



O P S C I E N T I A

INCF Google Summer Of Code 2021 - Proposal

**Securing BIDS Neuroimaging Datasets Stored on
Decentralized File Sharing Networks with Self-Sovereign
DIDs and IPFS**

Achintya Kumar

achintya.191it203@nitk.edu.in

National institute of Technology Surathkal, India

Location : Patna, Bihar, India

[Linkedin profile](#)

SkypeID: live:cid.fba4f86a7dee76a8

SYNOPSIS

Overview

This Google Summer of Code project's primary goal is to contribute to the development of an open-source application for GDPR-compliant self-sovereign scientific data management and peer-to-peer sharing. My contributions will leverage the Interplanetary File System (IPFS) for content-addressable peer-to-peer data storage, the IDX API to manage decentralized identifier (DID) indices and Ceramic protocol for reading and writing mutable records to IPFS.

Problem Statement

The field of neuroimaging has approached a critical mass of high spatiotemporal resolution datasets siloed in individual labs - often in proprietary formats and inaccessible to the public. The Brain Imaging Dataset Structure (BIDS) specification was recently introduced to address dataset interoperability concerns. However, there are still outstanding problems regarding

- Accessibility
- Provenance
- Regulatory compliance of open datasets

The proposed application will solve these problems by making content-addressable encrypted datasets persistently available on IPFS and utilising IDX permissions management to grant research participants and researchers real-time opt-in/out privileges.

Working

The application accepts new or existing BIDS datasets and assigns DIDs to consenting participants and the researcher, curating and uploading the data.

Participants can see a dashboard of their data and manage access by third parties. Research investigators can only see the datasets with consenting participants and deploy encrypted datasets to IPFS with custom permissions. Other researchers can request access to datasets published by investigators, utilising the rich BIDS specification to build sophisticated queries. Compliance with data self-sovereignty regulations is achieved by granting research participants universal opt-in/out privileges that supersede researchers' sharing preferences.

Flowchart

The decision tree for uploading a dataset to the platform and setting permissions is shown below:

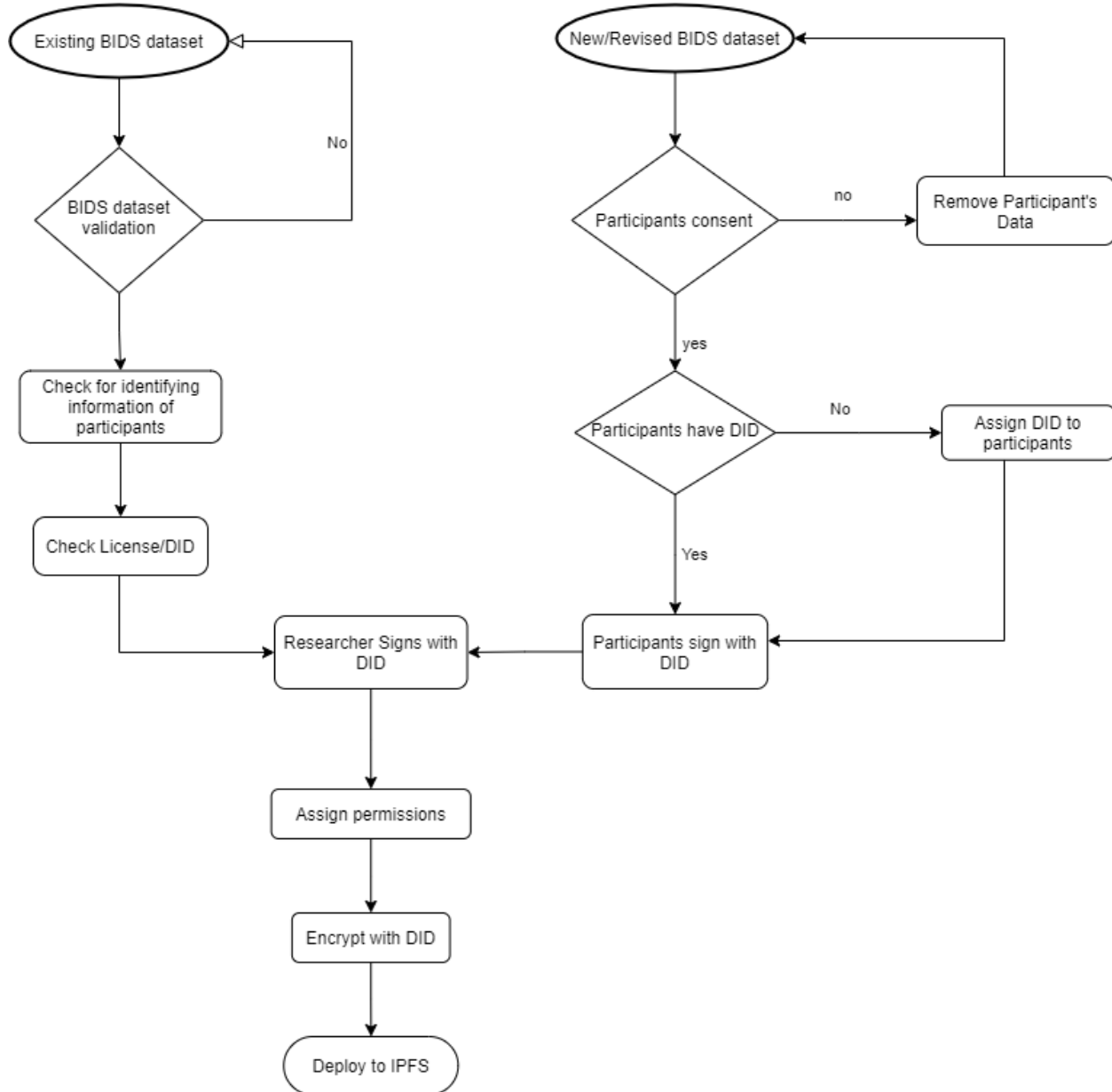


Fig 1 (Flowchart depicting how the Platform will function)

- Left-hand side flow describes how an existing dataset is validated for BIDS specification, identifying the participant's information and any current license provided by the Researcher.
- The right-hand side flow describes the steps to add a new or append new data to a study, first starting with participant's consent, assigning a DID and signing use terms of their data with their DID.
- The researcher's DID signature validates the consent form; Permissions are assigned for accessing/altering data.
- Signed data is encrypted with private keys.

- Finally, the data is deployed to IPFS, where the participant has superseding control over access privileges, followed by the researcher or dataset curator.

PROJECT IN DETAIL

The project can be broken into three parts and is a part of the [issue](#)

1. Building Frontend for the Application
2. Developing IDX schema for mapping schemas to users
3. Integrating Ceramic to deploy encrypted datasets to IPFS

1. Building Frontend for the application

The Frontend of the application will be built upon using the [Ocean market fork](#). The current fork is a React app built with Gatsby.js + TypeScript + CSS modules and connects to Ocean components in Rinkeby by default. It utilises GraphQL for querying data and has many Ocean Libraries for different purposes.

We intend to build our application based on existing work. The UI will include a profile and landing page for the user, A marketplace to search data so any user after login can search for datasets they are looking for by providing the author's DID or metadata-associated keywords.

Users will be presented with multiple log-in options: signing in through a pre-existing DID via **3ID-Connect API**, the ability to create a new DID at login via the **IdentityWallet SDK**, or with an existing Google Account.

3ID-Connect is an account & key management application run in an iFrame, which is privately controlled by users and is accessible by wallets and decentralized applications (dApps) during onboarding. 3ID-connect supports a w. So it is well suited for our application as the participants can be diverse.

```
const authSecret = new Uint8Array([ ... ]) // 32 bytes of entropy used to authenticate
const authId = 'myAuthenticationMethod' // a name of the auth method
const ceramic = ... // An instance of Ceramic (either @ceramicnetwork/core, or @ceramicnetwork/http-client)

const threeId = await ThreeIdProvider.create({ getPermission, authSecret, authId, ceramic })
```

Fig 2 (Assigning DID from the authentication method used)

Users that are not a part of the existing DID ecosystem will be assigned a DID to them using **IdentityWallet**, a JavaScript SDK that allows developers to create and manage 3ID identities. This SDK has the advantage of already being a part of the Ceramic ecosystem and thus will integrate easily with IDX and 3ID-Connect.

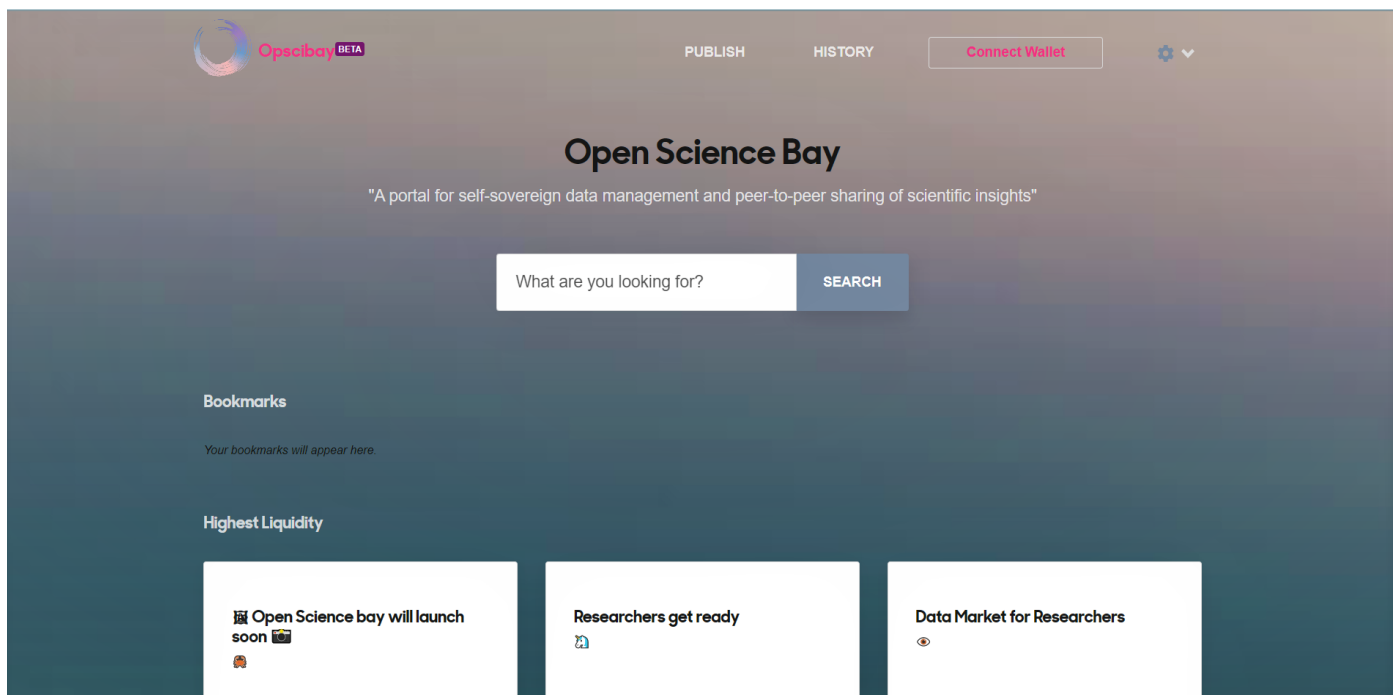


Fig 3 (A UI design made using Ocean's Market page)

Keeping the [long term goals](#) in mind, The application will be designed to leave room for future additions.

2. Developing IDX schema

IDX is an identity protocol that replaces user tables with a single user-centric index. The index provides a unified place where the application can register and discover data sources associated with a user. The index allows your application to store data in any storage backend while still providing users with the aggregation, metadata, schemas, and routing required to make this data consumable by any other application.

IDX would enable publishing a BIDS schema or any arbitrary dataset schema and parse for each identity.

To further explain how IDX functions, The development process is broken into two parts:

A. Writing records

1. The developer creates a schema.
2. The developer creates a definition and includes the **schemaURL**.
3. The user creates a record that conforms to the definition.
4. The user adds the **definitionID** and **recordID** to their index.

B. Reading records

1. User queries an index using a DID and a **definitionID** or an alias.
2. The user gets back the record that corresponds to the **definitionID**.

The following class diagram can elaborately explain the functioning of IDX by IDX developers:

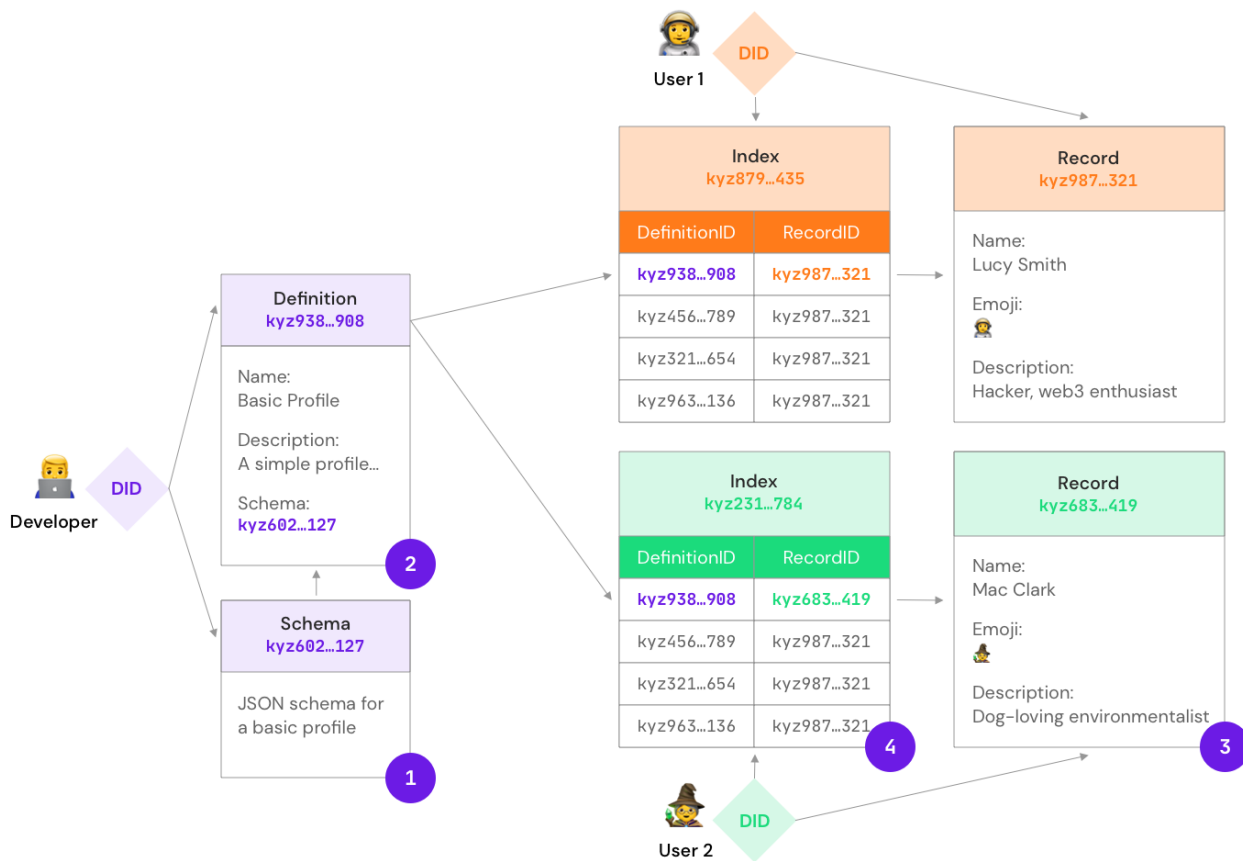


Fig 4 (Working of IDX by IDX developers)

- The developer defines a Schema(1), which forms a part of the Definition(2).
- Every Data is defined by definitions and stored in records(3).
- These records are stored in indexed(4) user tables that can be used across different applications.


```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "BasicProfile",
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "maxLength": 150
    },
    "image": {
      "$ref": "#/definitions/imageSources"
    },
    "description": {
      "type": "string",
      "maxLength": 420
    },
    "emoji": {
      "type": "string",
      "maxLength": 2
    },
    "background": {
      "$ref": "#/definitions/imageSources"
    },
    "birthDate": {
      "type": "string",
      "format": "date",
      "maxLength": 10
    },
    "url": {
      "type": "string",
      "maxLength": 240
    },
    "gender": {
      "type": "string",
      "maxLength": 42
    }
  }
}
```

Fig 5 (A sample IDX schema)

The benefit of IDX for a use case like this is that it makes it easy for other applications to pull in the data. For instance, if all the papers produced by a lab or all the papers where a certain person was a coauthor were queried using a web interface. With IDX, we can verify authorship and create an access control system based on authorship.

3. Using Ceramic for Deploying

Ceramic Protocol is a crucial element for the application. It will be used to create, update, query and upload data to IPFS.

While IDX is an index that maps data to an identity, Ceramic is the protocol for creating data, updating it, and managing its ownership. Essentially, IDX is a framework built upon Ceramic protocol.

Ceramic provides a universal graph of verifiable documents. Ceramic documents are signed, append-only, tamper-proof objects stored in IPFS, encoded using IPLD, and anchored in one or more blockchains. Due to its hybrid design relying on IPFS/IPLD, Ceramic's document graph is interoperable, scalable, permissionless, and low cost.

Ceramic uses DocID to give a unique id to every document to perform operations efficiently. Latest commits can also be tracked using `commitID` of updated docs.

Parameter	Required?	Value	Description
anchor	optional	boolean	Request an anchor after the update was made
publish	optional	boolean	Publish the update to the network
sync	optional	boolean	Sync document updates from the network

Fig 6 (List of methods when create/change/load a doc)

For example, a lab's research would be published to Ceramic by the lab and put on the lab's IDX index, then linked to each author's IDX index. The opt-in/out feature mentioned in the [synopsis](#) will be implemented in this phase with Ceramic. After the participant's consent, the document will be signed, encrypted and uploaded to IPFS.

Documentation

Good documentation is an integral part of any successful project. It also helps in receiving a wider acceptance from the developer community. During the course of development, I would write extensive documentation of various aspects of the project. Some of the aspects are listed below :

- Local setup guide and various configurations will have a separate section. I will try to cover as many “common issues” as possible to make the process of reproduction of the application smooth.
- Parts of the application that interact with the Ceramic backend using API requests will be documented along with typical usage.
- Discussion based on the technology itself and the importance of Desci will be documented for users to realise why to use this application over other conventional apps.

If selected, I also plan to make a weekly blog describe my work during the entire week. I believe that the problems faced by me and their solutions would undoubtedly help other fellow developers later, just like I have received great help from various blogs on the internet.

Testing

The Test suite for unit tests is already set up, with Jest as a test runner for the marketplace. The react-testing library will handle the front end testing while Ceramic-test packages and Jest will be utilised for the backend. The IPFS Deployment can locally be tested with IPFS-Cluster. IPFS Cluster provides data orchestration across a swarm of IPFS daemons by allocating, replicating and tracking a global pinset distributed among multiple peers. To put it simply, Combined with Docker, IPFS-cluster provides an excellent way of mimicking IPFS network testing.

Timeline and deliverables

Minimal Set of deliverables

- Build the Frontend of the application
- Integrate the IDX schema to map users to records
- Use ceramic to deploy to IPFS/Private server

Additional deliverables

- Build the Night mode feature for the application
- A Reward system using tokens for incentive to publish data and a Leaderboard feature based upon tokens
- Build a video guide for new users
- Build a FAQ/Discussion section inside the application

Long Term Goals

- Handle version-controlled datasets with Datalad
- Tokenize BIDS datasets into anonymous computing using the Ocean protocol and iExec confidential distributed cloud computing resources.

Detailed Timeline

Milestone	Tasks	Dates
Community Bonding Period		
0.0	I will take this time to interact with the broader neuro community and attend the OHBM conference this summer. I plan to discuss the project with the Ceramic community to get their advice on implementation and gather more ideas.	May 17th to June 6th
1- Building Frontend Application		
1.1	I will take this time to build a basic UI for searching and uploading data and access granting/revoking for participants.	June 7th to June 14th
1.2	I'll spend this time finalising the layout with my mentor and make the application work with basic authorisation and testing.	June 15th to June 21st
2 - Developing IDX Schemas		
2.1	This week I will write the basic code for the backend to validate the data and its integration with the frontend.	June 22nd to June 30th
2.2	After integration, The Schemas for IDX will be tested for BIDS specification and simple upload and delete functions will be put up.	1st July to 8th July
2.3	The whole frontend is to be finished, and a presentation for the first evaluation will be prepared. Bugs and suggestions will also be taken care of during this period.	9th July to Evaluation
Evaluation 12th - 16th July		
3 - Ceramic deployment to IPFS		
3.1	Add the update, delegate, delete, search features from ceramic API.	16th July to 24th July
3.2	Add the permission hierarchy system for participants and users as mentioned in the synopsis, as well as the opt-out feature	25th July to 2nd August
3.3	I will take this time to test all the features and detect bugs using a private server instead of IPFS to upload data.	2nd August to 7th August
3.4	Write documentation and usage examples. (Less time is given here since a chunk of the documentation would have been written at the time of development itself)	8th August to Submission
Final Submission		

Communication

I will be in regular contact with mentors using Discord or Skype. Besides, Opscientia regularly has open hacking office hours every week hosted on discord. We will have our weekly reviews and discussions during these hours.

About Me

I'm **Achintya Kumar**, a Sophomore at the National Institute of Technology Karnataka, Surathkal, India. I am currently pursuing **B.Tech** in Information Technology.

Motivation

The transformative potential of IPFS for powering decentralized applications on Web 3.0 inspires me to consider how the internet can be made more inclusive, censorship-resistant, and empower the individual through self-sovereignty of personal data. I believe the decentralized web is approaching maturity for the widespread use and will soon transform various aspects of our life, making the world more open and at the same time upholding personal privacy!

I see this project as an excellent opportunity to make a meaningful contribution to the Neuroscience community by solving low hanging problems for implementing IPFS and DIDs within existing software.

Match

My existing skillset includes Javascript- Node and Angular, Smart contract scripting and deployment via Hyperledger Fabric. I was an intern at **CSD-NITK** and made Virtual laboratory experiments for the Undergraduate Courses at the behest of the Ministry of Education, Government of India. You can find some of the projects on [Github](#)

My experiences have made me a good frontend developer, and my love for blockchain and Web3.0 made me explore existing open-source projects. Being a part of a new organisation such as Opscientia and building tools that are going to be used by many, from the ground up is an experience that I look forward to.

Other Project

This is the only Project I will apply for this summer at GSoC.

Working time and Commitments

I will be working full time on this project. My school work for the current semester will be over well before the community bonding period. So, I am willing to devote 20 hrs a week as I have no other commitments during the GSoC period.

Other Plans

Near July end, my Institute may open for next semester. The timeline is put up to make sure most of the coding will be over by then. Besides, academics are easier to deal with during the beginning of the semester.



Other Details

Resume: [resume](#)

DID: [did:3:bafyreideiwcju2cweccxesu7woyufadeot67ichrn7kuefmlqpqgrloeq](#)