**Evaluation**

**Module A.2 - FPGA based Visual Capture Device**

DeepShield: A distributed and cryptographic approach for authenticating digital content, based on Trusted Execution environments (TEE), establishing ownership, and detecting unwanted manipulations this document for evaluation with focus on the Field Programmable Gate Array (FPGA) based visual capture device, together with integration with the overall DeepShield approach.

Stage 1 (Months 1–7, timeboxed, final demonstration 28th Mail 2025):

Research and develop core technologies; conduct initial testing/evaluation and demonstration.

|  |  |
| --- | --- |
| Document No.: | **A2\_FPGA\_VCD\_Evaluation** |
| Issue: | A |
| Date: | 12.05.2025 |

|  |  |
| --- | --- |
|  |  |

**Revision Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Description / Change** | **Modified Chapters** |
| A | 12.05.2025 | Delivered version | All |

Contact: langewisch@secublox.com

# **Table of contents**

[Table of contents 2](#_Toc197972615)

[Introduction 3](#_Toc197972616)

[List of Abbreviations 4](#_Toc197972617)

[Overall DeepShield Approach 5](#_Toc197972618)

[Evaluation A.2 – FPGA VCD 6](#_Toc197972619)

[1. Test-Case: Connect FPGA based image capture device with Ethernet-LAN of Laptop 8](#_Toc197972620)

[2. Test-Case: Register DeepShield platform accounts 9](#_Toc197972621)

[3. Test-Case: Create and share private RSA keys 10](#_Toc197972622)

[4. Test-Case: Configuration of the FPGA image capture device 11](#_Toc197972623)

[5. Test-Case: Create image content by FPGA device 13](#_Toc197972624)

[6. Test-Case: Create digital watermark while routing process and storing image 14](#_Toc197972625)

[7. Test-Case: Register authenticated image data in the decentralized network 16](#_Toc197972626)

[8. Test-Case: Evaluation of the FPGA based device together with climate chamber 17](#_Toc197972627)

# Introduction

This document focusses on the evaluation of the developed module A.2 a technical prototype of the FPGA based visual capture devices together with the integration into the overall DeepShield approach.

**Technology:**

Prototype of the FPGA visual capture device will demonstrate and met following criteria, to support the overall DeepShield goals:

* How image content can be reliably created and authenticated in real-time
* Integration into the overall DeepShield approach
* Different use cases and detailed test steps will be evaluated, how authenticated and secured image data is registered in decentralized Blockchain network by which it is possible to process such tamper-proof content through different platforms like social media or news portals.
* Scalability and adaptability with different digital platforms

## List of Abbreviations

Below in Table 1 is a list of abbreviations used in this document.

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| BC | Blockchain |
| DLT | Distributed Ledger Technology |
| DN | Decentralized Network |
| DSC | DeepShield |
| FPGA | Field-Programmable Gate Array |
| IPFS | InterPlanetary File System |
| KYC | Know Your Customer |
| LLM | Large Language Model |
| NFT | Non-Fungible Token |
| PRNG | Pseudo-Random Number |
| TEE | Trusted Execution Environments |
| TRL | Technology Readiness Level |
| VCD | Visual Capture Device |
|  |  |

## Overall DeepShield Approach

The overall DeepShield approach focuses on following requirements to demonstrate the defined goals at the end of stage 1 by following functionalities:

|  |  |
| --- | --- |
| **A: Content Creation & Secure Watermarking** |  |
| A user captures a photo or video with on a public blockchain registered device.  These photos or videos are watermarked on-device within a secure, trusted execution environment making use of the private device key and then stored locally or elsewhere. |  |
|  |  |
| **B: Content Registration**  Additionally, to the preregistered public key of the devices, the original watermarked content can be registered on a public blockchain. This allows for ensuring context integrity using further meta information such as reference vector embeddings. |  |
| **C: Content Verification & Context Integrity**  The registered public key enables verification of the content’s authenticity, confirming its origin from a specific device based on the watermark. Degradations in the watermark indicate content manipulation.  Using meta information such as reference vector embeddings, contextual integrity can be verified, in a multilayered approach, helping to prevent disinformation. |  |
| **D: Verification and Revocation via Distributed Ledger**  Public Key Management: Verifiers can retrieve the device’s public key from the blockchain to authenticate the watermark and digital signature, ensuring transparent, verifiable content authenticity.  Revocation Notices: The ledger will contain information about revoked or compromised keys, ensuring that only active and trusted devices can be verified  Smart Contracts for Automation: Implement smart contracts to handle device registration, key updates, and revocation seamlessly |  |

# Evaluation A.2 – FPGA VCD

To investigate the described challenges the DeepShield demonstrator for Stage 1 will be based on following overall approach, divided into following modules:

A diagram of a blockchain network

AI-generated content may be incorrect.

Figure 1: DeepShield – Stage 1 – Main Modules

The overall DeepShield approach consists of the following main modules:

* **Module A (Create & Register Content)**: Authentication by integrating cryptographic, digital identity, watermarking together with FPGA and trusted execution environments logic (A.1: Smartphone app / A.2: FPGA-based machine). In additional signed data will be transferred to the decentralized network (Module B).
* **Module B (DLT Network):** Prototype of a decentralized network, to register digital identities (devices, accounts), image signatures/integrity and image reference content. In additional to verify existing images.

**Module C (Verify Content):** Prototype of the Artificial Intelligence (AI) services to verify and detect deep fakes of images connected with decentralized network.

* **Module D (Management & Integration):** Develop a prototype of the Integration-Framework (Dashboard, APIs, Integration)

This document focuses on the evaluation and testing of Module A.2 – FPGA based Visual Capture Device.

Module A.2 focuses on image authenticated content creation. A user captures an image with a FPGA based Visual Capture Devices, which authenticate and encrypt the image in real-time and register this authenticated content on a public blockchain, for further image processing by external platforms (e.g. social media or news portals or for the defense industry to process image content for multi-domain-operation scenarios).

A diagram of a brand identity

Description automatically generated

Figure 2: DeepShield – Module A.2 – FPGA Module

To evaluate the implementation of Module A.2 following test cases are created to evaluate the results.

**Test-Cases: Stage 1 of Module A.2**

Based on the requirements for Stage 1 of Module A.2 following test cases were created upfront, to evaluate the implementation of module A.2 for final demonstration:

1. Test-Case: FPGA based image capture device connected by Ethernet-LAN with the Routing-Node provided by a Laptop (Macbook Pro)
2. Test-Case: User account for content registration (“c\_creator\_1”) adds and register three FPGA image capture devices for the DeepShield Smart-Contract. This includes to create a wallet (public / private key) and assign it to the digital Identity of each device.
3. Test-Case: Create and share private RSA keys for encryption of image content by FPGA image capture device and handling by DeepShield platform.
4. Test-Case: Configuration of the FPGA image capture device for integration and communication with the DeepShield platform and decentralized network.
   * Device Connection
   * Signal Settings
   * Colour Settings
   * Settings for climate check
   * Control settings
     + Image encryption
     + Watermarking
5. Test-Case: Create digital image together with metadata by FPGA device
6. Test-Case: Create digital watermark while routing process and storing image for decentralized network
7. Test-Case: Register authenticated image data in the decentralized network for further image processing, based on the DeepShield-Library.
8. Test-Case: Evaluation of the FPGA based device together with a climate chamber to test some ground aspects for

## 1. Test-Case: Connect FPGA based image capture device with Ethernet-LAN of Laptop

Following test seps executed, to evaluate the integration between FPGA VCD and Laptop:

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Activity** | **Results** | **Comment** |
| 1 | Connect FPGA VCD with Laptop (Macbook Pro) with LAN-Ethernet cable |  | Termin command Ifconfig shows successful connection between FPGA VCD and Laptop |
| 2 | Configure FPGA device for data exchange between FPGA VDC and Routing Node |  | Based on FPGA setup and installation guideline successful configuration of the FPGA device and receiving video content and captured image content on the local file system of the laptop |
| 3 | Update Firmware of FPGA VCD |  | New provided firmware successful uploaded to the FPGA VCD from connected Laptop |
| 4 | Update IP address of FPGA |  | Update of new IP address assigned by FPGA VCD from connected Laptop successful |

## 2. Test-Case: Register DeepShield platform accounts

Based on the DeepShield-Flow of the requirement catalogue, following test seps executed, to evaluate the registration of user and device accounts by <https://deepshield.secublox.com>:

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Activity** | **Status** | **Results** |
| 1 | Register content creator by DeepShield operator user account | A green square with a white tick  AI-generated content may be incorrect. | User account “c\_creator\_1” successful added to DeepShield platform, assigned to a Blockchain wallet and registered for DeepShield-Smart-Contract:  <https://sepolia.arbiscan.io/address/0x857aC625A78cA86a330E6cDd2232f831ffDeDa67> |
| 2 | Register FPGA VCD 1 by content creator account | A green square with a white tick  AI-generated content may be incorrect. | FPGA Image Capture Device ”cc\_device\_1” with serial ID “59.0.0.1” successful registered for DeepShield-Smart-Contract:  <https://sepolia.arbiscan.io/address/0xE4db2F851c1c2188FF4CB61E753A6Df66b951307> |
| 3 | Register FPGA VCD 2 by content creator account | A green square with a white tick  AI-generated content may be incorrect. | FPGA Image Capture Device “cc\_device\_2” with serial ID “59.0.0.2” successful registered for DeepShield-Smart-Contract:  <https://sepolia.arbiscan.io/address/0x851D3017b3cC8aaa5123f836c1786dB873bCfA25> |
| 4 | Register FPGA VCD 3 by content creator account | A green square with a white tick  AI-generated content may be incorrect. | FPGA Image Capture Device “cc\_device\_3” with serial ID “59.0.0.3” successful registered for DeepShield-Smart-Contract:  <https://sepolia.arbiscan.io/address/0x851D3017b3cC8aaa5123f836c1786dB873bCfA25> |

## 3. Test-Case: Create and share private RSA keys

To en- and decrypt image content along the image processing from creation until verification private RSA keys are requirement. To evaluate to create and share RSA keys between DeepShield Platform and FPGA VCD following test steps are executed:

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Activity** | **Results** | **Results** |
| 1 | Login by DeepShield operator account and open RSA Dashboard | A green square with a white tick  AI-generated content may be incorrect. | User account “secublox” successful logged into the DeepShield platform end opens the RSA Dashboard: |
| 2 | Create private RSA keys | A green square with a white tick  AI-generated content may be incorrect. | All RSA Keys created for FPGA Image Capture Device 1, 2 and 3 |
| 3 | Export RSA keys | A green square with a white tick  AI-generated content may be incorrect. | For each FPGA Image Capture Device (1,2 and 3) assigned keys successful exported based on JSON format as following file example:  AES\_Keys(cc\_device\_1).json  with following data structure:  {  "content\_creator\_type": "device",  "content\_creator\_username": "cc\_device\_1",  "AES256\_keys": {  "key\_1": "1ae6ed81…x0279f71",  "key\_2": "40c44d90…90a3ade4",  …  }  } |
| 4 | Import RSA keys for FPGA VCD | A green square with a white tick  AI-generated content may be incorrect. | Open management application of FPGA VCD and switch to tab “Control”.    By “Load Keys JSON” RSA keys for each single FPGA VCD (1,2 and 3) successful imported and shared with each registered device. |

## 4. Test-Case: Configuration of the FPGA image capture device

To setup each registered FPGA VCD to process image for the DeepShield approach, FPGA VCD should be configured. To evaluate the configuration of each FPGA VCD, following test steps are executed:

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Activity** | **Results** | **Results** |
| 1 | Check FPGA VCD route node connection | A green square with a white tick  AI-generated content may be incorrect. | Status of the FPGA VCD connection shown under management application of tab “Device”.    After successful setup based on the guideline, the connection status with the route node successful shown by field “Model” and “Serial”. |
| 2 | Signal Settings | A green square with a white tick  AI-generated content may be incorrect. | Status of the FPGA VCD signals shown under management application of tab “Signal”.    All required signal information successfully shown. |
| 3 | Color Settings | A green square with a white tick  AI-generated content may be incorrect. | Status of the FPGA VCD colors shown under management application of tab “Color”.    All required color properties configured and successfully shown. |
| 4 | Climate check | A green square with a white tick  AI-generated content may be incorrect. | Status of the FPGA VCD climate information shown under management application of tab “Advanced”.    All required climate properties configured and successfully shown. |
| 5 | Control settings | A green square with a white tick  AI-generated content may be incorrect. | Status of the FPGA VCD control information shown under management application of tab “Control”.    All required control properties configured and successfully shown. |

## 5. Test-Case: Create image content by FPGA device

To create image content by FPGA VCD for evaluation, following test steps are executed:

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Activity** | **Results** | **Results** |
| 1 | Start image stream | A green square with a white tick  AI-generated content may be incorrect. | Open “Secublox Viewer” of FPGA VCD management application:    Image stream of the connected and registered FPGA VCD shown by “Secublox Viewer” |
| 2 | Capture Image | A green square with a white tick  AI-generated content may be incorrect. | Press on “Capture” to capture the current visible image.  The captured image is automatically transferred by the Ethernet LAN connection to the local file area of the laptop.  The stored image container includes the raw image data together with the Metadata information (en- or decrypted). |

## 6. Test-Case: Create digital watermark while routing process and storing image

To add a watermark to an image created by FPGA VCD through the routing node for evaluation, following test steps are executed:

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Activity** | **Results** | **Results** |
| 1 | Activate watermarking | A green square with a white tick  AI-generated content may be incorrect. | Open “Secublox Viewer” of FPGA VCD management application and activate the checkbox of field “Watermark Enabled”:    Capturing images with watermark now active and successful configured. |
| 2 | Capture Image | A green square with a white tick  AI-generated content may be incorrect. | Open “Secublox Viewer” of FPGA VCD management application and click on “Capture”:    Select directory: |
| 3 | Select Directory | A green square with a white tick  AI-generated content may be incorrect. | After capturing image content, select a directory to store the data to the local directory. |
| 4 | Image Content stored | A green square with a white tick  AI-generated content may be incorrect. | After capturing and storing an image to the local directory, the image container with following information is available for further image processing (example by FPGA VCD number 3):  encrypted  Metadata  20250506\_230816\_59003\_01\_meta.bin  Image  20250506\_230816\_59003\_01.bin  decrypted  Metadata  20250506\_230816\_59003\_01\_meta.txt  Image  20250506\_230816\_59003\_01.jpg  Naming logic of the file:  DATE\_TIME\_DEVICEID\_RsaKeyIndex |

## 7. Test-Case: Register authenticated image data in the decentralized network

Register authenticated image data in the decentralized network through the routing node for further image processing for evaluation, based on the DeepShield-Library. Following test steps are executed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Activity** | **Results** | **Results** |
| 1 | Start Routing node | A green square with a white tick  AI-generated content may be incorrect. | The routing node which is connected with the decentralized network by the DeepShield library monitors the local directory about new received captured images, started by following python script:  python images\_FPGA.py    After the log information of the terminal window is showing “Waiting for new images -------” the routing node is successful started and monitors to receive new images. |
| 2 | Register captured images for decentralized network | A green square with a white tick  AI-generated content may be incorrect. | After the routing node identifies a new image it registers the image content in decentralized network.  A successful registration in the Blockchain of an captured image by the FPGA VCD is shown by the example of following Blockchain transaction:    <https://sepolia.arbiscan.io/tx/0x919a92b91ef64b4273164aca2673b2b59085de2bf14f9a3a1f11c0b7028fffa3> |

## 8. Test-Case: Evaluation of the FPGA based device together with climate chamber

Evaluation of the FPGA based device together with a climate chamber to test aspects about space grade readiness.

Following figures shows the main test configuration of each test of the FPGA VCD together with the climate chamber successfully executed.

|  |  |  |
| --- | --- | --- |
| Cimate test: +75°C | Cimate test: -55°C | Cimate test: +85°C |

Following test steps are executed and detailed measurement of the climate chamber tests together with the FPGA VCD documented in following document: FPGA\_VCD\_Climatic\_Chamber\_Test\_Results.xlsx

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Activity** | **Results** | **Results** |
| 1 | Temperature set point at 55°C for 45 minutes | A green square with a white tick  AI-generated content may be incorrect. | Imaging and encoder successful working along the full timeline.  FPGA temp process value successful. |
| 2 | Temperature set point from 60°C up to 85°C for 135 minutes | A green square with a white tick  AI-generated content may be incorrect. | Imaging and encoder successful working along the full timeline.  FPGA temp process value successful. |
| 3 | Temperature set point from 85°C down to -55°C for 102 minutes. | A green square with a white tick  AI-generated content may be incorrect. | Imaging and encoder successful working along the full timeline.  FPGA temp process value successful. |
| 4 | Power to FPGA VCD cut off for 27 minutes. | A green square with a white tick  AI-generated content may be incorrect. | Power to camera cut off successful and allowed internal components to take the ambiant temp successful. |
| 5 | Cold start of FPGA VCD from -55°C up to 100°C for 135°C minutes | A green square with a white tick  AI-generated content may be incorrect. | After cold start of FPGA, power is provided and stabilized instantly successful. |
| 6 | Temperature set point >100°C | A green square with a white tick  AI-generated content may be incorrect. | Internal Temp of the FPGA went over 100°C.  Components of this type of camera are rated up to 100°C.  Image and encode working “FALSE” based on components specification successful. |
| 7 | Temperate set point from >100°C to 40°C | A green square with a white tick  AI-generated content may be incorrect. | Internal Temp of FPGA <100°C  Imaging and encoder switched back to “TRUE” and the device is successful working. |