Web5: The Decentralized Web Platform

A new evolution of the Web that enables decentralized apps and protocols.
In the current Web model, people are users who do not own their data or identity. They are given accounts by companies and their data is held captive in app silos.

To create a new class of decentralized apps and protocols (e.g. tbDEX) that put individuals at the center, we must empower them with self-owned identity and restore control over their data.
The Decentralized Web Platform (DWP) enables developers to write Decentralized Web Apps (DWAs) using Decentralized Identifiers (DIDs) and Decentralized Web Nodes (DWNs), returning ownership and control over identity and data to individuals <-- We are calling this Web5
The Pillars of Web5

Decentralized Identifiers

Self-owned identifiers that enable decentralized identity authentication and routing.

Verifiable Credentials

Data formats and models for cryptographic presentation and verification of claims.

Decentralized Web Nodes

Data storage and message relay nodes that serve as the foundation for decentralized apps and protocols.
**Decentralized Identifiers (DIDs)**

- Identifiers are self-generated and self-owned
- No centralized providers or trusted authorities
- No special utility tokens or subjective consensus
- Highly resistant to all forms of interdiction
- IDs can be made universally discoverable

Bob can discover all DIDs indexed in the network to independently resolve and verify them.

Alice self-generates her DIDs and anchors PKI (key and endpoint) state changes in the network.

```
did: ion: 1a2b3c4d...

\{\ldots\} + \{\ldots\} = \{\ldots\}
```

\[ T_0 \quad T_N \quad \text{Compiled} \]
Verifiable Credentials (VC)

**Issuance of a Verifiable Credential**

- Acme Bank IDV Issuer
  - Anchor Issuer’s DID, public keys, and endpoints into ION

- Acme Bank DID: did:example:bank
- Subject: did:example:alice
- Claims:
  - acct: 12345678
  - name: Alice Smith
  - issuer’s digital signature

**Alice’s Wallet**

- Resolve public key material

**Present Bank Account Proof**

- Present Bank DID: did:example:hardtron
- Verification Method: { JWK }
- Issuer’s digital signature
- Subject’s digital signature

**PFI Verifier**

- tbd
Decentralized Web Nodes (DWN, DWeb Nodes) is an emerging standard for data storage and relay that enables entities of any type (people, organizations, etc.) to send and store encrypted or public messages and data, enabling a wide variety of decentralized apps and protocols to be built on top.

**Features**

- **Universally Addressable**
  - Crawlable DID-relative addressing of data

- **Replicated**
  - Masterless eventually consistent replication of instances across devices and clouds

- **Secure**
  - Data can be optionally encrypted with an individual's DID keys

- **Semantic Discovery**
  - Discover any form of published data simply by knowing its semantic type

- **Async Message Threads**
  - Send and receive messages over a DID-encrypted universal network

- **Supports any identity type**
  - Designed to support individuals, companies, machines, or any other entity
The Anatomy of an Identity Wallet

**Data Management**
Provide UI and functionality to manage credentials and app data stored in DWNs

**Credential Functions**
Sign, verify, discover, and present credentials to verifying parties

**DID Auth**
Perform authentication and manage authorizations (e.g. DWN authz capabilities)

**DID Functions**
Support create, update, and recovery of DIDs across all supported DID Methods

**Context Management**
Maintain and enforce which DIDs are used with different people, apps, and services
The combination of Decentralized Identifiers and Decentralized Web Nodes produces a Web of DID-secured messaging, data sharing, and credential exchange that can replace one-off protocols (encrypted messaging, photo sharing, etc.) with universal standards for all types of semantic data exchange.
What are PWAs?

PWA stands for Progressive Web App, a standard for installable web apps that is implemented in all major browsers today. PWAs are websites that took their vitamins and have special powers.
From PWAs to DWAs

**PWA**

- Web App
- Service Worker
- Local Cache
- Centralized App Server

**DWA**

- Web App
- DWN SDK
- Service Worker
- Local Cache
- Decentralized Web Node
Many activities in our world require the establishment of trust between participants. DIDs + DWeb Nodes allow individuals, organizations, and companies to publish credentials anyone can discover and independently verify.
When viewed from 10,000 feet, tbDEX is a DWN-based threaded messaging and data exchange protocol layer that runs atop the substrate of DIDs and Decentralized Web Nodes.

Alice’s tbDEX-aware app knows the DIDs of various PFIs and sends them Ask messages to initiate an exchange.

1. Alice’s remote DWN
2. Alice’s local DWN

PFIs respond to Alice’s Asks via Bid messages sent back to Alice’s DWeb Node.

navigator.did.send({
  schema: "tbd.website/schemas/ask"
})

navigator.did.send({
  schema: "tbd.website/schemas/bid"
})
It’s not fun to regurgitate your playlists over and over again for different music apps, so let’s stop doing that.

Alice’s remote DWN

1. Groove secures the ability to write `schema.org/MusicPlaylist` objects to Alice’s DWNs, and adds a new entry.

   navigator.did.store({
   schema: "schema.org/MusicPlaylist",
   data: { ... }
   })

2. TIDAL, which previously secured the ability to read Alice’s `schema.org/MusicPlaylist` entries, can read the entry Groove added.

   navigator.did.request({
   schema: "schema.org/MusicPlaylist",
   data: { ... }
   })
Your preferences, tickets, reservations, and other travel data are strewn across 100s of different hotel, airline, and travel apps in a massive, unworkable mess. DIDs + Decentralized Web Nodes can help unify these flows and experiences.

1

Alice grants her hotel, airline, and rental car provider the ability to add `schema.org/Reservation` objects to her collection of trip-related data.

```javascript
navigator.did.store({
  schema: "schema.org/Reservation",
  data: { ... }
})
```

2

Alice can grant any app she chooses access to reservations and tickets stored in her `schema.org/Trip` data to help her visualize her itinerary.

```javascript
navigator.did.request({
  schema: "schema.org/Trip"
})
```
Web5 is a Decentralized Web Platform that enables developers to leverage Decentralized Identifiers, Verifiable Credentials, and Decentralized Web Nodes to write Decentralized Web Apps, returning ownership and control over identity and data to individuals.