**TRSA Description and GUI Requirements**

The Odum designed Trusted Remote Storage Agent (TRSA) has been developed to afford a data provider the ability to keep data at their institution and still make it discoverable by remote users. To accomplish this goal the data providers remote storage is trusted by a Dataverse installation to ensure data is persistent and matches metadata indexed by that approved Dataverse for that data set. The Dataverse installation will contain all the metadata for the target data set as well as file level metadata generated at the remote TRSA for objects not physically transferred to Dataverse. The approved curators at the remote data providers site are responsible for maintaining the content and updating Dataverse relevant data set records. This is supported by MOUs between the organizations.

The original use case for the ImPACT project was that we needed a remote sensitive data storage location as required by some data providers where they are not allowed to move the data. At the same time, we needed to create file level metadata on quantitative data files for Dataverse to index for discovery. This workflow could also be used when the data is too large or impractical to transfer to Dataverse local storage.

**Functions of TRSA**

Allow user to create TRSA Profile

Allow user to identify which SAFE, Notary Service, and data access policies or DUAs to use

Allow user to select ZIP file containing local data files then generate and register metadata with Dataverse as new dataset.

Allow user to add single remote files metadata to an existing Dataverse dataset

Allow user to query Dataverse for list of datasets already submitted by TRSA

Allow user to connect to SAFE

Allow user to connect to Notary Service

**Guidelines to help keep light weight and lower maintenance**

Let Dataverse Server do as much as possible to reduce local code.

Assign DOIs on the Dataverse Server side

Track submissions using text-based logs rather than database

Use Dataverse APIs for metadata transfer

Only Metadata Transferred all local data files remains local

**User Preconditions**

User has permission from target Dataverse to be an authorized TRSA. I.e. MOU in place

User has already created target Dataverse

User has already created target Dataset and uploaded non-sensitive data using IMPACT metadata template manually entering metadata at dataset level.

User has generated and knows their API key for use by TRSA

User has already created SAFE and Notary Service policies

**Required Dataverse Core Additions**

Dataverse to track remote TRSA Sites that are authorized: Need new table to identify multiple potential TRSA’s approved. TRSA Name, TRSA Contact Info, TRSA required Notary Service etc...

Dataverse API that allows TRSA to update information above from TRSA GUI

Dataverse to track new required TRSA metadata for data access. TRSA Remote Files access URL, DUA locations in Safe and remote file type. Note: Possible leverage of DDI Distributor fields could help. Right now, all of this is URLs

Dataverse recognizes TRSA registered file objects and on download request forwards users to required Notary Service for Data Access Request.

**TRSA Function Descriptions**

***Create/Edit Local TRSA Profile Button Workflow***

*User Enters Below information in a form or edits existing form*

*Name of installation*

*Administration contact*

*Host Dataverse URL*

*Admin Dataverse API Access Key*

*Data storage location i.e. local file system path*

*Data access information*

*Approved Notary Service URL*

*Approved SAFE Service URL*

***Register Data with Dataverse for Indexing Button Workflow***

*User Browses to File or Directory*

*User Select File or Directory*

*User Selects begin local metadata creation*

*System Actions*

*Create Variable Level Metadata*

*Add TRSA Local Metadata*

*Add TRSA Data Location URL*

*User Reviews metadata package prior to submission and assigns Dataset DUA pointer/URL in NS/SAFE via form*

*User Approves remote Dataverse dataset ID or Create New Dataset*

*System Actions*

*Send JSON to Dataverse via API*

*Retrieves DOI Assigned by Dataverse*

*Creates Local Log entry of action using DOI as identifier*

*User Views Success Report of DOI and package assigned by Dataverse Server*

*Note: There are two variation on the above workflow that need to be addressed. The first is a brand-new data set creation. This will be the least used. The second the primary case where a user will create and add metadata for single file to existing Dataverse Dataset already created.*

***Query Dataverse Server for list of Registered Data Button Workflow***

*User selects show me my registered data*

*Systems uses Dataverse API to search and retrieve Dataset and File IDs associated with the local TRSA*

*User prints or saves as file results or User Performs Audit*

*If Audit Selected system compares results to local log files and displays the differences.*

***Connect to SAFE Button***

*Forward User to Profile SAFE*

***Connect to Notary Service Button***

*Forward User to Profile Notary Service*