Impact Evaluator Grant

Milestone 2

What's Cookin` Team

Impact Evaluator Git Repository

Impact Evaluator Components

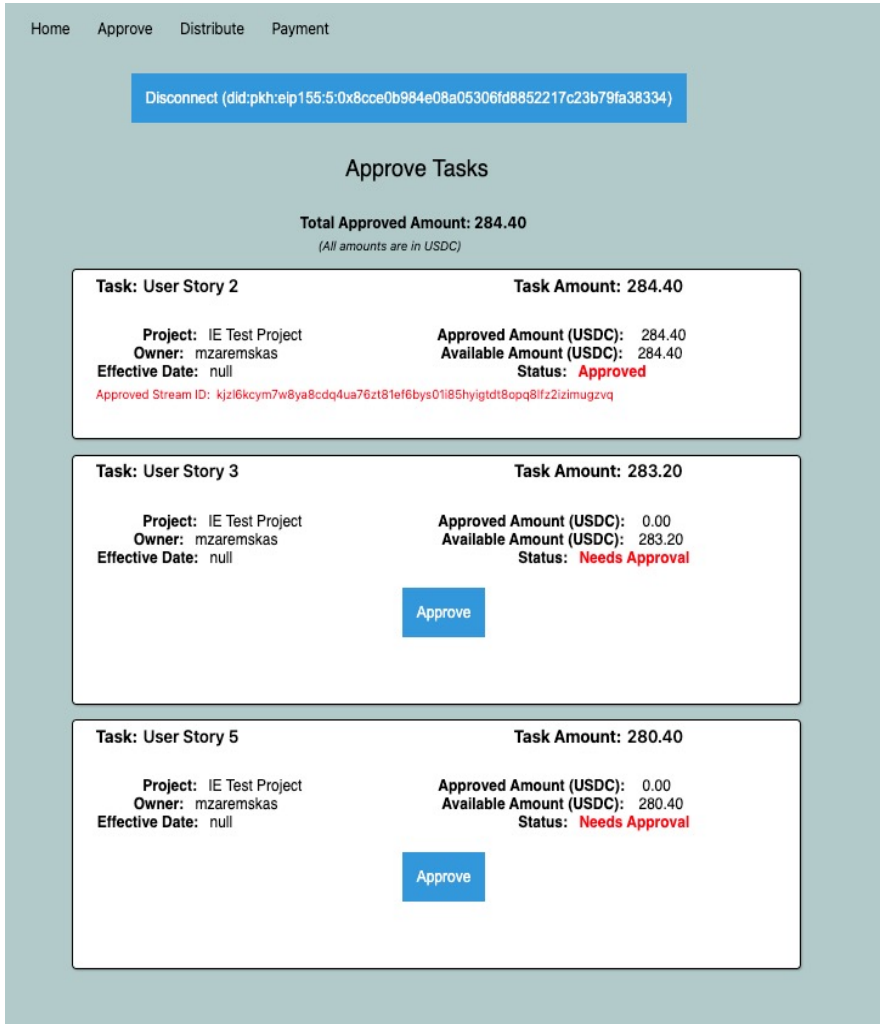
- **Impact Evaluator Console**
  - POC is implemented using Taiga as the source
  - The goal is to support a pluggable source for Claims.
  - The console enables a workflow flow of:
    - Approve Tasks – (”Approved” tasks result in creation of “Approved” claims in composeDB
    - Distribute Funds for the completed task to the award recipient (”Earned” tasks result in creation of “Earned” claim in composeDB)
    - Submit for Payment:
      1. Retrieves “Approved” and “Distributed” claims from composeDB
      2. Pin claims to web3.storage
      3. Initiate Bacalhau Calc using web3.storage CID.
- **Bacalhau Server**
  - Provides method for a user interface to initiate a Bacalhau program. It allows for information such as parameters and IPFS CIDs to be passed down to the Bacalhau program and any returned files to be passed back to the front end
- **Bacalhau Calc**
  - Receives the CID for the claims pinned to web3.storage
  - Calculates the rewards and balances them to the amount in the Approved Reward.
  - Returns a Merkle tree of rewards.
- **Wrapper Contract**
  - Receives a Merkle tree of rewards
  - Sets a one-day challenge
  - Host the token airdrop address

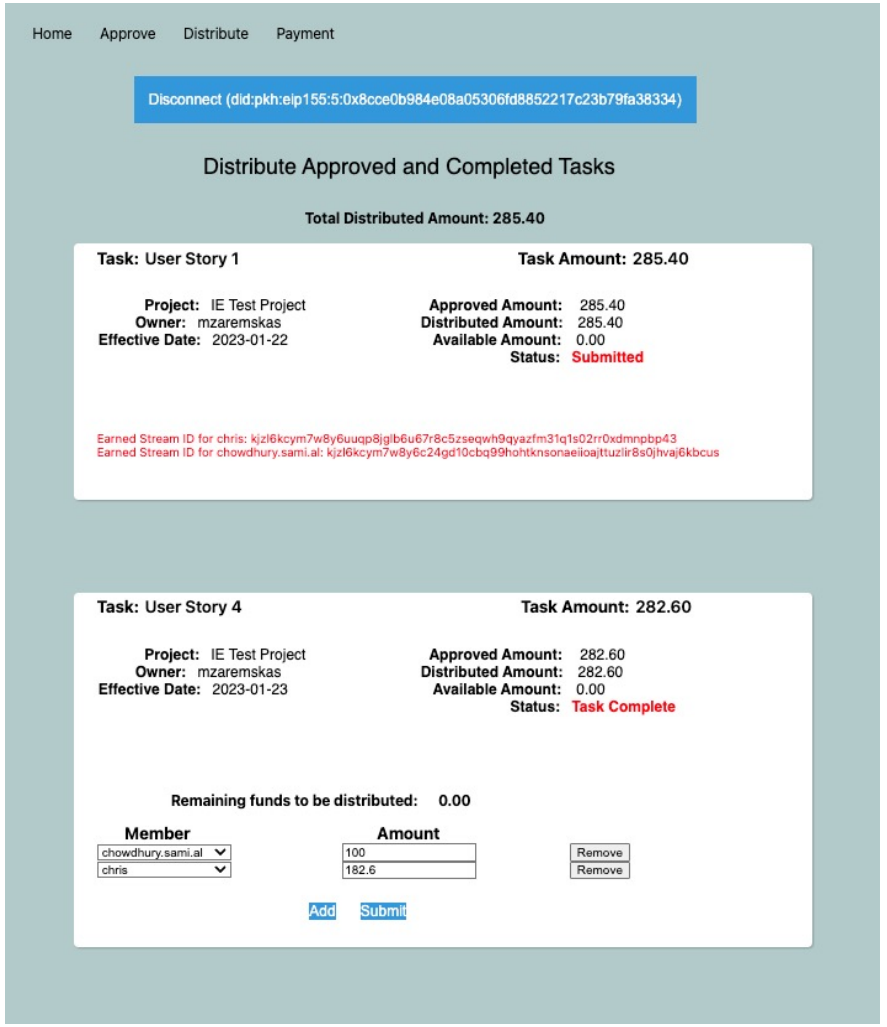
Reusability

Each of the components within this solution offers opportunities for reusability. The following outlines each component's current level of reusability and opportunities for improving future reusability.

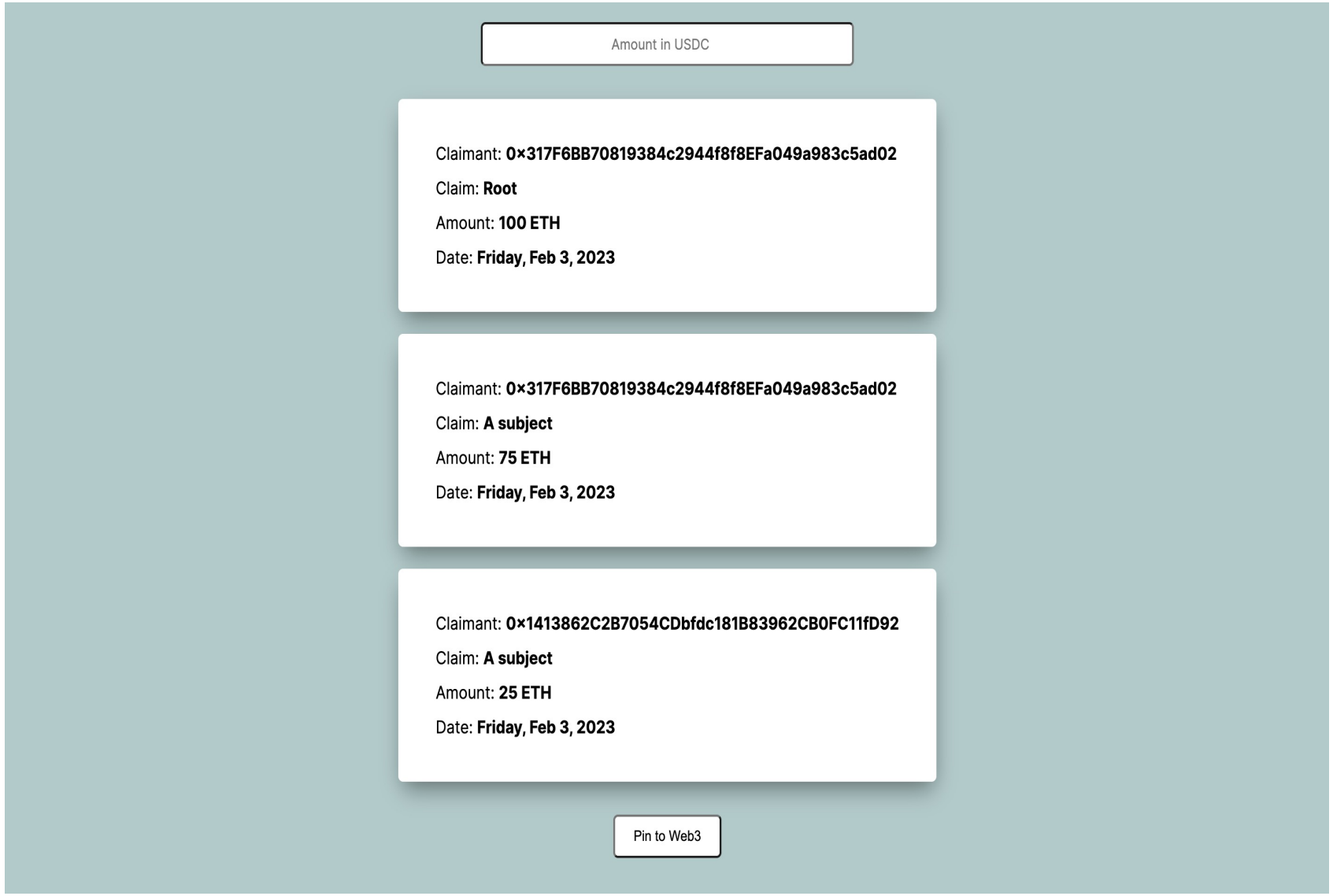
- IE Console – the sources for the data fed into the evaluator for each round can be a pluggable component. The POC is implemented with Taiga as the source of the data, but additional sources can be added. The pluggable component would simply be an extract of the source data and a mapping to the standard claim model.
- Bacalhau – The compute component can be made a pluggable component allowing the implementor to customize the rules of the calculation.
- Wrapper Contract – The wrapper can support any ERC20 token.

IE Console Screenshots

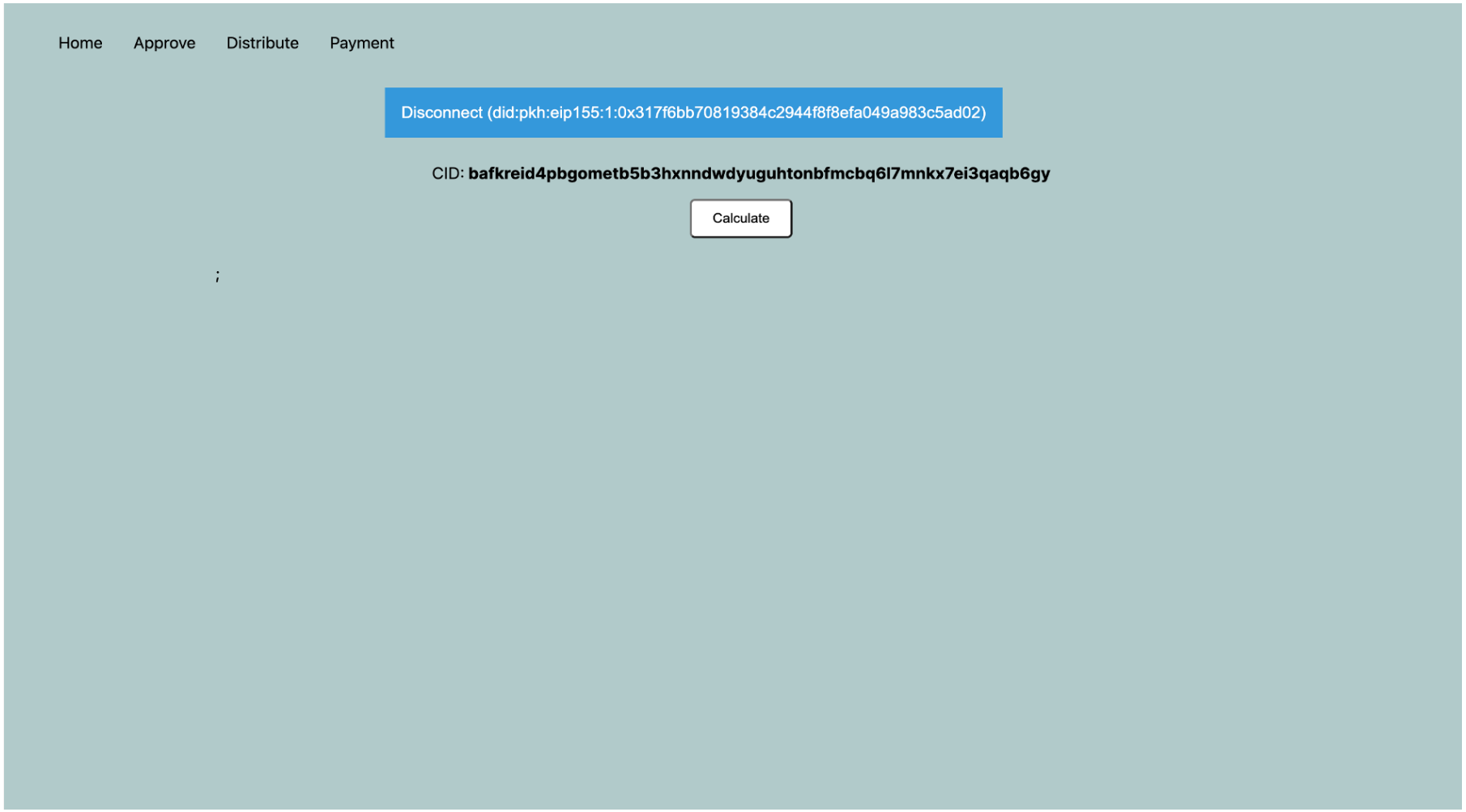




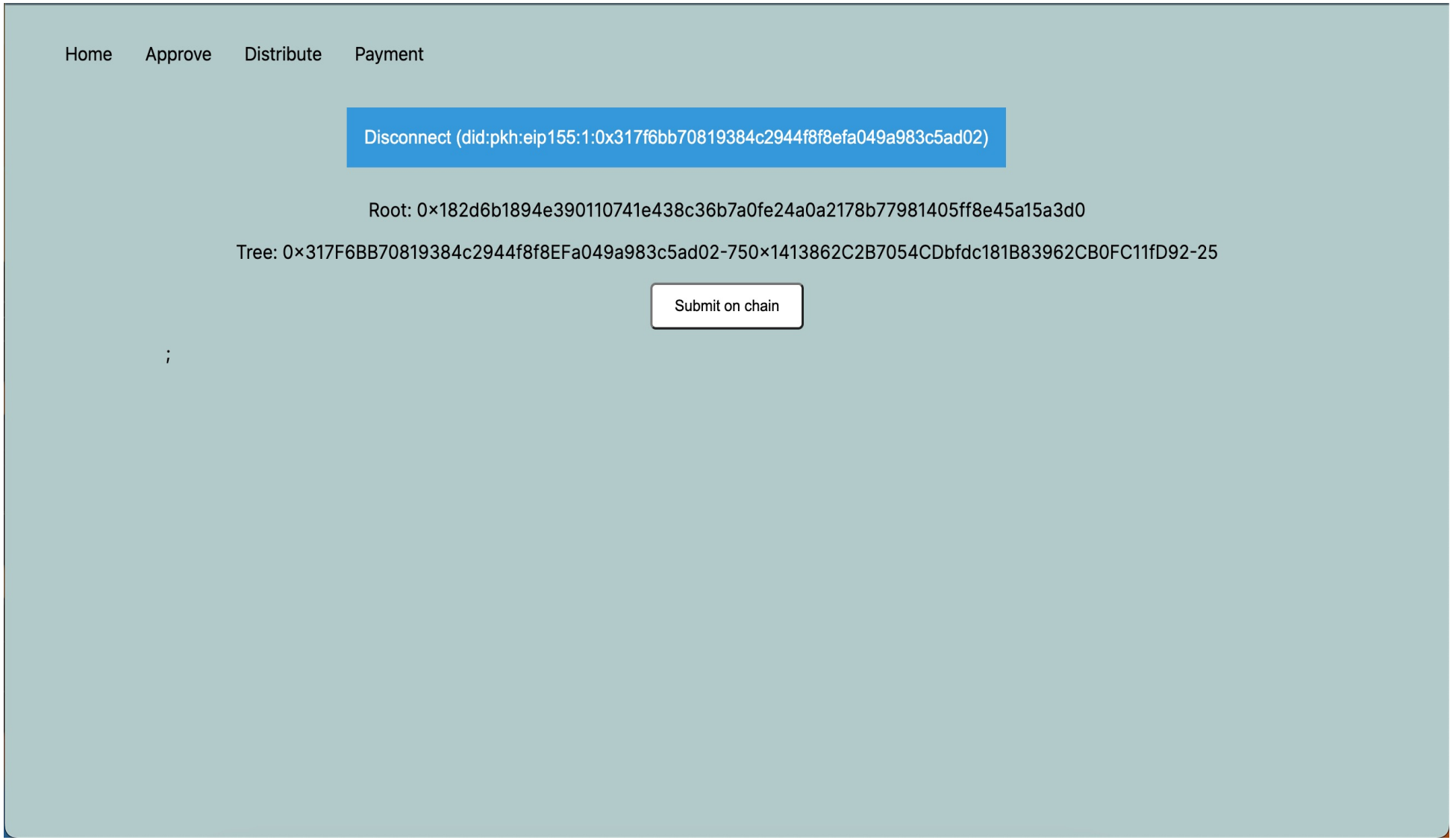
IE Console Screenshots



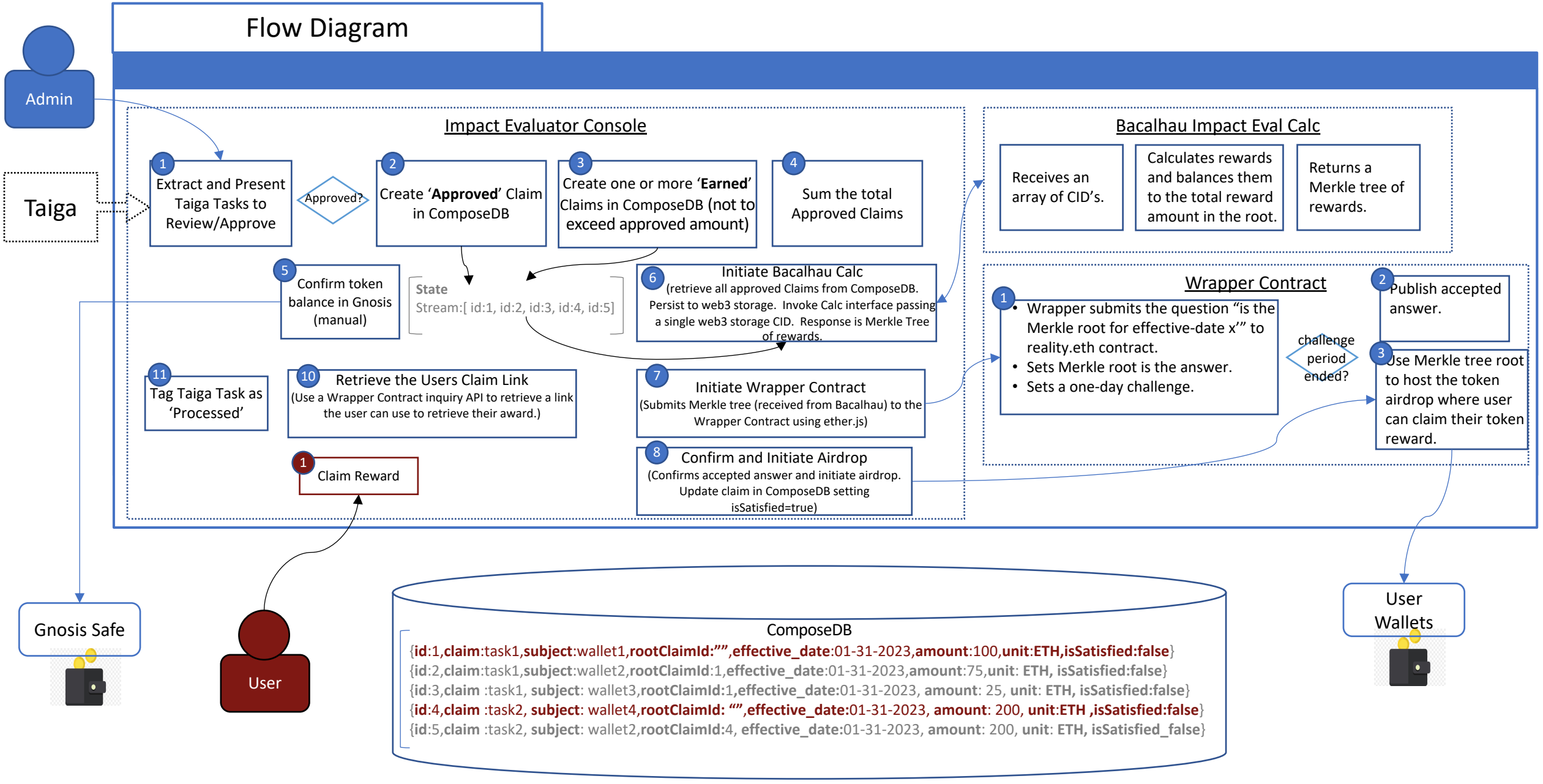
IE Console Screenshots



IE Console Screenshots



Flow Diagram



Claim Model

```
type IEClaim
@createModel(
accountRelation: LIST
description: "Claim or attestation, possibly from 3rd party sources"
) {
subject: String @string(minLength: 1, maxLength: 256)
claim: String @string(minLength: 1, maxLength: 1024)
root_claim_id: String @string(minLength: 1, maxLength: 1024)
round: String @string(minLength: xx, maxLength: xx)
amount: Int
amountUnits: String @string(minLength: 1, maxLength: 16)
isSatisfied: Boolean @boolean
effective_date: String! @string(minLength: 1, maxLength: 10)
statement: String @string(minLength: 1, maxLength: 16384)
source: String @string(minLength: 1, maxLength: 1024)
digestMultibase: String @string(minLength: 1, maxLength: 256)

aspect: String @string(minLength: 1, maxLength: 256)
object: String @string(minLength: 1, maxLength: 1024)
confidence: Float @float(min: 0, max: 1)
rating: Float @float(min: -1, max: 1)
intendedAudience: String @string(minLength: 1, maxLength: 256)
respondAt: String @string(minLength: 1, maxLength: 1024)
}
```

Resources

[Impact Evaluator Git repository]

Approved Claims will not have a root_claim_id value. Earned claims will have the stream id of the Approved claim.