

Cisco Partner Support Service



Smart APIs Leading Practice Guide

June 2015

Contents

Purpose.....	3
Introduction	3
Overview	4
Different Ways to Get and Store Data	5
Leading Practices When Calling and Using APIs.....	6
Usage: Inventory Extended Details API	7
Usage: Inventory (Paginated) APIs	8
Usage: Inventory (Non Paginated) API	9
Usage: Field Notice Alert APIs	11
Usage: PSIRT Alert APIs.....	12
Usage: IntelliShield Alert APIs.....	13
Usage: Hardware Eox APIs.....	14
Usage: Software Eox APIs	15
Usage: Not Processed Device Report API	16
Usage: Last Updated API	17
Do's and Dont's.....	18
Handling Error Conditions.....	19
1. Not Authorized	19
2. Request timed out.....	19
3. EXCEED-MAX-LIMIT-101	19
Appendix A: References and Support Resources	20
References	20
PSS Support Community.....	20
Smart Services Bureau.....	20

Purpose

This document provides a high level overview on how and when to use PSS API services for Cisco Partners and their application developers.

Note: Refer to the [PSS API Developer Guide](#) for comprehensive details on using the PSS APIs to develop custom applications.

Introduction

The PSS Smart APIs provide you with installed base management and alert reporting information comparable to what you would access via the PSS Smart Portal. PSS APIs can be used to construct your own custom applications.

APIs are designed as an alternative way to access some of the information that is available through the PSS Smart Portal. All data retrieved by Smart APIs is transmitted from Cisco to the partner via the SSL protocol, using an HTTPS browser connection. Only Simple Object Access Protocol (SOAP) is supported in this release.

Nine PSS API services provide you the ability to obtain information from a variety of areas such as inventory, contract, and alert information. The nine PSS API services are:

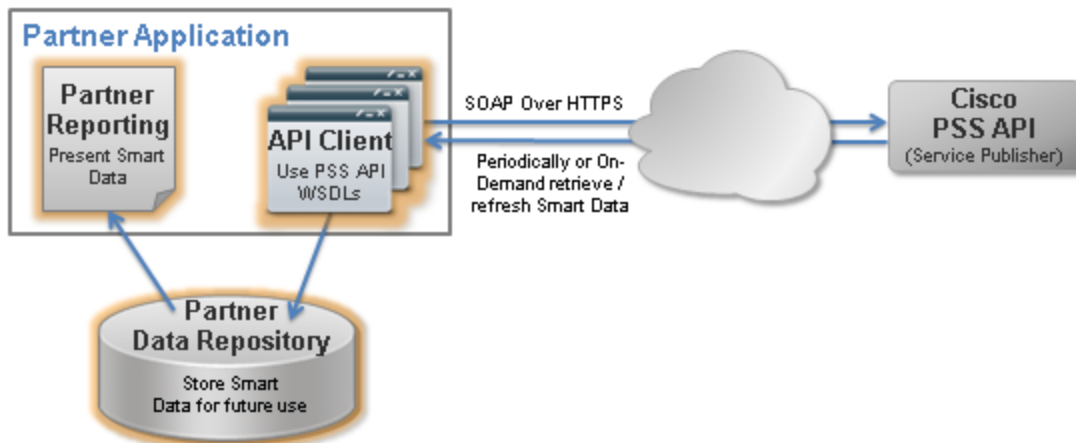
Service	Description	API Service Calls
Inventory	Pull inventory information for a customer's network including the physical inventory at a chassis level and module level	<ul style="list-style-type: none"> • <code>getCustomerInventoryID</code> • <code>getCustomerExtendedInventoryDetails</code> • <code>getCustomerInventoryPaginatedDetails</code> • <code>getCustomerInventoryDetails</code>
<p>Note: Input parameters (customer ID, inventory ID, and device ID) required for the following services are returned in the API Inventory service calls described above.</p>		
Contract	API to pull contract information for the devices on a customer's network. Entitlement checks are performed to make sure sensitive information is not returned if you are not eligible to access the customer information.	<ul style="list-style-type: none"> • <code>getContractCoverageDetails</code>
Field Notice	Get Field Notice alerts for a specified customer, inventory and devices or fetch information related to a selected Field Notice alert	<ul style="list-style-type: none"> • <code>GetFN</code> • <code>GetFNDetails</code>
PSIRT	Fetch details for a specific PSIRT alert or get all PSIRT alerts for specified customer, inventory and devices.	<ul style="list-style-type: none"> • <code>getPSIRT</code> • <code>getPSIRTDetails</code>
IntelliShield	Get all IntelliShield alerts for specified customer, inventory and devices or fetch information related to a selected IntelliShield alert	<ul style="list-style-type: none"> • <code>getIS</code> • <code>getISDetails</code>
Hardware EoX	Pull hardware end-of-life details associated with your customer's collected data.	<ul style="list-style-type: none"> • <code>getHwEox</code> • <code>getHwEoxBulletin</code> • <code>getHwEoxMigrationPIDs</code>
Software EoX	Pull software end-of-life details associated with your customer's collected data	<ul style="list-style-type: none"> • <code>getSoftwareEox</code> • <code>getSoftwareEoxBulletin</code>

NPDR	The Not Processed Device Report API will return all devices in an inventory upload that were not processed by PSS.	<ul style="list-style-type: none"> • <code>getNPRDDetails</code>
Last Updated	This service call provides the last updated dates for the following three APIs— Inventory, Contracts and Alerts along with Partner name, Entitled Company, Inventory Name, Appliance ID, and response timestamp.	<ul style="list-style-type: none"> • <code>getLastUpdatedDetails</code>

Overview

The intention of the PSS API platform is to provide a batch access facility rather than an interactive facility. This approach removes the issues associated with UI and latency associated with large data sets or slower links. The batch model requires a data repository local to your Partner application. Consider the following diagram which shows a high level overview of how your constructed application can use PSS APIs to provide you with your own reports using PSS API calls.

From an API client application you can use the PSS Web Services Description Language (WSDL) to call an API service. The smart data returned from the API call should then be stored in a local data repository at the Partner site. Once you have the smart data in your local repository, you can present the data as needed in your own Partner specific reports or UIs. Periodically or on demand you may retrieve / refresh the smart data and store it in your partner data repository.

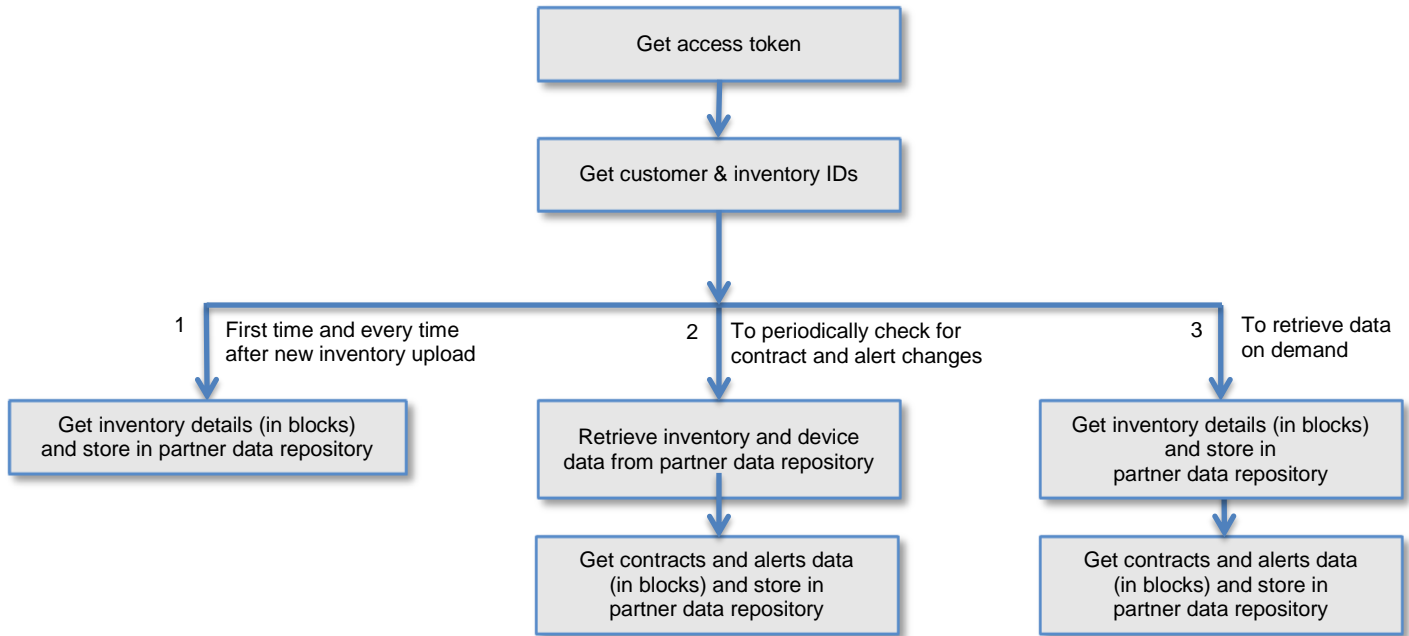


- PSS APIs are most efficiently used when they are called by a client application. The data returned from the API call should then be stored locally in your partner data repository. This practice ensures your partner application will run more quickly because data is accessed from the local data repository instead of from Cisco.
- Because inventory data rarely changes, repeatedly calling the Cisco PSS API Service Publisher to collect inventory details could cause an extremely slow application response time and therefore is not recommended.
- Depending on your data requirements, there may be a need to make occasional, on-demand API calls to retrieve contract or alerts information for one device.
- After your API client application calls the PSS APIs, the application should store the data returned from the call into your partner data repository
- Your application can fetch recently pulled data from your local data repository and use it within your partner specific custom reports and UIs.

Different Ways to Get and Store Data

Customer inventory collection information can be accessed using smart API calls. Refer to the following diagram to help you see:

- When data changes frequently you should retrieve that data more often from Cisco.
- When data changes infrequently, conserve API calls by making fewer inquiries from Cisco against the data.



The first time you upload a new inventory and every time after a new inventory upload, call the inventory API service and store the returned data into your partner data repository.

Contract and alerts have more frequent changes. A periodic set of API calls to update your local data repository with the latest contract changes and alerts published that affect devices in your inventory is recommended. Before placing an API Service call, retrieve the inventory and device data from your partner data repository so you can use it in the contract and alerts API calls to fetch the appropriate data. Place the API Service call to retrieve the new or changed data and store it in your data repository.

If you are placing a real time API call to retrieve contracts and/or alerts data for only one device, you can program your application to retrieve the inventory details in real time and then store the returned data in your Partner data repository. Then use the returned data to retrieve the contract and/or alerts data and store it in your data repository.

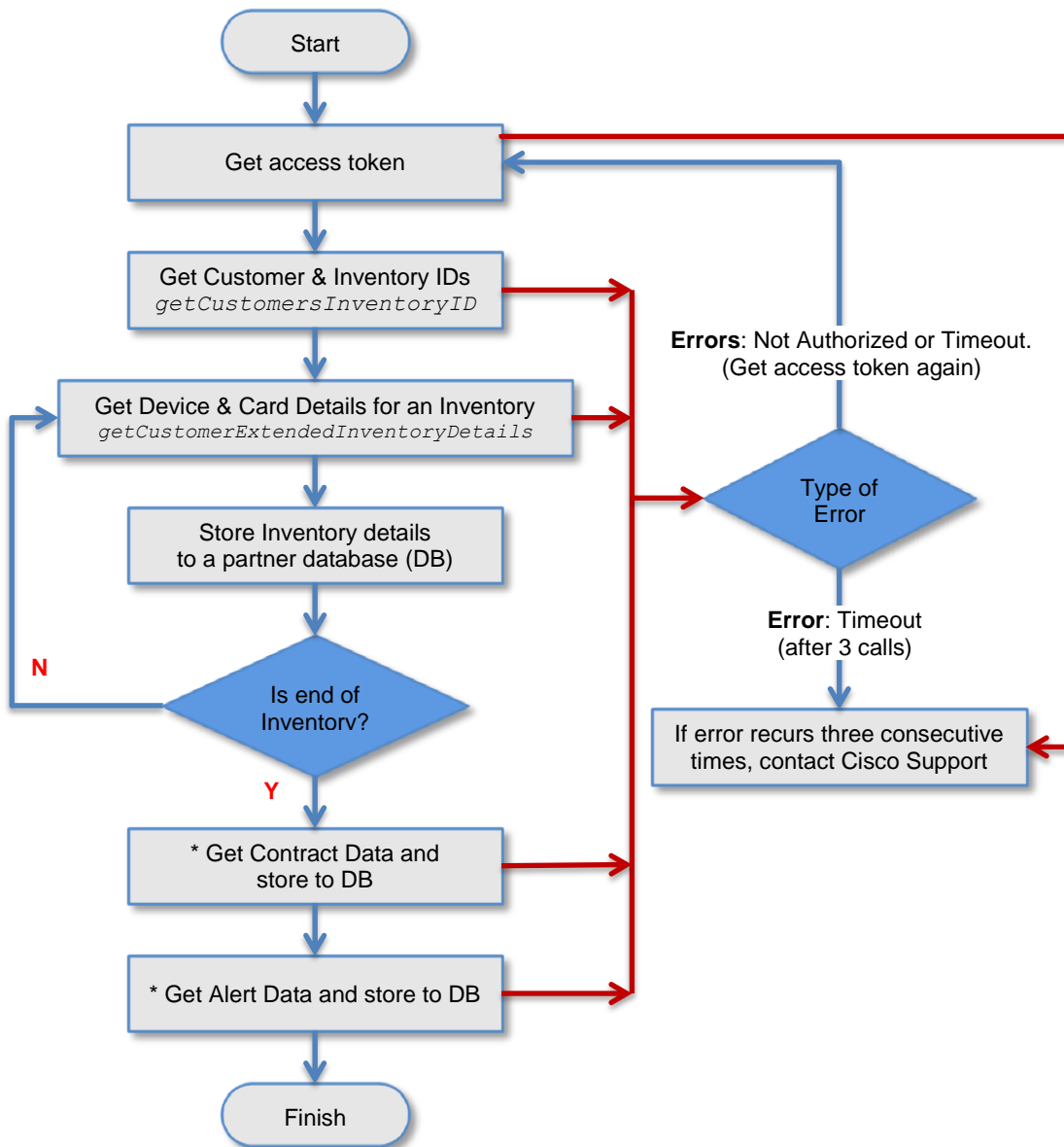
Leading Practices When Calling and Using APIs

The list of practices below will assist you when designing and creating custom applications which use PSS API Services.

- Client ID/Secret credentials are provided to you after you register an application on the API Console. These credentials are needed to get an access token, which can then be used to make PSS API Service calls. Because the Client ID/Secret credentials give access to all inventory details, you should share your Client ID / Secret only with individuals in your organization who actually need it.
- For optimum application performance, call PSS APIs only when needed and then store the data returned by the call in a local partner data repository for offline usage.
- Call the Inventory API only when a new upload has been completed because the data repository only reports what you have uploaded.
- To prevent issues with the timeout of API calls, use paginated APIs (for example, *getCustomerInventoryPaginatedDetails* to get smaller blocks of inventory data, than if you use the *getCustomerInventoryDetails* instead, the inventory details for only the first 500 devices will be returned. The *getCustomerExtendedInventoryDetails* service also offers the ability to paginate.
- To achieve a quicker response time when calling the PSS API Services for contract and alert information, call APIs by passing a specific Device ID or by calling a limited block of data. Trying to capture too much information can result in timeout errors and slow response times.
- Include logic to handle the errors returned by APIs in your application code as per the recommendations in the [Handling Error Conditions](#) section below.
- One common error when a partner application makes a PSS API service call is the time out error. If this happens in your application, attempt the request again (up to three times). If the time out error recurs three consecutive times, contact the [Smart Services Bureau](#) for support.
- Log the status of API calls with a timestamp in your application. Timestamps are used for audit purposes and to better track and troubleshoot API calls or issues.
- Have your application code get a fresh token every 60 minutes. All PSS API calls issued during the 60 minutes after you requested an access token can successfully use the token. If you do not re-create the token and your application issues API calls after the 60 minutes have elapsed, a timeout error will be returned.

Usage: Inventory Extended Details API

The paginated Inventory Extended Details API call, `getCustomerExtendedInventoryDetails` can be used to pull inventory details; controlled by pagination. Besides the Customer and Inventory input parameter's, you must also specify the `pageStart` parameter – usually set to 1.

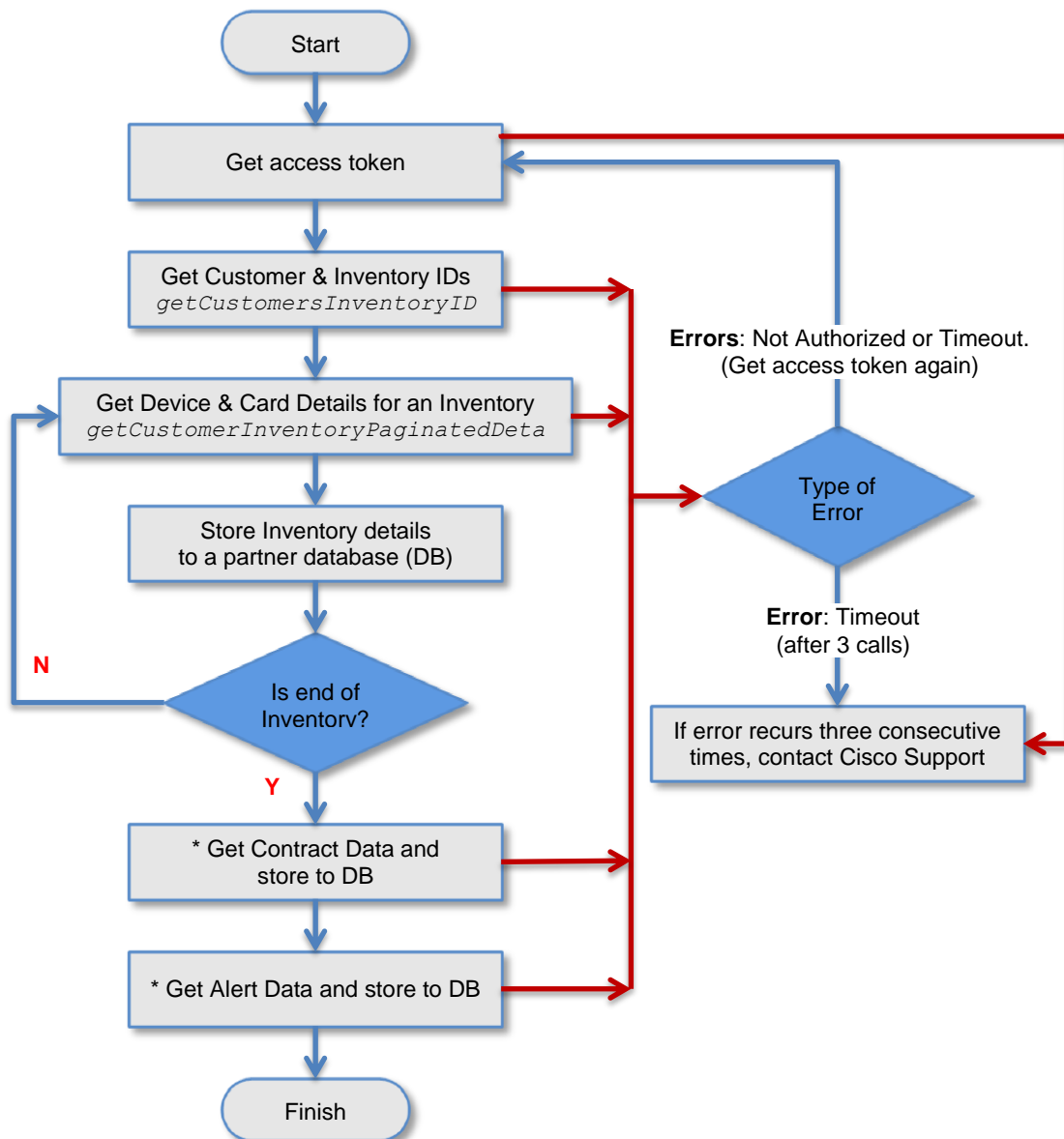


* This API returns page information, including page size, current page, total pages and PID count .

Usage: Inventory (Paginated) APIs

The Inventory APIs are used to fetch a list of all your end customers and their associated inventories that have been uploaded. When you use the `getCustomersInventoryIds` API service call and do not include any customer IDs, the response returned by the call will include the name and customer ID for each of your registered customers as well as their inventory name and ID.

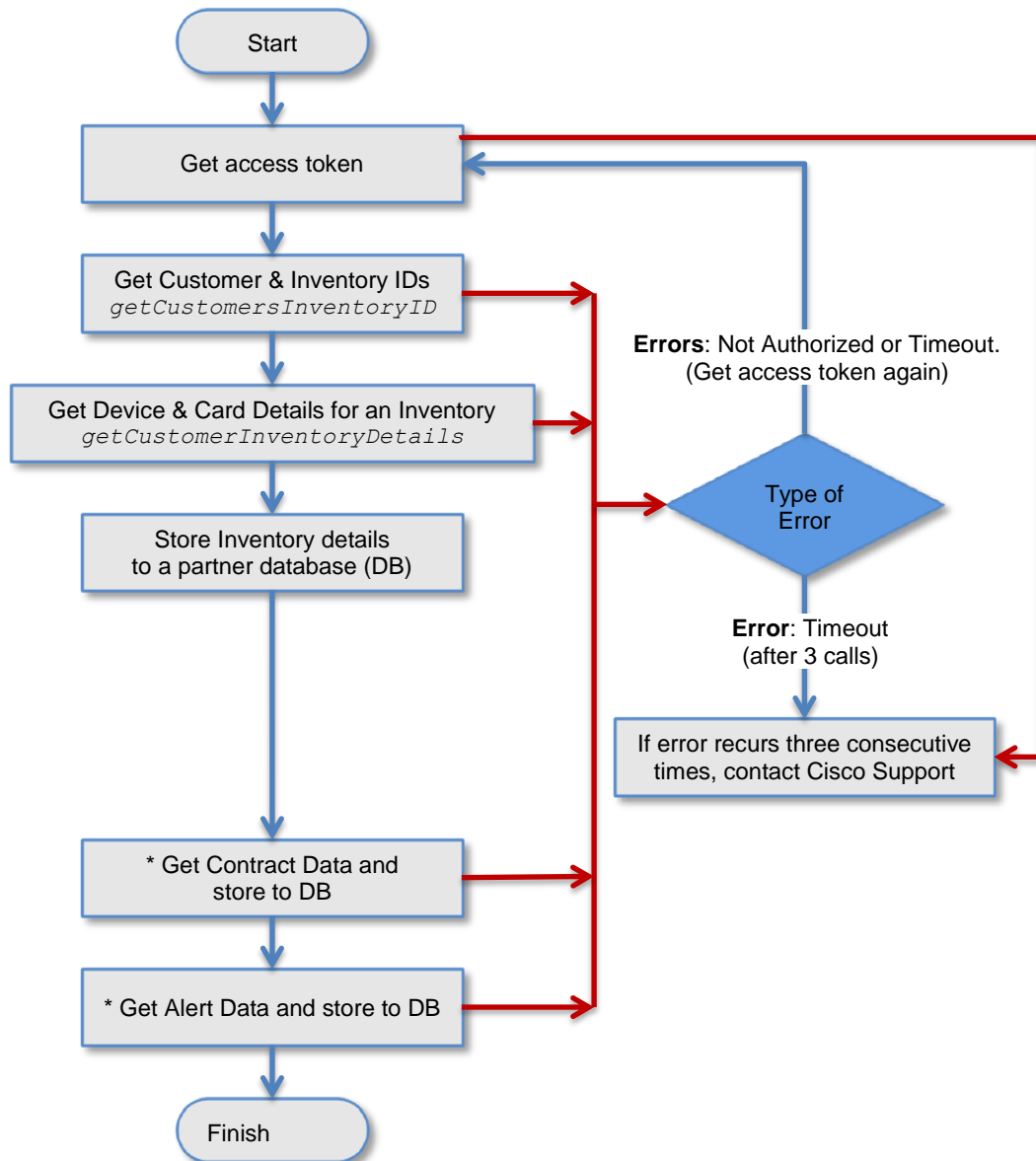
The `getCustomerInventoryPaginatedDetails` is the recommended API service call to return a collection of all Inventory elements (device details, chassis details) for a given customer inventory. This PSS API call includes a parameter to return data in manageable blocks called pages and it helps reduce time out errors. Red arrows in the following diagram indicate what to do when errors occur.



* After inventory details have been stored to your partner data repository you are not required to use the PSS API Contract or Alert service calls immediately. They can be called as shown above or later in a separate operation.

Usage: Inventory (Non Paginated) API

Note Cisco recommends to use the `getCustomerExtendedInventoryDetails` API or the `getCustomerInventoryPaginatedDetails` API for all Inventory Service API usage. The `getCustomerInventoryDetails` does not support pagination, so it is limited to small customer network collections, and therefore, not recommended for use as a standard practice.



* After inventory details have been stored to your partner data repository you are not required to use the PSS API Contract or Alert service calls immediately. They can be called as shown above or later in a separate operation.

Usage: Contract API

The Contract service has only one API call associated to the service. This PSS API Service call returns all the contract information, including contract details, coverage and site address information.

Prerequisites:

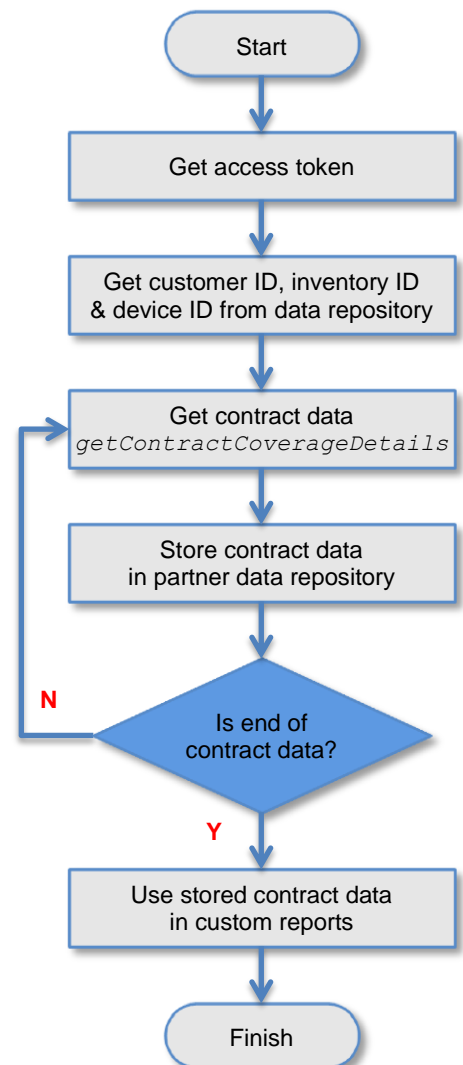
Get input data stored in your local database before calling the Contract APIs:

- Customer IDs (required)
- Inventory IDs (required)
- Device IDs (optional)

These should have been fetched and put in your data repository after a PSS API *getCustomerInventoryIDs* API call described in the [Inventory](#) section above.

Steps:

1. Obtain an access token from the Cisco API Console by providing a valid Client ID and Client Secret.
2. Obtain the customer ID, inventory ID and device ID from your data repository.
3. Pass the customer ID, inventory ID, device ID, and token to the *getContractCoverageDetails* API to retrieve contract information.
4. Store the contract information in your data repository.
5. Use this inventory and contract data in custom reports.



Usage: Field Notice Alert APIs

The FN Service includes two API calls and returns all Field Notices for the specified customer, inventory and devices.

- `getFN`
- `getFNDetails`

Prerequisites:

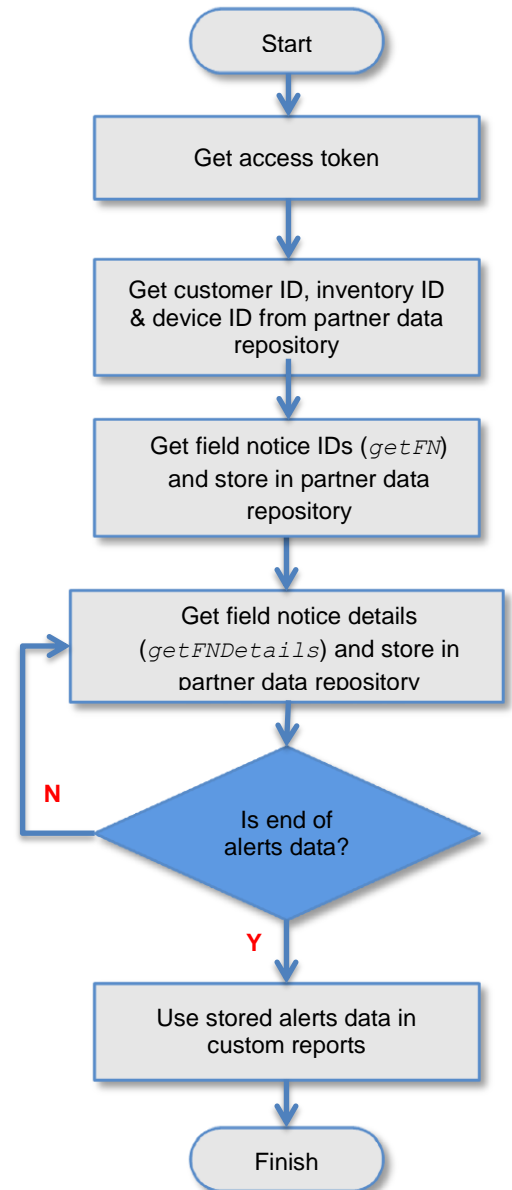
Get input data stored in your local data repository before calling Field Notice APIs:

- Customer IDs (required)
- Inventory IDs (required)
- Device IDs (optional)

These should have been fetched and put in your data repository after a PSS API `getCustomerInventoryIDs` API call described in the [Inventory](#) section above.

Steps:

1. Obtain an access token from the Cisco API Console by passing a valid Client ID and Client Secret.
2. Retrieve the customer ID, inventory ID, and device ID from your data repository.
3. Obtain the field notice IDs and store them in your data repository.
4. Obtain field notice details and store them in your data repository.
5. Use this data in your custom reports.



Usage: PSIRT Alert APIs

The PSIRT Service includes two API calls and returns all PSIRT alerts for the specified customer, inventory and devices.

- `getPSIRT`
- `getPSIRTDetails`

Prerequisites:

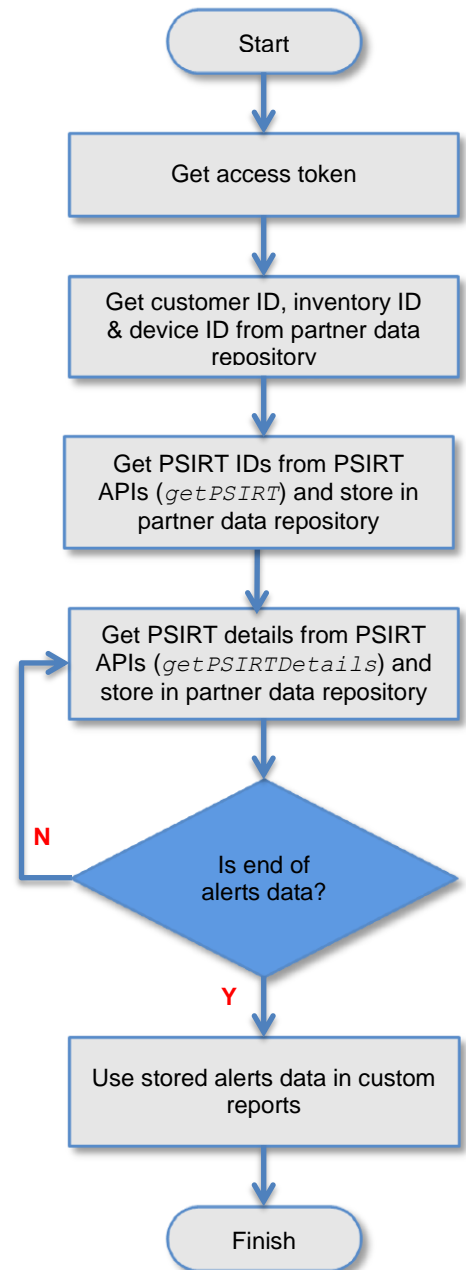
Get input data stored in your local data repository before calling PSIRT APIs:

- Customer IDs (required)
- Inventory IDs (required)
- Device IDs (optional)

These should have been fetched and put in your data repository after a PSS API `getCustomerInventoryIDs` API call described in the [Inventory](#) section above.

Steps:

1. Obtain an access token from the Cisco API Console by passing a valid Client ID and Client Secret.
2. Retrieve the customer ID, inventory ID, and device ID from your data repository.
3. Obtain the PSIRT IDs from PSIRT APIs and store them in your data repository.
4. Obtain details for a specific PSIRT and store them in your data repository.
5. Use this data in your custom reports.



Usage: IntelliShield Alert APIs

The IntelliShield Service includes two API calls and returns all IntelliShield alerts for the specified customer, inventory and devices.

- `getIS`
- `getISDetails`

Prerequisites:

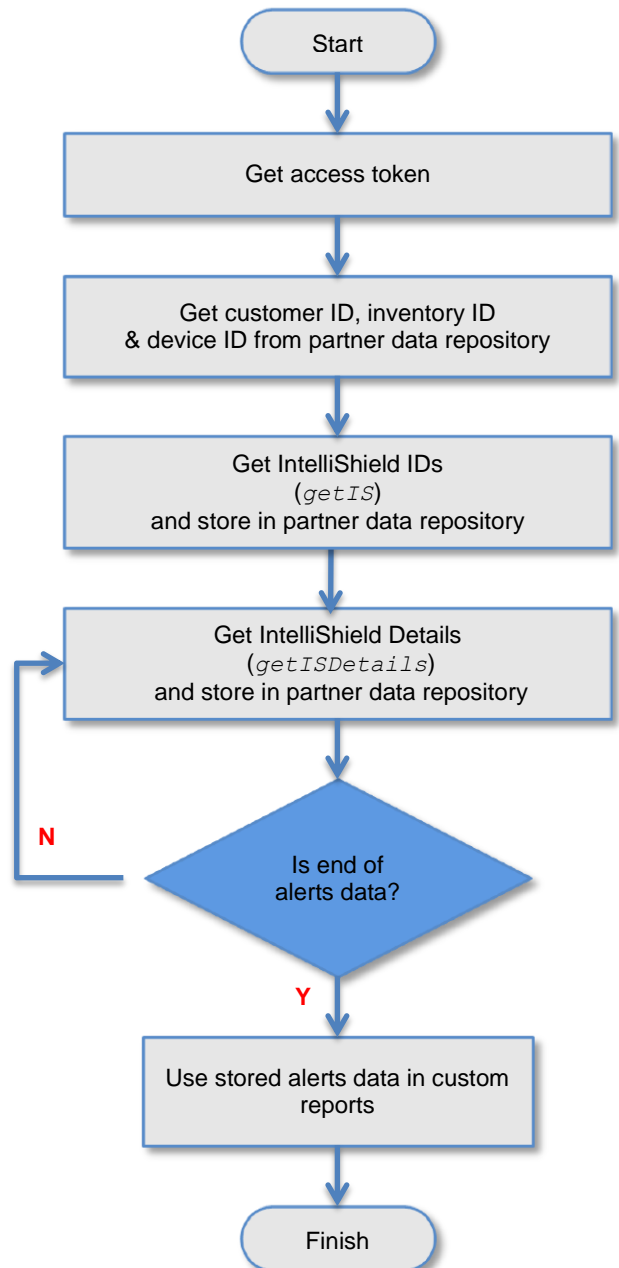
Get input data stored in your local data repository before calling IntelliShield APIs:

- Customer IDs (required)
- Inventory IDs (required)
- Device IDs (optional)

These should have been fetched and put in your data repository after a PSS API `getCustomerInventoryIDs` API call described in the [Inventory](#) section above.

Steps:

1. Obtain an access token from the Cisco API Console by passing a valid Client ID and Client Secret.
2. Retrieve the customer ID, inventory ID, and device ID from your data repository
3. Obtain the IntelliShield IDs from IntelliShield APIs and store them in your data repository.
4. Obtain IntelliShield details and store them in your data repository.
5. Use this data in your custom reports.



Usage: Hardware Eox APIs

The Hardware Eox Service includes three API calls and returns hardware end-of-life ID and product ID information for the specified customer, inventory and devices.

- `getHwEox`
- `getHwEoxBulletin`
- `getHwEoxMigrationPIDs`

Pre-Requisites:

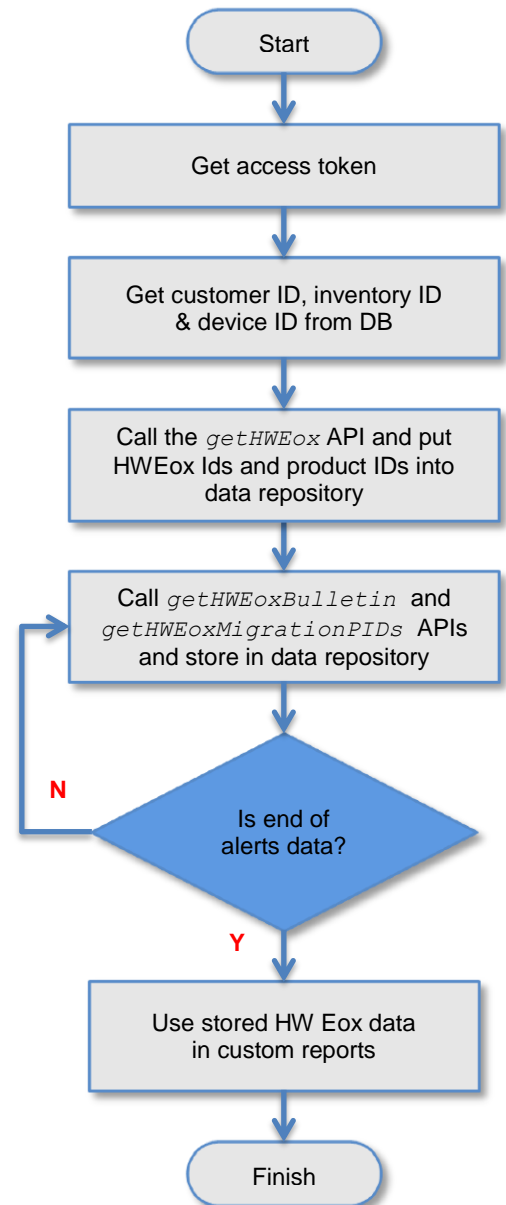
Need input data already fetched through Inventory APIs and stored in your local data repository before calling HW Eox APIs:

- Customer IDs (required)
- Inventory IDs (required)
- Device IDs (optional)

These should have been fetched and put in your data repository after a PSS API `getCustomerInventoryIDs` API call described in the [Inventory](#) section above.

Steps:

1. Obtain an access token from the Cisco API Console by passing a valid Client ID and Client Secret.
2. Retrieve from your data repository the customer ID, inventory ID, and device ID to be used in the HWEox Service API calls.
3. Obtain the hardware end-of-life IDs and product IDs using the `getHWEox` API call and store them in your data repository.
4. Fetch descriptive details using the `getHwEoxBulletin` and `getHwEoxMigrationPIDs` API calls and store them in your data repository.
5. Use the hardware end-of-life data stored in your data repository in your custom reports.



Usage: Software Eox APIs

The Software Eox Service includes two API calls and returns software end-of-life ID and product ID information for the specified customer, inventory and devices.

- `getSoftwareEox`
- `getSoftwareEoxBulletin`

Prerequisites:

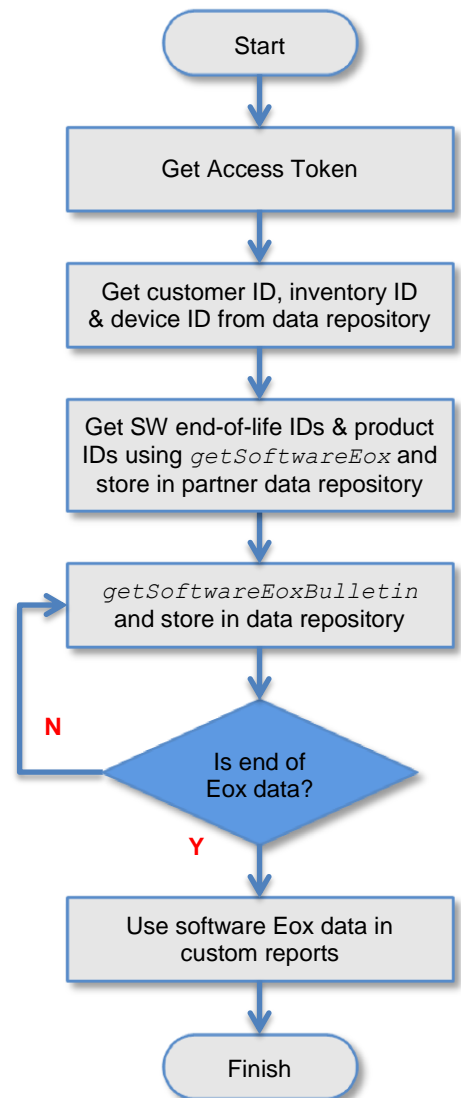
Need input data already fetched through Inventory APIs and stored in your local data repository before calling Software Eox APIs:

- Customer IDs (required)
- Inventory IDs (required)
- Device IDs (optional)

These should have been fetched and put in your data repository after a PSS API `getCustomerInventoryIDs` API call described in the [Inventory](#) section above.

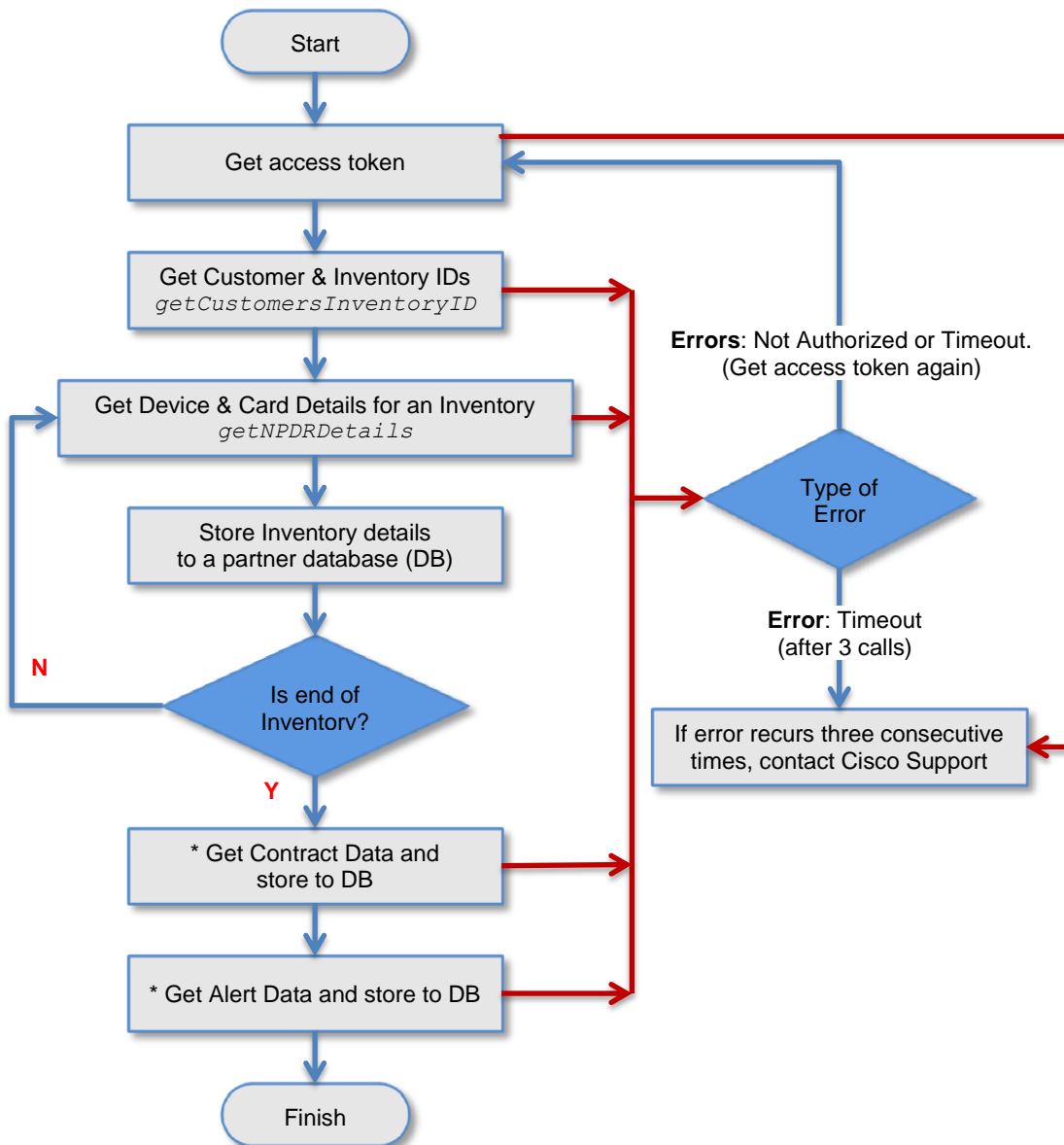
Steps:

1. Obtain an access token from the Cisco API Console by passing a valid Client ID and Client Secret.
2. Retrieve from your data repository the customer ID, inventory ID and device ID to be used in the SWEox Service API calls.
3. Obtain the software end-of-life ID and product IDs using the `getSoftwareEox` API call and store them in your data repository.
4. Obtain software end-of-life bulletin information using the `getSoftwareEoxBulletin` API call and put them in your data repository.
5. Use software Eox data stored in your data repository in your custom reports.



Usage: Not Processed Device Report API

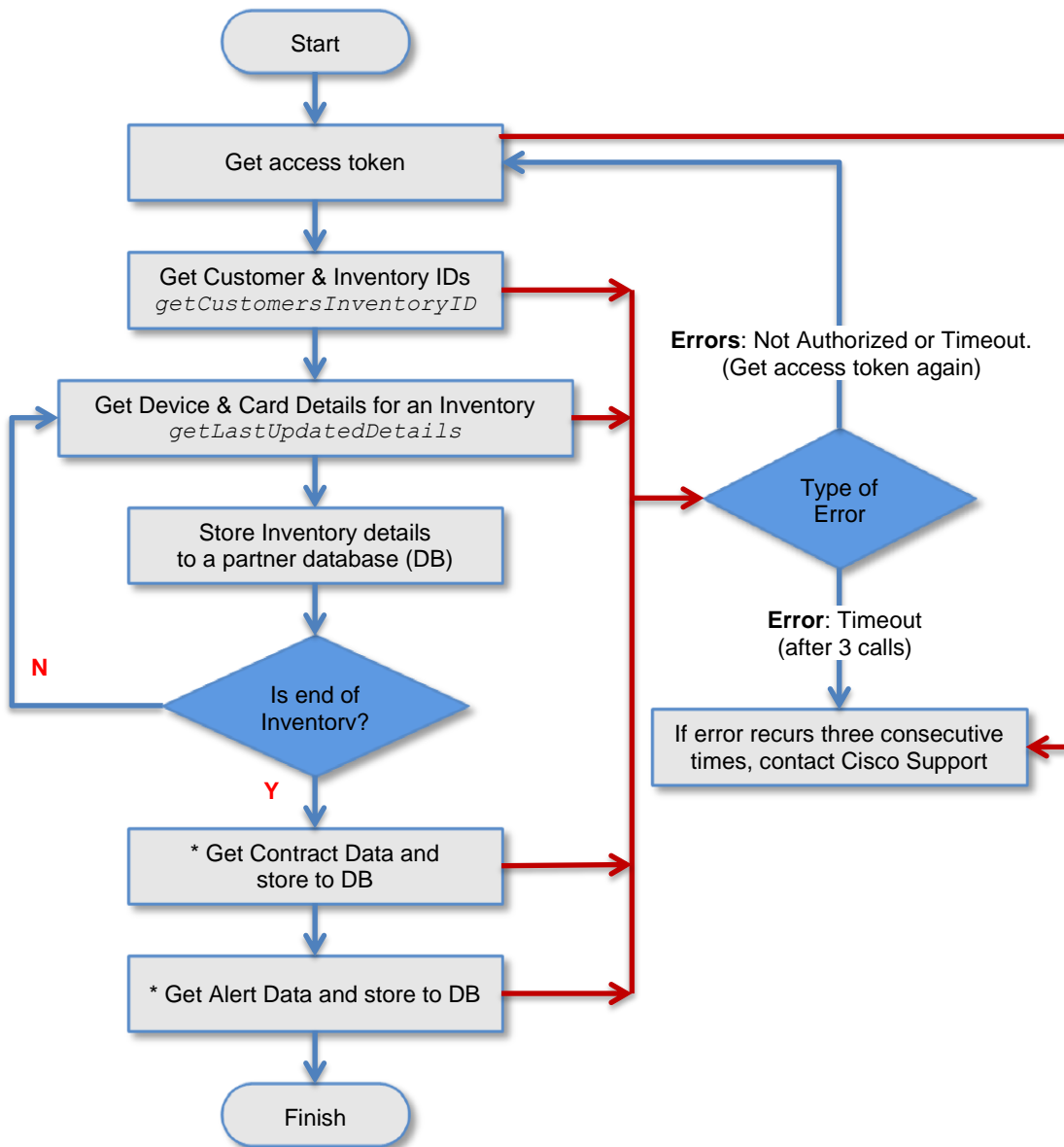
The Not Processed Devices API call, `getNPDRDetails` , can be used to report those devices in an inventory upload that were not processed by PSS and the Cisco datacenter. The most common reason for this is that the device is not recognized as a Cisco device (PID + SN).



* Optionally you can send the applianceID parameters(s) to filter the response; else the API will return a full collection of data for the given InventoryID for the given CustomerID .

Usage: Last Updated API

The Last Updated (date & time) API call, `getLastUpdatedDetails` provides the last updated dates for the following three APIs—Inventory, Contracts and Alerts along with Partner name, Entitled Company, Inventory Name, Appliance ID, and response timestamp..



* The calls to the Inventory, Contract and Alert databases are to fetch LastUpdateTime only.

Do's and Dont's

The following list of suggestions will help your PSS organization use PSS API Service calls efficiently.

- *Do* use the paginated Inventory API to retrieve blocks of 500 Product IDs. This practice will avoid errors because your inventory is more than 500 PIDs
- *Do* call contract and alert APIs in smaller blocks when more than 10,000 Product IDs are in a client inventory. This practice will help you avoid timeout errors with your application.
- *Do* call token API to get a fresh token every 60 minutes.
Tokens are only good for 60 minutes. If you use a token for more than 60 minutes, an error message, "ERR_403_NOT_AUTHORIZED", will be returned. The not authorized error indicates that a new token needs to be generated.
- *Do* try calling APIs up to three times before raising an incident with support, in case of errors. Cisco's API engine may be busy and may time out when the volume of traffic is too high or the internet speed between your network and Cisco is too slow.
- *Do not* call APIs when the data being called already exists in your partner data repository. Your application will return results more quickly when the data is retrieved from a local data repository.

Handling Error Conditions

This section describes causes for the three most common error messages returned by APIs as well as steps needed to resolve the errors.

1. Not Authorized

Cause

This error occurs when an incorrect/invalid Client ID and Secret are being used while generating the access token, or when the token has expired while making service API calls.

Solution

Be sure that the Client ID and Client Secret are valid and then generate a new token to resolve this error.

2. Request timed out

Cause

A service call has timed out because processing time took longer than expected. This error occurs when a customer has a very large inventory, when Cisco's API publisher is busy, or when speed between your application and Cisco is slow.

Solution

Here are four possible solutions:

1. When using the Inventory service calls, retrieve smaller blocks of data with the *getCustomerInventoryPaginatedDetails* API.
2. Retrieve only the data needed by the Contract and Alert service calls by providing specific Customer ID, Inventory ID and Device IDs.
3. Store the data returned from API service calls in your partner data repository. For reports that will include a lot of data, retrieve the data to be used in custom reports from your partner data repository.
4. Once Inventory data has been stored in your data repository, call the Inventory service APIs only if a new upload has been completed. Otherwise, use the inventory data already stored in your data repository.

3. EXCEED-MAX-LIMIT-101

Cause

The API cannot respond due to the large size of the inventory. This error occurs when the number of device product IDs (PIDs) returned for a specific API Contract or Alert service call for a particular inventory is more than 10,000. **This error occurs for the Contract and Alert APIs only**

Solution:

Retrieve contracts or alerts data in smaller blocks.

All API calls for retrieving contracts or alerts data include an optional parameter, *device IDs*. When you enter a list of device IDs into an API call, it will only return the contract or alerts data for the device IDs you specified. We recommend entering up to 100 device IDs into an API call as a method for eliminating the EXCEED-MAX-LIMIT-101 error.

Note: For all other error codes, please refer to the [PSS API Developer Guide](#).

Appendix A: References and Support Resources

References

1. [PSS API Quick Start Guide](#)
2. [API Console User Guide](#)
3. [PSS API Developer Guide](#)

PSS Support Community

The private [Cisco Support Community for Partner Support Service](#) is a starting point to learn about PSS APIs. From there you can view short videos, download documentation and access the API console. To subscribe to the [Support Community for Partner Support Service](#) send your full name, company name, and Cisco ID to psscommunity@external.cisco.com.

Smart Services Bureau

As a Cisco Partner, you may contact the [Smart Services Bureau](#) for PSS API support. This great resource, included with collaborative service contracts, is available globally during local business hours, 5 days a week.

For partners in the US/Canada contact:

- Phone **1-877-330-9746** from 8 a.m. to 5 p.m. local business hours.
- E-mail support is available 24 x 7 at: ask-smart-services@cisco.com

For partners in Europe phone:

- Austria +08 0 000 6206
- Belgium +08 0 049 913
- France +08 0 511 9745
- Germany +08 0 058 91725
- Italy +80 0 085 681
- Netherlands +08 0 002 01276
- Spain +80 0 600 472
- Switzerland +08 0 084 0011
- UK +08 002 795 112

For Partners in Asia contact:

- Australia: 1-800-805-227
- China: 400-810-995
E-mail: ask-smart-services-cn@cisco.com
- Korea: 080-377-0880, option 6, pin code 8899
E-mail ask-smart-services-kr@cisco.com
- Japan: 0120-832-038
E-mail ask-smart-services-jp@cisco.com
- Taiwan: 0800-666-588, option 5, pin code 8899 (Mandarin support)
E-mail: ask-smart-services-kr@cisco.com
- Hong Kong: 30714911, option 6 (English support)
E-mail: ask-smart-services-kr@cisco.com

For all other countries

Please select your local phone number from Cisco worldwide contact directory:
http://www.cisco.com/en/US/support/tsd_cisco_worldwide_contacts.html



Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at www.cisco.com/go/trademarks. Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)