One Washington
Integration Strategy

Table of Contents

1.0 Executive Summary

The Integration Strategy defines the future-state interface approach between the One Washington ERP (Finance, Procurement, HR/Payroll, and Budget applications) and other systems with which the ERP will interface. Interfacing systems may include other state enterprise systems, systems managed by the various state agencies, and external systems.

While the Integration Strategy defines the future-state, it also serves as the foundation for the development of the Integration Implementation Plan which will detail the expectations for agencies, roles and responsibilities, implementation methodologies and expectations of effort between the One Washington program and other state agencies.

2.0 Background and Introduction

This section outlines the high-level integration strategy and guiding principles for interfaces between the One Washington ERP (Finance, Procurement, HR/Payroll, and Budget) and other systems with which the ERP will interface. To facilitate the development of the Integration Implementation Plan, this strategy will discuss the following topics:

- High-level approach and guiding principles
- Integration methodologies
- Data conversion approach

One Washington's integration strategy will design an open architecture approach that facilitates data exchange and application interoperability with multiple legacy and external systems while supporting multiple technologies. Current state systems are not well integrated with one another, rely on aging technologies, and require extensive effort by staff to maintain and function.

The final objective of the Integration Strategy is to support the guiding principle of providing a unified system of record for Finance, Procurement, Budget, and HR/Payroll. A unified system of record is a term that describes an information storage system that is the authoritative data source for a given set of data. ERP solutions provide a unified system of record and provide the following benefits:

- Accurate and timely data for decision makers
- Reduced risk of major system failure
- More staff time devoted to delivering the mission rather than maintaining systems
- Critical capabilities maintained without having to own all the technology
- Process efficiencies as routine tasks are automated

The Integration Strategy was developed by the One Washington program based on a review of existing documentation and discussions with technical groups and agency staff which included:

- Applicable Policies – such as "Securing Information Technology Assets" – Policy: 141
- Current Capabilities – such as Informatica and Business Objects
- Current state of infrastructure

3.0 Supporting Activities

To finalize the Integration Implementation Plan, the One Washington program will work with agencies to identify and document current interfaces, interfacing systems, and requirements. This information will be used to determine the level of effort and remediation required during implementation. Remediation considerations will include data conversion, data clean-up, and other technology requirements.

After the procurement of a specific software solution is complete, One Washington will work with agencies to finalize the interface types, standards and formats. The One Washington program will include limited functional SMEs, developers and testers to work with agencies throughout implementation. During implementation, agency resources will be required to provide test files to or from the ERP. Further details on roles, responsibilities, and other considerations will be described in the Integration Implementation Plan.

4.0 Integration Methodologies

The integration strategy is based on the principle of leveraging service-oriented architecture (SOA) to provide automated real-time interfaces. SOA would allow agencies to send and receive data in a variety of formats and methods that support standard specifications. The standard specifications for integrations with the ERP will be further defined during the implementation design phase of the One Washington program. The integration plan will also need to support multiple implementation waves for Finance and Procurement. There will be one release for Budget and HR/Payroll. One Washington will plan for the availability of temporary interfaces between the ERP and legacy systems until all agencies are migrated to the ERP (see figure 2.5.1).

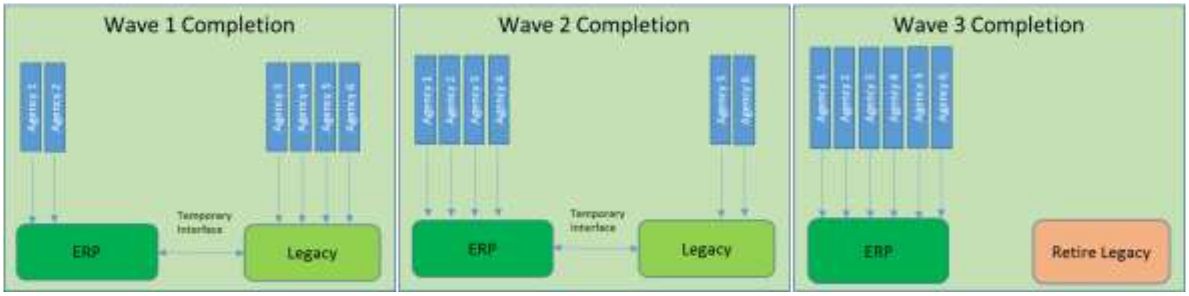

Figure 2.5.1: Implementation Waves

Integration with Inbound (to One Washington ERP) Interfaces

The preferred method of integration with the ERP is leveraging a SOA solution which provides the greatest flexibility. One Washington will also support direct and indirect interfacing methods with the ERP system:

1. Interface with middleware
2. Interface directly
3. Interface indirectly

Interface with middleware

Middleware involves using a secondary application which will connect an agency's line of business application to the ERP. This secondary application acts as an "interface layer" to the ERP application. Middleware typically involves the use of certain applications:

- Service-Oriented Architecture (SOA) Applications: SOA provides services to integrate different components over a network. It is independent of ERP applications and can be used for a wide variety of purposes, such as ETL (Extract, Transform, Load) and ESB (Enterprise Service Bus).
- Platform-as-a-Service (PaaS): PaaS is a cloud computing service that allows customers to develop and manage applications to support the ERP. Like SOA, this can also be used for a wide variety of purposes and in a SaaS solution provides an effective platform for developing custom applications.

Middleware provides added flexibility by removing limitations imposed by the ERP solution. Middleware can also provide added capabilities such as data transformation, crosswalks availability (mapping legacy to new solution) and business rules. This would be the recommended interface method and would provide a solution for the following scenarios for the state:

- Business needs are not met within the ERP application
- Data processing has complex functions, rules, or requirements
- The interface requires complex functions or calculations
- The ERP application does not deliver an API function for integration purposes

The One Washington program is researching current capabilities and tools that may be leveraged as a middleware (for example Informatica). Additional analysis and discussions are needed with the state technical teams and owners to evaluate if these tools would be the right choice for the ERP solution. The results of these analyses and discussions will be documented in the final Blueprint and Integration Implementation plan.

Interface directly

Direct interfacing involves direct communication with the ERP application. There are no middle systems to filter or pre-process data sent to or from the ERP application. Direct interfacing with the ERP involves the use of the following mechanisms:

- Application Programming Interfaces (API): APIs are a set of communication methods between software components. An API facilitates the communication between multiple systems. The advantage of APIs is that the interface is designed to be relatively independent of the programs or system it is interfacing with, thus providing greater flexibility to re-use the interface in a variety of different business functions.
- Web Services: Web services are services that facilitate the interaction between systems over a network like the internet. They are often used as a form of API. Web Services allow a standards-based integration point to expose underlying business functionality. Like the APIs, Web Services also have the advantage of being independent.
- ERP delivered loading/extracting mechanisms: ERP applications deliver mechanisms to process data from a specified location within the network. Files sent to the ERP application in a specific location can be directly processed by the ERP application.

The advantage of direct interfacing is that data can be processed quickly and synchronization issues are minimized. For example, Web Services watch for events like data being passed to it and process that data immediately. This requires that any consumer of the Web Service must adhere to all layouts and standards set by the owner of the Web Service. This standardization limits the many possible variations of data formats and forces other systems to comply with the ERP application's rules. The disadvantage is that sometimes, if no staging area is available, data gets loaded directly into the system. This is not desirable if certain data needs to be validated before loading.

Interface indirectly

Indirect interfacing involves using another application in conjunction with the ERP application or middleware. This application acts as a "middleman" to the middleware from the legacy system. Indirect interfacing is essentially the exceptions process in the event a system cannot utilize middleware or interface directly.

The advantage of using indirect interfacing is that it can be a link in the interface chain from very old legacy systems, with very limited functionality, to the newer ERP applications. An example is if source data is needed from a legacy system, but the middleware does not have the functionality to directly interface with that legacy system, then a special program will need to be created. That program would not necessarily load the data to the ERP application because there may be gaps in the data that the middleware is capable of filling in. he program would send the data to the middleware, the middleware will process the data, and send the data to the ERP.

Interfacing indirectly would be the least recommended method and used only on an exception basis. The indirect method has very little support for standardization, as the typical deployment of this solution is a customized solution to serve a unique function. The indirect method will only be used if the ERP or the middleware cannot perform the functions needed. Use of the indirect method would require an exception and would need to be used if:

- Neither the ERP application nor the middleware is capable of supporting the legacy/external system directly
- Special steps are required to integrate with either the ERP application or middleware

Integration with Outbound (from the One Washington ERP) Interfaces

The integration strategy will support capabilities and requirements to send outbound files to external systems. During the development of the integration implementation plan, the following analysis will be completed to determine the interface type:

1. Review the capabilities of the external system
2. Review the requirements of the external system
3. Apply the same interface methods described in the prior section: “Integration with Inbound Interfaces”

Reviewing the capabilities of the external system helps to determine what technology is required to deliver the interface. For example, if an external system cannot handle a certain file type, then other options will need to be explored.

Reviewing the requirement(s) will define any specific data or processing needs of the external system. An example of a requirement would be if a data element in the ERP application needs to be transformed into a coded value so that it can be loaded into the external system.

The general methodology for interfacing mirrors that of the “Integration with Inbound Interfaces” section. Based on the capabilities and the requirements of the external system, the best interface methodology should be chosen.

Data and Transmission Standards for the New ERP

The data and transmission standards described in this section encompass both new and legacy formats. Legacy formats may be leveraged due to technology and other constraints of interfacing legacy systems.

File Types

There are three main data standards available that can be used across most platforms. They are:

- Extensible Markup Language (XML): A file format used to create common information formats and share both the format and the data on the internet, intranets and elsewhere using standard ASCII text. XML is similar to HTML and is self-descriptive. It is also multi-dimensional and can handle complex data requirements.
- Flat File: A standard two-dimensional file standard that lists data in a specific format that a program can read. The available formats are:

  - Fixed Width: A file standard where the length of each input line must be the same as the other lines. Also, the size of the fields contained within each line must also be of a fixed size for each line in the exact same order.
  - Delimited: A file standard where the data fields in each line of input are separated by a signifier called a delimiter. The file layout must be the same for each line in the file, but the field size need not be consistent across different lines.
- Excel (for spreadsheet uploads): The standard spreadsheet format for office spreadsheets.

Modern ERP applications can support all the file types listed above. Older legacy systems mainly use flat files.

The One Washington program recommends the following priority order for file types:

1. XML
2. Flat Files
3. Excel

XML is the preferred format as it is the most versatile and is compatible with newer technology. It is also very robust and descriptive, making it easier to read and process. Flat files are the most widely supported option because all systems are capable of generating and processing flat files. It is second in the priority list because it has limitations in ERP applications, namely that the data is only two-dimensional, whereas ERP application data is multidimensional. Excel is last on the priority because not all ERP applications and legacy systems can support it.

The capabilities of each of the systems involved must be examined to decide on the file type to be used to integrate. Typically, the file type common to the systems involved would be used to integrate.

Encryption

Encryption is the process of transforming data into coded data to prevent unauthorized access. This process prevents an unauthorized person from viewing the data contents by recoding (or scrambling) the data in a way that no other program can read. The only way to read the data is by decrypting it and that requires the data consumer to have the "key" to the data.

Encryption can be used in two ways:

- Transport Layer Encryption – encrypting the communication channel of the data when sending
- File Encryption – encrypting the data file itself

Transport mechanisms like Secure File Transfer Protocol (SFTP) provide an encryption layer that encrypts data through the "transmission channel". The encryption of the transmission channel is an example of transport layer encryption and is often acceptable for many organizations' security requirements. The mechanism that transmits the file is encrypted. Despite this layer of encryption, the actual file remains unencrypted; the file is only protected by encryption while being transmitted.

One Washington will work with state and agency technical teams to align the file and transmission encryption approach to current and future policies and guidelines.

Transferring/Transmitting Files

Files generated from the ERP application and files that are loaded into the ERP application require transport; they need to be transmitted from or to other systems. The most common solution to perform that transmission is FTP (File Transfer Protocol). Section 4.4 "Secure Data Transfer" in Policy 141.10 requires the use of encryption in data transmission. Normal FTP servers and programs provide little security and are not encrypted. However, there are two secure and encrypted options: SFTP and FTPS. Some of their features are listed below.

FTPS	SFTP
• Secured connection – FTP using Secure Sockets Layer (SSL) protocol • User ID/Password authentication • SSL Certificate verification • Multiple Ports – Firewall incompatibilities are present because of port switching	• Secured connection – FTP over Secure Shell (SSH) protocol • User ID/Password authentication • Public Key authentication • Single Port – Easy firewall connection

The One Washington program will use an SFTP solution, as it is firewall-friendly and supports public key authentication. This solution provides high security without sacrificing ease of use and connectivity. Additionally, it meets all security requirements set by Policy 141.10. This solution can also be utilized by all agencies regardless of their legacy system capabilities because SFTP programs are independent systems. The only baseline requirement is a server to host the file transfer capabilities and files.

Currently, Washington provides an SFTP solution, referred to as Secure File Transfer (SFT). This product is likely capable of supporting the file transmission requirements for the One Washington program and is compliant with state security policies. The functionality will be examined further to determine if it is a web driven FTP service, Public Key authentication is supported, and connectivity with local SFTP programs for running it in batch are supported.

Batch Processing

Automation is the process of allowing interfaces to run without the need of user intervention. Instead of having a user manually control a process from start to end, the system executes batch files, or scripts, on a schedule that performs the same tasks. These tasks can range from being very simple to very complex. Automating these tasks increases the productivity of users because those users will no longer be involved in a potentially long and complex process. It also decreases the risks that steps may be accidentally skipped in a process or a file is produced with inconsistent data formats.

ERP applications themselves can automate some processes, but are typically limited to simple processes. For more complex processes, a batch scripting and scheduling program is required. In these cases, a “batch script” is created that tells the system what to do and what tasks it needs to perform to run a specific process, retrieve the generated output and then send it to an external system. Many times, it will be a combination of the two, where a process is scheduled to run on the ERP and batch scripts handle the resulting output. In any case, automation should be taken advantage of wherever possible to increase productivity.

The One Washington program will utilize a solution to manage batch scripts and schedules. Further examination into the batch scripting capabilities of the State of Washington is needed. Considerations include, but are not limited to:

- Current State of Washington scripting capabilities and software
- Capabilities of the scripting software to execute processes on remote SaaS ERP servers
- Business requirements to execute these automated processes in near real-time.

Any additional scripting software required to support the ERP solution will need to be acquired by the One Washington program. Scripting owner’s roles and responsibilities will be further defined in the Integration Implementation Plan.

5.0 Data Conversion

Data conversion is a key component of the Integration Implementation Plan. The data conversion approach is to convert data in waves:

- The initial functionality for Finance and Procurement will occur in three waves. In each wave, a set of state agencies will be implemented and converted.
- HR/Payroll and Budget will follow the Finance and Procurement rollout, and each of these business function will have one planned release.

Both data conversion and interface implementation are required for successful transition to the new ERP solution. Once all agencies are migrated, the ERP will be the single system of record for the corresponding business function. Further details on data conversion, including methodologies, are documented in the Blueprint data conversion section.

6.0 Summary

The Integration Strategy is aligned to the methodologies and approaches defined in the One Washington Program Blueprint to facilitate integration between the One Washington ERP applications (Finance, Procurement, HR/Payroll, and Budget) and other systems with which the ERP will interface. Finance and procurement functionality will be rolled out leveraging a phased agency/phased functionality approach. The strategies described in the Integration Strategy may be leveraged to an extent to support the phased roll-out approach. Further analysis and details on how the integration strategy will support the roll-out approach will be described in the Integration Implementation Plan. In the future state, the preferred method of integration with the ERP is leveraging a SOA solution which provides the greatest flexibility.