Information Technology Architecture Office | Alastair Thomson
NIHRFC0070 | OD/OCIO/ITAO
Category: Informational | August 15, 2011

# NIH at a Crossroads
# An Enterprise Application Architecture for the NIH

## v1.0

## Status of this Memo

This document provides information on proposed enterprise application architecture for the National Institutes of Health (NIH) and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

# Table of Contents

# 1 Introduction

## *1.1 Purpose*

The purpose of this informational NIHRFC is to provide a foundation for discussion. The Information Technology Architecture Office (ITAO) seeks to elucidate the business challenges that NIH currently faces, and then propose a high-level software application architecture that could meet those challenges. This document is primarily focused on a technical audience and as such summarizes the business needs rather than discussing them in detail. The detailed business analysis is presented in a companion whitepaper and readers with an interest in such an analysis should contact the ITAO to obtain this paper – enterprisearchitecture@nih.gov

## *1.2 Summarizing the Business Need – NIH at a Crossroads*

The NIH has seen a lot of change in the last 15 years; from a doubling in the base funding level to the move towards funding high-risk, high-reward, multi-disciplinary, clinical and translational research, the way we do business has changed as has the sheer volume of research projects seeking funding. Adding to these business-focused changes have been very high paces of technology change, including the move to electronic submission of grant applications, and the move of many activities to the web. We have also faced a variety business challenges that have required significant IT effort:

- The recognition that barriers to entry for new investigators were becoming too high and that we needed to adapt our processes to encourage the next generation of investigators.
- The improvements made to the peer review process
- The increasing pressure to demonstrate the health outcomes of the research we sponsor and the resulting need for reporting mechanisms to support this such as STAR METRICS.
- The reporting requirements in the NIH Reauthorization Act and the American Recovery and Reinvestment Act

If past experience is anything to guide us, this pace of change will not only continue, but also increase.

As we have responded to each of these changes, we have generally taken two approaches to our IT systems: we either adapted the existing systems to support the new needs, or we created new systems. The first approach has required us to extend our systems in ways that the original system designers never imagined and in many cases has stretched them to a point where further change carries high risk and high cost. The second has led us to systems that duplicate data, or have built with a variety of technologies that often result in a reduced ability to interoperate and can be problematic to manage. A change in one system results in the need to change all the systems that depend on it. This model has been used for so long that there are many cases where we no longer fully grasp all the interdependencies, and the need for change is only noted when a change in one systems results in a failure in another system.

Despite efforts of IT teams across the NIH that are at times heroic, this situation poses a real threat to NIH’s ability to do business effectively. If we are to continue to support this increasing pace of change, we must make fundamental decisions about how we build and maintain our IT systems. Thanks to the efforts of the eRA program, the IMPAC II system has been extended and modified repeatedly, in order to meet those changing needs; but, in doing so, the system has reached the limits of what the original design could achieve. It has already been extended far beyond what the original designers intended. As a result, the system has become increasingly difficult to modify without risking the introduction of new problems. Many of the eRA systems are therefore in need of re-engineering if they are to continue to meet the challenges described above with reasonable cost and levels of risk. The eRA program has initiated this process and is currently re-engineering their systems to reduce this difficulty. Many of the ideas presented in this paper should be credited to their efforts.

Throughout the NIH, new technologies are providing new opportunities for us to improve how we do business. For example, NIAID's electronic Scientific Portfolio Assistant (eSPA) has helped to assess where research should be focused, and where new research topics are emerging. A concept-mapping tool was applied by NINDS[1] to their research portfolio to help them understand their research portfolio and then to realign their organizational structure to the research being performed, and NHLBI is developing a grants portfolio management system based on advanced natural language processing to aid in the evaluation of the Institute's extramural portfolio. These and other innovative initiatives have led to significant improvements in ICs' ability to meet their goals. However, the integration of these systems into the existing infrastructure is complex and costly and is at a point where the complexity and cost pose significant barriers to adoption of these valuable tools beyond a single IC.

The NIH remains in an environment where however much the budget could reasonably be seen as growing, there will always be more worthy research proposals than the NIH can fund. This produces a pressure to make the research funding and management processes as efficient and effective as possible. A reduction in the administrative costs required to operate the NIH could result in more funding available for research whether that be made available as grants, or to increase the availability of staff or resources for the intramural program. The ability to manage and optimize the research portfolio can ensure that NIH is spending research dollars where they can have the most impact. Both of these requires that IT systems be adaptable and flexible enough to support continuing change, optimization of our business processes and new technologies with the dual goals of reducing the administrative burden on each research funding dollar, and ensuring that the research funding is targeted where it can be most effective.

The factors above suggest that the NIH needs greater agility, efficiency and effectiveness in managing research and research funding, but the following barriers to such improvements exist:

- Core NIH systems lack the flexibility in process and data required for business agility.
- The systems lack interoperability, thereby reducing the ability to analyze and manage the research portfolio to make the most effective use of the limited funds available.
- The cost and risk associated with integrating emerging technologies that could greatly benefit the NIH are too high.

This state of affairs occurs at a time when the NIH is investing significantly in the re-engineering of two of the NIH's major mission support systems – NBS and eRA. This planned investment presents an opportunity to address this isolation and inflexibility by re-engineering these systems using a new software design paradigm rather than the existing models that will perpetuate the current problems. Such an opportunity is unlikely to be repeated for at least a decade. Therefore, the NIH is at a crossroads: We can choose to take a path that will provide significant benefits both in terms of cost savings in the operations and maintenance of our IT systems, as well as an improved ability to respond to the rapidly changing needs of research, or we can take the same path we have always taken and perpetuate the issues we currently have into the future.

The NIH needs a new way of thinking about software development that will increase flexibility, reduce costs and risks associated with change, and allow for interoperability between central systems and the innovative systems being developed by ICs, as well as leveraging emerging technologies. This change will require a major shift in understanding of the concept of a software application to one in which processes are supported by orchestrated sets of reusable services, rather than by a monolithic system.

---

[1] http://www.nihmaps.org/

## 1.3 Previous Work

This whitepaper builds upon a foundation without which this paper would be mere speculation. This previous work has exercised many of the principles described herein, and provides a foundation in reality for what is proposed. These are referenced here and more detail can be obtained by contacting The Information Technology Architecture Office (ITAO).

- Current State business process models for grants processes developed by ITAO and subject matter experts from OER and the ICs in 2006
- Security business process models developed by ITAO in 2007 and 2008 primarily driven by HSPD-12 and the clearance and badge processes
- The “eRA Needs Assessment” completed in September 2008 by the Office of Extramural Research and ITAO together with the Center for Scientific Review which included strategy papers, a draft application architecture and a pilot for re-engineering the eRA modules supporting the application referral processes using business process modeling.
    - This needs assessment resulted in the development of the eRA “Evergreening” strategy currently being employed by the eRA program in the re-engineering of IMPAC II.
- Heuristic/”Could Be” business process models for grants developed by ITAO and OER for potential ways to improve the grants business processes
- Current state business process models developed by ITAO and contracting subject matter experts for Research and Development contracts
- The development and approval of Conceptual Data Models (CDM) for Grants and Contracts.
- Current trends in the enterprise IT world as described in a variety of reports from Gartner Inc. analysts including trends towards the use of Enterprise Architecture techniques coupled with Service Oriented Architectures (SOA) and Business Process Management (BPM) as mechanisms for improving the alignment between business needs and IT, as for improving business agility and flexibility.
- Previous proposals from the Enterprise Information Management Branch of CIT for strategies for improvement to the reporting and data warehousing capabilities of the NIH.

## 1.4 Scope

While the title of this NIHRFC states that it is proposed as an enterprise application architecture for the NIH, some explanation of the proposed scope is required.

The architecture proposed is intended to apply as broadly to the NIH environment as makes sense. Some areas of business clearly lie within the scope such as the management of grants and contracts, the administrative functions such as finance and human resources, and in some cases into the management and administration of intramural research e.g. the management of core labs.

Just as clearly, the architecture is not intended to apply to other areas such as instrument control, and patient care management – cases where failures have high impact and where a different architecture is appropriate.

There are however a great many areas where the architecture could be applied but the decision as to its use should be made on a case-by-case basis. Examples include patient record management, research data management and clinical data management.

## *1.5 Assumptions*

As with any document that seeks to set a target architecture, certain assumptions have been made. Broadly speaking these assumptions fall into two areas:

- Business Assumptions
- Technology Assumptions

### 1.5.1 Business Assumptions

- Budgets for the foreseeable future will be constrained to at best the current level and at worst will have significant decreases.
- The fundamental nature of the NIH as an organization that funds research with a component that conducts research will be maintained.
- The separation between extramural and intramural programs will remain.
- Congress will continue to require accurate reporting on NIH activities and that the scope of reporting and the level of scrutiny it will receive will likely increase.
- The structure of the NIH will remain largely as it is today i.e. a federation of semi-independent institutes and centers.
- Collaborative science will continue to expand its role in the NIH portfolio.
- The drive for faster translation of science from bench to bedside will continue and increase as Congress expects more results for less money.

### 1.5.2 Technology Assumptions

- Technologies will be adopted as they become well established in the market and that NIH will not, for mission critical systems, be on the “bleeding edge” of technology.
- Technology will advance for the foreseeable future much as it has advanced over the past three decades with steady decreases in the cost of computing power, steady increases in the capacity of CPUs, storage and networks.
- Technology is making a fundamental shift from centralized computing resources to distributed computing with the evolution of cloud computing a natural progression of that process.
- Security will be of increasing importance and the structures of software must be compatible with security concepts such as spread risk and multiple layers of security

## *1.6 Definitions*

Throughout this document terms are used that may have several meanings in different contexts. The following table provides the definition of terms used in this document.

| Term | Definition |
|---|---|
| Enterprise System | A system that serves all NIH ICs and that are funded via central budget |

| | |
|---|---|
| | managed by the IT Working Group and supported by a specific program, e.g, eRA, NBS, NED and ITAS |
| Central System | A system used by more than one IC and managed by a single IC. IAAs may be used across ICs to obtain access e.g. POTS (NINDS), CTDB (NICHD), FTS (NHLBI) and eSPA (NIAID) |
| Local System | A system developed by and for a single IC |
| Data Warehouse | A large scale database that contains data from multiple data domains, sourced from multiple transactional systems, kept up to data by an extract-transform-load (ETL) process, with temporal dimensions and meta-data management to handle schema changes. |
| Data Mart | A database that is optimized for reporting typically using techniques such as star and snowflake schema, pivot tables, OLAP cubes, conformed dimensions and materialized views. A data mart draws data from the data warehouse and provides access via reporting and business analytics tools such as Cognos or JReview |
| Business Owner (System) | The organizational unit(s) that is responsible for ensuring that an IT system meets the needs of the business. In some cases such as enterprise systems, the business ownership is distributed across multiple organizational units and a council, governance board, steering committee or user group becomes the business owner in order to represent potentially diverse needs. |
| Business Owner (Data) | The organizational unit(s) that is responsible for the management of a specific data domain. They are responsible for ensuring that the data needed by a business domain is available, reliable and consistent. |
| Data Steward | The organization unit that is responsible for the management of a data domain from an IT perspective on behalf of a business domain. E.g. eRA is the steward of data for the Extramural Program. |

# 2 Aligning IT with the business need

A central viewpoint taken in this paper is that there is no such thing as a general "best" architecture; there are only architectures that are more or less appropriate to the business needs of the organization for which they are designed.

The selection of an architecture for a specific business then becomes a task of assessing the appropriateness of a particular architecture to the business needs of the organization. In most cases, the assessment of appropriateness is driven not so much by the functional requirements the architecture must meet, as it is by the non-functional requirements.

Functional requirements describe what a system needs to do in order to fulfill the business need for which is it designed. They describe specific behaviors or functions that the system needs to support. A system to support peer review of grant applications and contract proposals for example, might require that reviewers can enter a score and a critique for a grant application or contract proposal.

Non-functional requirements specify other characteristics of the system beyond behavior and tend to be expressed as metrics. Using the example above, a non-functional requirement for the review system might be that it must to support 3,000 applications being scored over a two-week period.

While the functional requirement above can be met many different ways, a Scientific Review Officer might simply e-mail a spreadsheet to the reviewers to collect scores, the reviewers could work with an MS Access database on a shared drive to record their scores, or they could use an iPhone or web application to record them. Each mechanism described clearly supports the functional requirement for recording scores. However, some just as clearly do not meet the non-functional requirement of 3,000 applications in 2-weeks; some solutions would simply be untenable. While this is an extreme example, the same carries through for all non-functional requirements. While there are many ways of performing a function in a system, there are relatively few ways to architect the system that are appropriate for the non-functional requirements, whether those requirements are for scalability, reliability, security, platform independence, etc. Therefore, a clear understanding of the non-functional requirements which an architecture must meet is essential to identifying an appropriate architecture.

While the examples above are focused on the extramural program, the same applies to the systems that provide administrative support for the intramural program; whether they are in the tracking of bio-specimens or the approval process for protocols.

The following section enumerates some of the business challenges currently faced by the NIH and some of the non-functional requirements that flow from these challenges. These are dealt with in more detail in the business-focused companion to this document available for interested readers from ITAO.

## *2.1 Advancing Scientific Methods and Technology*

Scientific methods and technology have advanced to the point where the traditional models of having a single extramural Principal Investigator or intramural lab director running a project are no longer appropriate to the complexity of the science. Therefore, the NIH needs to support:

- Multi-investigator teams
- Multi-disciplinary teams
- Teams co-funded by the NIH and other agencies or non-profits
- Intramural teams collaborating across ICs, or with other research agencies
- Integration of multiple funding mechanisms in supporting a project or complex of inter-related projects

  - Extramural Grants
  - Intramural Projects
  - R&D Contracts
  - Inter-Agency Agreements
  - Others

Requires the architecture to support:

- Evolving funding models without significant re-engineering with each change
- Focus on driving funding models through flexible business rules
- Support for collaboration across boundaries of organizations
- Support user interfaces created from existing components, rather than through complex development

## 2.2 The Need for Integrated Portfolio Management

The growing complexity of research models and the inter-relationship between research initiatives makes management of the NIH research portfolio in a manner that integrates research across all ICs, and across the intramural and extramural programs essential.

Requires the architecture to support:

- Consistent definitions and other descriptive metadata for widely used data elements
- Clearly defined business owners for data domains
- Clearly defined IT data managers for data domains
- Integration of data warehousing and reporting across data domains
- Integration of non-structured data (documents, audio, video etc.) in reporting
- Support for evolving visualization tools

## 2.3 The Challenge of New Technology

Technology is continually presenting new challenges and benefits to the NIH. As new technologies appear, it should be relatively easy to evaluate, assess and incorporate into the NIH IT infrastructure.

Requires the architecture to support:

- Extensibility to allow new technologies to be integrated without major impacts to existing systems
- Low dependence on specific technologies
- Descriptions of abstract, technology-agnostic interfaces
- Functionality partitioned around the cohesion of business processes and data in order to limit the impact of change

## 2.4 The Need for Agility

Change is a natural part of the NIH world, and our systems need to be able to rapidly adapt to external change factors. Agility to support change requires that the architecture support:

- Flexible business process that are supported by equally flexible IT processes:
  - The ability to change who does what when and by what means
  - The ability to reconfigure processes for exceptions or special classes of business data
  - Support for the needs of ICs for variation in process
- Extensible data formats that allow for additional data elements to be defined for existing objects without major re-engineering including the ability to meet the needs of ICs specific to their unique mission.
- Elemental functionality to limit the impact of change

## 2.5 Budget Pressures

Within the constraints of reason no matter how much funding is made available to the NIH, there will always be greater demand for research funds than the funds that are available. This constraint provides a constant pressure on the administrative aspects of research funding and management to be as efficient and cost effective as possible.

Requires that the architecture supports:

- Rapid change to business processes for minimal costs to support experimentation with new process alternatives and optimization of processes by measurement of metrics
- Re-use of IC system components across the NIH to reduce cost and risk
- Interoperability to support component re-use
- Technology agnosticism to avoid costs associated with vendor lock-in

# 3 The Heuristic Future State Application Architecture

Based on the discussion of business needs above, the Office of the Architect is proposing an application architecture that is believed to meet the requirements derived from both current and projected business needs. The discussion below seeks to set some foundational concepts that have determined how the application architecture is described and evaluated for appropriateness to the business need.

## *3.1 Describing an Application Architecture*

Let us begin by discussing what is meant when we discuss “Application Architecture”. The application architecture is a component of the overall Enterprise Architecture and describes how software is structured to support the business needs described by the Business Architecture and Information Architecture.

The application architecture lies at a level of abstraction above “Solution Architecture” where specific technologies and design patterns are selected for implementation in order to meet a specific set of business requirements. Therefore this paper is not intended to define precisely how a solution should be architected in order to meet business requirements, it is instead designed to provide a framework within which a specific solution architecture can be defined that has the desirable characteristics described above and can do so with reduced cost, schedule and risk.

The proposed Application Architecture is focused on a view that is distinct from the traditional view of software “systems” that support an area of business or an organizational unit. Instead, the architecture focuses on a definition of software built by assembly of reusable software components that are partitioned according to the business value streams, that is that the components are aligned to the activities that produce value for the NIH, and the bounds of those components are directly tied to the bounds of the value stream components. This model will be elaborated further in section 3.5.4.3 below.

### 3.1.1 Technology Agnosticism

One of the significant goals of this paper is to remain agnostic as to specific technologies. This principle is central to allowing solution architects select the most appropriate technologies to meet their requirements. In practical terms it is likely that the abstractions and patterns defined by the architecture will lead to agreement that certain technologies should be used throughout the NIH where there is an enterprise need. This definition of specific products and technologies is an activity separate from the application architecture – and is in fact part of the Technology Architecture layer of the Enterprise Architecture.

There are some cases in this document where specific technologies are mentioned – XML and Web Services and their associated WSDL and SOAP standards for example. ITAO feels that these technologies are sufficiently abstract and independent of specific vendors, as well as having extremely high value that these are mentioned as mechanisms for implementing specific goals. In the case of XML, while there are other potential extensible data formats available, the wide acceptance of XML coupled with its flexibility lead the authors to conclude that we are safe in declaring that XML will be a primary standard for data interchange throughout the NIH and throughout the world.

### 3.1.2 Defining and Standardizing the Architecture

This paper is written in largely abstract terms. In order for the application architecture described herein to become of real, practical use, standards must be agreed in order to define how these mechanisms should be implemented. We have chosen to describe application architecture using abstract design patterns that fall into two distinct categories: Structural patterns that define how functionality is partitioned within the architecture and behavioral patterns that define how components of the architecture interact. The structural patterns provide for consistency of construction that ensures that components are developed with a level of granularity that is conducive to reuse, interoperability, development by composition (application assembly) and manageability. Behavioral patterns describe consistent ways in which application components can interact again aiding interoperability, evolvability and extensibility desired in the architecture.

As a means to building consensus across the NIH on these patterns, we will use the NIHRFC process for the development of Bricks and Patterns to define the standards requires as this model recognizes that different solutions may require different technologies.

## 3.2 Architectural Goals

Based on business drivers described above, the following architectural goals have been identified. Many of these can be categorized as the oft-repeated “need for flexibility”. The definitions below seek to make the term “flexibility” more precise, concrete and quantifiable. Table 1 below describes these goals and discusses the specific aspects of the business drivers they address. These goals are identified from sources including the NIH Service Design Principles[2], and publications from the Association for Computing Machinery (www.acm.org), particularly the ACM Software Engineering Special Interest Group (SIGSOFT) and the Institute of Electrical and Electronic Engineers (www.ieee.org).

[2] http://enterprisearchitecture.nih.gov/ArchLib/AT/IA/Integration/SDPrinciples.htm

Table 2 below indicates the specific architectural goals that contribute to the NIH's ability to meet each business goal.

**Table 1. Architectural Goals**

| Goal | Description | Rationale |
|---|---|---|
| Interoperability | The ability of the system to interact with other systems | The need for NIH systems to interact with other systems is growing whether they be systems provided by DHHS, other government agencies such as NSF or other research related systems within the NIH. The ability to interoperate with these systems will continue to increase |
| Evolvability | The ability of a system to be tolerant of changing technologies | Technology changes. A system should be tolerant of those changes whether foundation products change, or they are retired to be replaced by new products |
| Scalability | The ability of a system to increase the number of users, amount of data, number of transactions etc. it supports without negatively impacting system performance | The number of users of NIH systems has grown from a few hundred during the 60s and 70s to hundreds of thousands with the introduction of direct access to NIH for grant applicants, researchers and the public. This trend is expected to continue as more and more of the interaction between NIH and researchers and the public is supported electronically. |
| Reusability | The ability of the system to reuse components in different roles where the core functionality required is the same | Significant pieces of functionality required for research funding and management are required in many places in the business process. |
| Functional Extensibility | The ability to add functionality to the system with minimal cost and impact on the existing system. | Many business processes vary across ICs, primarily by adding IC specific processes on top of the common processes used by all ICs |
| Data Extensibility | The ability to add to the data entities and data elements managed by the system with minimal cost and impact on existing systems | As NIH's mission, policy and business processes evolve, the need for data to be captured, managed, analyzed and reported upon also changes, largely by extension – new data entities, data elements or new relationships. |

| Goal | Description | Rationale |
|---|---|---|
| Data Quality | The completeness, consistency, accuracy, accessibility, availability, security and reliability of data. | In order to manage the NIH, NIH's research portfolio or to conduct research it is essential that NIH staff have available data that is consistent, accurate and reliable. |
| Data Transparency | The ability to clearly understand the lineage of all data; where, when, by whom and how data is created, modified, accessed, and deleted; and how it is summarized and aggregated. | With the centrality of data to the NIH's mission it is essential that the consumers of data have a clear understanding of how data is created, modified and used. |
| Manageability | The ability to operate the system with minimal effort and cost | As research funding and management systems increase in complexity and incorporate components of other systems (whether by extension or replacement of components), the need to manage those systems in order to maintain high reliability and performance becomes critical. |
| Reliability | The ability of the system to continue to operate | Increasing migration from manual or paper based processes to electronic systems requires increasing levels of reliability in order to avoid risks to business continuity |
| Security | The ability of the system to appropriately protect sensitive data including allowing only authorized users to access systems and data and only when that access is required for business purposes | As the need to provide access to personnel outside of NIH increases, the need to protect sensitive data increases |
| Process Flexibility | The ability to modify the system to support *ad hoc* and planned changes to business processes | Business processes need to evolve over time in order to allow for process optimization and to meet changing business drivers. In addition, there is the need to support *ad hoc* changes under special circumstances – a research program that needs to have all applications follow a special process e.g. Pioneer awards, or a single application that requires special handling because of special circumstances |

| Goal | Description | Rationale |
|---|---|---|
| Composability | The ability for new applications to be assembled from existing components, often by users or IT support rather than requiring fresh development | There are many cases where the functionality required by one application is also required by another, or may be required at several points in a business process. Reusing existing components to fulfill those needs reduces cost and risk of developing systems, and of making changes to the business process. |
| Semantic | The ability to categorize data in a consistent and rigorous manner using standard ontologies and taxonomies | Increasingly data created in one context is used in another context. By providing semantic context the data can be used with consistent meaning across those contexts. |

**Table 2. Alignment of Architectural and Business Goals**

| | | Business Goals | | | | |
|---|---|---|---|---|---|---|
| | | Flexible Funding Methods | Integrated Portfolio Management | Adaptability to New Technologies | Reduced Administrative Costs | Agile Business Processes and Data |
| FArchitectural Goals | Interoperability | | ● | ● | ● | ● |
| | Evolvability | | | ● | | ● |
| | Scalability | | | | ● | |
| | Reusability | | | | ● | ● |
| | Functional Extensibility | ● | | ● | ● | ● |
| | Data Extensibility | ● | ● | ● | ● | ● |
| | Data Quality | | ● | | ● | ● |
| | Data Transparency | | ● | | ● | |
| | Manageability | | | | ● | |
| | Reliability | | | | ● | |
| | Security | | ● | ● | | |
| | Process Flexibility | ● | | ● | ● | ● |
| | Composability | ● | ● | ● | | |
| | Semantic | | ● | ● | | ● |

## *3.3 Constraints on Systems Using the Proposed Architecture*

In the definition of any application architecture, there is of necessity the definition of some level of constraints on the solution architectures that fit into the application architecture. Such constraints, if not applied appropriately can easily stifle the innovation that is one of the key values of the NIH, particularly in IC developed systems.

Therefore, the application architecture discussed in this whitepaper seeks to define constraints at a level of abstraction that supports the architectural goals without overly constraining the technologies chosen by system designers. There is also the recognition that in some cases, the need for interoperability, flexibility etc. described above may be much lower than in other cases and that development should not therefore always follow the application architecture described. However, consideration of these factors should be a part of the process of planning a development effort and as such will likely be evaluated by ITAO in their role as a Critical Partner in the EPLC process.

### 3.3.1 Constraints Required for Interoperability

In order for software components to interoperate there is a need for conformity to a set of agreed standards that support:

- Data interchange with shared semantics for data

- Invocation of functionality provided by software components
- Return of results from service invocation
- Handling of exceptions in a consistent and well understood manner including behavior and communication of error statuses
- Consistent understanding and representation of business rules

#### 3.3.1.1 Data Interchange

The foundation for interoperable data interchange is found in the development of a shared understanding of the data objects that participate in business processes, the roles they play, and the relationships between the objects. Without a clear understanding of the objects, roles and relationships, it is all but impossible to define data interchange standards that are not subject to continual change. Therefore in order to achieve interoperability, data standards must be defined that change only when the business they support changes. In the NIH Enterprise Architecture the Conceptual Data Model (CDM) for each segment of business provides this foundation, and is elaborated by specific logical data models developed during system development lifecycles.

#### 3.3.1.2 Invocation Standards

In addition to the data definition, roles, relationships and standards are required to allow systems to invoke functionality provided by software components and obtain the results. Should the components be developed in differing technologies, then this invocation may be problematic – it is very difficult for instance to execute functionality that is implemented using Java technologies directly from a system that is constructed using Microsoft technologies. In order for interoperability to occur between the system and the components from which it is composed, there must be some level of constraint on the technologies used, or bridges must be constructed that allow for cross-technology execution. Clearly not every interface in a system requires such interoperability – which is positive since such interoperability comes at a cost of complexity and performance. Therefore the use of interoperable interface standards should be considered during the development of the solution architecture and applied where they are most needed.

Using traditional development techniques and technologies these connections or interfaces between the application and the components from which it is assembled tend to be “brittle” unless care is taken to support evolution. Such brittleness means that a change to either the consuming application or the components will very likely cause the interface to be broken resulting in the need to make changes at both ends of the interface. This technology then becomes a constraint on the ability of providers of components and consumers to innovate, as the cost of every change now has to include all the change required at the other end of the connection. While the issues tends to be manageable when systems are constructed in silos as they are today, as the NIH is evolving towards development of systems that are developed as orchestrated compositions of software services, failure to address the issue of potentially brittle interfaces early and in a comprehensive, enterprise wide manner is likely to reduce the ability for the new development paradigm to produce the desired results.

One solution to the issue described above is to tightly control and manage the interfaces, however this approach increases cost and risk, runs counter to the NIH’s need to innovate and is already one of the problems that is vexing the developers of systems across the NIH today.

#### 3.3.1.3 Return of Results

Just as the invocation mechanisms for required standardization in order to support interoperability, standards are required for the return of results including both data standards for returned results, and standards for the mechanism including the use of status result codes, the use of output parameters in invocations or the definition of expected return messages where asynchronous invocation is used.

#### 3.3.1.4 Exception Handling

Interoperability extends to the needs of exception handling within a composite application built from a set of loosely coupled software components. The semantics of the exception must be well understood so that appropriate action may be taken upon failure, and so that exception cause and transaction status information may be correctly interpreted so that the invoking system or application can provide the appropriate cleanup and messaging to end users. This is particularly critical where multiple components are invoked as a single atomic transaction from the user perspective; and therefore where it is critical to data integrity that the atomicity of the transaction must be retained. Within database systems this is generally handled using a two phase commit process with an overseeing transaction monitor handling rollback of all parts of an atomic transaction. In the loosely coupled world of interoperable services proposed, such transaction monitors do not generally exist, placing the burden on the invoking application to handle exceptions cleanly.

#### 3.3.1.5 Business Rules

In any complex system, business rules are defined and applied through the course of execution of an application. In a world of true interoperability, both the provider of a service and the system that is invoking the functionality must share a common understanding of these business rules. The need for shared understanding leads to the need for the use of common languages and standards for the definition of business rules.

### 3.3.2 The Solution – Abstraction and Standards

The solution to these problems is to define the requirements for interoperability at a higher level of technology abstraction than traditional Interface Control Documents (ICDs) and technology interfaces. The architecture proposed here attempts to define standards for interoperability at level of abstraction where the standards allow the developer of a service to choose the technology stack that is most appropriate to their specific needs, but conforms at a higher level of abstraction in order to support interoperability and reuse. As well as the general philosophies described above, ITAO feels that this must be supplemented by the use of a series of technologies that directly support interoperability.

Within the area of data interoperability, ITAO proposes standardization on the use of XML, XML Schema and XML namespaces founded on the NIH Conceptual Data Model (CDM), extended by ontologies defined using Ontology Web Language (OWL) to ensure that all semantics are well defined.

Abstraction and the use of common standards and technologies provide the solution to question of interoperable service invocation.[3] These would include the lower level standards for service invocation and security such as the WS series of standards promulgated by the World Wide Web Consortium, coupled with the use of Model Driven Architecture as the mechanism to both communicate standard semantics, and as a way to ensure that services can be created with the structures required for interoperability.

The need for exception handling can be met through the use of common standards for communicating exception statuses and messages, including end user messages using the exceptions standards of the SOAP protocol for RPC style services coupled with a glossary that defines the semantics of the exceptions. Here avoidance of implementation specific exceptions is critical; the invoking application should receive exceptions that look the same regardless of the technology used for implementation of the service. ITAO also proposed the extension of this model of exception reporting to asynchronous service invocation using messaging with the use of the SOAP standard for exceptions with the exception data embedded in specially defined messages to be returned to the original sender of the invocation message.

---

[3] http://www.w3.org/2004/12/rules-ws/paper/113/

The need for atomic transactions that require participation by multiple services provides a special challenge. While a service based transaction monitor based on the standards for proposed by OASIS[4] would handle this problem best, the lack of implementation of this standard by vendors suggests, at least in the interim, service providers should provide for both invocation and rollback of transactions that modify the state of data and that applications be responsible for invoking rollback services when portions of atomic transactions fail.

Lastly, in order to address the need for common semantics for business rules, it is proposed that the NIH adopt a standard notation for business rules coupled with a well-defined ontology of terms used within the business rules. Unfortunately no such standard exists today from standards organizations such as w3.org and OASIS, though both have been involved in discussions across the industry[5] with the view to defining standards. This situation has led to business process management and business rules product vendors adopting proprietary "standards". This is an area where care must be taken in order to avoid locking the NIH, or individual systems into a model for rule representation that becomes a dead end.

#### 3.3.2.1 Scope of Interoperability

All of the proposals made above for interoperability come at a cost; whether that cost is in real terms such as increased development and testing resource requirements, or at a cost of performance from the additional overhead required to support interoperability. Therefore, ITAO does not propose the wholesale adoption of these interoperability mechanisms; rather it is proposed that they be adopted only at certain levels within the overall architecture. In many cases, these levels will be at the scope of a business process, particularly when that business process is contained within a single system, particularly within an enterprise system such as NBS or eRA. However, there will be cases where the interoperability standards will need to be applied within a business process. Such an example exists within the Peer Review process where individual ICs may wish to have systems that support identification and selection of peer reviewers outside the processes and tools provided by eRA. Here the interoperability within the process is designed to allow an IC to step out of the enterprise process and either supplement or replace a portion of the overall process with their own system or services. At this level, the alignment model may become less of an alignment to process and rather to a locus of control – see section 3.5.3 below for details.

## *3.4 Architectural Style*

Within the context of the discussion above on levels of abstraction, in order to meet the business needs and architectural goals above, this paper presents a view based on an architectural style called Service Oriented Architecture (SOA) with Business Process Management (BPM), coupled with the Event Driven Architecture (EDA) style, sometimes termed "SOA 2.0". Throughout this paper, this architectural style will be referred to as "SOA+BPM+EDA" with specific components of the architecture referenced separated e.g. SOA.

### 3.4.1 Service Oriented Architecture

Service Oriented Architecture (SOA) is a style of software architecture aimed at ensuring that a system is flexible and tolerant of changes in business requirements. The primary architectural element of SOA is the Service, a software component that exhibits a set of characteristics allowing the architecture to meet these goals. The sections below describe each of these characteristics and how each contributes to the desired goals. The Service should not be considered to be synonymous with Web Services and the set of technologies that surround them. It is quite possible to implement services using .NET, Java RMI or JMS as the interface technology as well as Web Services; and in many cases avoidance of Web Services makes

[4] http://www.oasis-open.org/committees/ws-tx/

[5] http://www.w3.org/2004/12/rules-ws/

good design sense. While Web Services provide very high levels on interoperability and extensibility, they do so with a significant penalty in performance. Here the structural aspects of the architecture in terms of how behavior is partitioned become critically important.

A group of solution architects from throughout the NIH together with ITAO staff have developed a series of principles for service design. These principles and the impacts on each of the architectural goals noted above are documented in the "Design Principles for SOA" document available from ITAO.

### 3.4.2 Business Process Management

As has been suggested above, the incorporation of Business Process Management (BPM) into a Service Oriented Architecture allows for the orchestration of complex business activities using a configurable workflow manager. The process management system ensures that services are invoked at the appropriate point in the business process, that the appropriate data is conveyed to the service, and that where user involvement is required, the process management system informs the appropriate individual to perform some action. The process management system is generally also capable of tailoring the specific business process to be applied for some data entity flowing through the system. For example, a grant application received by the Center for Scientific Review (CSR) may require special handling because it is for a multi-disciplinary research project and requires review by multiple review groups. Within constraints of legislation, regulation and policy, a user could then tailor the process to send the application to all the required review groups, whether in sequence or in parallel.

Such process management systems can also apply what has been called the "Executive Correspondence Model", where a specific workflow can be tailored for the specific needs of a given data entity or document. This model supports the ability to deal with exceptional circumstances as a normal part of business; for example, sending a piece of highly sensitive executive correspondence to a specific set of reviewers and legal counsel before it is sent to the recipient. Similar examples for exceptions can be observed in other processes where a specific class of data entities need to follow a different process than usual: grants under the American Recovery and Reinvestment Act (ARRA), for example, had to follow a process different from the normal grant process. The executive correspondence model allows for these exceptions to be managed within the context of the existing systems rather than requiring external intervention and special handling.

The combination of the SOA and BPM provides the ability to continuously improve business processes or to adapt to changing business needs with minimized involvement required by IT services. Where involvement of IT is required, changes tend to be isolated to specific services and through the loose coupling characteristics result in lowered risk to the overall operation of the business process when changes are made. When problems do arise, the decoupling results in significant improvements in fault tolerance with services not impacted by the fault able to continue operation and if necessary the workflow management software allows a business user to make the decision to temporarily bypass the problematic service step or move it to a later point in the workflow for completion. It is anticipated that this level of control may be provided to the end user of a workflow management system, or at least to an individual within a business unit with sufficient knowledge to provide such support to end-users.

### 3.4.3 Event Driven Architecture

One means of achieving the loose coupling that makes an architecture tolerant of change, extensible and evolvable is the use of Event Driven Architecture (EDA). In an EDA, rather than using specific synchronous invocations of services or components during a business process, services are invoked using asynchronous messaging with the sending of messages tied to the occurrence of specific business events. An example of the use of EDA in the NIH environment might be the processing that needs to occur when a grant is awarded. There are clearly parties that are interested in that event. The grantee institution and the IC that is funding the grant for example both want to know when a grant is awarded so that they can perform their specific business processes for this event. However, there may be others that are less obvious and that may vary based on the some characteristic of the grant. A grant made using American Recovery and Reinvestment Act (ARRA) funds for example must eventually be posted to Recovery.gov so there is the potential interest of Recovery.gov in the award event, rather than NIH sending data to Recovey.gov whenever a grant is awarded, Recovery.gov could listen for the award events for ARRA funded grants. Conventionally these interactions have been handled via calls to specific software components or the delivery of a data file for processing asynchronously. While these models work, they are inflexible – the addition of another interested party requires specific development to support communicating with the new party. The responsibility for getting the data to the party lies with the system that supports the specific business event – in the case of grant award, eRA.

In the EDA model however, the responsibility for receiving notification of an event lies with the interested party. In the examples above, the responsibility for eRA ends with the announcement of the fact that a grant has been awarded. Interested parties register for notification when this event occurs and the EDA infrastructure – an Enterprise Service Bus (ESB)[6] – is responsible for ensuring that the message reaches its destination. The EDA model provides for great flexibility as the addition of a new interested party requires no change on the part of the event provider – the new party simply registers their interest in the event and the ESB ensures that they are notified when it occurs.

## 3.5 Separation of Concerns

One of the principles used to guide the definition of the proposed architecture is the principle of “Separation of Concerns”, that is to separate software systems into elements that overlap as little as possible in functionality, and then to couple each area of concern as loosely with one another as is feasible.

In complex application architectures the separation of concerns can be described from several viewpoints. Each viewpoint looks at a different aspect of the architecture largely independent of the others. An individual element of the architecture therefore can be categorized according to each of the viewpoints and needs separation into compartments where a single set of viewpoints are present. The following viewpoints are discussed below

- Business domain
- Data domain
- Locus of management control
- Layering

### 3.5.1 Separation of Business Domain Concerns

A Business Domain is an area of business activity that plays a significant role in the business’s value chain and hence contributes to a subset of the business’s goals. A business domain has a set of business processes that are designed to produce business value, and contribute to meeting business goals. A domain’s business goals may be distinct, they may support the business goals of another domain; i.e. one business domain may form a complete value chain together with another business domain. While there may be a strong relationship between business domains and organizational structures, the two are not synonymous; rather the organizational structure tends to arise as a result of the separation of business domains and their associated distinct processes.

The following business domains have been tentatively identified, though this list may not be complete:

[6] http://enterprisearchitecture.nih.gov/ArchLib/AT/TA/EnterpriseServiceBusPattern.htm

- Scientific portfolio management e.g.
    - Scientific portfolio analysis
    - Research agenda management
    - Funding opportunity development
- Research funding e.g.
    - Receipt and Peer Review of Grant Applications
    - Management of grants
    - Progress and financial reporting for grant recipients
    - Research and development contracts
    - Inter-agency agreements
- Research performance and management e.g.
    - Annual reporting
    - Clinical data management
    - Protocol management
    - Patient accrual
    - Animal model development and management
    - Bio-specimen management
- Fostering of new scientists e.g.
    - Training and career development grants
    - On-site training and mentoring e.g. internships
- Supporting services e.g.
    - Technology transfer
    - Acquisition management
    - Budget and financial management
    - Human resource management
    - Capital resource management (Assets, facilities etc.)
    - Security – both physical and information
- Knowledge dissemination
    - Storing and aggregation of knowledge
    - Conference management

The importance of the business domain is rooted in the need for governance and management of change in the application architecture with the business stakeholders for a specific domain having governance control over the IT systems that support their value chain. However, an analysis of the business domains above often quickly reveals commonality of processes across business domains.

The management of clinical research data and the management of patients in the NIH Clinical Center for example share some common processes – patients need to be registered when they enter the Clinical

Center, just as they need to be registered on the protocols in which they are participating. As they continue through the protocol, appointments are made and clinical data is captured – both the regular clinical observations made of a patient staying in a hospital, as well as capture of specific clinical data related to the specific protocol in which the patient is enrolled. While the data varies between patient management and clinical data management, the processes are similar and the data captured in patient management and clinical data management frequently need to be combined for analysis.

In a similar way, the management of grants, R&D contracts and acquisition management for example all have the need to perform some kind of review and scoring of technical merit. While the criteria and methods for scoring vary, the fundamentals of the process are essentially equivalent i.e. a group of people with expertise in the area of interest are selected, conflicts of interest are managed, they evaluate and score the materials, meetings are held to review the scoring and reach consensus on a result, and a formal result summary is produced.  This is true whether the group is evaluating grant applications or research contract proposals.

The commonality described above suggests that some common software services could be developed to support this common process across these business domains. Such commonality directly results in cost savings and reduced overhead since the systems that support multiple areas are implemented by reusing existing, proven software components.

However, this reuse across business domains requires that the traditional view of complete control over the IT systems that support that domain by a single business organization be modified as change to common services will impact more than one business domain. Therefore, change to these shared services needs to be managed by cross-domain governance bodies.

### 3.5.2  Separation of Data Domains

The Business Domains perspective partitions the architecture by business function and ties the partitioning closely to the NIH mission and the support services required for that mission. While these business domains are separate in terms of the business goals they service, they cross boundaries in terms of the data that is used to meet the business goals. The NIH Extramural Program for example, is supported by budget and financial management, acquisitions, scientific portfolio management and consumes and produces data for other business areas. An extramural project may use an R&D contract, have a budget, and use personnel that are directly managed by the acquisition, budget and human resource management business domains. The mandate for reporting participation of women and minorities in clinical research (Population Tracking), requires data from grants, R&D contracts, and intramural clinical research.

Therefore, there is value in separating the business domain viewpoint from a Data Domain viewpoint. While there are certainly parallels between the two viewpoints and in many cases the data domain may directly align with a business domain, there is a cross-cutting aspect to the data domains that suggests that the viewpoint is distinct. Each data domain has an associated “Business Owner” and “Custodian”. The business owner of a data domain is the organization primarily responsible for governance of the data and for definition of data semantics. For example the Office of Financial Management would be the likely business owner for financial data. The custodian of the data denotes the system and associated development program that manages and controls the data. For patient data for example, the custodian would be likely be the program that manages and maintains CRIS and BTRIS.

Just as common software services will support multiple business domains, data domains will support multiple business domains. People data will be used across the several business domains including grants, financial management, security, clinical research, and others. Therefore, a similar cross-domain model for governance of data is required.

Based on past analysis performed during the development of the NIH Enterprise Conceptual Data Model the following data domains have been identified, and more may exist.

**Table 3. Candidate Data Domains**

| Data Domain | Description |
|---|---|
| People | The People data domain records information about all People associated with the NIH, whether staff, contractors or researchers funded by an NIH grant |
| Grants | The Grants data domain records information relating to all NIH grants |
| Contracts | The Contracts data domain records information relating to all NIH contracts including R&D contracts |
| Financial | The Financial domain contains all data relating the financial transactions and planning |
| Inventions | Contains data relating to inventions and patents created as a result of NIH sponsored research, whether intramural or extramural |
| Research | Contains the data associated with a scientific program including extramural and intramural research projects |
| Clinical | Contains clinical research data including bio-specimens |
| Patient | Contains data related specifically to patients at the NIH Clinical Center |
| Facilities | Contains data associated with physical facilities |
| Organization | Data relating to organizations within NIH and external organizations that do business with NIH |

### 3.5.3 Separation of Locus of Control – Service Taxonomy

In an architecture such as that defined in this paper, there is a need to have a clear definition of the locus of management control to be applied to architectural elements. While business domains are defined based on coherent sets of business goals and associated processes, the locus of control viewpoint examines where governance and often funding exists organizationally. The locus of control described below is directly related to the "Service Taxonomy" that was developed by a senior level domain team led by Gartner Inc.

It is appropriate for example that architectural elements that are unique to a system or application that is designed meet a program's specific business needs be governed and funded by the program that develops the system. At the other extreme, architectural elements that are used across all of NIH to meet common needs should be governed and funded at an enterprise level.

The following table identifies the locus of control contexts that were identified by the domain team and describes the characteristics of each context. It should be noted that an architectural element may begin life at one level, e.g. Local, and become another level if it is found to be useful to a broader audience e.g. Enterprise level. When this change in context occurs, it would be likely that additional investment would be required in order to meet the stricter scalability, availability and reliability requirements at the higher level. This need for additional investment also suggests that there is an advantage to making a determination of the likelihood of a service becoming higher-level service early in the development lifecycle. Such an early determination would allow for the required enterprise level investment to be made during initial development in order to reduce total overall costs, as retrofitting is likely to be significantly more expensive that original construction.

**Table 4. Locus of Control Contexts**

| Taxonomy Classification | Locus of Control | Characteristics | Technology |
|---|---|---|---|
| Multi-Enterprise Services | Multi-Enterprise<br>• Development may be by a specific system or an enterprise development group.<br>• Assumes that funding will come from an enterprise resource pool and may be supplemented by the external consumers of a service. | • Consumers: External to NIH, other agencies, grantees, commercial organizations<br>• Service provider does not typically know all of the service consumers though in cases were consumers are limited to specific groups or organizations this might not be the case<br>• Covered by an enterprise Service Level Agreement<br>• Registered in the NIH and potentially external service directory<br>• Typically supports NIH business processes that require interaction with external organizations, or Federal Government level business processes in which the NIH participates<br>• Interfaces and behavior governed by an external change control board with NIH participation<br>• Data exchanged uses agreed Conceptual Data Models including the NIH enterprise CDM as well as other external standards e.g. HL7<br>• Run time binding to interfaces | Web Services<br><br>Constructed using technologies for high scalability, availability and reliability<br><br>For secure services requires<br>• Federated Authentication<br>• Potentially PKI<br><br>Examples:<br>• caBIG Services<br>• PubMed Search Service<br>• eRA Services for Grant Submission Service Providers<br>• Grants.gov<br>• iTrust (NIH Login) |

| Taxonomy Classification | Locus of Control | Characteristics | Technology |
|---|---|---|---|
| Enterprise Services | Enterprise<br>• Development may be by a specific system or an enterprise development group such as eRA or NBS<br>• Assumes that funding comes from an enterprise resource pool specifically for enterprise services. Where a service is developed and maintained by a specific system, enterprise funding is provided to both develop the service to enterprise level and to maintain the service levels required by the enterprise. | • Consumers: Significant portions of the NIH<br>• Service provider typically does not know all of the consumers of a service<br>• Covered by an enterprise Service Level Agreement<br>• Registered in NIH enterprise level service directory<br>• Typically support an enterprise business model<br>• Interfaces and behavior governed by an enterprise level change control board<br>• Data exchanged confined to the Enterprise Conceptual Data Model<br>• Run time binding to interfaces<br>• May have external requirements | Web Services<br>Constructed using technologies for high scalability, availability and reliability<br>Secured by:<br>• NIH Authentication (NIH Login)<br>• NIH Authorization<br>Examples:<br>• Enterprise Person Search (NED)<br>• Purchase Requisition Service used by POTS and AMBIS |

| Taxonomy Classification | Locus of Control | Characteristics | Technology |
|---|---|---|---|
| Shared Services | Shared<br>• Development by a specific system supporting the business domain (Note that Shared Services developed by an enterprise system such as eRA, NBS or CRIS are by definition Enterprise Services).<br>• Where a service is developed by an IC, funding from other sources may be provided from an assumed enterprise resource pool to the provider in order to support the broader domain needs, whether supporting additional functional requirements or enhanced service level agreements | • Consumers: Specific domain area e.g. Extramural Funding, Financial Management, Acquisitions etc. that crosses IC boundaries<br>• Service provider typically knows all of the consumers of the service<br>• Covered by a domain service level agreement<br>• Typically supports point to point interfaces<br>• Interfaces and behavior governed by a domain specific change control board<br>• Data exchanged conform to the domain specific data model e.g. the Grants Segment Conceptual Data Model<br>• Run time binding to interfaces | Services, often Web Services, but in some cases may use messaging and the Enterprise Service bus or specific agreed technologies<br><br>Constructed using technologies for high scalability, availability and reliability<br><br>Security<br>• NIH authentication |

| Taxonomy Classification | | Locus of Control | Characteristics | Technology |
|---|---|---|---|---|
| Local Services | Business Domain Specific Services e.g. Grants, Acquisitions, Finance | Application<br>• Developed by a specific system to meet the system's requirements<br>• Funded entirely by the system | • Consumers: Internal use by a specific system<br>• No service level agreements<br>• Interfaces and behavior governed by system's CCB<br>• Data exchanged conform to the domain specific data model e.g. Grants Segment Conceptual Data Model<br>• Run time or compile time binding to interfaces<br>• May be an enterprise system such as eRA or NBS for internal use, or an IC developed application | System determined<br><br>Constructed using technologies appropriate to the level of scalability, availability and reliability required by the application<br>May conform to EA Bricks and Patterns<br><br>Examples:<br>eRA Grant Application Checklist |

| Taxonomy Classification | | Locus of Control | Characteristics | Technology |
|---|---|---|---|---|
| | Application Components | Application<br>• Developed by a specific system to meet the system's requirements<br>• Funded entirely the system | • Consumers: Internal use by a specific system<br>• Does not necessarily strictly conform to the SOA paradigm e.g. internal frameworks that services depend on for operation<br>• Governed by the system change control board<br>• Data exchanged may be in whatever data model is appropriate to the needs of the system including the physical data model<br>• Compile time binding to interfaces<br>• May be an enterprise system such as eRA or NBS for internal use, or an IC developed application | System determined<br>Constructed using technologies appropriate to the level of scalability, availability and reliability required by the application<br>Conforms to EA Bricks and Patterns<br>Examples:<br>eRA Application Framework<br>NED Application Framework |

### 3.5.4 Layering

The third viewpoint of the architecture to be presented is that of architectural layering. Layering is a type of structural pattern that classifies services into strata where each layer is assigned specific characteristics and constraints on the types of operations the layer supports, and how the elements contained in the layer are allowed to communicate with other layers. The goal of layering is to ensure that the granularity of architectural elements is defined at appropriate levels to allow for interoperability and for applications to be constructed by composition of elements, a task that is much easier to deal with when there is consistent granularity. The sections below discuss two distinct classes of services that have been identified in the proposed application architecture and how their granularities differ, and how functionality should be partitioned between the two classes based on business process analysis.

It should be noted at this point that the layering model of the architecture is entirely orthogonal to the locus of control i.e. regardless of the partitioning of functionality that places a service within a specific layer of the architecture, that service may be categorized with any of the loci of control described above; a service may be an "Atomic Business Service" and have an Enterprise scope just as much as it could be a "Shared" or "Local" service. Further, it is possible a service within a particular layer will evolve from a Local Service to a Shared Service and potentially to an Enterprise Service – without changing the layer in which it exists. The rationale for this conclusion is that the locus of control is primarily about management and governance of the service, whereas the layering is determined by service granularity and the functionality provided.

Figure 1 below illustrates the proposed overall layering model, and the Table 5 below summarizes the role and characteristics of each layer.

**Table 5. Layering in the architecture**

| Layer | Role | Characteristics |
|---|---|---|
| Web Browser | Displays the Management Console that is the user's interface to the system | • Browser must support asynchronous update mechanisms similar to Ajax and HTML5 |
| Management Console | An Enterprise Mashup of user interface components created by or for the user | • Uses user interface e.g.Javascript events mediated through a broker to communicate with back end services and update mashup components. |
| Business Process Management | A workflow engine that orchestrates service invocations | • Uses a standard workflow language for orchestration e.g. BPEL |
| Orchestrated Business Services | Coarse grained services that support small sets of business activities | • Typically has a user interface that appears in the user's mashup (management console)<br>• May be composed from lower level services<br>• Typically does not access data directly from a database<br>• Typically supports a single use case |

| Layer | Role | Characteristics |
| --- | --- | --- |
| Atomic Business Services | Finer grained services that produce a single result of value to a user | • Service interface appropriate to the locus of control context<br>• No user interface<br>• May be composite<br>• Typically does not access data directly from a database |
| Data Services | Specialized Atomic Services designed for transactional data access and update.<br>Two classes: Data Providers, Data Updaters | • Service interface appropriate to the locus of control context<br>• No user interface<br>• Updaters must support transactional safety<br>• Designed to return small numbers of data records |
| Data Loading | Loads data from online system datastores to Data Warehouse | • May use traditional Extract Transform Load (ETL) techniques<br>• May use real-time pushed data<br>• May use real-time events to initiate pull of data<br>• May use real-time events to push data |
| Federated Data Warehouse | A fully temporal database storing historical data from online systems optimized for reporting. Includes links to non-structured data (documents, images, video etc.) stored in a document management system and indexed using semantic analysis tools | • Full history of data changes<br>• Built from multiple domain specific physical data warehouses<br>• Includes semantic indices for non-structured data<br>• Supports semi-structured data indexing e.g. XML or HTML |
| Reporting Data Services | Provides access to reporting data from the data warehouse for regular "canned" reports | • Delivers aggregated and summarized data<br>• Delivers data using pre-defined criteria<br>• Provides support to specific business processes |
| Data Marts | Provides access data stored in the data warehouse optimized for business intelligence, data analytics and portfolio analysis | • Multiple Data Marts with data structures optimized for specific business needs<br>• May draw in data from sources other than the NIH data warehouse |

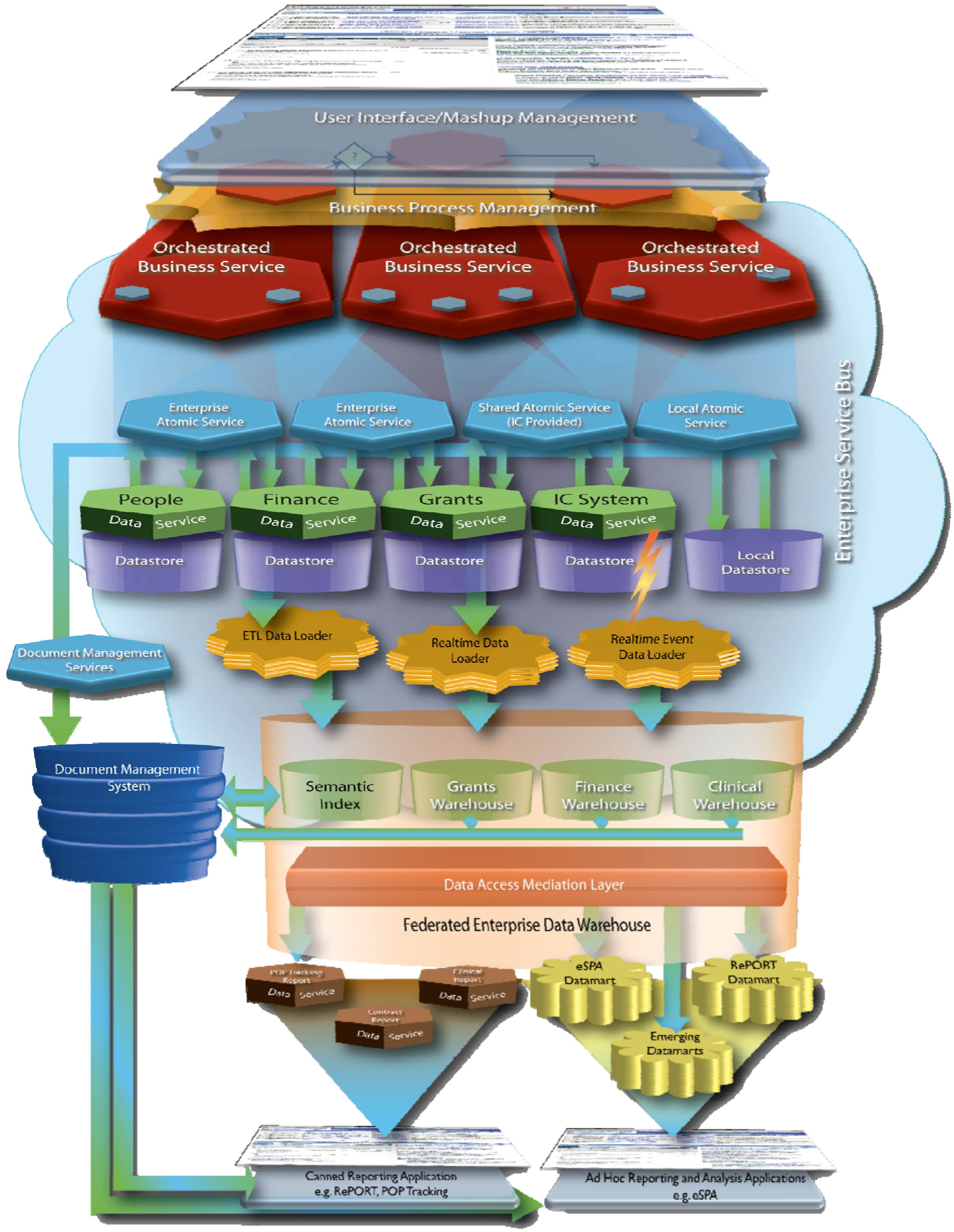


**Figure 1. Layering in the proposed architecture**

### 3.5.4.1 Atomic Services Layer

Some services provide a single result of value such as the transformation of a data entity from one state to another as in the final process for issuing a grant award once all information required has been gathered and all approvals obtained. In the architecture such services are termed "Atomic" in that they provide a single result of business value without need for user intervention.

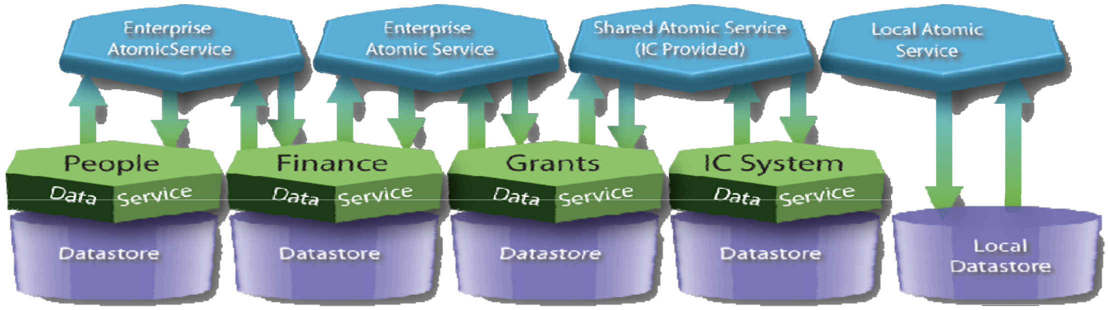


**Figure 2. Atomic Business and Data Services**

Within the class of Atomic Services are two distinct sets of services: Atomic Business Services that support specific business needs, and Data Services designed to support access to and update of data – illustrated along with their relationships to Atomic Services in Figure 2 above. Each data domain would have an associated set of services for access and update (see Table 3. Candidate Data Domains above). In order to ensure that data is managed appropriately, direct access to and update of data entities in a physical data store is limited in the architecture to the custodian of the data i.e. only the custodian system may perform database operations on the data. All others systems will access the data mediated through a Data Provider Service and a Data Update Service – both developed and provided by the custodian of the data. It should be noted that it is unlikely that these data services will be designed to execute arbitrary queries, they are more likely to provide a more limited set of focused query interfaces that provide specific subsets of data elements based on specific criteria. Given the focus on these services in supporting operational rather than reporting systems, this approach is likely less restrictive than it sounds and provides the data custodian the ability to carefully tune the execution of queries in the physical data store.

By abstracting data access and update in this manner, the custodian of the data is able to manage and control access to the data allowing for the custodian to optimize their physical data store as needed for performance and other factors while retaining the consistent semantics provided by the use of canonical data formats. These data services also provide a single security chokepoint for data access that can be managed and controlled by the custodian. This chokepoint allows the custodian to selectively provide or hide (blind) data elements that the consumer of the service may not see. Such blinding requires the service to use a standard pattern for authentication and authorization that will be defined in Part II of this whitepaper.

### 3.5.4.2 Orchestrated Business Services

In simplistic models of architectures using SOA+BPM, the full functionality of an application or system is provided by assembling a set of atomic services orchestrated by a BPM tool – often with the BPM tool also providing the user interface for the application. This model may be achievable in an environment where development is started from the ground up and there are few existing systems, but in the NIH environment where integration between existing systems is a specific requirement such a model is difficult and undesirable, as it tends to reduce reuse of existing, well-tested systems.

Therefore, the proposed architecture defines a layer containing a set of non-atomic services termed "Orchestrated Business Services" that support much coarser grained chunks of business process functionality. Each is designed to be reusable, with each proving its functionality via orchestration of a set of atomic services and interaction with users. We term this process of lower level workflow within an Orchestrated Business Service "Micro-Orchestration" illustrated in Figure 3 below. This orchestration may be achieved through the use of a workflow or BPM tool, maybe the same tool used at the higher level of orchestration but maybe not. The orchestration may also be achieved through other means such as conventional state management using status codes in data entities managed by the Orchestrated Business Services.

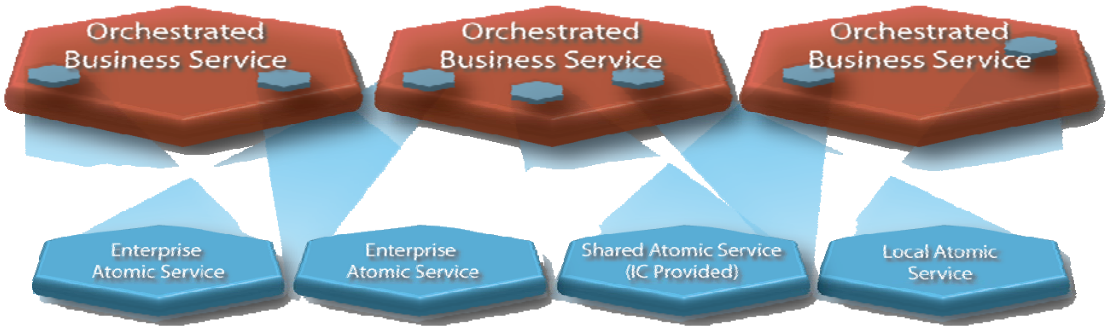


**Figure 3. Orchestrated Business Services as Compositions of Atomic Services**

Given the long running nature of services that require human interaction, it should be expected that access to these services would not use a synchronous RPC style call whether it be web services or some other remote invocation model. Instead asynchronous models will be the most appropriate. Invocation of these services may be via a synchronous RPC style call but with the response returned indicating that the orchestrated workflow within the orchestrated business service has been completed. Alternatively, asynchronous messaging may be use to invoke the service. Such a model also presents significant challenges for integrating user interface components provided by the orchestrated services into a coherent user interface. This issue will be addressed in a forthcoming whitepaper that addresses specific proposals for key mechanisms of the architecture.

In both cases, the system invoking the orchestrated service needs to know when the service has completed its work and often the result of that process so that it may continue with the macro workflow that required the orchestrated service. There are two candidate mechanisms for this process. The first uses a callback to a service endpoint designated in the original orchestrated service invocation. The second uses messaging in which the orchestrated service published an event indicating that the service has completed. The invoking system subscribes to these notifications and moves the workflow to the next process step when the message is received.

A concrete example of such decomposition into atomic and non-atomic orchestrated services might be the requirement for managing Conflicts of Interest (COI) in peer reviewers for grants and contract, council members and NIH staff ethics reporting. In this case the need for managing COI is very similar in each context: It requires checking a person against their collaboration history, employment and financial involvements, and then having a human review those interactions to determine if they are significant enough to bar the person from involvement with a specific person, grant etc. Here the non-atomic service might be termed "Manage Conflicts of Interest". It would support micro-orchestration of several atomic services such as "Match People with Publications" to identify co-publishers, "Match People with Institutions" to identify places where people worked at the same institution and "Identify Potential COI for Financial Involvement" where a financial relationships are identified with organizations. The non-atomic service might invoke each of the atomic services to produce a list of potential COI, then present this list to a user for them to evaluate and make decisions on with the decisions recorded using another atomic service called "Waive COI" or "Record COI Assessment".

#### 3.5.4.3 The tie to business processes

In order to achieve the model described above, it is essential that the granularity of services should closely map to the business process with atomic services supporting single activities in a business process and non-atomic services supporting a small number of closely related steps that are functionally decomposed from the activity supported by a non-atomic process. Such a statement presupposes that this decomposition has been performed appropriately. The NIH standard Business Process Model is designed to ensure that such decomposition is performed to produce activities with the appropriate level of granularity[7].

The distinctions between atomic and non-atomic service described by this methodology, when coupled with the loose coupling and encapsulation characteristics of SOA, leads to a system that is has significantly improved maintainability as compared to traditional architecture. This improvement in maintainability arises because the partitioning of functionality by business process activity means that system changes driven by a change to business practices or processes tend to be isolated to those services supporting the specific business process steps that are changing with all other services remaining untouched. The result is a reduction in the need for code changes and associated regression testing for the unmodified components, which in turn reduces the overall cost of implementing new or changed functionality.

As was observed earlier, the level of abstraction at which such central structural guidelines as layering are created is essential to an architecture meeting its requirements. Therefore, the proposed architecture defines layering at the level of services. By defining layering at the level of services we benefit from the coarse-grained nature of the services and allow for individual service providers to make decisions about internal layering based on their chosen technology stack while ensuring that at the coarse grained level services are interoperable where they need to be.

#### 3.5.4.4 Business Process Management

Well-defined services and the applications built from them provide significant benefits in terms of maintainability, interoperability, data and functional extensibility and evolvability. However they do not address the question of process flexibility, that is the ability to change business processes without significant re-engineering of the systems that support those processes. The most common need for process flexibility comes from one of two motivating factors:

- The desire to improve or optimize the business process
- The need to support a special, custom process in a limited set of cases such as handling of exceptional cases or the need to tailor the process for a new need

Commonly, systems track the flow of a process using data elements that represent the state of the process. In managing the grant process from receipt through award for example, IMPAC II updates a set of data elements for the Grant Application entity that reflect where the application is in the process and in turn uses these data elements to drive the availability of functionality in the user interface. This method, while effective lacks flexibility. When a change is required in the business process, not only do the applications have to change in order to support new values for the data elements managing state, but the user interface code has to change as well, and data must be migrated from one state to another in order to achieve a smooth transition. The difficulty of this kind of change essentially precludes the ability of the system to respond to the desire to optimize the process or to experiment with optimizations that may be of benefit.

---

[7] Additional information or training in business process modeling methodologies is available from ITAO – enterprisearchitecture@nih.gov.

The situation for handling exceptional cases or process for special classes of data is even worse. Should an exceptional case arise where the process needs to modified for this single case, the process has to be handled manually and often outside the system – without making ad hoc changes to the data elements representing state these cases cannot be handling within the system. The same problem prevents developing a special process for a special class of data to be moved through the system. The issue is well illustrated by the process modifications required to support grants made with ARRA funds. Here a special set of grant applications needed to flow through the system in a different manner than that used for regular applications. There was substantial effort required to support these applications that made the achievement of the desired results significantly more complex and expensive than they could have been. Such inflexibility similarly impacts the ability of ICs to develop special funding programs that require a different type of process and has caused a proliferation of grant mechanism that have had to been implemented in IMPAC II, or the use of grant mechanisms that were not precisely what was desired but was close enough to be able to avoid extensive engineering work.

The solution to these issues is well established in the software development world. The use of data driven workflow tools supports precisely the kind of flexibility that is described above. As it is proposed in this whitepaper, workflows would be created to support specific business processes with specific processes tailored to the characteristics of the data entities that are manipulated through the process. Such a model is currently in use within the NIH Enterprise Directory (NED) application to support PIV card issuance under HSPD-12[8].

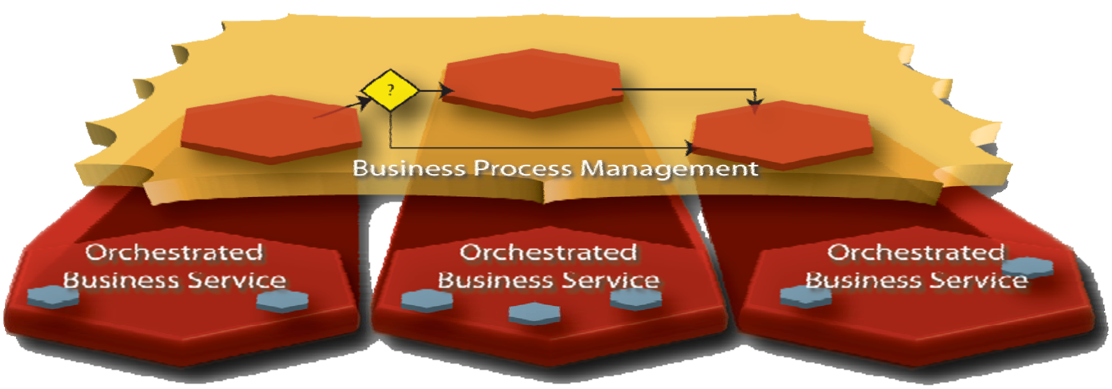


**Figure 4. The BPM Layer of the Architecture**

What is described above is best termed “Workflow” – the management of having people or systems perform their roles in a process. Business Process Management (BPM) however is more than just workflow. BPM adds to workflow the measurement and monitoring of performance metrics to the workflow that allows for analysis of processes and optimization based on those metrics and hence supports management of the business. These metrics include quantitative factors such as the time taken to complete certain process activities or the number of times a process has to be repeated due to errors. It also allows for recording of qualitative factors such as assessment of how well a particular set of data met certain criteria at points in the process. This functionality is crucial to the ability of the NIH to optimize processes, improve outcomes and improve efficiency and effectiveness.

### 3.5.4.5 User Interface Management

In any system that interacts with users, the interface between user and system must be managed by some component of the system regardless of how information is presented – whether via web browser, an

[8] HSPD-12 – Homeland Security Presidential Directive 12

Applet or ActiveX control or a rich client model typically following a Model-View-Controller (MVC) design pattern. As has been described above, Orchestrated Business Services in the architecture will present their own user interface. There will likely also be components of the user interface that are produced and managed by a BPM tool. While it is expected that most orchestrated business services will use an MVC model internally, this model does not readily extend to a user interface synthesized from multiple orchestrated service user interfaces and a BPM tool's interface.

The model proposed for user interface management in this whitepaper is that of the "Enterprise Mashup". An Enterprise Mashup is a web application hybrid in that it integrates several web-based applications into a single coherent and interacting user interface. Whereas MVC applications manage and present data, mashups manage content delivered in XML, HTML, RSS and Atom. Where MVC applications manage interactivity directly, mashups manage interactivity using well defined APIs that support updating component UIs based on actions taken by a user such as clicking on a link in another component.

All of this must be managed and controlled in order for the orchestrated services to be synchronized in the data that are displaying or managing. We propose that the task of managing the mashup be split between two layers: A layer residing in the user's web browser and another residing on the server.

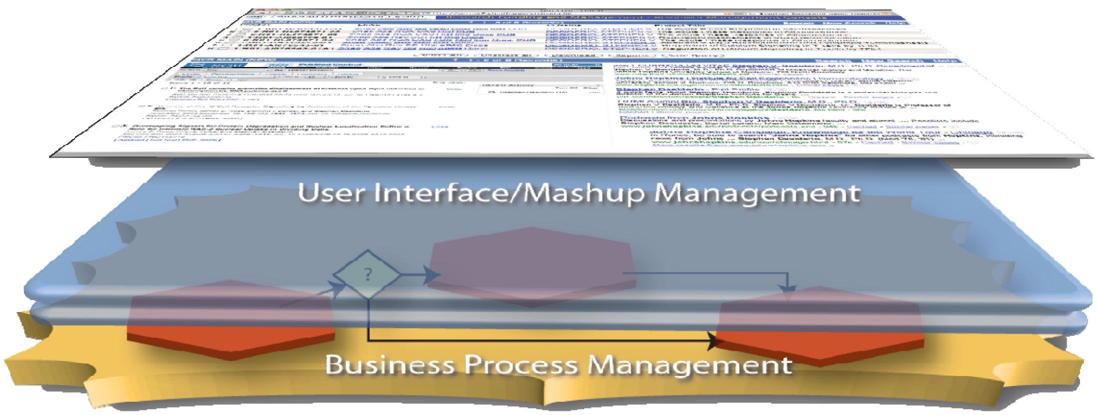


**Figure 5. User Interface Management Layers - Management Console and Web Browser**

The web browser layer implemented using scripting languages such as Javascript, partial page updates managed through the browser's Document Object Model (DOM) and the use of well known APIs for retrieving data from a web server such as those provided by the XMLHttpRequest object. Readers will likely recognize this model as the Ajax[9] (Asynchronous JavaScript and XML) model currently in vogue for constructing web based application. This paper does not directly propose Ajax as a standard since the history of such models is brief and is rapidly evolving. It does however propose this type of model as one that can be effective in providing the required functionality to support mashups.

The second component we have termed, for want of a better name, a "Management Console". The Management Console role provides the interactivity between orchestrated services displayed in the web browser. The need for this layer can be illustrated with an example. A conventional web application built using Ajax takes a user event such as clicking on a link on a page, which is then handled by the browser executing a call to the server, and then updates the page object model based on the response to display the appropriate changes to in the UI to the user. This model requires that the script handling the user click has knowledge of the document object model of all the user interface components to be updated, and the

[9] http://en.wikipedia.org/wiki/Ajax_(programming))

server request that must be issued to retrieve the required data. This model simply does not work in the world of mashups as by definition, orchestrated business services are independent from one another and the internal scripts and document object model structure of their user interface components should be hidden from outside view. The mechanism proposed for handling this problem is discussed in a forthcoming whitepaper that discusses specific proposals for such key mechanisms.

### 3.5.4.6 Data Warehouse

As was noted above, reporting is a critical issue for the NIH. Therefore the architecture attempts to address the question of integrated, enterprise wide reporting. We propose the progressive implementation of a federated enterprise data warehouse – illustrated in Figure 6 below. The federation of the data warehouse is driven by the fact that there have been significant investments in a data warehouse that supports several business domains including financial, human resources, inventions and patents, and some grants data (nVision) and a data warehouse for clinical data (BTRIS) along with other data marts in the ICs. These investments demonstrated some success for their stakeholders; though not complete success. Further data domains will need to be integrated into the data warehouses. Based on a literature review, it is the view of the author that while some of this may effectively be done in the existing data warehouses, a more rational approach may be to move towards a more clearly federated data warehouse model with multiple warehouses focused on specific data domains.

While nVision, BTRIS and other data marts exist today, much of the data that is used for reporting currently comes from one of the production transactional systems such as eRA/IMPAC II, either by direct query or via locally held copies of these databases. The continued direct use of these transactional databases for reporting poses significant threats to the performance and scalability of both the transactional and reporting systems. Performance impacts on eRA/IMPAC II have been observed in the recent past that can be directly attributed to this model of use. The use of local copies of data results in strong dependencies between the reporting and transactional systems that increases the cost and risk of development and maintenance on both sides. Further, the use of transactional system for reporting essentially precludes the ability to perform queries that request data "as of" a certain date unless the transactional databases are designed using temporal database techniques and technologies – which tends to have a negative impact on the performance on non-temporal queries and hence reduce overall system performance.

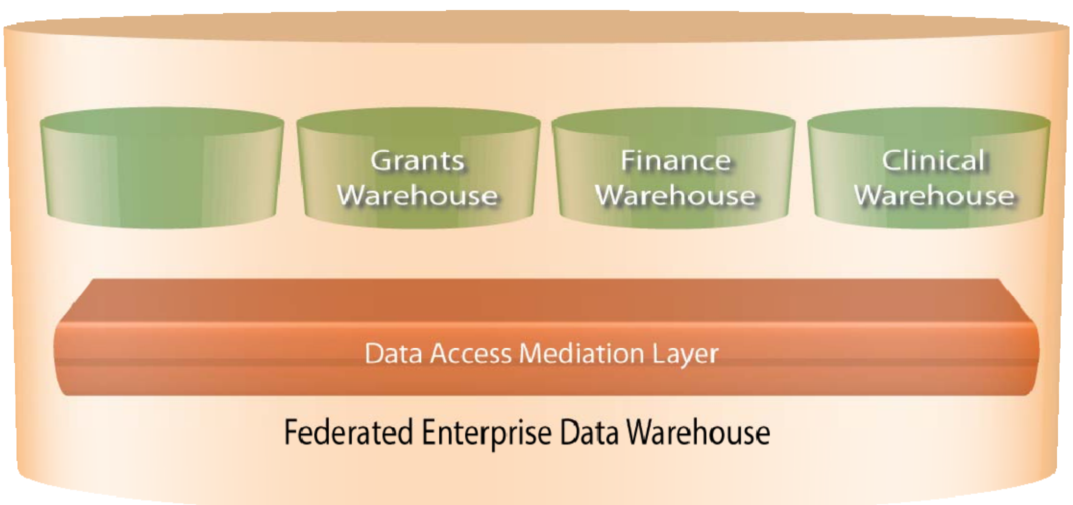


**Figure 6. The Federated Data Warehouse**

The model we propose is then, for a federated model of enterprise data warehouse where the existing data warehouses remain largely independent and new data warehouses be constructed for other data domains rather than attempting to integrate all into a single data warehouse. This recommendation is made in part based on the Kimball[10] model of allowing the development of business focused data marts that then evolve into a true data warehouse for their specific domain. This federated model requires a mediation layer between the physical data warehouses and the consumers of this data. The mediation later integrates data coming from the physical data warehouses based on well known data elements that provide the links between data entities in each physical data warehouse. In addition, there is likely a role for the mediation layer in ensuring that users of the data do not have access to data to which they do not have the appropriate rights e.g. PII or blinded clinical data. The specific design of the mediation layer is not proposed, but it is anticipated that it will provide for access via multiple mechanisms including RESTful services and more conventional direct data access by tools such as Cognos and JReview.

We also propose that the data warehouses contained within the federated warehouse all provide functionality for storing temporal or historical data. The types of temporal data use described above including the "official record" model required for grants can be supported by the "Bi-temporal" model[11] of temporal database. The Bi-temporal model associates two distinct temporal dimensions with each data record. One temporal dimension represents the time period for which the data was valid in the real world. The second temporal dimension represents the time period for which the data was known to the database. The bi-temporal model allows for queries that result in data that is as it was in the real world as of a specific point in time, as well as how the database would have appeared to a user at any given point in time. We propose that all data warehouses within the federated warehouse use the bi-temporal model.

### 3.5.4.7 Data Loading Layer

A major part of any reporting architecture is the layer that performs data loading. Traditionally this function has been performed by Extract Transform and Load (ETL) software. While this model is well understood it presents several challenges including the difficulty of managing changes to the physical data model in the source operational data stores, and the fact that physical database changes (updates, inserts and deletes) frequently occur without alignment to the business event that caused them. In conventional ETL systems, the ETL software looks at database transaction logs to determine what has changed and then makes changes to the temporal data in the data warehouse. This results in multiple temporal entries in the database with the temporal boundaries disconnected from the business event that caused them. To take a concrete example: As a clinician is working with a patient on a clinical trial, they may capture multiple data direct data observations into a clinical data management system. They may also order labs or other tests that generate additional clinical data that is disconnected in time – lab tests may take a day or two to run, reading of a MRI image may happens a week or more later. Using a conventional ETL system, this single event – observations and test results for a specific patient taken at a single visit – appear in the temporal data warehouse as distinct events associated with when the data is recorded in the clinical data management system and the link between them is to some degree lost[12]. Similarly, in the process of awarding a grant, a Grants Management Officer may create several updates to the grant award before finally awarding the grant. In the physical database model each update to the award generates a new transaction to be tracked in the data warehouse. There is no business value in recording these transactions; the only update with business significance is the final award when all data has been finalized.

---

[10] Kimball, Ralph; Margy Ross (2002). The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition ed.). New York: Wiley. ISBN 0-471-20024-7.

[11] http://en.wikipedia.org/wiki/Temporal_database#Bitemporal_Relations

[12] While the dates recorded for tests and observations allow for reconstruction of an accurate timeline, this process requires additional effort and is not directly reflected in the temporal data store and prevents the use of conventional bitemporal "As Of" queries, thus limiting the usefulness of the data.

This legacy ETL model is included in the architecture for data where there is no better option such as when a COTS product provides the source data store, and where reporting data does not have to be provided in near real-time. It should be anticipated that new or reengineered systems would load data to the data warehouse using the techniques described below.

For these cases the architecture proposes two models: One based on the event and messaging infrastructure and the other based on a direct push into the data warehouse. Both models seek to align data updates to the warehouse with business events and use the logical data model and canonical XML rather than physical table updates as the language for communication. The use of the logical data model reduces the need for changes to the processes that extract, transform and load the data to the data warehouse because changes are only required when the logical model changes – something that occurs much less frequently than changes to the physical since these include both business related changes, and changes made to tune or optimize the physical data model for transactional use.

This will be addressed in a forthcoming whitepaper that addresses specific proposals for key mechanisms of the architecture.

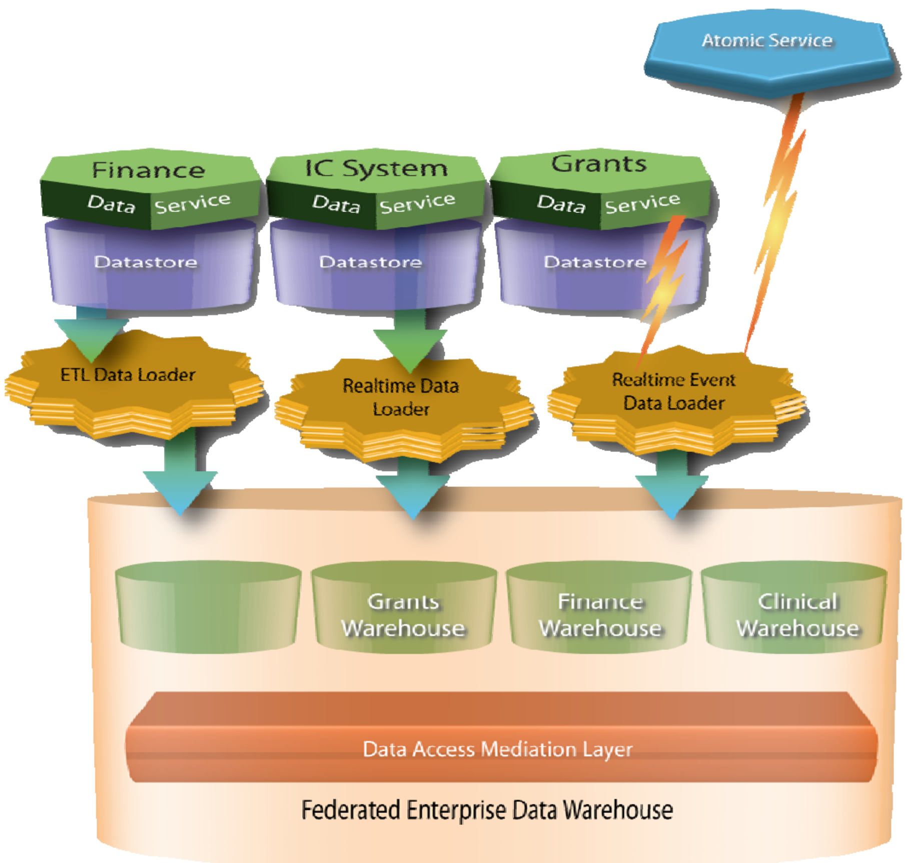


**Figure 7. Data warehouse data loading alternatives**

### 3.5.4.8 Reporting Data Services and Data Marts

Once the data is in the data warehouse reporting users have to then access the data in order to produce reports and perform analysis. In this area we identified two classes of report: Recurring and ad hoc. In recurring reports the report format and structure remains essentially the same from reporting period to reporting period with only the data reported changing. For ad hoc reporting, including business intelligence and data analytics, the format and structure of the report varies frequently. Examples of the recurring report include the population tracking reports of clinical trial enrollment and the budget reports generated for congress each year, while ad hoc reports include a broad category of management and analytical reporting.

For the recurring reports architecture we propose implementation of specialized data reporting services optimized for the specific report to be generated, for example the "Pop Tracking" report on clinical trial participation by gender, race and ethnicity the NIH produces for Congress biennially. The population tracking report would have an associated data reporting service that performs all the data aggregation and summarization required for the report and then delivers it to a user interface for formatting. These services would likely not be implemented as RPC style web services since the amount of data they deliver would be high and generally requires in memory buffering of the data. Alternative mechanisms including Representational State Transfer (REST[13]) or data delivery to web servers might also be appropriate.

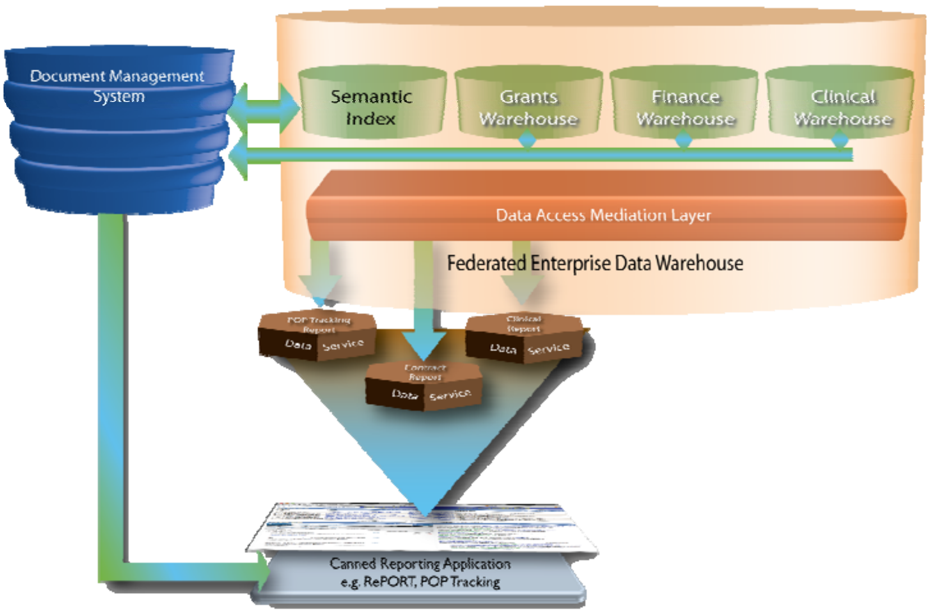


**Figure 8. Data reporting services for recurring reports**

For ad hoc reporting, we propose that the current trend to implement data marts continue. The data mart is designed to focus on a specific area of business or a specific set of users. The data mart extracts relevant data from the federated data warehouse and then stored it locally in a form optimized for the types of queries and analysis the users of the data marts require. This model allows for the kind of innovation that led to the development of the eSPA system (essentially a data mart) while gaining the benefits of using a single authoritative source for the data. Layered on top of the data marts would be a variety of tools for performing data analysis and visualization including both COTS products and custom analysis tools.

A key part of the support for business analytics proposed in the architecture is the application of multiple mechanisms for semantic analysis of structured, semi-structured (HTML, XML etc.) and non-structured (document) data. The model provides for a variety of semantic analysis tools to be used, using a variety of ontologies, and allows both the tools and ontologies to evolve over time. It is felt that such evolvability in this area is critical as it is a field of technology that is advancing rapidly. The use of such tools to generate a "Semantic Index" for the data allows for data selection to be performed using whatever evolving ontologies are available, independent of the definition of the ontologies used by individual systems. The utility of this type of model has been demonstrated recently by the reuse of the Research, Condition and

---

[13] http://en.wikipedia.org/wiki/REST

Disease Categorization (RCDC) budget category fingerprint data in the Query, View and Report (QVR) reporting tool to implement the ability to find research "LIKE" a given grant application. The future application for more advanced semantic analysis techniques using more other ontologies is seen as a major benefit to the NIH's ability to manage the scientific mission.

While the nVision system provides for storage of unstructured data, the mechanism used (Oracle BLOB data types), is not seen as a long term solution. As referenced in Figure 8 above, the authors see an enterprise scale document management system as a critical component of the overall strategy for reporting.

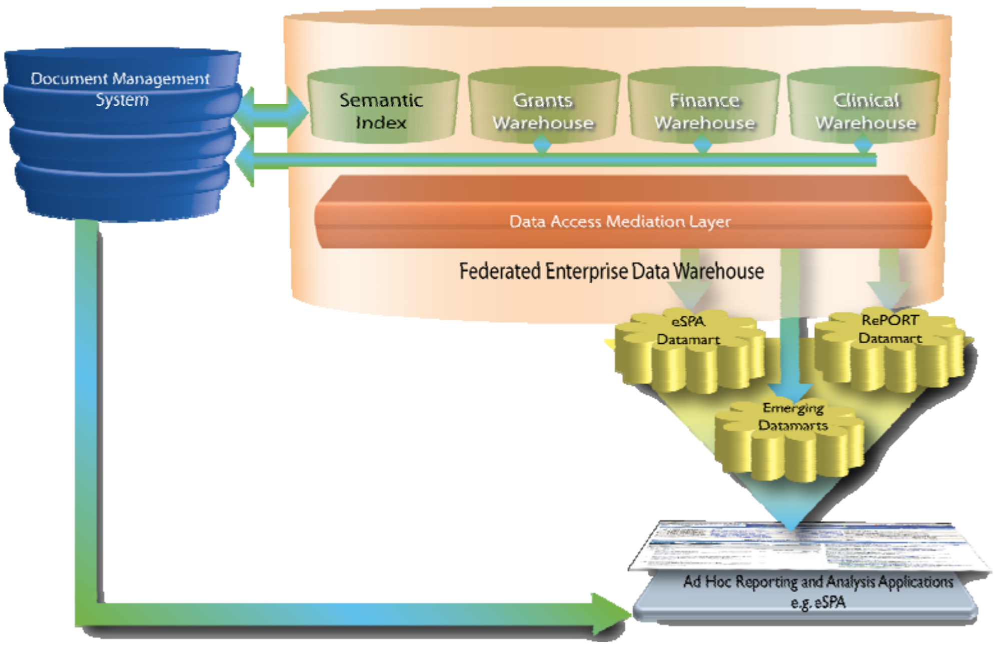


**Figure 9. Ad hoc reporting and Data Marts**

## 3.6 A Note on Security

The reader will have observed that the question of specific mechanisms for security has not been discussed in this whitepaper. The author recognizes that these are critical areas for discussion including:

- The mechanisms for carrying authentication and authorization information down through multiple layers of services, with those services potentially provided by multiple service providers.
- Application of the same authorization rules in the reporting segment of the architecture.

However, given the complexity and very specific expertise required in these areas, it is felt that this discussion should take place as one of the next steps from this NIHRFC.

## 3.7 A Note on Governance

Implicit in this architecture is the need for governance of services and systems. Without a solid model for governing and managing change and for funding the development of services that may be shared or enterprise in scope, a migration to this architecture will fail. A discussion of the governance and funding models for this architecture is an essential next step.

# 4 Conclusion

The preceding has been designed to prompt discussion amongst solution architects and others throughout the NIH. While there is little documented here that can be termed truly new and innovative, the combination of the identified structures and mechanisms is designed to facilitate the development of new and innovative solutions to real business problems. It is hoped that the specifics of the architecture will evolve over the coming months as solution architects provide their input to the Information Technology Architecture Office, particularly in terms of existing solutions that can be adopted throughout the NIH. To this end, the NIH IT Architect will be publishing further whitepapers to extend this document with concrete proposals for design patterns to implement the key mechanisms required to support this architecture.

## *4.1 Proposed Next Steps*

During the initial publication of this NIHRFC, the Information Technology Architecture Office held a series of "roadshow" presentations to gain feedback on the proposals made here. In general, there was a broad consensus that the ideas presented here were consistent with the needs of both IT and business stakeholders and there were many suggestions for next steps.

The following section captures the sense of these roadshow participants as to the next steps, along with some rooted in comments provided outside the roadshow meetings.

Develop a business case for the overall architecture with a focus on communicating to the business. It was felt that buy-in from business owners was critical to success and the adoption of the architecture will require investment from enterprise systems, CIT and ICs if it will be successful.

Engagement with EIMB to evaluate the proposed separation of reporting into a temporal, 3NF data warehouse and independent data marts using conformed dimensions and de-normalization with the goal on reaching consensus on an approach, and the publication of an NIHRFC standard for reporting and warehousing.

Collaborate with EIMB to more clearly define the role and implementation mechanisms for the proposed data mediation layer.

In order to keep the scope narrow, the ITAO will work with both NIH Enterprise Systems and ICs to identify potential business cases where the architecture could be applied. One recommendation from NCI is to focus first on the grants processes. NCI also recommended identifying business cases where the use of services is potentially bi-directional with both an enterprise system and IC systems acting as service providers. Once business cases are identified, develop concrete plans for proof of concept, pilot and implementation.

Examine standards being set by other organizations such as USDA, NSF, Vivo etc as the foundation for data communication with a particular focus on identifying data governance models and ontologies that we can adopt.

Consider the business case for expanding the SOA infrastructure provided by the Integration Service Center to include services such as run-time SOA governance, service monitoring and control to provide an environment where ICs can quickly and reliably deploy services.

# 5 References

The following references were used in the development of this informational NIHRFC. The reader is encouraged to review these as background to the proposals made in this document.


Inmon, W.H., Data Warehouse and Client Server Processing, Inmon Data Systems Inc., 2000

Simitsis, A., Wilkinson, K., Castellanos, M., and Dayal, U. 2009. QoX-driven ETL design: reducing the cost of ETL consulting engagements. In Proceedings of the 35th SIGMOD international Conference on Management of Data (Providence, Rhode Island, USA, June 29 - July 02, 2009). SIGMOD '09. ACM, New York, NY, 953-960

Kimball, R., Ross, M., Thornthwaite, W., Mundy, J. and Becker, B., The Data Warehouse Budget, Inmon, W.H., Inmon Data Systems Inc., 2000

W.H. Inmon, Building the Data Warehouse, Wiley Publishing Inc., Indianapolis, 2005

The Data Warehouse Lifecycle Toolkit, 2nd Ed., Wiley Publishing Inc., Indianapolis, 2008

http://it.toolbox.com/blogs/confessions/kimball-vs-inmonor-how-to-build-a-data-warehouse-10987, Viewed 8/20/2009

Golfarelli, M., Rizzi, S., and Vrdoljak, B. 2001. Data warehouse design from XML sources. In Proceedings of the 4th ACM international Workshop on Data Warehousing and OLAP (Atlanta, Georgia, USA, November 09 - 09, 2001). DOLAP '01. ACM, New York, NY, 40-47.

http://enterprisearchitecture.nih.gov/ArchLib/AT/TA/BrokerOperationalDataStoreWarehousePattern.htm

http://enterprisearchitecture.nih.gov/ArchLib/AT/TA/IntegrationBrokerSuitesBrick.htm

http://enterprisearchitecture.nih.gov/ArchLib/AT/IA/Integration/SDPrinciples.htm

http://www.eaipatterns.com/PublishSubscribeChannel.html, viewed 8/28/2009

Ballou, D. P. and Tayi, G. K. 1999. Enhancing data quality in data warehouse environments. Communications of the ACM 42, 1 (Jan. 1999), 73-78

Batini, C., Cappiello, C., Francalanci, C., and Maurino, A. 2009. Methodologies for data quality assessment and improvement. ACM Comput. Surv. 41, 3 (Jul. 2009), 1-52.

Golfarelli, M., Rizzi, S., and Vrdoljak, B. 2001. Data warehouse design from XML sources. In Proceedings of the 4th ACM international Workshop on Data Warehousing and OLAP (Atlanta, Georgia, USA, November 09 - 09, 2001). DOLAP '01. ACM, New York, NY, 40-47

van den Heuvel, W., Zimmermann, O., Leymann, F., Lago, P., Schieferdecker, I., Zdun, U., and Avgeriou, P. 2009. Software service engineering: Tenets and challenges. In Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems (May 18 - 19, 2009). PESOS. IEEE Computer Society, Washington, DC, 26-33. DOI= http://dx.doi.org/10.1109/PESOS.2009.5068816

Zimmermann, O., Doubrovski, V., Grundler, J., and Hogg, K. 2005. Service-oriented architecture and business process choreography in an order management scenario: rationale, concepts, lessons learned. In Companion To the 20th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (San Diego, CA, USA, October 16 - 20, 2005). OOPSLA '05. ACM, New York, NY, 301-312. DOI= http://doi.acm.org/10.1145/1094855.1094965

Zdun, U., Avgeriou, P., Hentrich, C., and Dustdar, S. 2008. Architecting as decision making with patterns and primitives. In Proceedings of the 3rd international Workshop on Sharing and Reusing Architectural

Knowledge (Leipzig, Germany, May 13 - 13, 2008). SHARK '08. ACM, New York, NY, 11-18. DOI= http://doi.acm.org/10.1145/1370062.1370066

Pautasso, C., Zimmermann, O., and Leymann, F. 2008. Restful web services vs. "big"' web services: making the right architectural decision. In Proceeding of the 17th international Conference on World Wide Web (Beijing, China, April 21 - 25, 2008). WWW '08. ACM, New York, NY, 805-814. DOI= http://doi.acm.org/10.1145/1367497.1367606

Gmach, D., Krompass, S., Scholz, A., Wimmer, M., and Kemper, A. 2008. Adaptive quality of service management for enterprise services. ACM Trans. Web 2, 1 (Feb. 2008), 1-46. DOI= http://doi.acm.org/10.1145/1326561.1326569

Landre, E., Wesenberg, H., and Rønneberg, H. 2006. Architectural improvement by use of strategic level domain-driven design. In Companion To the 21st ACM SIGPLAN Symposium on Object-Oriented Programming Systems, Languages, and Applications (Portland, Oregon, USA, October 22 - 26, 2006). OOPSLA '06. ACM, New York, NY, 809-814. DOI= http://doi.acm.org/10.1145/1176617.1176728

Ghalsasi, S. Y. 2009. Critical success factors for event driven service oriented architecture. In Proceedings of the 2nd international Conference on interaction Sciences: information Technology, Culture and Human (Seoul, Korea, November 24 - 26, 2009). ICIS '09, vol. 403. ACM, New York, NY, 1441-1446. DOI= http://doi.acm.org/10.1145/1655925.1656191

Teymourian, K. and Paschke, A. 2009. Towards semantic event processing. In Proceedings of the Third ACM international Conference on Distributed Event-Based Systems (Nashville, Tennessee, July 06 - 09, 2009). DEBS '09. ACM, New York, NY, 1-2. DOI= http://doi.acm.org/10.1145/1619258.1619296

Kim, H. 2002. Predicting how ontologies for the semantic web will evolve. Commun. ACM 45, 2 (Feb. 2002), 48-54. DOI= http://doi.acm.org/10.1145/503124.503148

# 6 Contact

To contact the NIHRFC Editor, send an email message to EnterpriseArchitecture@mail.nih.gov.

To comment on this or other NIHRFCs, visit the NIHRFC Discussion Area at http://enterprisearchitecture.nih.gov/YourPart/Collaborate/NRFCDiscussionArea.htm.

# 7 Security Considerations

As informational, this NIHRFC raises no security considerations. However, as mechanisms described herein are standardized as Principles, Bricks and Patterns, these are likely to require careful scrutiny with respect to security.

# 8 Changes

| Version | Date | Change | Authority | Author of Change |
|---|---|---|---|---|
| 0.1 | 7/2009 | Original Draft for Internal Review | | Alastair Thomson |
| 0.2 | 1/2010 | Included refinements from review by eRA Architecture | | Alastair Thomson |
| 0.3 | 8/2010 | Changes based on feedback from Jack Jones | | Alastair Thomson |
| 0.4 | 9/21/2010 | Revised for publication as an informational NIHRFC0070 | NIHRFC0001 | Alastair Thomson |
| 0.5 | 09/22/2010 | Formatting changes, spelling and punctuation edits | NIHRFC0001 | Zahra Ashraf |
| 0.6 | 09/23/2010 | Changes from review by ITAO Staff – Synchronization with SOA Principles, language changes, changes to examples | NIHRFC0001 | Alastair Thomson |
| 0.7 | 9/24/2010 | Reordered slightly for better flow based on feedback | NIHRFC0001 | Alastair Thomson |
| 0.8 | 6/12/2011 | Revisions from initial posting for comments and roadshows for Architects, Project Managers, Customer Relationship Managers, Enterprise Systems, NHLBI, NCI and NIAID | NIHRFC0001 | Alastair Thomson |
| 1.0 | 7/20/2011 | Revisions based on comments made following initial release of the document – See Section 11 below for a detailed disposition for comments. | NIHRFC0001 | Alastair Thomson |
| 1.0 | 8/31/2011 | Presented to the ARB | NIHRFC0001 | Alastair Thomson |

| | | | | |
|---|---|---|---|---|
| | | and received approval | | |

# 9 Author's Address

Alastair Thomson (Contractor)
National Institutes of Health
10401 Fernwood Road
MSC 4806
Bethesda, Maryland 20817
Phone: 301-594-6961
Email: Alastair.Thomson@nih.gov

# 10 Acknowledgements

The author would like to acknowledge the contributions of the following individuals to the development of this NIHRFC. Without their support, input, critiques and comments, this paper would not have been possible.

- Ms. Helen Schmitz, Acting NIH Chief IT Architect
- Mr. Jim Seach, Ms. Natasha Globus-Martin and Mr. Pete Morton, Office of Research Information Systems
- Dr. Paul Sheehy, NIGMS
- Dr. Tony Scarpa and Dr Cheryl Kitt, CSR
- Dr. Izja Lederhendler, ORIS
- Ms. Angela Thomas, Mr. Joe Klosky, Mr. Jay Shah, Mr. Bharath Kannan, Ms. Kiley Ohlson and Ms. Erin O'Leary, ITAO
- Dr. Sally Rockey, Acting Deputy Director for Extramural Research
- Dr. Della Hann, Deputy Director, OER
- Dr. Larry Tabak, Principal Deputy Director, NIH
- Dr. Norka Ruiz-Bravo, Former Deputy Director for Extramural Research
- Dr. Story Landis, Director, NINDS
- Dr. Jack Jones, NIH CIO

Portions of the research performed in connection with the development of this NIHRFC were funded by an evaluation grant from the Division of Program Coordination, Planning, and Strategic Initiatives #04-2007-OER.

# 11 Review Change Log

The following section records changes to this document made as a result of the NIHRFC comment period.

| Comment | Source | Changes |
|---|---|---|
| Comments on NIHRFC0070-NIH Whitepaper 01-24-2011.docx (Documented posted to NIHRFC discussion area) | Enterprise Information Management Branch | • Credit given for previous work and proposals from EIMB. The author recognizes that some misconceptions of the nature of nVision were included in this whitepaper and in the referenced reporting whitepaper. These misconceptions have been corrected and references to the other whitepaper have been removed. The author appreciates the input from EIMB in correcting these errors.<br>• Item 1: Corrected references to nVision as a “Financial” data warehouse<br>• Item 2: Removed references to data warehousing whitepaper containing the disputed assertion. The author understands that the model used by EIMB is close to that proposed in this whitepaper. As such, the model described is left unchanged and includes engagement with EIMB on this question as a next step following approval of this NIHRFC.<br>• Item 3: This comment references the removed data warehousing whitepaper. However, the author understands that nVision currently stores unstructured data in Oracle BLOB data fields. The assertion that it would be impractical to migrate data into nVision was based on this fact. The author agrees with the commenter that the right approach is to store these documents in a document management system and then link to the data. This is now explicitly referenced in the document.<br>• Item 4: The root of this comment is semantic i.e. the definition of “Data Mart” and “Data Warehouse”. The authors have chosen to define “Data Mart” as the databases used for reporting that are optimized using conforming dimensions and denormalization to optimize reporting. We have also chosen to define “Data Warehouse” as a database containing historical data in bi-temporal format, and third normal form. Based on these definitions, nVision is then a hybrid data mart/warehouse. It is the author’s contention that the optimal approach is a clear distinction between the two and hence that a separate temporal, 3NF data warehouse with meta-data management for temporal schema changes is proposed. The author understands that this opinion may differ from that of EIMB and the |

| | | |
|---|---|---|
| | | resolution of this difference is noted as one of the next steps.<br>• Item 5: Removed references to the document that refers to Star Schema. The author understands that nVision uses a variety of mechanisms.<br>• Item 6: Document has been updated to reflect a broader set of data sources than the financial data previously noted.<br>• The role of the Data Mediation Layer has been clarified – It was never the intent that this solely provide data via services and that more conventional access methods would also be available. The role of the mediation layer is to integrate across the data warehouses and this can be achieved by multiple means including materialized views and associative tables as proposed by EIMB. The author agrees with that assertion. With that said, the author does see a role for XML based data access via the mediation layer, though it would likely be used in conjunction with the physical integration. One aspect of this proposal is the potential for the mediation layer to be intelligent and to construct optimized reporting structures based on frequency of use. Such technologies are emerging and as such were not included in this paper and are mentioned here to provide some thoughts on future directions. Examination of this question is included as a next step.<br>• The intent of the discussion of the people data store is much more about OLTP systems than it is about reporting. The author agrees strongly with the observations of the commenter regarding the disparate systems used to manage people of different types. The proposal here is to integrate these better from an OLTP perspective, with the flow on effects to the ETL processes and consistency of data for reporting.<br>• In the concluding remarks, the commenter states that they believe that "nVision already fulfills much of what is required and has a head start in this strategy". The author does not dispute this assertion and notes that the proposal is not to replace nVision but is to extend reporting in a variety of ways. In some cases it will make sense to extend the investment in nVision, but it is the authors view that in others, a different approach may be better, particularly if the overall approach of separation of the data warehouse (storage) from the data mart (use) is followed. |
| Suggestion that this should not be an NIHRFC and | Rick Rodriguez, ORS | The direction to submit this as an informational NIHRFC came from the NIH CIO. |

| | | |
|---|---|---|
| instead be a whitepaper | | |
| ITAO and OCITA use | Steve Thornton, ITAO | References to OCITA corrected to ITAO |
| Where are the non-functional requirements that this architecture supports? Or is this reference to the "Business Goals"? | Steve Thornton, ITAO | The non-functional requirements are expressed at high level by the Architectural Goals (Section 3.2) |
| Although this is only intended to represent a proposed future state, I can't help but wonder what the sequence of implementing the different application components would be. Also, the future state only addresses new elements or reconfigured existing elements. It seems to be missing what will no longer exist in the future state – or any discussion thereof. | Steve Thornton, ITAO | The document is designed to describe the target state and not speculate on how we might achieve that state. The next steps noted from the roadshows included some suggestions for the beginnings of a transition plan. The lack of a list of elements that may no longer exist is consistent with that viewpoint. It is however an important question and will need to be addressed as the proposals in the document evolve. |
| How does this architecture help to meet IC specific needs (both functional and data extensibility)? | Steve Thornton, ITAO | The references to "Agile processes and data" is intended to reflect the need for ICs to be able to meet their needs while continuing to integrate with the overall architecture. Clarified in the text. |
| Assumptions - What assumptions have you made in the development of the application architecture? These will need to be validated to ensure the proposed future | Steve Thornton, ITAO | New assumptions section added |

| | | |
|---|---|---|
| state is a valid one. | | |
| Throughout the document there are references to other documents that can be "obtained from ITAO". Why not just include a hyperlink to them? | Steve Thornton, ITAO | At the time of writing, these additional documents had not been reviewed sufficiently for publication and as such ITAO desired to be able to provide them to specific requestors with caveats. |
| The repeated references that a white paper is available from the EA email address or through a forthcoming white paper becomes very distracting | Steve Thornton, ITAO | See above |
| It might be helpful to highlight what the other organizational change implications of the proposed architecture are. Perhaps this is best referenced in another concise document, but it is hard to buy-in to the validity of the approach without it mapped to some new organizational future state (structure, process, rules, culture, etc.). Unless of course, the organizational future state is the as-is state. But there are indirect references in the document that imply that this isn't the case. | Steve Thornton, ITAO | The question of organizational implications is important, but is beyond the scope of this document. |
| Perhaps we need a clear | Steve Thornton, ITAO | Added a new section for Definitions in the introduction |

| | | |
|---|---|---|
| definition also at the outset of “central system”, “enterprise system”, “local system,” and the similar delineations for the data. | | |
| Lacking in the discussion is the requirement of exception handling, error handling and internal controls.  The lack of internal controls is often a source of data quality issues and should be addressed at some layer in the architecture.  Garbage in-garbage out. Implicit in all of the layers are strong governance models and agreed upon business rules that don't exist today and which will be necessary for the success of the system | Steve Thornton, ITAO | While the document seeks to provide broad structural and behavioral patterns for the architecture the question of specific key mechanisms such as exception and error handling. These, and other mechanisms should be addressed in the development of patterns and standards that flow from this document.<br><br>A section on governance and has been added. |
| Specificity of title | Robert Malick, Custom Application Branch, CIT | The title has been changed to reflect a broader scope and a section has been added specifically to address the places where it is anticipated the architecture would and would not be applied. |
| Under supporting services there is the word "bio-" with nothing else after. Is it bioinformatics, biomedical informatics or something else? | Jim Seach, ORIS | Typo, removed |
| The definition of business owner used in this section | Jim Seach, ORIS | A definition of “Business Owner” has been added. |

| | | |
|---|---|---|
| probably needs to be explored. I'm not sure it is as simple as having one organizational unit designated as the data business owner. Although that would be ideal from an IT perspective, I'm not sure it represents NIH's current reality ala (sic) IC responsibility versus OD. This is especially true in the scientific program areas.<br><br>It is the tension between the needs of the central organizations and the ICs (and even within the ICS) and among the ICs that leads to the number of extension systems we have. | | A discussion of the diverging needs of ICs has been added to section 2.4. |
| 3.4.2 "Within constraints of legislation, regulation and policy, a user could then tailor the workflow to send the application to all the required review groups, whether in sequence or in parallel." - I'm not certain how likely it would be for an end-user to "tailor' a workflow. Perhaps they will select an approved alternate path? | Jim Seach, ORIS | Clarified in the document. |

| | | |
|---|---|---|
| Usually the workflow is the output of a policy decision. It may be that you are interchanging the roles of "workflow manager" and "user" here.<br><br>How would the presence of BPM work for local process needs? The implication here is not just for business process management but also for standardized business processes - and by extension roles.<br><br>To date NIH has not been very good at standardizing processes among the ICs and often times even within the ICs. Nor have some enterprise systems been particularly interested in accommodating local IC business needs - that are sourced from IC specific scientific program management and congressional reporting requirements. | | |
| Section 3.3.1 Interoperability constraints - Not just similar data | Jim Seach, ORIS | Suggestion incorporated into document |

| | | |
|---|---|---|
| models but also requires a consistent understanding and representation of business rules. | | |
| 3.3 Constraints - There seems to be only one constraint for interoperability described, which is way down in 3.3.1.1. There may be others embedded here (like data standards) but I think they need to be specifically addressed and highlighted. The goal is so that we can verify if the constraint is realistic and what the impacts to operations (current and future) might be as a result of the constraints.<br><br>3.3.1.1 Need to clarify the solution in plainer language. I believe the implication is that the interface is designed at the level of an entire executable business process as opposed to one step in the business process. Is that what this is saying? Also, there are some grammar issues that are making it confusing to understand. | Jim Seach, ORIS | Section re-written for greater clarity and extended to address the specific issues noted. |