Functions required for Verification

Image registration:

This module enables registration of watermarked images on the DeepShield platform across multiple programming platforms (initially focusing on Python for the first PoC but should be able to do the same with the Programming language of the final PoC).

- **Register Images and Metadata**
    - **Function**: `register_image(watermarked_image, watermark_data)`
    - **Purpose**: Register the watermarked image into the blockchain and store the related information on IPFS.
    - **Parameters**: The watermarked image and the watermark matrix.
    - **Returns**: Confirmation of successful registration or error details in case of an error

- The registration involves storing a vector representation of the image on a VectorDB. For this first PoC, this would happen on the IABG side, but later this needs to happen on Secublox side.

Image verification process:

The Python library should allow IABG team to interact with the blockchain easily, so we can submit and retrieve data. Here are the key functions it should provide:

- **Function:** `get_evaluation_requests(auth_token)`
    - **Purpose:** Retrieve evaluation requests and associated data that require processing through the verification pipelines.
    - **Parameters:** `auth_token` (optional): A token or API key for authentication and access control.
    - **Returns:** A list of evaluation requests with relevant data.
- **Function:** `get_watermark_information(registered_images_ids)`
    - **Purpose:** Retrieve watermark data for a set of registered images that are potential generators of a candidate image based on similarity search results.
    - **Parameters:** `registered_images_ids` (list): A list of image IDs for which watermark information is requested.
    - **Returns:** Watermark details for the specified images.
- **Function**: `submit_verification_result(request_id, verification_data)`
    - **Purpose**: Submit the results of the verification process (watermark authenticity, generator image if found, semantic integrity) to the blockchain. This function will interact with the smart contract to store the results. Store the image and verification result data on IPFS and return the IPFS URL for access.
    - **Parameters**: Verification data (verification request id, result of watermark check, similarity score, etc.).
    - **Returns**: Transaction ID or status indicating successful submission.

- The functions that IABG is implementing for handling the vectorDB in this PoC should also be taken care of and provided in the library of Secublox later on.

Later improvements:

- Image retrieval needs to happen based on similarity search. IABG could help with this.
- In order to save the confidentiality of the watermarked image and the watermark itself.
    - Image registration happens without the actual image.
    - Watermark verification happens on encrypted data.