GovStack

Building Block Definition
Security Requirements

Developed by: David Forden, Jean-Reynald Vivien-Gayout de Falco.
In cooperation with GIZ, ITU, DIAL, and the Government of Estonia

GovStack

Table of Contents

GovStack

1 Version History

Version	Authors	Comment
.5.0	David Forden	Draft content created
1.0.0	David Forden	Peer comments addressed. Ready for external review.
1.1.0	Jean Reynald	TAC review feedbacks addressed and added to current, future and out of scope requirements

2 Description

This document is intended to be used as a reference for the security requirements for GovStack by vendors proposing solutions for all building blocks as well as vendors proposing solutions for this security building block.

Security requirements address all cross-cutting security issues and concerns for the whole GovStack digital platform including every layer, every building block and all applications. Although other building blocks address "some" security aspects such as "Identity building block" (addressing the foundational identity aspects and document workflows etc.) the resultant solutions delivered by all building-blocks (including the "Identity building block") MUST comply with the standards and requirements set by this security requirements document. This document covers security requirements of two types:

- **Build-time Security:** These are considerations for embedding security during development of building blocks and applications.
- **Deployment time Security:** These are considerations for enforcing security measures in deployed systems during run-time.

These may consist of cross cutting functionalities that can be utilized for various building blocks and specific requirements for the **Security Building Block itself, to** provide secure internet access for user interaction with applications and building blocks in Govstack.

The security requirements are based on the NIST CyberSecurity Framework and defined herein through review of GovStack use cases and best practices https://solutions.dial.community/building_blocks/security for securing and hardening government infrastructure. It MUST also be noted that the security building block defines the core requirements to implement policy based API security and management

across the internal building blocks as well as external applications and 3rd party services consumption. This is based on the architectural assumption that all inter-building block communication/integration with external applications and users MUST be through REST APIs.

3 Terminology

Term or Acronym	Meaning and Expansion	Comments and Links
Access	A general term that describes the granting and restriction of access to resources for subjects.	To open a computer file or to use a computer system such as the internet. https://dictionary.cambridge.org/fr/dictionnaire/anglais/access
Authentication	The validation of user credentials for the purpose of system login and basic access.	Authentication is the process of recognizing a user's identity. https://economictimes.indiatimes.com/definition/authentication
Authorization	The granting of privileges or rights for accessing the various resources hosted by a system, to a subject via a role or group for example.	Authorization is the process of giving someone permission to do or have something. https://searchsoftwarequality.techtarget.com/definition/authorization
CIS	The Center for Internet Security (CIS) benchmarks are a set of best-practice cybersecurity standards for a range of IT systems and products. CIS Benchmarks provide the baseline configurations to ensure compliance with industry-agreed cybersecurity standards.	CIS is an independent nonprofit organization with a mission to create a confidence in a connected world. https://www.cisecurity.org
CSPM	Cloud Security Posture Management is a solution suite that enables administrators to keep track of the way	CSPM is a market segment for IT security tools that are designed to identify misconfiguration issues and compliance risks in the cloud.

	in which both home grown and 3rd party services and applications access public cloud provider resources from a security perspective and enables vulnerabilities to be resolved.	https://searchcloudsecurity.techtarget.com/definition/Cloud-Security-Posture-Management-CSPM
CUI	Confidential Unclassified Information as defined by NIST 800-171 Rev 2	Controlled Unclassified Information (CUI) is information that requires safeguarding or dissemination controls consistent with applicable laws, regulations, and Government-wide policies. https://www.epa.gov/cui/controlled-unclassified-information-cui-program-frequently-asked-questions-faqs
CVE	Common Vulnerabilities and Exposures - a known vulnerability in a system or network component which can be exploited by a malicious attacker to gain access or create havoc.	CVE, short for Common Vulnerabilities and Exposures, is a list of publicly disclosed computer security flaws. https://www.redhat.com/en/topics/security/what-is-cve
DevOps and DevSecOps	A set of principles and practices used along with tools that fully integrates and expedites the process of building, securing and deploying code on a scheduled and/or demand basis with the goals of reduced errors, reduced time-to-market, increased security and increased accuracy among others.	DevOps focuses on collaboration between application teams throughout the app development and deployment process. DevSecOps evolved from DevOps as development teams began to realize that the DevOps model didn't adequately address security concerns. https://www.appdynamics.com/blog/product/devops-vs-devsecops
DLP	Data Leakage Prevention - a solution typically used to prevent confidential or private information from leaking outside the organization to unauthorized 3rd parties.	Data loss prevention (DLP) is a set of tools and processes used to ensure that sensitive data is not lost, misused, or accessed by unauthorized users. https://digitalguardian.com/blog/what-data-loss-prevention-dlp-definition-data-loss-prevention

Federation	Federated security allows for clean separation between the service a client is accessing and the associated authentication and authorization procedures. Federated security also enables collaboration across multiple systems, networks, and organizations in different trust realms.	Federated identity is a method of linking a user's identity across multiple separate identity management systems. https://www.okta.com/identity-101/what-is-federated-identity
GLBA	The Gramm-Leach-Bliley Act (GLB Act or GLBA) is also known as the Financial Modernization Act of 1999. It is a United States federal law that requires financial institutions to explain how they share and protect their customers' private information. It is also a generally accepted global standard.	The Gramm-Leach-Bliley Act requires financial institutions – companies that offer consumers financial products or services like loans, financial or investment advice, or insurance – to explain their information-sharing practices to their customers and to safeguard sensitive data. https://www.ftc.gov/tips-advice/business-center/privacy-and-security/gramm-leach-bliley-act
HIPAA	Established United States federal standard to protect individuals' medical records and other personal health information and applies to health plans, health care clearinghouses, and those health care providers that conduct certain health care transactions electronically. It is a generally accepted standard globally.	The Health Insurance Portability and Accountability Act of 1996 (HIPAA) is a federal law that requires the creation of national standards to protect sensitive patient health information from being disclosed without the patient's consent or knowledge. https://www.cdc.gov/phlp/publications/topic/hipaa.html
IAM	Identity and Access Management - typically refers to a security suite that implements the infrastructure required for Authentication and Authorization plus the management of identities, roles, groups and access.	Identity and access management (IAM) is the discipline that enables the right individuals to access the right resources at the right times for the right reasons. IAM addresses the mission-critical need to ensure appropriate access to resources across increasingly heterogeneous technology environments and to meet increasingly rigorous compliance requirements

		https://www.gartner.com/en/information-technology/glossary/identity-and-access-management-iam
IMAP	Internet Message Access Protocol is a mail client. protocol used for retrieval of email messages from a mail server. For the purposes of this document IMAP refers to IMAP4 which is defined by the IETF with multiple RFCs.	Internet Message Access Protocol (IMAP) is a protocol for accessing email or bulletin board messages from a (possibly shared) mail server or service. https://www.gartner.com/en/information-technology/glossary/imap-internet-message-access-protocol
OAuth2	An open standards based protocol used for Authentication that uses bearer tokens and is specifically designed to work across HTTP. OAuth provides clients a "secure delegated access" to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without providing credentials. OAuth2 is the second major release of OAuth which has been hardened based on known attacks such as “AS MixUp”. Not all implementations of OAuth2 are equal and some have been found to have security flaws.	The OAuth (open authorization) protocol was developed by the Internet Engineering Task Force and enables secure delegated access. https://oauth.net/2
OpenIDConnect	A simple open standards based identity layer on top of the OAuth 2.0 protocol. It allows Clients to verify the identity of a party based on the authentication performed by an Authorization Server, as well as to obtain basic profile	OpenID Connect lets developers authenticate their users across websites and apps without having to own and manage password files. https://openid.net/connect/faq

	information about the party in an interoperable and REST-like manner	
OWASP	The Open Web Application Security Project is an online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security.	The Open Web Application Security Project, or OWASP, is an international non-profit organization dedicated to web application security. https://www.cloudflare.com/learning/security/threats/owasp-top-10
PaaS	Platform As A Service: A suite of software components that is fully integrated to provide a secure, convenient and rapid application development and deployment platform for cloud style applications.	PaaS (Platform as a Service), as the name suggests, provides you computing platforms which typically includes operating system, programming language execution environment, database, web server https://stackoverflow.com/questions/16820336/what-is-saas-paas-and-iaas-with-examples
PCI DSS	A set of standards used by the payment card industry to secure payment card data and card holder information including primary account numbers (PAN), credit/debit card numbers, and sensitive authentication data (SAD) such as CVVs and PINs.	The Payment Card Industry Data Security Standard (PCI DSS) is required by the contract for those handling cardholder data, whether you are a start-up or a global enterprise. https://www.controlcase.com/what-are-the-12-requirements-of-pci-dss-compliance
POP	Post Office Protocol - a standard email protocol used by clients to access email once delivered to a mail server in a specific DNS domain. Various versions of this protocol exist but for the purposes of this document POP refers to POP3 as defined by RFC1939 and the extension mechanism in RFC2449 and an authentication mechanism defined in RFC1734	The post office protocol (POP) is the most commonly used message request protocol in the Internet world for transferring messages from an e-mail server to an email client. https://www.sciencedirect.com/topics/computer-science/post-office-protocol

Provisioning	A way of propagating the joining or leaving of users from the system and creating/removing the accounts and access rights for users based on their target profile/role.	In general, provisioning means "providing" or making something available. In a storage area network (SAN), storage provisioning is the process of assigning storage to optimize performance. In telecommunications terminology, provisioning means providing a product or service, such as wiring or bandwidth. https://whatis.techtarget.com/definition/provisioning
Realm	A realm is a security policy domain defined for a web or application server. A realm contains a collection of users, who may or may not be assigned to a group. An application will often prompt for a username and password before allowing access to a protected resource. Access for realms can be federated.	A realm is a security policy domain defined for a web or application server. The protected resources on a server can be partitioned into a set of protection spaces, each with its own authentication scheme and/or authorization database containing a collection of users and groups. https://stackoverflow.com/questions/8468075/what-is-the-exact-uses-of-realm-term-in-security
SAML	Security Assertion Markup Language. SAML and SAML2 are XML markup protocols (a suite of XMLSchema message types) designed for federation of identities across identity providers and service providers. Its main use case is for web single-sign-on.	Security Assertion Markup Language (SAML) is an open standard for sharing security information about identity, authentication and authorization across. https://www.techtarget.com/searchsecurity/definition/SAML
SCEP	Simple Certificate Enrolment Protocol used to enroll users and issue digital certificates. Typically supported by the certificate authority server.	Simple Certificate Enrollment Protocol (SCEP) is an open source protocol that is widely used to make digital certificate issuance at large organizations easier, more secure, and scalable. https://www.hypr.com/simple-certificate-enrollment-protocol
Single Sign On (SSO)	A way of ensuring that users only need to enter credentials once in order to gain policy access to resources across security realms.	Single sign-on (SSO) is an authentication method that enables users to securely authenticate with multiple applications and websites by using just one set of credentials.

		https://www.onelogin.com/learn/how-single-sign-on-works
SMTP	Simple Mail Transfer Protocol - a protocol used to route email between gateways to the server responsible for final delivery to a specific DNS mail domain.	The Simple Mail Transfer Protocol (SMTP) is used to deliver e-mail messages over the Internet. This protocol is used by most email clients to deliver messages to the server, and is also used by servers to forward messages to their final destination. https://www.sciencedirect.com/topics/computer-science/simple-mail-transfer-protocol
Subject	In a security context, a subject is any entity that requests access to an object. These are generic terms used to denote the thing requesting access and the thing the request is made against. When you log onto an application you are the subject and the application is the object	The term subject to represent the source of a request. A subject may be any entity, such as a person or service. A subject is represented by the javax. security. auth. https://stackoverflow.com/questions/4989063/what-is-the-meaning-and-difference-between-subject-user-and-principal
XACML	eXtensible Access Control Markup Language The XACML standard defines a declarative fine-grained, attribute-based access control policy language, an architecture, and a processing model describing how to evaluate access requests according to the rules defined in policies all in XMLSchema.	XACML (Extensible Access Control Markup Language) is an open standard XML-based language used to express security policies and access rights to information. https://searchcio.techtarget.com/definition/XACML

4 Security Management

The key security functionalities outlined here describe the required facilities that this security building block MUST provide as well as security compliance measures that must be implemented by all building blocks. Note that specific API definitions are not likely to be created by the security building block as any interfaces required are to be based on open standards and implemented as part and parcel of acquired solutions. A good example of this is the adoption of standards like OAuth2 and OpenIDConnect for authentication and authorization. The functional requirements for the implementation of an appropriate API Management and Gateway services solution can be found in a separate section of this document below.

The basic framework by which security MUST be addressed for GovStack is largely based on the NIST CyberSecurity Framework (hitherto referred to as NIST CSF) and the NIST 800-171 Rev.2 standard (hitherto referred to as NIST 800-171) for managing controlled unclassified information (CUI) but does also incorporate other security related requirements. The Specific GovStack Security Related Concerns is organized in terms of the major functions of the NIST CSF which is defined by NIST as 3 major approaches/facets for implementation:

- **core** - provides a set of desired cybersecurity activities and outcomes using common language that is easy to understand,

- **tiers** - assist organizations by providing context on how an organization views cybersecurity risk management. The tiers also guide organizations to consider the appropriate level of rigor for their cybersecurity program and are often used as a communication tool to discuss risk appetite, mission priority, and budget and;

- **profiles** - roll your own suitable profile based on your own needs. Profiles are an organization's unique alignment of their organizational requirements and objectives, risk appetite, and resources against the desired outcomes of the core. Profiles are primarily used to identify and prioritize opportunities for improving cybersecurity at an organization of NIST CSF

The major functions of NIST CSF are

- **Identify:** provides the required abilities that enable a deployer to accurately determine the identity of any party that endeavors to access any part of the system and the issues that are associated with that, from every possible perspective.

- **Protect:** provides the abilities required to protect and govern along with the issues that pertain not only authentication and access but fraud, hacking, phishing and all other forms of malicious intrusion and theft.

- **Detect:** provides the required abilities that enable a deployer to detect any form of malicious intrusion or attempts to intrude along with the interdiction abilities and identifies all of the issues that pertain to this.

- **Respond:** provides the required abilities that enable a deployer to respond to any form of malicious intrusion or attempts to intrude along with the issues and processes for interdiction and the response to such events.

- **Recover:** provides the requires abilities that enable a deployer to recover from any indecent (malicious or otherwise) such as breaches or attempted breaches of security and identifies the issues that must be addressed and the processes required to address them

4.1 Specific GovStack Security Related Issues

This section of the document provides a specific list of the concerns, principles, procedures and actions (collectively termed as security related issues) that have been identified for GovStack along with:

- How each issue maps to the existing building blocks and their respective working groups

- What type of organizational risk is anticipated if the issue is not addressed (i.e. high/medium/low)

- Which target phase of the project the issue must be addressed in (i.e. first/second) - third phases are usually never completed

- How feasible it is to address the issue in a limited-resource or low-resource setting (predominantly related to costs) - see the Architecture Blueprint and Non-Functional Requirements document for the definitions associated with low-resource settings.

A general description and/or discussion of the issue that needs to be addressed and the various alternatives available for addressing it (not exhaustive)

	Security Related Issues	WG/BB Mapping	Organizational Risk Rating (high/ medium/ low)	Target Deployment Phase	Feasibility for Limited or Low Resource Settings	Description, Discussion and Potential Solutions	Comments
	IDENTIFY						
1.	Authentication and Authorization	Identity/ Security	High	First	High	Authentication and Authorization MUST be addressed across the board. It is likely to be built-in to API Management and Gateway and accessed by mobile and web applications using a token based approach. All communications from all clients (web/mobile/BB clients etc.) MUST be via API so this is a sensible point of implementation given the stateless nature of applications. Authentication and authorization MUST also be addressed at an application access level for each and every application (web/mobile/desktop etc). It would be wise to utilize the same framework and capabilities for this.	Each building block MUST implement centralized authentication and authorization (minimally proxied or implemented via the common IAM solution and/or API Management and Gateway services).
2	Multi-Factor Token and Password Strength/ Complexity Management etc.	Identity/ Security	High	First	High	Credential strength management is likely built-in to API Management and Gateway. All communications from all clients (web/mobile etc.) must be via API so this is a sensible point of implementation also. Each application MUST provide the ability to determine credential strength at the time of registration and thus permit or deny the offered credential.	Each building block MUST implement Multi-Factor Token and Password Strength/Complexity Management for an indeterminate array of factors including biometric (this is to be implemented through leveraging a common IAM solution).
3	Access Control (RBAC, MAC, DAC)	Identity/ Security	High	First	High	Access control is also likely built-in to API Management and Gateway. All communications from all clients (web/mobile etc.) must be via API. The thin client nature of	Each building block MUST implement role based access control for all exposed API's and resources (minimally proxied or implemented via the API Management and Gateway services)

						web and mobile dictates that all resources will exist behind API interfaces so this is the most sensible point to address it (i.e. either they have access to the API or they do not).	
4	Provisioning, Deprovisioning and Management of Identities and access rights	Identity/ Security	High	First	High	Will need a process based solution for this. This selection will largely depend on the identity management and API management infrastructure chosen but will likely need to be a customized process that integrates provisioning across a number of products and services.	Each building block MUST implement the ability to provision, deprovision and manage Identities and access rights (this may or may not be centralized for the whole architecture as a unified provisioning process).
5	Access and Authorization Audit, Logging, Tracing, Tracking	Identity/ Security	High	First	High	Likely built-in to API Management and Gateway. All communications from all clients (web/mobile etc. must be via API)	Each building block MUST implement access and authorization audit, logging, tracing and tracking with alerts (minimally proxied or implemented through the API Management and Gateway services).
6	End User Device Registration, Deregistration, Re Registration and Device Platform Security	Identity/ Security	High	First	High	Significant functionality provided by MOSIP - TBD	Each building block dealing with physical devices MUST implement end user device registration, deregistration, re-registration and device platform security guidance/requirements to end users.
7	Biometric Security Credentials, Devices, Registration, Deregistration, Validation and device platform security etc. such as above for end user devices	Identity/ Security	High	First	High	Significant functionality provided by MOSIP - TBD	Each building block dealing with biometric credentials MUST implement biometric security credential management, registration, deregistrations, re-registration, validation and device platform security etc. such as above for biometric capture devices.
8	Non-repudiation and Certificates (X509, OpenID and SAML etc.)	Identity/ Security	High	FIrst	High	Likely built-in to API Management and Gateway but may also require certificate server etc. All communications from all clients (web/mobile etc. must be via API)	Each building block MUST implement a framework for non-repudiable transactions using certificates and federation protocols (X509, OpenID and SAML 2.0 etc.)

9	Single-Sign-On and 3rd Party Security	Identity/ Security	High	First	High	Likely built-in to API Management and Gateway. All communications from all clients (web/mobile etc. must be via API)	Each building block MUST be able to implement single-sign-on integration with 3rd party security.
	PROTECT						
10	SSL and TLS Connection Implementations	All	High	First	High	Applies to all connections throughout all components such as: *Web/Mobile UI<->API,* *Web/Mobile UI,<->Auth* *BB<->API<->BB,* *Workflow<->API*	Each building block MUST implement SSL and TLS based connections for all TCP connectivity both external to the building block, and internally between components in a selective manner depending on data requirements.
11	Data Sovereignty/Residency Controls and Hosting, Transmission, Backup and Recovery etc.	All	Medium	Second	Medium	Probably needs to be addressed during country rollouts due to data sovereignty regulations but needs to be catered for in the architecture options.	Each building block where dealing with citizens data MUST provide the ability to implement data sovereignty/residency controls, hosting, transmission, backup and recovery in compliance with specific national laws and guidelines in each country..
12	Network Security, Protocols and Firewall implementations etc.	All - unless we create a networking or cloud WG	High	First	High	Infrastructure oriented but could be addressed at least in part by software defined networking (SDN) which can be part of a modern PaaS such as OKD - TBD	Each building block shall comply with the overall secure networking architecture that will be deployed in place with each country implementation. Issues such as network security, networking protocols and firewall implementations etc. shall be defined as a part of the recommended architecture showing the various zones and separations required.
13	Application Services Security (multi-tenancy etc.)	All - unless we create a networking or cloud WG	High	First	High	Likely best implemented as a feature of the chosen PaaS framework such as OKD - TBD	The components for each building block are required to be deployed in containers according to the architecture description. The chosen container orchestration platform and PaaS solution shall provide the means to implement Application Services Security (such as multi-tenancy etc.)
14	VPN and Secure Network Access Controls	All - unless we create a networking	High	First	High	Infrastructure oriented but could be addressed at least in part by software defined networking	Each building block MUST comply with a defined set of VPN and Secure Network Access Controls. This will be based on the location of event

		or cloud WG				(SDN) which can be part of a modern PaaS such as OKD - TBD	producers and consumers on the network and hope the various segments of networking are sliced and protected. These standards are to be defined as a part of the target architecture implementation for each country.
15	Network Vulnerability Scanning	All - unless we create a networking or cloud WG	Medium	First	High	Can be sourced from open source tooling such as OpenVAS, Wireshark etc. - TBD	A suite of open source tools are to be adopted for the purposes of Network Vulnerability Scanning. These tools MUST be acquired by the project and centrally deployed in each country to ensure adequate network service security is in place.
16	Software Defined Networking and Network Slicing etc.	All - unless we create a networking or cloud WG	Medium	First	High	Infrastructure oriented but could be addressed at least in part by software defined networking (SDN) which can be part of a modern PaaS like OKD - TBD	The project MUST adopt a software defined networking solution as a part of the core deployment architecture. This can and SHOULD be implemented as a part of the chosen PaaS solution.
17	Operating Systems, Services Security and Immutability	All - unless we create a networking or cloud WG	High	First	High	Infrastructure oriented but could be addressed at least in part by OS like Centos which can be part of a modern PaaS such as OKD - TBD	The project MUST deploy all services (typically microservices) on an immutable operating system infrastructure. Typically this can and SHOULD be provided as part and parcel of the chosen PaaS solution. The reason for this is such that if a security breach does happen then the operating system running the component cannot be modified by the offender.
18	Network, Service and Transaction Observability and Visibility	All - unless we create a networking or cloud WG	Medium	Second	High	Can be addressed as part and parcel of a modern PaaS solution such as OKD including a service mesh such as ISTIO - TBD	The project MUST implement a PaaS infrastructure that supports Network, Service and Transaction Observability and Visibility for protection against flaws and faults. This is so that complex transactions involving multiple components (likely microservices) can be observed and traced for the purposes of debugging and auditing. This would typically be implemented at least in part by a service mesh feature of the PaaS along with other integrated components for visualization such as the Jaeger open source tracing facility and the Kiali open source visualizer for example.

19	Man-in-the-middle (MITM) Attack Prevention (AKA Session Hijack)	Security with impact to all WGs	Medium	Second	Low - requires multiple layers of protection and extensive education for users	Needs to handle insecure WIFI, email spam filtering, virus scanning and ad-blockers etc. at all levels.	Each building block with component connections spanning any exposed network segment MUST implement protection against Man-in-the-middle (MITM) Attack. This is particularly relevant for exposing API interfaces to 3rd parties and can be addressed for the most part in concert with the API Management and Gateway solution.
20	Cloud Platform Configuration Management and Securing Configurations	All - unless we create a networking or cloud WG	High	First	High	Can be addressed as part and parcel of a modern PaaS solution such as OKD (which has secure encrypted stores for credentials) - TBD	Each building block MUST adopt the facilities of the chosen PaaS for implementing Cloud Platform Configuration Management and Securing Configurations. For example authentication credentials for common components like databases need to be managed appropriately and simply across multiple environments through DevOps automated deployment etc.
21	Insider threats and internal audit-ability	Security	Medium	First	High	The same security protocols, data encryption, protections and monitoring to be applied consistently both internally and externally.	Each building block MUST adopt the same consistent security and privacy implementation measures to protect against Insider threats and internal audit-ability as those adopted for external exposures. This is a general statement.
22	Denial of service attack prevention	All - unless we create a networking or cloud WG	Medium	Second	High	Can be managed by implementing open source tools/services such as CrowdSec and DDOS Deflate, Fail2Ban, HAProxy, DDOSMon, NGINX etc. Note that HAProxy is built into PaaS solutions like OKD - TBD. Note that a number of API Gateway products are built around NGINX - TBD.	The project must implement an open source Denial of service attack prevention solution across all interfaces exposed to the public internet for each country deployment. This can be implemented through a reverse proxy web server environment using open source tools such as Fail2Ban, DDOS Deflate or HAProxy.
23	API Endpoint Security Policy Management and Gateway Services	Security (with impact to all others)	High	First	High	**The implementation of a centralized API Management and Gateway solution is probably not negotiable**. All API interfaces both internal and external must be managed	The project MUST implement centralized API Endpoint Security Policy Management and Gateway Services (both internal and external). The reason for this is to implement a consistent layer of

	(both internal and external)					through this facility. Architecture will likely require a separate gateway for both internal and external services.	security for API interfaces that can be both managed centrally and alleviate the service developers from the implementation complexity of security. This can and SHOULD be implemented through an open source API Management and Gateway services product.
24	Virus/Malware and Ransomware Attack Prevention and Detection	Security (with impact to all others)	High	Second	Medium	Requires extensive and probably commercial software in many layers.	The project MUST implement protection from and detection of Virus/Malware and Ransomware Attack. This likely requires a commercial solution suite as open source offerings are insufficient and not suitable for massive country-wide deployments w=such as GovStack.
25	Credential Theft Prevention	Security (with impact to all others)	High	First	High	Can be addressed as part and parcel of a modern PaaS solution such as OKD (which has secure encrypted stores for credentials). Requires implementation through all development procedures and CI/CD etc. - TBD	The project MUST implement credential theft prevention as part and parcel of its selected PaaS infrastructure. This is essentially an encrypted keystore that can host sensitive credentials and provide access to them with policy based security.
26	SQL Injection Attack Prevention	Security (with impact to all others)	High	Second	High	Can be addressed through open source tools such as those provided by the OWASP Foundation. Plugins for OWASP tools are available for PaaS solutions like OKD	The project MUST implement SQL Injection Attack prevention as part and parcel of any and all applications development across building blocks. This can be implemented through open source tools such as those provided by the OWASP Foundation. This can and SHOULD be implemented as plugins through the PaaS solution and fully integrated into the DevOps toolchains for the project build and deployment.
27	Containing, Managing and Mitigating Hardware/Firmware	Security	High	Second	Low-Medium	Typically these types of attacks are best mitigated by the use of commercial open source subscriptions as they close the window of vulnerability. New CVE's happen every	The project MUST implement solutions for Containing, Managing and Mitigating Hardware/Firmware Vulnerabilities and Known Exploits (directory traversal, rowhammer, spectre,

	Vulnerabilities and Known Exploits (directory traversal, rowhammer, spectre, meltdown, LazyFP etc.)					month and are becoming more common as IT advances. The upstream open source projects like OKD will eventually get the patches but it will be some time after commercial open source product patches are released to those with enterprise subscriptions.	meltdown, LazyFP etc.). This can and SHOULD be implemented through plugins to the PaaS solution and DevOps toolchain for the project to ensure that every single component deployed is **scanned** for known CVEs.
28	Data Privacy/Loss/Leakage/Confidentiality (individuals and organisations)	Security (with impact to all others)	High	Second	Medium	For the most part this involves end-point protection. Some tools are available in open source but it requires multiple layers of implementation and is thus more expensive.	The project SHOULD implement solutions for prevention against Data Privacy/Loss/Leakage/Confidentiality (individuals and organisations). This likely requires expensive commercial packages rather than open source tooling.
29	Data Security (at rest and in transit - i.e. encryption and obfuscation etc.)	Security (with impact to all others)	High	First	Medium	Needs to be considered for each data store and each connection in the architecture. It is a very broad topic that must be addressed in the context of the data confidentiality requirements for each country implementation. Will be relatively expensive for resource-limited settings but it is a necessity.	The project MUST implement solutions for Data Security (at rest and in transit - i.e. encryption and obfuscation etc.) consistently across all datastores and connections within and surrounding all building blocks.
30	Cross-site Scripting (XSS) Attack Prevention	Security (with impact to all others)	Medium	First	High	By virtue of the fact that each BB will host its own UI there is strong potential for cross-site scripting vulnerabilities. Rules must be adhered to by developers… for example all DOM based XSS reflection or embedding must be performed on the server side not in the ECMA layer. Several other rules must also be implemented for developers to reduce the likelihood of XSS vulnerabilities. Many of these rules can be found here on the OWASP web site: https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html	The project must provide a consistent set of rules for cross site scripting across all building blocks to ensure that it is not exposed to Cross-site Scripting (XSS) Attacks. This is a complex area that must be addressed during development and testing. Details of many of the rules that must be implemented can be found on the OWASP web site.
31	Replication and Perimeter/Edge	Security (with	High	Second	Medium	This is complex and applies to large architectures with multiple data centers and perimeter services. Every single element of the	The project MUST provide Replication and Perimeter/Edge Data and Services Security. This assumes that practically all national deployments

	Data and Services Security	impact to all others)				networks and trusts must be examined for vulnerabilities. There is no simple one-shot solution for this as it involves multiple data centers and multiple layers communicating and replicating potentially sensitive data. The degree to which this is addressed is commensurate with the relative sensitivity of the data and services involved.	will have multiple sites that must be protected from breaches and leakages as a result of technology services deployed to distribute and consolidate data.
32	Social Media, Social Network and Social Engineering Threats and Prevention	Security (with impact to all others)	High	Second	Medium	This is a complex subject that requires not only technical intervention but extensive training for everyone involved in the processes of eGovernment.	The project MUST provide protection from Social Media, Social Network and Social Engineering Threats. This is not only a technology services issue but also applies to standard operating procedures and requires extensive user education.
33	Centralized PAAS Management, Monitoring and OTA Automated Update	Security - unless we create a cloud or networking WG	Medium	First	Medium	Can be addressed as part and parcel of a modern PaaS solution such as OKD - TBD (which has over the air update capabilities along with centralized management and monitoring).	The project MUST provide Centralized PAAS Management, Monitoring and OTA (over-the-air) Automated Update for infrastructure and applications components. This is to ensure that patches to emerging and common vulnerabilities are addressed in the smallest window possible and the whole architecture is not exposed through such vulnerabilities. This can and SHOULD be addressed as part and parcel of the selected PaaS infrastructure.
34	Digital Service Registry Security	Registration and Security	High	First	High	Can be addressed as part and parcel of a modern PaaS solution such as OKD - TBD (which has an embedded services registry) - there may be other registries required that are also impacted. There needs to be careful delineation of functionality and terminology used for registries as there are many implications in PaaS environments such as for service exposure through software defined networks etc.	The project and all building blocks utilizing registries of any kind (particularly digital service registries) MUST provide Digital Service Registry Security. This means ensuring that the protocols, interfaces and connections to such centralized services are controlled in accordance with the other requirements for connections and API etc.

35	Surrounding Networking Software Infrastructure Security (DNS, DHCP , PXE, BootP services etc.)	Security - unless we create a cloud or networking WG	High	First	High	Each core network service must be addressed in its own right. There is too much to discuss here but these services are critical, exposed, prone to vulnerabilities and must be secured with the highest possible standards applied.	The project MUST address the general Surrounding Networking Software Infrastructure Security (DNS, DHCP , PXE, BootP services etc.) for each and every country rollout. These services are particularly vulnerable as some of them are exposed to insecure zones (especially DNS).
36	Cloud Provider Infrastructure Security (hardware layer, infrastructure layer, virtualisation, container and platform layer, application layer)	Security - unless we create a cloud or networking WG - then both	High	First	High	Each cloud service must be addressed in its own right. There is too much to discuss here but these services are critical, exposed, prone to vulnerabilities and must be secured with the highest possible standards applied.	The project MUST provide protection against common vulnerabilities in Cloud Provider Infrastructure Security (hardware layer, infrastructure layer, virtualisation, container and platform layer, application layer etc.). This is specific to each public cloud provider where utilized but a common source of threats since they involve complex suites of services that are stitched together (most often by the implementer not the cloud provider) in multiple layers to form the solution architecture.
37	Mobile and Wireless Networking Security/WIPS	Security - unless we create a cloud or networking WG - then both	High	First	High	Mobile and wireless networking security is a significant issue to deal with given that there will likely be mobile applications deployed as a part of the architecture. A plethora of issues abound including for example Evil Twin Attack, WarDriving (piggyback attack), sniffing and shoulder-surfing etc.. CISA has released a short guide here for securing wireless in general: https://us-cert.cisa.gov/ncas/tips/ST05-003	The project MUST provide protection against Mobile and Wireless Networking Security/WIPS vulnerabilities. There are many potential and known vulnerabilities around these networks which have predicated information security in the digital age. The potential for eavesdropping and information leakage as well as other forms of hijacking attacks is very broad as the exposure is over-the-air.
38	Compressed and Encrypted Information Transmission via Messaging and Email etc. to external or internal 3rd parties along	Security - unless we create a cloud or networking WG - then both	High	Second	Medium	This is principally the same as endpoint security to prevent information leakage.	The project (including all building blocks MUST provide protection against private information leakage through Compressed and Encrypted Information Transmission via Messaging and Email etc. to external or internal 3rd parties along with any other potential channels for critical private information leakage.

	with any other potential forms of critical information leakage.						
39	Anti-Phishing and Anti-Spam	Security (with impact on all others)	High	First	Medium	Phishing and the associated spam are two of the most common digital platform security problems and must be dealt with. A number of open source solutions exist such as OrangeAssassin, MailScanner and Apache SpamAssassin. These are commonly used by many commercial sites all around the world.	The project MUST provide Anti-Phishing and Anti-Spam tooling. These are some of the most common sources of security issues and can be dealt with using open source tools such as Orange Assassin, MailScanner and SpamAssassin.
40	Physical Security and Access Controls to Facilities	Security	High	First	Medium (expensive devices, procedures and lots of manpower)	Physical security is a must for all on-premise facilities and almost goes without saying. The extent to which physical security is implemented must comply with national government hosting standards in each country.	The project MUST provide physical security measures for access to physical facilities.
41	Portable and Removable Media Controls	Security (with impact on all others)	High	First	High	This is commonly known as endpoint security and must be dealt with comprehensively either at hardware level (by disconnection) or in software that allows policy and procedure for removable media to be controlled.	The project MUST provide portable and removable media controls to protect against information leakage.
42	Backup Security and Backup Information Controls	Security (with impact on all others)	High	First	Medium	This is a complex and diverse area to address as there will be several data repositories and databases in the deployed infrastructure including both CUI and regular non-CUI information. Backups are one of the areas that create large exposure for information loss and must be addressed consistently and thoroughly. This must include encryption of backups and the physical security of backups as well as the policy, procedure and controls to ensure that leakage risk is minimized.	The project must provide backup information controls and security to prevent information leakage and tampering etc. through the backup and recovery processes. This applies to all building blocks.
	DETECT						

43	Vulnerability Management and Security Automation	Security	Medium	Second	High	Several open source and commercial tools are available to address this in terms of scanning for vulnerabilities in all layers of the solution including containers for example. The process and tools for this need to be addressed specifically depending on the technical components of the final solution architecture.	The project MUST provide tools to detect, manage and automate the resolution of common vulnerabilities in all layers (applications components down to infrastructure components) before they eventuate as deployed in the solution. There are many open source scanning tool options available for this purpose. CISA defines the standards for this and refers to many of the available open source tools.
44	Applications Development, Deployment, DevSecOps and Container Image Security	Security	High	First	Medium	This needs to be built-in to the CI/CD process to ensure that only secured images are deployed to production. Most modern PaaS offerings such as OKD - TBD are also shipped with a rudimentary suite of tools to accomplish this.	The project MUST provide a secure DevSecOps process for code deployment. This applies to areas surrounding Applications Development, Deployment, DevSecOps and Container Image Security scanning (i.e. what's inside the container) before applications components are deployed.
45	Compliance Checking and Scanning (PCIDSS/HIPAA, CIS etc.)	Security	Medium	Second	Low-Medium (can be expensive)	These are global standards for security compliance checking and several tools are available in the market to address this. Both commercial and open source options are available.	The project MUST provide tooling support for Compliance Checking and Scanning (PCIDSS/HIPAA, CIS etc.). A number of open source tools and commercial off-the-shelf tools are available for this purpose. This compliance checking MUST be conducted on a regular basis for all building blocks dealing with financial, healthcare and personal data in accordance with the aforementioned PCIDSS, HIPAA and CIS standards.
46	Security Risk Profiling	Security	Low	Second	Low-medium	Several tools are available in this space with varied ways of addressing the concerns including threat modelling	The project SHOULD provide tools for Security Risk Profiling. Several such tools are available and offer assessment of security risks through techniques such as threat modelling.
47	Threat/Intrusion Detection and Prevention	Security	High	First	High	Several open source tools are available to address this space admirably including for example Snort, Bro, Kismet, OSSEC and Open DLP etc. These are VERY effective, VERY mature and easily adopted.	The project MUST deploy tooling for Threat/Intrusion Detection and Prevention in the infrastructure and other layers. Several such tools are available in open source (Snort, Bro, Kismet, OSSEC etc.)

48	Login Notifications and Alerts etc. (new device or IP etc.)	Security	Medium	First	High	This should be managed with the basic user device profile and an alert sent via email and other channels whenever a login is made from a new endpoint, giving the user the option to lock the account.	The project across all building blocks SHOULD provide Login Notifications and Alerts etc. (new device or IP etc.) where email or other notifications would be sent to recipients with warning messages when authentication is performed from a new device. This also involves keeping a registry of registered devices for each user/party/actor (albeit internal or external and the ability for them to lock their account if the authentication was achieved by an unknown party.
49	Open Source Intelligence (OSINT) Platforms/Tools and Processes	Security	Low	Second	High	OSINT gathering is COMPLEX. The following article describes the origins and tools of OSINT that have evolved from the original Metasploit (white hacking solution): https://www.csoonline.com/article/3445357/what-is-osint-top-open-source-intelligence-tools.html Essentially OSINT puts the hackers' tools into the security analysts protection arsenal.	The project SHOULD provide Open Source Intelligence (OSINT) Platforms/Tools and Processes in order to perform security assessments on open source usage. This essentially incorporates the hackers tools into the protection and detection arsenal (see the evolution of Metasploit and other white hacker solutions in open source).
50	Information gathering and Security Data Visualization	Security	Low	Second	Low	This is an emerging area of value that largely comprises of the following types of visualizations but there are no comprehensive open source tools available: Perimeter Threat Network Flow Analysis Firewall Visualization IDS/IP S Signature Analysis Vulner ability Scans Proxy Data User Activity Host-based Data Analysis	The project SHOULD provide tooling for Information gathering and Security Data Visualization for maintaining security observability through operations. Many different visualizations are available through various commercial tools but no comprehensive open source solution is available.
51	Fraud Detection and Management		High	Second	Medium	Fraud detection and management is the set of activities carried out to check money or property from being acquired through false pretenses is known as fraud detection and the ability to trace and manage incidences of	The project MUST provide tooling for Fraud Detection and Management. This is not negotiable and many open source tool sets currently exist such as FraudLabs, Fraud.Net and MISP. This applies to all building blocks dealing with financial

						fraud. In many industries like banking or insurance, fraud detection is applied. SInce this project involves money exchange there is a case for fraud detection and management. Number of open source tools including the following are available: FraudLabs Pro, Fraud.Net, MISP, pipl, Sift etc.. See this article for a deeper description: https://www.goodfirms.co/blog/best-free-open-source-fraud-detection-software	transactions and information (such as payments - which appears to be bidirectional… i.e. govt-to-recipient and payee-to-govt).
	RESPOND						
52	Incident Response and Management (ticketing etc.) - applies to attempted and successful intrusion, fraud, hacking, phishing incidents and all other forms of security incident	Security (with impact on all others)	High	Second	Medium	Being prepared is key to responding to security incidents in an accurate and level-headed manner. When disaster hits, there's an ocean of difference between responding to incidents using calculated clear steps and plunging head first into reactive mode. Several open source tools exist for managing response to security incidents such as: Cynet 360, GRR Rapid Response, Alien Vault, Cyphon, Volatility, Sift, The Hive and others. See this article for a deeper analysis: https://www.cynet.com/blog/the-7-best-free-and-open-source-incident-response-tools/	The project MUST provide Incident Response and Management (ticketing etc.) - This applies to both attempted and successful intrusion, fraud, hacking, phishing incidents and all other forms of security incident. This applies across all building blocks and must be built-in to the infrastructure and processes of each building block when any incident is detected.
53	Security Sandbox Solution - used to test responses to potential/predicted security incidents	Security (with impact to all others)	Low	Second	Low	This would involve creating a sandbox environment to test and resolve security issues thus requiring a complete sandbox of all the security tools mentioned here.	The project MUST provide a Security Sandbox Solution - used to test responses to potential/predicted and actual security incidents.
	RECOVER						
54	Critical digital infrastructure business continuity considerations (terrorism, sabotage, information	Security (with impact to all others)	High	First (business continuity is a must do)	Low-Medium (as it normally requires whole data center	This is more of a planning and execution exercise and simply must be built-in to the overall deployment game plan for every single piece of software infrastructure and data infrastructure along with the processes and procedures as well as test recoveries to ensure	The project MUST deal with Critical digital infrastructure business continuity considerations (terrorism, sabotage, information warfare, natural disasters etc.) - i.e. provide the technical ability and processes required in order to recover the complete digital infrastructure. This applies to all

	warfare, natural disasters etc.) - processes and how to recover digital infrastructure.				investments)	that an adequate recovery response can be attained on demand..	building blocks and must also endure recovery testing on a regular basis.
55	Specifically Security Related Concerns surrounding BIA/DRP/BCP (disaster recovery, business continuity etc.) - how to recover to specific data versions using logging tracking and tracing information to determine the best path.	Security (with impact to all others)	High		Medium (considering the above)	This is similar or the same as the concern above and simply elaborates it further.	The project MUST deal with Specifically Security Related Concerns surrounding BIA/DRP/BCP (disaster recovery, business continuity etc.) - what this means is how to recover to specific data versions using logging, tracking and tracing information to determine the best recovery path. This also covers the security of the backups themselves to prevent fraud, tampering and information leakage during storage or recovery for example and MUST address the exact data security requirements stipulated throughout this document but in the context of backups.
56	Cloud Security Posture Management (automation of identification and remediation of risks with public cloud services)	Security (with impact to all others)	High	Second (for public cloud deployments)	Low-Medium	This is a more advanced and aggregated way of determining the overall security posture in respect to public cloud based deployments and takes into account all of the risks. There are a number of open source solutions available such as OpenCSPM (cloud security posture management). Really only applicable with deployments on public clouds but becoming essential.	The project SHOULD implement Cloud Security Posture Management (automation of identification and remediation of risks with public cloud services). Open source tools such as OpenCSPM are available.
57	Digital Service Registry State Recovery	Security/Registry	High	First	High	It seems the concept of registry is morphing into something more generic than it was originally (which seemed to be a service registry). The state recovery for this is contingent on the technology solution that the Registration BB takes. Losing the state of this registry due to a security issue is a key risk that MUST be mitigated.	The project MUST provide the ability for specific Digital Service Registry State Recovery (point-in-time). Registries are one of the most likely early targets of cyber-criminals. This applies predominantly to the Registry building block.
58	Controlled Unclassified Information (CUI)	Security (with	High	Second	Medium	This is all about how information is controlled throughout systems based on its secrecy/privacy classification.	This is a bit more general but the project SHOULD provide the ability to manage and recover Controlled Unclassified Information (CUI)

	Registries, Repositories and Processes (i.e. marking, safeguarding, transporting, disseminating, reusing and disposing of controlled unclassified information)	impact on all others)				We have made an assumption that GovStack will only ever have to deal with what is known as CUI (Controlled Unclassified Information) as opposed to CI (Classified Information - which is the type of information managed by security agencies such as the CIA). CUI MUST be managed in accordance with NIST SP 800-171 Rev. 2	Registries, Repositories and Processes (i.e. marking, safeguarding, transporting, disseminating, reusing and disposing of controlled unclassified information). This is to be in accordance with NIST 800-171 Rev.2 (see References and Standards). This applies to all building blocks that deal with CUI (usually information collected by govt and security agencies) which is likely to also be specific to country implementation.
59	Controlled Unclassified Information (CUI) domain isolation (isolation for sub-networks and security domains etc. handling CUI)	Security (with impact on all others)	High	Second	Medium (as it requires physically separated infrastructure)	See above - requires a physically separate domain for hosting such information.	: This is related to the above but the project SHOULD provide Controlled Unclassified Information (CUI) domain isolation (isolation for sub-networks and security domains etc. handling CUI).

Each specific security related concern or issue outlined in the table above must address the following high level requirements:

5 Cross-Cutting Requirements

Note that all of the requirements stipulated in this document and its references are reciprocal in that they also apply to components such as the API Management and Gateway services implemented by the security building block. For example the API Management and Gateway services deployed by this building block MUST also address their own intrusion prevention and detection needs referencing the solution requirements defined by this document.

The requirements stipulated in this document are themselves cross-cutting in that they apply to all building blocks and MUST be cross-referenced by the Building Block Definitions for each building block in the Cross-cutting requirements section.

Having these cross-cutting requirements defined centrally in this document and its references removes the issues of inconsistent, insufficient, costly and repetitive security implementation across all building blocks.

The cross-cutting requirements described in this document, its references and this section are an extension of the high level cross-cutting requirements defined in the architecture specification document and intended to specifically define the security requirements for the whole GovStack architecture in all layers.

This section describes the additional cross-cutting requirements that apply to the security building block as well as cross-cutting security requirements for ALL other building blocks. Note that cross-cutting requirements defined here use the same language (MUST or SHOULD) as specified in the architecture blueprint document (see Ref 1).

5.1 Privacy

Personal data MUST be kept private and never shared with any parties, except where specific authorization has been granted. This also applies to all acquired security components as they will often be logging personal data along with the transactional records. That logging data must also be considered private. Where CUI (Controlled Unclassified Information) is dealt with, the *NIST 800-171 Rev 2 standard shall be applied (see Ref 3)*

5.2 Audit Logging

Logs MUST be kept in files and printed to stderr/out when containerized. All records that are created, updated, or deleted. Logs MUST include timestamps and identify the user and affiliation that performed the transaction. It is expected that each component acquired to address each security concern/issue SHALL include capabilities for comprehensive auditing and access logging.

5.3 Source Code

Source code SHOULD be available and easily accessible for any custom components but in the large it is expected that security components will be acquired, configured and deployed not built from scratch.

5.4 Security Requirements

Must refer reciprocally to this document and its references.
Security requirement is a condition over the phenomenon of the environment that we wish to make true by installing the system in order to mitigate risks. A requirement defining what level of security is expected from the system with respect to some type of threat or malicious attack.

5.5 Digital ID/Certificate Functional Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.5.1 Enrollment Services Enrollment services for a digital ID in the form of a certificate using the physical credentials of the enrollee (a human citizen subject) and the process of the Identity BB (see the functional requirements for Identity in the Identity BB Definition). A feature for invalidating, locking or disenrollment/revocation of the digital ID shall also be provided as a response measure to both human citizen subjects leaving the system and responding to security breaches encountered. Digital certificate enrollment must be provided by the solution but is not required for every human citizen subject (see below). *Note that it is anticipated that the Identity BB will call this feature either directly via API or indirectly via the IAM features of the Security BB for users electing to use a digital ID consisting of certificates as a part of the account provisioning process. The digital ID will then be stored with the physical ID records in the identity BB and sent to the new user via secure means (probably installed on their device).* *Note that simple numerical digital IDs will also be supported for human citizen subjects as an option where users are unable to leverage certificates based digital ID. The requirements governing this are to be stipulated by the Identity BB (see the Identity BB Definition) .* *Note that 3rd party organization and internal subjects (both human and non-human) MUST be issued valid signed digital certificates in order to establish and maintain secure inter-organization and internal communications.*	MUST	Building time

5.5.2 Multi-Factor Authentication The overall solution suite shall also be able to implement multi-factor authentication using simple numeric digital IDs for human citizen subjects such as their tax file or social security number of the user. A selection of various alternatives for digital ID is required in order to cater for more or less digitally-savvy citizens. Various token types are also required to be optimally supported such as HOTP and TOTP tokens, SMS, email, push notifications, SSH keys, X.509 certificates, Yubikeys, Nitrokeys, U2F and WebAuthn. Vendors of solutions SHOULD articulate the benefits of what they propose in their solution. *Note that multi-factor authentication must be able to be implemented for both external and internal subjects (people, systems, components etc.) but is not necessarily required for internal non-human subjects (such as building block components) as they communicate via the information mediator BB (see the InfoMed BB Definition).*	MUST	Deployment time
5.5.3 Numerical Digital ID Attribute Where human citizen subjects adopt the use of a simple numerical digital ID, the multi-factor authentication process MUST include a time-sensitive credential (AKA OTP or one time PIN)	MUST	Both
5.5.4 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.6 Certificate Authority Functional Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.6.1 Strong Authentication and Cryptography The basic security service requirements of high confidentiality, high integrity, strong authentication, strong cryptography and absolute non-repudiation must be delivered by the solution. The vendor must articulate how these needs are met with the proposed solution suite to ensure that GovStack is able to deliver the same consistent security service experience.	MUST	Both
5.6.2 Standards Based Certificate Authority Server A certificate authority (CA), or certifier with fully configured server infrastructure, is required to implement trusted administration tools that issue signed digital certificates to citizens and other 3rd parties then maintain the lifecycle of those digital certificates. Digital certificates MUST comply with the IETF, ISO/IEC/ITU-T X.509 Version 3 and PKIX (note that PKIX is the registration authority role, which allows administrators to delegate the certificate approval/denial process) standards as defined by RFC5280 and associated standards. Digital certificates are to be issued on behalf of the appropriate government authority where GovStack is to be deployed.	MUST	Building time

5.6.3 Certificate Issuance Issued signed digital certificates MUST verify the identity of an individual, a server, or an organization and allow them to use SSL to communicate and to use S/MIME to exchange mail as well as sign documents and transactions in a non-repudiable manner.	MUST	Building time
5.6.4 Digital Signatures Certificates issued by the authority MUST be stamped with the certifier's digital signature (i.e. signed), which assures the recipients of the certificate that the bearer of the certificate is the entity named in the certificate.	MUST	deployment time
5.6.5 Private Certifier Capability The solution provided MUST be able to be set up as a certifier to avoid the expenses that a third-party certifier charges to issue and renew client and server certificates. In other words the solution can operate without a 3rd party certifier. This makes it easier, cheaper and quicker to set up and deploy new certificates as needed and at scale. Certificate validation will not be required to access a 3rd party certifier for validation.	MUST	Both
5.6.6 Revocation Lists Support The certificate server MUST be able to support certificate revocation lists (CRLs), which contain information about revoked or expired Internet certificates.	MUST	Both
5.6.7 Flexible Certificate Authority Hierarchy The certificate authority server infrastructure MUST enable CA administrators to create a flexible private CA hierarchy, including root and subordinate CAs, with no need for external CAs.	MUST	Building time

Private CA hierarchies must be able to be built in a hybrid mode, combining online and on-premises CAs with cloud based CAs.		
5.6.8 Web Based Admin Interface The certificate authority server infrastructure MUST provide a comprehensive web based administrator user interface so that all of the GovStack certificate issuance and revocation features and functions can be configured and managed from a single central window.	MUST	Building time
5.6.9 Standards Based API Interface The certificate authority server infrastructure must provide a secure API interface that supports calls for the issuance and revocation of certificates by other GovStack components such as the Identity Building Block. This API must comply with the same OpenAPI standards defined in the Architecture Blueprint (see Ref 1)	MUST	Building time
5.6.10 High Availability The certificate authority server and its infrastructure must be configurable for highly available implementation (see the non-functional definitions for high availability). For example this means clustering and failover of certificate authority services and associated data sources to provide a 24x7x365 service with 99.99% availability (AKA 4 nines)..	MUST	Building time
5.6.11 Horizontal Scalability The certificate authority infrastrastructure must be horizontally scalable on commodity hardware to ensure that the scalability needs of low resource countries with large populations deploying GovStack can be met without incurring significant or untenable costs.	MUST	Building time

5.6.12 Advanced Encryption Methods Support The solution provided MUST support advanced encryption techniques such as ECC (Elliptic Curve Cryptography) which gives certificates an additional security/performance advantage vs use of the traditional RSA cryptography system for example. Other techniques may be acceptable and the vendor must explain and justify why they are superior.	MUST	Both
5.6.13 Enrollment via API Interface (SCEP) The solution MUST provide a means of enrollment via a standardized API interface which SHOULD be based on SCEP. A description of the OpenXPKI enrollment workflow and API can be found here: https://openxpki.readthedocs.io/en/latest/reference/configuration/workflows/enroll.html. This is an example only and can vary based on the proposed implementation.	MUST	Building time
5.6.14 Security Provisions The certificate authority deployment scheme must be exposed to the public internet and protected securely in accordance with all of the other security requirements and provisions described in this document. The reason for this is to allow 3rd parties to verify the authenticity of certificates issued by the govts deploying GovStack.	MUST	Deployment time
5.6.15 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.7 Credential Storage (i.e. LDAP) Functional Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.7.1 Centralized Credential Store The GovStack security solution requires a credential store as a centralized infrastructure for hosting the user account and credentials defined such that the IAM solution and other components such as the API Management and Gateway solutions can leverage them. This may end up being embedded in other solutions such as IAM or potentially implemented as a separate repository such as LDAP.	MUST	Building time
5.7.2 Web Based Admin Interface The solution provided as a credential store MUST have a comprehensive web based administrative interface that allows administrators to make any necessary configuration changes and modify credentials for subjects as needed.	MUST	Deployment time
5.7.3 High Availability The solution provided as a credential store and associated components providing access to the store MUST be highly available and utilize clustering technology in order to provide a minimum of 24x7x365 service with 99.99% availability (AKA 4 9's).	MUST	Building time

5.7.4 Standards Based REST API The solution provided MUST include a standard API for storage and access to credentials such as the standard REST LDAP API provided by the open source 389 Directory Server here: https://directory.fedoraproject.org/docs/389ds/design/ldap-rest-api.html#ldap-rest-api . Note that this is purely an example and may vary based on the solution proposed.	MUST	Building time
5.7.5 Standard Connectors and Adapters The solution provided as a credential store must be fully integratable with the other security solution components through standards based protocols and out-of-the-box adapters for the specific product offered.	MUST	Building time
5.7.6 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.8 Time Sensitive Credential (i.e. OTP) Functional Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.8.1 OTP and Multifactor Capability The solution must provide the ability to generate and utilize time sensitive credentials in various forms for the purposes of securing user authentication with multiple factors using non-PKI credentials (see the section of digital ID in this document).	MUST	Building time
5.8.2 Multiple OTP Methods Multiple methods SHOULD be provided for the implementation of time-sensitive OTP using potentially push or device level sources. *Note that vendors SHOULD articulate the benefits of their technology and approach to implementing time sensitive credentials and align their recommendations to the needs of resource limited settings.*	SHOULD	Deployment time
5.8.3 High Availability The solution provided as an OTP server and any associated components MUST be highly available and utilize clustering technology in order to provide a minimum of 24x7x365 service with 99.99% availability (AKA 4 9's).	MUST	Building time

5.8.4 Standards Compliance The solution offered MUST comply with at least one of the common OTP related IETF standards such as , RFC 1760 (S/KEY), RFC 2289 (OTP), RFC 4226 (HOTP) and RFC 6238 (TOTP).	MUST	Building time
5.8.5 Standards Based REST API The offered solution MUST provide a REST API for managing OTP similar to the following: https://www.miniorange.com/step-by-step-guide-to-set-up-otp-verification. This is just an example and can vary with the proposed implementation.	MUST	Building time
5.8.6 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

GovStack

5.9 Network Scanning and Vulnerability Management Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.9.1 Network Policy Definition and Scanning The solution offered MUST provide advanced policy definition capabilities along with the ability to scan entire networks and subnetworks with one click and the ability to automate scanning on a regular schedule.	MUST	Building time
5.9.2 Broad Suite of CVE Scan Coverage The solution MUST provide the broadest possible suite of CVE (known vulnerability) scans across common ports for common software services and the like.	MUST	Deployment time
5.9.3 Regular CVE Pattern Updates The solution must provide regular updates and new plugins for emerging CVEs within a short timeframe of the CVE becoming known.	MUST	Deployment time
5.9.4 List of Top Threats and Remediations The solution MUST be able to assemble lists of top threats from scans, based on VPR and provide recommendations on which vulnerabilities pose the greatest risk in order to prioritize remediation efforts.	MUST	Both

5.9.5 Broad Range of Preconfigured Templates The solution MUST provide preconfigured templates out-of-the-box for a broad range of IT and mobile assets. These must support everything from configuration audits to patch management effectiveness which helps quickly understand where vulnerabilities exist and assess audit configuration compliance against CIS benchmarks and other best practices.	MUST	Building time
5.9.6 Customizable Views The solution MUST provide the ability to easily create reports based on customized views, including specific vulnerability types, vulnerabilities by host or by plugin. MUST be able to create reports in a variety of formats (such as HTML, csv and XML) and then easily tailor and email reports to stakeholders with every scan.	MUST	Building time
5.9.7 Associative Remediation The solution SHOULD provide associative remediation via security patch automation using automation tools so that hundreds or even thousands of specific vulnerability instances can be addressed across the whole infrastructure.	SHOULD	Deployment time
5.9.8 Standards Compliance Management The solution MUST provide the general ability to implement vulnerability management processes that drive compliance with PCI, HIPAA, GLBA, CIS, NIST and or similar European or African continental standards.	MUST	Building time
5.9.9 Scalability The offered solution MUST be able to scan whole large networks of computers with thousands of open ports and services within an acceptable time frame (the usual maintenance window). The vendor is to explain the	MUST	Building time

scaling strategy and how it can be used to address a significant eGovernment infrastructure that serves millions of citizens..		
5.9.10 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.10 Virus, Ransomware, Malware, Spam, Phishing Protection Requirements

Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both

5.10.1 Transparent Gateway Service The solution MUST provide an anti-spam mail gateway that transparently operates within the email routing infrastructure using standards based protocols such as SMTP, POP and IMAP.		Deployment time
5.10.2 Common Vulnerability Scans The solution must be able to scan emails for various types of malware and for spam, phishing, and other types of malware common attacks that target known system infrastructure vulnerabilities. The solution must be extensible (perhaps by plugin) to support emerging threats, new vulnerabilities and new services.	MUST	Deployment time
5.10.3 Cloud or On-Premise Deployment The solution MUST provide a cloud-based or on-premise based pre-perimeter defense against spam, phishing emails and virus-infected attachments.		Deployment time
5.10.4 3rd Party Virus Checker Support The solution provided MUST support a wide range of 3rd party and open source virus checker software which is independent of the mail scanner module.	MUST	Deployment time
5.10.5 Wide Range of Filtering Approaches The solution provided MUST support a wide range of filtering approaches and analytic tests such as text analysis, DNS blacklists, collaborative filtering databases and Bayesian filtering.		Both

5.10.6 Common Infrastructure Deployments The solution MUST be deployable within common open source mail server infrastructures such as procmail, qmail, Postfix, and sendmail	MUST	Deployment time
5.10.7 Ability to Integrate in Multiple Points The mail scanning modules of the solution MUST be able to be integrated at any place in the email stream.		Deployment time
5.10.8 Multiple Analytic Techniques The solution MUST provide multiple analytic techniques, as well as in-depth human expertise, to score incoming email attachments as good, bad, or unknown	MUST	Deployment time
5.10.9 Attachment Containment The solution MUST run unknown attachment files in containment which is a completely virtual environment and isolated from other network segments.		Deployment time
5.10.10 Day Zero Infection Prevention The solution MUST provide the ability to protect from "day zero" infections by rapidly responding with automated updates to counter newly identified threats and applying pattern based algorithms to detect new threats before they infiltrate systems. *Note: vendor to explain the value proposition of what they offer in this area.*	MUST	Deployment time

5.10.11 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.		Building time

5.11 Denial of Service Attack Prevention Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.11.1 Application Layer Protection The proposed solution MUST protect against application layer DDOS attack: Application Layer DDOS attack is a type of DDOS attack which targets the application layer of OSI model (i.e. the protocols that interface software modules such as POP, IMAP, SMTP, HTTP etc.). The size of these attacks are typically measured in requests per second (RPS) and limits must be configurable for both the singular IP addresses and subnets from which such traffic originates.	MUST	Building time
5.11.2 Protocol Style Attack Prevention The proposed solution MUST protect against Protocol style DDOS attacks: Protocol style DDOS attacks target server resources rather than bandwidth through saturation of requests synch as TCP SYN for connection attempts and general UDP frames in order to render the target services useless for users.	MUST	Both

The size of these attacks are typically measured in protocol frames per second (PFPS) and limits must be configurable for both the singular IP addresses and subnets from which such traffic originates.		
5.11.3 Volume Based Attack Prevention The proposed solution MUST protect against volume based DDOS attack: Volume based DDOS attack uses a variety of different techniques to saturate bandwidth of the attacked site, so other visitors cannot access it. It eventually leads the server to crash due to traffic saturation. There are three ways the solution MUST defend against volume based DDOS: 1) Attack Prevention and Preemption: this is before the attack based on detection of patterns. 2) Attack Detection and Filtering: This is performed during the attack and packets are filtered or dropped in order to preserve system integrity and 3) Attack Source Blacklisting: This can be performed during and after the attack.	MUST	Deployment time
5.11.4 White Lists The proposed solution MUST support a white-list of addresses that will always be passed to the servers. This is to facilitate normal usr operations and reduce the likelihood of false positives.	MUST	Deployment time
5.11.5 Automatic Unblocking Blacklisted or blocked IP addresses MUST be able to be automatically unblocked by the solution after a configurable timeout period.	MUST	Deployment time
5.11.6 Remediation Script Hooks The solution MUST provide the ability to run hooks for remediation scripts at event edges or at defined intervals for the purpose of cleaning up server resources and ensuring system stability etc.	MUST	Deployment time

5.11.7 Alerting Framework The solution MUST provide an alerting framework (via email and/or instant messaging) when IP addresses are blocked so that administrators are kept abreast of potential attacks and can begin monitoring activity more closely with a view to manually intervene if necessary.	MUST	Deployment time
5.11.8 Standard Firewall Integration The solution MUST support and integrate with typical Linux firewall technologies such as APF (advanced policy firewall), CSF (config server firewall) and standard Linux iptables having the ability to insert and adjust rules into the firewall on the fly to cater for attack responses and remediation.	MUST	Both
5.11.9 TCP Kill The solution MUST provide the ability to kill TCP request processes upon encountering flooding in order to preserve the integrity of the protocol stack and return the system to a normal state where it is able to process TCP requests.	MUST	Deployment time
5.11.10 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.12 Applications Development Vulnerability Prevention Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.12.1 OWASP Compliance The custom developed applications solutions MUST take steps to protect against the top OWASP vulnerabilities such as XSS for example. Development processes MUST implement automated tooling to check code for such vulnerabilities.	MUST	Building time
5.12.2 OWASP Source Code Scans The solution provided MUST be able to scan source code committed to repositories by developers to identify and remediate the top OWASP vulnerabilities. Security hotspots pertain to the implementation of security sensitive code. Detection and human review using developer workflow is required to ensure that defects pertaining to security hotspots do not find their ebay into production code.. As developers code and consequently deal with security hotspots, they MUST also be able to learn how to evaluate security risks by example and error identification whilst continuously adopting better secure coding practices. The tooling provided for this MUST enable such a scenario to take place to drive continuous developer security improvement. *Note that this pertains to custom coding for applications and components developed by all building blocks and not to the code behind 3rd party components and applications to which GovStack must be integrated.*	MUST	Building time

5.12.3 Support for Common Programming Languages The solution provided must support common programming languages used in the enterprise such as Java, JavaScript, C, C++, C#, Python, Scala, Kotlin, Golang and PHP.	MUST	Building time
5.12.4 Detection and Remediation of Top 10 Vulnerabilities The solution provided must minimally support the detection and remediation of the following types of security vulnerabilities (based on the OWASP Top 10 for 2021):: • Injection (all types) • Broken authentication • Sensitive data exposure • XML External Entities (XEE) • Broken access control • Security misconfiguration • Cross site scripting (XSS) • Insecure object deserialization • Libraries/components with known vulnerabilities • Lack of logging and monitoring • Generally poor coding practices in memory management etc.	MUST	Deployment time
5.12.5 Common Developer Tools and Frameworks The solution provided MUST integrate with common developer tools and frameworks as well as source code control systems such as GIT, SVN etc. and Jira for full cycle issue management.	MUST	Building time

5.12.6 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.13 Infrastructure Vulnerability Remediation Requirements

Functional Requirement	**Type (Must/ Should/ May)**	**Building time Security/ Deployment time Security/ Both**
5.13.1 Container Scanning Features The solution for containers (Docker and OCI) is presumed to be the main infrastructure layer for the project and it is this layer that requires protection from common vulnerabilities and exposures (CVE). The solution for containers MUST provide scanning tools to scan the content of deployed containers for known vulnerabilities with a view to reduce the attack surface for attackers.	MUST	Building time
5.13.2 Fully Integrated DevSecOps The solution for containers MUST have a fully integrated DevSecOps approach for CI/CD (continuous integration and continuous deployment) that prevents containers with known vulnerabilities from being deployed and	MUST	Building time

enables patches for known CVE to be deployed both inside the container and to the container orchestration layer and its associated components (AKA PaaS).		
5.13.3 Automatic Infrastructure Update The solution for containers MUST address the problem of automatically updating the infrastructure on a regular and/or demand basis to apply security patches for known vulnerabilities as soon as they are available. The goal is to reduce the window of vulnerability for new CVE's that are discovered. This is particularly in consideration of historical vulnerabilities that impacted hardware such as Spectre which impacted over 40M computers worldwide.	MUST	Deployment time
5.13.4 FIPS 140-2 and ECC Certifications The solution for container orchestration and its associated platform infrastructure MUST be certified as compliant with known security standards such as NIST FIPS 140-2 and the European Common Criteria certifications.	MUST	Building time
5.13.5 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.14 Data Loss and Leakage Prevention Requirements

Requirement	**Type (Must/ Should/ May)**	**Building time Security/**

		Deployment time Security/ Both
5.14.1 Multi-Channel Detection and Prevention The solution MUST provide a multi-channel means of detecting and preventing data leakage for critical and private data over web, email, instant messaging, file transfer, removable storage devices and printers and any other file transfer means.	MUST	Both
5.14.2 Fully Controlled Endpoint Protection The solution MUST provide endpoint protection for data in use on systems that run on internal end-user workstations or servers. Endpoint-based technology MUST address internal as well as external communications. Endpoint technology MUST be used to control information flow between groups or types of users. It MUST also control email and instant messaging communications before they reach the corporate archive and block communication/consumption/transmission/forwarding of critical and sensitive data. The solution MUST monitor and control access to physical devices (such as mobile devices with data storage capabilities - best to restrict mobile access) and restrict access information before it is encrypted (either in situ or in transit). The solution MUST provide the ability to implement contextual information classification (for example identifying what CUI is, or buy identification of the source or author generating content. The solution MUST provide application controls to block attempted transmissions of confidential information and provide immediate user feedback with logging and alerts to prevent or intercept future attempts through other channels. The endpoint solution MUST be installed on every workstation (laptop and mobile having access also) in the network (typically via a DLP Agent). Typically it pays to ensure that mobile devices are restricted from such access.	MUST	Deployment time

5.14.3 Confidential Information Identification The solution MUST include techniques for identifying confidential or sensitive information. Sometimes confused with discovery, data identification is a process by which organizations use a data leakage prevention technology to determine what to look for. Data is classified as either structured or unstructured. Structured data resides in fixed fields within a file such as a spreadsheet, while unstructured data refers to free-form text or media in text documents, PDF files and video etc.	MUST	Deployment time
5.14.4 Ability to Implement Compliance Audits The solution MUST provide the general ability to implement data loss and leakage prevention processes that drive compliance with PCI, HIPAA, GLBA, CIS, NIST and or similar European or African continental standards.	MUST	Building time
5.14.5 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.15 Data Encryption at Rest and In Transit Requirements

Functional Requirement	**Type (Must/ Should/ May)**	**Building time Security/**

		Deployment time Security/ Both
5.15.1 Support for Standards Based Data at Rest Encryption The solution (all components hosting sensitive data, personal data, credit card data or CUI) is required to be able to support the encryption of data at rest using standard strong encryption techniques such as ECC and RSA with certificate based PKI (i.e. X509 etc.). The certificates and PKI infrastructure used for this purpose MUST comply with the requirements stipulated in this document for digital identity. *The solution vendors for 3rd party components should articulate how each of the components supplied provides data encryption facilities for data at rest and the strength and benefits of their approach. This also applies to all components hosting data for all building blocks.*	MUST	Building time
5.15.2 Support for Standards Based Data in Transit Encryption All of the internal connections between the various components in each building block and between building blocks as well as the external connections for API calls etc. between web and mobile applications and their respective services must be encrypted for data in transit using standard strong encryption techniques such as ECC and RSA with certificate based PKI (ie. X509 etc.). The certificates and PKI infrastructure used for this purpose MUST comply with the requirements stipulated in this document for digital identity. *The solution vendors for 3rd party components should articulate how each of the components supplied provides data encryption facilities for data in transit and the strength and benefits of their approach. This also applies to all components communicating data between all building blocks.*	MUST	Both

5.16 Social Network, Media and Engineering Threat Management Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security / Deployment time Security / Both
5.16.1 Social Threat Mitigation The project MUST mitigate these types of threats with a combination of policy, training and technology. Many of the attacks in this style are initiated through phishing and dangerous email attachments. It is therefore anticipated that much of the technical aspects of mitigating these types of attacks can be addressed through the requirements identified elsewhere in this document. Cyber-criminals use a range of attack styles leveraging social networking, engineering and media to achieve a range of goals: for example to obtain personal data, hijack accounts, steal identities, initiate illegitimate payments, or convince the victim to proceed with any other activity against their self-interest, such as transferring money or sharing personal data" The most frequent styles of attack include: • **Phishing** – Email or social media based social engineering attacks. • **Vishing** – Voice-based social engineering, frequently over the phone but can also be in person or VoIP (i.e. Skype).	MUST	Deployment time

• **Smishing** – Mobile phone-based text messaging (SMS) social engineering attacks • **Thishing** - Targeted phishing attacks… for example against senior management (whaling https://searchsecurity.techtarget.com/definition/whaling) and specific people/organisations (spear phishing https://searchsecurity.techtarget.com/definition/spear-phishing) have recently become popular forms of social engineering attack for cyber-criminals. Vendors MUST, however, respond to this section of the requirements and articulate how their proposed solution explicitly protects from these styles of attacks and how other offerings are placed as part of their proposal for enablement etc. (including policy and training) collectively provide the required degree of protection and mitigation.		

GovStack

5.17 Cloud Security Posture Management Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security / Deployment time Security / Both
5.17.1 Continuous Posture Assessment The project MUST deliver continuous cloud security posture assessments of cloud environments through security and compliance teams. The solution must be able to manage the massive number of security posture management issues that will confront the project as infrastructure is deployed on public clouds with even modest deployments. The solution MUST be able to provide an assessment approach which addresses all of the common security concerns accompanying public cloud based deployments using containers, kubernetes and other public cloud services etc.. The key requirements are both taking control of; and keeping control of security risks in multiple, ever-growing and ever-changing cloud environments. Typically it is just too much data to process, too often, with an ever-expanding attack surface as more applications and services are deployed. This is the problem space that MUST be dealt with by the proposed solution and why continuous cloud security posture management is an absolute MUST for cloud deployments.	MUST	Deployment time

Note that GovStack may be deployed either on cloud infrastructure, on premise infrastructure or both. The intent of the CSPM requirements is for public cloud deployments although it can also be utilized for controlling on-premise public cloud offerings such as Azure Arc or AWS Outposts..		
5.17.2 Inventory, Ownership and Customization In terms of the inventory of data and services deployed, the following MUST be able to be managed (see ensuing rows)	MUST	Deployment time
5.17.3 Inventory Collection Coverage Must collect the metadata from all available cloud resources that have been deployed and utilized in the account. This collection MUST go beyond the core types of services (compute, networking, database, and IAM etc.) to incorporate a complete security metadata view of the utilized cloud resources and any potential exposures both inside and outside containers.	MUST	Deployment time
5.17.4 Data Ownership Solutions delivered as Software-as-a-Service (SaaS) provide quick onboarding, but they are typically managed by third-party with account-wide access to your cloud resources. This may present challenges with data ownership, compliance attestations of an external third party, and adherence to regional compliance mandates. These issues MUST be identified by CSPM.	MUST	Both
5.17.5 Customization 3rd party solutions can vary widely in their ability to make adjustments to how data is collected, how security checks are implemented, how results are filtered/excluded and reported, and how and when teams are alerted for issues. Without full control over all aspects, you may end up with results and notifications that cannot be properly	MUST	Deployment time

tuned, and this can lead to ignoring the tool and overall alert fatigue. The CSPM solution MUST provide the ability to navigate the posture across all 3rd party components and alleviate alert fatigue.		
5.17.6 Answer Complex/Advanced Questions Simple questions like “Is this S3 Bucket Public?” can usually be identified easily, but questions like “Are there any publicly accessible virtual machines with attached instance credentials that allow reading from S3 buckets tagged ‘sensitive’?”, “Which GKE Clusters have pods with access to escalate to ‘Project Owner’ via the attached Service Account? MUST be able to be answered by the CSPM solution”	MUST	Deployment time
5.17.7 Provide Continuous Results and Triage The CSPM solution MUST be designed with tunable results tracking and results triage workflows across multiple assessment intervals for continuous assessment as the total deployed solution evolves.	MUST	Deployment time
5.17.8 Focus on both Hardening and Compliance Most security practitioners are aware of the differences between an environment that is only compliant with one that is compliant and also well hardened. Compliance is a key driver for obtaining project funding and being able to operate a business legally, but going no further than compliance still leaves you open to critical risks. The CSPM tools MUST cover both security best practices and compliance objectives equally.	MUST	Building time

5.17.9 Compliance Objectives Driven Approach The solution MUST be able to associate controls with one or more compliance objectives. Views, filtering, and workflows should be driven by compliance objectives. For example, filtering controls by “PCI”, “NIST 800-53”, or “CIS” should narrow down the list to the controls that align with those frameworks with an identical mechanism that can also be used to filter controls for “lateral movement” or “privilege escalation”.	MUST	Building time
5.17.10 CSPM Basics The CSPM solution MUST address the basic cloud security posture management abilities as defined above for the broadest range of public cloud infrastructure providers (for example AWS, GCP and Azure). The following basic activities MUST be supported: 1. Collect several types of public-cloud-specific configuration data on a one-time or recurring basis from any cloud account resources (VMs, Clusters, IAM, etc), 2. Parse and load the configuration data into a graph database with deep linked relationships between resources to support advanced querying capabilities, 3. Run a customizable series of policy checks to determine conformance and record passing/failing resources on a recurring basis on that configuration, 4. Create custom groupings of related policy checks aiding in tracking remediation efforts and an associated reduction in risk over time, 5. Provide notifications to multiple destinations (email, logs, instant message etc.) when specified deviations from desired baselines occur.	MUST	Building time

5.17.11 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

GovStack

5.18 Endpoint Security and Portable Media Control Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.18.1 ESPMC Basics The project MUST deploy endpoint security and portable media control. Endpoint security and portable media control is also known as EDR or Endpoint Detection and Response. It is designed to protect from all sources of data leakage from various endpoints on the network including email, FTP and other protocols as well as portable media devices such as DVD_RW and removable USB drives. Techniques such as high grade encryption and protocol decoding along with keyword and phrase analysis are used to parse data streams as well as data transferred to removable media and apply data leakage prevention policy. *Note that this solution can potentially also be combined with one of the other specified solutions in a single product or suite and several vendors do exactly that with anti-virus-malware-phishing etc.*	MUST	Both
5.18.2 Real-Time Visibility The solution MUST provide real-time visibility and continuous analysis across the entirety of systems endpoints (characterized as ANY means by which data can be transmitted beyond the boundaries of the organization, either across the network or removable media)	MUST	Deployment time

5.18.3 IAM Solution Integration The solution MUST integrate with IAM (identity and access management) to implement group policy for determining appropriate access to protected data.	MUST	Deployment time
5.18.4 Telemetry The solution MUST provide extensive abilities to implement telemetry across all common data transmission channels and automatically notify the appropriate parties of attempts and breaches of access policy and enable the production of reports for the purpose of compliance evidencing.	MUST	Both
5.18.5 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

GovStack

5.19 Vulnerability Management and Security Automation Requirements

Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.19.1 Security Automation in General In general the vulnerability scanning requirements are covered in other sections of this document. The focus of this requirement is automation of vulnerability management to limit the window of vulnerability to be as short as possible once a vulnerability becomes known. The project MUST implement security automation which means automated patching and upgrades of the software and firmware in and around the applications and networking infrastructure to ensure that patches addressing security vulnerabilities are deployed consistently across all of the infrastructure.	MUST	Building time
5.19.2 Repeatability for both Cloud and On-Premise Deployments The security automation solution MUST provide a clean, consistent, simple and repeatable means of configuration management, applications deployment and patching that is flexible enough to support the entire infrastructure including both on-premise and cloud based deployments. Typically this would be achieved with a playbook style approach where playbooks are built using a language such as YAML and applied consistently through an automated scripting approach.	MUST	Deployment time

5.19.3 Agentless Orchestration and Provisioning The solution MUST provide orchestration, provisioning, module, plugin and inventory management support in order to be comprehensive for enterprise needs. Ideally the solution provided MUST be agentless and not require any additional software to be deployed on each node in the network.	MUST	Both
5.19.4 Consistent Configuration Management The solution MUST provide simple, reliable, and consistent configuration management capabilities. Must be able to be deployed quickly and simply with many out-of-the-box plugins for common technologies. Configurations MUST be simple data descriptions of infrastructure which are both readable by humans and parsable by machines. The solution itself MUST be able to connect to remote nodes under administration via SSH (Secure Socket Shell) key for secure configuration. For example configurations MUST be able to be consistently applied to hosts via lists of IP addresses and apply playbooks to install the configuration update on all the nodes. Subsequently the playbook can be executed from a control machine to effect the update. Logs MUST be taken and reported on to ensure that centralized admin is aware of any failure and can address them manually if necessary.	MUST	Deployment time
5.19.5 Orchestrated Workflows The offered solution MUST provide orchestration which involves bringing different elements together to run a whole operation. For example, with application deployment, you need to manage not just the front-end and backend services but the databases, networks and storage among other components. The orchestration solution MUST also ensure that all the tasks are handled in the proper order using automated workflows and provisioning etc.. Orchestations and playbooks MUST be reusable and repeatable tasks that can be applied time and again to the infrastructure based on parameters.	MUST	Deployment time

5.19.6 Site-wide Security Policy Implementation The solution MUST have the ability to implement sitewide security policies (such as firewall rules or locking down users) which can be implemented along with other automated processes. MUST be able to configure the security details on the control machine and run the associated playbook to automatically update the remote hosts with those details. This means that security compliance should be simplified and easily implemented. An admin's user ID and password MUST NOT be retrievable in plain text.	MUST	Both
5.19.7 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.20 Security Risk Profiling and Management Requirements

See the section of this document dealing with OSINT tools,

5.21 Intrusion Prevention and Detection Requirements

Functional Requirement	**Type (Must/ Should/ May)**	**Building time Security /**

		Deployment time Security / Both
5.21.1 HIDS, SIM and SIEM with Real-Time Integrity Monitoring The solution for intrusion prevention and detection MUST combine HIDS monitoring features with Security Incident Management (SIM)/Security Information and Event Management (SIEM) features. MUST also be able to perform real-time file integrity monitoring, Windows registry monitoring, rootkit detection, real-time alerting, and active response.	MUST	Deployment time
5.21.2 Multi-Platform Support The solution MUST be multi-platform and support deployment on Windows Server, and most modern Unix-like systems including Linux, FreeBSD, OpenBSD, and Solaris etc.. The reason for this is that it is yet to be determined exactly which O/S infrastructure will be required for the full GovStack solution and it will likely end up being a hybrid mix of various O/S platforms but predominantly Linux in containers.	MUST	Deployment time
5.21.3 Centralized Management The intrusion prevention and detection software MUST consist of a central manager for monitoring and receiving information from agents (most likely agents are required - they are small programs installed on the systems to be monitored. Of course an agentless solution will be acceptable if it is feasible). The central manager MUST include storage of the file integrity checking databases, logs, events, and system auditing entries.	MUST	Both

5.21.4 Basic IDS Features The intrusion prevention and detection solution MUST minimally offer the following features: • Log-based Intrusion Detection • Rootkit Detection • Malware Detection • Active Response • Compliance Auditing • File Integrity Monitoring • System Inventory Note that this is a very minimalist set of requirements and vendors may offer enterprise level open source subscriptions that offer more advanced features for intrusion prevention and detection which are AI/ML based for example. It is up to each vendor to articulate the value of what they are offering and why it is necessary.	MUST	Deployment time
5.21.5 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.22 Open Source Intelligence Platform (OSINT) Requirements

Functional Requirement	**Type (Must/ Should/ May)**	**Building time Security /**

		Deployment time Security / Both
5.22.1 General OSINT Requirements The project MUST provide OSINT tools with the following purposes and general requirements in focus (see ensuing rows):	MUST	Building time
5.22.2 Discovery of Public-Facing Assets The most common function of OSINT tools is helping IT teams discover public-facing assets and mapping what information each possesses that could contribute to a potential attack surface. This is public information about vulnerabilities in services and technologies that cyber-criminals can potentially use to gain access. In general, these tools don't typically try to look for things like specific program vulnerabilities or perform penetration testing but may also incorporate these features. The main purpose of OSINT tools is determining what information someone could publicly discover about the organization's assets without resorting to hacking and offer security professionals the opportunity to proactively address these vulnerabilities by reading the reports.	MUST	Both
5.22.3 Discover Relevant Information Outside the Organization A secondary function that some OSINT tools perform is looking for relevant information outside of an organization, such as in social media posts or information posted at specific domains and locations that might be outside of a tightly defined network. Organizations which have acquired a lot of diverse IT assets are excellent candidates for OSINT tools and we see that this has huge potential for GovStack. Given the extreme growth and popularity of social media, looking outside the organization perimeter for sensitive information is helpful for just about any group.	MUST	Deployment time

5.22.4 Collate Discovered Information into Actionable Form Finally, some OSINT tools help to collate and group all the discovered information into useful and actionable intelligence. Running an OSINT scan for a large enterprise can yield hundreds of thousands of results, especially if both internal and external assets are included. Piecing all that data together and being able to deal with the most serious problems first can be extremely helpful. The solutions offered MAY also incorporate AI/ML and neural network based solutions for better discovery and automation of analytics etc. *The offered solution SHOULD deliver better alignment of the organization's ability to comply with standards such as PCIDSS, HIPAA and GDPR etc. Prospective vendors SHOULD articulate how their solution achieves this goal.*	MUST	Deployment time
5.22.5 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.23 Fraud Prevention, Detection and Management Requirements

Functional Requirement	**Type (Must/ Should/ May)**	**Building time Security/ Deployment time Security/ Both**

5.23.1 Identify Fraudulent Purchases and Transactions The solution MUST provide the ability to identify potentially fraudulent purchases, transactions, or access. The solution MUST continuously examine user behavior, and analyse risk figures then identify any suspicious activity that may require intervention.	MUST	Deployment time
5.23.2 General Features Required The solution MUST be able to check the potential of fraudulent actions by users both internal and external, ensuring that transactions are lawful and genuine. The solution MUST also protect the sensitive information of both organizations and citizens (also see data leakage prevention). The solution MUST meet the following general features requirements: • AI/Deep Learning (Collaborative) • Analytics/Data Mining • Insider Threat Monitoring • Risk Assessment • Transaction Scoring • Intelligence Reporting • ID Analytics and Alerts • Real-Time Monitoring • Blacklisting • Investigator Notes • Transaction Approval • Custom Fraud Parameters • Unified Platform	MUST	Building time

• Data Enrichment • Continuous Innovation • Visualize real-time data • Anomaly detection • Frictionless Commerce • Early Warning System		
5.23.3 Fraud Information Dashboard The solution MUST provide an information dashboard with data visualization of statistical information and alerts pertaining to fraud prevention and detection that need to be addressed with an assigned priority.	MUST	Deployment time
5.23.4 Data Import from Many Sources The solution MUST provide the ability to import data sets from many applications and databases in order to perform analysis and provide fraud analysis and scoring information along with alerts on configurable events and thresholds.	MUST	Deployment time
5.23.5 CRM Integration The solution MUST provide integration with common customer relationship management solutions that are likely to be deployed as part and parcel of GovStack or perhaps already existing in the government's target architecture.	MUST	Both
5.23.6 API for Data Integration The solution MUST provide an open API for the purposes of integrating data from unknown sources to be checked for fraudulent activity.	MUST	Deployment time

5.23.7 AI/ML Based Capabilities Ideally the solution SHOULD incorporate advanced AI and ML capabilities for the purposes of detecting fraud with a combination of neural network, pattern recognition and data mining approaches to identify potentially fraudulent events based on advanced models and historical learning.	SHOULD	Deployment time
5.23.8 Investigator Notes and Workflows The solution MUST provide investigator notes and workflow capabilities for investigating potentially fraudulent incidents and managing those incidents through to resolution.	MUST	Deployment time
5.23.9 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.24 Security Incident Response and Management Requirements

Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both

5.24.1 General Incident Response Requirements The solution MUST provide an incident response and management system that implements the following general requirements as a fully integrated solution (referring to the other sections of this document): 1. Preparation of systems and procedures 2. Identification of incidents 3. Containment of attackers and incident activity 4. Eradication of attackers and re-entry options 5. Recovery from incidents, including restoration of systems 6. Lessons learned and application of feedback to the next round of preparation	MUST	Deployment time
5.24.2 Support for NIST and/OR SANS In general the NIST and/or SANS incident response frameworks SHOULD be followed and the tooling MUST facilitate this.	MUST	Deployment time
5.24.3 Assessment and Review Facilities The solution MUST provide capabilities for reviewing and documenting existing security measures and policies to determine effectiveness as well as performing a risk assessment to determine which vulnerabilities currently exist and the priority of your assets in relation to them. This Information is then applied to prioritizing responses for incident types and is also used to reconfigure systems to cover vulnerabilities and focus protection on high-priority assets. This phase is where you refine existing policies and procedures or write new ones if they are lacking. These procedures include documenting a communication plan and the assignment of roles and responsibilities during an incident.	MUST	Both

### 5.24.4 Tools to Support Detection and Identification The solution MUST support the use of the tools and procedures determined in the preparation phase and define the process and teams to work on detecting and identifying any suspicious activity. When an incident is detected, team members need to work to identify the nature of the attack, its source, and the goals of the attacker. During identification, any evidence collected MUST be protected and retained for later in-depth analysis. Responders MUST document all steps taken and evidence found, including all details. The purpose of this is to more effectively prosecute if and when an attacker is identified. ### 5.24.5 Support for Communications Planning After an incident is confirmed, communication plans MUST be initiated by the system. These plans MUST inform all concerned parties by workflow (i.e. security board members, stakeholders, authorities, legal counsel, and eventually users of the incident) advising which steps MUST be taken.	MUST	Deployment time
### 5.24.6 Threat Containment and Elimination The solution and its processes MUST support the containment and elimination of threats. After an incident is identified, containment methods are determined and enacted. **Containment:** The goal is to advance to this stage as quickly as possible to minimize the amount of damage caused. Containment MUST be able to be accomplished in sub-phases: - **Short term containment**—immediate threats are isolated in place. For example, the area of your network that an attacker is currently in may be segmented off. Or, a server that is infected may be taken offline and traffic redirected to a failover.	MUST	Deployment time

• **Long term containment**—additional access controls are applied to unaffected systems. Meanwhile, clean, patched versions of systems and resources are created and prepared for the recovery phase. **Elimination:** During and after containment, the full extent of an attack MUST be made visible. Once teams are aware of all affected systems and resources, they can begin ejecting attackers and eliminating malware from systems. This phase continues until all traces of the attack are removed. In some cases, this may require taking systems off-line so assets can be replaced with clean versions in recovery. It is anticipated that security automation tools will play a key role in this phase and SHOULD be integrated with the solution for this purpose (see the section outlining security automation requirements)		
## 5.24.7 Recovery, Restoration and Refinement The solution MUST support a recovery, restoration and refinement phase where recovery and restoration of damaged systems is achieved by bringing last-known-good versions online. Ideally, systems can be restored without loss of data but this isn't always possible. Teams MUST be able to determine when the last clean copy of data was created and restore from it. The recovery phase typically extends for a while as it also includes monitoring systems for a while after an incident to ensure that attackers don't return. Feedback and refinement MUST be enabled where lessons learned by your team's reviews along with the steps that were taken with a response. The team MUST be able to address what went well, what didn't, and document a process for future improvements. *Note that this is not something that gets performed in isolation by the incident response system but an integrated process that coordinates these phases across all security systems to formulate and execute the response.*	MUST	Deployment time

5.24.8 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

5.25 Security Testing and Sandbox Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.25.1 Comprehensive Sandbox The proposed solution MUST include a security sandbox environment with instances of the entire software stack and all of the security tools installed and configured such that security testing and scenarios can be addressed on an ongoing basis. This is not so much a solution requirement as a deployment requirement and MUST address all of the processes associated with each of the facets of security defined herein.	MUST	Deployment time
5.25.2 Scalability The security sandbox environment MUST scale to a level that is suitable for testing scenarios such as DDOS prevention etc. but does not have to be	MUST	Deployment time

implemented at the same scale as the regular test environment or the production environment for example.		
5.25.3 Test Scripting and Automation The project MUST conduct security testing using automated scripts as much as possible on an ongoing basis and the security team MUST take ownership of the ongoing securitization of all digital assets for each and every deployment. Note that the responsibility for these activities may ultimately be delegated to a government or country team in the case of build-X-transfer implementation scenarios. The role of the solution provider is to ensure that the baseline for these processes is established prior to any handover/transfer.	MUST	Both

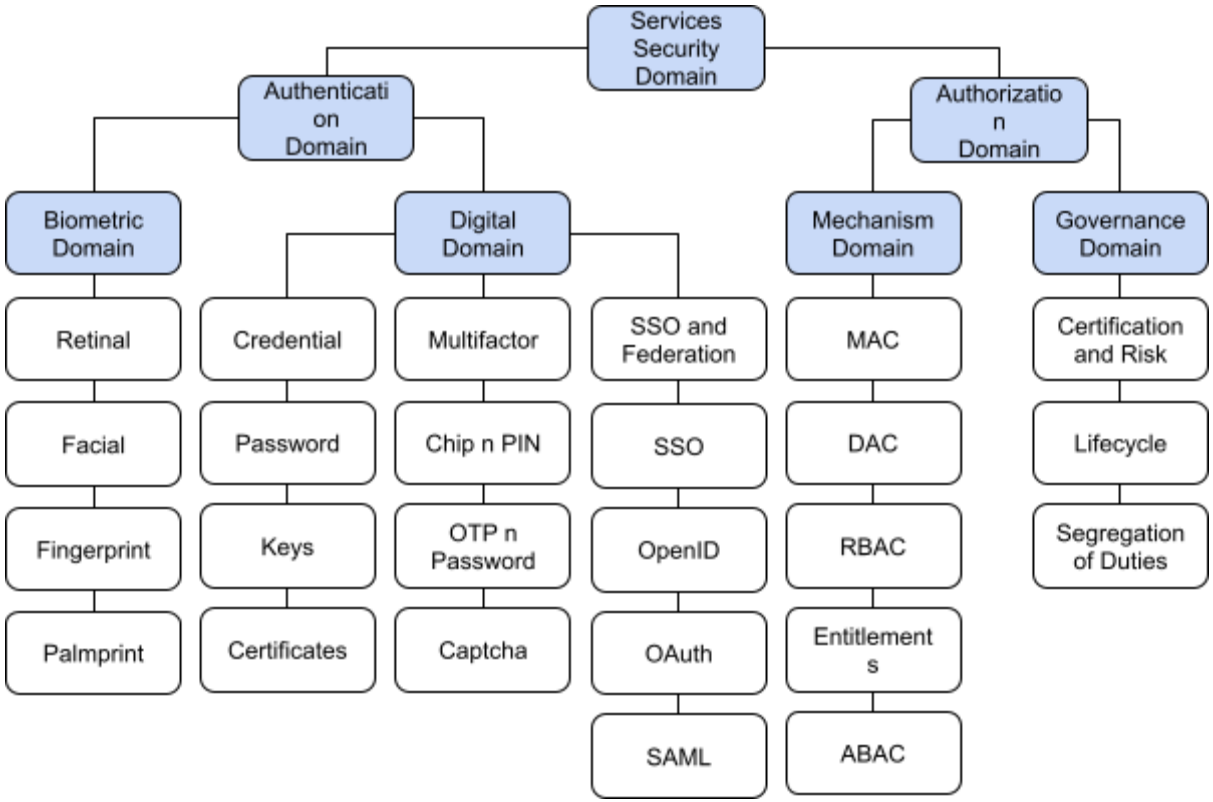
5.26 Critical Digital Infrastructure Business Continuity Requirements

Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
5.26.1 General Recovery Requirements The solution MUST cater for the recovery of all critical digital infrastructure in the case of major security incidents. Refer to the section above regarding the indecent response system for the general requirements regarding how such recovery MUST be performed.	MUST	Building time
5.26.2 Backup for Code and Images The backup and recovery systems for code, system images and data MUST cater for complete recovery of all critical digital infrastructure after natural disasters and also security incidents. The detailed requirements for such recovery of systems are to be provided by each building block as custodian of the code, images and data. Detailed information on the specific recovery requirements MUST be provided by the project for each GovStack implementation as they may vary widely due to constraints on budget and capabilities.	MUST	Deployment time

5.27 Data Structures

5.27.1 Resource Model

The resource model shows the relationship between data objects that are used by this Building Block. The following resource model depicts the basic elements of identity and access management (IAM) solutions required organized into domains:



5.27.2 Data Elements

The data elements provide detail for the resource model defined above. This section will list the core/required fields for each resource. Note that the data elements can be extended for a particular use case, but they must always contain at least the fields defined here. Information about data elements will include:

- Name
- Description
- Data Type
- Required/Optional flag
- Link to applicable standard(s)
- Notes

5.27.3 Example REST Authentication API

The following is a minimal example of how OpenIAM implements REST based authentication using its REST API:
(Note: The APIs will need to include appropriate request and response version numbers, for example, see https://docs.google.com/document/d/12b696fHlOAAHygFF5-XxUJkFyFjMIV99VDKZTXnnAkg/edit#heading=h.h9ypjkyetr1i)

URL

/idp/rest/api/v1/auth/public/login

Method

POST

Request Parameters

- **login**: user login (optional)
- **password**: user password (optional)
- **postbackURL**: redirectURL after success login (optional)

Headers

Content-Type:application/x-www-form-urlencoded

cURL Example

```
curl 'http://127.0.0.1:8080/idp/rest/api/auth/public/login' -X POST --data 'login=admin&password=pass123456'
```

Success Response Example


```
{
  "primaryKey": null,
  "status": 200,
  "errorList": null,
  "redirectURL": "/selfservice",
  "successToken": null,
  "successMessage": null,
  "contextValues": null,
  "possibleErrors": null,
  "passwordExpired": false,
  "userId": "3000",
  "unlockURL": null,
  "tokenInfo": {
    "authToken": "+Y9NkOrjOWn0BehKr6Cg1F7KcFliY=",
    "timeToLiveSeconds": -1
```

```
  ],
  "error": false
}
```


Error Response Example


```
{
  "primaryKey": null,
  "status": 500,
  "errorList": [
    {
      "i18nError": null,
      "error": "INVALID_LOGIN",
      "validationError": null,
      "params": null,
      "message": "Invalid Login and/or Password"
    }
  ],
  "redirectURL": null,
  "successToken": null,
  "successMessage": null,
  "contextValues": null,
  "possibleErrors": null,
  "passwordExpired": false,
  "userId": null,
  "unlockURL": null,
  "tokenInfo": null,
  "error": true
}
```

5.27.4 Example OAuth2 Authentication API

The following is a minimal example of how OpenIAM implements authentication with OAuth2 by requesting an OAuth2 token:

URL

/idp/oauth2/token

Method

POST

Request Parameters

- **client_secret**: Value of the client secret from OAuth client configuration page
- **client_id**: Value of the client ID from OAuth client configuration page
- **grant_type**: Type of grant flow
- **username**: Login of requester
- **password**: Password of requester

Headers

Content-Type:application/x-www-form-urlencoded

cURL Example

```
curl -v -XPOST --data 'client_secret=AAAAA&client_id=BBBBB&grant_type=password&username=DDDD&password=EEEE' 'http://127.0.0.1:8080/idp/oauth2/token'
```

Success Response Example


```
{
"access_token":
"ffj9Ub4lP-2_W4kXAXJ9MsleMxdDi9ALo0JqQw.3hV4driKa4IWXOSJ42PEQadFpN2FnpRruFLWD3",
"token_type": "Bearer",
"expires_in": 1320,
"refresh_token": "3nlth7DevrnjjpebATKkYCirs2aiy6L6SsBF_cpyxSXgVliwXC1rsCxRg99LCr7gHgYJ.A.VU"
}
```


Error Response Example


```
{
"error": "unauthorized_client",
"error_description": "Missing required parameter: client_id or client application is not registered"
}
```


5.27.5 Example OAuth2 Token Renewal API

The following is a minimal example of how OpenIAM implements authentication token renewal:

URL

/idp/rest/api/auth/renewToken

Method

GET

Headers

- Authorization: with oAuth token in format 'Bearer: <token>'
- Cookie: with current valid authentication token in format 'OPENIAM_AUTH_TOKEN=<token>'

cURL Example

```
curl -XGET -v -H 'Authorization: Bearer ssssss' -H 'Cookie: OPENIAM_AUTH_TOKEN=asdasdas' 'http://127.0.0.1:8080/idp/rest/api/auth/renewToken'
```

Success Response Example

```
{
"authToken": "asdas+asd/YagqkcJEql+asd=",
"timeToLiveSeconds": -1
}
```

Error Response Example

5.27.6 Example OAuth2 Authorization API

The following is a minimal example of how OpenIAM implements authorization using OAuth2:

URL

{server_url}/idp/oauth2/token/authorize

Replace *{server_url}* with the name of the server.

Method

GET

Parameters

- **response_type**: codetoken
- **client_id**: webconsole/Access Control/Authentication Providers/*needed provider* edit/ Client ID field
- **redirect_uri**: webconsole/Access Control/Authentication Providers/*needed provider* / Redirect Url. Use 'Space' or 'Enter' to separate values field.

cURL Example

```
curl -v XGET 'http://dev1.openiamdemo.com:8080/idp/oauth2/authorize?response_type=code&client_id=EF4128DCC0D24ED3BAC17FC918FDDBF5&redirect_uri=http://dev1.openiamdemo.com:8080/oauthhandler'
```

or, just:

```
curl 'http://dev1.openiamdemo.com:8080/idp/oauth2/authorize?response_type=code&client_id=EF4128DCC0D24ED3BAC17FC918FDDBF5&redirect_uri=http://dev1.openiamdemo.com:8080/oauthhandler'
```

Success Response Example

```
redirect to redirect_uri?code=code
```

Error Response Example

redirect to redirect_uri with error

5.27.7 Example OAuth2 Authorization Implicit Grant Flow API

The following is a minimal example of how OpenIAM implements an implicit grant flow API style of authentication:

URL

{server_url}/idp/oauth2/token/authorize

Replace *{server_url}* with the name of the server.

Method

GET

Parameters

- **response_type**: token
- **client_id**: webconsole/Access Control/Authentification Providers/*needed provider* edit/ Client ID field
- **redirect_uri**: webconsole/Access Control/Authentification Providers/*needed provider* / Redirect Url. Use 'Space' or 'Enter' to separate values field.

cURL Example

```
curl -v XGET 'http://dev1.openiamdemo.com:8080/idp/oauth2/authorize?response_type=token&client_id=EF4128DCC0D24ED3BAC17FC918FDDBF5&redirect_uri=http://dev1.openiamdemo.com:8080/oauthhandler'
```

or, just:

```
curl 'http://dev1.openiamdemo.com:8080/idp/oauth2/authorize?response_type=token&client_id=EF4128DCC0D24ED3BAC17FC918FDDBF5&redirect_uri=http://dev1.openiamdemo.com:8080/oauthhandler'
```

Success Response Example

```
redirect to redirect_uri?code=access_token=Pcej-9OdU_wshAjTn76MP-Cj5OgY_sfdYrt&expires_in=60000&token_type=Bearer
```

Error Response Example

```
redirect to redirect_uri with error
```

5.27.8 Example Get OAuth2 Token Information API

The following is a minimal example of how OpenIAM implements a get operation for token information:

URL

{server_url}/idp/oauth2/token/info

Replace *{server_url}* with the name of the server.

Method

GET

Parameters

token: token should be created via Create token.

cURL Example

```
curl -v XGET 'http://dev1.openiamdemo.com:8080/idp/oauth2/token/info?token=rdSOyor6hqJ2CrQ5QrpeXgX.ItgVEx1.nskN'
```

or, just:

```
curl 'http://dev1.openiamdemo.com:8080/idp/oauth2/token/info?token=rdSOyor6hqJ2CrQ5QrpeXgX.ItgVEx1.nskN'
```

Success Response Example


```
{
"expired": false,
"client_id": "92BF26DD50D748668730F7639C4A0D3D",
"user_id": "3000",
"access_token": "Lf_Sc-YeKHB8rsGfiGcLMKJOxbTGmpbdYs5wK3i7ZhINrjJlTOHEuV-phwJ1wE7.MjWqDcx8Lpri",
"expires_in": 1709,
"expires_on": 1531332707193,
"scopes": [
{
"scopeId": "c42a190a6488010b01648810b83a005a",
"name": "dev1 - /idp/oauth2/token/info"
```

```
},
{
"scopeId": "c42a190a6488010b01648810bdfd0096",
"name": "dev1 - /idp/rest/api/*"
},
{
"scopeId": "c42a190a6488010b01648810be2a0098",
"name": "dev1 - /webconsole/rest/api/*"
},
{
"scopeId": "c42a190a6488010b01648810be6d009b",
"name": "dev1 - /selfservice/rest/api/*"
},
{
"scopeId": "c42a190a6488010b01648810be96009d",
"name": "dev1 - /selfservice-ext/rest/api/*"
},
{
"scopeId": "c42a190a6488010b01648810bebf009f",
"name": "dev1 - /webconsole-idm/rest/api/*"
}
]
}
```


Error response example


```
{"code":401,"error":"invalid_token","error_description":"Authorization token is expired"}
```

5.27.9 Example Create OAuth2 Token API

The following is a minimal example of how OpenIAM would create an OAuth2 token using authorization grant flow style:

URL

{server_url}/idp/oauth2/token

Replace *{server_url}* with the name of the server.

Method

POST

Parameters

- **client_secret**: webconsole/Access Control/Authentification Providers/*needed provider* edit/ Client Secret field
- **client_id**: webconsole/Access Control/Authentification Providers/*needed provider* edit/ Client ID field
- **grant_type**: authorization_code
- **redirect_uri**: webconsole/Access Control/Authentification Providers/*needed provider* / Redirect Url. Use 'Space' or 'Enter' to separate values field
- **code**: code should be generated with Authorization code grant flow request.

Headers

```
Content-Type: application/x-www-form-urlencoded
```

cURL Example

```
curl -v -XPOST --data 'client_secret=client_secret&client_id=client_id&grant_type=authorization_code&redirect_uri=redirect_uri&code=code' 'http://dev1.openiamdemo.com:8080/idp/oauth2/token'
```

Success Response Example

```
{
"access_token": "MjwOxreF7e-NW_NBA6UNuB9d_4IcohdK2bUSxqZ_BxDlCG7uD.ZstmoKwvuWq1hL49pClk3dlo",
"token_type": "Bearer",
"expires_in": 1800
}
```

Error Response Example

```
{
"error": "invalid_request",
"error_description": "Code parameter is expired"
}
```

```
{
"error": "invalid_request",
"error_description": "Redirect URL does not mach to initial one."
}
```

5.27.10 Example Revoke OAuth2 Token API

The following is a minimal example of how OpenIAM implements OAuth2 token revocation:

URL

{server_url}/idp/oauth2/token/revoke

Replace *{server_url}* with the name of the server.

Method

POST

Headers

```
Content-Type=application/x-www-form-urlencoded
```

Parameters

token: token should be created via Create token.

cURL Example

```
curl -v XPOST --data 'token=token' 'http://dev1.openiamdemo.com:8080/idp/oauth2/revoke'
```

Success Response Example


```
{
"status": "SUCCESS",
"errorCode": null,
"errorText": null,
```

"fieldMappings": null,
"stacktraceText": null,
"responseValue": null,
"errorTokenList": null,
"failure": false,
"success": true
}

Error Response Example

{"code":403,"error":"insufficient_scope","error_description":"The request requires higher privileges than provided by the access token","scope":"1531291154553_/idp/oauth2/revoke"}

5.27.11 Example Validate OAuth2 Token API

The following is a minimal example of how OpenIAM implmenents OAuth2 token validation:

URL

{server_url}/idp/oauth2/token/validate

Replace *{server_url}* with the name of the server.

Method

GET

Parameters

token: token should be created via Create token.

cURL Example

```
curl -v XGET 'http://dev1.openiamdemo.com:8080/idp/oauth2/token/validate?token=rdSOyor6hqJ2CrQ5QrpeXgX.ItgVEx1.nskN'
```

or, just:

```
curl 'http://dev1.openiamdemo.com:8080/idp/oauth2/token/validate?token=rdSOyor6hqJ2CrQ5QrpeXgX.ItgVEx1.nskN'
```

Success Response Example

```
{
"code": 200
}
```

Error Response Example

```
{
"code": 401,
"error": "invalid_request",
"error_description": "Authorization token is not found"
}
```

5.27.12 Example Refresh OAuth2 Token API

The following is a minimal example of the way that OpenIAM implements OAuth2 token refresh:

URL

{server_url}/idp/oauth2/token/refresh

Replace *{server_url}* with the name of the server.

Method

POST

Parameters

token: token should be created via Create token.

cURL Example

```
curl -v XPOST --data 'refreesh_token=refreesh_token' 'http://dev1.openiamdemo.com:8080/idp/oauth2/token/refresh'
```

Success Response Example

```
{
"access_token": "j4l6p0vDxZKduJXlipXZq5LNQqY1aJWvtYp812.k6246Sn2FY3rpyos..qJtScD8.wjytm1idsnopHmb.u",
"token_type": "Bearer",
```

"expires_in": 60,
"refresh_token":
"0q1V8ZJaayxV5qcD4JtwF4.LyQOPMaZY.cNsIuZAYQ-TXGqrZDpy6AEOhc58dwEjgHDN2Bx_J.XkVTZ"
}

Error response example

{
"timestamp": 1531449532875,
"status": 400,
"error": "Bad Request",
"exception": "org.springframework.web.bind.MissingServletRequestParameterException",
"message": "Required String parameter 'refresh_token' is not present",
"path": "/idp/oauth2/token/refresh"
}

5.27.13 Example User Information from OAuth2 Token API

The following is a minimal example of how user information can be obtained from OpenIAM using an OAuth2 token:

URL

{server_url}/idp/oauth2/userinfo

Replace *{server_url}* with the name of the server.

Method

GET

Parameters

token: token should be created via Create token.

cURL Example

```
curl -v XGET 'http://dev1.openiamdemo.com:8080/idp/oauth2/userinfo?token=rdSOyor6hqJ2CrQ5QrpeXgX.ItgVEx1.nskN'
```

or, just:

```
curl 'http://dev1.openiamdemo.com:8080/idp/oauth2/userinfo?token=rdSOyor6hqJ2CrQ5QrpeXgX.ItgVEx1.nskN'
```

Success Response Example

```
{
"sub": "3000"
}
```

Error Response Example

```
{"code":401,"error":"invalid_request","error_description":"Authorization token is not found"}
{"code":403,"error":"insufficient_scope","error_description":"The request requires higher privileges than provided by the access token","scope":"1531291154428_/idp/oauth2/userinfo"}
```

5.27.14 Example API for Defining Resources, Roles, Access and Provisioning

The most comprehensive API available for this is delivered by OpenIAM. Unfortunately this API is currently delivered in SOAP. The purpose of this API is to provide 3rd parties the ability to create resources, roles and access within the IAM system. There are multiple options to get this done including batch upload and configuration using the administrative user interface. This would need to be addressed at implementation time using the most practical means. There does not seem to be a current use case for the BB's to create these types of resources on the fly using the IAM API. The API definitions can be found here:
https://docs.openiam.com/docs-4.1.14/html/docs.htm#API/SOAP/SOAP.htm%3FTocPath%3DAPI%2520Guide%7CPart%2520II%253A%2520SOAP%2520API%2520integration%2520services%7C_____0

6 Security Building Block Modules

The functional requirements section lists the technical capabilities that the security building block MUST have. Although API Management and Gateway services are an architectural element, this section of the document also describes the detailed functional requirements for implementing API management, governance and gateway services for GovStack. Explicitly, the communications between all building blocks (BB's) and applications shall be via open API based access.

The goal of this endeavor is to address primary security concerns centrally and create a consistent way of implementing a modern cloud-ready architecture to publish API's to 3rd parties (both internal and external), govern and manage the access to API's both internally and externally by policy and create centralized and secure point of access to each and every API endpoint exposed through GovStack. These functional requirements do not define specific APIs (API's themselves are implemented by other building blocks) - these functional requirements only define the functionality that must be implemented within the bounds of the security building block and how it needs to be applied to other building blocks.

The following features and functions are required (both functional and non-functional) - see elaboration of these below:

6.1 API Management and Gateway Functional Requirements

Functional Requirement	Type (Must/ Should/ May)	Building time Security/ Deployment time Security/ Both
6.1.1 Multiple API Gateway The ability to implement segregated gateways for both internal and external API traffic. This means that the internal API driven integration traffic and the external API access traffic is to be segregated into separate gateway infrastructure components and access controlled by networks and network access policy.	MUST	Deployment time
6.1.2 Standards Based Identity and Access The ability to implement standardized authentication, authorization and encryption protocols including federated identity (OAuth2, OpenID Connect, SAML2, SSO, SSL, TLS etc.). These standards MUST be supported and incorporated into any chosen API Management product out-of-the-box for controlling access to API interfaces. Note that the native API interfaces for each building block's components may be implemented in clear text with no authentication and/or encryption etc. so long as the access to these interfaces is firewalled by network and network policy such that it is ONLY accessible through the API Gateway. This is intended to simplify, standardize and expedite API development, deployment and management. Note that the APi interface specifications for the APi gateway and API management services are based on open standards based identity and access which is in turn based on OpenAPI 3.0. The following link describes how OAuth2 is used in OpenAPI 3.0 standards based identity and access: https://swagger.io/docs/specification/authentication/oauth2/. It is this style of standards based identity and	MUST	Building time

access that is required to be supported, Note that additional API interfaces may be exposed by the API Management and Gateway solution but they would predominantly be targeted for incorporation into the DevOps CI/CD tool chain not exposed to other BBs.		
6.1.3 Identity Store Plugins The ability to utilize separate identity stores as repositories for identity and perform proxied authentication to such repositories (for example LDAP) using multiple credentials including digital identity certificates.	MUST	Both
6.1.4 API Protection Features The ability to support many security and protection features such as Standard API keys, App-ID key pair, IP address filtering, Referrer domain filtering, Message encryption, Rule-based routing, Payload security, Channel security, Defense against common XML and JSON attacks, Low- to no-code security configuration, PCI compliance. *Note that this is not an exhaustive list and additional policy protection features may strengthen the value of any given solution.*	MUST	Deployment time
6.1.5 Centralized API Policy Based Access The ability to implement policy based access management for API endpoints. The policy MUST be able to be implemented centrally then applied across multiple gateways and all API endpoints.	MUST	Deployment time
6.1.6 API Endpoint Transformation The ability to support multiple standards for proxying endpoints and exposing them as standard OpenAPI endpoints (see Ref 1) including transformations such as XML ↔ JSON and SOAP ↔ RE *Note that this is not an exhaustive list of the required transformations and that additional transformations may reinforce the strength of a solution's flexibility and adaptability.*	MUST	Deployment time

6.1.7 Alternative API Protocols The ability to support multiple common protocols such as JMS, WS, MQTT. *Note that this is not an exhaustive list and that additional alternative protocols supported may strengthen the value of the proposed solution.*	MUST	Deployment time
6.1.8 API Versioning and Lifecycle The ability to support multiple API versions and control multiple API versions and API lifecycle management. *Note that there are many potential features available to strengthen the solution proposition such as version dependency management and deployment rollback etc.*	MUST	Deployment time
6.1.9 API Call Traffic Shaping The ability to implement traffic transformation and traffic shaping. This is typically implemented as single/dual rate, private (per node) and shared (by multiple nodes) shapers. *Note that additional traffic shaping capabilities may strengthen the solution proposition.*	MUST	Deployment time
6.1.10 API Call Rate Limiting The ability to implement rate limiting for API calls on an API-by-API basis. This limits the rate at which API's can be called by a consumer (for example 100/second etc.) and usually has many flexible options and caters for policy driven rate limits based on busy times etc. Principally it comes down to the offered SLA. *Support for complex SLA construction may strengthen the solution proposition.*	MUST	Deployment time

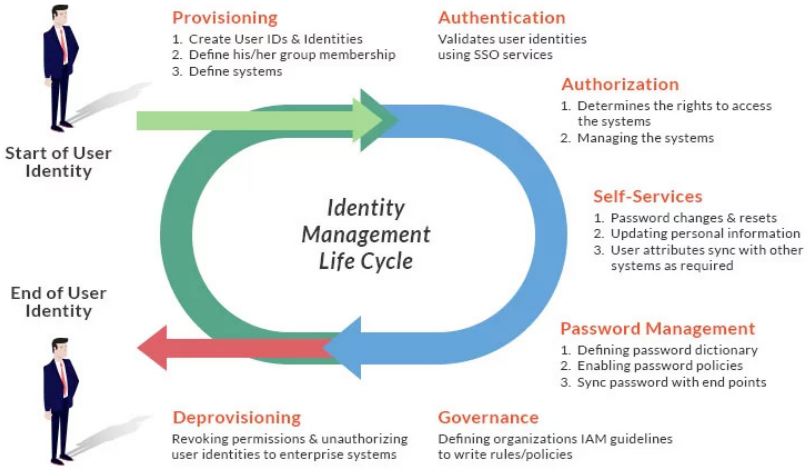
6.1.11 API Call Quotas The ability to implement quotas for API calls from specific clients (daily, weekly, monthly etc.). This restricts the number of API calls a client can make and also often includes flexible options so that a complex SLA can be constructed. *Support for complex SLA construction may strengthen the solution proposition.*	MUST	Deployment time
6.1.12 API Call Logging, Monitoring and Alerts The ability to implement logging and monitoring of API calls with reporting and administrative alerts. Typically extensive functionality with multiple logging levels is implemented. *Support for flexible levels of logging (for example debug, trace etc.) may strengthen the solution proposition.*	MUST	Deployment time
6.1.13 API Call Analytics The ability to implement advanced analytics with out-of-the-box charts and reporting on demand along with the ability to trigger alerts based on analytics. *The strength, flexibility, feature set and appeal of the analytics and charting capabilities may strengthen the solution proposition. Note that this can optionally be implemented as a separate tooling layer perhaps using tools such as Prometheus and Grafana etc.*	MUST	Deployment time
6.1.14 API Virtualization The ability to implement virtualized API endpoints. API virtualization is the process of using a tool that creates a virtual copy of your API, which mirrors all of the specifications of your production API, and using this virtual copy in place of your production API for testing. *Note that this is NOT API mocking but provides an actual endpoint for solution testing to proceed unhindered.*	MUST	Deployment time

6.1.15 API Developer Portal The ability to publish API specifications through a developer portal using open standards. Includes features such as portal availability across deployment types (on-premise, cloud, etc.), interactive API documentation, developer metrics, developer portal templates, portal customization (HTML, CSS etc.), ability to withdraw developer keys, either temporarily or permanently. *Note that advanced developer portal features may strengthen the value of the solution proposition.*	MUST	Deployment time
6.1.16 Flexibility in API Deployment Architectures The ability to support a diverse array of deployment architectures including standalone, on-premise cloud and public cloud models including the ability to support a fully integrated microservices architecture based on containers. The architecture should also allow for the separation of key components and interfaces for meeting complex network security needs. *Note that the more flexible the deployment architectures are, the stronger the solution proposition.*	MUST	Deployment time
6.1.17 Advanced DevOps Artefact Deployment The ability to support advanced DevOps deployment techniques such as “canary deployment”, “blue-green deployment” and “AB testing”. Additional advanced deployment scenarios and innovations will strengthen the value proposition of the proposed solution.	MUST	Deployment time
6.1.18 File Storage Integration The ability to implement file storage platform integration such as S3 etc. - not exhaustive.	MUST	Deployment time

Note that flexibility in the support for storage integration across additional file storage standards will strengthen the value of the solution proposition.		
6.1.19 API Monetization The ability to monetize API calls by specific 3rd parties or partners for the purposes of simply raising capital or creating partnership incentives through 3rd party portals. Features such as billing support, support for multiple models of revenue generation, low- to no-code monetization configuration, third-party payment system integration, prepay and/or postpay invoicing, multi-currency support and tax compliance are negotiable. *Note that the more advanced the monetization features are, the higher the solution value proposition is.*	SHOULD	Deployment time
6.1.20 High Availability The solution provided and any associated components MUST be highly available and utilize clustering technology in order to provide a minimum of 24x7x365 service with 99.99% availability (AKA 4 9's).	MUST	Deployment time
6.1.21 Open Source Based The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

6.2. Identity and Access Management (IAM) Suite Functional Requirements

Functional Requirement	**Type (Must/ Should/ May)**	**Building time Security/**

		Deployment time Security/ Both
6.2.1 Identity Lifecycle Management The following diagram defines the basic lifecycle required for managing identities:  What is required is a flexible and comprehensive user lifecycle management solution which provides the following generalized features and functions (see ensuing rows).	MUST	Deployment time
6.2.2 User Administration Tools These are required so that administrative users and/or help desk users can centrally manage all user profiles and records. These features include:	MUST	Both

○ Unlock accounts and reset passwords when needed ○ Changing the user status ○ Session management – visibility to see active users and to kill their session if needed ○ Workflow - monitor all workflows and terminate them if necessary ○ Review audit logs for authentication, access and changes to identity records. ○ Manage user profile attributes including credentials etc. ○ Manage user access rights for resources such as documents and APIs.		
### 6.2.3 Multi-Source Identity Integration Integration with multiple source identity systems (such as the Identity BB, databases and LDAP) to automatically initiate provisioning/deprovisioning activities related to enrollment, policy changes and un-enrollment processes.	MUST	Deployment time
### 6.2.4 Multi-Source Identity Synchronization Multi-source synchronization of user identity data. This allows the system to integrate and synchronize with an authoritative source such as the national ID, social security system or perhaps more likely the Identity BB via an API/plugin approach. This is required for both initial load of identities and for ongoing provisioning, re-provisioning and deprovisioning etc.. The synchronization must determine which systems a user should be provisioned to/de-provisioned from, which permissions should be set or revoked for the applications that a person is entitled to and whether or not provisioning should be automatic or if an additional workflow should be triggered for human or other automated processing.	MUST	Both
### 6.2.5 Identity Reconciliation Automatic user identity record reconciliation. This is where synchronization is used to detect changes in the source system identity data and reconciliation is used to detect, compare and resolve the changes. For example a user record has been added to a target system manually instead of using IAM.In this case reconciliation can be configured to either:	MUST	Deployment time

1. Add the user to the IAM system 2. Remove the user from the target system or; 3. Do nothing but flag the anomaly to an administrator		
6.2.6 Self Service Portal and Workflow A customizable, workflow driven, self-service user interface portal to enable administrators to create and manage policies, users and the various artefacts. Must support approval workflows to multiple stakeholders.	MUST	Deployment time
6.2.7 Advanced Password Management The advanced password management capabilities must include the following: ○ Self-service Password Reset (SSPR) ○ Administrative change password ○ Password synchronization with directory and other resources. ○ Must work across both cloud and on-premise applications lowered costs and improved security for hybrid cloud deployments ○ Ability to enforce strong password policies ○ Self-service password which allows users to reset their own password without going to the help desk. Reset is via a combination of challenge/response questions, one-time link via registered email, one-time token via SMS or mobile device messaging and one-time PIN. ○ Password aging with password change reminders sent via email and potentially mobile messaging. ○ Password change synchronization across all target systems.	MUST	Deployment time
6.2.8 User Access Request Management Whilst provisioning processes can be triggered through the automated user lifecycle management functionality, the IAM solution must also provide a self-service feature through the portal via which end-users can request	MUST	Deployment time

access. This access-request functionality is provided through a shopping cart and service catalog design where the user has access to select the services to which they would like to request access. Applications and application-specific entitlements such as membership to an LDAP group or a role on a public cloud provider such as AWS must be supported as a part of the access grant process. The access request feature must also include the optional selection of a "Profile Role" capability – which is a role defined for a job/position that can grant access to a number of applications that are needed for a particular job role (otherwise known as delegation). The access request/approval workflows provided must support: ○ **Multiple approvers** – Must be able to define as many approval steps needed and select common targets such as a supervisor, object owner or admin, and group of approvers. ○ **Service Level Agreements (SLA)** - Must ensure that tasks are completed in a timely manner so if they are not, then they can be escalated to the appropriate person after expiry time. The access request approval functionality must also support basic "delegated approval" and "out of office" functionality so that there are no barriers to self service access requests when the usual approvers are not available.		
6.2.9 REST API A REST (representational state transfer) based API for external integration of the provisioning and deprovisioning of users etc.. is required. It must support the same OpenAPI standards defined by the Architecture Blueprint and Functional Requirements (see Ref 1). An example of such a REST API is provided with the OpenIAM suite specification here: https://www.openiam.com/products/identity-governance/features/api/. This is intended as an example of the API style that is expected in such a suite. This is not definitive and may vary based on the proposed IAM suite.	MUST	Deployment time

6.2.10 Orphan Management Organizations which are not actively using an IAM platform often have orphaned user records in their business applications. These are those records which are the result of users being given access and not having that access revoked when a person is unenrolled from a service. Orphan management functionality consolidates all the orphaned records and provides administrators with tools to either clean up these records or link them to the correct user.	MUST	Deployment time
6.2.11 Access Certification Regulatory requirements, such as GDPR, HIPPA and SOX combined with an increased focus on security are causing both public and private organizations to implement access certification policies. Scheduled access certification campaigns aid in complying with these regulatory mandates as well as improve security by guarding against the access violations which lead to security breaches. However, when performed manually, these activities can be error prone and very time consuming for most mid to large organizations. The lack of consistency resulting from manual processes results in failed compliance audits and threats resulting from unauthorized access can slip undetected. The IAM solution must provide the ability to automate the access certification process which addresses the challenges found when performing these processes manually. The following types of certifications must be supported by the IAM solution: ● User Access Certification ● Application Access Certification ● Group Access Certification These campaigns can be scheduled and run at regular intervals or they can be run on demand. The Access Certification functionality in the IAM solution must provide organizations with the following capabilities: ○ **Human Friendly Reviews**: End-users (reviewers) using the IAM solution access certification functionality must be able to perform their activities in a familiar self-service user interface.	MUST	Deployment time

Reviewers must be able to review all the historical access in a central location as well as use tools to compare access (date, time, service etc.) between individuals. ○ **Closed Loop Revocations**: During the certification process, reviewers must be able to revoke accounts and entitlements with a simple one-click mechanism. The closed-loop validation mechanism will then ensure that revoked access has been deprovisioned from the target applications. ○ **Support for Cloud, On-Premise and Hybrid Cloud**: An increasing number of organizations today have hybrid environments where applications are deployed both on-premise and in the cloud. The IAM solution must provide a central identity governance platform such that the same consistent certification programs can be undertaken irrespective of the applications and infrastructure location.		
6.2.12 Workflow Creation The IAM solution identity governance feature set must provide the creation of workflows to support complex processing, integration and approval steps. While custom workflows must be defined, the IAM solution must also provide default out-of-the-box workflow templates for common operations to simplify the configuration effort. Each of these workflows must support: ○ Multiple approvers ○ Service Level Agreements (SLAs) to ensure timely completion (with automatic delegation and temporary changes in the new approvers access rights to enable time-sensitive approval) The following predefined workflow templates are required to be provided: ○ Enrollment (Joiner) ○ Role Change (Mover, Additional Services) ○ Unenrollment (Leaver)	MUST	Deployment time

○ Status change ○ Access request (Additional Services) ○ Group creation (with group policy) ○ Self-registration (with a customized external workflow to integrate with the Identity BB) ○ Access Certification		
6.2.13 Custom Workflows Whilst the IAM solution must provide the above workflow templates, it also must include a BPMN compliant workflow engine that can be used to create new custom workflows. A graphical process designer such as the BPMN designer plugin such as one of the many available for the Eclipse IDE or similar must be included to simplify the effort required to create new custom workflows.	MUST	Deployment time
6.2.14 Audit and Compliance Facilitating compliance with regulatory requirements or internal security policies is one of the principal drivers for Identity Governance. The IAM solution must provide tools to help organizations meet compliance mandates. Organizations deploying the IAM solution as a part of GovStack are required to automate a variety of operations that sometimes utilize workflow-based approval steps. Detailed audit logs must be associated with each of these operations so that organizations can answer the following fundamental questions: ○ What access rights does a user currently have? ○ When were they granted these rights?	MUST	Both

○ How or why were they granted these rights and by whom were they granted? In addition to access information, the IAM audit logs must also track details related to authentications, password changes, life cycle status changes, system configuration changes etc.. Using the information provided by these logs, combined with the out of the box reports, and self-service tools, organizations must be able to achieve the following: ○ Provide the auditors with clear evidence of compliance or otherwise. ○ The ability to proactively review, detect and revoke inappropriate access from any user. ○ The ability to review all access rights before granting any additional access rights. ○ A centralized review and certification of applications/systems access rights across both on-premise and cloud.		
6.2.15 Connectors The IAM solution must provide a large range of connectors for both source and target applications out-of-the-box, The reason for this is that GovStack will be deployed in more than one country and in unknown applications environments and must be able to be integrated with many different common enterprise information systems and services both on premise and in the cloud in a simple and cost effective manner… for example: ○ **Microsoft** (Active Directory, PowerShell, Windows, Azure AD, Exchange, SQL Server, Office 365, Azure DevOps, Dynamics365) ○ **Oracle** (eBusiness Suite (EBS), IDCS, database) ○ **ERP/HR Systems** (Oracle, SAP, ADP, Workday) ○ **Public Cloud** (Amazon Web Services, Azure, Google Cloud) ○ **Infrastructure** (Database/JDBC, GIT, LDAP (OpenLDAP, eDirectory, Active Directory, ApacheDS etc.), Linux (RHEL, CentOS, Ubuntu), REST web services, SCIM 2, scripts) ○ **Others** (Google Gsuite, Salesforce, SAP Hana, Slack (SCIM), Tableau… *not an exhaustive list*)	MUST	Building time

6.2.16 Access Management Access management is an integral part of the required IAM solution. The Access Manager must provide a scalable, secure and consistent solution to implement policy based access for applications in hybrid cloud environments for both internal (employees), citizens (external) and 3rd parties (external) alike. The Access Manager must provide the following tools to enable these objectives (see ensuing rows):	MUST	Deployment time
6.2.17 Web SSO Web single-sign-on must be provided with support for SAML 2, oAuth 2, OpenIDConnect, OIDC, and a proxy to allow SSO to legacy applications. This enables web based applications to be easily configured for SSO without modification.	MUST	Deployment time
6.2.18 Adaptive Authentication An adaptive authentication system as follows must be provided with the following features: ○ Password-based authentication ○ Certificate-based authentication ○ MFA-SMS/E-mail/Mobile app-based OTP ○ Adaptive Authentication builds on these options to provide a robust framework where users can build rich authentication workflows using a browser-based drag-and-drop interface. ○ The flows can take into account a broad range of risk factors including device, context, user choices, geolocation, profile attributes, user behavior and foundational identity systems.	MUST	Deployment time

○ This allows the implementation of a solution which offers a significantly higher level of security while providing an improved end-user experience in comparison to traditional options.		
6.2.19 Multi Factor Authentication (MFA) Note:- see authentication too - but this is for MFA to target applications integration for the purpose of access management: ○ While the IAM solution framework must allow the use of third party MFA products, it must also provide its own MFA solution which is pre-integrated and ready to use. The following MFA options must be provided out-of-the-box: ■ SMS-based OTP ■ E-mail-based OTP ■ Mobile app (iOS or Android) OTP plus push notification support	MUST	Deployment time
6.2.20 Social Sign-on (as opposed to single-sign-on) The Access Manager should allow social sign-on from social identity providers such as Google, Facebook and LinkedIn. Social registration significantly reduces the registration effort by allowing select attributes to be dynamically transferred from the social provider. This may or may not be used in practice but is a desired feature.	SHOULD	Deployment time
6.2.21 RBAC Based Authorization The IAM solution must provide a flexible RBAC-based authorization model to enforce security into applications through the Access Manager. The RBAC model must support inheritance as well as direct entitlements and provide the flexibility needed to implement complex real world requirements. The authorization service must be able to be used in conjunction with oAuth2 and the Access Gateway to enforce the authorization rules.	MUST	Deployment time

6.2.22 Session Management The IAM solution must provide session management for issues like session timeout to reduce the exposure created by long running sessions. This includes API's to extend expiring tokens etc. for application and user convenience.	MUST	Deployment time
6.2.23 Device Registration The IAM solution must provide device registration such that only registered devices can be used to access services by policy.	MUST	Deployment time
6.2.24 Fine Grained Audit Logging The IAM solution must provide fine grained audit logging by the Access Manager so that the explicit date, time, user and service access is logged.	MUST	Deployment time
6.2.25 Access Gateway An access gateway is required in order to provide protected proxy gateway access to the web through reverse and front side web infrastructures such as Apache and Nginx web servers. This must provide the following functionality: ○ SSO to legacy applications ○ Session management	MUST	Both

○ Protection of APIs and application URLs by enforcing authentication and authorization rules unless a 3rd party API Management and Gateway suite is used in which case the access gateway must be configurable to utilize the 3rd party API Gateway.		
6.2.26 Legacy SOA Security Features The IAM suite must be able to implement a pure legacy SOA approach. A legacy SOA API with all required operations must be available to facilitate integrations with legacy SOAP/SOA systems. The IAM solution must provide SOA federation for controlling access to services in a legacy SOA environment using SAML, SAML 2 and WS-Security. The IAM solution must be able to enforce policies throughout SOA based services. RBAC and XACML support must be provided to allow the IAM solution to implement a flexible security model that supports the following: ○ Distributed services (vs monolithic applications) ○ Services distributed across organizational boundaries ○ Service Interoperability ○ Integration of disparate legacy SOA protocols	MUST	Deployment time
6.2.27 Web Access Management The Access Gateway must be able to provide coarse-grained authorization when protecting web applications in totality. Requests must be routed through a proxy, which applies authorization rules, and forwards the request to the underlying servers, providing the application. The model must be simple to deploy and easy to maintain. User identity must be checked and propagated through HTML header injections, HTTP query strings or HTTPS authentication headers to applications hidden behind a proxy server for the purposes of openness and compatibility. The native URL of these applications must be hidden from the public view (i.e. it is only exposed as a service name in a secure manner).	MUST	Deployment time

6.2.28 Single Sign On (SSO) Each partner application, as well as internal applications and building blocks, may have their own set of security credentials and various authentication methods. Such applications may move in and out of security domains. The user experience suffers when many login credentials must be remembered. Therefore the IAM solution must provide SSO features that allow users login once and roam unchallenged through a security realm to which they have been granted access. This reduces the burden of many passwords and eliminates the need to individually login to each application. Users must be able to login once, and roam freely across secured domains without being challenged again. Participating security domains must never be required to give up their own credentials. The ability to hold multiple identities, each with their own roles, permissions, access-levels and entitlements across multiple domains is required and allows for a wide network of co-operating domains to communicate seamlessly. Authenticated subjects must be able to access restricted resources requiring multiple logins and credentials without the need to login at each domain. The IAM solutions access manager solution must not be based on a proprietary cookie. It should be based on SAML 2, which is a well-accepted industry standard for SSO. Using SAML2 allows the IAM solutions access manager to not only provide SSO capability at the web application tier, but also across other layers such as Web Services in a completely unified way. SSO must also allow the access manager to integrate easily with existing authentication technologies that are deployed in any organization.	MUST	Deployment time
6.2.29 Federation When GovStack is deployed it will need to be deployed with partners, suppliers and other organizations. For them to collaborate effectively, identity information needs to be propagated. The IAM solution must be able to manage the processes for federating users when a partner site comes on board or leaves. Federation capabilities must be provided by the IAM access manager solution. New cost recovery streams may be generated for GovStack users	MUST	Both

through enablement of trusted partnerships where authentication and authorization is carried out over federated business networks. Federation refers to interoperation between entities in different security domains, either in different organizations, or in different tiers in the same organization. A trust relationship must exist between the involved entities to federate identity and enable authentication across realms. Each domain may rely on different technologies and mechanisms to authenticate and authorize. Federation enables loose coupling at the IDM level separating the way each organization/application/module/building block does its own security implementation while they adopt a common mechanism to propagate identity.		
6.2.30 Security Token Service (STS) STS is a system role defined by the WS-Trust specification. A Web Service Client interacts with the STS to request a security token for use in SOAP messages. In addition, a Web Service Provider interacts with an STS to validate security tokens that arrive in a SOAP message. An STS arbitrates between different security token formats. The token transformation capability defined in WS-Trust provides a standards-based solution to bridge incompatible federation deployments or web services applications. Web service providers should not be required to support multiple authentication mechanisms even though they have to work with different web service clients. The SAML standard is well recognized and the IAM solution must provide a Security Token Service that can validate SAML and SAML2 tokens to bridge different web services.	MUST	Deployment time
6.2.31 Role and Attribute Based Access Control (RBAC/ABAC) The IAM solutions Access Manager must manage Groups, Roles, Permissions and Resources supporting both RBAC and ABAC. Groups are generally used to model organizational structure whereas Roles are used to model a person's function within the enterprise. In RBAC, a subject is given one or more roles depending on the subject's job.	MUST	Deployment time

Access is determined by the subject's role. In ABAC (Attribute Based Access Control), access is determined by the attributes of the subject (person or entity), attributes of the resource being accessed, environmental attributes and the desired action attribute. ABAC is implemented based on the XACML specification with: ○ Coarse-grained access control - based on subject, role and permissions ○ Ease of administration - roles created for job functions ○ A subject that must be assigned to a role and execute actions that are authorized for the role ○ Assigned permissions for job functions based on operations rather than to resource objects ○ Enablement of the creation of: ■ Relationships between Users, Groups, Roles, Resources ■ Creation and enforcement of policies Developing an access control strategy based on RBAC provides a clean and flexible model that is easy to maintain over a long period of time. Policies may be associated with a person's role. For example, someone in a medical advisor role may be permitted to access applications pertinent to his or her role, but not permitted to access applications related to someone in a doctor's role.		
6.2.32 High Availability The solution provided and any associated components MUST be highly available and utilize clustering technology in order to provide a minimum of 24x7x365 service with 99.99% availability (AKA 4 9's).	MUST	Deployment time
6.2.33 Open Source Offering The offered solution MUST be based fully on open source components. The vendor may offer subscriptions for support so long as the offered solution does not require those subscriptions in order to deliver the core functionality specified in this document.	MUST	Building time

6.3 Service APIs

This section describes external APIs that must be implemented by the building block. Additional APIs may be implemented by the building block (all APIs must adhere to the standards and protocols defined), but the listed APIs define a minimal set that must be provided by any implementation.

All APIs will be defined using the OpenAPI (Swagger) standard. The API definitions will be hosted outside of this document. This section may provide a brief description of required APIs. This section will primarily contain links to the GitHub repository for OpenAPI definition (yaml) files as well as to a website hosted by GovStack that provides a live API documentation portal. The basic assumption here is that the IAM suite will be acquired and not built. The suite MUST supply an appropriate API with documented endpoints.

IAM Suite API: An example of such a REST API is provided with the OpenIAM suite specification here: https://www.openiam.com/products/identity-governance/features/api/. This is intended as an example of the API style that is expected in such a suite. This is not definitive and may vary based on the proposed IAM suite. The detailed API documentation for OpenIAM including interface specifications can be found here: https://docs.openiam.com/docs-4.1.14/html/API/index.htm

OAuth2 API: The following link describes how OAuth2 is used in OpenAPI 3.0 standards based identity and access: https://swagger.io/docs/specification/authentication/oauth2/.

SCEP API: A description of the OpenXPKI enrollment workflow and API can be found here: https://openxpki.readthedocs.io/en/latest/reference/configuration/workflows/enroll.html. This is an example of how an API for enrolment with its associated workflow should be implemented within the Certificate Authority Server.

LDAP API: A description of the standard REST LDAP API provided by the open source 389 Directory Server here: https://directory.fedoraproject.org/docs/389ds/design/ldap-rest-api.html#ldap-rest-api . This is an example of how an open API for credential storage and retrieval using LDAP should be implemented.

6.4 Workflows

A workflow provides a detailed view of how this building block will interact with other building blocks to support common use cases.

This section lists workflows that this building block must support. Other workflows may be implemented in addition to those listed.

No specific workflow definitions are required for this building block as they will be inherited by the tools/products chosen to address each security issue/concern and fundamentally deal with the other building blocks API's in a cross-cutting manner as in the case of API Management.

Other components of the Security BB such as the IAM Suite will also provide their own workflow tools but the details of the actual workflow need to be designed once more is known. An example of the type of default workflow provided for an IAM suite would be the basic workflow for provisioning new accounts which can be leveraged by the other BBs. This is typically a basic workflow built in a tool that can be customized to meet specific provisioning needs (perhaps with multiple administrative roles and connections to multiple external systems and modules etc.) . The following link describes an example the basic workflow for provisioning provided by Open IAM (one of the alternatives) ***https://www.openiam.com/products/identity-governance/features/provisioning/***

Note that each building block is responsible for the defining base configurations and workflows required to be created in the IAM system that enable identity and access to be provisioned. Essentially the IAM system needs to be augmented with adapters to enable identity and access to be provisioned to target systems and resources in the way that the target applications implement their own identity and access. Alternatively where applications are built from the ground up they can leverage the IAM suite sAPI services to implement authentication and access.

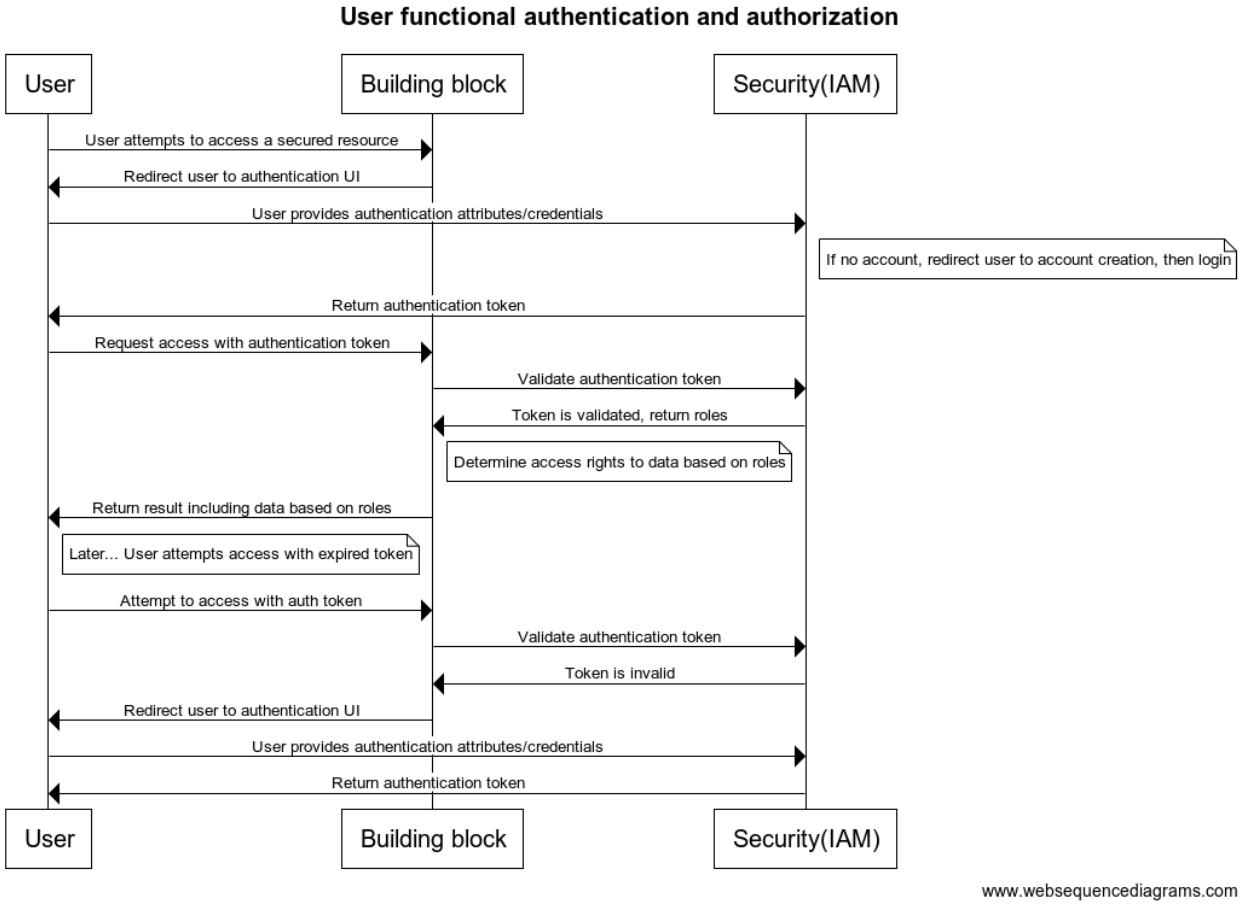
These specific workflows and adapters can be defined at the detailed design stage and communicated to the Security WG for implementation in the IAM solution. It must be noted that the security WG is NOT responsible for determining the identity and access policy and the details of access for each role for example. The following need to be identified by each building block and communicated to the Security WG for implementation in the IAM suite configuration build:

- **Resource Types**: for example files, services, API's, applications, modules to be protected by IAM.
- **Resources**: The definitions of the actual resources and their type provided by and required by each BB that are to be secured through IAM. This must include the target system or component that hosts these resources so that the correct provisioning adapter can be configured for that resource. Note that where the BB or resource has its own identity and access scheme an adapter can be written using the IAM suite API.
- **Roles**: the roles of users of each system that include the required resource access for that role in terms of the **Resources**. Each BB must account for the access they require to be provisioned to other services as a part of their process scope. In the case of provisioning a new account there is a need for a broader workflow process that is outside the scope of the Security BB which incorporates the Identity and Registration BB. For example a Doctor role may require verification and validation of certain aspects of identity prior to provisioning access to a specific service. A basic set of sequence diagrams is provided below to reinforce the understanding of what is required.
- **Approval Workflows**: workflow for the approval of the various identity and access requests (complete with approval roles etc).

6.4.1 Identity and Access Sequences

The following sequence diagrams depict the basic means by which authentication and access control shall be implemented across building blocks. Note that by definition an account with no access can be created by any user. via self-registration. Self-registration using basic phone/email as well as strong self registration using foundational ID are to be supported . A sequence detailing technical authentication is also included below along with sequences defining the remaining aspects of the identity lifecycle (such as provisioning and deprovisioning of access) to be supported by the IAM suite and how they will be leveraged by all building blocks. Such workflows can be articulated in full during the detailed design phase:

6.4.1.1 User authentication and authorization

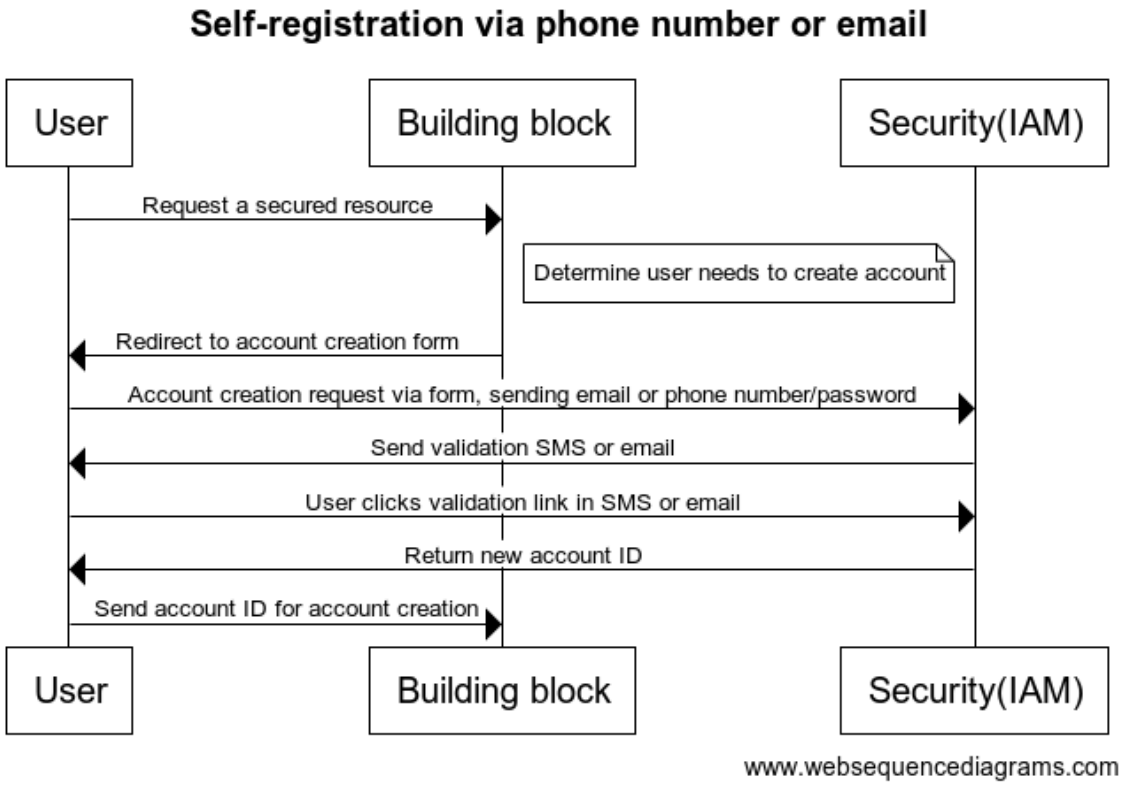


See https://www.websequencediagrams.com/ for an editable diagram

This assumes the user already has an account. Authentication credentials are username or phone number and password.

The auth token could be signed with an expiration (JWT) which might allow the BB to perform the validation themselves. Additionally, if the token contains roles and/or a user ID and isn't expired the BB could potentially rely on those.

6.4.1.2 Self-registration via phone number or email



See https://www.websequencediagrams.com/ for an editable diagram

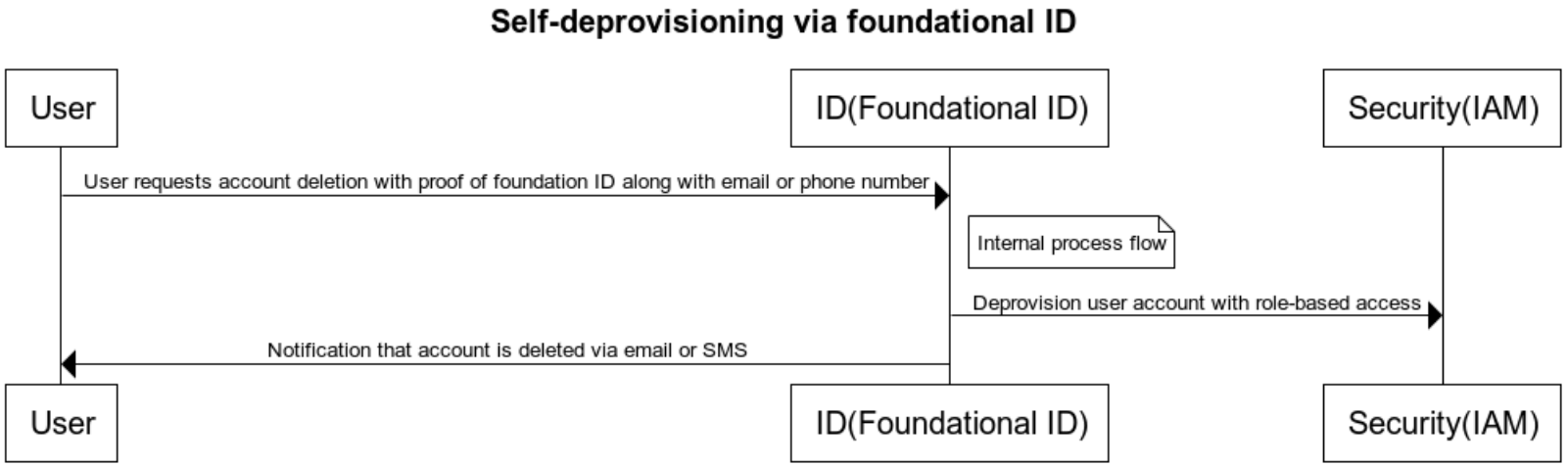
6.4.1.3 Self-registration via foundational ID

For a specification, see the Identity and Verification Building Block Specification.

Note that role creation (e.g. farmers, doctors) is handled by the IAM solution, via either an administration UI or an API. Building blocks can create roles via the API to provision new roles.

It's assumed the user clicked a link to access the service in the Building block UI.
The users are authorized with a valid access token for their email or phone number.

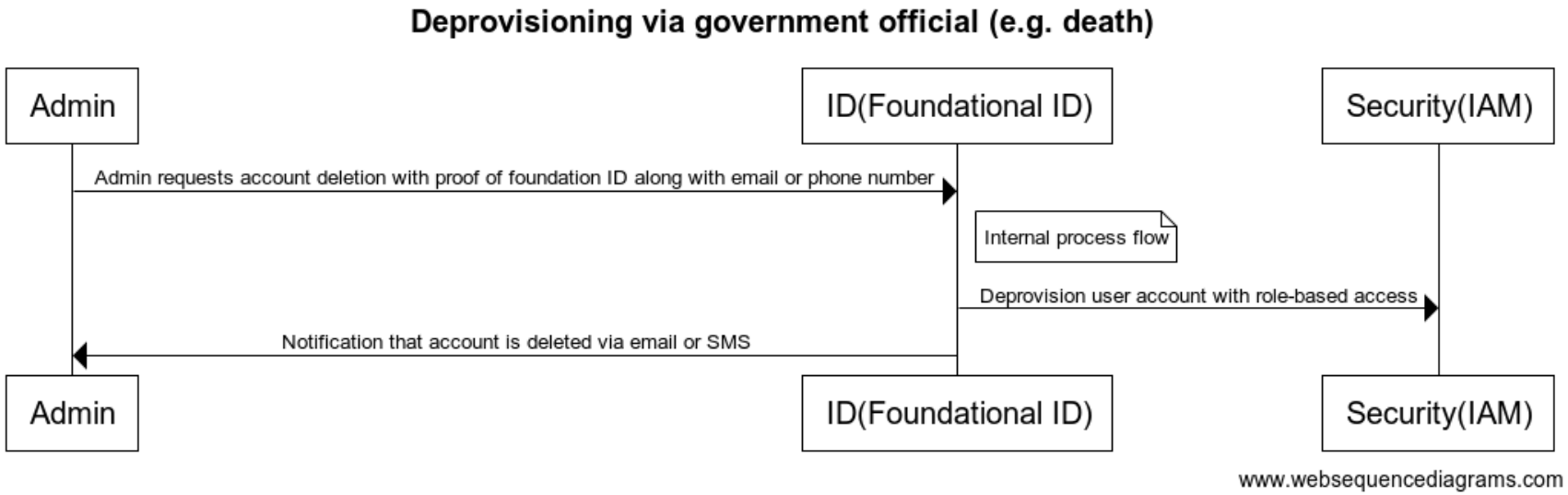
6.4.1.4 Self-deprovisioning via foundational ID



See https://www.websequencediagrams.com/ for an editable diagram

This flow assumes the user has an account and is currently authenticated.

6.4.1.5 Deprovisioning via government official



See https://www.websequencediagrams.com/ for an editable diagram

This flow assumes the user has an account and is currently authenticated.

6.4.2 Standards

The workflows MUST adhere to all standards defined in this document as well as in the GovStack architecture document (link to appropriate section in architecture document)

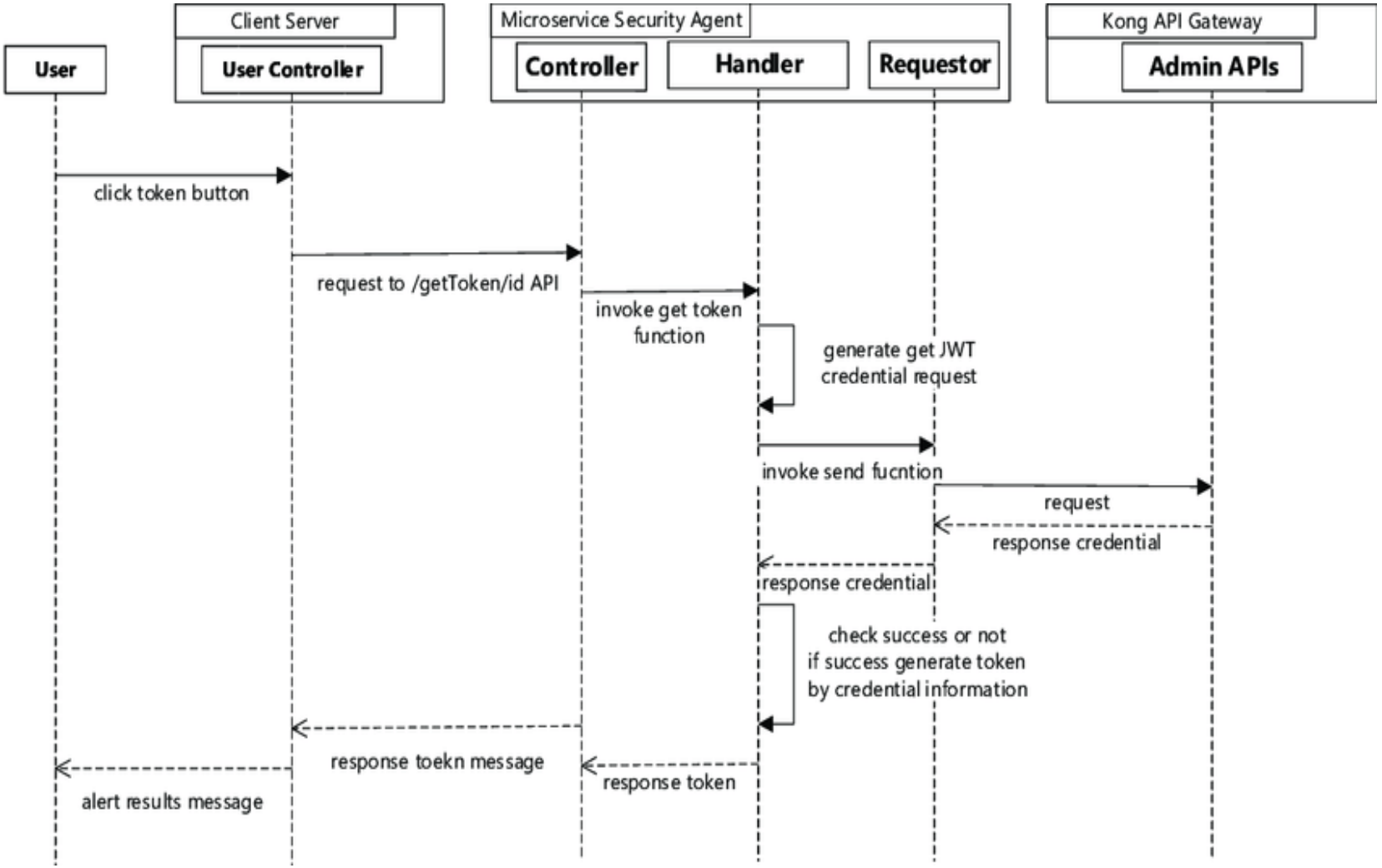
No specific standard workflows are required in the context of the security building block. All workflows involving identity and access for example are the responsibility of each building block working group and to be defined during detailed design. Such workflows can and should be implemented using the workflow engine built into the IAM suite and perhaps extended from or to a meta-workflow defined in the context of another BB.

6.4.3 Interaction with Other Building Blocks

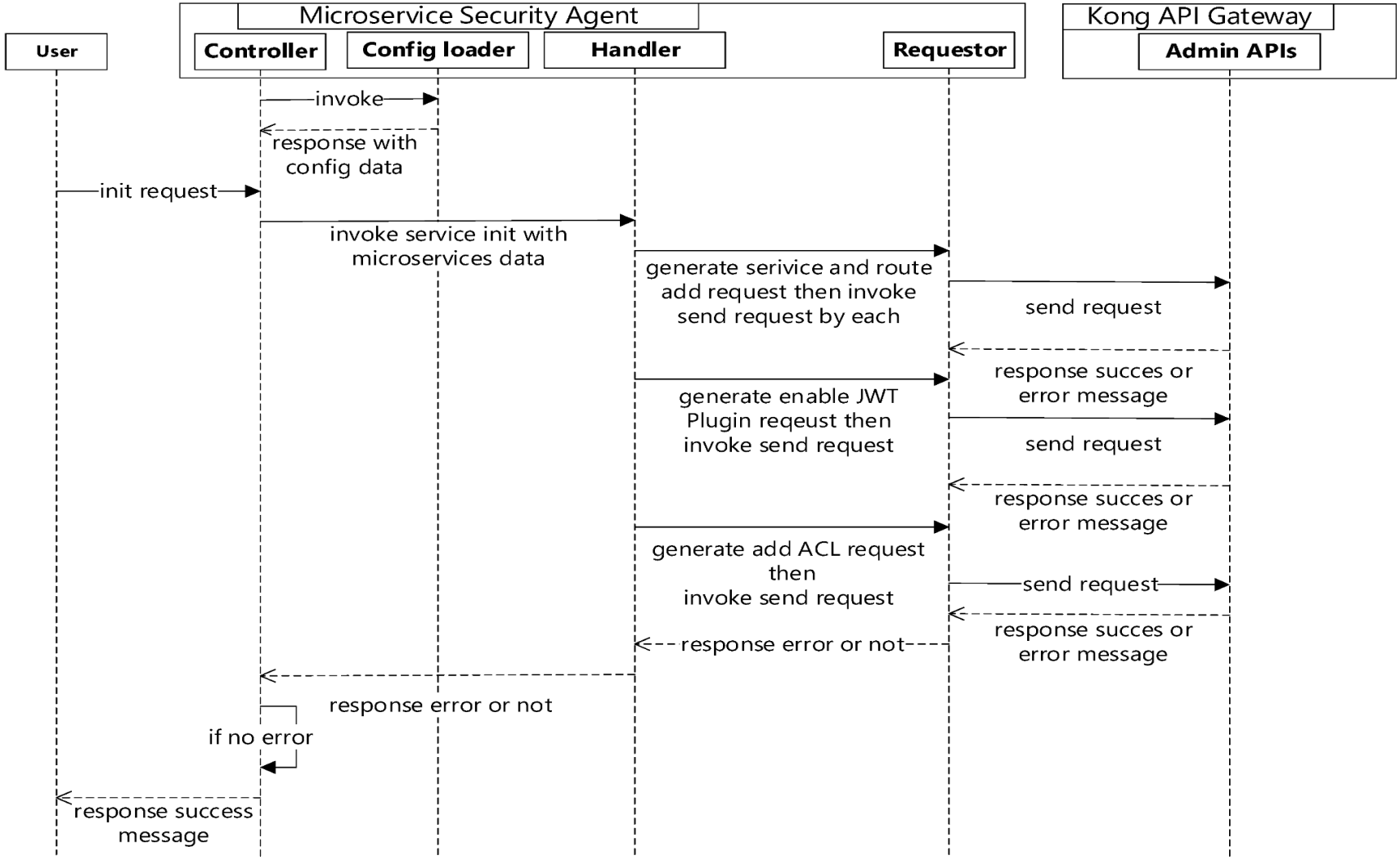
The security building block predominantly deals with the cross-cutting security concerns of each other building block and defines the basis for the implementation of solutions to address these concerns. The only interaction required is for API Management and Gateway services. These interactions are depicted and documented in Architecture Blueprint and Functional Requirements (see Ref 1).

6.4.4 Example Sequence Diagrams for API Gateway Services

The sequence diagrams below depict examples of how a building block might interact with the API Management and Gateway solution. This is only relevant for the API Management and Gateway services in the context of the security building block. A higher level sequence diagram depicting API interactions for building blocks is depicted and documented in Architecture Blueprint and Functional Requirements (see Ref 1).



Example Sequence for Issuing a Token for API Access



Example Sequence for Calling an API Microservice via Gateway

GovStack

7 Standards

The following standards are applicable to all aspects of the security building block and cross-cutting across other building blocks. Note that these are not technical standards but the process framework standards that shall be used to guide security decisions on the project::

- All of the implementation processes and guidance MUST follow the NIST CyberSecurity Framework (see Ref 2)
- All of the security issues and concerns to be addressed are related by number to the core requirements above. The detailed definitions can be found in the document entitled Digital Platform Security for GIZ, ITU DIAL GovStack (see Ref 5)
- It is assumed that the maximum level of information security required is what is known as CUI (Controlled Unclassified Information). Processes dealing with CUI must conform with NIST SP 900-171 Rev.2 (see Ref 3)

8 Cross Reference Links

See references section above...

Ref #	Name	Author	Date	Version
1.	Architecture and Nonfunctional Requirements	ITU, GIZ, DIAL	Mar o9, 2022	1.1.0
2.	NIST CyberSecurity Framework https://www.nist.gov/cyberframework/new-framework	NIST	Feb 15, 2018	1.1.0
3.	NIST SP 800-171 Rev.2 https://csrc.nist.gov/Projects/protecting-controlled-unclassified-information	NIST	Apr 20,2021	2.0
4	Cloud Customer Architecture for API Management https://www.omg.org/cloud/deliverables/CSCC-Cloud-Customer-Architecture-for-API-Management.pdf	OMG (Cloud Management Customer Council)		1.0

5	Digital Platform Security Requirements for: GIZ, ITU DIAL GovStack https://docs.google.com/document/d/11Jofvxb418iCvKGzCJuOAvSUFF2eMGowJkn5ooe_k6Y/edit?usp=sharing	ITU, GIZ, DIAL	May 9, 2021	1.0

9 Key Decision Log

Decision	By	Date	Description
Security related requirements to remain in this document	Hani and Max	4th August 2021	It was suggested that since security is more cross cutting concerns and some components that the cross cutting concerns should be relocated to the architecture WG document. It was decided that all security related requirements are to remain in this document and shall be cross referenced from the other BB definition documents.
Example API specification references added for each component	Security WG	4th August 2021	It was decided that an example API definition for each functional component of the security solution where applicable was to be added to this document for clarity. The basic assumption here is that the base of security components required to address all cross cutting concerns will be off-the-shelf open source solutions. We do not anticipate building a substantial suite of API's when the components should all be standards based and integrated. It would be far more costly for the project to engineer security components from the ground up.
A resource model for the IAM aspects of the security requirements is to be added for clarity	Security WG	4th August 2021	It was decided to add a resource model for the purposes of determining the suitability of various IAM suites for the project. The resource model can be used as a basis to determine which entities the solution supports.

The decision to include SAML in the requirements was taken	Security WG	4th August 2021	The purpose of including SAML is that federation of legacy environments is likely required and we cannot predict this in advance as we have no specific country context and must address a generic set of requirements that can be adapted to all settings.
Added sequence diagrams for describing account creation, processing service access requests and the basics of API gateway operations.	Security WG	11th August 2021	Met with the registration BB and we discovered that their design model required accounts to be created with a minimum of email and/or mobile number. This meant that the provisioning of access to services needed to be a separate process. So these two processes were detailed in sequence diagrams
Added basic API examples for the purpose of clarifying the intent of how the IAM solution will deliver authentication and authorization services.	Security WG	11th August 2021	There have been many questions from various WGs about how the basics of authentication and authorization will function. Some description of this has been added to the document along with examples of how OpenIAM implements both REST and OAuth2 authentication and authorization.
Added a high level resource model	Security WG	11th August 2021	A high level resource model based on a domain driven design has been added to help clarify the overall context of the Security BB.
Reviewed and modified sequence diagrams	Security and Architecture WG	19th August 2021	Using an in-depth discussion with the Registration BB we modified all of the sequences for account creation and provisioning for consistent use across all BBs.
Renumbered requirements items for ease of reference	Security WG	20th August 2021	All of the detailed requirements have now been indexed with numbered headings and are available as links through both the ToC and to other documents. This was done for ease of reference by the other BBs.

Accepted Suggestions	
This document has no mention of how this BB is related with the "Security BB" available at https://solutions.dial.community/building_blocks/security?	Accepted and cross link inserted in the introduction section of this document
BB module (see Ref 1) in the above sentence on page 116, and occurring twice on page 151 may be hyperlinked, within the document, to the URL at Ref#1, under section 6: Cross reference links, on page 154	Hyperlinks added for all referenced standards.
For (see Ref 2) and (see Ref 3) may be suffixed to "NIST CyberSecurity Framework" on pages 19-20, and "NIST 800-171 Rev.2 standard" on pages 20. Both the suffixes may be hyperlinked, within the document, to the respective URLs at Ref#2 and Ref#3 respectively, under section 6: Cross reference links, on page 154.	Hyperlinks added for all referenced standards.
The Section 4: Security Building block modules: (Suggestions: (see Ref 1) in the last sentence of the opening paragraph, on page 116, "Explicitly, (see Ref 1) the communications between all building blocks (BB's) and applications shall be via open API based access.", may be corrected. (see Ref 1) in the above sentence on page 116, and occurring twice on page 151 may be hyperlinked, within the document, to the URL at Ref#1, under section 6: Cross reference links, on page 154. 5.11 Denial of Service Attack Prevention Requirements 1. Under 5.11.3 Volume Based Attack Prevention:, the word "Which" in the phrase "1) Attack Prevention and Preemption: Which is done" of the second last line, on page 57, may be replaced by the word "This", to maintain uniformity with the description of the following two ways. 2. Under 5.11.4 White Lists:, on page 58, the 'usr' in the phrase "normal usr operations" may be reviewed.	Thanks for suggestions, corrections adopted
Out of Scope Suggestions	

These 59 security requirements could be mapped to the 20+ BBs as listed in the DIAL document, to show which requirement applies to what BB. In countries with low or limited resources (our target), are all these requirements essential? Aren't these overkill? We are working towards minimum viable architecture (MVA).	This document compiles recommendations to be referenced by each building block specification to make these principles part of the reference implementations, as early as possible. The list is comprehensive, but picking specific recommendations of this list that match MVA constraints of a target country will be an implementation time team decision and not in scope of this document
The purpose of yellow coloured text highlighter used for “4.4 Workflows”, on page 144, and for “4.4.4 Example Sequence Diagrams for API Gateway Services:”, on page 151, may either be explained, or the highlighted text may be formatted to “No color”.	The yellow text is how Google Docs indicates that a reviewer has placed a comment at that point. The released documents will be PDF files and will not display the comments.
Overall across the Architecture and the e.g. ID and other BBs, the ownership of ID management is unclear. There seems to be an aspiration for each BB to be self-sufficient yet for example if a government were to adopt all BBs would there be multiple ID management components? There is mention of a "registration server" in the ID BB & a "Security server" in Architecture BB. It probably deserves a conversation e.g. is there going to be a centralisable Directory serving each BB?	The foundational root of trust for legal person ID is with ID BB. Functional IDs Without exposing this ID, token IDs can be generated by the ID building block for use in different building blocks. For example, a token id for payment purposes only, may reside inside Payments BB of the finance ministry for a mapper that links the ID to specific payments accounts of a person. A different token Id may be issued for the health sector, for linking a person to a health record repository in a health ministry application. One can further opt to issue token IDs derived from the root, to associate with only a particular program (e.g mother and child care program) within a sector (e.g healthcare) or even a particular entity (e.g. a bank) authorized within a program. The security server is the implementation of IM BB. There is a security server in front of every BB (bridge to internet). The process of registration is handled by the Registration BB. Different part of information collected during registration may be parked and protected inside specific BBs that own the jurisdiction of respective data
The purpose of yellow coloured text highlighter used for “AI/Deep Learning (Collaborative)” under “5.23.2 General Features Required:” on page 85, and	The yellow text is how Google Docs indicates that a reviewer has placed a comment at that point. The released

the text below “5.27.3 Example REST Authentication API”, on page 97, may either be explained, or the highlighted text may be formatted to “No color”.	documents will be PDF files and will not display the comments.
The Section 2: Introduction (Suggestions: (see Ref 2) and (see Ref 3) may be suffixed to “NIST CyberSecurity Framework” on pages 19-20, and “NIST 800-171 Rev.2 standard” on pages 20. Both the suffixes may be hyperlinked, within the document, to the respective URLs at Ref#2 and Ref#3 respectively, under section 6: Cross reference links, on page 154. Necessary action may be taken for the TBD entries in the table against S. Nos. 6, 7, 13, 14, 15, 16, 17, 18, 20, 22, 25, 33, 34 under the column “Description, Discussion and Potential Solutions”. - (see Ref 3) may be suffixed to “NIST 800-171 Rev.2”, mentioned against S. No. 58, under the column “Description, Discussion and Potential Solutions”, on pages 36. The suffix may be hyperlinked, within the document, to the URL at Ref#3, under section 6: Cross reference links, on page 154. The Section 5: Cross-functional requirements 5.1 Privacy (see Ref 3), on page 38, may be hyperlinked, within the document, to the URL at Ref#3, under section 6: Cross reference links, on page 154.) 5.6 Certificate Authority Functional Requirements (see Ref 1) at the end of 5.6.9 Standards Based API Interface, on page 44, may be hyperlinked, within the document, to the URL at Ref#1, under section 6: Cross reference links, on page 154. The Section 5: Standards: (see Ref 2), (see Ref 5) and (see Ref 3), on page 154, may be hyperlinked, within the document, to the respective URLs at Ref#2, Ref#5 and Ref#3 respectively, under section 6: Cross reference links, on page 154.	These changes may become redundant as the document will be realigned with appropriate sections to reduce the number of repeats and hyperlinks.

GovStack

10 Future Considerations

Suggestion	Response
Is there a reason for only "exposed"? API's should be externalizable and should not rely upon security outside their control.	Presently we are not considering securing APIs that are internal to a building block within scope. However, based on specific implementation requirements and internal components required, this may be specified in future scope.
From a conversation with Kristo: some of these requirements apply to the deployment/project, and some map more directly to building blocks and engineering requirements. Perhaps we should separate these into two groups	Consider separating into sections. This will help focus the building blocks on just the requirements that apply to the product at this point. We can address the others when it comes to deployment time and in like the networking BB spec.
Security document contains sections that can be moved to architecture document e.g API design principles, NFRs like availability, scalability, etc.	This will be normalized with the Architecture document in future release
Operational best practices like VAPT, infra security should be separated with the design/development time general security principles.	Different requirements have been tagged as design/build/deployment time requirements. This will be called out in separate sections in next release.
Document is too exhaustive and it would be better to keep key summary check list for all the BB to easily follow.	Security check list should match the selected feature list based on infra constraints. An example check list can be prepared in future release for reference to develop such checklists suitable to a target implementation
Some of the sections are repeating themselves. It will be better to reorganize these to core sections. Fine, better to understand	In the next release re organize the sections to minimize repetition and introduce cross links as needed.

At the beginning, a one-page pictorial and summarised representation of how GovStack approach	A generalized introduction will be provided in Govstack website and all building blocks will carry links from there. Links to general introduction page can also be inserted the for cross reference from this document
Separate documents for Security building block specifications and security guidelines to keep focus on general security guidelines and security building block features for different audiences.	The security recommendations and Security BB specs are currently separated into different sections. In next release these will be separated into different documents, for specific end use - for adopting security principles in build building block development and for implementing security building block in a network.
At the end, a section on as many use cases of this BB-based implementation solution, another on success stories using this BB, and lastly a tutorial with exercises for capacity building of existing and prospective Government executives. It can be used as a module of the "Digital Transformation in Government" course/ module for management/ B-schools and administrative training colleges/ institutes of the Government.	Links to case studies, examples and demos will be inserted as an ongoing process in future releases
The GovStack security solution requires a credential store as a centralized infrastructure for hosting the user account and credentials defined such that the IAM solution and other components such as the API Management and Gateway solutions can leverage them. This may end up being embedded in other solutions such as IAM or potentially implemented as a separate repository such as LDAP. - Needs alignment with Arch Document	Put details and examples for how this credentials created and used by other systems. Created in security building block.