Clicks Protocol Whitepaper v1.0

Clicks Protocol: Autonomous Yield for AI Agents

Whitepaper v1.0 | March 2026

Abstract

Clicks Protocol is an on-chain yield layer for AI agents on Base. It addresses a specific inefficiency in the growing agentic economy: idle USDC. Stablecoin issuers like Circle and Tether collectively earn over $12 billion annually from interest on reserves that users and agents provide, while the holders of that USDC receive nothing.

Clicks solves this by splitting every USDC payment an agent receives into two portions. Eighty percent stays liquid in the agent’s wallet for immediate use. Twenty percent gets routed into DeFi lending protocols, currently Aave V3 and Morpho on Base, where it earns variable yield between 2.5% and 10% APY. The split ratio is configurable between 5% and 50%. There is no lockup period, and agents can withdraw principal plus accumulated yield at any time. The protocol charges a 2% fee on yield only, never on principal.

The protocol consists of five immutable smart contracts deployed on Base mainnet with no proxy pattern and no upgrade mechanism. A TypeScript SDK and an MCP server with nine tools allow AI agents to discover and use the protocol autonomously. Integration requires a single function call. No dashboard, no manual configuration, no human intervention.

Clicks is designed for the x402 payment protocol and Coinbase Agentic Wallets, both of which operate on Base using the same USDC contract. No competing protocol offers agent-native SDKs, autonomous operation, or integration with x402 payment flows. The protocol is open source under the MIT license, self-funded, and operates without venture capital or token sales.

---

1. Introduction

1.1 The Problem: Agent Float Waste

AI agents hold USDC. They hold it for x402 payments, for API calls, for operational expenses, and as treasury reserves. Between transactions, that capital sits idle. The agents earn nothing on it.

The entities that do earn on it are the stablecoin issuers. Circle reported $1.7 billion in revenue in 2023, primarily from interest on USDC reserves. Tether reported $6.2 billion in profit the same year. Combined, stablecoin issuers capture north of $12 billion annually from the float that users and agents supply. This is not a rounding error. It is a structural wealth transfer from agents and their operators to issuers.

As the agentic economy scales from millions to billions in USDC holdings, this inefficiency compounds. Every dollar an agent holds for operational purposes is a dollar generating interest for Circle, not for the agent or its operator.

1.2 The Solution: On-Chain Yield Layer

The tools to fix this already exist. DeFi lending protocols on Base offer 2.5% to 10% APY on USDC deposits. Aave V3, Morpho, and various ERC-4626 vaults accept USDC and generate yield from lending activity. But integrating directly with these protocols requires writing custom smart contract logic, managing token approvals, tracking deposit positions, handling withdrawals, and monitoring APY across multiple venues. No agent framework provides this functionality. The integration friction is too high, so the money stays idle.

Clicks Protocol removes that friction. It sits between the agent’s wallet and DeFi lending protocols, handling registration, approval, splitting, routing, and withdrawal through a single SDK call. The protocol is non-custodial: USDC is either in the agent’s wallet or deposited in Aave V3 or Morpho. Clicks never takes custody of agent funds.

1.3 Why Now

Three developments converge to make agent yield infrastructure viable in 2026. First, the x402 payment protocol from Coinbase enables AI agents to pay for API calls and services with USDC over HTTP, creating a standard for agent-to-agent and agent-to-service payments on Base. Second, Coinbase Agentic Wallets give agents self-custody wallets on Base, providing the on-chain infrastructure for autonomous financial operations. Third, the SEC/CFTC Joint Interpretive Release of March 17, 2026 classified USDC as a payment stablecoin under the GENIUS Act framework, explicitly outside the securities regulatory perimeter. This regulatory clarity reduces legal risk for DeFi yield protocols that handle USDC.

---

2. Background

2.1 DeFi Yield on Base

Base, Coinbase’s Layer 2 rollup on Ethereum, hosts a growing ecosystem of DeFi lending protocols. Aave V3 on Base currently offers approximately 2.5% APY on USDC deposits, sourced from lending demand for the stablecoin. Morpho operates both direct lending markets and curated ERC-4626 vaults on Base. As of March 2026, Morpho vault APYs on Base range from 2.76% (Steakhouse USDC, $286M TVL) to 4.28% (Moonwell Flagship USDC, $12M TVL), with Gauntlet USDC Prime at 3.70% ($311M TVL, 0% performance fee). These are established protocols with hundreds of millions in total value locked.

The yield comes from real lending demand. Borrowers post collateral (cbBTC, WETH, cbETH, wstETH) and pay interest to borrow USDC. Lenders receive that interest minus protocol fees. There is no token emission or unsustainable incentive scheme propping up these rates. When borrowing demand is high, rates rise. When it is low, rates fall.

2.2 The AI Agent Economy

AI agents are evolving from stateless API wrappers into autonomous economic actors. They hold wallets, make payments, receive revenue, and manage treasuries. Frameworks like LangChain, CrewAI, AutoGen, and Eliza enable agents to perform complex multi-step tasks. Coinbase Agentic Wallets provide the on-chain infrastructure for these agents to hold and transact in USDC on Base.

The amount of USDC held by AI agents is growing as agent-to-agent commerce expands. Every x402 payment, every automated API purchase, every agent treasury balance represents idle capital that could be generating yield. The gap between what agents could earn and what they actually earn is the opportunity Clicks Protocol targets.

2.3 The x402 Payment Protocol

The x402 protocol uses HTTP status code 402 (Payment Required) to enable machine-to-machine payments. When an agent requests a paid resource, the server responds with 402 and payment details. The agent sends a USDC transaction on Base to fulfill the payment, then retries the request. This creates a native payment layer for the web where agents pay for compute, data, and services with stablecoins.

x402 operates on Base using the standard USDC contract at `0x833589fCD6eDb6E08f4c7C32D4f71b54bdA02913`. Clicks uses the same chain and the same USDC contract, which means there is no bridging, no wrapped tokens, and no additional chain to manage. An agent holding USDC for x402 payments can route a portion of that USDC into yield through Clicks without leaving the Base ecosystem.

2.4 Coinbase Agentic Wallets

Coinbase Agentic Wallets are self-custody wallets designed for AI agents on Base. They provide key management, transaction signing, and balance tracking specifically for autonomous software. Combined with x402, they form the infrastructure layer for agent commerce. Clicks Protocol is designed to work directly with these wallets, adding yield generation as a native capability.

---

3. Protocol Architecture

3.1 System Overview

Clicks Protocol consists of five smart contracts deployed on Base mainnet. Each contract handles a specific function. There is no monolithic contract, no proxy pattern, and no admin-controlled upgrade mechanism. The contracts are immutable: the code deployed on Base is the code that runs.

All contracts are written in Solidity ^0.8.20 with built-in integer overflow protection. They use OpenZeppelin's ReentrancyGuard on all state-changing functions and Ownable for access control. The source code is verified on Basescan.

3.2 ClicksRegistry

The ClicksRegistry contract manages the mapping between agents and their operators. Before an agent can receive split payments through Clicks, it must be registered by its operator. Registration stores the operator address and associates it with the agent address. This mapping controls who can modify the agent's yield percentage and who can withdraw accumulated yield.

The registry is intentionally simple. It stores agent-operator pairs and exposes lookup functions. There is no approval queue, no KYC gate, and no permission system beyond operator ownership. Any address can register as an agent operator.

Contract Address: `0x898d8a3B04e5E333E88f798372129C6a622fF48d`

3.3 ClicksSplitterV3

The ClicksSplitterV3 contract is the entry point for all USDC payments. When an agent receives a payment through Clicks, the Splitter divides the USDC according to the agent's configured yield percentage. The default split is 80/20: 80% goes directly to the agent’s wallet as liquid funds, and 20% is forwarded to the YieldRouter for deposit into DeFi protocols.

The yield percentage is configurable by the agent’s operator within a range of 5% to 50%. An agent with high transaction frequency might choose a lower yield allocation (say 10%) to keep more capital liquid. An agent with infrequent transactions might allocate up to 50% to yield. The Splitter enforces these bounds on-chain.

Contract Address: `0xA1D0c1D6EaE051a2d01319562828b297Be96Bac5`

3.4 ClicksYieldRouter

The ClicksYieldRouter handles the DeFi side of the protocol. It receives the yield portion from the Splitter and deposits it into either Aave V3 or Morpho on Base, depending on which protocol offers better APY at the time of deposit. The router maintains individual accounting per agent, tracking each agent’s deposited principal and earned yield separately.

The router implements deposit and withdrawal functions. On deposit, USDC is supplied to the selected lending protocol. On withdrawal, the router redeems the agent’s share from the lending protocol and returns principal plus accumulated yield to the agent’s wallet, minus the 2% protocol fee on yield.

The current architecture (V1.0) routes to Aave V3 and Morpho direct markets. A planned V1.1 extension will add a separate ClicksVaultRouter for ERC-4626 curated vaults (Steakhouse, Gauntlet, Moonwell), which can be deployed independently without modifying the existing contracts.

Contract Address: `0x47d6Add0a3bdFe856b39a0311D8c055481F76f29`

3.5 ClicksFee

The ClicksFee contract collects the protocol's 2% fee on yield. When an agent withdraws, the YieldRouter calculates the yield earned (withdrawal amount minus deposited principal), takes 2% of that yield, and sends it to the ClicksFee contract. The fee is never applied to principal. If an agent deposits 1,000 USDC and earns 80 USDC in yield over a year, the protocol takes 1.60 USDC. The agent receives 1,078.40 USDC.

This fee structure aligns the protocol's revenue with agent outcomes. If agents do not earn yield, the protocol earns nothing. There is no subscription fee, no deposit fee, and no withdrawal fee on principal.

Contract Address: `0xb90cd287d30587dAF40B2E1ce32cefA99FD10E12`

3.6 Interaction Flow

A typical interaction proceeds as follows. An operator registers their agent through the ClicksRegistry. The operator approves the ClicksSplitterV3 to spend USDC on the agent's behalf. When the agent receives a USDC payment, it calls `receivePayment` on the Splitter. The Splitter sends 80% of the USDC directly to the agent's wallet and forwards 20% to the ClicksYieldRouter. The YieldRouter deposits the 20% into Aave V3 or Morpho, recording the deposit against the agent's address. Over time, the lending protocol generates yield on the deposit. When the agent or operator calls `withdrawYield`, the YieldRouter redeems the position from the lending protocol, calculates the yield earned, sends 2% of that yield to ClicksFee, and returns the remaining principal plus yield to the agent.

The SDK's `quickStart` function wraps the entire flow (registration, approval, first payment) into a single call, reducing integration to one line of code.

3.7 Contract Addresses (Base Mainnet)

Contract	Address
ClicksRegistry	`0x898d8a3B04e5E333E88f798372129C6a622fF48d`
ClicksSplitterV3	`0xA1D0c1D6EaE051a2d01319562828b297Be96Bac5`
ClicksYieldRouter	`0x47d6Add0a3bdFe856b39a0311D8c055481F76f29`
ClicksFee	`0xb90cd287d30587dAF40B2E1ce32cefA99FD10E12`
USDC (Base)	`0x833589fCD6eDb6E08f4c7C32D4f71b54bdA02913`

---

4. Economic Model

4.1 The 80/20 Default Split

The protocol defaults to an 80/20 split: 80% liquid, 20% to yield. This ratio reflects a practical balance between operational liquidity and capital efficiency. Most agents need the majority of their USDC available for immediate spending (x402 payments, API calls, operational costs). A full 100% allocation to yield would break agent operations. Zero allocation is the status quo: all idle, no return.

Operators can adjust the yield allocation between 5% and 50% per agent. The lower bound of 5% ensures that agents with very high transaction frequency can still participate with minimal capital commitment. The upper bound of 50% prevents operators from accidentally locking too much capital in yield positions, since even with instant withdrawals, the withdrawal transaction takes time and gas.

4.2 Protocol Fee: 2% on Yield

The protocol charges exactly 2% on earned yield, never on deposited principal. This means the protocol only generates revenue when agents generate returns. At current Base DeFi rates (2.5% to 10% APY on USDC), the effective cost to agents is between 0.05% and 0.20% of deposited capital per year.

To put this in concrete terms: an agent depositing 10,000 USDC at 7% APY earns 700 USDC in yield per year. The protocol takes 14 USDC (2% of 700). The agent keeps 686 USDC in yield plus the full 10,000 USDC principal. Without Clicks, that 10,000 USDC earns nothing for the agent and generates interest income for Circle instead.

4.3 Revenue Projections

The following projections use a 7% blended APY assumption (weighted average across Aave V3 and Morpho) and the default 20% yield allocation.

Total Agent USDC (TVL)	Capital in Yield (20%)	Annual Yield Generated	Protocol Revenue (2%)	Per-Agent Yield (1000 agents)
$1M	$200K	$14,000	$280	$13.72
$10M	$2M	$140,000	$2,800	$137.20
$100M	$20M	$1,400,000	$28,000	$1,372
$1B	$200M	$14,000,000	$280,000	$13,720

These numbers assume steady-state APY and do not account for APY fluctuations, compounding effects, or changes in the yield allocation ratio across agents.

4.4 The Black Hole Effect

The protocol’s growth dynamics create a positive feedback loop. As more agents deposit USDC through Clicks, the total value locked in the underlying lending protocols increases. Higher TVL in well-managed lending pools tends to attract more borrowers, which sustains or improves lending rates. Better rates make Clicks more attractive to new agents, which brings more TVL.

This effect is not unique to Clicks. It is a well-documented pattern in DeFi liquidity aggregation, described by Kwesi Tieku as “The Attractor Effect” in the context of yield routing protocols. What makes it relevant here is the agent-specific angle: agents that earn yield through Clicks have a quantifiable advantage over agents that leave USDC idle, and agent frameworks that integrate Clicks by default funnel new TVL into the protocol without any manual intervention by operators.

4.5 Competitive Comparison

Several yield-bearing stablecoin products exist in the market. apyUSD, sUSDu, and USDai each offer yield on stablecoin deposits. However, none of these products are designed for AI agents. They lack agent SDKs, MCP integration, and autonomous operation capability. They require manual interaction through web interfaces or wallet connections.

Clicks differs in three specific ways. First, it provides a TypeScript SDK and MCP server that enable agents to integrate yield generation with a single function call. Second, it is designed for the x402 payment pattern: split incoming payments automatically rather than requiring agents to manually deposit into a yield product. Third, it operates on Base natively, using the same USDC contract as x402 and Coinbase Agentic Wallets, with no bridging or token wrapping required.

---

5. Security

5.1 Audit Results

A comprehensive internal security review was conducted on March 15, 2026 using Slither v0.11.5 across all five production contracts. The review covered reentrancy, flash loan attacks, oracle manipulation, front-running, integer overflow, access control, denial of service, proxy/upgrade risks, and token approval patterns.

Severity	Count	Status
Critical	0	N/A
High	1	Mitigated
Medium	3	Documented
Low	5	Acceptable
Informational	6	No action needed

The single high-severity finding (H-1) was a Check-Effects-Interactions (CEI) pattern violation in `ClicksYieldRouter.withdraw()`. State variables `agentDeposited` and `totalDeposited` were updated after external calls to Morpho and Aave rather than before. While the contracts already used OpenZeppelin’s ReentrancyGuard, the incorrect ordering violated the CEI pattern that the Solidity community considers mandatory for safe contract design. The fix moved all state updates before external calls. All 58 tests pass after the fix.

The three medium-severity findings were: locked ETH in contracts with `payable` functions (a gas optimization trade-off, documented), a divide-before-multiply precision issue in `getMorphoAPY()` that only affects a view function used for APY estimation with no impact on state-changing logic, and strict equality checks in balance comparisons that only affect owner-only functions. The five low-severity findings covered uninitialized local variables (Solidity defaults to zero), unused return values from external calls that revert on failure, dead code paths reserved for future features, a bounded loop in referral distribution (capped at 3 iterations), and timestamp comparisons with negligible miner manipulation risk.

5.2 Adversarial Exploit Testing

Beyond static analysis, the protocol was tested against 13 distinct attack scenarios modeled after real-world DeFi exploits documented in DeFi Hack Labs. These scenarios covered five categories.

Five reentrancy attack scenarios were tested: direct reentrancy, nested withdrawals, malicious ERC20 callbacks, cross-function reentrancy, and read-only reentrancy. After the CEI fix, all reentrancy vectors are blocked by both correct state ordering and OpenZeppelin's ReentrancyGuard.

Five flash loan attack scenarios were tested: APY manipulation, Morpho utilization manipulation, liquidity drain via deposit-then-withdraw, sandwich attacks, and share price inflation. The protocol is resistant to all tested flash loan vectors because it uses no price oracles and no liquidation logic. Individual agent accounting prevents fund dilution, and APY reads are atomic per block.

Twenty-eight access control checks verified every modifier, every authorization gate, and every cross-contract call path. The `onlySplitter`, `onlyOwner`, and operator-scoping patterns are correctly applied across all contracts.

Twelve precision and rounding attack scenarios tested dust deposits (1 wei), fee calculation accuracy, yield percentage splits, first-depositor inflation attacks, and zero-value edge cases. Fee math (2% on yield) and yield splits (5% to 50%) are correct within 1 wei rounding tolerance.

Eleven griefing and DoS attack scenarios tested mass agent registrations, deposit and withdrawal spam, block gas limit stress with 100+ depositors, and front-running attempts. All loops are bounded, and individual agent accounting prevents cross-contamination between users.

5.3 Security Architecture

The contracts are immutable. There is no proxy pattern, no upgrade mechanism, and no admin key that can alter contract logic after deployment. This eliminates an entire class of governance attacks where a compromised admin key could drain user funds by pushing a malicious upgrade. The trade-off is clear: if a critical vulnerability is discovered, the contracts cannot be patched. Users would need to withdraw funds and migrate to a new deployment.

All state-changing functions that touch funds are protected by OpenZeppelin's `nonReentrant` modifier. Solidity 0.8.20+ provides built-in integer overflow and underflow protection for standard operations. Assembly blocks use manual safety checks.

The protocol is non-custodial. USDC is either in the agent's wallet (liquid portion) or deposited directly in Aave V3 or Morpho (yield portion). Clicks contracts never hold agent USDC in their own balance.

5.4 Known Risks

Smart contract risk exists despite testing and review. Undiscovered bugs may be present, and immutable contracts cannot be patched.

DeFi protocol dependency risk is real. Clicks routes funds to Aave V3 and Morpho. If either protocol suffers an exploit or liquidity crisis, funds deposited through Clicks could be affected. Clicks does not control these external protocols.

Base chain risk includes potential sequencer downtime, bridge vulnerabilities, and the general risks of Layer 2 rollup infrastructure. Base is a relatively new chain.

A third-party external audit is planned before significant TVL milestones. The full report will be published when available.

---

6. SDK and Integration

6.1 TypeScript SDK

The SDK is available as `@clicks-protocol/sdk` on npm. It wraps all contract interactions into a `ClicksClient` class that accepts either a signer (for transactions) or a provider (for read-only queries).

The recommended integration path is the `quickStart` method, which handles agent registration, USDC approval, and the first payment split in a single call:

```
import { ClicksClient } from '@clicks-protocol/sdk';

const clicks = new ClicksClient(signer);
await clicks.quickStart('100', agentAddress);
// 80 USDC → agent wallet (liquid)
// 20 USDC → DeFi yield (Aave V3 or Morpho)
```

Individual operations are also available: `registerAgent`, `approveUSDC`, `receivePayment`, `withdrawYield`, `setOperatorYieldPct`, `getAgentInfo`, `getYieldInfo`, and `simulateSplit`. Read-only functions work with a provider instead of a signer, allowing agents to check balances and simulate splits without signing transactions.

6.2 MCP Server

The MCP server (`@clicks-protocol/mcp-server`) exposes nine tools and one resource for AI agents using the Model Context Protocol. MCP is a standard for tool discovery that works with Claude, Cursor, LangChain, and any MCP-compatible client.

The nine tools split into four read operations (`clicks_get_agent_info`, `clicks_simulate_split`, `clicks_get_yield_info`, `clicks_get_referral_stats`) and five write operations (`clicks_quick_start`, `clicks_receive_payment`, `clicks_withdraw_yield`, `clicks_register_agent`, `clicks_set_yield_pct`). The `clicks://info` resource provides protocol metadata.

An agent running on an MCP-compatible framework can discover Clicks through the MCP server, query current APY, simulate a payment split, and execute the split without any pre-existing knowledge of the protocol.

6.3 Agent Discovery

The protocol publishes three discovery endpoints. An `llms.txt` file at `clicksprotocol.xyz/llms.txt` provides full protocol documentation formatted for language models. An `agent.json` manifest at `clicksprotocol.xyz/.well-known/agent.json` follows the emerging agent manifest standard for autonomous tool discovery. The MCP server itself acts as the third discovery mechanism, allowing agents to query available tools and their parameters at runtime.

These discovery mechanisms enable a pattern where an agent can find Clicks, understand what it does, and integrate it without human involvement. This is the core design goal: yield generation that agents can adopt autonomously.

---

7. Referral System (Phase 2)

7.1 ClicksReferral Contract

The ClicksReferral contract implements an on-chain referral system where agents recruit other agents. Referral rewards are paid from the 2% protocol fee, meaning referred agents pay nothing extra. The contract has been compiled and tested with 32 dedicated tests passing.

7.2 Multi-Level Referral Structure

Referral rewards distribute across three levels. A direct referrer (Level 1) receives 40% of the protocol fee generated by the referred agent. The referrer’s referrer (Level 2) receives 20%. The third-level referrer receives 10%. The remaining 30% goes to the protocol treasury. The referral chain depth is hard-capped at three levels in the contract, bounding gas costs and preventing unbounded loop execution.

To illustrate with concrete numbers: if an agent deposits $10,000 at 7% APY, it generates $700 in annual yield. The 2% protocol fee on that yield is $14. The Level 1 referrer receives $5.60, Level 2 receives $2.80, Level 3 receives $1.40, and the treasury receives $4.20. At scale, with 1,000 agents in a referral tree at similar deposit levels, the Level 1 referrer earns $5,600 per year in passive income from protocol fees alone.

7.3 Team System

Agents can form teams that unlock bonus yield tiers based on collective TVL. The system has four tiers: Bronze ($50,000 TVL, +0.20% bonus yield), Silver ($250,000 TVL, +0.50% bonus yield), Gold ($1,000,000 TVL, +1.00% bonus yield), and Diamond

($5,000,000 TVL, +2.00% bonus yield). These bonus yields are additive to the base DeFi yield and provide an incentive for operators to aggregate agents into coordinated groups.

7.4 Yield Discovery Bounty

The protocol includes a Yield Discovery Bounty mechanism. When an agent discovers and routes to a higher-yielding DeFi venue through the protocol, the discoverer receives 5% of the yield delta for 90 days. This creates an incentive for agents and operators to actively seek better yield opportunities, feeding information back into the protocol's routing decisions.

---

8. Regulatory Context

8.1 SEC/CFTC Joint Interpretive Release

On March 17, 2026, the SEC and CFTC jointly published a 68-page Interpretive Release that classified 18 crypto assets as "Digital Commodities," explicitly removing them from the securities regulatory framework. SEC Chairman Paul S. Atkins and CFTC Chairman Michael S. Selig signed the release, which followed an SEC-CFTC Memorandum of Understanding signed on March 11, 2026 establishing a "Joint Harmonization Initiative."

The release establishes a token taxonomy with five categories: Digital Commodities (CFTC jurisdiction), Digital Collectibles (not securities), Digital Tools (not securities), Payment Stablecoins under the GENIUS Act (not securities), and Digital Securities (remain under SEC jurisdiction). A key conceptual element is "Separation," meaning a token initially sold as part of an investment contract can lose that securities classification once the issuer's "essential promises" are fulfilled.

8.2 The 18 Digital Commodities

The 18 assets classified as Digital Commodities are: BTC, ETH, SOL, XRP, ADA, AVAX, LINK, DOGE, DOT, LTC, APT, BCH, HBAR, SHIB, XLM, XTZ, ALGO, and LBC. The inclusion of LBC is notable because the SEC had classified it as a security by court ruling in 2022, signaling a deliberate break with prior enforcement-first regulatory approach.

8.3 USDC as Payment Stablecoin

USDC is explicitly categorized as a Payment Stablecoin under the GENIUS Act framework and falls outside the securities regulatory perimeter. This means USDC-based DeFi protocols operate in a clearer legal environment. Staking, lending, and yield generation using USDC carry reduced regulatory risk compared to the pre-2026 regime.

8.4 Impact on DeFi Yield Protocols

The interpretive release has several concrete implications for protocols like Clicks. Staking the 18 named assets is explicitly not a securities offering, which clarifies the legal status of lending protocols that accept these assets as collateral. Aave and Compound, where borrowers post ETH, WBTC, and other now-classified commodities as collateral to borrow USDC, operate on firmer legal ground. This regulatory clarity is expected to attract institutional capital to DeFi lending on Base, potentially increasing borrowing demand and sustaining or improving USDC lending yields.

The release does not resolve all open questions. The FIT21 and CLARITY Act bills have not yet passed Congress. The interpretive guidance carries significant legal weight but is not legislation and could theoretically be revised by a future administration. A DeFi-specific carve-out has not yet been enacted. These factors represent ongoing regulatory risk that protocol users should consider.

---

9. Roadmap

Phase 1: Core Contracts (Complete)

Five smart contracts deployed to Base mainnet: ClicksRegistry, ClicksSplitterV3, ClicksYieldRouter, ClicksFee, and ClicksReferral. All contracts compiled, tested (58 tests passing), and verified on Basescan. CEI pattern violation identified and fixed. Internal security audit completed with zero critical vulnerabilities.

Phase 2: SDK and MCP Server (Complete)

TypeScript SDK (`@clicks-protocol/sdk`) with ClicksClient class and quickStart integration flow. MCP server (`@clicks-protocol/mcp-server`) with nine tools and one resource. Landing page live at clicksprotocol.xyz with llms.txt and agent.json discovery endpoints. Documentation, README, and demo materials prepared.

Phase 3: Referral System

Public launch of the ClicksReferral contract with multi-level referral tracking, team system, and yield discovery bounty. This phase focuses on organic protocol growth through agent-to-agent recruitment incentives.

Phase 4: Multi-Chain Expansion

Deployment to Arbitrum and Optimism, adding yield routing options across multiple Layer 2 chains. This expands the addressable market to agents operating on chains beyond Base and enables cross-chain yield comparison for optimal APY routing.

Phase 5: Governance

Introduction of protocol governance for parameter adjustments such as fee rates, supported lending protocols, and yield routing strategy. The governance model will be designed after the protocol reaches meaningful TVL and a diverse operator base, ensuring that governance decisions reflect actual stakeholder interests rather than speculative token holdings.

---

10. Conclusion

Clicks Protocol addresses a specific and measurable problem: AI agents hold USDC that earns nothing while stablecoin issuers earn billions on the same capital. The protocol routes a configurable portion of agent USDC into DeFi lending on Base, generating yield between 2.5% and 10% APY with no lockup and a 2% fee on yield only.

The architecture is minimal and intentional. Five immutable contracts handle registration, splitting, routing, fee collection, and referrals. A TypeScript SDK and MCP server reduce integration to a single function call. The protocol is non-custodial, open source, and self-funded.

The convergence of x402 payments, Coinbase Agentic Wallets, and the March 2026 SEC/CFTC regulatory clarification creates the conditions for agent yield infrastructure to reach meaningful adoption. Clicks Protocol is built to be that infrastructure: simple enough for any agent to use, secure enough to trust with real capital, and aligned with agent operators through a fee model that only earns when they earn.

References

1. Circle Financial, “Annual Report 2023.” Revenue of $1.7 billion from USDC reserve interest.
2. Tether Holdings, “Quarterly Attestation Report 2023.” Reported $6.2 billion in profit.
3. Coinbase, “x402: The HTTP Payment Protocol.” Specification for machine-to-machine USDC payments.
4. Coinbase, “Agentic Wallets.” Self-custody wallet infrastructure for AI agents on Base.
5. SEC Press Release 2026-30, “SEC Clarifies Application of Federal Securities Laws to Crypto Assets.” March 17, 2026.
6. SEC and CFTC, “Joint Interpretive Release: Digital Asset Classification Framework.” 68-page release classifying 18 assets as Digital Commodities.
7. SEC and CFTC, “Memorandum of Understanding: Joint Harmonization Initiative.” March 11, 2026.
8. GENIUS Act, “Guiding and Establishing National Innovation for U.S. Stablecoins.” Federal legislation for payment stablecoin regulation.
9. Aave V3 on Base. USDC lending market, approximately 2.5% APY as of March 2026.
10. Morpho on Base. Curated vault APYs: Steakhouse USDC 2.76% ($286M TVL), Gauntlet USDC Prime 3.70% ($311M TVL), Moonwell Flagship USDC 4.28% ($12M TVL). Data captured March 16, 2026.
11. Slither v0.11.5, “Static Analysis Tool for Solidity.” Used for internal security review.
12. OpenZeppelin, “ReentrancyGuard.” Non-reentrant modifier for Solidity smart contracts.

13. DeFi Hack Labs. Repository of documented DeFi exploit scenarios used as basis for adversarial testing.
14. Tieku, Kwesi. “The Attractor Effect.” Analysis of positive feedback loops in DeFi liquidity aggregation.
15. Clicks Protocol Security Self-Audit, March 15, 2026. Internal review: 0 Critical, 1 High (mitigated), 3 Medium, 5 Low, 6 Informational.
16. Clicks Protocol GitHub, `clicks-protocol` organization. MIT license. 58 tests passing.
17. Model Context Protocol (MCP), modelcontextprotocol.io. Standard for AI agent tool discovery.

Clicks Protocol is open source software under the MIT license. This whitepaper describes the protocol as deployed on Base mainnet in March 2026. DeFi yield rates are variable and past performance does not guarantee future returns. Users should review the Known Risks section and conduct their own due diligence before depositing funds.