

**An
Information Technology
Architecture
for
Emory University**

**Document 2:
Designing Emory's IT Architecture**

*Adopted by CIRT
February 20, 2002*

Committee on
Information Technology Architecture

March 6, 2002
Version 2.6.1

DOCUMENT REVISION HISTORY

Release	Description	Date
1.0	Incorporate ITA feedback on best practices	April 17, 2000
2.0	Incorporate PMM feedback and other sources	April 27, 2000
2.1	ITA feedback, remove domain-specific principles	May 18, 2000
2.2	Group principles, new title; respond to current state assessment	May 30, 2000
2.3	Incorporate PMM feedback; revise info about domains	June 6, 2000
2.4	Fix version number; revise B.1 note and B.4	June 8, 2000
2.4.1	Refinements to prose & ITA feedback on A.2	June 26, 2000
2.4.2	ITA meeting feedback: C.3, Next Steps	June 30, 2000
2.5	Address META Group feedback	July 26, 2000
2.5.1	HK feedback	Aug. 4, 2000
2.5.2	Update Next Steps, revise B.6, and other principles	Oct. 18, 2000
2.5.3	Revise B.6, add B.8	Oct. 24, 2000
2.5.4	Revise Next Steps, updated B.6	Nov. 15, 2000
2.5.5	Update domain description, examples to B.8, copy-edit, add glossary	Jan. 12, 2001
2.5.6	Update descriptions of domains, especially application & intranet	Jan. 18, 2001
2.5.7	Add Configuration & Implementation principles; adjust title capitalization	Feb. 14, 2001
2.5.8	Make edits, changes to wording & updates especially in preface, domains, next steps, and appendix 1	April 13, 2001
2.5.9	Number implications in B-7 & C-4; separate TOC & Revision pages	May 11, 2001
2.6	Change status to "Ready for Adoption"	Sept. 19, 2001
2.6.1	Changed status to Adopted; added copyright & additional bookmarks	March 6, 2002

TABLE OF CONTENTS

1. Preface	4
2. Conceptual architecture	5
3. Enterprise-wide IT architecture design principles	7
4. How the design principles support the requirements	9
5. Configuration and implementation principles	10
6. Enterprise-wide technical architecture domains	11
7. Next steps	13
Appendix 1. Justification and implications of the principles	14
A.1. Manage the IT architecture as a unity	16
A.2. Manage Enterprise IT infrastructure as an Emory asset	17
A.3. Manage the evolution of the architecture	18
B.1. Deploy systems that are prepared for change.....	19
B.1.1. Deploy scalable infrastructure.....	20
B.1.2. Make it easy to integrate new technology.....	21
B.1.3. Deploy modular, loosely coupled components	22
B.1.4. Enable as much reuse as is feasible	23
B.2. Reduce overall complexity.....	24
B.3. Standardize judiciously.....	25
B.4. Use process-event driven systems.....	26
B.5. Provide a common security layer	27
B.6. Facilitate access to IT resources.....	28
B.7. Document flows of information	29
B.8. Provide access controls for IT resources.....	30
C.1. Costing and pricing should promote desirable behavior.....	31
C.2. Develop staff competencies in areas of strategic importance	32
C.3. Consider outsourcing in the context of risk to Emory's future	33
C.4. Use industry standard solutions when feasible	34
Appendix 2. Technical Architecture Requirements	35
Appendix 3. Glossary	37

1. Preface

In Document 1, we agreed on a statement of what Emory wants to achieve, and how it will use an IT architecture to help its goals. At the end of that document, we arrived at a statement of the requirements our technical architecture should meet, e.g. “facilitate change in academic and administrative processes” and “provide a campus network that allows communication and exchange of information.”

In Document 2, we proceed to the next phase. Here we identify the design principles we will use as we evolve our new IT environment, and we identify the categories (called “architecture domains”) where we need to decide on policies, technologies, standards, etc.

Section 2 of this document introduces this “conceptual architecture,” Section 3 lists the design principles that we have selected, Section 4 shows how these principles support the architecture requirements from the previous document, and Section 5 derives additional principles for configuration and implementation. Section 6 proposes the specific architecture domains, and Section 7 gives next steps. Appendices provide more detail about the design principles and a convenient list of the architecture requirements from Document 1.

Once the design principles are in place, the next phase of the process results in principles, technologies, standards, etc. for each architecture domain.

2. Conceptual architecture

Document 1 derived requirements of the technical IT architecture to support and enable Emory's mission, goals, priorities, strategies and strategic information requirements. The next level of detail, called the Conceptual Architecture, is intended to articulate the important shared vision and values for the technical architecture process and content. It provides a high-level set of enterprise-wide principles, strategies, goals, practices, policies, and standards to guide the design, construction, deployment, and management of enterprise-wide and campus-wide Information Technology (IT) infrastructure and systems at Emory that support Emory's technical architecture requirements. The Conceptual Architecture also identifies categories (called "architecture domains") where we need to decide on more specific policies, technologies, standards, products and configurations.

The goal of the IT architecture is to create an IT environment that can respond to Emory's needs, delivering needed capability in the needed time frame. What gets in the way of timely response?

Example 1. A university has systems and databases for reserving rooms, assigning classrooms to classes, maintaining standard room and building names, keeping up with the location of equipment, and storing the office address of staff. Over the years, as each system was added, a separate, unique interface was created to each of the other systems' databases to keep them synchronized, resulting in 4 interfaces per system and 10 different pairs of interfaces in all. With this arrangement, each system must know which other systems to contact and when. Also, each system duplicates the processing effort to manage communications and handle errors when a database is not responding. Changes to a database potentially impact all the systems. The complexity of this arrangement makes adding another system or making changes difficult.

Existing conditions and methods can impede responsiveness. The independent acquisition of many applications in the past can produce an environment in which integrating new applications is so complex as to inhibit responding quickly enough. In addition, applications, servers, networks, and other IT infrastructure must continually be added, upgraded, or expanded to meet organizational needs. While general requirements can be anticipated (such as a need for more disk space), the details (such as how much disk space) may be impossible to predict.

The overall approach to creating responsive information technology is to:

- Anticipate needs based on organizational goals and environmental forces and trends.
- Investigate IT solutions to anticipated needs.
- Use tested and successful principles and practices to create a flexible IT environment that can be quickly adapted as needs change.
 - ◇ For example, reducing complexity can enable faster response by making it easier to understand the impact of a change. One way to reduce complexity is through standardization. Having a standard set of clients can lead to a smaller supported set to test, making for faster testing of a change to a server. Having a standard interface for exchange of data between applications makes for faster integration of new applications.
 - ◇ Providing room to grow and having extra equipment at the ready allows for a quicker increase in capacity.
 - ◇ Creating reusable components with standard interfaces that can be used as building blocks for systems reduces the time to create new solutions.

To improve our IT environment, the architecture must affect our enterprise-wide Information Technology decisions at Emory. Our consultant's experience has been that it is impossible to specify enough rules in enough detail to cover all decisions that will arise. So our approach is to create a set of principles to guide us when we face choices. We do not intend these principles to be used blindly, because each situation can have its own particular circumstances. However, because the principles generally provide better results more often than not, decision-makers should expect to justify an exception if they wish to disregard them.

Example 2. A university had to choose between two application systems to be used university-wide. One of them clearly best met the functional requirements, but an architectural review revealed that this system was difficult to integrate and extend, while the other was easy. To obtain the most future flexibility and minimize integration problems, the university chose the easier to integrate and more extensible system, then used its extensibility to add the missing functionality.

3. Enterprise-wide IT architecture design principles

In the following summary of the Enterprise-wide Conceptual Architecture principles, the titles are provided only for convenience. Read the actual statements of the principles to learn caveats and nuances. Consult Appendix 1 on page 14 for scope, usage, justification and implications.

A. Governance principles

- A.1. **Manage the IT architecture as a unity.** The planning and management of Emory's technical architecture should be unified. (Page 16)
This means that Emory's IT departments and unit leaders have a common vision of what it means to have Emory-wide architecture and standards, and there is a process to implement and enforce a consistent use of architecture and standards across Emory.
- A.2. **Manage Enterprise IT infrastructure as an Emory asset.** Enterprise applications, data and IT infrastructure should be managed as Emory assets. (Page 17)
Corollaries: a. Emory's enterprise IT organizations should be responsible for managing the common infrastructure. b. Whenever practical, all enterprise infrastructure projects should use Emory standard project management methodology and tools to enable faster and more certain project operation. c. Stewards should be identified and responsible for each of the architecture domains. d. All enterprise IT project and architecture documentation should be accessible via Emory's Intranet. e. To the extent feasible, the components of the enterprise infrastructure should be selected or designed to allow for remote, highly automated management.
- A.3. **Manage the evolution of the architecture.** The evolution of the enterprise-wide technical architecture should be planned and governed across the enterprise, with at least a yearly review at a point in the budget cycle that allows it to influence the proposal of projects for funding. (Page 18)

B. Principles for responsive infrastructure and applications

- B.1. **Deploy systems that are prepared for change.** IT systems, services and infrastructure should be designed to support the practice of anticipating likely types of future requests and having capability at the ready to respond quickly, even though the details and timing may be unknown. (Page 19)
Note: Although the following principles are instances of B.1, they are considered to be just as important as B.1 and the other principles. Also B.1 is not limited to them.
 - B.1.1. **Deploy scalable infrastructure.** Any portion of the enterprise-wide technical infrastructure should be scalable; that is, able to expand quickly and economically in capacity, capability, scope, availability, reliability and maintainability as needed. (Page 20)
 - B.1.2. **Make it easy to integrate new technology.** The architecture should promote easy integration of new IT devices, systems and solutions with the existing infrastructure. (Page 21)
 - B.1.3. **Deploy modular, loosely coupled components.** IT solutions and infrastructure should be engineered with a bias toward using highly discrete, loosely coupled components. (Page 22)
 - B.1.4. **Enable as much reuse as is feasible.** All aspects of the architecture should be reusable and be reused to the extent feasible. (Page 23)
- B.2. **Reduce overall complexity.** The enterprise IT infrastructure should be no more complex than it has to be to serve Emory's mission. (Page 24)

- B.3. **Standardize judiciously.** Standards should be seen as a means to support university priorities, the requirements and principles of the IT architecture, or the common good at Emory. Standards that limit individual choices of personal technologies should require justification on the above basis. (Page 25)
- B.4. **Use process-event driven systems .** Enterprise IT systems should be process-event driven rather than batch-oriented. After acting on an event, any results for use by other systems should be made available immediately as an event. (Page 26)
- B.5. **Provide a common security layer.** The infrastructure should present a consistent, uniform and robust security layer across all infrastructure components regardless of their physical location. (Page 27)
- B.6. **Facilitate access to IT resources.** Provide a way to find IT resources that indicates information about the contents of the resources and how to request access to them. (Page 28)
- B.7. **Document flows of information.** The flow of information into, out of, and between components of the architecture should be documented and made available for access via the Emory Intranet. (Page 29)
- B.8. **Provide access controls for IT resources.** The IT architecture should provide access controls that can make public the IT resources that should be public, and make private the IT resources that should be private. It should further classify private resources as needed to take into account differing security needs. (Page 30)

C. Principles for resource allocation

- C.1. **Costing and pricing should promote desirable behavior.** IT costing and pricing should promote the architecture's goals, principles and practices. Costing and pricing should also encourage and facilitate action for the common good, university-level management of the common infrastructure, rational and informed decisions, manageable and affordable infrastructure expansion, computation of total costs, and equitable cost allocation. (Page 31)
Corollaries: a. IT projects should be evaluated and alternatives compared using total cost and benefit over the full life cycle, including cost of training, support, maintenance, entry and exit, payback or return on investment, and adjustments for risk. b. All infrastructure services should define and track the following metrics over time: performance, health, available capacity, who is using the service and how much.
- C.2. **Develop staff competencies in areas of strategic importance.** Emory's internal core IT competencies must be more fully developed in areas that contribute to Emory's distinctiveness, as well as other areas of strategic importance to Emory. (Page 32)
- C.3. **Consider outsourcing in the context of risk to Emory's future.** Outsourcing an IT service must not put Emory's future at unacceptable risk. Whether a service is outsourced or not, all IT service planning and management must be done internally. In particular, Emory's enterprise architecture development must be done internally. (Page 33)
- C.4. **Use industry standard solutions when feasible.** Enterprise information technology selection and infrastructure decisions should be based upon industry proven and supported components, methods, standards and tools. Custom solutions should only be used when feasible industry standard alternatives cannot be affordably obtained, or when outside support for such a solution does not provide needed changes fast enough. Customization of an industry standard product or system is a custom solution. (Page 34)

4. How the design principles support the requirements

Principles support requirements		Technical Architecture Requirements														
		1: Facilitate Change	2: Workflow	3: Integration	4: New Devices	5: Authorization	6: Meta data	7: Protection	8: Data Types	9: Internet & I2	10: Volume of Info.	11: Data Facility	12: Interoperability	13: Accessibility	14: Network	
Conceptual Architecture Principles	A.1: Manage architecture as a unity	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
	A.2 Infrastructure as University asset	■		■		■	■	■		■	■	■	■	■	■	■
	A.3 Manage architecture evolution	■	□	■	■	□	■	□	■	■	■	□	■	□	■	■
	B.1 Systems prepared for change	■	■	■	■				■	□	■		■	■	■	■
	B.1.1 Deploy scalable infrastructure	■	■	■	□	□			■	■	■		■	■	■	■
	B.1.2 Easy technology integration	■	■	■	■	□		□	■	■	□	□	■	□	■	■
	B.1.3 Modular loosely-coupled	■	■	■	■	□			□		□		■		■	■
	B.1.4 Enable reuse	■	■	■	■	■							■	■	■	■
	B.2 Reduce overall complexity	■	□	■	■	□	■	□	■				■	■	■	■
	B.3 Standardize judiciously	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
	B.4 Process-event driven systems	■	■	■	□	■	□	□	□		■					
	B.5 Common security layer	■	■	■	■	■	□	■	□	■	■	■	■	■	■	■
	B.6 Access to IT Resources	□	□	□	□						■	■	□	■	□	□
	B.7 Documented flows of information	□	■	□	□	□	□	■	□		□		■	■		
	B.8 Access Controls		□	□		■		■				□		■		
	C.1 Support by costing and pricing	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
	C.2 Strategic staff competencies	■	□	■	■	■	■	■	■	■	■	■	■	■	■	■
C.3 Outsource only non-strategic etc.	■															
C.4 Use industry standard solutions	■	□	■	■	□	□	□	■	■	□	□	■	■	■	■	

Requirements support Strategies		Technical Architecture Requirements													
		1: Facilitate Change	2: Workflow	3: Integration	4: New Devices	5: Authorization	6: Meta data	7: Protection	8: Data Types	9: Internet & I2	10: Volume of Info.	11: Data Facility	12: Interoperability	13: Accessibility	14: Network
Strategies	1: Research Excellence	■	□	□	■	□	■	■	■	■	■	■	■	■	■
	2: Teaching Excellence	■	□	■	■	□	■	□	■	■	■		■	■	■
	3: Intellectual Community	■		□	□	■	□	□	□	□			■	■	□
	4: Interdisciplinary Scholarship	■	□	■	■	■	■	■	■	■		■	■	■	■
	5: Internationalization	■		■	□	■	■	■	■	■			■	■	□
	6: Standing and Reputation	■			■					■			□	□	■
	7: Increased Resources	■	■	■	■		■		■	■	■		□		

5. Configuration and implementation principles

The following additional principles follow from the previous principles.

D. Basic configuration principles

- D.1. When choosing a product to be part of the architecture, avoid the risk of using a leading-edge product until it is proven in the field. Test leading edge products that are under consideration to determine how well they work. Thoroughly test products before putting them into production. Also use prudence related to the size of the risk, ability to inspect the function of the product and see how well it performs, complexity of the product, and non-obvious ramifications.
- D.2. Choose systems for the enterprise architecture that support Web-enabled applications. The goal is to simplify the user environment and increase accessibility by having a common interface. That is currently a web interface.
- D.3. To the extent possible, choose enterprise systems that support Internet standards, especially access via TCP/IP. This helps move toward a standard, small set of protocols.
- D.4. Reduce support costs by reducing the number of supported common software packages.
- D.5. Consolidate systems management tools to gain better integration and cost control.
- D.6. Reduce support costs by reducing the number of supported vendors.
- D.7. Select configurations that minimize the support labor required.
- D.8. Select cost-effective configurations, but err on the side of over-capacity rather than over-supporting.

E. Basic implementation principles

- E.1. Choose a system or product after considerations of university mission and priorities are determined. Make system decisions after the university makes some basic determinations about the following items:
 - E.1.1. *Growth*: Must the system accommodate substantial growth parameters, beyond the current data and transactional volumes?
 - E.1.2. *Scalability*: Must the system, in the light of the suggested growth, be able to start small and grow continuously in small increments. Alternatively, would it be acceptable if growth happens in major, discontinuous increments.
 - E.1.3. *Lock-in*: Will it be required to move usage from one system to another?
 - E.1.4. *Openness*: What are the implications to the university if a proprietary system is used, thus eliminating the option to choose systems components from many vendors?
- E.2. Consider several criteria when selecting a system or product by including some of the following: financial viability of the vendor; compatibility with or support by other products; ability to meet the university requirements; adherence to the predetermined standards; cost of the supporting infrastructure; availability of skill sets (internal versus external) to support this system and to use this system; and service terms and conditions.
- E.3. Before deploying a new technology or product, see if the life of a technology or product already in use can be extended at a reasonable cost to meet the need.

6. Enterprise-wide technical architecture domains

The following is an initial list based on sources recommended by META: How IT is organized at Emory, topics from the Current State Assessment, current trends, and META Group documents. ***This list is subject to change*** as architectures are developed for each domain and as categories of applications arise that are so strategic that they need their own architectures.

Basic domains

Domain Name	Definition
Data and information management	The Data and information Management Architecture defines the components and standards for accessing, exchanging, modeling, storing, converting, organizing, and distributing data and information. Product and technology categories governed by this domain include databases, data warehouses, data marts, data repositories, report writers, document management, imaging, data modeling tools, data replication tools, data administration tools, data extract tools, data movement tools, and data cleansing tools.
Platform	The Platform Architecture defines the technical computing components of the infrastructure: the client and server hardware, the operating systems executing on that hardware, and the database environments and interfaces.
Network	The Network Architecture defines the voice, video and data communication infrastructure for the distributed IT environment. It covers structure and topology, bandwidth management, cable plant, electronics (hubs, PBX, routers, switches), protocols (access, routing, naming, DNS), carrier services (frame relay, leased channels, ATM, WAN, SONET ring), wireless, and Internet connections.
Distributed Environment Management	The Distributed Environment Management Architecture defines how the hardware and software components of the environment will be controlled. It focuses on issues of configuration management, fault detection and isolation, testing, capacity planning, performance management, problem reporting, upgrades, change control, asset management, business recovery, and help desk.
Security	The security architecture defines the component processes, data feeds, and deployed hardware and software to electronically protect, preserve and control access to Emory's information technology assets. For the purposes of the security architecture, these assets are considered to be the computing, telecommunications, video, and associated network facilities of the university. These assets include but are not limited to computers, printers, and software, administrative and research data, and the campus-wide network with its routers and connection(s) to the Internet. Also included are those assets that the university does not own, but which are in its custodial care. This domain covers such technologies as identification, authentication, authorization, firewalls, cryptography, intrusion detection, vulnerability detection, and virus prevention, detection, and removal.
Integration	The Integration Architecture defines the components to create uniform platform and network-independent mechanisms to integrate applications. The systematic tying together of disparate applications is known as enterprise application integration. The main technology for this domain is middleware (APIs for message exchange). Middleware is sometimes called plumbing,

	<p>because it connects two sides of an application and passes data between them. A common application of middleware is to allow programs written for access to a particular database to access other databases. Middleware is distinct from import and export features that may be built into one of the applications. Examples of types of products that would be governed by this domain: DCE environments, RPC systems, Object Request Brokers (ORBs), message passing, Open Database Connectivity (ODBC), Java Database Connectivity (JDBC), Universal Database Connectivity (UDBC), Messaging-Oriented Middleware (MOM), Transaction Processing (TP) Monitors, and database gateways.</p>
--	--

Applied domains

Application	<p>The Application Architecture defines how applications are designed, procured, and developed, how they cooperate, and where they reside. It covers application development tools and languages, testing tools, Enterprise Resource Planning (ERP) applications, Customer Relationship Management (CRM) applications, project management, case tools, and office productivity applications.</p>
Intranet and e-Commerce	<p>The Intranet Architecture defines the technologies, standards, and guidelines for seamless, platform-independent enterprise communications and universal access to information. It includes web browser, inter/intranet servers (mail, web, news, proxy, ftp), intraware (middleware for inter/intranet), content management, web database connectivity, search engines, languages, Java development tools, web authoring tools, e-commerce, and web-based portals.</p>
Online Learning	<p>The online learning architecture provides standard components and tools for online teaching and learning. It covers online course and instructional delivery (electronic distribution and collection of course materials, testing, experimentation, collaboration, and interaction with classmates and teachers), course administration, and content distribution and management (distribution systems that make Emory's academic content available to faculty and students both on campus and off).</p>
Collaboration	<p>The Collaboration Architecture defines the environment for automating conversation-focused and activity-focused aspects of human/computer interactions. It includes workflow, rule engines, collaborative tools and groupware (email, interactive messaging, shared calendars and scheduling, A/V conferencing (audio, video, text, chat, whiteboard, shared computing), listserv, bulletin boards, newsgroups), mail gateways, and intelligent agents.</p>
Directory	<p>The directory service architecture defines the infrastructure to create a single, unified naming scheme that uniquely identifies entities, and provides the capability for a network-attached service, system, or device to use an entity's name to obtain information about the entity.</p>

7. Next steps

Current Phase (Phase 3): Conceptual architecture phase

During the current phase 3 we have refined the conceptual architecture principles and the choice of domains and their definitions through consultation, discussion and feedback. This work overlapped the preparation for and the beginning of the next phase (phase 4). Phase 3 will end with the adoption of the conceptual architecture principles and domain definitions. However, the domain definitions are subject to further refinement during phase 4.

Next Phase (Phase 4): Domain architecture phase

The next phase (phase 4) works out the details of each domain at least to the point of defining its subdomains, if any, and specifying design principles for the domain and its subdomains. These "domain principles" are partially based on how the conceptual principles apply in that domain. Additional domain principles are derived from applicable principles of other domains. For example, principles for managing data and information are potentially applicable to any domain in which data is collected, organized, stored and accessed.

The work will be done by task forces that are knowledgeable in the domain's topic and that represent the diverse perspectives of the widespread Emory community. These task forces will disband when their work has been reviewed by appropriate campus groups and accepted by the ITA on behalf of university leadership. New task forces will be formed as needed in the future for additional domain architecture work.

The first two domains chosen for development were Directory Service and Security. Their development began before the completion of phase 3, because they were needed right away to enable progress in many other areas. Overlap with phase 3 was not a problem, since by then the phase 3 principles had been considerably reviewed and refined by then.

Although it is best if all the domains are developed at the same time so that the results of one domain can be used where applicable in other domains, in fact at most two at a time are being done due to resource constraints (such as available people and time). While not the preferred method, this approach does have the advantage of allowing iterative refinement of the domain development process.

Phase 5: Planning, implementation, and ad hoc usage phase

As the Domain Architectures are developed, they specify the target IT architecture in enough detail to begin assessing the current IT infrastructure relative to that target and planning for migration to it. An important part of that planning will be estimating migration costs and their impact by budget year, so that units can build any resulting additional expenses into their future budgets.

By the end of Phase 4 there is enough detail to use the target architecture to provide guidance in day-to-day IT activities, such as IT procurement, selecting software, etc. (called "ad hoc use"). Although such use is less holistic than gap analysis and migration planning, it is still of great value, because it increases the likelihood that current work will help move toward the desired future state by maintaining a focus on appropriate principles and standards.

Appendix 1. Justification and implications of the principles

Definition: The Conceptual Architecture is a framework of high level IT categories, along with principles, recommended practices, guidelines, policies, and standards that direct the design, construction, deployment, and management of enterprise-wide and campus-wide Information Technology (IT) infrastructure and systems at Emory in support of Emory's IT architecture requirements. By "infrastructure" is meant the foundation on which to build IT solutions for Emory business problems and productivity systems. Thus infrastructure consists of the physical components of a computing setup: the wiring, routers, switches, operating systems, middleware, mainframes, servers, and sometimes includes desktop machines.

Objective: The objective of the conceptual architecture is to guide Emory in the implementation of an enterprise-wide technical infrastructure that creates an enterprise-wide IT environment that is aligned with Emory's goals, priorities, and strategies, and that can deliver needed capability cost-effectively and affordably in the needed time frame.

Audience: The audience of the conceptual architecture is anyone whose plans cause changes to the IT infrastructure, who designs or implements such changes, or who wants to understand the implementation rationale. Thus the audience can include members of university leadership, unit and enterprise IT staff, and those who just use IT at Emory.

Organizational Scope: The architecture addresses the needs of the "University" and its community, that is, the enrolled students and the employed faculty and staff of Emory University no matter where they reside, keeping in mind that education and research involving faculty and students of the Health Sciences also occur at Emory Hospital, Crawford Long Hospital, Emory Clinic, satellite clinics, the V.A. Medical Center, Grady Hospital and its buildings, and Emory West. The term "Emory" alone includes all its sites. Healthcare clinical priorities and architectural requirements will need to be addressed elsewhere. Both architectures will need to address issues of interoperability between the University and Healthcare insofar as University Health Sciences are concerned.

Level of applicability: The conceptual architecture applies to the enterprise-wide infrastructure. It is applicable to all changes to the IT infrastructure and to all parts of the enterprise-wide solution life cycle, including system design, construction, deployment, management, product selection, systems integration, database development, standards, and configurations relating to enterprise-wide information technology at Emory. Unit IT decisions that are of local benefit and do not adversely affect university resources are normally at the discretion of the unit. However, local decisions should consider the effect on the overall system. To the extent that Emory units adapt the enterprise-wide architecture for local use, the conceptual architecture can provide a basis for common local architecture across the units of Emory.

Usage: We do not intend these principles to be used blindly, because each situation can have its own particular circumstances. In addition, the principles can sometimes overlap or conflict, requiring tradeoffs. IT experts must ultimately recommend the approach to take. With that proviso, because the principles generally provide better results more often than not, decision-makers should expect to justify an exception if they wish to disregard them.

Status: During the period when the principles are being established, each principle has a status as follows:

Proposed	Someone has suggested adopting the principle, and the architecture committee has agreed to consider it, but has not yet discussed it.
Under discussion	The architecture committee thinks the principle needs further discussion (modification is likely).
Standards track	The architecture committee has discussed this principle and proposes to make it a standard.
Adopted	The principle has been reviewed by appropriate campus groups and approved by university leadership.

A.1. Manage the IT architecture as a unity

Status: Adopted

The planning and management of Emory's technical architecture should be unified.

This means that Emory's IT departments and unit leaders have a common vision of what it means to have Emory-wide architecture and standards, and there is a process to implement and enforce a consistent use of architecture and standards across Emory.

This is a prerequisite even when IT responsibility is decentralized.

Justification

- A common vision of what it means to have a process to implement and enforce the consistent use of standards across Emory simplifies governance. A common understanding makes it easier to establish priorities, principles, product standards, and configurations for the good of Emory as a whole, especially when such decisions are not optimal for a particular unit.
- A unified approach, as opposed to just a centralized approach, helps to accommodate the diversity of perspectives present in a diverse organization. Broad representation enables decisions of design and product selection to take into account the full context and implications of the decisions.
- An enterprise view helps to identify situations where IT architecture could potentially add value. Examples include situations where process and data sharing occur across the enterprise, and where the units working together can achieve economies of scale.
- A unified approach facilitates a systemic view in which organizational processes, their supporting information technologies, and the groups that implement and change the processes and technologies are seen as interdependent parts. This approach helps to ensure that the IT architectural domains are designed, deployed, and modified in parallel with each other, remain linked to the processes they are intended to support, and remain logically consistent. Logical consistency makes for easier understanding, management, and change.

Implications

1. A unified approach will require organizational structures that enable cooperative decision-making and cooperative enforcement.
2. Organizationally, unification will have to represent the perspectives and needs of many constituencies informally or formally, including enterprise, unit, and central IT.
3. The extent of standardization and unification will depend on the extent to which information is shared across Emory for making university decisions.
4. Core infrastructure resources will need to be designed to be reusable and expandable at the local level.

A.2. Manage Enterprise IT infrastructure as an Emory asset

Status: Adopted

Enterprise applications, data and IT infrastructure should be managed as Emory assets.

Corollaries

- a. Emory's enterprise IT organizations should be responsible for managing the common infrastructure.
- b. Whenever practical, all enterprise infrastructure projects should use Emory standard project management methodology and tools to enable faster and more certain project operation.
- c. Stewards should be identified and responsible for each of the architecture domains.
- d. All enterprise IT project and architecture documentation should be accessible via Emory's Intranet.
- e. To the extent feasible, the components of the enterprise infrastructure should be selected or designed to allow for remote, highly automated management.

Justification

- Enterprise-wide IT resources need to be managed for the common good.
- Emory's operations, strategic initiatives and programs depend on its enterprise IT resources and infrastructure.
- Stewards are needed to ensure that all parts of the infrastructure receive attention.
- Infrastructure requires use of reliable products and methods. It must be maintained and upgraded.
- IT project and architecture documentation is a resource of enterprise interest, and is consistent with principle "B.6 Facilitate access to IT resources. "
- Infrastructure requires active management. Being able to automate the management and do it remotely (while maintaining needed security) is necessary to control costs.

Implications

1. A separate project management office responsibility may eventually be needed to ensure that infrastructure projects are managed according to standard and adhere to the technical architecture.
2. The enterprise IT infrastructure will need to be engineered to provide data and error reports, and permit active probing and management.
3. Enterprise components will need to have features such as hot swapping of hardware components, load sharing and balancing, and redundancy.
4. Plans should provide for a level of redundancy appropriate for the visibility of a particular enterprise infrastructure component, determined by such indicators as the number of components or the amount of capacity that would be disabled by the component's failure.
5. Emory's assets will need to be identified and classified so that needed levels of security, maintainability and availability can be decided.

A.3. Manage the evolution of the architecture

Status: Adopted

The evolution of the enterprise-wide technical architecture should be planned and governed across the enterprise, with at least a yearly review at a point in the budget cycle that allows it to influence the proposal of projects for funding.

Justification

- Campus needs and available technologies change continually, so planning needs to be ongoing and the architecture needs a regular review.
- The architecture review needs to happen soon enough before budget setting to allow its implementation needs to influence the proposal of projects for funding.
- There must be a way to add to and remove from the architecture.
- Establishing IT architecture takes time and involves a lot of change.
- Work on architecture can add value by bringing new technologies and the opportunities they provide to the attention of decision-makers. The earlier a technology can be understood, the more quickly it can be put to use and the more effective its use can be.
- Architecture needs to be treated as a process rather than a one-time event or project and should result from a collaboratively produced and documented consensus.
- Changes to the architecture must be well thought out. Good change requires collaboration and collective planning.
- Prioritization and reprioritization are necessary across all IT initiatives. Short term versus long term must be constantly re-examined.
- Dependencies of portions of the infrastructure on other portions must be understood and supported.

Implications

1. The architecture will need to provide for the level of the infrastructure to be increased in a manageable way.
2. There will need to be a process by which architecture is developed and refreshed. Processes will be needed for adding, removing, phasing in and phasing out products, technologies, standards, policies and procedures.
3. The primary design goal of responsiveness must not be compromised, because giving equal priority to more than one design goal typically results in mediocre results.
4. Architecture needs to be considered as a strategic investment and its value measured by return on equity.
5. Decisions on the acquisition, implementation, deployment, and continued viability of any piece of IT infrastructure should consider its economic sustainability (see C.1) and contribution to mission-related competitive advantage.
6. To keep the cost of evolution and change under control, the components must have affordable exit costs. In particular, to avoid the cost of lock-in, it must be possible to load information and business rules into and dump them from components as applicable.

B.1. Deploy systems that are prepared for change

Status: Adopted

IT systems, services and infrastructure should be designed to support the practice of anticipating likely types of future requests and having capability at the ready to respond quickly, even though the details and timing may be unknown.

Example 3. When a CIO spotted a trend to increase speed and convenience of service delivery by giving users access from any computer anywhere at any time, he sent some of his top developers for training on web access to mainframe applications. Then he arranged with one of the school's senior managers to do a pilot project of low visibility. Working with an experienced consultant, the IT organization acquired the needed technologies and skills. When the registrar mentioned concern about speeding up registration, the IT organization was able to give a realistic ball-park estimate for a system that enabled students to register via the web, and then implement it within time and budget. When thousands of students unexpectedly tried to access the scheduling application at the same time, the staff and systems were able to respond, because the mainframe and other servers had excess capacity or were poised for rapid upgrade.

Justification

- This approach allows faster response. In particular, it allows new capabilities, additional capacity, increased processing speed, or changes to the configuration, distribution, engineering, or deployment of components and systems to occur more quickly.

Implications

1. When designing IT infrastructure, engineer *in* the ability to change and engineer *out* inhibitors to change.
2. Design reviews will likely be needed to ask what changes might be needed later and how to design within the current scope of work to allow making those changes quickly when needed.
3. Infrastructure and systems will need to be designed to be extensible. Generally that will involve making them more modular and providing more "hooks" or interfaces.
4. Having additional capacity at the ready may involve paying for unused capacity and thus cost more.
5. The amount of headroom should take into account expected life span, likely demands, cost trends, and the cost to add more headroom now versus cost to add it later.
6. Servers should be purchased with room for growth; additional processors, memory, and disk drives should be stocked locally or be quickly available. Pathway for cable should be overbuilt when the opportunity arises, since marginal cost to add capacity is usually low once the project is underway but very expensive to get started.
7. Applications should be organized as separate presentation, application logic, data access, and database modules, with the capability to run the modules on separate servers to allow increased scalability, flexibility, and reuse.
8. Trends should be spotted, University and unit goals should be assessed, technologies should be evaluated, and training and experience should be obtained for those technologies and methods most likely to be required in the future.
9. There should be thresholds for components such that any component that is running at a capacity above its threshold for a significant period of time would be upgraded.

B.1.1. Deploy scalable infrastructure

Status: Adopted

Any portion of the enterprise-wide technical infrastructure should be scalable; that is, able to expand quickly and economically in capacity, capability, scope, availability, reliability and maintainability as needed.

Example 4. A university Human Resources (HR) department desires to computerize its scheduling of new employees for orientation, which is being done by a single individual "scheduler" using a paper planner. The system must allow scheduling orientation for new hires even before they are in the HR system. Instead of the scheduler simply using a computerized planner, the standard university enterprise approach and tools are used to create a web-enabled application with multiple components, including a web front-end that calls an application logic module to access a database and use the standard university authentication and authorization services. Initially, all application-specific components run on the individual's personal computer using a web browser with update access restricted to the individual. Subsequently, the HR department desires to speed up orientation by allowing any HR staff person to schedule new employees at the most convenient moment. The system is scaled quickly by installing a standard server, moving the modules and database to the server, and updating the university security database to allow update access by the HR staff. Some time later, HR decides to allow new hires to schedule their orientation without involving the HR staff. The service is able to scale quickly by adding memory, processor cards, and disk space to the server, and by allowing departments to give new hires access to the application.

Justification

- Enterprise solutions must be able to scale to support all of Emory, and possibly all its alumni and others with university relationships.
- Scalability, reliability, availability, and maintainability are needed to support increasing reuse.
- Infrastructure should be able to grow gracefully. It should make provision for affordable and rapid expansion to more extensive, higher capacity infrastructure.
- There is a trend of implementing self-service to allow affordably increasing the number of people served without a corresponding increase in support staff. Infrastructure must be prepared to scale to support this trend.

Implications

1. Local solutions that may later need to become enterprise wide should be built so they can scale.
2. Solutions that provide an enterprise-wide service but are only accessed by specialists must be prepared to scale to support enterprise-wide self-service.
3. Core infrastructure resources should be both reusable and expandable at the local level.
4. Infrastructure systems should be based on components that can handle a large amount of simultaneous shared use. In particular, their processing modules should have multi-user capability and their data access should implement record locking. Personal productivity tools such as Microsoft Excel, Access, and Word, while appropriate for use by individuals, are not intended for shared use, and do not scale well enough to base infrastructure systems on them.

B.1.2. Make it easy to integrate new technology

Status: Adopted

The architecture should promote easy integration of new IT devices, systems and solutions with the existing infrastructure.

Example 5. A university needed to install a non-standard server to run an application that it needed but could not run on one of its standard servers. Its architecture supported integration by means of an "integration hub," which provided a common place to connect using a standard interface. With this approach, the university only needed to interface the new server to the integration hub. Then the new server could interact with existing applications and they could interact with it using the standard interface.

Example 6. Instead of repeating the same application logic on a web server, on an interactive voice response server, and on a specialized desktop program, putting the application logic on a server that can be accessed via the web, the telephone, and the specialized program allows the application logic to be reused by a variety of means of access.

Justification

- Adding a new system or device increasingly requires that it be able to interoperate with other existing systems.
- Delay in getting a system running or a device supported is often due to problems integrating it with the existing infrastructure.
- The more complex it is to integrate a system or device, the longer integration takes, and the more difficult the whole integrated system is to maintain and change.
- This principle contributes to overall reduction in complexity, but does not necessarily imply it, since an infrastructure containing a large number of different types of systems integrated by means of a integration hub is complex, but integration with it is easy. This principle is also not a consequence of overall reduction in complexity, since an infrastructure of a single system without interfaces to its functions would be simple, but integration with it would be difficult.

Implications

1. Standard interfaces to services will be required, and might have to be created for many services.
2. Since not all vendors' products will be equally easy to integrate, the number of vendors and products that are suitable might decrease.
3. Achieving this goal will require "configuration discipline." The number of configurations in the environment will likely decrease.
4. Sacrificing performance or functionality in some instances might be necessary to make integration easier.
5. To keep the cost of evolution and change under control, the components must have affordable exit costs. In particular, to avoid the cost of lock-in, it must be possible to load information and business rules into and dump them from components as applicable. Thus closed parts of components might have to be isolated from use to prevent lock-in.
6. Reliance on "infrastructure subassemblies" supplied by vendors will likely increase.
7. Established services that have become impediments to integrating new technologies might require "wrappers" or interfaces until they can be replaced.

B.1.3. Deploy modular, loosely coupled components

Status: Adopted

IT solutions and infrastructure should be engineered with a bias toward using highly discrete, loosely coupled components.

"Highly discrete" means built of as many parts (called "components") with clearly defined usage, function, or purpose, as are distinctly useful. Components with functions should have as few functions as feasible, and a standard way to access all the functions. "Loosely coupled" means that interacting components affect each other as little as possible.

Justification

- Reducing the functional scope of the components reduces their complexity and increases opportunities for their reuse.
- Loose coupling reduces the complexity of a system of interacting components. It allows making internal changes to one component without affecting other components. It improves availability and scalability of the system, since problems with one component are less likely to impact other components.
- This principle leads to infrastructure in which components can be reliably changed more quickly than otherwise would be the case.
- Providing access through the interfaces to all the functionality of the components is necessary to allow hiding internal details.

Implications

1. Designs will need to be based on standard, replaceable components used as building blocks.
2. Components will need to hide their internal details from one another.
3. Logical boundaries between components will need to be established to prevent a component from depending on the internal details of another and to enforce use of the interfaces. The logical boundaries must be "firm;" that is, they must be enforced.
4. Design reviews will be required to ensure that the boundaries are kept intact.
5. Using this approach, there will be more components, and they will be smaller.
6. To provide strong isolation, the interfaces across separate logical boundaries will need to be message-based and use asynchronous messages to the extent possible. Using asynchronous messaging in which the sender of the message does not wait for a reply reduces dependencies between the components and lessens the likelihood of blocking and deadlocks.
7. Senders and receivers of messages will need to be able to verify each other's identity ("authentication"). Senders will need to be able to secure the content of their messages so that only the intended recipient can read it ("encryption"). Recipients will need to be able to verify that what they received is what was sent ("integrity").

B.1.4. Enable as much reuse as is feasible

Status: Adopted

All aspects of the architecture should be reusable and be reused to the extent feasible.

- This applies to products, tools, designs, systems, applications, methods, and data.

Example 7. Establishing a repository of general interest data with clearly documented meaning and verified accuracy avoids the cost of duplicated effort and enables all to reap the fruits of collecting and verifying the data just once. The repository can be made available on more than one server and in more than one location through replication.

Example 8. • Web page templates allow reuse of good designs to cut the time to get a web page in place. • Subroutines provide logic in a form that is reusable by multiple applications. • Saving answers to questions in a searchable database allows reusing them to avoid repeating the work to find the answer. • Placing identical system images on all computers in a lab reuses configurations to reduce setup time.

Justification

- Reusing something is faster than creating it anew.
- Reuse reduces cost by reducing duplication of effort.
- Reuse of a resource does not necessarily imply that the resource must be centralized. A resource can be distributed by replicating it.

Implications

1. General solutions are preferred over specific ones, because fewer changes are required for them to be usable in a new situation. However, general solutions are harder to implement than something more specific.
2. Achieving reuse can be difficult: • The nature of the anticipated reuse can be difficult to predict. • Resistance to reuse of others' components is fostered by the uncertainty of their quality, features, and use. • Implementers and administrators might not trust what others create compared to what they create themselves. • To achieve reuse, the components need to be trustworthy, useable, useful, documented, and easy to find.
3. As the amount of its reuse grows, a resource may need to provide higher availability and capacity (at additional cost).
4. The role of a "reuse librarian" will likely be needed at least on a part-time basis to organize a library of reusable components and their documentation and help people find the components they need.
5. IT work will need to be realigned to support reuse.
 - The reward system will need to reward reuse more than creating something new.
 - Separate roles may be needed to distinguish between those who build components and are rewarded for others reusing their components, and those who assemble solutions using components and are rewarded for the extent they use existing components.
 - To deliver reusable infrastructure services to business applications, infrastructure development will need to be a separate role and a core competency.
 - Design reviews will be required to establish trust in components built at Emory.

B.2. Reduce overall complexity

Status: Adopted

The enterprise IT infrastructure should be no more complex than it has to be to serve Emory's mission.

Example 9. • The complexity of a network environment of routers and switches from multiple vendors is unnecessary when one vendor can meet requirements. • The complexity of supporting many protocols is unnecessary when fewer protocols can provide needed access.

Example 10. Using the same configurations and versions of hardware and software across multiple servers reduces the complexity of managing them, even when more careful tuning would allow some of them to be configured with less capacity. This approach also saves money by taking advantage of the trend of decreasing hardware costs compared to the increasing cost of personnel to manage the servers. Over the life of the systems, the extra cost to manage the systems due to their not being identical is more than the extra cost to make them all identical.

Example 11. Interaction between systems is less complex when the systems all use a common interface, rather than each interacting pair having its own interface.

Example 12. To stay competitive, a university needed to provide to its faculty a specialized capability only available from a certain application that would not run on any of the university's standard systems. An architectural review of the proposed system uncovered a number of problems that were not apparent from the demo. The university acquired the application and the type of system needed to run it anyway, because the value of having the capability was great enough to justify the cost of dealing with the problems and learning and supporting an additional type of system. The university was better able to set expectations for how quickly it could get the system working and better able to address the problems as a result of getting early knowledge of the nature of the system.

Justification

- Reducing the complexity of an environment makes it easier to manage, test, and reliably change, which increases the speed of response.
- Less complexity makes the effect of a change easier to understand, which leads to faster and more reliable changes.

Implications

1. Additional complexity should only be allowed when it adds enough value in the service of Emory's mission to overcome the additional cost.
2. Reducing complexity typically involves reducing the number of permutations and combinations of infrastructure parts.
3. Reducing complexity will require the discipline to establish and enforce use of reusable building blocks and standards for interfaces and products.
4. Reducing complexity will require cooperative action across Emory.
5. The number of vendors, products and standards used at Emory might decrease.
6. Achieving this goal will require "configuration discipline." The number of configurations in the Emory environment will likely decrease.

B.3. Standardize judiciously

Status: Adopted

Standards should be seen as a means to support university priorities, the requirements and principles of the IT architecture, or the common good at Emory. Standards that limit individual choices of personal technologies should require justification on the above basis.

Example 13. A university's schools and divisions individually began to investigate document management and workflow solutions to reduce paper handling and automate their own processes. The enterprise IT architecture committee realized that interoperable solutions could be leveraged not just within the units but across them as well. Working together, the units agreed to choose the same product, since at that time standards for interoperation were not reliably and robustly implemented. Although the choice was not the favorite product of some units, they all cooperated for the greater good of the university.

Justification

- Standardization must not be allowed to be unduly constraining. In a research university IT needs differ widely. Even more important, much of the strength of the institution comes from the freedom of its individual faculty to work in innovative, divergent, and unconstrained ways.
- Standardization is a key architectural strategy.
 - It enables faster response and lowers support costs by reducing the number of permutations and combinations of hardware, software, and configurations that must be maintained and supported.
 - It promotes reuse and interoperability.
 - It provides a critical mass of common interest where people can help each other.
 - It enables cost-effective internal stocking of parts.
 - It enables purchasing economies.
 - It expedites decision-making, since a standard configuration is likely to be suitable.
 - It simplifies training and transfer of skills.
- Changing a product standard is easier the more standardization is in place.

Implications

1. Effective university processes will be needed to determine what things should be standardized, and what things must be left to individual choice.
2. Effective governance will be needed to enforce standards.

B.4. Use process-event driven systems

Status: Adopted

Enterprise IT systems should be process-event driven rather than batch-oriented. After acting on an event, any results for use by other systems should be made available immediately as an event.

"Process-event driven" means that events relevant to what the system does initiate processing within the system. "Batch-oriented" means that transactions are queued at the sending system and sent as a batch at the discretion of the sending system.

Example 14. A university's online system wrote transactions to a file for later batch processing, at which time the online system had to be shut down to release the transaction file. To process transactions more quickly required this to be done more often. If the online system immediately made the transactions available in a queue for access by the batch programs at any time, then the throughput of the system could be increased without affecting the online system by simply running the batch programs more often.

Example 15. When someone joins or leaves Emory, the systems that record people's status should be immediately notified. Those systems in turn should immediately process this information and make the results available for other systems that need to know, such as systems that control access to Emory resources.

Example 16. When scheduling a transaction for the future, the first transaction is the request to schedule. It goes immediately to the scheduling system and is logically queued. Arrival of the future time is an event that causes the scheduling system to go through its queue and send transactions that are ready to go.

Example 17. For an approval process in which a request goes from one approver to another, the transaction should be sent to the next approver in the chain as soon as the previous approver finishes with it.

Corollary

- a. The same principle applies to components used to build systems, with the word "systems" replaced by the word "components" and the word "events" referring to events of the system.

Justification

- Enterprise IT systems support processes. A process is a series of events. Examples of events are receipt of a student application, acceptance of a student, student enrollment, student matriculation, posting of grades, arrival of a date or time, faculty and staff joining or leaving Emory, and changes in address, department, and status.
- Having the system respond to events, rather than assuming a certain series of events and asking for needed information ("procedure driven"), allows the system to adapt to a change in the series of events by just adding support for new events.
- Making the results available immediately allows a receiving system to change its rate of processing independently of the sender. Such an approach makes speeding up the system easier and reduces complexity by reducing dependencies.
- Systems whose components adhere to the principle can be made to comply with the principle more easily than when a component of the system violates the principle.

Implications

1. Infrastructure will use asynchronous event-oriented systems rather than batch-oriented systems. Asynchronous logic will replace batch logic, and asynchronous processing will replace batch processing.
2. Implementation of process-event driven systems requires systemic thinking, because event-based processing generally crosses traditional system boundaries.
3. The event-driven approach makes easier the implementation of the "push" model of information delivery in which information is pushed to the receiver rather than waiting for the receiver to check for it ("pull" model).
4. IT systems that receive information from other systems should be able to respond to such information as soon as it becomes available.

B.5. Provide a common security layer

Status: Adopted

The infrastructure should present a consistent, uniform and robust security layer across all infrastructure components regardless of their physical location.

Justification

- A common security layer:
 - Enables reuse (B.1.4) and supports authorized access to services and resources no matter where located (Arch 13).
 - Provides easier integration with the infrastructure (Arch 3).
 - Provides easier authorization of access (Arch 5).
 - Supports interoperability, communication and exchange of information (Arch 12).
- Emory needs a common set of security tools, policies, security levels, and interoperable security infrastructure that includes firewalls and monitoring (intrusion, attack, vulnerabilities).

Implications

1. Common security services will be needed.
2. Initially it might be necessary to use multiple kinds of security systems to implement this capability.
3. The ability to follow this principle will depend on the existence and quality of an enterprise directory service.
4. A way is needed to handle visiting faculty and others who have short-term relationships with Emory.

B.6. Facilitate access to IT resources.

Status: Adopted

Provide a way to find IT resources that indicates information about the contents of the resources and how to request access to them.

IT resources include data and information stored in systems.

Justification

- Making known the existence of a resource increases the possibilities for its reuse. It also supports architecture requirements #6 and #11.
- Data and Information in Emory systems are University assets whose value increases when they are used. It is a long-standing Emory value that information be shared subject to privacy and confidentiality requirements. The value of making a resource more accessible is often not recognized by the custodian of the resource.
- Information sharing allows for faster and more effective decision-making, and contributes to the discovery of new relationships. Sharing also results in the data being more widely reviewed, which increases the accuracy of the data.
- Sharing information with external partners increases the effectiveness of the partner relationships.
- When a change requires new access to a resource, having the resource already accessible eliminates delay to get access and thus increases responsiveness.

Implications

1. IT resources will need to be identified, documented, and classified. The information will need to be stored in a directory into which identifiable individuals and systems can enter, maintain and search the information as authorized.
2. Management of Emory-wide and enterprise resources will need to be unified.
3. Authoritative sources for enterprise data and information will need to be identified and documented.
4. Data repositories need to be developed to facilitate availability of information for decision making.
5. The Emory policy on information and resource access will need to be better known by the Emory community.
6. Units might be called upon to provide wider access to some resources that are currently under their control.
7. Some data will need to be restructured for easy access and management.

B.7. Document flows of information

Status: Adopted

The flow of information into, out of, and between components of the architecture should be documented and made available for access via the Emory Intranet.

Justification

- Documenting flows of information will:
 - Facilitate understanding of information usage and relationships across functions and systems.
 - Allow Emory to make better predictions of the impact of changes in one area on information used by another.
 - Allow Emory to see the similarities in need for information management across functions.
- Information delivery is the vehicle through which "IT's value" will be delivered.

Implications

1. A preliminary high-level information model must be developed to identify and prioritize architecture-targeted information entities (e.g., student).
2. All major projects must contribute to or take direction from the evolving information architecture.
3. A centrally managed effort is required to control the evolution of an information architecture that is comprehensive.
4. The scope of this principle is so large that an evolutionary approach to its implementation is required.
5. The documentation effort can build on work from Phase 1 of the architecture effort. That phase provides a head start by identifying high-level business information requirements aligned with Emory's priorities and strategies.

B.8. Provide access controls for IT resources.

Status: Adopted

The IT architecture should provide access controls that can make public the IT resources that should be public, and make private the IT resources that should be private. It should further classify private resources as needed to take into account differing security needs.

IT resources include data and information stored in IT systems.

Example 18. A university further classified its private resources as Restricted, Confidential, or Proprietary according to the impact to Emory if the resource is compromised, lost, stolen or damaged.

Justification

- Data and Information are valuable assets whose value increases when they are used. The value of making a resource more accessible is often not recognized by the custodian of the resource. It is a long-standing Emory value that information be shared subject to privacy and confidentiality requirements. Consistent with this principle, Emory University and Emory Healthcare seek to provide appropriate access to their resources among their faculty, physicians, staff, and students. Access to those resources, however, carries with it the responsibility to protect privacy, confidentiality and integrity.
- Much Health Science research depends on the use of aggregated patient data. Healthcare information and data that can be traced to an individual patient are confidential. They must be kept private and only shared to the extent necessary to provide patient care.

Implications

1. IT resources will need to be identified, documented, and classified to determine security controls.
2. Access to some resources will be given based on an individual's role. Access to other resources will be at the discretion of an individual. The infrastructure will need to provide an easy and flexible way to implement both.
3. Management of Emory-wide and enterprise resources will need to be unified.
4. Authoritative sources for enterprise data and information will need to be identified and documented.
5. Data repositories need to be developed to facilitate availability of information according to policy for decision making.
6. Some data will need to be restructured for easy access and management.
7. The Emory policies on information and resource access need to be better known by the Emory community.
8. Units and individuals may be called upon to provide information about Emory resources that are currently under their control.

C.1. Costing and pricing should promote desirable behavior

Status: Adopted

IT costing and pricing should promote the architecture's goals, principles and practices. Costing and pricing should also encourage and facilitate action for the common good, university-level management of the common infrastructure, rational and informed decisions, manageable and affordable infrastructure expansion, computation of total costs, and equitable cost allocation.

Corollaries

- a. IT projects should be evaluated and alternatives compared using total cost and benefit over the full life cycle, including cost of training, support, maintenance, entry and exit, payback or return on investment, and adjustments for risk.
- b. All infrastructure services should define and track the following metrics over time: performance, health, available capacity, who is using the service and how much.

Example 19. A university desired to make databases available to students and faculty in its library. One alternative (the "workgroup solution") was to install a workgroup server to be accessed over a local area network from desktop computers located in the library. Those who could run the client could also access the server over the university network. However, the client required a more powerful desktop computer than some people had, and would run only under one of the types of systems popular on campus. An alternative with a considerably higher acquisition cost was to put the databases on a larger computer that could make access available via the web. The committee understood that the total cost included more than the acquisition cost. The workgroup solution introduced technology that was more expensive to manage, because its drives were not as reliable as those used by the web solution. The workgroup solution also had hidden costs associated with an inability to scale, because its search client had to transfer across the network all the records that it checked. In addition, there was a cost to deploy, update, and support clients distributed to people's personal computers. Although the workgroup solution could be replaced later by the web solution, there were costs to change the technology and exit from the workgroup approach. A difficult to quantify hidden cost was the extra time spent by those who could not run the client, such as protesting their lack of remote access and visiting the library to gain access. Ultimately, the university chose the web solution rather than the workgroup solution.

Justification

- Solutions to IT requirements should be cost effective and competitive.
- Total cost over lifetime must be used to correctly understand the cost of alternatives in support of Emory's cost control and efficiency strategy.
- The indicated metrics provide relevant data for rational management and informed decisions.
- The metrics are needed to maintain capacity to respond quickly and to manage the life cycle.

Implications

1. IT project management and costing expertise will need to be acquired.
2. A separate project management office responsibility may eventually be needed to ensure that infrastructure costs are computed according to standard.
3. Every addition of capacity or function needs to include the costs to install and maintain it.
4. To provide ongoing renewal, most infrastructure costs should be treated as operating costs, not capital costs. Exception: Pathway and cabling.

C.2. Develop staff competencies in areas of strategic importance

Status: Adopted

Emory's internal core IT competencies must be more fully developed in areas that contribute to Emory's distinctiveness, as well as other areas of strategic importance to Emory.

Justification

- Today's IT skills are very expensive.
- Investment in IT skills must support development of areas of strategic importance to Emory.
- Emory's distinctiveness is of strategic importance.
- Emory must be able to change its processes and IT systems as quickly as needed to maintain its distinctiveness and support its strategies.
- Those processes and IT systems require support from internal IT resources with a vested interest.

Implications

1. Areas of strategic importance to Emory must be identified.
2. Priority alignment will help define the areas that are of strategic importance to Emory; they need to be supported by internal resources with vested interests.

C.3. Consider outsourcing in the context of risk to Emory's future

Status: Adopted

Outsourcing an IT service must not put Emory's future at unacceptable risk. Whether a service is outsourced or not, all IT service planning and management must be done internally. In particular, Emory's enterprise architecture development must be done internally.

Corollaries

- a. An outsourced IT service must conform to Emory's enterprise IT architecture.
- b. In evaluating risk, consider the effect of a non-performing service provider on Emory's goals, initiatives, priorities, programmatic activities, and distinctiveness, taking into account the ease of replacing the non-performing provider.
- c. Avoid outsourcing a service that must rapidly adapt to maintain Emory's distinctiveness or to address changing needs critical to Emory's future.

Justification

- Outsourcing a service can put Emory's future at unacceptable risk due to the outsourcer's failure to perform. Outsourcing typically involves an agreement that specifies a level of performance within a standardized scenario for service delivery. Thus services that are less standardized in their delivery are more risky. An outsourcer's performance will depend on incentives and motivations created by the contract, not on what is best for Emory. The outsourcer may be unable or unwilling to respond quickly enough.
- Outsourcing is less risky when there are multiple choices of service providers who can be easily and quickly replaced. Interchanging providers becomes easier as the service matures and the nature of what is to be provided becomes more standardized. A service is most mature in this sense when the main differentiator is price. The service is then called a "commodity."
- Outsourced services that do not conform to Emory's enterprise architecture create an unacceptable risk to the future of Emory's IT infrastructure and thus to the Emory goals, initiatives, priorities, and programmatic activities that depend on it.
- Any outsourced service will still require campus-based planning and management. Responsibility for a service cannot be outsourced.

Implications

3. Commodity services are candidates to be considered for outsourcing.
4. Off the shelf components within standard designs can be used to obtain some benefits similar to outsourcing.
5. Outsourcing arrangements will need careful management to ensure that the needs of campus programs drive the services. Even when outsourcing all operations, architectural direction must be set by resources internal to Emory that are driven by the needs of campus programs.
6. Architecture becomes a function of IT, just like network support, application development, and database management. Architecture can have external assistance, but ultimately must have an internal leader.

C.4. Use industry standard solutions when feasible

Status: Adopted

Enterprise information technology selection and infrastructure decisions should be based upon industry proven and supported components, methods, standards and tools. Custom solutions should only be used when feasible industry standard alternatives cannot be affordably obtained, or when outside support for such a solution does not provide needed changes fast enough. Customization of an industry standard product or system is a custom solution.

Example 20. Sometimes customization is required. A university sought to attain distinctiveness in student recruitment. It realized that speed of response to applicants was crucial to obtaining the students that it wanted. Using the same process and applications as the current top performers allowed it to be only as good and as fast as they were, not faster. To exceed them, the university created custom modifications and additions to the process and its supporting IT systems that allowed it to obtain, enter, and process applications electronically. It created a custom knowledge-based filter that allowed its own admissions specialists to automatically select applicants for further examination based on the specialists' own criteria for the applicants most likely to interest them. As a result, the admissions department was able to respond within one workday. The extra cost to develop, maintain, and enhance the customizations was part of the cost to be number one.

Justification

- Availability of industry standard solutions is increasing and will continue to increase. Others with which Emory competes will take advantage of these solutions.
- The Internet community will use and develop open standards, providing low cost and high innovation.
- Reusing an existing solution is typically faster and less expensive than building one or customizing an existing one.
- Customizations to an existing solution typically delay upgrading to a new version due to the need to reimplement the customizations.
- A custom solution is appropriate to provide capability that does not already exist or where quick adaptation to rapidly changing needs is required. External suppliers typically cannot respond fast enough.
- Using industry standard solutions avoids a dependence on the skills to build or customize and maintain the custom solution. Such skills are in short supply indefinitely, and might not be available later when needed. This principle also helps to avoid high exit costs due to lock-in.

Implications

1. Emory should identify the areas where it will need to provide custom solutions and invest in the skills, tools, and technology for custom development, modification or integration in those areas. The general nature of those areas is increasingly becoming information delivery and decision support, rather than operations and transactions.
2. Use commodity goods when possible.
3. Use open standards when functional needs can be met.
4. Take into account total lifetime cost, payback or return on investment, and exit cost.
5. Use vendor-supplied solutions to provide accountability and a contractual obligation.
6. Wait for industry standard solutions when possible. Short-term solutions should minimize capital expense and exit costs and maximize operating budget to allow redeployment later.

Appendix 2. Technical Architecture Requirements

- Arch 1: Facilitate change as quickly as needed in academic and administrative processes and the applications that enable and support them.
- Arch 2: Enable work flows to be quickly and easily implemented and integrated with already integrated systems.
- Arch 3: Provide a standard facility to allow a system to be quickly and easily integrated with other already-integrated systems to allow exchange of information with them as quickly as the information becomes available.
- Arch 4: Provide standard mechanisms that allow access to services to be extended quickly and easily as needed to allow access using new types of devices.
- Arch 5: Provide the capability for custodians of data items, services and resources to easily authorize (and change) access to services and people inside and outside Emory, individually or by membership in a group.
- Arch 6: Provide a capability to quickly document the existence of information, its attributes, meaning, use, and how to access it.
- Arch 7: Provide strong security mechanisms to help protect systems, resources, information, data, and communications from disruption, corruption, and loss, as well as unauthorized access, use or disclosure.
- Arch 8: Support storage, access and conversion of multiple types of data in multiple media and formats.
Types of data include numbers, text fields, textual documents, drawings, images, binary files, documents, audio, video, hypertext, etc.
- Arch 9: Provide access from Emory's campus network to the commodity Internet and Internet2 using world standards for interoperation, communication and data access.
Although the more general requirement is for connectivity with networks readily accessible by the public worldwide, and for high-performance access to other research institutions, for the planning period those networks are the commodity Internet and Internet2, respectively.
- Arch 10: Enable rapid increases in the volume of information that will be processed by university-wide systems.
- Arch 11: Provide a facility for managing the storage and dissemination of increasing amounts of data at Emory that can handle rapid increases over the next 5 years.
- Arch 12: Support broad-based standards that will increase interoperability, communication and exchange of information.
- Arch 13: Provide the capability for services and resources to be accessible through the network by any authorized service, resource or person at anytime from anyplace.

Arch 14: Provide a high-performance, fully transparent, flexible and reliable campus network that allows people, systems, and services to communicate and exchange information.

"High performance" implies high bandwidth, low delay, and small variation in delay. A "transparent" network does not require knowledge of its internals to use it. A "flexible" network can be quickly and easily configured or upgraded as needed. Communication includes one-to-one, one-to-many, and many-to-many. "Information" can include numbers, text fields, textual documents, drawings, images, binary files, documents, audio, video, hypertext, etc.

Appendix 3. Glossary

asynchronous messaging	A message can be sent at any time without waiting for the receiver to indicate that it is ready.
authoritative source	The "official" process or system that maintains a piece of data, and the place where data validity issues and creation and update rules for that piece of data is addressed.
commodity	A product or service that is so standardized that the main differentiator between suppliers of it is price.
component	A part that has a clearly defined usage, function, or purpose and that can be accessed in a defined way without knowledge of its internals. A component can be a hardware part, software, a database table, a software routine, a code module, a server, a database, a system, etc.
data	Factual material such as numbers, text, sounds, or images suitable for communication, interpretation or processing. It is raw and without meaning. Cf. Information.
Emory	In the context of organizational scope, it is all sites of the Emory enterprise.
enterprise, Emory	The entire legal entity to which the University and Emory Healthcare belong.
highly discrete	Built of as many parts (called "components") with clearly defined usage, function, or purpose, as are distinctly useful.
information	Information is data with meaning understood in context.
infrastructure, IT	The foundation on which IT systems are run. The basic stuff you need to have in place before you can start to build IT solutions. It provides storage, bandwidth and processing power. It consists of the components of a computing setup: the wiring, routers, switches, operating systems, middleware, mainframes, servers and sometimes desktop machines.
intranet	Use of Internet technologies to deliver IT services internally to an organization. It is typically an internal internet that is separated from the global Internet by at least one firewall, and may employ additional Internet technologies to increase security.
IT	Information Technology
loosely coupled components	Components depend as little as possible on knowledge of the state or on the performance of other components in order to use them.
open	Unencumbered specifications are freely available, independent branding and certification processes exist, and multiple implementations of a single product may be created.
resource	Anything the university has for the purpose of pursuing its mission. An IT resource is any information technology that is a resource.
steward	Of a resource: Responsible to the owner of the resource and sets policy for the resource. Typically has an active, working knowledge and understanding of a resource and its needs. Often a senior university official with planning and policy level responsibility for a resource created or maintained within a functional area or business process of the university. May empower a custodian to manage the resource.
University, the	In the context of organizational scope, it is the enrolled students and the employed faculty and staff of Emory University no matter where they are located.