# Sybil- and Poisoning-Robust Budget-Feasible Mechanisms for Federated Learning with Auditable Compatibility

Liz Lemma Future Detective

January 14, 2026

**Abstract**

Federated learning (FL) procurement mechanisms increasingly operate in open, multi-tenant environments where strategic workers can create sybil identities or poison model updates. Existing budget-feasible truthful mechanisms for FL (e.g., CARE) assume exogenous worker quality and benign behavior, leaving them fragile under modern threats. We propose a clean, tractable mechanism design framework that layers auditability and refundable deposits onto compatibility-aware, budget-feasible procurement. Starting from any DSIC, budget-feasible base mechanism under compatibility constraints, we add randomized audits with imperfect detection and a deposit-forfeiture rule. We characterize closed-form deposit thresholds that simultaneously (i) preserve dominant-strategy truthfulness for cost reporting (deposits enter utilities as bid-independent constants) and (ii) deter both poisoning and sybil attacks by making the expected penalty exceed the attacker's benefit per identity. We provide welfare and accuracy guarantees that degrade smoothly with audit cost and false-negative risk, and we outline an empirical validation plan under secure aggregation constraints using standard poisoning benchmarks. The results modernize compatibility-aware FL procurement toward deployable 2026-era marketplaces where identities are cheap and adversaries are present.

## Table of Contents

3. 3. Model: agents, budgets, compatibility constraints, base mechanism M_0; malicious benefit and detection; audit technology; limited liability and deposit primitives.

4. 4. Mechanism: Robust-CARE wrapper (deposit requirement, audit lottery, forfeiture rule, audit financing); how it composes with any CARE-like allocation/payment rule.

5. 5. Theory I (incentives): DSIC preservation for cost reporting; deterrence conditions for poisoning; sybil-proofness conditions under per-identity deposits.

6. 6. Theory II (efficiency and budgets): welfare/quality approximation relative to M_0; explicit additive degradation from audit cost and false negatives; budget-feasibility conditions including audit spend.

7. 7. Design choices and comparative statics: optimal audit probability vs deposit size; effect of detection tech (attestations) on required deposits; discussion of when numerical optimization is needed.

8. 8. Empirical plan: poisoning and sybil simulations under compatibility constraints; metrics (accuracy, reputation/value, spend); secure aggregation/noisy signals and audit proxies.

9. 9. Discussion and extensions: heterogeneous deposits, endogenous audit targeting, repeated interactions, partial verification; limitations.

10. 10. Conclusion: deployable recipe for hardening budget-feasible FL procurement against modern threats.

# 1 1. Introduction: why DSIC+budget feasibility is insufficient in open FL markets; threat model (sybil + poisoning) and compatibility constraints; overview of deposit-audit design.

Federated learning (FL) procurement markets are increasingly organized as open, multi-requester environments: many requesters $j \in A$ simultaneously seek updates from a pool of worker identities $i \in S$, each identity submits a bid $b_i$ for participation, and a platform allocates identities to requesters subject to hard feasibility constraints and requester budgets $B_j$. A natural starting point is the budget-feasible procurement literature: we want a mechanism $M_0 = (X, P)$ that is dominant-strategy incentive compatible (DSIC) for private costs $c_i$, ex post individually rational (IR), and respects budgets while producing a near-optimal assignment for a quality objective such as $\sum_{i,j} x_{ij} v_i$. In a closed market with trustworthy participants, these desiderata largely capture what one needs: truthful bidding, feasible payments, and good performance.

In open FL markets, however, DSIC and budget feasibility are not enough. The reason is conceptual: DSIC speaks to truthful revelation of *costs* by a fixed set of agents, but it does not discipline *post-selection behavior* nor *identity formation*. FL admits an additional strategic margin that is absent from standard procurement: after selection, a worker can choose whether to behave honestly (contribute a benign update) or maliciously (poison the training process), and an adversary can cheaply create multiple identities (sybils) to increase selection probability or to circumvent participation limits. Thus, even if $M_0$ elicits truthful costs and makes payments within budgets, it can still select workers who have strong incentives to attack, and the resulting allocation can be arbitrarily harmful to requester utility.

The threat model we have in mind is deliberately simple but economically pointed. Each selected identity $i$ may obtain a one-shot gross gain $g_i$ from being malicious and *not* being detected—this $g_i$ can be interpreted as extortion value, competitive sabotage benefit, or the value of degrading a rival's model. The requester suffers a corresponding loss in value (captured abstractly by a damage parameter $D_i$), but the key is that the adversary's private benefit $g_i$ need not be correlated with the reported participation cost $b_i$ or the public proxy $v_i$. This breaks the usual alignment between selection rules (which reward low bids and/or high quality) and social welfare. Without additional instruments, a poisoning-capable identity can bid aggressively to get selected, receive payment $p_i$, and then impose harm while retaining the upside $g_i$. In short: DSIC can ensure "truthful costs," yet the platform still faces a moral hazard problem on "honest training."

Sybil behavior amplifies this concern. Standard single-parameter mechanism design typically treats each bidder as a distinct agent. In practice,

one real-world actor can control many accounts, each behaving as a separate $i \in S$. If selection is competitive and payments are positive, multiplying identities increases the attacker's chances of receiving at least one allocation—and, with it, an opportunity to poison. Moreover, sybils interact sharply with feasibility constraints that were designed for benign heterogeneity. In many real deployments, requesters impose "compatibility" or "incompatibility" constraints reflecting jurisdictional rules, provenance, device type, or communication channel diversity. We formalize this via groups $G_\ell$ and caps $\tau_{\ell j}$, requiring $|S_j \cap G_\ell| \leq \tau_{\ell j}$ for each requester $j$. These constraints are motivated by practice (e.g., "no more than $k$ devices from the same manufacturer/channel/jurisdiction"), but they also create an adversarial surface: if group membership can be manipulated (or if the attacker can simply populate many groups), sybils can be used to saturate caps across multiple $G_\ell$, reducing diversity, increasing correlated risk, and crowding out honest participants.

These observations motivate our central design question: how do we preserve the attractive economic properties of a CARE-style base mechanism—DSIC for costs, budget feasibility, and approximation guarantees under compatibility constraints—while adding credible deterrence against poisoning and sybil proliferation? Our answer is to augment $M_0$ with two levers that are natural in digital markets and increasingly feasible in FL platforms: refundable deposits and probabilistic audits.

The economic logic is straightforward. Payments $p_{ij}$ reward participation, but they cannot be made contingent on "no poisoning" if poisoning is hard to observe under secure aggregation or privacy constraints. Audits introduce a probabilistic signal: with probability $q$, a selected identity is audited, and conditional on audit a malicious action is detected with probability $\delta$. Deposits $d_i$ provide a transferable stake that can be forfeited upon detection. Together, these instruments convert an otherwise non-contractible behavior (honest training) into an incentive-compatible choice under risk of forfeiture. In expectation, a selected identity that attacks obtains

$$p_i - c_i - \kappa(1) + (1 - q\delta)g_i - q\delta\, d_i,$$

where $\kappa(1)$ is any direct cost of mounting the attack. If the worker behaves honestly, the deposit is refunded and expected utility is simply $p_i - c_i$. Thus, we can deter malicious behavior by choosing deposits large enough that the expected penalty $q\delta\, d_i$ dominates the undetected gain $(1 - q\delta)g_i$, uniformly over an adversary class with $g_i \leq \bar{g}$. This shifts the strategic problem away from unverifiable "intent" and toward verifiable financial commitments.

Crucially, the deposit-audit layer can be designed so that it does *not* unravel the DSIC and budget-feasible properties of the base mechanism for honest workers. Intuitively, if honest identities expect their deposits to be refunded, then deposits do not change their marginal incentives to misreport costs in the allocation stage—they only impose a liquidity requirement.

Audits similarly do not distort cost revelation if they are applied after selection and do not alter payments for honest behavior. This separation of roles is important: we use $M_0$ to solve the classical procurement problem (who should be hired, given bids, budgets, and compatibility constraints), and we use deposits/audits to solve the security problem (how to make "honest training" the best post-selection action). The two layers interact only through feasibility: audits incur resource costs $C_a$, and deposits impose participation constraints that may shrink the effective supply of eligible workers.

The interaction with budgets is where the mechanism design becomes nontrivial and policy-relevant. Audits are not free: whether the platform pays $C_a$ directly or charges requesters an audit fee $\phi_j$, audit intensity $q$ must be financed within the same economic ecosystem as payments. High $q$ improves deterrence but consumes resources that could otherwise fund more workers; high deposits $d$ improve deterrence but may exclude liquidity-constrained honest workers, potentially reducing competition and lowering welfare. Compatibility constraints further tighten the feasible set: when each requester $j$ can hire at most $\tau_{\ell j}$ from each group $G_\ell$, losing even a small fraction of eligible workers due to deposit requirements can have outsized effects on allocation quality. Our goal is therefore not to claim "audits solve poisoning," but to formalize the tradeoff between (i) incentive alignment against attacks, (ii) budget feasibility inclusive of audit expenditures, and (iii) the approximation performance inherited from $M_0$.

We emphasize two practical interpretations. First, deposits need not be viewed as punitive; they are a collateral mechanism akin to security deposits in rental markets or performance bonds in procurement. In digital labor settings, similar instruments appear as escrow, staking, or bonded participation. Second, audits need not be invasive: in FL they may take the form of anomaly checks, attestation of secure enclaves, spot-check tasks, or statistical tests on updates. Our model compresses these operational details into an effective detection probability $\delta$, explicitly acknowledging that privacy-preserving aggregation and partial observability may limit $\delta$ below one.

Finally, we are explicit about what this framework does *not* attempt to do. We do not model the full dynamics of repeated interaction, reputation, or adaptive adversaries; we focus on a one-shot procurement round where deterrence must come from immediate incentives. We also do not assume perfect attribution of harm: audits are probabilistic and may produce false negatives, summarized by $1 - q\delta$. The point of the model is to illuminate a design space in which a platform can make a transparent commitment—choose $(q, \delta)$ through auditing technology and set deposits $d_i$ (possibly uniform)—that simultaneously respects requester budgets, retains DSIC cost revelation for honest participants, and makes sybil-mediated poisoning economically unattractive for bounded adversaries. This provides a clean bridge between classical budget-feasible mechanism design and the operational security concerns that dominate real-world FL deployments.

## 2 Related work.

Our setting sits at the intersection of three literatures that are typically studied in isolation: budget-feasible procurement mechanisms (and, more recently, compatibility-aware FL procurement), adversarial/robust federated learning (poisoning and sybils), and audit/collateral instruments used to enforce behavior when outcomes are only imperfectly observable. We briefly review each strand and clarify how our approach borrows from, and departs from, existing solutions.

**Budget-feasible procurement and compatibility-aware mechanisms.** A large mechanism-design literature studies procurement with privately known seller costs under a hard buyer budget. The canonical model is a single buyer with a monotone value function over subsets of sellers and a budget constraint; the goal is a truthful (often DSIC) mechanism that is individually rational and approximately optimal while never exceeding the budget. A central technical challenge is that the budget constraint breaks the VCG template: paying marginal contributions can exceed the available funds, so one typically relies on posted-price, threshold-payment, or greedy-selection constructions coupled with careful payment rules to maintain truthfulness. For additive values, knapsack-like constraints, and matroid feasibility constraints, there is a now-standard toolkit of constant-factor approximation mechanisms under DSIC/IR/budget feasibility. Extensions consider multi-unit procurement, submodular values, and combinatorial constraints, often using reductions to "greedy + critical payment" or random sampling to estimate prices.

Federated learning procurement inherits these constraints but adds structure. First, the "buyer" side can be multi-requester: many requesters simultaneously procure from a shared pool of workers, so the allocation is closer to a bipartite assignment or matching problem than a single knapsack. Second, platforms often impose feasibility constraints that encode operational or policy requirements—jurisdictional limits, channel diversity, provenance constraints, device-type limits, or per-requester caps on correlated risk. These constraints are naturally modeled as group caps (e.g., $|S_j \cap G_\ell| \leq \tau_{\ell j}$), which resemble partition-matroid constraints layered on top of assignment feasibility. Compatibility-aware FL mechanisms in the CARE family can be viewed as importing budget-feasible procurement principles into this multi-requester, constrained-allocation environment: they explicitly track requester budgets, incorporate caps and diversity constraints into the allocation rule, and preserve incentive properties (truthfulness for costs) while delivering approximation guarantees for a quality proxy objective. Conceptually, this line of work answers the question: *if the only strategic issue is cost misreporting, how can we allocate FL participants across heterogeneous requesters without violating budgets or policy constraints?*

Our work takes this machinery as a starting point rather than an endpoint. The mechanisms above are designed for *selection* under private costs; they are not designed to discipline *post-selection behavior* (e.g., whether an assigned worker contributes an honest update), nor do they directly address identity proliferation. In this sense, our contribution is orthogonal to the approximation and incentive-compatibility advances in procurement: we preserve the procurement layer and ask what additional market instruments are needed when the selected action is not contractible.

**Robust and adversarial federated learning: poisoning and sybils.** The second strand concerns adversarial participants in distributed and federated training. A substantial algorithmic literature studies robustness to Byzantine or malicious updates, proposing aggregation rules (coordinate-wise median, trimmed mean, geometric median, Krum-like selection, norm clipping) and anomaly detection heuristics intended to limit the influence of outliers. Parallel work studies targeted poisoning and backdoor attacks that can evade simple anomaly filters, especially in heterogeneous data regimes where benign updates are naturally diverse. There is also a growing body of work on sybil attacks in FL, where one attacker controls many clients to increase weight in the aggregation, to bypass participation rules, or to mimic heterogeneity. Defenses span identity verification, device attestation, reputation systems, and cross-round consistency checks.

From an economic perspective, these defenses are typically *technology-side* mitigations: they aim to reduce the damage conditional on participation, but they do not directly realign the attacker's incentives. Moreover, several widely deployed privacy and security primitives (notably secure aggregation) make fine-grained monitoring difficult: the platform may not observe individual updates or may observe them only through a privacy-preserving interface. This creates a familiar enforcement problem: even if the platform can sometimes detect poisoning (via statistical tests, holdout validation, attestation, or challenge tasks), detection is imperfect and costly, and therefore cannot be treated as a deterministic contractible outcome.

Our work is complementary to robust aggregation: rather than improving the aggregator, we model robustness as an *effective detection probability* and ask how to use economic penalties to deter attacks that slip past algorithmic defenses. This is important in practice because "robust FL" and "secure FL" can be in tension: stronger privacy can reduce observability and thereby weaken purely algorithmic enforcement. By abstracting these details into an audit technology with imperfect detection, we can analyze how much deterrence can be obtained from financial instruments even when the platform's signal is noisy.

**Audits, deposits, and collateral in digital markets.** A third literature studies how markets enforce desirable behavior when actions are hidden or quality is imperfectly observed. Classical procurement uses performance bonds, retainage, and warranties: suppliers post collateral or accept payment holdbacks that are forfeited under verified nonperformance. Labor and platform markets use escrow, staged payments, and dispute resolution. Financially, these tools share a common logic: they create "skin in the game" that can be seized conditional on an adverse signal, thereby converting a moral hazard problem into one with enforceable incentives.

Random audits are a particularly common enforcement device. In regulatory economics (e.g., tax compliance, environmental enforcement), the canonical inspection model trades off audit intensity against enforcement costs and uses expected penalties to deter violations. In mechanism design, random verification and "audit with probability $q$" appear in models of costly state verification and fraud deterrence. In cryptographic and blockchain systems, staking and slashing implement a similar deterrence mechanism: participants lock collateral that is automatically forfeited upon provable misbehavior; probabilistic monitoring and challenge games can reduce verification costs while maintaining incentives.

Our setting adapts these ideas to FL procurement, but with two domain-specific complications. First, enforcement must coexist with budget-feasible payments: the ecosystem funding audits is the same ecosystem paying workers, so audit expenditures are not an exogenous add-on. Second, the relevant strategic actor may be a sybil controller: collateral must deter not just a single identity's deviation but also profitable identity splitting, where an attacker creates many accounts to increase selection probability. Collateral and audits can address this by making the expected gain from any single selected identity negative, which in turn eliminates positive expected profit from creating additional identities (absent cross-identity complementarities). This "per-identity deterrence" viewpoint echoes how staking systems scale to many pseudonymous participants.

**Positioning and limitations.** Relative to procurement mechanisms, we keep the selection/payment layer intact and add an enforcement layer. Relative to robust FL, we do not propose a new aggregation defense; instead we treat detection as imperfect and ask what economic commitments can compensate for limited observability. Relative to audit/collateral models, we highlight the multi-requester, compatibility-constrained nature of the market and the fact that enforcement must be financed without breaking budget feasibility.

At the same time, our approach inherits limitations common to collateral-based enforcement. Deposits can exclude liquidity-constrained but honest workers, which may reduce competition and degrade allocation quality; au-

diting can be expensive and may be politically or operationally constrained; and imperfect detection implies residual risk when audits are rare or weak. These limitations motivate the formal model that follows: we make the enforcement technology explicit, track how it interacts with budgets and feasibility constraints, and state guarantees in terms of transparent parameters (audit intensity, detection effectiveness, and collateral requirements) rather than assuming away the underlying frictions.

# 3 Model and primitives.

We study a procurement layer for federated learning (FL) that is *already* well understood—private worker costs, requester budgets, and compatibility constraints—and then add a minimal enforcement layer that captures post-selection poisoning incentives under imperfect observability. Our goal in this section is to state the ingredients in a way that makes the subsequent "wrapper" construction modular: any CARE-like budget-feasible assignment mechanism can be treated as a black box $M_0$, and we will reason about what must be added so that selection incentives and post-selection incentives are jointly well behaved.

**Agents, tasks, and feasibility.** There is a finite set of requesters $A$, indexed by $j$, with $|A| = m$, and a finite set of worker *identities* (accounts) $S$, indexed by $i$, with $|S| = n$. We emphasize "identities" because multiple identities may be controlled by a single real-world actor; the platform cannot perfectly link them. Each requester $j$ posts a monetary budget $B_j$ used to pay selected workers (and, depending on the audit-financing rule, possibly to fund audits). Each worker identity $i$ can be assigned to at most one requester. Let

$$x_{ij} \in \{0,1\} \quad \text{and} \quad x_i := \sum_{j \in A} x_{ij} \in \{0,1\}$$

denote the allocation indicators.

Operational and policy constraints enter through *incompatibility groups* $G_\ell \subseteq S$ for $\ell \in \{1, \dots, L\}$. These groups can represent correlated-risk classes (same ISP or device vendor), provenance classes, jurisdictions, or any attribute for which a requester/platform wants a cap on concentration. Each requester $j$ is subject to caps

$$|S_j \cap G_\ell| \;\le\; \tau_{\ell j} \qquad \forall \ell,$$

where $S_j = \{i \in S : x_{ij} = 1\}$ is the set assigned to $j$. We treat group membership as observable and verifiable at the level needed to enforce the cap (e.g., via compliance attestation or KYC-like checks), but we do *not* assume the platform can link multiple identities belonging to the same underlying actor.

**Costs, bids, and the base procurement mechanism $M_0$.** Worker identity $i$ has a private participation cost $c_i \geq 0$, capturing local compute, communication, and any opportunity cost of contributing an update. The worker reports a bid $b_i$ strategically. We also allow the platform to use an exogenous, publicly observed quality or reputation proxy $v_i \geq 0$ (e.g., historical reliability, device class, or dataset richness), which enters the platform's selection objective but is not itself a strategic report in the baseline model.

A *base mechanism* $M_0 = (X, P)$ maps the bid profile $b = (b_1, \ldots, b_n)$ into an allocation $X(b) = \{x_{ij}(b)\}$ and payments $P(b) = \{p_{ij}(b)\}$. Let $p_i(b) := \sum_j p_{ij}(b)$ denote the total payment to identity $i$. We take as given that $M_0$ (i) respects the feasibility constraints above (assignment plus group caps), (ii) is dominant-strategy incentive compatible (DSIC) for cost reporting and ex post individually rational (IR) for honest participants, (iii) is budget feasible in the sense that requester payments do not exceed budgets, and (iv) attains a $\beta$-approximation to a canonical additive objective $\sum_{i,j} x_{ij} v_i$ subject to the feasibility constraints and budgets. Concretely, one can think of CARE-like allocation rules that implement a greedy selection under per-requester budget constraints with threshold (critical) payments, but we keep $M_0$ abstract to emphasize composability.

Two modeling remarks clarify what $M_0$ does *not* do. First, $M_0$ addresses selection under private costs; it does not contract on the content of model updates. Second, $M_0$ treats identities as atomic: if an adversary creates many identities, $M_0$ may select multiple of them unless compatibility caps happen to bind.

**Post-selection behavior: honest contribution vs. poisoning.** After selection, each winning identity $i \in \mathcal{W}$ (where $\mathcal{W} = \{i : x_i = 1\}$) chooses an action $m_i \in \{0, 1\}$, with $m_i = 0$ interpreted as "honest contribution" and $m_i = 1$ as "malicious/poisoning behavior." We deliberately model maliciousness in reduced form, since the space of attacks is large (label-flipping, backdoors, targeted poisoning, gradient manipulation, etc.) and observability depends on the training pipeline.

If identity $i$ behaves maliciously and is *not* detected, it obtains a one-shot gross benefit $g_i \geq 0$. This benefit can represent extortion value, sabotage value, or a competitor's gain from degrading a model. If detected, the identity may be penalized via loss of a posted deposit (introduced below). We allow an additional direct resource cost $\kappa(1) \geq 0$ of mounting an attack, with $\kappa(0) = 0$; in the cleanest case $\kappa(1) = 0$, so deterrence must come from expected penalties.

On the requester side, an undetected malicious identity may impose a loss $D_i \geq 0$ in realized value. We do not require the platform to observe $D_i$ or to price it directly; we introduce it mainly as an accounting device for welfare and approximation-loss bounds later. When $m_i = 0$, we assume no

negative externality beyond the worker's cost $c_i$, i.e., honest participation is socially beneficial in the sense intended by the base objective.

**Audit technology and imperfect detection.** The platform can audit selected identities using a randomized procedure: each winner $i$ is audited with probability $q \in (0, 1]$, committed ex ante. Conditional on auditing a malicious identity, the audit detects misbehavior with probability $\delta \in (0, 1]$. Thus the overall detection probability against a malicious winner is

$$p_{\text{det}} = q\delta.$$

This reduced-form detection probability is meant to subsume both statistical detection (e.g., anomaly checks, holdout validation) and systems signals (e.g., secure enclaves, attestation, challenge tasks). Importantly for FL practice, privacy-preserving protocols such as secure aggregation can reduce per-client observability; in our model this simply corresponds to a lower effective $\delta$. We treat audits across identities as independent conditional on selection, which is the natural benchmark when the platform uses independent sampling; correlated audit schemes could be incorporated but are not needed for the core deterrence logic.

Auditing is costly: each audit incurs cost $C_a \geq 0$ to the platform (or equivalently to the ecosystem if charged back to requesters). We do not yet specify how audit costs are allocated across requesters—that is part of the mechanism wrapper—but we include an audit fee term $\phi_j$ in requester utility to keep track of the fact that enforcement must ultimately be financed.

**Deposits, limited liability, and penalties.** Because malicious behavior is not perfectly observable, we introduce a standard enforcement primitive: refundable deposits (collateral). Each identity $i$ can be required to post a deposit $d_i \geq 0$ upon participation/selection (depending on implementation). If the identity is audited and detected malicious, it forfeits $d_i$; otherwise the deposit is refunded. This captures limited liability in a way that matches many digital-market settings: the platform can credibly escrow and slash a posted amount, but cannot impose unbounded ex post fines on pseudonymous accounts. In particular, the maximum penalty that can be enforced per identity is $d_i$, so deterrence must come from making the *expected* loss $q\delta d_i$ outweigh the expected gain from undetected misbehavior.

Let $p_i$ denote the payment the identity receives from $M_0$ if selected. The worker's ex post utility can be written as

$$u_i(b_i, m_i) = p_i(X(b)) - c_i x_i - \kappa(m_i) x_i - \mathbf{1}\{x_i = 1\}\mathbf{1}\{\text{audit}\}\mathbf{1}\{m_i = 1\}\mathbf{1}\{\text{detect}\}\, d_i + \mathbf{1}\{x_i = 1\}\mathbf{1}\{m_i = 1$$

Taking expectations conditional on selection ($x_i = 1$), if the identity is malicious ($m_i = 1$),

$$\mathbb{E}[u_i \mid x_i = 1, m_i = 1] = p_i - c_i - \kappa(1) + (1 - q\delta)g_i - q\delta d_i,$$

whereas if it is honest ($m_i = 0$),

$$\mathbb{E}[u_i \mid x_i = 1, m_i = 0] = p_i - c_i,$$

since the deposit is refunded in expectation (and deterministically if audits only trigger forfeiture upon detection).

**Adversarial capability and bounds.** A key design input is an upper bound $\bar{g}$ on the one-shot malicious benefit among the adversarial identities we seek to deter. We do not assume the platform observes $g_i$ or can distinguish adversarial from honest identities; rather, $\bar{g}$ is a policy/engineering parameter reflecting the threat model (e.g., the maximum plausible external gain from a single round of poisoning). This bound is what makes a uniform deposit requirement meaningful: if $g_i \le \bar{g}$, then one can set deposits to make deviation unprofitable. Of course, if some attackers have $g_i \gg \bar{g}$, then purely financial deterrence may be insufficient without stronger detection ($\delta$) or higher audit rates ($q$); we return to this limitation when interpreting guarantees.

Finally, requester $j$'s realized utility is

$$U_j = V_j(S_j) - \sum_i p_{ij} - \phi_j,$$

where $V_j(\cdot)$ is the requester's value from its assigned set, and $\phi_j$ accounts for audit financing. We do not assume $V_j$ is directly observed by the platform; the mechanism's approximation properties will be stated relative to the additive proxy $\sum_{i \in S_j} v_i$, while $D_i$ and $\phi_j$ allow us to track how enforcement affects actual welfare.

This completes the model. The next section specifies a wrapper around $M_0$ that (i) preserves DSIC/IR for cost reporting by honest workers, (ii) makes $m_i = 1$ strictly dominated (in expectation) for all identities with $g_i \le \bar{g}$, including when an attacker can create many identities, and (iii) accounts explicitly for audit expenditures within requester budgets.

# 4 Mechanism: Robust-CARE wrapper.

We now describe a simple enforcement "wrapper" that can be layered on top of *any* CARE-like budget-feasible assignment rule $M_0 = (X, P)$. The wrapper has two design goals. First, it should be *modular*: we want to reuse the allocation and threshold-payment logic of $M_0$ without re-deriving incentives from scratch. Second, it should be *operational*: the platform should be able to implement it with standard marketplace primitives (escrow, randomized inspections, and deterministic billing rules) while keeping requester budgets and compatibility constraints explicit.

**High-level structure.** Fix public parameters $(q, \delta)$ for auditing and detection, an audit cost $C_a$, and a deposit schedule $\{d_i\}_{i \in S}$ (often uniform $d_i \equiv d$). The Robust-CARE mechanism, denoted

$$M^{\mathrm{R}} \;=\; \mathrm{Wrap}(M_0;\, q, \delta, \{d_i\}, C_a, \mathrm{financing}),$$

takes as input requester budgets $\{B_j\}$, worker bids $b$, and the feasibility system (assignment and group caps), and outputs (i) an allocation $X$, (ii) base payments $P$ as in $M_0$, (iii) an audit plan, and (iv) deposit refunds/forfeitures realized after auditing.

Conceptually, we keep $M_0$ responsible for *who is selected* and *how much they are paid for participation*, and we use deposits and audits to control *what a selected identity prefers to do after selection*. The wrapper does not require observing the full FL update; all post-selection enforcement is mediated by the reduced-form audit technology summarized by $\delta$.

**Step 0: Commitment and eligibility.** Before bids are submitted, the platform commits publicly to: (i) the base mechanism $M_0$; (ii) the audit sampling rule (encoded by $q$); (iii) the detection process conditional on audit (captured by $\delta$); (iv) the deposit schedule $d_i$ and the forfeiture rule; and (v) a deterministic audit-financing rule ensuring budgets cover both worker payments and audit expenditures.

Workers participate by submitting bids $b_i$ *and* accepting the deposit requirement. Operationally, this can be implemented either by (a) pre-escrowing $d_i$ at bid time, or (b) pre-authorizing a payment instrument and escrowing immediately upon provisional selection. We write the mechanism as if escrow is effective whenever $x_i = 1$; if an identity cannot post the deposit, it is treated as ineligible.[1]

A key modularity choice is that $d_i$ is set as a function of public features (or uniformly), and *not* as a function of the worker's bid. This separation is what allows us to preserve the cost-reporting incentives inherited from $M_0$.

**Step 1: Audit-budget reservation and effective budgets.** Audits must be financed. To keep budget feasibility transparent, we reserve audit funds *before* invoking $M_0$ by shrinking requester budgets. Let $K_j$ be a known upper bound on the number of identities that requester $j$ could ever be assigned (e.g., a posted "client count" target, or a systems cap implied by training). Define a deterministic audit reserve

$$R_j \;:=\; C_a \cdot \lceil qK_j \rceil\,, \qquad \widetilde{B}_j \;:=\; \max\{B_j - R_j,\, 0\}.$$

[1] In practice, to avoid rerunning allocation when a winner fails to post collateral, platforms typically require pre-escrow or pre-authorization. Our theoretical arguments later only use that posting is required whenever an identity becomes a winner.

The interpretation is: requester $j$ sets aside enough funds to pay for up to $\lceil qK_j \rceil$ audits at cost $C_a$ each; the remaining $\widetilde{B}_j$ is the budget available for worker payments in the base mechanism. This rule is conservative (it guarantees that the audit reserve suffices ex post under the audit plan described below), yet it is simple and does not depend on bids.

Other financing rules are possible (e.g., charging per realized audit ex post, or pooling audit costs platform-wide), but the reserve-based rule is convenient for composition: after the reserve is carved out, the remainder is a standard multi-requester budget-feasible procurement instance to which $M_0$ applies directly.

**Step 2: Run the base mechanism unchanged.** We run $M_0$ on the eligible worker set with budgets $\{\widetilde{B}_j\}$ and the original feasibility constraints (assignment plus group caps):

$$(X, P) \;=\; M_0(b;\, \{\widetilde{B}_j\}, \{\tau_{\ell j}\}, \{G_\ell\}, \{v_i\}).$$

This step is intentionally "black-box": the wrapper does not modify the allocation rule or payment formula beyond swapping $B_j$ for $\widetilde{B}_j$. In particular, if $M_0$ is implemented via greedy selection with critical (threshold) payments, we use exactly those critical payments here.

Let $S_j = \{i : x_{ij} = 1\}$ and $\mathcal{W} = \cup_j S_j$ denote the winners. For each winner $i \in \mathcal{W}$, the platform escrows $d_i$ (if not already escrowed). The base payment $p_i = \sum_j p_{ij}$ is determined at this point but may be disbursed later (see below).

**Step 3: Audit lottery (randomized inspections with deterministic cost).** The wrapper specifies an audit lottery that approximates per-identity audit probability $q$ while keeping realized audit expenditures within the reserved amount $R_j$ for each requester. For each requester $j$, define the number of audits to run as

$$a_j \;:=\; \min\big\{|S_j|,\; \lceil qK_j \rceil\big\}.$$

Then the platform draws a subset $A_j \subseteq S_j$ uniformly at random of size $a_j$ (sampling without replacement). Each $i \in A_j$ is audited; each $i \in S_j \setminus A_j$ is not audited. This construction ensures that the total audit cost for requester $j$ is exactly $C_a a_j \le C_a \lceil qK_j \rceil = R_j$, i.e., funded ex post by the reserved amount.

When $|S_j| \le K_j$, each winner under $j$ faces audit probability approximately $q$ (exactly $a_j/|S_j|$, with the usual floor/ceiling discretization). The analysis in the next section uses only the fact that there is a known lower bound on the audit probability for each selected identity; the without-replacement lottery is simply a convenient way to make costs deterministic.

**Step 4: Forfeiture/refund and payment execution.** After training contributions are delivered, the platform executes audits for $i \in A_j$. If $i$ is honest, nothing happens. If $i$ is malicious, detection occurs with probability $\delta$; upon detection, the deposit $d_i$ is forfeited (slashed), and upon non-detection it is refunded.

Operationally, to prevent "hit-and-run" identities from collecting $p_i$ and disappearing before a slash can be applied, the wrapper can disburse payments only after the audit window closes. Concretely:

- deposits $d_i$ are escrowed at selection time;
- base payments $p_i$ are authorized at selection time but settled after auditing;
- if $i$ is detected malicious, the platform withholds (some or all of) $p_i$ to the extent permitted by the payment rail, and in any case forfeits $d_i$.

Our theory does not rely on withholding $p_i$ (the deterrence comes from $d_i$), but delayed settlement is a standard practical complement.

Finally, the audit reserve is used to pay auditors: requester $j$ is charged $\phi_j = C_a a_j$ (or equivalently, the platform debits $R_j$ and refunds any unused portion $R_j - \phi_j$ if $a_j < \lceil qK_j \rceil$). This makes the requester's total spend equal to $\sum_i p_{ij} + \phi_j \le \widetilde{B}_j + R_j \le B_j$, guaranteeing budget feasibility ex post under the wrapper.

**Why this composes cleanly with CARE-like rules.** Three design choices deliver modularity. First, the allocation and base payments are computed entirely by $M_0$ using shrunken budgets $\widetilde{B}_j$; deposits and audits are layered *after* selection. Second, the deposit schedule $\{d_i\}$ and audit sampling are independent of bids, so they do not create additional bid-dependent transfers that would typically interfere with DSIC arguments for cost reporting. Third, audit costs are handled via a deterministic reserve and a deterministic (allocation-dependent) audit count $a_j$, so the wrapper does not create stochastic budget overruns that would otherwise force ad hoc rationing.

Put differently, we can view Robust-CARE as turning each requester's budget $B_j$ into two earmarked accounts: a procurement account $\widetilde{B}_j$ that feeds into $M_0$, and an enforcement account $R_j$ that finances randomized auditing. The next section shows that, with an appropriate deposit level, this separation ensures that (i) truthful bidding remains optimal for honest workers as in $M_0$, and (ii) post-selection poisoning becomes strictly unprofitable in expectation for identities whose one-shot attack gain is bounded—importantly, on a per-identity basis, which is what enables sybil-style robustness.

# 5 Theory I: Incentives and deterrence.

We now formalize why the wrapper preserves the *cost-reporting* guarantees of the base rule $M_0$, while adding a *post-selection* deterrent against poisoning. The key observation is that the wrapper intervenes only through (i) bid-independent collateral and (ii) bid-independent random audits. This separation lets us analyze the mechanism by backward induction: we first study the optimal post-selection action $m_i$ for a selected identity, and then we study bidding incentives given those continuation payoffs.

**A useful lower bound on audit risk.** Fix any realized winner set $S_j = \{i : x_{ij} = 1\}$ for requester $j$. Under the without-replacement lottery described earlier, each $i \in S_j$ is audited with probability $a_j/|S_j|$, where $a_j = \min\{|S_j|, \lceil qK_j \rceil\}$. In particular, whenever $|S_j| \le K_j$ we have $a_j = \lceil qK_j \rceil$ and hence $a_j/|S_j| \ge q$ (up to discretization). Our incentive conditions below only require a known *lower bound* on the per-winner audit probability; we therefore write the deterrence condition in terms of a design parameter $q$ that is interpreted as such a lower bound. The overall per-identity detection probability when malicious is then

$$p_{\text{det}} \;=\; q\delta.$$

## 5.1 DSIC preservation for cost reporting (honest identities).

We begin with the workers whose strategic choice is only their cost report $b_i$, i.e., they intend to act honestly after selection ($m_i = 0$). For these identities the wrapper adds escrow and (possibly) the inconvenience of delayed settlement, but in our quasilinear model it does not add bid-dependent transfers.

Formally, fix a worker $i$ with true cost $c_i$ who plans $m_i = 0$. Conditional on any bid profile $b$, the wrapper runs $M_0$ on the shrunken budgets $\{\widetilde{B}_j\}$, producing an allocation/payment pair $(X(b), P(b))$. If $x_i = 0$, the worker's utility is 0. If $x_i = 1$, the worker posts $d_i$ and later receives it back with certainty (because $m_i = 0$), so the expected utility is simply

$$\mathbb{E}[u_i \mid m_i = 0] \;=\; p_i(X(b)) - c_i x_i,$$

exactly as in the base mechanism. In particular, the deposit does not enter the payoff for honest behavior except through a liquidity channel that we do not model (we return to this limitation briefly below).

**Proposition 5.1 (DSIC for honest cost reporting).** Assume $M_0$ is DSIC and ex post IR for costs under budgets $\{\widetilde{B}_j\}$ and the given feasibility constraints (assignment plus group caps). Then in the wrapped mechanism $M^{\mathrm{R}}$, truthful bidding $b_i = c_i$ is a dominant strategy for any worker identity that plays $m_i = 0$, and ex post IR holds for such identities.

*Proof sketch.* Holding other bids fixed, the allocation rule $X(\cdot)$ and base payments $P(\cdot)$ faced by worker $i$ are exactly those of $M_0$ on $\{\widetilde{B}_j\}$. The wrapper adds only (i) escrow of $d_i$ when selected and (ii) a refund of the same amount when $m_i = 0$; both are independent of $b_i$. Hence worker $i$'s utility as a function of $b_i$ coincides with its base-mechanism utility. DSIC and IR therefore carry over directly. □

**Interpretation.** This proposition is the modularity payoff: we do not need to re-prove incentive compatibility of the procurement rule; we only need to ensure that the wrapper's additional transfers do not depend on bids. Put differently, the wrapper is designed to manipulate the *continuation game* (what a winner does after being selected), not the *selection game* itself.

## 5.2 Post-selection deterrence: when does poisoning become unprofitable?

We now analyze the winner's post-selection choice $m_i \in \{0, 1\}$. Conditional on being selected ($x_i = 1$), the expected utility from honest behavior is

$$\mathbb{E}[u_i \mid x_i = 1,\, m_i = 0] \;=\; p_i - c_i.$$

If instead the identity behaves maliciously, it obtains the one-shot benefit $g_i$ if undetected, and loses its deposit if audited and detected. Using $p_{\text{det}} = q\delta$, the expected utility is

$$\mathbb{E}[u_i \mid x_i = 1,\, m_i = 1] \;=\; p_i - c_i - \kappa(1) + (1 - p_{\text{det}})g_i - p_{\text{det}} d_i.$$

The difference between malicious and honest behavior (conditional on selection) is therefore

$$\Delta_i \;:=\; \mathbb{E}[u_i \mid x_i = 1, m_i = 1] - \mathbb{E}[u_i \mid x_i = 1, m_i = 0] \;=\; -\kappa(1) + (1 - p_{\text{det}})g_i - p_{\text{det}} d_i.$$

**Proposition 5.2 (Deposit deterrence condition).** Fix an identity $i$ and suppose it is selected. If the deposit satisfies

$$d_i \;\geq\; \frac{g_i}{q\delta},$$

then $m_i = 1$ is strictly worse than $m_i = 0$ whenever $g_i > 0$ and $q\delta > 0$; hence honest behavior is a strict best response in the post-selection stage.

*Proof.* Under the stated condition,

$$\Delta_i \;\leq\; -\kappa(1) + (1 - q\delta)g_i - q\delta \cdot \frac{g_i}{q\delta} \;=\; -\kappa(1) - q\delta\, g_i \;<\; 0,$$

where strictness follows from $q\delta\, g_i > 0$ (and/or $\kappa(1) > 0$). □

**Uniform deposits and bounded adversaries.** The platform typically cannot tailor $d_i$ to a private $g_i$. The natural design route is therefore to posit an adversary class with known bound $g_i \le \bar{g}$, and choose a uniform deposit $d$ that deters that class:

$$d \;\ge\; \frac{\bar{g}}{q\delta}.$$

This is conservative but transparent: if audits are rare ($q$ small) or weak ($\delta$ small), then deterrence requires more collateral.

**Equilibrium implication.** With such a deposit, the continuation game has a unique dominant action for winners—honesty. As a result, the wrapped mechanism effectively implements the base mechanism $M_0$ on $\{\widetilde{B}_j\}$ in equilibrium, but with an added enforcement layer that removes profitable poisoning deviations for the bounded adversary class.

### 5.3 Sybil-proofness from per-identity collateral.

We finally address why per-identity deposits speak directly to sybil-style attacks. The threat model is that a single real-world adversary may control many marketplace accounts, each treated as a distinct $i \in S$. Such an adversary can attempt to (i) increase the probability that *some* controlled identity is selected, and then (ii) poison using the selected identities. The wrapper does not prevent (i) at the identity layer; instead it makes (ii) yield no positive expected profit *for each selected identity*, so scaling the number of identities does not scale profitable attack surplus.

**Proposition 5.3 (Sybil-style profit non-positivity).** Assume a uniform deposit $d \ge \bar{g}/(q\delta)$. Consider any real-world actor controlling a set $T \subseteq S$ of identities, each with attack benefit $g_i \le \bar{g}$. For any bids (possibly coordinated across $T$) and any post-selection poisoning plan, the actor's expected incremental payoff from choosing $m_i = 1$ on any subset of selected identities is weakly negative, and strictly negative whenever at least one poisoned selected identity has $g_i > 0$.

*Proof sketch.* Conditional on the realized allocation, the actor's continuation payoff is the sum of the continuation payoffs of its selected identities (utilities are quasilinear and deposits are posted per identity). By Proposition 5.2, for each selected identity $i$ with $g_i \le \bar{g}$, switching from $m_i = 0$ to $m_i = 1$ reduces expected utility. Summing across any number of selected identities preserves non-positivity, so creating more identities cannot create positive expected poisoning profit. □

**What this does and does not guarantee.** The conclusion is a *profit-based* notion of sybil-robustness: an adversary cannot obtain positive expected private benefit from poisoning by scaling identities. This is precisely

the notion needed for marketplaces where attacks are economically motivated (extortion, competitive sabotage, resale value of disruption). It does *not* rule out ideologically motivated attackers willing to incur losses, nor does it address collusion that directly manipulates the base mechanism's approximation properties; those concerns require complementary defenses (e.g., reputation systems, stronger attestations that increase $\delta$, or eligibility screening).

**A practical limitation (liquidity and participation).** Our model treats deposits as refundable transfers with no deadweight cost when honest. In practice, collateral can exclude cash-constrained honest workers or impose opportunity costs. One can partially mitigate this by (i) allowing deposits via credit lines, (ii) risk-tiering deposits using public signals, or (iii) increasing $\delta$ through better attestations so that the required $d$ falls. These considerations do not affect the DSIC logic above, but they matter for participation and thus for the efficiency guarantees we quantify next.

# 6 Theory II: efficiency and budgets.

We now quantify the efficiency consequences of the enforcement wrapper. Because deposits are refunded on-path (when winners behave honestly), the wrapper's *first-order* welfare impact comes from two sources only: (i) the resources spent on auditing, and (ii) any residual harm from undetected malicious behavior when deterrence is imperfect (e.g., when the attacker's benefit exceeds the class bound used to set the deposit, or when one allows loss-incurring attackers).

## 6.1 Financing audits while preserving budget feasibility

Let $a_j$ denote a hard cap on the number of audits associated with requester $j$'s task in a round. The simplest way to make expenditures non-random from the requester's perspective is to reserve a worst-case audit budget up front:

$$\widetilde{B}_j \;:=\; B_j - C_a a_j,$$

and run the base mechanism $M_0$ using $\{\widetilde{B}_j\}$ in place of $\{B_j\}$. (In the common case where requester $j$ aims to hire at most $K_j$ workers and we audit each winner with probability at least $q$, a natural choice is $a_j = \min\{K_j, \lceil qK_j \rceil\}$, which upper-bounds the without-replacement audit rule described earlier.)

**Proposition 6.1 (Ex post budget feasibility with audit reserves).** Suppose the wrapper (i) charges requester $j$ an audit reserve $\phi_j = C_a a_j$ and (ii) runs $M_0$ on budgets $\widetilde{B}_j = B_j - \phi_j$. If the audit rule never audits more than $a_j$ winners attached to requester $j$, then for every realized bid

profile and every realization of the audit lottery, requester $j$'s total outlay (payments to workers plus realized audit spend) is at most $B_j$.

*Discussion.* The point is modular: $M_0$ already guarantees $\sum_i p_{ij} \le \widetilde{B}_j$ ex post, and the wrapper ensures realized auditing costs are bounded by $C_a a_j = \phi_j$. Summing yields $\sum_i p_{ij} + C_a \cdot \#\text{audits}_j \le \widetilde{B}_j + \phi_j = B_j$.

A practical corollary is that we do *not* need to rely on deposit forfeitures to finance audits. In equilibrium under full deterrence (Section 5), forfeitures are essentially off-path revenue; designing audit funding to be budget-feasible without them avoids circularity.

## 6.2 Quality approximation: what is lost relative to the base rule?

Recall that $M_0$ is assumed $\beta$-approximate for the additive proxy objective

$$Q(X) \;:=\; \sum_{i,j} x_{ij} v_i,$$

subject to assignment, compatibility caps $\{|S_j \cap G_\ell| \le \tau_{\ell j}\}$, and requester budgets. Under the deterrence condition (e.g., uniform $d \ge \bar{g}/(q\delta)$), winners play $m_i = 0$ in the continuation game, so the wrapper implements exactly the allocation produced by $M_0$ on the net budgets $\{\widetilde{B}_j\}$. Hence the approximation guarantee transfers directly to the *net-budget* feasible set.

**Proposition 6.2 (Quality guarantee under audit reserves).** Let $\mathrm{OPT}(\widetilde{B})$ denote the maximum achievable proxy quality $Q(\cdot)$ over all feasible allocations under budgets $\{\widetilde{B}_j\}$ and the same compatibility constraints. Then in equilibrium (with honest post-selection behavior),

$$Q(X^{\mathrm{R}}) \;\ge\; \beta \cdot \mathrm{OPT}(\widetilde{B}),$$

where $X^{\mathrm{R}}$ is the wrapped mechanism's allocation.

This statement is intentionally "clean": audits affect efficiency only by shrinking the budgets available for procurement. To relate performance back to the *original* budgets $\{B_j\}$, we need a way to convert dollars of reserved audit spend into a bound on forgone quality. A standard and mild regularity condition is a bounded *quality-to-cost density*: assume there exists $\rho < \infty$ such that for all workers $i$, $v_i \le \rho\, c_i$ (or, more conservatively for analysis, $v_i \le \rho\, b_i$ on truthful bids). This rules out arbitrarily large quality available at arbitrarily small cost, which otherwise makes any additive "loss per dollar" bound impossible.

**Lemma 6.3 (Budget shrinkage implies additive quality loss).** Under the density bound $v_i \le \rho c_i$, for any two budget vectors $B$ and $\widetilde{B} \le B$

(componentwise),

$$\mathrm{OPT}(\widetilde{B}) \;\geq\; \mathrm{OPT}(B) \;-\; \rho \sum_j (B_j - \widetilde{B}_j).$$

With $\widetilde{B}_j = B_j - C_a a_j$, this becomes

$$\mathrm{OPT}(\widetilde{B}) \;\geq\; \mathrm{OPT}(B) \;-\; \rho\, C_a \sum_j a_j.$$

Combining Proposition 6.2 and Lemma 6.3 yields an explicit additive degradation relative to the original-budget optimum:

$$Q(X^{\mathrm{R}}) \;\geq\; \beta\, \mathrm{OPT}(B) \;-\; \beta\, \rho\, C_a \sum_j a_j.$$

Two interpretations are worth emphasizing. First, the $\beta$ factor is inherited from $M_0$; the wrapper does not worsen the approximation ratio *conditional on* the net budgets. Second, the additive term is exactly the "price of enforcement" under deterministic audit reserves: auditing consumes budget that could otherwise purchase quality, at a rate governed by $\rho$.

### 6.3 Accounting for false negatives: residual damage when deterrence is imperfect

The prior bounds treat quality as a function only of allocation and ignore that malicious updates can reduce the *real* requester value. To capture this, let $D_i \geq 0$ denote the (requester-level) damage caused when identity $i$ poisons and is not detected. Conditional on selection and choosing $m_i = 1$, the probability that the damage is realized is $1 - q\delta$. Thus, for any realized winner set $\mathcal{W}$ and any post-selection behavior profile $m$, the expected value loss from false negatives is

$$\mathbb{E}[\text{Damage} \mid X, m] \;=\; (1 - q\delta) \sum_{i \in \mathcal{W}} \mathbf{1}\{m_i = 1\}\, D_i.$$

This yields a generic value bound that cleanly separates *allocation quality* from *integrity loss*:

$$\mathbb{E}[V \mid X, m] \;\geq\; Q(X) \;-\; (1 - q\delta) \sum_{i \in \mathcal{W}} \mathbf{1}\{m_i = 1\}\, D_i,$$

where we are using $Q(X)$ as the additive "gross value" proxy absent poisoning.

When the deposit is chosen so that the bounded adversary class is fully deterred (Section 5), equilibrium has $m_i = 0$ for all winners in that class, and the entire false-negative term drops out. The reason to keep it explicit

is that platforms rarely want to assume *all* attackers satisfy $g_i \le \bar{g}$, and in practice $\delta$ may be materially below one under secure aggregation or weak attestations. In such environments, the wrapper still buys a quantitative improvement: the expected realized damage scales linearly with the false-negative rate $1 - q\delta$. Put differently, holding the set of would-be attackers fixed, improving detection technology ($\delta \uparrow$) or increasing audit intensity ($q \uparrow$) tightens the additive harm term one-for-one.

### 6.4 Putting it together: a welfare-style bound

Finally, if we account directly for audit expenditures (as a real resource cost) and for residual damage, a simple expected welfare proxy takes the form

$$\mathbb{E}[W] \;\approx\; Q(X^{\mathrm{R}}) \;-\; C_a \sum_j \mathbb{E}[\#\mathrm{audits}_j] \;-\; (1 - q\delta) \sum_{i \in \mathcal{W}} \mathbb{P}(m_i = 1)\, D_i,$$

where we have omitted transfers $p_{ij}$ (which cancel in total surplus accounting) and kept the terms that represent real losses. Under the capped audit rule, $\mathbb{E}[\#\mathrm{audits}_j] \le a_j$, so the enforcement-related welfare loss is explicitly bounded by an audit-cost term $C_a \sum_j a_j$ and a false-negative term proportional to $(1 - q\delta)$.

The key message of this section is therefore structural: the wrapper preserves the base mechanism's approximation properties on net budgets, and the efficiency gap to the unwrapped world is controlled by *two* transparent "engineering knobs"—how much budget we reserve for audits, and how often malicious behavior escapes detection. The next section uses this decomposition to discuss design choices and comparative statics in $(q, \delta, d)$.

## 7 Design choices and comparative statics in $(q, \delta, d)$

The prior section isolates the enforcement "price" into two levers: (i) how much auditing we finance, and (ii) how often malicious behavior escapes detection. We now use the same decomposition to guide design choices. The central point is that $(q, \delta, d)$ are not substitutes in a purely mechanical way: deposits deter by shifting the attacker's payoff, while audits and detection technology determine how strongly that shift is applied. In practice, one chooses $(q, \delta, d)$ subject to budget feasibility, liquidity constraints on workers, and engineering limits on $\delta$.

### 7.1 Audit probability versus deposit size: a simple deterrence frontier

Fix a worker identity $i$ who has been selected by $M_0$. Comparing expected utility from being malicious versus honest yields the incentive gap

$$\Delta_i(q, \delta, d_i) \;:=\; \mathbb{E}[u_i \mid x_i = 1, m_i = 1] - \mathbb{E}[u_i \mid x_i = 1, m_i = 0] \;=\; -\kappa(1) + (1 - q\delta) g_i - q\delta d_i.$$

A sufficient condition for strict deterrence for all adversarial identities with $g_i \le \bar{g}$ is

$$q\delta d \;\ge\; (1-q\delta)\bar{g} + \kappa(1), \qquad \text{i.e.} \qquad d \;\ge\; \frac{(1-q\delta)\bar{g} + \kappa(1)}{q\delta}. \tag{1}$$

This is the natural "frontier" that trades audits against deposits: higher audit intensity $q$ (or better detection $\delta$) reduces the deposit needed for deterrence, and conversely a larger deposit allows a smaller audit rate. Our earlier choice $d \ge \bar{g}/(q\delta)$ is a conservative simplification of (1) (it ignores the $(1-q\delta)$ term and any $\kappa(1)$), but the comparative statics are identical.

Two monotonicities are immediate from (1) (taking $\kappa(1)$ as fixed). First, $d$ is decreasing and convex in $q\delta$: marginal improvements in audit *effectiveness* (probability of detection) reduce the required deposit more when $q\delta$ is small. Second, the relevant object is the product $q\delta$. This is useful operationally: one can compensate for weaker attestations (lower $\delta$) by auditing more aggressively (higher $q$), but only up to the point where audit capacity and requester budgets allow.

### 7.2 Why "more deposit, less audit" is not always optimal

If deposits were frictionless—no liquidity limits, no participation distortion, no regulatory cap—then one might push $q$ toward zero and rely on very large $d$ to enforce deterrence. Two practical considerations break this corner solution.

**Liquidity and participation constraints.** Even though deposits are refunded on-path, workers must still *post* them. If some honest, low-cost workers cannot lock up capital, a high uniform $d$ shrinks the effective supply side and can reduce quality by excluding high-$v_i$ participants. A simple way to represent this is a participation feasibility constraint $d \le \overline{d}_i$, where $\overline{d}_i$ is a (private) liquidity limit. Under such constraints, the platform may be forced to choose a larger $q$ (or invest in $\delta$) to meet (1) without exceeding feasible deposits.

**Risk, disputes, and false positives.** Our clean model takes $\delta$ as the probability of detecting malicious behavior conditional on audit and abstracts from false positives. In reality, audits may sometimes wrongly flag honest updates (or be contested), creating expected loss or risk premia for honest workers. A larger $q$ then imposes a direct welfare cost through expected disputes, delayed payments, or risk-bearing, even if deposits are refunded in expectation. This pushes against very large $q$ and motivates using deposits (and better $\delta$) to keep audit intensity moderate.

These frictions suggest that the designer's problem is not merely to satisfy deterrence, but to do so *cost-effectively* subject to operational constraints.

### 7.3 A stylized optimization: picking $q$ given a deposit friction

To make the tradeoff explicit, consider a stylized round in which each requester $j$ hires up to $K_j$ workers and audits each winner independently with probability $q$. Suppose deposits impose a per-winner deadweight cost $\lambda d$ capturing liquidity/risk premia (with $\lambda = 0$ returning us to the frictionless benchmark). Under full deterrence for the bounded class, we can set $d$ on the frontier (1) and minimize an enforcement proxy cost

$$\text{EnfCost}(q) \;\approx\; C_a \cdot q \sum_j K_j \;+\; \lambda \cdot d(q) \sum_j K_j, \qquad d(q) = \frac{(1 - q\delta)\bar{g} + \kappa(1)}{q\delta}.$$

This objective is convex in $q$ on $(0, 1]$ for typical parameter values: the first term increases linearly in $q$, while the second decreases roughly like $1/q$ for moderate $q\delta$. The implication is a genuine interior optimum $q^\star$ whenever $\lambda > 0$: we audit *some*, but not too much, and we require a deposit that is high enough to make the marginal audit unnecessary.

Even when we do not literally solve this optimization, it provides a clear design heuristic:

> Audit more (increase $q$) when audits are cheap ($C_a$ low), deposit frictions are high ($\lambda$ high or liquidity is scarce), or detection is weak ($\delta$ low). Use higher deposits (increase $d$) when audits are expensive, deposit frictions are low, and $\delta$ is strong.

### 7.4 Detection technology $\delta$: attestations as "deposit multipliers"

Equation (1) makes a policy-relevant point: improving $\delta$ has a leveraged effect because it scales the effectiveness of *both* audits and deposits. Concretely, holding $q$ fixed and ignoring $\kappa(1)$ for clarity,

$$d(\delta) \;=\; \frac{(1 - q\delta)\bar{g}}{q\delta} \quad \Rightarrow \quad \frac{\partial d}{\partial \delta} < 0, \;\; \text{and} \;\; d(\delta) \to \infty \text{ as } \delta \downarrow 0.$$

Thus, if secure aggregation or limited telemetry drives $\delta$ materially below one, the deposit required for deterrence can become implausibly large. This is precisely where engineering investments—hardware-backed attestations, provenance checks, robust statistics that increase detectability, or better logging that raises $\delta$—substitute for financial instruments. Put differently, $\delta$ turns out to be a "budget amplifier": by lowering required deposits and/or audits for a fixed security target, it relaxes both liquidity constraints on workers and audit financing pressure on requesters.

Of course, $\delta$ is not free. If raising $\delta$ requires per-round infrastructure cost $C_\delta(\delta)$, then the design problem becomes tri-variate: pick $(q, d, \delta)$ to balance (i) audit cost, (ii) deposit frictions, and (iii) detection-tech investment. The

comparative statics remain intuitive: improvements in $\delta$ are most valuable when $C_a$ is high, when deposit frictions bind, or when the threat class $\bar{g}$ is large.

### 7.5 When do we need numerical optimization?

Closed-form comparisons become unreliable once we move beyond the single-parameter bounded-adversary abstraction. Three features in particular push the design problem into "simulate and optimize" territory.

**Heterogeneous and state-dependent risk.** If workers differ in $g_i$, $D_i$, or in detectability (effectively $\delta_i$), then the optimal policy is rarely uniform. One may prefer risk-based audits $q_i$ or tiered deposits $d_i$ (e.g., higher $d_i$ for accounts with weak provenance). But risk-based rules interact with DSIC and feasibility: changing eligibility via deposit requirements can alter bids, participation, and the allocation $X$ in ways that are difficult to bound analytically.

**Compatibility constraints and multi-requester coupling.** The caps $|S_j \cap G_\ell| \le \tau_{\ell j}$ create nontrivial substitutability patterns across worker types. Excluding some identities via large deposits can force the mechanism to backfill with lower-$v_i$ workers from other groups, and the welfare impact depends on the entire feasible matching polytope. These effects are typically instance-specific, suggesting numerical evaluation under realistic group structures.

**Partial deterrence and endogenous attacking.** If the threat model admits attackers with $g_i > \bar{g}$, or if some attackers are willing to incur losses (e.g., ideological sabotage), then the equilibrium may feature a nonzero attack rate that depends on $(q, \delta, d)$. Bounding expected damage then requires modeling (or estimating) the distribution of $g_i$ and $D_i$, and integrating over the induced "attack region" $\{\Delta_i(q, \delta, d) > 0\}$. This is exactly the situation in which empirical calibration and numerical search are warranted.

These considerations motivate an empirical plan that treats $(q, \delta, d)$ as tunable knobs and evaluates the resulting accuracy, spend, and integrity outcomes under realistic poisoning and sybil behaviors, including settings where $\delta < 1$ due to secure aggregation and where audits rely on noisy proxies.

## 8 Empirical plan: poisoning and sybil simulations under compatibility constraints

Our theoretical results isolate a small set of design knobs—$(q, \delta, d)$ layered on top of a DSIC, budget-feasible base mechanism $M_0 = (X, P)$—and characterize how they deter bounded attackers in expectation. The natural next

step is to test how these knobs perform in regimes that matter operationally: (i) poisoning attacks that manifest as downstream model degradation rather than direct payment fraud, (ii) sybil behavior in which one actor controls many identities and strategically bids to increase selection probability, (iii) compatibility constraints $|S_j \cap G_\ell| \leq \tau_{\ell j}$ that couple requesters and shape who can be hired, and (iv) secure aggregation or limited telemetry that reduces what the auditor can observe, effectively lowering $\delta$. In this section we lay out a simulation-based empirical plan that maps these primitives into measurable outcomes and produces actionable design charts.

## 8.1 Simulation environment: multi-requester procurement with group caps

We propose a Monte Carlo harness that generates instances $\mathcal{I}$ consisting of: (i) a set of requesters $A$ with budgets $B_j$ and desired cohort sizes $K_j$; (ii) a pool of worker identities $S$, each with a group label in one or more incompatibility partitions $G_\ell$ (e.g., jurisdiction, channel, provenance class); (iii) compatibility caps $\tau_{\ell j}$ per requester; and (iv) worker primitives $(c_i, v_i)$ plus a strategic bid $b_i$. For each draw, we run $M_0$ to obtain $(X, P)$ subject to budgets and caps, then overlay deposits, audits, and forfeitures. The key is to generate instances where caps sometimes bind: we will vary the tightness of $\tau_{\ell j}$ (from slack to binding) and the concentration of high-$v_i$ workers within particular groups, because these are the cases in which excluding identities via deposit requirements can have the largest allocative effect.

To connect to federated learning, we additionally attach to each selected worker an update-generation process. For a given requester $j$, selected identities $S_j = \{i : x_{ij} = 1\}$ produce local updates under a chosen FL algorithm (e.g., FedAvg with fixed client epochs), and the requester's realized value $V_j(S_j)$ is proxied by final validation accuracy (and, when relevant, robustness metrics). This lets us evaluate not only the proxy objective $\sum_{i,j} x_{ij} v_i$ but also task-level performance under adversarial behavior.

## 8.2 Threat models: poisoning, sybils, and collusion

We simulate three increasingly adversarial regimes.

**Independent bounded attackers.** A subset of identities are adversarial with one-shot benefit $g_i$ from being malicious and undetected, with $g_i$ drawn from a distribution supported on $[0, \bar{g}]$ in the baseline. Given selection, each adversarial identity chooses $m_i \in \{0, 1\}$ to maximize expected utility under the implemented $(q, \delta, d)$. This regime directly tests whether the deterrence logic observed in the model appears in end-to-end training outcomes.

**Sybil actors with many identities.** We introduce real-world actors $r \in R$, each controlling a set of identities $S(r) \subseteq S$. A sybil actor chooses bids $\{b_i : i \in S(r)\}$ (and, if modeled, participation decisions under deposit requirements) to maximize total expected utility across its identities. We stress-test both "horizontal" sybils (many low-bid identities to increase selection probability) and "targeted" sybils (tuning bids to exploit cap structure so that their identities become pivotal substitutes within a constrained group). We treat group labels as potentially correlated within an actor (e.g., all sybils share the same provenance class) to capture the operational idea that provenance-based caps can limit sybil amplification.

**Poisoning implementations and damage measurement.** Conditional on $m_i = 1$, we implement concrete update manipulations: label-flip, sign-flip, norm inflation, model replacement, and backdoor insertion with a chosen trigger. For each requester, we measure damage $D_i$ in terms of (a) drop in clean validation accuracy, (b) increase in loss on a protected holdout, and (c) backdoor success rate. These metrics provide an empirical counterpart to the welfare loss from undetected malicious participation.

### 8.3 Auditing under limited observability: proxies for $\delta$ with secure aggregation

In practice, the auditor rarely observes raw per-client updates; under secure aggregation, it may only see the aggregate update, possibly with privacy noise. We therefore treat $\delta$ as an *effective* detection probability induced by an auditing protocol and the system's observability constraints, and we measure it empirically rather than assuming it.

We consider several audit proxies, each with a tunable threshold that traces out an ROC curve (true-positive rate versus false-positive rate): (i) *challenge rounds* in which a small audited subset is required to train on a hidden canary dataset or produce a verifiable statistic; (ii) *influence-based tests* that estimate whether removing a client's contribution materially improves holdout loss (possible when per-client contributions are partially recoverable via secure enclaves or cryptographic commitments); (iii) *consistency checks* such as update norm bounds, cosine similarity to the aggregate direction, or agreement with a robust estimator (median, trimmed mean) computed within an audit enclave; and (iv) *attestation-based audits* (hardware-backed measurements, provenance proofs) that detect specific classes of policy violations.

For each proxy, we estimate $\widehat{\delta}$ by injecting known malicious behavior into audited identities and recording detection frequency, while separately measuring the false-positive rate $\alpha$ on honest identities. Although our clean model abstracts from $\alpha$, the simulations will report how $\alpha$ interacts with participation (through perceived risk) and with realized audit expenditures

(through disputes and re-training). Operationally, this produces a mapping from "engineering choices" to the economic parameter $\delta$, allowing us to plot feasible $(q, \delta, d)$ triples given observability constraints.

### 8.4 Metrics: accuracy, value/reputation, spend, and integrity

We propose four metric families, reported both per requester and aggregated across requesters.

**Learning outcomes (accuracy and robustness).** For each requester $j$, we report final validation accuracy, worst-group accuracy (if the task is group-fairness sensitive), and robustness measures such as backdoor success rate and accuracy under distribution shift. These are the primary "customer-facing" outcomes and provide an empirical analogue of $V_j(S_j)$.

**Procurement quality and reputational value.** We report the proxy objective $\sum_{i,j} x_{ij} v_i$, the realized allocation composition across groups $|S_j \cap G_\ell|$, and concentration indices (e.g., Herfindahl) to capture how caps and deposits shape diversity. When $v_i$ is interpreted as a reputation score, we also track how often high-$v_i$ identities are excluded by deposit frictions, which is an empirically important channel through which overly aggressive $d$ can harm outcomes even when it improves integrity.

**Spend and budget feasibility.** We track requester payments $\sum_i p_{ij}$, audit fees $\phi_j$ (or total audit cost $C_a \cdot$ #audits), total deposits posted, and forfeitures collected. The headline statistic is the fraction of instances in which each requester remains within budget after accounting for audit financing rules. We also report a decomposition of enforcement cost into (i) audit spend, (ii) liquidity deadweight (modeled or measured as dropout), and (iii) residual expected damage from undetected attacks.

**Integrity: attack prevalence and undetected harm.** We report the equilibrium attack rate among selected adversarial identities (empirically induced by the utility comparison), the fraction of malicious identities detected, and the expected number of malicious-but-undetected winners. We additionally report the distribution of realized damage conditional on undetected attacks, since heavy-tailed damage is precisely where average-case deterrence can be insufficient for practice.

### 8.5 Experimental design: parameter sweeps and stress tests

We will run factorial sweeps over $(q, d)$ and audit-proxy settings (which determine $\delta$ and $\alpha$) across multiple instance classes: (i) slack versus binding $\tau_{\ell j}$; (ii) low versus high audit cost $C_a$; (iii) weak versus strong observability

(secure aggregation noise levels, enclave availability); (iv) deposit frictions (dropout probability as a function of $d$, or hard liquidity caps $d \leq \overline{d}_i$); and (v) threat severity (support of $g_i$, including misspecification where some attackers have $g_i > \bar{g}$).

For each cell, we compute confidence intervals over many draws and produce design charts that answer two practical questions: (a) what $(q, d)$ achieves a target integrity level (e.g., expected undetected malicious winners $\leq \varepsilon$) given a measured $\widehat{\delta}$; and (b) among the policies that meet the target, which minimizes enforcement-adjusted spend while maintaining learning performance.

Finally, we benchmark allocations against two comparators: the base mechanism without enforcement, and an "oracle" that excludes all malicious identities (or uses perfect detection). This quantifies the performance gap attributable to partial verification and provides a disciplined way to interpret residual losses due to $(1 - q\delta)$ events in realistic training pipelines.

This empirical plan is intended to be modular: it treats $M_0$ as a black box satisfying the usual feasibility and incentive properties, while making the integrity layer testable under realistic FL observability constraints. The next section discusses extensions that naturally arise once the empirical evidence reveals where uniform deposits and uniform auditing are too coarse.

# 9 Discussion and extensions: heterogeneity, targeting, dynamics, and partial verification

Our baseline integrity layer is deliberately blunt: we pick an audit probability $q$, an effective detection probability $\delta$, and a refundable deposit $d$ (possibly uniform), and we show how this suffices to deter any identity with $g_i \leq \bar{g}$ in one-shot expected utility. The appeal of this starting point is implementability and clean incentive separation: the base mechanism $M_0$ handles cost revelation and feasibility, while $(q, \delta, d)$ acts as an ex post enforcement overlay. In practice, however, platforms will want to tailor enforcement to heterogeneous risks, learn over time, and operate under partial observability. We discuss several extensions that preserve the spirit of the model while clarifying where additional design work is needed.

## 9.1 Heterogeneous deposits and liquidity constraints

Uniform deposits are attractive operationally, but they are rarely efficient. In most FL marketplaces, the attack benefit $g_i$ varies widely with (i) the strategic value of the task (e.g., competitive intelligence), (ii) the attacker's outside options, and (iii) the ease with which poisoning yields downstream harm. If we can bound $g_i$ more tightly for some identities (via provenance,

reputation, or task class), then the canonical deterrence condition

$$d \;\geq\; \frac{\bar{g}}{q\delta}$$

is overly conservative. A natural extension is a per-identity (or per-class) deposit schedule $d_i$, with deterrence requiring $d_i \geq g_i/(q\delta)$ for the relevant bound on $g_i$. Economically, this is a standard screening-versus-friction tradeoff: higher $d_i$ reduces expected attack rents but can exclude honest, liquidity-constrained workers, thereby harming the proxy objective $\sum_{i,j} x_{ij} v_i$ and (more importantly) realized learning value $V_j(S_j)$.

Two implementation constraints matter. First, we typically cannot condition $d_i$ on private $g_i$, but we can condition on public or auditable correlates: account age, stake history, verified hardware attestation, jurisdictional licensing, or coarse risk tiers. Second, heterogeneous deposits interact with compatibility caps $|S_j \cap G_\ell| \leq \tau_{\ell j}$. When caps bind, excluding even a small set of identities within a scarce group (e.g., a regulated jurisdiction) can force the mechanism to substitute into low-$v_i$ or high-cost workers outside that group, magnifying the allocative cost of enforcement. This suggests a practically useful design rule: if caps are tight in some $G_\ell$ for key requesters, deposit schedules should be *group-aware*, but in the opposite direction one might first guess—i.e., avoid imposing the highest liquidity burdens on the scarcest groups unless their risk is truly dominant.

A simple way to formalize this is to augment the platform's objective with an explicit participation (or dropout) function $\pi_i(d_i) \in [0,1]$, decreasing in $d_i$, and treat the induced expected feasibility set as stochastic. Even without fully solving the resulting optimization, the comparative statics are clear: when $\pi_i$ is steep (high liquidity sensitivity), audit improvements that raise $\delta$ are first-best relative to raising $d_i$, because they reduce required deposits without reducing participation.

## 9.2 Endogenous audit targeting without breaking incentives

Uniform auditing is similarly coarse. Audits are costly ($C_a$) and attention-limited, so it is tempting to target audits toward "riskier" winners. The challenge is incentive compatibility: if audit probability depends on a worker's bid $b_i$ or on allocation-relevant reports in a way that changes expected utility from misreporting costs, we may inadvertently undermine DSIC properties of $M_0$.

One robust guideline is to restrict targeting to signals that are (i) exogenous to the bid and (ii) not manipulable by the worker at the time of bidding. Examples include provenance class, device attestation status, historical dispute rate, or a coarse risk score computed from past audits (with appropriate safeguards against strategic manipulation). Let $q_i$ denote an identity-specific audit probability chosen after allocation. Then deterrence

becomes

$$d_i \;\geq\; \frac{\bar{g}_i}{q_i \delta_i},$$

where $\delta_i$ may also vary with observability (e.g., some devices support richer attestations). This formulation highlights a useful substitution: targeting can raise $q_i$ for high-risk identities while keeping average audit load $\sum_{i \in \mathcal{W}} q_i$ (and thus expected cost) fixed.

Targeting also interacts with sybil incentives. If the platform's targeting rule is monotone in features that sybils can cheaply imitate, then the main effect may be to increase the attacker's audit exposure without reducing their selection probability, which is desirable. But if the targeting rule inadvertently lowers audits on "fresh" accounts (e.g., to encourage onboarding), it can create an obvious sybil channel. Our model therefore suggests a policy-level constraint: onboarding subsidies, if any, should be implemented via payments or reduced deposits for verified low-risk classes, not via reduced audit probability.

### 9.3 Repeated interaction, reputation, and dynamic deterrence

The one-shot model treats malicious benefit $g_i$ as immediate and ignores future consequences beyond deposit forfeiture. Many FL procurement settings are repeated: workers participate in multiple rounds, and requesters (or platforms) maintain reputational histories. Repetition can strengthen deterrence substantially, potentially allowing smaller deposits.

A parsimonious extension is an infinite-horizon model where an identity that is detected malicious is banned (or suffers a reputation drop that reduces future selection probability or payments). Let $R_i$ denote the present value of expected future surplus from continued participation when honest. Then the expected loss from being caught includes not only $d_i$ but also the continuation loss $R_i$. The deterrence constraint becomes

$$(1 - q\delta) g_i \;\leq\; q\delta \left(d_i + R_i\right) + \kappa(1),$$

which immediately implies that even modest deposits can deter sizable $g_i$ when $R_i$ is large (i.e., when the identity has valuable future rents). This dynamic perspective also clarifies why sybils are challenging: newly created identities have low $R_i$, so deposits and audits must shoulder more of the burden early in the lifecycle. Operationally, this motivates "stake-building" regimes in which $d_i$ (or $q_i$) is higher for low-history accounts and decreases as verifiable honest participation accumulates.

Repeated interaction also raises an important limitation: collusive attackers may coordinate across rounds, sacrificing some identities (burner sybils) to learn audit thresholds and preserve others. This pushes the design toward randomized audits and rotating challenge tasks, consistent with our

interpretation of $\delta$ as an effective detection probability induced by a family of audit protocols rather than a fixed test.

## 9.4 Partial verification, false positives, and multi-dimensional misbehavior

We have treated $\delta$ as a reduced-form probability of detecting malicious behavior conditional on audit, and we have abstracted from false positives. This is analytically convenient but operationally incomplete. In FL, many "audits" are statistical and noisy: they can have nontrivial false-negative rates for sophisticated poisoning and nontrivial false-positive rates for benign but heterogeneous data (which can look anomalous).

A more realistic variant distinguishes attack types $t \in T$, each with benefit $g_i(t)$ and detection $\delta(t)$. The relevant constraint for a worst-case bounded adversary becomes

$$d_i \;\geq\; \max_{t \in T} \frac{\bar{g}_i(t)}{q\,\delta(t)}.$$

This immediately highlights why "robust aggregation alone" is not a substitute for enforcement: if some attacks have very low $\delta(t)$ under secure aggregation, deposits must rise sharply unless we can change the audit technology (increase $\delta(t)$) or the audit frequency $q$. Conversely, if $\delta(t)$ is high for the attacks that matter most for damage $D_i(t)$, then enforcement can be targeted to those classes without over-penalizing benign outliers.

Introducing false positives $\alpha$ changes incentives for honest workers: expected utility from participation is reduced by the chance of erroneous forfeiture or costly disputes. In a refined model, one would either (i) make forfeiture contingent on a higher standard of proof (reducing $\delta$ but also reducing $\alpha$), (ii) allow an appeals process (adding cost and delay), or (iii) replace strict forfeiture with a graduated penalty. The main design takeaway is that $\delta$ and $\alpha$ should be treated jointly: raising $\delta$ by lowering thresholds can backfire if $\alpha$ induces honest dropout, especially in capped groups where participation is scarce.

## 9.5 Budget feasibility with audit financing and forfeiture recycling

Our baseline accounting treats audit expenditures as an add-on cost that must fit within requester budgets net of some financing rule. In practice, platforms will also observe deposit inflows and forfeitures. A tempting policy is to "recycle" expected forfeitures to finance audits, reducing requester fees $\phi_j$. However, because deterrence aims to make attacks unprofitable, equilibrium forfeitures may be rare; relying on them for financing can therefore be fragile.

A conservative approach is to require ex ante budget feasibility without counting on forfeitures, treating forfeitures as windfall that can fund public goods (e.g., improved auditing) or be rebated in a way that does not distort incentives. If rebates depend on bids or allocations, they can again interfere with DSIC; if they are lump-sum (e.g., periodic fee reductions for all requesters), they are safer.

### 9.6 Limitations and scope conditions

Several assumptions delimit what our mechanism guarantees. First, the sybil-proofness statement is bounded: we require $g_i \leq \bar{g}$ for the relevant adversary class. If an attacker's benefit is unbounded (e.g., catastrophic sabotage value), no finite deposit is sufficient, and the design must shift toward strong identity, strict access control, or high-assurance verification that raises $\delta$ near one. Second, we treat deposits as frictionless aside from liquidity; in reality, deposits may be legally constrained, operationally costly to escrow, or infeasible for some worker populations, raising equity and access concerns. Third, we have not modeled strategic requesters or cross-requester externalities in detail; a requester may benefit from sabotaging another's model, which would shift incentives and potentially justify different audit financing rules.

Finally, our analysis takes $M_0$'s DSIC and feasibility properties as primitives. While the integrity overlay is designed to be separable, some extensions—especially audit targeting based on endogenous signals—can inadvertently reintroduce incentive concerns. For deployment, the practical discipline is to keep enforcement rules as bid-independent as possible and to document any remaining dependencies explicitly.

These extensions clarify the broader message: deposits and audits are not a substitute for careful mechanism design, but they are a tractable and modular way to harden an otherwise standard budget-feasible procurement pipeline. In the conclusion, we distill this into a deployable recipe—what to measure, what to set, and how to iterate when observability, liquidity, and threat severity change.

## 10 Conclusion: a deployable recipe for hardening budget-feasible FL procurement

We can now state the main practical implication of the model in operational terms. A budget-feasible FL procurement mechanism can be hardened against a broad class of modern threats by treating integrity as a *modular enforcement layer*—implemented with deposits and audits—that sits on top of a standard DSIC, feasibility-respecting procurement rule. The base mechanism $M_0 = (X, P)$ continues to do what it is designed to do: elicit costs,

satisfy budgets and compatibility caps, and approximately maximize a transparent proxy objective (e.g., $\sum_{i,j} x_{ij} v_i$). The integrity layer $(q, \delta, d)$ does something conceptually orthogonal: it prices the post-selection deviation (poisoning, noncompliance, data exfiltration) by ensuring that the expected marginal gain from misbehavior is negative for any bounded adversary.

A key virtue of this separation is governance: it lets platforms improve security posture without repeatedly rewriting the economic mechanism. In particular, provided auditing and deposit rules are kept bid-independent (or depend only on non-manipulable signals), the DSIC logic of $M_0$ is preserved for honest participants. The hard part becomes *calibration*: choosing enforcement parameters that deter attacks while preserving participation, respecting requester budgets, and operating under partial observability.

We therefore close with a deployable recipe—a sequence of design steps that map naturally to quantities the platform can estimate, commit to, and iteratively refine.

**Step 1: Define the adversary class and a defensible bound $\bar{g}$.** The deterrence logic is only as meaningful as the bound on malicious upside. In many FL deployments, the relevant one-shot benefit $g_i$ is not literally "profit from poisoning" but the value of (i) degrading a competitor's model, (ii) extracting sensitive information, or (iii) causing reputational harm. A platform does not need to identify each $g_i$, but it must commit to a threat model that implies a policy-level bound $\bar{g}$ for the class of identities to be deterred by economic enforcement. This is where organizational practice enters: $\bar{g}$ can be task-class specific (e.g., higher for high-stakes domains), jurisdiction-specific, or aligned with a compliance standard. The model's warning is crisp: if catastrophic sabotage yields effectively unbounded $g_i$, then deposits cannot be the primary defense; one must instead raise $\delta$ via stronger verification and access control.

**Step 2: Choose an audit protocol and treat $\delta$ as an engineering KPI.** In the mechanism, $\delta$ is a reduced-form detection probability conditional on audit. Operationally, $\delta$ is shaped by tooling: update validation, challenge tasks, secure enclaves/attestation, statistical backdoor tests, canary data, and forensics. The most important design discipline is to convert security investments into an explicit "effective detectability" target. In our framework, raising $\delta$ is economically powerful because it reduces the deposit needed for deterrence without excluding liquidity-constrained honest workers. This suggests an implementable prioritization rule: if participation is scarce or compatibility caps are tight in critical groups $G_\ell$, investments that increase $\delta$ are typically less distortive than increases in $d$.

**Step 3: Set the deterrence parameters via a transparent inequality.** Given an audit probability $q \in (0, 1]$ and effective detection $\delta \in (0, 1]$, a uniform deposit $d$ that ensures strict unprofitability for any identity with $g_i \leq \bar{g}$ is

$$d \;\geq\; \frac{\bar{g}}{q\delta}.$$

This is the simplest "knob-turning" rule that can be communicated to stakeholders and audited externally. If the platform uses heterogeneous enforcement (by task class or identity tier), the same logic applies with class-specific bounds and audit rates:

$$d_k \;\geq\; \frac{\bar{g}_k}{q_k \delta_k} \qquad \text{for each risk tier } k.$$

We emphasize what the inequality means for deployment: lowering deposits without compensating increases in $q\delta$ is not merely "taking some risk"; it reintroduces positive expected attack rents, which in turn invites sybils and repeated attempts. Conversely, if a platform is willing to impose large deposits, it can economize on audits; but this shifts cost from requester budgets to worker liquidity and may violate inclusion objectives.

**Step 4: Make audit financing budget-feasible ex ante.** Audits cost real resources $C_a$. The implementable constraint is that requester budgets $B_j$ must cover expected procurement payments *plus* expected audit costs under the chosen policy, without relying on forfeitures. A simple conservative accounting is: when requester $j$ receives $k_j$ winners, expected audits are $qk_j$, so expected audit cost is $C_a q k_j$. The platform can then (i) charge requesters an explicit fee $\phi_j = C_a q k_j$, or (ii) deduct an expected audit reserve from the budget used by $M_0$. The important governance point is to avoid equilibria that depend on "attack revenue" to fund enforcement. If deterrence works, forfeitures are rare; robust systems treat forfeitures as windfalls that can be used for security upgrades or neutral rebates that do not feed back into allocation incentives.

**Step 5: Keep the integrity layer strategically "orthogonal" to bidding.** The easiest way to preserve DSIC properties of $M_0$ is to commit that $q$, $\delta$, and $d$ depend only on bid-independent information: task class, verified device capabilities, provenance tiers, or history-based reputation that cannot be cheaply manipulated at bid time. If audit probability is made a function of $b_i$ (or other allocation-relevant reports), then a worker's expected utility changes with misreports in a way that can break the base mechanism's incentive properties. When targeting is essential, the safe compromise is: compute a risk tier before bids are submitted or using strictly exogenous signals, then run $M_0$ within the induced eligible set.

**Step 6: Integrate sybil resistance through *per-identity* economics and lifecycle policy.** Because deposits are posted per identity, the deterrence condition scales to an arbitrary number of accounts: if each identity faces negative expected profit from attacking, creating more identities does not help. However, platforms must enforce that deposits are genuinely identity-binding (e.g., cannot be trivially laundered or insured) and must decide how quickly new identities can "graduate" to lower-friction tiers. A practical lifecycle regime is: higher $q$ and/or $d$ for low-history accounts, with reductions contingent on verifiable honest participation. This is not merely punitive; it is the economic substitute for the continuation value $R_i$ that established identities naturally have in repeated settings.

**Step 7: Monitor the welfare gap and iterate on the right margin.** Relative to the base $\beta$-approximation, the integrity layer introduces two mechanical wedges: audit cost (scaling with $C_a$ and audit frequency) and residual undetected risk (scaling with $1 - q\delta$). These wedges are policy levers. If operational metrics show high honest dropout, the correct response is usually to improve $\delta$ (better audits) or adjust tiering, not simply to reduce $d$ while keeping $q\delta$ fixed. If audits are too expensive, one can lower $q$ only if deposits rise accordingly or if the bound $\bar{g}$ can be justified as lower for the relevant class. The model thus yields a disciplined "change management" rule: any reduction in one enforcement dimension must be compensated by an improvement in another, or by a credible tightening of the threat bound.

**Step 8: Document scope conditions and failure modes.** Finally, deployment requires explicit statements of what the mechanism does *not* guarantee. Deposits and random audits deter bounded, myopic deviations; they are not a substitute for secure engineering against unbounded adversaries, nor do they by themselves resolve false-positive disputes, collusion, or sophisticated attacks that drive $\delta$ close to zero. Likewise, deposit requirements raise equity and access concerns; if liquidity constraints bind, the platform may need subsidized credit, insurance products, or alternative enforcement (higher $\delta$ via verification) to avoid excluding valuable participants, especially within constrained compatibility groups.

Putting these steps together, the implementation philosophy is straightforward: commit to a base procurement mechanism with clear incentive and feasibility guarantees, then choose an integrity overlay that (i) makes post-selection misbehavior strictly unprofitable up to a stated bound, (ii) fits within budgets without relying on misbehavior for funding, and (iii) remains as bid-independent as possible. In this sense, the model illuminates a trade-off that practitioners already manage informally: when observability is weak ($\delta$ low) and audits are expensive ($C_a$ high), integrity must be purchased with either higher deposits (liquidity burden) or stricter access controls (reduced

participation). Making that tradeoff explicit is the core contribution of the framework, and the reason a "recipe" can be credibly deployed rather than merely discussed.