Functional Architectural Design of a Digital Forensic Readiness Cybercrime Language as a Service

Stacey Omeleze Baror[1], Richard Adeyemi Ikuesan[2] and Hein S. Venter[3],
DigiForS Research Group, University of Pretoria, South Africa[1,3] and Computing and Applied Technology Department, College of Technological Innovation, Zayed University, UAE[2]

stacey.baror@cs.up.ac.za[1]
richard.ikuesan@zu.ac.ae[2]
hventer@cs.up.ac.za[3]

Abstract: Developing a generic digital forensic solution in a cloud computing platform that can address the functional requirements of digital forensic stakeholders is a complex process. The solution would require a technology-independent architectural design that addresses the challenges of incident threat identification, triggering, incident threat isolation and investigation. Existing approaches are limited to the functionality that treats these four challenges individually without the due diligence to consider their interoperability. This study proposes a context-independent and technology-neutral architecture to address these issues by developing a digital forensic readiness (DFR) based on a human language communication interaction (HLI) system that could create a cybercrime language as a service (DFClaaS). The functional architectural design of the proposed DFR HLI DFClaaS system comprises microservices, layered and event/component-based architectural patterns on top of cloud architectural patterns. The DFR HLI DFClaaS system integrates flexibility and other quality requirements to separate concerns while accommodating rigid requirements like security and reliability. The developed architecture is essential for any human-centred digital forensic solution. Therefore, integrating the developed architecture presents a reliable baseline for the digital forensic community.

Keywords: Cybercrime language, Semantic identifier, NLP, Digital forensic readiness, Cybercrime lexicon.

Introduction

The purpose of digital forensic software applications is to determine the flow of events before, during and after an alleged incident. It identifies the cause of an incident and the events preceding it, proving the consistency of potential evidence recovered. Despite the available software for digital forensic incident discovery and investigations, there is a growing demand for software that can proactively address cybercrime attacks that are non-technical-based (Omeleze and Venter, 2013); (Valjarevic et al. 2014). Therefore, this means that the approach to designing such digital forensic applications must adapt to rapid software system design, development and reusable subsystems.

This study proposes a fundamental component to actualizing this goal by developing a context-independent functional architecture of a digital forensic readiness (DFR) framework based on natural human language communication and interaction (HLI) that uses service to show the various components, such as cybercrime language as a service termed digital forensic readiness cybercrime language as a service (DFCLaaS). Investigation of digital forensic in a cloud could be infeasible due to the volatile nature of cloud instances. One approach often employed to address the anti-forensic characteristic is the software deployment technique as a forensic readiness process. Given the volatile nature of cloud instances, studies have adduced that digital forensic readiness is one of the ways to safeguard cloud computing (Singh et al., 2022).

In a previous study (Baror et al., 2021), the digital forensic readiness (DFR) framework for human-natural language interaction was proposed, albeit without a functional design architecture. For this study, an architectural design employs a technology-neutral software requirement specifications design and the use-case responsibility-driven by analysis and design (URDAD) methodologies. The URDAD is a service-oriented method to construct a domain-specific language (DSL) (Solms et al., 2011). It uses microservices architecture in a logical design of a DF tool. The separation of concerns is achieved at both the abstraction level and deployment phases using model-view-controller architectural patterns. Therefore this study proposed a functional architectural design of a DFR HLI cybercrime language as a service.

This paper proposes a context-independent and technology-neutral functional architectural design of a digital forensic readiness software application using human language interaction. The proposed approach applies to any digital forensic readiness solution, especially for the SaaS public cloud. This study considers a context-independent approach to digital forensic readiness. Furthermore, this study introduced microservice architecture as an independent interface for forensic readiness without loss of investigation interactivity. Again,

this provides a comprehensive taxonomy of stakeholders in the forensic process through a use-case scenario to address different stakeholders' concerns during the tool design and development.

This paper is structured as follows: An introductory overview of a human-natural language interaction system's digital forensic readiness architecture. Then a presentation of a proposed approach and evaluation of the architectural design. A conclusion with proposed future studies.

The Proposed DFR HLI DFClaaS System

In order to improve the efficiency, performance and scalability of any extensive system, serverless computing is one approach. Serverless computing enables cloud services to provide a big data-focused architecture to users. The architectures are made up of various sub-services (i.e., microservices) that facilitate the end-to-end extract, transform and load (ETL) workflow needed to accomplish data analysis tasks (Chen et al., 2013). This study proposed a digital forensic readiness framework that utilizes human-natural language interaction, referred to as the DFClaaS system, that leverages microservices.

The DFClaaS system presents technology-neutral software requirement specifications, starting with the functional design of a digital forensic readiness application. It uses the use-case responsibility-driven analysis and design (URDAD) methodology to achieve the user's needs Solms et al. (2016). This is because the URDAD is a domain-specific software design method. Furthermore, detailed software requirements specifications (SRS) and engineering process is employed to justify the architectural decisions at various steps of the system development. The functional requirements of the DFR HLI DFClaaS system are presented as the first step designs that trigger the architectural decisions.

2.1 The Functional Requirements of DFR HLI DFClaaS System

The functional requirements of a system are presented as a natural, plain and understandable language to all stakeholders. It identifies and defines the set of system components, inputs, the behaviour of the information at processing, and the expected, resulting output (Len Bass Paul Clements 2012). It focuses on the behavioural conditions that describe the use cases by capturing the role of the system's users and applying their various functions to the system (Len Bass Paul Clements, 2012). Functional requirements thus provide an in-depth view of the various subsystems of the DFClaaS.

Table 1: DFR HLI DFClaaS Functional Requirements

ReqNo	Requirements Description	ReqNo	Requirements Description
FR1	The DFR HLI DFClaaS system must enable users to sign up and create an account	FR2	The DFR HLI DFClaaS system must allow authorized users to access resources.
FR3	The DFR HLI DFClaaS system should restrict users' actions and access to data based on their user permissions	FR4	The system should be able to analyze the parsed data to determine anomalies, patterns, and relationships within the data.
FR5	The DFR HLI DFClaaS system should allow DF investigators and other authorized users to input search query criteria	FR6	The system must be able to auto-identify and verify ISP and cloud service provider.
FR7	DFR HLI system should allow the system manager (admin) user to maintain and manage the permission of all users registered to access the system	FR8	The DFR HLI DFClaaS system should be able to analyze the parsed data to determine anomalies, patterns, and relationships within the data.
FR9	The users should be able to view, update and edit submitted cybercrime incident	FR10	The system must validate the date and time of incident report submission
FR11	The users must be able to notice if a user continually uploads a cybercrime reports	FR12	The system must enable users to report cybercrime without the requirement to submit any personal details.
FR13	The DFR HLI DFClaaS System must be able to allow users to report cybercrime	FR14	The System must allow users to describe cybercrime incidents in text, and details of the incidents, such as time, date, and location.
FR15	The system should identify anomalies within the parsed data	FR16	The system should analyze semantics, and sentiments within the parsed data.

ReqNo	Requirements Description	ReqNo	Requirements Description
FR17	The system should identify patterns within the text data.	FR18	The system should determine relationships within the parsed text data semantics and sentiments
FR19	The system should visualize the gathered and analyzed data in multiple ways	FR20	The system should be able to allow users to subscribe to the DFR cybercrime semantic and language identifier.
FR21	The system should display a map showing analyzed and gathered information.	FR22	The system should display a graph link diagram showcasing relationships between the data.
FR23	The system should notify a user if an anomaly occurs.	FR24	The system should visualize the gathered data in a timeline diagram.
FR25	The system should be able to give access to authorized and verified users to download and use the DFR semantic trigger API.	FR26	The system should display a graph link diagram showcasing relationships between the data..
FR27	The system should be assigned roles to users such as the DF investigator to gain access to the system.	FR28	The system should be able to allow authorized independent legal or judiciary teams to verify the validity of test data used in the development of the cybercrime language trigger
FR29	The system must be able to show the processes used to preserve the potential digital evidence.	FR30	The authorized users should be able to upload the forensic reports of past cybercrime incident investigations to the digital forensic cybercrime language database
FR31	The system should provide access to an authorized user to activate and use the DFR semantic trigger API.	FR32	The system should allow users to exit the use of the DFR HLI DFCLaaS application system at any given time.
FR33	The DFR HLI DFClaaS system needs an interface to com- munícate with the database	FR34	The DFR HLI DFCLaaS system needs to send an email to the System Admin (manager) that adds cybercrime reports.

The functional requirements (Table 1) should be addressed to design the architecture of any software application system. The users' requirements align with the system's overall requirements, so the properties of the architectural and requirement attributes are adequately selected. The next section presents a conceptual design of the proposed DFR HLI DFClaaS framework.

2.2 Microservices Architectural Design Structure of the DFR HLI System

The architecture at the top-level design is the cloud reference architecture (Len Bass Paul Clements, 2012). The microservice architectural styles are chosen as the core architectural patterns to implement the system, as shown in Figure 1. The individual microservice is further represented as a layered architecture with a model-view-controller architecture style to support the component-to-component and user interaction flexibility, as shown in Figure 2. The system consists of functionality and connections with various parts (see Figure 1). Therefore, using microservice architectural styles achieves loosely coupled. The choice of microservices is such that re-usability and flexibility quality attributes are met without trade-off security. For the DFClaaS system, the microservice architectural style allows for a large application to be subdivided into smaller independent services, whereby each service has its role and responsibility.

Firstly, as depicted in Figure 1, the microservice architecture is built on layered architectural pattern depicted in Figure 2. Then an event/component-based architecture to allow for a loosely coupled system and an independent performance-enhanced system, as shown in Figure 2. The microservice function exposes and consumes functionality using contracts, events, and messages. The component-based architecture does create reusable functional and logical components of the DFR HLI DFClaaS design that are location transparent and expose well-defined communication interfaces. The link between these components and the upper layer is the API. Thus, the API serves as the primary interface gateway for the system. The benefits of a microservice architecture are individual processes that communicate with each other over the network to fulfil a goal.

2.2.1 API Gateway

The API gateway service of the DFR HLI DFClaaS system facilitates the integration. It integrates external services, such as security and other device-specific APIs. The API gateway provides a single entry point for all the client applications to

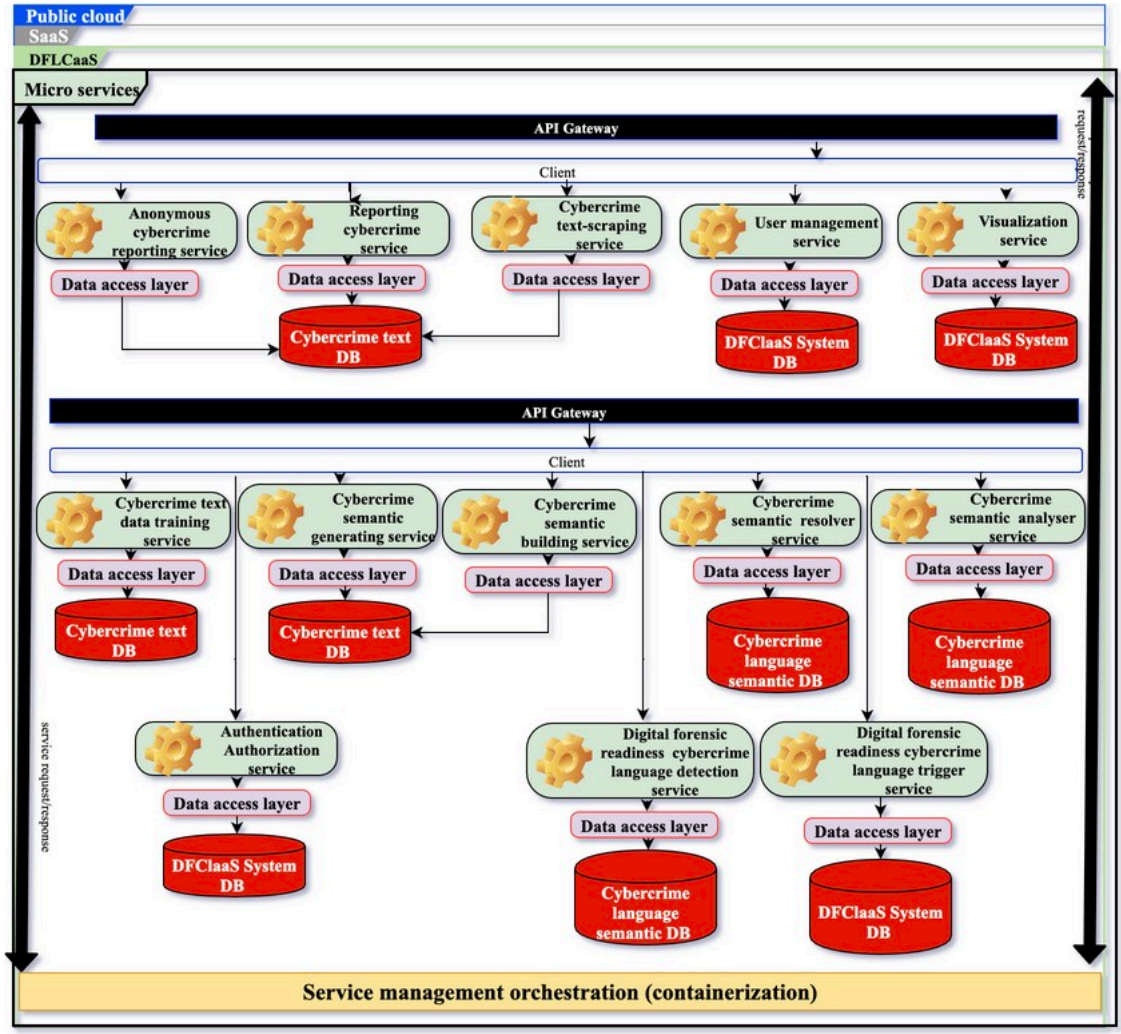

Figure 1: The Microservices Architectural view of the DFR HLI DFClaaS System

communicates with the service requests and receives responses per the request. The API gateway also facilitates communication between the different services. Heavy-tailed calls could quickly slow down the system's performance at the user interface. Web interfaces built on single-page applications (SPA) frameworks such as Angular and React (further explained in future work that presents the prototype implementation of the DFClaaS system) could encounter similar issues. The integration requirements ensure successful deployment criteria following the architectural structure. The DFR HLI DFClaaS system consists of the following services:

2.2.2 Anonymous cybercrime reporting service

The anonymous cybercrime reporting service allows victims of cyberattacks to report the incident without providing any reporter's personal information. They also facilitate the provision of cybercrime strings to increase the volume of cybercrime data. The value of the string data provided by the victims is an input used to build the database that further creates the cybercrime semantic, lexicon and language components of DFClaaS.

2.2.3 Reporting cybercrime service

The reporting cybercrime service of the system is aligned with data gathering of the interactive communication between the cybercrime victim and the cyber attacker. In the process, collect and store data input for building cybercrime language patterns. The reporting cybercrime service gathers data from external sources that can be,

parsed, cleaned and trained using both the natural language processing techniques and the machine learning components (see Figure 1). During data processing, string data parsing determines the sentiment and builds semantics to create a DFR cybercrime language. This is achieved by running the parsed string data through the deep learning model to measure the sentimental, numeral and contextual differences between the reported and stored historical cybercrime string data in the process identifies anomalies.

2.2.4 Cybercrime text scraping service

The cybercrime text scraping service is centred on allowing the system to carry out the function of web data extraction, scrapping and extracting string data from web pages by accessing the World Wide Web using the web browser. It is one of the means to increase the volume of the cybercrime data set available for training. The cybercrime text-scraping service automates the processes of using a bot or web crawler to gather cybercrime-specific string data.

2.2.5 User management service

The user management service is the human interface interacting with the system. It can be a web or mobile app, the application's client side. The model-view-controller of the layered architecture separates the user interaction from other parts of the system that is a microservice that contains the structure of the DFR HLI DFClaaS (Baror et al., 2021) (Baror et al., 2020),(Baror and Venter, 2019). The microservice architecture facilitates all users' actions, and layered architecture manages activities such as logging in, creating and updating profile details to access the system, and uploading, viewing or accessing the DFR HLI application system. The layered architecture comprises the access, business logic, and persistence layers. The access layer allows communication between the user management service through a controller, while the business layer contains all the logic for the user management service. The persistence layer involves storing, retrieving, updating, and the atomicity, consistency, isolation, and durability (ACID) functions. The ACID properties of the database transactions ensure that the cybercrime data used to develop the DFClaaS are valid.

2.2.6 Visualization service

The data visualization service provides the functionality that allows authorized users to visualize raw data, analyze it, identify patterns, and enable the user to use the historical string data to visualize cybercrime language patterns in the DFClaaS system. This service benefits the users; it enhances the usability quality requirement. The model-view-controller architectural patterns, to be discussed in detail in section 2.3.1, are employed to achieve the visualization functions of the system. The view (visualization) is a separate entity from the model controller. Another advantage of having the visualization services of the system is that it allows for easy modifications, interoperability and integration of the visualization component.

2.2.7 Cybercrime text-data training service

The training of the cybercrime text data employs a deep learning auto-encoder technique that narrows the accuracy of the resultant cybercrime string data trigger using recurrent neural networks (RNN). The RNN relies on feature extraction to extract salient features from documents represented using word embedding. The RNN processes sequential temporal information, such as sentences, frequently added to the reported cybercrime database (see Figure 1). The autoencoder deep learning service learns efficient data representation (encoding) by training the network to ignore signal (noise). It is associated with the transformation and extraction of features to establish a relationship between the stimuli, the historical cybercrime text data, and associated neural responses, in the form of a 'new' document or other form of string data. In training the collected cybercrime text data, the propagation of an accurate forensic process is first employed. This implies that the data collected must follow the rule of evidence to maintain the chain of custody and rule of evidence. This is by applying strict access control, maintaining logging, and authorization mechanisms

The central concept of the DFR HLI system is to build an all-in-one digital forensic readiness system that uses reported cybercrime and cybercrime-related data sets to carry out data analysis, feature, semantic and lexicon extractions and to create a digital forensic investigation-ready output that could use the user's language interaction styles as trigger detection.

2.2.8 Cybercrime semantic builder service

The cybercrime semantic builder service uses the input from the generated semantic string. The cybercrime semantic database uses a lexical-semantic analyzer to determine the validity of the available cybercrime string. The semantic information is analyzed for consistency and similarities to the historical cybercrime string. The cybercrime semantic database builder provides the platform to generate semantics uniquely available to forensic investigators.

2.2.9 Cybercrime semantic analyser & resolver service

The cybercrime semantic resolver in this study concerns meanings, words reference (denotation) and associated concepts (connotations). It further focuses on semantic reconciliation to determine the semantics value of the historical cybercrime text data. In this case, the source is meant to compare incoming text data to the existing database of the receiving API in the language service. The function is to create a valid semantic to construct a digital forensic cybercrime language trigger service. In NLP, the two main components of semantics are logical and lexical semantics (Hirschberg and Manning, 2015),(Zellers et al., 2019), which are the baseline for developing the DF cybercrime language. The cybercrime semantic generating, building and resolving services of the system use the cybercrime text data for lexical semantic functions, which are some of the processes employed to create the DFClaaS service. The authors (Baror et al., 2021) summarised the lexical semantics that analyzes meanings of the cybercrime text that have been gathered from morphology similarity, sentiment and the relationship between the words. The cybercrime semantics are generated in two instances, from the cloud user's platform, for example, the SMEs business owners and the reported cybercrime database service (see Figure 1). The analyser extracts sentiments, language patterns and semantics to identify potential trends within a cybercrime corpus. The corpus are further explored with deep learning autoencoder, recurrent neural networks, feature extraction and clustering deep learning techniques to continually analyze the 'newly' submitted and existing text data stored in the historically reported cybercrime text data Figure 1. The cybercrime semantic resolver compares the instances and selects the appropriate option as output. In cases where instances are not resolvable, the corresponding data is further parsed over to the text-data training service and semantic builder service for consideration.

2.2.10 Digital forensic readiness cybercrime language detection and trigger service

The digital forensic readiness cybercrime language detection and trigger services are concerned with the forensic readiness language of the DFR HLI DFClaaS framework. In an earlier research (Baror et al., 2021), the authors described the process employed to extract semantic value from historical cybercrime text data in a SaaS cloud platform. It performs a similarity check between the historical cybercrime text data and the SaaS cloud instance semantic to activate the detection and the trigger service. It combines a cloud-based lexicon and a cybercrime lexicon executed with pattern observation and extraction that assumes a phrase's collective polarity is the sum of member words' polarity in conjunction with the cloud lexicon comparison to extract a potential cybercrime language trigger. For the detection to occur, actions like validation of the semantic similarity of the text data of an 'in-flow' of SaaS cloud instance and the corresponding cybercrime historical text data in the cybercrime database are fed to the training services of the system. Furthermore, the activities that the cybercrime language detection implements in the NLP pipeline, such as the parsing service that provides the functionality of extracting and parsing text data in a format that will be useful for creating cybercrime semantics and the language trigger service. Both contents are fused into the cybercrime semantic database builder to activate the language detection and trigger services.

2.2.11 Authentication & Authorization services

The function of the authentication and authorization services of the DFR HLI DFClaaS system handles two aspects: verifying a user's identity to allow the user access the DFR HLI DFClaaS system. While the authorization functions enable the user permission to access specific resources, the user is supposed to gain access to the functions, components and other resources of the DFR HLI DFClaaS system, ensuring that users are who they claim to be. Furthermore, an authorized user's access is created using specific user details such as identity number, name, surname, email, and a random password. An authorized user's identity (ID) information is static and constitutes the primary key in the ACID domain of the system. As indicated in section 2.2, the core architectural style of the DFR HLI system is the microservice architecture. The following section describes how the architectural styles are applied to the DFR HLI system.

2.3 Layered Architectural Patterns of the DFR HLI DFClaaS System

Figure 2 presents the third level of granularity of the architecture of the DFR HLI DFClaaS system. The first level of architectural granularity is cloud computing architecture. The microservices architecture follows this at the second level. At the third level, the microservice architecture hosts layered architectural patterns. Finally, model-view-controller (MVC) architecture is presented at the fourth level of granularity.

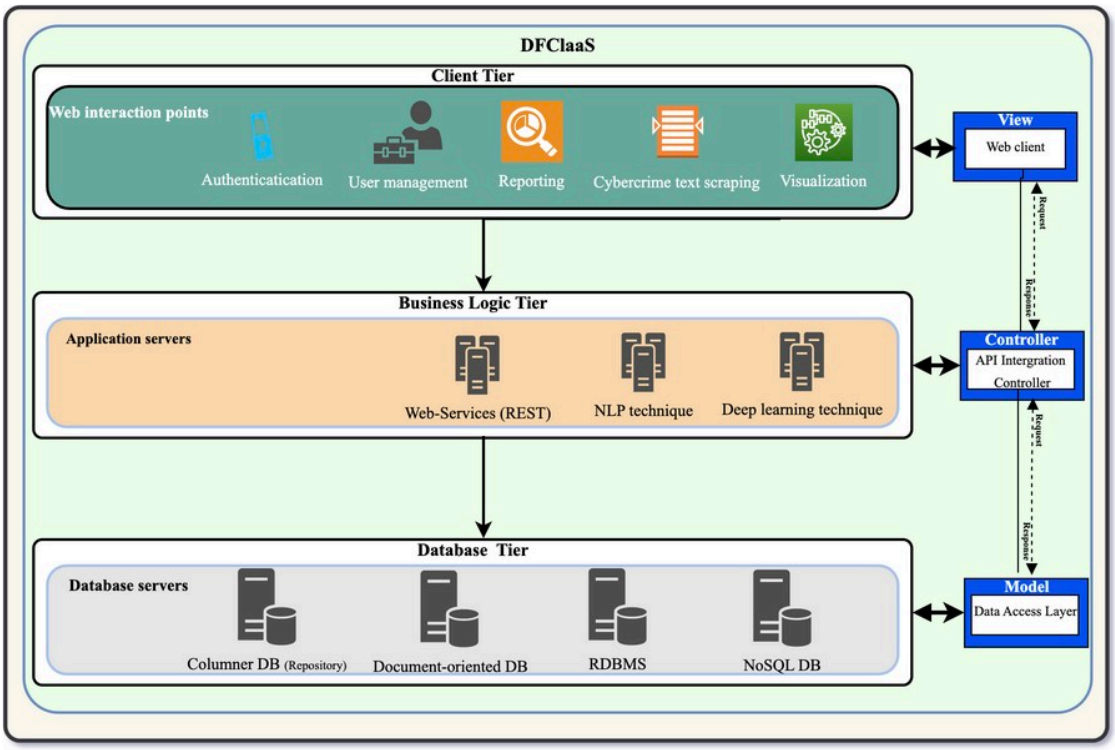

Figure 2: Layered Architectural Patterns of the DFR HLI DFClaaSystem

The architectures are displayed in Figure 2, showing layered architecture. The client tier consists of the actions that users can perform on the web interface of the DFR HLI DFClaaS system. It consists of the authentication, user management, cybercrime reporting, cybercrime text scraping and visualization. They are connected to the web client of the MVC architectural pattern (Adam et al., 2016). The MVC addresses the need for the front end of the DFR HLI DFClaaS system, which is designed for fast dynamic changes to ensure that data is displayed and visualized accurately. It provides a reactive functional design that encapsulates the system's core requirements to make the system usable and address the digital forensic readiness needs of the users. The view component implements the actual user interface to show the state of the model (see Figure 2), while also facilitating the user's interaction with other system components. The segregation of the DFR HLI system into a client, business logic, and persistence function ensures seamless service request to the server and the flexible reusable of functions. The business logic tier is the application servers layer, which consists of logic that collaborates to manage the system's working activities. They are web services, NLP and deep learning techniques. It implement digital forensic readiness and control security access. The API controller manages the functions that ensure the physical separation of the various layer while ensuring active communication interactions. The database tier hosts and manages the columnar database that maintains the cybercrime data repository, the document DB that handles any upload of cybercrime investigation documents by digital forensic investigation. The layered architecture further segregates the system into two applications, where the client makes a service request to the server.

The services interact in messaging, and the event-driven implicit invocation approach of message sourcing of events occurs in components of the microservices. The paradigm promotes the prevention, detection, consumption of, and reaction to each of the services' various activities and events when it encounters cybercrime text-data semantics previously flagged by the NLP components. The business logic layer contains the logic of the DFR HLI DFClaaS system, like the semantic, morphology and lexical businesses rule, gathering and identifying the logic within the system that determines how the behaviour correlates to other services. Finally, the database layer handles the storage, retrieval, and updating. The system database is designed to handle continually data

increase and are open-source and scalable to handle the large volumes of data, while continuous learning of the historical cybercrime data. As shown in Figure 2, the database tier (persistence) API allows interaction between the micro-services, the layered and the MVC architecture at the database level while maintaining a decoupled state. Next, the core quality requirements that must be addressed by the DFR HLI DFClaaS are discussed.

2.3.1 The Core Quality Requirements of the DFR HLI DFClaaS System

Quality requirements of any system are identified based on the functional requirements for the DFR HLI DFClaaS are shown in Table 1.

Security The security component focuses on the need to address confidentiality, integrity, and access control while complying with legal requirements of forensic soundness and chain of evidence rules. The security needs of the DFR HLI DFClaaS system depend on the end-to-end encryption of the communication channels to ensure that attacks, such as man-in-the-middle-attack are addressed. This ensures that every activity is traceable and retractable. Other aspects to be considered in the system security setup include: (i) The system should not read or decrypt any information in a message package that is not critical for delivering the message, cryptographic hash and digital signatures are some architectural strategies that addressed this (ii) Communications with the server are encrypted, critical messages are signed to prevent tampering, and messages between two users are encrypted. (iii) Regular monitoring and backups of the database are incorporated, and the SHA512 hashing algorithm is employed. In addition, peer-to-peer encryption of the communication channel, such as AES-256 is enabled at the coding level. (iv) Users authorization is determine their access levels and privileges. (v) Role-based access control ensures that users only authorized to interact with the system's aspects they have been granted permission. (vi) Unscrupulous users could flood the system with junk data, that could create denial-of-service attacks, cross-site request forgery, service side request forgery, and similar cyber-attacks.

Usability & Availability - It ensures that the application meets user expectations and improves the users' experience. It ensures intermittent, weak connections do not degrade the user's experience, maintains reliability and refers to the system's uptime. (i) The system is designed and developed always to provide real-time, updated information on the degree of availability of the system. (ii) in database failure or corruption, the server retrieves replacement and backups. (iii) The uptime and downtime of the system are recorded regularly to calculate and improve the system's availability. (iv) The server should only have a maximum of 0.0285% (2.5) or fewer hours of downtime in a year. (v) The overall system is designed to maintain at the minimum of 95% rating over 24 hours.

Reliability The attributes focus on the application's ability to continue working as it should, irrespective of possible latency. Therefore high fault tolerance is a built-in place to ensure that the system does not crash; thorough testing is to ensure durability at all pipeline and service levels. This is maintained by thoroughly implementing the following: (i) The database update process at the microservice level, and the layered architecture are be able to roll back updates when failure occurs. (ii) The application's SLA is estimated to be at 98% at its lowest operation level. If a crash happens, the system provide adequate data on why the crash occurred, allowing the administrator to fix the error, with no single point of failure with 99% uptime. The probability of a failure occurring on demand is less than 1 in 500 instances.

Performance The system is developed to be available at least 99% of the time, so the effect of fall-back, heavy-tailed traffic, and low latency are mitigated; therefore, a 95% up-time is expected. The application compensates with the response, process request and resource utilization, more specifically, the speed at which the system loads, this quality attribute is addressed to handle call requests and returns as follows: (i) The system should be able to process 1000 transactions per second. (ii) The system should have fewer than 3 seconds response time. This will be tested with performance testing. (iii) All web pages should be downloaded from the server within 5 seconds.

Architectural strategies involve making explicit trade-offs as to which feature of a quality requirement is more important to address the for the DFR HLI DFClaaS system.

Discussion

In the last few months after the COVID-19 pandemic, cybercrime incidents have risen, (Buil-Gil et al., 2021). This, therefore, means that individuals and businesses are the most affected. This study considered cybercrime in line with the victim's and the perpetrator's communication interaction. Irrespective of the type of cyber incident, be

it technical or non-technical, some form of string data communication occurred between the human users, providing insights to investigate the incident (Baror and Venter, 2019),(Baror et al., 2021). The criteria for evaluating cloud platforms and services are security, performance, scalability, adaptability, accessibility, and usability, which are the core quality attribute of the DFClaaS system is as follow:

- Allow users to report cybercrime either anonymously or otherwise. In the process generating valid historical cybercrime string to develop cybercrime semantics, lexicon and language using deep machine learning techniques.
- To learn and build knowledge features from the cybercrime string that show patterns consisting of identifiable cybercrime strings.
- The learned historical cybercrime string data detect and trigger units to notify users, subsystem or organizations of potential cybercrime.
- The distinguishing feature of the DFR HLI DFClaaS is the 'all-in-one' deployment. That is, the data analytical functions, the natural language processing techniques, deep learning and the digital forensic incident discovery services are integrated as one 'large' system. There is also the novel concept of anonymous cybercrime report, available historical string data stored and cybercrime language services.

This paper presents a functional software architectural design of a digital forensic readiness service based on human-language interaction. The DFR HLI DFClaaS system is especially advantageous for small and medium-sized enterprises (SMEs) during COVID-19, where most organizations opt to work from home to fulfil their customers' needs (Huaman et al., 2021).

The DFR HLI DFClaaS is designed using the URDAD method. This technology-independent architecture combines the system's complexity while presenting a high flexibility that allows for service and component re-usability. The DFR HLI DFClaaS is compartmentalized to address incident threat identification, incident threat trigger, incident threat isolation, incident threat investigation and the generation of cyber semantics for future cyber incident discovery. Therefore, developing the DRF HLI application with the overall architecture outlined in this paper is unique and essential for any human-centred digital forensic solution.

One drawback of the DFR HLI system is that user's must subscribe to DFClaaS service at the SaaS level of their public cloud usage to use the system. Similarly, the anonymous cyber-attacks reporting function is one challenge of the DFR HLI system as it is a potential loop hole to flood the system with junk string data and spike traffic. The DFR HLI DFClaaS system require a large volume of cybercrime data sets to increase accuracy. This is because deep learning techniques used for The DFR HLI DFClaaS system went a step further with a built-in level of security - i.e., authorization. The security architectural style requires users to satisfy the authorization requirements, which is a critical factor to verify users' actions and roles access services (Baror et al., 2020).

Conclusion

The digital forensic field tends to focus on the aftermath of an incident. Since the COVID-19 pandemic, small and medium-scale businesses (SMEs) have been affected (Parilla, 2021) by various cybercrime attacks.

The DFR HLI DFClaaS system is designed to gather cybercrime string data, learn the interaction behaviour of the cyber criminal's communications of the data and then provide a cyber semantics, lexicons and language preparedness using NLP and deep learning techniques. This is by measuring the sentimental, numeral, and contextual differences and anomalies of cybercrime data. Since the DFR HLI DFClaaS system is designed as a microservice-based architecture, each service is presented as an entity that is a callable API object. Therefore, integrating the DFR HLI DFClaaS system into any existing digital forensic application is easy.

References

Kai Adam, Arvid Butting, Robert Heim, Oliver Kautz, Bernhard Rumpe, and Andreas Wortmann. Model-driven separation of concerns for service robotics. In Proceedings of the International Workshop on Domain-Specific Modeling, pages 22–27, 2016.

Stacey O Baror, Hein S Venter, and Richard Adeyemi. A natural human language framework for digital forensic readiness in the public cloud. Australian Journal of Forensic Sciences, 53(5):566–591, 2021.

Stacey Omeleze Baror and Hein Venter. A taxonomy for cybercrime attack in the public cloud. In International Conference on Cyber Warfare and Security, pages 505–X. Academic Conferences International Limited, 2019.

Stacey Omeleze Baror, Richard Adeyemi Ikuesan, and Hein S Venter. A defined digital forensic criteria for cybercrime reporting. In International Conference on Cyber Warfare and Security, pages 617–XVIII. Academic Conferences International Limited, 2020.

David Buil-Gil, Fernando Miro-Llinares, Asier Moneva, Steven Kemp, and Nacho D´´ıaz-Castano. Cybercrime and shifts in ˜ opportunities during covid-19: preliminary analysis in the uk. European Societies, 23(sup1):S47–S59, 2021.

Yongzhen Chen, Yi Wang, Yajun Ha, Miguel Rodel Felipe, Shuqin Ren, and Khin Mi Mi Aung. saes: A high throughput and low latency secure cloud storage with pipelined dma based pcie interface. In 2013 International Conference on Field-Programmable Technology (FPT), pages 374–377. IEEE, 2013.

Julia Hirschberg and Christopher D Manning. Advances in natural language processing. Science, 349(6245):261–266, 2015.

Nicolas Huaman, Bennet von Skarczinski, Christian Stransky, Dominik Wermke, Yasemin Acar, Arne Dreißigacker, and Sascha Fahl. A {Large-Scale} interview study on information security in and attacks against small and medium-sized enterprises. In 30th USENIX Security Symposium (USENIX Security 21), pages 1235–1252, 2021.

Rick Kazman, Len Bass, Paul Clements. Software Architecture in Practice, 3rd Edition. Addison-Wesley Professional. Part of the SEI Series in Software Engineering series, 2012.

Stacey Omeleze and Hein S Venter. Testing the harmonised digital forensic investigation process model using an android mobile phone. In 2013 Information Security for South Africa, pages 1–8. IEEE, 2013.

Eric Santos Parilla. Effects of covid-19 pandemic on micro, small, and medium-sized enterprises in the province of ilocos norte philippines. In RSF Conference Series: Business, Management and Social Sciences, volume 1, pages 46–57, 2021.

Avinash Singh, Richard Adeyemi Ikuesan, and Hein Venter. Secure storage model for digital forensic readiness. IEEE Access, 2022.

Fritz Solms, Stefan Gruner, and Cuen Edwards. Urdad as a quality-driven analysis and design process. In New Trends in Software Methodologies, Tools and Techniques, pages 141–158. IOS Press, 2011.

Fritz Solms, Priscilla Naa Dedei Hammond, and Linda Marshall. Constraints-based urdad model verification. In ENASE, pages 148–155, 2016.

Aleksandar Valjarevic, Hein S Venter, and Melissa Ingles. Towards a prototype for guidance and implementation of a standardized digital forensic investigation process. In Information Security for South Africa (ISSA), 2014, pages 1–8. IEEE, 2014.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 93–104, Brussels, Belgium, 2019. Association for Computational Linguistics. doi: 10.18653/v1/d18-1009. URL https://www.aclweb.org/anthology/D18-1009.