scenarios allowed variations in the number of driver agents, total parking slots (always a multiple of 4, split evenly among the four spot types: Short, Long, Covered Short, Covered Long), the type of connection (local or remote BigchainDB server), and the interaction patterns of the drivers.

Experiments were executed on a machine with an Intel i5-12400F processor, NVIDIA RTX 2060 Super GPU, 16GB RAM, 512GB SSD, and Windows 11 with Ubuntu 22.04.5 LTS via WSL.

5.1. Experiments and Results

Nine experiments (A to I) were conducted, varying the number of *Drivers*, parking *Spots*, and *Connection* types. The results, including messages sent per agent type (*Manager* or *Driver*), *Transactions* via Velluscinum, and the negotiation *Success* rate is presented at Table1.

Tabela 1. Experiments Settings and Results.

Exp	*Drivers*	*Spots*	*Connection*	*Manager*	*Driver*	*Transactions*	*Success*
A	100	100	Local	214	1317	622	30/35
B	100	100	Remote	87	895	559	27/30
C	100	52	Local	190	1262	569	18/24
D	100	24	Local	250	1384	660	20/30
E	50	52	Local	143	678	549	23/32
F	50	52	Remote	81	408	523	28/35
G	30	32	Local	143	411	503	28/31
H	30	32	Remote	84	277	497	32/36
I	30	16	Local	114	322	528	33/42

5.2. Analysis and Comparison

Message volume was influenced by connection type: local connections typically involved more messaging than remote ones, especially in scenarios with many or few agents (experiments A, B, E, F, G, H). Remote connections sometimes resulted in more successful negotiations, as shown in Figure 5.

Figure 6 highlights that driver agents sent significantly more messages than managers, suggesting that increasing the number of drivers impacts system load, potentially

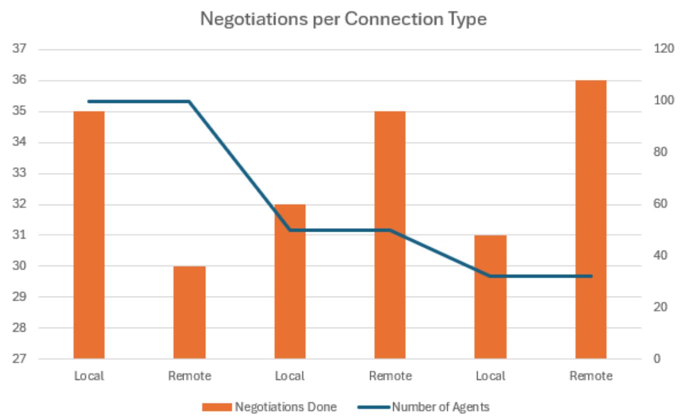


Figura 5. Negotiation x Connection Type.

affecting performance. In experiment D, with 100 drivers and only 24 slots, message volume peaked—likely due to repeated unsuccessful attempts, as supported by the success rates in Figure 7.

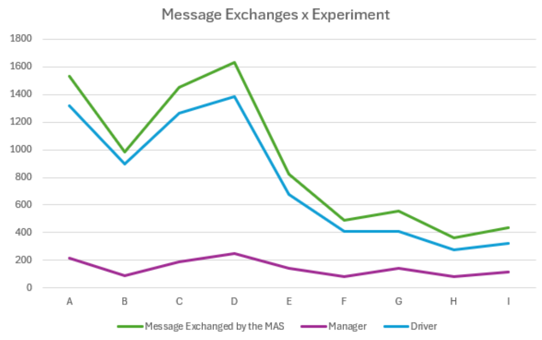


Figura 6. Messages x Experiment.

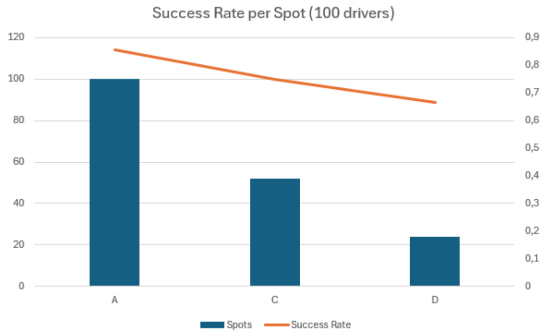


Figura 7. Success Rate in Negotiations.

Comparing experiments A, C, and D reveals that success rates are tied to slot availability. The higher the competition among drivers, the lower the chance of successful negotiation. For instance, experiment A (100 drivers, 100 slots) had the highest success rate (0.85), while experiment D (100 drivers, 24 slots) had the lowest (0.66).

6. Conclusion

This work aimed to develop a BDI-based multi-agent system integrated with blockchain technology, using the Velluscinum middleware to enable decentralized negotiations of digital assets in a Smart Parking environment. To achieve this, several goals were set: analyzing digital asset negotiations in similar systems, defining negotiation mechanisms compatible with digital assets, implementing a BDI architecture within those mechanisms, comparing negotiation protocols, and validating the implemented system through testing.

The results indicate that these goals were largely met. The system functioned as intended, allowing BDI agents to interact with digital assets in a decentralized setup. The experiments confirmed the system's performance in simulated scenarios, supporting its potential application in real-world smart parking management.

However, some limitations emerged. One was the centralization of asset management in a single agent, which created bottlenecks in scenarios with many agents—as seen in Table **??**. Another was the system's reliance on a stable network connection for blockchain transactions, which could affect performance during connection instability.

While digital asset negotiations are decentralized, the MAS itself is not—only the Manager agent is allowed to create and manage NFT representing parking slots. This centralization led to performance issues in complex scenarios, where the Manager became a bottleneck. This was evident in Section 5, where increasing the number of Driver agents did not proportionally increase successful negotiations.

Also, since the system depends on blockchain connectivity, network instability can delay transactions and negatively affect user experience.

These limitations highlight areas for future improvement. A decentralized MAS architecture could solve scalability issues, while new negotiation methods (such as peak-time slot auctions) could enhance agent communication and enable the simulation of more diverse scenarios.

Smart Parking is just one part of a broader Smart City framework. Integrating this MAS with other urban systems could offer deeper insights into how smart cities operate. Simulations could reveal improvements for urban planning, benefiting cities aiming to adopt smart infrastructure.

Overall, combining BDI multi-agent systems with blockchain through middleware has proven to be a promising strategy for addressing complex decentralized negotiation challenges. This research supports the development of safer, more transparent, and scalable solutions for managing resources in smart cities. Future work can expand the impact of this approach by applying it in new domains and integrating emerging technologies.

Referências

Alves, B. R., Alves, G. V., Borges, A. P., and Leitão, P. (2019). Experimentation of negotiation protocols for consensus problems in smart parking systems. In *HoloMAS 2019*, pages 189–202, Cham. Springer. DOI: 10.1007/978-3-030-27878-6_15.

Boissier, O., Hübner, J. F., and Ricci, A. (2016). The jacamo framework. In *Social Coordination Frameworks for Social Technical Systems*, pages 125–151, Cham. Springer International Publishing. DOI: 10.1007/978-3-319-33570-4_7.

Bratman, M. E., Israel, D. J., and Pollack, M. E. (1988). Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4(3):349–355. DOI: 10.1111/j.1467-8640.1988.tb00284.x.

Calvaresi, D., Calbimonte, J.-P., Dubovitskaya, A., Mattioli, V., Piguet, J.-G., and Schumacher, M. (2019). The good, the bad, and the ethical implications of bridging blockchain and multi-agent systems. *Information*, 10(12). DOI: 10.3390/info10120363.

Calvaresi, D., Dubovitskaya, A., Retaggi, D., F. Dragoni, A., and Schumacher, M. (2018a). Trusted registration, negotiation, and service evaluation in multi-agent systems throughout the blockchain technology. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, pages 56–63. DOI: 10.1109/WI.2018.0-107.

Calvaresi, D., Mattioli, V., Dubovitskaya, A., Dragoni, A. F., and Schumacher, M. (2018b). Reputation management in multi-agent systems using permissioned blockchain technology. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, page 719–725. DOI: 10.1109/wi.2018.000-5.

Kumar, K. P. V., Lakshmi, B., Sai Kumar, S., V, M., Sai, N. R., and Nagamalleswara, V. (2023). Blockchain technology: A comprehensive review and future directions. In *ICAISS 2023*, pages 1362–1368. DOI: 10.1109/ICAISS58487.2023.10250707.

Lazarin, N., Pantoja, C., and Viterbo, J. (2023). Towards a Toolkit for Teaching AI Supported by Robotic-agents: Proposal and First Impressions. In *Proceedings of the 31st WEI*, pages 20–29, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/wei.2023.229753.

Liu, S., Hennequin, S., and Roy, D. (2021). Enterprise platform of logistics services based on a multi-agents mechanism and blockchains. *IFAC-PapersOnLine*, 54(1):825–830. DOI: 10.1016/j.ifacol.2021.08.097.

Luo, F., Dong, Z. Y., Liang, G., Murata, J., and Xu, Z. (2019). A distributed electricity trading system in active distribution networks based on multi-agent coalition and Blockchain. *IEEE Transactions on Power Systems*, 34(5):4097–4108. DOI: 10.1109/tpwrs.2018.2876612.

McConaghy, T., Marques, R., and Müller, A. (2016). Bigchaindb: a scalable blockchain database. Available at: `https://scholar.google.com.br/scholar?cluster=8300213592391874791`.

Mhamdi, H., Soufiene, B. O., Zouinkhi, A., Ali, O., and Sakli, H. (2022). Trust-based smart contract for automated agent to agent communication. *Computational Intelligence and Neuroscience*, 2022:1–11. DOI: 10.1155/2022/5136865.

Minarsch, D., Favorito, M., Hosseini, S. A., Turchenkov, Y., and Ward, J. (2021). Autonomous economic agent framework. In *EMAS 2021*, page 237–253, Berlin, Heidelberg. Springer-Verlag. DOI: 10.1007/978-3-030-97457-2_14.

Mori Lazarin, N., Machado Coelho, I., Pantoja, C. E., and Viterbo, J. (2023). Velluscinum: A middleware for using digital assets in multi-agent systems. In *PAAMS 2023*, pages 200–212, Cham. Springer. DOI: 10.1007/978-3-031-37616-0_17.

Oliveira, G., Alves, G., and Lazarin, N. (2023). A mapping of multi-agent systems and digital assets with an application proposal in smart parking. In *WESAAC 2023*, pages 32–43, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/wesaac.2023.33435.

Papi, F. G., Hübner, J. F., and de Brito, M. (2022). A blockchain integration to support transactions of assets in multi-agent systems. *Engineering Applications of Artificial Intelligence*, 107:104534. DOI: 10.1016/j.engappai.2021.104534.

Rajasekaran, A. S., Azees, M., and Al-Turjman, F. (2022). A comprehensive survey on blockchain technology. *Sustainable Energy Technologies and Assessments*, 52:102039. DOI: 10.1016/j.seta.2022.102039.

Rao, A. S. (1996). AgentSpeak(L): BDI agents speak out in a logical computable language. In *Agents Breaking Away*, Berlin. Springer. DOI: 10.1007/BFb0031845.

Sichman, J. S., Demazeau, Y., and Boissier, O. (1992). When can knowledge-based systems be called agents? In *Proceedings of the IX SBIA*, pages 172–185, Rio de Janeiro. SBC. Available at: `https://scholar.google.com.br/scholar?cluster=12989357293789094729`.

Smith (1980). The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113. DOI: 10.1109/TC.1980.1675516.

Song, Y., Gao, X., Li, P., and Yang, C. (2022). Resilience Network Controller Design for Multi-Domain SDN: A BDI-based Framework. In *VTC2022-Spring*, pages 1–5. DOI: 10.1109/VTC2022-Spring54318.2022.9860710.

Souza De Castro, L. F., Vaz Alves, G., and Pinz Borges, A. (2017). Using trust degree for agents in order to assign spots in a smart parking. *ADCAIJ*, 6(4):5–14. DOI: 10.14201/ADCAIJ201764514.

Wooldridge, M. J. (2009). *An introduction to multiagent systems*. John Wiley & Sons, Chichester, U.K, 2nd ed edition.