



perm.pub/dsi:0123456789abcdefghijklmnopq/1

Additional formats and editions available online.

Author date: 2024-04-03

DeSci dPID vs Document Succession Identifier

E. Castedo Ellerman  (castedo@castedo.com)

Citation:

E. Castedo Ellerman (2024) "DeSci dPID vs Document Succession Identifier" perm.pub <https://perm.pub/dsi:0123456789abcdefghijklmnopq/1>

Copyright:

creativecommons.org/licenses/by/4.0/
2024 © The Authors. This document is distributed under a Creative Commons Attribution 4.0 International license.

Abstract

[Document Succession Identifiers](#) (DSIs) and the infrastructure developed by [DeSci Labs](#) both implement [Persistent Identifiers](#) (PIDs). This document summarizes the basic similarities and differences between the two.

Terminology

DeSci vs DSI

DeSci Concept	Similar DSI Concept	Notes
dPID	base DSI	dPID is human-friendly; DSI is only URL-friendly
dPID contract	N/A	alias registry
node (a version of)	Baseprint document snapshot	static version of a research output
node (all versions of)	Baseprint document succession	series of versions of a research output

DeSci vs DSGL

Document Succession Git Layout (DSGL) is a specific format for storing a document succession. Currently, only document successions stored in DSGL are identified by DSIs.

DeSci Concept	Similar DSGL Concept	Notes
IPFS directory	Git tree	application of hash tree
IPFS CID	SWHID & Git hash	low-level identifier of a static version of research output
Ceramic stream	Git history in DSGL	update mechanism of research output
Ceramic stream root hash	Git initial commit hash	low-level identifier of all versions of research output
Decentralized Identifier (DID)	SSH signing key	digital identity updating research output

dPID & root hash vs base DSI

A base DSI is a DSI without an edition number. For example:

| VGajCjaNP1Ugz58Khn1JWOEdMZ8

A dPID can be drastically shorter and more human-friendly, for example:

| 46

A dPID is assigned to a root hash, which is primarily for internal use and usually not seen by a user. A base DSI is a hash but is shorter than a DeSci node's root hash.

A dPID and root hash are designed to fill different needs with different trade-offs. A DSI is a compromise designed to only partially satisfy the objectives of a dPID and a root hash.

DeSci Node vs Baseprint document succession

Both DeSci nodes and Baseprint document successions track versions of digitally encoded snapshots of a research object. A Baseprint document is a narrow type of research object.

Versioning in DeSci Nodes is done with positive integers, whereas a document succession can use multi-level edition numbers, similar to semantic versioning of software packages.

DeSci Alias Registry

A dPID is an alias to the root hash of a DeSci node. dPIDs can be very short and human-friendly because they rely on a decentralized alias registry, which makes a dPID an [extrinsic identifier](#). In contrast, a DSI is an intrinsic identifier, which does not require a registry at the cost of being longer and less human-friendly than a dPID.

DeSci research object vs Baseprint document snapshot

As of 2023, DSI are only used for a very narrow type of research object, specifically static research documents (that are JATS XML based) stored in [Document Succession Git Layout](#) (DSGL). Like JATS XML packages as found in PubMed Central, these research documents are not intended to exceed tens of megabytes and are typically much smaller.

DeSci Nodes are designed for a much more general type of research object that brings manuscripts, data, and code together in one place. DeSci Nodes are designed to handle vastly larger research objects, even ones that are hundreds of gigabytes.

IPFS directory vs Git tree

A DeSci research object is digitally encoded in an IPFS directory, whereas a Baseprint document snapshot is encoded in a Git tree.

IPFS CID vs SWHID & Git hash

An IPFS directory (or fixed version of) is identified by a CID (content identifier), whereas a Git tree is identified by a Git hash or SWHID (Software Hash Identifier).

A CID has a self-describing format, whereas a Git hash does not.

Ceramic stream vs Git history in DSGL

A Git history in DSGL ([Document Succession Git Layout](#)) serves a role to DSI similar to the role [Ceramic](#) streams serve for a dPID.

A Git history in DSGL is stored in Git repositories or git-compatible archives such as the Software Heritage Archive, whereas Ceramic streams of DeSci Nodes use IPFS and the Ethereum blockchain.

DID vs SSH signing key

Both Git histories in DSGL and Ceramic streams require a mechanism such that only certain entities (such as authors) can update the distributed research objects.

DeSci Nodes uses DIDs (Decentralized Identifiers), whereas DSGL uses SSH signing keys for this function.