[Screen flow](#_jrhing9vvio4)

[DB tables](#_tml1tr1tlp9o)

[Design considerations](#_rjw1huahx49y)

[JSON REST APIs](#_e4xk2ulu3efz)

## Screen flow

One Main Screen (keep things simple)

It displays top rated missions.

Screen 1 - User sign up

User\_name

Email

Password

Public\_key (optional)

Screen 2 - Missions

Suggest a mission, such as for asteroid Ryugu, 1989 ML, Nerous (see <https://en.wikipedia.org/wiki/Asteroid_mining> for profitable asteroid names)

Browse current missions

Screen 3 - proposals under a mission

For each mission, users can suggest different proposals. Users can vote on the proposals.

Screen 4 - contributions under a mission

People can contribute in different categories: legal, technical, monetary, operation (companies who actually go to an asteroid). Users can vote on the contribution. Contributions are divided into these categories:

Good step towards the right direction

Significant

Breakthrough

(they indicate small, medium, big contributions)

Screen 5 - display a mission’s status

Display at-a-glance, what’s the major block

Display stakeholders, list how much each contributors owned.

## DB tables

0) Events\_table

* event\_ID (UUID)
* Time (unix epoch time)
* User\_id (uuid)
* Event\_info (a json array, such as technical, legal, monetary contribution)

1) Users (Stakeholders)

* user\_ID (UUID, just in case a user initially doesn’t have an ethereum public key)
* User\_name (string, for easy collaboration, to identify which user yet with privacy)
* email (string)
* password (string / hash)
* Public key
* join\_date (timestamp)

2) Missions

* mission\_id (UUID)
* start\_date (timestamp)
* name (string)
* description (text)
* Funding\_goal (float)
* Funding\_current (float)
* Status (integer)

3) Proposals

* ID (int)
* previous\_ID (int) - link to previous version of same proposal
* external\_link (string) - link to external document storage for this proposal, could be IPFS, centralized database, Google Docs, Dropbox
* checksum (string) - hash of external document to check it’s the same
* IPFS / checksum link
* Status
* **Users - Proposals stakeholding (TBD) work out how to record how much each user has as stake**

## Design considerations

1. To protect privacy, use public key instead of email. Can we assume every participant have an ethereum account?
2. Although it’s challenging to evaluate one’s contribution exactly, a good way to divide them into three categories (small, medium, big and we phrase them positively). Similarly, the contribution in year 1 obviously is more significant than contribution in later years, we can assign a time value for each passing year to encourage people to start early, say at the end of the year, give each person a 3% value increase (prorated on the months).
3. Any party can propose a mission with funding goal (scientific organization, organizers/operators/startups, etc). People can discuss/vote or contribute. Contribution includes a) financial b) legal c) scientifical d) operational. They take 70%, 10%, 10%, 10% respectively. Once funding is reached, it will be frozen. In case the funding is increased or decreased, the date and reason of change will be recorded.
4. Voting scheme: people vote with their money. voting power = number of tokens you hold.
5. A smart contract: when a contribution (technical or legal) gets enough votes for one category (small, medium, big), it will be recorded. At the project freeze period, it will be converted to tokens: suppose a small contribution is 1, a medium contribution is 3, a big contribution is 7. Suppose there is 10, 2, 1 contributions in each category (small, medium, big) respectively, total tokens is 10,000 which is 10% of total tokens of this project, a small contribution will be converted to 10,000/ (1\*10 + 3\*2 + 7\*1) tokens.
6. Each mission has two buttons: contribute (4 types of contribution), discuss.
7. A legal or technical breakthroughs are applicable to different missions. So under each mission, those breakthroughs proposed for other missions are also displayed.
8. People can also vote their satisfaction for each mission. The vote is anonymous. This will determine the reputation of operators, scientists, legal professionals. This will help with future adventure.
9. All papers will be stored in Cassandra. Computed hash of the article and timestemp will be stored on a blockchain. If a paper is modified, we’ll know.
10. An operator group can decide how to allocate 10% of total tokens of a projects among its team members.
11. A scientist can decide to contribute his/her discovery to two missions. A scientific breakthrough may help with many missions. Of course, it’s too complicated to associate one contribution to unlimited missions.
12. Store technical discussion, legal documents into IPFS. Amazon S3, compute checksum, hash, store them together with a link.

## Smart contracts in English

====================================Section 1, smart\_contract\_rasie\_fund

What's unique about each mission:

mission\_name:

mission\_uuid:

**mission\_goal**:

**raised\_so\_far**:

wallet\_tech:

wallet\_legal:

Wallet\_operation:

wallet\_investment:

wallet\_for\_each\_investor:

wallet\_funding:

funding\_freeze\_time:

ACTION 1: initialization, send tokens to those three wallets (tech, legal, operation)

ACTION 2: whenever an investor sends ether to funding\_wallet, the wallet will return same amount of mission tokens (first check we have enough mission tokens)

total\_supply = a large number (total number of ether?)

Portion allocation:

5% to tech\_portion, 5% to legal\_portion, 5% to operation\_portion,

85% to financial\_portion

85/5 = 17

step 1, a mission is created

mission\_uuid: 123

goal: 10 million

raised\_so\_far: 0

total\_supply: a large number so we never run out of supply

legal\_portion: goal/ 17

tech\_portion: goal/ 17

operation\_portion: goal/ 17

financial\_portion: goal

compute the portions, transfer them to these wallets:

tech\_portion,

legal\_portion,

operation\_portion,

step 2, investor with id 899 invested 10 ether

Check if we have enough token (if not, something wrong since we allocated a huge number)

Whenever an investor transfer some ether to Funding\_wallet, he/she will receive same amount of token from us.

//check if the remaining financial token is >= the number of ether invested by this investor

//if so, transfer 10 token to this investor's wallet

//otherwise, put 1000 million more into the total supply

//compute the portions, transfer them to these wallets: tech\_portion, legal\_portion, //operation\_portion, //financial\_portion respectively

then transfer 10 token to this investor's wallet

transfer 10 ether to funding\_wallet

check if goal is reached

if raised\_so\_far + investment\_amount >= goal,

actual\_investment\_amount = goal - raised\_so\_far

if actual\_investment\_amount > 0:

transfer to funding\_wallet

transfer token to this investor

call contract\_release\_fund

otherwise:

actual\_investment\_amount = investment\_amount

transfer\_to\_funding\_wallet

transfer token to this investor

offchain or IPFS to store legal contract, computation result and papers

step 3, (occasionally) funding goal is increased or decreased

if increased, add increased amount to the total supply

become to accept more investments

add more tokens to tech\_portion, etc.

if funding is decreased, a fair way to do is:

suppose it's decreased to 80%

each investor will receive a refund of investment\*20%

token will be reduced to 80%

tech\_portion, etc will be reduced to 80%

====================================Section 2, smart\_contract\_release\_fund:

for each mission,

raised\_so\_far:

wallet\_funding:

wallet\_for\_sucessful\_bidder:

Voting:

Each token holder can send a blank message or a message with only one bit to a voting\_booth\_address, of course, gas will be spent. We tally the vote, a vote from an owner of more tokens carries more voting power.

//fund has raised.

//it may have some time till project launch date which provides preparation time.

several companies proposed to do this project

step 1, is voting period ended (give three days to vote)

step 2, which company has the highest vote? does it have enough vote

voting power is same as how much tokens they have

if they get more than 50% of voting tokens

transfer one third of ether from funding\_wallet to the successful bidder's wallet

step 3, after 2 months, how did they do for the 1/3 of the ether

auditor provides a report

team voting, continue?

if more than 50% of votes, release 2/3 of the ether

====================================Section 3, smart\_contract\_Share\_profit:

for each mission:

total\_tokens: raised\_so\_far \* (1 + 1/17 + 1/17 + 1/17)

Profit\_wallet:

allow people to trade the token (kind of like equity token)? people can trade themselves

(exchange a token with some ether, depending on market speculation). We track new token owners.

suppose mission 1 brings back 100 million Profit (selling the precious metals)

when to sell the metals? within one month, use the money to buy ether

each token will entitle: total ether /number of tokens for this mission

distribute the ether to token holders

the tokens are removed since this single mission.

People send tokens to the profit wallet which returns corresponding portion of ether to them.

## JSON REST APIs

1. Pass mission uuid to blockchain smart contracts
2. Allocate funding
3. Release funding
4. Profit sharing

Parent smart contract

Child smart contract

Call ERC20,

For each mission, create ERC token, set initial total supply

Potential questions from judges and audience:

Is this a kickstarter on blockchain, for the purpose of space exploration