pdpc
PERSONAL DATA
PROTECTION COMMISSION
SINGAPORE

PRIVACY ENHANCING TECHNOLOGIES (PETs):

GUIDE ON FEDERATED LEARNING

A Technique for
Collaborative AI Development

Jointly developed with

GOVTECH
SINGAPORE

Supported by

INFOCOMM
MEDIA
DEVELOPMENT
AUTHORITY

CONTENTS

EXECUTIVE SUMMARY

Privacy Enhancing Technologies (PETs) enable organisations to process, analyse and extract insights from data without revealing any underlying personal or commercially sensitive information. **Federated Computation** is one such technique addressing a specific and growing challenge: how to enable collaborative AI development across organisations, whose datasets are isolated by regulatory, competitive or operational constraints. It does so by bringing computation to where data resides rather than centralising raw sensitive information. It encompasses two complementary capabilities: **Federated Learning (FL)** and **Federated Analytics (FA)**.

This guide primarily focuses on **FL** due to its greater technical complexity and privacy risks. It includes structured assessment for evaluating problem-solution fit and implementation readiness, practical guidance on design decisions, as well as governance controls and technical measures to mitigate risks. **FA** is also addressed, both as a stepping stone to FL and a useful standalone application.

The target audience for this guide includes Chief Information Officers (CIOs), Chief Technology Officers (CTOs), Chief Data Officers (CDOs), AI and data scientists, data protection practitioners, and technical decision-makers who may directly or indirectly be involved in the evaluation, implementation, and governance of federated systems.

FL is a technology that is being actively researched and developed at the time of publication. Hence, this guide is not intended to provide a comprehensive or in-depth review of the technology or its assessment methods. The guide is intended to be a living document and will be updated to ensure its recommendations remain relevant.

READER'S GUIDE

Privacy Enhancing Technologies (PETs): Guide on Federated Learning (FL)	
Objectives	To demystify FL and equip potential adopters with the practical knowledge needed to conceptualise, evaluate, prepare for and confidently navigate the FL adoption journey. It also introduces FA as a common precursor and complementary capability to FL deployments.
Intended Audience	Business, technology and data protection stakeholders (e.g. CIOs, CTOs, CDOs, data scientists, data protection practitioners) who have developed a foundational awareness of PETs and are exploring whether FL is the right solution to address their specific data sharing and collaboration challenges.
How to Use This Guide	Use this guide as a practical reference at any stage of your FL journey. If you are still exploring whether FL is right for your organisation, start with Sections 2.1 to 2.3 to build your understanding. If you are closer to implementation, the assessments in Section 2.4 and the risk guidance in Section 2.5 will be most immediately useful. The annexes provide deeper technical detail for teams ready to move into the design-and-build portion. While FL implementation relies on underlying data systems and the broader AI lifecycle, this guide assumes that the adopting organisations already meet baseline standards for data protection, cybersecurity, incident response and general AI governance. Rather than repeating these foundations, this guide focuses on the unique risks and controls introduced by FL's distributed and multi-party architecture.

FEDERATED COMPUTATION

FEDERATED COMPUTATION[1]

The federated approach represents a significant shift[2] in how **Machine Learning (ML)** is conducted, moving away from the traditional centralised process towards a distributed process. Rather than aggregating raw data in a single location, the federated approach brings computation to where participants' data resides. Only aggregated insights or model updates (e.g. parameters, gradients or representations) are exchanged, enabling participants to collectively develop models and insights while keeping their underlying data private and under local control.

The federated approach encompasses two complementary capabilities:

- **Federated Learning (FL)** is an ML setting where multiple participants collaborate to solve a machine learning problem, under the coordination of a central server or service provider.[3] A well-designed FL system aims to go beyond basic model training by acting as a collaborative framework with privacy-preserving properties.[4]
- **Federated Analytics (FA)** is the practice of applying data science methods to the analysis of raw data stored locally on the participants' side.

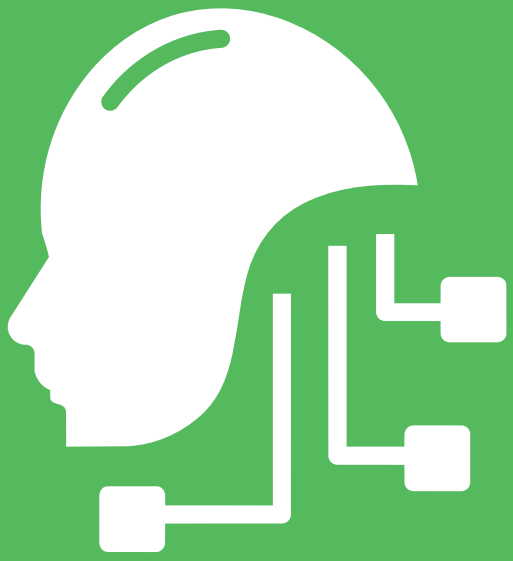
FA can be used as a precursor to FL, to support data discovery, quality assessment, harmonisation and feasibility analysis. It can also be used as a standalone approach to generate shared insights, support reporting, conduct research or inform operational decisions without pooling raw data.

[1] *Bharadwaj, A., & Cormode, G. (2024). Federated computation: a survey of concepts and challenges. Distributed and Parallel Databases, 42, 299–335. https://doi.org/10.1007/s10619-023-07438-w*

[2] *Reddi, S. J., et al. (2021). Adaptive federated optimization. International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=LkFG3lB13U5*

[3] *Bonawitz, K., Kairouz, P., McMahan, B., & Ramage, D. (2022). Federated learning and privacy. Communications of the ACM, 65(4), 90–97. https://doi.org/10.1145/3494834.3500240*

[4] *Daly, K., Eichner, H., Kairouz, P., McMahan, H. B., Ramage, D., & Xu, Z. (2024). Federated learning in practice: Reflections and projections. arXiv preprint. https://arxiv.org/pdf/2410.08892*



FEDERATED LEARNING

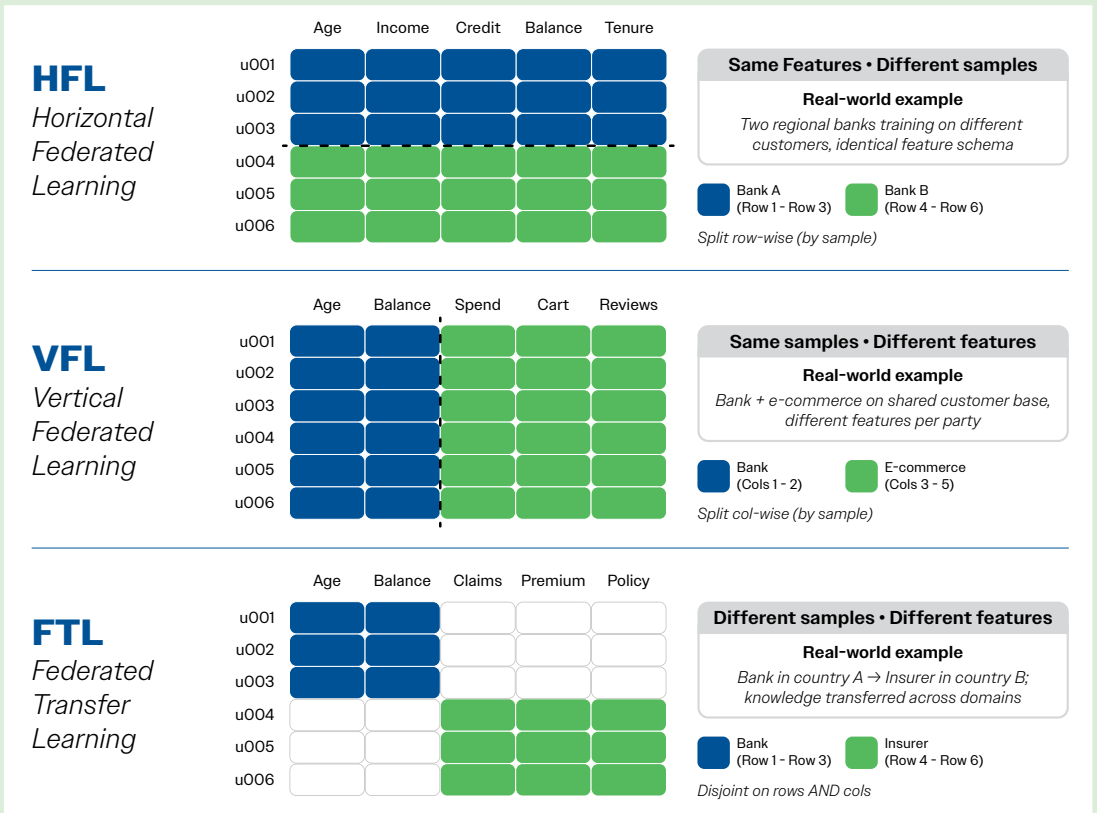
2.1 WHAT IS FEDERATED LEARNING?

2.1.1 Federation Types and Coordination Architectures

FL systems can be categorised in two ways: (1) by Federation Types, and (2) by Coordination Architectures.

Firstly, the nature of data distribution[5] across participants gives rise to different **Federation Types**[6] (including horizontal, vertical and transfer learning). **Figure 1** and **Table 1** give a high-level elaboration of these federation types, whilst **Annex A** provides more comprehensive technical details. Based on industry observations, modern FL deployments, particularly in healthcare and finance, are increasingly likely to require hybrid approaches that combine multiple federation types within a single system. For instance, a network of healthcare providers might employ horizontal FL for patient records across hospitals whilst simultaneously using vertical FL for specialist diagnostic data from imaging centres, creating a multi-layered federated architecture that addresses the complex data distribution patterns typically found in real-world scenarios.

Figure1: Illustration of Three Types of Federated Learning



[5] *Data distribution refers to how datasets are divided across participants in terms of which features (columns) and samples (rows) each participant possesses.*

[6] *IEEE. (2021). IEEE guide for architectural framework and application of federated machine learning. IEEE Std 3652.1-2020. https://ieeexplore.ieee.org/document/9382202*

Type	Feature Space[7]	Sample (ID) Space [8]	Example
HFL *(Horizontal Federated Learning)*	Same/ Large Overlap	Different/ Small Overlap	Two regional banks operating in different cities share the same data schema (e.g. transaction records, credit scores) but serve largely distinct customer bases, so they collaborate to train a more generalisable model without exchanging raw records.
VFL *(Vertical Federated Learning)*	Different/ Small Overlap	Same/ Large Overlap	A bank and an e-commerce platform serve the same users but know different things about them. One holds credit history, the other, purchase behaviour. By combining these complementary views, they can jointly train a model neither could build alone.
FTL *(Federated Transfer Learning)*	Different/ Small Overlap	Different/ Small Overlap	A bank and an insurer operate in different industries, so they share little common ground with regard to the users they serve and the data they collect. Yet the underlying patterns, like financial behaviour, are similar enough that the knowledge gained in one domain can still be meaningfully transferred to the other.

Secondly, the FL implementation can follow different **Coordination Architectures** (including centralised, decentralised and hierarchical). Detailed descriptions of coordination architectural options and alternative coordination mechanisms are available in **Annex B**. The architecture selection is influenced by multiple factors beyond trust and central party availability. The detailed architecture selection considerations are provided in **section 2.4.3.**

[7] *Feature space refers to the set of data features / attributes / variables collected about each subject. E.g. age, income or transaction history.*

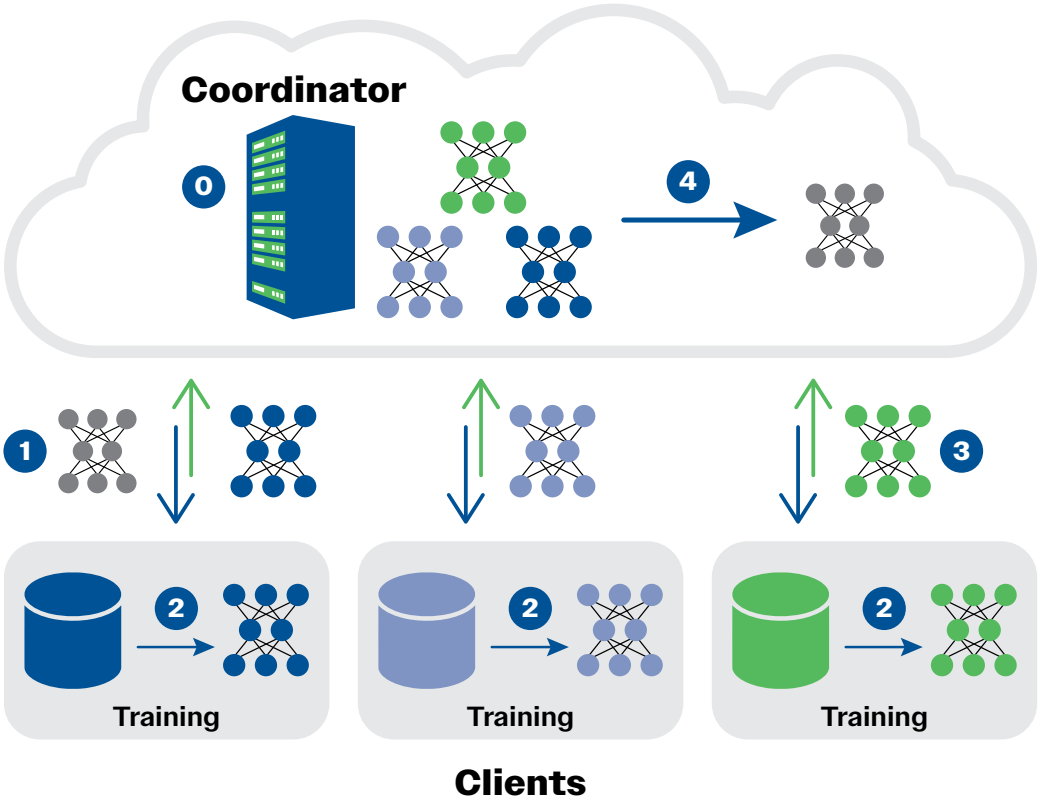
[8] *Sample (ID) space refers to the set of entities or individuals that a party has data on. E.g. the actual customers in a bank's database.*

2.1.2 Training Process and Mechanism

FL is a distributed machine learning approach. Unlike traditional centralised ML, where all data is available in one environment and training occurs in a single system, **FL involves coordinating across distributed participants**, each holding their own private data. Each participant trains their own local model[9] on their own data. These local models are then iteratively combined to update a shared global model.[10]

FL is an iterative process that runs across multiple rounds until the global model converges. **Each round follows a structured sequence:** (0) model initialisation, (1) participant selection and model distribution, (2) local training, (3) update collection and aggregation, and (4) model synchronisation. Various aggregation techniques can be employed at Step (3), including FedAvg and FedProx, which are discussed in detail in **Annex C**. For a step-by-step breakdown of each stage, please refer to **Annex D**.

Figure 2: Illustration of FL Model Training Process



[9] *Local model enables personalisation, as it captures patterns and insights specific to that participant's dataset, which may reflect unique characteristics such as regional preferences, demographic distributions, or operational contexts. Local models are essential because they preserve data privacy by keeping sensitive information on-device, whilst the model updates derived from local training serve as the foundation for knowledge sharing.*

[10] *Global model represents the aggregated knowledge acquired from all participating local models, incorporating insights from the collective datasets without directly accessing any individual participant's raw data. Global model is crucial because it captures patterns and relationships that may not be apparent in any single participant's limited dataset, potentially achieving superior performance compared to the models trained on individual datasets alone.*

2.2 HOW IS FEDERATED LEARNING USED?

FL has emerged as a practical response to several growing barriers in AI development. Traditional centralised training is increasingly constrained[11] by the exhaustion of high-quality public data, tightening data regulations, computational bottlenecks and concerns about the concentration of AI capabilities among a few large technology companies. By enabling organisations to collaborate on model training without pooling their raw data, FL makes AI development more accessible, representative and privacy-respecting.

2.2.1 Use Case Archetypes

The landscape of FL applications can be broadly categorised into the following distinct archetypes:

Table 2: FL Use Case Archetypes

Participation Scale	Archetype-Specific Best Practices	Representative Use Cases
Use Case Archetype 1: Cross-device *For edge device learning and personalisation[12] (B2C)* 		
Massive consumer participation on voluntary basis	• Implement a lightweight model for resource-constrained devices. • Use hierarchical aggregation and edge computing to address the limitations of the Internet of Things (IoT). • Design fault-tolerant systems for device dropouts. • Manage battery consumption and network usage on mobile devices. • Use a synchronised FL algorithm to accommodate devices with intermittent connectivity and variable response times.	• **Consumer Technology:** Keyboard prediction, voice assistants, biometric authentication • **E-commerce:** Personalised recommendations and search • **Automotive**: Autonomous driving and vehicle-to-vehicle learning • **Telecommunications:** Network optimisation and service personalisation • **IoT:** Swarm coordination, underwater sensor networks for monitoring

[11] *Economic Policy. AI monopolies. 79th Economic Policy Panel. https://economic-policy.org/79th-economic-policy-panel/ai-monopolies/*

[12] *Tan, A. Z., Yu, H., Cui, L., & Yang, Q. (2022). Towards personalized federated learning. IEEE Transactions on Neural Networks and Learning Systems. https://doi.org/10.1109/TNNLS.2022.3160699*

Participation Scale	Archetype-Specific Best Practices	Representative Use Cases
Use Case Archetype 2: Cross-silo *For cross-organisational collective learning (B2B)*		
Institutional participants with formal agreements	• Establish clear data governance frameworks. • Implement robust participant authentication. • Create standardised data schemas across institutions. • Address institutional competitive concerns through fair contribution metrics.	• **Financial Services:** Cross-bank fraud detection and risk assessment • **Healthcare:** Multi-hospital diagnosis model development and drug discovery
** Note: Cross-border is a specialised variant of cross-silo FL, distinguished by increased scale, coordination complexity and additional legal/regulatory considerations across jurisdictions.*		
Government and multinational participants with treaty-level agreements	• Ensure compliance with varying privacy regulations. • Create protocols that accommodate different national technical standards and varying levels of infrastructure.	• **Healthcare[13]:** Multi-institutional medical imaging analysis for diagnosis

A note on terminology: This guide uses several terms to refer to entities that contribute data and computation to a federated system. "Participants" is the general term used throughout. In cross-device contexts, these are often called "clients" or "devices"; in cross-silo contexts, they may be referred to as "nodes" or "parties". These terms are broadly interchangeable unless the context specifies otherwise.

[13] *Lifebit. Federated learning applications.* https://lifebit.ai/blog/federated-learning-applications/

2.2.2 CASE STUDIES

The following case studies illustrate how these archetypes play out in practice.

A **Cross-Device:**

Personalised Language Modelling in Google Gboard[14]

Problem Statement: Delivering accurate next-word predictions and autocorrect features requires training on vast, diverse linguistic patterns of a global user base. However, keystroke data is highly sensitive, often containing passwords, financial details or private conversations. Traditional centralised ML, which would require uploading this raw text to a cloud server, creates a massive privacy "honeypot" and poses privacy risks.

Solution: Google addresses this challenge by employing decentralised, privacy-preserving architectures that ensure raw user text is never visible to a central server. For model training, Gboard utilises FL, where devices compute local model updates that are cryptographically combined via secure aggregation, ensuring only collective intelligence leaves the device. For broader analytical tasks such as discovering trends, the system leverages Confidential FA. Under this paradigm, devices upload end-to-end encrypted data that remains completely inaccessible to the infrastructure until it is aggregated privately inside a server-side, hardware-isolated Trusted Execution Environment. Together, these complementary mechanisms ensures that whether insights are computed on-device or within a secure cloud enclave, individual raw data is never exposed.

Key Benefits: By decoupling utility from data visibility, this architecture eliminates the traditional single-point-of-failure data breach risk. The global system learns only collective, aggregated linguistic trends, while raw, individual user text remains entirely inaccessible to anyone but the user.

[14] *Gboard, et al. (2023). Private federated learning in Gboard. arXiv. https://arxiv.org/html/2306.14793v1*

Distributed Battery Range Estimation of Electric Vehicle with NVIDIA FLARE[15]

Problem Statement: Reliable battery range prediction is essential for reducing "range anxiety". Yet, training accurate AI models traditionally requires transmitting massive amounts of sensitive vehicle data to a central cloud. This centralised approach faces significant hurdles, including high communication costs, the need for expensive infrastructure and serious privacy concerns regarding the tracking of individual driving patterns and locations. Furthermore, a one-size-fits-all global model often fails to account for the unique driving styles and environmental conditions of individual EV owners.

Solution: Toyota implemented FL using a hybrid fleet of physical remote-controlled cars (equipped with NVIDIA Jetson Nano™) and simulated vehicles via the SUMO traffic model. Instead of uploading raw telemetry data, each vehicle performs local training on its own battery consumption data. Using NVIDIA FLARE™ as the orchestration engine, only the model weights and updates are shared with a central server to refine a global prediction model. Once the global model is established, each car performs local fine-tuning to personalise the AI to its specific driver, ensuring high accuracy while keeping all raw data securely on the vehicle.

Key Benefits: This decentralised approach offers a practical and cost-effective path to smarter mobility by drastically reducing the resource burden of AI development. By sharing only model updates, the system uses approximately 10x less network and computing resources compared to traditional cloud-based training. Privacy is maintained by design, as sensitive driving data never leaves the car's onboard computer. Crucially, the combination of collective fleet intelligence and individual fine-tuning results in more precise range estimates tailored to the user. This demonstrates that FL can provide the robust, real-time predictions necessary to build driver confidence and optimise charging infrastructure without compromising data security.

[15] *NVIDIA. (2025). Distributed battery range estimation of electric vehicle via personalized federated learning. NVIDIA FLARE Day 2025.* https://www.nvidia.com/en-us/on-demand/session/nvidiaflareday25-nvfd25/?playlistId=playList-4dd15c3d-2422-425c-b9d9-d21694397574

B Cross-Silo:

Duality Collaborative AI for Oncology Research at Dana Farber[16]

Problem Statement: Training robust AI models for cancer classification requires massive, diverse datasets that span multiple research institutes. However, because pathology images constitute Protected Health Information (PHI), sharing them across organisations triggers strict privacy, security and intellectual property regulations. This makes acquiring external oncology data a slow and complex process, creating a significant bottleneck for collaborative research.

Solution: FL allows healthcare institutions to train AI models collaboratively without moving or exposing raw patient data. Training occurs locally at each site, and only encrypted model updates are transmitted. These updates are then securely aggregated within the Google Cloud Confidential Space to generate a unified global model.

Key Benefits: FL's privacy-preserving approach unlocked historically siloed medical imaging data across multiple clinical centres. By safely scaling data access, the platform significantly enhances model accuracy and generalisability, resulting in AI systems capable of more precise cancer diagnoses and tailored treatment plans across diverse patient populations.

[16] *Duality Technologies. Collaborative AI for oncology research.* https://dualitytech.com/use-cases/healthcare-industry/collaborative-ai-for-oncology-research/

Collaborative Customer Engagement Prediction by Ant International and its Partner Wallets[17]

Problem Statement: Ant International and its partner digital wallets seek to optimise marketing campaigns by predicting customer voucher redemptions offered on the Alipay+ Rewards platform. However, the necessary data is fragmented. Ant International holds voucher redemption history, while the wallets hold purchase history and demographic data. Because personal data concerns and business sensitivities prevent the sharing of common identifiers or the movement of data outside their respective environments, the entities cannot aggregate their datasets. This data silo constraint prevents the training of an accurate prediction model, limiting the ability to deliver relevant promotions to customers.

Solution: To bridge their data silos, Ant International and Wallet deployed a Vertical FL framework integrated with Multi-Party Computing (MPC) and Homomorphic Encryption (HE). The solution was to first identify common customers through Private Set Intersection, then co-train decision tree models locally within each party's environment. Throughout the training process, only encrypted model updates and secret shares were exchanged — never actual customer data. Each party maintains an encrypted model, which is used to generate collaborative predictions about customer voucher redemption likelihood, whilst keeping all sensitive information within their respective secure environments. These encrypted models do not reveal any information about the whole model or the dataset and cannot be used on their own. The result is only revealed to authorised parties.

Key Benefits: The Proof-of-Concept (POC) proved that the solution enables secure access to fragmented datasets, significantly improving customer pre-selection for marketing campaigns without compromising data privacy.

[17] *Infocomm Media Development Authority (IMDA). (2026). Enhancing customer engagement with privacy preserving AI: IMDA PET Sandbox – Ant International case study.* https://share.google/913TklMAbktmfAQow

Global (Cross-Border) Federated Learning for Rare Cancer Boundary Detection[18]

Problem Statement: Training robust AI models requires large, diverse datasets, but this is particularly challenging for "rare" diseases like glioblastoma, where individual hospitals lack sufficient data volume to train generalisable models. While pooling data from multiple institutions could solve the scarcity issue, centralised data sharing is often infeasible due to patient privacy concerns, data ownership disputes and strict regulatory hurdles. Consequently, models trained at single sites often fail to generalise to "out-of-sample" data from other institutions.

Solution: The study orchestrated the largest FL effort to date, connecting 71 sites across 6 continents, to train a consensus model on 6,314 patients, the largest dataset ever reported for this disease. The system used a central aggregation server where participating sites shared only model parameter updates, ensuring raw MRI scans never left the local institutions. The federation implemented a standardised preprocessing pipeline to handle the heterogeneity of MRI scanners and acquisition protocols across the 71 different sites. To mitigate risks during the aggregation phase, the workflow utilised Trusted Execution Environments (TEEs) using Intel SGX hardware to ensure the confidentiality and integrity of the model updates.

Key Benefits: The global consensus model achieved a 33% improvement in delineating the "tumour core" and a 23% improvement in the complete tumour extent compared to a model trained on publicly available data. The study proved that FL allows models to gain knowledge from diverse global data, significantly outperforming local models when tested on unseen "out-of-sample" populations. This use case demonstrates that FL can handle complex tasks at a massive global scale, effectively alleviating the need for centralised data sharing.

[18] *Pati, S., Baid, U., Edwards, B., Shelman, M., Shinohara, R. T., Abe, S., ... & Bakas, S. (2022). Federated learning enables big data for rare cancer boundary detection. Nature Communications, 13(1), 7346.* https://doi.org/10.1038/s41467-022-33407-5

Growing Relevance of Federated Learning in Generative AI[19] Era

The rise of generative AI introduces new contexts to which FL is applied.

Rather than training models from scratch, organisations now seek to customise powerful, pre-existing models using their own proprietary data. This process is known as federated fine-tuning.[20] Applying FL to this fine-tuning process is far more complex than traditional FL settings, given the sheer scale of the models and data involved. Nevertheless, FL remains a promising approach for enabling privacy-preserving collaboration in this space, with frameworks like FlowerTune[21] and Mozilla.ai[22] already supporting these workflows in enterprise settings.

[19] *Sani, L., Iacob, A., Cao, Z., Marino, B., Gao, Y., Paulik, T., ... & Lane, N. D. (2024). The Future of Large Language Model Pre-training is Federated. arXiv preprint arXiv:2405.10853. https://arxiv.org/abs/2405.10853*

[20] *NEC Corporation. (2023). Federated learning technology that enables collaboration while keeping data confidential and its applicability to LLMs. NEC Technical Journal, 17(2). https://www.nec.com/en/global/techrep/journal/g23/n02/230215.html*

[21] *Flower Labs. FlowerTune LLM. https://flower.ai/docs/examples/flowertune-llm.html*

[22] *Mozilla AI. Finetune an LLM using federated learning. Mozilla AI Blueprints. https://blueprints.mozilla.ai/all-blueprints/finetune-an-llm-using-federated-learning*

2.3 KEY IMPLEMENTATION CHALLENGES

While FL offers data enrichment and data localisation benefits, it introduces certain challenges[23] across business, operational and algorithmic dimensions.

Business Challenges

FL implementations require extensive cross-organisational coordination and strategic alignment. Key business challenges include:

- **Intellectual Property:** FL collaborations require clear, binding agreements regarding model ownership, derivative works and downstream commercialisation rights to protect proprietary assets while sharing collaborative benefits.
- **Cost and Resource Allocation:** Designing, deploying and maintaining a secure infrastructure demands significant upfront and ongoing engineering investments.
- **Security Guarantees and Risk Calibration:** While FL keeps raw data localised, sharing iterative model updates can still leak indirect signals or statistical patterns of the underlying data. Because the level of security depends heavily on the implementation design and layered privacy-enhancing techniques, organisations must align with each other on the technical guarantees required to match their shared risk tolerance. Potential attack vectors are detailed in **Annex E**, while technical controls to manage risks are covered in **Annex F**.

[23] *Kairouz, P., et al. (2019). Advances and open problems in federated learning. Foundations and Trends in Machine Learning, 4 (2021). https://arxiv.org/abs/1912.04977*

Operational Challenges

Successful FL requires alignment between use cases and available datasets across participants. Key operational challenges include:

- **Data Harmonisation:** Efforts are needed to standardise data formats, feature definitions and quality metrics across participants.
- **Communication and Computational Overhead:** Transmitting model updates across networks multiple times may introduce latency and incur bandwidth costs. Each participant would also need to maintain sufficient local processing capabilities.
- **System heterogeneity:** Differences in hardware, software and network configurations complicate coordination, monitoring and debugging across organisational boundaries.
- **Reproducibility:** Ensuring consistent results across different federated configurations and participant combinations poses challenges for validation and regulatory compliance.

Algorithmic Challenges

FL systems frequently operate across participants with heterogeneous data and environments (e.g. non-independent and identically, non-IID, distributed data), alongside differences in context, system constraints, annotation standards and data modalities. This multi-dimensional heterogeneity can affect model convergence, training stability and overall performance, particularly in applications such as large language models and multimodal systems. Techniques for assessing and mitigating data heterogeneity are discussed in **Annex H.**

Given these multifaceted challenges, organisations require a structured approach to assess their readiness and systematically address potential risks. The following sections provide practical frameworks for FL adoption and risk management.

2.4 FEDERATED LEARNING ADOPTION ROADMAP

We recommend a phased approach that transitions from strategic alignment to operational planning and technical execution.

This journey spans three core stages:

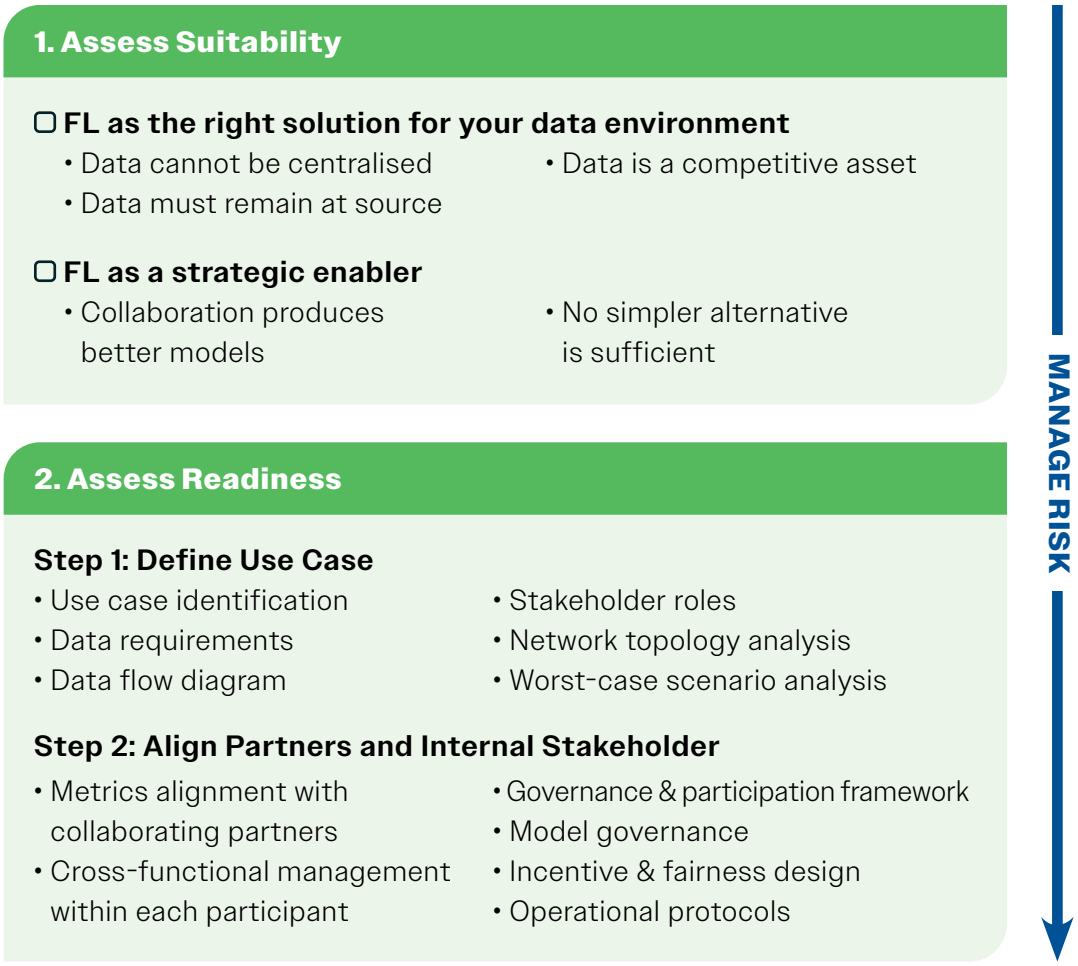
1. **Assess Suitability**, which establishes strategic alignment;

2. **Assess Readiness**, which focuses on operational planning; and

3. **Design and Configure**, which delivers the technical execution.

The following sections explore each element in depth. As illustrated in Figure 3, organisations should navigate these while keeping **Risk Management** (detailed in Section 2.5) as the central, connecting concern that underpins the entire roadmap.

Figure 3: Federated Learning Adoption Roadmap



2. Assess Readiness

Step 3: Understand Internal Technology and Data Landscape

- System compatibility and infrastructure provisioning
  Data standardisation
- Cost and infrastructure implication
- ML maturity
- Data sample alignment

Step 4: Validate FL Solution

- Experimental performance validation
- Technical capabilities
  Vendor and platform strategy
- Audit and compliance monitoring

3. Design Configuration

☐ Before Training

- Federation type
- Coordination architecture
- Coordinator trust model
- Model size
- Model initialization
- FL Framework
- Complementary PETs
- Access control

☐ During Training

- Aggregation method
- Contribution weighting
- Participant selection
- Handling slow participants
- Local training effort
- Update compression

☐ Across the Programme Lifecycle

- Data heterogeneity
- Fairness across participants
- Model quality validation
- Model lifecycle management

MANAGE RISK

2.4.1 Assess Suitability

FL should be considered where it addresses genuine constraints or strategic needs. If data can be feasibly centralised, or if simpler PETs can meet the requirements, these conventional approaches are typically more efficient.

Organisations may use the conditions listed in the table below to evaluate whether FL fits their specific needs.

FL Suitability Assessment

Reasons to Choose FL *These are the conditions where FL provides unique value.*	
FL as the Right Solution for Your Data Environment	• **Data Cannot Be Centralised:** There is a clear and specific reason why data cannot or should not be centralised, such as regulatory restrictions, contractual obligations or operational constraints. • **Data Must Remain at Source:** Data is generated and must be used where it lives, such as on edge devices, within secure systems, or across national boundaries where cross-border data transfer would introduce unacceptable latency, security exposure or data integrity risks. • **Data Is a Competitive Asset:** Participants hold data that is competitively sensitive or represents a core business asset, such as proprietary trading patterns or clinical trial results, where sharing raw data would undermine strategic interests even if legally permissible. FL enables collective intelligence without any participant exposing what gives them a competitive edge.
FL as a Strategic Enabler	• **Collaboration Produces Better Models:** Training on diverse, cross-organisational data produces models that are statistically more robust, better generalised and higher performing than any model trained on a single participant's dataset alone, particularly where individual datasets are small, sparse or unrepresentative of the full population. FL delivers collaborative performance gains that no single participant could achieve on their own. • **No Simpler Alternative Is Sufficient:** Simpler privacy-preserving methods have been considered but cannot meet the objective. For example, anonymisation techniques does not adequately preserve data utility, and FA alone cannot meet the modelling requirements.

2.4.2 Assess Readiness

The following considerations are intended to help organisations think through key dimensions of FL readiness. Teams in early exploration may focus on the use case definition and stakeholder alignment, while those closer to implementation will find the technical and solution assessment sections more immediately useful.

Step 1: Define Use Case

Before committing to FL, it is important to clearly define what problem you are trying to solve and whether your data environment requires a federated approach. The following points can help structure that thinking:

FL Readiness Assessment

From the checklist published in the PETs Adoption Guide[24] (adapted for FL)

- **Use Case Identification** — Identify use cases with data collaboration needs blocked by data sharing barriers. Document specific constraints (regulatory, policy, competitive) that FL could overcome.
- **Data Requirements** — In FL, the focus is on **"mapping what data exists locally across participants"**. Organisations may:
  - identify available features in each participant and whether data schemas require harmonisation;
  - assess minimum local data quality and volume thresholds for meaningful participation;
  - consider the privacy implications of shared model updates (e.g. gradients or weights), which may reveal information about underlying records; and
  - review whether training datasets contain or comprise personal data. If there is personal data:
    - Where required, ensure that the individuals involved have given consent and have been informed that their data will be used for model training.
    - Where possible, apply data minimisation principles to ensure only necessary data is used for training.
    - Where possible, ensure that direct and indirect identifiers are removed or transformed to reduce identifiability, e.g. through obfuscation, hashing and removal of links to identity or across datasets.

FL Readiness Assessment

From the checklist published in the PETs Adoption Guide[24] (adapted for FL)	• **Data Flow Diagram** — FL introduces bidirectional flows between the global model and participants. Beyond standard data flow mapping, organisations may:
	  ◦ map how the global model is distributed to participants and how local updates return to the aggregation server;
	  ◦ analyse data distribution across participants to determine whether horizontal, vertical or transfer FL is appropriate; and
	  ◦ consider if there may be cross-border transfer of personal data between the global model and participants. If so, ensure that the applicable legal requirements are met.
	• **Stakeholder Roles** — FL distributes governance responsibilities across participating organisations. Each role should clearly define responsibilities, access rights and accountability mechanisms. Where personal data is being processed, roles and responsibilities as data controllers and data intermediaries should be clearly defined to support compliance with applicable data protection laws. Additional roles may include:
	  ◦ aggregation servers combining model updates;
	  ◦ secure execution environments providing cryptographic safeguards; and
	  ◦ model validation or monitoring roles.

FL-specific additions	• **Network Topology Analysis** — Organisations may assess how federated coordination integrates with existing workflows across participants. This may involve:
	  ◦ identifying organisations with existing local training pipelines;
	  ◦ assessing infrastructure and skills required for participants without local training capability; or
	  ◦ designing architecture diagrams illustrating model updates and parameter flows between participants and the coordination server.
	• **Contingency Analysis** — Identify and document contingencies, such as regulatory exposure, harm potential, IP leakage, reputation damage and partner fallout. If these scenarios are unacceptable, stronger safeguards or an alternative architecture should be considered.

[24] *The PETs Adoption Guide is available at https://www.imda.gov.sg/assets/4301ab33-a760-4f02-afd3-c8371e6a15e2.pdf Its Annex 2 provides an implementation checklist across three phases: pre-implementation, during implementation and post-implementation. This guide focuses on pre-implementation phase only, with items either adapted from the pre-existing checklist, or added as FL-specific additions.*

Step 2: Align Partner and Internal Stakeholder

FL is multi-party by design. Hence, it depends not only on your organisation's readiness but on the readiness, willingness and trustworthiness of every participant in the federation.

From the checklist published in the PETs Adoption Guide (adapted for FL)

- **Metrics Alignment with Collaborating Partners** — Agree on clear ways to measure model training progress beyond basic business and privacy metrics:
  - Agree on how fast the shared AI model should learn and the number of update rounds needed to finish training.
  - Set minimum expectations for the amount and quality of data each partner should bring to the table.
  - Define the minimum performance level the final model should achieve for each individual participant.

- **Cross-Functional Management within Each Participant** — FL requires coordination across multiple corporate functions to address regulatory requirements and operational constraints. Organisations may:
  - form cross-functional teams (business, technology, data protection);
  - establish protocols for model updates and consent withdrawal;
  - address organisation-specific requirements and transparency about local data influence; and
  - ensure relevant data protection safeguards are put in place, especially if personal data is being processed.

FL-specific additions

- **Governance & Participation Framework** — Establish shared responsibility structures and formal agreements:
  - Document roles for coordination server operators and participants.
  - Define the collective project governance structure that would be used for decision-making.
  - Negotiate participation agreements covering data contribution requirements, computational commitments and exit provisions.

  Governance structures differ by deployment model:
  - **Cross-device:** Service providers implement automated privacy protection systems for millions of individual users through provider-managed rights protection.
  - **Cross-silo:** Institutional participants operate under formal agreements with peer-to-peer validation and established governance frameworks.

- **Model Governance** — Define ownership and usage arrangements for the global model:
  - Clarify which party owns the trained model and derived versions.
  - Establish policies for model access, redistribution and revocation.
  - Set deletion requirements for intermediate model components.

- **Incentive & Fairness Design** — Develop mechanisms for sustained participation and equitable value distribution. Examples include cost-sharing arrangements and sharing agreements.

- **Operational Protocols** — Define procedures for managing federation operations:
  - Establish dropout and failure recovery procedures.
  - Implement straggler handling mechanisms (timeouts, asynchronous updates).
  - Set up data validation and anomaly detection for corrupted updates.

FL-specific additions

The complexity level differs by deployment model:

- **Cross-device:** Simplified governance through platform-managed user terms, standardised incentives and automated operational protocols.
- **Cross-silo:** Extensive multi-organisational coordination requiring formal consortium governance, complex incentive negotiations and sophisticated operational protocols.

Step 3: Understand Internal Technology and Data Landscape

This step takes stock of each organisation's existing technology infrastructure and data landscape to identify gaps that need to be addressed before FL can be deployed effectively.

From the checklist published in the PETs Adoption Guide (adapted for FL)

- **System Compatibility and Infrastructure Provisioning** — Assess distributed computing capabilities and software compatibility:
  - Verify whether each participant can support the selected FL framework.
  - Maintain consistent versions across the federation.
  - Evaluate infrastructure requirements based on the deployment model.

Infrastructure requirements differ by deployment model:

- **Cross-device:** Assess edge device constraints (mobile CPU limitations, intermittent connectivity, storage restrictions) and evaluate the scalable coordination infrastructure capable of managing millions of heterogeneous devices with asynchronous updates and efficient bandwidth utilisation.
- **Cross-silo:** Evaluate enterprise-grade computational resources and verify framework compatibility across organisational IT environments.

From the checklist published in the PETs Adoption Guide (adapted for FL)	• **Data Standardisation** — Establishing unified preparation standards across participants may include the following:
	  • Aligning data dictionaries and feature definitions
	  • Standardising preprocessing pipelines and data format handling
	  • Ensuring temporal alignment of training datasets across participants
	• **Cost and Infrastructure Implication** — Assess resource requirements across participants and understand the cost structure. Refer to Annex G for details on cost drivers and develop optimisation strategies.

FL-specific additions	• **ML Maturity:** Build a strong centralised baseline first
	  • Make sure your ML can successfully solve the core problem under normal conditions before adding the extra technical complexity of FL.
	  • Use your standard model's results as a benchmark. This gives you a clear target to measure against, helping you quantify exactly how much accuracy you might lose or gain when you switch to a federated setup.
	• **Data Sample Alignment:** Check if participants have enough overlapping data to build a quality model. Identify any privacy constraints that might restrict how you match data (e.g. requiring Private Set Intersection).
	  • Find the common links: determine which shared identifiers (like Customer IDs, emails or timestamps) will be used to connect data across different participants.
	  • Set quality standards: Test how reliable these matching links are. Set a minimum acceptable match rate and decide how to handle unmatched data (e.g. delete it, fill in the blanks or use alternative AI techniques).

Step 4: Validate FL Solution

Before committing to full-scale implementation, it is important to validate that FL delivers the expected benefits in your specific context. This step covers how to test and verify that.

From the checklist published in the PETs Adoption Guide (adapted for FL)	• **Experimental Performance Validation** — Validate federated learning value through testing:
	  • Conduct POC experiments with synthetic and/or representative real-world data.
	  • Include benchmarking against centralised and local baselines to evaluate accuracy, convergence speed, and robustness.
	  • Demonstrate measurable benefits to justify coordination complexity.
	  • Explicitly evaluate expected accuracy trade-offs: FL may exhibit accuracy degradation compared to centralised training due to data heterogeneity, communication constraints and privacy-preserving mechanisms (e.g. differential privacy).
	  • Perform due diligence on solution providers' track records.
	• **Technical Capabilities** — Verify security and performance standards:
	  • Confirm adherence to established cryptographic standards.
	  • Request detailed technical documentation and architecture reviews.
	  • Obtain POC demonstrations and penetration testing reports.

FL-specific additions	• **Vendor and Platform Strategy** — Decide on a build versus buy approach:
	  • Evaluate open-source frameworks (control and portability vs specialist engineering requirements).
	  • Assess commercial platforms (reduced build effort vs dependency risks).
	  • Ensure contracts cover data handling on termination and migration support.

FL-specific additions	• **Audit and Compliance Monitoring** — Confirm comprehensive logging and alerting activities: • Verify the logging of model updates, aggregation events and participant actions. • Assess monitoring tools for performance degradation and anomalous behaviour. • Implement automated alerting for security incidents and compliance breaches.

2.4.3 Design Configuration

FL implementations involve numerous architectural and technical decisions. Instead of treating an FL solution as a rigid, one-size-fits-all blueprint, it helps to view it as a series of deliberate design trade-offs. Decisions about performance, privacy and complexity are interconnected, meaning a choice in one area will always affect the others. The table below outlines the key design considerations[25] to help guide your collaborative discussions with internal engineering teams or external solution providers.

Design Area	Typical Options	Considerations
Before Training		
Federation type *How is data distributed across participants?*	• Horizontal • Vertical • Transfer learning	Federation type is largely determined by how data is distributed. Horizontal FL is the most mature and widely supported. Vertical FL requires overlapping records between participants. Transfer learning may be used when both records and features differ.

[25] *For an illustrative example of how these design considerations can be applied in a reference deployment, refer to GovTech Singapore, Federated Learning Deployment Guide.* https://go.gov.sg/privacy-fl-guide-public

Design Area	Typical Options	Considerations
Before Training		
Coordination architecture *How are participants coordinated?*	• Centralised • Hierarchical • Decentralised	Centralised architectures are simpler but introduce a single coordination point. Hierarchical approaches improve scalability but add operational complexity. Fully decentralised approaches are less common in production systems. Architecture selection is influenced by multiple factors: 1. **Trust Model and Governance Structure:** The degree of mutual trust among participants and the availability of a trusted central coordinator. When a trusted central party exists, a centralised architecture (e.g. parameter server) is natural. When no single party is trusted, decentralised or peer-to-peer approaches become necessary. 2. **Privacy and Security Requirements:** The sensitivity of the data and the regulatory constraints governing it. Scenarios requiring strong cryptographic guarantees (e.g. healthcare or financial data) may necessitate architecture incorporating secure aggregation, homomorphic encryption or blockchain-based verification, regardless of whether a central server exists. 3. **Network Topology and Communication Constraints:** The physical and logical connectivity among participants. Hierarchical architecture suits scenarios with edge devices reporting to regional servers (e.g. IoT deployments), while fully decentralised topologies may be preferred when participants have comparable computer resources and direct peer connectivity.

Design Area	Typical Options	Considerations
Before Training		
Coordinator trust model *Can the coordinator see individual updates?*	• Trusted coordinator • Secure aggregation • Multi-party coordination	Secure aggregation[26] is commonly adopted to prevent the coordinator from viewing individual participant updates. More distributed trust models increase operational complexity. Many FL deployments assume a trusted or semi-trusted central server for orchestration and aggregation. Organisations are encouraged to explicitly define trust assumptions and consider alternatives (e.g. secure aggregation, TEEs) if trust is limited.
Model size *What model scale is appropriate?*	• Small (e.g. logistic regression) • Medium (e.g. gradient boosting) • Large models (e.g. LLMs)	In FL, model size affects communication costs at every participant and training round. Smaller models often reduce training overhead without major accuracy loss.
Model initialisation *Should training start from existing weights*	Random initialisation, pre-trained model	Using pre-trained weights can reduce the number of training rounds required and accelerate convergence.
FL framework *Which platform will be used?*	Open-source frameworks, commercial platforms	Framework choice should consider feature support, ecosystem maturity and the team's operational familiarity. Changing frameworks later can require substantial re-engineering.

[26] *In this guide, our use of "Secure Aggregation" refers to the security objective that can be achieved through various techniques, rather than a standalone technology category. Different technical implementations (e.g. MPC-based secret sharing, Paillier homomorphic encryption) offer distinct trade-offs in terms of computational overhead, communication costs and security guarantees.*

Design Area	Typical Options	Considerations
Before Training		
Complementary PETs *Which privacy mechanisms are required?*	Secure aggregation, differential privacy, trusted execution environments	These protections address different risks and are often deployed together. Each introduces different trade-offs in computational overhead and system complexity. Refer to **Annex F - Table 13 "FL Integration with Other PETs"** for more guidance on when to apply PET and the respective trade-off considerations.
Access control *Who can access the global model?*	Full access, restricted access, tiered access	Access policies should consider intellectual property protection, governance requirements and participant trust relationships.
During Training		
Aggregation method *How should participant updates be combined?*	• Simple averaging • Drift-corrected methods • Adaptive methods	Basic averaging is simple but may struggle with heterogeneous data. More advanced methods improve stability but may increase per-round compute costs.
Contribution weighting *Should participant updates be weighted?*	• Data-volume weighting • Equal weighting • Quality-based weighting	Weighting by dataset size may cause large participants to dominate the model. Alternative weighting schemes may be able to better balance contributions.
Participant selection *Who participates in each round?*	• All participants • Random subset • Scheduled selection	Using subsets can reduce communication overhead and improve training efficiency in large federations.

Design Area	Typical Options	Considerations
During Training		
Handling slow participants *What happens if some participants respond slowly?*	• Wait for all participants • Deadline-based participation • Asynchronous updates	Strict synchronisation can slow training. Deadline or asynchronous approaches improve efficiency but may introduce update variability.
Local training effort *How much local training occurs per round?*	• Few local epochs • Moderate epochs • Many epochs	More local training reduces communication rounds but may increase divergence between participant models.
Update compression *Can model updates be reduced before transmission?*	• Full precision • Sparse updates • Reduced precision	Compression techniques can significantly reduce network overhead with limited impact on model quality.
Across the Programme Lifecycle		
Data heterogeneity *How will differences in participant data be handled?*	• Model regularisation • Personalisation layers • Clustered training	Strategies vary depending on the level of heterogeneity. Personalised models may improve local performance but increase system complexity.
Fairness across participants *Does the model perform consistently across sites?*	• Global metrics only • Per-participant evaluation	Aggregate performance metrics may hide poor performance for certain participants. Monitoring site-level metrics helps detect such issues.

Design Area	Typical Options	Considerations
Across the Programme Lifecycle		
Model quality validation *How will model performance be assessed?*	• Central validation dataset • Local validation datasets • Combined evaluation	Evaluating across multiple participant datasets provides a more representative performance assessment.
Model lifecycle management *How will key lifecycle decisions be governed?*	• Release criteria • Training termination • Model provenance • Rollback capability • Retraining strategy	These choices are a part of any AI system but require explicit coordination and agreement among all participating parties in a federated setting. In a federated collaborative AI system, no single group can decide on their own when to launch the model, when to stop training or how often to update it. All participants in the FL system must agree on a shared set of rules.

Whilst these design considerations provide a useful reference, organisations need not address every aspect from scratch. A rich ecosystem of FL tools, benchmarks and tutorials already exists, allowing organisations to prototype and evaluate different approaches before committing to full-scale implementation. A comprehensive list of available resources is provided in **Annex I.**

2.5 RISK MANAGEMENT

FL reduces data exposure risks by keeping raw datasets local during training, minimising the chances of direct breaches involving sensitive or proprietary information. However, FL still carries a range of risks that organisations need to be aware of and manage. FL's distributed, multi-party architecture introduces distinct risks and coordination challenges that go beyond those of traditional single-organisation AI systems.

Organisations adopting FL should consider the following areas as part of their risk management approach:

- **Distributed governance and legal frameworks:** Establish clear contractual agreements covering participant responsibilities, decision authorities, data protection obligations and incident response coordination across all participating organisations.
- **Federated data processing:** Ensure data processing agreements account for multiple jurisdictions and define how data subject rights requests are handled across participants.
- **Collaboration and permission controls:** Define who can participate, under what conditions and the outputs they can access, with mechanisms to trace participation and actions for auditability.
- **Technical safeguards:** Implement controls to verify participant identity, protect model updates in transit, detect malicious or anomalous contributions, and manage the lifecycle of the global model securely.

Annex F provides a detailed description of FL risk types, along with the process and technical safeguards organisations can apply. At a high level, the four key risk areas are: (1) private information leakage through model updates, (2) corruption of the global model through malicious or faulty participant contributions, (3) disruption of the training process through denial-of-service or Byzantine attacks, and (4) supply chain compromise through tampered software or model artefacts. No single safeguard addresses all of these. Effective risk management requires layering multiple controls from the outset.

Rather than an afterthought, risk mitigation must run through the entire FL adoption roadmap.

FEDERATED ANALYTICS

FEDERATED ANALYTICS

While FL enables the collaborative training of ML models across distributed datasets, many organisations simply need statistical insights (such as counts, averages, correlations or regression results) rather than developing shared predictive models. FA supports this by enabling distributed statistical analysis without centralising raw data. Participating organisations execute analytical queries locally and share only aggregated outputs.

In practice, FA is also commonly used to support data discovery, quality assessment and harmonisation across participants, enabling organisations to understand distributed datasets before further collaboration or model training. This section provides a high-level overview of FA concepts.[27]

[27] *For further guidance on Federated Analytics implementation, refer to GovTech Singapore, Federated Analytics Guide. https://go.gov.sg/privacy-fa-guide-public*

3.1 WHAT IS FEDERATED ANALYTICS?

In FA, analytical queries are executed locally within each participating organisation, and only aggregated results are shared for analysis. Deployments typically involve a **coordinating service** that distributes queries, participants that execute computations locally, and an **aggregation mechanism** that combines results from multiple organisations.

A typical FA query-based workflow involves the following steps:

- **Query submission:** An authorised user submits an analytical query through a shared interface.
- **Query distribution:** The coordinating system distributes the query to participating organisations.
- **Local computation:** Each organisation executes the query locally on its dataset.
- **Aggregation:** Local outputs are returned and aggregated into a combined result.
- **Result release:** Governance checks may be applied before results are released.

3.2 TYPES OF ANALYTICS SUPPORTED

FA is most suitable for analytical tasks that can be expressed as distributed statistical computations. As results are derived from a single analytical workflow and do not involve iterative model training, these analyses typically require less coordination between participants when compared to FL. The table below illustrates types of analytical tasks that FA can support.

Table 3: Types of Analytics Supported by FA

Analytical task	Description	Example Outputs
Descriptive statistics	Basic summaries of distributed datasets	Counts, averages, percentiles, distributions
Trend analysis	Identifying patterns or changes over time across organisations	Monthly case counts, service utilisation trends
Correlation analysis	Measuring relationships between variables across datasets	Correlation coefficients, covariance matrices
Set comparisons	Identifying overlaps or differences between datasets held by different organisations	Number of shared entities, intersection sizes
Statistical modelling	Certain statistical models can be computed using aggregated intermediate statistics	Linear or logistic regression coefficients

3.3 CASE STUDIES

In practice, most deployments of FA focus on generating **aggregated statistical indicators across multiple organisations**, enabling shared insights without centralising sensitive data. FA is also commonly used in earlier stages of federated workflows to understand data distributions and assess readiness for collaboration.

3.3.1 Multi-Institution Healthcare Analytics

The NHS Federated Data Platform[28] illustrates that FA enables **system-wide analytics within a single jurisdiction** across many institutions, addressing fragmentation and supporting coordinated decision-making at scale. The platform connects 240+ NHS organisations through distributed instances, where each trust retains local control over its data while using standardised analytical tools. Analytical workflows are executed locally, with results aggregated to provide system-level visibility on metrics such as service utilisation, patient flows and capacity planning. This enables timely, cross-institution insights while ensuring sensitive patient data remains within each organisation's governance boundary.

3.3.2 Cross-Border Collaboration

FA can support international research collaborations where sensitive datasets are governed under different national regulatory frameworks. For example, a UK–US study involving NHS England, the US National Cancer Institute (NCI), and the UK Department for Science, Innovation and Technology (DSIT) enabled joint analysis of rare paediatric cancer data across jurisdictions[29]. Analytical queries were executed locally within each institution, with only encrypted intermediate results transmitted and securely aggregated using trusted execution environments and differential privacy safeguards. This allowed researchers to harmonise datasets, perform over 450 federated queries and generate insights on incidence and outcomes without centralising raw data.

[28] *NHS England. NHS federated data platform.* https://www.england.nhs.uk/digitaltechnology/nhs-federated-data-platform/

[29] *Government Digital Service UK. (2025). Using privacy enhancing technologies to enable international data sharing.* https://gds.blog.gov.uk/2025/10/09/using-privacy-enhancing-technologies-to-enable-international-data-sharing/

3.4 RISK MANAGEMENT

Although FA avoids many of the risks associated with FL (e.g. model poisoning or leakage from model updates), it still presents privacy and governance risks related to **analytical queries and the release of aggregated outputs**. Organisations should therefore evaluate FA platforms not only for statistical accuracy, but also for built-in query governance capabilities such as query whitelisting, rate limiting and audit logging.

Table 4: Technical Risk Management for FA

Risk	Description	Safeguards
Small-cell disclosure from aggregated outputs	Analytical results may reveal sensitive information if outputs relate to very small populations or contain minimal counts.	Apply aggregation thresholds and release only aggregated statistical outputs that meet defined disclosure controls.
Query chaining and reconstruction	Users may submit multiple related queries and compare outputs to infer sensitive information about specific individuals or small groups.	Restrict permitted query types, define governance rules on analytical scope and monitor query patterns.
Data heterogeneity across participants	Differences in data definitions, coding standards or schema structures across organisations may produce misleading or inconsistent analytical results.	Establish common data definitions, standardised schemas and an agreed analytical scope across participating organisations.
Participant collusion	Participants may combine outputs outside the system to infer sensitive information beyond what individual results reveal.	Establish governance agreements between participants and restrict analytical outputs to approved aggregated results.

CONCLUSION

FL and FA represent a meaningful shift in how organisations can collaborate on AI development and data analysis without compromising data privacy. As part of the broader suite of PETs, they embody the principle that data protection and data innovation are not opposing goals. Organisations can extract value from sensitive data assets, meet regulatory obligations and build trust with the individuals whose data they steward all at the same time.

In practice, FL and FA are most powerful when combined with complementary PETs such as differential privacy, secure multi-party computation, homomorphic encryption (HE), and trusted execution environments, each addressing different dimensions of the privacy and security challenge.

As this guide has outlined, FL and FA also introduce distinct architectural, operational and governance challenges that require deliberate planning across technical, legal and organisational dimensions. Organisations that invest in understanding these nuances early will be better positioned to unlock the value of distributed data assets responsibly. The adoption roadmap and risk management guidance presented here are intended to provide the structured thinking needed to navigate this journey with confidence.

ANNEX A: FEDERATION TYPES

When building privacy-preserving FL systems, understanding how data is distributed amongst participants is crucial. The partitioning scheme fundamentally shapes which privacy-preserving techniques are applicable and effective. There are three main approaches:

Horizontal Federated Learning (HFL)

In HFL, participants have datasets that share the same feature space or labels (columns) but differ in the sample IDs (rows). This is common in scenarios where organisations collect similar data about different populations. It's also called "sample-partitioned" or "example-partitioned" FL. For example, imagine several different hospitals, each with patient records that use the same data fields (symptoms, age, test results), but for different patients.

Vertical Federated Learning (VFL)

In VFL, participants have datasets with the same sample IDs (rows) but inhabit different feature spaces or labels (columns). This setup is typical in collaborations between organisations that hold complementary information about the same data subjects. It's also called "feature-partitioned" FL. For example, a bank and an e-commerce company have data on the same customers, but the bank holds financial information (credit scores, transaction history) and the e-commerce company holds purchase history.

VFL systems often require a preliminary step to match and find overlapping sample IDs across datasets. Compared to HFL, the privacy challenge of VFL extends beyond protecting feature values to securely identify which samples overlap between participants. Specialised secure Multi-Party Computation (MPC) protocols, such as Private Set Intersection (PSI), are essential here to perform secure entity alignment, as participants need to determine common entities without revealing their entire customer base. VFL Training is generally more complex than HFL.

In practice, sample IDs may not always exist or be precisely matchable across participants. Before selecting a federation type, organisations should assess the degree of sample alignment achievable — for example, through entity resolution, fuzzy matching on quasi-identifiers or address-based linkage. For benchmarks that address this challenge, see Wu et al. (2026).

Federated Transfer Learning (FTL)

FTL addresses scenarios where participants have datasets that differ in both features and samples, meaning there's limited or no overlap in either dimension. This represents the most challenging and general case. This technique uses shared data as a bridge to transfer AI models created for one domain to another. It is particularly valuable for cross-industry collaboration. For example, in collaboration between an insurance company and a real estate company, the insurance company can discover potential real estate customers from its own customers and utilise them for referrals. Conversely, referrals from the real estate company to the insurance company are also possible.

ANNEX B: FL ARCHITECTURAL APPROACH

FL is generally categorised into three primary architectural frameworks[30] based on how the global model is aggregated and how the network participants communicate.

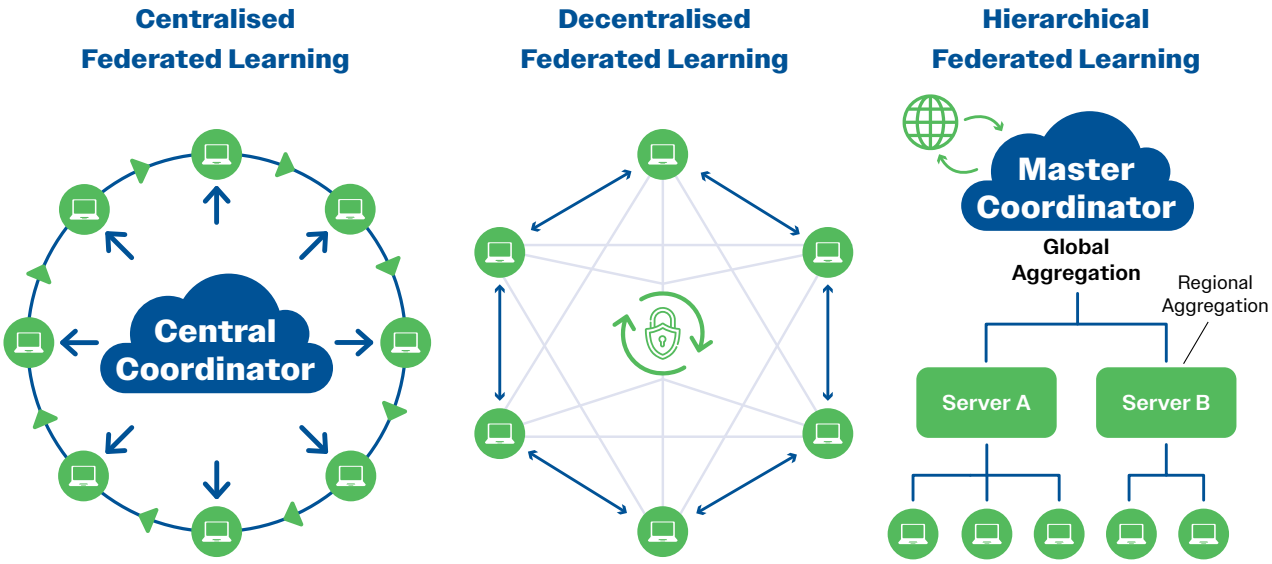
- **Centralised FL:** In a centralised framework, a single trusted authority (the "Server") manages the entire learning process, much like a conductor leading an orchestra. The individual participants (the "Clients") send only the summarised insights to the central authority to improve a master model. This type of architecture offers the highest level of administrative control, allowing for rigorous auditing and the implementation of advanced privacy protections like Differential Privacy at a single point. However, it also introduces a "honeypot" risk. If the central server is compromised, it could theoretically be used to orchestrate an attack on all connected participants, or an adversary could attempt to reverse-engineer private information from the aggregated summaries stored there. In practice, many production FL systems rely on a trusted or semi-trusted aggregation server. The assumption of a trusted central managing authority should be explicitly validated in threat models, governance frameworks and contractual agreements.

- **Decentralised FL:** Decentralised learning operates as a peer-to-peer network where no single entity holds ultimate authority. Instead of reporting to a central server, participants share their model improvements directly with one another, reaching a consensus on the best version of the model through collective agreement. This architecture eliminates the risk of a single point of failure; there is no central server to attack. However, privacy becomes harder to govern because every participant needs to interact with multiple others. Without a central gatekeeper, it is more difficult to detect a "poisoned" participant who might be intentionally sharing bad information to corrupt the model, making the system reliant on robust, often complex, peer-verification protocols.

[30] *Nasim, M. A. A., Soshi, F. T. J., Biswas, P., Ferdous, A. S. M. A., Rashid, A., Biswas, A., & Gupta, K. D. (2025). Principles and components of federated learning architectures. arXiv preprint.* https://doi.org/10.48550/arXiv.2502.05273

- **Hierarchical FL:** The hierarchical approach is a "middle-ground" strategy designed for massive, geographically spread networks. It introduces regional "middle-managers" that collect and refine information from local clusters before passing a condensed report up to a global authority. This model provides a "defence-in-depth" benefit where security breaches can often be contained within a single regional tier without compromising the entire global network. By processing data closer to where it is generated, it also reduces the "exposure window" during transmission. The primary challenge for governance here is the complexity of the chain of custody, as data summaries pass through multiple intermediary hands, requiring clear governance frameworks for whoever is responsible at each tier of the hierarchy.

Figure 4: FL Topologies



FL Architecture	Centralised	Decentralised	Hierarchical
Topology	Star	P2P/Mesh	Tree
Primary Coordinator	Central Server	None (Consensus-based)	Multiple (Master + Regional)
Single Point of Failure	High	Low	Moderate
Communication Cost	High (Long distance)	High (Peer density)	Low (Logical grouping)

Alternative Coordination Mechanisms

- **Blockchain-Based Aggregation:** In blockchain-based FL, a distributed ledger replaces the central aggregation server. Each participant submits encrypted local model updates to the blockchain, where smart contracts execute the aggregation logic (e.g. weighted averaging). The blockchain provides an immutable audit trail of all contributions and aggregation steps, ensuring transparency and tampering resistance. This approach is particularly suited to multi-party settings where no single participant is trusted to perform honest aggregation. However, it introduces additional latency and computational overhead from consensus mechanisms.

- **Homomorphic Encryption (HE) for Secure Peer-to-Peer Updates:** HE allows computations (e.g. addition, multiplication) to be performed directly on encrypted model parameters without decrypting them. In an FL context, participants encrypt their local model updates using HE before sharing them. An aggregator—or even other participants—can compute the aggregated model on the ciphertexts without ever seeing the plaintext updates, eliminating the need for a trusted aggregation party. A notable example is SecureBoost, a privacy-preserving gradient boosting framework for Vertical FL that uses Paillier HE to enable secure training across parties without requiring a central server. The trade-off of HE-based approaches is the significant computational cost, though recent advances in libraries such as Microsoft SEAL and OpenFHE have improved practical feasibility.

ANNEX C: FL AGGREGATION TECHNIQUES

Several aggregation techniques are available for FL, each with different trade-offs in accuracy, robustness and computational overhead. However, the choice of the aggregation method is often constrained by what the chosen platform natively supports. Organisations are encouraged to confirm which aggregation algorithms a platform supports, whether they are customisable and whether the platform allows pluggable aggregation for specialised use cases. Aggregation flexibility should be included as an explicit criterion in vendor evaluation and procurement. The techniques covered in this section represent a basic introductory overview and are not exhaustive. Readers seeking a more comprehensive treatment are encouraged to consult specialised FL literature.

Horizontal Federated Learning (HFL) Aggregation Techniques

In HFL, participating clients share the same feature space but possess different data samples. The following two basic sample techniques focus on aggregating complete model parameters or updates across clients.[31]

- **Federated Averaging (FedAvg[32]):** FedAvg performs a simple weighted average of client model parameters based on dataset sizes. It serves as the baseline FL approach due to its simplicity and minimal communication requirements. However, FedAvg struggles significantly with heterogeneous, non-independent and identically distributed (non-IID) data environments due to client drift and lacks the sophisticated mechanisms needed to handle data distribution disparities between participating clients.

- **Federated Proximity (FedProx):** FedProx enhances FL by incorporating a regularisation term that constrains local model updates to remain close to the global model, effectively mitigating client drift during local training. This approach delivers superior performance on heterogeneous data distributions and accommodates partial client participation with theoretical convergence guarantees, though it introduces additional computational overhead and requires careful hyperparameter tuning to optimise the regularisation strength.

[31] *Other specialised HFL approaches may also be relevant for advanced use cases, such as control-variate methods (e.g. SCAFFOLD), personalised methods (e.g. FedPer), matched-averaging methods (e.g. FedMA), contrastive methods (e.g. MOON), and prototype-based methods (e.g. FedProc), but are not covered in detail here.*

[32] *McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Artificial Intelligence and Statistics, 1273-1282. https://arxiv.org/abs/1602.05629*

Vertical Federated Learning (VFL) Considerations

In VFL, participating clients possess different feature sets for overlapping data samples. They require fundamentally different aggregation approaches that combine partial model embeddings or gradients across feature dimensions rather than averaging complete model parameters. VFL aggregation typically involves secure MPC for gradient aggregation, HE for feature combination, split learning architectures and specialised alignment techniques to combine heterogeneous feature representations whilst preserving data privacy. VFL aggregation requires specialised frameworks and expertise. For comprehensive coverage of VFL-specific aggregation methods, readers should refer to specialised VFL literature.[33]

[33] *Khan, A., ten Thij, M., & Wilbik, A. (2025). Vertical federated learning: a structured literature review. Knowledge and Information Systems, 67, 3205–3243. https://doi.org/10.1007/s10115-025-02356-y*

ANNEX D: FL TRAINING ROUND PROCEDURAL STEPS

Federated learning operates through an iterative training process coordinated by one or more servers or coordinating entities. The process begins with a one-time model initialisation, followed by repeated training rounds that continue until the predetermined stopping criteria are met or the entity monitoring the training determines that satisfactory performance has been achieved. Each training round consists of several key steps that form a continuous cycle, with coordination mechanisms facilitating communication between participants.

Step 0: Model Initialisation

Before federated learning begins, the global model must be initialised, and all participants' local data must be prepared and aligned. This involves two sub-processes. First, schema matching and harmonisation require all participants to agree on a common data schema, including feature names, data types and units of measurement. For Vertical FL, this includes agreeing on which features each party contributes, while for Horizontal FL, this ensures consistent feature definitions across participants. Second, data linkage and sample alignment involve participants performing privacy-preserving entity resolution (e.g. using Private Set Intersection) to identify overlapping samples where applicable. The alignment method, match rate and handling of unmatched samples should be documented, as this step is particularly critical for Vertical FL and Fuzzy VFL.

Step 1: Participant Selection and Model Distribution

At the start of each training round, participants are selected from the pool of available nodes, and the current model weights, along with training instructions, are distributed to them. The selection mechanism varies depending on the FL setting and timing paradigm. In synchronous FL, participants are selected at discrete round intervals. Asynchronous FL allows participants to join and contribute updates at any time without waiting for round coordination, whilst streaming FL enables continuous model updates as new data arrives. One-shot FL eliminates the iterative process entirely, requiring participants to compute and share updates only once. The distribution mechanism depends on the architecture and may involve broadcasts from a central server, peer-to-peer sharing or retrieval from a shared repository.

Step 2: Local Training

Upon receiving the model weights and training instructions, each selected participant independently computes an update to the model by executing the training process on their local data. This local computation typically involves running optimisation algorithms such as stochastic gradient descent on the participant's private dataset. The specific training approach may follow methods such as Federated Averaging, Federated Stochastic Gradient Descent or other federated optimisation algorithms. Crucially, raw data remains local to each participant, preserving data privacy whilst still enabling model improvement.

Step 3: Update Collection and Aggregation

After local training is complete, participants transmit their computed model updates to the designated aggregation point(s). In centralised settings, a server collects and aggregates updates; in decentralised settings, aggregation may occur through peer-to-peer protocols or hierarchical structures. The aggregation process combines individual updates to form a consolidated update to the model. To maintain efficiency, mechanisms may be implemented to handle slow or non-responsive participants, such as timeout policies or asynchronous aggregation schemes. This stage also serves as the integration point for various advanced techniques, including secure aggregation protocols for added privacy protection, compression algorithms to reduce communication overhead and the addition of calibrated noise combined with update clipping to achieve differential privacy guarantees.

Step 4: Model Synchronisation

The aggregated update is used to improve the model. In centralised architectures, the server updates the shared global model based on the aggregated update. In decentralised architectures, the updated model may be propagated through the network via consensus protocols or gossip algorithms, or each node may maintain its own version of the model that converges over time. It is worth noting that whilst aggregation improves generalisation, convergence challenges arising from non-IID data, partial participation or system heterogeneity may lead to lower accuracy as compared to centralised training. Careful tuning, validation and benchmarking are therefore required. Once synchronisation is complete, the process returns to Step 1, beginning a new training round with the improved model.

Note:

- The procedural steps in this annex are primarily oriented toward Horizontal FL. For Vertical FL, Fuzzy VFL or other federation types, additional data preparation steps (particularly data linkage and feature partitioning) are required. Organisations may adapt these steps according to the federation type being deployed.

- The descriptions above are oriented towards the traditional synchronous round-based approach for simplicity, as this serves as an introduction to FL concepts. However, modern FL deployments increasingly employ asynchronous, streaming and one-shot paradigms to address real-world constraints such as device availability, network latency and deployment timelines.

ANNEX E: TECHNICAL ATTACK VECTORS IN FL

FL was designed to enhance privacy by keeping raw data local on client devices and only sharing model updates to the system. However, researchers have discovered that FL is susceptible to privacy attacks.[34] The following attack vectors[35] illustrate how risks can arise. Understanding them is crucial for implementing appropriate defences and ensuring that FL systems truly protect user privacy. Threats in FL are generally categorised by the security principle they violate, including:

- **Confidentiality** (stealing data or models)
  These attacks aim to extract sensitive information from the system, violating the core promise of FL.

  - **Inference Attacks:** Even without accessing raw data, attackers can infer sensitive information by analysing the model updates (gradients) or the final model's behaviour.

    - *Membership Inference:* The attacker determines whether a specific individual's record was used to train the model. This is critical in healthcare; knowing a patient was in a "cancer treatment" dataset reveals their diagnosis.

    - *Property (Attribute) Inference:* The attacker deduces aggregate statistics or specific attributes of a participant's local dataset (e.g. inferring that a specific hospital's dataset lacks diversity in a certain demographic).

  - **Reconstruction Attacks (Gradient Inversion):** Unlike inference attacks, reconstruction attacks aim to recover the original raw training data (such as images or text) by reverse-engineering the gradients or parameters uploaded by a client. Techniques like Deep Leakage from Gradients (DLG) demonstrate that an attacker can reconstruct training inputs, pixel-by-pixel, by analysing the weight updates sent to the server.

  - **Model Extraction Attacks:** These target the intellectual property of the model owner. An outsider queries the global model (often via an API in the prediction phase) to extract the model's parameters or hyperparameters, effectively "stealing" the functionality of the trained AI without paying for the compute or data.

[34] *Zhao, J., Bagchi, S., Avestimehr, S., Chan, K., Chaterji, S., Dimitriadis, D., Li, J., Li, N., Nourian, A., & Roth, H. (2025). The federation strikes back: A survey of federated learning privacy attacks, defenses, applications, and policy landscape. ACM Computing Surveys, 57(9), Article 230.* https://doi.org/10.1145/3724113

[35] *Chen, Y., Gui, Y., Lin, H., Gan, W., & Wu, Y. (2022). Federated learning attacks and defenses: A survey. arXiv preprint.* https://doi.org/10.48550/arXiv.2211.14952

- **Integrity** (corrupting model behaviour)
  These attacks aim to corrupt the global model, causing it to make incorrect predictions. These are particularly dangerous because they can be "targeted"—meaning the model behaves normally for most inputs but fails on specific triggers.

  - **Data Poisoning:** A malicious client corrupts the training process by tampering with their local data before training begins. Dirty Label Poisoning refers to flipping labels (e.g. labelling images of cats as "dogs") to confuse the model. Clean-Label Poisoning refers to injecting malicious data that looks valid but degrades the model's ability to generalise.

  - **Model Poisoning:** Instead of manipulating data, the attacker (a malicious client) directly alters the model parameters or gradients before uploading them to the server. Research indicates that model poisoning is often more effective than data poisoning because the attacker has direct control over the "learned" insights being sent to the global model.

  - **Data Source Poisoning:** A subtler threat exists when training data is sourced from the Internet or shared public datasets. In FL environments, poisoned samples may already be interspersed within otherwise clean public sources that multiple clients access independently. Even well-intentioned participants can unknowingly contribute compromised data, as detection is difficult without dedicated safeguards. Unlike traditional data poisoning, this can affect the global model even when all participating clients are benign.

  - **Backdoor Attacks:** This is a sophisticated form of poisoning where the attacker injects a "trojan" or trigger into the model. The model performs normal tasks accurately but behaves maliciously when it encounters a specific trigger (e.g. an autonomous vehicle ignoring a stop sign only if the sign has a specific yellow sticker on it).

- **Availability** (preventing model training)
  These attacks aim to disrupt the training process itself, preventing the model from converging or making the system unusable.

  - **Byzantine Attacks:** Malicious participants (Byzantine workers) arbitrarily upload random or destructive model updates to the server. The goal is to maximise the error rate of the global model or ensure it never finishes training.
  - **Denial of Service (DoS):** Attackers may flood the communication channels or target the aggregation server to halt the FL process entirely.

Table 7: Classification of FL Attacks

Security Principles	Attack Type	Primary Phase
Confidentiality	Inference	Training/Prediction
	Reconstruction	Training
	Model Extraction	Prediction
Integrity	Data Poisoning	Training
	Model Poisoning	Training
	Data Source Poisoning	Training
	Backdoor	Training
Availability	Byzantine	Training
	Denial of Service	Training/Prediction

ANNEX F: MANAGING FL RISKS

This annex outlines the key risk types associated with FL systems, followed by the process safeguards and technical safeguards[36] that organisations may adopt to mitigate those risks.

The controls described here are specific to FL and assume that organisations already have baseline standards for data protection, cybersecurity, incident response and AI governance.

Types Of Risk Concerns

Various risks may arise through different types of technical attacks or system failures affecting the training process, model behaviour or supporting infrastructure. The following section focuses on the **risk outcomes and system impact** that organisations should consider when deploying FL systems.

A Training Data Leakage

Even though raw data does not leave the local data silos, the model updates contain information about the data. The information from the updates can be used to:

- **Reconstruct training data:** Reverse engineer the updates to create the actual images, text or records that were used for training.
- **Determine who is in the training set:** Determine where a specific person's data has been used to train the model. This risk is particularly relevant when models are trained on highly sensitive personal data, where confirming that an individual's record was part of the training set could reveal private information about them.
- **Infer aggregate properties:** Infer aggregate statistical characteristics of a participant's dataset (e.g. demographic distribution of customers or prevalence of certain conditions), despite the raw data remaining local to the participant.

[36] *For an illustrative example of how to implement such technical controls, refer to GovTech Singapore, Federated Learning Deployment Guide.* https://go.gov.sg/privacy-fl-guide-public

B Model Integrity Compromise

A dishonest participant may attempt to manipulate the FL process by submitting malicious or falsified updates. This can distort the model's behaviour, degrade its accuracy or introduce hidden vulnerabilities.

- **Poisoning attacks:** A bad actor can deliberately send false or bad information that degrades the AI's accuracy for everyone or produce false information in generative AI (e.g. GPT models).
- **Backdoor attacks:** The AI can be secretly programmed to behave incorrectly in specific situations. For example, attackers can create vulnerabilities in generated code. This is nearly invisible under normal testing.
- **Fake identities (Sybil attacks):** An attacker creates multiple fake participants in the FL process to increase their influence over model training. By controlling many participants, the attacker can bias the model updates and steer the model towards outcomes that are favourable to them.

C Training Process Disruption

In cross-device FL, attackers may attempt to disrupt the training process by overwhelming participating devices or the central coordinating server. For example, attackers may launch denial-of-service (DoS) or distributed denial-of-service (DDoS) attacks by flooding the system with excessive requests or traffic. This can cause legitimate participants to drop out of training rounds or prevent the central server from coordinating the learning process effectively.

D Supply Chain Compromise

The FL system may also be vulnerable to attacks through components in the software and operational environment used during training.

- **Compromised software dependencies:** Libraries, frameworks or packages used to train or run the FL system may be tampered with or contain hidden vulnerabilities. Experimental or non-production-ready libraries specifically may include security weaknesses or backdoors. If participants install compromised software on their devices or servers, malicious code could affect the integrity of the training process and potentially impact all participants.
- **Sensitive information in logs:** System logs used for monitoring, debugging, or performance analysis may inadvertently capture sensitive information such as model inputs, intermediate outputs or metadata. If these logs are not properly protected, they could expose information that bypasses the privacy protections built into the federated learning protocol itself.

Process Safeguards for Managing FL-Specific Risks

The following guidance highlights key FL-specific considerations that organisations should evaluate as supplements to existing data protection and model AI governance practices. These measures are not exhaustive, but they reflect the current understanding of FL risks and may evolve as implementation experience grows.

1. Federated Threat Modelling and Risk Assessment

- Define and maintain a FL threat model, including assumed attacker capabilities (e.g. malicious participants, collusion, cross-organisational inference attacks) and trust assumptions between participants.
- Establish acceptable risk levels and document residual risks that cannot be mitigated through technical controls alone.

2. Distributed Governance

- Establish contractual agreements defining governance structure, liability allocation and each participant's responsibilities, including:
  - decision authorities for participant admission, exit and model releases;
  - federated governance roles including cross-organisational security oversight, operational coordination;
  - data protection obligations, minimum security requirements and prohibited activities; and
  - breach notification requirements and incident response coordination procedures.

3. Federated Data Processing

- Establish clear data processing agreements that coordinate the legal basis and purpose across multiple jurisdictions and organisations.
- Define procedures for handling data subject rights requests that may span multiple participants in the federated system.

4. Collaboration and Permission Controls

- Define how participants interact within the federated system, including:
  - which parties can collaborate and under what topology (e.g. bilateral, hub-and-spoke, many-to-many);
  - who can initiate training, contribute updates and access intermediate or final outputs;
  - the rules governing cohort sizes, feature usage, join conditions and output restrictions;
  - how participants are onboarded, approved and managed across training rounds; and
  - which participants can access which outputs, and whether the outputs differ across participants.
- Mechanisms to trace participation, actions taken and outputs generated for auditability.

Technical Controls by Risk Type

Table 8: Technical Safeguards for FL Systems

Risk	Verify who's participating	Hide individual participant's updates	Catch bad updates from individual participant	Limit what the global model reveals	Control the global model's lifecycle
Private Information Leakage		●		●	
Corruption of the AI Model	●		●		●
Disruption of the Training Process	●		●		●
Supply Chain Compromise	●				●

● *Primary safeguard* ● *Supporting safeguard*

Risk 1: Private Information Leakage

In FL, raw training data remains within each participant's environment. However, model updates shared during training still contain information derived from that data. If exposed, these updates (or the final trained model) may reveal sensitive information.

Safeguards therefore focus on hiding individual updates and limiting what the trained model reveals.

Table 9: Technical Safeguards to Address Risk 1

Safeguard	What it protects against	Implementation examples
Hide individual updates	Prevents the coordinator or other parties from observing individual participant updates that may reveal training data	Use secure aggregation so only the combined update is visible. Protocols such as the Bonawitz secure aggregation rely on pairwise masking or secret sharing. Aggregation methods should tolerate participant dropouts and prevent replay of updates across rounds.
Limit what the model reveals	Reduces the extent to which the trained model reflects information about specific individuals in the training data	Apply differential privacy during training, typically using DP-SGD, where gradients are clipped and noise is added. Track the privacy budget (ε) across training rounds and stop participation when the allocated budget is exhausted. Where required, machine unlearning[37] techniques can be applied post-training to remove the influence of specific data contributions.
Privacy testing	Verifies that privacy safeguards work as expected before releasing a model	Conduct simulated attacks such as membership inferences or gradient reconstruction tests. These tests help detect unintended information leakage from the model.

[37] *Machine unlearning is an emerging area of research.*

Risk 2: Corruption of the AI Model

Participants in FL contribute model updates during training. If a participant submits malicious or abnormal updates (intentionally or unintentionally), the resulting model may degrade in performance or contain hidden behaviours. These attacks are often referred to as model poisoning or backdoor attacks.

Safeguards therefore focus on detecting suspicious updates and controlling how trained models are validated and released.

Table 10: Technical Safeguards to Address Risk 2

Safeguard	What it protects against	Implementation examples
Catch bad updates	Detects malicious or abnormal updates that could corrupt the model	Monitor update magnitudes and distributions. Apply gradient clipping to cap update size and flag suspicious contributions for review.
Robust aggregation	Reduces the influence of extreme or malicious participant updates	Use aggregation methods designed to tolerate outliers, such as trimmed mean, coordinate median or other Byzantine-resilient algorithms.
Participant verification	Prevents attackers from creating multiple fake participants to influence training (Sybil attacks)	Require verified identities and participant registration controls. Implement certificate-based authentication or device attestation where appropriate.
Control the model's lifecycle	Prevents compromised models from reaching production systems	Use automated validation pipelines that test model accuracy, detect hidden behaviours, verify fairness across participants and confirm privacy budgets remain within approved limits before deployment.
Machine unlearning	Removes the influence of specific participants or compromised data from a trained model without requiring full retraining	Apply unlearning techniques to remove contributions from specific participants (e.g. poisoning or withdrawal). This may involve partial retraining, update tracking or approximation-based methods depending on the model.

Risk 3: Disruption of the Training Process

FL systems rely on communication between participants and a coordinating service. Attackers may attempt to disrupt the training process by targeting participating devices, network connections or the coordinating infrastructure. Many of these threats are not unique to FL. They are common to any distributed ICT system, and standard mitigations (e.g. traffic filtering, rate limiting and infrastructure monitoring) apply. However, FL systems introduce additional operational considerations beyond baseline protection measures.

Table 11: Technical Safeguards to Address Risk 3

Safeguard	What it protects against	Implementation examples
Verify who's participating	Prevents unauthorised devices from joining training rounds	Require strong authentication mechanisms such as **mutual TLS** and certificate-based identity management.
Resilient aggregation	Ensures training can continue even when some participants disconnect	Use aggregation protocols that tolerate partial participation so rounds can complete despite participant dropouts.
Coordinator resilience	Prevents the training system from failing if the coordinating service becomes unavailable	Implement model checkpointing and failover mechanisms so training can resume after infrastructure disruptions.

Risk 4: Supply Chain Compromise

Federated learning requires distributing software, models and configuration artefacts across multiple participant environments. If any component in this supply chain is compromised (e.g. through tampered software or malicious updates), the integrity of the training process may be affected.

Safeguards therefore focus on verifying software integrity and maintaining tamper-resistant records.

Table 12: Technical Safeguards to Address Risk 4

Safeguard	What it protects against	Implementation examples
Verify participating software	Ensures participants run approved versions of the training software	Use signed software packages or **remote attestation** to verify that participant environments are running authorised code.
Control the model lifecycle	Prevents modified or untested models from being deployed	Require controlled release pipelines where models need to pass validation checks before deployment.
Keep tamper-proof records	Enables auditing and investigation of the training process	Digitally sign model artefacts and maintain provenance records including code versions, participating organisations, training rounds and validation results.
Secure logging	Prevents operational logs from exposing sensitive information	Record aggregate statistics and operational events rather than individual participant updates. Protect logs using append-only or digitally signed logging systems.

Complementary Privacy Enhancing Technologies (PETs)

In practice, FL deployments combine multiple PETs to balance privacy, utility, trust and performance. The table below summarises common approaches and their trade-offs. These are typically used together and selected as part of an integrated system design, rather than as standalone controls.

The following table summarises several commonly used approaches and their trade-offs:

Table 13: FL Integration with Other PETs

Technology	What It Does	When to Use It
Differential Privacy	Adds carefully calibrated random noise so that no single person's data has a meaningful influence on the model	When a quantifiable privacy guarantee is required. Commonly used in many FL deployments.
Secure Multi-Party Computation	Splits the aggregation process across multiple independent servers so that no single party can see individual contributions	When no single coordinator can be fully trusted, such as when competing organisations collaborate without a trusted intermediary.
Fully/ Partially Homomorphic Encryption	Allows computation to be performed directly on encrypted data; the coordinator can combine updates without decrypting them	When privacy needs to be preserved without relying on specialised hardware. Currently practical mainly for simple aggregation tasks.
Trusted Execution Environments	Performs aggregation inside a hardware-protected environment that even the server operator cannot inspect	When the coordinator's infrastructure cannot be fully trusted. It provides hardware-level isolation during computation.

ANNEX G: COST CONSIDERATIONS

This section outlines factors that influence the engineering cost of FL deployments and practical considerations for managing these costs.[38] The discussion focuses on **compute, communication and security controls**. Depending on the deployment setup, engineering costs may represent a substantial portion of total programme costs, while in other cases, legal, governance, organisational or programme coordination costs may be more significant. Those costs are not included here and should be considered separately.

1. Compute and Infrastructure

FL distributes model training across multiple participating organisations. Each participant trains the model locally on its own infrastructure and periodically sends model updates to a coordinating server. As a result, compute costs arise both at the coordinating environment and across participant environments.

Table 15: Cost for Compute and Infrastructure

Cost Item	What Drives It Up	How to Bring It Down
Model size	Training cost generally scales with the number of model parameters. Because training occurs at each participant site and across multiple rounds, FL can amplify compute requirements compared to centralised training.	Evaluate smaller models using a centralised baseline before deploying FL. In practice, the cost difference between small and large models can be substantial across many training rounds.
Number of rounds	The total cost is approximately the cost per round multiplied by the number of training rounds. Non-IID data can increase the number of rounds required for convergence. If data heterogeneity is not well characterised, cost estimates may be uncertain.	Aggregation methods designed for heterogeneous data (e.g. drift-corrected approaches) can reduce the number of required rounds (see Annex C). Initialising from a pre-trained model and tuning training parameters early may reduce training time and compute requirements.

[38] *For an illustrative example of how to consider and manage engineering costs in practice, refer to GovTech Singapore, Federated Learning Deployment Guide.* https://go.gov.sg/privacy-fl-guide-public

Table 15: Cost for Compute and Infrastructure

Cost Item	What Drives It Up	How to Bring It Down
Participant-side compute	Each participant typically performs a full local training pass in every round. Under-provisioned sites may become bottlenecks for the entire training process.	Clarify compute responsibilities across participants early in the programme and assess whether each site can support the required training workload. Where feasible, allow participants to use elastic or on-demand compute resources during training rounds instead of permanently provisioned infrastructure.
Retraining	Each retraining cycle incurs costs similar to the initial training run. Frequent retraining (e.g. monthly) can significantly increase the total compute expenditure.	Retrain only when data distributions change meaningfully. Where possible, define fixed retraining schedules in advance, which are generally easier to plan and budget for than ad hoc retraining triggers.
Local training effort	Increasing the number of local training passes increases compute requirements for each participant. Excessive local training may also introduce divergence across participants.	Begin with a small number of local passes (e.g. one to three) and tune during pilot deployments. The appropriate level depends on the degree of data heterogeneity across participants.
Confidential computing	Hardware-isolated environments (e.g. trusted execution environments) typically involve higher instance costs and increased engineering complexity.	Such environments are generally required only when the coordinating party cannot be fully trusted. If secure aggregation and differential privacy are sufficient, standard compute infrastructure may be more cost-effective. Organisations may assess TEE readiness during cloud infrastructure planning, including regional availability, workload overhead, and whether attestation is automated or requires manual integration. Examples include Azure Confidential Computing and AWS Nitro Enclaves.

2. Communication

FL relies on repeated communication between participants and the coordinating server during training. In each round, the coordinator distributes the current global model, participants train locally on their data, and updated model parameters are returned for aggregation. Because this process repeats over many rounds, network transfer and coordination overhead can become a significant component of system cost.

Table 16: Cost for Communication

Cost Item	What Drives It Up	How to Bring It Down
Per-round bandwidth	In each training round, participants upload model updates (often similar in size to the model itself) and download the updated global model. Total transfer volume therefore scales with model size, number of participants and number of rounds.	Compression and update-sparsification techniques[39] can significantly reduce transfer size and are typically supported in standard FL frameworks.
Slowest participant	In synchronous training, each round can only proceed once all participants finish computing their updates. A slower participant can therefore delay the entire training process.	Introduce time limits for each round or train using a subset of participants per round to reduce the impact of slower nodes.
Cross-border transfer	Intercontinental network links introduce higher latency, and cloud providers may charge cross-region data transfer (egress) fees.	Hosting participants within the same cloud provider or region may reduce egress costs. Regional coordinators can also aggregate updates locally to reduce cross-border traffic.

3. Security and Privacy Controls

FL deployments require standard system security controls such as authentication, encrypted communication and audit logging. These baseline measures are common to most distributed systems and are typically required regardless of whether FL is used.

Additional costs may arise from PETs that protect model updates and training processes. These safeguards are often used together rather than as alternatives, depending on the sensitivity of the data and the level of trust between participants.

[39] *Compression techniques reduce the size of model updates before they are transmitted (e.g. by using lower-precision representations). Update sparsification techniques reduce the amount of data sent by transmitting only the most significant model updates rather than the full update.*

ANNEX H: ADDRESSING DATA HETEROGENEITY

FL operates across multiple participants whose datasets often differ due to their own users, populations and operational environments, unlike centralised ML where data is pooled.

As a result, federated systems encounter heterogeneity across multiple dimensions, including statistical differences (commonly seen as **non-IID data**), as well as variations in context, system constraints, annotation standards and data modalities (e.g. text, image). Managing this heterogeneity is essential for stable training and good model performance.[40]

While heterogeneity spans multiple dimensions, this annex focuses on statistical heterogeneity (non-IID data), which is the most widely studied and directly impacts model convergence. Other forms (e.g. system, modality, annotation differences) typically require broader design considerations.

1. Types of Federated Architectures and Where Heterogeneity Appears

FL Architecture	Data Distribution Pattern	Typical Sources of Heterogeneity	Example
Horizontal Federated Learning	Same features, different records across participants	• **Label skew** — class distributions differ across participants • **Feature distribution skew** — same variables have different statistical properties • **Quantity skew** — dataset sizes vary significantly	Five hospitals each hold chest X-ray images with the same schema, but for different patients. One hospital may see mostly severe cases while another sees mostly routine scans.

[40] *For practical considerations on assessing and addressing data heterogeneity in practice, refer to GovTech Singapore, Federated Learning Deployment Guide.* https://go.gov.sg/privacy-fl-guide-public

FL Architecture	Data Distribution Pattern	Typical Sources of Heterogeneity	Example
Vertical Federated Learning	Same entities, different features held by each participant	• **Feature coverage differences** — some attributes exist only at certain organisations • **Data quality differences** — missing values or inconsistent feature definitions • **Entity overlap limitations** — only a subset of entities appears in all datasets	A bank holds transaction histories while a telecommunications company holds call-record metadata for overlapping customers.
Federated Transfer Learning	Different features and different samples, with limited overlap	• **Domain shift** — data originates from different environments or populations • **Task mismatch** — related but distinct prediction objectives • **Label scarcity** — some participants have few labelled examples	A hospital in one country trains a model to detect disease presence, while another institution trains it to predict disease severity using a different dataset.

2. What Non-IID Data Looks Like in Practice

In FL, each participant's dataset reflects its own operational context. Consequently, the IID assumption rarely holds. The following are examples of how non-IID data arise in practice:

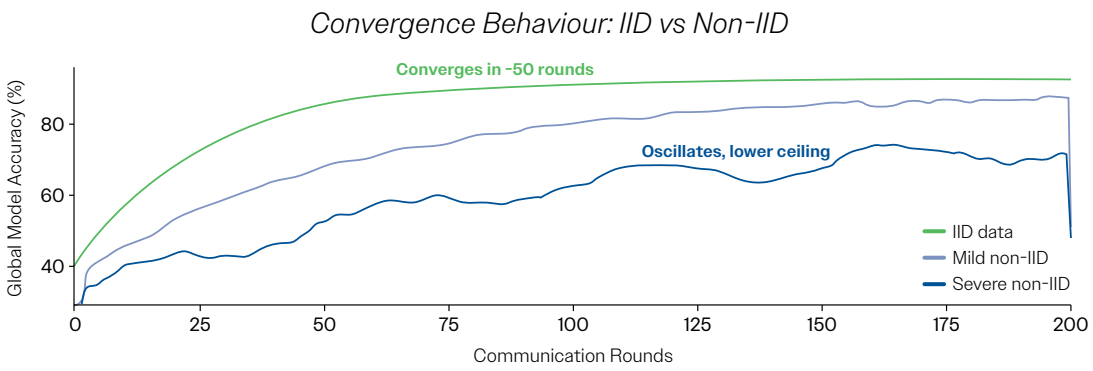
Type of Heterogeneity	Description and Example
Label skew	Participants observe different proportions of outcome labels. For example, Bank A may see 3% fraud cases while Bank B sees 0.1% fraud. Similarly, an oncology hospital may see mostly cancer cases, while a general hospital sees mostly normal scans.

Type of Heterogeneity	Description and Example
Feature skew	Participants have the same features but different data distributions. For example, the feature transaction amount may follow different patterns for a domestic bank compared to a cross-border payments provider. Behavioural signals such as keystroke timing may also differ across populations in different countries.
Quantity skew	Participants contribute very different amounts of data. For example, one organisation may have 200,000 records, while another may only have 500 records. Without adjustments, the larger dataset may dominate model updates.
Temporal skew	Participants train on data from different time periods. For example, one organisation may use last week's transaction data, while another uses data from the previous quarter, which may reflect different patterns or conditions.
Missing classes	Some participants may have no examples of certain categories. For example, a specialised cardiac hospital may have no dermatology cases. As a result, local training at that site does not learn patterns for those categories.

3. Why Data Heterogeneity Matters

In FL, each participant trains a model using only its **local dataset**. This means the model updates produced at each site reflect the patterns in that participant's data. When participants' datasets differ significantly in size, distribution or coverage, their local updates may move the model in **different directions**. When these updates are combined, the global model may take longer to converge, become unstable during training or perform unevenly across participants.

Figure 5. Effect of Data Heterogeneity on FL Convergence



When participant datasets follow similar distributions (IID), training converges smoothly. As data heterogeneity increases (non-IID), convergence becomes slower, training may oscillate and the achievable accuracy may be lower.

Impact	Description
Slower convergence	Tasks that converge in tens of training rounds under IID conditions may require hundreds of rounds when data distributions differ significantly across participants. More training rounds increase communication overhead and infrastructure cost.
Client drift	Local models may move away from the global optimum because each participant trains on its own data distribution. When these divergent updates are averaged, the resulting model may perform worse than some individual local models.
Degraded accuracy	The global model should represent a compromise across different data distributions. If these distributions differ substantially, the resulting model may perform poorly across all participants.
Fairness failures	Participants with larger datasets may dominate the training process, biasing the model toward their data distribution. Missing classes for certain participants can also degrade performance for those categories globally.
Unstable training	Model accuracy may fluctuate significantly between training rounds. This makes it harder to determine when the model has converged or whether a drop in performance indicates a real issue.

4. Mitigations

Data heterogeneity is common in FL deployments. While it may affect convergence and model performance, established techniques can often mitigate these effects, and heterogeneous datasets do not necessarily prevent useful models from being trained. The approaches below are commonly used to improve stability and performance when training across heterogeneous datasets:

Approach	Description
Better aggregation	Drift-corrected methods (e.g. FedProx, SCAFFOLD) help address the differences between local datasets and often converge in fewer rounds under non-IID conditions. Further discussion of aggregation techniques is provided in **Annex C**.
Server-side optimisers	Techniques such as FedAdam and FedYogi apply adaptive learning rates when combining client updates. This makes training less sensitive to conflicting gradients across participants.
Personalisation	A shared global model can be trained collaboratively and then fine-tuned locally for each participant. This allows each organisation to adapt the model to its own population, though it requires managing multiple models.
Data-level techniques	Small, shared datasets or synthetic data can help regularise training across participants. Local data augmentation and stratified participant selection can also help address label gaps or imbalances.
Clustering	Participants with similar data distributions can be grouped together and trained as clusters. Within-cluster heterogeneity is typically lower, which can improve convergence.
Architecture-Specific Approaches	
For vertical FL	Standardise feature representations, handle missing features carefully and invest in high-quality entity alignment. Systems should also account for temporal misalignment between datasets.
For federated transfer learning	Assess domain similarity before deployment, progressively unfreeze model layers during training and draw from multiple source participants where possible.

ANNEX I: FL QUICK START RESOURCES AND TOOLS

FL Frameworks

- **Flower (flwr):** A framework-agnostic FL framework supporting PyTorch, TensorFlow, and JAX. GitHub: https://github.com/adap/flower
- **NVIDIA FLARE:** Enterprise-grade FL platform with built-in privacy mechanisms. GitHub: https://github.com/NVIDIA/NVFlare
- **TensorFlow Federated (TFF):** Google's open-source framework for ML on decentralised data. GitHub: https://github.com/google-parfait/tensorflow-federated
- **FedML:** Supports diverse FL paradigms (cross-silo, cross-device) with MLOps integration and built-in benchmarking capabilities for non-IID datasets across simulation, distributed, and on-device modes. GitHub: https://github.com/FedML-AI/FedML
- **PySyft:** Privacy-preserving ML framework by OpenMined, combining FL with differential privacy and secure computation. GitHub: https://github.com/OpenMined/PySyft

FL Benchmarks

- **NIID-Bench:** A comprehensive benchmark for evaluating FL algorithms under various non-IID data partition strategies. Published at ICDE 2022. GitHub: https://github.com/Xtra-Computing/NIID-Bench
- **VertiBench:** A benchmark suite for vertical federated learning, supporting algorithms including SecureBoost, C-VFL, FedTree, and others. Published at ICLR 2024. GitHub: https://github.com/Xtra-Computing/VertiBench
- **WikiDBGraph:** A data management benchmark suite for collaborative learning over database silos, featuring 100,000 real-world databases interconnected by 17 million edges. Published at ICDE 2026. GitHub: https://github.com/Xtra-Computing/WikiDBGraph
- **LEAF:** A modular benchmarking framework for learning in federated settings, providing realistic non-IID datasets (FEMNIST, Shakespeare, CelebA, Reddit, Sent140). Published at NeurIPS 2019 Workshop. GitHub: https://github.com/TalwalkarLab/leaf
- **FedScale:** A scalable and extensible FL benchmarking suite with real-world datasets and system heterogeneity profiles for cross-device FL evaluation. Published at ICML 2022. GitHub: https://github.com/SymbioticLab/FedScale
- **FLamby:** A cross-silo FL benchmark suite focused on healthcare, providing naturally partitioned medical datasets (e.g., pathology, dermatology, ECG) with realistic hospital-level data splits. Published at NeurIPS 2022. GitHub: https://github.com/owkin/FLamby
- **pFL-Bench:** A comprehensive benchmark for personalised federated learning, evaluating over 10 pFL methods across diverse datasets and heterogeneity settings. Published at NeurIPS 2023. GitHub: https://github.com/alibaba/FederatedScope/tree/master/benchmark/pFL-Bench

Tutorials and Getting Started Guides

- Flower tutorial series: https://flower.ai/docs/framework/tutorial-series-what-is-federated-learning.html
- TFF tutorials (image classification, text generation): https://www.tensorflow.org/federated/tutorials/tutorials_overview
- NVIDIA FLARE documentation: https://nvflare.readthedocs.io/en/2.6/getting_started.html
- GovTech Singapore, Federated Learning Deployment Guide: https://go.gov.sg/privacy-fl-guide-public
- Starter Kit for Testing LLM-Based Applications for Safety and Reliability: https://www.imda.gov.sg/-/media/imda/files/about/emerging-tech-and-research/artificial-intelligence/starter-kit-for-testing-llm-based-applications-for-safety-and-reliability.pdf

ACKNOWLEDGEMENTS

The PDPC and Infocomm Media Development Authority (IMDA) would like to express sincere appreciation and acknowledgment for all the valuable feedback received from the following organisations (in alphabetical order):

- Ant International
- A*STAR Institute of Advanced Intelligence and Computing (A*STAR IAIC)
- Chinese University of Hong Kong (CUHK), Department of Computer Science and Engineering
- Duality
- Google
- InfoSum
- Nvidia
- Nanyang Technological University (NTU), College of Computing & Data Science
- National University of Singapore (NUS), School of Computing
- Pryvx
- SingHealth and Duke- NUS Artificial Intelligence in Medicine Institute (AIMI)

Jointly developed by

Supported by