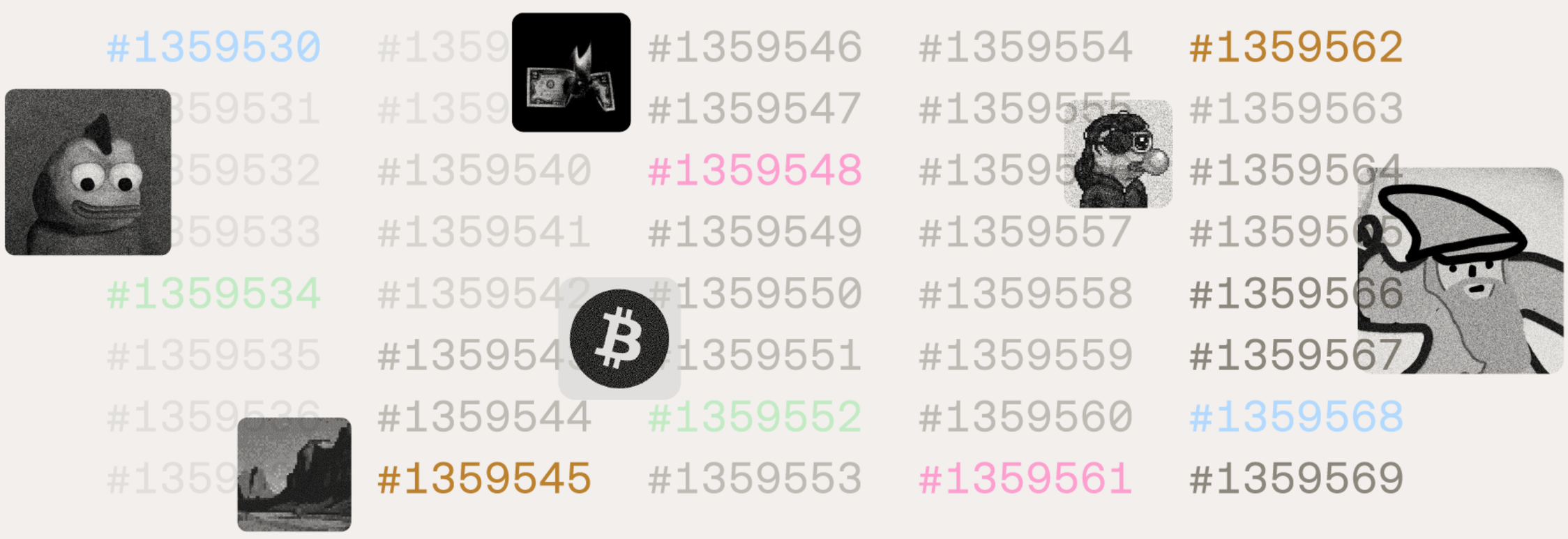
/-/iro : Books

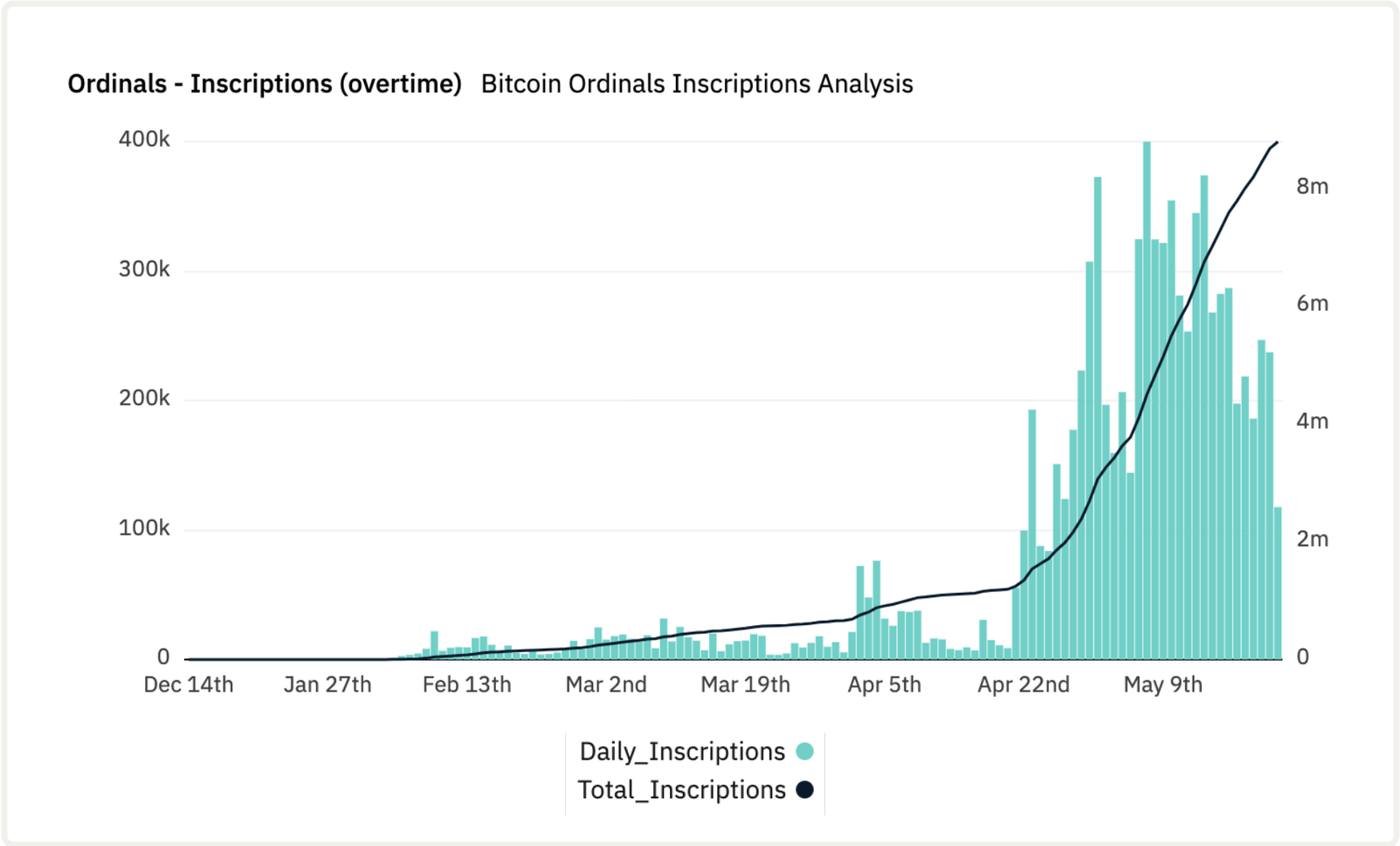
A Developer's Guide to Bitcoin Ordinals



© 2023 Hiro Systems

/ Introduction

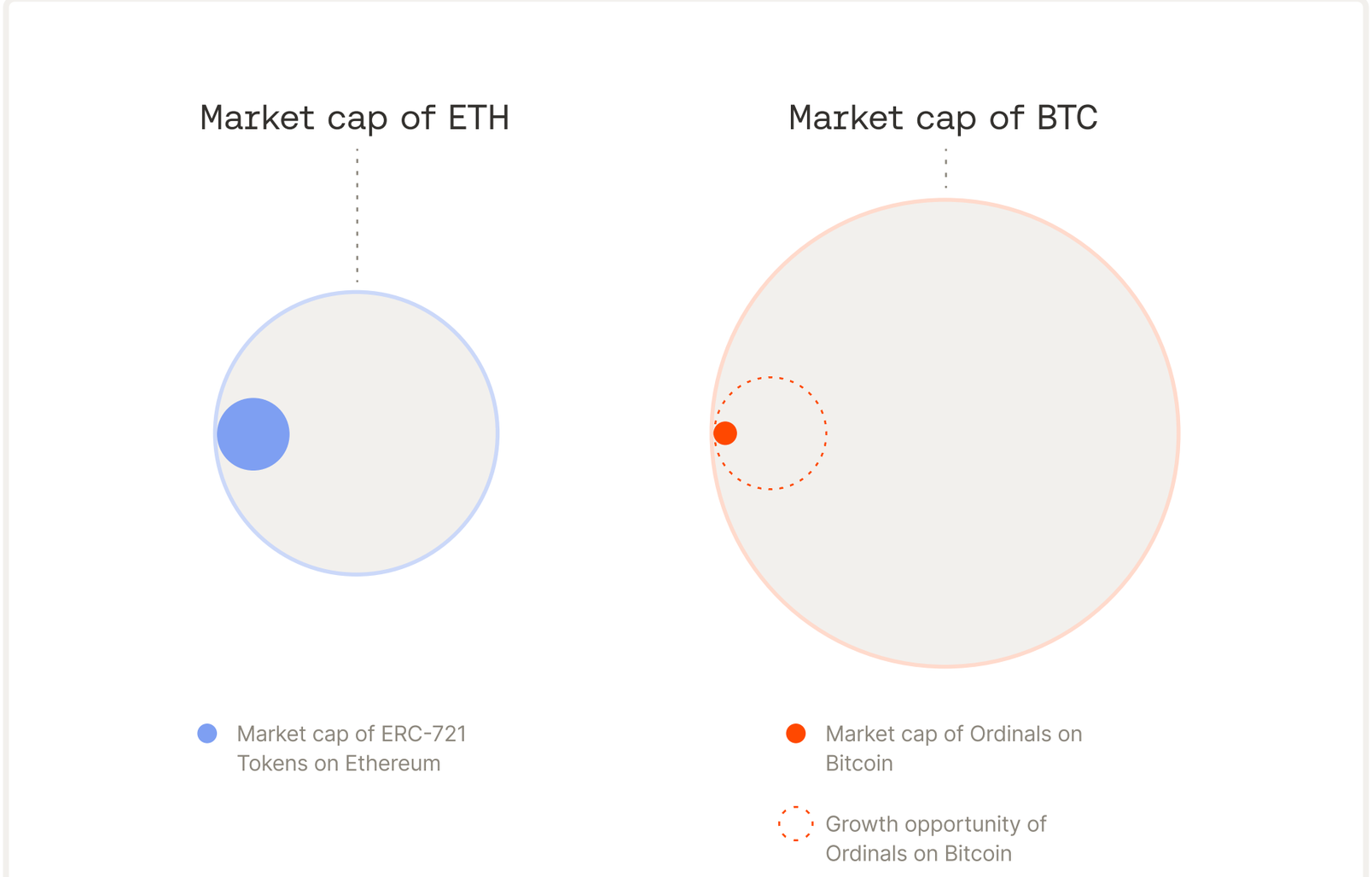
Since their launch on January 21, 2023 ordinals have surged in popularity, and innovation on Bitcoin is back in the mainstream conversation. To date, more than 8.5M ordinals have been inscribed on the Bitcoin blockchain, and ordinals growth is rapidly accelerating.



Source: Dune

As a builder, why should you care? Sure, inscriptions are growing quickly, but what does that mean for you? At a high level, ordinals enable people to put new information on Bitcoin. That could be any kind of data, whether it is art or news, that is stored natively on Bitcoin.

> "In short, you can think of this innovation as the creation of Bitcoin-native NFTs. That is a huge market opportunity for builders."



Ordinals unlock new use cases for Bitcoin, and in this book, we'll explore ordinals in detail and talk about the developer tooling landscape today, so you can start building new apps and experiences that incorporate ordinals.

/ Table of Contents

01 / What Are Bitcoin Ordinals? page_05

Putting Data on Bitcoin Isn't a New Phenomena
A Brief Overview of Recent Bitcoin Innovations

02 / How Ordinals Work page_12

A Primer on Bitcoin Blocks
What Is Ordinal Theory?
What Are Inscriptions?

03 / Challenges with Ordinals Today page_18

04 / A Look at Ordinal Tooling page_19

Tooling for Inscribing Ordinals
Ordinals Wallets
Ordinals Indexers
Ordinals Explorers
Ordinals Marketplaces

05 / The Future Is ~~Extraordinary~~ Extraordinally page_23

01 / What Are Bitcoin Ordinals?

Ordinals enable "digital artifacts" to be inscribed onto the Bitcoin blockchain, creating a new variation of Bitcoin-native NFTs that leverages a loophole in Taproot, the latest upgrade to Bitcoin core. Today, people are inscribing all kinds of things. At a high level, we can categorically break out the types of ordinals by media (data as of May 22nd, 2023):

- 91% text
- 8% images
- 1% other

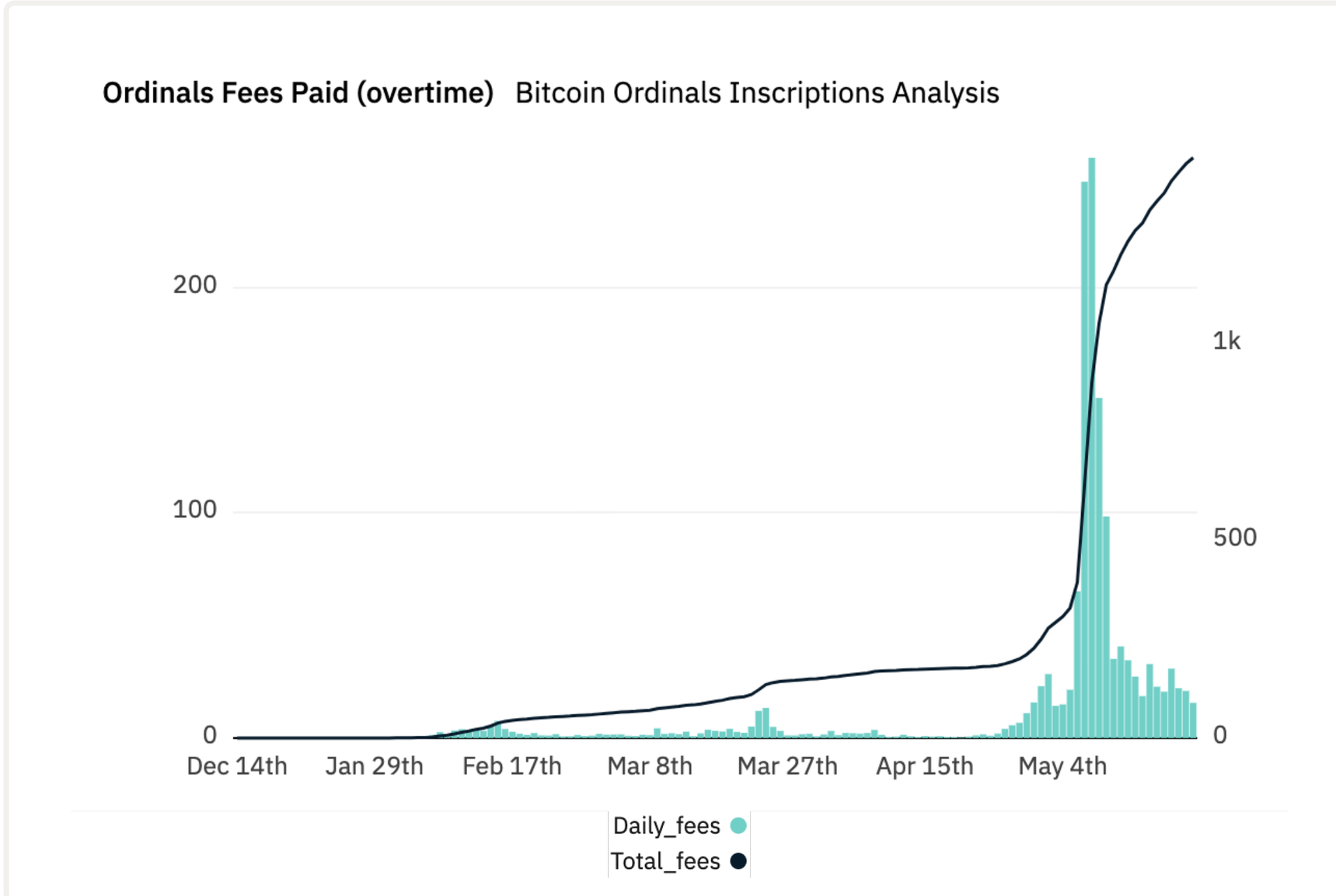
People have gotten very creative with that 5%. For example someone inscribed the 1993 video game DOOM, and now you can literally play a video game on Bitcoin.

> Now you can literally play a video game on Bitcoin.

Ordinals have also attracted noteworthy cross-overs, including ordinals support from Magic Eden (a cross-chain NFT marketplace that started on Solana) and a new collection from Yuga Labs, the company behind NFT collections like Bored Ape Yacht Club and Otherside. In fact, this is the first time Yuga Labs has launched a collection on a blockchain that isn't Ethereum, a clear signal of the demand and interest for art on Bitcoin.

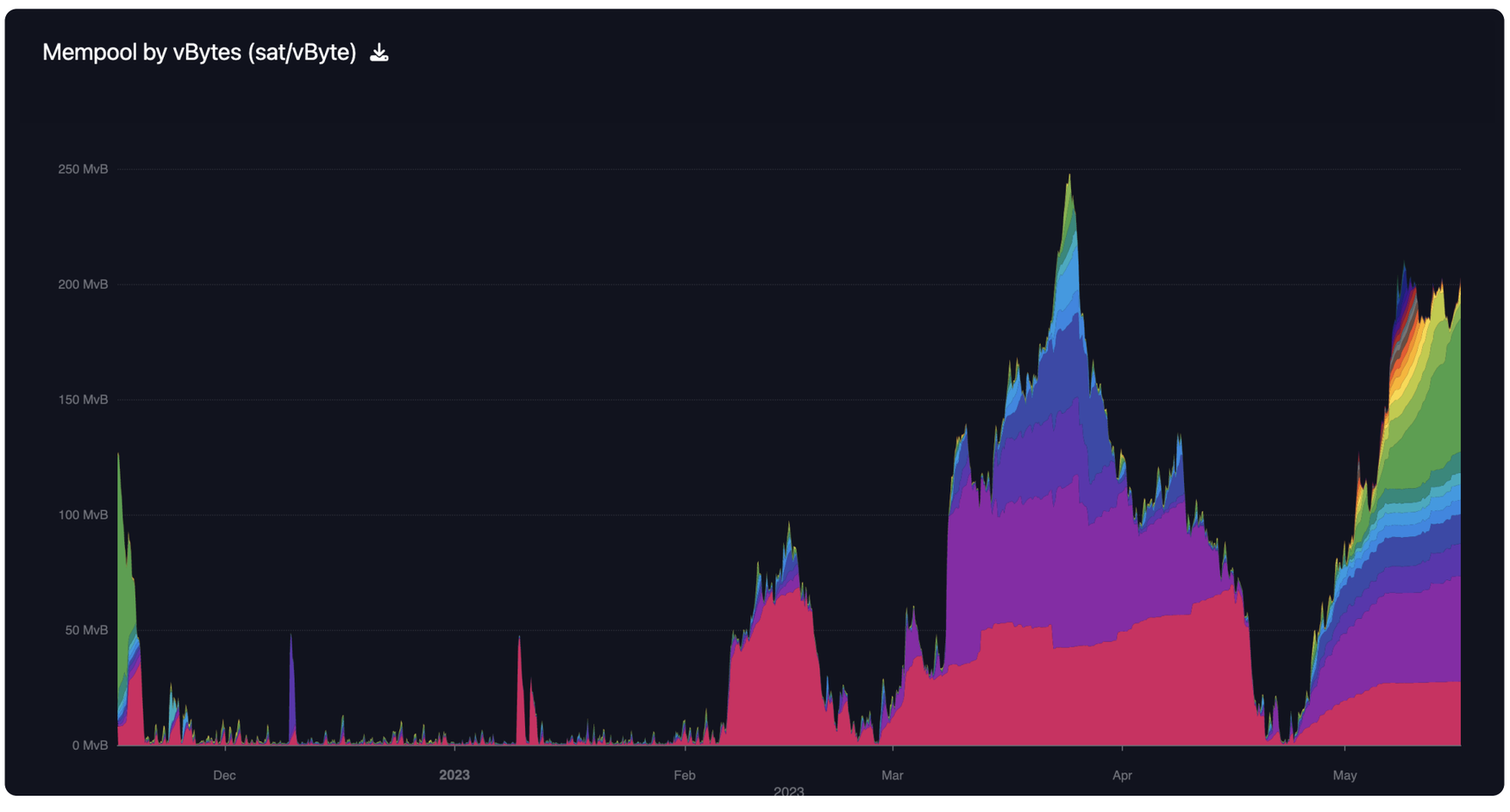
Part of the demand is rooted in the possibility that ordinals may well have more longevity and durability than NFTs on other blockchains. Unlike NFTs found on Ethereum and other L1s, Bitcoin ordinals are inscribed entirely on-chain. There is no IPFS or external storage solution, with the NFT itself just containing a url that points to the NFT (what happens if what is on the other side of that url changes?). Instead, ordinals are self-contained on Bitcoin, tying neatly into Bitcoin's whole ethos of decentralization and security.

That simple concept has captured the public imagination, and people have shown they are willing to pay a premium to inscribe ordinals on Bitcoin.



Source: Dune

In just a few months, ordinals have generated more than $40M in fees for Bitcoin miners, and you can see the effect ordinals have had on the amount users are willing to pay per vByte on Bitcoin, with a rapid climb beginning in February 2023 as ordinals took off.



Source: mempool.space

Ordinals are changing behavior on the Bitcoin blockchain, increasing network fees for miners and making block space more competitive. These are good things for Bitcoin. However, there's no such thing as a free lunch, and there are trade-offs that come with this innovation too.

Pros	Cons
• Drives up fees for miners, increasing the economic rewards that keep Bitcoin secure • Attracts new users • Drives adoption of Taproot transactions • Product-market fit (art on Bitcoin) • Good for Bitcoin layers and the growth of the Bitcoin economy	• Drives up fees for users, increasing economic cost for "regular money" transactions • Could potentially impact the fungibility of BTC and its ability to serve as digital money • Drives speculation • Drives increased tracking and on-chain monitoring • Bloats the network

This chart is inspired by this source.

Putting Data on Bitcoin Isn't a New Phenomena

While the debate around whether ordinals are good or bad is a new one, the act of putting data on the Bitcoin blockchain is in fact very old: people have been putting data on Bitcoin from the beginning. In fact, Satoshi Nakamoto himself put data in the very first Bitcoin block—the genesis block—a sentence that reads "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks."

> "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks."

As for another fun easter egg, a few years ago, another Bitcoiner put a bible verse in block 666,666 that read "Do not be overcome by evil, but overcome evil with good - Romans 12:21".

scriptpubkey (asm)	OP_RETURN OP_PUSHBYTES_70 446f206e6f74206265206f766572636f6d6520627920 6576696c2c20627574206f766572636f6d6 5206576696c207769746820676f6f64202d 20526f6d616e732031323a3231
scriptpubkey (hex)	6a46446f206e6f74206265206f766572636 f6d65206279206576696c2c20627574206f 766572636f6d65206576696c20776974682 0676f6f64202d20526f6d616e732031323a 3231
op_return data	Do not be overcome by evil, but ove rcome evil with good - Romans 12:21

The point is that putting data on Bitcoin isn't new; it has just accelerated in the past few months thanks to a loophole introduced in Taproot, the latest upgrade to Bitcoin core. We'll get into the mechanics of how ordinals work in a bit, and in just the past few months, not only have ordinals skyrocketed to 3M inscriptions, but other innovations have pushed the concept of ordinals further, including BRC-20 and Bitcoin STAMPS.

A Brief Overview of Recent Bitcoin Innovations

	Ordinals	BRC-20 *(A subset of Ordinals)*	Stamps
Launch Date	January 2023 by Casey Rodamor	March 2023 by Domodata	April 2023 by MikeinSpace
Traction *(Data as of May 22nd, 2023)*	8.6M+ inscriptions OrdiPunk sold for $200K	14K+ token types $1B market cap	46K+ inscriptions
Fungibility	Non-fungible	Fungible	Semi-fungible (1 of 1, 1 of many)
Use Case	Good for many formats (image, text, apps)	Good for meme coins	Good for tiny images (24×24 pixels) but works with other formats
Mechanics	Stored in Witness data	Stored in Witness data	Stored in UTXO (OP_RETURN)
Prunable?	Prunable, gives option to node runners	Prunable, gives option to node runners	Impossible to prune

BRC-20

BRC-20 is a variation on Bitcoin ordinals and uses largely the same implementation but instead of providing a way to inscribe non-fungible data on Bitcoin, BRC-20 enables the inscribing of fungible data. The name itself is a play on ERC-20 (the fungible token standard on Etheruem) and is short for "Bitcoin Request for Comment."

Over 3,400 tokens have been launched through this BRC-20 protocol, and the most successful to date have been memecoins such as ordi, pepe, meme, and many more.

Bitcoin STAMPS

Bitcoin STAMPS, short for Secure Tradeable Art Maintained Securely (not a typo—there isn't a word for "P" and they actually put secure in there twice), has shown faster adoption than Bitcoin ordinals when comparing the first 15 days of their launches. STAMPS store data on Bitcoin using the Counterparty protocol, which stores data via multisig transactions and `OP_RETURN`.

In other words, STAMPS data is stored in Bitcoin Outputs (unspent transaction outputs) whereas ordinals data is stored in witness data (we get into the details of ordinals below, but for now just know that storage in witness data is 75% cheaper).

> STAMPS are stored in Bitcoin UTXOs whereas ordinals are stored in witness data.

Storage in this format is expensive, and so the creators recommend only attempting to inscribe a few bytes of data (such as an image with a max resolution of 24 × 24 pixels). The STAMPS standard is also more flexible than either ordinals or BRC-20 and enables the creation of fungible and non-fungible assets on Bitcoin.

An important differentiator between STAMPS and the other implementations is the concept of pruning. Both ordinals and BRC-20 are prunable, meaning if you are running a Bitcoin node, and you don't want to download all of the data associated with ordinals, you don't have to! You can prune it and exclude that data from your node.

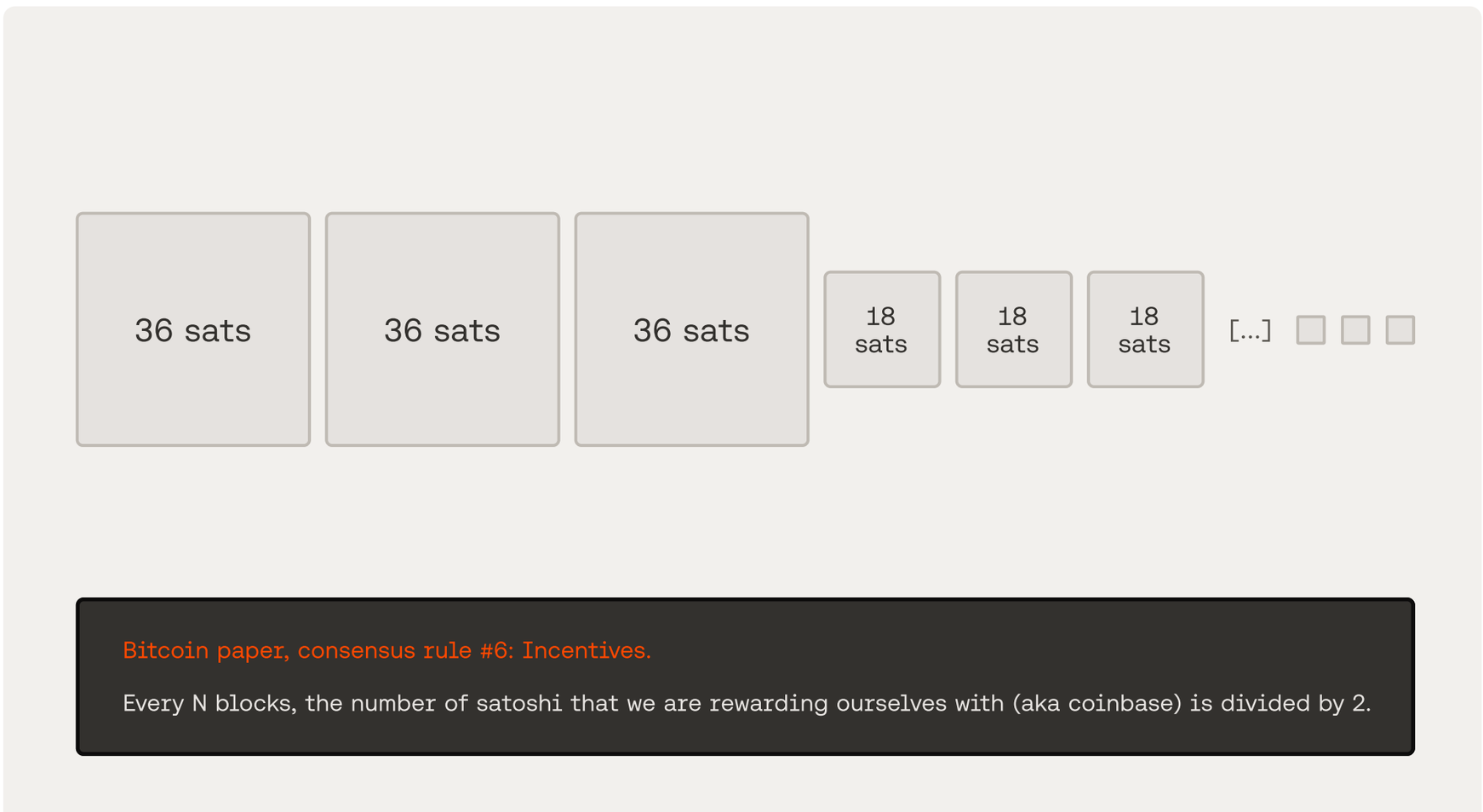
However, STAMPS stores data inside of Bitcoin UTXOs, which makes them impossible to prune. All Bitcoin nodes must process them. This makes them more censorship resistant, but some people don't like them because they bloat the blockchain more.

While STAMPS are interesting and worth investigating further, the rest of this book is focused on Bitcoin ordinals specifically. With more than 3M inscriptions (a count that is rapidly climbing), ordinals seems to be the implementation gaining the most market traction, and the one we at Hiro are focused on. Let's dive in.

02 / How Ordinals Work

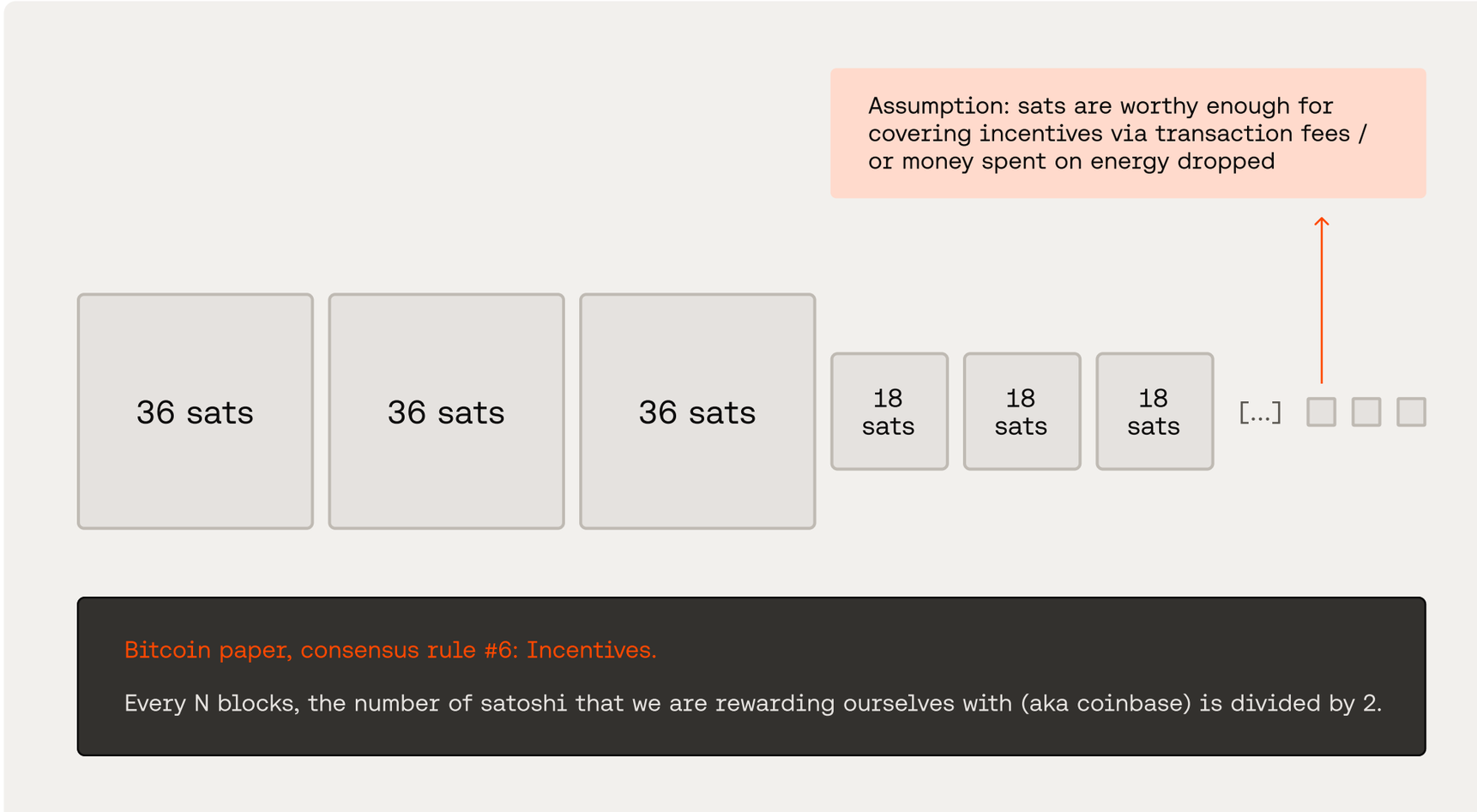
A Primer on Bitcoin Blocks

Bitcoin has a fixed supply of 21M Bitcoin. There will never be more than those 21M, and today there are already 19M in circulation. The remaining ~2 million Bitcoin will be released at regular intervals during the mining of new blocks.



Each mined block triggers an emission of Bitcoin (called the "coinbase") that goes to the miner as a reward for doing the work of packaging transactions and mining that block. This introduces new Bitcoin to the circulating supply and is where the term "mining" comes from: the analogy of an ore miner mining new silver, adding to the total circulating supply of silver, is an apt one.

That coinbase—the Bitcoin rewards of each new block—halves (is divided by 2) every four years. Today, the coinbase reward is 6.25 BTC, next year it will be 3.125 BTC, and 4 years after that, it will be 1.56 BTC. One day in the future, the coinbase will be denominated in "sats" (or satoshis), the smallest transferable unit of Bitcoin—each Bitcoin has 100M sats—and at some point around 2140, the last BTC will be mined.

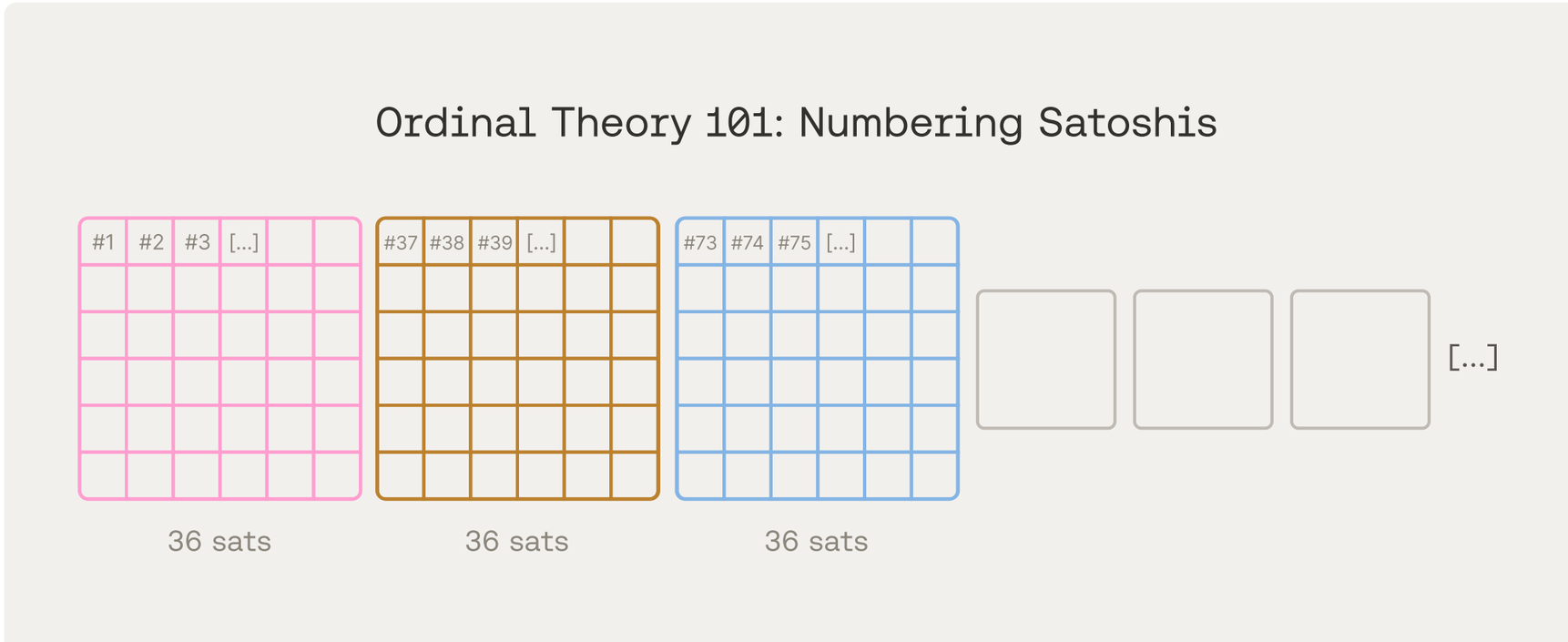


The assumption of this halvening is that eventually the fees miners collect from transaction fees alone will make mining profitable, and the coinbase will no longer be needed to incentivize miners.

To put this in perspective, there are roughly 144 Bitcoin blocks that get mined everyday. With a coinbase reward of 6.25 per block today, that's currently a daily emission of 900 BTC. Assuming a price of $30K, that's $27M per day. As we mentioned earlier, mining fees from ordinals are a little over $6M over three months, so a lot of network growth is needed for fees to replace coinbase rewards.

Now you're thinking, "I have a grip on Bitcoin, blocks, and sats. Where do ordinals come in?" We thought you'd never ask.

What Is Ordinal Theory?



The concept behind ordinal theory is relatively simple. Ordinal theory is in essence just a numbering system. We know the order of Bitcoin blocks, and we know the coinbase rewards for future Bitcoin blocks. What if we could agree on a way to order the satoshis within each Bitcoin?

Ordinal theory proposes numbering satoshis in the order they are mined, so that we can uniquely number each and every sat that has ever existed as well as every sat that will exist in the future. To be clear, this is a convention—the theory is only valuable to the degree that people accept the numbering system. And in fact, a numbering system isn't a new idea and goes back to as early as 2012 (and has been proposed by several different entities).

That is essentially the theory behind ordinals, but how do they work mechanically?

> How do you add non-fungibility to a fungible token?

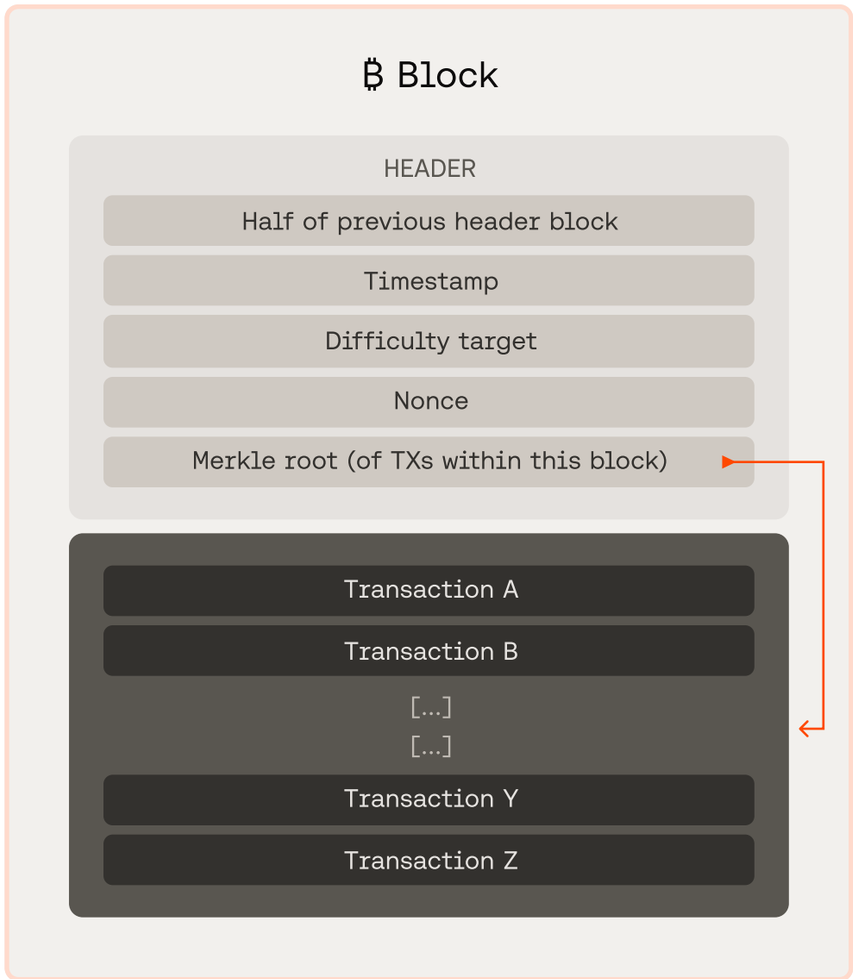
What Are Inscriptions?

Ordinal Theory 102: Inscriptions

Note: Ordinal number ≠ Inscription number

A numbering/naming scheme by itself has limited utility. Once you're able to attach some content to a name/number, then things get a lot more interesting. To talk about how ordinals accomplish this, we first need to establish some basic concepts to understand how it all works.

Inside a Bitcoin Block



You can think of a Bitcoin block as having two kinds of information:

- Information about the block itself, including its timestamp, nonce, etc
- Information about transactions within the block

Ordinals are inscribed within the latter: Bitcoin transactions themselves.

Inputs and Outputs

A Bitcoin `transaction` is basically information that describes the flow of bitcoins. Transactions can have multiple outputs and inputs:

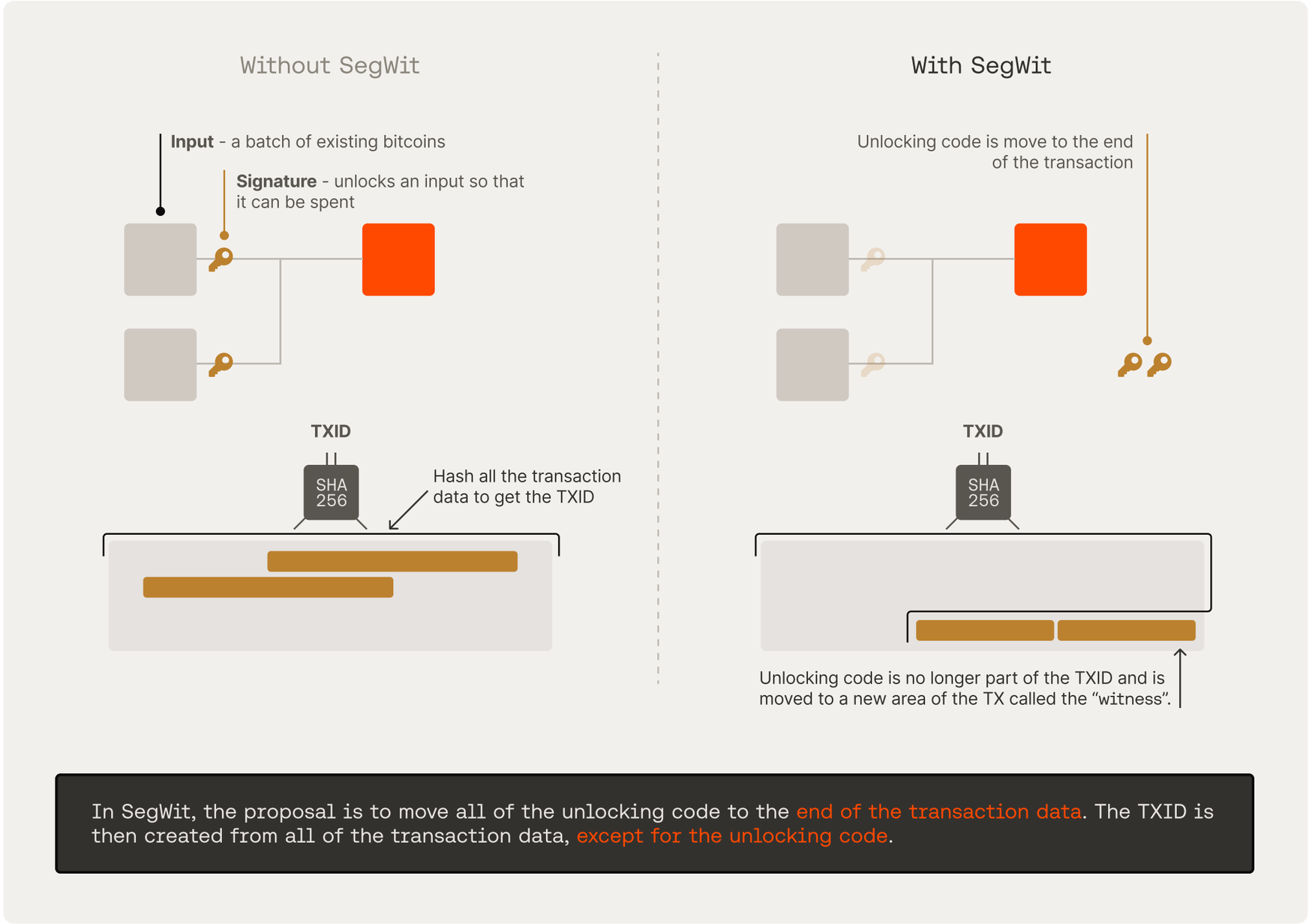
- `Outputs` are the bitcoins generated by a transaction. Each output has a "lock", to ensure that it can only be spent in a future input by those who have the correct "key".
- `Inputs` are just outputs from previous transactions that this transaction intends to spend. Per above, an input must supply the right key to unlock the output.

This "key" is often referred to as a signature, or proof, or "witness".

SegWit

Prior to 2017, the way Bitcoin transactions were formatted, the inputs and outputs were laid out sequentially along with their corresponding locks and keys. Thus the unlocking code was spread out throughout the transaction (and hence the block).

In 2017, an upgrade – known as `SegWit` – went into effect that collects all the unlocking code in a dedicated section at the end of the transaction. Hence SegWit – for "segregating" all the "witness" data.



Source

The Taproot upgrade, which came later in November 2021, introduced some new capabilities to the Bitcoin network. Relevant to us here is that the upgrade supports more complex "proofs".

To use an oversimplified analogy, Taproot lets you construct outputs with a "lock" that can be opened by one of many "keys". Critically, to "unlock", one need not "reveal" all the possible keys, only the one specific key that the input trying to spend that output is using.

How that works out in practice is something like:

- Transaction 1: the first transaction creates an output that says "here's a lock that can be unlocked by one of N keys (one multisig, one biometric, one time-locked, etc)"

- Transaction 2: the second transaction is an input that "reveals" one of the keys (and only that one)

Taproot puts relatively few constraints on these keys (also called "tapscripts" or "spend scripts" in this context, because the script is trying to spend a previous output). And so you could, in theory, encode arbitrary data within a spend script.

Ordinals takes advantage of this Taproot loophole and encodes data inside of these spend scripts, which exist in the segregated witness data of a Bitcoin transaction. This detail is important because the witness data offers a larger space for storing data than exists inside of UTXOs using `OP_RETURN` (like what Bitcoin STAMPS uses).

SegWit's design offers a 75% discount for stored witness data (only 25% of witness data counts towards transaction size). There are a couple of reasons for this: 1) it was an incentive for the network to upgrade to SegWit's optmized transaction structure 2) it allowed the network to increase block size and 3) witness data only exists inside of inputs—it never enters the UTXO data set, which means that the data is prunable and nodes can discard the data if they choose to.

This all means that ordinal inscriptions enjoy cheaper data rates compared to Bitcoin transactions, which means that you can get more creative with what you inscribe. After all, it's more affordable.

Inscribing an Ordinal

Putting all of that together, this is how you inscribe an ordinal:

- Transaction 1: the first transaction is an output of 1 sat (the ordinal) that "commits" to a Taproot spend script – meaning, specifies a lock that can only be unlocked by a specific tapscript.
- Transaction 2: the second transaction is an input that spends the above output, revealing the contents on-chain in the process.

The specific encoding of content is done by using the following structure. The various `OP_*` instructions are called opcodes – they are keywords in the Bitcoin scripting language. The main thing to note is that the contents of this script are never executed because of the `OP_FALSE` conditional at the top. In the sample below, this ordinal is a text inscription that reads "Hello, world!"

```
01  OP_FALSE
02  OP_IF
03    OP_PUSH "ord"
04    OP_1
05    OP_PUSH "text/plain;charset=utf-8"
06    OP_0
07    OP_PUSH "Hello, world!"
08  OP_ENDIF
```

Note: the ordinal number of an individual satoshi is different from its inscription number. An ordinal number refers to the order a particular satoshi was mined; an inscription number refers to the order a particular satoshi was inscribed. These are not the same thing.

03 / Challenges with Ordinals Today

Ordinals are a very new innovation, and there are still several kinks being worked out. For example, many Bitcoin wallets aren't equipped to support ordinals. Users could accidentally send their ordinals when sending a BTC transaction. Remember: there are 100M satoshis per Bitcoin—that's a lot of potential ordinals to track—and most wallets today aren't designed to support the ordinals use case.

Ordinals in their current incarnation also don't have a clean path towards collections: how can you identify whether a set of ordinals are part of a singular collection? Similarly, there are challenges around provenance. How do you prove the history of a particular ordinal? Who created it? What happens if someone inscribes the same satoshi more than once?

These issues get thornier when you consider that ordinals data can be pruned. If different Bitcoin nodes are looking at different subsets of ordinals data, based on what they prune and when, it can be harder to agree on who owns what ordinal and what exactly is subscribed on a particular sat.

Lastly, ordinals, BRC-20, and Bitcoin STAMPS are all new experiments. They may yet fail due to some undiscovered bug, they may all likely go to zero (the BRC-20 creator even says that explicitly), and they may all have unforeseen consequences on Bitcoin. Building and experimenting with any project in its infancy is a challenge in itself.

04 / A Look at Ordinal Tooling

If you want to inscribe ordinals on your own, you can do so, but the process is tedious. It involves:

- Running a Bitcoin core node
- Creating an `ord` wallet
- Indexing the entire history of ordinals with that `ord` wallet
- Broadcasting your ordinal txns to the mempool

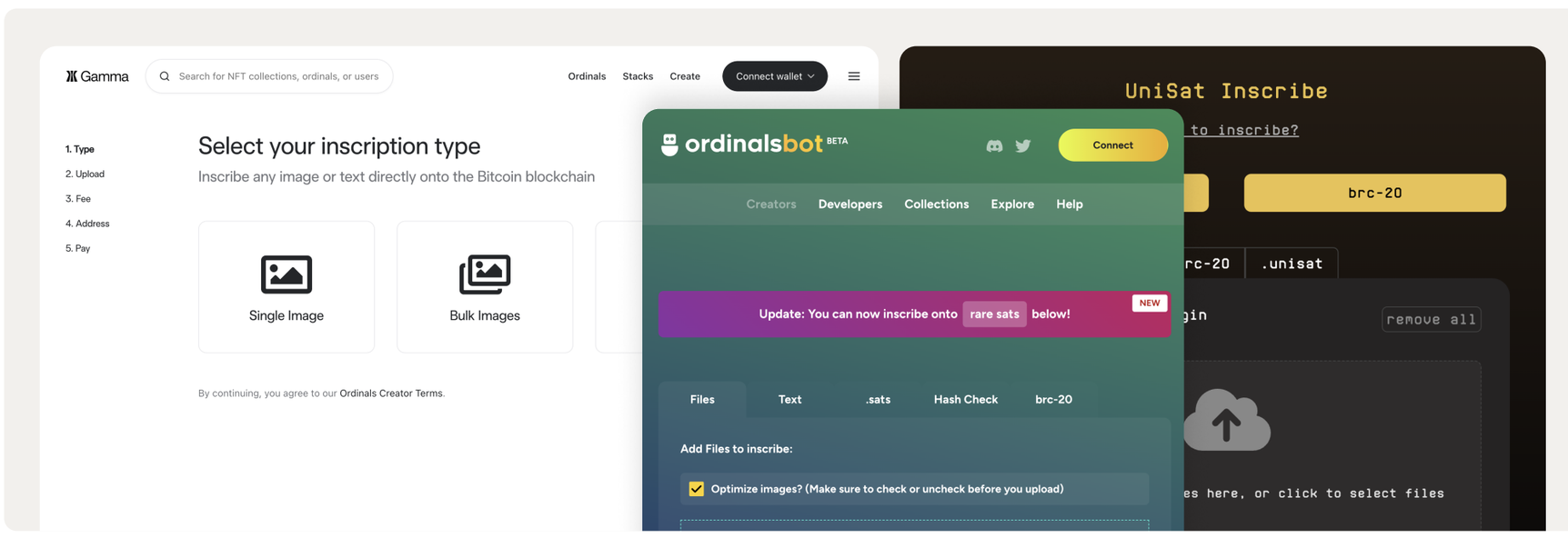
That's a lot of work for the average developer (you can read about inscribing your own ordinals in greater detail here). The good news is that you don't have to create ordinals on your own anymore. In just a few months, a number of tools have emerged to make developers' lives easier, and tooling around this new innovation will continue to improve in time.

But if you're considering launching an ordinals collection or want to integrate ordinals into your application, you'll likely want to use some combination of the tools below.

Tooling for Inscribing Ordinals

The process of inscribing ordinals described above is far less complicated now, thanks to several new tools that remove the need for running a full Bitcoin node in order to create ordinals. Those no-code tools offer web portals that let you inscribe any image, text, and related file types, letting you specify your transaction fee and ordinal recipient address. These tools include:

- Ordinalsbot
- Gamma
- Unisat
- OrdSwap
- Ordimint



Ordinals Wallets

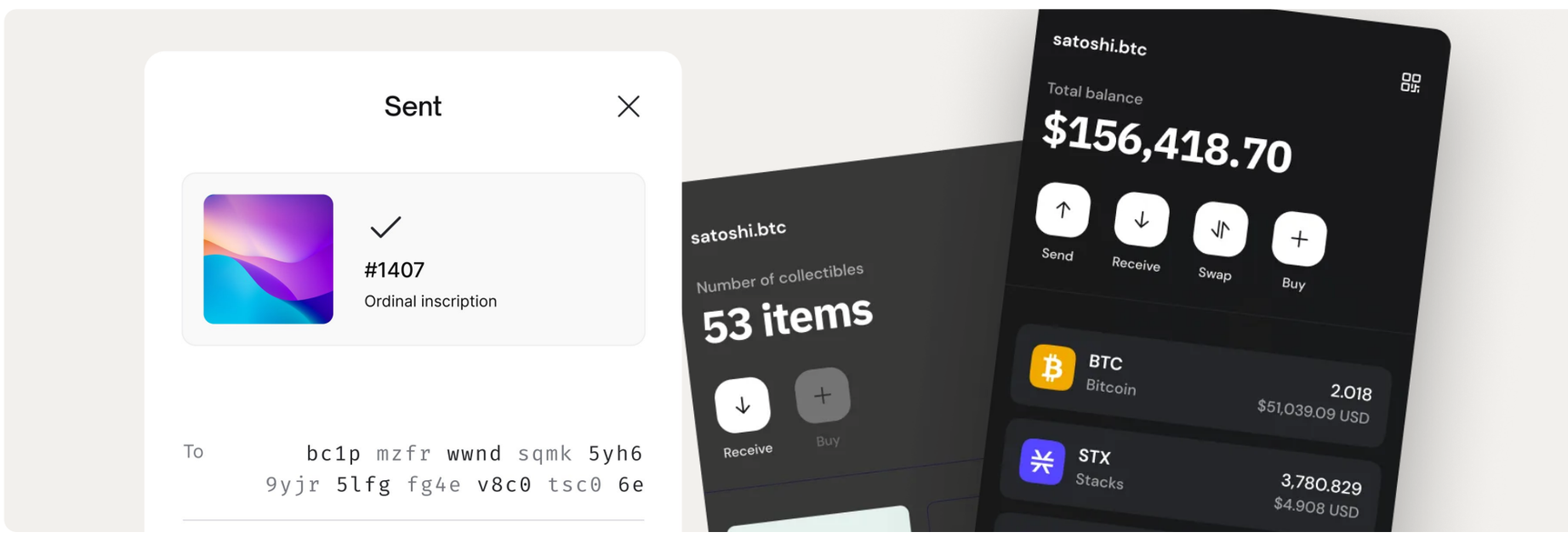
Users need a way to store their ordinals. How can they separate their individually inscribed sats from the rest of their Bitcoin, and how can their wallet make sure that they don't accidentally send their ordinal when sending some BTC? Plenty of teams are working on just that.

Today, you have a number of wallets that you can integrate with that offer the following capabilities:

- The ability to hold and store ordinals
- The ability for users to see the ordinals in their wallet
- The ability to transfer those ordinals to and from other addresses

Early wallet providers include:

- Xverse
- Hiro Wallet
- Ordinals Wallet
- Unisat
- Sparrow Wallet
- Electrum Wallet
- Alby



Hiro Wallet & Xverse Wallet

Ordinals Indexers

If you're building an application around ordinals, you need a way to index ordinals data on the Bitcoin blockchain. Who owns what? What new ordinals are being inscribed? You need a source of truth.

One way to index ordinals data is to run `ord` .This is the process described above, where you run your own Bitcoin node, install and boot up `ord` , and then you can index all ordinals data. Voilà! You are now self-reliant and are verifying all on-chain data yourself.

But like we mentioned, that's a lot of work, and maybe you don't want to do it. There are a number of tools that handle this indexing layer for you. Among them:

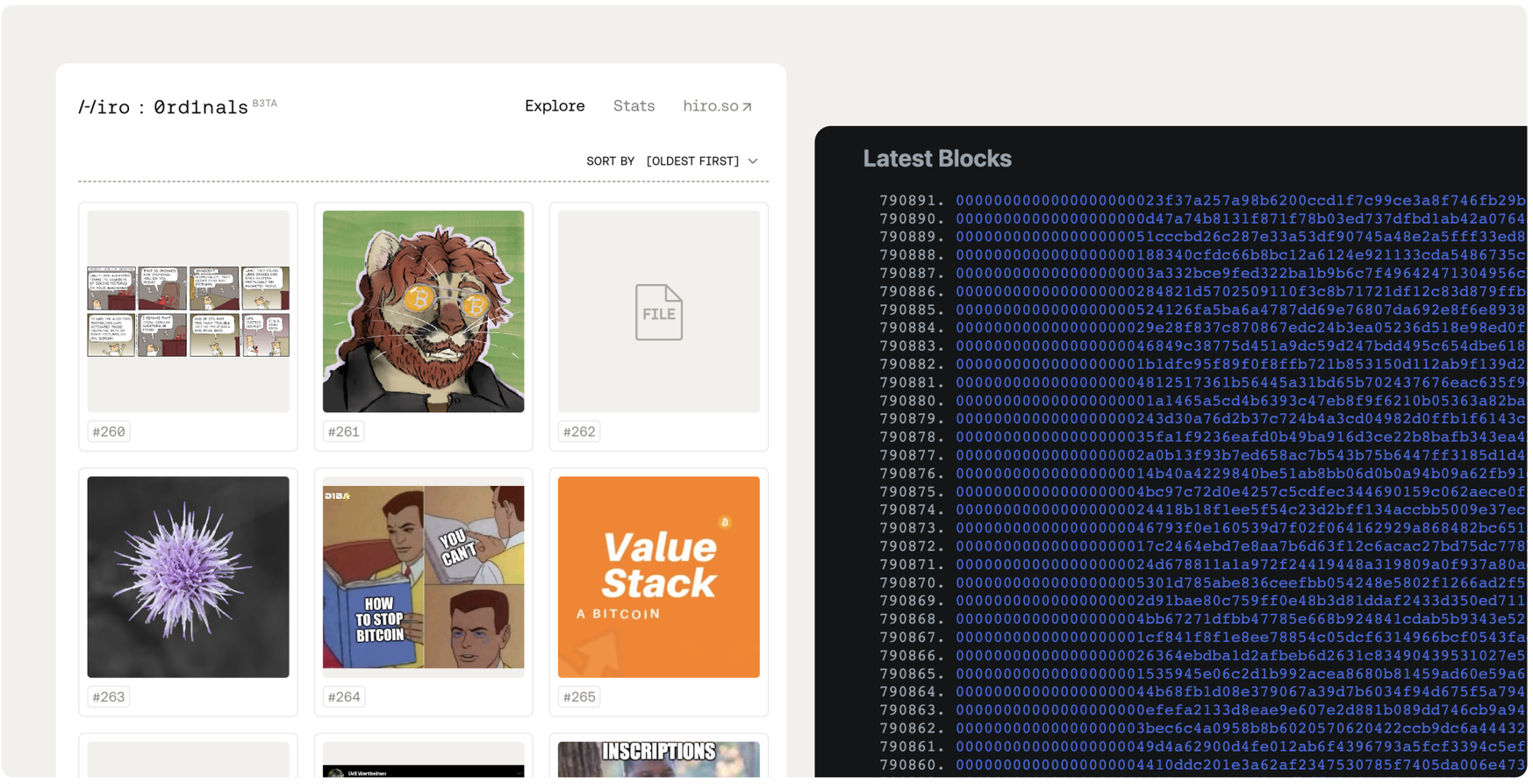
- Ordinals HTTP API — an HTTP API that outputs JSON. This tool leverages `ord` and makes your life easier by exposing data from `ord` via easy to use endpoints.

- Hiro's Ordinals API — at Hiro, we offer an Ordinals API that is based off our own implementation of the ordinal theory, something we are calling `/-/ord`. Unlike `ord`, the first implementation of ordinal theory, `/-/ord` is reorg resistant, so when the Bitcoin blockchain forks or goes through a reorg, this API can gracefully handle that shift without the indexer breaking down.

Ordinals Explorers

You may also need (or just want) an explorer, a way to, uh, explore ordinals and their latest activity. What's being minted? What is the status of your users' inscriptions? Where can users check the status of their transactions? There are a number of tools that can do just that, including:

- Ordinals.com
- Hiro's Ordinals Explorer



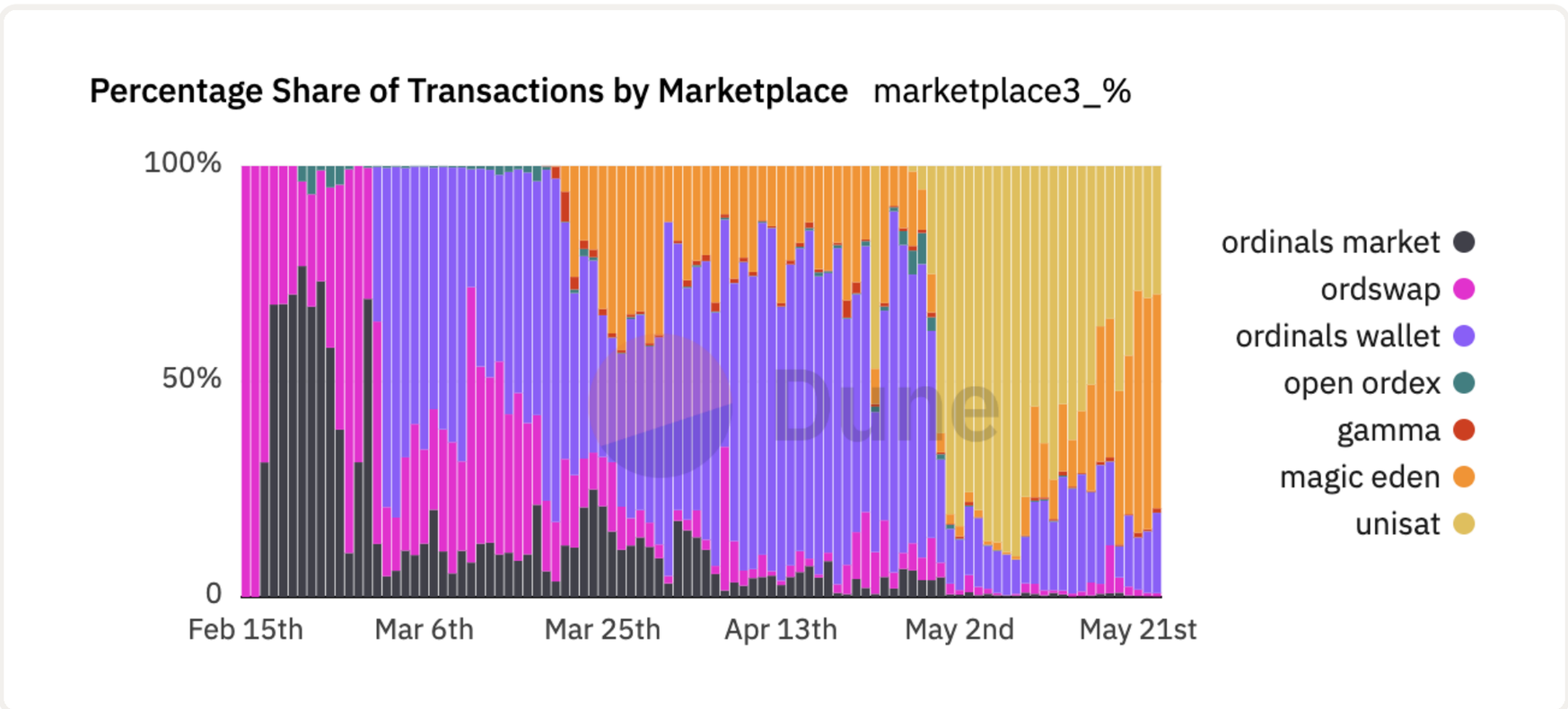
Ordinals Marketplaces

When ordinals first launched, there wasn't a way to trade them. That didn't stop builders from quickly shipping, and within a matter of weeks, users could wrap ordinals via Emblem Vault (which wraps the asset as an ERC-20 token), and then trade ordinals on OpenSea.

Today, there are a lot more options for developers and users to choose from to trade ordinals, including:

- Unisat
- Ordinalswallet.com
- Magic Eden — an Ethereum marketplace that has added ordinals support
- Gamma
- Ordinals.market
- Ordswap.io
- Ordynals.com
- Generative.xyz
- Ordsea.io
- OrdinalHub.com
- OpenOrdex
- ordkit_

While it is early days yet and far too early to declare winners and losers, you can see which marketplaces have gained some preliminary traction below:



Source: Dune

05 / The Future Is ~~Extraordinary~~ Extraordinally

A simple way of thinking about ordinals is that they are a "dumb" data store on Bitcoin with little to no semantic structure. But despite that simplicity, there is a clear product-market fit, based on how quickly users have latched on to the concept, how quickly behavior on the Bitcoin chain has changed, and how quickly developers have rolled out all kinds of tooling and applications.

We haven't seen this much excitement around Bitcoin in years, and we love to see it! If after reading this you're inspired to start building with ordinals, we're here to help you. Chat with us on the HIRO-PUBLIC channels on Discord, and play around with Hiro's Ordinals API and Ordinals Explorer.

Happy building.

A Developer’s Guide to Bitcoin Ordinals

/-/iro

hiro.so