

Federated Learning for Enterprise Information Systems: Security, Privacy, Architecture, and Deployment Challenges—A Review

Li Fei*

*School of Artificial Intelligence, Neijiang Normal University, No. 1 Hongqiao Street, Dongxing District, Neijiang, Sichuan, 641100, China
10001310@njtc.edu.cn
Corresponding author

Keywords: Federated Learning; Enterprise Information Systems; Information Systems Security; Privacy-Enhancing Technologies; Cross-Silo Learning

Abstract: Federated learning enables organizations to develop shared models while retaining local control of training records. Exchanged updates can nevertheless reveal sensitive information, making privacy protection a central system-design requirement. This narrative review examines cross-silo collaboration from the choice of data architecture to the operation of a shared prediction service. It compares differential privacy, secure aggregation, secure multi-party computation, homomorphic encryption, and trusted execution environments, explaining their protection boundaries and the costs of putting them into practice. Evidence from communication-efficient learning and encrypted-training systems makes the computational trade-offs concrete. Healthcare, credit-risk, and energy studies supply examples of cross-institutional prediction and its operating requirements. The synthesis distinguishes confidentiality during computation from disclosure through released outputs, and examines the tension between hidden updates and integrity checks. A seven-stage technical deployment framework turns these findings into a workflow for selecting protections, validating local performance, and maintaining the federation as its participants and services change. The review provides a basis for choosing complementary controls and planning deployment around both the learning task and its operational requirements.

1. Introduction

Enterprise data often span several organizations: financial institutions hold complementary customer histories, hospitals observe different patient populations, and industrial sites record distinct operating conditions. Confidentiality and contracts limit the exchange of raw records, so federated learning (FL) trains locally and exchanges learning signals instead; Federated Averaging (FedAvg), evaluated on five model architectures and four datasets, reduced communication rounds 10–100-fold relative to synchronized stochastic gradient descent [1].

Differences in data and computing resources, studied through participation patterns, data

partitioning, and statistical and systems heterogeneity [2–4], become part of the shared system in cross-organizational training; the resulting model needs an operating arrangement for serving predictions and recovering from failures. Model updates are also not inherently anonymous: gradient-reconstruction studies recover training records under specific conditions [5, 6], other attacks infer membership and dataset properties [7, 8], and reviews survey attacks and defenses [9–11]. Organizational research addresses orchestration and ownership [12]. Privacy protection is therefore part of the architecture rather than a consequence of local training alone.

This narrative review makes two contributions for cross-silo enterprise collaboration: a comparison of privacy-enhancing mechanisms by protection scope, trust assumptions, and system costs; and a seven-stage technical deployment framework that carries these choices through validation and operation. Sources were chosen for enterprise relevance; formal guarantees, attack demonstrations, and application results are treated as distinct forms of evidence.

2. System Architecture for Cross-Silo Enterprise Federated Learning

Cross-device FL commonly involves many intermittently available devices, whereas cross-silo FL involves fewer organizations with more persistent identities [2, 4]. Because one site’s failure can interrupt such a collaboration, cross-silo designs emphasize authentication, local integration, and recovery. Horizontal FL uses different records with comparable features, while vertical FL uses complementary features for overlapping entities and often requires private entity alignment; federated transfer and hybrid approaches address further mismatches [3]. Table 1 summarizes representative settings; its final row is illustrative rather than drawn from an application study.

Before horizontal training, participants agree on the target and preprocessing conventions, including units and missing-value treatment, because the same field name may denote labels produced by different local practices. In vertical collaboration, such as between a bank and a retailer with complementary features for overlapping customers [3], private matching identifies the records used for joint training. If a prediction needs features from both organizations, the service depends on their availability and access permissions after training has finished, so the deployment agreement must cover output recipients, service responsibilities, and participant withdrawal.

Table 1: Representative enterprise FL configurations and system-design requirements.

Setting	Data arrangement	System-design requirements
Healthcare studies [13–15]	Horizontal across institutions	Shared preprocessing and labels; local validation; patient-signal protection
Credit risk study [16]	Across different portfolios	Comparable targets and holdouts; calibration; portfolio confidentiality
Load forecasting experiments [17]	Distributed time-series forecasting	Privacy unit; error and communication measures; dropout tolerance
Supply chains, business units	Horizontal, vertical, or hybrid [3]	Entity alignment; compatible interfaces; inference and exit arrangements

3. Threat Model and Security Risks in Enterprise Federated Learning

3.1. Inference Risks from Exchanged Signals

Information can escape through training signals (gradients, parameter changes, or activations), prediction outputs, supporting services, and auxiliary records such as identifiers, diagnostics, and validation summaries. Gradient-inversion studies recover training information under particular conditions [5, 6]; batch size, local computation, model, and loss can change what an update reveals, and auxiliary knowledge or control over the interaction can strengthen an attacker. Membership and

model-inversion studies examine exposure through learned models [18, 19], and collaborative and federated studies probe update access and active interaction [7–9]. Customer composition or operating characteristics may be commercially sensitive even if no record is recovered. A threat assessment starts from concrete attacker capabilities—a server reading individual updates, a user querying the released model, or a malicious server changing the training code. The label “federated” guarantees neither privacy nor data-localization compliance.

3.2. Integrity Attacks and the Confidentiality–Inspection Trade-Off

An authenticated participant can still submit a poisoned update; Bagdasaryan et al. demonstrate a backdoor attack through a malicious participant [20]. Data poisoning alters local examples, whereas model poisoning alters the submitted update directly; a backdoored model makes targeted errors while otherwise behaving normally. Authentication shows who supplied a contribution, not its effect. Because secure aggregation hides individual contributions, including malicious ones, from the orchestrator [20, 21], robustness methods that need plaintext updates require a compatible validation path. The proposed framework therefore pairs approved software and local validation with aggregate checks, poisoning tests, and rollback or quarantine procedures.

3.3. Orchestration and Trust Boundaries

An orchestrator can steer learning without reading individual updates: it may select participants, alter the model, reschedule aggregation, or control releases and logs. Governance research treats orchestration and ownership as design choices [12]; separating infrastructure operation from authority over model purpose and access limits each operator’s power.

4. Privacy-Enhancing Technologies and System Trade-Offs

A system may hide inputs during computation yet leak information through its outputs; cryptography and differential privacy address these risks respectively, as hybrid privacy-preserving FL research illustrates [22]. Table 2 compares the main mechanisms.

4.1. Differential Privacy: Protection Units and Utility

Differential privacy (DP) bounds changes in an output distribution between neighboring datasets [23]. The neighboring relation defines the protected unit—one record, one person’s records, or an entire client’s data—so patient-level and hospital-level protection differ. Clipping and calibrated noise implement these objectives in private and federated learning [24–26], and the guarantee is meaningful only when the protected unit, privacy parameters, and released outputs are explicit. Under central DP, the party that sees unprotected contributions remains trusted. Repeated training, tuning, and releases consume a cumulative privacy budget tracked by an accountant [23, 27]. Protecting an entire organization can substantially reduce utility in a small federation, and the trade-off varies with data volume, heterogeneity, and optimization and privacy settings [23–26]; participant-level results show whether a configuration leaves a small site with an unusable model.

4.2. Secure Aggregation and Secure Multi-Party Computation

Secure aggregation releases a permitted aggregate while concealing individual inputs; Bonawitz et al. provide a practical construction with dropout handling [21]. The recipient can still read the sum, and colluders who know all but one input can recover the remaining one, so protection de-

depends on participation thresholds and collusion assumptions. Secure multi-party computation (MPC) supports broader joint operations, as in two-server SecureML [28]; protocols differ in adversary model and communication cost, and integration must handle aborted computations. Secure aggregation can itself use MPC or homomorphic encryption, so Table 2’s categories overlap.

4.3. Homomorphic Encryption and Trusted Execution

Homomorphic encryption (HE) supports computation on ciphertexts [29] and can aggregate encrypted updates in horizontal FL. BatchCrypt encrypts batches of quantized gradients packed into long integers; against stock FATE on geo-distributed Amazon EC2 nodes, it reported 23–93-fold iteration speedups and 66–101-fold reductions in communication per iteration [30]. Numerical encoding can affect accuracy, and key custody, decryption authority, rotation, and recovery must be arranged. Trusted execution environments (TEEs) isolate computation within an attested hardware boundary, as in PPFL [31]. Side-channel coverage follows the platform’s threat model; memory limits constrain workloads, while outputs and external services require separate controls.

Table 2: Protection scope, assumptions, and deployment trade-offs.

Mechanism	Protection scope and assumptions	Utility, overhead, and integration
Differential privacy [23–27]	Bounds output-distribution change; trust depends on noise placement	Clipping and noise affect utility; specify unit, parameters, accountant
Secure aggregation [20, 21]	Hides inputs but reveals the aggregate; depends on collusion limits	No added noise; protocol traffic, dropout handling, validation limits
Secure MPC [28]	Limits disclosure in joint computation under a collusion model	Interactive communication; aborted runs affect availability
Homomorphic encryption [29, 30]	Protects inputs during supported computation; key holders see outputs	Ciphertext overhead; encoding affects accuracy; key custody and recovery
TEE-based protection [31]	Attested hardware isolation; coverage follows the platform threat model	No added noise; hardware, attestation, patching, fallback

4.4. Selecting Complementary Protections

Protection selection starts by identifying the observer and the information to protect. Secure aggregation may suffice to hide individual updates from the server, whereas an output-level guarantee such as DP limits what a released model reveals; HE, MPC, or a TEE can protect additional computations. These options are evaluated under a common threat model, not chosen as interchangeable privacy labels; integrity checks and access control are designed alongside them within utility and resource targets.

5. Operational Security, Governance, and Accountability

Where the EU General Data Protection Regulation (GDPR) applies, controllers determine the purposes and means of processing, while processors act on their behalf [32]. A data-flow map helps identify these roles for activities within the federation, including exchanges or outputs about identifiable people; lawful purpose, data minimization, and safeguards remain part of the operating design. The NIST Privacy Framework and AI Risk Management Framework add complementary perspectives [33, 34]. In FL, disclosure protection is evaluated alongside model reliability and fairness, each with an assigned owner. Legal analysis depends on the applicable jurisdictions and intended uses, including cross-border processing.

Admitting a participant or releasing a model is also a governance decision. Barbereau et al. examine FL governance through data markets, infrastructure, and ownership [12]; in practice,

authority over software changes, membership, model access, logs, and the suspension of training must be allocated deliberately. Incident investigation requires records of code and model versions, participants, aggregation settings, protection and validation results, and releases. Credential revocation stops future participation but does not remove earlier contributions from a trained model, so retraining or an evaluated removal procedure may be needed. Notice periods help the remaining members replace a departing feature source or inference service.

6. Deployment and Systems Challenges

A shared model can converge while remaining poorly suited to a small participant or a rare but consequential event. FedProx adds a proximal term to local objectives [35], SCAFFOLD uses control variates to reduce client drift [36], and adaptive federated optimization applies adaptive optimizers at the server [37], but local usefulness remains an empirical question. A well-tuned local model is each institution's baseline, and in fraud or risk tasks, calibration at the operating threshold or errors on a costly minority class may matter more than aggregate accuracy.

Institutions rarely share identical software. Flower's 2020 design accommodates diverse programming languages and machine-learning frameworks, including mobile and embedded clients [38], yet integration involves data connectors, identity services, and network policies. Failures differ in consequence: too few participants may stall a secure-aggregation round, whereas a missing feature service may block vertical inference after training succeeds. Recovery planning follows service dependencies: checkpoint restoration and rollback address some failures, while compromised credentials and incompatible software need other responses.

In a 2024 healthcare review, Teo et al. included 612 studies and classified 5.2% as real-life FL applications, with literature searched through August 31, 2023 [39]; implementation research examines related integration and translation challenges [40]. Deployment thus extends beyond training: computation, networking, and cryptographic processing consume resources, and changes to keys, software, and membership create recurring costs. Measuring these costs over a common operating horizon allows comparison of architectures with similar utility. Ownership and infrastructure arrangements allocate model access and responsibility for shared services [12]; agreements that account for smaller sites' rare cases or useful feature coverage can help sustain participation.

7. Application Evidence in Enterprise Information Systems

Pati et al. studied glioblastoma boundary detection with data from 71 sites across six continents [13]. Dayan et al. trained EXAM on data from 20 institutions to predict the oxygen requirements of symptomatic COVID-19 patients; using vital signs, laboratory data, and chest X-rays, it achieved a mean area under the receiver operating characteristic curve above 0.92 for outcomes 24 and 72 hours after emergency-room presentation [14]. Sheller et al. also evaluated multi-institutional medical learning without sharing patient data [15]. EXAM's defined prediction horizon illustrates how an output should be tied to a local decision, and healthcare analyses emphasize data harmonization and implementation support [40, 41].

Lee et al. investigate FL for credit-risk assessment [16]. A shared model must benefit each institution without exposing its portfolio, so the framework first compares it with local models under comparable targets and holdouts. For residential short-term load forecasting, Delgado Fernández et al. examine privacy-utility trade-offs [17]; operating such a model adds per-site error monitoring, update communication, and reliance on participants.

8. A Technical Deployment Framework for Enterprise Federated Learning

The seven-stage framework in Table 3 turns the preceding analysis into a deployment workflow, informed by the mechanism comparison, governance research, and risk-management frameworks [12, 33, 34]. Stages 1–2 define participant-level utility targets and record data flows and threats, giving the team a privacy objective rather than only a preferred technology. Stages 3–5 select the architecture, assess each protection against what each component can observe, and assign control over participation, code, keys, and model access. Stage 6 tests local performance, recovery from the failures identified in Section 6, and resource use. Stage 7 monitors operation, so a new participant or changed service can trigger reassessment of only the affected controls.

In an illustrative fraud-detection consortium, members begin with local baselines and agreed error costs. Choosing horizontal learning with secure aggregation to hide individual updates rules out validation methods that require plaintext updates, so the aggregation and validation paths are resolved together before key control and release authority are assigned. If DP is included, its protected unit and privacy objective are fixed before utility testing, and poor utility may call for a different model, task, or participation arrangement within that objective.

Table 3: Seven-stage technical deployment framework for enterprise federated learning.

A / DEFINE SYSTEM OBJECTIVES		
1	System objective	Task, baselines, acceptance, utility criteria
2	Data and threats	Data flows, protected units, permissions, adversaries
B / DESIGN SECURITY AND ARCHITECTURE		
3	Architecture	Participation, partitioning, aggregation, inference
4	Protection controls	Mechanisms, trust assumptions, residual risks, overhead
5	Authorization and ownership	Model access/release, keys, logs, membership, exit
C / VALIDATE AND OPERATE		
6	Deployment readiness	Local validation, interoperability, recovery, budget
7	Continuous assurance	Performance, drift, incidents, cost, threat-model review
Throughout: security, privacy, and compliance. Feedback: revisit affected stages when participants, services, threats, or requirements change.		

9. Future Research Directions

Comparative deployments of the organizational archetypes examined by Barbereau et al. [12] could show whether authority over code and model release affects incident containment, and compare onboarding and retention; discontinued collaborations would reveal hard-to-sustain commitments. Benchmarks should evaluate privacy and integrity jointly, testing whether defenses remain effective when individual updates are hidden. Federated foundation-model adaptation raises specific aggregation problems. FLoRA identifies errors in conventionally aggregating low-rank adaptation (LoRA) parameters and uses stacking to support heterogeneous adapters [42]. Such parameter-efficient adaptation provides no privacy guarantee by itself. Separately, extraction research demonstrates memorized-text disclosure from large language models [43], so exposure must be evaluated alongside efficiency. Distributed retrieval creates another data flow because retrieved passages and query outputs may reveal information even when no model is trained. Longitudinal studies could measure the cost of version changes and participant exits.

10. Conclusion

Enterprise FL keeps records local, but privacy depends on the exchanged signals and the controls applied to them, not merely on where training takes place. Hiding updates during computation and

limiting disclosure through released models are different objectives; the mechanism comparison makes this distinction explicit, together with each approach's trust and resource requirements and the tension between confidential updates and integrity checks. The seven-stage framework carries these choices into operation, because a viable federation needs both a useful model and an operating arrangement that keeps its protections effective as the collaboration evolves.

References

- [1] McMahan, H.B., Moore, E., Ramage, D., et al. (2017) *Communication-Efficient Learning of Deep Networks from Decentralized Data*. *Proceedings of AISTATS*, PMLR, 54, 1273-1282.
- [2] Kairouz, P., McMahan, H.B., Avent, B., et al. (2021) *Advances and Open Problems in Federated Learning*. *Foundations and Trends in Machine Learning*, 14(1-2), 1-210.
- [3] Yang, Q., Liu, Y., Chen, T., et al. (2019) *Federated Machine Learning: Concept and Applications*. *ACM Transactions on Intelligent Systems and Technology*, 10(2), Article 12.
- [4] Li, T., Sahu, A.K., Talwalkar, A., et al. (2020) *Federated Learning: Challenges, Methods, and Future Directions*. *IEEE Signal Processing Magazine*, 37(3), 50-60.
- [5] Zhu, L., Liu, Z. and Han, S. (2019) *Deep Leakage from Gradients*. *Advances in Neural Information Processing Systems*, 32.
- [6] Geiping, J., Bauermeister, H., Dröge, H., et al. (2020) *Inverting Gradients—How Easy Is It to Break Privacy in Federated Learning?* *Advances in Neural Information Processing Systems*, 33.
- [7] Melis, L., Song, C., De Cristofaro, E., et al. (2019) *Exploiting Unintended Feature Leakage in Collaborative Learning*. *IEEE Symposium on Security and Privacy*, 691-706.
- [8] Nasr, M., Shokri, R. and Houmansadr, A. (2019) *Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-Box Inference Attacks Against Centralized and Federated Learning*. *IEEE Symposium on Security and Privacy*, 739-753.
- [9] Bai, L., Hu, H., Ye, Q., et al. (2025) *Membership Inference Attacks and Defenses in Federated Learning: A Survey*. *ACM Computing Surveys*, 57(4), Article 89.
- [10] Mothukuri, V., Parizi, R.M., Pouriyeh, S., et al. (2021) *A Survey on Security and Privacy of Federated Learning*. *Future Generation Computer Systems*, 115, 619-640.
- [11] Sum, A.S.I., Pritee, Z.T., Saha, A.K., et al. (2026) *A Systematic Review on Privacy Preservation in Federated Learning*. *International Journal of Information Security*, 25, Article 65.
- [12] Barbereau, T., Delgado Fernandez, J. and Potenciano Menci, S. (2025) *The Governance of Federated Learning: A Decision Framework for Organisational Archetypes*. *Data & Policy*, 7, e53.
- [13] Pati, S., Baid, U., Edwards, B., et al. (2022) *Federated Learning Enables Big Data for Rare Cancer Boundary Detection*. *Nature Communications*, 13, Article 7346.
- [14] Dayan, I., Roth, H.R., Zhong, A., et al. (2021) *Federated Learning for Predicting Clinical Outcomes in Patients with COVID-19*. *Nature Medicine*, 27, 1735-1743.
- [15] Sheller, M.J., Edwards, B., Reina, G.A., et al. (2020) *Federated Learning in Medicine: Facilitating Multi-Institutional Collaborations without Sharing Patient Data*. *Scientific Reports*, 10, Article 12598.
- [16] Lee, C.M., Delgado Fernández, J., Potenciano Menci, S., et al. (2023) *Federated Learning for Credit Risk Assessment*. *Proceedings of the 56th Hawaii International Conference on System Sciences*.
- [17] Delgado Fernández, J., Potenciano Menci, S., Lee, C.M., et al. (2022) *Privacy-Preserving Federated Learning for Residential Short-Term Load Forecasting*. *Applied Energy*, 326, Article 119915.
- [18] Shokri, R., Stronati, M., Song, C., et al. (2017) *Membership Inference Attacks Against Machine Learning Models*. *IEEE Symposium on Security and Privacy*, 3-18.
- [19] Fredrikson, M., Jha, S. and Ristenpart, T. (2015) *Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures*. *Proceedings of ACM CCS*, 1322-1333.
- [20] Bagdasaryan, E., Veit, A., Hua, Y., et al. (2020) *How to Backdoor Federated Learning*. *Proceedings of AISTATS*, PMLR, 108, 2938-2948.
- [21] Bonawitz, K., Ivanov, V., Kreuter, B., et al. (2017) *Practical Secure Aggregation for Privacy-Preserving Machine Learning*. *Proceedings of ACM CCS*, 1175-1191.
- [22] Truex, S., Baracaldo, N., Anwar, A., et al. (2019) *A Hybrid Approach to Privacy-Preserving Federated Learning*. *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, 1-11.
- [23] Dwork, C. and Roth, A. (2014) *The Algorithmic Foundations of Differential Privacy*. *Foundations and Trends in Theoretical Computer Science*, 9(3-4), 211-407.
- [24] McMahan, H.B., Ramage, D., Talwar, K., et al. (2018) *Learning Differentially Private Recurrent Language Models*. *International Conference on Learning Representations*.

- [25] Geyer, R.C., Klein, T. and Nabi, M. (2017) *Differentially Private Federated Learning: A Client-Level Perspective*. arXiv:1712.07557.
- [26] Abadi, M., Chu, A., Goodfellow, I., et al. (2016) *Deep Learning with Differential Privacy*. *Proceedings of ACM CCS*, 308-318.
- [27] Near, J.P., Darais, D., Lefkowitz, N., et al. (2025) *Guidelines for Evaluating Differential Privacy Guarantees*. NIST SP 800-226, National Institute of Standards and Technology.
- [28] Mohassel, P. and Zhang, Y. (2017) *SecureML: A System for Scalable Privacy-Preserving Machine Learning*. *IEEE Symposium on Security and Privacy*, 19-38.
- [29] Gentry, C. (2009) *Fully Homomorphic Encryption Using Ideal Lattices*. *Proceedings of the 41st ACM Symposium on Theory of Computing*, 169-178.
- [30] Zhang, C., Li, S., Xia, J., et al. (2020) *BatchCrypt: Efficient Homomorphic Encryption for Cross-Silo Federated Learning*. *USENIX Annual Technical Conference*, 493-506.
- [31] Mo, F., Haddadi, H., Katevas, K., et al. (2021) *PPFL: Privacy-Preserving Federated Learning with Trusted Execution Environments*. *Proceedings of ACM MobiSys*, 94-108.
- [32] European Parliament and Council of the European Union (2016) *Regulation (EU) 2016/679 (General Data Protection Regulation)*. *Official Journal of the European Union*, L 119, 1-88.
- [33] Boeckl, K. and Lefkowitz, N. (2020) *NIST Privacy Framework: A Tool for Improving Privacy Through Enterprise Risk Management, Version 1.0*. NIST CSWP 01162020, National Institute of Standards and Technology.
- [34] Tabassi, E. (2023) *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1, National Institute of Standards and Technology.
- [35] Li, T., Sahu, A.K., Zaheer, M., et al. (2020) *Federated Optimization in Heterogeneous Networks*. *Proceedings of Machine Learning and Systems*, 2.
- [36] Karimireddy, S.P., Kale, S., Mohri, M., et al. (2020) *SCAFFOLD: Stochastic Controlled Averaging for Federated Learning*. *Proceedings of ICML, PMLR*, 119, 5132-5143.
- [37] Reddi, S.J., Charles, Z., Zaheer, M., et al. (2021) *Adaptive Federated Optimization*. *International Conference on Learning Representations*.
- [38] Beutel, D.J., Topal, T., Mathur, A., et al. (2020) *Flower: A Friendly Federated Learning Research Framework*. arXiv:2007.14390v1.
- [39] Teo, Z.L., Jin, L., Liu, N., et al. (2024) *Federated Machine Learning in Healthcare: A Systematic Review on Clinical Applications and Technical Architecture*. *Cell Reports Medicine*, 5(2), Article 101419.
- [40] Li, M., Xu, P., Hu, J., et al. (2025) *From Challenges and Pitfalls to Recommendations and Opportunities: Implementing Federated Learning in Healthcare*. *Medical Image Analysis*, 101, Article 103497.
- [41] Rieke, N., Hancox, J., Li, W., et al. (2020) *The Future of Digital Health with Federated Learning*. *npj Digital Medicine*, 3, Article 119.
- [42] Wang, Z., Shen, Z., He, Y., et al. (2024) *FLoRA: Federated Fine-Tuning Large Language Models with Heterogeneous Low-Rank Adaptations*. *Advances in Neural Information Processing Systems*, 37.
- [43] Carlini, N., Tramèr, F., Wallace, E., et al. (2021) *Extracting Training Data from Large Language Models*. *30th USENIX Security Symposium*, 2633-2650.