A Single-Sign-On Security Platform for Private and Decentralized Apps

The PlusUltra platform deploys a client-side encrypted, peer-to-peer synchronized, and backed up data store with a clean, easy to use account-style interface.

The PlusUltra platform supplies a fully client-side account that creates, manages, and encrypts private keys, public keys, and App data on the user's client device and uses server side assistance for backup and sync of encrypted data and for authenticating users.

PlusUltra platform users retain full control of their private keys at all times with neither PlusUltra nor third parties having access to private keys or transaction data.

The platform contains optional blockchain "plugins" supporting transaction functionalities on several blockchains such as BTC, LTC, BCH, ETH, and ERC20 ETH tokens.

Data security and backup is implemented in a cross platform Javascript library compatible with NodeJS, browsers, and React Native.

Referred to as PlusUltra Core (PUC), PUC implements all account and wallet creation, user authentication, management of transaction data, encryption, and sync across devices.

The PUC architecture is inherently “client-strong” meaning nearly all logic exists and operates on the client device.

The servers exists only to assist in authentication, for encrypted backups, and as nodes on the blockchain network.

PUC leaves control of the private keys entirely in the hands of the user or app which controls the login and password of the account.
PlusUltra servers have no capability of spending funds on users’ behalf.

Client-Server Architecture

PUC uses a “95% decentralized” client-server architecture which handles 3 separate types of servers.

1) The first is a conventional app server developed in Python/Django with a PostgreSQL database. Although this is auto-replicated and backed up, this is still a centralized server. It is used for creating new accounts and authenticating existing accounts. In PUC, it is referred to as the ‘auth server’.

This server incorporates automatic dual server replication and encrypted hourly backups to an offsite server. Unavailability of the auth-server still allows PlusUltra SDK apps to function for blockchain transactions and synchronization between devices which have accounts that were previously logged in.

2) PUC uses a semi-decentralized,P2P network of data-sync backup servers running the Git protocol and software. Referred to as ‘sync-servers’, this pool of Git servers only hosts data that is pre-encrypted by the client devices running PUC. These are true P2P servers in that there are no primary or secondary servers, however all sync-servers and the auth-server are operated by PlusUltra. A client can connect to any of the sync servers to upload or download data.

Any new or modified data sent to a sync-server is automatically replicated to other sync-servers, using Git protocol to resolve conflicts and provide modification history and rollback if needed. To scale this architecture, “pools” of sync-servers will be created in the future such that each pool will replicate data across the sync-servers in the pool, but not across other pools. If all sync-servers were unavailable, PlusUltra SDK apps can still function to transact funds directly from public blockchain nodes.

3)PUC relies on public blockchain full nodes running Electrum for doing transaction detection and broadcast. These servers are publicly accessible nodes hosted by PlusUltra and other community members. In this way, apps utilizing PlusUltra can seamlessly send and receive funds even if all of the PlusUltra server infrastructure were to be down, a feature we deem very necessary.

Data Model

The PlusUltra data model contains 2 types of data:

- Login data
- Synced data

The login data allows users to access their account data. It consists of encrypted keys to the account, as well as the hashed credentials, such as usernames and passwords, needed to access those keys.

The PlusUltra auth server maintains a copy of the login data and provides it to authenticated users upon request (subject to rate-limiting, 2-factor authentication, and other safeguards).

Once users gain access to their login data, they can decrypt the keys to their account data:

The account and wallet data are both located on the PlusUltra sync servers, which are general-purpose data stores.

The account store typically holds user settings, a list of wallets, and the private keys for those wallets.

The wallet store typically holds transaction and address metadata.

Applications using the PlusUltra SDK can write any data they like to these data stores.

Accessing either type of data store requires two keys:

- syncKey - 160-bit shared secret used to authenticate with the sync server
- dataKey - 256-bit AES-CBC encryption key for the data store contents

The dataKey is never revealed to the sync server, so PlusUltra cannot read the contents of the synced data.

PlusUltra Login

The PlusUltra platform provides users with a Single-Sign-On experience allowing them to use a single PlusUltra account (username/password) to secure the private keys and data of multiple applications that use the PlusUltra SDK.

This alleviates managing multiple private keys for each blockchain app.
Users are able to “login” to each partner application by simply scanning a barcode on the partner app using their PlusUltra mobile app.

Every PlusUltra partner app has an “appId”. When a user logs into a partner application, the PlusUltra mobile app client code creates a new bundle of login data associated with that particular appId.
Each bundle has its own loginKey, which is encrypted with the master PlusUltra loginKey. Thus, gaining access to the root PlusUltra loginKey provides access to all application-specific loginKeys.
However, gaining access to a single child loginKey does not provide access to any other child loginKey or to the root PlusUltra loginKey.
This minimizes the damage a single compromised application can do.

Each child login also has its own pin2Key. This makes it possible to perform PIN login into a partner app without requiring access to the root PlusUltra login.
Only the root login has a password or recovery answers, though, since those provide access to the entire tree in any case.

To request a PlusUltra login, partner apps places its appId into a communications "lobby" on the PlusUltra auth server. The partner app then displays a barcode with the lobby's address. Assuming the user scans
the barcode and approves the request, the PlusUltra app simply returns the application-specific loginKey and login data and through the lobby. At that point, the partner app is logged in.

The lobby is protected with ECDH cryptography, so nobody but the communicating parties can read the login reply, including the auth server itself.

Lobby Cryptography

To create a lobby, the requesting app begins by generating a random secp256k1 keypair, which will be used to encrypt any lobby replies.
The lobby's id is derived from the keypair:

lobbyId = SHA256(SHA256(publicKey.x))

The lobbyId can be truncated to any length, but current implementations take the first 80 bits. This is long enough to be unique, and is strong enough to make tampering with the public key impractical. The requesting application sends the lobbyId, public key, and request contents to the auth server. It then displays the lobbyId as a barcode.

The replying app scans the barcode and fetches the lobby contents from the auth server. It immediately verifies that the public key contained in the lobby still matches the lobbyId, preventing any man-in-the-middle attacks (the QR code is the root of trust in this security model).
Assuming the user approves the request, the replying application creates its own secp256k1 keypair and performs a standard ECDH multiplication to derive a shared point. It turns the shared point into an encryption key using the following KDF:

lobbyKey = HMAC-SHA256(0x00000001 | secretX, "dataKey")

The replying application encrypts its reply using this key, and uploads it to the lobby along with the matching public key.
The requesting app then retrieves the reply, performs the matching ECDH derivation, and decrypts the contents.
Since only the requesting app knows the original secret key, only the requesting app can decrypt the reply.
Once the requesting app is able to decrypt the reply, it has access to its store of private keys for that particular user.
It can then use those keys for transacting on the appropriate blockchain of that app.

PlusUltra Login is a fundamental paradigm shift in the way single-sign-on has been deployed in the past.
Centralized solutions from companies like Google, Twitter, and Facebook do not provide client-side security and fall short of the security needed for blockchain private keys.
PlusUltra Login provides the familiarity, ease, and functionality of single-sign-on, while still giving users full control of their data
and in the case of blockchain keys, control of their digital assets.

PlusUltra Beyond the Blockchain

While the PlusUltra SDK is focused on solutions for the blockchain ecosystem, it has tremendous applicability to conventional apps that need to provide users with secure, private data.
Any application that secures user data can leverage the PlusUltra SDK to drastically decrease their security risk at a fraction of the price of enterprise style security infrastructure.
App developers can enable a high powered vault in
the pockets of each of their end-users.

PlusUltra SDK can have great usage in the fields of:

●Internet of Things (smart locks, home security cameras, smart thermostats)
●Financial Applications (Quicken, Quickbooks, TurboTax)
●Healthcare (Practice management applications, healthcare records.)
●Personal Info (Contacts, calendars, credit card & bank info, passports, SSNs, etc)
●Authentication credentials (usernames, passwords, 2FA tokens)
●Identity (Public/private key authentication)
●Secure Messaging (PGP email, encrypted messaging)

PlusUltra Token

PUL Token it´s an ERC20 Token based on the Ethereum platform.
Total Max Supply has been set to 200,000(200K) PUL Tokens.
We wanted to create a low supply-high demand token instrument for PlusUltra users store their multiple assets value into a single Token.
We strongly believe that PUL token will consistently reach high value due to its low supply nature along the continuous development of the PlusUltra platform and its ecosystem.

Token Contract Address: 0xac8a812257fd80192de67f7ae27facfa76d5cc9f
Read the Contract HERE
Read the Contract code HERE

PlusUltra Prepaid Debit Card

PlusUltra Fin S.A. it's the registered company in Uruguay that will operate the PlusUltra Debit card.
We are partnering with an International Private Bank (located in the Free Zone) to become our Card Issuer

The company is located inside Zonamerica Free Trade Zone.
A modern and internationally recognized Free Zone Law that makes Zonamerica a world-class location.
The Uruguayan Law No. 15,921, passed on December 17, 1987, establishes that free Trade Zones are areas isolated from the rest of the national territory, where economic activity is stimulated by a specific legislation.
In Uruguay, these special economic areas are granted with customs and tax exemptions, and they are also excluded from the jurisdiction of state monopolies.
This differentiates them from most free trade zones of the world, where only customs benefits are granted and tax exemptions are generally limited.