AutoSwarm NFTs Stamper (2/3)

Empowering Eternal Digital Ownership

7th November 2023

AutoSwarm Schedule (3 grants)

1/3 : AutoSwarm PoC (done)

AutoSwarm demo via One Page example : Topping Up an existing NFT by One year.

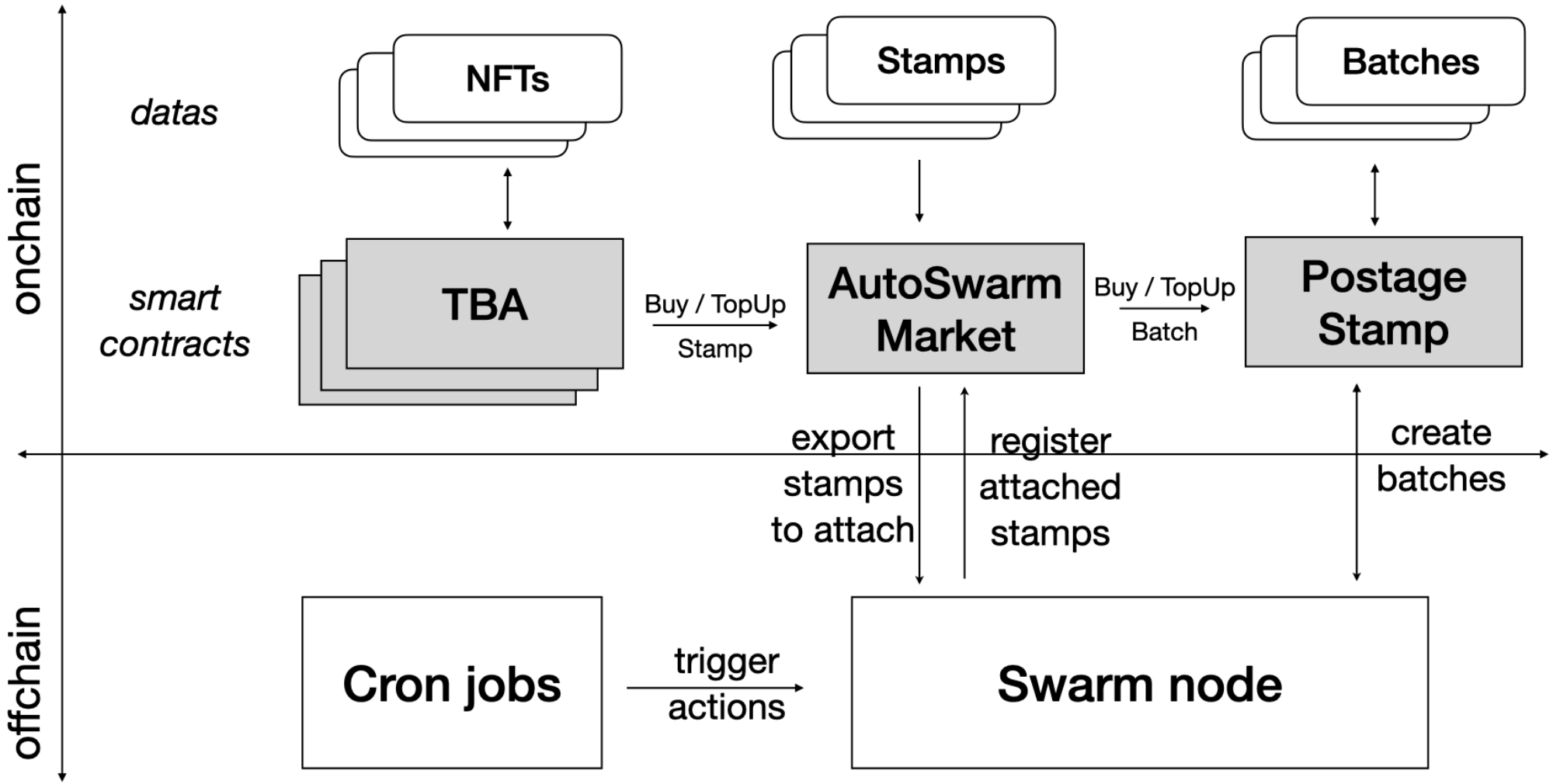
2/3 : AutoSwarm Stamper (this grant)

PoC prooved the need to manage Stamps one by one, this is the purpose of this development, results in same demo with Stamp TopUp instead of Batch TopUp

3/3 : AutoSwarm ReSaver (next)

To be able to manage -- to AutoSwarm -- whatever NFTs, this means re-store on Swarm any existing NFTs, to be able then to ToPup them

AutoSwarm Architecture



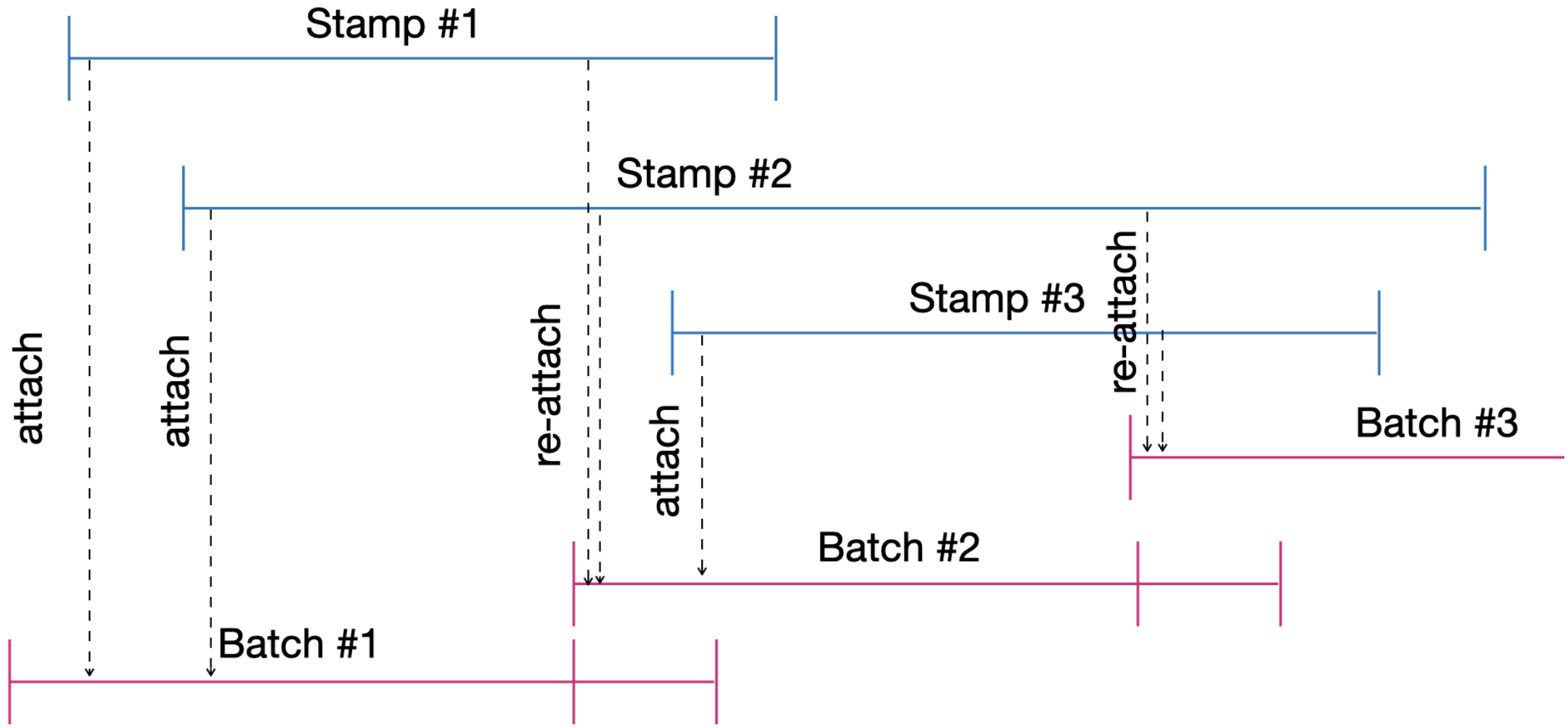
AutoSwarm Market

PostageStamp : ***manage Batch of Stamps with depth and ttl***

AutoSwarm Market : ***manage Stamps with size and ttl***

Stamps created are attached to current batch
then re-attach on new batches (if not expired)

AutoSwarm Stamps & Batch Timeline



Onchain : AutoSwarm Market Smartcontract

Done

- 3 versions to get the final working one, with Architecture decisions
- similar mechanism than PostageStamp

ToDo *during next grant*

- use same sorted list structure than PostageStamp
  not mandatory in testing phase, but needed to scale number of stamps

Offchain : AutoSwarm cron jobs

Done

- One time manual attachment of Stamp

ToDo :

- offchain attachment of Stamps to batches : to be done within AutoSwarm Re-Saver : same scripts to develop
- cron jobs :
    - one each day to trigger attachement on new Stamps,
    - one each month (on new Batch creation) to trigger re-attachement on new Batch

Annexes

Architecture decisions :

- One Stamp per NFT, containing NFT metadata and content and maybe other assets
- One Year Stamp by default (365 days precisely),
- One Month Batch by default (30 days precisely)
- One Week overlapping time between two Batches (7 days)
- Stamps attached to one Batch (two during overlapping time), changed during overlapping period (if Stamp not expired)
- One Batch at a time (or two during overlapping period)
- Stamps use same remaining balance mechanism as Batches, but with size and ttl, instead of depth and ttl

PostageStamp vs AutoSwarm Market

PostageStamp : *manage Batch of Stamps with depth and ttl*

- topUp Batch => more ttl / same depth
- dilute Batch => more depth / less ttl
- extends Batch *added* => more depth / same ttl

AutoSwarm Market : *manage Stamps with size and ttl*

- topUp Stamp => for immutable NFT / fixed size
- extends Stamp *later*, for mutable NFT or any other useCase, any Swarm file

AutoSwarm Stamps & Batch Timeline

1. Offchain : **Create new Batch** on a Swarm node
2. Onchain : **Register Batch** on AutoSwarm Market
3. Onchain : **Export Stamps** to attach from AutoSwarm Market
4. Offchain : **Attach Stamps** (not expiring) Stamps to the new Batch on a Swarm node
5. Onchain : **Register Stamps** on this Batchid after attachement
6. Offchain : **Wait** one month
7. Offchain : **Back to 1.**

AutoSwarm Stamp Creation Workflow

0. Offchain : **Choose NFT**
1. Onchain : **Create TBA** (NFT Token Bound Created TBA)
2. Onchain : **Send Bzz** to TBA
3. Onchain : **TopUp NFT** => creates a Stamp to store NFT metadata
4. Offchain : **Attach Stamp** to the Batch on a Swarm node
5. Onchain : **Register Stamp** on this Batchid after attachement

Grant Proposals

« AutoSwarm Stamper » Grant Proposal

After our experience with the AutoSwarm Proof of Concept (PoC), we've recognized a **critical lesson**: managing NFT content through **a single batch of stamps falls short** of needed requirements. When multiple NFTs are associated with a batch stamps, any action taken on one affects all others. Consequently, it has become evident that we need to pivot from a sort of wholesale approach to a retail one, specifically focusing on the individual **management of unit stamp** instead of batch of stamps.

Our proposal is to refine the management of stamps at the NFT level, via an onchain « **AutoSwarm Market** » empowering **each NFT to have its own unique stamps**. The "AutoSwarm Market" will then leverage batches of stamps and distribute them individually to NFTs.

This "AutoSwarm Stamper" sub-project aligns with a **Grant budget of $10,000**. It forms an integral part of AutoSwarm, a one-page service envisioned as a **Public Good**, supported by the **Web3 community**. Through this service, both existing and new NFTs will gain the ability to **securely store over the long term** their metadata and content.

Milestones from T0, start of the project, assuming 1/10/2023

AutoSwarm Specification (pdf)
- Architecture specification update
- Market Specification: stamps management rules, unit price calculation

T0 + 1 weeks = 8/10/2023

AutoSwarm Smart Contract v0.2 (stack solidity/foudry):
- AutoSwarm SmartContract update - NFT token bound account (TBA)
- New AutoSwarmMarket SmartContract

T0 + 3 weeks = 15/10/2023

AutoSwarm OnePage Dapp v0.2 (stack sveltekit/typescript/viem):
- OnePage NFT topping of unit stamp

T0 + 4 weeks = 29/10/2023

Code repo on GitHub: https://github.com/Kredeum/AutoSwarm

« AutoSwarm ReSaver » Grant Proposal

After our initial experience with the AutoSwarm Proof of Concept (PoC), it became evident that we could significantly expand the project's scope:

*Topping one NFT on Swarm is fine, but **what about all the other NFTs** ?*

Many NFTs are currently stored on decentralized storage solutions, while others reside in **less ideal locations like Google Drive or Amazon S3**.

The core concept behind this project is to **securely migrate NFT metadata and content onto the Swarm storage**. AutoSwarm ReSaver, our proposed solution, aims to offer a **second life to NFTs**, particularly those currently stored on centralized platforms.

This could be done, by **reusing sone components** of our existing Kredeum NFTs plugin, that already integrates NFT minting on Swarm.

We also plan to explore the possibility of **launching an EIP** (Ethereum Improvement Proposal) related to this matter. Since NFT SmartContracts are inherently tied to their owners' NFTs, this will enable wallets to recover NFTs from centralized storage in the event of unforeseen disasters.

Milestones from T0, start of the project, assuming 1/11/2023

AutoSwarm Specification (pdf)
- Architecture specification update
- UI design , somewhat similar to our our Kredeum Dapp but dedicated to a one page service

T0 + 2 weeks = 15/11/2023

AutoSwarm OnePage Dapp v1.0 (stack sveltekit/typescript/viem):
- NFT picker (from you own NFT, from a specific collection or network)
- OnePage AutoSwarm Dapp

T0 + 4 weeks = 29/11/2023

Code repo on GitHub: https://github.com/Kredeum/AutoSwarm