Juicebox Project Template Strategizing

The Juicebox protocol, in conjunction with clients adhering to its abstractions, is now well-suited to support custom templates that narrow down the protocol's open-ended money and membership affordances to specific use cases.

Templates for Juicebox projects have been in demand since the protocol's inception. It makes sense to streamline the user experience in order to simplify onboarding and make it easier to use. However, Juicebox templates are not straightforward to address solely on the client-side due to:

- Juicebox projects unfolding over a sequence of Cycles, which are periods during which a project's rules are locked. A project owner can reconfigure future cycles to address emerging needs and establish constraints to prevent abuses of power. Although a project's first Cycle can be templated, subsequent cycles remain open-ended. Concealing contractually available powers from payers and project owners exposes risk.
- The Juicebox protocol operating entirely onchain, meaning modifications to a project's Cycles can still be made bypassing any website that purports to limit a project's operations to a subset of simple, templated actions. Concealing contractual information from payers and project owners exposes risk.

When done correctly, Juicebox templates simplify information and actions while reducing contractual risk for all involved parties (or explicitly exposing new risks as features). This can be achieved by leveraging contractual project owners who hardcode conditions for Cycle changes, and custom delegate contracts that govern the handling of funds entering or leaving a treasury. Clear standards, easily replicable by other open-ended template initiatives and closely coordinated with clients supporting access to various template-based projects, are likely to yield the best results.

These standards for templating have already begun taking shape with the development of the 721 Delegate, Defifa, and Blunt prototypes. I feel we're approaching a good time to formalize the existence of templates in documentation and in web clients who care to support them.

I recommend starting with a specific cohort of template applications, diverse enough to stress test their open-ended capabilities but focused enough to work closely with implementers of documentation, web clients, and protocol standards. This will allow for the careful navigation of initial unknowns that will inevitably arise.

For clients like juicebox.money, I think it also makes sense to begining by becoming more template-aware on the project page – create flow awareness can come later. If Blunt, Defifa, and Bananapus staking become as usable from juicebox.money as 721 Tiered Delegate has, we'll have designed sufficient patterns along the way extensible to many other templated application types into the future.

Here are a couple of well-staffed candidates I'd be excited to work with, and an overview of their unique character:

721 Tiered Delegate

Type:

- custom pay data source.
- optional custom redemption data source.

Features:

- custom create flow.
- custom pay modal parameters.

Status:
The current version of the 721 Tiered Delegate is already supported in full by JBM.

Defifa

Type:

- contractual project owner.
- custom pay data source.
- custom redemption data source.

Features:

- custom create flow.
- hard-coded cycle sequence, with custom public actions to trigger them.
- custom pay parameters, similar to 721 Delegate.

Status:
Full third party support at defifa.net.

Blunt

Type:

- contractual project owner.
- custom pay data source.
- custom redemption data source.

Features:

- custom create flow.
- hard-coded cycle sequence, with custom public actions to trigger them.
- funding targets and limits.

Status:
Full third party support at blunt.finance.

Bananapus Distributor

Type:

- custom project owner.
- custom pay data source.
- custom redemption data source.
- custom pay terminal.

Features:

- custom create flow.
- usage of ERC-20 payment terminal.
- custom pay parameters, similar to 721 Delegate.

Status:
No interfaces yet, but this'll be a fork from the familiar 721 Tiered Delegate.