1. Abstract

AIエージェントは、デモを出て実務に入った。コードを書き(Agentic Coding)、問い合わせとバックオフィスを回し(Agentic Support & BPO)、脅威を検知して対応する(Agentic Security)。そして複数あるメインユースケースの中でも、エージェントが「お金」そのものに触れる最前線が**金融**である——運用・決済・取引。Humans today, agents tomorrow。人が使う今から、エージェントが使う明日へ、移行はすでに始まっている。

エージェントが金融で働くとき、何より重要なのは**AIの動きが見えること**である。エージェントが何を判断し、なぜその取引を選び、どのリスクを取っているか——それが外から見えなければ、委託は成立せず、監査は再現できず、リスク管理は始まらない。AIに資産を触らせる時代の前提条件は、モデルの賢さではなく、**行動の可視性と説明可能性**である。

いま、この問いの最前線が**オンチェーンファイナンス**(ブロックチェーン上で動く金融。いわゆるDeFi)である。理由は構造にある——**ABC = AI × Blockchain × Coin**。意図を持つエージェントは24/7自律で動き続け(A)、ブロックチェーンはエージェントが共有する、止まらず・改竄耐性があり・誰でも検証できる台帳となり(B)、ステーブルコインは人に依存しないプログラム可能なマシン決済を可能にする(C)。エージェントとオンチェーンは構造的に相性がよく、先行はEthereumのオンチェーン金融である。Total Value Locked(TVL)は2026年初頭に1,300億ドル規模まで回復した(米ドル建てでは2021年ピークに次ぐ水準、ETH建てでは過去最高水準とされる。出典: DefiLlama)。

では、そのオンチェーンファイナンスでいま何が起きているか。ハッキング被害は直近年で34億ドル規模に達し(Chainalysis)、不正の重心はコントラクトのコード脆弱性から**署名インターフェース**へ移った——攻撃者は署名画面に移ったが、守る側はまだ動いていない。ブロックチェーンは「全ての取引が公開される透明な金融」として設計されたが、現実には、バイトコードは人間に読めず、Vaultの内部配分は不可視であり、calldataは暗号の羅列にしか見えない。**透明性を標榜する金融が、最も不透明な金融になった**——この逆説の上に、2026年、2つの事実が重なった。Ethereum FoundationによるClear Signing公式化(5月12日)は「署名対象を見える化する」標準化を始動させ、4月のrsETH事件(約2.92億ドル相当の無担保mint)は、被害が個々のプロトコル(node)ではなく**プロトコル間の構造的依存(edge)を伝って**、当該資産に触れてもいない参加者へ届くことを実証した。「見える」の標準化が進むほど、その先の2つの空白——**見えても読めない**、そして**自分のnodeを固めても伝播は防げない**——が鮮明になっている。

Ninjaは、この空白を埋めるAgentic Security Controlplaneであり、**意図(Intent)と実行(Execution)の間に立つTrust Layer**である。エージェント金融のスタックでは、認証・委任(AP2 / ACP等)とウォレット・支払い(x402 / MPP等)の標準化が先行して進んでいる。間に残る3つ目の席——**監視・制御**——にNinjaが立つ。Ninja Intelligence Coreは3つのIntelligence領域を通じ

て、コントラクトの信頼性評価(**NinjaCheck**)、トランザクション意図の解析(**NinjaScan**)、クロスプロトコル依存グラフによるポジション波及予測(**Ninja Position** — Systemic Position Forecast)を提供する。全てのIntelligenceは決定論的なLayer 1(事実の確定)とLLM補完によるLayer 2(人間語への変換)の2層構造に従い、再現性と説明可能性を両立する。スローガンとして、我々はClear SigningのWYSIWYSの一歩先を**What You Read Is What You Sign(WYRIWYS)**と呼ぶ——「見える(see)」に「わかる(understand)」を重ねて、初めて「読める(read)」になる。読めて初めて、署名していい。さらにその上の層として、Intelligenceに基づくポリシー制御・自動アクションを担うShoGun(統べる / 止める)を設計しており(未出荷、Day3以降に段階導入)、「知る」から「止める」までを単一のプレーンで貫くことを目指す。

本版(2026-06版)の最大の更新は、記述の多くが「計画」から「実装済みの現在地」に変わったことである。NinjaScan / NinjaCheckはTelegram Bot上で稼働中であり、**MCP(Model Context Protocol)エンドポイントは登録不要・匿名アクセスで公開済み**(APIはB2B提供)、**x402プロトコルによるpay-per-call課金も実装済み**である——エージェントが、人を介さず呼び、払える。Ninja Positionは、対応プロトコル上にポジションを持つウォレットであれば、アドレスを1つ入力するだけで依存グラフとインシデント波及の再生が動くPoCとして稼働している(対応範囲は段階拡大中)。

本書の主読者は、プロトコル運営者・機関投資家・AIエージェント開発者、そして自衛モニタリングを必要とするオンチェーンファイナンス利用者である。本ホワイトペーパーでは、Ninjaの技術アーキテクチャ、9,000万件超のEVMコントラクト・デプロイメントを対象に解析したデータベース「not-so-smart-contracts」、rsETH事件のケーススタディ、プロダクトロードマップ、ビジネスモデル、及び競合環境について記述する。

---

2. Problem — オンチェーンファイナンスとAgentic Financeの構造的不透明性

オンチェーンファイナンスの成長は目覚ましい。TVLは大きく回復し、Morphoのような新興プロトコルは140万アドレス(公表値)を超えた。DeFAI(DeFi + AI)領域が急成長し、AIエージェントがオンチェーンで自律的に取引を行うAgentic Financeの時代が到来している。

しかしこの成長の陰で、4つの構造的な問題が解決されていない。

2.1 コントラクトの信頼性が不明

ユーザーがオンチェーンファイナンスプロトコルにアクセスする際、そのコントラクトが安全かどうかを判断する手段は極めて限られている。Verified済みかどうか、Proxy構造を持つか、OFACリストに該当するか、バイトコードの特性がexploitパターンと類似するか——これらを個人が逐一検証することは現実的ではない。

2.2 「見える」だけでは署名を守れない — Clear Signing後に残る3つの限界

署名しようとしているトランザクションが実際に何を行うのかを、ユーザーは正確に理解できない。そして不正の重心は、まさにこの点へ移動している。業界分析によれば、2022年には被害の約8割がコントラクトのコード脆弱性に起因していたのに対し、2024年にはインターフェース・署名起因が被害の大半(約86%)を占め、その被害額は2年で約3.4倍($0.7B \rightarrow$2.4B)に拡大した。**攻撃者は署名画面に移った。守る側は、まだ動いていない。** この「ブラインド署名」問題に対し、Ethereum Foundationは2026年5月12日、Clear Signingを公式活動として発表した。ERC-7730(構造化された表示記述子フォーマット)、descriptorを共有するレジストリ、記述子の正確性を裏付けるERC-8176 attestationの3点セットで、「署名対象を人間に見える形で表示する」ことを標準化する取り組みである。我々はこれを歓迎する。署名前の可視性が業界標準の課題として扱われ始めたことは、本質的な前進である。

しかし、実務の現場——1日に数十から数百の署名・マルチシグ承認に向き合うwhaleや運用者——を想定すると、「見える」が標準化されてもなお3つの限界が残る。

限界1: 量。見えても、読み切れない。 100ページの契約書を1日100回読めと言われて読み切れる人間はいない。署名数が増えるほど、確認は静かに形骸化する。怠慢ではなく、注意資源の現実的制約である。必要なのは全文表示ではなく、**その人にとって意味のある差分と危険箇所だけを、署名前に短く示すこと**である。

限界2: 読むべき場所は、読む人の役割で変わる。 同じ署名でも、個人投資家にとって重要なのは金額であり、whaleにとっては権限に期限があるかであり、DAO署名者にとっては相手コントラクトが本物かである。Web2の契約レビューでも、法務は解除条項を、経理は支払条件を、事業責任者は相手方と金額を読む。承認を回す人間は「ここを見て」と印を付けてきた。同じ仕組みがオンチェーン署名の前にも要る。

限界3: Clear Signingには構造的に見えない領域がある。 Clear Signingが表示できるのは、descriptorが存在するトランザクションだけである。詐欺師は自分の悪性コントラクトに「これは危険な権限を要求します」というラベルを付けて出荷しない。descriptorが無ければウォレットはcalldataを翻訳できず、結局ブラインド署名に戻る——**最も危険なトランザクションには、ラベルが無い。**さらに、descriptorは原則としてアプリケーション開発者の自己申告であり、嘘のdescriptorやレジストリ汚染も理論上可能である。重要なのは、これがClear Signingの欠陥ではないことだ。ERC-7730は仕様の中で「トランザクションが安全かどうかは判定しない」と明示している。設計されたnon-goalである。ゆえに、**安全性は標準の外側で、descriptorから独立したソースによって照合されるしかない。**これは選択肢ではなく論理的必然である。銀行が担保評価で借り手の自己申告だけを信じず、登記・鑑定・市場価格と突合するのと同じ構造である。

2.3 構造的伝播(edges)は、誰の管轄でもない

ウォレットの残高を表示するツールは存在する。しかし問題は「何を持っているか」の先にある。2026年4月のrsETH事件(詳細は§6)では、rsETHもAaveも一切触れていない——Fluid Liteに預けただけの——ユーザーの出金が凍結した。被害は、担保設定・共有Reserve・資産裏付けという**プロトコルの壁を越えた構造的依存関係(edge)** を伝って到達したからである。

各プロトコルは自分の境界の内側にしか責任を持てず、内側しか見えない。AaveからFluid Liteの内部は見えず、Fluid LiteからAave Reserveの全体状態は見えない。事件後の業界の対策(borrow cap、rate-limiting、isolated pool等)はいずれも妥当だが、すべて「自分のnodeをどう固くするか」である。**境界を越えるedgeそのものは、構造上、誰の管轄にも自然には入らない。** この空白は誰かの怠慢ではなく、構造の帰結である——そして、だからこそ第三者レイヤーでしか埋まらない。

2.4 AIエージェントの行動が監視不能

AI Agentが自律的にオンチェーンファイナンスプロトコルと対話し、取引を執行する時代が始まっている。しかし、Agentがどのコントラクトにアクセスし、どのような判断基準で取引を実行し、どのようなリスクを取っているかを、委託者である人間が把握する手段がない。前哨となる事例は既にある。2026年3月のResolv事件では、秘密鍵の侵害により約8,000万USRが無担保で鋳造され、約2,500万ドルの損失が発生した。この事件自体はAI Agentによるものではない。しかし、周辺プロトコルの自動処理ロジックがデペグ後も止まらずに機能し続け、二次被害を拡大させた——**人間の判断を介在させない自動実行が被害を増幅する**という、Agentic Financeで桁違いに増える障害と同型の構造である。自律的なAgentが大規模に資金を動かす時代には、この型のインシデントが質・量ともに拡大する。

これらの問題が重なることで、年間34億ドルのハッキング被害、無数のフィッシング詐欺、そして機関投資家・AI Agent運営者のオンチェーンファイナンス参入障壁が維持され続けている。

---

3. Vision

Ninjaのビジョンは、オンチェーンファイナンスとAgentic Financeにおけるセキュリティと透明性のControlplaneを構築することである。

3.1 Controlplane

「Controlplane」とは、個別のセキュリティ機能を提供するツールではなく、セキュリティに関する判断・ポリシー・対応を統合的に制御する基盤を意味する。ユーザーもAI Agentも、相手を知り(Entity)、行動を読み(Action)、資産の波及を予測し(Position)、その上でShoGunを通じてポリシーを定義し、自動化されたアクションを実行できる。

3.2 What You Read Is What You Sign(WYRIWYS)

Clear Signingの礎石はWYSIWYS——What You See Is What You Sign——である。我々はその一歩先を掲げる。**「見える(see)」に「わかる(understand)」を重ねて、初めて「読める(read)」になる。** 標準化された表示の隣に、可視化された情報を「その人が、その業務量の中で、実際に理解できる」ところまで運ぶ理解レイヤーが要る。§2.2の3つの限界——量・役割・見えない領域——への回答が、それぞれNinjaの設計に対応する:意味のある差分と危険箇所だけを短く示すこと、読む人の役割に合わせて要約を変えること、そしてdescriptorに依存しない独立ソース(オンチェーン事実と9,000万件の解析DB)からの照合である。

3.3 グラフを最も自然に描けるのは、中立な第三者レイヤーである

クロスプロトコルの依存グラフを、単一のプロトコルが描き切ることは構造的に難しい。各プロトコルの管轄と責任境界は自分のnodeに閉じており、壁の外側まで含めた全体像を維持する構造的インセンティブが働かないからである。散らばった断片を1枚に繋ぐ仕事は、**中立な第三者レイヤー**が最も自然に担える。かつてBloombergが市場に共通の情報基盤を与え、結果として全参加者を強くしたように、オンチェーンファイナンス依存グラフはプロトコルの敵ではなく、エコシステムを次の段階に押し上げる共通基盤である。Ninjaはこの立ち位置——特定のプロトコル・ウォレット・チェーンに与しない中立層——を戦略の中心に置く。

3.4 オンチェーンファイナンスからAgentic Financeへの連続性 — Intent と Execution の間の信頼レイヤー

オンチェーンファイナンスは「責任主体の曖昧さ」ゆえに規制当局から敬遠されてきた。この問題はAI Agentの登場により、逆説的に解決の糸口を得る。決定論的な行動ログとLayer 1/Layer 2の2層構造に基づく説明可能性は、「誰が・なぜ・その判断をしたか」を技術的に証明する基盤となる。

そしてAgentic Financeにおいても、構造は人間の署名と同じである。Clear Signingが読みやすさを標準化しても、判断と制御のレイヤーが無ければ、AI Agentは「読めるのに、止められない」状態に置かれる。自律的に署名するAgentにとって、それはそのままインシデントを意味する。Ninjaが目指すのは、**意図(Intent)と実行(Execution)の間に立つ信頼レイヤー**である——実行前に評価し(知る)、ポリシーと照合し(統べる)、必要なら止める(止める)。エージェント金融のスタックでは、認証・委任(AP2 / ACP等)とウォレット・支払い(x402 / MPP等)の標準化が先行して進んでおり、間に残る3つ目の席——監視・制御——が、Ninjaの取り組む場所である。

---

4. Architecture — Ninja の全体構造

Ninjaのアーキテクチャは、3つの層で構成される。

```
Ninja (Agentic Security Controlplane)
│
├── Ninja Intelligence Core(知る)
│   ├── Entity Intelligence   — 相手を知る(NinjaCheck)
│   ├── Action Intelligence   — 行動を読む(NinjaScan)
│   ├── Position Intelligence — 波及を予測する(Ninja Position / SPF)
│   └── Learning              — 学んで進化する
│
├── Ninja Delivery Layer(届ける)
│   ├── NinjaScan / NinjaCheck(Telegram Bot)
│   ├── Dashboard
│   ├── API
│   └── MCP + x402(エージェントネイティブの参照・課金レール)
│
└── ShoGun(統べる / 止める)── Day3以降に段階導入
    ├── ポリシーエンジン
    ├── ホワイトリスト管理
    ├── 承認フロー
    └── 自動アクション
```

4.1 3つの層

Ninja Intelligence Core(知る) は、Ninjaプラットフォームの中核をなす知性エンジンである。3つのIntelligence領域とLearningを通じて、判断の根拠となるIntelligenceを生成する。この分離により、解析結果は全チャネルから一貫して利用でき、ShoGunのポリシーエンジンや自動アクションのトリガーとしても機能する。人間ユーザーだけでなく、AI Agentからの呼び出しにも同一のIntelligenceを提供する。

Ninja Delivery Layer(届ける) は、IntelligenceをユーザーとAI Agentに届けるインターフェースである。現在、Telegram Bot(NinjaScan / NinjaCheck)、API、そして**MCP(Model Context Protocol)エンドポイントが稼働中**である。MCPは登録不要・匿名アクセスで公開されており、AIエージェントやAI開発環境からNinjaのIntelligenceをそのまま参照できる。Dashboardはリスク比較・ユーザー通報などの機能から段階的に実装を進めている。

ShoGun(統べる / 止める) は、Intelligenceに基づくセキュリティポリシーの定義と実行を担う層である。**現時点では未出荷であり、Day3以降に段階導入する**——ポリシーエンジン、ホワイトリスト管理、承認フロー、自動アクション(アラート発報、ポジション凍結提案等)。将来的にはAI Agentの行動ポリシー定義にも拡張する設計である。

4.2 製品ライン

製品	役割	状態
NinjaScan	トランザクション意図の解析(`/scan`)。Action Intelligenceの照会窓口	稼働中
NinjaCheck	コントラクトの信頼性評価(`/check`)。Entity Intelligenceの照会窓口。**Wallet Address評価を追加予定**	稼働中(拡張予定あり)
Ninja Position(副題: Systemic Position Forecast、以下 SPF)	クロスプロトコル依存グラフによるポジション波及の予測・可視化	PoC稼働(ウォレットアドレス1つでグラフ生成・デモ可)

4.3 エージェントネイティブの参照・課金レール — MCP + x402

Delivery Layerの設計思想は「人間にもAgentにも、同じIntelligenceを、最小の摩擦で」である。現在地は次の通り。

- **MCP(実装済)**: AIエージェントがNinjaをSSOT(信頼できる単一情報源)として参照する標準インターフェース。登録不要・匿名アクセスで呼び出せる。自社システムやAIエージェントへの組み込みは今日から始められる
- **x402 pay-per-call(実装済)**: HTTP 402(Payment Required)を基にしたWebネイティブの従量課金プロトコルに対応。アカウント登録や事前契約なしに、Agentが呼び出し都度に支払う形でNinjaのIntelligenceを利用できる。AIエージェントが自律的にサービスを購買するAgentic Finance時代の課金レールであり、Day5(AI2AI Controlplane)への布石でもある

4.4 Risk Engine Marketplace(開放されたIntelligence)

Ninja Intelligence Coreは、単一ベンダーの閉じた箱として設計されていない。サードパーティのMLリスク評価エージェントを**受入** → **評価** → **カタログ化**する枠組み(Risk Engine Marketplace)を設計しており、カタログ閲覧基盤は本番環境で稼働済みである。品質ゲートを通過した第三者の検知エンジンをIntelligence Coreに加えられる構造を目指す。x402と組み合わせることで、評価エージェントの提供者が評価を売り、呼び出し側が従量で支払う——リスク評価そのものがエージェント経済圏になる——方向を見据えている。

---

5. Ninja Intelligence Core — 3 Intelligence + Layer 1 / Layer 2

5.1 設計原則:2層アーキテクチャ

Ninjaの全Intelligenceは、以下の2層構造に厳密に従う。

Layer 1(決定論的層)

- 事実を確定する。再現性100%。LLMを使用しない
- 同一入力に対して常に同一の出力を返す
- オンチェーンデータ、バイトコード解析、既知パターンマッチング等に基づく
- この層の出力が、全ての判断の基盤となる

Layer 2(LLM補完層)

- Layer 1の出力を人間が理解しやすい形に変換する
- 事実の補完・説明を行うが、Layer 1の判定を上書きしない
- LLMが生成するのは「説明」であり「判定」ではない

この設計は、セキュリティ製品において最も重要な再現性と説明可能性を確保するためのものである。LLMの出力は確率的であり、同一入力に対して異なる結果を返し得る。セキュリティ判定の基盤をLLMに委ねることは、誤判定の原因となるだけでなく、監査やインシデント対応において判定根拠の再現が困難になる。Layer 1で事実を確定し、Layer 2はその事実を読みやすい形に変換するのみ——この原則が、Ninjaの信頼性の基盤である。

この2層構造は、WYRIWYS(§3.2)の実装原理でもある。「この人がいま見るべき箇所」を短く示すのはLayer 2の仕事だが、**事実の取得を確率的なモデルに委ねない。**誰が・いくら・どの権限を・過去の挙動とどれだけ似ているか——事実は決定論的に取得し、その上で役割に応じた要約を重ねる。AI Agentの行動監視においても同じである。Agentの取引を事後的にLLMで「解釈」するだけでは、解釈そのものが再現不能になる。Layer 1で取引の事実構造を確定し、Layer 2で監査可能な形に変換する——この構造が、Agentの説明責任を技術的に実装する基盤となる。

5.2 Trust & Reliability — 守る製品自身が、守られているか

セキュリティ製品は、それ自身が攻撃対象になる。Ninjaは自社プラットフォームに対して、OWASPの体系に整合した内部セキュリティレビューを実施し、認証・セッション管理・入力検証からサプライチェーン衛生に至る主要領域の指摘事項を、ハードニングプログラムとして体系的に解消した(2026-06時点)。

特筆すべきはLLM層の扱いである。Layer 2にLLMを用いる以上、プロンプトインジェクションはNinja自身の攻撃面になり得る。Ninjaは**プロンプトインジェクション耐性の評価基盤(注入パターンを経路×手法で体系化したコーパスによる継続的評価)を運用**しており、防御実装と評価を反復している。これはOWASP Top 10 for LLM Applicationsが第一に挙げるリスク(LLM01: Prompt Injection)への直接の対応であり、「LLMに判定を委ねない」というLayer 1/Layer 2原則を、運用面から支える取り組みである。

(個別の脆弱性情報・検査結果の詳細は、セキュリティ上の理由から本書には記載しない。機関投資家・パートナーのデューデリジェンスにはNDAの下で個別に対応する。)

5.3 Entity Intelligence — 相手を知る(NinjaCheck)

Entity Intelligenceは、コントラクトおよびアドレスの信頼性を多角的に評価する。照会窓口となる製品が**NinjaCheck**である(Telegram Botの`/check`として稼働中)。

Layer 1 の評価項目:

- バイトコードML分析:コントラクトのバイトコードから特徴量を抽出し、機械学習モデルで悪性パターンを検出
- Risk Score算出:複数の評価軸を重み付け統合したスコアリング
- ブラックリスト/OFAC照合:制裁リストおよび既知の悪性アドレスとの照合
- コントラクト属性検証:Verified状態、Proxy構造、デプロイからの経過時間等

Layer 2 の補完:

- 評価結果の自然言語サマリー生成
- リスク要因の優先順位付きの説明
- 類似コントラクトとの比較コンテキスト

評価を支えるMLリスクエンジンは複数の専門エージェント(悪性コントラクト検知、ポンジ型プロトコル検知等)へのfan-out構成として本番統合済みである。

拡張予定: 現在のNinjaCheckはコントラクトアドレスの評価を対象とする。次の拡張として**Wallet Address評価**(取引相手のウォレットの履歴・関連性・リスクプロファイル評価)を追加する。さらにその先に、§3.2で述べた役割別要約——「この署名で、あなたがいま見るべき箇所」——をNinjaCheckの照会体験に統合していく。

動作モード:

- **On-demand**: `/check` コマンドによるユーザー起点の即時評価
- **Continuous**: 依存先コントラクトに変更(アップグレード、オーナー変更等)が検出された場合の自動再評価

5.4 Action Intelligence — 行動を読む(NinjaScan)

Action Intelligenceは、トランザクションのcalldataを解析し、そのトランザクションが「何をしようとしているか」を判定する。人間が署名するトランザクションだけでなく、AI Agentが自律的に生成・実行するトランザクションも同一の解析パイプラインで処理する。照会窓口となる製品が**NinjaScan**(`/scan`)である。

Layer 1 の解析パイプライン(Action Intelligence Pipeline):

```
ParseCheck
  → TxIntentMapper(意図分類)
    → ContractArtifactResolver(コントラクト情報の解決)
      → 並列Check(複数の検証を同時実行)
        → Layer 1 Output(構造化された事実)
          → LLM補完(Layer 2)
```

1. **ParseCheck:** トランザクションデータの構文解析と基本検証
2. **TxIntentMapper:** calldataのfunction selectorから意図カテゴリ(13種)へのマッピング
3. **ContractArtifactResolver:** 対象コントラクトのABI、ソースコード、メタデータの解決
4. **並列Check:** 複数の検証ロジック(approval額検証、既知exploit署名照合、再入可能性検査等)を並列実行
5. **Layer 1 Output:** カテゴリ(13種)、アラートコード(17種)、信号機(SAFE / WARNING / DANGER)の構造化出力
6. **LLM補完:** Layer 1の構造化出力を人間が読める解析レポートに変換

13カテゴリ: Native Transfer、Token Transfer、Token Approve、Token Approve All、Permit、Permit2、DeFi Swap、Multicall、Multisig Exec、Ownership Transfer、Proxy Upgrade、Generic Call、Unknown

信号機システム:

- **SAFE:** 既知のパターンに合致し、異常なし
- **WARNING:** 注意を要する要素あり(高額approval、未検証コントラクト等)
- **DANGER:** 高リスク要素を検出(既知exploit署名、制裁アドレスとの関連等)

多層トランザクションの解析: 実際の攻撃は、しばしば多層にネストされたトランザクションの内側に隠れる。2025年のBybit事件で用いられたSafe(マルチシグ)の `execTransaction` 型攻撃を踏まえ、NinjaScanはマルチシグ実行の内側のターゲットをデコードし、内側に潜む新規コントラクトや危険な呼び出しを検出対象とする(より深い多層ネストへの対応を拡大中)。

マルチチェーン: 解析パイプラインはチェーン中立に設計されており、Ethereumに加えて新興チェーンへの対応を順次拡大している(raw トランザクションからのchainId自動検出、検証エクスプローラ横断でのProxy解決を含む)。

ERC-7730との相補性: Clear Signingのdescriptorが普及すれば、Action Intelligenceの表示文脈はさらに豊かになる。同時に、descriptorの無い・偽装されたトランザクション(§2.2 限界3)にこそ、descriptorから独立したオンチェーン事実と解析DBに基づくNinjaScanの照合が効く。標準化された「see」と、独立ソースによる「understand」は競合せず、明確に相補である。

動作モード:

- **On-demand:** `/scan` コマンドによるユーザー起点の即時解析
- **Continuous:** 監視対象ウォレットのトランザクションを自動検出・解析(AI Agentのウォレットを含む)

5.5 Position Intelligence — 波及を予測する(Ninja Position / SPF)

Position Intelligenceは、本版で最も大きく進化した領域である。製品名は**Ninja Position**、副題はSystemic Position Forecast(SPF)——「ポジションの一覧」ではなく、**クロスプロトコル依存グラフの上で「次に何があなたに届くか」を予測する**ことを目的とする。なお本書で言う「予測(Forecast)」とは、依存構造上の到達可能性——どの経路を通って影響があなたに届き得るか——を事前に提示することを指し、価格・時期の確率的予測ではない。

5.5.1 4-edge model — 被害はnodeではなくedgeを伝う

ウォレットが何を持っているか(active edge)は、`positionsOf(actor)` を呼べば誰でも見える。難しさはそこにはない。被害を実際に運ぶのは、**金が動いていなくても構造的にあなたを縛る3種類の潜在エッジ**である。

エッジ	内容	例
active edge(顕在)	実際の保有・供給・借入ポジション	Vault share、Aave供給/借入
CollateralEdge(潜在)	担保設定:この資産を担保にあれが借りられる、という規則	rsETHを担保にWETH借入可
shared Reserve(潜在)	共有市場:同じReserveを複数のactorが使っている	同一のAave WETH Reserve
Asset.backing(潜在)	資産の裏付け連鎖	rsETH → stETH → ETH

rsETHカスケード(§6)は、この4種のエッジを1回トラバースするだけで完全に再現できる。

5.5.2 hub reverse-lookup — 自分から辿っても到達できないリスク

最も効くクエリは、共有Reserveをハブとした**逆引き**である。あるReserveにポジションを持つ全actorを逆方向に列挙すると、「あなたと攻撃者が同じWETH Reserveで繋がっている」という関係が見える。これは自分のウォレットから素直に `positionsOf(0xYou)` を辿るだけでは原理的に到達できないリスクである。

なお、逆引きという操作自体は、構造が見えてしまえば誰にでも書ける——隠れた魔法ではない。難しいのはエッジを描くプリミティブではなく、**プロトコルの壁を越えて散らばった断片を1枚のグラフに繋ぐこと**であり、それを最も自然に担えるのが中立な第三者レイヤー(§3.3)である。

5.5.3 1つのグラフ、3つの読み出し

グラフが1枚で描ければ、ユースケースはそこから読み出しとして導かれる。

- **actorの予見**: グラフを1つのnodeから照会する——「T+1分後のあなたの次の一手」。個人・whale・運用者の自衛モニタリング
- **プロトコルのシステミックリスク**: グラフを集計する——どのedgeに依存が集中しているか、ストレスがどこまで伝導するか
- **Incident Overlay(時系列再生)**: グラフを時間軸で再生する——事件がどの経路で、どのactorに、どの順で届いたか

5.5.4 現在地と、正直なスコープ

Ninja PositionはPoCとして稼働している。ウォレットアドレスを1つ入力すると、オンチェーン状態からVault share・貸借ポジション・担保設定・資産裏付けを読み取り、構造グラフを動的に生成する。rsETH事件を題材としたIncident Overlayでは、平常時(T=-1)から異常mint検知(T=0)、攻撃者の供給(T=1)、共有Reserveを介した波及(T=2)までを構造グラフの上に時系列で重ね、**影響経路があなた(YOU)に届くまで**を画面上で追跡できる。対応プロトコルは主要レンディング系(Aave、Fluid、Spark等)とLST/LRTの裏付け関係から段階的に拡大中である。なお、担保設定・裏付け関係の一部は検証済みの静的定義に基づいており、中長期的にはオンチェーン監視による自動更新へ移行する。

同時に、スコープには正直でありたい。**上流の自動検知——メインネットを購読して異常mintをリアルタイムに捕捉する仕組み——は、Ninja Positionは作らないし、持たない。** それは既存の優れたサービス(Hypernative、Forta、Blockaid、Gauntlet、Chaos Labs等)の領域である。Ninja Positionが集中するのは、**edgeを描くレイヤー=クロスプロトコル依存グラフそのもの**である。可視化は終点ではない。見えて初めて、事前のシミュレーション、出口経路の設計、リスクの定量化、危機時の備えの計上が始まる——条件をコードで組み、結果を誰でも検証できるProgrammable Financeが最も得意とする領域が、その先に開ける。

5.6 Learning — 学んで進化する

Ninjaは、3つのメカニズムを通じて継続的に進化する。

1. **ユーザー通報 → Ground Truth → モデル再学習**: ユーザーからのフィードバック(誤検知報告、不審トランザクション通報等)をGround Truthとして蓄積し、検知モデルの精度向上に

活用する。通報UIは実装済みであり、判定精度を体系的に検証するための評価データベースの整備も進んでいる

2. **アラートコード蓄積**: 新たなインシデントが発生するたびに、そのパターンを分析し、新しい検知ルール(アラートコード)として追加する。検知対象は時間とともに拡大し続ける(脅威データベースの整備を継続中)
3. **外部知見吸収**: 公開脅威インテリジェンスやコミュニティデータ等の外部ソース(公開フィード・公式API経由)から継続的に情報を取り込み、Ninjaの知識ベースを更新する

---

6. Case Study — The rsETH Cascade(2026年4月)

本章では、波及範囲において2026年上半期を代表するシステミック事件であるrsETH事件を、依存グラフの視点から再構成する。本章の分析は、Lidoが2026年5月21日に公開したpost-mortem *Kelp Incident Review* への我々の返信(Medium、2026-06-01)に基づく。

6.1 事件の経緯

2026年4月18日、KelpDAOのrsETHを支えるLayerZeroブリッジが侵害され(Lazarus Groupの関与が指摘されている)、**約116,500 rsETH(約2.92億ドル相当)が裏付けなしでmint**された。5週間後、事件は表向き「解決」した。KelpDAOは裏付け比率を100.01%まで回復させ、償還は正常化し、lockboxはLayerZeroとChainlinkの二重検証で再保護された。LidoのEarnETH vaultは5月15日に再開し、最終損失143.98 ETHはDAOのfirst-loss layerが全額負担した。AaveのDAOは攻撃者の凍結資金の清算を可決した。

しかし、各プロトコルが修復したのは自分の**node**である。被害を運んだ**edge**——プロトコルを跨ぐ依存グラフ——は、いまも公共財として誰でも参照できる形では存在しない。

(本章の経緯・数値はLido *Kelp Incident Review*(2026-05-21)および公開ガバナンス記録に基づく。時点は2026-06-01。)

6.2 カスケードの構造分解 — 4-edgeトラバース1回で再現できる

事件を構造として書き直すと、次の4段になる。

1. 攻撃者が裏付けなしでrsETHをmintした → rsETHの裏付け資産連鎖(**Asset.backing**)が壊れた
2. Aaveには「rsETHを担保にWETHを借りられる」という規則があった → これが**CollateralEdge**である
3. 結果、AaveのWETH Reserveが利用率99%に張り付き、同じWETH Reserveを共有する全actorに資金調達ストレスが伝播した(**shared Reserve** edge)
4. そして、**rsETHに一度も触れたことのない人の出金が、凍結した**

最後の1行を具体的に辿る。Fluid Liteのvaultに預けただけのユーザーがいた。rsETHは持っていない。Aaveでポジションを開いたこともない。持っていたのはFluid Liteのvault shareだけである。しかしFluid LiteはそのshareをAave WETH ReserveからWETHを借りてstETHループを回していた。rsETH事件がそのWETH Reserveを99%に張り付かせた瞬間、引き出せるWETHは物理的に消え、Fluid Liteはループを巻き戻せず、預金者の出金は凍結した。

これはFluid Liteに固有の話ではない。複数の主要レンディング市場のrsETH関連ポジションや、共有WETH Reserveに依存していた多数のループポジションが、相次いで凍結や巻き戻し失敗に陥った。**攻撃者からあなたまで、金は一度も直接動いていない。それでも被害は届く——被害はnode(個々のポジション)ではなくedge(構造的依存)を伝うからである。**

6.3 Ninja Position による再現 — Incident Overlay

Ninja PositionのPoCは、このカスケードを4-edge modelの1回のトラバースとして再現し、テスト済みである。Incident Overlayは事件を4つの局面で時系列再生する。

局面	グラフ上で起きること
T=-1(平常時)	構造グラフ:あなたの保有、Fluid LiteのAaveポジション、Aaveの担保設定、資産裏付けが静的に見える
T=0(検知)	rsETHの異常mintをroot eventとして検知。rsETHと関連Reserveが警戒対象として強調される
T=1(供給)	攻撃者がrsETHをAaveに供給。攻撃者・rsETH・Aave rsETH Reserveの関係がtemporal edgeとして重なる
T=2(波及)	共有WETH Reserveを介した影響経路が描かれ、**経路の終点としてYOU(あなたのポジション)に到達する**

机上の図ではない。対応プロトコル(現在: Aave / Fluid / Spark ほか拡大中)にポジションを持つウォレットであれば、アドレスを1つ入力するだけで、あなた自身のバージョンのこのグラフがPoC上で動く。

6.4 プロトコル側の応答と、残された空白

Lidoのreviewは7項目以上のフレームワーク変更を提示する、業界に向けた真摯な提案である。Aaveのガバナンスフォーラムでも、60名超の参加者による150を超える投稿の活発な議論が行われた(2026-06-01時点)。挙げられた対策——borrow cap、rate-limiting、cooldown、isolated pool設計、信用階層別の借入、供給側の直接監視——はいずれも的確で必要である。しかし共通する視点は「自分のプロトコル(node)をどう固くするか」である。

興味深いのは、Lidoの変更項目を読むと、主要な変更の多くが依存グラフの「edge」に自然に対応することだ。

Lidoのフレームワーク変更	対応するグラフのedge
① 供給側の直接監視	Asset.backing edge(裏付けの劣化を見る)
② 二次効果の一級市民化	shared Reserve edge(共有Reserve上の波及)
⑤ 再帰戦略のより深いレビュー	edgeに沿った経路シミュレーション
⑥ 利回りと出口品質の分離	nodeのストレススコアリング(高APY≠安全)

プロトコル側も、独立に同じ構造へ辿り着きつつある。これは競合の出現というより、**このカテゴリが実在することの何よりの証拠**である。ただし各プロトコルは自分の管轄内のedgeしか描けず、断片は自然には1枚に繋がらない。1枚のグラフを最も自然に担えるのは中立な第三者レイヤーである(§3.3)——それがNinja Positionの立つ場所である。

---

7. not-so-smart-contracts — 大規模EVMコントラクト解析DB

Ninjaは独自に構築した大規模EVMコントラクト解析データベース「not-so-smart-contracts」を保有している。具体的には、**EVM互換チェーンを対象に、9,000万件超のコントラクト・デプロイメントから重複排除したユニークバイトコード1,500万件以上を解析**している(※ 同一バイトコードの再デプロイを統合した実解析対象数)。このデータベースの名称は、Trail of Bits(Crytic)が公開した同名のスマートコントラクト脆弱性パターン集(https://github.com/crytic/not-so-smart-contracts)に敬意を表して命名したものである。Trail of Bitsのリポジトリがパターン学習を目的とした教育リソースであるのに対し、Ninjaのnot-so-smart-contractsはEVMコントラクトを大規模に解析したプロダクションデータベースであり、用途・規模ともに異なる。このデータベースは、Ninja Intelligence Coreを2つの方向で強化する。

7.1 Action Intelligence強化:selectorマッピング

解析対象コントラクト群から抽出したfunction selectorとメソッド名のマッピングのうち、有効性の高いものを選別してAction Intelligenceの意図分類に統合している(**本番稼働済み**)。これにより、ABIが公開されていない未検証コントラクトのトランザクションに対しても、function selectorからメソッド名を広く解決できる。従来「Unknown」と判定されていたトランザクションの意図分類が大幅に改善され、Action Intelligenceのカバレッジが拡大した。

7.2 Entity Intelligence強化:バイトコード埋め込み

各コントラクトのバイトコードから特徴量を抽出し、ベクトル埋め込み(embedding)として格納している。これにより、新規コントラクトのバイトコードを既知のコントラクト群と類似度比較できる。例えば「過去のexploit対象コントラクト群と高い構造的類似」といった判定が可能になり、Entity Intelligenceのリスク評価に定量的な根拠を提供する。未知のコントラクトであっても、過去のexploitパターンとの構造的類似性から早期にリスクを検出できる。

7.3 独立ソースとしての意味

§2.2で述べた通り、Clear Signingのdescriptorは開発者の自己申告である。9,000万件のデプロイメントを横断するこのデータベースは、自己申告に依存しない**独立ソース**としてNinjaの照合能力を支える。descriptorの無いコントラクト、偽装されたdescriptorに対しても、バイトコードと履歴という「オンチェーンの事実」から評価を組み立てられる。

---

8. Product & Customer Roadmap — Day1 to Day5 と現在地

Ninjaのプロダクトは、Day1からDay5までの5段階で拡張される。各Dayはターゲット顧客・チャネル・Intelligence領域・動作モード・課金モデルが段階的に拡張される構造を持つ。

8.1 現在地(2026年6月)

ロードマップを記述する前に、現時点で実装・稼働しているものを明示する。

領域	**稼働中(2026-06)**
Bot	NinjaScan `/scan` (TX解析)、NinjaCheck `/check` (コントラクト評価)— 公開ローンチ済
API / MCP	B2B API基盤、**MCPエンドポイント(登録不要・匿名アクセス)**
課金レール	**x402 pay-per-call(実装済)**
Intelligence	MLリスクエンジンの本番統合(fan-out構成)、selector DB統合、Safe `execTransaction` 再帰デコード、マルチチェーン対応の拡大
Position	**Ninja Position PoC**(依存グラフ生成 + rsETH Incident Overlay、デモ可)
Dashboard	段階実装中(リスク比較、ユーザー通報UI)
運用	利用メトリクス計測、運用ダッシュボード、セキュリティハードニング(§5.2)

当初Day3に置いていたMCP対応は、AIエージェントからの参照需要を踏まえて前倒しで出荷した。x402はロードマップ策定時点では採用判断をしていなかった項目であり、Agentic Financeの進展に合わせて追加実装したものである。

8.2 Day1:Foundation — B2Cカスタマーベース獲得【現在】

- **ターゲット顧客**: B2C個人
- **チャネル**: NinjaScan / NinjaCheck(Telegram Bot)+ MCP(AIエージェント)
- **Intelligence**: Entity Intelligence + Action Intelligence
- **動作モード**: On-demand(ユーザー能動)
- **課金**: 基本無料(Free)+ x402による従量アクセス。Day1の最大目的はカスタマーベース獲得であり、サブスクリプション課金は次Dayから本格化する
- **並行施策**: B2Bパイロット1〜2社を確保し、Day2以降のB2B展開の前準備を並行進行する。**NinjaCheckのWallet Address評価**を近期拡張として追加する

8.3 Day2:Continuous Monitoring — B2C受動監視 + 課金開始

- **ターゲット顧客**: B2C個人(継続)
- **チャネル追加**: Dashboard(一般提供)
- **Intelligence追加**: Position Intelligence(Ninja PositionのPoC→製品化)+ Learning初期(通報データ統合)
- **動作モード追加**: Continuous監視(受動アラート)
- **課金**: 本格開始。Free / Lite / Pro の3プラン
- **前提条件**: Day1での有料化意思検証が完了し、B2Bパイロットが並行進行していること

8.4 Day3:Protocol Expansion — B2B API + ShoGun初期

- **ターゲット顧客**: オンチェーンファイナンスプロトコル運営
- **チャネル追加**: B2B API(Read / Monitor / Control APIの3階層構成)の本格展開
- **機能追加**:
  - プロトコル個別監視項目(フラッシュローン対策、Vault戦略逸脱検知、ガバナンス提案分析)
  - **Ninja Positionのプロトコル向け読み出し**(システミックリスク集計、§5.5.3)
  - ShoGun(統べる / 止める)初期(ポリシーエンジン、ホワイトリスト、承認フロー)
- **Intelligence**: Learning本格稼働

8.5 Day4:Institutional Control — ShoGun拡充

- **ターゲット顧客**: CEX・機関投資家・ファミリーオフィス・コンプライアンスチーム

- **機能追加**: 個別ガバナンスポリシーDSL、自動アクション拡充、コンプライアンスレポートの自動生成
- **チャネル**: 機関向け専用Dashboard、SLA保証付きAPIライン

8.6 Day5:AI↔AI Controlplane

- **ターゲット顧客**: AIエージェントプロバイダー、AI2AIエコシステム
- **機能**: MCP対応(AI→Ninja参照)から一歩進み、**AIエージェント同士のポリシー仲介・契約検証・インシデント隔離**を提供
  - Agent間取引ポリシー合意プロトコル
  - 行動ログの暗号学的証明
  - 異常Agent自動隔離
- **位置付け**: Agentic Finance時代の信頼性インフラとしての完成形。**実装済みのMCP(参照レール)とx402(決済レール)は、このAI2AI Controlplaneの土台である**

統合ロードマップ表

フェーズ	主要顧客	チャネル	Intelligence	動作モード	課金
Day1 Foundation【現在】	B2C個人 + AI Agent	NinjaScan / NinjaCheck + MCP	Entity + Action(+ Position PoC)	On-demand	基本無料 + x402従量
Day2 Continuous	B2C個人	+ Dashboard	+ Position製品化 + Learning初期	+ Continuous	Free / Lite / Pro
Day3 Protocol	オンチェーンファイナンスプロトコル	+ B2B API本格	+ Learning本格	+ ShoGun初期	B2B API課金
Day4 Institutional	CEX・機関	+ 専用Dashboard	(同上)	+ ShoGun拡充(ガバナンスDSL)	Enterprise課金
Day5 AI2AI	AI Agentプロバイダー	+ Agent間プロトコル	(同上)	+ AI2AI仲介	Agent課金(x402基盤)

---

9. Competitive Landscape

オンチェーンファイナンスセキュリティ、ポートフォリオ可視化、Agentic Financeのセキュリティ領域には複数の既存プレイヤーが存在する。本章ではそれらを機能の性質に基づいて3つの類型に分類し、Ninjaとの差を可視化する。

9.1 類型の定義と機能対応マトリクス

- **Security Detection(検知・遮断)**:危険の検知・通知を中核とし、一部は実行ブロックまで提供する類型。Blockaid(Cosigner)、GoPlus、Hypernative(Firewall)、Forta等が該当する
- **Asset Visualization**:ポジション表示に特化し、セキュリティ評価を提供しない類型。Zapper, Zerion, DeBank, Exponential.fi等が該当する
- **Control Layer**:ポリシーに基づく統合的な実行制御まで踏み込む類型。DeFi Saver(自動執行に特化、部分的)と**Ninja**(ShoGunは段階導入中)が該当する

※ 本表は調達規模・採用実績のある主要事業者を対象とした、2026年6月時点の公開情報に基づく独自評価である。エージェントネイティブ系の新興(Openfort、Human.tech等)については§9.5の留意点を参照。

類型	サービス	Entity	Action	Position	LLM説明	アラート	ダッシュボード	実行制
Security Detection	Blockaid	△	○	x	△	○	△	○(Cosig
Security Detection	GoPlus	○	△	x	x	△	x	x
Security Detection	Hypernative	△	○	x	x	○	○	○(Firewa
Security Detection	Forta	△	○	x	x	○	△	△
Asset Viz	Zapper	x	x	△	x	x	○	x
Asset Viz	Zerion	x	x	△	x	x	○	x
Asset Viz	DeBank	x	x	△	x	△	○	x
Asset Viz	Exponential.fi	x	x	○	△	x	○	x
Control Layer	DeFi Saver	x	x	△	x	○	○	○(自動執行)
Control Layer	**Ninja**	○	○	○	○	○	△(段階実装中)	計画(ShoGun

○ = 主要機能として提供 △ = 部分的に対応 x = 非対応(2026年6月時点の公開情報に基づく独自評価)

Security Detection類型は「危険を知らせる」ことに優れ、BlockaidのCosignerやHypernativeのFirewallのように個別トランザクションの実行ブロックまで踏み込むプレイヤーも存在する——「既存ツールはアラートを鳴らすだけ」という区分は、もはや正確ではない。一方で、クロスプロトコルの依存グラフ(Position)を主要機能として提供する事業者は本表に見当たらない。
Asset Visualization類型は「何を持っているか」の可視化に優れるが、セキュリティ評価は提供

しない。**Control Layer類型**として、検知・可視化・説明・ポリシー制御・AI向けレールを単一のプレーンに統合する事業者は依然として少なく、AI向けControl Layerは現時点で需要検証段階の新興領域である。Ninjaの差別化は個々の機能の有無ではなく、この**単一プレーンへの統合**と**依存グラフという持ち場**にある。

9.2 nodeにアラートを鳴らすか、edgeを描くか

Position領域の差は、機能の有無だけでなく**視点の違い**にある。プロトコル単位の異常検知(Forta、Hypernative等)は、node——個々のプロトコルやコントラクト——にアラートを鳴らす。これは必要かつ優れた仕事である。Ninja Positionが描くのはnodeの間の**edge**であり、「他所で起きた異常が、どの構造的依存を伝って、あなたに届くか」までを翻訳する。両者は同じ問いの異なる持ち場であり、§5.5.4で述べた通り、上流検知はNinjaの領域ではない——edgeの翻訳は既存の検知サービスの領域ではない。

9.3 空白市場 — 「誰に」×「何を」の4象限

リスク情報の市場を「誰に届けるか(Protocol ↔ Customer)」×「何を届けるか(Alert ↔ Recommendation)」で整理すると、空白が見える。

	Alert(知らせる)	**Recommendation(動き方まで)**
Protocol向け	Forta、Hypernative 等	Gauntlet、Chaos Labs 等
Customer向け(預金者・whale・運用者)	DeBank、Zapper、Nansen 等	**(2026-06時点で主要プレイヤー不在)← Ninja Position**

プロトコルにはアラートもリスク助言も届く。顧客(資産を預ける側)にはアラートまでは届くが、「あなたのポジションはこの経路で影響を受ける。だからこう動く選択肢がある」という**Customer × Recommendation**の象限には、2026年6月時点で主要プレイヤーが見当たらない。Ninja Positionはこの象限に立つ。

※ 本表のRecommendationとは「クロスプロトコル波及経路の予測に基づく、ポジション固有の行動選択肢の提示」を指す。DeFi Saverの自動執行(単一ポジションの自動化)や、Exponential.fiの静的リスク格付けは、この定義では象限外である。なお、プロトコル向けの読み出し(§5.5.3)は、Gauntlet等の経済シミュレーションと競合するものではなく、構造グラフ起点の相補的アプローチである。

9.4 Clear Signingエコシステムとの関係 — 競合ではなく相補

Ethereum FoundationのClear Signing(ERC-7730 / registry / ERC-8176)は、署名対象の「見える化」を標準化する活動であり、Ninjaの競合ではない。むしろ標準化が進むほど、その設計上の

non-goal——安全性の判定——を埋める独立評価レイヤーの必要性が鮮明になる(§2.2)。descriptorが整備された世界では、NinjaのAction Intelligenceはより豊かな表示文脈を獲得し、descriptorの無い・偽装された領域では独立ソースとしての照合価値が際立つ。両者は明確に相補である。

9.5 Ninjaの差別化ポイントと留意点

- 検知(Entity / Action)・可視化(Position)・説明(LLM)・制御(ShoGun、段階導入)・AI向けレール(MCP / x402)を**単一プレーンに統合**する設計
- 決定論的Layer 1とLLM補完Layer 2の2層アーキテクチャによる再現性と説明可能性の両立
- **クロスプロトコル依存グラフ(Ninja Position / SPF)という、中立第三者レイヤーが最も自然に担える持ち場**
- AIエージェント向けの参照(MCP)・課金(x402)レールを**実装済み**で持ち、Day5(AI2AI)までのロードマップを明示している

留意点:AI向けセキュリティは「空白」ではなく**急速に埋まりつつある領域**である。既存大手のAI対応(GoPlusのAI Agent Security API等)に加え、エージェントネイティブの新興(Openfort、Human.tech等)も登場している。Ninjaは空白が永続する前提に立たず、依存グラフ × 中立層 × 単一プレーン統合の組合せで先行を維持する戦略を取る。Customer × Recommendation市場も現時点で需要検証段階にあり、B2C → B2Bプロトコル → AI向けへと段階展開し、各フェーズの検証結果を踏まえて次の投資を意思決定する(※ 競合評価は2026年6月時点の公開情報に基づく独自評価)。

---

10. Business Model

Ninjaのビジネスモデルは、個人向けフリーミアム、B2B API、そしてエージェント向け従量課金(x402)の3軸で構成される。

個人向け(B2C)

Free / Lite / Pro の3プラン。上位プランほど利用回数・機能が拡張される。

プラン	内容
Free	`/check` 3回/日 + `/scan` 3回/日(2026-06現在)
Lite	`/check` 拡張回数/日 + `/scan` 拡張回数/日
Pro	無制限 + 優先レスポンス + 詳細レポート出力

Day1は基本無料プランでカスタマーベース獲得に専念し、Day2からLite/Proを本格投入して収益化を開始する。Free層はDay1以降もユーザー獲得チャネルとして継続的に機能する。

B2B API

Read API / Monitor API / Control API の3階層。上位階層ほど機能範囲が拡大し、価格も段階的に上昇する。価格は公開情報および市場ヒアリングに基づくベンチマークを参考に、パイロット顧客との合意を経て個別に決定する。

プラン	機能	対象カテゴリ(例)
Read API	Entity / Action Intelligenceの参照	キュレーター、ウォレットプロバイダー
Monitor API	+ Continuous監視 + アラート配信 + プロトコル個別監視項目	オンチェーンファイナンスプロトコル、AI Agentプロバイダー
Control API	+ ShoGunポリシーエンジン + 自動アクション + ガバナンスDSL	機関投資家、コンプライアンス、CEX

エージェント向け従量課金 — x402 pay-per-call(実装済)

第3のレールとして、**x402プロトコルによるpay-per-call課金を実装済み**である。アカウント登録・事前契約・APIキー発行のいずれも不要で、AIエージェント(または人間)が呼び出しの都度、HTTPレイヤーで支払いを完了してIntelligenceを取得する。サブスクリプションにもエンタープライズ契約にも馴染まない「自律的に動くAgentの単発利用」を収益化する、Agentic Financeネイティブの課金モデルであり、Day5のAgent課金の基盤となる。

※ Enterprise向けフルスイート(MCPインテグレーション、SLA保証、専用サポート)は個別対応とする。

---

11. Team

コアチーム

- **CEO 金城**: Accenture 13年。金融機関向けAIプロジェクトを多数リード。戦略立案からテクノロジーデリバリーまでの一貫した経験
- **CTO 村上**: 大規模システム開発の技術リーダー。分散システム設計、パフォーマンスエンジニアリングに深い知見

- **恵上**: 暗号資産取引所およびFireblocks(機関向け暗号資産インフラ)での実務経験。セキュリティオペレーションの実践知識。Ninja Position(SPF)PoCの開発をリード
- **大月 海(Kai Otsuki)**: B4TI(IEEE International Workshop on Blockchain for Decentralized Trust and Digital Identity)General Co-Chair。Ethereum Foundation貢献、IETF SD-JWT(RFC 9901)標準化への参画
- **CMO 菅沼**: 3言語(日本語、英語、中国語)対応。グローバルマーケティング・コミュニティ構築

アドバイザー

- ICANN DNSSEC鍵管理者(Trusted Community Representative)。インターネットインフラの最も重要な暗号鍵管理セレモニーに参画する、世界で限られたセキュリティ専門家の一人(氏名非公開)

---

12. 市場環境と規制動向

市場データ

- **オンチェーンファイナンス TVL**: $130B超(2026年初頭。米ドル建てでは2021年ピークに次ぐ水準、ETH建てでは過去最高水準とされる。出典: DefiLlama)
- **Web3セキュリティ市場**: $2.9B(2025年) \rightarrow$15.8B(2032年予測、CAGR 26.4%)
- **年間ハッキング被害額**: $3.4B(2025年、出典: Chainalysis)
- **rsETH事件**: 2026年4月、LayerZeroブリッジ侵害により約116,500 rsETH(約$292M相当)が無担保mint。共有Reserveを介して複数プロトコルの利用者に波及した、Systemic連鎖型インシデントの決定版(§6)
- **Resolv事件**: 2026年3月、秘密鍵の侵害により約8,000万USRが無担保で鋳造、$25Mの損失。周辺プロトコルの自動処理ロジックがデペグ後も機能し二次被害が拡大。Agentic Financeのセキュリティリスクを顕在化
- **Morpho**: 140万アドレス超(公表値)。Vault型オンチェーンファイナンスの急拡大を象徴
- **DeFAI(DeFi + AI)領域**: 急速に成長するセクター。AIエージェントによるオンチェーン取引が新たな市場カテゴリを形成

オンチェーンファイナンスリスクの性質は時代とともに変化している。単独プロトコルの事件期(2021-22)、Bridge事件期(2023)、LRTネスト期(2024-25)を経て、2026年上半期は**1つの事件がプロトコルの壁を越えて連鎖するSystemic連鎖期**に入った。rsETH事件はその典型であり、node単位の防御からedge単位の可視化へという本書の主張は、この市場環境の変化に対応するものである。

規制・標準化動向

Ethereum Foundation Clear Signing 公式化(2026年5月12日)

EFはClear Signingを公式活動として発表した。ERC-7730(構造化データのClear Signingフォーマット、2026年4月にV2へ更新、Draft段階)をdescriptorフォーマットとし、EFが中立スチュワードとしてホストするレジストリでdescriptorを共有、ERC-8176 attestationで正確性を裏付ける構成である。レジストリ収載は監査や安全性の保証を意味しないことをEF自身が明示しており、安全性の判定は標準の外側——独立した評価レイヤー——に委ねられている。本書の主張(§2.2、§9.4)の通り、これはNinjaにとって市場の必要性を裏書きする追い風である。

SEC DeFiフロントエンド不処分意見(No-Action Position)(2026年4月13日)

米SEC取引・市場局は、一定条件を満たすDeFiフロントエンド提供者についてブローカー・ディーラー登録を求めないとするスタッフ声明(Staff Statement、不処分意見)を発出した。主要条件は(1)ユーザーの自律性、(2)中立性、(3)透明性(利益相反の開示、サイバーセキュリティ対策、MEV戦略の公開)であり、5年間の時限措置(2031年4月13日まで)である。特に第3条件の透明性要件は、Ninja Intelligence Coreが提供する価値と方向性が整合する。NinjaのB2B APIは、フロントエンド提供者による条件充足の実務(サイバーセキュリティ対策・透明性確保)を支援し得る。

その他の規制動向

- **MiCA(EU)**: 暗号資産市場の包括的規制枠組み。2024年12月に全面施行済み(移行期間最終期限: 2026年7月)。オンチェーンファイナンスプロトコルへの適用範囲は細則策定を注視する段階にあるが、透明性とセキュリティへの要求が高まる方向性は追い風である
- **PSA改正法(2026年6月施行、日本)/ FIEA改正**: 資金決済法改正は主に暗号資産交換業者を対象とするが、取引所経由のオンチェーンファイナンスアクセスにおけるセキュリティ要件強化につながる可能性がある。2026年4月に閣議決定された金融商品取引法改正案は暗号資産を「金融商品」として再分類する方針を示しており、中期的な制度転換として注視する
- **EU AI Act**: 2024年8月発効済み、**2026年8月2日にハイリスクAI要件・透明性義務等の主要条項が適用開始**となる(一部は2027年まで段階適用)。Agentic Financeの文脈では、AI Agentの行動の説明可能性・監査可能性が規制対象となる可能性がある。Ninjaの2層アーキテクチャ(決定論的Layer 1 + LLM補完Layer 2)とプロンプトインジェクション耐性の継続評価(§5.2)は、この規制方向性と整合する

総じて、標準化(Clear Signing)と規制(SEC・MiCA・AI Act)の双方が、「見える化」とその先の「独立した評価・説明責任」への需要を中期的に生むものと位置付けている。

Getting Started

Ninjaは今日から試せる。

- **NinjaScan / NinjaCheck(Telegram Bot):** @NinjaScanBot にコントラクトアドレスを送ると、その場でリスク評価が返る
- **ドキュメント / API / MCP:** https://ninja.zksc.io/docs — MCPは登録不要・匿名アクセスで接続できる
- **PoC・パートナーシップ・投資家のお問い合わせ**: kaneshiro@zksc.io(ZKSC Inc. https://zksc.io)

References

1. DeFi Llama — Total Value Locked (TVL) Statistics. https://defillama.com/
2. Chainalysis — 2025 Crypto Crime Report. Hacking losses data ($3.4B).
3. Morpho Labs — Protocol metrics and growth data. https://morpho.org/
4. Hypernative — Company information and product documentation. https://hypernative.io/
5. Blockaid — Company information and product documentation. https://blockaid.io/
6. GoPlus Security — API documentation and company metrics. https://gopluslabs.io/
7. U.S. SEC Division of Trading and Markets — Staff Statement on DeFi Front-End Providers (April 13, 2026).
8. European Commission — Markets in Crypto-Assets Regulation (MiCA).
9. Financial Services Agency, Japan — Payment Services Act amendments.
10. European Commission — EU Artificial Intelligence Act.
11. Forta Network — Decentralized threat detection. https://forta.org/
12. Resolv Incident Report — March 2026. Compromised key exploit — unauthorized minting of 80M USR ($25M loss).
13. Web3 Security Market — $2.9B(2025)to15.8B (2032 projected, CAGR 26.4%). Source: Electronics Media / Market Research analysis (March 2026).
14. Model Context Protocol (MCP) — Anthropic (2024). https://modelcontextprotocol.io/
15. Ethereum Foundation Blog — Clear Signing: Making Transaction Approvals Safer on Ethereum (May 12, 2026). https://blog.ethereum.org/
16. ERC-7730 — Structured Data Clear Signing Format (Draft; V2 updated April 2026). https://eips.ethereum.org/EIPS/eip-7730
17. ERC-8176 — Descriptor Attestations (referenced in EF Clear Signing announcement).
18. Lido — Kelp Incident Review (post-mortem, May 21, 2026).

19. x402 — An open protocol for internet-native payments (HTTP 402-based pay-per-call). https://www.x402.org/
20. Ninja / ZKSC — "Kelp recovered. But the graph that carried the cascade is still unmapped." (Medium, June 1, 2026).
21. Ninja / ZKSC — "Ethereum Clear Signing — Now Visible. Defensible Is Still Another Question." (Medium, June 9, 2026).

Ninja — Agentic Security Controlplane for On-chain Finance & Agentic Finance What You Read Is What You Sign. Humans today. Agents tomorrow. ShoGun — See Through On-chain Finance, Control Agentic Finance