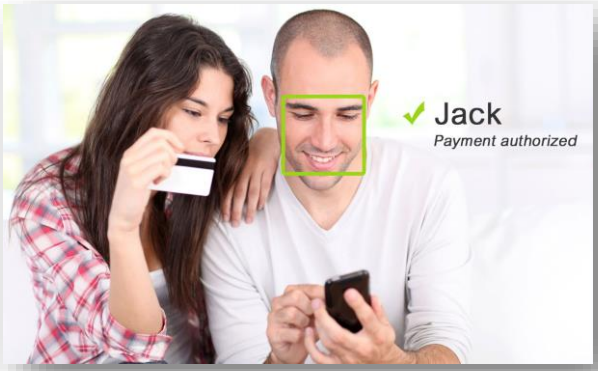
KeyLemon Online Authentication Solution
Oasis for Mobile Biometric

OVERVIEW

For rapid development of mobile apps utilizing face biometric authentication, KeyLemon provides the OASIS client-server library. Supporting both Android and iOS, and with a backend server that can be deployed privately on premises, the OASIS platform allows developers to quickly build out a scalable, mobile-centric biometric authentication platform.



SOLUTION ARCHITECTURE

A client-server architecture is an excellent choice for organizations that need to move quickly and are securing resources that are already online, such as financial transactions, corporate intranets, medical records, single sign-on password vaults, online courses, and private web traffic.

Client-server architectures are exceptionally fast and easy to set up, and highly scalable. Other advantages of a client-server architecture include the ease of applying technology updates and the ability to share biometric models across multiple devices via a centrally hosted database.

Oasis Solution is composed of a client and server library which have to be integrated as illustrated in Figure 1. The aim of the **Oasis Client library** is to acquire and prepare biometric samples for the Oasis Server library. The client library opens the camera, ensures the user is correctly placed in front of the camera with good lighting conditions, acquires biometric samples and prepares a data packet for the Oasis Server library. The **Oasis Server library** handles payload prepared by the Oasis Client and performs user enrollment (creates a face model) and user verification (test a user against a face model). A user face model is what uniquely identifies a user and is created during the user enrolment. Verification results are returned to the client library.

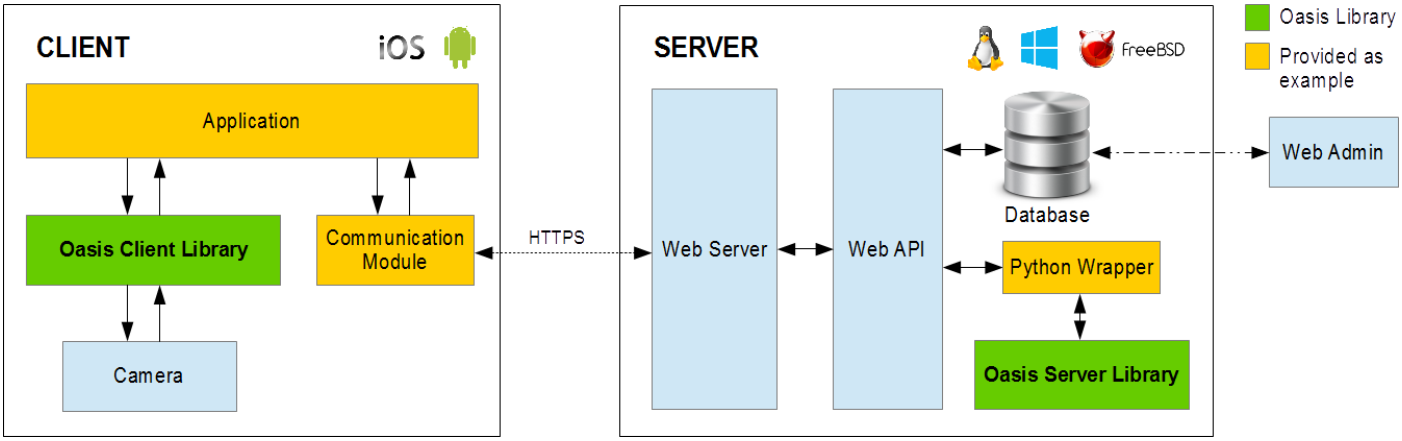
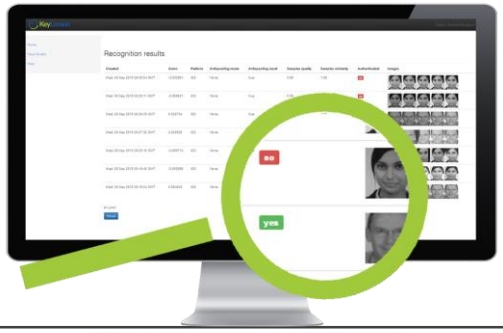


Figure 1. Oasis Architecture

With Oasis Solution, you keep full control on the Graphical User Interface (GUI) of the client application and communication between client and server application. Application (with GUI) and server communication are provided as example.

FEATURES

- Face enrollment and verification
- Manage access to the camera and provide guidance in real time to help user capture optimal biometric sample
- Liveness detection
- Rapid Mobile App Development



KeyLemon S.A., rue du Collège 1, 1920 Martigny, Switzerland
www.keylemon.com / info@keylemon.com

the way your friends ...
... recognize you

KeyLemon Online Authentication Solution
Oasis for Mobile Biometric

A) Oasis Client Library (Android / iOS)

Oasis Face Client Library is the client side part of the KeyLemon's Oasis solution. It is a software library that can be integrated in native Android or iOS mobile applications. The aim of the library is to acquire and prepare biometric samples to be sent to server integrating the Oasis Server library. The library opens the camera, ensures the user is correctly placed in front of the camera with good lighting conditions, acquires biometric samples and prepares a data packet for the Oasis Server.

The library can prepare two different packets for the server:

- A packet for enrollment, used to create a new biometric model
- A packet for verification, used to verify the identity of a person

The library can be controlled with a set of functions and callbacks which provide all the needed information to update the graphical user interface (GUI).

The acquisition process is performed in two steps.

1. The first step is the image quality step that ensures user's face is frontal and lighting conditions are good enough.
2. The second step is the biometric sample acquisition during which a sequence of images will be captured and processed.

The following image quality checks are performed during the first step of acquisition and will help you to provide guidance to the user in real time to help adjust for optimal conditions to capture biometric sample.

- Face position checks: check that the face is well detected, correctly centered and in a good distance from the camera.
- Face angle checks: check that the user is facing the camera with a correct head angle (in terms of pitch and yaw angles)
- Image conditions checks: check the image quality in terms of lighting.

Spoofing countermeasures can optionally be activated to ensure the user isn't trying to spoof the system during the verification phase. Three different spoofing countermeasures can be activated in the library:

- Eye blink detection
- Head movement challenge-response
- Combination of eye blink detection and head movement challenge-response

Oasis client package contains the sources of an example application which illustrates a possible integration of Oasis Client into an application. You are free to use it as a starting point for your implementation. Please note however that you remain responsible for the GUI and the communication with the server.

ANDROID SPECIFICATIONS

The library is available as an Android Archive Library (AAR) for Android Studio. The library is available for development in Java but includes native code.

Requirements:

- Android 4.1 Jelly Bean (API level 16) or greater
- A front camera with a sufficient framerate (> 15fps)
- CPU supporting armeabi-v7a or amr64-v8a
- Computing power of the Samsung Galaxy S3 or higher

iOS SPECIFICATIONS

The library is distributed as an iOS Framework for Objective-C development in Xcode.

Requirements:

- iOS 8 or greater
- iPhone 5 or newer
- 2012 model iPad or newer
- CPU supporting armv7 or arm64

B) OASIS Back-end Server Libraries

The OASIS Face Server is a library to be integrated in a server getting requests from an OASIS Face Client. This C library provides functions to create face models and test images against existing models. The library is available for Windows (64-bit) Linux (64-bit) and FreeBSD (64-bit).

SERVER SPECIFICATIONS

The server library provides function to:

- Process a packet for enrolment generated by Oasis Client and create a face model
- Process a packet for verification generated by Oasis Client and returns a verification result.

During enrollment and verification, image quality indicator is also computed and will help you to have another indicator of the quality of the samples acquired by the client. It combines analysis of brightness, blur, sharpness and other factors.

During verification, samples quality similarity is also computed and will give a comparison of whether the quality of the current attempt is close to the quality of the enrollment.

The library interface is in the C programming language, making it easy to integrate and interface with any other language. This library comes with code examples (including a python wrapper) and documentation.