Trollip's Degen Almanack

Justin Trollip

2024-12-09

Table of contents

Welcome

This Almanack represents an attempt to document cryptocurrency and Web3 knowledge without the conflicts of interest that plague our industry. Just as Henry Varnum Poor created his Manual of Railroads to bring transparency to America's railroad boom, this Almanack aims to bring clarity to the cryptocurrency revolution.

Why This Almanack Exists

The cryptocurrency industry suffers from a unique problem: those with the deepest knowledge often have the strongest financial incentives to mislead others. Protocol developers promote their own chains, influencers pump their own holdings, and "researchers" serve those who pay them. Meanwhile, crucial information remains locked behind paywalls and exclusive groups.

This Almanack breaks that pattern by providing:

- Comprehensive knowledge from basic concepts to advanced strategies
- Independent analysis free from token-holder influence
- Technical depth that doesn't sacrifice accessibility
- Practical guidance for both users and developers

How to Use This Almanack

Whether you're seeking financial sovereignty or building the future, this Almanack offers multiple paths:

For Those Seeking Independence
Start with "Regular Person's Path to Independence" to understand how to safely participate in the cryptocurrency ecosystem without falling prey to scams or losing your funds.

For Technical Minds
The "Technical" section provides deep dives into cryptography, blockchain architecture, and protocol design patterns. Consider starting with "Web3 Essentials" to establish a common vocabulary.

For Market Participants
The "Financial" section offers frameworks for analyzing digital assets, understanding market structures, and evaluating protocols.

Living Document

This Almanack is version controlled through Git and continuously updated by community contributions. Every technical claim is justified, every strategy explained, and every risk clearly stated. You're reading an early draft, dated December 2024.

Contributing

Knowledge critical to digital independence should be freely accessible. This work is licensed under Creative Commons Attribution-ShareAlike 4.0, ensuring it remains open while preventing commercial exploitation. Learn more in our "Contributing" section.

Begin your journey with the topics listed in the navigation menu, or proceed systematically through each section. Welcome to your guide to digital independence.

Part I

Introduction

Copyright Notice

Motivation

Knowledge critical to achieving financial sovereignty should never be locked behind paywalls or restrictions. The crypto industry already suffers from enough artificial barriers - gated Discord servers, exclusive research groups, and insider knowledge networks that perpetuate inequality.

This Almanack exists to break down these barriers. Just as Bitcoin enables permissionless financial transactions, this work aims to enable permissionless learning. However, we must also prevent others from creating new barriers around this knowledge.

This is why we've chosen the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). This copyleft license ensures that:

1. Anyone can freely access, share, and build upon this work
2. Any derivative works must maintain the same freedoms
3. Commercial use is permitted, but cannot restrict access
4. Attribution protects the community's contributions
5. The viral nature of ShareAlike prevents future enclosure

Think of it like a smart contract for knowledge: immutable rules that protect freedom while enabling innovation.

Legal Text

You are free to:

- Share — copy and redistribute the material in any medium or format
- Adapt — remix, transform, and build upon the material for any purpose, even commercially

Under the following terms:

- Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

Version Control

This work is version controlled through Git. Each version represents a distinct iteration of the Almanack, identified by commit hashes and release tags. While the content may evolve, this license remains constant across all versions.

You can find the complete version history and contribute to future versions at: https://github.com/HariSeldon23/almanack

Attribution Example

When sharing or building upon this work, please provide attribution in the following format:

"This work is based on Trollip's Degen Almanack © 2024 Justin Trollip, used under CC BY-SA 4.0. [Your modifications, if any]"

Contact

For questions about usage or licensing, contact: jtrollip@protonmail.com

---

The above represents the formal terms under which this Almanack is shared. In the spirit of transparency and accessibility that cryptocurrency enables, we've chosen these terms to ensure this knowledge remains as free as the protocols it describes.

Dedication

For my two skebangas. Huxley & Morrissey

Epigraph

"Arise, you have nothing to lose but your barbed wire fences!"
— Timothy C. May (1988)

Preface

The cryptocurrency revolution has created unprecedented opportunities for financial independence, but it has also spawned an industry rife with exploitation. As someone deeply embedded in this world, I face a moral dilemma: I can either participate in a system that profits from obscuring knowledge, or I can work to make that knowledge freely accessible to all.

This Almanack represents my choice of the latter path. Just as the cypherpunks believed privacy and freedom of information could coexist, I believe we can build a more equitable crypto ecosystem without compromising its innovative potential. The fundamental act of friendship among cryptocurrency enthusiasts – what I call "degens" – should be the sharing of knowledge, not the hoarding of it.

The technical complexity of cryptocurrency has created a particularly insidious form of gatekeeping. Essential market data and analysis hide behind expensive paywalls. Crucial insights remain locked in exclusive chat groups and private networks. This artificial scarcity of knowledge directly contradicts the core ethos of Web3: permissionless access and decentralized power.

This Almanack aims to break down these barriers. It will serve as a comprehensive, freely accessible resource covering everything from basic concepts to advanced trading strategies. More importantly, it will evolve through community contributions, ensuring it stays relevant and accurate as the industry develops.

Some might question why I would freely share knowledge that others sell at a premium. My answer is simple: the long-term health of our ecosystem depends on educated participants making informed decisions. When knowledge is concentrated in the hands of a few, it creates the very centralization of power that cryptocurrency was meant to disrupt.

Looking ahead, I envision this Almanack becoming a living document that adapts to the rapid changes in our industry. I'm exploring sustainable open-source funding models, potentially through platforms like Mirror that allow for anonymous contributions. This would ensure the Almanack can continue growing while maintaining its independence from traditional financial incentives.

This is just the beginning. The path to truly democratized crypto knowledge will require continuous effort and collaboration. If you share this vision, I invite you to contribute your expertise, suggest improvements, or simply help spread the word. Together, we can build something that honors the original promise of cryptocurrency: financial sovereignty for all.

The work begins now.

1 Introduction

The crypto industry suffers from a unique paradox: those with the deepest knowledge often have the strongest incentives to obscure rather than illuminate. This troubling reality became clear to me during a technical debate I witnessed on Twitter. Both participants made valid points, yet instead of building understanding, they resorted to personal attacks. What struck me wasn't just the hostility, but the absence of a shared framework for discussion – a common language that could bridge their perspectives.

This observation rekindled an idea I'd been considering for years: creating a comprehensive blockchain taxonomy. But as I began mapping out the technical architecture of various networks, I realized the scope needed to be much broader. The same knowledge gaps that hindered technical discussions were even more pronounced in everyday conversations about cryptocurrency investment and usage.

I thought about the countless conversations I've had with friends and family seeking crypto advice. These weren't developers or traders – they were curious individuals trying to understand a complex new technology. Despite years of experience in the industry, I struggled to give them guidance that was both accessible and comprehensive. The existing resources either oversimplified to the point of uselessness or drowned readers in technical jargon.

This Almanack aims to bridge that gap. Drawing inspiration from Henry Varnum Poor's Manual of Railroads, which brought transparency to America's railway boom, we're creating a resource that serves both technical and non-technical audiences. Just as Poor's manual helped investors understand the revolutionary technology of his time, this Almanack aims to demystify the crypto revolution for everyone.

The name "Trollip's Degen Almanack" might seem unusual. Initially, I considered calling it a blockchain taxonomy or a Web3 guide. But these terms felt too limiting. The word "degen" – short for degenerate – carries special meaning in crypto culture. While often used ironically to describe risk-taking traders, being a degen also implies deep engagement with the technology and markets. It's through this engaged experimentation that true understanding emerges.

What sets this Almanack apart is its commitment to intellectual honesty and practical utility. We're building it as an open-source, community-reviewed resource that will evolve alongside the technology it describes. The documentation uses version control through Git, allowing readers to track how understanding changes over time. As the industry matures, we plan to integrate real-time data sources, transforming this from a static document into a dynamic tool for decision-making.

The Almanack is organized into distinct paths catering to different needs:

- The Path to Independence guides those seeking financial sovereignty
- The Technologist's Path provides deep technical understanding
- The Financial sections offer frameworks for analysis and investment

Each section builds upon the others, creating a comprehensive resource that grows with your understanding. Whether you're a developer looking to understand economic implications, an investor studying technical fundamentals, or someone simply seeking financial independence, you'll find your path forward here.

This is an ambitious undertaking, and like any first edition, it will contain errors and omissions. But by maintaining rigorous standards for evidence, encouraging community contributions, and staying true to the open-source ethos, we aim to create something valuable: a trusted guide through the often confusing world of cryptocurrency and Web3.

The journey begins here. Welcome to your guide to digital independence.

2 How to Use This Almanack

This almanack serves multiple audiences with different needs and backgrounds. Whether you're seeking financial independence, building decentralized systems, or analyzing crypto markets, this guide will help you navigate the content effectively.

2.1 Understanding the Structure

The almanack is organized into progressive sections that build upon each other while remaining independently accessible. Think of it like a city with different districts - you can start in any area that interests you, but the main roads connect everything in a logical way.

2.1.1 For Those Seeking Financial Independence

If your primary goal is achieving financial sovereignty through cryptocurrency, begin with:

1. Start with the "Foundations" section to understand basic tools and security
2. Move to "Path to Independence" which provides step-by-step guidance
3. Reference the "Financial Systems" section as you advance
4. Explore other sections based on your growing interests and needs

2.1.2 For Technical Minds

If you're approaching from a technical perspective, particularly as a builder or developer:

1. Begin with "Technical Architecture" to understand fundamental concepts
2. Explore "Applications & Use Cases" for implementation patterns
3. Study "Financial Systems" to understand the economic aspects
4. Reference "Governance & Social Coordination" for broader ecosystem context

2.1.3 For Market Participants

If you're focused on trading, investing, or market analysis:

1. Start with "Financial Systems" for core concepts
2. Explore "Technical Architecture" to understand underlying technology
3. Study "Governance & Social Coordination" for market-moving factors
4. Reference other sections as needed for comprehensive understanding

2.2 Using the Rating Systems

This almanack includes several rating frameworks to help you evaluate:

- Digital assets and protocols
- Security considerations
- Market opportunities
- Technical architectures

When using these ratings:

- Consider them starting points rather than absolute truth
- Look at the underlying metrics and methodology
- Understand how different factors are weighted
- Apply the frameworks to your own analysis

2.3 Working with Technical Content

Technical sections include:

- Code examples
- Architecture diagrams
- Mathematical formulas
- Protocol specifications

These are designed to be both rigorous and accessible:

- Begin with conceptual overviews
- Study detailed explanations
- Experiment with provided examples
- Reference external resources when needed

2.4 Understanding Risk

Throughout the almanack, risk is treated as a fundamental concept:

- Security risks are covered in technical sections
- Market risks are detailed in financial sections
- Social risks are explored in governance sections
- Operational risks are discussed in implementation guides

Always consider risk factors before implementing any strategy or system described here.

2.5 Contributing and Staying Updated

This is a living document that improves through community contribution:

- Check version numbers and update dates
- Review the contribution guidelines
- Submit improvements or corrections
- Join related discussions

2.6 Making the Most of Examples

The almanack includes numerous examples:

- Case studies of successful and failed projects
- Code implementations
- Market analyses
- Governance scenarios

Use these to:

- Understand theoretical concepts in practice
- Learn from others' experiences
- Identify patterns and anti-patterns
- Develop your own analytical frameworks

2.7 Following the Learning Paths

While sections can be read independently, certain learning paths are recommended:

2.7.1 Beginner Path

1. Basic terminology and concepts
2. Setting up essential tools
3. Security fundamentals
4. First interactions with crypto

2.7.2 Intermediate Path

1. Understanding market dynamics
2. Technical architecture basics
3. Risk management principles
4. Advanced tools and strategies

2.7.3 Advanced Path

1. Complex technical concepts
2. Advanced market analysis
3. Protocol design patterns
4. Governance mechanisms

2.8 Getting Help

If you encounter difficulties:

- Review prerequisite sections
- Check the glossary for unfamiliar terms
- Reference external resources
- Engage with the community
- Submit questions through appropriate channels

Remember that this almanack is designed to grow with you. As your understanding deepens, previously complex sections will become more accessible, and new layers of insight will emerge from familiar content.

3 Web3 Terminology

3.1 Introduction

Understanding Web3 requires familiarity with a unique vocabulary that spans multiple disciplines: cryptography, economics, computer science, and social coordination. This guide organizes these terms into logical categories and provides clear explanations with relevant examples.

3.2 Evolution of the Web

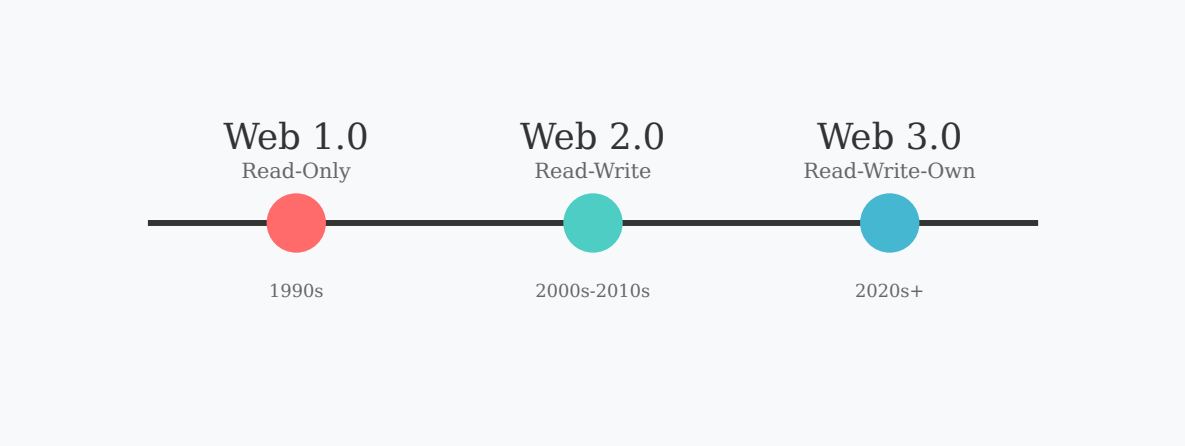

Figure 3.1: Evolution of the Web

3.2.1 Web 1.0 (1990-2004)

The first iteration of the worldwide web consisted primarily of static websites that users could only read. Information flowed in one direction - from website owners to visitors. Think of early news websites or company homepages that rarely changed and offered no interaction.

3.2.2 Web 2.0 (2004-2020)

The social web emerged, characterized by user-generated content, social networks, and interactive platforms. Users could both read and write content, but platforms owned and controlled the data. Facebook, Twitter, and YouTube exemplify Web 2.0 platforms where users create content but don't truly own or control it.

3.2.3 Web3 (2020-Present)

The ownership web represents a fundamental shift where users can read, write, and own their digital assets and data. Instead of trusting platforms to manage our digital lives, Web3 uses cryptographic protocols and economic incentives to enable direct ownership and peer-to-peer interactions.

3.3 Core Concepts

3.3.1 Decentralization

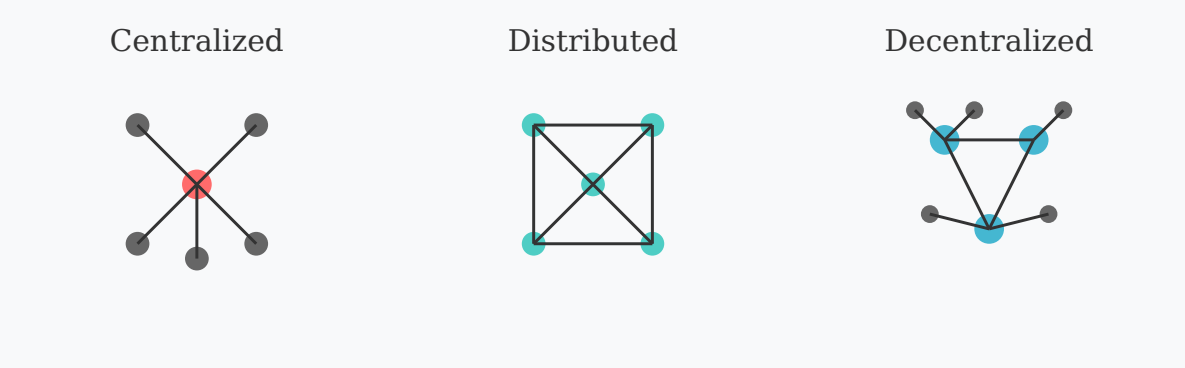

Figure 3.2: Decentralization

Decentralization refers to the distribution of power, control, and decision-making across a network rather than concentration in a single entity. It exists on a spectrum:

1. **Architectural Decentralization**: How many physical computers comprise the system?
2. **Political Decentralization**: How many individuals or organizations control those computers?
3. **Logical Decentralization**: Does the interface and data structures appear more like a single monolithic object, or an amorphous swarm?

Examples help illustrate these distinctions:

- Bitcoin is architecturally and politically decentralized but logically centralized (one shared ledger)
- Email is architecturally decentralized but politically centralized (few major providers) and logically centralized (standardized protocol)
- Language is decentralized across all three dimensions

3.3.2 Blockchain

A blockchain is a distributed database that maintains a continuously growing list of records (blocks) that are cryptographically linked to previous records. Key characteristics include:

1. **Immutability**: Once data is recorded, it cannot be altered without changing all subsequent blocks
2. **Transparency**: All transactions are public and verifiable
3. **Consensus**: Network participants agree on the state of the system without trusting each other

The term "blockchain" has become somewhat limiting - many modern systems use different data structures while maintaining similar properties. This is why some prefer broader terms like "distributed ledger technology" or "decentralized incentive networks."

3.4 Account Types

3.4.1 Simple Accounts

Simple accounts represent the most basic way to interact with blockchain networks. They have:

- A public key (like an email address)
- A private key (like a password)
- The ability to hold and transfer native network tokens
- No additional programmable logic

3.4.2 Smart Accounts

Smart accounts extend simple accounts with programmable functionality:

- Custom validation logic
- Multi-signature requirements

- Automated actions
- Integration with smart contracts

For example, a smart account might require two out of three designated signatures to approve transactions or automatically split incoming payments between multiple recipients.

3.5 Digital Assets

3.5.1 Native Coins

Native coins (sometimes called protocol tokens) are the primary digital assets of blockchain networks. They serve several purposes:

- Pay for transaction fees (gas)
- Secure the network through staking or mining
- Participate in governance
- Transfer value

Examples include:

- Bitcoin (BTC) on the Bitcoin network
- Ether (ETH) on Ethereum
- SOL on Solana

3.5.2 Tokens

Tokens are digital assets created on top of blockchain platforms. They differ from native coins because they don't secure the underlying network. Major categories include:

3.5.2.1 Fungible Tokens

Interchangeable tokens where each unit is identical to every other unit. Think of them like traditional currency - any dollar bill can be exchanged for any other dollar bill. Categories include:

- Stablecoins (USDC, DAI)
- Governance tokens (UNI, AAVE)
- Security tokens
- Utility tokens

3.5.2.2 Non-Fungible Tokens (NFTs)

Unique digital assets where each token has distinct properties. Common uses include:

- Digital art and collectibles
- Gaming items
- Domain names
- Membership passes
- Real estate titles

3.6 Financial Concepts

3.6.1 Decentralized Finance (DeFi)

Financial services built on blockchain networks that operate without traditional intermediaries. Key components include:

3.6.1.1 Automated Market Makers (AMMs)

Smart contracts that create liquidity pools allowing users to trade tokens without traditional order books. Instead of matching buyers with sellers, users trade against a pool of tokens with prices determined by mathematical formulas.

3.6.1.2 Yield Farming

The practice of providing liquidity or assets to DeFi protocols in exchange for rewards, typically in the form of governance tokens or trading fees.

3.6.1.3 Impermanent Loss

A unique risk in liquidity provision where the value of assets deposited in an AMM pool can decrease relative to simply holding those assets, due to price movements and the AMM's constant product formula.

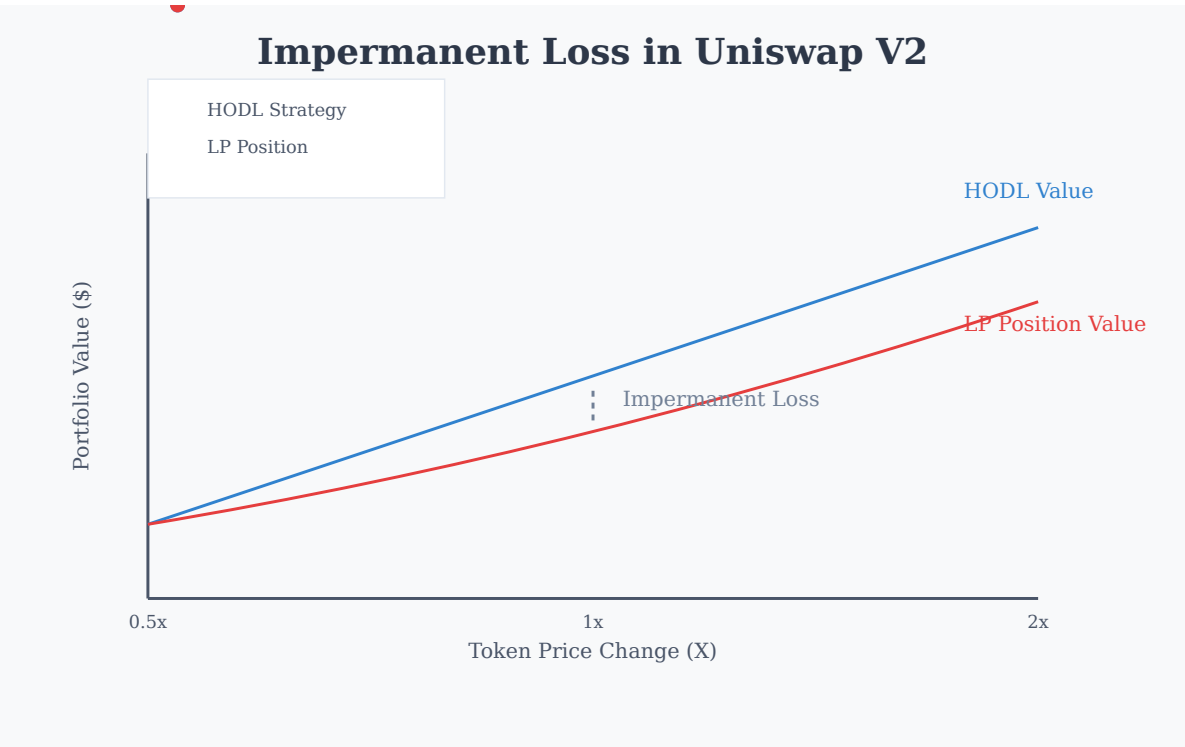

Figure 3.3: Impermanent Loss

3.7 Cultural Terms

3.7.1 HODL

Originally a misspelling of "hold" that became crypto slang for maintaining long-term positions regardless of market conditions. The term evolved to mean "Hold On for Dear Life" and represents a conviction-based investment strategy.

3.7.2 Degen

Short for "degenerate," this term began as criticism of high-risk trading behavior but has been reclaimed by the community to describe sophisticated traders who:

- Take calculated risks
- Deeply understand protocol mechanics
- Stay ahead of market trends
- Actively participate in new protocols

3.7.3 Gas

Transaction fees paid to network validators, typically priced in the network's native token. Gas prices fluctuate based on network demand, with higher prices during periods of congestion.

3.7.4 Gwei

A denomination of ETH, specifically 10^-9 ETH. Commonly used to express gas prices on Ethereum and EVM-compatible networks.

3.8 Security Concepts

3.8.1 Seed Phrase

A sequence of 12-24 words that serves as a backup for private keys. Also called a mnemonic phrase or recovery phrase. The words are selected from a standardized list of 2048 words and must be stored securely, as anyone with access to the seed phrase can control the associated accounts.

3.8.2 Smart Contract

Self-executing programs stored on a blockchain that automatically enforce and execute agreements between parties. Key characteristics:

- Immutable once deployed
- Transparent and verifiable
- Execute exactly as programmed
- No downtime or censorship

Part II

Foundations

4 Digital Independence

The story of digital independence begins not with blockchains or cryptocurrencies, but with a profound recognition: the tools that brought unprecedented convenience to our lives have also created unprecedented control over them.

Consider your daily financial life. Your morning coffee purchase creates a data point. Your salary arrives through systems you don't control. Your savings exist primarily as numbers in someone else's database. This convenience comes with hidden costs - your transactions can be blocked, your accounts frozen, your privacy compromised. Each small sacrifice of control seemed reasonable in isolation, but together they've created golden handcuffs of financial dependence.

This isn't accidental. The post-industrial financial system runs on centralization because it's efficient. Banks can process thousands of transactions per second. Credit cards work seamlessly across borders. Mobile payments happen with a fingerprint. But this efficiency masks a fundamental truth - you're asking permission to use your own money.

The early cypherpunks understood this tradeoff. In 1993, Eric Hughes wrote in the Cypherpunk Manifesto:

"Privacy is necessary for an open society in the electronic age. Privacy is not secrecy. A private matter is something one doesn't want the whole world to know, but a secret matter is something one doesn't want anybody to know. Privacy is the power to selectively reveal oneself to the world."

They saw that digital privacy would become inseparable from freedom. Without privacy in our transactions and communications, true independence would be impossible. But they also understood that simply criticizing the system wasn't enough - they needed to build alternatives.

This brings us to public key cryptography, the foundation of digital independence. Imagine having a special lock that anyone can use to send you messages or money, but only you can open. No permission needed, no middlemen required, no central authority to approve or deny. This isn't just technical theory - it's a practical tool for independence.

Bitcoin emerged from this foundation, but it would be a mistake to see it as just digital money. It proved that we could create systems where trust comes from mathematics and consensus rather than institutions. Where rules are enforced by code rather than policy. Where participation is permissionless rather than granted.

The path to digital independence isn't about rejecting modern convenience - it's about reclaiming control while preserving it. We'll learn to:

- Hold assets that can't be frozen or seized
- Communicate without surveillance
- Trade without gatekeepers
- Build systems that resist control

But this power comes with responsibility. In traditional systems, mistakes can often be reversed. Passwords can be reset. Transactions can be disputed. In truly independent systems, you alone are responsible for your security. Your privacy. Your choices.

This Almanack exists because that responsibility requires knowledge. Not just technical knowledge, though that's important, but practical wisdom. Understanding not just how these systems work, but why they matter. Learning not just to use tools, but to think independently about digital freedom.

The journey of digital independence is both personal and collective. Each person who takes control of their digital life strengthens the network for everyone. Each developer who builds privacy-preserving tools expands what's possible. We're not just users of a new system - we're participants in its evolution.

In the coming chapters, we'll explore both the philosophical foundations and practical tools of digital independence. Whether you're a developer looking to build these systems or someone seeking to use them, understanding these foundations is essential. Because digital independence isn't given - it's learned, practiced, and ultimately, earned.

5 The Path to Digital Money

From Ciphers to Smart Contracts

The story of cryptocurrency begins long before Bitcoin. It starts with a simple question that has challenged mathematicians and philosophers for millennia: How can we share secrets safely?

Ancient Rome used the Caesar cipher to protect military communications. Medieval merchants developed complex codes to secure trade routes. During World War II, the breaking of the Enigma machine's encryption changed the course of history. Each advance in cryptography came from the need to communicate privately in an unsafe world.

But the true revolution began in the 1970s with two breakthroughs that would eventually make digital money possible: public key cryptography and the development of secure network protocols.

In 1976, Whitfield Diffie and Martin Hellman solved a problem that had seemed impossible: how could two people who had never met establish a shared secret over an insecure channel? Their solution, known as public key cryptography, created the foundation for secure digital communications. A few years later, Ron Rivest, Adi Shamir, and Leonard Adleman developed the RSA algorithm, making these theoretical ideas practical.

The 1980s saw the birth of the Cypherpunk movement. These technologists and privacy advocates believed that cryptography could protect individual liberty in the digital age. They weren't just mathematicians – they were philosophers who saw privacy as essential to human dignity and freedom.

David Chaum, a pioneer in cryptographic privacy, proposed the first digital cash system in 1983. His DigiCash company later implemented these ideas, but ultimately failed – partly because it remained centralized, requiring trust in a single company.

The 1990s brought both advances and setbacks. Phil Zimmermann released PGP (Pretty Good Privacy), bringing strong encryption to everyday users. The U.S. government, viewing strong cryptography as a national security threat, tried to mandate backdoors through the Clipper Chip. The resulting "Crypto Wars" ended with cryptography being classified as protected speech.

Through these battles, the Cypherpunks refined their vision. In 1993, Eric Hughes published "A Cypherpunk's Manifesto," declaring that "privacy is necessary for an open society in the

electronic age." The movement explored various approaches to digital money: Nick Szabo's bit gold, Wei Dai's b-money, and Adam Back's Hashcash all contributed crucial ideas.

Then came 2008. As the global financial system teetered on the brink of collapse, an anonymous figure calling themselves Satoshi Nakamoto published a paper titled "Bitcoin: A Peer-to-Peer Electronic Cash System." Bitcoin solved a problem that had stymied previous attempts at digital money: how to achieve consensus about ownership without trusting any central authority.

Bitcoin's genius lay in combining existing technologies in a novel way. It used public key cryptography for identity, cryptographic hash functions for mining, and a distributed ledger to record transactions. Most importantly, it created an economic incentive structure that made the system self-sustaining.

The early Bitcoin years were marked by experimentation and growing pains. The first known commercial transaction occurred in 2010 when Laszlo Hanyecz paid 10,000 BTC for two pizzas – bitcoins that would later be worth hundreds of millions of dollars. Mt. Gox, the largest Bitcoin exchange, collapsed in 2014, teaching harsh lessons about the risks of centralized custody.

By 2013, a young programmer named Vitalik Buterin saw both Bitcoin's potential and its limitations. Bitcoin's simple scripting language was intentionally restricted to prevent complex computations that could slow the network. Buterin proposed Ethereum, a platform that would add a complete programming language to blockchain technology.

Ethereum launched in 2015, introducing the concept of smart contracts – self-executing programs stored on the blockchain. This opened up entirely new possibilities. Instead of just transferring value, users could create complex financial instruments, digital organizations, and decentralized applications.

The next major innovation came in 2018 with the launch of Uniswap, which introduced automated market makers (AMMs) to crypto. Traditional order book exchanges require active market makers to provide liquidity. AMMs used smart contracts to create passive, algorithm-driven markets that could operate 24/7 without human intervention.

This sparked the DeFi (Decentralized Finance) revolution. Compound introduced algorithmic lending markets. Aave pioneered flash loans. Curve optimized stable asset trading. Each innovation built on previous ones, creating increasingly sophisticated financial infrastructure.

But Ethereum's success brought scaling challenges. High transaction fees during peak usage made smaller transactions impractical. This spurred development of alternative approaches. Solana launched in 2020, using a proof-of-history mechanism to achieve higher throughput. The Move programming language, first developed for Facebook's Libra project, influenced platforms like Aptos and Sui that prioritized safer smart contract development.

Layer 2 scaling solutions emerged as another approach. Optimistic rollups like Arbitrum and Optimism, and zero-knowledge rollups like zkSync and StarkNet, aimed to increase Ethereum's

capacity while inheriting its security. Each made different tradeoffs between speed, cost, and decentralization.

This brings us to today. We've moved from simple ciphers to programmable money, from centralized experiments to decentralized networks. Each step built on previous innovations while solving new challenges. Understanding this history helps us appreciate both how far we've come and the challenges that still lie ahead.

The cypherpunks believed privacy tools could reshape society. They were right, but perhaps not in the way they expected. Cryptocurrency has indeed changed how we think about money, but its greatest impact may be in showing that alternative financial systems are possible. As we look to the future, we carry forward their core insight: mathematical tools, properly designed, can create new forms of human coordination and freedom.

6 The Search for Alternatives

Promise and Compromise

When you deposit money in a bank, you're making a trade. You gain convenience but surrender control. Your money becomes an entry in someone else's database. Your privacy becomes a policy in someone else's rulebook. Your financial freedom becomes subject to someone else's discretion.

For decades, we accepted this trade because we had no alternative. But different groups have increasingly questioned whether this bargain serves their interests. Their reasons reveal both the promise of decentralized systems and the challenges they face.

6.1 Regular People's Motivations

For many people, the search for alternatives begins with personal experience. Perhaps it's a Venezuelan family watching inflation devour their savings, or a Nigerian entrepreneur unable to receive international payments. Maybe it's an American discovering their bank account has been frozen without explanation, or a privacy-conscious individual uncomfortable with every purchase being tracked.

These experiences share a common thread: the realization that traditional financial systems grant enormous power to intermediaries while offering users surprisingly few protections. Consider these common scenarios:

- A bank can freeze your accounts without warning or immediate recourse
- Payment processors can refuse to serve legal but "high-risk" businesses
- Governments can impose capital controls during economic crises
- Financial surveillance tracks nearly every transaction you make
- Inflation can steadily erode purchasing power
- International transfers incur high fees and lengthy delays

Each of these represents a failure of centralized systems to serve basic human needs: the need to save without fear of confiscation, to transact without excessive surveillance, to maintain privacy without sacrificing convenience.

6.2 Technologists' Motivations

For technologists, the appeal of decentralized systems often starts with recognizing their elegant solutions to complex problems. How do you create digital scarcity without central control? How do you achieve consensus among untrusting parties? How do you build systems that remain secure even when some participants are malicious?

But deeper motivations often emerge:

- The desire to build systems that can't be corrupted by concentrated power
- The technical challenge of creating truly trustless protocols
- The vision of enabling new forms of human coordination
- The goal of making financial services universally accessible
- The drive to reduce society's dependence on trusted intermediaries

6.3 Core Principles

These diverse motivations converge around several core principles:

- **Self-Sovereignty**: The idea that individuals should have direct control over their assets and data. This means holding private keys rather than trusting custodians, controlling your own identity rather than relying on centralized providers, and maintaining ownership of your data rather than surrendering it to platforms.
- **Censorship Resistance**: The ability to transact and communicate without permission from authorities. This doesn't just mean resistance to government censorship – it includes resistance to corporate policies, payment processor restrictions, and other forms of private sector control.
- **Trustless Systems**: Protocols that work through mathematical guarantees rather than institutional trust. Instead of having to trust that a bank will honor its obligations, users can verify the system's operation directly through code and cryptography.

6.4 The Compromise of Success

Yet crypto's growing success has brought compromise. The same industry that began as a rebellion against financial intermediaries now builds new forms of intermediation. Venture capital firms that once missed the internet boom now rush to stake their claims. Wall Street, initially dismissive, now sees opportunities for familiar forms of financial engineering.

This transformation is evident in several trends:

- The rise of centralized exchanges that function much like traditional brokerages
- The push toward regulatory compliance that recreates existing financial structures
- The focus on token prices over technological advancement
- The concentration of wealth and influence among early investors
- The emphasis on speculation over actual utility

Some compromises were perhaps inevitable. Mass adoption requires user-friendly interfaces, institutional investment needs regulatory clarity, and complex systems benefit from specialized service providers. But other changes represent a more fundamental drift from crypto's founding principles.

Consider how: - "Not your keys, not your coins" became "Trust our custody solution" - "Censorship resistant" became "Compliant with all regulations" - "Trustless" became "Trust our proprietary trading engine" - "Decentralized" became "Controlled by a small group of token holders"

6.5 Finding Balance

Yet this story isn't simply one of ideals corrupted by commerce. The reality is more nuanced. While some projects have abandoned core principles in pursuit of profit, others maintain a careful balance. True decentralization coexists with user-friendly interfaces. Privacy-preserving protocols operate alongside regulated exchanges.

The challenge ahead lies in preserving crypto's essential promise – financial sovereignty, censorship resistance, and trustless operation – while making these benefits accessible to ordinary users. This might mean:

- Building better self-custody solutions that match centralized convenience
- Creating privacy-preserving compliance mechanisms
- Developing truly decentralized governance systems
- Focusing on real-world utility over speculation
- Maintaining open protocols alongside commercial services

The original vision of cryptocurrency remains powerful: a world where financial freedom doesn't require anyone's permission, where privacy is protected by mathematics rather than policies, and where trust comes from transparent code rather than opaque institutions. Realizing this vision means neither rejecting all compromise nor accepting every dilution of principle, but rather finding ways to make radical ideas practical.

For both regular people and technologists seeking alternatives to centralized systems, the core question remains: How do we build systems that preserve freedom while serving human needs? The answer may lie not in choosing between idealism and practicality, but in finding ways to achieve both.

7 Tools

7.1 Introduction

The Web3 ecosystem requires specialized tools for interacting with blockchains, managing digital assets, and analyzing on-chain data. This guide provides a detailed overview of essential tools across different categories, helping users understand when and how to use each one effectively.

7.2 Wallets

Wallets serve as your primary interface with blockchain networks. Understanding the different types and their security implications is crucial for safely participating in Web3.

7.2.1 Hardware Wallets

Hardware wallets store private keys in secure hardware devices, providing the highest level of security for long-term storage.

7.2.1.1 Ledger

- Supports multiple application models (Account, UTXO, Resource, Actor)
- Available on desktop and mobile
- Proprietary secure element chip
- Regular firmware updates
- Supports 5000+ cryptocurrencies

Key features:

- Clear signing screen for transaction verification
- Secure element certification
- Desktop and mobile companion apps
- DApp integration capabilities

7.2.1.2 Trezor

- Open-source hardware and firmware
- Supports Account (EVM) and UTXO models
- Browser-based interface
- Shamir backup feature
- Focuses on Bitcoin and EVM chains

7.2.2 Software Wallets

Software wallets offer greater convenience but with increased security risks compared to hardware solutions.

7.2.2.1 MetaMask

- Primary gateway to EVM networks
- Browser extension and mobile app
- Built-in token swap feature
- Custom network support
- Extensive DApp integration

Best practices:

- Never share seed phrase
- Use with hardware wallet for large amounts
- Regularly check token approvals
- Keep browser extension updated

7.2.2.2 Phantom

- Multi-chain support (Solana, EVM, Bitcoin)
- Mobile and browser extension
- Built-in NFT support
- SOL staking integration
- Token swap functionality

7.2.2.3 Safe (formerly Gnosis Safe)

- Multi-signature wallet platform
- Advanced security features
- Treasury management tools
- Transaction batching
- Custom access controls

7.3 Centralized Exchanges (CEXs)

While centralized exchanges present counterparty risks, they remain crucial infrastructure for entering and exiting the crypto ecosystem.

7.3.1 Key Considerations for CEX Selection

1. Security Track Record

- Historical hacks or breaches
- Insurance coverage
- Cold storage policies
- Security certifications

2. Liquidity Depth

- Trading volume verification
- Order book depth
- Market maker relationships
- Wash trading detection

3. Regulatory Compliance

- Licensing status
- Jurisdictional restrictions
- KYC/AML procedures
- Asset segregation

4. Feature Set

- Supported cryptocurrencies
- Trading pair availability
- Fiat on/off ramps
- Advanced trading features

7.3.2 Notable Exchange Failures and Lessons

1. Mt. Gox (2014)

- Loss: 850,000 BTC
- Lesson: Importance of proof of reserves

2. FTX (2022)

- Loss: $8-10 billion
- Lesson: Dangers of commingled funds

3. Celsius Network (2022)

- Loss: $4.7 billion
- Lesson: Risks of CeFi lending platforms

7.4 Analysis Tools

7.4.1 On-Chain Analysis Platforms

7.4.1.1 Block Explorers

Block explorers provide detailed information about transactions, addresses, and smart contracts on specific networks.

1. Etherscan (Ethereum)

- Transaction tracking
- Contract verification
- Gas tracker
- Token approvals
- API access

2. Solana Explorer

- Program interaction tracking
- Account management
- Stake delegation monitoring
- Vote account tracking

7.4.1.2 Portfolio Trackers

7.4.1.2.1 DeBank

- Comprehensive DeFi position tracking
- Cross-chain support
- Real-time updates
- Historical performance
- Risk monitoring

Best for:

- Active DeFi users
- Multi-chain portfolios
- Yield farming tracking

7.4.1.2.2 Zapper

- NFT integration
- Bridge tracking
- Portfolio history
- DeFi dashboard
- Cross-chain support

Ideal for:

- NFT collectors
- DeFi participants
- Multi-chain users

7.4.2 Professional Analysis Tools

7.4.2.1 Data Query Platforms

1. Dune Analytics

- SQL-based queries
- Custom dashboard creation
- Community-driven insights
- Historical data access
- Real-time analytics

2. Nansen

- Wallet labeling
- Smart money tracking
- Token flow analysis
- NFT market insights
- Investment signals

7.5 Security Tools

7.5.1 Smart Contract Analysis

7.5.1.1 OpenZeppelin

- Security auditing tools
- Contract templates
- Upgrade management
- Access control
- Gas optimization

Best practices:

- Regular security reviews
- Automated testing
- Upgrade planning
- Access control management

7.6 Privacy Tools

Privacy tools help protect user identity and transaction privacy while interacting with Web3 platforms.

7.6.1 VPN Services

When selecting a VPN for Web3:

- Look for no-log policies
- Check jurisdiction
- Verify cryptocurrency payment support
- Test connection stability
- Evaluate server locations

Recommended Services:

- Mullvad VPN (accepts Bitcoin, focused on privacy)
- ProtonVPN (cryptocurrency support, Swiss jurisdiction)

7.6.2 TOR Network

- Provides network-level privacy
- Multiple relay encryption
- Bridge support for censorship resistance
- Integration with privacy-focused tools

8 Protecting Your Assets

The decentralized nature of cryptocurrency creates unique security challenges. While traditional finance offers safety nets like fraud protection and account recovery, crypto operates on the principle of absolute ownership - which means absolute responsibility. This guide will help you understand and protect against the major ways people lose their crypto assets.

8.1 Understanding Private Key Security

Your private key is like the master key to a vault - anyone who has it can access everything inside. Unlike a physical key, it can't be copied by someone who briefly sees it, but it also can't be replaced if lost. This creates two opposing risks we must balance: the risk of loss and the risk of theft.

8.1.1 Securing Your Private Key

Think of your private key (usually represented as a seed phrase) as the most sensitive information you own. Good security practices include:

1. Physical Security

    - Write your seed phrase on durable materials (steel or titanium for long-term storage)
    - Store copies in multiple secure locations
    - Consider dividing the phrase into parts stored separately
    - Never store digitally or take photos

2. Access Planning

    - Create a clear inheritance plan
    - Document recovery procedures for family members
    - Consider multi-signature setups for large holdings
    - Test recovery procedures periodically

8.1.2 Common Private Key Mistakes

Many losses occur through simple oversights:

- Taking photos of seed phrases
- Storing phrases in cloud services or password managers
- Using phrases generated by others
- Entering phrases on suspicious websites
- Sharing phrases with "support staff"

8.2 Understanding Technical Risks

Technical risks often arise from misunderstanding how blockchain systems work. Let's examine the most common technical failures and how to prevent them.

8.2.1 Network Selection Errors

Blockchain networks are separate universes - sending assets to the wrong network often means permanent loss. Protection requires:

1. Always verify the network before transactions
2. Start with small test transactions
3. Use address book features in wallets
4. Understand bridge mechanisms between networks

8.2.2 Gas and Transaction Mechanics

Transaction failures often come from misunderstanding gas (transaction fees):

1. Low Gas Issues
    - Transactions can get stuck
    - Some tokens can become temporarily locked
    - Emergency cancellation may require high fees
2. High Gas Mistakes
    - Overpaying during network congestion
    - Not understanding fee calculations
    - Falling for gas token scams

8.2.3 Smart Contract Interactions

Smart contracts introduce complex risks:

1. Token Approvals
    - Never approve unlimited spending
    - Regularly review and revoke approvals
    - Use token allowance checkers
    - Understand the contracts you're interacting with
2. Contract Verification
    - Check contract addresses on block explorers
    - Verify official documentation
    - Be wary of cloned contract names

8.3 Understanding Social Engineering

Social engineering attacks exploit human psychology rather than technical vulnerabilities. These attacks are particularly dangerous because they bypass security measures by tricking you into taking harmful actions.

8.3.1 Common Attack Patterns

1. Authority Exploitation
    - Fake customer support
    - Impersonated team members
    - False urgency messages
    - Regulatory compliance scams
2. FOMO (Fear of Missing Out) Manipulation
    - Limited time offers
    - Exclusive access promises
    - Artificial scarcity
    - Pump and dump schemes
3. Trust Exploitation
    - Fake testimonials
    - Manufactured social proof

- Community infiltration
- Long-term relationship building

8.3.2 Protection Strategies

1. Verification Procedures

   - Always use official channels
   - Verify team member identities
   - Check multiple sources
   - Never act under time pressure

2. Communication Hygiene

   - Ignore direct messages about crypto
   - Never share private information
   - Be skeptical of unsolicited offers
   - Verify URLs carefully

8.4 Smart Contract Vulnerabilities

Smart contract risks require special attention because they can affect many users simultaneously and often can't be fixed once discovered.

8.4.1 Risk Categories

1. Implementation Flaws

   - Logic errors
   - Mathematical errors
   - Access control issues
   - Race conditions

2. Economic Vulnerabilities

   - Flash loan attacks
   - Price manipulation
   - Liquidity attacks
   - Governance attacks

3. External Dependencies

- Oracle failures
- Bridge compromises
- Network congestion
- Protocol interactions

8.4.2 Protection Measures

1. Due Diligence

   - Check audit reports
   - Review attack history
   - Understand dependencies
   - Monitor protocol metrics

2. Risk Management

   - Start with small amounts
   - Diversify across protocols
   - Monitor security alerts
   - Maintain exit strategies

8.5 Building Security Habits

Security in crypto requires developing consistent habits:

1. Regular Security Reviews

   - Check wallet connections
   - Review token approvals
   - Update security software
   - Test backup procedures

2. Transaction Hygiene

   - Verify all details multiple times
   - Use test transactions for new operations
   - Maintain separate wallets for different purposes
   - Keep detailed records

3. Continuous Learning

   - Study new attack vectors
   - Update security practices

- Share knowledge with others
- Learn from others' mistakes

Remember: In crypto, security isn't a destination - it's a continuous process of learning, adapting, and staying vigilant. The best security measures are the ones you actually use consistently.

Part III

Path to Independence

9 Starting Your Web3 Journey

9.1 Phase 1: Understanding the Basics

Before touching any money or creating accounts, let's build your foundation of knowledge. Think of this like learning to recognize road signs before driving a car.

9.1.1 Essential Terms

Start by familiarizing yourself with the fundamental vocabulary in Chapter "Terms". Key concepts you need to understand first include:

- Web3 and decentralization
- Blockchain basics
- Public and private keys
- Gas fees and transaction costs
- Smart contracts

9.1.2 Cultural Context

The Web3 community has its own culture, largely expressed through memes. Review Chapter "Memes" to understand:

- "Not your keys, not your coins" - why self-custody matters
- "HODL" - the philosophy of long-term holding
- The evolution of different crypto communities
- Common scam warnings and red flags

9.1.3 Required Tools

You'll need specific tools to interact with Web3. Reference Chapter "Tools" for details on:

- Wallets (MetaMask, hardware wallets)
- Block explorers
- Portfolio trackers

- Security tools

9.2 Phase 2: Security First

Before handling any real money, we need to establish secure practices. Security isn't optional - it's the foundation everything else builds upon.

9.2.1 Protecting Your Assets

From the Chapter "Don't Lose Your Money":

1. Private Key Management
    - Never share your seed phrase
    - Secure storage methods
    - Backup procedures
    - Emergency access plans
2. Creating Safe Wallets
    - Setting up MetaMask securely
    - Creating a "burner" wallet for testing
    - Understanding hardware wallet benefits
    - Using multiple wallets for different purposes
3. Privacy Considerations
    - The Tornado Cash situation
    - Legal implications of privacy tools
    - Alternative privacy methods
    - Balance between privacy and compliance

9.3 Phase 3: Getting Started

Now that you understand the basics and security fundamentals, it's time to enter the ecosystem.

9.3.1 Your First Crypto Purchase

Following the Chapter "First Steps":

1. Start Small

    - Buy your first $10 of Bitcoin
    - Understand exchange basics
    - Learn about order types
    - Practice secure withdrawals

2. Understanding Fees

    - Different fee markets (overview with links to Technical section)
    - How gas prices work
    - Choosing the right time to transact
    - Estimating transaction costs

9.3.2 Creating Your Web3 Identity

1. Setting Up MetaMask

    - Proper installation and security
    - Network configuration
    - Backup procedures
    - Test transactions

2. Creating Clean Wallets

    - Understanding wallet separation
    - Privacy tool options
    - Legal considerations
    - Operational security

9.4 Phase 4: Basic Operations

Now you're ready to start using Web3 services while maintaining security.

9.4.1 Essential Skills

1. Using DEXes (Decentralized Exchanges)
    - Trading on Uniswap
    - Understanding liquidity pools
    - Managing slippage
    - Avoiding common pitfalls
2. Bridging Between Networks
    - Understanding bridge risks
    - Choosing reliable bridges
    - Managing cross-chain assets
    - Minimizing bridge costs

9.4.2 DeFi Fundamentals

1. Lending Markets
    - How lending protocols work
    - Borrowing against your Bitcoin
    - Managing collateral ratios
    - Understanding liquidation risks
2. Yield Strategies
    - Basic yield farming
    - Risk assessment
    - Sustainable practices
    - Tax considerations

9.5 Phase 5: Advanced Operations

Once comfortable with basics, you can explore more sophisticated strategies.

9.5.1 Sustainable Practices

1. Cash Flow Management

    - Using lending markets effectively
    - Managing borrowed positions
    - Creating sustainable yield
    - Emergency procedures

2. Off-ramping Strategies

    - Converting crypto to fiat
    - Tax compliance
    - Banking relationships
    - Local regulations

10 Your First Steps

10.1 Introduction

Taking your first steps into Web3 can feel like learning to walk in a new world. The concepts may be unfamiliar, and the stakes feel high since real money is involved. This chapter will guide you through your initial journey, ensuring you start with strong fundamentals while maintaining security at every step.

10.2 Your First Cryptocurrency Purchase

10.2.1 Choosing Your Entry Point

Your first cryptocurrency purchase represents more than just buying digital assets—it's about learning to navigate a new financial system. We'll start small, with just $10 worth of BTC (Bitcoin). This amount is chosen carefully: it's enough to learn the mechanics but small enough that mistakes won't be devastating.

10.2.2 Selecting an Exchange

For your first purchase, we'll use a regulated cryptocurrency exchange. While decentralized options exist, centralized exchanges offer important advantages for beginners:

- Familiar payment methods (bank transfers, credit cards)
- Customer support for common issues
- Regulated entities with clear legal obligations
- Simple user interfaces

Popular options include:

- Coinbase: Known for ease of use (recommended for this guide)
- Kraken: Strong security history
- Gemini: Regulatory compliance focus

10.2.3 Step-by-Step Purchase Process

1. Registration

    - Use a strong, unique password
    - Enable two-factor authentication **immediately**
    - Complete identity verification (KYC)
    - Secure your recovery options

2. Funding Your Account

    - Start with a small test deposit
    - Document all transaction details
    - Understand processing timeframes
    - Verify fees before proceeding

3. Making Your First Purchase

    - Navigate to the Bitcoin trading page
    - Select "Buy" or "Market Buy"
    - Enter $10 USD (or local equivalent)
    - Review the quoted price and fees
    - Confirm the transaction

4. Understanding Order Types

    - Market Orders: Instant execution at current price
    - Limit Orders: Set your desired price
    - Stop Orders: Automated triggers for buying/selling
    - Pros and cons of each approach

10.2.4 After Your Purchase

Immediately after buying, familiarize yourself with:

- Transaction history viewing
- Price alerts setting
- Account statements
- Tax reporting requirements

10.3 Creating Your Web3 Identity

10.3.1 The Importance of Self-Custody

While keeping your first Bitcoin purchase on the exchange is acceptable temporarily, true participation in Web3 requires self-custody through wallets you control. This begins with setting up Phantom.

10.3.2 Phantom Setup Guide

1. Installation
    - Use official sources only
    - Chrome/Firefox/Brave supported
    - Mobile options available
    - Verify extension authenticity
2. Initial Configuration
    - Create new wallet
    - Record seed phrase properly
    - Set strong password
    - Understand recovery options
3. Security Best Practices
    - Never share seed phrase
    - Use hardware wallet for large amounts
    - Regularly check connected sites
    - Update extension promptly
4. Network Configuration
    - Understanding Bitcoin Mainnet
    - Recognizing test networks
    - Managing network switching

10.3.3 Creating Clean Wallets

As you progress in Web3, wallet separation becomes crucial. Think of wallets like different bank accounts—each serving a specific purpose.

1. Wallet Types
    - Main Wallet: Your primary identity
    - Trading Wallet: For DeFi interactions
    - Gaming Wallet: For Web3 games
    - Test Wallet: For trying new protocols
2. Privacy Considerations
    - Transaction history is public
    - Address clustering risks
    - Block explorer visibility
    - Network analysis implications
3. Operational Security
    - Different devices for different wallets
    - Clean transaction patterns
    - Cross-chain considerations
    - Interaction compartmentalization
4. Legal and Privacy Tools
    - VPN usage pros and cons
    - Mixer considerations
    - Jurisdiction awareness
    - Compliance documentation

10.4 Your First Transactions

10.4.1 Understanding Network Fees

Before making your first transaction, understand that every blockchain action has a cost:

1. Fee Components
    - Base network fees
    - Priority fees (tips)
    - Contract interaction costs

- Failed transaction fees

2. Timing Considerations

    - Network congestion patterns
    - Gas price variations
    - Weekend vs weekday differences
    - Time zone impacts

10.4.2 Practice Transactions

Start with small test transactions to build confidence:

1. Send a minimal amount between your own wallets
2. Interact with a simple smart contract
3. Add/remove liquidity from a DEX
4. Try a cross-chain bridge

10.4.3 Common Pitfalls to Avoid

1. Technical Mistakes

    - Insufficient gas allocation
    - Wrong network selection
    - Incorrect address input
    - Contract approval limits

2. Security Risks

    - Phishing websites
    - Fake tokens
    - Malicious smart contracts
    - Social engineering attempts

10.5 Next Steps

After completing these initial steps, you'll be ready to:

- Explore DeFi protocols
- Participate in DAOs
- Trade on DEXs
- Investigate yield opportunities

Remember: The goal of these first steps isn't to make money—it's to build a strong foundation for your Web3 journey. Take your time, double-check everything, and don't rush into complex interactions until you're completely comfortable with the basics.

10.6 Emergency Procedures

Keep this information readily available:

- How to revoke contract approvals
- Emergency contact information for exchanges
- Local crypto-friendly legal resources
- Asset recovery procedures

Stay curious, but always prioritize security and understanding over speed and potential profits.

11 DeFi Fundamentals

11.1 Introduction

Decentralized Finance (DeFi) represents a fundamental reimagining of financial services. While traditional finance relies on banks, brokers, and other intermediaries, DeFi uses smart contracts – self-executing computer programs – to enable direct peer-to-peer financial activities. This shift removes gatekeepers and creates opportunities for anyone with an internet connection to access sophisticated financial services.

While Ethereum pioneered DeFi, its high transaction costs can make learning expensive, with fees sometimes exceeding $50 per transaction. This is why we'll start our journey on Base, a Layer 2 network built on top of Ethereum. Base offers the same capabilities but with dramatically lower fees, typically under $1 per transaction. This makes it perfect for learning DeFi without fear of expensive mistakes.

11.2 Getting Your First ETH on Base

Now that you have Bitcoin in your Phantom wallet from following our earlier steps, we need to get some ETH on Base to begin exploring DeFi. We'll use Coinbase for this process since they built Base and offer direct deposits.

11.2.1 Direct Purchase Method

The most straightforward approach is:

1. On Coinbase Exchange:
    - Purchase $50-100 worth of ETH
    - Select "Withdraw"
    - Choose "Send to Base"
    - Send to your Phantom wallet's Base address

This method is optimal because: - Coinbase handles the bridge transaction for you - You avoid paying Ethereum mainnet gas fees - The process is simpler than manual bridging - Base transaction fees are much lower

11.2.2 Understanding Base vs Ethereum

Base maintains a special relationship with Ethereum: - It inherits Ethereum's security - Uses the same wallet addresses - Runs the same smart contracts - Costs much less to use

Think of Base like an express lane on a highway - it's faster and cheaper while still getting you to the same destination safely.

11.3 Your First DeFi Steps

Before diving into trading and lending, let's verify everything is working:

1. Initial Setup Check
    - Confirm ETH appears in your Phantom wallet on Base
    - Ensure you've selected Base network
    - Have your Phantom wallet connected to Base
    - Keep transaction records for taxes
2. Test Transaction
    - Send a tiny amount of ETH ($1 worth) between your own addresses
    - Notice how the gas fees are much lower than Ethereum
    - Get comfortable with Base network mechanics
    - Understand transaction confirmation times

11.4 Trading Fundamentals

When it comes to trading tokens on Base, using a DEX aggregator provides significant advantages over using any single decentralized exchange. Let's understand why and how to trade effectively.

11.4.1 Understanding DEX Aggregators

DEX aggregators are like smart shopping assistants that find the best prices across multiple exchanges. We recommend ParaSwap for several reasons:

1. Gas Efficiency
    - Optimizes routes to minimize gas costs
    - Particularly effective on Base's already low-fee environment
    - Helps make small trades viable

2. User Experience
    - Clean, intuitive interface
    - Clear transaction previews
    - Easy-to-understand settings
3. Price Optimization
    - Splits orders across multiple DEXs when beneficial
    - Protects against price impact
    - Finds efficient trading paths

11.4.2 Your First Trade Using ParaSwap

Let's walk through converting some ETH to USDC:

1. Preparation
    - Visit ParaSwap's official website (always verify the URL)
    - Connect your Phantom wallet
    - Verify you're on Base network
    - Ensure sufficient ETH for both trade and gas
2. Making the Trade
    - Select ETH as your "from" token
    - Choose USDC as your "to" token
    - Enter a small test amount ($10 worth)
    - Review the quoted price and gas fees
    - Understand the suggested route
    - Confirm the transaction

11.4.3 Understanding Slippage and Price Impact

When trading on Base, it's crucial to understand how trade size affects price:

1. Slippage
    - The difference between expected and executed price
    - Larger trades typically experience more slippage
    - ParaSwap helps minimize this through smart routing
2. Price Impact
    - How your trade affects market price
    - Depends on available liquidity
    - Visible before trade execution

11.5 Understanding Gas on Base

While Base uses the same gas system as Ethereum, the costs are much lower:

1. Typical Base Costs
    - Simple transfers: $0.01-0.05
    - Token swaps: $0.10-0.30
    - Complex DeFi operations: $0.20-0.50 Compare this to Ethereum's fees which can be 50-100x higher!
2. Gas Management
    - Keep $5-10 worth of ETH for gas
    - Monitor network conditions
    - Understand peak usage times
    - Plan non-urgent transactions

11.6 Lending Markets with Aave on Base

Now that you're comfortable with basic transactions and trading, let's explore lending markets. Aave is one of DeFi's most battle-tested lending protocols and has deployed on Base.

11.6.1 How Aave Works

Think of Aave like a decentralized bank where you can both lend and borrow crypto assets:

1. Lending Assets When you lend on Aave:
    - You receive aTokens representing your deposit
    - Interest accrues automatically in your wallet
    - You can withdraw anytime (subject to liquidity)
    - Your deposit can serve as collateral for borrowing
2. Understanding Collateral and Borrowing Aave uses an overcollateralized lending model:
    - You must maintain a minimum health factor
    - Different assets have varying collateral factors
    - Monitor your position's health regularly
    - Understand liquidation risks
3. Safety First
    - Always maintain a health factor above 1.5
    - Set up notifications for position health

- Keep some ETH for emergency transactions
- Know how to repay loans quickly

11.7 Monitoring Your DeFi Activity

Stay informed about your DeFi positions:

1. Essential Tools
    - DeFiLlama for protocol TVL and health
    - Base block explorer for transactions
    - Aave dashboard for loan positions
    - ParaSwap analytics for trading data
2. Regular Checks
    - Monitor lending positions daily
    - Review transaction history weekly
    - Track portfolio performance
    - Stay updated on protocol changes

Remember: While Base's low transaction costs make DeFi more accessible, never invest more than you can afford to lose. Start small, learn the mechanics, and scale up gradually as you gain confidence and understanding.

12 What Should I Invest In?

The path to financial independence through cryptocurrency requires careful consideration of historical data, risk tolerance, and investment timeline. While many assets and strategies exist in the cryptocurrency space, this guide will focus on what has proven most reliable for long-term wealth building.

12.1 Understanding the Bitcoin Investment Thesis

When examining historical cryptocurrency returns, Bitcoin has demonstrated remarkable consistency over 8-year holding periods:

- 2013-2021: ~4,000% return
- 2014-2022: ~2,900% return
- 2015-2023: ~8,500% return
- 2016-2024: ~17,500% return

These returns suggest that patient accumulation of Bitcoin over long periods has been the most reliable path to wealth building in cryptocurrency. However, it's important to understand why simply calculating future value using traditional compound interest formulas doesn't tell the whole story.

12.2 Why Traditional Investment Calculations Fall Short

Traditional future value calculations assume consistent periodic investments of equal purchasing power. With Bitcoin, this assumption doesn't hold true because:

1. The dollar amount you invest monthly buys different amounts of Bitcoin as the price changes
2. Market cycles create varying entry points that significantly impact returns
3. The deflationary nature of Bitcoin means later purchases may acquire less BTC than earlier ones

Despite these mathematical complexities, the historical data presents a compelling case: consistent Bitcoin accumulation over 8+ year periods has outperformed virtually every other asset class.

12.3 The Simple Truth About Crypto Investment

After analyzing countless strategies, tokens, and approaches, we've reached a straightforward conclusion: for most people, the optimal cryptocurrency investment strategy is:

1. Save what you can afford monthly (even if it's just $10)
2. Convert it to Bitcoin
3. Hold for at least 8 years
4. Repeat

This advice might seem overly simple, but it's based on several key observations:

- Bitcoin's network effect and first-mover advantage continue to strengthen
- Alternative cryptocurrencies (altcoins) have historically underperformed BTC over long periods
- Complex trading strategies typically underperform simple accumulation
- The 8-year holding period has captured full market cycles historically

12.4 For Those Seeking More Detail

While we maintain that simple Bitcoin accumulation is optimal for most investors, we understand some readers may want to explore additional options. For those interested in evaluating other cryptocurrencies or investment strategies:

1. Review our comprehensive ratings system in the Ratings section
2. Understand the risk factors we've identified for different asset types
3. Study the technical fundamentals behind various protocols
4. Consider your own risk tolerance and investment timeline

However, we strongly caution that additional complexity rarely leads to better returns for most investors.

12.5 A Note on Financial Independence

The goal of this investment approach is to achieve financial independence - having enough passive income to cover living expenses without relying on central authorities. While the exact amount needed varies by location and lifestyle, consistently accumulating Bitcoin has historically provided a path toward this goal.

The beauty of this approach is its accessibility - whether you can save $10 or $10,000 monthly, the strategy remains the same: consistent accumulation and patient holding of Bitcoin.

12.6 Conclusion

While cryptocurrency offers many investment opportunities, our analysis suggests that simple is better. Regular Bitcoin accumulation over long time periods has proven the most reliable path to wealth building in this space. Rather than seeking complex strategies or alternative assets, focus on:

1. Establishing a consistent savings habit
2. Converting savings to Bitcoin regularly
3. Securing your Bitcoin properly
4. Maintaining conviction through market cycles

For most people, this straightforward approach will likely outperform more complex strategies while requiring less time, stress, and expertise.

13 The Evolution of Stablecoins

Stablecoins represent one of the most important innovations in cryptocurrency, bridging the gap between volatile crypto assets and stable traditional currencies. However, their history contains important lessons about the challenges and risks of maintaining stability in decentralized systems.

13.1 The Great Unstablecoin: Understanding UST's Collapse

The collapse of Terra's UST stablecoin in May 2022 stands as a watershed moment in cryptocurrency history. To understand why it failed and what it teaches us, we need to examine its mechanism and the events that led to its downfall.

13.1.1 How UST Worked

Terra's UST attempted to maintain its dollar peg through an algorithmic relationship with LUNA, its sister token. The system worked like this:

- Users could always burn 1 UST to mint $1 worth of LUNA
- Users could always burn $1 worth of LUNA to mint 1 UST
- This mechanism was supposed to keep UST at $1 through arbitrage

The fatal flaw lay in its circular dependency: UST's stability relied entirely on LUNA's value, which in turn derived much of its value from UST's utility. This created a potential death spiral if confidence wavered.

13.1.2 The Death Spiral Unfolds

On May 7, 2022, large UST withdrawals from Anchor Protocol (which was offering unsustainable 20% yields) triggered the beginning of the collapse:

1. Initial selling pressure pushed UST slightly below its peg
2. Traders began redeeming UST for LUNA, increasing LUNA's supply
3. LUNA's price started falling due to increased supply
4. This caused more UST holders to redeem, fearing loss of the peg

5. The cycle accelerated, leading to hyperinflation of LUNA and complete collapse of UST

Within a week, over $40 billion in value was destroyed, affecting millions of users and triggering contagion throughout the crypto ecosystem.

13.2 Understanding Different Stablecoin Models

The UST collapse highlighted the importance of understanding different stablecoin designs and their risk profiles.

13.2.1 1. Fiat-Backed Stablecoins (e.g., USDC, USDT)

- Backed 1:1 by traditional currency in bank accounts
- Requires trust in the issuer and banking system
- Most straightforward but most centralized
- Examples: USDC by Circle, USDT by Tether

13.2.2 2. Crypto-Collateralized Stablecoins (e.g., DAI)

- Backed by cryptocurrency held in smart contracts
- Requires overcollateralization to handle volatility
- More decentralized but less capital efficient
- Examples: DAI by MakerDAO, sUSD by Synthetix

13.2.3 3. Algorithmic Stablecoins (e.g., failed UST)

- Attempts to maintain peg through algorithmic incentives
- No direct collateral backing
- Highest risk of catastrophic failure
- Historical examples: UST, IRON, BASIS

13.2.4 4. Hybrid Models (e.g., FRAX)

- Combines multiple stability mechanisms
- Partially collateralized, partially algorithmic
- Attempts to balance efficiency and stability
- Examples: FRAX, USDP

13.3 The New Wave: Understanding Delta-Neutral Stablecoins

A new class of stablecoins, exemplified by Ethena's USDe, has emerged that uses sophisticated financial engineering to maintain stability. Let's examine how they work and what makes them different.

13.3.1 How Delta-Neutral Stablecoins Work

These stablecoins maintain their peg through a combination of:

1. Spot holdings of the underlying asset (like ETH)
2. Short positions in perpetual futures markets
3. Sophisticated risk management systems

Taking USDe as an example:

- When users deposit ETH, the protocol:
  - Holds the ETH as collateral
  - Opens a corresponding short position
  - Mints USDe tokens
- The opposing positions cancel out price exposure
- Yield comes from funding rates on perpetual futures

13.3.2 Advantages of Delta-Neutral Design

- Not dependent on faith in the system (unlike UST)
- More capital efficient than overcollateralized models
- Generates sustainable yield from market neutral activities
- Less vulnerable to bank runs or confidence crises

13.3.3 Risks and Considerations

- Reliance on functioning derivatives markets
- Potential for basis risk between spot and futures
- Smart contract risks
- Market capacity limitations

13.4 Evaluating Stablecoin Safety

When assessing any stablecoin, consider these key factors:

1. Collateralization Method

- Is it fully backed by real assets?
- How can backing be verified?
- What's the liquidation mechanism?

2. Track Record

- How long has it maintained its peg?
- Has it weathered market stress?
- What's its daily trading volume?

3. Transparency

- Are reserves regularly audited?
- Is the mechanism fully documented?
- How decentralized is control?

4. Market Depth

- How much liquidity exists?
- What's the redemption process?
- How efficient is price discovery?

13.5 Building a Stablecoin Strategy

Given these considerations, here's a framework for using stablecoins safely:

1. Core Holdings

- Use proven, fully-backed stablecoins (USDC, USDT) for main positions
- Verify reserves and audits regularly
- Maintain multiple options for exit

2. Yield Generation

- Consider newer models like USDe for portion of holdings
- Start small with new protocols
- Monitor peg stability and market depth

3. Risk Management

- Never rely on a single stablecoin
- Keep majority in conservative options
- Have exit plans for each position

Remember that in the world of stablecoins, boring is often better. The goal is stability, not maximizing yield at any cost.

14 Moving Beyond Basic DeFi

Now that you're comfortable with basic DEX trading and lending protocols like Aave, let's explore more sophisticated ways to grow your crypto holdings through staking yields and strategic trading.

14.1 Understanding Liquid Staking

Staking is one of the most reliable ways to earn yield in crypto, but traditionally it came with a significant drawback: your assets were locked and unusable. Liquid staking tokens solve this problem by giving you a tradeable token that represents your staked assets.

14.1.1 How Liquid Staking Works

When you stake ETH through a liquid staking protocol like Lido, you receive stETH in return. This stETH token:

- Automatically accrues staking rewards
- Can be used as collateral in lending protocols
- Trades freely on DEXs
- Can be sold back to ETH when needed

The current staking yield for ETH is around 3-4% annually, but by using your liquid staking tokens strategically, you can potentially earn additional yield.

14.1.2 Smart Liquid Staking Strategies

Instead of just holding stETH, consider this yield-stacking approach:

1. Stake ETH to receive stETH
2. Use stETH as collateral on Aave
3. Borrow stable coins at a lower interest rate than your staking return
4. Use the borrowed stable coins for limit order opportunities

14.2 Setting Smart Limit Orders

Rather than trying to time the market perfectly, we can use data-driven approaches to set limit orders. Here's a strategy that uses moving averages to find potentially profitable entry points.

14.2.1 The 20/50 Moving Average Strategy

The strategy involves:

1. Observing the 20-day and 50-day moving averages (MA)
2. Setting limit orders 2-3% below the 20-day MA when it's above the 50-day MA
3. Setting limit orders 4-5% below the 20-day MA when it's below the 50-day MA

Here's how to implement this:

1. Use a charting tool like TradingView to view the moving averages
2. Calculate your limit order prices:
    - Normal market: Current 20MA price $\times$ 0.97
    - Weak market: Current 20MA price $\times$ 0.95

14.2.2 Example with ETH

Let's say ETH has:

- Current price: \$3,000
- 20-day MA: \$2,900
- 50-day MA: \$2,800

Since the 20-day MA is above the 50-day MA, we'd set our limit orders at: \$2,900 $\times$ 0.97 = \$2,813

This gives us a reasonable entry point below the moving average while still maintaining a strong technical position.

14.3 Combining Strategies for Maximum Efficiency

The real power comes from combining these approaches. Here's a comprehensive strategy:

1. Keep 50% of your ETH in liquid staking tokens earning base yield
2. Use 25% of your liquid staking tokens as collateral
3. Borrow stablecoins worth 15% of your collateral
4. Set limit orders with the borrowed stablecoins using the MA strategy

This approach provides:

- Baseline yield from staking
- Additional yield from lending
- Potential buying opportunities through limit orders
- Limited risk due to conservative borrowing

14.3.1 Risk Management

While these strategies can enhance your returns, they come with risks:

- Smart contract risk from multiple protocols
- Price impact risk if stETH depegs
- Liquidation risk from borrowed positions
- Opportunity cost if limit orders don't fill

To manage these risks:

- Never borrow more than 30% against your collateral
- Set stop-loss orders above your liquidation price
- Diversify across multiple liquid staking protocols
- Monitor the stETH/ETH peg regularly

14.4 Looking Ahead

As you become comfortable with these strategies, you can explore more advanced approaches:

- Multiple timeframe moving averages
- Yield farming with liquid staking tokens
- Cross-protocol arbitrage opportunities
- MEV-protected limit orders

Remember that in DeFi, simpler strategies often outperform complex ones over the long term. Start conservatively, monitor your positions regularly, and scale up only as you gain confidence in your understanding of the mechanics and risks involved.

The key to success with these strategies is patience and consistent execution rather than trying to maximize every possible yield opportunity. Focus on sustainable yields and high-probability setups rather than chasing the highest possible returns.

Part IV

Web3 Essentials

15 Meme Culture

15.1 Introduction

Memes serve as a unique form of cultural communication in the Web3 space, combining humor with important lessons and shared experiences. They often capture complex ideas in accessible formats and help create a sense of community among participants. This guide explores the most significant memes in Web3, their origins, and their deeper meanings.

15.2 Foundational Memes

15.2.1 "Not Your Keys, Not Your Coins"

This fundamental meme emerged from the early Bitcoin community and gained renewed significance after several high-profile exchange collapses. It encapsulates one of the core principles of cryptocurrency: self-custody.

Origin: The phrase likely originated on Bitcoin forums around 2011-2012 but gained prominence after the Mt. Gox collapse in 2014.

Cultural Impact: - Serves as a constant reminder of cryptocurrency's original purpose: financial self-sovereignty - Often shared after exchange hacks or failures - Has evolved into a teaching tool for newcomers - Frequently referenced in discussions about centralized exchanges and custody solutions

Real-world examples that reinforced this meme: - Mt. Gox (2014): 850,000 BTC lost - QuadrigaCX (2019): $190 million lost - FTX (2022): Billions in customer funds lost

15.2.2 HODL

Perhaps the most famous cryptocurrency meme, "HODL" originated from a typo that became a battle cry for long-term investors.

Origin: December 18, 2013, when GameKyuubi posted "I AM HODLING" on the BitcoinTalk forum during a market crash. The poster, admittedly drunk, wrote a passionate defense of holding Bitcoin despite price volatility.

Evolution: 1. Initial phase: Simple misspelling joke 2. Secondary meaning: “Hold On for Dear Life” 3. Modern usage: Philosophical approach to cryptocurrency investment

Cultural significance: - Represents conviction in long-term value over short-term trading - Creates solidarity among investors during market downturns - Differentiates investment approaches (HODLers vs. traders) - Has spawned related terms like “diamond hands”

15.3 Character-Based Memes

15.3.1 Wojak Variations

Wojak (also known as “feels guy”) has become the protagonist of countless crypto trading stories, with several important variations:

15.3.1.1 The Crypto Wojak Timeline:

1. Basic Wojak: Represents the average retail trader
2. NPC Wojak: Symbolizes those who follow the crowd without thinking
3. Brainlet: Depicts poor trading decisions
4. Cozy Wojak: Represents comfortable HODLers during market turbulence

15.3.2 The Bogdanoff Twins

A complex meme centered around the late Bogdanoff twins, imagining them as all-powerful market manipulators.

Key elements: - “Dump it”: Order given after someone buys - “Pump it”: Command issued after someone sells - “He sold?”: The setup for market manipulation

Cultural impact: - Reflects the feeling of powerlessness many traders experience - Personifies the seemingly manipulated nature of crypto markets - Creates humor from shared experiences of poor timing

15.4 Educational Memes

15.4.1 The Bell Curve (Midwit) Meme

This meme format illustrates the concept of sophisticated simplicity in cryptocurrency:

Left (Low IQ): Simple but effective approach Middle (Average IQ): Overcomplicated strategies Right (High IQ): Return to simplicity with deep understanding

Examples: - Bitcoin storage: Hardware wallet → Complex multi-sig setup → Hardware wallet with proper backup - Trading: HODL → Complex trading strategies → HODL with occasional rebalancing - Security: Write down seed phrase → Use elaborate encryption → Write down seed phrase and store in safe

15.5 Market Condition Memes

15.5.1 "This is Fine" Dog

A cartoon dog sitting in a burning room, often used during market crashes. The meme represents: - Forced calm during market turbulence - Coping with significant losses - The crypto community's resilience

15.5.2 "When Lambo?"

Origin: 2017 bull run Meaning: Represents the dream of crypto wealth Evolution: 1. Sincere question from newcomers 2. Ironic joke about unrealistic expectations 3. Critique of get-rich-quick mentality

15.6 Community-Specific Memes

15.6.1 Ethereum Memes

- "Vitalik clapping": Celebrating network achievements
- "Ultra sound money": ETH's post-merger monetary policy
- "Merge delayed": Historical references to Ethereum's upgrade timeline

15.6.2 Bitcoin Memes

- "Number go up": Bitcoin's long-term price appreciation
- "Stack sats": Encouraging regular small purchases
- "PayPal of crypto": Mocking misunderstandings of Bitcoin's purpose

15.7 Best Practices for Meme Literacy

Understanding crypto memes helps participants: 1. Navigate community sentiment 2. Identify market cycles 3. Learn from shared experiences 4. Connect with the broader culture

15.8 Modern Usage Guidelines

When engaging with crypto memes: - Understand their historical context - Recognize their educational value - Use them appropriately in discussions - Appreciate their role in community building

15.9 Conclusion

Memes in Web3 are more than just humor - they're a sophisticated form of cultural communication that encodes important lessons, shared experiences, and community values. Understanding these memes helps participants better navigate both the technical and social aspects of the cryptocurrency ecosystem.

As the space evolves, new memes emerge while old ones gain additional layers of meaning. This living document will be updated to reflect these changes and maintain its relevance as a cultural guide to Web3.

Part V

Technologist's Path

16 The Technologist's Path

The journey to understanding Web3 technology begins not with blockchains or cryptocurrencies, but with a more fundamental question: How do we create systems that enable cooperation without requiring trust? This question has driven decades of research in cryptography, distributed systems, and mechanism design. Now, these previously separate fields have converged in blockchain technology, creating new possibilities for human coordination.

16.1 The Foundation: Cryptographic Primitives

At its core, Web3 relies on cryptographic tools that provide mathematical guarantees rather than promises. Public key cryptography lets us create digital signatures that can't be forged. Hash functions generate unique digital fingerprints that can't be reversed. Zero-knowledge proofs allow us to verify facts without revealing underlying information. These building blocks make it possible to create systems with unprecedented security properties.

Consider how these tools work together: When you make a blockchain transaction, you use a private key to create a signature (public key cryptography), which is then combined with transaction data to create a unique identifier (hash functions), which might be verified privately (zero-knowledge proofs). Understanding these primitives helps you see not just how current systems work, but how future ones might be designed.

16.2 Building Blocks: Protocol Design

Moving up from cryptographic primitives, we encounter protocol design – the rules and mechanisms that govern how network participants interact. This is where we face fascinating engineering tradeoffs:

- How do we ensure network security without excessive energy consumption?
- What's the right balance between transaction throughput and decentralization?
- How can we make protocols upgradeable without creating centralization risks?

These questions have led to a proliferation of approaches. Some networks prioritize absolute security, others emphasize performance. Some opt for simplicity, others for flexibility. Each choice creates distinct characteristics that ripple through the entire technology stack.

16.3 The Execution Layer: Smart Contracts

Smart contracts represent a revolutionary advancement: programmable money. But writing them requires thinking differently about software development. Traditional programs can be updated when bugs are found. Smart contracts are often immutable – once deployed, they can't be changed. This permanence demands new approaches to:

- Security and formal verification
- Testing and deployment strategies
- Upgradeability patterns
- Inter-contract communication

Understanding smart contract development means learning not just new programming languages like Solidity, but new ways of thinking about program correctness and risk management.

16.4 The Data Layer: State and Storage

Blockchains introduce novel challenges around data management. Every node must process every transaction, making efficiency crucial. Different networks take varied approaches to state management:

- Account-based models like Ethereum
- UTXO models like Bitcoin
- Newer approaches like the Move language's resource model

Each model creates different tradeoffs in terms of parallelization, privacy, and programmability. Understanding these models helps developers choose the right platform for their specific needs.

16.5 The Network Layer: Communication and Consensus

Distributed networks must solve complex coordination problems. How do nodes discover each other? How do they agree on the current state? How do they handle network partitions? These questions have spawned various consensus mechanisms:

- Proof of Work's energy-intensive but highly secure approach
- Proof of Stake's economic-based security model
- Hybrid systems that attempt to combine advantages of multiple approaches

Each consensus mechanism influences the entire network's characteristics, from transaction finality to decentralization potential.

16.6 The Economic Layer: Incentive Design

Perhaps the most innovative aspect of blockchain systems is their use of economic incentives to maintain security and drive desired behaviors. This requires understanding:

- Game theory and mechanism design
- Token economics and monetary policy
- Market design and price discovery
- MEV (Miner/Maximal Extractable Value)

Good protocol design aligns incentives so that participants' self-interest promotes network health.

16.7 The Path Forward

As you explore these topics in depth, you'll discover how they interconnect. A change in consensus mechanism affects smart contract design. New cryptographic tools enable novel state management approaches. Economic incentives influence network behavior.

This interconnectedness means that while you might specialize in one area, understanding the broader context makes you a more effective developer or architect. The following sections will dive deep into each component, but always with an eye toward how it fits into the larger system.

Remember: The goal isn't just to learn how current systems work, but to understand the principles well enough to help design future ones. The field is still young, and many fundamental problems remain unsolved. Your journey through these topics isn't just about using existing tools – it's about helping to create what comes next.

17 Cryptography

Building Blocks of Digital Trust

Imagine trying to have a private conversation in a crowded room where everyone can hear you. Or proving you own something without showing your ID. These real-world challenges mirror the problems cryptography solves in the digital world. Before we can understand how cryptocurrencies work, we need to understand the cryptographic tools that make them possible.

17.1 The Evolution of Digital Privacy

The history of cryptography is a journey from simple code substitutions to sophisticated mathematical systems. While ancient Romans used the Caesar cipher to shift letters in the alphabet, modern cryptography uses advanced mathematics to achieve what might seem impossible: proving things without revealing secrets.

17.2 Symmetric Cryptography: Shared Secrets

Let's start with the simplest form of modern cryptography. Symmetric cryptography is like having a special key that both locks and unlocks a message. If Alice wants to send Bob a secret message, they need to share this key beforehand.

The most widely used symmetric algorithm is AES (Advanced Encryption Standard). It's incredibly fast and secure when used properly. However, it has one major challenge: how do Alice and Bob safely share the key in the first place? This is known as the key distribution problem.

This limitation led to one of the most important breakthroughs in cryptographic history.

17.3 Asymmetric Cryptography: The Public Key Revolution

In 1976, Whitfield Diffie and Martin Hellman proposed something revolutionary: what if you could have two different but mathematically related keys? One to lock (encrypt) and another to unlock (decrypt)? This became known as public key cryptography.

Here's how it works:

1. Every person has two keys: public and private
2. The public key can be freely shared with anyone
3. The private key must be kept secret
4. Messages encrypted with the public key can only be decrypted with the private key
5. Digital signatures created with the private key can be verified by anyone with the public key

This elegant system solves multiple problems:

- Secure communication without pre-sharing keys
- Digital signatures that can't be forged
- Identity verification without revealing secrets

17.3.1 The Math Behind Public Keys

While the mathematical details can be complex, the core idea relies on something called "trapdoor functions" - calculations that are easy to perform in one direction but extremely difficult to reverse. A common example is multiplication versus factoring: it's easy to multiply two large prime numbers, but very difficult to factor their product back into the original primes.

17.4 Hash Functions: Digital Fingerprints

Hash functions are perhaps the most widely used cryptographic tools in Web3. A hash function takes any input (a file, a message, or even an entire blockchain) and produces a fixed-size output that's like a unique digital fingerprint.

Good cryptographic hash functions have several crucial properties:

1. Deterministic: The same input always produces the same output
2. Fast to compute: You can quickly calculate the hash of any input
3. Pre-image resistant: Given a hash, it's infeasible to find the original input
4. Collision resistant: It's extremely unlikely to find two different inputs that produce the same hash

In blockchain systems, hash functions serve many purposes:

- Creating unique block identifiers
- Linking blocks together in a chain
- Generating addresses from public keys
- Securing the mining/validation process

17.5 Zero-Knowledge Proofs: Proving Without Revealing

One of the most powerful and counterintuitive concepts in modern cryptography is the zero-knowledge proof. Imagine proving you know your password without actually typing it, or proving you're over 21 without revealing your birthday.

Zero-knowledge proofs come in two main varieties:

17.5.1 Interactive Zero-Knowledge Proofs

These involve back-and-forth communication between the prover and verifier. Think of it like proving you know the solution to a Sudoku puzzle by answering specific questions about individual cells without showing the entire solution.

17.5.2 Non-Interactive Zero-Knowledge Proofs (ZK-SNARKs and ZK-STARKs)

These more advanced systems can create proofs that anyone can verify without interaction. They're crucial for privacy-preserving blockchain applications but require significant computational resources.

17.6 Homomorphic Encryption: Computing on Encrypted Data

Homomorphic encryption represents the cutting edge of cryptographic research. It allows computations to be performed on encrypted data without decrypting it first. While currently too computationally expensive for many applications, it holds promise for future privacy-preserving blockchain systems.

17.7 The Future: Post-Quantum Cryptography

Looking ahead, the development of quantum computers poses potential threats to current cryptographic systems, particularly public key cryptography. The field of post-quantum cryptography aims to develop new algorithms that remain secure even against quantum computers.

Key areas of research include:

- Lattice-based cryptography
- Hash-based signatures
- Multivariate cryptography
- Supersingular isogeny cryptography

17.8 Practical Applications in Web3

Understanding these cryptographic primitives helps explain how Web3 systems achieve their key properties:

1. Digital Signatures enable non-custodial wallets and transaction authorization
2. Hash functions create tamper-evident records and link blocks together
3. Zero-knowledge proofs power privacy-preserving transactions
4. Public key cryptography enables secure communication and identity verification

Each new Web3 innovation typically combines these basic tools in novel ways to solve specific problems. Understanding the tools helps you evaluate new protocols and anticipate their strengths and limitations.

18 Blockchain Technology

18.1 Introduction

Blockchain technology represents one of the most significant innovations in computer science since the internet itself. At its core, it solves a fundamental problem in digital systems: how to create absolute certainty about events and ownership without requiring trust in any central authority. This achievement opens up entirely new possibilities for human coordination and value exchange.

18.2 Fundamental Concepts

18.2.1 The Digital Trust Problem

Before blockchain, digital systems relied entirely on trusted intermediaries to maintain authoritative records. Your bank confirms your account balance, social media companies maintain your posts, and email providers manage your messages. This centralization creates vulnerabilities, requires faith in institutions, and imposes artificial limitations on digital interactions.

18.2.2 The Blockchain Solution

Blockchain technology introduces a radical alternative: a system where records are maintained simultaneously by thousands of computers worldwide, with sophisticated cryptographic mechanisms ensuring these records remain consistent, immutable, and accessible. When someone wants to add new information - whether it's a cryptocurrency transaction, smart contract execution, or any other digital event - this network of computers works together to verify, record, and protect that information.

18.3 Core Properties

18.3.1 1. Immutability

Once information is recorded on a blockchain, it becomes practically impossible to change. This isn't just a rule - it's a mathematical certainty based on cryptographic principles. Each block contains a hash of the previous block, creating a chain where altering any historical record would require simultaneously changing all subsequent blocks on thousands of computers.

18.3.2 2. Transparency

Every transaction and state change is visible to all participants. While the actors involved might be pseudonymous, their actions are public and verifiable. This creates unprecedented accountability in digital systems and enables new forms of coordination and trust.

18.3.3 3. Decentralization

No single entity controls the system. This distribution of power means that: - The network continues operating even if some participants fail - No single entity can unilaterally change the rules - Censorship becomes extremely difficult - Trust requirements are minimized

18.4 Application Models

The way blockchains handle state and computation fundamentally defines their capabilities and limitations. Different application models represent distinct approaches to managing blockchain data and computation.

18.4.1 UTXO Model (Unspent Transaction Output)

Bitcoin pioneered the UTXO model, which treats the blockchain like a ledger of unspent coins. Similar to physical cash, when you spend a Bitcoin UTXO, you consume it entirely and create new UTXOs as change.

Key characteristics: - Natural parallelization since each UTXO can only be spent once - Simple verification process - Limited programmability - Strong privacy properties - Excellent scalability potential

Implementation considerations: - Complex for smart contracts - Requires specialized development approaches - Better suited for simple value transfer - Enables efficient light clients

18.4.2 Account Model

Ethereum popularized the account model, which works more like a traditional bank account with persistent state. Each address maintains its own state, including: - Balance - Transaction nonce - Smart contract code (if applicable) - Contract storage

Advantages: - Intuitive for users and developers - Enables complex smart contracts - Simplifies application development - Natural fit for tokens and digital assets

Challenges: - More difficult to parallelize - Higher state growth - Potential privacy concerns - More complex to scale

18.4.3 Resource Model

The resource model, introduced by systems like Move, treats digital assets as unique resources that can only exist in one place at a time. This combines UTXO's safety with account model programmability.

Key features: - Linear types ensure resources can't be copied or destroyed - Natural representation of digital assets - Built-in protection against common vulnerabilities - Efficient parallel execution

Implementation details: - Requires specialized programming language support - Different development paradigm - Strong safety guarantees - Complex tooling requirements

18.4.4 Object Model

Object-oriented blockchain models treat each piece of state as a distinct object that can be owned and modified independently.

Benefits: - Natural modeling of complex assets - Clear ownership semantics - Efficient parallel processing - Flexible programming model

Considerations: - More complex state management - Different security considerations - Specialized development tools - Novel scaling approaches

18.5 Consensus Mechanisms

Consensus mechanisms determine how blockchain networks agree on the current state and validate new transactions. Different approaches make varying tradeoffs between speed, security, and decentralization.

18.5.1 Proof of Work (PoW)

Bitcoin's revolutionary consensus mechanism requires participants (miners) to solve complex mathematical puzzles to add new blocks.

Technical characteristics: - Based on SHA-256 or similar hash functions - Difficulty adjusts automatically - Natural fork resolution - High energy consumption

Security properties: - Sybil resistance through real-world resource commitment - Historical security track record - Clear incentive alignment - Objective leader selection

18.5.2 Proof of Stake (PoS)

Modern networks like Ethereum use proof of stake, where validators must lock up (stake) tokens to participate in consensus.

Key features: - Energy efficient - Economic security model - Complex slashing conditions - Validator selection algorithms

Implementation challenges: - Nothing-at-stake problem - Long-range attack considerations - Complex incentive design - Stake concentration risks

18.5.3 Practical Byzantine Fault Tolerance (PBFT)

Some networks use PBFT-style consensus where a known set of validators cooperate to agree on transaction ordering.

Advantages: - High performance - Quick finality - Low resource requirements - Suitable for permissioned networks

Limitations: - Less decentralized - Higher communication overhead - Validator set management - Scale limitations

18.5.4 Hybrid Mechanisms

Many modern networks combine different consensus approaches:

Variations include: - PoS with PBFT finality - PoW for validator selection - Layered consensus protocols - Federation-based systems

18.5.5 Advanced Consensus Innovations

The field continues to evolve with new approaches:

Recent developments: - Proof of History for better ordering - Avalanche consensus protocols - Zero-knowledge consensus - Federated consensus models

18.6 Network Architecture

18.6.1 Sovereign Networks

Independent blockchain networks that maintain their own security and consensus. Examples include Bitcoin and Ethereum.

Characteristics: - Complete control over protocol - Independent security model - Native asset issuance - Full sovereignty over rules

18.6.2 Settlement-Dependent Networks

Networks that rely on other blockchains (usually Ethereum) for ultimate security and settlement.

Types: - Optimistic rollups - Zero-knowledge rollups - State channels - Plasma chains

18.6.3 Application-Specific Networks

Blockchains designed for particular use cases: - High-speed trading - Privacy-focused transactions - Specialized applications - Industry-specific solutions

18.7 Technical Innovations

18.7.1 Scaling Solutions

As networks grow, various scaling solutions have emerged:

Layer 2: - State channels - Rollups - Sidechains - Plasma

Sharding: - Data sharding - State sharding - Transaction sharding - Network sharding

18.7.2 Privacy Enhancements

Modern blockchains incorporate various privacy technologies:

Cryptographic tools: - Zero-knowledge proofs - Ring signatures - Confidential transactions - Homomorphic encryption

Protocol-level privacy: - Network-level privacy - Transaction privacy - State privacy - Computational privacy

18.7.3 Cross-Chain Integration

Technologies enabling different blockchains to interact:

Bridge types: - Trusted bridges - Trustless bridges - Optimistic bridges - Zero-knowledge bridges

Interoperability protocols: - Atomic swaps - Cross-chain messaging - Asset wrapping - State verification

18.8 Future Directions

The blockchain field continues to evolve rapidly:

Research areas: - Post-quantum cryptography - Formal verification - Zero-knowledge scaling - Privacy-preserving computation

Emerging applications: - Decentralized AI - Real-world asset tokenization - Identity systems - Institutional adoption

18.9 Conclusion

Blockchain technology represents a fundamental shift in how we can coordinate and verify digital activity. Its continued evolution enables new forms of human coordination and value exchange that were previously impossible. Understanding the technical foundations, from application models to consensus mechanisms, is crucial for building and working with these systems effectively.

19 Modular Blockchain Architecture

19.1 Introduction

Traditional "monolithic" blockchains attempt to handle all functions - execution, settlement, consensus, and data availability - within a single system. While this approach works, it creates inherent scalability limitations as each node must process everything. Modular blockchain architecture takes a different approach by separating these functions into specialized layers, allowing each to scale independently.

19.2 Understanding Blockchain Functions

Before diving into modular designs, let's understand the core functions that any blockchain system must provide:

19.2.1 Execution

The computation of state transitions - processing transactions, running smart contracts, and updating account balances. This requires significant computational resources and historically has been the main bottleneck for scaling.

19.2.2 Data Availability (DA)

The guarantee that the data needed to verify state transitions is publicly available. Without this guarantee, users can't validate the chain's state or detect fraud. This function requires significant storage resources and network bandwidth.

19.2.3 Settlement

The final determination of state transitions, including dispute resolution and finality guarantees. This layer provides the ultimate security and must be highly decentralized and resistant to manipulation.

19.2.4 Consensus

The mechanism by which nodes agree on the ordering of transactions and the current state. This must be both secure and efficient while maintaining decentralization.

19.3 The Data Availability Layer

19.3.1 Celestia's Approach

Celestia introduced a groundbreaking approach by creating a blockchain focused solely on data availability. Its innovation lies in how it enables secure sampling of block data.

Key features:

- Data availability sampling (DAS)
- 2D Reed-Solomon encoding
- Light client optimizations
- Namespace-based organization

The core innovation is that clients can verify data availability without downloading entire blocks:

```
Traditional verification: Download entire block (O(n) data)
Celestia verification: Sample few random bytes (O(log n) data)
```

19.3.2 Ethereum Data Availability

Ethereum's approach to data availability has evolved with the introduction of Proto-Danksharding and blob transactions:

Key characteristics:

- Dedicated blob space in blocks
- Separate fee market for blob data
- KZG commitments for efficient verification
- Time-limited data availability

The blob transaction format:

```
struct BlobTransaction {
    // Regular transaction fields
    address from;
    address to;
    uint256 value;

    // Blob-specific fields
    bytes32[] commitments;    // KZG commitments
    bytes[] blobs;            // The actual data
    uint256 blobGasPrice;     // Separate fee for blob space
}
```

19.4 The Execution Layer

19.4.1 Optimistic Rollups

Optimistic rollups execute transactions off-chain but post transaction data and state commitments to the settlement layer. They rely on fraud proofs to ensure correctness.

Key components:

- Sequencer for transaction ordering
- State commitment scheme
- Fraud proof system
- Challenge period

Example of an optimistic transaction submission:

```
contract OptimisticRollup {
    struct StateUpdate {
        bytes32 stateRoot;
        bytes32 parentStateRoot;
        bytes transactions;
        uint256 timestamp;
    }

    function submitStateUpdate(StateUpdate calldata update) external {
        require(verifyStateTransition(update), "Invalid state transition");
        startChallengeWindow(update);
    }
}
```

19.4.2 ZK Rollups

ZK rollups use zero-knowledge proofs to validate state transitions, providing faster finality and stronger security guarantees than optimistic solutions.

Core elements:

- SNARK/STARK proof generation
- Efficient state encoding
- Proof verification circuit
- On-chain verification

19.5 The Settlement Layer

19.5.1 Ethereum as Settlement

Ethereum has emerged as the primary settlement layer for most scaling solutions. Its role includes:

- Storing state commitments
- Verifying proofs
- Resolving disputes
- Managing asset bridges

Key settlement functions:

```
interface ISettlement {
    // For optimistic rollups
    function challengeStateTransition(
        bytes32 stateRoot,
        bytes proof
    ) external;

    // For ZK rollups
    function verifyStateTransition(
        bytes32 newState,
        bytes proof
    ) external returns (bool);
}
```

19.5.2 Cross-Layer Communication

Communication between layers requires careful design:

- Message passing protocols
- State verification schemes
- Asset transfer bridges
- Finality guarantees

19.6 Practical Implementations

19.6.1 Layer 2 Scaling Solutions

Current production systems demonstrate different approaches to modular architecture:

Optimistic solutions:

- Arbitrum: Advanced sequencing
- Optimism: EVM equivalence
- Base: Modified Optimism stack

ZK solutions:

- zkSync: Custom VM
- StarkNet: Cairo VM
- Polygon zkEVM: EVM compatibility

19.6.2 Data Availability Solutions

Multiple approaches to DA are emerging:

1. Celestia

- Dedicated DA chain
- Namespaced data organization
- Efficient sampling

2. EigenDA

- Distributed DA network
- Economic security model
- Restaking mechanics

3. Ethereum blobs

- Native DA solution
- Time-bounded availability
- Separate fee market

19.7 Future Developments

19.7.1 Cross-Domain MEV

Modular architectures create new considerations for MEV:

- Cross-layer arbitrage
- Sequencer extractable value
- DA fee markets

19.7.2 Interoperability

Standardization efforts are emerging:

- Cross-layer messaging protocols
- Unified bridge standards
- Shared security models

19.8 Conclusion

Modular blockchain architecture represents a fundamental shift in how we scale blockchain systems. By separating key functions into specialized layers, we can achieve greater scalability while maintaining security. The field continues to evolve rapidly, with new solutions and optimizations emerging regularly.

The success of modular designs will likely determine the future of blockchain scalability, as they enable each layer to evolve and optimize independently while maintaining interoperability through standardized interfaces.

20 Smart Contract Languages

20.1 Introduction

Smart contract languages face unique challenges in blockchain development. They must provide robust safety guarantees while enabling developers to write immutable code that often handles millions of dollars in value. Each language makes different tradeoffs between expressiveness, safety, and performance.

20.2 Solidity

As the pioneer of smart contract development, Solidity has shaped how we think about blockchain programming. Its design combines familiar object-oriented concepts with blockchain-specific safety features.

Key characteristics:

- Static typing system
- Contract-oriented programming
- Rich ecosystem of tools
- Extensive security features

Example of modern Solidity patterns:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.19;

contract ModernContract {
    // Events for off-chain tracking
    event ValueUpdated(address indexed updater, uint256 newValue);

    // Immutable variables for gas optimization
    address immutable owner;

    // Custom errors instead of strings
    error UnauthorizedAccess(address caller);
```

```
    error InvalidValue(uint256 value);

    constructor() {
        owner = msg.sender;
    }

    // Use custom errors and checks
    function updateValue(uint256 newValue) external {
        if (msg.sender != owner) {
            revert UnauthorizedAccess(msg.sender);
        }
        if (newValue == 0) {
            revert InvalidValue(newValue);
        }

        emit ValueUpdated(msg.sender, newValue);
    }
}
```

20.3 Move

Originally developed for Facebook's Libra (now Diem) project, Move is a Rust-based language that introduces powerful new concepts for digital asset management. Its adoption by Aptos and Sui has led to distinct dialects of the language.

20.3.1 Core Move Concepts

Move's fundamental innovation is its resource types, which ensure digital assets can't be copied or accidentally destroyed:

```
module example::basic_token {
    struct Token has key {
        value: u64,
    }

    public fun transfer(token: Token, recipient: address) {
        // Resources must be moved explicitly
        move_to<Token>(recipient, token);
    }
}
```

20.3.2 Aptos vs Sui Move

While both platforms use Move, they implement it differently:

Aptos Move:

- Global storage model
- Sequential transaction execution
- Account-based resources
- More traditional blockchain model

```
// Aptos Move example
module example::counter {
    struct Counter has key {
        value: u64,
    }

    public fun increment(account: &signer) acquires Counter {
        let counter = borrow_global_mut<Counter>(signer::address_of(account));
        counter.value = counter.value + 1;
    }
}
```

Sui Move:

- Object-centric model
- Parallel transaction execution
- Object-based ownership
- Novel consensus approach

```
// Sui Move example
module example::counter {
    struct Counter has key {
        id: UID,
        value: u64,
    }

    public fun increment(counter: &mut Counter) {
        counter.value = counter.value + 1;
    }
}
```

20.4 CosmWasm

CosmWasm brings WebAssembly-based smart contracts to Cosmos-based blockchains. Written in Rust, it provides a robust environment for cross-chain contract development.

Key features:

- Rust-based development
- Cross-chain compatibility
- Strong typing system
- IBC integration

Example CosmWasm contract:

```
#[derive(Serialize, Deserialize, Clone, Debug, PartialEq)]
pub struct InstantiateMsg {
    pub count: i32,
}

#[entry_point]
pub fn instantiate(
    deps: DepsMut,
    _env: Env,
    info: MessageInfo,
    msg: InstantiateMsg,
) -> Result<Response, ContractError> {
    let state = State {
        count: msg.count,
        owner: info.sender.clone(),
    };
    set_contract_version(deps.storage, CONTRACT_NAME, CONTRACT_VERSION)?;
    STATE.save(deps.storage, &state)?;

    Ok(Response::new()
        .add_attribute("method", "instantiate")
        .add_attribute("owner", info.sender)
        .add_attribute("count", msg.count.to_string()))
}
```

20.5 Solana Programs

Solana's smart contracts (called "programs") are typically written in Rust, offering high performance but requiring careful management of accounts and data.

Key characteristics:

- Account-based model
- Explicit data ownership
- High performance
- Complex account management

Example Solana program:

```
use solana_program::{
    account_info::AccountInfo,
    entrypoint,
    entrypoint::ProgramResult,
    pubkey::Pubkey,
};

entrypoint!(process_instruction);

pub fn process_instruction(
    program_id: &Pubkey,
    accounts: &[AccountInfo],
    instruction_data: &[u8],
) -> ProgramResult {
    // Solana programs explicitly manage accounts
    let accounts_iter = &mut accounts.iter();
    let account = next_account_info(accounts_iter)?;

    // Program logic here

    Ok(())
}
```

20.6 Bitcoin Script

While limited compared to modern smart contract platforms, Bitcoin Script was the first blockchain programming language. It uses a stack-based approach for transaction validation.

Key features:

- Stack-based execution
- Limited instruction set
- No loops or complex control flow
- Focus on transaction validation

Example Bitcoin Script:

```
OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
```

20.7 Security Considerations

Each language presents unique security challenges:

Solidity:

- Reentrancy attacks
- Access control issues
- Integer overflow (pre-0.8.0)
- Gas optimization risks

Move:

- Resource handling complexity
- Formal verification needs
- Platform-specific quirks
- Module linking security

CosmWasm:

- Cross-chain vulnerabilities
- State management complexity
- Message handling security
- Contract upgradeability

Solana:

- Account validation
- Data race conditions
- Cross-program invasion
- Resource exhaustion

20.8 Future Trends

Smart contract languages continue to evolve:

- Zero-knowledge support
- Formal verification tools
- Cross-chain compatibility
- Advanced type systems

The field remains dynamic, with new languages and features emerging as blockchain technology matures and new use cases develop.

21 Gas

The Fuel of Web3

21.1 Introduction

Imagine trying to mail a package without paying for postage, or running a car without fuel. In Web3, gas serves a similar fundamental purpose - it's the essential resource that powers all blockchain operations. But unlike postage or gasoline, blockchain gas represents something more complex: it's a dynamic pricing mechanism that manages network resources, incentivizes operators, and helps secure the entire system.

This chapter explores gas from multiple perspectives: as a practical tool users must understand, as a technical mechanism that enables network operation, and as an economic system that shapes Web3's evolution.

21.2 Understanding Gas: First Principles

21.2.1 What is Gas?

At its most basic level, gas represents computational effort. Every operation on a blockchain - from simple token transfers to complex smart contract interactions - requires computational resources from the network. Gas measures these resources and assigns them a cost.

Key characteristics of gas include:

- It measures computational complexity
- It's priced dynamically based on network demand
- It's paid in the network's native token
- Failed transactions still consume gas

21.2.2 Why Gas Exists

Gas serves three essential functions:

1. Resource Management
    - Prevents infinite loops and spam attacks
    - Allocates network capacity fairly
    - Creates predictable operational costs
2. Economic Security
    - Compensates network operators
    - Makes attacks economically expensive
    - Aligns incentives across participants
3. Priority Mechanism
    - Determines transaction ordering
    - Manages network congestion
    - Enables price discovery for blockspace

21.3 Gas Mechanics

21.3.1 Basic Components

Every gas transaction involves several components:

1. Gas Limit
    - Maximum computational units allowed
    - Set by the user
    - Must be sufficient for operation
    - Excess is refunded
2. Gas Price
    - Cost per unit of gas
    - Determined by network demand
    - Usually measured in small denominations (e.g., Gwei)
    - Can change rapidly
3. Total Cost
    - Gas Limit $\times$ Gas Price

- Paid upfront
- Maximum possible cost
- Actual cost may be lower

21.3.2 Network-Specific Implementations

Different networks handle gas in distinct ways:

1. Ethereum

    - Base fee + priority fee model
    - EIP-1559 burning mechanism
    - Complex gas calculations for different operations
    - Block gas limits

2. Layer 2 Networks

    - Usually cheaper than Layer 1
    - May have different gas tokens
    - Often bundle L1 and L2 costs
    - Can have unique gas mechanics

3. Alternative Networks

    - May use different resource metrics
    - Often optimize for specific use cases
    - Can have fixed or variable costs
    - Might separate different resource types

21.4 User's Guide to Gas

21.4.1 Practical Gas Management

1. Setting Gas Limits

    - Understanding operation costs
    - Adding safety margins
    - Avoiding out-of-gas errors
    - Estimating complex transactions

2. Choosing Gas Prices

    - Reading gas price oracles

- Understanding urgency tradeoffs
- Timing transactions
- Using gas price alerts

3. Common Pitfalls

    - Insufficient gas limits
    - Overpaying during congestion
    - Failed transaction costs
    - Network-specific quirks

21.4.2 Advanced Gas Strategies

1. Gas Optimization

    - Batching transactions
    - Using gas tokens
    - Timing non-urgent transactions
    - Contract interaction efficiency

2. Cross-Network Considerations

    - Bridge gas costs
    - Network selection
    - Cost comparison tools
    - Gas token economics

21.5 Economic Implications

21.5.1 Fee Markets

Gas creates a market for blockspace with unique characteristics:

1. Supply Mechanics

    - Fixed block space
    - Regular block intervals
    - Network-specific limits
    - Upgrade considerations

2. Demand Factors

    - User activity levels

- Market conditions
- Bot competition
- MEV opportunities

21.5.2 Market Impact

Gas mechanics influence broader market behavior:

1. Layer 2 Adoption

    - Cost comparison driving usage
    - Network effects
    - Migration patterns
    - Protocol competition

2. Protocol Design

    - Gas optimization requirements
    - Economic model constraints
    - User experience trade-offs
    - Scaling solutions

21.6 Future of Gas

21.6.1 Evolving Models

Gas systems continue to develop:

1. Technical Innovations

    - Account abstraction
    - Meta-transactions
    - Gas-less transactions
    - Resource-specific pricing

2. Economic Experiments

    - Alternative fee mechanisms
    - Novel burning models
    - Hybrid systems
    - Cross-chain standardization

21.6.2 Implications for Users

As gas systems evolve, users should:

- Stay informed about changes
- Adapt strategies accordingly
- Understand new opportunities
- Manage changing risks

21.7 Key Takeaways

1. Gas is fundamental to Web3:
    - Essential for network operation
    - Drives economic security
    - Shapes user behavior
2. Understanding gas is crucial for:
    - Effective network usage
    - Cost management
    - Strategy development
    - Risk assessment
3. Gas systems are evolving:
    - New models emerging
    - Greater efficiency possible
    - More complexity likely
    - Continued innovation certain

21.8 Practical Exercises

To reinforce your understanding:

1. Calculate total gas costs for different operations
2. Compare gas prices across networks
3. Optimize a multi-step transaction
4. Analyze historical gas patterns

21.9 Further Reading

- Gas optimization guides
- Network-specific documentation
- Economic analysis papers
- Technical proposals

22 Decentralized Applications (dApps)

22.1 Introduction

Decentralized applications (dApps) represent the practical application of blockchain technology beyond simple value transfer. Unlike traditional applications where logic and data reside on centralized servers, dApps execute their core logic through smart contracts while interfacing with users through traditional web technologies.

22.2 Core Categories

22.2.1 Decentralized Finance (DeFi)

DeFi applications form the largest and most sophisticated category of dApps, handling billions of dollars in value through various financial primitives.

22.2.1.1 Decentralized Exchanges (DEXs)

DEXs enable trustless trading of digital assets through different market-making mechanisms:

Automated Market Makers (AMMs):

- Uniswap: Pioneered the constant product formula
- Curve: Optimized for stable asset swaps
- Balancer: Enables multiple asset pools
- TraderJoe: Adapted AMM model for Avalanche

Order Book DEXs:

- Serum: On-chain order book on Solana
- dYdX: Layer 2 perpetual trading
- IDEX: Hybrid order book model

22.2.1.2 Lending Protocols

Lending platforms enable collateralized borrowing and lending of digital assets:

- Aave: Multi-token lending with variable/stable rates
- Compound: Pioneered the c-token model
- Euler: Risk-based lending markets
- Morpho: Peer-to-pool-to-peer lending

22.2.1.3 Derivatives

Derivative protocols create synthetic assets and complex financial instruments:

- Synthetix: Synthetic asset platform
- GMX: Decentralized perpetual exchange
- dYdX: Perpetual futures trading
- Opyn: Options protocols

22.2.2 NFT Ecosystems

NFT platforms have evolved from simple marketplaces to complex ecosystems:

22.2.2.1 Marketplaces

- OpenSea: General NFT marketplace
- Blur: Pro trading platform
- Magic Eden: Solana NFT marketplace

22.2.2.2 NFT Finance

- JPEG'd: NFT-collateralized lending
- BendDAO: NFT liquidity protocol
- NFTfi: Peer-to-peer NFT loans

22.2.3 Gaming

Blockchain gaming applications attempt to create sustainable play-to-earn economies:

- Axie Infinity: Pioneer of play-to-earn
- StepN: Move-to-earn application
- Illuvium: AAA quality blockchain game
- Gods Unchained: Trading card game

22.2.4 Social Platforms

Decentralized social applications aim to create censorship-resistant communication platforms:

- Lens Protocol: Social graph protocol
- Farcaster: Decentralized social network
- Mirror: Web3 publishing platform

22.3 Infrastructure dApps

22.3.1 Identity Solutions

Decentralized identity systems provide the foundation for Web3 interaction:

22.3.1.1 Name Services

- ENS (Ethereum Name Service)
- Lens Handles
- Solana Name Service
- Stacks Name Service

22.3.1.2 Authentication

- Sign-in with Ethereum
- Worldcoin
- Civic

22.3.2 Oracle Networks

Oracles provide external data to smart contracts:

- Chainlink: Market leader in oracle services
- Pyth: High-performance oracle network
- UMA: Optimistic oracle system

22.3.3 Privacy Solutions

Privacy-preserving protocols enable confidential transactions:

22.3.3.1 Mixing Services

- Tornado Cash: Fixed-denomination mixer
- Privacy Pools: Compliant privacy solution

22.3.3.2 Zero-Knowledge Protocols

- Railgun: Private DeFi interactions
- Aztec: Private smart contract platform

22.4 Architecture Patterns

Modern dApps typically implement several key architectural patterns:

22.4.1 Account Abstraction

Social recovery wallets and gasless transactions:

- Safe (formerly Gnosis Safe)
- Biconomy
- Stackup

22.4.2 Composability

Building on existing protocols:

- Yearn Finance: Yield aggregation
- Convex: Curve gauge optimization
- Frax: Algorithmic-collateral model

22.4.3 Cross-Chain Integration

Enabling multi-chain functionality:

- LayerZero
- Stargate
- Axelar
- Wormhole

22.5 Development Frameworks

Popular frameworks for dApp development:

22.5.1 Frontend

- Web3.js
- Ethers.js
- wagmi
- RainbowKit

22.5.2 Smart Contract Development

- Hardhat
- Foundry
- Truffle
- Anchor (Solana)

22.6 Security Considerations

Common security patterns and vulnerabilities:

22.6.1 Access Control

- Multi-signature requirements
- Time locks
- Role-based permissions

22.6.2 Economic Security

- Oracle manipulation
- Flash loan attacks
- Infinite mint exploits

22.6.3 Technical Security

- Reentrancy guards
- Integer overflow protection
- proper state management

22.7 Future Trends

Emerging patterns in dApp development:

22.7.1 Modular Design

- Separating execution, settlement, and data availability
- Protocol-level specialization
- Cross-chain optimization

22.7.2 Privacy Integration

- Zero-knowledge proofs
- Homomorphic encryption
- Private state management

22.7.3 Real-World Assets

- Tokenized securities
- Real estate platforms
- Carbon credit markets

22.8 Conclusion

Decentralized applications represent the frontier of blockchain innovation, constantly evolving with new patterns and capabilities. Success in this space requires understanding both traditional application development and the unique constraints and opportunities of blockchain platforms.

23 Maximal Extractable Value (MEV)

23.1 Introduction

Maximal Extractable Value (MEV) represents one of the most fascinating and complex phenomena in blockchain systems. Originally called "Miner Extractable Value," the term has evolved to "Maximal" as extraction techniques have expanded beyond just miners. At its core, MEV emerges from the ability to reorder, insert, or censor transactions within a block for profit.

Think of a blockchain as a series of auctions where the auctioneer (block producer) can not only decide which bids to accept but also their order. This power creates opportunities for profit that don't exist in traditional markets.

23.2 Understanding MEV

23.2.1 The Mechanics of MEV

When users submit transactions to a blockchain network, they enter a temporary holding area called the mempool. Block producers can then:

- Choose which transactions to include
- Determine the order of transactions
- Insert their own transactions
- Create custom transactions in response to user activity

This control creates several profit opportunities:

1. Arbitrage
    - Spotting price differences across DEXs
    - Exploiting temporary market inefficiencies
    - Capturing price movements from large trades
2. Liquidations
    - Competing to liquidate undercollateralized positions

- Racing to claim liquidation rewards
- Front-running other liquidators

3. Sandwich Attacks

   - Placing trades before and after a large swap
   - Exploiting price impacts for profit
   - Extracting value from user slippage tolerance

23.3 MEV Extract Methods

23.3.1 Searchers and Builders

The MEV ecosystem has evolved into specialized roles:

Searchers:

- Develop algorithms to identify MEV opportunities
- Create optimal transaction bundles
- Bid for block space through builders

Builders:

- Aggregate transaction bundles from searchers
- Construct optimal blocks
- Compete to have their blocks chosen by validators

23.3.2 MEV-Boost

MEV-Boost represents a crucial development in MEV extraction, separating block building from block validation:

```
User Transactions → Searchers → Builders → Relays → Validators
```

This separation aims to:

- Increase competition among builders
- Improve block construction efficiency
- Distribute MEV more evenly

23.4 Proposer Builder Separation (PBS)

PBS represents the next evolution in MEV management, aiming to institutionalize the separation of block building and proposal.

23.4.1 Core Concepts

PBS divides block production into three distinct roles:

1. Proposers (Validators)
    - Select the most valuable block
    - Earn fees from block proposals
    - Don't need powerful hardware
2. Builders
    - Construct optimal blocks
    - Compete through bid prices
    - Require specialized infrastructure
3. Relays
    - Connect builders and proposers
    - Verify block validity
    - Ensure fair competition

23.4.2 Technical Implementation

PBS requires several key components:

1. Builder API

```
interface IBuilder {
    function submitBundle(
        bytes[] calldata transactions,
        uint256 bid
    ) external returns (bytes32 bundleHash);
}
```

2. Relay Protocol

```
interface IRelay {
    function submitBlock(
        bytes[] calldata transactions,
        bytes32 parentHash,
        uint256 bid
    ) external returns (bool success);
}
```

23.5 Multiple Concurrent Leaders (MCL)

MCL represents a novel approach to MEV that allows multiple validators to propose blocks simultaneously.

23.5.1 Design Goals

MCL aims to:

- Reduce MEV extraction opportunities
- Increase network throughput
- Improve censorship resistance

23.5.2 Technical Challenges

Implementing MCL requires solving several complex problems:

1. Leader Selection
    - Determining concurrent proposers
    - Managing overlapping proposals
    - Resolving conflicts
2. Block Merging
    - Combining parallel proposals
    - Handling transaction conflicts
    - Ensuring deterministic outcomes

23.6 MEV Protection Strategies

23.6.1 For Users

Users can protect themselves from MEV through several strategies:

1. Commitment Schemes

```
// Example of a commit-reveal scheme
contract CommitReveal {
    mapping(bytes32 => bool) public commits;

    function commit(bytes32 commitHash) external {
        commits[commitHash] = true;
    }

    function reveal(bytes32 secret, uint256 value) external {
        require(commits[keccak256(abi.encodePacked(secret, value))]);
        // Execute trade with committed value
    }
}
```

2. Intent-Based Trading
    - Expressing desired outcomes rather than specific paths
    - Using specialized intent protocols
    - Allowing builders to optimize execution

23.6.2 For Protocols

Protocols can implement MEV-resistant designs:

1. Batch Auctions
    - Aggregating trades into discrete batches
    - Using uniform clearing prices
    - Preventing front-running
2. Time-Weighted Average Prices (TWAP)
    - Spreading execution over time
    - Reducing manipulation opportunities
    - Improving price stability

23.7 Future of MEV

23.7.1 Emerging Solutions

Several promising approaches are being developed:

1. Zero-Knowledge MEV
    - Privacy-preserving order flow
    - Encrypted mempool designs
    - Confidential transaction ordering
2. Fair Ordering Services
    - Decentralized sequencing
    - Verifiable delay functions
    - Random beacon systems

23.7.2 Regulatory Considerations

The MEV landscape faces increasing scrutiny:

1. Market Manipulation
    - Sandwich attack legality
    - Front-running regulations
    - Fair market requirements
2. User Protection
    - Disclosure requirements
    - Best execution standards
    - Consumer protection rules

23.8 Conclusion

MEV represents a fundamental challenge in blockchain design, sitting at the intersection of mechanism design, game theory, and market structure. While it cannot be eliminated entirely, continued innovation in areas like PBS and MCL promises to make MEV extraction more efficient and equitable. Understanding MEV is crucial for anyone building or participating in blockchain systems, as it affects everything from protocol design to trading strategies.

Part VI

Financial

24 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.

24.1 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.

24.1.1 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

24.1.2 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.

24.1.3 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.

24.1.4 Derivatives

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

24.1.4.1 On Chain Derivatives

24.1.4.2 Off Chain Derivatives

24.2 Tokens

24.2.1 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

24.2.1.1 ICS20

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

24.2.2 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.

24.3 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.

24.4 Hodling

Wallets

24.4.1 Custodial

24.4.2 Non-Custodial

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

24.5 Yield Properties

Major categories are:

- Network Yield
- Trading Yield
- Protocol Yield

24.5.1 Trading Yield

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

24.5.2 Network Yield

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

24.5.3 dApp Yield

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

24.5.3.1 Liquidity Provision

We break this down into the following Categories

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

24.5.3.1.1 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

25 Market Structures

Financial markets are complex systems that have evolved over centuries to facilitate the efficient exchange of assets. In this chapter, we'll explore the fundamental structures that make modern markets possible, with a particular focus on how traditional market mechanisms have influenced and been adapted by cryptocurrency markets.

25.1 Understanding Market Structure Basics

At its core, a market is where buyers and sellers come together to trade. But the way this meeting happens has profound effects on how prices are discovered, how fairly participants are treated, and how efficiently trades are executed. Let's examine the key concepts that underpin all market structures.

25.1.1 Price Discovery

Price discovery is the process by which markets determine the true price of an asset. Think of it as the market's way of collectively agreeing on what something is worth. This process is influenced by:

- The flow of new information
- The number of market participants
- The transparency of trading activity
- The rules and mechanisms of the trading venue

For example, when a company announces unexpected positive earnings, traders rushing to buy the stock help discover its new, higher price. In crypto markets, this process happens 24/7, with global participation leading to near-instant price adjustments as new information emerges.

25.1.2 Liquidity

Liquidity is the ease with which an asset can be bought or sold without causing a significant price movement. It's like the depth of a swimming pool - the deeper it is, the less splash you make when jumping in. High liquidity is characterized by:

- Tight bid-ask spreads
- Large order book depth
- High trading volumes
- Minimal price impact from trades

25.2 Central Limit Order Books (CLOBs)

The Central Limit Order Book (CLOB) is perhaps the most important innovation in market structure history. It's essentially a sorted list of all the prices at which market participants are willing to buy (bids) and sell (asks) an asset.

25.2.1 How CLOBs Work

A CLOB operates on simple but powerful principles:

1. **Order Types**
    - Limit Orders: Instructions to buy or sell at a specific price or better
    - Market Orders: Instructions to buy or sell immediately at the best available price
2. **Price-Time Priority**
    - Orders are matched based on price priority (best prices first)
    - When prices are equal, earlier orders get priority
    - This creates a fair "first come, first served" system

Here's a simplified visualization of an order book:

```
     ASKS
Price | Size
105   | 10
104   | 15
103   | 20
-----------------
102   | 25
101   | 30
100   | 35
     BIDS
```

25.2.2 Order Matching Logic

When new orders arrive, they're matched against existing orders following strict rules:

1. Market orders match immediately with the best available price
2. Limit orders join the book if they can't match immediately
3. Partial fills are possible when order sizes don't match exactly

For example, if someone submits a market buy order for 30 units in the above book, they would: - Buy 20 units at 103 - Buy 10 units at 104 - Pay an average price of 103.33

25.3 Evolution to Electronic Markets

The transition from physical trading floors to electronic markets marked a revolutionary change in market structure. Electronic markets brought:

- Faster execution speeds
- Lower transaction costs
- Broader market access
- More sophisticated trading strategies
- Better price transparency

However, they also introduced new challenges like: - Need for robust technology infrastructure - Complex failure modes - High-frequency trading arms race - New forms of market manipulation

25.4 Crypto Market Adaptations

Cryptocurrency markets have taken traditional market structures and adapted them for a decentralized world. This has led to several innovations:

25.4.1 Centralized Exchange Order Books

Crypto exchanges like Binance and Coinbase operate traditional CLOBs but with some key differences: - 24/7 trading - Global access - Multiple quote currencies - Faster settlement - Novel order types

25.4.2 On-Chain Order Books

Attempting to put order books entirely on-chain has revealed interesting challenges: - High gas costs for order placement - Front-running vulnerability - Block time limitations - Settlement finality considerations

25.4.3 Hybrid Solutions

Modern crypto trading often uses hybrid approaches that combine the best of both worlds: - Off-chain order books with on-chain settlement - Layer 2 scaling solutions - State channels for high-frequency trading - Automated Market Makers (AMMs) as complementary liquidity sources

25.5 Market Structure Implications

The choice of market structure has far-reaching implications for:

25.5.1 Trading Strategy

Different market structures favor different trading approaches. CLOBs are ideal for market making and arbitrage, while AMMs excel at providing passive liquidity.

25.5.2 Market Quality

Market structure affects: - Price efficiency - Transaction costs - Market stability - Fair access

25.5.3 Regulatory Compliance

Market structure choices impact: - Regulatory oversight capability - Market manipulation risk - Customer protection measures - Systemic risk management

25.6 Looking Ahead

Market structures continue to evolve as technology advances and new requirements emerge. Future developments may include: - Greater integration between TradFi and DeFi markets - Novel hybrid market structures - Improved privacy solutions - More efficient cross-chain trading mechanisms

26 Key Takeaways

- Market structures fundamentally shape how assets are traded
- CLOBs remain the gold standard for price discovery and fair trading
- Electronic markets have transformed trading but introduced new challenges
- Crypto markets are innovating on traditional structures while maintaining their core principles
- The future likely holds further convergence between traditional and crypto market structures

27 Further Reading

- Flash Boys by Michael Lewis
- Trading and Exchanges by Larry Harris
- "Understanding Market Microstructure" series on the CME website

27.0.1 Market Caps

Let's start with a practical example. Imagine two tokens:

Token A: - Total supply: 1 million tokens - Current price: $1 - Market cap: $1 million - Liquidity in DEX pools: $500,000

Token B: - Total supply: 1 billion tokens - Current price: $0.001 - Market cap: $1 million - Liquidity in DEX pools: $5,000

While both tokens show the same market cap, they tell very different stories. Token A has deep liquidity - you could sell $100,000 worth without crashing the price. Token B might crash 90% if someone tries to sell just $1,000 worth.

27.1 Types of Market Cap

In DeFi, we need to understand several variations:

1. Circulating Market Cap = Current Price × Circulating Supply The value of tokens currently trading in the market
2. Fully Diluted Valuation (FDV) = Current Price × Total Supply The theoretical value if all tokens were in circulation
3. Realized Market Cap = Sum of (Price × Amount) for each token last moved A measure that helps identify actual economic activity

27.2 The Liquidity Ratio

One of the most important metrics rarely discussed is the liquidity ratio: Liquidity Ratio = Total DEX Liquidity / Market Cap

Generally: - Ratio > 0.1: Healthy liquidity - Ratio 0.01-0.1: Exercise caution - Ratio < 0.01: High manipulation risk

27.3 Market Cap Manipulation

Understanding how market cap can be manipulated helps avoid common traps:

1. Supply Manipulation
    - Burning tokens to artificially reduce supply
    - Hidden minting capabilities
    - Lock-up periods that temporarily restrict supply
2. Price Manipulation
    - Wash trading to create fake volume
    - Thin liquidity pools easily moved by small trades
    - Strategic buying to push price before token launches
3. Liquidity Games
    - Temporary liquidity adds before major announcements
    - Cross-chain liquidity that's hard to track
    - Flashloan attacks that distort price momentarily

27.4 Real Value vs. Market Cap

Market cap becomes more meaningful when viewed alongside other metrics:

- Daily Active Users (DAU)
- Revenue or fees generated
- Treasury holdings
- Protocol-owned liquidity
- Cross-chain presence and activity

Think of market cap as just one instrument in an orchestra - it only makes sense when played in harmony with other metrics.

27.5 Red Flags in Market Cap Analysis

Watch for these warning signs: - Market cap growing faster than user adoption - Large gaps between circulating and total supply - Sudden changes in supply without clear reason - Market cap much higher than total value locked (TVL)

27.6 Using Market Cap in Trading Decisions

Market cap can be valuable when used correctly:

1. For relative valuation between similar protocols
2. Identifying potential manipulation
3. Assessing room for growth
4. Understanding total risk exposure

Remember: Market cap is a trailing indicator showing where a token has been, not necessarily where it's going.

27.7 Conclusion

Market capitalization in DeFi requires a more sophisticated understanding than in traditional finance. While it shouldn't be ignored, it should never be the only metric you consider. The most successful traders and investors learn to look beyond market cap to understand true value and risk.

28 Ratings

This section will describe our ratings model for Digital Assets.

28.1 Core Rating Categories (60% of Total Rating)

28.1.1 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

28.1.2 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

28.2 Risk Categories (40% of Total Rating)

28.2.1 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

28.2.2 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

28.2.3 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

28.2.4 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

28.3 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

28.4 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

28.5 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

28.6 Review Framework

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

28.7 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

28.8 Risk Factors

28.8.1 1. Technical Risks

- Implementation bugs
- Security vulnerabilities
- Upgrade coordination failures
- Network splits

28.8.2 2. Governance Risks

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

28.8.3 3. Social Risks

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

28.9 Future Considerations

28.9.1 1. Emerging Trends

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

28.9.2 2. Challenges

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

28.9.3 3. Opportunities

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

28.10 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.

28.11 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

29 Trollip's Index

Trollip's index, taking a cue from the S&P 500, will aim to classify 500 Digital Assets.

The general approach would be to classify the top 500 via marketcap. However, a good characteristic of a degen is to not follow conventional wisdom and groupthink. So as the Almanack grows and more strategies are added to the the Degen Chapters, we'll add the assets as we discuss strategies around them. This means that it will contain some absolute shit coins. This will also help us to refine our Risk models

29.1 BTC

Bitcoin has emerged as the world's first successful digital store of value, creating a new paradigm for wealth preservation in the digital age. Much like gold served ancient civilizations through to modern times as a reliable store of wealth, Bitcoin provides similar characteristics but with distinct advantages native to the digital realm.

At its foundation, Bitcoin's value proposition rests on its absolute scarcity - there will never be more than 21 million bitcoins. This hard cap, combined with a transparent and immutable issuance schedule through mining "halvings" every four years, creates a predictability that even gold, with its uncertain mining output, cannot match. When new gold deposits are discovered or mining technology improves, supply can increase unexpectedly. Bitcoin's supply schedule, in contrast, is mathematically certain.

Bitcoin's digital nature offers significant advantages over traditional stores of value. Unlike gold, it can be transferred instantly across borders, divided into microscopic units, stored without physical vault costs, and verified for authenticity without specialized equipment. Its self-custody properties allow individuals to maintain direct control over their wealth without relying on third-party custodians or financial institutions.

The network's security model, backed by massive computational power through proof-of-work mining, has proven remarkably resilient over more than a decade. This track record has gradually built confidence among institutional investors, who increasingly view Bitcoin as a legitimate asset class for portfolio diversification and inflation hedging. Major financial institutions now offer Bitcoin investment products, while some corporations have adopted it as a treasury reserve asset.

While Bitcoin began as a peer-to-peer electronic cash system, its evolution into a store of value mirrors how gold transformed from a medium of exchange into a wealth preservation tool. The emergence of Layer 2 solutions like the Lightning Network now handles Bitcoin's payments functionality, allowing the base layer to focus on its primary role as digital gold - the foundational layer of monetary security in the cryptocurrency ecosystem.

29.1.1 Derivatives

29.1.1.1 cbBTC

Coinbase Base Bitcoin (cbBTC) represents a novel approach to Bitcoin tokenization, launched by Coinbase to bridge the gap between Bitcoin and Ethereum-based DeFi applications. It functions as an institutional-grade wrapped version of Bitcoin, backed 1:1 by actual Bitcoin held in Coinbase's custody. What makes cbBTC particularly noteworthy is its institutional focus, leveraging Coinbase's reputation as a publicly traded company and its robust custody infrastructure to provide a secure and regulated way to use Bitcoin across different blockchain ecosystems.

The technical architecture of cbBTC operates through a burn-and-mint mechanism on the Base network, Coinbase's layer-2 blockchain built on top of Ethereum. When users deposit Bitcoin into Coinbase's custody, an equivalent amount of cbBTC is minted on Base. This process allows Bitcoin holders to participate in Base's growing DeFi ecosystem while maintaining exposure to Bitcoin's value. The token implements additional security measures, including proof of reserves and regular audits, making it particularly appealing to institutional investors who require high levels of security and regulatory compliance. Unlike some other wrapped Bitcoin tokens, cbBTC's key differentiator is its direct integration with Coinbase's established infrastructure and its focus on institutional-grade security and compliance measures.

29.2 ETH

Ethereum represents a revolutionary leap in computing - it's humanity's first attempt at creating a global, decentralized computer that's always running and accessible to everyone. Like how the internet connected computers worldwide for information sharing, Ethereum connects computers globally to create a single, unified computational platform that no one controls but everyone can use.

Think of Ethereum as a massive, worldwide computer with some unique properties: it never shuts down, can't be censored, and maintains perfect records of everything it processes. Instead of storing photos or documents like a regular computer, this world computer specializes in running smart contracts - pieces of code that automatically execute agreements and handle digital assets without needing intermediaries.

ETH, the network's native asset, serves as the essential "fuel" that powers this world computer. Every computation, whether it's processing a DeFi trade or minting an NFT, requires ETH to run. After Ethereum's transition to Proof of Stake through The Merge, ETH also gained a new role - network validators must stake ETH to participate in securing the network, similar to how a computer needs electricity to function and stay secure.

What makes Ethereum truly revolutionary is its programmability. Just as early personal computers transformed from specialized calculators into general-purpose machines that could run any software, Ethereum evolved blockchain technology from a simple transaction ledger into a platform that can run any programmable application. This has spawned entire new industries: decentralized finance (DeFi) protocols that operate 24/7 without human intervention, NFT marketplaces that enable digital ownership and royalties, and DAOs that coordinate human activity through code rather than hierarchies.

The platform continues to evolve through ambitious technical upgrades. Layer 2 scaling solutions like rollups act as specialized processors that handle heavy computations off the main chain, while the planned implementation of sharding will divide the network into parallel processing units - similar to how modern computers use multiple cores to increase performance. These improvements aim to make the world computer more efficient and accessible while maintaining its core properties of decentralization and security.

As this world computer grows in capability and adoption, ETH's value proposition strengthens - it's not just a digital asset, but the essential resource needed to access and use what might become the foundation of our digital future.

Part VII

Social

30 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.

30.0.0.1 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

30.0.0.2 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

30.0.0.3 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

30.0.1 Governance Mechanisms

30.0.1.1 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

30.0.1.2 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

30.0.1.3 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

30.0.2 Improvement Proposal Systems

30.0.2.1 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

30.0.2.2 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

30.0.3 Centralization Factors

30.0.3.1 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

30.0.3.2 2. Governance Centralization

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

30.0.3.3 3. Infrastructure Centralization

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

30.0.4 Best Practices

30.0.4.1 1. Change Management

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

30.0.4.2 2. Community Engagement

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

30.0.4.3 3. Technical Implementation

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

31 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

31.1 Current requirements

31.1.1 Translations

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

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

31.1.2 Data Dynamism

I'd like an easy way to embed live data in every version publish. So for example if I want to reference the current ETH price, I should be able to do something like {{eth.current_price}} and it will embed the current price during the Quarto render with the latest price.

References

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