UNISYS

Biocial

TITLE: Edge-driven Biometrics and Facial Recognition

TEAM DETAILS

S.no	Name	Role	E-Mail ID	Phone. No
1	**Sharan S**	**Leader**	**sharan.indumathi@gmail.com**	**8525044391**
2	Sivaram Murugan	Member	srmsiva1613@gmail.com	7639862661
3	Saiprasad J	Member	saiprasadjjj@gmail.com	7540028149
4	Vijayalakshmi P	Member	vijayalakshmipethuraj@gmail.com	8903364802

MENTOR DETAILS

S.no	Name	Role	E-Mail ID	Phone. No
1	**Dr. R. Suresh**	**Project Guide**	sureshramanujam78@gmail.com	9655228850

NAME OF THE INSTITUTION: Sri Manakula Vinayagar Engineering College, Madagadipet, Puducherry-60510.

UNISYS

CONTENTS

ABSTRACT

We are in a pursuit of automating our day-to-day process and the full potential of biometrics verification process is yet to be utilized as we haven't completely moved to contactless biometrics verification systems. And the recent corona threat has increased the need for a contactless system. This attempt to make use of a contactless biometrics verification system will increase our dependency on the internet. This process can be optimized by reducing the network latency and by removing our dependency on the internet. Using this project, we are trying to achieve an edge-based biometrics computation system that would reduce the network latency and optimize the process.

INTRODUCTION

Using our edge-based biometrics computation system, we are in an attempt to achieve an optimized version of the existing system by intelligently storing the frequently accessed records within the local machine or LAN and by transferring the computation process to the local machine, thus reducing the dependency on the internet for these records. The current system involves the usage of the internet for almost every individual verification process, which introduces network latency. Our solution optimizes the process by reducing the latency using the above said approach.

SOLUTION

Understanding Problem Statement

To develop an edge based biometric system where the edge computation system caches a slice of appropriate data using a priority based heuristic algorithm and reduce the network latency associated with the current biometrics verification system.

Example: A University is trying to install a biometric verification system at the entrance and the exit of all its campuses (A, B and C). Each campus has its repetitive set of students and faculties visiting regularly. Students and Faculties visiting campus A are frequent visitors for that particular campus and visit the other two campuses on an ad hoc basis. Our system caches records of frequent visitors on priority basis, which removes the dependency on the internet to perform biometrics verification for the frequent individuals.

EXISTING SOLUTION

Existing Biometrics Verification System:

"So how does facial recognition work? Technologies vary, but here are the basic steps:

➢ A picture of your face is captured from a photo or video. Your face might appear alone or in a crowd. Your image may show you looking straight ahead or nearly in profile.

➢ Facial recognition software reads the geometry of your face. Key factors include the distance between your eyes and the distance from forehead to chin. The software identifies facial landmarks — one system identifies 68 of them — that are key to distinguishing your face. The result: your facial signature.

➢ Your facial signature — a mathematical formula — is compared to a database of known faces.

➢ A determination is made. Your faceprint may match that of an image in a facial recognition system database "

- **Alison Grace Johansen**
  for *NortonLifeLock*

"A record of a person's unique characteristic is captured and kept in a database. Later on, when identification verification is required, a new record is captured and compared with the previous record in the database. If the data in the new record matches that in the database record, the person's identity is confirmed."

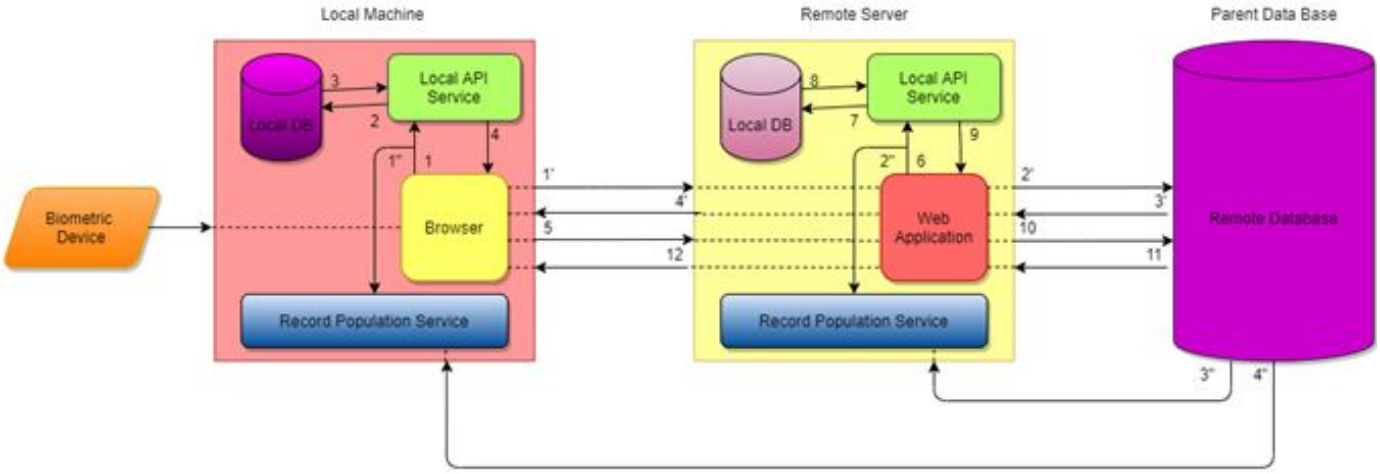
- **Margaret Rouse** (*searchsecurity.techtarget.com*)

Features	ZENTECH	ADONAI	BIOMATIQUES	BIOCIAL
Control over Data and System				
Optimization in Network Latency				
Efficient Computation				

Data Security				
Network Dependency				

OUR SOLUTION

Architecture:



Enrollment process

1' - Enrolment request sent to the Remote server.
2' – Enrolment request sent to the Parent DataBase.
3' – Response sent from the Parent DataBase to the Remote Server.
4' – Response sent from the Remote Server to the Local Machine.

Verification process

1 – Verification Request sent to the Local API Service in the Local Machine.
2 – Request for the verification of data Present in the Local Database of the Local Machine.
3 – Response sent to the Local API Service from the Local Database of the Local Machine.
4 – Response sent to the Browser from the Local API Service.
5 – Verification Request sent to the Web Application from the Local Machine.
6 - Verification Request sent to the Local API Service in the Remote Server.
7 - Request for the verification of data Present in the Local Database of the Remote Server.
8 - Response sent to the Local API Service from the Local Database of the Remote Server.
9 - Response sent to the Web Application from the Local API Service.
10 - Verification Request sent to the Parent Data Base from the Remote Server.
11 - Response sent from the parent DataBase to the Remote Server.
12 - Response sent from the Remote Server to the Local Machine.

Process of RPS

1" – Collection of Requests made by the User.
2" – Collection of Requests made by the User.
3" – Request to populate the local Database with Frequently accessed Record.
4" – Request to populate the local Database with Frequently accessed Record.

Technical Architecture:

When a verification request is made from the user end along with the biometrics captured data. The request is sent to the local API Service which in turn redirects the local DB of the local machine and checks whether the requested record is available there in the local DB. If the Record is present in the local DB, that particular record is returned to the Browser-end. In case of the absence of the requested record, the request is redirected to the remote server. The request is sent to the local API Service from the web application back-end in the remote server which in turn again redirects the request to the local DB of the remote server and checks whether the record is available there in the local DB. If the record is present in the local DB of the remote server, that particular record is returned to the Browser-end. Again, if the requested record is not found, then the request is again redirected to the parent DB from the remote server and the response from the parent DB is returned to the remote server which in turn returns to the Browser. The local databases have the frequently

accessed records, which is populated by the RPS. So, if the local databases have the requested record, it is returned to the Browser-end avoiding further redirection to the remote server or the remote parent DB, thus reducing the network latency.

Record Population Service (RPS): Record population service runs a Machine Learning Algorithm, which uses the metadata of the requests made by the users and populates the local DB with the Frequently accessed records. The local DB in the local machine has the frequently accessed records and the local DB in the remote server has frequently accessed records, but these records are accessed less frequently compared to those in the local machine. So, whenever a request is made, the request is initially sent to the local DB of the local machine. The frequently accessed records are fetched from the collection of records in the local machine. And those which are accessed frequently but not as frequent as those in local machines, are fetched from the remote server. The records are fetched from the parent DB. So, these requests are redirected based on the frequency to either the local host or to the remote end to overcome the network latency associated with the current verification process. The Record Population Service uses an ML algorithm and populates the local DB with the frequently accessed data.

Use Case

Any place where biometric verification is necessary:

- IT sector
- College campuses
- Immigration border security

System Requirements

- **Processor** : Intel Core i5-8365U Processor
- **RAM** : 8GB DDR4 2400 RAM
- **HDD/SSD** : 500GB(HDD)/120 GB (SSD)
- **OS** : Microsoft Windows

Technical Stack

Front-end languages and Frameworks:

HTML:

Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

JavaScript:

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Bootstrap:

Bootstrap is a framework to help you design websites faster and easier. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, etc. It also gives you support for JavaScript plugins.

Back-end Framework and Database:

Node Js:

Node.js is a cross-platform runtime library and environment for running JavaScript applications outside the browser. This is a free and open source tool used for creating server-side JS applications. This framework offers a rich library of various JavaScript modules to simplify web development processes.

PostgreSQL:

PostgreSQL is a powerful, open source object-relational database system that uses and extends the SQL language combined with many features that safely store and scale the most complicated data workloads.

Machine Learning:

Python :

The **Python** language is one of the most accessible programming languages available because it has simplified syntax and is not complicated, which gives more emphasis on natural language.

J48 (Algorithm):

The J48 algorithm is used to classify different applications and perform accurate results of the classification. J48 algorithm is one of the best machine learning algorithms to examine the data categorically and continuously.

OpenCV (Library):

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

Visual Studio Code:

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control.

Data Encryption:

Triple DES:

Triple DES is an encryption technique which uses three instances of DES on the same plain text. It uses three different types of key choosing technique: first all used keys are different and in second two keys are same and one is different and in third all keys are same.

EXPECTED RESULT

The expected result is to come up with an edge-based biometric verification system which gives the same results as the existing system through an optimized process than that of the existing system.

REFERENCES

- https://us.norton.com/internetsecurity-iot-how-facial-recognition-softwareworks.html#:~:text=A%20facial%20recognition%20system%20uses
- https://www.electronicid.eu/en/blog/post/face-recognition/en
- https://www.idenfy.com/facialrecognition/#:~:text=A%20facial%20recognition%20system%20uses,helps%20to %20verify%20personal%20identity
- https://machinelearningmastery.com/introduction-to-deep-learning-for-facerecognition/
- https://www.intechopen.com/books/new-approaches-to-characterization-andrecognition-of-faces/automatic-face-recognition-system-for-hidden-markovmodel-techniques
- https://searchsecurity.techtarget.com/definition/biometric-verification.