BLOCKCHAIN COMMONS

PROVENANCE MARKS

SOURCES

- BCR-2025-001
  Provenance Marks: An Innovative Approach for Authenticity Verification
  - https://github.com/BlockchainCommons/Research/blob/master/papers/bcr-2025-001-provenance-mark.md
  - https://provemark.com

Scarcity + Provenance → Value

VALUE

- What is *value*?
  - Value is that which induces people to *trade*.
- Value is created by the intersection of:
  - **Scarcity** (limited supply) and
  - **Provenance** (verifiable authenticity).

VALUE

	❌ Provenance	✅ Provenance
❌ Scarcity	- Ambient air - Tap water - Mass-generated AI images	- Cryptographically signed Linux ISOs - Creative-Commons art - Google Fonts
✅ Scarcity	- Counterfeit handbags - Forged artwork - Ungraded trading cards	- Certified organic produce with traceability - Authentic luxury goods with serials - Museum-grade artwork with chain of custody - Bitcoin (fixed supply + public ledger)

When bits are never scarce, how does establishing provenance help creators manage scarcity?

Introducing Provenance Marks

WHAT ARE PROVENANCE MARKS?

- “Smart Serial Numbers”
- Usable for physical or digital works
- Establish an object’s originating entity
    - *Not* current ownership, but we’ll come back to that
- Situate a work in a time-sequenced stream of works
- Can prove that a work has been unaltered
- Provide any other metadata

REQUIREMENTS

- Globally Unique
- Negligible cost to generate and verify
- Small and easy to handle
- Flexible and extensible
- Non-repudiable
- Easy for small creators to pick up and use
- Scalable to industrial applications

NO INFRASTRUCTURE NEEDED

- No public key
- No signatures
- No CA
- No global ledgers
- No expensive consensus algorithms

PRECONDITIONS

- A private cryptographic seed
  - Held by the originating entity
  - Carries the same protection concerns as any private key
- The originator's publicly published chain of marks
  - Can be sole-source
    - My chain is a GitHub Gist: https://provemark.com/wolf
  - The more widely published and copied the better

COMPARISON TO OTHER SYSTEMS

	Cryptographic Event Logs (CEL)	Coalition for Content Provenance and Authenticity (C2PA)	Provenance Marks (PM)
Primary purpose & first-class domain	Secure record-keeping for digital events, like software components and supply chain activities.	Tracks authenticity of media files (photos, videos, audio) for end-users.	Simple origin tracking for any digital or physical work, ideal for individual creators, scalable to industrial applications.
Integrity & trust architecture	Secure Merkle tree structure where operators sign checkpoints; users get proof their data was included.	Embeds history inside files with digital signatures; links changes through hashes without central database. Requires X.509 or OIDC certs.	Private cryptographic seed, forward-committed hash chain, published chain of marks, additional signatures and metadata optional.
Data-model complexity & extensibility	Simple core design that can hold any data; fixed proof format; extensible.	Complex standard (200 pages) with many data types; extensible.	Minimal but extensible structure.
Adoption footprint & implementation burden	Available in multiple programming languages; early testing stage; needs a trusted log operator, multiple witnesses recommended.	Used by major cameras, Adobe, TikTok, and Google; requires significant technical work to implement.	Simple command-line tool; works without servers; easy to implement.
Governance, licensing & maturity	In development with W3C Credentials Community Group aiming for Working Group in 2025; open-source license; led by open-source companies.	Managed by Linux Foundation; open-source license; mature spec; backed by major companies.	Published by Blockchain Commons; freely licensed; early development with focus on creator tools.

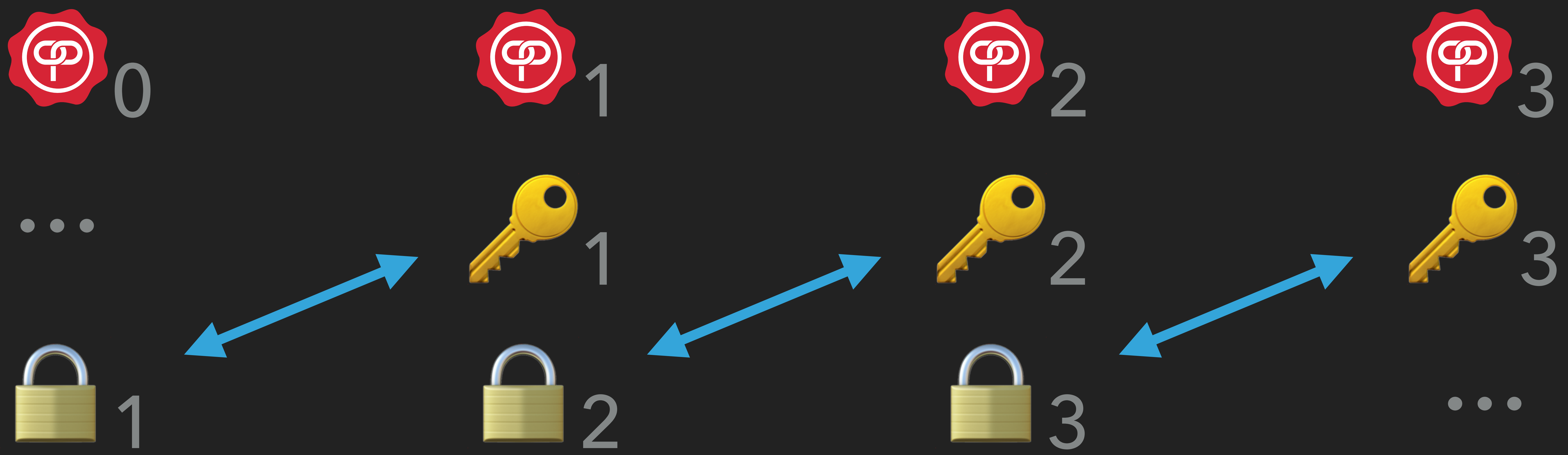
PROVENANCE MARKS & DECENTRALIZED IDENTITY

- Self-sovereign trust anchor
  - Enable DID URIs without reliance on external authorities or global ledgers.
- Forward-linked hash chains
  - Ensure tamper-evident continuity across DID Document versions.
  - Support simple revocation and rotation.
- Lightweight, non-repudiable cryptography
  - Uses a single private seed–no signatures, public keys, or consensus needed.
  - Efficient *O(1)* hash verification, ideal for mobile and low-resource environments.
- Flexible, decentralized storage
  - Provenance chains can live in Git, IPFS, or social media.
- Preimage resistance of SHA-256 is preserved under quantum attacks.
  - No need for post-quantum cryptographic algorithms.

How do Provenance Marks Work?

CORE CONCEPT: FORWARD HASH CHAIN

- Each mark:
  - Links itself into the chain by revealing a secret key pre-committed to in the previous mark
  - Pre-commits to the still-secret `nextKey` by publishing the `hash`, which includes in its image the `nextKey` and other data from the current mark



FORWARD HASH CHAIN: CONCEPTUAL LINEAGE

- 1981 – **Lamport OTP**: Each login reveals the next pre-image, irrevocably fixing the whole future chain
- 1995-97 – **S/Key / RFC2289**: Internet adaptation of Lamport with explicit counter in clear text
- 1996 – **PayWord micropayments**: Buyer signs chain root; successive pre-images serve as "coins," all pre-committed at purchase time.
- 2000 – **TESLA broadcast auth**: Each packet carries commitment, later disclosure authenticates every earlier packet.
- 2017 – **T/Key & modern OTPs**: single-secret, storage-efficient Lamport chains for two-factor auth.
- 2015 – **Gitcoin Blockchain commit-reveal**: on-chain hash commit, later reveal proves bids, randomness, etc.

CORE MECHANISM: PSEUDORANDOM NUMBER GENERATOR (PRNG)

- Purpose: generate the sequence of keys that is
    - Deterministic
    - Hard for attackers to predict
- Provenance Marks use the *Xoshiro256*** PRNG
    - Chosen for its speed, portability, and statistical quality
    - Does not have to be cryptographically strong because we *want* a deterministic sequence
- Originator holds:
    - A secret 32-byte seed (that *is* generated using a crypto-quality RNG)
    - Optionally, the current PRNG state
        - Makes generating the next key O(1)

ANATOMY OF A PROVENANCE MARK

key	hash	id	seq	date	info

- Binary structure with five mandatory, fixed length, ordered fields:
  - `key` Current output of the PRNG
  - `hash` SHA-256 of next mark’s PRNG output and the other fields of this mark
  - `id` Unique identifier of this chain
  - `seq` Sequence number of this mark within the chain, increases monotonically
  - `date` Date of mark generation, must be ≥ previous mark’s `date`
- May include optional sixth, variable-length field at the end:
  - `info` dCBOR data of any kind embedded in the mark

SIDEBAR: DCBOR

- dCBOR is conformant CBOR (RFC8949)
- ...with a few restrictions:
    - Numeric values all encoded in shortest form
    - Floating point values that *can* be encoded as integers *must* be
    - No NaNs with payloads *(Did you know NaN has "payloads"?)*
    - Map keys: sorted, no duplicates
    - No indefinite-length types
    - Only the "simple values" `true`, `false`, and `null` are allowed
    - Strings must be in Unicode Normalization Form C (NFC)
- Specified in IETF *draft-mcnally-deterministic-cbor*
- Reference implementations in Rust and Swift
- Third-party implementations available

GENESIS MARK

- The first mark establishes the basis of trust for a chain
- `key` The first bytes output by the PRNG after seeding
- `id` Same as `key`: you cannot choose your chain’s `id`
- `seq` Must be zero
- The genesis mark of a chain may be recognized by:

```
key == id && seq == 0 → true
```

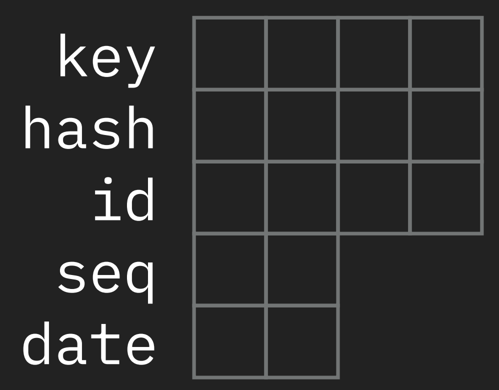
RESOLUTION

- Four *resolutions* provide size/security tradeoffs, analyzed in the white paper.
- Established at chain creation, applies to all marks in a chain

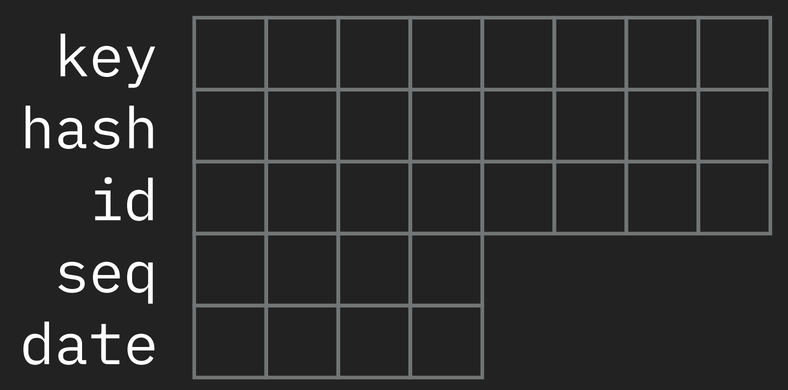
	linkLen	seqLen	dateLen	total
low	4	2	2	16
medium	8	4	4	32
quartile	16	4	6	58
high	32	4	6	106

```
total = 3 * linkLen + seqLen + dateLen
```

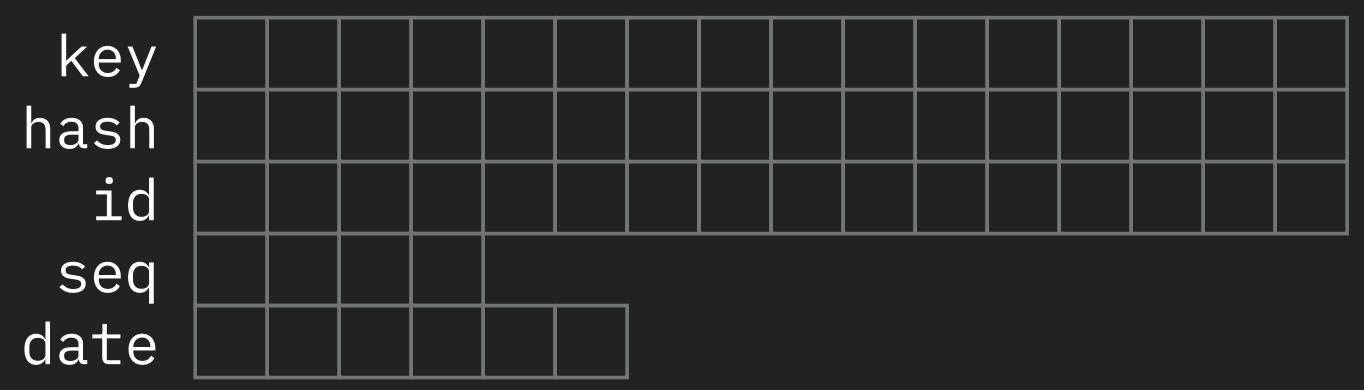
LOW - 16 bytes



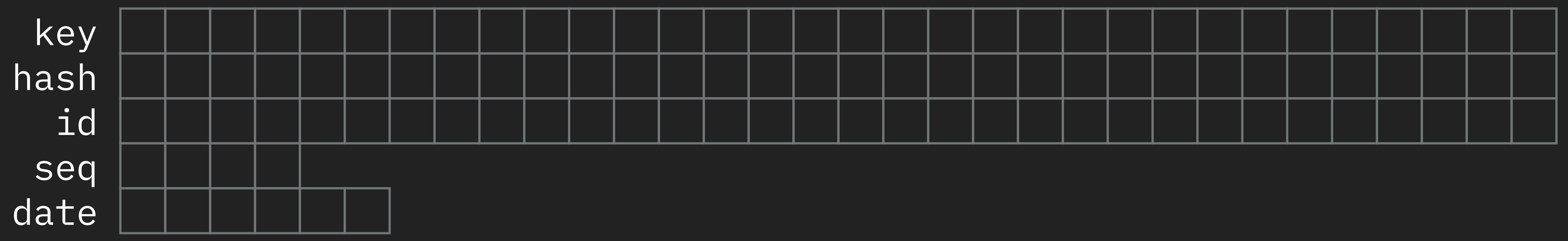
MEDIUM - 32 bytes



QUARTILE - 58 bytes



HIGH - 106 bytes



THE HASH

- The image for `hash` is formed by concatenating these fields in order:

`key` || `nextKey` || `id` || `seq` || `date` || `info`

- … then truncating the SHA-256 image of the digest to `linkLen` bytes
- `nextKey` is the pre-commitment to `key` in the next mark
- Anything included in `info` becomes bound to the mark

	linkLen	seqLen	dateLen	total
low	4	2	2	16
medium	8	4	4	32
quartile	16	4	6	58
high	32	4	6	106

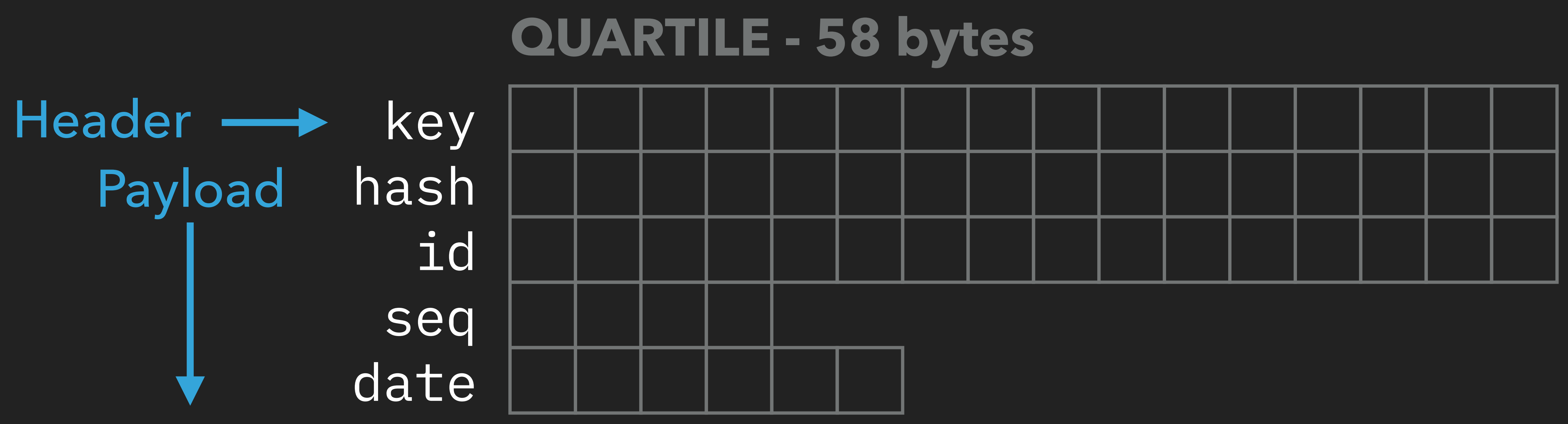
THE DATE

- `low`: `date` encoded as 16 bits
  - 1-day accuracy
  - Allows dates from 2013 to 2150
- `medium`: `date` encoded as 32 bits
  - 1-second accuracy
  - Allows dates from 2001 to 2137
- `quartile` and `high`: `date` encoded as 48 bits
  - 1-millisecond accuracy
  - Allows dates from 2001 to 9999

	`linkLen`	`seqLen`	`dateLen`	total
`low`	4	2	2	16
`medium`	8	4	4	32
`quartile`	16	4	6	58
`high`	32	4	6	106

- `date` must be ≥ `date` of previous mark
- Marks may have equal `date`, as long as `seq` increases

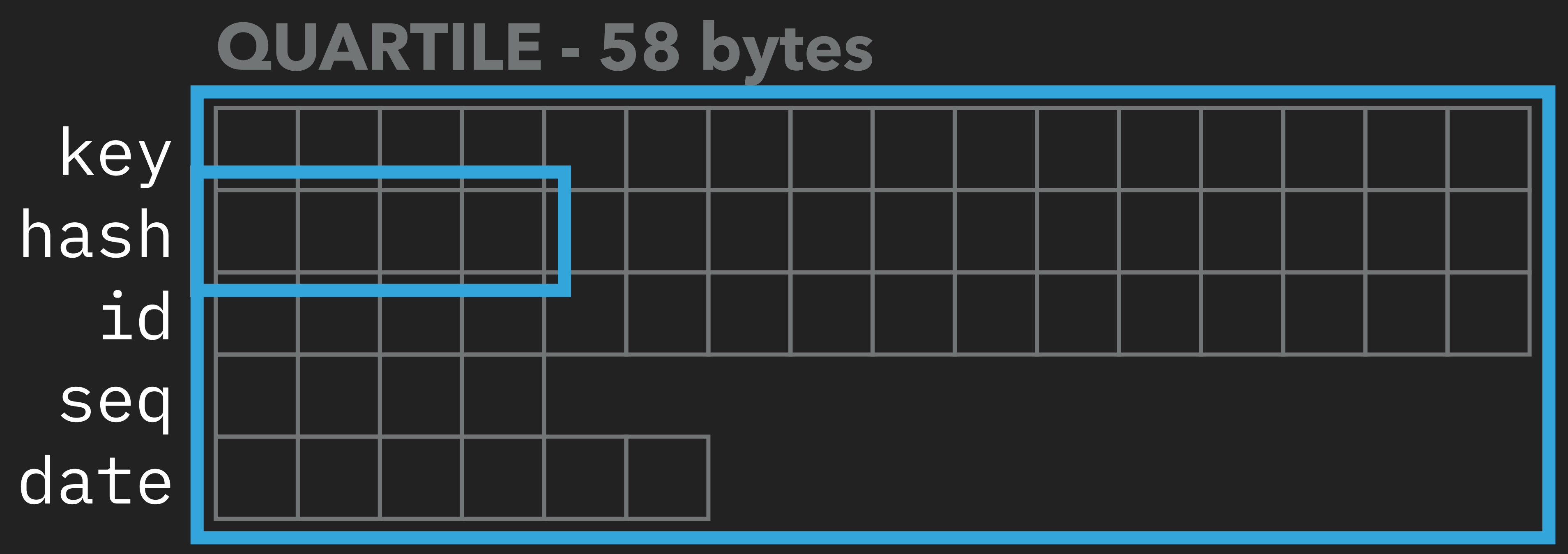
OBFUSCATION

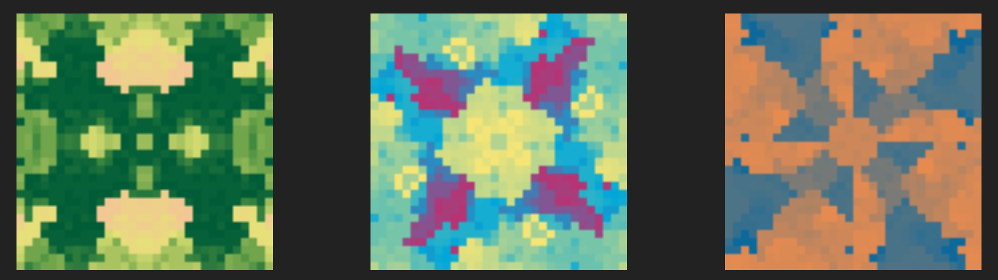


- When serialized, `key` is the structure `header`
  - Generated by PRNG, always statistically random
- The rest of the fields are `payload`
  - `id`, `seq`, `date`, `info` not statistically random
- `key` is not secret
  - used as symmetric key for *ChaCha20* cipher on `payload`
  - Transforms `payload` to be statistically random
  - Keeps message size constant
  - Adds a layer of error detection

HUMAN IDENTIFICATION

- Each provenance mark is globally unique, but statistically random
- Humans need quick ways to distinguish between and search for unique digital objects like marks
- The entire Provenance Mark can be used as input to a visual hashing algorithm like *LifeHash*.
- The first four bytes of `hash` can be used as a 32-bit identifier and converted to human-friendly forms that can be used as search keys:
  - *ByteWords*
  - *Bytemoji*





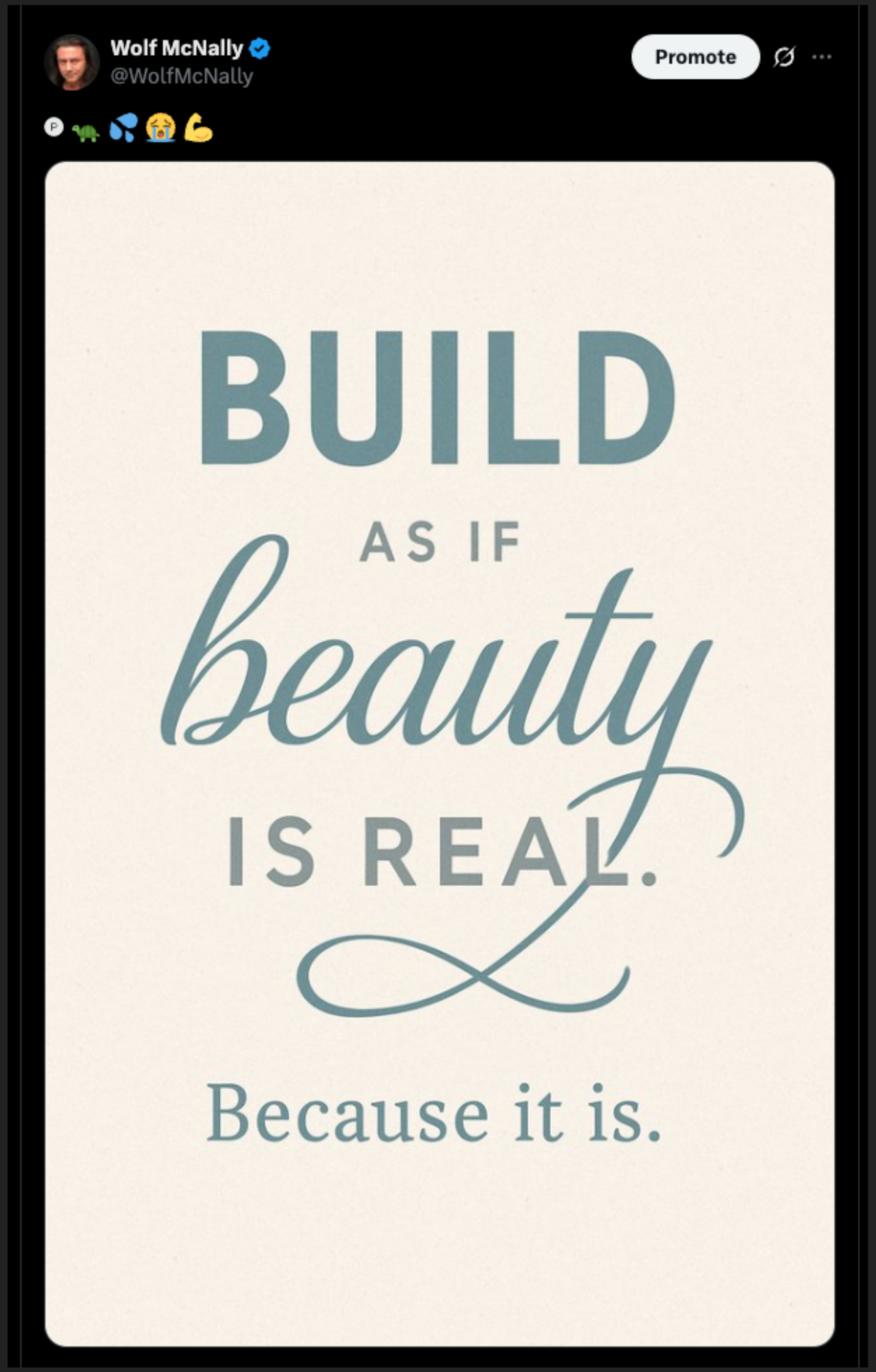
P KNOB BETA AQUA NOON
P SONG WALL RACE RICH
P KEYS VETO DRAW WORK

P 🍒 📫 🧵 🥑
P 🦄 🪼 💎 👕
P 📚 🥠 🌀 🫒

Examples

EXAMPLE 1

Post on 𝕏



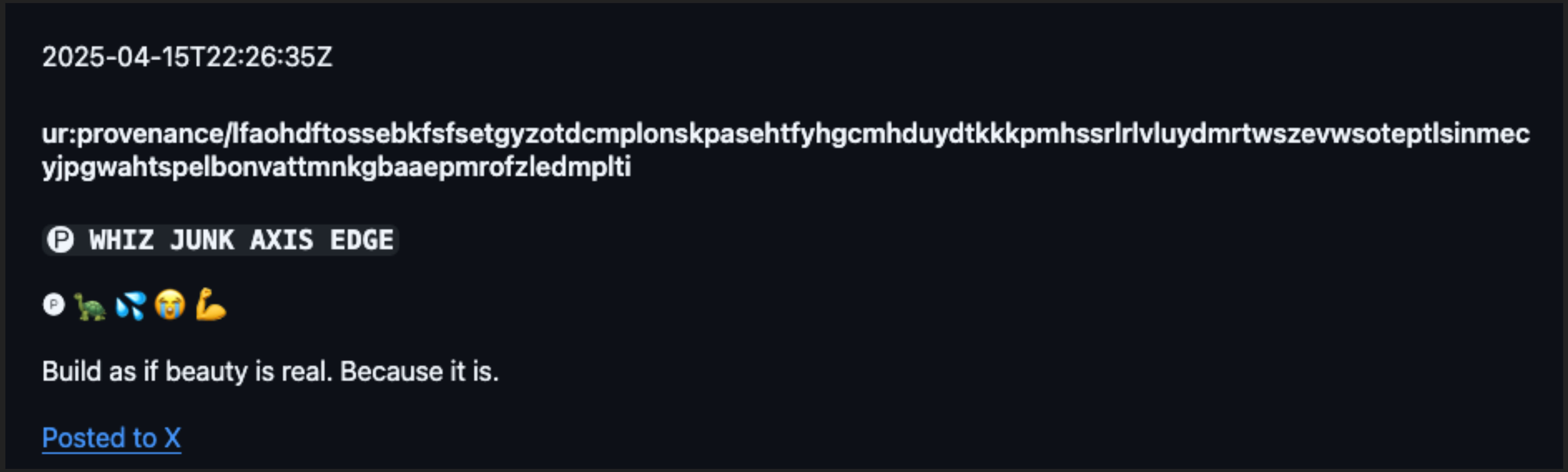
Wolf McNally
@WolfMcNally

Ⓟ 🐢 💦 😭 💪

Bytemoji mark identifier used as search key on https://provemark.com/wolf

Ⓟ 🐢 💦 😭 💪

Search Result



2025-04-15T22:26:35Z

ur:provenance/lfaohdftossebkfsfsetgyzotdcmplonskpasehtfyhgcmhduydtkkkpmhssrlrlvluydmrtwszevwsoteptlsinmecyjpgwahtspelbonvattmnkgbaaepmrofzledmplti

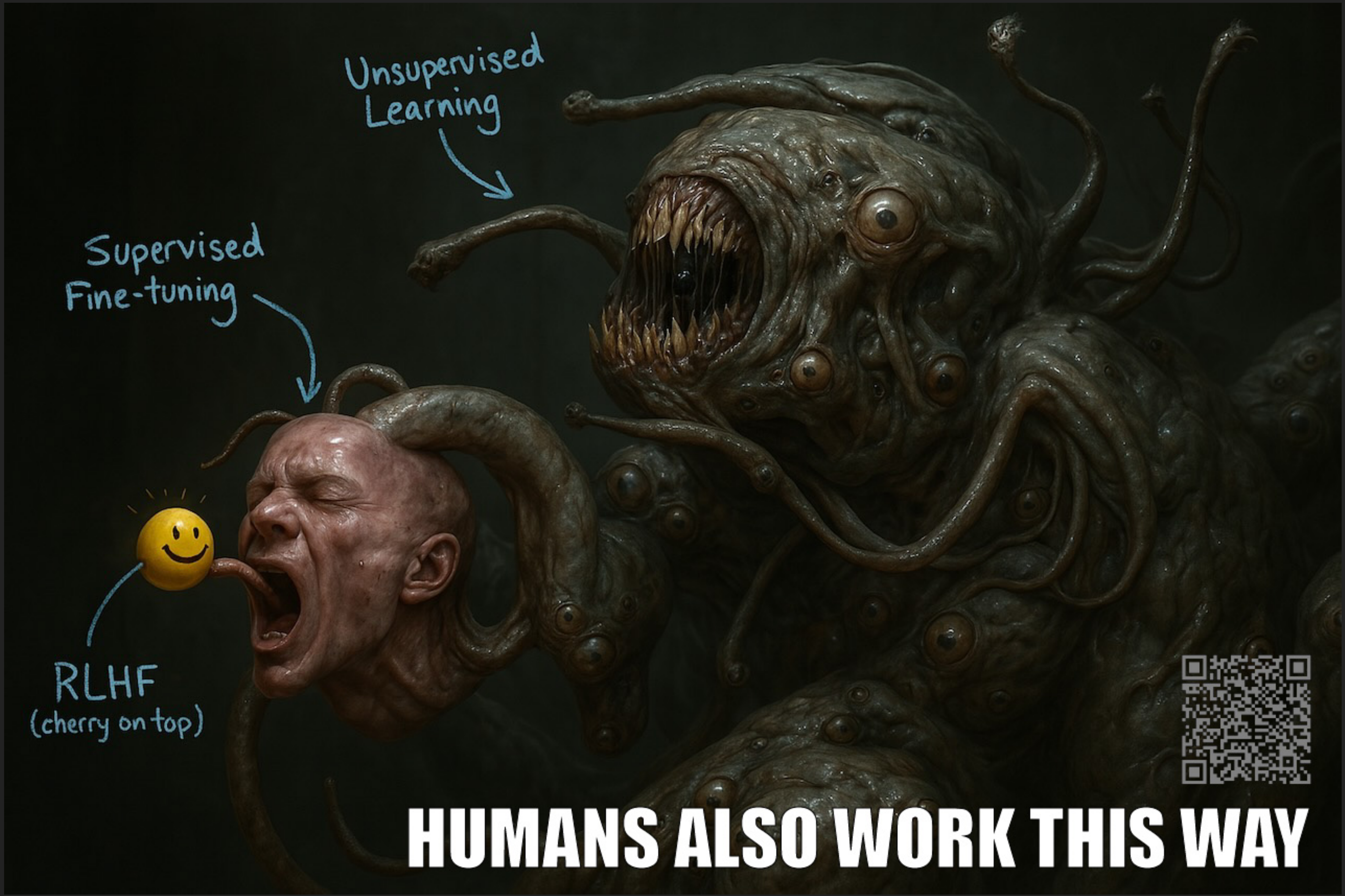
`Ⓟ WHIZ JUNK AXIS EDGE`

Ⓟ 🐢 💦 😭 💪

Build as if beauty is real. Because it is.

Posted to X

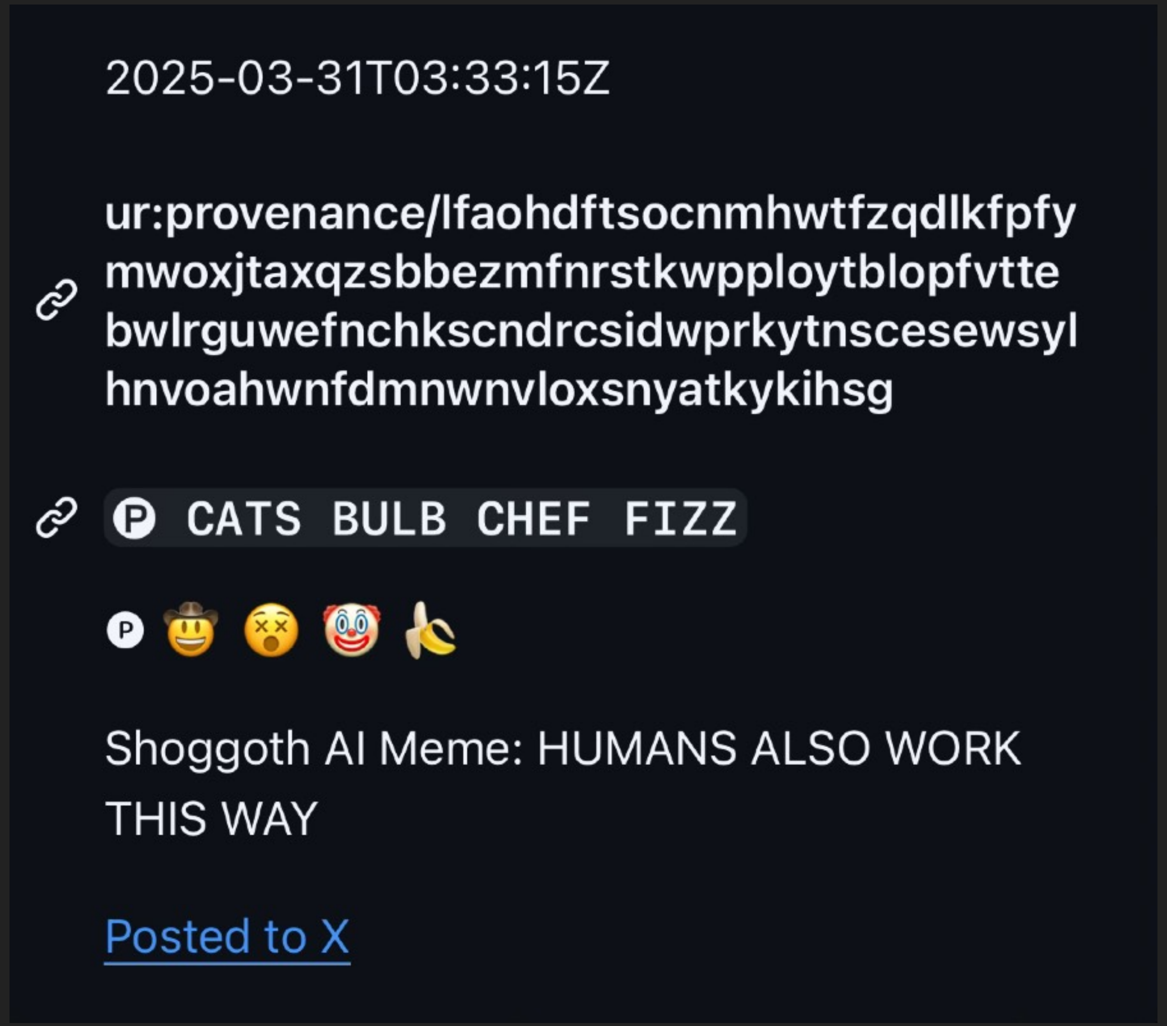
Original Artwork



URL in QR code

```
https://gist.github.com/
wolfmcnally/
86bce635a34fd991dce38e54869368e8
#Ⓟ-cats-bulb-chef-fizz
```

Link Target: ByteWords anchor



2025-03-31T03:33:15Z

ur:provenance/lfaohdftsocnmhwtfzqdlkfpfymwoxjtaxqzsbbezmfnrstkwpploytblopfvttebwlrguwefnchkscndrcsidwprkytnscesewsylhnvoahwnfdmnwnvloxsnyatkykihsg

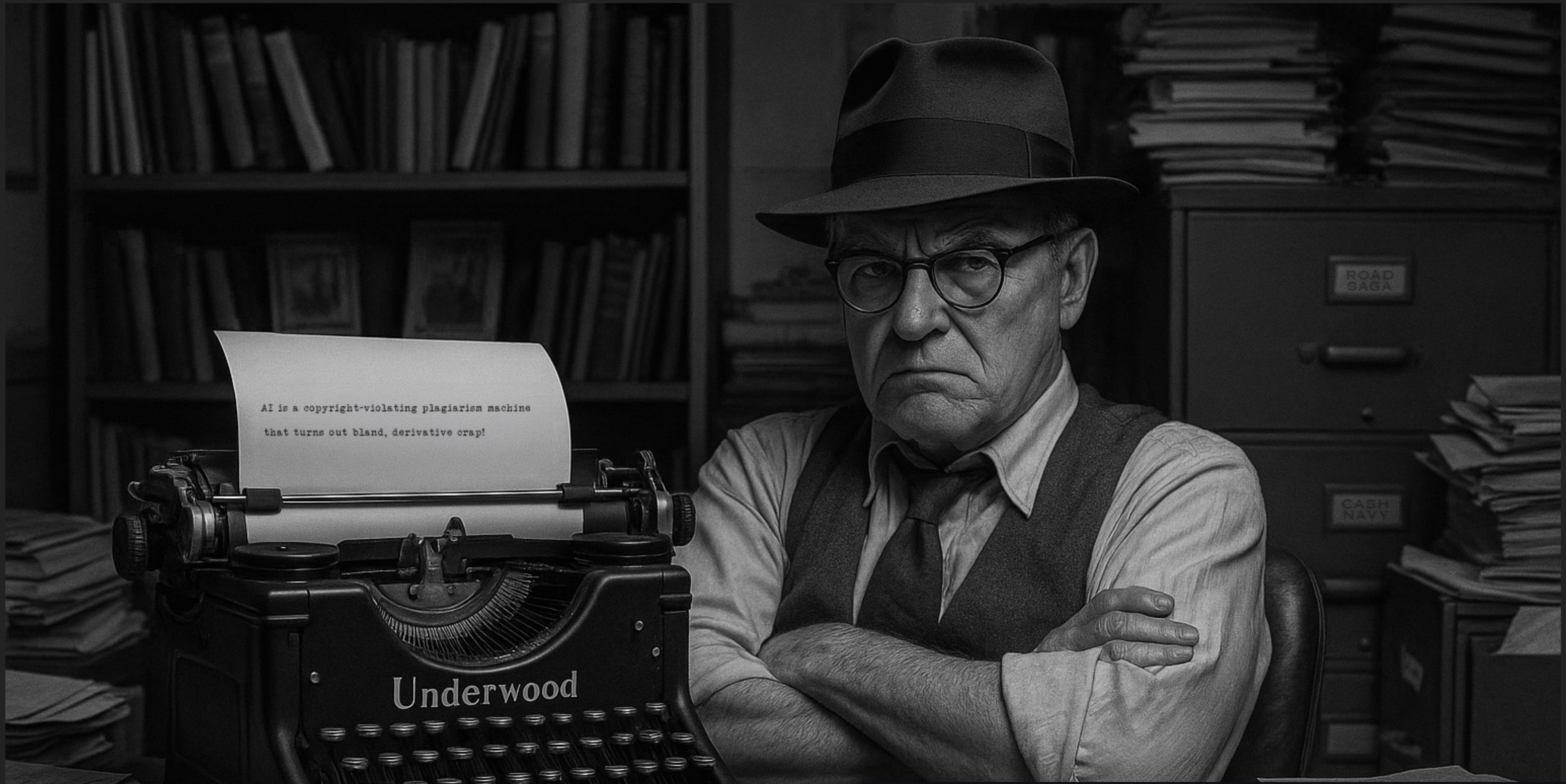
`Ⓟ CATS BULB CHEF FIZZ`

Ⓟ 🤠 😵 🤡 🍌

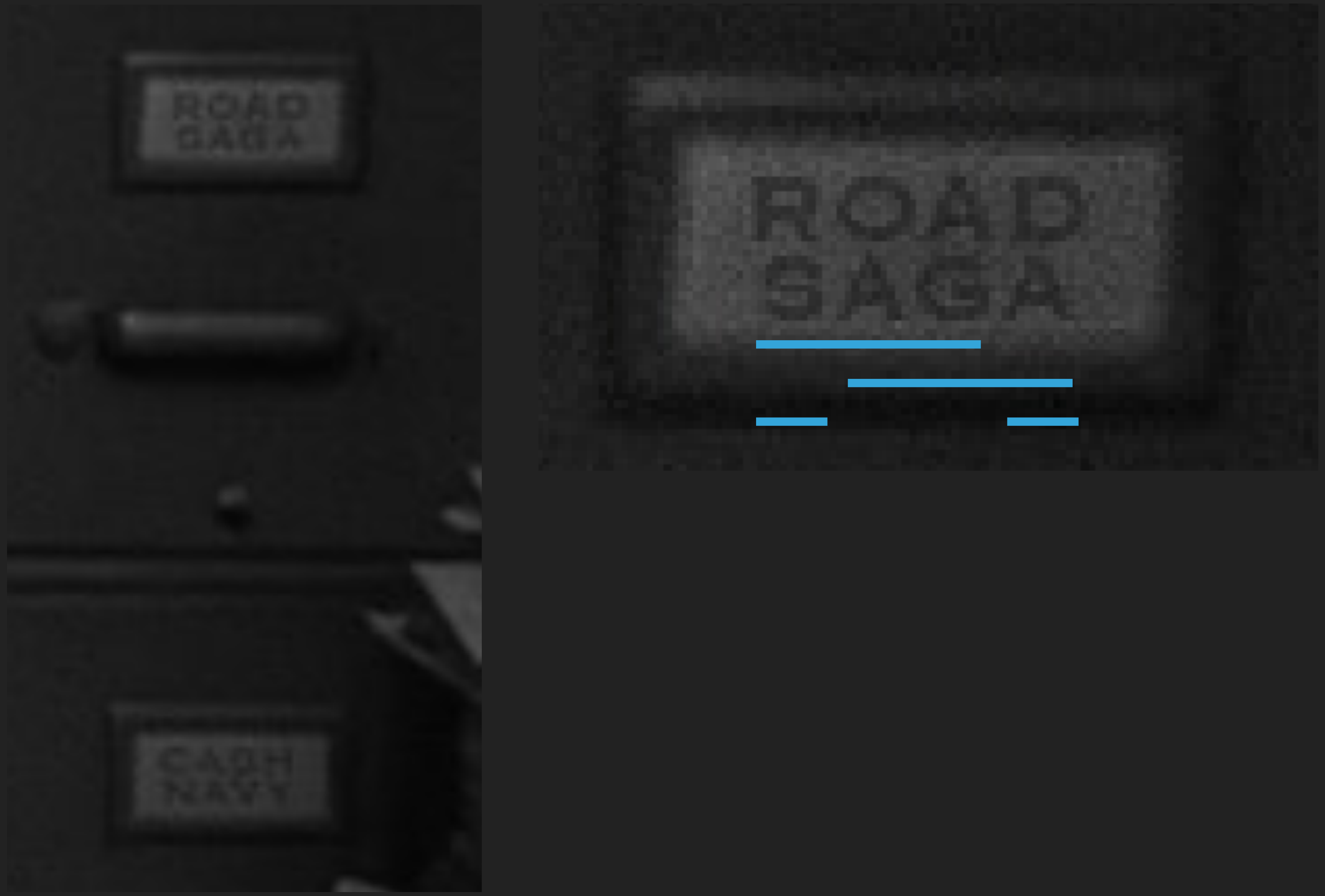
Shoggoth AI Meme: HUMANS ALSO WORK THIS WAY

Posted to X

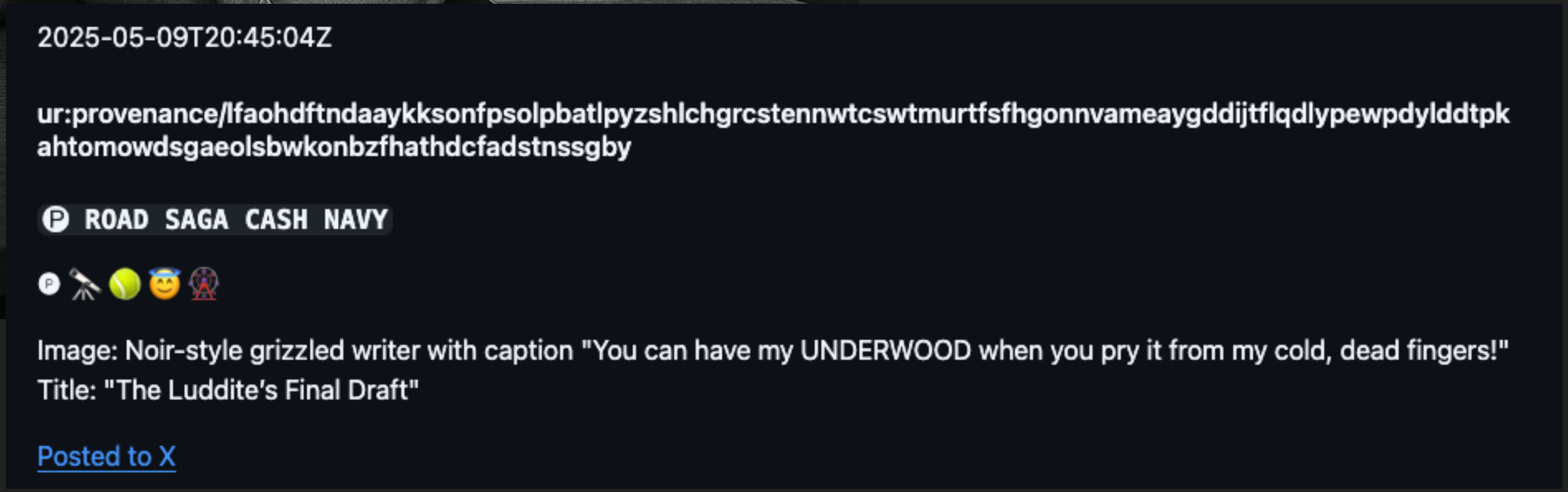
Original Artwork


AI is a copyright-violating plagiarism machine
that turns out bland, derivative crap!
Underwood
ROAD SAGA
CASH NAVY
YOU CAN HAVE MY
WHEN YOU PRY IT FROM MY

Detail: ByteWords Provenance Mark Identifier



Search Result



2025-05-09T20:45:04Z

ur:provenance/lfaohdftndaaykksonfpsolpbatlpyzshlchgrcstennwtcswtmurtfsfhgonnvameaygddijtflqdlypewpdylddtpkahtomowdsgaeolsbwkonbzfhathdcfadstnssgby

🅟 ROAD SAGA CASH NAVY

🅟 🔭🎾😇🎡

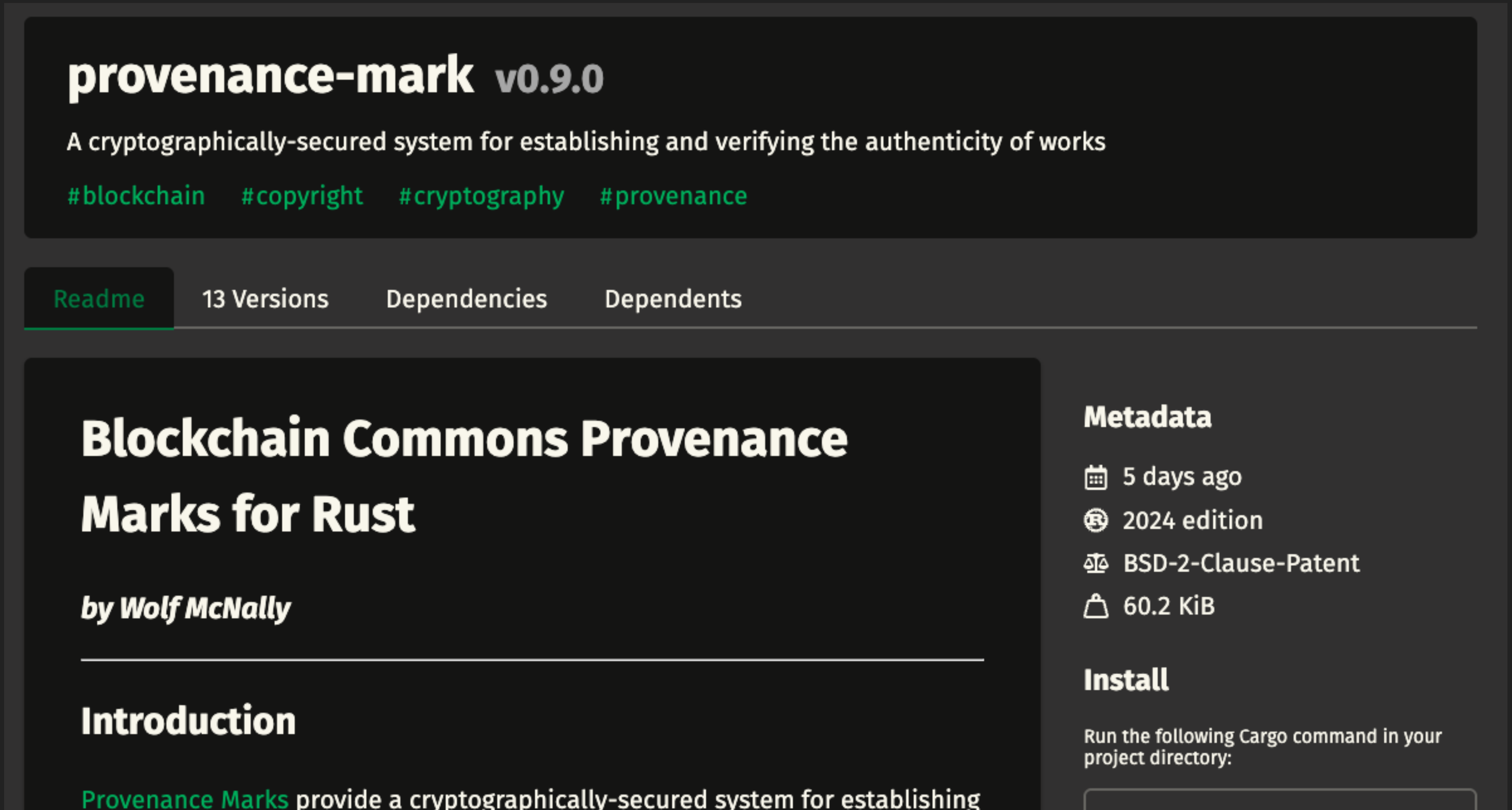
Image: Noir-style grizzled writer with caption "You can have my UNDERWOOD when you pry it from my cold, dead fingers!"
Title: "The Luddite's Final Draft"

Posted to X

Tooling

RUST REFERENCE IMPLEMENTATION

- https://crates.io/crates/provenance-mark



provenance-mark v0.9.0

A cryptographically-secured system for establishing and verifying the authenticity of works

#blockchain #copyright #cryptography #provenance

Blockchain Commons Provenance Marks for Rust

by Wolf McNally

Introduction

Provenance Marks provide a cryptographically-secured system for establishing

Metadata

- 5 days ago
- 2024 edition
- BSD-2-Clause-Patent
- 60.2 KiB

Install

Run the following Cargo command in your project directory:

COMMAND LINE TOOL

```
$ cargo install provenance-mark-cli

$ provenance new MyChain

Provenance mark chain created at: MyChain

Mark 0 written to: MyChain/marks/mark-0.json

---

2025-06-02T23:34:19Z

#### ur:provenance/
lfaohdftlpykeomechldjsbbmhchtdswaodafwtynstthkaytswensndeysfhpqdrdkbkosssonegsaecpdajtprisbefeoxylpaztgels
lsfwbnwtlkhybbrhstcprtzokn

#### `Ⓟ KING ZOOM DELI FLUX`

Ⓟ 🌎 🐳 😹 🍓

Genesis mark.
```

COMMAND LINE TOOL

```
$ provenance next --comment "My New Work" MyChain

Mark 1 written to: MyChain/marks/mark-1.json

---

2025-06-02T23:52:36Z

#### ur:provenance/
lfaohdftfehnfwjpfpnnvydpieclzcuyheietbotftbsnyskmkonwzmnsbzemdethdnelgfsahoxhgltbemwmoondlndhtztcymobytorf
hesppdltzeiagafsfgttglndrl

#### `🅟 CYAN TASK WAVE AWAY`

🅟 💀 🧶 🐝 😘

My New Work
```

PROVENANCE COMMAND LINE TOOL

```
$ cat MyChain/marks/mark-0.json

{
  "ur": "ur:provenance/
lfaohdftlpykeomechldjsbbmhchtdswaodafwtynstthkaytswensndeysfhpqdrdkbkosssonegsaecpdajtprisbefeoxylpaztgels
lsfwbnwtlkhybbrhstcprtzokn",
  "bytewords": "🅟 KING ZOOM DELI FLUX",
  "bytemoji": "🅟 🌎 🐳 😹 🍓",
  "comment": "Genesis mark.",
  "mark": {
    "seq": 0,
    "date": "2025-06-02T23:34:19Z",
    "res": 2,
    "chain_id": "hfUzkReJcRSQF9LGAiVC1A==",
    "key": "hfUzkReJcRSQF9LGAiVC1A==",
    "hash": "e/8nQ+4pxpEVwp21xeuIAg=="
  }
}
```

PROVENANCE COMMAND LINE TOOL

```
$ cat MyChain/generator.json

{
  "res": 2,
  "seed": "BjFAdC21es7vPIFFtQvzx8tcg1fiRUG49BEer7xMHz4=",
  "chainID": "hfUzkReJcRSQF9LGAiVC1A==",
  "nextSeq": 2,
  "rngState": "NJ397j+J82dy3em4dxEi+Na7ZzJaAcJwnvb/NLz7d1A="
}
```

Interoperability

CBOR INTEROPERABILITY

- Provenance Marks are *not* CBOR
  - If you don't use `info`, you don't need to know anything about CBOR or dCBOR
  - Using `info` for simple objects like cryptographic digests of digital works or descriptive text is trivial, adding only a few bytes of overhead for the CBOR type/length
  - Using Gordian Envelope in `info` as a principle carrier of complex metadata is recommended
- Provenance Marks are CBOR-friendly
  - CBOR-encoded as 2-element array `[resolution: number, mark: bstr]`
  - IANA-registered CBOR tag for Provenance Marks: `1347571542` (`'prov'`)
  - When so tagged, Provenance Marks are self-identifying
  - UR-type `ur:provenance` goes with the tag, allowing Provenance Marks to be handled as URIs

EMBEDDING PROVENANCE MARKS

- Provenance Marks can be embedded in any kind of documents
  - Blockchain Commons Extensible Identifier (XID) documents can include Provenance Marks to verify authenticity and ordering of document updates.
- Documents can be embedded in provenance marks
  - The info field can hold complex structures like Gordian Envelope that may themselves hold Provenance Marks.

USES OF THE INFO FIELD

- Counterparty signatures
- Blind signatures
- Digests of and links to claimed objects
- Trees of third-party works incorporated, adapted, or attributed
- Logs of work and chain of custody

BUILDING THE ECOSYSTEM

- User-friendly apps and tools
- Public registries and verification services
- Seed backup and recovery services
- Integration with existing services including social media
- Public standards for use of `info`
- Multiparty control of seeds via distributed computation and ZK proofs

When bits are never scarce, how does establishing provenance help creators manage scarcity?

ESTABLISHING VALUE POST-SCARCITY

- A: commissions work from B
- B: accepts commission from A
- B: Logs events in creative process, establishing value of work
- B: Requests payment for finished work from A
- A: Pays B for finished work
- B: assigns ownership of finished work to A
- A: receives ownership of work from B
- A: later transfers ownership to C
- C: receives ownership of work from A

FURTHER TOPICS IN THE WHITE PAPER

- Security analysis of the four resolutions
- Increased security using heartbeat marks
- Seed rotation
- Potential fields of application

Q&A

CHRISTOPHER ALLEN

✉ christophera@lifewithalacrity.com

𝕏 @BlockchainComns

WOLF MCNALLY

✉ wolf@wolfmcnally.com

𝕏 @WolfMcNally