DriftPay: Budget-Feasible Truthful Procurement for Federated Learning via Model-Contrastive Representation-Drift Scores

Liz Lemma   Future Detective

January 14, 2026

Abstract

In modern federated learning (FL) deployments, worker "quality" is not a static reputation but an endogenous property of the learning dynamics under heterogeneous (non-IID) data. Existing budget-feasible FL procurement mechanisms—e.g., compatibility-aware mechanisms that handle multiple budgeted requesters and incompatible workers—optimize exogenous reputation proxies and therefore misprice contribution when drift and representation collapse dominate performance. We propose DriftPay, a compatibility-aware reverse-auction mechanism that replaces reputation with an algorithm-aware, MOON-style representation-drift score computed on a fixed probe set using the announced training rule. This score is bid-independent, stable under secure aggregation noise, and predictive of generalization in heterogeneous regimes. Building on CARE's critical-price and max-flow allocation machinery, DriftPay selects workers to maximize total drift-score subject to per-requester budgets and compatibility caps, pays threshold prices, and guarantees ex post IR, DSIC, budget feasibility, and constant-factor approximation to an oracle that knows workers' true costs and drift-scores. We further prove robustness: estimation noise in drift-scores induces only bounded welfare loss. Empirically (to be developed), DriftPay improves accuracy-per-dollar and reduces vulnerability to gaming relative to reputation-based baselines in multi-tenant FL.

Table of Contents

1. 1. Introduction: multi-tenant FL procurement in 2026; why exogenous reputation fails; preview of DriftPay and guarantees.
2. 2. Related Work: budget-feasible mechanism design; FL incentives (reverse auctions); incompatibility/compatibility constraints; MOON and representation drift; contribution measurement.

3. 3. System and Scoring Model: MOON-style drift scoring on a probe set; compatibility groups; budgets; strategic costs; discussion of privacy/secure aggregation and what is observable.

4. 4. Baseline Benchmarks and Failure Modes: reputation proxy mismatch under non-IID; accuracy-based scoring instability; manipulation channels; motivation for drift-score.

5. 5. The DriftPay Mechanism (Pooled Budgets): allocation rule via critical price and max-flow; threshold payment rule; complexity.

6. 6. Theoretical Guarantees (Pooled Budgets): DSIC, ex post IR, budget feasibility, approximation; closed-form bound.

7. 7. Extensions: (a) Non-cooperative per-requester budgets (sketch adapting PEA/Care-NO ideas), (b) submodular drift-value variant (coverage of representation clusters), (c) optional performance-adjusted bonuses with IC caveats.

8. 8. Robustness to Noisy Drift Estimates: welfare sensitivity bounds; conditions under which DSIC is preserved; audit/refresh of scores.

9. 9. Empirical Plan (Brief): multi-tenant FL simulation; non-IID via Dirichlet; compare to CARE with reputation and to accuracy-based scoring; sensitivity to score noise and congestion.

10. 10. Conclusion: mechanism–algorithm co-design as a 2026 agenda; limitations and next steps.

1 1. Introduction: multi-tenant FL procurement in 2026; why exogenous reputation fails; preview of DriftPay and guarantees.

By 2026, federated learning (FL) has largely moved from a single "model owner + voluntary clients" paradigm to a multi-tenant procurement setting. In a typical deployment, a platform intermediates between multiple requesters (model owners) who want to train comparable models and a population of heterogeneous workers (clients) who can supply local compute and data. The practical constraint is no longer merely participation, but *budgeted selection*: requesters must decide whom to pay, under hard spending limits and operational constraints (compliance, fairness, or risk controls) that cap how many participants can be drawn from particular regions, device types, institutions, or other categories. This shift makes the economics of FL look less like a cooperative protocol and more like a market for (privacy-preserving) learning contributions.

A first instinct in such markets is to import familiar tools: reputation systems, historical win-rates, or coarse quality tiers. Yet in FL these "exogenous reputation" proxies systematically misfire. The reason is not only strategic (workers can game reputational signals) but also statistical and algorithmic. When data are non-IID, the marginal value of a worker is highly state-dependent: it varies with the current global parameters, with the training algorithm's inductive biases, and with the particular mixture of other selected workers. A client that was "high quality" last month, or under a different objective, may produce an update that is redundant, misaligned, or even harmful under today's representation-learning dynamics. In other words, reputation is typically *time-invariant* and *task-agnostic*, while FL contributions are *round-dependent* and *algorithm-aware*. This wedge becomes especially salient in modern contrastive and representation-alignment methods (e.g., MOON-family objectives), where the relevant question is not whether an update is large, but whether it *moves representations in a direction that generalizes* for the requester.

We therefore start from an operational premise: the platform should score potential workers using an announced, bid-independent rule that is aligned with the learning objective, and then run a procurement mechanism on top of those scores. Concretely, before bids are considered, the platform computes for each worker a *representation drift-score* on a fixed probe dataset. Intuitively, this score measures whether the worker's standardized local update produces a representation consistent with the current global representation on inputs that all parties agree are relevant for evaluation. The probe set can be public, requester-provided, or constructed from a permissible reference distribution; what matters for incentives is that it is fixed ex ante and that the scoring rule is committed to in advance. This design choice is more than an ML detail: it is the key that keeps private information one-dimensional (the cost), thereby enabling dominant-strategy incentive compatibility in a reverse auction environment.

The procurement problem we address is thus: given costs privately known to workers, publicly known budgets on the requester side, and group-based compatibility caps, how should the platform assign workers to requesters and set payments so that (i) workers truthfully reveal costs, (ii) payments respect budgets, and (iii) the resulting assignment achieves high total drift-score? The presence of compatibility constraints is not cosmetic. In multi-tenant FL, requesters often cannot absorb arbitrary concentrations of clients from a single group (e.g., one hospital network, one jurisdiction, one device manufacturer) because of governance rules, risk concerns, or simply to hedge correlation in data sources. These caps create complementarities and "packing" effects: the best worker by score-per-dollar may be infeasible to use if the relevant group quota is already saturated for a given requester. Any practical mechanism must therefore optimize subject to these caps rather than treat the selection as a simple knapsack.

Standard mechanism design benchmarks do not directly resolve this. VCG-style mechanisms are neither budget-feasible nor computationally convenient in the large-scale, constrained setting that FL platforms face. Conversely, ad hoc heuristics that "pay top-k" workers or that greedily assign by reputation typically lack incentive guarantees: small changes in bids can create discontinuous jumps in allocation, and payments may exceed budgets or fail individual rationality when constraints bind. Our approach is to adapt the budget-feasible reverse auction template to an FL-native notion of value—drift-score—while explicitly accommodating group/requester caps.

We call the resulting mechanism *DriftPay* (and, in its compatibility-constrained pooled-budget instantiation, DriftPay-CO). At a high level, DriftPay proceeds in three conceptual steps. First, it ranks workers by a cost-effectiveness ratio, "bid per unit drift-score," thereby formalizing the platform's preference for low-cost, high-alignment contributors. Second, it identifies a maximal affordable prefix of this ranking under the available budget. Third, it computes a feasible assignment of that prefix to requesters subject to the compatibility caps and the rule that each worker can be assigned at most once. The assignment step is not left as an oracle: with additive drift-scores and cap constraints, it admits a polynomial-time reduction to a max-flow problem. This is important for practice, because it yields an implementable mechanism that scales with the market size rather than relying on exponential search or brittle local heuristics.

Payments follow the familiar "threshold" logic that underpins truthful procurement: each selected worker is paid a critical amount that depends on the first excluded worker (or on the binding budget constraint), not on the worker's own bid. This structure is what makes bidding truthfully a dominant strategy: if a worker raises her bid, she can only hurt her chances of being selected; if she lowers it, she risks being selected at a payment that no longer covers her true cost. At the same time, by calibrating the threshold using the affordable prefix, DriftPay maintains budget feasibility by construction: total payments cannot exceed the budget available to the platform (either a pooled budget B or requester-level budgets B_j, depending on the variant). Ex post individual rationality follows because any selected worker is paid at least her reported bid, and truthful bidding implies payment covers true cost.

A central practical concern is measurement error in contribution signals. Drift-scores are computed from finite probe data, possibly with noise from stochastic evaluation, quantization, or secure aggregation constraints. We therefore also analyze a robust variant in which the platform observes $\hat{v}_i$ satisfying a multiplicative error bound relative to the true drift-score. The key observation is that incentive compatibility is preserved so long as the score remains bid-independent: noisy scoring may degrade welfare, but it does not reintroduce a direct channel by which workers manipulate allocation through bids. Our welfare guarantees degrade gracefully with the noise level, providing a transparent tradeoff between scoring fidelity and economic efficiency.

Our contributions are therefore conceptual, methodological, and practical. Conceptually, we argue that the right "currency" for FL procurement is not generic reputation but an algorithm-aware, state-dependent contribution proxy that can be computed prior to bidding. Methodologically, we combine budget-feasible reverse auction design with a compatibility-constrained assignment subroutine that is polynomial-time via flow, thereby extending the range of constrained FL markets that admit strong incentive guarantees. Practically, the mechanism is implementable on modern FL platforms: it requires only a committed scoring rule, a fixed probe set, and standard optimization tooling for max-flow—no bespoke cryptography or heavy iterative equilibrium computation.

We also acknowledge limitations. Drift-score is not a universal measure of social value: it is tailored to a class of representation-alignment objectives and depends on the choice of probe set D_0. If D_0 poorly reflects the requester's evaluation distribution, the score can mis-rank workers, and no mechanism can recover the correct ordering without better information. Similarly, compatibility caps capture an important class of governance constraints, but not all externalities (e.g., privacy leakage risks or correlated failures) are reducible to group quotas. We view these as design inputs rather than bugs: the mechanism clarifies what must be specified—and audited—so that incentives and learning objectives align. In this sense, the model illuminates a tradeoff that practitioners already face: to run FL as a market, one must jointly choose (i) a contribution signal that is hard to game and (ii) an allocation/payment rule that respects budgets and constraints while remaining computationally viable.

The remainder of the paper situates DriftPay relative to prior work in budget-feasible mechanism design and FL incentives, formalizes the model and mechanism, and establishes truthfulness, feasibility, and approximation guarantees under both exact and noisy scoring.

2 2. Related Work: budget-feasible mechanism design; FL incentives (reverse auctions); incompatibility/compatibility constraints; MOON and representation drift; contribution measurement.

A large literature in mechanism design studies procurement under hard budget constraints, where a buyer seeks to purchase a subset of items (here, worker participations) subject to a spending cap. Classical truthful mechanisms such as VCG are generally incompatible with exogenous budgets: when payments are pinned to externalities, total transfers can exceed the available budget even if the chosen allocation itself is "affordable" in terms of reported costs. This tension motivated the budget-feasible mechanism design agenda, initiated in the context of crowdsourcing and experimental design and developed for a range of valuation classes, including additive, submodular, and knapsack-like objectives. Our work builds on this line in the specific regime where the platform's value is additive in per-worker quality signals (drift-scores), but feasibility is constrained by assignment structure rather than a single knapsack.

Within budget-feasible procurement, a particularly influential template is the family of ratio-based, threshold-payment mechanisms that (i) sort agents by a cost-effectiveness ratio and (ii) select a maximal affordable prefix, with payments determined by a critical ratio rather than by each agent's own report. Variants of this idea appear across problems with additive weights, sometimes described as "proportional share" or "critical bid" mechanisms, and have been sharpened into constant-factor approximation guarantees for welfare subject to budget feasibility and dominant-strategy incentive compatibility (DSIC). In our setting, the ratio takes the form b_i/v_i, which is economically natural: we are purchasing "units of algorithm-aligned contribution" rather than raw participation. The technical novelty is not the existence of a threshold rule per se, but that the allocation step must respect additional compatibility constraints and multi-requester assignment, which complicates the usual monotonicity and feasibility arguments.

A second relevant thread concerns budget-feasible mechanisms under combinatorial feasibility constraints (e.g., matroids, matchings, and their intersections). Compatibility constraints in procurement can be understood as restricting feasible sets beyond a simple cardinality bound, thereby introducing packing structure that is closer to bipartite b-matching or matroid intersection than to knapsack. Prior work has shown that if the feasibility system is downward-closed and admits polynomial-time optimization (or approximation) under weights, then one can often wrap it with a truthful, budget-feasible outer mechanism using critical thresholds. However, the constants and computational primitives depend sensitively on the constraint family. Our constraints are operationally motivated—caps by group and requester—and they admit an exact polynomial-time optimization primitive via a max-flow reduction. This places the problem in a tractable corner of constrained procurement: we can separate (a) the economic wrapper that enforces truthfulness and budgets from (b) the combinatorial subroutine that enforces compatibility.

A related empirical and policy literature motivates why such compatibility constraints arise in practice. In real deployments, "group caps" reflect governance (jurisdictions, business units), compliance (cross-border data handling), risk management (correlated failures), and fairness or representativeness objectives (preventing over-concentration on a single subpopulation). In online labor markets and ad auctions, similar constraints appear as quota controls, diversity constraints, or category caps; in public procurement they appear as vendor concentration limits. The common lesson is that ignoring these constraints at the mechanism level typically yields infeasible recommendations, and retrofitting feasibility via ad hoc post-processing tends to destroy incentive guarantees. Hence, the mechanism should internalize these caps in its allocation rule rather than treating them as an afterthought.

On the federated learning (FL) side, there is a fast-growing literature on incentives and participant selection. Early systems work focused on device availability and communication constraints (e.g., selecting clients to meet latency targets), largely treating clients as non-strategic. More recent work recognizes that participation is costly—compute, energy, opportunity cost, privacy risk—and models clients as strategic agents who require compensation. Proposed approaches include contracts, posted pricing, reputation schemes, and auctions. Reverse auctions are particularly natural when the platform is the buyer of compute/data contributions and clients are sellers: clients submit bids b_i, the platform selects a subset, and payments are set to induce truthful reporting. Many FL auction designs, however, either (i) use quality proxies tied to bids (undermining DSIC), (ii) neglect hard budgets (leading to overspending), or (iii) treat selection as a simple top-k rule (which fails under multi-tenant and constraint-rich environments). Our contribution is to connect the reverse-auction viewpoint to budget-feasible truthfulness guarantees, while using an FL-native quality signal.

A central obstacle is that "value" in FL is not directly observed at selection time. Unlike buying a commodity, the platform cannot a priori verify how helpful a client's update will be without running training, and even ex post evaluation is confounded by interaction effects among clients. This has led to an extensive literature on contribution measurement. Proposed metrics include loss decrease on a validation set, gradient norms, update similarity to the global gradient, influence-function approximations, and Shapley-value-inspired allocations. These methods highlight a key tradeoff: metrics that are more faithful to downstream generalization tend to be more expensive, noisier, and sometimes manipulable; metrics that are cheap and stable tend to be weakly connected to true learning value, especially under non-IID data. From a mechanism design perspective, the crucial property is not only predictive validity but also incentive compatibility: if the "quality" signal is itself a function of the bid or is easily manipulated after observing the mechanism, then cost truthfulness can fail even if payments are threshold-based.

Our approach is most closely related to "algorithm-aware" scoring rules that evaluate a standardized update relative to the current model state. In representation learning and contrastive objectives, the geometry of updates matters: two clients may have similar loss reductions but move representations in different directions, with different implications for transfer and generalization. This motivates scoring based on representation alignment, such as cosine similarity between embeddings, distance in a projected representation space, or regularizers that penalize divergence from a reference representation. The MOON family of methods formalizes this idea by encouraging local representations to stay close to global representations (and/or past local representations) through contrastive alignment losses. While MOON is typically presented as an optimization device, it implicitly defines a measurable notion of "representation drift" that can be estimated on a probe distribution. Our drift-score can be viewed as extracting this notion as an explicit currency for procurement: rather than paying for participation or for raw loss decrease, we pay for alignment with the algorithm's representational objective.

There is also an adjacent literature on drift detection and client selection in FL that uses public data, proxy tasks, or lightweight evaluation rounds to decide which clients to include. These methods often aim to stabilize training (e.g., exclude outlier updates) or to improve robustness (e.g., filter potentially poisoned or low-quality clients). Conceptually, we share with this literature the idea that a small reference dataset D_0 can be used to evaluate updates in a way that is comparable across clients. Economically, we differ in that we treat client cost as private information and explicitly design payments and allocations to satisfy DSIC, individual rationality, and budget feasibility. In particular, "filtering" rules alone do not specify how to pay selected clients, and naive payment rules can reintroduce incentives to misreport costs or to strategically alter behavior around the evaluation protocol.

Finally, privacy and observability constraints interact tightly with contribution measurement. Secure aggregation, differential privacy, and system-level constraints may limit what the platform can observe about local updates, making rich ex post evaluation infeasible. Conversely, any scoring protocol that requires extensive per-client inspection may be unacceptable in regulated settings. This motivates a minimal, ex ante committed scoring interface: clients submit a standardized object (e.g., an update under a fixed local routine), the platform evaluates it on a fixed probe set (potentially within a privacy-preserving pipeline), and the resulting scalar score is used for procurement. This design keeps the strategic interface one-dimensional in costs while allowing the scoring rule to remain aligned with the learning objective.

These literatures jointly motivate our modeling choices in the next section. We formalize the multi-requester procurement environment, the compatibility structure, and—critically—the MOON-style drift scoring protocol on a probe set, clarifying what is observable to the platform under realistic privacy and secure aggregation constraints.

3 System and Scoring Model

We study a procurement layer that sits "above" a standard federated learning training loop. The platform's role is to decide which clients participate and how much they are paid, subject to hard spending limits and operational compatibility restrictions. The key modeling move is to separate (i) a *bid-independent* measurement of each worker's algorithm-aligned contribution from (ii) a *strategic* cost report that the mechanism must elicit truthfully. This keeps the strategic type one-dimensional (cost) while allowing the platform's objective to reflect learning-relevant quality.

Agents, tasks, and budgets. There is a set of workers (clients) S with $|S| = n$, indexed by i. Each worker incurs a true private participation cost $c_i \geq 0$ that captures compute/energy, opportunity cost, and any disutility from engaging in the protocol. Workers submit bids b_i as cost reports. There is a set of requesters (model owners) A, $|A| = m$, indexed by j. Requester j brings a budget B_j. Depending on the institutional setting, budgets can be *separate* (each requester pays its own assigned workers) or *pooled* (a joint budget $B := \sum_{j \in A} B_j$ funds the full allocation). The platform chooses an assignment $x = \{x_{ij}\}$, where $x_{ij} \in \{0, 1\}$ indicates whether worker i is assigned to requester j. We impose one-assignment, $\sum_{j \in A} x_{ij} \leq 1$, reflecting that a client's local update is typically tied to a single training task per round.

We model the platform's "value" from assigning worker i as an observable score $v_i > 0$ (defined below). In the baseline additive case, requester j's gross benefit from its assigned set $S_j = \{i : x_{ij} = 1\}$ is $F(S_j) = \sum_{i \in S_j} v_i$, so total platform value is $\sum_{j \in A} \sum_{i \in S} x_{ij} v_i$. This additive structure is a tractable proxy: it matches the operational reality that selection must be made *before* rich interaction effects are observed, while still allowing the objective to reflect algorithmic alignment rather than raw participation. A limitation is that real learning dynamics can be non-additive (substitutes/complements across clients); we return to this when discussing benchmarks and failure modes.

Compatibility groups and caps. To represent governance and operational constraints, we partition workers into L groups $G = \{G_1, \ldots, G_L\}$ (e.g., jurisdiction, device class, enterprise unit, risk domain). Compatibility constraints limit how many workers from a given group can be assigned to a given requester. Formally, for each requester j and group l, we have a cap $\tau_{lj} \in \mathbb{Z}_{\geq 0}$ and require

$$\sum_{i \in G_l} x_{ij} \leq \tau_{lj} \qquad \forall j \in A,\ \forall l \in [L].$$

These caps generalize familiar "quota" controls: they can encode compliance (e.g., cross-border processing limits), diversification (avoid over-reliance on a single population), or correlated-failure management (limit exposure to a shared vulnerability). Economically, these constraints are not mere post-processing: because they restrict the feasible set of allocations, they shape the marginal value of including a worker and therefore must be internalized by the allocation rule if we want incentive and budget guarantees to survive.

Scoring interface: MOON-style drift on a probe set. Selection is driven by an algorithm-aware score computed from a fixed probe/validation dataset D_0 and an announced scoring rule. Let w^t denote the global model at round t, and let $R_w(\cdot)$ be the representation mapping induced by w (e.g., an encoder or a projection head). For scoring, each worker produces a standardized local update (or local model) using a protocol fixed by the platform—e.g., a prescribed number of local steps, optimizer, and (optionally) regularization strength. Denote the resulting local model by w_i^t. The platform then evaluates how much the worker's update "drifts" in representation space relative to the global model on the probe distribution. Concretely, we can define a drift-score v_i as either a similarity or a negative distance, for instance

$$v_i := \mathbb{E}_{x \sim D_0}\big[\mathrm{sim}\big(R_{w_i^t}(x), R_{w^t}(x)\big)\big] \quad \text{or} \quad v_i := -\mathbb{E}_{x \sim D_0}\big[\|R_{w_i^t}(x) - R_{w^t}(x)\|_2\big],$$

with the convention that higher v_i is more desirable. This is "MOON-style" in the sense that it operationalizes the representation alignment objective that MOON-family methods impose during training, but uses it as an ex ante scalar currency for procurement. We assume boundedness $0 < v_{\min} \leq v_i \leq v_{\max}$, which can be enforced by score normalization or clipping. The boundedness assumption is not purely technical: it reflects that extreme scores are often artifacts of unstable evaluation, adversarial updates, or out-of-distribution probe points, all of which a deployed system would want to dampen.

Two aspects of this interface matter for incentives. First, the platform commits to the scoring rule g and the probe set D_0 (or at least to an auditable hashing of D_0) *before* bids are submitted. Second, v_i is computed from the standardized update and (w^t, D_0), and does not depend on b_i. This bid-independence is what allows us to treat c_i as the sole strategic dimension in the reverse auction: the mechanism can sort and threshold on b_i/v_i without opening a direct channel for quality manipulation via bid shading. Of course, workers may still try to manipulate the update itself; we interpret the scoring protocol as part of the platform's technical enforcement (and we discuss manipulation channels explicitly in the next section).

Observability, privacy, and secure aggregation. The scoring design is constrained by what the platform can observe in realistic FL deployments. In many systems, raw local data are never revealed, and even local updates may be hidden behind secure aggregation, trusted execution environments, or differential privacy. Our model accommodates these constraints by treating v_i as the *only* worker-specific statistic that must become visible to the platform for selection and payment, while the underlying update can remain encrypted or ephemeral. One implementation is: each worker computes the standardized update locally; an enclave (or secure scoring service) evaluates g on D_0 and returns a signed scalar v_i; the platform uses $\{v_i\}$ and bids $\{b_i\}$ to run the mechanism. This is attractive from a policy standpoint because it minimizes exposure: the platform need not inspect gradients, intermediate activations, or client data, and requesters need not learn per-client model deltas.

We nevertheless acknowledge two limitations. First, the existence of a probe set D_0 is a substantive assumption: it must be representative enough that representation alignment on D_0 correlates with downstream performance, yet non-sensitive enough to be shared or securely handled. Second, any scoring pipeline introduces measurement noise. We therefore allow the platform to observe $\hat{v}_i$ with multiplicative error $|\hat{v}_i - v_i| \leq \eta v_i$, capturing stochastic evaluation, privacy noise, or approximate computation. Importantly, this noise is still assumed independent of bids.

Timing and strategic interface. A round proceeds as follows. The platform first commits to D_0, the scoring rule g, and the procurement mechanism M. Workers then (optionally) submit the standardized object needed for scoring; the platform computes $v_i := g(i, w^t, D_0)$ (or $\hat{v}_i$). Next, workers submit bids b_i. Given (b, v) and public constraints (B_j, τ_{lj}), the platform chooses an allocation $x(b, v)$ and payments $p(b, v)$, with worker utility

$$u_i(b; M) = p_i(b, v) - x_i(b, v)\, c_i, \qquad x_i := \sum_{j \in A} x_{ij}.$$

Requesters are treated as non-strategic in the baseline: they post budgets and accept the platform's assignment so long as payments satisfy budget feasibility. This abstraction isolates the core procurement problem—eliciting private costs under budgets—while still capturing the multi-tenant structure via (B_j, τ_{lj}). Extensions with strategic requesters (e.g., budget misreporting or competition across tasks) are conceptually important but orthogonal to the DSIC-in-costs property we target here.

This system-and-scoring model sets up the central comparison we pursue next: what fails if one uses cheap reputation proxies, accuracy-based scoring, or post hoc feasibility fixes instead of an algorithm-aware, bid-independent drift-score coupled with an allocation rule that internalizes compatibility and budget constraints.

4 Baseline Benchmarks and Failure Modes

Before introducing our mechanism, we clarify what goes wrong with several natural procurement baselines that appear in deployed federated learning systems. The point is not that these heuristics are irrational—many are attractive precisely because they are simple, auditable, and easy to communicate to stakeholders. Rather, the lesson is that once we simultaneously require (i) hard budgets, (ii) compatibility caps, and (iii) strategic cost reporting, seemingly innocuous design choices can produce systematic inefficiency, instability, or incentive failures. These failure modes motivate why we use an algorithm-aware *drift-score* as the mechanism's currency and why we insist that scoring be bid-independent and feasibility-aware *during* allocation (not as an afterthought).

Benchmark 1: bid-only selection ("lowest cost wins") and its blind spots. A first baseline ignores learning relevance and simply selects the cheapest feasible set of workers (e.g., lowest bids subject to compatibility and a budget constraint). This is occasionally justified operationally when the platform believes "any data help." Under non-IID data, however, the marginal contribution of a worker to global progress is highly heterogeneous. If we select on cost alone, we can exhaust the budget on workers whose updates are redundant (e.g., many clients from the same distribution mode) while excluding a small number of more expensive but crucially diverse clients. In practice, this often looks like over-sampling the majority population because those clients are abundant and competitively priced.

Compatibility constraints exacerbate this. Suppose group caps τ_{lj} are binding for some groups (e.g., only a limited number of clients in a sensitive jurisdiction can be used per requester). A bid-only policy will fill slack groups with cheap clients even when the learning signal would be better obtained by spending more to recruit scarce clients in a constrained group.

The mechanism designer's problem is therefore not merely to minimize cost; it is to buy *value* under compatibility and budget feasibility, which is exactly what our drift-score formalizes.

Benchmark 2: reputation proxies and mismatch under non-IID. A more sophisticated baseline weights selection by a reputation proxy r_i: historical participation count, past "accuracy contribution," uptime, device class, or a rolling average of prior validation performance. Operationally, r_i is appealing because it is easy to compute and seems to reward reliability. A typical procurement rule then ranks workers by b_i/r_i (cost per unit reputation) and selects greedily subject to feasibility.

The central problem is that under non-IID data, reputation is often a poor predictor of *current-round* usefulness. Reputation is inherently backward-looking: it aggregates performance under earlier global models $w^{t'}$, different cohorts, and potentially different training hyperparameters. Yet the marginal value of a worker's update depends strongly on the current representation space and which other workers are selected. Concretely, we can have two workers with similar r_i but very different alignment with the current global model. In extreme cases, reputation is *anti*-correlated with contribution: a highly reliable client from an over-represented distribution mode can look "good" historically while adding little incremental information in later rounds, whereas a sporadic client from a minority mode can be pivotal precisely when it appears.

This mismatch worsens as heterogeneity increases. In the common Dirichlet non-IID model, smaller concentration β induces higher dispersion in client label distributions; empirically (and consistent with our comparative statics), dispersion in any algorithm-aligned utility signal increases as β decreases. A reputation proxy that does not track this dispersion effectively collapses these differences, leading to systematic under-purchase of "rare but valuable" clients. From a policy perspective, this is not just an efficiency loss: it can embed a governance failure, because clients representing underserved or regulated populations are precisely those that compatibility caps and auditing regimes seek to manage explicitly. A reputation-only procurement layer can therefore undermine compliance goals by pushing the system toward whichever clients are easiest to recruit repeatedly.

Benchmark 3: accuracy-based scoring and why it is unstable as a payment currency. Another tempting baseline is to score a worker by some notion of accuracy improvement. For instance, one may define

$$s_i := \mathrm{Acc}(w_i^t; D_0) - \mathrm{Acc}(w^t; D_0),$$

or its loss analogue on a validation set, and then rank by b_i/s_i. This approach sounds aligned with learning, but it is fragile for three reasons.

First, accuracy (or loss) on a fixed probe set can be high-variance at the per-client level, especially when local steps are few or when D_0 is small due to privacy and governance constraints. In that regime, s_i can change sign across rounds even for the same client, so selection becomes unstable and hard to audit.

Second, accuracy is a blunt instrument for multi-objective training. Many modern FL algorithms (including MOON-family approaches) explicitly care about representation alignment, not just immediate loss reduction. A local update can temporarily improve probe accuracy while pushing representations in a direction that reduces future transferability or increases client drift, especially under label skew. Thus, accuracy-based scoring can systematically overpay for short-run gains that are not robust.

Third, accuracy-based scores are particularly vulnerable to strategic manipulation when they become a payment currency. If the score is computed on D_0, a worker can overfit to the probe distribution—even without seeing D_0 directly—by using adaptive updates that exploit known evaluation pipelines or by engaging in model poisoning that inflates probe performance while harming generalization. The more tightly payments are tied to accuracy deltas, the stronger the incentive to exploit these channels. In contrast, a representation drift signal can be engineered to be less sensitive to such attacks (e.g., by restricting the scoring protocol, clipping, and using similarity-based metrics), though we do not claim it is fully manipulation-proof.

Post hoc feasibility fixes break monotonicity and can destroy truthfulness. A subtle but important failure arises when platforms implement feasibility as a repair step. A common pattern is: (i) rank by some score (bid-only, reputation, or accuracy), (ii) take a prefix until the budget is "about" exhausted, and (iii) if compatibility caps are violated, drop or swap some workers until the assignment becomes feasible. This is operationally convenient, but it generally makes allocation *non-monotone* in bids. A worker who slightly lowers b_i can change the repair path (which other workers are swapped out to satisfy τ_{lj}), potentially causing that worker to be excluded. Once monotonicity fails, threshold payments cease to be well-defined and DSIC arguments collapse. Put differently, "greedy then fix" is not just suboptimal; it can be fundamentally incompatible with dominant-strategy procurement when feasibility constraints are combinatorial.

This is why we insist on internalizing compatibility in the allocation computation itself (via an optimization routine such as max-flow), rather than treating it as an engineering constraint to be enforced after ranking.

Manipulation channels and the case for bid-independent scoring. Beyond the above benchmark-specific issues, we highlight two generic manipulation channels that the scoring interface must close.

(i) *Bid-dependent scoring.* If the score used for selection can be influenced by b_i (directly or indirectly), workers acquire a second strategic lever beyond costs, and classic single-parameter reverse-auction guarantees no longer apply. Even seemingly harmless designs—e.g., allowing workers to "pay for verification" or to choose the amount of computation used for scoring as a function of their bid—can correlate score and bid in a way that invalidates DSIC.

(ii) *Update gaming.* Even when scores are bid-independent, workers may try to manipulate the standardized update to inflate their score. This is unavoidable in any performance-linked procurement scheme; the relevant question is whether the platform can standardize and audit the scoring pipeline enough that manipulation is costly, detectable, or bounded. Our drift-score interface is designed to be compatible with such enforcement (fixed local steps, fixed optimizer, bounded scoring, and secure evaluation), and to avoid needing worker-reported statistics.

Why drift-score is the right "currency" for procurement. These failures motivate our design choice: a scalar v_i that is (a) computed by the platform (or a trusted scoring service) under a committed rule, (b) aligned with the training algorithm's representation objectives rather than a noisy endpoint metric, (c) bounded and therefore well-behaved under budgets, and (d) usable as a weight in a feasibility-aware allocation routine under group caps. Economically, v_i is a tradable unit of "learning-relevant quality" that allows us to apply cost-effectiveness logic (rank by b_i/v_i) without expanding the strategic type beyond cost.

With this motivation in place, we now turn to the mechanism itself. The next section shows how to combine a critical-price allocation rule with a max-flow feasibility oracle to obtain a polynomial-time, budget-feasible reverse auction—DriftPay—that remains DSIC in costs while explicitly respecting compatibility constraints.

5 The DriftPay Mechanism (Pooled Budgets)

We now describe our baseline mechanism in the pooled-budget setting, where the platform controls a single procurement budget $B := \sum_{j \in A} B_j$ and is free to assign selected workers across requesters subject to the compatibility caps τ_{lj}. Conceptually, DriftPay buys "units of learning-relevant value" measured by the drift-score v_i, and pays for them using a single critical price (a cost-per-score threshold) that is determined endogenously from the bids. The two technical ingredients are: (i) a *critical-ratio* allocation rule based on sorting by b_i/v_i, and (ii) a feasibility-aware optimization oracle that internalizes the caps $\sum_{i \in G_l} x_{ij} \le \tau_{lj}$ rather than repairing violations after the fact.

Inputs and bid-independent scoring. At the start of the round, the platform has already committed to a scoring protocol and computed drift-scores $v_i \in [v_{\min}, v_{\max}]$ for each worker $i \in S$ using the fixed probe set D_0 and the announced MOON-style rule $g(\cdot)$. Workers then submit bids b_i (cost reports). From the mechanism's perspective, each worker is a single-parameter agent with private cost c_i and a publicly known "size" v_i. We emphasize that $v_i > 0$ ensures the cost-effectiveness ratio b_i/v_i is well-defined and that boundedness of v_i will later allow a clean approximation bound.

Cost-effectiveness ranking. Given bids $b = (b_i)_{i\in S}$ and scores $v = (v_i)_{i\in S}$, we define each worker's bid-per-score ratio

$$\rho_i \;:=\; \frac{b_i}{v_i}.$$

We sort workers in nondecreasing order of ρ_i, breaking ties deterministically (e.g., by a fixed worker index) to avoid ambiguity. Let this order be $1, 2, \ldots, n$, and let $S_k := \{1, 2, \ldots, k\}$ denote the prefix of the k most cost-effective workers under ρ. Intuitively, if we were buying divisible value, we would purchase as much drift-score as possible from the lowest ρ_i workers first. Our main complication is that workers are indivisible and must be routed to requesters under the compatibility caps; DriftPay handles this by solving, for each candidate prefix, the best feasible assignment under those caps.

The feasibility-aware oracle (ORP). Fix a candidate set of workers S_k. Define the *Optimal Routing Problem* on S_k (ORP) as

$$M(S_k) \;:=\; \max_{x} \sum_{j\in A}\sum_{i\in S_k} x_{ij} v_i \quad \text{s.t.} \quad \sum_{j\in A} x_{ij} \le 1 \;\forall i, \quad \sum_{i\in G_l} x_{ij} \le \tau_{lj} \;\forall (l,j), \quad x_{ij} \in \{0,1\}.$$

Because v_i does not depend on the requester j, ORP is "choose as many high-v_i workers as possible" subject to the existence of a feasible routing to requesters given the group caps. We solve ORP in polynomial time via a max-flow reduction (formalized later), and we use both outputs: the optimal value $M(S_k)$ and an optimal integral assignment $x^{(k)}$ that attains it.

One convenient flow construction is layered by (j,l)-buckets. Create nodes: a source s; one node for each worker $i \in S_k$; one node for each requester-group pair (j,l); one node for each requester j; and a sink t. Add edges $s \to i$ of capacity 1. For each worker $i \in G_l$, add edges $i \to (j,l)$ of capacity 1 for all $j \in A$. Add edges $(j,l) \to j$ of capacity τ_{lj}, and edges $j \to t$ of capacity $+\infty$ (or an optional per-requester staffing cap if one exists operationally). Any integer s-t flow corresponds to a feasible assignment satisfying one-assignment and compatibility; maximizing $\sum_i v_i$ can be implemented by a standard transformation (e.g., min-cost max-flow with costs

$-v_i$ on worker-admission, or by selecting a maximum-weight feasible set in a bipartite b-matching formulation). The key point for the mechanism is that feasibility is computed *inside* the optimization, so we never need an ex post "drop-and-swap" repair that would destroy monotonicity.

Choosing the winning prefix via a critical affordability test. We next determine how far down the ρ-ranking we can go while keeping the procurement affordable under budget B. DriftPay uses the following affordability condition: for a prefix S_k, consider paying every unit of drift-score at the *marginal ratio* $\rho_k = b_k/v_k$. If we were to purchase $M(S_k)$ units of score at price ρ_k per unit, the total payment would be $\rho_k \cdot M(S_k)$. We therefore define $k^\star$ as the largest index k such that

$$\rho_k \cdot M(S_k) \;\leq\; B.$$

Operationally, we can find $k^\star$ by scanning $k = 1, 2, \dots$ and solving ORP on each prefix until the inequality fails (or by using a doubling/binary-search strategy with cached flow solutions when n is large). Denote the resulting selected prefix by $S_{k^\star}$, its ORP-optimal value by $M(S_{k^\star})$, and an associated assignment by $x^{(k^\star)}$. The mechanism's allocation rule is then:

$$x \;:=\; x^{(k^\star)} \quad \text{and} \quad x_i := \sum_{j \in A} x_{ij} \in \{0, 1\}.$$

That is, we allocate exactly the feasible set that maximizes total drift-score among workers whose bid-per-score ratio is at most the critical index $k^\star$.

Threshold payments (critical price per unit score). Given the chosen prefix and assignment, DriftPay pays each winning worker a score-proportional threshold payment based on a single per-unit "price" π. Let $\rho_{k^\star+1}$ denote the next ratio in the sorted list (with the convention $\rho_{n+1} = +\infty$ if $k^\star = n$). Define

$$\pi \;:=\; \min\Big\{\rho_{k^\star+1},\; \frac{B}{M(S_{k^\star})}\Big\}.$$

For each selected worker i (i.e., $x_i = 1$), we set

$$p_i \;:=\; v_i \cdot \pi,$$

and for each unselected worker we set $p_i := 0$. This payment rule has two complementary interpretations. The term $\rho_{k^\star+1}$ is the familiar "next-best" critical ratio: it is the smallest ratio that would displace a winner if they raised their bid enough to move behind the $(k^\star + 1)$-st worker. The term $B/M(S_{k^\star})$ is a budget-based cap: it is the maximum uniform price per unit score that can be paid to exactly finance the purchased total score $M(S_{k^\star})$.

Taking the minimum ensures that the per-unit price is simultaneously a competitive threshold (protecting incentives) and fiscally feasible (protecting the hard budget).

In implementation terms, p_i is simple to audit: the platform posts the critical per-unit price π and pays each winner proportionally to their publicly computed score v_i. The assignment across requesters is already embedded in x; under pooled budgets we can treat payments as drawn from the common pot, while still recording the realized routing x_{ij} to certify that each requester's caps τ_{lj} were respected.

Complexity and practical remarks. DriftPay's runtime is polynomial. Sorting by ρ_i takes $O(n \log n)$. Each ORP call is solvable in polynomial time via max-flow/min-cost flow on a network with $O(n + mL + m)$ nodes and $O(nm)$ edges (more precisely, $O(nm)$ worker-to-(j, l) edges, since each worker connects only to the (j, l) nodes consistent with its group G_l). A straightforward implementation that scans prefixes performs at most n oracle calls, yielding total time $O(n \cdot \mathrm{Flow}(n, m, L))$. In many regimes this is acceptable because m and L are modest compared to n, and because flow computations are highly optimized; nonetheless, we view the oracle as an explicit design choice: we pay a computational cost to avoid the economic cost of non-monotone, post hoc feasibility fixes. We also note a modeling limitation: pooled budgets abstract away from internal cost accounting across requesters. In applications where per-requester budgets B_j must be respected individually, we adapt the same logic with a slightly richer feasibility oracle; we treat that extension separately to keep the exposition clean.

6 Theoretical Guarantees (Pooled Budgets)

We now formalize what DriftPay delivers in the pooled-budget benchmark. Economically, the platform is running a reverse auction with a hard budget and nontrivial feasibility constraints (compatibility caps and one-assignment). The core design goal is to retain the classic procurement virtues—truthful cost revelation and fiscal discipline—while still approximating the best achievable learning-relevant value, measured here by total drift-score.

Theorem 6.1 (DSIC, ex post IR, and budget feasibility)**.** *Fix bid-independent scores $v_i \in [v_{\min}, v_{\max}]$ and pooled budget B. Under DriftPay-CO (i.e., sorting by $\rho_i = b_i/v_i$, selecting the maximal affordable prefix $k^\star$, allocating via the ORP optimum on $S_{k^\star}$, and paying winners $p_i = v_i\pi$ with $\pi = \min\{\rho_{k^\star+1}, B/M(S_{k^\star})\}$), the mechanism is: (i) dominant-strategy incentive compatible (DSIC) in costs, (ii) ex post individually rational (IR) for truthful bidders, and (iii) budget-feasible: $\sum_i p_i \leq B$ for every bid profile. Moreover, given an exact ORP oracle, the mechanism runs in polynomial time.*

Intuition. Two ingredients jointly drive Theorem 6.1. First, because v_i is fixed independently of b_i, each worker is a single-parameter agent (only cost is private), so the standard monotonicity+threshold-payment logic can apply. Second, DriftPay prices *score* rather than *headcount*: a single per-unit score price π is computed from the marginal competitor ($\rho_{k^\star+1}$) and the hard budget cap ($B/M(S_{k^\star})$). This makes the payment simultaneously "competitive" (no winner can demand more than what an excluded rival would justify) and "affordable" (total payments never exceed B).

Proof sketch (economic logic). We outline the main steps; full details follow the CARE-CO template adapted to our ORP feasibility structure.

Step 1: Monotonicity of the allocation in b_i. Fix b_{-i} and scores v. Consider worker i and let $\rho_i = b_i/v_i$. If i raises her bid, ρ_i weakly increases, so i moves (weakly) later in the global ρ-ordering. The maximal affordable prefix index $k^\star$ is defined by the inequality $\rho_k \cdot M(S_k) \le B$, and this affordability test becomes (weakly) harder to satisfy when one agent's ratio increases and all others are fixed. Thus, increasing b_i cannot cause the mechanism to include any worker whose ratio is *higher* than before; in particular, i cannot become newly selected by bidding higher. Conversely, lowering b_i only improves i's rank and can only expand the set of prefixes for which i is eligible; with deterministic tie-breaking in both the ρ-ordering and the ORP solution selection, the allocation rule is monotone in the standard single-parameter sense: once i is selected at some bid, she remains selected at any lower bid.[1]

Step 2: Threshold payments. Given monotonicity, each worker has a *critical* bid $\theta_i(b_{-i})$ such that i is selected iff $b_i \le \theta_i(b_{-i})$. DriftPay's payment is score-proportional at a single critical ratio π, so the implied critical bid is $\theta_i = v_i\pi$. The term $\rho_{k^\star+1}$ ensures competitiveness: if i were to bid above $v_i\rho_{k^\star+1}$, she would be (weakly) less cost-effective than the marginal excluded worker and would not survive the critical-prefix selection. The term $B/M(S_{k^\star})$ ensures fiscal feasibility: even if competitors are sparse, the platform cannot pay more than B in total, which pins down a maximal uniform per-score price. Taking π as the minimum of these two bounds makes $v_i\pi$ a valid critical payment.

Step 3: DSIC and ex post IR. In single-parameter procurement, monotone allocation plus threshold payments implies DSIC: truthful bidding maximizes $u_i = p_i - x_i c_i$ regardless of b_{-i}. Ex post IR follows from the fact that every winner is paid at least her reported cost: for any selected i,

$$\pi \;\ge\; \rho_{k^\star} \;\ge\; \rho_i \quad\Rightarrow\quad p_i = v_i\pi \;\ge\; v_i\rho_i = b_i.$$

Under truthful bidding ($b_i = c_i$), this gives $u_i = p_i - c_i \ge 0$. (If a worker strategically overbids above her true cost, she may forgo selection; if she

[1] This is exactly where we pay for an exact, feasibility-aware oracle: post hoc "repair" procedures can create non-monotone discontinuities and break DSIC.

underbids below cost, she risks negative utility, which DSIC rules out as an equilibrium incentive.)

Step 4: Budget feasibility. Let $W = \{i : x_i = 1\}$ be the winning set produced by the ORP assignment on $S_{k^\star}$. By construction, the achieved drift-score equals the ORP optimum on that prefix:

$$\sum_{i \in W} v_i \;=\; M(S_{k^\star}).$$

Therefore total payments are

$$\sum_i p_i \;=\; \sum_{i \in W} v_i \pi \;=\; \pi \cdot M(S_{k^\star}) \;\le\; \frac{B}{M(S_{k^\star})} \cdot M(S_{k^\star}) \;=\; B,$$

where the inequality uses $\pi \le B/M(S_{k^\star})$.

Approximation guarantee and a closed-form bound. Truthfulness and budgets matter only insofar as we still procure meaningful value. We therefore compare DriftPay's achieved total score

$$ALG \;:=\; \sum_{i \in W} v_i \;=\; M(S_{k^\star})$$

to the full-information optimum OPT under the same compatibility constraints and budget B.

Theorem 6.2 (Constant-factor approximation). *Assume* $0 < v_{\min} \le v_i \le v_{\max}$. *In the pooled-budget setting,*

$$ALG \;\ge\; \frac{1}{2 + v_{\max}/v_{\min}}\, OPT.$$

Proof sketch (why bounded scores are enough). Let $i^\dagger := k^\star + 1$ denote the marginal excluded index (with $\rho_{n+1} = +\infty$ by convention). Consider the optimal feasible solution under budget B and split its selected workers into: (a) those lying in the prefix $S_{k^\star}$, and (b) those outside it. Part (a) contributes at most $M(S_{k^\star}) = ALG$ by definition of the ORP optimum on $S_{k^\star}$.

For part (b), every worker outside $S_{k^\star}$ has ratio at least $\rho_{i^\dagger}$. With truthful bidding ($b_i = c_i$), cost equals $\rho_i v_i$, so any such worker costs at least $\rho_{i^\dagger}$ per unit score. Hence, under budget B, the *total* score of the outside part is at most $B/\rho_{i^\dagger}$.

We now relate $B/\rho_{i^\dagger}$ to ALG using the maximality of $k^\star$. Since $k^\star$ is maximal, the affordability test fails at $k^\star + 1$:

$$\rho_{i^\dagger} \cdot M(S_{k^\star+1}) \;>\; B.$$

Moreover, adding one additional worker to the candidate set can increase the optimal routable score by at most that worker's score, so

$$M(S_{k^\star+1}) \;\le\; M(S_{k^\star}) + v_{i^\dagger} \;=\; ALG + v_{i^\dagger}.$$

Combining yields

$$\frac{B}{\rho_{i^\dagger}} \;<\; ALG + v_{i^\dagger}.$$

Therefore,

$$OPT \;\le\; ALG + \frac{B}{\rho_{i^\dagger}} \;<\; ALG+(ALG+v_{i^\dagger}) \;=\; 2\,ALG+v_{i^\dagger} \;\le\; \Big(2+\frac{v_{\max}}{v_{\min}}\Big)\,ALG,$$

where the last step uses $v_{i^\dagger} \le v_{\max}$ and $ALG \ge v_{\min}$ (the mechanism selects at least one worker whenever any feasible procurement is possible). Rearranging gives Theorem 6.2.

Remark (measurement noise). If the platform sorts by $\rho_i = b_i/\hat{v}_i$ where $|\hat{v}_i - v_i| \le \eta v_i$, DSIC in costs is preserved because $\hat{v}_i$ remains bid-independent; what changes is welfare. A standard sandwich argument implies a multiplicative degradation on the order of $(1-\eta)/(1+\eta)$ relative to the noiseless ALG, so the constant-factor approximation degrades smoothly with score noise.

Limitations. The guarantees above rest on two commitments that are operationally meaningful: (i) bid-independent scoring (to keep the problem single-parameter), and (ii) feasibility computed *within* the optimization (to avoid non-monotone repairs). When either is relaxed—e.g., workers can manipulate v_i through the scoring update, or the platform uses ad hoc feasibility fixes—DSIC can fail even if the allocation appears "reasonable" from a machine-learning perspective.

7 Extensions

The pooled-budget benchmark isolates the worker-side incentive problem and lets us emphasize the role of bid-independent drift scoring. In deployments, however, two additional features are often salient: (i) budgets are held by multiple requesters who may not wish to pool funds, (ii) the learning-relevant value of a *set* of workers can be non-additive because workers' updates overlap in the representation space, and (iii) platforms sometimes want to add ex post performance bonuses. We briefly sketch how each can be incorporated, and where the economic logic becomes more delicate.

(a) Separate budgets and (mildly) non-cooperative requesters. Suppose each requester $j \in A$ has a hard budget B_j that cannot be reallocated across requesters, while compatibility caps τ_{lj} remain requester-specific. The platform's feasibility constraints then include

$$\sum_{i \in S} p_{ij} \leq B_j \qquad \forall j,$$

rather than the pooled constraint $\sum_i p_i \leq B$. From a mechanism-design perspective, the worker side remains single-parameter as long as scores v_i (or $\hat{v}_i$) are bid-independent; the new difficulty is that affordability is now *vector*-valued: an assignment can be feasible in aggregate but infeasible for an individual requester.

A natural adaptation of the CARE-NO/PEA-style idea is to retain a *single global ordering* by cost-effectiveness $\rho_i = b_i/v_i$, but to couple prefix selection with a per-requester affordability test. Concretely, for a candidate prefix S_k, we solve an ORP-like optimization that also respects *budget-implied score caps*:

$$\max_x \sum_{i \in S_k} \sum_{j \in A} x_{ij} v_i \quad \text{s.t.} \quad \sum_{i \in G_l} x_{ij} \leq \tau_{lj}, \; \sum_j x_{ij} \leq 1, \; \sum_{i \in S_k} x_{ij} v_i \leq \frac{B_j}{\pi} \; \forall j,$$

for a candidate per-score price π. Intuitively, if all workers assigned to requester j were paid $p_{ij} = v_i \pi$, then requester j can afford at most B_j/π total score. This turns the separate budgets into linear constraints in the flow formulation (one may interpret them as capacities on requester-to-sink edges measured in score units rather than headcount).

The remaining design choice is how to pick π and k without destroying monotonicity in each b_i. One robust (if conservative) approach is a two-sided "price escalation" heuristic: start from $\pi = \rho_{k+1}$ (competitive pressure from the marginal excluded ratio), and if some requester's implied spending $\pi \cdot \sum_i x_{ij} v_i$ would exceed B_j, increase that requester's shadow price until its score demand shrinks to feasibility, recomputing the ORP under the adjusted caps. This is reminiscent of PEA: we expand the feasible set by increasing prices on the *tight* budgets rather than by reallocating funds. Worker-side DSIC can still be preserved under deterministic tie-breaking provided (i) the final allocation remains monotone in each ρ_i, and (ii) each winning worker is paid a threshold of the form $p_{ij} = v_i \pi_j$ where π_j is the minimal per-score price at which requester j would still demand (and be assigned) that worker given others' bids.[2] What we gain is a mechanism compatible with institutional realities (each requester pays its own bill). What we lose, relative to pooling, is some efficiency: separate budgets can create unused slack (one

[2] We view this as an implementability constraint: if one instead computes an allocation first and then "clips" it to satisfy some B_j, the repair step is typically non-monotone and can reintroduce incentives to shade bids.

requester runs out of budget while another has remaining funds but cannot reallocate to the constrained side). In practice, this is precisely where policy choices enter (e.g., whether limited budget transfers are permissible within a consortium).

(b) Submodular drift value: diversity/coverage in representation space. The additive objective $\sum_{i,j} x_{ij} v_i$ treats each worker's drift contribution as independent. When drift is computed from representation alignment, this can be too linear: two workers whose updates move representations in the *same* direction may be partially redundant, while a set covering multiple "modes" of drift can be more valuable than its parts.

A simple way to model this is to define a monotone submodular set function over selected workers. For example, using the probe set D_0, cluster representations into $r \in [R]$ "regions," and let $z_{ir} \geq 0$ measure worker i's drift-improvement on region r under the announced scoring protocol. Then a coverage-style value

$$f(S') \;=\; \sum_{r=1}^{R} \max_{i \in S'} z_{ir}$$

is monotone submodular (diminishing returns). One can analogously define requester-specific $f_j(\cdot)$ if tasks differ mildly but share the same embedding space.

Mechanism-wise, truthful budget-feasible procurement for submodular objectives is known to be harder than for additive value, but constant-factor DSIC mechanisms exist in the single-parameter setting (notably Singer-style mechanisms) by combining (i) a greedy rule based on marginal value per cost and (ii) carefully defined threshold payments. In our setting, the additional compatibility constraints mean that "adding a worker" is only meaningful if the ORP can still route the set to requesters. Operationally, we can run a feasibility-aware greedy that, at each step, selects the worker with highest marginal gain per bid, *subject to* the existence of a compatible assignment (checked via the same max-flow oracle). Payments then follow the critical-bid construction induced by this monotone greedy (often requiring either randomization or a restricted greedy order to ensure monotonicity under general constraints).

The conceptual takeaway is that DriftPay's scoring layer is flexible: as long as we can compute a bid-independent marginal contribution $\Delta_i(\cdot)$ from (w^0, D_0), we can move beyond "sum of scores" toward "coverage of representational gaps." The limitation is that the clean single-price-per-score payment structure is generally lost; submodularity pushes us toward marginal-pricing payments, which are less transparent and may be less attractive operationally.

(c) Performance-adjusted bonuses (and why IC can fail if we are careless). Platforms often want to reward ex post training outcomes: e.g., pay a base procurement price and then add a bonus tied to validation improvement, gradient usefulness, or robustness metrics. Bonuses can be valuable in practice (they reduce the platform's risk from noisy ex ante scores and can motivate effort), but they interact subtly with DSIC.

The key economic point is that DSIC is a statement about incentives *at the bidding stage*. If we append an ex post bonus β_i to the payment of selected workers without adjusting the allocation/payment rule, then the effective "critical bid" changes. In particular, a worker with cost slightly above the original threshold may now find it profitable to underbid to get selected because the bonus makes participation worthwhile; truthful reporting would no longer be dominant. Thus, performance bonuses are DSIC-safe only when they are incorporated into the mechanism *ex ante* in a way that preserves the threshold structure.

Two conservative designs avoid this pitfall. First, a *fixed* bonus schedule that is publicly known and independent of bids and realized outcomes (e.g., a flat stipend for completing the standardized scoring update) can simply be folded into the affordability test and threshold payments. Second, an *outcome-contingent* bonus can be used only if it does not affect allocation incentives—e.g., if it is paid to *all* participants regardless of selection (then it is outside the mechanism), or if selection is determined using a bonus-adjusted scoring/payment rule that explicitly treats the bonus as part of the payment mapping and respects budget feasibility for worst-case bonus payouts.

In contrast, if bonuses depend on worker actions after bidding (effort, data curation, adaptive local training) or on strategic interaction (collusion, poisoning, free-riding), then the model is no longer single-parameter: workers have additional private or manipulable dimensions beyond cost. In that regime, our DSIC claim should not be expected to survive without stronger assumptions (verifiable effort, proper scoring rules, audits, or cryptographic attestations). Our recommendation is therefore pragmatic: use bonuses as an engineering tool, but treat them as a separate contracting layer whose incentive properties must be audited independently, rather than as a "free" add-on to a truthful reverse auction.

(d) Robustness to noisy drift estimates and score governance. A practical platform rarely observes the "true" drift score v_i without error. Even if the scoring rule $g(\cdot)$ is fixed and bid-independent, the realized estimate $\hat{v}_i$ can be noisy because (i) D_0 is finite and induces sampling error, (ii) local updates are stochastic (minibatching, dropout, data augmentation), and (iii) the server model w^t evolves, so the same worker may induce different representation shifts across rounds. We therefore separate two questions:

(1) when does dominant-strategy truthfulness in costs survive the use of $\hat{v}_i$? and (2) how sensitive is welfare (total drift-score) to such score noise?

DSIC conditions under noisy scores. The worker side remains single-parameter as long as the mapping from bids to allocation/payments depends on b_i only through a monotone statistic (e.g., $\rho_i = b_i/\hat{v}_i$) and the score signal $\hat{v}_i$ is *bid-independent.* Formally, if the platform commits to a protocol that generates $\hat{v}_i$ as a function of $(w^0, D_0, \Delta_i, \xi)$—where Δ_i is a standardized local update and ξ is platform-controlled randomness—and this protocol is executed before bids are submitted (or at least independently of them), then for each fixed realization of $\hat{v}$, the procurement mechanism is simply the additive-value reverse auction run on weights $\hat{v}_i$. In that case, the usual monotonicity-plus-threshold-payment logic applies: if lowering b_i can only weakly improve i's rank by $\rho_i = b_i/\hat{v}_i$ and cannot affect others' ranks, then the set of bids at which i is selected is an interval, and paying the critical bid preserves DSIC in costs. In particular, the "noise" does not harm DSIC *per se*; it only changes the realized ordering and thresholds.

This observation also clarifies the main failure mode. DSIC can break if workers can strategically influence $\hat{v}_i$ after seeing (or anticipating) how it affects allocation and payments. Examples include (i) choosing a non-standard local training procedure that inflates measured drift on D_0 without improving true contribution, (ii) adversarially overfitting to a public D_0, or (iii) selectively withholding updates until the scoring window. These behaviors introduce an additional private (and manipulable) dimension—"score engineering"—that is not captured by the cost-only model. Our baseline DSIC claim should therefore be read as contingent on a *score-governance assumption*: the platform can enforce or verify the standardized scoring update and the evaluation pipeline well enough that $\hat{v}_i$ is not a decision variable of the worker at the bidding stage. When this is not realistic, one must explicitly model multi-dimensional incentives (cost plus effort/quality manipulation), and DSIC is no longer the right benchmark without stronger enforcement tools.

Welfare sensitivity bounds. Holding the governance assumption fixed, we can ask how far the mechanism's achieved welfare can fall when using $\hat{v}_i$ instead of v_i. A convenient and interpretable condition is multiplicative error:

$$|\hat{v}_i - v_i| \leq \eta v_i \qquad \text{for all } i, \text{ with } \eta \in [0, 1).$$

This bound implies that score estimates preserve relative magnitudes up to a factor of $(1 \pm \eta)$ and, crucially, that "cost-effectiveness" ratios are distorted but not arbitrarily:

$$\frac{b_i}{\hat{v}_i} \in \Big[\frac{1}{1+\eta} \cdot \frac{b_i}{v_i},\ \frac{1}{1-\eta} \cdot \frac{b_i}{v_i}\Big].$$

Since DriftPay-CO (and its compatibility-aware ORP step) effectively chooses a prefix in the ordering by b_i/v_i and then realizes the best compatible assignment within that prefix, the preceding inclusion yields a sandwich comparison: the prefix selected under $\hat{v}$ is "close" to a slightly smaller or larger prefix under v, up to the $(1 \pm \eta)$ multiplicative distortion. Translating this into welfare, one obtains a clean multiplicative degradation bound of the form

$$ALG(\hat{v}) \;\geq\; \frac{1-\eta}{1+\eta} \cdot ALG(v),$$

and hence, combining with the constant-factor approximation guarantee under true scores,

$$ALG(\hat{v}) \;\geq\; \frac{1-\eta}{1+\eta} \cdot \alpha \cdot OPT(v),$$

where $\alpha = \frac{1}{2+v_{\max}/v_{\min}}$ in the pooled-budget benchmark (and analogous constants apply under the same bounded-score regime). Economically, this says that if score noise is controlled in *relative* terms, then the mechanism remains robust: it may mis-rank some workers on the margin, but it does not systematically allocate budget to dramatically less cost-effective workers.

There is, however, an unavoidable caveat: multiplicative bounds are meaningful only when v_i is bounded away from zero. This is precisely why we maintain $v_i \geq v_{\min} > 0$ in the theory. In deployments, workers with near-zero measured drift are effectively indistinguishable under the ratio rule because small absolute score errors induce large ratio swings. Practically, one should treat "low-score" workers as a separate regime: either exclude them by design (a minimum quality screen), or use a different scoring rule with better signal-to-noise properties near zero.

Conservative variants under uncertainty. If the platform is risk-averse to overestimating scores, a simple modification is to run the mechanism on lower confidence bounds $\tilde{v}_i$ rather than point estimates. For example, if $\hat{v}_i$ is computed as an empirical average over $|D_0|$ probe points, concentration inequalities can produce $\tilde{v}_i = \hat{v}_i - \epsilon_i$ such that $\tilde{v}_i \leq v_i$ with high probability. Mechanism-wise, this is still DSIC (the scores remain bid-independent), but it is intentionally conservative: it reduces the chance that the platform "overpays per unit true drift" due to optimistic measurement. The cost is reduced utilization of the budget and potentially lower realized welfare in benign environments. This trade-off is often acceptable when budgets are hard and overruns are institutionally costly.

Audit and refresh of scores. Finally, robustness is as much a governance problem as an estimation problem. Even when $\hat{v}_i$ is computed honestly, two forces motivate periodic refresh: (i) *model drift*—the mapping $R_{w^t}(\cdot)$ evolves, so yesterday's representation shift is not today's; and (ii) *gaming pressure*—if D_0 and g are fixed and predictable, some workers may learn to optimize the score rather than genuine contribution.

A simple, DSIC-compatible approach is to commit to an audit/refresh schedule that is exogenous to bids: every T rounds (or with fixed probability each round), the platform re-scores a random subset of workers on a freshly sampled probe set $D_0^{(t)}$ drawn from a committed distribution, and updates $\hat{v}_i$ for subsequent auctions. Crucially, to preserve bid-independence and avoid endogenous manipulation, the platform should (i) commit publicly to the scoring rule and the sampling procedure (e.g., via a verifiable random seed), (ii) keep the realized probe instances secret until evaluation (to reduce overfitting incentives), and (iii) separate the scoring update from the training update used for the actual FL task, so that the "measurement channel" is standardized and auditable.

Audits also enable deterrence. If a worker's re-scored $\hat{v}_i$ is systematically inconsistent with prior submissions (beyond what sampling variation predicts), the platform can down-weight future scores, impose temporary exclusion, or require stronger attestations (secure execution, reproducibility logs). These interventions do not require changing the reverse-auction logic; they change the *measurement layer* that supplies bid-independent scores. In our view, this modularity is a feature: the mechanism's incentive guarantees are cleanest when the economic contract (bids → allocation/payment) is separated from the ML measurement pipeline (updates → scores), with the latter governed by standard tools from auditing and statistical quality control.

Taken together, the message is that noisy drift estimates are not fatal to truthful procurement. When score errors are bounded and bid-independent, we retain DSIC and only lose welfare proportionally to the noise level. The more delicate issues arise when scores become strategically manipulable; then robustness requires explicit score governance—refresh, audits, and enforcement—rather than purely mechanism-theoretic adjustments.

(e) Empirical plan: multi-tenant FL simulations to stress-test mechanism–score coupling. Our theory isolates the economic layer (bids, allocation, payments) from the measurement layer (updates → scores). An empirical evaluation should therefore do two things in parallel: (i) test whether using algorithm-aware drift scores v_i actually improves the downstream ML objective relative to plausible alternatives, and (ii) test whether the mechanism's predicted comparative statics (budgets, compatibility tightness, score noise, and congestion) appear in realistic multi-tenant federated learning regimes. We outline a simulation-based plan that is intentionally modular so that one can swap in different FL algorithms, scoring rules $g(\cdot)$, and market parameters without changing the experimental logic.

Environment and multi-tenant structure. We simulate a platform with a fixed worker population S and m requesters A. Each requester j runs a training job over a shared pool of workers, but with a requester-specific objective (e.g., different label sets, domains, or loss weights) so that competition for workers is meaningful. The platform executes a procurement round at each FL communication round t: workers submit bids b_i^t (generated from a cost model plus noise, with the option to introduce strategic deviations for robustness checks), the platform computes drift-based scores v_i^t from a standardized scoring update, and then runs DriftPay-CO (pooled budget) or its per-requester extension under $\{B_j\}$, respecting compatibility constraints $\sum_{i \in G_l} x_{ij} \le \tau_{lj}$ and one-assignment $\sum_j x_{ij} \le 1$. After allocation, each requester performs one FL round with its assigned workers and logs test performance.

Non-IID control via Dirichlet heterogeneity. To systematically vary the severity of worker heterogeneity, we partition data across workers using a Dirichlet model with concentration parameter β, as is standard in FL benchmarking. Concretely, for a K-class task, worker i's class proportions are drawn as $\pi_i \sim \mathrm{Dirichlet}(\beta\mathbf{1})$, and we sample local datasets accordingly. Smaller β induces stronger non-IIDness and, in our interpretation, larger and more variable representation drift. We treat β as a primary axis because it directly operationalizes the intuition behind using drift-aware scores: when client data are more heterogeneous, the informational content of "reputation" or naive accuracy proxies becomes less reliable, and algorithm-aware measures of contribution should matter more.

Scoring rules and probe design. We compare multiple scoring rules $g(\cdot)$ that fit within our bid-independent framework. The main specification uses a MOON-style representation alignment score computed on a fixed probe set D_0:

$$v_i^t \;=\; \mathbb{E}_{x \sim D_0}\big[\mathrm{sim}(R_{w_i^t}(x), R_{w^t}(x))\big],$$

or an equivalent negative drift norm. We then vary the probe set size $|D_0|$ to control measurement variance, and we generate noisy estimates $\hat{v}_i^t$ by either subsampling D_0 or injecting multiplicative perturbations consistent with $|\hat{v}_i - v_i| \le \eta v_i$. This lets us connect observed welfare degradation to the theory's $(1-\eta)/(1+\eta)$ benchmark, while also highlighting when violations arise (e.g., scores near zero, distribution shift in D_0, or heavy-tailed noise).

Baselines: what we must beat empirically. The relevant benchmarks are not only "optimal" but also what a platform might plausibly deploy today.

- **CARE with reputation weights.** Replace drift scores v_i by a bid-independent reputation proxy r_i (e.g., exponentially weighted moving average of past participation, or historical test improvement when selected). Allocation and payments follow CARE-CO logic but with r_i in place of v_i. This isolates the value of algorithm-aware scoring from the value of the mechanism itself.

- **Accuracy-based scoring.** Replace v_i by an accuracy (or loss) proxy on D_0, such as $a_i^t = -\mathcal{L}(w_i^t; D_0)$ or $\Delta a_i^t = a(w_i^t; D_0) - a(w^t; D_0)$. This is a natural baseline because it is easy to explain to stakeholders, but it may be misaligned with representation learning dynamics, especially under non-IID data and multi-round training.

- **Bid-only / random selection under budgets.** Use a lowest-bid heuristic subject to compatibility, or random feasible assignment with budget feasibility. These baselines quantify how much of the gain is driven by any quality signal at all.

Where feasible, we also compute an *offline* upper bound: given realized (c_i, v_i) for a round, solve the full-information assignment under budgets and compatibility to approximate OPT for comparison.

Outcomes and evaluation metrics. We measure outcomes at both the mechanism level and the ML level.

1. **Economic welfare proxy:** realized total drift score $\sum_{i,j} x_{ij} v_i$ (or $\sum x_{ij} \hat{v}_i$ when only noisy scores are observed). This directly targets the mechanism objective.

2. **Downstream learning:** requester-specific test loss improvement $\Delta\mathcal{L}_j$ (or accuracy gain) over rounds, and the aggregate $\sum_j \Delta\mathcal{L}_j$. This tests whether the drift proxy is predictive in practice, complementing the assumption in Proposition 6.

3. **Budget utilization and prices:** total payments $\sum_i p_i$ (pooled) or $\sum_i p_{ij}$ (per requester), critical ratios, and the fraction of budget left unused due to compatibility or conservative scoring.

4. **Constraint stress:** frequency with which caps τ_{lj} bind, and concentration of assignments across groups G_l (useful for interpreting "congestion" effects and diversification pressure).

5. **Stability:** overlap of selected sets across adjacent rounds, and sensitivity of allocations to small perturbations in $\hat{v}$ or bids.

Comparative statics and congestion experiments. We explicitly vary the parameters highlighted by the theory.

- **Budgets.** Sweep pooled B (or $\{B_j\}$) and verify that selected welfare and critical prices weakly increase. We also compare pooled versus separated budgets to quantify the value of cross-requester budget sharing in congested markets.

- **Compatibility tightness.** Sweep τ_{lj} from restrictive (forcing heavy diversification) to permissive (near unconstrained) to test whether drift-aware scoring provides larger gains when the feasible set is tight (where good ranking is most valuable) or when it is slack (where quantity dominates).

- **Market congestion.** Vary the ratio m/n and introduce requester heterogeneity in budgets (few large buyers vs many small buyers). Congestion should amplify the importance of both accurate scores and compatibility-aware assignment (the ORP/max-flow step), because mistakes in ranking propagate through scarce capacity.

- **Score dispersion.** Induce dispersion by changing model architecture, local epoch counts, or heterogeneity β, and relate realized performance to $v_{\max}/v_{\min}$ as a diagnostic for when constant-factor bounds become loose.

Truthfulness stress tests (behavioral, not equilibrium). While we do not claim workers play equilibrium strategies in simulations, we can still perform falsification-style checks consistent with DSIC: fix others' bids and scores, then vary a single worker's bid b_i and empirically verify monotonicity of selection and threshold-like payment behavior. These tests help catch implementation bugs (e.g., non-monotone tie-breaking) and clarify how numerical issues in the ORP solver might translate into incentive violations in practice.

Limitations and what the empirical plan cannot certify. Two limitations are intrinsic. First, the strongest strategic failure mode is *score manipulation* (changing $\hat{v}_i$), which is governance-dependent; simulations can illustrate vulnerability but cannot replace enforcement. Second, mapping drift scores to downstream accuracy is algorithm- and task-dependent; empirical success will be strongest evidence in favor of the drift-to-generalization link, but not a proof. For that reason, we treat the empirical plan as a disciplined exploration of when the mechanism–algorithm co-design is most valuable (high heterogeneity, tight compatibility, moderate noise) and when simpler procurement heuristics suffice (low heterogeneity, slack constraints, or very noisy scoring).

Conclusion: mechanism–algorithm co-design as a 2026 agenda. We have argued for a simple organizing principle: in multi-tenant federated learning markets, the procurement layer and the learning layer should be designed together rather than bolted onto one another. The economic problem is not merely to buy "participation," but to buy *useful updates* under tight budgets, congestion, and compatibility constraints that reflect operational realities (conflicts of interest, regulatory separation, geography, device classes, or redundancy limits). The algorithmic problem is not merely to optimize a loss, but to produce an *auditable, bid-independent* contribution signal that can enter a reverse auction without breaking incentive guarantees. Our structured context makes this complementarity concrete: we treat representation drift (or alignment) as a score v_i that is computed from a fixed probe set and an announced scoring rule, and we then use a budget-feasible, compatibility-aware procurement mechanism that is truthful in the one-dimensional private type (cost). The model illuminates the tradeoff: we gain tractable incentives and approximation guarantees by insisting that quality signals be exogenous to bids, but we also inherit a governance burden—the integrity of the scoring pipeline becomes a first-order design object.

From a practice perspective, this co-design view reframes what a "marketplace for FL" should standardize. A 2026-ready platform cannot only specify an API for training; it must specify (i) a scoring protocol (what update is submitted for scoring, under what constraints), (ii) a probe-and-audit regime for D_0 (who curates it, how it evolves, how leakage is prevented), and (iii) a mechanism contract (allocation and payments) that is robust to the platform's measurement error and to the operational constraints imposed by requesters. Once these are explicit, familiar questions become measurable: how much budget is wasted because τ_{lj} bind; whether pooling $B = \sum_j B_j$ creates allocative gains or simply raises threshold prices; and how sensitive outcomes are to the score noise level η. In other words, the platform can move from ad hoc heuristics (lowest bid, static reputation, or opaque "quality scores") to a disciplined procurement policy whose failure modes are legible.

At the same time, we should be candid about what our guarantees do *not* cover. The mechanism is DSIC with respect to costs because the allocation rule is monotone in bids given fixed scores, but the most strategically important dimension in FL settings is often *not* cost—it is the ability to influence measured contribution. Our baseline model treats v_i (or $\hat{v}_i$) as bid-independent and produced under a standardized protocol, which is a meaningful design constraint but not a theorem of nature. If workers can manipulate the scored update, overfit to D_0, or induce representation changes that look beneficial on the probe while harming downstream objectives, then the marketplace risks devolving into a Goodhart's-law regime. In that world, mechanism design alone is insufficient: we need enforcement (attestation, rate limits, anomaly detection), robustness (multiple probes, randomized scoring, or holdout rotation), and possibly explicit penalties for detected manipulation. Economically, the right interpretation is that our DSIC result is a guarantee about *one channel* of misreporting (costs), conditional on institutional choices that close off other channels.

A second limitation is objective misspecification. We maximize additive total score $\sum_{i,j} x_{ij} v_i$ subject to budgets and caps, which is the correct objective only insofar as (i) the scoring rule is aligned with the algorithm's true marginal value of participation and (ii) additivity is an adequate approximation. In real multi-round FL, interactions are ubiquitous: a worker's value may depend on who else is selected, on the requester's current model state, and on diversity considerations that are not well captured by group caps alone. These complementarities do not invalidate the approach, but they point to the next modeling step: richer valuation classes where $F(S_j)$ may be submodular, state-dependent, or explicitly diversity-regularized. Mechanism-theoretically, that pushes us toward different approximation techniques, and algorithmically it pushes scoring toward *marginal contribution estimates* rather than stand-alone scores. The open question for the agenda is where the tractability frontier lies: how much complementarity can we admit while preserving polynomial-time allocation, budget feasibility, and interpretable threshold payments?

Third, our treatment of compatibility is intentionally stylized. Group caps τ_{lj} provide a clean bridge to max-flow allocation, but in practice compatibility constraints can be endogenous and contested. Requesters may demand exclusivity; regulators may require separation across sensitive attributes; platforms may wish to impose anti-collusion or anti-sybil limits. These constraints are not merely feasibility constraints; they have distributional consequences. A policy-relevant next step is to treat τ_{lj} (and even the partition G) as governance choices, and to evaluate them with explicit fairness or market-power objectives. For example, one may wish to bound concentration (no requester receives too many high-score workers), to guarantee minimum access for small-budget requesters, or to prevent systematic under-selection of certain worker groups. Incorporating such constraints will generally reduce measured welfare, and the appropriate question becomes: what is the welfare cost of a given governance rule, and can we quantify it ex ante?

The dynamic nature of FL procurement is a further frontier. In deployments, auctions repeat over rounds t, budgets replenish, and both costs and scores evolve with participation and learning. This creates two feedback loops: selection affects future scores (through model state and worker incentives), and payments affect future bidding (through participation and outside options). Our static mechanism can be run round-by-round, but doing so ignores intertemporal incentive issues, such as bid shading to influence future thresholds or strategic waiting to be selected later when prices rise. A natural next step is a repeated-game or dynamic mechanism layer that commits to a policy over time (e.g., budget pacing, reserve prices, or participation guarantees) while preserving the core monotonicity property needed for cost truthfulness. A complementary empirical step is to measure how rapidly v_i^t drifts with t and whether the ranking stability is high enough that simple round-wise procurement is already near-optimal.

Finally, we see a concrete engineering agenda that connects theory to implementation. The most immediate deliverables are: (i) open benchmark suites for multi-tenant FL procurement that report not just accuracy, but also budget utilization, threshold prices, and constraint binding; (ii) standardized scoring interfaces that make bid-independence auditable; (iii) secure execution or attestation pathways for the scored update to reduce manipulation; and (iv) diagnostic tooling that reports $v_{\min}$, $v_{\max}$, and effective η so that approximation guarantees are not purely symbolic. More ambitious deliverables include privacy-preserving scoring (so that D_0 can be sensitive), collusion-resistant procurement (when workers coordinate bids), and mechanisms that incorporate requester heterogeneity beyond budgets (risk, latency, or compliance). We view these as design choices that can be evaluated with the same discipline as learning algorithms: specify assumptions, measure violations, and quantify welfare loss.

The broad lesson is pragmatic. A marketplace for federated learning will be judged not only by test accuracy, but by whether it can *reliably* translate money into learning progress under real constraints and strategic behavior. Mechanism–algorithm co-design offers a path to that reliability: it forces the platform to declare what it measures, why it is aligned with learning, and how it will pay for it. Our model does not claim to settle the problem; it clarifies where guarantees are available today, where they hinge on governance, and where the next round of work should focus if we want FL procurement to be more than a heuristic—namely, an institution that is both computationally implementable and economically credible.