Enterprise Architecture

Program Start-up Primer Kit

Version 1.0

October 27, 2008

Sponsor: Administrative Office of the Courts, Washington State Courts

Contents

Executive Summary

The purpose of this document is to provide information needed to get the Enterprise Architecture Program started. It provides information describing enterprise architecture, the organizational benefits, business drivers, a proposed methodology and a set of fundamental principles.

Enterprise Architecture (EA) describes a comprehensive framework for information technology and business that supports the Administrative Office of the Courts (AOC) Strategic Plan. EA facilitates the application of information technology (IT) to business initiatives and objectives and aids subsequent change in an orderly, efficient manner by describing a direction for current and future activities, supported by underlying principles, standards, and best practices. EA effectively supports and enhances the business of the Washington State Courts and improves the ability to deliver responsive, cost-effective functions and services. The purpose of EA is to create a holistic perspective of information technology strategies and its infrastructure support for state programs and business functions. This perspective can be achieved through effective governance, practical and effective policies and standards, and best practices for designing, developing, and implementing technology systems.

Business drivers are external or internal influences that significantly impact and/or set direction for the EA program. The business drivers are the reasons why it is important to have a successful enterprise architecture program. This document has identified five? business drivers: business alignment, cost effectiveness, increased agility, interoperability between organizations, and leveraging existing assets. These business drivers will be prioritized so that EA activities can be focused on resolving the most important areas first.

The Open Group Architecture Framework (TOGAF) has been selected by AOC due to the close alignment with the AOC ISD needs and the vast amount of information and consulting services available. TOGAF was originally designed to support the Technology Architecture. Over its years of evolution, however, it has acquired many of the facets of a framework and method for enterprise architecture. TOGAF is an architectural framework. It enables IT users to design, evaluate, and build the right architecture for their organization, and reduces the costs of planning, designing, and implementing architectures based on open systems solutions.

Enterprise architecture principles provide the foundation for the Enterprise Architecture Program. The principles are the driving force for creating the standards and best practices that govern and provide direction to achieving the AOC Information Services Division (ISD) Strategic Plan. The EA principles are foundational in that they are used to create the policies, standards and best practices that guide the operational usage by the organization.

1.0 Introduction

Over the past few years, as software and systems engineering has matured, it has become accepted that there is a clear need for an "architectural view" of systems. This need has grown as a result of the increasing complexity of systems and their interactions within and between businesses. Furthermore, continued pressure to reduce IT costs and deliver real, quantifiable business benefit from solutions necessitate a clear understanding of how systems support and enable the business.

1.1 Document Purpose

The purpose of this document is to provide initial definition and direction in starting up the Enterprise Architecture Program. It summarizes what EA is, how it will be used, and identifies the primary business drivers. The document also establishes a proposed methodology to follow and provides foundational EA principles.

1.2 What is the Enterprise?

From a practical viewpoint, the Enterprise is comprised of agencies that interact with each other and whose business processes often share common characteristics. Agencies within the Enterprise also interact with external constituencies, including other governmental and private sector organizations. As these interactions become increasingly electronic, common standards for information technology that will apply to the entire Enterprise become necessary to ensure consistent, reliable and adaptive operations. Various external entities, though not formally part of the Enterprise as defined above, nevertheless interact with the State and are thus affected by the standards and practices adopted by AOC for its EA. These include courts, law enforcement agencies, other state agencies, justice and care providers and private or public-sector business partner organizations that provide services to the state government.

1.3 What is Enterprise Architecture?

Enterprise Architecture is a term used to describe the practice of documenting the elements of business strategy, business case, business model and supporting technologies, policies and infrastructures that make up an enterprise.

It is the highest level, widest scope, longest term kind of architecture related to how information systems support the business processes of an enterprise. Thus, it is distinguishable from solution architecture and software architecture or enterprise application architecture, though it shares some general principles and techniques with architecture at those lower levels.

It describes the logical organization of business processes and IT infrastructure. It reflects the integration and standardization requirements of the agency's(?) operating model. The primary purpose of creating enterprise architecture is to ensure that business strategy and IT investments are aligned. As such, enterprise architecture allows traceability from the business strategy down to the underlying technology.

The general practice has gradually come to include a broad category of activities that help decision makers to understand, justify, optimize and communicate the structure and relationships between various business entities and elements. Initially, EA focused on application architecture, data

architecture and technology architecture. Additional architectural elements were soon added. Among these latter practices are business architecture and process architecture.

Enterprise architecture is the formal organization of the components, structures and processes required or relevant to the attainment of the goals and visions invested or envisioned in an enterprise.

Often used in the context of information systems applications in an enterprise, enterprise architecture is really concerned with all aspects of an enterprise with information technology as a sub-context.

Enterprise architecture involves developing an architecture framework to describe a series of "current," "intermediate," and "target" reference architectures and applying them to align change within the enterprise. Another set of terms for these are "as-is," "migration plan," and "to-be."

These frameworks detail all relevant structure within the organization including business, applications, technology and data. This framework will provide a rigorous taxonomy and ontology that clearly identifies what processes a business performs and detailed information about how those processes are executed. The end product is a set of artifacts that describes in varying degrees of detail exactly what and how a business operates and what resources are required. These artifacts are often graphical.

Given these descriptions whose levels of detail will vary according to affordability and other practical considerations, decision makers can make informed decisions about where to invest resources, where to realign organizational goals and processes and what policies and procedures will support core missions or business functions.

A strong enterprise architecture process helps to answer basic questions like:

- Is the current architecture supporting and adding value to the organization?
- How might architecture be modified so that it adds more value to the organization?
- Based on what we know about what the organization wants to accomplish in the future, will the current architecture support or hinder that?

1.4 How will the Enterprise Architecture be used?

An EA is, in some sense a statement of philosophy. Like all philosophies, it must begin with assumptions about the present state and the desired future state. Thus, AOC Enterprise Architecture will:

- Support the business and program priorities of the courts.
- Reduce the total cost of ownership of information systems and the processes they support.
- Reduce the complexity of the technology environment.
- Provide the ability to quickly respond to changing needs.
- Facilitate increased interoperability.
- Capitalize on the AOC's existing investments in applications and technology.

1.5 Enterprise Architecture Business Drivers

Business drivers are external or internal influences that significantly impact and/or set direction for the EA program. Identifying and prioritizing these are the critical first step in creating an EA that best supports achieving the AOC ISD's strategic objectives. The following is an un-prioritized list of business drivers that are the primary reasons why the EA program is needed.

1.5.1 Business Alignment

Business processes in the courts and justice community was relatively stable during the pre-electronic age. With the introduction of available computer technology, the processes in the courts were typically driven by automated systems capabilities. As time progressed, the automated capabilities that at first aided productivity, became a restrictive force on business capabilities. Court business found it aligning how it operated based on the implemented technology. Inefficient applications resulted in fragmented work-arounds, redundant data management, and other non-optimized processes being performed to get the business of the courts accomplished. Enterprise architecture will provide the framework in which business needs can drive technology solutions.

There is a tremendous potential benefit of having business centric alignment. A typical business process alignment and integration can typically provide 20 - 50 percent increase in productivity and cost savings at the low end. In contrast, a typical technology-driven endeavor could provide 10 – 25 percent increase in productivity and cost savings at the high end. In addition, there are more opportunities

The shift from technology driving business to business driving technology will be a long and difficult one. It is a paradigm shift that many organizations are not prepared for. It will take considerable time and will encounter many difficult challenges prior to being fully realized.

1.5.2 Cost Effectiveness

The Enterprise Architecture will maximize the cost-effectiveness of its information technology efforts. The Enterprise will strive to reduce procurement, development, integration, and support costs associated with duplicative architectures and obsolete or unused technologies by providing a common architecture that is flexible, reusable and cost effective across the Enterprise. Migration of existing Enterprise applications to the common architecture will be gradual, with consideration given to cost and available budgets. Future Enterprise applications and technologies will leverage AOC's existing assets where available and applicable.

1.5.3 Reduce Diverse Complexity

The Enterprise Architecture will reduce the complexity of the AOC's information technology environment. Especially complex technology solutions – for example, "spaghetti code" or wildly diverse hardware platforms – make change especially difficult, if for no other reason than the fear of breaking the whole complex mess by changing some small inter-related part of it. As complexity is reduced, the ability of the Enterprise to adapt and change is increased. Reduced complexity should also reduce the cost of products and their support through the leverage of Enterprise-wide buying power. As a practical matter, this principle implies the need to impose and maintain the discipline to reduce the number of platforms, configurations, and products in the Enterprise, thereby reducing training and support requirements.

Reducing the diverse complexity will require simultaneously controlling the creation of new diversity and providing a migration path for existing technology to converge on the architectural target technology. This is a difficult proposition since both the existing and architectural target are constantly evolving.

1.5.4 Increased Agility

It has been said that every good plan is correct until the time of engagement. Any plan, however well conceived, requires adjustment to changing conditions. Implementing a plan as first conceived has the risk of creating not exactly what is needed. The creation of products and services that are not what was expected by the customer occurs more than desired with information systems. The cause of this is due to both the original plan and to not changing the plan according to current conditions. Agility is defined as the ability to change velocity or direction in response to stimulus. The ability to be agile requires that an organization has the ability to sense changing conditions, analyze alternatives, and change course in a controlled manner.

Increased agility is needed more than ever due to the rate at which change is occurring in technology and judicial systems. A rapid response in the correct direction will increase business value and overall customer satisfaction.

1.5.5 Interoperability between Organizations

The Washington State courts, law enforcement, justice and care agencies, legal representation, and the public interact in a complex array of events and activities. A simple event such as a speeding ticket would involve, at minimum: the recipient of the ticket, the law enforcement agency, the court of jurisdiction, and, the Department of Licensing. Additional information participants could possibly include: the Department of Transportation, an attorney, an insurance company, a third party payment processing company, a collection agency, or a driving school. Data that is created in one organization is often times needed by other organizations. The data is sometimes being transferred manually and can be inconsistent due to having multiple versions of the same information.

The computer information systems that track and process these events and activities are managed and operated by multiple organizations. The systems operate using different technology platforms and have varying levels of customized integration. The AOC has developed a large number of custom exchanges over the years. The exchanges are almost exclusively the exporting of data from the JIS data repository. There are several specialized data imports, such as the case data received from Seattle Municipal Court and attorney data received from the Washington State Bar Association. The customized integration of the AOC with other systems is costly and time consuming. There is a huge backlog of unresolved requests for data exchanges with the AOC. As a result, organizations that can afford to do so have created custom methods (e.g., screen scraping, Open Database Connectivity (ODBC), etc.) to get data from and to AOC data repositories. The local methods are not sustainable in the long term and will be negatively impacted as existing systems are replaced.

An architecture that allows the movement of data to and from AOC-managed data repositories will provide universal methods in which systems from different organizations can communicate. The sharing of information will:

1. Reduce data entry collection costs by eliminating redundant data entry.
2. Assist courts to enable better decisions.
3. Allow organizations to streamline processes that are people, paper, and time intensive, resulting in more efficient operations.

- Improving data quality that would improve efficiency and reduce errors.
- Reduce organizational liability by having timely and accurate data during key decision making processes.

1.5.6 Leverage Existing Assets

The AOC has a very large and diverse application, database and infrastructure support asset inventory. The applications contain business logic and rules that are used to support court operations. The database repositories contain both daily operational information as well as very valuable historical information. The xxx

1.6 How should the Enterprise Architecture be implemented?

A value-based approach to implementing EA is recommended in order to realize quick wins, most notably when the team is first being formed. An analysis of key questions that provides the most value in an organization should lead the Enterprise Architecture team towards their highest priority tasks. Teams that spend too much time documenting the plan, without providing real value to decision makers, will be at risk of being disbanded.

This EA implementation strategy is to create an initial architecture through a set of basic principles, then to expand those principles out to policies, standards, guidelines and best practices. This effort will be performed by both AOC staff and consultants. AOC will create an EA principles draft, then solicit consulting assistance to verify the principles and start-up the EA program. A Letter of Solicitation (LOS) will be published to acquire a consultant, verify the principles, and to assist with developing an EAQ Program Start-Up consultation Request for Proposal (RFP). The goal of the RFP will be to establish the methods and procedures to make the EA program fully operational.

The following rules will be used to guide the program implementation:

1. **Time is of the essence.** Provide the momentum needed for the organization to behave in "Internet time."
2. **Honest and constant communication is critical.** Personnel may experience confusion about goals, roles and responsibilities and feel unable to resolve conflicts and problems effectively.
3. **Deal with implementation considerations all along.** In developing architecture, key implementation considerations such as roles and responsibilities and span of control must be dealt with on an ongoing basis.
4. **Early definition of a candidate end state is a must.** The development of the solution is the result of top-down, management-led and bottom-up, stakeholder-driven processes.
5. **Prove that the architecture development process is fair.** Employing a "use case" approach to define concrete scenarios from which principles are derived by the project participants.
6. **Document all architecture decisions.** Provide traceability and the rationale of what is being done and why. The architecture development process and decision making should be clear and transparent.
7. **Manage change actively.** Models and blueprints do not help much with the actual implementation of the architecture over time. What is really important is to define the actions required to realize the architecture.

1.7 Enterprise Architecture Domains

The Enterprise Architecture will be multi-tiered to maximize flexibility, adaptability, and stability. In a multi-tiered model, the overall Enterprise Architecture consists of several distinct but highly interrelated sub-architectures, each of which can be conceptualized as having its own distinct architecture. These sub-architectures are:

- Business Architecture
- Information Architecture
- Technology Architecture
- Applications Architecture

1.7.1 Business Architecture

Business architecture describes the structure and behavior of a business system (not necessarily related to computers). It covers business goals, business functions or capabilities, business processes and roles, etc. Business functions and business processes are often mapped to the applications and data they need. “Business Architecture” is an architecture that structures the accountability over business activities prior to any further effort to structure individual aspects (processes, data, functions, organization, systems, applications, etc.). A Business Architecture arranges the accountabilities around the most important business activities into domains.

A high level definition of business architecture is that business architecture is a disciplined approach to creating and maintaining a set of business-owned information assets that serve as a blueprint for the planning and execution of strategy. Business architecture defines:

1. How a business is structured.
2. What it does (processes).
3. What it needs to perform to achieve its goals. This view of business architecture focuses on the capabilities (people, process, and infrastructure) that a business needs to acquire to deliver on its Business Strategy.

The main elements of the Business Architecture are “business domains.” They can best be looked at as “areas of accountability.” Within the Business Architecture, a high level description is provided of how the business processes are dealt with by these domains and which domain is responsible for specified business functions or objects. Thus: the main elements of business architecture are “business domains,” which are clusters of coherent business functions and objects (concepts), over which meaningful responsibility can be taken in business processes. A business architecture contains:

1. The layout of business domains (including their occurrences on various levels) and their assigned business activities and added value (“business case”).

2. The business functions and business concepts (high-level data descriptions) that these business domains need (and are responsible for) to perform their assigned business activity.

3. The high level business processes, which shows how these domains work together (interface) to achieve the organizational goals and strategies. Such business architecture shows higher level management how their strategy will be implemented in their corporation.

1.7.2 Information Architecture

Information architecture is the design of data for use in defining the target state and the subsequent planning needed to hit the target state. Data architecture describes the data structures used by a business and/or its applications. There are descriptions of data in storage and data in motion; descriptions of data stores, data groups and data items; and mappings of those data artifacts to data qualities, applications, locations, etc. Essential to realizing the target state, data architecture describes how data is processed, stored, and utilized in a given system. It provides criteria for data processing operations that make it possible to design data flows and also control the flow of data in the system.

During the definition of the target state, the Data Architecture breaks a subject down to the atomic level and then builds it back up to the desired form. The Data Architect breaks the subject down by going through three traditional architectural processes:

1. Conceptual - represents all business entities.

2. Logical - represents the logic of how entities are related.

3. Physical - the realization of the data mechanisms for a specific type of functionality.

In this second, broader sense, information architecture includes a complete analysis of the relationships between an organization's functions, available technologies, and data types.

Information architecture should be defined in the planning phase of the design of a new data processing and storage system. The major types and sources of data necessary to support an enterprise should be identified in a manner that is complete, consistent, and understandable. The primary requirement at this stage is to define all of the relevant data entities, not to specify computer hardware items. A data entity is any real or abstracted thing about which an organization or individual wishes to store data.

There are certain elements that must be defined as the data architecture schema of an organization is designed. For example, the administrative structure that will be established in order to manage the data resources must be described. Also, the methodologies that will be employed to store the data must be defined. In addition, a description of the database technology to be employed must be generated, as well as a description of the processes that will manipulate the data. It is also important to design interfaces to the data by other systems, as well as a design for the infrastructure that will support common data operations (i.e., emergency procedures, data imports, data backups, external transfers of data).

Without the guidance of a properly implemented data architecture design, common data operations might be implemented in different ways, rendering it difficult to understand and control the flow of data within such systems. This sort of fragmentation is highly undesirable due to the potential increased cost, and the data disconnects involved. These sorts of difficulties may be encountered with rapidly growing enterprises and also enterprises that service different lines of business (e.g. insurance products).

Properly executed, the data architecture phase of information system planning forces an organization to specify and delineate both internal and external information flows. These are patterns that the organization may not have previously taken the time to conceptualize. It is therefore possible at this stage to identify costly information shortfalls, disconnects between departments, and disconnects between organizational systems that may not have been evident before the data architecture analysis.

1.7.3 Technical Architecture

Technical architecture describes the structure and behavior of the technology infrastructure of an enterprise, solution, or system. It covers the hardware configuration, the infrastructure applications that run on them, the infrastructure services they offer to applications, the protocols and networks that connect applications and nodes. It addresses issues such as performance and resilience, storage and backup. Technical architecture also includes part of software architecture.

Technical architecture is a part of software architecture, which focuses on how to deal with certain aspects of the software engineering process. It allows us to design better systems by:

1. Meeting system requirements and objectives: Both functional and non-functional requirements can be prioritized as "must have," "should have," or "want;" where "must have" identifies properties that the system must have in order to be acceptable. An architecture allows us to evaluate and make tradeoffs among requirements of differing priority. Though system qualities (also known as non-functional requirements) can be compromised later in the development process, many will not be met if not explicitly taken into account at the architectural level.

2. Enabling flexible partitioning of the system: A good architecture enables flexible distribution of the system by allowing the system and its constituent applications to be partitioned among processors in many different ways without having to redesign the distributable component parts. This requires careful attention to the distribution potential of components early in the architectural design process.

3. Reducing cost of maintenance and evolution: Architecture can help minimize the costs of maintaining and evolving a given system over its entire lifetime by anticipating the main kinds of changes that will occur in the system, ensuring that the system's overall design will facilitate such changes, and localizing as far as possible the effects of such changes on design documents, code, and other system work products. This can be achieved by the minimization and control of subsystem interdependencies.

4. Increasing reuse and integration with legacy and third party software: An architecture may be designed to enable and facilitate the (re)use of certain existing components, frameworks, class libraries, legacy or third-party applications, etc.

1.7.4 Application Architecture

Applications architecture describes the structure and behavior of applications used in a business, focused on how they interact with each other and with users. It is focused on the data consumed and produced by applications rather than their internal structure. A well architected application or system means it is scalable, flexible, and secure. When multiple applications are developed, it is important that they have similar behavior and to the extent possible, reuse a common architecture so that they can satisfy many of the business drivers such as the reduction to diverse complexity and reduced costs.

Designing a distributed application involves making decisions about its logical and physical architecture and the technologies and infrastructure used to implement its functionality. To make these decisions effectively, you must have a sound understanding of the business processes that the application will perform (its functional requirements), and the levels of scalability, availability, security, and maintainability required (its nonfunctional, or operational, requirements). The goal is to design an application that:

1. Solves the business problem it is designed to address.
2. Addresses security considerations from the start, taking into consideration the appropriate authentication mechanisms, authorization logic, and secure communication.
3. Provides high performance and is optimized for common operations across deployment patterns.
4. Is available and resilient, and can be deployed in redundant, high-availability data centers.
5. Scales to meet the expected demands, and supports a large number of activities and users with minimal use of resources.
6. Are manageable; allowing operators to deploy, monitor, and troubleshoot the application as appropriate for the scenario.
7. Is maintainable. Each piece of functionality should have a predictable location and design taking into account diverse application sizes, teams with varying skill sets, and changing business and technical requirements.
8. Works in various application scenarios and deployment patterns.

1.8 Architecture Principle Definition and Usage

1.8.1 What is an Information Technology Principle?

An information technology principle is a statement that supports decision making. Principles can support decision making in areas such as:

1. Business Development and Prioritization
2. Project Feasibility
3. System Design
4. Funding Decisions
5. Hardware/Software Acquisition
6. System Design
7. System Replacement

1.8.2 Architecture Principles Definition

1. Are established rules that guide technology decision-making.
2. Provide the foundation upon which architectural solutions are built.
3. Are a source of criteria for evaluating and choosing among alternatives.
4. Should be kept simple and high-level.
5. Are statements of intent or purpose.
6. Reflect a vision of maximum use of ways to use technology to benefit the enterprise.
7. Describe preferred practices to be followed when implementing new or upgraded systems.
8. Provide the foundation upon which the enterprise architecture is built.
9. Should be based on the needs of the enterprise and on a vision of an achievable IT environment.
10. Principles are a foundation for the development of enterprise architecture.

1.8.3 Criteria for Developing Architecture Principles

1. Should be Understandable
    a) Should be succinct and unambiguous.
    b) Understandable to people throughout the enterprise.
    c) Must be worded to facilitate a clear understanding of the concepts and the intent behind them.
    d) Consistent usage of defined terminology.
2. Should be Complete and Consistent
    a) Should not contain major omissions.
    b) Should not have duplication, overlap, or contractions.

c) Should be consistent with higher level strategies.

d) Should form the basis for policies, standards, guidelines and processes.

3. Should be Enduring

a) Technologies implementing the principles may change over time, but the guiding principles should remain constant and clear.

b) Independent of technology; who, where, when, policies and procedures.

c) Vendor and product neutral.

d) Should have a timeless quality because they define a value system.

4. Should guide decision making

a) Should provide a clear basis the rationale of what is the best choice.

b) Should be enforceable.

c) Compliance should be clear.

1.8.5 Benefits of Architecture Principles

Agreement on architectural principles assists the AOC in developing an enterprise architecture that achieves the following benefits:

1. Consistency

a) Increases the consistency of IT related decision-making across the enterprise.

b) Provides common processes and methods.

2. Compatibility

a) Increases the compatibility of solutions developed in different organizations across the enterprise and outside the enterprise.

3. Leverage Assets

a) Enables the sharing/reuse of information assets and IT solutions (or components).

b) Enable improved IT asset management and investment.

4. Quality

a) Increases the linkage of the IT decisions to the business.

5. Productivity

a) Enables increased productivity and lower learning curve for users and developers.

6. Responsiveness

a) Improves flexibility of the IT infrastructure in responding to changing business needs.

b) Provides a rational process for managing the introduction of new technologies into the enterprise.

7. Communications

    a) Results in improved communications between business sponsors, IT users, and developers.

1.8.6 Principles – Examples

Example 1 – Minimal Platforms Principle

1) The number of computing platforms will be limited and should conform to those defined in the technical architecture. New technologies will be deployed in compliance with the IT architecture while leveraging legacy systems and technology investments to meet the needs of the enterprise.

Rationale:

1) Simplicity fosters efficiency and clarity and lowers costs. Multiple platforms and multiple vendors raise complexity; inhibit efficiency and clarity and raises costs.

Implications:

1) That there are platforms suitable for multiple purposes. That there are platforms that can scale across enterprise variations in transaction volume, application security and reliability requirements, etc.

Example 2 – Source of Decision Support System (DSS) Data Principle

1) The data used for decision-making will be made available in a timely manner, originate from a single authoritative source and be managed in order to insure repeatability and trace-ability. Timely, accurate, and complete decision-support information will be made available to authorized users through standard tools.

Rationale:

1) Since critical decisions will be made based upon DSS, it is imperative that the data and its origin is understood correctly, that an analysis of its summaries are possible and that it be available for re-analysis as needed over time.

Implications:

1) That there is a consistent process and procedure for the extraction of DSS data from authoritative sources. That an audit trail of extractions is maintained.

2.0 Enterprise Architecture Methodology

An Enterprise Architecture Methodology is needed to effectively implement an EA Program. The Open Group Architecture Framework (TOGAF) has been selected by AOC due to the close alignment with the AOC ISD needs and the vast amount of information available. Also, TOGAF is strongly supported by many consulting organizations that could be available to supplement AOC staff resources.

2.1 TOGAF

TOGAF was originally designed to support the Technology Architecture. Over its years of evolution, however, it has acquired many of the facets of a framework and method for enterprise architecture. As of TOGAF Version 8, these different facets have been integrated, and TOGAF has undergone a major redevelopment, with the result that it is now a fully-fledged enterprise architecture framework.

2.2 The Role of TOGAF

TOGAF in its Enterprise Edition remains what it has always been, namely an architecture framework - a set of methods and tools for developing a broad range of different IT architectures. It enables IT users to design, evaluate, and build the right architecture for their organization, and reduces the costs of planning, designing, and implementing architectures based on open systems solutions.

The key to TOGAF remains a reliable, practical method - the TOGAF Architecture Development Method (ADM) - for defining business needs and developing an architecture that meets those needs, utilizing the elements of TOGAF and other architectural assets available to the organization.

A number of enterprise architecture frameworks already exist and are widely recognized, each of which has its particular advantages and disadvantages - and relevance - for enterprise architecture.

Although a number of enterprise frameworks exist, there is no accepted industry standard method for developing an enterprise architecture. The goal of The Open Group with TOGAF is to work towards making the TOGAF ADM just such an industry standard method, which is neutral towards tools and technologies, and can be used for developing the products associated with any recognized enterprise framework - such as the Zachman Framework, Federal Enterprise Architecture Framework (FEAF), Treasury Enterprise Architecture Framework (TEAF), and C4ISR/DoD Framework - that the architect feels is appropriate for a particular architecture.

The TOGAF ADM therefore does not prescribe any specific set of enterprise architecture deliverables - although it does describe a set by way of example. Rather, TOGAF is designed to be used with whatever set of deliverables the TOGAF user feels is most appropriate. That may be the set of deliverables described in TOGAF itself; or it may be the set associated with another framework, such as the Zachman Framework, FEAF, etc.

In fact, TOGAF has always done this: it does not prescribe a specific set of "architecture views," but describes an example taxonomy of the kinds of views that an architect might consider developing, and why; and it provides guidelines for making the choice, and for developing particular views, if chosen.

With the migration of TOGAF to an enterprise architecture framework, this flexibility becomes even more important. TOGAF is not intended to compete with these other frameworks; rather, it is intended to perform a unique role, in distilling what these other frameworks have to offer, and providing a generic ADM that can be adapted for use with any of these other frameworks.

The Open Group's vision for TOGAF is as a vehicle and repository for practical, experience-based information on how to go about the process of enterprise architecture, providing a generic method with which specific sets of deliverables, specific reference models, and other relevant architectural assets, can be integrated.

2.3 TOGAF and Architecture Governance

As governance has become an increasingly visible requirement for organizational management, the adoption of governance into TOGAF aligns the framework with current business best practice and also ensures a level of visibility, guidance, and control that will support all architecture stakeholder requirements and obligations. The benefits of architecture governance include:

1. Increased transparency of accountability, and informed delegation of authority.
2. Controlled risk management.
3. Protection of the existing asset base through maximizing re-use of existing architectural components.

4. Proactive control, monitoring, and management mechanisms.
5. Process, concept, and component re-use across all organizational business units.
6. Value creation through monitoring, measuring, evaluation, and feedback.
7. Increased visibility supporting internal processes and external parties' requirements.

In particular, increased visibility of decision-making at lower levels ensures oversight at an appropriate level within the enterprise of decisions that may have far-reaching strategic consequences for the organization.

2.4 Using TOGAF

2.4.1 TOGAF Overview

Two of the key elements of any enterprise architecture framework are:

1. A definition of the deliverables that the architecting activity should produce.
2. A description of the method by which this should be done.

With some exceptions, the majority of enterprise architecture frameworks focus on the first of these - the specific set of deliverables - and are relatively silent about the methods to be used to generate them (intentionally so, in some cases).

Because TOGAF is a generic framework, as mentioned above, and intended to be used in a wide variety of environments, it does not prescribe a specific set of deliverables; rather it talks in general terms about the types of deliverables that need to be produced, and focuses instead on the methods by which these should be developed.

As a result, TOGAF may be used either in its own right, with the generic deliverables that it describes; or else these deliverables may be replaced by a more specific set, defined in any other framework that the user architect considers relevant.

As a generic framework and method for enterprise architecture, TOGAF also complements other frameworks that are aimed at specific vertical business domains, specific horizontal technology areas (such as security or manageability), or specific application areas (such as e-Commerce). The concept of leveraging other relevant architectural assets in this way is known within TOGAF as the Enterprise Continuum.

2.4.2 The Enterprise Continuum

TOGAF embodies the concept of the Enterprise Continuum to reflect different levels of abstraction in an architecture development process. In this way TOGAF facilitates understanding and cooperation between actors at different levels. It provides a context for the use of multiple frameworks, models, and architecture assets in conjunction with the TOGAF ADM. By means of the Enterprise Continuum, architects are encouraged to leverage all other relevant architectural resources and assets, in addition to the TOGAF Foundation Architecture, in developing an organization-specific IT architecture.

In this context, the TOGAF ADM can be regarded as describing the process of moving from the TOGAF Foundation Architecture to an organization-specific architecture (or set of architectures), leveraging the contents of the Enterprise Continuum along the way, including the TOGAF Foundation Architecture and other relevant architecture frameworks, models, components, and building blocks.

2.4.3 TOGAF Architecture Framework Summary

TOGAF thus does not seek to compete with or duplicate other frameworks. What TOGAF does seek to provide is a practical, industry standard method of doing enterprise architecture - leveraging all relevant assets in the process - that is freely available and supported by a number of different architecting consultancies, and that is sufficient for an organization to use "as-is" or to adapt as the basis of an enterprise architecture method for other, deliverables-focused frameworks.

3.0 Governance

Enterprise architecture management includes the decision-making structures and processes needed to govern the overall Enterprise Architecture and the development and procurement of Enterprise systems. This section describes the governing structure to achieve the Enterprise Architecture and also the core day-to-day processes that must be managed in keeping with EA principles.

3.1 The Need for Governance

The EA will be planned and managed through a formal governance process. Architecture support and review structures shall be used to ensure that the integrity of the architecture is maintained as systems and infrastructure are acquired, developed, and enhanced. A structured review process will be needed to ensure that information systems comply with the IT Architecture and related standards, and also to determine the appropriateness of any proposed exceptions to EA standards. Processes incorporating these EA principles must be developed for all application procurement, development, design, and management activities – building upon the current “Intent to Purchase” reviews, but not limited to that process. The compliance process must allow for the introduction of new technology and standards, and also for the vetting and approval where appropriate of exceptions to standards. Conceptual Architecture and Technical Domain principles should be used as evaluation criteria for purchasing as well as developing software.

3.2 Architecture Review Process

The EA will include an Enterprise-level architecture review process, which will include representatives of organizations that vary in size, responsibility, and location, to oversee the alignment of Enterprise systems with the EA. This review process will evaluate the impact of new and continuing systems activities against the Enterprise Architecture principles. For example, the review process will assess the extent to which a proposed solution leverages and possibly enhances existing, already available, components.

3.3 Evaluation of New Technologies

The EA will be continuously reviewed to assess the potential impact, positive and negative, of advances in technology and industry trends. The integrity of the architecture will be maintained, but the implementation reviewed as technology evolves. Compliance to the EA principles must allow for the introduction of new technology and standards. An adoption process, incorporating conceptual and technical concerns, should be developed and followed for the introduction of any new technology standard. Caution should be used when acquiring any newly developed products or technologies, especially when we do not have control over the source code or design. A necessary corollary to this principle is that there will be a periodic review of technologies that have become obsolete and need to be phased out.

3.4 End-user Involvement

Relevant business area experts will be included in application development/acquisition teams, and end-users will be consulted throughout the application lifecycle. Success within most agency-based initiatives has always required collaboration between the program management, program specialists and IT staff. The business area or program experts are those stakeholders that are best equipped to provide feedback throughout the development process. Their involvement in any development team is thus essential. Clearly, participation in development teams requires both a time and energy investment in communication between the team members. This requires methods of communication and patterns of work that facilitate ongoing collaboration between end-user communities and IT staff. Joint teams of technology and business experts present a unique challenge when those teams arise from different organizations. Determination of an overall project management process that incorporates the needs and priorities of each involved organization is especially important to the success of multi-agency initiatives. The general increase in the use of previously built components described elsewhere in this document would make it easier for end-users to envision the system they will be using and to suggest improvements. This stands in contrast with the elaborate paper requirements process that preceded a long period of development for systems in the past.

3.5 Enterprise Impact Analysis

All decisions concerning acquiring, developing, and enhancing systems will include an analysis of the relative impact of the decision on the Enterprise. Applications and information are valuable state assets and must be protected. Enterprise business impact analysis will yield valuable information to assist IT decision makers in planning, acquiring, designing, developing, enhancing, and recovering systems, thereby ensuring the alignment of technology decisions with the mission of the Enterprise while protecting the integrity of applications and information.

3.6 Total Cost of Ownership

The Enterprise will adopt a total cost of ownership (TCO) model for information technology. Consideration of the total costs of ownership -- including those in affected agencies – associated with a system over its entire life span will result in significantly more cost effective system choices and will enable improved accuracy in the planning and budget decision-making across the Enterprise. Incomplete consideration of TCO can lead to idiosyncratic and “silo” solutions that became ineffective in a short time period. The Enterprise must agree upon a TCO model. The model should include the costs of acquisition, development, support, enhancements, training, operations, disaster recovery and retirement against the costs of flexibility, scalability, ease of use, and reduction of integration complexity. The TCO model should be used to inform the decision process, in conjunction with other criteria.

3.7 Intermediate Deliverables

The EA will promote solutions for large projects that include intermediate deliverables for end-users. IT initiatives are frequently large endeavors, with timetables often measured in years. “Big Bang” approaches to project delivery often fail because problems are not identified until it is too late in the life cycle to correct. Creative use of intermediate deliverables will reduce the incidence of project failure, improve communications between the customer and the supplier, and build momentum that can only

enhance project success. Identifying meaningful intermediate deliverables, particularly when deploying projects that interface with legacy systems, is a challenging task. Ensuring that this process does not slow down the process of deployment is an activity that must be monitored closely by both customer and the project team. A well planned and implemented architecture will make it easier to provide frequent deliverables because of its use of pre-existing components and foundation in common patterns across the Enterprise.

3.8 Software Engineering Practices

Enterprise applications will be developed using software engineering practices that are consistent with accepted industry standards. The principles outlined in this document will require consistent commitment across the Enterprise. Adhering to industry-accepted software engineering practices is a necessary part of this discipline. It also reduces training costs and improves quality assurance. In the longer run, it can provide the basis for benchmarks and other measurements. It is essential that appropriate engineering practices be described and promulgated as part of the EA, along with training of staff in these practices as required and consistent with available resources. Similarly, all third party developers will be expected to follow compatible industry standard practices.

3.9 Use of Industry-proven Technology

Enterprise applications and infrastructure will use commercially viable, industry proven, widely-used technology to the maximum extent possible. Use of industry-proven, widely-used technology allows for easier access to affordable skills and a large base of proven software solutions. It can reduce risk, and helps ensure robust product support. Wherever practical, the Enterprise should strive to implement commercial-off-the-shelf technology as a first preference over completely custom applications.

3.10 Open Standards

The EA will favor products and solutions that use open standards to facilitate interoperability between applications, systems and organizations. Open standards are technology specifications that are publicly available and affirmed by an industry-recognized standards body. The use of open standards that allow for interoperability between applications and vendor-specific products is essential for the Enterprise Architecture to be successful. Requiring that products selected for Enterprise systems support open standards will also help ensure the flexibility and adaptability of the EA.

4.0 Enterprise Architecture Principles

EA principles provide the foundational concepts upon which the EA Program will operate. This section describes the governing structure to achieve the Enterprise Architecture and also the core day-to-day processes that must be managed in keeping with EA principles. The Principles have been grouped into the following sub-sections:

1. Business Architecture
2. Information Architecture
3. Application Architecture
4. Technical Architecture

4.1 Business Architecture

The business process architecture deals with the goodness-of-fit fit between information systems and the business processes they are meant to facilitate. This includes business process analysis and, where appropriate and feasible, business process re-engineering. Goals include common solutions for business process needs shared by multiple entities within the Enterprise, development of business logic models and components that can be reused across multiple applications, and increasing the efficiency of Enterprise business processes.

4.1.1 Common Solutions

The Enterprise Architecture will facilitate common solutions for business processes and needs shared by the courts statewide. Where there are common business processes across multiple organizations in the Enterprise, the Enterprise Architecture will facilitate the development of common solutions. This is in contrast to the traditional process of merely converting a set of physical requirements, without further analysis, into a delivered (silo) application. The key is the designer's understanding of the process in concept, not how the users label it. To use a simple example, a software component that allows for the scheduling of a generic resource at a particular date and time could be part of many applications, as diverse as scheduling people or machines or whatever the user has in mind. In this view, the requirements for applications are built from the bottom-up, driven by business users, as before. However, the solutions are architected, in part, from a set of modular, but Enterprise-level, components which can be used by any agency within the Enterprise. This should help accelerate software development, while keeping costs down. Clearly, an inventory of these common solutions and components, as well as methodologies for maintenance of these components, will need to be established if this principle is to be achieved in practice. Appropriate mechanisms will need to be determined as part of overall EA governance.

4.1.2 Business Logic

The EA will facilitate Enterprise-wide business logic code reuse, smooth integration of Enterprise business processes, and maximum utility for end-users. In so doing, the business process logic must be designed, acquired, developed, or enhanced such that processes can be shared and integrated across the Enterprise and with our partners. As with other tiers, the implementation of business rules will employ reusable components across the Enterprise.

4.1.3 Business Process Analysis and Re-engineering

Business process re-engineering will be considered when defining requirements for new Enterprise applications. Technology is a critical tool to continuously improve government services. However, the return on technology investment will be limited if all it does is to automate inefficient and ineffective business practices. We envision business process improvements that result from the optimization of both human and technology resources. Business Process Reengineering (BPR) utilizing Information Technology as an automation tool will help reshape the way the Enterprise conducts business. Adopting a BPR model will reduce the total cost of ownership, increase efficiency, improve customer services, and reduce development times across the Enterprise. All Enterprise systems development must begin with BPR as an essential predicate to systems design. To the extent that such systems affect multiple entities

within the Enterprise, a common BPR model will be adopted to ensure all relevant business operations are reviewed and streamlined before development and design progresses.

4.2 Information Architecture

The information architecture deals with the modeling and use of the information assets of the Enterprise. “Information assets” includes both the information needed by the Enterprise to carry out its business functions, and also the information generated by the Enterprise. An overall goal for the information architecture is to maximize the utility of these information assets.

4.2.1 Information is an Enterprise Asset

The Enterprise Architecture will facilitate information sharing across the Enterprise to enhance and accelerate decision-making. Information itself is an Enterprise asset. Decisions and services are better if they are based on the appropriate use of information. Thus, the EA needs to facilitate the use of information by decision makers and program staff. It needs to protect information from loss and corruption. Data elements need to have compatible definitions and consistent coding across the Enterprise. The capacity must exist to share data appropriately across the Enterprise, however, always within the boundaries established by law. A traditional Enterprise asset management strategy is well understood and implemented for the various physical inventories found in an Enterprise. Where traditional asset management falls short is in recognizing that information is as much an Enterprise asset as any of the physical equipment. By addressing information as a valuable Enterprise asset at the outset, decision making, technology investment, and Enterprise-wide access to information can be accommodated. Following this principle will require identification and authentication of information assets as well as a unified information asset management system. The information assets will need to be structured for easy access and management without compromises in security, privacy, and confidentiality. Systems must be designed, acquired, developed, or enhanced such that data can be shared and integrated across the Enterprise and with our partners to the extent permitted by law. This will require the establishment of message protocol standards for common types of data records to facilitate information exchange.

4.2.2 Enterprise Data Standards

The Enterprise Architecture will promote Enterprise-wide data standardization, reuse, interoperability, and information management across applications and agencies. This principle implies the need for a central authority that maintains the standards to be used for such common data fields as name, address, etc. and also builds the appropriate meta-data that can enable the sharing of data. Industry data standards should be adopted whenever feasible to facilitate business transactions between local, state and national government entities and between government and the private sector. A formal process for the development of Enterprise data standards should be established as part of overall Enterprise Architecture governance.

4.2.3 Support for Data Warehousing

The Enterprise Architecture will support data warehousing and other information centric end-user computing. The Information Architecture must support both online transaction processing (OLTP) and data warehousing applications. However, these two classes of applications require very different data

models and make very different demands on database systems. On-line transaction processing focuses on quick updates of the data. Often, the speed of these updates can be dramatically slowed down by the processing generated by user queries. For this reason, it is best to separate the data warehouse from the OLTP. In so doing, we also provide for a more secure environment for both OLTP and data warehousing. To successfully support data warehousing, the Enterprise must ensure that IT staff understand the different database design and performance requirements of OLTP and data warehouse applications, and that there are sufficient numbers of staff skilled in key data warehousing methodologies such as dimensional modeling, in which data models are partially de-normalized to facilitate easy end-user querying. The Enterprise Architecture must include Enterprise-wide approaches to data warehousing, including methods for data warehousing that span agency boundaries as permitted by law in order to facilitate appropriate information sharing and integration, Enterprise-wide views of information assets, and elimination of unnecessary redundancies in data warehouse content, data storage, and end-user information product development. Recent advances in federated data warehousing, which allow for the integration through middleware of logically and physically disconnected data warehouse resources, should be investigated for their potential usefulness in this area in order to meet the operational demands of the Enterprise while at the same time ensuring privacy and confidentiality as required by law. In addition to traditional methods for querying the data warehouse, publish-subscribe methods of access can increase the utility of data warehousing applications. Publish-subscribe methods include such features as automatic end-user notification of changes and updates to data warehouse content in areas of particular interest that they are authorized to access, and automatic receipt of particular data warehouse reports that they are authorized to receive. These methods can avoid the need to have users make continual requests of the database in order to determine if anything has changed. Thus, publish-subscribe methods can help reduce the overall volume of end-user queries, and thereby mitigate potential performance degradation from increasing numbers of end-users. However, such methods must be implemented in ways that ensure there are no inadvertent unauthorized disclosures of protected information.

4.3 Technical Architecture

The infrastructure architecture addresses the underlying enabling hardware, software and network that support Enterprise applications. It also addresses the communications between architecture layers and between systems.

4.3.1 Interoperability

The EA will promote interoperability and integration across Enterprise applications. The more applications are interoperable, the more likely it is that the Enterprise can respond flexibly and inexpensively to new user requirements. By its nature, interoperability also enhances integration across the Enterprise, thus enabling the Enterprise to provide better, more coordinated services.

4.3.2 Partitioning and Decoupling of Application Components

The EA will favor solutions that are highly partitioned, modular in design, that are comprised of components that are maximally decoupled, and that use standards-based messaging protocols for communication between external and internal systems. An essential part of any strategy to reduce complexity and enhance flexibility in an Enterprise Architecture is to break down the traditional

monolithic systems and to reduce as much as possible the extent of the coupling of different components. In this way, any internal change or improvement in a component will not require massive changes in software to be found elsewhere in the Enterprise. The use of message-based communications is an especially useful tactic in the integration of legacy, silo systems into the Enterprise. This allows for the creation of a thin layer that wraps the legacy application, providing a more modern means for other software components to communicate with that application. Then the legacy system can be replaced in an orderly fashion, without that change causing further impact on the Enterprise. Such an approach makes possible the use of efficient, user-friendly techniques. In particular, we would encourage the use of publish-subscribe, rather than continual request-reply, modes of data acquisition by the user. This kind of modular implementation will allow for the upgrade, exchange, and reuse of products with minimal retooling or disruption to the overall environment. Modularity will reduce the complexity and upgrade time of IT assets while providing the Enterprise with geographical independence, skill-leveraging, and improved functionality. The implications of this approach include:

1. A culture of reuse and sharing of code and components must be further nurtured within the Enterprise.
2. Enterprise-wide component management will become a core competency.
3. Design reviews will become an integral part of IT decision-making.
4. Modular components will be shared across organizational boundaries, to the maximum extent permissible.

4.3.3 Uniform Standards

The EA will establish uniform standards for Enterprise technology. Standardization will facilitate consistency and uniformity across applications. It will simplify software design, reduce application development time, facilitate learning, improve system maintenance and support, and promote information-sharing among organizations within the Enterprise, and thus reduce total cost of ownership. By fostering the development of the same “look and feel” across the Enterprise, standardization will enhance end-user friendliness and satisfaction, and contribute to the identity of the Enterprise. As part of this effort, existing IT platforms must be identified and documented and compared to Enterprise-wide configuration standards, which in turn must be established. A review process must be developed for setting standards, reviewing and revising them periodically, and granting exceptions where appropriate.

4.3.4 Standard Configurations

The EA will define a small number of standardized, easily-reproducible system configurations for deployment across the Enterprise. Establishing configurations that are easily reproduced will cut down on costs associated with support and maintenance as well as simplifying training and knowledge transfer. This will also mean that any proposed changes must function correctly and consistently throughout the entire organization. Consistent configurations will help provide a similar look and feel to end-users, as well as being able to support users of all ability levels, from novice to expert. This principle also makes possible the end-to-end systems management that is a necessary part of reliable delivery of technology services to our users. This will require a change in some decision making standards. For example, we will deploy applications on uniformly configured servers. Proprietary software used in

standardized system configurations will be maintained at vendor supported version levels. For these reasons, there may be an initial increase in capital investment. In the short run, this means that we will strive to replace multiple, non-standard configurations with a smaller number of consistent configurations. In the long run, this means that we plan for the retirement and replacement of obsolete platform components and configurations.

4.3.5. Scalability

Enterprise applications and infrastructure will be scalable in size, capacity, and functionality. Scalable components will better adapt to the changing needs of the Enterprise and will be less likely to impede its growth. Scalability, in effect, will serve to improve Enterprise-wide integration, maximize resource utilization, and minimize duplication and application redundancy.

4.3.6 Conformance with Statutory Requirements and Policy

Enterprise systems will adhere to all applicable security, confidentiality, and privacy policies and statutes. This helps to safeguard confidential and proprietary information, as well protects its integrity. It enhances public trust and the proper stewardship over public information. In addition to the technical requirements of security, education on issues of privacy and confidentiality must become a routine part of normal business processes.

4.3.7 Enterprise-wide Authentication

The EA will use a single method of user authentication to control access to Enterprise applications and services. The security framework will eliminate multiple IDs/passwords, substituting a single login (user authentication via a common LDAP directory service). User authentication will be augmented to include dual-factor methods (e.g., RSA tokens, smart cards, biometrics, and certificates) for sensitive applications where such extra measures are necessary to achieve the required level of security.

4.3.8 Universality and Flexibility

The EA will support variable, application-specific security and data retention requirements. Security and data retention requirements will vary with the type of information and the Enterprise Architecture must support these various requirements. The types of information a system builds, edits and/or displays will dictate its security requirements. No matter what type of data, systems should be designed with security as an integral part. An application should make use of rights assigned by the user login to the environment rather than building access controls within itself. Because of the continued existence of legacy systems, the security framework should be designed so that it can be retrofitted to new architectures.

4.4 Application Architecture

The application architecture deals with the interface and interaction between Enterprise technology and the end-user. Goals include user-interface consistency across applications, in keeping with system requirements, and user-interface designs that maximize usability.

4.4.1 Standard Look and Feel

Enterprise applications will be designed with a standard "look and feel" to facilitate ease of use, software design simplification, and branding. As with other layers of the architecture, the presentation layer will make use of standardized, reusable components. Standardized presentation layer components will provide Enterprise Applications with a common, consistent end-user interface. The benefits of this approach in the presentation layer are similar to those in other architecture layers: it allows for the upgrade, exchange, and reuse of products with minimal retooling or disruption to the overall environment. It encourages deeper functionality and reduces training time, since users are more likely to have experience with the component from other applications.

4.4.2 Standard Set of Information Technology Tools and Services

The Enterprise Architecture will define standards for basic information technology tools and services (e.g., email, voicemail, internet access) that employees should have available to them, consistent with available resources and job functions. Employees across the Enterprise who perform relevant job functions should be given access to training tools and/or facilities to all basic services/applications, to the extent permitted by available resources. This will allow for greater volume discounts to be realized by the entire Enterprise, as well as having a larger pool of similarly trained users. It will also allow for an easier integration of new employees into agencies.

4.4.3 N-Tier Application Architecture

The Enterprise Architecture will employ an N-Tier Application Architecture. In software engineering, multi-tier architecture (often referred to as n-tier architecture) is a client-server architecture in which an application is executed by more than one distinct software agent. For example, an application that uses middleware to service data requests between a user and a database employs multi-tier architecture. The most widespread use of "multi-tier architecture" refers to three-tier architecture. The concepts of layer and tier are often used interchangeably. However, one fairly common point of view is that there is indeed a difference, and that a layer is a logical structuring mechanism for the elements that make up the software solution, while a tier is a physical structuring mechanism for the system infrastructure. At a minimum, applications should have a three-tiered architecture consisting of the following:

1. Presentation Tier - This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.
2. Application Tier (Business Logic/Logic Tier) - The logic tier is pulled out from the presentation tier and, as its own layer, it controls an application's functionality by performing detailed processing.

3. Data Tier - This tier consists of Database Servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.

4.4.4 Thin Client Solutions

The Enterprise Architecture will employ server-based thin client solutions requiring only network access and a web browser for end-user access wherever such solutions are technically appropriate. Through the use of server-based applications, thin client technologies (especially web-based clients), portals, and gateways, organizations can reduce the cost and complexity of all IT functions, making it easier to implement, deploy, manage, and monitor Enterprise applications and information resources. Server-based architecture provides the ability to rollout new applications and upgrades to the entire organization simultaneously.

Appendix A - Acronyms, Abbreviations and Definitions

Interoperability is a property referring to the ability of diverse systems and organizations to work together (inter-operate). The term is often used in a technical systems engineering sense, or alternatively in a broad sense, taking into account social, political, and organizational factors that impact system to system performance.

Taxonomy is the practice and science of classification. The word comes from the Greek τάξις, taxis (meaning 'order') and νόμος, nomos ('law' or 'science'). Taxonomies, or taxonomic schemes, are composed of taxonomic units known as taxa (singular taxon), or kinds of things that are arranged frequently in a hierarchical structure. Typically they are related by subtype-supertype relationships, also called parent-child relationships. In such a subtype-supertype relationship, the subtype kind of thing has by definition the same constraints as the supertype kind of thing plus one or more additional constraints. For example, car is a subtype of vehicle. So any car is also a vehicle, but not every vehicle is a car. Therefore, a thing needs to satisfy more constraints to be a car than to be a vehicle.