

## Article

# Leveraging Multi-Agent Systems and Decentralised Autonomous Organisations for Tax Credit Tracking: A Case Study of the Superbonus 110% in Italy

Giovanni De Gasperis <sup>1,\*</sup>, Sante Dino Facchini <sup>1,\*</sup> and Ivan Letteri <sup>2,†</sup>

<sup>1</sup> Department of Information Engineering, Computer Science and Mathematics, Università degli Studi dell'Aquila, Via Vetoio snc Coppito, IT67100 L'Aquila, Italy

<sup>2</sup> Department of Medicine SVA, Università degli Studi dell'Aquila, Piazzale S.Tommasi 1, IT67100 L'Aquila, Italy; ivan.letteri@univaq.it

\* Correspondence: giovanni.degasperis@univaq.it (G.D.G.); santedino.facchini@graduate.univaq.it (S.D.F.)

† These authors contributed equally to this work.

**Abstract:** This study aims to develop a Secured Fiscal Credits Model to address the challenges of managing Italy's "Superbonus 110%" tax credit. Using a decentralised governance approach, our research objective is to provide a feasible system to track and control the entire tax credit process, from generation to redemption. The method integrates Artificial Intelligence and blockchain technology within a Decentralised Autonomous Organisation architecture, combined with a Multi-agent System to establish a tokenomics model. The system is structured to prevent accidental errors, such as double spending or overspending, and detect fraudulent behaviours, like false claims of completed work. Our main findings indicate that deploying two Decentralised Autonomous Organisations on the Algorand blockchain significantly enhances trust and security, supporting effective oversight of the Superbonus process and facilitating transparent value exchange among stakeholders. This decentralised governance model introduces substantial automation, reduces biases, and offers a viable solution to strengthen tax credit management. This work proposes an innovative, technology-driven framework that can be generalised to similar fiscal and governance contexts, enhancing transparency and control.

**Keywords:** decentralised autonomous organisation; multi-agent systems; blockchain; tax credit tracking; tax relief measure; secured fiscal credits



**Citation:** De Gasperis, G.; Facchini, S.D.; Letteri, I. Leveraging Multi-Agent Systems and Decentralised Autonomous Organisations for Tax Credit Tracking: A Case Study of the Superbonus 110% in Italy. *Appl. Sci.* **2024**, *14*, 10622. <https://doi.org/10.3390/app142210622>

Academic Editor: Seung-Hoon Yoo

Received: 20 September 2024

Revised: 6 November 2024

Accepted: 11 November 2024

Published: 18 November 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Italian energy policies and regulations, heavily influenced by European Union guidelines, focus on enhancing energy performance and reducing energy waste in residential systems. Over the past few years, key measures implemented by the Italian government involved incentives for energy-efficient building renovations, becoming central to determining the Next Generation EU funding policies ([https://energy.ec.europa.eu/topics/energy-strategy\\_en](https://energy.ec.europa.eu/topics/energy-strategy_en), accessed on 10 November 2024).

As an EU Member State, Italy has implemented various measures to encourage increased energy efficiency in residential buildings. A tax deductions and fiscal bonus program, initially introduced in 1997, was relaunched in 2011 and extended until 2019. In 2020, Italy introduced the Superbonus 110% tax credit system (Article 119 of Decree-Law 34/2020) (<https://www.gazzettaufficiale.it/eli/id/2020/05/19/20G00052/sg>, accessed on 10 November 2024), enabling residential owners to undertake energy efficiency upgrades and structural improvements to existing buildings without any upfront costs through a tax credit mechanism.

Next Generation EU funds, allocated in response to the COVID-19 crisis, support recovery and growth across EU member states. Italy's Decree-Law 34/2020, introducing

the SuperBonus 110%, aimed to stimulate the economy by offering 110% tax credits for energy-efficient renovations. Applicability measures included broad eligibility criteria and streamlined digital platforms for applications. However, challenges arose, including bureaucratic delays, high demand outpacing supply chains, and potential fraud due to complex procedures. The Italian government later tightened controls and adjusted eligibility to enhance transparency, but implementation remains challenging due to administrative burdens and delays in reimbursements.

This research develops the Secured Fiscal Credits Model (SFCM) to propose innovative methods for enhancing tax credit policies, aimed at addressing the challenges encountered. This study integrates Artificial Intelligence (AI) and blockchain technology within a Multi-agent System (MAS) to optimise the management of tax credits among a diverse set of stakeholders. The SFCM is designed to rigorously oversee the Superbonus 110% process, ensuring comprehensive control and traceability to mitigate risks including double spending, overspending, and fraudulent claims on completed work. By leveraging Decentralised Autonomous Organisation (DAO) and blockchain technologies, the SFCM offers high trust and security in a competitive environment prone to potential fraud. This decentralised model ensures efficient management of value exchange among actors with a high degree of automation, thereby mitigating bias.

The Superbonus 110% initiative presents a considerable opportunity for citizens and serves as a valuable stimulus for national economic recovery. However, it also introduces challenges, such as speculative practices, identity fraud, and potential for credit misallocation to fraudulent claimants [1]. For instance, clients may request a higher amount than they are owed or claim more properties than permitted.

The SFCM decentralised governance structure enables comprehensive monitoring across the entire Superbonus 110% life cycle, supporting the efficient management of value exchange among actors. This framework maintains the stability of each token within the Superbonus circuit and ensures the precise control and tracking of all transactions. This robust system ensures that the process is secure, transparent, and free from fraudulent activities, thereby enhancing the overall integrity of the tax credit mechanism.

This proposal builds on the integration of Artificial Intelligence (AI) and blockchain technologies within a decentralised platform for tax credit management, drawing inspiration from the SingularityNET Project (<https://singularitynet.io> accessed on 10 November 2024), which allows the implementation of a decentralised marketplace for AI services interconnected through a blockchain-based network [2]. SingularityNET aims to democratise access to AI services and capabilities, enabling developers to publish and monetise their AI services on a globally accessible network. On the other hand, our project targets the entire Superbonus 110% process with a practical system to monitor and manage the controls. Furthermore, our innovative approach leverages AI and blockchain technology to achieve stability in the value of each token introduced in the Superbonus circuit with a control based on a DAO architecture. This architecture is simulated in a MAS environment to track each token transaction. We aim to achieve stability in the value of each token introduced in the Superbonus circuit to induce a 1:1 correspondence to the underlying fiat currency without actually exchanging any amount, allowing only the use of tokens among members of the DAO. Each token transaction is controlled and tracked to certify and trace all spending on services between parties within the included intelligent contracts, which are natively implemented in a blockchain context.

Our approach seeks to answer the following questions: (i) How can the integration of Multi-agent Systems with blockchain technology reduce operational inefficiencies in tax credit management? (ii) In what ways can DAOs contribute to minimizing fraud and double spending in the Superbonus 110% program? (iii) How can an automated, blockchain-based tracking model enhance transparency and trust among stakeholders?

By integrating the MAS with two DAOs, we implement token minting and transfer of tax credits among the operators. The synergy between DAOs and MASs enables highly automated, unbiased management of value exchange among users through decen-

tralised governance without requiring complete trust among users. As shown in [3,4], a MAS has been demonstrated to monitor the ethical behaviour of dialogue systems in different domains.

This study makes several contributions to both the literature on blockchain and economic practice in tax management: (i) introduces a novel DAO-based model for fraud-resistant tax credit tracking, enhancing transparency and accountability in tax incentive programs; (ii) demonstrates the application of Multi-agent Systems for managing complex tax credit transactions, contributing to the broader literature on blockchain governance in economic applications; (iii) provides a practical framework for reducing administrative overhead in tax credit management, which can be applied in other governmental and fiscal programs; and (iv) offers insights into the economic implications of blockchain-based tax credit systems, particularly in reducing operational costs and improving stakeholder trust through secure, decentralised mechanisms.

The present work is an extension of our previous publication [5] presented to the PAAMS 2022 Conference and as the evolution of the abstract presented to Doctoral Consortium of IJCAI 2022 [6].

The structure of this paper is organised as follows: Section 2 outlines the background and related works. Section 3 details the methods employed, followed by an exploration of the considered scenarios and the system design in Sections 4 and 5, respectively. Finally, Section 6 presents the conclusions and directions for future works.

## 2. Background and Related Works

Several recent studies have explored the role of Superbonus 110% in Italy, highlighting both the benefits and critical issues that have emerged from the implementation of this policy. The analysis by Huerto-Cardenas et al. [7] demonstrates the effect of the Superbonus in terms of the energy efficiency improvement and economic impact but also the social challenges, including inequalities in access to incentives and an increase in construction costs. According to Codogno [8], the Superbonus caused a significant strain on public finances and a distortion in the distribution of resources, illustrating the need for a review of fiscal policies during economic crises.

### 2.1. Superbonus 110%

In this section, we describe the Superbonus 110% program and the procedure for claiming the invoice discount and the accrued credit from taxable income. According to current legislation, these benefits can be obtained by submitting specific documentation to demonstrate compliance with the requirements and the proper execution of the works.

This documentation includes an energy performance certificate, a statement that the construction work meets specific technical standards and an asseveration of the accounting and bookkeeping processes throughout the project. Furthermore, the execution of the works must adhere to particular requirements and procedures. The overall project must enhance the building's energy efficiency and improve its structural response to earthquakes.

#### 2.1.1. The Legislation

Superbonus 110%, unlike other tax allowances, initially provided for a higher-than-investment rate of deduction and a different way of allowances and claims.

The legislation stipulates that for every EUR 100 spent on renovation works, the house owner will be entitled to a EUR 110 tax deduction for the next five years, divided into five equal annual instalments. Since the yearly deduction is discounted from the taxpayer's gross tax (IRPEF), it is recoverable within the limit of such an amount. It cannot be carried forward or claimed back. Any deductible allowance exceeding the gross tax would be lost. A thorough and updated source of info on the Superbonus 110% regulation can be found on the Agenzia delle Entrate Italian website (<https://www.agenziaentrate.gov.it/portale/superbonus-110%25>, accessed on 10 November 2024). The site contains a news section

with recent updates on Superbonus, an informative section, and a collection of the laws approved on the specific sector.

Since 2024, Italian law has changed the overall Superbonus model due to a shift in the politically oriented government, but our system can be smoothly adapted and implemented as required.

### 2.1.2. Discount on the Invoice and the Accrued Credit

The legislator has provided an alternative solution to direct deduction, whereby a discount can be applied to the invoice received for works. Specifically, it is possible to transfer or sell the accrued credit to a third party, thereby enabling the customer (i.e., the house owner) to automatically transfer the deductions gained through interventions allowed by the Superbonus 110% scheme, even in cases where there is insufficient tax liability to offset. Furthermore, the customer may opt for a discount on the invoice totalling 100% of the amount due, which is applied directly by the supplier or General Contractor and corresponds to the maximum amount payable for the works, including VAT. In this scenario, the customer would transfer the tax credit, equivalent to 110% of the invoice value, to the issuer.

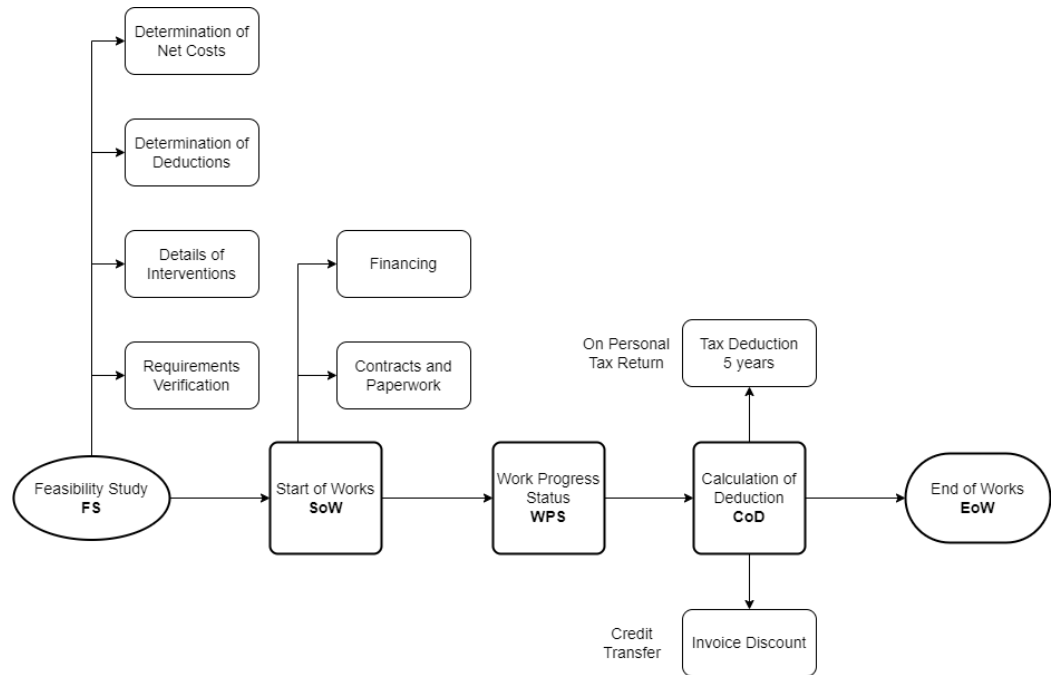
For the supplier/General Contractor, the benefits of this alternative solution are significant. They can set off the discounted invoice applied to the purchaser as a tax credit or transfer it to other financial intermediaries or credit institutions. In addition, they will have an extra 10% calculated on the discounted invoice amount. This additional amount, estimated by the government to cover financial costs related to the overall project, further enhances the economic benefits for the supplier/General Contractor.

The *Agenzia delle Entrate*, to apply the Superbonus 110%, refers to the contract signed between the customer (the taxpayer commissioning the work) and the construction company (General Contractor). The prices for such construction work are determined in the official price lists fixed quarterly in each region, the so-called *Prezziari*.

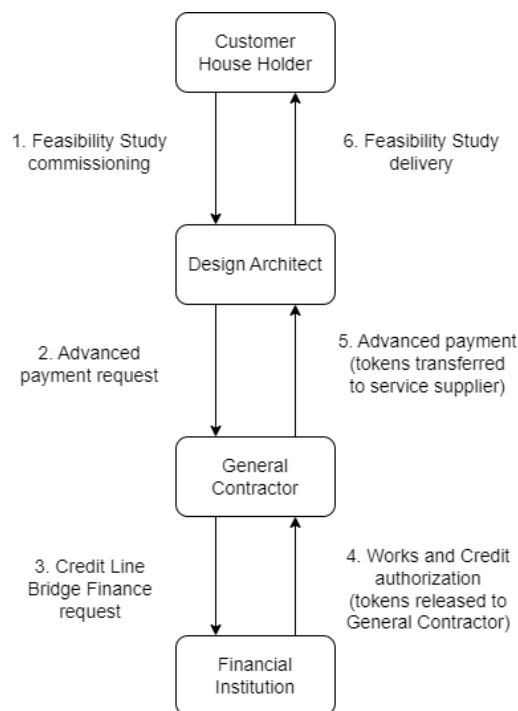
### 2.1.3. The Procedure

There are four crucial stages in implementing interventions under the Superbonus 110% scheme. First, the feasibility study determines whether the property is eligible. Second, the construction work starts after obtaining permits (Start of the Work). Third, deductions are calculated to determine cost savings. Finally, the closing of the work completes the process. Figure 1 shows the primary steps and their respective sub-options. An example of the credit request workflow, which involves the relations with the actors, is exposed in Figure 2, while the detailed distribution and timing of works steps are shown in Figure 3.

- Step 1—The feasibility study (FS) consists of three steps: (i) verification that the envisaged intervention qualifies for relief such as building permits and urban planning compliance; (ii) detailing of the hypothesised interventions and calculation of the deductions due (e.g., estimated metric calculation and detailed estimates); and (iii) determination of the net investment cost (i.e., actual expenditure credit).
- Step 2—The Start of the Work (SoW) represents the final balance of the budget in the feasibility study. There are two scenarios: (i) the client who has the financial means, and can therefore afford, to wait until the end of the work to accrue the tax credit; (ii) the client who needs to finance the works with the accrual of the tax credit occurring in the steps during the execution of the works (Work Progress Status). This second option is the one that happens most of the time. The General Contractor usually manages to set up the bridge finance with banks/financial institutions.
- Step 3—Work Progress Status (WPS) is the accounting act functional to the payment of the work completed until that moment; it summarises all the works and all the supplies carried out from the beginning of the contract up to the day of issue. A copy of the lists of prices (*Prezziari*) is attached to the WPS *amassimali*.



**Figure 1.** Time-sequenced phases for the acquisition of Superbonus 110%: (i) Starting with the FS by assessing costs, deductions, interventions, and verification. (ii) SoW definition of funding and contracts of the project. (iii) WPS paperwork and reports on the progress status of building interventions. (iv) CoD tracking and calculating payments to be deducted as tax credits. (v) EoW closing of all works and asseverations, with a record of the intervention and archiving of documents.

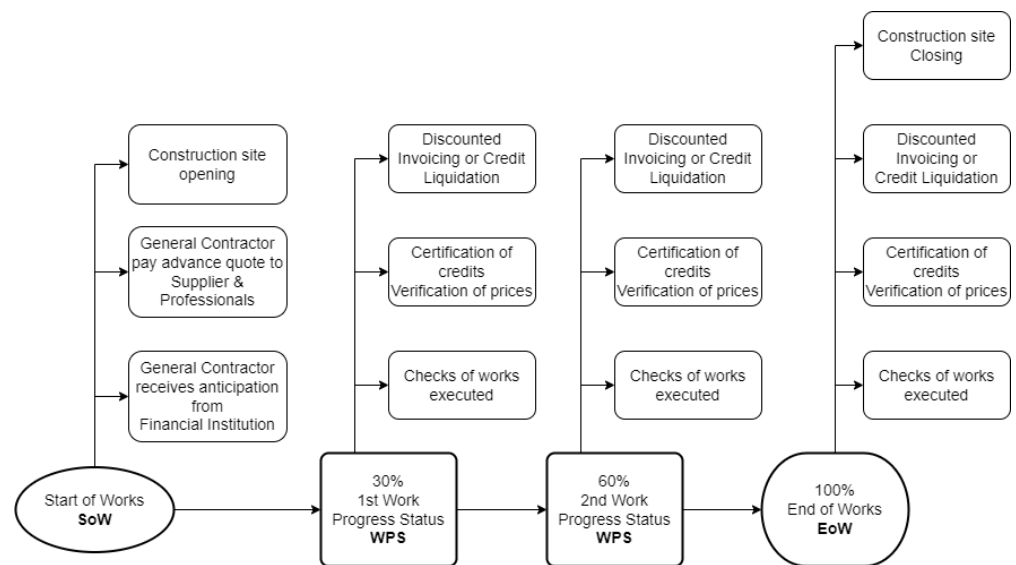


**Figure 2.** Example of interaction between the actors involved in the credit request process. Boxes represent the actors, while the arrows have numbered actions that connect the ordered relations and/or documents to be created between them.

- Step 4—Calculation of Deduction (CoD) The annual deduction is recoverable up to the limit of the personal gross tax (IRPEF). There are two possible scenarios: (i) if the client has sufficient gross tax to absorb the annual deduction, it will be recovered

in the tax return; (ii) if the tax is insufficient, the client may opt for the discount on the invoice. For example, the supplier will request a contribution in the form of a discount on the invoice up to a maximum amount equal to the consideration. Then, the supplier recovers it by accruing a credit with the Tax Agency. This credit will be used for offsetting or assigned to third parties (i.e., the deduction due is transferred to third parties (banks, insurance companies, post office, intermediary, other companies, and individuals).

- Step 5—End of the Work (EoW) All the works are checked and tested, and a final balance of the intervention is drafted. Design Architects and Tax Auditors certify expenses and congruity on the technical and fiscal sides. Once all these activities are performed, the documents are deposited to the Italian Tax Agency, which may check them within seven years.



**Figure 3.** Typical Work Progress Status distribution. The lower boxes represent the phases of work. The upper boxes are the actions/sub-phases requested for each principal phase of the workflow.

As shown in Figure 3, there are three WPSs: (i) the first is stated at 30% of work, (ii) the second at 60%, and (iii) the remaining one after the entire work (100%). By calculating the average time to carry out the work, it is possible to project the state of the WPS. Thus, the money required may be reserved over the foreseen period (on average eight months per building site for ecobonus+sisma -bonus (Sisma-bonus regards the building enhancement towards seismic vulnerabilities or six months for single eco-bonus).

### 2.2. The Blockchain

Tax Audit examines records to determine whether a taxpayer has correctly reported its tax liabilities. Karakostas et al. [9] expose the problem of facilitating tax auditing assuming “programmable money”, i.e., digital monetary instruments managed by an underlying Distributed Ledger Technology (DLT).

A DLT is a type of database that keeps multiple copies of information in different nodes in the network, which can be updated consistently. This allows a full copy of the shared ledger, verifiable by the interleaved chain of data blocks and cryptographically signed, maintaining integrity and availability by a protocol of consensus [10,11].

The blockchain, a specific form of DLT, employs chains of blocks to record transactions, ensuring immutability and transparency. Nowadays, blockchain is considered an institutional innovation for the economic coordination of organisations; through a scalable blockchain, it is possible to increase the productivity of existing processes by lowering transaction costs and avoiding costly intermediations [12]. Such crypto-economic systems can

provide an institutional infrastructure that facilitates various socio-economic interactions to influence participants in their behaviour [13].

The Ethereum blockchain [14] made it possible for Turing-complete code pieces named *smart contracts* to be executed on a blockchain as tasks of a virtual processor machine. Smart contracts allow formalising the interaction rules between blockchain transactions and digital workflows to coordinate economic activity and ensure that automated processes run according to the predefined rules and state changes in the DAOs.

### 2.3. Decentralised Autonomous Organisation (DAO)

A DAO is considered a multi-criteria decision-making problem that deals with evaluating a set of possible alternatives [15] to present a decision model to participants that can converge to a deliberation consensus [16].

In this work, we focus on blockchain to scale scenarios on the Superbonus 110% tax credit's tracking previously exposed in [5].

DAOs are organisations controlled entirely by computational algorithms known as smart contracts, which determine the rules by which the parties involved in the DAO cooperate. The idea of DAOs was outlined for the first time during the late 1990s and was connected to the application of MAS to intelligent home sensors [17]. The first transposition of an actual company model on the blockchain was defined with the introduction of the Digital Autonomous Corporations concept (DACs) (see [18]). Later, with the introduction of the bitcoin time-chain [10]—which defines a public ledger with transparent transactions—and, in 2015, the Ethereum blockchain, it allowed the tokenisation of assets and fully automated and incorruptible smart contracts. So, an actual governance process of an organisation became possible [14]. DAOs take on the characteristics of the underlying DLT layers, including decentralised control, security through asymmetric cryptography keys, and the ability for smart contracts to self-execute. Due to their decentralised nature, DAOs are not bound to any particular regulation or law. For these reasons, DAOs are natural candidates to represent real-world companies and complex organisations in blockchain environments [19]. The DAO internal architecture comprises four significant mechanisms, described as follows: (i) smart contracts, (ii) consensus protocol, (iii) tokens, and (iv) blockchain. Combining these four elements makes it possible for a DAO to run correctly at all times.

## 3. Methodology

In this section, we define our methodology based on the Multi-agent System (MAS) scheme, which follows two steps: (i) The definition of a constraints knowledge base to capture domain knowledge, including the objectives and preferences of decision-makers. This knowledge base also includes data on the available resources, technologies, and regulations. (ii) The designing of agents to represent decision-makers, stakeholders, and other entities involved in the problem. Agents are programmed to communicate with each other and negotiate based on the knowledge base and scenarios. In the implementation section, we explain how the Algorand platform is utilised to simulate interactions between the actors involved in the Superbonus 110% process. In the same section, we outline the implementation of the MAS scheme using the MESA framework, which enables the simulation of complex social systems by creating agents and modelling their interactions.

### 3.1. Constraints Knowledge Base

The Decreto Aiuti n. 50/2022 (<https://www.gazzettaufficiale.it/eli/id/2022/05/17/22G00059/sg>, accessed on 10 November 2024) introduced the rules for calculating the percentage of intervention achieved. When assessing the WPS of 30%, the overall intervention must be considered, including works carried out on the dwelling that does not fall within the scope of the Superbonus 110%, such as design costs, certifications, and other professional fees. The term “total intervention” is intended to define all expenses arising from the general economic framework of the intervention, including the technical costs relating to the services performed in connection with the works carried out on the building.

The following are the six constraints designed:

- Constraint 1. A specific agent (Director of Work) verifies compliance with the original schedule plan. Starting from a monthly forecast, the agent will be able to estimate the maturity of the WPS based on the previously mentioned criterion about the distribution of WPS. This translates into the fact that the possible states of the agents (workflow agent; see Section 5.4) are the following:

$$x_i = \{Open, Anticipation, Sal1, Sal2, Eow, Archived\},$$

or in brief  $x_i = \{Op, Ant, S1, S2, Eow, Arc\}$ . We can, at this point, formalise the following expressions for each value of the state variable:

$$x_i(0) \Leftarrow Op \wedge \neg Ant \wedge \neg S1 \wedge \neg S2 \wedge \neg Eow \wedge \neg Arc$$

$$x_i(1) \Leftarrow \neg Op \wedge Ant \wedge \neg S1 \wedge \neg S2 \wedge \neg Eow \wedge \neg Arc$$

...

$$x_i(5) \Leftarrow \neg Op \wedge \neg Ant \wedge \neg S1 \wedge \neg S2 \wedge \neg Eow \wedge Arc$$

The Knowledge Base (KB) constraint can be, at this point, defined as follows:

$$true \Leftarrow x_i(0) \vee x_i(1) \vee x_i(2) \vee x_i(3) \vee x_i(4) \vee x_i(5)$$

- Constraint 2. Ensure token demand in Operator DAO aligns with Investor DAO forecasts. Based on previous WPSs, it is necessary to check financial expenditures and future needs at the closing of the WPS and thus verify that the expected demand for tokens in the Operator DAO ( $dao_o$ ) does not exceed that forecast in the Investor DAO ( $dao_i$ ). So if  $d_t$  is the demand for the token and  $f_t$  is the forecast need for a token at time  $t$ , we must have:

$$d_t \leq f_{t-1}$$

- Constraint 3. Do not spend more than allocated. Electronic invoices will be the source of the flow of economic transactions; they allow a transparent view of all expenditures in the WPS, so all payments performed ( $p_s$ ) by the workflow agent must not exceed the payments received ( $p_r$ ) as anticipation for that state. Both must be made by traceable means. If  $w_t$  is the state of the workflow agent at time  $t$ ,

$$p_s(w_t) \leq p_r(w_{t-1})$$

- Constraint 4. The homeowner may have at most two concessions per building unit to be refurbished. This translates in the condition that each client ( $c_i$ ) may not have more than 2 workflow agents ( $w$ ) opened:

$$|w(c_i)| \leq 2$$

- Constraint 5. Each project  $i$  must be certified by both technical and financial asseveration  $a$  raised by chartered engineers  $eng$  and accountants  $acc$  to grant the fairness of the expenses to the interventions listed in the WPS. Thus, if engineer  $j$  and accountant  $k$  are hired to asseverate project  $i$ :

$$true \Leftarrow a(eng_j)_i \wedge a(acc_k)_i$$

- Constraint 6. The General Contractor (GC) may have more than one construction site assigned. The maximum limit is determined by the tax credit that may be handled or the value set in the Attestation Organisation Certification (AOC) he has in place (e.g.,



1 million euros works for private buildings renovation category). So if the GC has  $j$  works of value  $w_j$  each:

$$\sum w(GC_i)_j \leq AOC(GC_i)$$

### 3.2. Agents Design: Actors, Roles, and Actions

The agents are the actors who have their roles and execute actions. The first step is identifying the environment and the entities involved in the process. We spotted two main areas: (i) the *investors group*, which includes all actors getting financial burdens and benefits in the process, and (ii) the *operators group* that, in turn, aggregates all actors involved. Table 1 shows, in the DAO column, how some actors may take both groups' properties.

**Table 1.** Relationship between the actors and their affiliation DAOs in comparison to the user actions.

DAO	Actor	Actions
Investors	Investor	Invest money to buy credits and benefits from its selling
Investors Op-erators	Financial institution	Banks that buy/sell credits and lend money to Operators
Investors Op-erators	Customer	House owner sells/transfers tax credits from works
Operators	General Contractor	Manages work, obtains credits from customers, sells to financial institutions
Operators	Sub-contractor	Hired and paid by General Contractor
Operators	Supplier	Hired and paid by General Contractor
Operators	Design Architect	Hired and paid by General Contractor
Operators	Tax Auditor	Hired and paid by General Contractor

In the operators group, we have all the actors with operational functions on the construction site and in the design and consultancy area. They use the Operator's DAO to receive and make payments for goods and services and to exchange and certify documents. In the investors group, we have all the actors with a passive interest in the Superbonus process, professional and non-professional investors and financial entities. They use the Investor DAO to invest their money and make a profit.

The actors on the Investor DAO are only the investors: In this category are all the actors (companies and individuals) interested in investing money in the Superbonus 110%. They deposit fiat money into their client accounts in the financial institution and are compensated with the tokens minted in the investor's DAO.

The actors involved in the Investor and Operator DAOs are the following:

- Customers are the various individuals (companies can not apply to this tax relief program) eligible for the Superbonus. They include natural persons and discuss expenses incurred for energy efficiency measures carried out on individual property units up to a maximum of two.
- Financial institutions (FIs) are the "qualified" entities acting as guarantors of the entire system composed by the DAOs. The FIs are banks or registered financial intermediaries or a company belonging to a banking group that is also registered, as are authorised insurance companies. Their role in our model is to manage both the Investor and Operator DAOs, thus minting, distributing, and burning tokens. They distribute tokens in the Operator DAO and pay back investors at the end of the program by redeeming Investor DAO tokens and giving back fiat money.

The actors involved only in the Operator DAO are the following:

- General Contractor (GC) acts as a construction company, executing part of the works and as coordinator of other suppliers and Sub-contractors. It is also a proxy payer and coordinator for the Design Architects and professionals involved. It is in charge of managing all the paperwork involved in the Superbonus 110, like discounted invoicing and tax redemption and transfer processes; consequently, it is the subject that monitors all the payment processes. As the construction manager, it shall ascertain and record all events and works generating expenses as soon as they occur so that it can at any moment perform the following activities: (i) Issue the progress statements of the works within the deadline fixed in the tender documentation and in the contract, to prepare the paperwork for the advance payments. (ii) Monitor the progress of the works and promptly issue the necessary actions for their execution within the limits of the time and sums authorised. (iii) Communicate, as construction manager, to the financial institutions/investors in all the work progress states and interact with Tax Auditors and Design Architects to certify works and fiscal credits matured.
- Sub-contractor (SC) is the firm carrying out the works, interacts mainly with the General Contractor both for receiving working orders and for payment issues, all invoicing is done to the General Contractor and does not claim credits.
- Supplier is the entity that supplies materials. It deals with Sub-contractors or directly with GCs to procure materials for construction sites.
- Design Architect (DA) are designers such as engineers and architects. It is responsible for producing all the technical material (e.g., reports and drawings) necessary for the intervention to comply with the regulations in force. They assess the technical aspects of the project.
- Tax Auditor (TA). In the event of professional negligence or misapplication of the rules, the professionals and the technicians in charge of the asseverations and issuing the compliance certificates could be liable for the sums unduly used. In this regard, periodic checks are carried out by a Tax Auditor to ensure compliance with the constraints defined in the smart contracts. He also checks the results of financial transactions, including correct invoicing of expenses (Financial asseveration).

In Table 2, we report who/what interacts with the system and with what roles, actions, and related assignments define specific tasks and responsibilities to achieve overall goals and ensure effective execution.

**Table 2.** Roles, actions, and related assignments are interdependent elements that contribute to defining and executing organisational tasks.

Role	Actor	Assignments
CCA	TA	Check and control fiscal and financial aspects
CM	GC	Supervises activities on the construction site
DoW	DA	Control expenditures of the works
CW	TA and DA	Check technical and administrative compliance
SG&S	Supplier, Sub contractor, DA	Supply materials and professional services

- Role 1: Compliance Control Agent (CCA). This role is connected with the activity of the Tax Auditor's actor and comprises all the activities of checking and controlling fiscal and financial aspects. It also certifies all intermediate and final steps of credit maturation and liquidation, and it further controls that other professionals' invoices are correct and adherent to price-list approval.
- Role 2: Construction Manager (CM). The General Contractor (GC) actor coordinates all administrative and operational activities on the construction site. This includes acting as a proxy for payments on behalf of the customer towards all players operating in

the Superbonus process. This vital coordinating role must be considered to run all the processes smoothly.

- Role 3. Director of Work (DoW). The Design Architect (DA) actor has the task of controlling the expenditure connected with the execution of the works through the accurate and timely compilation of the accounting documents, which are public acts to all effects of the law. He also ascertains and registers the facts producing expenditure. The DoW (i) checks if the work is performed according to the project, (ii) transfers the measurements made to the accounting ledger to define the progress of the expenditure. He must pay special attention to guarantee compliance with the Superbonus Designer prescriptions, particularly the laying of materials. As a result, the Director of Works holds accountability for both the quality and outcomes of the work.
- Role 4. Checker Work (CW). During the work and at the end of the construction process, the Tax Auditor and Design Architect actors verify the work's quality and issue a report to be sent to the investor. They also assess the financial and technical aspects of the works.
- Role 5. Suppliers Goods and Services (SG&S). The supplier, Sub-contractor, and designer architect actors are all the subjects that somehow supply materials and professional services to customers and General Contractors. The General Contractor pays suppliers and does not claim tax credits.

#### 4. Scenarios

Obtaining the Superbonus 110% reliefs involves working with technical experts, contractors, bankers, and other professionals to plan and perform the necessary renovation work. In this section, we expose a couple of main scenarios describing the information flow and the actions related to the agents.

##### 4.1. Scenario 1: Homeowner and Technical Expert

Agent  $a_{ho}$  is a homeowner who wants to renovate their home to improve its energy efficiency and reduce energy bills as well as the seismic response of the building to earthquakes. They contact agent  $a_{gc}$ , a General Contractor, to manage all the aspects of the renovation works. Technical expert  $a_{te}$ , qualified to provide advice on the requirements for the Superbonus 110, is then appointed by  $a_{gc}$ . Then,  $a_{te}$  visits the home of  $a_{ho}$  and surveys to identify the work needed to make it more energy efficient and what anti-seismic improvements can be made.  $a_{te}$  then prepares a project outlining the necessary work and estimating the costs.  $a_{ho}$  accepts the project proposal, then General Contractor  $a_{gc}$  applies for financing and credit transfer from their bank and obtains approval. They then contract experienced Sub-contractor  $a_{sc}$  to carry out the renovation work. After  $a_{sc}$  completes the work,  $a_{te}$  conducts a final inspection and issues a certificate of compliance. After these passages,  $a_{gc}$  can invoice and start the discount process to redeem the tax credit accrued. All the sequence interactions among the agents are shown in Figure 4.

Four main user classes are involved in this process: homeowner, technical expert, bank, and contractor (see Figure 5). Each class has its attributes and methods that relate to their role in the process:

- Homeowner: The individual who owns the property that will undergo the energy efficiency and/or seismic renovation. The class has attributes such as name, email, phone, and methods such as `applyForFinancing()` and `signContract()`.
- Technical Expert: The expert who conducts the energy efficiency survey and provides a project proposal for the renovation work. The technical expert class has attributes such as name, email, and phone, and methods such as `conductSurvey()`, `prepareProposal()`, `conductFinalInspection()`, and `issueCertificate()`.
- Bank: The financial institution is financing the energy efficiency renovation. This class has a single method called `approveFinancing()`.

- Contractor: The company or individual who completes the renovation work. The contractor class has attributes such as name, email, phone, and a single method called completeRenovationWork().

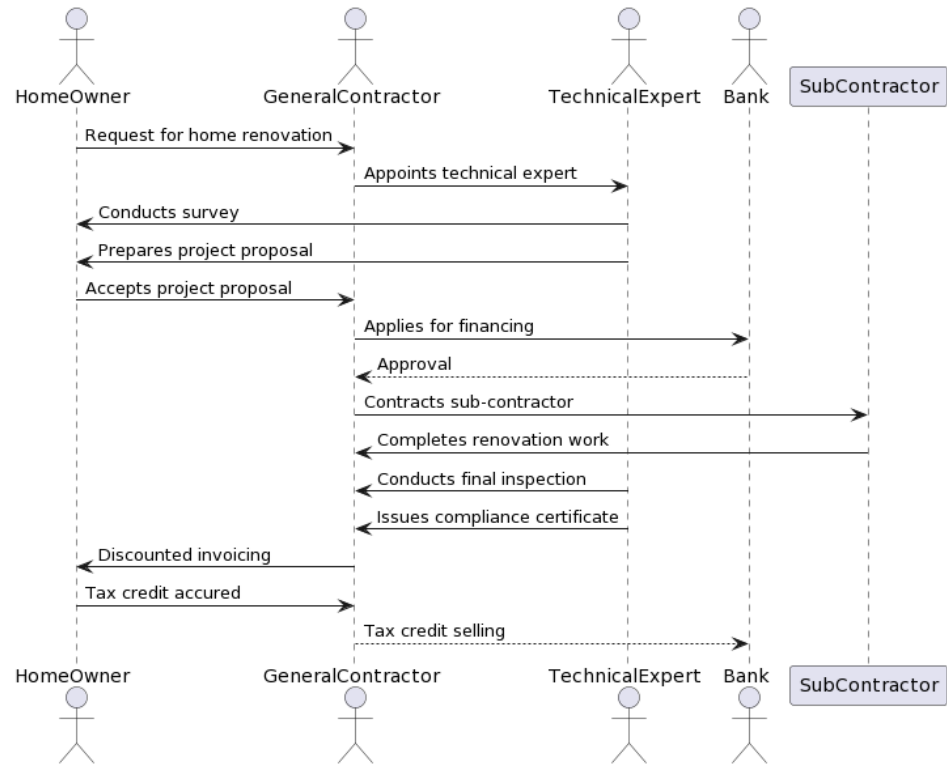


Figure 4. Sequence diagram of Scenario 1. Interactions between the homeowner, technical expert, bank, and contractor in the first scenario are illustrated.

Superbonus 110 Class Diagram

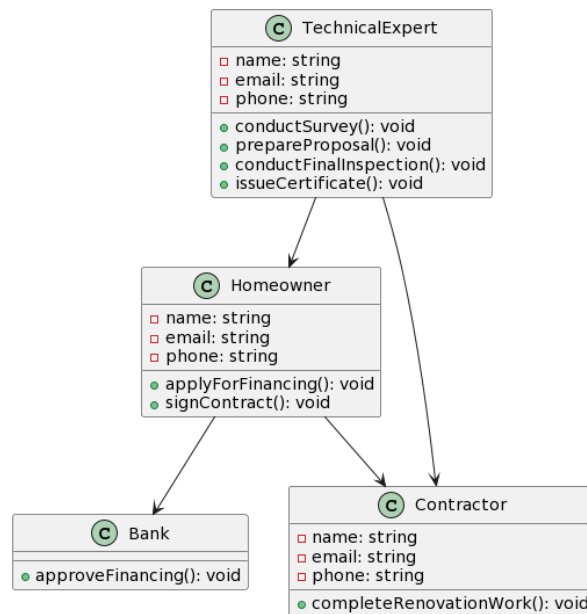


Figure 5. Class diagram of Scenario 1. Shows the classes involved in the first scenario, including the homeowner, technical expert, bank, and contractor. The classes have attributes such as name, email, and phone, as well as methods such as applyForFinancing(), conductSurvey(), and completeRenovationWork(). The interactions between the classes are shown with arrows and labels.

4.2. Scenario 2: Financial Institution, General Contractor and Investor

During the Superbonus 110% reliefs process, the financial institution operators manage the Investor DAO ( $dao_{IN}$ ) and Operator DAO ( $dao_{OP}$ ). In this simplified scenario, a single financial institution  $FI_i$  and a single General Contractor  $GC_i$  are considered.  $FI_i$  is the founder on both DAOs and is the guarantor of the tokens  $t$  minted in the  $dao_{IN}$ . It is responsible for the minting  $\mu(t_{dao_{IN}})$  and distribution  $\delta(t_{dao_{IN}})$  of minted tokens to investors. The  $dao_{IN}$  is a closed and restricted fund until the end of the program, with a fixed number of investors. Each euro of fiat money invested generates one token  $t$  in the  $dao_{IN}$ .

The minting phase  $\mu(t_{dao_*})$  and burning phase  $\beta(t_{dao_*})$  occur in both DAOs, with the freezing phases  $\phi(t_{dao_{IN}})$  and minting phases  $\mu(t_{dao_{IN}})$  in the  $dao_{IN}$ . (i) The  $\beta(t_{dao_{OP}})$  and release  $\rho(t_{dao_{OP}})$  phases occur in the  $dao_{op}$ , where the tokens are redeemed against the  $FI_i$ . (ii) In the  $dao_{IN}$ , the phase of releasing  $\rho(t_{dao_{IN}})$  occurs once the credit has matured, and (iii) the  $FI_i$  can sell the credit on the  $GC_i$ . (iv) Finally, the burning phase occurs in  $dao_{IN}$  at the end of the program when the fiat money, including profits, is returned to the investors. An example of such interactions is described in the following paragraph and corresponding sequence diagrams.

a. Phase of Freeze  $\phi(t_{dao_{IN}})$  and Minting  $\mu(t_{dao_{IN}})$ . The freezing takes place in the Investor DAO. For every 100 tokens requested by the Operator DAO, 95 tokens from the Investor DAO are frozen. With the verified constraint 2, specified in Section 3.1, the link between the two tokens is established, and 95 Operator DAO tokens are minted and assigned to the requesting General Contractor's wallet (see sequence diagram in Figure 6).

b. Phase of Burning  $\beta(t_{dao_{OP}})$  and Release  $\rho(t_{dao_{OP}})$ . Once the tokens in the Operator DAO end their life cycle (e.g., arrive at a supplier that needs to monetise the goods supplied), they can be redeemed against the financial institution. Tokens in the Operator DAO are burned and taken away from the operator wallet; at the same time, tokens previously frozen in the Investor DAO are released and are available for releasing or burning (see sequence diagram in Figure 7).

c. Phase of Releasing in the Investor DAO  $\rho(t_{dao_{IN}})$ . Once the credit has matured via spending and raise of discounted invoicing (see above phases), the financial institution can sell the credit on secondary markets. Once this happens, the FI (let us say EUR 110 credit is sold at EUR 105) releases the 90 tokens frozen in the Investor DAO and mints the exceeding 15 tokens, generating those profit for the investors (see sequence diagram in Figure 8).

d. Phase of Burning in the Investor DAO  $\beta(t_{dao_{IN}})$ . Tokens are burned in the Investor DAO at the end of the program once fiat money, including profits, is given back to the investors (see sequence diagram in Figure 9).

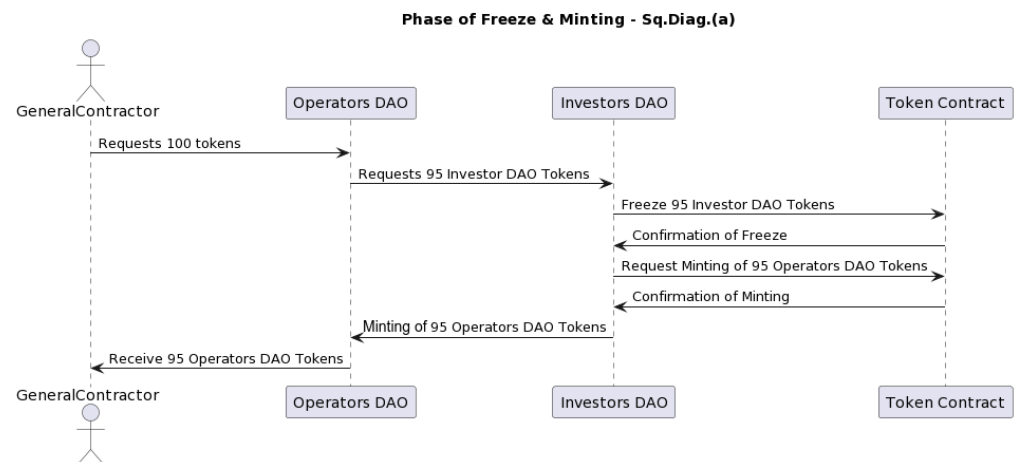
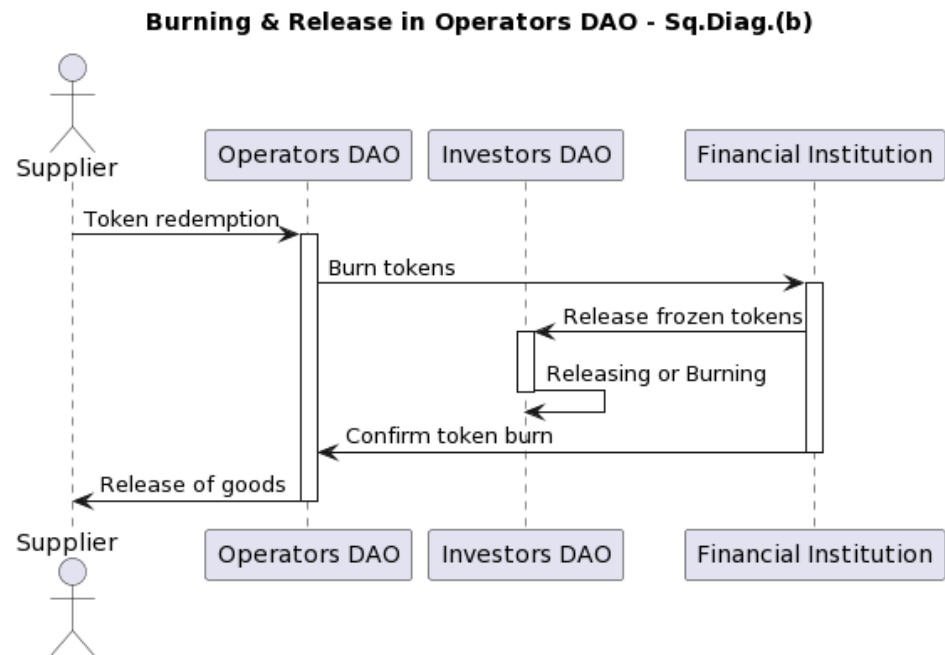
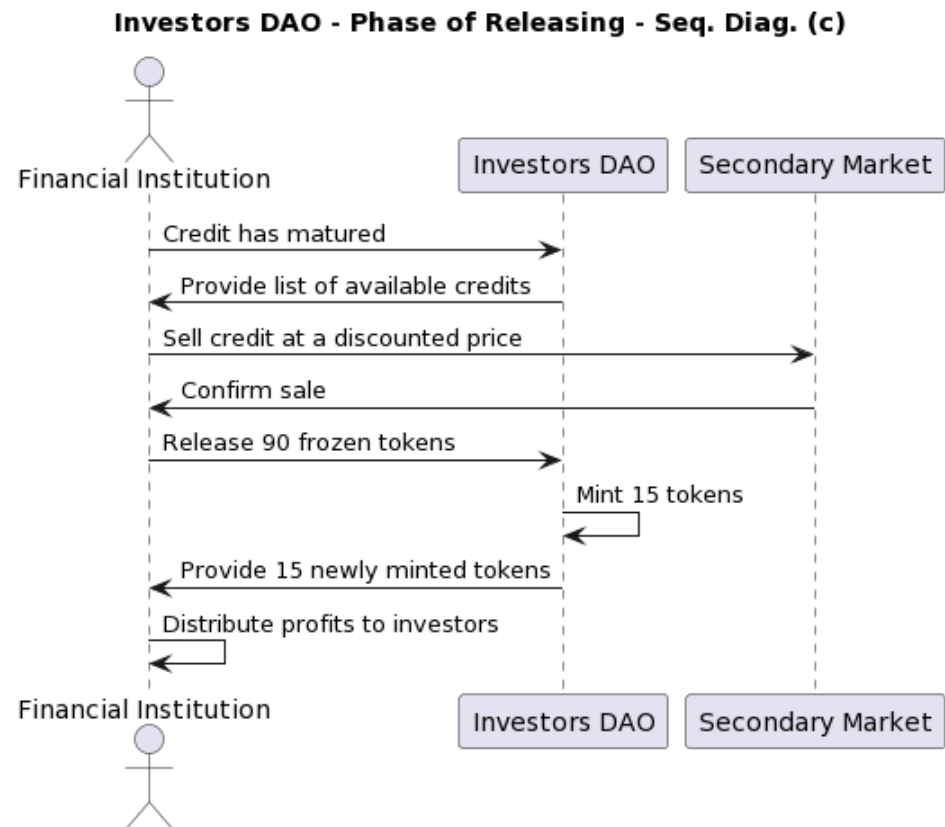


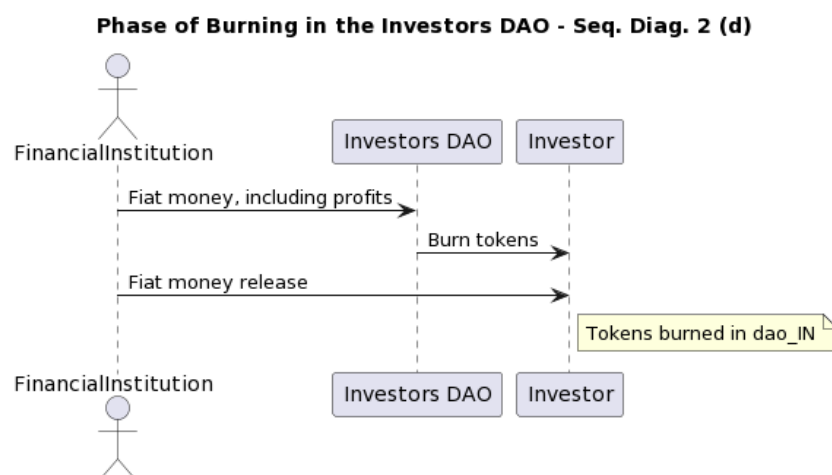
Figure 6. Sequence diagram of Scenario 2. (a) Freeze and minting in the Investor DAO due to a General Contractor request for a new project.



**Figure 7.** Sequence diagram of Scenario 2. (b) Burn and release in the Operator DAO following a fiat liquidation request from suppliers.



**Figure 8.** Sequence diagram of Scenario 2. (c) Example of value generation in the Investor DAO due to tax credit selling.



**Figure 9.** Sequence diagram of Scenario 2. (d) Profit distribution to investors and correspondent Investor DAO token burning.

## 5. The Demonstrator

Our SFCM demonstrator has three pillars. The first is to provide a secure and usable environment, such as the traceability and immutability of transactions; the second is to realise an automated and unbiased control of functionalities; and the third is to achieve the ability to forecast risky situations.

In detail, SFCM offers tracking features for the entire WPS procedure, mitigating risks for Superbonus beneficiaries. Additionally, by implementing the MAS on the Algorand blockchain using Algorand Smart Contracts, we examine secure fiscal credit DAOs. The Investor DAO enables investors to purchase credits and gain from their sales. At the same time, the Operator DAO acts as a marketplace for financial institutions to distribute fungible tokens to General Contractors.

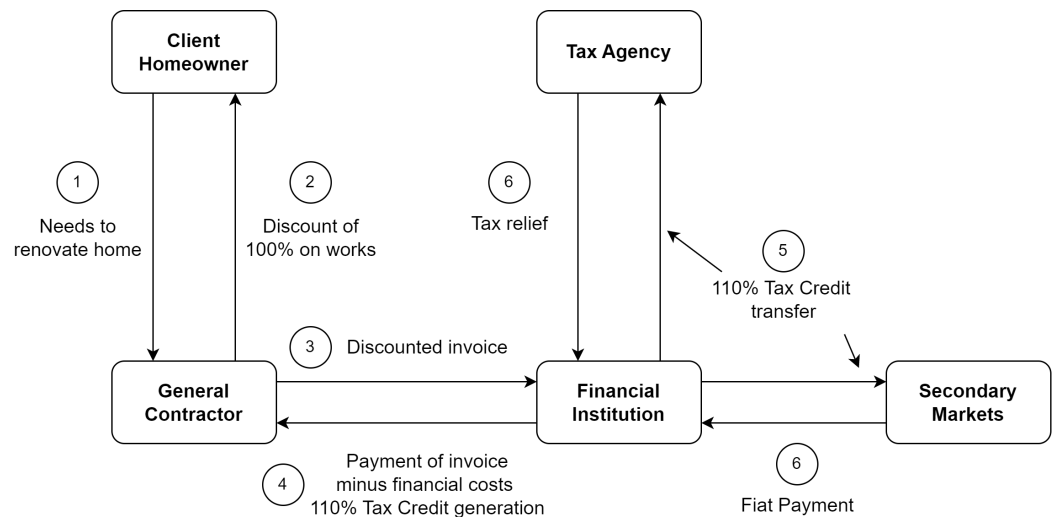
### 5.1. Software Architecture

The tracking functionality of the entire WPS procedure aims to protect against risk situations that the Superbonus beneficiary may incur, such as the liability of professionals, suppliers, and assignees in the case of damage due to professional negligence, inexperience, or the misapplication of rules.

The architecture proposed is based on the separation between the investors' world and the operators' one; this allows financial institutions to raise funds from clients or investors and tokenise such assets. The process is designed assuming a simplified model, where banks receive fiat transfers on their accounts and mint the correspondent value of tokens on the Investor DAO.

Another critical feature is to guarantee that a fiat asset backs money anticipation performed by financial institutions and the correspondent underlying credit generation and is thus repayable when the chain of credit is closed.

The option considered and modelled in this paper is the invoice discount, as it is the most widely used solution. Unlike direct selling or direct deduction, this option requires credits to be tracked and managed because the benefits may be transferred through many subjects during the generation and redemption process. Another assumption is that the customers use a General Contractor to manage the entire renovation process on the construction site and the associated paperwork. This simplifies the model and makes it easily extendable to a multi-supplier situation. Furthermore, we assume to represent the most common scenario in Figure 10. The General Contractor will be the only party responsible for dealing directly with the customer and coordinating and paying all the Sub-contractors, professionals, and suppliers involved in the work.



**Figure 10.** Cycle of the invoice discount tax credit generation. At time 5, the financial institution may decide either to redeem the credit or to use it to pay taxes to the tax agency.

The system architecture implemented following the preliminary assumptions consists of three main blocks interacting with each other:

- A Multi-agent System consisting of peer-to-peer connected nodes, onto which a decentralised and distributed application based on blockchain technology is implemented. It is deployed on Algorand (as outlined in Section 5.6). The system is designed using the Mesa Framework (<https://mesa.readthedocs.io/stable/>, accessed on 10 November 2024) an agent-based modelling framework based on Python. This framework provides a modular environment for system visualisation and testing.
- Two modules consisting of two DAOs based on the Aragon framework (see Section 5.3) used to develop the system’s business logic. Aragon makes it possible to transparently manage the communication between the dApp and the Ethereum Virtual Machine.

### 5.2. Secured Fiscal Credit DAOs

The Investor DAO allows investors to liaise with financial institutions interested in purchasing and resale credit. The General Contractor sends invoices to the customer for the work at the end of the WPS. Therefore, the General Contractor interfaces with the Investor DAO, becoming the collector of all credits accrued within the scope of the work. The financial institution regulates all monetising because it receives fiat currencies from investors and must guarantee the transparency of financial transactions and the licence to collect funds. At the end of the investment cycle, there are  $n$  investors who have issued fiat currencies and received, in proportion to the investment,  $n$  tokens.

In other words, this DAO is meant to represent a simplified investing fund where participants receive tokens proportionally to their investment, keep them blocked until the end of the fund duration, and obtain a reward proportional to their initial quota at the end. Here, we have fungible tokens minted upon fiat deposit or credit selling by the financial institution, freezing or unfreezing when a guarantee from the Operator DAO is requested or released.

The investors group includes all actors receiving financial burdens and benefits in Superbonus 110%. The investors belong to the investors group, which invests money to buy credits and benefits from their selling. In contrast, all the other operators, General Contractors, Sub-contractors, suppliers, Design Architects, and Tax Auditors belong to the operators group, which is hired and paid by the General Contractor.

The gain of the investors is proportional to their quote in the DAO. Let us assume  $X$  as a set of investors  $x_i \in X$ , with  $|X| = n$ . The investor’s share  $sh_{x_i}$  is equal to the sum of tokens  $\sum t_{dao_{1N}}$  of  $x_i$  at the beginning of the process. Given  $T'$ , the sum of tokens in the



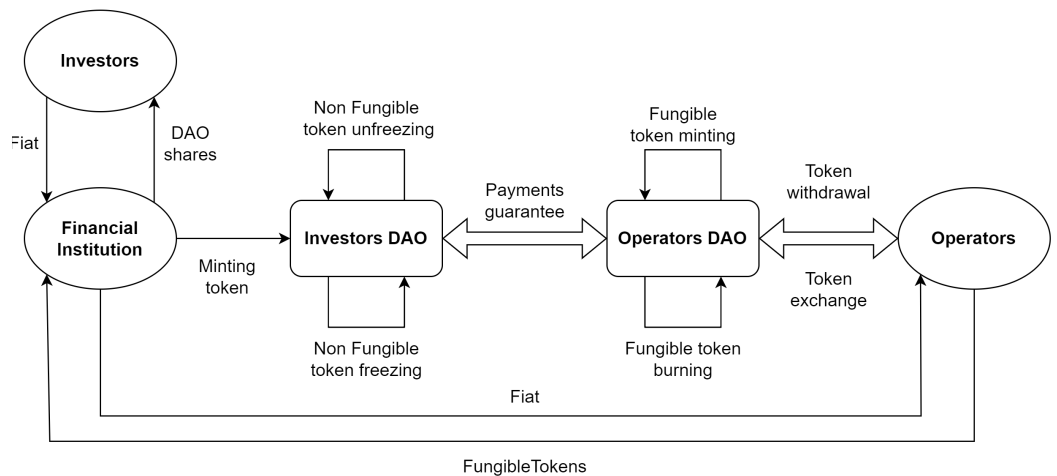
Investor DAO at the starting time  $t_0$ , and  $T''$ , the sum of tokens in the Investor DAO at the end time  $t_n$  of the Superbonus procedure, we define earn  $E_{x_i}$  for an investor to be equal to

$$E_{x_i} = (T'' - T') \times \frac{sh_{x_i}}{\sum_{i=1}^n sh_{x_i}},$$

where  $T'' - T'$  is the total earning in term of tokens, and  $\frac{sh_{x_i}}{\sum_{i=1}^n sh_{x_i}}$  is the percentage of share held by  $x_i$  at time  $t_0$ .

The second DAO is a marketplace for financial institutions to distribute fungible tokens to General Contractors; these are minted and burned to satisfy operator credit requests and are spent on buying goods and services. Each minting on the Operator DAO corresponds to a token freezing of the same amount on the first DAO to guarantee coverage. The tokens are redeemable by the financial institution, the guarantor of the whole system. The first DAO NFTs are unfrozen once the second DAO tokens are redeemed. Each fungible token minted here is coupled via a unique code to the tax credit generated by that spending of money. A scheme of interaction among DAOs and actors is shown in Figure 11.

The *community* that makes up our system consists of the actors listed in Section 3.2. Their interaction initiates the *token distribution* in DAOs operating as a closed fund. In particular, as a first step, the Investor DAO will collect all the fiat funds from investors and convert them into local tokens. At this moment, all relevant financial information and the duration of the operation will be connected to the end of government incentives. The assigned *governance tokens* are used as voting power rights in the DAO, and minting tokens means creating tokens, either from scratch at the beginning of the system or increasing the supply to add more later with credit transactions.



**Figure 11.** DAOs interaction diagram. The architecture can be seen as a Distributed Trust and Reputation Management System (DTRMS) implemented through two DAOs. The life cycles of fungible and Non-Fungible Tokens are also described.

A token is created for every euro paid in on the Investor DAO. Freezing is performed each time minting occurs on the Operator DAO (i.e., an operator’s request for the monetising of accrued credit). Burning happens at the end of the DAO when investments are redeemed. Token minting means creating tokens from scratch at the beginning of the system or increasing the supply to add more later on as a function of credit transactions. A token is created for every euro paid on the Investor DAO. Burning is performed each time minting occurs on the Operator DAO (i.e., an operator’s request for the monetising of accrued credit). Economic dynamics live around the tokens minted in the Investor DAO and issued exclusively to the GC. The GC can execute transactions on its own as the customer’s contact person using the two DAOs as follows:

- **Investor DAO:** At the fund's closing, the FI will convert the tokens accumulated in investors' wallets into fiat and redistribute the shares to them, including dividends. On the one hand, the excess tokens will be generated by the disposals towards the Operator DAO, which considers 100 to be the amount of the work and will receive a lower share of tokens, for example, 95, to compensate for the financial costs. On the other hand, the 100 tokens will generate 110 tax credits that the FI will sell on secondary markets to obtain, for example, 105. Thus, the investors for an investment of 100 will receive 115 (see Section 4 on token burning and minting).
- **Operator DAO:** For example, the GC receives 100 tokens, of which 30 go to the supplier, 20 to the Design Architect, and 10 to the Tax Auditor for audits based on invoices they have issued. The GC can make a profit of 40 tokens for the operation. All these tokens are redeemable into fiat either upfront or after a determined time to consent of the financial institution to organise payments.

### 5.3. DAOs Development

The framework chosen to develop the DAO modules of the demonstrator is Aragon (<https://aragon.org>, accessed on 10 November 2024). This is a second-level platform based on the Ethereum network that is natively designed to model government systems of public and private entities; it is easy to use and allows the deployment of test applications at very cheap transaction costs using Ethereum's test nets such as Goerly (<https://goerli.net>, accessed on 10 November 2024).

It is, furthermore, an independent platform oriented to design and implement DAOs and is thus particularly suitable for creating configurable governance structures. Aragon is ultimately a software for creating and governing organisations such as companies or cooperatives (<https://docs-staging.aragon.org/getting-started/>, accessed on 10 November 2024). The platform is based on the Aragon Network Token (ANT), which grants voting rights to its holders; polls are used to make decisions on the development of a DAO within Aragon. The model described is implemented through several smart contracts written with Solidity (<https://soliditylang.org>, accessed on 10 November 2024), an object-oriented language meant explicitly for contract implementation. The choice of such a stiff language is because we want the execution to be strictly what it is supposed to be without the possibility of modifications.

We implement the demonstrator on Aragon to simulate the operating principles of Investor and Operator DAOs; in future development, the integration with this demonstrator and the one described in this article could be implemented. The Aragon demonstrator, along with a representation of Smart contract's states and definition of the global and local constraint, data structures, and actors involved, is illustrated in [5].

### 5.4. Multi-Agent System Module

The MAS module of the demonstrator is composed of two main sub-systems, one internal meant to manage the core functions of the Superbonus 110% processes, and the second external to interface the whole MAS to the Double-DAOs module.

The interaction with DAOs happens through two service agents whose functionality is to interface Investor and Operator DAOs with the MAS; they supervise all information on basic operations from/to the DAOs. This part is not implemented at the moment in the demonstrator. Each agent has the relevant logic and internal communication functions integrated; in this way, blockchain interactions happen on a singular basis and are limited to the agent's needs. This part is implemented and described in Section 5.6.

Basic features of the MAS module are implemented through several agent roles to model all those defined in the previous section at Table 2. General Contractors, Sub-contractors, suppliers, Design Architects, and Tax Auditors have similar roles as negotiating agents. They are defined by homogeneous data (registry data) and similar functions to perform (checking payments received, verifying docs exchanged, etc.). This agent will react to human requests (e.g., approve a credit certification) and other agents' requests. Clients

(homeowners) have the initial function of hiring General Contractors and technicians, while financial institutions fund General Contractors once the client appoints them. In our system, a particular role is given to the workflow agent, a special agent used to model paperwork and evaluate the advances of the works. There are five states modelled (Open, Anticipation, Wps1, Wps2, and Eow), plus the final one for archiving (Archived) as per the workflow described in Section 2.1. The relevant features of agent roles are as follows:

- The **workflow agent** represents each Superbonus 110% process opened. Once the instance is created, the agent will be on position  $(x, 0)$ ; as soon as the work is approved, it will move to position  $(x, 1)$  corresponding to the **SoW** state. It will move once the state passage condition is met (e.g., the related Tax Auditor agent approves the credit for that stage) until it reaches the final position  $(x, 5)$ . Once the paperwork is complete, the agent will be archived (possibly for five months). The agent logic thus evaluates if the represented paperwork is due to be moved to the next state, i.e., the technicians have asseverated the current SoW, and the General Contractor has paid the anticipation for the next state. When the agents have instantiated, the total value of the work is generated by a random procedure with a value comprised of 1 mln and ten mln microAlgos.

To avoid excessive complexity, we model five pieces of paperwork for the demonstrator to handle, thus having five workflow agents. The logic is realised through a smart contract deployed on the Algorand testnet wallets of agents and is activated by opting into such a smart contract. Considering the current state of the workflow agent as  $x_{cur}$ , the last asseverated state as  $x_{asv}$ , and the previous paid as  $x_{pay}$ , the propositional logic notation for such a constraint evaluation is the following:

$$x_{cur}(i+1) \Leftarrow \neg(x(i)_{cur} = x_{asv}(i)) \wedge (x_{asv}(i) = x_{pay}(i)) \wedge \neg x_{arc}(i).$$

The workflow agents also pay the technician agents once the state is successfully updated; this happens by sending between Algorand wallets the amount calculated as a percentage (adjustable from the specific home page slider) of the state value (adjustable again from relevant sliders); see Figure 12 on Section 5.5.

- The **General Contractor agent** has the function of approving, on the financial side, the evolution to the next step of the paperwork. It simulates the passage of state, generating a random number to be checked against a threshold configurable as an input parameter similar to the workflow agent's procedure. If the approval is successful, the agent sends the amount due from their wallet on Algorand in anticipation of the next state to the related workflow agent. The amount is configurable, and the percentages of each state are set against the total value of the paperwork. To keep the system simple, we model a single General Contractor agent.
- **Technical agent:** This class merges the functionalities of both Design Architects and Tax Auditors with approvals randomly generated and checked against adjustable thresholds. In our simulator, we model two technician agents, each with an Algorand wallet, to obtain payments for asseverations.
- The **financial agent** represents the financial institution and acts as a sort of notary and banking entity, where all other General Contractor agents refer for funding approval, payment requests, and other Superbonus 110%-related financial issues. This agent also interacts with Communication Agents to perform all the token operations between the DAOs but will be modelled in the next version of the demonstrator.
- **Client agents:** Clients model the homeowners' activating Superbonus 110% interventions, appointing the General Contractor and technicians executing all the relevant works and certifications required by law. Following the condominium assembly, the workflow is created, and the agents described above are assigned. It is not modelled in this demonstrator version, and the General Contractor and technicians assignment to workflow agents is hard coded.

A common feature of agents is a wallet on the selected blockchain to perform and receive relevant payments, token operations, or other service activities. For the SFCM

architecture design, we use the MESSAGE/UML paradigm to model the agent interaction and behaviour [20] in Figure 13. In addition to the previous features, the agent has the following properties:

- **Property 1.** [*Workflow Agent*] To increase the performance of the operative players of the Superbonus 110% environment, the workflow agent evaluates the performance of its suppliers. The penalties and rewards are based on parameters like request response time, tariff discounts, and respect for delivery deadlines. Thus, at each WPS step, the agent evaluates the bonus or malus of each supplier involved in its construction site, modifying their score. It also applies a token penalty to “bad” agents that can be used to reward “good” agents. Operators who are not performing well must increase their performance to avoid losing their reputation and money. Consider  $t_{avg}(GC_{x_i})$  to be the historical average time for General Contractor GC to complete WPS states  $x_i$ ,  $d(GC_{x_i})$  the actual discount proposed for such a state, and  $p_{avg}(GC_{x_i})$  the historical percentage of on-time WPS completed. We can calculate a weighted value

$$w(GC_j) = w_1 \cdot t_{avg}(GC_{x_i}) + w_2 \cdot d(GC_{x_i}) + w_3 \cdot p_{avg}(GC_{x_i})$$

of such parameters for supplier  $GC_j$  and check it against a threshold limit value  $l$  to determine if a supplier is “good” or “bad”:

$$w(GC_j) \leq l$$

- **Property 2.** [*Financial Agent*] This functionality is a credit anomaly detector that checks anomalous credit transfers. The idea is to spot and evaluate possible fraud and misuse situations and prevent them from damaging customers, financial institutions, and operators. It enriches constraints 2, 3, and 4 of Section 3.1 and is realised through a more complex analysis of token movements between parties, such as too-fast token redemption, meaning works claimed could be fake. If General Contractor GC asks for a WPS credit redeem, having performed WPS states  $x_1$  and  $x_2$  in time  $t_{GC} = t(GC_{x_1}) + t(GC_{x_2})$ , then we can spot the suspicious behaviour of GC if

$$t_{GC} \leq s \cdot [t_{avg}(GC_{x_1}) + t_{avg}(GC_{x_2})]$$

where  $s$  is a suspicion rate percentage to be tuned with data from simulation (e.g., if  $s = 0.5$ , this means that claims performed in half the average time are considered suspicious).

### 5.5. MAS Framework and Implementation

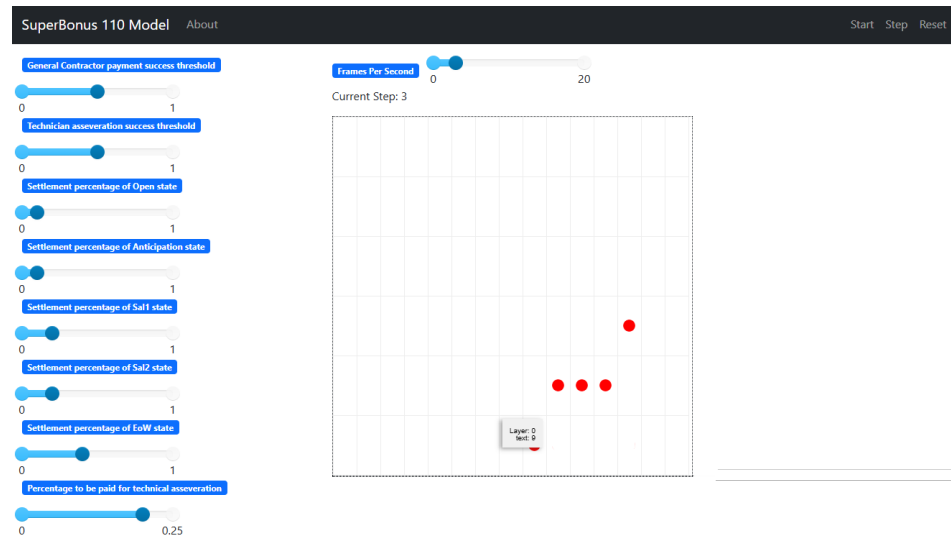
The MAS framework employed in this study is the MESA framework, a modular architecture written in Python. MESA enables the simulation of complex social systems by creating agents for each actor and designing their interactions within a specified environment.

Figure 13 shows how each agent role described in Section 5.4 is modelled within the system using different types of agents, including reactive agents, and the mechanisms for implementing communication and coordination among them.

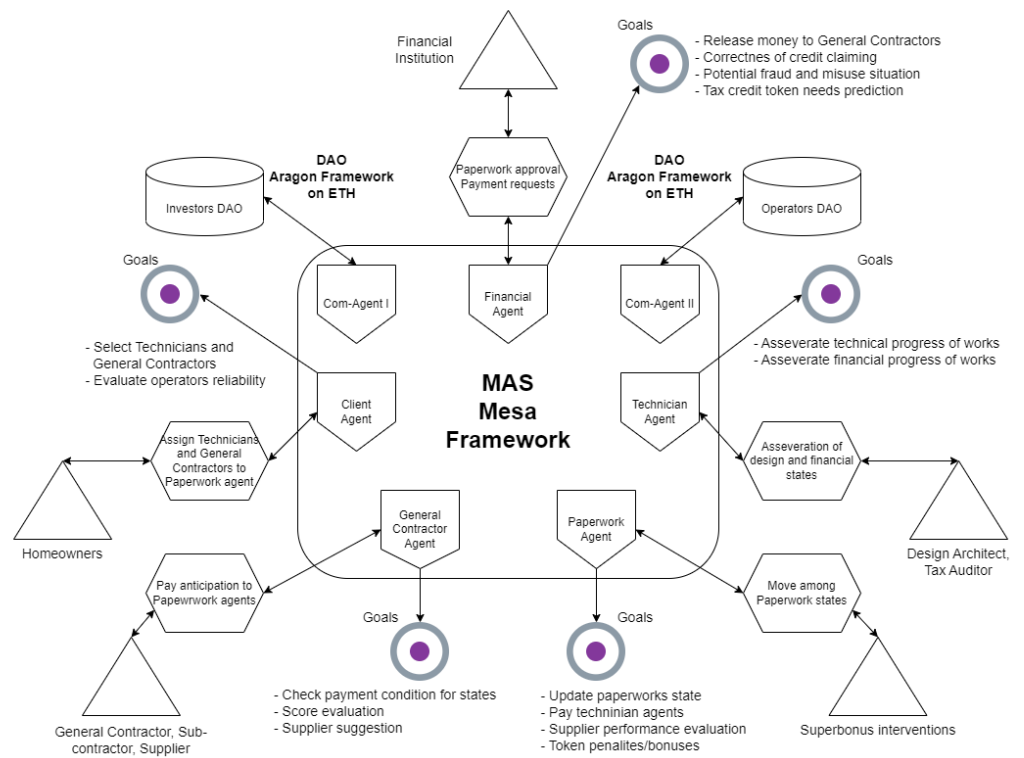
The MESA framework encompasses a variety of tools and modules essential for constructing, executing, and analysing MAS models. These include built-in agent behaviours, scheduling mechanisms, and data collection primitives [21].

The MESA web application framework incorporates visualisation tools, such as 2D grid representations of agent instances, time-series graphs related to variables evolution, and tabular data displays while offering real-time interactivity. This interactivity allows users to control simulation parameters and flow (start, pause, and reset), thereby facilitating a thorough analysis and exploration of complex MAS as shown in Figure 12. On the left-hand side, the first two sliders set the probability of the approval of the General Contractor payment and technician asseveration success. The second group of sliders represent the percentage of funds distributed across the five SAL levels. The final slider sets the

percentage allocated to the technicians (e.g., engineers, accountants, and construction managers). The central panel displays the grid representation of agent states.



**Figure 12.** On the left side, the demonstrator dashboard for setting the parameters is varied by moving each slider, thus modifying the success threshold of each scenario. To the right side is the agent’s status during the simulation.



**Figure 13.** Multi-agent System architecture of the demonstrator illustrates the interaction between the DAOs and MAS to Superbonus 110% simulation.

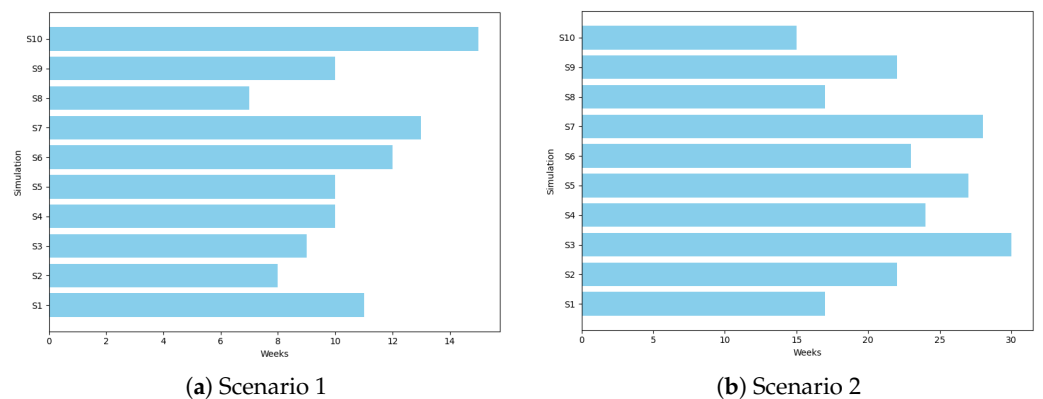
Table 3 summarises the agents involved in the two scenarios of the SFCM: the GC agent initially holds funds in their wallet and pays the technician agents for their assessments. Each SAL (i.e., Open, ANT, SAL1, SAL2, and EOW) is recorded by the paperwork agent, along with the entire Superbonus process documentation.

**Table 3.** Percentage of approvals in two different Superbonus scenarios.

Scenario	GC Payment	Tech Asseveration	Open	ANT	SAL1	SAL2	EOW	Tech Fee
1	50%	50%	10%	10%	30%	40%	10%	15%
2	25%	25%	10%	10%	30%	40%	10%	15%

- Scenario 1. High GC approval and high technician asseveration success rates:** This scenario examines the system’s behaviour under a high rate of both GC approval and technician asseveration for the submitted SALs. We simulate a 50% probability of approval for SALs in each cycle of the demonstrator. It is important to note that unapproved SALs are not rejected but remain pending and are carried over to the subsequent cycle. This introduces a time delay, mirroring real-world situations where approvals might be delayed due to factors such as the following:
  - Variations in local authority processing times;
  - The temporary unavailability of key personnel;
  - The need for additional documentation or clarifications.

As depicted in Figure 14a, high SAL approval rates result in a fluid workflow. State transitions occur with minimal delays, leading to an average project completion time (EoW: End of Work) of 10.3 weeks.



**Figure 14.** Histograms of the periods required to complete the works across the 20 simulations.

- Scenario 2. Low GC approval and low technician asseveration success rates:** This scenario simulates a less efficient approval process, reflecting potential bottlenecks stemming from contractor-side factors. We set a 25% approval probability per cycle but introduce delays to represent challenges such as the following:
  - Contractor-induced rework due to quality issues;
  - Delays in submitting required documentation by the contractor;
  - Resource constraints on the contractor’s end.

Figure 14b illustrates the impact of low SAL approval rates. The workflow progresses more slowly, with pronounced delays at various stages. Consequently, the average time to reach EoW increases to 23.9 weeks.

The visual representations and data derived from the SCFM provide valuable insights into the operational dynamics of Multi-agent Systems and the efficacy of various parameter configurations. Datasets generated from all simulations are publicly available on the Zenodo [22] platform, whilst the source code resides in the AAI-DISIM-UnivAQ GitHub repository ([https://github.com/AAI-DISIM-UnivAQ/Secure\\_Fiscal\\_Credit\\_Model](https://github.com/AAI-DISIM-UnivAQ/Secure_Fiscal_Credit_Model), accessed on 10 November 2024).

## 5.6. Blockchain Integration

The demonstrator implemented integrates several technologies to create a seamless interaction between smart contracts on the Algorand blockchain and agent-based simulations using the MESA framework. The key technologies employed are illustrated in the next subsections.

### 5.6.1. Algorand Blockchain

This is a decentralised platform known for its efficient consensus mechanism, enabling fast transaction finality and low fees. We deploy an account for each agent on the Algorand testnet. Each account is a wallet to collect and pay fees and an application area to upload smart contracts. The environment is set up on a Windows 11 machine through a WSL GNU/Linux virtual machine and Docker Desktop. The tool used for setting up the testnet is Algorand Sandbox and its three docker containers (indexer, Postgres and algod). The accounts are manually set up, and consent is given to open a wallet, which is used to simulate payments through the various agents and deploy applications. To implement, test, and deploy the smart contract on the workflow agents, we instead use Algokit Sandbox (<https://developer.algorand.org/docs/get-started/algokit>, accessed on 10 November 2024) with Dappflow (<https://github.com/dappflow>, accessed on 10 November 2024).

Each agent has an ALGO wallet connected to the vault in the testnet (<https://testnet.algoexplorer.io>, accessed on 10 November 2024). In our demonstrator, the smart contracts are written in Python and integrated into Algorand using PyTeal (<https://github.com/algorand/pyteal>, accessed on 10 November 2024) libraries and deployed to the blockchain using the Algokit SDK. As a testing tool, we used Beaker (<https://github.com/algorand-devrel/beaker>, accessed on 10 November 2024), a development environment to test and deploy Pyteal programs.

### 5.6.2. Implementation of Smart Contracts Using PyTeal

Algorand Smart Contracts are blocks of programmable logic hosted on the Algorand blockchain, accessible remotely. These contracts are responsible for implementing the specific logic of distributed applications. They can initiate asset creation and execute payment transactions on the blockchain using an assembler-like language known as Teal. The following snippet demonstrates a smart contract that updates the state of paperwork agents. The PyTeal library enables these contracts to be written and integrated directly in Python.

```

"""
app_pwork_.py
A simple example of PyTeal code to implement the state update of the
Paperwork agent
"""
from beaker import *
from pyteal import *

app = Application("PaperworkAgent")

@app.delete(bare=True, authorize=Authorize.only(Global.creator_address()))
def delete() -> Expr:
    return Approve()

@app.external
def update_state(
    actual: abi.String, assevered: abi.String, payed: abi.String, *, output:
        abi.String
) -> Expr:
    return (
        If((actual.get() != assevered.get()) and (actual.get() == payed.get())
        )
        .Then(output.set("True"))
        .Else(output.set("False"))
    )

```

```

)

if __name__ == "__main__":
    app.build().export("./artifacts")

```

The application deployment is achieved by compiling the smart contract and deploying it onto the Algorand network, using the Beaker library within the Algorand SDK for Python. This process involves initializing an Algorand client, constructing the Teal files (referred to as *artifacts*), and submitting the transaction to create the contract.

```

"""
deploy_pwork.py
The program to build artifacts from app_pwork.py application and deploy them
on the Algorand blockchain
"""
from beaker import client, localnet

from app_pwork_ import app, update_state

# Smart Contract is build and artifacts are created
app.build().export("./artifacts")

accounts = localnet.kmd.get_accounts()
sender = accounts[0]

app_client = client.ApplicationClient(
    client=localnet.get_algod_client(),
    app=app,
    sender=sender.address,
    signer=sender.signer_new,
)

app_client.create()

#Contract is initialised with default values
return_value = app_client.call(
    update_state, actual="Open", assevered="Ant", payed="Ant"
).return_value

print(return_value)

```

### 5.6.3. Setting Up and Running Simulations with MESA

**Model Definition:** We define a class, *SB110Model*, to represent the MESA model, incorporating a grid environment (see Figure 14).

Agents in the *SB110Model* class are instantiated with the following attributes: unique ID, name, data manager (dm), wallet, tech\_threshold, and private key. Agent behaviours are specified within the “step” function, which reviews the active paperwork, retrieves the current paperwork status, verifies assignment to the current technician, and sets an approval probability to simulate real-world scenarios.

The following code snippet for the technician agent integrates functionalities from both the project designer and accountant roles.

```

from mesa import Agent, Model
from mesa.time import RandomActivation
from mesa.space import MultiGrid

class TechnicianAgent(mesa.Agent):

    def __init__(self, unique_id, model, name, dm, wallet, private_key,
                 tech_threshold):

        super().__init__(unique_id, model)
        self.name = name

```



```

self.dm = dm
self.wallet = wallet
self.private_key = private_key
self.choose = False
self.wealth = 0
self.tech_threshold = tech_threshold

def step(self):

    for agent in self.dm.get_pwork_list():

        if agent.technician == self.unique_id:

            status = str(self.dm.get_pwork_status_from_dict(agent.
                unique_id))

            # Random simulation of Technician approving a new status
            approval = random.uniform(0, 1)
            self.choose = approval < self.tech_threshold

            # Success of approval, Tech Agent asseverates a new status
            if self.choose:
                # Technician inserts new asseverated status
                pos = int(self.dm.pwork_possible_states.index(status))
                new_status = self.dm.pwork_possible_states[pos + 1]

                # Update status of paperwork in Asseveration dictionary
                self.dm.update_tech_asseveration_in_dict(
                    agent.unique_id, new_status
                )
                print(f"New Asseveration state is: {new_status}")

                agent.set_tech_treshold (0) # if approved reset to zero
                    probability
                    threshold

            else:
                t = agent.get_tech_treshold() + 0.05
                agent.set_tech_treshold(t)

```

**MESA Integration:** Currently, the agent's logic is not implemented on the blockchain; rather, it operates within real-time data structures. Each agent class includes primitives that establish a connection with the Algorand testnet, enabling the initiation of transactions to pay or receive funds owed to relevant parties. The Mesa scheduler activates agents randomly, one class at a time, as illustrated in the code provided below.

```

self.schedule = mesa.time.RandomActivationByType(self)

# Adding Agents in the Scheduler
self.schedule.add(f_univaq)
self.schedule.add(t_facchini)
self.schedule.add(t_de_gasperis)
self.schedule.add(gc_letteri)
self.schedule.add(c_condo_via_salaria)
self.schedule.add(c_condo_via_roma)
self.schedule.add(p_condo_roma)
self.schedule.add(p_condo_salaria)
self.schedule.add(p_condo_vetoio)
self.schedule.add(p_condo_pontieri)
self.schedule.add(p_condo_rivera)

```

#### 5.6.4. Agent-Blockchain Connectivity

Upon instantiation, each agent is assigned a wallet address and a private key. For research purposes, these values are hard coded into each agent along with a unique identifier

and name. Concurrently, other parameters, such as the success threshold for document approval, are configured on the MESA web interface.

Transaction handling for agents is executed via the `send_transaction` function provided by the Algorand SDK, which verifies transactions using the sender's private key.

The code box below provides a sample code snippet demonstrating agent instantiation, Algorand socket initiation, and transaction triggering.

```
t_tech_1 = TechnicianAgent(  
    2, #unique id  
    self,  
    "Ing Facchini", # name  
    dm, #data manager  
    "*****", # wallet address  
    "*****", # private key  
    tech_threshold,  
)  
  
# Opening socket with Algorand testnet client  
# Token of Algorand testnet is used  
self.algod_address = "http://localhost:4001"  
self.algod_token = (  
    "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"  
)  
self.algod_client = algod.AlgodClient(self.algod_token, self.algod_address)  
  
send_transaction(self.private_key, self.wallet, agent.wallet, amount)  
  
pwork_account_info = self.algod_client.account_info(agent.wallet)  
print(f"Amount sent: {amount} Paperwork Account balance: {pwork_account_info.  
    get('amount')} microAlgos\n")
```

The Mesa MAS environment is integrated with the Algorand blockchain to extend the system-wide benefits of blockchain technology, particularly in terms of trust and traceability. We create a unique account for each agent on the Algorand testnet, where each account functions as a wallet, allowing agents to collect and pay fees for transactions, including both asseverations and work-related activities. Smart contracts are stored within these accounts, facilitating secure, autonomous interactions among agents.

## 6. Conclusions

In this study, we introduced a Secured Fiscal Credits Model (SFCM) integrating Multi-agent Systems (MASs) and Decentralised Autonomous Organisations (DAOs) to enhance the management of Italy's Superbonus 110% tax credit program. By leveraging blockchain technology, this model addresses critical challenges in tax credit tracking, including transparency, fraud prevention, and automation, thus setting a new precedent for decentralised fiscal governance. Theoretical implications include contributions to decentralised governance literature, demonstrating how MAS and DAOs can create secure, trust-free environments for managing public funds. The SFCM model benefits stakeholders by streamlining the tax credit life cycle, reducing administrative burdens, and increasing trust through transparent, automated credit exchanges. Nevertheless, the model's applicability depends on digital accessibility and regulatory adaptation, underscoring the potential limitations for broader adoption. Future research could expand on this work by exploring different fiscal environments, experimenting with alternative blockchain platforms, and examining the integration of real-time AI-based auditing. Ultimately, this research highlights the potential for decentralised, blockchain-driven systems to modernise government-backed incentive programs, providing a replicable framework that could extend beyond Italy's Superbonus initiative to support similar tax relief systems worldwide.

We have improved the modules of the multi-agents area, defining the knowledge base constraints and implementing a web-based software to simulate the interaction of a set of Superbonus 110% players. We also detailed both the DAO properties and the intelligent

part of the system, implementing the number and the functions of agents and suggesting a potential area of investigation to develop intelligent MAS controlling.

In Section 5.4, we defined two properties to spot fraudulent behaviour of agents and a reward system to encourage suitable suppliers. The model used in our demonstrator could be easily exported to other areas, both in the institutional and private sectors. In particular, public administrations may benefit from a reliable governance and voting system to provide services requiring tracking items and documents (e.g., board resolutions and certifications) [23]. In general, every process where the tracking of some asset or document is required could benefit from the architecture of blockchain in the environment of Distributed Trust and Reputation Management Systems (DTRMSs) [24].

A possible extension of the SCFM could also comprise (i) training machine learning models to conduct automated smart contract intent detection [25]; (ii) using a client-side validated state like RGB (<https://www.rgbfaq.com> accessed on 1 November 2024), a suite of protocols for the scalable and confidential smart contracts system operating as a side-chain of the Bitcoin ecosystem.

## 7. Authors Disclaimer

Declaration of Generative AI use: authors used occasionally online generative AI tools to improve the readability of the text; they reviewed and edited the content as needed, and took full responsibility for the content of the publication.

**Author Contributions:** Conceptualization, G.D.G. and S.D.F. and I.L.; methodology, G.D.G. and S.D.F. and I.L.; software, G.D.G. and S.D.F. and I.L.; validation, G.D.G. and S.D.F. and I.L.; formal analysis, G.D.G. and S.D.F. and I.L.; investigation, G.D.G. and S.D.F. and I.L.; resources, G.D.G. and S.D.F. and I.L.; data curation, G.D.G. and S.D.F. and I.L.; writing—original draft preparation, G.D.G. and S.D.F. and I.L.; writing—review and editing, G.D.G. and S.D.F. and I.L.; visualization, G.D.G. and S.D.F. and I.L.; supervision, G.D.G. and S.D.F. and I.L.; project administration, G.D.G. and S.D.F. and I.L.; funding acquisition, G.D.G. and S.D.F. and I.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** Partially funded by the European Union—NextGenerationEU under the Italian Ministry of University and Research (MUR) National Innovation Ecosystem grant ECS00000041 VITALITY CUP E13C22001060006.

**Institutional Review Board Statement:** Non applicabile.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data are publicly available.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

DAO	Decentralised Autonomous Organisation
MAS	Multi-agent System
SFCM	Secured Fiscal Credits Model
AI	Artificial Intelligence
FS	Feasibility Study
SoW	Start of the Work
WPS	Work Progress Status
CoD	Calculation of Deduction
EoW	End of Work
DLT	Distributed Ledger Technology
GC	General Contractor
SC	Sub-contractor
DA	Design Architect
TA	Tax Auditor

CCA	Compliance Control Agent
CM	Construction Manager
DoW	Director of Work
CW	Checker Work
SG&S	Suppliers Goods and Services
NFT	Non-Fungible Token
ANT	Aragon Network Token

## References

- Romano, V. Euractiv Superbonus 110 Description. 2022. Available online: <http://www.euractiv.com/section/energy/news/italys-feted-superbonus-for-building-renovation-comes-under-scrutiny/> (accessed on 1 Novembre 2024).
- Goertzel, B.; Iklé, M.; Potapov, A.; Ponomaryov, D. *Artificial General Intelligence: 15th International Conference, AGI 2022, Seattle, WA, USA, 19–22 August 2022, Proceedings*; Springer Nature: Berlin/Heidelberg, Germany, 2023; Volume 13539.
- Dyoub, A.; Costantini, S.; Lisi, F.A.; Letteri, I. Ethical Monitoring and Evaluation of Dialogues with a MAS. In *Proceedings of the 36th Italian Conference on Computational Logic, Parma, Italy, 7–9 September 2021*; CEUR Workshop Proceedings; Monica, S., Bergenti, F., Eds.; CEUR-WS.org, 2021; Volume 3002, pp. 158–172.
- Dyoub, A.; Costantini, S.; Lisi, F.A.; Gasperis, G.D. Demo Paper: Monitoring and Evaluation of Ethical Behavior in Dialog Systems. In *Proceedings of the Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness. The PAAMS Collection—18th International Conference, PAAMS 2020, L'Aquila, Italy, 7–9 October 2020, Proceedings*; Lecture Notes in Computer Science; Demazeau, Y., Holvoet, T., Corchado, J.M., Costantini, S., Eds.; Springer: Cham, Switzerland, 2020; Volume 12092, pp. 403–407. [[CrossRef](#)]
- De Gasperis, G.; Facchini, S.D.; Susco, A. Demonstrator of Decentralized Autonomous Organizations for Tax Credit's Tracking. In *Proceedings of the Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection*; Dignum, F., Mathieu, P., Corchado, J.M., De La Prieta, F., Eds.; Springer: Cham, Switzerland, 2022; pp. 480–486.
- Facchini, S.D. Decentralized Autonomous Organizations and Multi-agent Systems for Artificial Intelligence Applications and Data Analysis. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, Vienna, Austria, 23–29 July 2022*; Doctoral Consortium; Raedt, L.D., Ed.; International Joint Conferences on Artificial Intelligence Organization: Montreal, Canada, 2022; pp. 5851–5852. [[CrossRef](#)]
- Huerto-Cardenas, H.E.; Aste, N.; Buzzetti, M.; Del Pero, C.; Leonforte, F.; Miglioli, A. Examining the role of the superbonus 110% incentive in Italy through analyses of two residential buildings. *E3S Web Conf.* **2024**, *546*, 2007. [[CrossRef](#)]
- Codogno, L. *Italy's Superbonus 110%: Messing Up with Demand Stimulus and the Need to Reinvent Fiscal Policy*; IMK Studies 93-2024; IMK at the Hans Boeckler Foundation, Macroeconomic Policy Institute: Düsseldorf, Germany, 2024.
- Karakostas, D.; Kiayias, A. Filling the Tax Gap via Programmable Money. In *Proceedings of the Data Privacy Management, Cryptocurrencies and Blockchain Technology*; Garcia-Alfaro, J., Muñoz-Tapia, J.L., Navarro-Arribas, G., Soriano, M., Eds.; Springer: Cham, Switzerland, 2022; pp. 281–288. [[CrossRef](#)]
- Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: <https://bitcoin.org/bitcoin.pdf> (accessed on 1 July 2015).
- Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the Security and Performance of Proof of Work Blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA, 24–28 October 2016*; CCS '16; pp. 3–16. [[CrossRef](#)]
- Catalini, C.; Gans, J.S. Some Simple Economics of the Blockchain. *Commun. ACM* **2020**, *63*, 80–90. [[CrossRef](#)]
- Voshmgir, S.; Zargham, M. *Foundations of Cryptoeconomic Systems. Working Paper Series/Institute for Cryptoeconomics/Interdisciplinary Research 1*; WU Vienna University of Economics and Business: Vienna, Austria, 2020.
- Buterin, V. A next-generation smart contract and decentralized application platform. *White Pap.* **2014**, *3*, 2-1.
- Baninemeh, E.; Farshidi, S.; Jansen, S. A decision model for decentralized autonomous organization platform selection: Three industry case studies. *Blockchain Res. Appl.* **2023**, *4*, 100127. [[CrossRef](#)]
- Farshidi, S.; Jansen, S.; de Jong, R.; Brinkkemper, S. A Decision Support System for Cloud Service Provider Selection Problem in Software Producing Organizations. In *Proceedings of the 2018 IEEE 20th Conference on Business Informatics (CBI), Vienna, Austria, 11–13 July 2018*; Volume 01, pp. 139–148. [[CrossRef](#)]
- Dilger, W. Decentralized autonomous organization of the intelligent home according to the principle of the immune system. In *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997*; Volume 1, pp. 351–356. [[CrossRef](#)]
- Kypriotaki, K.; Zamani, E.; Giaglis, G. From Bitcoin to Decentralized Autonomous Corporations - Extending the Application Scope of Decentralized Peer-to-Peer Networks and Blockchains. In *Proceedings of the 17th International Conference on Enterprise Information Systems—Volume 3, Barcelona, Spain, 27–30 April 2015*; ICEIS, INSTICC; SciTePress: Setúbal Municipality, Portugal, 2015; pp. 284–290. [[CrossRef](#)]
- Sun, X.; Stasinakis, C.; Sermpinis, G. Decentralization illusion in DeFi: Evidence from MakerDAO. *arXiv* **2022**, arXiv:2203.16612 [[CrossRef](#)]
- Ozkaya, M. Analysing UML-based software modelling languages. *J. Aeronaut. Space Technol.* **2018**, *11*, 119–134.

21. Masad, D.; Kazil, J. MESA: An agent-based modeling framework. In Proceedings of the 14th PYTHON in Science Conference, Citeseer, Austin, TX, USA, 6–12 July 2015; Volume 2015, pp. 53–60.
22. De Gasperis, G.; Facchini, S.D.; Letteri, I. Leveraging Multi-Agent Systems and Decentralised Autonomous Organisations for Tax Credit Tracking: A Case Study of the Superbonus 110% in Italy. Available online: <https://zenodo.org/records/13742335> (accessed on 10 November 2024).
23. Chohan, U. The Decentralized Autonomous Organization and Governance Issues. *SSRN Electron. J.* **2017**. [[CrossRef](#)]
24. Bellini, E.; Iraqi, Y.; Damiani, E. Blockchain-Based Distributed Trust and Reputation Management Systems: A Survey. *IEEE Access* **2020**, *8*, 21127–21151. [[CrossRef](#)]
25. Huang, Y.; Zhang, T.; Fang, S.; Tan, Y. Deep Smart Contract Intent Detection. *arXiv* **2022**, arXiv:2211.10724. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.