Pareto prototype concept

Rel. 1.0
by Roland
22.3.2024

Executive summary

This concept is based on the Pareto project paper v.1, my recommendations in Pareto v.1 feedback and the discussion with Milosz in Gotha.

There are three essential project goals:

- censorship resistant content hosting
- collaboration
- truth approximation

To achieve a prototype (MVP) asap we suggest to focus on goal one.
This includes the content creation, storage and distribution.
With these package it is possible to showcase the whole process (with existing Nostr clients and platforms) and highlight the Pareto usp (censorship resistance).

We estimate the production of that prototype within three month and a budget of less than 50k (good case scenario).

Table of content

1. IAM

User ID for

- login to services
- profiling
- charging
- searching, filtering and grouping on Nostr

Nostr-address with own domain for Pareto → xxx@pareto.???

LN-address with own domain for Pareto → xxx@pareto.???
(self-custodial or managed)

2. Creation

Product
kind:30023 event (json format)
kind:1 event for distribution on other SM clients (Mastodon, Bluesky + Threads with

https://github.com/nostr-protocol/nips/blob/master/23.md
This NIP defines `kind:30023` (a *parameterized replaceable event*) for long-form text content, generally referred to as "articles" or "blog posts". `kind:30024` has the same structure as `kind:30023` and is used to save long form drafts.
"Social" clients that deal primarily with `kind:1` notes should not be expected to implement this NIP.

Censorship resistant multimedia content (via links) must be stored in decentralized storages (see storage → Fediverse / Blossom Drive)

Samples

https://yakihonne.com/write
https://habla.news/de/write

OS-Tools - MD editors with NIP23 format export

https://opensource.com/article/21/10/markdown-editors
https://github.com/mundimark/awesome-markdown-editors

MarkText, Typora, VSCodium, Markor, StackEdit, Boost Note, and UberWriter.
0 UberWriter is another open-source Markdown editor that has a centered UI and restricts word length in a row to increase content visibility.
1 VSCodium is a free open-source code release of Microsoft's VSCode editor that can function as a markdown editor.
2 MarkText is an MIT licensed open-source project that is still in development and can be downloaded for free from the GitHub release page.
3 Typora removes unnecessary distractions and provides a real live preview feature to help users concentrate on the content.
4 Markor is an OS text editor for Android that utilizes simple markup formats like Markdown and todo.txt for note-taking and list management

"Inspired by the design and UI/UX of apps like Notion, and utility of open-source apps like StackEdit, I decided to create a minimalistic, local-only WYSIWYG Markdown editor.

Some features worth highlighting:

Monaco editor and Prettier integration for code snippets
Tables (apparently the holy grail of WYSIWYG editing)
Embeds (for CodePen, CodeSandbox and YouTube, most useful for HTML or JSON exports)
Accepts Markdown paste-in, and "exports"/generates HTML, Markdown and JSON outputs
Collaboration (with real-time awareness and initial commenting system, available only when logged in)
GPT-3.5 integration (only when logged-in with the corresponding extension installed)
Stack used: TipTap, ProseMirror, Solid.js, HocusPocus, Fastify, tRPC.

Some notable drawbacks:

Not best mobile support
Collaboration available only between signed-in users, in the same workspace;
I tried my best to support most common MD formatting, pasting and in-editor shortcuts, though there might still be room for improvement
Self-hosting isn't easy right now, though you should be able to figure it out from the source code

The editor itself is a standalone app, extracted from the larger Vrite CMS project (https://github.com/vriteio/vrite) which you can also test out (only with sign-in) here: https://app.vrite.io/" https://docs.vrite.io/getting-started/introduction/

3. Storage

Product

a. for notes
naddr1qq2kw52htue8wez8wd9nj36pwucyx33hwsmrgq3qyzvxlwp7wawed5vgefwfmugvumtp8c8t0etk3g8sky4n0ndvyxesxpqqqp65w6998qf

b. URL for multimedia content
https://peertube.tv/w/qiD84Z88kCruCVUZSpuccy
blossom:ef1c26172f55017c9d9d6afa7cf22605b237b0fe92425e81e3b5e24d46c95
(each client can choose how (HTTP, torrent, I2P, etc.) and where (public servers, private servers, etc.) to retrieve it from)

Solutions

a. Federated solutions based on ActivityPub (Fediverse)

Video PeerTube (Version 6.0 → mit Android-App)
Features: Apps / Account / Downloads / Playlists / Search / Chapter / Transscriptions / Moderation / Donations / Podcast 2.0 / Value4value /

Audio PeerTube (ab 6.0) oder Funkwhale
Features: Apps / Account / Downloads / Playlists / Search / Chapter / Transscriptions / Moderation / Donations / Podcast 2.0 / Value4value /
https://www.funkwhale.audio/ https://blog.funkwhale.audio/

Bilder Pixelfed
Features: Albums / Collections / Stories / Filters / Comments (Moderation?) / Notifications / Hashtags / Likes / Shares /
https://pixelfed.org/ https://pixelfed.de

b. Blossom Drive or other decentralized solutions

Insights based on discussions about torrents, hypercore, nostr relays, and IPFS. There were a few things I learned from all these conversations:

1. All the existing solutions have one thing in common. A universal ID of some kind for files
2. HTTP is still good. we don't have to throw the baby out with the bath water

Blossom Drive Blobs stored simply on media servers Introduction

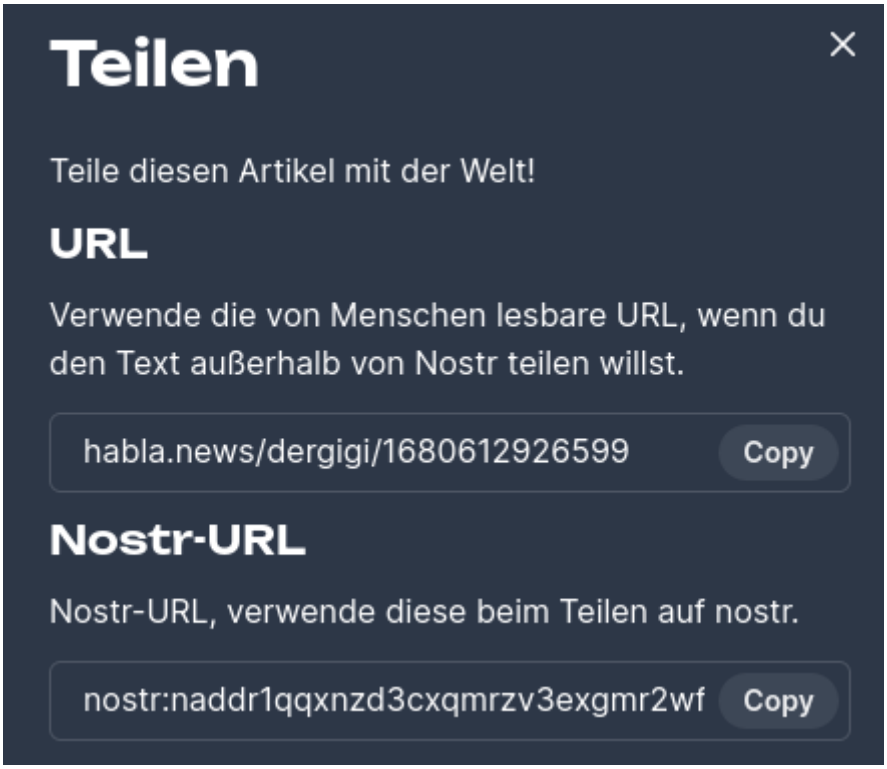
Blossom is a set of HTTP endpoints that allow nostr users to store and retrieve binary data on public servers using the sha256 hash as a universal id. Blossom Drive is a nostr app built on top of blossom servers and allows users to create and manage folders of blobs.
Drives are just nostr events (kind 30563) that store a map of blobs and what filename they should have along with some extra metadata.

Blossom is basically copying what nostr did for notes and applies it to arbitrary files. Instead of relays handling events, there's simple HTTP servers handling files. Like relays, servers are interchangeable as they share the same interface, encouraging duplication and redundancy. Instead of uploading something to a single server, you might upload it to five different servers.
Gone are the days of finding a thing and uploading again. You just need the hash, and the thing will appear. You could even insert images directly in notes with something like a blossom:ef1c26172f55017c9d9d6afa7cf22605b237b0fe92425e81e3b5e24d46c95448 and each client can choose how (HTTP, torrent, I2P, etc.) and where (public servers, private servers, etc.) to retrieve it from.

4. Distribution

a. Standard via URL https://habla.news/dergigi/1680612926599

b. Censorship resistant
nostr:naddr1qqxnzd3cxqmrzv3exgmr2wfeqgsxu35yyt0mwjjh8pcz4zprhxegz69t4wr9t74vk6zne58wzh0waycrqsqqqa28pjfdhz



share options

- inside the nostr universe
- via https://openvibe.social/ (Nostr, Mastodon, Bluesky, Threads)
- via other SM clients
- via mail
- on websites

5. Monetization

LN-address for users with own domain for Pareto → xxx@pareto.???
(self-custodial or managed)

tips / zaps in social media apps (LNTipBot in TG / Bot in Discord / other solutions?)

zaps in all nostr clients (via NWC)

more options on the blogspace

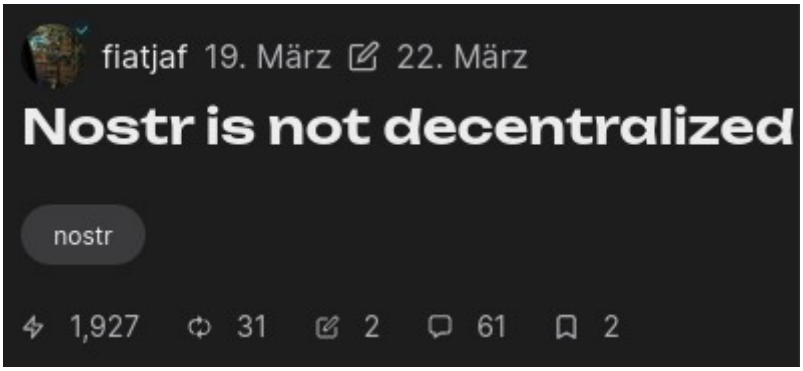
- abo model
- paywall (for pages and / or articles)
- streaming
- split payments

6. Interaction

Client side

- like
- vote up / down
- comment
- rewrite
- bookmark
- zap

Sample Habla.news



7. Blogspace

Habla

https://habla.news/	only Browser?	
https://habla.news/de/c	Communities	z.B. Bitcoin
https://habla.news/de/write	Editor	
https://habla.news/de/faq	FAQ	

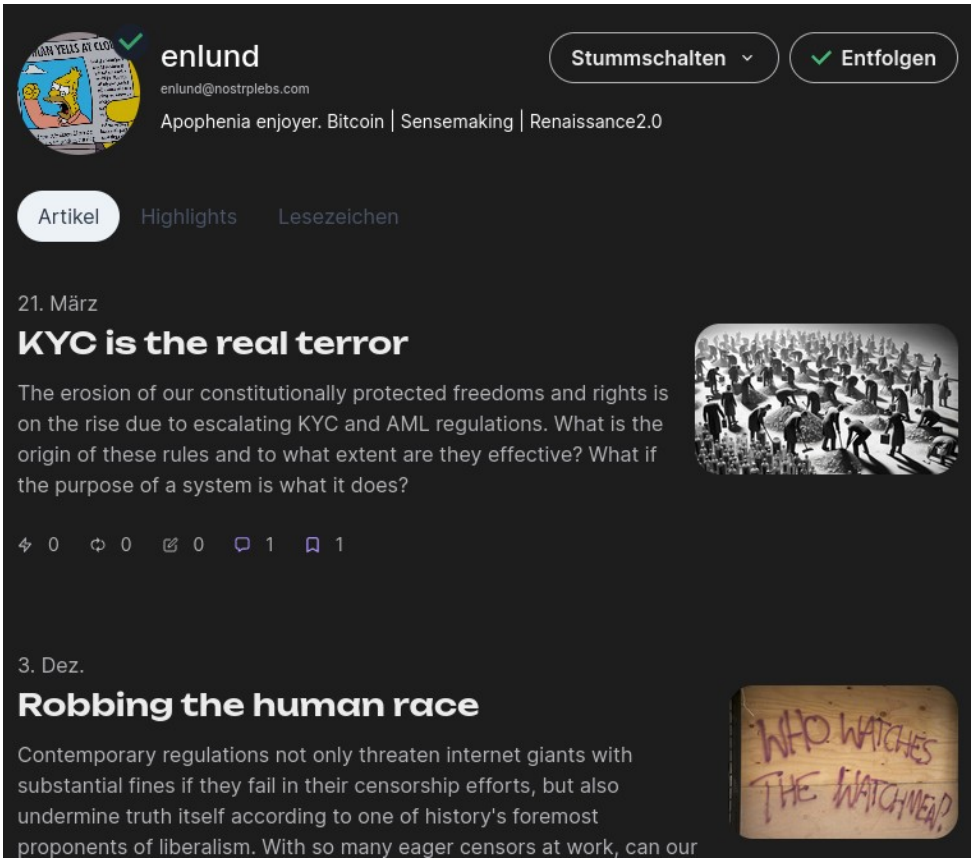
YakiHonne

https://yakihonne.com/	Browser and mobile apps

Two pillars of decentralized media
→ Decentralized publication (storage) / Decentralized content moderation

Writes / Comments / Ranking
Flash News + Uncensored Notes
(permissionless community notes → better than X community notes)

Individual author pages → filter on NIP05 enlund@nostrplebs.com or filter npub



8. Client

own client (forked) or integration in existing platforms?

quick success via communities https://habla.news/de/c z.B. Bitcoin
or via curations (at YakiHonne) z.B. Austrian economics

Clients (Web, PWA, iOS, Android)
- content presentation (notes, images, audio, video)
- likes, votes, zaps, comments, validation (community notes), communities / curations
- link forwarding → to SM clients (content pushing)