*Notes on:*

I/T Architecture in Action

By Richard J. Reese (2008)

Table of Contents

[Enterprise Architecture 2](#_Toc367046605)

[Principle-Based Architecture 3](#_Toc367046606)

[Architecture Governance 3](#_Toc367046607)

[Architecture Framework 6](#_Toc367046608)

[Application Architecture 7](#_Toc367046609)

[Information Architecture 7](#_Toc367046610)

[Network Architecture 7](#_Toc367046611)

[Platform Architecture 7](#_Toc367046612)

[Operations Management Architecture 7](#_Toc367046613)

[Service-Oriented Architecture 7](#_Toc367046614)

[Managing the Function 7](#_Toc367046615)

# Enterprise Architecture

## Enterprise Architecture Defined

“An I/T system architecture is the specification of components which form an information technology-based business solution. When the architecture spans business functions within a company, it is an enterprise systems architecture.”

Major companies spend between $500 million and $1 billion in I/T costs, with about 20% being spent just on I/T management. Nobody would put that much money into a project without knowing what they were going to build. The architects design the business’s I/T group in the same way that architects design businesses. A company’s technical footprint can be viewed as a reflection of technology decisions made over time, and ideally those decisions will be made with a long-term strategy and plan in place.

## The Case for Enterprise Architecture

A business has different groups with different goals, each requiring I/T support. Enterprise architecture exists to design the I/T part of the company in a way that can support the various business groups. This encourages re-use of I/T components and functions within a company and helps resolve resource conflicts that arise between business groups.

## The Architecture Maturity Curve

A company’s I/T architecture can be focused on short-term, tactical solutions or long-term strategy. It will generally fall somewhere on the following scale:

* No formal architecture
* Architect standard exist
* Architecture review board established
* Design reviews conducted
* Executive steering committee formed
* Business principles established
* Business & I/T strategies merged

In the first four, the company is focusing on the “total cost of ownership” (TCO) for I/T functions to make sure that each expense is contained and profitable to the business. Once the executive steering committee is formed, the architecture group begins to focus more on long-term strategy.

It is also important for I/T budgets to grow in line with the business so that they can properly support business activities. Typically, I/T receives an initial investment which is too large for what a start-up company really needs. If that budget is kept fixed, the company will one day exceed its I/T capacity and be underfunded. At that point, I/T must grow to be able to service the business adequately.

# Principle-Based Architecture

# 

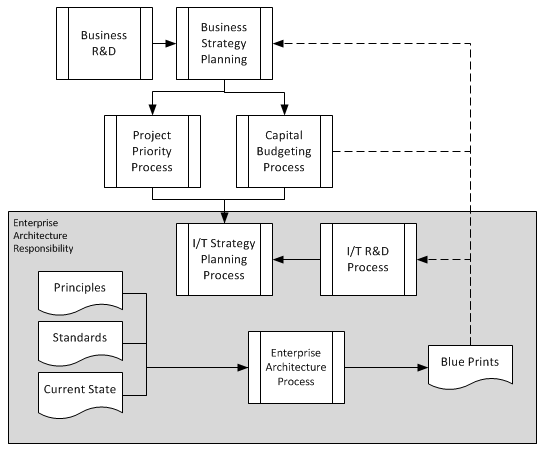
## Business Strategy Planning

Businesses can organize themselves in 4 primary ways:

* Holding company – Owns several subsidiary companies. The companies all run independently. They report finances separately and will not have any I/T systems that integrate with each other.
* Federated company – A company consists of smaller groups (almost like subsidiaries) which are mostly independent but may still have some shared services. The smaller groups will still want to be somewhat independent so they can be spun off if needed.
* Line-of-business – The company has several lines of businesses which run as separate groups but with several shared resources. Business lines cannot easily be spun off to run as separate companies.
* Core business – Every part of the company is completely integrated and will never be spun off. These parts of the company are necessary for profitability and market share.

A business may even use a mix of these strategies, but it is important to identify which parts of the business fall into which category and to match I/T decisions with them. A holding company will keep I/T systems highly separated in its subsidiaries, whereas they may be more integrated in the core business. The core business will also likely be focused on cost-cutting and total cost of ownership. Lines of businesses may be more opportunistic and able to experiment with new technologies.

## I/T Strategy Planning



## Architectural Model

The I/T architecture must be matched to the business architecture:

|  |  |
| --- | --- |
| Business Model | Enterprise Architecture Model |
| Holding Company | Data Interchange Architecture – Each company has its own I/T systems. These systems only communicate with each other through external messaging systems (EDI, VPN) |
| Federated Company | Loosely Coupled Architecture – Most of the I/T systems are run from the federated groups. However, some shared systems do exist. |
| Line-of-business | Tightly Coupled Architecture – Most of the I/T systems are run centrally. Systems can integrate with each other and exchange messages/services in real-time. |
| Core Business | Integrated Architecture – Databases and software can be shared amongst all parts of the company. |

## Architecture Principles

Principles provide a policy for how I/T systems will work within a business context. They are a set of guidelines or even a contract for how the I/T group will make decisions internally to support business processes. While these can differ from company to company, sample principles are:

* Centralization Principle – Enterprise-wide architecture is set by a centralized authority in the company
* Portfolio Principle – Each business system is considered to be part of an “ecosystem” of applications. (Applications must conform to the enterprise-wide architecture. Teams must also know how their application fits into the “big picture” of the company’s I/T strategy.)
* Knowledge Management Principle – Data is a corporate asset. (Data must be inventoried, valued, protected, stored, and recoverable. Business owners should be identified for each piece of data and must sign-off when data is shared with others).
* Evolutionary Principle – Business systems are expected to evolve to full functionality. (Lifecycles and roadmaps should be presented to the architecture review board to make sure each system continues to operate within the overall strategy).
* Design Simplicity Principle – Simplicity wins over complexity yielding best “time-to-market.” (Existing components should be re-used, and components should be designed to be extensible to meet unknown functional requirements. Complexity should be hidden behind simplified wrappers or APIs)
* Time-to-Market Principle – Buy before build to gain time-to-market advantages. (A vendor solution with 80% functionality is better than a home-grown solution with 100% functionality. Internal solutions will only be preferred if a unique solution is more important that time-to-market).
* Extensibility Principle – Solutions will be naturally extensible through open standards-based interfaces. Vendor solutions that are “closed” and do not allow access to their internals could be rejected for use within the company.
* Enablement Principle – Investments in infrastructure are necessary to enable long-term delivery consistency. (These investments allow systems to talk to each other and will offer long-term business opportunities even if they are not profitable in the short-term)
* Open Standards Principle – Externally defined technology standards are used as the basis for internal decisions. (This includes ANSI, W3C, ISO, etc)
* Economy Principle – Technology supports business initiatives and does not stand alone. (Supporting the overall business is more important than supporting a single business group. Systems that put an unusually large burden on the I/T group must have their costs paid for by the business group that it supports).
* Fiduciary Principle – Systems are secure and recoverable. (Enterprise security standards must be met and business groups must set availability/recoverability requirements for each system).
* Resource Sharing Principle – All resources are company owned and shareable across line of business or department. (Business groups should be able to re-use existing I/T systems, and anything that improves business productivity should be pursued).

## Principle-Based Architecture

The goal of principle-based architecture is to help business and I/T strategies align. It is the process and discussion around defining these principles that really makes this occur. Discussion should begin with middle-managers. Once consensus is reached, they can be shared with executives (and probably re-drafted again). Then they can be circulated throughout the company. The goal is to get these principles to become active in all parts of the company and become the “invisible hand” which keeps I/T decisions aligned within the company.

# Architecture Governance

## Architecture Governance Process

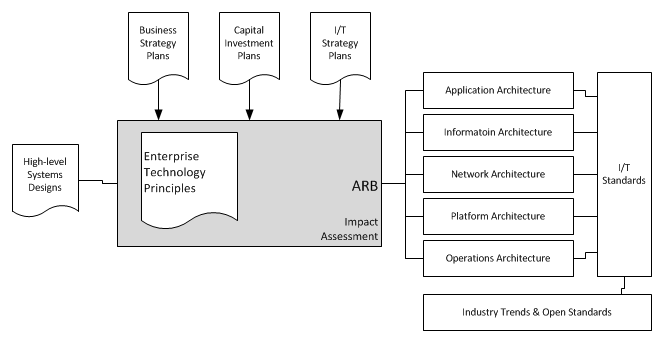
Three steps to manage a company’s technical architecture are:

1. Develop and maintain the “constitution” in the form of principles
2. Establish a “legislative branch” (The Architecture Steering Committee) to set policy, establish direction, and ratify standards.
3. Create and run the “governance body” (The Architecture Review Board) to recommend compliance and suggest changes to standards.

The Architecture Steering Committee should consist of the CIO, Application VPs, and Infrastructure VPs. It can meet monthly, quarterly, or as needed. The real day-to-day work is done by the Architecture Review Board. This group should include all parts of I/T, including operations, architecture, and application/functional areas. The review board should review high-level architectural design plans, answer questions from other groups, and provide input to projects and other application designs. Ideally, this group would function as a “Virtual CTO” and accomplish the following:

* Understand where the business is heading and coordinate business plans with technical decisions
* Commission the development of high-level roadmaps for each major architectural layer (application, information, network, platform, operations).
* Comment on the overall fit of proposed projects into the technical direction
* Report gaps to upper management and response to requests from internal audits

When possible, the actual CTO should also chair the review board.



## Governance and Managerial Behavior

While it can be difficult for the architecture groups to justify their existence (since they don’t produce any actual applications), they should make their work visible by measuring the following:

* Number of high-level designs that were reviewed
* Number of adverse opinions (rejected architectures) published
* Number of designs approved
* Number of designs approved with exceptions to company standards
* What costs were saved by re-using existing solutions?

# Architecture Framework

# 

The architecture framework is composed of 5 layers that together provide the backbone of an I/T organization. Just like the steel columns in the middle of a building, a strong architecture is necessary to support the organization. The 5 layers (which will be detailed in later chapters) are:

|  |  |
| --- | --- |
| Application  Architecture | Software and applications are the most visible part of the architecture framework as they are used directly by the business and support business functions. A company’s applications consist of both new innovations and decades of “legacy” applications. Some of the large trends in software architecture include:   * Batch processing (1960’s and 70’s) * Online transaction processing (1970’s) * Database oriented computing (1980’s) * Distributed computing (1990’s) * Internet/Web 1.0 Processing (late 90’s) * Wireless & Web 2.0 (2000, onward) |
| Information  Architecture | Information technology produces only one thing: information. Information is delivered to both people and other systems. Companies must manage their information by keeping an inventory of it and making it discoverable and re-usable. |
| Network  Architecture | Networks are the delivery mechanisms for information. They have evolved from WANs to LANs and new innovations are being made in wireless and mobile networks. |
| Platform Architecture | Platforms consist of the raw hardware that enables computing. This includes hard drives, processors, and memory. The speed of a company is largely determined by the speed with which it moves information through its systems. |
| Operations Management Architecture | Operations management defines the processes that “keep the lights” on in an I/T company. These processes are in place to make sure that systems maintain high availability and that failures are either rare or at least hidden from business users. This includes:   * Problem management * Change management * Configuration management * Asset management * Capacity Planning |

# Application Architecture

## Applications Run Industries

An example is given of two companies (airlines and credit cards) where legacy systems are still determining the pace of business. Airlines have developed a simple, message-queue system which processes important data in real-time. Duplicate systems run in parallel such that if any fails (or if one is undergoing maintenance) there is instant failover and the remaining systems can run the entire business. Less important activities are delegated to “offline” systems with less reliability, which only update the master data copy in batches. Credit card companies run batch processing systems which update their master account data overnight. The speed at which banks and credit card companies can operate is still largely determined by how quickly this overnight processing can occur.

## Evolving Application Architecture

Application architecture is always changing. Newer systems are being driven by Java and .NET and the open source communities around them. Legacy systems often exist side-by-side with these more modern constructions. New languages and technology standards are also evolving outside of the company, and will be important for I/T groups to keep up with these changes and be a part of these communities. There are 3 frameworks driving application architecture today and into the future:

|  |  |
| --- | --- |
| N-Tier Architecture | Applications can be designed to use the following tiers:   * Presentation Tier * Business Logic Tier * Communications Tier * Database Tier   Designing them in this way allows changes to be isolated in each level. A new presentation can be put on an application without changing the underlying layers. Databases and business logic can also be re-used by other applications. |
| Web Services and the N-Tier Architecture | Web services allow applications to expose their business logic over the network. SOAP/XML standards allow front-end applications to be written in different languages and deployed in different environments (PDA’s, phones). They also make the presentation layer “light weight” since it doesn’t need to encapsulate complicated business rules. |
| Wireless Access and the N-Tier Architecture | Wireless networks and mobile devices are allowing people to take work to new locations. Users can also be “online” at all times and able to communicate with each other. An N-Tier architecture with web services exposing the business logic layer are key to allowing I/T groups to easily transition to these new devices and environments. |

*Enterprise Resource Planning (ERP) Architectures*

ERP systems centralize business data. They are normally third-party applications that can be applied to almost any business with payrolls, accounting functions, and inventories. These systems allow companies to:

* Centralize key data and provide a “single source of truth” for the business
* Provide better reporting to the business
* Integrate supply, demand, and manufacturing chains
* Reduce overall I/T costs

An ERP system will be successful if it encapsulates business logic (without letting other systems duplicate or modify this logic). It also must integrate with other applications in the company. Enterprise Application Integration (EAI) frameworks have been developed to help move data into and out of ERP systems. Web services can also provide access to the ERP system so that it can be used by other systems in real-time.

## Application Service Provider Model

Application Service Providers (ASPs) emerged in the 1990’s among the “dot-com” bubble. They offer external hosting of web sites, data systems, and processing systems for companies. These can be low-cost alternatives for startup companies that don’t want the complexity of managing their own systems. However, as a business grows it will likely need to manage its on systems. The goals of an ASP (filling systems as close as possible to capacity and using low-cost hardware for as long as possible) are often contrary to the goals of businesses.

# Information Architecture

## The Physics of Information

Anything that can be digitized can be considered as “information,” and the amount of information in our world is growing at an explosive rate. Information has the following properties:

* Identity – how is the information referenced?
* Subject – what is the information about?
* Validity – is the information from a trusted source?
* Freshness – is it out of date?
* Context – is it part of other information?
* Value – does it yield benefits to information consumers?
* Dimension – is it complete?
* Location – where is it kept, and how can it be found?

## Information Management

To manage information, a company must keep a “metadata” repository that helps answer these questions. Information is the product that I/T produces, and as such an inventory should be kept. (No other business would consider running itself without an inventory of its products). Databases usually have built-in metadata tables, and other solutions exist to inventory information outside of RDBMS’s. Values that a metadata repository should hold include:

* Logical (business) name of data field/element
* Description
* Data type (numeric, character, date)
* Source of data field
* Target of data field
* Data steward identifier (who controls it)
* Encryption/security indicator (is this data required to be encrypted?)
* Valid values (list of allowed values, if applicable)
* Date entered into repository
* Date (and user ID) of last update

It should also be easy to import/export data to/from the repository.

## Data Governance

A committee similar to the architecture review board should be established to help manage a company’s data. This group should be chaired by the CTO and consist of data managers, data stewards, audit, and security representatives. The data stewards are key to the success of this group. The stewards are the controlling authority for different pieces of information. They are normally from the business (non I/T) and are responsible for naming, defining, and managing access to data. Authorization to access data must be granted by the stewards and access rights are normally maintained in an LDAP system. Stewards are also responsible for determining if data can be shared with external companies and setting requirements around whether data must be encrypted or whether groups/companies with access to the data must be audited. Most importantly, data stewards should be able to put data into its context, understanding its meaning in terms of business operations.

## Data Architecture

Data can be grouped into four types by its use:

* Operational data – primary data created by a business process (automated or manual) or used in business operations.
* Analytic data – data used to perform business analysis. This can be generated from operational systems or it may be obtained from outside the company.
* Reporting data – data used to provide reports to information consumers
* Archival data – data saved for any reason which can be recalled by the business when needed

The type of data determines the type of system that stores it. Operational data is stored on high-speed, high-availability (i.e. expensive) media. Operational Data Store (ODS) can also be used to support operational systems. ETL/ESB system may be used to copy data to/from the ODS system. Analytic and reporting data does not need necessarily need to be stored on high-speed media. Data is usually loaded into analytic and reporting environments by ETL (Extract, transform, load) processes and may be copied or reformatted to be more useful to the business. Archival data may be large in volume and can be stored on low-performance tapes or even offsite.

# Network Architecture

# 

## Evolution to a Single Standard

Since the development of computer networks, standards have emerged to allow many different types of systems to communicate with each other. TCP/IP is the key enabling technology here (although variations on how TCP/IP is used can optimize network communications for either speed or reliability). Ethernet technology has become the dominant technology for LANs. Voice over IP (VOIP) is also set to become a disruptive technology for telephone communication companies.

## Wireless Comes of Age

Wireless network protocols are allowing mobile devices to connect to these networks. GSM, CDMA, 3-G, and 4-G standards are emerging. Mobile apps are also set to become a new trend in computing. The JME (Java Mobile Edition) environment and BREW SDK are the primary development kits for phones. Other important technologies include:

* RFID (used primary for inventory tracking)
* Near-field Communications (NFC) – used for payment transfers or secure messaging over distances of about 1 centimeter
* Bluetooth – for personal area networks
* Wi-fi – for home or business networks

## The Other Network

While everyone knows of the internet, the other most important network which remains hidden to most people is the credit and debit processing networks. These networks are designed in a way that encourages the primary credit companies to maintain a monopoly over them (described as the “chicken and the egg” problem – getting on the network is costly and requires updating all hardware on the network – which will only ever be done for companies that are already big and on the network). ACH (Automated Clearing House) is used to process payments from checking accounts. This presents much lower barriers to entry and lower transaction costs.

Some of the most promising innovations are occurring in the field of networking by combining different types of networks. Sending ACH transfers over SMS (phone text messages) is a recent innovation. Creating “eWallet” applications on mobile phones, linking them to the credit and ACH networks is also a new development.

## Security Architecture

Companies typically divide the network into 3 zones:

* Zone 1 – web servers (accessible to the public)
* Zone 2 – intranet sites (accessible only within the company)
* Zone 3 – databases (restricted to authorized users/accounts)

Firewalls generally separate each zone. VPN networks can also be established to allow businesses to communicate with each other. LDAP has emerged as the de-facto standard for managing authentication and authorization for users of these systems.

# Platform Architecture

## Types of Platforms

The term “platform” is used instead of “computer” to indicate that hardware must be combined with an operating system and other programs. Platforms can be divided into 4 groups:

* Uniprocessing – an OS running on a physical system with one processor.
* Multiprocessing – an OS running with several processors on one board
* Symmetric Multiprocessing (SMP) – multiple processors on multiple boards, but all sharing the same memory
* Mainframe Computing – A type of SMP
* Massively Parallel Processing (MPP) – Multiple processors on multiple boards, but each board has separate memory. These overcome hardware scaling limitations of SMP’s (which are currently limited to 128 processors per “backbone”) and can scale to thousands of CPUs. The fastest supercomputers in the world are MPPs.
* Grid Computing (grid) – Separate systems with their own hardware and Operating Systems, connected over the internet. Work can be distributed across the grid, but special software is needed and connection speeds are slow compared to MPPs.

## Scale-Up vs. Scale-Out

The problem platform designers run into is: whether to use large, fast (and expensive) machines, or to use several, smaller computers working in a grid. The two most important factors to consider when sizing a platform to a business use, is speed (measured in transactions per second), and memory (RAM) requirements. If a lot of RAM is required, you will need to “scale up” and use large SMP systems. If lower memory requirements exist, you can instead “scale out”, using MPPs or even grids. The author advises that “scaling up” should be chosen as the default over “scaling out” unless you are sure that the solution does not require real-time computing. Usually, capital must be appropriated before the details of the design are finished, and you don’t know if you are going to run into platform speed constraints until it is too late to avoid them. Resizing an SMP system (adding CPUs or RAM) to achieve real-time processing can be done without re-designing the system. Scaling out may introduce network latency, memory problems, and transaction management problems that will be difficult to overcome.

## Virtual Platforms

Virtual machines (VMs) help lower I/T costs by increasing hardware utilization rates. More applications can be put onto the same hardware, with different VMs and applications sharing resources. VMware and XenSource are leading solutions here and allow hardware to be re-allocated or re-assigned while the system is still running (without requiring downtime). Of course, systems that need to “scale up” for big problems may not be able to share resources in a VM environment.

Citrix is also a solution that helps virtualize software by running desktop applications on a remote server. Just like VMs, the goal of a Citrix admin is to increase the number of users per Citrix server to as high as possible without having users experience decreased performance.

## Designing for High Availability

While most systems have some downtime available for updates (usually 3 hours per week), outside of these windows business users expect the systems to be available for use. Critical systems with high availability requirements can be put into three representative groups:

|  |  |
| --- | --- |
| Critical Level 1 | > 99.999% availability  (less than 5 minutes downtime per year) |
| Critical Level 2 | > 99.99% availability  (less than 1 hour downtime per year) |
| Critical Level 3 | >99.9% availability  (less than 8.6 hours of downtime per year) |

It’s also useful to define:

* Recovery Time Objective (RTO) – How fast must a system be recovered in the event of an outage?
* Recovery Point Objective (RPO) – How current must the system be when it is restored?

Business critical systems that must be covered quickly and without loss of data usually require “active-active” mirroring of systems, providing instant failover in the event of an outage without loss of data. Ideally, the business would never even be aware of a hardware failure. “active-passive” designs can be used for less critical systems. These systems are backed up (or copied to a mirror) on a periodic basis. This could be done after each “transaction”, or hourly, or daily. The system could still be restored quickly (within minutes) and presumably brought back up to date by some batch process that may involve copying data from other systems or archives.

## Storage Architecture

The biggest consideration when determining the type of storage device necessary is to balance speed, reliability, and lifespan with cost. Low cost, less-reliable storage work perfectly fine for some use cases. The other big decision that needs to be made is between SAN and NAS storage:

* Storage Area Network (SAN) – Disks are managed in an array, attached by high-speed, fiber-optic cable, with SAN-switches to route traffic. Failover can be configured so that several disks can fail without any loss of data.
* Network Attached Storage (NAS) – Similar to SAN, but instead of the disks being managed by one set of hardware, the disks are managed by UNIX servers. One server acts as the “head”, directing traffic to the others. NAS systems are less expensive, but they are also slower. They can also be more complicated to manage, and if frequent changes are expected to be made, SAN might be preferred.

# Operations Management Architecture

# 

If I/T systems aren’t available, they aren’t useful. Operations management is concerned with keeping systems available for business users, and making the complicated orchestrations behind the scene appear simple to business users. Responsibilities for operations management include:

|  |  |
| --- | --- |
| Problem Management | Identify and prioritize problems, oversee response, and perform root-cause analysis. Problems are usually grouped into three types of severity:   * SEV1 – a major business service (perhaps a revenue-generating service) is unavailable and must be repaired immediately. * SEV2 – an important service is unavailable, but revenue is not impacted * SEV3 – a smaller, “nuisance” to business operations, which can go a few days before being resolved. |
| Change Management | Determines how and when changes can be introduced into the I/T environment. Since more frequent releases typically correspond with greater problems, changes may be limited and restricted to a change calendar. |
| Asset Management | Maintains a record of hardware and software in the organization. This is typically handled by accountants who are managing the expenses associated with these systems. A unique “asset ID” is also assigned by this group to every piece of hardware that is inventoried. |
| Configuration Management | Manages detailed information about systems, software, and interdependencies between these systems. This information can be used to troubleshoot problems and help architects manage solutions (specifically applications, networks, and platforms). A configuration management database (CMDB) system can be used to manage this information. Ideally, it will have an auto-discovery process to detect new systems on the network and record them. |
| Capacity Management | Ensure that systems are being utilized and that adequate capacity exists for them. Since monitoring systems in real-time is expensive, snapshots are usually taken to make sure that hard drives, processors, memory, databases, and networks have adequate utilization rates. |

The Information Technology Infrastructure Library (ITIL) is a standard set of policies/manuals for running I/T operations. (These can be helpful here, although they are sometimes criticized for not including architecture or strategy.) Typically, enterprise architects do not get heavily involved in operations although they may be helpful in establishing standards.

# Service-Oriented Architecture

“Service-Oriented Architecture (SOA) is a software design discipline that leverages open standards to provide reusable services matching the business model of the company.”

In many cases SOA has simply become synonymous with “web services,” but really it means much more. They key benefit of a service-oriented architecture is that it allows I/T services to be aligned with business groups and activities. Services provided by I/T systems can be grouped as follows:

|  |  |
| --- | --- |
| Foundational Service | A service that provides key functionality to a single business group or function. This service can contain business logic for that function. |
| Enterprise Service | A service that provides functionality to the enterprise across several business groups. Business logic must be centralized in this one place to avoid duplication. |
| Utility Service | A service that can be used by any part of the business, but does not contain any business logic. |

Grouping services in this way allows I/T to better manage these systems. Separate groups may support each type of service, and foundational services can more easily be aligned with the business groups they support. All services can also work together, communicating over the Enterprise Service Bus (ESB) so that each application can interface with the necessary data and business logic that it requires. However, the key to success is not simply building web services, it is aligning those web services with business functions by building specialized “service centers” to support various business activities.

Key terms/functions in SOA include:

|  |  |
| --- | --- |
| Service Adapter | Program that exposes data or business logic using agreed upon standards (SOAP/XML) and a “service contract” for its functionality. |
| Enterprise Service Bus | A logical (not physical) entity which determines how messages are routed between services in the company. This will likely be a network using TCP/IP. It may also extend into defining the types of web services used (SOAP, RMI) and the availability of these services. |
| Service Orchestration | When several web services must be used in a single business transaction, an application must manage the state of that transaction and orchestrate the manner in which it calls each individual service. |
| Service Registry | A system that documents the existence of all services in the company. This can use UDDI for dynamic service discovery. However, most companies avoid this and configure services when software is deployed (since dynamic service discovery can be slow) |

# Managing the Function

When an enterprise architecture group is put in place it should have a mission statement and charter stating its objective and function. The CIO will also need to determine the focus of the architecture group: is it standards and governance or strategy and innovation. These two focuses are polar opposites of each other, with the first focused on cost reduction, optimization, and continuing operations while the second is focused on new projects and technology improvements that may be highly disruptive/transformational to the I/T group. The CIO will also need to determine how “domain architects” and “application architects” who may not report directly to the head of enterprise architecture are grouped into the organization. There is no single answer here, and the best implementation of the architecture function in a company depends on the situation.

A sample EA group might be laid out as shown:

|  |  |
| --- | --- |
| Vice President / Director of EA | Reports to the CIO/CTO  Ideal Candidate: 10-15 years of experience. Leadership and communication skills. Mastery of all 5 architecture areas. Key influencer of technical decisions that affect the company on a long-term basis. |
| Director/Manager of  Governance  (2 resources) | Runs the ARB  Maintains website  Oversees standards & audits  Candidate: approves policies, standards, and procedures. Maintains enterprise architecture repository. Responsible for maintaining the website and communication. |
| Direct/Manager of  Solutions Architects  (4-6 resources) | Project support  Architecture overviews  Architecture assessments  Vendor architecture reviews  Candidate: serves as ambassador for EA to the software teams. Works closely with the business to define needs and make sure that these are translated into software solutions. Mentors other I/T developers on the set of standards the company uses to make sure they are implemented. |
| Direct/Manager of  Domain Architects  (4-6 resources) | Platform planning/strategy  Software architecture  Network architecture  Data architecture  Security architecture  Candidate: coordinates policy on all 5 architectural design areas. Makes sure that these are in alignment and helps mediate any conflicts that arise. Brings issues of conflicting or disconnected efforts to the attention of I/T managers. |
| Direct/Manager of  Emerging Technology  (2-4 resources) | R&D Lab  Proof of architecture  Emergency technology  Tools evaluation  Candidate: responsible for keeping I/T up to date and leading all proof-of-architecture (POA) concepts. Also responds to internal requests for information (RFIs) |
| Research Services Coordinator  (1-2 resources) | Analysis/studies  Trade show management  Competitor evaluation  Candidate: Single point of contact to research firms for I/T. Manages contracts with research firms and requests for research. Provides research summaries to I/T managers in the form of newsletters. |

Overall, EA is responsible for the following activities:

* Requests for standards exceptions
* Requests for architecture assessments
* Requests for support on new projects
* Vendor product evaluations
* Requests for design reviews
* Support with problem determination
* Requests for long-term technical plans

Because it can be hard to see the benefit of these activities, it is important to record the activity of the EA group. The number of standards exceptions, architecture assessments, project assessments, etc. that are conducted should be recorded. The value of systems retired, the value created by new architecture designs, and the number/benefit of shared services developed are all good measures to track as well.