International FAIR Digital Objects Implementation Summit 2024

Extended Abstracts

https://doi.org/10.52825/ocp.v5i.1054

Published: 18 Mar. 2025

An Overview of Decentralized Web Technologies as a Foundation for Future IPFS-Centric FDOs

Andrei Vukolov[1,*], Erik Van Winkle[2,3], Erik Schultes[4], Line C Pouchard[5], Sina Iman[2,3], Philipp Koellinger[2,3], and Christopher Hill[2,3]

[1]Elettra Sincrotrone Trieste, Italy

[2]DeSci Labs, Switzerland

[3]DeSci Foundation, Switzerland

[4]GO FAIR Foundation, Netherlands

[5]Brookhaven National Labs, US

*Correspondence: Andrei Vukolov, andrei.vukolov@gmail.com

Abstract. dPIDs are an emerging PID technology based on decentralized architectures and self-sovereign identity [1]. dPIDs are PID containers, forming persistent storage systems where each object is identified by a unique PID. dPIDs are immune to content drift and resolves deterministically their mapped content, providing a reproducible binding between the (meta)data and identifier. As dPIDs take a decentralized net-work protocol approach to PIDs, their implementation of FDOF recommendations may require further explanation [2]. This presentation is a primer on the decentralized technologies behind dPID and their associated benefits, including a discussion of their potential usefulness for FDOs. dPIDs can form the fabric for a persistent, interoperable FDOs landscape.

Data replication via the underlying content-addressed peer-to-peer network facilitates the implementation of FDO-G2 [3], ensuring long-term persistence and mitigating the risk of data loss via implicit data replication and storage redundancy between network participants. Content addressing gives dPID the property of deterministic and verifiable resolution, exceeding the requirements of FDO-PIDR2. A subsequent benefit of this open protocol-based approach is that dPIDs prevent the formation of vendor-lock-in and data silos, facilitating FDO-PIDR1 and FDO-G1. The provenance of data and updates to dPIDs are registered by digital signatures based on W3C de-centralized identifiers (DID), facilitating FDO-PIDR6. Data sovereignty is facilitated using a Directed Acyclic Graph (DAG) approach compliant with FDO-GR4, FDO-GR5 and FDO-GR6. DAGs also allow for granular machine actionability in compliance with FDO-GR1 and FDO-GR11. As PIDs are logged on Blockchain, tomb-stones for dPIDs are inherently permanent in line with FDO-GR12.

Keywords: dPIDs, Decentralized Architectures, Persistent Identifier, Deterministic Resolution, W3C Decentralized Identifiers, Content-Based Addressing, Data Sovereignty, Directed Acyclic Graph (DAG), Blockchain Logging, FAIR Digital Objects

1. dPID's Underlying Technology

The main technologies underpinning dPID are: Content-addressed storage net-works, DAG-based linked data, the Sidetree Protocol, Decentralized Identifiers (DIDs), and blockchain. All

the algorithms and associated documentation are open, and publicly accessible, providing re-usability (FDO-PIDR1) [4].

dPIDs are built on a content-addressable storage network called the Inter-Planetary File System (IPFS) [5]. In IPFS, and content-addressed networks in general, payloads are identified by a hash computed from their content. This hash is called a Content Identifier (CID). The collection of all possible hashes used to locate the stored data blocks forms a global secure and decentralized namespace called the distributed hash table (DHT). An end user enters the CID as a key to specify the data that should be retrieved, then the DHT responds with the addresses of the nodes storing the data. The client then connects directly to one of these nodes to retrieve the data. Through CIDs and the DHT, IPFS completely eliminates content drift and significantly mitigates the risks of link-rot and vendor lock-in by providing built-in data replication capabilities between repositories (G1, G2). An additional benefit, the resolution of dPIDs based on CID resolution is deterministic (FDO-PIDR2).

The dPID technology can be viewed as a containerized file structure, using (Interplanetary Linked Data) IPLD to create DAGs, mapping CIDs with (meta)data in a fashion similar to JSON-LD, where URIs are replaced by CIDs (FDO-GR4, FDO-GR5 and FDO-GR6) [6]. dPIDs are currently constructed for the purpose of identifying research objects [7]. A simplified version of a dPID can be seen in Figure 1 [8].

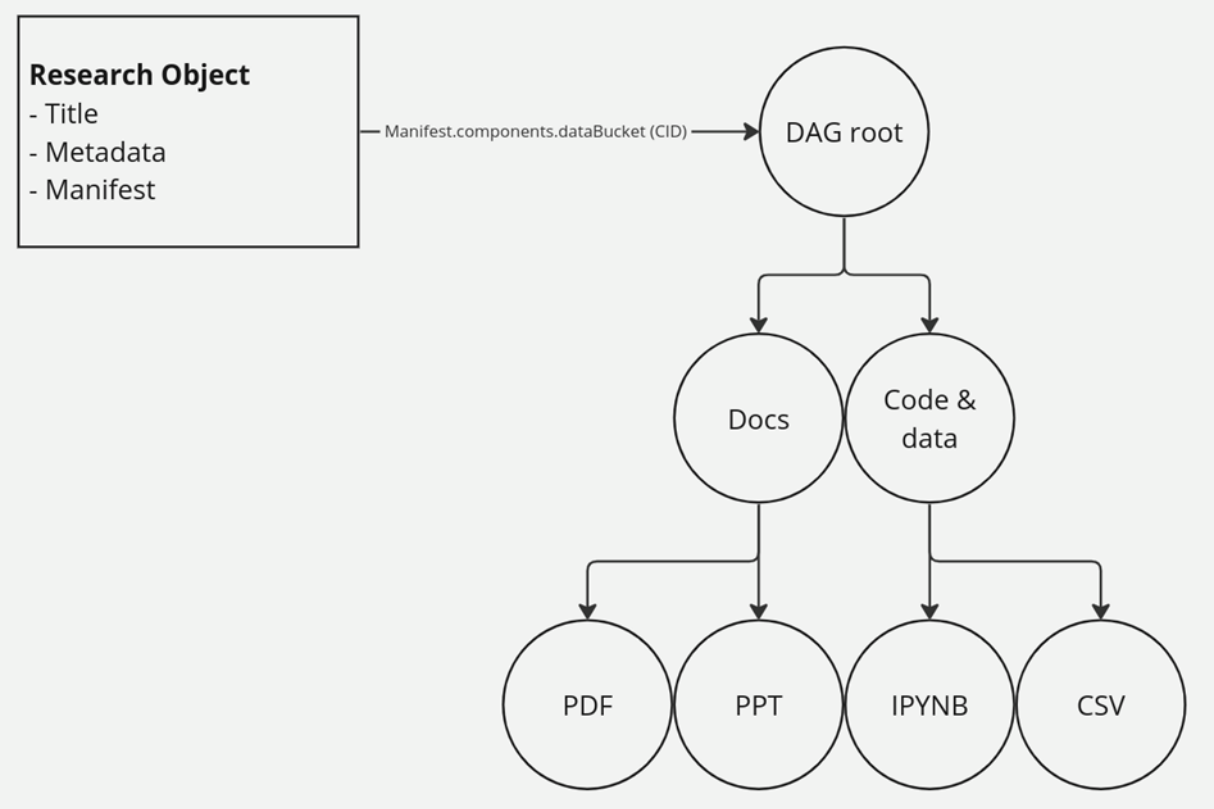

Figure 1. *A research object entity and a visualization of its link to the data DAG*

Decentralized Identity Foundation's (DIF) Sidetree is a higher order protocol which allows nodes in a network to collaborate and keep track of sequential updates to CID-based data objects made by end users, in a trustless fashion (FDO-PIDR6) [9]. Decentralized Identifiers are used to sign these data object commits and gate update rights on the sequential streams. The dPID protocol is built on top of a Side-tree implementation (Ceramic) and associated composable database (ComposeDB), which enables distributed indexing and querying capabilities. While Sidetree on its own doesn't provide a way to index, discover, organize, and query for existing streams (nodes) depending on which schema (entity) they implement, or track references made to other streams, ComposeDB is a type of graph database built on SideTree to achieve all of these properties. An example of a graph of independent authors, their publications, and the relations between the different types of entities can be seen in Figure 2, explaining how dPIDs can cooperate with other PIDs and DIDs in the space (ORCID, DOI, etc).

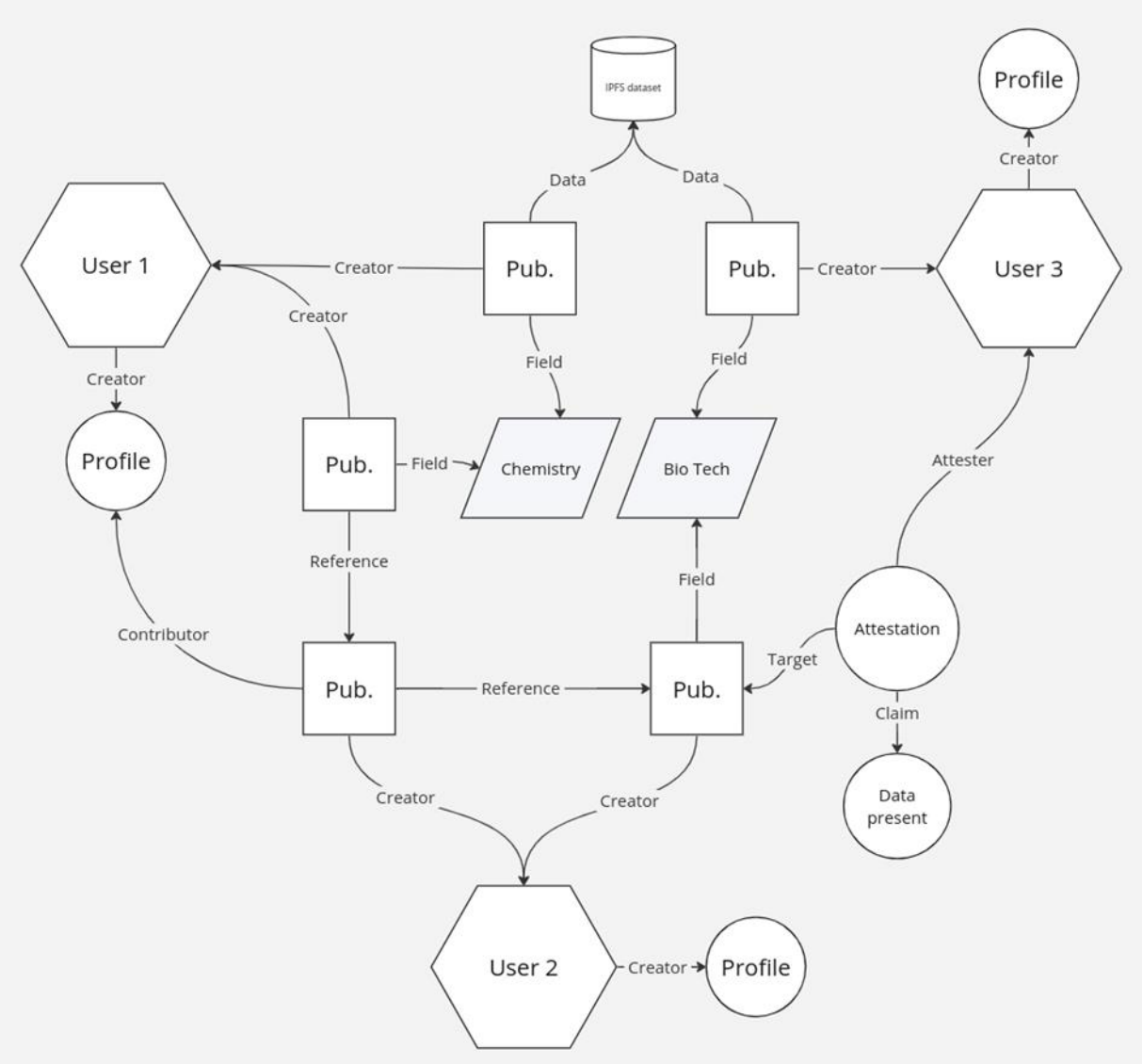

Figure 2. *An example of a graph of independent authors, their publications, and the relations between the different types of entities.*

Finally, blockchain is used as the Anchoring point for Sidetree once nodes in the network have reached consensus on the global state (FDO-GR12). Furthermore, by registering the mutable root CID-based identifiers on a blockchain smart contract registry, dPIDs achieve strong consistency (as opposed to Sidetree's eventual consistency) and enable a namespace that square Zooko's triangle [10], i.e. decentralized, secure, and human readable. The latter property is leveraged to create compatibility with PID schemas which are not hash-based, such

as providing a root identifier conformant to the existing PID schemas (.e.g, DOI, ARK, RAID). In this sense, blockchain-based registries provide aliases enabling cross-compatibility between existing PID namespaces and the dPID technology.

2. Benefits of dPIDs

2.1 dPID Resolution Anatomy

Content-addressed storage (CAS) allow for natively deterministic resolution of CIDs to their mapped content. As such, inheriting the deterministic resolution properties of CAS networks is a design goal for dPIDs (REF). dPIDs can be construed as persistent file systems linking digital objects which allow for granular referencing and resolution to their mapped content (FDO-GR1, FDO-GR11). As such, dPIDs act as “PID containers". The current resolution anatomy of a dPID is as follow in Table 1 [8]:

Table 1. *URL Anatomy of dPIDs*

URL Anatomy Component	Component Placement	Example
Registry	1	beta
Version Identifier	2	v2
Content	3	dataset.pdf
Component Suffix	4	?jsonld

2.2 Deterministic Resolution

To enable persistent resolution of nodes by address, we employ an algorithm for deterministic graph traversal. This deterministic traversal simply requires the root dPID, exposing the IPLD DAGs and their associated CIDs. Currently planned traversal algorithms for deterministic resolution include the options listed in Table 2 [8].

Table 2. *Traversal Algorithms for deterministic resolution of dPIDs in ComposeDB*

Resolution Case	Overview	Traversal Algorithm
Root node	Addressing the latest state of some node N:	Query network for status of node N
Particular version	Addressing a particular commit C of a node N:	Query network for status of node N at commit C
Particular time	Addressing the state of a node N as of time T:	1. Query network for update history of node N 2. Find the newest commit C that was anchored before or at time T Resolve node N at commit C
Particular version index	Addressing a particular version k of a node N:	1. Query network for update history of node N 2. Select commit C at index k in update history Resolve node N at commit C

Outgoing edge	Addressing of an outgoing edge from a node N made against some other node:	1. Resolve N 2. Get value R from reference field Resolve node R
Versioned outgoing edge	Addressing a versioned outgoing edge from a node N made against some other node:	1. Query network for status of node N 2. Get value of reference field R and version field C Resolve node R at commit C
Incoming edges	Addressing of an incoming edge to node N, from some node N2 of entity type T:	1. Query network for all nodes of type T 2. Find node N2 with a. Reference field set to N1 Resolve N2
Versioned incoming edges	Addressing of an incoming edge to node N as of version C, from some node N2 of entity type T:	1. Query network for all nodes of type T 2. Query network for update history U of node N 3. Find versions V of node N2 with a. Reference field set to N b. Version field value in U before version C Resolve N2, considering V the update history while targeting N at C
Versioned data DAG paths	Addressing a DAG node through UnixFS path P, in research object N as of version C:	1. Resolve N at version C 2. Get value DAG in manifest CID field 3. While P not empty a. Pop first segment S from P b. Set DAG' = lookup(S, DAG) c. Loop with DAG' as DAG DAG is now the addressed node or leaf

2.3 Strong Consistency and High Throughput

An inherent trade-off exists between achieving strongly consistent PIDs which satisfy the three desirable properties of a namespace and enabling high throughput. dPIDs solve this tension by anchoring the version-invariant identifier on a block-chain-based registry and leaving all subsequent updates of the mapped DAGs to SideTree's consensus mechanism. For applica-

tions that do not require backward compatibility with existing PID namespaces and/or conveniently short URLs for hu-mans (as opposed to long CIDs), Sidetree's eventual consistency can be sufficient [9].

3. Considerations and Conclusion

We present a novel PID technology based on content-address networks, decentral-ized identifiers, and the Sidetree protocol. While dPIDs natively satisfy many of the FDO Forum's FDO Requirement Specifications. dPID FDOs will come with different implementation specifications, just as DOIP FDOs and Linked Data FDOs have implementation differences [2].

dPIDs preserve many characteristics of existing systems such as the need for sustainable and certified repositories as custodians of data. By laying a foundation of mutual understanding around the dPID technology, we aim to facilitate conversations around the possibilities offered by deterministic resolution for the emerging world of FDOs and the role of open, content-addressed networks in preserving scientific artifacts.

Data availability statement

No data was used in the creation of this proposal.

Underlying and related material

- The canonical research object encompassing this proposal and all subsequent artifacts can be found at the Persistent Identifier https://beta.dpid.org/152.
- The original pre-print submission is at https://beta.dpid.org/152/v2/root/FDOF2024 - An Overview of Decentralized Web Technologies as a Foundation for Future Consumption of IPFS-centric PID Profiles.pdf
- Reviewer commentary can be found at https://beta.dpid.org/152/v3/root/ReviewerCommentary.txt
- The revised manuscript preprint can be found at https://beta.dpid.org/152/v3/root/FDOF2024 - An Overview of Decentralized Web Technologies as a Foundation for Future Consumption of IPFS-centric PID Profiles.pdf
- Detailed outlines of FDO Forum requirement specifications can be found at https://beta.dpid.org/152/v3/root/dPID Draft - FDO Forum FDO Requirement Specifications.xlsx
- Upon publication, the final formatted manuscript will be posted by the authors at https://beta.dpid.org/152/v4/root/Published Manuscript - FDOF2024 - An Overview of Decentralized Web Technologies as a Foundation for Future Consumption of IPFS-centric PID Profiles.pdf

Author contributions

Andrei Vukolov: Conceptualization, Writing - Original Draft Preparation.

Erik Van Winkle: Conceptualization, Writing - Original Draft Preparation.

Erik Schultes: Writing - Review & Editing.

Chris Hill: Validation, Writing - Review & Editing.

Line C. Pouchard: Supervision.

Philipp Koellinger: Supervision.

Sina Iman: Validation, Visualization, Software.

Competing interests

The authors of this paper have no competing interests to declare.

Funding

No funding was given for the creation of this proposal.

Acknowledgement

Submitted on Behalf of the dPID Working Group.

References

[1] D Hook, B Kramer, P Koellinger, N Quaderi (2023). “Innovation, Technology and Infrastructure - APE Conference 2023.” figshare. Presentation. https://doi.org/10.6084/m9.figshare.21953864.v1
[2] S Soiland-Reyes, C Goble, P Groth (2023). “Evaluating FAIR Digital Object and Linked Data as Distributed Object Systems.” arXiv:2306.07436v2 [cs.DC]. https://doi.org/10.48550/arXiv.2306.07436
[3] A Ivonne, B Christophe, B Daan, et al. (2023). FDO Forum FDO Requirement Specifications. https://doi.org/10.5281/zenodo.7782262
[4] Desci Labs. (2024). “nodes.” GitHub. https://github.com/desci-labs/nodes (accessed January 12, 2024).
[5] J Benet (2014). “IPFS - Content Addressed, Versioned, P2P File System.” arXiv:1407.3561v1 [cs.NI]. https://doi.org/10.48550/arXiv.1407.3561
[6] IPLD. (2023). “Docs.” IPLD Documentation. https://ipld.io/docs/ (accessed 13 December, 2023).
[7] S Soiland-Reyes, P Sefton, M Crosas, et al. (2022). “Packaging research artefacts with RO-Crate.” Data Science, 5(2). https://doi.org/10.3233/DS-210053
[8] DeSci Labs & DeSci Foundation. (2023). “DeSci Codex: The Collaborative Data Exchange.” https://codex.desci.com/ (accessed 13 December 2023).
[9] D Buchner, O Steele, T Ronda (Editors). (2023). “Sidetree Protocol Specification.” Decentralized Identity Foundation. https://identity.foundation/sidetree/spec/ (accessed 13 December 2023).
[10] A Swartz (2011). “Squaring the Triangle: Secure, Decentralized, Human-Readable Names.” http://www.aaronsw.com/weblog/squarezooko (accessed 13 December 2023).