# Revision History

<table>
<thead>
<tr>
<th>Description</th>
<th>Date</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Version</td>
<td>10/30/15</td>
<td>1.0</td>
</tr>
<tr>
<td>Updates after major architecture changes, and addressing feedback from</td>
<td>7/13/2016</td>
<td>2.0</td>
</tr>
<tr>
<td>government review.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Table of Contents

1 INTRODUCTION .................................................................................................................. 6

2 GENERAL OVERVIEW AND DESIGN GUIDELINES/APPROACH ..................................... 6

   2.1 Assumptions / Constraints / Standards ........................................................................ 6

   2.1.2 Assumptions ........................................................................................................... 6

   2.1.3 Constraints .......................................................................................................... 6

   2.1.4 Standards ............................................................................................................. 6

   2.2 Approach .................................................................................................................. 6

3 ARCHITECTURE DESIGN ................................................................................................... 7

   3.1 Logical View ............................................................................................................ 7

   3.2 Hardware Architecture ............................................................................................ 9

   3.2.2 The Cloud-Hosted Environment ........................................................................... 9

   3.3 Software Architecture ............................................................................................ 11

   3.3.1 Application Tiers ................................................................................................ 11

   3.3.1.1 Web Client ........................................................................................................... 11

   3.3.1.2 Web Services Tier .............................................................................................. 12

   3.3.1.3 Business Tier ................................................................................................... 12

   3.3.1.4 Data Tier .......................................................................................................... 12

   3.3.1.5 Subsystem Messaging ..................................................................................... 12

   3.3.1.6 Server Utilities ................................................................................................. 12

   3.4 Security Architecture ................................................................................................ 12

   3.4.1 Cloud Account Creation ...................................................................................... 12

   3.4.2 SDSFIE Online Authorization & Authentication .................................................. 12

   3.4.3 SSL/TLS .............................................................................................................. 13

4 SYSTEM DESIGN ................................................................................................................. 13

   4.1 Database Design ....................................................................................................... 13

   4.1.1 Identity ................................................................................................................ 13

   4.1.2 Ticketing ............................................................................................................. 14

   4.1.3 Registry4 ............................................................................................................. 15

   4.1.4 Application ......................................................................................................... 18

   4.1.5 Backups .............................................................................................................. 19
4.2 Application Program Interfaces .......................................................................................... 19
  4.2.1 SdsfieDataAccess ....................................................................................................... 19
  4.2.2 OnlineLib .................................................................................................................. 19
4.3 User Interface Design ....................................................................................................... 19
4.4 Interaction with Registry4 ............................................................................................. 20
4.5 System Performance ........................................................................................................ 20

5 CLOUD IMPLEMENTATION .......................................................................................... 20
  5.1 Launching an Amazon Instance .................................................................................. 20
  5.2 Required Ports & Protocols, Web Servers .................................................................. 22
    • RDP 3389 (open) ............................................................................................................. 22
    • HTTP 80 (open) ............................................................................................................. 22
    • HTTPS 443 (open) ........................................................................................................ 22
    • TCP/IP to MSSQL 1433 ............................................................................................ 22
  5.3 Required Ports & Protocols, Database Server ............................................................. 22
    • RDP 3389 (open) ............................................................................................................. 22
    • TCP/IP to MSSQL 1433 ............................................................................................ 22
  5.4 Disaster Recovery ........................................................................................................ 22

6 INSTALLING TOOLS ..................................................................................................... 23
  6.1 Installing Amazon Elastic Compute Cloud (EC2) Tools .............................................. 23

7 Architectural Needs ........................................................................................................ 25

Appendix 1: List of Acronyms ............................................................................................ 26
Table of Figures

Figure 1: Cloud-hosted Web Diagram ............................................................................................................................. 7
Figure 2: High-Level System Diagram .......................................................................................................................... 8
Figure 3: Cloud Hosting Overview ............................................................................................................................ 10
Figure 4: Amazon Instance State Diagram .................................................................................................................. 10
Figure 5: Web Client .................................................................................................................................................... 11
Figure 6: RBAC Database Diagram ............................................................................................................................. 14
Figure 7: OTRS Database Diagram .............................................................................................................................. 15
Figure 8: Registry4 Configuration Table and Relationships ............................................................................................. 16
Figure 9: Registry4 Model Element Table and Relationships ........................................................................................ 17
Figure 10: Application Object Database Diagram ......................................................................................................... 18
Figure 11: Amazon Instances in the UI ......................................................................................................................... 21
Figure 12: SDSFIE Production Web Server Details ...................................................................................................... 21
Figure 13: SDSFIE UAT Web Server ............................................................................................................................ 21
Figure 14: SDSFIE Production OTRS Server ................................................................................................................ 22
Figure 15: SDSFIE Staging OTRS Server .................................................................................................................... 22
Figure 16: Access Key Created For Download ............................................................................................................... 23
Figure 17: My Computer->Properties ........................................................................................................................... 24
Figure 18: System Properties Advanced Settings ........................................................................................................ 24

Table of Tables

Table 1: SDSFIE Production Instance Specifications .................................................................................................. 20
Table 2: OTRS Production Instance Specifications .................................................................................................... 20
1 INTRODUCTION

The system architecture document outlines the hardware and software configurations for the Spatial Data Standards for Facilities, Infrastructure and Environment (SDSFIE) website application, “SDSFIE Online”. The intended audience for this document is the project manager, project development team, and stakeholders.

2 GENERAL OVERVIEW AND DESIGN GUIDELINES/APPROACH

This section describes the policy and technical bounds used in designing and implementing the SDSFIE system.

2.1 Assumptions / Constraints / Standards

2.1.2 Assumptions

- All information provided by the government and made available by the system does not require classification or marking considerations; making it suitable for public release.

- The system does not host geospatial ‘data’, rather content and metadata standards related to geospatial data models.

2.1.3 Constraints

- The system is unconstrained by Department of Defense Security Technical Implementation Guides (DoD STIGs). The system is not intended for deployment within a DoD enclave; no Authority To Connect (ATC) is required.

2.1.4 Standards

- Federal Risk and Authorization Management Program (FedRAMP) controls are primarily managed by the cloud service provider, Amazon Web Services (AWS). This process has been initiated by AWS. FedRAMP considerations may be additionally addressed under separate documentation.

- Though some system products conform to published data standards of various types, there are no generalized architectural standards that apply to the system itself.

2.2 Approach

The system requirements in the scope of work (SOW) suggest a cloud-hosted web solution. This solution addresses:

- geographically disparate end users

- central management of access controls

- wide variance in end user environments, especially domain security
3 ARCHITECTURE DESIGN
This section outlines the system and hardware architecture of the system.

3.1 Logical View
The general solution for a cloud-hosted web solution is very simple: end users access a web client via the client host (a web server). The client host accesses data from a data store host as illustrated in Figure 1.

The SDSFIE Online web application is created by integrating multiple subsystems with distinct roles. The subsystems are listed below.

- Database Server
- Web Server
- Open Technology Real Services (OTRS) Ticketing System
- Amazon Web Services (AWS) Simple E-mail Service (SES)

Figure 2, below, describes the major system elements, their subsystems and the communication between the systems/subsystems.
Below is a numbered list associated with the diagram where each number describes the subsystem marked in the diagram and its communication with the other subsystems.

1. Web Tier: The web tier is a subsystem on the AWS server which houses the website application. The application is an integration of multiple components which are all necessary for the application.
   a. The Authentication and Authorization component is done using Microsoft Identity 2.0 and is responsible for verifying username/password combinations, approval status and roles.
   b. The Web Client component communicates with the client workstations to display the necessary web pages and content.
   c. The Web Services component is responsible for communicating with the system database to gather the appropriate data for the web client to pass along to the user.
   d. The Ticketing Service component communicates with OTRS to provide help desk ticket status to the user and allow users to enter help desk ticket information.

2. Data Tier: The data tier is a subsystem on the AWS server which consists of two databases, a user database and a model database, and the manipulation of the data in these databases through SQL statement calls or stored procedures.
   a. The user database tracks the user information, passwords, roles and model/workflow access.
   b. The model database stores the SDSFIE metadata content for the Gold organization and the adaptations created by the military components.
3. Mail Services: The mail services exist on a second AWS, server separate from the Web and Data Tiers, and is controlled entirely by AWS.
   a. The mail server redirects e-mail to the OTRS Ticketing system to enter help desk tickets from users
   b. The mail server sends e-mails to administrators and component managers as notifications for website and additional roles access.

4. OTRS Ticketing System: The OTRS Ticketing System is a subsystem on a third AWS server, separate from the other two already being utilized for the system. It consists of a database and a mail server.
   a. The OTRS Ticketing System database stores help desk ticket information such as person submitting ticket, e-mail, bug found and status of the ticket.
   b. The OTRS Ticketing System mail server is used to send reply and ticket status updates to the users via the AWS SES. It also generates new tickets from incoming e-mail messages.

5. User Environments: Any workstation which can connect to the internet. These workstations can exist anywhere and be behind different levels of firewalls and security. They have different internet browsers which are used to access the web application but the preferred browsers are FireFox and Internet Explorer 11 or later.

3.2 Hardware Architecture

3.2.2 The Cloud-Hosted Environment
Cloud computing enables the SDSFIE web application to obtain a flexible, secure, and cost-effective IT infrastructure. Currently, the cloud environment provides the SDSFIE application with compute power, storage, databases, and messaging. Services currently being leveraged by the cloud are outlined in Figure 3 (next page). AWS components which are available but not enabled for our current system architecture are struck through with a red line in Figure 3.
Figure 3: Cloud Hosting Overview

Figure 4, below, illustrates the state diagram of an Amazon cloud instance.

Figure 4: Amazon Instance State Diagram
3.3 Software Architecture

The application architecture is a ‘tiered’ architecture. Separate components are responsible for service, business, data, and UI responsibilities. The underlying technology used is ASP.Net (c#) and JavaScript. In addition to the custom code written, these third-party toolsets are leveraged:

- ICSwithCode.SharpZipLib: File compression software used for zipping up generated XML/Excel documents and unzipping uploaded zip files in validation
- Saxonica: XML Schema Validation
- SyncFusion: Plugin for allowing the code to easily export Excel documents and for the code to be written easier when importing and reading through imported Excel documents
- Error Logging Modules and Handlers (ELMAH): Used for logging errors to the database for debugging user errors
- SignalR: Server/Client communication plugin for long server processes i.e. generating documents
- Dapper.Net: Object-relational mapping
- Jil: JSON messaging serialization, which is faster and more reliable than NewtonSoft package
- ExtJS 6.0.2: JavaScript UI Framework upgraded from ExtJS 4.1.1 and ExtJS 3.1.2 packages previously in the code
- OTRS: Ticketing tracking system
- Bootstrap: CSS support selected for its ability to provide layouts for different screen sizes

3.3.1 Application Tiers

3.3.1.1 Web Client

The web client technologies are JavaScript, ExtJS, bootstrap, and ASP.Net. Page-scoped server-side processing utilizes the ASP.Net page event model. Layout is achieved via a combination of bootstrap CSS styles and native ExtJS style. Some elements of client-side data processing are handled with standalone JavaScript. While the ExtJS components follow a loose interpretation of MVC, generally the application is best described as autonomous-UI. ExtJS components contain native XML-http support, and as such require no discrete web services proxy tier. See Figure 5.

![Figure 5: Web Client](image-url)
3.3.1.2 Web Services Tier
The web services tier, in general, is a pass-through tier. It exposes the web endpoints to the client. In pass-through cases, parameters are deserialized into .Net types and passed to business logic in the business tier. Return types are serialized JSON.

3.3.1.3 Business Tier
The business tier is implemented as a .Net library. Object representations of business objects (e.g. users, reports, data transformations…) are defined and implemented here.

3.3.1.4 Data Tier
In general, data access to the system data stores are implemented here. In the general case, repository types invoke stored procedures to return various representations of recordsets or scalars. In some cases, dynamic SQL is generated in server-side code and invoked via database command objects, bypassing the data access tier. Largely, this is a legacy implementation that predates the data tier implementation.

3.3.1.5 Subsystem Messaging
Messaging between subsystems occurs via three protocols depending on the coupled systems: SMTP for mail interactions, HTTP for web service calls, and TCP/IP for database command invocations. In addition, synchrony between subsystem states in some circumstances is managed via both programmatic CRUD operations on coupled data stores and some manual interventions in data content.

3.3.1.6 Server Utilities
Server utility features are implemented as plugins to a common console application, Sdsfie.ServerUtilities.exe.

3.4 Security Architecture

3.4.1 Cloud Account Creation
The cloud account management is the responsibility of the contracting company.

Account creation, at the cloud management interface level is completed using the following procedure:

2. Follow the onscreen instructions.

AWS Identity and Access Management (IAM) are implemented to provide user-specific credentials for making AWS infrastructure requests.

Access to SDSFIE AWS resources is given by creating users under the AWS SDSFIE account. The Administrator group and user group has been established to provide access.

3.4.2 SDSFIE Online Authorization & Authentication
The system integrates a custom Role-based Access Control scheme that is fully integrated with ASP.NET Identity 2.0. As system users register for access, their accounts are eligible for enabling. Administrative system users – generally characterized as ‘Component Managers’ or ‘Helpdesk’ – enable accounts and grant roles. All system web services are public and enable HTTP GET. As such, they are accessible to
any web client and offer an unchecked mechanism for invoking web methods to retrieve and insert data, or invoke system processes.

3.4.3 SSL/TLS
The web application utilizes SSL/TLS for information security in transit. The Open Source Ticket Request System (OTRS) instance, on the other hand, does not utilize SSL and is accessible to the public internet via HTTP. OTRS information is therefore vulnerable to man-in-the-middle attacks at any stage of the message route. This is a concern and should be looked at in the future to discuss best course of action and what level of risk is considered acceptable to the IGI&S Governance Group (IGG).

4 SYSTEM DESIGN
4.1 Database Design
The system utilizes four discrete data stores, each with their own general responsibilities. In some cases, relationships are unconstrained. Largely, unconstrained relationships utilize natural keys rather than synthetic (or primary) keys. These relationships are vulnerable to orphan records in the event that a natural key’s value changes.

4.1.1 Identity
Identity is the updated authorization and authentication plugin by ASP.NET. Identity has persistent authorization and authentication objects to allow site access control and system privileges. This includes the intrinsic ASP.NET Identity objects and custom objects for user registration and Role-Based Access Control (RBAC). Fine-grained access control is achieved by relating Authorized Features to Roles, where Authorized Features may be web application features, datasets, data filters, and other controlled elements.

User information is contained in three separate tables. One table (AspNetUsers) tracks all of the user personal information such as Username, password, E-mail, Name, Telephone, Etc. There is a second relationship table which is connected using a Unique User Id which relates the user with a Service ID. The third table relates the Service ID to the service name.

The Role Based Access Control which is used to determine website features and access levels for each user is described in the Figure 6 below.

- Aspnet_Roles – Lists all of the roles which are available in the SDSFIE application.
- Aspnet_UsersInRoles – Table which ties the role to the user.
- Aspnet_Users – Table containing information about the user.
- RoleAuthFeature – Table which ties roles to authorized features so that roles translate into privileges in the application.
- AuthorizedFeature – Table which contains the authorized features for the application.
- AuthorizedFeatureType – Table which ties the authorized features to an authorized feature type (i.e. Browse, Generate, Validate)
- ComponentRole – Table which ties roles to a specific component.
• ComponentAllowableRoles – Table which lists the roles a user is allowed based on the component
• Rule – Table which associates roles to certain rules (i.e. is public, apply by default for component)

4.1.2 Ticketing
The intrinsic objects from the OTRS subsystem are integrated with custom SDSFIE system-specific objects, which can be seen in Figure 7, to build a ticketing system where users can submit change requests and website tickets. The custom objects, prefixed ‘CMP’, are designed to be unconstrained by primary keys, unique keys, and foreign keys. This design allows for duplicates and orphaned records. All behavior of the CMP tables must be inferred from usage, as these tables are used to interface with the OTRS software which provides limited documentation on the functionality.

These challenges are due to OTRS as a third-party solution that is open source. As a free resource, this repository allowed for sufficient help desk support for the project’s need. As a workaround, the integrity is
programmatically enforced—the greatest risk occurs when new features are added or defects are corrected.

OTRS is known to be a security risk and historically has had troubles with updates. Due to these two problems there are plans to remove this application in the future and merging a different ticket tracking software into the application. Until that time comes the application has been slowly phasing out the OTRS tables where possible as updates are made.

![Figure 7: OTRS Database Diagram](image)

4.1.3 Registry4

The Registry4 database is a database which was created for SDSFIE v4.0 Gold. The Registry4 database contains all of the tables and relationships necessary to display, build, validate and migrate the SDSFIE adaptations. The Registry4 database schema aligns closely with the National System for Geospatial Intelligence (NSG) Application Schema (NAS) platform independent model so that the information in the registries could be shared with a smaller amount of effort. They are not identical because we have requirements that the NAS model does not satisfy.
Figure 8: Registry4 Configuration Table and Relationships
Figure 9: Registry4 Model Element Table and Relationships
4.1.4 Application
Support tables for the web application and common tables are displayed in Figure 10 (below). Objects are designed to be unconstrained by unique keys and foreign keys. This design allows for duplicates and orphaned records. All behavior of the CMP tables must be inferred from usage, as these tables are used to interface with the OTRS software which provides limited documentation on the functionality.

- ServiceNames – Table which stores all of the components which can used for registering.
- UserToService – Table which related the user to the service chosen when registering.
- UserProfiles – Table which stores all of the information about a user.
- RoleInfo – Table which stores the user’s role information.
- CMPRepNotifications – Table which ties the component representative to change requests submitted by users within their component
- CMPNotificationEvent – Table tracking e-mails sent through OTRS to component representatives.
- CMPRep – Table which stores information about the component representative
- UserToCMP – Table relating the user to a component and service ID
- CMPServices – Table relating Service ID to the service Name
- CRSteps – Table created by configuring the software which tracks the CR process.

![Application Object Database Diagram](image-url)

Figure 10: Application Object Database Diagram
4.1.5 Backups

Database backups are created before and after every production push. The database backups are named with the name of the database followed by the date which they were taken. All database backups are stored in an Amazon Simple Storage Service (S3) bucket and on a shared drive locally at the Contractor location.

Microsoft Team Foundation Server (TFS) is used for source code control through the use of Git. TFS and the Git source control packages are installed on a local server at the Contractor location. The TFS server is backed up on a nightly basis and the backup is stored on an external drive located at Contractor location.

AWS backups are run twice weekly on Sunday and Wednesday evenings for the Production servers on AWS. These database backups are stored on AWS in the S3 buckets.

The Contractor is responsible for all backups.

4.2 Application Program Interfaces

Note: Across all APIs – in general – interfaces are implemented in order to support and encourage a baseline which can then be reused in other parts of the application.

4.2.1 SdsfieDataAccess

This interface contains type definitions and implementations for data access, and was implemented utilizing the repository pattern. The ‘Models’ namespace contains Data Transfer Objects (DTOs) which are models used to transfer data between the web application and database. No class behavior is modeled in these types. Rather, they are solely designed to support data movement between system tiers. In most cases, the concrete types are annotated to automatically utilize .NET serialization for movement from the web services tier to the web client tier. Recordsets are generally represented as .NET generic collections of these types. The ‘Repository’ namespace contains types whose responsibility is invocation of stored procedures in the data tier.

4.2.2 OnlineLib

This interface contains type definitions and implementations for web page base types, web services, business tier objects, and other types aggregated here for cohesion. This is legacy code which is slowly being phased out as different parts of the web application are being updated to implement a full MVC application, and some outdated subsystems are removed/replaced.

4.3 User Interface Design

UI design in general is described in the combined system Software Requirements Specification (SRS). In most cases, UI elements related to a specific feature are tailored to the workflow related to that feature (e.g. data grids where tabular data is manipulated). The common style elements (e.g. fonts, colors) are encapsulated in CSS files. Largely, the default themes for ExtJS controls (e.g. tabs, combo boxes, grids…) are utilized out-of-the-box.

Any deviations that occur are due to the catalog of an API being finite/limited; in order to meet client UI requirements, custom behavior and style were utilized. MVC, Autonomous UI, Repository, and Singleton patterns were implemented where efficiencies could be gained.
4.4 Interaction with Registry4

Significant interaction with the SDSFIE Registry4 database occurs in the Workflow tools under the Models and Workflows drop-down. These elements utilize the tables referenced in Figures 8 and 9 located in Section 4.1.3 Registry4.

4.5 System Performance

The SDSFIE Production instance is set up with the following specifications:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Type</td>
<td>M3.xlarge</td>
</tr>
<tr>
<td>Memory</td>
<td>15 GB RAM</td>
</tr>
<tr>
<td>Storage</td>
<td>60 GB Primary, 500 GB Secondary</td>
</tr>
<tr>
<td>CPU</td>
<td>Intex Xeon Quad Core 2.5 GHz</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows Server 2008 R2 x64</td>
</tr>
</tbody>
</table>

The OTRS Production instance is set up with the following specifications:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance Type</td>
<td>T2.medium</td>
</tr>
<tr>
<td>Memory</td>
<td>4 GB RAM</td>
</tr>
<tr>
<td>Storage</td>
<td>60 GB Primary</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel Xeon Dual Core 2.5 GHz</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows Server 2008 R2 x64</td>
</tr>
</tbody>
</table>

Amazon AWS guarantees 99.95% uptime for EC2 instances.

5 CLOUD IMPLEMENTATION

5.1 Launching an Amazon Instance

An Amazon EC2 Windows instance starts with the launch of the instance. Next, it briefly goes into the pending state while registration takes place. Then it moves to the running state, where instances can be rebooted, stopped, and then re-started. The Windows instance remains active until you initiate a shutdown process that terminates the instance. You can create an image of your instance and launch additional instances while your Amazon EC2 Windows instance is in the running state. This state transition process can be seen in Figure 4 above. This feature allows you to scale your infrastructure on demand. (Note: After an Amazon EC2 Windows instance is terminated, its infrastructure is no longer available. To continue working with the same infrastructure, a new instance must be launched.)
SDSFIE currently has two production instances running in AWS. One production instance (SDSFIE Production Server V3 FY2015) contains the SDSFIE web application and related databases while the second production instance runs the OTRS application (OTRS Server FY16). Both of these instances have been cloned to set up a User Acceptance Testing (UAT) environment. A list of these instances can be seen below.

![Figure 11: Amazon Instances in the UI](image)

The following tables provide system information about the four AWS instances which are used for the production and UAT environments.

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>SDSFIE Production Server V3 FY2015</td>
</tr>
<tr>
<td>Machine Instance ID</td>
<td>i-9fda8160</td>
</tr>
<tr>
<td>Security Group</td>
<td>Production SDSFIE</td>
</tr>
<tr>
<td>Region</td>
<td>us-east-1a</td>
</tr>
<tr>
<td>Public DNS</td>
<td>ec2-52-7-94-169.compute-1.amazonaws.com</td>
</tr>
<tr>
<td>External IP Address</td>
<td>52.7.94.169</td>
</tr>
<tr>
<td>Internal IP Address</td>
<td>172.30.0.9</td>
</tr>
</tbody>
</table>

![Figure 12: SDSFIE Production Web Server Details](image)

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>SDSFIE UAT Server FY2016</td>
</tr>
<tr>
<td>Machine Instance ID</td>
<td>i-3b6c648a</td>
</tr>
<tr>
<td>Security Group</td>
<td>Production SDSFIE</td>
</tr>
<tr>
<td>Region</td>
<td>us-east-1a</td>
</tr>
<tr>
<td>Public DNS</td>
<td>ec2-52-5-172-144.compute-1.amazonaws.com</td>
</tr>
<tr>
<td>External IP Address</td>
<td>52.5.172.144</td>
</tr>
<tr>
<td>Internal IP Address</td>
<td>172.30.0.28</td>
</tr>
<tr>
<td>Key</td>
<td>Value</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Name</td>
<td>OTRSI Server FY2016</td>
</tr>
<tr>
<td>Machine Instance ID</td>
<td>i-1ec79caf</td>
</tr>
<tr>
<td>Security Group</td>
<td>Production SDSFIE</td>
</tr>
<tr>
<td>Region</td>
<td>us-east-1a</td>
</tr>
<tr>
<td>Public DNS</td>
<td>ec2-52-21-188-168.compute-1.amazonaws.com</td>
</tr>
<tr>
<td>External IP Address</td>
<td>52.21.188.168</td>
</tr>
<tr>
<td>Internal IP Address</td>
<td>172.30.0.174</td>
</tr>
</tbody>
</table>

**Figure 14: SDSFIE Production OTRS Server**

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>OTRSI Server</td>
</tr>
<tr>
<td>Machine Instance ID</td>
<td>i-c200c2ef</td>
</tr>
<tr>
<td>Security Group</td>
<td>TestGroup22</td>
</tr>
<tr>
<td>Region</td>
<td>us-east-1a</td>
</tr>
<tr>
<td>Public DNS</td>
<td>ec2-184-72-249-96.compute-1.amazonaws.com</td>
</tr>
<tr>
<td>External IP Address</td>
<td>184.72.249.96</td>
</tr>
<tr>
<td>Internal IP Address</td>
<td>10.154.15.59</td>
</tr>
</tbody>
</table>

**Figure 15: SDSFIE Staging OTRS Server**

### 5.2 Required Ports & Protocols, Web Servers
- RDP 3389 (open)
- HTTP 80 (open)
- HTTPS 443 (open)
- TCP/IP to MSSQL 1433
- HTTP 8888 (open)

### 5.3 Required Ports & Protocols, Database Server
- RDP 3389 (open)
- TCP/IP to MSSQL 1433

### 5.4 Disaster Recovery
Database backups are created before and after every production push. The database backups are named with the name of the database followed by the date which they were taken. All database backups are stored in an Amazon S3 bucket and on a shared drive locally at the Contractor location.

Microsoft Team Foundation Server (TFS) is used for source code control through the use of Git. TFS and the Git source control packages are installed on a local server at the Contractor location. The TFS server is backed up on a nightly basis and the backup is stored on an external drive located at the Contractor location.
AWS backups are run twice weekly on Sunday and Wednesday evenings for the Production servers on AWS. These database backups are stored on AWS in the S3 buckets.

In case of catastrophic failure there will be 1 work day of downtime while the environments and web application are set back up.

6 INSTALLING TOOLS
Amazon Web Services provides a web interface for configuring/monitoring the instances assigned to the account but Amazon also provides a command line interface (CLI) tool which can be used to interact with these instances as well. Provided below are the steps to set up the command line interface.

6.1 Installing Amazon Elastic Compute Cloud (EC2) Tools

1. Download Security Certificates
   b. Log into your primary EC2 account
   c. Select “Rotate your access keys”
   d. Select “Manage User Access Keys”
   e. Select the “Security Credentials” tab
   f. Select “Create Access Key”
   g. Select “Download Credentials”

2. Install Required Software

c. Download EC2 CLI Tools and put in C:\Program Files (x86)\ec2-cli-api folder

3. Add System Variables

a. Right Click on Computer and select Properties

b. Click Advanced system settings and then click the Environmental Variables button

Figure 17: My Computer->Properties

b. Click Advanced system settings and then click the Environmental Variables button

Figure 18: System Properties Advanced Settings
c. Add Python to Path System Variable

d. Add EC2_HOME System Variable and point at EC2 CLI Tools

e. Add JAVA_HOME System Variable and point at Java install location

7 Architectural Needs

While working with the system architecture it has been decided that the OTRS application is not a suitable tracking solution for the web application. OTRS has a large number of security risks and the version which is integrated with SDSFIE is several years old and outdated. The OTRS application cannot be updated without major changes to the integration portion. Due to these two reasons removing the OTRS application and replacing it with a more suitable ticket tracking solution is recommended.
Appendix 1: List of Acronyms

API  Application Program Interface
ASP  Active Server Pages
ATC  Authority To Connect
AWS  Amazon Web Services
CLI  Command Line Interface
CMP  Change Management Process
CRUD  Create, Read, Update, Delete
CSS  Cascading Style Sheets
DNS  Domain Name System
DoD  Department of Defense
DTO  Data Transfer Objects
EC2  Elastic Compute Cloud (Amazon)
ELMAH  Error Logging Modules and Handlers
FedRAMP  Federal Risk and Authorization Management Program
HTTP  Hypertext Transfer Protocol
HTTPS  HTTP Secure (over TLS or SSL)
IAM  Identity and Access Management
IGG  IGI&S Governance Group
IGI&S  Installation Geospatial Information and Services
IP  Internet Protocol
IT  Information Technology
JS  JavaScript
JSON  JavaScript Object Notation
MSSQL  Microsoft SQL
MVC  Model–View–Controller
NAS  National System for Geospatial Intelligence (NSG) Application Schema
NSG  National System for Geospatial Intelligence
OTRS  Open Technology Real Services
RBAC  Role-Based Access Control
RDP  Remote Desktop Protocol
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>S3</td>
<td>Simple Storage Service</td>
</tr>
<tr>
<td>SDSFIE</td>
<td>Spatial Data Standards for Facilities, Infrastructure, and Environment</td>
</tr>
<tr>
<td>SES</td>
<td>Simple E-mail Service</td>
</tr>
<tr>
<td>SMTP</td>
<td>Simple Mail Transfer Protocol</td>
</tr>
<tr>
<td>SOW</td>
<td>Scope of Work</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirements Specification</td>
</tr>
<tr>
<td>SSL/TLS</td>
<td>Secure Sockets Layer / Transport Layer Security</td>
</tr>
<tr>
<td>STIG</td>
<td>Security Technical Implementation Guide</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TFS</td>
<td>Team Foundation Server (Microsoft)</td>
</tr>
<tr>
<td>UAT</td>
<td>User Acceptance Testing</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>