Digital Forensic Application Requirements Specifications Process

Stacey Omeleze[1] and Hein S. Venter[2]
Information and Computer Security Architecture (ICSA) Research Group
Computer Science Department
University of Pretoria, South Africa
someleze@cs.up.ac.za[1] and hventer@cs.up.ac.za[2]

Word Count=8590

Abstract: The requirements to identify the cause of an incident, following the trail of events preceding the incident, as well as proving the consistency of the potential evidence recovered from the alleged incident, ultimately demand a proactive approach towards the design of digital forensics (DF) applications. Success in the use of digital evidence for analysis or validation in digital forensic investigations depends on established processes, scientific methods, guidelines and standards that are used in the DF application designs and developments.

Adding legal and scientific processes capable of absorbing the constant upgrades and updates in the design of DF applications is a boost to the already existing DF processes and standards. However, there is need for such processes to be clearly defined, so that non-technical audience involved in crime investigations and decisions, can easily comprehend the DF application designs and development processes.

To proactively overcome this challenge, this paper proposes the digital forensic application requirements specifications (DFARS) process. The proposed DFARS process outlines an easy-to-apply design process for designing any DF application. It further demonstrates in a case scenario, how to apply the DFARS process using the online neighbourhood watch (ONW) system. The ONW system is a DF application that crowd-source potential digital evidence (PDE). One of the objectives of the ONW system is to increase the volume of available PDE to enhance success in prosecution of neighbourhood crime.

Therefore using the DFARS process with the ONW system as a case scenario, the result shows that the DFARS process ensures an easy application of modifiability, pluggability and reliability features at any point in the life-cycle of a DF application. This thereby accommodates the constant upgrades and changes associated with electronic devices, operating systems, hardware and other requirements. It further shows an easy-to-follow process that is understandable to both technical and non-technical audiences in the field of digital forensics.

Keywords: Digital evidence; Requirements engineering; Community-policing, Crowd-sourcing; Online neighbourhood watch; Crime; Architectural requirements.

1 Introduction

In the not-too-distant future, digital evidence may potentially be the most available evidence to conduct any crime investigation, whether it relates to cybercrime or physical crime [25]. However, digital evidence presented to a court of law may be brought under scrutiny, such as questioning the validity of the DF application used in the investigative process, leading to requests for the process employed in the design and development of the DF application, be made explicitly clear to the non-technical audience or stakeholders. Generated potential digital evidence (PDE) may be used by law enforcement agents (LEAs) or other authorised users in crime investigations, when the processes employed to acquire the evidence are easily presented to all stakeholders. The working definition of PDE used through out this paper, is that PDE is raw digital data evidence gathered from cybercrime or physical crime scenes [31], [21].

The motivation for this paper therefore, is to show a process that can be employed to design a DF application

used in the analysis of PDE, that can easily be comprehended by non technical stakeholders involved in the investigation and prosecution of crime. According to the computer emergency response team (CERT) for the Software Engineering Institute at Carnegie Mellon University [24], "requirements specification problems are the primary reason that most projects are significantly over budgeted, past schedule, reduced in scope and poor-quality applications are delivered, which turn out to be of little use once delivered, or are scrapped altogether". Having emphasised the need for requirements specifications that gives a clear definition of what needs to be addressed when building any application software, therefore the DF applications used in crime investigations must be given detailed and clear design specifications. This is a provision that makes the process employed in the DF application's design understandable to technical and non-technical stakeholders. Digital forensic applications used for PDE generation, collection and analysis must employ sound design specifications, such as requirements specifications, guidelines like the Daubert's [3]) and standards, such as the ISO/IEC 27043 [30], [31].

Therefore the problem addressed in this paper focuses on the fact that there is no easy-to-apply design process for DF applications that can be easily understood by both a technical and non-technical audience. The need for an easily understood process is because DF is a field that encapsulates law, government policies, cyber, physical and computer related crime, and thus involves a variety of professions [14]. Hence, this paper proposes the digital forensic application requirements specifications (DFARS) process as its main contribution. Other contributions of this paper are: (i) Using the proposed DFARS process to design a PDE crowd-sourcing digital forensic applications (ii) a clear presentation of related literature, showing the very little-to-no-literature for requirements specifications specifically for DF applications. (iii) A case scenario that applies the DFARS process is also presented. The case scenario focused on applying DFARS process to designing a crowd-sourcing DF application, termed the online neigbourhood watch (ONW) system, that was proposed by the authors in a previous paper [18], [21]. The ONW system is a community policing application that could be used by the law enforcement agents and other authorised users requiring potential evidence of neighbourhood crime in South Africa.

The proposed DFARS process is necessary for an easy re-build of a DF application to accommodate changes that are inherent in digital forensic investigation practices - in line with legal standards and in keeping with current trends in information security.

The remainder of this paper is structured as follows: Subsections 1.1 and 1.2 gives an overview of digital forensics and requirements engineering describing the terminologies and background knowledge. Section 2 presents the proposed DFARS process, while section 3 applies the proposed DFARS process to designing the online neighbourhood watch (ONW) system. Section 4 discusses related literature, section 5 evaluates the DFARS process, and section 6 concludes this paper. Overview of digital forensics and requirements engineering is presented next.

1.1 Digital Forensics

Digital forensics use models, techniques, processes and systems to solve problems related to incidents or disputes involving data recovery, electronic data and storage devices in legal and other related enquiries [?], [19], [4]. It is a field that works in parallel with law, science and technology to identify patterns in incidents under investigation, and that uses digital forensic standards to support or refute the usefulness of digital evidence in an enquiry [32]. Digital forensic investigation seeks to obtain justice by employing laws and regulations that govern digital evidence admissibility and it demonstrates events related to crimes using DF tools [6], [32].

According to Grobler et al. [7], digital forensics employs proactive, reactive or cohesive methods to identify or solve various crime incidents [7]. In doing so, various means such as digital forensic tools or applications, digital forensic standards, expert witness testimonies and scientific theories are adopted to achieve the objectives of digital forensic investigations. However, evidence presented to a court of law may well be brought under scrutiny and doubt cast on the application used in the analysis of evidence.

Questions may be asked, such as whether the DF application used adhered to known scientific processes, theories or methods. Such methods could include the Daubert guidelines, examining the known error rate of the application, and proof of consistency, in the form of expert witness [4]. The Daubert guidelines are used by a presiding Judge in a legal proceeding to determine when an expert's scientific testimony is based on reasoning or scientifically valid methodologies [3]. The Daubert guidelines require that the digital forensic process

or methodologies must be tested, peer-reviewed, have a level of acceptance in the scientific community, and have a known potential error rate with standards and controls. Employing a known process such as software requirements specifications when designing a DF application, enhances the DF application's validity when it is comes under scrutiny in any court of law. Requirements engineering concept is presented next.

1.2 Requirements Engineering

Requirements engineering of a software system focuses on the quality of software products. It takes into account the various aspects of a system's design decisions and the conditions required to address the system's problems[23]. The success of a software system is measured by the extent to which the various conflicting aspects and stakeholders needs are managed, while adhering to the system's intended purpose. To effectively convey these requirements, proper communication aligned with the applications' functional and architectural requirements and their constraints constitutes the foundation for achieving a solid system design[27],[16]. In any system design, the requirements engineering processes employed include the identification of the system's functional requirements (also known as the users' requirements), the architectural requirements and constraints.

Functional requirements identify and define the system components, inputs, the behaviour of the inputs and the resulting output at the system's design stages[16],[15]. They focus on the behavioural requirements that describe the use cases by capturing the role of the system's users and applying their various functions to the system. The identified needs of the users are mapped to the system's architectural requirements.

Architectural requirements are the system's components that are required to commence its high-level infrastructural design[16],[27]. These are quality requirements, architectural patterns and architectural strategies that are used to address the core quality requirements of a system and in turn fulfil the users' requirements. Complying with the quality requirements of a system is critical to the success of its core objectives.

The quality requirements are the infrastructural elements that enable the system to allay stakeholder concerns. It is a means to concretely align the desired system's objectives with the architectural roadmap by using metrics and scales to quantify the system's expected output and identify the trade-offs. For example, in fulfilling the flexibility requirements of a system, security may have be traded off. Likewise, to realise performance, reliability may be a requirement to trade off. Furthermore, some quality requirements (when overlooked at the system design stage) may require a complete rebuild of the system to incorporate the forgone attribute[27]. Examples of quality requirements of any system are security, reliability, auditability, pluggability and maintainability.

Architectural patterns, also known as architectural styles are reusable solutions that enable innovation to be incorporated at the system's design stage. They are a set of principles that can be adapted for systems[13],[17]. They provide an infrastructure to realise the quality requirements of various systems[27]. Architectural patterns include pipes- and-filters, microkernel, blackboard, layered and service-oriented architecture.

Architectural strategies (also known as architectural tactics) are used to concretely address the quality requirements of a system. Such strategies specify how to carry out a design to concretely comply with a single quality requirement[16]. This makes trade-off decisions explicit, and assist in deciding on features that best address a quality requirement. Two or more architectural strategies can be used to address one quality requirement.

Architectural constraints are the conditions attached to a system's requirements specifications[23] and they are mostly imposed by stakeholders. Constraints can be legal, environmental, economic, technological or time factors that restrict the development of a system. It is the system's task to identify a way to incorporate its constraints while at the same time meeting the various requirements made by all stakeholders.

In summary, employing requirements engineering for the design of DF applications, ensures that their architectural requirements, such as expected output, incremental pluggability and maintainability functions, are met. Moreover, requirements engineering specifications as they concern digital forensics are essential to stay current with the constant advancement in hardware devices and frequent upgrades in the operating systems (OSs) used by these devices. The author's proposed process for designing DF applications that take requirements engineering specifications into account, is presented next.

2 The Proposed Digital Forensic Application Requirements Specifications Process

To effectively design a DF application, this paper proposes a digital forensic application requirements specifications (DFARS) process. The DFARS process determines the user's and application's needs, while incorporating the architectural requirements and constraints at all levels of the application's design.

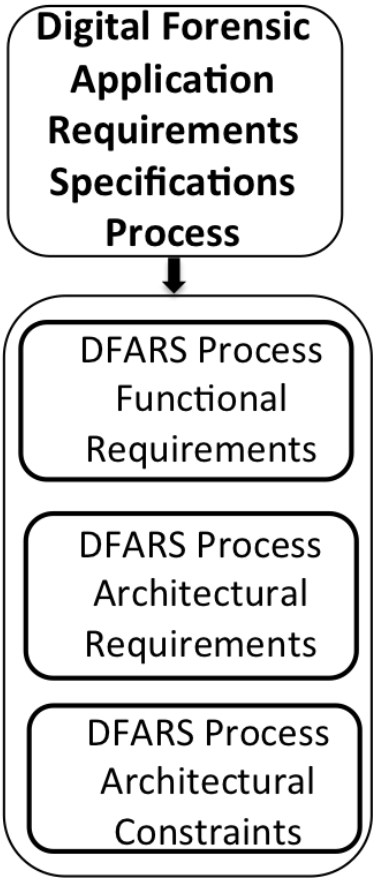

Figure 1: Digital Forensic Application Requirements Specifications Process

The "users" in this context are potentially the digital forensic investigators or examiners, the law enforcement agents and the members of the judiciary. As depicted in Figure 1, the DFARS process consists of three sub-processes **(i) DFARS Process functional requirements. (ii) DFARS Process architectural requirements. (iii) DFARS Process architectural constraints.** Each of the sub-processes are decomposed to a lower level granularity for easy application to design a DF application and are presented next.

2.1 DFARS Process for Functional Requirements

The DFARS process for functional requirements involves eliciting the activities to be accomplished by the DF application [12].
As depicted in Figure 2, the functional requirements of any DF application are: **(i) The end-users requirements. (ii) The application requirements. (iii) Other stakeholders requirements.** The DFARS Process for functional requirements determines the needs of a DF application user. These three sets of users needs represent different requirements for a DF application. As depicted in Figure 3, there are common needs essential to all identified users during a functional requirements elicitation. This point of convergence reveals

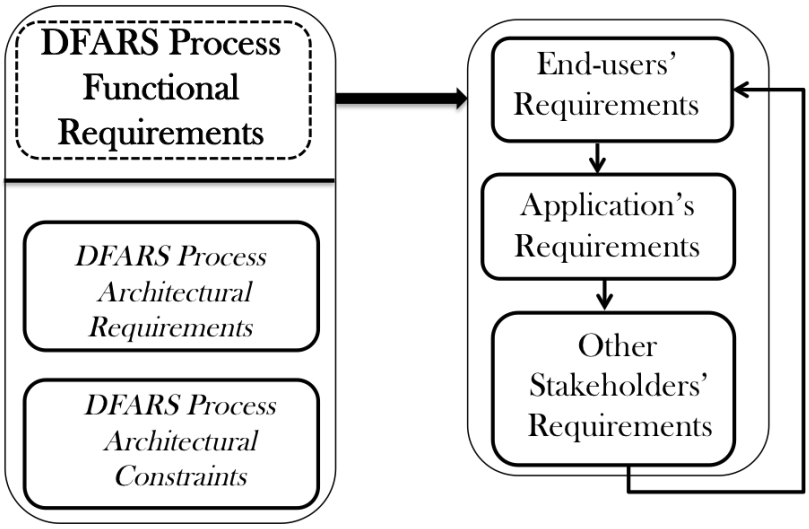

Figure 2: DFARS Process for Functional Requirements

the **core needs** of the DF application which must then be prioritised.

As shown in Figure 3, the core needs of a DF application must withstand all constraints and take precedence over other requirements when the DF application is being designed. An advantage of clearly specifying the end-users' requirements of a DF application is to present the DF application to a non-technical audience during its early stages of design.

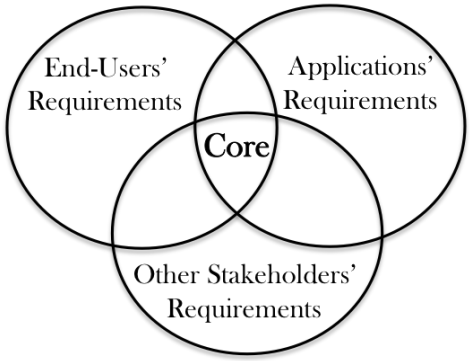

Figure 3: Typical Users of a DF Application

For example, a DF application that is to determine the number of messages (short message services (SMSs)) received by a mobile device since it became operational, should centre its core need on its auditability functions. Therefore, in designing such an application, the core design decisions must focus on the SMS messaging protocol and collaborate with the global system for mobile communications (GSM) network provider to achieve its auditability and interactive functions. Since the messaging protocol is one of the core requirements of the DF application, other identified constraints must be aligned with this DF application's primary need of auditing. To address the functional requirements of a DF application, the authors identified the DFARS process for functional requirements to be as follows:

(a) **End-users' Requirements:** Firstly, identify the needs of the application's end-users, since the end-users' needs hold a unique position in the design decisions of any DF application (see Figure 3). Using the functional requirements of the DFARS process, the end-users' requirements are interpreted by employing use case, activity or sequence diagrams[10]. These users' needs (which are depicted as Use-Case components) are constantly reviewed by all users of the DF application to keep up to date with the current trends in

digital forensics, as well as adhere to the core needs of the DF application.

(b) **The Application's Requirements:** The application's requirements are intertwined with those of the end-users to the extent, that the end-users' needs are interpreted and addressed. These needs are used to determine the best technology and techniques that addresses the need of the DF application and realises the DF application's requirements. The application requirements is interpreted by the developers which focuses on the application's needs. and base its specifications on the architectural and integration requirements to best address the overall goal of the DF application.

(c) **Other Stakeholders' Requirements:** These requirements are at issue when the DF application incorporates the needs of users other than the end-users during the design and development of a DF application. By incorporating the requirements of other stakeholders of a DF application, continuity and maintainability at the DF application design phase are ensured. The constraints of stakeholders range from technology preference to the identification of core quality requirements of the DF application. These stakeholders constraints when incorporated at the early stages of a DF application's functional requirements specifications ensure the efficiency and effectiveness of the DF application.

In summary, as shown in Figure 3, to elicit the functional requirements of a DF application, the end-users' requirements, the application's requirements and other stakeholders' requirements are considered respectively. The next section focuses on the specifications for architectural requirements, which is derived using the functional requirements specifications.

2.2 DFARS Process for Architectural Requirements

The DFARS process for identifying the architectural requirements involves assembling the information required to design the architecture of a DF application.

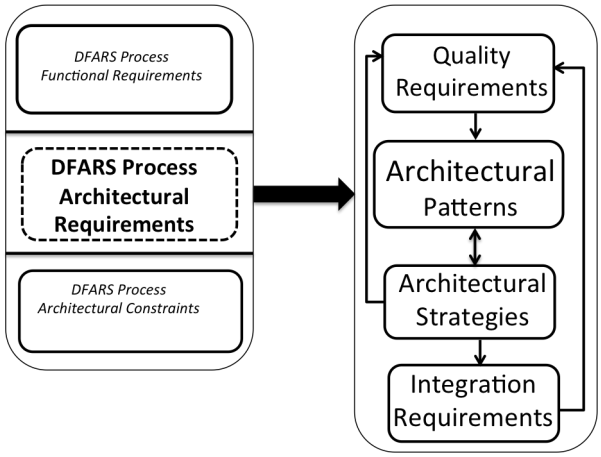

Figure 4: DFARS Process for Architectural Artefacts

As depicted in Figure 4, the architectural requirements of a DF application consist of the following: (i) Quality requirements. (ii) Architectural patterns (iii) Architectural strategies. (iv) Integration requirements of the DF application. The different processes employed to define the quality requirements of the DFARS architectural requirements are presented next.

2.2.1 The DFARS Process for Quality Requirements

The DFARS process for quality requirements are used to address the identified needs specified in the elicited functional requirements of any DF application. As shown in Figure 4, they are realised using architectural strategies and architectural patterns. The author proposed the following steps to derive the DFARS process for quality requirements:

(i) Aligning the identified functional requirements: To identify the quality requirements of a DF application, the users' requirements (i.e. the functional requirements) must be revisited. (For example, using the DFARS process to design a DF application that identifies the number of SMSs a mobile device has received/sent in the last few months - whether deleted or saved SMSs). The users requirements from the

DF application is to identify the number of SMSs sent/received, the details of the recipients of the SMSs, as well as the locations where the SMSs were received. Quality requirements to address these needs must be aligned to the DF application's other requirements.

(ii) The next step is to assign use case components to quality requirements: This is where the various identified user requirements that were aligned using use case diagram is attributed to one or more quality requirements. The quality requirements are then realised using architectural patterns. This is dealt with in the next section.

2.2.2 DFARS Process for Architectural Patterns

Architectural patterns addresses one or more quality requirements of a DF application. Choosing the best architectural patterns is therefore a critical aspect of a DF application's design and development. To determine the architectural patterns that best address the identified quality requirements of a DF application using the DFARS process, the authors proposed the following steps:

(i) Use an identified quality requirement to determine an architectural pattern: Decisions around the DF application's overall architectural responsibilities are made and addressed using the architectural patterns. For example, for an application with security as its priority, ensuring users' privacy and maintaining access control are the architectural responsibilities that must be addressed. Architectural patterns such as mode-view- controller (MVC), pipes-and-filters, and microkernel architectural patterns are then used to address the security requirements of the DF application.

(ii) Align architectural pattern to a quality requirement: The architectural patterns chosen must be identified using the critical quality requirements. Architectural strategies are employed to ensure that the various identified architectural patterns used to address the quality requirements of the DF application are concretely implemented.

2.2.3 DFARS Process for Architectural Strategies

Architectural strategies consist of individual approaches that specify the means to concretely address an aspect of an identified quality requirement of a DF application. Architectural strategies must be aligned with the architectural patterns to fulfill an aspect of a quality requirement. The authors identified the following steps to determine the architectural strategies of the DFARS process.

(i) Identify the architectural strategies to address the quality requirements. Choose the best architectural strategies that address each of the individual needs of the DF application as aligned with a quality requirement. For example, to address a DF application in which data originality and authenticity are major concerns, cryptographic hash and digital signatures are some of the architectural strategies to focus on. On the other hand, for the security requirements of an application that has the prevention of data breach as one of its main concerns, encryption and access control are some of the architectural strategies to utilise. Architectural strategies also involve making explicit trade-off decisions as to which feature of a quality requirement is more important to address, in order for a DF application to meet its stakeholder's requirements.

(ii) Select the architectural strategies in-order of importance. This order is firstly based on the priorities, i.e. the core needs of the DF application as demonstrated in Figure 3. For example, if privacy is the priority of the DF application, then encryption is considered ahead of other aspects of security requirements such as limited access or event logging. Secondly, select architectural strategies that can assist in addressing the identified architectural patterns. Two or more strategies can be used to address one aspect of a quality requirement. Next section gives the details of the DFARS process for integration requirements.

2.2.4 DFARS Process for Integration Requirements

Integration is the interaction of a DF application with humans, and other components of the DF application and the access channels of other systems. DF applications are often required to interact with internal and external resources thereby harnessing components such as WiFi connections, GSM networks and other connections required when conducting a DF investigation. The integration requirements of the DF application require the flexibility of the various components that can accommodate future upgrade and add-ons to a DF application during its lifespan.

In relation to the SMS example, the integration requirement relies on integrating with the GSM network grids to integrate with the internet protocol that enables users to receive/send SMSs independent of the GSM service provider's standard communications protocols. The DF application's design must also ensure that the auditability functions are reliable, easy to use, upgradable and integrated with other existing system components for the DF application's overall usefulness. On the other hand, integration requirements are the artefacts that ensure that the various components of a DF application can function as one entity and use communication channels effectively, without adversely affecting the overall operations of the application. Integration requirements clarify the needs of DF applications in order to achieve consensus on the priority of the DF application.
In designing a DF application, the integration of various channels using adapters, protocols and messaging channels is a key requirement. To determine the integration requirements of a DF application using the DFARS process, the following steps are considered.

(i) Define the integration scope: Firstly, clarify the scope of the DF application's integration requirements, so that the application's expected output is achieved and that the integration artefacts are specified. For example, designing a DF application to capture external network traffic of a personal computer (PC) should focus its design on two aspects: (a) The interaction and data-capturing process when the PC is connected to the internet, and (b) identifying the various possible protocols that could transmit or receive data from the PC.

(ii) Identify the interconnected activities to be performed by the DF application in order to clearly define each input and expected output in relation to other tasks and activities of the DF application.

(iii) Define both the machine and human actors of the DF application and their roles to clearly identify which integration requirements best address the functions of all actors. Identifying the integration requirements of any DF application paves the way for addressing the DF application's constraints, this is treated next.

2.3 DFARS Process for Architectural Constraints

The DF architectural constraints revolve around the other sub-processes of the DFARS process, namely the functional, architectural and integration requirements. An efficient DF application design must accommodate the constant changes in devices and maintain forensic soundness of the DF application, while adhering to the application's end-users requirements. Architectural constraints are the concerns of the stakeholders, which must be considered when the functional and architectural design decisions are made.
As shown in Figure 5, the common architectural constraints that must be taken into account during DF application's design are: (i) Jurisdictional (ii) Legislative (iii) Technological. The DFARS process for incorporating architectural constraints is as follows:

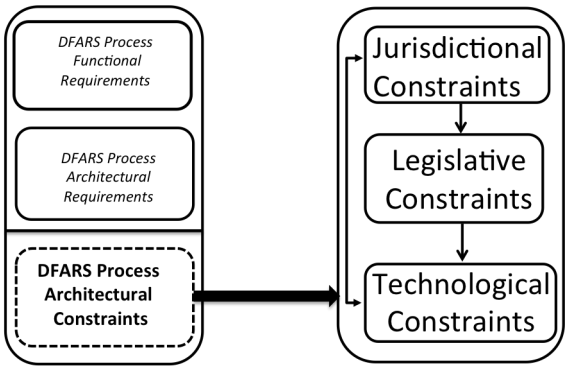

Figure 5: DFARS process to define Architectural Constraints

(i) Jurisdictional constraints: Identify the jurisdictional constraints of the DF application as they can influence the design and development of the DF application. The prescriptions, norms and laws of the jurisdiction where the DF application is to be used must be considered during its architectural decisions stages. For example, the data retention/ownership requirements of one domain may differ from the next. [5], [26].

(ii) Legislative constraints: The prescriptions of legislation and the Constitution of the countries where the application is developed and used must be adhered to. A DF application design must pay attention to legislation that governs digital evidence admissibility, for example the Protection of personal Information

(PoPI) Act of South Africa[5],[26].

(iii) Technological constraints: In terms of technology, the impact of the updating and upgraded of the features of mobile devices are also constraints that must be factored in while designing a DF application. Taking into account the architectural constraints of a DF application is the final part of the DFARS process.

Next sections focuses on using the proposed DFARS process to design the online neighbourhood watch (ONW) system as a case scenario.

3 Applying the Proposed DFARS Process to the ONW system

The ONW system is a digital forensic application that is used to crowd-source potential digital evidence (PDE), where community members generate PDE of neighbourhood crime. The PDE crowd sourcing is achieved by community members using their mobile devices to capture and upload evidence of crime in their neighbourhood. The PDE generated are available to the law enforcement agents (LEAs), the justice system and digital forensic investigators (DFIs) to be used in crime investigation[18],[21].Because of the involvement of these different stakeholders, the ONW system must show clearly, its design process that can be understood by technical and non technical audiences alike. Therefore, using the DFARS process, the design process of the ONW system is presented. Starting with the functional requirements specifications, followed by the architectural specifications, and finally the constraints of the ONW system are outlined.

3.1 Functional Requirements Specifications of the ONW System

In the proposed DFARS process for designing DF applications, functional requirements are identified firstly to address the users' needs of the DF application. As depicted in Figure 6, the main functional requirements components of the ONW system are as follows: (i) End-users' requirements. (ii) The application's requirements. (iii) Other stakeholders' requirements.

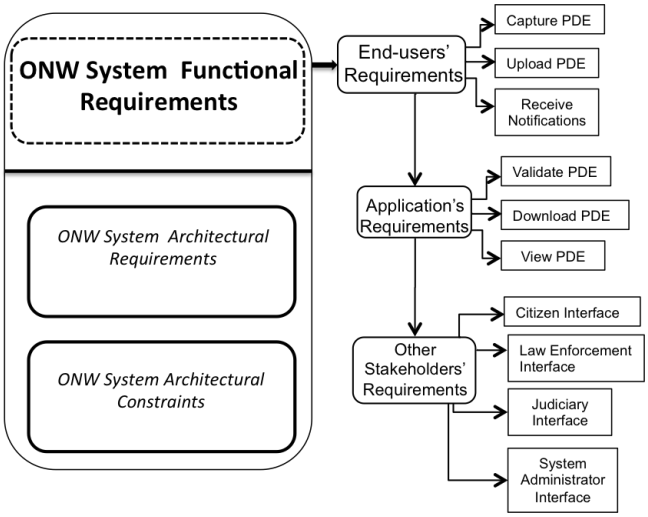

Figure 6: Applying the DFARS Process to designing the ONW System's Functional Requirements Specifications

3.1.1 End-users Requirements

As depicted in Figure 6, the end-user's requirements are as follows: (i) Capture PDE (ii) Upload PDE (iii) Receive notifications. As depicted in Figure 7, these identified needs of the ONW system functional requirements are represented as components in a use case diagram.

(a) Capture PDE

This is the first point of contact with the ONW system which is the capture of PDE of crime using mobile devices.

The captured data can be in the form of audio recording, a photo, or a video of what the citizen perceives to be a crime. At this point the captured digital data is potential digital evidence (PDE). In a previous research [19],[20], the detailed processes employed to ensure forensic soundness of the crowd-sourced PDE captured using the ONW system was dealt with. It will be up to the judiciary and law enforcement agents to determine if the PDE is actual evidence that is admissible in court.

(b) Upload PDE
This use case component depicts the function/task of the citizen having captured PDE of a crime. The citizen uploads the captured PDE to the ONW system's repository. This option to upload can be performed immediately or it may be done later if the citizen feels in danger and needs to leave the scene of the crime, or if there is not a good data connection to upload the captured PDE.

(c) Receive Acknowledgment
The ONW system communicates with the citizen at various times throughout the PDE lifecycle. The citizen receives the first acknowledgment notifications at a successful upload of the PDE. The second acknowledgment is received when the LEA or DFI downloads the PDE uploaded by the citizen. Another acknowledgment is received by the citizen when PDE is used as evidence in court. Receiving acknowledgments is a feature of the ONW system that encourages and motivates the citizen to participate and stay involved in the community policing process. Moveover, citizens are motivated to participate when they realise the difference their generated PDE makes in crime solving.

3.1.2 Application's Requirements

As depicted in Figure 6 the needs that must be met by the DF application to ensure the DF application manages the application-side of the system effectively is as follows:

(a) Validate PDE
This is an application requirement function carried out by the law enforcement agent or the digital forensic investigator, it is to ensure that the PDE is valid by checking the forensic soundness indicators. If the PDE has been tampered with or is found to be invalid, the evidence is discarded.

(b) Download PDE
Once the LEA or DFI is satisfied that the PDE is still forensically sound, they proceed to download the PDE. The downloaded PDE is analysed in an investigative process, as well as made available to the members of the judiciary and/or presented as evidence in a court.

(c) View PDE
This function is carried out by the members of the judiciary, who need to view PDE as it relates to a case that they are dealing with in court. The members of the judiciary are not entitled to download or have direct access to the ONW repository. But by viewing the PDE, the presiding judge can assess the relevance of the PDE and determine its evidential weight, while the prosecuting and defense counsels can also employ the PDE for arguments during legal proceedings.

3.1.3 Other Stakeholders Requirements

As depicted in Figure 6 the needs of the ONW system that cover the interests of other stakeholders of the system are as follows:

(a) Case Management
The ONW system is designed to allow every PDE to be associated with a crime case, and all PDE is labelled with a description of what type of crime it captured. This provides an easy way to categorise PDE that is from the same crime scene. The case management function allows the system administrator or LEA to create, delete and assign PDE to cases. The creator of a case is by default assigned rights to view the case, and other authorised users can be assigned to the case, which gives them the rights to view the cases. The authorised users include the law enforcement agents (LEA) or the digital forensic investigator (DFI) and the judiciary.

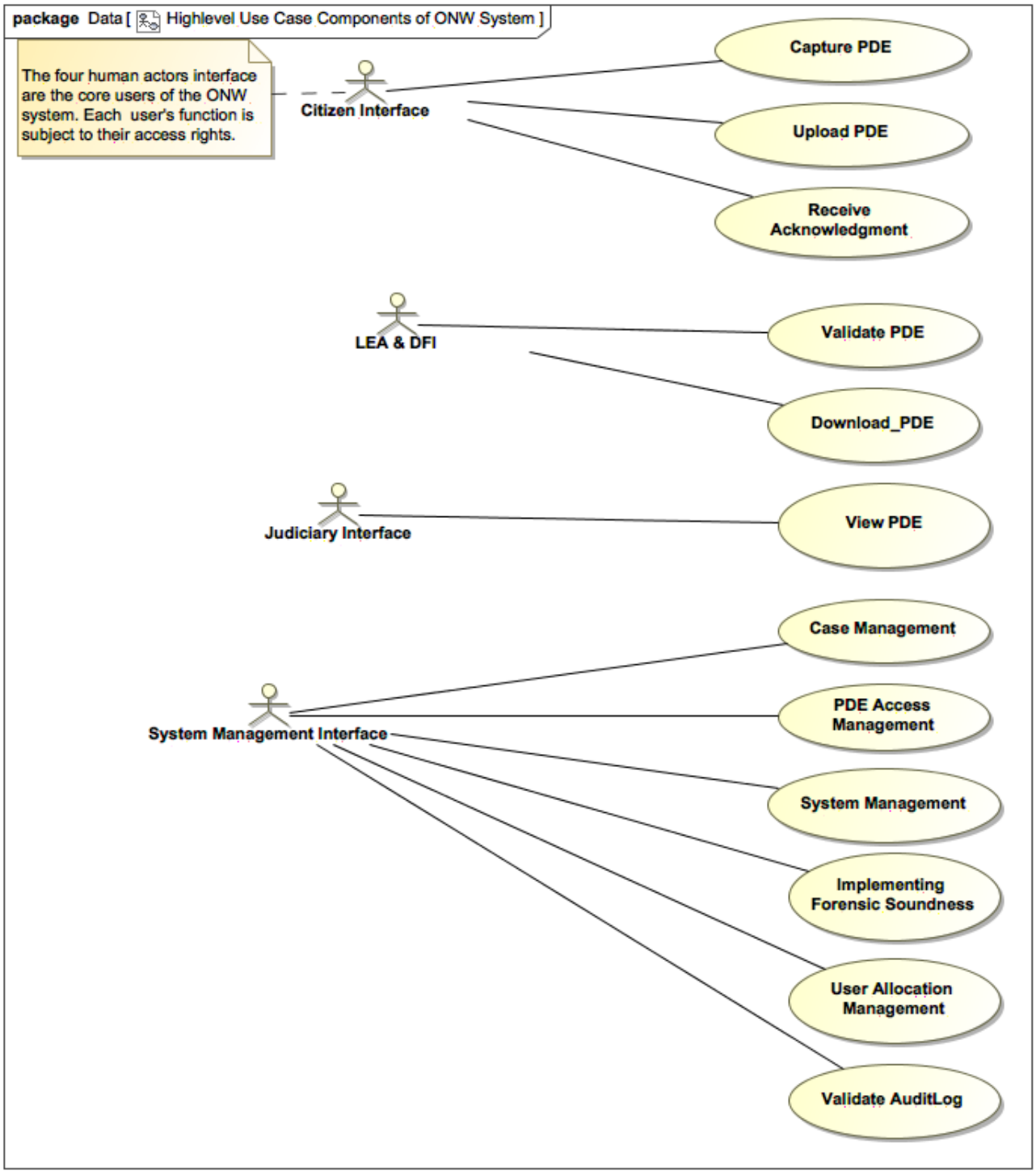

Figure 7: High-Level Use Case Components of the ONW System Functional Requirements

(b) PDE Access Management
The system administrator manages who accesses the stored PDE in the ONW repository. The system administrator also creates cases, assigns cases to the authorised users, allocates or removes a user from a case and downloads PDE. At authentication, a case is selected from the ONW repository, whereby its contents can be viewed and/or downloaded by an authorised user. The access function is implemented using the attribute-based access control (ABAC) policies. The PDE access management function ensures that PDE is not tampered with while stored in the ONW repository, thereby maintaining the forensic soundness of PDE.

(c) System Management
The add/remove user use case allows users such as the LEA, Judiciary and the DFI to use the ONW system. The ONW system administrator grants access to the authorised users. This ensures that only authorised users are allowed to log in to the system. At first login, authorised users are required to change their password that was generated by the system administrator during registration.

(d) Implementing Forensic Soundness
Forensic soundness is the requirement that verifies that DF applications retain PDE evidentiary weight as stipulated by law, from the time of PDE capture to the time of its use in criminal, civil or organisational enquiries[21]. The process employed to ensure the forensic soundness of PDE captured and stored within the ONW system constitutes one of the most critical aspects of the ONW system.

(e) User Allocation Management
The user allocation management is the use case that manages both actors in the system as well as the case allocation functions. The add/remove function governs who can be registered as users and determine whether they need to be users at any point in time.

(f) Validate Audit Log
The audit log keeps track of all actions performed by users who access the ONW system. In crime investigation the collection and preservation of evidence are crucial steps because if they have been carried out incorrectly, the PDE may be inadmissible in any court of law. The audit log serves as a proof of confidentiality and originality of the PDE of the crime being viewed or investigated. Employing the DFARS process, the functional requirements specifications of the ONW system, shows the stepwise process, to technical or non technical stakeholders. Elicitation of the DF application functional requirements is followed by the DFARS architectural specifications.

3.2 Architectural Requirements Specifications of the ONW System

As depicted in Figure 8, this section models the DFARS process to identify the architectural requirements of the ONW system. The functional requirements as identified in the previous step are used to determine the architectural requirements of the ONW system. Also based on the functional requirements specifications of the ONW system, the quality requirements, the architectural patterns, strategies, and the integration requirements as shown in Figure 8 are identified.

3.2.1 Quality Requirements of the ONW System

The quality requirements of the ONW system, which are also identified based on the functional requirements, are follows: (a) Security requirements.(b) Reliability requirements. (c) Usability requirements. (d) Auditability requirements.

(a) Security Requirements of the ONW System
Security is the ability of the ONW system to remain dependable in the face of mischance, error, or faults [2,28]. The security requirements of the ONW system are achieved using the information security services of confidentiality, integrity, authentication, authorisation and non-repudiation (CIAAN). How these are employed in the ONW system is described in more detail below:

(i) Confidentiality - To maintain the confidentiality of PDE stored in the ONW repository, an advanced encryption standard (AES) 256 bits symmetric encryption algorithm is used to ensure that stored PDE remains confidential even in the unlikely event of data breach. The AES 256 bits encryption is also

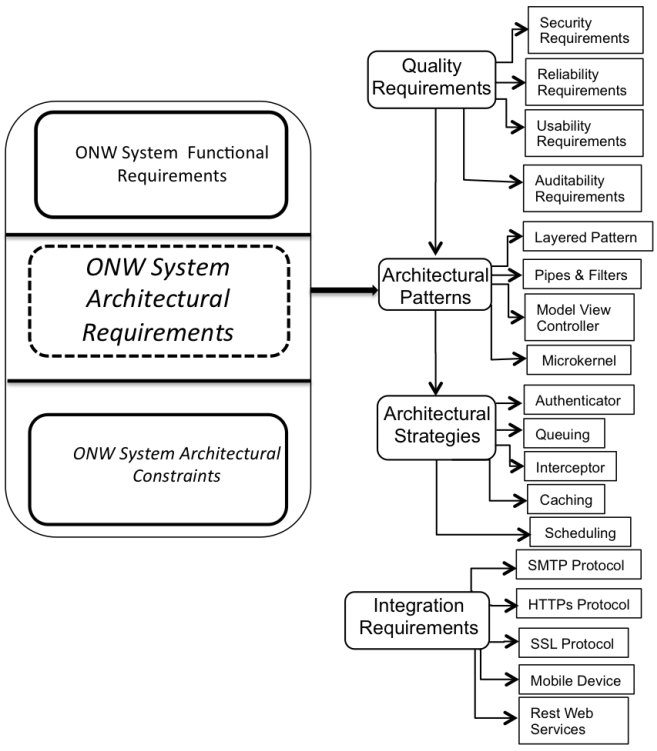

Figure 8: Applying the DFARS Process to designing the ONW System's Architectural Requirements

used to encrypt the citizen's (uploader's) information and his/her device's metadata. Confidentiality specifications of the information security services are used to protect the validity of PDE in order to ensure its admissibility in any legal proceeding. The ONW system is designed to be accessed via https for a secure encryption communication link between the front-end and the back-end. This also ensures that PDE is not altered in transit or at storage between the uploader and the downloader.

(ii) Integrity - The ONW system is so designed that an unauthorised user should not be able to access it. The integrity function is implemented by using a cryptographic hash function where captured PDE is hashed using the SHA512. The hashed value is then encrypted and stored in the database relative to the PDE. Other integrity measures adopted by the ONW system are the use of geographical location, timestamp and other metadata.

(iii) Authorisation - The authorisation process employed for the ONW system involves granting access rights of PDE resources to the LEA/DFI and other authorised users only. The ONW system implements its authorisation functions using the combined policies of role-based access control (RBAC) and attribute-based access control (ABAC). In addition, the authorisation enforcer architectural pattern handles the authentication logic actions across the front-end.

(iv) Authentication - The ONW system uses a session authenticator, username and password for authentication. The user's associated information is important as it is required for the authentication process. A user provides a username and password in order to be logged in and a session authentication is then created. Session authentication is applicable to all users of the ONW system. The login credentials are authenticated against the user's details stored in the ONW system.

(v) Non-repudiation - The ONW system achieves non-repudiation by implementing a digital signature to identify who captured, upload or download PDE at any given point of the system's life cycle. Using non-repudiation, PDE uploader's original intention is upheld. For example, an uploader of PDE data may not at a later time deny his/her intentions in the creation or transmission of an alleged PDE.

The architectural patterns and strategies used to address the security requirements of the ONW system are microkernel, MVC, pipes-and-filters. They are placed on a layered architectural pattern which is at the first level of granularity of the ONW system structure. The security of the ONW system is achieved by using these architectural patterns in conjunction with the architectural strategies (i.e. cryptographic hash, digital signature,

encryption, decryption, port lockdown and limited access) discussed in the CIAAN mechanisms above.

(b) Reliability Requirements of the ONW System
In designing the ONW system, the microkernel architectural pattern is used to ensure reliability of the ONW system. This is achieved using the microkernel architecture bus, thereby maintaining reliable communication channels from all access points to various components of the system. The microkernel architecture bus enables the ONW system to detect any form of PDE alteration at upload or download and maintains a tamper-proof system throughout the ONW system's transactions cycle. Likewise, the pipes-and-filters architectural pattern encapsulates the encryption and decryption features of the ONW system, and at the same time manages attacks such as eavesdropping and data interception.

(c) Usability Requirements of the ONW System
Usability functions that enhance users' experience, such as remembering users' tasks, the last-typed URL and recalling usernames and passwords (especially when using the same device), are features embedded in the ONW system during the design phase. The mode-view-controller (MVC) architectural pattern is used to provide a flexible user interface that allows for a component-oriented design and separation of concerns, thereby restricting dependence and avoiding the re-designing of components when the user interface changes.

(d) Auditability/Monitorability Requirements of the ONW System
The auditability functions of the ONW system provide the functional service that enables the ONW system to log the input and output activity of the internal and external resources of the system in order track users and system activities. The auditability functions ensures that the ONW system manages a crash by activating the roll-back function, thereby returning the system to its last stable state. Date and timestamps are additional methods of auditability implemented in the ONW system.

(e) Integration Requirements of the ONW System
The integration requirements of the ONW system ensure the cohesive collaboration of all aspects of the ONW system's components. The integration requirements ensure that the external components such as the simple mail transfer protocol (SMTP) for email communication which enables users of the ONW system to receive email communications for user registration, to receive notifications, password verification or recovery and updates.

3.3 Architectural Constraints of the ONW System

The DFARS process takes cognisance of the architectural constraints that must be dealt with in designing any DF application. Therefore in the design of the ONW system design three sets of constraints (as shown in Figure 9) are incorporated. These are (i) Jurisdictional constraints (ii) Technological constraints (iii) Legislative constraints.

(a)Jurisdictional Constraints of the ONW system
Since the ONW system is focused on crowd-sourcing PDE of crimes within neigbourhoods in South Africa, the South Africa jurisdictional requirements that abides by the Constitution, norms and traditions of the South Africa community must be adhered to. For example, the PDE storage system makes provision for multi-tenancy requirements, thereby rendering PDE captured using the ONW system valid and admissible in a South African court and in other country with 'mixed' legal system blending in the Roman-Dutch law [29]. The jurisdictional constraints are embedded to the ONW system's using the MVC architecture pattern, where the controller of the ONW system's karnal enhance the auto locations check at the upload of PDE to the ONW system's repository. Putting this function at the architectural design stage ensure that the importance of this aspect of the ONW system's components is maintained continuously through the ONW system's life cycle.

(b) Technological Constraints of the ONW system
Using the DFARS process, technological constraints are also incorporated at the ONW system's at design stages. Technology requirements such as supports for standard HTTPs protocol based on Firefox, Safari, Chrome and other popular web browsers is inserted at the architectural level. Prescriptions such as database type, object relational mappers (ORM) and framework are decided and abide by as some of the technological constraints of the ONW system.

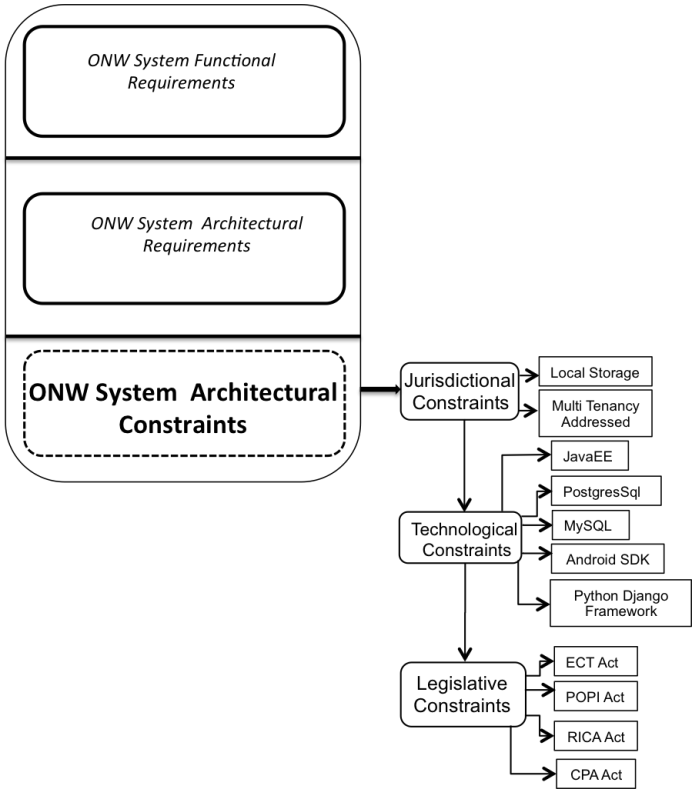

Figure 9: Applying the DFARS Process for ONW Architectural Constraints

(c) Legislative Constraints of the ONW System
Another constraints that was dealt with is the legislative constraints. For citizens to participate in the PDE sourcing process, their privacy rights must be indemnified at the architectural specifications phases of the ONW system. For example, encryption of citizen's personal information generated during the user registration process. This process is ensured at the architectural specifications level, relaying on microkernel, pipes and filters architectural patterns to achieve the privacy challenges.

Incorporating the architectural specifications designs into the ONW system design is the foundation that ensures the system's accountability and auditability during the system's development stages and therefore provide easy re-traceability of any input or expected output of the ONW system. Next the evaluation of this paper beginning with related literature.

4 Related Works

Garfinkel et al.[6] are concerned about the state of digital forensics and argue that digital forensic practitioners need tools that do more than search and prevent, in other words tools for reconstruction, analysis, clustering and data mining. These tools must be tested in order to generate accurate results, which are surprisingly low in digital forensics. They highlight the need for reproducible techniques and results, since digital forensics is a science. The need to employ a standardised scientific process in designing a digital forensic application as proposed by Garfinkel and others has provided the motivation for this paper. The DFARS process is a partial response to the need identified by Garfinkel et al.[6] to develop statistical and other approximation techniques to ensure that the interpretation of digital evidence is grounded in facts and science and not simply upon opinion.

Herlea et al.[9] deal with the integration of behavioural requirements specifications within compositional knowledge engineering. They define requirement engineering as a process that addresses the development and validation of methods for eliciting, representing, analysing and confirming system requirements. Their paper argues

that requirements engineering is a consultative process that must take place over time and involve all stakeholders. This process combines the discussion of requirements and scenarios, which are continually honed over time to arrive at a system description that satisfies all stakeholders. Herlea et al.[9] emphasise the involvement of the users, who are considered the "experts" at the critical early stage of software development. However, not only are the users important participants in the consultation, but all stakeholders - who include domain experts, system customers, managers, and developers - are important participants in the consultation process. The contributions that Herlea and others made to the requirements engineering literature are significant for the fact that they highlight the need to involve the users in a consultative process.

Cohen et al.[4] presented the Advanced Forensic Format (AFF), which is a new specification and toolset. They redesigned the architecture of the AFF to make it the basis for a globally distributed evidence management system. The new architecture is capable of storing multiple heterogeneous data types that could arise in modern digital investigations, including data from multiple storage devices, new data types and extracted logical evidence. The ONW system, on the other hand, is designed using the DFARS process, which incorporates software engineering methods to address the need for DF applications to be easily interpreted by any non-technical user, as well as to easily accommodate the upgrade of devices susceptible to DF investigation.

Hedstrom[8] explores the concept of digital libraries with regard to preserving digital material due to technology change. According to her research, technology changes in cycles of three to five years, therefore artefacts used in the preservation of digital data may soon be out of date. The technique becomes outdated due to obsolete preservation methods and exposes digital material to falling prey to attacks due to technological obsolescence. The ONW system took cognisance of the research of Hedstrom[8] when it implemented a component-based design for the uWatch application and the NW system. The component based design of the ONW system caters for technology change, so that if any component becomes obsolete, that component can be re-developed without necessarily affecting the entire state of the ONW system. The ONW system has been developed as a flexible component-based system that uses layered, microkernel and pipes-and-filters architectural patterns to enable pluggability, upgrading or changing of one aspect of the system without affecting the entire system's functionality.

This paper takes cognisance of the need to consult users in creating DFARS process for designing DF software tools or applications. For example, the DFARS process, in determining the functional requirements, takes into account the needs of the DF applications users, the stakeholders and other stakeholders in this regard. A comprehensive survey of requirement engineering specifications literature or digital forensic application designs lies beyond the scope of this paper, but this section has examined papers that are focused on the importance of requirements engineering in digital forensics. Garfinkel et al.[6] call for definitive standards for designing DF software applications, while Herlea et al.[9] highlight the need for a consultative approach in the process of designing DF applications. The DFARS process implements both of these elements and more, as it not only provides a definitive process as an approach for designing DF software tools or applications, but also presents a case study in which the DFARS process was actually employed to design a DF software application - the ONW system.

5 Discussion

Digital evidence presentation in any court of law can be brought under scrutiny, and part of this scrutiny is challenging the design processes of the digital forensic applications used to extract the evidence. This has brought about the need for a transparent design process for DF applications.

The main objective of this paper was to address concerns that could arise when digital forensics evidence is presented in court. To address these concerns, a digital forensic application design process was proposed. The proposed process is the digital forensic application requirements specifications (DFARS) process, used specifically to design DF applications or tools. The DFARS process defines DF application's depth and width thereby overcoming the two major challenges when working with DF applications. These challenges are (i) the need to accommodate the constant upgrade in devices susceptible to digital forensic investigation and (ii) To provide a means to easily demonstrate to both technical and non-technical audiences the processes employed at the design and development of a DF application used during the analysis, investigation and presentation of criminal

evidence. These stakeholders and users of DF applications who need to understand the DF application design processes include, but not limited to, law enforcements agents, the judiciary, and the victims and perpetrators of the said crime.

Another contribution of this paper was to show how the DFARS process can be employed when designing a DF application. The ONW system was used to show this application of the DFARS process. The online neighbourhood watch (ONW) system is a crowd-sourcing DF application that generates potential digital forensic evidence. The PDE generated assists the law enforcement agents in the generation, storage and usage of evidence in crime [19]. The DFARS process employed to design the ONW system began by identifying the functional requirements of the ONW system, i.e. it focused on the ONW application's and users' core requirements. The identified core requirements of the ONW system were subsequently mapped to identify the quality requirements of the ONW system and these were in turn addressed using architectural patterns, strategies and integration requirements that adhered to the ONW system's identified architectural constraints. The DFARS process also added a layer of abstraction at the design of a DF application to accommodate device upgrades. Continuous upgrades in devices, especially mobile devices, are a result of the frequent changes in technology and sometimes in legal requirements, as it concerns electronic evidence [11]. The DFARS process further identified the architectural constraints of the ONW system, which had been incorporated during the design stages. These technology-neutral processes were combined to determine the best technologies that would address the needs of the ONW system at its development.

In evaluating the DFARS process therefore, the factors that are considered include the impact of the DFARS process on designing the ONW system and the benefits of using it to design any other DF application. Employing the DFARS process focuses on ensuring a component-based design approach where every aspect of the ONW system application is maintained as an entity. The DFARS process harnessed advantages such as:

(i) Making provision for an accessible DF application design process, thereby enabling an easy evaluation of the DF software application - by peers and other stakeholders alike - to ascertain the forensic soundness of the DF application design process. The usability and delivering of the end-users' expected output in a simple and easy-to-understand DF application is therefore, a core requirement when eliciting the end-users' needs for a DF application employed in a digital crime investigation.

(ii) Exposing a DF application's core design process and eliminating technical details that would otherwise make the design and development process not easily followed by stakeholders, such as the legal team and non-technical stakeholders, is achieved using the DFARS process. This exposure is essential, especially when a DF application that was used in a crime investigation is brought under scrutiny. Furthermore, the DFARS process facilitates design decisions that align the application's constraints to the core needs of stakeholders.

(iii) Another finding of the DFARS process is the designing of the ONW system as an effective and extensible system, that ensures measurable quality requirements with respect to expected output. Such measurable quality requirements are defined by concretely quantifying the capabilities of the ONW system using use case components to align the application to its quality requirements.

(iv) The drawback of the DFARS process on the other hand time constraint. This is because detailed requirements elicitations that could absorb the various aspects of the ONW system or any other DF application components require longer project management time [1]. However, at the first detailed requirements specifications of the DF application, a subsequent upgrade or update is added as a service [22].

6 Conclusion

In the field of digital forensics, tools and applications are used to unravel the cause of incidents, especially during digital evidence acquisition, analysis and examination. This is especially important for evidence gleaned from devices that are inherently susceptible to digital crime and investigations.

In conclusion, this paper has proposed an easy-to-apply stepwise process for designing DF applications, thereby enabling all stakeholders to easily understand especially non-technical audiences. The DFARS process easily addresses the constant changes to and upgrades of digital devices. The DFARS process also shows a process that allows for incremental pluggability and modifiability functions of all aspects of the ONW system, thereby providing a detachable design that ensures continuous advancement of the functions of DF applications by

using a component-by-component design method. Changes that are inherent in digital forensic investigation practices in line with legal standards and current trends in information security are easily accommodated using the DFARS process.

References

[1] Shaun Abrahamson, David Wallace, Nicola Senin, and Peter Sferro. Integrated design in a service marketplace. *Computer-Aided Design*, 32(2):97–107, 2000.

[2] Ross Anderson. *Security engineering*. John Wiley & Sons, 2008.

[3] George M Brilis, Jeffrey C Worthington, and A Dallas Wait. Quality science in the courtroom: Us epa data quality and peer review policies and procedures compared to the daubert factors. *Environmental Forensics*, 1(4):197–203, 2000.

[4] Fred .A Cohen. *Digital Forensic Evidence Examination*. Fred Cohen and Associates out of Livermore, third edition, 2009.

[5] Government-Gazette ECT-Act. Electronic Communications and Transactions Act, Act 25 of 2002. Technical report, PDF Scanned by Sabinet [Online - Accessed 08 February, 2014], August 2002. South Africa Government Gazette - Legislation - South Africa - National/Acts and Regulations/E/Electronic Communications And Transactions Act No. 25 Of 2002/The Act.

[6] Simson Garfinkel, Paul Farrell, Vassil Roussev, and George Dinolt. Bringing science to digital forensics with standardized forensic corpora. *digital investigation*, 6:S2–S11, 2009.

[7] CP Grobler, CP Louwrens, and Sebastiaan H von Solms. A framework to guide the implementation of proactive digital forensics in organisations. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, pages 677–682. IEEE, 2010.

[8] Margaret Hedstrom. Digital preservation: a time bomb for digital libraries. *Computers and the Humanities*, 31(3):189–202, 1997.

[9] Daniela E Herlea, Catholijn M Jonker, Jan Treur, and Niek JE Wijngaards. Integration of behavioural requirements specification within knowledge engineering. In *Knowledge Acquisition, Modeling and Management*, pages 173–190. Springer, 1999.

[10] Ivar Jacobson, Grady Booch, James Rumbaugh, James Rumbaugh, and Grady Booch. *The unified software development process*, volume 1. Addison-Wesley Reading, 1999.

[11] Orin S Kerr. Digital evidence and the new criminal procedure. In *Columbia Law Review*, volume 123, pages 279–318. JSTOR, 2005.

[12] Andrew J Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, et al. The state of the art in end-user software engineering. *ACM Computing Surveys (CSUR)*, 43(3):21, 2011.

[13] David Kung. *Object-oriented Software Engineering: An Agile Unified Methodology*. McGraw-Hill Higher Education, 2013.

[14] Christopher Lee. Archival application of digital forensics methods for authenticity, description and access provision. *Comma*, 2012(2):133–140, 2012.

[15] Dean Leffingwell and Don Widrig. *Managing software requirements: a use case approach*. 2003. ISBN-13: 978-0321122476, ISBN-10: 032112247X.

[16] 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. ISBN-13: 000-0321815734 ISBN-10: 0321815734 3rd Edition.

[17] JD Meier, David Hill, Alex Homer, Taylor Jason, Prashant Bansode, Lonnie Wall, Rob Boucher Jr, and Akshay Bogawat. Microsoft application architecture guide. *Microsoft Corporation*, 2009.

[18] Stacey Omeleze and Hein S Venter. Towards a model for acquiring digital evidence using mobile devices. In *INC*, pages 173–186, 2014.

[19] Stacey Omeleze and Hein. S Venter. Towards a model for acquiring digital evidence using mobile devices. In *Tenth International Network Conference (INC 2014) and WDFIA 2014 Plymouth University, UK*, pages 1–14. Plymouth University, UK, 2014.

[20] Stacey Omeleze and Hein S Venter. Proof of Concept of the Online Neighbourhood Watch System. In *International Conference on e-Infrastructure and e-Services for Developing Countries*, pages 78–93. Springer, 2015.

[21] Stacey Omeleze and S. Hein Venter. A model for access management of potential digital evidence. In *10th International Conference on Cyber Warfare & Security (ICCWS)*, pages 491–501. CSIR, University of Vender and Academic Conferences Limited, 2015.

[22] Michael P Papazoglou, Paolo Traverso, Schahram Dustdar, and Frank Leymann. Service-oriented computing: State of the art and research challenges. *Computer*, 40(11), 2007.

[23] Klaus Pohl. *Requirements engineering: fundamentals, principles, and techniques.* Springer Publishing Company Incorporated, 2010. ISBN:3642125778 9783642125775.

[24] CERT Program. CERT Square Researchcarnegie mellon university, 2016.

[25] James Robertson. Forensic science, an enabler or dis-enabler for criminal investigation? *Australian Journal of Forensic Sciences*, 44(1):83–91, 2012.

[26] Pamela-Jane Schwikkard and Steph E Van der Merwe. *Principles of evidence.* Juta and Company Ltd, 2009. ISBN: 978 0 7021 79501.

[27] Fritz Solms. What is software architecture? In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, pages 363–373. ACM, 2012.

[28] Heru Susanto, Mohammad Nabil Almunawar, and Yong Chee Tuan. Information security management system standards: A comparative study of the big five. 2011.

[29] Ph J Thomas, Cornie G Van der Merwe, and Ben C Stoop. *Historical foundations of South African private law.* Butterworth-Heinemann, 1998.

[30] Aleksandar Valjarevic and Hein S Venter. Harmonised digital forensic investigation process model. In *ISSA*, pages 1–10. IEEE, 2012.

[31] Aleksandar Valjarevic and Hein S Venter. Introduction of concurrent processes into the digital forensic investigation process. *Australian Journal of Forensic Sciences*, 48(3):339–357, 2016.

[32] Murdoch Watney. Admissibility of electronic evidence in criminal proceedings an outline of the south african legal position. *Journal of Information*, 2009.