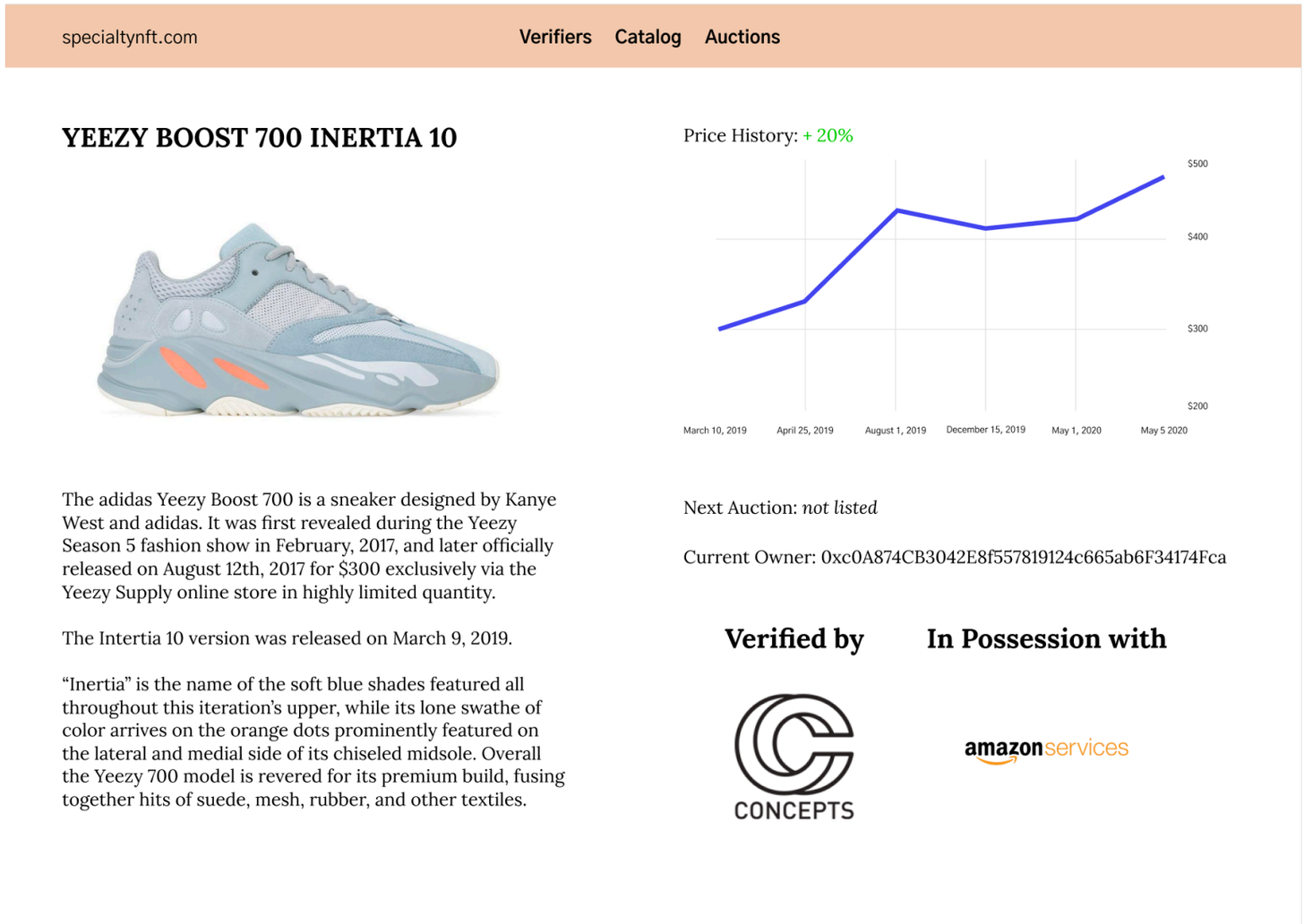
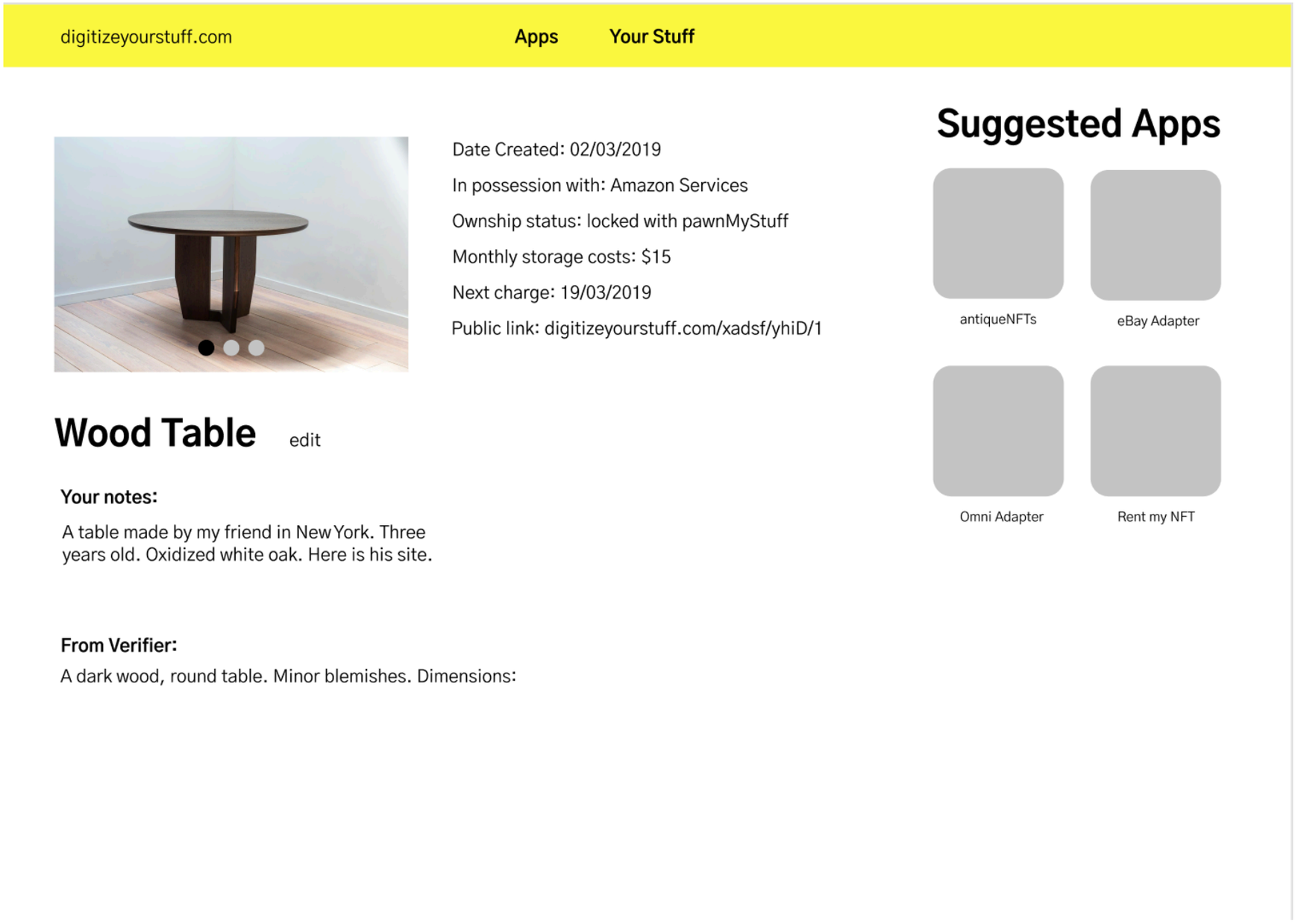
Progress in thinking on Komerka

Two clear go-to markets

1. **Speciality NFTs**: Yeezys, Supreme, etc. NFTs present a better instrument to facilitate the speculation that is already happening with these assets. Making it trusted would require our asset-backed infrastructure. Probably a sustainable business here in that there is a network effect with verifiers.



2. **Digitize your stuff**: ship us your stuff, we'll photograph it, store it, and give you an NFT that represents "rights to collection." These NFTs could be used in a new world of apps, from loans to eBay listing as a service.



So, there’s the go to market question. But right now I am concerned with is what the protocol layer should look like and what value we could capture there, if any.

Reminder: The distinguishing feature of our design is the mapping of real world assets to two tokens: ownership and possession. I believe this is a useful general framework that allows for lots of interesting game theory, trading, and other things.

Recently my concern has been that anything general purpose would have no moats and so no value could be captured at the protocol layer.

I now see a way to build something that is general purpose and defensible.

i.e. I see a path to capture value at the protocol layer

The general idea is to have a single contract *and a single instance* that would allow anyone to leverage our design to make their own tokens.

If all goes well, assets from this instance will become the standard. They'll be used in a host of dApps and services and, going forward, customers will demand that new services are compatible with our assets.

What does “if all goes well mean”

Probably requires:

- Being first to market
- Building the best tools for developers, near term
- Have some success with a consumer facing application

Key to making this work is that the single contract/instance is open to the participation of others and robust to their various wants.

What would this look like?

My idea is for an architecture based on ERC1155.

This is a contract that allows for the creation of arbitrary amounts of NFTs, fungible tokens, and semi-fungible tokens.

It was motivated by game designers who didn’t want to manage hundreds of ERC20 or ERC721 contracts for their in-game assets.

The unique characteristic of our version would be that mint events would create three mappings, rather than the usual one: token => owner.

The three mappings:

Ownership token => owner
Possession token => owner
Possession token => ownership token

All other functionality would be fairly traditional. The Possession => ownership mapping would be fixed, but the others could be changed at will.

Anyone could use this instance to create new assets.

Let’s think on how the ecosystem might develop to meet various user demands.

Example “layer 2” (in our world) functionality.

A first, simple example that might be common is a multi-signature contract to serve as possession custodian. This contract would require that any possession transfer be approved by the owner of the ownership token which corresponds to the possession.

Something more complex:

Let’s say a group of storers want to create a federation, as they think they can make more money working together. They decide rules for membership, storage fees, shipping rules, etc. and encode them in a smart contract.

By some means, possession assets would come to this federation. Ownership token holders might want to work with them because they’re trusted and have guarantees that aren’t in the Komerka contract.

In the Komerka contract’s view, this federation has possession of a lot of assets. Within the federation contract, there is a separate ledger tracking the mappings of Komerka Possession assets to federation members.

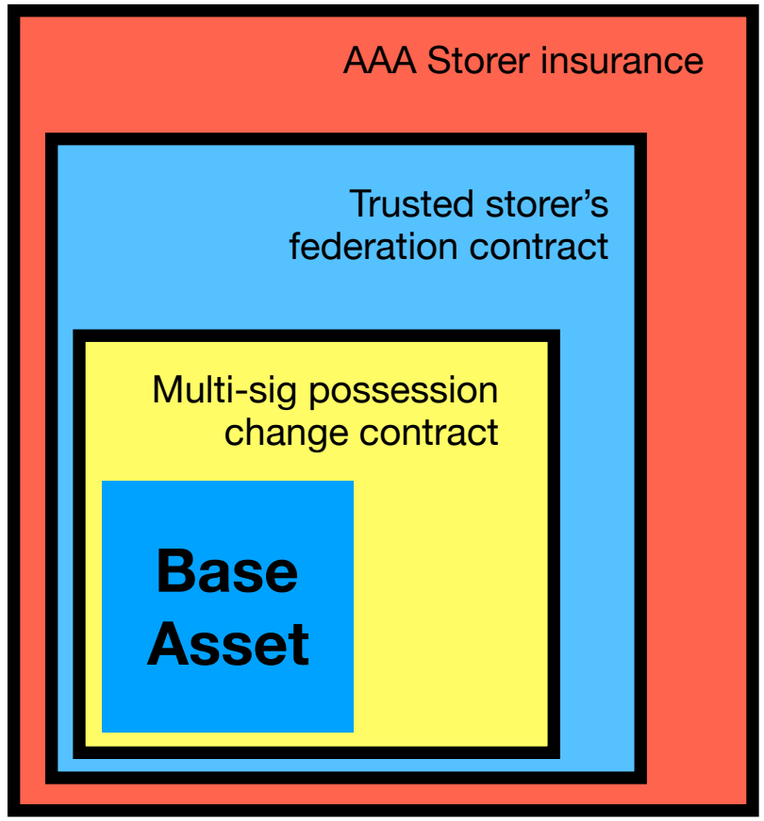
As things like this developed, we can imagine certain addresses in the Komerka contract becoming famous:

This minter is Christie’s auction house.

This storer is Amazon Services.

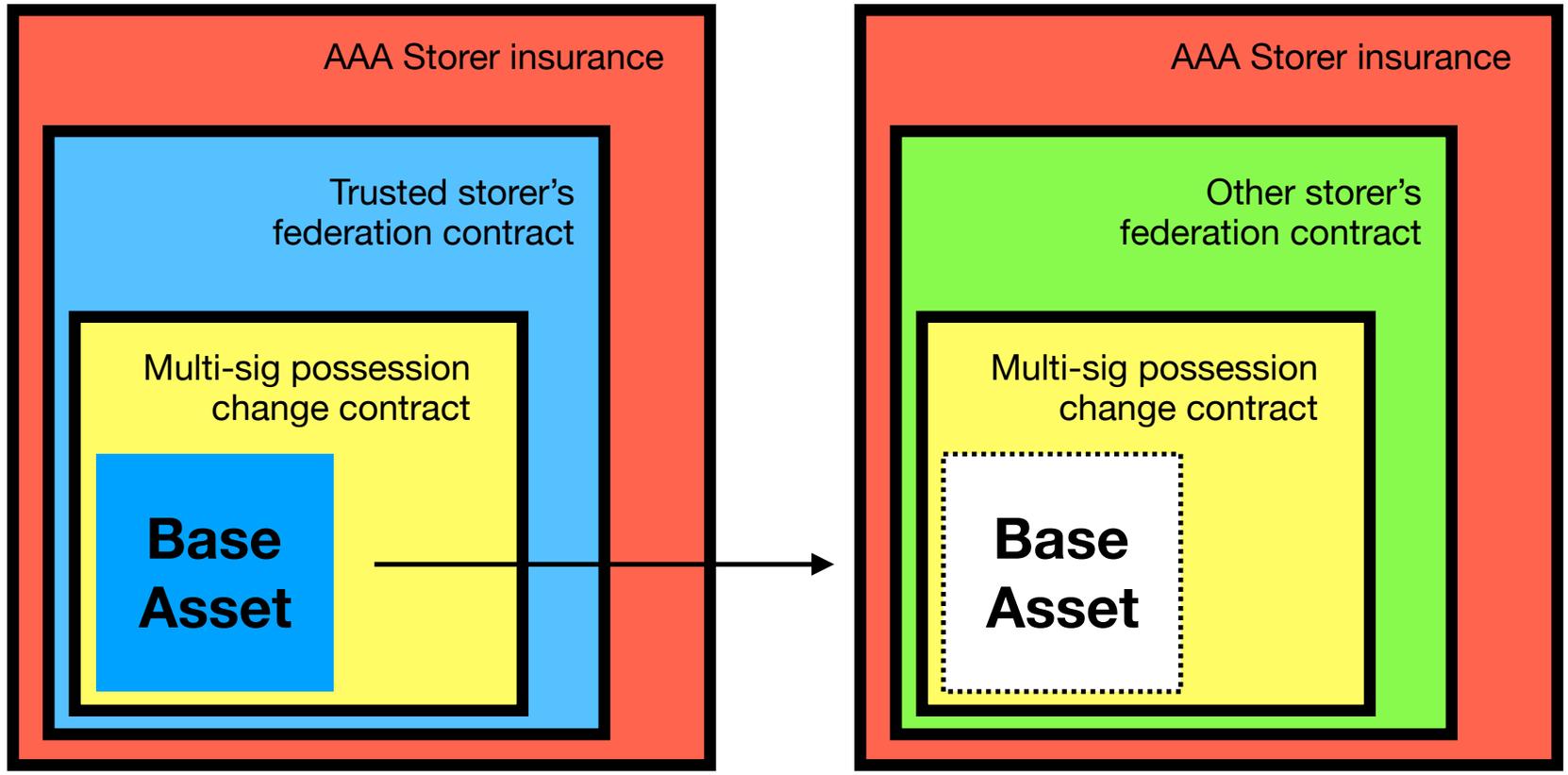
We can also imagine layer 3 applications: e.g. insurance contracts that depend on you have your goods stored with this federation, etc.

Assets might end up wrapped in layers of contracts and services.



And this is why I think it is defensible.

Customers might demand a sort of “backwards compatibility” with their assets. They’d want to be able to unwrap these layers and enter into new service agreements, while still using the same base asset.



Of course, if early on there was a hostile storage federation that had 99% of users, they could probably leave the Komerka Contract without consequence.

But everything in crypto has this risk. E.g. Ethereum successes moving to Polkadot.

Again, I think our risk could be mitigated by:
- Early layer 2 successes are friendly to us, i.e. probably built by us
- Our team continues to build the best tools, which are not friendly to forks
- Value capture in the Komerka contract is tolerable

So what's the value capture means?

I’m not sure yet. I would like to do a token, just because I see them to be such a valuable tool in building a community and funding.

My current idea is a mix of things. I think the Komerka contract could charge something small, like 10 Gwei, on all mint events.

50% of this would go to an admin address and 50% would go to token holders.

Tokens could be used for governance.

An alt variant is that there are no fees, or reduced fees, if you hold the token.