TRUSTED DIGITAL IDENTITY

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

COMPUTER SCIENCE AND ENGINEERING/INFORMATION TECHNOLOGY

By

Shivam Tyagi (161345)

UNDER THE SUPERVISION OF

Mr. Gaurav Singh

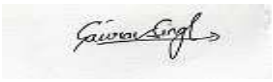
to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan – 173234, Himachal Pradesh.

CERTIFICATE

This is to certify that the work titled "TDI – Trusted Digital Identity" submitted by "Shivam Tyagi" of B. Tech of Jaypee University of Information Technology University, Solan, Himachal Pradesh has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of any other degree or diploma.

Signature of Supervisor	
Name of Supervisor	Mr. Gaurav Singh
Designation	Senior Project Lead (TDI)
Date	25-May-2020

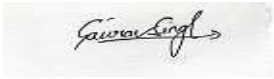
DECLARATION

I hereby declare that the work presented in this report entitled **"TDI – Trusted Digital Identity"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from Feb 2020 to May 2020 under the supervision of **Mr. Gaurav Singh** (TDI Team Lead, Thales).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Shivam Tyagi, 161345.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.



(Supervisor Signature)
Mr. Gaurav Singh
TDI Team Lead
Thales

ACKNOWLEDGEMENT

I would like to place on record my deep sense of gratitude to Mr. Gaurav Singh, Senior Project Lead (TDI), Thales for his generous guidance, help and useful suggestions.

I also wish to extend my thanks to my friends and colleagues for their insightful comments and constructive suggestions to improve the quality of this project work.

Signatures of Student:

Shivam Tyagi

(161345)

TABLE OF CONTENT

LIST OF ABBREVIATION

Acronyms	Full Form
TDI	Trusted Digital Identity
MNO	Mobile Network Operator
JSON	JavaScript Object Notation
JWT	JSON Web Token
BPMN	Business Process Model and Notation
PPI	Pixels Per Inch
OEM	Original Equipment Manufacturer
XML	Extensible markup language
API	Application Programming Interface
URL	Uniform Resource Locator
MRZ	Machine Readable Zone
SDK	Software Development Kit

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In this world where everything become digital we face lots of cyber threats. So to building up a believed situation is pivotal to adapt to the present cybersecurity threats. The TDI encourages you to create identities, ensure advanced information moves and oversee accreditations put away in a cryptographic device, while meeting the most elevated models with regards to security, robustness and quality.

A trusted digital identities comprises of a lot of checked attributes (like government documents or biometrics), along these lines giving a genuine connection between an individual and their digital identity. These traits may likewise incorporate confirmation with third parties, for example,
Government databases, credit card number, social identity or mobile records.

For MNOs, OEMs and different ventures, trusted digital identities mean smoother advanced work processes, quicker client obtaining forms and reliable client information. Working expenses are diminished and the client experience improved. In addition, a TDI can fill in as a passage for endorsers of access different security-sensitive services, for example, mobile money, eGov and internet banking, permitting MNOs to take a lead in these quickly developing divisions.

This is the place Trusted Digital Identity Services Platform becomes an integral factor, a one-stop administrations stage to digitalize portable supporter enlistment.

CHAPTER - 1

INTRODUCTION

1.1 General Introduction

A TDI is made once the data provided by the end-user has been verified or checked for authenticity. It arrange everything/ customize the software according to the required by MNO's and OEM's to digitalize client enlistment, together with the capture, verification and certify the biometric and client credentials. Drawing on Thales' in-depth expertise and experience in these fields, similarly as complementary services from trustworthy partners worldwide, the platform allows efficient enlistment, each offline and over the internet.

As a result OEMs, MNO's and different organization will expedite their medical aid ways, launch latest products, particular services, fight fraud and meet rules.

A TDI is made by following 3 required steps: capture, verify, digitalize there identity. The main points in every step could differ in line with the expanse of the data that a particular MNO's needs and must be capture and therefore the rules they're subject to, as an example, about personal information security.

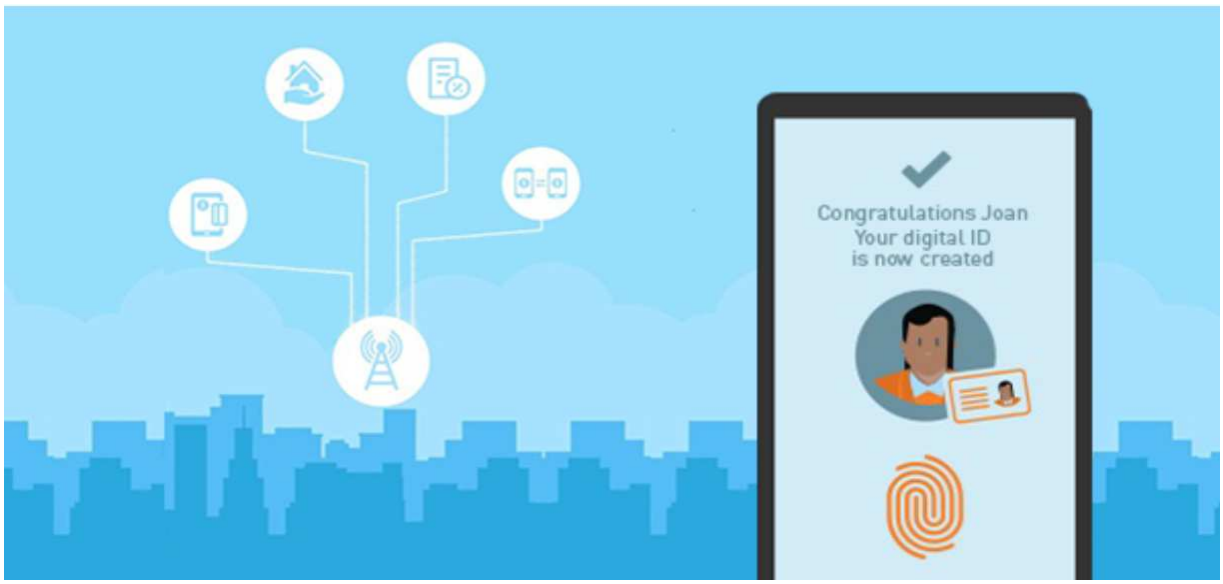


Fig 1: Trusted Digital Identity

1.2 Relevant current/open problems

The objective of TDI guidelines is to prevent MNO's from being used, intentionally or unintentionally, by criminal elements for customer data breach. Related procedures also enable MNO's to better understand their customers and there identity. This helps the customer manage their threat in a proper manner. Today not only the MNO's but also different online businesses can implement TDI. TDI approaches have been growing for quite a while and they have become significant gloablly. With issues relating to defilement, information penetrates, untrusted personalities TDI approaches have now advanced into a significant apparatus to battle unlawful confirmation in the OEM's and Telecom field. TDI permits organizations to secure themselves by guaranteeing that they are working together lawfully and with genuine elements, and it additionally ensures the people who may somehow or another be hurt by untrusted characters.

Numerous Telecoms and OEM's beginnings their TDI strategies by just gathering essential information and data about their clients, in a perfect world utilizing electronic personality confirmation. Information incorporate any administration ID (Like identification, driving permit, and so on.), biometric.

When this fundamental information is gathered, MNO's by and large contrast it with arrangements of people that are known for defilement, on a rundown of approvals, associated with being engaged with a wrongdoing, or at a high danger of untrusted people groups.

From that point, the MNO's at that point measures the amount of a hazard their customer seems, by all accounts, to be and that they are so prone to get associated with criminal activity. When this estimation has been made, the MNO's distinguishes the unapproved individual and report them right away.

1.3 Problem Statement

The whole process to generate the TDI can be a very inefficient and hectic. Searching and traveling to the verification centres, different one each time. So, instead of the people went to the concerned verification centre, it is easier for only their IDs and biometrics to go. And that generates the demands for TDI web portals.

So we provide our solution to the product based companies customised web portals which receives ID and other relevant data (like selfie, fingerprint, etc.) from the customer and verify those data to generate the digital identity.

This makes the TDI process much flexible and easier for both the customer and the end users.

CHAPTER-2

LITERATURE SURVEY

Our main concern revolves around building the required system. Now here are some of the different definitions that different books and different websites provides us.

2.1 JSON

JSON is an open source file format, what's more, an arrangement of information exchange that utilizes comprehensible content to store and transmit information objects comprising of characteristic worth combines and cluster information structure (or the other serializable worth). It's a typical information record position, with a different fluctuate of utilizations, such as filling in as a substitution for XML in frameworks of AJAX.

JSON could be a language-autonomous data. It had been gotten from JavaScript, anyway a few elegant programming dialects epitomize code to concoct and examine JSON-design information. The official web media kind for JSON is application/json. JSON filenames utilize the augmentation .json

2.2 JWT

JSON Web Token is a web standard for making data with elective signature and/or elective encoding whose payload holds JSON that asserts some variety of claims. The tokens square measure signed either employing a non-public secret or a public/private key. For instance, a server must create a jwt token that has the claim "logged in as admin" and supply that to a consumer. The consumer might use the token so that the token used to prove that it's logged in as administrator. The tokens may be inscribed by only one party's non-public key (usually the server's) in order that party will afterwards verify the token is legitimate. On the off chance that the contrary party, by some suitable and reliable implies that, is in control of the comparing open key, they too square measure prepared to check the token's authenticity. The tokens square measure intended to be reduced, URL-safe, and usable especially in an exceedingly internet browser single-sign-on (SSO) setting. JWT cases will as

a rule be wont to pass personality of validated clients between Associate in Nursing character provider and an assistance provider, or the other sort of cases PRN by business forms.

2.3 Web Service

- A web service is any piece of software package that creates itself on the market over the net and uses an even XML electronic communication system. XML is employed to cypher all communications to an online service. As an example, a shopper invokes an online service by causing associate degree XML message, then waits for a corresponding XML response. As all communication is in XML, internet services aren't tied to anybody software or programming language—Java will speak with Perl; Windows applications will speak with UNIX operating system applications.
- Web services are self-contained, modular, distributed, dynamic applications that may be represented, published, located, or invoked over the network to form merchandise, processes, and provide chains. These applications will be native, distributed, or web-based. Web services are designed on prime of open standards like TCP/IP, HTTP, Java, HTML, and XML.
- Web services are XML-based info exchange systems that use the net for direct application-to-application interaction. These systems will embody programs, objects, messages, or documents.
- A web service may be an assortment of open protocols and standards used for exchanging knowledge between applications or systems. Software package applications written in varied programming languages and running on varied platforms will use internet services to exchange knowledge over pc networks just like the net during a manner like inter-process communication on one pc. This ability (e.g., between Java and Python, or Windows and Linux applications) is thanks to the utilization of open standards.

2.4 BPMN

An ordinary Business Process Model and Notation can give organizations with the bent of understanding their inside business methodology in an exceedingly graphical documentation

and can give associations the adaptability to talk these strategies in an exceedingly standard way. Additionally, the graphical documentation can encourage the comprehension of the presentation joint efforts and business exchanges between the associations. This ensure organizations will see themselves and members in their business and can change associations to manage to new inward and B2B business conditions rapidly.

2.5 Flowable

Flowable is a very powerful and light-weight business method engine written in Java. The Flowable method engine permits you to deploy BPMN 2.0 method definitions (an trade XML customary for outlining processes), making method instances of these method definitions, running queries, accessing active or historical method instances and connected knowledge, and rather more. This section can step by step introduce numerous ideas and API's to try to that through examples that you just will follow on your own development machine.

Flowable is very versatile once it involves adding it to your application/services/architecture. You'll be able to insert the engine in your application or service by together with the Flowable library that is on the market as a JAR. Since it's a JAR, you'll be able to add it simply to any Java environment: Java SE; servlet containers, like house cat or mole, spring; Java applied science servers, like JBoss or WebSphere, and so on. Instead, you'll be able to use the Flowable REST API to speak over hypertext transfer protocol. There are many Flowable Applications (Flowable creator, Flowable Admin, Flowable IDM and Flowable Task) that provide out-of-the-box example UIs for operating with processes and tasks.

2.6 Docker

Docker is a platform which provide developers to develop, package and run application as a lightweight, versatile, independent compartment, which can run for all intents and purposes anyplace. A container is a virtual environment on top of the OS kernel to capture all of its software - libraries,

Dependencies, etc.

A Docker container is an isolated environment running within a host machine's kernel that

allows us to run application-specific code. Once the application and all its dependencies are packed into a Docker container we are able to run it in any environment. The process of one container cannot affects the process of other containers.

2.7 Jenkins

Jenkins is an open source mechanization server. It adjusts the components of programming bundle advancement related with building, testing, and sending, encouraging nonstop coordination and ceaseless conveyance. It's a server-based framework that runs in servlet compartments like Apache Tomcat. It underpins variant administration apparatuses, along with AccuRev, CVS, Subversion, Git, Mercurial, Perforce, ClearCase and RTC, and may execute Apache Ant, Apache Maven and sbt based for the most part comes likewise as discretional shell contents and Windows clump orders.

Constructs will be activated by various recommends that, for instance by submit during a form framework, by programming by means of a cron-like instrument and by mentioning a chose assemble uniform asset locator. It might likewise be activated once the contrary forms inside the line have finished. Jenkins common sense will be reached out with modules.

2.8 MySQL

Structure Query Language (SQL) is a query language for databases used for storing and managing data RDBMS. SQL was the primary business language introduced for E.F Codd's relative model of data. these days the majority RDBMS(MySql, Oracle, Infomix, Sybase, MS Access) use SQL as a base query language for databases. SQL is employed to perform every kind of data operations and manipulation in RDBMS.

In our product we basically use MySQL to store and manipulate the data according to our requirement. IT provide easy handling of data by providing the simple workbench and command line interface to develop and handle the data. We store person ID’s and biometrics for a specified person in our database.

2.9 Cucumber

Cucumber could be a computer code tool that supports behavior-driven development (BDD. Integral to the Cucumber BDD approach is its ordinary language PC program alluded to as

Gherkin. It licenses expected PC code practices to be per an intelligent language that clients will see. Accordingly, Cucumber allows the execution of highlight documentation written in business-confronting content. It's generally utilized for testing distinctive PC code. It runs machine-driven acknowledgment tests written in an exceedingly conduct driven turn of events (BDD) vogue.

Cucumber was initially composed inside the Ruby programing language. Furthermore, was initially utilized exclusively for Ruby testing as a supplement to the RSpec BDD system. Cucumber as of now bolsters a spread numerous} programming dialects through different usage, along with Java and JavaScript. The open gracefully port of Cucumber in .Net is named Spec Flow. For instance, Cuke4php and Cuke4Lua zone unit PC code connects that change testing of PHP and Lua comes, severally. Various executions may just use the Gherkin PC program though actualizing the rest of the testing structure inside the objective language.

2.10 Selenium

Selenium may be a moveable framework for testing web applications. So gives a playback instrument to writing commonsense investigate some time not the prerequisite to discover a test scripting language (Selenium IDE). It moreover gives an investigate area explicit language (Selenese) to expressly state tests in an exceedingly scope of in style programming dialects, just as C#, Groovy, Java, Perl, PHP, Python, Ruby and Scala. The tests will at that point run against most recent web programs. Se runs on Windows, Linux, and macOS. it's ASCII content document PC code released beneath the Apache License 2.0

2.11 Gradle

Gradle is AN ASCII content record manufacture automation framework that expands upon the thoughts of Apache Ant and Apache Maven and presents a Groovy-based space explicit language instead of the XML kind used by whiz for announcing the undertaking setup. Gradle utilizes a guided non-cyclic chart to work out the request inside which errands will be run.

Gradle was intended for multi-venture manufactures, which may develop to be very monstrous. It underpins dynamic forms by demonstrating knowledge concluding that components of the develop tree are to date; any errand subordinate exclusively on those components doesn't need

to be constrained to be re-executed

2.12 Process to create Trusted Digital Identity

A TDI is generated by following three major steps:

1. Capture
2. Verify
3. Digitalize

The main points of every step could vary in line with the extent of the data the MNO needs to capture and therefore the rules they're subject to, as an example, around personal data privacy.

2.12.1 Capture ID documents and Biometrics

Capture ID documents

The subscriber's ID information is caught from partner degree character report (identification, national ID, driving permit or inhabitant grants and so on.). During this strategy, information like name and birthdate is separated through picture investigation (optical character acknowledgment). This innovation helps ensure that right and expand customer information is entered inside the CRM.

Fig 2: Capture ID

Capture Biometrics

A biometric catch devices (for example, camera of cell phones, web camera, stand or tablet or particular unique mark scanner) is utilized to catch the client's biometric data. Biometrics which will be gathered grasp data from the face or unique mark.

Fig 3: Capture Biometric

2.12.2 Verify ID documents and biometrics

Verify ID documents

After capturing of the required data, the solution verify the genuineness of an ID archive with determined programming. Various techniques can be utilized to check the security highlights of the gave character record against an ID database. In this progression, the individual data of the cardholder can likewise be recovered and naturally fill the fields in enlistment shapes, the CRM, and so forth. This outcomes in a more straightforward and quicker onboarding process for clients, just as more prominent information security for MNOs and time investment funds.

There are totally various degrees of record confirmation: typical, upheld ID picture from actinic radiation, progressed with safety efforts existing in white, infra-red and bright and chip-based with electronic archive check.

Fig 4: Verify ID document

Verify Biometric

Biometric technology is employed to verify if the person document which was presented by the person is who so ever he/she claim to be. It conjointly presents a chance for later use as an easy and modern way to access services that need identification — and above all once verification has to be conducted remotely.

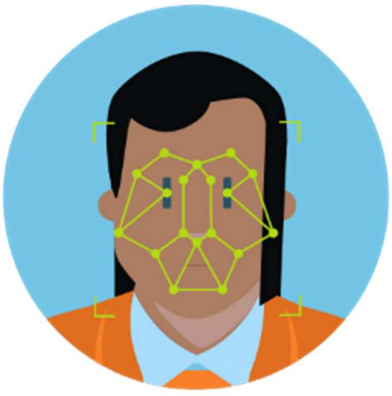


Fig 5: Verify biometrics

2.12.3 Digitalize ID created

After a match is identified between the biometric data and which is presented on the identity

document, a digital ID could also be created. So by using the digital ID the user is able to login t the service and able to use them. The access will be granted by merely presenting the requested biometric attribute, like their face data and fingerprint. This operate permits users to seamlessly get access to the MNO's services or third-party services.

Fig 6: Digital Id created

CHAPTER-3

ANALYSIS, DESIGN AND MODELING

TDI is currently working on the deployment of a new mobility solution named digital ID. We use the three phase done on a pure ID Verification solution (ID documents and Biometric Verification), TDI highlighted a lack of fraud detection when the ID card picture is modified. Most of the time the solution is not able to detect such modification which cause a problem for their Cybersecurity committee.
Even if this weakness is common to any level1 based solution, we proposed a new solution to reinforce the fraud detection for the build phase to come, based on a combination of electronic document reading with an All About Me service.

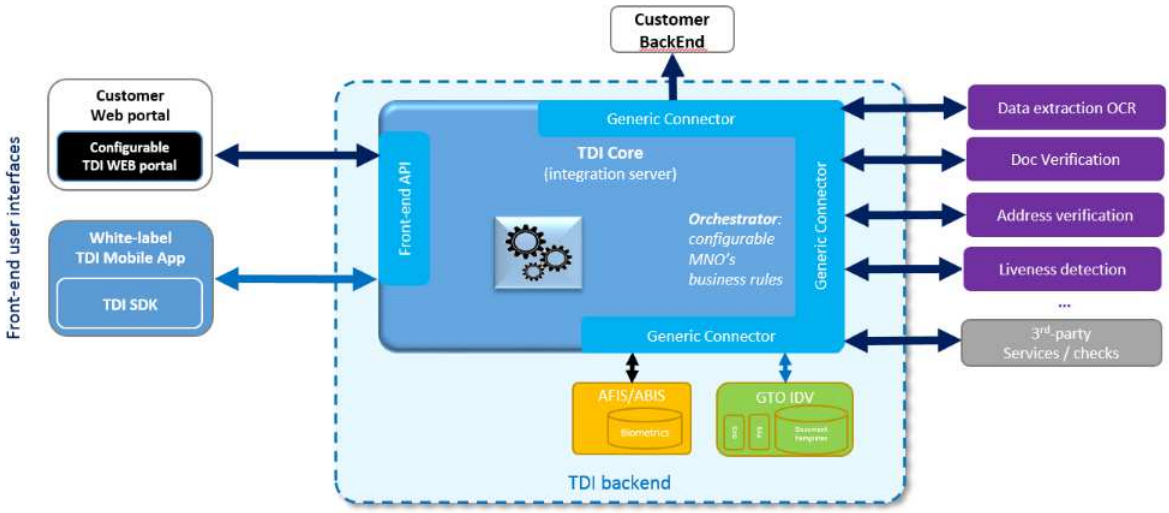


Fig 7: High level TDI architecture

Considering that fraud detection we provide the solutions, Thales proposed to TDI the following scenarios:

- Ask the end user to upload or take picture of ID document (like passport, driving license, etc.)
- Ask the end user to take the selfie.
- Detect the liveness of the picture using third party API's

- Information is sent to the TDI core.
- The Verification of provided data is instantiated.
- Perform an ID verification on the document and the end user selfie.
- After Verification of the data the form fields are populated automatically by retrieving the information from the presented documents.
- The Digital Identity is created by the solution.
- Results of the verification are sent back to TDI back end server.

3.1 Technical Requirement

S3

To transfer your data (photos, documents, recordings and so forth.) to Amazon S3, you have to starting produce Associate in making S3 container in one in everything about AWS Regions. You'll at that point move any scope of articles to the basin.

In execution terms, containers and articles are assets, and Amazon S3 can gives API's to you to deal with those assets. For instance, you'll produce a can and move objects exploitation the Amazon S3 API. You'll furthermore utilize the Amazon S3 reassure to play out these tasks. The reassure utilizes the Amazon S3 API's to send solicitations to Amazon S3.

This segment discloses the best approach to work with cans. For information concerning working with objects, see working with Amazon S3 Objects.

An Amazon S3 pail name is all inclusive unmistakable, and furthermore the namespace is shared by all AWS accounts. This suggests once a pail is framed, the name of that basin can't be utilized by another AWS account in any AWS Region till the can is erased. You should not depend on explicit can naming shows for accommodation or security confirmation capacities. For can naming pointers, see Bucket Restrictions and Limitations.

Amazon S3 makes basins during a Region you indicate. To enhance idleness, limit costs, or address regulative necessities, select any AWS Region that is geologically near the very edge of you. For instance, in the event that you live in Europe, you may understand it beneficial to

frame basins inside the Europe (Ireland) or Europe (Frankfurt) Regions

We in our product use Amazon S3 bucket use to provision manually the TDI configuration files.

ECR

Amazon Elastic Container Registry (ECR) is managed fully by Docker container instrumentality written record that creates it simple for engineers to store, oversee, and send Docker compartment pictures. Amazon ECR is incorporated with Amazon Elastic Container Service (ECS), streamlining your improvement to creation headway. Amazon ECR takes out the need to work your own instrumentality vaults or stress with respect to scaling the hidden foundation. Amazon ECR has your photos during an incredibly available and climbable structure, allowing you to steadfastly convey compartments for your applications. Mix with AWS Identity and Access Management (IAM) gives asset level administration of each storehouse. With Amazon ECR, there are no immediate expenses or duties. You pay only for the number information on information on information} you store in your archives and information moved to the net.

In our product we use ECR for integrator to provision (by using Jenkins jobs) the Docker image.

ECS

Amazon Elastic Container Service (Amazon ECS) could be a totally managed container orchestration service.

ECS could be a nice option to run compartments for some reasons. To start with, you'll favor to run your ECS bunches abuse AWS Fargate that is server less figure for holders. Fargate evacuates the need to arrangement and oversee servers, empowers you to determine and purchase assets per application, and improves security through application disengagement deliberately. Second, ECS is utilized broadly among Amazon to control administrations like Amazon SageMaker, AWS Batch, Amazon Lex, and Amazon.com's suggestion motor, ensuring ECS is tried widely for security, reliableness, and handiness.

Furthermore, because of ECS has been a fundamental column for key Amazon administrations, it will locally incorporate with elective administrations like Amazon Route fifty three, Secrets Manager, AWS Identity and Access Management (IAM), and Amazon Cloud Watch giving

you an all-around perceived skill to send and scale your compartments. ECS is furthermore ready to rapidly coordinate with elective AWS administrations to carry new abilities to ECS. For example, ECS allows your applications the flexibleness to utilize a blend of Amazon EC2 and AWS Fargate with Spot and On-Demand rating decisions. ECS conjointly coordinates with AWS App Mesh that could be an assistance work, to bring well off recognizability, traffic controls and security highlights to your applications. ECS has full grown rapidly since dispatch and is by and by propelling 5X extra holders every hour than EC2 dispatches examples.
We use ECS in our product to deploy/ apply (make it live on production) all the new configuration files and new Docker that has already provision on AWS. We do this by simply restart the ECS tasks.

3.2 Basic workflow of solution

3.2.1 Registration of Customer

Before any authentication/authorization can happen, the Customer/Tenant has to be registered in the TDI Authorization Server.

During this step, TDI shall at least:

- Create a customer Id which is a unique string representing the customer registration. This information is not a secret and can be disclosed.
- Create a password for that customer. This is an information that the Client shall keep secret. As the password is not intended to be used by a human it can have a significant length.
- Allocate a client_id to the Customer. The value may be the same as customer Id. The Customer may be allocated several client_id.
- Set the type of Authorization Grant mode "**Authorization code**" flow

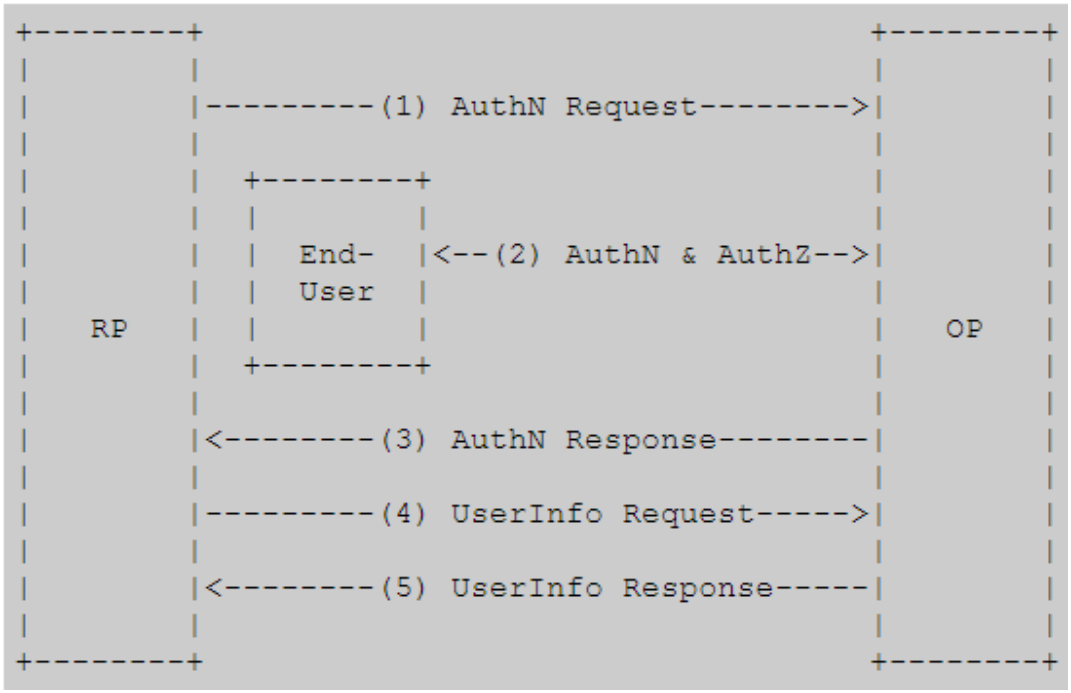


Fig 8: Authorization Code Sequence

RP- Client and OP- OpenID Provider

The Authorization Code Flow follow these required steps:

1. Client creates an Authentication Request which include the desired request parameters.
2. Client submit the request to the Authorization Server.
3. Authorization Server authenticates the identity of End-User.
4. Authorization Server receives End-User Consent/Authorization.
5. Authorization Server submit the End-User back to the Client with an Authorization Code.
6. Client requests a response to the Token Endpoint by using the Authorization Code.
7. Client gets a response that have an ID Token and Access Token in the response body.
8. Client verify the ID token and retrieves the End-User's Subject Identifier.

- Set a lifetime for the granted Access Token.

This step can be performed by a manual process operated by a Thales operator.

No specific UI is required to allow a self-service process by the Customer.

3.2.2 Registration of End-Users

Before any authentication/authorization can happen, this scheme requires that each End user of the Customer is registered in the TDI Authorization Server.

For Customer convenience and efficiency, TDI SHALL provide a dedicated UI and a RESTful API allowing to register a new End user for the Customer.

The RESTful API is mandated because it could be that a huge number of End users could have to be registered. Hence automation is required.

The UI is recommended for Consumer convenience when End users can be individually added or removed.

This RESTful API/page SHALL allow to, at least:

- Set a user Id value
- Set its email address.
- Set an authorized scope (i.e. the list of services this End user is allowed to use)
- Generate a first password

Others attribute may be added during study phase.

It is acceptable for a first release that the End user registration is performed by a Thales operator by a non-documented means, based on inputs from the customer.

Once the End user is registered, an email SHALL be send automatically to the end user to change its password. The mail contains a navigable link on a dedicated UI for changing the password and more generally is profile information that we will have defined.

3.2.3 Authentication of the End-Users

This is based on "Authorization Code Flow"

Precondition for this step:

- The Customer is registered.
- The customer's End user is registered
- The Customer or web UI is configured with:
    - The client_id allocated during Customer registration
    - The TDI authorization endpoint URL
    - The TDI token endpoint URL

Detailed description of the flow:

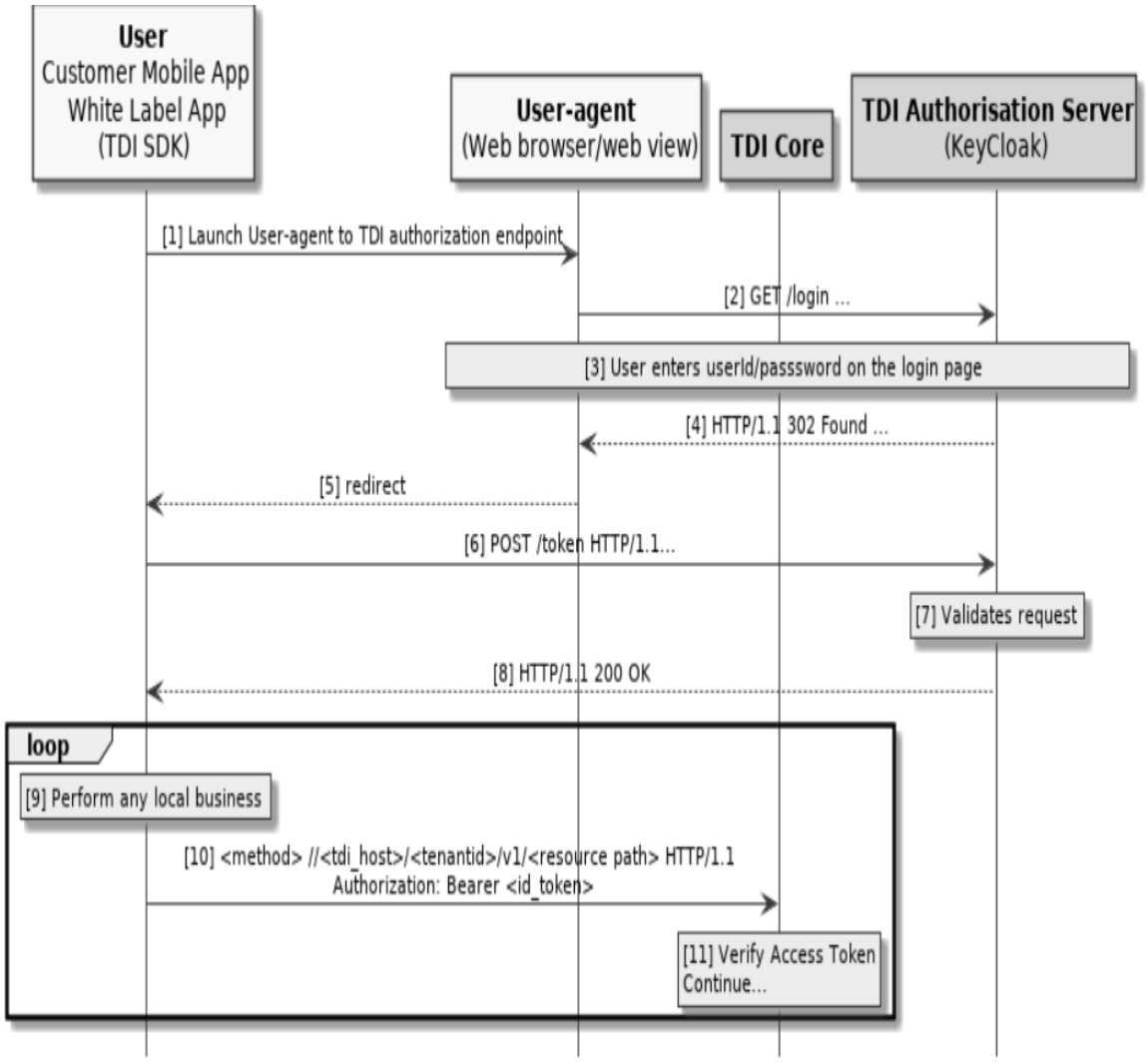


Fig 9: Sequence Diagram of authorization of end user

Step 1: The Customer launch a User-agent (an external Web browser or a web view) with the TDI authorization endpoint URL

Here is an example:

```
HTTP/1.1 302 Found
Location: https://server.example.com/authorize?
  response_type=code
  &scope=openid%20profile%20email
  &client_id=s6BhdRkqt3
  &state=af0ifjsldkj
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
```

- The final host name and login path values will be determined during the study.
- The response_type SHALL be set to "code" to indicate the Authorization Code Flow.
- The scope SHALL be set to "openid". The final scope values will be done during the study.
- The client_id allocated during Customer/Tenant registration.
- The state can be used by the Customer or White Label Mobile App to set any context that will be returned after User-agent redirection.
- The redirect_uri is set with the uri that the User-agent will use to return to the Customer or White Label Mobile App. Assuming that the Customer or White Label Mobile App has previously registered its custom uri on the device platform.

Step 2: The User-agent sends the HTTP request to the TDI Authorization end point. Here under is the HTTP request example corresponding to the previous step

Here is an example:

```
GET /authorize?
  response_type=code
  &scope=openid%20profile%20email
  &client_id=s6BhdRkqt3
  &state=af0ifjsldkj
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb HTTP/1.1
Host: server.example.com
```

Step 3: The User is presented a Login page where it can enter its userId / password as set during the registration. During this step the TDI Authorization Server validates the request, authenticates the End user and scope as defined in OpenID Connect that is:

In the event that the solicitation is legitimate, the Authorization Server makes an endeavor to show the End-User or decides if the End-User is bore witness to, depending upon the solicitation parameter esteems utilized. The techniques utilized by the Authorization Server to show the End-User (for example username and positive recognizable proof, meeting treats, and so on.) square measure on the far side the extent of this particular. An Authentication program is additionally shown by the Authorization Server, depending upon the solicitation parameter esteems utilized and furthermore the validation techniques utilized.

The Authorization Server ought to imagine to show the End-User inside the accompanying cases:

The End-User isn't as of now bore witness to.

The Authentication Request contains the brief parameter with the value login. during this case, the Authorization Server ought to reauthenticate the End-User despite the fact that the End-User is as of now verified.

The Authorization Server ought NOT act with the End-User inside the accompanying case:

The Authentication Request contains the brief parameter with the value none. during this case, the Authorization Server should come a slip if AN End-User isn't now bore witness to or couldn't be silently authenticated.

When interacting with the End-User, the Authorization Server should use acceptable measures against Cross-Site Request Forgery

Step 4: After the successful authentication, the authorization code is returned to the User-agent as query parameters to the redirect_uri, as illustrated here under:

```
HTTP/1.1 302 Found
Location: https://client.example.org/cb?
  code=SplxlOBeZQQYbYS6WxSbIA
  &state=af0ifjsldkj
```

Step 5: The User-agent redirects to the Customer or White Label Mobile App with the code (authorization code) and the state. The Customer or White Label Mobile App SHALL validate the returned state.

Step 6: Having the authorization code, the Customer or White Label Mobile App asks for an ID Token (also called access token). This can be achieved as specified in OpenID Connect.

```
POST /token HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW

grant_type=authorization_code&code=SplxlOBeZQQYbYS6WxSbIA
  &redirect_uri=https%3A%2F%2Fclient.example.org%2Fcb
```

The grant_type SHALL be set to "authorization_code", with the code , client_id and redirect_uri attributes. The code contains the the authorization code as received in previous step.

The client_id and redirect_uri repeats the values used during previous step. The client_id attribute SHALL be present as the Customer or White Label Mobile App as the Client is not authenticated (not a confidential Client).

Step 7: The TDI Authorization Server validates the token request.

Step 8: After the successful validation of the token request, the TDI Authorization Server replies as illustrated here under:

For Example:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store
Pragma: no-cache

{
 "access_token": "SlAV32hkKG",
 "token_type": "Bearer",
 "refresh_token": "8xLOxBtZp8",
 "expires_in": 3600,
 "id_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWdkazcifQ.ewogImlzc
   yI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4Mjg5
   NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiAibi0wUzZ
   fV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDEzMTEyODA5Nz
   AKfQ.ggW8hZ1EuVLuxNuuIJKX_V8a_OMXzR0EHR9R6jgdqrOOF4daGU96Sr_P6q
   Jp6IcmD3HP99Obi1PRs-cwh3LO-p146waJ8IhehcwL7F09JdijmBqkvPeB2T9CJ
   NqeGpe-gccMg4vfKjkM8FcGvnzZUN4_KSP0aAp1tOJ1zZwgjxqGByKHiOtX7Tpd
   QyHE5lcMiKPXfEIQILVq0pc_E2DzL7emopWoaoZTF_m0_N0YzFC6g6EJbOEoRoS
   K5hoDalrcvRYLSrQAZZKflyuVCyixEoV9GfNQC3_osjzw2PAithfubEEBLuVVk4
   XUVrWOLrLl0nx7RkKU8NXNHq-rvKMzqg"
}
```

Step 9: The user performs any local business (e.g. capture of the ID, of the faces) depending on the Customer or White Label Mobile App. At the end of this step, the description assumes that there is the need to connect the TDI Core (e.g. to submit ID pictures for verification). This is represented in the next steps.

Step 10: The Customer or White Label Mobile App sends the HTTP request, comprising the required <method> (e.g. GET, POST), and the Authorization. This is illustrated here under:

POST //<tdi_host>/<tenantid>/v1/<resource path> HTTP/1.1

Authorization: Bearer <id_token>

…..

The Authorization SHALL contain the Bearer key word and the id_token as provided in the previous step

Step 11: The TDI Core verifies the token value. And so on...

3.2.4 Documents Verification Result

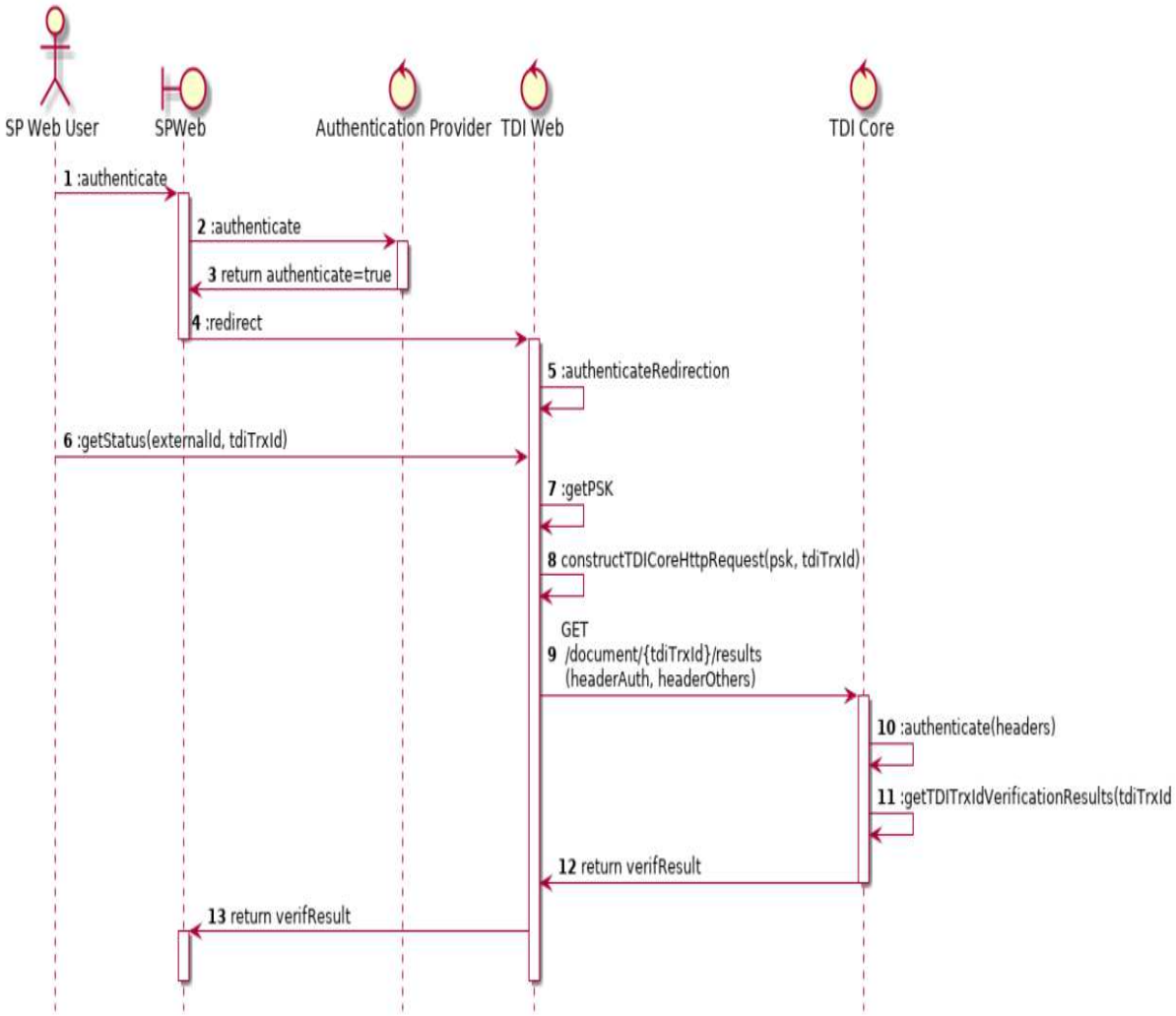


Fig 10: Sequence diagram of documents verification

GET verification result API:

GET /abcdef/0.9/document/TDI-999/results

::Headers::
^Authentication: hmac abcdef:123456:b450db328d7d36f2f4a8de2d97867c3817f08b14a684b573504040c64b905dfc
^Date: Sun, 06 Nov 1994 08:49:37 GMT
^Accept: application/json
^Accept-Languange: en-US
^X-TDI-User-Agent: device_os_name="Android", device_os_version="4.1.1", device_brand="Samsung", device_model="Galaxy Nexus", client_type="sdk", client_version="0.9", client_cert_hash="8964255d2087a747598c048256101c8cc746243c", referral_id="com.gemalto.tdi.sample", referral_version="1.0"

::Body::

Http Response Status

HTTP Status Code	**Indication**
200 OK	Response to a successful GET.
404 Not Found	When a non-existent resource is requested.
410 Unauthorized	When no or invalid authentication details are provided.
500 Internal Server Error	A generic error message, given when an unexpected condition was encountered and no more specific code is suitable.

503 Service Unavailable	Service unavailable. Typically the server is down.

Table 1: Response Status table

ID Verification

- **ID document supported format**

  Security Documents are supposed to in this format:

I. Passports ID size (MRTD´s, electronic MRTD´s);
II. ID Cards ID-2 and ID-1 size (MRTD´s, electronic MRTD´s);

- **ID Verification Service - Verification Level**

The ID Verification process is configured for identity verification, i.e. based on image of ID using visible light range

Categories to Verify

I. Data Integrity
II. Data Format
III. List of Security Features for the supported documents

- **ID Document aspect**

The ID Verification process must support Security Documents that are:

1) Not damaged and clean by in-appropriate use or tear or wear.
2) Fully visible: no object, finger, frame or cover shall be wear on the document during the biometric.
3) If the submitted document does not support with the above, the ID Verification process may still be executed but:
4) The document may not be recognized,
5) Failure in data integrity,
6) Improper OCR extraction,
7) Chances of functional error.
8) The performance and correctness of the ID Verification process may not comply with the proper performance.

- **Two-sided record utilizing a similar goals**

The ID Verification administration should bolster two-sided archives gave front and closing pages are transmitted together inside a solitary solicitation and with same DPI definition

- **Image position upheld for JPEG (counting JPG2000) and PNG demand**

Reports pictures should show 1 archive side for each page.

Different sides archive confirmation will be submitted utilizing two distinctive picture for each side. Different sides of the report submitted on a solitary page will be dismissed Means that divert must be designed along these lines.

No different relics or reports will be imagined along with the record pictures

- **Image goals upheld by ID check arrangement**

The ID confirmation administration will bolster pictures transmitted in lossless

configurations PNG/JPEG/JPEG2000 To removewith a base goals of 300 ppi.

- **Image Compression bolstered by ID confirmation arrangement**

The ID Verification administration will bolster packed pictures agreeable with to table 2, 3 and 4 of the ISO 19794-4:2005 (E) and ISO/IEC 15444-1:2004 for JPG2000.

- **ID Verification Service - Image pivot for archive with MRZ**

The ID Verification administration will bolster pictures of archive face with MRZ with any pivot. The administration will pivot the pictures.

- **Verification Service - Image pivot for archive without MRZ**

The ID Verification administration will bolster pictures of record face without MRZ with a most extreme revolution of +/ - 90° from its level position. The administration will pivot the pictures.

- **Image editing bolstered by ID check arrangement**

The ID Verification administration will bolster pictures got with a white edge of any size. The administration will at long last yield the pictures.

Face Verification

- **Face Verification - Image quality:** The Face Verification administration will bolster pictures transmitted in lossless organizations PNG/JPEG2000/TIFF or lossy JPG pressure. The base upheld picture width is 480 pixels and requires at least 90 pixels between the eyes.

The face check administration will bolster picture gained utilizing:

I. The FVS Mobile SDK application and transmitted by this SDK to the FVS server

II. An outside picture catch gadget and submitted to the FVS server Web Service utilizing its committed face confirmation URL and AP.

Face Verification - Face size and position

The face check administration will bolster picture of the individual applying for Identity Verification if:

I. The picture is taken confronting frontward

II. The face is completely obvious (no glasses if not present on ID card picture, hairs or any covering thing)

III. The face involves in any event 40% of the whole picture submitted.

3.3 Security and additional considerations

Client authentication mode

The flows in this page illustrate a Basic authentication as described in RFC 7617. The complete HTTP request prototype is defined on the RFC 7617. Other authentication schemes may be used. This will be determined during the study.

Access Token Lifetime

Authorization token, ID Token and Access Token SHALL have a short validity period.

Lifetimes SHALL be configurable differently per Customer/Tenant.

In case the ID Token (or Access token) expires during the session, the Client SHALL offer the End user to reconnect to renew the tokens. This could be achieved transparently if the Authorization token is still valid (to be further studied)

3.4 Implementation Screenshot

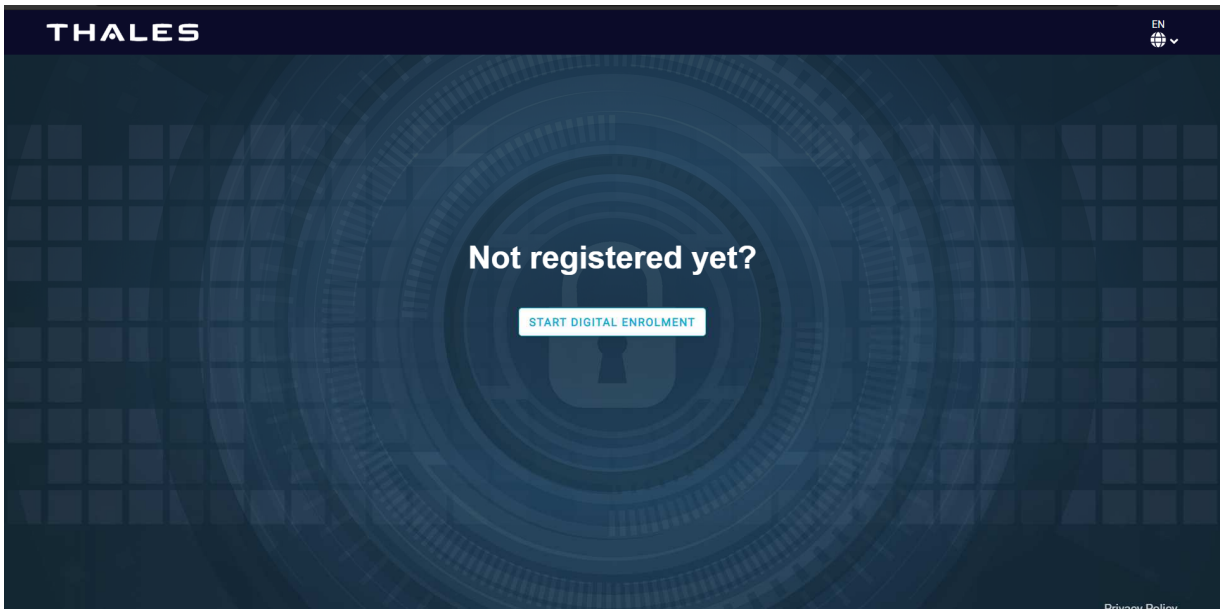


Fig 11: Starting page of web UI

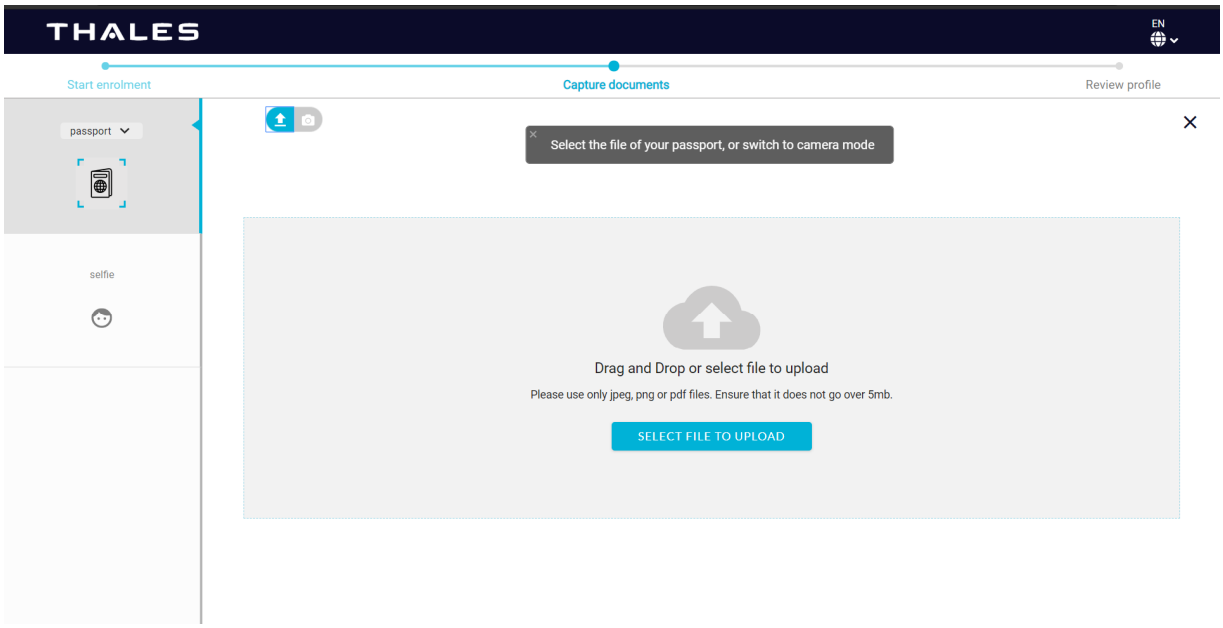


Fig 12: Capture/ Upload ID documents

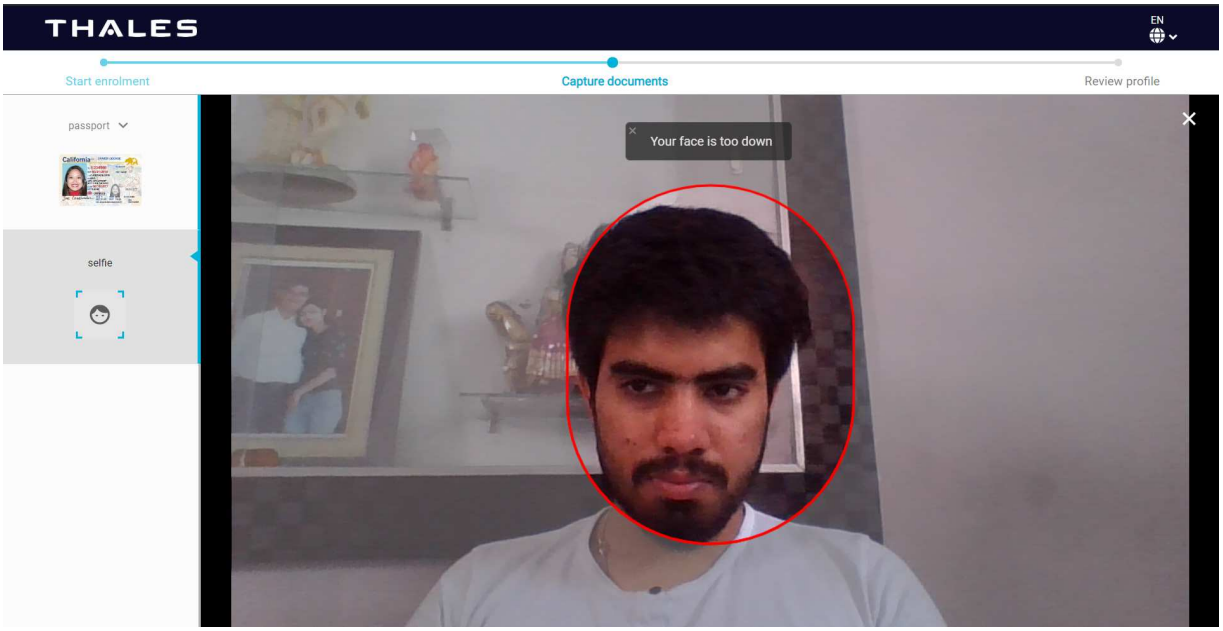


Fig 13: Capture Selfie as biometric

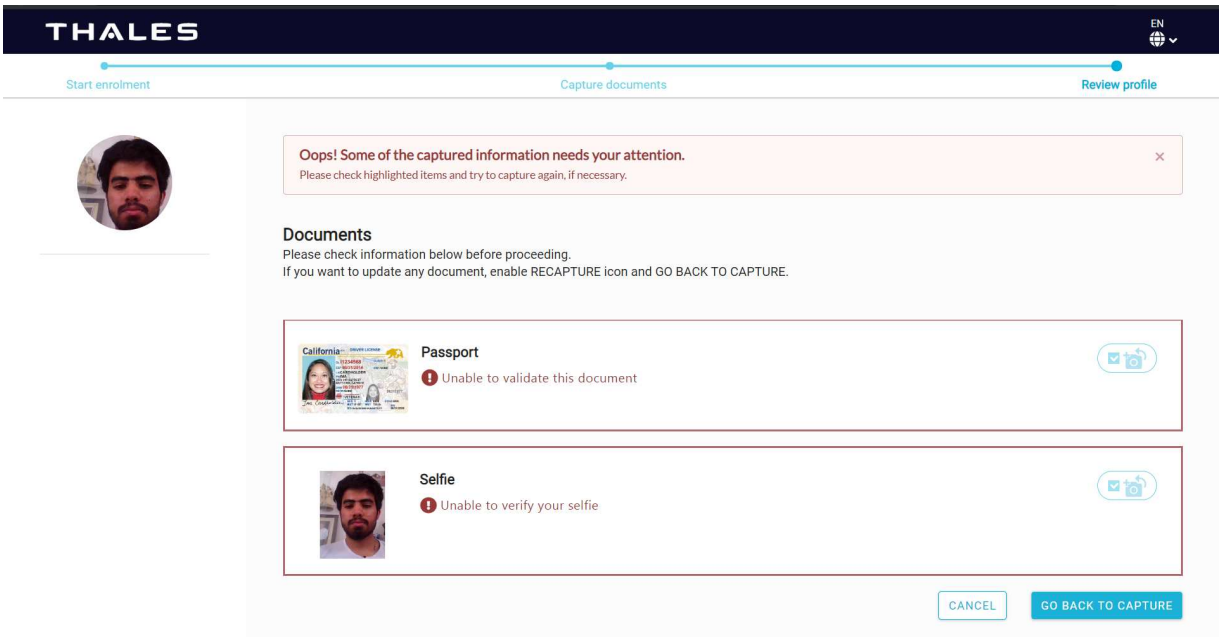


Fig 14: If Documents and Biometrics does not match

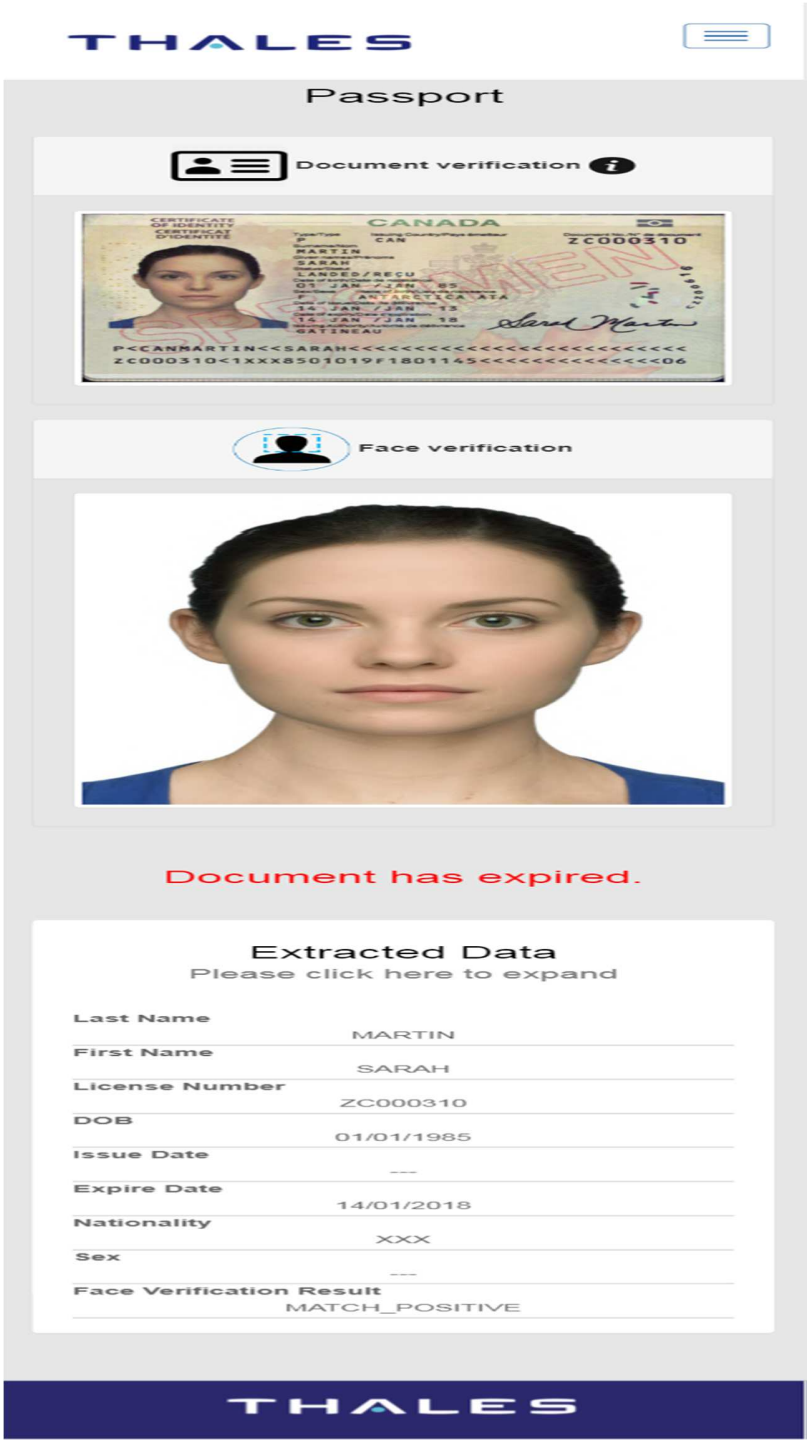


Fig 15: Match Found and Data Extraction

CHAPTER-4

TESTING PLAN

4.1 Testing Plan

To test the web application completely, it is distributed into two parts those are front-end part of the solution and back-end part of the solution. We are distributing this it into two parts not just because they are front-end and back-end but the technology used in both part are different. The back-end part of the application is further divided into 3 parts i.e. persistence layer, service layer and infrastructure layer (REST API). And so testing is carried out in these four major parts.

4.2 Component decomposition and type of testing required

As mentioned above the solution provided is distributed into two parts i.e. front-end part and back-end part. Back-end part of the application is further decomposed into 3 parts i.e. persistence layer, service layer and infrastructure layer. First let's consider the back-end of the solution. The service layer, persistence layer and infrastructure layer the testing is carried out by using unit testing and integration testing. The unit testing will be done using JUnit testing framework and PMD. All the classes undergoes in JUNIT testing. POSTMAN framework is used for integration testing. The back-end of the application is tested using the POSTMAN when the front-end of the application is in under progress. Using the angular built-in feature naming angular cli by providing the "ng test" in the terminal we test the front-end of the application. Hence the front-end of the application is tested.

List all test cases in prescribed format

Purpose	To check navigation bar buttons are working accordingly or not.
Input	Click on any of the button on navigation bar
Expected Output	The buttons should be redirected to its respective page
Actual Output	The button redirects to its respective page
Result	Pass

Table 2: Test Case of Navigation bar (Home, About us, Login)

User Login Module

- Test cases for login page

1) Admin login

Purpose	Enter admin id and opening admin component
Input	Enter valid username and password
Excepted Output	Opening of admin component
Actual Output	Opening of admin component
Result	Pass

Table 3: Test case for admin login page (username)

Purpose	To check validation on password Field
Input	Don’t enter any password
Expected Output	Error message should be there saying "Enter password"
Actual Output	Shows "Enter password"
Result	Pass

Table 4: Test Case for admin login page (password)

Purpose	User can reset password with help of Secret Key
Input	Enter secret key
Expected Output	Setting new password
Actual Output	Password is changed successfully

Result	Pass

Table 5: Test case for admin forgot password page

2) Customer login:

Purpose	Enter customer id (7 digit) and opening admin component
Input	Enter 7 digit valid id
Expected Output	Opening of employee component
Actual Output	Opening of employee component
Result	Pass

Table 6: Test case for customer login page (username)

Password:

Purpose	To check validation on password field
Input	Don't enter any password
Expected Output	Error message should be there saying "Enter Password"
Actual Output	Shows "Enter Password"
Result	Pass

Table 7: Test case for customer login page (password)

Purpose	User can reset password with help of secret key
Input	Enter secret key
Expected Output	Setting new password
Actual Output	Password is changed successfully
Result	Pass

Table 8: Test case for customer forgot password page

4.3 Error and Exception Handling

The error and exception are handled and well checked in the proposed solution. All the places where there is a high risk of exception or error occurrence may possible are all surrounded with try-catch blocks to properly handle the exception. All the exception and error case whenever happened, there is a document named properties record that will contain the additionally importance full clarification of the mistake or exemption happened and that clarification would be shown at the hour of structure filling or at some other occasion occurred while utilizing the web application. Aside from this a log document will get created in the administrator's framework, logging all the exemption happened with the name of the client by whom special case is happened and at what time the special case happened the timestamp.

4.4 Limitations of the solution

- The application is constrained to one specific organization, an increasingly nonexclusive form can be made, with various sanctuaries as indicated by it vender's necessity, and the favored layout can be purchased by the merchant.
- The client may ordinarily get record confirmation fizzled, however will never know the genuine explanation behind, and will have just the choice to retry.
- There is no real way to decide whether the items/stores included are right or not. By right methods stores from that zone are included not from outside the given territory.

CHAPTER-5
FINDINGS AND CONCLUSION

Findings

While creating the proposed web application, lots of various difficulties were faced. Image altering and image compression on the UI side before submitting the request to the particular APIs. Handling unauthorized access and Managing request queues. For making the user interface responsive by using the JQuery. The proposed application is built on spring boot, and using REST web services is the biggest task.

Conclusion

The system was finally developed using Hibernate, AngularJS, Spring boot, RESTful API's and to deploy for test run we use Apache Tomcat server. We overcome many different kind of faults which were observed while using S3, ECR and ECS this was definitely came out to be the best in various terms than other e-commerce websites. User interface was developed using basic HTML, CSS and to add responsiveness we also use Bootstrap. We had comprehended the offbeat conduct of Angular completely as at numerous focuses the framework should be simultaneous in nature. The site created was finished effectively in reproduced time period.

Future Work

The proposed web application "TDI Webportal" is finished in its own specific manner yet there is considerably more things left for the future upgrades. To give some examples:

- The database of entire web application can be executed by Thales itself.
- Since the web application is neighborhood to a given region, it could be extended to the a lot bigger region restricted to one specific organization, yet a nonexclusive entryway for all.
- The Front-end of the application is executed utilizing JQuery-JS and it can moved to Angular 7

- The outsider API's must be executed by the Thales itself like.
    - Document verification
    - Liveliness detection
    - Data Extraction OCR
    - Address Verification

REFRENCES

[1] https://en.wikipedia.org/wiki/JSON

[2] https://jwt.io/

[3] http://www.bpmn.org/

[4] https://flowable.com/open-source/docs/

[6] https://en.wikipedia.org/wiki/Jenkins_(software)

[7] Thales Internals

TRUSTED DIGITAL IDENTITY

ORIGINALITY REPORT

19%	15%	2%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.thalesgroup.com Internet Source	3%
2	img.sauf.ca Internet Source	2%
3	Submitted to Zambia Centre for Accountancy Studies Student Paper	2%
4	aws.amazon.com Internet Source	1%
5	javatechnologycenter.com Internet Source	1%
6	docs.aws.amazon.com Internet Source	1%
7	Submitted to Jaypee University of Information Technology Student Paper	1%
8	Submitted to Southern New Hampshire University - Continuing Education Student Paper	1%

Shivam