Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Cybersecurity in physically entangled networks

Ammar Qureshi
Master in Computer Science
Dissertation
University of Dublin, Trinity College
Supervisor: Dr. Khurshid Ahmad

School of Computer Science and Statistics
O'Reilly Institute, Trinity College, Dublin 2, Ireland

Submitted to the University of Dublin, Trinity College, April, 2019

Contents

List of Figures

List of Tables

Declaration

I, Ammar Qureshi, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request.

Signed: _____ Date: _____

Acknowledgements

First and foremost, I would like to thank my thesis supervisor, Professor Khurshid Ahmad, for his continuous and valuable help over the course of this dissertation. My sincere gratitude for his willingness to dedicate his time generously.

To the friends, I have made during the past five years at Trinity. I will always remember the countless late night team submissions we have endured, as it was our firm belief that procrastination always gave you something to look forward to.

Last but not least, I would like to thank my parents for their constant support throughout my academic journey.

Summary

As we network and connect technology and infrastructure to an already complex system, we create tremendous new value but also potentially leave the system more vulnerable to cyber attacks. One such example of a complex system is the power grid, which unarguably is one of the most critical systems for a functioning modern day society. However, with many small scale cyber attacks which remain unnoticed and the highly publicised large scale cyber attacks such as the Estonian cyber attack in 2007 and the Ukraine Blackout in 2015 indicates that such complex systems are far from fully secure.

This paper explores the conventional modelling techniques in the modelling community and provides reasons for modelling the behaviour and interactions of entities in a complex system with Agent-Based Modelling. Furthermore, this paper will explore key literature on how complex systems such as the smart grids are secured from cyber-attacks. A brief overview of blockchain technology will be presented and the justifications for why blockchain technology may have the potential to secure interactions of stakeholders in a complex system.

Additionally, this paper will build an agent-based simulation of the smart grid consisting of the end-user, supplier and generator with the Mesa framework, where agent interactions interface with smart contracts on the Ethereum blockchain through Web3.py. To study the behaviour of interactions under cyber-attack conditions, a threat model, more specifically a simple virus which infects other nodes and tampers with data will be introduced to the model. The behaviour of the various heterogeneous stakeholders and the state of the overall systems will be studied with and without the threat model through various case studies in order of increasing complexity. Also, final remarks on the effectiveness, efficiency and security of the blockchain technology and modelling complex systems are concluded.

Abstract

As we network and connect technology and infrastructure to an already complex system, we create tremendous new value but also potentially leave the system more vulnerable to cyber attacks. One such example of a complex system is the power grid, which unarguably is one of the most critical systems for a functioning modern day society. However, with many small-scale cyber attacks which remain unnoticed and the highly publicised large scale cyber attacks such as the Estonian cyber attack in 2007 and the Ukraine Blackout in 2015 indicates that such complex systems are far from fully secure.

This paper explores the various modelling techniques to model complex systems of systems such as the smart grid. Additionally, this paper builds a simulation by pipelining two relatively new concepts in computing, namely Agent Based Modelling and Blockchain technology with a user interface which allows for dynamic structuring of the network. The agent interactions interface via blockchain based smart contracts. Furthermore, the simulation introduces a simple threat model into the modelled system secured by blockchain technology to study the behaviour of the various heterogeneous stakeholders in the system and the overall global behaviour of the system through various case studies in order of increasing network complexity. Finally, remarks on efficiency and security of blockchain technology and the limitations of agent based models are concluded.

Abbreviations

Acronym	Meaning
CPS	Cyber-Physical System
SoS	System of Systems
CPSoS	Cyber-Physical System of Systems
ABM	Agent Based Model
DES	Discrete Event Simulation
SD	System Dynamics

Background

System

EU Project DSOS((Dependable System-of-systems IST-1999-11585)) defined a system as "An entity that is capable of interacting with its environment and may be sensitive to the progression of time". A system is sensitive to the progression time if the system reacts differently at different points in time to the same pattern. A classic example is a time-controlled central heating system, where one of the factors affecting temperature is the current time.

System of Systems(SoS)

The transportation systems, the global banking industry, a water-supply system, military equipment and a great number more, strongly rely on SoS.

A SoS is a concept from the domain of systems engineering.
Maier et al. [3] put forth five key characteristics of a SoS:

- Operational independence of the components of the overall system
- Managerial independence of the components of the overall system
- Geographical distribution
- Emerging behaviour
- Evolutionary development processes

The purpose of SoS is to bring together a set of cooperating systems for a task that systems cannot accomplish independently or cannot be provided in an as efficient manner as by the whole system.
Components in a system have Operational Independence if the disassembled component systems can usefully operate independently. That is, the components fulfil customer-operator purposes on their own.

Components in a system have Managerial Independence if the disassembled components not only can be operated independently but do operate independently. The

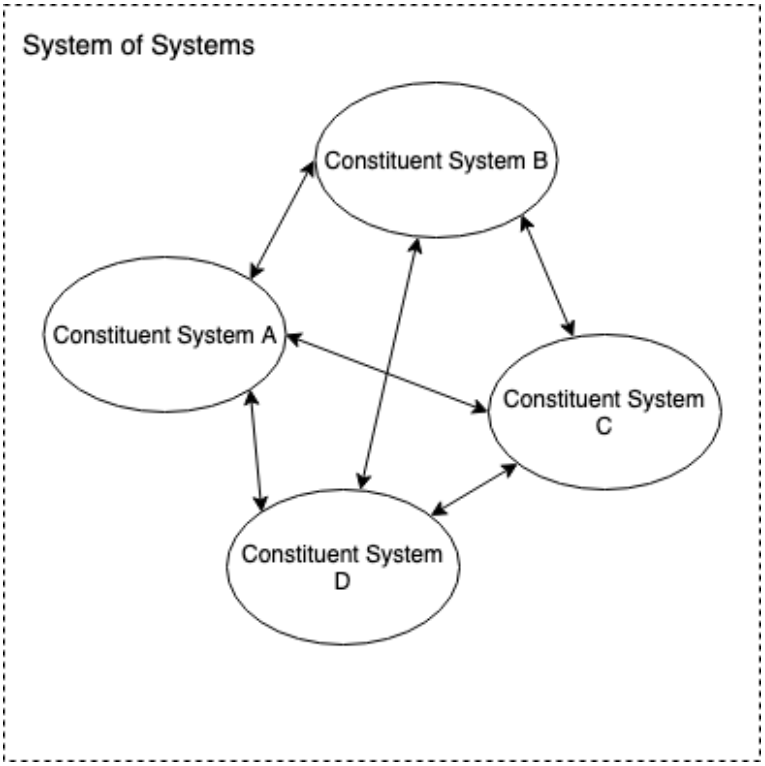


Figure 1: System of Systems

component system are separately acquired and integrated but maintain a continuing operational existence independent of the SoS. According to Maier et al. [3] if a system does not meet criteria of Operational Independence and Managerial Independence, it is not considered a SoS regardless of the geographic distribution of its components.

Such independence leads to a dynamic change of structure and connectivity of systems over time, where systems can be added, upgraded, connected or disconnected and the whole system and its constituent components can be dynamically reconfigured.

Partial autonomy of several components is essential for the concept of SoS. Each constituent system keeps its own management, rules, and resources while coordinating within SoS to satisfy the goals of the whole system (see Figure 1). It does not necessarily mean human-free operations as human supervision and intervention are an important element of autonomy of constituent system as well as of the whole system.

Table 1 shows the comparison between Monolithic Systems and SoS through various

characteristics.

Characteristic	**Monolithic System**	**System of Systems**
Scope of System	Fixed(known)	Not known
Clock Synchronisation	Internal	External(ie CPS)
Structure	Hierarchical	Networked
Requirements and Specification	Fixed	Changing
Evolution	Version Control	Uncoordinated
Implementation Technology	Given and Fixed	Unknown
Testing	Test phases	Continuous
Faults(Physical,Design)	Exceptional	Normal
Control	Central	Autonomous
Emergence	Insignificant	Important

Table 1: Comparison of Monolithic system and SoS(Adapted from [7])

Cyber Physical Systems of Systems(CPSoS)

Technological advancements have been fueled by simultaneous development in data science, artificial intelligence, telecommunication, computation, sensors, actuators, materials and augmented reality. Different perspectives on these developments have led to the creation of Cyber-Physical Systems, IoT, Industry 4.0, to name a few. Cyber-Physical Systems (CPS) are smart systems that include engineered interacting networks of physical and computational components [2] shown in Figure 2.

Engell et al. [4] describes a CPS as a "large complex physical systems that are interacting with a considerable number of distributed computing elements for monitoring, control and management which can exchange information between them and with human users".

Engell et al. [4] defines CPSoS as a CPS which exhibit the features of SoS in correspondence of Maier et al. [3] :

- Large, often spatially distributed physical systems with complex dynamics
- Partial autonomy of the subsystems
- Dynamic reconfiguration of the overall system on different time-scales
- Continuous evolution of the overall system during its operation
- Possibility of emerging behaviours.

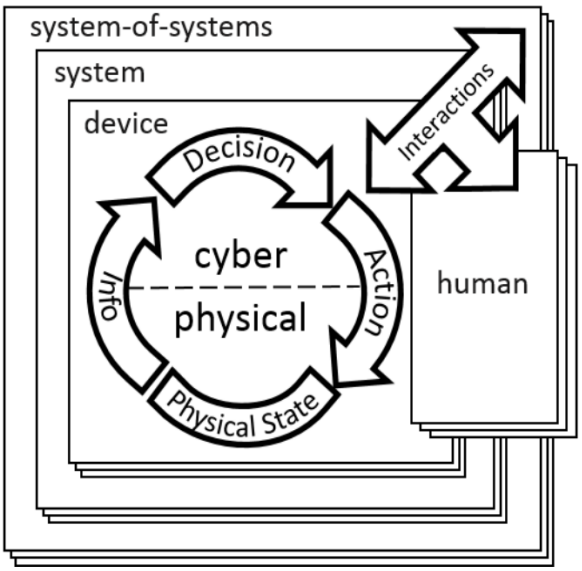


Figure 2: NIST CPS Conceptual Model [2]

Emergence

Aristotle succintly communicates the concept of emergence with the following quote:

"The whole is greater than the sum of its parts"

The interactions of constituent systems form complex systems, with abnormal behaviours, properties and structure that go beyond the attributes of any individual constituent system. Bondavalli et al. [7] provide a more formal definition of emergence as "A phenomenon of a whole at the macro-level is emergent if and only if it

is of a new kind with respect to the non-relational phenomena of any of its proper parts at the micro level."

The concept of emergence is quite common in large scale CPSoS. Figure 3 shows a schema for the classification of emergence, the emergent phenomena can be split into the domain, predictability, explanation and consequences.

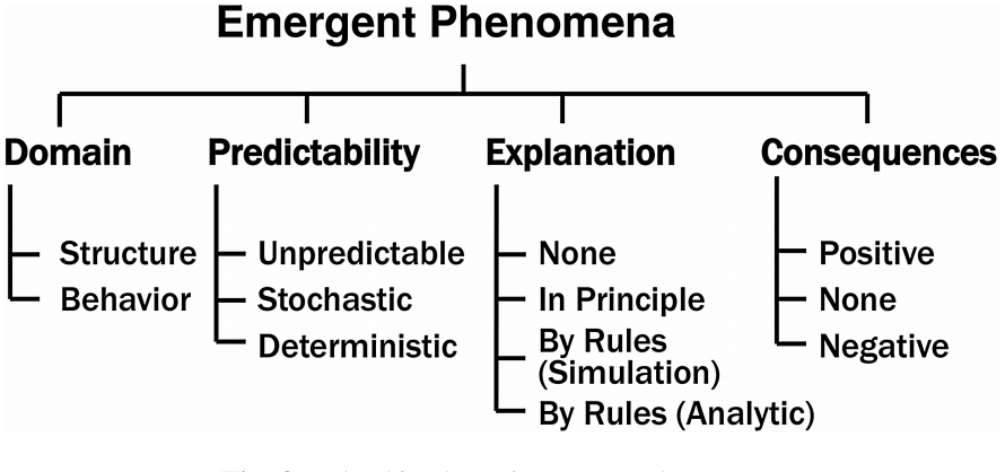


Figure 3: Classification of Emergence [7]

In many cases, the principles that can explain emergent phenomena are formulated post facto as it would require very knowledgeable and conscious minds to predict a priori all possible emergent phenomena that can come into existence out of interactions of many constituent systems.

Managing emergence is essential to avoid undesired detrimental situations generated from smart grid interactions. System safety has been acknowledged as an emerging property, because its meaning at the SoS level does not have the same meaning for the individual constituent system, and it cannot be merely expressed as the sum of the individual parts.

The Power grid: A CPSoS

The current grid infrastructure is, for the most part, a hierarchical system. The producers and suppliers at the top of the hierarchy ensure energy transmission to

customers' loads at the bottom of the chain. The system is inherently unidirectional where the source has no real-time information about the consumers. Therefore, the grid has been over-engineered to withstand maximum anticipated peak energy demand, where such anticipated demand peaks occur infrequently resulting in inefficiency.

Furthermore, with the growing demand for energy combined and the lack of investment in bulk power plants will result in a decrease in system reliability and stability. Any unanticipated surge in demand or anomalies can cause cascading failures due to the current topology of the system triggering disastrous blackouts.

Figure 4 shows the existing grid architecture. The generation, transmission systems, network of substations and distribution network are centralised with basic data network. Whereas at the customer end there is no data network.

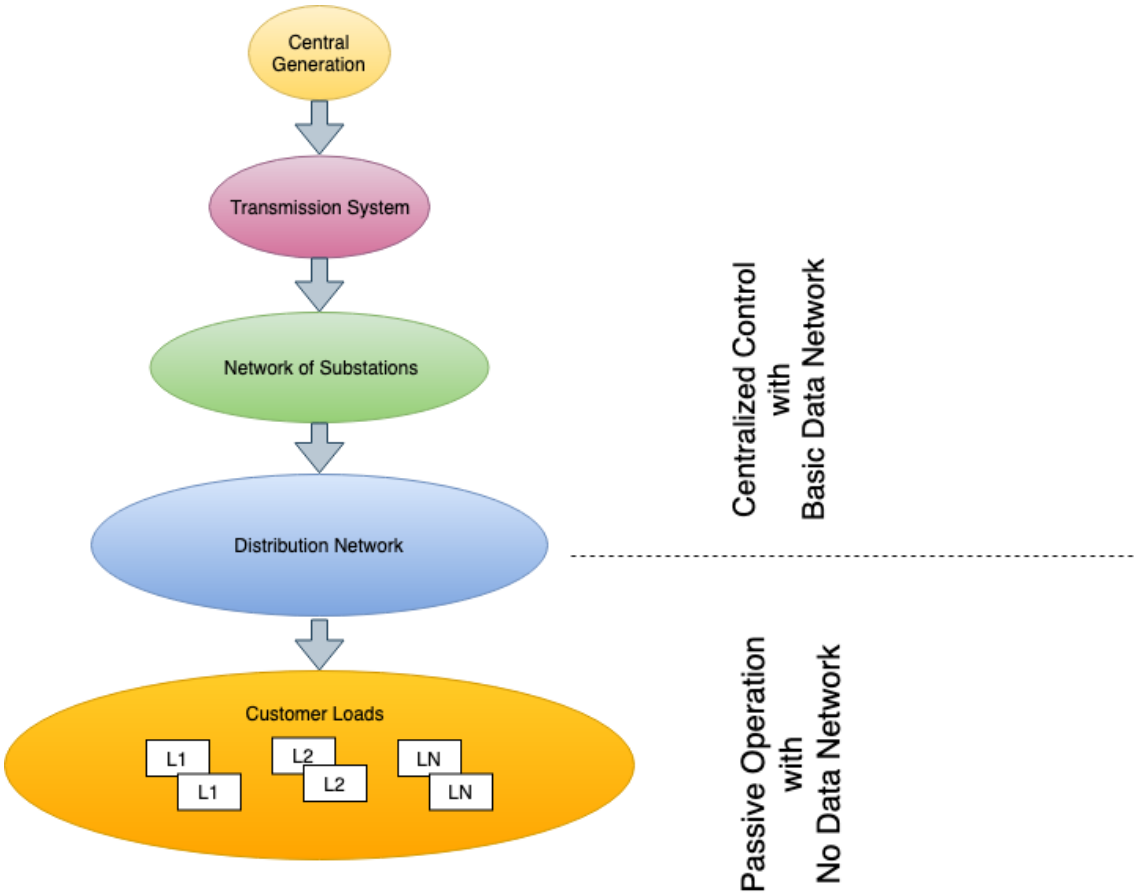


Figure 4: Existing Grid Architecture [6]

Smart grids are envisioned to address the major flaws of the current power grid. The smart grid is a CPSoS, where interacting CPSs (producers, distribution system operator, transmission system operator, consumers and prosumers (who produce and consume energy)) cooperate to continually optimise the use of energy resources while minimising operational and maintenance costs and maximising stability and dependability of the grid.

A smart grid handles dynamicity of the grid by constantly re-configuring itself in order to balance energy production and consumption loads. Furthermore, smart grids need to be adaptive during runtime as constituent systems are in constant renovation, upgrade or extension towards new requirements or technological advances. Even though such a large-scale CPSoS is continuously evolving, it must continuously provide dependable service 24 hours a day, seven days a week, 365 days a year. Figure 5 shows the various interactions among stakeholders in the highly networked and large-scale CPSoS.

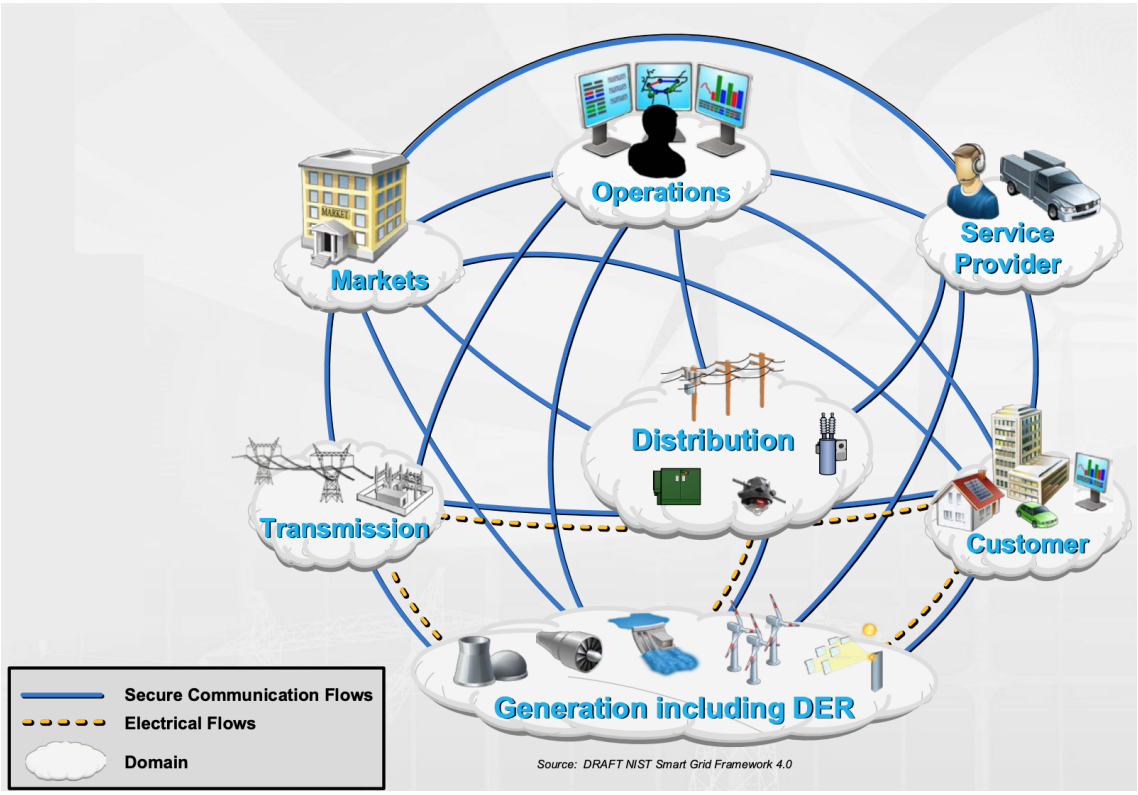


Figure 5: NIST Smart Grid Conceptual Model [5]

One of the key technologies is the concept of the smart meter which will manage devices in smart homes to minimise energy inefficiencies and in turn reducing energy

costs for consumers. Smart meters will also interact with the grid; this bidirectional communication will help reduce peaks in demand, enable dynamic optimisation of electric-system operations, maintenance, planning and the integration of new energy technologies such as solar and wind energy.

Table 2 compares characteristics of the existing grid and the the smart grid.

Characteristics	**Existing Grid**	**Smart Grid**
Communication	One-way	Two-way
Topology	Centralized	Network
Generation	Centralized	Distributed
Failure Restoration	Manual	Self-Healing
Checks	Manual	Remote
Consumer participation	Non-participative	Informed and Involved
Energy Fluctuation	Failures and Blackouts	Adaptive and Islanding

Table 2: Comparison of characteristics between existing grid and smart grid(Adapted from [6])

1 Introduction

Roughly speaking a complex systems is one with constituent parts, whose behaviours are both highly variable and strongly dependent on the behaviour of other parts. One can imagine the complexity which arises when we introduce new nodes into an exiting network. This is further complicated by the fact that the new node itself would be a complex object.

In the real world we can see examples of such complex networks such as the introduction of mobile devices into a network, passengers on an aeroplane using their mobile systems, or the introduction of smart meters in network. They appear as an innocuous addition to an existing network, apparently improving the efficiency which in the wrong hands can be used as a medium to launch attacks on other nodes in the network.

Figure 6 shows a "tree" of various attacks and threats based on the functional model of Cyber-Physical Systems. Branches of the "tree" include different types of attacks which can lead to a Cyber-Physcial System Failure, they include attacks on:

- sensor devices(Sensing)
- actuators(Actuation)
- computing components(Computing)
- communications(Communication)
- feedback(Feedback)

The modelling of complex systems or as fashionably called System of System(SoS) requires that each constituent system be simulated using an appropriate mathematical model. A mathematical model in itself is an approximation and it is generally difficult to build a mathematical model of a complex systems which can be simulated.

Figure 6: Tree diagram of cyber threats and attacks on a Cyber-Physcial System(CPS) [14]

One complex system which has attracted the attention of authors recently is the electricity grid. The electricity grid has become a vital component of modern day society, with every critical infrastructure, from transportation and telecommunication systems to water supply systems being dependent on electricity. The National Academy of Engineering ranking the electricity grid as the greatest engineering achievement of the 20th-century [1].

Figure 7 shows the evolution of the grid infrastructure. During the pre-Internet era, to communicate the energy demand of the consumers, fixed-line communication was

established between the system operators and the generation power plants. In the post-Internet era, as the grid evolves, increased connectivity and new systems in the smart grid are introduced. This connectivity gives rise to a stratified level of interactions resulting in a higher degree of structural and behavioural complexity, making it difficult and in some cases impossible to understand the system.

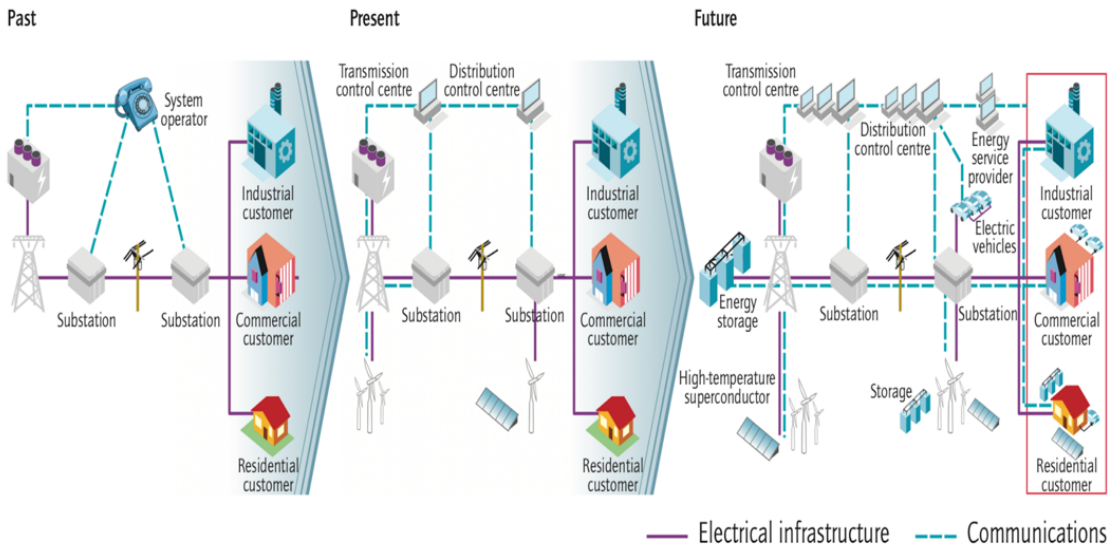


Figure 7: Evolution of the Grid [16]

One can consider energy security from two perspectives. In one sense this phrase refers to being careful in generating and using energy as both contribute to global warming and source depletion. Second, is the danger of losing control of parts of the network. The complexity of the system amplifies with the modernisation of the grid. Due to the deep integration of both cyber and physical layer, attacks from the cyber layer have the potential to mislead decision-making in the control centre and cause system disturbances, financial loss and large scale blackouts which can result in disturbances in day to day life or possible loss of life.

Granted that smart components are vital for the full realisation of smart grids, they undoubtedly also increase the vulnerability of the grid system. This is because cybersecurity often is an afterthought for vendors and consumers as they prioritise functionality and cost, leaving the power grid vulnerable to cyber-attacks
Smart meters are components that are usually connected to local networks or the Internet. Therefore, as the grid evolves, the number of Access Points(AP) increase

compared to the traditional power grids leaving the grid more vulnerable to cyber attacks.

Adversaries can launch attacks by hacking remote terminal units(RTUs) such as sensors placed in substations or smart meters in smart homes which can compromise physical data measurements and state estimations leading to distorted energy demand and supply figures. Data tampering can be launched by consumers to reduce their energy bills, competing corporations, or hostile countries, aiming to compromise data measurements to increase the cost of energy distribution and smart grid operations or even causing large scale blackouts which could result in loss of life. In this sense, data vulnerability has become an unneglectable issue.

Many cyber attacks are left unnoticed or unpublished, however, some of the acknowledged and published cases of large scale malicious cyber attacks are the Estonian Cyber Attack in 2007 [82] [83] and the Ukraine Blackout in 2015 [10] [11].

The National Electric Sector Cybersecurity Organization Resource (NESCOR) published a detailed report [33] on the possible electricity sector failure scenarios and impact analyses that include both malicious and non-malicious cybersecurity events.

Network simulation models have been developed with these objectives in mind - to model complex systems and study the behaviour of the network. Some of the components and actors involved in the network cannot be modelled mathematically so alternative suggestion has been to use the so called Agent Based Model(ABM which can be programmed using AI techniques to reproduce the behaviour of those systems through production rules (if-then) coupled with fuzzy logic and Machine Learning programs.

In one sense, an agent is an object which communicates with other objects as Object Oriented languages but also knows to select its own input and produce its own output with the view of achieving certain goals which have been specified at the matter level.

The above security issues mentioned implies that many smart grid operations are not conducted in a secure environment and much of the multiparty transactions are not auditable, and a significant proportion of the transactions and exchanges

depend on middle-men, leaving plenty of room for accounting errors to outright fraud. Ensuring the integrity and consistency of data is of critical importance for the secure and economical operation of complex systems. Furthermore, existing communication and storage of critical data in critical and complex systems are less than fully effective against cyber-attacks. Figure 8 shows the steps of current data communication. Cyber attackers are able to manipulate data during data collection phase, during data transmission phase due to weak or absence of encryption(plain-text) and when the data is received and stored in centralised control centres.

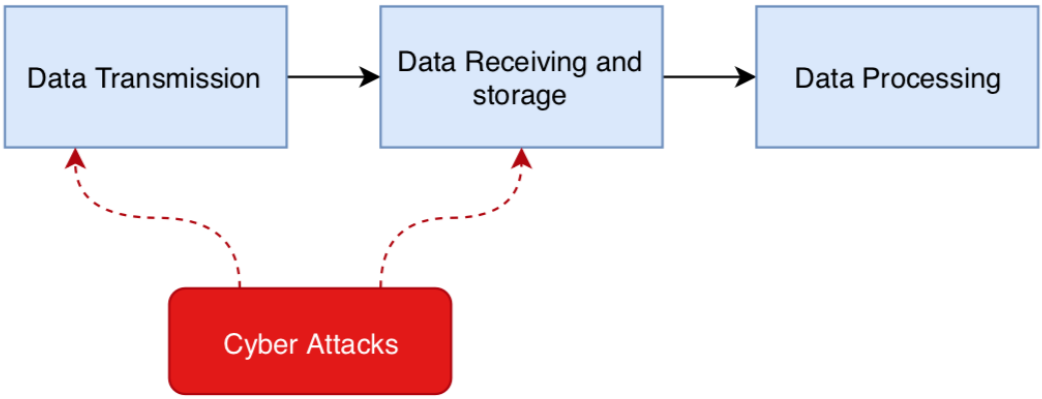


Figure 8: General procedure for existing data communication(Adapted from [40])

ABM have been identified to be suitable for modelling electricity networks [80]. There are other simulation techniques are currently being used by one of the major R&D institutions in the U.S to simulate the electricity network [38].

For aspects of security, which is the protection against cyber attacks, elaborate models of small number of device within an electricity network, mainly design with sensors and control devices, to assess the vulnerability of the network against such attacks [38]. Here the emphasis is on using smart contracts between stakeholders generated using Blockchain technology. The authors claim that applying blockchain based smart contracts presents an opportunity to increase the speed, scale and security of transactive energy applications.

Blockchain technology could help mitigate the tampering of data as the ledger would record the energy transaction time and use data as registered in a block. This

provides a means of verifying what data is valid and what data is invalid, enabling the blockchain platform to quarantine data, drop malicious commands not contained in the smart contract and return to a steady state.

Figure 9 shows the steps of blockchain based data communication. Data is encrypted before it is transmitted to the network, the data is validated and verified based on a distributed consensus mechanism by the peers in the network and is finally stored in the distributed ledger.

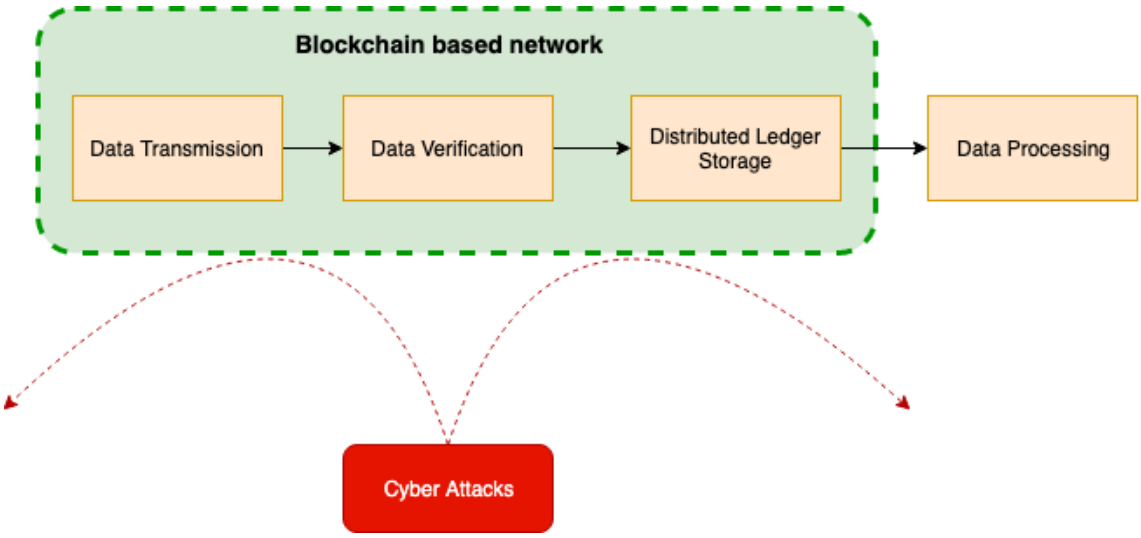


Figure 9: Blockchain based procedure for data communication(Adapted from [40])

It is to be noted that a Blockchain solution does not guarantee 100% security or prevention of cyber attacks. For example, Blockchain may not prevent access to behind the meter systems [38]. Instead, it improves security through authentication, encryption and ability to verify the integrity of the data. Furthermore, blockchain may reduce or possibly removes the need for intermediaries to clear transactions. This potentially reduces the attack landscape by reducing the number of nodes susceptible to attack. However, its security value is more about securing or protecting integrity once an attacker is already in a system. Blockchain can help detect manipulation of configurations, or critical systems are changed or the terms of smart contract are manipulated.

Currently, this appears to be ambitious because Blockchain technology and its application, smart contracts in themselves are vulnerable to attacks, they also present

low throughput, scalability and high energy consumption which is not suitable for large complex systems. It is not clear such smart contracts can be deployed for thousands of customers and devices including smart meters. However, it is important to see what happens during a small scale simulation to assess this situation.

Key results :
A deeper appreciation of modelling complex systems has been gained by modelling heterogeneous stakeholders in the power grid systems through different modelling techniques, especially Agent Based Model(ABM). Moreover, a working understanding of Blockchain technology, both permissioned and permissionless, including smart contracts has been acquired for exploring questions in security of networks.

Contribution:
We have developed a prototype system with GUI, where we have modelled the behaviour of end users, suppliers and generators. Using ABM, the various agents can interact through smart contracts on a Blockchain.

The rest of the paper is organised as follows. Section 2 discusses the literature review, we appreciate as well as critique the work of relevant authors in the field of cybersecurity in the energy sector space and their modelling techniques, if any. Section 3 presents the Methodology; we will introduce the two computing aspects our paper, namely operational modelling and cybersecurity. We then present the design, tools used and the implementation for our blockchain integrated agent based simulation. Section 4 provides a few case studies of the program and evaluation of the results. Finally, Section 5 briefly provides future work, challenges and final remarks.

2 Literature Review and Motivation

Rusitschka et al. [37] proposed a real-time smart meter data management system based on the Smart Grid Data Cloud. Their Cloud Computing model allows collaboration and exchange of information between consumers, retailers, virtual power plant operators of highly distributed generation as well as the network operators.

Ye et al. [36] proposed a data-driven, cloud-based ICT framework for smart grids in order to allow utility companies to have maximum security control over data. By leveraging Big Data analytics, Cloud Computing and other information sources(i.e. weather forecasts, news, social network, stock markets) they claim to have better forecast prediction compared with only historical data of electricity consumption.

The author's cloud-based framework is secured by Identity-Based(ID-based) Security Scheme(IBSC), which performs the function of both digital signature and encryption simultaneously. The scheme makes use of public key cryptography. In their scheme, instead of generating keys randomly, public key generation is computed based on the ID of client together with a given time after which computation of the public key is not permissible by the system.

However, the proposed Cloud Computing platforms of Rusitschka et al. [37] and Ye et al. [36] do not possess characteristics of data immutability and non-repudiation. Both of which are not only attractive but necessary characteristics to improve the security, resiliency and transparency of smart grid networks.

Mylrea el al. [38] explore the application of Blockchain and smart contracts to improve smart grid resiliency and to secure DER transactions and exchanges. The authors state the proposed application would lessen or even remove the reliance upon third parties, strengthen the security of transactions and exchanges resulting in easier adoption and monetisation. Furthermore, it will increase speed, scale and security of modern grid, which are essential properties for real-time energy transactions. In their model, energy transaction is through blockchain based smart meters interfaced via smart contracts which will update the blockchain. The Blockchain would be used to verify time, user, energy transaction and protect the data with immutable crypto signed ledger.

The authors further provide a brief overview of two testbeds at PNNL that are currently being developed:

1. PNNL B2G Cyber Testbed [41]: provides the capability to model and simulate energy delivery systems form the distribution substation all the way to end consumers. Furthermore, it simulates various cyber-attack scenarios, threats and vulnerabilities.

2. PNNL Connected Campus: provides the necessary speed and scale to validate and verify Blockchain application.

The authors state that combining the two PNNL testbeds mentioned above may improve the state of the art blockchain application to security, speed and sale for transactive energy applicaitons.

Gao et al. [39] introduce sovereign blockchain technology, named GridMonitoring to provide a monitoring system on the smart grid. Their sovereign technology contains multiples threads of blockchain in the network, each thread uniquely identifying a consumers identity. Threading side blocks to their parent block is to maintain a contiguous well-ordered log of requests by different consumers. The content of the side blocks appended to their parent blocks are reports from the smart contracts. The significance of such structure is to help maintain an effective log and efficient retrieval of blocks emphasising querying and investigation for the occurrence of the breach of terms by consumers and utility companies.

The GridMonitoring platform is based on four layers:

1. Registration and Authentication Layer

2. Smart meter

3. Processing and consensus nodes

4. Data processing on the smart grid network

The authors claim the GridMontitoring system is very efficient as the user can monitor how the electricity is used, and it also provides a platform where there is no manipulation from either party. A further claim is that their GridMonitoring

platform possesses efficient data manageability whereas the platform of Mylrea et al. [38] does not.

Liang et al. [40] propose a new, distributed blockchain-based protection framework to enhance the self-defensive capability of modern power systems against cyber-attacks. It can resist against data manipulation that is launched by cyber attacks such as Field Data Injection Attacks (FDIA) [28]. To guarantee data accuracy, their framework employs a consensus mechanism which is automatically implemented in every smart meter and has the following characteristics:

1. Setting of public/private key update frequency
2. Block generation
3. Miner selection
4. Release of meter's memory periodically

In this section, we have reviewed key literature on the cybersecurity in the smart grid space. Although the proposed blockchain frameworks reviewed in this section can provide with transparency, immutability and non-repudiation, the simulation environment in the reviewed literature is either in progress, lacking or non-existent.

Liang et al. [40] IEE 118-bus system simulation focuses on cyber attacks on sensors; furthermore there is no significant simulation present on the effects of the attacks on the smart grid network as a whole.

In the case of Mylrea et al. [38], the testbeds mentioned are Discrete Event Simulation (DES) based. DES requires a mathematical model of operation, in a complex systems such as the smart grid, one requires more than just a mathematical model to simulate the every growing complex interaction between various heterogeneous stakeholders. In DES one could formulate how a SCADA operates or for a formula for the energy profile; however, DES cannot formulate the complex interactions and behaviours between stakeholders in the smart grid environment. Additionally, the authors propose the need for threat models in the system; however, no specific details are provided.

We propose an ABM approach to simulate stratified levels of interactions between heterogeneous stakeholders in the smart grid with threat modelling. ABM is justified because we do not have a quantitative solution of the whole stakeholder community and how agents evolve in the network. With ABM we can place knowledge of agents in the model. Granted many cyber attacks can be carried out, our paper will only focus on data manipulation of malicious user's requests and the response of possible tampering of data.

To further strengthen the justification of preferring ABM over other DES or System Dynamics (SD) when simulating a smart grid network, Borshchev et al. [44] states that ABM is designed to go beyond the limits of SD and DES approaches, especially in the case where the "system being modelled contains active objects with timing, event ordering or other kinds of individual behaviour". Macal et al. [50] provide further justification. The authors state that "ABS allows people to model their real-world systems of interest in ways that were either not possible or not readily accommodated using traditional modelling techniques, such as DES or System Dynamics (SD)".

Table 3 shows the comparison of various metrics of key literature reviewed.
The metrics include:

- Information sharing: ability of utility companies to make data generated available to third parties for research purposes
- Data Immutability: if data is unalterable
- Data Integrity: ability to detect unauthorised modifications to data
- Data confidentiality: how secure the data on the system is against intrusion
- Data provenance and auditing: the ability to track changes that are made to data, where data originates and moves to, and who makes changes to it over time
- Consumer demand forecast: ability of consumer to send forecasts to its utility company
- Simulation: simulation and modelling techniques employed

- Threat Modelling: ability to model threats into the simulation model and study the behaviour of the network
- Agent Based Model: if ABM was used as a modelling and simulation technique.

Data confidentiality in blockchain would be dependent on the nature of the blockchain, whether it is permissioned or permissionless. With permissionless blockchain, every single historical transaction is recorded an can be viewed by any entity without special permission allowing for accountability and transparency. This is contrasted with a permissioned blockchain, where data can only be read and written by users with the required access control.

In addition to data immutability, data integrity, data provenance and auditing which is offered by blockchain technology, our ABM supports consumer demand forecasts and a simple threat model which infects end-user's devices resulting in tampered data before transmitting to the blockchain network; this allows us to study the interactions between agents through smart contracts and the evolutionary behaviour of the network.

Metric	Ye et al. [36]	Rusitschka et al. [37]	Mylrea et al. [38]	Gao et al. [39]	Liang et al.[40]
Information Sharing	Y	Y	Y	Y	Y
Data Immutability	N	N	Y	Y	Y
Data Integrity	Y	Y	Y	Y	Y
Data confidentiality	Y	Y	Y	Y	Y
Data provenance and auditing	N	N	Y	Y	Y
Consumer side demand forecast	N	N	Y	N	Y
Simulation	N	N	Y(PNNL testbeds)	N	Y(IEEE-118 benchmark system)
Threat Modelling	N	N	Y	N	Y
Agent Based Model	N	N	N	N	N

Table 3: Comparison of metrics between various papers in the cybersecurity and modelling space of smart grids

3 Methodology

3.1 Operational Modelling

3.1.1 Introduction

Modelling allows one to solve problems that occur in the real world; it is applied when experimenting with real system becomes expensive or impossible; such is the case in smart grid networks. Furthermore, constructing a model can prove useful in achieving a greater understanding of complex systems.

Borshchev et al. [44] state that "Modeling includes the process of mapping the problem from the real world to its model in the world of models, – the process of abstraction, – model analysis and optimization, and mapping the solution back to the real system.". Computer modelling and simulation relates to the manipulation of a computational model in order to enhance the analysis of systems' behaviour and to assess strategies for its functioning in the descriptive or predictive modes.

Many different simulation techniques can be employed depending on the application domand and the nature of the components one is modelling in the domain. Figure 10 shows the three most common ones and the core components used to build the respective simulation. Each of these techniques has its advantages and limitations, and it is vital that the modeller picks the methods which best reflects the system.

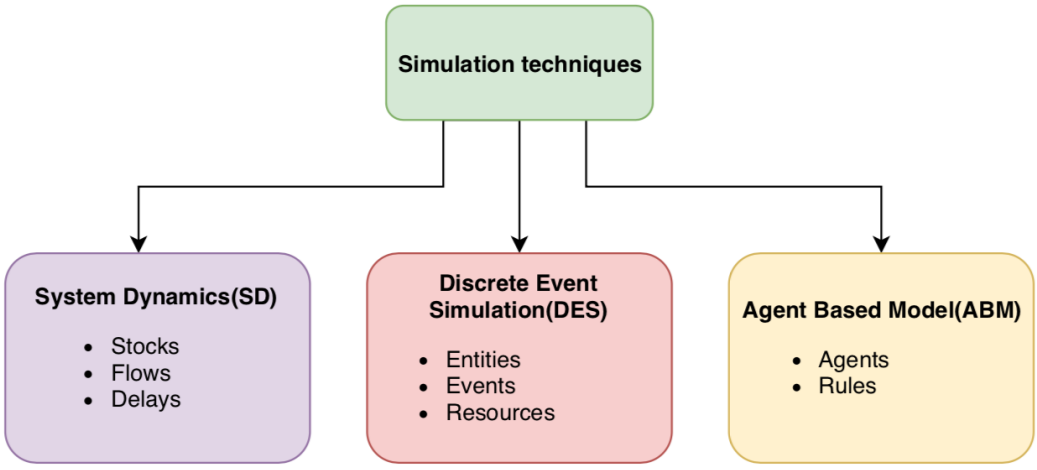


Figure 10: Simulation techniques

Now, we will briefly describe each of the simulation techniques mentioned above [76].

Discrete Event Simulation (DES) models are built using:

- Entities - The general name for the objects that move through the system
- Events - The processes which the entities pass through
- Resources - Objects which are needed to trigger events

System Dynamics (SD) focuses on flows around networks rather than queueing systems such as in DES, it considers:

- Stocks - basic stores of objects
- Flows - define the movement of objects between different stocks in the system
- Delays - delays between the measuring and then acting on that measurement

Agent Based Model (ABM) is a relatively new simulation technique. ABM consists of:

- Autonomous Agents - These are self-directed objects which move about the system
- Rules - which the agents follow to achieve their objectives

Table 4 presents a comparison between the two most widely adopted modelling techniques: DES models and Agent Based Simulation models.

DES models	***ABS models***
Process oriented (top-down modelling approach); focus is on modelling the system in detail, not the entities	Individual based (bottom-up modelling approach); focus is on modelling the entities and interactions between them
Top-down modelling approach	Bottom-up modelling approach
One thread of control (centralised)	Each agent has its own thread of control (decentralised)
Passive entities, that is something is done to the entities while they move trough the system; intelligence (eg, decision making) is modelled as part in the system	Active entities, that is the entities themselves can take on the initiative to do something; intelligence is represented within each individual entity
Queues are a key element	No concept of queues
Flow of entities through a system; macro behaviour is modelled	No concept of flows; macro behaviour is not modelled, it emerges from the micro decisions of the individual agents
Input distributions are often based on collect/measured (objective) data	Input distributions are often based on theories or subjective data

Table 4: Attributes that define DES and ABS models [50]

3.1.2 Agent Based Model(ABM)

3.1.2.1 Introduction

In ABMs, a complex system is represented by a collection of agents that are programmed to follow some behaviour rules [47]. The components of an ABM are a collection of agents and their states, the rules governing the interactions of the agents and the environment within which they live [78]. Through constituent agent interactions, system properties will emerge(see Emergence in Background Section).

Bradshaw [46] defines agents as "objects with attitudes". More formally, agents are software entities, described through computer algorithms to mimic the behaviour of their real-world counterparts. Agents have their own set of goals, behaviours, and thread of control.

3.1.2.2 Suitability

ABMs take agents and their interactions as central modelling focus points. Of the presented modelling techniques(see Figure 10), ABM is the only one that is adaptive, generative and multiformal [79].

ABMs conceptualise the world as resulting from the interactions of many different entities. The algorithmic nature of agents means that they can encode many different formalisms. Analytical solutions to agent interactions, however, are often impossible. Furthermore, ABMs are modular in nature, which allows for different formalisms to be encoded(multiformal). [79]

Agent-Based approach is a more general and powerful approach for modelling complex systems compared to other techniques since it allows capturing more complex structures and dynamics.
Borshchev et al. [44] state that "you may know nothing or very little about how things affect each other at the aggregate level, or what is the global sequence of operations, etc., but if you have some perception of how the individual participants of the process behave, you can construct the ABM and then obtain the global behaviour." In other words, it provides for constructing models in the absence of knowledge about global interdependencies.

Macal et al. provide a list of features for when an ABM is a good fit for the specific problem [50], here we have stated a few:

- when agents have relationships with other agents, especially dynamic relationships—agent relationships form and dissipate, for example, structured contact, social networks;
- when it is important that agents learn or adapt, or populations adapt;
- when agents engage in strategic behaviour and anticipate other agents' reactions when making their decisions
- when it is important to model agents that cooperate, collude, or form organisations;
- when scale-up to arbitrary levels is important that is, extensibility

3.1.2.3 Structure

Figure 11 presents the typical structure of an agent in an ABM, the three basic elements are [48]:

1. A set of agents, their corresponding attributes and behaviours.
2. A set of agent relationships and methods of interaction: An underlying topology of connectedness defines how and with whom agents interact.
3. The agents' environment: Agents interact with the environment in addition to other agents.

3.1.2.4 Process of Modelling

In general, ABM follows an incremental modelling process, starting with a simple model and evolving into a more complex model. Figure 12 shows the processing of agent-based modelling

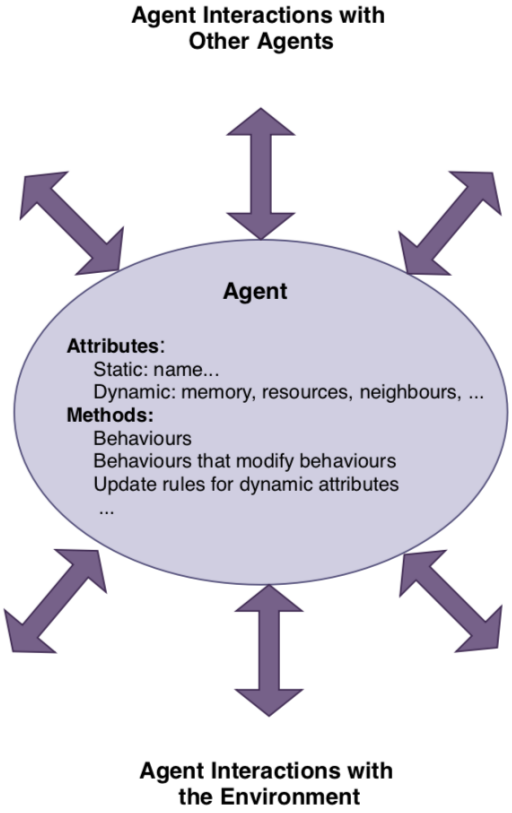


Figure 11: A typical agent structure[48]

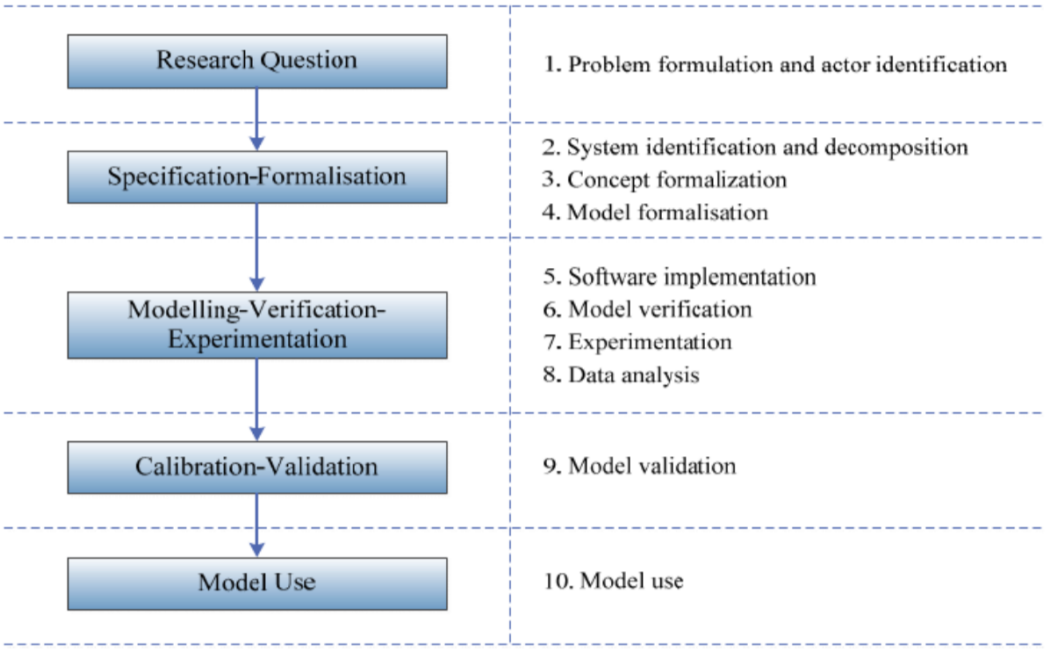


Figure 12: Process of modelling an Agent Based Model(ABM) [77]

3.1.3 Conclusion

In this section we briefly described what modelling is, the main simulation techniques, we then delved into details of ABM: its suitability, structure and the process of modelling.

3.2 Cybersecurity

3.2.1 Introduction

Both physical and electronic ledger are typically maintained by third parties creating a "trusted" environment that could be abused and manipulated by human actions such as tampering and destroying sensitive records and double spending. When we place our trust in third parties, we are exposing ourselves to the possible misbehaviour of that party.

Currently, when a renewable-power plant generates electricity, a meter outputs data which gets logged in a spreadsheet. This spreadsheet containing the output generation of the renewable generator is sent to a registry provider, where data logged into a new system, and a certificate is created. Another set of intermediaries brokers deal between buyers and sellers of these certificates, and yet another intermediary verifies the certificate after they are purchased [73].

Adversaries can launch attacks by hacking remote terminal units(RTUs) such as sensors placed in substations or smart meters in smart homes which can compromise physical data measurements and state estimations leading to distorted energy demand and supply figures. Data tampering can be launched by consumers to reduce their energy bills, competing corporations, or hostile countries, aiming to compromise data measurements to increase the cost of energy distribution and smart grid operations or even causing large scale blackouts. In this sense, data vulnerability has become an unneglectable issue.

Grid optimisation and resilience improvements are essential operations as we modernise the power grid. However, cybersecurity often is an afterthought for vendors and consumers as they prioritise functionality and cost, leaving the power grid vulnerable to cyber-attacks. Mylrea et al. [38] claim that Blockchain may help solve several optimisation and reliability challenges that have been introduced with the modernisation of the grid. The authors believe that applying blockchain based smart contracts presents an "opportunity to increase the speed, scale and security of transactive energy applications".

Figure 13 by the US Department of Homeland Security (DHS) provides a decision model which suggests the system designer to look at the nature and function of a data set used by the stakeholders and decide whether the data set will be safe in a spreadsheet, in a conventional database, an encrypted database, a managed database or to use blockchain technology to secure the system.

In the case of smart grids, adopting blockchain could be a potential option in securing parts of the system. By following Figure 13, we provide the following reasons that lead to the blockchain use case for smart grids:

1. The interactions between stakeholders involve physical and informational transactions. The smart grid stakeholders require a shared, consistent data store. There is a need for historically consistent data for energy transactions, energy consumption logs, pricing, statistics in the smart grid network.

2. There are multiple writers in the smart grid system. For example, a consumer needs to write to the blockchain for the amount of energy it requires, a supplier needs to record the amount of energy supplied and the price charged to its customers and a Distribution System Operator(DSO) needs to record the amount charged to the stakeholders for carrying the energy to buildings and homes.

3. The smart grid network requires immutable records for auditing and non-repudiation, once recorded no modification of historical data should be allowed.

4. Storing of sensitive identifiers is not required in order for the smart grid to function properly.

5. The smart grid stakeholders may not place their full trust on other stakeholders in the grid to be responsible for running the data store. One cannot trust an online third party in the smart grid system; transparency and decentralisation are required to secure the network. The intentions of all the smart grid stakeholders cannot be fully trusted, from honest mistakes to deliberate, malicious tampering of data.

6. The smart grid stakeholders need a tamperproof log of all the writes performed by all stakeholders to audit what happened and when it happened.

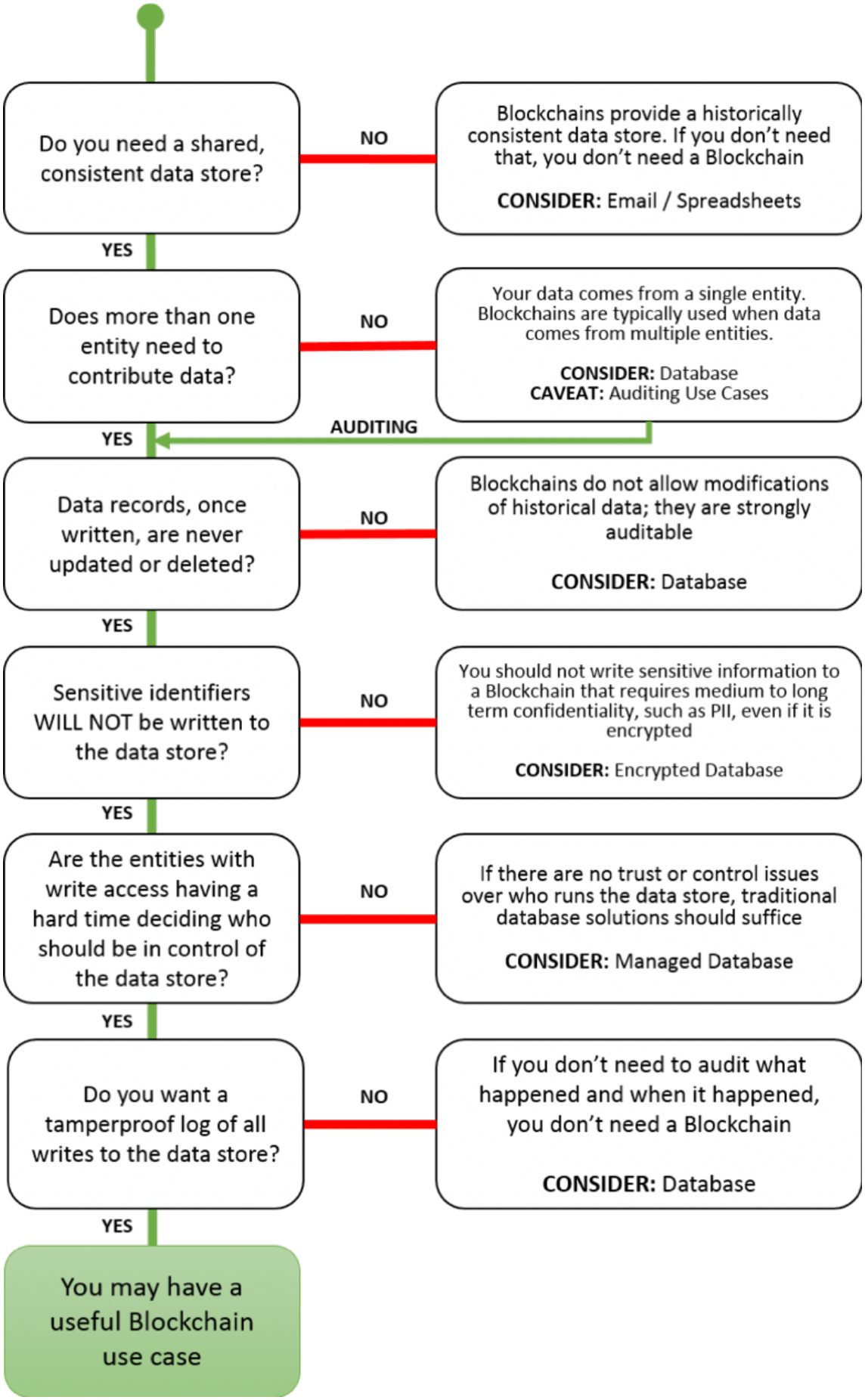


Figure 13: DHS Science and Technology Directorate Blockchain decision model[57]

3.2.2 Blockchain

In 2008, Satoshi Nakamoto published a white paper [53] that presented a detailed method for using a decentralised, encrypted digital ledger that solves third-party abuse and vulnerability issue.

Davidson et al. [55] define blockchain as "[a new digital technology that combines peer-to-peer network computing and cryptography to create an immutable decentralised public ledger.]". The real novelty of blockchain lies in its ability to reach consensus about the correct and true state of a ledger without relying on central authorities or intermediaries such as auditors and exchange markets.

Most Blockchain technology at its core are composed of three elements [56]:

- Peer-to-Peer Networking: a distributed application architecture that partitions the task between peers. Peers can communicate with each other without relying on central authority.
- Asymmetric cryptography: an encryption scheme that uses two mathematically related, but not identical, keys - a public key and a private key. In Bitcoin and Ethereum, asymmetric cryptography is used to create a set of credentials for your account, to ensure that only you can perform the transactions.
- Cryptographic hashing: a method for generating a unique 'fingerprint' of any data, allowing quick comparison of datasets and verifying that data has not been altered.

3.2.2.1 Blockchain Structure

As shown in Figure 14, the blockchain structure consists of a sequence of blocks which are linked together by their hash values. A block comprises of multiple transactions(TX1,TX2,..TXN). The blockchain is extended by each additional block and hence represents a complete ledger of transaction history. In addition to the transactions included in the block, each block contains a timestamp, the hash value of the previous block and a nonce which is a random number for verifying the hash. This is the core concept which ensures the integrity of the entire blockchain through to the first block(genesis block). Hash values are unique, and tampering can be effectively prevented since changes of a block in the chain would immediately change the hash value of the block.

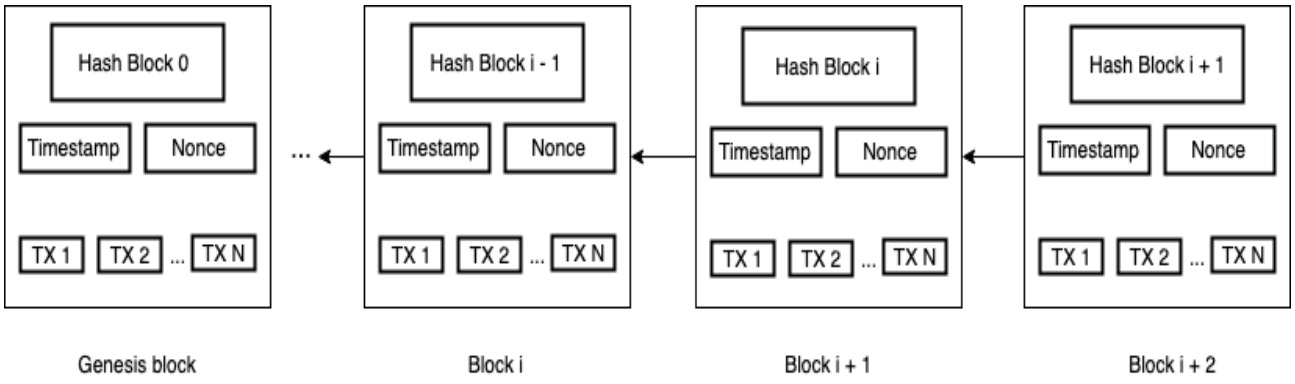


Figure 14: Blockchain structure [62] [64]

3.2.2.2 Digital Signature Process

Figure 15 shows the process of signing and verification in a blockchain network.
The private key is used to sign the transaction data. The digitally signed transactions are spread throughout the whole network, and the signed transaction is accessed by public key, which is visible to everyone in the network. When user Alice wants to sign a transaction, she first generates a hash value derived from the transaction. She then encrypts this hash value by using her private key and sends to another user Bob the encrypted hash with the original data. Bob verifies the received transaction by comparing the decrypted hash (by using Alice's public key) and the hash value derived from the received data by the same hash function as

Alice's. The typical digital signature algorithms used in blockchains include Elliptic Curve Digital Signature Algorithm (ECDSA) [62].

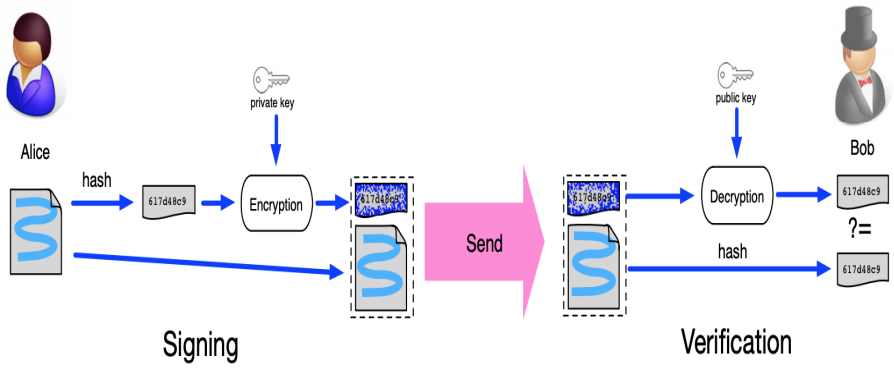


Figure 15: Digital signature in Blockchain [62]

3.2.2.3 Working Methodology of Blockchain

According to Swanson [54], consensus mechanism "is the process in which a majority (or in some cases all) of network validators come to agreement on the state of a ledger. It is a set of rules and procedures that allows maintaining coherent set of facts between multiple participating nodes".

To append a block to the chain, the majority of nodes in the network must agree by a consensus mechanism on:

- the validity of transactions in a block
- the validity of the block itself

Figure 16 shows the how a transaction is processed. When a user joins a network, it will own a pair of public and private key. First, a user signs a transaction using its private key and broadcasts it to its peers. The cryptographic signature of the transactions will ensure non-repudiation in the blockchain distributed ledger. Once the peers receive the transaction in the network, they will validate the transaction and broadcast it over the network. The parties involved in the transaction validate the transaction to meet a consensus agreement. Once a distributed consensus

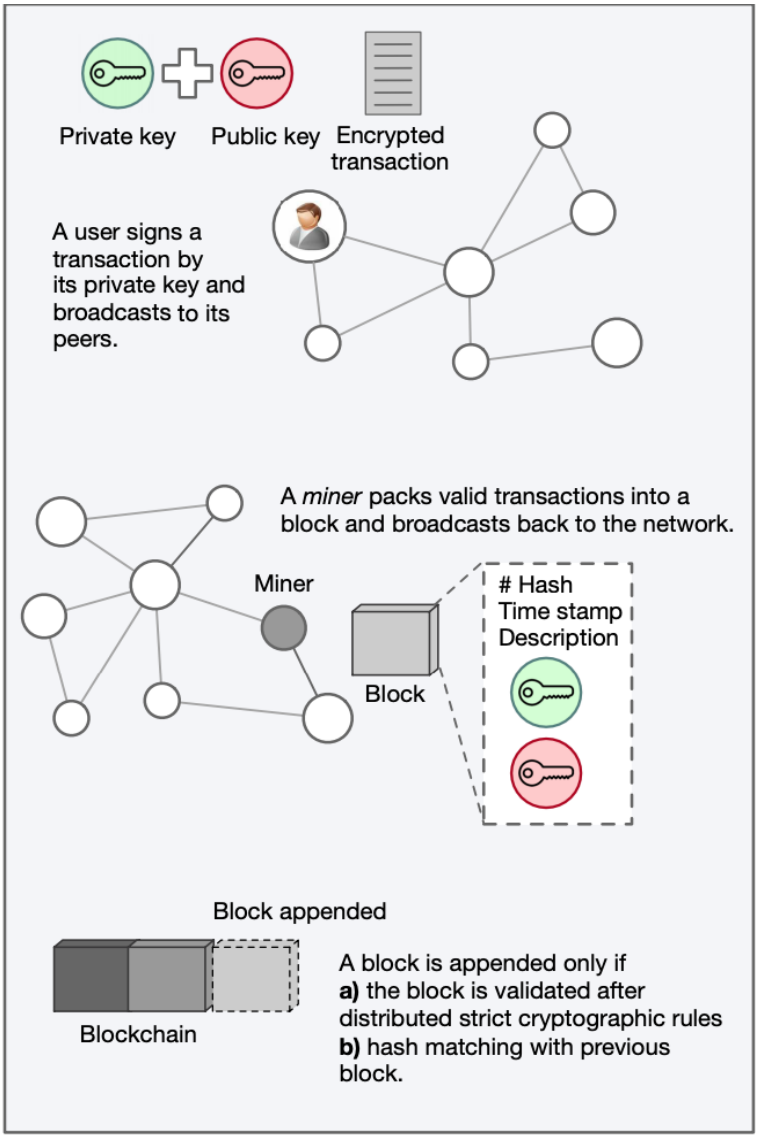


Figure 16: Blockchain working methodology [63]

is reached, the miner includes the valid transaction into a timestamped block and broadcasts it back into the network. After validating the broadcasted block containing the transaction and matching the hashes with the previous block, the block is finally appended to the blockchain.

3.2.2.4 Smart Contracts

The term 'Smart Contract' was first coined by Nick Szabo in 1994, defining it as "a computerized transaction protocol that executes the terms of a contract" [65].
Szabo suggested translating contractual clauses into code and embedding them into hardware or software that can self-enforce them, to minimise the need for trusted intermediaries between transacting parties, and the occurrence of malicious or ac-

cidental exception [66]. With the advent of blockchain technology, we can realise the benefits of Smart Contracts since they can be utilised more efficiently by applying blockchain technology compared to the technology present at the time of their invention in 1994.

Yaga et al. define a smart contract as "[a collection of code and data (sometimes referred to as functions and state) that is deployed using cryptographically signed transactions on the blockchain network]" [57]. A smart contract aims to improve the observability, verifiability and enforceability of a contract.

Some claim a smart contract has the following properties [72]:

- Automatically executable
- Enforceable
- Semantically sound
- Secure and unstoppable

A smart contract can reside on the blockchain, and hence its code can be inspected by every participant node in the network. Furthermore, since all the interactions with a contract occur via signed messages on the blockchain, all the network participants receive a cryptographically verifiable trace of the contract's operations. Any attempts to change the smart contract will be rejected, and the stakeholders will be automatically notified of the attempted modification. Hence, in a multi-party scenario, it can provide attestable data and transparency resulting in confidence among the stakeholders in the enforcement of the rules embedded in the smart contract, reduced costs from reconciliation that exists in traditional business applications and reduce time to complete a transaction.

3.2.2.5 Blockchain Network Overview

Figure 17 shows the network view of a blockchain. The Internet at the bottom layer provides with the basic communication layer for any network. In this case, a peer-to-peer network runs on top of the internet, which hosts another layer of blockchain.

That layer contains transactions, blocks, consensus mechanisms, state machines, and blockchain smart contracts. All of these components are shown as a single logical entity in a box, representing blockchain above the peer-to-peer network. Finally, at the top, there are users or nodes that connect to the blockchain and perform various operations such as consensus, transaction verification, and processing.

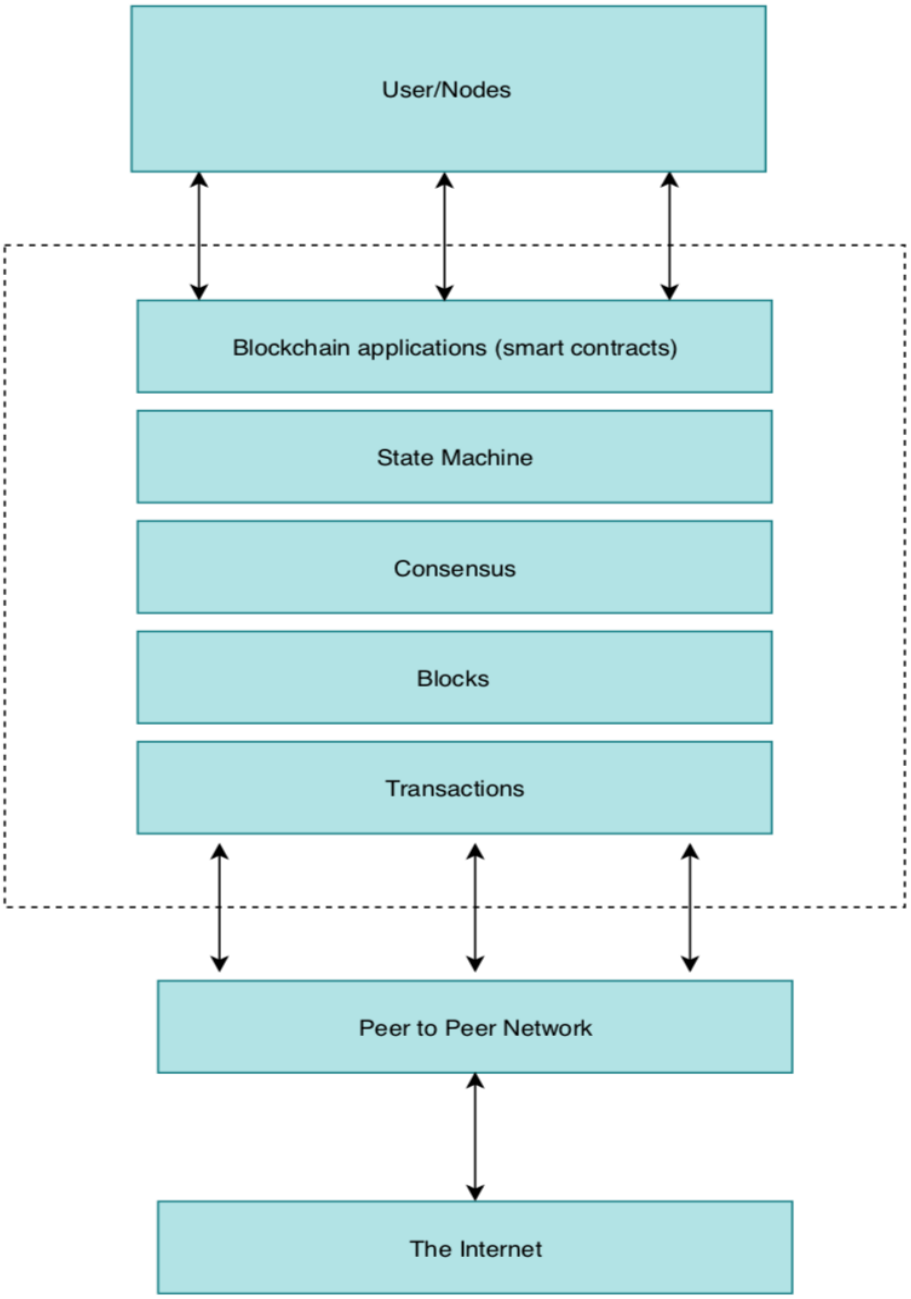


Figure 17: Blockchain network view [72]

3.2.3 Conclusion

In this section, we started with claim from Mylrea et al. [38] on applying blockchain based smart contracts to increase speed, scale and security of transactive energy applications.We then provided with key points to justify why blockchain technology may be a possible solution to enhance cybersecurity in smart grid with the aid of the U.S DHS decision model. We then briefly provided with an overview of blockchain: its characteristics, its structure, the working methodology of blockchain, smart contracts and the network overview.

3.3 Design

Figure 18 shows the proposed architecture for our program which integrates ABM and blockchain technology. We will have a user interface, which controls the environment and the agents. Agents will have their own knowledge base which they refer to take actions. The agents interact with the blockchain network, more specifically smart contracts to carry out operations to reach their goals. To understand how the smart grid network evolves, the visualisation tool retrieves information from agents and the blockchain network and displays the important aspects the modeller may be interested in.

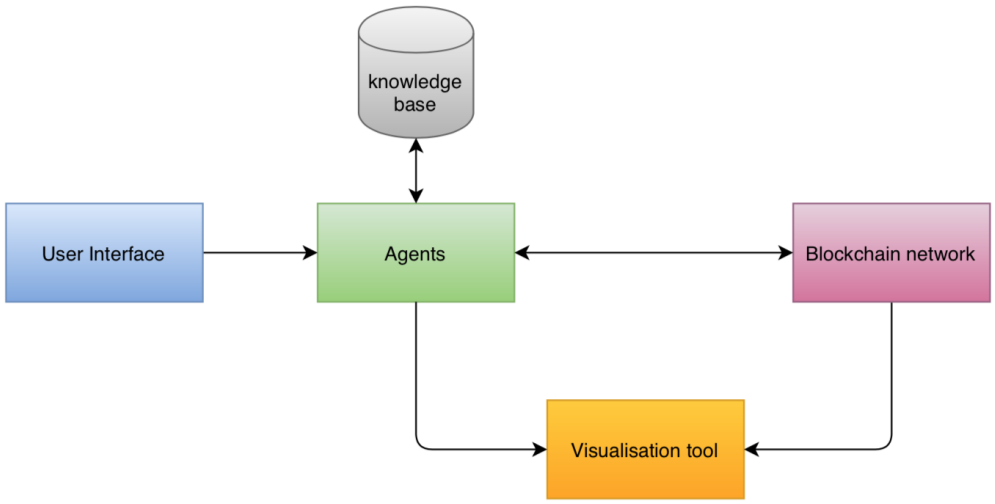


Figure 18: Design of program consisting of user interface, agents, blockchain network and visualisation tool

Mylrea et al. [38] provides a detailed blockchain architecture diagram of the smart grid space, shown in Figure 19. The consumer can transact energy with prosumers(producer and consumer) with DER such as solar panels, with Electric Vehicles(EVs) and primary producers/generators.

As consumer transact electricity, the blockchain based meter updates the distributed ledger with the units of electricity transacted, creating a unique time-stamped block for verification in a distributed ledger. The blockchain can be leveraged by Distribution System Operator(DSO) to receive energy transaction data to charge their

network costs to consumers or utility company. Similarly, Transmission System Operator(TSO) can benefit by blockchain as it would reduce data requirements and increase the speed of transaction clearance since transactions could be executed and settled by actual consumption.

Smart contracts execute and record the transaction on the ledger through blockchain enabled smart meters. These smart contracts can facilitate consumer trading excess energy within the microgrid, providing additional energy storage and help substations load balance from primary energy generators. Smart contracts can also facilitate energy transactions between consumers and suppliers and also between suppliers and generators.

Moreover, with decentralised storage of all transactions of energy flows and business activities secures the smart contract data. This highlights the disruptive potential for blockchain on energy markets through the introduction of a more autonomous and decentralised transaction model. This peer to peer system may reduce or even replace the need for a meter operator if the distribution system operator is provided with access to the blockchain network.

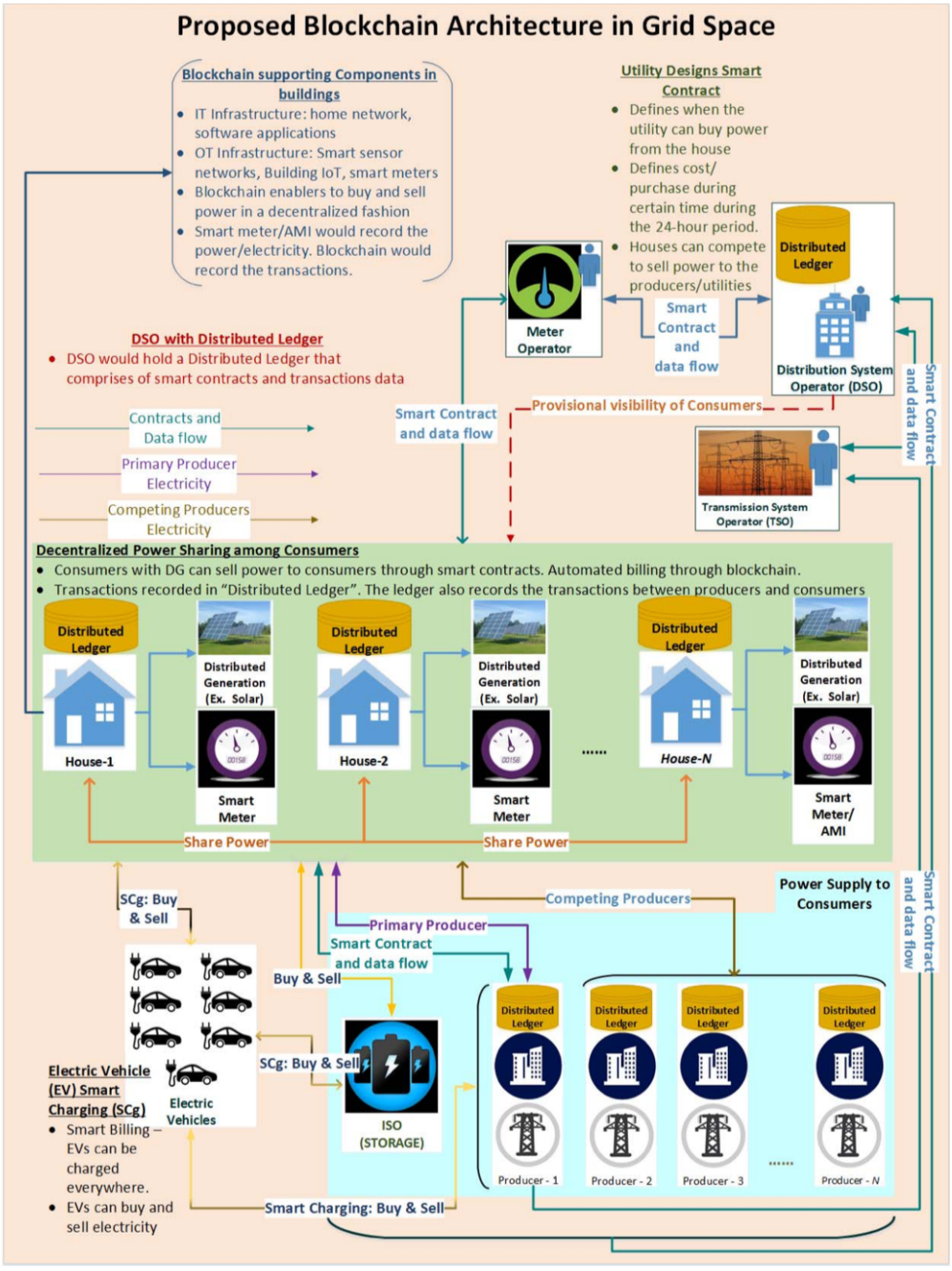


Figure 19: Proposed Blockchain Application to Electricity infrastructure [38]

From the proposed architecture by Mylrea et al. [38], see Figure 19, multiple potential agents could be modelled into a smart grid network as shown in Figure 20

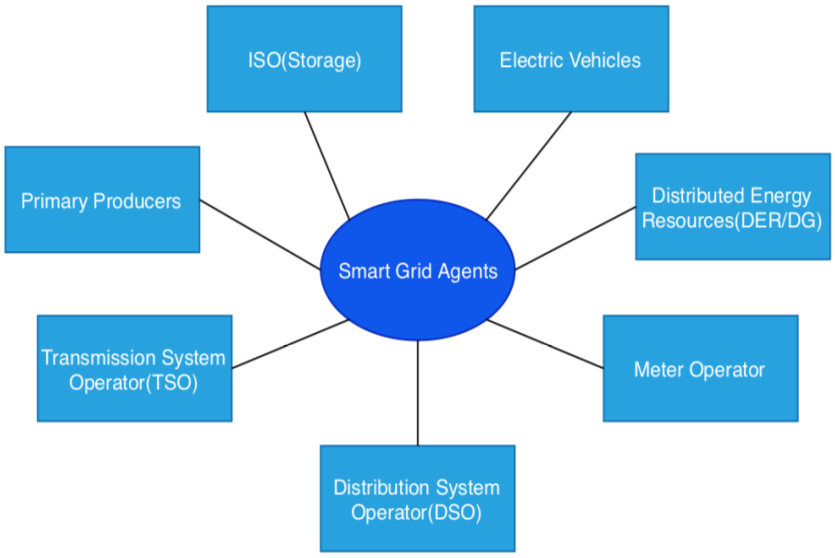


Figure 20: Potential Agents identified in Mylrea et al. smart grid architecture [38]

However, we will simplify the architecture into three main stakeholders of the smart grid network, namely:

- Consumer Agent
- Supplier Agent
- Generator Agent

In our model, agents can establish predefined rules between themselves on matters such as requesting energy ,supplying energy, recording energy and the energy pricing mechanism through a smart contract, which is then deployed on the blockchain.

We carry the assumption that the smart home computing platform has the following capabilities:

- the smart home computing platform can forecast energy demands of the inhabitants of the house by analysing patterns of usage and other information sources such as future weather conditions, news and social network, and be able to communicate the energy request to the smart meter.

- there is a wired/wireless communication between the smart home computing platform and smart meter
- with the installation of smart meters, the smart meters will contain a unique address by which it will be asked to join the network and authorised every time it makes interaction with the blockchain network.

Only the meters which are authorised by the grid will be able to carry read/write operations. The interaction among nodes in the network are automatically performed based on the requirement of the stakeholder, i.e. smart home may request more energy on behalf of the residents or record the energy consumption of the residents.

As shown in Figure 21, each node(i.e. the stakeholders: consumers, suppliers and generators) in the blockchain network will have a copy of the distributed ledger. This copy of the distributed ledger will contain transactions made through smart contracts and the smart contracts themselves.

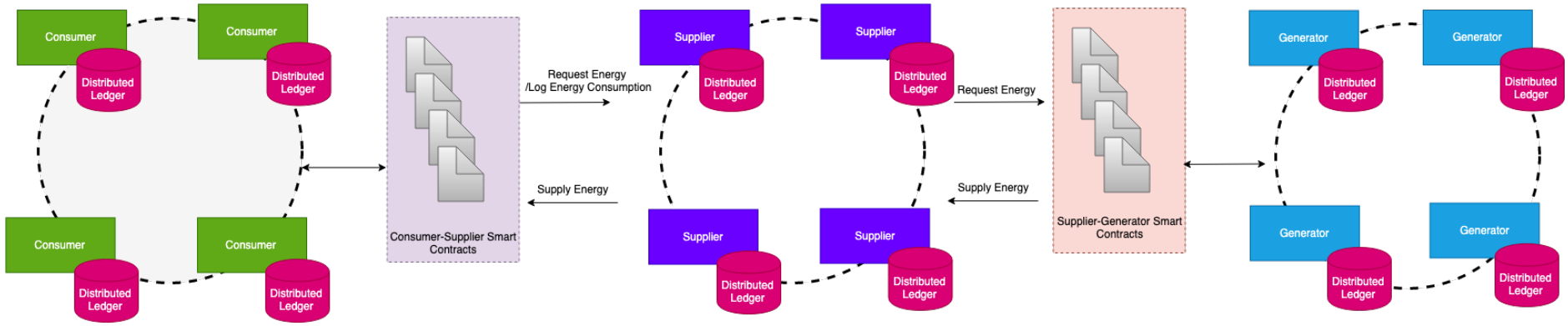


Figure 21: Smart grid overview of the agents with blockchain and smart contracts

Let us suppose we have a consumer named *Consumer A* and a supplier named *Supplier B*. *Consumer A* and *Supplier B* have reached an agreement on various rules on transacting and exchange of energy such as request energy, supplying energy, rewards and penalties. After an agreement has been established, the rules are translated into a computational smart contract code, which is then deployed to the blockchain network, let us call it *ConsumerA-SupplierB* smart contract.

If *Consumer A* wants to request energy from its partnered supplier, *Supplier B*, it will have to send the energy request to the deployed *ConsumerA-SupplierB* smart contract. Each node has a copy of *ConsumerA-SupplierB* smart contract and will self-execute and self-enforce the rules embedded in the smart contract whenever the smart contract is invoked, as shown in Figure 22. As the smart contract is executed, it will take appropriate actions based on input such as penalising or rewarding consumer's request and energy usage.

The smart contract triggers certain events based on the inputs to the smart contract. For example, if *Consumer A* requests energy through the *ConsumerA-SupplierB* smart contract, the smart contract will execute and trigger certain events based on the input provided. For *Supplier B* to be aware of *Consumer A*'s energy request, it will implement a consumer energy request event listener which will listen for requests of its consumers on the network.

For consumers, suppliers and generators to be aware that a transaction has been performed on their respective smart contracts, event listeners should be implemented by each agent, listening for events they are authorised to and interested in. These event listeners will check for specified events on blocks. For example, if *Supplier B* has updated the pricing of energy in kWh through *ConsumerA-SupplierB* smart contract, *Consumer A*'s event listener will notify *Consumer A* about the event that has taken place through the established smart contract, namely *ConsumerA-SupplierB* smart contract.

Figure 23 shows the event listeners for the smart grid blockchain network and the smart grid stakeholders that may be interested in implementing these event listeners. A consumer agent would possibly be interested in monitoring prices charged by the supplier(Consumer Price Event Listener) and the prices that the generator charges

the supplier per kwh(Supplier Price Event Listener). Similarly, a Supplier agent may be interested in listening for event requests by consumers and energy generation events by its partnered generators through the smart contracts established.

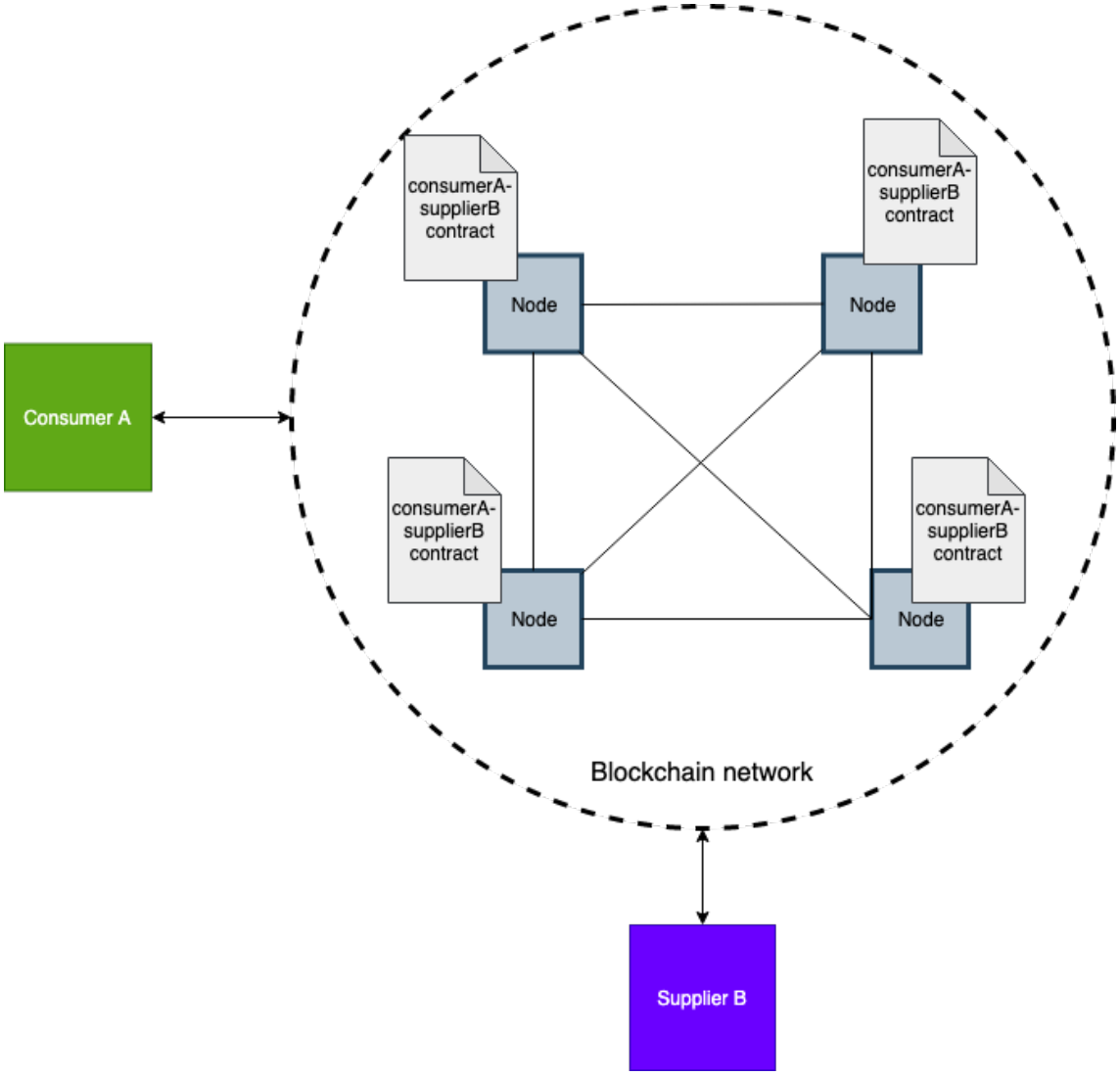


Figure 22: Consumer and Supplier interaction via a smart contract

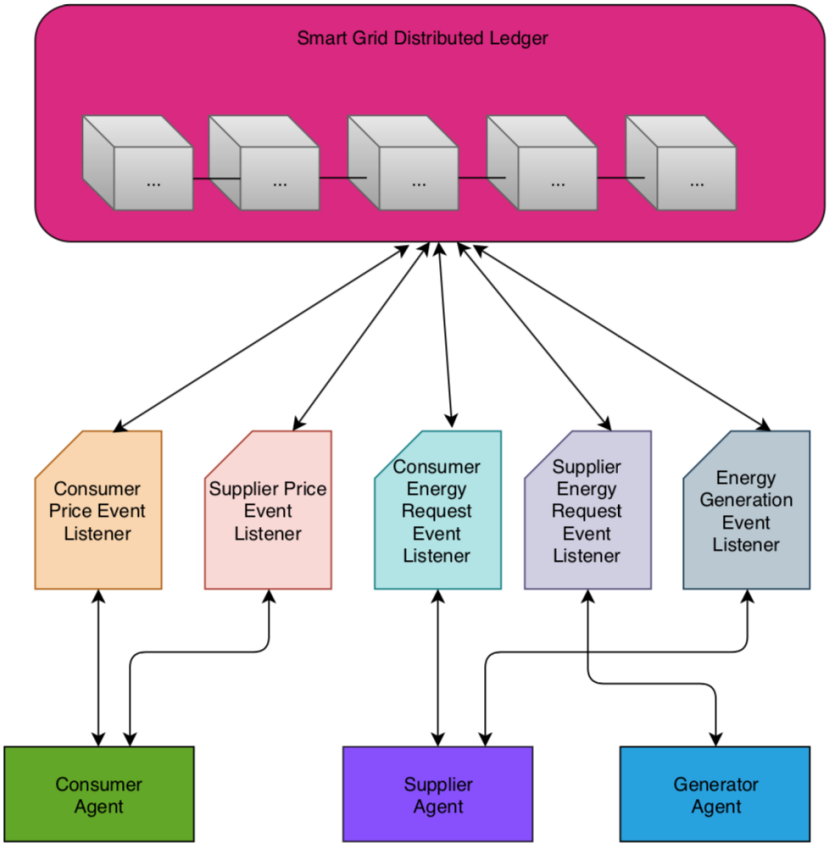


Figure 23: Event listeners for agents in the network

Figure 24 shows the activity diagram for the consumer agent. At each step t in the ABM, the consumer agent forecasts its energy demands for the next step and also records its actual energy consumption through the smart contract for the current step. Once the energy forecast has been made for the next step, it has the option to trade within its microgrid(i.e. with EV's or with prosumers) or trade with its partnered primary supplier with probability p. In our ABM, we are only focusing on the interactions between consumers and suppliers and not between consumer to consumer interactions. Hence in our ABM we use the concept of probability when a consumer agent decides to trade energy.

If the consumer agent decides to trade with its primary supplier(determined by the probability p), it sends the energy request to the smart contract that has been established by both parties and waits for the response from the supplier. The supplier can either accept or reject the request; it can either reward or penalise the consumer for its actions. If the request is rejected, the error is displayed on the smart meter of the consumer and the reason for the denial of the request. If the request is accepted, the price and the energy supplied will be displayed on the smart meter of the consumer.

Figure 25 shows the activity diagram for the supplier agent. The supplier will monitor the blockchain through its appropriate event listeners at each step *t*. At each step t the supplier agent will accumulate the anomaly request events recorded by the smart contract and also accumulate accepted request events by the smart contract and take the appropriate actions accordingly. After dealing with accepted and anomaly requests, the supplier agent will update energy request threshold for its smart contract and update price per kWh for next step *t+1*.

Figure 26 shows the activity diagram for the generator agent. Similar to the supplier agent activity diagram, it will monitor the blockchain at each step and take appropriate actions on the anomaly supplier request events and the accepted supplier request events by the smart contract. It will then update the supplier request threshold and the wholesale price to be charged by the supplier for the next step.

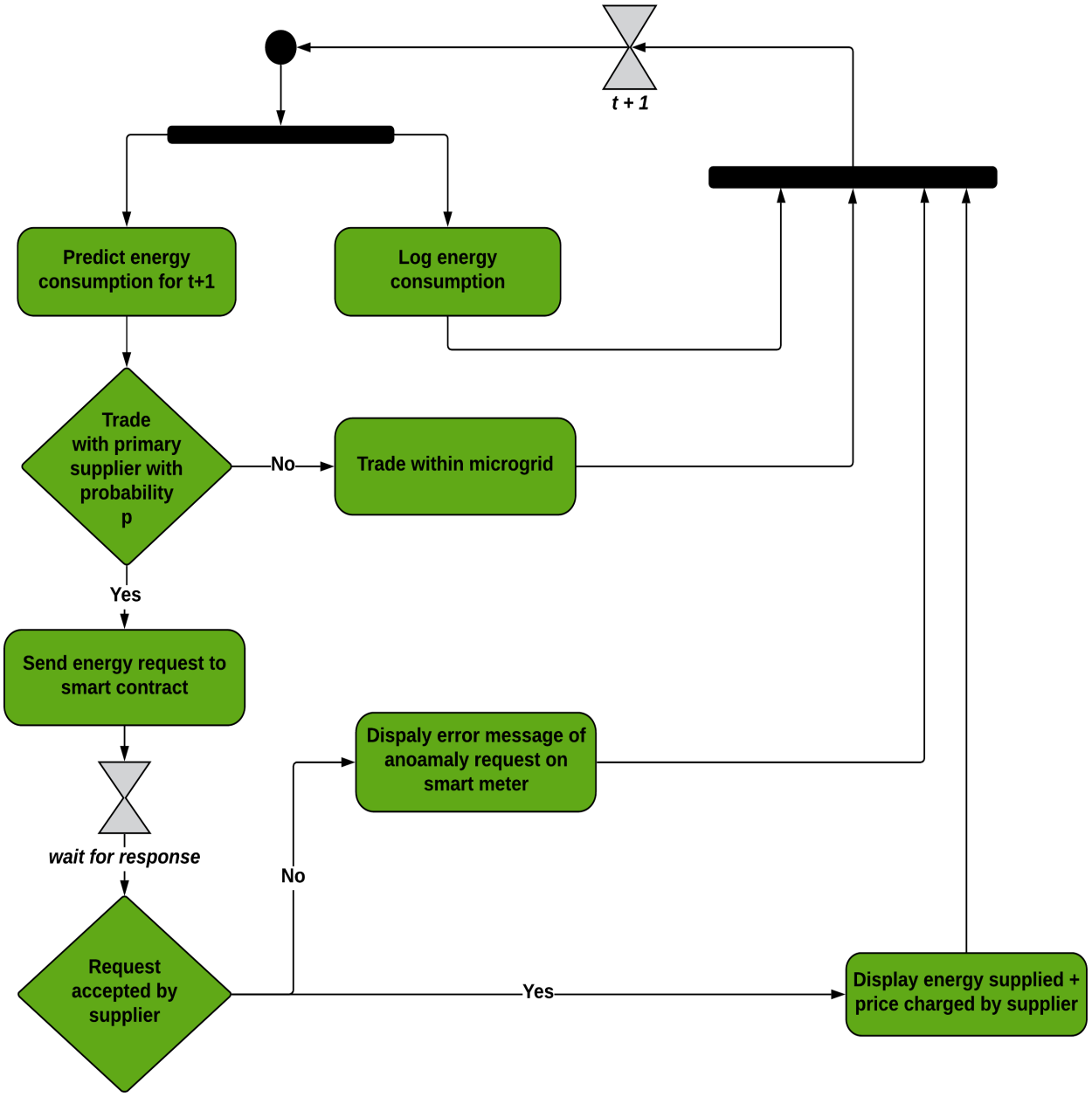


Figure 24: Consumer Agent Activity Diagram

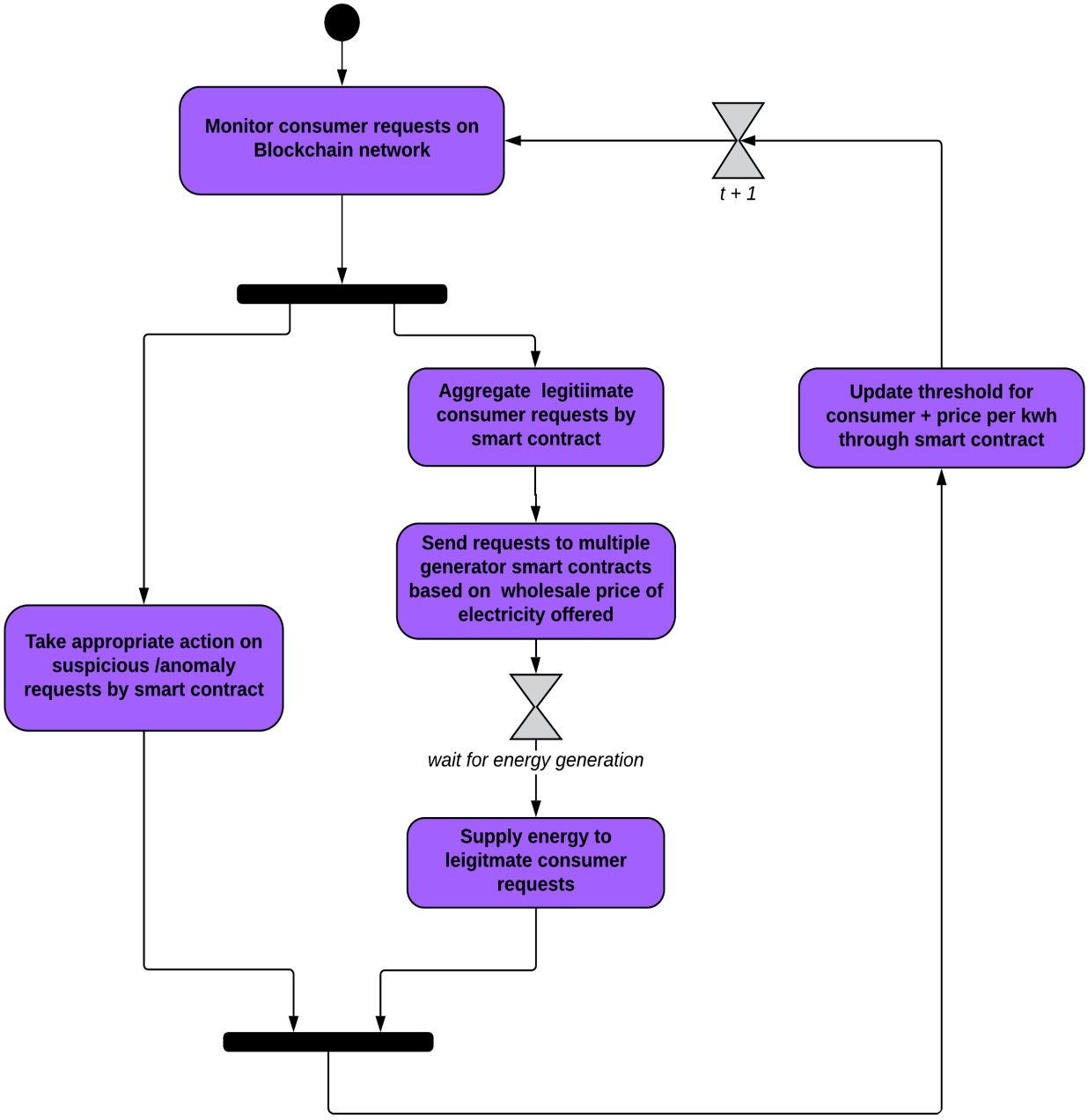


Figure 25: Supplier Agent Activity Diagram

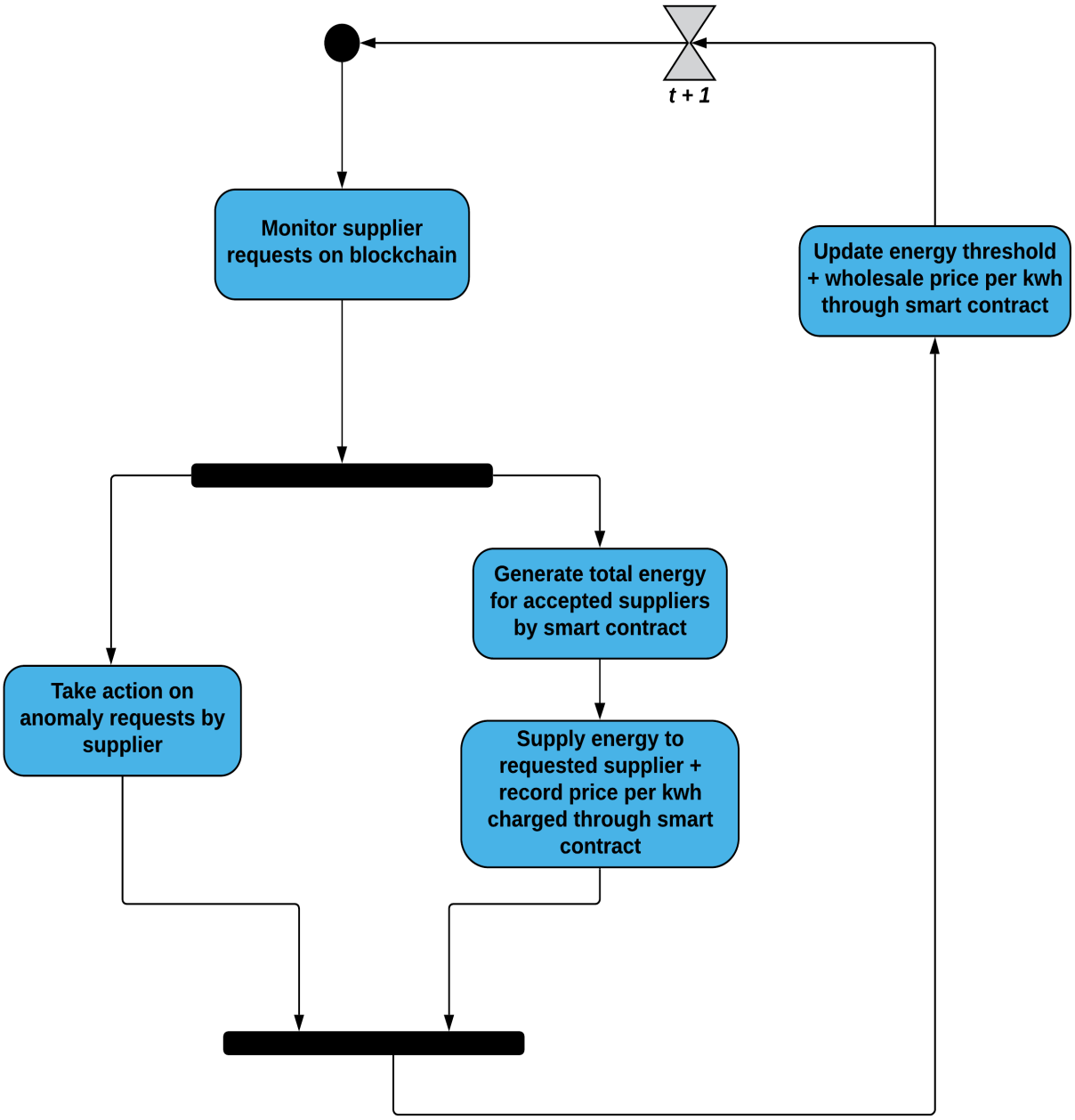


Figure 26: Generator Agent Activity Diagram

3.4 Platforms and Tools

3.4.1 Operational Modelling

Abar et al. [51] present a comparative literature survey of various agent-based tools present, highlighting particular features, advantages and shortcomings of each software tool.

Although NetLogo seems to be the framework of choice for a majority of the ABMs in academic papers, it is implemented in a domain-specific language called NetLogo. The NetLogo language proves to be a significant hindrance when one is attempting to integrate blockchain technology and smart contracts with the agent-based model as there are no frameworks to bridge the NetLogo language with Ethereum, HyperLedger or any other popular blockchain technology.

The preferred framework to implement the agent-based model was Mesa, a python alternative to NetLogo, RePast or MASON. Mesa provides with modular compnents, browser-based visualisation and built-in tools for analysis. Furthermore, the Mesa framework was relatively straightforward to integrate with web3.py(implemented in python)to interact with the smart contracts. Table 5 provides a brief description of the Mesa framework.

ABMS Software Tool ---------- **License / Availability**	**Source Code**	**Type of Agent based on its Interaction Behaviour**	**Coding Language or Application Programming Interface (API) for Model Development** ---------- **Integrated Development Environment (IDE)**	**Compiler** ---------- **Operating System (OS)** ---------- **Implementation Platform**	**Model Development Effort**	**Modelling Strength / Simulation Models' Scalability Level**	**ABMS Scope or Application Domain**
Mesa https://pypi.python.org/pypi/Mesa/ https://github.com/projectmesa/mesa/ ---------- Open source, License: Apache 2.0, Free	Python	Agents as class constructs (having a unique identifier consisting of variable and action)	Python ---------- Visualisations in a browser window, using JavaScript; Result analysis using Python's data interpretation tools	Python virtual environment (virtualenv) ---------- Desktop computer	Moderate	Light-weight / Small-scale ~ Medium-scale	General purpose artificial life related simulations (Basically, Mesa is a Python 3 based alternative to NetLogo, Repast, or MASON)

Table 5: Mesa framework properties [51]

3.4.2 Cybersecurity

The first choice for integrating blockchain technology with the ABM was Hyper-Ledger Fabric as it allowed for permissioned blockchain. However, IBM Cloud charges hourly for each of its worker nodes. The financial factor for this project lead us to use the Ethereum platform. Ethereum was chosen since it is the most mature platform to implement smart contracts and also allows for an easy implementation of smart contracts [61]. Bogner et al. [60] defined Ethereum based smart contract as '[a cryptographic box which stores information, processes inputs, writes outputs and is only accessible to the outside if certain predefined conditions are met.]'

Figure 27 shows the taxonomy of the Ethereum development ecosystem. There are multiple languages to develop smart contracts in Ethereum. The Ethereum community no longer supports low-level Lisp-like Language (LLL) and Serpent, and their usage has almost diminished [72]. We developed our smart contracts in Solidity as it is the most commonly used language in the Ethereum ecosystem [72]. Web3.py [69], a python library was implemented to interact with Ethereum, its API derived from the popular Web3.js Javascript API.

Ethereum found a virtual currency, called Ether, on a blockchain based proof-of-work. Ethereum's ledger is more general that Bitcoin's ledger as it allows to store Turing-complete programs in the form of EVM bytecode, and it enables transactions as function calls into that code, with additional data in the form of arguments. These programmable smart contracts can access non-volatile storage and log events which are both recorded in the ledger [70].

The Ethereum Virtual Machine (EVM) is designed to serve as an isolated and sandboxed runtime environment for smart contracts based on Ethereum. As such, the code that executes on the EVM will not have access to any resources external to the virtual environment, this results in increased security, deterministic execution and allows untrusted code to be run on the Ethereum blockchain [72].

EVM executes bytecode instructions to transform the system state from one state to another. EVM is a Turing-complete machine; however, it is limited by the amount

of gas that is required to run any instruction. Since the execution is gas-bound, infinite loops which can result in denial of service attacks are not possible in the network [72].

Figure 28 shows the smart contract workflow, from translating agreement into a computational smart contract to deployment and invocation of the contract.

Figure 29 show an overview of the workflow of the Ethereum network. Once the smart contract code is written, it is compiled into bytecode that is understandable by the EVM using the Solidity compiler called solc. The EVM bytecode is then deployed to the Ethereum network. The initiator of a transaction pays a fee for its execution, measured in units of gas. The miner that manages to append a block including the initiator's transaction is rewarded a fee in Ether. Therefore, Ethereum can be thought of as a distributed computing platform where anyone can run code by paying for the associated gas charges [70].

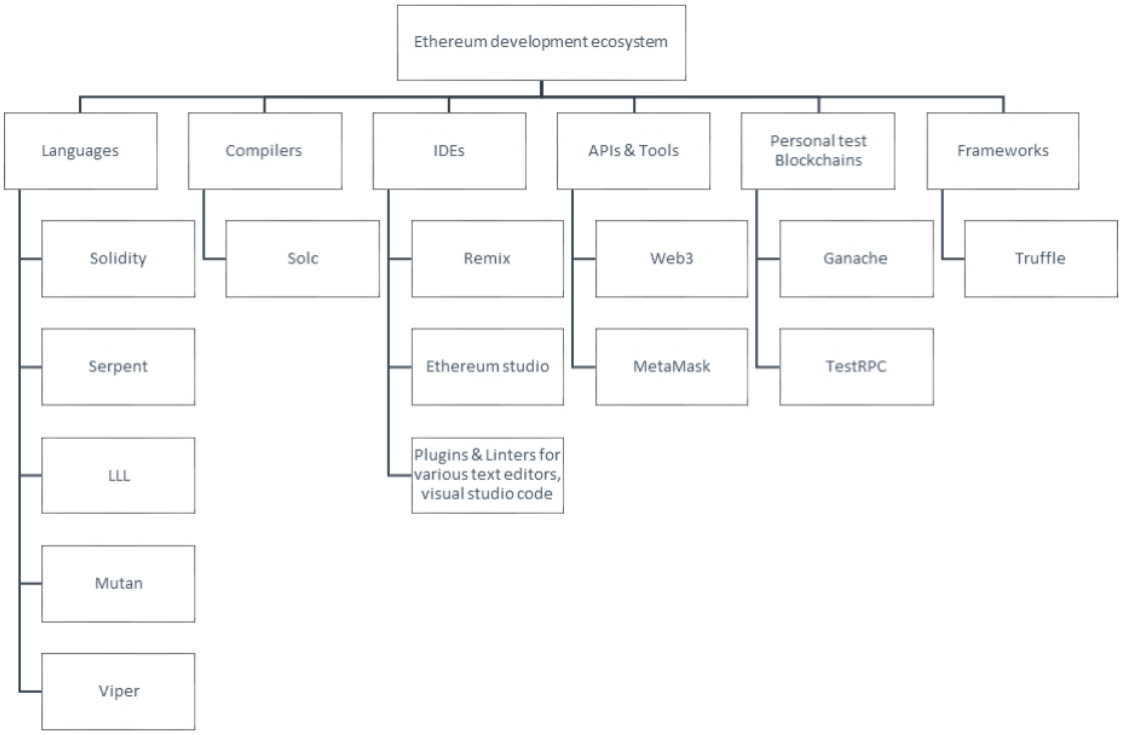


Figure 27: A taxonomy of Ethereum development ecosystem components[72]

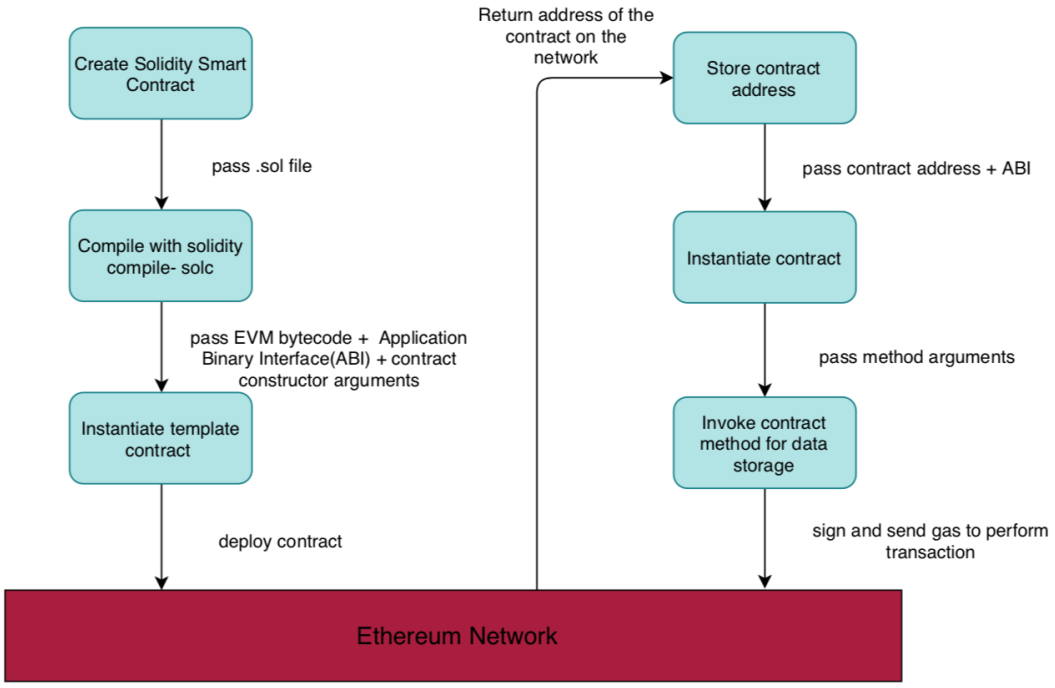


Figure 28: Smart contract workflow from creation to deployment to execution

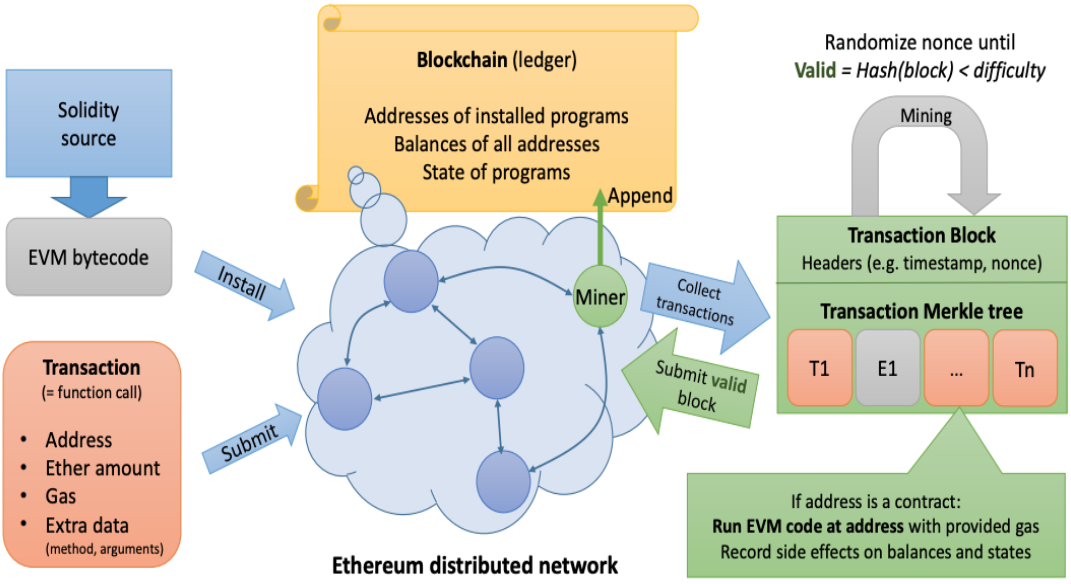


Figure 29: High level workflow overview in the Ethereum network [70]

3.5 Implementation

3.5.1 Architecture

Figure 30 shows the pipelined architecture of our program. The Mesa framework consists of the various agents that interact with other agents and the environment. The networking of agents, visualisation, and settable parameter modules are bundled together and deployed to the browser for the modeller to interact with.
The following sections will briefly describe the steps taken and some of the modules used to implement and integrate two major technologies: ABM and blockchain technology.

3.5.2 Defining Agents and Interactions

First and foremost, we define agents behaviours, its goals and interactions with other agents and the environment. The interaction is carried by interfacing with blockchain based smart contracts on the Ethereum platform which is made possible with Web3.py. Web3.py can read and write to the Ethereum blockchain, and interact with smart contracts. Furthermore, we implemented the event listeners for each agent in Web3.py as well.

3.5.3 Building the network

The UserSettableParameter module allows the modeller to set parameter to be interactive. The Interface section will go into detail of the parameters that the user can interact with to build the network. It is important that the model i.e the grid network, is able to grow and shrink which can be done through interactive parameters.

To build the network dynamically, we used networkx, which is a python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.

After the skeleton of the network is obtained, based on the number of agents that the modeller can dynamically set through the user interface, we dynamically create Ethereum accounts for each agent, then we establish smart contracts between agents

and deploy them to the Ethereum network. After the accounts are created and smart contracts deployed, each agent is dynamically placed into their respective nodes with the NetworkGrid module in order to interact with other agents and the environment.

3.5.4 Scheduling Agents

Time in most ABM moves in ticks or more popularly called steps. At each step of the model, one or more agents, usually all of them are activated and take their own step, changing internally and possibly interacting with other agents and the environment.

Mesa offers different schedulers to control the order in which agents are activated. Based on the scheduler, all the agents may activate in the same order every step; their order might be shuffled every steps or all the agents activate at the same time. Our program made use of SimultaneousActivation scheduler to activate all of the agents simultaneously. The activation regime may seem unimportant, however scheduling patterns can have an impact on the results [43].

3.5.5 Data Analysis

It is important to collect data generated from the model to analyse the behaviour of the model. Since one of the main goals of agent-based modelling is generating data for analysis, we use DataCollector module which is able to store model-level variables and agent-level variables at each step. The DataCollector can export the data it has collected as a pandas DataFrame, for easy interactive analysis.

3.5.6 Visualisation

Building a model and analysing static data output of the model is not sufficient. One of the advantages of ABMs is that we can often watch them run step by step, potentially identifying emergent behaviours and patterns, or bugs or developing new hypothesis, intuitions and insights. Other times, watching a model run can explain it to an unfamiliar audience better than static data output.

The visualisation is done through a browser window, using JavaScript to draw the network with NetworkModule at each step of the model. A small web server is

launched, which runs the model and converts the model into JSON data and passes it to the browser, which then draws it in the browser window.

Another element to the visualisation is the chart, with the help of ChartModule, it will pull data from the model's DataCollector and draw the different model-level variables and agent-level variables at each step using Chart.js Javascript libraries.

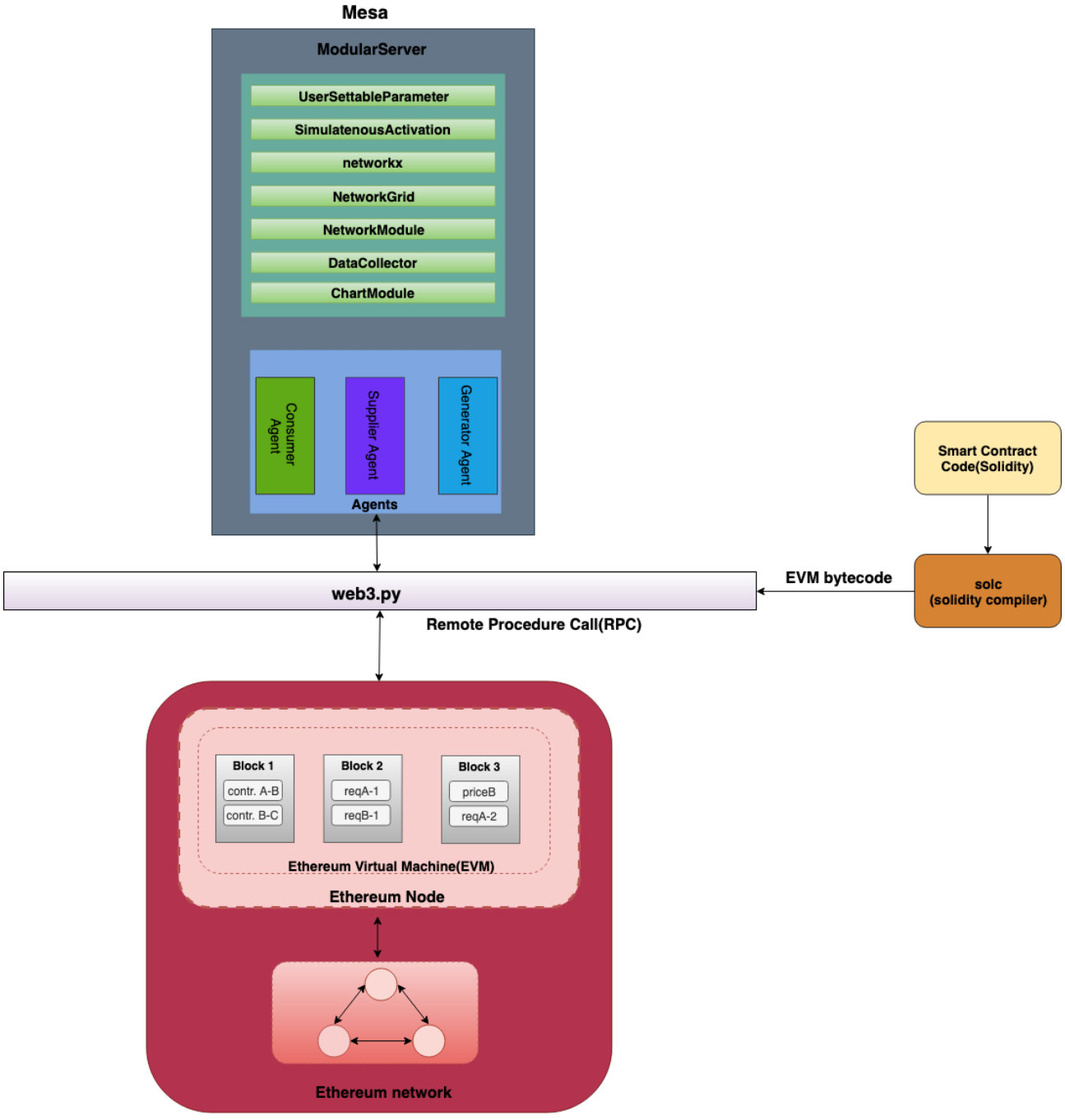


Figure 30: Pipelined architecture of the program

3.6 Interface

The user interface in this program are the interactive settable parameters as shown in Figure 31. The decision to add dynamic parameter manipulation is because one of the reasons we want to be able to observe a model execute is to conduct ad-hoc experiments, for example, we want to get a better understanding of how the model changes with different parameter values. Hard coding interested parameter values are not the ideal approach as one would have to stop the simulation, edit the parameter values in the code and relaunch the model every time.

In this model, the consumers have a one-to-one relationship with consumers within its cluster(e.g. Neighbourhood Area Network(NAN)), and each consumer has a one to one relationship with its supplier determined. The minimum number of consumers per supplier can be set with *Consumer per Supplier.* parameter slider. The supplier has a one-to-many relationship with generators which can be dynamically set with *Number of Generator* parameter slider.

The *Upstream Communication Frequency* parameter in the user interface determines how often consumers will request energy from their energy suppliers, this ties in with the Consumer Activity Diagram(Figure 24), where the probability of interaction between consumer and supplier is set by the probability p. In this case, probability p is the *Upstream Communication Frequency.*

It may be the case the virus can propagate through consumers, infecting IoT devices inside smart homes, transmitting manipulated/compromised data to smart meters. The following are the parameters we choose to model a simple virus:

- *Initial outbreak size*: sets the number of initial infected node in a random cluster.
- *Virus Spread Chance*: the probability of a malicious/infected node infecting its neighbour within the cluster
- *Virus Check Frequency*: infected nodes are not immediately aware that they are infected. Only every so often (determined by the virus check frequency slider) do the nodes check whether they are infected by a virus. This might

correspond to a regularly scheduled virus-scan procedure, or simply someone noticing unusual behaviour of the system.

- *Recovery Chance*: the probability that a node can recover from the virus once it is infected.
- *Gain resistance*: if a node recovers, there is a probability it will become resistant to the virus in the future

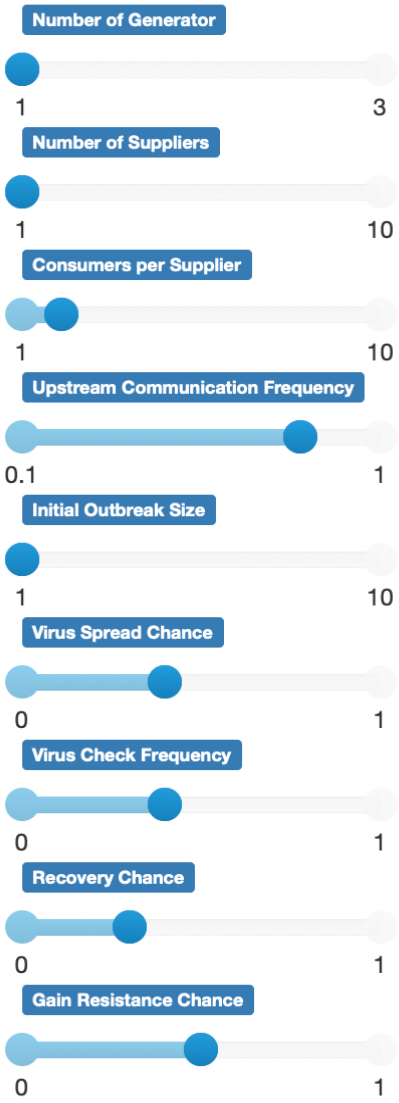


Figure 31: User interface parameters

4 Case Study and Evaluation

We will now present three case studies in order of increasing system complexity and evaluate the results of the interaction between the consumer, supplier and generator. The nodes represents one of the three agents, namely consumer agent, supplier agent and generator agent as shown in Figure 32. The biggest-sized nodes represent the generators, the medium-sized nodes represent the suppliers, and the smallest-sized nodes represent the consumer. In our program, one can hover over the nodes to get more information about the agent.

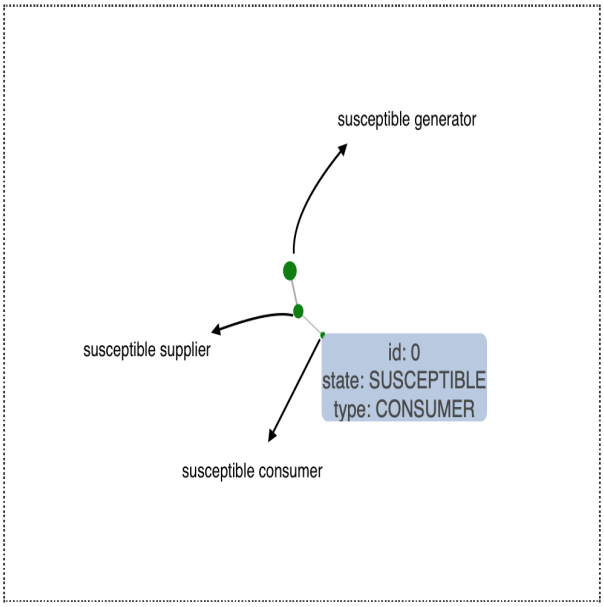


Figure 32: Visual agent classification

The threat model introduced is a simple data tampering of consumer requests before it is broadcasted to the network. In the case study, we assume, a virus can be spread across consumer nodes causing data tampering of consumer requests. We expect for the smart contracts to self-enforce its embedded rules. If the terms of contracts are breached, an event is logged on the blockchain which will be read by the authorised agent and the appropriate actions will be taken.

We will begin with a simple case study and work our way to a higher complexity case study with multiple agent interactions and evaluate the results of supply and generation.

The logic implemented behind our program allows scalability in the sense that based

on the parameters set by the modeller, the network can dynamically adjust to the parameters. Furthermore, Ethereum accounts can be dynamically created, communications can be dynamically established between agents and the respective smart contracts can be deployed to the Ethereum network. However, for comprehensibility, we have kept it to a small number of agent interactions in this case study. We utilise data from data.gov [75] buildings in this simulation.

4.1 Case Study 0

Case study 0 consists of the interaction between one consumer agent, one supplier agent and one generator agent. In this case study, no threats are present; this is to demonstrate the regular interactions between stakeholders in a system with no malicious usage.

Figure 33 shows the start state. All of the stakeholders in the system are susceptible to viruses or other malicious effects. Since outbreak size is set to 0, the virus spread chance, virus check frequency, recovery chance and gain resistance chance are not in operation. The *Upstream Communication Frequency* is set to one and hence whenever a consumer requires energy it will make a request to the supplier for energy through the established smart contract between the consumer and supplier.

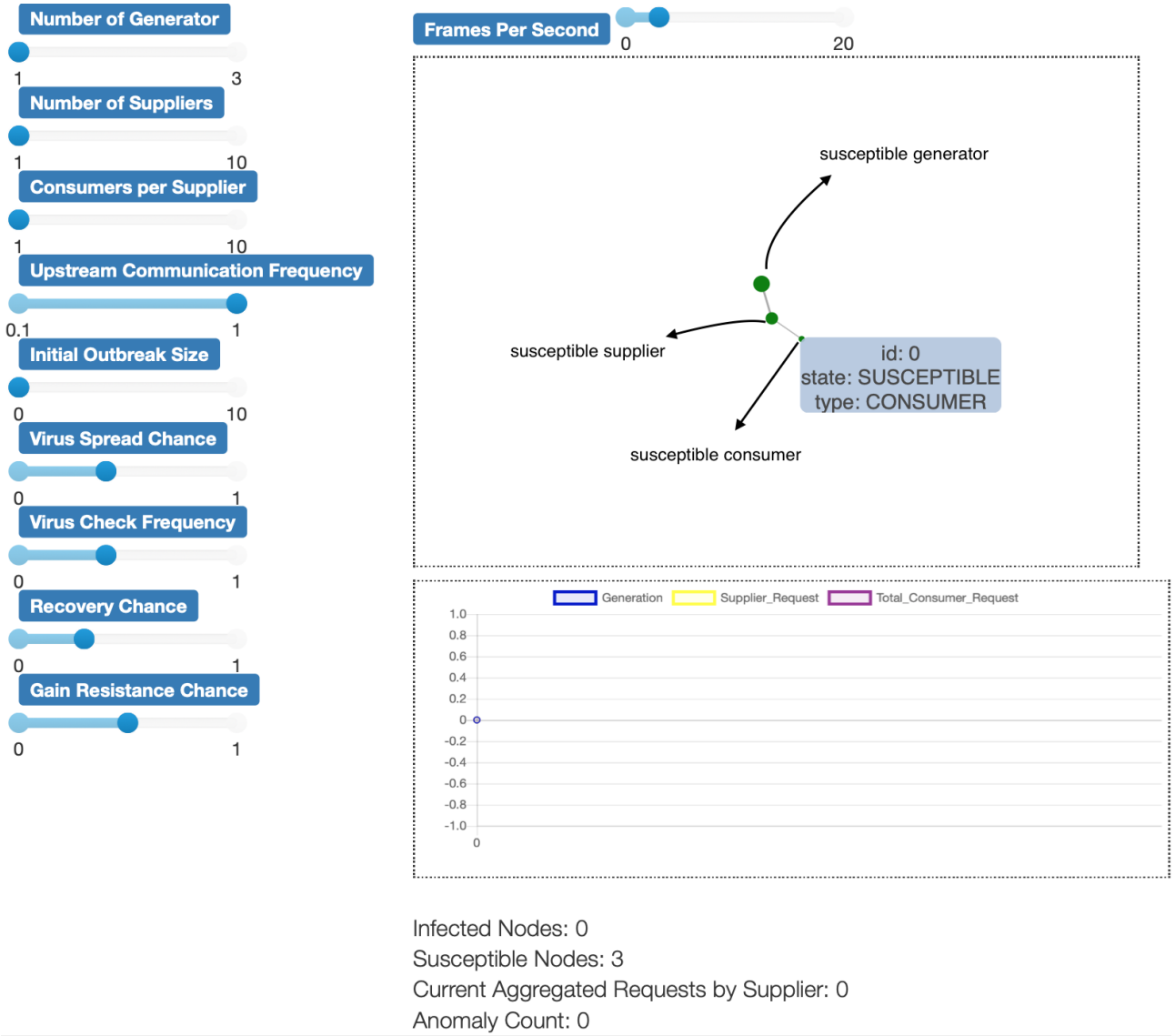
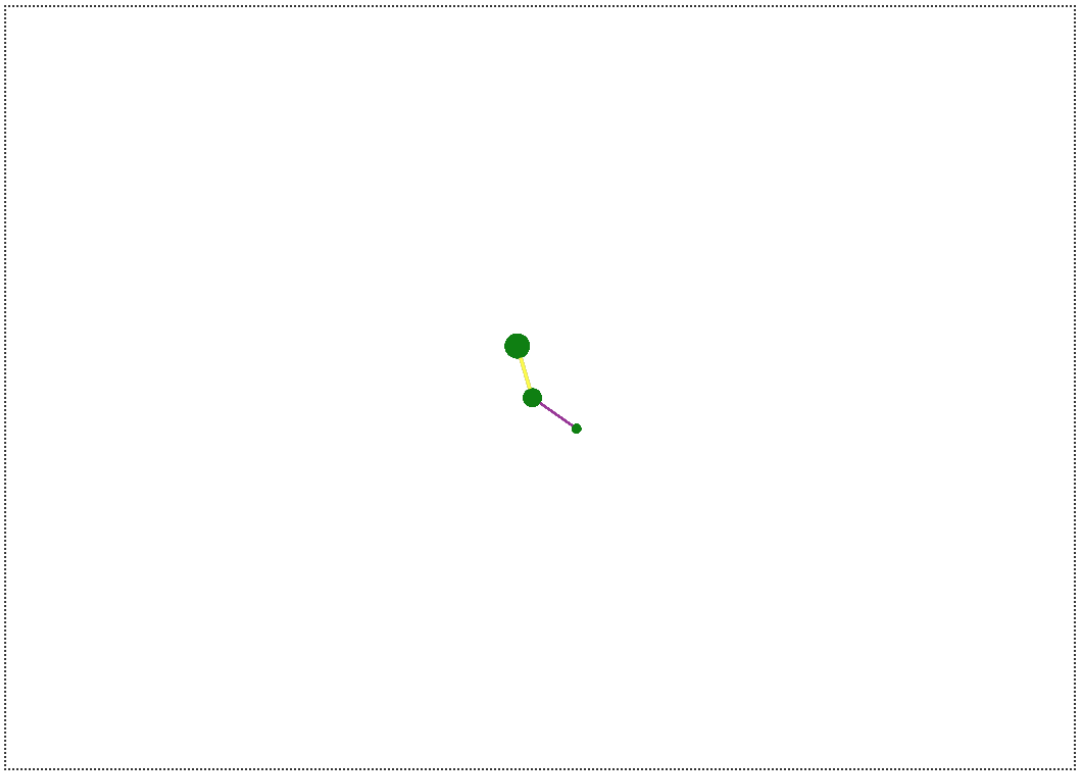
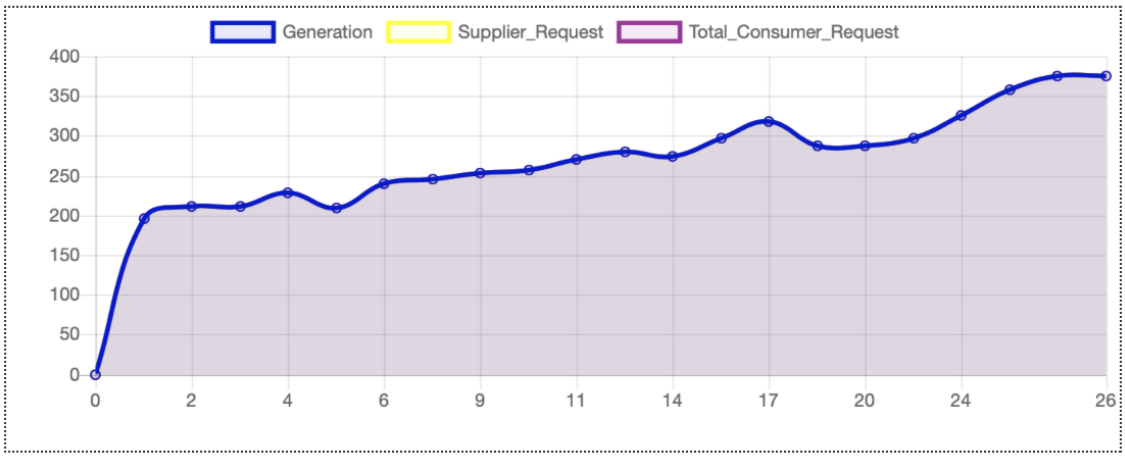


Figure 33: Case Study 0 start state

Figure 34 shows the progress of the interaction between consumer, supplier and generator.





Infected Nodes: 0
Susceptible Nodes: 3
Current Aggregated Requests by Supplier: 376
Anomaly Count: 0

Figure 34: Case Study 0 execution state

Figure 35 shows the result of Case study 0, it shows the energy requested by the consumer and the energy supplied at each step. We can see from the graph that the requests from the consumer at each step match with the energy generation by the generator. This is because no terms of contracts are breached.

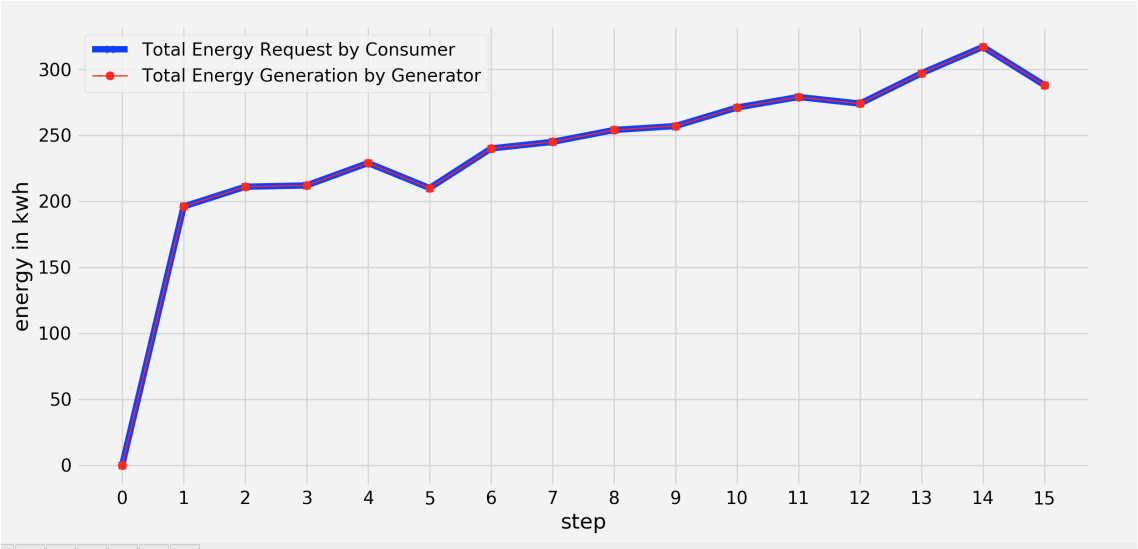


Figure 35: Result of Case Study 0

4.2 Case Study 1

Case Study 1 consists of interaction between seven consumers, one supplier and one generator. This case study will be partitioned into two. First, we will demonstrate the interaction with no data tampering and then demonstrate the effects after data tampering. The data tampering will take place with the simple threat model which will introduce a simple virus which infects end users devices and tampers with the requests.

4.2.1 No outbreak size

As we can see from Figure 36 the *Initial Outbreak Size* parameter is set to 0.

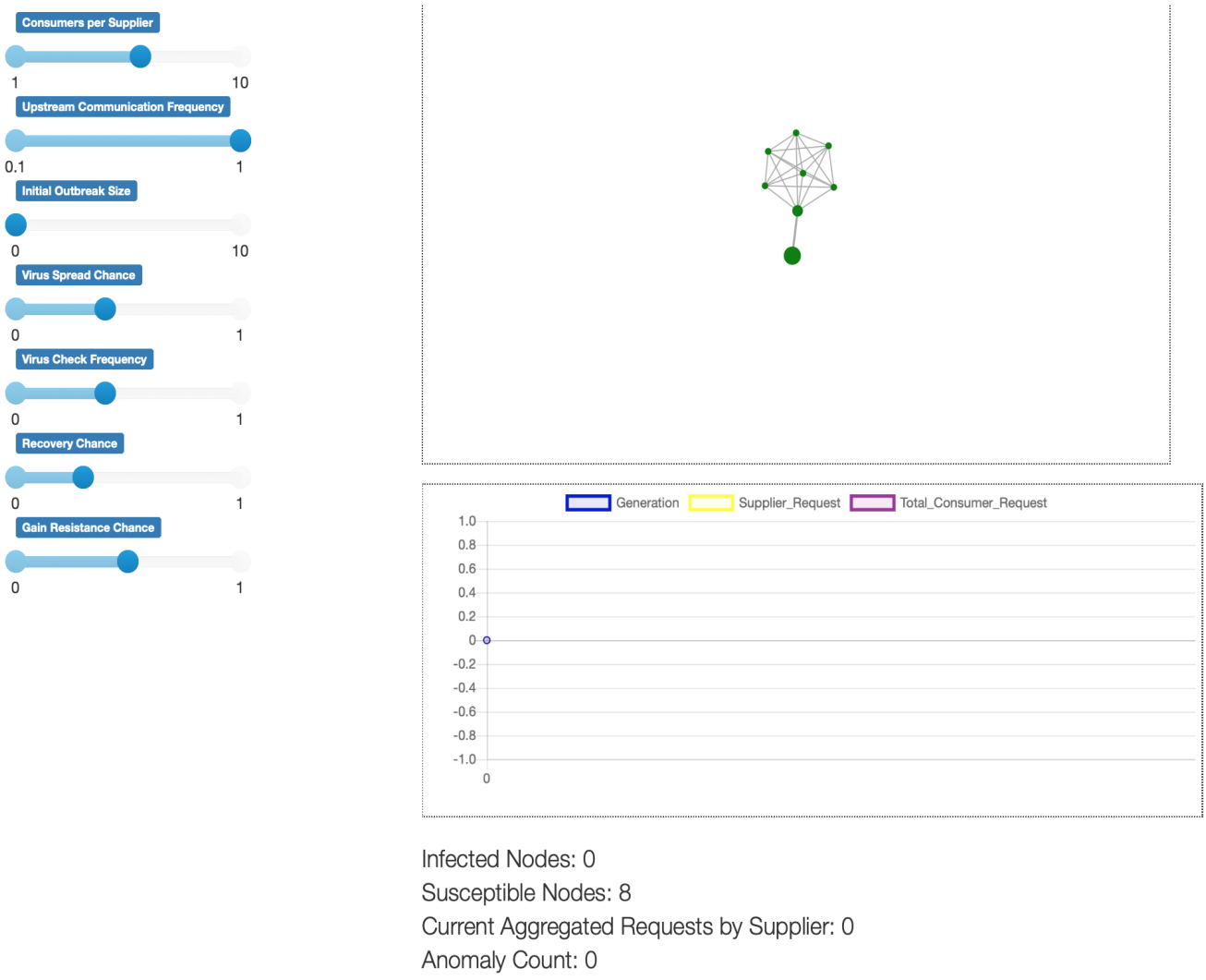
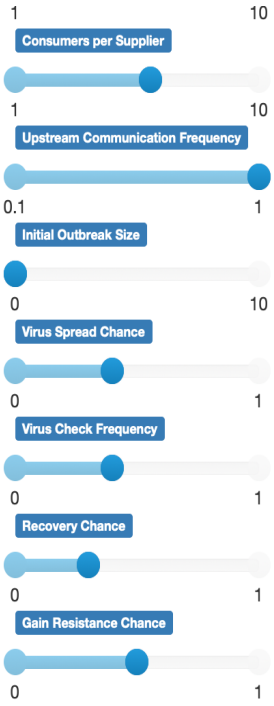
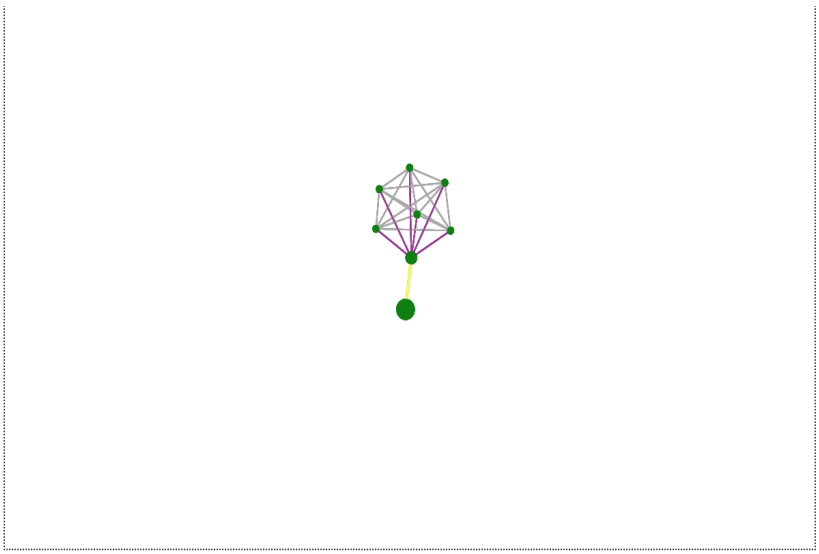
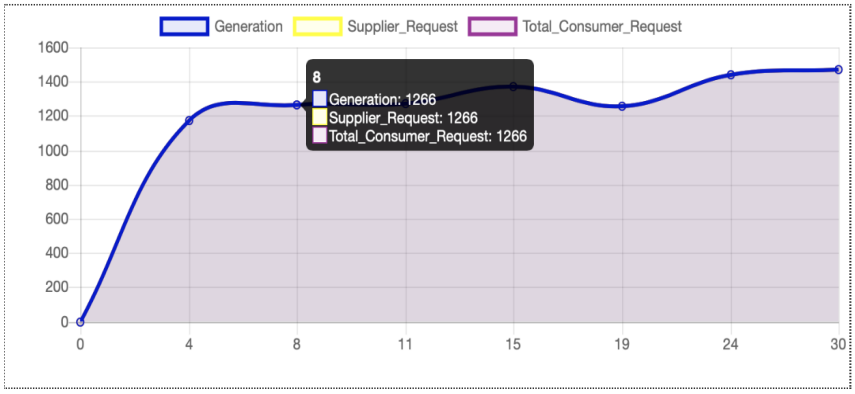


Figure 36: Case study 1 start state - no outbreak

Figure 37 shows the progress of interactions with no outbreak.







Infected Nodes: 0
Susceptible Nodes: 8
Current Aggregated Requests by Supplier: 1470
Anomaly Count: 0

Figure 37: Case study 1 execution state - no outbreak

Figure 38 shows the result of the first section of Case Study 1 which was the simulation with no virus outbreak. We can see that the consumer requests match the total energy generation requested by the supplier, which is similar behaviour observed from the result of case study 0.

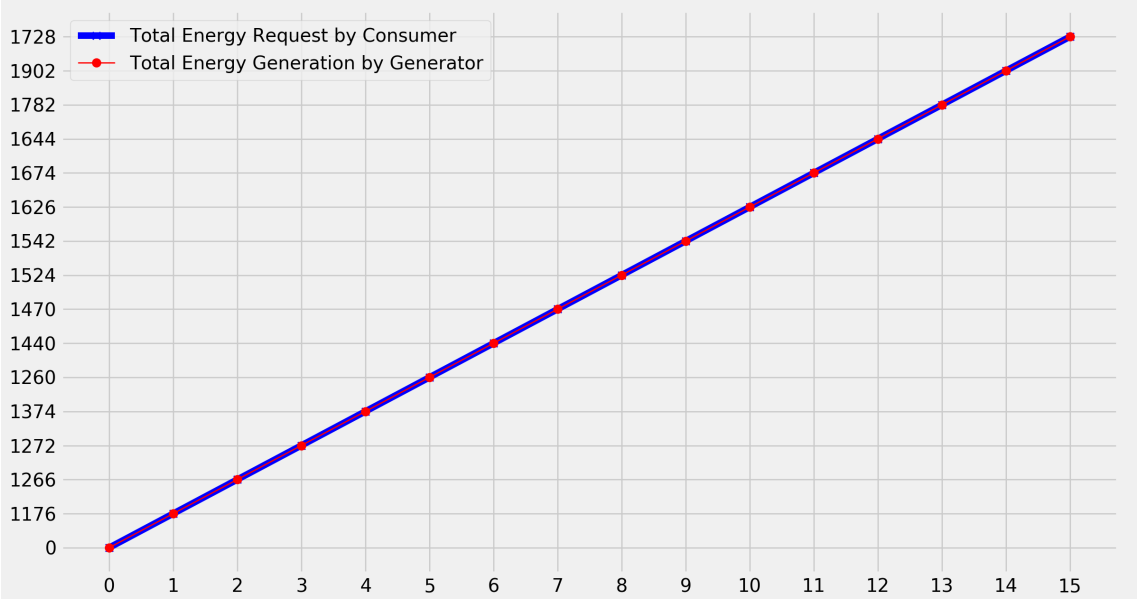


Figure 38: Result of Case Study 1 - with no outbreak

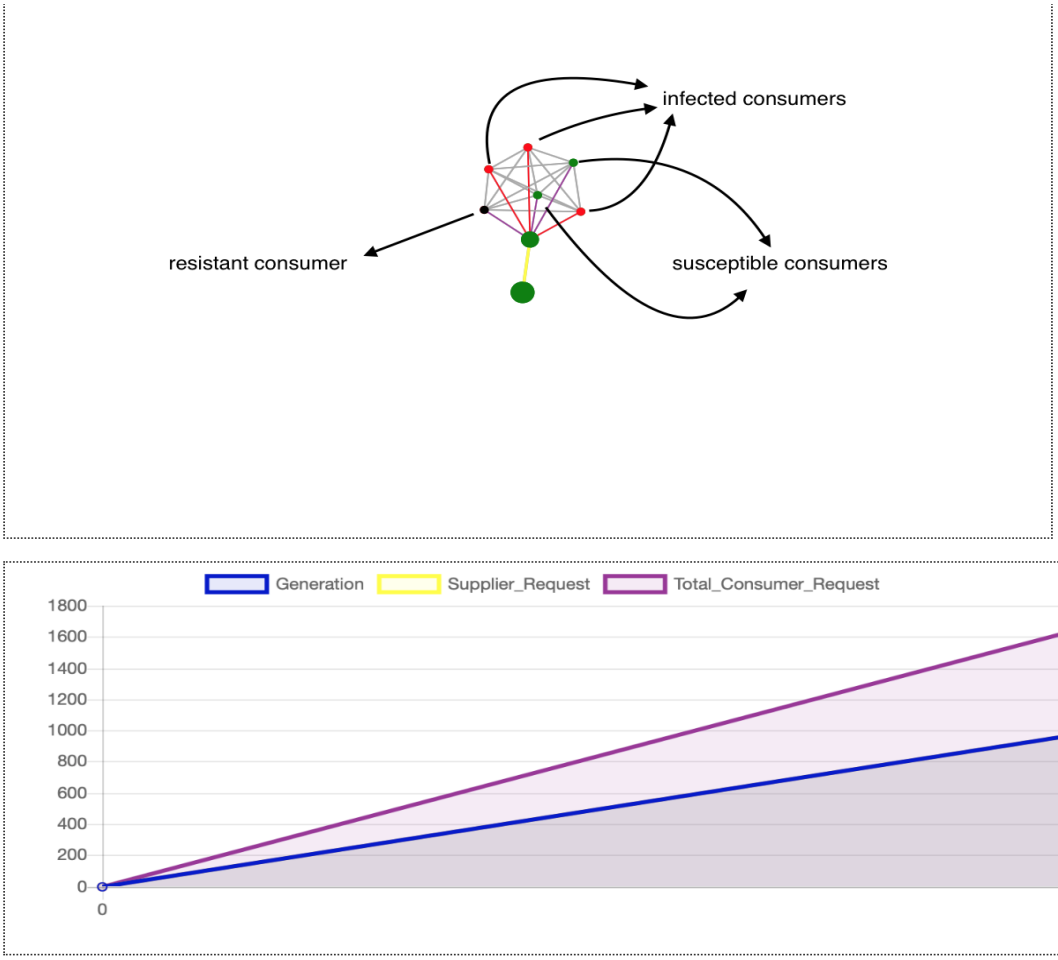
4.2.2 With outbreak size of 1

We will now randomly introduce a virus into the consumer cluster and evaluate the behaviour of the network by setting the *Initial Outbreak Size* to one.

Figure 39 shows the introduction of the virus which randomly infects one of the consumers within one the clusters.

Figure 39: Case Study 1 with outbreak - start state

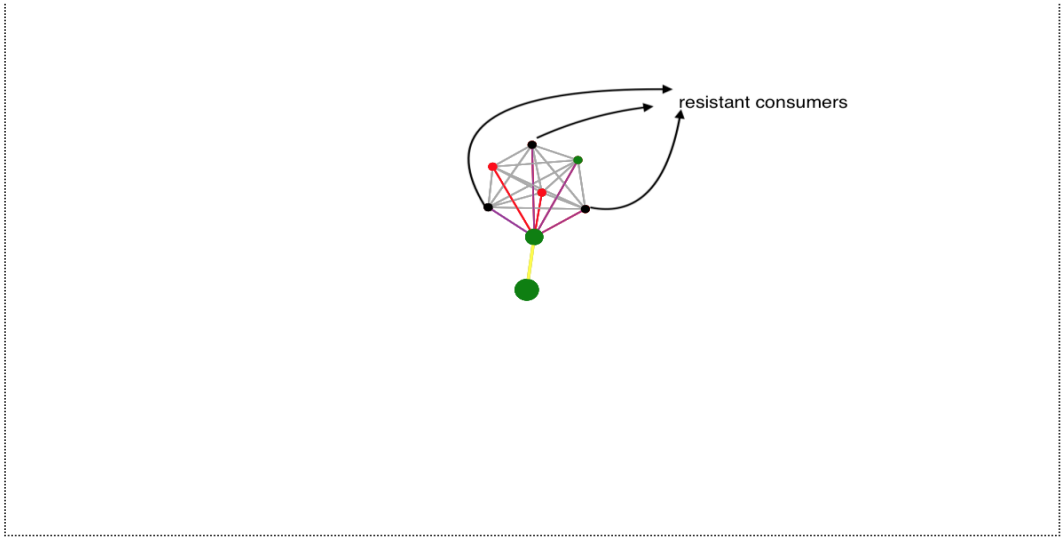
Figure 40 shows the virus spreading across the cluster, one of consumers becoming resistant to the virus whereas other consumers are still susceptible to the virus.

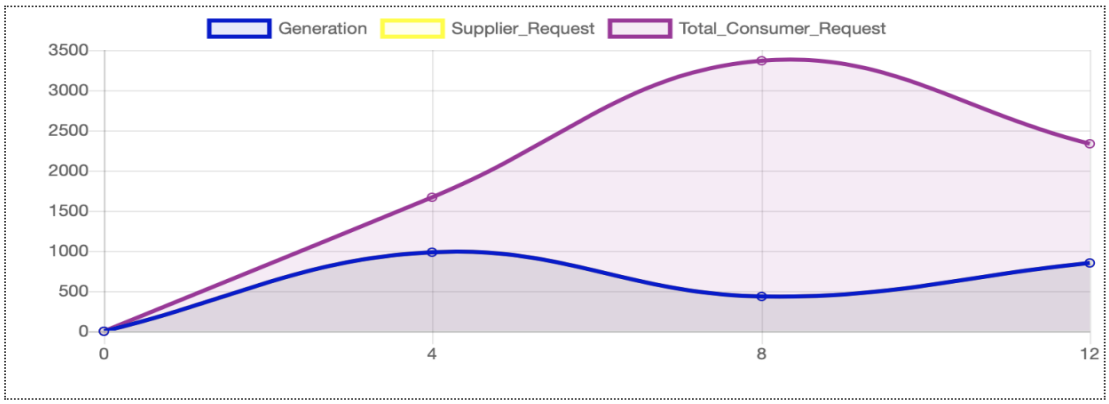


Infected Nodes: 3
Susceptible Nodes: 4
Current Aggregated Requests by Supplier: 980
Anomaly Count: 1

Figure 40: Case Study 1 with outbreak - virus spreading

Figure 41 shows the virus outbreak residing, with more nodes becoming resistant to the virus.

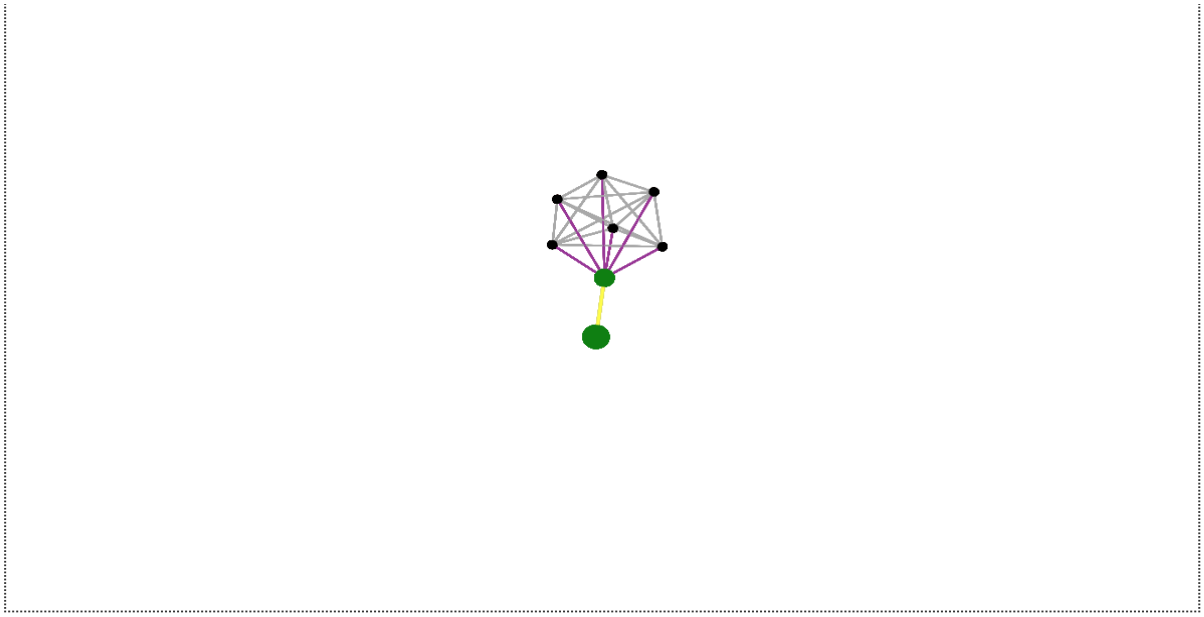


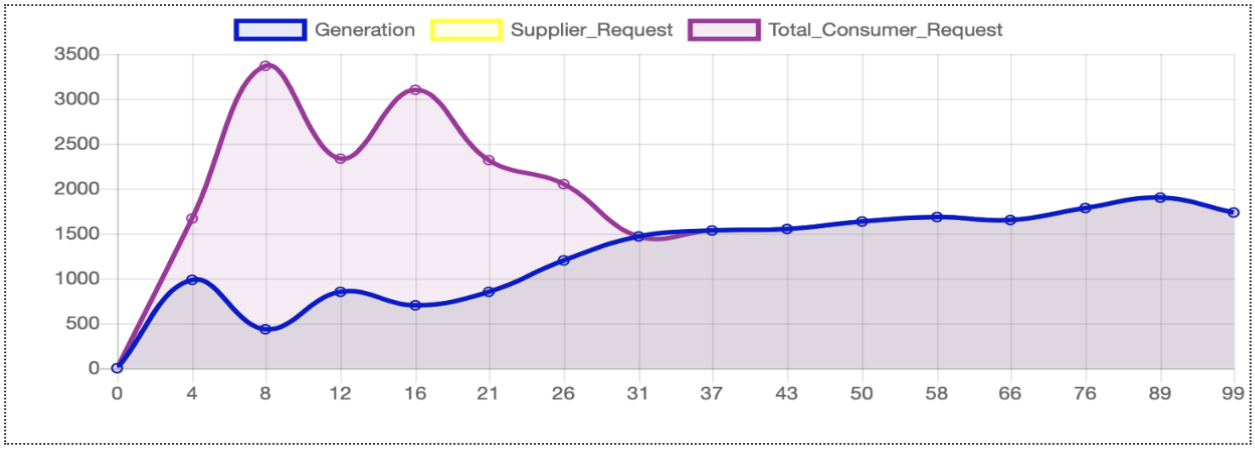


Infected Nodes: 2
Susceptible Nodes: 3
Current Aggregated Requests by Supplier: 848
Anomaly Count: 7

Figure 41: Case Study 1 with outbreak - nodes becoming resistant

Figure 42 shows that the virus outbreak has resided, with all consumer nodes resistant to the virus. From the chart module, we can see that when the nodes became resistant, the generation and total consumer request matched at point 31.





Infected Nodes: 0
Susceptible Nodes: 2
Current Aggregated Requests by Supplier: 1728
Anomaly Count: 13

Figure 42: Case Study 1 with outbreak - nodes resistant

Figure 43 shows the result of the second part of case study 1. The graphs shows total consumer request at each step and the energy supplied by generator. We can see between steps 0 and 7 that the consumer request and energy generation do not match, this is because the supplier will not accept requests that have breached terms of contract and hence will not request energy from the generator for that consumer. As the virus detected and removed in step 7, the consumer request, supply and generation levels are back to normal.

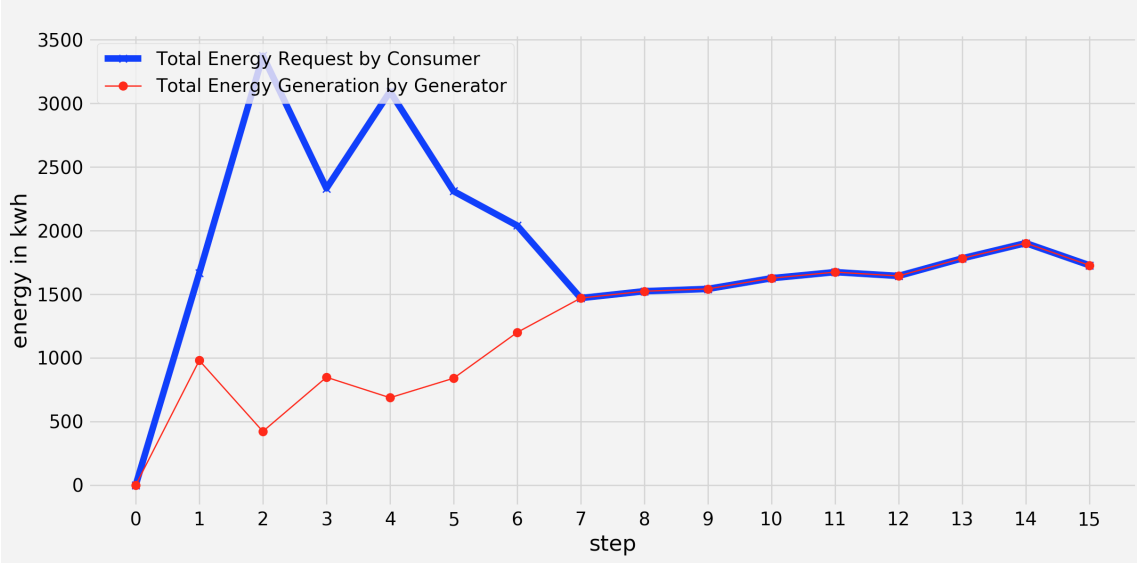


Figure 43: Result of Case Study 1 - with outbreak size of 1

4.3 Case Study 2

This case Study consists of interaction between multiple consumers(minimum four), four suppliers and one generator. Similar to the second part of Case Study 1, we will randomly introduce a virus into one of the clusters by setting the outbreak size to one and now study the behaviour with multiple consumers and multiple suppliers.

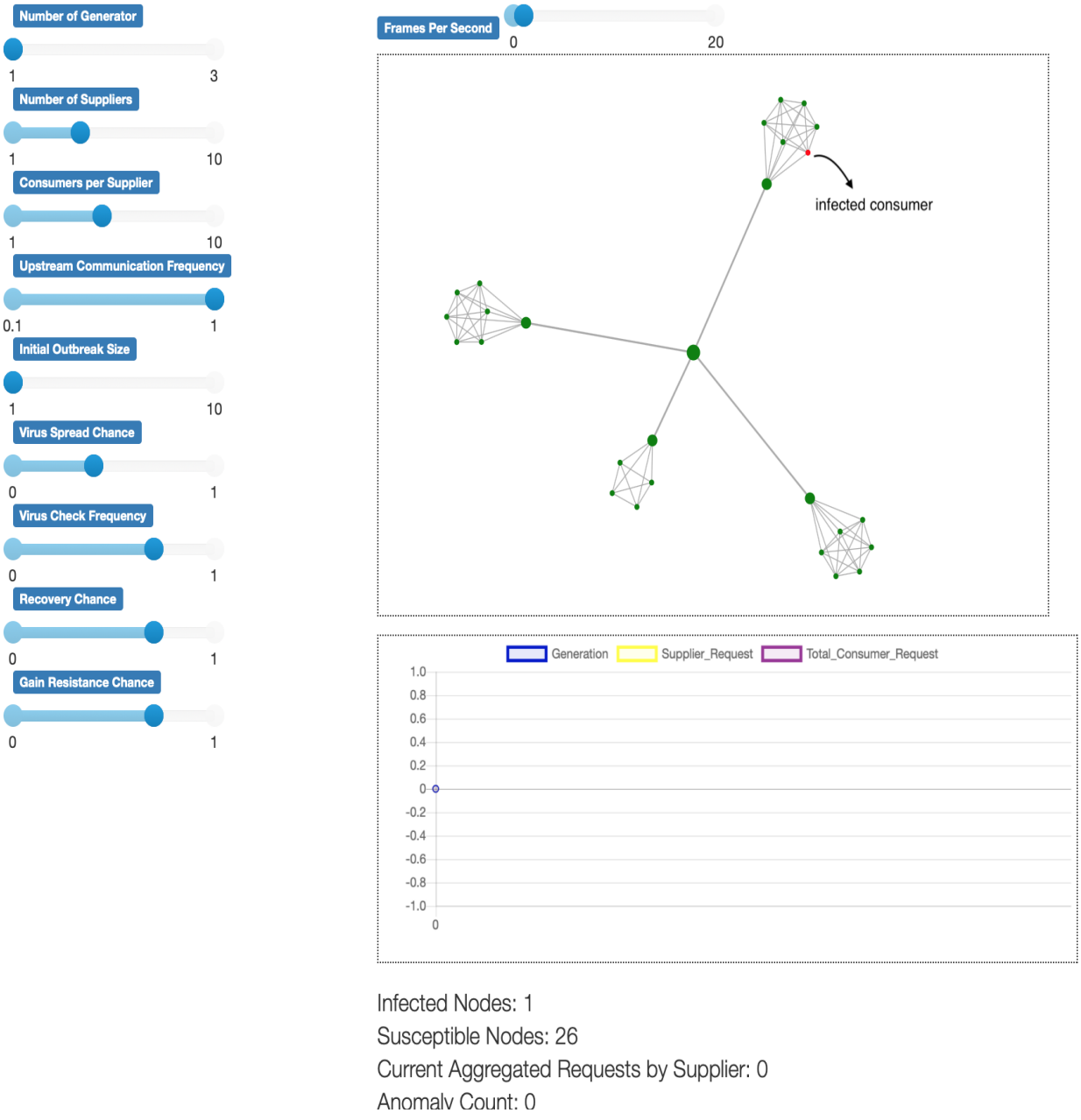


Figure 44: Case Study 2 start state

Figure 45 shows the progress of the virus, the majority of the nodes are infected, and some of the nodes become resistant to the virus and are supplied with the requested energy.

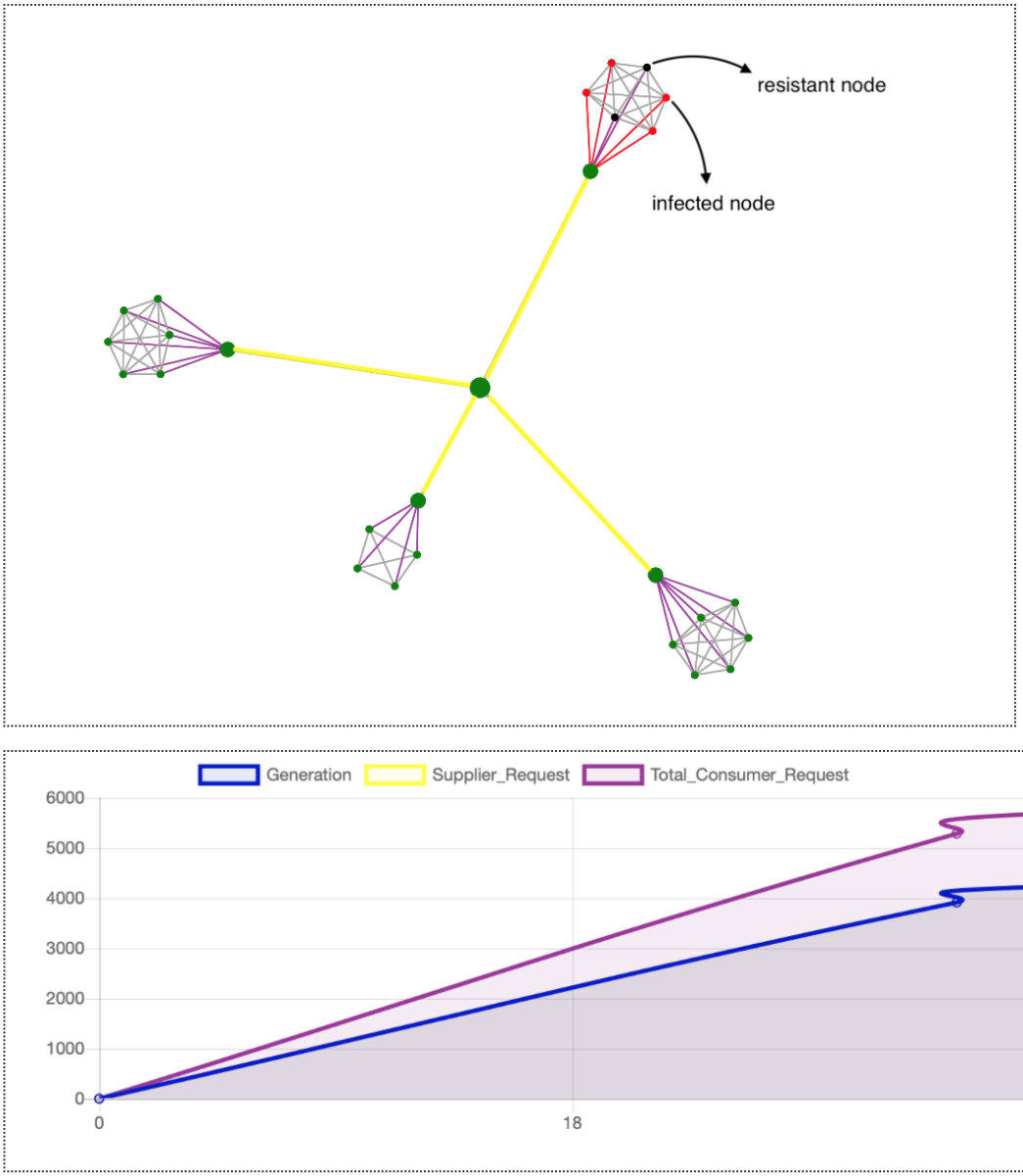


Figure 45: Case Study 2 progress

Figure 46 shows the cluster becoming resistant to the virus at point 801, and the total energy generation, the supplier request and consumer request matching.

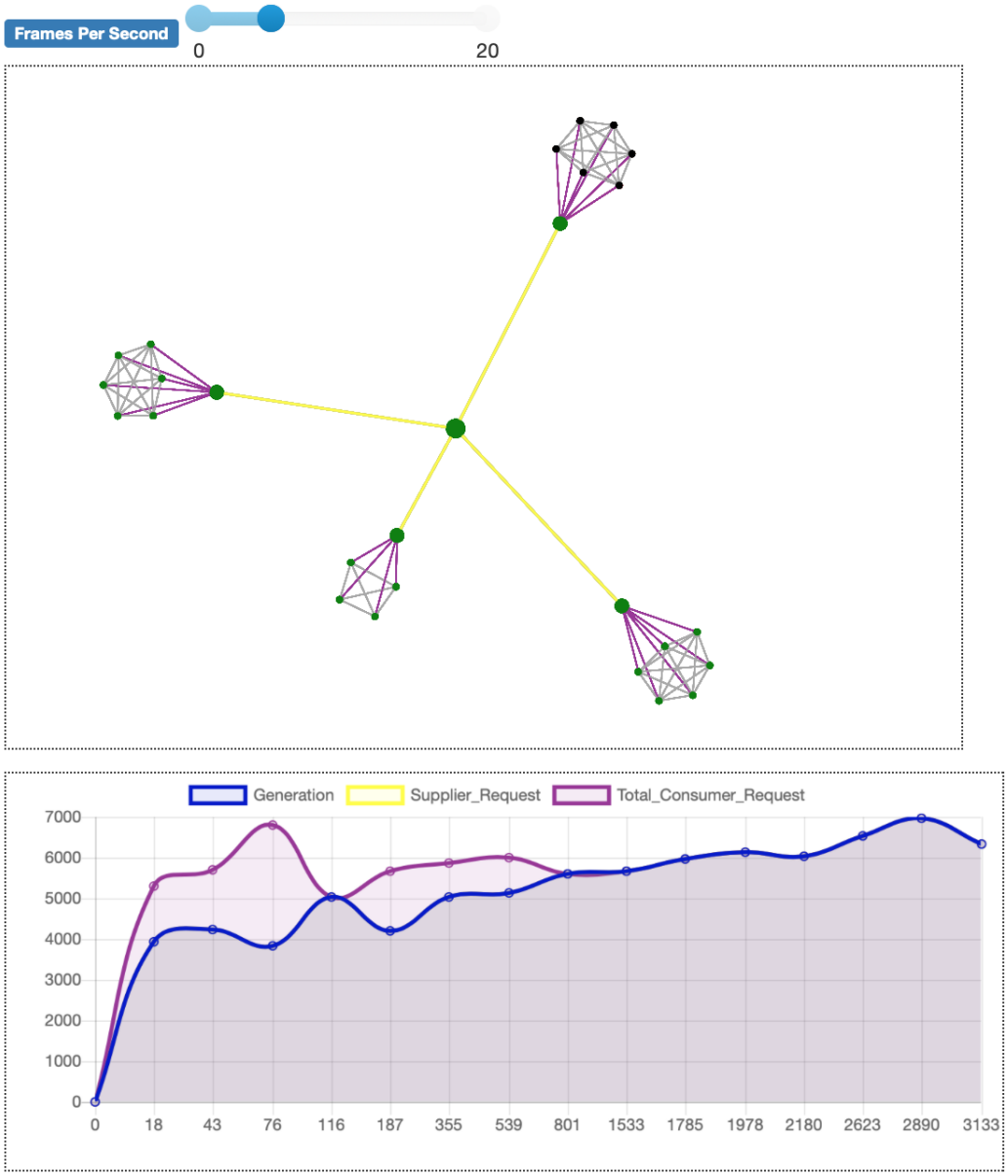


Figure 46: Case Study 2 nodes becoming resistant

Figure 47 shows the result of Case Study 2, showing the energy requests by consumer and the total energy generation at each step.. We can see from Figure 47 that between step 0 and 8, the request and generation do not match for the system as there are contract breaches. From step 8 and onwards, the operation is back to normal, this showcases similar behaviour to the second part of Case Study 1.

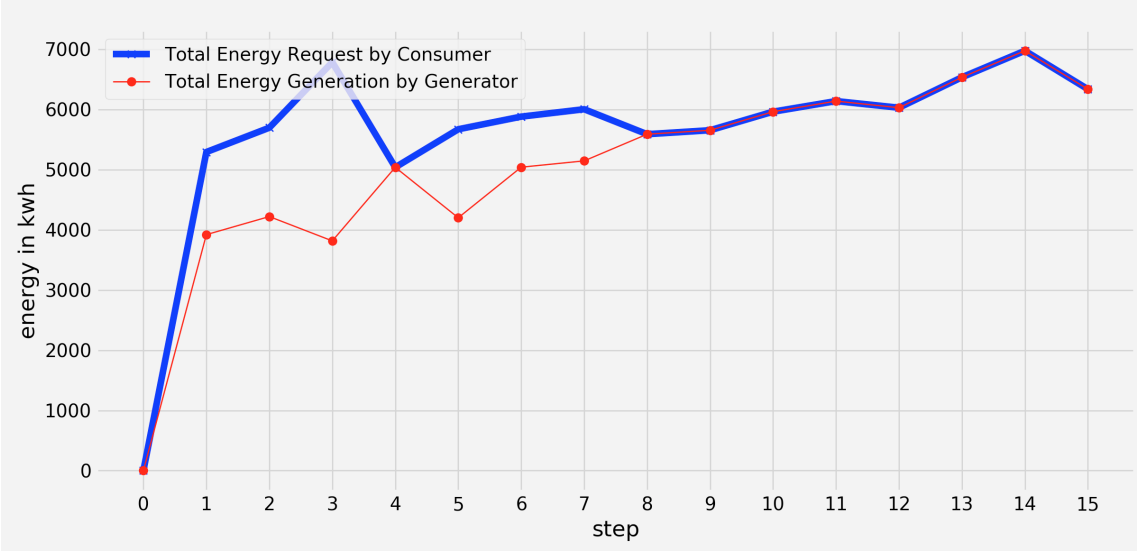


Figure 47: Result of Case Study 2

5 Afterword

5.1 Operational Modelling

ABM does come with its weaknesses, the challenges are as follows. First, the dynamics of ABM in complex systems are usually very complicated that the modellers themselves may not fully understand how the global behaviours emerge. This makes it difficult to distinguish whether observed significant results are legitimate logical implications of the assumptions that the modeller is interested in, or they are due to errors in the design or implementation of the model. Second, ABM ignores the interactions between agents and macro factors. Third, it is very detailed to simulate over extended periods because of the large number of parameters and rules which become difficult to identify and may require extensive analysis to determine the prediction robustness. Nonetheless, ABM is an effective modelling method which combines time and space dimension, emphasising spatial and social interactions between individuals and the environment.

When modelling a particular application domain we should not view ABM and DES as strictly alternatives techniques but as complementary simulation techniques which can model different components in an application domain. ABM prove to be useful in simulating complex autonomous stakeholders in the system whereas DES appears to be more suitable for issues concerned with the operational level, such as distribution and transportation planning, routing and scheduling. With the ABM approach, the agents can also include the processes and activities modelled by DES, and thus the ABM can include all the logical components of a DES model in addition to having other capabilities.
Future work for this project in terms of modelling would be as follows:

- A hybrid simulation approach consisting of DES simulators and ABM simulators. DES could be used to model SCADA control systems in the power grid. One can use a DES-type simulator for example with VSM methodology and ARENA software to simulate the physical world (the objective states of material objects), and agent simulators, which simulate the internal (subjective) states of agents and their behaviours.

- To extend the agents to be autonomous by integrating AI techniques, fuzzy logic and Machine Learning programs which will result in global emergent behaviours as explained in the Background section.

5.2 Cybersecurity

Currently, the development process of writing high performing and secure smart contracts is a difficult task as it involves the application of unconventional programming paradigms. A survey of possible attacks on Ethereum contracts was published by Atzei et al. [86]. By following the best design practices, developers can address the common security problems and mitigate typical attack scenarios [71]. Furthermore, analysis tools such as Oyenete [84] and ZEUS [85] for Ethereum and Hyperledger Fabric platform may help in detecting vulnerability patterns in smart contracts. Some claim the choice of Turing-complete languages is a limiting factor in the verification process and that non-Turing complete, human-readable languages may overcome this issue [86]. Research on the smart contract topic has only begun in industry and academia, and more work is required on this topic.

Blockchain technologies will need to prove they can offer scalability, speed and security in physically entangled networks such as smart grids and smart cities. The ongoing research efforts on distributed consensus mechanisms are crucial for achieving scalability, speed and security. Early adopters of blockchain technologies in the energy sector may face the challenge of selecting the right consensus mechanism and system architecture, without having a clear long-term picture of the advantages and downsides that each approach has to offer.

Whether blockchain technology is viable as a cybersecurity mechanism in complex systems such as the power grid will not only be determined by its technical capabilities but will also depend on the regulatory and legal framework and the economic viability of investments.

Introduction of blockchain technology to the smart grid space may raise privacy concerns. Ensuring blockchain solution complies with legal privacy requirements such as EU's General Data Protection Regulation (GDPR) could become a challenge in a decentralised solution such as blockchain compared to a conventional data

store. To preserve privacy, we would require novel techniques of pseudonymising data on the ledger (i.e. to make the information not traceable to individual users) or permissioned ledgers(which has its limitations [87]) such as Hyperledger Fabric [88] where access to data will be restricted to authorised entities.

Both, research and commercial bodies are currently pursuing blockchain innovation in the energy sector. The work of Andoni et al. [74] shows that most projects in the energy sectors space are in the early development phase, and research is still ongoing on key improvement areas that would allow desired scalability, speed and security. We still need additional research initiatives, projects and collaborations to determine if the technology can reach its full potential and prove its commercial and economic viability and if it can be adopted into the mainstream for securing complex SoS.

References

[1] Wulf, W. A. (2000). Great achievements and grand challenges. The Bridge, 30(3), 4.

[2] Griffor, E. R., Greer, C., Wollman, D. A., Burns, M. J. (2017). Framework for Cyber-Physical Systems: Volume 1, Overview (No. Special Publication (NIST SP)-1500-201).

[3] Maier, M. W. (1998). Architecting principles for systems[U+2010]of[U+2010]systems. Systems Engineering: The Journal of the International Council on Systems Engineering, 1(4), 267-284.

[4] Engell, S. (2014). Cyber-physical systems of systems—definition and core research and innovation areas. Working Paper of the Support Action CPSoS.

[5] Nist.gov. (2018). Update of the NIST Smart Grid Conceptual Model (Discussion DRAFT). [online] Available at: https://www.nist.gov/sites/default/files/documents/2018/09/10/draft_smart_grid_conceptu [Accessed 15 Feb. 2019].

[6] Farhangi, H. (2010). The path of the smart grid. IEEE power and energy magazine, 8(1), 18-28.

[7] Bondavalli, A., Bouchenak, S., Kopetz, H. (Eds.). (2016). Cyber-Physical Systems of Systems: Foundations–A Conceptual Model and Some Derivations: the AMADEOS Legacy (Vol. 10099). Springer.

[8] Ceccarelli, A., Bondavalli, A., Froemel, B., Hoeftberger, O., Kopetz, H. (2016). Basic concepts on systems of systems. In Cyber-Physical Systems of Systems (pp. 1-39). Springer, Cham.

[9] Crutchfield, J. P. (1994). The calculi of emergence: computation, dynamics and induction. Physica D: Nonlinear Phenomena, 75(1-3), 11-54.

[10] Liang, G., Weller, S. R., Zhao, J., Luo, F., Dong, Z. Y. (2017). The 2015 ukraine blackout: Implications for false data injection attacks. IEEE Transactions on Power Systems, 32(4), 3317-3318.

[11] Case, D. U. (2016). Analysis of the cyber attack on the Ukrainian power grid. Electricity Information Sharing and Analysis Center (E-ISAC).

[12] ELECTRICITY GRID MODERNIZATION- GAO(U.S GOVERNMENT ACCOUNTABILITY OFFICE). (2011). Retrieved November 17, 2018, from https://www.gao.gov/new.items/d11117.pdf

[13] Bahrami, S., Wong, V. W., Huang, J. (2018). An online learning algorithm for demand response in smart grid. IEEE Transactions on Smart Grid, 9(5), 4712-4725.

[14] Alguliyev, R., Imamverdiyev, Y., Sukhostat, L. (2018). Cyber-physical systems and their security issues. Computers in Industry, 100, 212-223.

[15] Bouarfa, S., Blom, H. A., Curran, R., Everdij, M. H. (2013). Agent-based modeling and simulation of emergent behavior in air transportation. Complex Adaptive Systems Modeling, 1(1), 15.

[16] Technology Roadmap - Smart Grids. (2011). Retrieved February 10, 2019, from https://www.iea.org/publications/freepublications/publication/ smartgrids_roadmap.pdf

[17] Yu, X., Xue, Y. (2016). Smart grids: A cyber–physical systems perspective. Proceedings of the IEEE, 104(5), 1058-1070.

[18] Törngren, M., Grogan, P. (2018). How to Deal with the Complexity of Future Cyber-Physical Systems?. Designs, 2(4), 40.

[19] Mwasilu, F., Justo, J. J., Kim, E. K., Do, T. D., Jung, J. W. (2014). Electric vehicles and smart grid interaction: A review on vehicle to grid and renewable energy sources integration. Renewable and sustainable energy reviews, 34, 501-516.

[20] Islam, M. A., Hasanuzzaman, M., Rahim, N. A., Nahar, A., Hosenuzzaman, M. (2014). Global renewable energy-based electricity generation and smart grid system for energy security. The Scientific World Journal, 2014.

[21] Ashibani, Y., Mahmoud, Q. H. (2017). Cyber physical systems security: Analysis, challenges and solutions. Computers Security, 68, 81-97.

[22] Sun, C. C., Hahn, A., Liu, C. C. (2018). Cyber security of a power grid: State-of-the-art. International Journal of Electrical Power Energy Systems, 99, 45-56.

[23] Mahmoud, M. S., Hamdan, M. M., Barudi, U. A. (2019). Modeling and control of Cyber-Physical Systems subject to cyber attacks: A Survey of recent advances and challenges. Neurocomputing.

[24] Tao, H. Y. S., Bahabry, A., Cloutier, R. (2015). Customer centricity in the smart grid model. Procedia Computer Science, 44, 115-124.

[25] Liang, G., Weller, S. R., Zhao, J., Luo, F., Dong, Z. Y. (2017). The 2015 ukraine blackout: Implications for false data injection attacks. IEEE Transactions on Power Systems, 32(4), 3317-3318.

[26] Liang, G., Weller, S. R., Luo, F., Zhao, J., Dong, Z. Y. (2018). Distributed blockchain-based data protection framework for modern power systems against cyber attacks. IEEE Transactions on Smart Grid.

[27] Shi, H., Xu, M., Li, R. (2018). Deep learning for household load forecasting—A novel pooling deep RNN. IEEE Transactions on Smart Grid, 9(5), 5271-5280.

[28] Xie, L., Mo, Y., Sinopoli, B. (2011). Integrity data attacks in power market operations. IEEE Transactions on Smart Grid, 2(4), 659-666.

[29] Yang, Q., Yang, J., Yu, W., An, D., Zhang, N., Zhao, W. (2014). On false data-injection attacks against power system state estimation: Modeling and countermeasures. IEEE Transactions on Parallel and Distributed Systems, 25(3), 717-729.

[30] Chen, P. Y., Yang, S., McCann, J. A., Lin, J., Yang, X. (2015). Detection of false data injection attacks in smart-grid systems. IEEE Communications Magazine, 53(2), 206-213.

[31] Jiongcong, C. H. E. N., Liang, G., Zexiang, C. A. I., Chunchao, H. U., Yan, X. U., Fengji, L. U. O., Junhua, Z. H. A. O. (2016). Impact analysis of false data injection attacks on power system static security assessment. Journal of Modern Power Systems and Clean Energy, 4(3), 496-505.

[32] Tan, S., Song, W. Z., Stewart, M., Yang, J., Tong, L. (2018). Online data integrity attacks against real-time electrical market in smart grid. IEEE Transactions on Smart Grid, 9(1), 313-322.

[33] Lee, A. (2013). Electric sector failure scenarios and impact analyses. National Electric Sector Cybersecurity Organization Resource (NESCOR) Technical Working Group, 1.

[34] Primadianto, A., Lu, C. N. (2017). A review on distribution system state estimation. IEEE Transactions on Power Systems, 32(5), 3875-3883.

[35] Guo, Y., Ten, C. W., Hu, S., Weaver, W. W. (2016). Preventive maintenance for advanced metering infrastructure against malware propagation. IEEE Transactions on Smart Grid, 7(3), 1314-1328.

[36] Ye, F., Qian, Y., Hu, R. Q. (2015, December). An identity-based security scheme for a big data driven cloud computing framework in smart grid. In 2015 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.

[37] Rusitschka, S., Eger, K., Gerdes, C. (2010, October). Smart grid data cloud: A model for utilizing cloud computing in the smart grid domain. In 2010 First IEEE International Conference on Smart Grid Communications (pp. 483-488). IEEE.

[38] Mylrea, M., Gourisetti, S. N. G. (2017, September). Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale and security. In 2017 Resilience Week (RWS) (pp. 18-23). IEEE.

[39] Gao, J., Asamoah, K. O., Sifah, E. B., Smahi, A., Xia, Q., Xia, H., ... Dong, G. (2018). Gridmonitoring: Secured sovereign blockchain based monitoring on smart grid. IEEE Access, 6, 9917-9925.

[40] Liang, G., Weller, S. R., Luo, F., Zhao, J., Dong, Z. Y. (2018). Distributed blockchain-based data protection framework for modern power systems against cyber attacks. IEEE Transactions on Smart Grid.

[41] Sridhar, S., Ashok, A., Mylrea, M., Pal, S., Rice, M., Gourisetti, S. N. G. (2017, September). A testbed environment for buildings-to-grid cyber resilience research and development. In 2017 Resilience Week (RWS) (pp. 12-17). IEEE.

[42] Pop, C., Cioara, T., Antal, M., Anghel, I., Salomie, I., Bertoncini, M. (2018). Blockchain based decentralized management of demand response programs in smart energy grids. Sensors, 18(1), 162.

[43] Comer, K. W. (2014). Who Goes First? An Examination of the Impact of Activation on Outcome Behavior in Agent-based Models (Doctoral dissertation).

[44] Borshchev, A., Filippov, A. (2004, July). From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools. In Proceedings of the 22nd international conference of the system dynamics society (Vol. 22). Oxfort.

[45] Haitao, L., Xiaomin, C. (2012). Multi-Agent Technology Applied to Mobile Communication. In Green Communications and Networks (pp. 1591-1596). Springer, Dordrecht.

[46] Bradshaw J (1997). Software Agents. MIT Press: Cambridge, MA, (cited in Odell J (2002). Objects and agents compared. J Object Technol1(1): 41–53.

[47] Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems(Cited in P O Siebers(2010)). Proceedings of the national academy of sciences, 99(suppl 3), 7280-7287.

[48] Macal, C. M., North, M. J. (2010). Tutorial on agent-based modelling and simulation. Journal of Simulation, 4(3), 151-162.

[49] Macal, C. M. (2016). Everything you need to know about agent-based modelling and simulation. Journal of Simulation, 10(2), 144-156.

[50] Siebers, P. O., Macal, C. M., Garnett, J., Buxton, D., Pidd, M. (2010). Discrete-event simulation is dead, long live agent-based simulation!. Journal of Simulation, 4(3), 204-210.

[51] Abar, S., Theodoropoulos, G. K., Lemarinier, P., O'Hare, G. M. (2017). Agent based modelling and simulation tools: a review of the state-of-art software. Computer Science Review, 24, 13-33.

[52] Buterin, V. (2014). A next-generation smart contract and decentralized application platform. white paper.

[53] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[54] Swanson, T. (2014). Great Chain of Numbers. A Guide to Smart Contracts, Smart Property, and Trustless Asset Management, publisher= Creative Commons-Attribution 4.0 International.

[55] Davidson, S., De Filippi, P., Potts, J. (2018). Blockchains and the economic institutions of capitalism. Journal of Institutional Economics, 14(4), 639-658.

[56] Dannen, C. (2017). Introducing Ethereum and Solidity. Berkeley: Apress.

[57] Yaga, D., Mell, P., Roby, N., Scarfone, K. (2018). NISTIR 8202 Blockchain Technology Overview. Retrieved from National Institute of Standards and Technology, US Department of Commerce.

[58] Wüst, K., Gervais, A. (2018, June). Do you need a Blockchain?. In 2018 Crypto Valley Conference on Blockchain Technology (CVCBT) (pp. 45-54). IEEE.

[59] Casino, F., Dasaklis, T. K., Patsakis, C. (2018). A systematic literature review of blockchain-based applications: current status, classification and open issues. Telematics and Informatics.

[60] Bogner, A., Chanson, M., Meeuw, A. (2016, November). A decentralised sharing app running a smart contract on the ethereum blockchain. In Proceedings of the 6th International Conference on the Internet of Things (pp. 177-178). ACM.

[61] Delmolino, K., Arnett, M., Kosba, A., Miller, A., Shi, E. (2016, February). Step by step towards creating a safe smart contract: Lessons and insights from a cryptocurrency lab. In International Conference on Financial Cryptography and Data Security (pp. 79-94). Springer, Berlin, Heidelberg.

[62] Zheng, Z., Xie, S., Dai, H. N., Chen, X., Wang, H. (2018). Blockchain challenges and opportunities: a survey. International Journal of Web and Grid Services, 14(4), 352-375.

[63] Ferrag, M. A., Derdour, M., Mukherjee, M., Derhab, A., Maglaras, L., Janicke, H. (2018). Blockchain technologies for the internet of things: Research issues and challenges. IEEE Internet of Things Journal.

[64] Nofer, M., Gomber, P., Hinz, O., Schiereck, D. (2017). Blockchain. Business Information Systems Engineering, 59(3), 183-187.

[65] Szabo, N. (1994). Smart Contracts. Retrieved March 20, 2019, from http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

[66] Szabo, N. (1997). Formalizing and securing relationships on public networks. First Monday, 2(9).

[67] Christidis, K., Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. Ieee Access, 4, 2292-2303.

[68] Solidity - v0.5.3. (2019). Retrieved March 21, 2019, from https://solidity.readthedocs.io/en/v0.5.3/

[69] Merriam, P., Carver, J. (2018). Web3.py. Retrieved March 21, 2019, from https://web3py.readthedocs.io/en/stable/

[70] Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Gollamudi, A., Gonthier, G., Kobeissi, N., ... Zanella-Béguelin, S. (2016, October). Formal verification of smart contracts: Short paper. In Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security (pp. 91-96). ACM.

[71] Wohrer, M., Zdun, U. (2018, March). Smart contracts: Security patterns in the ethereum ecosystem and solidity. In 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE) (pp. 2-8). IEEE.

[72] Bashir, I. (2017). Mastering blockchain. Packt Publishing Ltd.

[73] Orcutt, M. (2017, October 16). How Blockchain Could Give Us a Smarter Energy Grid (Energy experts believe that blockchain technology can solve a maze of red tape and data management problems). Retrieved March 25, 2019, from https://www.technologyreview.com/s/609077/how-blockchain-could-give-us-a-smarter-energy-grid/

[74] Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., ... Peacock, A. (2019). Blockchain technology in the energy sector: A systematic review of challenges and opportunities. Renewable and Sustainable Energy Reviews, 100, 143-174.

[75] The National Archives - Energy Consumption. (2019). Retrieved March 27, 2019, from https://data.gov.uk/dataset/da9a88d6-6535-4c7f-8d54-a93a50b2f177/the-national-archives-energy-consumption

[76] Maidstone, R. (2012). Discrete event simulation, system dynamics and agent based simulation: Discussion and comparison. System, 1(6), 1-6.

[77] Ding, Z., Gong, W., Li, S., Wu, Z. (2018). System dynamics versus agent-based modeling: A review of complexity simulation in construction waste management. Sustainability, 10(7), 2484.

[78] Shalizi, C. R. (2006). Methods and techniques of complex systems science: An overview. In Complex systems science in biomedicine (pp. 33-114). Springer, Boston, MA.

[79] Nikolic, I. (2009). Co-evolutionary method for modelling large scale socio-technical systems evolution.

[80] Ma, T., Nakamori, Y. (2009). Modeling technological change in energy systems–from optimization to agent-based modeling. Energy, 34(7), 873-879.

[81] Cullen, A., Ferraro, P., King, C., Shorten, R. (2019). Distributed Ledger Technology for IoT: Parasite Chain Attacks. arXiv preprint arXiv:1904.00996.

[82] Shackelford, S. (2009). Estonia two-and-a-half years later: a progress report on combating cyber attacks. Journal of Internet Law, Forthcoming.

[83] Czosseck, C., Ottis, R., Talihärm, A. M. (2011). Estonia after the 2007 cyber attacks: Legal, strategic and organisational changes in cyber security. International Journal of Cyber Warfare and Terrorism (IJCWT), 1(1), 24-34.

[84] Luu, L., Chu, D. H., Olickel, H., Saxena, P., Hobor, A. (2016, October). Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 254-269). ACM.

[85] Kalra, S., Goel, S., Dhawan, M., Sharma, S. (2018, February). Zeus: Analyzing safety of smart contracts. In 25th Annual Network and Distributed System Security Symposium, NDSS (pp. 18-21).

[86] Atzei, N., Bartoletti, M., Cimoli, T. (2017, April). A survey of attacks on ethereum smart contracts (sok). In International Conference on Principles of Security and Trust (pp. 164-186). Springer, Berlin, Heidelberg.

[87] Vukolić, M. (2017, April). Rethinking permissioned blockchains. In Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts (pp. 3-7). ACM.

[88] Cachin, C. (2016, July). Architecture of the hyperledger blockchain fabric. In Workshop on distributed cryptocurrencies and consensus ledgers (Vol. 310).