E-GRANTS

IT ARCHITECTURE OVERVIEW

The E-Grants IT architecture will be compatible and compliant with the standards documented by the E-Gov Architecture initiative and with overall Federal IT standards and architecture. E-Grants will support the Federal Enterprise Architecture Framework (FEAF) as the framework for the system architecture. The use of a standards-based logical architecture also serves to facilitate integration with other agencies systems and architectures.

The E-Grants storefront will provide a common set of business processes within a robust framework that ensures system security. Grantee users will access the E-Grants storefront to perform pre-award and eventually post-award grants business processes through various interfaces. The E-Grants storefront will eventually interface to agency legacy systems to interface documents such as grant applications and project status reports submitted by grantee users. All components of the E-Grants storefront will utilize commercial off-the-shelf (COTS) products to the maximum extent possible and be developed in accordance with industry and Federal best practices for e-business systems.

An Integration Toolkit will be developed to assist agencies that need to integrate their existing grants systems and business processes with the E-Grants storefront. The toolkit will include recommendations of COTS products and other components helpful to agencies in this regard. Additionally, because Federal agencies will have different levels of technical capability, the Integration Toolkit may support alternate interfaces to accommodate the different capabilities of Federal agencies. XML is proposed as the data representation format and the primary strategy will be to simplify the interfaces and reduce the burden on agencies to the maximum extent possible.

Scalability

In order to plan for future horizontal and vertical growth of the E-Grants system, the system will be segmented into logical and physical modules. To provide secured access with high availability, a duplexed enterprise server environment will be required. The enterprise server will store the data and interface with the Web application server. In turn, the Web application server will interface with the Policy, Directory, and Messaging servers. Storage area networks (SANs) will be used to store and backup the data. Load balancing will be used to manage processing. As the number of transactions increase, the server CPU, disk and chassis can be expanded to handle the load. When fully operational the initial pilot routers will be replaced with high speed interconnect devices and larger transmission capacity. An enterprise database will be used that has the capacity to expand either in a centralized environment or in a distributed environment.

It is estimated that the system must be able to store a minimum of three versions of 500,000 applications, which contain about 100 data elements of about 100 bytes per element (text and image attachments will require more disk). Hence, a minimum of 15 Terabytes will be required to handle the current application volume plus the possible increase of application volume due to enhanced and simplified electronic application capabilities.

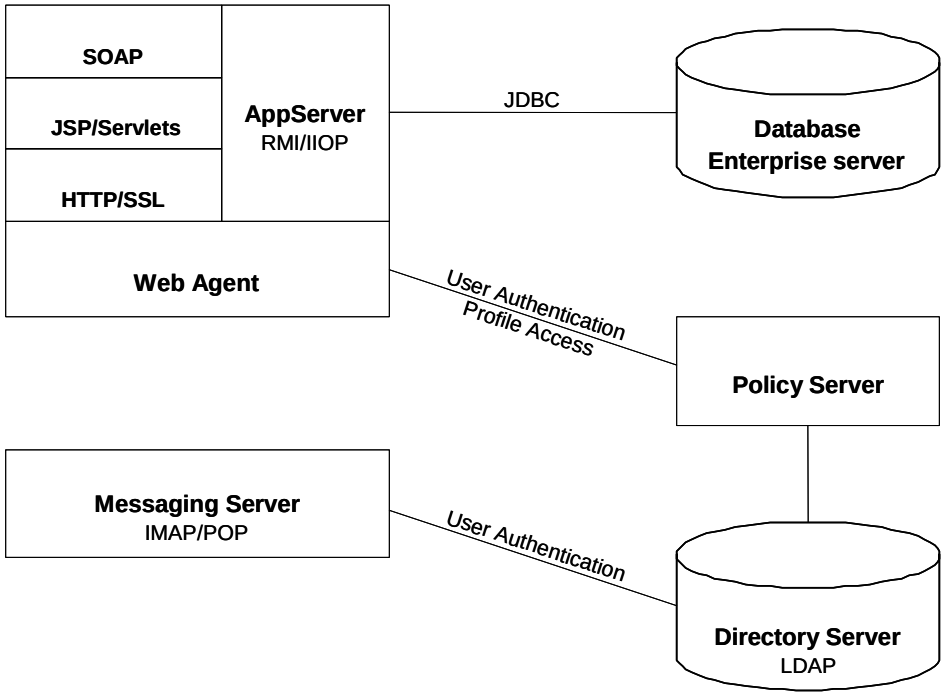

Figure I-1: Server Architecture

Web Application Architectures

The following architectures were considered as alternatives In addition, the following objectives were addressed:

- Scalability & performance
- Ease of development and maintenance
- Future extensibility

Server Architecture	Pros	Cons
JSP/Servlets w/ Business Objects	Platform independent, strong support	No EJB framework or services
J2EE w/ Session Beans + DOL	EJB framework (security, scalability)	Uses custom data object layer
J2EE w/ Session + Entity Beans	Simplified data access	Entity bean performance issues
ASP, COM, ADO	Microsoft supported technologies	Platform and vendor dependent

Table I-1 – Server Architecture Options

The first three options consist of different combinations of technologies included in the Java 2 Enterprise Edition (J2EE) framework. The J2EE Blueprints provide official guidance for developing enterprise web applications, including various architectures and guidelines based on best practices and recommended use of the latest J2EE technologies. The final option consists of the Microsoft supported technologies that are the primary competitor to Sun's J2EE framework.

The second option (i.e., J2EE with Session Beans and Data Object Layer) was considered the best fit for the E-Grants Storefront architecture. This hybrid option combines the strengths of the first option with the increased flexibility and scalability

provided by the EJB framework. It avoids, however, a number of performance issues by using a custom data object layer and limiting the use of EJB Entity Beans. The following is recommended:

Use the J2EE framework and architecture as depicted below.

Use loosely coupled layers to simplify development and maintenance by reducing complexity.

Use a modular combination of JavaBeans (Model), JSP (View), and Servlets (Controller) to implement the presentation layer and upper business logic layer.

Use a combination of EJB Session Beans and ordinary Java classes used to implement the business services layer. The use of EJB provides a number of major benefits including integrated security model, transaction support, and scalability.

A data object layer (DOL) will be used to encapsulate the database interface. Minimal use of EJB Entity Beans will avoid many of the issues and performance problems associated with their use. Investigate emerging standards such as Java Data Objects (JDO) to help encapsulate and simplify database access in the future.

Figure I-2 – Recommended J2EE Architecture outlines the recommended architecture for E-Grants Storefront web applications. Layers are created to help reduce the coupling between code and ease maintenance. A controller (written as a Java Servlet) is used to encapsulate global control logic such as access control and error handling. Business logic is implemented using a combination of EJB Session Beans and Java classes. All database access is encapsulated primarily as a layer of Java classes that use JDBC to interact with the database server. EJB Entity Beans will be used in limited cases where performance issues can be minimized or eliminated. Finally, JSP pages and JavaBeans will be the primary mechanism to code the HTML user interface and interact with the business logic layer.

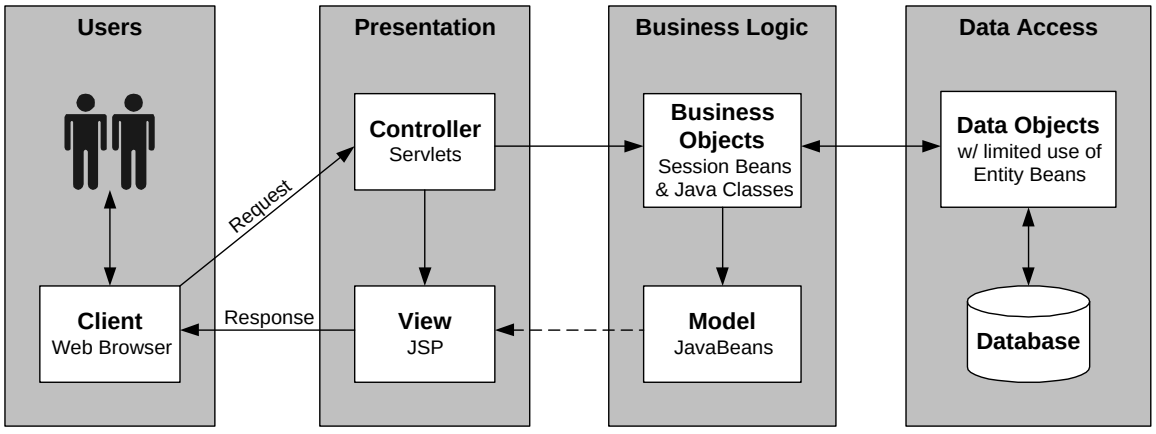

Figure I-2 – Recommended J2EE Architecture

Recommended Integration Strategy

Based on the evaluation of communication protocols and the E-Grants Storefront requirements, the following integration strategy is recommended. The precise integration architecture and guidelines will be further defined in the Federal Enterprise Architecture guidelines documents.

Use SOAP as the protocol for synchronous communication with agencies.

Use IMAP/POP to asynchronously deliver files (grant applications and reports) to smaller agencies.

Avoid proprietary products and protocols that require agency investment or buy-in.

Simplify agency integration by providing two tiers of services.

Provide an application status proxy service for smaller agencies that cannot support real-time status queries.

Focus on standards now and plan migration to COTS products as the market matures. The integration product market is still emerging and products are quite expensive

Providing two tiers of services as outlined in the table below allows easier integration with smaller agencies that cannot support or afford a sophisticated grants management system and real-time application status module. The approach to support these agencies is to allow easy retrieval of grant applications and reports via low-cost COTS software (e.g., e-mail client) and to simplify application status checking by providing a "proxy server" which smaller agencies can update on a periodic basis. A SOAP-based interface can be defined so that agencies can develop automated tools to upload their internal status data to the proxy server.

Process	Small Agencies	Large Agencies
Grant Applications	Asynchronous retrieval via IMAP/POP	Retrieval via SOAP (or IMAP)
Status Checking	Send periodic updates via SOAP	Synchronous query via SOAP
Reporting	Asynchronous retrieval via IMAP/POP	Retrieval via SOAP (or IMAP)

Table I- 2 – Two-Tier Integration Approach

IT Modernization and Architecture

Technical Standards

Standards will adhere to the following policy mandates where appropriate:

- Government Information Security Reform Act
- Critical Infrastructure Protection (Presidential Decision Directive 63)
- OMB Circular A-130
- OMB Circular A-127 Section 508, Rehabilitation Act of 1973

Software development will adhere to and be compliant with:

- ISO 9000
- FIPS
- JFMIP

Emerging interoperability standards such as Electronic Data Interchange (EDI), Mark-up Language (XML), and Universal Discovery Description and Integration (UDDI)

Key IT Standards

Standard	Compliance / Usage
Data Communication Services	
Extensible Markup Language (XML)	Standard format for data representation and exchange.
Simple Object Access Protocol (SOAP)	Web services integration with external systems (e.g., Federal agencies).
Internet Inter-ORB Protocol (IIOP)	Integration with back-end systems and services.

ANSI X12 EDI	Integration with large customers and pre-existing systems.
Internet Message Access Protocol and Post Office Protocol (IMAP / POP)	E-Mail delivery of XML grant applications and reports to agencies.
Multipurpose Internet Mail Extensions (MIME)	Standard encoding for document delivery via e-mail.
Data Management Services	
Lightweight Directory Access Protocol (LDAP)	Retrieve and update data stored in hierarchical directories.
SQL 92	Retrieve and update data stored in relational databases.
Java Database Connectivity (JDBC)	Programming API for database access from Java.
ODBC	Programming API for database access.
Programming Languages	
Java 2 Enterprise Edition (J2EE)	Platform independent and widely available framework for the development of web-based and e-business systems.
User Interface Services	
Hypertext Markup Language (HTML)	Thin-client (i.e. web browser) user interfaces.
Dynamic HTML (DHTML)	
Security Services	
Transport Layer Security (SSL/TLS)	Data encryption and security over internet protocols (e.g., HTTP, SOAP, IMAP)
X.509 Digital Signature Certificate	
Virtual Private Network (VPN)	Data encryption and security between back-end systems and key partners.
Network Services	
TCP/IP (IETF STD 5 & 7)	
Simple Network Management Protocol (SNMP)	

Key IT Assumptions

Assumption	Strategy
E-Grants storefront must be able to integrate with Federal agencies at various levels of technical capability.	Support commonly available standards and tools (e.g., XML, IMAP) and provide alternate interfaces where necessary.
E-Grants storefront must serve the needs of a diverse grantee community (e.g., state & local governments, universities, small-businesses).	Provide multiple grantee interfaces to support the needs and capabilities of different grantees. This will include a web-based person-to-system interface as well as one or more system-to-system interfaces
The system must be adaptable, scalable, and maintainable.	Standards-based COTS and open-source products will be used to the maximum extent possible.
Security and privacy issues are critical to the success of the E-Grants storefront.	Security will be designed from the start and will be continue to be addressed throughout the entire system lifecycle.

Key IT Risks

Risk	Strategy
Implementations of large, complex, multi-agency initiatives have significant failure rates.	Use a modular design and keep the interfaces as simple as possible to reduce the burden on agencies and other partners.
Federal agencies may not be able to adapt to significant process and technology change without appropriate support.	The E-Grants storefront Integration Toolkit will be designed to assist agencies in integrating their systems and business processes with the E-Grants storefront.
Basing production systems on prototypes intended for evaluation and proof-of-concept often results in failure.	The E-Grants initiative will use prototypes and pilots as mechanisms to investigate techniques and implementation approaches, but will rely on proven system development strategies for any E-Grants-based development and production deployment.

Security and Privacy

Security and privacy are of fundamental importance to the E-Grants. For this analysis, security is divided into two main categories. First, server security deals with protecting the E-Grants servers from unauthorized access or malicious attacks. Second, transaction security deals with protecting the privacy and integrity of data transmitted between the grantee community and the E-Grants storefront, and between the E-Grants systems and Federal agencies. Both aspects must be addressed jointly to develop an effective and unified security strategy.

Furthermore, the E-Grants project understands that individual security practices and controls cannot be planned in isolation. A coherent, comprehensive, and integrated security policy and framework is required to develop effective security controls. On these grounds, the following assumptions and guidelines will be taken into account while developing the E-Grants systems:

Security controls should be consistent with the medium to large procurement scenario from the CIO Council recommendations document.

Transaction security should be based on industry supported security standards and protocols (e.g., SSL/TLS, S/MIME).

The E-Grants systems should be prepared to use appropriate security mechanisms including user id and passwords, and public-key infrastructure (PKI) for authentication and non-repudiation, where appropriate.

A prudent, risk-based approach will be used to define security policy and controls.

Privacy issues will be carefully considered related to professional profile information.

Multiple layers of security controls are necessary to provide adequate security for the system as a whole. Security should be imbedded at the database layer whenever possible.

While the E-Grants systems may use initially an enhanced system of usernames and passwords as the initial user authentication mechanism, it is planned that the E-Grants initiative will eventually leverage other Federal projects to provide public-key infrastructure (PKI) authentication mechanisms. Particular care has been made in the design of the E-Grants systems to provide a clear upgrade path to PKI. In addition, the

E-Grants project team will work closely with the E-Authentication team to ensure that E-Grants solutions are compatible and compliant with overall E-Gov requirements and capabilities.

Additional Security Guidelines / Assumptions

Web-Based User Access Control

- Encrypted connection (HTTP over SSL/TLS) required for all non-public functions.
- Enhanced username/password authentication mechanism.
- Use COTS products to provide centralized security policy and access control.
- Prepare to support certificate based PKI authentication.

E-Grants Server Security

- Use a secure hosting environment.
- Conduct frequent security audits.
- Solid perimeter defense (e.g., Firewalls).
- Intrusion detection
- Strong server access control (e.g., SSH).
- Strong database security controls.
- Use VPN for back-end server communication.

Integration with Federal Agencies

- Use 128-bit SSL/TLS with username/password login or mutual certificate-based authentication to ensure delivery to the intended agency.
- Investigate S/MIME for e-mail delivery of documents.
- Use firewall and TCP wrappers to limit connections to known partners.
- Use a VPN if necessary to provide security in special cases.

XML / EDI Application Submission

- Encrypted connection (SSL/TLS) required for submission of applications via HTTP or other protocols.
- Investigate S/MIME for e-mail delivery of grant applications and reports.
- Use firewall or TCP wrappers to limit connections to known partners.