Sneaky Blockchain

Business White Paper

For Managers

5/10/2025

For business inquiries or development information:
sneakylabs@proton.me

This Paper

This is the first of two Sneaky white papers. The next will be a more technical document for developers. This is a business paper. It is unfinished. We'll add more content as the project evolves.

Introduction

Sneaky is a high-performance permissioned blockchain. It uses a centralized clock for block ordering. We call it a post-trustless blockchain. Trust comes from mathematics. Sneaky has no cryptocurrency. This eliminates expensive mining protocols.

Sneaky is the first AI-powered blockchain. Users can query their data in plain language through SneakyAI. Future versions will include voice commands.

Sneaky runs on minimal electricity. It uses no more power than a desktop computer.

Business Case

Sneaky serves businesses that need immutable data storage. It removes the complexity and unpredictable costs of traditional blockchains.

Every enterprise needs immutable data storage. Blockchains promise this capability better than any alternative. Sneaky delivers on that promise.

We describe this as post-trustless architecture. In traditional decentralised blockchains, trust is distributed among anonymous participants – ideal for privacy hobbyists, but inefficient and impractical for enterprises.

Think of it like a bank: A bank doesn't hand the keys to its vault to a democracy of depositors; it assigns them to trusted custodians. Sneaky follows the same principle - but replaces human discretion with cryptographic certainty. A single, auditable server – operated

by the enterprise or its delegate – acts as the time authority, stamping and curating blocks.

The result is a ledger that’s immutable, efficient, and under the organisation’s control, without sacrificing verifiability.

Businesses can store any data on Sneaky. This includes transactions, emissions records, and legal documents. Special tools like Carbon Explorer help manage essential data. These tools ensure both compliance and practicality.

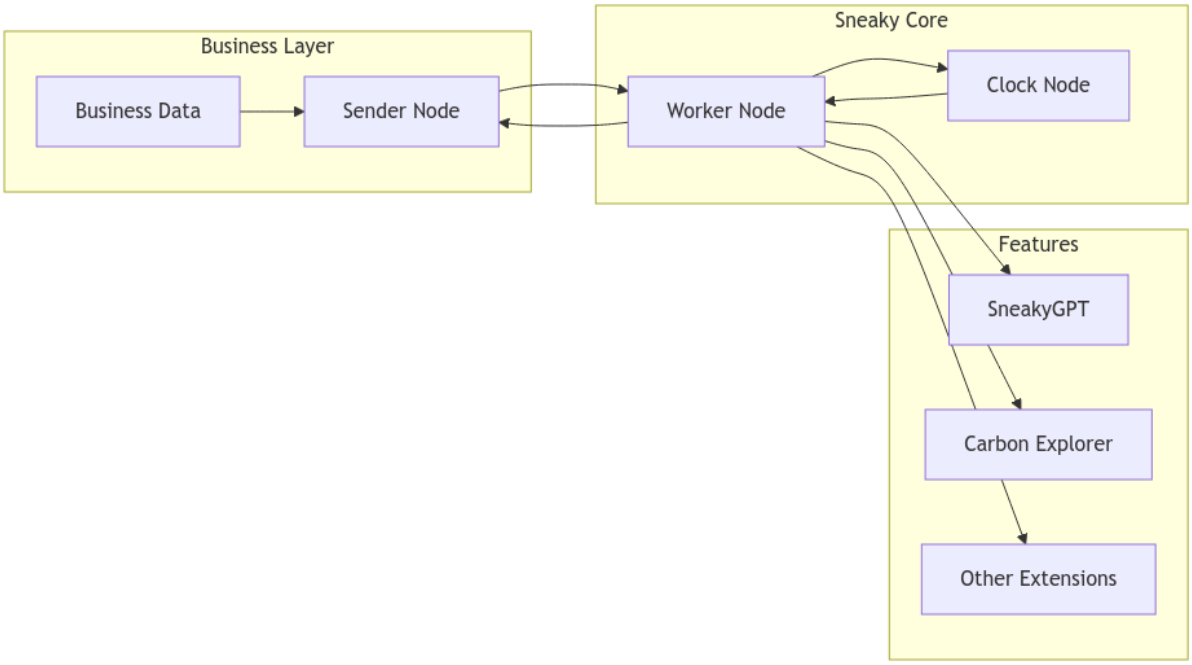

Figure 1: Sneaky's streamlined architecture. Business data flows through Sender Nodes to Worker Nodes, with the Clock Node providing central verification. Extensions like SneakyAI and Carbon Explorer integrate seamlessly with the core system.

The State of Blockchain

Blockchain has evolved beyond cryptocurrency. Today, businesses want blockchains for permanent data storage. They see the value in data permanence and trust. Current solutions often disappoint.

Most enterprise blockchains like Hyperledger are complex. They need expensive infrastructure. They demand specialized expertise. Setup costs can exceed smaller business budgets.

Ethereum 2.0 took a different path. It reduced energy use through Proof of Stake. Costs dropped significantly. Yet Ethereum depends on

cryptocurrency. This ties costs to token prices. Its public nature limits business data privacy.

Other blockchains exist. Most rely on cryptocurrencies. Many lack key business features. Private chains need complex consensus systems. This adds cost and complexity. Public chains face token price volatility.

The business case for blockchain remains strong. Yet implementation often seems impractical. Companies need secure, simple, affordable data storage. They can't accept unpredictable costs or complexity.

Introducing Sneaky

Sneaky represents a new blockchain category. We built it for business data storage. It eliminates mining processes and high energy costs.

Sneaky has no cryptocurrency. This removes unpredictable pricing issues.

A central clock signs and sequences blocks. The clock's public key sits in a trusted, public location. Any node can verify blocks. They first check the clock's signature. Then they verify the creator node's signature. This clock-synchronous approach delivers blockchain security without complexity.

Processing speed cannot overcome implementation time losses. This creates the main blockchain adoption barrier. A Hyperledger setup might take twelve months. Sneaky offers equivalent functionality in days. This simplicity makes blockchain viable for previously impossible projects.

Sneaky suits businesses needing immutable storage. It requires no lengthy development process.

Sneaky Encryption

Sneaky uses standard blockchain encryption methods. Elliptic Curve Cryptography (ECC) forms its security core. ECC provides strong

encryption with small key sizes. This keeps processes fast and lightweight. We offer optional post-quantum encryption for additional security.

Post-quantum encryption resists future quantum computing threats. These threats don't exist yet. Experts don't expect them immediately. But they will arrive. We will provide upgrade paths for existing chains. This combination of ECC and post-quantum options ensures long-term security.

Sneaky includes flexible payload encryption. Payloads store user data. They remain unencrypted by default. Nodes can request end-to-end encryption. This restricts data access to specific nodes. They can also use Asymmetricbc for data escrow.

Asymmetricbc encrypts data using beneficiary public keys. It can add referee over-encryption. This requires referee permission for decryption. Example: A business could encrypt regulatory data. The regulator becomes the beneficiary. A business delegate becomes the referee. Only the regulator can read released data. These features come standard. They need no special coding.

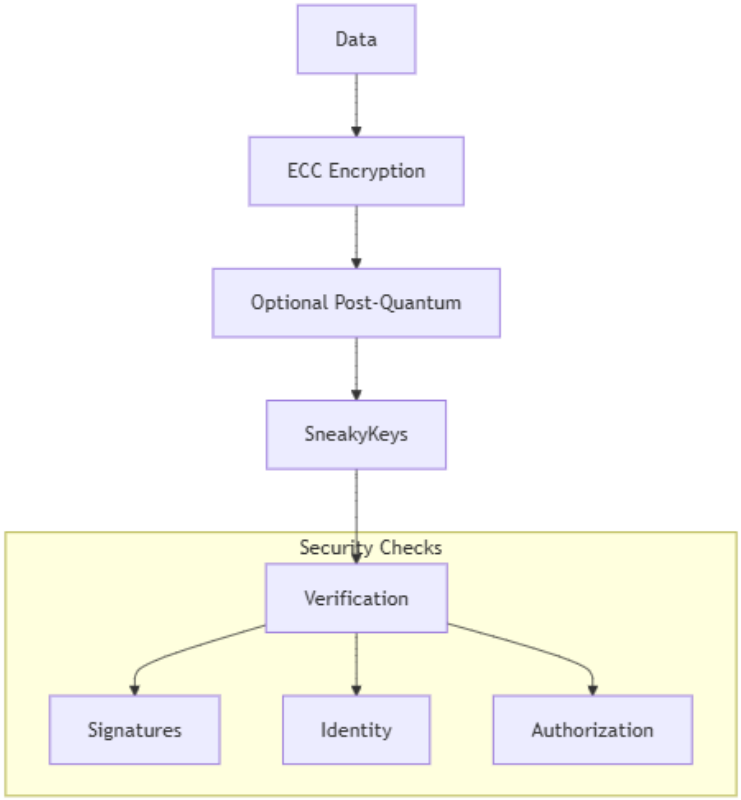

Figure 2: Sneaky's comprehensive security stack. Data passes through multiple protection layers, from basic encryption to optional quantum-resistant security. Asymmetricbc enables controlled data access for regulatory compliance.

SneakyAI

Sneaky leads as the first AI-powered blockchain. Other systems use AI features. Sneaky integrates AI at its core.

SneakyAI enables natural language data queries. Users can generate complex reports without technical knowledge. SneakyAI understands Sneaky's architecture. It receives continuous training on Sneaky data.

Future updates will add AI security features. These include automated node monitoring. They will provide advanced risk response protocols. Voice interaction will arrive soon. Users will talk directly to their blockchain.

Sneaky makes blockchain both secure and accessible.

Sneaky Architecture

Sneaky operates through networked nodes. Any .NET computer can serve as a node. Nodes communicate through asynchronous microservices. Each node type serves specific functions. Together they process transactions and manage operations.

Clock Node

The clock node anchors Sneaky's architecture. Chain owners host their clock node. SneakyLabs hosts the clock for our Sneaky public blockchain. The clock manages block order and security. It numbers and signs new blocks with its private key. This ensures correct block sequencing.

Traditional blockchains operate asynchronously. They create transactions and blocks without timing coordination. Consensus negotiations then establish order. Sneaky uses clock synchronization instead. Transactions and blocks still form asynchronously. The clock then numbers and synchronizes them. Workers share the signed blocks. This creates efficiency and cost savings.

Worker Nodes

Worker nodes handle transaction processing. Each worker maintains a transaction pool. New transactions wait here for block inclusion. Workers package transactions into blocks at volume thresholds. They send blocks to the clock for signing. Signed blocks go to all listening worker nodes. Returning nodes receive missed blocks automatically. This simple process maintains synchronization without consensus mechanisms.

Worker node setup requires minimal components:

- Sneaky installer
- Clock-generated access token
- .NET-enabled computer

Any .NET device could theoretically run a worker node. Only performance limits prevent phone or appliance deployment. The installer generates device keypairs. This enables end-to-end encryption.

Sender Nodes

Sender nodes initiate transactions. Any device can act as a sender. It needs a worker-generated access token and connectivity. Offline senders store transactions until reconnection. New senders must register their public key. This goes into an identity block for validation. The installer creates device keypairs for encryption.

Senders can optionally store shards. In Sneaky, shards are local blockchains. They contain only blocks with sender-produced transactions.

Access Tokens

Access tokens use GUIDs. The clock or authorized worker nodes generate them. Current distribution happens manually. This mirrors early cryptographic key exchange. Token issuers must maintain security until user delivery. Tokens expire based on clock settings. This ranges from hours to days. We plan expanded controls for future releases.

Workflow and Security Checks

Nodes verify all incoming transactions and blocks. They check signatures against identity blocks. This confirms proper authorization. Failed checks trigger security alerts. These go to the clock node and security administrators.

Security settings offer configuration options. Maximum security pauses the chain after exceptions. Administrator restart becomes necessary. Minimum security simply ignores invalid items. Default

settings pause offending nodes. Paused components continue pooling transactions and blocks.

This combines enterprise and blockchain security effectively.

Clock Redundancy

The clock presents a single point of failure for Sneaky. Our Sneaky public instance uses Azure cloud redundancy. This ensures reliable clock uptime. Azure distributes services across multiple data centres. This maintains high availability during localized outages.

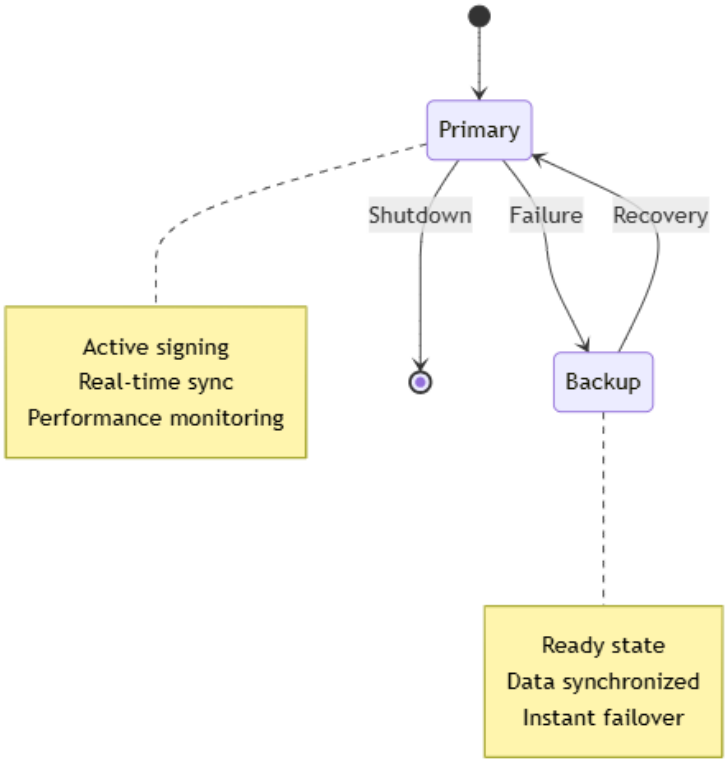

Figure 3: Sneaky's enterprise-grade redundancy system. The primary clock maintains continuous operation while the backup system ensures instant failover if needed. This design eliminates single points of failure.

Enterprise Sneaky deployments should implement similar redundancy. We recommend these best practices:

1. Cloud-Based Redundancy
    - Use multi-region cloud providers
    - Options include AWS, Azure, or Google Cloud
    - Host clock nodes on these platforms

2. Secondary Clock Node
    - Set up a backup clock node
    - Only one clock signs blocks at a time
    - Configure immediate failover capability
3. Key Backup Protocol
    - Store encrypted clock node private keys
    - Maintain offsite backups
    - Limit access to authorized administrators
4. Monitoring System
    - Deploy automated clock node monitoring
    - Configure immediate failure alerts
    - Enable quick administrator response
5. Regular Testing
    - Test failover systems periodically
    - Verify smooth transitions
    - Simulate real conditions
6. Deployment Options
    - Consider on-premises backup systems
    - Evaluate hybrid configurations
    - Match security requirements

c network disruption.

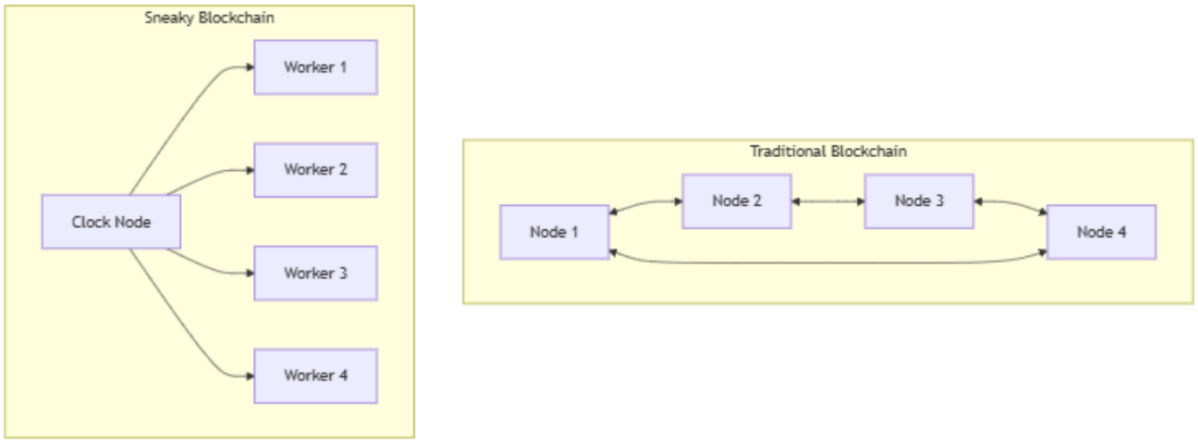

Figure 4: Traditional blockchains (top) require consensus among all nodes, creating complexity and high energy use. Sneaky's approach (bottom) uses a trusted clock node for simpler, more efficient verification.

Sneaky: A Post-Trustless Blockchain

Public blockchains rely on decentralization. They use anonymous consensus for security. This works for cryptocurrency. Businesses have different needs. Many organizations trust internal systems over networks. Sneaky addresses this requirement.

Sneaky anchors trust in the enterprise clock node. This node orders and signs blocks. It creates a verifiable event chain. Businesses gain direct control. They avoid relying on what we call a "democracy of strangers."

Consider bank security. Banks trust employees over customer consensus. Customer majorities might have economic incentives for fraud.

The clock establishes initial trust. Sneaky then becomes trustless. Users rely on mathematics instead of authorities. Internal procedures and cryptographic proofs ensure security. Sneaky matches conventional blockchain security. The trust stays within the system.

Clock setup creates mathematical security. We publish the clock's public key openly. Examples include the SneakyLabs website and New York Times classifieds. Any node can verify blocks. They check the clock's signature. This confirms block authenticity and order.

This approach combines advantages. It provides blockchain security without external voting. The system runs efficiently under full control.

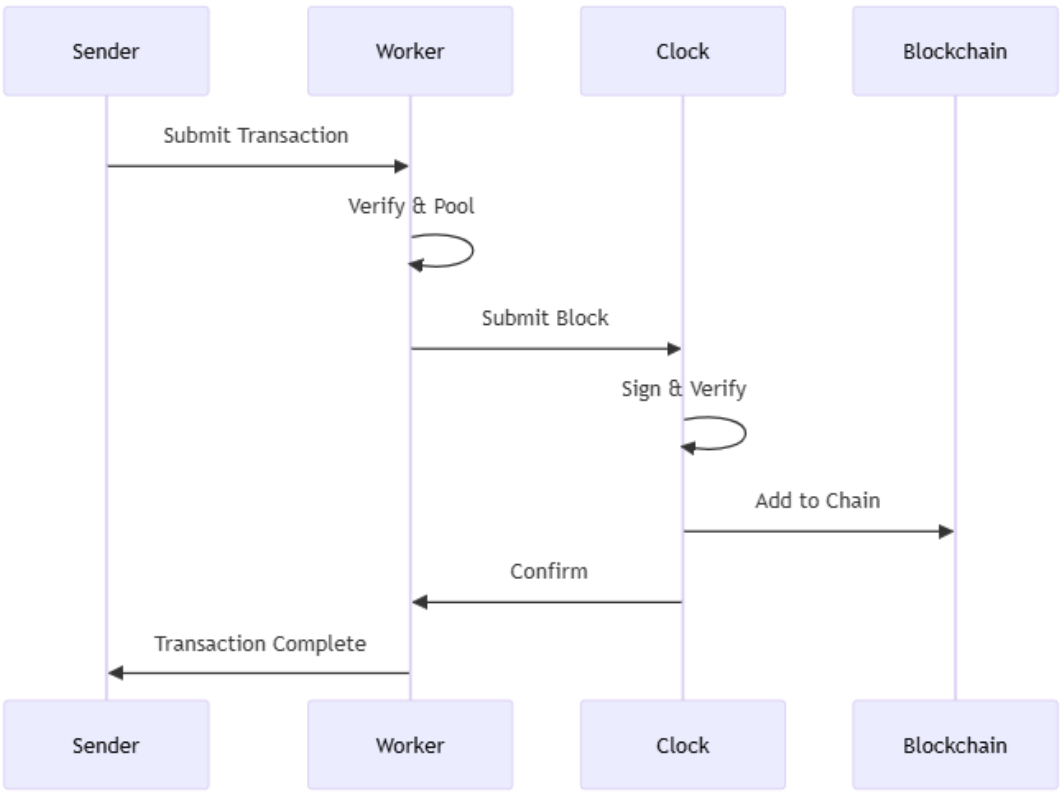

Figure 5: Sneaky's straightforward transaction process. Data moves from Sender through Worker to Clock, with verification at each step. This replaces complex consensus mechanisms with simple, secure verification.

Example Use Cases

Compliance and Regulatory Storage

Sneaky excels at compliance data storage. Its immutable structure makes records permanent. This ensures verifiability and traceability. Clock synchronization provides ordered, signed records. Book tampering becomes nearly impossible. This suits industries with strict compliance needs:

- Finance
- Healthcare
- Government

Environmental Reporting

The Carbon Explorer add-on enables environmental data storage. Companies can record greenhouse gas data on-chain. This demonstrates

environmental commitment. It reduces greenwashing risk. Carbon Explorer shows how developers can build commercial Sneaky extensions.

NFT Creation

Sneaky reduces NFT storage costs. Artists can create NFTs affordably. This cuts transaction fees significantly. Independent creators gain blockchain access. They avoid expensive platforms and speculators.

Green Technology

Traditional blockchains consume massive energy. Bitcoin's electricity use makes headlines monthly. Pre-Merge Ethereum used 78 terawatt-hours yearly. This matched a medium country's consumption. Post-Merge usage equals a medium town. Sneaky reduces this dramatically. It matches a desktop computer's energy use.

Developer Integration

Sneaky requires minimal code integration. Two pathways exist:

1. Call gateway services on worker nodes
2. Install the Sneaky sender library on .NET devices

Both methods use gRPC. This flexibility suits various technical environments. Simple coding requirements enable quick adoption.

Technology Stack

Core Components

- C# and .NET: Microsoft's enterprise development platform
- Blazor: Microsoft's browser-based web framework
- Progressive Web App: Native app experience through browsers
- SQLite Database: Lightweight, file-based storage

Security and Compliance

Sneaky meets strict security standards. Its adaptive encryption protects business data. The system satisfies regulatory requirements.

Key Security Features

- Clock-signed block sequence
- Public key verification
- End-to-end encryption
- ECC with optional quantum resistance
- Configurable security settings
- Real-time monitoring
- Automated alerts

Compliance Features

- Transparent record keeping
- Complete transaction tracing
- Identity block verification
- Keypair triangulation
- Customizable security levels

GDPR Compliance

Sneaky's design supports GDPR requirements. Its controlled registration enables EU-compliant data management.

GDPR Features:

- Geographic node restrictions
- Off-chain personal data storage
- Data minimization options
- Anonymization capabilities
- Selective data removal
- Blockchain integrity preservation

Development Roadmap

Sneaky's future focuses on key improvements:

Security Enhancements

- AI-based monitoring
- Automated risk response
- Stricter node protocols

Performance Upgrades

- Multi-clock support
- Improved block storage
- Faster processing

Developer Tools

- Industry-specific APIs
- Platform integration kits
- Quantum security migration
- New business explorers

Community Growth

- Open-source developer portal
- Certification programs
- Integration resources

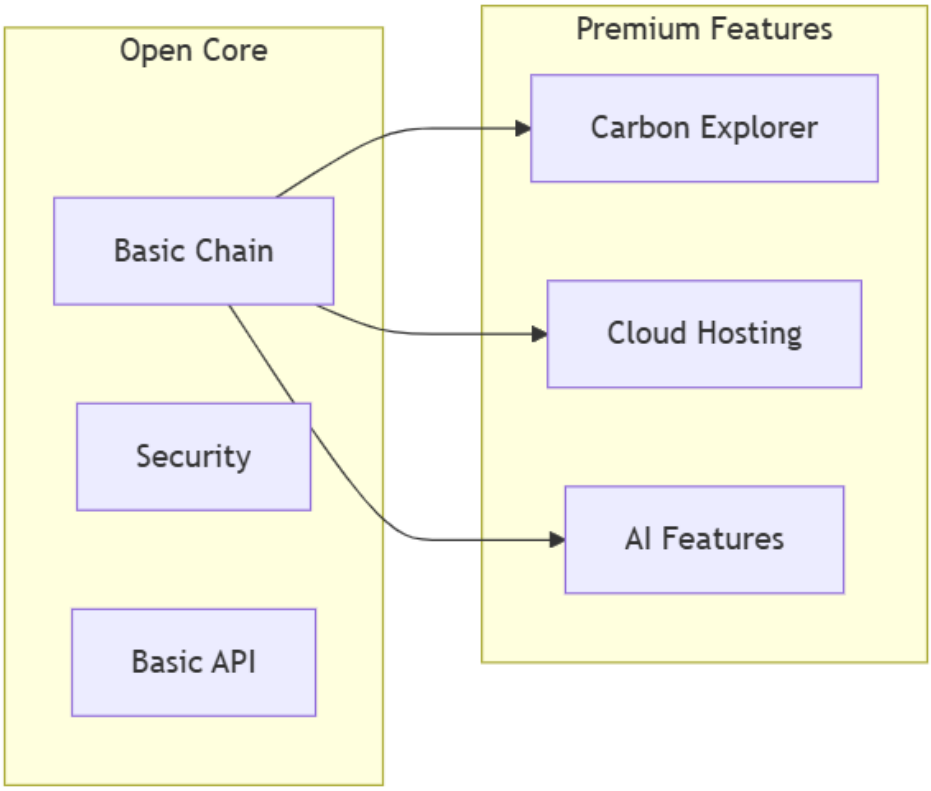

Figure 6: Sneaky's open-core business model. The core blockchain platform is free and open source, with premium features available for specific business needs. This approach enables flexible adoption and scaling.

Business Model

Sneaky uses open-core licensing. The main system remains free and open source. This uses the Apache 2.0 license. Users can review and modify the code.

We offer premium add-ons. Third-party developers can create tools too. Examples include:

- Carbon Explorer for environmental reporting
- Managed cloud hosting
- Custom clock node services

This model provides flexibility. Organizations start with free features. They add premium tools as needed.

Conclusion

Sneaky revolutionizes business data protection. It enables:

- Confident data security
- Regulatory compliance
- Clear audit trails
- Voice-controlled access
- Quick implementation
- Simple node deployment
- Interactive monitoring

Sneaky makes blockchain accessible. Our open-source approach suits any business size. The future of secure data storage starts here.

Contact

For business inquiries or development information:
sneakylabs@proton.me

Bibliography

Bernstein, D. J., Buchmann, J., & Dahmen, E. (2009). *Post-Quantum Cryptography*. Springer.

Berkhout, F., Smith, A., & Stirling, A. (2004). Socio-Technological Regimes and Transition Contexts. In B. Elzen, F. W. Geels, & K. Green (Eds.), *System Innovation and the Transition to Sustainability*. Edward Elgar Publishing.

Bostrom, N. (2014). *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press.

Castro, M., & Liskov, B. (1999). Practical Byzantine Fault Tolerance. *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, 173-186.

Deloitte Insights. (2019). Breaking Blockchain Open: Deloitte's 2019 Global Blockchain Survey. Retrieved from https://www2.deloitte.com/global/en/insights/topics/understanding-blockchain-potential.html

Drescher, D. (2017). *Blockchain Basics: A Non-Technical Introduction in 25 Steps*. Apress.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

IBM Institute for Business Value. (2018). *Building your Blockchain Advantage: How to Start Planning for Blockchain's Business Impact Today*. IBM.

Jiang, P., Wu, J., Chen, J., & Zhao, S. (2019). Blockchain-Based Distributed Energy Trading for Sustainable Development: A Game-Theoretic Approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems, 4*(3), 382-401.

Mougayar, W. (2016). *The Business Blockchain: Promise, Practice, and the Application of the Next Internet Technology*. Wiley.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Retrieved from https://bitcoin.org/bitcoin.pdf

National Institute of Standards and Technology. (2020). Post-Quantum Cryptography: NIST's Plan for the Future. Retrieved from https://csrc.nist.gov/Projects/Post-Quantum-Cryptography

Peters, G. W., Panayi, E., & Chapelle, A. (2015). Trends in Blockchain Technology and Security. In *Handbook of Blockchain, Digital Finance, and Inclusion* (Vol. 1, pp. 241-265). Academic Press.

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

Tapscott, D., & Tapscott, A. (2016). *Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World*. Portfolio.

Von Krogh, G., Spaeth, S., & Lakhani, K. R. (2003). Community, Joining, and Specialization in Open Source Software Innovation: A Case Study. *Research Policy, 32*(7), 1217-1241.