D4.2 System architecture

Document Due Date:	31.05.2020 (M09)
Document Submission Date:	29.05.2020 (M09)

Work Package 4:
Platform setup and integration

Document Dissemination Level:	Public

D4FLY - **D**etecting **D**ocument frau**D** and i**D**entity on the fly
This project has received funding from the European Union's Horizon 2020
research and innovation programme under grant agreement No 833704

Abstract

This report introduces detailed documentation of the proposed architecture for the D4FLY system. The system consists of various subsystems and components, which can be configured to meet the needs of the different scenarios described in the Grant Agreement. In order to meet all requirements from partners, end users and other stakeholders within the consortium the architecture is designed on a modular basis. Standalone modules (components) will be developed by individual partners and will form part of one or more of the following configurations: enhanced document verification, biometric corridor, imposter detection, coach checking and enrolment kiosk.

This report presents the setup of the configurations and how they are applied in the different D4FLY use cases. Furthermore, the technologies used for messaging between the components are dscribed.

Project Information

Project Name	Detecting Document frauD and iDentity on the fly
Project Acronym	D4FLY
Project Coordinator	Veridos GmbH
Project Funded by	European Commission
Under the Programme	Horizon 2020 Secure Societies
Call	H2020-SU-SEC-2018
Topic	SU-BES02-2018-2019-2020 Technologies to enhance border and external security
Funding Instrument	Research and Innovation Action
Grant Agreement No.	833704

Document Information

Document reference	**D4.2**
Document Title	**System Architecture**
Work Package reference	WP04 System Architecture
Delivery due date	31.05.2020 (M9)
Actual submission date	29.05.2020
Dissemination Level	Public
Author(s)	Damjan Gicic (VD) Susanne Kränkl (VD) Jindrich Kodl (VD)
Reviewer(s)	Armin Reuter (VD) Henri Bouma (TNO) Bartłomiej Jankiewicz (WAT)

Document Version History

Version	Date created	Beneficiary	Comments
0.1	07.04.2020	VD	Initial draft
0.2	08.05.2020	VD	Revised draft
0.3	12.05.2020	VD	Revised draft after 1st review
0.5	12.05.2020	VD	Revised draft, ready for 2nd review
0.6	18.05.2020	VD	Revised draft based on 2nd review
0.7	20.05.2020	VD	Further edits, draft ready for 3rd review
1.0	28.05.2020	VD	Final edits
1.1	24.06.2021	VD	Minor edits after review

List of Acronyms and Abbreviations

ACRONYM	EXPLANATION
BCSt	Border Control Station
BFM	Biometric Fusion Module
BG	Border Guard
CLI	Changeable Laser Image
EC	European Commission
EU	European Union
D4FLY	Detecting Document frauD and iDentity on the fly
DB	Database
GUI	Graphical User Interface
IMEI	International Mobile Equipment Identity
IR	Infrared
JSON	JavaScript Object Notation
MLI	Multiple Laser Image
MRZ	Machine Readable Zone
NFC	Near Field Communication
OVI	Optically Variable Ink
PAD	Presentation Attack Detection
PROTOBUF	Google Protocol Buffer
ROCA	Return Of Coppersmith's Attack
SC	Sensor Controller
UV	Ultra Violet
VD	Veridos GmbH
VL	Visible Light
WP	Work package
ZMQ	ZeroMQ

Table of Contents

1 INTRODUCTION

1.1 Background

The D4FLY research project aims at exploring and implementing ways to augment the current capabilities and capacities of border authorities in countering emerging threats in document and identity verification methods in manual as well as in highly automated border control points. The major objectives are the research of novel ways of document fraud detection and the exploration of new biometric modalities for seamless border crossing and enhanced verification accuracy. D4FLY focuses on improving the verification processes in various typical scenarios:

- Enhanced document verification of travel documents in the context of document issuance and border control
- Highly automated verification of high volume of travellers at the point of border crossing
- Enhanced document verification and impostor fraud detection at a land border post
- Secure and fast person verification for cross-border coach travel with supporting technology

The objective of this task is to design an overall system architecture, which serves as the underlying basis for the different system configurations in the different scenarios. A modular approach to the system design is chosen to cover the different needs from the different scenarios by allowing for parallel and independent development of the various components.

1.2 Aim of this document

Deliverable D4.2 presents the initial system architecture of the D4FLY System including the different components that will be developed and tested in prototypes and demonstrators. The architecture is based on modules and subsystems that can be configured to meet the requirements for each of the different usage scenarios. During the course of the project the system architecture will be refined reflecting the developments on the individual biometric verification processes; an update on the changes will be given in deliverable D4.6 towards the end of the project.

1.3 Input / Output of this document

The main inputs to the system architecture are the high level functional requirements, that were derived from discussions and workshops with the partners, stakeholders and the end users within the consortium. Furthermore, experiences and lessons learnt from the preceding H2020 project PROTECT (grant agreement no. 700259)[1] were considered during the design of the system architecture.

The output of this document is the first description of the D4FLY system architecture, which should serve as a specification and documentation for the implementation of the initial prototype system as it shall be developed during the D4FLY project.

As changes and refinements are expected throughout the course of the project, this document (due in M9) also establishes the basis for an updated report on the final system architecture

in deliverable D4.6 – “System Architecture 2”, which is due close to the end of the project in M33.

1.4 Outline of the document

The outline of the document is as follows:

- Section 2 gives an overview of the general system architecture design. The concepts of the messaging system and the message serialization mechanism are introduced.
- Section 3 presents the used technologies for messaging and data serialization, i.e. ZeroMQ and Protobuf.
- Section 4 describes the system architecture and proposed use case for the *Enhanced document verification* scenario.
- Section 5 describes the system architecture and proposed use case for the *Highly automated border post* scenario.
- Section 6 describes the system architecture and proposed use case for the *Advanced imposter fraud countermeasures* scenario.
- Section 7 describes the system architecture and proposed use case for the *Coach* scenario.

2 GENERAL SYSTEM ARCHITECTURE

2.1 Architectural requirements

The D4FLY system architecture was designed based on the following main high level requirements:

- The system shall be modular and configurable for each of the scenarios.
- The architecture shall support flexibility in a sense, that components can be executed either on the same hardware or on separate hardware (e.g. separate computers or laptops)
- Communication between the components, specifically between components acting as a "master" and components acting as a "slave", shall be using the same messaging concept and be realised using the same concepts and libraries.
- Message translation at all networked modules shall adhere to the same message synchronization method.
- Component design shall support remote integration testing.
- The architecture design shall support many common operating systems (e.g. Windows, Linux, Android, etc.), allowing the developer to choose the best suitable environment for component realisation
- The system shall be failure tolerant, meaning that the system shall still be functional, even if one of the components does not work as expected.

The collection and consolidation of specific and detailed requirements for the D4FLY system, based on user needs and inputs from stakeholders is still ongoing at the time of writing of this deliverable. They will be related to in the second deliverable related to the system architecture towards the end of the project.

The system architecture concept has been designed considering data protection, data privacy aspects, IT security as well as ethical aspects. In the course of the project, all these aspects are continuously monitored and will be reviewed in a privacy, data protection social and ethical impact assessment, which will be reported in separate public deliverables.

2.2 D4FLY solution

The overall goal of the D4FLY project is to enhance the current document and identity verification processes at various types of border control points (e.g. air, land or sea crossings for passengers travelling on foot, by car, coach, ship or plane). In order to achieve that, a collection of operational scenarios and proposed innovative technologies is specified in the project proposal and is to be used as the basis for the D4FLY solution.

The proposed D4FLY solution consists of a family of various components, which are defined by the novel D4FLY as well as existing technologies. The components are organised to form specific system configurations (or sub-systems) that tackle the challenges of each operational scenario. Since the addressed scenarios have elements in common, their system configurations also share some components. The overview of the relationship between configurations and components within D4FLY solution is depicted in Figure 2-1, where each configuration corresponds to the proposed solution for respective scenario. The various components in this figure and their functionality is later described in greater detail in section 2.6.

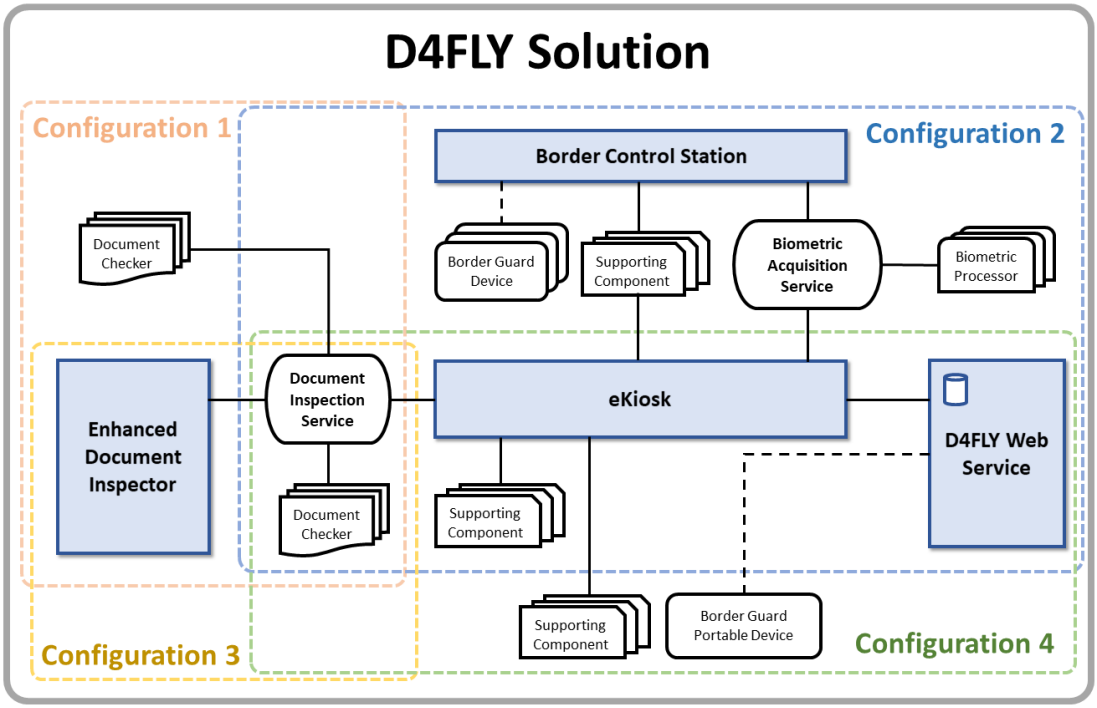

FIGURE 2-1: D4FLY SOLUTION

2.3 System modularity

The main design concept that is used in the D4FLY system is modularity. Therefore, the proposed system is designed in modular form, where separate functionalities and components can be described, developed and tested separately. This approach leaves the developer of each component freedom to choose the platform, programming language and development environment, which is deemed optimal for the specific component. The components themselves are developed as standalone modules, enabling independent operation of the main functional components, even if the component is not functioning as an integral part of the system. This approach poses some challenges to the system integration and enforces a strict definition of interfaces and communication protocols between the components.

One of the advantages of this approach is that in case of malfunction or slow execution of one component, the operation of other components and the overall system is not affected. The modular design employs the idea of distributed processing (i.e. processes are executed over multiple computers or processors) using client-server and master-slave communication models between networking components.

The same design concept was used in the antecedent PROTECT project where such approach was proven to be the best fit for H2020 projects with distributed component development amongst multiple partners. This design removes many dependencies during the development process, giving all the partners freedom to choose best platform for their tasks. However, in order for the interaction between the individual nodes of the distributed system to be effective, few constraints have to be introduced:

- All networked components (e.g. server and clients) in the system shall support a standardised messaging system
- All networked components in the system shall support a standardised message serialization mechanism

2.4 Messaging system

The selected messaging system is based on standard internet network protocol allowing the same communication mechanisms to be used, no matter if the communicating components are implemented on different computers or on the same computer. As the data being exchanged between the components can become fairly large, efficient and fast communication is one of the most important architectural goals. This is achieved through a unified messaging system, which is responsible for transferring data between two or more networked components, allowing the components to focus on their intended function without additional burden of data transmission and data sharing implementations. Messaging within the distributed system utilises the concept of reliable message queuing and provides temporary message storage, where the messages are queued asynchronously. There are two types of messaging patterns which are used in this project:

1. *"request-reply" (req-rep)*: one component sends a request to another for desired data or action. The contacted component responds by sending appropriate message (e.g. data, acknowledgment of reception, etc.)
2. *"publish–subscribe" (pub-sub)*: one-way data distribution, where the server pushes updates to a set of clients.

2.5 Message serialization

Messages sent between the system components need not only to conform to the correct messaging protocol, but also to the correct data format for each transmitted message, introducing the need for a unified message serialization format. In a nutshell, message serialization is a process of translating data structures or object state into a series of bits that can be stored or transmitted and later reconstructed (deserialization). Standardisation of this process shall avoid message synchronization or compatibility issues even when utilised across different computer environments.. When the resulting series of bits is re-read by the receiving module according to the serialization format, it can be used to create a semantically identical clone of the original object.

2.6 List of components

This section introduces components that make up the working system for each scenario, and that act as system modules with some specialised function. In D4FLY all main components can be split into six categories according to their role within the planned system:

- Core components
- Service components
- Document checkers
- Biometric processors
- Border guard terminals

- Supporting components

Each category is described in their respective subsections below.

2.6.1 Core components

The core components establish the base of a configuration or a major phase (e.g. traveller enrolment, traveller verification, document inspection, etc.) within the configuration. Their role is to tie together and manage other components. These are:

- **eKiosk** – manages the enrolment process
- **Enhanced Document Inspector** – manages the document (passport) inspection and verification process
- **Border Control Station** – manages the corridor verification process
- **D4FLY Web Service** – hosts secure database for traveller data and manages the authorised client access as well as some additional services

2.6.2 Service components

The service components manage connection and data exchange between number of components of similar function (e.g. document checkers, biometric processors). They can also be thought of as hubs for particular components. The proposed service components are:

- **Document Inspection Service** – manages document checker type of components
- **Biometric Acquisition Service** – coordinates biometric processors during enrolment and verification phases

2.6.3 Document checkers

Document checkers are remote components used for document content verification. These are:

- **PrintTech Recognizer** – recognises printing technique on a provided copy of passport holder page. More details can be found in D8.2.
- **TravelPattern Extractor** – handles stamp extraction from a provided copy of passport visa page. More details can be found in D8.5.
- **Kinegram Checker** – recognises and checks kinegrams on a provided copy of passport holder page. More details can be found in D8.2.
- **SecElement Checker** – recognises and checks passport security elements (OVI, MLI, MagicID ….) from a provided copy of passport holder page. More details can be found in D8.2.
- **FaceMorph Checker** – evaluates provided passport chip face image for morphing. More details can be found in D7.2.
- **ROCA Checker** – checks passport signer certificates. More details can be found in D8.7.

2.6.4 Biometric processors

Biometric processors are all components which are used for biometric data processing. The following list uses a Sensor Controller (SC) as the general term for a component used to acquire images, extract biometric templates and match biometrics of the same type. This group also includes all Presentation Attack Detection (PAD) components that are associated with their respective SCs. The list of components follows:

- **Iris SC** – enrols and verifies travellers irises
- **Somatotype SC** – enrols and verifies travellers somatotype
- **Face 3D** – enrols and verifies travellers face in 3D
- **Thermal Face SC** – enrols travellers face in the thermal spectrum and verifies it in the visible spectrum
- **Thermal PAD** – detects biometric spoof based on thermal imaging
- **Mobile SC** – enrols and verifies travellers based on data collected from mobile phone sensors
- **Biometric Fusion Module (BFM)** – fuses all biometric verification results from the Sensor Controllers.
- **Face Matcher** – matches travellers face with the chip face image from a passport
- **Blood flow PAD** – detects biometric spoof based on face blood flow screening

2.6.5 Border guard terminals

Border guard terminals cover applications used on various devices by the border guards to help them to perform their duty. These are:

- **Border Guard Monitoring App** – an application used by a border guard to monitor the biometric verification process
- **Border Guard Verification App** – a traveller verification application on a mobile device used by a border guard in multiple traveller verification

2.6.6 Supporting components

Supporting components are used for correct function and integration of other components within the working system. These are:

- **En/Decryptor** – encrypts and decrypts data
- **Indexer** – retrieves indexes
- **Secret Key Provider** – retrieves secure key
- **Gate Controller** – operates the corridor gates
- **Person Tracker** – tracks the person inside the corridor
- **Image Provider** – operates image acquisition
- **Passport Reader** – reads content of an e-passport (image scan and/or electronic data)
- **Ticket Service Agent** – communicates with a ticketing web service
- **NFC Communicator** – communicates with the smartphone via NFC

3 USED TECHNOLOGIES

Reflecting on past experiences from the H2020 project PROTECT (grant agreement no. 700259), the D4FLY system utilises different technologies for messaging and data serialization in order to increase performance, efficiency, flexibility and communication reliability.

3.1 Challenges identified in PROTECT project

The implementation of the messaging standard in the PROTECT project between the modules utilised the widely used WebSocket protocol, which is frequently used by web browsers to listen to server-side events. Despite its efficiency and performance, issues are surfacing when large messages (byte arrays of >300kB) are being sent. Furthermore, when data is transmitted, the WebSocket mechanism attaches additional data to each transmitted packet, meaning that transmitted data size is significantly bigger than the original data size. This causes poor efficiency when sending large amounts of data via mobile network as well as poor reactivity of the system where near real-time responsiveness is needed. Finally, the WebSocket protocol allows for different implementations of message generation. During multi-partner development this could be problematic, because different messaging solutions might make connections and data transactions unreliable, resulting in instability.

To exchange the data between the remote components the PROTECT project utilised the JSON format, transmitting the message data as a human-readable plain text. The messages and their contents were defined in the PROTECT system architecture deliverable, however, each partner had to implement their own method of creating and parsing each message string. Hence in projects with distributed development many different solutions to message construction and deconstruction might be developed, potentially causing message compatibility issues. This became apparent during the integration phase, which identified inconsistencies in the partner generated messages (typos, field names, etc.) Furthermore, since the generated message strings were in human readable form and translated to very large byte arrays, the overall system performance and efficiency suffered due to extended encoding, decoding and transmission times.

3.2 ZeroMQ

D4FLY system architecture replaces the WebSocket communication protocol with ZeroMQ (ZMQ)[2]. ZMQ is a high-performance asynchronous messaging library, aimed at use in distributed or concurrent applications. Unlike WebSocket, ZMQ has much less data overhead, i.e. smaller demand on performance resources, making it much more efficient when transmitting data through a mobile network.

Importantly, ZMQ supports the required messaging patterns specified by the system architecture, i.e. *request-reply* (see Figure 3-1) and *publish-subscribe*. Another important advantage of ZMQ lies in the availability of multi-language APIs and support of most modern operating systems (e.g. Linux and Windows), something that is essential for a multi-partner project like D4FLY.

FIGURE 3-1: REQ-REP PATTERN WITH ZMQ

3.3 Protocol Buffers

Protocol Buffers (protobuf) is an open-source language- and platform-neutral binary encoding format for serializing structured data, proposed and developed by Google[3]. The method is described in the Google documentation: "The method involves an interface description language that describes the structure of some data and a program that generates source code from that description for generating or parsing a stream of bytes that represents the structured data."[4] See Figure 3-2.

Since this method unifies the message generation for all developers the risk of inconsistent definitions or typographical errors is eliminated. Additionally, the byte stream that is created by using the protobuf library is only slightly larger compared to the original data.

All message specifications (data structure, etc.) for the inter-component communication within the system are consolidated and defined by the system architecture team, and distributed amongst all partners (on EMDESK) for implementation in their respective components.

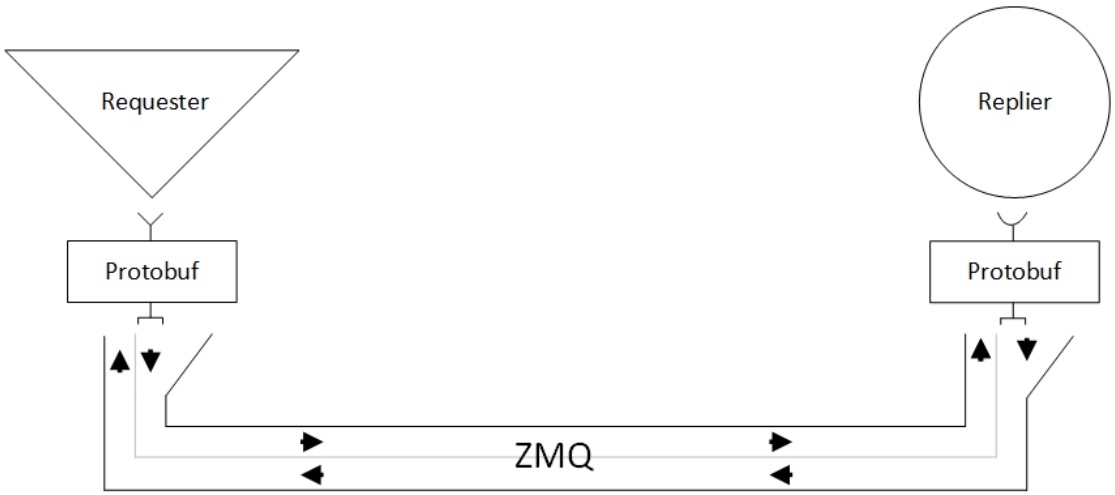

FIGURE 3-2: PROTOBUF COMMUNICATION BETWEEN DIFFERENT PLATFORMS

4 CONFIGURATION 1: ENHANCED DOCUMENT VERIFICATION

Configuration 1 in the scope of the D4FLY solution (Figure 2-1) proposes the system configuration for Scenario 1. As defined in the Grant Agreement, Scenario 1 is aimed at enhancing the document verification process by researching and developing automated methods, which would improve the passport check procedures as they are done today. Scenario 1 uses the document checker modules (listed in Section 2.4.2).Some of the explored methods include:

- Automatic recognition and analysis of the printing technology of a passport holder page
- Automatic recognition and analysis of kinegrams used on passport holder page
- Automatic recognition and analysis of other security elements (e.g. OVI, MLI/CLI)
- Automatic recognition of stamps on visa pages and extraction of travel pattern
- Automatic recognition and checks of digital information and conduct of risk analysis

4.1 Work cycle of document verification in Scenario 1

The detailed use cases are described in D4FLY Task 2.1 in WP2 and can be found in Deliverable D2.1 due in M15. For better understanding of the described system configuration for this scenario, the main process steps of scenario 1 is described here:

1. Border guard (BG) requests the passport from a traveller.
2. The traveller hands out his/her passport to the BG.
3. BG places the passport holder page on the passport reader.
4. SYSTEM – Scans holder page in UV, IR and VL spectrum and extracts MRZ lines and sends the scanned data to the advanced passport checker components for verification
5. SYSTEM – Reads the digital content and sends the data to particular passport checker
6. SYSTEM – Displays all check results for passport holder page on the user interface.
7. SYSTEM – Instructs BG to scan all stamped visa pages.
8. foreach (scanned stamped visa page)
    a. BG places stamped visa page for scanning
    b. SYSTEM – Sends scanned image to remote component for stamp extraction
    c. SYSTEM – Receives stamps in digital format and stores them locally
9. BG instructs the system to evaluate travel pattern.
10. SYSTEM – Sends all stored digitalised pages to the remote component for travel pattern evaluation
11. SYSTEM – Displays travel pattern to BG

4.2 System configuration of document verification in Scenario 1

The general system architecture solution for the enhanced document verification scenario is summarised in Figure 4-1.

The system consists of the BG workstation, which is composed of the Enhanced Document Inspector as a main component with Passport Reader component and Document Inspection Service component connected to it. The Document Inspection Service further connects to the various server components that perform specialised checks and operations. These components belong to the passport checker components category and comprise of Kinegram

Checker, FaceMorph Checker, SecElement Checker, ROCA Checker, PrintTech Recogniser and TravelPattern Extractor. The execution architecture of the system showing the components and their connections is depicted in the component diagram (Figure 4-2).

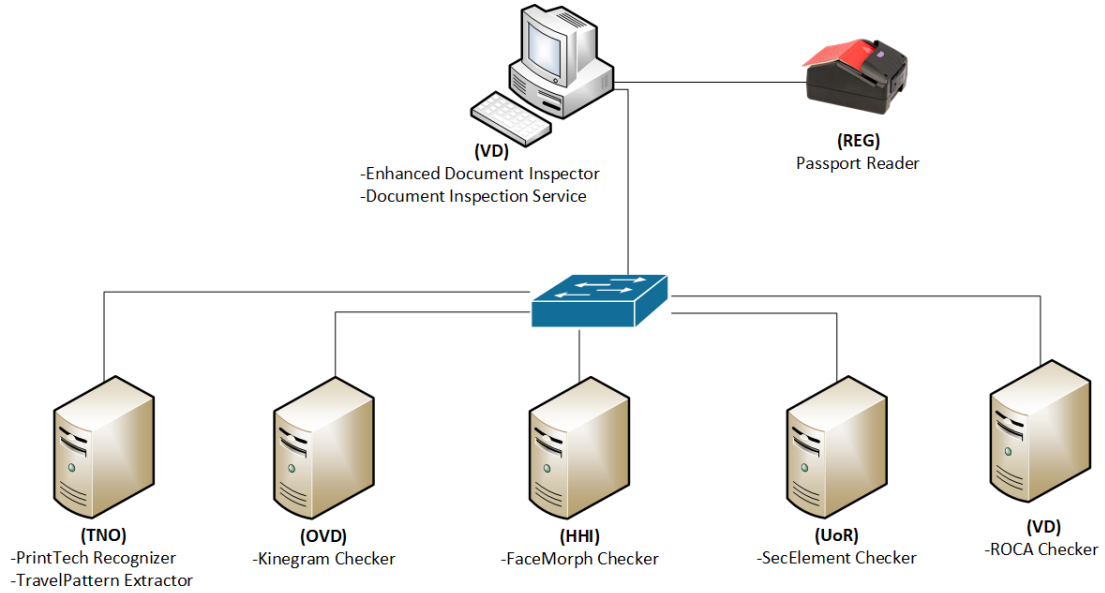

FIGURE 4-1: SYSTEM CONFIGURATION FOR SCENARIO 1

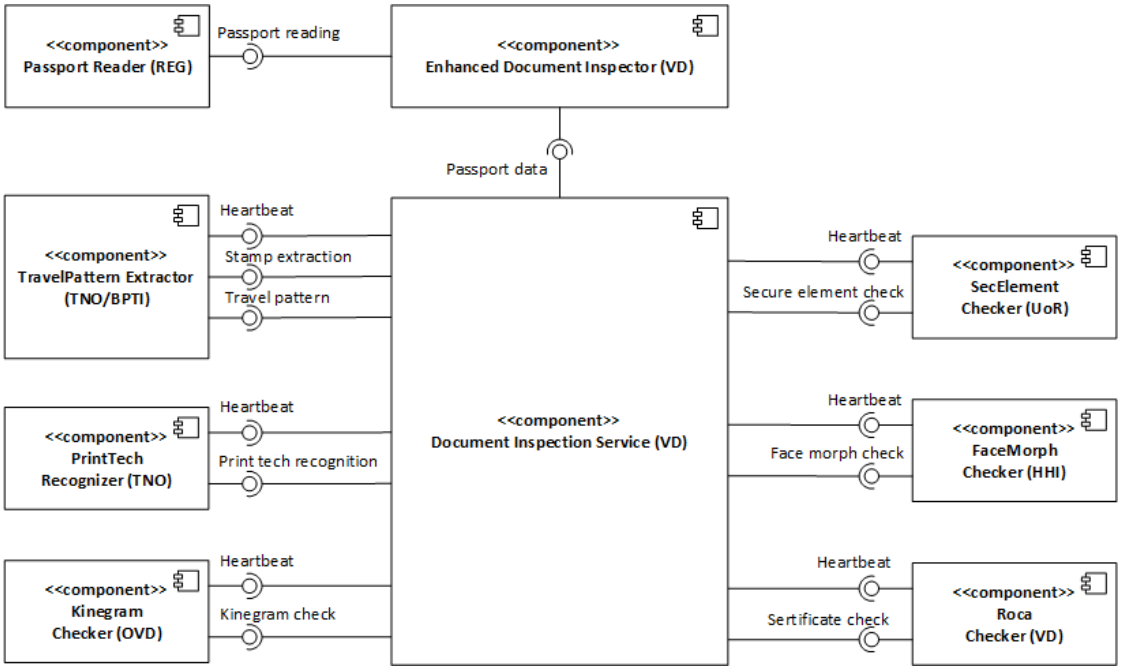

FIGURE 4-2: COMPONENT DIAGRAM FOR SCENARIO 1

The passport verification process itself cycles through three distinct phases:

1. Analysis of the visible passport elements
2. Analysis of the digital passport content

3. Analysis of stamps and travel pattern evaluation

During each phase the Document Inspection Service (client) component distributes the data obtained from the Enhanced Document Inspector to the various checker (server) components.

In the first phase the Document Inspection Service obtains the scan of the holder page captured in visible, infra-red and ultraviolet light and distributes it to other components - PrintTech Recognizer for printing technique recognition, Kinegram Checker for kinegram checking, and SecElement Checker for secure element checking.

In the second phase the Document Inspection Service transmits the facial image, that is stored in the e-passport to the FaceMorph Checker component to check the face image for face morphing effects. It also sends the chip certificates to the ROCA Checker component to check for potential ROCA vulnerabilities.

Finally, in the third phase the Document Inspection Service communicates only with TravelPattern Extractor in order to digitalise all visa page stamps and evaluate travel pattern.

5 CONFIGURATION 2: HIGHLY AUTOMATED BORDER POST

Configuration 2 in the scope of the D4FLY solution (Figure 2-1) proposes the system configuration for Scenario 2 defined in the Grant agreement. Scenario 2 addresses border points with high volumes of travellers arriving in waves and the aim is to automate and speed up the border control procedure.

The proposed system is split into two phases defined by traveller's experience: traveller enrolment and traveller verification within the biometric verification corridor.

During the enrolment phase, data is captured from the travellers passport and following biometric data are captured using the different biometric sensors in the enrolment kiosk:

- Face 2D (used only for passport holder verification)
- Face 3D
- Iris
- Somatotype

After the enrolment is completed, the enrolled data is encrypted and securely transmitted to a database via the proposed standardised messaging system.

The enrolled data are later retrieved when the traveller is about to enter the verification corridor. While the traveller moves through the biometric corridor, the installed biometric sensors capture the same traveller's biometrics and match them with the enrolled data.

5.1 Enrolment process in Scenario 2

5.1.1 Work cycle of the enrolment process in Scenario 2

The detailed use cases are described in T2.1 in WP2 and can be found in D2.1 due in M15. For better understanding of the described system configuration for this scenario, the main process steps of the enrolment in Scenario 2 is described here:

Preconditions:

- eKiosk mobile application installed on travellers phone
- eKiosk mobile application started and configured

Process steps:

1. The Traveller approaches the enrolment kiosk
2. SYSTEM – Displays welcome screen and selection of supported languages
3. Traveller selects the display language
4. SYSTEM – Displays the consent form in selected language
5. Traveller reads the consent form and accepts it by pressing the accept button
6. SYSTEM – Instructs the traveller to tap his/her smartphone on the NFC reader
7. Traveller taps the smartphone on the NFC reader
8. SYSTEM – Receives smartphone unique ID and uses it as index
9. SYSTEM – Instructs traveller to place his/her passport on the passport reader
10. Traveller puts his/her passport on the passport reader
11. SYSTEM – Scans the passport holder page and reads the content of the chip
12. SYSTEM – Instructs the traveller to remove the passport from the passport reader
13. Traveller removes the passport

14. SYSTEM – Instructs the traveller to look into the face capture camera
15. Traveller looks into the camera
16. SYSTEM – Verifies only the live face image with the passport image
17. foreach (available sensor controller)
    a. SYSTEM – Instructs the traveller how to enrol
    b. Traveller is following instructions and enrols
18. SYSTEM – Encrypts the enrolled data
19. [optional] SYSTEM - Instructs the traveller to tap his/her smartphone on the NFC reader again
20. SYSTEM – Receives IMEI and matches with first IMEI, transfers additional information
21. SYSTEM – Sends the encrypted data and index key to D4FLY database
22. SYSTEM – Instructs the traveller that enrolment procedure is successfully completed
23. Traveller leaves the kiosk

NOTE: Step 19 , as well as some other details are dependent on results of other work packages and may change in the course of the project.

5.1.2 System configuration of the enrolment process in Scenario 2

The system configuration for the enrolment procedure in Scenario 2 is illustrated in Figure 5-1. The core of the system is formed by the eKiosk component complemented by various supporting components. Notable supporting components are the Image Provider and Passport Reader that are accompanied by the Face Camera and Regula Reader peripherals respectively. Optionally the Document Inspection Service component may be included in the system configuration. The traveller personal instance indexing is provided by the NFC Communicator component, which acquires the traveller smartphone's unique ID and passes it to the eKiosk component. The Biometric Acquisition Service as master component, also directly connected to the eKiosk, coordinates the enrolment of the biometric modalities and handles exceptions when alarmed by any of the PAD components. In the final stage of the enrolment process the eKiosk sends the encrypted enrolment data to the D4Fly database for secure storage.

The architecture and configuration of the system including the components and their connections is depicted in the component diagram in Figure 5-2.

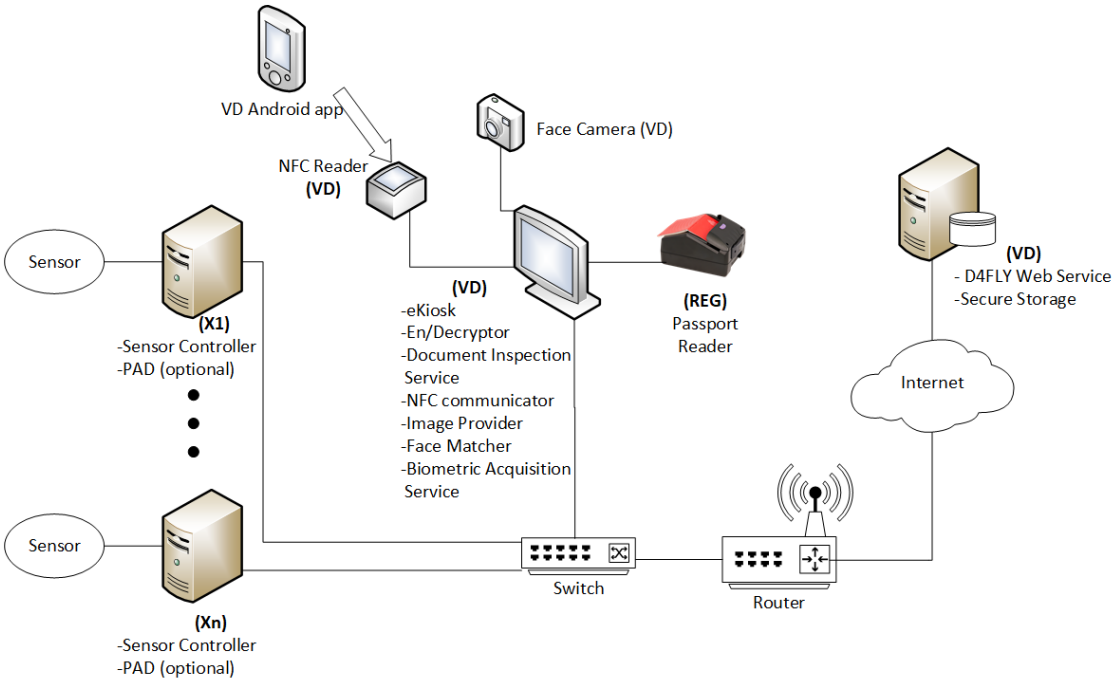

FIGURE 5-1: SYSTEM CONFIGURATION FOR ENROLMENT IN SCENARIO 2

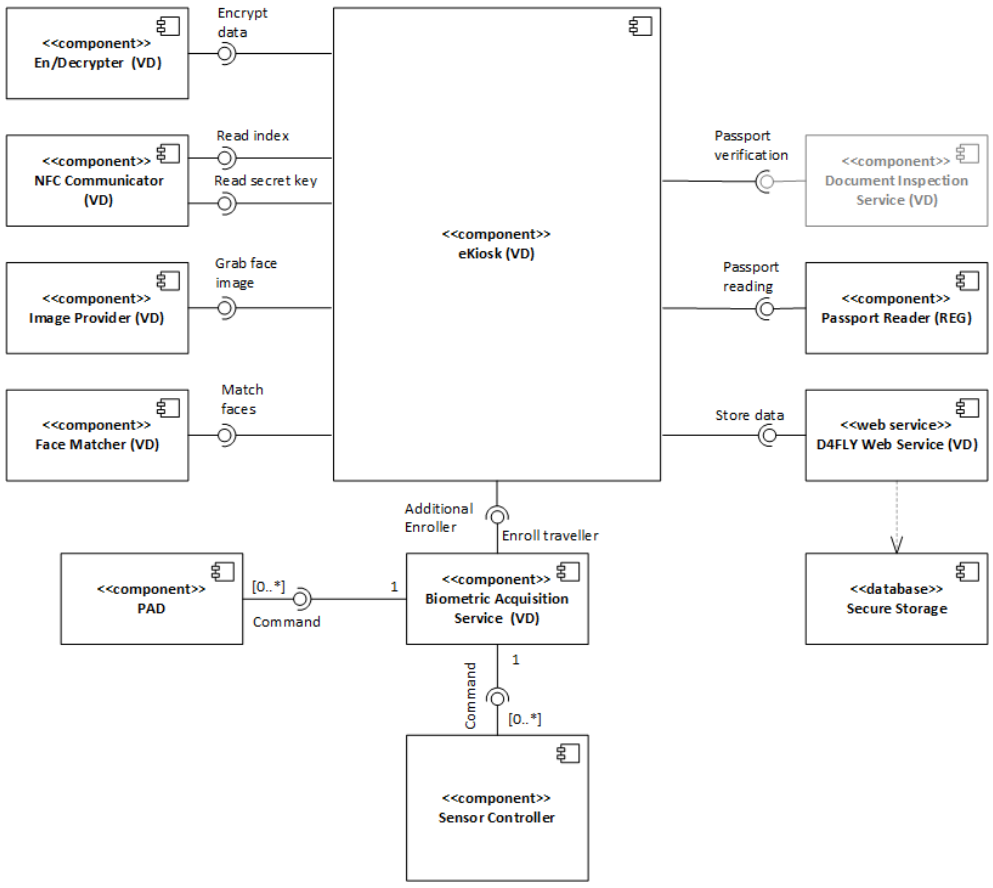

FIGURE 5-2: ENROLMENT COMPONENT DIAGRAM

The enrolled data is handled by the D4FLY Web Service core component, which is implemented for the D4FLY prototype and demonstrator using a cloud computing platform. Its purpose is to store and link encrypted enrolled data with unique identifier (index), which is provided by authorised enrolment kiosk. Conversely, it allows only authorised clients to upload and download enrolled data. The connection over all communication channels between the web service and its clients is secured.

5.2 Verification process in Scenario 2

5.2.1 Work cycle of the verification process in Scenario 2

Preconditions:

- Traveller is enrolled with the Enrolment Kiosk
- Biometric corridor is in operation and ready to accept new traveller

Process steps:

1. Traveller starts the D4FLY mobile application
2. Traveller moves to front of the corridor
3. Traveller taps the smartphone on the NFC reader
4. SYSTEM – Receives unique ID and decryption key from the smartphone
5. SYSTEM – Downloads travellers encrypted enrolled data
6. SYSTEM – Decrypts the data
7. SYSTEM – Distributes enrolled data to respective biometric sensor components
8. SYSTEM – Opens the entry gate
9. Traveller enters corridor and walks through the corridor
10. SYSTEM – Closes the entry gate once traveller passes it
11. SYSTEM – Detects that the traveller and entered the biometric corridor
12. SYSTEM – Instructs SC components to start with verification procedure
13. SYSTEM – Detects that traveller approaching exit gate of the biometric corridor
14. SYSTEM - Instructs SC components to stop verification procedures
15. SYSTEM – Receives the fusion result from the Biometric Fusion Module
16. SYSTEM – Sends final result to connected BGD
17. SYSTEM – Opens the exit gate
18. Traveller leaves the biometric verification corridor
19. SYSTEM – Deletes all personal data which was temporarily stored in the corridor system
20. SYSTEM – Closes exit gate
21. SYSTEM – Ready for next traveller

5.2.2 System configuration of the verification process in Scenario 2

The general system architecture solution for the verification procedure in Scenario 2 is summarised in Figure 5-3. The core of the system is formed by the BCSt, with various supporting components attached to it. These are En/Decryptor, Indexer, Gate Controller, Person Tracker and Secure Key Provider. Biometric Acquisition Service acts as a master component with SC slaves connected to it.

The traveller enrolment details are read from the traveller's smartphone via the NFC Communicator component. The traveller's enrolment data are acquired from the D4FLY

database, based on the traveller index provided and sent to the BCSt component. The pre-enrolled data is decrypted (En/Decryption component) and distributed via Biometric Acquisition Service to the various biometric sensor controller components involved in the verification. SCs and their respective PADs then pass the verification results to the biometric fusion component for biometric fusion processing, which in turn provides the final verification outcome back to the BCSt.

The system configuration including the components and their connections is depicted in the component diagram in Figure 5-4.

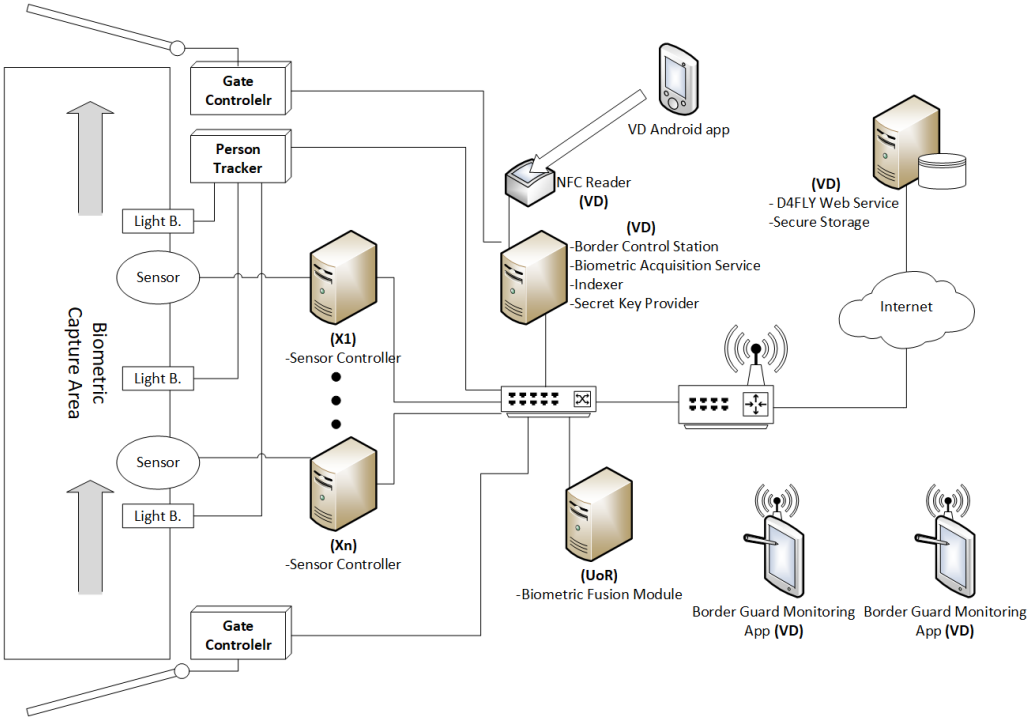

FIGURE 5-3: SYSTEM CONFIGURATION FOR VERIFICATION IN SCENARIO 2

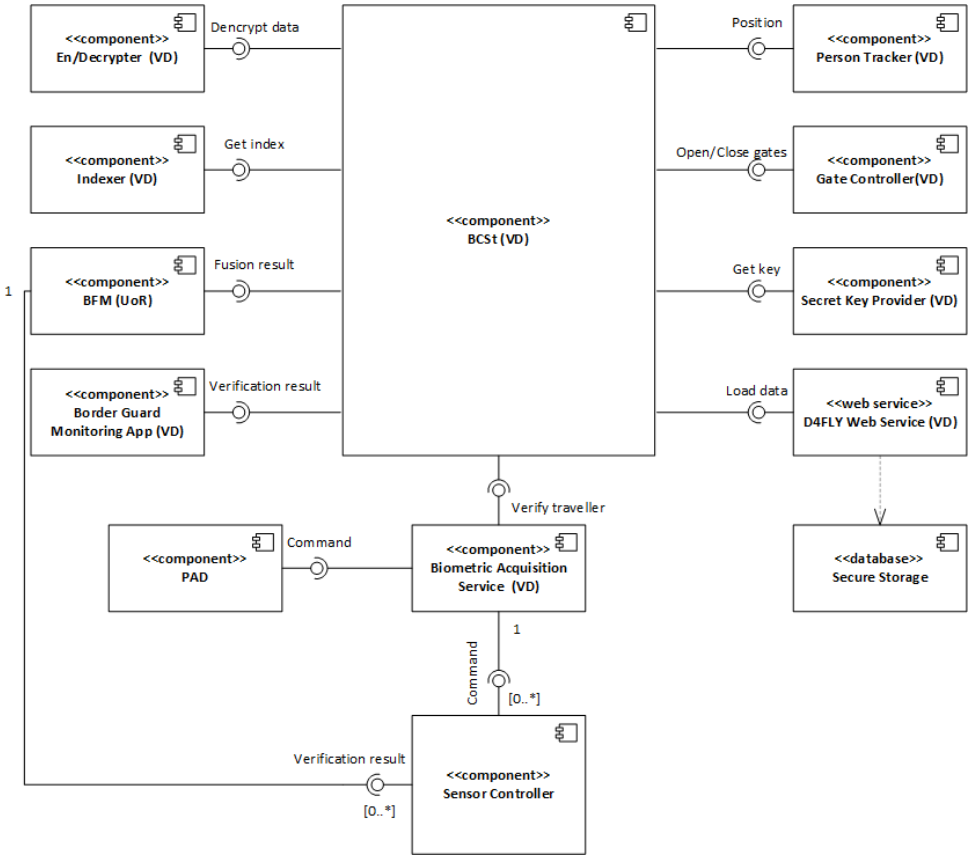

FIGURE 5-4: VERIFICATION COMPONENT DIAGRAM

6 CONFIGURATION 3: LAND BORDER SCENARIO

Configuration 3 in the scope of the D4FLY solution (Figure 2-1) proposes the system configuration for Scenario 3 defined in the Grant agreement. The third scenario focusses on advanced imposter fraud countermeasures and intends to support the first line manual border check in the land border scenario. Enhanced document checks are applied and in addition BG gets additional support for the face verification of the traveller.

6.1 Work cycle of the verification process in Scenario 3

1. BG requests the passport from the traveller
2. Traveller gives his/her passport to BG
3. BG places the passport holder page on the passport reader
4. SYSTEM – Scans holder page in visible, infra-red and ultra-violet light spectrum, extracts MRZ and sends the scanned data to respective passport checker components for verification
5. SYSTEM – Reads the digital content and sends the data to respective passport checker
6. SYSTEM – Displays all check results for passport holder page
7. SYSTEM – Displays passport face image. Special facial features in the picture are marked and point to regions of interest for closer visual inspection
8. BG decides whether or not the traveller can cross the border or additional inspection shall be performed

6.2 System configuration of the verification process in Scenario 3

The system configuration for this scenario is based on Configuration 1 (see Section 4) with reduced number of components (Figure 6-1). Similarly to Scenario 1, the system places BG workstation with Enhanced Document Inspector core component at the centre of its architecture. However, in contrast to Configuration 1, the attached Document Inspection Service requires connection solely to FaceMorph Checker and connections to all other server components are optional. TravelPattern component is removed altogether.

The execution architecture of the system showing the components and their connections is depicted in the component diagram in Figure 6-2.

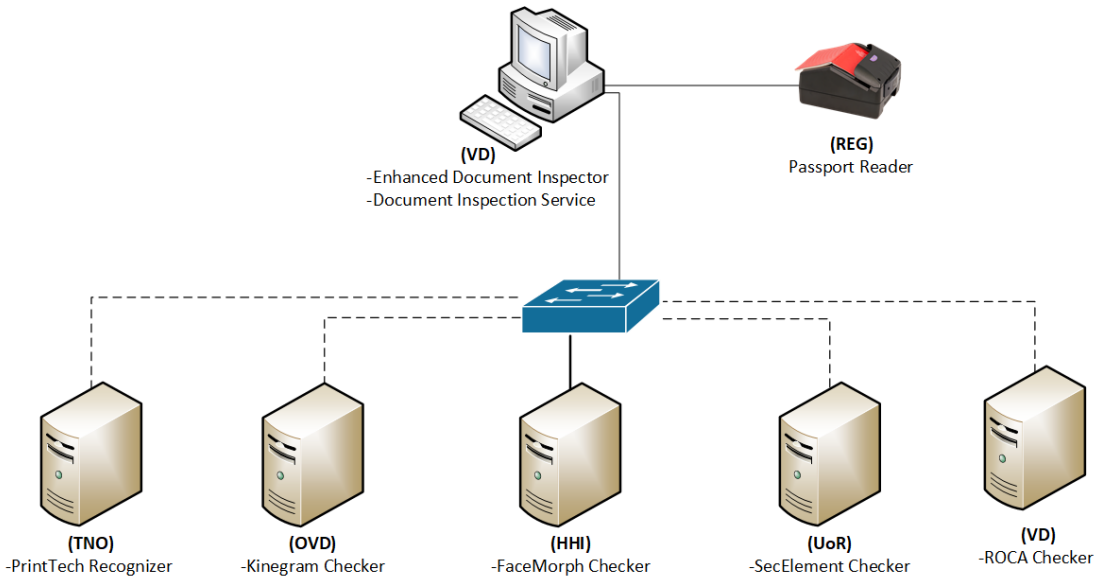

FIGURE 6-1: SYSTEM CONFIGURATION FOR SCENARIO 3

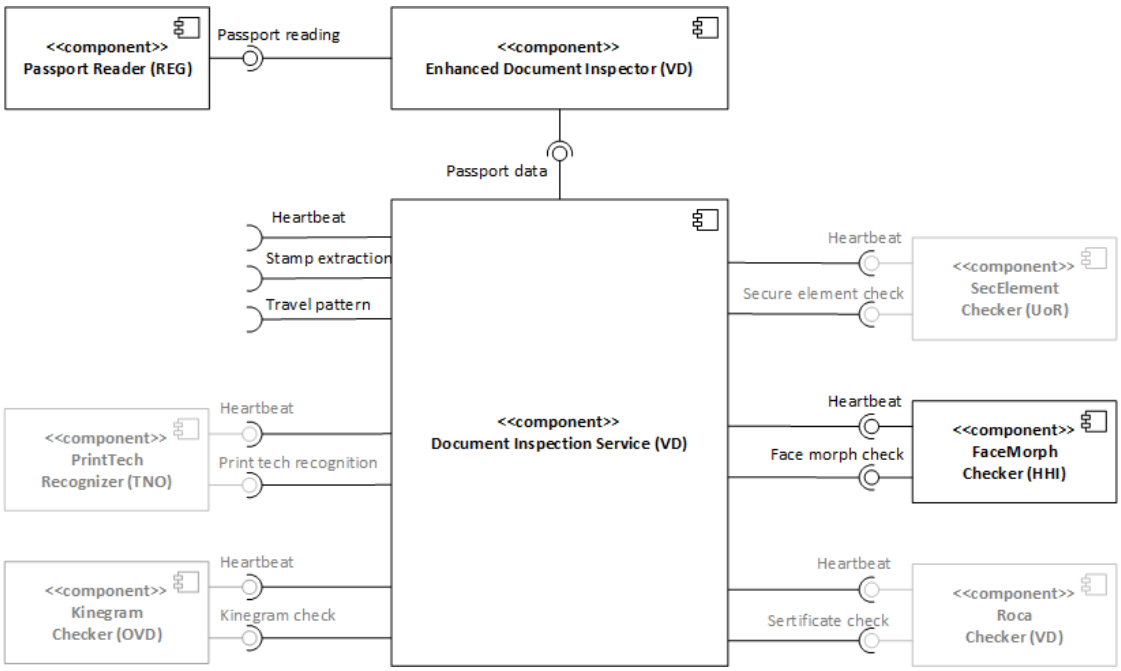

FIGURE 6-2: COMPONENT DIAGRAM FOR SCENARIO 3

The passport verification procedure has two main steps:

1. Checking the visual passport content
2. Checking the digital passport content

For each step, the Document Inspection Service component distributes the data obtained from the Enhanced Document Inspector to the connected checker components.

In the first phase the Document Inspection Service obtains the scan of the holder page captured in visible, infra-red and ultraviolet light and distributes it to three components; PrintTech Recognizer component for printing technique recognition, Kinegram Checker component for kinegram checking, and SecElement Checker for secure element checking.

In the second phase the Document Inspection Service passes face chip image to FaceMorph Checker component to check the face image for face morphing effects. Additionally, the component enriches the travellers passport image with information, which is displayed on the screen to assist in the visual inspection process. During this phase the Document Inspection Service also sends chip certificates to the ROCA Checker component to check for any abnormalities.

The system architecture allows for easy modification of configurations. In general, it is possible to connect all document verification checker modules connected in this configuration. The configuration as shown in Figure 6-2 is one possible configuration, which might be used for the field test, however, the selection of checker components, which are used for specific field tests can deviate from the configuration that is shown here.

7 CONFIGURATION 4: COACH SCENARIO

Configuration 4 in the scope of the D4FLY solution (Figure 2-1) proposes the system configuration for Scenario 4 defined in the Grant agreement. Scenario 4 addresses the checking of the travellers by the border guard in confined spaces (e.g. in coaches) crossing the border using a smartphone application. An enrolment step is used to acquire the data that is necessary for the later verification, also enabling pre-checking of the data. Main goal of this scenario is to perform risk assessment of passengers based on their travel documents prior to reaching the border control, as well as to check the coach travellers without them having to leave the coach, thus accelerating the border crossing process.

The overall process is divided into two distinct phases as experienced by the traveller: pre-boarding enrolment and verification using the BG smartphone application. The enrolment takes place before the traveller boards the coach. After the data capturing is completed, the data is encrypted and sent to the D4FLY database. This dataset is then downloaded from the database using the web service by the BG smartphone application, decrypted and used by the BG to verify travellers inside the coach at the border crossing.

The emphasis in this scenario is to:

- Check travellers and their passports before boarding a coach
- Speed up the border checking process especially for low risk travel groups

7.1 Enrolment process in Scenario 4

7.1.1 Work cycle of the enrolment process in Scenario 4

The detailed use cases related to the different scenarios are described in T2.1 in WP2 and can be found in D2.1 due in M15. For better understanding of the described system configuration for this scenario, the main working cycle of the enrolment in Scenario 4 are described here. The enrolment steps are very similar to the ones of Scenario 2:

1. Traveller approaches the enrolment kiosk
2. SYSTEM – Displays welcome screen and selection of supported languages
3. Traveller selects the display language
4. SYSTEM – Displays the consent form in selected language
5. Traveller reads the consent form and accepts it by pressing the accept button

[Two alternatives of the next steps for prototype system are considered]

6. A. SYSTEM – Instructs the traveller to place his/her journey ticket on the ticket reader
7. A. Traveller scans the ticket

or

6. B. SYSTEM – Instructs the traveller to select correct journey from a list
7. B. Traveller selects the correct coach journey from a list

8. SYSTEM – Extracts the unique journey number from the ticket and uses it as index
9. SYSTEM – Instructs traveller to place his/her passport on the passport reader
10. Traveller puts the passport on the passport reader
11. SYSTEM – Scans the passport holder page and reads the content of the chip
12. SYSTEM – Instructs the traveller to remove the passport from the passport reader

13. Traveller removes the passport
14. SYSTEM – Instructs the traveller to look into the face capture camera
15. Traveller looks into the camera
16. SYSTEM – Matches the captured face image with the passport image
17. SYSTEM – Encrypts the enrolled data
18. SYSTEM – Sends the encrypted data and index key to D4FLY web service to be stored in the database
19. SYSTEM – Instructs the traveller that enrolment procedure is successfully completed
20. Traveller leaves the kiosk

7.1.2 System configuration of the enrolment process in Scenario 4

The configuration of the system for Scenario 4 is summarised in Figure 7-1. The core of the system is formed by eKiosk component with various supporting components attached. Notable supporting components are the Image Provider and Passport Reader that are accompanied by the Face Camera and Regula Reader peripherals respectively. Optionally Document Inspection Service client component may be included in this system configuration. The traveller indexing is provided by the Ticket Service Agent, which acquires the travellers journey number from journey list or QR scanner. In the final stage of the enrolment process eKiosk sends the encrypted enrolment data to the D4Fly database for secure storage using the D4FLY Web Service.

The architecture and configuration of the system including the components and their connections is depicted in the component diagram (Figure 7-2).

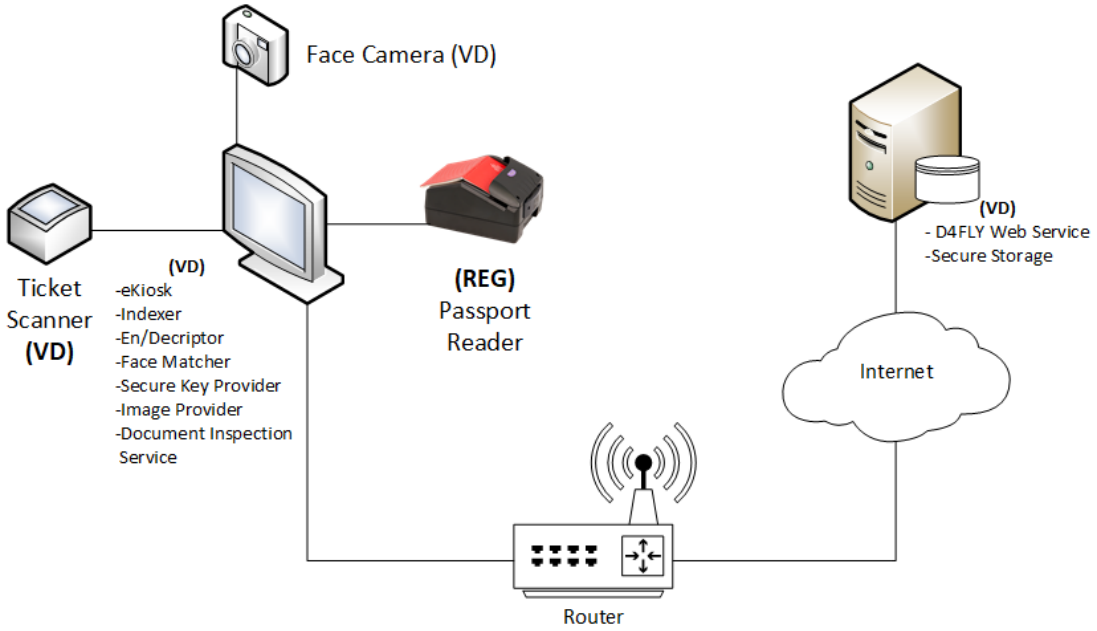

FIGURE 7-1: SYSTEM CONFIGURATION OF ENROLMENT FOR SCENARIO 4

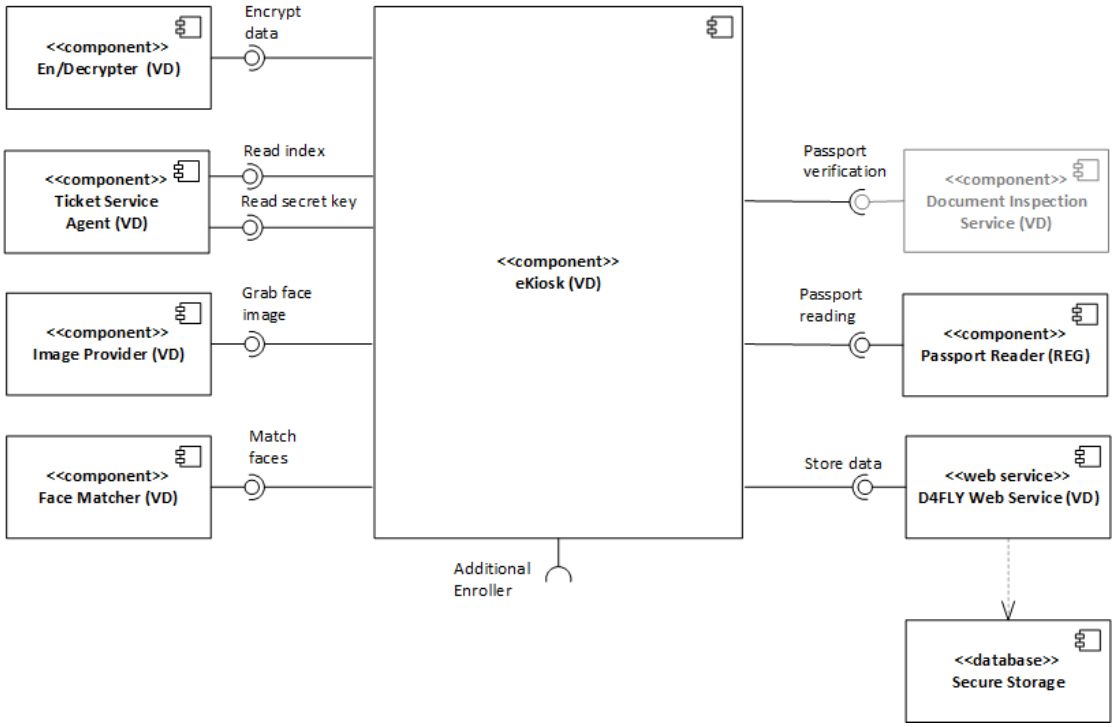

FIGURE 7-2: CONFIGURATION INCLUDING COMPONENTS FOR SCENARIO 4

The enrolled data is handled by D4Fly Web Service core component, which is implemented using a cloud computing platform for the prototype and demonstrator of the system. Its purpose is to store and link encrypted enrolled data with unique identifier (journey number), which is provided by authorised enrolment kiosk in a database. Conversely, it allows only authorised clients to upload and download enrolled data. The connection over all communication channels between the web service and its clients will be secured.

7.2 Verification process in Scenario 4

7.2.1 Work cycle of the verification process in Scenario 4

Preconditions:

- All travellers in the coach are enrolled
- BG is authorised to use Border Guard Verification App on the smartphone
- Enrolment data for all travellers are downloaded and stored on the smartphone

Process steps:

1) BG enters the coach
2) foreach (traveller in the coach)
   a) BG instructs the traveller to look into the smartphone camera
   b) Traveller looks into the camera
   c) BG captures facial image of the travellers face
   d) SYSTEM – extracts face biometric template and matches it in the downloaded dataset of enrolled data
   e) SYSTEM – Displays the verification results

3) BG leaves the coach

7.2.2 System configuration of the verification process in Scenario 4

The general system architecture solution for the verification procedure in Scenario 2 is summarised in Figure 7-3. The system consists simply of a Border Guard Verification App on a portable device used by BG (e.g. smartphone). The app connects remotely to the D4FLY Web Service to load all necessary verification data.

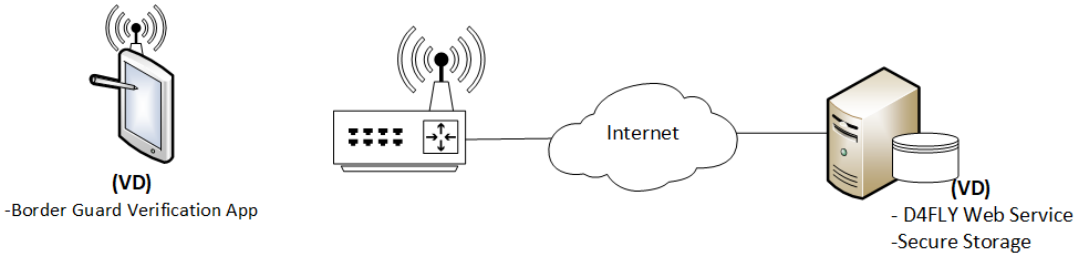

FIGURE 7-3: SYSTEM CONFIGURATION OF VERIFICATION FOR SCENARIO 4

8 SUMMARY

In this report, the system architecture of the D4FLY system is described, explaining the main design decisions on partitioning, modular approach and communication. The configurations for the four scenarios as they are planned at the time of writing this deliverable are presented with the focus on the implementation of the prototype and demonstrator system.

As it has to be expected that changes might be necessary based on the analysis of the final requirements, experiences during the further development, as well as the feedback from the field tests. These changes and the final status will be reported in D4.6.

The next steps include

- Analysis of the requirements (T2.2) and if necessary, adaptation of the system architecture and configurations
- Implementation of the described architecture and components
- Internal and external testing (field test for prototypes for each scenario) and incorporation of results, changes or improvements based on the test results

REFERENCES

[1] http://projectprotect.eu/
[2] http://zguide.zeromq.org/
[3] https://developers.google.com/protocol-buffers/
[4] https://en.wikipedia.org/wiki/Protocol_Buffers/

ANNEX A: BASIC COMMUNICATION PROTOCOLS

Annex A describes basic communication protocols between the client and master hubs and remote components connected to them using sequence diagrams.

A.1 Heartbeat protocol

During normal operation the client and master hubs will listen for periodic messages, "heartbeats" coming from the connected local server and slave components respectively. The "heartbeat" signals of each verification component are sent periodically using *publish-subscribe* pattern and contain the component's status information (see Figure A-1). Hub will contact the core component only if there is a state change in remote component.

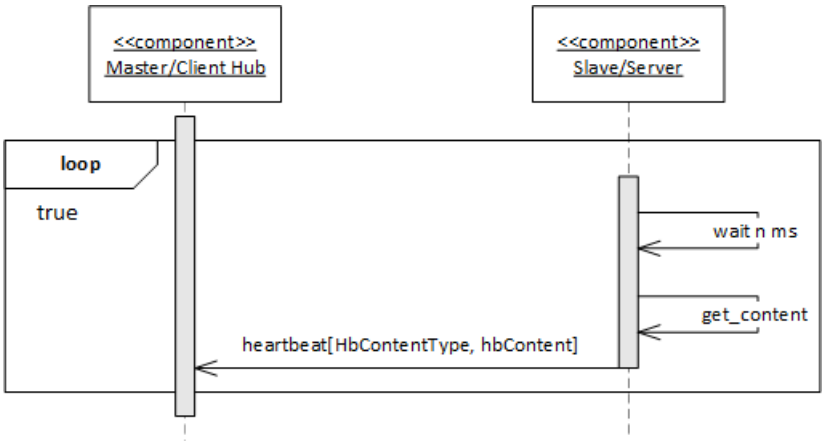

FIGURE A-1: HEARTBEAT PROTOCOL

A.2 Client-Server basic communication protocol

In normal operation the client hub will contact the remote server component by sending requests, containing request type and a request message, and after that it will wait for reply. On another side the remote server component will listen for client requests. Every server component has to serve its clients in reasonable time slot, predefined by the BG (see Figure A-2). The same protocol is also used by core components when contacting D4FLY Web Service.

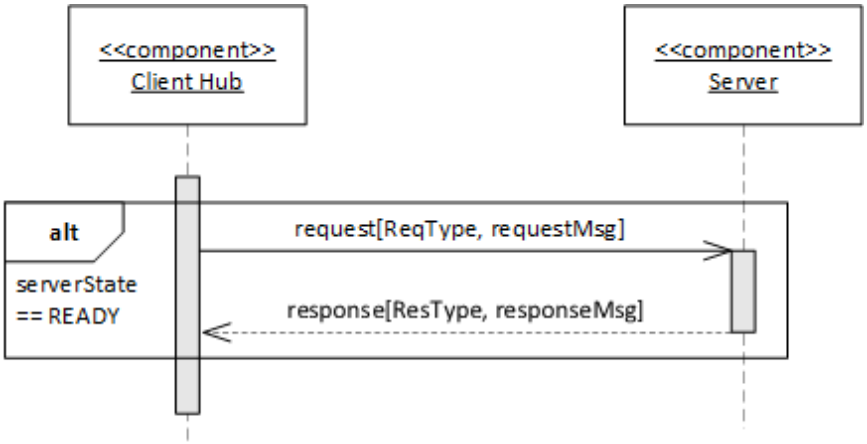

FIGURE A-2: CLIENT-SERVER BASIC REQUEST PROTOCOL

A.3 Master-Slave basic communication protocols

In normal operation the master hub component will contact the remote slave component by sending a commands or requests containing command/request type and a command/request message. Command are sent asynchronously, meaning that the master component will not wait for any response back from the slave components (see Figure A-3).

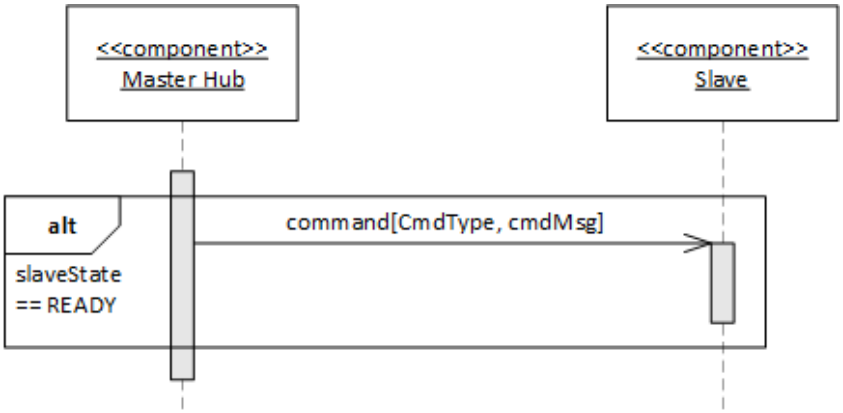

FIGURE A-3: MASTER-SLAVE BASIC COMMAND PROTOCOL

When requesting master hub component will use the request protocol shown in Figure A-4.

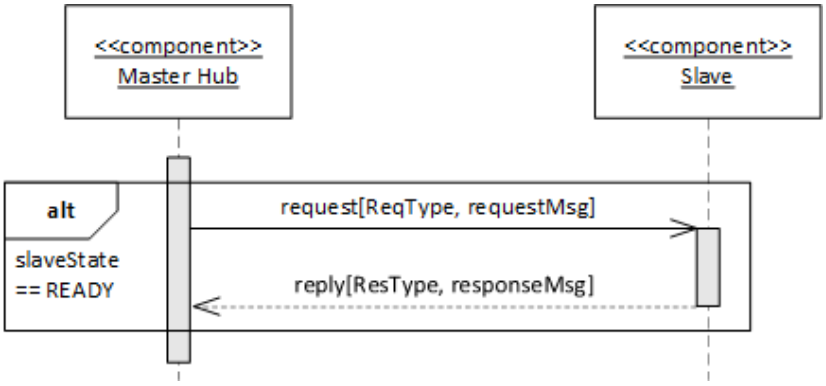

FIGURE A-4: MASTER-SLAVE BASIC REQUEST PROTOCOL