Entry

How Entry changes the way we authenticate

Entry's mission is to facilitate the shift from a service provider owning user identities, sensitive PII, and credentials to a user-owned digital identity, PII, and credentials. It was designed to put a user in control of their verified digital identity, to provide transparency and control over how and where their data is used, with an ability to govern access to it.

This article provides a comprehensive overview of what we built (and how we did it) and the reasoning behind the product. The first part explains the challenges we are tackling, the value we add to both end-users and companies, and sets the goals in plain English. The second part is more technical, with the quantitative evaluations and a more detailed discussion of security and privacy concerns.

New possibilities and existing challenges

Face recognition, as a new medium of authentication, enables its users to do multiple essential things that were not possible with a combination of password and push notifications:

- Secure high-value transactions.
- Protect accounts from phishing and theft.
- Confirm your real identity — e.g., when renting a car or signing up for a gig job.
- Retrieve access to existing accounts easily.

These use cases are handled by Identity Providers (IdP) — complex web applications doing all the work behind the scenes when you are logging in, confirming your email, or allowing you to share your profile info with other apps you use. On the one hand, developers "outsource" all login and signup processes to them. On the other hand, developers depend on providers' capabilities, often bearing extra costs of having customer support manually validating users. Everyone in our team has been annoyed with forgotten passwords, user verification requirements, or account activation taking an extra day. So we have built a solution that takes all these problems away, both for users and developers, enabling the latter to provide an even better user experience.

Putting face recognition technology out there was not the biggest challenge. The most difficult was to make it safe and to make it play well with the other systems. Boiled down to a few essential requirements, it meant:

- Give our users full transparency and control over their biometrics, stored data, and permissions they give to other apps.
- Make it impossible to abuse or steal personal data by anonymizing it internally.
- Allow any developer to integrate Entry into their app effortlessly without learning new APIs or disrupting their workflow.

These requirements are intentionally generalized: there are hundreds of hours of work and countless tickets behind each of them. In the next section, we explain how we achieve these goals

and how we measure our progress.

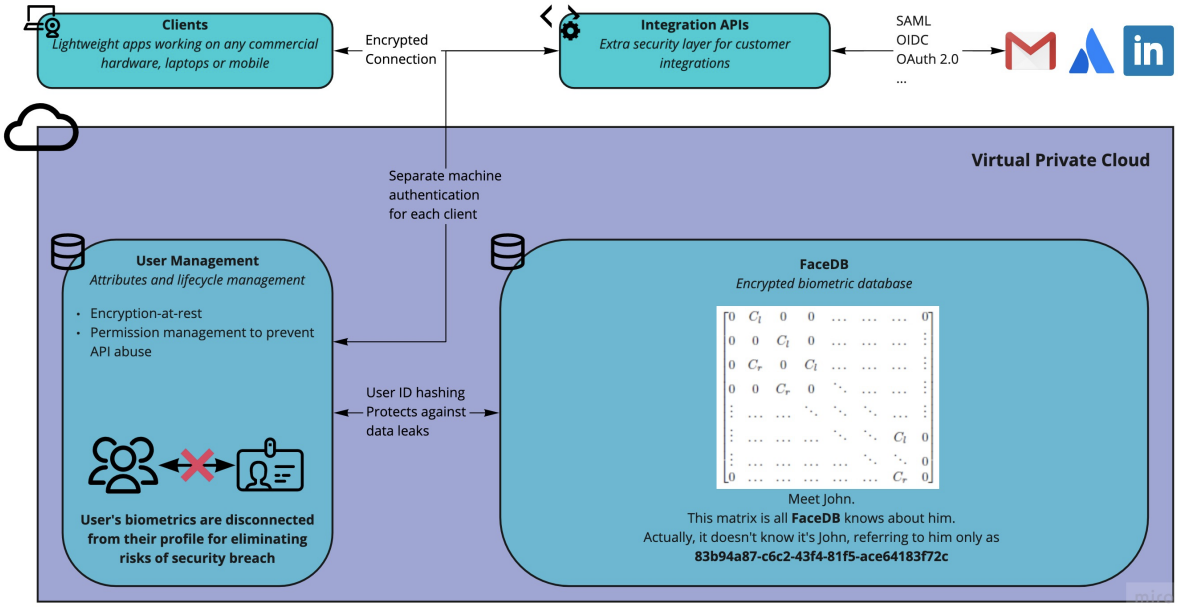
Technical overview

This section provides a technical overview of Entry, and can be divided into three logical parts:

- **Architecture**: this section explains how we design the application and data management systems, starting with the first principles.
- **Anti-spoofing**: this is a deep dive into our models for detecting attacks, with quantitative evidence of our model outperforming current state of the art in this space.
- **Privacy and security concerns.**

Architecture

Here we will limit ourselves to discussing the main principles, goals achieved by following them, and general approach to implementing components of the system. We intentionally omit the critical details for security reasons.



High-level architecture diagram of Entry, focusing on the main concerns from the security and privacy perspective.
FaceDB: safe storage of personal sensitive data eliminates the risk of data leaks.
User Management: user data (including PII) and user biometrics are separated.
Clients and integrations: communications are encrypted and verified, all enterprise clients are compartmentalized.

Zero trust between machines

> All internal services are treated as though they are external, and potentially compromised.

All internal services are treated as though they are external, and potentially compromised. Therefore, all interactions between internal components are fundamentally restricted:

- No direct access to the databases.
- Cloud-based components authenticate to each other.
- Minimal access policies for all services and APIs.

Biometric privacy

> It's impossible to trace an existing user profile "John Smith" to a stored set of biometric data belonging to John Smith, but very easy to verify that the person in front of the camera is, in fact, John Smith.

The goal here is to compartmentalize actual data that belongs to users (e.g. biometrics) and metadata that is being used by our systems (e.g. for granting a user access to their apps after a successful identification). To achieve that goal we adhere to following principles:

- **Physically separate storage**: data and metadata are managed by separated infrastructure components.
- **Elimination of actual links**: since all we need to grant access is a successful verification of biometrics, we don't store any foreign keys that would allow to trace someone's biometric profile from their account.

Availability for commercial hardware

Building a general-purpose identity provider required us to design it for working on anything with a webcam — not only desktops, laptops, and smartphones, but also more specialized hardware, e.g. smart locks.

Our client applications are supported for all modern browsers and mobile platforms, making Entry immediately available for end users without the need to obtain hardware tokens or install additional apps.

Core concepts and methods

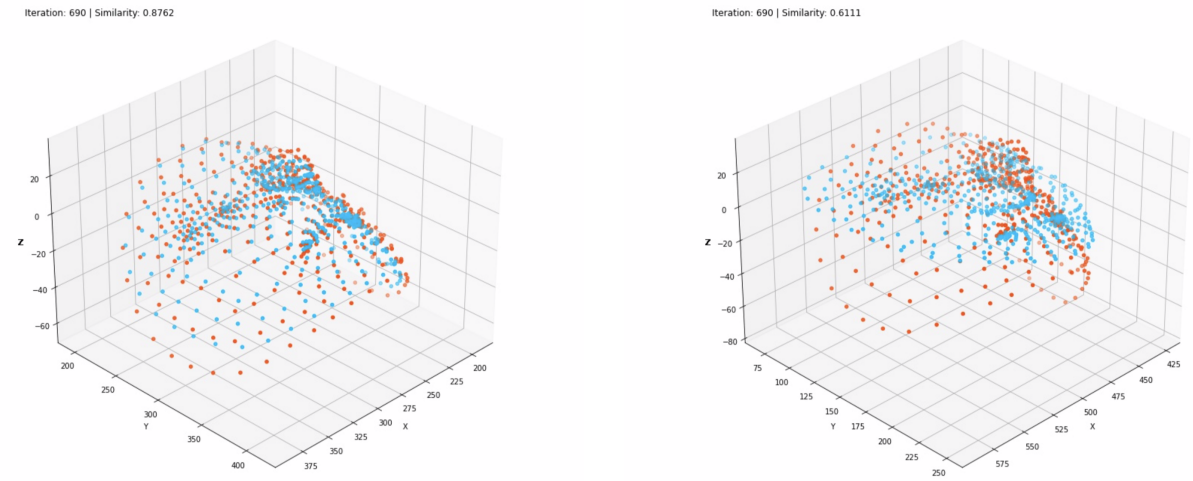
Biometric Sessions

We consider Sessions to be sequences of base classifications with subsequently applied voting. These sequences have variable lengths measured in number of input frames, and each one culminates in a voting operation that is treated as the final classification — both for identification and spoofing detection.
Generally speaking, this is a method of "smoothing" predictions that does not influence how the performance is measured directly but is aimed at self-correcting internal errors instead. Overall improvements achieved by this method depend on the proprietary voting method, which we don't explain in this whitepaper — but providing the quantitative results that allow both comparisons to open benchmarks and assessing potential performance in real-life application, which is our primary goal.
To make the final decision on whether a session is an attack, we use a combination of voting applied to base predictors, a voting algorithm, and face reconstruction model output.

Legacy: Face reconstruction

In previous versions of Entry we used to perform a 3D face reconstruction from multiple frames along with the attack detection. Operating on consumer hardware (anything with a camera, without any requirements for stereo or IR capability) was the main reason behind that — incorrectly reconstructed face from an otherwise normal video feed is a very reliable indicator of an attack.

The new version of Entry anti-spoofing model eliminated the need in face reconstruction, outperforming it both in accuracy and inference time — but we consider it important to highlight how much information about the face in the video is analyzed implicitly in the new version.



Face Reconstruction results: blue dots are facial landmarks from candidate Session, orange dots are from a target reconstruction obtained during registration. Left: legitimate session; Right: printed attack.

Applicability

Our architecture required defining Sessions in this generalized way for several reasons:

- We needed to have a separate voting model that we would be able to update without radical changes to prediction models. This approach ensures overall stability and backward compatibility of both recognition and anti-spoofing, which is critical.
- Base models that work on single frames can fail in unpredictable ways — mainly because they are not rich enough to cover most real-life interactions reliably. But when the datasets for backend models reach a certain point, an efficient voting mechanism balances out existing blind spots, making improvements steadier when we train on more data.
- In our security model, real-life interaction between the end-user and the application is a critically important in evaluating the user's authenticity. Depending on how the user interacts with the camera, we can require more or less video input to make a confident prediction. We explore further how this method flexibly adjusts Sessions to make them easier for more "trustworthy" subjects.
- A wide range of biometric attacks work very well on systems working with single frames ("replay" attacks, printed faces, or photos from phone/tablet screens). Switching to video input by itself makes it easier to detect them.

Data

Base predictors are trained exclusively on our proprietary datasets. To better showcase its performance, focusing on how well it generalizes across the wide range of benchmarks, we don't finetune it on open data and use the same parameters everywhere.
Our dataset contains over 80000 recordings of biometric sessions, collected from ~45000 individuals. Apart from being significantly larger than any other known dataset, it has a highly unusual attack/legitimate session split — the proportion between attacks and legitimate sessions is usually closer to 95/5, mostly because of the cost of collecting such data. In practice, it causes anti-spoofing systems to be overly sensitive, destroying user experience with frequent false detections. We have solved this problem by including a portion of legitimate sessions that is highly representative of what we generally expect our users to do, achieving a very stable performance of base models, and preparing them for organic improvement: production usage improves the models further, reducing the chances of false detections without compromising security.

Metrics

Half Total Error Rate (HTER)

The main metrics and their statistical background are described in https://www.isca-speech.org/archive_open/archive_papers/odyssey_04/ody4_237.pdf. For quantitative comparison on academic datasets we report HTER percentage (all values multiplied by 100 for readability):

$HTER = 100 \cdot \frac{FAR+FRR}{2}$ where

$FAR = \frac{FA}{NI}$, false acceptance rate

$FRR = \frac{FR}{NC}$, false rejection rate

FA — number of false acceptances

FR — number of false rejections

NI — number of sessions performed by impostors

NC — number of sessions performed by authentic clients

Notice that for existing benchmarks we can safely define sessions without any changes in interpretation of results.

As with all other error rates, lower is better.

Results

We present evaluations on several difficult benchmarks used in face anti-spoofing security research:

- **MSU Mobile Face Presentation Attack Database** was the baseline benchmark for evaluation, it contained videos from 55 subjects taken on MacBook Air and Google Nexus cameras, 330 videos of attacks.
- **OULU NPU** is the larger dataset with different lighting conditions and multiple input types: six mobile devices, 1340 videos of attacks.

- **IDIAP Replay Attack** dataset contains six type of attacks, each one using both handheld and mounted devices, 1300 videos total.

Reference URLs:

- MSU Mobile Face Presentation Attack Database: http://biometrics.cse.msu.edu/Publications/Face/WenHanJain_FaceSpoofDetection_TIFS15.pdf
- OULU NPU: https://sites.google.com/site/oulunpudatabase
- IDIAP Replay Attack: https://www.idiap.ch/dataset/replayattack

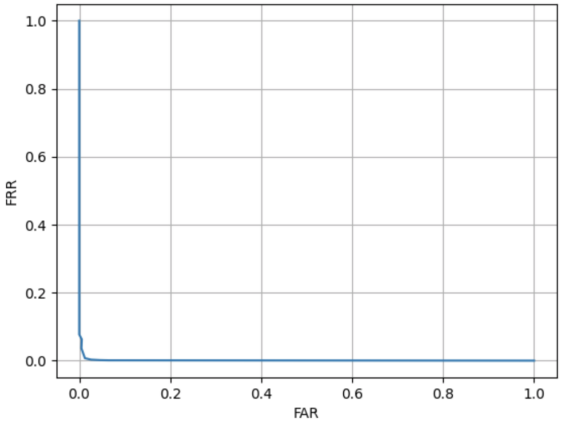
HTER (percentage) comparison: Entry vs. 2021 State-of-the-Art results

Aa Benchmark	# Entry V2	# SOTA (June 2022)
MSU Mobile Face Presentation Attack Database	0	23.5
OULU NPU	2.6	3.9
IDIAP Replay Attack	0	6.5

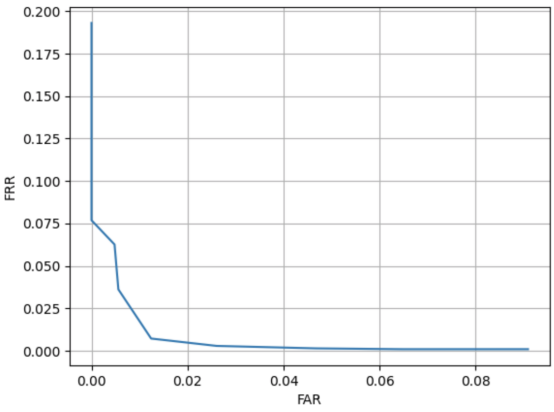
Testing on internal benchmarks

Aa Metric	# Entry: base model	# Entry
HTER	3.54	0.74

Full Entry detection model consistently outperforms base classifiers, showing that our voting algorithm improves an already robust detector's overall reliability.
Below is our ROC curve for analyzing the performance of attack classifier:



Full ROC curve on internal test benchmarks. We have to confess that the only reason we added a canonical ROC curve is to show off, to see the actual tradeoffs please see the zoomed-in version on the right.



Zoomed-in version of ROC curve: for spotting the remaining tradeoffs.

Privacy and security concerns

When we approach end-user privacy challenges, we distinguish the problem and the method: the actual requirement that we need to deliver on can be very different from expressed concerns. For

most identity providers and applications handling sensitive personal data, the main concerns from the end user's perspective are:

- How well does it protect users from non-technical security problems, e.g., fraud, phishing, identity theft?
- Is the app resistant to data leaks? What happens when the credentials or databases are leaked/stolen?
- What is held on my device, and what's stored in the cloud? Is storing data in the cloud safe?

Data leak protection: storing biometrics in the cloud

Storing and managing user biometrics and attributes in the cloud is the central part of our architecture. Internally we have two major components: the biometric identity part and the user management part. Biometrics are handled by a specialized database developed in-house. One of the main privacy-preserving features is that you cannot trace individual biometric records to a particular person outside just by looking at the data. There are no credentials in the usual sense of the word, so there is nothing to steal from the DB that would potentially have value. The user management part stores external IDs that are meaningless by themselves — they merely provide the connection between customers and their hashed identification results. It allows us to preserve their privacy to the point when even having full access to all databases does not allow a hacker to examine or steal someone else's biometric identity.

Recovery of stolen accounts

Hacking an account once doesn't make it easier to hack it again since there are no credentials to steal. The response protocol for such a situation is to terminate compromised user sessions and flag the account as "requiring extra protection" — meaning that for some time, the user would need to interact with the camera a bit more during login challenges.

Protection from the main threats: fraud, phishing and identity theft

When a user logs in using some credentials (e.g., password, hardware keys, trusted device), services implicitly trust those credentials, bearing the risk that they could be stolen or forged. There is no way to know who is behind those credentials — or, for that matter, if that's a real person. Face biometrics with reliable liveness detection and anti-spoofing provides the main guarantee: on every authentication, we are certain that there's a specific real person on the other end.

Acknowledgements

Thanks to Alex Krizhevsky, Daniel Gackle, Dr. J. Seth Strattan for reviewing the draft of this post.