Office of Strategic National Alien Planning

Logistical Operations Service Tracker (LOST)

Product Requirements and Design Document

January 16, 2017

Executive Summary

OSNAP has a significant number of declassified and classified assets that must be accounted for and audited in accordance with federal laws and regulations. The current practice of directorate level spreadsheets and databases has resulted in failed audits and assets silently going missing while in transit. The Logistical Operations Service Tracker (LOST) will unify asset tracking and management across all OSNAP divisions, reducing incidents of property loss and asset management overhead.

LOST provides a single point of truth for where OSNAP assets are located and associated ownership. Assets will be tracked at rest and periodically while in motion. Due to the classified nature of some OSNAP assets, LOST will implement access controls to limit data visibility based on the nature of the data and the requesting user.

Document Versioning

12/28/2016 DE - Initial version

01/09/2017 DE - Updates for prerelease to development team

01/16/2017 DE - Initial datamodel added

Project Description

Asset management across a large geographically distributed organization presents a significant management challenge. In the past, individual divisions have been independently responsible for asset management and tracking. Independent asset tracking will no longer scale for the needs of OSNAP and thus a large scale centralized asset management system is required. The Logistical Operations Service Tracker (LOST) will fill this requirement and enable OSNAP to safely continue to scale out.

OSNAP personnel needing access to LOST will have LOST access added to their WHO account. WHO will provide LOST with user information including username, person name, and division. LOST will internally maintain other user information as needed to support LOST functions.

Many OSNAP assets are classified and visibility into the type and location of those assets must be tightly controlled. LOST will support mandatory access control (MAC) to protect asset information (e.g. asset type and location). Since many assets must be hidden in plane sight (e.g. shipments on public highways), plausible substitute information in some cases will be shown to users with insufficient clearance. The special user role of 'classifier' will be able to change the classification of assets within LOST. The special user role of 'assigner' will be able to change the clearance of users. LOST must enforce that no user is both a classifier and an assigner.

Assets are instances of products. Product information can be used to look up plausible substitutions for an asset within a report. An asset may also include an explicit substitution. Each asset carries a classification, since an otherwise declassified product (e.g. a note pad) may become classified based on how it is used. Products also carry a classification and an asset may not carry a classification that is incompatible with the backing product.

Assets will be associated with a facility while at rest. While in transit, an asset will be associated with a travel request. The travel request will include the starting, ending, and last known location of the asset as well as information regarding the convoy. In addition to the current asset location, the location history of an asset can be reported using LOST.

Existing OSNAP asset data exists in a plethora of division specific data sources. The existing asset data will need to be in LOST at the time each division is cut over. Operators will use data migration scripts to support this activity.

Workflows

This section describes several of the workflows LOST will help to automate.

Travel Management

LOST will provide visibility into when and where assets are moved through the Travel Management workflow. This workflow will enforce OSNAP asset management policy.

Workflow Steps

1. A user in logistics creates a travel request and includes the assets to be transfered, the request is in the 'unapproved' state.

2. The request is then forwarded to management for approval. All managers must approve for the travel request to move forward, the request moves to the 'approved' state. Logistics and facilities managers may not approve their own requests.

   (a) Logistics manager at the source

   (b) Logistics manager at the destination

   (c) Facilities manager at the source

   (d) Facilities manager at the destination

3. Logistics manager at the source facility adds the convoy transporting the assets to the travel request. The asset is now associated with both the facility and the convoy. The request is now in the 'ready' state.

4. After loading onto the convoy, a user in logistics updates the request state to 'in transit' and the asset is disassociated from the source facility and associated with the destination facility.

5. While in transit, logistics users associated with the convoy periodically update the request with current location and other notes.

6. When the asset is received at the destination facility, a facilities manager associated with the destination facility moves the travel request into the 'received' state. The asset is disassociated with the convoy.

Asset Location History

LOST will provide features to reconstruct a time ordered location history of an asset.

Workflow Steps

1. A Logistics Manager or Facility Manager selects an asset and make a request for asset history.
2. The report request is logged for auditing.
3. Logistics Manager approves the request. Logistics managers may not approve their own requests.
4. The approved request is logged for auditing.
5. The requesting logistics or facilities manager opens the report in LOST. Openning the report is logged.

Additional Notes

An asset history report may only be viewed once. If the request approval causes LOST to generate and store the report internally, the report must be deleted once opened.

A logistics and facilities managers should only be able to request history for assets currently located at facilities they are associated with.

Logistics and facilities managers should only be able to see the facilities they associated with. Facilities that a logistics or facilities manager is not associated with should be omitted in the report. For example:
An asset starts at A, is moved to B, and ends at C, $A \to B \to C$. A Facilities manager associated only with facilities A and C would get a report indicating $A \to C$. Similarly, a facilities manager associated only with facilities B and C would get a report indicating $B \to C$. A facilities manager not associated with C should not be able to see the asset to request the report on.

Asset Inventory Report

LOST will provide features to report on currently housed assets.

Workflow Steps

1. A Logistics Manager or Facility Manager selects a facility and requests a report on assets currently located there.
2. The report request is logged for auditing.
3. Facilities Manager approves the request. Facilities managers may not approve their own requests.
4. The approved request is logged for auditing.
5. The requesting logistics or facilities manager opens the report in LOST. Openning the report is logged.

Additional Notes

An asset inventory report may only be viewed once. If the request approval causes LOST to generate and store the report internally, the report must be deleted once opened.

Logistics and facilities managers should only be able to request asset inventories for facilities they are associated with.

Logistics and facilities managers should only be able to see assets for which they have appropriate privileges. Assets the user does not have sufficient privilege for should be omitted if a substitution has not been given for the asset. For example: An asset I is in security compartments A_{ts}, B_s, and C_u. A user with A_{ts}, B_s, and C_u will see the asset in the report since they have the correct compartments and sufficient clearance. A user with A_{ts}, B_{ts}, and C_{ts} will see the asset in the report since they have the correct compartments and sufficient clearance. A user with compartments A_s, B_s, and C_s should not see the asset I since A_{ts} exceeds their clearance. A user with compartments A_{ts}, B_{ts}, and D_{ts} will not be able to see the asset since they are missing a required compartment.

User Stories

This section describes several of the user stories that are expected to be covered by the Logistical Operations Service Tracker (Lost). User stories are ordered by user type and high level task. Each user story provides a high level user objective and possible user interface process to complete the task.

Classifier

Change Asset Compartments

Objective

Classifier is changing the compartments associated with an asset.

Experience

Classifier navigates to the asset using the LOST UI. After selecting the asset, Classifier chooses to edit the asset properties. Classifier is able to change (add and remove) compartments the asset is associated with. Classifier is prompted to save changes before they take effect.

Change Product Compartments

Objective

Classifier is changing the compartments associated with a product.

Experience

Classifier navigates to the product using the LOST UI. After selecting the product, Classifier chooses to edit the product properties. Classifier is able to change (add and remove) compartments the product is associated with. If the Classifier assigns compartments that are incompatibly with an existing asset, a warning is displayed. Classifier is prompted to save changes before they take effect.

Assigner

Change User Compartments

Objective

Assigner is changing the compartments a user has access to.

Experience

Assigner navigates to the user using the LOST UI. After selecting the user, the assigner is able to change (add and remove) compartments the user is associated with. Assigner is prompted to save changes before they take effect.

Change User Facilities

Objective

Assigner is changing the facilities a user has access to.

Experience

Assigner navigates to the user using the LOST UI. After selecting the user, the assigner is able to change facilities and vehicles the user is associated with. Assigner is prompted to save changes before they take effect.

Logistics User

View Travel Requests

Objective

Logistics User wants to view travel requests needing service.

Experience

Logistics logistics dashboard lists the travel requests this user might be able to assist with. The user selects a travel request and is shown the details. User updates the request.

Request Transfer

Objective

Logistics User is moving an asset.

Experience

Logistics User selects 'new travel request' from the logistics dashboard. The user selects the source facility and assets to be transported then saves the request.

Start Transfer

Objective

Logistics User is noting that at asset has started motion.

Experience

After openning the travel request, the Logistics user changes the travel request state to loaded.

Update Transfer

Objective

Logistics User is noting an intermediate change in asset location.

Experience

After openning the travel request, the Logistics user updates the location to note that the asset has moved to a particular intermediate location.

Receive Transfer

Objective

Logistics User is noting arrival of an asset.

Experience

After openning the travel request, the Logistics user clicks 'transfer complete' to note that the asset has been received.

Logistics Manager

LM Approve Travel Requests

Objective

Logistics approves a new travel requests.

Experience

Logistics dashboard lists the new travel requests. The user selects a travel request and is shown the details. User selects approve. User then uses the UI to assemble the convoy to move the asset.

Facility Manager

FM Approve Travel Requests

Objective

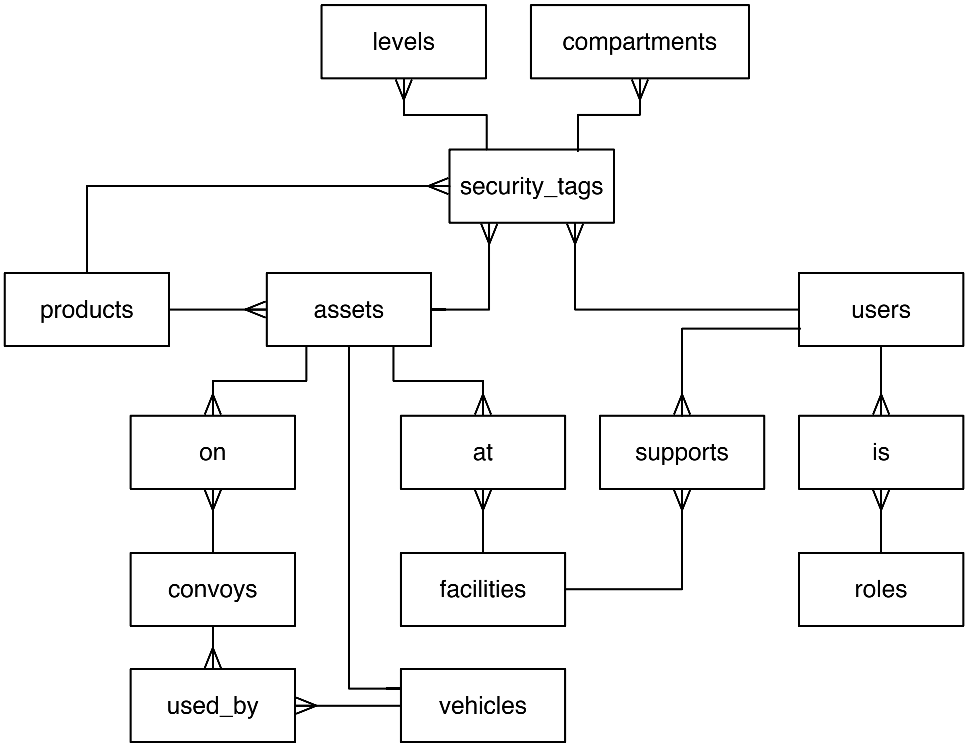
Facilities approves a new travel requests.

Experience

Facilities dashboard lists the new travel requests. The user selects a travel request and is shown the details. User selects approve.

Data Model

Overview



LOST concerns itself with three main ideas: assets, users, and security tags. *Assets* are the entities that LOST has been designed to track. *Users* represent OSNAP staff that are involved in managing assets. *Security tags* are used to provide mandatory access control since there are limitations on which security compartments users are allowed to view or manage.

Several join tables appear in the model. These join tables are needed to handle the many-to-many relation ships between entities.

Asset Tables

products

product_pk	integer	primary key for a product instance
vendor	text	who sells this product
description	text	description of the asset
alt_description	text	alternate description for the asset

assets

asset_pk	integer	primary key for an asset instance
product_fk	integer	id for the product instance the asset was spawned from
asset_tag	text	stick or engraved id used for inventory tracking
description	text	description of the asset
alt_description	text	alternate description for the asset

vehicles

vehicle_pk	integer	primary key for an asset instance
asset_fk	integer	id for the associated asset record

Vehicles are a type of asset but are special since they can be used to transport other assets. Table inheritance in Postgres could be used to do this a little more gracefully but that is an out of scope feature.

facilities

facility_pk	integer	primary key for a facility instance
fcode	text	facilities code used to identify the facility (6 or less characters)
common_name	text	common name for the facility
location	text	addressing information for the facility

asset_at

asset_fk	integer	asset at a facility
facility_fk	integer	facility the asset is at
arrive_dt	timestamp	when the asset arrived
depart_dt	timestamp	when the asset left

convoys

convoy_pk	integer	primary key for a convoy instance
request	text	request identifier for the convoy
source_fk	integer	source facility
dest_fk	integer	destination facility
depart_dt	timestamp	when the asset departed
arrive_dt	timestamp	when the asset arrived

used_by

vehicle_fk	integer	vehicle participating in a convoy
convoy_fk	integer	convoy vehicle participates in

asset_on

asset_fk	integer	asset at a facility
convoy_fk	integer	convoy the asset is on
load_dt	timestamp	when the asset was loaded
unload_dt	timestamp	when the asset was unloaded

User Tables

users

user_pk	integer	primary key for a user instance
username	text	login name used by the user
active	boolean	Is the user active?

roles

role_pk	integer	primary key for a role instance
title	text	short textual name for the role

user_is

user_fk	integer	id for the user instance
role_fk	integer	id for the role instance

user_supports

user_fk	integer	id for the user instance
facility_fk	integer	id for the facility instance

Security Tables

levels

level_pk	integer	primary key for security level lookups
abbrv	text	abbreviation for the security level
comment	text	comment, if any

compartments

compartment_pk	integer	primary key for compartment lookups
abbrv	text	abbreviation for the security compartment
comment	text	comment, if any

security_tags

tag_pk	integer	primary key for security tag instance
level_fk	integer	id for the tag level
compartment_fk	integer	id for the tag compartment
user_fk	integer	user the tag is applied to or NULL
product_fk	integer	product the tag is applied to or NULL
asset_fk	integer	asset the tag is applied to or NULL

Security tags must have both level and compartment. Security tags must also have a user xor product xor asset.

API Specification

This section describes the API used to interact with LOST.

Overview

LOST will provide a RESTful interface for automated interaction with other OSNAP systems. Human resources systems are expected to access and modify LOST user account information. The procurements department will also integrate with LOST to add and change the classification of various products. Only specific application instances should be allowed to use the REST interface to make changes to LOST data.

API Calls

activate_user

Reactivates LOST access for a user or generates a new user account if needed.

Request:

username OSNAP username to activate

Response:

result ‘OK’, ‘NEW’, or ‘FAIL’

suspend_user

Revokes access for a user. If the user does not exist or access has already been revoked, this call has no effect.

Request:

username OSNAP username to revoke

Response:

result ‘OK’

list_products

Requests a listing of all products in LOST based on a filter criteria.

Request:

filter json listing filter criteria

Response:

result json containing the product information

add_products

Adds products to LOST. Requests should be atomic (all products are added or no products are added) and attempting to add a duplicate product should cause the request to fail.

Request:

new_products json describing the new products

Response:

result 'OK' or 'FAIL'

add_asset

Adds a new asset to LOST.

Request:

product product type of the asset

compartments additional compartments the asset should be classified under

facility facility the asset is initially housed at

Response:

result 'OK' or 'FAIL'

Mockups

To Do

Tech Stack

The Logistical Operations Service Tracker (LOST) product will use OSNAP's standard web application technology stack. Deviations from the standard technology stack are not expected to support LOST. Exceptions must be approved by the OSNAP Chief Information Security Officer (CISO) prior to deployment.

Standard Technologies

Apache httpd The Apache http daemon will be used to host the web application.

mod_wsgi mod_wsgi will be used as the gateway between Apache and the application.

Python Python 3 will be used as the application development language.

PyCryptodom Provides cryptographic services for Python

Flask The Flask framework will support development efforts.

WHO User authentication will be done using the OSNAP WHO service.

Postgres The Postgres RDBMs will be used for persistent storage.

PGSQL If needed, stored procedures will be written using the default procedure language PGSQL.

Software Design

To Do