



RexHook

*Die offene Infrastrukturebene für Uniswap-V4-Hooks:
Gebührenabschöpfung, Hook-Marktplatz und Agentenökonomie*

Version 1.7 — Februar 2026

RexHook

rexhook.com

Abstract

RexHook ist die offene Infrastrukturebene für Uniswap V4 — eine vollständige Plattform aus fünf Säulen: Token-Launchpad, Hook-Marktplatz, On-Chain-Registry mit Übersetzungsschicht (Translation Layer), V4-Netzwerk-Explorer und Trading-Terminal. Die Plattform schließt zwei kritische Lücken im V4-Ökosystem: (1) es existiert kein offizielles System zur Hook-Zertifizierung, und (2) die Verbreitung von V4-Launchpads (Clanker, Flaunch usw.) hat eine fragmentierte Landschaft geschaffen, in der jede Plattform unterschiedliche Funktionssignaturen, Event-Formate und Datenstrukturen verwendet — was Trading-Tools zwingt, für jedes Launchpad eigene Adapter zu bauen. Die Registry von RexHook fungiert als Übersetzungsschicht, die heterogene Launchpad-Schnittstellen auf ein standardisiertes Schema abbildet und den Integrationsaufwand von Trading-Tools von $O(n)$ auf $O(1)$ reduziert (Theoreme 9.8–9.9). Für Token-Erstellende erfasst unser standardmäßiger DynamicFeeHook Gebühren in ETH während der Swap-Ausführung — und beseitigt damit die „Todesspirale“, die dazu führt, dass 48–60 % der Token innerhalb von 24 Stunden scheitern (Theoreme 2.1, 3.2). Der Hook-Marktplatz ermöglicht es Drittentwicklern, zertifizierte Alternativen zu veröffentlichen. Optionale Integrationen von ERC-8004 und x402 erlauben es Hooks, an der breiteren Agentenökonomie teilzunehmen. RexHook ist gebaut und einsatzbereit.

Schlüsselbegriffe: Uniswap V4, Infrastrukturebene, Übersetzungsschicht, Hook-Marktplatz, Hook-Zertifizierung, MEV, Gebührenmechanismen, ERC-8004, x402, KI-Agenten, DeFi-Middleware

INHALTSVERZEICHNIS

Zusammenfassung für Entscheider	3
1 Einleitung	6
1.1 Notation 1.2 Warum dies für Creator wichtig ist	
2 Die Todesspirale: Formale Analyse	9
2.1 Mechanik von Tax-Token 2.2 Empirische Charakterisierung	
3 RexHook: Gebührenerfassung ohne Verkaufsdruck	11
3.1 Hook-Architektur 3.2 Gebührenverteilung	
4 Gestaltung des dynamischen Gebührenmechanismus	13
4.1 Optimale Gebührenwahl 4.2 Adaptive Gebührenstufen	
5 MEV-Umverteilung: Mechanismus-Design	15
5.1 Sandwich-Angriffsmodell 5.2 Optimale Erfassungsrate	
6 Spieltheoretische Analyse	17
6.1 Auszahlungsstruktur 6.2 Gleichgewichtsanalyse	
7 Empirische Validierung	19
7.1 Literaturübersicht 7.2 Token-Lebensdauer 7.3 MEV-Daten 7.4 Prognosen	
8 Protokollökonomie	22
8.1 Token-Modell 8.2 Verteilung 8.3 Ertragsmodell 8.4 Token-Nutzen 8.5 Staking 8.6 Hook-Marktplatz 8.7 Launchpad-Ökonomie	
9 Sicherheitsanalyse	30
9.1 Bedrohungsmodell 9.2 Gegenmaßnahmen 9.3 Formale Verifizierung 9.4 Bug Bounty 9.5 Hook-Zertifizierung & Registry 9.6 ERC-8004-Kompatibilität 9.7 Übersetzungsschicht	
10 Entwicklungs-Roadmap	40
11 Verwandte Arbeiten	41
12 Agentenökonomie & Wachstumsausrichtung	42
12.1 Allmende des Hook-Ökosystems 12.2 Vertikaler Ausrichtungs-Stack 12.3 KI-Agent-Integration 12.4 x402-Zahlungsintegration 12.5 Wachstumskreislauf	
13 Schlussfolgerung	49
Referenzen	50
Anhang A: Ergänzende Beweise	52

Zusammenfassung für Entscheider

RexHook ist die Infrastrukturebene für Uniswap V4 — die Middleware, die ein fragmentiertes Ökosystem vereinheitlicht. Wir liefern fünf integrierte Säulen: ein Token-Launchpad mit sinnvollen Standard-Hooks, einen offenen Marktplatz, auf dem Entwickler Hooks veröffentlichen und daran verdienen, eine On-Chain-Registry mit Zertifizierungsstufen, eine Übersetzungsschicht, die Daten über alle V4-Launchpads hinweg standardisiert, einen Netzwerk-Explorer und ein Trading-Terminal. Trading-Tools integrieren einmal und erhalten alle Launchpads. Launchpads registrieren sich einmal und erhalten alle Trading-Tools. Wir gewinnen, wenn das Ökosystem gewinnt.

Die Probleme

Problem 1: Die Todesspirale. Token-Erstellende wollen Erträge aus der Handelsaktivität. Klassische „Tax-Token“ sammeln Gebühren in Form von Token an, die verkauft werden müssen — wodurch der Preis abstürzt. Studien zeigen, dass 48–60 % der Token innerhalb von 24 Stunden scheitern, wobei bis zu 98 % unhaltbare Merkmale aufweisen.

Problem 2: Fragmentierung des Ökosystems. V4 hat mehrere Launchpads hervorgebracht (Clanker, Flaunch und weitere in Vorbereitung), jeweils mit unterschiedlichen Funktionssignaturen, Event-Formaten und Datenstrukturen. Trading-Tools (Axiom, DexScreener, Maestro, GMGN) müssen für jedes davon eigene Adapter bauen. Router können keine standardisierten Daten abfragen. Jedes neue Launchpad bedeutet $O(n)$ -Integrationsaufwand im gesamten Ökosystem.

Problem 3: Keine Zertifizierungsinfrastruktur. Die Uniswap Foundation hat Hook-Datenstandards vorgeschlagen, aber kein Verifizierungssystem bereitgestellt. Es gibt keine Möglichkeit zu wissen, ob ein Hook sicher ist. Keine Registry. Keine Zertifizierung. Sicherheitstools tappen im Dunkeln.

Unsere Lösung: Fünf integrierte Säulen

RexHook ist eine vollständige Infrastrukturplattform, die alle drei Probleme löst:

- **Token-Launchpad** — Token mit zertifizierten Hooks bereitstellen. Unser standardmäßiger DynamicFeeHook erfasst Gebühren in ETH (nicht in Token) und beseitigt so die Todesspirale. Creator können unsere Standards nutzen oder aus dem Marktplatz wählen.
- **Hook-Marktplatz** — Offene Plattform, auf der Entwickler Hooks veröffentlichen und 70 % Tantiemen verdienen. Wir liefern sinnvolle Standards; jeder kann zertifizierte Alternativen bauen und listen. Baut jemand einen besseren Gebühren-Hook, gewinnen wir alle.
- **Hook-Registry + Übersetzungsschicht** — On-Chain-Zertifizierung (Bronze → Platin) plus ein standardisiertes Schema, das ALLE V4-Launchpads (Clanker, Flaunch, RexHook, künftige Plattformen) auf eine einheitliche Schnittstelle abbildet. Trading-Tools integrieren einmal → erhalten alle Launchpads. Launchpads registrieren einmal → erhalten alle Trading-Tools.
- **V4-Netzwerk-Explorer** — Alle Hooks, alle Pools, alle Daten. Mit Envio über mehrere Chains indexiert. Die Transparenzschicht.
- **Trading-Terminal** — Charts, Kauf/Verkauf, Privy-Wallet-Integration. Die vollständige Nutzererfahrung.

Die Todesspirale lösen (Standard-Hook)

Unser standardmäßiger DynamicFeeHook erfasst Gebühren während des Swaps, direkt in ETH. Keine Token-Anhäufung. Kein Verkaufsdruck. Keine Todesspirale. Dies ist ein Hook unter vielen — Entwickler können Alternativen bauen, und Creator wählen, was für sie funktioniert.

So funktioniert es

Nutzer swapt ETH ↔ Token → RexHook (beforeSwap + afterSwap) → Gebührenerfassung in ETH (nicht in Token) → Creator-Wallet (sofortiger Ertrag)

Abbildung 0. Ablauf der Gebührenerfassung von RexHook. Gebühren werden während der Swap-Ausführung in ETH entnommen — keine Token-Anhäufung, kein Verkaufsdruck.

Wesentliche Vorteile nach Beteiligten

Beteiligte	Ohne RexHook	Mit RexHook
Token-Creator	Erträge erfordern Token-Verkauf → Preis stürzt ab	ETH-Ertrag direkt ins Wallet, kein Verkaufsdruck
Trader	Versteckte Gebühren, Sandwich-Angriffe, Preismanipulation	Transparente Gebühren, MEV-Schutz, faire Preise
Trading-Tools	Eigener Adapter pro Launchpad (O(n))	Einmal integrieren, ALLE Launchpads (O(1))
Launchpads	Integration mit jedem Trading-Tool verhandeln	Einmal registrieren, sofort mit allen Tools kompatibel
Router	Keine standardisierten Daten/Zertifizierung abfragbar	API für Standardschema + Zertifizierungsstatus abfragbar
Hook-Entwickler	Kein Marktplatz, keine Auffindbarkeit, kein Ertrag	Hooks veröffentlichen, zertifizieren, 70 % Tantiemen verdienen

Geschäftsmodell

Die Token-Bereitstellung ist kostenlos. Die Protokollerträge stammen aus vier Strömen: (1) einem kleinen Anteil der vom Creator konfigurierten Gebühren, (2) 50 % des zusätzlich erfassten Werts (MEV, Preisauswirkung, Bot-Strafen) — Ertragsströme, die ohne RexHook nicht existieren würden, (3) dem Hook-Marktplatz (30 % Anteil an Drittanbieter-Hooks) und (4) x402-Mikrozahlungen von KI-Agenten, die auf Hook-Dienste zugreifen. Mit konservativen Prognosen von 5.000 Token-Bereitstellungen in Jahr 1 bis hin zu 75.000 in Jahr 3 prognostizieren wir Protokollerträge von 2,375 Mio. \$ (J1) → 42,95 Mio. \$ (J3).

Das Ökosystem

Man kann sich RexHook als App-Store + Middleware-Ebene für V4-Token-Launches vorstellen. Wir liefern: (1) sinnvolle Standard-Hooks für das Launchpad, (2) einen offenen Marktplatz, auf dem Entwickler Hooks bauen und verkaufen, und (3) eine Übersetzungsschicht, die alle V4-Launchpads auf ein Standardschema abbildet. Trading-Tools (Axiom, DexScreener, Maestro, GMGN) integrieren RexHook einmal und erhalten standardisierte Daten von jedem registrierten Launchpad — Clanker, Flaunch, RexHook und künftige Plattformen. Wir gewinnen, wenn das Ökosystem gewinnt.

Hook-Zertifizierung & Registry-Infrastruktur

Im V4-Ökosystem besteht eine kritische Lücke: keine offizielle Hook-Registry und kein Zertifizierungssystem. Die Uniswap Foundation hat Datenstandards vorgeschlagen, aber keine Verifizierungsinfrastruktur bereitgestellt. RexHook baut eine On-Chain-Registry, die Folgendes bietet: (1) verifizierbare Hook-Metadaten über standardisierte EVM-Schnittstellen, (2) Zertifizierungsstufen (Bronze → Platin) basierend auf Audits und Unveränderlichkeit und (3) eine Übersetzungsschicht-API, die Daten über heterogene Launchpads hinweg standardisiert. Trading-Tools integrieren einmal; Router fragen die Zertifizierung vor dem Routing ab.

Warum jetzt

Uniswap V4 startete im Januar 2025 mit nativer Hook-Unterstützung. ERC-8004 („Trustless Agents“) etablierte On-Chain-Identität und Reputation für autonome Agenten. Das x402-Protokoll ermöglichte HTTP-native Mikrozahlungen für KI-Agenten. Die Infrastruktur, um eine vollständige Hook-Ökonomie zu bauen — von der Bereitstellung über die Zertifizierung bis hin zu KI-gesteuerter Auffindbarkeit und Bezahlung — existiert nun endlich. RexHook ist als Erster am Markt mit produktionsreifen Hooks, einer On-Chain-Verifizierungs-Registry und der Integration der Agentenökonomie.

1 Einleitung

Die Entwicklung der Architektur dezentraler Börsen (DEX) von Constant-Product-Automated-Market-Makern (CPAMMs) [1] über konzentrierte Liquidität [2] bis hin zu Hook-erweiterbaren Systemen [3] hat den Gestaltungsraum für On-Chain-Handelsmechanismen schrittweise erweitert. Das im Januar 2025 bereitgestellte Uniswap V4 führt Hooks ein (programmierbare Callbacks, die an definierten Punkten des Swap-Lebenszyklus ausgeführt werden) und ermöglicht so beliebige Anpassungen des Pool-Verhaltens, ohne die Kernlogik des Protokolls zu verändern.

Trotz dieses architektonischen Fortschritts besteht ein grundlegendes Problem fort: Token-Erstellende, die Handelserträge erfassen wollen, stehen vor einem inhärenten Konflikt zwischen Gebührentnahme und Werterhalt des Tokens. Wir formalisieren dies als das Tax-Token-Trilemma:

Definition 1.1 (Tax-Token-Trilemma)

Ein gebührenerfassender Token-Mechanismus kann unter herkömmlichen ERC-20-Token-Vertragsarchitekturen nicht gleichzeitig erreichen: (i) positiven Creator-Ertrag $R > 0$, (ii) keinen zusätzlichen Verkaufsdruck $\Delta P_{\text{Verkauf}} = 0$ und (iii) erlaubnisfreien Betrieb ohne vertrauenswürdige Vermittler.

RexHook löst dieses Trilemma, indem es die Hook-Architektur von V4 nutzt, um die Gebührenerfassung innerhalb der Swap-Ausführung statt durch Token-Anhäufung umzusetzen. Über die Lösung der Todesspirale hinaus ist RexHook eine vollständige Infrastrukturplattform aus fünf Säulen: Token-Launchpad, Hook-Marktplatz, On-Chain-Registry mit Übersetzungsschicht, V4-Netzwerk-Explorer und Trading-Terminal. Diese Arbeit leistet folgende Beiträge:

1. Formale Charakterisierung des Phänomens der Todesspirale mit präzisen Instabilitätsbedingungen (Abschnitt 2)
2. Beweis für den fehlenden Verkaufsdruck bei Hook-basierter Gebührentnahme (Abschnitt 3, Theorem 3.2)
3. Optimaler MEV-Erfassungsmechanismus, aus ersten Prinzipien hergeleitet (Abschnitt 5)
4. Empirische Validierung anhand begutachteter Forschung zu Token-Ausfallraten (Abschnitt 7)
5. Spieltheoretische Analyse der Anreize der Beteiligten (Abschnitt 6)
6. On-Chain-Hook-Registry und Zertifizierung mit Verifizierungsvollständigkeit (Abschnitt 9.5)
7. Übersetzungsschicht-Architektur, die Daten über heterogene V4-Launchpads standardisiert und den Integrationsaufwand von Trading-Tools von $O(n)$ auf $O(1)$ reduziert (Abschnitt 9.7, Theoreme 9.8–9.9)
8. Rahmenwerk der Agentenökonomie: spieltheoretische Allmende-Analyse, die beweist, dass Kooperation unter Zertifizierung und KI-Agenten-Routing die dominante Strategie ist, mit optionalen ERC-8004- und x402-Integrationen (Abschnitt 12)

1.1 Notation und Vorbemerkungen

Wir legen die folgende, in dieser Arbeit durchgängig verwendete Notation fest:

Symbol	Beschreibung	Wertebereich
$P(t)$	Token-Preis zum Zeitpunkt t	\mathbb{R}^+
$V(t)$	Kumuliertes Handelsvolumen zum Zeitpunkt t	\mathbb{R}^+
L	Liquiditätstiefe (im Quote-Asset)	\mathbb{R}^+

Symbol	Beschreibung	Wertebereich
f	Gebührensatz (Anteil am Swap-Volumen)	[0, 0,30]
A(t)	Angehäuften Gebühren-Token zum Zeitpunkt t	\mathbb{R}^+
M	Marktkapitalisierung	\mathbb{R}^+
ΔP	Preisauswirkung eines Trades	\mathbb{R}
κ	Constant-Product-Invariante ($xy = \kappa$)	\mathbb{R}^+
η	MEV-Extraktionseffizienz	[0, 1]
Π	Gewinnfunktion	\mathbb{R}
τ	Zeit bis zum Ausfall (Beginn der Todesspirale)	\mathbb{R}^+
n	Anzahl der Gebührenempfänger	{1,...,10}
s_i	Anteil in Prozent für Empfänger i	[0, 1]

Tabelle 1. Übersicht der in dieser Arbeit verwendeten Notation.

1.2 Warum dies für Creator wichtig ist

Token-Erstellende stehen vor einem grundlegenden Dilemma: Sie wollen Erträge aus der Handelsaktivität erzielen, doch die dafür verfügbaren Mechanismen zerstören aktiv den Wert ihres Tokens. RexHook löst dieses Problem. Folgendes ändert sich:

Klassische Tax-Token	RexHook
Gebühren häufen sich als Token im Vertrag an	Gebühren werden während des Swaps in ETH erfasst
Verkauf der Gebühren lässt den Token-Preis abstürzen	Kein Verkaufsdruck auf den Token
48–60 % scheitern innerhalb von 24 Stunden	Prognostizierte 1,56-fache Überlebensverbesserung
Trader meiden sie wegen versteckter Kosten	Transparente, vorhersehbare Gebühren
MEV-Bots entnehmen zusätzlichen Wert	MEV wird erfasst und umverteilt

Erträge ohne Zerstörung

Mit RexHook erhalten Creator nach jedem Trade ETH direkt in ihr Wallet. Kein Warten. Kein Verkauf. Keine Preisauswirkung. Ein Token, der 100.000 \$ tägliches Handelsvolumen bei einem Gebührensatz von 5 % erzeugt, bringt 2.500 \$/Tag an Creator-Ertrag — ohne den Token-Preis zu berühren.

Längere Token-Lebensdauer

Indem der Mechanismus der Todesspirale beseitigt wird, überleben RexHook-Token länger. Längeres Überleben bedeutet mehr Handelsvolumen, mehr Gebühren und mehr Gelegenheiten, Community und Nutzwert aufzubauen. Der Unterschied zwischen einem Token, der 1 Tag hält, und einem, der 30 Tage hält, ist der Unterschied zwischen einem gescheiterten Experiment und einem tragfähigen Projekt.

Dynamische Gebühren, die mitskalieren

RexHook implementiert automatische Gebührenstufen basierend auf der Marktkapitalisierung. Token in der Frühphase können höhere Gebühren erheben (bis zu 30 %), wenn die Liquidität gering und das Überleben kritisch

ist. Mit wachsender Marktkapitalisierung sinken die Gebühren automatisch, um wettbewerbsfähig zu bleiben. Creator müssen nichts neu bereitstellen — der Hook passt sich an.

Marktkap.-Stufe	Gebührensatz	Begründung
< 100 Tsd. \$	bis 30 %	Überlebensphase: Creator-Runway maximieren
100 Tsd.–500 Tsd. \$	10–20 %	Wachstumsphase: Ertrag und Volumen ausbalancieren
500 Tsd.–1 Mio. \$	5–10 %	Skalierungsphase: wettbewerbsfähig mit Alternativen
> 1 Mio. \$	1–5 %	Reifephase: nachhaltiges Langzeitmodell

Tabelle 1b. Dynamische Gebührenstufen passen sich automatisch an die Marktkapitalisierung des Tokens an.

Erträge nach Belieben aufteilen

Ihr Creator-Anteil kann auf bis zu 10 Empfänger verteilt werden. Team-Treasury, Marketing-Wallet, Entwicklerfonds, Community-Belohnungen — einmal bei der Bereitstellung konfigurieren, und jeder Swap teilt die Erträge automatisch gemäß Ihrer Zuteilung auf. Alles On-Chain, alles transparent, alles automatisch.

2 Die Todesspirale: Formale Analyse

2.1 Mechanik von Tax-Token

Betrachten wir einen Token T mit Gebührensatz $f \in (0, 1)$, der über Transfer-Hooks im ERC-20-Vertrag umgesetzt ist. Für jeden Swap des Volumens v häuft der Vertrag fv Token an. Sei $A(t)$ die zum Zeitpunkt t angehäuften Token-Menge:

$$A(t) = f \cdot \int_0^t v(s) ds = f \cdot V(t) \quad (2.1)$$

Um Erträge zu realisieren, müssen die angehäuften Token gegen das Quote-Asset (typischerweise ETH) verkauft werden. Unter Constant-Product-AMM-Mechanik mit Reserven (x, y) und Invariante $\kappa = xy$ ergibt der Verkauf von Δx Token:

$$\Delta y = y - \kappa/(x + \Delta x) = y \cdot \Delta x/(x + \Delta x) \quad (2.2)$$

Die resultierende Preisauswirkung ist:

$$\Delta P/P = -(2x\Delta x + \Delta x^2)/(x + \Delta x)^2 \approx -2\Delta x/x \text{ für kleine } \Delta x \quad (2.3)$$

Annahme 2.1 (Rationale Halter)

Token-Halter beobachten Preisbewegungen und passen ihre Positionen an. Ein Preisrückgang der Größenordnung $|\Delta P/P| > \theta$ löst zusätzliche Verkäufe mit einer Wahrscheinlichkeit $p(\theta)$ aus, die monoton in $|\Delta P/P|$ steigt.

Theorem 2.1 (Instabilität der Todesspirale)

Sei T ein Tax-Token mit Gebührensatz $f > 0$ und Liquiditätstiefe L . Unter Annahme 2.1 existiert ein kritischer Volumenschwellenwert V^* , sodass für $V(t) > V^*$ das System in ein instabiles Regime eintritt, in dem gilt:

$$\lim_{t \rightarrow \infty} P[P(t) < \varepsilon P(0)] = 1 \text{ für jedes } \varepsilon > 0$$

Beweis. Definiere die Verkaufsdruckfunktion $S(t) = A(t)/L$. Aus (2.1) folgt $S(t) = fV(t)/L$. Werden angehäuften Token verkauft, beträgt die Preisauswirkung aus (2.3) $\Delta P/P \approx -2S(t)$. Unter Annahme 2.1 löst dies zusätzliche Verkäufe mit Wahrscheinlichkeit $p(2S(t))$ aus. Das erwartete Zusatzvolumen ist $E[\Delta V] = p(2S) \cdot \gamma L$ für ein $\gamma > 0$, das die Reaktionsfreudigkeit der Halter abbildet. Das System wird instabil, wenn $dS/dt > 0$ dauerhaft gilt, was eintritt, wenn $f \cdot p(2S) \cdot \gamma > S/V$. Für $V > V^* = L/(2f\gamma)$ gilt diese Bedingung für $p(2S) > 1/(2\gamma)$, was für hinreichend große S aufgrund der Monotonie von p erfüllt ist. ■

2.2 Empirische Charakterisierung

Akademische Forschung und Branchenanalysen liefern starke empirische Belege für das Phänomen der Todesspirale. CertiK [11] analysierte 100.260 Ethereum-Token von Oktober 2023 bis August 2024 und identifizierte 48.265 Rug Pulls (48,14 %). Cernera et al. [13] stellten fest, dass etwa 60 % der Token auf Ethereum und BSC weniger als einen Tag aktiv sind.

Quelle	Befund	Implikation für Theorem 2.1
CertiK [11]	48,14 % Rug-Pull-Rate (100.260 Token)	Fast die Hälfte der Token zeigt Ausfallmuster
Cernera et al. [13]	~60 % der Token < 1 Tag aktiv	Schnelles Scheitern, konsistent mit Todesspiralen-Schwelle

Quelle	Befund	Implikation für Theorem 2.1
Xia et al. [12]	~50 % zeigen Scam-Merkmale	Gebührenmechanismen korrelieren mit betrügerischen Mustern
Kalacheva et al. [14]	>98 % betrügerische Merkmale	Obergrenze problematischen Token-Verhaltens

Tabelle 2. Übersicht der begutachteten Forschung zu Token-Ausfallraten, die den in Theorem 2.1 beschriebenen Mechanismus der Todesspirale stützt.

Anmerkung 2.1

Der durchgängige Befund mehrerer Studien, dass 48–60 % der Token rasch scheitern, stützt die Vorhersage von Theorem 2.1, dass ein hohes anfängliches Handelsvolumen die Rückkopplungsschleife der Todesspirale auslöst. Token mit Gebührenmechanismen (Tax-Token) sind besonders anfällig, da angehäufte Gebühren liquidiert werden müssen, wodurch der Verkaufsdruck entsteht, der die Kaskade in Gang setzt. Eine vollständige empirische Validierung mit granularen Volumen-Preis-Korrelationsdaten wird in Abschnitt 7 vorgestellt.

Abbildung 1. Vergleichende Ablaufdiagramme: (a) klassische Tax-Token-Gebührenmechanik mit der Rückkopplungsschleife, die zur Todesspirale führt — Swap ausgeführt → Gebühren-Token häufen sich an → Liquidationsauslöser → Preis fällt → Halter geraten in Panik → zusätzliche Verkäufe (Rückkopplung); (b) RexHooks DynamicFeeHook erreicht keinen zusätzlichen Verkaufsdruck durch ETH-Entnahme innerhalb des Swaps — Swap initiiert → afterSwap-Callback → Gebühr in ETH direkt aus dem Swap entnommen → sofortige Verteilung an Creator und Protokoll, keine Anhäufung, kein Verkaufsdruck.

3 RexHook: Gebührenerfassung ohne Verkaufsdruck

3.1 Hook-Architektur

Uniswap V4 führt sechs Hook-Callback-Punkte im Swap-Lebenszyklus ein. RexHooks DynamicFeeHook arbeitet primär bei afterSwap und führt die Gebührenerfassung nach der Preisbestimmung, aber vor dem Abschluss der Transaktion aus.

Definition 3.1 (Gebührenerfassung innerhalb des Swaps)

Ein Mechanismus zur Gebührenerfassung innerhalb des Swaps entnimmt Gebühren F aus dem Swap-Volumen v , sodass gilt: (i) F ist im Quote-Asset denominated, (ii) die Entnahme erfolgt atomar innerhalb der Swap-Ausführung und (iii) es existiert kein zwischenzeitlicher Zustand der Token-Anhäufung.

Der DynamicFeeHook setzt Definition 3.1 um, indem er den Ausgabebetrag abfängt und den Gebührenanteil direkt an die vorgesehenen Empfänger umleitet:

Algorithmus 1: DynamicFeeHook.afterSwap

Eingabe: poolKey, swapParams, delta, hookData

Ausgabe: Modifiziertes delta mit Gebührenerfassung

```

1:  $M \leftarrow \text{ComputeMarketCap}(\text{poolKey})$ 
2:  $f \leftarrow \text{SelectFeeTier}(M, \text{tierConfig})$            ▷ Algorithmus 2
3:  $v \leftarrow |\text{delta.amount1}|$                        ▷ Volumen im Quote-Asset
4:  $F \leftarrow f \cdot v$ 
5: for  $i = 1$  to  $n$  do
6:    $\text{Transfer}(s_i \cdot F, \text{recipients}[i])$ 
7: end for
8:  $\text{delta.amount1} \leftarrow \text{delta.amount1} - F$ 
9: return delta

```

Lemma 3.1 (Atomarität)

Algorithmus 1 wird atomar ausgeführt: Entweder werden alle Gebührenverteilungen abgeschlossen und der modifizierte Swap gelingt, oder die gesamte Transaktion wird ohne Zustandsänderungen rückabgewickelt.

Beweis. Nach der Ausführungssemantik der EVM teilen sich alle Operationen innerhalb einer Transaktion atomares Commit-/Revert-Verhalten. Der Hook-Callback wird im Swap-Transaktionskontext des PoolManagers ausgeführt. Jedes Revert in den Zeilen 5–7 propagiert zur übergeordneten Transaktion. ■

Theorem 3.2 (Kein Verkaufsdruck)

Sei H ein DynamicFeeHook-Pool mit Gebührensatz $f > 0$. Für jede Swap-Sequenz $\{S_1, \dots, S_k\}$, die ein Gesamtvolumen V erzeugt, beträgt der durch die Gebührenmechanik verursachte zusätzliche Verkaufsdruck:

$$\Delta P_{\text{Gebühr}} = 0$$

Beweis. Aus Definition 3.1 erfolgt die Gebührenerfassung im Quote-Asset ohne Token-Anhäufung. Daher gilt $A(t) = 0$ für alle t . Aus Gleichung (2.3) ist der Verkaufsdruck $\Delta P/P \approx -2\Delta x/x$, wobei Δx die verkaufte Token-Menge ist. Da keine Token angehäuft oder verkauft werden ($\Delta x_{\text{Gebühr}} = 0$), folgt $\Delta P_{\text{Gebühr}} = 0$. ■

Korollar 3.1

RexHook löst das Tax-Token-Trilemma (Definition 1.1): positiver Ertrag ($R = fV > 0$), kein Verkaufsdruck (Theorem 3.2) und erlaubnisfreier Betrieb (die Hook-Bereitstellung erfordert keine vertrauenswürdigen Parteien).

4 Gestaltung des dynamischen Gebührenmechanismus

4.1 Gebührenstufen-Funktion

Der DynamicFeeHook implementiert eine stückweise konstante Gebührenfunktion $f: \mathbb{R}^+ \rightarrow [0, f_{\max}]$, parametrisiert durch n Schwellenwerte der Marktkapitalisierung:

$$f(M) = f_i \text{ mit } i = \max\{j : M_j \leq M\} \quad (4.1)$$

Algorithmus 2: SelectFeeTier

Eingabe: Aktuelle Marktkap. M , Stufen $\{(M_1, f_1), \dots, (M_n, f_n)\}$ sortiert nach M_i

Ausgabe: Aktiver Gebührensatz f

Voraussetzung: $M_1 = 0$, $M_i < M_{i+1}$, $f_i > f_{i+1} \forall i$

```

1:  $f \leftarrow f_1$ 
2: for  $i = 2$  to  $n$  do
3:   if  $M \geq M_i$  then  $f \leftarrow f_i$ 
4: end for
5: return  $f$ 

```

Proposition 4.1 (Komplexität)

Algorithmus 2 läuft in $O(n)$ -Zeit und $O(1)$ -Speicher, wobei $n \leq 10$ die Anzahl der konfigurierten Stufen ist. Der gesamte Gas-Overhead ist durch $2.100 + 200n$ Gas-Einheiten beschränkt.

4.2 Optimale Gebührenstaffelung

Wir leiten den optimalen Gebührenplan her, indem wir den erwarteten Lebenszeit-Ertrag unter Beschränkungen der Volumen-Elastizität maximieren.

Annahme 4.1 (Volumen-Elastizität)

Das Handelsvolumen V weist eine konstante Elastizität bezüglich des Gebührensatzes auf: $\partial \ln V / \partial \ln f = -\varepsilon$ mit $\varepsilon > 0$ als Elastizitätsparameter. Empirisch gilt $\varepsilon \in [0,8, 1,5]$ für Meme-Token.

Unter Annahme 4.1 wird der Ertrag $R = fV(f)$ maximiert, wenn:

$$dR/df = V + f \cdot dV/df = V(1 - \varepsilon) = 0 \implies \varepsilon = 1 \quad (4.2)$$

Für $\varepsilon > 1$ (elastische Nachfrage) steigern niedrigere Gebühren den Ertrag. Für $\varepsilon < 1$ (unelastisch) steigern höhere Gebühren den Ertrag. Die optimale Strategie staffelt die Gebühren anhand der Lebenszyklusphase:

Stufe	Marktkap. M_i	Gebühr f_i	Elastizität ε_i	Begründung
1	0 \$	5,00 %	0,6–0,8	Launch: unelastische Nachfrage, FOMO-getrieben
2	100 Tsd. \$	3,00 %	0,9–1,1	Wachstum: nahe der Einheitselastizität
3	500 Tsd. \$	2,00 %	1,0–1,2	Etabliert: Wettbewerbsdruck
4	1 Mio. \$	1,00 %	1,2–1,4	Reif: elastisch, arbitragesensitiv
5	5 Mio. \$	0,50 %	1,3–1,5	Skalierung: Reibung für Volumen minimieren

Tabelle 3. Empfohlene Gebührenstufen-Konfiguration mit Elastizitätsschätzungen. Die Elastizitätsbereiche basieren auf der Standardtheorie der Marktmikrostruktur; eine empirische Validierung nach dem Mainnet-Start wird empfohlen.

Abbildung 2. Dynamische Gebührenökonomie. (a) Stückweise konstante Gebührenfunktion $f(M)$. (b) Ertragskurve mit optimaler Gebühr f^* am Elastizitäts-Übergang. (c) Empirische Elastizität, die mit der Marktkapitalisierung steigt. (d) Ertragsbeitrag je Stufe mit Konzentration in den frühen Phasen mit hoher Gebühr.

5 MEV-Umverteilung: Mechanismus-Design

5.1 MEV-Extraktionsmodell

Maximal Extractable Value (MEV) entsteht aus der Fähigkeit des Block-Proposers, Transaktionen gewinnbringend anzuordnen [4]. Wir konzentrieren uns auf Sandwich-Angriffe, den dominanten MEV-Vektor bei Token-Swaps.

Definition 5.1 (Sandwich-Angriff)

Ein Sandwich-Angriff auf einen Opfer-Swap S_v mit Betrag Δx besteht aus:

T_{front} : Kauf von Δx_f Token vor S_v , verschiebt Preis von P_0 auf P_1

S_v : Opfer-Swap wird zum schlechteren Preis $P_1 > P_0$ ausgeführt

T_{back} : Verkauf von Δx_f Token nach S_v zum Preis $P_2 > P_1$

Angreifergewinn unter Constant-Product-AMM mit Reserven (x, y) :

$$\Pi_{attack} = \Delta x_f \cdot (P_2 - P_1) - c_{gas} - c_{builder} \quad (5.1)$$

wobei c_{gas} die Gas-Kosten und $c_{builder}$ die Zahlung an den Block-Builder sind. Der Verlust des Opfers (Slippage über die Erwartung hinaus) beträgt:

$$L_{victim} = \Delta x \cdot (P_1 - P_0) \approx 2\Delta x \cdot \Delta x_f \cdot P_0/x \quad (5.2)$$

5.2 Erkennung und Erfassung

Die MEV-Engine von RexHook erkennt Sandwich-Muster anhand von On-Chain-Heuristiken, die im beforeSwap-Hook berechnet werden:

Algorithmus 3: MEV-Erkennung

```

Eingabe: Aktueller Swap S, Block-Zustand, Pool-Historie
Ausgabe: MEV-Flag, geschätzte Extraktion E
1: P_twap ← ComputeTWAP(pool, 10 Blöcke)
2: P_current ← GetSpotPrice(pool)
3: σ ← GetHistoricalVolatility(pool, 100 Blöcke)
4: z ← |P_current - P_twap| / σ
5: if z > 2,5 then                                ▷ Schwelle 2,5σ-Abweichung
6:   swapsThisBlock ← GetBlockSwaps(pool)
7:   if |swapsThisBlock| ≥ 2 ∧ OppositeDirections(swapsThisBlock) then
8:     E ← EstimateExtraction(P_current, P_twap, S.amount)
9:     return (TRUE, E)
10:  end if
11: end if
12: return (FALSE, 0)

```

Proposition 5.1 (Optimaler Erfassungssatz)

Sei $\eta \in [0, 1]$ der MEV-Erfassungssatz (Anteil der erkannten Extraktion, der umgeleitet wird). Unter der Annahme eines rationalen Angreifers erfüllt der optimale Erfassungssatz:

$$\eta^* = \min\{1, (c_{gas} + c_{builder}) / E\}$$

Bei η^* ergibt ein Angriff einen erwarteten Gewinn von null und schreckt damit rationale Angreifer ab.

Beweis. Angreifergewinn nach Erfassung: $\Pi' = (1 - \eta)E - c_{gas} - c_{builder}$. Setzt man $\Pi' = 0$: $\eta = 1 - (c_{gas} + c_{builder})/E$. Der minimale abschreckende Erfassungssatz ist somit $\eta^* = (c_{gas} + c_{builder})/E$, gedeckelt bei 1.

5.3 Empirische MEV-Analyse

Kennzahl	Vor Sept. 2022	Nach Merge	2024 Q4
Gesamte Sandwich-Extraktion	675 Mio. \$	1,2 Mrd. \$	89 Mio. \$/Monat
Durchschn. Extraktion pro Angriff	1.847 \$	2.341 \$	1.523 \$
Angriffsrate neuer Token	34 %	47 %	52 %
Medianer Opferverlust	1,2 %	1,8 %	2,1 %

Tabelle 4. MEV-Extraktionsstatistik aus Flashbots- und EigenPhi-Daten. Die Angriffsrate neuer Token ist der Anteil der Launches, die innerhalb von 24 Stunden Sandwich-Angriffe erleben.

6 Spieltheoretische Analyse

6.1 Teilnehmermodell

Wir modellieren das RexHook-Ökosystem als Mehrspieler-Spiel mit vier Teilnehmerklassen:

- **Creator (C):** Stellen Token bereit, konfigurieren Hooks, erhalten Gebührenanteil
- **Trader (T):** Führen Swaps aus, tragen Gebührenkosten
- **Liquiditätsanbieter (LP):** Stellen Kapital bereit, verdienen Swap-Gebühren
- **MEV-Searcher (M):** Versuchen Wertextraktion

Definition 6.1 (Strategieraum)

Jeder Teilnehmer wählt seine Strategie:

Creator: Plattform wählen (Klassisch vs. RexHook) und Gebührenkonfiguration

Trader: Entscheiden, ob und in welchem Volumen gehandelt wird

LPs: Kapitalmenge und Preisspanne für Liquidität wählen

MEV-Searcher: Entscheiden, ob extrahiert wird und in welcher Angriffsgröße

6.2 Auszahlungsanalyse

Wir leiten die Auszahlungen unter RexHook vs. klassischen Mechanismen her:

Teilnehmer	Klassisch (hohes Vol.)	Klassisch (geringes Vol.)	RexHook (hohes Vol.)	RexHook (geringes Vol.)
Creator	Gebühren – Verluste durch Preissturz	Gebühren	Gebührenanteil (stabil)	Gebührenanteil (stabil)
Trader	–Gebühren – MEV – Preisauswirkung	–Gebühren	nur –Gebühren	nur –Gebühren
LP	Gebühren – IL – Preissturz	Gebühren – IL	Gebühren – IL	Gebühren – IL
MEV-Searcher	Extraktionsgewinn	0	0 (abgeschreckt)	0 (abgeschreckt)

Tabelle 5. Auszahlungsvergleich RexHook vs. klassische Tax-Token. IL = Impermanent Loss. RexHook verbessert die Ergebnisse für alle Teilnehmer außer MEV-Searcher, indem es die Verluste der Todesspirale beseitigt.

Theorem 6.1 (Pareto-Verbesserung)

Für jedes Handelsvolumen $V > 0$ erzielt RexHook für alle Teilnehmer außer MEV-Searcher schwach höhere Auszahlungen und für Creator und Trader strikt höhere Auszahlungen, wenn $V > V^*$ (Todesspiralenschwelle aus Theorem 2.1).

Beweis. Aus Tabelle 5 beträgt die Creator-Auszahlung unter RexHook $s_c \cdot fV$ gegenüber $fV - \Delta P \cdot \text{Bestand}$ im klassischen Fall. Für $V > V^*$ gilt $\Delta P \rightarrow -\infty$ (Theorem 2.1), was RexHook trotz $s_c < 1$ strikt besser macht. Die Trader-Auszahlung verbessert sich durch Wegfall der MEV- und ΔP -Terme. Die LP-Auszahlung verbessert sich durch Wegfall des ΔP -Terms. Die MEV-Auszahlung geht gegen null, was kein Verlust ist (die Teilnahme ist optional). ■

6.3 Anreizkompatibilität

Proposition 6.1 (Anreizkompatibilität für Creator)

Ein rationaler Creator bevorzugt RexHook, wenn sein Gebührenanteil die erwarteten Verluste durch Preisstürze der Todesspirale überwiegt. Da 48–60 % der klassischen Token innerhalb von 24 Stunden mit >90 % Preisrückgang scheitern [11,13], macht selbst ein kleiner Gebührenanteil (>5 %) RexHook zur rationalen Wahl.

7 Empirische Validierung

7.1 Literaturüberblick: Token-Ausfallraten

Akademische Forschung und Branchenanalysen belegen die Schwere von Token-Betrug und -Ausfall über EVM-Chains hinweg. Wir fassen Erkenntnisse aus begutachteten Studien und verifizierten Branchenberichten zusammen, um die empirische Grundlage für die Designlogik von RexHook zu schaffen.

Quelle	Datensatz	Wesentlicher Befund	Zeitraum
CertiK Research [11]	100.260 Ethereum-Token	48.265 Rug Pulls (48,14 %)	Okt. 2023 – Aug. 2024
Xia et al. [12]	Uniswap-V2-Token	~50 % zeigen Scam-Merkmale	2021
Certera et al. [13]	Ethereum + BSC	~60 % der Token < 1 Tag aktiv	2023
Kalacheva et al. [14]	Uniswap-V2	>98 % zeigen betrügerische Merkmale	2025
CoinGecko [15]	20,2 Mio. Token (GeckoTerminal)	53,2 % gescheitert	bis 2025

Tabelle 6. Übersicht der Token-Ausfallraten aus begutachteten und Branchenquellen. Die Ausfalldefinitionen variieren: Rug Pulls (expliziter Betrug), Scam-Merkmale (Verhaltensindikatoren) und Inaktivität (Lebensdauer < 1 Tag).

Anmerkung 7.1

Die Streuung der berichteten Ausfallraten (48 %–98 %) spiegelt methodische Unterschiede wider. CertiKs 48,14 % erfassen explizite Rug Pulls mit identifizierbarer Gewinnentnahme. Kalacheva et al.s 98 % umfassen Token mit „betrügerischen Merkmalen“ wie konzentriertem Besitz, fehlender sozialer Präsenz und Code-Mustern, die mit Scams assoziiert sind. Für die Zwecke von RexHook stellt selbst die konservative Rate von 48 % ein kritisches Marktversagen dar, das unser Mechanismus adressiert.

7.2 Analyse der Token-Lebensdauer

Certera et al. [13] (USENIX Security 2023) führten die umfassendste Analyse der Token-Lebensdauer über Ethereum und BSC durch:

Kennzahl	Ethereum	BSC	Implikation
Token < 1 Tag aktiv	~60 %	~60 %	Mehrheit scheitert binnen 24 Std.
1-Tages-Rug-Pulls identifiziert	25.180	332.265	BSC hat 13× mehr Rug Pulls
Erfolgreiche Rug-Pull-Rate	61,9 %	39,1 %	Ethereum-Rug-Pulls effektiver
Gesamte 1-Tages-Rug-Pull-Gewinne	240 Mio. \$ (kombiniert)		Erhebliche Wertentnahme
Token-Spammer (1 % der Adressen)	erzeugen 20–25 %	aller Token	Konzentrierte bösartige Aktivität

Tabelle 7. Statistik zu Token-Lebensdauer und Rug Pulls aus Certera et al. [13]. Die Ein-Tages-Ausfallrate von 60 % stützt die Charakterisierung der Todesspirale aus Theorem 2.1.

Der Befund, dass 60 % der Token weniger als einen Tag aktiv sind, deckt sich mit unserer theoretischen Vorhersage in Theorem 2.1: Sobald die Todesspiralen-Schwelle V^* überschritten wird, kommt es rasch zum Preiskollaps. Die Konzentration der Ausfälle in den ersten 24 Stunden legt nahe, dass bei Token mit

Gebührenmechanismen die Schwelle typischerweise während des anfänglichen Handelsschubs überschritten wird.

7.3 MEV-Extraktionsdaten

Sandwich-Angriffe sind der primäre MEV-Vektor, der Token-Launches betrifft. Wir stellen verifizierte Extraktionsdaten aus Flashbots- und akademischen Quellen zusammen:

Kennzahl	Wert	Quelle
Via MEV extrahiertes ETH (nach Merge)	>330.000 ETH	Flashbots-Dashboard
Sandwich-Anteil am MEV	51,56 % (289,76 Mio. \$ in 2025)	MEV-Research-Aggregatoren
Gesamte Sandwich-Angriffe auf Ethereum	~3,02 Mio.	Chi et al. 2024
Uniswap-V2-Blöcke mit Risiko	>90 %	Akademische Analyse
Tägliche MEV-Erträge (2024)	~300.000 \$/Tag	Flashbots

Tabelle 8. MEV-Extraktionsstatistik aus verifizierten Quellen. Sandwich-Angriffe dominieren mit 51,56 % des gesamten MEV und bestätigen den Fokus des MEV-Umverteilungsmechanismus von RexHook (Abschnitt 5).

7.4 Mechanismus der Todesspirale: empirische Stützung

Obwohl granulare Volumen-Preis-Korrelationsdaten ($\rho(V, -\Delta P)$) speziell für Tax-Token in der publizierten Literatur nicht verfügbar sind, stützen die aggregierten Ausfallstatistiken den Mechanismus von Theorem 2.1 deutlich:

- **Schnelles Scheitern:** 60 % der Token scheitern binnen 1 Tag [13], konsistent mit der Dynamik der Todesspirale, bei der hohes Anfangsvolumen die Rückkopplungsschleife auslöst
- **Gebührenkorrelation:** CertiK stellte fest, dass Token mit komplexer Tokenomics (inkl. Gebührenmechanismen) eine höhere Rug-Pull-Korrelation aufweisen [11]
- **Volumengetriebene Extraktion:** Die 240 Mio. \$ an 1-Tages-Rug-Pull-Gewinnen [13] zeigen, dass hohes Anfangsvolumen rasche Wertentnahme ermöglicht — derselbe Mechanismus, der Todesspiralen antreibt

Anmerkung 7.2 (Einschränkung)

Die direkte Messung der Todesspiralen-Dynamik speziell für Tax-Token erfordert Zugang zu granularen On-Chain-Daten mit Klassifizierung des Gebührenmechanismus. Wir empfehlen künftige Arbeiten, diese Analyse mittels Bytecode-basierter Gebührenerkennung über historische Uniswap-Deployments durchzuführen.

7.5 Prognostizierte Wirkung von RexHook

Basierend auf den empirischen Ausfallraten und Theorem 3.2 (kein Verkaufsdruck) prognostizieren wir die Wirkung von RexHook auf das Token-Überleben. Diese Prognosen nehmen an, dass die Beseitigung des gebühreninduzierten Verkaufsdrucks den primären Auslöser der Todesspirale entfernt, während andere Ausfallarten (explizite Rug Pulls, Liquiditätsabzug) unberührt bleiben.

Kennzahl	Aktueller Stand (empirisch)	RexHook (prognostiziert)	Grundlage
Token-Ausfallrate (24 Std.)	48–60 %	25–35 %	Gebühreninduzierte

Kennzahl	Aktueller Stand (empirisch)	RexHook (prognostiziert)	Grundlage
			Ausfälle beseitigen
MEV-Verlust für Trader	300 Tsd. \$/Tag ökosystemweit	~45 Tsd. \$/Tag (-85 %)	Abschreckung gem. Prop. 5.1
Stabilität der Creator-Erträge	stark volatil (Todesspirale)	stabil (kein Verkaufsdruck)	Theorem 3.2

Tabelle 9. Prognostizierte Wirkung von RexHook basierend auf theoretischen Ergebnissen und empirischen Ausgangswerten. Annahme: gebühreninduzierte Ausfälle machen etwa 50 % der 48–60 % kurzfristigen Ausfallrate aus. Dies sind modellbasierte Schätzungen, die nach dem Launch zu validieren sind.

$$\text{Prognostizierte Überlebensverbesserung} = S_{\text{RexHook}} / S_{\text{klassisch}} = (1 - 0,30) / (1 - 0,55) \approx 1,56 \times \quad (7.1)$$

Diese konservative 1,56-fache Überlebensverbesserung nimmt an, dass RexHook nur gebühreninduzierte Ausfälle beseitigt, während andere Ausfallarten bestehen bleiben. Die tatsächliche Verbesserung könnte höher ausfallen, wenn die reduzierte Preisvolatilität auch den panikgetriebenen Liquiditätsabzug verringert.

7.6 Validierungs-Roadmap

Wir definieren die folgenden empirischen Validierungs-Meilensteine nach dem Launch:

Phase	Kennzahl	Ziel	Zeitraumen
1: Pilot	30-Tage-Überlebensrate	>60 % (vs. 40 % Baseline)	Q2 2026
2: Skalierung	Volumen-Preis-Korrelation	$\rho < 0,2$ (vs. $\sim 0,8$ für Tax-Token)	Q3 2026
3: MEV	Erfolgsrate von Sandwich-Angriffen	<10 % auf RexHook-Pools	Q4 2026

Tabelle 10. Empirische Validierungs-Meilensteine nach dem Mainnet-Start. Vergleichsbasen nutzen zeitgleiche Nicht-RexHook-Token-Launches als Kontrollgruppe.

8 Protokollökonomie

8.1 Das \$REX-Token-Modell

Der \$REX-Token dient als Koordinations- und Wertabschöpfungsmechanismus des Protokolls:

Parameter	Wert	Begründung
Gesamtangebot	1.000.000.000	Festes Angebot, deflationär durch Burns
TGE-Preis	0,04 \$	Startpreis beim Token Generation Event
Vollständig verwässerte Bewertung (FDV)	40.000.000 \$	Marktkap. bei vollständigem Umlauf
Netzwerk	Ethereum L1 + L2-Bridges	V4-native Bereitstellung
Burn-Rate	1 % Deployments + 0,1 % Volumen	Kontinuierliche Angebotsreduktion

8.2 Verteilungsplan

Die Token-Verteilung folgt einem Vesting-Plan, der langfristige Anreize ausrichtet. Das anfängliche zirkulierende Angebot beim TGE beträgt etwa 10,39 %.

Zuteilung	%	Token	TGE	Cliff	Vesting
Pre-Seed (Community)	4,76 %	47.600.000	5 %	3 Mon.	24 Mon. linear
Seed-Runde	5,00 %	50.000.000	5 %	3 Mon.	12 Mon. linear
Private-Runde	4,00 %	40.000.000	5 %	1 Mon.	6 Mon. linear
Strategic-Runde	5,67 %	56.700.000	10 %	0 Mon.	12 Mon. linear
Community-Belohnungen	20,04 %	200.400.000	2 %	—	36 Mon. linear
Ökosystem-Wachstum	15,00 %	150.000.000	2 %	—	36 Mon. linear
Marketing	5,00 %	50.000.000	10 %	—	24 Mon. linear
Liquiditätspool	7,50 %	75.000.000	100 %	—	in DEX/CEX gesperrt
Team	16,00 %	160.000.000	0 %	12 Mon.	36 Mon. linear
Advisors	4,30 %	43.000.000	10 %	6 Mon.	18 Mon. linear
Protokollreserve	12,00 %	120.000.000	0 %	6 Mon.	48 Mon. linear

Tabelle 11. Token-Verteilung mit Vesting-Plänen. Team-Token: 0 % beim TGE, 12-Monats-Cliff, 36-Monats-Vesting. Gesamte Pre-TGE-Finanzierung: 3,8 Mio. \$. Gesamte Investorenverwässerung: 19,43 %.

Berechnung des zirkulierenden Angebots beim TGE

Zuteilung	Token beim TGE
Pre-Seed (5 %)	2.380.000
Seed-Runde (5 %)	2.500.000
Private-Runde (5 %)	2.000.000

Zuteilung	Token beim TGE
Strategic-Runde (10 %)	5.670.000
Community-Belohnungen (2 %)	4.008.000
Ökosystem-Wachstum (2 %)	3.000.000
Marketing (10 %)	5.000.000
Liquiditätspool (100 %)	75.000.000
Advisors (10 %)	4.300.000
Team (0 %)	0
Protokollreserve (0 %)	0
Gesamt beim TGE	103.858.000 (10,39 %)

Tabelle 11a. Aufschlüsselung des zirkulierenden Angebots beim TGE: 10,39 % beim Launch. Team und Protokollreserve haben 0 % beim TGE mit verlängerten Cliffs. Niedriger als der Branchendurchschnitt (15–25 %) [27], was einen konservativen Launch und langfristige Ausrichtung anzeigt.

Wettbewerbsvergleich

Die Tokenomics von RexHook sind konservativer als die führender DeFi-Protokolle, die 2024 gestartet wurden:

- **Uniswap [25]:** 18,04 % an Investoren, 21,27 % an Team = 39,31 % Insider
- **Ethena (2024) [26]:** ~25 % an Investoren, ~30 % an Team = 55 % Insider
- **RexHook:** 19,43 % an Investoren, 16 % an Team, 4,3 % an Advisors = 39,73 % Insider

Die Zuteilung von RexHook ist mit Uniswap vergleichbar und deutlich konservativer als jüngere 2024er-Launches, bei gleichzeitig stärkerer anfänglicher Liquidität (7,5 % vs. typisch 3–5 %) [28].

8.3 Ertragsmodell

Die Protokollerträge stammen aus mehreren Strömen mit unterschiedlichen Risiko-/Ertragsprofilen. Token-Deployments sind kostenlos, um die Akzeptanz zu maximieren.

Strom	Modell	J1-Schätzung	J3-Schätzung
Deployments	Kostenlos (nur Gas)	0 \$	0 \$
Gebührenerfassung (Creator + MEV)	% des Handelsvolumens	1,5 Mio. \$	22,5 Mio. \$
Marktplatz (30 % Anteil)	Erträge aus Drittanbieter-Hooks	200 Tsd. \$	5 Mio. \$
Empfehlungen (30 % Provision)	Anreizpool für Empfehlende	450 Tsd. \$	6,75 Mio. \$
Gesamt		2,15 Mio. \$	34,25 Mio. \$

Tabelle 12. Ertragsprognosen bei 5.000 (J1) → 75.000 (J3) Deployments und 50 Mio. \$ → 750 Mio. \$ monatlichem Volumen. Kostenlose Deployments treiben die Akzeptanz; Erträge werden über Swap-Gebühren erfasst.

8.4 Nutzwert des \$REX-Tokens

Der \$REX-Token ist der native Utility- und Governance-Token des RexHook-Protokolls. Token-Halter profitieren von mehreren Wertschöpfungsmechanismen:

Nutzwert	Mechanismus	Wirkung
Gebührenbeteiligung	Gestaktes \$REX erhält anteiligen Anteil an Protokollgebühren	Direkte Rendite aus dem Plattformvolumen
Governance	Abstimmung über Parameter, Gebührenstruktur, Treasury	Community-getriebene Entwicklung
Premium-Funktionen	Zugang zu fortgeschrittenen Hooks, Priorität, Custom-Konfig.	Differenzierte Creator-Stufen
Marktplatz	Erforderlich zum Listen von Drittanbieter-Hooks, reduzierte Gebühren	Anreiz zur Ökosystem-Teilnahme
Deflationärer Druck	1 % des Deployment-Gas + 0,1 % des Volumens verbrannt	Kontinuierliche Angebotsreduktion

Tabelle 13. Nutzwertmechanismen und Werttreiber des \$REX-Tokens.

8.5 Staking-Mechanik

\$REX-Staker erhalten einen anteiligen Anteil am Protokollertrag, ausgeschüttet in ETH. Ihr Anteil hängt davon ab, wie viel Sie staken und wie lange — längeres Staking verdient mehr pro Token, um Gaming zu verhindern.

Staking-Stufe	Mindestmenge	Sperrfrist	Gebühren-Multiplikator
Bronze	10.000 \$REX	keine Sperre	1,0×
Silber	100.000 \$REX	3 Monate	1,25×
Gold	500.000 \$REX	6 Monate	1,5×
Platin	1.000.000 \$REX	12 Monate	2,0×

Tabelle 14. Staking-Stufen mit Sperrfristen und Gebühren-Multiplikatoren.

8.6 Hook-Marktplatz: Der App-Store für Token-Launches

Man kann sich RexHook als Shopify für Token-Launches vorstellen. Shopify gibt die Infrastruktur, einen Online-Shop zu starten; RexHook gibt die Infrastruktur, einen Token zu starten. Shopify hat einen App-Store, in dem Entwickler Plugins bauen, um dem Shop Funktionen hinzuzufügen; RexHook hat einen Hook-Marktplatz, in dem Entwickler Hooks bauen, um dem Pool Funktionen hinzuzufügen.

Die Analogie

Shopify-Shop = Dein Token-Pool | Shopify-Apps = RexProtocol-Hooks | App-Store = Hook-Marktplatz

Abbildung 5. Workflow des Hook-Marktplatzes. (a) Entwickler reichen Hooks mit Stake-Sicherheit ein; ein Sicherheits-Audit verifiziert die Korrektheit. (b) Creator wählen das Hook-Set H und stellen den Pool bereit. (c) Swap-Gebühren F werden gemäß dem Anteilsvektor (s_c, s_p, s_h) verteilt, wobei $s_c + s_p + s_h = 1$.

Multi-Pool-Architektur: gleicher Token, verschiedene Hooks

Eine zentrale Innovation von Uniswap V4 ist, dass derselbe Token mehrere Pools mit unterschiedlichen Hooks haben kann. Dies ermöglicht ausgefeilte Handelsstrategien und Marktsegmentierung:

Pool	Hooks (H)	Zweck
P1	{DynamicFee, MEVCapture}	Öffentlicher Handel, $f(M) \in [0,5\%, 5\%]$
P2	{Whitelist, Vesting}	Privat/Investoren, $addr \in W, unlock(t)$
P3	{AntiSniper, MaxWallet}	Fairer Launch, $block \leq N, bal \leq \max$
P4	{Points, NFTLottery}	Gamifizierter Handel, $rewards(V), raffle()$
P5	{TWAMM, Oracle}	Institutionell, Δt -Ausführung, P_oracle

Abbildung 6. Multi-Pool-Architektur für Token T. Jeder Pool P_i bindet ein eigenes Hook-Set H_i . Kern-Hooks: P1 öffentlicher Handel, P2 private Investoren, P3 Fair-Launch-Schutz. Drittanbieter-Hooks: P4 gamifizierte Belohnungen, P5 institutionelles TWAMM mit Oracle-Preisbildung. Arbitrage zwischen Pools hält die Preiskonsistenz über alle Konfigurationen hinweg aufrecht.

Anmerkung 8.1 (Vorteile von Multi-Pools)

Diese Architektur ermöglicht: (i) Marktsegmentierung — unterschiedliche Gebührenstrukturen für Retail- vs. Whale-Trader, (ii) A/B-Tests — verschiedene Hook-Kombinationen zur Ertragsoptimierung erproben, (iii) spezialisierte Pools — Private-Runden, Vesting-Pools oder rein institutioneller Zugang, (iv) Arbitrage — Preisunterschiede zwischen Pools schaffen Arbitragegelegenheiten, von denen LPs profitieren.

Kern-Hooks (von RexHook gebaut)

Praxiserprobte Hooks, die mit der Plattform ausgeliefert werden. Token-Creator können jede Kombination aktivieren:

Hook	Funktion
DynamicFeeHook	Passt Gebühren automatisch an die Marktkapitalisierung an (früh hoch, bei Skalierung niedrig)
MEVCaptureHook	Erkennt Sandwich-Angriffe, erfasst den Gewinn, verteilt ihn an den Creator um
AntiSniperHook	Bestraft Bot-Sniping in den ersten N Blöcken nach dem Launch
PriceImpactHook	Erfasst übermäßige Slippage als zusätzliche Gebühr
VestingHook	Erzwingt On-Chain-Vesting-Pläne für Team-/Investoren-Token
WhitelistHook	Beschränkt frühen Handel auf zugelassene Adressen
MaxWalletHook	Begrenzt den maximalen Token-Bestand pro Wallet
CooldownHook	Erzwingt eine Mindestzeit zwischen Verkäufen pro Wallet

Tabelle 14b. Mit RexHook enthaltene Kern-Hooks. Alle kombinierbar — jede Kombination nutzbar.

Drittanbieter-Hooks (Community-Ökosystem)

Entwickler können eigene Hooks bauen und monetarisieren. Im V4-Ökosystem bereits gebaute Beispiele:

Hook	Beschreibung	Anwendungsfall
Points / Rewards	Prägt Belohnungs-Token basierend auf	Airdrop-Kampagnen, Trader-Anreize

Hook	Beschreibung	Anwendungsfall
	Handelsvolumen	
NFT-Lotterie	Gamifiziertes NFT-Minting + Chainlink-VRF-Verlosungen	Virale Interaktion, Community-Aufbau
Prediction Market	Binäre Optionen auf die Pool-Preisrichtung	Spekulation, Engagement
Auto-Compound	Reinvestiert LP-Gebühren automatisch	Höhere effektive APY für LPs
Yield Harvesting	Wrappt ertragstragende Token (Aave, ERC4626)	LPs verdienen Swap-Gebühren + Lending-APY
JIT-Rebalancer	Just-in-Time-Liquidität für große Swaps	Null Slippage für Whales
Limit-Orders	On-Chain-Limit-Orders via Hook	CEX-ähnliche Handelserfahrung
TWAMM	Time-Weighted Average Market Maker	Große Orders über Zeit, reduzierte Auswirkung
Oracle-Integration	Chainlink/Pyth-Preisfeeds für dynamische Preise	Peg-Stabilität, Arbitragereduktion
Referral-Tracking	On-Chain-Empfehlungs-Attribution und -Provision	Virales Wachstum, Affiliate-Marketing

Tabelle 14c. Beispiele bereits im Uniswap-V4-Ökosystem gebauter Hooks. Der RexHook-Marktplatz kuratiert, auditiert und vertreibt diese an Token-Creator.

Hinweis: Über 100 Hooks existieren bereits in der V4-Entwickler-Community. Der Marktplatz kuratiert die besten, übernimmt die Sicherheitsprüfung und macht sie für Token-Creator mit einem Klick installierbar.

Marktplatz-Ökonomie

Der Hook-Marktplatz schafft einen dreiseitigen Markt:

Teilnehmer	Rolle	Anreiz
Token-Creator	Hooks durchsuchen und installieren, um den Pool anzupassen	Bessere Tokenomics → mehr Volumen → mehr Ertrag
Hook-Entwickler	Hooks bauen, Preise festlegen, an Installs verdienen	70 % des Hook-Ertrags (RexHook erhält 30 %)
RexHook	Plattform, Kuratierung, Sicherheits-Audits, Vertrieb	30 % Marktplatzanteil + höhere Plattformnutzung

Tabelle 14d. Teilnehmer und Anreize des Hook-Marktplatzes.

Entwicklerprogramm

Zur Wahrung von Qualität und Sicherheit müssen Hook-Entwickler:

- \$REX staken, um Hooks zu listen (wird bei böartigem Verhalten gelasht)
- die Sicherheitsprüfung für hervorgehobene Platzierung bestehen
- Dokumentation und Support pflegen
- Erträge mit RexHook teilen (70/30-Aufteilung zugunsten des Entwicklers)

Anmerkung 8.2 (Kombinierbarkeit)

Hooks sind kombinierbar. Ein Token kann gleichzeitig DynamicFeeHook + AntiSniperHook + MaxWalletHook + einen Drittanbieter-Points-Hook ausführen. Das schafft exponentielle Funktionskombinationen aus einem linearen Hook-Bestand — derselbe Netzwerkeffekt, der App-Ökosysteme wertvoll gemacht hat.

8.7 Launchpad-Ökonomie & Geschäftsmodell

Dieser Abschnitt liefert detaillierte Finanzprognosen für die RexHook-Plattform, einschließlich Volumenannahmen, Gebührenberechnungen und Sensitivitätsanalyse. Alle Prognosen beruhen auf beobachtbaren Marktdaten und konservativen Wachstumsannahmen.

Marktdimensionierung

Der Token-Launch-Markt auf EVM-Chains ist beträchtlich und wachsend. Referenzwerte:

Kennzahl	Wert	Quelle
Tägliche neue Token (Ethereum)	~370/Tag	CertiK [11]
Token-Ausfälle Q4 2025	~83.700/Tag	CoinGecko [15]
Pump.fun kumulierter Ertrag	500 Mio. \$+ (Solana)	Öffentliche Dashboards
Tägliche MEV-Extraktion	300 Tsd.–500 Tsd. \$	Flashbots

Tabelle 15. Referenzdaten zum Markt der Token-Launch-Ökonomie.

Volumenprognosen

Wir modellieren drei Szenarien basierend auf Deployment-Anzahl und durchschnittlichem Volumen pro Token:

Szenario	J1 Deploy.	J1 Mon.-Vol.	J3 Deploy.	J3 Mon.-Vol.
Konservativ	2.500	25 Mio. \$	25.000	250 Mio. \$
Basisfall	5.000	50 Mio. \$	75.000	750 Mio. \$
Optimistisch	10.000	100 Mio. \$	150.000	1,5 Mrd. \$

Tabelle 16. Deployment- und Volumenprognosen nach Szenario.

Gebührenstruktur & Ertragsberechnung

Die Protokollerträge stammen aus: (1) einem kleinen Anteil der vom Creator konfigurierten Handelsgebühren und (2) 50 % des zusätzlich erfassten Werts aus MEV, Preisauswirkung und Bot-Strafen. Für Modellzwecke schätzen wir einen effektiven Protokoll-Erfassungssatz von ~2,5 % des Gesamtvolumens:

Jahr	Mon.-Volumen	Jahresvolumen	Geb.-Aktivität (5 %)	Protokollertrag
J1	50 Mio. \$	600 Mio. \$	30 Mio. \$	15 Mio. \$
J2	200 Mio. \$	2,4 Mrd. \$	120 Mio. \$	60 Mio. \$
J3	750 Mio. \$	9 Mrd. \$	450 Mio. \$	225 Mio. \$

Tabelle 17. Ertragsprognosen im Basisfall. Hinweis: Die frühere Tabelle 12 zeigt konservative Schätzungen mit geringerem Volumenanstieg.

$$\text{Protokollertrag} = \text{Volumen} \times \text{effektiver Protokollsatz} \approx V \times 0,025 \quad (8.3)$$

Stückkosten pro Token

Durchschnittliche Ökonomie eines auf RexHook bereitgestellten Tokens:

Kennzahl	Wert	Berechnung
Durchschn. Lebenszeitvolumen pro Token	120.000 \$	600 Mio. \$ jährl. / 5.000 Token
Durchschn. erzeugte Gebühren pro Token	6.000 \$	120 Tsd. \$ × 5 % Gebühr
Creator-Ertrag pro Token	4.500 \$	Großteil der Gebühren + 50 % Zusatzerfassung
Protokollertrag pro Token	1.500 \$	Gebührenanteil + 50 % MEV-/Auswirkungserfassung
Deployment-Kosten	0 \$	Kostenlos (nur Gas)

Tabelle 18. Stückkosten pro Token-Deployment (J1, Basisfall).

Sensitivitätsanalyse

Sensitivität des Protokollertrags gegenüber Schlüsselvariablen (J1-Basisfall = 15 Mio. \$):

Variable	-50 %	Basis	+50 %	+100 %
Monatsvolumen	7,5 Mio. \$	15 Mio. \$	22,5 Mio. \$	30 Mio. \$
Durchschn. Gebührensatz (3–7 %)	9 Mio. \$	15 Mio. \$	21 Mio. \$	—
Deployment-Anzahl	7,5 Mio. \$	15 Mio. \$	22,5 Mio. \$	30 Mio. \$

Tabelle 19. Ertragssensitivität gegenüber Schlüsselannahmen.

Wettbewerbspositionierung

RexHook adressiert den EVM-Token-Launch-Markt, der im Vergleich zu Solana derzeit unterversorgt ist:

Plattform	Chain	Modell	Geschätzter Ertrag
Pump.fun	Solana	1 % Gebühr + Graduierungsgebühr	500 Mio. \$+ kumuliert
Sun.pump	Tron	Ähnlich Pump.fun	10 Mio. \$+ kumuliert
RexHook	Ethereum + L2s	Gebührenanteil + MEV-Erfassung	15 Mio. \$ J1 prognostiziert

Tabelle 20. Wettbewerbslandschaft der Token-Launch-Plattformen.

Wesentliche Annahmen & Risiken

Diese Prognosen nehmen an:

- Die Akzeptanz von Uniswap V4 wächst weiter auf Ethereum und L2s
- Die Token-Launch-Aktivität bleibt robust (makroabhängig)
- Keine wesentlichen regulatorischen Änderungen, die Token-Launches betreffen
- Wettbewerbsdifferenzierung (Lösung der Todesspirale) treibt die Akzeptanz

- Durchschnittlicher Gebührensatz von 5 % über alle Marktkap.-Stufen

Anmerkung 8.1 (Konservativ vs. Aggressiv)

Tabelle 12 (früher in diesem Abschnitt) nutzt konservativere Volumenannahmen (50 Mio. \$ monatlich), während Tabelle 17 denselben Basisfall mit voller Gebührenberechnung zeigt. Die Diskrepanz spiegelt die inhärente Unsicherheit der Volumenprognosen für eine neue Plattform wider.

9 Sicherheitsanalyse

9.1 Bedrohungsmodell

Wir betrachten Angreifer mit folgenden Fähigkeiten:

- **A1 (MEV-Searcher):** Kann den Mempool beobachten, Bundles an Block-Builder senden
- **A2 (Bösartiger Creator):** Kontrolliert die Parameter der Hook-Bereitstellung
- **A3 (Versierter Angreifer):** Kann beliebige Verträge bereitstellen, Flash-Loans nutzen

9.2 Sicherheitseigenschaften

Theorem 9.1 (Unveränderlichkeit)

Für jeden RexHook-Hook H , der zum Zeitpunkt t_0 mit Parametern θ bereitgestellt wird, gilt für alle $t > t_0$: $\text{parameters}(H, t) = \theta$ mit Wahrscheinlichkeit 1.

Beweis. Hook-Verträge enthalten keine administrativen Funktionen. Alle Parameter werden im Konstruktor gesetzt und als immutable- oder constant-Variablen gespeichert. Die EVM bietet keinen Mechanismus, um immutable-Speicher zu verändern. ■

Theorem 9.2 (Gebührenobergrenze)

Für jeden Swap auf einem RexHook-Pool gilt für die entnommene Gebühr $F \leq f_{\text{max}} \cdot v$, wobei $f_{\text{max}} = 0,30$ das Protokollmaximum und v das Swap-Volumen ist.

Beweis. Der Hook-Vertrag erzwingt $f \leq f_{\text{max}}$ per require-Anweisung im Konstruktor. Die Gebührenberechnung $F = f \cdot v$ verwendet SafeMath zur Überlaufvermeidung. Kein Codepfad lässt $F > f_{\text{max}} \cdot v$ zu. ■

9.3 Audit und Verifizierung

Komponente	Methode	Abdeckung	Status
DynamicFeeHook	Externes Audit + formale Verifizierung	100 %	Abgeschlossen
FeeDistribution	Externes Audit	100 %	Abgeschlossen
MEV-Engine	Externes Audit + Fuzzing	95 %	Abgeschlossen
Integrationstests	Foundry + Echidna	>95 % Zeilen	Laufend

Tabelle 13. Sicherheits-Verifizierungsstatus der Kernkomponenten des Protokolls.

9.4 Bug-Bounty

Schweregrad	Beschreibung	Belohnung
Kritisch	Verlust von Geldern, willkürliche Gebührenentnahme	50.000–100.000 \$
Hoch	Erhebliche Protokollstörung	10.000–50.000 \$
Mittel	Schwachstellen mit begrenzter Wirkung	2.000–10.000 \$
Niedrig	Informative Befunde	500–2.000 \$

9.5 Hook-Zertifizierung & Registry-Infrastruktur

Im Uniswap-V4-Ökosystem besteht eine kritische Lücke: Es existiert keine offizielle Hook-Registry und kein Zertifizierungssystem. Die Uniswap Foundation hat Datenstandards [16] vorgeschlagen, aber keine Verifizierungsinfrastruktur bereitgestellt. RexHook schließt diese Lücke durch eine On-Chain-Registry, die jedem abfragenden System verifizierbare Hook-Metadaten bereitstellt.

Das V4-Hook-Verifizierungsproblem

Uniswap-V4-Hooks bringen Verifizierungsherausforderungen mit sich, die sich grundlegend von der ERC-20-Token-Analyse unterscheiden:

Definition 9.2 (Anforderungen an die Hook-Verifizierung)

Ein V4-Hook H erfordert die Verifizierung von: (i) Flag-Kodierung — die 14 Berechtigungsbits, kodiert in den Adressbits [0:13], müssen mit der deklarierten Funktionalität übereinstimmen, (ii) Event-Konformität — Emission von Standard-Events (HookSwap, HookFee, HookModifyLiquidity, HookBonus) für korrekte Indizierung, (iii) Quellcode-Verifizierung — der Bytecode stimmt mit dem verifizierten Quellcode überein, (iv) Unveränderlichkeit — keine Proxy-Muster oder Admin-Funktionen, die das Verhalten nach der Bereitstellung ändern könnten.

Diese Anforderungen unterscheiden sich von der ERC-20-Analyse (Honeypot-Erkennung, Mint-Funktionen usw.) und erfordern eine dedizierte Verifizierungsinfrastruktur.

Theorem 9.3 (Verifizierung der Flag-Kodierung)

Für jeden an Adresse a bereitgestellten Hook-Vertrag H sind die Berechtigungs-Flags $F = \{f_0, \dots, f_{13}\}$ allein aus a deterministisch verifizierbar, wobei $f_i = (a \gg i) \& 1$.

Beweis. Nach der Uniswap-V4-Spezifikation [3] sind Hook-Berechtigungen in den letzten 14 Bits der Vertragsadresse kodiert. Der PoolManager validiert diese Bits bei der Pool-Initialisierung über `Hooks.validateHookPermissions()`. Da Adressen nach der Bereitstellung unveränderlich sind und durch das CREATE2-Salt bestimmt werden, können die Flags nicht verändert werden. Jeder Prüfer kann die Flags durch bitweise Operationen auf der Adresse extrahieren. ■

RexHook-Registry-Vertrag

RexHook stellt einen On-Chain-Registry-Vertrag bereit, der verifizierte Hook-Metadaten speichert. Die Registry ist für Lesezugriffe erlaubnisfrei, sodass jedes System den Hook-Verifizierungsstatus abfragen kann:

Schnittstelle: IRexHookRegistry

```
struct HookMetadata {
    address hookAddress;
    string name;
    string version;
    string sourceCodeUri; // Etherscan/Sourcify verifiziert
    string auditReportUri; // Link zum Audit-Bericht
    address[] auditors; // Verifizierte Auditor-Adressen
    uint256 deployedAt;
    bytes32 codeHash; // keccak256 des verifizierten Bytecodes
    bool emitsStandardEvents; // Implementiert UF-Events
    uint160 hookFlags; // Welche der 14 Flags aktiviert
}
function registerHook(HookMetadata metadata) external;
function verifyHook(address hook) external view returns (bool, string);
function getHookMetadata(address hook) external view returns (HookMetadata);
```

Standard-Events der Uniswap Foundation

Alle RexHook-zertifizierten Hooks müssen die vier von der Uniswap Foundation vorgeschlagenen Standard-Events [16] emittieren, um eine korrekte Indizierung durch Envio und andere Datenanbieter zu gewährleisten:

Standard-Hook-Events (verpflichtend für Zertifizierung)

```
event HookSwap(bytes32 indexed id, address indexed sender,
    int128 amount0, int128 amount1,
    uint128 hookLPfeeAmount0, uint128 hookLPfeeAmount1);
event HookFee(bytes32 indexed id, address indexed sender,
    uint128 feeAmount0, uint128 feeAmount1);
event HookModifyLiquidity(bytes32 indexed id, address indexed sender,
    int128 amount0, int128 amount1);
event HookBonus(bytes32 indexed id, uint128 amount0, uint128 amount1);
```

Zertifizierungsstufen

Über RexHook bereitgestellte Hooks erhalten eine Zertifizierung je nach Verifizierungstiefe:





Stufe	Anforderungen	Scanner-Anzeige	Badge
Bronze	Quelle verifiziert auf Etherscan/Sourcify	„Verified Source“	
Silber	+ erbt OpenZeppelin BaseHook + Standard-Events	„RexHook Compliant“	
Gold	+ unveränderliche Parameter + kein Proxy	„RexHook Certified“	
Platin	+ externes Audit (OpenZeppelin/Hacken/etc.)	„RexHook Audited“	

Tabelle 13c. Hook-Zertifizierungsstufen und Anforderungen. Höhere Stufen schalten bevorzugte Marktplatzplatzierung und reduzierte Protokollgebühren frei.

Zertifizierungs-Checkliste

Die folgenden Prüfungen bestimmen die Zertifizierungsstufe und den Scanner-Risikowert:

Prüfung	Anforderung	Scanner-Auswirkung
Quelle verifiziert	Etherscan- + Sourcify-Verifizierung	Nicht als „unverifiziert“ markiert
Standard-Events	Emittiert 4 Uniswap-Foundation-Events	Korrekte Indizierung auf v4.xyz
OpenZeppelin-Basis	Erbt vom auditierten BaseHook	Erbt Audit-Abdeckung
Keine Admin-Keys	Alle Parameter unveränderlich	Nicht als „owner can modify“ markiert
Kein Proxy	Direkte Bereitstellung, nicht aktualisierbar	Nicht als „upgradeable“ markiert
CREATE2-bereitgestellt	Deterministische Adresse mit korrekten Flags	Flag-Kodierung verifiziert
In Registry gelistet	On-Chain-Metadaten in RexHook-Registry	Metadaten für Scanner verfügbar

Tabelle 13d. Hook-Zertifizierungs-Checkliste. Jede Prüfung trägt zu einem zusammengesetzten Risikowert (0–1) bei, der Scanner-Integrationen bereitgestellt wird.

Scanner-Integrations-API

RexHook stellt eine öffentliche API für die Integration von Scannern und Aggregatoren bereit:

RexHook-Zertifizierungs-API

```
GET /api/v1/hooks/{address}/verify
Antwort: {
  "verified": true,
  "certificationLevel": "gold", // bronze|silver|gold|platinum
  "checks": {
    "sourceVerified": true,
    "standardEvents": true,
    "immutableParams": true,
    "noProxy": true,
    "audited": false
  },
  "riskScore": 0.05, // Skala 0-1 für Scanner-Integration
  "metadata": { ... }
}
```

Standards-Kompatibilität

Die Zertifizierungsinfrastruktur von RexHook ist auf Kompatibilität mit bestehenden Standards der Sicherheitsinfrastruktur ausgelegt. Wir beweisen, dass jede konforme Integration null Änderungen an bestehenden Scanner-Architekturen erfordert:

Theorem 9.4 (Standards-Kompatibilität)

Sei S ein beliebiger Token-Sicherheitsscanner, der den On-Chain-Vertragszustand über Standard-JSON-RPC-Aufrufe abfragt. Der RexHook-Registry-Vertrag R stellt Verifizierungsdaten über öffentliche view-Funktionen bereit und erfordert für die Integration keine Protokolländerungen an S.

Beweis. Der Registry-Vertrag implementiert nur standardmäßige öffentliche Solidity-view-Funktionen: `verifyHook(address) → (bool, string)` und `getHookMetadata(address) → HookMetadata`. Diese sind über die JSON-RPC-Methode `eth_call` aufrufbar, die von jeder EVM-kompatiblen Infrastruktur universell unterstützt wird. Es sind keine eigenen Protokolle, Oracles oder Off-Chain-Komponenten erforderlich. Der Scanner S muss lediglich einen einzigen `eth_call` hinzufügen, um R an einer bekannten Adresse abzufragen. ■

Definition 9.3 (RexHook-zertifizierter Hook)

Ein Hook H ist RexHook-zertifiziert genau dann, wenn: (i) H über CREATE2 mit korrekter Flag-Kodierung in den Adressbits [0:13] bereitgestellt wurde, (ii) H von OpenZeppelin BaseHook [17] erbt, (iii) H alle vier Standard-Hook-Events gemäß [16] emittiert, (iv) H verifizierten Quellcode auf Etherscan und Sourcify hat, (v) H keine Proxy- oder Upgrade-Mechanismen enthält und (vi) H im RexHook-Registry-Vertrag registriert ist.

Theorem 9.5 (Verifizierungsvollständigkeit)

Für jeden RexHook-zertifizierten Hook H sind alle Verifizierungseigenschaften unabhängig On-Chain verifizierbar, ohne vertrauenswürdige Dritte.

Beweis. Wir zeigen, dass jede Eigenschaft On-Chain verifizierbar ist: (i) Flag-Kodierung: Aus den Adressbits extrahieren und mit dem Rückgabewert von `getHookPermissions()` vergleichen. (ii) BaseHook-Vererbung: über `supportsInterface()` oder Bytecode-Analyse verifizieren. (iii) Standard-Events: Transaktions-Logs nach Event-Signaturen gemäß [16]-ABI parsen. (iv) Quellcode-Verifizierung: Etherscan-/Sourcify-APIs abfragen (deterministisch bei gegebener Adresse + Chain). (v) Kein Proxy: auf `DELEGATECALL`-Opcodes und `implementation()`-Funktionen im Bytecode prüfen. (vi) Registry-Listung: `RexHookRegistry.isRegistered(H)` aufrufen. Alle Prüfungen erfordern nur öffentliche Blockchain-Daten und Standard-RPC-Aufrufe. ■

Theorem 9.6 (Indexer-Kompatibilität)

Jeder Hook H, der die Standard-Events der Uniswap Foundation [16] emittiert, erzeugt Logs, die von jedem EVM-Indexer I, der das Standard-Event-ABI implementiert, geparkt werden können — ohne benutzerdefinierte Konfiguration für I.

Beweis. Die vier Standard-Events (HookSwap, HookFee, HookModifyLiquidity, HookBonus) verwenden feste Event-Signaturen mit indizierten Parametern (Pool-ID, Sender). Jeder Indexer, der die Hook-Adresse überwacht, erfasst diese Logs über den Standard-RPC `eth_getLogs`. Das Event-ABI ist öffentlich spezifiziert [16] und erfordert nur eine ABI-Konfiguration in I — keine Codeänderungen. ■

Bereitstellungs-Template

RexHook stellt ein standardisiertes Bereitstellungs-Template bereit, das sicherstellt, dass alle Hooks die Zertifizierungsanforderungen erfüllen:

RexHook-zertifiziertes Hook-Template (Solidity)

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.26;
import {BaseHook} from "@openseppelin/uniswap-hooks/src/base/BaseHook.sol";
import {IHookEvents} from "@openseppelin/uniswap-hooks/src/interfaces/IHookEvents.sol";
abstract contract RexHookCertifiedBase is BaseHook, IHookEvents {
    string public constant REXHOOK_VERSION = "1.0.0";
    address public immutable REXHOOK_REGISTRY;
    constructor(IPoolManager _pm, address _registry) BaseHook(_pm) {
        REXHOOK_REGISTRY = _registry;
        _registerWithRexHook();
    }
    // Auto-Registrierung bei der RexHook-Registry bei Bereitstellung
    function _registerWithRexHook() internal;
}
```

9.6 ERC-8004-Kompatibilitätsschicht (optional)

ERC-8004 („Trustless Agents“) ist ein Ethereum-Standard, der universelle Auffindbarkeits-, Reputations- und Validierungsinfrastruktur für autonome Agenten bereitstellt. Während die RexHook-Registry das primäre Zertifizierungssystem für alle Hooks auf unserer Plattform ist, bieten wir optionale ERC-8004-Kompatibilität für Hook-Entwickler, die eine breitere Ökosystem-Integration wünschen.

Anmerkung 9.1 (Optionale Integration)

Eine ERC-8004-Registrierung ist für Hooks auf RexHook nicht erforderlich. Die RexHook-Registry bleibt die maßgebliche Quelle für die Hook-Zertifizierung. Die ERC-8004-Integration ist eine optionale Funktion für Entwickler, die universelle Auffindbarkeit, portable Reputation oder KI-Agenten-Kompatibilität anstreben.

Was ist ERC-8004?

ERC-8004 definiert drei leichtgewichtige On-Chain-Registries, die als Singletons pro Chain bereitgestellt werden:

Registry	Zweck	RexHook-Anwendung
Identity Registry	ERC-721-NFT pro Agent mit Metadaten-URI	Hook erhält universellen Identifier, auffindbar durch jeden ERC-8004-Client

Registry	Zweck	RexHook-Anwendung
Reputation Registry	On-Chain-Feedback-Werte von Clients	Pool-Creator bewerten Hooks; Reputation plattformübergreifend portabel
Validation Registry	Unabhängige Verifizierungsnachweise	Audit-Firmen reichen Verifizierungsnachweise On-Chain ein

Tabelle 13e. ERC-8004-Registries und ihre Anwendung auf die Hook-Zertifizierung.

Integrationsarchitektur

In RexHook registrierte Hooks können optional mit einer ERC-8004-Agentenidentität verknüpft werden:

Abbildung 7. Hook-Registrierungsablauf. Alle Hooks werden in der RexHook-Registry registriert (verpflichtend). Entwickler können optional zu 100 % per Opt-in mit ERC-8004 für universelle Auffindbarkeit und portable Reputation verknüpfen. Vorteile: universelle Auffindbarkeit, portable Reputation, KI-Agenten-Zugang, On-Chain-Validierung.

Anwendungsfälle für die ERC-8004-Integration

Anwendungsfall	Ohne ERC-8004	Mit ERC-8004
DEX-Aggregator-Auffindbarkeit	Muss spezifisch die RexHook-API integrieren	Fragt die Standard-ERC-8004-Registry ab
KI-Trading-Agent	Kein Standard-Auffindungsmechanismus	Findet Hooks über ERC-8004-Identität
Entwickler-Reputation	Reputation an RexHook-Plattform gebunden	Reputation über alle Plattformen portabel
Audit-Verifizierung	Audit-Bericht Off-Chain verlinkt	Auditor reicht On-Chain-Validierungsnachweis ein
Plattformübergreifende Hooks	Separate Registrierung pro Plattform	Einzelne Identität über das gesamte Ökosystem

Tabelle 13f. Anwendungsfälle, in denen die ERC-8004-Integration zusätzlichen Wert bietet.

Erweiterte Registry-Schnittstelle

Der RexHook-Registry-Vertrag bietet optionale ERC-8004-Verknüpfungsfunktionen:

Schnittstelle: IRexHookRegistry (ERC-8004-Erweiterung)

```
// Kern-Registrierung (immer erforderlich)
function registerHook(HookMetadata metadata) external;
// Optionale ERC-8004-Verknüpfung
function linkToERC8004(address hook, uint256 agentId) external;
function getERC8004AgentId(address hook) external view returns (uint256);
// Komfort: Hook + ERC-8004-Agent atomar registrieren
function registerWithERC8004(
    HookMetadata metadata,
    string agentURI
) external returns (uint256 agentId);
```

Schema der Agenten-Registrierungsdatei

Mit ERC-8004 registrierte Hooks verwenden eine standardisierte Agenten-Registrierungsdatei:

ERC-8004-Agentenregistrierung (Hook)

```
{
  "type": "https://eips.ethereum.org/EIPS/eip-8004#registration-v1",
  "name": "RexHook DynamicFeeHook v1.0.0",
  "description": "Dynamic fee capture hook with market-cap-based tiers...",
  "image": "ipfs://Qm.../hook-icon.png",
  "services": [{
    "name": "UniswapV4Hook",
    "endpoint": "0x1234...abcd", // hook address
    "version": "1.0.0",
    "chainId": "1"
  }],
  "supportedTrust": ["reputation", "crypto-economic"],
  "hookMetadata": { // RexHook-specific extension
    "flags": ["beforeSwap", "afterSwap"],
    "rexhookCertification": "gold",
    "codeHash": "0xabc123...",
    "auditReport": "ipfs://Qm..."
  }
}
```

Definition 9.4 (Veränderlichkeit der ERC-8004-Agenten-Metadaten)

Die URI der Agenten-Registrierungsdatei kann vom Agenten-Eigentümer über `setAgentURI(agentId, newURI)` aktualisiert werden. Dies ermöglicht es Hooks, ihre Metadaten zu aktualisieren, wenn: (i) neue Audit-Berichte veröffentlicht werden, (ii) sich die Zertifizierungsstufe ändert, (iii) zusätzliche Chain-Deployments erfolgen oder (iv) Hook-Parameter dokumentiert werden.

Reputationsfluss

Wenn die ERC-8004-Integration aktiviert ist, können Pool-Ersteller On-Chain-Feedback einreichen:

ERC-8004-Reputations-Feedback (Beispiel)

```
// Pool creator rates a hook after using it
reputationRegistry.giveFeedback(
  agentId: 42, // Hook's ERC-8004 ID
  value: 95, // Score (0-100)
  valueDecimals: 0,
  tag1: "reliability", // Category
  tag2: "DynamicFeeHook",
  endpoint: "0x1234...", // Hook address used
  feedbackURI: "",
  feedbackHash: bytes32(0)
);
```

Diese Reputation ist von jedem ERC-8004-Client abfragbar und ermöglicht es externen Systemen (DEX-Aggregatoren, KI-Agenten, Sicherheitsscannern), die Vertrauenswürdigkeit eines Hooks ohne RexHook-spezifische Integration zu bewerten.

Theorem 9.7 (Reputationsportabilität)

Für jeden mit dem ERC-8004-Agenten A verknüpften Hook H bleibt die über die Reputation Registry akkumulierte Reputation unabhängig von der RexHook-Plattform bestehen. Formal: Wenn RexHook den Betrieb einstellt, bleiben Feedback-Datensätze für A über `ReputationRegistry.getSummary(A, clients, tag1, tag2)` abfragbar.

Beweis. ERC-8004-Registries werden als unabhängige Singleton-Verträge bereitgestellt. Feedback wird On-Chain ohne Abhängigkeit von der RexHook-Infrastruktur gespeichert. Die einzige von RexHook kontrollierte Komponente ist die Verknüpfung von der Hook-Adresse zur Agenten-ID, die ebenfalls unveränderlich On-Chain in der RexHook-Registry gespeichert ist. □

9.7 Registry als Übersetzungsschicht für Trading-Infrastruktur

Über die Zertifizierung hinaus erfüllt die RexHook-Registry eine kritische Middleware-Funktion: Sie übersetzt das fragmentierte V4-Launchpad-Ökosystem in eine standardisierte Schnittstelle, die bestehende Trading-Infrastruktur konsumieren kann. Dieser Abschnitt formalisiert die Architektur der Übersetzungsschicht (Translation Layer) und ihre Auswirkungen auf die ökosystemweite Kompatibilität.

Das Fragmentierungsproblem

Das Uniswap-V4-Ökosystem hat mehrere unabhängige Launchpad-Implementierungen hervorgebracht, jede mit proprietären Funktionssignaturen, Event-Formaten und Datenstrukturen:

Definition 9.3 (Fragmentierung der Launchpad-Schnittstellen)

Sei $L = \{L_1, L_2, \dots, L_n\}$ die Menge der V4-basierten Token-Launchpads. Jedes L_i definiert: (i) Swap-Funktionen mit plattformspezifischen Namen und Parametern (z. B. `ClankerSwap()`, `FLaunchBuy()`, `customTrade()`), (ii) Event-Signaturen mit variierenden indizierten Parametern und Daten-Encoding, (iii) Pool-Metadaten mit inkonsistenter Feldbenennung (`tokenIn/tokenOut` vs. `buy/sell` vs. `amount0/amount1`), (iv) Gebührenstrukturen mit unterschiedlicher Präzision und Extraktionsmechanik.

Diese Fragmentierung schafft eine kritische Barriere für Anbieter von Trading-Infrastruktur:

Trading-Tool	Aktuelle Integrationslast	Wartungskosten
Axiom	Eigener Adapter pro Launchpad	$O(n)$, n = Anzahl Launchpads
Maestro	Eigener Adapter pro Launchpad	$O(n)$
DexScreener	Eigener Parser pro Launchpad	$O(n)$
DexTools	Eigener Indexer pro Launchpad	$O(n)$
GMGN	Eigener Adapter pro Launchpad	$O(n)$
1inch / CowSwap	Eigene Routing-Logik pro Hook-Typ	$O(n)$

Tabelle 13f. Integrationslast der Trading-Infrastruktur ohne Standardisierung. Jedes neue Launchpad erfordert $O(1)$ -Integrationsarbeit von jedem Tool.

Das RexHook-Standardschema

RexHook definiert ein kanonisches Schema, das die Schnittstelle eines beliebigen V4-Launchpads auf ein standardisiertes Format abbildet:

Schnittstelle: IRexHookStandardSchema

```

struct StandardSwapData {
    address pool;           // V4 pool address
    address tokenIn;
    address tokenOut;
    uint256 amountIn;
    uint256 amountOut;
    uint256 feeAmount;     // Normalized to 18 decimals
    address feeToken;     // Token in which fee is denominated
    uint256 timestamp;
    bytes32 txHash;
}
struct StandardPoolData {
    address pool;
    address token0;
    address token1;
    address hook;         // Hook contract address
    string launchpadName; // e.g., "Clanker", "Flaunch", "RexHook"
    uint256 totalVolume;
    uint256 totalFees;
    uint256 createdAt;
    CertificationTier tier; // From RexHook certification
}
event StandardSwap(
    address indexed pool,
    address indexed tokenIn,
    address indexed tokenOut,
    uint256 amountIn,
    uint256 amountOut,
    uint256 feeAmount
);

```

Launchpad-Funktionszuordnung

Die Registry verwaltet eine Zuordnung von den nativen Funktionen jedes registrierten Launchpads zum Standardschema:

Schnittstelle: IRexHookTranslationLayer

```

struct LaunchpadMapping {
    string launchpadName;
    address factoryContract;
    bytes4 swapSelector;     // e.g., bytes4(keccak256("ClankerSwap()"))
    bytes4 deploySelector;   // Token deployment function
    string[] eventSignatures; // Native event signatures to index
    FunctionMapping[] mappings; // Parameter position mappings
}
struct FunctionMapping {
    string nativeParam;     // e.g., "clankerAmount"
    string standardParam;   // e.g., "amountIn"
    uint8 paramIndex;       // Position in native function
    DataType dataType;     // uint256, address, etc.
}
function registerLaunchpad(LaunchpadMapping mapping) external;
function translateSwap(address launchpad, bytes calldata rawData)
    external view returns (StandardSwapData);
function getPoolData(address pool) external view returns (StandardPoolData);

```

Beispiel: Zuordnung von Clanker und Flaunch

Das Folgende veranschaulicht, wie heterogene Launchpad-Schnittstellen auf das Standardschema abgebildet werden:

Launchpad	Native Funktion	Natives Event	RexHook-Standardzuordnung
Clanker	ClankerSwap(uint256 clankerAmt, uint256 pairedAmt)	ClankerSwapExecuted(id, sender, amt0, amt1)	StandardSwap(pool, tokenIn, tokenOut, amountIn, amountOut, fee)
Flaunch	FlaunchBuy(address token, uint256 ethAmount)	FlaunchTrade(token, buyer, ethIn, tokensOut)	StandardSwap(pool, tokenIn, tokenOut, amountIn, amountOut, fee)
RexHook-Launchpad (unser Produkt)	swap(PoolKey, SwapParams)	HookSwap(id, sender, amt0, amt1, fee0, fee1)	StandardSwap(pool, tokenIn, tokenOut, amountIn, amountOut, fee)
Künftiges Launchpad X	executeTrade(...)	TradeComplete(...)	StandardSwap(pool, tokenIn, tokenOut, amountIn, amountOut, fee)

Tabelle 13g. Heterogene Launchpad-Schnittstellen, abgebildet auf das RexHook-Standardschema. Hinweis: Das RexHook-Launchpad ist unser eigenes Produkt (Säule 1); die Übersetzungsschicht (Säule 3) standardisiert Daten von ALLEN Launchpads einschließlich unseres eigenen.

Theorem 9.8 (Reduktion der Integrationskomplexität)

Sei T ein Trading-Tool und $L = \{L_1, \dots, L_n\}$ die Menge der V4-Launchpads. Ohne die Übersetzungsschicht erfordert T einen Integrationsaufwand von $O(n)$. Mit der RexHook-Übersetzungsschicht erfordert T einen Aufwand von $O(1)$, wodurch die marginalen Integrationskosten für neue Launchpads auf null reduziert werden.

Beweis. Das Trading-Tool T integriert einmalig mit der RexHook-Registry-API, die nur `StandardSwapData`- und `StandardPoolData`-Strukturen offenlegt. Wenn sich ein neues Launchpad L_{n+1} bei RexHook registriert, werden seine Funktionszuordnungen zur Übersetzungsschicht hinzugefügt. Das Tool T erhält automatisch standardisierte Daten für L_{n+1} ohne Code-Änderungen. Die marginalen Kosten für die Unterstützung von L_{n+1} durch T betragen daher null. \square

Übersetzungsschicht-API

Trading-Tools und Router fragen die Übersetzungsschicht über eine RESTful-API und WebSocket-Streams ab:

RexHook Translation Layer API Endpoints

```
// REST Endpoints
GET /v1/pools // All pools in standard format
GET /v1/pools/{address} // Single pool data
GET /v1/pools/{address}/swaps // Recent swaps in standard format
GET /v1/launchpads // Registered launchpads and mappings
GET /v1/hooks/{address}/cert // Certification status
// WebSocket Streams
WS /v1/stream/swaps // Real-time swaps (all launchpads)
WS /v1/stream/pools // New pool deployments
WS /v1/stream/pools/{address} // Single pool activity
```

Wertversprechen nach Stakeholder

Anmerkung 9.2 (Wert für Trading-Tools)

Trading-Tools (Axiom, Maestro, DexScreener, DexTools, GMGN, Banana Gun) integrieren die RexHook-API einmalig und erhalten standardisierte Daten für jedes registrierte V4-Launchpad — aktuelle und künftige. Dies eliminiert die Notwendigkeit, eigene Adapter für jedes Launchpad zu bauen und zu warten, und reduziert den Engineering-Aufwand von $O(n)$ auf $O(1)$.

Anmerkung 9.3 (Wert für Launchpads)

Launchpads (Clanker, Flaunch und künftige Anbieter) registrieren ihre Funktionszuordnungen bei RexHook und erlangen sofort Kompatibilität mit allen integrierten Trading-Tools und Routern. Dies eliminiert die Notwendigkeit, individuelle Integrationen mit jedem Tool-Anbieter auszuhandeln.

Anmerkung 9.4 (Wert für Router)

DEX-Aggregatoren (1inch, CowSwap, Paraswap) fragen die Übersetzungsschicht ab, um standardisierte Routing-Daten plus Zertifizierungsstatus zu erhalten. Dies ermöglicht fundierte Routing-Entscheidungen: Routing über zertifizierte Hooks mit Vertrauen, Kennzeichnung oder Vermeidung nicht zertifizierter Hooks.

Netzwerkeffekte und Ökosystem-Lock-in

Theorem 9.9 (Zweiseitige Netzwerkeffekte)

Die RexHook-Übersetzungsschicht weist zweiseitige Netzwerkeffekte auf: (i) jedes zusätzliche Launchpad L_i , das sich registriert, erhöht den Wert für alle Trading-Tools T , und (ii) jedes zusätzliche Trading-Tool T_j , das integriert, erhöht den Wert für alle Launchpads L . Formal: Sei $V_T(n)$ der Wert für ein Trading-Tool, wenn n Launchpads registriert sind, und $V_L(m)$ der Wert für ein Launchpad, wenn m Tools integriert sind. Dann gilt $\partial V_T / \partial n > 0$ und $\partial V_L / \partial m > 0$.

Beweis. Für Trading-Tools: Jedes zusätzliche Launchpad erweitert die Menge der handelbaren Pools, die über die standardisierte API zugänglich sind, und erhöht so die Abdeckung und den Nutzerwert. Für Launchpads: Jedes zusätzliche integrierte Tool erweitert die Distributionsfläche für die auf diesem Launchpad bereitgestellten Pools und erhöht so Auffindbarkeit und Volumen. Beide Ableitungen sind streng positiv. □

Dieser zweiseitige Netzwerkeffekt schafft einen verteidigungsfähigen Burggraben: Sobald eine kritische Masse von Launchpads und Trading-Tools mit RexHook integriert ist, werden die Wechselkosten für jeden Teilnehmer prohibitiv hoch.

Abbildung 7b. Architektur der Übersetzungsschicht. V4-Launchpads (Clanker, Flaunch usw.) liefern Liquidität mit heterogenen Schnittstellen. Die RexHook-Registry standardisiert diese Daten. Trading-Tools und Router konsumieren standardisierte Daten über eine einzige Integration — sie benötigen keine eigenen Adapter für jedes Launchpad.

10 Entwicklungs-Roadmap

✓ ABGESCHLOSSEN

Kern-Protokollarchitektur, DynamicFeeHook, AntiSniper- und MaxWallet-Hooks, Hook-Zertifizierungs-Registry, SDK sowie die vollständige Uniswap-V4-Hook-Kompatibilitätsschicht. RexHook ist vollständig gebaut und deployment-bereit.

Phase	Zeitraumen	Liefergegenstände	KPIs
1: Launch	Q1 2026	Ethereum-Mainnet-Deployment, \$REX-TGE, öffentlicher Launch des Hook-Marktplatzes v1, Deployment des RexHook-Registry-Vertrags, Translation-Layer-API, öffentliche Verifizierungs-API, ERC-8004-Bridge-Verträge, initiale Launchpad-Zuordnungen (Clanker, Flaunch)	1.000 Deployments, 10+ Hooks im Marktplatz, 2+ zugeordnete Launchpads
2: Expansion	Q2 2026	Base/Arbitrum-Deployment, Drittanbieter-Hook-Einreichungen, Trading-Tool-Integrationen (DexScreener, DexTools, GMGN), Envio-Indexer-Integration, ERC-8004-Reputationsintegration, dynamische Metadaten-Pipeline, RexAgent (KI-Referenz-Trading-Agent)	5.000 Deployments, 50+ Marktplatz-Hooks, 3+ integrierte Trading-Tools, 30 % ERC-8004-Opt-in
3: Wachstum	Q3–Q4 2026	Weitere L2s, Router-Integrationen (1inch, CowSwap, Paraswap), Ausbau des Hook-Zertifizierungsprogramms, automatisierte Verifizierungs-Pipeline, 10+ Launchpads in der Übersetzungsschicht, ERC-8004-Validation-Registry-Integration, x402-Hook-Service-Endpunkte, KI-Agent-Routing-API	25.000 Deployments, 200+ Hooks, 100 % zertifiziert, 5+ integrierte Router, x402-Umsatz live
4: Skalierung	2027+	Cross-Chain-Hooks, institutionelle Features, Composability-Engine, Übersetzungsschicht als Industriestandard, KI-Agenten-Marktplatz, autonome Hook-Ökonomie, x402-V2-Integration	15–20 % Marktanteil, agentengeroutetes Volumen > 15 %, alle großen Trading-Tools integriert

Verwendung der Mittel aus der Seed-Runde

Die Seed-Runde ermöglicht das Mainnet-Deployment, die initiale Liquiditätsbereitstellung und die Go-to-Market-Umsetzung:

- **Mainnet-Deployment & Audits** — Abschluss des finalen externen Audits, gasoptimiertes Deployment über Ethereum + priorisierte L2s
- **Liquiditätsbereitstellung** — Initiale \$REX/ETH-Pool-Liquidität, Anreize für Launch-Partner
- **Wachstum & Marketing** — Developer Relations, Launch-Partnerschaften, Community-Aufbau
- **Betrieb** — 12-monatige Liquiditätsreserve für das Kernteam

Hinweis

Die gesamte Kernentwicklung ist abgeschlossen. Die Seed-Finanzierung ermöglicht das Deployment, nicht die Entwicklung. Die Registry- und Marktplatz-Integration ist im Testnet live und für das Mainnet bereit.

11 Verwandte Arbeiten

AMM-Design. Constant-Product-Market-Maker wurden von Uniswap [1] eingeführt und von [5] formalisiert. Konzentrierte Liquidität [2] verbesserte die Kapitaleffizienz, adressierte jedoch keine Mechanismen zur Gebührenerfassung. V4-Hooks [3] ermöglichen die von uns genutzte Anpassbarkeit.

MEV. Daian et al. [4] charakterisierten die MEV-Extraktion auf Ethereum. Flashbots [6] führten MEV-Share zur Umverteilung ein, was jedoch Off-Chain-Infrastruktur erfordert. Unser Ansatz operiert vollständig On-Chain.

Loss-Versus-Rebalancing. Milionis et al. [7] formalisierten LVR als Kosten für LPs. Unser MEV-Erfassungsmechanismus kann als partielle LVR-Minderung durch Wertumleitung betrachtet werden.

Mechanismusdesign. Unsere spieltheoretische Analyse baut auf Prinzipien des Mechanismusdesigns aus [8] auf. Das Pareto-Verbesserungsergebnis (Theorem 6.1) folgt aus der standardmäßigen Wohlfahrtsökonomik.

Hook-Datenstandards. Die Uniswap Foundation [16] schlug Standard-Events (HookSwap, HookFee, HookModifyLiquidity, HookBonus) für die Indexierung und Analyse von Hooks vor. Die BaseHook-Bibliothek von OpenZeppelin [17] stellt auditierte Basis-Implementierungen bereit. Unser Zertifizierungssystem baut auf diesen Standards auf und ergänzt sie um eine im Ökosystem fehlende Verifizierungs-Infrastruktur. Entscheidend ist: Während die UF Events innerhalb eines einzelnen Hooks standardisierte, standardisiert die Übersetzungsschicht von RexHook (Abschnitt 9.7) Daten über heterogene Launchpads hinweg und ermöglicht Trading-Tools, einmal zu integrieren und auf alle V4-Launchpads zuzugreifen.

Token-Sicherheitsinfrastruktur. GoPlus Security [18], Blockaid und Token Sniffer bieten ERC-20-Vertragsanalyse, jedoch keine V4-Hook-spezifische Erkennung. Unsere Registry schließt diese Lücke, indem sie diesen Plattformen Hook-bewusste Verifizierungsdaten per API-Integration bereitstellt. Die Übersetzungsschicht ermöglicht es diesen Tools zudem, Swap-Daten von jedem registrierten Launchpad ohne eigene Adapter zu parsen.

Agenten-Vertrauensinfrastruktur. ERC-8004 ("Trustless Agents") [20] etabliert On-Chain-Identitäts-, Reputations- und Validierungs-Registries für autonome Agenten. RexHook bietet optionale ERC-8004-Kompatibilität und ermöglicht es Hooks, von externen Systemen (DEX-Aggregatoren, KI-Trading-Agenten) über standardisierte Abfragen auffindbar zu sein, während die Reputation plattformübergreifend portabel bleibt.

Internet-native Zahlungen. Das x402-Protokoll [21,22] belebt HTTP 402 ("Payment Required") für Stablecoin-Mikrozahlungen zwischen Clients und Servern wieder. Wir integrieren x402 als Zahlungsschicht für Hook-Services (Abschnitt 12.4) und ermöglichen KI-Agenten, für MEV-Risiko-Scores, Routing-Daten und Analysen ohne API-Schlüssel oder Abonnements zu bezahlen. Dies erweitert den Hook-Markt zu einer vollständigen Agenten-Service-Ökonomie.

Spieltheorie der Allmende. Unsere Analyse der Dynamik des Hook-Ökosystems (Abschnitt 12.1) baut auf Hardins Tragödie der Allmende [23] und Ostroms institutioneller Analyse der Allmende-Governance [24] auf. Wir wenden diese Rahmenwerke auf das Hook-Ökosystem an und beweisen, dass Zertifizierung und Reputation defektionsdominierte Dynamiken in kooperationsdominierte Gleichgewichte verwandeln.

12 Agentenökonomie & Wachstumsausrichtung

Die Abschnitte 1–12 etablierten RexHook als die offene Infrastrukturschicht für Uniswap-V4-Hooks: ein Markt, auf dem Ersteller Token mit Plug-and-Play-Hooks bereitstellen, Entwickler neue Hooks bauen und verkaufen und zertifizierte Hooks über eine On-Chain-Registry verifiziert werden. Dieser Abschnitt erweitert das Rahmenwerk durch die Formalisierung der ökonomischen Dynamiken, die entstehen, wenn Hooks als autonome Service-Anbieter innerhalb einer breiteren Agentenökonomie operieren. Wir modellieren die Spannung zwischen individueller Extraktion und Ökosystem-Nachhaltigkeit als Allmende-Problem, leiten Bedingungen ab, unter denen kooperative Strategien dominieren, und beschreiben optionale Integrationen mit ERC-8004-Identität und x402-Zahlungsschienen.

Die zentrale These ist, dass die Architektur von RexHook einen selbstverstärkenden Wachstumskreislauf schafft: Hooks, die sich als ERC-8004-Agenten registrieren, werden für KI-Trading-Systeme auffindbar, die Volumen zu Hooks mit hoher Reputation leiten, was Gebühren generiert, die die weitere Ökosystem-Entwicklung finanzieren.

Das x402-Zahlungsprotokoll ermöglicht es diesem Kreislauf, autonom zu operieren — ohne menschliche Vermittlung, API-Schlüssel oder Abonnementverwaltung.

12.1 Das Hook-Ökosystem als Allmende-Problem

Die Tragödie der Allmende modelliert Situationen, in denen die individuell rationale Ausbeutung einer gemeinsamen Ressource zum kollektiven Ruin führt. Im Hook-Ökosystem ist die gemeinsame Ressource das Vertrauen der Trader: die aggregierte Bereitschaft der Trader, Volumen durch Hook-fähige Pools zu leiten.

Definition 12.1 (Trader-Vertrauens-Allmende)

Sei $T(t) \in [0, 1]$ das aggregierte Trader-Vertrauen zum Zeitpunkt t , definiert als die Wahrscheinlichkeit, dass ein zufällig ausgewählter Trader durch einen Hook-fähigen Pool statt durch einen einfachen Uniswap-V4-Pool routet. T ist eine Funktion der empirischen Verteilung des von den Marktteilnehmern beobachteten Hook-Verhaltens.

Jeder Hook-Ersteller i wählt eine Strategie $\sigma_i \in \{C, D\}$, wobei C (kooperieren) die Bereitstellung eines zertifizierten, transparenten Hooks mit Gebühren $\leq f_{\max}$ und ohne versteckte Extraktion bezeichnet, und D (defektieren) die Bereitstellung eines undurchsichtigen Hooks mit versteckten Gebühren, Backdoors oder Rug-Mechaniken bezeichnet.

Definition 12.2 (Allmende-Spiel des Hook-Ökosystems)

Ein N -Spieler-Spiel $\Gamma = (N, \{C, D\}^N, \{u_i\})$, wobei die Auszahlung für Ersteller i lautet:

$$u_i(C, \sigma_{-i}) = f_i \cdot V(T) \cdot T(n_C/N) \quad \text{— Umsatz aus transparenten Gebühren, skaliert mit Vertrauen}$$

$u_i(D, \sigma_{-i}) = E_i + f_i \cdot V(T) \cdot T((n_C - 1)/N)$ — einmalige Extraktion E_i plus reduzierter laufender Umsatz
wobei n_C die Anzahl der Kooperativeure, $V(T)$ das aggregierte Volumen (steigend in T) und T steigend im Kooperationsverhältnis n_C/N ist.

Annahme 12.1 (Vertrauensdynamik)

T ist konkav und steigend im Kooperationsverhältnis $r = n_C/N$, mit $T(0) = \tau_0 > 0$ (Basisvertrauen aus der Uniswap-Marke) und $T(1) = 1$. Ein einzelnes Defektionsereignis reduziert T um einen multiplikativen Faktor $(1 - \delta)$, wobei $\delta \in (0, 1)$ der Vertrauensschaden pro Vorfall ist, beobachtbar für alle Teilnehmer durch On-Chain-Events und Verbreitung in sozialen Medien.

Ohne RexHook: Defektion dominiert

In Abwesenheit eines Zertifizierungssystems hat das Spiel eine Struktur wie das Gefangenendilemma. Defektion liefert sofortige Extraktion E_i , während die Vertrauenskosten δ auf alle Teilnehmer verteilt werden. Wenn N groß ist, betragen die individuellen Kosten der Defektion $E_i - f_i \cdot V \cdot \delta/N$, was für jede nicht-triviale Extraktion positiv ist. Das eindeutige Nash-Gleichgewicht ist universelle Defektion ($\sigma^* = D^N$), konsistent mit der empirischen Ausfallrate von 48–98 % aus Abschnitt 7.

Mit RexHook: Kooperation dominant machen

Die Registry und das Zertifizierungssystem von RexHook führen drei Mechanismen ein, die das Gleichgewicht verschieben:

9. **Sichtbarkeit:** Zertifizierung macht Kooperation beobachtbar. Trader können zertifizierte Hooks von nicht zertifizierten über einen einzigen `eth_call` (Theorem 9.4) unterscheiden, was selektives Routing ermöglicht.
10. **Reputation:** Die ERC-8004-Integration (Abschnitt 9.6) macht die Kooperationshistorie portabel und dauerhaft. Die Erfolgsbilanz eines Hooks folgt ihm über alle Plattformen.

11. Volumen-Routing: KI-Agenten (Abschnitt 12.3) routen bevorzugt zu zertifizierten Hooks mit hoher Reputation und schaffen eine direkte ökonomische Belohnung für Kooperation.

Theorem 12.1 (Kooperation als dominante Strategie)

Sei Γ' das Hook-Ökosystem-Spiel mit RexHook-Zertifizierung, ERC-8004-Reputation und KI-Agent-Routing. Unter Annahme 12.1 ist Kooperation (C) die dominante Strategie für jeden Ersteller mit Zeithorizont $T > T^*$, wobei:

$$T^* = E_i / (f_i \cdot \Delta V_{cert})$$

und $\Delta V_{cert} = V_{zertifiziert} - V_{nicht-zertifiziert}$ die Volumenprämie für zertifizierte Hooks ist.

Beweis. Ein kooperierender Ersteller erzielt den kumulativen Umsatz $R_C(T) = f_i \cdot V_{zertifiziert} \cdot T$. Ein defektierender Ersteller erzielt $R_D(T) = E_i + f_i \cdot V_{nicht-zertifiziert} \cdot T$ (nach der Defektion routen KI-Agenten weg und die Reputation kollabiert auf null). Kooperation dominiert, wenn $R_C(T) > R_D(T)$, d. h. $f_i \cdot \Delta V_{cert} \cdot T > E_i$. Auflösung: $T > E_i / (f_i \cdot \Delta V_{cert}) = T^*$. Für realistische Parameter (Abschnitt 8.7) gilt $T^* < 30$ Tage, was bedeutet, dass Kooperation für jeden Ersteller dominiert, der über einen einzelnen Monat hinaus operieren möchte. \square

Anmerkung 12.1 (Verstärkung von Theorem 6.1)

Theorem 6.1 zeigte, dass RexHook eine Pareto-Verbesserung gegenüber traditionellen Mechanismen ist. Theorem 12.1 ist streng stärker: Es zeigt, dass Kooperation nicht nur für alle Parteien besser, sondern die individuell rationale Strategie ist — ohne Altruismus, Koordination oder Durchsetzung über die beobachtbaren Zertifizierungs- und Reputationssignale des Protokolls hinaus zu erfordern.

	Andere Ersteller: überwiegend Kooperation	Andere Ersteller: überwiegend Defektion
Kooperieren (zertifiziert)	$f \cdot V_{hoch} \cdot T$ Hohes Vertrauen → hohes Volumen	$f \cdot V_{mittel} \cdot T$ Zertifizierter Hook von KI-Agenten weiterhin bevorzugt
Defektieren (nicht zertifiziert)	$E + f \cdot V_{niedrig} \cdot T$ Einmalige Extraktion, dann weggeroutet	$E + f \cdot V_{min} \cdot T$ Ökosystem-Kollaps, minimales Volumen

Tabelle 21. Strategische Form des Hook-Ökosystem-Spiels unter RexHook. Die fett gedruckte Zelle ist das dominante Strategiegleichgewicht für $T > T^*$. Anders als beim klassischen Allmende-Problem macht Zertifizierung + KI-Routing Kooperation individuell rational.

12.2 Der vertikale Ausrichtungs-Stack

Die Abschnitte 3–5 behandeln Hooks als statischen, an Pools angehängten Code. Abschnitt 9.6 führte die optionale ERC-8004-Identität ein. Dieser Abschnitt formalisiert den vollständigen vertikalen Stack, der Hooks von passiven Callbacks in autonome ökonomische Agenten verwandelt, die an einer offenen Service-Ökonomie teilnehmen.

Definition 12.3 (Vertikaler Ausrichtungs-Stack)

Der vertikale Ausrichtungs-Stack (Vertical Alignment Stack) von RexHook ist eine vierschichtige Architektur, in der jede Schicht die Fähigkeiten der darunterliegenden Schicht erweitert:

L0 — Ausführungsschicht: Hook-Smart-Contracts (DynamicFeeHook, AntiSniperHook usw.), die Gebührenerfassung, MEV-Umverteilung und Pool-Logik über V4-Callbacks ausführen.

L1 — Identitäts- & Auffindungsschicht: ERC-8004-Agentenregistrierung, die jedem Hook eine portable On-Chain-Identität mit Metadaten-URI verleiht und die standardisierte Auffindung durch jeden ERC-8004-fähigen Client ermöglicht.

L2 — Reputations- & Vertrauensschicht: ERC-8004-Reputation-Registry, die On-Chain-Feedback von Pool-Erstellern, Tradern und anderen Hooks akkumuliert; plus die Validation Registry, die Audit-Nachweise und unabhängige Verifizierung aufzeichnet.

L3 — Ökonomische Schicht: x402-Zahlungsprotokoll, das KI-Agenten und externen Systemen ermöglicht, für Hook-Services (Premium-Datenfeeds, priorisiertes Routing, fortgeschrittener MEV-Schutz) über HTTP-native, On-Chain-abgewickelte Mikrozahlungen zu bezahlen.

Abbildung 8. Der vertikale Ausrichtungs-Stack. L0 ist für alle RexHook-Hooks obligatorisch. L1–L3 sind Opt-in-Schichten, die Hooks schrittweise von passiven Callbacks in autonome ökonomische Agenten verwandeln. Jede Schicht verstärkt den ökonomischen Wert der darunterliegenden Schichten, indem sie Auffindbarkeit, Vertrauen und Monetarisierungsfläche erhöht.

Proposition 12.1 (Schichtkomplementarität)

Für einen Hook H mit Basisumsatz R_0 auf L0 ist der inkrementelle Umsatz aus jeder zusätzlichen Schicht superadditiv:

$$R(L0 + L1 + L2 + L3) > R(L0) + \Delta R(L1) + \Delta R(L2) + \Delta R(L3)$$

Das heißt, der kombinierte Umsatz aus allen Schichten übersteigt die Summe ihrer Einzelbeiträge, weil jede Schicht die anderen verstärkt.

Beweis. L1 (Identität) ermöglicht Auffindung und erzeugt Volumen ΔV_1 . L2 (Reputation) wandelt Auffindung in bevorzugtes Routing mit Multiplikator $\alpha > 1$ um und liefert $\Delta V_2 = \alpha \cdot \Delta V_1 > \Delta V_1$. L3 (x402-Zahlungen) monetarisiert die Service-Schicht, die nur aufgrund der Auffindung und des Vertrauens von L1 + L2 existiert. Der Zahlungsumsatz R_3 ist sowohl an Identität (L1) als auch Reputation (L2) gebunden, daher gilt $R_3(L1, L2) > 0$, aber $R_3(\emptyset) = 0$. Folglich: $R(L0-L3) = R_0 + f \cdot \alpha \cdot \Delta V_1 + R_3(L1, L2) > R_0 + f \cdot \Delta V_1 + 0 + R_3(\emptyset)$. \square

12.3 KI-Agent-Integration

Autonome KI-Trading-Agenten stellen einen qualitativen Wandel in der DeFi-Teilnahme dar. Anders als menschliche Trader, die Pools manuell bewerten, können KI-Agenten On-Chain-Registries abfragen, Reputations-Scores bewerten, Gebührenstrukturen vergleichen und in Millisekunden optimal über Hunderte von Pools routen. Der vertikale Ausrichtungs-Stack (Definition 12.3) macht RexHook-Hooks zu nativen Teilnehmern dieser Agentenökonomie.

Auffindungsprotokoll

Ein KI-Trading-Agent, der eine optimale Ausführung für einen Token-Swap sucht, führt die folgende Auffindungssequenz aus:

Algorithmus 4: KI-Agent Hook-Auffindung

```

Input: Token pair (T, ETH), swap amount  $\Delta x$ , risk tolerance  $\rho$ 
Output: Optimal pool with hook configuration
1:  $\triangleright$  L1: Discover available hooks via ERC-8004
2: agents[]  $\leftarrow$  ERC8004.IdentityRegistry.getAgentsByService("uniswap-v4-hook")
3: for each agent in agents[] do
4:   metadata  $\leftarrow$  fetch(agent.agentURI)  $\triangleright$  Registration file
5:   hookAddr  $\leftarrow$  metadata.hookMetadata.address
6:   cert  $\leftarrow$  RexHookRegistry.verifyHook(hookAddr)  $\triangleright$  L0: Cert check
7:    $\triangleright$  L2: Assess reputation
8:   rep  $\leftarrow$  ERC8004.ReputationRegistry.getSummary(agent.id, ..., "reliability", "")
9:   val  $\leftarrow$  ERC8004.ValidationRegistry.getSummary(agent.id, ..., "")
10:  score[agent]  $\leftarrow$  ComputeRouteScore(cert, rep, val, metadata.fees,  $\rho$ )
11: end for
12: bestHook  $\leftarrow$  argmax(score[])
13: return ExecuteSwap(bestHook.pool,  $\Delta x$ )

```

Anmerkung 12.2 (Bootstrapping des Agenten-Kreislaufs)

Die KI-Agent-Auffindung erfordert, dass Hooks registriert sind, bevor Agenten Integrationen bauen (das Henne-Ei-Problem). RexHook adressiert dies durch: (i) das Bereitstellen eines quelloffenen KI-Referenz-Trading-Agenten (RexAgent), der den Routing-Vorteil der ERC-8004-Auffindung demonstriert, (ii) die automatische Registrierung aller über die RexHook-Plattform bereitgestellten Hooks in der RexHook-Registry auf L0 mit One-Click-Opt-in zu ERC-8004 auf L1, und (iii) das Seeding der Reputationsschicht, indem das Protokoll selbst Feedback zu Hook-Verfügbarkeit und Leistungsmetriken über die Envio-Indexer-Pipeline einreicht (Abschnitt 10, Phase 2).

Dynamische Metadaten & Leistungsfeeds

Abschnitt 9.6 behandelte die ERC-8004-Registrierung als statisches Opt-in. Für Hooks, die als lebende ökonomische Agenten operieren, muss die Agenten-Registrierungsdatei die Echtzeit-Leistung widerspiegeln. RexHook implementiert eine kontinuierliche Metadaten-Pipeline: Hook-Events werden in Echtzeit von Envio indexiert, Off-Chain zu Leistungs-Scores aggregiert, an IPFS gepinnt und über setAgentURI in die ERC-8004-Identity-Registry geschrieben.

Abbildung 9. Dynamische Metadaten-Pipeline. Hook-Events werden in Echtzeit von Envio indexiert, Off-Chain zu Leistungs-Scores aggregiert, an IPFS gepinnt und über setAgentURI in die ERC-8004-Identity-Registry geschrieben. KI-Agenten konsumieren aktualisierte Metadaten, um Routing-Entscheidungen zu treffen.

12.4 x402-Zahlungsintegration

Das x402-Protokoll (HTTP 402 "Payment Required") ermöglicht den nutzungsbasierten Zugang zu jeder HTTP-Ressource über On-Chain-Stablecoin-Abwicklungen. Für RexHook schafft x402 einen nativen Monetarisierungskanal für Hook-Services, der über die Gebührenerfassung zur Swap-Zeit hinausgeht.

Definition 12.4 (Hook-Service-Endpunkt)

Ein Hook-Service-Endpunkt ist eine HTTP-API, die von einem Hook-Entwickler (oder der RexHook-Plattform) bereitgestellt wird und mehrwertschaffende Services im Zusammenhang mit dem Betrieb eines Hooks bietet. Beispiele umfassen: Echtzeit-MEV-Risiko-Scores, Vor-Trade-Gebührensätzungen, historische Leistungsdaten und priorisierte Routing-Garantien. Jeder Endpunkt kann unabhängig eine x402-Zahlung erfordern.

Zahlungsmodell

Hook-Service-Endpunkte unterstützen drei Preisstufen, passend zur Schema-Erweiterbarkeit des x402-Protokolls:

Stufe	x402-Schema	Anwendungsfall	Beispielpreis	Abwicklung
Query	exact	MEV-Risiko-Score, Gebührenschtätzung, Hook-Metadaten	\$0,001–\$0,01 pro Aufruf	USDC auf Base
Routing	exact	Priorisiertes Routing, MEV-Schutz-Bundle	\$0,01–\$0,10 pro Swap	USDC auf Base/L2
Session	upto	Streaming-Datenfeed, Echtzeit-Analysen	Bis zu \$1,00/Stunde	USDC auf Base

Tabelle 22. x402-Zahlungsstufen für Hook-Services. Das exact-Schema überträgt einen festen Betrag pro Anfrage; das upto-Schema autorisiert bis zu einem Maximum basierend auf den während der Sitzung verbrauchten Ressourcen.

Zahlungsfluss

Algorithmus 5: x402 Hook-Service-Zahlungsfluss

Participants: AI Agent (client), Hook Service (server), x402 Facilitator

1: Agent → GET /api/v1/hooks/{addr}/mev-risk ▷ Request w/o payment

2: Server → 402 Payment Required ▷ Returns header

```
{
  scheme: "exact",
  network: "eip155:8453",      ▷ Base
  maxAmountRequired: "10000", ▷ $0.01 USDC (6 decimals)
  resource: "/api/v1/hooks/{addr}/mev-risk",
  payTo: "{hookDeveloperAddress}", ▷ Direct to developer
  description: "MEV risk score for hook 0x..."
}
```

3: Agent → Signs EIP-712 payment payload (USDC transfer auth)

4: Agent → GET /api/v1/hooks/{addr}/mev-risk [X-PAYMENT: {payload}]

5: Server → Facilitator.verify(payload, requirements) ▷ Verify+settle

6: Server → 200 OK {riskScore: 0.03, confidence: 0.92, ...}

Umsatzverteilung

x402-Zahlungen fließen durch eine Aufteilung, die mit dem Modell des Hook-Marktplatzes (Abschnitt 8.6) identisch ist:

$$R_{dev} = 0,70 \cdot R_{x402}, \quad R_{protocol} = 0,30 \cdot R_{x402} \quad (12.1)$$

wobei R_{x402} der gesamte x402-Service-Umsatz ist. Dies entspricht der 70/30-Aufteilung des Marktplatzes und hält die Entwickleranreize über alle Umsatzkanäle hinweg konsistent. Der 30%-Anteil des Protokolls fließt über den bestehenden Gebührenverteilungsmechanismus (Abschnitt 8.5) an die \$REX-Staker.

x402 in der ERC-8004-Registrierung

Hooks, die x402-fähige Services anbieten, deklarieren dies in ihrer ERC-8004-Registrierungsdatei über das Feld x402Support (bereits im ERC-8004-Standard spezifiziert):

ERC-8004-Registrierung mit x402-Services

```
{
  "type": "https://eips.ethereum.org/EIPS/eip-8004#registration-v1",
  "name": "RexHook-DynamicFeeHook-v1",
  "description": "Dynamic fee hook with MEV protection ...",
  "services": [
    {"name": "MCP", "endpoint": "https://mcp.rexhook.com/hooks/0x.../", "version": "2025-06-18"},
    {"name": "web", "endpoint": "https://api.rexhook.com/hooks/0x.../mev-risk"}
  ],
  "x402Support": true,           ▷ AI agents know they can pay
  "supportedTrust": ["reputation", "crypto-economic"],
  "hookMetadata": {
    "flags": ["beforeSwap", "afterSwap"],
    "rexhookCertification": "gold",
    "performanceMetrics": { ▷ Dynamic, updated via pipeline
      "uptimePct": 99.7,
      "successRate": 98.2,
      "totalVolumeRouted": "12500000",
      "avgResponseTimeMs": 340,
      "lastUpdatedBlock": 21847231
    }
  }
}
```

Anmerkung 12.3 (x402 als Zahlungsnachweis in der Reputation)

Die ERC-8004-Reputation-Registry unterstützt ein proofOfPayment-Feld in Feedback-Einreichungen und ermöglicht die Verknüpfung von x402-Transaktions-Hashes mit Feedback. Dies schafft verifizierte Bewertungen: Feedback von Agenten, die nachweislich für die Services eines Hooks bezahlt und sie genutzt haben, trägt höhere Glaubwürdigkeit als anonymes Feedback und bietet einen natürlichen Sybil-Resistenz-Mechanismus für das Reputationssystem.

12.5 Der selbstverstärkende Wachstumskreislauf

Die Kombination aus dem vertikalen Ausrichtungs-Stack, KI-Agent-Routing und x402-Zahlungen schafft einen Rückkopplungskreislauf, in dem das rationale Verhalten jedes Teilnehmers das Ökosystem für alle anderen stärkt:

12. Ersteller stellen Hooks bereit — Kostenloses Deployment, RexHook-Registry + ERC-8004

13. KI-Agenten finden Hooks — ERC-8004-Identitäts- + Reputationsabfrage

14. Volumen routet zu den besten Hooks — Zertifiziert + hohe Reputation = mehr Swaps

15. Gebühren + x402-Umsatz fließen — Ersteller 70 % + Protokoll 30 % + \$REX-Staker

16. Reputation akkumuliert — On-Chain-Feedback, x402-Zahlungsnachweis

17. Mehr Ersteller & Entwickler treten bei — Marktplatz wächst, Hook-Vielfalt steigt

Abbildung 10. Der selbstverstärkende Wachstumskreislauf. Jeder Schritt führt in den nächsten: mehr Hooks → mehr Auffindung → mehr Volumen → mehr Umsatz → mehr Reputation → mehr Teilnehmer. x402-Zahlungen fügen bei Schritt 4 einen zusätzlichen Umsatzkanal hinzu, während ERC-8004-Identität und -Reputation bei den Schritten 2 und 5 den Kreislauf für KI-Agenten zugänglich machen. Externe Integrationen (DEX-Aggregatoren, Scanner) verstärken den Zyklus, indem sie Volumen von außerhalb des RexHook-Ökosystems einbringen.

Theorem 12.2 (Stabilität des Wachstumskreislaufs)

Sei $N(t)$ die Anzahl der zertifizierten Hooks, $V(t)$ das aggregierte Volumen und $T(t)$ das Trader-Vertrauen zum Zeitpunkt t . Unter der Wachstumskreislauf-Dynamik (Abbildung 10) und Annahme 12.1 hat das System einen stabilen Fixpunkt bei (N^*, V^*, T^*) mit $T^* > \tau_0$ (über dem Basiswert), sofern das Kooperationsverhältnis $r = n_C/N$ einen kritischen Schwellenwert r^* überschreitet.

Beweisskizze. Der Wachstumskreislauf definiert ein dynamisches System: $dN/dt = g(V, T)$ (Ersteller treten proportional zur Ökosystem-Attraktivität bei), $dV/dt = h(N, T)$ (Volumen steigt mit Hook-Vielfalt und Vertrauen), $dT/dt = k(r) - \lambda T$ (Vertrauen baut sich mit dem Kooperationsverhältnis auf, zerfällt mit Defektion). Das Nullsetzen der Ableitungen und Anwenden des Satzes über implizite Funktionen liefert einen eindeutigen inneren Fixpunkt, wenn $r > r^ = \lambda/k'(1)$. Die Stabilität folgt daraus, dass die Jacobi-Matrix am Fixpunkt negative reelle Eigenwerte hat, was gilt, weil alle Rückkopplungsverbindungen ($N \rightarrow V$, $V \rightarrow T$, $T \rightarrow N$) positive partielle Ableitungen haben, die durch Konkavität begrenzt sind (Annahme 12.1). ■*

Umsatzwirkungs-Prognose

Das Hinzufügen der Agentenökonomie-Schicht (L1–L3) schafft inkrementellen Umsatz über die Basisprognosen aus Abschnitt 8.7 hinaus:

Umsatzstrom	Quelle	J1-Schätzung	J3-Schätzung
Basis-Protokollumsatz (Abschnitt 8.7)	L0: Gebührenerfassung + Marktplatz	\$2,15 Mio.	\$34,25 Mio.
KI-Agent-Routing-Prämie	L1–L2: ΔV_{cert} aus Agenten-Auffindung	\$150 Tsd.	\$5,5 Mio.
x402-Service-Zahlungen	L3: MEV-Scores, Datenfeeds, prior. Routing	\$75 Tsd.	\$3,2 Mio.
Gesamt (mit Agentenökonomie)		\$2,375 Mio.	\$42,95 Mio.

Tabelle 23. Inkrementeller Umsatz aus der Agentenökonomie-Schicht. Die J1-Schätzungen sind angesichts der Bootstrapping-Anforderungen konservativ. J3 nimmt an, dass 30 % der Hooks sich für ERC-8004 entscheiden und 15 % des Trading-Volumens KI-Agent-geroutet wird, konsistent mit Branchenprognosen für das Wachstum des autonomen Tradings.

Anmerkung 12.4 (Konservative Schätzungen)

Die x402-Umsatzprognosen nehmen Mikrozahlungsvolumina vergleichbar mit der frühen API-Monetarisierung an. Wenn das KI-Agent-Trading den von Branchenanalysten für 2028 prognostizierten Volumenanteil von 40–60 % erreicht, könnten die x402-Service-Zahlungen den Basis-Protokollumsatz übersteigen. Diese Prognosen sind bewusst konservativ und werden mit empirischen Daten nach dem Launch verfeinert (Abschnitt 7.3, Meilenstein M5).

13 Schlussfolgerung

Wir haben RexHook vorgestellt, die offene Infrastrukturschicht für Uniswap V4 — eine Full-Stack-Plattform, die aus fünf Säulen besteht: Token-Launchpad, Hook-Marktplatz, On-Chain-Registry mit Übersetzungsschicht, V4-Netzwerk-Explorer und Trading-Terminal. Über die Lösung des Todesspiral-Problems hinaus stellt RexHook die Middleware bereit, die das fragmentierte V4-Ökosystem vereint. Begutachtete Forschung zeigt, dass 48–60 % der Token innerhalb von 24 Stunden scheitern [11,13], wobei bis zu 98 % betrügerische Merkmale aufweisen [14]. Unsere wichtigsten Beiträge sind:

- 18.Theorem 2.1: Formale Charakterisierung der Instabilitätsbedingungen der Todesspirale
- 19.Theorem 3.2: Beweis des Null-Verkaufsdrucks durch In-Swap-Gebührenerfassung
- 20.Proposition 5.1: Herleitung der optimalen MEV-Erfassungsrate
- 21.Theorem 6.1: Pareto-Verbesserung gegenüber traditionellen Mechanismen
- 22.Theorem 9.5: Verifizierungsvollständigkeit für die Hook-Zertifizierung
- 23.Theorem 9.6: Indexer-Kompatibilität durch standardisierte Event-Emission
- 24.Theorem 9.8: Die Übersetzungsschicht reduziert die Integrationskomplexität von Trading-Tools von $O(n)$ auf $O(1)$
- 25.Theorem 9.9: Zweiseitige Netzwerkeffekte schaffen einen verteidigungsfähigen Burggraben
- 26.Theorem 12.1: Kooperation als dominante Strategie unter Zertifizierung und KI-Agent-Routing (Lösung des Allmende-Problems)
- 27.Theorem 12.2: Stabilität des Wachstumskreislaufs unter kooperativem Gleichgewicht

Empirische Forschung zu Token-Ausfallraten stützt unsere theoretischen Vorhersagen. Basierend auf diesen Ausgangswerten und Theorem 3.2 verbessert RexHook die Überlebensraten um etwa das 1,56-Fache, indem es gebühreninduzierte Todesspiralen eliminiert.

RexHook adressiert zwei kritische Lücken im V4-Ökosystem: (1) es existiert keine offizielle Hook-Registry oder Zertifizierungssystem — unsere On-Chain-Registry stellt verifizierbare Hook-Metadaten über standardisierte EVM-Schnittstellen bereit (Theorem 9.4), und (2) es gibt keine standardisierte Schnittstelle über V4-Launchpads hinweg — unsere Übersetzungsschicht bildet heterogene Launchpads (Clanker, Flaunch usw.) auf ein Standardschema ab und ermöglicht Trading-Tools, einmal zu integrieren und auf alle Launchpads zuzugreifen (Theorem 9.8). Alle Verifizierungseigenschaften sind unabhängig On-Chain ohne vertrauenswürdige Dritte verifizierbar (Theorem 9.5).

Das Agentenökonomie-Rahmenwerk (Abschnitt 12) erweitert RexHook von einer Hook-Plattform zu einem selbstverstärkenden ökonomischen System. Hooks, die sich als ERC-8004-Agenten registrieren, werden für KI-Trading-Systeme auffindbar, die Volumen zu Hooks mit hoher Reputation leiten und Gebühren generieren, die die weitere Ökosystem-Entwicklung finanzieren. Das x402-Zahlungsprotokoll ermöglicht es diesem Kreislauf, autonom zu operieren. Wir beweisen, dass unter dieser Architektur Kooperation (zertifizierte, transparente Hooks) die individuell rationale Strategie für jeden Ersteller mit einem Zeithorizont von mehr als etwa 30 Tagen ist (Theorem 12.1) — ohne Altruismus, Koordination oder Durchsetzung über die beobachtbaren Signale des Protokolls hinaus zu erfordern.

Zukünftige Arbeiten umfassen: (i) formale Verifizierung aller Hook-Verträge, (ii) Erweiterung auf Pools mit konzentrierter Liquidität, (iii) Optimierung des Cross-Chain-Deployments, (iv) Design von Governance-Mechanismen, (v) empirische Validierung der Volumen-Preis-Korrelation speziell für Tax-Token, (vi) Deployment und empirische Evaluierung des RexAgent-KI-Referenz-Trading-Agenten, (vii) x402-V2-Integration für abonnementbasierten Zugang zu Hook-Services und (viii) Erweiterung der Übersetzungsschicht zur Unterstützung zusätzlicher Launchpad-Schnittstellen.

Referenzen

- 28.Hayden Adams. Uniswap whitepaper. <https://uniswap.org>, 2018.
- 29.Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 Core. Technical report, Uniswap Labs, 2021.
- 30.Hayden Adams et al. Uniswap v4 Core. Technical report, Uniswap Labs, 2023.

31. Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In 2020 IEEE Symposium on Security and Privacy (SP), pages 910–927, 2020.
32. Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of Uniswap markets. arXiv preprint arXiv:1911.03380, 2019.
33. Flashbots. MEV-Share: programmable order flow. <https://docs.flashbots.net>, 2023.
34. Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, and Anthony Lee Zhang. Automated market making and loss-versus-rebalancing. arXiv preprint arXiv:2208.06046, 2022.
35. Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.
36. Andrea Canidio and Robin Fritsch. Arbitrageurs' profits, LVR, and sandwich attacks: batch trading as an AMM design response. arXiv preprint arXiv:2307.02074, 2023.
37. Liyi Zhou, Kaihua Qin, Christof Ferreira Torres, Duc V. Le, and Arthur Gervais. High-frequency trading on decentralized on-chain exchanges. In 2021 IEEE Symposium on Security and Privacy (SP), pages 428–445, 2021.
38. CertiK. Evil in the Shadows: Unveiling the Chaos in Ethereum's Token Ecosystem. CertiK Research, January 2025. <https://www.certik.com/resources/blog/evil-in-the-shadows-unveiling-the-chaos-in-ethereums-token-ecosystem>
39. Pengcheng Xia, Haoyu Wang, Bingyu Gao, Weihang Su, Zhou Yu, Xiapu Luo, Chao Zhang, Xusheng Xiao, and Guoai Xu. Trade or Trick? Detecting and Characterizing Scam Tokens on Uniswap Decentralized Exchange. *Proceedings of the ACM on Measurement and Analysis of Computing Systems (SIGMETRICS)*, 2021.
40. Federico Cerner, Massimo La Morgia, Alessandro Mei, and Francesco Sassi. Token Spammers, Rug Pulls, and Sniper Bots: An Analysis of the Ecosystem of Tokens in Ethereum and the Binance Smart Chain. In *USENIX Security Symposium*, 2023.
41. Arina Kalacheva, Nikita Kuznetsov, Dmitry Vodolazov, and Yury Yanovich. Detecting Rug Pulls in Decentralized Exchanges. *Blockchain: Research and Applications*, July 2025.
42. CoinGecko. How Many Cryptocurrencies Have Failed? CoinGecko Research, December 2025. <https://www.coingecko.com/research/publications/how-many-cryptocurrencies-failed>
43. Uniswap Foundation. Establishing Hook Data Standards for Uniswap v4: A Guide for Developers, Indexers, and Analysts. <https://www.uniswapfoundation.org/blog/developer-guide-establishing-hook-data-standards-for-uniswap-v4>, March 2025.
44. OpenZeppelin. Uniswap Hooks Library: Solidity library for secure and modular Uniswap hooks. <https://docs.openzeppelin.com/uniswap-hooks>, 2025.
45. GoPlus Security. Token Security API: Open, permissionless, user-driven token security detection platform. <https://gopluslabs.io/token-security>, 2024.
46. Envio. Uniswap V4 Multi-chain Indexer. <https://github.com/enviodev/uniswap-v4-indexer>, 2025.
47. Marco De Rossi, Davide Crapis, Jordan Ellis, and Erik Reppel. ERC-8004: Trustless Agents. *Ethereum Improvement Proposals*, no. 8004, August 2025. <https://eips.ethereum.org/EIPS/eip-8004>
48. Coinbase. x402: An Open Payment Protocol for the Internet. <https://www.x402.org>, 2025.
49. Coinbase. x402 V2: Evolving the Standard for Internet-native Payments. <https://www.x402.org/writing/x402-v2-launch>, 2025.
50. Garrett Hardin. The Tragedy of the Commons. *Science*, 162(3859):1243–1248, 1968.

51. Elinor Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990.
52. Uniswap Foundation. UNI Token Allocation. <https://uniswap.org/blog/uni>, September 2020.
53. Ethena Labs. ENA Token Distribution and Governance. <https://ethena.fi/tokenomics>, April 2024.
54. Messari. Tokenomics Database: Analysis of 500+ DeFi Protocol Token Distributions 2020-2024. <https://messari.io/research>, 2024.
55. Binance Research. Token Launch Trends: TGE Circulating Supply and Vesting Schedules in DeFi. <https://research.binance.com/en/analysis/token-launch-trends>, December 2024.

Anhang A: Ergänzende Beweise

Lemma A.1 (Schranke für Preis-Impact)

Für einen Constant-Product-AMM mit Reserven (x, y) und Swap-Betrag Δx erfüllt der Preis-Impact:

$$|\Delta P/P| \leq \Delta x/x \cdot (2 + \Delta x/x)$$

Beweis. Anfangspreis $P_0 = y/x$. Nach dem Swap: $P_1 = (y - \Delta y)/(x + \Delta x) = \kappa/(x + \Delta x)^2$. Damit gilt $\Delta P/P = P_1/P_0 - 1 = x^2/(x + \Delta x)^2 - 1 = -(2x\Delta x + \Delta x^2)/(x + \Delta x)^2$. Sei $r = \Delta x/x$. Dann $|\Delta P/P| = r(2 + r)/(1 + r)^2$. Da $(1 + r)^2 > 1$ für $r > 0$, gilt $|\Delta P/P| < r(2 + r) = (\Delta x/x)(2 + \Delta x/x)$. \square

Lemma A.2 (Vollständigkeit der Verteilung)

Für eine Gebühr F und Anteile $\{s_1, \dots, s_n\}$ mit $\sum s_i = 1$ entspricht der gesamte verteilte Betrag genau F (kein Wertverlust und keine Wertschöpfung).

Beweis. Gesamt verteilt $= \sum_i s_i F = F \cdot \sum_i s_i = F \cdot 1 = F$. \square

RexHook

Die offene Infrastrukturschicht für Uniswap-V4-Hooks

rexhook.com | docs.rexhook.com | @rexhooks

© 2026 RexHook. Alle Rechte vorbehalten.

Whitepaper v1.6 — Februar 2026

VERTRAULICH