FLAME: Trusted Fire Brigade Service and Insurance Claim System using Blockchain for Enterprises

Journal:	*IEEE Transactions on Industrial Informatics*
Manuscript ID	TII-22-0996
Manuscript Type:	SS on Augmented Intelligence of Things for Smart Enterprise Systems
Keywords:	blockchain, Cyber-Physical Systems, Internet of Things, Smart Contracts, Network Slicing, Insurance, Fire brigade Service

FLAME: Trusted Fire Brigade Service and Insurance Claim System using Blockchain for Enterprises

***Abstract*—Recently, Cyber-Physical Systems (CPS) and blockchain are collaboratively used to develop emerging industrial applications. In which, CPS enables network connectivity to monitor the physical world to take necessary actions whereas, blockchain incorporates robustness, immutability, trust, and transparency to protect against data forgery. In this context, we propose a novel Trusted Fire Brigade Service and Insurance Claim (FLAME) system using blockchain for enterprises to immediately process the fire brigade service request and fire insurance claim for the enterprise owner. A narrowband Internet of Things based network slicing is designed to transmit real-time information to the monitoring station to protect from serious fire damage. Further, a service queue length approach is utilized to select an appropriate fire department. A prototype of the FLAME system is developed on the Hyperledger Besu blockchain by implementing smart contracts using Solidity language. For result validation, we compared the performance matrices with three existing consensus algorithms.**

***Index Terms*— Blockchain, Smart Contract, Network Slicing, Internet of Things, Cyber-physical System, Insurance, Fire brigade**

I. Introduction

At present, researchers and industries from various fields are paying close attention to Cyber-Physical Systems (CPS) due to their demand in many emerging applications such as smart healthcare systems, smart transportation systems, smart grid systems, smart fire detection systems, and smart agriculture systems. The term CPS was initially coined by Helen Gill at the US National Science Foundation in 2006 [1]. The CPS can be described as an input-output system that connects the cyber and physical world by integrating sensing, networking, and computing in a well-defined context to serve a specific purpose. The CPS consists of various elements such as sensors and actuators that are connected with a centralized system for the decision-making and execution of a determined task. In recent years, the CPS connects with the multi-disciplinary field including Wireless Sensor Networks (WSN), and the Internet of Things (IoT) for redesigning the existing centralized system for enterprises or industries with more intelligence [2].

Among all the emerging applications, smart fire detection systems hold their importance to provide early warnings in the event of a fire to protect against a serious hazard such as property or business loss. The smart fire detection system typically consists of a smoke detector, heat detector, and carbon monoxide detector which transmit data to the centralized system using WSN communication technology to remotely monitor the received information. Also, it manages a manual call point or fire alarm to warn the people present in the building, factory, or home. The smart fire detection system is generally equipped with sprinkles that automatically turn on in the presence of fire to cure against small-scale fire but for large-scale fire, we required an immediate fire brigade service [3]. The existing smart fire detection system based on the centralized architecture informs the fire department but does not ensure the arrival of the fire brigade [4]. Also, they did not consider any approach to select an appropriate fire brigade among all possible options to reach the burning location in a minimum time. However, we need a trusted system that instantly informs the fire department and ensures the arrival of a fire brigade to protect from serious fire damage.

The industrial sector that belongs to enterprises such as factories or workshops (e.g., chemical, plastic, and garment) suffers from an unexpected fire event which causes property damage and income loss as well. To protect against the huge damage caused due to fire, the insurance company offers fire insurance. Fire insurance covers different types of losses and provides financial support to the enterprise owner to restart its business again. For buying a suitable fire insurance policy, the enterprise owner signs a fire insurance policy agreement with the insurance company based on their terms and conditions. During a fire insurance claim, the insurance company allocates an experienced investigation officer to investigate the situation. The investigation officer collects the evidence and prepares a report, which takes several months or even years to complete the process that causes a delay in providing fire insurance claim compensation. In addition, the enterprise owner has no access to view the collected information by the investigation officer due to this an enterprise owner is inadequate to verify the correctness of the collected data, which leads to insurance fraud [5]. Thus, there is a necessity to bring transparency to the insurance system in which the enterprise owner views the collected information and relies upon the decision-making process of the insurance company.

Blockchain technology has received tremendous attention in developing many industrial applications [6] due to its unique feature such as decentralization, immutability, trust, privacy, transparency, anonymity, etc. The term blockchain came up with Bitcoin in 2008 by an unknown person known as Satoshi Nakamoto to transfer digital assets (e.g., tokens) between individuals to exemplify intermediaries [7]. The blockchain utilizes smart contracts to code agreements for establishing trust relationships among individuals. The insurance industry is adopting blockchain to transform its various insurance policies into smart contracts to automate insurance operations. The ongoing research in the insurance industry focuses on auto-mobile insurance [8], agriculture insurance [9], and healthcare insurance [10] while others on surveys and case studies [11]. However, less work is implemented to cover the fire insurance to benefit enterprises to provide the insurance claim

compensation in a short duration.

The objective of this paper is to fulfill the fire brigade service request for enterprises on time to protect them from serious fire damage and payout to the enterprise owner for the future insurance claim. The main contribution of the paper is summarized as follows:

1) A Trusted Fire Brigade Service and Insurance Claim (FLAME) system is developed using blockchain for enterprises (i.e., factories and workshops) to provide an instant fire brigade service at the enterprise location. Further, a claim settlement is performed between the enterprise owner and insurance company based on the terms and conditions of the fire insurance policy in a transparent way to protect against insurance fraud.
2) A Narrowband Internet of Things (NB-IoT) based network slice is designed to transmit various sensors data instantly to the monitoring station for real-time data analysis to protect against serious fire. In addition, a service queue length mechanism is adopted to select an appropriate fire department among all possible options.
3) A prototype of the FLAME system is developed on the Hyperledger Besu blockchain by implementing smart contracts using the Solidity programming language. Further, the Hyperledger Caliper benchmarking tool is used to evaluate the performance indicator (i.e., latency and throughput) under three different workloads. For comparison, three consensus algorithms including Istanbul Byzantine Fault Tolerance (IBFT) 2.0, Clique, and Ethash are utilized.

The rest of the paper is organized as follows: Section II and Section III briefly presents the literature review and background, respectively. Section IV describes the functioning of the proposed framework. Section V explains the implementation and Section VI concludes the paper with future work.

II. Literature Review

A. CPS and IoT-enabled Fire Detection System

In [12], the authors proposed an IoT-based fire alarm and authentication system that detects a fire and provides the location of the affected region using Global System for Mobile (GSM) communication. In [13], the authors presented an emergency response system using IoT for fire hazards that utilizes messaging queue for telemetry transport to alert the fire department. In [14], the authors designed an early fire detection system for the smart home to prevent serious hazards using multiple sensors that transmit data to the sink node through Zigbee protocol which further informs the fire department and police. In [15], the authors deployed a smart fire detection system that transmits threshold values to the control system which further sends an alert message to the owner, local police station, and fire department using GSM communication. In [16], the authors proposed an efficient architecture to detect real-time fire events using Convolution Neural Network (CNN). In [17], the authors designed an early fire detection system for forest using WSN, IoT, and image processing that transmit data to the cloud platform using GSM communication to get rid of fire events. In [18], the authors designed an early fire detection system for the residential, commercial, or industrial area using multiple sensors, and a CNN, which notify the end-user through a web-based notification system. In [19], the authors proposed a fire detection system to quickly detect forest fire using low power devices through Long Range Wide Area Network (LoRaWAN). However, the above studies fulfill their objectives but are inefficient to ensure the arrival of fire brigade services on time to cure against serious damage. Also, no mechanism is considered to select an appropriate fire department from all available choices to provide instant fire brigade service. The utilization of WSN technology in the existing work is unable to provide best Quality of Service (QoS), high bandwidth, maximum coverage, and low latency to massive IoT devices for critical data analysis. Further, the CPS and IoT-enabled fire detection system did not integrate the blockchain technology to provide transparency, and trust while calling a fire brigade service for the end-user.

B. Blockchain-based Insurance System

In [20], the authors presented a decentralized interplanetary file system and blockchain-enabled auto-insurance system that regulates insurance claim activates and automates payments. In [21], the authors proposed a blockchain-based framework for auto-insurance claims in which automated vehicles utilize sensors to share information. In [22], the authors designed a blockchain-based framework for an insurance use case to offer a fine-grained access control using smart contracts and transaction processing for the insurance process. In [23], the authors developed an automatic medical insurance claims service system using blockchain and smart contracts to solve risk control and anti-money laundering problems. In [24], the authors proposed a collaborative blockchain-based insurance system to automate the insurance policy, claim handling, and payment using smart contracts. In [25]-[26], the author presented a prototype of fine-grained transportation insurance based on hybrid blockchain and IoT to share the real-world data of drivers using a Global Positing System (GPS). In [27], the authors proposed a vehicle insurance system using blockchain to record vehicle insurance information which acts as evidence during disputes. In [28], the authors proposed a decentralized peer-to-peer crop insurance framework for farmers using blockchain to cover excessive rainfall risk. In [29], the authors presented a blockchain-based smart contract framework for the drought insurance system. After analyzing the existing work on the blockchain-based insurance system, most of the authors focus on auto-mobile insurance while others on healthcare and agriculture insurance. None of them considered the fire insurance to provide financial support to the policyholder under serious fire hazards.

III. Background

This section explains the fundamental concepts of network slicing and Low Power Wide Area Network (LPWAN).

A. Network Slicing

Network slicing is an evolution toward Fifth Generation (5G) technology to run many 5G services as a network slice for specific applications or use cases. The Next Generation Mobile Network alliance first introduced the network slicing concept back in 2015. Network slicing is a virtual network architecture that builds on the principle of Software Defined Networking

(SDN) and Network Function Virtualization (NFV). The SDN is used to manage traffic flow through the application programming interface of a central control panel, whereas NFV controls the lifecycle of network slices and their infrastructure resources. Network slicing creates multiple virtual networks (i.e., network slices) on top of a common physical infrastructure that comprises an independent set of logical network functions such as speed, capacity, coverage, and connectivity. These virtual networks are customized based on the specific requirement of applications, services, operators, or devices to provide a virtualized end-to-end network connectivity.

C. Low Power Wide Area Network Technology

The LPWAN is a wireless wide area network technology based on the star topology which connects several remote devices with the base station. The LPWAN is categorized into two types: cellular network-based LPWAN e.g., (a) NB-IoT, and (b) long term evolution for machine type communication (LTE-M), and non-cellular network-based LPWAN e.g., (a) LoRaWAN and (b) SigFox. The focus of this study is cellular network-based LPWAN especially NB-IoT because it works on the licensed band and does not contain any duty cycles for data transmission as compared to the non-cellular-based LPWAN. The NB-IoT is a wireless communication standard developed by 3GPP in releases 13 and 14 [30] to address the requirements of IoT applications by providing various advantages: (i) improved indoor coverage, (ii) high data reliability, (iii) reduce device power consumption, and (iv) support low data rate, (v) long-range transmission and (vi) moderate network latency (e.g., less than 10 seconds). The NB-IoT offers better scalability, QoS, security and connects thousands of devices with one NB-IoT base station, as opposed to short-distance technology such as Zigbee, WiFi, Bluetooth, etc. NB-IoT is better suited for static network devices and near real-time data analysis.

IV. FLAME: Trusted Fire Brigade Services and Insurance Claim System

In this section, a detailed overview of various essential elements such as proposed system architecture, smart fire detection system, smart contracts, and NB-IoT enabled network slice are provided to completely understand the working of the FLAME system is explained as follows.

A. System Architecture

The proposed system architecture consists of six main components: a smart fire detection system, a monitoring station, a fire department management system, an enterprise owner, an insurance company, and the blockchain network, as shown in Fig. 1. The smart fire detection system is placed inside an enterprise to capture the presence of fire to protect against serious damage. The multiple sensors record the surrounding information and wirelessly provide it to the fire alarm control panel. The fire alarm control panel periodically passes received information to the monitoring station through gNodeb. The monitoring station is connected with the blockchain network via a validator node. The validator node performs read/write/validate operations. It continuously analyses received data to identify whether multiple sensors values are reaching the threshold or not. If yes, the monitoring station calls a smart contract and forwards the threshold value on the blockchain network. The fire department management system is also linked with the blockchain network through the validator node. It manages multiple fire departments and keeps their service queue length information. The fire department management system selects a suitable fire department with a minimum service queue length and forwards the received fire brigade service request. An enterprise owner is linked with the blockchain network using a besu node. The besu node only carries out read/write operations. An owner invokes a smart contract to request compensation amount if its enterprise is burnt out due to the fire. An insurance company is connected with the blockchain network using a validator node. The insurance company inspects the enterprise and based on its decision it transfers the compensation amount as tokens to the owner's wallet address by calling a smart contract on the blockchain network. The administrator is used to set up the blockchain network and associated with the blockchain using the boot node. The blockchain network provides a platform for all types of nodes to interact with smart contracts to perform various blockchain-related operations for the FLAME system.

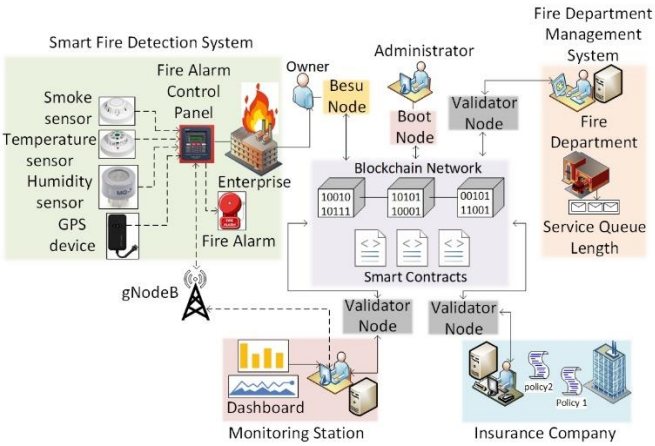

Fig. 1: Architecture of FLAME System

B. Smart Fire Detection System

It is assumed that a smart fire detection system is assembled in i^{th} owner's enterprise which provides the surrounding information to j^{th} monitoring station without human involvement, where $i \in 1,2,\dots I$. To enable the functioning of a smart fire detection system, it consists of an NB-IoT driven fire alarm control panel, IoT sensors, an actuator, and a global positing system (GPS) that work together to perform a specific task. During the assembly time, an owner identification number ($OIDN_i$) is provided to i^{th} owner to distinguish its sensor's information from others and to get the exact location of its enterprise. The multiple sensors such as a temperature sensor (T_i, t) smoke sensor (S_i, t) and humidity sensor (H_i, t) transmit the captured data to the fire alarm control panel at a certain period (t) (e.g., 5 mins.). The fire control panel collects sensors data, enterprise location (x_i^{loc}, y_i^{loc}) using GPS, owner identification number, and transmitting it to j^{th} monitoring station, where $j \in 1,2,\dots J$. The j^{th} monitoring station stores the data in its database and continuously monitors the received information. Under a threshold condition, it automatically calls a smart contract on the blockchain network for i^{th} owner's enterprise and forwards the same data. Simultaneously, the fire

alarm control panel instructs the actuator i.e., the fire alarm to alert the staff member. Also, it sends a notification with sensors values to i^{th} owner to inform about the presence of fire in its enterprise.

C. Smart Contracts

The proposed FLAME system consists of six smart contracts: register owner (Reg_Own) contract, register policy (Reg_Ply) contract, service request (Ser_Req) contract, service response (Ser_Rsp) contract, policy claim (Ply_Clm) contract, and refund token (Rfd_Tkn) contract, as shown in Fig. 2. These smart contracts bring trust while calling a fire brigade service for an owner's enterprise and transparency during the claim settlement process. The Reg_Own contract registers the owner on the blockchain network. The purpose of Reg_Ply contract is to purchase a suitable insurance policy by a registered owner for its enterprise. The Ser_Req contract is utilized by the monitoring station to request a fire brigade service on behalf of the owner of its burning enterprise, whereas Ser_Rsp inform the monitoring station that its fire brigade service request is accepted on the blockchain network and a fire brigade is allocated to reach the enterprise location. The Ply_Clm contract is used by a registered owner to perform an insurance policy claim for its burnt enterprise to receive a compensation amount to restart its business again. Finally, the Rfd_Tkn contract transfers the claim settlement amount as tokens in the owner's wallet address if all the terms and conditions are satisfied according to the purchased insurance policy.

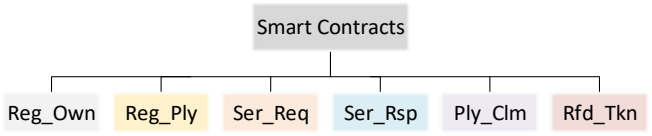

Fig. 2: Smart Contracts for the FLAME System

D. NB-IoT based Network Slicing

We designed an NB-IoT based network slicing for our proposed system, as shown in Fig. 3. To provide instant fire brigade services without delay, a set of dedicated network resources such as NFV, bandwidth, QoS, are allocated for high availability. In i^{th} owner's enterprise, the smoke sensor, temperature sensor, humidity sensor, and fire alarm are connected with the fire alarm control panel using NB-IoT technology. This fire alarm control panel contains an NB-IoT gateway for data forwarding. The fire alarm control panel is further connected with the gNodeB to send sensors information to the MEC node using the User Plane Function (UPF). The role of UPF is to accurately route the information to the correct destination to improve efficiency and end-user satisfaction. The 5G core is utilized to send the received information from the MEC node to the cloud server. The 5G core consists of Access and Mobility Function (AMF) for connection, reachability, and mobility management whereas, Session Management Function (SMF) is utilized for data inspection, routing, forwarding, handling QoS, etc. The j^{th} monitoring station is connected with the cloud server to directly monitor received sensors information for real-time data monitoring.

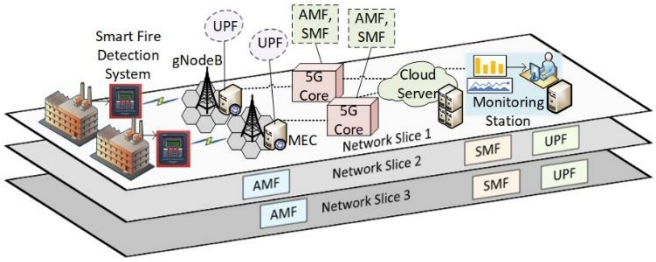

Fig. 3: Network Slicing for the FLAME System

F. Integration of Smart Contracts in FLAME System

This section explains the system initialization, and working of the proposed system for enterprises using smart contracts.

System Initialization: The administrator establishes the Besu blockchain network and deploys smart contracts for the FLAME system to perform blockchain-oriented operations. The administrator nominates a set of validator nodes such as the fire department management system, insurance company, and monitoring station to achieve consensus on the received transactions and broadcast a valid block in the blockchain network. For anonymity, the public keys and wallet addresses of the owner, fire department management system, monitoring station, and insurance company are considered, no real identities are stored on the blockchain network. Further, cryptographic tokens called FBI tokens are used to purchase an insurance policy and claim settlement in our proposed system.

ALGORITHM 1: WORK FLOW OF PHASE-I

Input: N, PH, FA, FOP, OIDN
Output: PB, PR, WA, WB, mnemonic, PN, premium amount is not received

```
1. Reg_Own contract {
2.  event Credential(uint, uint, address, uint, string);
3.    for i = 1; i <= I; i + + {
4.        struct Owner {string N_i; uint PH_i; uint EA_i; uint EOP_i,
            uint OIDN_i}
5.        function registerOwner (string N_i; uint PH_i; uint
          EA_i; uint EOP_i; uint OIDN_i) public _ownerOnly{
          Owner memory o = Owner (//function body );
          owners. push(o);
6.        emit (PB_i, PV_i, WA_i, WB_i, mnemonic);
7.     }// end function } //end For } //end contract
8. Reg_Ply contract {
9.      event policyIdentificationNumber (uint);
10.     event amountNotReceived (string);
11.     for i = 1; i <= owners. length; i + + {
12.       for k = 1; k <= K; k + + {
13.           struct Policy { string PN_i, uint PA_i; address WA_k }
14.           function registerPolicy(string PN_i, uint PA_i, address
          WA_k) public {Policy memory p = Policy {
                //function body ); policy. push(p);
15.           if (WA_k == "received PA") { emit (PN_i);
16.           else emit ("premium amount is not received"); }
17.     }// end function }} //end For } //end contract
```

Phase-I: In this phase, the owner registers itself on the blockchain network to purchase a fire insurance policy for its enterprise, as shown in Algo. 1.

1) The i^{th} owner calls Reg_Own contract for registration and provides its essential details such as name (N_i) phone number (PH_i), enterprise address (EA_i), enterprise ownership proof (EOP_i) (e.g., registry number), and owner identification number ($OIDN_i$).

2) After receiving i^{th} owner information, the blockchain returns a public key (PB_i), private key (PR_i), wallet address (WA_i), and a mnemonic.
3) The i^{th} owner invokes Reg_Ply contract to purchase a fire insurance policy for its enterprise by inserting required details such as the insurance policy name (PN_i), pay a premium amount (PA_i) using FBI tokens in k^{th} insurance company's wallet address (WA_k) and generates a signed transaction $Tx: \langle PR_i(PN_i||PA_i||WA_k)\rangle$ on the blockchain network, where $k \in 1,2,\dots K$.
4) The insurance company receives the signed transaction and check if the premium amount credit in its wallet address, generates a signed transaction $Tx: \langle PR_k(PB_i(PN_i)\rangle$ for i^{th} owner that contains a policy number (PN) for the future insurance claim. Otherwise, obtains a message that indicates the premium amount is not received. Here, PR_k represents the private key of k^{th} insurance company.

ALGORITHM 2: WORK FLOW OF PHASE-II

Input: $S^{th}, H^{th}, G^{th}, x^{loc}, y^{loc}$, OIDN
Output: SRN, request accepted, request ended
1. **Ser_Req** contract
2. **event** serviceRequestIdentificationNumber(uint);
3. **for** j = 1; j ≤ J; j + + {
4. **for** i = 1; i ≤ I; i + + {
5. **struct** Request { uint S_i^{th}; uint H_i^{th}; uint G_i^{th}, uint x_i^{th}, uint y_i^{th}, uint $OIDN_i$ }
6. **function** serviceRequest (uint S_i^{th}, uint H_i^{th}, uint G_i^{th}, uint x_i^{th}, uint y_i^{th}, uint $OIDN_i$} public { Request memory r = Request {//function body); requests. push(r); **emit**(SRN_j);
7. }// **end** function }} //**end** For } //**end** contract
8. **Ser_Rsp** contract {
9. **event** Credential(string);
10. **for** l = 1; l <= L; l + + {
11. **struct** Response {uint FDIN }
12. **function** serviceResponse (uint FIDN) public { Response memory r = Response (//function body); responses. push(r);
13. **if** (queue_length ≤ maximum) {**emit** ("request accepted");
14. **else emit** (" request is in waiting"); }
15. }// **end** function } //**end** For } //**end** contract

Phase-II: In this phase, an instant fire brigade service is provided for an owner's enterprise either in the presence or absence of it to protect from serious damage caused due to the fire, as shown in Algo. 2.

1) The j^{th} monitoring station remotely monitors i^{th} owner's factory smoke sensor, temperature sensor, and humidity sensor values.
2) Once all sensor's values are continuously reaching above the threshold, j^{th} monitoring station invokes Ser_Req contract to request a fire brigade service for i^{th} owner's enterprise. The j^{th} monitoring station sends current threshold values of smoke sensor (S_i^{th}), temperature sensor (T_i^{th}), humidity sensor (H_i^{th}) along with enterprise location, and owner identification number. Further, it generates a signed transaction $Tx: \langle PR_j(S_i^{th}||T_i^{th}||H_i^{th}||x_i^{loc}||y_i^{loc}||OIDN_i)\rangle$ on the blockchain network. Here, PR_j indicates the private key of j^{th} monitoring station. In return, j^{th} monitoring station receives a fire brigade service request number (SRN_j).
3) The l^{th} fire department management system receives the signed transaction data and searches for a suitable fire department with a minimum service queue length in its database. Further, it informs the selected fire department to schedule a fire brigade at i^{th} owner's enterprise location earlier. Simultaneously, l^{th} fire department management system calls Ser_Rsp contract, enters the selected fire department identification number $(FIDN)$ along with a message that represents j^{th} monitoring station fire brigade service request accepted, and generates a signed transaction $Tx: \langle PR_l(PB_j(FIDN||request\ accepted)\rangle$ on the blockchain network. Otherwise, obtain a message that reflects j^{th} monitoring station fire brigade service request is in waiting due to the unavailability of fire brigades. Here, PR_l and PB_j indicates private and public keys of l^{th} fire department management system and j^{th} monitoring station, respectively, where $l \in 1,2,\dots L$.
4) After providing the service at i^{th} owner's enterprise location. The selected fire department offline informs to l^{th} fire department management system. Further, l^{th} fire department management system generates a signed transaction $Tx: \langle PR_l(PB_j(OIDN_i||request\ ended\rangle$ for j^{th} monitoring station which contains the i^{th} owner identification number and a message that indicates j^{th} monitoring station fire brigade service request is ended.

Phase-III: In this phase, the insurance company decides whether an owner receives a claim settlement amount for its burnt enterprise or not, as shown in Algo. 3.

1) The i^{th} owner invokes Ply_Clm contract, insert the policy number (PN_i), owner identification number $(OIDN_i)$, and generates a signed transaction $Tx: \langle PR_i(PN_i||OIDN_i)\rangle$ on the blockchain network. In returns, i^{th} owner receive a policy claim number (PCN_i).
2) The k^{th} insurance company receives the signed transaction and confirm the occurrence of the fire at i^{th} owner' enterprise from j^{th} monitoring station and l^{th} fire department management system by providing its owner identification number. Both of them, return transaction identities to k^{th} insurance company $Tx: \langle PR_j(PB_k(Tx\ identity))\rangle$ and $Tx: \langle PR_l(PB_k(Tx\ identity))\rangle$ respectively for validation purposes.
3) After receiving transaction identities, the k^{th} insurance company allocate an expert investigation officer to survey i^{th} owner's enterprise location. The investigation officer prepares a report that contains the percentage of enterprise damage (PED_i), and the compensation amount (CA_i) for i^{th} owner. Both the investigation officer and i^{th} owner sign the prepared report for a mutual agreement and provide it to k^{th} insurance company. If the report fulfills all the terms and conditions, k^{th} insurance company generates a signed transaction $Tx: \langle PR_k(PED_i||CA_i||claim\ allowed)\rangle$ along with a message that indicates i^{th} owner's insurance claim is allowed. Otherwise, reflecting a message that represents an insurance claim is not allowed with a suitable reason.

4) The k^{th} insurance company calls Rfd_Tkn contract, enter compensation amount as the number of FBI tokens, i^{th} owner's wallet address, a message that indicates fund transferred and generates a signed transaction $Tx: \langle PR_k(CA_i||WA_i||fund\ transferred\rangle$ on the blockchain network.

ALGORITHM 3: WORK FLOW OF PHASE-III

Input: PIDN, OIDN, CA, WA
Output: PCN, Fund transferred
1. **plyClm** contract {
2. **event** generatePolicyClaimNumber(uint);
3. **for** i = 1; i <= I; i + + {
4. **struct** Request { string PCN_i, $OIDN_i$; } Request [ ]requests;
5. **function** claimRequest (uint PCN_i) public {
Request memory r = Request (//function body);
requests. push(r); **emit** (PCN_i);
6. }// **end** function } //**end** For } //**end** contract
7. **Rfd** contract {
8. **event** endPolicy(string);
9. **for** k = 1; k <= K; k + + {
10. **for** i = 1; i <= I; i + + {
11. **struct** Response { string CA_i; address WA_i}
Response [ ]responses;
12. **function** claimResponse (string CA_i, address WA_i) public {
Response memory r = Response (//function body);
response. push(r); **emit** ("Fund transferred");
13. }// **end** function }} //**end** For } //**end** contract

V. IMPLEMENTATION

This section explains the network configuration, performance indicators, and blockchain-oriented results for the proposed system.

A. Network Configuration

To start the Besu blockchain network for the proposed FLAME system in the docker environment, the prerequisite files are installed. The Besu blockchain network runs on Ubuntu 20.04 LTS with 32 GB RAM and 16 CPU. The Hyperledger Besu v20.10.4 with JDK 15 is installed. The docker-engine v20.10.5, docker-compose v1.29.2, and npm v6.14.14 are used to set up the Kubernetes environment to pull Besu images inside the cloud server using Helm charts. The boot node (i.e., administrator) configures and deploys the genesis.json file on the Besu blockchain network. Further, the various smart contracts are written in the Solidity programming language on VS code editor v6.14.2. The web3.js-eea libraries are used to deploy and execute smart contracts on the Besu blockchain network. To run the experiments, a maximum of 40 nodes are created in which 1 is an administrator, 1 is a monitoring station, 1 is a fire department management system, 1 is an insurance company, and the rest are representing owners. Thus, a set of 3 validator nodes are nominated to achieve consensus in the proposed system architecture. A fire department management system is holding a maximum of 5 fire departments. Each fire department consists of 6 fire brigades to provide a fire brigade service. This means the maximum service queue length for a fire department is 6.

B. Performance Analysis and Discussion

The performance indicators are evaluated to monitor the performance of the proposed system. Three consensus algorithms i.e., IBFT 2.0, Clique, and Ethash are used to carry out the performance evaluation. Further, a Hyperledger Caliper benchmarking tool is employed to view the performance behavior of the proposed system by giving a load of 1000 transactions for three different workloads such as open, query, and transfer. For performance indicators, we considered latency and throughput, which are described as follows.

Latency: The latency is defined as the difference between transaction confirmation time and transaction submission time.

Throughput: The throughput refers to the number of transactions confirmed by the blockchain network per second.

We assumed different threshold values for sensors to identify the presence of fire in the enterprise environment, as shown in table-I.

TABLE-I: IOT SENSORS THRESHOLD VALUES

	IoT sensors			Status
Fire	Humidity (%)	Temperature (°C)	Smoke (ppm)	Fire occur
	>=48	>=53	>=210	

C. Blockchain-oriented Results

{"transactionHash":"0xb215a92e9058c1792823fcf8d884677d92868b4414851f2a5883f1a6ee091d54","status":true,"to":"0x20f0160ea04f1d8c465650416e86f5ced2159e26","blockNumber":498832,"mnemonic":"civil chase blue wife general crucial property country hamster magic property sick","walletAddress":"0xA37Dd2b61564A6Ba169135Af4e296eC06c87DDc45","publicKey":"0x04bdfe805cf3a9bfe474e1895207768e25aedaa463cb44161ed31f58deeedcec9db48941416289727ccb28ccd40b4ec9449bd712832b9e5adbb11c54198adc8a17","privateKey":"0xc00543c675be77dbb2361a1fa8380b9bfe48064ef592219fc68197a6480a17a1"}

Fig 4: Result of invoking Reg_Own contract

Fig. 4 shows the public key, private key, wallet address, and mnemonic for i^{th} owner after executing Reg_Own contract. These credentials are used while purchasing a fire insurance policy.

{"transactionHash":"0xabdbc596c967f9caf420be995a5b1ad70beb83fc1f6115cb2016639742fe0930","status":true,"to":"0x90cc68ffe6140bbdf541aaa441c0d8616561e8740","blockNumber":498957,"policyNumber":"830416712024619127367513319866724102998025310069531874013738026074699011594 54"}

Fig 5: Result of invoking Reg_Ply contract

Further, Fig. 5 indicates a policy number for i^{th} owner after calling reg_Ply contract. This policy number is utilized by i^{th} owner while applying for a fire insurance claim in the future.

{"transactionHash":"0x7a9c2c8438bb89d51c304c3bcba9b1613061bdf2bbaad98de76fdbe28bc071ec","status":true,"to":"0x09e6f5e5ea58ea80860e57feb02b8c2f99d4af6f","blockNumber":499744,"serviceRequestNumber":"3241699195802019976541893341018193751491044494539164844030451087525 2894222610"}

Fig 6: Result of invoking Ser_Req contract

Fig. 6 shows, the service request number for j^{th} monitoring station after calling Ser_Req contract. This service request number indicates that j^{th} monitoring station request is accepted successfully now a fire brigade is reaching to the i^{th} owner's enterprise location.

Fig. 7 shows, i^{th} owner claim is approved based on the insurance policy terms and conditions. Further, the k^{th} insurance company invokes Rfd_Tkn contract and transfer the number of FBI tokens as complementation amount in i^{th} owner wallet address.

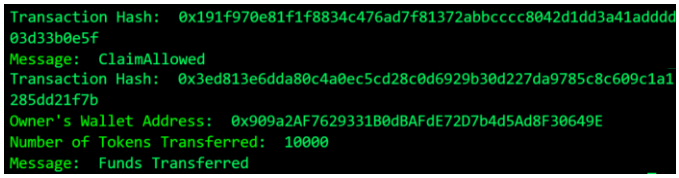

Fig. 7: Result of invoking Rfd_Tkn contract

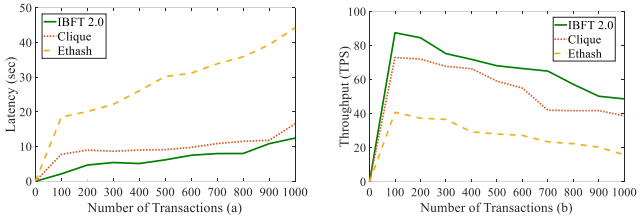

Fig. 8: Open workload performance (a) Latency and (b) Throughput

A relationship between latency and the number of transactions (i.e., 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000) for three consensus algorithms under open workload, as shown in Fig. 8 (a). The open workload includes write-only operations performed using proposed smart contracts e.g., registration of owner, registration of insurance policy, and applying for an insurance claim. The latency for IBFT 2.0 under a transaction load of 700 is 7.99 sec, whereas it is 10.88 sec, and 33.86 sec for Clique, and Ethash, respectively. It is observed that the latency for Ethash is drastically increasing for a large number of transactions and it is around 3.11x times greater than the Clique and 4.23x times than the IBFT 2.0.

A comparison for IBFT 2.0, Clique, and Ethash consensus algorithm between throughput and the number of transactions w.r.t open workload, as shown in Fig 8 (b). For IBFT 2.0, the blockchain network achieves 212.6 Transaction per Second (TPS) out of 400. In comparison, Clique and Ethash produce 102.4 and 40.95 TPS, respectively. It is identified that IBFT 2.0 is approximate 2.07x times faster than the Clique and 5.17x times than Ethash. Thus, it is concluded that for open workload IBFT 2.0 performs better for latency and throughput.

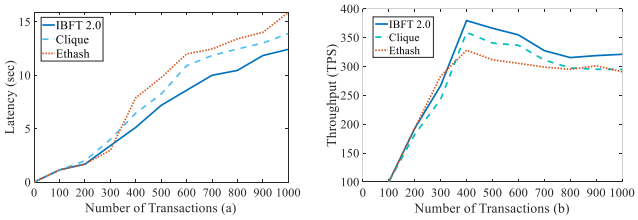

Fig. 9: Query workload performance (a) Latency and (b) Throughput

A relationship between latency and the number of transactions corresponding to three types of consensus algorithms under the query workload, as shown in Fig. 9 (a). The query workload represents read-only operations in the proposed framework e.g., return of the policy number, service request number, transaction identities, request accepted, request ended, etc. Initially, Ethash is achieving low latency till 300 transactions which are later increasing as compared to others. The latency for IBFT 2.0 is 9.99 sec under a load of 700 transactions whereas it is 11.84 sec and 12.45 sec for Clique and Ethash, respectively. Thus, it is observed the latency for all three consensus are very close to each other as the number of transactions increases because it does not require any transaction validation.

A comparison between throughput and number of transactions for IBFT 2.0, Clique, and Ethash w.r.t the query workload, as shown in Fig. 9 (b). Initially, all three consensuses are approaching the same throughput between 182.2 and 192.8 TPS out of 200 transactions. After that, a slow change is visible in which IBFT 2.0, Clique, and Ethash are gaining a maximum throughput of 379.5, 358.9, and 338.1 TPS out of 400 transactions, respectively. It is identified that all three consensuses are linearly reducing after an interval of 400 transactions and attaining nearly the same throughput for a large number of transactions.

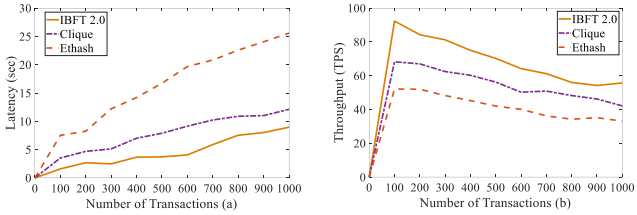

Fig. 10: Transfer workload performance (a) Latency and (b) Throughput

A comparison between latency and the number of transactions for the transfer workload w.r.t all consensus algorithms, as shown in Fig. 10 (a). The transfer workload simulates read-write operations performed using various smart contracts such as the transfer of tokens in other's wallet addresses for policy purchasing and receiving of claim compensations. The maximum latency attained by IBFT 2.0 for 1000 transactions is 8.97 sec, whereas it is 12.13 sec and 25.59 sec for Clique and Ethash, respectively. It is observed that IBFT 2.0 outperforms approximately 1.36x times better than Clique and 2.98x times Ethash.

A comparative analysis between throughput and number of transactions for three types of consensus algorithms under transfer workload, as shown in Fig. 10 (b). All three algorithms achieve maximum throughput i.e., 92.3, 68.23, and 52.10 TPS respectively over a load of 100 transactions and linearly decreasing as the number of transactions increases. The throughput of IBFT 2.0 at 1000 transactions is 55.71 TPS. In comparison, Clique achieves 42.12 TPS and Ethash fulfills 33.15 TPS. It is evident from the result, Ethash is 1.68x times and 1.28x times lesser than Ethash and Clique, respectively. It is identified that IBFT 2.0 performs well in the case of latency and throughput.

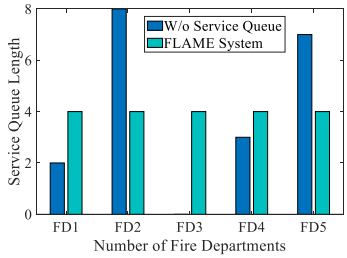

Fig 11: Impact of Service Queue Length

A comparison between service queue length and the number of fire departments (i.e., FD1, FD2, FD3, FD4, and FD5)

corresponding to the proposed FLAME system and without service queue mechanism, as shown in Fig. 11. We considered 20 fire brigade service requests from j^{th} monitoring station for modeling the result. It is observed that in our proposed system the workload on each fire department is equally distributed whereas, in the without service queue mechanism, the fire brigade service requests are randomly assigned to the fire departments. This is because of l^{th} fire brigade management system is not holding any service queue length information in its database. Due to this, the workload on FD2 and FD5 is maximum in comparison the FD3 is siting ideal to receive fire brigade service requests. The integration of service queue length in our proposed system benefits owners to receive an instant fire brigade service at its location to protect from maximum damage to its property.

VI. Conclusion

This paper proposed a Trusted Fire Brigade Services and Insurance Claim (FLAME) system for enterprises using blockchain. An NB-IoT enabled network slice is designed for the proposed system to transmit time-critical information to the monitoring station to protect against serious fire damage. Further, a service queue length approach is considered to select an appropriate fire department to fulfill the fire brigade service request. The proposed system's prototype is implemented on the Hyperledger Besu blockchain in which six smart contracts are deployed using Solidity to perform various blockchain-related operations. The performance of the proposed framework is evaluated on the Hyperledger Caliper benchmarking tool using performance indicators (i.e., latency and throughput) corresponding to three consensus algorithms such as IBFT 2.0, Clique, and Ethash. Thus, we can conclude that IBFT 2.0 based FLAME system achieved better performance in terms of latency and throughput as compared to the Clique and Ethash. In the future, we will design a real-time mobile application of our proposed system and integrate a machine learning approach for early fire detection to benefit numerous enterprises.

References

[1] K. J. Park, R. Zheng, and X. Liu, "Cyber-physical systems: Milestones and research challenges," *Comput. Commun.*, vol. 36, no. 1, pp. 1–7, 2012, doi: 10.1016/j.comcom.2012.09.006.

[2] Bhawana and S. Kumar, "A Review on Cyber-Physical Systems based on Blockchain: Possibilities and Challenges," *2021 IEEE 6th Int. Conf. Comput. Commun. Autom. ICCCA 2021*, pp. 691–696, 2021, doi: 10.1109/ICCCA52192.2021.9666299.

[3] D. Toradmalle, J. Muthukuru, and B. Sathyanarayana, "Certificateless and provably-secure digital signature scheme based on elliptic curve," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 4, pp. 3228–3231, 2019, doi: 10.11591/ijece.v9i4.ppxx-xx.

[4] N. Sarwar, "and an Adaptive Neuro-Fuzzy Inference System," vol. 3, 2019.

[5] H. L. Sithic and T. Balasubramanian, "Survey of Insurance Fraud Detection Using Data Mining Techniques," no. 3, pp. 62–65, 2013.

[6] T. Lepoint, G. Ciocarlie, and K. Eldefrawy, "BlockCIS - A blockchain-based cyber insurance system," *Proc. - 2018 IEEE Int. Conf. Cloud Eng. IC2E 2018*, no. April, pp. 378–384, 2018, doi: 10.1109/IC2E.2018.00072.

[7] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Accessed: Mar. 04, 2022. [Online]. Available: www.bitcoin.org.

[8] F. Lamberti, V. Gatteschi, C. Demartini, M. Pelissier, A. Gomez, and V. Santamaria, "Blockchains Can Work for Car Insurance: Using Smart Contracts and Sensors to Provide On-Demand Coverage," *IEEE Consum. Electron. Mag.*, vol. 7, no. 4, pp. 72–81, 2018, doi: 10.1109/MCE.2018.2816247.

[9] N. Jha, D. Prashar, O. I. Khalaf, Y. Alotaibi, A. Alsufyani, and S. Alghamdi, "Blockchain based crop insurance: a decentralized insurance system for modernization of indian farmers," *Sustain.*, vol. 13, no. 16, pp. 1–17, 2021, doi: 10.3390/su13168921.

[10] L. Zhou, L. Wang, and Y. Sun, "MIStore: a Blockchain-Based Medical Insurance Storage System," *J. Med. Syst.*, vol. 42, no. 8, 2018, doi: 10.1007/s10916-018-0996-4.

[11] R. Brophy, "Blockchain and insurance: a review for operations and regulation," *J. Financ. Regul. Compliance*, vol. 28, no. 2, pp. 215–234, 2020, doi: 10.1108/JFRC-09-2018-0127.

[12] A. Imteaj, T. Rahman, M. K. Hossain, M. S. Alam, and S. A. Rahat, "An IoT based Fire Alarming and Authentication System for Workhouse using Raspberry Pi 3," *ECCE 2017 - Int. Conf. Electr. Comput. Commun. Eng.*, no. February 2010, pp. 899–904, 2017, doi: 10.1109/ECACE.2017.7913031.

[13] R. K. Kodali and S. Yerroju, "IoT Based Smart Emergency Response System for Fire Hazards," *2017 3rd Int. Conf. Appl. Theor. Comput. Commun. Technol.*, pp. 194–199, 2017.

[14] F. Saeed, A. Paul, A. Rehman, W. H. Hong, and H. Seo, "IoT-Based Intelligent Modeling of Smart Home Environment for Fire Prevention and Safety," 2018, doi: 10.3390/jsan7010011.

[15] H. Alqourabah, A. Muneer, and S. M. Fati, "A Smart Fire Detection System using IoT Technology With Automatic Water A Smart Fire Detection System using IoT Technology With Automatic Water Sprinkler," no. October, pp. 2994–3002, 2020, doi: 10.11591/ijece.v9i4.ppxx-xx.

[16] D. Abeyrathna, P. C. Huang, and X. Zhong, "Anomaly proposal-based fire detection for cyber-physical systems," *Proc. - 6th Annu. Conf. Comput. Sci. Comput. Intell. CSCI 2019*, pp. 1203–1207, 2019, doi: 10.1109/CSCI49370.2019.00226.

[17] A. Sharma, P. K. Singh, and Y. Kumar, "ur na l P re of," *Sustain. Cities Soc.*, p. 102332, 2020, doi: 10.1016/j.scs.2020.102332.

[18] R. A. Sowah, K. Apeadu, F. Gatsi, K. O. Ampadu, and B. S. Mensah, "Hardware Module Design and Software Implementation of Multisensor Fire Detection and Notification System Using Fuzzy Logic and Convolutional Neural Networks (CNNs)," vol. 2020, 2020.

[19] N. C. Gaitan and P. Hojbota, "Forest Fire Detection System using LoRa Technology," vol. 11, no. 5, pp. 18–21, 2020.

[20] N. Nizamuddin and A. Abugabah, "Blockchain for automotive: An insight towards the IPFS blockchain-based auto insurance sector," *Int. J. Electr. Comput. Eng.*, vol. 11, no. 3, pp. 2443–2456, 2021, doi: 10.11591/ijece.v11i3.pp2443-2456.

[21] C. Oham, R. Jurdak, S. S. Kanhere, A. Dorri, and S. Jha, "B-FICA : BlockChain based Framework for Auto-insurance Claim and Adjudication," pp. 1–10.

[22] B. Lakshma Reddy, A. Karthik, and S. Prayla Shyry, "A blockchain framework for insurance processes in hospitals," *Int. J. Recent Technol. Eng.*, vol. 7, no. 5, pp. 116–119, 2019.

[23] C. Chen, Y. Deng, W. Tsaur, C. Li, C. Lee, and C. Wu, "A Traceable Online Insurance Claims System Based on Blockchain and Smart Contract Technology," pp. 1–37, 2021.

[24] F. Loukil, K. Boukadi, and R. Hussain, "CioSy : A Collaborative Blockchain-Based Insurance System," pp. 1–15, 2021.

[25] Z. Li *et al.*, "Blockchain and IoT Data Analytics for Fine-grained Transportation Insurance," pp. 1022–1027, 2018, doi: 10.1109/ICPADS.2018.00137.

[26] Z. Xiao, Z. Li, P. Chen, and W. Liu, "Blockchain and IoT for Insurance : A Case Study and Cyberinfrastructure Solution on Fine-Grained Transportation Insurance," no. November, 2020, doi: 10.1109/TCSS.2020.3034106.

[27] M. Demir, "Blockchain Based Transparent Vehicle Insurance Management," pp. 213–220, 2019.

[28] V. Iyer, K. Shah, S. Rane, and R. Shankarmani, "Decentralised Peer-to-Peer Crop Insurance," pp. 3–12, 2021, doi: 10.1145/3457337.3457837.

[29] N. Tq, D. Ak, and T. Lt, "NEO Smart Contract for Drought-Based Insurance," *2019 IEEE Can. Conf. Electr. Comput. Eng.*, pp. 1–4, 1982.

[30] "Release 13." https://www.3gpp.org/release-13 (accessed Feb. 28, 2022).