HyperVIBES MVP Coordination

[2021-oct] This is a living document that will be maintained during the initial 3-week MVP development of the HyperVIBES project by Rarible DAO. It will help coordinate product/frontend/contract/graph development.

Project Owners:
Brandon Valosek zaknotzak lootsoot@pm.me mcsanchez6281@gmail.com

Links

- Contract repo - https://github.com/R-Group-Devs/hypervibes-contracts
- Frontend repo - https://github.com/R-Group-Devs/hypervibes-frontend
- Figma Sketches - https://www.figma.com/file/sHnjfrTd6rd11NMRbDKml8/Flows
- Product Brief - HyperVIBES - Product Brief

Glossary

- **Tenant** - A siloed environment and configuration data specific to a single project. Tenants are created and modified by admins.
- **Admin** - Anybody who has set up or can manage a tenant. Creating a tenant is fully open to the public, and so anybody can be admin.
- **End-User** - Somebody who is infusing or claiming from NFTs. End-users' actions are always within the scope of a tenant.
- **Infuse** - The process of depositing ERC-20 tokens into an NFT.
- **Claim** - The process of withdrawing claimable ERC-20 tokens from an infused NFT
- **Collection** - A specific ERC-721 NFT contract
- **Constraints** - On-chain per-tenant configuration that limits the way the protocol can be used
- **Token Balance -** Total amount of infused ERC-20 tokens inside of a specific NFT for a specific tenant
- **Infusion Amount** - Amount of an ERC-20 token being added in a single infusion transaction

Use Cases / User Stories

- As an admin, I want to view all tenants I'm admin for
- As an admin, I want to create a new tenant
- As an admin, I want to modify an existing tenant
- As an admin, I want to view details for an existing tenant
- As an end-user, within the context of a specific tenant, I want to view all infused NFTs
- As an infuser (end-user), within the context of a specific tenant, I want to infuse any NFT from any ERC-721 contract
- As a claimer (end-user), within the context of a specific tenant, I want to claim ERC-20 tokens from any infused NFT I own

Contract

- Immutable Tenant Data
  - Tenant ID (auto assigned)
    - ❓ ~~Alt: client specified ID~~
  - Name, Description (no uniqueness constraints, emitted once during creation)
  - ERC-20 token (stored once during creation)
- Mutable Tenant Data
  - Admins
  - Infusers. May allow *public infusion*
  - Collections. May allow any collection (*open infusion*)
  - Constraints

Constraints

Creating an interesting set of variables that support a wide design space for projects built on top of the protocol.

constraint	**default**	**notes**
min/max daily rate	min=0 max=infinity	Revert if rate outside of range
min/max infusion amount	min=0 max=infinity	Revert if infusion amount outside of range
max token balance	infinity	Infusion amount is clamped if token balance would exceed max
require nft is owned	false	Asserts ownerOf nft = infuser
allow multi infuse	true	Allow re-upping infusion, daily rate cannot change
allow public infusion	true	Dont require infuser is on allowlist, but assert infuser=msg.sender
allow all collections	true	Dont require collection is on allowlist

Examples

min daily rate = max daily rate = 1000 min infuse = 50,000, max infusion = ~1m require nft is owned = true allow multi infuse = false allow all collections = true	This is a basic curatorial configuration similar to the OG VIBES protocol
min daily rate = max daily rate = 0 min infuse = max infuse = 1 require nft is owned = false allow multi infuse = false	Create a "curation game" by airdropping people 0.25 tokens. People must coordinate to get at least 1 token together to curate a piece into the network, "from a distance"

min daily rate = max daily rate = infinity allow multi infuse = true	Infused tokens are instantly claimable, works like being able to "stake" and "unstake" tokens
min daily rate = max daily rate = 100 require nft is owned = false allow multi infuse = false allow all collections = false allow public infusion = false	A collectible/artist project launches and wants to have all its pieces infused post-launch after they've already been distributed. Nobody else can infuse the pieces, but the collectors can mine the tokens over time

Interface Sketch

- Marketing/FAQ section (a la https://0n1gear.io/)
- Select NFT to infuse
- Infuse (tenantId, nft contract, tokenId, amount, rate)
- Select NFT to claim tokens from
- Claim (tenantId, nft contract, tokenId, amount)
- Setup Tenant
    - Select/input NFT collection contract
    - Select/input infusable ERC-20 token
    - Other constraints