Whole System

**Functional Requirements**

1. The system shall receive data from an arbitrary number of external sources.
2. The system shall label the classification level (i.e. unclassified, secret, and top secret) for any data received.
3. They shall store this data in a database.
4. The system shall be interfaced via an API endpoint for web applications.
5. The system shall authenticate the user every time they access an individual API.
6. The system shall execute or deny requests based on the user’s security context.
7. ~~The system shall enforce Multilevel Security (MLS) policy.~~
8. Users shall be able to read data, write data, and view reports of data which they have access to.

**Non-Functional Requirements**

1. The installation procedure must work in an offline/disconnected environment.

**Constraints**

1. APIs shall be written in Python since Python has several libraries which can interface with postgres. This has been decided by the team and approved by Skyward Federal.
2. Unit testing shall be done in Python’s native unit testing suite, unittest
3. System testing suite is TBD
4. Skyward Federal has provided several AWS services for the API gateway and serving endpoints.
5. The final configuration must run on CentOS 7 with SELinux enabled (and enforcing)

Data Storage

**Functional Requirements**

1. Data elements shall be stored in the database alongside their SELinux labels
2. The user shall only be allowed to access data they have the privilege to access and shall be restricted from accessing any other data.
3. Users shall be able to submit data for storage after they label it.
4. There shall be one account for each service that needs access to the data

**Constraints**

1. The database will be configured and run as an SE-PostgreSQL database since Skyward Federal has suggested we use a postgres database with SELinux labels.
2. SE-Postgres only provides data protection at the column and table levels.

Container Runtime

**Functional Requirements**

1. The container shall provide compute resources for RESTful services behind the API Gateway.
2. *~~The RESTful services shall be able to authenticate users using the IdAM Component.~~*
3. This component shall start up a container with the enforced security context of the requested service.
4. ~~This component shall ensure that only authorized users can access each container.~~
5. The service running inside the container must gracefully handle ‘Access Denied’ errors when attempting to retrieve data from the Data Storage component.
6. This container shall be able to be activated and deactivated at arbitrary times to serve data to users with the proper security context.
7. This component shall ensure multiple instances of RESTful services running inside each container cannot share information with each other.

**Non-Functional Requirements**

1. It shall ensure container startup time is as fast as possible.
   1. The Docker images shall include all executables, libraries, and configuration data so that application startup is the only action required when the container starts.

**Constraints**

1. The system shall be run within a Docker container which has been provided by Skyward Federal.
   1. One Docker container for each running service for a consumer. The container that is running in the system depends on the user’s security levels and categories that are requesting the service.
2. This component shall be installed on the same CentOS 7 host that will be running the Data Storage Component (due to the dependency on SELinux).
3. Skyward Federal has dictated that this component shall allow connections from the API Gateway to the Docker Daemon on TCP Port 2376 with configured encrypted communication.
   1. The request shall set the “--security-opt label=…” option when starting the container in order enforce security.
4. Each RESTful service shall be given login credentials to connect to the SE-PostgreSQL database within the Data Storage component.
   1. The service shall be written to expect that it will have access to different information depending on the SELinux labels.
5. The RESTful services shall handle incoming requests on TCP Port 443.

Stretch Goals

**Functional Requirements**

1. All logs shall be forwarded to an external log aggregator.
2. If a failure of log forwarding occurs, an alert shall be sent where an operator will see it.
3. ~~The system shall encrypt the data based on its label.~~

**Non-Functional Requirements**

**Constraints**

Identity and Access Management (IdAM)

**Functional Requirements**

1. The IdAM shall facilitate isolation (encryption and access control) by authenticating users between security levels of:
   1. Data at rest
   2. Compute resources
   3. Data transiting through the API Gateway
2. The IdAM shall authenticate users via an external IdAM system (such as a Directory Server) or with internal user/group configuration
3. The IdAM shall retrieve Authorization information from an external IdAM system (such as a Directory Server)
4. The IdAM shall respond to queries about accesses for specific users
5. The IdAM shall handle login across multiple applications

**Non-Functional Requirements**

1. TBD

**Constraints**

1. The IdAM system shall be managed by Keycloak

Platform Controller

**Functional Requirements**

1. Authenticate users using KeyCloak
2. Single access point to
   1. Multiple RESTful services
   2. Secure data storage
3. Encrypt data between producers and consumers
4. Only authorized customers access certain services
5. Log flow of information

**Non-Functional Requirements**

1. All build dependencies must be mirrored locally.

**Constraints**

1. Use an OpenJDK distribution