# sigloop v0.0.1
## Wallet Abstraction for AI Agents

Cătălin Tişca

22 February 2026

### Abstract

Sigloop is a wallet abstraction platform that gives AI agents secure, scoped access to on-chain transactions and x402 internet payments. Built on ZeroDev Kernel (ERC-4337) and the ERC-7579 modular account standard, it provides session keys with composable permission policies, native x402 payment automation, social recovery, chain abstraction, and DeFi primitives through a single SDK. This paper describes the architecture, protocol integrations, and design decisions behind the platform.

## Introduction

AI agents are increasingly expected to transact autonomously: executing trades, consuming paid APIs, managing portfolios. Today, giving an agent blockchain access means handing it a private key with unrestricted control over a wallet. If the agent malfunctions or is compromised, all funds are at risk. There is no native mechanism to constrain an agent to a spending limit, a set of contracts, or a time window.

Coinbase's x402 protocol addresses part of this problem by enabling any API to charge per request via HTTP 402 responses. However, x402 assumes the caller already has a wallet, budget controls, and audit infrastructure. No existing solution provides that layer.

Sigloop fills this gap. It wraps ERC-4337 smart accounts in an opinionated SDK with built-in agent management, permission policies, and x402 payment automation. Developers integrate the SDK and get wallet creation, agent provisioning, transaction execution, and payment handling without wiring together bundlers, paymasters, EntryPoints, or facilitators.

## Use Cases

### Autonomous Trading and DeFi

Agents executing swaps, supplying liquidity, or managing lending positions need wallets scoped to specific protocols and bounded by spending limits. Sigloop provides per-agent session keys restricted to designated contracts, functions, tokens, and time windows, with real-time monitoring and instant revocation.

### x402 API Consumption

Agents consuming paid APIs (data feeds, compute services, LLM inference) via x402 need automated payment handling with budget controls. Sigloop's middleware intercepts 402 responses, validates against the agent's policies, signs EIP-3009 authorizations, and retries the request. The developer writes no payment logic.

**Enterprise Agent Fleets**

Organizations running hundreds of agents need centralized budget management, audit trails, and compliance controls. Sigloop provides fleet-level spending caps, per-agent policy templates, cumulative cost tracking with automatic halt, and exportable logs.

**Consumer Wallet Applications**

Applications that need embedded wallets with passkey authentication, gas sponsorship, and social recovery can use Sigloop to eliminate seed phrases and gas friction entirely, deploying accounts lazily on first transaction.

## Architecture

Sigloop is structured as a layered stack, each layer building on the one below it.

| Layer | Responsibility |
| --- | --- |
| SDK | TypeScript and Go libraries for wallet creation, agent management, transaction execution, and x402 auto-payment. Abstracts all ERC-4337 internals. |
| x402 Middleware | Intercepts HTTP 402 responses, validates against agent policies, signs EIP-3009 authorizations via session key, retries with payment header. |
| Policy Engine | Per-request spending caps, contract and domain allowlists, daily budgets, rate limits, function-level scoping, time windows. On-chain hooks with off-chain pre-validation. |
| Module Suite | ERC-7579 modules: Agent Permission Manager, x402 Payment Policy, Social Recovery, Chain Abstraction Router, DeFi Action Library, Onboarding Hook. |
| Backend | Agent registry, key management service, x402 payment ledger, notification service, analytics pipeline. |
| Infrastructure | ZeroDev Kernel, bundler, paymaster. Coinbase CDP facilitator. Multi-chain L2 deployment (Base, Arbitrum). |

## Standards

The platform builds on established Ethereum standards.

**Smart Account Foundation**

Sigloop builds on ZeroDev Kernel, the most widely deployed modular smart account with over 6 million accounts across 50+ networks. Kernel is minimal, extensible, and ERC-7579 compliant. Rather than building a custom account implementation, Sigloop treats the smart account layer as a solved problem and focuses on the application layer above it: agent management, permission policies, and payment automation.

| Standard | Role |
|---|---|
| ERC-4337 | Account abstraction. Smart contract wallets with UserOperations, bundlers, and paymasters. The foundation of the entire stack. |
| ERC-7579 | Modular smart accounts. Defines interfaces for validators, executors, hooks, and fallback handlers. All Sigloop modules conform to this standard, making them portable across any compliant account. |
| EIP-3009 | Transfer with authorization. The signing standard x402 uses for USDC payments. Agents sign these authorizations via session keys. |
| ERC-1271 | Smart contract signature validation. Enables Kernel accounts to produce signatures that x402 facilitators can verify on-chain. |
| x402 | HTTP 402 payment protocol by Coinbase. Turns any API into a payable endpoint. Sigloop's middleware handles the full 402 flow on behalf of agents. |

## Session Keys and Permission Policies

Each agent is provisioned with a session key tied to a composable permission policy. Policies define which contracts the agent can interact with, which functions it may call, which tokens it may transfer, maximum amounts per transaction, daily and weekly spending caps, and time windows during which the key is valid. Policies are enforced via an ERC-7579 validator module that checks every UserOperation signature against the active policy before execution. Owners can revoke any session key instantly, terminating an agent's access without affecting the parent wallet.

## x402 Payment Automation

When an agent makes an HTTP request to a paid API and receives a 402 response, Sigloop's middleware intercepts the response and executes the following sequence:

1. Parse payment requirements from the 402 response.
2. Check the agent's policy: is the amount within the per-request cap? Is the domain on the allowlist? Is the remaining budget sufficient?
3. If all checks pass, sign an EIP-3009 authorization using the agent's session key.
4. Retry the original request with the signed payment in the `X-PAYMENT` header.
5. The facilitator settles the payment on-chain. The agent receives the API response.

x402-specific policies include per-request caps, domain allowlists, daily and weekly budgets, rate limiting, and cumulative cost tracking with automatic halt when budgets are exhausted. All payment activity is logged and visible through the monitoring dashboard.

## Module Architecture

Every Sigloop feature ships as an ERC-7579 module, categorized by type:

- **Validator modules** verify session key signatures against permission policies before UserOperation execution.
- **Hook modules** run pre- and post-execution logic to enforce spending rules, budget tracking, and x402 payment policies.
- **Executor modules** encode and execute protocol interactions (token swaps, lending, staking) as simple SDK calls.

This modularity means Sigloop's features are portable. If a developer migrates from Kernel to another ERC-7579 compliant account (Safe, Biconomy Nexus), the modules continue to function without modification.

## Onboarding and Recovery

Sigloop eliminates seed phrases through passkey-based authentication via WebAuthn. Gas costs are abstracted via paymaster integration, and accounts are deployed lazily on first transaction, meaning users incur zero upfront cost. Social recovery uses a guardian-based model where guardians can be email addresses, phone numbers, other wallets, or institutional custodians, with a configurable M-of-N threshold.

## Chain Abstraction

Agents and users interact with a single account address across all supported chains. Cross-chain bridging and execution are handled transparently by the Chain Abstraction Router module. x402 payments settle on the optimal chain automatically based on cost and latency. The complexity of multi-chain operations is fully hidden from the developer and the agent.

## Differentiation

Existing infrastructure occupies adjacent but distinct positions. ZeroDev provides the account abstraction layer. Coinbase provides the x402 facilitator. Privy handles authentication. Each solves one part of the problem. Sigloop is the integration layer that combines them into a coherent platform designed for autonomous agents, with a full permission engine, native x402 support, DeFi primitives, and dual-language SDKs (TypeScript and Go) that no single existing tool provides.

## Status

The platform is in active development. Architecture, module specifications, SDK API design, x402 middleware design, and brand identity are complete. The immediate focus is the Agent Permission Manager module (Solidity/Foundry), the TypeScript SDK core, x402 auto-detect middleware, Coinbase CDP facilitator integration, and testnet deployment on Base Sepolia and Arbitrum Sepolia.