GEA-ET 2.0 – Government Enterprise Architecture Update and Roadmap

2024-2029

Advancing a Connected, User-Centred, Data-Driven, Open and Secure Digital-First Government

Connected User Centred Data-Driven Open Secure

The GEA-ET 2.0: Government Enterprise Architecture Update and Roadmap was financed by European Union – Project No. 300052766 – SIEA-2018-14397 “Prepare e-Government Strategy and Enterprise Architecture for Ethiopia (2023-2027/28)”.

The project was implemented by:

DT Global

In consortium with

OCA GLOBAL

DAI

Disclaimer: The contents of this publication are the sole responsibility of **DT GLOBAL IDEV Europe SL** and cannot be taken to reflect the views of the European Union.

Acknowledgements

The Digital Government Strategy and Implementation Roadmap and the Enterprise Architecture were commissioned by the Government of the Federal Democratic Ethiopia and funded by the European Union.

His Excellency Dr. Belete Molla, Minister of Innovation and Technology guided the development of the Strategy and Implementation Roadmap and Enterprise Architecture. Dr Abiot Sinamo, National Electronic Government Development CEO, oversaw the project.

The Ministry of Innovation and Technology counterpart team that provided ongoing coordination and substantive support included Biruhtesfa Taye, Fekade Getahun, Admas Mamo, Helen Mergia, Miftah Jemal, Babesha Kenaw, Meklit Teshome and Bethlehem Bedilu. We thank them all for their dedication and ongoing support to the fruition of this project. The Digital Government Strategy and Implementation Roadmap and Enterprise Architecture have also benefited from consultation and guidance of different government institutions, the private sector, academia and development partners that have provided invaluable input and feedback.

The Strategy and Implementation Roadmap and the GEA-ET have also benefited from the consultation and guidance of the EU Delegation in Ethiopia. Iona and Leul Habte provided the project's overall coordination and substantive guidance. We appreciate the support of the European Union Delegation.

The analysis and writing of Government Enterprise Architecture Update and Roadmap was carried out by Dr. Mweene Monze (Distinguished Architect) and managed by the DT Global Consortium. Dora Gombocz of DT Global, Lluis Palau Meseguer of OCA Global and Vesna Kahrimanovic of DAI provided the coordination and technical support to the project. Lishan Adam (Team Leader), Samuel Cahiri, Constantine Bitwayiki, Vladimir Abramytchev, Mouad Boumahdi, Abraham Kibret, Gezahegn Chalchisa and Mesay Hailemariam provided peer review of the Government Enterprise Architecture Update and Roadmap.

Contents

Figures

Tables

Abbreviations

Term	Description
AAU	Addis Ababa University
ADM	Architecture Development Method
API	Application Programme Interface
BA	Business Architecture
CA	Continuous Availability
CSF	Cybersecurity Framework
DevOps	Development and Operations
EDGS	Ethiopian Digital Government Strategy
ENDS	Ethiopia National Data Set
DRP	Disaster Recovery Plan
EA	Enterprise Architecture
EAF	Enterprise Architecture Framework
ENEAF	Ethiopia National Enterprise Architecture Framework
EGIF	Ethiopian Government Interoperability Framework
ESB	Enterprise Service Bus
DEF	Digital Ethiopia Factory
FDRE	Federal Democratic Republic of Ethiopia
GEA	Government Enterprise Architecture
GEA-ET	Ethiopia Government Enterprise Architecture
GEA-NZ	New Zealand Government Enterprise Architecture
GEAF	Government Enterprise Architecture Framework
GoE	Government of Ethiopia
IAM	Identity and Access Management
IT	Information Technology
ITIL	Information Technology Infrastructure Library
ITSM	Information Technology Service Management
MDA	Ministries, Departments, and Agencies
MInT	Ministry of Innovation and Technology (Ethiopia)
MVA	Minimum Viable Architecture
NIST	National Institute of Standards and Technology (USA)
PA	Planned Availability
PDP	Public Digital Platform
SDLC	Solution Delivery Life Cycle
SOA	Service Oriented Architecture
SOA-RA	SOA Reference Architecture
SVS	Service Value System
TOGAF	The Open Group Architecture Framework
WoG	Whole-of-Government

1 Executive Summary

This report documents the results of a project undertaken to develop a government enterprise architecture (GEA) for Ethiopia. This work represents the second significant effort to execute such an undertaking, following the GEA development work reported in 2011 [@] and continued in 2019 [@]. The present work is referred to in the sequel as GEA-ET 2.0 (or simply GEA-ET).

The diagnostic assessment undertaken for GEA-ET found that frequent restructuring within and across MDAs has been the single most significant historical impediment to GEA development in Ethiopia, since each change has created discontinuities and forced multiple restarts in the coordination and execution of GEA activities. Inadequate governance and skills, technical complexity, and the lack of concrete implementation guidance and action plans also ranked highly as critical GEA delivery failure factors.

Whilst GEA-ET suggests mitigations for MDA restructuring risks, it really focuses on addressing issues that fall within architectural spheres of influence, such as governance, skills, complexity, and planning:

- Governance: Governance refers to an existing baseline (comprising architectures, standards, principles, plans, etc.) and how conformance to that baseline is maintained. In this case the GEA-ET represents the 'baseline' and role of GEA-ET governance is to ensure a high degree of ongoing fit between the GEA-ET and the operational and strategic imperatives of the Ethiopian government. The GEA-ET diagnostic found that there have been consistent failures to establish well-defined governance structures and processes.

  The architectural governance issue requires careful attention because multiple governance structures have been proposed and sometimes initiated in the past, but have not been established and sustained (e.g., steering committees, architecture boards, governance, technical and expert councils, and the like).

- Competence: The acquisition, development, and retention of GEA-ET skills is a pervasive challenge at all levels of government, particularly at the regional levels where architectural competence is virtually non-existent. Effective structures for hosting and developing architectural leadership currently do not exist, which is problematic because such leadership is required to ensure that GEA-ET initiatives consistently receive the right prioritisation and resourcing levels.

- Complexity: Based on the proposition that complexity is intrinsically a function of structure, dynamics, scope and scale, it is evident that Ethiopia has a complex and evolving GEA-ET landscape. High architectural complexity creates dependence on advanced technical skills and it will likely discourage GEA alignment within less resourced MDAs, so the GEA-ET delivery strategy must be designed to deliberately mitigate complexity.

- Planning: The planning process must set out a delivery strategy that coordinates and executes the changes needed to realise the GEA-ET, whilst considering the bounding constraints determined by the broader EDGS programme.

  The target outcome should be more specific GEA-ET implementation guidance than has been provided in the past, in the form of feasible resource plans, which define the structures and processes needed to execute the strategy, and viable delivery schedules (or roadmaps), which organise initiatives along timelines that realistically match delivery effort with resource capacity and availability.

Building on this understanding, the developed GEA-ET specification consists of five main components:

- Development strategy that sets out the approach to addressing the above-cited primary concerns,
- Framework layer that specifies GEA-ET components at the framework level,
- Solution layer that specifies the common solutions required to support a Whole-of-Government (WOG) approach,
- Governance strategy that addresses the GEA-ET components of the overall governance plan for the EDGS, and
- Delivery roadmap that defines high-level GEA-ET resource and delivery schedules that align with the digital government strategy priorities and roadmap.

In this regard, it is important to note that GEA-ET guidance broadly delineates into three categories, depending mainly on the depth and quality of the diagnostic assessment findings on the GEA landscape in Ethiopia, and the extent to which guidance around a particular stakeholder concern must consider significant contextual factors like the proposed digital government strategy, for instance:

- When knowledge of the GEA-ET landscape is low and the need for context awareness is high, GEA-ET generally recommends that the affected concern be addressed [in due course] as part of the GEA-ET implementation phase, possibly also providing guidelines for developing the architectural specifications that will address the concern.

An example is the reference business architecture which must carefully consider strategy and landscape factors; the available landscape information is limited but GEA-ET provides extensive architecture development guidance. Another example is the reference DevOps (development) architecture for which it is prudent to acquire a better understanding of current tools and practices before specific architecture(s) can be recommended.

- When landscape knowledge is high and the context awareness requirement is low, GEA-ET provides very specific recommendations based on diagnostic assessment findings like stakeholder feedback, international best practices, and industry trends.

  For example, a detailed specification has been crafted for the reference integration architecture which draws on standards-based and universally applicable patterns; thus, it is generally not affected by deployment context.

- For concerns that provide adequate landscape and context understanding, GEA-ET attempts to develop detailed architectural specifications and implementation guidance. Specific instances are the proposed GEA-ET governance and capacitation frameworks which leverage extensive stakeholder feedback, TOGAF frameworks, and the author’s own experiences, to craft detailed governance- and capacitation-related architecture development and implementation guidance.

Consequently, in some aspects, GEA-ET is a framework that simply identifies concerns and then provides guidelines (as required) as to how to address them, for example, how to create a specific deliverable. In other aspects GEA-ET provides concrete specifications for deliverables like reference architectures and capacitation plans.

The point of this explanation is to highlight that the GEA-ET attempts to be knowledge-based and context-aware: it recognises that the nature of stakeholder concerns and the state of the enterprise must be considered to make valid architectural decisions, because no one approach is suitable in all circumstances. Such thinking needs to be carried forward as a best practice as the resulting GEA-ET develops and matures over time.

2 GEA-ET Strategy

In general, the first element of the GEA-ET strategy is about reformulating the traditional GEA framework as a hierarchy of abstract model layers or tiers that describe GEA-ET components at the framework, solution, and MDA levels. The second element is to initially treat the GEA-ET layers and their major components as independent entities that can be developed concurrently, and then to only map the relationships between them after they have been completed. The third element of the strategy is to prioritise pragmatism, in way that relaxes governance rigour for the purpose of maintaining GEA-ET delivery and adoption momentum through accelerated delivery of stakeholder value.

2.1 Content Model

Figure 1 shows a hierarchical GEA-ET structure which defines separate content specification tiers at the framework, Whole-Of-Government (WOG), and MDA levels, being a version of New Zealand's GEA-NZ model that has been adapted to align with Ethiopia's GEA-ET goals, objectives, and landscape.

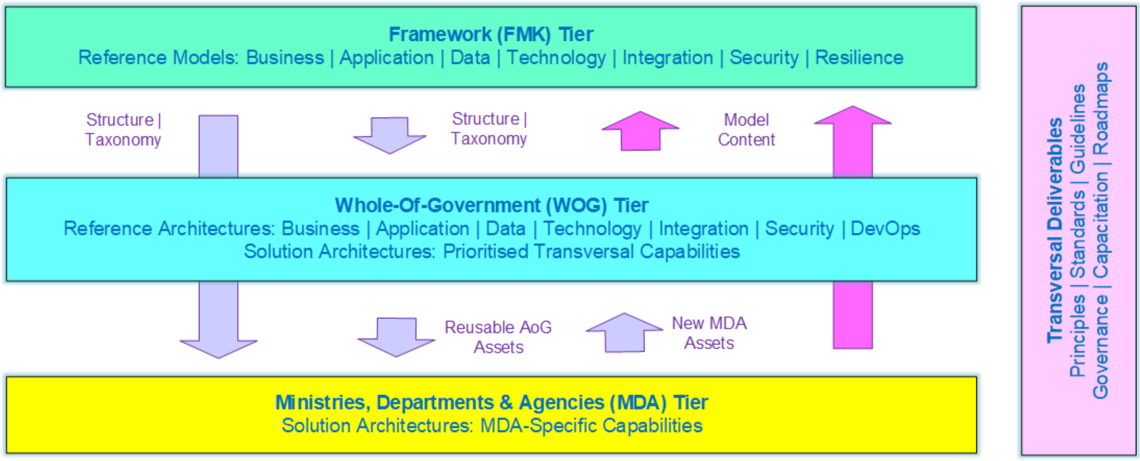

Framework (FMK) Tier
Reference Models: Business | Application | Data | Technology | Integration | Security | Resilience
Structure | Taxonomy
Structure | Taxonomy
Model Content
Whole-Of-Government (WOG) Tier
Reference Architectures: Business | Application | Data | Technology | Integration | Security | DevOps
Solution Architectures: Prioritised Transversal Capabilities
Reusable AoG Assets
New MDA Assets
Ministries, Departments & Agencies (MDA) Tier
Solution Architectures: MDA-Specific Capabilities
Transversal Deliverables
Principles | Standards | Guidelines
Governance | Capacitation | Roadmaps

Figure 1: Tiered GEA-ET Content Model

- Framework Tier: This is the top layer which enables the single unifying framework, information taxonomy, and structure for the standardised syntax and semantics used by the other two tiers in the model. It provides the overall model framework and contains the reference models, which are described below. Note that it does not contain content such as standards, common capabilities, or technologies. Further, it does not develop a performance reference model because the strategy if to use a integrated monitoring and evaluation platform for all digital government initiatives.
- Whole-Of-Government Tier: This is the central tier which is built on the structure and classifications inherited from the framework tier. It is also called the Solution tier because it hosts the reference architectures for the common solutions deployed to enable WOG capabilities. The WOG roadmap is also represented at this tier.

- MDA Tier: The bottom tier is designed for MDA use, including agency-specific standards, technologies and architectures. As with the WOG tier, the Agency tier inherits the structure and taxonomies from the framework tier to enable the identification of areas of commonality. Where applicable, the MDA model can reuse content from the MDA tier, and MDA tier models can provide content (e.g. reference architectures) that could be fed back for reuse at the WOG tier.

The GEA-ET strategy leverages this layered approach to partition the GEA-ET development process into three independent streams: framework, WOG, and MDA streams. Streaming addresses the complexity concern in two significant ways: (1) each tier can be linked to a separate and independent roadmap which is not dependent on the other tiers, (2) the delivery of GEA-ET capabilities can be prioritised within and across tiers.

The strategy proposes that the middle or solution tier be prioritised ahead of the other two tiers, since it contains the common capabilities that support a WOG approach, and will enable early delivery of stakeholder value. In terms of priority, the solution tier can then be followed by the framework tier which has to adopt a longer-term perspective because it must develop the reference models, identify and catalogue existing and proposed GEA-ET capabilities at the WOG and MDA levels, and then map the relationships between them. Within the solution tier itself, it is recommended that the business (process), data, integration, and security architectures take precedence in that order, because they underpin all GEA-ET capabilities.

The point to note is that reference model design and implementation is likely to be a multi-year process, given the high complexity of the GEA-ET landscape in Ethiopia, but it is not prudent to delay commencement of solution tier developments pending completion of this exercise. Rather, it makes more sense to concurrently execute the framework and solution tier roadmaps, and to also plan for periodic synchronisation of the framework and solution tier content, since this approach will support the rapid delivery of stakeholder value, which is important to maintain GEA-ET development momentum.

2.2 Frameworks, Models & Architectures

Architectural frameworks, reference models, reference architectures, and solution architectures are established constructs that can be used to provide different perspectives on an enterprise architecture. The GEA-ET description makes extensive use of these concepts, so it makes sense clarifying their meaning and the relationships between them upfront, to aid understanding of the GEA-ET specifications presented in the sequel. The GovStack Ecosystem Reference Architecture (GERA) (see Figure 2) provides an accessible visual description of models and architectures that can serve this purpose, which also aligns with the GEA-ET viewpoint on how these constructs should be positioned and interpreted:

- Architecture Framework: Although not explicitly depicted in the diagram, an architecture framework may be thought of as a methodology used to create and manage these constructs as part of an enterprise architecture. It provides an approach to describe and identify the necessary inputs to a particular architecture as well as a means to describe that architecture. So, architecture frameworks give architects the tools they need to adequately describe and collect requirements, without mandating any specific architecture type.

  An architecture framework describes an example taxonomy of the kinds of architectural "views" that an architect might consider developing and provides guidelines for making the choice for developing particular views. The TOGAF standard is one such framework and its "views" include a content framework, reference models, and an enterprise continuum (see also Figure 4).

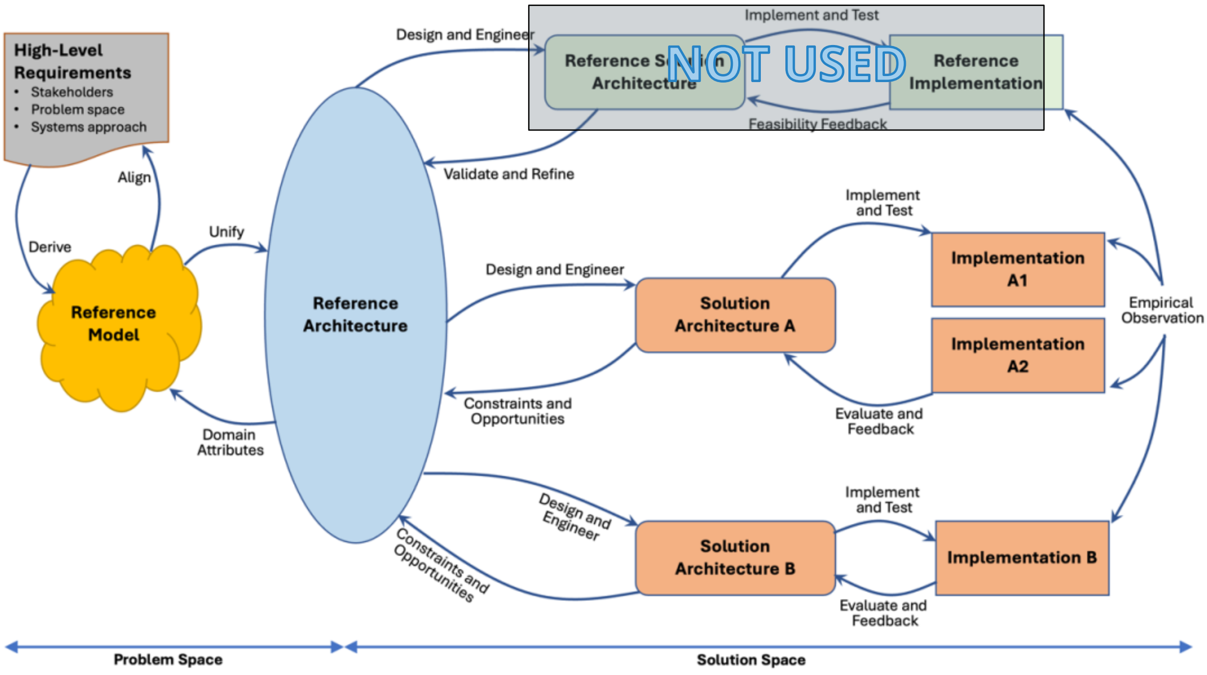


Figure 2: GovStack Ecosystem Reference Architecture: Reference Model Components

Source: The Open Group Guide: GovStack Ecosystem Reference Architecture (GERA) [@]

- Reference Model: Reference models provide abstract frameworks for understanding significant relationships among the entities of specific architectural landscapes, and for the development of consistent standards or specifications that support architectural work in those landscapes. A reference model is based on a small number of unifying concepts and may be used as a basis for education and explaining of standards to a non-specialist. A reference model is not directly tied to any standards, technologies or other concrete implementation details, but it does seek to provide a common syntax and semantics that can be used unambiguously across and between different implementations.

- Reference Architecture: A reference architecture [builds on the reference model] by providing recommended structures and integrations of IT products and services to form a solution. The reference architecture embodies accepted industry best practices, typically suggesting the optimal delivery method for specific technologies. An well-formed reference architecture should offer IT best practices in an easy-to-understand format that guides the implementation of complex technology solutions.
- Solution Architecture: A solution architecture describes a specific implementation of a reference architecture which may include identification of specific products and services and a description of the prescribed methods (guidelines and techniques) to be used to create an instance of that architecture.

Note that the GEA-ET does not make use of the reference solution architecture and implementation constructs that the diagram depicts as they are largely superfluous, and their GERA meaning is thought to be ambiguous.

2.3 Service-Oriented Architecture

Service-oriented architecture (SOA) defines a way to make software components reusable and interoperable via service interfaces. Services use common interface standards and an architectural pattern so they can be rapidly incorporated into new applications (see Figure 3).

SOA describes an information system model in which service consumers access services implemented by service providers in service layers through well-defined service interfaces. A system component can be both a service provider and service consumer, and it can consume or provide services within its own layer or other layers. The service interface specifies and provides access to all the services defined within the service layers, and system components within the service layers provide the implementation for, realization of, or operation on services.

Broadly speaking, SOA realisation consists in (a) identifying providers and specifying their service interfaces, (b) specifying the interaction patterns between consumers and providers, and (c) configuring providers to support their service interfaces.

The GEA-ET makes extensive use of the SOA layered model in two principle ways:

- Firstly, the SOA layers form the first-level classifiers of an informal taxonomy for categorising GEA-ET artefacts such as models, architectures, standards, and principles. The usage of this simple taxonomy is reflected in the tiers of the GEA-ET content model of Figure 1, which classifies content according to the SOA layers, as well as other GEA-ET models discussed in the sequel.

- Secondly, it serves as the highest-level abstraction for identifying, organising and relating architectural components at both the logical and physical levels. Thus, whilst the 'SOA taxonomy' classifies GEA-ET components, the 'internal' structures of the components themselves are expressed as reference solution architectures the WOG and MDA levels of the content hierarchy, which, in turn, serve as blueprints for solution implementation projects.

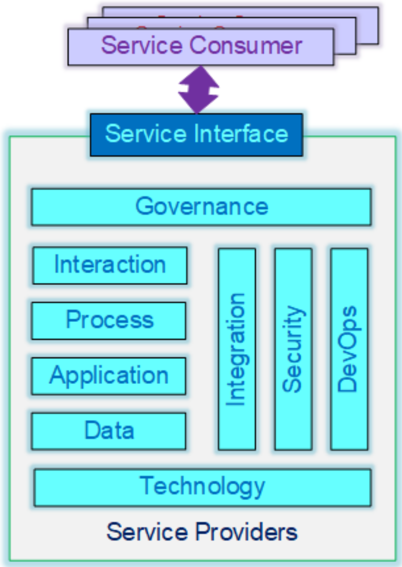
Perspective	Description	
Interaction	The capabilities to deliver functional requirements to service consumers (end-users) and interfaces for inter-application communication.	 Service Consumer Service Interface Governance Interaction Process Application Data Integration Security DevOps Technology Service Providers Adapted from http://www.opengroup.org/soa/source-book/intro/index.htm
Process	The composition and orchestration methods for aggregating loosely coupled services as business process driven service invocations. In this context, the process dimension also includes business or eServices ("services") which include functions that are made available to service consumers (end-users) as part of a portfolio of GEA-delivered digital government services.	
Application	The components that implement and execute the eServices which are integrated and orchestrated to deliver business processes.	
Data	The capabilities to search, access, analyse, and integrate information hosted in various data providers within and outside the enterprise.	
Technology	Includes network, compute (cores and memory), and storage that are configured to deliver quality-of-service (QoS) requirements.	
Integration	The mediation, routing and transportation of service requests and service replies, between service consumers and service providers.	
Security	Relates to the controls used to protect the confidentiality, integrity, and availability of services, including security threats associated with users, applications, data, and infrastructure.	
DevOps	A moniker for Development + Operations; operations refer to capabilities that realise specified qualities of service, whilst development refers to capabilities that support the service implementation lifecycle. DevOps manages the service life cycle, from creation to retirement, and monitors and assures specified qualities-of-service attributes, including function, capacity, availability, security, and performance.	
Governance	Architectural capabilities that develop and manage the GEA-ET and oversee its implementation and operation; such capabilities include domain- and enterprise-level design expertise, project/portfolio management, change management, and compliance oversight.	
NOTE: *The four horizontal layers are more functional in nature and relate to the functionality of the SOA solution itself. The vertical layers represent realisations cross-cutting concerns that span the functional layers but are clustered around independent notions themselves as cross-cutting concerns of the SOA architectural style*.		

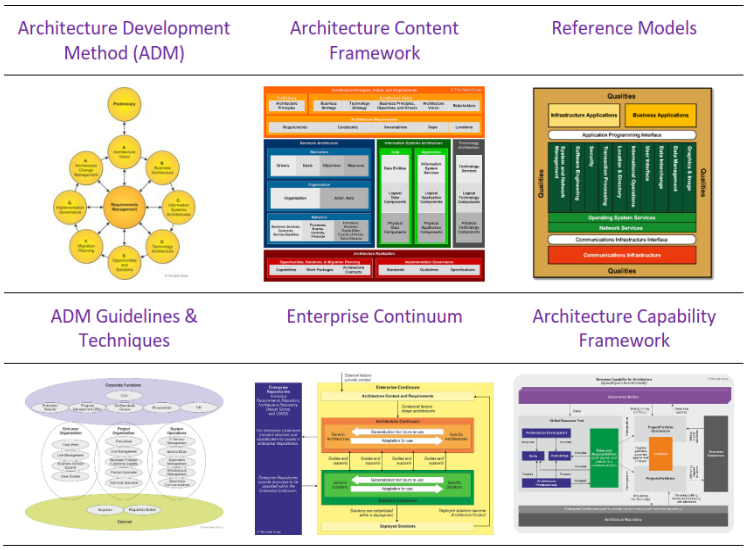
Figure 3: Service-Oriented Architecture: Conceptual View

SOA enables a less formal approach to classifying and organising GEA-ET components, which is expected to be effective in the initial stages of GEA-ET development for two main reasons: (a) early efforts of the GEA-ET initiative can focus on solution level concerns whilst the necessary GEA-ET architecture development and governance capabilities are established, and (b) SOA's layered service structure is particularly suited to organising and communicating solution level specifications in an accessible and informal (non-normative) way.

To validate the ongoing viability of SOA as an architectural reference model, decision gates can be placed at appropriate junctures in the GEA-ET roadmap, to trigger assessments as to whether adoption of formal (normative) forward reference models is beneficial (e.g., the business, application, data, security reference models). Further, should a normalised framework be selected, the SOA-based taxonomy can always be retrospectively mapped onto the selected framework; this is the essence of the late mapping that is implied by the concurrent development approach that Figure 10 conveys.

2.4 TOGAF Adaptation

TOGAF provides an industry standard for architecture that was recommended by [EA-2011] and adopted for use in Ethiopia and will be retained to guide development of the GEA-ET. The complete TOGAF specification, whose content is extensively referenced in the sections that follow, may be accessed here [@].


Architecture Development Method (ADM)

Architecture Content Framework

Reference Models

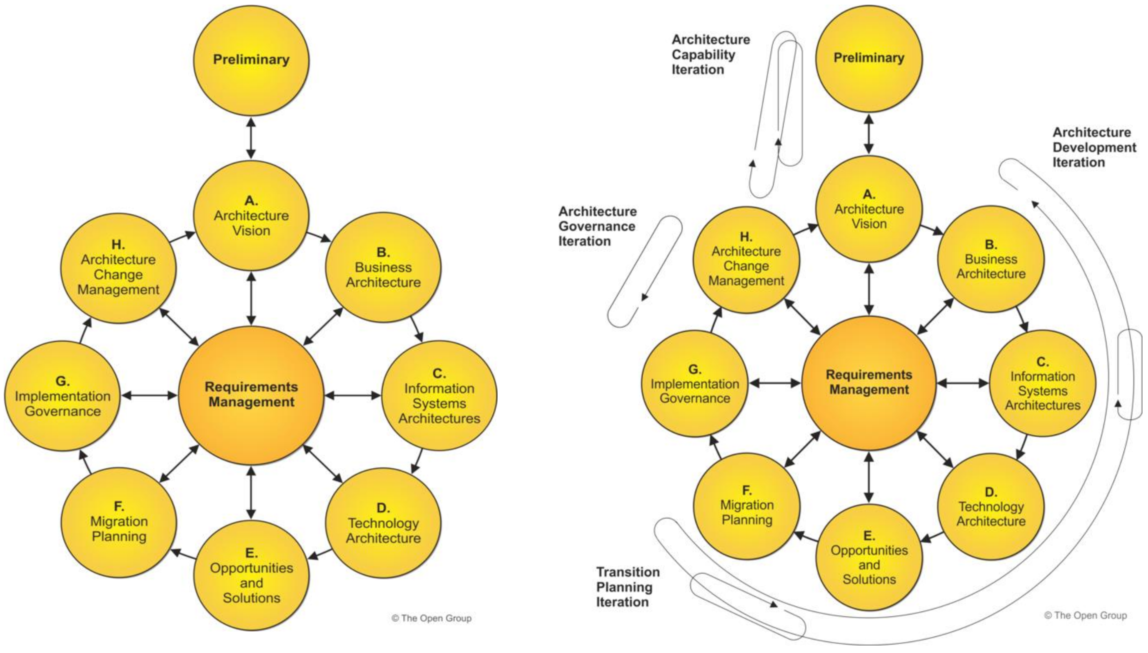
ADM Guidelines & Techniques

Enterprise Continuum

Architecture Capability Framework

- Architecture Development Method: A tested and repeatable process for developing architectures which includes capability establishment, content development, transitioning, and governance.
- Architecture Content Framework: Describes the structure and content of the completed architecture.
- Reference Models: Abstract frameworks for understanding significant relationships among the entities of [an] environment, and for the development of consistent standards or specifications supporting that environment.
- ADM Guidelines & Techniques: Method portfolio that supports the execution of specific tasks within the ADM.
- Enterprise Continuum: Sets the context for an architecture and explains how generic solutions can be specialized to support the requirements of specific organizations.
- Architecture Capability Framework: Describes creation and usage of the processes, skills, and tools needed to operate an ADM.

Figure 4: TOGAF Component Portfolio


Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management
© The Open Group
Architecture Capability Iteration
Architecture Development Iteration
Architecture Governance Iteration
Transition Planning Iteration
Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management
© The Open Group

(a) Standard Workflow

(b) Iterative Workflows

Figure 5: TOGAF Architecture Development Method

The TOGAF standard is collection of frameworks (Figure 4) with the Architecture Development Method (ADM) at its core (Figure 5). The ADM specifies a methodology for developing and managing the lifecycle of an Enterprise Architecture consisting of phases (Preliminary, A-H) that describe the inter-related inputs, activities, and outputs of the lifecycle management process. In addition to the ADM, TOGAF proposes frameworks and resource collections that complete the TOGAF methodology, namely:

- Architecture Content Framework: A structured content model that allows major architectural work products to be consistently defined, structured, and presented. The content framework groups architectural work products into three categories: deliverables that are formally specified and approved, artifacts that describe an aspect of the architecture, generally in the form of catalogues, matrices, and diagrams, and building blocks which represent (possibly re-usable) enterprise capabilities that can be combined to deliver architectures and solutions (Figure 6).
- Reference Models: Abstract frameworks for understanding significant relationships among the entities of [an] environment, and for the development of consistent standards or specifications for that environment.
- Guidelines and Techniques: A set of varied resources that guide practitioners in executing ADM tasks, including guidelines, templates, checklists, procedures, and other detailed materials.
- Enterprise Continuum: A view of the architecture repository that shows linkages between related architectures from generic to specific, from abstract to concrete, and from logical to physical (Figure 7).
- Architecture Capability Framework: Provides guidance on establishing specific architectural capabilities, including organisation structures, processes, roles, responsibilities, and competencies (Figure 8).

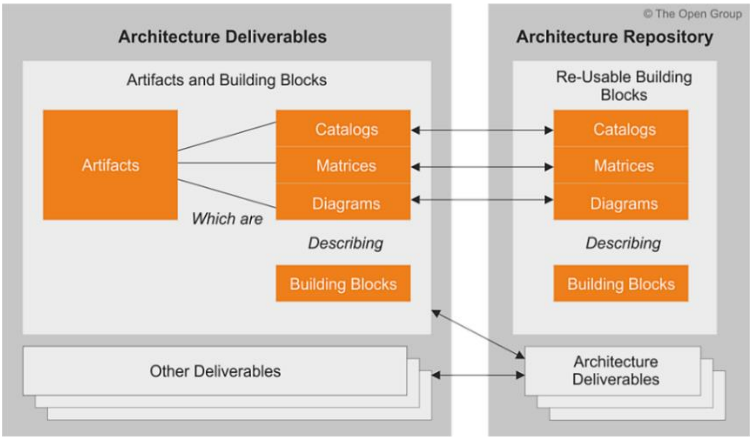
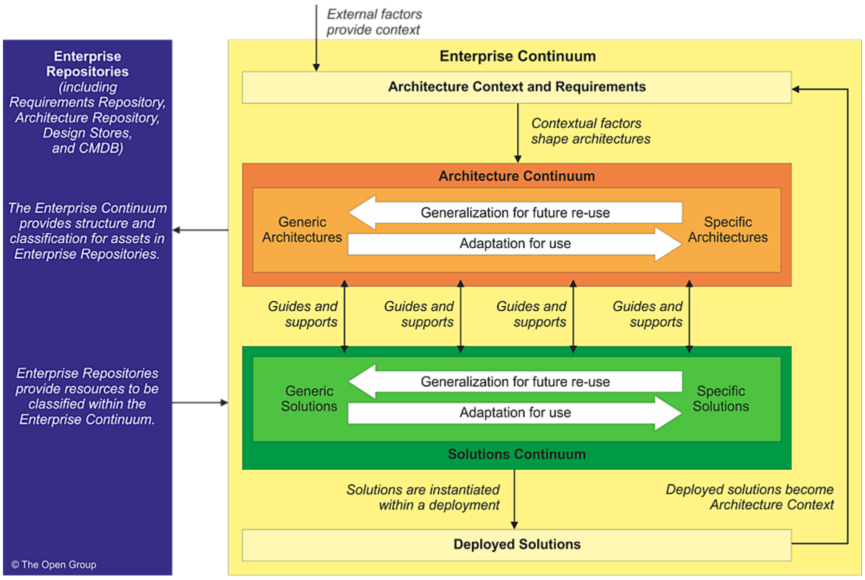


Figure 6: Architecture Content Framework: Deliverables, Artefacts & Building Blocks

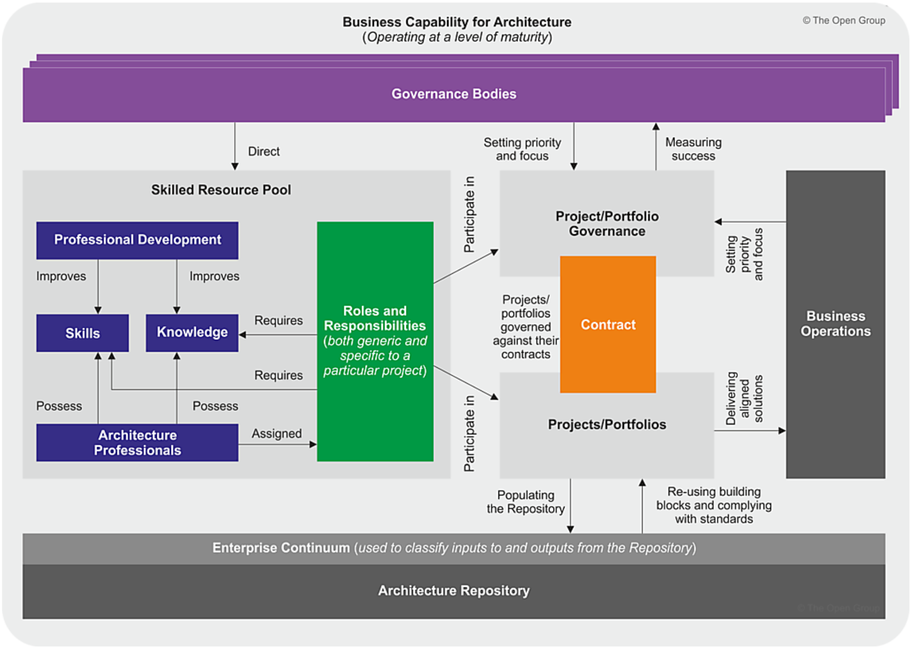
This expansive collection of resources and frameworks makes TOGAF a large and complex methodology which can easily overwhelm the limited GEA-ET development and governance capacity in Ethiopia.

Therefore, it is necessary to adapt the methodology so that it can be used effectively in such an environment, guided primarily by the “adequate”, “minimal”, and “focused”, and “service-orientation” architectural principles enumerated in Table 1 (for guidance on their interpretation, see Table 19), to come up with methodology adjustments that are devised specifically to reduce GEA-ET development and governance complexity.



External factors provide context
Enterprise Repositories (including Requirements Repository, Architecture Repository, Design Stores, and CMDB)
The Enterprise Continuum provides structure and classification for assets in Enterprise Repositories.
Enterprise Repositories provide resources to be classified within the Enterprise Continuum.
© The Open Group
Enterprise Continuum
Architecture Context and Requirements
Contextual factors shape architectures
Architecture Continuum
Generic Architectures
Generalization for future re-use
Adaptation for use
Specific Architectures
Guides and supports
Guides and supports
Guides and supports
Guides and supports
Generic Solutions
Generalization for future re-use
Adaptation for use
Specific Solutions
Solutions Continuum
Solutions are instantiated within a deployment
Deployed solutions become Architecture Context
Deployed Solutions

Figure 7: Enterprise Continuum: Generalised and Specialised Capabilities



Business Capability for Architecture
(Operating at a level of maturity)
© The Open Group
Governance Bodies
Direct
Setting priority and focus
Measuring success
Skilled Resource Pool
Professional Development
Improves
Improves
Skills
Knowledge
Requires
Requires
Roles and Responsibilities (both generic and specific to a particular project)
Possess
Possess
Architecture Professionals
Assigned
Participate in
Participate in
Project/Portfolio Governance
Projects/ portfolios governed against their contracts
Contract
Projects/Portfolios
Setting priority and focus
Delivering aligned solutions
Business Operations
Populating the Repository
Re-using building blocks and complying with standards
Enterprise Continuum (used to classify inputs to and outputs from the Repository)
Architecture Repository

Figure 8: Architecture Capability Framework

Table 1 TOGAF Adaptation Framework Principles

Adequate System Engineering (Framework-2 Principle)	
Statement	Prioritise pragmatism and delivery ahead of idealised notions of architectural completeness.
Rationale	+ Previous GEA-ET development efforts in Ethiopia have emphasised analysis of architectural requirements (what is to be done) with limited implementation guidance (how it is to be done). + This has resulted in complex requirements and over-engineered artefacts which are difficult to transfer to traditional MDA implementation contexts.
Implications	+ The balance of GEA-ET development effort must be shifted from analytical completeness to implementation concerns as this will inject some much-needed momentum and traction into GEA-ET initiatives. + This implies prioritising reference architectures and roadmaps which are defined at the Whole-Of-Government level in the GEA-ET framework, ahead framework level concerns like reference models and the MDA level concerns that are not related to interoperability.
Minimal Viable Architecture (MVA) Paradigm (Framework-3 Principle)	
Statement	Develop and deploy the GEA-ET framework in an incremental, lightweight manner.
Rationale	+ MVA responds to the pervasive challenges associated with traditional ‘waterfall-style’ EA delivery cycles, such as limited business agility and responsiveness, over-engineered artefacts, the long time to value accrual, and the high upfront investment in people, processes and technologies. + MVA prioritises agility and tries to strike a balance between speed and completeness of architecture delivery.
Implications	+ Standards and principles must be applied flexibly (in a lightweight manner) at the GEA-ET implementation level since MVA places emphasis on speed, rather than completeness and rigour, which can always be applied retrospectively. + Early selection of core architectural components and functions that deliver early and tangible business value, along with meaningful and (re)usable technology capabilities - MVA deliverables are not disposable. + MVA means focusing on the core, foundational architectural components of the target solutions (a working baseline) and building on them to construct the rest of the architecture over time. + MVA applies to architectures for all domains (e.g., business, data, integration, etc.. so they must first identify the foundational components and then build improve them in a iterative fashion over time.
Focused Delivery Roadmap (Framework-4 Principle)	
Statement	Mitigate GEA-ET development and deployment complexity.
Rationale	+ Ethiopian GEA-ET landscape has many inherent complexities that arise from its large scope and scale, which must be carefully mitigated to prevent GEA-ET implementors from being overwhelmed in the detail. + A proven approach for managing problem complexity is to partition the problem into individual (possibly independent) 'bite-size' problem statements that are easier to comprehend.
Implications	+ The layered GEA-ET content organisation intrinsically mitigates complexity because it enables GEA-ET development at three independent levels: framework, WOG, and MDA. + Categorise requirements within and across the GEA-ET layers, and prioritise initiatives that WOG programmes; this will reduce the time to meaningful and usable business value.
Service-Orientation (Framework-6 Principle)	
Statement	Implement GEA-ET components as encapsulated services with well-defined, standards-based interfaces.

Table 1 TOGAF Adaptation Framework Principles	
Rationale	+ The service orientation paradigm requires service providers to have a well-defined interface that specifies how consumers can access the services they provide, without the need to know the internal implementation of those services. + This ensures loose coupling between providers and consumers, so that changes in the one do not affect the other.
Implications	+ Focus more on capabilities, standards, and principles that influence interoperability structures and process, with less emphasis on the internal implementation MDA solution components. + Develop a common language and classification framework to describe common shared capabilities, which does not mandate methods that MDAs can use for developing enterprise or solution architectures. + Reduces the architectural compliance validation effort since it reduces the validation depth and breadth. + Strong governance of service specifications and implementations is required.

In part, TOGAF's complexity is rooted in its flexible frameworks, which are designed to address the needs of diverse enterprises and projects. An example is the ADM lifecycle, which allows iteration within and across ADM phases and tiering of activities within phases to provide tailored architecture delivery pathways that support different types of architectural engagement [@]. Another example is the enterprise continuum, which is intended to track the evolution of multiple related architectures and solutions, rather than to provide snapshots of the architectural landscape at specific points-in-time.

TOGAF cites several benefits for the flexibility that permeates the standard, but one drawback with this approach is that the methodology must be 'configured' before it can be used effectively, by assessing and selecting the right set of options from an extensive 'menu' of available framework options. Such configuration requires architectural expertise at all GEA-ET levels - framework, WOG, and MDA – and the likelihood is that this expertise is not readily available in Ethiopia.

2.4.1 Adaptation Actions

To address these flexibility and capacity concerns, the GEA-ET proposes embedding an initial prescriptive configuration of the TOGAF methodology, with pre-selected and simplified framework options (a process that TOGAF refers to as "framework tailoring"). Specific tactics are applied to support this approach, including:

- Minimal Architecture Portfolio: Select only one reference architecture for each SOA perspective and the overall GEA, initially for the express purpose of supporting solution framing, selection and implementation activities; later, as a planned downstream activity, position these reference architectures and the solutions they realise as building blocks within the enterprise continuum.
- Reduced Artefact Resolution: Initially develop reference model and solution architecture specifications at higher levels of abstraction and then gradually increase the detail through incremental refinement of artefact structures and relationships; select an artefact resolution depth that carefully balances the benefits of higher resolution against artefact maintenance capacity to ensure GEA-ET sustainability.

- Deliverable Rationalisation: The standard content metamodel specifies an expansive set of work products (especially artefacts), many of which are not required to support a minimal viable architecture approach; the GEA-ET artefact portfolio selects the minimal set of work products required to support MVA.
- Concurrent Development: In line with the adequate system engineering principle, do not delay solution implementation pending completion of detailed reference models and architectures; rather, make implementation an inherent part of the design process and defer alignment of implementations and architectures (i.e., accept the potential rework and alignment risks and leverage assessments of MVA build outcomes to accrue early feedback on failure and success).
- Linear Delivery Iterations: Whilst the GEA-ET recommends an iterative GEA-ET development approach that incrementally builds out the GEA, the delivery cycle within each increment is linear to support predictability and simplicity; this is especially important in the early TOGAF methodology adoption stages when stability must take precedence over agility.
- Solution Focus: At the MDA level, limit building block content to solution architectures with emphasis on elaboration of interoperability requirements; focus the efforts of the WOG team on provision of interoperability guidelines for the MDA solution architectures. The implication is that MDAs will only be expected to share solution architecture content; the provision of MDA-specific reference architectures will be optional, developed as independent initiatives (preferably in collaboration with industry or private sector associations).
- Shift-Left: Based on successful Agile practices encourage early assessment of outcomes, in this context the ‘shift-left’ paradigm is proposed as mechanism for moving compliance focus to design and/or pre-procurement phases, in preference to conducting post implementation audits which can only uncover and respond to anomalies ‘after the fact’.

To further clarify this adaptation guidance, the following diagrams (Figure 10 - Figure 13) restate some of these modifications in a format that attempts to illustrate their effects visually, along with elaborated descriptions of the adaptations themselves.

2.4.2 Adaptation Outcomes

These adaptations apply fundamental changes to the way that the TOGAF methodology is applied in the GEA-ET with effects that are distributed across the framework portfolio and elaborated in the remainder of this report. Examples are minimal architecture portfolio adaptation, which results in singular proposals of reference and solution architectures, and reduced artefact resolution, which influences the proposed reference model structure and content. Not discussed are the adaptations applied to the TOGAF governance and management frameworks which are covered in Section 7.

Deliverable Rationalisation: The standard content metamodel specifies an expansive set of work products (especially artefacts), many of which are not required to support a minimal viable architecture approach; the GEA-ET artefact portfolio selects the minimal set of work products required to support the MVA approach (see Table 2).

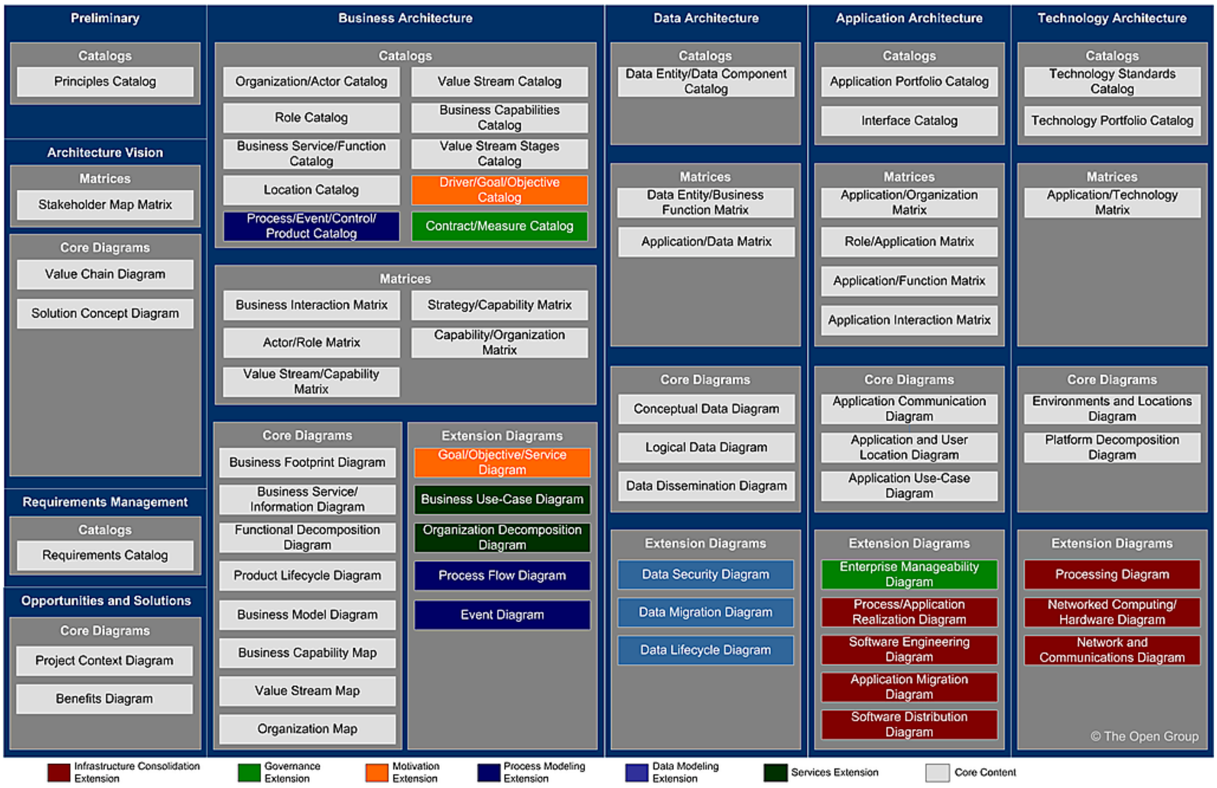

Preliminary
Catalogs
Principles Catalog
Architecture Vision
Matrices
Stakeholder Map Matrix
Core Diagrams
Value Chain Diagram
Solution Concept Diagram
Requirements Management
Catalogs
Requirements Catalog
Opportunities and Solutions
Core Diagrams
Project Context Diagram
Benefits Diagram
Business Architecture
Catalogs
Organization/Actor Catalog
Value Stream Catalog
Role Catalog
Business Capabilities Catalog
Business Service/Function Catalog
Value Stream Stages Catalog
Location Catalog
Driver/Goal/Objective Catalog
Process/Event/Control/ Product Catalog
Contract/Measure Catalog
Matrices
Business Interaction Matrix
Strategy/Capability Matrix
Actor/Role Matrix
Capability/Organization Matrix
Value Stream/Capability Matrix
Core Diagrams
Business Footprint Diagram
Business Service/ Information Diagram
Functional Decomposition Diagram
Product Lifecycle Diagram
Business Model Diagram
Business Capability Map
Value Stream Map
Organization Map
Extension Diagrams
Goal/Objective/Service Diagram
Business Use-Case Diagram
Organization Decomposition Diagram
Process Flow Diagram
Event Diagram
Data Architecture
Catalogs
Data Entity/Data Component Catalog
Matrices
Data Entity/Business Function Matrix
Application/Data Matrix
Core Diagrams
Conceptual Data Diagram
Logical Data Diagram
Data Dissemination Diagram
Extension Diagrams
Data Security Diagram
Data Migration Diagram
Data Lifecycle Diagram
Application Architecture
Catalogs
Application Portfolio Catalog
Interface Catalog
Matrices
Application/Organization Matrix
Role/Application Matrix
Application/Function Matrix
Application Interaction Matrix
Core Diagrams
Application Communication Diagram
Application and User Location Diagram
Application Use-Case Diagram
Extension Diagrams
Enterprise Manageability Diagram
Process/Application Realization Diagram
Software Engineering Diagram
Application Migration Diagram
Software Distribution Diagram
Technology Architecture
Catalogs
Technology Standards Catalog
Technology Portfolio Catalog
Matrices
Application/Technology Matrix
Core Diagrams
Environments and Locations Diagram
Platform Decomposition Diagram
Extension Diagrams
Processing Diagram
Networked Computing/ Hardware Diagram
Network and Communications Diagram
© The Open Group
Infrastructure Consolidation Extension
Governance Extension
Motivation Extension
Process Modeling Extension
Data Modeling Extension
Services Extension
Core Content

Figure 9: TOGAF Adaptation: Artefact Rationalisation

Table 2: Rationalised Deliverable Portfolio

Attribute	Description
Inception Plan	Architecture Vision | Architecture Principles | Stakeholder Matrix | Solution Phase/Stream Matrix
Business Architecture	Driver/Objective Catalogue | Stakeholder Matrix | User Journey Map | Process Flow Diagram | Actor/Role Matrix | Functional Decomposition Diagram | Business Service/ Capability Matrix
Data Architecture	Data Entity/Component Catalogue | Application/Data Matrix | Conceptual Data Diagram | Logical Data Diagram
Application Architecture	Application Portfolio Catalogue | Role/ Application Matrix | Application/Function Matrix
Technology Architecture	Application/Technology Matrix | Standards Catalogue | Deployment Diagram
Security Architecture	Capability Catalogue
Integration Architecture	Interface Catalogue | Transaction Process Diagram | API/Service Catalogue

Concurrent Development: In line with the adequate system engineering principle, do not delay solution implementation pending completion of detailed reference models and architectures; rather, make implementation an inherent part of the design process and defer alignment of implementations and architectures (i.e., accept the potential rework and alignment risks and leverage assessments of MVA build outcomes to accrue early feedback on failure and success). This concurrent approach means [1] implement WOG solution architectures, [2] design reference models, and [3] catalogue MDA solutions; then [4] retrospectively classify implemented WOG and MDA solutions using reference model structures and taxonomies. Note that many transversal deliverables have a dependence relationship on tier deliverables, whose development must therefore take precedence.

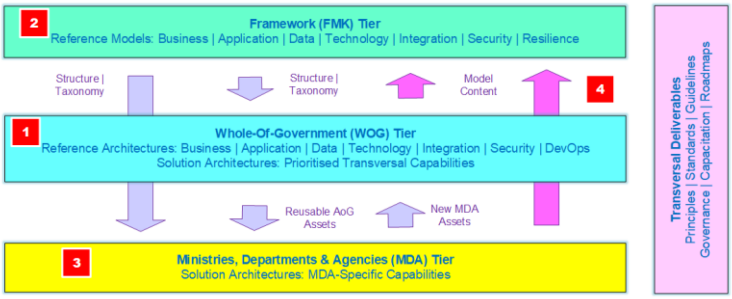

2
Framework (FMK) Tier
Reference Models: Business | Application | Data | Technology | Integration | Security | Resilience
Structure | Taxonomy
Structure | Taxonomy
Model Content
4
1
Whole-Of-Government (WOG) Tier
Reference Architectures: Business | Application | Data | Technology | Integration | Security | DevOps
Solution Architectures: Prioritised Transversal Capabilities
Reusable AoG Assets
New MDA Assets
3
Ministries, Departments & Agencies (MDA) Tier
Solution Architectures: MDA-Specific Capabilities
Transversal Deliverables
Principles | Standards | Guidelines
Governance | Capacitation | Roadmaps

Figure 10: TOGAF Adaptation: Concurrent Development

Minimal Architecture Portfolio: Select only one reference architecture for each SOA perspective and the overall GEA, initially for the express purpose of supporting solution framing, selection and implementation activities; later, as a planned downstream activity, position these reference architectures and the solutions they realise as building blocks within the enterprise continuum.

This means develop one reference architecture and the minimal number of solution architectures per SOA perspective, as 'seeds' for the enterprise continuum, and select only one reference architecture for each SOA perspective and the overall GEA-ET to set the continuum foundation. Note that transition architectures are not under consideration for WOG initiatives since they are mostly 'greenfield' architectures and deployments but may become relevant existing MDA deployments are onboarded.

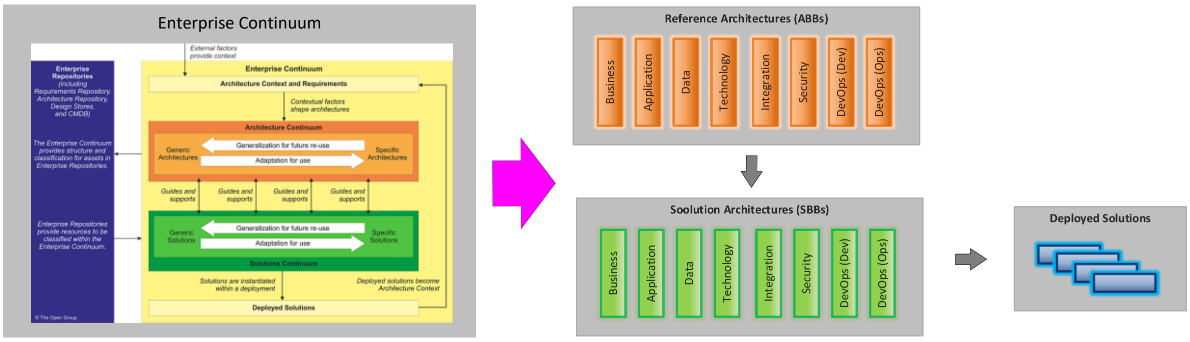

Enterprise Continuum
External factors provide context
Enterprise Repositories (including Requirements Repository, Architecture Repository, Design Stores, and CMDB)
The Enterprise Continuum provides structure and classification for assets in Enterprise Repositories.
Enterprise Repositories provide resources to be classified within the Enterprise Continuum.
© The Open Group
Enterprise Continuum
Architecture Context and Requirements
Contextual factors shape architectures
Architecture Continuum
Generic Architectures
Generalization for future re-use
Adaptation for use
Specific Architectures
Guides and supports
Guides and supports
Guides and supports
Guides and supports
Generic Solutions
Generalization for future re-use
Adaptation for use
Specific Solutions
Solutions Continuum
Solutions are instantiated within a deployment
Deployed solutions become Architecture Context
Deployed Solutions
Reference Architectures (ABBs)
Business
Application
Data
Technology
Integration
Security
DevOps (Dev)
DevOps (Ops)
Soolution Architectures (SBBs)
Business
Application
Data
Technology
Integration
Security
DevOps (Dev)
DevOps (Ops)
Deployed Solutions

Figure 11: TOGAF Adaptation: Minimal Architecture Portfolio

Reduced Artefact Resolution: Initially develop reference model and solution architecture specifications at higher levels of abstraction and then gradually increase the detail through incremental refinement of artefact structures and relationships; select an artefact resolution depth that carefully balances the benefits of higher resolution artefacts against artefact maintenance capacity to ensure GEA-ET sustainability.

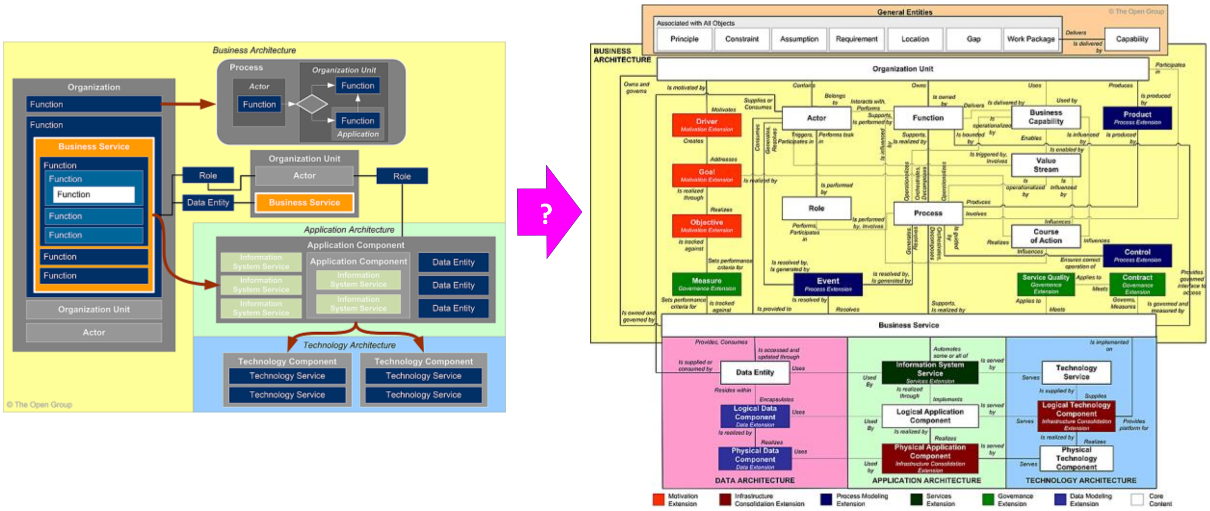


Figure 12: TOGAF Adaptation: Reduced Artefact Resolution

Linear Delivery Iterations: Whilst the GEA-ET recommends an iterative GEA-ET development approach that incrementally builds out the GEA, the delivery cycle within each increment is linear to support predictability and simplicity; this is especially important in the early TOGAF methodology adoption stages when stability must take precedence over agility. This approach may be thought of as embedding standard waterfall development sequences within an agile solution delivery lifecycle, with incrementally developed solution architectures guiding delivery for each solution iteration (see also Section 0).

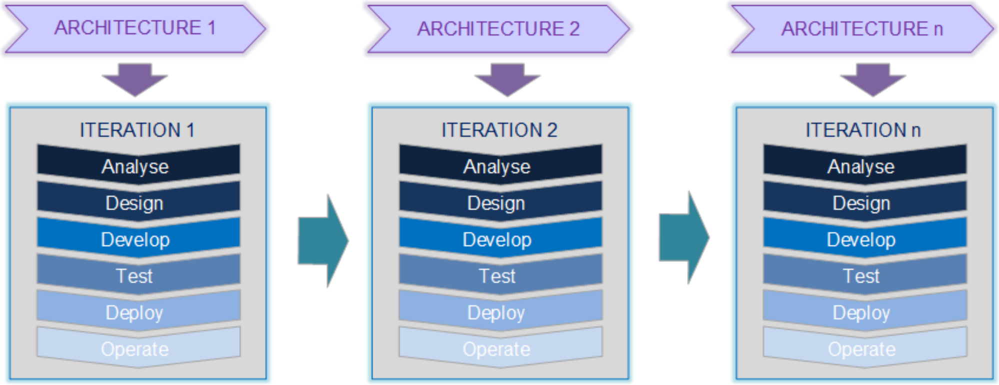
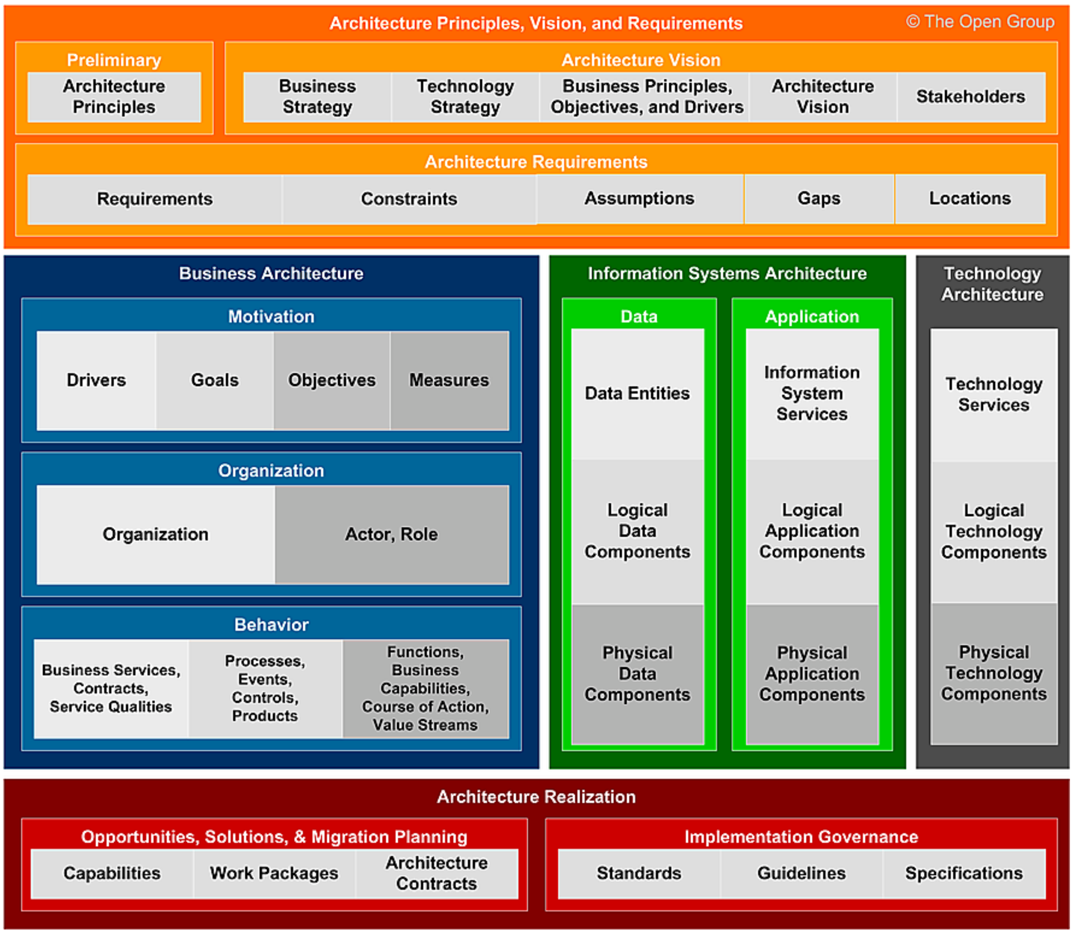


Figure 13: TOGAF Adaptation: Linear Delivery Iterations

3 Framework Tier

The framework tier consists primarily of reference models which provide an abstract framework for identifying and understanding significant relationships among the entities of specific environments, and for the development of consistent standards or specifications that support those environments. The TOGAF content metamodel [@] whose detailed view is shown in Figure 14, is proposed as the initial foundational content framework for all the GEA-ET reference models for two main reasons:



Architecture Principles, Vision, and Requirements
© The Open Group
Preliminary
Architecture Principles
Architecture Vision
Business Strategy
Technology Strategy
Business Principles, Objectives, and Drivers
Architecture Vision
Stakeholders
Architecture Requirements
Requirements
Constraints
Assumptions
Gaps
Locations
Business Architecture
Motivation
Drivers
Goals
Objectives
Measures
Organization
Organization
Actor, Role
Behavior
Business Services, Contracts, Service Qualities
Processes, Events, Controls, Products
Functions, Business Capabilities, Course of Action, Value Streams
Information Systems Architecture
Data
Data Entities
Logical Data Components
Physical Data Components
Application
Information System Services
Logical Application Components
Physical Application Components
Technology Architecture
Technology Services
Logical Technology Components
Physical Technology Components
Architecture Realization
Opportunities, Solutions, & Migration Planning
Capabilities
Work Packages
Architecture Contracts
Implementation Governance
Standards
Guidelines
Specifications

Figure 14: TOGAF Core Content Metamodel

- It provides an understandable normative metamodel that can be configured at different resolution levels without losing integrity, which aligns with all the other TOGAF components. As an example, the data elements of the information systems architecture can be specified at the system, entity, logical or physical component levels, allowing GEA-ET developers the latitude to start specifications at higher abstraction levels and then incrementally add the anticipated detail over time – such capability underpins GEA-ET's 'adequacy' and 'minimal' architectural principles and related tactics.

- Although the metamodel core only incorporates TOGAF's base business, information and technology domains, TOGAF provides guidance on how the metamodel can be extended to accommodate other architectural domains, without a loss of metamodel integrity. Thus, metamodel coverage can be broadened using "extensions", such as the event, control, and product process extensions shown in Figure 15 (termed "standard" extensions), and then even further extended to include content related to the GEA-ET SOA layers like integration and security (these would be termed "custom" extensions).

Thus, the TOGAF content metamodel provides a solid baseline on which the initial development iterations of the GEA-ET reference models can be initiated.

3.1 Metamodel Scope

The scope of the content metamodel scope is determined primarily by the number of entities and attributes defined for each metamodel entity. The GEA-ET recommendation is that the initial reference model scope be limited to the entities listed in Table 3 (also highlighted in Figure 15), where the "origin" column indicates the entity source as TOGAF core content, TOGAF standard extension, or GEA-ET custom extension.

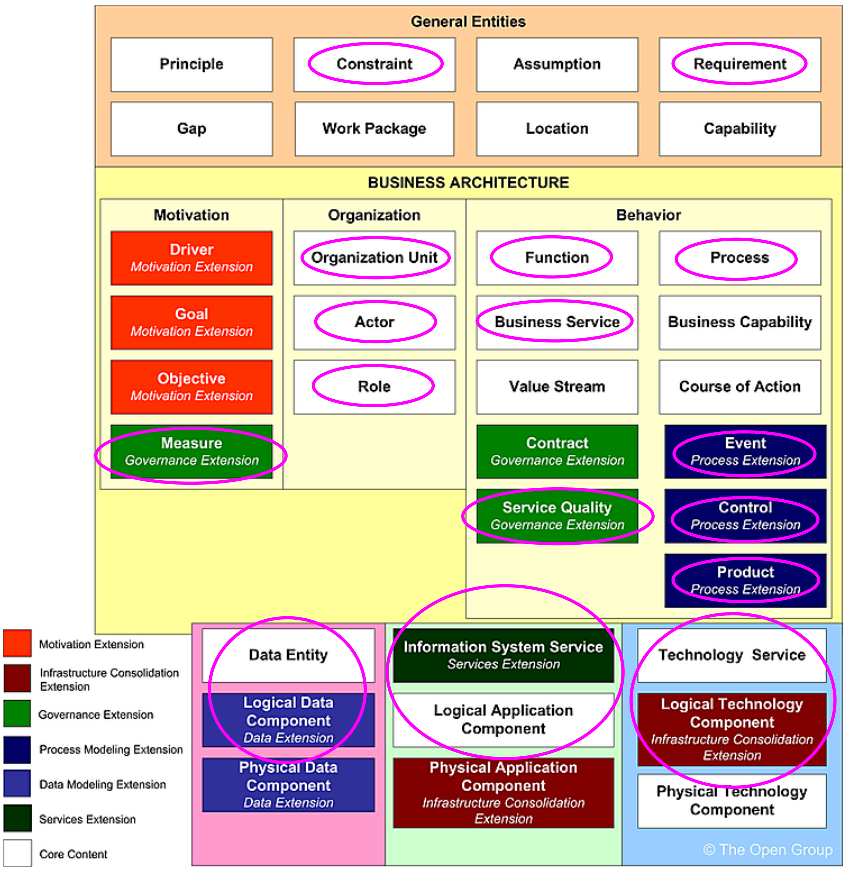


Figure 15: TOGAF Extended Content Metamodel

Similarly, Table 4 content is limited to the common entity attributes that TOGAF recommends, because the full attribute set is very large and can always be accessed here [@]. Attributes for the custom GEA-ET extension entities will be determined as part of the GEA-ET implementation process.

Note that this restricted scope should be treated as an initial suggestion which is driven by the TOGAF adaptation actions described in Section 2.4.1 (namely “reduced artefact resolution” and “deliverable rationalisation”), but this is expected to change as the architecture team gains a better understanding of the parameters needed to support their architecture work. In this regard, it should be borne in mind that a higher number of entities and attributes suggests better support for content discovery, sharing, (re)use and analysis, but it also implies greater metamodel development and maintenance effort.

Table 3: Minimal Content Metamodel Entities

Status	Entity	Description
Core	Business Service	Supports business capabilities through an explicitly defined interface and is explicitly governed by an organization.
Core	Function	Delivers business capabilities closely aligned to an organization, but not necessarily explicitly governed by the organization.
Core	Measure	An indicator or factor that can be tracked, usually on an ongoing basis, to determine ‘success’ or alignment with objectives and goals.
Core	Organization Unit	A self-contained unit of resources with goals, objectives, and measures, including external parties and business partners.
Core	Process	A process represents flow of control between or within functions and/or services (depends on the granularity of definition).
Core	Role	The usual or expected function of an actor, or the part played in a particular action or event.
Standard	Logical Component	An encapsulation of application, data, or technology capabilities or resources that is independent of a particular implementation.
Standard	Product	Output generated when the business executes a process.
Standard	Service	An element of behaviour that provides specific functionality in response to requests from actors or other services.
Custom	Interface	Collection of behaviours that provide specific functionality in response to requests from actors or other services; each behaviour is explicitly governed by a Contract.
Custom	Transaction Process	Analogous to a business process but is defined at the interface level to elaborate interface interaction structures and dynamics.
Custom	Security Service	Describes a control used to protect the confidentiality, integrity, and availability of business services.
Custom	Development Capability	A development-focused function that that fulfils, or supports the fulfilment of, solution development lifecycle requirements.
Custom	Operations Capability	A service management function that that fulfils, or supports the fulfilment of, solution operations lifecycle requirements.

Table 4: Common Content Metamodel Attributes

Entity	Attribute	Description
All	ID	Unique identifier for the architecture entity.
	Name	Brief name of the architecture entity.
	Description	Textual description of the architecture entity.

	Category	User-definable categorization taxonomy for each metamodel entity.
	Source	Location from where the information was collected.
	Owner	Owner of the architecture entity.

3.2 Metamodel Sources

TOGAF's content metamodel points to the type of content that should be captured by a reference model and can provide a solid baseline for this purpose – hence our use of the term "foundational content framework" – and also provide structures for that content (as specified in Table 4). The metamodel also defines entity relationships that are key traceability constructs and can aid in understanding simple dependencies between entities (see, for example, Figure 16 which shows the data extension relationships).

However, the main reference model development effort lies in defining (and aligning) content classification taxonomies and sourcing or developing the content itself, in a form that aids content discovery, sharing, (re)use and analysis, which is one of the core benefits cited for architectural reference models. Instead of starting such effort from ground up, it makes sense to exploit the high quality, reference models that have been open sourced by various countries and organisations.

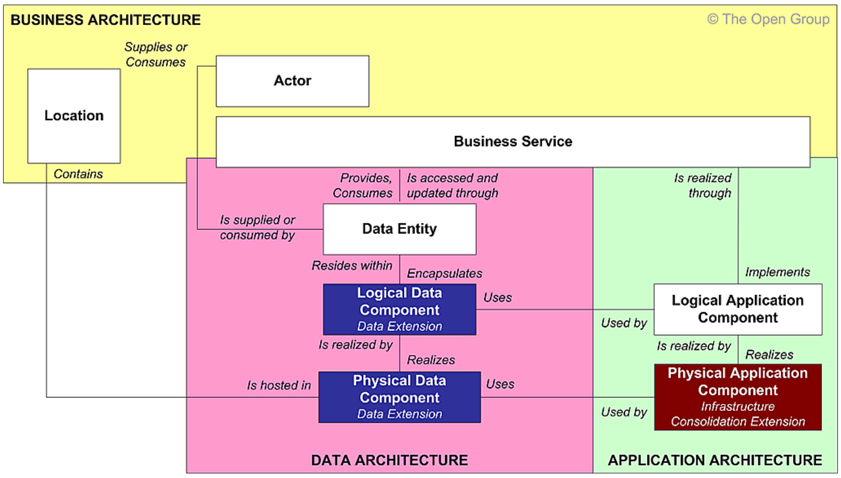


Figure 16: TOGAF Content Metamodel: Data Extension Relationships

For this purpose, Table 5 lists the sources that the GEA-ET diagnostic identified as potential structure and content sources for the GEA-ET reference models, including various generic-, domain-, and industry-specific reference models and content repositories. The illustrations below (Figure 18 - Figure 22) combine snapshots and commentary on the structure and/or content of some of these sources. Other important sources include solution owners within the MDAs, solution developer and user documentation, and any existing reference models and architectures which are especially useful when new capabilities are to be implemented.

It is anticipated that the GEA-ET development team will review and leverage these resources (and possibly discover additional relevant sources) to initialise (structure) the GEA-ET reference models, and then incrementally tailor and populate them with Ethiopia-specific content.

Table 5: Candidate Reference Model Sources

Publisher	Source	Description	Applicability
EA-2011 [@] | EA-2023 [@]	EA-2011 [@] | EA-2023 [@]	Developed by EA-2011 and revised by EA-2023, provides useful Ethiopia-specific catalogues of business, data and application services, that can guide reference model content development.	Business, Application, Data,
GovStack Community [@]	GovStack Ecosystem RA [@]	Collection of solution-focused architectural building blocks for government services, whose content can be analysed primarily to inform the development of reference and solution architectures, but only limited applicability to reference model and architecture development. However, it has been included in this compendium because a key GEA-ET objective is to leverage the GovStack community to bootstrap the development of GEA-ET reference models, as outlined in Section 8.5 which discusses ecosystem sourcing to drive Ethiopia's GEA-ET development goals.	Business, Application, Data, Integration
Business Architecture Guild [@]	Government Reference Model (GRM) [@]	A reference model component portfolio that enables construction of optimally-factored industry- and enterprise-specific reference models. This high-quality business model is referenced by several country GEA-ET initiatives, and although it sits behind a paywall (understood to be a one-time investment of around $400 for internal enterprise use), it is recommended that this asset be sourced to provide solid foundational building blocks for the GEA-ET business architecture.	Business
The Open Group [@]	TOGAF Government RM (GRM) [@]	Specifies a standard reference model template that enables description of public sector services and allows for different architecture approaches and analysis techniques. It has a simple 3-level classification structure which aligns well with the GEA-ET's MVA principle.	Business
The Open Group [@]	TOGAF Technology RM (TRM) [@]	Consists of a taxonomy, which defines terminology, and provides a coherent description of the components and conceptual structure of an information system, and graphical representation of the taxonomy, as an aid to understanding. The TRM objective is to enable structured definition of the standardized Application Platform and its associated interfaces, with the aim of ensuring that the higher-level building blocks which make up business solutions have a complete, robust runtime platform.	Technology
The Open Group [@]	TOGAF Integrated Information Infrastructure RM (III-RM) [@]	Like the TRM, consists of a taxonomy and visual representations, which expands the business and infrastructure application parts of the TRM to support the design of integrated information infrastructures that enable Boundaryless Information Flow.	Application, Data, Integration
Government of Ethiopia	Ethiopian National Data Set Master Plan [@]	Identifies common FDRE government data sets whose content can guide the scoping of business and data reference models. It should be viable to combine this content with process architecture development outputs, GovStack data models, and other sources, to define a canonical metamodel for the whole of government. This approach is preferable to ground up development because it will leverage government-focused datasets which already capture much of the Ethiopian data context.	Business, Data

Government of New Zealand	GEA-NZ Reference Taxonomies [@]	Comprises taxonomies provides as a common language to categorise GEA-ET components in the GEA-NZ framework. Core objectives include categorising the Government digital standards catalogue, and promoting service, information, system and technology interoperability. Others are reducing complexity by abstracting, organising and simplifying complex information sets, improving the overall consistency and cohesiveness of integrated services, shared services and common capabilities, and identification of opportunities for development or reuse of common solutions.	Business, Application, Data, Technology
NIST [@]	Cybersecurity Framework [@]	The NIST model is really an architectural framework (rather than traditional reference model) that supports the design of a comprehensive cybersecurity posture that incorporates information security management (ISM) and enterprise risk management (ERM) approaches. Although FEAF provides a succinct reference model which is tailored for structuring and classifying security architectures, the NIST model has been selected here because it provides better support for developing reference architectures at the WOG level which have GEA-ET prioritisation over framework tier components.	Security

Business Services Catalogue: Developed by EA-2011 and revised by EA-2023, provides useful Ethiopia-specific catalogues of business, data and application services, that can guide reference model content development. This description of the business services layer provides relevant content for the business reference model; similar descriptions are proposed for the data and application services layers.

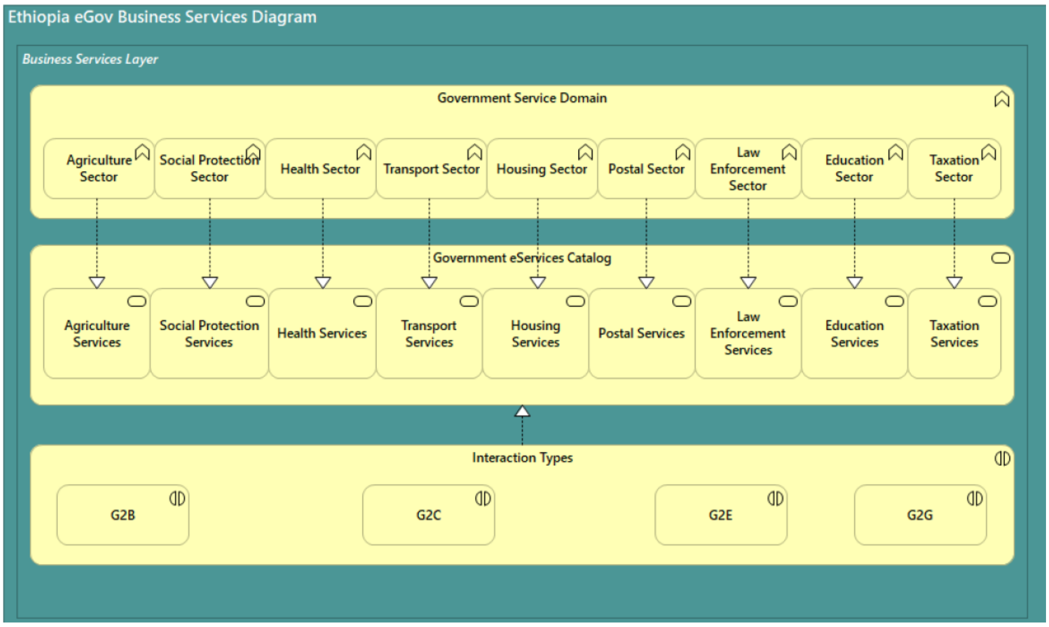
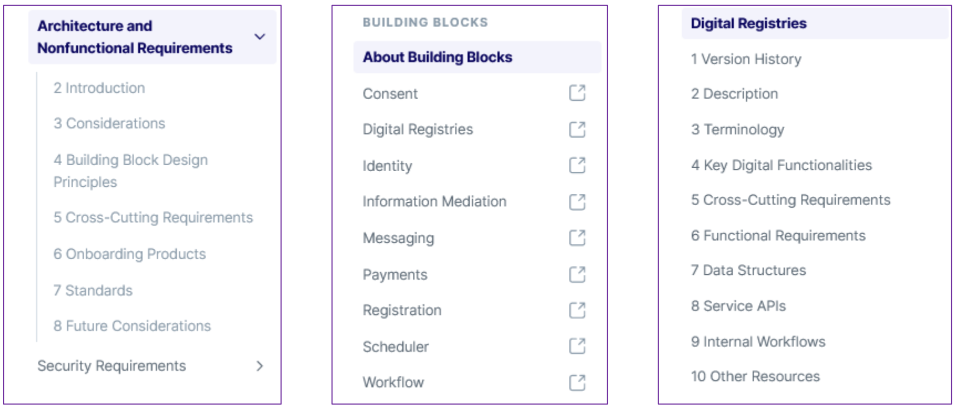


Figure 17: EA-2011 Business Services Layer Content

GovStack Ecosystem Reference Architecture: represents a collection of government-focused architectural building blocks, whose content can be analysed primarily to inform the development of reference and solution architectures but is limited in its applicability to reference model development. However, it has been included in this asset compendium because a key GEA-ET objective is to leverage the GovStack community to bootstrap the development of GEA-ET reference models, as outlined in Section 8.5 which discusses ecosystem sourcing as a deliberate strategy to drive Ethiopia's GEA-ET agenda forward.



Architecture and Nonfunctional Requirements

2 Introduction
3 Considerations
4 Building Block Design Principles
5 Cross-Cutting Requirements
6 Onboarding Products
7 Standards
8 Future Considerations

Security Requirements

BUILDING BLOCKS

About Building Blocks

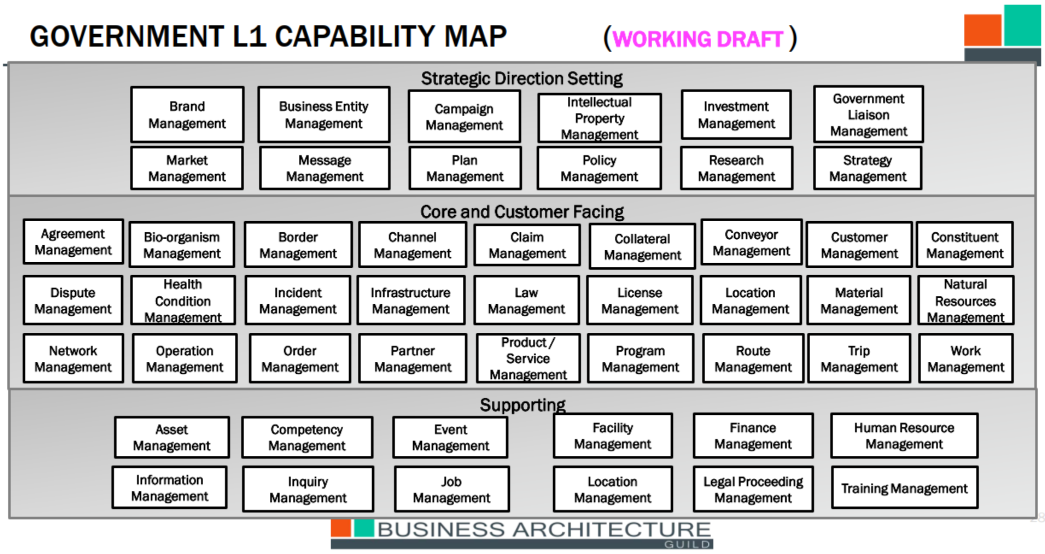
Consent
Digital Registries
Identity
Information Mediation
Messaging
Payments
Registration
Scheduler
Workflow

Digital Registries

1 Version History
2 Description
3 Terminology
4 Key Digital Functionalities
5 Cross-Cutting Requirements
6 Functional Requirements
7 Data Structures
8 Service APIs
9 Internal Workflows
10 Other Resources

Figure 18: GovStack Ecosystem Reference Architecture Outline

BA Government Reference Model: A reference model component portfolio that enables construction of optimally factored industry- and enterprise- specific reference models. This high-quality business model is referenced by several country GEA-ET initiatives, and although it sits behind a paywall (understood to be around $400 for internal enterprise use), it is recommended that this investment be made to give the GEA-ET business architecture a better-than-even chance of initiation on a solid foundation.



GOVERNMENT L1 CAPABILITY MAP (WORKING DRAFT)

Strategic Direction Setting

Brand Management | Business Entity Management | Campaign Management | Intellectual Property Management | Investment Management | Government Liaison Management
Market Management | Message Management | Plan Management | Policy Management | Research Management | Strategy Management

Core and Customer Facing

Agreement Management | Bio-organism Management | Border Management | Channel Management | Claim Management | Collateral Management | Conveyor Management | Customer Management | Constituent Management
Dispute Management | Health Condition Management | Incident Management | Infrastructure Management | Law Management | License Management | Location Management | Material Management | Natural Resources Management
Network Management | Operation Management | Order Management | Partner Management | Product / Service Management | Program Management | Route Management | Trip Management | Work Management

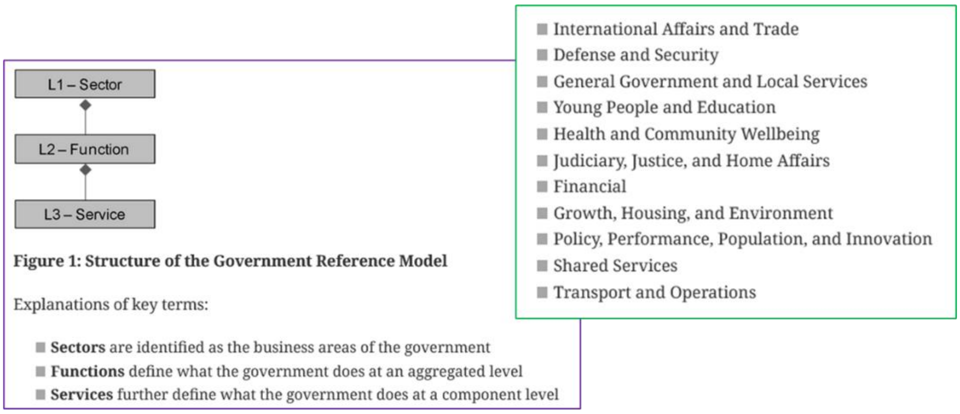
Supporting

Asset Management | Competency Management | Event Management | Facility Management | Finance Management | Human Resource Management
Information Management | Inquiry Management | Job Management | Location Management | Legal Proceeding Management | Training Management

BUSINESS ARCHITECTURE GUILD

Figure 19: BA Guild Government Reference Model Snapshot

TOGAF Government Reference Model: Specifies a standard reference model template that enables description of public sector services and allows for different architecture approaches and analysis techniques. It has a simple 3-level classification structure which aligns well with the GEA-ET's MVA approach.


L1 – Sector
L2 – Function
L3 – Service

- International Affairs and Trade
- Defense and Security
- General Government and Local Services
- Young People and Education
- Health and Community Wellbeing
- Judiciary, Justice, and Home Affairs
- Financial
- Growth, Housing, and Environment
- Policy, Performance, Population, and Innovation
- Shared Services
- Transport and Operations

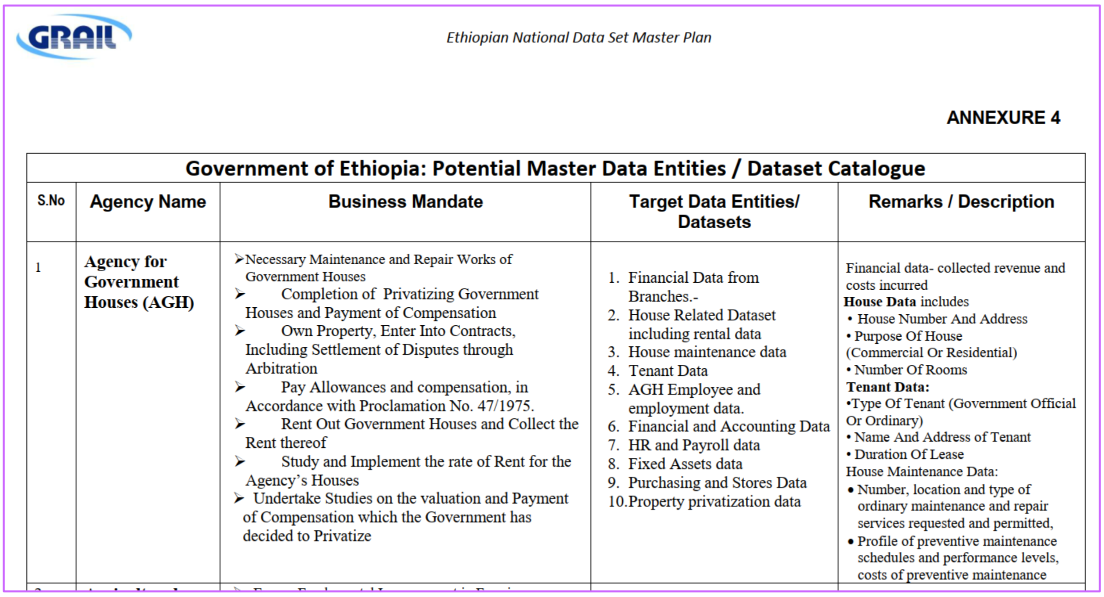
Figure 1: Structure of the Government Reference Model

Explanations of key terms:

- **Sectors** are identified as the business areas of the government
- **Functions** define what the government does at an aggregated level
- **Services** further define what the government does at a component level

Figure 20: TOGAF Government Reference Model Snapshot

Ethiopian National Data Set Master Plan: Identifies common FDRE government data sets whose content can guide the scoping of business and data reference models. It should be viable to combine this content with process architecture development outputs, GovStack data models, and other sources, to define a canonical metamodel for the whole of government. This approach is preferable to ground up development because it will leverage government-focused datasets which already capture much of the Ethiopian data context.



GRAIL *Ethiopian National Data Set Master Plan*

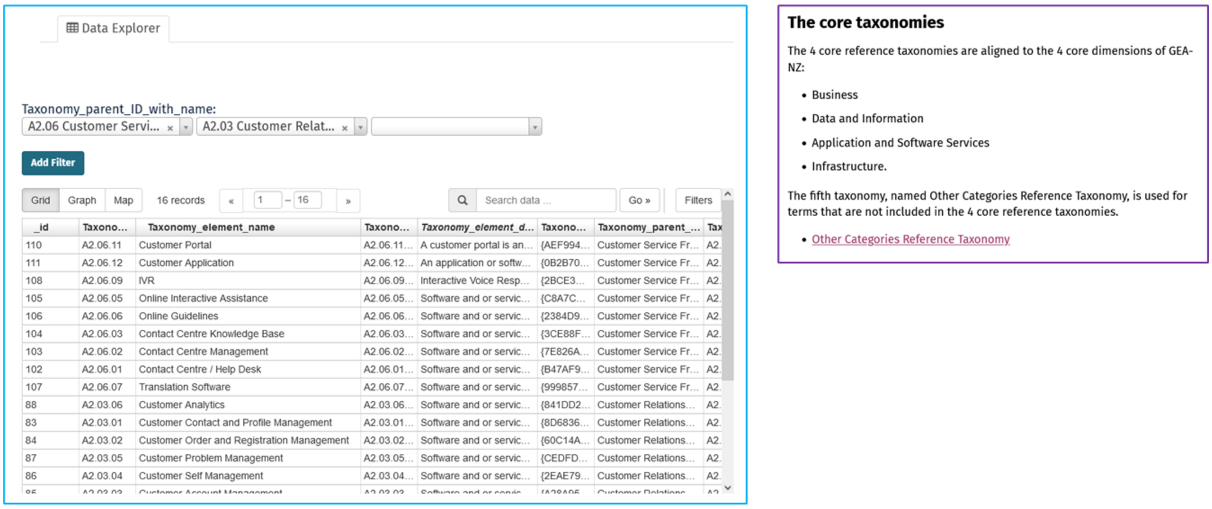
ANNEXURE 4

Government of Ethiopia: Potential Master Data Entities / Dataset Catalogue

S.No	Agency Name	Business Mandate	Target Data Entities/ Datasets	Remarks / Description
1	**Agency for Government Houses (AGH)**	➢Necessary Maintenance and Repair Works of Government Houses ➢ Completion of Privatizing Government Houses and Payment of Compensation ➢ Own Property, Enter Into Contracts, Including Settlement of Disputes through Arbitration ➢ Pay Allowances and compensation, in Accordance with Proclamation No. 47/1975. ➢ Rent Out Government Houses and Collect the Rent thereof ➢ Study and Implement the rate of Rent for the Agency's Houses ➢ Undertake Studies on the valuation and Payment of Compensation which the Government has decided to Privatize	1. Financial Data from Branches.- 2. House Related Dataset including rental data 3. House maintenance data 4. Tenant Data 5. AGH Employee and employment data. 6. Financial and Accounting Data 7. HR and Payroll data 8. Fixed Assets data 9. Purchasing and Stores Data 10. Property privatization data	Financial data- collected revenue and costs incurred **House Data** includes • House Number And Address • Purpose Of House (Commercial Or Residential) • Number Of Rooms **Tenant Data:** •Type Of Tenant (Government Official Or Ordinary) • Name And Address of Tenant • Duration Of Lease House Maintenance Data: • Number, location and type of ordinary maintenance and repair services requested and permitted, • Profile of preventive maintenance schedules and performance levels, costs of preventive maintenance

Figure 21: ENDS Master Plan Snapshot

GEA-NZ Reference Taxonomies: Comprises taxonomies that provide a common language to categorise GEA-ET components in the GEA-NZ framework. Core objectives include categorising the government digital standards catalogue, promoting service, information, system and technology interoperability, reducing complexity by abstracting, organising and simplifying complex information sets, improving the overall consistency and cohesiveness of integrated services, shared services and common capabilities, and identification of opportunities for development or reuse of common solutions.



Taxonomy_parent_ID_with_name:
A2.06 Customer Servi... ×
A2.03 Customer Relat... ×

16 records 1 – 16

_id	Taxono...	Taxonomy_element_name	Taxono...	*Taxonomy_element_d...*	Taxono...	Taxonomy_parent_...	Tax
110	A2.06.11	Customer Portal	A2.06.11...	A customer portal is an...	{AEF994...	Customer Service Fr...	A2.
111	A2.06.12	Customer Application	A2.06.12...	An application or softw...	{0B2B70...	Customer Service Fr...	A2.
108	A2.06.09	IVR	A2.06.09...	Interactive Voice Resp...	{2BCE3...	Customer Service Fr...	A2.
105	A2.06.05	Online Interactive Assistance	A2.06.05...	Software and or servic...	{C8A7C...	Customer Service Fr...	A2.
106	A2.06.06	Online Guidelines	A2.06.06...	Software and or servic...	{2384D9...	Customer Service Fr...	A2.
104	A2.06.03	Contact Centre Knowledge Base	A2.06.03...	Software and or servic...	{3CE88F...	Customer Service Fr...	A2.
103	A2.06.02	Contact Centre Management	A2.06.02...	Software and or servic...	{7E826A...	Customer Service Fr...	A2.
102	A2.06.01	Contact Centre / Help Desk	A2.06.01...	Software and or servic...	{B47AF9...	Customer Service Fr...	A2.
107	A2.06.07	Translation Software	A2.06.07...	Software and or servic...	{999857...	Customer Service Fr...	A2.
88	A2.03.06	Customer Analytics	A2.03.06...	Software and or servic...	{841DD2...	Customer Relations...	A2.
83	A2.03.01	Customer Contact and Profile Management	A2.03.01...	Software and or servic...	{8D6836...	Customer Relations...	A2.
84	A2.03.02	Customer Order and Registration Management	A2.03.02...	Software and or servic...	{60C14A...	Customer Relations...	A2.
87	A2.03.05	Customer Problem Management	A2.03.05...	Software and or servic...	{CEDFD...	Customer Relations...	A2.
86	A2.03.04	Customer Self Management	A2.03.04...	Software and or servic...	{2EAE79...	Customer Relations...	A2.
85	A2.03.03	Customer Account Management	A2.03.03...	Software and or servic...	{A28A95...	Customer Relations...	A2.

The core taxonomies

The 4 core reference taxonomies are aligned to the 4 core dimensions of GEA-NZ:

- Business
- Data and Information
- Application and Software Services
- Infrastructure.

The fifth taxonomy, named Other Categories Reference Taxonomy, is used for terms that are not included in the 4 core reference taxonomies.

- Other Categories Reference Taxonomy

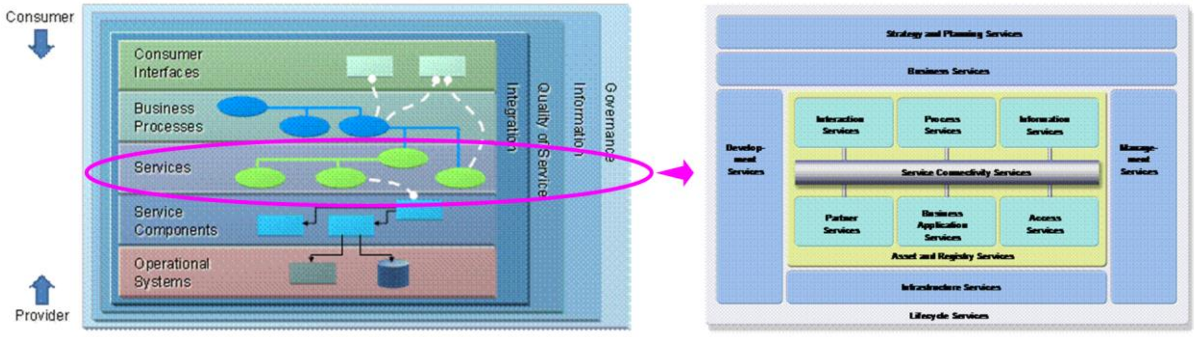
Figure 22: GEA-NZ Reference Taxonomies Snapshot

4 Whole-of-Government Tier

The Whole-Of-Government (WOG) tier may be thought of as a container for the common or shared capabilities required to support a WOG or collaborative approach to digital government, specifically government digital asset development, discovery, (re)usability, sharing, analysis, and the like. The operative term here is *collaborative*: the simple criterion for selecting reference and solution architectures into the WOG tier is that they must provide capabilities that support collaborative activities; the other architectures are placed in MDA tier because they are premised to focus on MDA-specific capabilities.

4.1 Architectural Perspectives

Again, the SOA model is used to formulate GEA-ET perspectives on the different architectural concerns that the WOG architectures must address. This formulation draws on the logical solution view of the [official] SOA reference architecture (SOA-RA) [@] (see Figure 23), which expands the conceptual SOA view of Figure 3 to reveal the logical components that implement a SOA, also organised within horizontal and vertical layers.



Logical Solution View

Service Layer Functional View

Figure 23: SOA-RA: Logical Solution & Functional Service Layer Views

In this view, the lower layers (services, service component, and operational systems) are provider concerns and the upper ones (services, business process, and consumer) are concerns for the consumer. The horizontal layers are more functional in nature and relate to the functionality of the SOA solution, whilst the vertical layers are supportive of cross-cutting concerns that span the functional layers but are clustered around independent notions themselves as cross-cutting concerns of the SOA architectural style. A more detailed description of the SOA logical solution view can be found in the SOA-RA documentation here [@].

Another view of interest is the functional delineation of SOA services into the service categories depicted in Figure 23; services are categorized according to what they do (i.e., their function or purpose). This view is of interest here because it can be used to assess the SOA's coverage of architectural requirements.

It can also be presented to stakeholders as a more accessible description of the SOA, which can aid in understanding the SOA and the portfolio of services that supports it. The SOA-RA specification provides a fuller description of the SOA service layer here [@].

Furthermore, it is important to note that the SOA-RA defines a reference or standard categorisation scheme for services. However, other delineation schemes are possible, and as part of the GEA-ET development process, it may be helpful to review other [non-SOA] schemes for coverage assessment purposes. Examples of such schemes are the GovStack public platform digital reference architecture (see Figure 24), and various country-developed architectures; although not all the services they depict are necessarily realised by the SOA, the categorisations of these schemes can point to SOA services gaps which may need to be resolved.

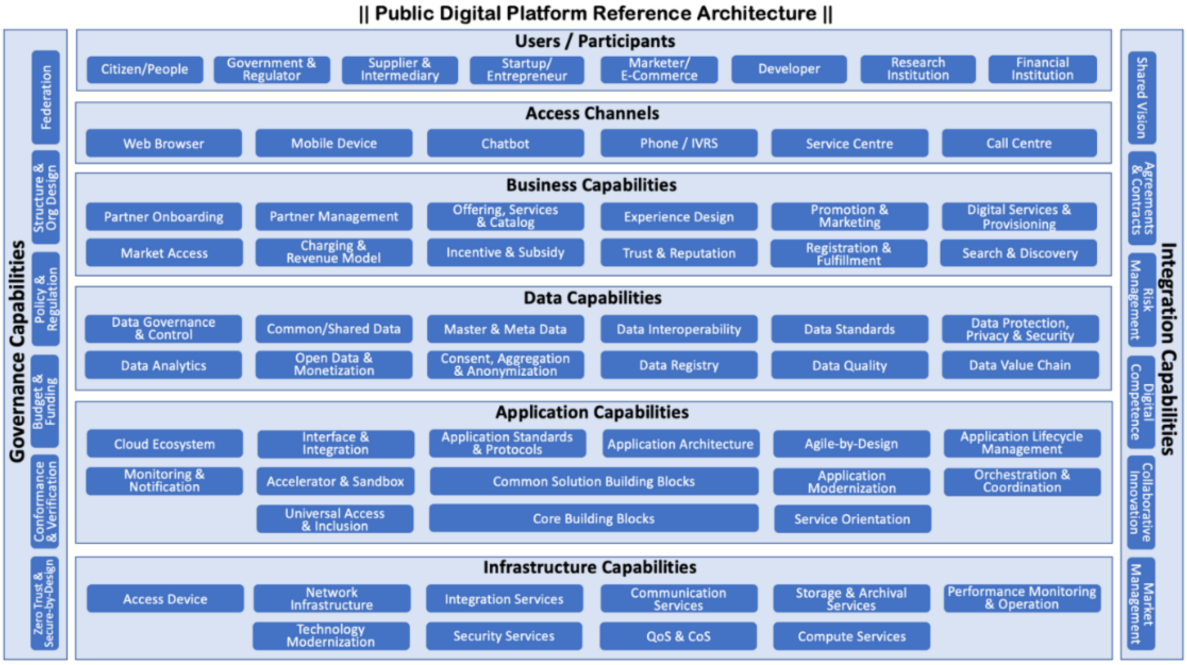


Figure 24: GovStack Public Digital Platform Reference Architecture

The linkage between service groupings and SOA layers should be understood as follows: service groupings represent functionally aligned capability clusters that address specific architectural concerns, whereas the SOA logical solution view specifies how those capabilities should be implemented by planned systems (or are implemented by existing systems) in support of each service grouping. Note that the service delineations are distinct and separate from the collection of SOA layers (there may be more than one layer) that collaborate to deliver each service capability.

GEA-ET specifies architectural perspectives for the WOG tier, that is, reference architectures that model idealised logical designs for each SOA layer and/or service group. In this regard, the modelled architectural perspectives were selected both to ensure coverage of strategy-prioritised capabilities and to increase stakeholders' understanding of the GEA-ET, resulting in the portfolio of domain-aligned reference architectures (deliverables) listed in Table 22.

The architecture portfolio at the WOG level also includes the solution architectures for the prioritised shared solutions (i.e., the priority DBA, ESB, NDS and DXP solutions) which have already been identified as the first set of GEA-ET implementation projects.

The solutions are specific implementations of the reference domain architectures and should viewed as new/enhanced systems that run alongside existing systems in the operational systems layer. For example, in general, the ESB solution implements integration layer requirements, whilst the AP solution fulfils services layer functions. The design of the new priority solutions should ensure interoperability across layers, but many existing systems will likely require modification to achieve such interoperability. Such modification may include creating application or data components in the service components layer, for access or orchestration by services in services layer or consumers in the consumer interfaces layer.

The mapping between architectural domains and reference architectures is generally one-to-one, but a similar mapping between reference and solution architectures (or solutions) is unlikely in practice, because multiple solutions are required to implement a reference architecture in the typical case. As an example, business automation and user portals would naturally delineate as components of the business architecture, even though they would likely be designed and implemented as distinct and separate solutions.

Furthermore, most reference architectures are composite arrangements in the sense that they may refer to solutions that have their own reference architectural representations; such references will often be reused in multiple architecture to provide clarity and should not be interpreted as duplications.

The sections that follow present descriptions of the GEA-ET reference architectures, along with some of the conceptual and technical frameworks that underpin their development.

4.2 Business Architecture

As expressed in the TOGAF standard, in part, a business architecture describes how an enterprise needs to operate to achieve its business goals and respond to its strategic drivers in a way that addresses stakeholder concerns.

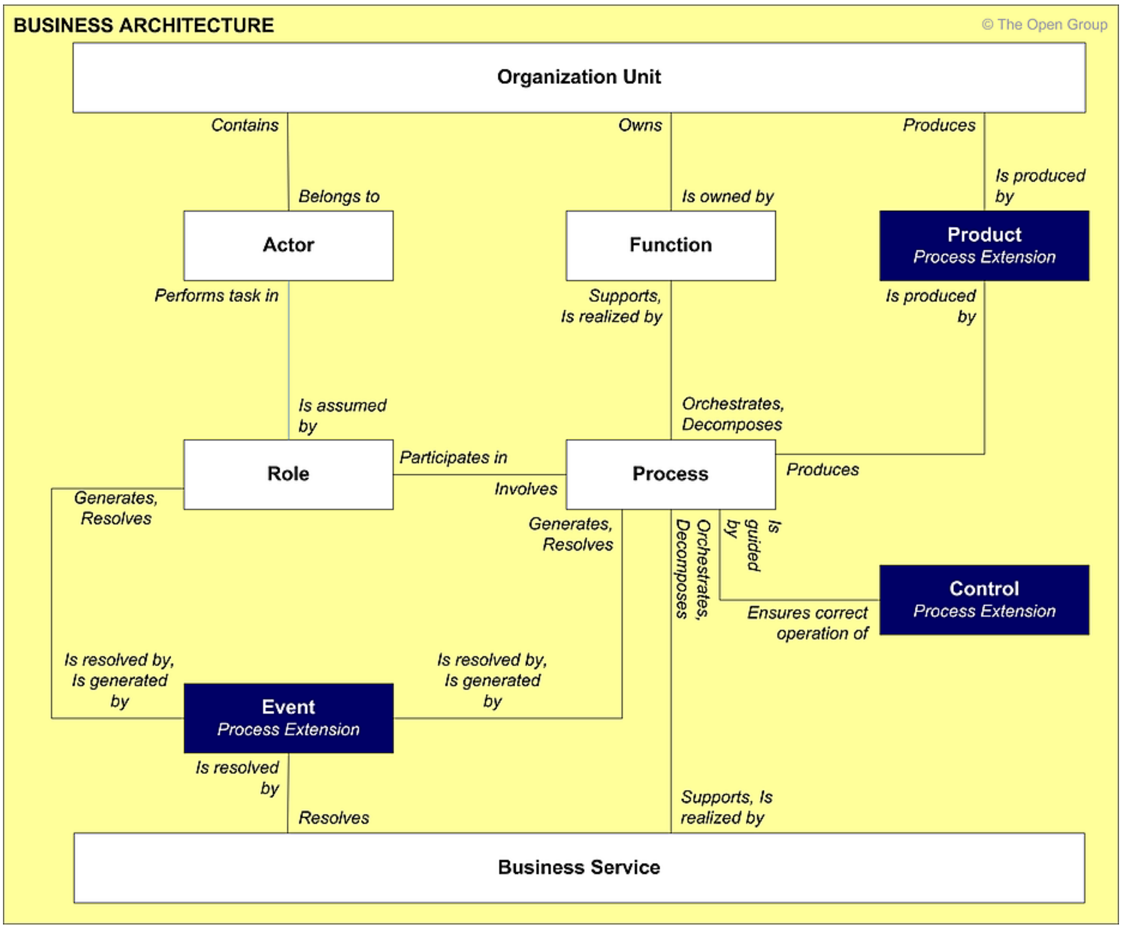


Figure 25: Business Architecture Content Framework

The GEA-ET business architecture translates the requirements identified in the digital government strategy and through the GEA diagnostic into concrete specifications, expressed in terms of the entities and entity relationships shown in Figure 25, as well as related artefacts in Table 2 that form part of the rationalised deliverable portfolio.

4.2.1 Development Strategy

As stated earlier for the reference model development scenario, the primary effort in developing the business architecture involves identification and elaboration of the business architecture entities. Phase B (Business Architecture) of the TOGAF ADM provides a detailed description of the business architecture development process, including usage of advanced analysis techniques and guidelines [@].

However, given the complexity of the standard TOGAF approach, these broad steps describe an adapted and simplified process that should suffice for developing the GEA-ET business architecture:

[1] Apply design thinking techniques to derive user journey maps that identify the key service users (including programmatic service consumers) and their experiences, which are, essentially, high-level end-to-end business process descriptions;

[2] Elaborate journey maps in business architecture terms to identify key entities like organisations, actors, roles, functions, business services and processes (including events, controls, and products); this activity generates artefacts like stakeholder catalogues, role/actor matrices, process flow diagrams, and business service/capability matrices;

[3] Use the outputs of steps [1] and [2] to identify the required capabilities within the consumer layers (consumer interfaces and business processes), as well the corresponding capability providers within the provider layers (service components and operational systems); this step should also crystallise the services layer catalogue and identify any gaps between needs and capabilities.

The outputs of the specified business architecture are then used to guide development of the solution architectures that translate the [logical] reference architectures into detailed implementable designs. In turn, the designed solution architectures guide the downstream configuration and/or development of the related solutions.

Described next are two reference architectures proposed to address the prioritised horizontal business architecture concerns – business automation, which delivers capabilities for the business process layer, and interaction portal, which fulfils consumer interface functions.

4.2.2 Digital (Business) Automation Architecture

Digital business automation enables improvement of enterprise operations by streamlining the way that people participate in business workflows, automate repeatable decisions, and provide business users with the ability to edit and change the business logic involved in these business processes. Business automation also aims to make documents easy to store and retrieve, capture and structure document content, and automate manual tasks with robotic process automation.

The reference business automation architecture (Figure 26) points to the kind of automation solutions that can be deployed to streamline or automate these business dimensions with the following capabilities:

- Workflow management: Orchestrates tasks between humans and systems, keeps track of what is being processed, provides visibility on team workloads and workflow status and progress, and tracks execution to derive business improvement insights.
- Content services: Stores and organises a variety of content, so users can easily access and retrieve relevant content in a governed manner.

- Decision management: Automates repetitive decisions and provides decision makers with information in a readable, easily updated format to reflect policy changes.
- Document processing: Use optical character recognition (OCR) and other data recognition techniques to analyse, classify, and extract data from documents.
- Robotic process automation (RPA): Automate repetitive tasks, such as keying in data, across multiple user interfaces and systems.

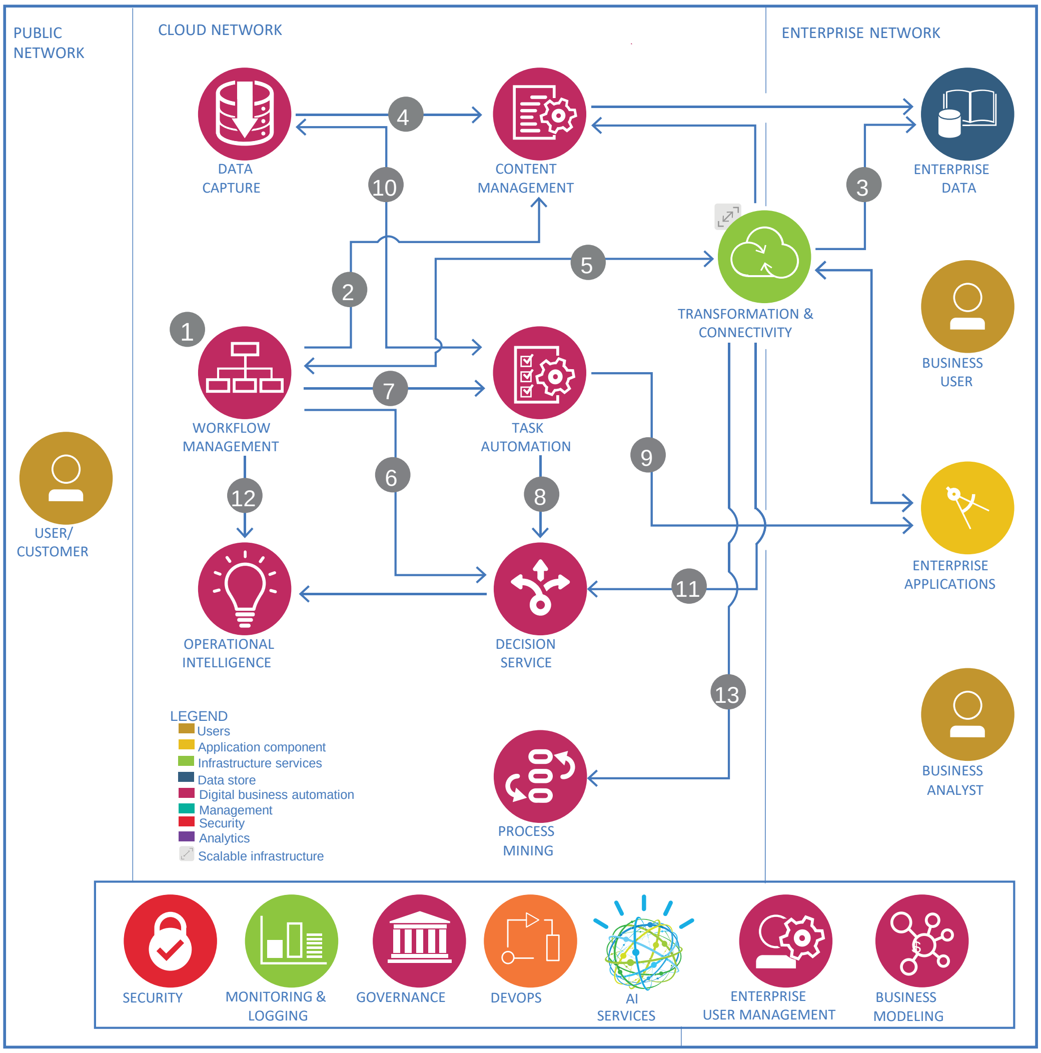


Figure 26: Digital Business Automation Reference Architecture

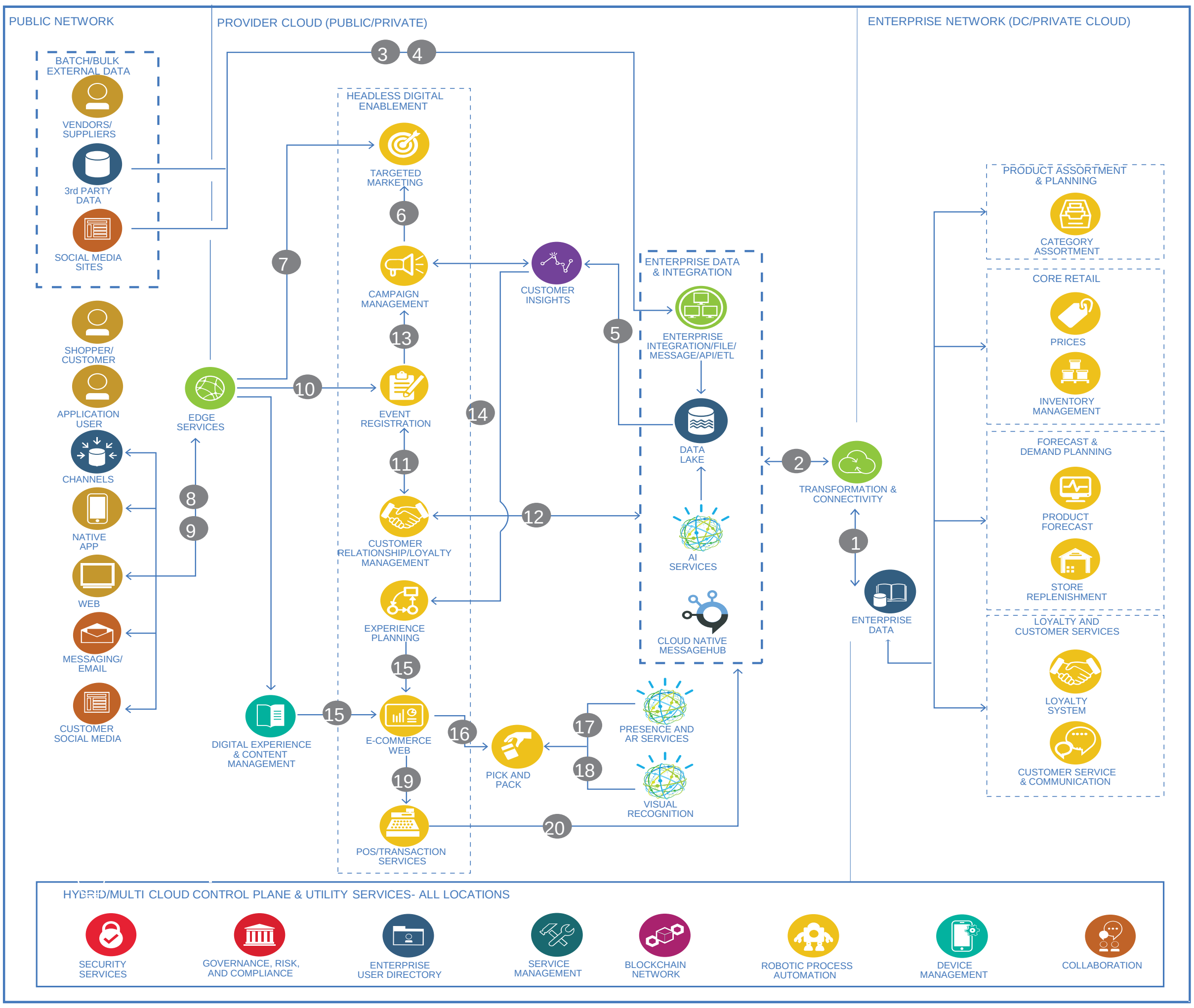
Source: IBM Architecture Centre (Adapted) [@]

Workflow management is a key capability for the SOA business process layer. It controls and instruments business operations that can involve humans or systems interacting and contributing to the execution of a business process via components in the SOA consumer interfaces layer. For human interactions, a workflow management solution ideally provides a single portal and user experience across all workloads for business visibility.

The solution can enhance automation by orchestrating calls to other services and systems, like data capture and content services, via integration layer services denoted by the “transformation and connectivity” node. It captures operational data from the execution of these workflows to compute key performance indicators (KPIs) and dashboards that impart business insights.

4.2.3 Digital User Experience Architecture

Closely connected to digital business automation concerns is the digital user experience (DUX) design problem, which seeks to craft digital experience platforms (DXPs) that can reach and engage disparate audiences across multiple digital touchpoints to enhance the overall engagement experience for the consumers of digital services.



Source: IBM Architecture Centre (Adapted) [@]

Figure 27: Digital Experience Platform Reference Architecture

Whilst some governments have made progress in improving their user engagement models, it is evident the retail industry is a clear leader in this space, having developed and deployed best-practice DXPs in an effort to attract and retain customers through optimised customer experiences. Therefore, GEA-ET proposes leveraging relevant aspects of DXP architectural patterns that have been developed in the retail industry and adapting them to support the Ethiopian digital government strategy.

Although retailers mainly target customers whilst GEA-ET targets a diverse user base (e.g., citizens, businesses, NGOs, MDAs, etc.), this retail-based approach makes sense because of the convergence in digital experience objectives, as demonstrated by the [non-exhaustive] listing of common objectives in Table 6.

Table 6: Common Retail & Government DUX Objectives

Objective	Retail Target	GEA-ET Target
Enhanced service access	Multi-channel customer engagement	Multi-channel user engagement
Offering uptake	Increased sales	Increased use of eServices
User retention	Repeat sales	Repeat use of eServices
Channel redirection	Increased use of digital channels (ahead of physical channels)	Ditto

The digital experience platform reference architecture of Figure 27 depicts a solution arrangement that supports best-practice [retail] user experiences, which includes the following important DXP capabilities (some are only implied and not explicitly shown):

- Campaign management: Supports the design of specific campaigns to drive attainment of the offering uptake and user retention objectives.
- Multi-channels: Enables seamless, consistent, and exciting experiences across multiple channels, including physical contact centres, web, mobile, voice, and more.
- Experience insights: Uses engagement tracking and user feedback data to derive insights that can help improve user experiences.
- Customer experience management: Handles and correlates bidirectional customer contacts via various channels such as call, e-mail, social media, contact form, and online forum, including complaints and information requests.
- eCommerce web: Integrates e-commerce technologies such as online shopping to support ordering and distribution of digital and physical products and services.
- Event management: Manages planned face-to-face and virtual user engagement events including event registration, attendance tracking, and user event feedback.
- Experience planning: Enables development of user-group specific experience plans to enable detailed experience tracking and insights.
- Multi-payments: Supports multiple payment types including POS, cards, mobile money, bank deposits, and more.
- Social media sites: Targets the use social media as a means to engage with users on multiple topics.

It will be appreciated that these diversified solution options cannot all be adopted from the outset, but this DXP reference architecture provides some useful insight into what is possible with DXP technologies.

In this regard, a key action for the GEA-ET business architecture team is to assess the functional and technical design of the existing GoE digital experience platform to determine and prioritise improvements can be made to deliver these reference capabilities as part of the GEA-ET implementation.

4.3 Data Architecture

As shown in the data extension view of the TOGAF content metamodel (Figure 28), a data architecture consists of data entities and logical/physical data components, which have various usage and consumption relationships with business and application architecture components, that exist within the context of business services. The data components reside within the operational systems layer of the SOA-RA (Figure 23), where they implement service components that provide capabilities to support functional services categories like information and access services.

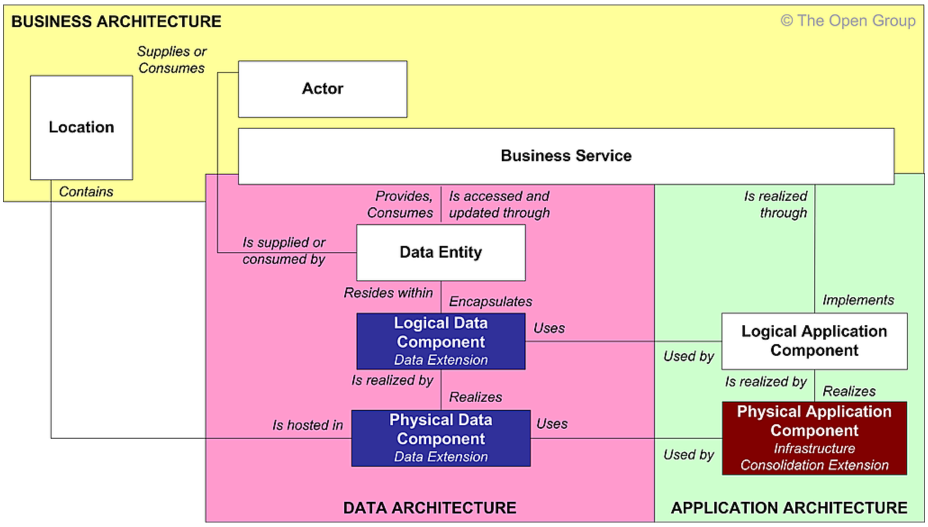


Figure 28: TOGAF Content Metamodel: Data Extension View

In this regard, it is to be noted that GEA-ET is not concerned with the internal structuring and management of data within the operational systems. Rather, it focuses exclusively on capabilities that enable data interoperability across those systems, to enable creation, discovery, (re)use, and protection of shared data. Thus, MDAs have the latitude to decide the way data is to be organised and managed internally, but they must align with the mandated standards for exchanging shared data.

4.3.1 Capabilities

For prioritised common services, the data capabilities cluster around two competencies:

- Data management: The practice of acquiring, organising, and distributing data to support productivity and decision-making, incorporating a wide range of tasks such as integrating disparate data from diversified sources, governing how data is used and accessed by people and apps, protecting data, and ensuring data privacy.
- Data analytics: Refers to monitoring, reporting and visualization of resources and activities, and leveraging that information to optimise strategy and operations, including capabilities, such as integrated reporting, real-time updates, and aggregate analytics, entity-aligned activity monitoring.

The expected capability portfolios across the data management and analytics competencies are listed in Table 7 and Table 8 respectively.

Table 7: Data Management Capabilities

1	Data Capture	Data capture refers to the manual entry of new data by organisational staff or the capture of data generated by devices used in various processes and distributed throughout the organisation.
2	Data Acquisition	The collection of existing data that is produced within and outside the organisation.
3	Data Storage	Once created, data needs to be retained in primary, secondary, or tertiary storage, protected with the appropriate level of security, and retention ensured by implementing a robust backup and recovery process.
4	Data Staging	Data staging refers to the transitional or intermediate storage of data in a staging area, or landing zone, for the purposes of performing quality checks and transformation processes as the data moves between various source and target applications, such as operational and analytical data stores, or other data repositories.
5	Data Quality Management (DQM)	Data quality management provides a context-specific process for improving the fitness-for-purpose of data that's used for operations, analysis and decision making, with the goal of gaining insights into the health of that data using various processes and technologies on various data sets.
6	Data Transformation	Data transformation essentially entails the conversion of data structure and content from a source to a target format: the source structure is mapped to the target structure, and then the content of each target data element is replaced with the corresponding source data element in the target data format.
7	Data Usage	Data usage is a phase of the data lifecycle when data is used to support various applications and business processes within an organisation. Data can be viewed, processed, modified, and saved, and an audit trail should be maintained for all critical data to ensure that all modifications to data are fully traceable.
8	Data Archival	Data Archival is the process copying of data to an ‘off-line’ environment where it is stored in case it is needed again in an active production environment, and the removal of this data from all active production environments.
9	Data Destruction	Data destruction is the removal of every copy of a data item which is often done from an archive storage location; it is important to ensure that the data is properly destroyed and to that the data items have exceeded their required regulatory retention period before purging.
10	Data Lifecycle Management (DLM)	Data lifecycle management describes a process used to control data throughout its lifecycle, whose steps may be defined generically as creation (capture or acquisition), storage, usage, archiving, and destruction - these steps are described in other entries in this table.
11	Metadata Management	Metadata management is a transversal agreement on how to define informational assets for converting data into an enterprise asset. It incorporates taxonomy management which classifies data into categories and sub-categories to provide a unified view of the data and common terminologies and semantics across multiple systems. Establishing a hierarchy within a set of metadata and segregating it into categories creates a better understanding of the relationships between data points; enumerators form key components of such hierarchies.

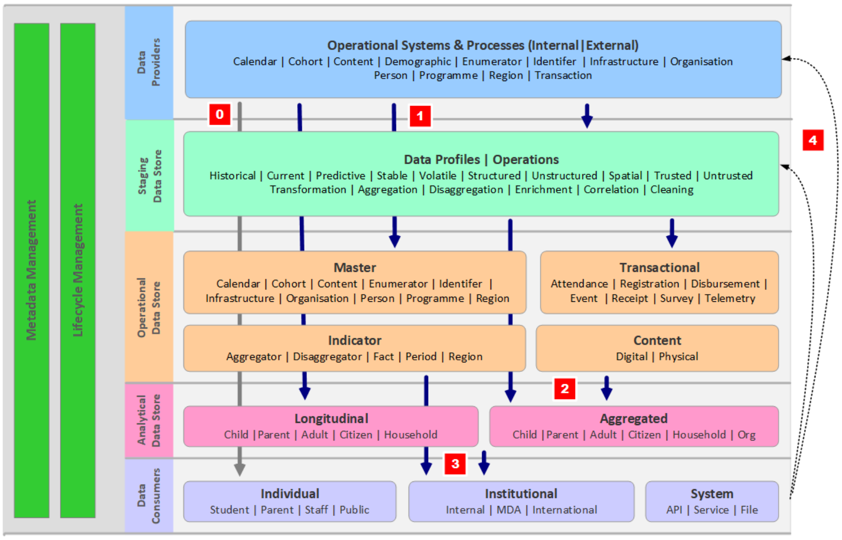
Table 8: Data Analytics Capabilities

1	Integrated Reporting	Consolidates reporting across all sources by leveraging the standardised reporting platform that a metamodel provides.
2	Trend Analysis	Compares performance data over time to identify consistent performance trends and patterns (both positive and negative) that can be used to guide strategic and operational decisions, and it is important to enable analysis of these across multiple dimensions e.g., education level, admin level, region (location), time, etc.
3	Dashboards	Enables operational systems to immediately communicate (push) status changes in monitored resources and operations so that they are analysed and made visible to analytics consumers in near real-time; achieving this objective obviously depends on network

		connectivity conditions but the goal in all cases should be to communicate all changes as soon as possible.
4	Aggregated Indicators	Supports static and trend analysis of all established sector indicators as mandated by, as well as indicators that may be required for external reporting purposes.
5	Individual/ Cohort Indicators	Relies on longitudinal tracking capabilities to monitor and report on activities at an individual or cohort level. A cohort is a collection of persons that can be managed and tracked as a group, typically created for operational reasons (e.g., special programmes, extra-curricular activities, disciplinary controls, etc.)
6	Self-Service Analytics	Provides analytics consumers with capabilities that enable a degree of self-directed data analysis and reporting, without excessive reliance data analytics experts; self-service is typically enabled by making datasets available to end-users in an accessible format.
7	Location-Based Analytics	Augments data analytics outputs with a layer of geographical data (including maps) to enable indicators and trends to be analysed by geographic location.

4.3.2 Pipeline Architecture

A data operations pipeline architecture (DPA) represents a near-pervasive data integration use case that can likely fulfil the early GEA-ET data management and analytics requirements. DPA is a minimal design that simply moves data from one system or format to another, with each movement executing operations such as acquisition (data extraction or reception from a source), transformation (cleaning and structuring), and distribution (making data accessible or loading it into a destination system) (see Figure 29). Data operations pipelines can be used for a variety of purposes, including data integration, data warehousing, automating data migration, and analytics.


Metadata Management
Lifecycle Management
Data Providers
Operational Systems & Processes (Internal|External)
Calendar | Cohort | Content | Demographic | Enumerator | Identifer | Infrastructure | Organisation
Person | Programme | Region | Transaction
0
1
4
Staging Data Store
Data Profiles | Operations
Historical | Current | Predictive | Stable | Volatile | Structured | Unstructured | Spatial | Trusted | Untrusted
Transformation | Aggregation | Disaggregation | Enrichment | Correlation | Cleaning
Operational Data Store
Master
Calendar | Cohort | Content | Enumerator | Identifer | Infrastructure | Organisation | Person | Programme | Region
Transactional
Attendance | Registration | Disbursement | Event | Receipt | Survey | Telemetry
Indicator
Aggregator | Disaggregator | Fact | Period | Region
Content
Digital | Physical
2
Analytical Data Store
Longitudinal
Child | Parent | Adult | Citizen | Household
Aggregated
Child | Parent | Adult | Citizen | Household | Org
3
Data Consumers
Individual
Student | Parent | Staff | Public
Institutional
Internal | MDA | International
System
API | Service | File

Core Components

- Data Providers: Internal and external operational systems whose business processes generate and consume the domain data which populates the ADS and ODS.
- Staging Data Store (SDS): Transient data store which holds data that is being prepared for downstream processing, analysis, distribution, and consumption.
- Operational Data Store (ODS): Persistent data store which sets common data standards by implementing the metamodel for master, transactional, and indicator data categories, and digital and physical content.
- Analytical Data Store (ADS): Data store whose structures are optimised for aggregated or longitudinal analytics, including data marts and warehouses, and data lakes.
- Data Consumers: Represents the core individual, organisational, and system data consumers who are serviced from the ODS and/or ADS.

Data Flow

[1] Provider data is acquired, transformed and persisted in the operational data store (ODS) and/or the analytics data store (ADS), optionally via the staging data store (SDS) for preparation,

[2] ODS data is transferred to and persisted in the ADS, optionally after optimisation for analytics workloads,

[3] ODS/ADS data is made available for distribution to data consumers, who,

[4] May also be data providers themselves.

Figure 29: Data Operations Pipeline Architecture (DPA)

DPA patterns optimise data integration use-cases in which multi-sourced data is acquired and processed to create information, and subsequently analysed and/or distributed to multiple consumers. Pipeline design depends on various factors, such as how data is received, the business use cases, and the data volume, with some of the common design patterns being:

- Raw Data Load: Moving and loading raw data from one location to another, such as between databases or from an on-premises compute centre to the cloud.

- Extract, Transform, Load (ETL): Widely used for loading data into data warehouses, lakes, and operational data stores, which involves the extraction, transformation, and loading of data from one location to another.
- Streaming ETL: Like the standard ETL pattern but with data presenting in streams rather than batches.
- Extract, Load, Transform (ELT): ETL-like pattern which can reduce transfer latency, but the loading operation precedes transformation.
- Change, Data, Capture (CDC): Injects freshness to data processed using ETL batches by sensing and proactively notifying data changes to downstream operations.
- Stream Processing: Data is continuously ingested and processed in-line (usually in-memory without persistence) and then continuously distributed to consumers.

Post acquisition, the data is typically routed through a ‘staging’ node (staging data store) where it undergoes transformation and classification (master, transactional, indicator, etc.) and is either preserved for operational use (operational data store) or routed straight through to analytical nodes (longitudinal and aggregate data stores) for analysis, or distributed to data consumers through various channels, who may include the source systems themselves.

The data pipeline architecture has a distinct relationship with the hybrid integration platform (HIP) architecture (see Section 4.5.2): DPA implements and manages data structures and repositories (the blocks in the diagram), whereas HIP is primarily concerned with executing and managing the movement of data between those repositories (the arrows in the diagram).

Note that the trend among vendors and developers of data management solutions is consolidation of DPA capabilities onto single integrated platforms (the previous common practice was to develop specialised products for each design pattern), and that the distinction between HIP and DPA offerings is increasingly becoming fuzzy.

4.3.3 Data Reference Architecture

GEA-ET proposes that the reference DPA instance shown in Figure 30 as a data reference architecture. It reflects data management and analytics solutions required to support the digital government strategy, including manual and automated data acquisition, and data staging, which can entail multiple tasks such as data qualification, transformation, and enrichment. The data reference architecture also incorporates capabilities that support aggregated and longitudinal data analytics, and distribution of raw and analysed data to system and human consumers.

It is important to note that the staging store represents a generalised repository for storing all incoming data, in a governed and secured environment where it can be ‘prepared’ *in-situ* for distribution to operational stores, analytics stores, or directly to consumers or operational systems. But it is not intended the serve the ‘data lake’ use case, which aims to enable *in-situ* analytics on raw, unprocessed data.

In contrast, the operational store provides three important capabilities: (a) implements the shared metamodel schema which can be used as an enterprise-wide data exchange standard, (b) stores standardised metamodel content, as required, and (c) serves as a source of metamodel-compliant reference data. Operational and staging store capabilities can be implemented on the same physical repository, but best-practice suggests that they be deployed as separate physical stores.

Table 9 provides summary descriptions of the numbered components in the data reference architecture diagram, along with their respective capabilities.

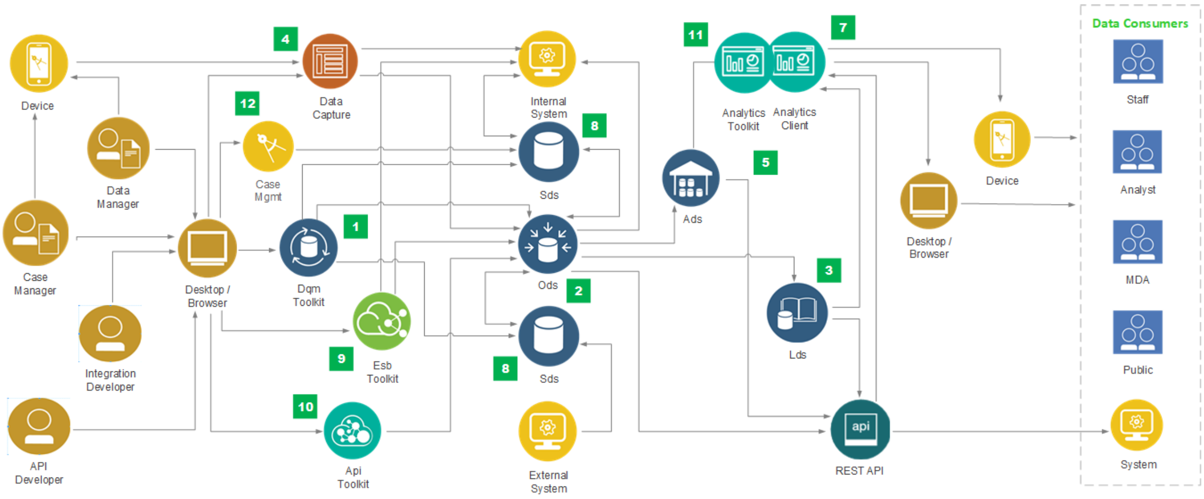


Figure 30: Data Reference Architecture

Table 9: Information Architecture: Components & Services		
1	Data Quality Manager	In line with the clean-source architectural principle, this data provider ‘firewall’ ensures that clean data enters the operational environment.
2	Operational Data Store	Primarily responsibilities are to: (a) implement the shared metamodel schema, (b) serve as a repository of standardised metamodel content, and (c) serve as a source of metamodel-compliant reference data.
3	Longitudinal Data Store	Implements a longitudinal data storage capability which enables process and event tracking and analysis with case-based follow-ups, at a personal or individual level.
4	Workflow Client	Supports structured capture of longitudinal process and event data with rapid configuration capabilities to support less structured data capture scenarios like surveys.
5	Aggregated Data Store	Supports the collection, analysis, visualization, and use of aggregate data with capabilities to organise data into specialised data marts for detailed analysis.
6	Metadata Repository	Supports the metadata management function by providing a metadata content repository for storage and management of all information-related assets.
7	Data Analytics Client	Provides data analytics capabilities such as visualizations, maps, and dashboards with real-time updates from data sources.
8	Staging Datastore	Serves as a transitory datastore for data quality management operations before data is released to target systems – logically separate staging stores must be created for internally and externally sourced data (Sds).
9	Esb Toolkit	In this context, application integration developers use the Esb Toolkit to implement the data transformations that align internal application data formats with agreed metadata standards.
10	Api Toolkit	In this context, API developers use the Api Toolkit to access the metadata standard schema for the purposes of defining API specifications.

11	Analytics Toolkit	This component represents the collection of tools that support self-service analytics, which are to be contrasted with the analytics client which provides pre-configured analytics.

The National Data Set (NDS) is representative of the kind of use case that the DPA model is designed to address. In this regard, the NDS initiative [@] independently proposed a conceptual architecture that reflects data management capabilities supportive of EDGS goals, although the content would require modernisation to reflect current technologies.

Comparison of the conceptual NDS and DPA designs reveals some notable similarities in approach, components, and content, suggesting that the NDS master plan could be a good starting point for developing data management and analytics solution architectures that align with GEA-ET.

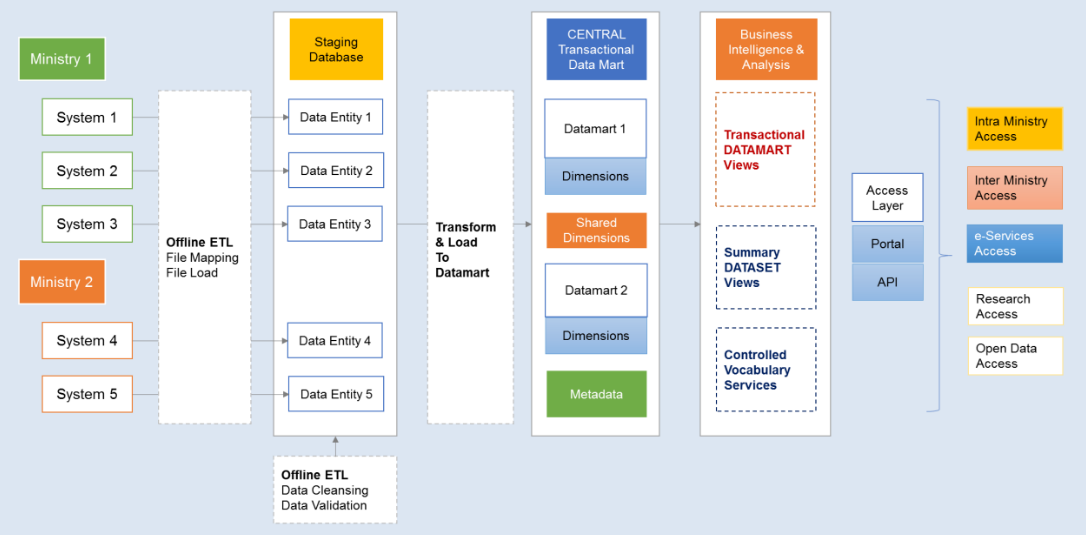


Figure 31: Conceptual ENDS Solution Architecture

4.4 Application Architecture

The GEA-ET application architecture has identified the common applications listed in Table 10, derived mostly from the EDGS, and adopts two different approaches to developing the related reference and solution architectures:

- Reference architectures are specified for all prioritised common applications at the WOG level (e.g., DBA, NDS, etc.) – the links in the table point to these specifications.
- However, the related solution architectures, along with the reference architectures for the other common and MDA applications (e.g. GRP, EML, MDA*, etc.), will be developed downstream.

Note that MDA* represents MDA applications that directly support the delivery of eServices and any other shared services, and that downstream architectural deliverables will be scheduled in alignment with the EDGS implementation roadmap, including common application and MDA reference and solution architectures.

Furthermore, in line with the suggested "solution focus" TOGAF adaptation tactic (see Section 2.4.1), the requirement to develop application reference architectures should be optional for MDAs who are expected to maintain focus on interoperability concerns at the solution level.

Table 10: Application & Service Category Portfolio

Tier	Application | Service Category	Description
WOG	Digital Business Automation (DBA)	Enables improvement of enterprise operations by streamlining the way that people participate in business workflows, automate repeatable decisions, and provide business users with the ability to edit and change the business logic involved in these business processes.
WOG	National Data Set (NDS)	Provides data modelling, extraction, aggregation, cleansing, validation, transformation and loading services for nationally relevant datasets; business intelligence and data analysis; consumer data interaction mechanisms.
WOG	Enterprise Service Bus (ESB)	Delivers process, data, and application integration capabilities across diversified data sources and sinks, including transformation, routing, orchestration, and API management services (aka enterprise service bus).
WOG	Digital Experience Platform (DXP)	Delivers capabilities that seek to reach and engage disparate audiences across multiple digital touchpoints to enhance the overall engagement experience for the consumers of digital services.
WOG	Programme Management (PGM)	Provides programme management functions that enable planning, execution and control multiple related projects and projects (aka monitoring and evaluation).
WOG	eProcurement (EPC)	Provides MDAs and partners with common platform to transact, with features such as demand aggregation, catalogue based procurement, dynamic pricing engine, etc
WOG	Human Resource Management (HRM)	Government wide application to handle all functions related to management of human resources e.g., recruitment, leave, transfer, payroll, etc.
WOG	ePayment Gateway (EPG)	Centralized payment platform enabling residents and businesses to transact around digital services in a secure and easy manner.
WOG	Customer relationship management (CRM)	Enables MDAs to better manage their client interactions through the introduction of reliable systems, processes, procedures and underlying operating model; these capabilities are partly covered by the DXP architecture.
WOG	Government resource planning (GRP)	Enables automation and processes in finance, human resources, manufacturing, supply chain, services, procurement, and more.

WOG	Knowledge/ collaboration management (KCM)	Aims to improve collaboration and knowledge sharing across MDAs, transforming them into knowledge-based organizations and providing features such as engagement management, self-services, and shared application access.
WOG	Document & records management (DRM)	Enables transformation of all documents and records to electronic files and store them in an indexed central repository.
WOG	Geospatial Data Platform (GDP)	Provides consolidated storage and analytics of geospatial data including maps information and integration with digital service delivery applications.
MDA	Solution Architecture (MDA*)	Represents the collection of MDA solutions that either exist or will be implemented to deliver digital services.

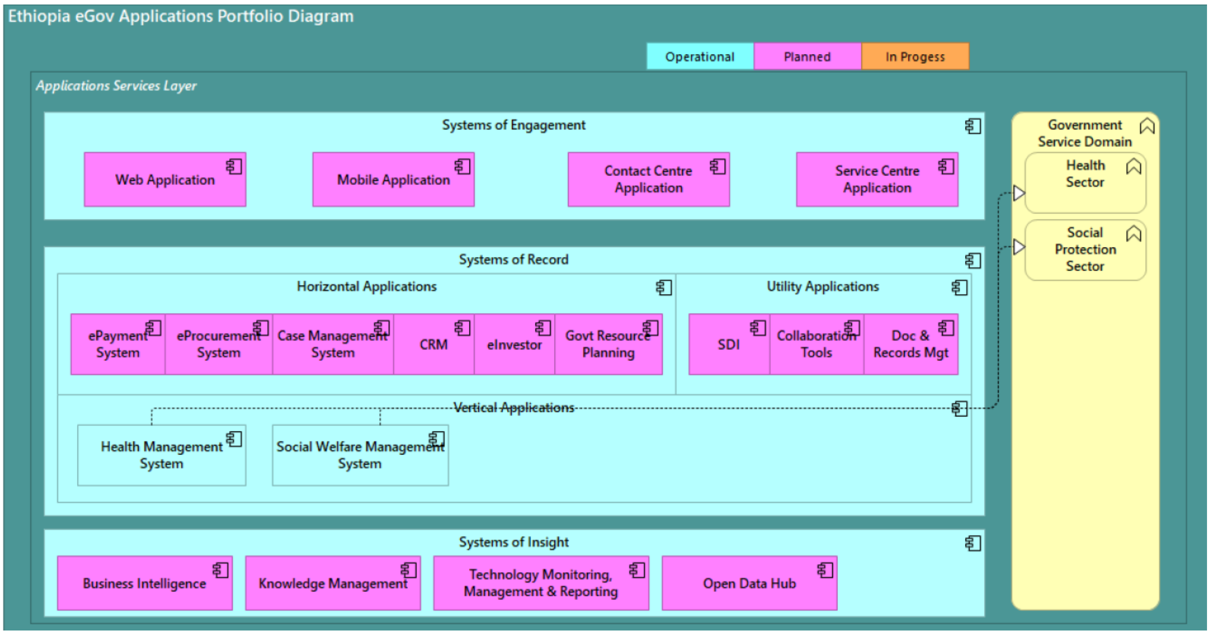


Figure 32: Reference Application Portfolio

The schematic of Figure 32 depicts a possible arrangement for the portfolio of applications listed in Table 10, which groups applications into three primary categories:

- Systems of Record: Represents high priority systems that serve as the authoritative data sources for mission critical information; identification of such systems for critical datasets is important when data is acquired from different source systems is then (re)processed and (re)used.
- Systems of Engagement: Denotes applications that facilitate and orchestrate the customer journey via more personalized, seamless interactions across the various touchpoints, as exemplified by the DXP architecture discussed in Section 4.2.3; they capture demographic, behavioural, interaction, transaction and affinity information on entities in "systems of record" to facilitate longitudinal tracking.
- Systems of Insight: Incorporates the class of applications that support and improve the user experience through the consumption, collection, and analysis of data from the combined sources of traditional "systems of record" and "systems of engagement.
- Systems of Enablement: Refers to applications that implement non-functional capabilities that enables other applications, including the integration, security,

development, and operations capabilities that are represented as vertical layers in the SOA-RA model (see Figure 23).

Such application typing is helpful as an aid to understanding the enterprise and MDA application landscape, since each category encapsules specific application functional traits, and application non-functional requirements (e.g., systems of record generally demand greater resilience and security levels than the other system types). Application typing information is particularly useful in guiding the planning and implementation of systems of enablement.

Vertical MDA applications generally delineate only as systems of record since MDAs legally own and execute government mandates, and in line with the service orientation principle, GEA-ET focuses on the external characteristics of MDA applications and does not need to understand the implementation of their insight and engagement capabilities.

4.5 Integration Architecture

Within the GEA-ET context, integration (aka interoperability) refers to interaction, process, application and data harmonisation across the operational systems in a SOA (see Figure 23). Previous related work [@] identifies three interoperability dimensions:

- Organisational Interoperability: Refers to collaboration between entities in the development, deployment and delivery of digital government services, and the interaction between such services and their supporting processes;
- Semantic Interoperability: Ensuring that the meaning or interpretation of information is standardised across disparate producer and consumer systems, where the primary concern is data content;
- Syntactic (Technical) Interoperability: The most basic interoperability concern which aims to ensure compatible data exchange mechanisms across producer and consumer systems, where the primary concern is data structure or syntax.

GEA-ET addresses organisational integration concerns in the business architecture, although this activity focuses on business process integration design within existing organisational structures, since matters of organisation design are addressed elsewhere. Semantic interoperability concerns are addressed by the data architecture since data meaning is properly addressed at the data entity level, this being one of the main objectives of the NDS initiative.

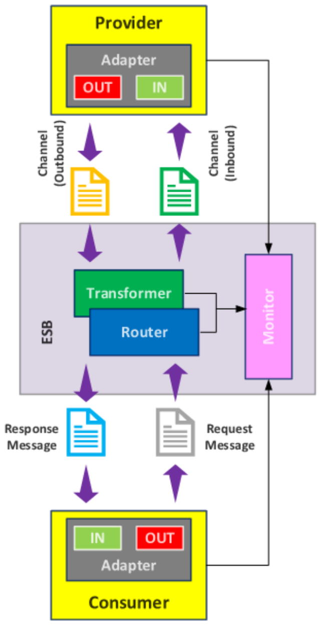
The integration architecture focuses exclusively on syntactic (or technical) interoperability concerns, including integration aspects such as data and application integration services interface specifications, connectivity services, data exchange structures and protocols, and protection of data communication channels.

4.5.1 Integration Framework

Figure 33 illustrates the key concepts that characterise the GEA-ET integration framework, where an integration process is modelled as a direct two-way exchange of data between a data provider and data consumer, or an indirect exchange via a mediating component (typically an ESB). An important related concept is the transaction process, which may be thought of as a sequence of integration process operations, which essentially involves the initiation of an integration process, and the passing its control from one component to another until it completes.

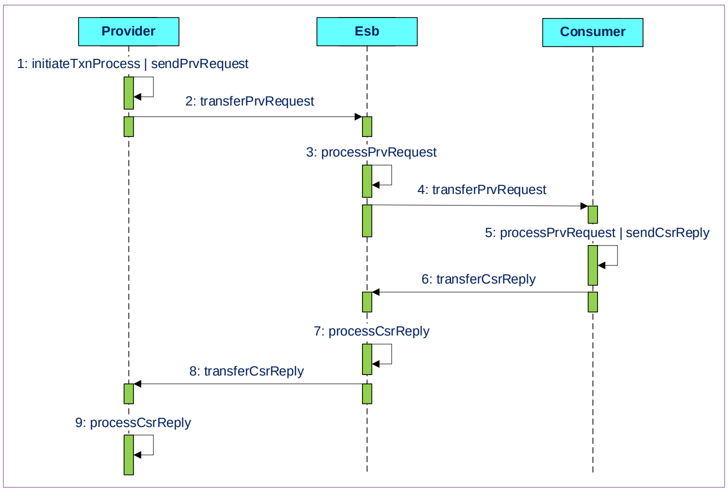
The sequence diagram of Figure 34 presents a structured representation (interface sequence diagram) of a specific transaction process pattern - a “Provider-initiated asynchronous request-reply” data interchange – which involves Provider (Prv), Esb, and Consumer (Csr) components, and Channels which are implied but not explicitly shown.

Integration flow sequences or patterns like this comprise reusable building blocks which can either be used alone or in concert to support simple or complex business processes; they can be characterised by the properties listed in Table 11:



- Application (Provider/Consumer): Represents the respective provider and consumer the data to be transmitted, in an integration process that can be initiated by the producer (push) or consumer (pull).
- Consumer (Receiver): The target application that receives the data to be transmitted.
- Channel: A virtual pipe over which messages are transmitted between a provider and a consumer; the channel is assumed to have duplex properties to enable bi-directional data flow.
- Message: An atomic packet of data that can be transmitted on a channel.
- Router (Broker): When a message flow is associated with multiple alternative channels and/or destinations, the original sender typically sends the message to a router which determines how to navigate the channel topology and direct the message to the final consumer(s) or at least to the next router in line.
- Endpoint (Adapter): An adapter is an application- and channel-aware component that performs several functions (a) connect the applications to the channel using a channel-compatible protocol, (b) provider adapters must break the data into one or more packets, wrap each packet as a message, and then send the message on an outbound channel; similarly, consumer adapters must extract the data from the message for processing after receiving it on an inbound channel, (c) optionally transform the data to a consumer-compatible format in the absence of a transformer component.
- Monitor (Manager): Monitors all transactions including sensing and notifying issues such as exceptions and performance bottlenecks, as well as general administration of the integration environment.
- Transformer (Translator/Mapper): Translates the message data into a format that is understood at both ends of the channel i.e., both provider and consumer applications.
- Enterprise Service Bus (ESB): A platform that provides integration services such as message transmission, transformation, routing, and monitoring, and publishes them for use by integration client applications.
- Transaction Process (Txp): Composite message flows which combine basic flows to execute the overall end-to-end needed to execute business processes.

Figure 33: Integration-Oriented Data Exchange Model



[1] Provider: Initiate transaction process | Generate provider request message; handover request message to outbound channel.

[2] Outbound channel: Receive request message from Prv; transport request message from Prv to Esb; handover request message to Esb.

[3] Esb: Receive request message from inbound channel; process (transform) request message; handover (route) request message to outbound channel.

[4] Outbound channel: Receive request message from Esb; transport request message from Esb to Csr; handover request message to Csr.

[5] Consumer: Receive request message from inbound channel; process request message; generate reply message; handover reply message to outbound channel.

[6] Outbound channel: Receive receive message from Csr; transport reply message from Csr to Esb; handover reply message to Esb.

[7] Esb: Receive reply message from inbound channel; process (transform) reply message; handover (route) reply message to outbound channel.

[8] Outbound channel: Receive reply message from Esb; transport reply message from Esb to Pvr; handover reply message to Pvr.

[9] Provider: Receive reply message from inbound channel; process reply message; generate ack message (optional); handover ack message to outbound channel (optional).

Figure 34: Transaction Process Model

Table 11: Transaction Process Properties

Property	Options
Initiator	Provider | Consumer | Monitor | Person
Routing	Request-reply | Pub-sub | Data-sync
Concurrency	Sync | Async
Granularity	Unary | Multiple
Protocol	Rest | Mq | Ftp | Manual | Sql | etc.
Format	Json | Xml | Csv | Fixed | Sql | etc.

- Initiator: The system component that initiates the data exchange can be a producer, consumer, monitor, or person, and the exchange can be triggered by varied events including business process steps, scheduled or conditional events, or person-driven manual actions. Note that the data of interest will always flow from provider to consumer, so that, it flows either on the inbound or outbound channel depending on the flow pattern.
- Routing: Reflects the pattern of data flow and the semantics of each step (message) in the flow; typical patterns include simple request-reply flows where senders wait for a receiver response, or request flows which are one-way message without receipt acknowledgement, and many others.
- Concurrency: Synchronous communication is characterised by a sender blocking to wait for a receiver response, whilst asynchronous exchange is non-blocking and can allow a sender to continue processing events and receive the reply at a later time.
- Granularity: This property refers to the number of transactions represented by message content which can be a single (unary) or composite (n-ary) transaction
- Protocol: This property reflects the channel communication protocol and supported standard (e.g., REST OAS, REST GQL or JMS).
- Format: The structure of the message payload or content (e.g., JSON or CSV).

These data exchange and transaction process models are described in detail here, because they explain concepts that are important to understand key characteristics of the proposed reference integration architecture. In addition, transaction process patterns provide an objective framework for specifying integration requirements (for example, in a solution architecture or a procurement solicitation) and assessing the capabilities of alternative integration platforms.

4.5.2 Reference Architecture

The GEA-ET integration reference architecture builds on the Hybrid Integration Platform (HIP) architecture (see Figure 35), to specify a single integration platform that can leverage new and existing integration assets to support the majority of enterprise integration requirements.

HIP is a services integration approach attributed to the Gartner Group, which facilitates centrally governed development, deployment and operation of the technologies, processes, and entities involved in the provision and consumption of integration services. It is a framework of on-premises and cloud-based integration and governance capabilities that enables differently skilled personas (integration specialists and non-specialists) to support a wide range of integration use cases.

HIP places API technology at the core of its design, positioning API Management as the primary façade (proxy) for interacting with and managing the HIP, to facilitate effective and centrally coordinated governance and operation of all technologies, processes, and entities involved in the provision and consumption of integration services.

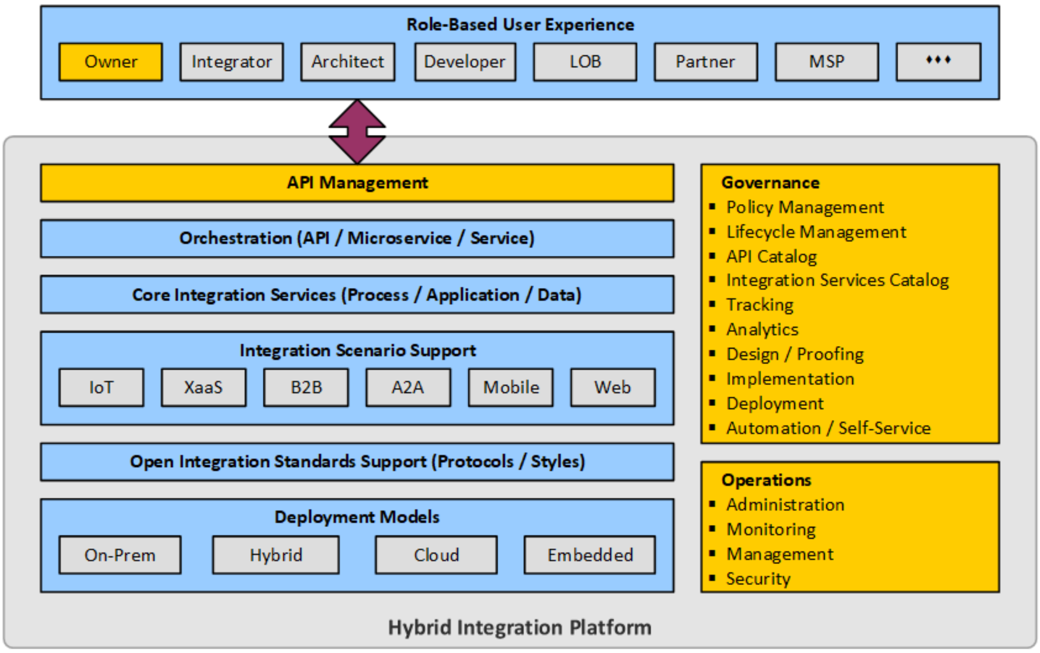


Figure 35: Hybrid Integration Platform (HIP) Architecture

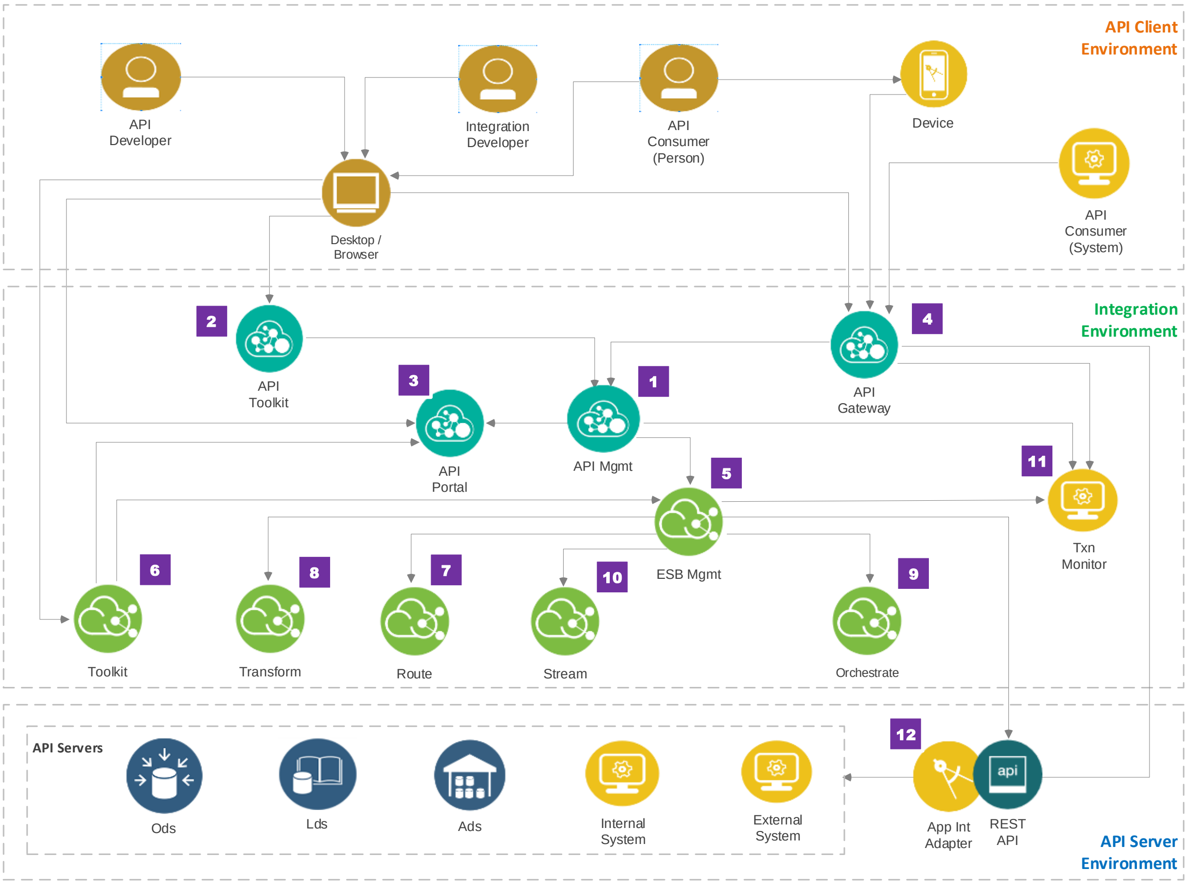


Figure 36: HIP-Influenced Reference Integration Architecture

A well-designed HIP should span and support the following important integration dimensions:

- Personas (Roles): Role-based experiences for various internal and external consumers and developers of integration services including architects, integration specialists, application developers, partners, and various lines-of-business (LOBs).
- Integration styles, domains, and use-cases: Integration capabilities that support the different integration styles (synchronous and asynchronous file transfer, data replication, RPC, and messaging) across different integration domains (interaction, process, application, data, business-to-business) using open protocols. Key use-cases include the data pipeline architecture (DPA) described in the prequel.
- Deployment models: Cloud (potentially across multiple cloud service providers), on-premises, hybrid (cloud and on-premises), mobile, Web, and embedded IoT.

Figure 35 depicts a component, service, and role arrangement that is representative of an HIP-based integration architecture, with the component roles summarised in Table 12.

Table 12: Integration Architecture: Components & Services

1	API Management	Manages the operations of the various servers in the API management platform, provides analytics about APIs and API users, and enables general administration of the platform.
2	API Toolkit	Development environment which is used primarily by API developers to create APIs and define API exposure specifications.
3	API Portal	Enables API providers to build a customized developer portal for application and integration developers to access published APIs so that they can be incorporated into application logic and integration flows.
4	API Gateway	Processes and manages security and protocols and stores relevant user and appliance authentication data; gateway servers can also enable APIs to integrate with various endpoints, but this function is typically delegated to the ESB.
5	ESB Management	Manages the operations of the various servers in the ESB platform, provides analytics about integration service clients, and serves as a registry for integration flow specifications.
6	ESB Toolkit	Development environment which is used primarily by integration developers to create integration flows incorporating various endpoints, including APIs.
7	Esb Routing	Message routing is a fundamental requirement when integrating applications or services; it involves
8	Esb Transformation	Message transformation is another basic requirement when integrating applications or services.
9	Esb Orchestration	Orchestration enables access to multiple fine-grained transactions using a single service which encapsulates and invokes the fine-grained services within single coarse-grained process flow.
10	Esb Streaming	Streaming capabilities enable treatment of all data sources as streams of data to which stream processing operations can be applied, and outputs published to one or more destinations.
11	Integration Monitor	Monitors all transactions including sensing and notifying of issues such as exceptions and performance bottlenecks, as well as administration of the integration environment and connected data repositories.
12	Application Adapter	Publishes REST API and implements the binding between the API and server application services; this enables client applications to expose their functionality via APIs.

4.6 Technology Architecture

The technology architecture addresses the structure and interaction of infrastructure-related services, namely compute (processor and memory), storage, and network interfaces, in the form of logical and physical technology components. The technology architecture enables the information components (data and application components), which in turn enable the business capabilities. Key design considerations include location (on-premise, cloud, or hybrid), availability (high, continuous), and scalability.

The GEA-ET technology architecture will be developed in alignment with the infrastructure modernisation project currently being undertaken World Bank Digital Foundation.

4.7 Security Architecture

The security domain encompasses all the planning, measures, and controls used to protect the confidentiality, integrity, and availability of information systems and services. The GEA-ET security architecture focuses on protection of all system components at the points of interaction, transportation, and storage, in the expectation that these actions will, in turn, protect the information that is 'touched' by those components.

4.7.1 Cyber-Security Framework

The security architecture is based on the NIST cybersecurity framework (CSF), which specifies CSF functions and categories, and the control components or capabilities that should be deployed to support each category (i.e., processes, technologies, skills, etc.).

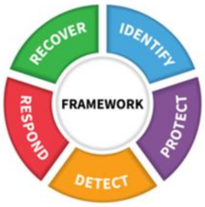


Figure 37: NIST Cyber-Security Framework

That is, the five high level functions - identify, protect, detect, respond, and recover - are segmented into categories which are designed to cover the breadth of cybersecurity objectives for an organization, and topics that span the cyber, physical, and personnel domains. Subcategories elaborate the category space with outcome-driven statements that provide considerations for creating or improving a cybersecurity program.

- Identify: Assists in developing an organizational understanding to managing cybersecurity risk to systems, people, assets, data, and capabilities. Understanding the business context, the resources that support critical functions, and the related cybersecurity risks enables an organization to focus and prioritize its efforts, consistent with its risk management strategy and business needs.
- Protect: Outlines appropriate safeguards to ensure delivery of critical infrastructure services by supporting the ability to limit or contain the impact of a potential cybersecurity event, and thereby ensuring the ongoing achievement of business objectives.
- Detect: Defines the appropriate activities to identify the occurrence of a cybersecurity incidents and events in a timely and verifiable fashion.
- Respond: Includes appropriate activities to take action regarding a detected cybersecurity incident by supporting the ability to contain the impact of a potential cybersecurity incident.

- Recover: Specifies appropriate activities to maintain plans for resilience and to restore any capabilities or services that are impaired due to a cybersecurity incident, by supporting timely recovery to normal operations to reduce the impact from a cybersecurity incident.

It is notable that the NIST framework is outcome driven and does not mandate how an organization must achieve those outcomes, therefore it enables risk-based implementations that are customized to an organization's needs. For ease of reference, Table 13 summarises elements of the NIST CSF that are relevant for GEA-ET.

Table 13: NIST Cyber-Security Framework Categories

Function	Category	Description
Identify	Asset Management	Identifying physical assets (devices and systems), software, communication workflows, external information systems, prioritized resources, and roles relating to cybersecurity to establish the basis of an asset management program.
	Business Environment	Identifying the Business Environment, the organization supports including the organization's role in the service chain, and its place in the critical infrastructure sector.
	Governance	Identifying cybersecurity policies established within the organization to define the Governance program as well as identifying legal and regulatory requirements regarding the cybersecurity capabilities of the organization.
	Risk Assessment	Identifying asset vulnerabilities, threats to internal and external organizational resources, and risk response activities as a basis for risk assessment.
	Risk Management Strategy	Defining a risk management strategy including establishing risk tolerances and using it to support critical business decisions.
	Supply Chain Risk Management	Identifying a risk management strategy for the supply chain including priorities, constraints, risk tolerances, and assumptions related to supply chain risk.
	Identity Management & Access Control	Protections for identity management and access control for device and user identities and credentials, including physical and remote access authentication, authorisation, and audit.
	Awareness and Training	Empower staff through cybersecurity training and awareness to perform their responsibilities in alignment with information security compliance policies and procedures.
Protect	Data Security	Establishing Data Security protection consistent with the organization’s risk strategy to protect the confidentiality, integrity, and availability of information
	Information Protection Processes	Implementing Information Protection Processes and Procedures to maintain and manage the protections of information systems and assets
	Maintenance	Protecting organizational resources through Maintenance, including remote maintenance, activities
	Protective Technology	Managing Protective Technology to ensure the security and resilience of systems and assets are consistent with organizational policies, procedures, and agreements
Detect	Anomalies and Events	Ensuring Anomalies and Events are detected, and their potential impact is understood
	Security Continuous Monitoring	Implementing Security Continuous Monitoring capabilities to monitor cybersecurity events and verify the effectiveness of protective measures including network and physical activities.
	Detection Processes	Maintaining detection processes to provide awareness of anomalous events.
Respond	Response Planning	Ensuring response planning process are executed during and after an incident
	Communications	Managing communications during and after an event with stakeholders, law enforcement, external stakeholders as appropriate.

	Analysis	Analysis is conducted to ensure effective response and support recovery activities including forensic analysis and determining the impact of incidents.
	Mitigation	Mitigation activities are performed to prevent expansion of an event and to resolve the incident.
	Improvements	The organization implements Improvements by incorporating lessons learned from current and previous detection / response activities
Recover	Recovery Planning	Ensuring the organization implements Recovery Planning processes and procedures to restore systems and/or assets affected by cybersecurity incidents
	Improvements	Implementing Improvements based on lessons learned and reviews of existing strategies.
	Communications	Internal and external communications are coordinated during and following the recovery from a cybersecurity incident.

Whilst the NIST CSF provides breadth and depth of coverage, the GEA-ET security framework focuses on the protection, detection, and response elements for two main reasons: (a) some categories are less applicable to cybersecurity risk management, than general risk management which has a broader enterprise scope, and (b) most of the identification and recovery elements are not defined by the GEA-ET but by other competencies within government (e.g., INSA, ETC) - however, they are recognised and integrated into the ESA.

It is notable the changing threat landscape features increased breaches of network boundaries, and this has prompted NIST and similar organisations to develop more advanced cybersecurity frameworks that focus on adaptive internal defence strategies (in addition to traditional perimeter protection techniques), including:

- Application Segmentation: Application segmentation is the practice of implementing Layer 4 controls that can both isolate an application's distinct service tiers from one another and create a security boundary around the complete application to reduce its exposure to attacks originating from other applications.
- Zero-Trust Architectures: A security concept centred on the belief that organizations should not automatically trust anything *inside* or *outside* its perimeters and instead must verify anything and everything trying to connect to its systems before granting access. In other words, adopt a 'deny' instead of 'allow' option as the default position for control of access to system resources, which means explicitly add permissions to a zero-access baseline, rather than remove permissions from a full access baseline.
- Cyber-Resilient System Engineering: A NIST project that focuses on using of system engineering techniques in the cybersecurity domain, to develop more survivable and trustworthy systems [@].

These advanced security techniques are not proposed for assessment or implementation within the current GEA-ET roadmap because it is felt that they require greater levels of cybersecurity maturity to be affective.

The GEA-ET Security Architecture covers the basic capabilities required to secure the confidentiality, integrity, and availability (primarily) of the WOG information systems, and the expectation is that most these capabilities can be delivered using assets that are either planned or are already available for reuse.

In addition, the benefits of using of external 'active' security service providers should be investigated as a possible option for Ethiopia, as a mechanism to protect government assets in the short-term whilst improved security capabilities are still under development. This practice is increasingly being adopted by governments and other organisations worldwide, as even well-resourced entities struggle to maintain adequate information security postures in the face of the ever-changing threat landscape.

4.7.2 Reference Architecture

Based on these security considerations, illustrates a representative CSF-influenced security architecture that aligns with the other reference architectures presented in the prequel, in the sense that it is designed to protect specific components of those architectures. Table lists the CSF functions and categories, with the control components or capabilities that should be deployed to support each category (i.e., processes, technologies, skills, etc.), and it outlines the proposed mitigating solutions and/or current solution which may either be deployed or planned.

Note that even though multiple parties are typically involved in implementing each risk mitigator, it should be possible to deduce the party with the primary responsibility from the architecture diagram, since it specifies the mitigator deployment platform.

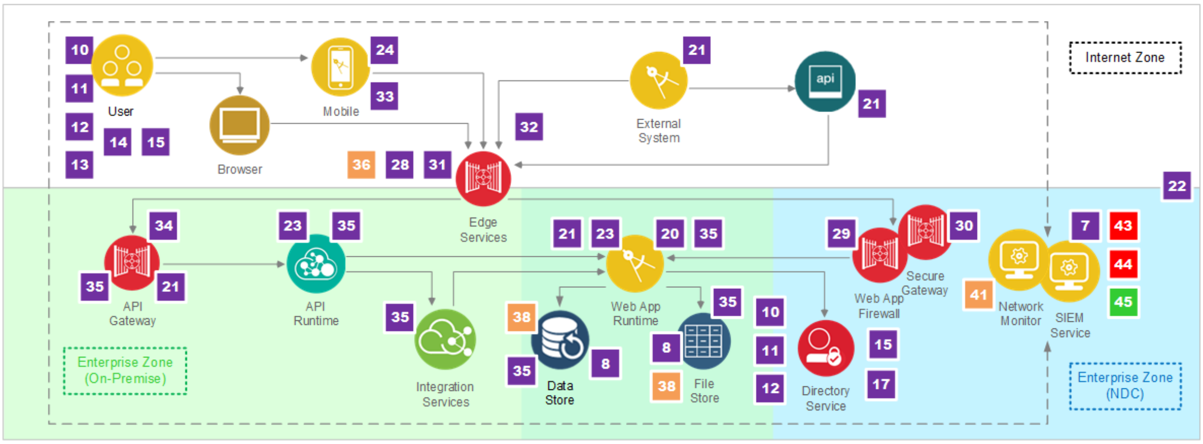


Figure 38: NIST-Influenced Reference Security Architecture

Table 14: NIST Cyber-Security Framework Categories

#	Function	Category	Control/Subcategory
[1]	Identify	Asset Management	Asset identity: Identify all physical and software assets to understand what needs to be protected.

Table 14: NIST Cyber-Security Framework Categories

#	Function	Category	Control/Subcategory
[2]	Identify	Asset Management	Application identity: Application whitelisting: Specify an index of approved software installations to protect endpoints from potentially harmful applications.
[3]	Identify	Information PPP	System backup: System operation assures backup of all critical components, along a day-to-year time spectrum based on business needs.
[4]	Identify	Information PPP	System resilience: Regular testing of recovery processes from local-, site-, or disaster-level outages to ensure system resilience at all times.
[5]	Identify	Risk Mgmt Strategy	Application availability: Application design assures availability along an agreed availability spectrum based on business needs.
[6]	Identify	Risk Mgmt Strategy	Infrastructure availability: System design assures availability along an agreed availability spectrum, to address risks from component-level outages to widespread disasters.
[7]	Protect	Awareness & Training	System: Threat intelligence feeds: including malicious URLs, malware hashes, and attack-associated email and IP addresses.
[8]	Protect	Data Security	Data integrity: Data encryption technology to protect all data at rest regardless of location.
[9]	Protect	Data Security	Data integrity: Data encryption technology to protect all data in transit regardless of channel.
[10]	Protect	Identity & Access Mgmt (IAM)	User identity: Single-sign-on (SSO) to enable user to log in to multiple systems once and access services without having to re-submit authentication factors.
[11]	Protect	Identity & Access Mgmt (IAM)	User identity: Multi-factor authentication (MFA) with traditional username/password AND other authentication factors (e.g. mobile device, email, IP address, device ID, third-party identity providers).
[12]	Protect	Identity & Access Mgmt (IAM)	User identity: Managed user directory services to host and control user credential profiles.
[13]	Protect	Identity & Access Mgmt (IAM)	User identity: Public key infrastructure (PKI) services to distribute digital identities to users and managed endpoints.
[14]	Protect	Identity & Access Mgmt (IAM)	User identity: 3rd party authentication services for external users.
[15]	Protect	Identity & Access Mgmt (IAM)	User identity: Access control: Manage user permissions on resources to understand allowed user access and potential security risks.
[16]	Protect	Identity & Access Mgmt (IAM)	User identity: Password management policies that minimise the likelihood of username/password compromise.
[17]	Protect	Identity & Access Mgmt (IAM)	Privileged user access: Secure privileges for service, application, root, and administrator accounts across the enterprise.
[18]	Protect	Identity & Access Mgmt (IAM)	Privileged user delegation/elevation: Manage local administrative rights at endpoints to prevent hijacks of privileged accounts.
[19]	Protect	Identity & Access Mgmt (IAM)	Privileged user monitoring: Monitor privileged user account activity, including access, delegation, and elevation.
[20]	Protect	Identity & Access Mgmt (IAM)	Application: PKI-based authentication for client applications.
[21]	Protect	Identity & Access Mgmt (IAM)	API: Flexibly secure communications by authenticating client and server identities at both ends of the API channel.
[22]	Protect	Information PPP	Network isolation: Partition network into firewall-separated zones to isolate threats.
[23]	Protect	Information PPP	Application integrity: All web-facing applications hardened by thorough vulnerability scanning.
[24]	Protect	Information PPP	Mobile application: Mobile device management (MDM) to secure mobile devices regardless of platform type and form factor.

Table 14: NIST Cyber-Security Framework Categories

#	Function	Category	Control/Subcategory
[25]	Protect	Information PPP	Application: All web-facing applications positioned downstream of edge security services within the DMZ, including social media platforms.
[26]	Protect	Information PPP	Application: Logical (and where necessary physical) isolation of development and operational environments.
[27]	Protect	Maintenance	System currency: Regular infrastructure and application maintenance schedule.
[28]	Protect	Protective Technology	Network perimeter (edge): Network access control.
[29]	Protect	Protective Technology	Network perimeter (edge): Edge security services.
[30]	Protect	Protective Technology	User: Protects users from web-based threats in addition to applying and enforcing corporate acceptable use policies.
[31]	Protect	Protective Technology	Network perimeter (edge): Repel incoming threats at the edge which are typically embedded in malicious packets.
[32]	Protect	Protective Technology	Network: Remote access to internal network to be made via secure virtual private networks (VPNs).
[33]	Protect	Protective Technology	Wireless network: Strong user and device authentication and indirect connection to internal networks via edge protection services.
[34]	Protect	Protective Technology	API: Protect against threats that specifically target API channels such as DDOS, malware injection, and cross-site scripting.
[35]	Protect	Protective Technology	End point: End point-level (as opposed to gateway-level) protection on both servers and end-user devices.
[36]	Detect	Anomalies & Events	Network perimeter (edge): Monitor and analyse post-edge network traffic for violations of system security policies.
[37]	Detect	Detection Processes	Network: Static IP addressing with structured network node identifiers.
[38]	Detect	Detection Processes	Data integrity: Detect anomalies in modifications to data records regardless of storage mode (e.g., file or database).
[39]	Detect	Detection Processes	Email integrity: Detect unauthorized access, loss or compromise via email.
[40]	Detect	Detection Processes	Email integrity: Leak detection to prevent unauthorised egress of confidential information via email.
[41]	Detect	Continuous Monitoring	Network: Network monitoring and analysis across all network nodes.
[42]	Detect	Continuous Monitoring	System: Security incident and event management (SIEM)
[43]	Detect	Continuous Monitoring	System status: Audit and compliance processes to validate implemented controls against security policy, industry compliance, and risk policies.
[44]	Respond	Response Planning	System: Documented alerting and communication processes for responding to detected security events or incidents.
[45]	Respond	Response Planning	System: Resolve vulnerabilities identified from any source, including proactive security posture assessments and reactive protective technologies.
[46]	Recover	Recovery Planning	System: Incident recovery planning processes and procedures to restore systems and/or assets affected by cybersecurity incidents.
[47]	Recover	Recovery Planning	System: Disaster planning processes and procedures to restore systems and/or assets affected by disaster-scale events.

4.8 DevOps Architecture (Development)

Development refers to the solution delivery lifecycle (SDLC) which may be thought of as a structured methodology used to govern the development of an IT system from inception to delivery and operation. The aim of an SDLC is to enable effective production of high-quality solutions that meet or exceed user expectations at all times, within agreed cost, time, and quality constraints.

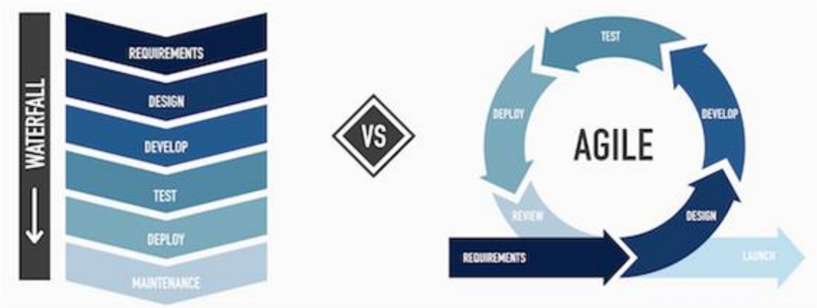


Figure 39: Waterfall vs. Agile SDLC Models

There are principally two SDLC approaches - *Waterfall* (or traditional) and *Agile* - each with its own sets of principles, drivers, and techniques, and therefore differences in the way development structures and processes are organised (see Table 15). One of their characterising divergences is the phasing of development activities:

Table 15: SDLC Feature List: Waterfall vs. Agile Approaches

Waterfall	Agile
Sequential development process in defined phases with governed hand-over of control between phases	Iterative development in short sprints which improves responsiveness to changing environment dimensions
Fixed, documented methodology agreed in advance	Flexible and adaptive methodology
Limited and delayed business feedback	Rapid business feedback to guide future development
Requirements and scope are agreed and fixed at inception	Change is anticipated at any point in the cycle
Infrequent, project-focused communication	Continuous, user-focused collaboration/communication
Scheduled, sequential SDLC phases with no overlap.	Early start ('shift-left') and overlap of SDLC phases.
Fixed hierarchy with fixed individual responsibilities	Flexible hierarchy and interchangeable team roles
Individual responsibility for each phase-bound deliverable	Joint team responsibility for all end deliverables
Adapted from: https://www.bmc.com/blogs/agile-vs-waterfall/#	

- *Waterfall* has a single development cycle in which phased activities are executed sequentially (with governed hand-over between phases) and solution acceptance occurs at the end of the cycle;
- On the other hand, *Agile* slices the solution into smaller deliverables which are completed within smaller time-boxed cycles, with 'usable value' being delivered and accepted within the time-box.

Waterfall and *Agile* both have strengths and weaknesses that can be inferred from the feature list, but there is general consensus that the nature of the solution and enterprise must be considered when deciding on SDLC approach: *Agile* is not suitable in all circumstances, and neither is *Waterfall*. *Agile* tends to be more effective for governing short projects in high-risk scenarios, whereas *Waterfall* is more suited to projects that take place in stable, predictable business environments.

For this reason, variants have emerged (and continue to emerge) that augment or integrate aspects of these methodologies, such as *DevOps* which focuses on closing the gap between the IT development (Dev) and IT operations (Ops) functions, Adaptive Software Development (ASD) which advocates flexible method selection to fit the task at hand and available resources, Scaled Agile Framework (SAFe) which scales agile methods and integrates some Waterfall techniques, and numerous others.

For the common components in the WOG tier, GEA-ET recommends the SAFe approach since it is based on agile thinking, but it integrates Waterfall-style structural and procedural disciplines; this makes it suitable for guiding major transitional initiatives like GEA-ET development and adoption. Furthermore, the expectation is that the skills required to implement this approach will be assembled with adequate capacity to support the prioritised WOG tier initiatives.

On the other hand, GEA-ET recommends against mandating a specific SDLC approach at the MDA tier level since efforts to support it may dilute MDA focus on higher priority concerns like MDA application alignment with the business, integration, and data architectures to support interoperability. The rationale for this strategy is the service-orientation principle, which directs that usage of a service does not require that its implementation be known by the service consumer.

4.9 DevOps Architecture (Operations)

The operations domain is also known as system management, service management or more formally as IT service management (ITSM). In traditional IT organisations, the operations function assumes responsibility for deploying and managing IT assets that are constructed by the development function, but the modern "DevOps" approach demands closer coupling and collaboration across development and operations competencies.

4.9.1 Operations Framework

The GEA-ET proposes an operations framework that leverages on the Information Technology Infrastructure Library (ITIL), a globally recognized standard that provides comprehensive, practical and proven guidance for establishing and maintaining a management system for IT-enabled services.

ITIL is framed around a service value system (SVS) which consists of guiding principles, governance, service value chain, practices, and continual improvement, along with a four-dimensional ITSM model that encompasses organisations and people, information and technology, partners and suppliers, and value streams and processes.

Table 16: ITIL Segments & Capabilities

#	Segment	Practice	Description
1	General	Architecture mgmt	Provides a view of all the different organisational components and how they interface and interrelate to enable the organization to achieve its objectives, continuously; architectural domains include business, information, technology, integration, security, and operations.
2	General	Continual improvement	To ensure continuous alignment of service and practice of an organization, by identifying and improving them on a continual basis. [NIST CSF: Recover/Improvements]
3	General	Information security mgmt	Protection of information assets by ensuring the confidentiality, integrity, and availability of information, so that information is safeguarded from unauthorized access and misuse. [NIST CSF]
4	Service	Availability mgmt	Ensuring that service availability meets the needs of the organisation, which means the service is available when needed.
5	Service	Capacity and performance mgmt	Ensures that sufficient capacity is available to the services and that service performs at the level expected and achieves the objectives demanded by the services in a cost-effective way.
6	Service	Change control	Change refers to adding, moving, modifying, improving, removing, etc., the capabilities of services & service components, including hardware, software, process, products, documents etc. which are used to compose a service; change control applies proper assessment, analysis, and authorization of changes.
7	Service	Incident mgmt	Ensures restoration of services to normal working conditions by resolving and restoring the services during the incidents, and minimizing the impact to business, which occurs due to the incidents. [NIST CSF: Respond: Response/Recovery Planning]
8	Service	Asset mgmt	Plans and manages the entire lifecycle of IT assets, including maximising value, controlling costs, and managing risks, as well as enabling decisions related to acquisition, operation, and retirement of assets. [NIST CSF: Identify: Asset Management]

Table 16: ITIL Segments & Capabilities

#	Segment	Practice	Description
9	Service	Monitoring and event mgmt	Ensures that services are observed systematically, to detect and notify any changes that significantly affect service status and performance. [NIST CSF: Identify: Detect: Detection Processes]
10	Service	Problem mgmt	Identifies the potential & actual causes of incidents and reduce the probability of the impacts of incidents by providing the solutions and workarounds, including the creation of known errors. [NIST CSF: Identify: Respond: Analysis/Mitigation]
11	Service	Release mgmt	Ensures that the new or changed services and their features are available to use.
12	Service	Service catalogue mgmt	Provides the consistent single source of information for all the services and service offerings, which is made available to authorised users.
13	Service	Service configuration mgmt	Ensures the availability of the information related to service configuration and the configuration items (CI) which are used to compose services; CI refers to all the software, hardware, people, documents, facilities, etc. which are used to compose a service.
14	Service	Service continuity mgmt	Ensures the availability of minimum services at a sufficient level for business to sustain in the event of disaster-scale incidents.
15	Service	Service design	Designs services to address the needs of both the service consumer and service provider in a cost-effective way.
16	Service	Service desk	Provides a single point for user interaction on issues, requests, queries, and suggestions; these interactions (aka tickets or service requests) are typically acknowledged, logged, classified, prioritised, actioned, and tracked through to resolution. [NIST CSF: Respond: Response Planning]
17	Service	Service level mgmt	Specifies and agrees clear service targets, so that service status and performance can be monitored and managed against those targets throughout the service lifecycle.
18	Service	Service request mgmt	Responds to user-initiated requests which are usually part of the standard or pre-defined sets of services available to users; such requests are typically handled by the service desk practice.
19	Service	Service validation and testing	Ensures that new or changed services and products are validated, so that they meet the defined and agreed service levels.
20	Technical	Deployment mgmt	Manages introduction of new or changed processes, software, hardware, documentation, or any service component production environments; must be closely coordinated with release management and change control while introducing a change to an environment.
21	Technical	Infrastructure and platform mgmt	Enables the monitoring and managing the technology solutions within an organisation, including compute, network, storage, middleware, operating system components
22	Technical	Software development and Mgmt	Ensures that software applications are fit for end-user consumption by managing entire software lifecycle from ideation through to retirement.

Against this framework, ITIL defines management practices for services, projects, products, design, transition, build, test, delivery, support, and the like, which are segmented into the following three parts:

- General Management: Applicable across the enterprise for the success of business and services provided by the organization.
- Service Management: Applicable for specific services being developed, deployed, delivered and supported in an organization environment.

- Technical Management: Adapted from technology management domains for service management purposes by expanding or shifting focus from technology solutions to IT services.

The ITIL model scope is comprehensive, and it extends to general management practices that are typically not defined as part of an ESA, so the GEA-ET operations architecture focuses on service and technical management practices that are more directly related to the ongoing creation and operation of IT assets.

Furthermore, to address the significant overlap between ITIL management practices and the NIST functions and categories, it is useful thinking of the operations elements as abstractions of the corresponding security elements, and the operations architecture as the execution platform for all the controls specified by the security architecture. A high-level mapping between these NIST and ITIL constructs is included in Table 16.

4.9.2 Reference Architecture

A key GEA-ET diagnostic finding was that no mechanism exists for centrally monitoring and tracking the end-to-end availability and performance characteristics processes and components on key systems. This suggests the absence of a service management platform that integrates the tools used to manage individual services.

Since little is known about current service management practices at both the WOG and MDA tier levels, the recommendation to start development of the operations architecture is that a more detailed assessment be conducted to understand current practices and determine future needs, using the ITIL capability listing of Table 16 as a guideline.

4.10 Resilience Architecture

System resilience refers to an organisation's ability to quickly adapt to disruptions while maintaining continuous business operations and safeguarding its assets; one of the most important of these assets consists in the information systems (systems) that process and store the data that is the lifeblood of many entities. Therefore, organisations can only be resilient if the information systems on which they rely are also resilient.

4.10.1 Risk Continuum

Resilience design is a risk management exercise in that risks or threats which can impact information systems (should they occur) need to be managed to retain specified service levels, so it makes sense to understand that nature of these risks and service levels.

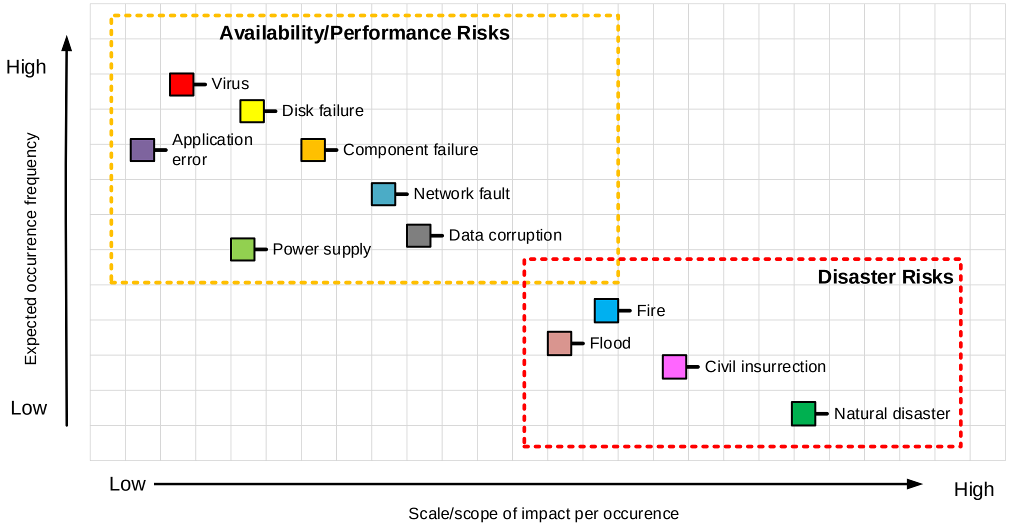


Figure 40: Resilience Risk Continuum

System risks can be classified in terms of expected occurrence frequency, and the scale and scope of the impact, as shown on an abstract scale in the above diagram. Note that this is only an indicative sample of potential risks and ratings and that these need to be assessed on a case-by-case basis. Disruptions can range, for example, from network faults that may have minimal impact and only result in a degradation in system performance, to natural disasters that can destroy an entire site (or data centre).

4.10.2 Resilience Objectives

Availability and performance are common measures of resilience that can be achieved using various architectural patterns including redundancy (aka replication), auto-scaling, immutability, and repeatability. Availability objectives classify availability options according to service availability levels (continuous or interruptible) and predictability of service interruptions (planned or unplanned).

These concepts are illustrated using an abstraction of the layered SOA model, which shows redundant nodes, layers and service interfaces, as well as the different availability objectives that may be placed on the components (nodes) within each service layer.

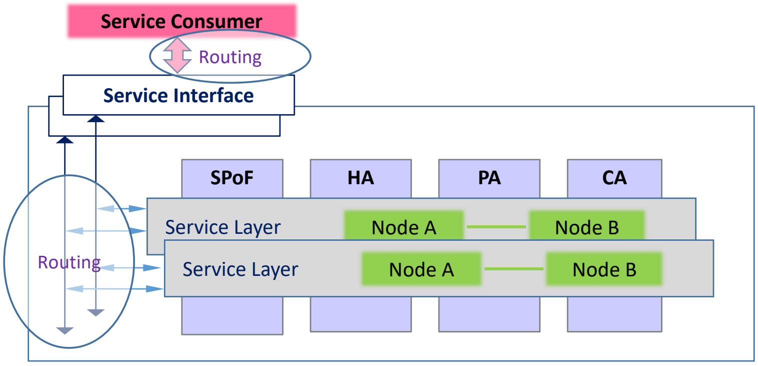


Figure 41: Resilience Objective Continuum

Table 17: Resilience Categories

Category	Description
Continuous Availability (CA)	The ability to continuously provide services whilst masking service consumers from planned or unplanned downtime, regardless of the scope and scale of any disruption (within reason). CA will usually be the most complex (and costly) option especially when deployed with stateful services since state must be replicated across all instances to ensure continuity.
Planned Availability (PA)	Describes the ability to continuously provide services whilst masking service consumers from planned downtime, which is typically scheduled to facilitate maintenance or change deployments and a key objective is to minimise downtime since system performance will be less than optimal during the interruption.
High Availability (HA)	The ability to provides services during defined periods, at specified levels, whilst masking service consumers from unplanned downtime.
Single-Point-of-Failure (SPoF)	Refers to any component that can cause downtime, and for which a specific counter-measure has not been implemented; this can be a valid availability objective for non-critical services.

At finer granularity levels, resilience objectives can be more precisely expressed in terms of these parameters:

- Uptime: Uptime is the percentage of time that a system is fully operational, usually measured as a percentage so that a system with 99.999% uptime has an expected downtime of less than 6 minutes in total per year. The desired uptime greatly influences risk coverage the choice of resilience patterns that need to be deployed.
- Recovery Time Objective (RTO): An RTO represents the amount of time an application can be down and not result in significant degradation to an organisation plus the time that it takes for the system to go from failure to recovery.; this recovery process includes the steps that must be taken to return the application and its data to its pre-disaster state. Implementing RTOs requires that applications first be sorted based on their priority and risk of loss, followed by allocation of resilience-enabling resources based on ranking.

- Recovery Point Objective (RPO): An RPO is a measurement of time from the failure, disaster or comparable loss-causing event, and it measures back in time to when your data was preserved in a usable format, usually to the most recent backup. Recovery processing usually preserves any data changes made before the disaster or failure. RPOs can also refer to how much data can be lost before your enterprise receives significant harm, also known as your enterprise's loss tolerance.

It is relatively straightforward to state the desired parameter levels, but it is significantly more difficult to assure that a given solution design can meet a specific resilience objective because of factors such as workload deployment across different environments and inter-dependencies between technology stack layers.

For these reasons, the resilience architecture initially specifies application resilience objectives in terms of resilience categories, but these will be refined into more precise parameters during the detailed design when the expectation is that application design parameters will be better understood.

4.10.3 Resilience Design Patterns

The redundancy pattern may be thought of a specific case of auto-scaling which can be extended into multi-node auto-scaling topologies known as clusters, availability groups, or swarms. Furthermore, nodes can have an active-active configuration in which both nodes are normally active, or an active-passive configuration in which one node is active and the other is passive during normal operations. Active-active configurations tend to be more effective at masking downtime, since active-passive configurations can interrupt services while operational control is switched from an active to a passive node.

Table 18: Resilience Design Patterns

Pattern	Description
Redundancy (Replication)	Redundancy means having secondary (usually replica) node available to back-up a primary node, which can continue to process computing workloads if the primary node fails; a switch from a primary to secondary node is said to be a failover, and the reverse action a failback.
Auto-scaling	Auto-scaling computing services such as servers or virtual machines to adjust their capacity up or down automatically, so that service capacity or performance is maintained, based on defined situations such as traffic or utilization levels.
Immutable Infrastructure	The immutability principle mandates that immutable infrastructure be replaced for every deployment, rather than being updated in place - this reduces configuration drift and ensures repeatable deployments anywhere from source.
Infrastructure-as-Code (IaC)	The management of infrastructure (networks, virtual machines, load routers, and connection topology) in a descriptive model, typically using the same version-controlled repository used for source code.
Stateless Services	Stateless services treat all client requests independently of prior requests or sessions, and do not store any information locally, so that any request can be handled by any available service instance; statelessness is a prerequisite for auto-scaling and immutable infrastructure.

On the other hand, active-active setups tend to consume more bandwidth because a common state needs to be shared between node replicas, typically utilising synchronous replication protocols, whereas asynchronous replication mechanisms that consume less bandwidth are often adequate for active-passive arrangements.

A critical enabler of the redundancy pattern is a routing capability (aka load balancer or “sprayer”) which has can use node availability status [ideally transparently] route client requests to available nodes in a balanced fashion (or reroute workloads in the case of node failure).

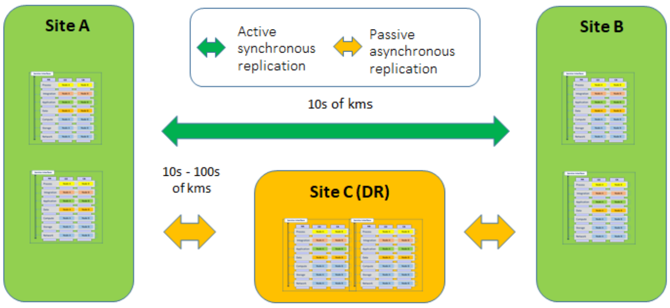

Site A
Active synchronous replication
Passive asynchronous replication
Site B
10s of kms
10s - 100s of kms
Site C (DR)

Figure 42: Disaster Mitigation Options

System design must also consider the potential scope and scale of the impact that a risk occurrence can create, as this can also influence the risk mitigation approach. As an example, whilst the ability to ‘hot-swap’ might mitigate server component failures, and thereby provide high availability, it will not help if no counter-measures are in place to address power supply failures at a site level. As another example, the geographic proximity of service providers must be considered if natural disasters are within the scope of the risks to be mitigated.

Figure 42 illustrates a classic approach to addressing this problem – which entails deploying one or more active sites (A and B) and a secondary (disaster recovery) site and ensuring that their dispersion is adequate to avoid the impact of geographically widespread natural disasters.

In this scenario, sites A and B are highly available configurations that make up a continuously available cluster whilst site C mitigates disaster impact; many variations exist that leverage this general resilience theme. Single point-of-failure can also be a valid SLC for non-critical components that can be ‘repaired’ in a reactive fashion, as and when they experience disruptive incidents.

4.10.4 Cloud Practices

Resilient system implementation typically demands deep development skills to design and configure services that support the resilience architectural patterns and availability objective configurations, as well as deep operations skills to deploy manage those services; this is especially true for on-premise deployments. On the other hand, commercial cloud service providers normally achieve high uptimes which are combined with a streamlined availability option selection/configuration process.

For example, AWS provides availability zones (AZs) which are isolated locations within data centre regions from which public cloud services originate and operate; regions are geographic locations in which public cloud service providers' data centres reside; organisations choose one or multiple worldwide availability zones for their services depending on business needs; cloud administrators can also choose to replicate services across multiple availability zones to decrease latency or to protect resources, and they can move resources to another availability zone in the event of an outage. These are some of the major benefits of cloud computing which are very difficult to replicate in a cost-effective way, even with distributed on-premise deployments.

An emerging trend among governments is partnering with established providers of public cloud services to implement private government clouds on the same platforms and technologies used in public clouds. All the major providers have private cloud platforms that can be deployed locally (in-country), including AWS, Azure, IBM and Google, as well as many other lesser-known providers. This approach should be seriously explored since represents a tangible mechanism for modernising Ethiopia’s technology infrastructure along resilience and other cloud-enabled dimensions.

4.10.5 Reference Architecture

This resilience framework can be used to guide the selection of resilience options for business services and system components that implement them. Note that because the framework is outcome driven and does not mandate how those outcomes are achieved, it enables need-based implementations that are customized for the operational environment. This approach is supported by the fact that most platforms and technologies provide multiple design approaches for achieving specific resilience objectives.

Therefore, GEA-ET does not make a specific recommendation around component-level resilience architectures as these will be determined at detailed solution design time using this resilience framework as a guide. Due the pervasive influence of cloud technologies on all aspects of computing, it is recommended that a cloud computing strategy be developed for Ethiopia, to guide decisions around resilience and technology designs at all levels of government.

5 Principles

As stated in the TOGAF standard (ADM Part III: Guidelines and Techniques) [XXX]:

> *Principles are general rules and guidelines, intended to be enduring and seldom amended, that inform and support the way in which an organisation sets about fulfilling its mission. In their turn, principles may be just one element in a structured set of ideas that collectively define and guide the organisation, from values through to actions and results.*

GEA-ET is influenced by two categories of principles: enterprise principles which provide a basis for decision-making throughout an enterprise and inform how the organization sets about fulfilling its mission, and architectural principles, that guide the development and deployment of architectures.

This specification focuses on architecture principles which build upon previous work:

- [EA-2011]: This 2011 report characterises, adds an enterprise-level category called "IT Principles", and reproduces the TOGAF exemplars [XXX].
- [EA-2019-3]: This report builds on [EA-2011-Pwc] to motivate and characterise architectural principles and link their scope and application to enterprise principles embedded in various policies adopted by the government (e.g., the FDRE Constitution, GTP-2, the UN SDGs, EoDB, and "home-grown economic reform" strategies). It also provides an extensive set of perspective-level architectural principles which translate the enterprise principles into architectural constructs that can enterprise architecture work.

Both reports draw on TOGAF principles which shall not be repeated here; rather, they can be accessed in the reports or directly in the TOGAF documentation.

Table 19: Architectural Principle Properties

Property	Description
Name	As short descriptor that reflects the essence of the rule and is designed to aid recall.
Statement	Succinctly and unambiguously communicates the fundamental rule.
Rationale	Highlights the benefits of adhering to the principle, and describes its relationship to other principles, if applicable.
Implications	Outlines the requirements, both for the business and IT, to execute the principle in terms of resources, costs, and activities/tasks.

Since most of the report content remain valid, it has been reviewed and reorganised into a spreadsheet which is easier to search and filter [@] and whose main columns are described in Table 19; the "Source" column identifies the source of the principle and the records the outcome of the review: Adopted (the rule was assimilated as-is with minor changes), Amended (the rule was changed to align with the GEA-ET strategy) or Dropped (discontinued because the rule is no longer relevant, is duplicated, or is not congruent with GEA-ET strategy).

Importantly, the GEA-ET adds several framework-level principles that were introduced in the GEA-ET Diagnostic Report, which are key influencers of the GEA-ET strategy and are applied to all its components – these principles all have a “GEAF” entry in the “Source” column.

The current list of +/-50 principles is relatively large since they need to be interpreted and applied when architectural decisions are made, and this has resource and cost implications. Therefore, a near-term action for the GEA-ET development team must be to engage stakeholders in a collaborative prioritisation effort that will identify the architectural principles whose application should be actively monitored; the remainder can be retained and propagated but their application will not be actively tracked.

6 Standards & Guidelines

This section describes standards & guidance that could, should, or must be used by GEA-ET stakeholders to support interoperability, digital services, and digital transformation for common business and technical capabilities across the FDRE. MDAs and specific government sectors may have additional standards and guidelines that apply.

It is recognised that successful digital government outcomes are dependent on more than just technology, so a digital standards catalogue should include categories across a broad range of subjects that relate directly to digital government. However, this catalogue focuses exclusively on business and technical standards that relate directly to interoperability, since this is the core capability required to support Whole-Of-Government services which have been prioritised in the GEA-ET strategy.

6.1 Standards, Guidelines & Procedures

As a preamble, it is worthwhile clarifying the role of policies, principles, standards, guidelines, and procedures, since the meaning of, and interaction between, these concepts tend to confuse GEA-ET developers and users alike. Within the GEA-ET context, these concepts can be interpreted as follows:

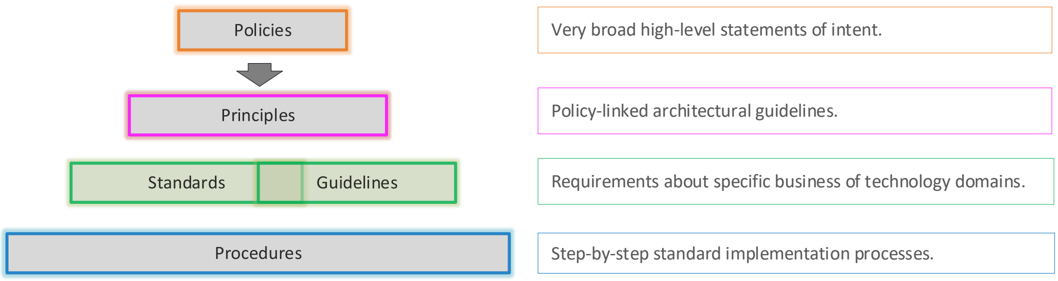

Policies
Very broad high-level statements of intent.
Principles
Policy-linked architectural guidelines.
Standards
Guidelines
Requirements about specific business of technology domains.
Procedures
Step-by-step standard implementation processes.

Figure 43: GEA-ET Guidance Pyramid

- Policy: Outlines the requirements or rules that must be met at the enterprise level, with scopes that tend to be broad, high level statements of intent. Policies are typically specified at an enterprise level and guided by enterprise mission and strategy statements, as well as applicable laws and regulations.
- Principle: A principle is a rule or guideline that derives from and underpins enterprise policies. The GEA-ET is concerned exclusively with architectural principles that inform and support the way in which the GEA-ET will be developed and deployed. Many architectural principles reference the specific enterprise policies that they enable or from which they are derived.
- Standard: A standard is a set of requirements that must be adhered to by all stakeholders, with a scope that tends to cover specifications related to a given business or technology domain.

- Guideline: A guideline is similar to a standard, but it differs in that unlike a standard, a guideline is merely a recommendation or suggestion that should be followed but is not necessarily required. Guidelines and standards are typically based on best practices curated by subject matter experts and they are frequently interchangeable.
- Procedure: A procedure defines the process that is followed to meet the requirements of a policy, standard, or guideline. The scope of a procedure is the specific step-by-step processes that should be followed for implementing a given standard or guideline.

Figure 43 helps to illustrate the relationship between these concepts in the 'GEA-ET Guidance Pyramid', bearing in mind that the GEA-ET only addresses principles, standards, and guidelines; policies are covered by the digital government strategy and procedures will be defined at the component implementation level. The pyramid reflects the fuzzy distinction between standards and guidelines, which means they are interchangeable and therefore presented together in the sequel.

As you go down the pyramid, the specifications get more detailed and are more subject to change. Thus, policies are broad and do not change often. Standards and guidelines are more detailed but more susceptible to change. Procedures are the most detailed and may change frequently as they incorporate new technologies, standards or practices.

6.2 Standards

[EA-2019-5] represents the only significant prior work that has been done on GEA-related technical standards for the FDRE, which focused on various classes of interoperability standard. It identifies a broad spectrum of interoperability-related standards (e.g., business process modelling, management protocols, etc.) which are classified into several technical areas (e.g., business engineering, enterprise service management, etc.).

Although augmentation and adoption of [EA-2019-5] was given serious consideration, a different approach was chosen for two main reasons: (1) the content is dated in that it omits important interoperability standards and includes deprecated standards, and (2) the catalogue format excludes relevant standards metadata. In addition, it was felt that the effort required to review and update [EA-2019-5] would exceed the effort needed to adapt and adopt an existing and current standards catalogue.

However, relevant standards proposed by [EA-2019-5] have been retained except those related to GEA-ET component internals as they violate the service-orientation principle. Notable exclusions include any standards that mandate specific technologies at any SOA layer, since they violate the GEA-ET portability and openness principles.

Therefore, the GEA-ET leverages the GEA-NZ digital government standards catalogue (DGSC), which provides a comprehensive listing of GEA-relevant business and technical standards, including those related to interoperability of common government services. Importantly it has a well-defined standards metadata set whose organisation facilitates mapping between standards and categories defined by GEA-ET reference models – such structuring enables solution rationalisation through analysis and comparison of business and technical standards and capabilities across MDAs.

Furthermore, the DGSC incorporates descriptions and references non-technical digital government standards, which can provide the FDRE with pointers as to what standards should be considered for development and adoption in the longer term, as part of the broader digital government programme. Another important feature of the DGSC is that it includes metadata to track the status of standards over time, as well as reasons for changes in that status.

As with architectural principles, GEA-ET standards are also presented in the form of a spreadsheet [@] with the metadata (properties) described in Table 20.

Table 20: Technical Standards Metadata

Property	Description
Domain	The first-level classification of the standard based on SOA layers and perspectives (see Figure 3).
Category	The second-level classification of the standard which is currently based on an unstructured taxonomy – the intent is to align this taxonomy with categories that will be defined by GEA-ET reference models, which are yet to be developed.
Name	The full official name or acronym for the standard and optionally reference, qualification, publication and version information, as applicable.
Description	A description of the standard which is derived from publicly available standard descriptions.
Link	A link to the official specification of the standard.
Status	The standard's catalogue status which can have the values listed in Table 21; these catalogue statuses are independent of the statuses set by standards bodies.
Status-Notes	Notes designed to aid in understanding any nuancing with respect to the standard status including recording any changes.
Predecessor	The name of the standard that preceded the standard record in this row.
Successor	The name of the standard that supersedes or replaces the standard record in this row.
Mandated-Org	The FRDE organisation (or role) that holds the mandate for disseminating and monitoring (or enforcing) the application of the standard; a "Self" entry indicates that no mandate has been issued so that GEA-ET implementors are responsible for applying the standard.
Jurisdiction	Displays jurisdiction is the specific jurisdiction that the standard applies to, for example: National (ET), Government (ET), State (Amhara).
Type	The content type of the standard catalogue record; the catalogue includes different content types, including formal standards and other useful related content.
Source	The organisation responsible for publishing the standard (the standards body).

Table 20: Technical Standards Metadata

Property	Description
Source-Status	The status of the standard from the standards body perspective or what can be inferred from publicly available information; the status values used vary widely across standards bodies.
Publish-Date	The date on which the standard was published in the format YYYY-MM-DD or just YYYY-MM.
Status-Date	The date of the last change in the standard status in the format YYYY-MM-DD or just YYYY-MM.

Table 21: Standard Status Values

Status	Interpretation
Mandated	Used where a standard is mandated by an FDRE entity (e.g. Parliament, Cabinet or other government organisation) with the authority to do so. Where the mandate applies only to specific organisations, regions or federal levels, this is noted in the standards description.
Recommended	Used where: (1) an FDRE entity with the responsibility to recommend a standard has done so, or (2) a standard has advantages that makes it preferred above other 'Accepted' standards. This could also be for strategic reasons to guide future digital investment.
Accepted	Used where catalogue content is acceptable as a viable standard but has no particular advantage over similar standards that would make it 'Recommended'. This value is generally used when multiple competing options are available.
Rejected	Used when it is determined that a standard is not to be used in the FDRE public sector. This could be where it is deemed to be incompatible with other standards or policies, including legislation.
Development	Used where a standard that will have a wide FDRE government application is under development by: (1) an FDRE entity, or (2) An external standards body with involvement by an FDRE government organisation.
Deprecated	Used where standards may still be in use in legacy situations but should be avoided for any new implementations.
Informational	Used where catalogue content: (1) from other jurisdictions is included for deeper understanding of a subject area and completeness, or (2) is referenced when developing our own standards.
Prospective	Used where a standard has been added to the catalogue but has not yet been reviewed and allocated a long-term catalogue status.
Tracked	Represents a emerging standard that needs to be tracked because it has been identified as a possible replacement or complement for an existing standard, or is a new and potentially useful standard.

The following actions are recommended as part of the initial standards adoption process:

- All metadata elements should be completed if relevant, especially identification of the organisation mandated to manage each standard.
- Since the use and validation of standards has multiple cost and resource implications, the proposed status values in the standards catalogue should be confirmed by a standards' working group.

- This initial listing of GEA-ET standards must be reconciled with those defined by Ethiopian entities, as well as applicable policies laws and regulations, to ensure alignment and eliminate redundancy and inconsistency.
- Industry associations should be engaged to identify sector-level standards that will support cross-MDA collaboration, the determine whether workgroups should be established to drive development and/or adoption of those standards.

It is important to note that for standards can consist of multiple components (i.e., incorporating related sub-standards), not all sub-parts are listed in the catalogue and in such cases, it can be assumed that standards and their dependants have the same status.

7 Governance

In order to successfully operate an architecture function, it is necessary to establish the structures, processes, roles, responsibilities, and skills required to realise an architecture capability, whose principal responsibilities include GEA-ET development and governance. In addition to the TOGAF ADM, the GEA-ET operations model draws from the TOGAF architecture capability framework (ACF), which is widely recognised as providing some of the most useful architecture capability guidance among the popular EA frameworks. The ACF covers topics such as:

- Capability Establishment: Guidelines on how to use the ADM to establish an architecture capability [@].
- Architecture Board: Guidelines for establishing and operating a cross-organisational board whose role is generally to oversee the EA governance function [@].
- Architecture Governance: The practice and orientation by which architectures (building blocks) are managed and controlled at an enterprise-wide level [@].

Detailed guidance on these topics may be obtained at the above cited references. This section is concerned mainly with the ACF governance framework, composed of conceptual (Figure 44) and organisational (Figure 45) structures which execute the following governance responsibilities:

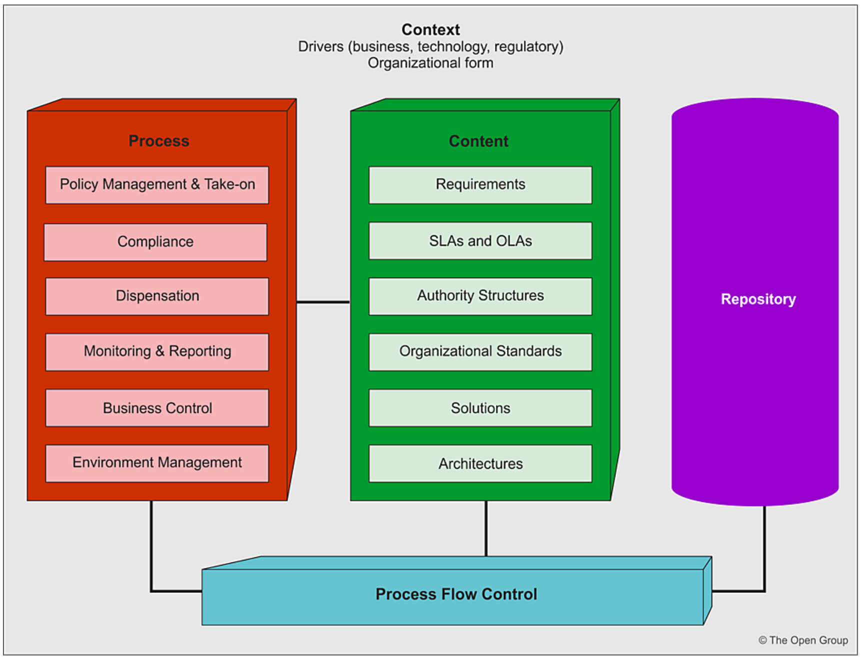


Figure 44: TOGAF Architecture Governance Framework – Conceptual Structure

- Implementing a system of controls over the creation and monitoring of all architectural components and activities, to ensure the effective management of architectures within the organisation;

- Implementing a system to ensure compliance with standards and regulatory obligations;
- Establishing processes for effective management of the above processes within agreed parameters;
- Developing practices that ensure accountability to a clearly identified stakeholder community, both inside and outside the organization.

The conceptual structure specifies the architecture governance processes (process block), the content which is governed (content block), and the content store (repository store), along with an abstract control block which determines flow and interaction among processes. Ideally, governance processes are used to ensure that all architecture work products are monitored on an ongoing basis with clear auditability of all decisions made.

With regard to the organisational structure, the area of primary interest is the "Develop" block containing the architectural roles, together with its interactions with the PMO and implementation projects. The chief architect role provides technical and business leadership to a team of enterprise architects, whose responsibility is to address enterprise-level concerns, and a domain architect team with similar responsibilities but with scopes that are limited to specific architectural domains (e.g., business, data, integration, etc.).

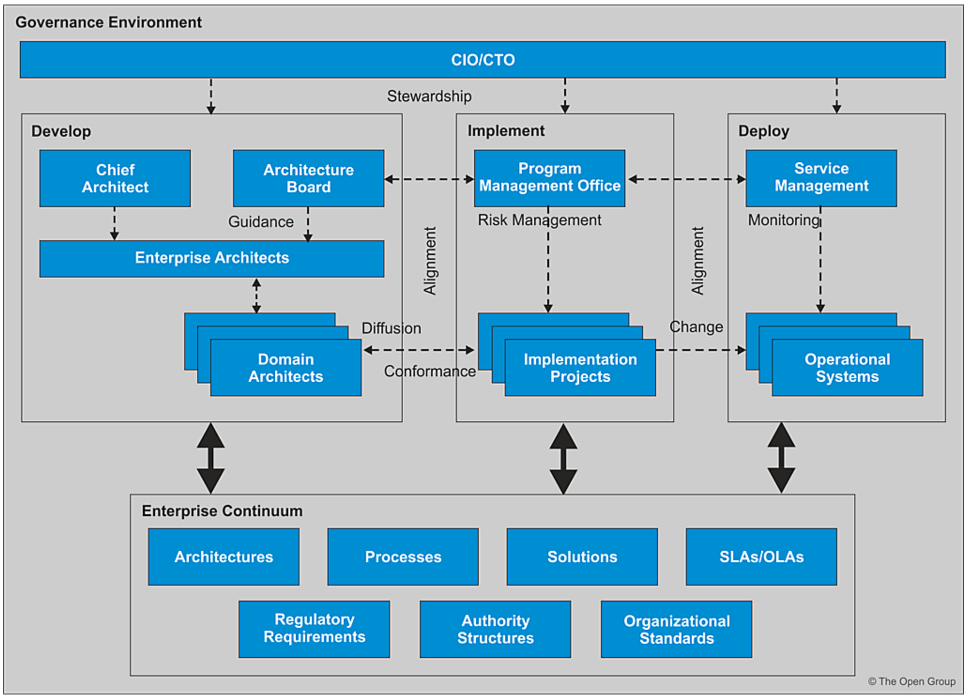


Figure 45: TOGAF Architecture Governance Framework – Organisational Structure

In this model, process-driven interactions between the architecture and implementation programme functions take place between the architecture board and PMO, and between the domain architects and implementation projects.

The latter is primarily focused on (a) diffusing architectural guidance to the implementation teams, (b) identifying and curating reusable assets from the implementation projects, (c) conducting project and solution conformance assessments. A significant feature of this structure is the enterprise continuum repository that shares architectural content between collaborating architecture, implementation and deployment teams.

The following describes how the TOFAG architecture governance model is to be interpreted and implemented for the GEA-ET, with some adaptations to align its structure and processes with the Ethiopian GEA landscape.

7.1 Structures

The GEA-ET modifies the generic TOGAF governance model to propose specific structures, processes, roles and responsibilities proposed for Ethiopia, which align with the TOGAF adaptations described in Section 2.4. Figure 46 illustrates the effect of these adjustments, which should be interpreted in conjunction with the deliverable assignment matrix of Table 22:

- The architectural roles and responsibilities are mapped onto the layered GEA-ET content model to reflect the core roles within each tier and their key deliverables; the listed solution architectures align with initiatives that have been prioritised by the digital government strategy.
- Tightly integrated programme architecture, development and operations leadership roles which all interact with the architecture board, since board responsibilities span all of these functions. The architecture board shall be convened as one of the technical working groups (TWGs) in the proposed digital government strategy oversight structure, to ensure its integration with other programme activities.

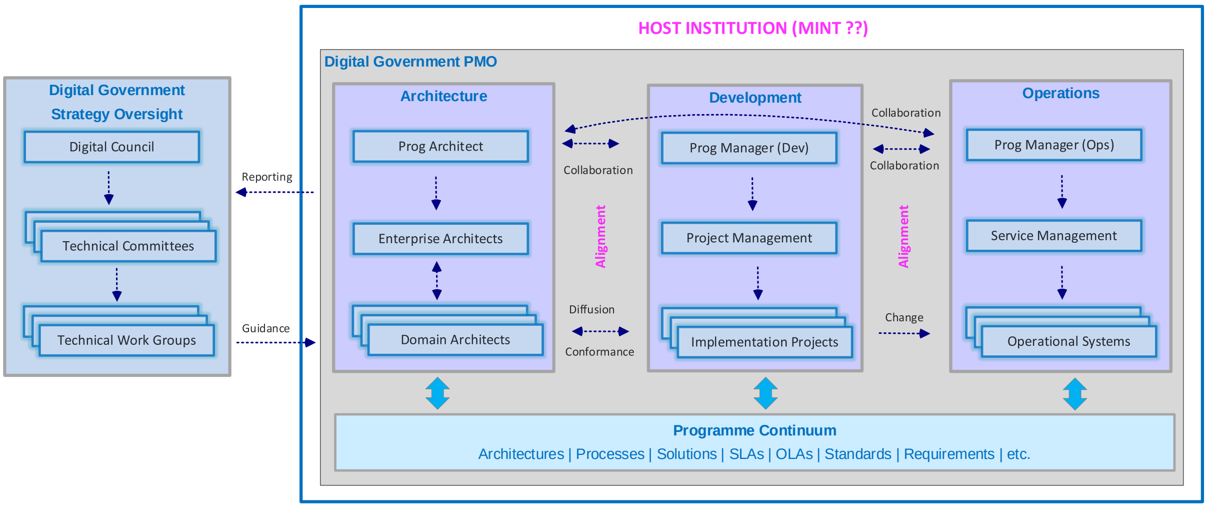


Figure 46: GEA-ET Governance Structure

Table 22: Architecture Role & Deliverable Matrix

Deliverable \ Architecture Role	Programme	Enterprise	Domain (Business)	Domain (Application)	Domain (Data)	Domain (Technology)	Domain (Integration)	Domain (Security)	Domain (Dev)	Domain (Ops)	Solution (Business)	Solution (Data)	Solution (Integration)	Solution (Interaction)
GEA-ET Framework | F	O	C	C	C	C	C	C	C	C	C				
Reference Model (Domain) | F		O	C	C	C	C	C	C	C	C				
Reference Architecture (Domain) | W			O	O	O	O	O	O	O	O	C	C	C	C
Principles (Framework) | V	O	R												
Principles (Domain) | V			O	O	O	O	O	O	O	O				
Standards/Guidelines (Framework) | V	O	R												
Standards/Guidelines (Domain) | V			O	O	O	O	O	O	O	O				
Solution Architecture (Domain) | W			A								O			
Solution Architecture (DBA) | W			A								O			
Solution Architecture (NDS) | W					A							O		
Solution Architecture (ESB) | W							A						O	
Solution Architecture (Portal) | W				A										O
Solution Architecture (MDA*) | M				A										O

O = Owner | C = Contributor | R = Reviewer | P = Recipient | A = Assessor
Solution: DBA = Bus Process Mgmt | ENDS = National Data Set | ESB = Enterprise Service Bus | Portal = eServices Portal
Domain: Interaction|Business|Data|Application|Technology|Integration|Security|DevOps|Resilience|Performance

- A single "programme" with one PMO is established to lead the delivery of all digital government services – architecture, development and operations – which are all viewed as integral programme capabilities, even though this report focuses on the architecture perspective.
- Although the TOGAF methodology is seen by some as distinct and separate from purpose-designed SDLC methodologies, the GEA-ET governance approach will be to integrate inputs and outputs into a single methodology to ensure that the "single programme" principle is viable.
- Similarly, a single 'logical' repository (termed the "Programme Continuum") is established to reference all programme content (including enterprise continuum), even though the content may be managed in different physical stores; such arrangement retards the proliferation of functional content silos and ensures transparent content sharing across architecture, development, and operations competencies.

The deliverable assignment matrix identifies the priority architecture deliverables (aligned with the digital government strategy priorities) and provides explicit guidance as to the specific roles required to create those deliverables. However, determination of the PMO host institution remains an open question, since it requires careful consideration to mitigate organisational restructuring risks.

8 Capacitation

The GEA-ET perspective on capacitation (aka capacity building) is that it is about acquiring and retaining the right collection of architectural skills to develop the GEA-ET components, whilst keeping an open mind as to where and how those skills are sourced and developed.

Broadly speaking, determining capacitation requirements consists in: (a) identifying the required roles and responsibilities, along with their skill profiles, (b) assessing existing and/or available skills to determine gaps against the required skill profiles, and (c) devising a capacitation strategy to address those gaps.

It is important to note that capacitation for nationwide digital government adoption programme can have wide scope that includes internal staff (architects, developers, project managers, etc.), external staff (MDAs and other service providers), and service users (individuals and businesses). The digital government strategy addresses these broader capacitation requirements; this section is concerned exclusively with developing a capacitation strategy for the architecture function.

8.1 Skill Requirements

The architecture deliverable/role matrix of Table 22 specifies the architectural roles required to deliver the programme's priority architectural deliverables, such as reference models, reference architectures, guidelines, standards, and solution architectures. The TOGAF architecture capability framework (ACF) proposes an architecture skills framework (ASF) [@] which defines the skills required to establish a 'typical' enterprise architecture practice in terms of skill and experience norms that are mapped to each role. The GEA-ET integrates the deliverable/role matrix and architecture skills framework to derive a specification of architectural skills requirements, with some modifications to incorporate specific GEA-ET needs and to simplify interpretation:

- Extends the ASF scope to include vertical integration, security, devops, and resilience domains, since it only covers the traditional architecture domains (enterprise, business, data, application, technology).
- Recodes the ASF-proposed Awareness/Knowledge/Expert taxonomy of achievement levels to a simplified Low/Medium/High scale (see Table 23).
- Maps the roles to ASF skill categories (see Table 24) rather than the detailed itemised skills, to provide an aggregated view of proficiency levels (expressed as a range) which is adequate for the purpose of assessing skills gaps in the GEA-ET case.
- In terms of the ASF, the GEA-ET Programme Architect role is equivalent to the EA Manager role, whilst the GEA-ET solution architect role is similar to the IT Designer role; the programme/project manager roles have the same meaning.

The ASF specifies these responsibilities for the enterprise, solution and segment (sector) architect roles:

- Enterprise Architect: Architectural design and documentation at a landscape and technical reference model level; often leads a group of the segment and/or solution architects related to a given programme; the focus of the is on enterprise-level business functions required. Domain architects are viewed as domain-specialised enterprise architects.
- Segment Architect: Architectural design and documentation of specific business problems or organizations; re-uses the output from all other architects, joining detailed technical solutions to the overall architectural landscape; the focus is on enterprise-level business solutions in a given domain, such as finance, human resources, taxation, etc.
- Solution Architect: Architectural design and documentation at a system or subsystem level, such as integration, management or security; may shield the enterprise/segment architect from the unnecessary details of the systems, products, and/or technologies; the focus is on system technology solutions; for example, a component of a solution such as an enterprise service bus.

The GEA-ET operational model positions solution architects as an integral part of the extended programme architecture team who play a critical role in integrating architecture and implementation project content and activities. In practice, the expectation is that solution architects will be part of the core programme architecture team since they are typically responsible for developing transversal components at the WOG level whose capabilities cut across MDAs. On the other hand, segment architects will typically be part of the MDA teams whose role is to develop MDA-specific solutions.

Table 23: TOGAF ASF / GEA-ET Competency Level Mapping

TOGAF ASF		GEA-ET		Description
Level	Achievement	Level		
1	Background	--	--	Not a required skill, though should be able to define and manage the skill if required.
2	Awareness	L	Low	Understands the background, issues, and implications sufficiently to be able to proceed further and to provide appropriate advice.
3	Knowledge	M	Medium	Detailed knowledge of subject matter and able to provide professional advice and guidance. Ability to integrate capability into architecture design.
4	Expert	H	High	Extensive and substantive practical experience and applied subject matter expertise.

Table 24: TOGAF ASF: Skill Categories

Category	**Description**
Generic	Leadership, teamworking, inter-personal skills, etc.
Domain specific	Architecture-level domain skills (e.g., business, data, security, etc.)
Solution (segment) specific	Solution-level technology or segment skills (e.g., taxation, integration, security, etc.)

Business Knowledge & Methods	Business cases, business process, strategic planning, etc.
Enterprise Architecture	Modelling, building block design, applications and role design, systems integration, etc.
Program or Project Mgmt	Managing business change, project management methods and tools, etc.
IT General Knowledge	Brokering applications, asset management, migration planning, SLAs, etc.
Technical IT	Software engineering, security, data interchange, data management, etc.
Legal Environment	Data protection laws, contract law, procurement law, fraud, etc.

Table 25: Architecture Skill Requirements

Skill Category \ Architect Role	Programme	Enterprise	Domain (Business)	Domain (Application)	Domain (Data)	Domain (Technology)	Domain (Integration)	Domain (Security)	Domain (Dev)	Domain (Ops)	Solution (Business)	Solution (Data)	Solution (Integration)	Solution (Interaction)
Generic	H-H	M-H	M-H	M-H	M-H	M-H	M-H	M-H	M-H	M-H	L-M	L-M	L-M	L-M
Architecture domains	L-M	L-M	H-H	H-H	H-H	H-H	H-H	H-H	H-H	H-H	L-M	L-M	L-M	L-M
Solution segments	L-M	L-M	L-M	L-M	L-M	L-M	L-M	L-M	L-M	L-M	M-H	M-H	M-H	M-H
Business Knowledge & Methods	H-H	H-H	H-H	M-H	M-H	M-H	M-H	M-H	M-H	M-H	M-H	L-M	L-M	L-M
Enterprise Architecture	H-H	H-H	M-H	H-H	M-H	M-H	M-H	M-H	M-H	M-H	L-M	L-M	L-M	L-M
Program or Project Mgmt	M-H	M-M	M-H	M-M	M-M	M-M	M-M	M-M	M-M	M-M	L-L	L-L	L-L	L-L
IT General Knowledge	M-H	M-H	M-H	M-H	M-H	M-H	M-H	M-H	M-H	M-H	M-M	M-M	M-M	M-M
Technical IT	M-M	M-H	L-M	M-H	M-H	H-H	M-H	H-H	H-H	H-H	M-M	M-M	M-M	M-M
Legal Environment	L-H	L-M	L-M	L-M	L-M	L-M	L-M	L-M	L-M	L-M	L-L	L-L	L-L	L-L

The result of integrating the ASF skill categories and architect roles in shown in Table 25, to provide an aggregated view of the architect skill profiles required to for the core programme team and to support the priority implementation projects. It is to be note that this statement of skill requirements is qualitive, in the sense that it reflects role rather than staff commitments; the expectation is that the number of staff full time equivalences per role will be determined when the delivery effort for the different initiatives is better understood at implementation planning time.

8.2 Skill Gaps

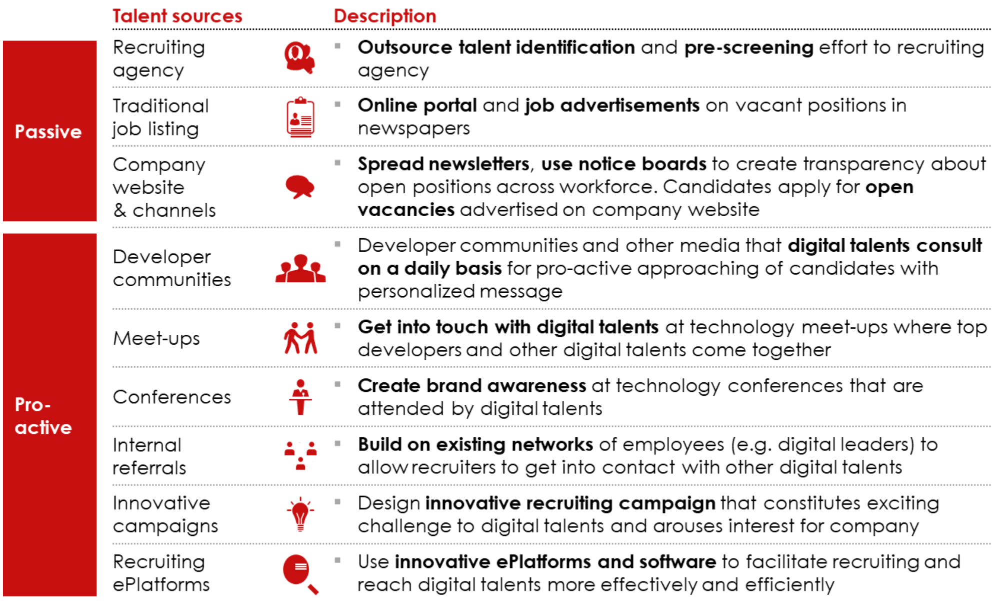
The GEA-ET diagnostic reported an acute lack of architecture skills in the FDRE government with no clear plan as to how they will be sourced, which is exacerbated by the absence of a capacity development plan that focuses on the needs of architecture professionals; other GEA-related assessment reports like [BE-2022-Giz] and [ES-2023-Tbi] concur with these findings. Therefore, the GEA-ET capacitation model assumes a 'greenfield' architecture skills baseline, even though some MDAs may have islands of architectural competence.

8.3 Strategy

Given this significant architecture skills gap, the GEA-ET capacitation strategy addresses skills acquisition, development, and retention concerns. It considers short-term actions to establish the architecture function in a 'reasonable' amount of time, and long-term actions to build a sustainable competence development programme that can provide architectural skills on an ongoing basis. Furthermore, it adopts a holist approach that caters for the needs of different groups of concern: the core programme architecture team comprised of enterprise, domain and solution architects, and the solution implementation teams at the WOG and MDA levels who will require a broad range of technology skills.

8.3.1 Talent Acquisition & Retention

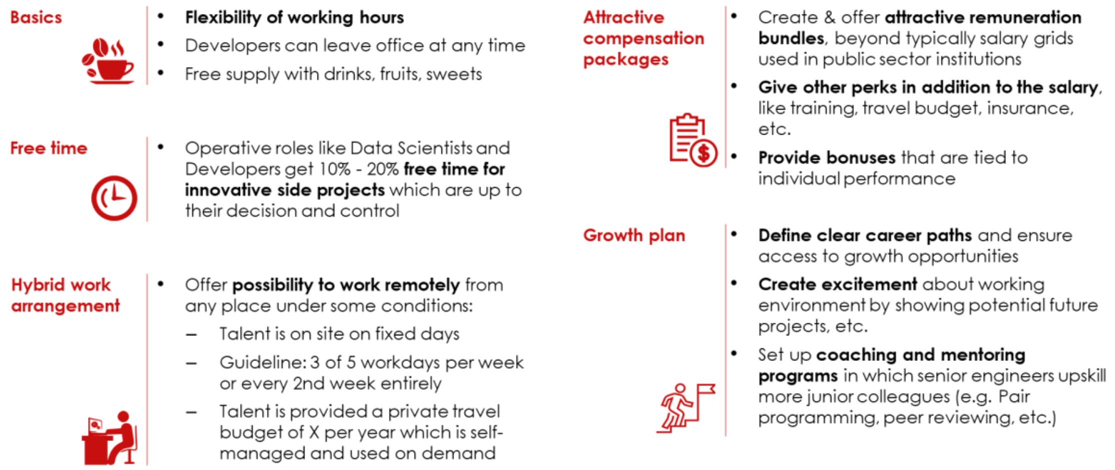
For the acquisition and retention skills for all the programme delivery teams (architecture, development and operations, see Figure 46), the GEA-ET proposes adoption of the Digital Ethiopia Factory (DEF) concept proposed by [ES-2022-Giz] as a capacity building model.



	Talent sources	Description
Passive	Recruiting agency	**Outsource talent identification** and **pre-screening** effort to recruiting agency
	Traditional job listing	**Online portal** and **job advertisements** on vacant positions in newspapers
	Company website & channels	**Spread newsletters**, **use notice boards** to create transparency about open positions across workforce. Candidates apply for **open vacancies** advertised on company website
Pro-active	Developer communities	Developer communities and other media that **digital talents consult on a daily basis** for pro-active approaching of candidates with personalized message
	Meet-ups	**Get into touch with digital talents** at technology meet-ups where top developers and other digital talents come together
	Conferences	**Create brand awareness** at technology conferences that are attended by digital talents
	Internal referrals	**Build on existing networks** of employees (e.g. digital leaders) to allow recruiters to get into contact with other digital talents
	Innovative campaigns	Design **innovative recruiting campaign** that constitutes exciting challenge to digital talents and arouses interest for company
	Recruiting ePlatforms	Use **innovative ePlatforms and software** to facilitate recruiting and reach digital talents more effectively and efficiently

Figure 47: External Skills Acquisition Channels

DEF aims to leverage various internal and external skills sources and acquisition channels (see Figure 47 which elaborates external sources) to build and sustain a talent pool assembled for the specific purpose of supporting business-focused aspects of Ethiopia's digital government strategy; the DEF approach is flexible enough to accommodate diverse sourcing models e.g., insourcing, outsourcing, out-staffing, etc. DEF also proposes a portfolio of incentives that can be put in place to attract and retain best-in-class digital talents (see Figure 48).



Basics

- **Flexibility of working hours**
- Developers can leave office at any time
- Free supply with drinks, fruits, sweets

Free time

- Operative roles like Data Scientists and Developers get 10% - 20% **free time for innovative side projects** which are up to their decision and control

Hybrid work arrangement

- Offer **possibility to work remotely** from any place under some conditions:
  - Talent is on site on fixed days
  - Guideline: 3 of 5 workdays per week or every 2nd week entirely
  - Talent is provided a private travel budget of X per year which is self-managed and used on demand

Attractive compensation packages

- Create & offer **attractive remuneration bundles**, beyond typically salary grids used in public sector institutions
- **Give other perks in addition to the salary**, like training, travel budget, insurance, etc.
- **Provide bonuses** that are tied to individual performance

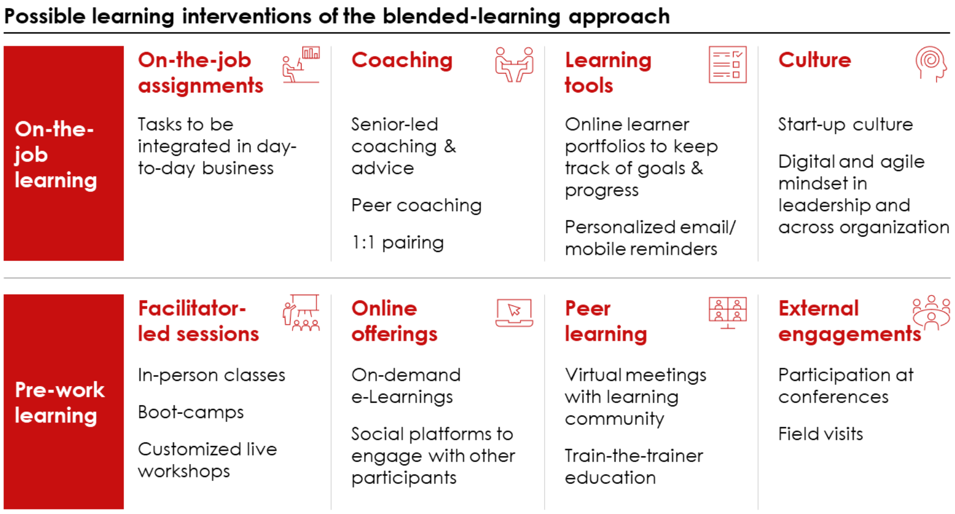
Growth plan

- **Define clear career paths** and ensure access to growth opportunities
- **Create excitement** about working environment by showing potential future projects, etc.
- Set up **coaching and mentoring programs** in which senior engineers upskill more junior colleagues (e.g. Pair programming, peer reviewing, etc.)

Figure 48: Talent Incentive Framework

8.3.2 Talent Development

A capacitation model that relies solely on the acquisition and retention of new talent is unlikely to satisfy architecture skills requirements and will likely prove to be unpopular with existing staff who may feel that they are being ignored.



Possible learning interventions of the blended-learning approach

On-the-job learning	**On-the-job assignments** Tasks to be integrated in day-to-day business	**Coaching** Senior-led coaching & advice Peer coaching 1:1 pairing	**Learning tools** Online learner portfolios to keep track of goals & progress Personalized email/ mobile reminders	**Culture** Start-up culture Digital and agile mindset in leadership and across organization
Pre-work learning	**Facilitator-led sessions** In-person classes Boot-camps Customized live workshops	**Online offerings** On-demand e-Learnings Social platforms to engage with other participants	**Peer learning** Virtual meetings with learning community Train-the-trainer education	**External engagements** Participation at conferences Field visits

Figure 49: Blended-Learning Capacitation Intervention Options

Therefore, the GEA-ET also proposes adoption of the DEF blended-learning model, which provides a framework for developing new and existing talent, using both on-the-job and pre-work learning approaches (see Figure 49). Components of this blended-learning portfolio can be identified and curated by subject-matter experts, and then combined to create individualised professional development plans.

The DEF model has been developed to support the Ease-Of-Doing-Business (EODB) component of Ethiopia's digital government strategy, it can be extended to other digital government initiatives since it provides a generic and well-structured capacity development framework.

8.4 Architect Development

To complement the above-described generic capacitation approaches, the GEA-ET proposes the establishment of a formal architect development programme that leverages The Open Group Certified Architect (Open CA) certification programme, which is "*designed to validate the existence of those qualities and skills in a professional that enable the effective practice of IT architecture*". The Open CA programme is generally recognised as the worldwide gold standard for professional [practice-based] architect certification, with multiple participant benefits that include:

- Provides a career framework with certification that is recognized by world-leading organizations;
- Provides an objective, reliable measure of candidate capabilities and qualifications;
- Ensures a more efficient and successful recruiting process with consistently positive results;
- Enables organizations to formalise and recognise career progression;
- Helps identify the best candidates for critical roles and responsibilities.

The Open CA programme is available for four architecture disciplines whose subject matter is summarised in Table 26 - business, digital, enterprise, and solution – leading to experience-based certification at three different levels: certified, master certified, and distinguished.

Table 26: Open CA Architecture Disciplines

Discipline	Subject Matter
Business Architecture	The formalized description of how an organization uses its essential competencies for realizing its strategic intent and objectives, as defined in the Open Business Architecture (O-BA) Preliminary Standard [@]. This typically describes the structure and interaction between the business strategy, organization, functions, business capabilities, and information needs.
Digital Architecture	The formalized description of the digital solutions and roadmap for an enterprise to determine how the enterprise can undergo Digital Transformation to enable innovation, and to achieve its strategic goals. It supports the synthesis of complex business, technology, and client issues into real-world strategies through the application of innovative approaches to solution crafting.

Table 26: Open CA Architecture Disciplines

Discipline	Subject Matter
Enterprise Architecture	The formalized description of the structure and operation of an enterprise (an organization or group of organizations with a common set of goals) that is intended to determine how the enterprise can most effectively achieve its current and future objectives. It defines the frameworks, models, mechanisms, and structures so that the capabilities of an enterprise evolve in a way that effectively supports the business strategy or intent. It provides guidance and governance of projects and transition steps to the future state.
Solution Architecture	The formalized description of solutions to business problems and opportunities developed through the reasoned application of technologies, people, and processes to successfully deliver systems and capabilities that support the needs of the business.

Depending on certification discipline and level, there are multiple pathways to Open CA certification which generally consist in candidates achievement of specific milestones that cover communication skills, accrued experience, and professional development [@] (see Figure 50); certification awards are determined by specially convened peer review boards.

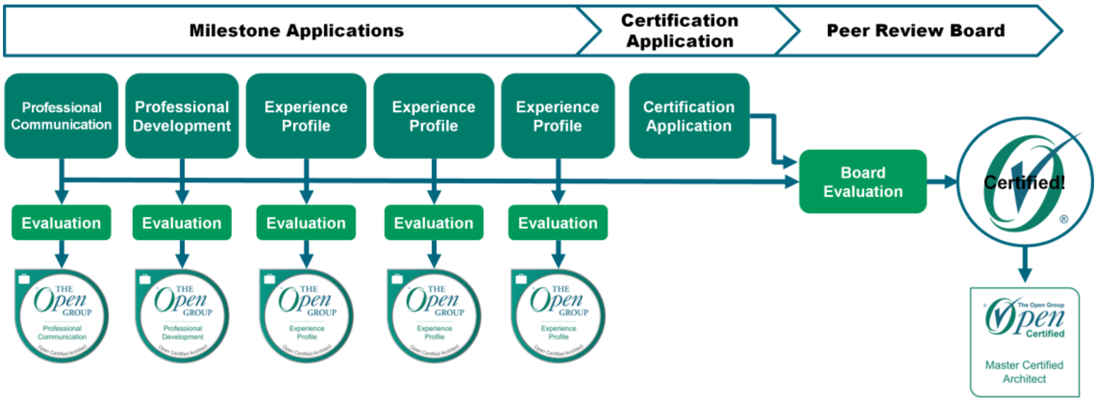


Figure 50: The Open CA Master Certified Certification Pathway

The Open CA programme allows for direct certification with The Open Group or indirect certification via an accredited external Open CA programme. Whilst the option exists for Ethiopia to establish an accredited Open CA, this must be undertaken as long-term project because the accreditation criteria are likely to be onerous for Ethiopia in the short- or even medium-term. Alternatively, Ethiopia could take the lead in establishing a partnership among like-minded [African ??] countries interested in pursuing Open CA certification for their architecture professionals – this approach could possibly accelerate creation of the critical expertise and participation mass required to support such an initiative, although accreditation criteria must still be met.

For these reasons, it makes sense for Ethiopia to pursue the direct certification pathway in the short-term, notably an approach that does not preclude the country partnership model. In this scenario, the FDRE government could approach The Open Group to assist in identifying and connecting with certified architects who might be interested in supporting the Ethiopian certification programme. This strategy is feasible because the Open CA programme recognises 'giveback' activities as valid certification achievement milestones; such recognition can serve as an incentive for prospective contributors.

Within Ethiopia, a capacity building organisation that focuses on digital talent development could host the programme, with the prime candidate being Addis Ababa University because it has participated in (and even led) multiple GEA-related initiatives in Ethiopia.

It is strongly recommended that the Open CA programme option be pursued to give the Ethiopian GEA-ET development initiative the best chance of long-term success, by establishing a competence development programme that can provide much-needed architectural skills on a sustainable foundation.

8.5 Ecosystem Sourcing

Another way to complement the above capacitation approaches is to tap into various ecosystems that have been established to underpin international development initiatives. The ‘tapping’ strategy rationale is that many of these ecosystems have been founded to bootstrap digital transformation initiatives in [mostly] LMICs like Ethiopia, so it makes sense for Ethiopia to source and collaborate with expertise from the skills pools embedded therein. In other words, the intent is to deliberately position these ecosystems as part of an extended GEA-ET delivery team.

Of particular interest in this grouping is the GovStack ecosystem, which was founded to "*accelerate countries' ownership of [digital government] solutions and in doing so improve services for social well-being*". The GovStack engagement model provides for countries to learn from other experts and to share best practices, such as making contributions to the development of GovStack-relevant assets. Ethiopia can leverage this provision to drive objectives like accelerated GEA-ET delivery and architecture capacitation in multiple ways.

GEA-ET reference modelling is a prime use case for the application of the capacity tapping strategy. In this regard, GovStack has solid solution building blocks but does not have normative structures (i.e., architectural reference models) to enable classification and comparative analysis of those building blocks. This presents an opportunity for Ethiopia to propose (and possibly lead) a GovStack reference modelling initiative that takes the form of a technical working group.

This approach taps ecosystems skills to accelerate attainment of Ethiopia's GEA-ET objectives, whilst creating assets of value to the wider GovStack ecosystem; it can be applied to develop other GEA-ET deliverables like reference and solution architectures. The collaborative engagement process ensures knowledge transfer from collaborating experts to the Ethiopian team. This is the essence of our proposed ecosystem sourcing model.

Further, it is worthwhile noting that appropriately structured ecosystem sourced initiatives can be positioned as qualifying milestones for the Open CA certification programme, providing an incentive for architect participation and other countries to join Ethiopia in partnerships around such initiatives.

9 Roadmap

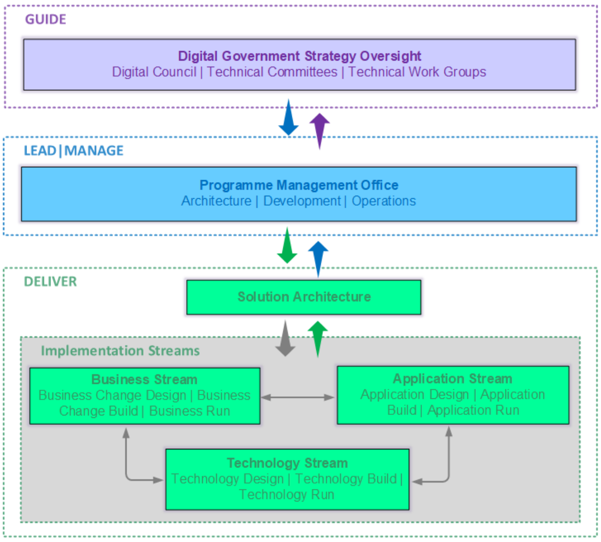
The government enterprise architecture roadmap (GEAR) sets out the plan for coordinating and executing the changes needed to realise the GEA, whilst considering the binding constraints set by the broader digital government roadmap (DGR) in which the GEAR has to be embedded and influencing initiatives like policy developments and stakeholder capacitation.

The GEAR itself is comprised of a resource schedule, which specifies delivery resources, a deliverable schedule which outlines the key activities and deliverables, and a delivery plan, which organises resource assignments and deliverables along a timeline. The delivery plan is developed iteratively, with each iteration attempting to more closely match initiative delivery effort with resource capacity and availability, within an activity flow that is designed to regularly deliver business value to GEA-ET stakeholders.

The proposed delivery plan represents a first iteration of the roadmap, a baseline which will undoubtedly change as more detail emerges around critical programme parameters like scale, scope and priorities, as well as the availability and quality of resources. Discussed first are two concepts that form elements of the delivery plan specification framework – operational model and planning horizon.

9.1 Operational Model

Derived from the governance framework of Figure 46, the operational model on which the GEAR is based is shown in Figure 51, which outlines the purpose of each grouping of programme operational units (denoted by the Guide, Lead|Manage and Deliver blocks) and the constituent parts of the delivery structure: the solution architecture function and the business, application, and technology streams.


GUIDE
Digital Government Strategy Oversight
Digital Council | Technical Committees | Technical Work Groups
LEAD|MANAGE
Programme Management Office
Architecture | Development | Operations
DELIVER
Solution Architecture
Implementation Streams
Business Stream
Business Change Design | Business Change Build | Business Run
Application Stream
Application Design | Application Build | Application Run
Technology Stream
Technology Design | Technology Build | Technology Run

Guide

- Digital Council (DGC): Makes strategic decisions and coordinates decision-making processes; mobilises and allocates resources, especially finance; formulates and promulgates policy; motivates, facilitates, and monitors stakeholder preparedness and participation; communicates programme goals, objectives, and progress [Quarterly interlock cycle].
- Technical Committee (TEC): Primary platform for coordinating and reporting on activities across initiatives, lines of business (sectors), architecture domains, and delivery streams; engages external advisors as required [Monthly interlock cycle] .
- Technical Work Group (TWG): Collection of resources assembled to support [typically] specific initiatives; members may provide oversight, advice and/or be directly involved in implementation activities.

Lead | Manage

- Programme Management Office (PMO): Executes the strategy communicated by DGC; coordinates resources and activities that cut across initiatives; drives programme visibility; secures resources and aligns their activities with the roadmap; manages interplays between initiatives; reports progress and issues for resolution to oversight bodies; provides management and architecture leadership to the initiatives; manages contracts with staff, and technology and services providers

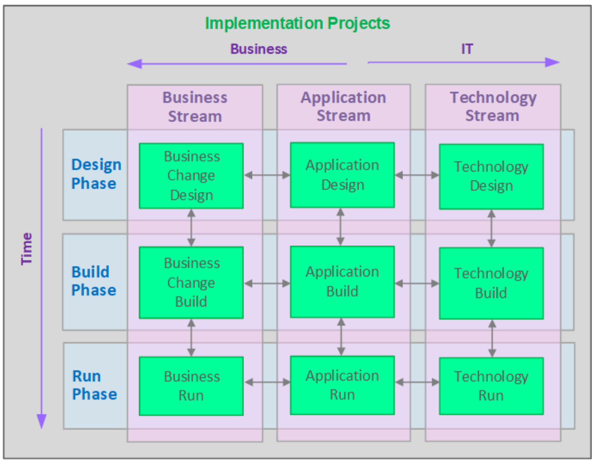
Deliver

- Business Stream: Identifies business goals and drivers, develops descriptions of business operations that respond to those drivers, deploys the operational changes, and supports execution of the deployed business operations.
- Application Stream: Designs, implements and operates the information systems that enable business operations, including application, data, integration, security, development, and operations services.
- Infrastructure Stream: Delivers the software, hardware, and network infrastructure on which the applications run and through which they interact.

Figure 51: Programme Operational Model

The project phase/stream matrix of Figure 52 further enumerates some of the options for organising the activity phases and competency streams of implementation projects: (a) solution-oriented structuring in which each project provides resources for all the streams and phases, (b) competence-oriented organisation in which competency clusters are setup for each stream (i.e., business, application, technology streams) to provide support for each project, or (c) a combination of the solution- and competence-based structures. Note that data, security and integration are part of the application stream, whilst development and operations are grouped into the technology stream.

The matrix derives from IBM's '10-box' IT service delivery model, which is designed to reflect traditional IT service delivery structures that can be easily understood by a broad range of stakeholders in IT services. The '10-box' model is implementation-focused, and it enables identification and scheduling of resources and activities for the delivery of IT services. Factors such as skills availability, skills sourcing model, existing delivery practices, and adoption lead time must be considered when selecting which configuration to deploy.



- Business Change Design: Based on the business architecture, defines revised (or new) business structures and processes, organisation and user stakeholder roles and responsibilities, and a change delivery roadmap.
- Business Change Build: Various change agents deploy (or build) the changes into the business operations of MDAs and other service providers.
- Business Run: Provides ongoing support, training, performance monitoring, and business improvement services to stakeholders as they utilise the deployed processes and structures.
- Application Design: Guided by the enterprise architecture, performs micro-design of the application components that enable SOA services.
- Application Build: Utilising development layer services, builds and deploys application components whilst ensuring compliance with the micro-design; build and run competencies must collaborate closely, especially during deployment activities.
- Application Run: Operates and manages the deployed applications, including ongoing support internal and external end-users, (typically) via a service management helpdesk.
- Technology Design|Build|Run: Guided by solution architectures, designs, implements, and operates the infrastructure on which the solutions run, including compute, storage, and network services.

Figure 52: Implementation Project Structure

The GEA-ET recommendation is to initially organise projects by solution (i.e., DBA, ESB, etc.) since the expectation is that the prioritised WOG projects will be adequately resourced; competence-based organisation can be considered when the MDA projects are scheduled to begin, since these initiatives will likely require extensive (shared) support from the programme architecture team.

9.2 Planning Horizon

The planning horizon may be thought of as the amount of time organisations will look into the future when preparing a strategic plan. In the GEA-ET context, the planning horizon is a timeline with a bounded duration with specific parameters that are used in specifying the GEA-ET roadmap:

- Duration: Sets an upper bound on the roadmap duration; in this case the TOR [@] specifies a 5-year horizon (2024-2028) for the digital government strategy.

- Granularity: The degree to which the timeline is resolved into named 'time slots' by which significant activity and deliverable events are scheduled (e.g., start, end, review, deploy, etc) and tracked; the resolution can initially be set to a calendar quarter for reasons explained below.
- Dimensions: The road uses two dimensions: the GEA-ET content model **tier** (Framework, WOG, MDA, Transversal) which essentially sets activity/deliverable priority, and the delivery **phase** (Plan, Design, Build, Run) which represents progression within the standard solution delivery lifecycle.

The interpretation and usage is that activities have deliverables and scheduled start and end dates with assigned to time slots (quarters) on the planning horizon and they are classified by tier and phase. Thus, the roadmap is described simply as a listing of activities grouped by tier and phase dimensions against a series of named timeslots.

The initial choice of resolution level must be carefully considered. It must align with the prevailing workplace culture, as well as the programme team's ability and opportunity to manage activities at a specified level of granularity. There is no point in expending the considerable effort needed to create a week-granulated schedule, if the workplace culture does not recognise weekly deadlines as important targets to be met. Similarly, fine-grained planning effort will be wasted if the requisite skills and experience are not available to manage the programme at refined granularity levels, or if the opportunity (time) is not provided to develop and apply such practices.

In such scenarios it makes sense to use reduced granularity levels and hence the management workload, but this invariably leads to less reliable schedules, which may be an acceptable concession if stakeholders are less sensitive to deadline transgressions. Initiating the programme with a quarter-based schedule makes sense because (a) use of finer granularity would be superfluous at this time when many 'unknowns' still exist, and (b) resolution is not fixed can always be increased as effective programme management tools and processed are deployed and become established.

9.3 Resource Schedule

Resources represent the persons assigned to the roles depicted in the resource schedule of Figure 53, which specifies architecture, development, and operations roles at the programme level, and business, development, and operations roles at the implementation project level. The following are important points to note regarding the resource schedule:

- The schedule identifies all the roles required for the architecture team but the roles in the other blocks should be viewed as placeholders for the development, operations, and business team roles that will be clarified at implementation time. They also serve as proxies for assignment of key architecture-related activities and deliverables.
- The schedule structure does not consider existing or planned organisational structures and require adjustment to map to and align with those structures.

- It must be emphasised because it is often not understood that a role does not equate to a person, since a person may assume multiple roles and/or the same role may be performed by multiple persons. The mapping of roles of persons depends on multiple factors such as aggregate skills availability, individual skill profiles, and activity content and volume.

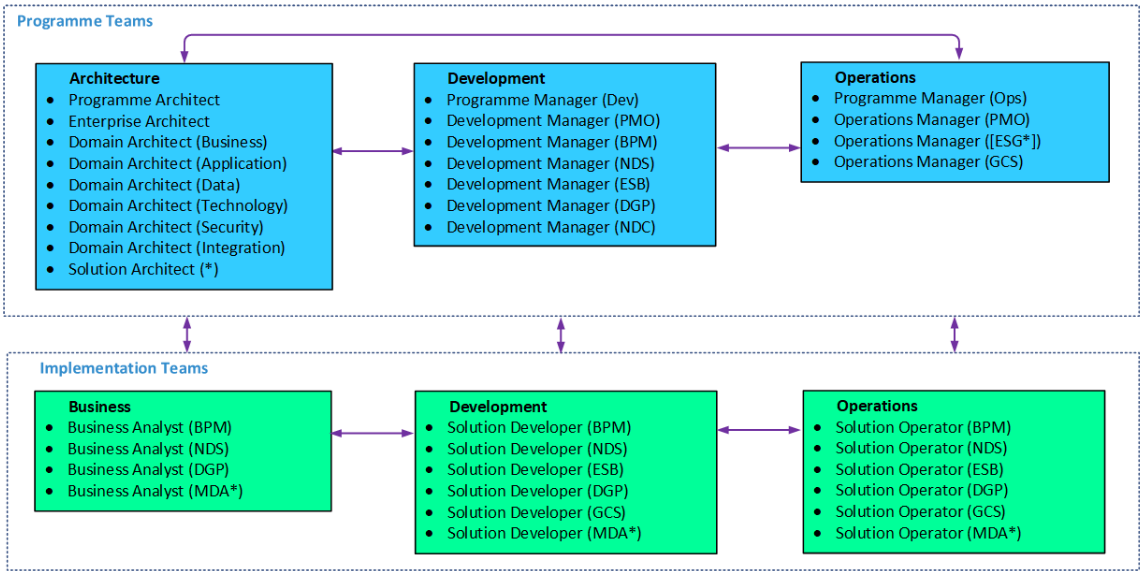


Figure 53: Programme Resource Schedule

- The MDA* notation represents the collection of MDA initiatives to be prioritised for implementation when the common WoG capabilities are available.
- The Solution Architect (*) notation signifies that multiple solution architects will likely be required to develop the WOG and MDA solution architectures and to support the MDA implementation teams.
- Regarding methodology, A prime responsibility of the (PMO) suffixed management roles is to setup, manage and operate the PMO tooling and platforms that will support PMO operations, including ADM, SDLC and M&E functions.

It is expected that the programme architecture and management leadership roles will be in place to lead selection and establishment of their teams, and to collaborate in selecting and deploying tools to support the selected programme delivery methodology.

9.4 Deliverable Schedule

Outlined in Table 27, the deliverable schedule is a simple listing of deliverables, with a description of the responsibilities associated with each role that is assigned to each deliverable. The listing contains the core architecture deliverables listed in Table 25, other foundational artefacts on whose completion they depend (e.g., cloud and resilience strategies), supporting deliverables like the SDLC methodology and tooling. Others are activity-based deliverables which form part of the architects' core responsibilities (e.g., oversight and compliance validation).

Table 27: Deliverable Schedule

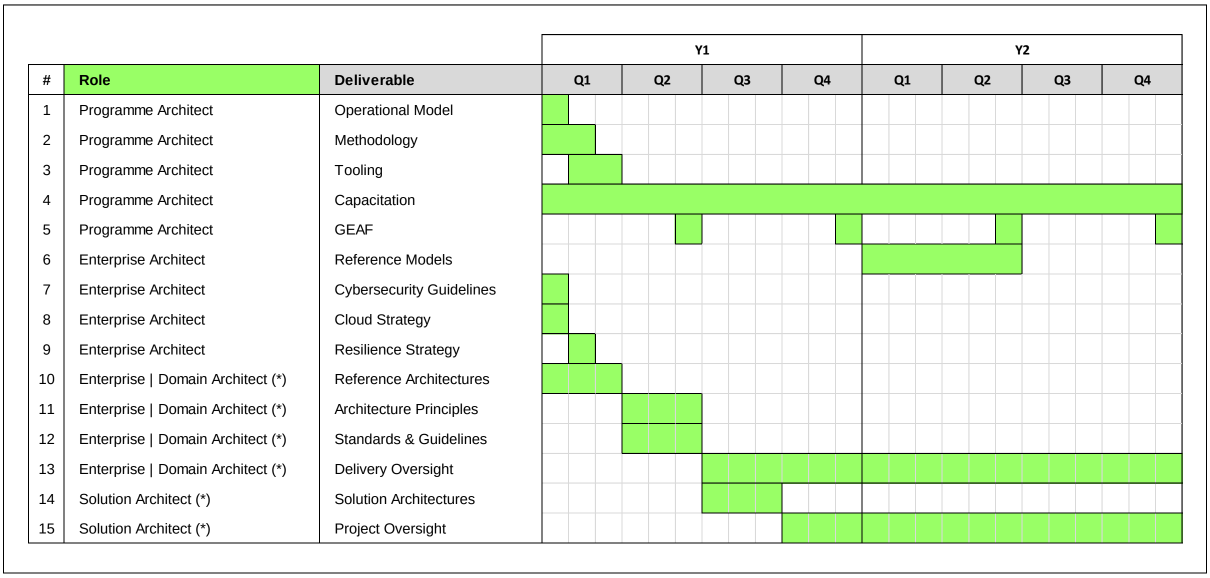
Role	Deliverable	Responsibilities
Programme Architect	Operational Model	Collaborate with stakeholders to agree the project organisation model and generate solution phase/stream matrix which identifies competence providers; this matrix provides critical guidance for developing the solution architectures.
Programme Architect	Methodology	Collaborate with programme leadership roles to select, configure and deploy programme SDLC methodology, which must address architecture, development, and operations functions; integrate GEA-ET methodology and tools into programme SDLC methodology and tooling.
Programme Architect	Communication Plan	Develop and execute a comprehensive communication plan to propagate and gain buy-in for GEA-ET goals and objectives among stakeholders.
Programme Architect	Tooling	Select, configure, and deploy tooling to operationalise GEA-ET methodology, using an approach that basically customises generally available TOGAF-compliant tooling.
Programme Architect	Capacitation	Develop and execute architect capacitation plan including acquisition, retention, and development strategies, based on guidance provided in Section 8.
Programme Architect	GEA-ET	Regularly review and revise GEA-ET structure and content based on industry trends and utility feedback from programme/implementation teams and other stakeholders.
Enterprise Architect	Reference Models	Develop reference models based on country and industry best practices as per guidance; develop content collection templates/guidelines and lead/coordinate collection of model data on WOG solutions.
Enterprise Architect (*)	Cybersecurity Guidelines	Develop concrete implementation guidelines for cybersecurity controls at the WOG and MDA levels, based on GEA-ET guidance (Section), FDRE government policies and regulations, and well as directives from agencies like RISA.
Enterprise Architect (*)	Cloud Strategy	Develop cloud strategy that identifies opportunities to adopt cloud services so that opportunities and risks are balanced; the strategy must address the relevant characteristics of public, private, and hybrid could services.
Enterprise Architect	Resilience Strategy	Develop a business resilience strategy to protect against, reduce the likelihood of the occurrence of, prepare for, respond to and recover from disruptions to business and IT services, when they arise.
Enterprise | Domain Architect (*)	Reference Architectures	Review and refine proposed reference architecture and validate alignment with related existing, in-progress and planned strategies, solutions, standards, regulations, and policies; develop reference architectures for development, operations, and resilience domains based on GEA-ET guidance in Section 4.
Enterprise | Domain Architect (*)	Architecture Principles	For respective domains, lead collaboration with stakeholders to review, rationalise, harmonise, prioritise, and agree statuses of the principles referenced in Section 5; manage records on an ongoing basis.
Enterprise | Domain Architect (*)	Standards & Guidelines	For respective domains, lead collaboration with stakeholders to review, rationalise, harmonise, prioritise, and agree statuses of the standards and guidelines referenced in Section 6, including completion of missing data elements; manage records on an ongoing basis.
Enterprise | Domain Architect (*)	Delivery Oversight	For respective domains, communicate reference architectures to solution architects along with related principles, standards, and guidelines; advise on respective domains on programme-wide basis.

Table 27: Deliverable Schedule

Role	Deliverable	Responsibilities
Solution Architect (*)	Solution Architectures	For respective WOG and MDA solutions, develop solution architectures based on related reference architectures, principles, standards, and guidelines; develop the minimal set of artefacts as per artefact rationalisation guidance provided in [XXX]
Solution Architect (*)	Project Oversight	Communicate solution architectures to implementation project teams; provide oversight and implementation guidance to ensure conformance with reference and solution architectures; lead and coordinate solution design and implementation compliance reviews.

9.5 Delivery Plan

As stated earlier, the delivery plan is a roadmap component which organises resource and deliverable pairings along the planning horizon, in a way that attempts match initiative delivery effort with resource capacity and availability. The governance function uses the delivery plan to track programme progress (or the lack thereof) against the roadmap, so that deviations can be corrected. The current plan in Figure 54, assigns the deliverables to quarters on the planning horizon, based the current understanding of priorities and dependencies. These assignments are only done for the first two years (Y1-Y2) since lengthier projections do not make sense given the limited information available at this time.



			Y1												Y2											
#	Role	Deliverable	Q1			Q2			Q3			Q4			Q1			Q2			Q3			Q4		
1	Programme Architect	Operational Model	🟩																							
2	Programme Architect	Methodology	🟩	🟩																						
3	Programme Architect	Tooling		🟩	🟩																					
4	Programme Architect	Capacitation	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
5	Programme Architect	GEAF						🟩						🟩						🟩						🟩
6	Enterprise Architect	Reference Models													🟩	🟩	🟩	🟩	🟩	🟩						
7	Enterprise Architect	Cybersecurity Guidelines	🟩																							
8	Enterprise Architect	Cloud Strategy	🟩																							
9	Enterprise Architect	Resilience Strategy		🟩																						
10	Enterprise | Domain Architect (*)	Reference Architectures	🟩	🟩	🟩																					
11	Enterprise | Domain Architect (*)	Architecture Principles				🟩	🟩	🟩																		
12	Enterprise | Domain Architect (*)	Standards & Guidelines				🟩	🟩	🟩																		
13	Enterprise | Domain Architect (*)	Delivery Oversight							🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩
14	Solution Architect (*)	Solution Architectures							🟩	🟩	🟩															
15	Solution Architect (*)	Project Oversight										🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩

Figure 54: Delivery Plan Baseline (Y1-Y2)

It is worthwhile restating that this plan constitutes a baseline that reflects the desired timelines and outcomes that have been expressed [mostly] by stakeholders. However, its viability needs to be tested once resources for each role have been identified, assessed, and secured, their capacities mapped against the work effort associated with the roadmap, and provision made for the impact of any significant inter-dependencies across activities. Concurrently, estimates of the work effort also need to be accurately computed for each deliverable, but this can only be done after inputs and delivery process parameters are known for most deliverables.