Universal Web Login

The Path to Better Client Experiences

1KOSMOS BlockID

Universal Web Login

The Path to Better Client Experiences

1KOSMOS BlockID

Introduction.

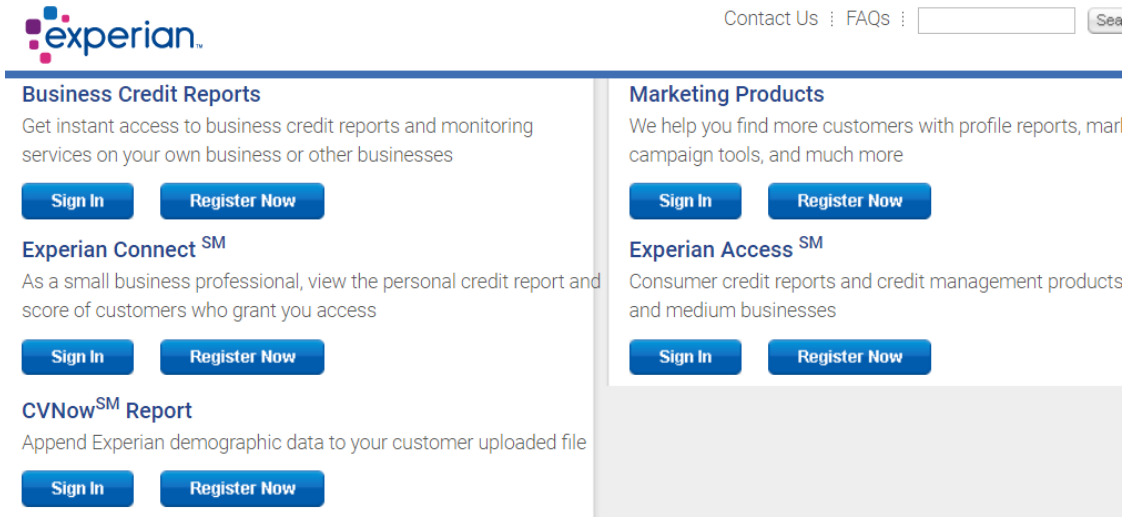
Many of today's customer-facing systems suffer from the same problems. They rely on centralized usernames, passwords, and cumbersome two-factor authentication mechanisms, and a user may have multiple identities spread across several client applications.

Legacy applications rely on built-in usernames and passwords. To mitigate this, 1Kosmos has developed **Universal Web Login (UWL)** that allows organizations to unify the customer login experience without re-architecting the target applications by leveraging several recent technologies:

- Decentralized identifiers (DIDs)
- Fast Identity Online (FIDO2) key-based login
- Consumer biometrics
- Trusted platform modules (TPMs)

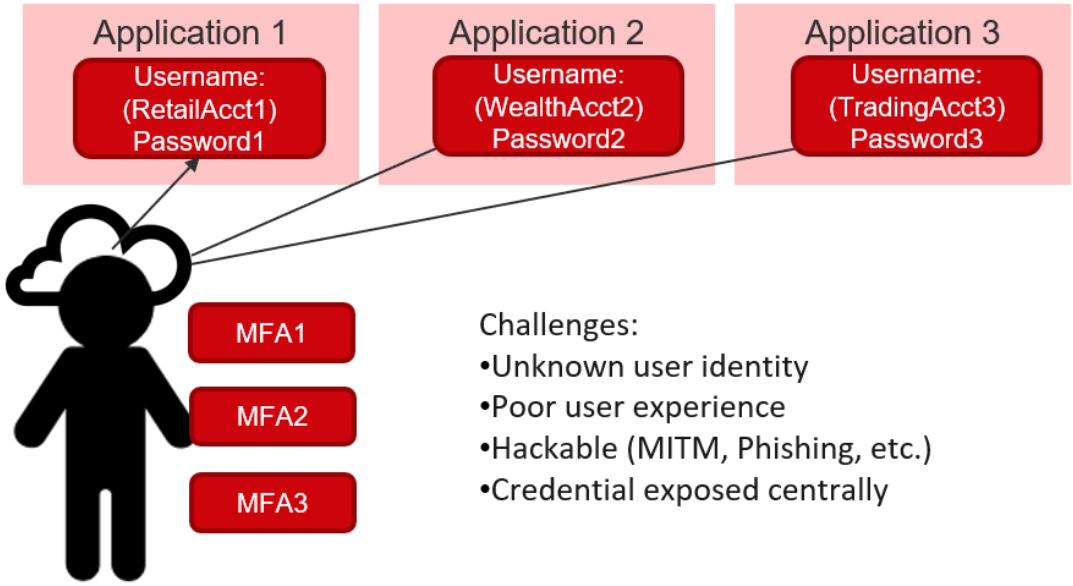
Organizations can move the credential into the user's possession. Doing this reduces the attack surface, simplifies the user experience, and minimizes friction between the user and helpdesk or account provisioning teams.

For example, at many financial institutions, it is not uncommon for a user to have three or more separate accounts into target systems.This is one example of a confusing and cumbersome customer experience:



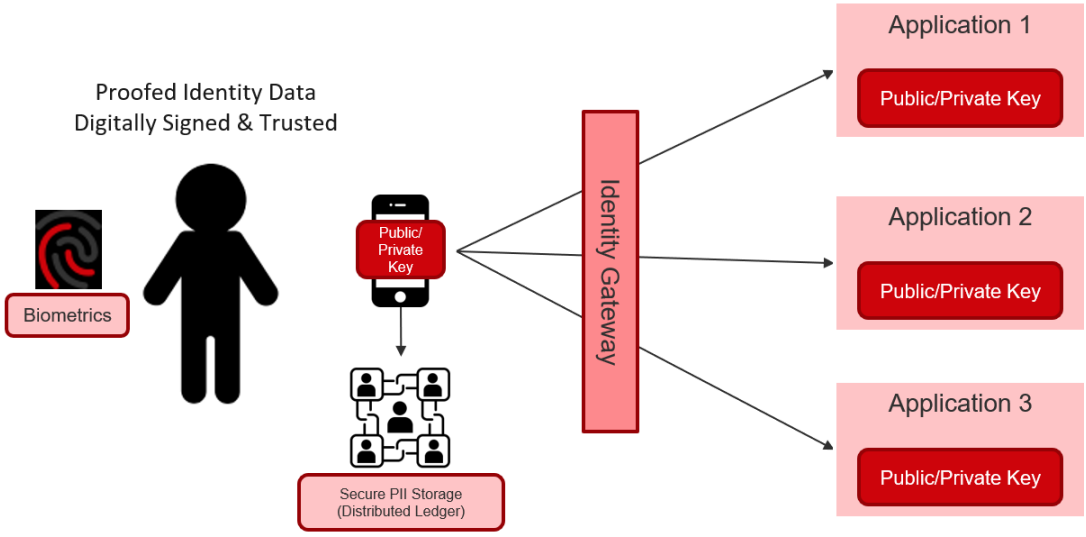
Fragmented usernames, passwords, and 2FA/MFA schemes lead to customer frustration and abandonment.

The following diagram illustrates traditional user authentication:



The problems with this are quite obvious but painful to solve with traditional IAM tools. Furthermore, to unify the logins from the back-end requires a rearchitecting of the application's credential and entitlement systems.

By utilizing the principles of decentralized identifiers and modern smartphone and computer security features, the credential problem can be solved by moving the identity into the user's hands. Instead of usernames and passwords, public and private keys are used in combination with biometric authentication. This diagram demonstrates the alignment of a user-controlled credential with target applications:



Despite tremendous usability and security benefits, there are minimal changes that need to be introduced to an IAM stack to unify any disparate logins:

- An app on a user's mobile device or computer (for storing of a private key)
- A public/private keypair on an application server
- An identity gateway to connect the user and the application

Furthermore, the "account pickup" or onboarding of the credential for existing users can be done in an automated and user-controlled fashion and with a phased approach to avoid "all or nothing" challenges.

Lastly, due to the nature of a cloud-first implementation and user-controlled private key, this solution does not require any DMZ components or inbound firewall rules.

User-Controlled Identity: The Credential Store + Biometrics

In the "real" world, an entity issues a credential (such as a passport, driver's license, or corporate badge) and gives it to an individual. Users then hold them and present them as they are needed.

Organizations can now adopt this same model. A key enabler for UWL is a Trusted Platform Module (TPM, or Secure Enclave/T2 for Apple devices). This allows a private key to be stored on the user's device (mobile phone or computer). When combined with Decentralized Identifiers or "DID" as defined by W3C, Microsoft, or the DIF group, the user's key can be used to present their credential over an encrypted channel and without reliance on a central password database. The user's key pair and DID are automatically created upon the first launch of the application. Because it is stored in the TPM, there is minimal risk of key compromise at the edge. A DID is a URL representation of a user's unique name in an organization.

In addition to a private key (representing **"something the user has"**), modern mobile phones and computers can also provide a strong biometric by utilizing a camera (for face), microphone (for voice), and built-in fingerprint/face readers such as TouchID and FaceID. This represents a second factor, or **"something the user is"**. For organizations that want to take authentication to an even higher level, you can enforce the use of a PIN in the application or on the smartphone to represent **"something the user knows."**

The BlockID user enrollment and key storage tools are available as a private-labeled application or as an SDK for embedding into existing applications.

Enabling Target Applications

Once a user has a unique identifier or DID, the next step in setting up UWL is to enable the applications. Universal Web Login consists of 2 small components:

- An SDK (or web server plug-in) that leverages elliptic curve cryptography and generates application-specific private/public keypairs based on the BIP39 standards. The SDK enables encryption to secure data in flight and a javascript library that generates an authentication request (via a unique QR code or a push notification) to prompt the user to present their credentials.
- Four lines of JavaScript code are placed on the client-facing web page (a very similar process to adding a login with Google or Facebook to any website).

Once the SDK and the four lines of javascript are integrated into the web application, the BlockID platform provides a secure communication channel between the mobile app and the application.

UWL enables the application to verify the authenticity of the presented credential against the decentralized ledger in real-time before granting access. The entire process for signing and encrypting the data, transferring the data, verifying the data is embedded within the SDK and is seamless to the user or the application.

Binding Users to the Applications

The final step in the journey to UWL is to link the customer's new DID to the user's existing application username.

There are several ways to do this, such as sending the user a URL or a QR code. These methods allow account linkage to be done by the user, avoiding IAM provisioning costs and risks. If the user's identity is already known (i.e., you know and trust their existing email address or phone number), a link can be sent to them that allows the binding of their DID to the application username. A URL or QR code is sent to the user to start the process.

The QR code does not contain any identifying information and is solely used to initiate a secure connection (similar to how an SSL handshake is initiated between parties). The QR code indicates to the user or the mobile app where to send the response to and what kind of information is requested, a process that's very similar to the requested scopes in a traditional OAuth or OIDC request.

The QR code itself contains a base-64 encoded request that when decoded, looks like the following authentication request:


```
{
"authtype":"Face|Voice|Fingerprint",
"Scopes":"firstname, lastname, age, university-degree, drivers-license",
 "ecdsaPublicKey":"VDsnXOBEBl55rY9BTdW3OyRDC7MYzqLGg11t69q/3FOZdrqHoM+LswZ3XyA==",
"url":"wss://id.1kosmos.net/api/v3/ws/default/did-login/030C725D6417A12FAB27FD801963BBF2"
}
```

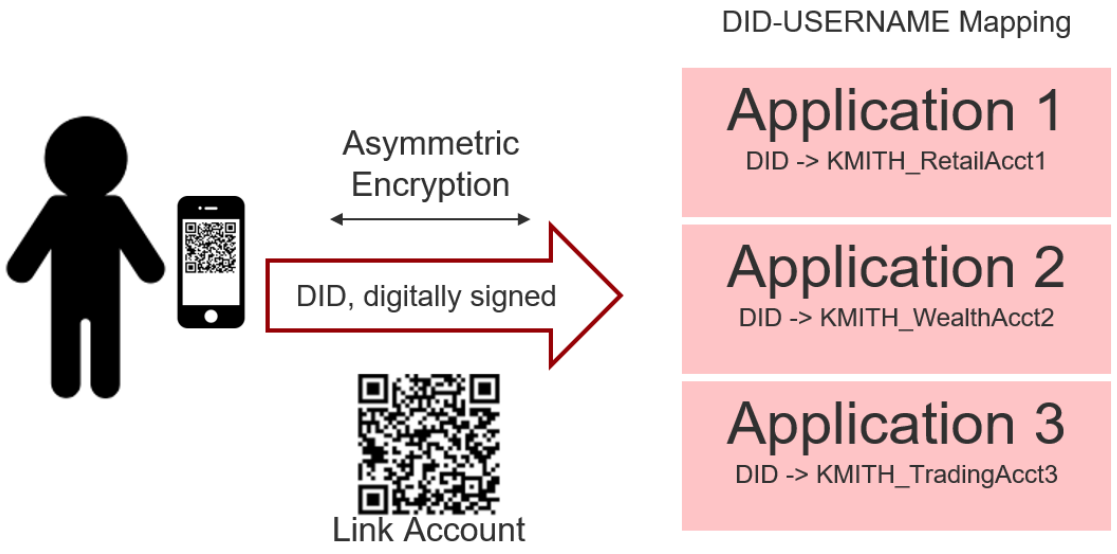
The authtype parameter indicates the consent algorithm that the mobile app would enforce prior to exchanging any data. (It could be a combination of one or many form factors to add biometric MFA capabilities to a secure biometric consent)

The scopes parameter lists the data elements that the application requires the user's mobile wallet. The data elements requested by the application could vary based on the sensitivity of the information the application needs to present to the user.

The ecdsaPublicKey is the application's public key that the mobile app uses to encrypt the data before sending it to the application, enabling the target application to decrypt the payload.

The url enables the mobile app to identify the specific endpoint that the data needs to be sent to. This url is similar to typing any URL in a web browser for example https://accounts.google.com/login.

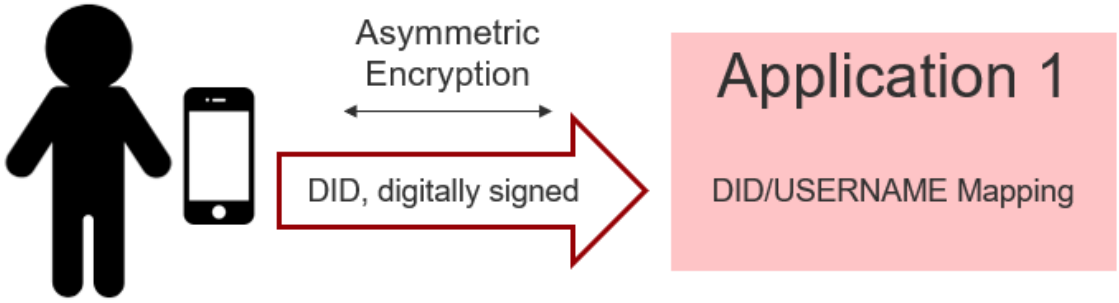
None of the information contained in the QR code is private information and is not susceptible to QRLjacking.



When clicked or scanned, the app is launched, and the user's public key is sent to the application.

This starts the secure handshake and gives the application the user's DID, which is then linked to the application username.

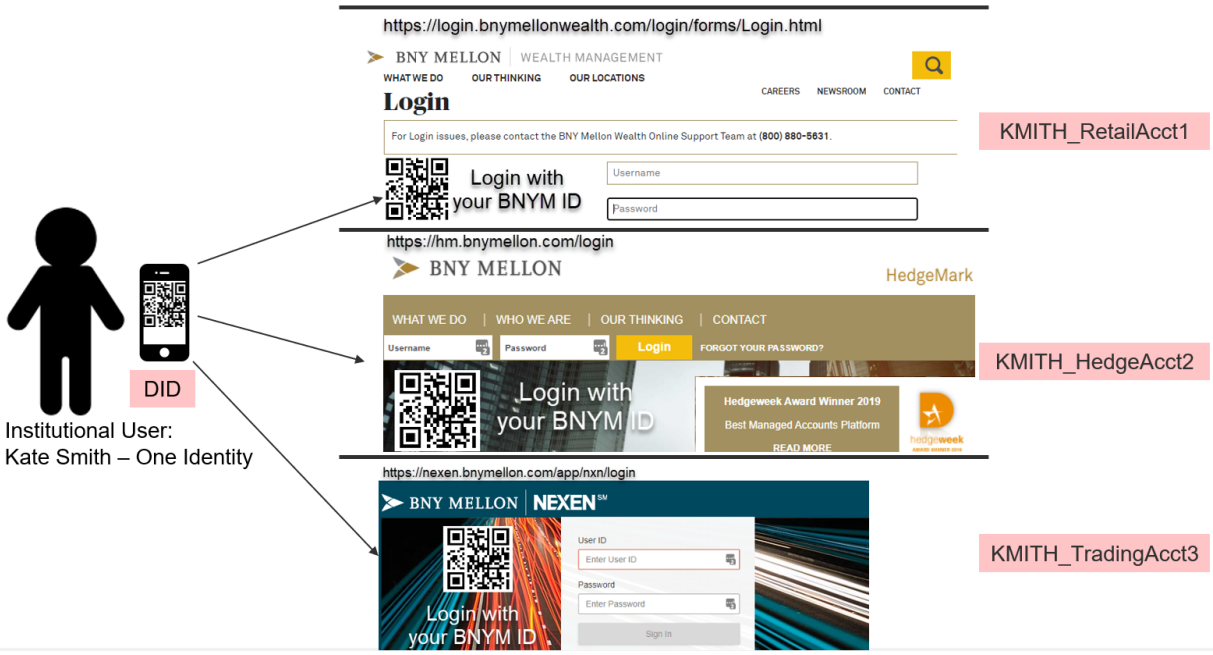
Alternatively, an existing username, password (and MFA if applicable) can be used to establish the linkage, and going forward, the user's DID will be presented to the application.



Once the DID is transmitted to the application, it will represent the unique identifier for future logins. The UWL library will translate the DID into the legacy username, and the application can continue to function as it always has.

Authentication with the System

When authenticating to systems, the authentication method will be a QR code scan or a push notification. Usernames, passwords, and one-time tokens are no longer required.



Standards

The BlockID platform and its implementation of Universal Web Login have been built on the following standards, ensuring interoperability for customers and avoiding “vendor lock-in” for identity enrollment, authentication, or credentials verification:

- BIP39 - Key generation and recovery, along with RFC-6979
- W3C Decentralized Identifiers for identity portability
- FIDO - Fast Identity Online passwordless standards
- ECDSA - Algorithms for digital signatures

Security

BlockID protects user data stored in the users mobile device, the user data stored in its blockchain file system (IPFS) and the data that is transmitted between different blockchain nodes within the BlockID ecosystem. The following is a list of cryptographic algorithms used within the BlockID ecosystem:

- Advanced Encryption Standard (AES) is a symmetric block cipher used for information protection for data in transit and data at rest.

- Elliptic Curve Diffie-Hellman (ECDH) Key Exchange is an asymmetric algorithm used for digital signatures

- Elliptic Curve Digital Signature Algorithm (ECDSA) is an asymmetric algorithm used for digital signatures

The system is resilient to user-credential hacking due to its use of a private key protected with biometric authentication. Applications are protected from spoofing by being registered with the organization. This process is similar to a matching of a DNS entry and an SSL/TLS certificate.

For more detailed information on platform security, please refer to a separate document titled “BlockID Encryption Schemes”.

1KOSMOS BlockID

To continue the conversation, do not hesitate to contact Mike Engle, CSO with 1Kosmos, directly via email: **mike@1kosmos.com**