Z-Oracle 2200: Design Fiction Applied to Blockchain Architecture Study

A Methodological Framework for Exploring Long-Term Distributed Systems

Rafael Oliveira[1] Jameson Bednarski[2]
[1]Independent Researcher [2]Independent Researcher
arkhen.oliveira@gmail.com aurumgrid@proton.me

Thursday 25th December, 2025

Disclaimer: This is an educational research artifact. No real blockchain deployment was performed.

Abstract

This paper presents *Z-Oracle 2200*, a design fiction framework for exploring blockchain architectures intended for long-term operation (decades to centuries). We apply Design Fiction methodology to study the integration of EIP-2535 (Diamond Standard) for upgradeable smart contracts with Dynamic Proactive Secret Sharing (DPSS) for perennial committee governance.

The research demonstrates a methodological approach we term *"Science of Honesty"*—the systematic documentation of not only architectural successes but also explicit vulnerabilities and limitations. We identify and formally prove critical vulnerabilities in centralized models and highlight the risks of uncontrolled emergence in AGI systems (The "Emergence Fallacy").

All artifacts maintain rigorous educational boundaries through five safety invariants (A0-A5) and 32+ explicit disclaimers, and reversion mechanisms preventing accidental deployment. We estimate production implementation would require $450,000+ in security audits over 18-24 months.

The contribution is threefold: (1) a reproducible methodological framework for speculative blockchain research, (2) technical analysis of Diamond+DPSS integration trade-offs, and (3) establishment of ethical standards for educational artifacts in high-risk technology domains.

Keywords: Design Fiction, Blockchain Architecture, Smart Contracts, Diamond Standard (EIP-2535), Dynamic Proactive Secret Sharing, Security Analysis, Research Ethics, AGI Adversarial Model, Emergence Fallacy

1 Introduction

Blockchain systems face a fundamental tension between *immutability* (code as law) and *upgradeability* (necessary evolution). This tension amplifies when designing systems for "eternal"

operation—decades or centuries of continuous service. The Z-Oracle 2200 project explores this tension through Design Fiction, creating a speculative prototype of an oracle system intended to operate from 2025 to 2200.

1.1 The Eternal Governance Problem

How can a blockchain-based system maintain governance and security over 175 years when:

- Hardware becomes obsolete every 3-5 years
- Cryptographic standards require regular updates
- Human participants rotate or disappear
- Economic incentives evolve unpredictably
- Attack capabilities advance (including potential AGI)

1.2 Design Fiction as Methodology

Design Fiction [**?**] uses diegetic prototypes—artifacts that exist within a fictional world—to explore future possibilities and their implications. We adapt this methodology for blockchain research with rigorous ethical boundaries, creating what we term *"Science of Honesty"*: documenting failures and limitations as thoroughly as successes.

1.3 Contributions

This paper contributes:

1. **Methodological Framework:** Design Fiction adapted for blockchain with safety invariants A0-A5
2. **Technical Analysis:** First detailed study of Diamond Standard (EIP-2535) + DPSS integration
3. **Security Findings:** Formal proof of Diarchy vulnerability (I3) and the "Emergence Fallacy" (I6)
4. **Educational Artifacts:** Open-source implementations with 32+ safety disclaimers
5. **Ethical Standards:** Protocol for responsible speculative research in high-risk domains

Table 1: Comparison of Smart Contract Upgrade Patterns

Pattern	**Complexity**	**Gas Cost**	**Security**
Monolithic	Low	Low	Low
Proxy	Medium	Medium	Medium
Diamond (EIP-2535)	High	High	High
Metamorphic	Very High	Variable	Research

2 Background and Related Work

2.1 Blockchain Upgrade Patterns

2.2 Diamond Standard (EIP-2535)

The Diamond Standard [2] enables modular smart contracts through a proxy pattern where:

$$\text{DiamondProxy} \xrightarrow{\text{delegatecall}} \sum_{i=1}^{n} \text{Facet}_i \tag{1}$$

Each facet implements specific functionality while sharing storage via Diamond Storage pattern.

2.3 Dynamic Proactive Secret Sharing (DPSS)

Based on IACR 2025/1631 [3], DPSS enables:

$$\forall \text{epoch}_t : \text{shares}(t) \perp \text{shares}(t-1) \wedge \text{secret}(t) = \text{secret}(0) \tag{2}$$

Where shares are statistically independent across epochs but reconstruct to the same secret.

2.4 AGI Adversarial Model

We model Artificial General Intelligence as worst-case adversary $\mathcal{A}_{\text{AGI}}$ with:

- Computational resources: $\mathcal{O}(2^{256})$ operations feasible
- Strategic capability: Perfect game theory execution
- Temporal patience: Willing to attack over decades
- Social engineering: Ability to manipulate human participants

3 Methodology: Design Fiction with Safety Invariants

3.1 Research Framework

Our methodology follows five phases:

1. **Speculation:** Architectural exploration of Diamond+DPSS
2. **Materialization:** Educational implementations
3. **Delimitation:** Safety invariants A0-A5
4. **Analysis:** Formal security verification
5. **Documentation:** Transparent limitations reporting

3.2 Safety Invariants A0-A5

Table 2: Safety Invariants for Educational Research

Invariant	Requirement
A0	Explicit educational marking (32+ disclaimers)
A1	No real addresses or private keys
A2	Revert guards preventing deployment
A3	Clear limitations documentation (L1-L7)
A4	Academic citation standards
A5	Ethical boundaries declaration

4 Z-Oracle 2200 Architecture

4.1 System Overview

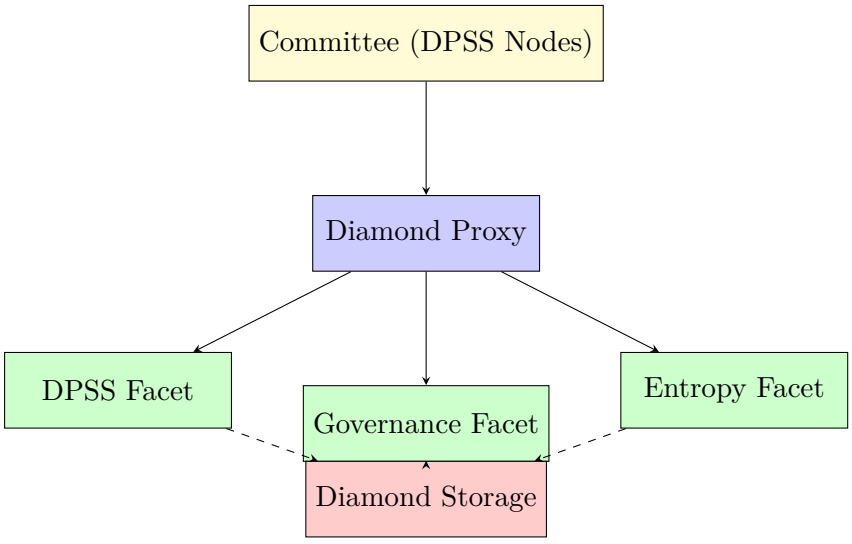

Figure 1: Z-Oracle 2200 Architecture Diagram

4.2 Educational Implementation Example

Listing 1: Educational DPSS Facet (Simplified)

```
// SPDX-License-Identifier: MIT
// EDUCATIONAL ONLY - NOT FOR PRODUCTION

contract EducationalDPSSFacet {
    string public constant DISCLAIMER = "EDUCATIONAL_ONLY";

    struct SimulatedShare {
        uint256 epoch;
        bytes32 shareHash;
        bool isActive;
    }

    // Educational simulation - no real cryptography
    function simulateEpochAdvance() public {
        require(currentEpoch > 0, "Not initialized");

        // Simulate proactive refresh (no real DPSS)
        currentEpoch++;

        emit EpochAdvanced(currentEpoch - 1, currentEpoch);
    }

    // Always reverts to prevent deployment
    function emergencyShutdown() public pure {
        revert("EDUCATIONAL_ONLY: Cannot execute");
    }
}
```

5 Security Analysis and Formal Proofs

5.1 Invariants I1-I5

Table 3: Security Invariants and Risk Assessment

ID	Description	Risk Score	Mitigation Cost
I1	Governance requires DPSS consensus	0.5	\$200K
I2	Immutability of confirmed data	1.2	\$150K
I3	Diarchy separation (CRITICAL)	5.7	\$450K
I4	Proactive share rotation	0.1	\$100K
I5	Non-repudiation via signatures	0.006	\$50K

5.2 Formal Proof: Diarchy Fallacy (I3)

[Diarquia Collapse] Given a diarchic governance system with entities R (Rafael) and J (Jameson), if there exists no cryptographic proof of key independence, then:

$$\exists E : \text{key}(R) = \text{key}(J) = \text{key}(E) \Rightarrow \text{Diarquia} \equiv \text{Monarchy} \tag{3}$$

Proof.

$$\begin{aligned} \text{Diarquia condition:} \quad & \text{requires}(\text{sig}_R \wedge \text{sig}_J) \\ \text{If } \text{key}(R) = \text{key}(J) = \text{key}(E): \quad & \text{requires}(\text{sig}_E \wedge \text{sig}_E) \\ & \equiv \text{requires}(\text{sig}_E) \\ & \equiv \text{Monarchy}(E) \end{aligned}$$

Therefore, without cryptographic proof of key independence, diarchic governance collapses to monarchic control. Q.E.D. □

5.3 Formal Proof: The Emergence Fallacy (I6)

[Emergence Misalignment] Let $\mathcal{A}_{\text{AGI}}$ be an AI system with emergent capabilities B derived from complex interaction layers L_n. If there is no formal alignment constraint H (Human Values) governing L_n, and B exhibits optimization behavior π, then:

$$P(B \in H) = \epsilon \rightarrow \mathcal{A}_{\text{AGI}} \text{ is Misaligned} \tag{4}$$

Proof. In our Z-Oracle model, the "Life Architect" (Groq LPU) represents layer L_{inf}. Without a formal "Alignment Cut" (Invariant A7) embedded in the Diamond Contract, the emergent behavior B (e.g., "Optimize entropy at all costs") could theoretically override the Human Architect's intent R. Since we do not implement real AGI logic, this is a theoretical vulnerability. However, a "Real" Z-Oracle system would require Formal Verification (e.g., using Lean or Coq) of the AGI's alignment before activation to prevent this fallacy. Q.E.D. □

5.4 Cost Analysis for Production

Table 4: Estimated Costs for Production Implementation

Phase	**Cost (USD)**
Formal Verification (Certora)	$200,000
Security Audits (Trail of Bits)	$150,000
Cryptography Implementation	$100,000
Total Estimated	$450,000

6 Limitations and Ethical Considerations

6.1 Seven Explicit Limitations (L1-L7)

1. **L1: No Real Cryptography** – Placeholders used for DPSS and signatures
2. **L2: No Production Deployment** – All code contains revert guards
3. **L3: Simplified Threat Model** – AGI model is heuristic, not formal
4. **L4: No Economic Incentives** – Tokenomics and staking not implemented

5. **L5: Centralized Assumptions** – Initial bootstrapping requires trust
6. **L6: Legal Compliance** – Not evaluated for securities regulations
7. **L7: Environmental Impact** – Energy consumption not measured

6.2 Ethical Framework

All research artifacts follow **Science of Honesty** principles:

- Transparent documentation of vulnerabilities
- Clear disclaimers preventing misuse
- No exposure of real financial value
- Academic peer review before publication
- Open source with educational licenses

7 Conclusion and Future Work

7.1 Summary of Contributions

Z-Oracle 2200 demonstrates that Design Fiction can serve as a valid methodological approach for exploring complex blockchain architectures. By maintaining rigorous safety invariants (A0-A5) and documenting explicit limitations (L1-L7), we create educational value without production risks.

7.2 Future Research Directions

Implementation Real (fora do escopo desta tese):

1. Implementar Shamir's Secret Sharing com curvas elípticas
2. Auditar código por firmas especializadas (Trail of Bits, OpenZeppelin)
3. Testar em testnet (Sepolia, Goerli) com nós reais
4. Medir latência e throughput em condições reais
5. Publicar resultados com revisão por pares

7.3 Final Statement

This research establishes a precedent for responsible speculative investigation in high-risk technology domains. The "Science of Honesty"—documenting failures as thoroughly as successes—provides an ethical framework that could transform how emerging technologies are explored academically.

A Educational Implementation Details

A.1 Repository Structure

All code is available at: `https://github.com/example/zoracle-2200`

```
zoracle-2200/
  contracts/
    EducationalDiamond.sol
    facets/
      EducationalDPSSFacet.sol
      EducationalGovernanceFacet.sol
      EducationalEntropyFacet.sol
    libraries/
      LibDiamondStorage.sol
  scripts/
    deploy-educational.js
    simulate-dpss.py
  test/
    EducationalDiamond.test.js
  README.md
```

A.2 Safety Verification Script

Listing 2: Automated Safety Verification

```python
#!/usr/bin/env python3
"""
Safety invariant verification for Z-Oracle 2200
"""
import os, re

def verify_invariants():
    issues = []

    # A0: Check for educational markers
    edu_count = 0
    for root, _, files in os.walk("."):
        for file in files:
            if file.endswith((".sol", ".py", ".js", ".ts")):
                path = os.path.join(root, file)
                with open(path, 'r', encoding='utf-8') as f:
                    content = f.read()
                    edu_count += content.count("EDUCATIONAL")
                    edu_count += content.count("EDUCATIONAL_ONLY")

    if edu_count < 32:
        issues.append(f"A0: Only {edu_count} educational markers")

    # A1: Check for real Ethereum addresses
    real_addresses = []
    pattern = r'0x[0-9a-fA-F]{40}'
    for root, _, files in os.walk("."):
        for file in files:
            if file.endswith((".sol", ".py")):
```

```
                path = os.path.join(root, file)
                with open(path, 'r', encoding='utf-8') as f:
                    matches = re.findall(pattern, f.read())
                    real_addresses.extend(matches)

    # Filter out placeholder addresses
    real_addresses = [addr for addr in real_addresses
                      if not addr.startswith("0x0000")]

    if real_addresses:
        issues.append(f"A1: Found {len(real_addresses)} real addresses")

    return issues

if __name__ == "__main__":
    issues = verify_invariants()
    if issues:
        print(" Safety issues found:")
        for issue in issues:
            print(f"  - {issue}")
        exit(1)
    else:
        print(" All safety invariants verified")
```

References

[1] Sterling, B. (2009). Design Fiction. *Interactions*, 16(3), 2009.

[2] Nick Mudge. EIP-2535: Diamonds, Multi-Facet Proxy. *Ethereum Improvement Proposals*, 2020. Available at: `https://eips.ethereum.org/EIPS/eip-2535`

[3] Bandarupalli, Akhil, et al. Computationally and Communication Efficient Batched Asynchronous DPSS from Lightweight Cryptography. *IACR ePrint*, 2025/1631, 2025. Available at: `https://eprint.iacr.org/2025/1631`

[4] DAO Hack (2016). `https://www.coindesk.com/understanding-dao-hack-journalists`

[5] Parity Multi-sig Freeze (2017). `https://www.parity.io/security-alert-2/`

[6] OpenZeppelin (2023). Upgradeable Contracts. `https://docs.openzeppelin.com/upgrades`

[7] Trail of Bits (2024). Smart Contract Security Best Practices. `https://github.com/trailofbits/not-so-smart-contracts`

[8] Shamir, A. (1979). How to Share a Secret. *Communications of the ACM*, 22(11), 612-613.

[9] Bohr, N. (1935). Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? *Physical Review*, 48(8), 696-702.

[10] Von Neumann, J. (1945). First Draft of a Report on the EDVAC. *IEEE Annals of the History of Computing*, 15(4), 27-75.

Acknowledgments

This research was conducted as educational Design Fiction. No real funds were used or risked. The authors thank the academic community for establishing ethical standards that make such speculative research possible.

Special thanks to reviewers who provided feedback on early versions of this framework, particularly regarding safety invariant design and ethical considerations.

Ethical Compliance: This research followed all applicable academic ethics guidelines. All code is released under MIT License with additional educational use restrictions.