Links

- Hedgey claims documentation: https://github.com/hedgey-finance/DelegatedTokenClaims
- Approved grant: https://app.charmverse.io/op-grants/gitcoin-og-c-7068876388885614

Workflow

1. Admin (dev) uploads approved grants into
    a. Assumption: this will be done periodically in batches (no more than weekly)
    b. Assumption: approved grants will be formatted in a CSV from Charmverse. Not read/written on chain
2. All uploaded grants are displayed publicly
    a. This should show metrics for delegated grants funds and total grants given
    b. Reading lock-ups, delegations, claims is all on-chain (via Hedgey indexed data)
3. A grantee can sign in with wallet to claim funds
    a. Must verify that sign-in wallet matches claiming wallet provided in CSV
4. Grantee delegates to approved delegate
    a. Will call Hedgey API
    b. Delegate info available via Agora API: https://vote.optimism.io/api_v1
    c. When claiming, you can find delegates via an explore interface
5. Grantee claims funds
    a. Hedgey API call
6. Public-facing view is updated with status and delegation

Data needed (grants applications)

Data will be sourced from Charmverse. You can see an example grant with available data in the links section (Gitcoin OGC).

- Claim UID (to link to Hedgey data)
- Grant title -> CSV?
- Grant description -> CSV?
- Total awarded amount -> CSV?
- Cycle
- Milestones (one column per milestone) -> JSON might be better for this
- Milestone status (one column per milestone) -> JSON might be better for this
- Milestone rewards
    - This is data that will likely need to be manually added by the grants team for current/future grants. It will allow us to map the rewards available to claim for the specific milestone.
- Grant claiming address
- Status
    - Only include grants in the export with status of In Progress or Complete

Example CSV:
https://github.com/user-attachments/files/17610993/CharmVerse.x.Gitcon.Data.Export.database.export.zip

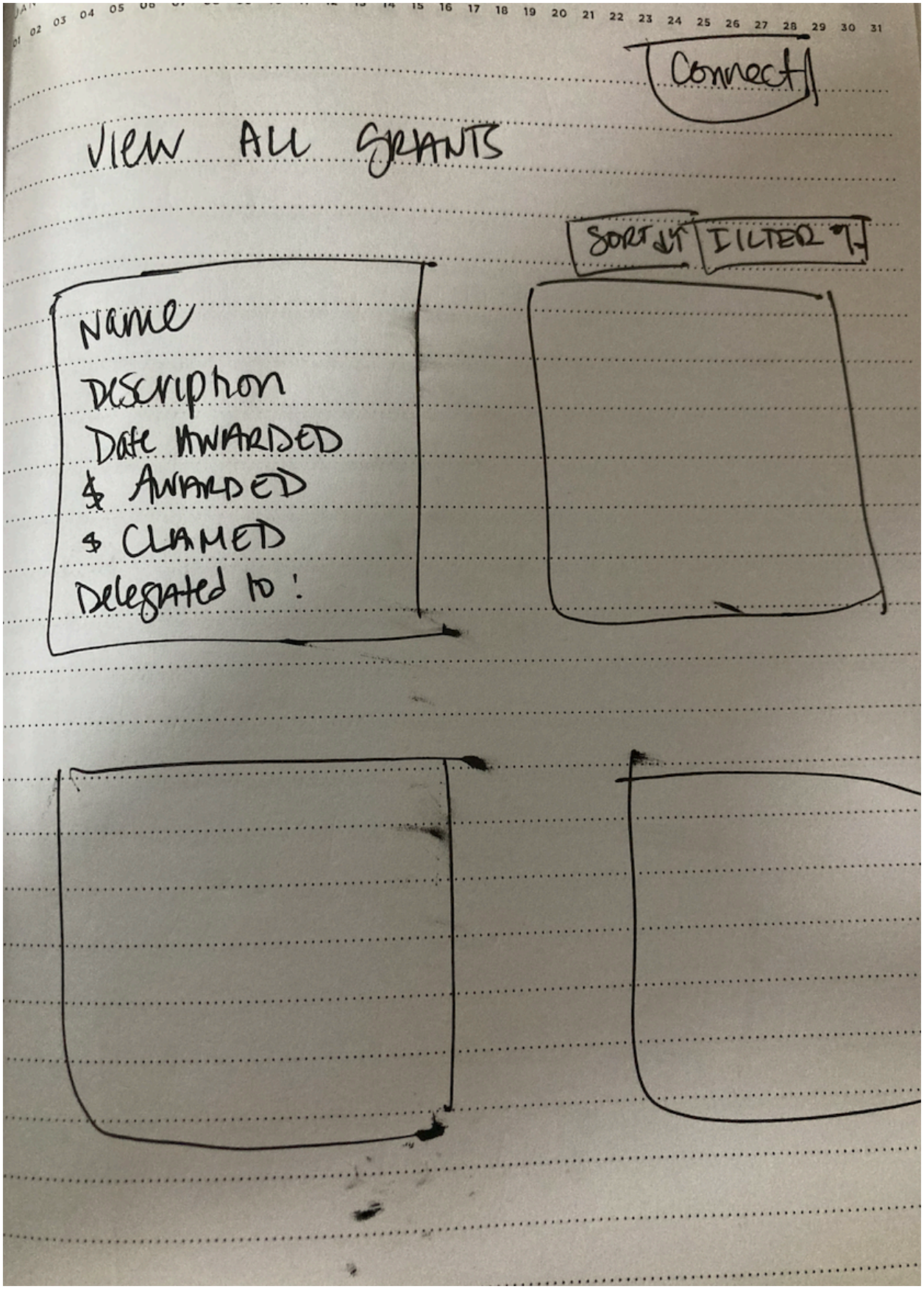
Data	Type	Source	Comments	Docs	In CSV
ClaimUID	UUID	OP after creating claims	References an allocated Claim in the Hedgey TokenDelegation contracts	https://hedgey.gitbook.io/hedgey-community-docs/for-developers/technical-documentation/token-claims/using-the-hedgey-apis-for-token-claims	
Grant title	string	OP after creating claims	Better UX for grantees than UUIDs		CHECK
Grant description	string	OP after creating claims	Better UX for grantees than UUIDs		CHECK
Total awarded amounts	bigint	Opt 1: OP after creating claims Opt 2: Hedgey API	Can fetch from Hedgey API, if we'd leverage the info in the CSV loading would be faster	https://hedgey.gitbook.io/hedgey-community-docs/for-developers/technical-documentation/token-claims/using-the-hedgey-apis-for-token-claims	
Delegated token address	address	OP	Not needed for first pass, all delegation are with OP		
Cycle	int	OP	Not needed for first pass, we focus on shipping a single		CHECK

			delegation round for the MVP		
Milestones	JSON Object or array	OP	Not needed for first pass, we focus on shipping a single delegation round for the MVP		??? they're free form text?
Milestones	JSON Object or array	OP	Not needed for first pass, we focus on shipping a single delegation round for the MVP		
Milestone rewards	JSON Object or array	OP	Not needed for first pass, we focus on shipping a single delegation round for the MVP		
Grant claiming address	address	User/wallet	Use connected account		
Status	Enum (string or number codes)	OP	Only 'In Progress' or 'Completed'		CHECK
Delegates	JSON[]	OP Agora API	We need a source for delegates, potentially with their profiles. Users can delegate to any address, so we also	Example: https://vote.optimism.io/api/v1/delegates?limit=10&offset=0	

			need a field input		

Data needed (Hedgey)

VERY ROUGH SKETCHES of what the flow could look like


Connect
VIEW ALL GRANTS
SORT
FILTER
NAME
DESCRIPTION
DATE AWARDED
$ AWARDED
$ CLAIMED
DELEGATED TO:

CLAIM $100 OP

BUT first, delegate!

View delegates here: LINK

Delegate to [0x ABC 123 ....]

[DELEGATE]

CONGRATS, YOU'VE DELEGATED!

Now claim your $100 OP

[CLAIM]