

# GIMIX: AMIX WITH GITHUB

Agoric Internal  
Hackathon 2023

DC, MM, BT, PC

The screenshot shows the GIMIX web interface. At the top, there is a dark green header with the text 'GIMIX' on the left, a 'Select Network' dropdown menu, a 'Logout of GitHub' button, and a user profile icon with the address 'agoric1...dz3xwq'. Below the header, there are two tabs: 'Propose Bounty' (active) and 'Claim Bounty'. The main content area is titled 'Claim Bounty' and contains the following text: 'If you submitted a pull request that closes an issue with a bounty attached to it, use this form to claim the bounty and provide a wallet address for payment.' The form has three input fields: 'Pull Request Uri' with the value 'https://github.com/dckc/awesome-ocap/pull/41', 'Job ID' with the value '1', and 'Wallet Address' with the value 'agoric1a3zu5aqw255q0tuxzy9aftvgheekw2wedz3xwq'. There is a 'Populate with Keplr' button next to the wallet address field. At the bottom right of the form, there is a green button labeled 'Request Claim Invitation'.

GIMIX

Select Network

Logout of GitHub

agoric1...dz3xwq

Propose Bounty

Claim Bounty

### Claim Bounty

If you submitted a pull request that closes an issue with a bounty attached to it, use this form to claim the bounty and provide a wallet address for payment.

Pull Request Uri

An approved pull request that closes an issue with a bounty.

Job ID

The jobId associated with a particular GitHub issue.

Wallet Address

Populate with Keplr

The wallet address where funds will be disbursed.

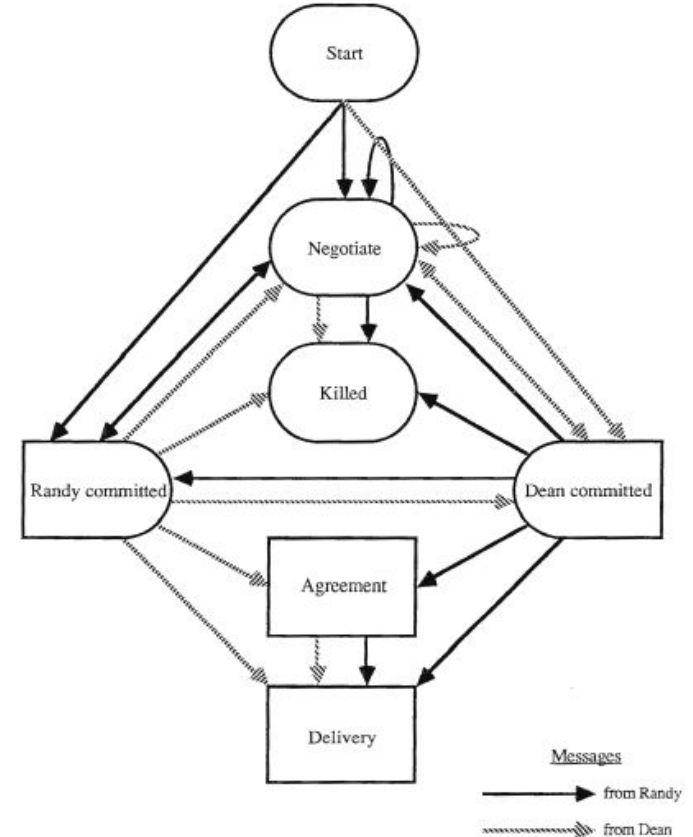
Request Claim Invitation

# AMIX

The American Information Exchange (AMIX) was a platform for the buying and selling of information, goods and services as well as the exchange of information, ideas, and certain kinds of intellectual work product, created ... in the 1980s ... -- [Wikipedia](#)

## Basic AMIX Consulting State Diagram

©1992, F. Randall Farmer and Eric Dean Tribble



# USER STORIES

- As an **open source maintainer**, I want to **incentivize** specific **contributions** to a project I maintain
  - “I'm putting up a \$10k bounty for any project written in Rust that passes > 95% of the prettier JavaScript tests.” - @Vjeux 3:50 PM · Nov 9, 2023
- As an open source maintainer, I want to be able to **charge for feature requests** to my project.
  - “BigCorp wants a backport of feature X”
- As a **developer**, I want to be **paid** for my **contributions** to an open source project.

# USER STORIES

- As an **open source maintainer** with a **specific constraint**
  - “I'm putting up a \$10k bounty for any project written in Rust that passes > 95% of the prettier JavaScript tests.”
- As an open source maintainer **for feature X**
  - “BigCorp wants to use feature X, but it's not supported in Rust. I'm putting up a \$10k bounty for any project written in Rust that passes > 95% of the prettier JavaScript tests.”
- As a **developer** looking to contribute to an open source project

I'm putting up a \$10k bounty for any project written in Rust that passes > 95% of the prettier JavaScript tests.



3:50 PM · Nov 9, 2023 · 531.5K Views

hat  
9, 2023  
arge  
ns to



Propose Bounty

Claim Bounty

### Propose Bounty

Attach a bounty or reward to a GitHub Issue. When a Pull Request that marks the issue closed is approved and merged, the bounty will be released to the PR author.

Issue Url

Provide a GitHub Issue URL.

Amount

Provide an amount you would like to award for this bounty.

Current balance: **1963.45 IST**

Deadline



Select a date for when this bounty will expire.

[Create Bounty](#)

Propose Bounty

Claim Bounty

### Claim Bounty

If you submitted a pull request that closes an issue with a bounty attached to it, use this form to claim the bounty and provide a wallet address for payment.

Pull Request Url

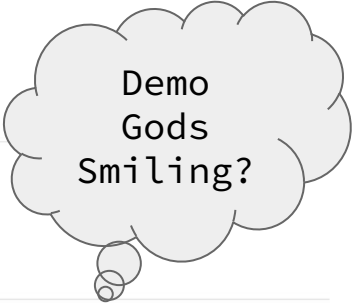
An approved pull request that closes an issue with a bounty.

Job ID

The jobid associated with a particular GitHub Issue.

Wallet Address

The wallet address where funds will be disbursed.



Demo  
Gods  
Smiling?

# USER FLOW - ALICE (PROPOSER), BOB (DEV)

Initial Design  
Brainstorming

1. Alice calls ``makeWorkAgreement()`, supplying payment for all 3 phases
  - a. (reply)--> `(issueUrl: string, terms: { agreement: Amount, delivery: Amount, acceptance: Amount })` {give: 3PhasesPayments})`
2. ... [agreement, negotiation phase)
3. Bob calls ``makeSubmitDelivery(issueUrl: string, prUrl: string)``, ``want: { Delivery: [Payment] }``
  - a. FE does 1) submits on-chain transaction 2) calls BE(oracle) which will verify then call the contract to confirm.
4. Contract has a "GH Oracle", that confirms ``prUrl`` is ``approved + `merged``, and ``prUrl`` closes ``issueUrl``.
5. Contract releases payment to Bob, alice calls ``getPayouts()``.

# CONTRACT DESIGN (DEPRECATED)

Initial Design  
Brainstorming

- Public Facet
  - makeWorkAgreement
    - {want: { brand: oracleBrand, value: issueUrl }, give: {Acceptance: PMT }, exit: {afterDeadline: ... } }
    - handleMakeWorkAgreement => seat.getOfferResult
      - Publish job# to vstorage
      - offerResult is a ~~`jobId` (string)~~ jobObject (invitation makers)
        - makeAcceptInvitation
  - makeChallengeInvitation
    - offerResult = sequenceNumber
    - Sign + put in gist
  - makeDeliverInvitation
    - Contract needs to publish list of Jobs (jobIds)
    - {want: { Acceptance: PMT }}
    - offerArgs: prUrl
      - handleVerifyDelivery()
- 
- Terms
  - githubOracleAddress



# CONTRACT DESIGN V3

*Spec Adequate that  
you can write the  
contract?*

- Startup (CoreEval)
  - Makes an oracle Invitation, sends to `githubOracleAddress`
  - namesByAddress is a term ( privateArg)
- Creator Facet
  - makeOracleInvitation
    - Offer result: { makeJobReportInvitation }
    - Gets called during startup
- Public Facet
  - makeWorkAgreement
    - {want: { brand: oracleBrand, value: issueUrl }, give: {Acceptance: PMT }, exit: {afterDeadline: ... } }
    - handleMakeWorkAgreement => seat.getOfferResult
      - offerResult is a job # (stored in Alice's vstorage)
      - Alice goes to GitHub, invites Bob to do work (@bob, you're hired!)
  - Bob goes to "oracle-enabled BE", i'm bob here's my agoric1masf...
    - I'm Bob, here's my address, here's the job #, and my PR.
    - GH verification. Checks GH to see that Bob has been nominated, PR approved,
    - Oracle calls `makeJobReportInvitation` to send an invitation to Bob, (includes Bob's address)
      - Handler: looks up Bob's address, finds deposit facet + sends him a DeliveryInvitation.  
E(namesByAddress).lookup(addr, 'depositFacet')
  - Bob makes an offer using makeDelivery invitation from the oracle
    - {want: { Acceptance: PMT }, offerArgs: PR }}

# gimix.js

```
import {  
  DeliverProposalShape  
  JobReportProposalShape  
  OracleInvitationProposalShape  
  makeWorkAgreementProposalShape  
  ReportShape,  
  JobsReportContinuingIKit,
```

```
commit 8751cddc03ceb985e736096947f8ada785f1f685  
Author: Mark S. Miller <erights@gmail.com>  
Date: Tue Nov 14 22:45:36 2023 -0800
```

fix: feature complete I think

Yes!

```
* @typedef {object} GiMiXTerms  
* @property {import('@agoric/vats').NameHub} namesByAddress  
* @property {import('@agoric/time/src/types').TimerService} timer  
*/  
  
/**  
* @param {ZCF<GiMiXTerms>} zcf  
* @param {unknown} _privateArgs  
* @param {import('@agoric/vat-data').Baggage} baggage  
*/  
  
export const prepare = async (zcf, _privateArgs, baggage) => {  
  const { namesByAddress } = zcf.getTerms();  
  
  const workByJob = provideDurableMapStore(  
    {
```

# TEST CONNECTS CONTRACT <-> UI

```
commit 8751cddc03ceb985e736096947f8ada785f1f685
Author: Mark S. Miller <erights@gmail.com>
Date: Tue Nov 14 22:45:36 2023 -0800
```

```
fix: feature complete I think
```



**dckc** 3:28 AM

replied to a thread: [@markm](#) I see more progress on the contract. I'm interested to start a test.

🥘 working soup-to-nuts!

just 2 small contract fixes, [@markm](#)

```
~/projects/agoric-sdk/packages/zoe$ yarn test test/unitTests/contracts/gimix/test-gimix.js
yarn run v1.22.19
$ ava --verbose test/unitTests/contracts/gimix/test-gimix.js

- [todo] make work agreement at wallet bridge / vstorage level
[bundleTool] bundles/ bundle-gimix.js valid: 119 files bundled at 2023-11-15T09:05:51.014Z with size 998733
✓ execute work agreement (669ms)
  i oracle received invitation {
    brand: Object @Alleged: Zoe Invitation brand {},
```

# DX NOTES

```
commit 8751cddc03ceb985e736096947f8ada785f1f685
Author: Mark S. Miller <erights@gmail.com>
Date: Tue Nov 14 22:45:36 2023 -0800
```

```
fix: feature complete I think
```

- Short-cut: used **agoric-sdk** repo for test / lint tooling
  - [contract is in agoric-sdk · Issue #6 · agoric-labs/gimix](#)
- Debugging support just barely tolerable
  - Contract bundle requires debugger statements

# RELEASE: PERMIT, SCRIPT, 1 BUNDLE

```
agoric-sdk/packages/zoe$ ORACLE_ADDRESS=agoric1rzl67k9zqpgg8s9w64d58erq809fvpg52433dn \  
  SCRIPT=start-gimix.js \  
  PERMIT=start-gimix-permit.json \  
  yarn test test/unitTests/contracts/gimix/test-gimix-proposal.js  
yarn run v1.22.19  
$ ava --verbose test/unitTests/contracts/gimix/test-gimix-proposal.js
```

```
[bundleTool] bundles/ bundle-gimix.js valid: 119 files bundled at 2023-11-16T18:25:30.617Z  
with size 998697
```

- ✓ **module to script: redact imports; omit export keywords**
- ✓ check / save gimix permit
- i** write permit to start-gimix-permit.json
- ✓ check / **render** gimix proposal
- i** write script to start-gimix.js

3 tests passed


```
agoric-sdk/packages/zoe$ yarn bundle-source --cache-json bundles/ \  
  src/contracts/gimix/gimix.js gimix
```

## ▼ Assets

5

 bundle-gimix.json	975 KB
 start-gimix-permit.json	373 Bytes
 start-gimix.js	3.63 KB

# INSTALL BUNDLE

 Agoric Gov Proposal Builder Local agoric1...dz3xwq



Text Proposal CoreEval Proposal **Install Bundle** Parameter Change Proposal

### Install Bundle

The install bundle message deploys and installs an external bundle generated during the `agoric run` process. The resulting installation can be referenced in a [CoreEval proposal](#) that starts or updates a contract.

---

**Bundle JSON** bundle-gimix.json - 975.27 KB

Upload Cost: 1997.362 IST  
**Insufficient balance.**

**Sign & Submit**

# COREVAL PROPOSAL



## CoreEval Proposal

This is a governance proposal that executes code after a passing vote. The JSON Permit grants [capabilities](#) and the JS Script can start or update a contract. These files can be generated with the `agoric run` command. For more details, see the [official docs](#).

Title

Description

GiMiX is an Agoric dapp to make a market for completing GitHub issues.

See <https://github.com/agoric-labs/gimix>

Write a few sentences about the proposal and include any relevant links. Before proposing to Mainnet, please ensure you've started a discussion on the [Community Forum](#).

Core Eval Files

start-gimix-permit.json - 399 Bytes

start-gimix.js - 4.82 KB



Add more files

Deposit

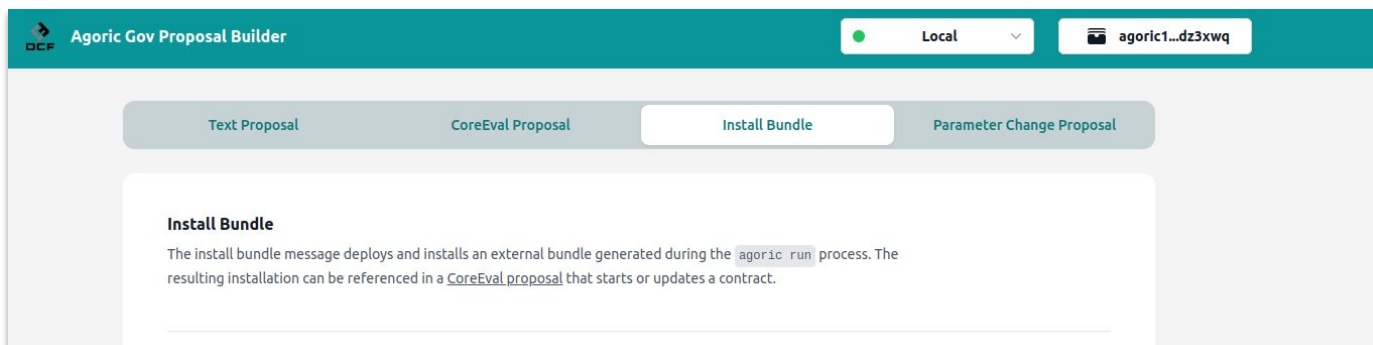
A proposal requires **10 BLD** to enter voting period.

Current balance: **329.937191 BLD**

Voting Period: **10s**

Sign & Submit

# INSTALL BUNDLE



```
gimix$ yarn docker:make mint4k
$ docker-compose exec agd make -C /workspace mint4k
make: Entering directory '/workspace'
agd tx bank send validator agoric1a3z...z3xwq 10000000000ibc/BA313C4...342FA \
  --keyring-backend=test --chain-id=agoriclocal --gas=auto
--gas-adjustment=1.2 --yes -b block
...
yarn --silent agops vaults open --wantMinted 4000 --giveCollateral 1000
```



# COREVAL PROPOSAL

## CoreEval Proposal

This is a governance proposal that executes code after a passing vote. The JSON Permit grants [capabilities](#) and the JS Script can start or update a contract. These files can be generated with the `agoric run` command. For more details, see the [official docs](#).

Title

start GiMiX market contract

Description

GiMiX is an Agoric dapp to make a market for completing GitHub issues.

See <https://github.com/agoric-labs/gimix>

Write a few sentences about the proposal and include any relevant links. Before proposing to Mainnet, please ensure you've started a discussion on the [Community Forum](#).

```
gimix$ yarn docker:make gov-q
$ docker-compose exec agd make -C /workspace gov-q
agd query gov proposals --output json | \
jq -c '.proposals[] | [.proposal_id,.voting_end_time,.status]'
...
["7","2023-11-14T18:22:18.099591606Z","PROPOSAL_STATUS_PASSED"]
["8","2023-11-17T03:12:35.720816716Z","PROPOSAL_STATUS_VOTING_PERIOD"]
```

```
gimix$ yarn docker:make vote PROPOSAL=8
```

REQUESTOR: { give: Acceptance; want: Stamp }

The image shows a web interface for 'Propose Bounty' on the GiMiX platform. The page has a green header with the GiMiX logo, a 'Local' dropdown menu, a 'Logout of GitHub' button, and a user profile icon labeled 'agoric1...dz3xwq'. The main content area is titled 'Propose Bounty' and contains a form with the following fields:

- Issue Url:**  Provide a GitHub Issue URL.
- Amount:**  Provide an amount you would like to award for this bounty. Current balance: 1963.45 IST
- Deadline:**  Select a date for when this bounty will expire.

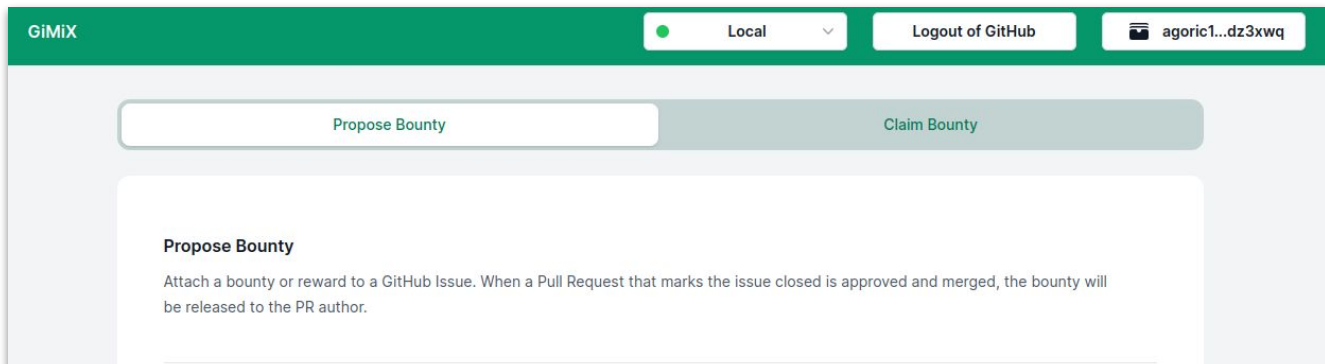
A green 'Create Bounty' button is located at the bottom right of the form.

Overlaid on the right side of the page is a 'Messages' panel. It shows a message titled 'Wallet Spend Action' with a circular refresh icon. The message content is:

- Execute an offer with the following parameters
- Give: 15 IST
- Want: 1 GimixOracle
  - - Fixed https://github.com/dckc/awesome-ocap/issues/38
  - '+1'
- ▶ Invitation Spec

The 'Messages' panel also includes a 'View data </>' link in the top right corner.

REQUESTOR: { give: Acceptance; want: Stamp }



```
{ "status": { "id": 1700192566298,
  "invitationSpec": {...},
  "proposal": {
    "exit": { "afterDeadline": {...} },
    "give": { "Acceptance": { "brand": {}, "value": "15000000" } },
    "want": { "Stamp": { "brand": {}, "value": { "payload": [
      "Fixed https://github.com/dckc/awesome-ocap/issues/38",
      "1" ] ] } } },
    "result": "1"
  }, "updated": "offerStatus" }
```

# NEGOTIATION: ASSIGNMENT (OFF-CHAIN)

## Genode - Sculpt Operating System 23\04 #38

Open

dckc opened this issue on May 16 · 2 comments · Fixed by #41

commented on May 16 · edited ▾

Owner



let's update with this news: [Genode - Sculpt Operating System 23.04](#)

bounty # 1: 15 IST



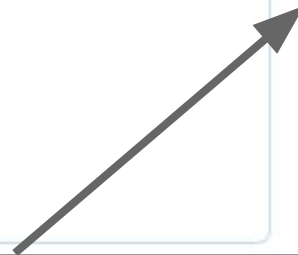
Assignees



dckc

Labels

None yet



Propose Bounty

Claim Bounty

### Claim Bounty

If you submitted a pull request that closes an issue with a bounty attached to it, use this form to claim the bounty and provide a wallet address for payment.

Pull Request Url

An approved pull request that closes an issue with a bounty.

Job ID

The jobid associated with a particular GitHub Issue.

Wallet Address

The wallet address where funds will be disbursed.

DELIVERY

# Genode - Sculpt Operating System 23\04 #38

Closed

dckc opened this issue on May 16 · 3 comments · Fixed by #41

commented on May 16 • edited

Owner

Assignees

dckc

Labels

let's update with this news: [Genode - Sculpt Operating System 23.04](#)

bounty # 1: 15 IST

## genode 23.10 update #41

Merged

dckc merged 1 commit into master from dckc-patch-1 1 hour ago

Conversation 0

Commits 1

Checks 0

Files changed 1



dckc commented 1 hour ago

Owner

fixes #38



genode 23.10 update

Verified

69a51e3

```
const ok =
  pull.author === issue.assignee &&
  pull.status === 'merged' &&
  issue.status === 'closed';
```



Bounty claimed successfully!



Claim Invitation

# ISSUES

- [contract is in agoric-sdk · Issue #6](#) (DX)
- [Oracle fee · Issue #13](#) (WIP)
- [Oracle sends JobReport offer \(TODO\) · Issue #15](#) (WIP)
- [Responder collects bounty \(TODO\) · Issue #16](#) (TODO)
- [Contract instance not found - reported only to console · Issue #9](#) (low risk)
- [timerService: hard-coded boardId \(boardAux?\) · Issue #4](#) (agoric-sdk [#5799](#))
- [claim an NFT rather than Delivery invitation? · Issue #12](#) (alternative design - simpler?)

# RISKS / THREAT MODELING

- Alice can edit the issue
- Bob is dependent on arbiter of the repository
- Bob's damage very little
  - bob can write buggy code

- Oracle
  - can siphon rewards to any party
  - can't steal more than everyone offers
- ~~5. Unhappy Path, No Malice~~
- ~~Joe can frontrun Bob's PR.~~



Q&A?

<https://github.com/agoric-labs/gimix>

# GH ORACLE REQUIREMENTS

- Delivery
  - GH Issue being closed
  - GH PR closing a specific Issue
  - GET PR/approvers
    - Vjeux approved the PR
  - GET PR/merged
- Alice is not impersonated
  - Want to prevent: Bob calls `submitClaim`, with alice's GH account + his wallet address

# NOV 13 - NOV 14 GOALS/GAMEPLAN

- 10am PT meeting
  - Mark has other meeting at 12pm PT
- Goals
  - MM: try implement to implement contract. Have understanding of where i'm confused and what aspects are missing.
  - PC: try to implement all of the oracle requirements. Come to meeting with an understanding of gaps on oracle be / web app side
    - GitHub App - bot
    - Web App (anything where an offer needs to be signed)

# CONTRACT DESIGN V2 (DEPRECATED)

- Public Facet
  - makeWorkAgreement
    - {want: { brand: oracleBrand, value: issueUrl }, give: {Acceptance: PMT }, exit: {afterDeadline: ... } }
    - handleMakeWorkAggreement => seat.getOfferResult
      - Publish job# to vstorage
      - offerResult is a jobObject (invitation makers)
        - makeAcceptInvitation (for alice's use)
        - makeApplyInvitation (invitation where bob can apply for the job)
  - makeRequestApplyInvitation
    - Bob gives deposit-only facet to Alice
    - Alice has to deposit applyInvitation to Bob's deposit-only facet
  - Bob uses apply Invitation
    - {want: { Acceptance: PMT } }
- Terms
  - githubOracleAddress

Vjeux's Want = [GH issue to be completed | PR that closes issueX]

- Inherits PR approval process from GitHub repository rules

[Impersonation, establish correspondence between GH contributor's handle, and on-chain presence of contributor/wallet] = agreement

Submitting PR = delivery

Merging PR = acceptance

Acceptance => done via GitHub

Escrow/Payment => happens on chain

# WEDS: NEXT STEPS

- DC: promote test coreEval to real core eval script
- DC: file a want {} bug; assign to MM
- MM: non-happy-path testing
- MM: add oracle fee
- PC: “claim” endpoint - verifying claims, reporting on jobs
  - Set up stargate signer on server
    - Mnemonic as env var
    - Stretch goal: bot writing github comments
  - Start work agreement form

Meet again 10amPT Thu