ZOC: Identification and Quantification of "Zombie On Chain" Smart Contracts for Blockchain Sustainability

Abstract

The immutable and persistent nature of blockchains leads to the exponential accumulation of obsolete code on the ledger. Analysis shows that up to **70% of crypto projects** end up failing, leaving behind a massive volume of digital debris.

This article proposes a new taxonomy to address this problem: the concept of **ZOC (Zombie On Chain) Smart Contract**.

A ZOC is defined as a deployed contract that exhibits continuous external inactivity for an initial period of **nine months** — a threshold chosen by analogy to a complete incubation cycle — and negligible economic value.

This delay, as well as the value thresholds, are established as an initial working hypothesis and are subject to validation and adjustment by data collected by the ZOC Tracker and future community consensus.

We detail the methodology for a tool (ZOC Tracker) capable of quantifying and classifying these digital entities. The precise identification of ZOCs is essential for refining activity metrics, securing erroneous calls, and improving the audit and durability of decentralized ecosystems.

---

1. Introduction: The Paradox of Persistence and the Need for a New Taxonomy

The promise of blockchains rests on **immutability** and **censorship resistance**. However, this permanence generates a structural consequence: the massive accumulation of **dead** or **abandoned code**. The problem is aggravated by the high failure rate in the ecosystem: analyses show that approximately **50% to 70% of crypto projects ultimately cease their activities**, leaving behind non-functional vestiges.

The root of the problem is structural and relies on three fundamental truths of smart contracts:

* A poorly designed contract **cannot be corrected** after deployment.

* It **remains immutable and public** on the ledger, sometimes even with locked funds (Locked ZOC Contracts).

* It becomes a tangible example not to follow, but also a **valuable object** of study for research and developer education.

These structural lessons confirm the necessity of a classification. This work introduces and formally defines the concept of **Zombie On Chain (ZOC) Smart Contract**.

Figuratively, when a smart contract is deployed but serves no purpose: it is there, visible, consumes space, but is functionally lifeless.

---

2. Formal Definition: The ZOC Concept (Zombie On Chain)

The term **ZOC** designates smart contracts that are deemed **non-operational and economically insignificant**.

2.1. ZOC Classification Criteria

For a contract to be categorized as a **ZOC**, it must simultaneously satisfy the following two criteria:
$$\text{Contract} = \text{ZOC IFF } (\text{Temporal Criterion X}) \land (\text{Value Criterion Y})$$

Temporal Criterion (Inactivity - X): Initial Period of 9 Months

The inactivity threshold is set at an initial period of **nine months (9 months)**. This period is chosen as a **working hypothesis**, based on the analogy of a complete gestation cycle, confirming abandonment or functional obsolescence.

The contract must not have registered **any external interaction** during this continuous period. On a network like Ethereum, this criterion represents approximately **2 million consecutive blocks** without activity.

Value Criterion (Economic Insignificance - Y)

The contract must present **negligible** total economic value. These thresholds are established as an **initial working basis** for the ZOC Tracker:

1. **Insignificant Native Balance**: The contract balance is less than 0.001 native unit of the network (e.g., 0.001 ETH).
2. **Minimal Secondary Assets:** The total market value of secondary assets held by the contract is **less than 10 USD**.

Methodological Note: The monetary thresholds and the nine-month period are subject to future adjustments. The deployment of the ZOC Tracker will allow, through statistical analysis of contract populations, to optimize these criteria so that they remain relevant to the evolution of blockchain economics and transaction costs.

2.2. Classification of ZOC Types

ZOC Type	Description	Security / Exploitation Potential
🧠 **Inert**	Deployed contract with no active logic, often libraries or tests.	Educational study, code design audit.
🔒 **Locked**	Contract containing funds that are inaccessible due to a withdrawal bug.	Critical bug documentation, analysis of unintentional honeypots.
🧨 **Dangerous**	Contract with a known, unpatched, but inactive vulnerability.	White Hat exploitation for security or exploit documentation.
🧳 **Abandoned**	Contract from a dead or migrated project, but still referenced by external entities.	Traffic recovery and gentle migration via proxy contract.

3. Towards the "ZOC Tracker": Methodology and Exploitation Strategy

The ZOC classification is implemented via the **ZOC Tracker** analytical tool.

3.1. Architecture and Data Acquisition

1. **Access to Raw Data**: The tool interfaces with an **Archive Node** (via API services like Alchemy or Infura) to ensure unlimited access to the blockchain history.
2. **Indexing Pipeline**: A data pipeline is responsible for the sequential reading of blocks. This pipeline (developed, for example, **in a language such as Python** with the Web3.py library, or **in Go** with native tools) filters and extracts the critical events necessary for ZOC analysis.
3. **Storage**: Indexed data is stored and optimized in an analytical database (e.g., PostgreSQL or **ClickHouse**), preparing the ground for complex and fast queries.

3.2. Exploitation Strategy and Traffic Recovery

The tool applies ZOC Boolean logic to generate an index of ZOC addresses and highlight opportunities.

- **Traffic Recovery (🧳) :** For Abandoned ZOCs, the **ZOC Tracker must use methods to identify residual calls** (transaction calls, delegatecalls, or interface references) that persist towards the inactive address. The tool then **proposes solutions such as implementing a new proxy contract (wrapper)** to simulate the ZOC interface, absorb erroneous calls, and redirect them to a healthy contract or information service.
- **Audit and Security (🧨 🔒) :** Dangerous and Locked ZOCs provide an essential database for **documenting vulnerabilities** and improving automated security audit tools.

3.3. Benefits and Added Value

The ZOC Tracker offers concrete benefits for the ecosystem: chain hygiene, securing erroneous calls, and potential valorization of abandoned addresses.

3.4. Advanced Exclusion Criteria: Distinguishing ZOCs from Strategically Dormant Contracts

The fundamental characteristic of blockchain persistence necessitates a rigorous approach to classification. Inactivity alone is insufficient to definitively classify a smart contract as a **ZOC** (Zombie On Chain). Certain contracts are designed to exhibit prolonged periods of dormancy for legitimate functional or strategic reasons (e.g., waiting for a time-lock expiry or consensus).

The **ZOC Tracker** integrates a set of **advanced exclusion criteria** to identify and filter out these **false positives**, ensuring the accuracy and credibility of the ZOC metric.

Exclusion Category	Purpose and Technical Rationale
Governance & Security Mechanisms	These contracts, such as **Multisig Wallets** (like Gnosis Safe) or **Decentralized Autonomous Organization (DAO) Vaults**, are often inactive for long periods while waiting for a **quorum** to be reached on a strategic proposal. If the contract's internal state (*storage slots*) indicates a pending, unexecuted transaction, the contract is flagged as **Strategically Dormant** (and therefore *not* a ZOC).
Time-Locked Contracts (Vesting/Timelock)	Contracts designed to hold and release assets on a predefined schedule (e.g., employee token vesting or delayed fund releases). Inactivity is an expected function until a specific future block or timestamp is reached. Exclusion is applied if the contract still holds unvested tokens or if the lockup period has not yet fully elapsed.
Infrastructure Components (Proxies/Libraries)	Contracts deployed solely as internal dependencies, such as **Upgrade Proxies** (e.g., UUPS or Transparent Proxy patterns) or **Logic Libraries**, are not meant to be called by end-users. Their primary function is to be called by other smart contracts. These must be entirely excluded from the inactivity criteria, as their lack of external calls is normal operational behavior.

Methodology for Exclusion: The identification of these exceptions is achieved through the **analysis of contract *bytecode* signatures** (identifying known contract patterns) and by actively **checking the contract's internal state** via Enhanced APIs (like those provided by Alchemy) to determine if a condition for activity is pending (e.g., checking if the Multisig quorum is met or if the Timelock date has expired).

This mechanism ensures that the ZOC Tracker provides a metric reflecting **true project abandonment** rather than intentional strategic quiescence.

---

4. Conclusion and Perspectives 🚀

The immutability of distributed ledgers is both their greatest strength and their greatest structural challenge.

This work has formalized one of the inevitable consequences of this permanence: the silent proliferation of abandoned smart contracts. By introducing the **ZOC** taxonomy (Zombie On Chain), we have provided the first rigorous analysis framework for distinguishing persistent code from functional code.

The **ZOC Tracker** is not just a diagnostic tool; it is a step towards more mature and responsible **chain hygiene**. By recognizing and measuring the existence of ZOCs, the technological community gives itself the means to better manage, audit with greater precision, and ensure better sustainability of the decentralized infrastructure.
