Trollip's Degen Almanack

Justin Trollip

Invalid Date

Table of contents

Preface

> "Arise, you have nothing to lose but your barbed wire fences!" (May 1988)

For my two princes. Huxley & Morrissey.

For the bois, the Lord's of Slumtown.

Why I must write

So that I can continue to use crypto without dishonor, I have decided to put together a sufficient body of free research so that I will be able to get along without any knowledge that is not free and accessible by anyone.

Why many others want to help

I have found that many within crypto are unhappy with the exploitation of digital currencies. The influx of grifters, charlatons and scammers. It may enable them to make more money, but it requires them to feel in conflict with humanity rather than feel as comrades.

The fundamental act of friendship among degens is the sharing of knowledge. Knowledge in crypto, due to it's technical complexity has attracted a special type of exploitation. High pay walls for access to data goes against the very ethos of Web3.

How you can contribute

Currently you can contribute by submitting knowledge in the form of a Pull Request to the Github repo.

I'm considering opening up this Almanack onto Mirror so people can donate financially and anonymously through there. I still need to do more research on Open Source financial models to ensure this Almanack can operate sustainably as a going concern.

Stay Tuned!

1 Introduction

The genesis of this book was reading a technical argument on Twitter that was steeped in ideology. I agreed with both parties on certain points, yet the two combatants resorted to ad hominen attacks, both lacking a standardized framework in which to couch their arguments. This led me to rekindle an idea I'd had years before, to create a blockchain taxonomy. Something that would be easily accessible and would be able to be used to quickly dispel bad faith arguments.

After building out the initial taxonomy, the breadth of the task started to dawn on me. The potential of it could be expanded to provide value not just to the technical community but to all parties interested in blockchain technology. I kept recalling the numerous conversations with family and friends asking for crypto investment advice. This wonderful patchwork of humanity I've had the fortune of spending my life with and how I could never give them an answer that spanned the breadth of this fascinating industry I love. Even sadder, I had no way of giving them advice concise enough for them to follow, that matched their investment goals.

What then if there was a book, an Almanack for lack of a better word, that I could give to these people. A book that would educate, illuminate and help them make well informed investing decisions. While still straddling a fine line of being technically honest on how to safely navigate these waters.

Taking a cue from history, following in the footsteps of Henry Vanrum Poor when he recognized the need for reliable information about the booming railroad companies. This takes inspiration from that venture with a modern twist. In line with the cypherpunk manifesto(Hughes 1993) and the open source communities (Raymond 1999) which I admire so much, this Almanack will be open source and community contributed. As a core tenet of the Cypherpunk manifesto, contributors identity will be less important than their reputation. The knowledge will be curated and go through an approval process via Github Pull Requests. The other innovation is that as the Alamack evolves over time it will integrate with real time data sources in order to provide a dynamic document that plugs into the market as it evolves.

A brief note on the naming of the book. Initially I wanted to call it blockchain taxonomy, reflecting it's early design goals. However as I worked more and more and it evolved into something much broader than a Taxonomy, Almanack made a lot more sense. Initially I liked Trollip's Web3 Almanack. But all the broader terms for this industry have severe limitations. I consider myself a degen. And while it's a loaded term, it's being a degen that has armed me with the knowledge to attempt this ambitious venture. Hence Trollip's Degen Almanack.

Part I

Web3 Essentials

2 Crypto Terms

This document I believe will be updated most frequently. Terms are constantly changing, definitions being debated on the public square and evolving at a quicksilver pace.

Please treat it as a living document. The Almanack uses version control, so it will be possilbe to see how terms change over time by going back to previous versions.

2.1 Core Concepts

2.1.1 Web3

While our industry is built on the back bone of cryptography, it encompasses not only a branch of mathematics, but economic, social, technological and political aspects. Web3 more closely aligns with the breadth of our industry.

Below you can see how the web has evolved into Web3.

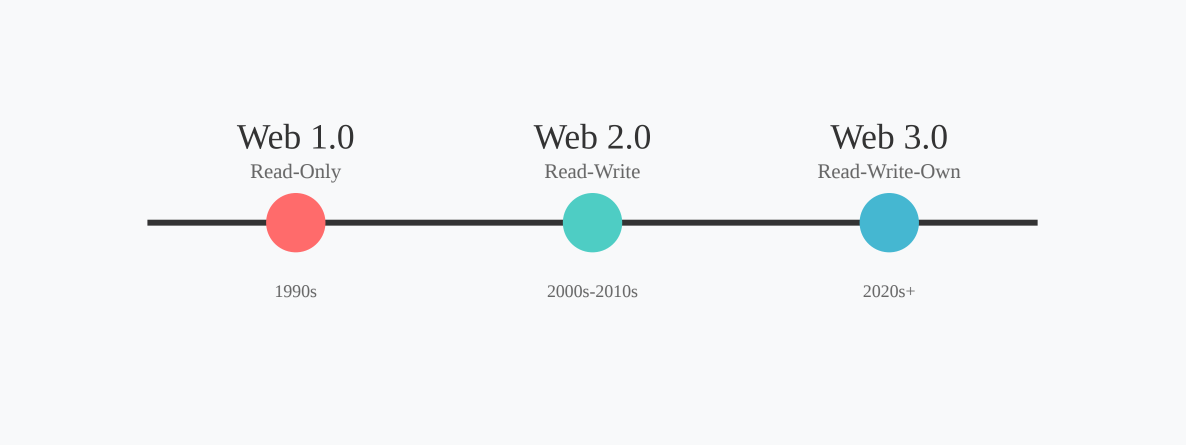

Figure 2.1: Evolution of the Web

- Web1 (1990s): static websites, read-only content
- Web2 (2000s - 2010s): Interactive social platforms, user-generated content
- Web3 (2020s+): Decentralized networks, blockchain, user-owned data and assets

2.1.2 Decentralization

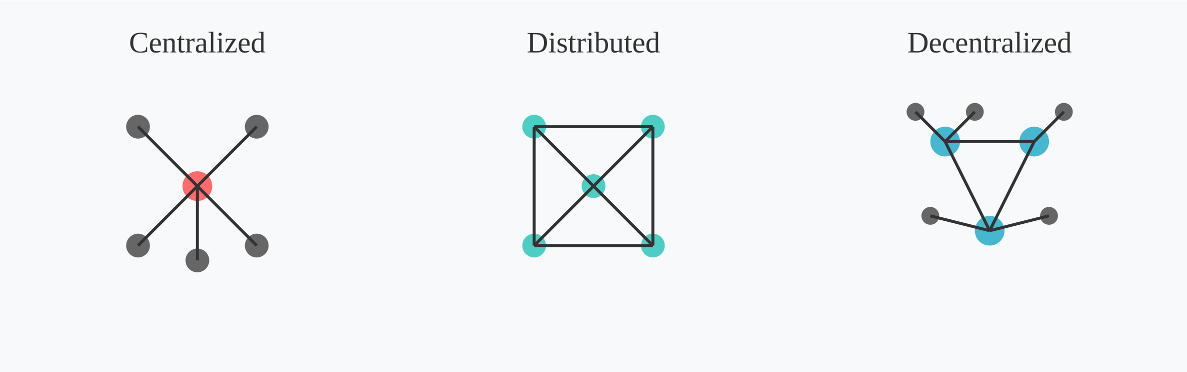

Figure 2.2: Network Topologies

2.1.3 Blockchains

2.1.3.1 Layer 1

Currently there is no formally and universally accepted definition of what comprises a L1 Network.

We have taken these sources in order to come up with our definition:

- (Nakamoto 2008) for Bitcoin
- (Buterin 2013) for Ethereum
- (Buterin 2021) for the rollups guide
- (Charbonneau 2023) for John's excellent guide to rollups

Nakamoto's Bitcoin whitepaper, while not explicitly using the term "Layer 1," established the core principles of what we now recognize as Layer 1 characteristics: a base protocol that handles consensus, security, and data availability without relying on any other blockchain system. This introduced the concept of a `self-contained, sovereign blockchain network`.

The Ethereum whitepaper expanded this foundation by introducing programmability and demonstrating that a Layer 1 could be more than just a payment system. It showed that a Layer 1 blockchain serves as a foundation for broader computational capabilities while maintaining the core properties of decentralization and security.

Buterin's rollup guide helps define Layer 1 by contrast - it clarifies what makes a base layer distinct from scaling solutions built on top of it. This work emphasizes that a Layer 1 blockchain must handle three critical functions: data availability, consensus, and execution, all while maintaining decentralization.

Charbonneau's report further refines our understanding by examining how Layer 1 blockchains interact with scaling solutions, highlighting their role as the security and settlement layer of the blockchain ecosystem.

Synthesizing these sources, we can define a Layer 1 blockchain as:

A sovereign blockchain network that provides three fundamental guarantees without relying on any other blockchain system:

1. consensus - the ability to agree on the state of the network
2. data availability - ensuring all transaction data is publicly accessible and verifiable
3. execution - the processing of transactions and state changes. It must do this while maintaining decentralization in its security model and serving as the ultimate settlement layer where the final state of all transactions is recorded.

2.1.3.1.1 Settlement Layer

This is kinda a meme at this point and no formal definition exists. Although lots of opinions, as is generally the case in crypto.

> ℹ Note

> ℹ Note

2.1.3.2 Layer 2

There is no canonical definition of L2's.

We use:

- (Gluchowski 2019)
- (Poon and Buterin 2017)
- (Poon and Dryja 2016)
- (Buterin 2021)
- (Charbonneau 2023)

To come up with the following Framework

1. Data Availability

- On-chain (Rollups)
- Off-chain (Validiums, Plasma, Sidechains)

2. Security Model

- Inherited from L1 (Rollups, Validiums)
- Independent (Sidechains)
- Hybrid (Plasma)

3. State Validation

- Fraud Proofs (Optimistic Rollups)
- Validity Proofs (ZK Rollups, Validiums)
- Independent (Sidechains)
- Merkle Roots (Plasma)

4. Settlement Mechanism

- Direct (Rollups)
- Challenge Period (Optimistic Rollups)
- Exit Games (Plasma)
- Bridge/Peg (Sidechains)

2.2 Accounts

While technically different blockchains will have their own nuances and their own technically correct terms for the concept of an "account", we will refer to them all as accounts for ease of reference.

2.2.1 Seed phrases

In order to create an Account on a blockchain, it needs to be secured by a public-private key pair. The most common way to do is via a Seed phrase generated by the wallet.

Since your private key is the singular point of security for your interaction on that blockchain, it needs to be incredibly secure. This means that your private key is a very large random number. Due to the difficulty of working with these large random numbers, Seed phrases were created to improve the user experience. Historically humans have used "code words" to obfuscate and hide meaning. Seed phrases are just a way to generate a private key in a more human relatable way.

2.2.2 Externally Owned Accounts

The formal definition of an EOA comes from Ethereum's Yellow Paper. (Wood 2014)

There it is defined as:

- A nonce (counter for transactions sent)
- A balance (in the native currency)
- No associated contract code
- No data storage

However we live in a multi-chain world so we'll define our own interpretation as:

- Direct control through a public-private key pair
- No additional programmable logic at the account level
- Transactions must be directly signed by the controlling private key

2.2.3 DeFi - Dentralized Finance

- Slippage

2.3 Colloqualisms

2.3.1 Degens

"Degen" (short for "degenerate") in crypto/Web3 culture refers to aggressive or risk-seeking participants in cryptocurrency markets and DeFi (decentralized finance) protocols. The term originated from gambling culture but has been embraced by the crypto community as a semi-ironic badge of honor.

- Trading Behavior: Degens are known for taking high-risk positions, often using significant leverage, and engaging in yield farming, liquidity mining, and other complex DeFi strategies.
- Cultural Identity: Unlike traditional finance's negative connotation of "degenerate gambling," the crypto community has reclaimed "degen" as a positive or playful identifier. It represents a willingness to experiment with new protocols and take calculated risks.
- Technical Sophistication: Despite the seeming recklessness implied by the term, many "degens" are highly knowledgeable about blockchain technology, smart contracts, and DeFi mechanics. They're often early adopters of new protocols and technologies.
- Community Role: Degens play an important role in crypto ecosystems by providing early liquidity to new protocols, testing experimental features, and contributing to the rapid evolution of DeFi products.

2.4 General Terms

- Smol Brain
- Big Brain
- Gigabrain
- FUD (Fear, Uncertainty, Doubt): While not strictly a meme, it's become memefied in crypto culture. Any negative news or criticism is often dismissed as "FUD" by cryptocurrency enthusiasts, sometimes ironically.
- "When Lambo?": This phrase emerged during the 2017 bull run, representing the dream of getting rich enough from crypto to buy a Lamborghini. It's both used seriously by newcomers and ironically by veterans.
- Diamond Hands / Paper Hands: Popularized during the GameStop saga but heavily adopted in crypto. "Diamond hands" refers to holders who don't sell despite market pressure, while "paper hands" sell at the first sign of trouble.
- "This is good for Bitcoin": Originally used seriously, it became ironic as people would claim any news, even negative, was somehow positive for Bitcoin's adoption or price.
- Bogdanoff Twins ("Dump It"): These memes feature the late Bogdanoff twins supposedly controlling the crypto markets, ordering price dumps right after someone buys or pumps after they sell.
- "Funds are SAFU": From a Binance CEO video where he mispronounced "safe," this became a way to reassure (often ironically) about exchange security.

2.4.1 Common Characters

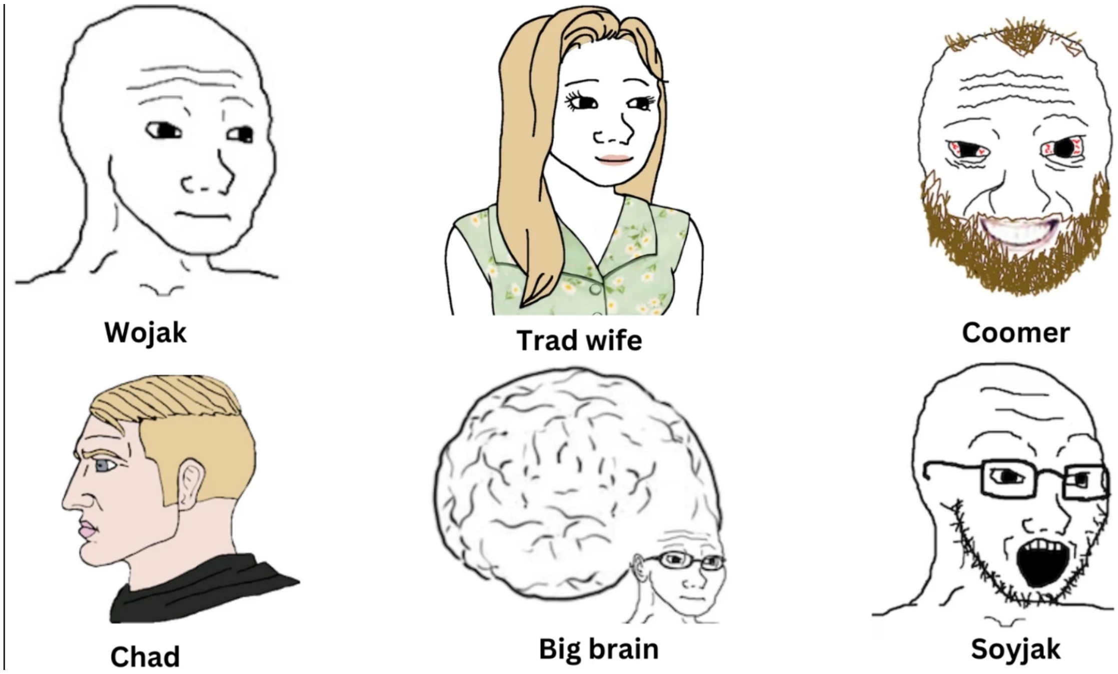

Figure 2.3: Crypto characters

- Wojak/NPC: Originally from 4chan, this simple face drawing represents the average retail trader who buys high and sells low. It's often paired with the "feels guy" emotion variations, particularly during market crashes.
- Soyjak: a derogatory variation of Wojak used to mock people perceived as overly enthusiastic, naive, or conformist. The term combines "soy" (from the controversial and scientifically unsupported claim that soy consumption reduces masculinity) with "Wojak."
- Coomer: a derogatory variation of Wojak used to mock people perceived as overly enthusiastic, naive, or conformist. The term combines "soy" (from the controversial and scientifically unsupported claim that soy consumption reduces masculinity) with "Wojak."

2.5 Acronyms

IBRL - Increase Bandwidth, Reduce Latency

3 Tools

3.1 Wallets

A wallet is the primary way you interact with a Blockchain.

Wallets have the following major attributes:

- Type - Hardware or Software wallets
- Platform - Mobile, Browser, Desktop or Hybrid
- Application Model - Type of blockchains it supports. Can support multiple models.

Table 3.1: Wallets

Wallet Name	Type	Platform	Application Models Supported
Ledger	Hardware	Desktop, Mobile	Account (EVM), UTXO, Specialized Account, Resource, Actor
Trezor	Hardware	Desktop, Browser	Account (EVM), UTXO, Specialized Account
MetaMask	Software	Browser Extension, Mobile	Account (EVM)
Phantom	Software	Browser Extension, Mobile	Capability (Solana), Account (EVM)
Safe	Software	Browser (Web App)	Account (EVM)
TrustWallet	Software	Mobile	Account (EVM), UTXO, Specialized Account
Rainbow	Software	Mobile	Account (EVM)

3.2 Analysis Tools

3.2.1 API

We'll start off by using Ortege. For full transparency I own and founded Ortege. This Almanack can and will use data from any trusted source. But in the early days while we require data specifically tailored to our requirements, we will be using Ortege. Anyone contributing however can integrate with any other reputable data sources.

We break down API's into the following categories:

- RPC Nodes. These are the API's provided by the Blockchain clients. Each Application Model will have a different API structure. The most famous example being the Bitcon RPC standard and the EVM RPC standard
- SQL based. These API's allow you to write SQL against their Datasets. Examples include Dune and Flipside. Much like the RPC based API's, these will all have a different API structure.
- Semantic Layer. Ortege is the first, that I'm aware of, API that aims to abstract away the blockchains implementation away from the developer and provide a uniform API that is consistent across blockchains.

3.2.1.1 Ortege

Coming Soon.

3.2.2 Explorers

3.2.2.1 Pure EVM

Etherscan is the most widely used and has versions on all the major EVM Chains.

Exploring Solidity

- Viewing the Solidity version
- Viewing the "standard" libraries the code inherits

3.2.3 Defillama

3.3 Security Tools

3.3.1 OpenZeppelin

3.4 Privacy Tools

3.4.1 DuckDuckGo

3.4.2 VPN

The most commercially popular VPN's are ExpressVPN, NordVPN and CyberGhost. I'm not an expert here, but I use Mullvad VPN as it accepts BTC as payment and seems most resitant to state attacks.

3.4.3 TOR

4 Memes

There will be a lot of references to meme's in your journey to become a Gigabrain Degen. It's important to understand them. Culture is a massive part of Web3 and will help you immensely in your journey.

4.1 HODL

HODL originated as a misspelling of "hold" in a 2013 Bitcoin forum post titled "I AM HODLING." The post's author, frustrated by their poor trading performance during a Bitcoin price crash, declared they would simply hold onto their Bitcoin rather than try to time the market. The typographical error caught on as both a meme and an investment philosophy in the cryptocurrency community.

The term has evolved beyond its humorous origins to represent a long-term investment strategy often summarized as "Hold On for Dear Life." This reflects the mindset of cryptocurrency investors who maintain their positions despite extreme market volatility, believing in the long-term value proposition of their assets rather than attempting to profit from short-term price movements.

HODL also represents a rejection of traditional trading wisdom. While conventional financial markets emphasize technical analysis and market timing, HODLing suggests that, particularly in the volatile cryptocurrency space, a simple buy-and-hold strategy might outperform active trading for many investors. This approach acknowledges the difficulty of timing a highly volatile, 24/7 global market.

The term has become so influential that it helps identify different types of market participants. "HODLers" are often contrasted with "traders" or "speculators," with HODLers viewed as providing market stability through their long-term commitment to holding assets. During market downturns, the community often encourages "HODL" as a way to maintain conviction and avoid panic selling.

The concept of HODLing has also influenced how blockchain projects design their token economics. Many projects now include staking mechanisms, vesting schedules, and other features that encourage long-term holding, showing how a simple misspelling has evolved into a fundamental aspect of cryptocurrency culture and economics.

4.2 Midwit (Bell Curve) Meme

The meme uses a bell curve (IQ distribution curve) with three characters:

- Left side (low IQ): A simple, happy figure who does something basic but effective. Often called Smol brain
- Middle of the curve (average IQ): A pseudo-intellectual character with overcomplicated strategies/thoughts, often wearing glasses and trying to sound smart
- Right side (high IQ): Returns to the simple approach but with deep understanding of why it works

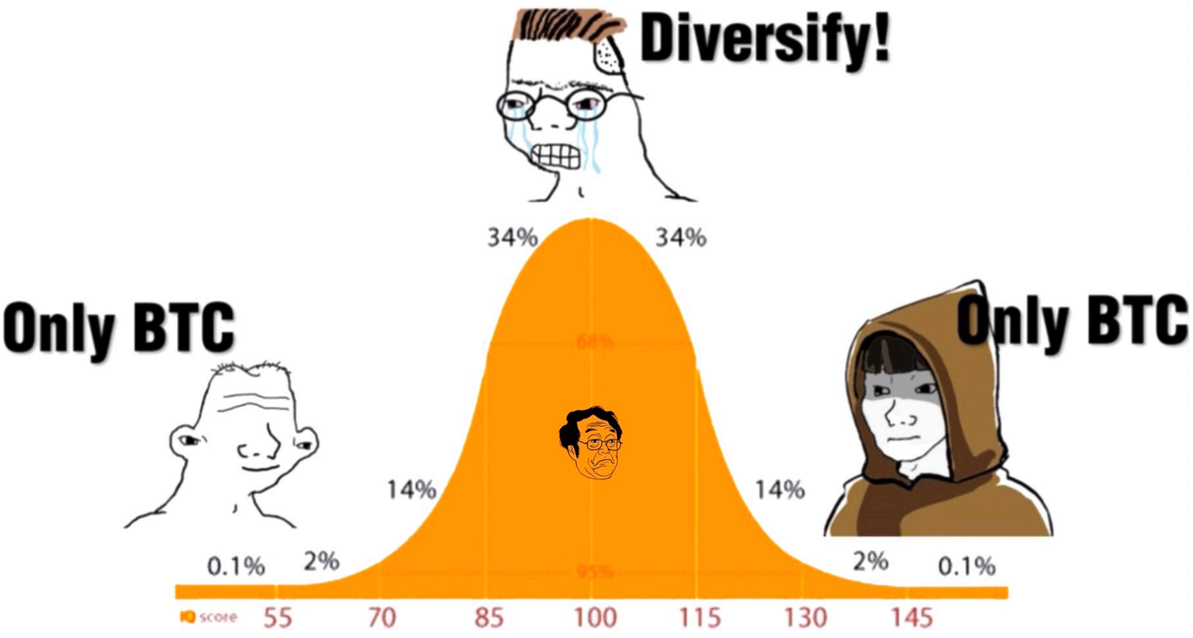

Figure 4.1: BTC Bell Curve meme

4.3 General

- “Sir, this is a Wendy’s”: Used when crypto discussions become overly technical or philosophical, reminding people not to take themselves too seriously.

> ℹ Note

5 Icons

5.1 Satoshi Nakamoto

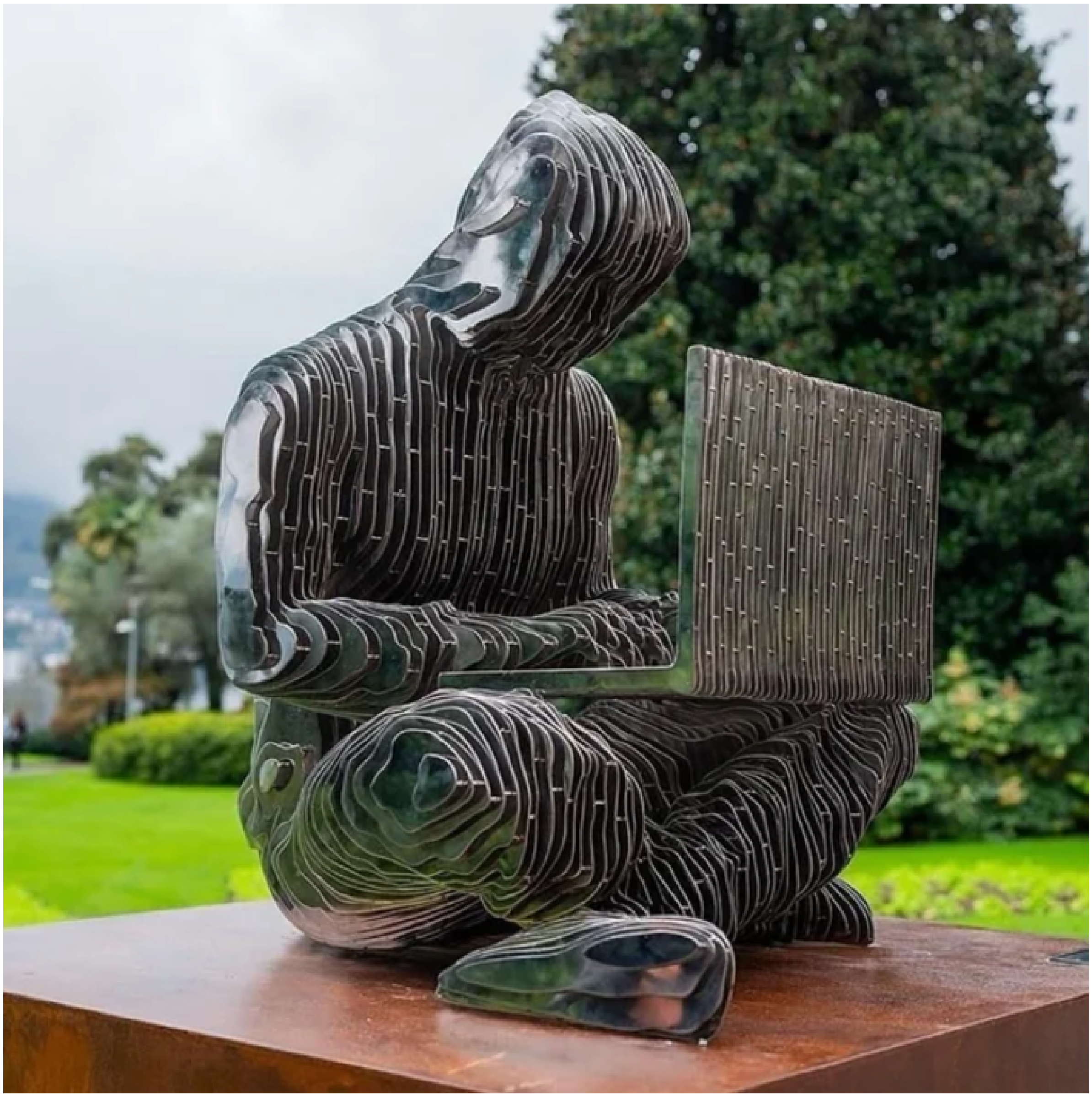

Figure 5.1: Satoshi artwork in Lugano, Switzerland

Satoshi is the most publicly known creator of our industry. He was preceded by the Cypherpunks but has engulfed the public's imagination due to the fact no-one categorically knows who he/she/they were.

5.2 People who protect us

Coffeezilla

Part II

Become a Degen

6 Don't lose your money

I'll be the first to admit that Web3 is full of sharks looking at newcomers like the chum you are. Before you even begin to tip your toes into this industry you need to be aware of the common pitfalls. Even after years of being in the industry, experienced degens can lose money to scams. If you want to make money and keep it, you need to be ever vigilant.

We break down the ways people can lose their money into the following categories:

- Private Key Management
- Technical User Errors
- Social Engineering and Scams
- Smart Contract Vulnerabilities and Hacks
- Rug Pulls and Exit Scams
- Market Manipulation
- Exchange and Platform Risks

6.1 Private Key Management

A large part of the early vision of crypto was to create a currency with the absence of a central authority (read Government). Indeed the first Bitcoin block, mined by Satoshi contained a message admonishing the central banks.

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

In order to have a currency not controlled by a central authority, you need to be able to exchange value with others while not relying on the authority of governments. If I give you 10 digital dollars, what's to prevent you from saying you never received it? Who proves the exchange occured? This is where Cryptography comes into it. We won't go into the technical details here, it's just important to note some general overview.

Your account/wallet on a blockchain is secured by a public-private key pair. You can think of the public key as akin to an email address and the private key as your password. Whenever you create a wallet and a seed phrase is generated, what's essentially happening is a public-private key is being generated.

Now whenever you interact with a blockchain, for example by sending someone some digital coins, you need to sign that transaction. What you're doing is essentially using your private key to sign that you do want to perform that transaction.

This is why keeping your private key safe is the most important thing you can do in Web3. Since we have no trusted authority, we can only rely on ourselves in this decentralized environment. If someone got access to your private key, they would be able to do anything with the assets that are owned by your account.

- Cold Storage
- Paper and Metal Storage
- Multiple Physical Locations
- Multisignature Wallets
- Glacier Protocol

6.2 Technical Errors

- Network selection error. If you send money to someone on a different network there is no guarantee the address will be the same
- Gas Fee misunderstandings
- Token Approval Dangers - never set max limits. Would be good to break down these risks by Application Model.
- Smart Contract Interaction Errors - Some UI's may not be linked to the actual smart contract you're looking for
- Impermanent Loss
- Wallet Address Verification
- Slippage and Frontrunning
- Sniping - new token launchs

6.2.1 Gas Fee misunderstandings

> ℹ Note

> ℹ Note

> ℹ Note

6.2.2 Impermanent Loss

Spoiler alert It's probably permanent

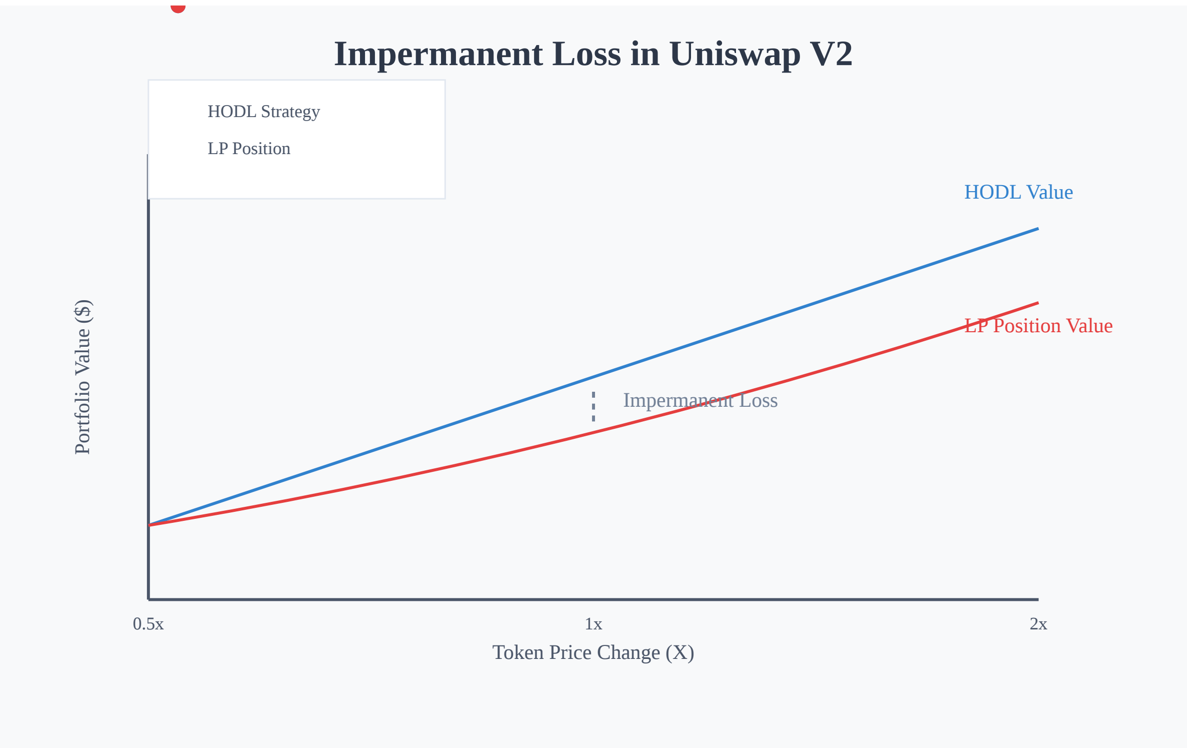

Figure 6.1: Chart showing Impermanent Loss

ℹ Note

6.2.3 Wallet Address Verification

This is one of the reasons Name Services have gained popularity. Although they do remove a further layer of anonymity if you're not careful.

6.3 Social Engineering Scams

- OneCoin. Ruja Ignatova (the "CryptoQueen") got away with $4 billion.

Common threads of social engineering scams * Authority Exploitation * Emotional Manipulation * Technical Obscurity * Community Building * Urgency Creation

Authority Exploitation While no mean authoritive, these are some of the biggest dodgy celebrities: * Kim Kardashian * Floyd Mayweather * DJ Khaled * Ben Phillips * Lindsay Lohan * Steven Seagal * Paul Pierce * Logan and Jake Paul

6.3.1 Network addresses

Scammers put in fake networks. Use Chainlist

6.4 Smart Contract Vulnerabilities and Hacks

Smart contract vulnerabilities represent one of the most critical risks in Web3, having led to billions in losses. This analysis examines major exploit categories based on historical hacks, their prevalence across different application models, and effective mitigation strategies.

- Access Control Flaws
- Logic and Arithmetic Errors
- Oracle Manipulation
- Reentrancy attacks
- Flash Loan Vulnerabilities
- Reentrancy attacks

7 Make some money

7.1 Purchasing your first Crypto

While as much as possible I'd like to respect and adhere to the ethos of the early Cypherpunks, there's also the practicality of expecting newcomers to be able to perform some of the advanced requirements that are needed. So we'll take a pragmatic approach and introduce those concepts in incremental levels of difficulty.

Important things to note: * You'll need about 10 US Dollars * We'll be using CEX's to purchase, while understanding the risks and implications * We'll use a Mobile wallet, Phantom, that has social recovery

So the first thing anyone entering Web3 needs to do is to purcahse some Crypto. In this Chapter we'll walk you through the process, with the important caveats below:

- You'll need at least 6 US shit coins. Or in normie speak, $5 US Dollars.

We will use Coinbase as the CEX for this section. You're more than welcome to use your local CEX. For users in countries that don't have access to CEX's, we'll detail how they can purchase their first crypto with Bisq.

Due to me wanting anyone to be able to get into crypto, I'm pretty adamanant that it shouldn't cost more than $5 for people to learn. This is why I'd like to tailor this to start off with people purchasing ETH

https://bisq.network/downloads/

I'm using the Bisq 2 version, although at the time of writing, it's still in beta. There is a minimum requirement of 6 USD. The median income is approximately $7 per day.

- The poorest 10% of the world live on less than $2 a day
- About 50% live on less thatn $7 a day

On Bisq the minimum amount I could find someone willing to trade with me is 10 USD. Also it's sold at a huge premium. Further disenfranchizing the world's poor. As such if anyone wants to purchase Bitcoin for $10 please reach out to me Bisq kinda sucks. Hodl Hodl looks like a much better solution

Get them to create an account at a CEX. Explain KYC/AML

Talk about why you never hold funds on a CEX. Go through FTX, Mount Gox and other failures. Then talk about wallets.

We're going with Phantom as it supports: * Bitcoin * Ethereum * Solana * Base * Polygon

As of writing, TVL across Ethereum has 54%, Solana 7%, Base and Bitcoin both ~3%. So while I'd love a wallet that supports Stacks, Move and Cosmos. That just doesn't exist today. So as a first time Web3 user, Phantom just makes the most sense. I also really wanted users first wallet to have social recovery. But no wallet currently supprots this. Argent used to, but they're now only on Starknet and Soul wallet has no downloadable version I could find.

Need to talk about storing of private key

- Physical storage. These have a shit user experience, but are relatively safe. Hardware
- Digital storage. Password manager, social recovery will come at some point

So we'll purchase ETH, transfer it to our Phantom wallet, and then stake it on Lido

7.2 Accounts

Here we'll go through Standard Accounts (EOA's) versus Smart Accounts. Which blockchains support the different paradigms.

Issue with social recovery is that you leave a fingerprint to your on chain activity depending on how the wallet handles the storing of your data

The first place you're going to have to start is actually converting fiat into crypto.

You can do this via:

- mining
- purchasing

We'll only talk about purchasing here, as solo mining is not profitable, although if you're not running at a loss, then it can still be considered away of converting fiat into crypto.

- Privacy respecting ways
  - Peer to Peer (P2P) Marketplaces
  - Bitcoin ATMs
- Non Privacy respecting ways
  - CEX's
  - OTC
  - DeFi on ramps

For complete privacy, the place where you purchase

7.3 Airdrop Farming

8 Gigabrain Degen

8.1 Converting FIAT into a Cryptocurrency

This may be the hardest part of being a Degen. A Degen values privacy. Important to distinguish there is a difference between privacy and secrecy. I'd like my cryptocurrency to display the same characteristics as cash. A means of exchange that has been working just fine for centuries. Cash is private by default and people have seemed pretty happy with that arrangement for over 3 centuries.

Currently the best way to do this is via:

- Bisq
- HODL HODL

Part III

Financial

9 Digital Assets

The logical place to start this Almanack is Digital Assets. These include all the financial instruments that exist within the Web3 sphere. For us to represent all Digital Assets within this space, it means we need to include all Web3 financial instruments, both on chain and off chain.

We take inspiration from the Fat Protocol Thesis(Monegro 2016) to define the major categories of Digital Assets within the Web3 realm. We also adhere to the naming convention that stipulates Coins are digital assets relating to the running and operation of a Blockchain, whereas Tokens are digital assets that are issued on a Blockchain.

We thus break down Digital Assets into the following Categories:

- Coins
  - Primary Networks
  - Secondary Networks
    - Ozempic
    - Sugar
  - Derivatives
- Tokens
  - Fungibles
  - Non Fungible

After describing the various categories and classes of Digital Assets we'll then delve into the Markets existing for these Digital Assets, as well as how an entity hodls the Digital Asset and the yield properties of the various types of Digital Assets.

9.1 Coins

This section deals primarily with Digital Assets as a Financial Instrument and as such any information relating to the technical makeup can be found in the Blockchain documentation.

9.1.1 Network Economics

Token Models Utility Tokens: Gas fees, staking, governance Security Tokens: Validator requirements, slashing deposits Network Tokens: Transaction fees, block rewards Incentive Structures Validator Rewards: Block rewards, transaction fees, staking yields User Incentives: Fee markets, priority mechanisms, rebate systems Developer Incentives: Grant programs, protocol fees, treasury funding Economic Security Minimum Stakes: Validator requirements, delegation minimums Slashing Conditions: Downtime penalties, malicious behavior penalties Market Making: Liquidity incentives, trading pair support

9.1.2 Base Networks

This Almanack distinguishes between the financial properties of Coins versus the technical properties of a Blockchain. As such we don't refer to networks here by Layer 1 or Layer 2. That's a classification and distinction you can explore here.

We define a Base Network that maintains it's own Sovereignity. This means that the Coin on the network is used to economically secure the chain as well as final settlement to occur on this network.

9.1.3 Secondary Networks

These are networks that market themselves as settling the transactions on another network. The nuances of how they settle is covered under the Blockchain chapter. We'll encompass the full breadth of L2's including, but not limited to Plasma, Sidechains, Rollups etc.

We then break these networks into Ozempic or Sugar networks. This reflects a hat tip to the Fat protocol metaphor. Ozempic networks are net extractors of value from their host chain, while Sugar networks cause the host chain to become fatter and therefore hold more value.

Please see Ozempic Effect to see how we determine if a network in a net ozempic or sugar network.

9.1.4 Derivatives

- On chain
  - Wrapped
    - Pure
    - Bridged
- Off Chain
  - Spot ETF's

9.1.4.1 On Chain Derivatives

9.1.4.2 Off Chain Derivatives

9.2 Tokens

9.2.1 Fungible

Fungible is a pretty terrible name, but it roughly means divisable. It's easier to explain via an example. If I have 10 dollars and I give you 3. I still have 7. It's divisable. If I have a car and I want to give you 30% of it, I cannot cut it up and give you a portion of it. It's Non fungible, or non divisable. There's more nuances which we can deal with in the vocabulary section. But that's the general idea of it.

We have the following types: * Stable Coins * Fiat backed * Crypto backed * Delta Neutral backed * Shit Coins * Governance * Meme * Utility

Then we have numerous standards. We'll add only the most popular and relevant ones here.

Namely: * ERC20 * ERC777 * ERC1363 * BRC20 * Runes * Solana's SPL Token Standard * ICS20

9.2.1.1 ICS20

The Inter-Chain Standard 20 is a Cosmos based standard for fungible token transfers between blockchains using the Inter-Blockchain Communication Protocol (IBC)

9.2.2 Non Fungible

Has the following attributes:

- Art & Collectibles
- Profile Picture (PFP)
- Gaming
- Domain Names and Identity
- Real World Assets (RWA)

The NFT floor price is the lowest price at which an NFT from a particular collection is listed for sale on a marketplace. It serves as a benchmark for the collection's market value and is widely used to assess the entry point for potential buyers and to gauge the collection's popularity and liquidity.

9.3 Markets

Where can I buy these Digital Assets? The major markets are regulated and unregulated.

These are then divided into spot vs derivatives.

For regulated it's interesting as RedBelly in a blockchain, but it has KYC/AML. So is that regularly compliant. It's probably more truthful to break it down not by regulatory compliance, but anonymity. For if your on chain activity can be tracked then most mature jurisdictions will be able to force an individual to be compliant.

9.4 Hodling

Wallets

9.4.1 Custodial

9.4.2 Non-Custodial

- Pure
  - Cold
  - Hot
- Smart
  - MPC
  - Smart Contract Based (Includes Account Abstraction)

9.5 Yield Properties

Major categories are:

- Network Yield
- Trading Yield
- Protocol Yield

9.5.1 Trading Yield

- Pricing appreciation
- Arbitrage Yield
  - Spot Arbitrage
  - Peg Arbitrage - These are unique to stable coins
- Options and Derivatives Premiums
- Futures funding rates

9.5.2 Network Yield

- Mining Yield
  - Solo Mining
  - Pool Mining
  - Cloud Mining
- Validating Yield
  - Staking
- Network Fee Yield
  - Gas
- MEV
  - Toxic
  - Non Toxic
- Derivatives
  - Liquid Staking

9.5.3 dApp Yield

- Staking
- Liquidity Provision
- Governance Participation
- NFT Rental Income

9.5.3.1 Liquidity Provision

We break this down into the following Categories

- AMM Pool fees
- Concentrated liquidity positions
- Order book market making
- Lending and Borrowing markets

9.5.3.1.1 Concentrated Liquidity Provision

Here we will break down rebalancing and focus on Loss Versus Rebalancing as per this paper https://arxiv.org/pdf/2208.06046

10 Ratings

10.1 Core Rating Categories (60% of Total Rating)

10.1.1 1. Protocol Value Capture (30%)

A. Network Effect Metrics - Daily Active Users (DAUs) - Total Value Locked (TVL) - Transaction volume - Fee revenue generated - Protocol revenue retained

B. Value Accrual Mechanisms - Token economics design - Fee distribution model - Staking mechanisms - Burns and supply dynamics

C. Ozempic Network Effects - Value extraction efficiency from L1 - Transaction fee capture rate - User migration metrics from L1 - TVL migration patterns - Gas savings versus L1 - L1-L2 Value Dynamics - Sequencer revenue distribution - MEV capture and distribution - Bridge volume and efficiency - Settlement layer costs - Sustainable Value Creation - Net new users versus L1 migration - Ecosystem-specific applications - Novel transaction types impossible on L1 - Cross-L2 interoperability metrics

10.1.2 2. Protocol Security & Risk Assessment (30%)

A. Smart Contract Security - Audit history and quality - Bug bounty program effectiveness - Historical vulnerability incidents - Code complexity metrics - Upgrade mechanism security - Testing coverage

B. Network Security - Consensus mechanism robustness - MEV exposure and protection measures - Node distribution - Network attack resistance - Cross-chain bridge security - Oracle dependency and security

C. L1 Dependency Risks - Settlement layer congestion exposure - Bridge security and liquidity depth - L1 fee market correlation - Sequencer centralization risk - Value extraction sustainability

10.2 Risk Categories (40% of Total Rating)

10.2.1 1. Technical Risk Assessment (15%)

A. Smart Contract Vulnerabilities - Code audit findings severity - Time-tested deployment - Complexity of interactions - Dependencies on external protocols - Historical incident analysis

B. Network Level Risks - MEV exposure metrics - Network partition resistance - Node centralization factors - Infrastructure dependencies - Cross-chain vulnerability exposure

C. Key Management & Wallet Security - Multi-sig implementation - Key generation processes - Hardware security modules usage - Social recovery mechanisms - Access control systems

10.2.2 2. Economic Risk Assessment (10%)

A. Market Dynamics - Liquidity concentration - Price impact resistance - Collateral quality - Market manipulation resistance

B. Economic Model Vulnerabilities - Game theory attack vectors - Incentive alignment analysis - Economic exploit resistance - Stress test scenarios - Flash loan attack surface

10.2.3 3. Operational Risk Assessment (10%)

A. CeFi/CeDeFi Risks - Centralization points - Custody arrangements - Third-party dependencies - Operational redundancy - Emergency procedures

B. Oracle Dependencies - Oracle manipulation resistance - Price feed reliability - Backup oracle systems - Historical oracle incidents - Data quality metrics

10.2.4 4. External Risk Assessment (5%)

A. Regulatory Risk - Jurisdictional exposure - Compliance frameworks - Regulatory clarity - Legal structure - Historical regulatory interactions

B. Social Engineering Risk - Team security practices - Access control policies - Social attack history - Security awareness training - Incident response readiness

10.3 Risk-Adjusted Rating Scale

AAA: Exceptional protocol with comprehensive risk mitigation - Multiple independent security audits with no critical findings - Proven resistance to all major attack vectors - Strong regulatory compliance framework - Decentralized operations with minimal points of failure - Multiple layers of economic security

AA: Strong protocol with robust risk management - Regular security audits with minor findings - Documented resistance to common attack vectors - Clear regulatory strategy - Limited centralization risks - Strong economic security measures

10.4 Risk Multipliers

Each risk category can apply a multiplier to the base rating: - Critical Risk: -3 rating notches - High Risk: -2 rating notches - Medium Risk: -1 rating notch - Low Risk: No adjustment - Minimal Risk: +1 rating notch

10.5 Continuous Monitoring Triggers

- Smart contract vulnerability disclosure
- Network attack detection
- Regulatory action
- Economic model stress
- Oracle deviation events
- Cross-chain bridge incidents
- Social engineering attempts
- MEV activity spikes

10.6 Review Framework

- Monthly security metric review
- Quarterly risk assessment update
- Annual comprehensive review
- Real-time monitoring of critical indicators
- Incident-triggered reassessment

10.7 Ozempic Effect

We'll base this upon value flows. Defillama doesn't actually display this. So we'll need to get this data directly from the smart contracts. We can start with Base, Arbitrum, BSC, Optimism and Polygon.

Let's build a comprehensive framework for tracking the true "Ozempic effect" of L2s on Ethereum. We'll need several interconnected metrics to understand the complete value flow dynamics.

1. Wallet Migration Analysis

- Track addresses that first appeared on Ethereum before a certain date (let's call them "Ethereum Native Wallets")
- Monitor their activity transition to L2s over time
- Analyze their ETH holdings distribution between L1 and L2s
- Calculate the ratio of their transaction activity on L2s versus L1

2. L2 Native User Analysis

- Identify wallets that first appeared on L2s
- Track what percentage never bridge to Ethereum
- Measure their total value held
- Calculate their transaction activity

3. Fee Flow Dynamics

- Track L2 sequencer fees paid back to Ethereum
- Calculate the net fee difference (what these transactions would have cost on L1 versus actual L2 + L1 calldata costs)
- Monitor the ratio of fees paid back to Ethereum versus fees retained by the L2

We could create a composite "Value Migration Score" that looks like:

```
def calculate_migration_score(l2_data):
    # Value Migration
    eth_native_migration = (
        eth_native_wallets_on_l2 / total_eth_native_wallets *
        eth_value_moved_to_l2 / original_eth_holdings
    )

    # Activity Migration
    activity_migration = (
        l2_transactions_by_eth_wallets /
        (l1_transactions_by_eth_wallets + l2_transactions_by_eth_wallets)
    )

    # L2 Native Growth
    l2_native_strength = (
        l2_only_wallets / total_l2_wallets *
        l2_native_value / total_l2_value
    )

    # Fee Economics
    fee_efficiency = (
        sequencer_fees_to_ethereum /
        theoretical_l1_fees_for_same_transactions
    )

    return (
        eth_native_migration * 0.3 +
        activity_migration * 0.3 +
        l2_native_strength * 0.2 +
        fee_efficiency * 0.2
    )
```

This would give us a quantitative measure of how effectively an L2 is: 1. Attracting value from Ethereum 2. Generating new value through native users 3. Maintaining a sustainable fee relationship with Ethereum

A true Ozempic L2 should show: - High migration of ETH native wallets - Strong L2 native user growth - Efficient fee generation that still provides meaningful revenue to Ethereum - Sustained or growing activity ratios over time

Would you like me to elaborate on how we could implement the data collection for any of these metrics, or should we explore additional factors to consider in the scoring?

- Net Value Extraction Rate = (Value Bridged In - Value Bridged Out) / Total Time Period

10.8 Risk Factors

10.8.1 1. Technical Risks

- Implementation bugs
- Security vulnerabilities
- Upgrade coordination failures

- Network splits

10.8.2 2. Governance Risks

- Capture by special interests
- Low participation
- Contentious decisions
- Unclear processes

10.8.3 3. Social Risks

- Community fragmentation
- Loss of developer support
- Reputation damage
- Market impacts

10.9 Future Considerations

10.9.1 1. Emerging Trends

- Automated governance systems
- AI-assisted proposal analysis
- Cross-chain governance
- Dynamic parameter adjustment

10.9.2 2. Challenges

- Scaling governance participation
- Balancing security and innovation
- Managing increasing complexity
- Maintaining decentralization

10.9.3 3. Opportunities

- Improved governance tools
- Better simulation capabilities
- Enhanced coordination mechanisms
- More sophisticated voting systems

10.10 Dependent Network Ratings

Polygon zkEVM Settlement Guarantees (10/10 weight): Score: 9/10 The zkEVM uses zero-knowledge proofs to validate all state transitions. Every transaction batch includes a proof that mathematically demonstrates the correctness of all computations and state changes. These proofs are verified by Ethereum's consensus mechanism, providing cryptographic certainty that state transitions are valid. This is nearly the highest level of settlement guarantee possible, just slightly below fully integrated L2s because of some optimizations in the proving system.

Dispute Resolution (9/10 weight): Score: 10/10 Ethereum serves as the absolute source of truth for the zkEVM. If there's ever a dispute about the state, the zero-knowledge proofs verified by Ethereum's consensus provide mathematical certainty about what is correct. There's no dependency on fraud proofs or challenge periods - the cryptographic proofs mean disputes are resolved immediately and with absolute certainty by Ethereum.

Economic Security Inheritance (8/10 weight): Score: 9/10 The zkEVM inherits its fundamental security from Ethereum. The validity proofs mean it can't confirm invalid state transitions, and its assets are secured by Ethereum's consensus mechanism. While it has its own token (MATIC) for gas fees and other purposes, the core economic security - particularly for assets like ETH and tokens - comes directly from Ethereum.

State Finality (7/10 weight): Score: 9/10 Once Ethereum confirms a zkEVM batch and its validity proof, that state is final with the same guarantees as Ethereum itself. The mathematical nature of the zero-knowledge proofs means there's no waiting period for finality beyond Ethereum's own finality period. This is as strong as state finality can get for a settlement-dependent network.

Exit Rights Guarantees (6/10 weight): Score: 8/10 Users can always withdraw their assets back to Ethereum by submitting a withdrawal request. These withdrawals are guaranteed by Ethereum's consensus - once a withdrawal is proven valid through a ZK proof, no one can prevent the user from claiming their assets on Ethereum. The only limitation is the normal proving and processing time.

State Progression Dependency (5/10 weight): Score: 7/10 While the zkEVM can process transactions independently, it can't finalize new states without submitting proofs to Ethereum and having them verified. This creates a strong dependency on Ethereum for state progression, though there's some independence in transaction processing.

Asset Movement (4/10 weight): Score: 10/10 Native ETH and ERC-20 tokens can move seamlessly between Ethereum and the zkEVM. When assets move to the zkEVM, they're locked on Ethereum and can only be unlocked through valid proofs. This provides the strongest possible guarantees for asset movement between the networks.

Total Score: 62/70 (approximately 89%)

This analysis places Polygon zkEVM firmly in the "Fully Non-Sovereign" category. Arbitrum Let me analyze Arbitrum's relationship with Ethereum as a Settlement-Dependent Network by carefully examining each criterion. This will help us understand how optimistic rollups differ from ZK rollups in their settlement dependency.

Settlement Guarantees (10/10 weight): Score: 7/10 Arbitrum uses an optimistic rollup design where transactions are assumed valid but can be challenged during a dispute period (currently 7 days). While this provides strong settlement guarantees, it's not as immediate or mathematically certain as ZK rollups. The challenge period introduces a time-based element to settlement finality. However, the ability to prove fraud on Ethereum's consensus layer still makes this a robust settlement mechanism.

Dispute Resolution (9/10 weight): Score: 8/10 Ethereum serves as the ultimate arbiter for Arbitrum through its fraud proof system. If someone identifies an invalid state transition, they can submit a fraud proof to Ethereum, which will automatically resolve the dispute and revert invalid transactions. This is strong dispute resolution, though not as immediate as ZK proofs since it requires active challengers and a challenge period. The key strength is that Ethereum's consensus automatically enforces the correct resolution once fraud is proven.

Economic Security Inheritance (8/10 weight): Score: 9/10 Arbitrum inherits its fundamental security from Ethereum. The ability to prove fraud on Ethereum means that any attempt to corrupt Arbitrum's state would require corrupting Ethereum itself. The sequencer role adds some centralization risk, but the fundamental economic security - particularly for assets - comes directly from Ethereum. Users can always force transactions through Ethereum if the sequencer fails.

State Finality (7/10 weight): Score: 6/10 While Arbitrum's state updates are recorded on Ethereum, true finality requires waiting through the challenge period. This creates a tradeoff between practical finality (which can be quite fast) and absolute finality (which requires waiting for the challenge period). This is lower than ZK rollups where finality is immediate once proofs are verified.

Exit Rights Guarantees (6/10 weight): Score: 8/10 Users can always withdraw their assets to Ethereum, guaranteed by Ethereum's consensus. While withdrawals require waiting through the challenge period, they cannot be prevented by Arbitrum's operators. The delay is longer than with ZK rollups, but the guarantee is just as strong once the period passes.

State Progression Dependency (5/10 weight): Score: 7/10 Arbitrum can process transactions independently but must submit state roots to Ethereum for potential verification. While it has more processing independence than some systems, it ultimately depends on Ethereum for final state confirmation, especially during disputes.

Asset Movement (4/10 weight): Score: 10/10 Native ETH and ERC-20 tokens move seamlessly between Ethereum and Arbitrum through a strong bridge mechanism backed by Ethereum's consensus. When assets move to Arbitrum, they're locked on Ethereum and can only be unlocked through valid withdrawals after the challenge period.

Total Score: 55/70 (approximately 79%)

This analysis places Arbitrum in the “Fully Non-Sovereign” category, though with a lower score than Polygon zkEVM. The main differences come from the challenge period required for absolute finality and the reliance on fraud proofs rather than validity proofs.

Stacks Settlement Guarantees (10/10 weight): Score: 4/10 Stacks uses Bitcoin for checkpointing and security anchoring However, it lacks cryptographic enforcement of settlement by Bitcoin’s consensus Bitcoin doesn’t automatically enforce or validate Stacks’ state transitions Falls into the “checkpoint systems” category rather than stronger settlement guarantees Dispute Resolution (9/10 weight): Score: 3/10 While Stacks records its state on Bitcoin, Bitcoin’s consensus doesn’t serve as the ultimate arbiter Disputes are primarily resolved within Stacks’ own consensus mechanism Bitcoin can’t automatically correct or resolve issues in Stacks’ state Economic Security Inheritance (8/10 weight): Score: 6/10 Miners must commit actual Bitcoin through PoX mechanism This creates some economic security dependency on Bitcoin However, Stacks maintains its own economic incentives through STX State Finality (7/10 weight): Score: 5/10 Stacks achieves finality through a combination of its own consensus and Bitcoin anchoring State is recorded on Bitcoin but not in a way that Bitcoin consensus enforces Provides stronger finality than fully independent chains but weaker than true L2s Exit Rights Guarantees (6/10 weight): Score: 4/10 With sBTC, users can move Bitcoin between chains However, this relies on Stacks’ mechanisms rather than being guaranteed by Bitcoin’s consensus Exit rights depend on threshold signatures rather than cryptographic guarantees State Progression Dependency (5/10 weight): Score: 7/10 Stacks blocks are linked to Bitcoin blocks through PoX State progression is tied to Bitcoin’s block progression However, Stacks can still process transactions independently within this framework Asset Movement (4/10 weight): Score: 5/10 sBTC enables Bitcoin movement between chains But this movement isn’t directly enforced by Bitcoin’s consensus Relies on threshold signatures and Stacks’ mechanisms Total Score: 34/59 (approximately 58%)

This places Stacks in the “Moderately Dependent” category on our spectrum.

10.11 Risks

Sub categories of risks include smart contract vulns regulatory risks centralization risks for CeDeFi and CEX’s MEV network level key and wallet compromise cross chain exploits oracle manipulation Social engineering exploitation of economic models

11 Trollip's Index

11.1 Bitcoin

11.2 Solana

11.3 Monero

11.4 Optimism

11.5 Ethereum

11.6 Filecoin

11.7 Starknet

11.8 Lido Staked Ethereum

11.9 Chainlink

11.10 Sui

11.11 Uniswap

11.12 Celestia

11.13 XRP

11.14 Dogecoin

11.15 Tether USDT

11.16 Circle USDC

11.17 Ethena USDe

11.18 Aptos

11.19 Arbitrum

Part IV

Technical

12 Cryptography

12.1 Symmetric Cryptography

Shared Keys. AES. But vulnerability is in sharing the Key between parties.

12.2 Asymetric Cryptography

Public Key Cryptography

RSA and elliptic curve cryptography. Slow than symetric cryptography. Enables secure communication over insecure channels

12.3 Hash functions

These are like digital fingerprints

12.4 Zero-Knowledge Proofs

They solve a unique problem: how can you prove you know something without revealing what you know? Imagine proving you're over 21 without showing your actual birthdate.

12.5 Homomorphic Encryption

It allows computations to be performed on encrypted data without decrypting it first. Think of it like being able to ask someone to bake a cake following your secret recipe, but without ever revealing the recipe to them. While still computationally intensive, this technology could revolutionize cloud computing and data privacy.

12.6 Post-Quantum Cryptography

cryptographic systems that can resist attacks from quantum computers. This forward-looking field is essential because many current cryptographic systems (especially asymmetric ones) could be broken by powerful quantum computers.

13 Blockchains

We would prefer to use the term Decentralized Incentive Network instead of blockchain. But we also want to align on common nomenclature within the industry, so we'll stick with Blockchain for now.

The point of blockchains:

- Censorship resistance
- Deterministic State Transition (Mempool creates interesting nuances and complexity here)
- Credible neutrality
- Trustless coordination at scale
- Asset digitization and ownership

Censorship resistance is indeed crucial - it enables truly permissionless systems where no entity can prevent valid transactions. However, there are several other compelling candidates for blockchain's primary purpose: One strong contender is deterministic state transition. Think about how traditional databases or financial systems might have ambiguity about the exact sequence or timing of transactions, especially across different locations or institutions. Blockchain provides absolute clarity about state changes - there's no ambiguity about which transaction came first or what the exact state was at any given moment. This property enables complex financial systems and smart contracts to operate with complete predictability. Another fundamental purpose could be creating credible neutrality in computational systems. This goes beyond just censorship resistance - it's about creating systems where the rules are explicit, unchangeable without consensus, and apply equally to all participants. Traditional systems often have hidden biases or special privileges for certain users, while blockchain systems enforce their rules uniformly through code. We could also argue that the primary purpose is enabling trustless coordination at scale. Before blockchain, coordinating economic activity among untrusting parties required trusted intermediaries like banks or governments. Blockchain enables direct peer-to-peer coordination without these intermediaries, potentially reducing costs and increasing efficiency. Asset digitization and ownership might be another contender. Blockchain enables digital scarcity and verifiable ownership in ways that weren't possible before. This property enables everything from cryptocurrencies to NFTs to tokenized real-world assets.

We won't be taking any idealogical approach to the point of blockchains but rather applying scores to DIN's and then comparing that to how the market values them.

A decentralized incentive network with state consensus

I've tried to simplify our definition as much as possible and even still it's verbose. Read that sentence to 99% of the world and I'll they'll give you a quizzical look and move onto the section about meme coins. There's a lesson there. A lesson the technical luminaries in this space ignore at their peril. Accessibility matters.

Let's break down each component of our definition.

Decentralized. This is how many Nodes participate. Centralized means 1. Therefore anything more than 1 is decentralized.

Network. This loosely means the participants computers/machines communicate via a communication method.

State Consensus. The network will record state and all nodes will agree about the state via consensus.

We don't need to worry too much about consensus yet, but you just need to know that the nodes must agree on the state. For example if Bob sends Alice 10 BTC. Everyone on the network must come to an agreement that Bob did indeed send Alice 10 BTC. Another important characteristic is that even if everyone agrees today, in the future that can't be disputed or changed. This is another important category called censorship resistance. We'll get to it.

Dependent Networks

A sovereign chain can be defined as a network that maintains complete independence in its settlement process, requiring no external chain to validate or guarantee its state transitions. This independence is fundamental to the concept of sovereignty in blockchain networks.

This section should start discussing why the need arose for dependent networks. Specifically Ethereum struggling with scale. Due to the trilemna: decentralization security scalability

Let's talk about scaling strategies monolithic Bigger blocks Faster blocks Higher minimum requirements for nodes Consensus optimizations modular Execution - The current crop of L2's Data availability - Storing and accessing blockchain data Celestia Consensus - Agreeing on the state of the network Settlement - Finalizing transactions and providing security guarantees. Ethereum and Bitcoin in the current Layer craze

13.1 Sovereign Networks

Pure Value Networks Pure value networks stick to the fundamental purpose of moving and storing value, avoiding the additional complexity that comes with being a platform for applications or other services.

What makes these networks "pure value" is what they don't include: They don't support complex smart contracts They don't host decentralized applications They don't provide programmable functionality beyond basic value transfer They don't serve as platforms for other tokens or applications

Application Chains Application Chains represent networks that go beyond pure value transfer to support various types of applications. They're divided into three main categories: Hubs, Specialized, and Generalized chains.

Hub Application Chains serve as central connection points in the blockchain ecosystem.

Specialized Application Chains focus on specific use cases or industries.

Generalized Application Chains aim to support a wide range of applications but differ from hubs in that they're not necessarily trying to be central connection points.

The key distinction between these categories lies in their approach to applications: - Hubs prioritize becoming central platforms that other networks depend on - Specialized chains optimize everything for specific use cases - Generalized chains provide broad functionality but focus on being self-contained ecosystems

13.2 Layer 2's

We break these down into:

- Rollups
  - Optimistic
  - ZK
- Validiums
- Plasma
- Sidechains

So Ethereum has chosen decentralization and security. This means Ethereum is shit at scaling.

So we need to talk about the theory of Layers in Blockchain

That's why L2's became a major part of the roadmap. How does the scalability limitation affect the network. Congestion and high gas fee's. So people don't interact on the chain and it hits a ceiling. So logically we should start any discussion with transaction fee's, as that's the direct result of the scalability failure and the main reason users use L2's. Besides yield opportunities.

So we should first consider the Fee Markets on Ethereum and Bitcoin which are the two major chains looking to scale with L2's. There is a lot of debate and a lot of misinformation around this topic.

Dependent networks can really be divided into chains: External DIN required to validate state transitions External DIN required to guarantee state transitions

We'll define a framework for how we define a Dependent Network. There are few things we look at, in terms of priority: censorship resistance. Polygon zkEVM sequencer can censure, but you can go through the contract directly on Ethereum theoretically if censorship occurs. Same thing for Arbitrum. Although this is a very weak guarantee due to the complexity for regular users. For Stacks, it's more censorship resistant that the other two. Also if I were to interact directly with the smart contract on Ethereum, that makes the L2's pointless. Why do I need them if not to deal with congestion and high gas. dispute resolution. I believe all chains need to hard fork. Potentially not Movement due to Move's resource model. None of the chains will automatically hard fork if a dispute is found. settlement. So the base chain must validate and guarantee state transitions on the dependent chain. This is kinda stupid. It's guaranteeing the state transitions but makes no attempt to resolve things like censorship resistance. finality of settlement. Optimistic 7 days, zk 30 minutes, Stacks 16 hours economic security model inheritance - this is actually pointless as it just means the state transitions match the rules. Nothing about censorship resistance.

So my qualification of dependent networks means that they must inherit the censorship resistance of the base chain, settlement must be validated and guaranteed by the base chain, dispute resolution must be automatic and/or decentralized. else the chain can just fork.

Settlement Types In order of strength Cryptographically enforced settlement Validity proofs Fraud proof systems Checkpoint systems

Core Infrastructure Components Network Security Attack Vectors Network Level: Eclipse attacks, BGP hijacking, DDoS attacks Protocol Level: Double spending, nothing-at-stake, long-range attacks Application Level: Smart contract vulnerabilities, oracle manipulation Defense Mechanisms Network Diversity: Geographic distribution, client diversity, network topology Economic Security: Slashing conditions, required stake amounts, timelock mechanisms Operational Security: Key management, upgrade processes, emergency responses Monitoring and Detection Network Health Monitoring: Peer count, block propagation, chain quality Threat Detection: Fork monitoring, unusual transaction patterns, validator behavior Incident Response: Alert systems, mitigation procedures, recovery processes Network Governance Governance Models On-Chain: Direct token voting, delegated voting, quadratic voting Off-Chain: Foundation-led, core team decisions, community forums Hybrid: Combined on-chain execution with off-chain discussion Decision Domains Protocol Parameters: Block size, gas limits, fee structures Treasury Management: Fund allocation, grant distribution, development funding Protocol Upgrades: Hard forks, soft forks, emergency changes Participation Mechanisms Proposal Systems: Submission requirements, discussion periods, voting periods Voting Methods: Token-weighted, identity-based, reputation-based Execution Methods: Automatic execution, timelock delays, manual implementation Cross-Chain Communication Bridge Types Trusted Bridges: Custodial, multi-signature, federated Trustless Bridges: Light clients, zero-knowledge proofs, relay networks Hybrid Bridges: Combined trusted and trustless elements Message Passing Protocol Standards: IBC (Inter-Blockchain Communication), cross-chain messaging Verification Methods: SPV proofs, merkle proofs, validity proofs Asset Standards: Wrapped tokens, synthetic assets, native bridges Security Considerations Bridge Security: Validator sets, challenge periods, fraud proofs Asset Security: Lockup mechanisms, minting controls, burning procedures Network Security: Cross-chain replay protection, nonce management

Network Performance Metrics Transaction Performance Throughput: Transactions per second (TPS), block size limits Latency: Block time, time to finality, confirmation time Costs: Transaction fees, gas costs, execution costs Network Performance Block Propagation: Time to reach network percentage Network Load: Memory pool size, peer connections, bandwidth usage Chain Quality: Uncle rate, chain reorganizations, block validity Resource Utilization Computing Resources: CPU usage, memory requirements, storage growth Network Resources: Bandwidth consumption, peer connections Storage Resources: State size, history size, pruning capabilities Node Architecture Node Types Full Nodes: Complete state and history validation Light Nodes: Header-only validation, SPV proofs Archive Nodes: Complete historical state storage Validator Nodes: Block production and validation Network Topology Peer Discovery: Bootstrap nodes, DHT, static peers Connection Management: Maximum peers, scoring systems, ban lists Data Propagation: Block propagation, transaction propagation Resource Management State Management: State storage, pruning policies, snapshot systems Network Management: Bandwidth limits, peer limits, prioritization Computing Management: Transaction pools, execution priorities Developer Infrastructure Development Tools SDKs: Language support, library ecosystems, tool chains Testing Tools: Local networks, test frameworks, simulation tools Monitoring Tools: Block explorers, metrics, logging systems API Infrastructure RPC Endpoints: JSON-RPC, GraphQL, WebSocket Node APIs: Client APIs, wallet interfaces, indexer APIs Data Access: Block data, state data, historical data Development Environment Local Development: Development networks, docker environments Testing Networks: Test networks, faucets, explorers Production Deployment: Network access, monitoring, maintenance

Common Concepts

State Growth & Pruning Mechanisms Across Networks Ethereum State Growth Unlimited state growth Historical state maintained by full nodes State size ~130GB and growing Uses Merkle Patricia Trie for state storage Pruning Mechanisms Archive nodes store full history Full nodes can prune historical states Light clients only store block headers No protocol-level state expiry yet (proposed) Solana State Growth Accounts model with rent mechanism State stored in memory for fast access Accounts can be marked rent-exempt Program data separated from account data Pruning Mechanisms Rent mechanism removes inactive accounts Two years of inactivity leads to account cleanup Validators can prune historical data Replicators store historical data Sui State Growth Object-centric storage model Objects can be deleted explicitly Storage fund for long-term costs Parallel execution reduces state bloat Pruning Mechanisms Objects can be explicitly deleted Storage rebates incentivize cleanup Validators can prune transaction history Maintains live object set separately Aptos State Growth Account-based model Resources stored under accounts Explicit resource deletion Storage fees for state growth Pruning Mechanisms Storage fees incentivize cleanup Explicit resource deletion Validators can prune historical states Maintains current state tree XRP State Growth Account-based ledger Reserve requirements limit state growth Objects can be deleted Amendable reserve requirements Pruning Mechanisms Online deletion for historical data Reserve requirements prevent spam Ledger header retention policy Configurable history retention Stellar State Growth Account-based ledger Minimum balance requirements Entry lifetime policies Base reserve adjustable by voting Pruning Mechanisms Built-in entry expiration Configurable history retention Minimum balance requirements Explicit deletion of unused entries Polkadot State Growth Trie-based state storage Storage rent (proposed) Bounded parachain growth Shared state across parachains Pruning Mechanisms State rent mechanism (planned) Historical pruning options Parachain state management Archive nodes for full history Layer 2 Solutions Optimistic Rollups (Optimism, Arbitrum) State growth tied to L1 data availability Compression techniques for state updates Fraud proof window affects state retention Can prune after challenge period ZK Rollups (zkSync, StarkNet) State updates verified by ZK proofs More efficient state representation Immediate finality enables pruning State diffs published to L1 Key Patterns Growth Control MechanismsStorage fees/rent Minimum balances Reserve requirements Explicit deletion Pruning StrategiesHistorical pruning State expiry Account cleanup Resource deletion Storage IncentivesRent mechanisms Storage rebates Deletion incentives Reserve requirements

13.3 Consensus Mechanisms

Consensus Mechanisms All DIN's have a consensus mechanism we have the consensus mechanisms. These can be divided into the following broad categories: Proof based mechanisms. Mainly Proof of Work and Proof of Stake Byzantine Fault Tolerance (BFT) Mechanisms. Tendermint Voting Based Mechanisms, Ripple and Stellar Directed Acrylic Graph based mechanisms Hybrid. PoS + BFT is used by Polkadot and Cosmos Novel Mechanisms

Need to talk about finality here about probabilistic vs deterministic. Also focus on Single Slot Finality Proof-Based Mechanisms Proof of Work (PoW) Security Model: Computational work as economic security Key Properties: Energy-intensive computational puzzles Natural chain selection through longest chain rule Strong resistance to Sybil attacks High latency to finality Finality: Probabilistic finality with increasing certainty over time Attack Resistance: 51% attack requires majority of network hashpower Very high cost of attack for established networks Examples: Bitcoin, Ethereum (pre-merge), Litecoin Proof of Stake (PoS) Security Model: Economic stake as security Key Properties: Energy efficient Validator selection based on stake Slashing conditions for misbehavior Lower latency than PoW Finality: Can achieve deterministic finality with additional mechanisms Attack Resistance: 33% or 51% stake required depending on implementation Economic penalties for malicious behavior Examples: Ethereum 2.0, Cardano, Tezos Byzantine Fault Tolerance (BFT) Mechanisms Classical BFT

Security Model: Agreement among known validator set Key Properties: High throughput Low latency Immediate finality Limited validator set size Finality: Instant finality once consensus reached Attack Resistance: Tolerates up to 1/3 Byzantine validators Requires 2/3+ honest validators Examples: Tendermint (Cosmos), Hyperledger Fabric Practical BFT (PBFT) Key Properties: Three-phase commit protocol View change protocol for leader failures Optimized for practical systems Performance Characteristics: Higher message complexity Better performance in stable conditions Examples: NEO, Hyperledger Sawtooth Voting-Based Mechanisms Federated Byzantine Agreement (FBA) Security Model: Trust between validator groups Key Properties: Quorum slices Flexible trust model Open validator set Finality: Quick finality within trust boundaries Attack Resistance: Based on trust overlap between validator groups Resistant to split votes through quorum intersection Examples: Stellar, Ripple Delegated Proof of Stake (DPoS) Security Model: Elected validator set Key Properties: Token holder voting Limited validator set Regular validator rotation Finality: Quick finality with selected validators Attack Resistance: Governance attacks through voter collusion Validator collusion risks Examples: EOS, TRON Directed Acyclic Graph (DAG) Based Pure DAG Security Model: Transaction confirmation through subsequent references Key Properties: Parallel transaction processing No explicit blocks Scalable throughput Finality: Probabilistic with increasing certainty Attack Resistance: Spam resistance through PoW Double-spend prevention through transaction ordering Examples: IOTA, Hedera Hashgraph Block-DAG Security Model: Hybrid of blockchain and DAG properties Key Properties: Multiple concurrent blocks Complex ordering rules Higher throughput than linear chains Examples: PHANTOM, SPECTRE Hybrid Mechanisms PoS + BFT Security Model: Economic stake with BFT finality Key Properties: Stake-based validator selection BFT consensus for finality Slashing for misbehavior Finality: Deterministic through BFT layer Attack Resistance: Economic security from stake Byzantine resistance from BFT Examples: Polkadot, Cosmos PoW + BFT Security Model: Dual layer security Key Properties: PoW for block production BFT for finality Higher security guarantees Examples: Decred Novel Mechanisms Proof of History (PoH) Security Model: Verifiable delay function for time ordering Key Properties: Built-in timestamping Efficient verification High throughput Finality: Quick finality with PoS integration Examples: Solana Avalanche Security Model: Metastable consensus through repeated sampling Key Properties: Sub-second finality High throughput Scalable validator set Finality: Probabilistic with quick practical finality Examples: Avalanche Key Considerations for Consensus Selection Performance Metrics Transaction throughput Time to finality Network overhead Hardware requirements Security Properties Byzantine fault tolerance Sybil resistance Double-spend prevention Network partition handling Decentralization Factors Validator set size Entry barriers Geographic distribution Economic concentration Network Requirements Synchronicity assumptions Bandwidth needs Storage requirements CPU/GPU demands Economic Considerations Operating costs Required stake/investment Reward distribution Slashing conditions

13.4 Application Models

Each application model represents a fundamental approach to how blockchains manage state, handle parallel execution, represent assets, enable composability, and provide safety guarantees.

We classify Application Models as:

- UTXO also includes Cardano's extened UTXO, Kaspa also falls into this
- Account
  - Pure EVM Ethereum, Polygon, Avalanche, TRON et al
  - Specialized XRP and Stellar
- Sharded account model Near falls into this.
- Object Sui
- Resource model Aptos and Movement
- Capability Solana
- Cell model TON
- Actor Based (ICP and CosmWasm

13.4.1 UTXO

The UTXO model treats the ledger as a set of unspent outputs that can be consumed as inputs to create new outputs. This model inherently supports parallel transaction validation since each UTXO can only be spent once.

Key Characteristics:

- State Management: Stateless, transaction-oriented
- Parallelization: Natural parallel validation
- Asset Representation: Native representation of tokens
- Composability: Limited without extensions
- Safety Guarantees: High through explicit ownership

13.4.2 Account Model

Accounts maintain state directly, with each account having properties like balance and nonce. This model enables rich programmability but can face challenges with parallelization. However if we look at EOA's, they only maintain state of the network token and not ERC20's.

Key Characteristics:

- State Management: Stateful, account-oriented

- Parallelization: Challenging due to state dependencies
- Asset Representation: Contract-based tokens
- Composability: High through contract interactions
- Safety Guarantees: Varies by implementation

13.4.3 Object Model

Objects are owned, independent state elements that can be transferred and modified. This enables parallel execution while maintaining rich programmability.

Key Characteristics:

- State Management: Object-oriented
- Parallelization: Natural through object independence
- Asset Representation: Native objects
- Composability: Through object references
- Safety Guarantees: Object-level ownership

13.4.4 Resource Model

Resources are linear types that cannot be copied or discarded, only moved between accounts. This provides strong safety guarantees for digital assets.

Key Characteristics:

- State Management: Resource-oriented
- Parallelization: Possible through resource independence
- Asset Representation: Native resources
- Composability: Through resource combination
- Safety Guarantees: Very high through linear types

13.4.5 Capability Model

Capability Model Access to resources is controlled through explicit capabilities, enabling fine-grained access control and parallel execution.

Key Characteristics:

- State Management: Capability-based
- Parallelization: High through capability isolation
- Asset Representation: Capability-protected resources
- Composability: Through capability delegation
- Safety Guarantees: High through access control

13.4.6 Cell Model

State is organized into cells that can be independently accessed and modified, enabling high parallelization.

Key Characteristics:

- State Management: Cell-based
- Parallelization: High through cell independence
- Asset Representation: Cell-based
- Composability: Through cell references
- Safety Guarantees: Cell-level isolation

13.4.7 Actor Model

Computation is organized around actors that can independently process messages, enabling natural parallelization.

Key Characteristics:

- State Management: Actor-local state
- Parallelization: Natural through actor independence
- Asset Representation: Actor-managed
- Composability: Through message passing
- Safety Guarantees: Actor isolation

Application Model	Blockchains	Key Features
UTXO	Bitcoin, Cardano (eUTXO), Kaspa, Ergo, Bitcoin Cash, Litecoin, Zcash, Dogecoin, Monero, BEAM	- Natural parallelism- Simple state model- High security
Account (Pure EVM)	Ethereum, Polygon, Avalanche C-Chain, BSC, TRON, Fantom, Arbitrum, Optimism, Mantle, WorldCoin, Gala, Flare	- Rich programmability- High composability- Standard tooling
Account (Specialized)	XRP, Stellar, Algorand, Stacks, Tezos, Hedera, VeChain, Quant, the Sandbox, StarkNet, Neo, BitTorrent	- Custom account models- Specific use-case optimization- Modified state management

Application Model	Blockchains	Key Features
Sharded Account	NEAR, Polkadot, Kusama, Elrond (MultiversX), Harmony	- Parallel execution- Cross-shard composition- Scalable state
Object	Sui	- Object-centric- Natural parallelism- Rich ownership model
Resource	Aptos, Movement, Flow, Virtual Protocol	- Linear types- Strong safety- Asset-oriented
Capability	Solana, Injective, SEI, Theta Network	- Fine-grained access- High parallelism- Explicit permissions
Cell	TON, Everscale	- Cell-based storage- High parallelism- Flexible structure
Actor	Internet Computer (ICP), CosmWasm chains (Osmosis, Celestia, Cosmos Hub), Fetch.ai, IOTA	- Message-passing- Natural isolation- Independent processing
Unknown	Filecoin, Arweave, Bittensor, Mantra, Ondo, Kaia, Brett, Jasmycoin	Still researching these

13.5 Communication Protocols

We have

- IBC (Inter-Blockchain Communication) - Cosmos
- XCMP (Cross-Chain Message Passing) - Polkadot
- CCTP (Cross-Chain Transfer Protocol) by Circl
- CCIP (Cross-Chain Interoperability) by Chainlink
- Warp Messaging for Avalanche
- IMP (Interchain Messaging Protocol)

We can then further classify these standard by the following properties: * Security Model * Centralized vs Decentralized * Message Scope (General (IBC,CCIP, Hyperlane) vs Specialized (Circle's CCTP and Warp)) * Open vs Close networks * Message Verification Method

13.6 Fee Markets

We break down fee markets into:

- Block Space Markets
- Resources Markets

13.6.1 Block Space Markets

These are the most fundamental type. Users are paying for space in the next block, regardless of what they're doing with that space. It only considers the price of inclusion while respecting the limits of the network.

13.6.2 Resource Markets

These evolved with the advent of Ethereum as blockchains became more complex. We break these down into:

- Compute
- Data Availability
- Storage
- Hybrid

13.6.2.1 Compute Markets

Transaction fee's sometimes called Gas is what you pay to execute a smart contract on a blockchain. On some networks, like Ethereum, there is an execution limit on how much computational work can be done in a block. This parameter directly impacts the computational resources required to run a node.

Nodes can propose a gas limit

So the attributes for Transactions Fee's are: Tips/Bribes How Transactoin fees are distributed. Burnt or redirected to Miners How fee's are calculated Base Fee Dynamic Fee (based on something like network congestion)

13.6.3 Data Availability

We will compare the mechanisms of the following major DA layers:

- Ethereum
- Celestia
- Eignlayer DA
- Avail

Ethereum uses dynamic blob fees based upon a target amount of blob space per block.

Celestia charges by size. Celestia is designed to scale to handle increase demand with adding more validators

Avail also charges by size but also the type.

EigenDA creates a marketplace for data availability providers

13.6.3.1 Storage

13.6.3.2 Hybrid

13.6.3.3 Congestion Measurement Mechanisms

Ethereum

- Measures block gas utilization vs 15M target
- 12.5% base fee adjustment per block
- Block size can flex up to 30M gas

Stellar

- Uses surge pricing mechanism
- Triggers when ledger capacity exceeds 50%
- Fee increases proportionally to network load

Polkadot

- Uses weight-based system
- Measures block weight against target
- Adjusts fees based on block fullness

Tezos

- Monitors block saturation
- Adjusts fees based on recent block usage
- Uses gas limits per operation type

13.6.4 Data Availability market

Ethereum uses blobs, which are special containers for Layer 2 rollups to post their transaction data. They also disappear after a time.

Blob fee is calculated based on the target blob gas per block.

13.7 Network Change Management

13.7.1 Overview

Network change management in DINs encompasses the processes, mechanisms, and stakeholders involved in proposing, discussing, approving, and implementing network modifications. This framework covers everything from minor parameter adjustments to major protocol upgrades.

TODO Design something that can simulate

13.8 Clients

13.9 Bridges

Classification of Bridges. Need to include Lazy Bridges here https://blog.celestia.org/lazybridging/

- Trusted
  - Custodial
  - Multi Signature
  - Federated
- Trustless Bridges
  - Light Client
  - ZK
    - Lazy Bridge
  - Relay

14 dApps

dApps can be single chain or multichain. However they must exist on a DIN and can't exist independently from a DIN. They don't need to have a token to be included here. If an NFT develops functionality, they they'll be inlcuded here too. They can be DeFi SocialFi GameFi CeDeFi Concepts Concepts Flash loans DeFi Derivatives Basis Trading DEX Lending Utility Yield Pendle 3k in Basis Trading Current dApps Ethena USDX BounceBit

Derivatives Liquid staking Derivatives such as Lido, and EtherFi is the primary element here Lending

SocialFi Meme Coins CeDeFi Circle fits in here GameFi

14.1 DEX's

14.2 Name Services

15 MEV

15.1 Proposer Builder Separation (PBS)

15.2 Multiple Concurrent Leaders

https://x.com/aeyakovenko/status/1810222589991583922

Part V

Social

16 Governance

This section will compare how Blockchains implement the three tiers of traditional governance. Namely:

- Legistlative - Who makes the rules and how they created
- Executive - Who executes the rules
- Judicial - Referee between them

So in Bitcoin BIP's cover the Legislative aspect, Executive is Node operators for Networks then for dapps it gets more complex. Judicial is where it gets challenging. This is pretty much the entire community. People interpret rules by economic activity and nodes. Look at Bitcoin Cash. No money. So everyone agreed with the block size of Bitcoin.

There is also the concept of Canvassing or Lobbying can also occur. Let's look at Ethereum. If I wanted to increase the gas limit from 30 million to 31 million. I'd need to canvas all the nodes to come along with me. Currently there is 5,333 nodes. So I'd need to convince ~3k node operators to increase the gas limit.

16.0.0.1 1. Hard Forks

- **Definition**: Protocol changes that make previously invalid blocks/transactions valid (or vice-versa), requiring all nodes to upgrade
- **Characteristics**:
  - Non-backwards compatible
  - Requires coordinated network upgrade
  - Creates potential for chain splits if not unanimously adopted
- **Use Cases**: Major protocol upgrades, fundamental rule changes, bug fixes
- **Examples**: Ethereum's merge to PoS, Bitcoin's SegWit upgrade

16.0.0.2 2. Soft Forks

- **Definition**: Backwards-compatible protocol changes that tighten rules without invalidating existing blocks
- **Characteristics**:

  - Backwards compatible
  - Old nodes can still participate (with limitations)
  - Lower coordination requirements
- **Use Cases**: Adding new features, incremental improvements
- **Examples**: Bitcoin's P2SH implementation, taproot upgrade

16.0.0.3 3. Parameter Updates

- **Definition**: Changes to network variables within predefined bounds
- **Characteristics**:
  - No code changes required
  - Often automated through on-chain governance
  - Lower risk than protocol changes
- **Use Cases**: Fee adjustments, block size modifications, staking parameters
- **Examples**: Tezos' regular parameter updates, Cosmos' governance parameters

16.0.1 Governance Mechanisms

16.0.1.1 1. Off-Chain Governance

- **Characteristics**:
  - Social consensus through discussion forums, social media, conferences
  - Informal decision-making processes
  - Relies on node operator coordination
- **Advantages**:
  - Flexible and adaptable
  - Allows for nuanced discussion
  - Natural resistance to capture
- **Disadvantages**:
  - Can be slow and messy
  - May lack clear resolution mechanisms
  - Potential for contentious outcomes

16.0.1.2 2. On-Chain Governance

- **Characteristics**:
  - Formal voting mechanisms
  - Smart contract-based execution
  - Token-weighted or identity-based participation
- **Advantages**:
  - Clear process and outcomes
  - Automated execution
  - Transparent participation
- **Disadvantages**:
  - Potential plutocratic capture
  - Reduced flexibility
  - Voter apathy risks

16.0.1.3 3. Hybrid Systems

- **Characteristics**:
  - Combines off-chain discussion with on-chain execution
  - Multiple stages of proposal refinement
  - Mixed participation models
- **Advantages**:
  - Balances flexibility with formality
  - Combines benefits of both approaches
  - Can adapt to different types of changes
- **Examples**: Polkadot's governance system, Cosmos Hub's proposal process

16.0.2 Improvement Proposal Systems

16.0.2.1 1. Structure

- **Stages**:
  - Draft: Initial proposal development
  - Review: Community feedback and refinement
  - Last Call: Final period for major objections
  - Accepted/Final: Ready for implementation

    - Rejected: Proposal declined

- **Components**:
    - Technical specification
    - Motivation and rationale
    - Backwards compatibility analysis
    - Reference implementation (if applicable)
    - Security considerations

16.0.2.2 2. Common Frameworks

- **BIP (Bitcoin Improvement Proposals)**:
    - Focus on consensus changes
    - Conservative approach
    - High emphasis on security
- **EIP (Ethereum Improvement Proposals)**:
    - Multiple tracks (Core, ERC, Interface)
    - Regular cadence of updates
    - Strong emphasis on standardization
- **Network-Specific Systems**:
    - Customized to network needs
    - Varying levels of formality
    - Different voting thresholds

16.0.3 Centralization Factors

16.0.3.1 1. Development Centralization

- **Core Development Teams**:
    - Control over codebase
    - Technical expertise concentration
    - Funding dependencies
- **Client Implementation**:
    - Diversity of node software
    - Implementation independence
    - Bug discovery and fixes

16.0.3.2 2. Governance Centralization

- **Voting Power Distribution**:
  - Token concentration
  - Delegate systems
  - Voter participation rates
- **Proposal Control**:
  - Who can propose changes
  - Filtering mechanisms
  - Discussion venue control

16.0.3.3 3. Infrastructure Centralization

- **Node Operation**:
  - Geographic distribution
  - Hardware requirements
  - Operating costs
- **Service Providers**:
  - API services
  - Block explorers
  - Development tools

16.0.4 Best Practices

16.0.4.1 1. Change Management

- Clear documentation of changes
- Adequate testing periods
- Coordinated upgrade schedules
- Emergency response procedures

16.0.4.2 2. Community Engagement

- Regular communication channels
- Multiple feedback mechanisms
- Transparent decision-making
- Educational resources

16.0.4.3 3. Technical Implementation

- Comprehensive testing frameworks
- Clear upgrade paths
- Fallback mechanisms
- Security audits

17 Contributing

This Almanack will change often and get things wrong. It's only by being intellectually honest that it can ever hope to be the canonical guide to crypto and Web3. We follow Sophocles as our North Star

> "All men make mistakes, but a good man yields when he knows his course is wrong, and repairs the evil. The only crime is pride." Here we'll list all the outstanding contributions we're looking for.

I'll also use it as a dumping ground where I can keep track of things I need to read to include:

- https://bitcoinrollups.org/
- https://tr3y.io/articles/crypto/bitcoin-zk-rollups.html

17.1 Current requirements

17.1.1 Translations

Be good to focus on the most widely spoken languages first. So

- Mandarin
- Hindi
- Spanish
- French
- Arabic
- Bengali
- Russian
- Portuguese
- Indonesian

References

Buterin, Vitalik. 2013. "Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform." https://ethereum.org/en/whitepaper/.

———. 2021. "An Incomplete Guide to Rollups." https://vitalik.ca/general/2021/01/05/rollup.html.

Charbonneau, Jon. 2023. "The Complete Guide to Rollups." Research Report. Delphi Digital. https://members.delphidigital.io/reports/the-complete-guide-to-rollups.

Gluchowski, Alex. 2019. "zkRollup Vs. Optimistic Rollup: Technical Comparison." Matter Labs. https://medium.com/matter-labs/zkrollup-vs-optimistic-rollup-deep-dive-ea141e71e075.

Hughes, Eric. 1993. "A Cypherpunk's Manifesto," March. https://www.activism.net/cypherpunk/manifesto.html.

May, Timothy C. 1988. "The Crypto Anarchist Manifesto." https://www.activism.net/cypherpunk/crypto-anarchy.html.

Monegro, Joel. 2016. "Fat Protocols." Union Square Ventures. https://www.usv.com/writing/2016/08/fat-protocols/.

Nakamoto, Satoshi. 2008. "Bitcoin: A Peer-to-Peer Electronic Cash System." https://bitcoin.org/bitcoin.pdf.

Poon, Joseph, and Vitalik Buterin. 2017. "Plasma: Scalable Autonomous Smart Contracts." https://plasma.io/plasma-whitepaper.pdf.

Poon, Joseph, and Thaddeus Dryja. 2016. "The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments." https://lightning.network/lightning-network-paper.pdf.

Raymond, Eric S. 1999. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. Sebastopol, CA: O'Reilly Media.

Wood, Gavin. 2014. "Ethereum: A Secure Decentralised Generalised Transaction Ledger." Ethereum Foundation. https://ethereum.github.io/yellowpaper/paper.pdf.