**API Authentication Process**

As part of application programming interface (API) submissions into STARS, proprietary systems will be required to manage the authentication of API submissions.

Once ACL confirms the submission of a signed Interconnected Security Agreement (ISA), Booz Allen will contact the lead developer of each proprietary vendor. Separate emails will be sent containing the following information:

1. A username and password to the API test environment. A test account will be created for developers to log into the test environment and review API submissions.
2. A Client ID and Client Secret.

* As a part of the OAuth2 Authorization workflow, the lead developer for each proprietary system will be provided with a set of client credentials in the form of the Client ID and Client Secret. The credentials will be used in the authentication process to generate an access\_token to be attached as an authorization header in the record's API request.
* The access tokens are temporary passwords that last for three minutes and are included in the submission as a security measure.
* Tokens are overwritten once the subsequent one is generated, only one access\_token can be active at any time.

Below you will find information required for the path, headers, and body of the API request:

Token Request

1. The header to the token path (/auth/oauth/token?grant\_type=client\_credentials) is as follows:

* "Authorization" : "*Basic Base64Encoded*('*clientID*:*clientSecret*')"

1. The body of the request is as follows:

* "{grant\_type: client\_credentials}"
* Note: The request should return an access token if successful

Record Submission

1. The header to the record's path is as follows:

* "Authorization" : "Bearer *access\_token*"

1. The body of the request consists of the Json formatted record.

It is important to understand these necessary steps to successfully submit a record. The process for generating tokens and applying them to the request allows for a variety of possible implementations. Below are a few token management suggestions, however, it is possible to implement and optimize the process however any proprietary system’s development team sees fit.

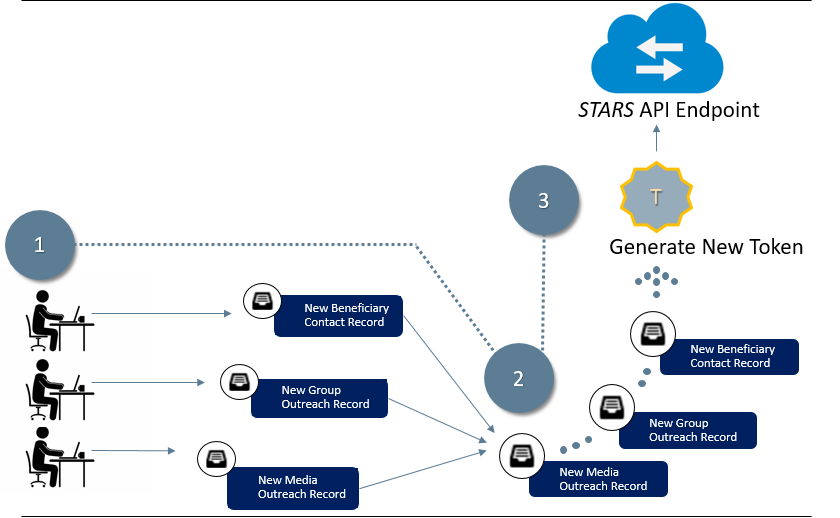
# API Token Management Alternative 1: API Record Queue

Suggested workflow in managing the token:

1. Proprietary System users log records into their system.
2. The records become aggregated into a single queue.
3. Each API submission request generates its own token.

Figure 1: Alternative 1, API Record Queue

1. 1 : 1 | Token/Request



# API Token Management Alternative 2: Token Auto-Generation

Suggested workflow in managing the token:

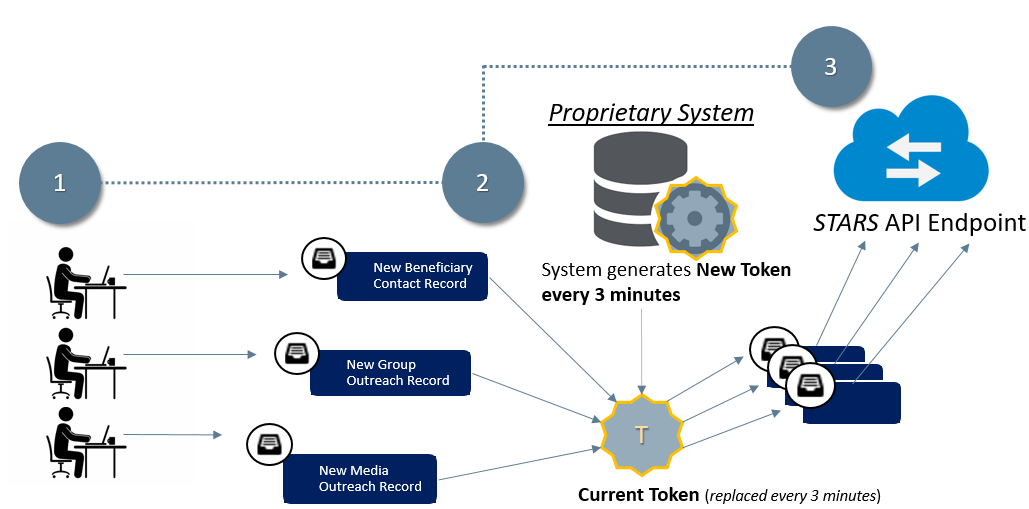
1. Proprietary System users log records into their system.
2. Proprietary System generates new token every 3 minutes.
3. As records are made, they grab the current token and API to STARS.
4. 1 : Many | Token/Requests

Figure 2: Alternative 2, API Record Queue