



APPLICATION ACCESS MANAGER

AAM INTEGRATION - TECHNICAL DOCUMENTATION TEMPLATE

DOCUMENT PURPOSE: THIS TEMPLATE IS TO BE COMPLETED BY PARTNER AND IS REQUIRED FOR CYBERARK SECURED CERTIFICATION. THE AAM INTEGRATION TECHNICAL DOCUMENTATION WILL BE MADE AVAILABLE TO PARTNERS, CUSTOMERS AND PROSPECTS.

Name of Company: USERCUBE

Website: <https://www.usercube.com/>

Name of Product: Usercube

Version: 5.1.4

Date: 14/09/2020

USERCUBE SOLUTION OVERVIEW

Usercube is an Identity Governance and Administration (IGA) solution. IGA brings together all an organization's processes to enable each identified individual to have the correct access rights at the right time for the right reasons. **This type of solution is important for several reasons:**

- Monitor everyone who has access, their missions and their locations
- Effectively manage employee arrivals, job changes and departures
- Optimize resource allocation workflows
- Increase security
- Facilitate audits with reliable and understandable data

Usercube V5.1.4 can be installed by 2 ways.

- Installed in the cloud, Usercube easily integrates with your information system without the need for specific expertise. Usercube benefits from MS-Azure's capacity. Your data is stored near you based on Azure's data-center locations, which increases the application responsiveness.
- Usercube can also be installed on-Premise, on the customer's servers. The solution allows for plug-and-play integration with cloud services as well as legacy applications.

KEY BENEFITS

Usercube builds a **repository of organizations**, sites, users and resources, from different sources, to become the centralized location for reliable and exhaustive information. The directory is powered by real-time jobs assignment of people, administration tasks and personal updates through the self-service screen. This constantly ensures a very high level of accuracy of the information contained. The repository can interface with other software to share knowledge of hierarchy and organization.

Inside organizations, people frequently move around (hires, departures, subsidiary creations, office closings, reorganizations, moves ...). Because the company's organization is defined in Usercube, workflows **notify the necessary people in real time**, making these actors more efficient and therefore reducing wasted time with each change. Additionally, the constantly updated repository and **the traceability of all requests** allows easy generation of compliance reports.

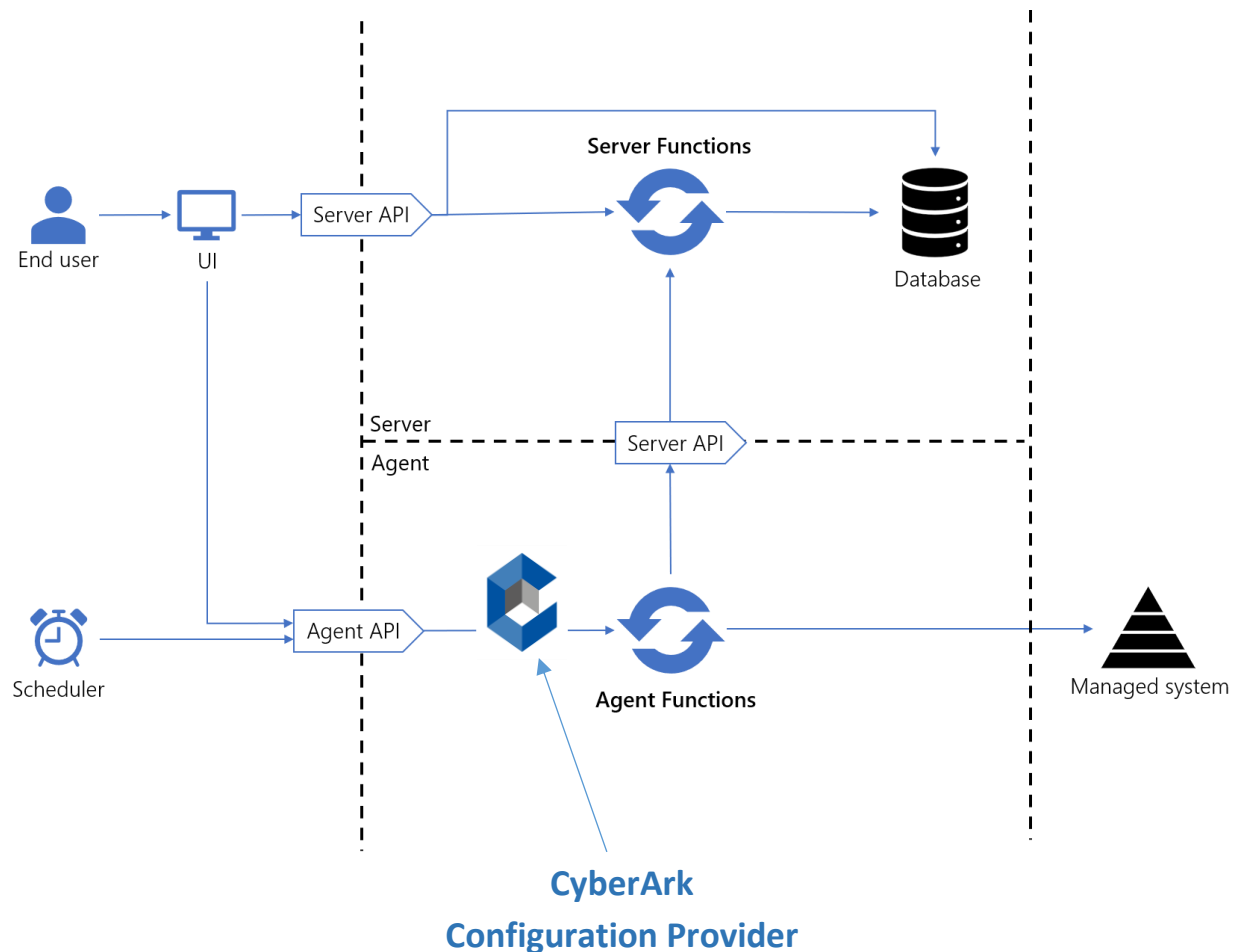
Usercube is the solution for centralizing **requests regarding computer rights and equipment endowments**. Each user, manager or local liaison, can make a request **to obtain, modify or revoke/return** access rights or equipment for the users in his/her scope.

Access rights for customer's on-premise applications or cloud services can either be provisioned manually via notifications to resource managers, or automatically via application connectors.

Usercube makes it possible to enumerate the IT system's Security Policy regarding access right controls and can automate the exercise of these controls. Every organization is thus protected from security breaches including breaches related to the **segregation of duties** (SoD).

Usercube interacts with many data sources and target systems. For this, Usercube uses service accounts. Integration with CyberArk AAM enables Usercube to benefit the highest security scheme by storing all credentials of its services accounts into CyberArk.

PRODUCT DIAGRAM & DESCRIPTION OF PRODUCT INTEGRATION



Usercube is composed of five main components:

- The **Server**: it provides the APIs used by the UI. It uses Server Functions when needed.
- The **Server Functions**: they are used by the Server. They perform module-specific computations such as data loading, assignment policy enforcement, assignment review notifications. The functions are fetching their configuration and data from the database.
- The **Database**: it stores all the data needed by Usercube. This includes, but is not limited to, the applicative configuration, the resources repository, the assignment policy, and the computed assignments.
- The **Agent**: it provides Agent Functions that are used to interact with the managed systems. Only Agent Functions credentials from the managed systems and the network are necessary to read/write data from/to the managed systems.
- The **Agent Functions**: they are usually involved in the synchronization (Up or Down) process. Some functions extract data from managed systems (taking a managed system data snapshot) and send it to the server. Other functions are fetching the provisioning orders prepared by the server during the assignment policy enforcement process and execute them on the target managed systems.

CyberArk AAM is integrated on the **Agent**, it makes the link between **Agent** and **Agent Functions** because it provides the credentials to launch an Agent Function.

We consider the Agent Function: fulfil an Active Directory. The Agent must provide the **Password** of the Active Directory to launch the Agent Function successfully.

Before the integration with CyberArk AAM, a JSON Configuration Provider is used. It reads a JSON file called "**appsettings.agent.json**" containing the settings for every Agent Function (Login, Password, Server etc...).

Usercube provides two CyberArk Configuration Providers:

- 1) **Agentless AAM**: call the Central Credential Provider Web Service using REST.
- 2) **Agent Based AAM**: need a Credential Provider to be installed locally. It uses the CyberArk C/C++ Application Password SDK.

The configuration to use these providers is at the end of this document in "**USERCUBE INSTALLATION & INTEGRATION CONFIGURATION**".

In the JSON file, the keys have the form:

InformationSystems:ManagedSystemSetting:ManagedSystemIdentifier:Setting.

In this example, the Agent Function needs a **Password** to connect to the managed system (Active Directory) so Usercube will search for the key: InformationSystems:ActiveDirectorySetting:AD_Export:**Password**.

All Configuration Providers implemented in Usercube search for this key but there are priorities. For example, CyberArk Configuration Provider has priority over the default provider. It means that if Usercube detects a value for this key in both providers then CyberArk's Configuration Provider will do the action.

The default Configuration Provider returns the value associated to the key in the JSON. It is the **Password** in our example.

CyberArk AAM Configuration Provider works with another JSON file called “**appsettings.cyberArk.agent.json**”. The previous key leads to a value in this JSON that corresponds to the account identifier of the account corresponding to the Active Directory in the CyberArk's safe. Thanks to this account identifier, Usercube can get the **Password** (in our example) but we can imagine getting other settings like the Login or the Server.

For every Agent Function we need at least two retrievals, one for the login and one for the password. Moreover, the Agent Functions are launched several times a day, so the Agent easily makes dozens of credential retrievals a day.

More explanations and examples are available at the end of this document in “**USERCUBE INSTALLATION & INTEGRATION CONFIGURATION**”.

AAM INSTALLATION

Refer to the “Credential Provider and ASCP Implementation Guide” or the “Central Credential Provider Implementation guide” for CyberArk Agent based or Agentless installation and configuration.

If you use the Credential Provider with Application Password SDK (Agent Based AAM) then the C/C++ Application Password SDK dynamic link library called **CPasswordSDK.dll** (on 32 bits systems) or **CPasswordSDK64.dll** (on 64 bits systems) must be placed in the **Runtime** folder of Usercube or in the system library folders to use it.

AAM CONFIGURATION

DEFINING THE APPLICATION ID (APPID) AND AUTHENTICATION DETAILS

To define the Application, here are the instructions to define it manually via CyberArk's PVWA (Password Vault Web Access) Interface:

1. Logged in as user allowed to managed applications (it requires Manage Users authorization), in the Applications tab, click **Add Application**; the Add Application page appears.

Add Application

Name: Usercube-Agent

Description: Usercube's agent that calls CyberArk to get logins and passwords.

Business owner

First Name:

Last Name:

Email:

Phone:

Location: \

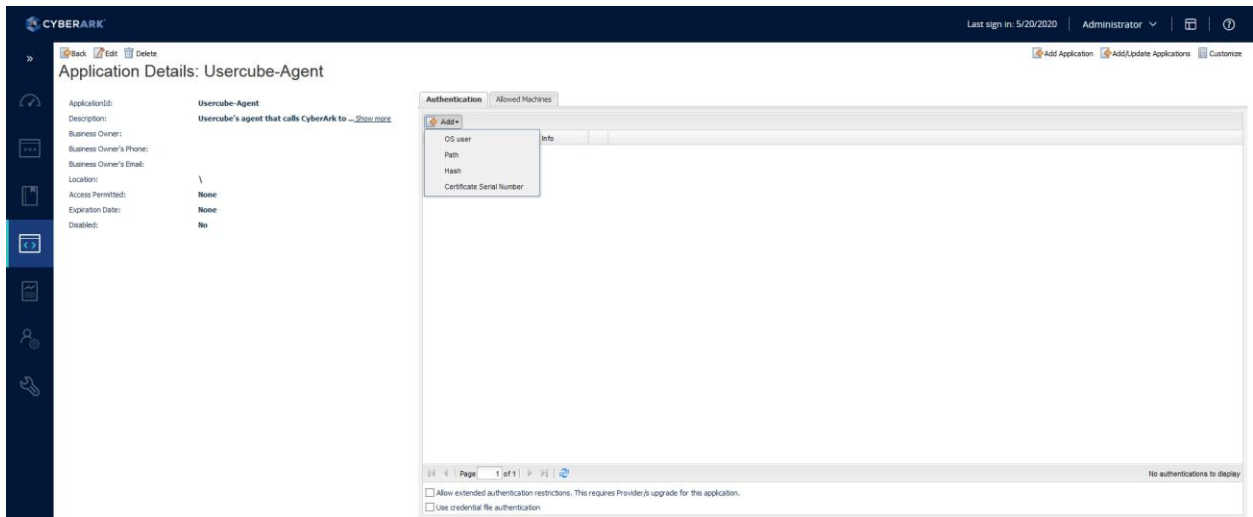
☐ Access Permitted: From: To:

☐ Expiration Date:

☐ Disabled

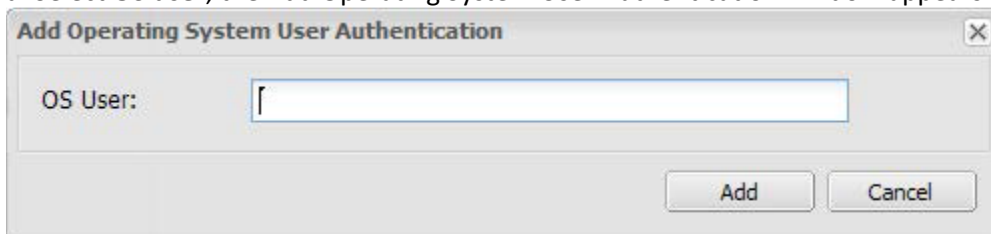
Add Cancel

2. Specify the following information:
 - In the Name edit box, specify the unique name (ID) of the application. The recommended Application ID for this integration is: APP ID = Usercube-Agent.
 - In the Description, specify a short description of the application that will help you identify it.
 - In the Business owner section, specify contact information about the application's Business owner.
 - In the lowest section, specify the Location of the application in the Vault hierarchy. If a Location is not selected, the application will be added in the same Location as the user who is creating this application.
3. Click **Add**; the application is added and is displayed in the Application Details page.

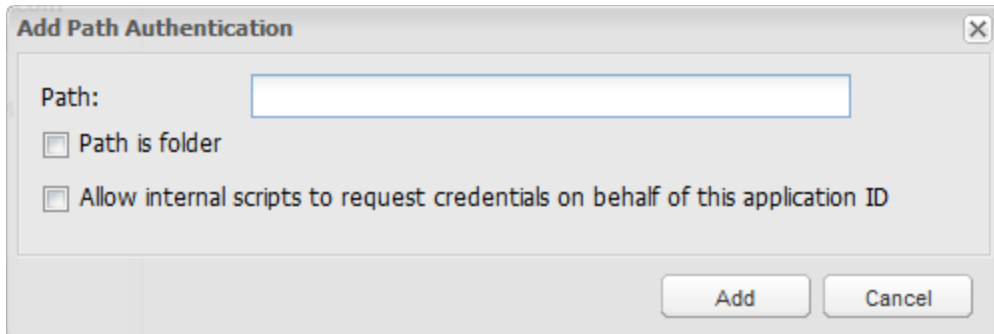


- **Allowing extended authentication restrictions.** This enables you to specify an unlimited number of machines and Windows domain OS users for a single application. Please check this box.
4. Specify the application's **Authentication** details. This information enables the Credential Provider to check certain application characteristics before retrieving the application password. Every authentication method is compatible with Usercube.
 - In the Authentication tab, click **Add**; a drop-down list of authentication characteristics is displayed.
 - Select the authentication characteristic to specify.

5. Specify the OS user:
 - a. Select **OS user**; the Add Operating System User Authentication window appears.



- b. Specify the name of the OS user who will run the application, then click **Add**; the OS user is listed in the Authentication tab.
6. Specify the application path:
 - a. Select **Path**; the Add Path Authentication window appears.

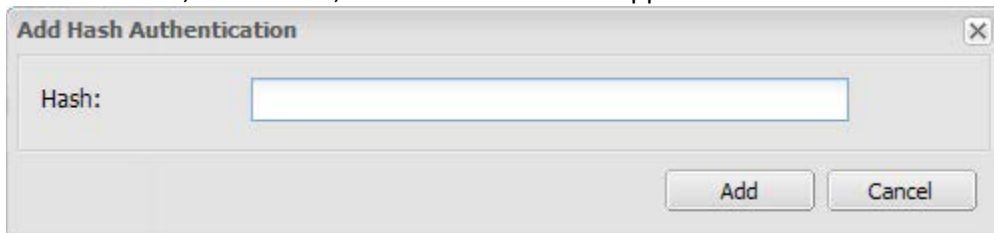


The 'Add Path Authentication' dialog box contains a 'Path:' text field, two checkboxes labeled 'Path is folder' and 'Allow internal scripts to request credentials on behalf of this application ID', and 'Add' and 'Cancel' buttons at the bottom right.

- b. Specify the path where the application will run.
- c. To indicate that the specified path is a folder, select **Path is folder**.
- d. To allow internal scripts to retrieve the application password for this application, select **Allow internal scripts to request credentials on behalf of this application ID**.
- e. Click **Add**; the Path is added as an authentication characteristic with the information that you specified.

7. Specify a hash:

- a. Run the AIMGetAppInfo utility to calculate the application's unique hash.
- b. Copy the hash value that is returned by the utility.
- c. In the PVWA, select **Hash**; the Add Hash window appears.



The 'Add Hash Authentication' dialog box contains a 'Hash:' text field and 'Add' and 'Cancel' buttons at the bottom right.

- d. In the Hash edit box, paste the application's unique hash value, or specify multiple hash values with a semi-colon. You can add additional information in a comment after each hash value specified for an application by specifying '#' after the hash value, followed by the comment.

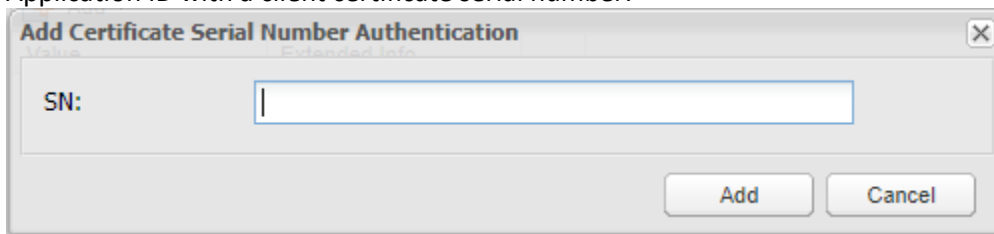
For example, OE883B7OD5B6E3EE37D37198049C9507C8383DB6 #app2

Note: The comment must not include a colon or a semicolon.

- e. Click **Add**; the Hash is added as an authentication characteristic with the information that you specified.

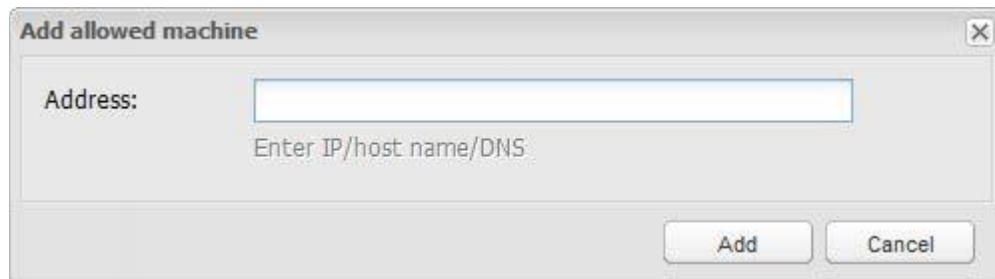
8. Specify the Certificate Serial number:

- a. When using the Agentless AAM (Central Credential Provider) there is an option to protect the Application ID with a client certificate serial number:



The 'Add Certificate Serial Number Authentication' dialog box contains an 'SN:' text field and 'Add' and 'Cancel' buttons at the bottom right.

9. Specify the application's **Allowed Machines**. This information enables the Credential Provider to make sure that only applications that run from specified machines can access their passwords.
 - In the Allowed Machines tab, click **Add**; the Add allowed machine window is displayed.

A screenshot of a Windows-style dialog box titled "Add allowed machine" with a close button (X) in the top right corner. The dialog has a light gray background. Inside, there is a label "Address:" followed by a text input field. Below the input field, there is a hint text "Enter IP/host name/DNS". At the bottom right of the dialog, there are two buttons: "Add" and "Cancel".

Add allowed machine

Address:

Enter IP/host name/DNS

Add Cancel

- Specify the IP/hostname/DNS of the machine where the application will run and will request passwords, then click **Add**; the IP address is listed in the Allowed machines tab. Make sure the servers allowed include all mid-tier servers or all endpoints where the AAM Credential Providers were installed.

PROVISIONING ACCOUNTS AND SETTING PERMISSIONS FOR APPLICATION ACCESS

For the application to perform its functionality or tasks, the application must have access to existing accounts, or new accounts to be provisioned in CyberArk Vault (Step 1). Once the accounts are managed by CyberArk, make sure to setup the access to both the application and CyberArk Application Password Providers serving the Application (Step 2).

1. In the Password Safe, provision the privileged accounts that will be required by the application. You can do this in either of the following ways:
 - **Manually** – Add accounts manually one at a time and specify all the account details.
 - **Automatically** – Add multiple accounts automatically using the Password Upload feature.

For this step, you require the **Add accounts** authorization in the Password Safe.

For more information about adding and managing privileged accounts, refer to **the Privileged Access Security Implementation Guide**.

2. Add the Credential Provider and application users as members of the Password Safes where the application passwords are stored. This can either be done manually in the Safes tab, or by specifying the Safe names in the CSV file for adding multiple applications.
 - i. Add the Provider user as a Safe Member with the following authorizations:
 - List accounts
 - Retrieve accounts
 - View Safe Members

Note: When installing multiple Providers for this integration, it is recommended to create a group for them, and add the group to the Safe once with the above authorization.

Add Safe Member

Search:

Search In:

Vault

Search

Selected Search: Vault

Name	Business Email	Full Name
------	----------------	-----------

☐ Access

☐ Use accounts

☒ Retrieve accounts

☒ List accounts

☐ Account Management

☐ Safe Management

☐ Monitor

☐ View Audit log

☒ View Safe Members

Add

Close

- ii. Add the application (the APPID) as a Safe Member with the following authorizations:

- Retrieve accounts

Add Safe Member

Search: Search In: Vault Search

Selected Search: Vault

Name	Business Email	Full Name
------	----------------	-----------

☐ Access
☐ Use accounts
☒ Retrieve accounts
☐ List accounts
☐ Account Management
☐ Safe Management
☐ Monitor
☐ View Audit log
☐ View Safe Members

Add Close

- iii. If your environment is configured for dual control:
 - In PIM-PSM environments (v7.2 and lower), if the Safe is configured to require confirmation from authorized users before passwords can be retrieved, give the Provider user and the application the following permission:
 - Access Safe without Confirmation
 - In Privileged Account Security solutions (v8.0 and higher), when working with dual control, the Provider user can always access without confirmation, thus, it is not necessary to set this permission.
- iv. If the Safe is configured for object level access, make sure that both the Provider user and the application have access to the password(s) to retrieve.

For more information about configuring Safe Members, refer to the **Privileged Access Security Implementation Guide**.

USERCUBE INSTALLATION & INTEGRATION CONFIGURATION

Refer to Usercube's documentation for the installation.

There are only one Application and one Safe dedicated to Usercube. Every account is stored in that safe.

In the safe, two accounts must not share the same account identifier even if they are in different folders.

To use CyberArk AAM Configuration Provider with Usercube you first have to fill the "**appsettings.json**" (Agent side):

1. For the Central Credential Provider (Agentless AAM):

```
89
90  ✓  "CyberArk": {
91      "UseCyberArkSetting": true,
92      "SafeName": "UsercubeSafe",
93      "ApplicationId" : "Usercube-Agent",
94      "Server" : "https://services-uscentral.skytap.com:14144",
95      "CertificateFilePath": "C:/UsercubeDemoCyberArk/cyberark_14144.pfx",
96      "CertificatePassword": "Cyberark1"
97  },
98
```

2. For the Credential Provider with Application Password SDK (Agent Based AAM):

```
88
89      "CyberArk": {
90          "UseCyberArkSetting": true,
91          "SafeName": "UsercubeSafe",
92          "ApplicationId": "Usercube-Agent"
93      },
94
```

Explanation of the settings:

- **UseCyberArkSetting** is the boolean (default: false) to activate the CyberArk Provider for Usercube.
- **SafeName** is the name of the safe containing the accounts used by Usercube.
- **ApplicationId** is the Application ID of the application that can access the safe.
- **Server** is the URL configured for the CyberArk Vault. It is recommended to use HTTPS for security purposes.

- **CertificateFilePath** is the file path to the client certificate. The application is authenticated by the client certificate serial number and machine address. The certificate must be in .pfx format.
- **CertificatePassword** is the password of the .pfx client certificate.

Notes:

- 1) The Server, CertificateFilePath and CertificatePassword attributes are only used with the CyberArk Central Credential Provider (Agentless AAM).
- 2) If you use the Credential Provider with Application Password SDK (Agent Based AAM) then the C/C++ Application Password SDK dynamic link library called CPasswordSDK.dll (on 32 bits systems) or CPasswordSDK64.dll (on 64 bits systems) must be placed in the Runtime folder of Usercube or in the system library folders to use it.
- 3) CertificateFilePath and CertificatePassword are not mandatory and can be used only with the Agentless AAM. Both must be not null otherwise they will be both considered null. They are mandatory if the AIM Web Service require SSL in IIS settings.

As explained earlier in this document, both CyberArk AAM Configuration Providers use the JSON file called “**appsettings.cyberArk.agent.json**”:

```

1  {
2      "InformationSystems": {
3          "ActiveDirectorySetting": {
4              "ADExport": {
5                  "Login": "ADExport_Account_Identifier"
6              }
7          }
8      }
9  }
```

This JSON contains the key “InformationSystems:ActiveDirectorySetting:ADExport:Login” and it is associated with the value “ADExport_Account_Identifier” that is an account identifier in a CyberArk safe.

Thanks to **SafeName** and **ApplicationId** in “**appsettings.json**” and the **account identifier** in “**appsettings.cyberArk.agent.json**”, the following query can be called:

```
var queryStr = $"Safe={SafeName};Object={account identifier}";
```

The query uses **Regexp** so the AAM will search the **account identifier** in every folder of the safe.

With this query and the initial conditions, Usercube is sure to identify accounts in a unique way.

Finally, when Usercube launch the Agent Function (here Export from Active Directory), these two JSON files will be read so the CyberArk Configuration Provider can give the Login in our example or other settings in other cases.

PARTNER CONTACT INFO

Business Contact	Name	DUPONT Charles
	Email	charles.dupont@usercube.com
	Tel	
Technical Contact	Name	MOYERE Nicolas
	Email	nicolas.moyere@usercube.com
	Tel	
Support Contact	Name	Support
	Email	support@usercube.com
	Tel	