humanID

One-Click, Anonymous Login

OUTDATED - last updated on 06/30/2020

Updates coming soon*

Summary

- humanID creates a **fully anonymous, privacy-first identity** that enables communities to block bots and abusive users
- humanID protects the identity of its users through an irreversible hash
- This enables platforms to decrease the cost of moderating and monitoring content, and to build civil and safe communities

Overview

humanID is an nonprofit, open-source initiative building a replacement to social media logins like "Login with Facebook". Unlike existing logins, humanID has developed technologies that massively increase the cost for automated accounts, bots and previously banned users to return to a website. humanID also allows users to sign onto third-party websites or apps with full anonymity, creating higher trust and protection from data breaches - fully GDPR/CCPA compliant.

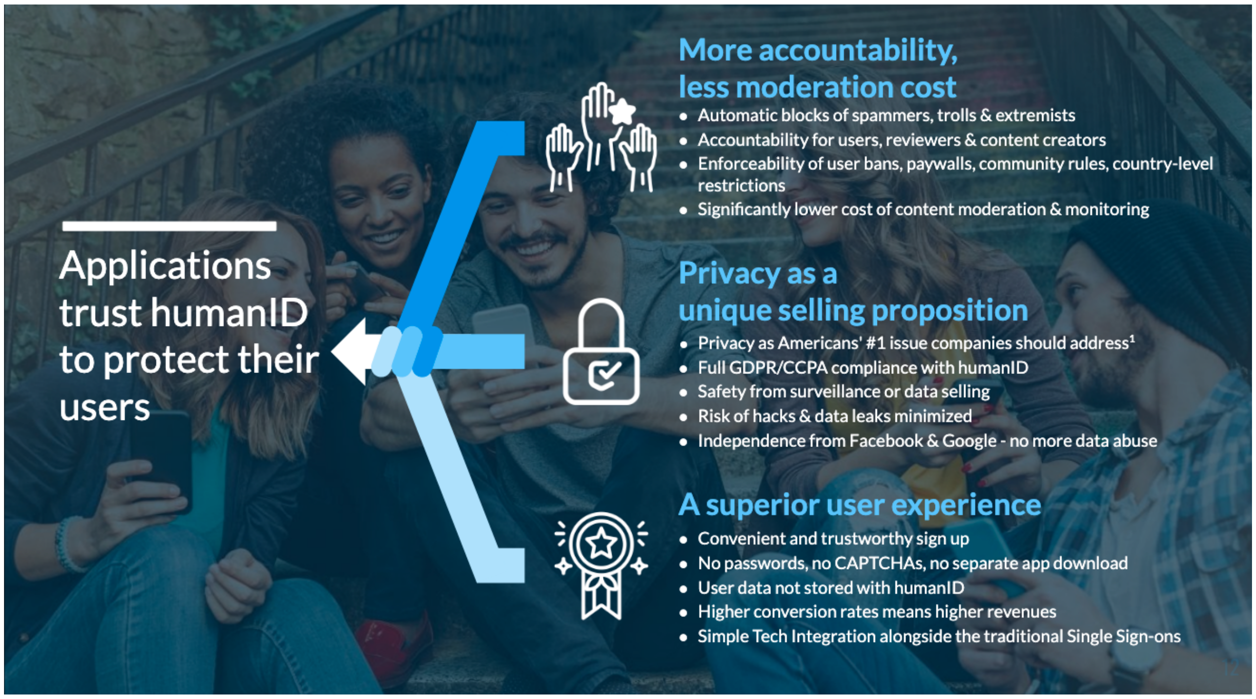
The team is proud of strong and diverse leadership on both the technical and business aspects, building world-class technology as public infrastructure for the web of tomorrow.

humanID's leadership team includes experienced cyber-security experts, seasoned entrepreneurs, and graduates from some of the best schools in the United States. Our early development was supported and funded by Harvard Business School.

Why humanID

- Communities currently do not have a way to block bots, spam and bullies from their communities, and enforce community rules.
- Malicious actors abuse the lack of an accountable, persistent online ID to return with new accounts, create bot networks, and manipulate everything from reviews to ranking algorithms.
- Big tech is distrusted by an increasing number of people, leading users to reject authentication methods such as "Login with Facebook"
- 15.1 Billion data records breached in 2019, costing U.S. companies an average of $3.92M, as well as the trust of their users - an existential threat to their business!
- Studies have shown that users that feel like their privacy is protected are more likely to engage with an online service, affecting conversion and engagement rates.

By bringing back security, privacy and accountability, humanID enables the safe & civil online communities of the future.



Applications trust humanID to protect their users

More accountability, less moderation cost

- Automatic blocks of spammers, trolls & extremists
- Accountability for users, reviewers & content creators
- Enforceability of user bans, paywalls, community rules, country-level restrictions
- Significantly lower cost of content moderation & monitoring

Privacy as a unique selling proposition

- Privacy as Americans' #1 issue companies should address[1]
- Full GDPR/CCPA compliance with humanID
- Safety from surveillance or data selling
- Risk of hacks & data leaks minimized
- Independence from Facebook & Google - no more data abuse

A superior user experience

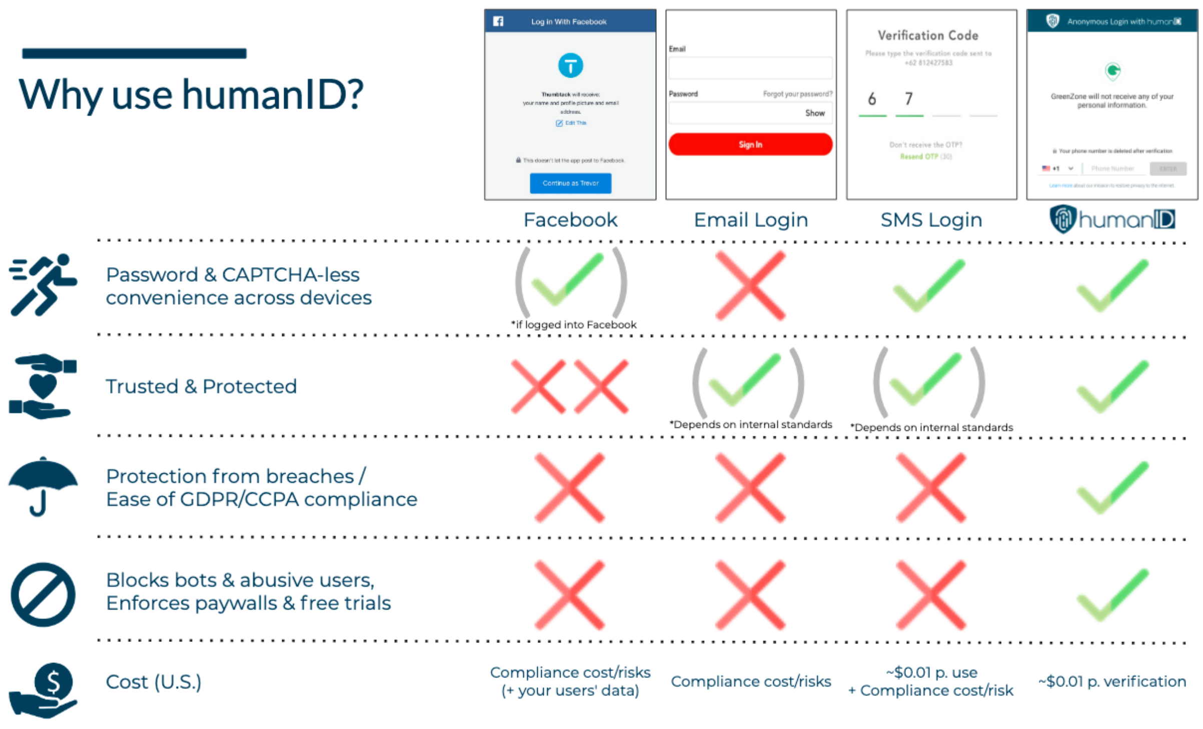
- Convenient and trustworthy sign up
- No passwords, no CAPTCHAs, no separate app download
- User data not stored with humanID
- Higher conversion rates means higher revenues
- Simple Tech Integration alongside the traditional Single Sign-ons

How does humanID compare

humanID creates a fully anonymous, privacy-first identity that enables communities to block bots and abusive users - creating accountable, civil communities.

We match Facebook in terms of convenience for both use and integration, while offering full privacy and anonymity to individuals, and a plug-and-play SDK for developers.
Other than Facebook, we do have no financial incentive to tolerate hundreds of millions of fake accounts on our platform - enabling safer, more civil communities.

Many private startups have recognized the clear need for a solution, but suffer from similar incentive reasons. Blockchain based companies build independent technology, but make it hard for users and developers to understand - and adopt - their technology.



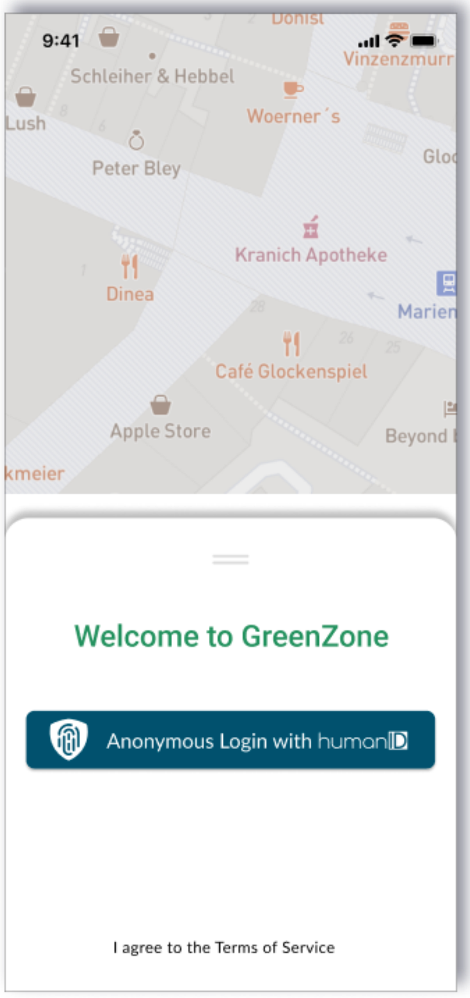
Why use humanID?

	Facebook	Email Login	SMS Login	humanID
Password & CAPTCHA-less convenience across devices	(✓) *if logged into Facebook	✗	✓	✓
Trusted & Protected	✗✗	(✓) *Depends on internal standards	(✓) *Depends on internal standards	✓
Protection from breaches / Ease of GDPR/CCPA compliance	✗	✗	✗	✓
Blocks bots & abusive users, Enforces paywalls & free trials	✗	✗	✗	✓
Cost (U.S.)	Compliance cost/risks (+ your users' data)	Compliance cost/risks	~$0.01 p. use + Compliance cost/risk	~$0.01 p. verification

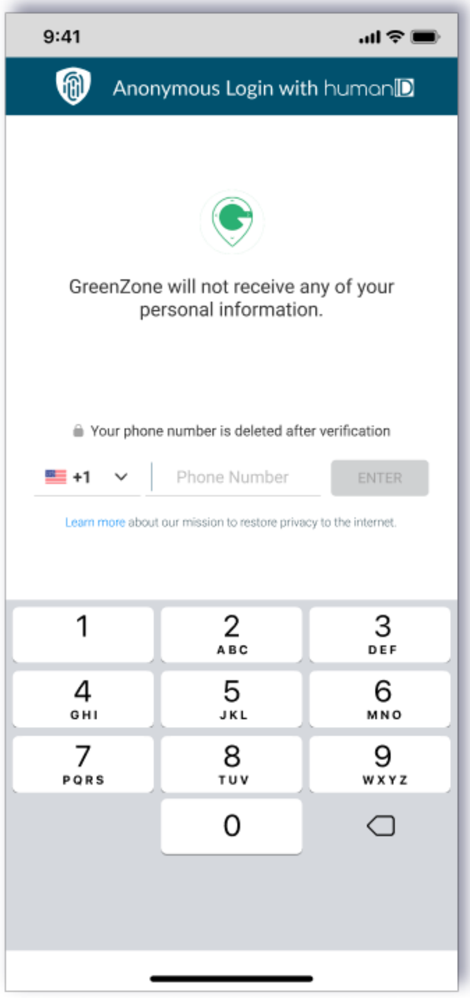
How does it work?

humanID keeps users' data completely private by deleting it after verification and never communicating it to our partners. We generate a cryptographic hash unique to a user's phone number and the specific platform. Thus, services such as health and dating apps, social networks, VPNs or even a COVID-19 spread tracker can be accessible entirely anonymously.

Example implementation:



I agree to the Terms of Service



GreenZone will not receive any of your personal information.

Your phone number is deleted after verification

Learn more about our mission to restore privacy to the internet.

For the current flow in action, please see this video.

humanID is and will be free for end users.

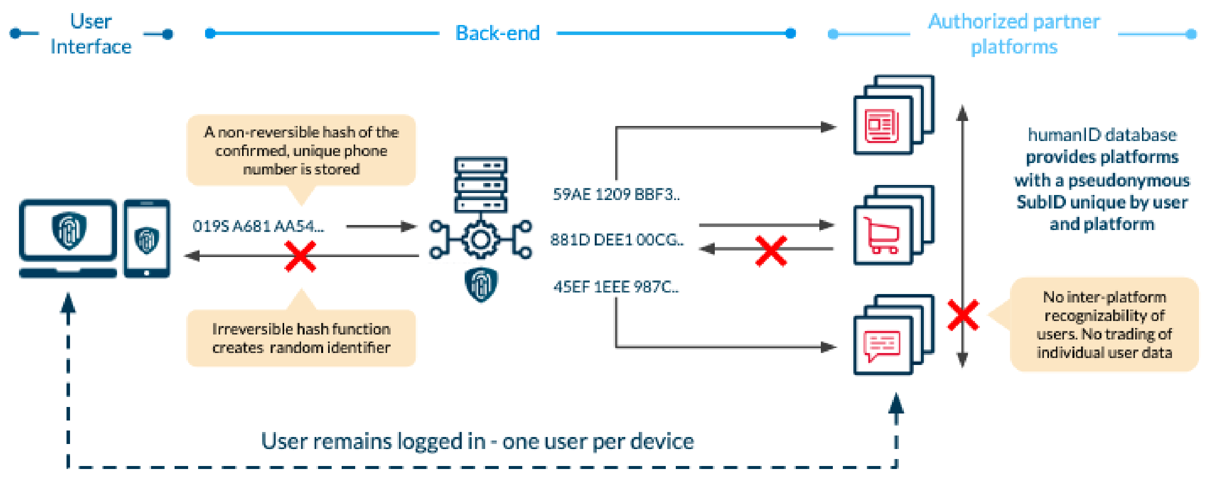
Third-party websites and applications pay humanID because we help them build better communities. We use this payment to cover the expenses of delivering the service. As a nonprofit with the clearly defined vision of one safe, digital ID for every human, we have no profit incentive to break our promise.

humanID blocks automated accounts, cyber-bullies, trolls and freeloaders from websites and applications.

By requiring a unique phone number, as well as setting strict limitations on the usage and switching of devices, we enable websites to hold users accountable, e.g. via permanent bans or limitations to a free trial period. Unsophisticated users will be unable to return, as they currently can, to a platform that limits them. Sophisticated bot networks will be at least 40x costlier to operate and easier to identify.
For communities and platforms, this will mean higher sign-up and engagement rates, better enforcement of rules and paywalls, lower moderation and monitoring cost, and overall a better customer experience and the ability to turn privacy and trust into a competitive advantage.

humanID enables a better internet by building private and safe identity online.

High Level Specifications

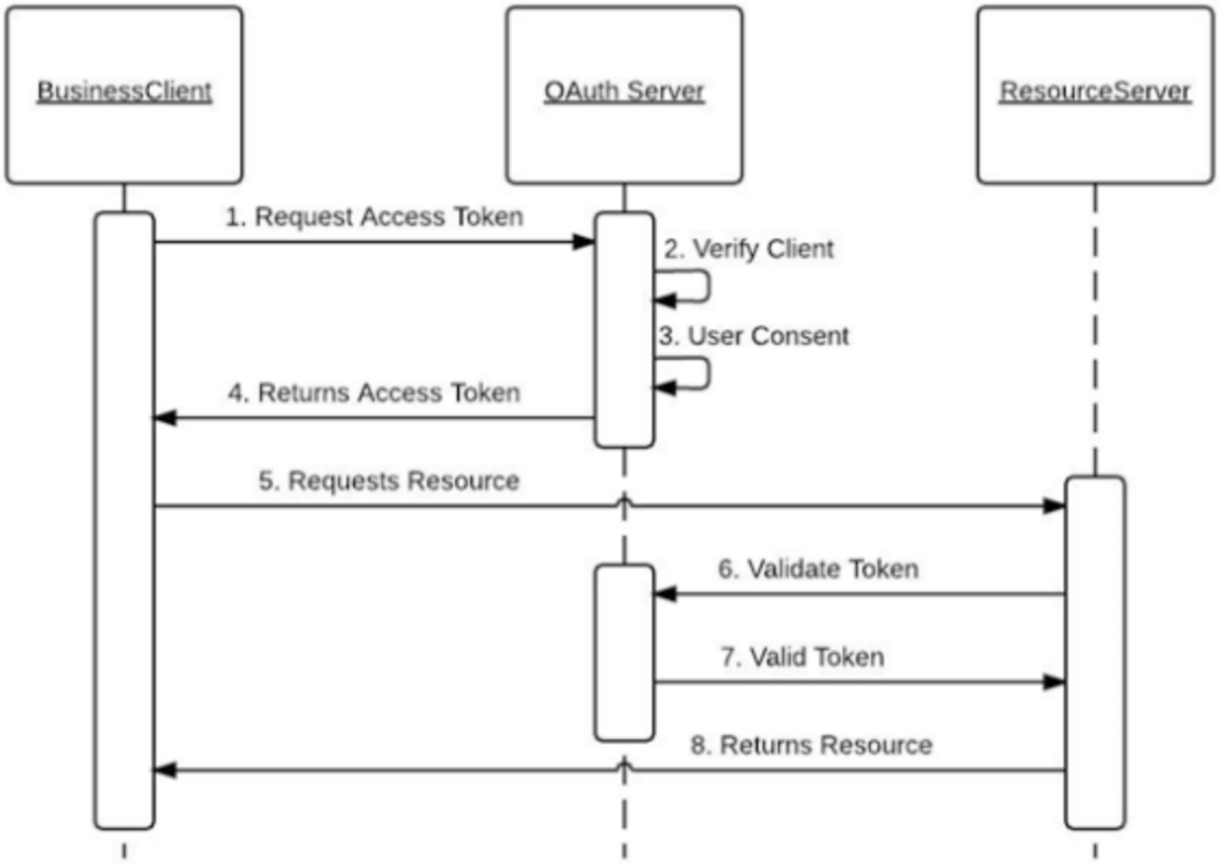

User Interface
Back-end
Authorized partner platforms
A non-reversible hash of the confirmed, unique phone number is stored
0195 A681 AA54...
Irreversible hash function creates random identifier
59AE 1209 BBF3..
881D DEE1 00CG..
45EF 1EEE 987C..
humanID database provides platforms with a pseudonymous SubID unique by user and platform
No inter-platform recognizability of users. No trading of individual user data
User remains logged in - one user per device

System Architecture Diagram

humanID implements the OAuth2 architecture..

The diagram below consists of three parts:

1. **BusinessClient**
   BusinessClient is the partner/3rd Party App. This is the client that needs to Request Access Token and Request Resource from the Resource Server. The client app can be either a mobile application or a web based application.
2. **OAuth Server**
   Implements standard OAuth2 implementation, this one is hosted by humanID, but can also be self-hosted by 3rd Parties.
3. **Resource Server**
   Resource Server is the 3rd Party API server that has the resources that are needed by BusinessClient. This is hosted by a 3rd Party.


BusinessClient
OAuth Server
ResourceServer
1. Request Access Token
2. Verify Client
3. User Consent
4. Returns Access Token
5. Requests Resource
6. Validate Token
7. Valid Token
8. Returns Resource

humanID Implementation

For humanID integration with a 3rd Party App, for each app built they will need to apply for humanID credentials by emailing developers@human-id.org

Every App created will have **App ID**, **App Server Secret Key,** and **App Client Key.**

App ID

The App ID (Application ID) is a unique ID for each app that is created and wants to use humanID. AppID will be auto-generated by the humanID server.

For example: **9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08**

App Server Secret Key

The App Server Secret Key is a key used by 3rd party App/Server/Host (Backend) that wants to communicate with the humanID server (See Figure 1. OAuth2 High Level Architecture, in the step 6 and 7).

This key **should be private and safely kept** in Server Host.

For security concerns:

1. Do not expose to client side, such as using in client side Javascript, or use it in Mobile App code (Android/iOS)
2. Do not store in git or any source code repository
3. Do not use the same key for Production environment, with Development or any other environment
4. The key can be re-generated**, but this will affect only communication between 3rd party Server App with humanID server (Configuration update will be needed)**

Usage of this key **must be restricted**, by registering the IP Address of the host of the 3rd party Server App.

App Client Key

The App Client Key will be used in the MobileSDK, to connect with humanID via API Call.

Usage of this key will be restricted by registering an Android Package ID with the sha-1 certificate fingerprint. By doing that this key can only be used by authorized apps. If this key is used in Web applications, the restriction will use a domain name, e.g. somecompany.com.

The implementation details:

1. **Request Access Token**
   Implemented by using **RequestLoginOTP**, from the **EndUser**(Business Client), to humanID
2. **Verify Client**
   This step consists of:
   1. CreateOTP, create a randomized 4-digit numbers
   2. SendOTP, send the OTP numbers to the user via SMS
   3. EndUser accept the OTP
   4. Call the method Login to the humanID server
   5. humanID server Validate the OTP:
      1. If valid, the humanID will check whether this user already has an account or not. The mechanism on generating a humanID Account will be documented in the next section, if the account is not found then it will auto create the account.
      2. If **not valid**, humanID will response **"Verification failed"**
3. **User Consent**
   Implemented by a User accepting the OTP SMS and putting the correct OTP in the humanID verification dialog.
4. **Return Access Token**
   Implemented by returning the "ExchangeToken" (have a limited time for usage). This token will be returned to the EndUser and stored in the client-side storage for temporary use. This token will be needed to be included in the next request to access 3rd party partner resources.

5. **Requests Resource**
   The client want to access the 3rd party partner resource, and include the humanID "ExchangeToken" in the Header Request
6. **Validate Token**
   Implemented on the 3rd party Server (Backend) that need to call humanID API to verifyExchangeToken
7. **Valid Token**
   humanID will return whether the token is valid to 3rd party Server App.
8. **Returns Resource**
   3rd party App creates a session in their apps for the user if needed, and return the resource (e.g JSON response)

humanID Account Generation

humanID Account ID is a generated ID for each user as an unique account for humanID. We identify the user using a phone number (MSISDN).

For each person with the same phone number, if sending a request to humanID, humanID needs to know whether this person already has an account or not. This is implemented by generating an Unique ID for each phone number.

But instead of storing the phone number in the database in plain-text, **we will create a "hash"** from the phone number itself, with a **one way hash**. Hash is a method to generate a string (sequence of characters from an input).

Hashing Details

Formula:

hash (x) = y

Example of hash:

hash("6281287765551") =
22bf509dee216179e84ad2ec7d417e6dc56558d6f9ac3714874d06baa9621a42

The specific hash we use is the **SHA512** hash.

Adding Salt Key Before Hash

To make it more difficult to brute force, when generating the humanID Account ID, we will concatenate the phone number with a Salt Key (another string that will be appended before the hash).

sha512hash (Salt_Key + Phone Number) = Hash Result

The Salt Key is the combination of lowercase letters, uppercase letters and numbers. For SHA512 hash which is 64-bits, the recommended salt key is 64-bits.

For illustration purpose only:

1. Example the Salt Key is: 2xMnLp9uXcMgHzBgTvDg1LmaX2z8z0q4KvL
2. Hashing of Salt Key and Phone Number: sha512hash (2xMnLp9uXcMgHzBgTvDg1LmaX2z8z0q4KvL + *6281287765551)*
3. The result is: *65426599e640526d2f18789bcd994dacb721b92d8a2fd7a12a499c417ca84f7c*

By doing this, the hash generated will be known as **humanID Account ID.**

Exchange Token Generation and Validation

Token Generation

Generating an Exchange Token will be handled by the humanID Server. The exchange token is generated and will be validated later on.

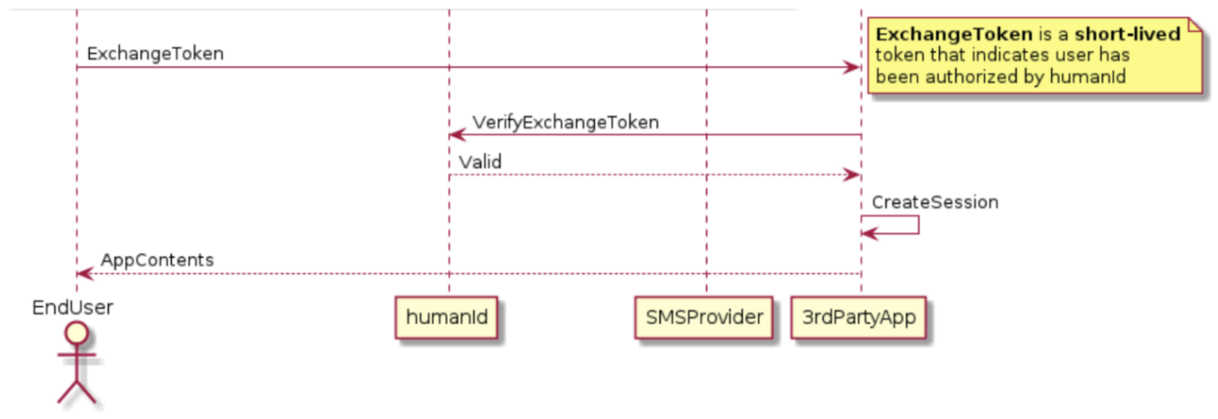
The exchange token will be returned to the EndUser (such as Web based app or Android/iOS App) and will be included in the Header Request when access the Resource/API from the 3rd party. 3rd party Server will need to call humanID API to check whether the ExchangeToken is valid or not.

ExchangeToken is **short-lived**, which means we have a time period we can set the period at humanID server, for example the token is only valid for one hours, one day, etc. If the token is expired then means the user needs to authenticate again.

There is also a mechanism to get a new token by using the old token (refreshing token), this can be implemented later in the future.

Token Validation

humanID will provide an API for validating an ExchangeToken. This API will be called by 3rd party App server. humanID will determine whether the ExchangeToken given is still valid or not.


ExchangeToken
ExchangeToken is a **short-lived** token that indicates user has been authorized by humanId
VerifyExchangeToken
Valid
CreateSession
AppContents
EndUser
humanId
SMSProvider
3rdPartyApp