FOR GOD AND MY COUNTRY

THE REPUBLIC OF UGANDA

GoU eGovernment Web Application Security Architecture Framework

Project: THE REPUBLIC OF UGANDA CONSULTANCY SERVICES FOR THE DEVELOPMENT OF A GOVERNMENT ENTERPRISE ARCHITECTURE (GEA) AND EGOVERNMENT INTEROPERABILITY FRAMEWORK (E-GIF)

Project duration: 10 November 2020 – 09 September 2021

31 August 2021

Table of Contents

1 Executive Summary

The Government of Uganda (GoU) is transforming their public services to meet the modern digital era. While introducing new ICT systems and e-services is guided by the eGovernment Interoperability Framework (e-GIF) in the context of the eGovernment Enterprise Architecture (GEA), one specific domain – the security of web applications – could render the transformation a success story or the opposite.

The world's most valuable resource is not oil or gold but data. Information systems process data in its digital form, enabling the internet we know today. eGovernment ICT systems and e-services are no different. Web applications are the primary solutions behind the more significant portion of modern technology, intended to provide e-services to users over the internet. The user role may be carried out by a human interacting with a graphical interface or another system automatically querying the application over an API (Application Programming Interface). To preserve the confidentiality, integrity, and availability of the data, securing the web application is crucial.

Each organisation delivers business services and has business goals which are the principal reason for the organisation's existence. Next, security objectives and requirements should be derived from these goals, and finally, respective controls should be applied. This kind of a logical chain appoints and justifies security activities and resources. Combining two well-respected security methodologies, SABSA and OWASP ASVS, and customising to meet the needs of GoU, results in the Web Application Security Architecture Framework (WASA Framework).

The WASA Framework is intended as an easy-reading guide and a practical Toolkit for GoU's public sector organisations who seek support for protecting their web application. Starting from understanding the business requirements, the WASA Framework guides through different architectural layers down to granular, specific, and technical requirements of the web application. As the lifecycle of an information system doesn't end with the deployment, additional attention is invested in the management architecture for daily operations and maintenance of the application.

To arrange the protection of GoU e-services, all MDAs must design, develop, and maintain their web applications, keeping security strongly in the focus all the time. The WASA Framework is a solid support on that journey.

2 Introduction

2.1 WASA Purpose and Scope

Web Application Security Architecture (WASA) Framework is an approach to protect the information processed by Uganda eGovernment e-services by securing the underlying web applications. Designing security deeply into the technical solutions, the WASA helps to counter fight cyber threats such as system compromise and data leakage.

The amount of professional personnel dealing with security and software architecture topics in public sector organisations is not comparable to large and mature enterprises with abundant resources. Therefore, one goal of the WASA Framework is to derive from business attributes (requirements) technical security requirements of web applications while keeping the methodological overhead as slim as possible.

The WASA Framework supports at least two scenarios:

- Guidance for developing new secure web applications applying security-by-design principle from scratch
- Guidance on validating security controls of existing web applications (i.e., security assessment/auditing, security/penetration testing).

The WASA Framework contains:

- Present guide document explaining the methodological aspects and providing guidance for the use of the WASA Toolkit (.pdf)
- WASA Toolkit (.xlsx) supporting the organisation moving through different SABSA architectural layers and providing a security requirement checklist for validating a web application.

The primary target group of the WASA Framework are Uganda public sector information security, software development and IT personnel and their partners with the following roles, but not limited to:

- Security Architect
- Software Architect
- Software Developer/Engineer
- System/Web Application Administrator
- Information Security Specialist/Engineer/Analyst
- Information Security Manager/Officer.

2.2 Approach

The WASA Framework stands on two pillars – SABSA (Sherwood Applied Business Security Architecture)[1] information security architecture methodology and OWASP ASVS (Application Security Verification Standard)[2]. Both are well-known and widely accepted open standards. While SABSA is missing technical and detailed specifics of web applications, then ASVS doesn't incorporate enterprise top-down security architecture principles. The WASA Framework acts as an interface and links SABSA with ASVS on an abstraction level where business attributes (requirements) of web applications are transformed to technical security requirements (chain of traceability). With some reservations, ASVS

[1] The SABSA Whitepaper. https://sabsa.org/sabsa-white-paper-download-request

[2] OWASP Application Security Verification Standard. https://owasp.org/www-project-application-security-verification-standard/

security requirements could also be seen as security controls that describe the activities or resources needed to protect valuable assets.

In the context of WASA, briefly explaining the terms *web application*, *web service,* and *web API* help to unify the background knowledge:

- Web service - a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Web service is defined by W3C [3].
- Web API (Web Application Programming Interface) – a set of functions and procedures that can be used to program application that interacts with other application. API is typically done in HTTP/REST architectural style. Output could be JSON/XML, input XML/JSON/or plain data. Web API is not officially defined.
- Web Application – an application software that runs on a web server and is accessed by the user through a web browser or by other applications over APIs or web services.

Regarding the WASA Framework, the term *web application* also covers the *web API* and *web service*.

Information technology and security principles, techniques and postures change constantly. While SABSA has remained rather a static methodology, then ASVS evolves continuously, and minor or major version updates are released periodically. That leads to the fact that WASA Framework's guide document has a prolonged life expectancy, but the Toolkit's ASVS security requirements checklist (the "Security Requirements" worksheet) needs occasional renewing. However, accepting some missing updates or text modification of the ASVS, the update of WASA Toolkit after every second year could still outcome a fair set of security requirements for robust web applications.

2.3 WASA Toolkit

Information security, software development and IT administration practitioners often lack time to analyse various somewhat theoretical methodologies and build practical tools on top of it. The WASA Toolkit intends to provide a step-by-step top-down walkthrough approach moving through SABSA's layered architecture, from contextual architecture down to the management architecture.

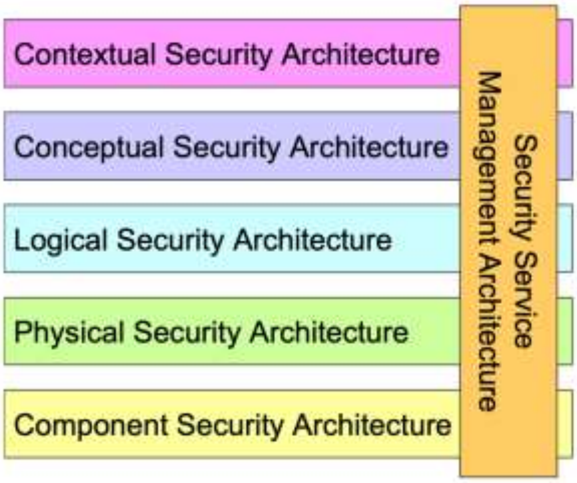
Contextual Security Architecture
Conceptual Security Architecture
Logical Security Architecture
Physical Security Architecture
Component Security Architecture
Security Service Management Architecture

Figure 1. The SABSA layered model for security architecture [4]

[3] Web Services Architecture. https://www.w3.org/TR/ws-arch/

[4] *SABSA White Paper.* Enterprise Security Architecture (John Sherwood, Andrew Clark and David Lynas. 1995 – 2009 SABSA Limited

The Toolkit consists of the following steps (worksheets):

0. **Introduction**: an overview of the Toolkit and versions
1. **Context**: a survey to understand the organisation's business and context of the web application. The Toolkit user should provide the answers to What? Why? How? Who? Where? When? This is the SABSA Contextual Architecture layer (Business View).
2. **Risk Assessment**: determination of the OWASP ASVS security verification level of the web application using Uganda NISF publications „Security Standard no 1 – Technical Risk Assessment" and „Security Standard no 3 – Security Classification". This is still the Contextual Architecture layer (Business View).
3. **Traceability Diagram**: a relationship diagram linking a web application's business attributes (requirements) with security attributes and ASVS security domains (also called as "chapters" in ASVS). This chain of traceability flows through the SABSA Contextual Architecture layer (Business View) to Conceptual Architecture layer (Architect's View) to Logical Architecture layer (Designer's View).
4. **Subdomains**: an overview of ASVS security domains and subdomains (also called as "sections" in ASVS). This is the SABSA Logical Architecture layer (Designer's View).
5. **Security Requirements:** a checklist of detailed ASVS security requirements (verification of security controls and mechanisms). This is the SABSA Physical Architecture layer (Constructor's View).
6. **Repository**: references to additional technical guides, standards, practices, catalogues. This is the SABSA Component Architecture layer (Technician's View).
7. **Operations**: a checklist of operational processes maintaining the application's security after the initial development and deployment. This is the SABSA Management Architecture layer (Manager's View).

There is also a hidden self-explanatory sheet **99. Toolkit settings.**

3 Contextual Architecture

Before a secure web application can be designed, the affected business processes, information and requirements must be identified. Missing a clear understanding of the business may not only lead to undesired or lacking functionality, but also to insecure processing of the business-critical data.

3.1 Contextual Survey

The WASA Framework starts building the understanding of the business at a high level in a form of a survey. This exercise should be feasible and achievable to each participant dealing with the security architecture. As the web application is developed to support some specific business process, the key information for this survey should be collected from the appropriate business function (such as the head of the business unit, member of the management board).

Considered and thoughtful answers to the following questions are successfully reflecting the context of the web application to the organisation's functions:

- **What**
  - is the web application used for (the main business goal the application is delivering)?
  - business function or assets need protection (e.g., personal, or other confidential data, reputation, government function, CII function)?

    - is the business need for information security (e.g., securing the digitalised business functions and information, enabling the business, compliance with regulations, operational continuity)?
- **Why** secure the web application? Describe the primary business risks regarding the web application (e.g., loss of reputation, regulatory sanctions, leakage of confidential data, CII service interruption).
- **How** are the business processes protected (e.g., alternative temporary workaround when the web application suffers downtime, prepared business communication with stakeholders during the downtime, redundant systems, effective service monitoring, QA and security testing)?
- **Who** are the stakeholders and interested parties of the web application (e.g., organisation's business unit, IT unit, security unit; user groups; software development unit or service provider; hosting provider; security testing service provider)?
- **Where** will it be used? Describe the geography and location-related aspects of the web application (e.g., internal usage only, between government organisations only, publicly accessible from the internet, hosting location and jurisdiction).
- **When** will it be used? Describe the time-dependencies and time-related aspects of the web application (e.g., working hours, business transaction throughput, lifetimes and deadlines, recovery time objective, recovery point objective).

3.2 Risk Assessment

SABSA risk-driven methodology emphasises business risks as the corner stones of information security. In the WASA Framework, the intention of risk assessment is the identification of ASVS security verification level of the web application under review. The application security verification level establishes the essential set of requirements to be verified – the higher the level, the more protection the web application needs, and the more requirements apply.

In a nutshell, ASVS security verification levels may be described as:

- Level 1 (Opportunistic): the very minimum for all web applications and suitable only for systems processing low-value non-sensitive data. While it comes to penetration/security testing of existent application, then Level 1 is the only level allowing black box testing and is focusing on the "low-hanging fruits".
- Level 2 (Standard): is suitable for most web applications processing sensitive business data, personal data, a special category of personal data, covering business areas like financial and health-care and similar.
- Level 3 (Advanced): is the highest level of assurance for areas like health and safety, critical infrastructure, military. The compromise of the application would lead to a significant threat to the whole organisation and beyond.

On a governmental level Uganda applies various national security standards. Avoiding any work duplication in the public sector, the WASA Framework employs two of them:

- "Security Standard no 1 – Technical Risk Assessment" to identify the technical risk level of the web application
- "Security Standard no 3 – Security Classification" to identify the highest level of information the web application is processing.

In the WASA Toolkit, the ASVS application security verification level is determined by selecting the levels of technical risks and processed information:

Security Standard no 1 – Technical Risk Assessment	OWASP ASVS
Risk Level	**ASVS Level**
Standard is not applicable	1
Very Low	1
Low	1
Medium	2
Medium-High	2
High	3
Very-High	3

Table 1. Security Standard no 1 - Technical Risk Assessment levels matched with ASVS levels

Security Standard no 3 – Security Classification			OWASP ASVS
Classification Level	**Impact Level**	**Business Impact**	**ASVS Level**
UNCLASSIFIED	0	Trivial	1
UNCLASSIFIED-PERSONAL	1	Low	2
OFFICIAL	2	High	2
SECRET	3	Extreme	3
TOP SECRET	4	Catastrophic	3

Table 2. Security Standard no 3 - Security Classification levels matched with ASVS levels

The final ASVS security verification level is the highest ASVS level from both tables.

This method and the Toolkit work accurately even if the technical risk assessment standard isn’t applicable for the organisation for some reason. It is still possible to conclude the information classification levels.

3.3 Business Attributes of Web Applications

Each web application supports the organisation and its business processes uniquely. Nevertheless, assessing common public sector information systems, most share a minimally viable set of business requirements. The following list[5] illustrates such essential business attributes (requirements) and shouldn’t be excluded without a strong justification:

- Access-controlled – access to information and web application functions within the system should be controlled in accordance with the authorized privileges of the party requesting the access. Unauthorized access should be prevented.
- Accountable – all parties having access to the system should be held accountable for their actions.
- Auditable – the actions of anyone with authorized access to the system, the outcomes of these actions as well as the complete chain of events should be recorded so that this history can be reviewed, along with the configuration of the system.

[5] The attribute explanation is retrieved from the *Appendix A. SABSA Business Attributes and Metrics* https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470476017.app1 and NIST Computer Security Resource Center. *Glossary* https://csrc.nist.gov/glossary.

- Business-Enabled – the primary objectives of the system design are to enable the business and fulfil the business objectives.
- Compliant – the system should comply with all applicable regulations, laws, contract, policies, and mandatory standards, both internal and external.
- Duty Segregated – for certain sensitive tasks, the duties should be segregated so that no user has access to both aspects of the task.
- Error-free – the system should operate without producing errors.
- Inter-operable – the system should interoperate with other similar systems (intersystem communication), both immediately and in the future.
- Maintainable – it's possible to maintain the system in a state of good repair and effective, efficient operation, and to do that within the normal operational condition of the system.
- Planned and Designed – the system should be undergo a thorough analysis of functional and non-functional specification and set the design, acquisition, development, implementation and operation activities accordingly.
- Private – personal information should be protected according to the relevant privacy legislation to meet the privacy expectations. Unauthorized disclosure should be prevented.
- Protected – the user's information and access privileges should be protected against abuse by other users or intruders.
- Reliable – the e-services provided to the user should be delivered at a reliable level of quality.
- Risk-managed – the design, acquisition, development, implementation, and operation of the system identifies and mitigates operational risk and prevents a wide range of potential abuses.
- Transparent – providing full visibility to the user of the logical process but hiding the physical structure of the system. In the data privacy context, transparent means that any information addressed to the public or to the data subject be concise, easily accessible and easy to understand, and that clear and plain language and, additionally, where appropriate, visualisation be used[6].
- Trusted – the system should be able to be trusted to behave in the ways specified in its functional specification and protecting against a wide range of potential abuses.
- Upgradeable – the system should be capable of being upgraded with ease to incorporate new releases of hardware and software.

The WASA Framework doesn't limit in any way the addition of new business attributes. Supplementary examples are listed in the SABSA whitepaper under Figure 4: The SABSA Taxonomy of ICT Business Attributes. Moreover, each organisation can add their unique attributes. In such a case, also security attributes and their relationships must undergo a respective review.

[6] EU General Data Protection Regulation. https://eur-lex.europa.eu/eli/reg/2016/679/oj

4 Conceptual Architecture

After understanding the business, the layered architecture moves one step down to the conceptual architecture (the architect's view). At this stage, the business attributes of the web application transform into security attributes.

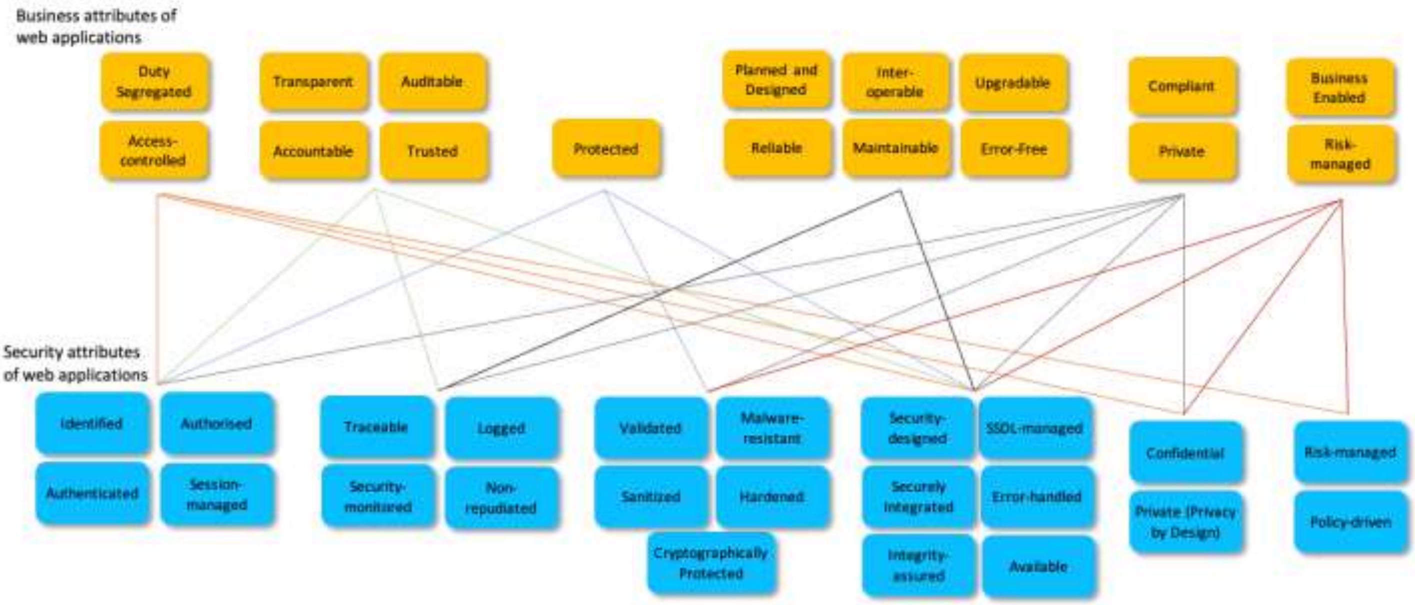

Figure 2. Business attributes' relationships to security attributes

Similarly to business attributes, WASA lists a minimal set of essential security attributes applicable for nearly each web application. Nonetheless, every organisation could add additional attributes while updating the relationships to business attributes and to ASVS security domains (see the Logical Architecture section).

Terminology wise, security attributes form the commonly known language of information security personnel. Different sources may provide slightly altered definitions, but on average, web application security attributes are explained as follows[7]:

- Authenticated – verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.
- Authorised – granting access based on a decision that evaluates a subject's attributes.
- Available – being accessible and usable upon demand by an authorized entity.
- Confidential – preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.
- Cryptographically Protected – providing protection via the principles, means, and methods for the transformation of data to hide their semantic content, prevent their unauthorized use, or prevent their undetected modification.
- Error-handled – handling a deviation from accuracy or correctness.
- Hardened – eliminating a means of attack by configuring the system (such as patching vulnerabilities, turning off nonessential services, using more strict settings).

[7] The source of the explanations are: ISO/IEC 27000:2016(E) *Information technology - Security techniques - Information security management systems - Overview and vocabulary*. NIST Computer Security Resource Center. *Glossary* https://csrc.nist.gov/glossary. ISACA, *Glossary* https://www.isaca.org/resources/glossary.

- Identified – discovering the identity of a user, process, or device from the entire collection of similar subjects.
- Integrity-assured – guarding against improper information modification or destruction.
- Logged – recording the events occurring within an organization’s systems and networks.
- Malware-resistant – resisting software or firmware intended to perform an unauthorized process that will have adverse impact on the confidentiality, integrity, or availability of a system.
- Non-repudiated – able to prove the occurrence of a claimed event or action and its originating entities.
- Policy-driven – driven by the statements, rules or assertions that specify the correct or expected behaviour of a system.
- Private (Privacy by Design) – being designed around the recognition of the right of a party to maintain control over and confidentiality of information about itself.
- Risk-managed – coordinating activities to direct and control a system with regard to risk.
- Sanitized – removing extraneous or potentially harmful data (e.g., malicious code) within a file or other information container (e.g., network protocol packet, query).
- Securely Integrated – customising (e.g., combining, adding, optimising) elements, processes, and systems in a way that mitigates security risk and prevents abuses.
- Security-designed – defining the system elements, interfaces, and other characteristics of a system of interest in a manner that reduces security risk and in accordance with the requirements and architecture.
- Security-monitored – capturing and interpreting information about the use of computers, networks, applications, and information to identify vulnerabilities and malicious usage.
- Session-managed – providing a persistent interaction between a user or service and an endpoint. A session begins with an authentication event and ends with a session termination event.
- SSDL-managed – (Secure Software Development Lifecycle) – planning, designing, developing, testing and implementing an application system or a major modification to an application system with a view to reduce security risk.
- Traceable – providing information that is sufficient to determine a specific aspect of an individual's activities or status.
- Validated – confirming, through a provision of objective evidence, that specified requirements have been fulfilled.

5 Logical Architecture

On the logical architecture layer (the designer’s view), WASA presents OWASP ASVS security domains, which in essence are 14 requirement groups:

- Architecture, Design and Threat Modeling Requirements
- Authentication Verification Requirements
- Session Management Verification Requirements
- Access Control Verification Requirements
- Validation, Sanitization and Encoding Verification Requirements
- Stored Cryptography Verification Requirements

- Error Handling and Logging Verification Requirements
- Data Protection Verification Requirements
- Communications Verification Requirements
- Malicious Code Verification Requirements
- Business Logic Verification Requirements
- File and Resources Verification Requirements
- API and Web Service Verification Requirements
- Configuration Verification Requirements

As a side note, security domains are called "chapters" in ASVS and the numerous subdomains are "sections".

The list of security domains is static and predefined in the OWASP ASVS standard. Hence, in this step, there is no action required by the implementer, except when additional security attributes were added in the previous layer. In such case, the relationship connectors between security attributes and security domains must be updated (see the next section Chain of Traceability). Each security domain is in more detailed explained in the WASA Toolkit under the worksheet "6.Repository".

6 Chain of Traceability

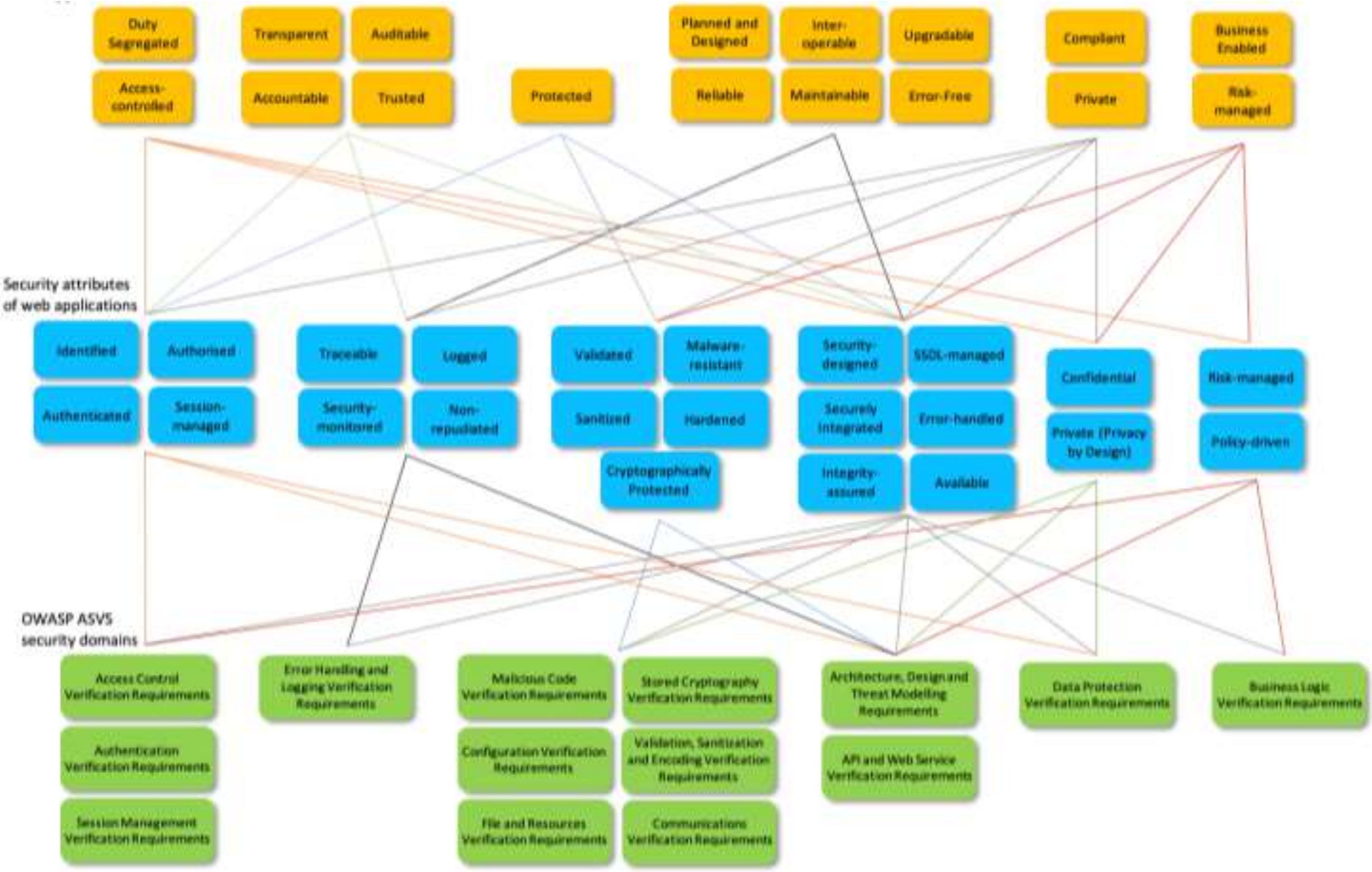
Duty Segregated
Access-controlled
Transparent
Auditable
Accountable
Trusted
Protected
Planned and Designed
Inter-operable
Upgradable
Reliable
Maintainable
Error-Free
Compliant
Private
Business Enabled
Risk-managed
Security attributes of web applications
Identified
Authorised
Authenticated
Session-managed
Traceable
Logged
Security-monitored
Non-repudiated
Validated
Malware-resistant
Sanitized
Hardened
Cryptographically Protected
Security-designed
SSDL-managed
Securely integrated
Error-handled
Integrity-assured
Available
Confidential
Private (Privacy by Design)
Risk-managed
Policy-driven
OWASP ASVS security domains
Access Control Verification Requirements
Authentication Verification Requirements
Session Management Verification Requirements
Error Handling and Logging Verification Requirements
Malicious Code Verification Requirements
Configuration Verification Requirements
File and Resources Verification Requirements
Stored Cryptography Verification Requirements
Validation, Sanitization and Encoding Verification Requirements
Communications Verification Requirements
Architecture, Design and Threat Modelling Requirements
API and Web Service Verification Requirements
Data Protection Verification Requirements
Business Logic Verification Requirements

Figure 3. Business attributes of web applications

The full relationship diagram of business attributes, security attributes, and ASVS security domains enables the bi-directional traceability as demonstrated on the following verification:

- Is every business requirement ensured by technical security requirements (and controls)?
- Is every technical security requirement (and control) justified by the business needs?

Using the WASA Toolkit, the chain of traceability has broader visibility. It makes the chain complete as the ASVS security domains are unfolded to subdomains and subdomains to individual technical security requirements (controls to be verified).

7 Physical Architecture

The physical architecture (the builder's view) of WASA comes in the form of the WASA Toolkit, where the worksheet "5.Security Requirements" outlines all ASVS individual technical security requirements applicable for the web application in a structured manner:

ASVS Security Domains -> Subdomains -> Security Requirement

The most significant value for the web application's security is formed on this architecture layer. Each security requirement reflects a presence of security control. All applicable security requirements must be fully understood and implemented in the software code or configuration to result in a robust and resilient application. Worth mentioning, it is so much more resource-friendly to design and include the security aspects to the source code from the beginning (security by design principle) than adding the missing puzzle pieces to a finished system. However, better late than never.

The WASA Toolkit is meant as a checklist where the progress of security verification checks can be stored. Each requirement should have a status:

- Not Verified
- In Progress
- Implemented
- Unsolved
- Not Applicable

As the very first action in this step, the filtering for only relevant requirements should take place based on the OWASP ASVS security verification level, which was identified in the risk assessment phase (Toolkit's worksheet "2.Risk Assessment).

At the same time, the Comments field may be used to highlight relevant information, such as:

- Person assigned to the task
- Deadline
- Short description of the implementation
- Short description of the issue
- Link or ID to a ticketing tracking system.

The MITRE CWE[8] Reference field provides additional optional justification for security requirements connecting them with common software weaknesses. It may come in handy while web application weaknesses and appropriate controls are under discussion.

[8] MITRE Common Weakness Enumeration. https://cwe.mitre.org/

8 Component Architecture

Belonging to their common expertise, web application developers, security architects and other security personnel comprehend ASVS security requirements well. As web technology varies quite broadly and evolves constantly, supporting material that helps to verify or implement the requirements may still come in handy. The component architecture (the tradesman's view) explains ASVS security domains, but also focuses on a granular level to individual controls. In the WASA Toolkit, the worksheet "6.Repository" provides external resources to technical guides, best practises, cheat sheets, and standards.

9 Management Architecture

Designing and integrating security into the web application during the software development phase is crucial but not the only critical stage in the software lifecycle. The absolute must-have information security management (ISM) processes occur after the application's deployment to the production environment. The management architecture (the service managers' view) in the WASA Framework focuses on operational security (OPSEC), also called security operations. OPSEC consists of processes and procedures related to the day-to-day operational functions supporting the security environment.

Originally, the OPSEC ideology stems from the US military intending to deny adversaries information about capabilities and intentions. Today, the term is used more widely to continuously maintain an organisation's agreed level of security. Once the level of security is established in the policies, we speak of a policy-driven security architecture. In other words, OPSEC brings the policy-driven security architecture to life.

The WASA Toolkit covers essential 13 questions for self-assessment:

- Are all assets (e.g., physical and virtual servers, web servers, database servers, application servers, load balancers) added to the IT asset repository?
- Are primary and secondary administrators assigned to each web application system component?
- Have all system components and platforms hardened configurations (e.g., based on well-known hardening guides provided by CIS Security, NIST NCP or similar)?
- Are all user and service accounts, which are related to the web application's front and backend system components, included in the organisation's primary Identity and Access Management (IAM) process?
- Are all relevant logs, which are related to the web application front and backend system components (e.g., logs of web server, API, application, database queries, TLS-session management), transported to the organisation's central log repository?
- Are the web application and all its components added to the organisation's primary security monitoring process?
- Are the web application and all its components added to the organisation's primary patch management process?
- Are the web application and its components added to the organisation's primary vulnerability management process, including the vulnerability scanning target list?
- Are all the relevant web application components and data added to the organisation's backup management process?
- Are all the relevant web application components and data added to the organisation's business continuity management process?

- Are all the outsourced suppliers related to the web application management and maintenance (e.g., cloud hosting providers, server room providers, further development partners) added to the organisation's supplier management and contract management process?
- Are the organisation's incident responders aware of and prepared for the incidents related to the web application?
- Has security testing (e.g., penetration testing based on OWASP ASVS, recommended as white-box testing) of the web application been performed?

Common ISM processes delivering the OPSEC function are:

- Asset Management (not always directly information security process)
- Security Administration
- Vulnerability and Patch Management
- Security Event and Log Management
- Security Incident Management
- Security Compliance, Testing and Measurement, providing objective and practical understanding about the maturity of the current ISM.

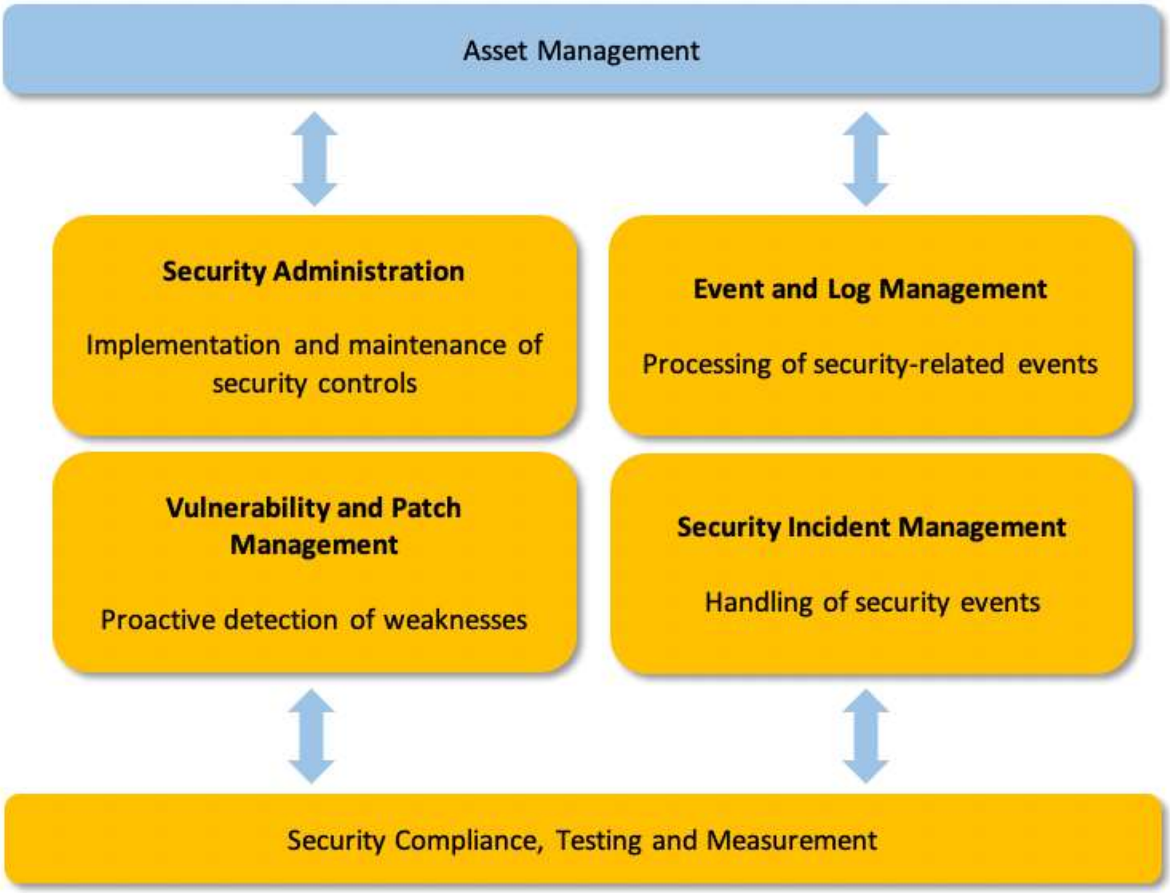
Asset Management
Security Administration
Implementation and maintenance of security controls
Event and Log Management
Processing of security-related events
Vulnerability and Patch Management
Proactive detection of weaknesses
Security Incident Management
Handling of security events
Security Compliance, Testing and Measurement

Figure 4. Operational Security Architecture

9.1 Asset Management

OPSEC success depends directly on the asset management process driven by a simple principle – it is only possible to protect those assets whose existence is acknowledged. An asset is defined as anything valuable to the organisation and worth protecting, either tangible or intangible, such as (but not limited to), software application, computing hardware, virtual device, IT network, facilities, people, reputation. Asset management is not counted only as a security activity but also part of the financial, accounting, and IT management process. ISM cannot achieve its goals without it; therefore, most security management standards and methodologies incorporate asset management.

9.2 Security Administration

By security administration we understand the configuration and hardening of assets against unauthorised or accidental misuse to ensure their confidentiality, integrity and availability; and deployment, implementation, enforcement and monitoring of security controls according to the established security policies. Deployment tasks were considered an IT function, but there has been a shift with the rapid increase of information security importance. The typical all-in-one CISO has been expanded into full-scale security teams, such as SecOps, SOC/NOC, CSIRT teams. With this change, the deployment and implementation of security controls and solutions may be the responsibility of the security function, depending on the organisation's decisions, resources and structure. The objective remains similar – implement, configure, enforce security controls to harden systems and infrastructure to meet security policies' security level, and ensure the availability, integrity and confidentiality of information.

Security Administration consists of two main processes:

- **Identity and Access Management (IAM)** – encapsulates people, processes and products to identify and manage the information used to authenticate users and grant or deny access rights to information, systems or facilities. The goal of IAM is to provide appropriate access to business resources. Next to technology, IAM also involves information classification, personnel security, supply chain and physical security.
- **System Configuration** – managing the security and privacy settings and setup of hardware and software systems. This process may be challenging as the number of different devices, their settings, possible controls, and relationships to other components are enormous.

9.3 Vulnerability and Patch Management

Vulnerability and patch management identifies - reactively and proactively - security weaknesses of assets and incorporates the patching of them. Awareness about the existence of assets, their specification, and risk ratings is crucial. Therefore, direct integration with asset management and Information Security Risk Management (ISRM) processes must be ensured.

The **reactive** approach includes the following steps:

- While onboarding a new asset, assign an administrator and an owner
- Sign up for and receive vulnerability notifications from the vendor
- Search the asset repository to find affected assets
- Conduct risk assessment
- If a patch is released, test it and deploy
- If a patch is not released, install an alternative security control for high-risk assets or wait for the patch for low-risk assets.

The **proactive** approach includes following steps:

1. While onboarding a new asset, assign an administrator and an owner
2. Launch periodical automated vulnerability scans and receive the assessment report
3. Search the asset repository to find other potentially affected assets
4. Conduct risk assessment
5. If a patch is released, test it and deploy

6. If a patch is not released, install an alternative security control for high-risk assets or wait for the patch for low-risk assets.

A correctly established vulnerability and patch management process involves reactive and proactive components.

9.4 Security Event and Log Management

Security event and log management provides a process for daily management (collection, normalisation, aggregation, correlation, reporting) of the security-related events of network, endpoints, storage systems and security solutions. Recording security events and generating evidence helps to enable security incident management. Security logs lay the foundation of automated monitoring systems like IDS, IPS, EDR, SIEM, and advanced detection methods, such as threat hunting.

Five key areas must be considered while designing the process strategy:

- Security logs must be collected in real time and stored securely
- Logs must be normalised, aggregated, correlated, and analysed
- A normal baseline must be defined
- A normal situation (baseline) must be defined
- Any deviation from the baseline must generate a notification and activate the security incident management process.

9.5 Security Incident Management

Security incident management aims to avoid or reduce the negative impact of security incidents by responding to threats and recovering the normal level of business operations. This process involves:

- Receiving alerts from the security event and log management process
- Analysis, response, including legal/regulatory response, and recovery
- Communication and reporting
- Learning, narrowing down the surface for security incidents and improving the process.

9.6 Security Compliance

Security compliance does not focus on compliance with regulations and law but ensures that the implemented technology meets the organisation’s policies, standards, guides, procedures and architecture. This activity should provide the answer to whether the deployed security controls are effective and whether the security function is delivering its objectives as defined in the policies. To achieve this conformity, the organisation must establish:

- A **monitoring** process and tools to collect information about implemented solutions and their settings, and to compare this information to a defined and desired state. Monitoring can be a manual review or an automated activity.
- An **alerting** process and tools to deliver notifications about identified abnormal situations to appropriate personnel. Alerting must be closely integrated with the security administration process, which fixes cases that are out of alignment.

9.7 Security Testing

While the security compliance process focuses on the conformity of the policies with the implemented controls, security testing identifies the resistance against malicious actions. Testing should not only verify the evidence (such as the presence of log records, review of configuration) but take steps forward by doing static or dynamic analyses and executing assessments in real time to compare actual with expected behaviour. However, the general goal remains the same – to see that the requirements defined in the policy are effectively deployed to the technology and management processes.

The reasonable approach of designing security testing should consider:

- The maturity of ISM in the organisation, especially the operational security – if the maturity is low, perform testing of single systems and components; if the maturity reaches higher levels, perform complex combined assessments, such as red teaming.
- The risk level of assets – test high-risk assets first, such as internet-facing web applications.
- Competence of the test team – it is not unusual that more complex testing is done in the production environment; therefore, the risk of breaking business services may become a reality if the team lacks skills and experience. Another issue with amateurs is false negative conclusions where the actual vulnerabilities are not correctly detected.
- Systematic and measurable testing methodology – testing should follow a known testing standard or method so that the outcome could be similarly understood and compared by security experts.

In addition, the following security testing standards, methodologies and guides may be considered:

Title	Purpose
OWASP Web Security Testing Guide[9]	Web application security testing
OWASP Mobile Application Security Verification Standard[10]	Mobile application security testing within the SDLC
MITRE ATT&CK[11]	Adversary assessment, threat modelling, and TTPs
PCI DSS Penetration Testing Guidance[12]	Payment card infrastructure testing guide
GCHQ NCSC Penetration Testing Advice[13]	Penetration testing planning considerations
TIBER-EU Framework[14]	Planning of threat-based ethical red teaming against banking sector
CREST Penetration Testing Guide[15]	Penetration testing planning and management considerations
NIST SP 800-115. Technical Guide to Information Security Testing and Assessment[16]	Practical recommendations for designing, implementing, and maintaining technical information security test

[9] https://owasp.org/www-project-web-security-testing-guide/

[10] https://owasp.org/www-project-mobile-security-testing-guide/

[11] https://attack.mitre.org/

[12] https://www.pcisecuritystandards.org/documents/Penetration-Testing-Guidance-v1_1.pdf

[13] https://www.ncsc.gov.uk/guidance/penetration-testing

[14] https://www.ecb.europa.eu/pub/pdf/other/ecb.tiber_eu_framework.en.pdf

[15] https://www.crest-approved.org/wp-content/uploads/CREST-Penetration-Testing-Guide.pdf

[16] https://csrc.nist.gov/publications/detail/sp/800-115/final

Title	Purpose
Open Source Security Testing Methodology Manual[17]	Methodology to test the operational security of physical locations, human interactions, and all forms of communications

9.8 Measurement

The key to improvement and efficient management is measurement. The main goal of measurement is to provide decision support to top management, managers, IT and security personnel. Security metrics provide the opportunity to:

- Understand to what extent security objectives are achieved
- Measure the effectiveness of security architecture, controls, processes
- Make informed risk management conclusions
- Define benchmarks, baselines and identify anomalies.

The true value of measurement comes only with sensible metrics. Avoiding less-than-useful interpretations and unsound decisions, the design of each metric must follow specific characteristics:

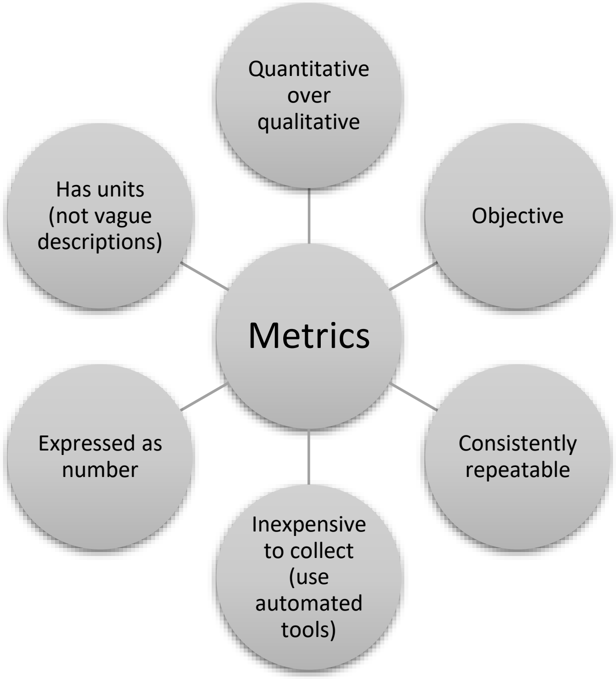
Quantitative over qualitative
Has units (not vague descriptions)
Objective
Metrics
Expressed as number
Consistently repeatable
Inexpensive to collect (use automated tools)

Figure 3. Measurement and Metrics

During the design of the metrics, it is important to understand the measurement objective and time dimension to ensure that the chosen measure is able to support the goal. There are three types of metrics:

[17] https://www.isecom.org/OSSTMM.3.pdf

Types of metrics based on time dimensions	**Input**	**Output**	**Example**
Historical metrics	Logs	Reports, Dashboards	Firewall report with the number of endpoints connected to a malicious IP within last month
Real-time metrics	Logs, Alerts	Dashboards, Notifications	Sudden growth (%) of SYN packets against the firewall
Predictive metrics	Logs, Models	Forecast	End of free resource in hh:mm because of the growth of SYN packets

The architecture of security metrics should include a clear specification about each metric, such as:

- Metric name/ID
- Metric objective
- Metric description
- Units of the metric
- Target value (benchmark)
- Formula, algorithm, logic of the metric
- Frequency
- Data sources
- Responsible parties (information customer, information collector)
- Reporting format and schedule

Mature organisations that need more in-depth security measurement architecture may follow the NIST SP 800-55 Performance Measurement Guide for Information Security[18]. In addition to design guidelines, it contains a sample set of measures (metrics) in Appendix A: Candidate Measures. Another great source for defining business attribute metrics is SABSA's Appendix A. SABSA Business Attributes and Metrics[19].

[18] NIST SP 800-55 Rev. 1. Performance Measurement Guide for Information Security. https://csrc.nist.gov/publications/detail/sp/800-55/rev-1/final

[19] Appendix A. SABSA Business Attributes and Metrics. https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470476017.app1

10 Abbreviations

Abbreviation	Meaning
API	Application Programming Interface
ASVS	Application Security Verification Standard
CII	Critical Information Infrastructure
CISO	Chief Information Security Officer
CSIRT	Computer Security Incident Response Team
e-GIF	eGovernment Interoperability Framework
EDR	Endpoint Detection and Response
GEA	eGovernment Enterprise Architecture
IAM	Identity and Access Management
IDS	Intrusion Detection System
IPS	Intrusion Protection System
ISM	Information Security Management
ISRM	Information Security Risk Management
OWASP	Open Web Application Security Project
OPSEC	Operational Security
SABSA	Sherwood Applied Business Security Architecture
SSDL	Secure Software Development Lifecycle
SecOps team	Security Operations team
SIEM	Security Information and Event Management
SOC/NOC	Security Operations Center / Network Operations Center

11 Glossary

Term	Definition
Business Process	A business process is a sequence of linked activities that creates value by turning inputs into a more valuable output. This can be performed by human participants or ICT systems, or both.
eGovernment	E-government is about using the tools and systems made possible by information and communication technologies (ICTs) to provide better public services to citizens and businesses.
eGovernment Enterprise Architecture (GEA)	The structure of e-government components, their relationships, and the principles and guidelines governing their design and evolution over time. In a broader sense, Enterprise Architecture aligns processes, people, and technology (which together make a system) with supporting information / IT systems to realize goals in an effective and efficient manner.
eGovernment Interoperability Framework (e-GIF)	The agreed approach to interoperability for the GoU MDAs that wish to work together towards the joint delivery of public services. Within its scope of applicability, it specifies a set of common elements such as vocabulary, concepts, principles, policies, guidelines, recommendations, standards, specifications, and practices.
Information Security Management (ISM)	Process that defines the security requirements, objectives, identifies information security risks, and supports the implementation of security controls. The ISM establishes, implements, maintains, and continually improves an Information Security Management System (ISMS) within the context of the organization. ISMS is sometimes referred to as "Information Security Program".
Operational Security (OPSEC)	Security domain, which focuses on processes and procedures related to day-to-day operational functions supporting the security

	environment to ensure that all objectives are achieved. Also called security operations.
Web API (Application Program Interface)	A set of functions and procedures that can be used to program application that interacts with other application. API is typically done as HTTP/REST architectural style, output could be JSON/XML, input can be XML/JSON/or plain data. Not officially defined standard.
Web Application	An application software that runs on a web server, and are accessed by the user through a web browser, or by other applications over APIs or web services.
Web Service	A software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Defined by W3C (https://www.w3.org/TR/ws-arch/).