Pre-Class Discovery Handout: Smart Contract Business Models

Activity 1: Trust-as-a-Service Canvas Detective

Scenario: Pick ONE of the following audit-and-insure vendors — Nexus Mutual, OpenZeppelin, Trail of Bits, Sherlock, or Certora — and investigate how that company actually makes money. Read their public website, their documentation, and any independent commentary you can find. Then fill in the canvas below. Focus on the structural mechanics of value creation, not on marketing language.

Canvas Element	Your Analysis
Value Proposition *Which trust claim does this vendor sell?*	
Customer Segments *Who actually pays the invoice?*	
Channels *How do customers find and buy the service?*	
Revenue Streams *Engagement, share, premium, or subscription?*	
Key Resources *Talent? Capital pool? Reputation? A library?*	

Q1: What is the single most important friction this vendor removes that a protocol team cannot remove on its own?

Q2: How does the vendor reach new protocol-team customers without a traditional sales force?

Q3: If this vendor disappeared tomorrow, what would protocol teams have to insource that they currently outsource?

Activity 2: Audit-and-Insure Mapping

Scenario: The cost-of-immutability paradox splits the work that "a contract is correct" used to mean into separate businesses. Match each vendor below to the part of the audit-and-insure stack it primarily occupies, then answer the questions.

Vendor		Stack Layer
OpenZeppelin	____	Component library + audit + post-deployment ops
Trail of Bits	____	Manual review and reputational anchor
Certora	____	Formal verification, sold one-off then as a tool
Sherlock	____	Audit contests with staked cover backing them
Nexus Mutual	____	Mutual cover for users of public-chain protocols

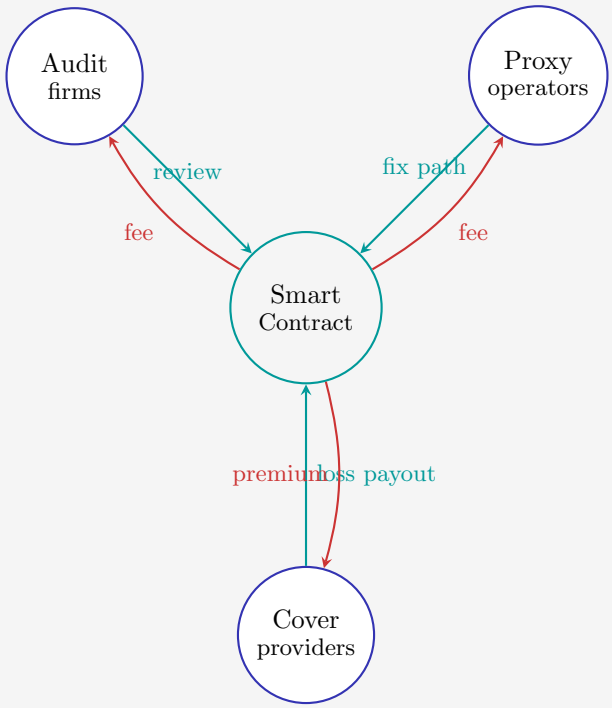
Q1: For each pair, describe in one sentence which trust claim the vendor sells and to whom.

Q2: Which of these vendors has begun adding products beyond its original wedge service? What did it add, and to which existing customers?

Q3: Why might a vendor that starts in one layer eventually want to sell adjacent services in the next layer up or down the stack?

Activity 3: The Three-Orbit Service Market

Scenario: An immutable smart contract sits at the centre. Around it three service markets orbit — each is a separate two-sided market, each is paid out of the same protocol budget, and each shrinks the perceived risk of the contract for someone different.



Q1: Why does each orbit look like a two-sided market in its own right? Identify the two sides for each orbit.

Q2: The "cold-start problem" for the cover orbit: which side is harder to attract first — cover buyers or cover underwriters — and why?

Q3: Once an orbit reaches critical mass (say, a deep cover pool), why is it hard for a competing pool to enter the same market?

Solutions

Activity 1: Trust-as-a-Service Canvas Detective

A1: **Model answer for OpenZeppelin:** The most important friction removed is the engineering cost of writing and re-auditing the same security-critical components – token standards, access control, upgradeability primitives – inside every protocol. By providing a continuously maintained, widely reviewed library, OpenZeppelin externalises a cost that every protocol team would otherwise carry alone, and replaces ad-hoc reinvention with a default import.

A2: OpenZeppelin reaches new protocol teams primarily through its open-source library: developers discover it via package managers and tutorials, embed it in production code, and only later become buyers of paid audits or post-deployment operations services. The free tier is the marketing channel; the paid tier is the cross-sell.

A3: Without OpenZeppelin, protocol teams would have to reimplement standard components, commission individual audits of each reimplementation, and run their own post-deployment monitoring infrastructure. The internal team would need to grow several specialist engineers and acquire incident-response tooling.

Canvas elements (OpenZeppelin):

- **Value Proposition:** Default trust components plus expert audit and post-deployment operations.
- **Customer Segments:** Primary – protocol engineering teams; secondary – enterprises building on permissioned ledgers using the same components.
- **Channels:** Open-source repositories, package-manager downloads, conference workshops, direct sales for the audit and operations products.
- **Revenue Streams:** Audit engagements, subscription tier on the operations platform, services revenue from custom engagements.
- **Key Resources:** Specialist engineering talent, the library codebase and its review history, brand recognition as the default component vendor.

Activity 2: Audit-and-Insure Mapping

A1: OpenZeppelin sells default trust components plus audit and operations services to protocol engineering teams. Trail of Bits sells reputational manual-review assurance to teams whose contracts will be examined by sophisticated counterparties. Certora sells formal-verification proofs of specification compliance, originally as one-off engagements and now as a continuous integration product. Sherlock sells competitive-discovery audits backstopped by an external staked cover pool. Nexus Mutual sells smart-contract cover to users of public-chain protocols, with cover capital provided by independent participants.

A2: Certora moved from one-off proof engagements to a CI plug-in to a subscription monitor – rebundling adjacent services to the same protocol customer. OpenZeppelin moved from a free library to paid audits to a paid post-deployment operations product. Both arcs grow customer lifetime value by selling a sequence of related services to a customer who has already adopted the wedge product.

A3: A single-layer vendor faces high acquisition cost per customer and a concentrated revenue stream. Once a protocol team trusts the vendor at one layer, the marginal cost of adding a service in an adjacent layer is much lower than the marginal cost of acquiring a new customer for that same adjacent service. Rebundling raises lifetime value and creates switching costs because the customer integrates more deeply with the vendor's tooling.

Activity 3: The Three-Orbit Service Market

A1: Each orbit is two-sided. **Audit:** reviewers (talent supply) and protocol teams (demand). **Proxy operations:** operators offering a service tier and protocol teams paying for it, with end users in the background as the people whose interests the proxy admin is trusted with. **Cover:** cover buyers (users seeking protection) and cover underwriters (capital providers). In each case the platform vendor sits between the two sides, taking fees from one or both.

A2: For the cover orbit, the harder side to attract first is usually the underwriters, because cover capital cannot be staked into an empty pool with no premium income. The vendor typically subsidises early underwriters or seeds the pool with its own capital, then attracts buyers once a credible payout reserve exists. This is the classic chicken-and-egg problem of two-sided markets, applied to a pooled-risk product.

A3: A deep cover pool enjoys cross-side network effects: more underwriters mean larger limits per policy, which attracts more buyers, which generates more premium income, which attracts more underwriters. A new pool faces a thin reserve that nobody trusts to pay out, which keeps both sides away. The incumbent's pool depth is itself the moat – a structural advantage that compounds with every additional participant.

