## **Forest Service e-Permitting Application Protocol Interface**

## **Statement of Objectives**

## **1.0 BACKGROUND**

The U.S. Forest Service is engaged in an ongoing effort to modernize and simplify their permitting processes. One facet of this effort is “ePermitting”, or making the applications for many Forest Service permits available online.

In the future, when applications for Forest Service permits are submitted online, they should pre-populate the related Forest Service database, the Special Use Data System (SUDS), a component of the National Resource Management system. This task order is for an Application Protocol Interface (API) layer future applications could use to read and write to SUDS, an Oracle system.

## **2.0 PERFORMANCE OBJECTIVES**

The needs of this task order are provided in user story format to facilitate agile development. The user stories provided below shall be considered as the objectives to be satisfied for successful performance of this task order. These user stories may be refined by later sprints in order to continuously meet user needs.

**User stories for API layer**

* As an API user, I would like to both be able to send and retrieve information from SUDS.
* As an API user, I would like to write new records, and modify existing database records in the SUDS system programmatically, if I have the proper permissions.
* As a system administrator, I would like to restrict which applications have the access to reading and writing to the SUDS database restricted.
* As an API user, I would like the data to be retrievable in a specification that is consistent with the [18F API standards](https://github.com/18F/api-standards).
* As a developer of front-end an application, I would like the API to be documented so that I can anticipate API endpoints for my requests in t[he Open API Specification](https://github.com/OAI/OpenAPI-Specification/tree/OpenAPI.next).
* As a system owner, I would like to submit field mappings in a static data format, so that I can adapt the API to work for additional permit types.

**Application Constraints:**

* The API will be a representational state transfer (REST) web service.
* The API will need to run on a Linux Operating System, Apache HTTP server, MySQL database, and use either Python, Ruby, PHP, or NodeJS to integrate smoothly with the future production environment that will be hosted within the [U.S. Department of Agriculture National Information Technology Center](https://www.ocio.usda.gov/about-ocio/data-center-operations/nitc-cloud-services).
* The API will need to serve data from the SUD system, an Oracle database system
* Access to the actual database will not be provided during development, but a clone of the database will be provisioned for development.
* Personal Identity Information (PII) shall be secure and kept through encryption from exposure to unauthorized users from uploaded documents.

**Additional requirements:**

* All software code delivered under this order shall comply with the [18F open source policy](http://https/github.com/18F/open-source-policy/) in effect as of the date of award.
* All software code delivered under this order shall comply with the [18F API standards](https://github.com/18F/api-standards) in effect as of the date of award.
* As part of this being purchased off of the Agile Blanket Purchase Agreement (aBPA), work will be conducted in one or two-week sprints and reviewed at the end of each sprint for acceptability per the Quality Assurance Surveillance Plan (QASP).

## **3.0 SCOPE**

This task order focuses on the creation of an API to encapsulate SUDS in order to enable rapid and modern development of other ePermitting tools. The high-level goals of the task order are the following:

* Create a RESTful API using JSON standard open protocol.
* Create the API to read/write to the existing database (for existing data fields and tables) and also read/write to a new database (for new data fields and tables).

#### **3.1 List of Deliverables**

**| DELIVERABLES | DUE DATES | CONTENT DESCRIPTION |**  
| Code Repository of Product | End of task order | Version-controlled Open Source repository of code that comprises prototype that will remain in the public domain |

| Design Deliverables | End of every applicable sprint | Mock ups and/or design files if applicable, or design changes reflected in the Development Prototype |

| Development Prototype | End of second sprint, and every sprint thereafter | In-progress development prototype, accessible on the web via staging server / development server |

| Transition plan | 3 business days after the conclusion of the second-to-last sprint | Provided in QASP |