EscrowAI User Manual v1.5

EscrowAI provides a secure collaboration environment for conducting analytics on protected data. Data is not shared. Rather, it is encrypted and remains in the environment controlled by the data steward. The analytics algorithm is also encrypted and brought to the data. Computation occurs in a Trusted Execution Environment, within which the data and algorithm are protected from any party by Total Memory Encryption.

Overview of EscrowAI

EscrowAI is a patented zero-trust, confidential computing platform for secure collaboration between algorithm developers and stewards of protected data. With EscrowAI, data remains within the data steward's secure cloud environment and is made available for computation in a hardware-based Trusted Execution Environment (TEE) instance.

EscrowAI is protected by U.S. Patents 11,531,904 and 11,748,633.

EscrowAI's Trusted Execution Environment (TEE)

Encrypted algorithms and encrypted data are brought into the TEE instance.

EscrowAI's TEE enforces the following essential aspects of confidential computing:

- **Data confidentiality:** Unauthorized entities cannot view data while it is in use in the TEE.
- **Data integrity:** Unauthorized entities cannot add, remove, or alter data while it is in use in the TEE.
- **Code integrity:** Unauthorized entities cannot add, remove, or alter code executing in the TEE.
- **Code Confidentiality:** Unauthorized entities cannot view code while it is in use. code while it is in use in the TEE.

An EscrowAI TEE instance runs in the data steward's cloud, enabled by cutting edge confidential computing technology.

Encrypted algorithms are brought into the TEE along with the encrypted data. In the TEE instance's protected memory, the algorithms and data are then decrypted with user-created private keys stored in an HSM key vault. The computation is executed, and only a predetermined output is allowed out of the TEE instance after verification by EscrowAI. After the computation is complete, the TEE instance is decommissioned and terminated.

Confidential Computing technology options

EscrowAI offers the following types of confidential computing technologies. These technologies underlie the virtual machines (VMs) and other secure resources in the TEE instance that EscrowAI manages for you.

Confidential Containers in an Intel Software Guard Extensions (SGX) Enclave

EscrowAI offers Microsoft Azure's confidential containers that are based on Intel® Software Guard Extensions (Intel® SGX). SGX is a set of security-related instruction codes built into some Intel Central Processing Units (CPUs). On a hardware-based Trusted Execution Environment (TEE) instance, application code runs in private regions of memory, called *enclaves*, which are protected from all other processes running at higher privilege levels.

VM Isolated Confidential Containers on Azure Container Instances (CC ACI)
EscrowAI supports Microsoft's Confidential Container on Azure Container Instances (CC ACI). For hardware-based attestation, CC ACI relies on Advanced Micro Devices (AMD) SEV-SNP-enabled Central Processing Units (CPUs).

EscrowAI security advantages

EscrowAI has the following important security features:

Data Stewards retain control of their data.

Data Stewards are responsible for the protection of data under their control. With EscrowAI, data is kept within the Data Steward organization's Azure Cloud environment and never leaves that environment. The Data Steward organization retains full control of their data.

Strong encryption protects data at rest, in transit, and in use.

Data is encrypted using AES-256 encryption. All connections use TLS-encrypted communication.

Data is protected in use.

Data and algorithms achieve high-isolation and memory encryption through hardware-based assurances. Secure attestation ensures the authenticity and integrity of the TEE execution environments.

Algorithm intellectual property is protected.

Algorithms are encrypted by developers and only decrypted within a TEE instance. Algorithm owners retain protection of their intellectual property.

How EscrowAI works

The collaboration summarized below between data stewards and algorithm owners is generalized in EscrowAI workflows and detailed step-by-step in I am a Data Steward and I am an Algorithm Owner.

1. An algorithm owner uploads their encrypted algorithm to EscrowAI. Their algorithm is built into a secure computing container.
2. A data steward curates a data set to meet the algorithm owner's requirements. The data set is encrypted and uploaded to an EscrowAI accessible zone within their secure cloud.
3. To validate the algorithm, EscrowAI initiates an attested, hardware-based Trusted Execution Environment (TEE) in the Data Steward cloud and loads the secure algorithm container and encrypted data into the TEE.
4. With the attested enclave:
    a. The algorithm and data are decrypted in the protected memory.
    b. The algorithm runs.
    c. A confidential report is created containing the algorithm's performance and the general characteristics of the data set.
5. That performance report is the only thing that leaves that secure computing enclave.
6. Finally, all the elements within the enclave are deleted, and the enclave is shut down and decommissioned.

Accounts and user management

Accounts

Accounts are established by BeeKeeperAI Customer Service as the first post-contract work. Setting up the account involves:

- Creating your organization in EscrowAI.
- Creating users identified by your organization.

User Management

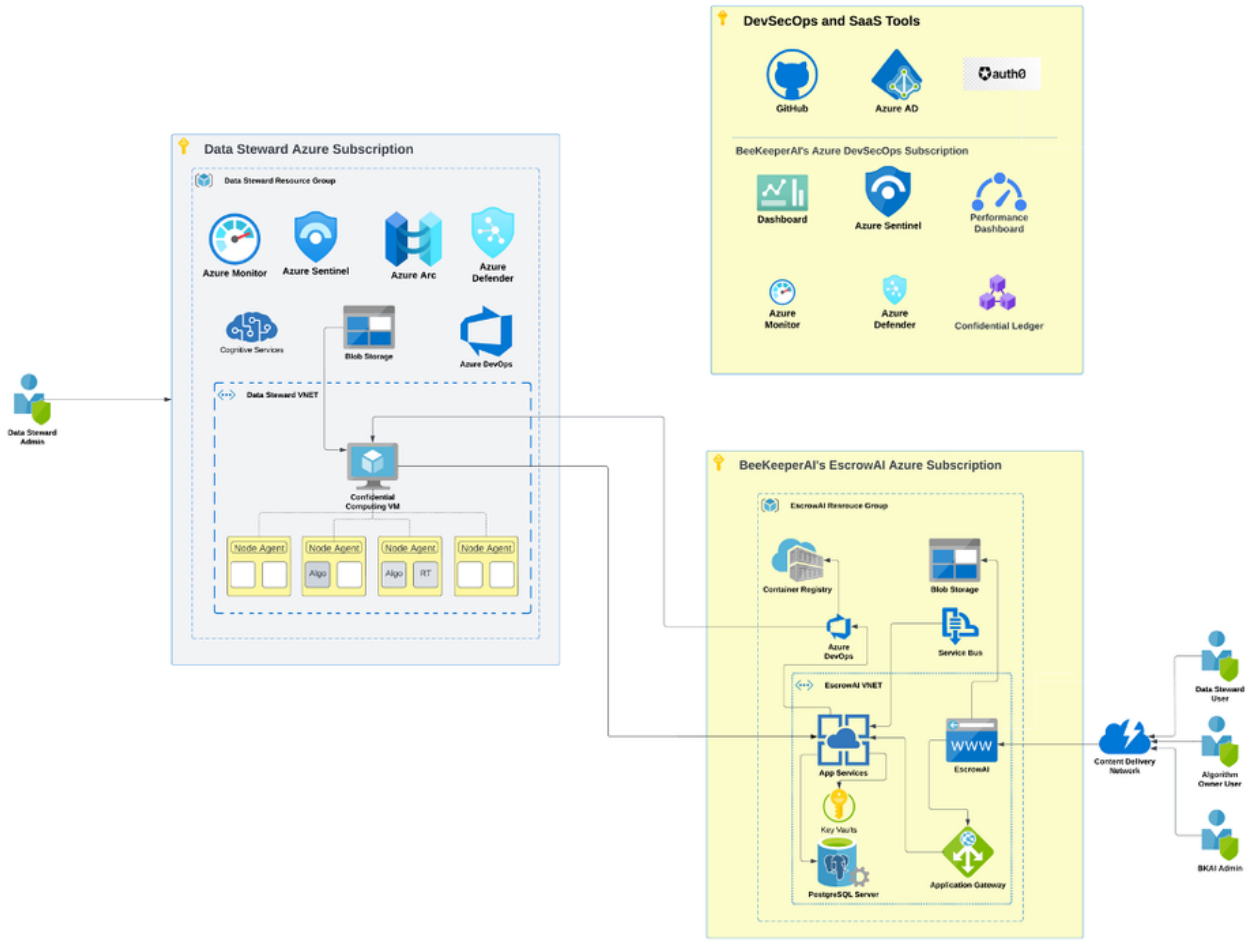
BeeKeeperAI will set up account users in EscrowAI at the direction of the account's point of contact. We can also configure Single Sign On (SSO) authentication to allow accounts direct access control for improved user convenience. EscrowAI supports common standards such as SAML and has out-of-the-box support for more than 15 cloud applications, including Microsoft Azure Active Directory.

Data Steward Azure technical setup

Azure Cloud

A Data Steward (DS) must have the required Azure infrastructure to support data access and deployment of confidential compute nodes within their hardened Azure cloud environment.

The configuration in the Data Steward Azure Subscription, shown below, represents the minimum amount of infrastructure required to operate. Institutions may have additional policies to layer on this configuration such as resources related to VPN and on-premises access to this content.

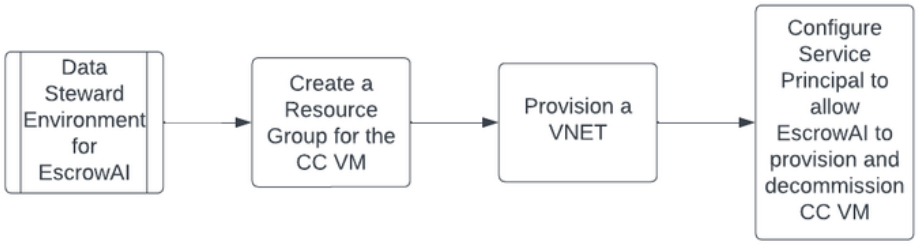


EscrowAI Architecture

Data Steward cloud services provisioning workflow

The following steps are for the DS to prepare the Azure environment for EscrowAI. The steps below are provided for setting up the needed services using the Azure Portal. It is recommended to perform the following steps using Infrastructure as Code (IAC) tools such as Terraform.

The following diagram describes a one-time provisioning of Azure services in the Data Steward's Azure subscription that is required by EscrowAI. These Azure services are needed to allow EscrowAI to provision and to decommission a Trusted Execution Environment in the Data Steward's Azure subscription automatically.



Data Steward one time setup

Terraform to setup the EscrowAI environment

It is recommended to use Terraform - Infrastructure as Code (IaC) to configure the Azure services required by EscrowAI. Terraform is an open-source tool that allows you to define infrastructure as code using a simple, declarative language. You can define the resources you need in a configuration file, and then use Terraform to automate the management of those resources by creating, modifying, and deleting of those resources in a cloud provider like Microsoft Azure.

Create a Virtual Network

Ensure you have an Azure account with an active subscription. If you don't have one, you can create an account for free.

1. Sign in to the Azure portal.
2. Search for and select Virtual networks.
3. On the Virtual networks page, select **Create**.
4. On the **Basics** tab of the Create virtual network screen, enter or select the following information:
   - **Subscription:** Keep the default or select a different subscription.
   - **Resource group:** Select Create new, and then name the resource group <resource group name>.
   - **Virtual network name:** Enter <Vnet name>
   - **Region:** Keep the default or select a different region for the network and all its resources.
5. Select **Next: IP Addresses** at the bottom of the page.
6. On the **IP Addresses** tab, under **IPv4 address space**, select the garbage can icon to remove any address space that already appears, and then enter <VNET address, for example 10.0.0.0/16>.
7. Select **Add subnet**.
8. On the **Add subnet** screen, enter the following information, and then select **Add**:
   - **Subnet name:** default
   - **Subnet address range:** <subnet address, for example 10.0.0.0/24>.

9. Share the following values with the BeekeeperAI team:
   a. VNet Name
   b. VNet Resource Group Name
   c. Subnet Name to launch a VM
   d. Resource Group Name for VM (optional)
   e. Region

Import EscrowAI application into Azure

EscrowAI relies on an application that you securely import into your Azure environment so that EscrowAI can launch confidential containers for you and take care of other housekeeping. After importing, in Azure you grant the application certain permissions to do its function.

Prerequisites

- You must be an Azure administrator to import the software.
- You need to know your Azure tenant ID. This value is indicated in these steps as `<your_azure_tenant_id>`.
- You need to know your Azure subscription ID. This value is indicated in these steps as `<your_azure_subscription_id>`.
- Decide on a descriptive Azure role name you will assign in Azure to the EscrowAI application; for example, “EscrowAI”.

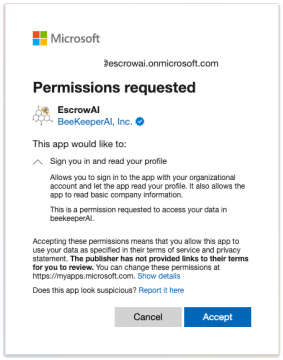
URL for EscrowAI application

The following URL is the EscrowAI application for import into Azure. Notice the `<your_azure_tenant_id>` variable you must supply:

`https://login.microsoftonline.com/<your_azure_tenant_id>/oauth2/authorize?client_id=28dbee55-66fa-4adf-872f-415495968c7e&response_type=code&redirect_uri=https%3A%2F%2Fwww.microsoft.com%2F`

General steps

1. As an Azure administrator, in your web browser, open the URL shown in URL for EscrowAI application.
   Be sure to specify your own Azure tenant ID for the variable `<your_azure_tenant_id>`.
2. At the prompt from Microsoft, accept the requested permissions.



3. After the import has succeeded, create a custom role for EscrowAI to launch containers. Create a JSON file with the following content.
   Be sure to supply your Azure subscription ID for the variable `<your_azure_subscription_id>`.

```
{
    "properties": {
        "roleName": "escrow-vm-launch",
        "description": "Escrow VM Launch Custom Role",
        "assignableScopes": [
            "/subscriptions/<your_azure_subscription_id>"
        ],
        "permissions": [
```

```
            {
                "actions": [
                    "Microsoft.Network/virtualNetworks/subnets/read",
                    "Microsoft.Network/networkSecurityGroups/read",
                    "Microsoft.Network/networkSecurityGroups/write",
                    "Microsoft.Network/networkSecurityGroups/delete",
                    "Microsoft.ContainerInstance/containerGroups/*",
                    "Microsoft.Resources/subscriptions/resourceGroups/read",
                    "Microsoft.Resources/subscriptions/resourceGroups/write",
                    "Microsoft.Resources/subscriptions/resourceGroups/delete",
                    "Microsoft.Resources/deployments/*"
                ],
                "notActions": [],
                "dataActions": [],
                "notDataActions": []
            }
        ]
    }
}
```

4. Create a custom role under **Subscriptions > IAM > Add Custom Role > JSON**. Copy the above JSON content, and then **Review** and **Create**.
5. Assign the Virtual Machine Contributor role (Built-In Role) and the custom role to the EscrowAI application.
   a. **Subscription > IAM > Add Role Assignment**. Under **Job Function** roles, select **Virtual Machine Contributor** role **>** select members **>** select the EscrowAI application, then **Review** and **Assign**.
   b. Repeat same steps above to assign the custom role created in step 4 to the EscrowAI application.

Create a Storage Account

1. Sign in to the Azure portal.
2. In the Azure portal, click **Create a resource**.
3. Search for **Storage account** in the search bar and select it from the list of options.
4. Click **Create** to start creating a new storage account.
5. In the **Basics** tab, select your subscription, create a new resource group or select an existing one, and give your storage account a unique name.
6. Select the region for the storage account.
7. Choose the performance tier and Redundancy.
8. Click **Next: Advanced** to move to the next tab.
9. Choose your desired settings for advanced options like virtual networks, data protection, and data transfer.
10. Click **Review + create** to review your storage account settings.
11. After reviewing your settings, click **Create** to create your storage account.

Set up and manage Blob Storage

The setup and management of Blob storage is repeated for each project. A project specific Container is created, and encrypted project data is uploaded as a blob into this Container. The workflows in these instructions assume the Storage Account and project Container are configured at the start of any project.

See Upload encrypted dataset to Azure Blob container.

It is recommended that the data site designate a role or individual that creates the necessary project containers and who can grant permission the project-specific data analyst 1) for uploading the encrypted data set to blob storage and 2) for creating the SAS URL.

Links to Microsoft documentation

- Create a Linux virtual machine in the Azure portal
- Manage resource groups in the Azure portal
- Use the Azure portal to create a virtual network
- Create a Windows VM in the Azure portal
- Create a storage account
- Upload, download, and list blobs in the Azure portal
- Use a managed identity to access Azure Storage - Linux
- Configure Azure Storage Firewalls and Virtual Networks

Project creation

A project is based on an agreed statement of work between collaborators. BeeKeeperAI creates the project in EscrowAI. The Project:

- Brings together the data set(s) and algorithm(s).
- Assigns roles to each organization (Algorithm Owner or Data Steward).
- Adds designated users from each organization to the project.
- Assigns the desired Confidential Computing technology platform to the project.

> The Confidential Computing technology platform cannot be changed within the project once it is created. If a different technology is desired after Project creation, a new Project must be created.

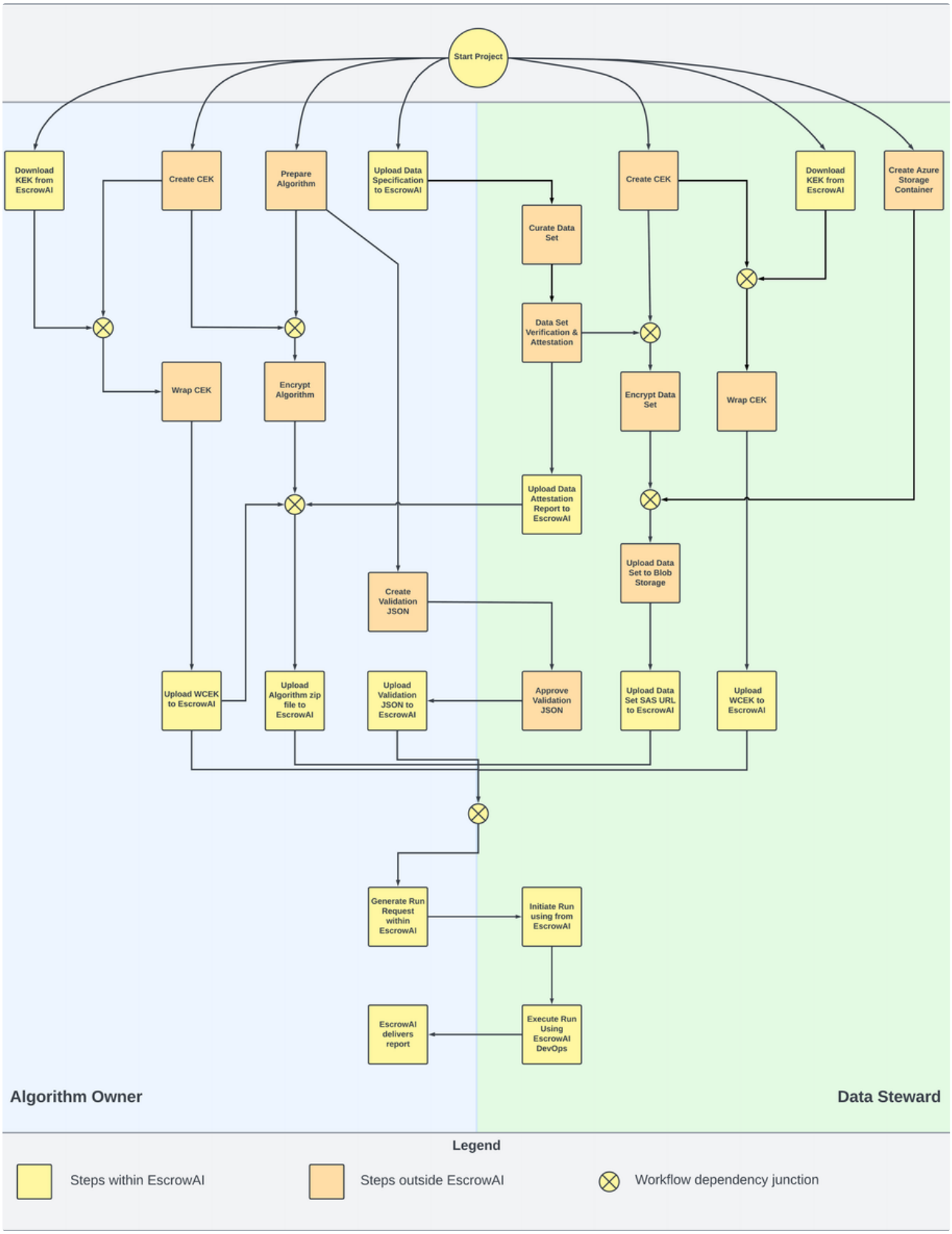
The new project appears in the Home Page of the assigned user, with the name given by the project owner and with placeholders for the project artifacts.

EscrowAI workflows

Overall EscrowAI workflow

This diagram illustrates the collaboration workflow within an EscrowAI project. As the graphic indicates, most tasks in the project workflow are completed in EscrowAI. Preparation of project artifacts to upload to EscrowAI takes place on your own local computers. In this workflow diagram the differentiation between the in- and out-of-EscrowAI application steps are color coded. The diagram also shows dependencies between workflow steps using a workflow dependency junction. All of the workflow steps connected into the junction must be completed before the step after the junction can be completed.

The user manual is organized by role workflows and the common cryptography workflow.



Data Steward's collaborative tasks

See the following section for the Data Steward's collaborative tasks: I am a Data Steward.

Algorithm Owner’s collaborative tasks

See the following section for the Algorithm Owner’s collaborative tasks: I am an Algorithm Owner.

Common encryption steps

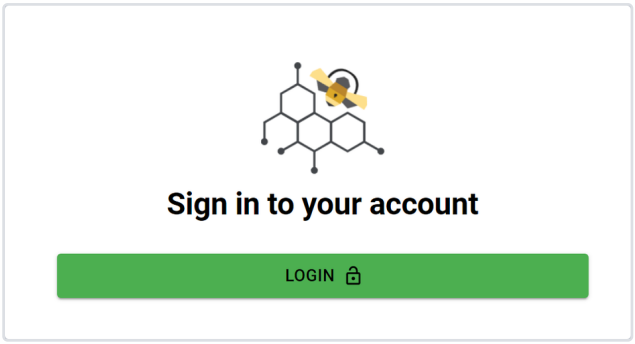
Data stewards and algorithm owners both independently utilize EscrowAI’s encryption tool complete the cryptography workflow.

- EscrowAI cryptography
- Common encryption steps

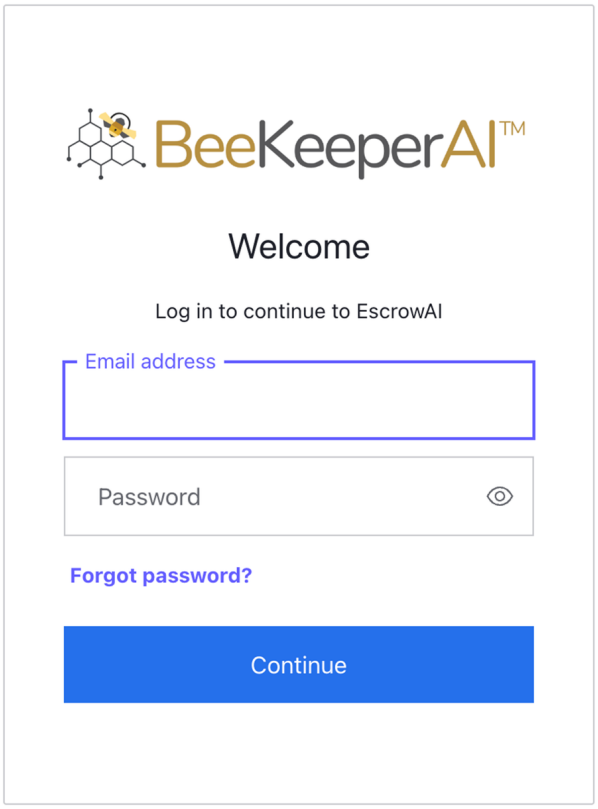
Login to EscrowAI

Native login

1. Open the login page at EscrowAI.



2. Enter your username and password in the appropriate field.



3. Click on the "Continue" button.

4. If the login information you entered is correct, you will be redirected to your EscrowAI home page.

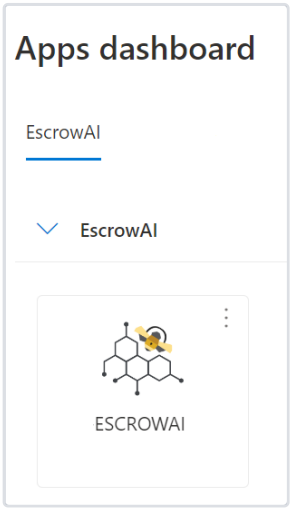
During your first login, you have the option to reset your password by clicking the **Forgot password?** button. To reset your password, follow the Password Reset instructions below.

EscrowAI requires a certificate from your browser to establish a secure connection. Depending on your operating system and browser, you may experience a pop-up message requiring you to select an appropriate certificate to submit to EscrowAI. Choose the appropriate certificate and submit. Occasionally, the certificate you choose may not work. Select a different certificate and resubmit. Repeat as needed.

If you are unable to login, make sure that you have entered your login credentials correctly, and that your internet connection is stable. If you're still having trouble logging in, you can try resetting your password.

Single Sign-On (SSO)

Single Sign On is an authentication scheme that allows a user to log into a variety of independent software systems with a single ID. EscrowAI supports a variety of Service and Identity Provider initiated SSO options. An example using a SSO integration with Azure Active Directory is shown below. Your organization's implementation may be different.



1. Go to My Apps.
2. Click on the EscrowAI application.

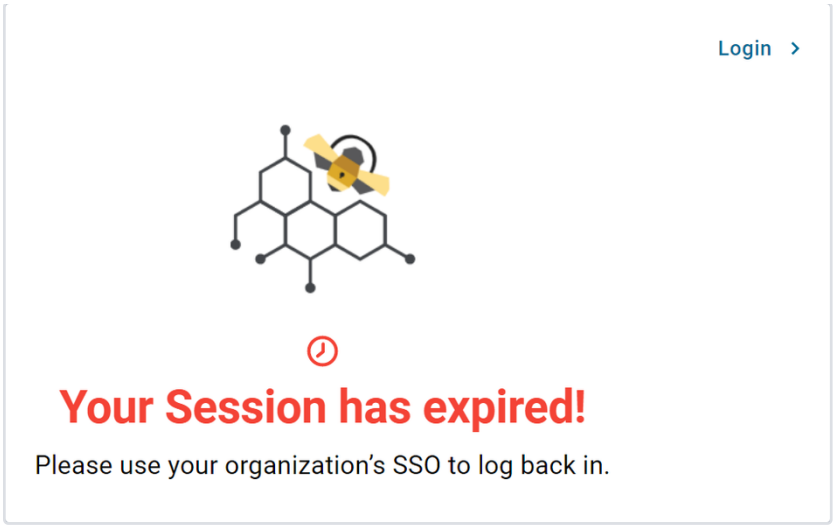
After initiating the Single Sign-On (SSO) process, the EscrowAI application will automatically authenticate your credentials and proceed to log you in. Once logged in, you will be directed to the Home page.

The specific steps for SSO login can vary depending on the organization and the Identity and Access Management platform being used. If you're having trouble logging in using SSO, contact your organization's IT department or the support team for assistance.

Session timeout

EscrowAI is configured to automatically log out a user after a predefined period of inactivity. This is a security precaution that helps to prevent unauthorized access to the user's account or sensitive information accessible within EscrowAI.

The user session is timed out after 12 minutes of inactivity.

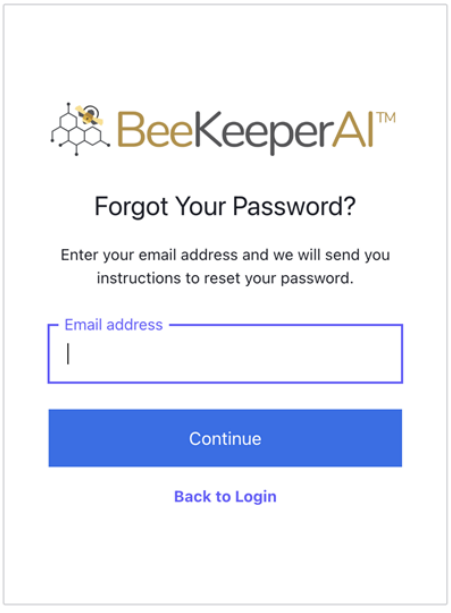


After a session time out, you must log in again using the method configured for your organization. You will be placed at your Home Page and can navigate to the Project Page necessary to resume work.

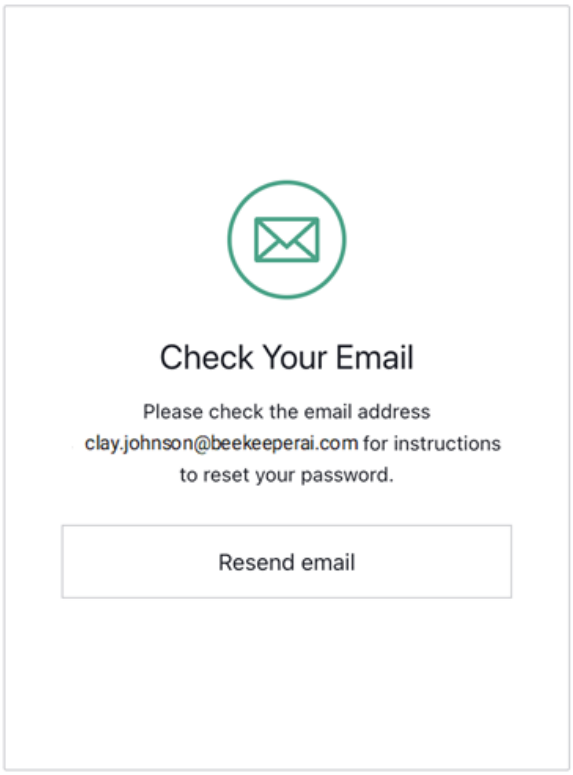
Password Reset

A password reset is required during first login, if you are not using Single Sign-On.

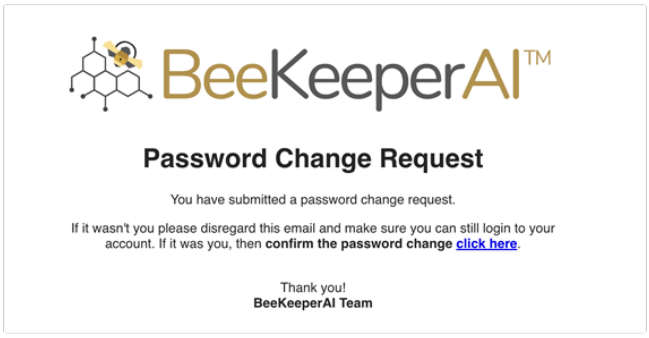
1. From the Login screen, click the **Forgot password?** link.
2. Follow the on-screen prompts to initiate the password reset process.



3. Check your email inbox for a verification message from EscrowAI.



4. Click the verification link provided in the email.



5. Follow the instructions on the password reset page to create a new password.

6. Log in using your newly created password.

EscrowAI home page

Home page

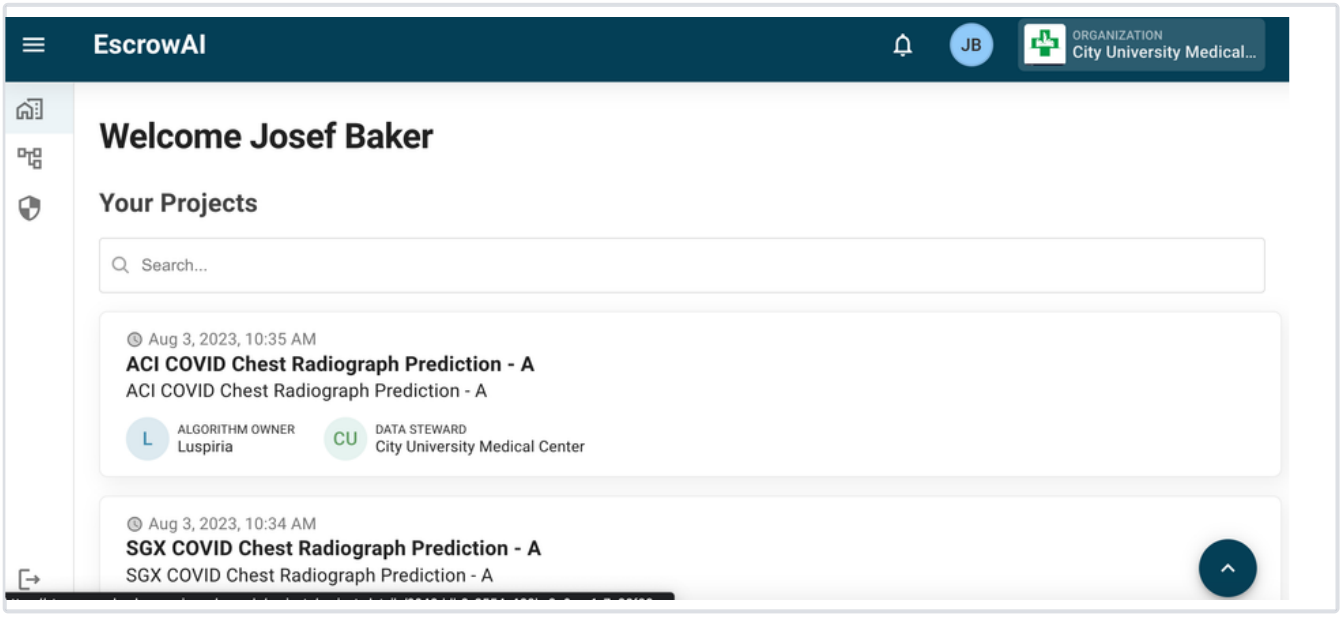
The EscrowAI home page is the user's initial landing page after logging in. It displays the list of Projects that have been assigned to the user by their organization.

The user's organization name and logo and the user's initials icon are displayed in the page heading banner.

The home page displays the list of Projects as cards with the most recent projects at the top. Each Project card includes descriptive metadata:

- Project name
- Project description
- Project creation date
- Project collaborator organizations and roles
- User initials icons

Clicking on a Project card will open the EscrowAI project page with the details of the collaboration.



Search

You can search for content within Projects from the home page using keywords. The search query returns a content list based on the metadata in the following fields.

- Project names and descriptions.
- Dataset names, descriptions, version names, and version descriptions.
- Algorithm names, descriptions, version tags.
- Project artifacts names, descriptions, version tags, and version descriptions.
- Run configuration names and descriptions.

Home page elements

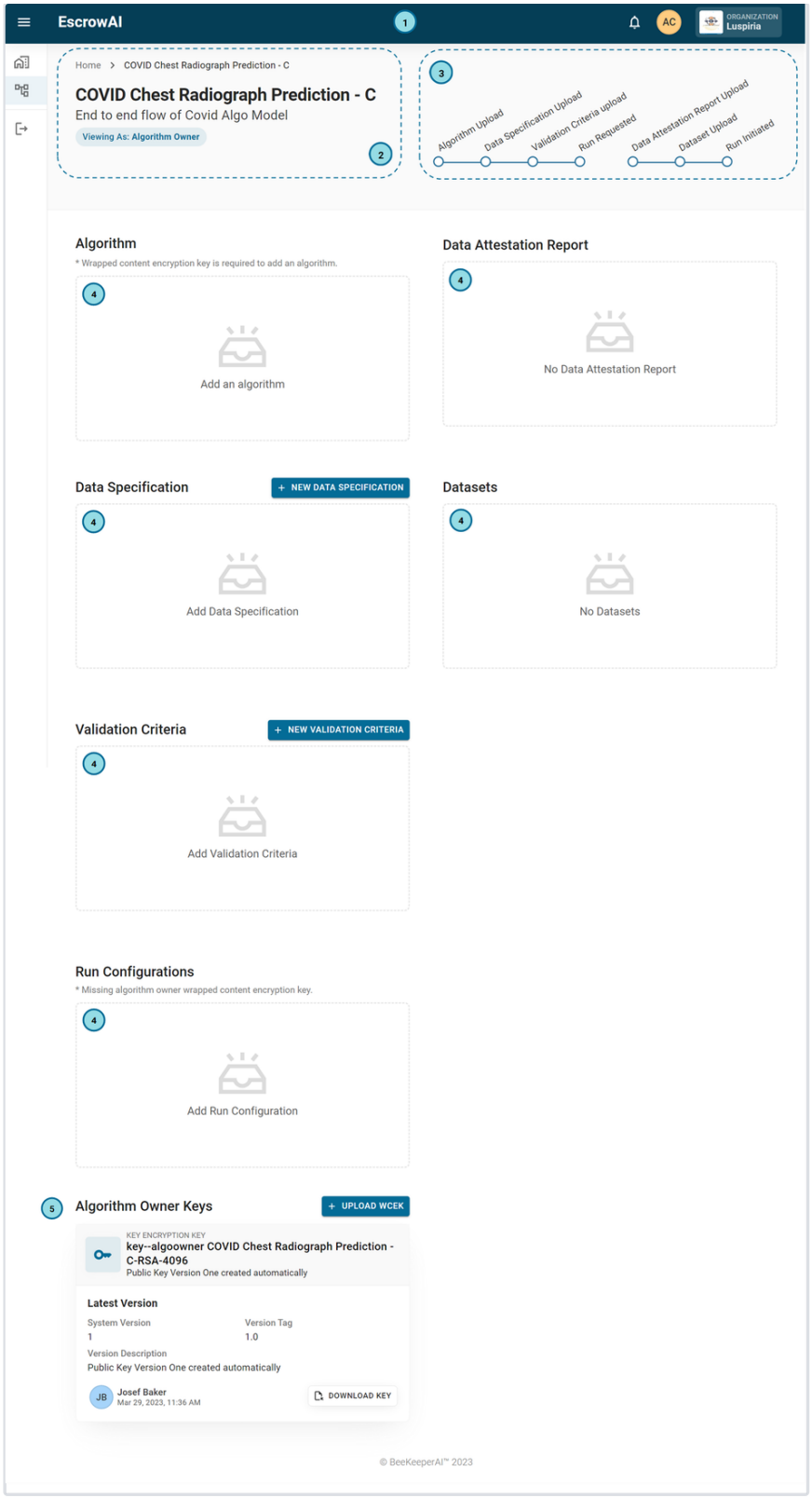
Element	Description
	System Notification icon. System notifications related to runs are displayed by clicking this icon. New notifications are indicated by a red dot on the icon.
	User initials icon. This element is used to identify the user to other collaborators throughout EscrowAI.
	Company logo. This is specific to the user that is logged in.
	Expand bar. Clicking this icon will display the Home, Project, and Log Out icons.
	Home icon. This navigates to the user's home page.
	Project icon. This navigates to the user's project listing.
	EscrowAI encryption tool for encrypting datasets and algorithms. For details, see EscrowAI encryption tool.
	On the lower right of the EscrowAI home page, the quick actions icon includes multiple functionalities: • a link to EscrowAI help and other information resources. For details, see EscrowAI Help Center. • chat between the project's data stewards and algorithm owners. For details, see Intra-project chat between users.
	Log Out icon. This allows the user to logout of the EscrowAI site.
App Version	EscrowAI release version number

EscrowAI project page

The EscrowAI Project Page is a central hub for all information related to your project. It provides an overview of the project's goals, progress, and key milestones, including the project name, a brief description, and relevant files. The workflow tracker, located on the upper right-hand side of the page, displays the progress of completed and incomplete tasks. In the center of the page, project cards show a title and the completion status of each step (e.g., 'Data Specification Uploaded,' 'Algorithm Added,' etc.). The Project Page is a critical resource for the collaboration of the Data Steward and Algorithm Owner to stay informed and on track toward successful completion. Permission to access or change artifacts is limited by role. For instance, the Algorithm Owner can add and change the algorithm, but the Data Steward access is limited to the information in the artifact card.

Project page elements

1	EscrowAI Title Bar Contains the active user's initials, the organization's identification, and the notification icon.
2	Project identification section. Contains the project name, description, user's role, and navigation breadcrumbs.
3	Project workflow milestones. As project artifacts are completed the ball on the timeline is filled in and version information is added.
4	Project artifact cards. These hold the named artifact, meta data, and version information.
5	Cryptographic key card. Keys for decrypting the data or algorithm content are uploaded here.



Collaboration

The project page is a collaboration environment between the project organizations and users. The collaboration workflows, with interactions and dependencies, are shown in the EscrowAI workflows page. Further detail is provided in the individual pages that describe the workflows for each role.

Data Steward (DS) responsibilities

Within the project page, the DS has tasks described in I am a Data Steward.

Algorithm Owner (AO) responsibilities

Within the project page, the AO has tasks described in I am an Algorithm Owner.

Intra-project chat between users

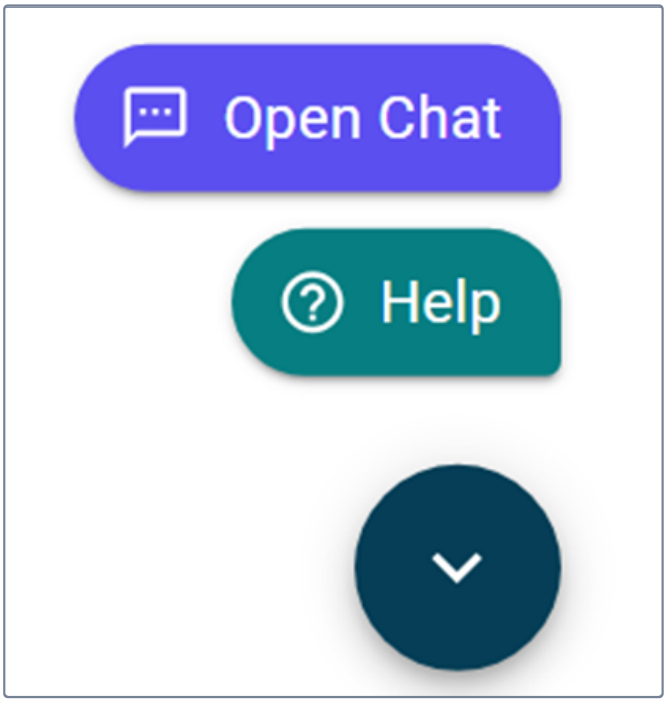
To enhance your collaboration, you can communicate with your project counterparts directly from within EscrowAI. Chat is limited to intra-project communications.

To start or view a chat:

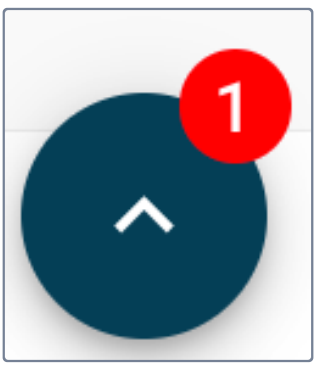
1. Go to the desired project page.
2. The chat widget is initiated by clicking on the **Quick actions** icon in the lower right corner.
3. Select **Open Chat**.

> A red bubble with number indicates that there are unread messages in Chat.

Quick actions icon



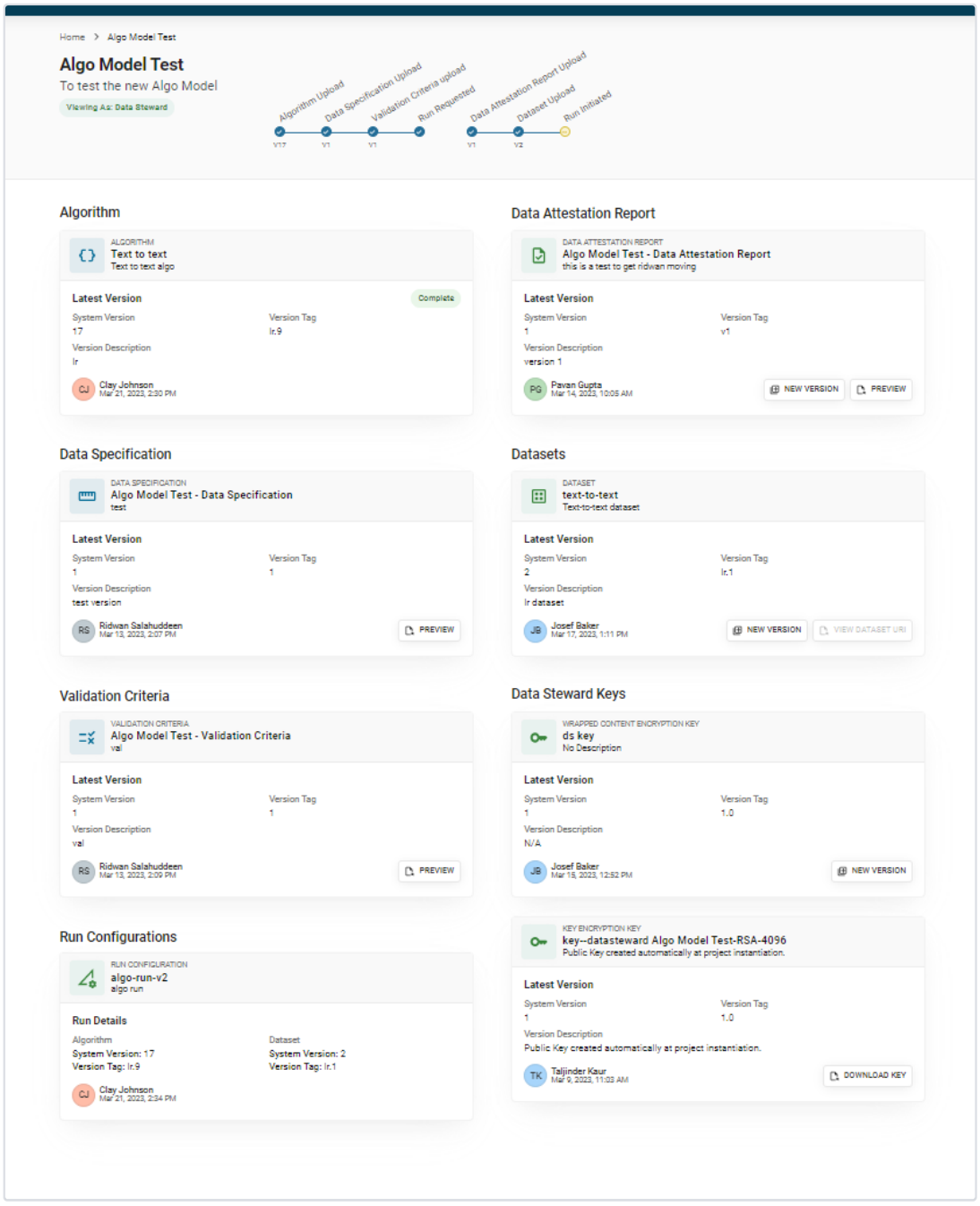
Expanded action menu



One unread message

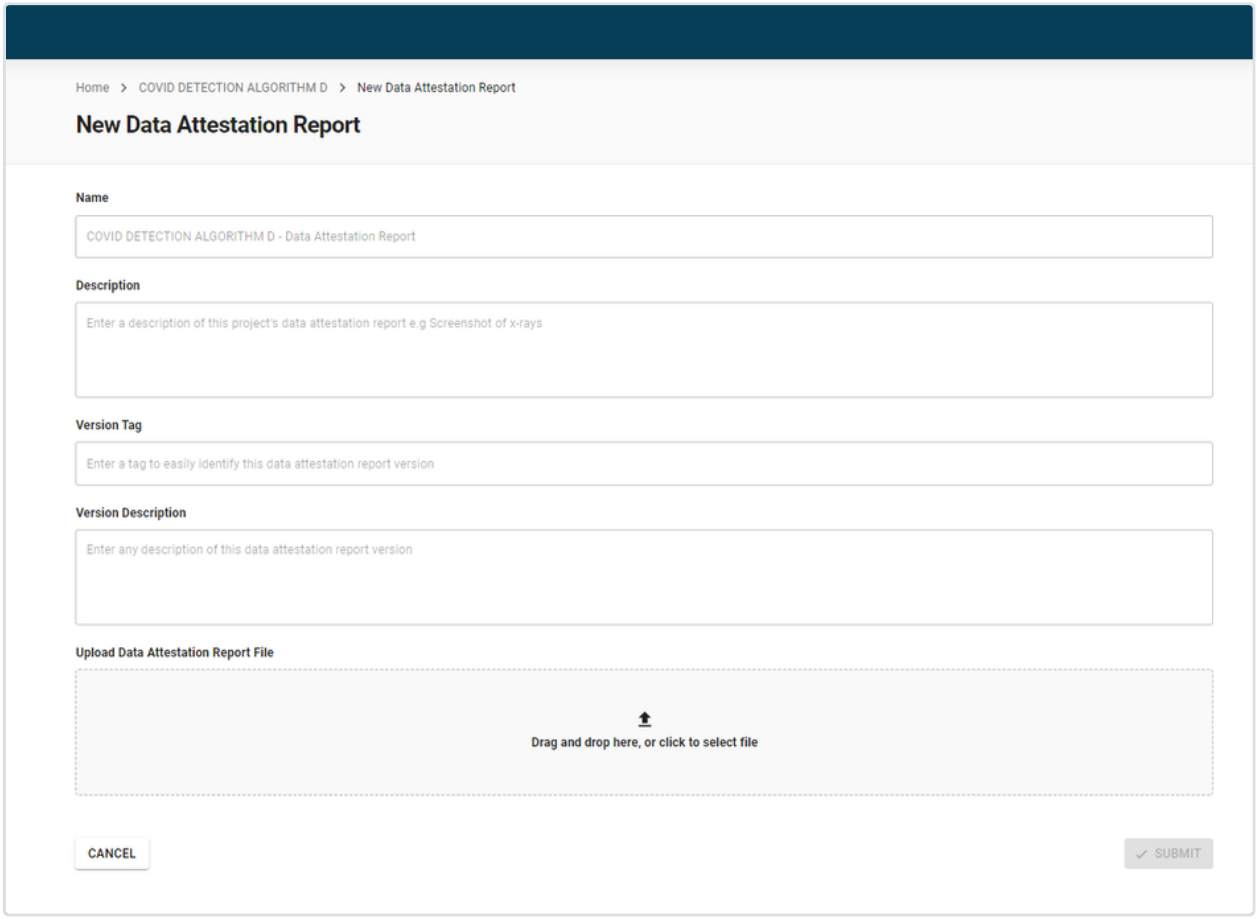
The populated project page

An example fully populated project page is shown below. All project artifacts have been loaded into EscrowAI, and the project is ready for a run of the algorithm against the data set.



Common elements of artifact cards

The artifact cards of the EscrowAI Project page contain common meta data elements: the Name field, Description field, Version Tag field, and Version Description fields display within each card. They are used to provide a description of what is different or new about this project. There is also an area where you can upload the file associated with the artifact. These elements are repeated on each Project page, and users are expected to fill in the appropriate information for each field.



Field	Description
Name	Name of the artifact. This is auto-populated by EscrowAI.
Description	Further describe the artifact as needed.
Version Tag	The user's version for the artifact. This can be any form of alphanumeric version tagging. EscrowAI creates fully unique and sequential version numbers for each artifact to ensure relational consistency and accurate record keeping. This is shown as the "System Version" associated with each artifact.
Version Description	A description of the version. This is useful for adding details about the uniqueness of the version. For example: • "Initial version" • "Changes in v2 include corrections to the gender classification in the data set to include genetic gender only."

	• “v2.0.3 of the algorithm corrects a defect that interrupted processing of images with less than 800x600 pixel resolution.”
Upload File	This is a place to drag and drop a file, or to open your file navigator to select and upload a file.

EscrowAI assigns a system version to each artifact version, in addition to the version tag entered by the user. This ensures versions are uniquely numbered in the system.

EscrowAI cryptography

Key manager

Encryption is an intrinsic part of confidential computing. Data and algorithms are encrypted at rest, in transit, and in process. EscrowAI uses both customer-centric and machine-centric encryption workflows to ensure absolute confidentiality and zero trust.

Backing up the platform is a data security management system that performs a number of functions. It is a FIPS 140-2 Level 3 Hardware Security Module (HSM) key manager that is outside of the management control of EscrowAI or BeeKeeperAI, and is not accessible by any user. The key vault generates and stores all the keys and secrets used within EscrowAI, with the exception of:

- The customer's Content Encryption Key (CEK). This is a data encryption key that they create with the Escrow AI encryption tool or with their own key management solution. It is stored in their own key manager.
- The Wrapped Content Encryption Key (WCEK) that the customer uploads to EscrowAI as part of the project. Because this is ciphertext, it can be stored in the EscrowAI platform.

EscrowAI encryption is based on certificates and public/private key encryption with TLS-encrypted communication and AES-256 ciphers.

Types of keys

Your use of EscrowAI relies on the following types of key material.

Key	Description
Key Encryption Key (KEK)	The EscrowAI generates a unique asynchronous key pair for each role within a project. The KEK is the public half of the asynchronous key pair. In a public-key encryption system, anyone with a public key can encrypt data yielding a ciphertext, but only those with the corresponding private key can decrypt the ciphertext to obtain the original data. The KEK is the cryptographic key that is used for the encryption of the CEK ("wrapping") to provide confidentiality and protection for that key, allowing it to be sent to EscrowAI as ciphertext. The KEK is available for download from the project page. The private half of this key pair (the half that can decrypt) is retained in the key vault after generation and is only available for decrypting the wrapped CEK within an attested Trusted Execution Environment initiated from the associated project. The private key is only released to the corresponding project confidential container operating in the TEE after an attestation process enables the TEE to cryptographically prove that: • A genuine TEE is running the code, built and signed by the user, unmodified. • The TEE platform is secure and running the necessary microcode updates at runtime. • The configuration requirements of the TEE are met by the hardware and software.
Content Encryption Key (CEK)	A CEK is a private, synchronous data-encryption key used to encrypt **and** decrypt data. You create your CEKs using EscrowAI's encryption tool or your organization's key manager. The CEK is used to encrypt a data set and the intellectual property within the algorithm.

Wrapped Content Encryption Key (WCEK)	A WCEK is a CEK that has been encrypted (“wrapped”) by a KEK. The WCEK is ciphertext (encoded information). You create your WCEK using EscrowAI’s encryption tool.

EscrowAI encryption tool

EscrowAI provides an encryption tool to facilitate the various user cryptography operations.

- Generate a Content Encryption Key
- Encrypt dataset
- Upload encrypted dataset to Azure Blob container
- Encrypt algorithm files
- Generate a Wrapped Content Encryption Key (WCEK)

The encryption tool runs on your local computer through the web browser using your local computer’s resources. This is indicated by a message from the encryption tool that it is running offline.

EscrowAI’s encryption tool is based on the well-known Web Crypto API. However, if you like, you can use a different key generator to create a Content Encryption Key, as long as it is a 32 byte AES-256-cipher-based key.

Sequence of key use

1. A project is created in EscrowAI. An asynchronous key pair is generated for each role in the project. The public half (KEK) is available to download from the project page.
2. Each role generates a private Content Encryption Key (CEK) using either the EscrowAI encryption tool or their organization’s key manager.
3. Use the CEK generated in (2) to encrypt your data or algorithm IP using the EscrowAI encryption tool.
4. Use the EscrowAI encryption tool to wrap the CEK created in (2) with the KEK downloaded in (1) to create your Wrapped Content Encryption Key (WCEK).
5. Upload the WCEK to the project page.

Make sure that you wrap the exact CEK that you used to encrypt your data or algorithm. An incorrect key will cause a decryption error in TEE during a run.

The WCEK must be uploaded before either the data set or algorithm can be uploaded.

Keep your CEKs secure!

You should secure your private Content Encryption Key (CEK). Guard them against unwanted access. Consider using a password manager or your organizations key manager.

Key material naming convention

EscrowAI’s encryption key file names follow the naming convention detailed here, where:

- `<project-name>` is the hyphenated title of your project
- `*` is a system-generated random string.

Type of Key	File Naming Convention
Key Encryption Key (KEK)	The `.der` suffix:

	• For Data Steward: `<project-name>*-ds.der` • For Algorithm Owner: `<project-name>*-ao.der`
Content Encryption Key (CEK)	`cek_*.key`
Wrapped Content Encryption Key (WCEK)	`wcek_*.key.bkenc`

Common encryption steps

EscrowAI's encryption tool is used to create the keys needed for encrypting a dataset or algorithm. The encryption tool is accessible by clicking on the shield icon in the left-hand toolbar.

EscrowAI encryption begins with these same steps by both the Data Steward and the Algorithm Owner.

Role-specific steps are included in these sections:

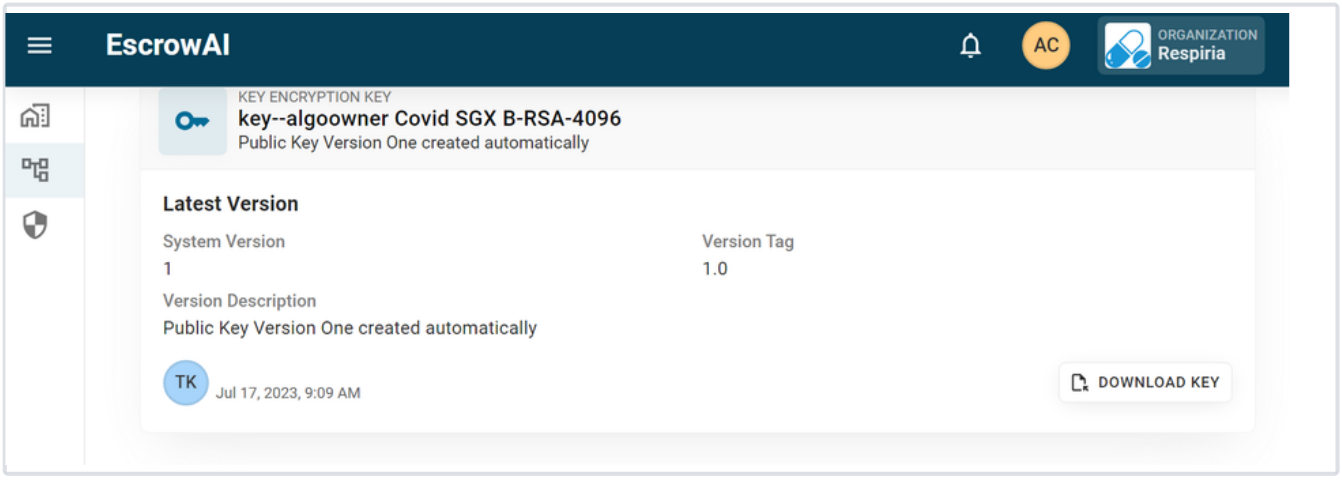
- I am a Data Steward
- I am an Algorithm Owner

Download Key Encryption Key from EscrowAI

Algorithm and Data content is encrypted by the user prior to submission to EscrowAI. In order to decrypt this data, the enclave must have access to the Content Encryption Key (CEK). To securely transfer the CEK to EscrowAI, the platform provides a separate encryption workflow that uses a Key Encryption Key (KEK) to encrypt or wrap the CEK.

Download the project's KEK

Download the Key Encryption Key (KEK) by clicking **DOWNLOAD KEY**. The KEK will be saved to your computer. Use the KEK to encrypt the Content Encryption Key (CEK)



Generate a Content Encryption Key (CEK)

Overview

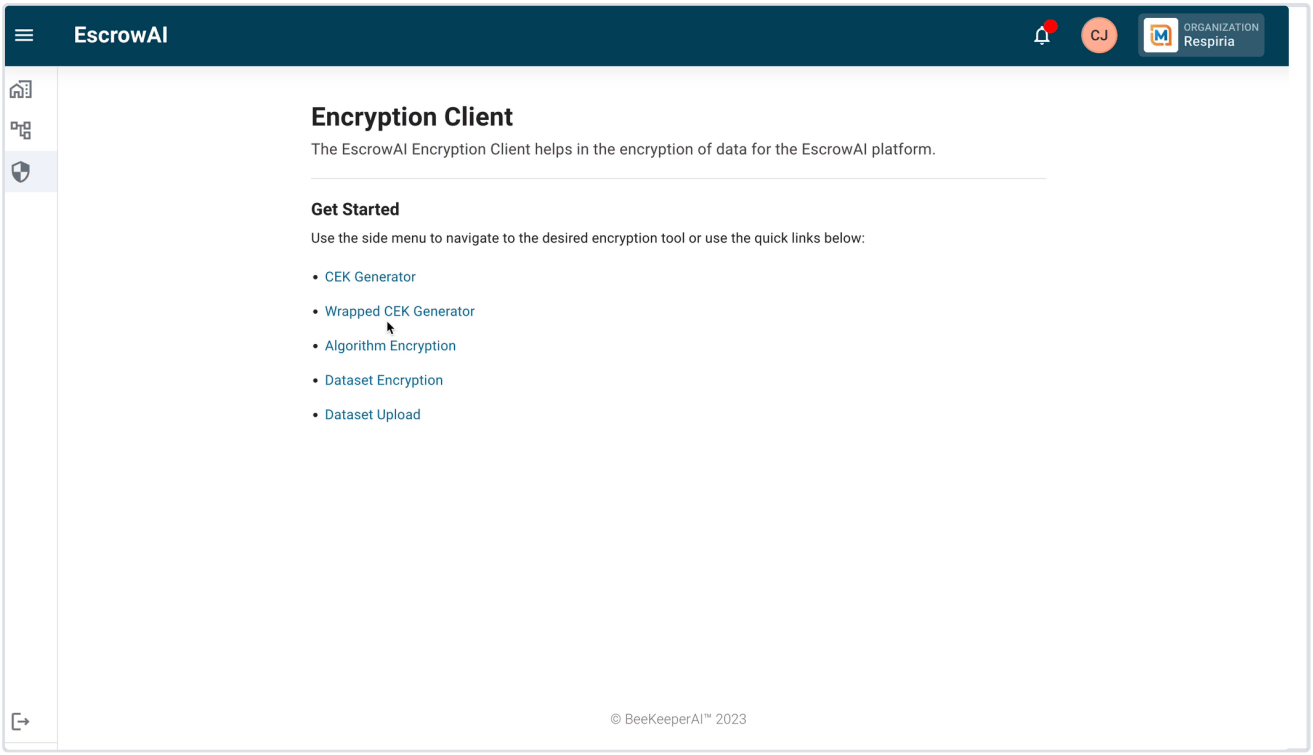
A Content Encryption Key (CEK) is used to encrypt the data and the portions of the algorithm that are confidential (the “Content”). This CEK is used to decrypt data within an attested Trusted Execution Environment associated with the project. The EscrowAI Encryption Client generates an AES-256 symmetric CEK using cryptographically robust randomness. This article explains the process of generating a CEK using the EscrowAI client.

> The end user may opt to use a different key generator to create a Content Encryption Key, as long the key is 32 bytes (256 bits) long.

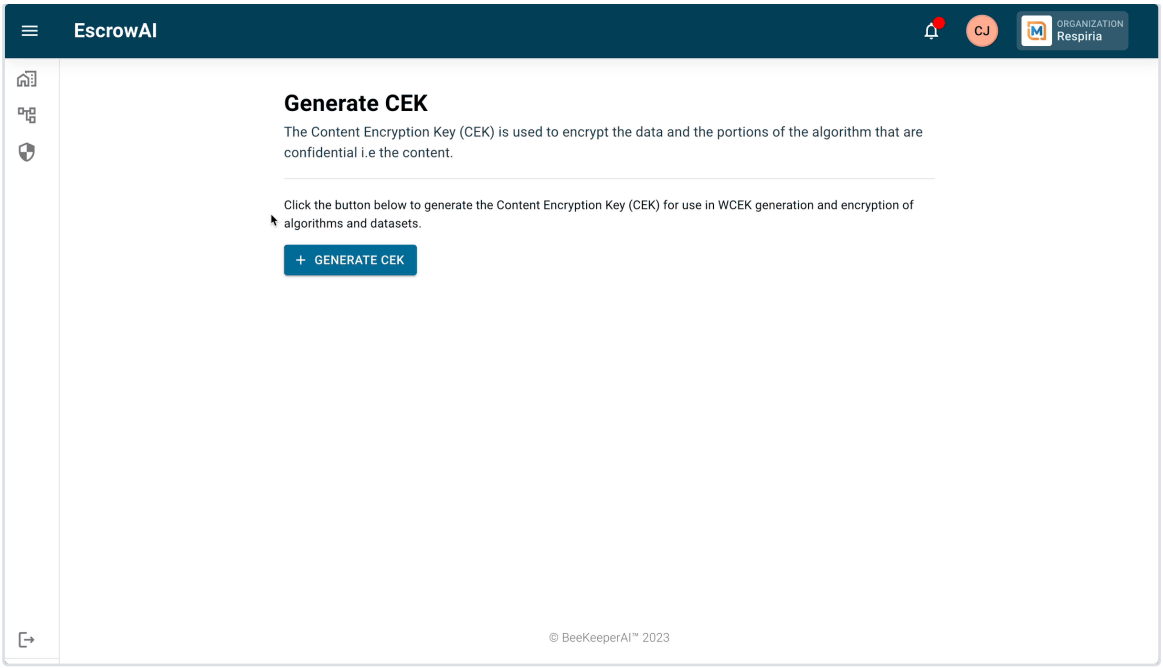
Steps

To generate a CEK with the encryption client:

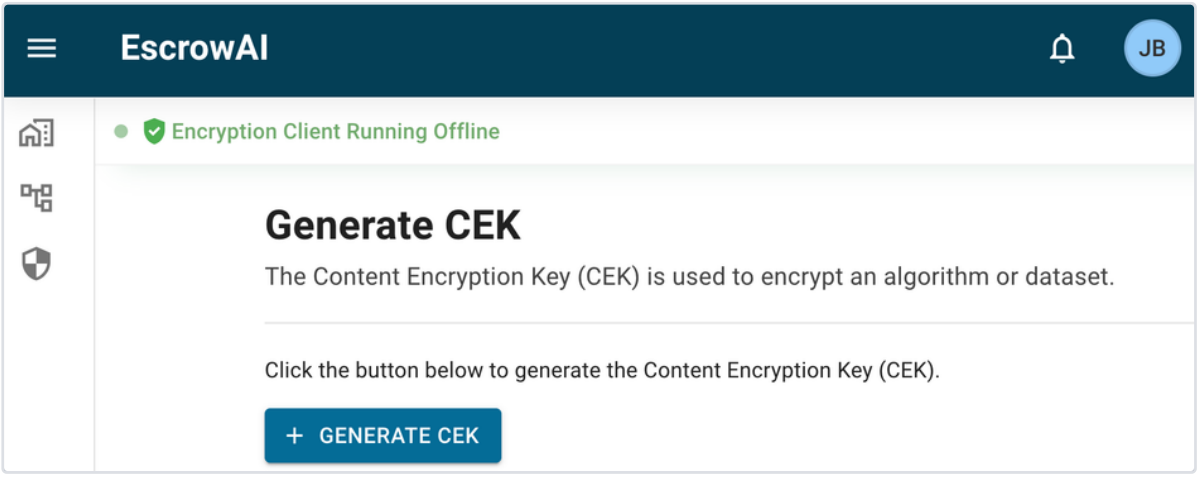
1. Select CEK Generator from the home screen or from the side menu.



2. Click **Generate CEK**. The CEK is automatically saved to your downloads folder.



3. After your CEK has been downloaded, you are returned to the **Generate CEK** page in case you want another CEK, as shown below.



Generate a Wrapped Content Encryption Key (WCEK)

Overview

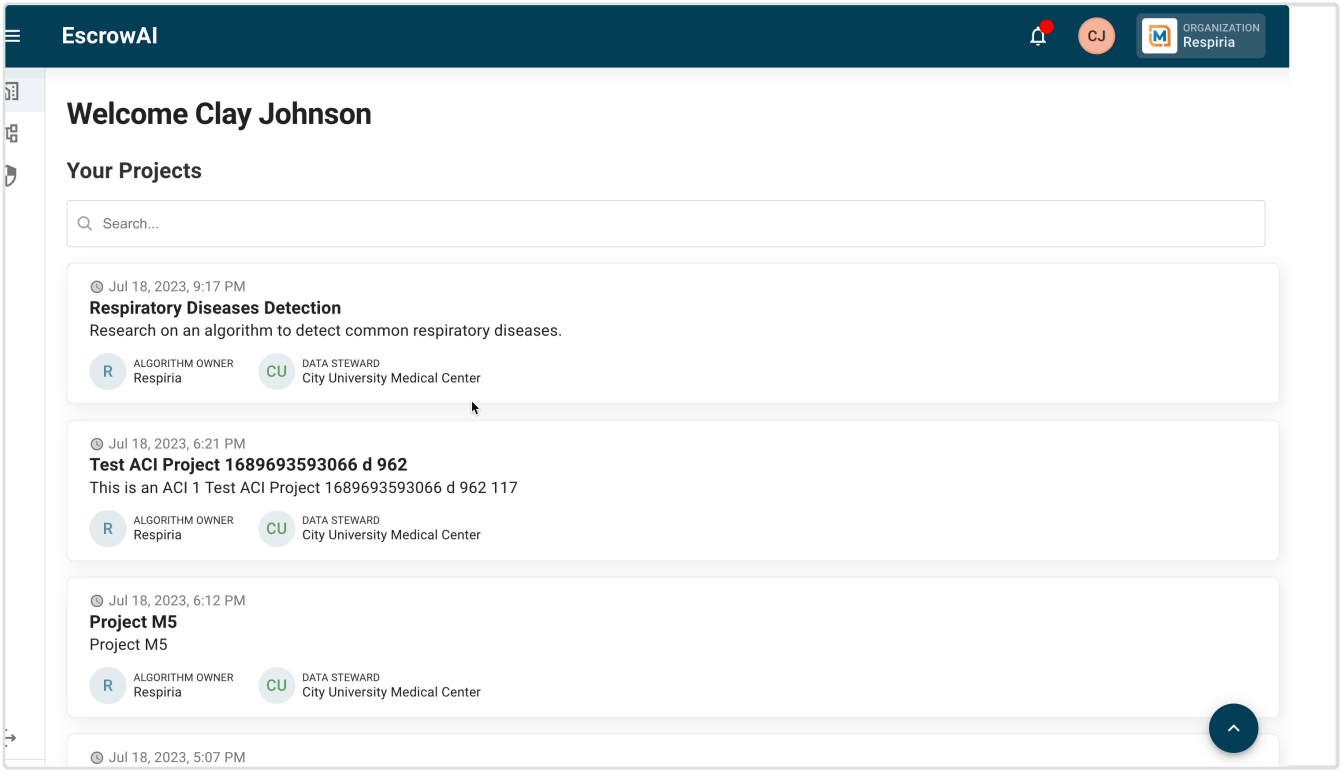
The Content Encryption Key (CEK) must be encrypted in order to be uploaded and stored in EscrowAI securely. This is accomplished using the public half of an RSA key from the EscrowAI application - the Key Encryption Key (KEK) - to encrypt or “wrap” the CEK you previously created.

This will generate a new WCEK that you will upload to EscrowAI. This WCEK is a cyphertext version of your CEK that will only be unwrapped within an EscrowAI Trusted Execution Environment associated with the project.

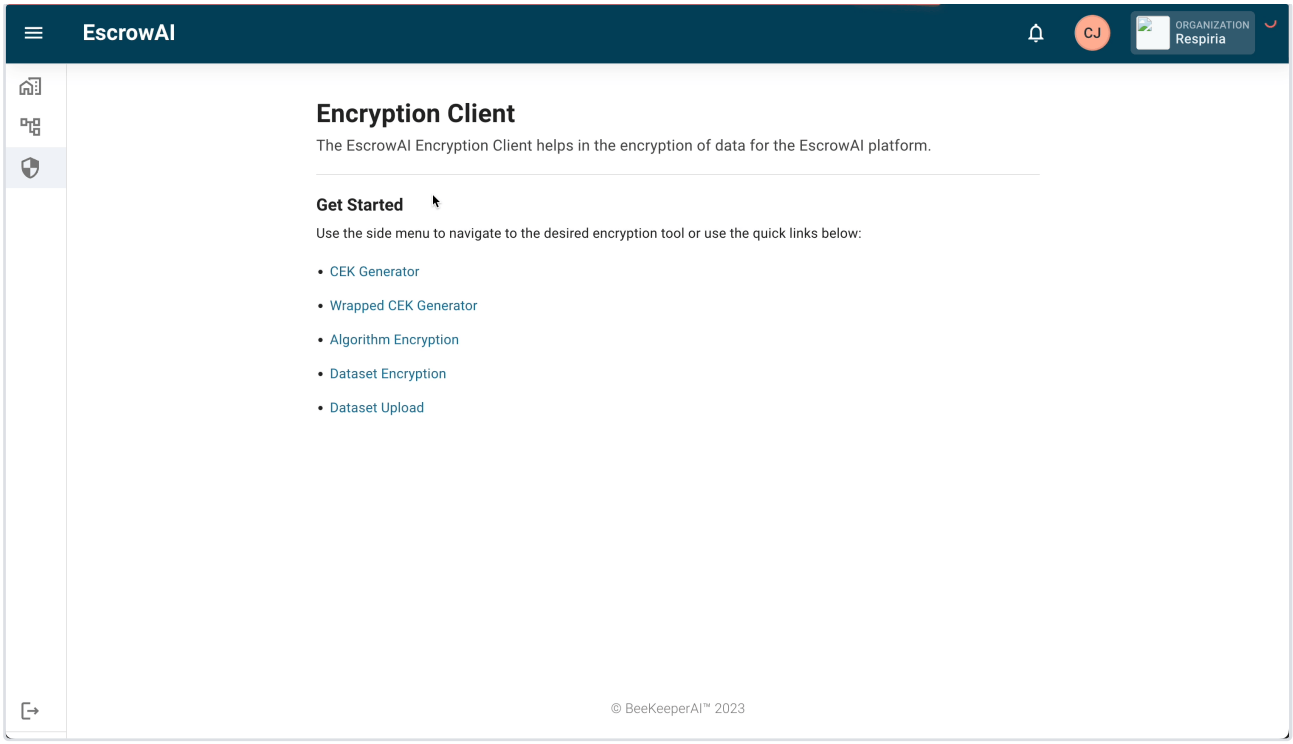
Steps

To generate a Wrapped Content Encryption Key (WCEK):

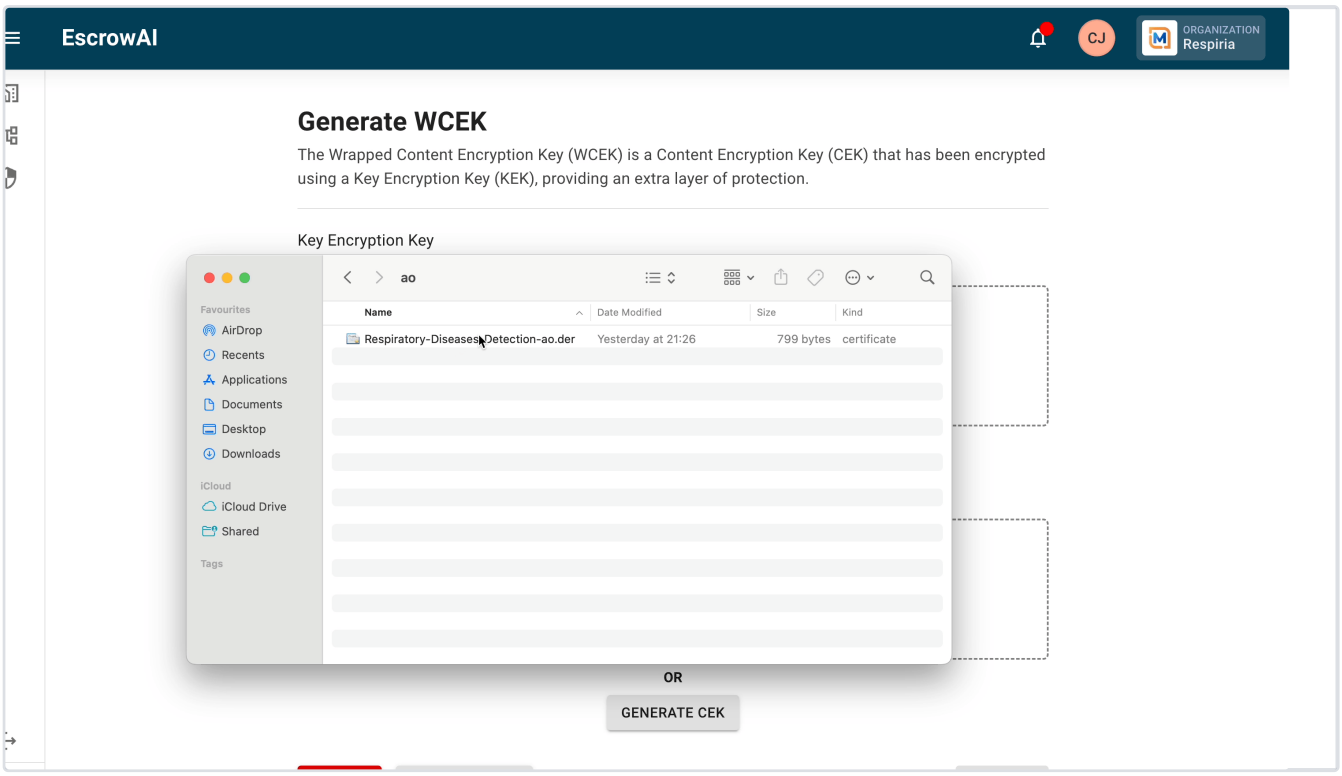
1. Download your project's public key, a **.der** file, from **EscrowAI**.



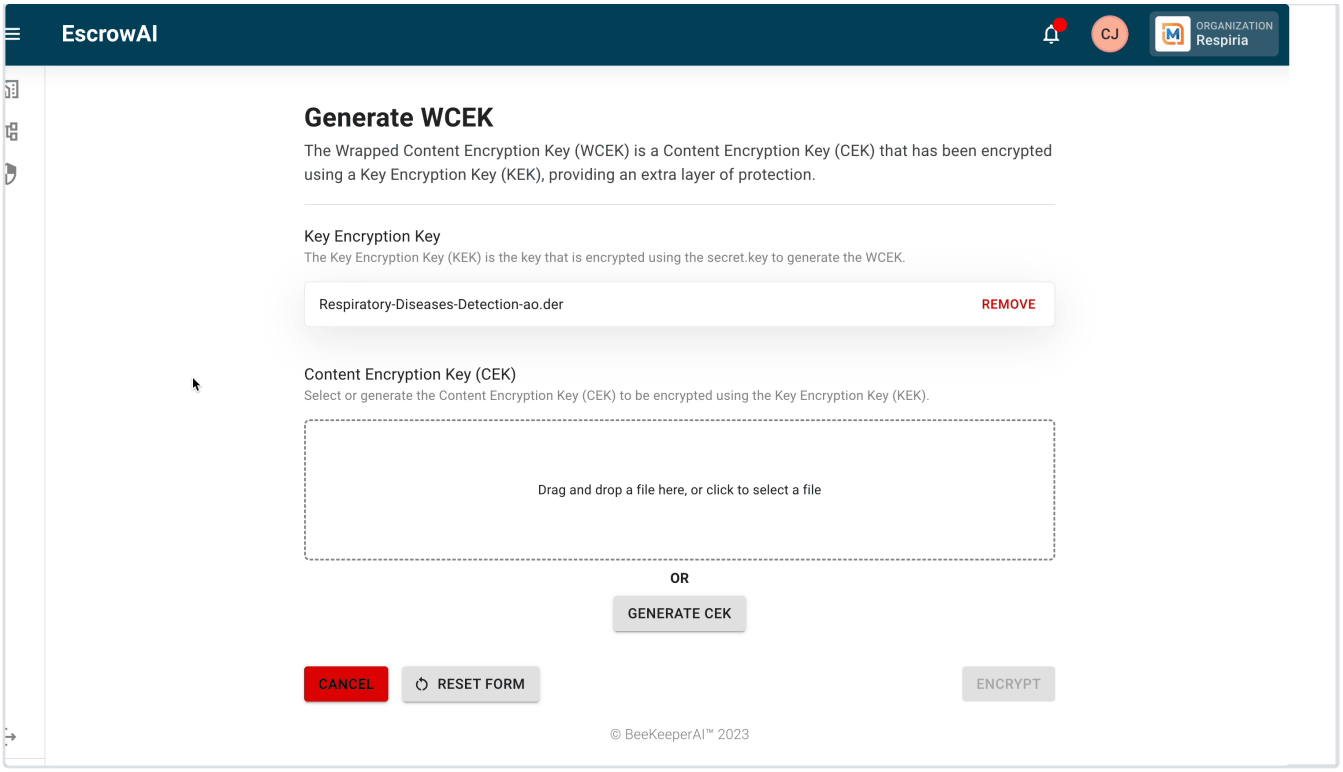
2. Select **Wrapped CEK Generator** from the Encryption client’s home page or **Generate WCEK** from the side menu.



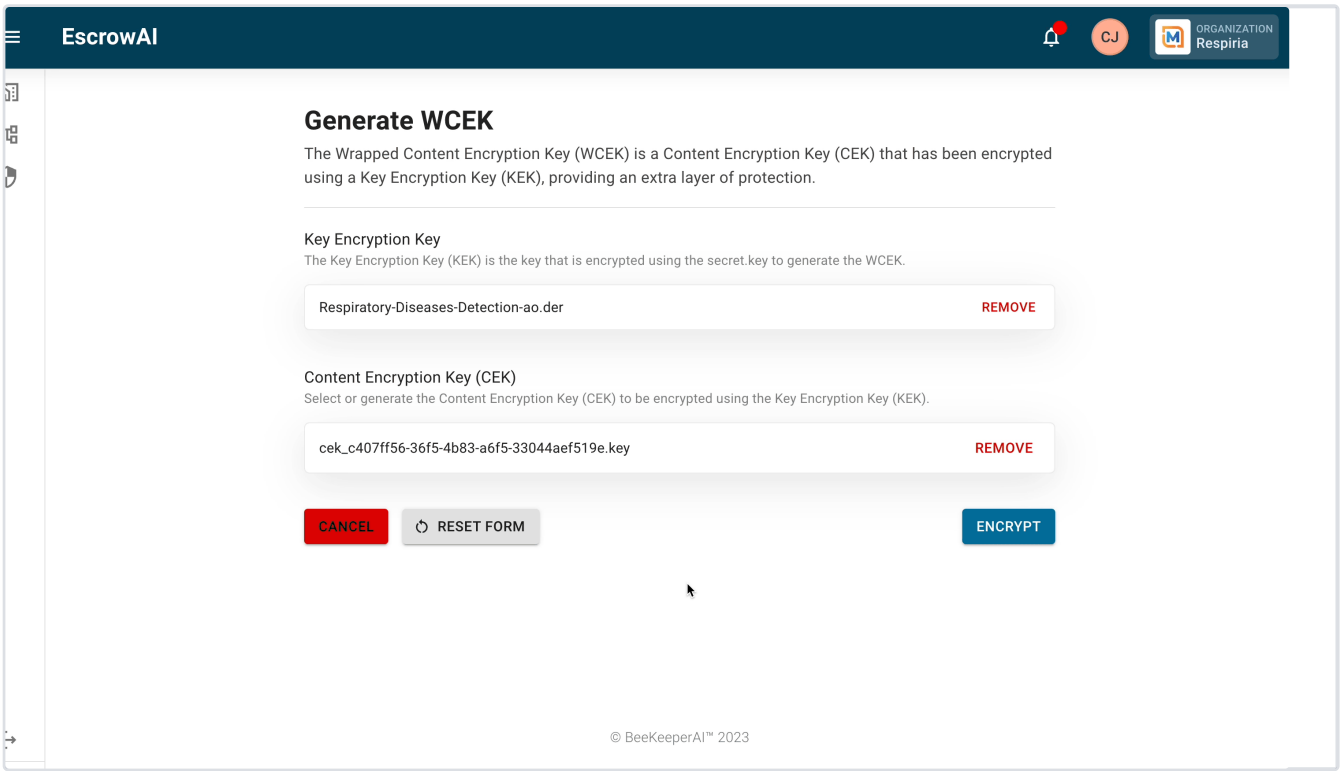
3. In the Key Encryption Key field, select the public key downloaded in Step 1.



4. In the Content Encryption Key field, select the file generated in the previous step (if you don't have a CEK, click on the Generate CEK button).



5. Click on the Encrypt button to begin encrypting the CEK. Once encryption is complete your WCEK will be automatically downloaded.



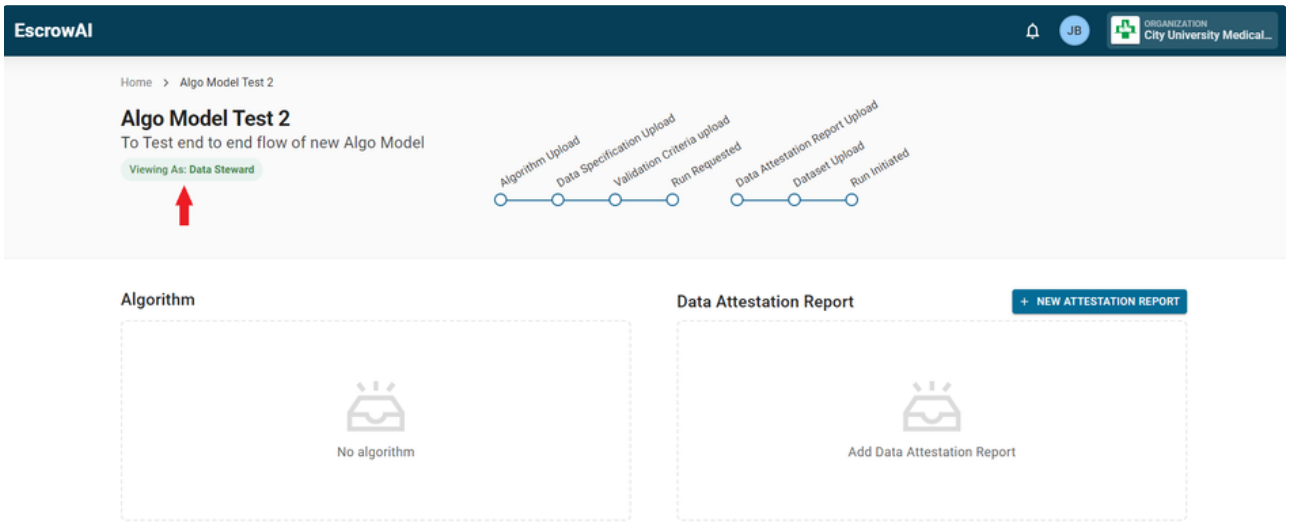
I am a Data Steward

What is a Data Steward?

Data Steward is a generic term assigned to entities that have very granular subject data that is proprietary and/or privacy protected and that are authorized to use the data for the purposes of a project within EscrowAI. Data Stewards curate a data set from their records systems in a form and quantity needed by Algorithm Owners to validate or train their AI models or support other analytical objectives.

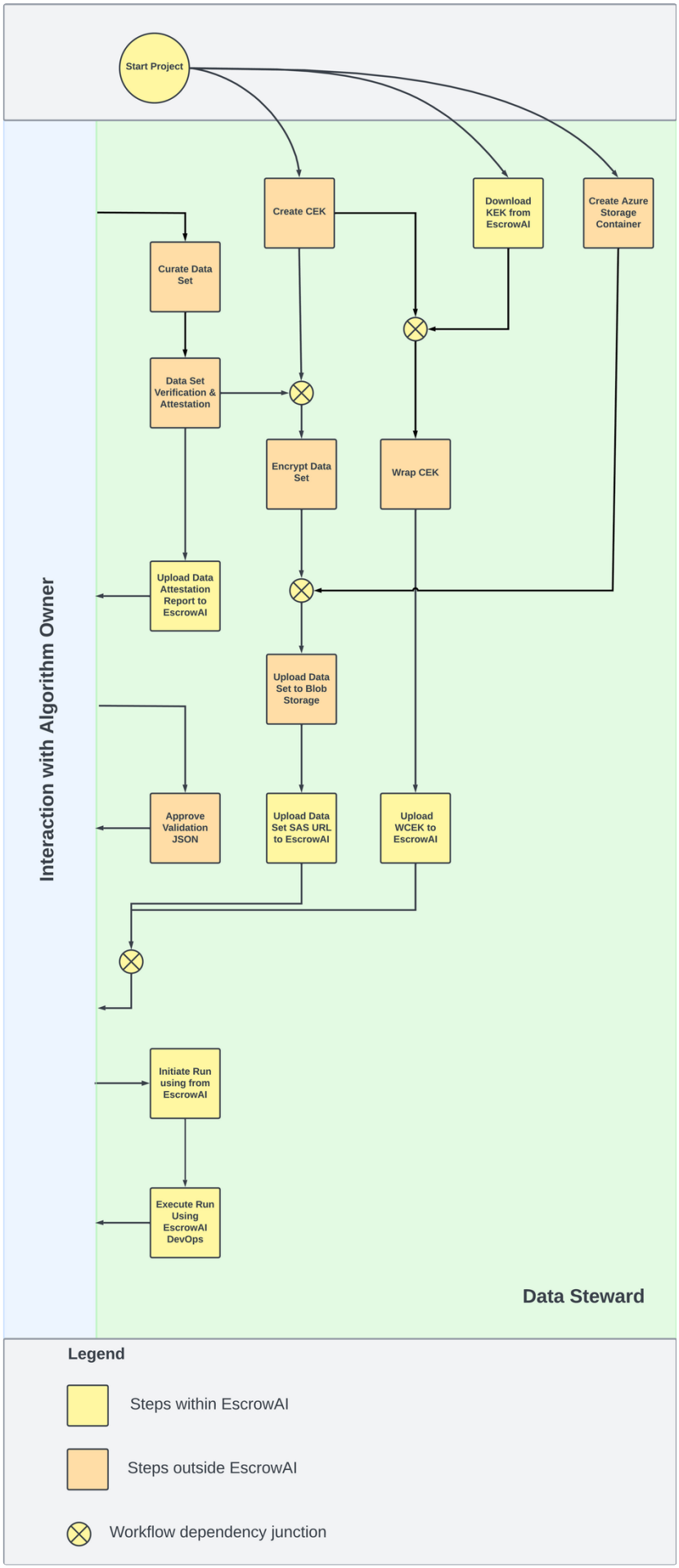
Project page

The Data Steward role is identified in the project page below the project name and description. The Data Steward's project interaction is focused on the right side of the project page.



Data Steward essential steps

The Data Steward project workflow is shown in the flowchart below and explained in the following pages within this section.


Start Project
Create CEK
Download KEK from EscrowAI
Create Azure Storage Container
Curate Data Set
Data Set Verification & Attestation
Encrypt Data Set
Wrap CEK
Upload Data Attestation Report to EscrowAI
Upload Data Set to Blob Storage
Approve Validation JSON
Upload Data Set SAS URL to EscrowAI
Upload WCEK to EscrowAI
Initiate Run using from EscrowAI
Execute Run Using EscrowAI DevOps
Interaction with Algorithm Owner
Data Steward
Legend
Steps within EscrowAI
Steps outside EscrowAI
Workflow dependency junction

Curate data per AO Data Specification

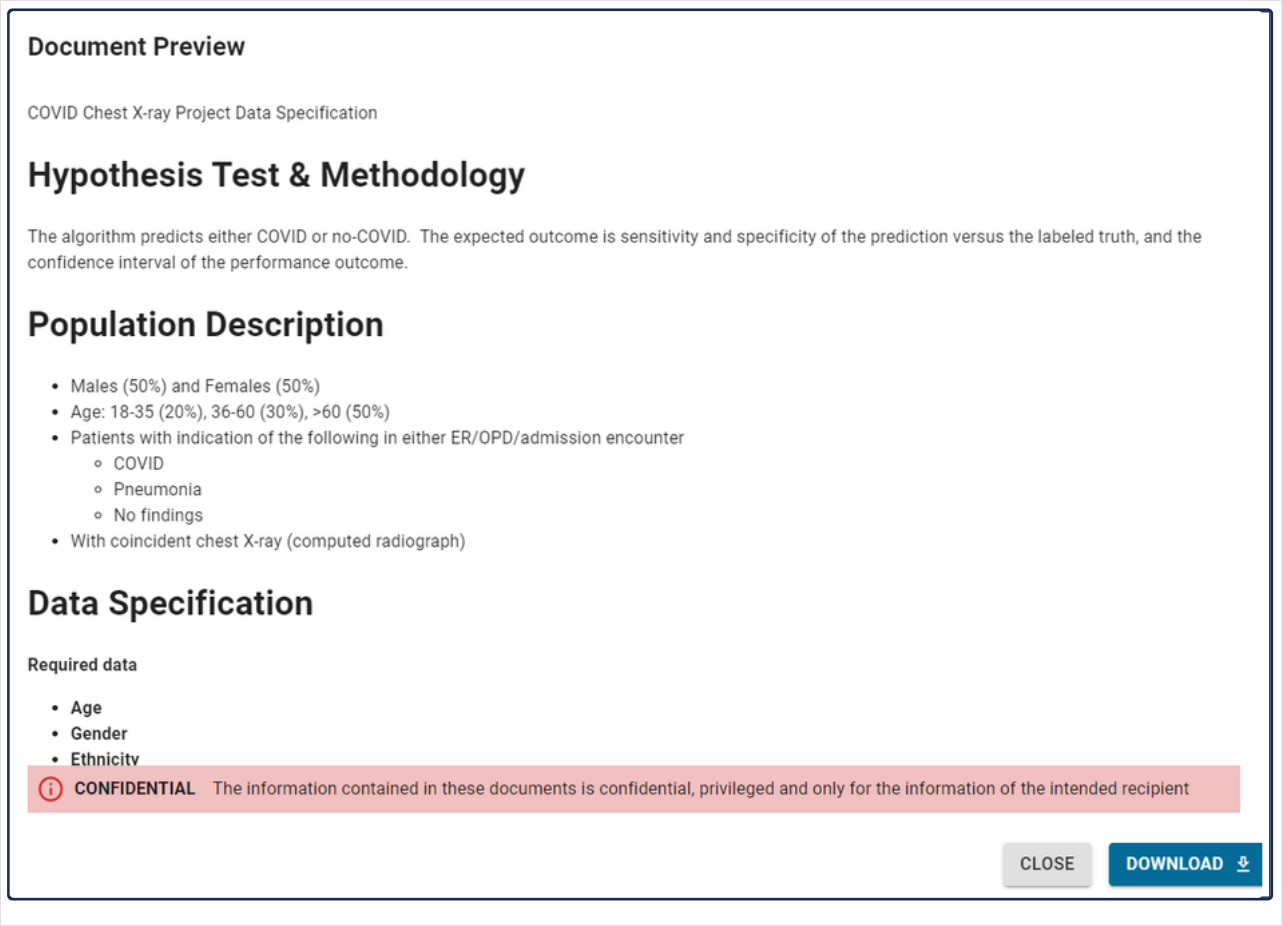
What is a Data Specification?

In EscrowAI, the data specification is a critical component of a project within which the Algorithm Owner defines the specific requirements for the data that will be used in the algorithm. This includes details such as the subject population (e.g., inclusion and exclusion criteria), the number of records, the data content and form specification, and the data set validation requirements. The data specification ensures that the data used in the algorithm is accurate, complete, and meets the necessary quality standards. It also helps to ensure that the algorithm produces reliable results that can be used to support decision-making or research. In summary, the data specification is a vital part of the collaboration between the Data Steward and Algorithm Owner, and it helps ensure the project's success by providing clear and concise requirements for the data used in the algorithm.

Downloading the Data Specification

As a Data Steward, you can download a data specification from the Project Page.

1. Go to the Data Specification section from the Project Details page.
2. Click the **PREVIEW** button from the Data Specification card. A Document Preview window is displayed.



3. Click **DOWNLOAD** to download the entire Data Specification.

Data curation

The Data Steward uses their organization's tools and processes to query, combine, form, and clean the data according to the Data Specification.

The creation of the truth standard used for model training and inference validation is an important step of data curation. The form of the truth standard is part of the Data Specification. The truth standard may be based on a subject matter expert’s adjudication of a classification (e.g., confirming a diagnosis through record or image review), or the SME’s expert labeling of an area of interest in a medical image. One or more SMEs may be needed to establish the truth standard.

Attest conformance to Data Specification

Data Attestation

Because patient data is never visible to the algorithm owner, the Data Steward can certify that the data meets the stated requirements using a “Data Attestation Report.” The report provides a summary of the population statistics that demonstrates compliance with the specification, along with the truth standard and the acceptance criteria used to validate the data set.

The Data Steward (DS) retrieves data from various data repositories to fulfill the data specification. The data is structured as required by the Data Specification. The DS verifies the data set by using the method specified in the Data Specification. The DS creates a Data Attestation Report that is approved by the Algorithm Owner (AO). Once approved, the DS encrypts the data set using a DS-controlled content encryption key and deploys it to Azure Blob storage within its Azure compliant cloud.

Upload the Data Attestation Report

1. Go to the Project Details page and click on the **NEW ATTESTATION REPORT** button to begin adding a new report.
2. The **Name** field is automatically populated based on the Data Specification name.
3. Provide a brief description of the data attestation report you are uploading in the **Description** field.
4. Enter a version number in the **Version Tag**. This version must attest to a data set that will be added later in the workflow. Ideally this is a unique number or alpha-numeric you can use to identify and search.
5. Add a version description in the text box provided.
6. You can either drag and drop your data attestation report into the designated area, or click on the gray box to open your file browser and select and upload the file from your computer or device.
7. Once the file has been uploaded, click the **SUBMIT** button to save.

Upload a new version of a Data Attestation report

1. Navigate to the Project Details page for the relevant project.

2. Select **NEW VERSION** from the Data Attestation Report card.
The Name and Description fields are automatically filled and cannot be edited.

3. Add the information and upload the new file, as described above.

6. Click **SUBMIT**.

Upload the dataset WCEK

Uploading the Wrapped Content Encryption Key (WCEK) is the means to securely transfer the private Content Encryption Key (CEK) to EscrowAI. The Key Encryption Key is used to encrypt (“wrap”) the private Content Encryption Key (CEK). The WCEK is uploaded in the Data Steward Keys section of the Project page.

EscrowAI retains only the **last** WCEK uploaded. **Data sets encrypted with a version of the CEK not encrypted within the current WCEK version cannot be decrypted in the TEE.**

This form of rolling your CEK gives you the power to disallow access to the data set by EscrowAI. You can roll your CEK at any time for any reason, but if you want the data set to be accessible you must either:

1. re-encrypt it with the new CEK, upload to Blob storage, create a new SAS URL, and upload the new SAS URL to EscrowAI as a new data set version, or
2. rewrap the CEK associated with the data set(s) and re-upload the WCEK.

See Common encryption steps for the common encryption workflows associated with downloading the Key Encryption Key, generating a Content Encryption Key, and encrypting (wrapping) the CEK.

Steps

To upload the dataset's corresponding WCEK:

1. Go to the desired project page.
2. Click **UPLOAD WCEK**.
3. Fill in the required fields and select the encrypted key file for upload.
4. Check the confirmation that the key is encrypted.
5. Click the **SUMBIT** button to save.

Encrypt dataset

Prerequisite encryption setup

Be sure that you have prepared the following:

- Download Key Encryption Key from EscrowAI
- Generate a Content Encryption Key (CEK)
- Generate a Wrapped Content Encryption Key (WCEK)

Steps

Be sure that your dataset is not zipped. You need to upload unzipped files or folders, not zipfiles.

To encrypt datasets using the encryption client:

1. Go to the desired project page.
2. On the left, click the shield icon to go to the encryption client page.
3. Select **Dataset Encryption**.
4. In the **Dataset** field, select the files or folder that are your dataset. Make sure you select unzipped files or folders.
5. After you select the files, you can choose to omit files you don't want encrypted. These will not be included in the encrypted output.
6. In the Content Encryption Key field select your CEK. This is the same CEK that is wrapped to create the WCEK.
7. Click **Encrypt** button to begin dataset encryption.

After encryption is complete a success notification is displayed, and the encrypted dataset is downloaded as a zip file.

Upload encrypted dataset to Azure Blob container

Background on Azure Blob storage

Azure Blob Storage is the storage mechanism used by EscrowAI to make the data available to the Azure-based confidential compute resource. The Data Steward (DS) must transfer the encrypted data to a Azure Blob Storage container within a storage account located in their Azure subscription.

Storage accounts

A storage account provides a unique namespace in Azure for your data. Every object that you store in Azure Storage has an address that includes your unique account name. The combination of the account name and the Blob Storage endpoint forms the base address for the objects in your storage account.

Containers

A container organizes a set of blobs, similar to a directory in a file system. A storage account can include an unlimited number of containers, and a container can store an unlimited number of blobs.

EscrowAI pulls a comprehensive data set out of blob storage into the confidential compute node. The data set is version controlled within the application so that there is traceability of results based on data and algorithm version. For this reason, a container must contain data at the Project-Data Set Version level. For example,

- DS-subscription
  - Projects-Storage-Account
    - Project-A-dataset-v1-container
      - blobs (folders, files)
    - Project-A-dataset-v2-container
    - Project-B-dataset-v1-container

⚠ Ensure data sets are put in unique containers at the Project-Data Set Version level.

Blobs

Azure Storage supports three types of blobs, Block, Append, and Page. The specific type of blob used may depend on the project type. The most common type for use with EscrowAI is a **Block** blob.

- **Block blobs** store text and binary data. Block blobs are made up of blocks of data that can be managed individually. Block blobs can store up to about 190.7 TiB.
- **Append blobs** are made up of blocks like block blobs, but are optimized for append operations. Append blobs are ideal for scenarios such as logging data from virtual machines.
- **Page blobs** store random access files up to 8 TiB in size. Page blobs store virtual hard drive (VHD) files and serve as disks for Azure virtual machines.

Instructions to upload encrypted data

Encrypted data can be uploaded/placed in Azure blob storage in several ways, depending on your own operations/requirements:

1. Azure Portal
2. Azure Storage Explorer
3. Azure CLI

4. Encryption tool in EscrowAI

Azure Portal

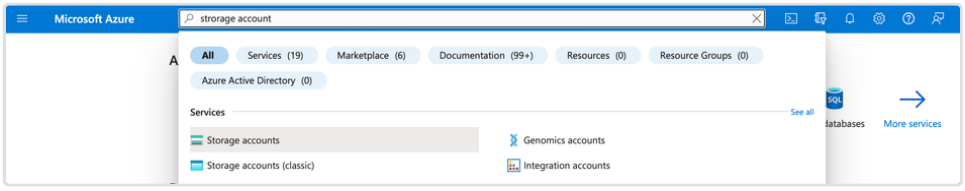
Azure Portal can be used to create and organize storage accounts and containers, and upload and manage Blobs.

1. Create a Storage Account

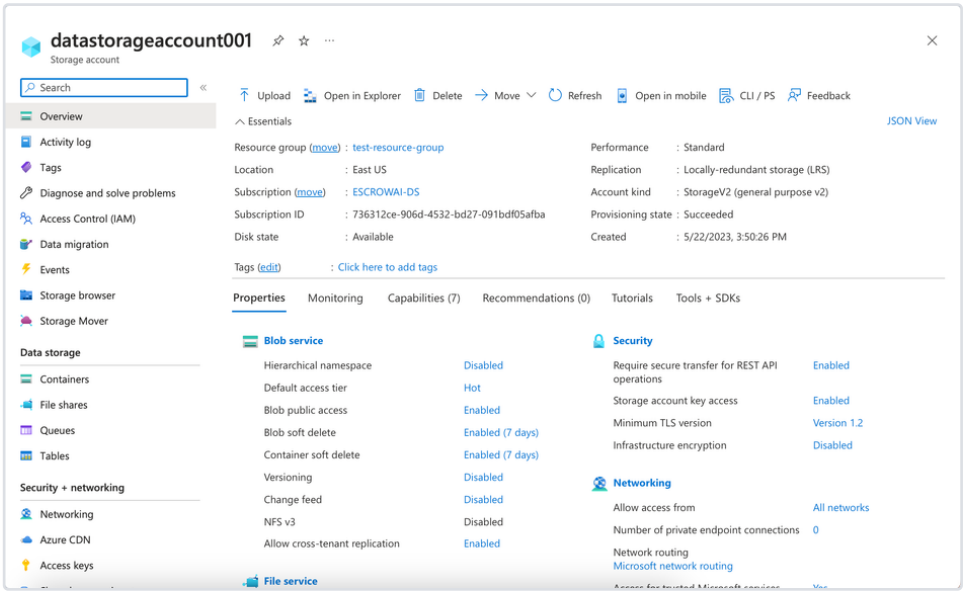
Creation of the Storage Account may be a one-time operation, depending on your organization's policies (e.g., you can create one Storage Account for all EscrowAI projects).

See also: EscrowAI-Azure Tenant Technical Setup

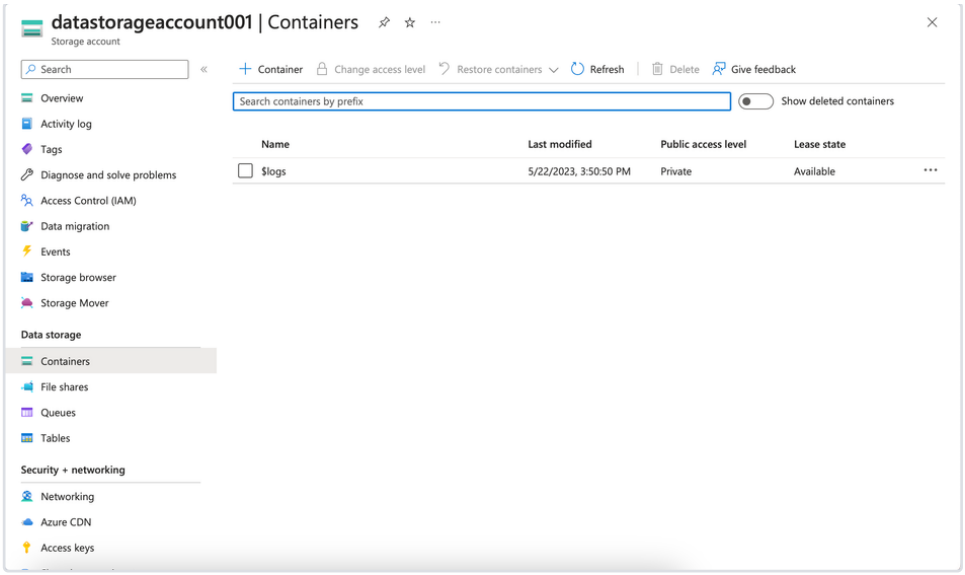
2. Create a Container
   a. Goto Azure Portal and search for Storage Account



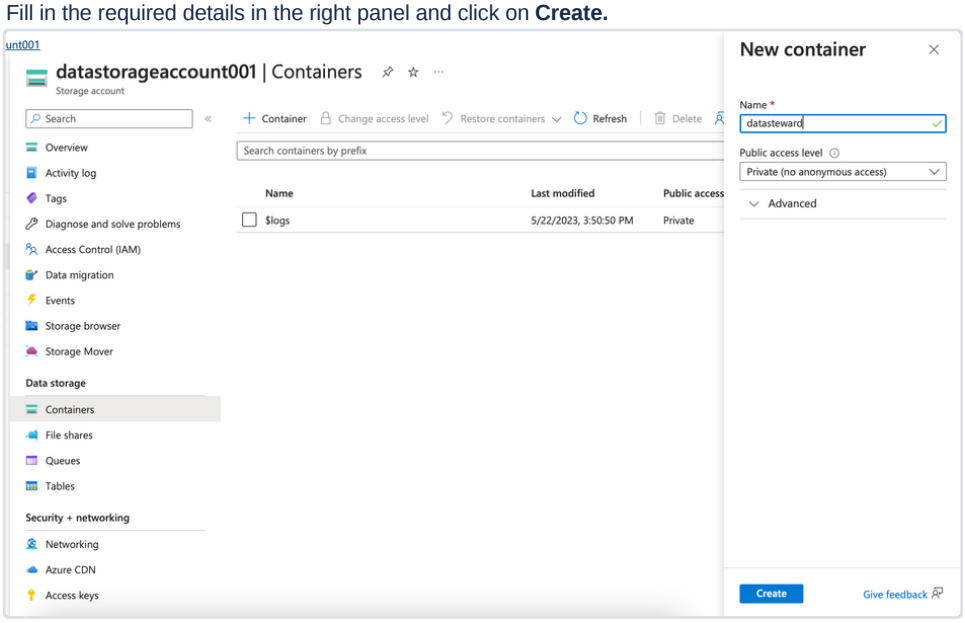
   b. Select the Storage Account used for EscrowAI projects.



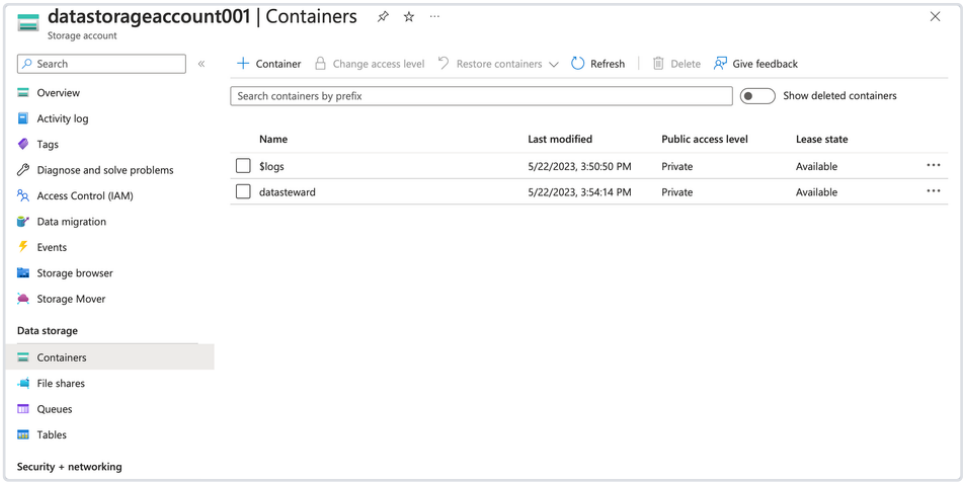
   c. Select the **Containers** from the left panel and click on **+ Container** to create a container in storage account.



d. Fill in the required details in the right panel and click on **Create.**

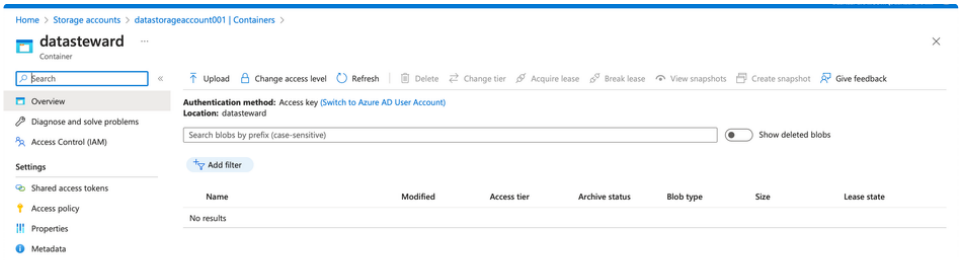


e. Once the container gets created, you will be able to see the container in the Storage account.

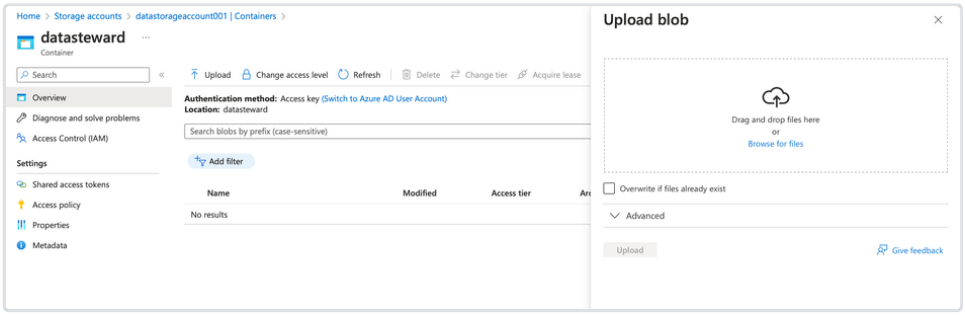


3. Upload Blobs using Azure Portal

a. Select the container that was created in previous step.



b. Click on **Upload** to upload the files to the container. Use the Advanced tab to refine the Blob details.



Azure Storage Explorer

Follow Microsoft's instructions from the below link to upload the files to the blob container via Azure Storage Explorer:

Create a blob with Azure Storage Explorer - Azure Storage

Azure Command Line Interface

Azure CLI can be used to perform the above operations. The Azure CLI can be invoked from the Bash environment in Azure Cloud Shell or run locally using a locally installed Azure CLI.

1. Open a command prompt or terminal window.
2. Log in to your Azure account using the following command:

```
az login
```

3. Create a storage account using the following command (optional):

```
az storage account create --name <account-name> --resource-group <resource-group-name> --location <location> --sk
```

4. Create a container in the storage account using the following command (optional):

```
az storage container create --name <container-name> --account-name <account-name> --account-key <account-key>
```

5. If you have storage account-key, upload data to the container using the following command. Replace

<container-name>, **<path-to-data>**, **<account-name>**, and **<account-key>** with your actual values.

```
az storage blob upload-batch --destination <container-name> --source <path-to-data> --account-name <account-name>
```

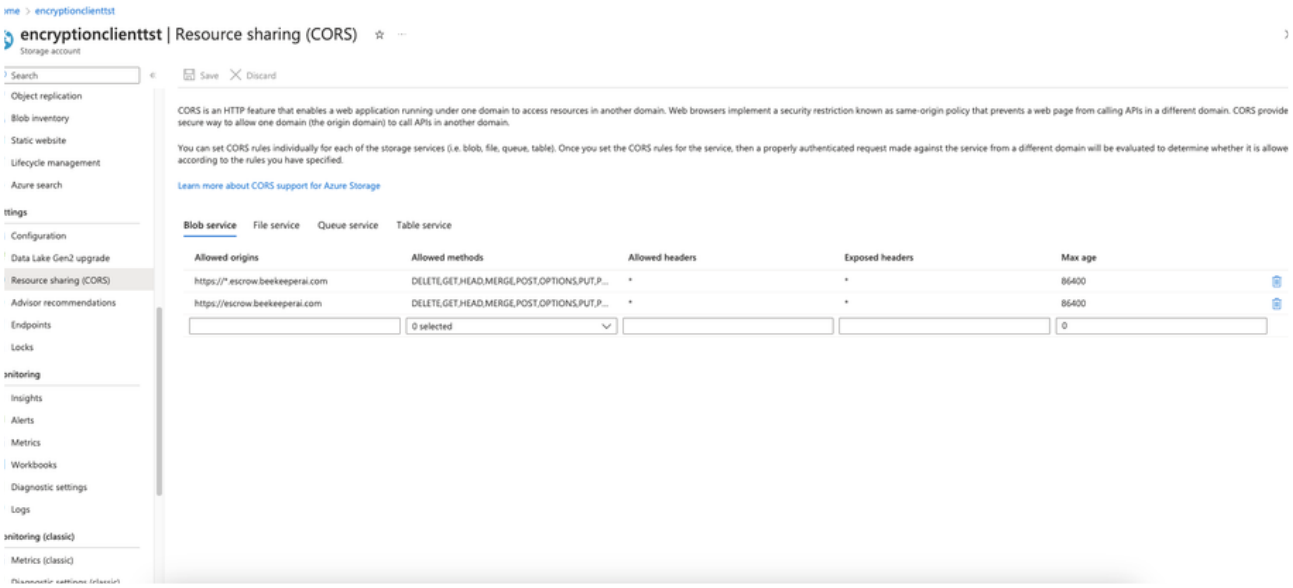
Generate a SAS URL for the Azure Storage Container

In order to upload to an Azure Blob Container, you'll need the container's Shared Access Signature (SAS) URL. To generate a valid SAS URL follow the instructions below:

Update Storage Account CORS Settings

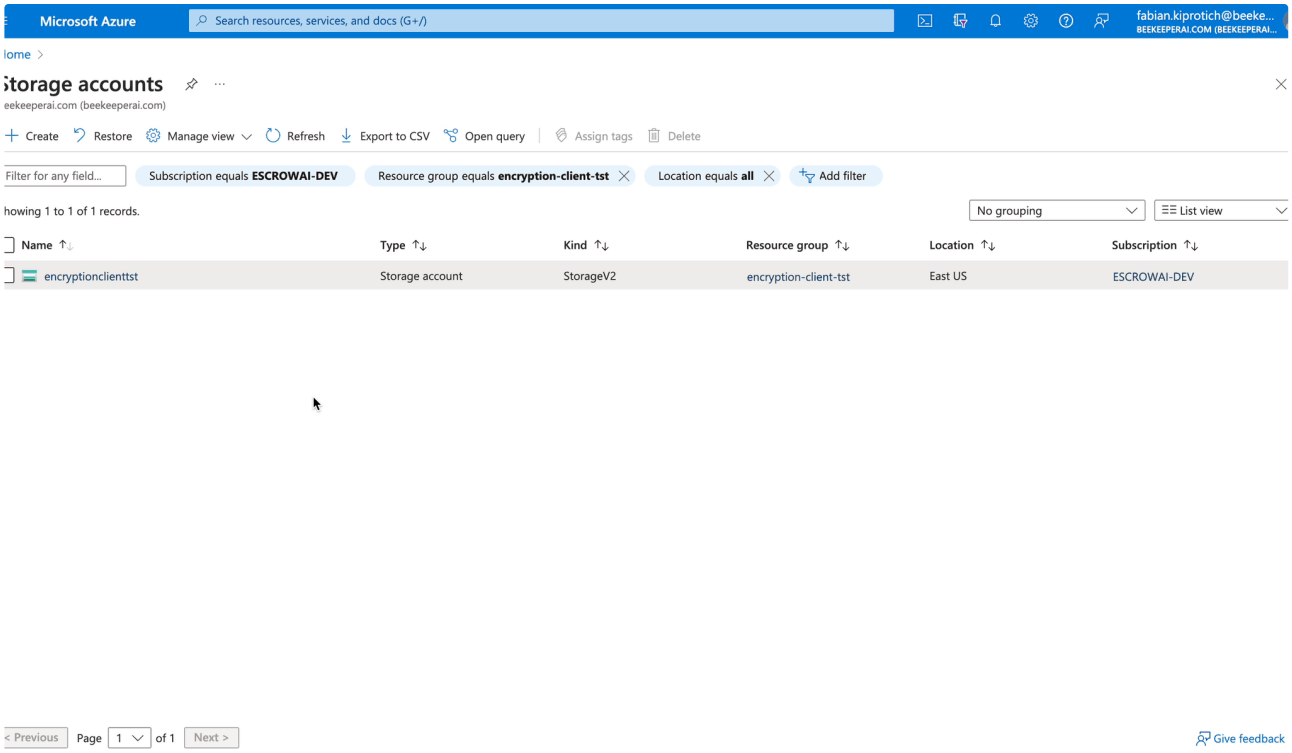
First, you will need to update the CORS settings of your Azure Storage Account with these origins:

1. *https://*.escrow.beekeeperai.com*
2. *https://escrow.beekeeperai.com*
3. To do this, visit the Azure Portal and select your storage account, navigate to the **Settings** section, and select **CORS**.



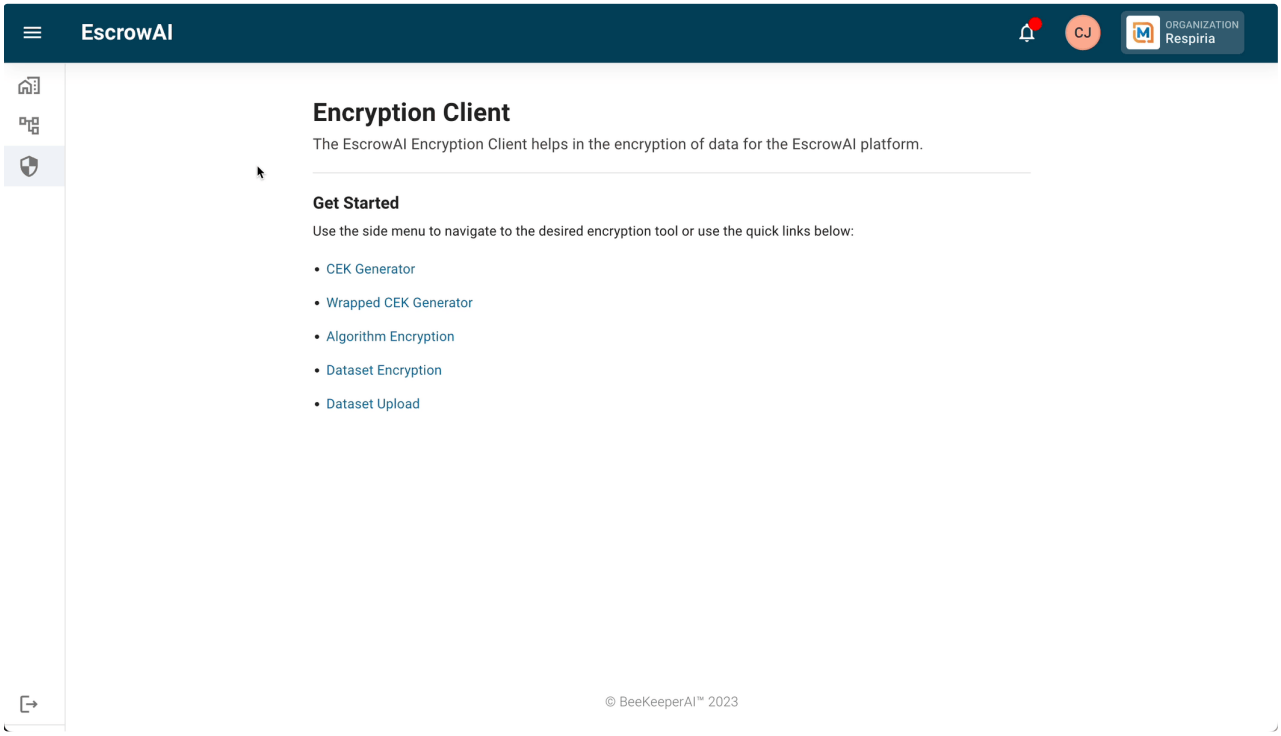
Generate SAS URL

1. Navigate to the **Containers** section of your storage account.
2. From the containers table, click on the settings icon of your container and select **Generate SAS** from the menu.
3. Set the relevant permissions of the SAS URL and click the "Generate SAS token and URL" button.
4. You can then copy the Blob SAS URL for use in the encryption client.

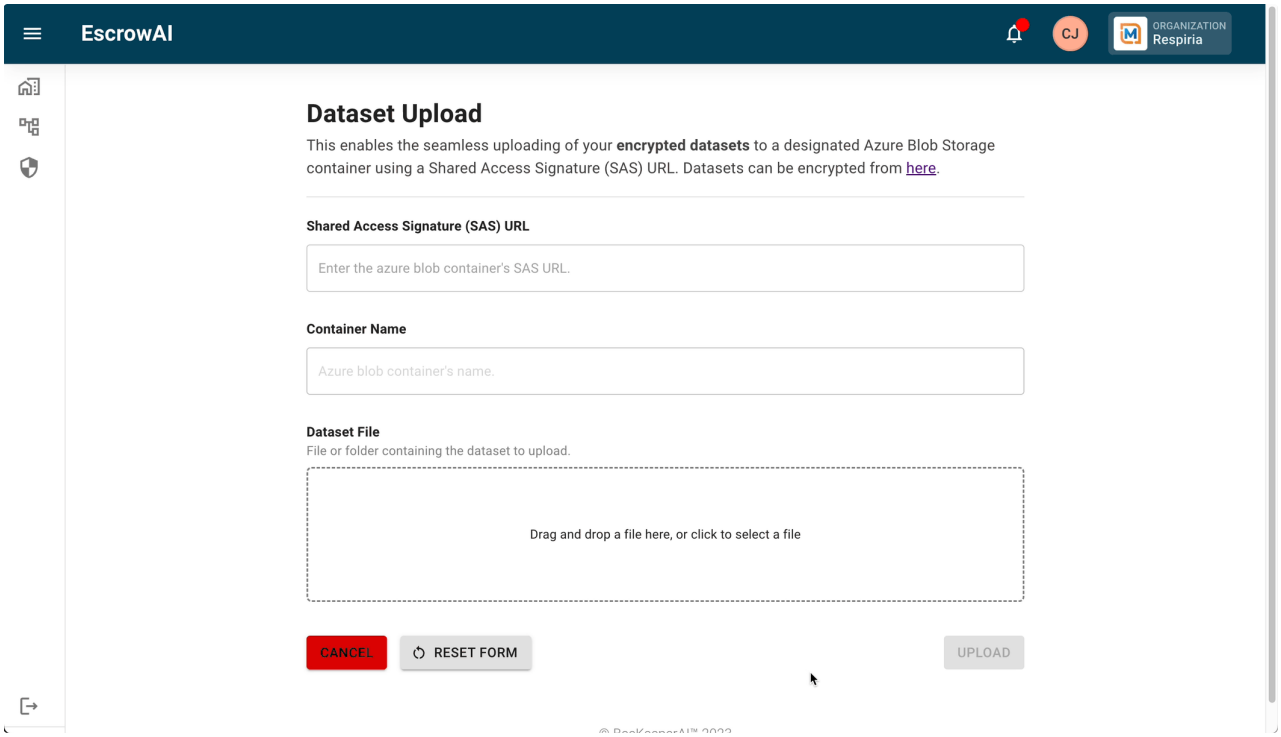


Upload the encrypted data

After you've obtained your SAS URL, in the EscrowAI UI, you can upload your dataset to the Azure blob container you defined as a prerequisite.

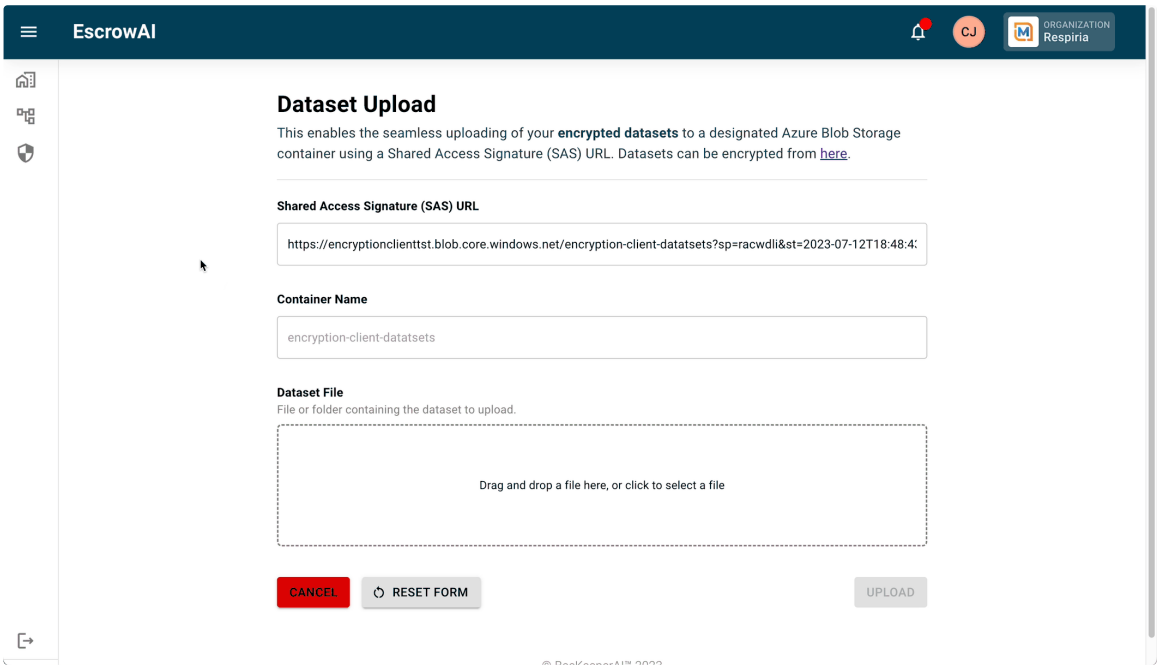
1. Select Dataset Upload from the Encryption Client's home page or side menu.
   - 

2. In the Shared Access Signature (SAS) URL field enter the SAS URL you generated.
   The validity of the SAS URL is checked by EscrowAI.
   The **Container Name** field is filled in by the system.

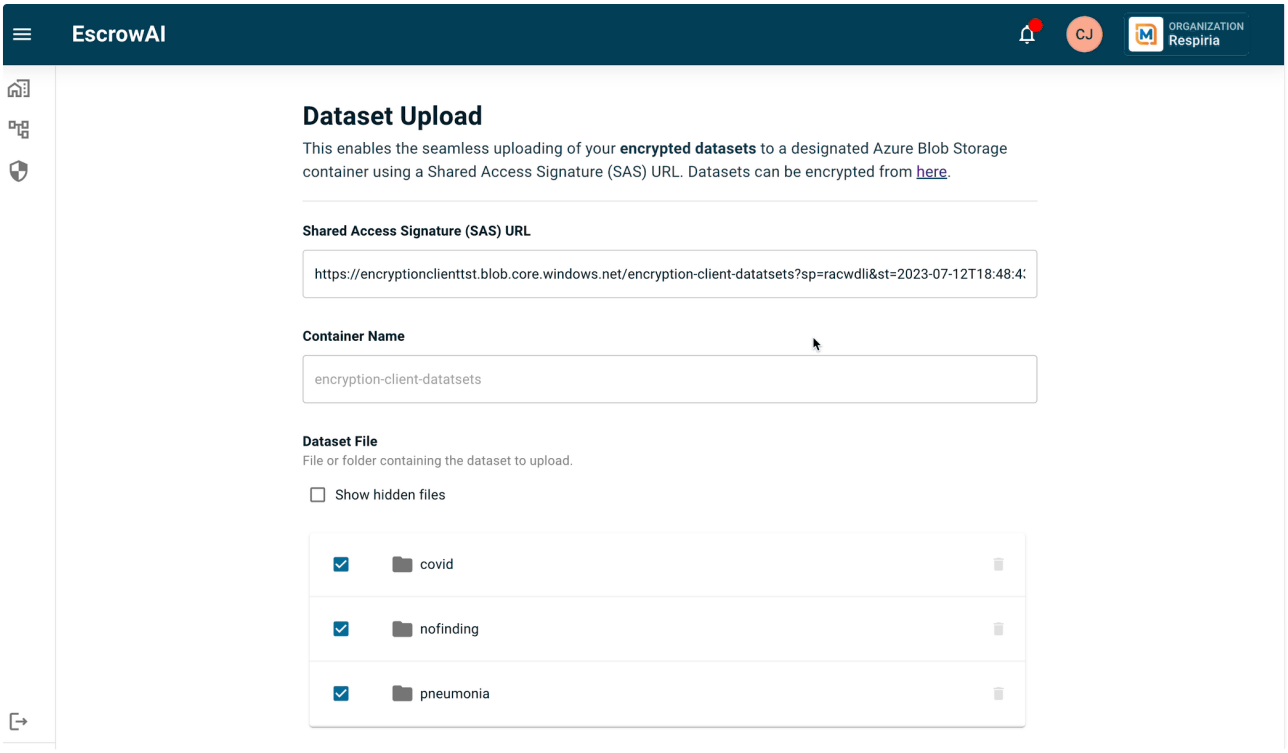


3. In the Dataset File field, select/drop your unzipped and encrypted dataset file.
   After that, you can use the EscrowAI UI to omit any files you don't want to be uploaded from the rendered directory view.

> Do not move your files on your local system until you complete and submit the form in EscrowAI.



4. Click **Upload** to begin uploading your dataset to the specified Azure Blob Container.



Microsoft documentation

Azure Storage Explorer – cloud storage management

Create a blob with Azure Storage Explorer - Azure Storage

Upload, download, and list blobs - Azure CLI - Azure Storage

Azure Cloud Shell Overview

Generate a Signed URL for the Dataset

Generating a shared access signature for a dataset allows you to securely and easily share access to your data with EscrowAI. A shared access signature (SAS) is a URI that grants restricted access rights to a specific dataset stored in the Data Steward's Azure Storage subscription. You need the appropriate permissions and access to the dataset to generate a signed URI.

EscrowAI requires an account SAS to be entered for the **Container** holding the data set. This allows the platform to pull the data set into the confidential compute node during a run. The SAS URI must be valid (within the start and expiry dates & times) when a run is initiated and in progress.

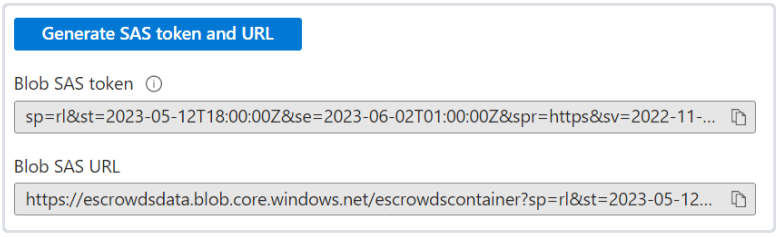
Generate a Signed URL for the Dataset

1. Log in to your Microsoft Azure portal and select the desired subscription.
2. Select the target Storage account.
3. Select **Containers** in the Resource menu.
   a. Click on the desired Container in the working pane.
   b. Click on the three dots on the right side of the row to expose the activity options menu.
   c. Select **Generate SAS**.
4. In the **Generate SAS** window, complete the options.

- Signing method: **Account key**
- Signing key: select desired key.
- Permissions dropdown: select **Read** and **List**.
- Specify the **Start** and **Expiry** dates and time for URI.
- Select **HTTPS only**.

> Data in the Container will be available to EscrowAI for the duration of the SAS token. The duration specified is at the discretion of the Data Steward and should balance the security needs with access over the project duration. When a SAS token expires, a new token must be generated and entered into EscrowAI before a new run can be initiated.

5. Click **Generate SAS token and URL**. Copy the **Blob SAS URL**, which includes the token (shown below), and paste it into the required field in the EscrowAI project.

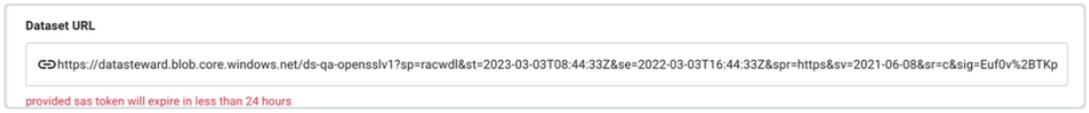


Add a dataset URL to EscrowAI

The process of adding a dataset to EscrowAI doesn't transfer any data to the platform. Instead, the SAS URL is given to the platform along with dataset meta data. No data leaves the Data Steward environment since the confidential compute node is spun up within the Data Steward's environment and the dataset is streamed into the confidential compute node from the Data Stewards Azure BLOB storage.

Add a new dataset

1. Go to the desired project page.
2. Click **NEW DATASET**.
3. Fill in the required fields.
4. Select the proper Data Specification and Data Attestation Report versions.
   This communicates to the Algorithm Owner the exact dataset contents.
5. Enter your Dataset SAS URL.
   EscrowAI checks the SAS URL for validity, with status shown below.
6. If the token is invalid, generate a new SAS URL and re-enter it in the **Dataset URL** field. See Generate a Signed URL for the Dataset.
7. Check the confirmation box that the associated dataset does not contain any unencrypted privacy-protected or confidential information.
8. Click **SUMBIT** to save.

Dataset URL	Description
Valid	The Dataset URL link is valid and can be successfully saved. 
Invalid	Either the SAS token was not provided or an invalid URL was entered. 
Expired	The provided SAS token will expire in less than 24 hours. 

Create a new dataset version

A new version of a dataset can be added by clicking on the **New Version** button on the Dataset card. Enter the information as described above.

> ⚠ If the new dataset is the result of a change in the Data Specification or is associated with a new Data Attestation Report, ensure that those artifacts are added to the project before adding the new dataset, so that the appropriate relationships can be selected.

Updating the SAS URL

A SAS URL has a specific expiration date. If this expiration will occur in 24 hours or less, the project page indicates that the SAS URL is expired. This means that the dataset itself has been marked as expired.

A new SAS URL must be added after the expiration date of an existing SAS URL. Adding a new SAS URL requires that a new dataset version be added.

> For clarity, add a comment in the Version Description field that the new version was created only to update the SAS URL. Select the appropriate Data Specification and Data Attestation Report.

Review run request and initiate or reject a run

A run request is configured and sent by the Algorithm Owner when the necessary algorithm, data set version, and validation criteria are available on the Project page.

Each run configuration is a unique combination of algorithm and data set versions. The AO can send a request for either a new or existing configuration.

Review a run request

When the Algorithm Owner (AO) sends a Run Request, the Data Steward (DS) receives an email notification.

The request is indicated by the **Run Requested** status on the DS view of the **Run Configurations** card.

Reject a run request

You can either accept the run request or reject it. If you reject, enter a reason for the rejection.

- The algorithm owner receives an email message stating the reason.
- A system notification will be posted to the system notification on the algorithm owner's home page. Unread notifications are indicated by a red dot on the bell icon in the upper right corner of the page.

Initiate a run

1. Go to the project in EscrowAI and select the **Run Configuration** card.

2. Review the configuration of the Run Request and ensure it is appropriate.

3. Click **INITIATE RUN**.

> EscrowAI allows only one run of a particular Run Configuration at a time. The INITIATE RUN button associated with any Run Configuration will be active only when there are no confidential computing resources processing that configuration.

Monitor and cancel a Run

Run Status

Once the Data Steward (DS) initiates the run, EscrowAI will start the requested TEE in the DS's environment and initiate the Run.

The run progresses through stages from building the runtime, launching the confidential computing virtual machine, and running the desired task or program. Monitor the top-level status of the Run in the **Run Updates** sidebar. This progression shows the progress of bringing up the TEE and initiating the run.

Run Updates	Description
Run Initiated	The run configuration has been submitted and the process of initiating the runtime environment is about to begin.
Runtime Build	The runtime environment is being built according to the specifications of the run configuration.
Workload Launch	The virtual machine (VM) that will run the algorithm is launched, the TEE is attested, and the run artifacts are brought into the TEE.
Run	The algorithm is currently running on the virtual machine.

Run Reporting and Cancellation

Status updates, run logs and the final report are displayed in the **Reports** window, as well as the ability for the Data Steward to cancel a run.

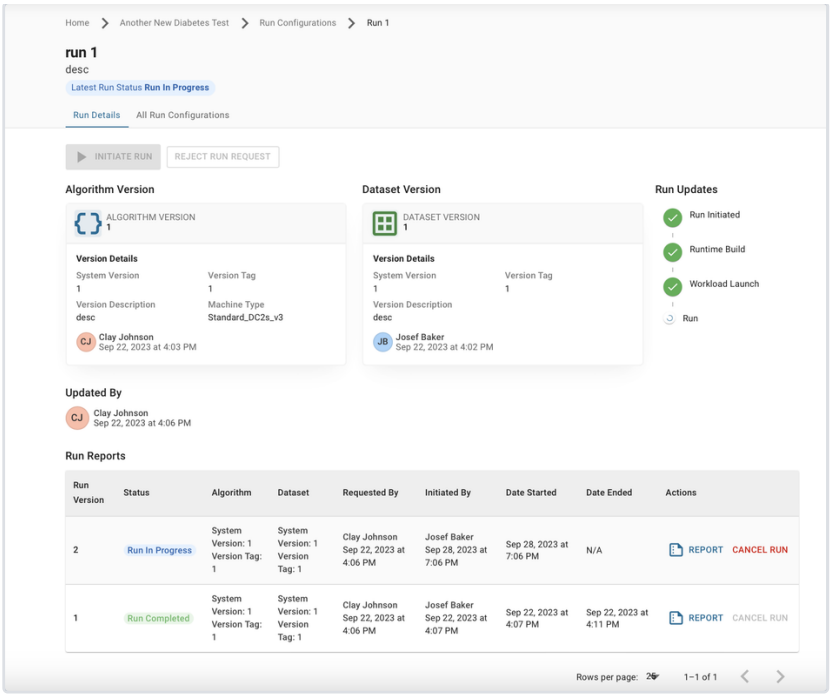
View the status of a run in progress

For a **Run in Progress** the Report will display a running status of the workload, updated at periodic intervals.

1. Go to the desired project.
2. Select the **Run Configurations** card. The **Run Details** tab for the last initiated Run is displayed.
3. The **Run Details** tab displays the Run Configuration details, the status of the **Run in Progress**, and a list of **Run Reports** associated with the configuration.
4. Click on the **Report** link in the Actions column for the Run In Progress to view the detailed run progress logs.

Cancel a run in progress

The Data Steward can cancel a run in progress by clicking the **CANCEL RUN** link in the Actions column under Run Reports.



Completed runs

After a run is finished, its status changes to either **Run Completed** or **Run Failed**. The final report (for the AO only) and run logs (AO and DS) are available from the same **Reports** window.

1. **Run Details**: This tab displays results for the latest Run Configuration. Click on any **Report** link in the Actions column to view the detailed run logs and report for this Run Configuration.
2. **All Run Configurations**: Click this tab to see all run results for any Run Configuration.
   a. Select the card for the Run Configuration of interest from the list of cards.
   b. Scroll to find the heading **Run Reports**.
   c. Click on the **Report** link in the Actions column to view the Report (AO only).
3. Click **Logs** to see the detailed run progress logs.

Run statuses and potential run errors

The tables below display the statuses and potential errors that can occur when the Report tab is selected during the ongoing run. They provide an overview of the current status and any possible issues encountered during the execution.

From VM	Status
In Progress	Compute node enrollment successful
	Connection to BeeKeeperAI container registry succeeded
	Algorithm container launch successful
	Algorithm container is initializing
	Algorithm container is running
Failed	Compute node enrollment failed
	Connection to BeeKeeperAI container registry failed

	Algorithm container launch failed
	Algorithm container initialization failed
	Data container launch failed
	Runtime container failed with a non-zero exit code

From SDK	Status
In Progress	BeeKeeperAI SDK initialized
	BeeKeeperAI SDK successfully initialized, workload starting
	Report has been generated, and validation is starting
Failed	Error while decrypting the secret key. Please encrypt using a new CEK and upload a new WCEK (Wrapped Content Encryption Key) using the EscrowAI-provided public KEK (Key Encryption Key) to proceed.
	While communicating with the Enclave HSM, we encountered an error with Enclave Authentication. Please contact BeeKeeperAI support.
	While communicating with the Enclave HSM, we encountered an error locating your Enclave key. Please try again after confirming that your CEK and WCEK are correct.
	Expected to find a WCEK at {}, but it was not found. This is a fatal error, please contact BeeKeeperAI support.
	While communicating with the Enclave HSM, we encountered an error while trying to decrypt your WCEK. Please try again after confirming that your Content Encryption Key and WCEK are correct.
	While decrypting and decoding the WCEK, we encountered an error. The CEK is invalid. This is likely due to the use of the wrong WCEK. Please encrypt using a new CEK and upload a new WCEK using the EscrowAI-provided public KEK to proceed.
	We successfully decrypted your CEK, but encountered an error while trying to write it to the file {} on the local filesystem. We are unable to proceed. Please contact BeeKeeperAI support.
	The CEK is invalid. This is likely due to the use of the wrong WCEK. Please encrypt using a new CEK and upload a new WCEK using the EscrowAI-provided public KEK to proceed.
	While decrypting your file, we encountered an error. Please try again after confirming that your CEK and WCEK are correct. If the problem persists, please contact BeeKeeperAI support.
	An error occurred while generating your report. Please contact BeeKeeperAI support
	Your report did not pass validation. Please contact BeeKeeperAI support.
	Secrets manifest could not be loaded
	Detected an invalid encryption mode. Please contact BeeKeeperAI support.
	Could not open the decrypted file in the secrets.yaml
	Could not open the encrypted file in the secrets.yaml

I am an Algorithm Owner

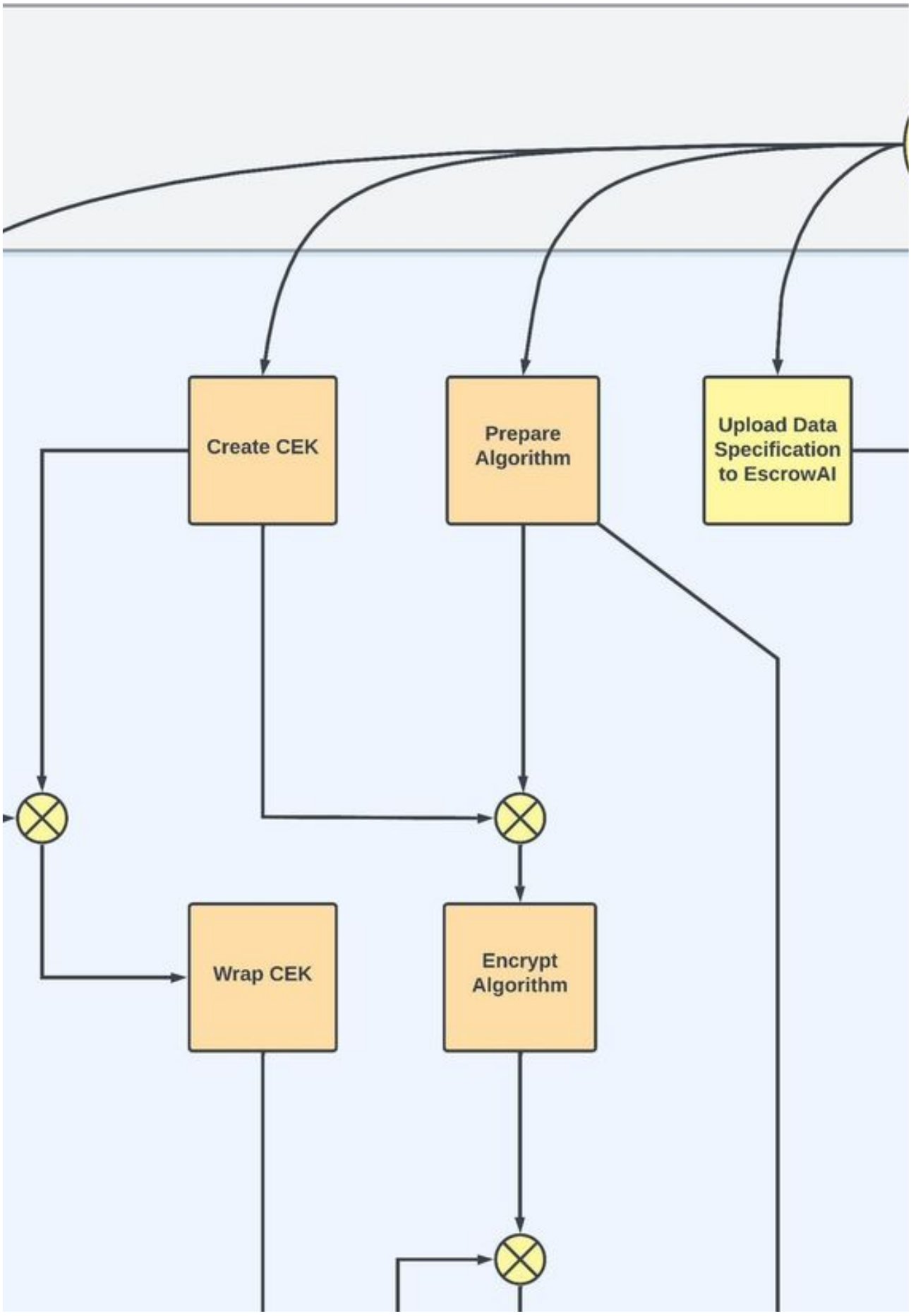
What is an Algorithm Owner?

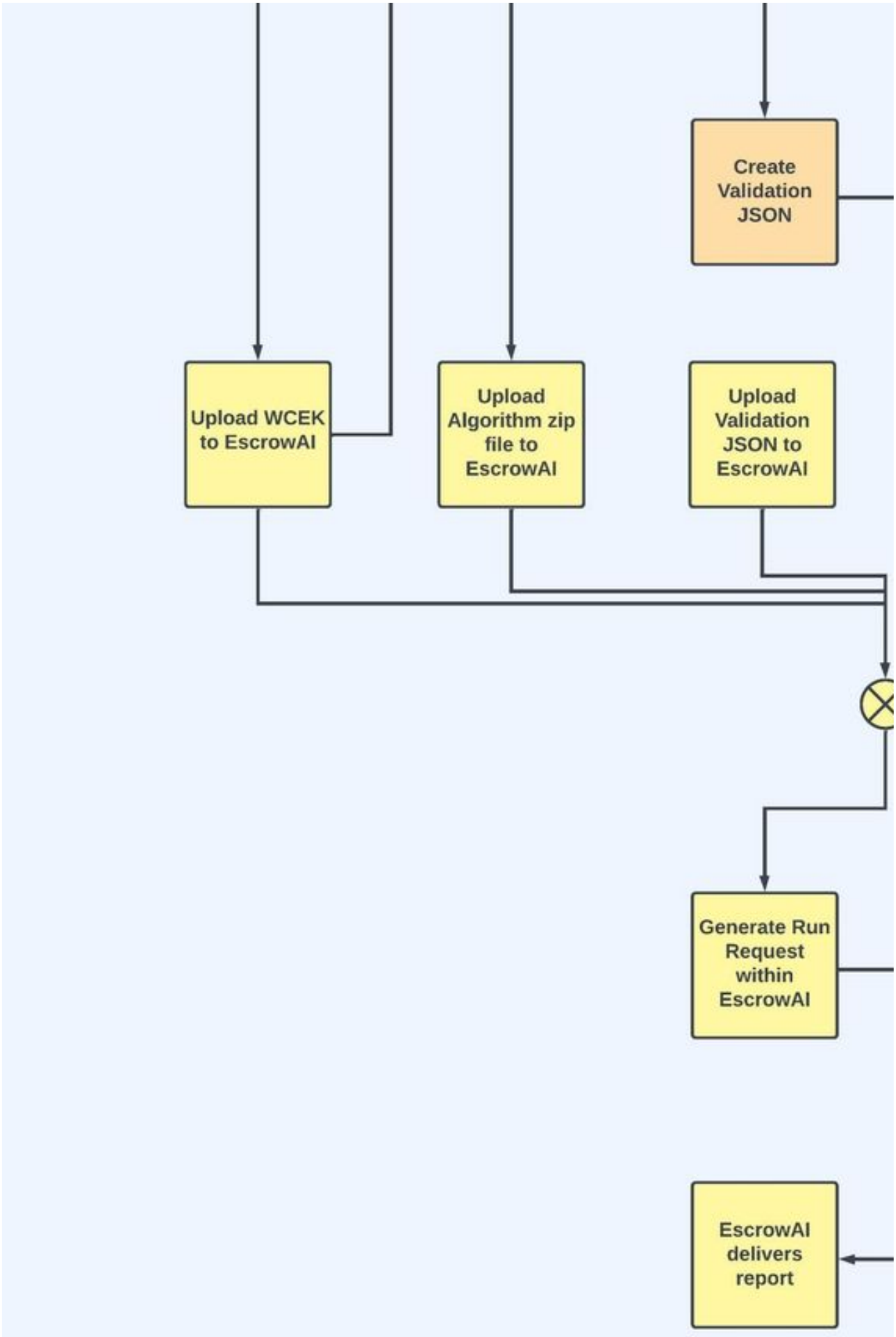
Algorithm Owner (AO) is a generic term assigned to entities developing analytic models, confidential queries, or artificial intelligence algorithms (or algorithms based on other modeling techniques) to address an issue or need that relies on access to privacy protected data. The AO defines the data needs using a Data Specification, prepares the algorithm to run in the confidential compute environment, proposes the output of the computation, and configures and initiates a Run Request.

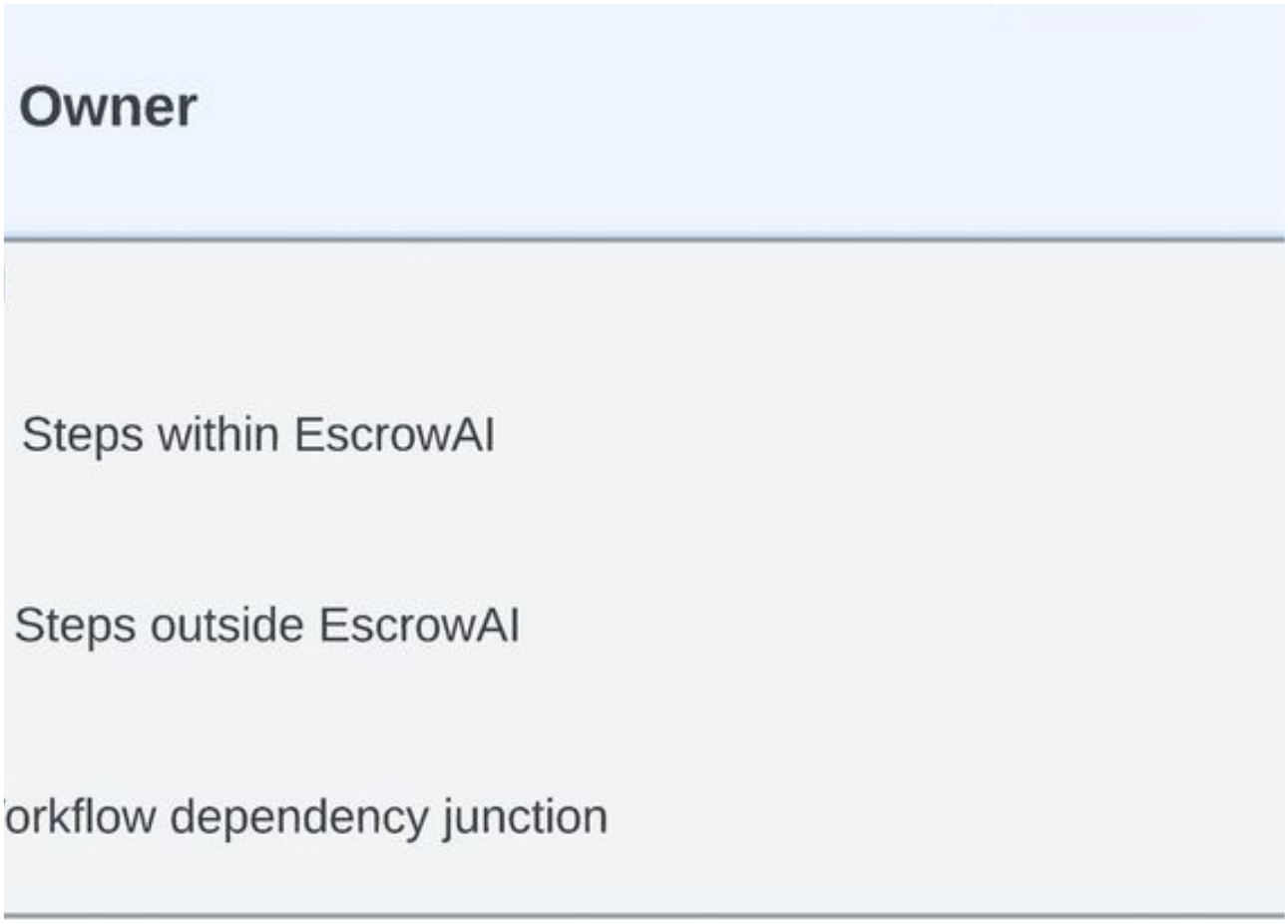
An algorithm or query is encrypted to protect the model's intellectual property and is brought by EscrowAI into a confidential compute node in the Data Steward's compliant cloud.

Algorithm owner essential steps

The Algorithm Owner project workflow is shown in the flowchart below, with corresponding instruction index on the left.


Create CEK
Prepare Algorithm
Upload Data Specification to EscrowAI
Wrap CEK
Encrypt Algorithm


Create Validation JSON
Upload WCEK to EscrowAI
Upload Algorithm zip file to EscrowAI
Upload Validation JSON to EscrowAI
Generate Run Request within EscrowAI
EscrowAI delivers report



Create and upload Data Specification

What is the Data Specification?

The Algorithm Owner (AO) proposes the Data Specification that supports the intent of the project and the implementation needs of the algorithm. In EscrowAI, the Data Specification is vital for the AO and Data Steward (DS) as it ensures that the data meets the requirements without known issues or errors and complies with industry standards and regulations.

For more information about the Data Specification from a Data Steward's perspective, see Curate data per AO Data Specification.

Suggested outline of Data Specification

The topics highlighted below should form the basis of a Data Specification, although the exact form and content are based on the needs and requirements of the collaborating partners. These sections are provided as a guide only.

Section 1: Hypothesis Test and Methodology

This section defines the purpose of the algorithm or query and its expected output.

Section 2: Population Description

This section describes the target population for the algorithm, including demographic information such as age, gender, and ethnicity. It also outlines the inclusion and exclusion criteria for individuals to be included in the dataset.

Section 3: Data Specification

This section details the required data for the algorithm, including the types of data (e.g., text, images, audio) and any specific format requirements (e.g., file type, encoding, ontology). It also describes any pre-processing steps that may be necessary.

Section 4: Truth Standard

This section describes the method for determining the ground truth for the dataset: the correct classification or label for each data record. It also outlines the criteria for each classification or label.

For example, a dataset of chest X-ray images labeled as either "normal" or "abnormal" for a pneumonia detection algorithm. The ground truth for this dataset could be determined by having a team of radiologists visually inspect each image and classify it as either normal or abnormal based on specific criteria, such as the presence of infiltrates, consolidation, or pleural effusion.

Section 5: Data Set Validation

This section outlines the method for validating the dataset, which may involve randomly sampling and inspecting a subset of data points to ensure accuracy and consistency with the population, inclusion and exclusion, data content and form, and ground truth.

Upload Data Specification

To upload the data specification:

1. Go to the desired project page.
2. Click **NEW DATA SPECIFICATION**.
3. Fill in the required fields and upload your Data Specification report.
   The **Name** field is auto-populated from the project name.
4. From your local computer, select or drag-and-drop the desired data specification.
5. Click **SUMBIT** and confirm to save or **CANCEL** to discard the details.

Integrate the EscrowAI SDK

EscrowAI SDK reference implementation

The EscrowAI SDK provides a simple interface to integrate with Confidential Computing environments. The SDK makes it easy to develop and test your algorithms both in local and containerized environments.

This document details a reference implementation, and describes the procedure required for Algorithm Owners to deploy their algorithms on the EscrowAI platform. The reference implementation utilizes the EscrowAI SDK (also known as BKRuntimeSDK). The SDK provides a simple interface to integrate with Confidential Computing environments and makes it easy to develop and test your algorithms both locally and in Docker containers running on a Trusted Execution Environment (TEE).

This reference implementation will be replaced with your application. You can swap the example application code with your code and you should be up and running very quickly. The algorithm application code runs in a Docker container, the creation of the Docker container is done by EscrowAI. The Algorithm runs in conjunction with a Runtime that is responsible for streaming data to the Algorithm.

High-level steps to integrate the SDK

To integrate EscrowAI SDK into your algorithm, follow the steps below:

1. Import the SDK
2. Instantiate the SDK based on the environment you're working in, it will decrypt the files specified in the `secrets.yaml` manifest file
3. Pass report to the Runtime
4. Review EscrowAI UI to view the report

Reference implementation

The reference implementation is an AI algorithm that interprets an X-ray image for the presence of either Covid, Pneumonia or the absence of both diseases. The reference implementation is made up of the following sub-folders and files, details of which are provided in Create algorithm package.

- Dockerfile
  - The Dockerfile is a helper file that assists in creating the container, packaging the algorithm files, that will execute in the TEE.
- Flask application file (`app.py`)
  - `app.py` file is a Flask application, which is the endpoint to which data is streamed in the TEE.
- `models` sub-folder
  - The models sub-folder holds the model file `multi-class-pg.pkl`.
- bkopen sub-folder
  - Contains the `entrypoint.py` file which executes the algorithm workflow in a sequential manner.
- Secrets file (`secrets.yaml`)
  - `secrets.yaml` file is a secrets manifest which instructs EscrowAI to decrypt the list of files in the manifest for execution in the TEE.

Here are a list of important requirements your code must meet:

1. Data is sent as a POST request to the endpoint https://127.0.0.1:5000/algo. This is how EscrowAI's Runtime can effectively communicate with the Algorithm application in the TEE.
2. Your secrets must be pre-encrypted for the TEE and defined in the `secrets.yaml` file.
3. Your code must run successfully in a container.

Understanding these pre-requisites, the following sections describe the process to convert the reference implementation into your own implementation, to test and run both locally and in the TEE.

Importing the SDK

In the reference implementation the EscrowAI SDK (alias BKRuntimeSDK) is imported in the `entrypoint.py`.

To import the SDK, use the following code:

```
import os
from beekeeperai_runtime_sdk import BKRuntimeSDK
```

Instantiating the SDK

Before using the EscrowAI SDK in your code, you need to instantiate it. Based on the environment you are working in (locally or TEE), the SDK will automatically detect whether it's running in a TEE or not. You need to instantiate the SDK in your entry point before your code is executed. This will allow the SDK to prepare the TEE and decrypt all content inside it.

For example, in your `entrypoint.py` file, you can instantiate the SDK as follows:

```
# If the secret.key is in the cwd, this is not an enclave. We make local development convenient by simulating dec
if(os.path.exists('secret.key')):
    sdk = BKRuntimeSDK(SECRET_KEY_DECRYPTED_LOCATION='secret.key', SECRETS_MANIFEST='secrets.yaml') # Initialize
# If the secret.key is present in /bkopen, this is still not an enclave. We make container development convenient
elif(os.path.exists('/bkopen/secret.key')):
    sdk = BKRuntimeSDK(SECRET_KEY_DECRYPTED_LOCATION='/bkopen/secret.key', SECRETS_MANIFEST='/bkopen/secrets.yaml
# If we don't find the secret.key in the cwd or /bkopen, this is an enclave. We must decrypt the WCEK.
else:
    sdk= BKRuntimeSDK() # Initialize the SDK, Decrypt WCEK, and Warm Enclave
```

After instantiating the SDK in the `entrypoint.py` file, you can execute your algorithm.

Running your algorithm

To execute your algorithm code after initializing the SDK in an entrypoint.py file, you can call the exec() function and provide the file name that contains your algorithm. The path for this file is determined by the structure of your Dockerfile.

Example

```
# We can now pass the baton to the customer's code
exec(open('app.py').read()) # Execute the app.py file
```

Create algorithm package

Algorithm package contents

Below are the step-by-step instructions for an Algorithm Owner (AO) to create and package the necessary files for uploading their algorithm to EscrowAI.

> It is recommended that these steps are complete before using the EscrowAI encryption tool to encrypt the algorithm. The tool produces the secrets.yaml file automatically and bundles the algorithm package into the needed zip file.

1. Gather all the algorithm files. A sample of these files and the corresponding folder structure that constitutes the algorithm framework (provided as a reference implementation) is available upon request:
   - `app.py`
   - `Dockerfile`
   - `models/`
   - `secrets.yaml`
   - `README.txt`
   - `bkopen/`
   - `Requirements.txt`
2. Create or edit the `Dockerfile`:
   - The `Dockerfile` is used to build the algorithm docker container.
   - It should be placed in the same directory as the other algorithm files.
   - Open a text editor and edit a file called `Dockerfile`.
   - The following is a sample `Dockerfile`:

```
FROM python:3.9-slim as bkstart

### BEEKEEPER START LOGIC BEGINS HERE ###
COPY requirements.txt requirements.txt
COPY ./bkopen/beekeeperai_runtime_sdk-0.1.8-py2.py3-none-any.whl /bkopen/beekeeperai_runtime_sdk-0.1.8-py2.
RUN pip3 install --no-cache-dir -r requirements.txt

# Copy critical BeeKeeper entrypoint logic to /bkopen
COPY ./bkopen/entrypoint.py /bkopen/entrypoint.py

# This is not altogether required, but we make executable the following
RUN chmod +x /bkopen/entrypoint.py

# We move our elements of encryption out to /bkopen, nothing here is exposed
COPY secret* /bkopen/

### BEEKEEPER START LOGIC ENDS HERE ###

##############################################################################################

### YOUR DOCKERFILE STEPS BEGIN BELOW
FROM bkstart as modelownerstart

# This is an example application that runs app.py to provide an inference interface to a
# COVID detection model
```

```
# Use working directory /app
WORKDIR /app

# Copy the content of current directory to /app
COPY app.py /app/app.py
COPY models /app/models

EXPOSE 5000
### YOUR DOCKERFILE STEPS END HERE ###

##############################################################################################

### BEEKEEPER ENTRYPOINT STARTS HERE ###
# It is crtiical that your original entrypoint be moved to the entrypoint.sh file
# Before your application starts, all of your secrets will not be redeable at runtime
# and you will therefore require BeeKeeper complete the decryption of your secrets
# prior to running your application
ENTRYPOINT ["python3", "/bkopen/entrypoint.py"]
### BEEKEEPER ENTRYPOINT ENDS HERE ###
```

3. Create or edit the `secrets.yaml` file:
   - This file helps the algorithm know where to locate the confidential files for decryption at runtime.
   - Open a text editor and create a file called `secrets.yaml`.
   - The following code is a sample of the `secrets.yaml` file, it must be modified per your requirements, please add an entry for every file that is encrypted:

```
# Each item in the manifest maps an encrypted source file to an unencrypted destination file
# All paths should reflect the container's layout
secretFiles:
 - ["models/multi-class-pg.pkl.bkenc", "models/multi-class-pg.pkl"]
 - ["models/model-1.pkl.bkenc", "models/model-1.pkl"]

```

> The secrets.yaml file is generated automatically if the EscrowAI encryption tool is used to encrypt the algorithm. See Encrypt algorithm files.

4. Create or edit the "`README.txt`" file:
   - This file should include instructions for running the algorithm and it is optional.
   - Open a text editor and edit the file called "README.txt".
   - Add any necessary instructions for running the algorithm.
5. Create or edit the "`bkopen`" folder:
   - This folder contains helper files to orchestrate the algorithm run in the enclave.
   - Create a folder called "`bkopen`" in the same directory as the other algorithm files.
6. Add or edit files to the "`bkopen`" folder:
   - Add the following files to the "`bkopen`" folder:
     - `bkrun.py`
     - `entrypoint.py`
7. Add files to the models folder:
   - Add the model file(s) to the "`models`" folder. This file(s) contains the confidential algorithm code.
8. Create or edit a "`requirements.txt`" file:
   - This file specifies the libraries that the algorithm code depends on.
   - Open a text editor and create a file called "`requirements.txt`".
   - Add the necessary libraries for the algorithm code to run.

- The following is a sample of the “`requirements.txt`” file:

```
aniso8601==9.0.1
attrs==21.4.0
blis==0.7.7
catalogue==2.0.7
certifi==2021.10.8
cffi==1.15.0
charset-normalizer==2.0.12
click==8.1.3
cryptography==37.0.2
cycler==0.11.0
cymem==2.0.6
fastai==2.6.3
fastcore==1.4.2
fastdownload==0.0.5
fastprogress==1.0.2
flasgger==0.9.5
Flask==2.1.2
Flask-RESTful==0.3.9
fonttools==4.33.3
idna==3.3
itsdangerous==2.1.2
Jinja2==3.1.2
joblib==1.1.0
jsonschema==4.5.1
kiwisolver==1.4.2
langcodes==3.3.0
MarkupSafe==2.1.1
matplotlib==3.5.2
mistune==2.0.3
murmurhash==1.0.7
numpy==1.22.3
packaging==21.3
pandas==1.4.2
pathy==0.6.1
Pillow==9.1.1
preshed==3.0.6
pycparser==2.21
pydantic==1.8.2
pyparsing==3.0.8
pyrsistent==0.18.1
python-dateutil==2.8.2
pytz==2022.1
PyYAML==6.0
requests==2.27.1
scikit-learn==1.0.2
scipy==1.8.0
six==1.16.0
smart-open==5.2.1
spacy==3.3.0
spacy-legacy==3.0.9
spacy-loggers==1.0.2
srsly==2.4.3
thinc==8.0.15
threadpoolctl==3.1.0
torch==1.11.0
torchvision==0.12.0
```

```
tqdm==4.64.0
typer==0.4.1
typing_extensions==4.2.0
urllib3==1.26.9
wasabi==0.9.1
Werkzeug==2.1.2
./bkopen/beekeeperai_runtime_sdk-0.1.8-py2.py3-none-any.whl
```

Create the final package

1. Create a zipfile of all the files in the directory.
2. Upload the zip file.
3. Login to EscrowAI and initiate the upload of the zip file.
4. Follow the instructions provided by EscrowAI to complete the upload process.

Upload the algorithm WCEK

Uploading the Wrapped Content Encryption Key (WCEK) is the means to securely transfer the private Content Encryption Key (CEK) to EscrowAI. The Key Encryption Key is used to encrypt (“wrap”) the private Content Encryption Key (CEK). The WCEK is uploaded in the Keys section of the Project page.

EscrowAI retains only the **last** WCEK uploaded. **Algorithms encrypted with a version of the CEK not encrypted within the current WCEK version cannot be decrypted in the TEE.**

This form of rolling your CEK gives you the power to disallow access to the algorithm by EscrowAI. You can roll your CEK at any time for any reason, but if you want the algorithm to be accessible you must either:

1. re-encrypt it with the new CEK and upload the algorithm package to EscrowAI as a new version, or
2. rewrap the CEK associated with the algorithm and re-upload the WCEK.

See Common encryption steps for the common encryption workflows associated with downloading the Key Encryption Key, generating a Content Encryption Key, and encrypting (wrapping) the CEK.

Steps

To upload the dataset's corresponding WCEK:

1. Go to the desired project page.
2. Click **UPLOAD WCEK**.
3. Fill in the required fields and select the encrypted key file for upload.
4. Check the confirmation that the key is encrypted.
5. Click the **SUMBIT** button to save.

Encrypt algorithm files

Algorithm encryption safeguards the confidentiality of your algorithm's sensitive content during storage, transmission, and execution. Contents are encrypted using your Content Encryption Key (CEK).

Prerequisite encryption setup

Be sure that you have prepared the following:

- Download Key Encryption Key from EscrowAI
- Generate a Content Encryption Key (CEK)
- Generate a Wrapped Content Encryption Key (WCEK)

Steps

To encrypt an algorithm container:

1. From the left, click the shield icon.
2. Select **Algorithm Encryption**.
3. In the **Algorithm Container Directory** field, select/drop the folder containing your algorithm files.
4. From the **Algorithm Container Files** section you can select the files you want encrypted. Only the selected files will be encrypted.
5. In the **Content Encryption Key** field, select/drop your CEK.
6. Click **Encrypt** to begin algorithm container encryption.
7. EscrowAI's encryption tool generates the necessary `secrets.yaml` file, which is included in the zip file.

After encryption is successful you will see a success notification, and the encrypted algorithm files downloaded as a zip file.

Upload algorithm

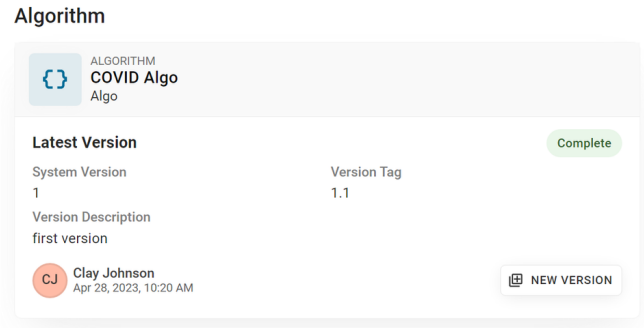
You can upload your algorithm after the AO has created the algorithm package, both the algorithm WCEK and the Validation Criteria are uploaded by the AO, and the Data Attestation Report is uploaded by the DS. The AO uploads the algorithm package zip file to the EscrowAI project space. EscrowAI builds the container, prepares it for use in the TEE, and stores it in the platform's container registry.

Steps

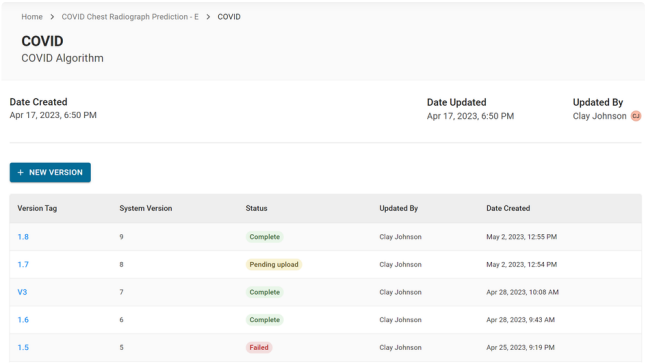
To upload a *new* algorithm:

1. Go to the desired project page.
2. Click **NEW ALGORITHM**.

Use the **NEW VERSION** button on the Algorithm card when uploading subsequent versions of the algorithm.



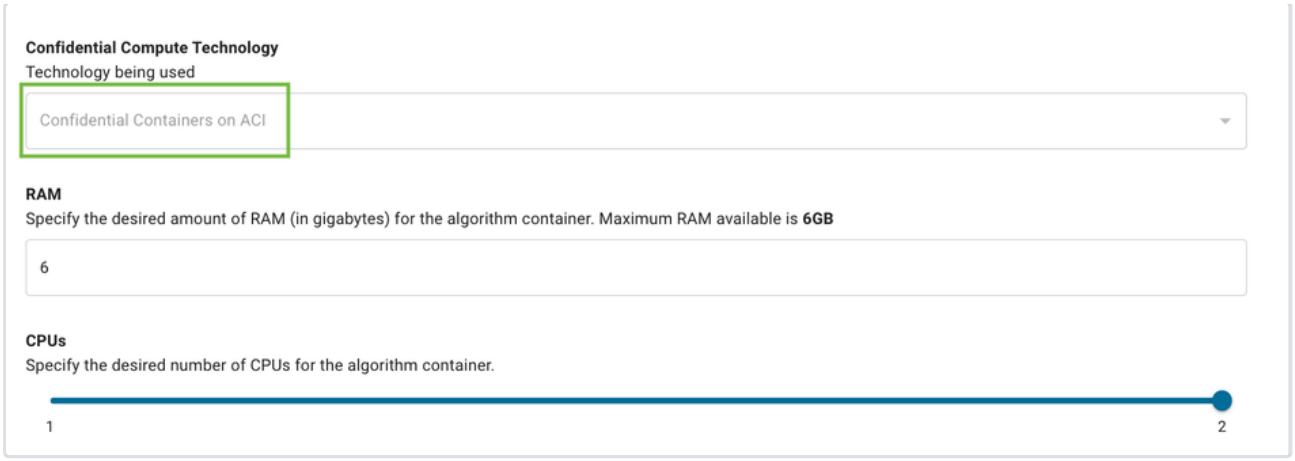
The list of previous versions of the algorithm is available by clicking on the **Algorithm** card. The algorithm versions are arranged in a sorted order, with the latest version being on top.



3. Fill in the required fields. The version tag and algorithm container must be a unique pair. If not, an error message is displayed.

4. Select the desired RAM size and VM type. These settings depend on the type of Confidential Computing technology established for the project by BeeKeeperAI Customer Service: CC ACI or Intel SGX.

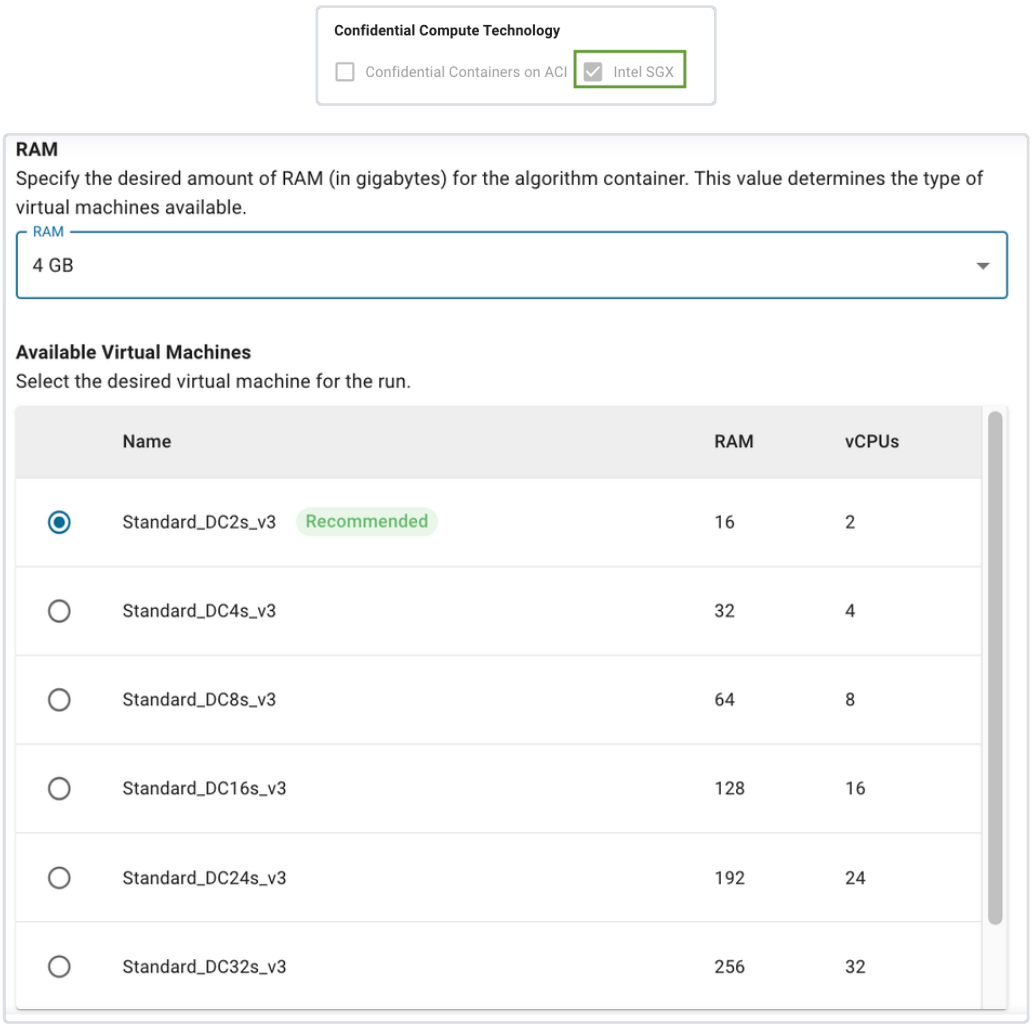
Confidential Containers on Azure Container Instances (CC ACI)



CC ACI RAM and CPU settings

CC ACI resources are controlled by Microsoft Azure. EscrowAI allows selection of the available resources.

Intel SGX



Intel SGX RAM and CPU settings

EscrowAI allows selection of the available Intel SGX resources in the Azure Region specified in the Data Steward profile.

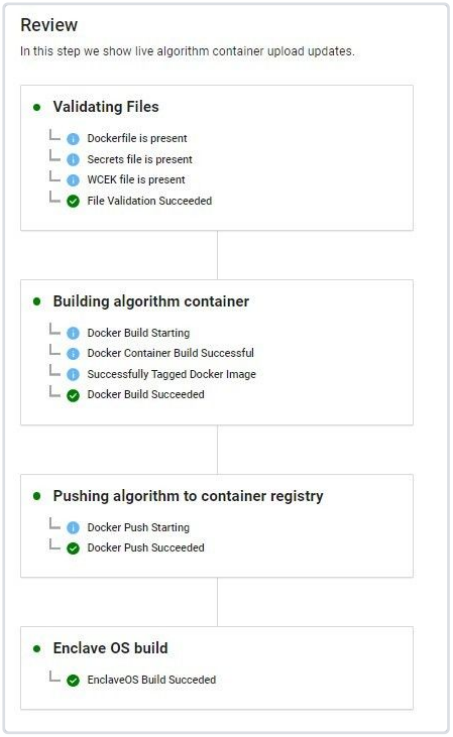
5. Add the zipped algorithm package. You can upload only a single zip file with the required contents.
6. Click **SUBMIT**.

After the zip file is uploaded, asuccess message is displayed.

View Upload Status

To see the status of the upload:

1. Go to the desired project.
2. The upload status is shown on the right sidebar of the **New Algorithm** page.
3. If you navigate away from the **New Algorithm** page before the upload is complete, or to check the upload details of any previous algorithm upload, follow these steps.
   a. Click on the **Algorithm** card in the project page.
   b. Click the **Version Tag**.
      The Version Details window is displayed.
   c. On the upper right, click **VIEW UPLOAD STATUS**.
      The **Review** panel displays each step of the upload status.



Validating files

This is the process of checking if the required files are present in the upload.

Status	Description
Dockerfile is present	A script that contains instructions for building a Docker container image. This step checks if the file is present.
Secrets file is present	A secrets file contains the list of files that have been encrypted by the AO. This file lets the enclave know which files need to be decrypted in the enclave during confidential compute. This step checks if the file is present.

WCEK file is present	A WCEK file contains the encryption key for confidential data. This step checks if the file is present.
File Validation Succeeded	Confirms that all required files are present and the validation has been successful.

Building algorithm container

This is the process of building a Docker container image that contains the algorithm.

Status	Description
Docker Build Starting	Starts the Docker container image building process.
Docker Container Build Successful	Confirms that the Docker container image building process has been successful.
Successfully Tagged Docker Image	Attaching a specific version tag to the built Docker container image.
Docker Build Succeeded	Confirms that the entire Docker container image building process has been successful.

Pushing algorithm to container registry

This is the process of pushing the built Docker container image to a container registry, which is a central location for storing and managing container images.

Status	Description
Docker Push Starting	The process of pushing the built Docker container image to the container registry.
Docker Push Succeeded	Confirms that the built Docker container image has been successfully pushed to the container registry.

Enclave OS build

This is the process of building the secure environment where the algorithm will run when using the Intel® Software Guard Extensions (Intel® SGX) confidential computing enclave.

Status	Description
Enclave OS Build Succeeded	Confirms the process of converting the Docker container in the container registry to include EnclaveOS and pushed it back to the container registry.

Create and upload the Validation Criteria

The algorithm output must be generalized to show performance or data characteristics without including protected privacy information, and the Validation Criteria is the policy that enforces this output. To ensure that the algorithm's report does not include any protected or private information, the Algorithm Owner (AO) must create a Validation Criteria in the form of a JSON file that exactly represents the algorithm's output. The Data Steward must agree to the Validation Criteria form and content in order to initiate a Run. EscrowAI checks the algorithm output within the TEE to ensure compliance with the Validation Criteria before pushing the report out to the AO project space.

For an example of validation criteria in JSON, see Example of JSON Validation Criteria.

Prerequisite before uploading validation criteria

The validation criteria must be approved by the Data Steward (DS) before uploading.

Steps

New Validation Criteria

To upload validation criteria:

1. Go to the desired project.
2. Click **NEW VALIDATION CRITERIA**.
3. Fill in the required fields and upload your validation criteria.
   The **Name** and **Description** fields will be auto-populated based on the project name.
4. Click **SUMBIT** to save.

New Version of Validation Criteria

To add another version of the validation criteria after the first upload:

1. Go to the desired project.
2. Under **Validation Criteria**, click **NEW VERSION**.
3. Fill in the required fields and upload your new validation criteria.
4. Click **SUBMIT** to save.

Example of JSON validation criteria

This is an example of a JSON structure for data validation.

```
{
    "report": {
        "type": "dict",
        "schema": {
            "accuracy": {
                "type": "dict",
                "schema": {
                    "value": {"type": "float", "min": 0, "max": 1},
                    "CI": {"type": "float", "min": 0, "max": 1},
                    "n": {"type": "integer", "min": 1}
                }
            },
```

```
            "specificity": {
                "type": "dict",
                "schema": {
                    "value": {"type": "float", "min": 0, "max": 1},
                    "CI": {"type": "float", "min": 0, "max": 1},
                    "n": {"type": "integer", "min": 1}
                }
            },
            "sensitivity": {
                "type": "dict",
                "schema": {
                    "value": {"type": "float", "min": 0, "max": 1},
                    "CI": {"type": "float", "min": 0, "max": 1},
                    "n": {"type": "integer", "min": 1}
                }
            }
        }
    }
}
```

Configure and submit a Run Request

Once the Algorithm Owner (AO) and Data Steward (DS) have fulfilled the prerequisites for creating a Run Request, the AO can build a Run Configuration and submit a Run Request to the DS.

Steps

1. Click the **NEW RUN CONFIGURATION** button from the Project page.

2. Enter the metadata. The confidential compute technology is populated with the type of technology configured at project initiation.

3. Select the **Run Artifacts** for this run configuration: **Algorithm Version** and **Dataset Version**.

> If a data set SAS URL is expired the dataset version will be greyed out and it will say **Expired**. Consult with the data steward about refreshing the data set URL.

> The combination of Algorithm Version and Dataset Version must be unique compared to prior Run Configurations. An error message is displayed if the combination entered already exists. To select the existing configuration, **Cancel** and navigate back to the Project page. Select the **Run Configuration** card and the **Other Run Configurations** tab to review the list of existing configurations.

4. Click the **SUBMIT** button to save.

5. Once the run configuration has been created, the status will change to **Pending**.

6. Click the Run Configuration card to view the Run Configuration details and send a Run Request

7. Click the **SEND RUN REQUEST** button.

> A given Run Configuration can have only one run active at a time. The **SEND RUN REQUEST** button for a particular Run Configuration will not be active until all confidential computing resources already in use for that configuration (from a previous request) have been terminated.

8. Once the **SEND RUN REQUEST** button has been selected, the status will change to **Run Requested**.

9. An email message with the run request is sent to the DS user(s) assigned to the project. The authorized user will review the request and initiate the run.

10. When the DS initiates the run, the status will change to **In Progress**.

11. If the DS rejects the run, the AO will receive a System Notification.

View run progress, log, and final report

Run status

Once the Data Steward (DS) initiates the run, EscrowAI will start the requested TEE in the DS's environment and initiate the Run.

The run progresses through stages from building the runtime, launching the confidential computing virtual machine, and running the desired task or program. Monitor the top-level status of the Run in the **Run Updates** sidebar. This progression shows the progress of bringing up the TEE and initiating the run.

Run Updates	Description
Run Initiated	The run configuration has been submitted and the process of initiating the runtime environment is about to begin.
Runtime Build	The runtime environment is being built according to the specifications of the run configuration.
Workload Launch	The virtual machine (VM) that will run the algorithm is launched, the TEE is attested, and the run artifacts are brought into the TEE.
Run	The algorithm is currently running on the virtual machine.

Run reporting and cancellation

Status updates, run logs and the final report are displayed in the **Reports** window, as well as the ability for the Data Steward to cancel a run.

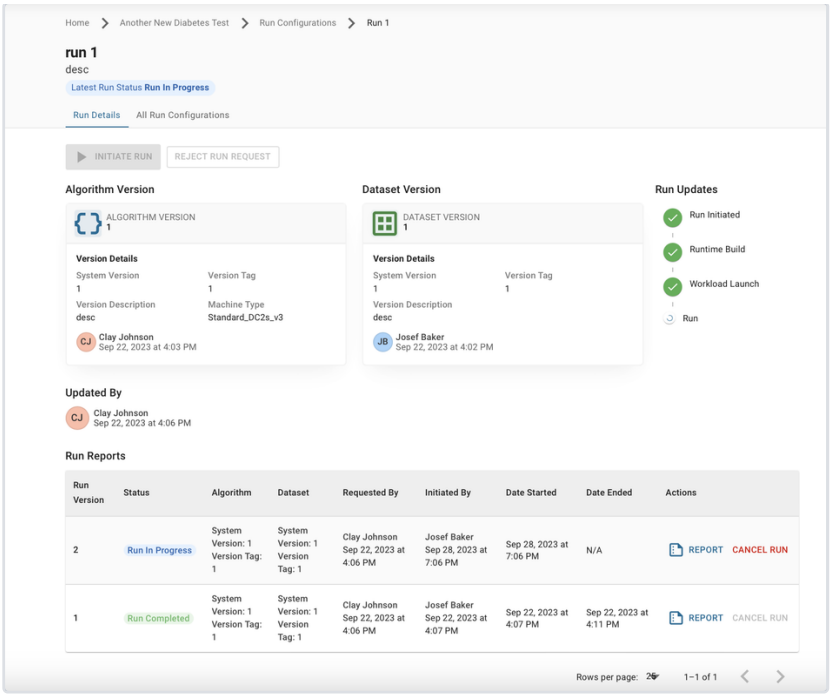
View the status of a run in progress

For a **Run in Progress** the Report will display a running status of the workload, updated at periodic intervals.

1. Go to the desired project.
2. Select the **Run Configurations** card. The **Run Details** tab for the last initiated Run is displayed.
3. The **Run Details** tab displays the Run Configuration details, the status of the **Run in Progress**, and a list of **Run Reports** associated with the configuration.
4. Click on the **Report** link in the Actions column for the Run In Progress to view the detailed run progress logs.

Cancel a run in progress

The Data Steward can cancel a run in progress by clicking the **CANCEL RUN** link in the Actions column under Run Reports.



Completed runs

After a run is finished, its status changes to either **Run Completed** or **Run Failed**. The final report (for the AO only) and run logs (AO and DS) are available from the same **Reports** window.

1. **Run Details**: This tab displays results for the latest Run Configuration. Click on any **Report** link in the Actions column to view the detailed run logs and report for this Run Configuration.
2. **All Run Configurations**: Click this tab to see all run results for any Run Configuration.
   a. Select the card for the Run Configuration of interest from the list of cards.
   b. Scroll to find the heading **Run Reports**.
   c. Click on the **Report** link in the Actions column to view the Report (AO only).
3. Click **Logs** to see the detailed run progress logs.

Run statuses and potential run errors

The tables below display the statuses and potential errors that can occur when the Report tab is selected during the ongoing run. They provide an overview of the current status and any possible issues encountered during the execution.

From VM	Status
In Progress	Compute node enrollment successful
	Connection to BeeKeeperAI container registry succeeded
	Algorithm container launch successful
	Algorithm container is initializing
	Algorithm container is running
Failed	Compute node enrollment failed
	Connection to BeeKeeperAI container registry failed

	Algorithm container launch failed
	Algorithm container initialization failed
	Data container launch failed
	Runtime container failed with a non-zero exit code

From SDK	Status
In Progress	BeeKeeperAI SDK initialized
	BeeKeeperAI SDK successfully initialized, workload starting
	Report has been generated, and validation is starting
Failed	Error while decrypting the secret key. Please encrypt using a new CEK and upload a new WCEK (Wrapped Content Encryption Key) using the EscrowAI-provided public KEK (Key Encryption Key) to proceed.
	While communicating with the Enclave HSM, we encountered an error with Enclave Authentication. Please contact BeeKeeperAI support.
	While communicating with the Enclave HSM, we encountered an error locating your Enclave key. Please try again after confirming that your CEK and WCEK are correct.
	Expected to find a WCEK at {}, but it was not found. This is a fatal error, please contact BeeKeeperAI support.
	While communicating with the Enclave HSM, we encountered an error while trying to decrypt your WCEK. Please try again after confirming that your Content Encryption Key and WCEK are correct.
	While decrypting and decoding the WCEK, we encountered an error. The CEK is invalid. This is likely due to the use of the wrong WCEK. Please encrypt using a new CEK and upload a new WCEK using the EscrowAI-provided public KEK to proceed.
	We successfully decrypted your CEK, but encountered an error while trying to write it to the file {} on the local filesystem. We are unable to proceed. Please contact BeeKeeperAI support.
	The CEK is invalid. This is likely due to the use of the wrong WCEK. Please encrypt using a new CEK and upload a new WCEK using the EscrowAI-provided public KEK to proceed.
	While decrypting your file, we encountered an error. Please try again after confirming that your CEK and WCEK are correct. If the problem persists, please contact BeeKeeperAI support.
	An error occurred while generating your report. Please contact BeeKeeperAI support
	Your report did not pass validation. Please contact BeeKeeperAI support.
	Secrets manifest could not be loaded
	Detected an invalid encryption mode. Please contact BeeKeeperAI support.
	Could not open the decrypted file in the secrets.yaml
	Could not open the encrypted file in the secrets.yaml

EscrowAI Help Center

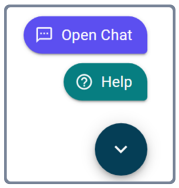
Online help for EscrowAI (this user manual) and other information resources are available via the **Quick actions** on any page in EscrowAI. From the help center, you can also submit questions/issues to BeeKeeperAI technical support.

Steps to access the Help Center

1. In EscrowAI, in the lower right, click the **Quick actions** icon.

Quick actions icon

2. Select **Help**.



Expanded action menu

You are redirected to the EscrowAI Help Center in a new browser tab.

View all resources on the Help Center

Resources within the Help Center are arranged by tiles on the home page. Click on any tile to access that resource. To see all the resources available on the help center, click on the BeeKeeperAI logo in upper left corner of the page.

> After logging in to the help center, you might be directed to the EscrowAI User Manual directly, rather than the Help Center home page. Click the BeeKeeperAI logo to access the home page.

Search the Help Center

You can search for information across all resources in the Help Center, including the EscrowAI User Manual.

1. Login to the EscrowAI help center, as detailed in Steps to access the help center.
2. Go to the help center main page, as detailed in View resources on help center.
3. In the displayed **Search** box, enter the desired keywords.
4. Hit **Return**.

Pertinent results are displayed.

Contact BeeKeeperAI technical support

You can contact BeeKeeperAI technical support if you do not find the information you are looking for or encounter a technical issue.

1. Login to the EscrowAI help center, as detailed in Steps to access the help center.
2. Go to the help center main page, as detailed in View resources on help center.
3. In the lower right, click **Help**.
4. Enter the details in the displayed form:
    - Please be as detailed as possible so we can best help you.
    - Attach any relevant files you think can help isolate/resolve the issue.
5. At the bottom of the form, click **Send**.

Glossary

A

Algorithm Owner

The developer of the mathematical algorithms for analyzing the data maintained by a Data Steward.

Attestation

The process of authenticating a Trusted Execution Environment instance. In Confidential Computing an attestation is the validation of a hardware signed report (an “attestation report”) of the measurements of the Trusted Computing Base.

Azure Blob Storage

Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data. A container organizes a set of blobs, similar to a directory in a file system. A storage account can include an unlimited number of containers, and a container can store an unlimited number of blobs.

Azure Subscription

An Azure subscription is a logical container used to provision resources in Azure

C

Compute Enclave

An ephemeral enclave that is initialized for the purpose of a confidential compute unit of work and destroyed after the confidential compute task ends.

Confidential computing

The protection of data in use by performing computation in a hardware-based, attested Trusted Execution Environment (TEE).

Confidential containers on Azure Container Instances

Confidential containers on Azure Container Instances enable customers to run Linux containers within a hardware-based and attested Trusted Execution Environment (TEE). Customers can lift and shift their containerized Linux applications or build new confidential computing applications without needing to adopt any specialized programming models to achieve the benefits of confidentiality in a TEE. Confidential containers on Azure Container Instances protect data-in-use and encrypts data being used in memory. Azure Container Instances extends this capability through verifiable execution policies, and verifiable hardware root of trust assurances through guest attestation.

Container

A container is a standard unit of software that packages up code and all its dependencies, so the application runs quickly and reliably from one computing environment to another.

Content Encryption Key (CEK)

A CEK is a private, synchronous data-encryption key used to encrypt **and** decrypt data. You create your CEKs using EscrowAI's encryption tool or your organization's key manager. The CEK is used to encrypt a data set and the intellectual property within the algorithm.

D

Data Attestation Report

A report by the Data Steward affirming that the data set curated for the project meets the requirements of the data specification.

Data Set Version

A snapshot in time of elements that make up a data set. The data set version will correspond to specific Data Specification and Data Attestation Report versions.

Data Specification

A document created by the Algorithm Owner that defines the population of the data set (e.g., the patient group) and the data elements needed for each member of the population, the form of each data element, and the form of the data set in its entirety. The data specification also includes the means of identifying the truth by which an inference is tested (as needed), and how the data set must be validated.

Data Steward

Legal entities with the responsibility/liability to protect privacy information in a legal and ethical way.

E

Enclave

An *enclave* is a protected memory region that provides confidentiality for data and code execution. It is an instance of a Trusted Execution Environment (TEE) which is usually secured by hardware.

Enclave Agent

A piece of code that runs in an enclave and checks for carefully controlled tasks to run within an enclave.

Encryption

Encryption is a way of hiding data from unauthorized parties by transforming it into a secret code that only the intended recipients can decipher.

I

Intel SGX

Intel Software Guard Extensions or Intel SGX helps protect data in use via application isolation technology.

J

Just in time model decryption

Pre-signed models are decrypted just in time by the model owner releasing sealed keys after they prove that the correct code is running in an authorized enclave.

K

Key Encryption Key (KEK)

The KEK is the cryptographic key that is used for the encryption of the CEK (“wrapping”) to provide confidentiality and protection for that key, allowing it to be sent to EscrowAI as ciphertext. The KEK is the public half of an asynchronous key pair. In a public-key encryption system, anyone with a public key can encrypt data yielding a ciphertext, but only those with the corresponding private key can decrypt the ciphertext to obtain the original data. In EscrowAI, the private half of this key pair (the half that can decrypt) is retained in the key vault after generation and is only available for decrypting the wrapped CEK within an attested Trusted Execution Environment initiated from the associated project.

L

Linux VM

Azure Virtual Machines are image service instances that provide on-demand and scalable computing resources with usage-based pricing. Linux is a family of operating systems commonly used on servers.

Managed Identity

Managed identities provide an identity for applications to use when connecting to resources that support Azure Active Directory (Azure AD) authentication

M

Manifest

A description of what assets a test run includes. Manifest is an adapted term from “ship’s manifest” which means a list of the shipments or cargo that a vessel is carrying

N

Network Interface

A network interface connects a virtual machine with other services in the virtual network.

Network Security Group

A network security group filters traffic to and from Azure resources

Node

A confidential computing machine within the data steward's security perimeter that has been joined into an enclave and is connected to EscrowAI core infrastructure. The computer communicates with EscrowAI through enclave agent software and has capabilities to setup secure enclaves for data science tasks performed against PHI data.

P

Package manager

A package manager or package-management system is a collection of software tools that automate the process of installing, upgrading, configuring, and removing content on a computer in a consistent manner.

Pre-encrypted Model

A machine learning model that is signing using a mechanism outside of the Beekeeper ecosystem.

Project

A canonical name used across sites to define a relationship between parties collaborating on a joint statement of work. Typically this involves a particular algorithm and a corresponding data set.

R

Resource Group

A resource group is a container that holds related resources for an Azure solution.

S

Sealed Enclave Key

A key that has been encrypted in such a way that only a single enclave can decrypt the key.

Secure Enclave

A protected black box where confidential computing occurs

SSL

[description]

Site

A canonical name for a counter party (e.g., `UCSF` for the University of California, San Francisco)

Storage Account

An Azure storage account contains all Azure Storage data objects, including blobs, file shares, queues, tables, and disks.

T

Trusted Computing Base

Totality of protection mechanisms within a computer system, including hardware, firmware, and software, the combination responsible for enforcing a security policy.

Trusted Execution Environment

A Trusted Execution Environment (TEE) is an environment in which the executed code and the data that is accessed are physically isolated and confidentially protected so that no one without integrity can access the data or change the code or its behavior. A TEE has three primary attributes: data integrity, data confidentiality, and code integrity. Four additional attributes may be present (code confidentiality, programmability, recoverability, and attestability) but only attestability is strictly necessary for a computational environment to be classified as Confidential Computing.

U

V

Virtual Machine

A computer system created using software on one physical computer in order to emulate the functionality of another separate physical computer.

Virtual Network

An Azure Virtual Network or VNet is a fundamental building block for building a private network in the cloud.

VM Disk

A physical disk, such as a hard driver or solid-state driver that is connected to a virtual machine.

W

Wrapped Content Encryption Key (WCEK)

A WCEK is a CEK that has been encrypted (“wrapped”) by a KEK. The WCEK is ciphertext (encoded information).

Z

Zero Trust Architecture

Zero Trust is a security model, a set of system design principles, and a coordinated cybersecurity and system management strategy based on an acknowledgement that threats exist both inside and outside traditional network boundaries. The Zero Trust security model eliminates implicit trust in any one element, node, or service and instead requires continuous verification of the operational picture via real-time information from multiple sources to determine access and other system responses. In essence, a Zero Trust Architecture allows users full access but only to the bare minimum they need to perform their jobs. If a device is compromised, zero trust can ensure that the damage is contained. The Zero Trust Architecture security model assumes that a breach is inevitable or has likely already occurred, so it constantly

limits access to only what is needed and looks for anomalous or malicious activity. Zero Trust Architecture embeds comprehensive security monitoring; granular risk-based access controls; and system security automation in a coordinated manner throughout all aspects of the infrastructure in order to focus on protecting data in real-time within a dynamic threat environment. This data-centric security model allows the concept of least-privileged access to be applied for every access decision, where the answers to the questions of who, what, when, where, and how are critical for appropriately allowing or denying access to resources based on the combination of several factors.
