Approval of the Technical Architecture indicates an understanding of the purpose and content described in this deliverable. By signing this deliverable, each individual agrees with the content contained in this deliverable.

<table>
<thead>
<tr>
<th>Approver Name</th>
<th>Title</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Version: 1.0
Revision Date: March 16, 2016
Contents

SECTION 1 DOCUMENT SCOPE .................................................................................. 4

SECTION 2 OVERALL TECHNICAL ARCHITECTURE ............................................. 5

2.1 System Architecture Context Diagram ................................................................. 5
2.2 System Architecture Model .................................................................................... 6
2.2.1 Overall Architectural Considerations ................................................................. 6
2.3 System Architecture Component Definitions ....................................................... 8
2.3.1 User Interface / Presentation Layer .................................................................... 8
2.3.2 Dependency Injection ......................................................................................... 9
2.3.3 Service Layer ..................................................................................................... 9
2.3.4 Data Layer ......................................................................................................... 9
2.3.5 Testing Framework ............................................................................................. 10
2.3.6 Enterprise Services .......................................................................................... 10

SECTION 3 SYSTEM ARCHITECTURE DESIGN ................................................. 12

3.1 User Interface / Presentation Layer .................................................................... 12
3.1.1 Component Functions ....................................................................................... 12
3.1.2 Technical Considerations .................................................................................. 12
3.1.3 Selected Product(s) ......................................................................................... 12
3.1.4 Selection Rationale ......................................................................................... 12
3.1.5 Architecture Risks ......................................................................................... 12
3.2 Dependency Injection .......................................................................................... 12
3.2.1 Component Functions ....................................................................................... 12
3.2.2 Technical Considerations ................................................................................ 12
3.2.3 Selected Product(s) ......................................................................................... 12
3.2.4 Selection Rationale ......................................................................................... 13
3.2.5 Architecture Risks ......................................................................................... 13
3.3 Service Layer ....................................................................................................... 13
3.3.1 Component Functions ....................................................................................... 13
3.3.2 Technical Considerations ................................................................................ 13
3.3.3 Selected Product(s) ......................................................................................... 13
3.3.4 Selection Rationale ......................................................................................... 13
3.3.5 Architecture Risks ......................................................................................... 13
3.4 Data Layer ........................................................................................................... 13
3.4.1 Component Functions ....................................................................................... 13
3.4.2 Technical Considerations ................................................................................ 13
3.4.3 Selected Product(s) ......................................................................................... 14
3.4.4 Selection Rationale

3.4.5 Architecture Risks

3.5 Testing Framework

3.5.1 Component Functions

3.5.2 Technical Considerations

3.5.3 Selected Product(s)

3.5.4 Selection Rationale

3.5.5 Architecture Risks

3.6 Enterprise Services

3.6.1 Component Functions

3.6.2 Technical Considerations

3.6.3 Selected Product(s)

3.6.4 Selection Rationale

3.6.5 Architecture Risks

SECTION 4 SYSTEM CONSTRUCTION ENVIRONMENT

4.1 Development Environment

4.1.1 Developer Workstation Configuration

4.1.2 Supporting Development Infrastructure Configuration

4.2 QA Environment

4.2.1 QA Workstation Configuration

4.2.2 Supporting QA Infrastructure Configuration

4.3 Acceptance Environment

4.3.1 Acceptance Workstation Configuration

4.3.2 Supporting Acceptance Infrastructure Configuration

LIST OF FIGURES

Figure 1 System Architecture Context Diagram

Figure 2 System Architecture Model

Figure 3 Production System Environment
Section 1 DOCUMENT SCOPE

Document Scope describes the context and the goals of this document in a narrative.

This document describes the Technical Architecture of the Procurement Development Application that satisfies business requirements as documented in the PDA High Level Requirements (As-Is and To-Be) Document, February 14, 2018 and implements the functionality and satisfies technical, operational and transitional requirements described in FDOT-OIT RFQ Procurement Development Application.

The goal of this Technical Architecture is to define the technologies, products, and techniques necessary to develop and support the system, and to ensure that the system components are compatible and comply with the enterprise-wide standards and direction defined by the Agency.

This document will also:

1. Identify and explain the risks inherent in this Technical Architecture;
2. Define baseline sizing, archiving and performance requirements;
3. Identify the hardware and software specifications for the Development, Testing, QA and Production environments;
4. Define procedures for both data and code migration among the environments.

The Document Scope narrative also provides an overview of the efforts conducted to understand the existing technical environment and IT strategic direction and to determine how the system’s proposed technical architecture fits into them.
Section 2 OVERALL TECHNICAL ARCHITECTURE

2.1 System Architecture Context Diagram

The System Architecture Context Diagram provides the “big picture” view of the system’s architecture, and puts it in context with the rest of the Performing Organization’s systems portfolio, illustrating how the system’s hardware and software platforms fit into the existing environment.

![System Architecture Context Diagram]

*Figure 1 System Architecture Context Diagram*
2.2 System Architecture Model
The System Architecture Model represents the various architecture components that comprise the system, and shows their interrelationships.

![System Architecture Model Diagram]

2.2.1 Overall Architectural Considerations
The Overall Architectural Considerations section defines how additional technical requirements have been addressed by the architecture.

Security Strategy
PDA utilizes Active Directory as the standard mechanism to explicitly authenticate users. Once authenticated, the application will utilize role based security to manage application permissions.

The web application resides within the FDOT Intranet and automatically authenticate anyone on the FDOT network using his/her Windows credentials (AD).

A clear distinction is made between FDOT employees and consultants. This is determined by the staff type returned by SRS for each user.

Performance Requirements

In compliance with section 4.1 of FDOT Web Application Standards (7/7/2014), all requested content will be received by the browser within 10 seconds of the user action.

Accessibility

The application adheres to compliance standards set by Section 508 of the Rehabilitation Act of 1973. The Section 508 Standards are part of the Federal Acquisition Regulation (FAR) and address access for people with physical, sensory, or cognitive disabilities.

508 compliance testing is performed using software based testing tools designed to identify potential issues.

Database Sizing

Currently the grading system is using 1054 cylinders of space. This is the equivalent of approximately 875 megabytes. There are a total of 13,265 contracts in the system with an average of approx. 600 professional services / design build contracts, 450 commodities / contractual services contracts and 3500 professional services / contractual services amendments entered each year.

Sizing for the new data will depend on the finalized schema and the data / transaction volumes for the new work.

Transaction Volumes

Per FDOT, to be determined at a later date.

Concurrent Users

The number of concurrent user’s is anticipated to be low. As per FDOT, the total number of concurrent users to be supported by PDA will be determined at a later date.

Data Import and Export

Consultant Evaluation (CE) information is imported to the PDA system by a nightly batch process.

Contract information is exported to the Consultant Evaluation (CE) system by a nightly batch process.

Federal Oversite information is imported from the Work Program via Information Data Warehouse (IDW) system by a nightly batch process.

Residual Encumbrance Balance information is imported from Contract Funds Management System (CFM) system by a nightly batch process.

Financial Data for BDI reports is imported from Consultant Invoice Transmittal System (CITS) system by a nightly batch process.
AFP Wage Rate Data is imported by a nightly batch process.

Disadvantaged Business Enterprise (DBE) is imported from Equal Opportunity Compliance (EOC) system by a nightly batch process.

The PDA system publishes advertisements to the department’s procurement website by an hourly batch job. New advertisements are published on a weekly basis with updates published every hour.

The PDA system sends email notifications to consultants and Contract mailer subscribers, to inform them of changes or updates to advertisements, by an hourly batch process.

Files uploaded to the PDA system are inserted in to the EDMS system by an hourly batch process.

**Data Encryption and Decryption**

Access to the application via web browser utilizes Transport Layer Security (TLS) version 1.1 or greater. The primary goal of the TLS protocol is to provide privacy and data integrity between two communicating computer applications.

**Disaster Recovery**

Backup and recovery functions have two major components:

1. Database backup and recovery procedures
2. Application server backup and recovery procedures

The DBAT group is responsible for ensuring the integrity and recoverability of MS SQL Server databases for enterprise applications.

Agency for State Technology’s State Datacenter (SDC) is responsible for maintaining the reliability and integrity of the servers that house the Department’s applications, and for standard data backup and retention of the servers.

The system’s data and source code will be backed up and recovered based on the standard OIT Enterprise Application backup and recovery processes. For more information on backup and recovery procedures for databases and servers, please contact each respective group.

### 2.3 System Architecture Component Definitions

The **Architecture Component Definitions** section provides narrative describing and explaining each architecture component in the System Architecture Model, and identifies specific elements that comprise that component in this system.

#### 2.3.1 User Interface / Presentation Layer

The PDA application utilizes ASP.NET MVC to provide a dynamic UI. Views are in HTML5 / CSS3 and utilize the Razor engine to populate and fill the view for use by the users.

The application leverages ASP.NET MVC controllers to provide the basic functionality for the application. These controllers provide a means of integrating business units (presenters and providers), and enable them to work within the systems UI. The controllers utilize a façade pattern, in that the controllers will not offer any business logic, but instead are responsible for providing a means of integrating the existing business units to allow them to complete their jobs.
The application is accessed via standard web browser. Users may be either FDOT employees or consultants and are authenticated via Active Directory. Doing so ensures that users are authenticated against safe, reliable and maintained authentication end points.

The initial release of PDA supports the following browser/versions:

- Internet Explorer (IE) 11
- Google Chrome (latest version)

2.3.2 Dependency Injection
Dependency injection is a software design pattern that implements inversion of control. This allows all classes of the application to be largely decoupled, and more stable. Dependency injection also allows developers to write more testable code. When dependencies can be injected into a component it is possible to inject mock implementations of these dependencies.

As it is currently an FDOT approved technology, PDA utilizes Ninject as the dependency injection container. Ninject is a lightweight dependency injection framework for .NET applications.

2.3.3 Service Layer
The service layer contains all of the business/domain logic. These are the rules and validation specific to the PDA application and particular function being processed. The business layer is composed of Presenters, Providers and Interfaces.

Presenters are classes that encapsulate the business rules of the application. They are the types of classes that are injected into the controllers, and the methods are designed to return ViewModels which are utilized in the passing of data back through the controller and up to the view. Providers and interfaces allow abstraction of reusable code allowing the application to reuse and utilize code for smaller tasks that might be performed at several points in the service layer.

In this layer all dependencies are constructor injected allowing the application to remain loosely coupled.

2.3.4 Data Layer
ORM
The data layer is a class library designed to handle interactions against the database. The data layer utilizes NHibernate, which is an Object-Relational Mapper. The goal of an ORM solution is to map an object-oriented domain model to a traditional relational database and reduce the amount of persistence-related programming.

Design Patterns
The Repository pattern is used to separate data access logic from the application into separate container classes. A separate repository is created for each Domain area of the application, which then manages the CRUD operations.

Supporting the Repository pattern, the application implements the Unit of Work pattern which is used to group one or more operations into a single transaction or “unit of work”.

Reporting
Reporting needs of the application is handled via Microsoft SQL Server Reporting Services. Additional reporting is available utilizing FDOT approved reporting tools.

Enterprise Services
Enterprise Service Adapters in this layer are responsible for interacting with the FDOT Enterprise Library.
2.3.5 Testing Framework
To ensure that code is working as expected, PDA includes a separate Visual Studio project for unit tests which utilizes the Microsoft Unit Testing Framework. Unit testing breaks down the functionality of the application into discrete testable behaviors that can be tested as individual units.

In order to facilitate this testing, the application incorporates mocking where applicable. Mocking is a process used in unit testing when the unit being tested has external dependencies. The purpose of mocking is to focus on the code being tested and not on the behavior or state of external dependencies.

2.3.6 Enterprise Services
PDA utilizes the FDOT Enterprise Library (FEL) Version 4.0. The following services are used:

- FDOT Enterprise Codes Service (DOTCODES) – DOTCODES is used to access enterprise codes.
- Database Connection Service (DBConnect) – DBConnect is part of the FDOT infrastructure and FDOT Enterprise Library used to obtain and manage connections to the database and any other secured resource using explicit authentication.
- Staff Repository Service (SRS) – The repository of all internal and external staff.
- Professional Services Information Prequalification (PSI-PQ) – Contains all vendors who are pre-qualified to do work on professional services contracts with their contact information and links to their vendor information in TVI.
- Transportation Vendor Information (TVI) – Professional Services Prequalification (PSI-PQ) – contains all vendors who are pre-qualified to do work on professional services contracts with their contact information and links to their vendor information in TVI. If PSI-CE only needs to read PSI-PQ vendor information stored in TVI then it can be read directly from the DB2 database but if any update on this data is required then the use of FEL 2 will be necessary.
- Consultant Evaluation (CE) – Contains Consultant Evaluations. Information from this system is batch loaded into PDA by a nightly basis.
- Work Program via Information Data Warehouse (IDW) – Contains Federal Oversite field. Information from this system is batch loaded into PDA by a nightly basis.
- Contract Funds Management System (CFM) – Contains Residual Encumbrance Balance. Information from this system is batch loaded into PDA by a nightly basis.
- Consultant Invoice Transmittal System (CITS) – Contains Financial Data for BDI reports. Information from this system is batch loaded into PDA by a nightly basis.
- AFP Wage Rate Data (AFP) – Contains the Wage Rate Data submitted by vendors. Information from this system is batch loaded into PDA by a nightly basis.
- Equal Opportunity Compliance System (EOC) – Contains the Disadvantaged Business Enterprise (DBE) Certification status. Information from this system is batch loaded into PDA by a nightly basis.
- Electronic Document Management System (EDMS) – Contains the upload documents associated with Advertisements, Contracts and Vendor Responses.
Section 3 SYSTEM ARCHITECTURE DESIGN

The System Architecture Design section provides detailed descriptions of each product implementing architecture components, and explains the rationale for product selection.

For each System Architecture Component (identified in Section 2.3 above), the narrative describes specific Component Functions, requirements and other Technical Considerations that were used in the decision-making process, as well as any specific Products selected to implement this component. The Selection Rationale identifies any other products that may have been considered, and provides rationale for the decision. Architecture Risks identifies any potential risks associated with the architecture element.

3.1 User Interface / Presentation Layer

3.1.1 Component Functions
PDA uses the Microsoft ASP.Net MVC 5 web application framework to implement the model-view-controller pattern. The purpose of this layer is to support routing and rendering of the application.

3.1.2 Technical Considerations
Provides a dynamic UI that promotes ease of use.

3.1.3 Selected Product(s)
- ASP.NET with Razor syntax
- Telerik Kendo UI Core

3.1.4 Selection Rationale
These products were chosen based on a number of factors:
- They align with the goals and outcomes of the application.
- They are FDOT supported technologies.
- They promote ease of maintainability and extensibility.
- Do not require any third party licenses.

3.1.5 Architecture Risks
Risks are minimal as all technologies in this layer are widely adapted in web development. It should be noted that while MVC version 5 is the most current stable release, MVC version 6 is currently in release candidate.

3.2 Dependency Injection

3.2.1 Component Functions
Software design pattern that implements inversion of control for resolving dependencies.

3.2.2 Technical Considerations
Allows the application layers to be largely decoupled allowing for better management of dependencies.

3.2.3 Selected Product(s)
- Ninject
3.2.4 Selection Rationale
- Widely used technology
- Does not require a 3rd party license
- An approved FDOT technology

3.2.5 Architecture Risks
Ninject is open source software. Future support of the technology relies on continued community interest.

3.3 Service Layer

3.3.1 Component Functions
To process business/domain logic and data retrieval tasks.

3.3.2 Technical Considerations
- There should be no concrete dependencies in this layer. All dependencies should be constructor injected.
- Minimize complexity.

3.3.3 Selected Product(s)
- Standard class libraries.

3.3.4 Selection Rationale
N/A

3.3.5 Architecture Risks
None

3.4 Data Layer

3.4.1 Component Functions
- Abstract and decouple the physical database from the application.
- Read/persist data within the database.
- Consume FDOT enterprise services.

3.4.2 Technical Considerations
- Application should know nothing about the physical database
- The data layer should utilize an FDOT approved ORM.
- The data layer must be able to communicate with FDOT Enterprise Services for common functionality.
3.4.3 Selected Product(s)
- NHibernate
- Microsoft SQL Server Enterprise 2012 SP3 64 bit
- IBM DB2 Connect client software to connect to DB2 for z/OS. This is a 32-bit software product.

3.4.4 Selection Rationale
- Widely used technology
- Does not require a 3rd party license
- An approved FDOT technology

3.4.5 Architecture Risks
NHibernate is open source software. Future support of the technology relies on continued community interest.

3.5 Testing Framework

3.5.1 Component Functions
- Provide a framework to support unit testing.

3.5.2 Technical Considerations
- Must support unit testing
- Must be an FDOT approved technology

3.5.3 Selected Product(s)
- Microsoft Unit Testing Framework
- Moq - a testing framework for mocking dependencies.

3.5.4 Selection Rationale
- Widely used technology
- Does not require a 3rd party license
- An approved FDOT technology

3.5.5 Architecture Risks
None

3.6 Enterprise Services

3.6.1 Component Functions
A code library that provides common functionality to FDOT applications.
3.6.2 Technical Considerations
- The Enterprise Library provides a standard set of functionality and interfaces to applications.
- Certain functionality contained in the Enterprise Library is required by all applications.
- Required-as-needed enterprise services meet the requirements of the application.

3.6.3 Selected Product(s)
- FDOT Enterprise Library (FEL) V 4

3.6.4 Selection Rationale
- Required technology

3.6.5 Architecture Risks
None
Section 4 System Construction Environment

The System Construction Environment section details the various environments necessary to enable system construction and testing.

Figure 3 Production System Environment

The Public Web Server, located in the DMZ, hosts the PDA Vendor Response application and is used by Vendors to submit responses to procurement advertisements.

The Web Server, located on the Intranet, hosts the PDA application.

The SQL Server database hosts the PDA applications data.

The Enterprise Data. See section 2.3.6 Enterprise Services for a complete list of Enterprise data accessed by PDA.

4.1 Development Environment

Develop occurs at the Computer Aid Inc. Harrisburg, PA development center. Some development resources, such as the development database are housed on the Microsoft Azure platform.

Developer workstations are configured as follows:

4.1.1 Developer Workstation Configuration

- Operating System: Window 7 Professional
- Development Software: Visual Studio Enterprise 2015
- Microsoft .NET Framework version 4.6.2

4.1.2 Supporting Development Infrastructure Configuration

- Source Control: Visual Studio Team Services
• Development Public Web Integration Server. This environment is utilized to test code integration during the development phase.
  o Location: Microsoft Azure
  o Operating System: Microsoft Windows Server 2012 R2
  o IIS Version: Microsoft Internet Information Services 8.0

• Development Web Integration Server. This environment is utilized to test code integration during the development phase.
  o Location: Microsoft Azure
  o Operating System: Microsoft Windows Server 2012 R2
  o IIS Version: Microsoft Internet Information Services 8.0

• Development Database Integration Server. This environment is utilized to host the development database.
  o Location: Microsoft Azure
  o Operating System: Microsoft Windows Server 2012 R2
  o SQL Server Version: Microsoft SQL Server Standard 2012 SP3 64-bit

Test implementations of the FEL 4 services are utilized where available to facilitate development when not connected to the FDOT internal network.

4.2 QA Environment

4.2.1 QA Workstation Configuration
Access to PDA is accomplished via standard web browser. Users must however be on the FDOT network to connect to the application.

Supported Browsers:
• Microsoft Internet Explorer Version 11
• Google Chrome – Latest version

4.2.2 Supporting QA Infrastructure Configuration

Network
• The QA environment resides and is only accessible via the FDOT network

Public Web Server Configuration
• Operating System: Microsoft Windows Server 2012 R2
• IIS Version: Microsoft Internet Information Services 8.0
• .NET Framework version: 4.6.2

Web Server Configuration
• Operating System: Microsoft Windows Server 2012 R2
• IIS Version: Microsoft Internet Information Services 8.0
• .NET Framework version: 4.6.2

Database Server Configuration
• Operating System: Microsoft Windows Server 2012 R2
• SQL Server: Microsoft SQL Server Enterprise 2012 SP3 64-bit
4.3 Acceptance Environment
For each environment necessary for system construction (Development, QA and Acceptance), provide detailed specifications for the Workstation and Supporting Infrastructure that will be used (including hardware, network and operating system requirements, all necessary installed packages and tools, and needed directory structures that will be utilized to store all construction components).

4.3.1 Acceptance Workstation Configuration
Access to PDA is accomplished via standard web browser. Users must however be on the FDOT network to connect to the application.
Supported Browsers:
- Microsoft Internet Explorer Version 11
- Google Chrome – Latest version

4.3.2 Supporting Acceptance Infrastructure Configuration
Network
- The QA environment resides and is only accessible via the FDOT network

Public Web Server Configuration
- Operating System: Microsoft Windows Server 2012 R2
- IIS Version: Microsoft Internet Information Services 8.0
- .NET Framework version: 4.6.2

Web Server Configuration
- Operating System: Microsoft Windows Server 2012 R2
- IIS Version: Microsoft Internet Information Services 8.0
- .NET Framework version: 4.6.2

Database Server Configuration
- Operating System: Microsoft Windows Server 2012 R2
- SQL Server: Microsoft SQL Server Enterprise 2012 SP3 64-bit