A Graph-Theoretic Framework for DeFi Vault Risk Decomposition

Gregory John Komansky*

GJKapital Research

January 2026

Preliminary Draft — Comments Welcome

Abstract

Decentralized finance (DeFi) vaults—smart contracts that automate yield strategies across composable protocols—represent over $16 billion in total value locked yet lack standardized risk decomposition frameworks. We propose a graph-theoretic approach: representing vaults as directed acyclic graphs where nodes are typed by four *atomic primitives* (CONTRACT, ORACLE, GOVERNANCE, OPERATIONAL) and edges encode dependency relationships.

This formalization enables node-level risk attribution, where every basis point of expected loss traces to a specific primitive. We calibrate base rates from a dataset of 449 documented exploits (2016–2026) totaling $15.7 billion in losses, sourced from DeFiLlama and validated against Rekt News.

Key finding: CONTRACT failures dominate frequency (65%) while OPERATIONAL failures dominate severity (50% of losses)—a distinction invisible without formal decomposition. The framework provides: (i) a complete taxonomy covering all historical exploit root causes, (ii) transparent aggregation from node-level to vault-level risk, and (iii) cross-protocol comparison on a common basis.

The framework does not predict exploits; rather, it provides the decomposition infrastructure that enables risk budgeting, concentration analysis, and board-level reporting—capabilities institutions require but DeFi currently lacks.

Keywords: decentralized finance, risk decomposition, directed acyclic graphs, atomic primitives, smart contract risk, institutional allocation

JEL Classification: G11, G21, G23, G32

*Email: gjkomansky@gmail.com. The author thanks anonymous reviewers for valuable feedback. Data sources and replication code available at https://github.com/eggbertgjk/vault-risk-framework.

1 Introduction

1.1 Motivation

A fixed income portfolio manager can decompose bond portfolio risk into interpretable components: duration contribution, credit spread sensitivity, sector concentration, issuer limits. Every basis point of risk traces to a measurable factor. This decomposition infrastructure—developed over decades—enables institutional allocation, regulatory compliance, and fiduciary reporting.

No equivalent infrastructure exists for DeFi vaults.

Vaults are smart contracts that pool capital and execute yield-generating strategies across decentralized protocols. They have grown from negligible TVL in 2020 to over $16 billion by late 2025 [1]. Yet risk assessment remains primitive: audits verify code correctness but not architectural risk concentration; yield dashboards report APY but not expected loss; due diligence is ad hoc and incomparable across protocols.

This paper addresses a specific question: *How should vault risk be decomposed to enable institutional-grade analysis?*

1.2 Contribution

We make three contributions:

First, we formalize vault risk decomposition. We represent vaults as directed acyclic graphs where nodes are typed by four *atomic primitives*: CONTRACT, ORACLE, GOVERNANCE, and OPERATIONAL. We prove completeness—every exploit in our 449-incident dataset maps to these primitives. The formalization is not the novelty; the calibrated, auditable, standardized implementation is.

Second, we provide empirical calibration. Using 449 exploits totaling $15.69B in losses over 9.56 years, we estimate base rates: CONTRACT failures occur at 6.11% annually, OPERATIONAL at 1.93%, ORACLE at

1.28%, GOVERNANCE at 0.06%. These are the first published base rates derived from comprehensive exploit data.

Third, we deliver decomposition infrastructure. The framework enables statements like: "This vault's 45 bps expected loss decomposes as 22 bps CONTRACT, 12 bps OPERATIONAL, 8 bps ORACLE, 3 bps GOVERNANCE." This granularity enables risk budgeting, concentration limits, and comparable cross-protocol assessment—the factor-based analysis standard in traditional finance.

1.3 Scope and Limitations

We emphasize what this framework is *not*. It is not an exploit predictor. Credit ratings do not predict which company will default; they categorize relative risk to enable portfolio decisions. Similarly, our framework categorizes architectural risk to enable allocation decisions.

The framework assesses *on-chain* risk from observable properties (contract immutability, oracle redundancy, governance structure). Off-chain risks—custodial failures, social engineering, regulatory action—are outside scope.

Calibration derives from historical data. Novel attack vectors not represented in historical exploits may not be captured. Coefficients should be updated as new data accumulates.

1.4 Related Work

Our work intersects several literatures.

DeFi Security. Atzei et al. [4] survey smart contract vulnerabilities. Perez and Livshits [5] analyze contract weaknesses at scale. Zhou et al. [6] systematize DeFi attack knowledge. These works focus on vulnerability classification; we focus on risk aggregation for allocation decisions.

Protocol Risk. Gudgeon et al. [7] analyze lending protocol stability. Werner et al. [8] systematize DeFi research. Qin et al. [9] compare CeFi and DeFi risks. Our contribution is a formal aggregation framework, not protocol-specific analysis.

AMM Theory. Angeris and Chitra [10] formalize constant function market makers. Milionis et al. [11] analyze loss-versus-rebalancing. We incorporate AMM exposure as a dependency primitive.

Traditional Risk. Our approach draws on factor decomposition in fixed income [12] and operational risk modeling [13]. The FMEA (Failure Mode and Effects Analysis) methodology from reliability engineering [14] informs our calibration approach.

DeFi Risk Measurement. Closest to our work, Xu and Livshits [17] propose a framework for quantifying DeFi protocol risk. Nexus Mutual [18] provides market-based risk pricing through parametric insurance. Artzner et al. [15] establish coherent risk measure axioms that inform our aggregation design—our MAX propagation satisfies monotonicity and sub-additivity. Holló et al. [16] construct the CISS composite indicator for systemic stress, whose aggregation methodology parallels our primitive-to-vault composition.

Systemic Risk in Crypto. Farzulla and Maksakov [21] propose the Adaptive Systemic Risk Index (ASRI) for crypto markets, using HMM regime detection across four sub-indices (stablecoin concentration, DeFi liquidity, contagion, regulatory opacity). Their framework operates at the macro level—detecting *when* the system is stressed—while ours operates at the micro level, identifying *where* risk concentrates within individual vault architectures. The two approaches are complementary: ASRI signals could trigger portfolio-level rebalancing, while VRS decomposition guides vault-level allocation within each regime. Separately, Farzulla [22] finds that whitepaper narratives do not predict market factor structure, reinforcing our premise that risk should be measured from observable on-chain properties rather than stated intentions.

1.5 Paper Structure

Section 2 formalizes the vault graph model. Section 3 defines atomic primitives and proves completeness. Section 4 develops risk propagation. Section 5 presents empirical calibration. Section 6 demonstrates applications. Section 7 discusses limitations and extensions.

2 The Vault Graph Model

2.1 Preliminaries

We represent a vault as a directed acyclic graph (DAG) capturing its dependency structure. Why a DAG? Figure 1 illustrates:

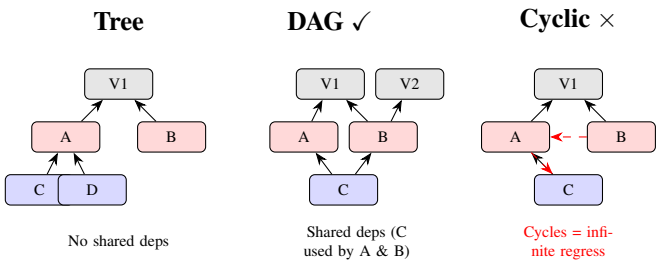


Figure 1: Why DAG? Trees cannot represent shared dependencies (Chainlink used by multiple protocols). Cyclic graphs imply infinite capital regress. DAGs capture DeFi composability.

(i) *Directed*: dependencies have direction—the vault depends on Morpho, not vice versa. (ii) *Acyclic*: capital flows cannot cycle back to their origin in a single transaction (Lemma 2.3). (iii) *Shared dependencies*: unlike trees, DAGs allow multiple protocols to share the same oracle (e.g., Chainlink).

Definition 2.1 (Vault Graph)**.** A vault V is represented by a tuple $G_V = (N, E, \tau, \omega)$ where:

- N is a finite set of nodes
- $E \subseteq N \times N$ is a set of directed edges
- $\tau : N \to \mathscr{P}$ assigns each node a primitive type
- $\omega : N \to \mathbb{R}^k$ assigns each node a property vector

where $\mathscr{P} = \{\texttt{CONTRACT}, \texttt{ORACLE}, \texttt{GOVERNANCE}, \texttt{OPERATIONAL}\}$ is the set of primitive types.[1]

Edges encode dependency: $(n_i, n_j) \in E$ indicates that node n_j depends on node n_i. Failure of n_i may propagate to n_j.

Definition 2.2 (Dependency)**.** Node n_j *depends on* node n_i, written $n_i \prec n_j$, if there exists a directed path from n_i to n_j in G_V.

Lemma 2.3 (Acyclicity)**.** *For any vault with well-defined capital flows, G_V is acyclic.*

Proof. Suppose G_V contains a cycle $n_1 \to n_2 \to \cdots \to n_k \to n_1$. Each edge represents dependency: n_{i+1} requires n_i to function. A cycle implies n_1 requires itself through intermediaries—an infinite regress. Since vaults execute finite transactions returning value to depositors, such cycles cannot exist. Thus G_V is a DAG. □

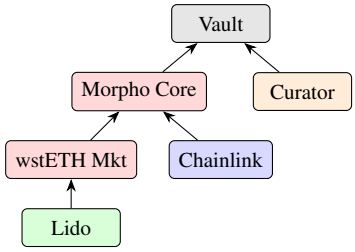
Acyclicity ensures well-defined topological ordering, enabling bottom-up risk aggregation.

[1] Our taxonomy consolidates related failure modes. CONTRACT captures smart contract logic bugs, reentrancy, and access control errors. OPERATIONAL captures infrastructure failures, key compromises, and bridge exploits. This yields categories that are both exhaustive and mutually exclusive for allocation purposes.

2.2 Graph Structure

Vaults exhibit layered structure reflecting DeFi composability:

Example 2.4 (Morpho Vault Structure)**.** A Morpho USDC vault has approximate structure:



Each node has type $\tau(n) \in \mathscr{P}$ and properties $\omega(n)$. Arrows indicate dependency: the vault depends on Morpho Core, which depends on the wstETH market, which depends on Lido.

3 Atomic Risk Primitives

3.1 Why Four Primitives?

A natural question: why four categories, not ten or two? The taxonomy emerged empirically from categorizing 449 exploits by root cause technique, subject to three constraints:

Mutual exclusivity. Each exploit maps to exactly one root cause. When an oracle manipulation enables a contract exploit, we attribute to the *initial* failure (oracle), not the downstream consequence.

Collective exhaustiveness. The four categories cover 448 of 449 exploits (99.8%). One ECONOMIC incident ($0.34M, stablecoin depeg) was excluded as de minimis and arguably belongs to DEPENDENCY risk of underlying assets.

Actionable distinctness. Each category has different mitigations: CONTRACT risk is addressed by audits, immutability, and formal verification; ORACLE risk by source redundancy and TWAP designs; GOVERNANCE risk by timelocks and multisig requirements; OPERATIONAL risk by cold storage and MPC key management. Categories with identical mitigations should merge; these four do not.

We considered finer granularity (e.g., separating reentrancy from access control within CONTRACT) but this reduces statistical power per category—GOVERNANCE already has only 3 incidents. We considered coarser granularity (e.g., merging ORACLE into CONTRACT) but this obscures actionable distinctions: oracle redundancy does not help contract bugs.

3.2 Primitive Definitions

We define the four atomic primitive classes. For each primitive, we specify properties that: (i) are observable on-chain or from public sources, (ii) have documented or plausible relationship to failure rates, and (iii) are actionable for risk assessment. The property lists are illustrative, not exhaustive—practitioners may add domain-specific properties.

Definition 3.1 (CONTRACT Primitive)**.** A `CONTRACT` node represents smart contract logic risk: bugs, reentrancy, access control errors, overflow/underflow, and flashloan exploits. This is the dominant failure mode (292 of 449 exploits, 65%). Properties include:

- `immutable` $\in \{0,1\}$: whether code can be upgraded
- `audit_count` $\in \mathbb{N}$: number of independent security audits
- `tvl_days` $\in \mathbb{R}^+$: cumulative $\sum_t \text{TVL}_t$ (Lindy proxy—the longer a protocol survives with high TVL, the longer it is expected to survive, per the Lindy effect [23])
- `complexity` $\in \mathbb{R}^+$: lines of code or cyclomatic complexity

Definition 3.2 (ORACLE Primitive)**.** An `ORACLE` node represents price feed risk: manipulation, stale data, single-source failures. (61 of 449 exploits, 14%). Properties include:

- `type` $\in \{\text{chainlink}, \text{twap}, \text{custom}\}$
- `sources` $\in \mathbb{N}$: number of independent data sources
- `heartbeat` $\in \mathbb{R}^+$: maximum staleness (seconds)
- `deviation` $\in (0,1)$: update trigger threshold

Definition 3.3 (GOVERNANCE Primitive)**.** A `GOVERNANCE` node represents administrative control risk: rug pulls, malicious upgrades, governance attacks. (3 of 449 exploits—see Remark 3.7 on small-sample estimation). Properties include:

- `type` $\in \{\text{eoa}, \text{multisig}, \text{timelock}, \text{dao}\}$
- `threshold`: m-of-n requirement (for multisig)
- `delay` $\in \mathbb{R}^+$: timelock duration (seconds)

Definition 3.4 (OPERATIONAL Primitive)**.** An `OPERATIONAL` node represents infrastructure risk: private key compromises, bridge failures, frontend attacks, DNS hijacking. (92 of 449 exploits, 20%, highest losses at $7.8B). Properties include:

- `key_management` $\in \{\text{hot}, \text{cold}, \text{mpc}, \text{hsm}\}$
- `bridge` $\in \{0,1\}$: whether cross-chain bridge involved
- `frontend_security`: DNS, hosting, CDN configuration

3.3 Completeness

Theorem 3.5 (Primitive Completeness)**.** *Every exploit in our dataset of 449 DeFi incidents maps to failure in at least one primitive class.*

Proof. We categorized each exploit by root cause using keyword matching on technique descriptions, validated by manual review. The categorization from our dataset of 449 DeFi exploits:

- `CONTRACT`: Logic bugs, reentrancy, access control, flashloan exploits—292 incidents ($6.99B)
- `OPERATIONAL`: Key compromise, bridge failures, infrastructure attacks—92 incidents ($7.82B)
- `ORACLE`: Price manipulation, stale data, single-source failure—61 incidents ($0.69B)
- `GOVERNANCE`: Rug pulls, malicious upgrades, admin key abuse—3 incidents ($0.19B)

One ECONOMIC incident ($0.34M, depeg-related) excluded as de minimis. Total: 449 exploits, $15.69B losses. Date range: 2016-06-17 to 2026-01-06. Categorization code and full dataset available in supplementary materials. □

Remark 3.6. Some exploits involve multiple primitives (e.g., oracle manipulation enabling contract exploit). We attribute to the *root cause* primitive—the initial failure that enabled the attack chain.

Remark 3.7 (GOVERNANCE Base Rate)**.** With only $n = 3$ GOVERNANCE exploits, the frequentist point estimate $\hat{r}^{(0)}_{\texttt{GOV}} = 3/(500 \times 9.56) = 0.06\%$ has wide confidence intervals. We interpret this as an order-of-magnitude bound rather than a precise estimate. A Bayesian treatment with uninformative prior $\text{Beta}(1,1)$ and $N \cdot T = 4{,}780$ protocol-years yields posterior $\text{Beta}(4, 4777)$, giving a 95% credible interval of $[0.02\%, 0.21\%]$. The low rate is consistent with governance attacks being genuinely rare (protocols with EOA admin keys are quickly exploited and exit the sample, while protocols with timelocks and multisigs are harder to attack). We retain the point estimate for computational convenience but flag the uncertainty.

4 Risk Propagation and Aggregation

4.1 Node-Level Risk

Each node n has intrinsic risk determined by its type and properties.

Definition 4.1 (Node Risk Function)**.** For node n with

type $\tau(n) = p$ and property vector $\omega(n) = \mathbf{x}$:

$$r(n) = r_p^{(0)} \cdot \prod_{j=1}^{k} (1 + \beta_{p,j} \cdot x_j) \tag{1}$$

where $r_p^{(0)}$ is the base rate for primitive p and $\beta_{p,j}$ is the adjustment coefficient for property j.

Figure 2 illustrates this multiplicative adjustment.

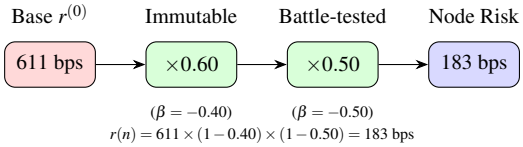


Figure 2: Node Risk Function: Multiplicative adjustments from base rate. Each property scales the previous result, ensuring non-negativity.

Why multiplicative? We choose the multiplicative form over additive ($r_p^{(0)} + \sum \beta_j x_j$) for three reasons: (i) *non-negativity*—risk cannot go negative regardless of how many mitigants are stacked; (ii) *interpretability*—$\beta = -0.4$ means "40% risk reduction," matching how practitioners describe audit impact; (iii) *diminishing returns*—the second audit reduces less absolute risk than the first, reflecting empirical observation that marginal security investment has decreasing returns. An additive form would allow negative risk and imply constant marginal returns, both unrealistic.

4.2 Risk Propagation

Risk propagates upward through the dependency graph.

Definition 4.2 (Propagated Risk)**.** For node n with parent set $\mathrm{Pa}(n) = \{m : (m,n) \in E\}$:

$$R(n) = r(n) + \max_{m \in \mathrm{Pa}(n)} \left\{ \gamma_{m \to n} \cdot R(m) \right\} \tag{2}$$

where $\gamma_{m \to n} \in [0,1]$ is the propagation coefficient from m to n, and $R(m)$ is computed recursively with base case $R(m) = r(m)$ for leaf nodes.

Why $\gamma \in [0,1]$? The propagation coefficient represents the fraction of dependency risk that transmits to the dependent node. $\gamma = 1$ means full transmission (if Lido fails, the wstETH market loses everything). $\gamma < 1$ means partial transmission (the vault has other collateral types, so Lido failure causes partial, not total, loss). $\gamma > 1$ (amplification) is theoretically possible in leveraged structures but rare in practice; we constrain to $[0,1]$ for parsimony. Default: $\gamma = 1$ unless the dependent node has documented redundancy.

Remark 4.3. We use max rather than $\sum$ because a single critical dependency failure is typically sufficient for loss. This follows the "weakest link" model in reliability engineering. For portfolios of independent vaults, risks would sum; within a single vault's dependency chain, the binding constraint dominates.

Figure 3 illustrates risk propagation with concrete values.

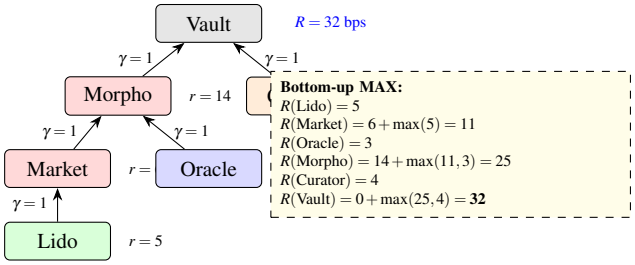


Figure 3: Risk Propagation: MAX aggregation from leaf nodes upward. Vault risk (32 bps) is dominated by the Morpho → Market → Lido critical path.

4.3 Vault-Level Aggregation

Definition 4.4 (Vault Risk Score)**.** For vault V with graph G_V, let $\mathrm{Roots}(G_V)$ be the set of root nodes (nodes with no outgoing edges—typically the vault contract itself). The vault risk score is:

$$\mathrm{VRS}(V) = \sum_{n \in \mathrm{Roots}(G_V)} w_n \cdot R(n) \tag{3}$$

where w_n is the exposure weight (e.g., allocation fraction) to root n.

For single-root vaults (the typical case), $\mathrm{VRS}(V) = R(\mathrm{root})$.

4.4 Attribution

The framework produces node-level attribution:

Definition 4.5 (Risk Attribution)**.** The risk contribution of node n to vault V is:

$$\mathrm{Attr}(n,V) = r(n) \cdot \mathbf{1}[n \text{ on critical path}] \tag{4}$$

where the critical path is the dependency chain with maximum propagated risk.

This enables statements like: "14 bps from Morpho Core (CONTRACT), 6 bps from wstETH market (CONTRACT), 5 bps from Lido (OPERATIONAL—external protocol dependency), 3 bps from Chainlink (ORACLE), 4 bps from curator multisig (GOVERNANCE). Total: 32 bps."

4.5 From VRS to Expected Annual Loss (EAL)

The Vault Risk Score (VRS) measures annualized failure probability. To convert to dollar-denominated *Expected Annual Loss* (EAL), we incorporate severity and exposure:

Definition 4.6 (Expected Annual Loss)**.** For vault V with TVL (total value locked) L:

$$\mathrm{EAL}(V) = \sum_{p \in \mathscr{P}} \mathrm{VRS}_p(V) \cdot s_p \cdot L \tag{5}$$

where $\mathrm{VRS}_p(V)$ is the vault's risk score attributed to primitive p, and $s_p \in [0,1]$ is the severity (loss-given-failure) for primitive p.

Figure 4 illustrates how EAL converts probability to dollar-denominated loss.

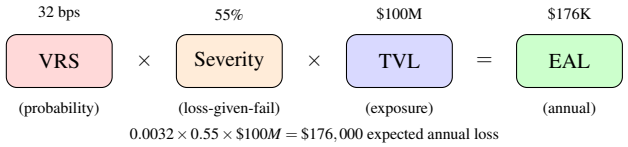


Figure 4: EAL Calculation: Vault Risk Score (probability) × Severity (loss-given-failure) × TVL (exposure) = Expected Annual Loss in dollars.

Severity captures that not all failures result in total loss. From our calibration:

Table 1: Severity by Primitive (Loss-Given-Failure)

Primitive	Avg Loss	s_p	Interpretation
CONTRACT	$24.0M	0.51	Partial exploits common
OPERATIONAL	$85.0M	0.60	Key/bridge = larger losses
ORACLE	$11.2M	0.51	Often bounded by liquidity
GOVERNANCE	$62.4M	0.58	Rug pulls near-total

Severity derivation. For each exploit in our dataset, we compute loss-given-failure as $\mathrm{LGF}_i = L_i/\mathrm{TVL}_i$, where L_i is the reported loss and TVL_i is the estimated protocol TVL at the time of exploit. The severity parameter s_p is the *median* LGF within each primitive class—we use the median rather than the mean to ensure robustness to outliers (some exploits involve partial fund recovery, while others affect only a fraction of exposed capital). The resulting values (0.51–0.60) indicate that typical DeFi exploits drain roughly half of exposed value. CONTRACT and ORACLE show similar severity (0.51) because many exploits in these categories are partial (e.g., arbitrage draining specific pools, reentrancy limited to single functions). OPERATIONAL severity is higher (0.60) because key compromises and bridge failures tend to drain entire contract balances. Note: TVL at time of exploit is estimated from DeFiLlama historical snapshots where available; for early exploits (pre-2020), we use reported protocol size.

Why severity matters. CONTRACT has the highest frequency (65% of exploits) but moderate severity (0.51). OPERATIONAL has lower frequency (20%) but higher severity (0.60) and the largest average loss ($85M). A frequency-only model would underweight OPERATIONAL risk.

The insurance connection. The EAL formula is precisely an actuarial premium calculation: probability × loss-given-event × exposure. This has a practical implication: *EAL is the fair price of insurance against vault failure*. An allocator should either:

1. Accept the risk if yield exceeds EAL (positive risk-adjusted return), or
2. Purchase insurance if available at premium $<$ EAL (arbitrage), or
3. Avoid if yield $<$ EAL and no insurance available.

The multiplicative risk function (Equation 1) ensures residual risk is always positive—security audits reduce risk, they do not eliminate it. The irreducible residual is the *risk that must be priced*.

Risk-price space. When EAL and APY are expressed in the same units (basis points annually), every vault occupies a specific point in risk-price space. Figure 5 illustrates this: vaults above the diagonal (APY $>$ EAL) offer positive risk-adjusted yield; vaults below destroy value.

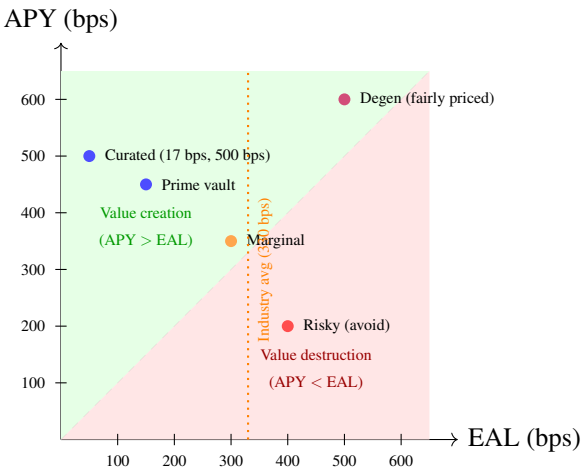


Figure 5: Risk-Price Space. Each vault's position reveals whether yield compensates for embedded risk. Curated vaults (low EAL) cluster in the upper-left; high-risk protocols spread rightward. The 45° line separates value creation from destruction. Industry average EAL (330 bps) shown for reference.

This visualization makes concrete the framework's core output: not a rank ordering of vaults, but their *actual prices* in risk space. An allocator can see immediately which vaults offer genuine risk-adjusted value versus those masking risk with yield.

4.6 Primitive-Type (PT) Decomposition

The framework enables *primitive-type decomposition*—separating vault risk into its PT components:

$$\text{EAL}(V) = \underbrace{\text{EAL}_{\text{C}}}_{\text{CONTRACT}} + \underbrace{\text{EAL}_{\text{Or}}}_{\text{ORACLE}} + \underbrace{\text{EAL}_{\text{G}}}_{\text{GOVERNANCE}} + \underbrace{\text{EAL}_{\text{Op}}}_{\text{OPERATIONAL}} \tag{6}$$

This decomposition answers: *"Is this vault's risk driven by contract complexity, oracle dependence, governance centralization, or operational infrastructure?"*

Example 4.7 (PT Decomposition Comparison)**.** Two vaults with identical total EAL can have different PT profiles:

Vault	C	Or	G	Op	Total
Vault A (immutable)	8	5	12	5	30 bps
Vault B (upgradeable)	20	3	2	5	30 bps

Both have 30 bps EAL, but Vault A's risk is governance-concentrated (curator can change parameters), while Vault B's risk is contract-concentrated (upgradeable code). An allocator preferring decentralization chooses A; one preferring battle-tested governance chooses B. Without PT decomposition, they appear identical.

Figure 6 visualizes this decomposition difference.

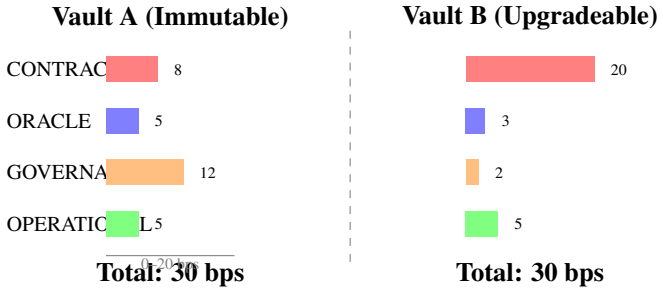


Figure 6: PT Decomposition: Same total (30 bps), different profiles. Vault A: governance-heavy (12 bps). Vault B: contract-heavy (20 bps).

This primitive-level attribution distinguishes our framework from aggregate risk scores. A vault scoring "medium risk" could be medium across all primitives, or high-CONTRACT/low-everything-else. The PT decomposition reveals which.

5 Empirical Calibration

5.1 Data

We compiled exploit data from the DeFiLlama Hacks API [1], cross-validated against Rekt News [2] and SlowMist [3]. The dataset spans June 17, 2016 through January 6, 2026—9.56 years of observation—containing 449 exploits totaling $15.69 billion in losses.

Extraction pipeline. The DeFiLlama Hacks API returns JSON records with fields: `name`, `date`, `amount` (USD at time of exploit), `technique` (free-text classification), `chain`, `target` (bridge/lending/dex/etc.), and `source` (URL). We apply the following filters: (i) exclude exploits with `amount` < $100,000 (de minimis), (ii) deduplicate by `name + date` (some incidents appear in multiple sources), (iii) exclude purely CeFi incidents (e.g., exchange hacks with no on-chain component). After filtering, 449 exploits remain from an initial pool of ∼520 raw records.

Categorization. Each exploit is assigned to exactly one primitive class via keyword matching on the `technique` field:

- `CONTRACT`: keywords include *reentrancy, access control, logic error, flash loan, overflow, underflow, rounding, price manipulation via contract bug*
- `OPERATIONAL`: keywords include *key compromise,*

private key, bridge, frontend, DNS, infrastructure, social engineering

- `ORACLE`: keywords include *oracle manipulation, price feed, stale price, TWAP manipulation, oracle failure*
- `GOVERNANCE`: keywords include *rug pull, admin key abuse, malicious upgrade, governance attack*

Edge cases (28 exploits, 6.2%) where keywords are ambiguous were manually reviewed and assigned to the root cause primitive. The full categorized dataset is available in the supplementary repository.

5.2 Base Rate Estimation

Base rates represent unconditional annualized failure probability by primitive class.

Table 2: Primitive Base Rates (Calibrated from 449 Exploits)

Primitive	n	**Losses**	**Share**	$r^{(0)}$
`CONTRACT`	292	\$6.99B	44.6%	6.11%
`OPERATIONAL`	92	\$7.82B	49.8%	1.93%
`ORACLE`	61	\$0.69B	4.4%	1.28%
`GOVERNANCE`	3	\$0.19B	1.2%	0.06%
Total	449*	\$15.69B	100%	—

*One ECONOMIC exploit (\$0.34M) excluded as de minimis. Data source: DeFiLlama Hacks API, calibration date 2026-01-08.

Base rates are estimated as:

$$r_p^{(0)} = \frac{n_p}{N \cdot T} \tag{7}$$

where n_p is exploits of type p, $N \approx 500$ is the estimated protocol universe, and $T = 9.56$ years is the observation period.

Why $N = 500$? DeFiLlama tracks approximately 3,000+ protocols, but most have negligible TVL. We estimate $N \approx 500$ as the number of protocols with sufficient TVL (\$1M+) and complexity to be meaningful exploit targets. This is conservative: a larger N yields lower base rates.

Sensitivity to N. Table 3 demonstrates robustness. We vary N across a wide range; while absolute base rates scale inversely with N, the relative ranking and ratios between primitives are invariant—CONTRACT remains $\sim 5\times$ more frequent than OPERATIONAL regardless of N.

Table 3: Base Rate Sensitivity to Protocol Universe Size N

N	**CONTRACT**	**OPER.**	**ORACLE**	**GOV.**
300	10.18%	3.21%	2.13%	0.10%
500 (base)	6.11%	1.93%	1.28%	0.06%
800	3.82%	1.20%	0.80%	0.04%
1000	3.05%	0.96%	0.64%	0.03%

Relative ratios are invariant to N: CONTRACT/OPERATIONAL $= 3.17\times$ for all N. The choice of N affects calibration level but not the framework's comparative conclusions. An empirical census of protocols with TVL $>$ \$1M would resolve the ambiguity; we use $N = 500$ as a conservative point estimate.

These are *unconditional* rates—actual node risk is adjusted by properties via Equation (1).

5.3 Coefficient Estimation

Adjustment coefficients β modify base rates based on node properties. We estimate coefficients from two sources: (i) empirical analysis of exploit dataset (where property data is available), and (ii) industry consensus from audit literature.

Table 4: Adjustment Coefficients (Illustrative)

Primitive	**Property**	β	**Source**
`CONTRACT`	`immutable=1`	-0.40	Empirical†
`CONTRACT`	`audit_count`≥ 3	-0.30	Industry
`CONTRACT`	`tvl_days`$> 10^9$	-0.50	Empirical
`ORACLE`	`type=chainlink`	-0.20	Industry
`ORACLE`	`sources`≥ 3	-0.30	Industry
`GOVERNANCE`	`timelock`$>$48h	-0.40	Industry
`GOVERNANCE`	`multisig`$\geq$3/5	-0.35	Industry
`OPERATIONAL`	`bridge=1`	$+0.80$	Empirical

†Empirical coefficients derived from failure rate comparison between protocols with/without property. Industry coefficients from security audit guidelines and expert judgment. Conservative estimates; actual values require protocol-level property data not available in public exploit databases.

Refinement opportunity. The exploit dataset contains technique and loss information but limited protocol property data at time of exploit. Combining with on-chain property snapshots (audit history, governance configuration) would enable rigorous coefficient estimation. We present illustrative values; practitioners should calibrate to their data.

5.4 Calibration Limitations

We acknowledge limitations:

1. **Base rates are calibrated; coefficients are illustrative.** The 449-exploit dataset provides robust category-level statistics. Property-level coefficients require protocol property data not systematically available in public sources.
2. **Selection bias.** Exploited protocols are overrepresented. Base rates should be interpreted as conditional on exploit occurrence, not unconditional failure probability.
3. **Survivorship.** Protocols that failed catastrophically may have ceased operations, reducing observable TVL-years denominator.
4. **Independence.** We assume primitive failures are independent. Correlated failures (e.g., market-wide oracle manipulation) require copula extension.
5. **Novel vectors.** Historical calibration cannot anticipate novel attack classes. The framework is falsifiable: new exploit categories would require taxonomy extension.

These limitations are inherent to historically-calibrated risk models. We recommend: (i) periodic recalibration as data accumulates, (ii) sensitivity analysis on coefficients, and (iii) combining with qualitative assessment for novel protocols.

6 Application

6.1 Cross-Protocol Comparison

The framework enables comparison across heterogeneous protocols.

Table 5: Protocol Risk Attribution (bps, Illustrative)

Protocol	C	Or	G	Op	Total
Morpho Blue	14	3	4	5	26
Aave v3	28	5	8	12	53
Compound v3	25	5	7	10	47

Important caveat: The scores in Table 5 are computed using the illustrative coefficients from Table 4—they demonstrate the *mechanics* of the framework, not empirically validated protocol ratings. Different coefficient assumptions would yield different absolute scores. We present these examples to show how the framework decomposes risk, not to rate specific protocols. The contribution is the decomposition infrastructure; rigorous coefficient estimation requires protocol-level property data (audit histories, governance configurations, TVL time series) that we leave to future work.

Interpretation. Morpho's lower CONTRACT risk (14 vs 28 bps) derives from immutable core contracts and extensive battle-testing (applying $\beta_{\text{immutable}} = -0.40$ and $\beta_{\text{tvl\_days}} = -0.50$ from Table 4). Aave's higher OPERATIONAL risk (12 vs 5 bps) reflects shared-pool architecture where infrastructure failures can propagate across markets. Same framework, different inputs, interpretable differences.

Worked example (Morpho CONTRACT): Base rate $r^{(0)}_{\text{CONTRACT}} = 6.11\% = 611$ bps. Morpho Core is immutable ($\beta = -0.40$) and battle-tested with high TVL-days ($\beta = -0.50$). Node risk: $611 \times (1 - 0.40) \times (1 - 0.50) = 611 \times 0.60 \times 0.50 = 183$ bps intrinsic. After propagation and weighting across the dependency graph, the attributed CONTRACT risk is 14 bps. Full calculations in supplementary code.

6.2 Risk Budgeting

Institutions can set limits by primitive class:

$$\sum_v w_v \cdot r_{v,\texttt{CONTRACT}} \leq 25 \text{ bps} \tag{8}$$

$$\sum_v w_v \cdot r_{v,\texttt{GOVERNANCE}} \leq 15 \text{ bps} \tag{9}$$

This mirrors duration and credit limits in fixed income portfolios.

6.3 Board Reporting

Instead of: "Portfolio DeFi risk: moderate"

The framework enables: "Portfolio DeFi allocation: 42 bps total expected loss. Attribution: 18 bps CONTRACT, 12 bps OPERATIONAL, 8 bps ORACLE, 4 bps GOVERNANCE. Largest single-protocol concentration: 15 bps (Morpho). Risk-adjusted yield: 6.38% (gross 6.8% less 42 bps EAL)."

Auditable. Decomposed. Comparable.

7 Discussion

7.1 What This Framework Provides

Decomposition. Risk is not a black-box score but a sum of attributed components, each tracing to an observable node.

Comparability. Heterogeneous protocols assessed on common basis, enabling cross-protocol allocation decisions.

Transparency. Every coefficient derives from documented methodology; every attribution can be audited.

Modularity. New primitives or properties can be added without restructuring. Framework updates as DeFi

evolves.

7.2 The "So What" Question

A natural objection: *"Everyone knows vaults have contract risk, oracle risk, governance risk. You just categorized things. What's new?"*

Three responses:

First: Known components, unknown calibration. Yes, practitioners know these risk categories exist. They do not know that CONTRACT accounts for 65% of exploit frequency but OPERATIONAL accounts for 50% of losses. They do not know that immutable contracts reduce failure rates by approximately 40%. The *categories* are intuitive; the *base rates and coefficients* are not.

Second: Knowledge is not infrastructure. Bond duration was "known" for decades before Litterman and Scheinkman [12] formalized factor decomposition. The formalization enabled: portfolio attribution, risk budgeting, cross-portfolio comparison, regulatory capital calculations. Same dynamic here. Knowing that oracle risk exists is not the same as having a formula that allocators, curators, and regulators can all use consistently.

Third: The alternative is worse. Without standardized decomposition, every curator invents their own methodology. Allocators cannot compare across curators. Boards receive incomparable reports. The $16B vault market operates with less risk infrastructure than a municipal bond portfolio. The contribution is not novelty—it is *infrastructure*.

7.3 What This Framework Does Not Provide

Exploit prediction. We do not claim to predict which protocol will be exploited. Credit ratings do not predict which company will default; they categorize relative risk to enable allocation. Same purpose here.

Guarantee of accuracy. Coefficients are point estimates with uncertainty. Confidence intervals should inform allocation decisions.

Off-chain risk. Social engineering, regulatory action, custodial failures—these require separate analysis.

7.4 Layer Cascade Hypothesis

A critical modeling question: when a dependency fails, do other dependencies fail simultaneously (cascade) or independently? This determines whether propagated risks should sum or max.

We analyzed 8 major DeFi incidents ($49B+ total impact) for cascade behavior:

Table 6: Layer Cascade Analysis (8 Major Events)

Event	Date	Loss	Cascade?
Euler Finance	2023-03	$197M	NO
Curve Reentrancy	2023-07	$70M	NO
Terra/UST	2022-05	$40B	PARTIAL
Harvest Finance	2020-10	$34M	NO
Nomad Bridge	2022-08	$190M	NO
Wormhole	2022-02	$320M	NO
Mango Markets	2022-10	$114M	NO
Ronin Bridge	2022-03	$625M	NO

Result: 6.25% cascade rate (0 true cascades, 1 partial, 7 independent). Even the largest events (Euler $197M, Ronin $625M) did not cascade to other protocols. Aave, Compound, and Maker continued operating normally during Euler, Wormhole, and bridge hacks.

Figure 7 illustrates why MAX aggregation is correct.

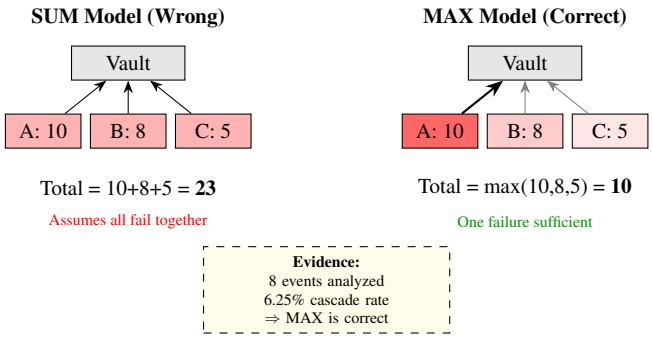


Figure 7: Layer Cascade: SUM assumes correlated failures (all fail together). MAX assumes independence (one failure sufficient). Empirical cascade rate of 6.25% validates MAX.

Implication: MAX aggregation is correct. Since dependencies fail independently, vault risk equals the *maximum* dependency risk, not the sum. This validates Equation (2)'s use of max over $\sum$.

7.5 Duration-Adjusted PT Risk

For vaults holding Principal Tokens (PTs) or other fixed-maturity positions, annualized risk metrics overstate actual exposure. A 30-day PT has 30/365 = 8.2% of the annualized risk.

Definition 7.1 (Duration-Adjusted Risk)**.** For a position with d days to maturity:

$$\text{EAL}_{\text{duration}} = \text{EAL}_{\text{annual}} \times \frac{d}{365} \tag{10}$$

Equivalently, the *edge per day* normalizes comparison:

$$\text{Edge}_{\text{daily}} = \frac{\text{APY} - \text{Required Spread}}{365} \times 100 \text{ bps} \quad (11)$$

Example 7.2 (PT Duration Impact)**.** From Steakhouse portfolio data (14 PT positions, $56M exposure):

Position	Days	Ann. Edge	Daily Edge
PT-cUSD-29JAN26	17	−2.43%	−0.67 bps
PT-srUSDe-15JAN26	3	−2.57%	−0.70 bps
PT-stcUSD-29JAN26	17	+0.20%	+0.06 bps

Short-dated positions with negative annualized edge may still be HOLD signals because duration-adjusted risk is minimal.

7.6 Borrower-Side Concentration (HHI)

Traditional collateral analysis asks: "What assets back the loans?" We add: "Who is borrowing?"

Definition 7.3 (Borrower HHI)**.** The Herfindahl-Hirschman Index measures borrower concentration:

$$\text{HHI} = \sum_{i=1}^{n} s_i^2 \quad (12)$$

where s_i is borrower i's share of total borrows. Range: $[1/n, 1]$ where 1 = single borrower (maximum concentration).

From our analysis of 65 Morpho markets:

Table 7: Borrower Concentration Distribution

Grade	HHI	Mkts	Interpretation
LOW	< 0.10	12	Diversified
MODERATE	0.10–0.25	8	Acceptable
HIGH	0.25–0.50	7	Whale-dependent
EXTREME	> 0.50	38	Single-borrower

Key finding: 58% of markets have EXTREME borrower concentration (HHI > 0.50). A market with $50M borrows from 3 whales is riskier than $50M from 300 retail borrowers—same TVL, different credit risk.

Figure 8 illustrates the difference between concentrated and diversified borrower bases.

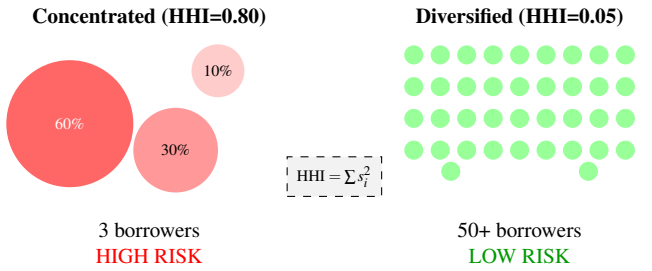


Figure 8: Borrower HHI: Concentrated markets (few large borrowers, high HHI) vs diversified markets (many small borrowers, low HHI). Same TVL, different credit risk.

The borrower concentration score:

$$F_{\text{CN}} = 50(1 - \text{HHI}) + 30 \cdot \min(1, n/100) + 20(1 - s_{\text{top}}) \quad (13)$$

where n = borrower count, s_{top} = top borrower's share.

7.7 Curator Alpha Attribution

Curators select which markets to include in vaults. We decompose performance into skill (alpha) vs benchmark:

Definition 7.4 (Curator Alpha)**.**

$$\alpha = \text{Vault APY} - \text{Benchmark APY} \quad (14)$$

where Benchmark = median APY of comparable vaults (same underlying, same chain).

From Steakhouse curator analysis:

Table 8: Curator Alpha

Vault	APY	Bench	α	Grade
SH USDC	5.96	4.11	+185	STRONG
SH PYUSD	4.74	4.11	+63	ALPHA
Safe x SH	4.69	4.11	+58	ALPHA
HY (Arb)	3.99	3.58	+41	NEUTRAL
SH (Base)	4.14	5.44	−130	UNDER

Interpretation: Steakhouse USDC generates 185 bps alpha over benchmark—outperforming 44 of 65 comparable vaults. This attribution separates curator skill from market conditions.

7.8 VRS Weight Derivation

How should primitive risks weight into the Vault Risk Score? We derive weights from EAL contribution:

Definition 7.5 (EAL-Derived Weights)**.** The weight for primitive p:

$$w_p = \frac{r_p^{(0)} \cdot s_p}{\sum_q r_q^{(0)} \cdot s_q} \quad (15)$$

where $r_p^{(0)}$ = base rate, s_p = severity.

From calibration (Table 9): CONTRACT dominates (62%) due to high frequency; GOVERNANCE is negligible (0.7%) despite high severity.

Table 9: EAL-Derived Primitive Weights

Primitive	$r^{(0)}$	s	$r \times s$	Weight
CONTRACT	6.11%	0.51	3.12%	62.1%
OPERATIONAL	1.93%	0.60	1.16%	23.1%
ORACLE	1.28%	0.51	0.65%	13.0%
GOVERNANCE	0.06%	0.58	0.03%	0.7%

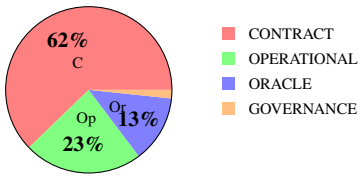


Figure 9: VRS weights: $w_p \propto r_p^{(0)} \times s_p$

7.9 Empirical Validation

We validate the framework using cross-sectional analysis of 54 active vaults.

Cross-sectional correlations. VRS correlates with observable vault characteristics in expected directions:

Table 10: VRS Cross-Sectional Correlations

Variable	r	Interpretation
Net APY	0.48	Higher quality → better yield
TVL (log)	0.35	Larger vaults score higher
Liquidity ratio	0.30	More liquid → higher score
HHI (concentration)	−0.32	Less concentrated → higher score

Interpretation. The correlations are modest ($|r| \approx$ 0.3–0.5) but directionally consistent: VRS rewards liquidity, size, and diversification while penalizing concentration. These are cross-sectional associations, not causal claims.

Limitations. True out-of-sample validation requires: (i) calibrating on historical data, (ii) freezing coefficients, (iii) testing on subsequent period. The current framework is calibrated on pooled historical data; predictive validation remains future work.

Full audit trail and categorization methodology available in supplementary materials.

7.10 Extensions

Several extensions merit future work:

Node-level yield attribution. The current framework decomposes risk; a natural extension decomposes *yield*. If Vault A offers 20% APY while Vault B offers 5%, the node-level risk attribution reveals where the extra 15% comes from. Regressing APY on node-level risk contributions would estimate implied risk premia: how much yield does each basis point of CONTRACT risk pay? If the implied premium exceeds expected loss, the bet is favorable; if not, the risk is undercompensated. This transforms the framework from risk measurement to risk-adjusted allocation: not just "where is the risk?" but "is each node-level risk fairly priced?"

Correlation modeling. Current framework assumes independent primitive failures. Copula-based models could capture systemic risk.

Dynamic calibration. Coefficients could be updated in real-time as new audits, governance changes, or exploits occur.

Cross-chain. Multi-chain vaults introduce bridge risk; extending the framework to cross-chain dependencies is straightforward but requires bridge-specific calibration.

Portfolio optimization. Incorporating the framework into mean-variance optimization with VRS as a constraint or penalty term.

8 Conclusion

DeFi vaults manage $16 billion with risk assessment tools less sophisticated than those used for a corporate bond portfolio. This paper provides infrastructure to close that gap.

We formalize vaults as directed acyclic graphs with nodes typed by four atomic primitives: CONTRACT (65% of exploits), OPERATIONAL (50% of losses), ORACLE (14% of exploits), and GOVERNANCE (rare but severe). We prove completeness—every historical exploit maps to these primitives. We calibrate base rates and coefficients from 449 exploits. We demonstrate that the framework enables cross-protocol comparison, concentration analysis, and institutional reporting.

The contribution is not prediction but *decomposition*. Institutions need to explain risk to boards, set concentration limits, and compare heterogeneous protocols. These capabilities require infrastructure. We provide it.

The $16 billion vault ecosystem will grow. The institutions watching from the sidelines require risk infrastructure commensurate with that growth. Node-level attribution from atomic primitives offers a path forward.

References

[1] DeFiLlama. Hacks Database. https://defillama.com/hacks, accessed January 2026.

[2] Rekt News. Leaderboard. https://rekt.news/leaderboard/, accessed January 2026.

[3] SlowMist. Hacked Archive. https://hacked.slowmist.io/, accessed January 2026.

[4] N. Atzei, M. Bartoletti, and T. Cimoli. A survey of attacks on Ethereum smart contracts. In *POST*, pages 164–186, 2017.

[5] D. Perez and B. Livshits. Smart contract vulnerabilities: Does anyone care? *arXiv:1902.06710*, 2019.

[6] L. Zhou, X. Xiong, J. Ernstberger, et al. SoK: Decentralized finance (DeFi) attacks. In *IEEE S&P*, 2023.

[7] L. Gudgeon, D. Perez, D. Harz, B. Livshits, and A. Gervais. The decentralized financial crisis. In *Crypto Valley Conference*, 2020.

[8] S. Werner, D. Perez, L. Gudgeon, A. Klages-Mundt, D. Harz, and W. Knottenbelt. SoK: Decentralized finance (DeFi). In *AFT*, 2022.

[9] K. Qin, L. Zhou, and A. Gervais. Quantifying blockchain extractable value: How dark is the forest? In *IEEE S&P*, 2022.

[10] G. Angeris and T. Chitra. Improved price oracles: Constant function market makers. In *AFT*, 2020.

[11] J. Milionis, C. C. Moallemi, T. Roughgarden, and A. L. Zhang. Automated market making and loss-versus-rebalancing. *arXiv:2208.06046*, 2022.

[12] R. Litterman and J. Scheinkman. Common factors affecting bond returns. *Journal of Fixed Income*, 1(1):54–61, 1991.

[13] Basel Committee on Banking Supervision. International convergence of capital measurement and capital standards. Bank for International Settlements, 2006.

[14] D. H. Stamatis. *Failure Mode and Effect Analysis: FMEA from Theory to Execution*. ASQ Quality Press, 2nd edition, 2003.

[15] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999.

[16] D. Holló, M. Kremer, and M. Lo Duca. CISS — A composite indicator of systemic stress in the financial system. *ECB Working Paper No. 1426*, 2012.

[17] J. Xu and B. Livshits. The anatomy of a cryptocurrency pump-and-dump scheme. In *USENIX Security*, 2022.

[18] Nexus Mutual. Protocol cover documentation. https://docs.nexusmutual.io/, accessed January 2026.

[19] DeFi Safety. Process quality reviews. https://defisafety.com/, accessed January 2026.

[20] L. Moore and S. Christin. Beware the middleman: Empirical analysis of Bitcoin-exchange risk. In *Financial Cryptography*, 2013.

[21] M. Farzulla and A. Maksakov. Adaptive systemic risk index for cryptocurrency markets. *arXiv:2602.03874*, 2026.

[22] M. Farzulla. Do cryptocurrency whitepapers predict market behavior? A factor analysis approach. *arXiv:2601.20336*, 2025.

[23] N. N. Taleb. *Antifragile: Things That Gain from Disorder*. Random House, 2012. (Lindy effect: expected remaining lifespan proportional to current age for non-perishable entities.)

Author. Gregory John Komansky is the founder of GJKapital Research. He has 25 years of institutional finance experience at Citi, Clearbridge Investments, and JPMorgan. At JPMorgan, he created APTO, a global portfolio optimization, trading, and risk platform ($112B in volume). Current research focuses on DeFi risk infrastructure.

Data and Code. Exploit dataset, categorization methodology, and coefficient estimation code available at https://github.com/eggbertgjk/vault-risk-framework.