Framework for IT Architecture

Recommended Guidelines for Establishing an IT Governance Model

Cook County CIO Roundtable
Agenda

- **Executive Summary**
- **Application Architecture**
- **Information Architecture**
- **Data Architecture**
- **Integration Architecture**
- **Network Architecture**
- **Platform Architecture**
- **Directory Services Architecture**
- **Systems Management Architecture**

Cook County CIO Roundtable
Executive Summary

- **Background and Vision:**
  - Aging systems, constrained resources, skills gaps, fragmented systems and processes.
  - Lack of IT Governance
- **Strategy:**
  - Establish IT Governance and Architectural Model
  - Refine Countywide IT Strategic Plan
  - Implement Disaster Recovery and Business Continuity Plan
  - Implement ERP
  - Refine Enterprise PMO
  - Establish Shared Services Environment
- **Building Blocks of IT Governance:**
  - Information
  - Application
  - Data
  - Middleware
  - Integration
  - Network
  - Platform
  - Directory Services
  - Systems Management

- **Definition:** Application Architecture identifies criteria and techniques associated with the design of applications for the County distributed computing environment.
- **Principles:**
    - Plan for extensibility and scalability.
    - Design application to reuse components.
    - Design applications to be highly granular and loosely coupled.
- **Best Practice:**
    - Design for the N-tier service oriented architecture.
    - Generalize application interfaces.
    - Assign responsibility for business rules to business units
    - Make business rules platform neutral
    - Implement business rules as discrete components.
    - Access data through business rules
    - Adopt coding standards.
- **Standards:**
    - Develop 3-tier or N-tier Applications
    - Isolate Customizations to Purchased Software
    - Avoid Common Gateway Interface for business logic or to publish information to the Web.

Cook County CIO Roundtable
Information Architecture (1/2)

- **Definition:** Information Architecture provides standards for accessing data for online analytical processing (OLA), including executive information systems (EIS) and decision support systems (DSS).

- **Principles:**
    - Information is one of the most valuable assets for making business decisions.
    - In general, there is no new data, but there is new information. Existing data from multiple sources is being transformed into intelligent and proactive information.
    - Decision-makers should not be overwhelmed with an excessive volume of unnecessary information.
    - Online transaction processing (OLTP) databases and online analytical processing (OLAP) information databases should have separate data storage areas.

- **Best Practice:**
    - Begin data warehouse efforts by addressing a specific requirement for a
    - specific decision support application, keeping growth and scalability in mind.
    - Identify specific requirements for data availability, freshness (i.e., live, 24
    - hours old, etc.), and recoverability.
    - Perform benchmarks on the database design before constructing the
    - database.
    - Allow only read only access to end users of data warehouses.
    - Direct all information queries against decision support databases, not OLAP databases. Conversely, operational transactions should be directed to operational databases only, not OLAP databases.
    - Store atomic-level data in the data warehouse in addition to summary data.
    - Perform periodic validity audits against the data warehouse information model to ensure a high level of confidence in the quality and integrity of the data.
    - Maintain a repository for every data warehouse.

- **Best Practice:**
  - Begin data warehouse efforts by addressing a specific requirement for a
  - specific decision support application, keeping growth and scalability in mind.
  - Use the data warehouse metadata repository to document the rules applying to data scrubbing.
  - Ensure data entry quality is built into new and existing application systems to reduce the risk of inaccurate or misleading data in OLTP systems and to reduce the need for data hygiene.
  - Move to Commercial off the Shelf data hygiene software.
  - During data warehouse design, determine the logic needed to convert the data, plan and generate the extraction and transformation routines, and quality assure the data populating the data warehouse.
  - Assess the source data that will populate a data warehouse for accuracy and quality.
  - If a vendor-supplied extraction and transformation product is selected, it should support the same metadata repository that supports the data warehouse. It should also support the physical data warehouse.
  - Replicated data should be read-only, except where business practices clearly allow inconsistencies.
  - Implement decision support and executive information applications using and N-tier application architecture.
  - There should be no ad hoc query access to OLTP databases.

- **Standards:**
  - When accessing relational databases, use the industry standard of ANSI Standard SQL.
  - Use ODBC from any data access programs rather than vendor-specific database access tools.
  - Implement a server-based ODBC solution rather than a workstation-based ODBC implementation.
  - Use domain name system (DNS) names for databases that are accessible via TCP/IP.

•**Definition:** Data Architecture establishes and maintains an adaptable infrastructure designed to facilitate the access, definition, management, security, and integrity of data across the County.

•**Principles:**
- •Design an adaptive data infrastructure.
- •Design enterprise data architecture so it facilitates sharing data across the enterprise.
- •Separate data sources for OLTP data and OLAP information.

•**Best Practice:**
- •Use and maintain a Centralized Metadata Repository (CMR) to store centralized metadata definitions.
- •Review the CMR for existing standard and proposed data elements before implementing a new database.
- •Use the CMR and a Metadata element review team to create centralized definition of enterprise level data and encourage the sharing of data across departments.
- •Identify authoritative sources for centralized metadata.
- •Establish a data infrastructure that can accommodate rapid changes in data models based on changes in requirements or DB technologies.
- •Centralize data that needs to be shared and current.
- •Design databases to be modular, business driven and aligned with application services, not monolithic.
- •Design for data to be accessed only by the programs and business rules owning the data, never by direct access to the database.

- **Best Practice:**
  - Implement a minimal amount of data access rules stored in the database as stored procedures and triggers to avoid vendor lock-in.
  - Use a MIS level interface engine for data sharing of legacy platform data or other data where the application source code cannot be modified or interfaced.
  - Each department should standardize on a common data-modeling tool for designing and maintaining all new database instances.
  - Centralize data that needs to be shared and current.
  - Design databases to be modular, business driven and aligned with application services, not monolithic.
  - Minimize the replication of data within operational applications by replicating only stable data when necessary and based on business requirements.
  - Design the data access infrastructure to support the transparency of the location and access of data by each application.
  - Design for data to be accessed only by the programs and business rules owning the data, never by direct access to the database.
  - Use ANSI-Standard SQL programming language to access a database.

- **Standards:**
  - Custom systems must comply with CMR standard data element definitions.
  - COTS systems supporting client-controlled data element definitions must comply with the CMR standard data element definitions.
  - Use Centralized Metadata Exchange Standards when exchanging data across departments.
  - Use ODBC or JDBC database access middleware when accessing a database.
  - Implement a server-based ODBC or JDBC solution as opposed to a workstation based.
  - Use a Service Broker for inter-department data sharing.

•**Definition:** Integration Architecture specifies how various automated applications operating on different platforms can effectively work together. Integration techniques should be used when new application systems need to access existing application systems, while maximizing the investment in existing systems and platforms.

•**Principles:**

- •An Integration Architecture addresses the correlating components of data interchange, business processing issues, and end-user presentation.

- •An Integration Architecture meets the needs of linking heterogeneous operational application systems while protecting existing investments.

- •When making integration decisions, the life span of the solution is a key factor.

- •Integration Architecture relies on middle service tiers such as interface engines, database gateways, messaging, integration services, XML and 3rd party tools.

- •Integration should be designed to minimize the impact to existing application systems.

- •Use Countywide technologies whenever possible.

- •Provide maximum flexibility to integrate heterogeneous systems when enhancing existing end-user functionality through the use of a middle service tier.

•**Best Practice:**

- •Design an integration solution that does not write directly to an operational database.

- •Use direct program-to-program interfaces for high transaction volumes.

- •Recommended priority of using components of application integration are interface engine first, middle ware systems second, direct program to program interfaces as third and last alternative.

- •Choose XML as a preferred mode for all application integration for new systems, wherever possible.

- •Developing the DTD/schemas can be a top down as well as a bottom up approach.

- **Standards:**
  - Clearly Define Application Interfaces.
  - The message structure must be documented.
  - The application must be able to transmit and receive messages using a client/server model.
  - Purchase line-of-business application software rather than custom developing it whenever possible.
  - Clearly define and publish DTD/schemas

Cook County CIO Roundtable
Network Architecture

- **Definition:** Network Architecture defines a common, uniform network infrastructure providing reliable and ubiquitous communication for the County distributed information-processing environment.
- **Principles:**
  - A single integrated wide are network (WAN) is the backbone of the enterprise architecture and supports a variety of communication requirements including voice, data, image, video.
  - Networks should be available seven days a week and twenty-four hours a day.
  - A Countywide network must be based on common, open, vendor neutral protocols.
  - User access should be a function of authentication and authorization, not of location.
- **Best Practice:**
  - Networks should be positioned for future growth in traffic and expansion of services.
  - Configure servers supporting mission critical applications to minimize service interruption
  - Include network expertise on the requirements and design teams.
  - Minimize data movement
  - Design network neutral applications
  - Consider the impact of middleware on network utilization.
  - When data has to be distributed to multiple points (e.g. software and content distribution), move it once and only once across each data link.
  - When designing distributed applications, make no assumptions about the speed of the network on which the application will be deployed.

- **Standards:**
  - The standard protocol technology is TCP/IP
  - The standard internet access technology is Domain Name System (DNS) and IP address assignments

Cook County CIO Roundtable
Platform Architecture

•**Definition:** Platform Architecture identifies hardware platforms and associated operating systems supporting the County business.

•**Principles:**
- Design servers with bias toward granularity in physical servers.
- Design mission critical systems without a single point of failure.
- Design all servers implementing a particular application, application suite, or tier within an application with binary compatibility.
- Utilize open, vendor-neutral systems standards, wherever possible.
- Design servers to allow multi-tasking and multi-threading.
- Design servers to be field upgradeable.

•**Best Practice:**
- Run mid-range application and database server on a 32-bit multi-tasking, multi-Threaded operating system, at a minimum.
- For reliability and ease of support, place each major application on a uniformly configured server. This may require that each major application be implemented on its own server.
- Consider normal anticipated future application growth when determining capacity requirements for server platforms
- Balance business adaptability and ease of systems management with server platform choices. However, when there is a conflict between business adaptability and ease of systems management, the business requirement for providing adaptability should have the highest priority.

•**Standards:**
- Run distributed application server on platforms supporting "open" operating systems.
- Make sure server platforms comply with third party certification.

•**Definition:** Platform choices also influence storage selection criteria such as capacity, transfer rate, and cost of ownership.

•**Standards:**

- •Make sure server platforms comply with third party certification.
- •Use either SCSI or FC-AL technology for the disk drive interface.
- •Use RAID with fault-tolerance in the storage subsystem
- •SAN based fiber channel technology for large scale storage deployments running mission-critical applications to be the defacto standard.

Cook County CIO Roundtable
Directory Services Architecture

- **Definition:** Directory Services Architecture identifies criteria and techniques associated with protecting and providing access to County information resources. It facilitates identification, authentication, authorization, administration, audit and naming services.
- **Principles:**
  - Apply a level of security to resources commensurate to its value to the organization and sufficient to contain risk to an acceptable level.
  - An accurate system date and time are essential to all security functions and accountability and must be maintained.
- **Best Practice:**
  - Implement a fault tolerant solution to provide 24-hour, 7-day availability to the enterprise directory.
  - Purchased applications and operating systems should be directory enabled.
- **Standards:**
  - Use County wide directory services infrastructure.
  - Integrate homogeneous directories into a single tree.
  - Use Service Broker services for directory functions.
  - Use Centralized Metadata Repository directory schema attributes and object classes.
  - Populate directory objects according to the minimum attributes defined in Distributed Computing Standards and Guidelines.
  - Use Lightweight Directory Access Protocol version 3 (LDAPv3) for directory access where strong security is not required.

- **Definition:** Systems Management Architecture defines the framework for efficient and effective management of the County distributed information processing environment in order to support and enhance the productivity of its automated business systems.

- **Principles:**
  - Business needs should have priority when making system management decisions.
  - Limit the amount of "unique" performance tuning to existing individual network components, particularly servers and desktops.
  - Increase capital investment when it offsets long-term support costs.
  - Utilize open, vendor-neutral standards whenever possible.

- **Best Practice:**
  - Equipment deployed in virtual data centers must be configured to facilitate remote management and support.
  - System management functions for the virtual data centers should be remotely performed.
  - Under a virtual data center concept, responsibilities of customers for systems management are limited.
  - System components should proactively alert in advance of failure including predictive capability.
  - Inventories of hardware and software configurations should be maintained real-time.

- **Standards:**
  - Use SNMP protocols
  - Use remote monitoring products
  - Conform to the desktop management interface standard