Choosing YT Thumbnail / Instagram Cover Photo / Hinge Profile Picture

\# <u>**Main goal:**</u> Get to know what pictures get me most engagement before uploading

1) Upload photos / assets
   => S3 + Pre-signed URLS

2.) Solana => pay via Solana, dispense via Sol

3.) Sol wallet adapter

\# 4.) How to do This via smart contract
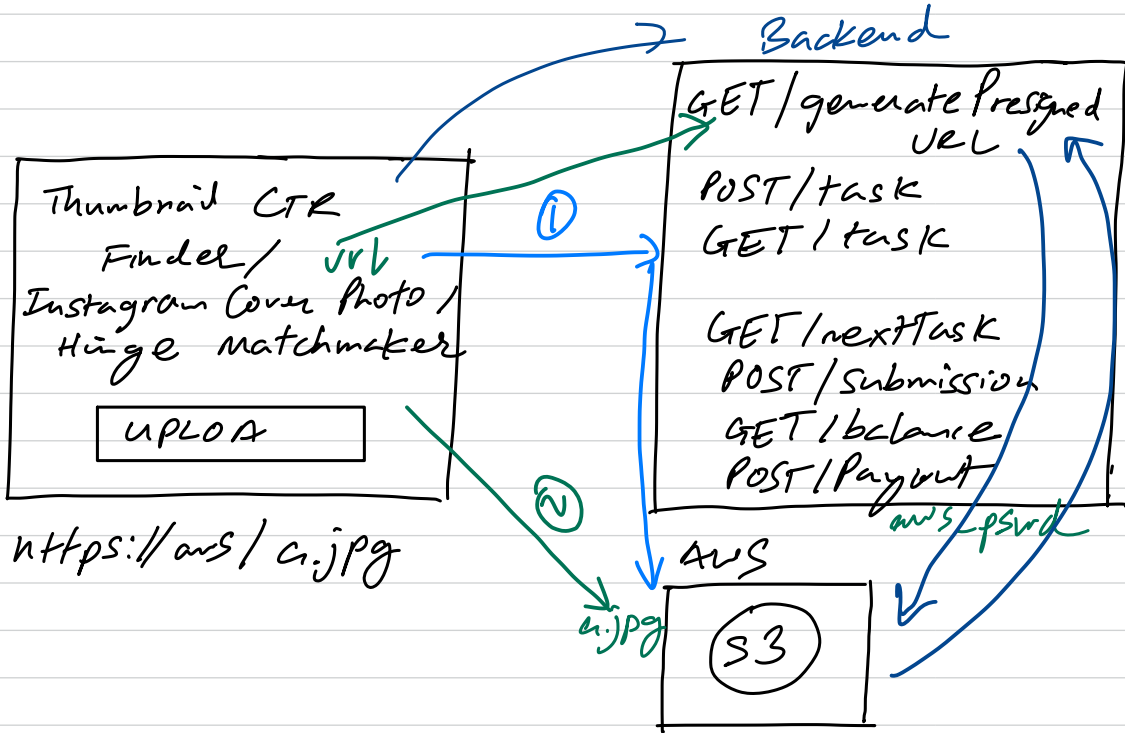
5.) Handling payouts

6.) NextJS, Pretty UI .

web 2	web 3
1.) S3 / object stores	1.) Wallet Adapter
2.) Pre-signed URLS	2.) Signing messages
3.) Content-delivery networks (CDNS)	3.) Transactions and Signing txns
4.) HTTP Servers	4.) Indexing blockchains
5.) Databases (DBs)	5.) RPCs
6.) Async architecture/ workers	6.) workers talking to the blockchain

1.) Letting users upload a photo.

Backend

GET /generatePresigned URL

POST /task
GET /task

GET /nextTask
POST /Submission
GET /balance
POST /Payout

aws/psvrd

Thumbnail CTR
Finder / url
Instagram Cover Photo /
Huge matchmaker

UPLOA

https://aws/a.jpg

AWS

a.jpg

S3

①

②

2.) Letting users pay and verify payment to backend

$20

[Parent wallet]

Link to user's account to verify payment

DB
lid 1 => 8xcEDofgk#9am

(wallet) Address

[Thumbnail CTR Finder / Instagram Cover Photo / Huge Matchmaker

UPLOA]

Backend

linked w/ wallet for verification

https://aws.com/a.png
https://aws.com/b.png
https://aws.com/c.png
sckox#9it862abx2
Transaction signature

Extra step:

Asking users to link their wallets to website (Governance related)

3.) Allowing labellers to payout $ to their own wallet.

wallet
$20

DB
$10
↳ verification starts
(Locked⇒) $10 ⇒ $0
not verified

https://imageuploader.com

YT / Insta / Hinge

Upload

Backend

Userid
—wallet address
private key

Helpers
Frontend

Use Solana CLI
↓
RPC server → transaction

Update?
Do not want to store the private key and still need a way to figure out transactions.