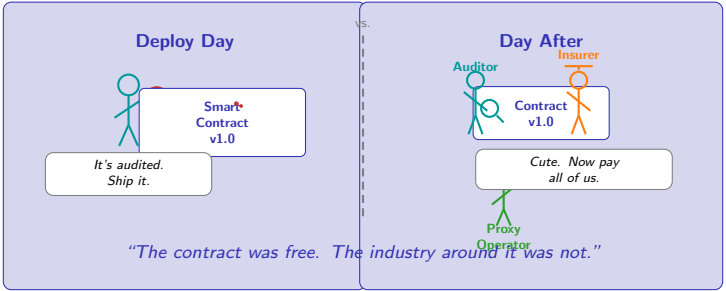
Smart Contract Business Models

Code that cannot be changed needs an industry to make sure it does not have to be

Digital Finance


Deploy Day
Smart Contract v1.0
It's audited. Ship it.
vs.
Day After
Auditor
Insurer
Contract v1.0
Cute. Now pay all of us.
Proxy Operator
"The contract was free. The industry around it was not."

If a Contract Cannot Be Changed, Whose Job Is It to Make Sure It Never Has To Be?

Unbundling = pulling one service out of a historical bundle and offering it alone; a *moat* = a competitive advantage rivals cannot easily copy, typically earned after the unbundled service gathers scale.

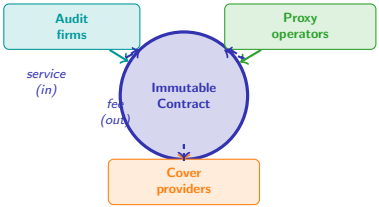
The Cost-of-Immutability Thesis

A contract that nobody can edit transfers an entire category of work *out of* the contract and *into* the surrounding industry. The pre-deployment review, the rescue plan, and the loss-recovery mechanism all become services that someone else has to sell, price, and take liability for.

The unbundling here is unusual: it is not a bank service being attacked from outside. It is a single function – "can we trust this code?" – being broken into three separate businesses that did not exist a decade ago. Each new business has its own customer, its own revenue model, and its own moat.

- **Auditors** sell pre-deployment assurance.
- **Proxy operators** sell post-deployment fixability.
- **Cover providers** sell post-loss compensation.

The protocol team pays all three – because the alternative is to absorb the risk on its own balance sheet, which most teams cannot do.



Three new revenue lines per deployed contract

Immutability does not eliminate the cost of correctness – it relocates that cost from the contract into a surrounding service industry.

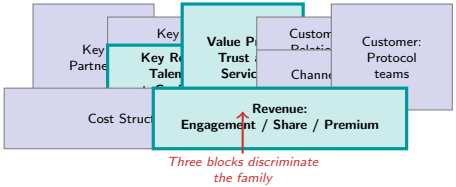
Which Three Canvas Blocks Define the Audit-and-Insure Business Model Family?

What is a Business Model Canvas (BMC)? Alexander Osterwalder's BMC is a one-page diagram of nine building blocks describing how any business creates, delivers, and captures value: Key Partners, Key Activities, Key Resources, Value Proposition, Customer Relationships, Channels, Customer Segments, Cost Structure, Revenue Streams. Reading across this family asks: which of these nine does each vendor own?

The Canvas Pattern
Reading the Business Model Canvas across audit firms, proxy operators, and cover providers reveals a shared logic. The three blocks that discriminate are not Customer Segments (always: protocol teams) – they are Value Proposition, Key Resources, and Revenue Streams.

- **Value Proposition** reduces to one of three claims: "we found the bug for you," "we let you fix the bug we missed," or "we pay you when the bug bites."
- **Key Resources** are scarce engineering talent, formal verification toolchains, and pooled capital – not customer data and not branch networks.
- **Revenue Streams** are upfront engagement fees, share of protocol revenue, or premium pools – never interchange and never float.

The canvas makes the family resemblance visible. All three are *trust-as-a-service* businesses, sold to the same customer, paid out of the same protocol budget, against the same underlying risk: a contract no one can patch.



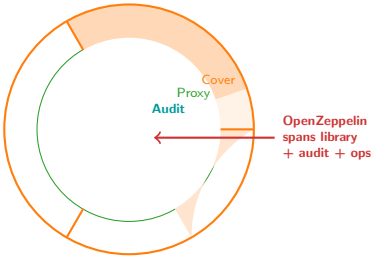
Across audit, proxy, and cover firms, only Value Proposition, Key Resources, and Revenue Streams differ – the rest of the canvas is shared.

How Does OpenZeppelin Sit at the Centre of the Audit-and-Insure Sunburst?

The OpenZeppelin Case
OpenZeppelin, the United-States-based smart-contract security firm founded in Argentina and now headquartered in San Francisco, began as a free library of audited Solidity components. Most teams writing a token, vault, or governance contract pulled in its implementations rather than rewriting them. That position – the default import statement – became the most valuable real estate in the entire audit-and-insure stack.

- **Open-source library** as the front door: free to use, distributed by package manager, embedded in nearly every serious protocol's codebase.
- **Paid audit practice** as the natural cross-sell: the authors of the components are also the most credible reviewers of contracts that depend on them.
- **Defender platform** as the post-deployment hook: monitoring, proxy administration, and incident response delivered as a recurring product.

The pattern is platform economics adapted to engineering: the developer side pulls in the library for free, the protocol-team side buys audits and operations services on top. Each free import increases the value of the paid audit – and vice versa.



The hub-and-orbit shape: free library at the centre, paid services in concentric rings – a sunburst of trust services around an immutable core.

How Does a Pure-Audit Firm Like Certora Rebundle Toward Continuous Verification?

Certora's Rebundling Arc

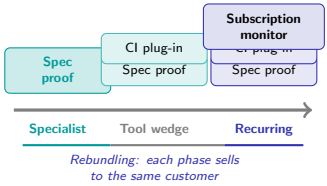
Certora, the formal-verification vendor founded in Israel and currently serving a global protocol-team customer base, started as a formal-verification specialist: a small team writing mathematical proofs that a given Solidity function obeys a given specification. The wedge product was deeply technical and sold to a narrow customer base – protocols willing to commission a specification file alongside the contract itself.

Phase one – specialist verification: a single audit-style engagement, delivered as a written report, with formal proofs as the deliverable. High price per engagement, low recurrence.

Phase two – developer-tool wedge: the same proof engine exposed as a CI plug-in, so protocol developers run lightweight checks on every commit, not only at major releases.

Phase three – continuous monitoring: subscription-priced integration that re-runs the proof set against every protocol change and flags drift – an audit that never ends.

The arc parallels classic FinTech rebundling: enter narrow, earn trust, then sell the next adjacent service to the same customer. The endpoint is a recurring relationship that looks very different from the one-off audit it started as.



Rebundling: a one-off audit engagement evolves into a developer tool, then a subscription monitor – recurring revenue from the same customer.

Where in the Pre-Deployment Value Chain Does Trail of Bits Choose to Insert Itself?

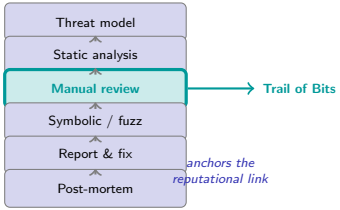
Value chain = the ordered sequence of activities a service passes through. Evans and Wurster (Boston Consulting Group) argued that when information is cheap, each link can be split off to a specialist — the chain *deconstructs* into independent layers.

The Pre-Deployment Value Chain

Trail of Bits, the United-States-headquartered security research and audit firm based in New York, anchors the heavy-end review link of this chain. The work between "a developer pushes code" and "the contract is live on mainnet" is itself a value chain. Each link is a separate service opportunity, and different vendors specialise in different links.

- **Threat modelling** – mapping what an attacker might want
- **Static analysis** – automated scans for known patterns
- **Manual review** – senior engineers reading every line
- **Symbolic / fuzz testing** – adversarial input generation
- **Public report & fix** – findings published, code patched
- **Post-mortem and lessons** – knowledge fed back to the team

Trail of Bits owns the heavy-end manual-review link, and from that position publishes its own open-source tools that shape every other link. Owning the most reputational link gives the firm an unbundling moat: clients return because they want the name on the report, not just the bug list. The pattern parallels how a top-tier law firm anchors a deal – the signature is part of the deliverable.



In the pre-deployment value chain, the manual-review link captures reputation rents – which is why specialist firms anchor there.

Is Sherlock's Audit Contest Model Regulatory Arbitrage on the Liability Itself?

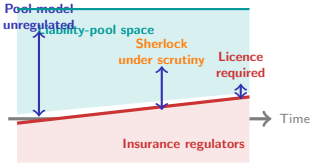
Regulatory arbitrage = a firm earns profit specifically because it faces a lighter rulebook than its competitors, not because it is better at the underlying business; the edge lasts only as long as the rulebook gap does.

The Liability-Arbitrage Tension
Sherlock, the audit-contest and staked-cover platform launched as an on-chain marketplace with an internationally distributed reviewer pool and a Cayman-domiciled sponsor entity, unbundles the audit and its liability backstop. Traditional audit firms carry professional indemnity insurance. Their liability is bounded by their cover and by negotiated limitation-of-liability clauses with each client. The cost of that insurance is a structural element of their pricing.

Sherlock relocates both pieces. Reviewers compete to find vulnerabilities in a fixed-window contest, and a separate pool of stakers underwrites the residual risk – if a covered exploit happens after audit, the staked capital pays out, not the auditors personally.

- **Marketplace mechanics** replace the partnership model: any qualified reviewer can compete, and the contest fee plus a payout pool replaces hourly billing.
- **Underwriting moves outside the firm**: cover capital is staked by independent participants, not held on the auditor's balance sheet.
- **Liability becomes a tradeable token-pool position**, unbundled from the engagement that produced it.

The arbitrage is real and useful – but inherently contestable. Regulators accustomed to professional-indemnity regimes can either recognise the staking pool as an insurance product or insist that the audit itself remains a regulated service. The gap is exactly the sort of "temporary advantage that needs to be converted into a licence" that defines the FinTech arbitrage tradition.



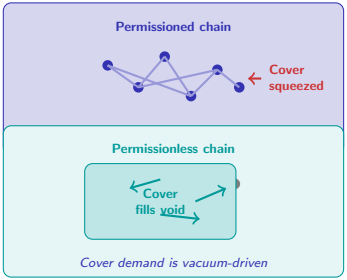
Audit-contest plus staked-cover models are arbitrage on liability classification – a temporary moat that closes as insurance regulators catch up.

Why Did Nexus Mutual Find Demand on Public Chains but Almost None on Permissioned Ones?

The Nexus Mutual Lesson

Nexus Mutual, the discretionary mutual registered in the United Kingdom and offering on-chain cover via a member-owned pool, illustrates how smart-contract cover sits in the public-chain vacuum. Smart-contract cover is a product whose demand depends entirely on whether the underlying contract is loss-bearing for users. On a public, permissionless chain, a depositor's funds are at risk from a code bug they cannot influence, cannot sue over, and cannot recover via insolvency proceedings. That is exactly the gap Nexus Mutual fills.

- Public chains generate cover demand because the user bears pure code risk – the protocol team is often pseudonymous and the contract carries no enforceable warranty.
- Permissioned chains generate almost none because a known operator stands behind the contract and absorbs loss through conventional channels.
- The mutual structure works because cover buyers and cover underwriters interact through a shared on-chain pool – network effects shrink the cost of cover as the pool grows.
- The lesson parallels infrastructure-vacuum FinTech: the cover business is not displacing an incumbent insurer; it is filling a void that conventional insurance refuses to step into.



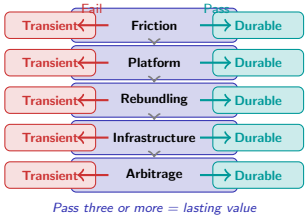
Smart-contract cover is a vacuum-driven product: it thrives where conventional insurance refuses to operate, and shrinks where operator liability already exists.

Which Five Tests Separate a Durable Audit-and-Insure Vendor from a Transient One?

The Five-Test Synthesis – Adapted

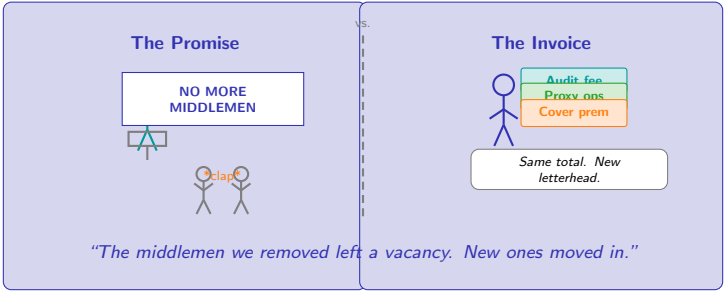
1. **Friction test:** Does the vendor remove a friction the protocol team cannot remove on its own – specialist verification, pooled cover capital, or hands-on incident response? Internal hires cannot replicate any of these cheaply.
2. **Platform test:** Does each new audited protocol or each new cover policy raise the value of the next one? Shared pool capital, shared incident playbooks, and shared component libraries all create cross-side network effects.
3. **Rebundling test:** Can the vendor sell an adjacent service to the same customer – monitoring on top of audit, proxy ops on top of monitoring, cover on top of ops? Single-service vendors are easy to disintermediate.
4. **Infrastructure test:** Is the vendor providing a service that nobody else can field, or is it competing in a crowded market for repeatable engagements? Vacuum-fillers face less price pressure.
5. **Arbitrage test:** Is the vendor's advantage based on a liability or licensing classification that regulators have not yet fixed? If so, can it be converted into a permanent licence before the gap closes?

Lasting value creation in the trust-as-a-service stack requires passing at least three of the five. Single-engagement audit shops typically pass only the first.



Pass three or more = lasting value

Lasting value in the trust-as-a-service stack requires passing at least three tests. Single-engagement audit shops typically pass only the first.


The Promise
NO MORE MIDDLEMEN
clap
vs.
The Invoice
Audit fee
Proxy ops
Cover prem
Same total. New letterhead.
"The middlemen we removed left a vacancy. New ones moved in."