Supplementary Material: FL-EHDS: A Privacy-Preserving Federated Learning Framework for the European Health Data Space

Fabio Liberti
Department of Computer Science
Universitas Mercatorum, Rome, Italy
fabio.liberti@unimercatorum.it
ORCID: 0000-0003-3019-5411

***Abstract*—This document provides supplementary material for the FL-EHDS paper, including complete algorithm pseudocode for all framework components, extended experimental figures, detailed algorithm comparison analysis, advanced FL paradigm descriptions, infrastructure component specifications, extended EHDS interoperability details, and clinical imaging experiment configurations. The open-source reference implementation (∼40K lines, 159 modules) is available at https://github.com/FabioLiberti/FL-EHDS-FLICS2026.**

I. PRISMA Flow Diagram

II. Algorithm Pseudocode

This section provides formal algorithmic descriptions of all FL-EHDS framework components. Each algorithm is presented with: (1) a contextual explanation of *why* the component is needed in the EHDS regulatory context; (2) the formal pseudocode; and (3) practical considerations for deployment. The algorithms are organized following the data flow through the three-layer architecture: governance validation (Layer 1), privacy-preserving aggregation (Layer 2), and local data processing (Layer 3).

Reading guide: Algorithms S1–S4 form the core FL-EHDS training pipeline. Algorithms S5–S6 address EHDS-specific challenges (non-IID data and citizen opt-out). Algorithms S7–S8 handle data preprocessing and privacy budget management.

A. FedAvg with EHDS Compliance

Algorithm S1 presents the core federated averaging procedure adapted for EHDS regulatory requirements, operating in a client-server architecture where the central aggregator coordinates training across distributed hospital nodes within a Secure Processing Environment.

Key Design Decisions:

- **ValidatePermit**: Before each round, the HDAB-issued permit is verified against temporal bounds and Article 53 permitted purposes.
- **SelectParticipants**: Configurable client selection—full participation or sampling for large federations.
- **FilterOptedOut**: Records from citizens who exercised Article 71 opt-out rights are excluded *before* gradient computation.
- **Weighted Aggregation**: Gradients weighted by local dataset size (n_h), following original FedAvg [13].
- **ClipGradient**: L2-norm clipping bounds individual contributions, providing sensitivity bounds for DP.

Relationship to subsequent components: The `ClipGradient` operation in Algorithm S1 establishes a bounded sensitivity C for each client's contribution. This bound is the prerequisite for Algorithm S2 (Gaussian DP), which calibrates noise proportional to C. Meanwhile, `ValidatePermit` invokes Algorithm S3 (Permit Validation) and `FilterOptedOut` invokes Algorithm S6 (Opt-Out Filtering).

B. Gaussian Differential Privacy Mechanism

Algorithm S2 implements the Gaussian mechanism for differential privacy, applied at the aggregation server after receiving clipped gradients.

Mathematical Foundation: The noise scale $\sigma = C \cdot \sqrt{2\ln(1.25/\delta)}/\varepsilon$ guarantees (ε, δ)-DP. The cumulative privacy expenditure is tracked using Rényi DP (RDP) [26] composition, providing 5–6× tighter bounds than naive composition.

Practical Considerations:

- $\varepsilon = 10$: moderate noise, expected accuracy drop in the 5–6pp range (consistent with Wei et al. [27])
- $\varepsilon = 1$: strong privacy, larger accuracy cost; a comprehensive ablation across ε values is planned
- The ε selection must be negotiated with HDABs during permit approval

Integration with privacy budget: The ε consumed by Algorithm S2 in each round is tracked by Algorithm S8 (RDP Privacy Budget Accountant). If the cumulative budget exceeds the threshold approved in the HDAB data permit, training is automatically terminated. The next algorithm (S3) formalizes the permit validation that authorizes each round.

C. HDAB Permit Validation

Algorithm S3 ensures all FL operations comply with the data permit issued by HDABs. Under EHDS Article 53, secondary use of health data is only lawful for specifically

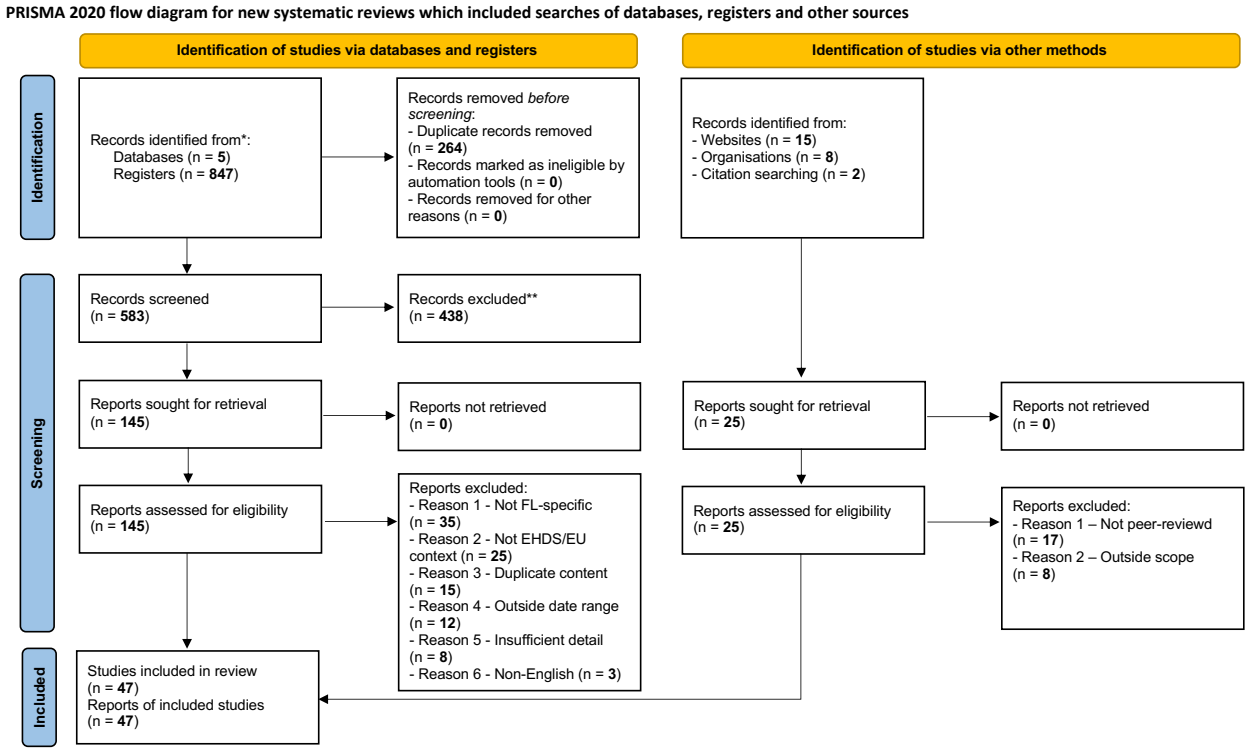

Fig. 1. PRISMA 2020 flow diagram for the systematic review. Database searches across PubMed, IEEE Xplore, Scopus, Web of Science, and arXiv identified 847 records; after deduplication (264 removed) and screening, 47 studies met inclusion criteria (2022–2026, FL/EHDS focus, peer-reviewed or recognized institutional origin). Additional records from institutional websites (n=15), organisations (n=8), and citation searching (n=2) were assessed but did not contribute to the final inclusion set. Adapted from Page et al. (BMJ 2021;372:n71), licensed under CC BY 4.0.

enumerated purposes (scientific research, public health surveillance, AI training for health). The permit validation module is invoked at the beginning of every FL round—not just at training start—to guarantee continuous compliance even if a permit is revoked mid-study or its temporal validity expires. Each validation event is logged as a GDPR Article 30 processing record, creating an immutable audit trail that regulators can inspect.

EHDS Governance Role: This algorithm is the enforcement point between the Governance Layer (Layer 1) and the FL Orchestration Layer (Layer 2). Without it, a permit expiring at round 15 of a 20-round study would allow unauthorized data processing for rounds 16–20.

Failure modes: Each exception type triggers a different response: `PermitExpiredError` terminates the entire study; `PurposeMismatchError` indicates a configuration error requiring researcher intervention; `UnauthorizedCategoryError` may allow continued training on the authorized subset of categories. All failure events are logged for regulatory audit. Once a round passes permit validation, the aggregation server collects client gradients using the secure protocol described next.

D. Secure Aggregation Protocol

Even though FL prevents raw data sharing, the gradient updates themselves can leak patient information: Zhu et al. [21] demonstrate that gradients can be inverted to reconstruct training images. In the EHDS context, where gradients encode patterns from sensitive health records across 27 Member States, this is an unacceptable privacy risk. Secure aggregation addresses this by ensuring the SPE aggregation server can compute $\sum_k \Delta_k$ without ever observing any individual hospital's gradient Δ_k.

Algorithm S4 implements this using Shamir's secret sharing and pairwise masking, ensuring the server observes only the aggregate gradient.

Protocol Phases: (1) Each client splits gradients into K shares using (t, K)-threshold Shamir secret sharing; (2) Clients add pairwise random masks negotiated via ECDH key

Algorithm S1: FL-EHDS FedAvg Training

Input: Hospitals $\mathcal{H} = \{h_1, \ldots, h_K\}$, permit P, rounds T
Output: Global model $\theta^{(T)}$

Server executes:
  Initialize $\theta^{(0)}$
  for round $t = 1$ to T **do**
    // Governance check (Layer 1)
    if not ValidatePermit(P, t) **then abort**
    $\mathcal{H}_t \leftarrow$ SelectParticipants($\mathcal{H}$)
    for each hospital $h \in \mathcal{H}_t$ **in parallel do**
      $\Delta_h^{(t)}, n_h \leftarrow$ LocalTrain(h, $\theta^{(t-1)}$)
    // Aggregation with privacy (Layer 2)
    $\theta^{(t)} \leftarrow \theta^{(t-1)} + \frac{1}{\sum_h n_h} \sum_{h \in \mathcal{H}_t} n_h \cdot \Delta_h^{(t)}$
    LogCompliance(t, $\mathcal{H}_t$)
  return $\theta^{(T)}$

LocalTrain(h, θ) **at hospital** h:
  // Opt-out filtering (Article 71)
  $\mathcal{D}_h \leftarrow$ FilterOptedOut($\mathcal{D}_h$, OptOutRegistry)
  $\theta_h \leftarrow \theta$
  for epoch $e = 1$ to E **do**
    for batch $\mathcal{B} \in \mathcal{D}_h$ **do**
      $\theta_h \leftarrow \theta_h - \eta \nabla \mathcal{L}(\theta_h; \mathcal{B})$
  $\Delta_h \leftarrow \theta_h - \theta$
  // Privacy protection (Layer 3)
  $\Delta_h \leftarrow$ ClipGradient(Δ_h, C)
  return Δ_h, $|\mathcal{D}_h|$

Algorithm S2: Gaussian DP Mechanism

Input: Gradient Δ, sensitivity C, privacy budget ε, δ
Output: Noisy gradient $\tilde{\Delta}$

// Compute noise scale from Gaussian mechanism
$\sigma \leftarrow C \cdot \sqrt{2 \ln(1.25/\delta)}/\varepsilon$
// Add calibrated Gaussian noise to each parameter
for each parameter $w \in \Delta$ **do**
  $\tilde{w} \leftarrow w + \mathcal{N}(0, \sigma^2)$
// Track cumulative privacy expenditure
PrivacyAccountant.spend(ε)
if PrivacyAccountant.budget_exhausted() **then**
  raise PrivacyBudgetExhaustedError
return $\tilde{\Delta}$

Algorithm S3: Data Permit Validation

Input: Permit P, round t, requested categories $\mathcal{C}$
Output: Boolean validity

// Check temporal validity
if CurrentTime() $> P$.valid_until **then**
  raise PermitExpiredError
// Check purpose alignment (Article 53)
if P.purpose $\notin$ AllowedPurposes **then**
  raise PurposeMismatchError
// Check data category authorization
for each category $c \in \mathcal{C}$ **do**
  if $c \notin P$.authorized_categories **then**
    raise UnauthorizedCategoryError
// Log access for GDPR Article 30
AuditTrail.log(permit=P, round=t, categories=$\mathcal{C}$)
return True

exchange; (3) The server computes the sum—masks cancel out and only the true aggregate remains.

Algorithm S4: Secure Aggregation (Pairwise Masking)

Input: Client gradients $\{\Delta_1, \ldots, \Delta_K\}$, threshold t
Output: Aggregated gradient Δ_{agg}

// Phase 1: ECDH key exchange + Shamir sharing
for each client k **do**
  shares$_k \leftarrow$ ShamirShare(Δ_k, t, K)
  Distribute shares$_k$ to other clients
// Phase 2: Add pairwise random masks
for each client k **do**
  $\hat{\Delta}_k \leftarrow \Delta_k + \sum_{j<k} r_{jk} - \sum_{j>k} r_{kj}$
// Phase 3: Server reconstructs aggregate
$\Delta_{agg} \leftarrow \sum_{k=1}^{K} \hat{\Delta}_k$
// Masks cancel: $\sum_k \sum_{j<k} r_{jk} - \sum_k \sum_{j>k} r_{kj} = 0$
if ActiveClients $< t$ **then**
  raise SecureAggregationError
return Δ_{agg}

Defense-in-depth: Secure aggregation (Algorithm S4) combined with differential privacy (Algorithm S2) provides layered protection: even if the aggregation server is compromised, it learns only the noisy aggregate—never individual hospital contributions. The combination addresses the unresolved GDPR question of whether model gradients constitute "personal data": with both mechanisms active, the information available to any single party is provably bounded. The following algorithms address how local training handles EHDS-specific data challenges.

E. FedProx for Non-IID Data

Algorithm S5 extends FedAvg with a proximal term that penalizes local model divergence from the global model [14]. In EHDS cross-border federations, data heterogeneity is a structural feature, not an exception: hospitals in different Member States serve distinct demographics, follow national clinical guidelines, and use different diagnostic thresholds. For instance, heart disease prevalence ranges from 39.2% in Rome to 62.6% in Amsterdam in our experimental setting. Without drift control, local models can diverge so far from the global consensus that aggregation produces a deteriorated global model. The proximal term $\frac{\mu}{2}\|\theta_h - \theta\|^2$ acts as a regularizer that keeps each hospital's local update within a controlled distance of the global model, balancing personalization with collaboration.

When to use in EHDS: Recommended for federations with moderate non-IID conditions and when client dropout is expected (hospitals may temporarily disconnect). FedProx tolerates partial participation better than FedAvg because the proximal term stabilizes local updates even with fewer training epochs.

Parameter Selection: $\mu = 0$ reduces to FedAvg; $\mu \in [0.01, 0.1]$ provides stable convergence; $\mu > 1$ may prevent local adaptation. The choice of μ should be documented in the data permit application so that the HDAB can assess the expected privacy-utility trade-off.

Algorithm S5: FedProx Local Update

Input: Local data $\mathcal{D}_h$, global model θ, proximal weight μ
Output: Local update Δ_h

$\theta_h \leftarrow \theta$
for epoch $e = 1$ to E **do**
  for batch $\mathcal{B} \in \mathcal{D}_h$ **do**
    $g \leftarrow \nabla \mathcal{L}(\theta_h; \mathcal{B})$
    // Proximal term: $\nabla \frac{\mu}{2} \|\theta_h - \theta\|^2$
    $g \leftarrow g + \mu(\theta_h - \theta)$
    $\theta_h \leftarrow \theta_h - \eta \cdot g$
$\Delta_h \leftarrow \theta_h - \theta$
return Δ_h

Before any local training begins (whether with FedAvg, FedProx, or any other algorithm), the framework must enforce citizen opt-out rights. The following algorithm ensures this compliance.

F. Article 71 Opt-Out Registry Protocol

Algorithm S6 implements the citizen opt-out mechanism mandated by EHDS Article 71. This article grants every EU citizen the right to object to secondary use of their electronic health data—a fundamental right that must be enforced *before* any gradient computation occurs. The algorithm queries the national opt-out registry maintained by each Member State and removes matching records from the local training dataset.

Granularity levels: (1) *Blanket opt-out*—citizen refuses all secondary use; (2) *Purpose-specific*—e.g., permitting scientific research but blocking commercial analytics; (3) *Category-specific*—e.g., allowing demographics but blocking genomic data. This granularity reflects the EHDS principle that citizens should have meaningful control, not merely a binary yes/no choice.

EHDS Governance Role: Opt-out filtering operates at Layer 3 (Data Holders) before local training. Registry lookups use LRU caching with configurable TTL to minimize latency ($<$10ms per round) while ensuring timely propagation of new opt-out decisions. All filtering statistics are logged for GDPR Article 30 audit compliance.

Impact on model quality: High opt-out rates reduce training data volume, potentially degrading model performance—particularly for underrepresented subpopulations. The audit log captures filtering statistics to quantify this impact and support transparency reporting. Once opted-out records are excluded, the remaining data must be harmonized into a consistent format before local model training can proceed.

G. FHIR R4 Preprocessing Pipeline

Algorithm S7 standardizes heterogeneous EHR data into FHIR R4 format for ML consumption. This preprocessing step is critical in the EHDS context because only 34% of European healthcare providers currently achieve full FHIR R4 compliance [7]. The remaining 66% use legacy formats (HL7v2, CDA, proprietary CSV exports) that must be harmonized before FL training can proceed on a consistent feature space.

Algorithm S6: Article 71 Opt-Out Filtering

Input: Local dataset $\mathcal{D}_h$, purpose p, categories $\mathcal{C}$
Output: Filtered dataset $\mathcal{D}'_h$

// Synchronize with national opt-out registry
OptOutRecords $\leftarrow$ FetchOptOutRegistry(MemberState)
$\mathcal{D}'_h \leftarrow \emptyset$
for each record $r \in \mathcal{D}_h$ **do**
  citizen_id $\leftarrow$ r.pseudonymized_id
  opted_out $\leftarrow$ False
  // Check purpose-specific opt-out
  if (citizen_id, p) $\in$ OptOutRecords **then**
    opted_out $\leftarrow$ True
  // Check category-specific opt-out
  for each $c \in \mathcal{C}$ **do**
    if (citizen_id, c) $\in$ OptOutRecords **then**
      opted_out $\leftarrow$ True
  if not opted_out **then**
    $\mathcal{D}'_h \leftarrow \mathcal{D}'_h \cup \{r\}$
// Log filtering statistics for audit
AuditLog.record(total=$|\mathcal{D}_h|$, filtered=$|\mathcal{D}'_h|$)
return $\mathcal{D}'_h$

Four-stage pipeline: (1) *Format detection* automatically identifies the source format; (2) *Terminology mapping* converts local codes to international standards (ICD-10 for diagnoses, ATC for medications, LOINC for laboratory results); (3) *FHIR transformation* produces validated FHIR R4 bundles using the six Article 33 data categories (Patient Summary, E-Prescription, Laboratory Results, Medical Imaging, Hospital Discharge, Rare Disease); (4) *Tensor extraction* converts structured FHIR resources into numerical tensors ready for model training.

EHDS Relevance: Without this harmonization step, hospitals in different Member States would produce incompatible feature spaces, making federated aggregation meaningless. The pipeline ensures that a gradient computed in a Finnish hospital is semantically compatible with one from an Italian hospital.

Algorithm S7: FHIR R4 Preprocessing

Input: Raw EHR records $\mathcal{R}$, feature specification $\mathcal{F}$
Output: Training tensors (X, y)

format $\leftarrow$ DetectFormat($\mathcal{R}$) *// HL7v2, CDA, CSV*
parser $\leftarrow$ GetParser(format)
records $\leftarrow$ parser.parse($\mathcal{R}$)
// Map to standard terminologies
for each $r \in$ records **do**
  r.diagnoses $\leftarrow$ MapToICD10(r.diagnoses)
  r.medications $\leftarrow$ MapToATC(r.medications)
  r.labs $\leftarrow$ MapToLOINC(r.labs)
fhir_bundle $\leftarrow$ ToFHIR(records)
ValidateFHIR(fhir_bundle)
$X \leftarrow$ ExtractFeatures(fhir_bundle, $\mathcal{F}$)
$X \leftarrow$ StandardScaler.fit_transform(X)
$y \leftarrow$ ExtractLabels(fhir_bundle)
return (X, y)

Validation requirements: The FHIR validation step rejects records with missing mandatory fields or invalid terminology codes, ensuring data quality before model training. Rejected

records are logged (without patient-identifiable content) for audit purposes. With harmonized data ready for training, the final core component manages the overall privacy budget across the entire study.

H. Privacy Budget Accountant

Algorithm S8 tracks cumulative privacy expenditure across FL rounds using Rényi Differential Privacy (RDP) moment accounting [26]. In the EHDS governance model, the total privacy budget ε_{total} is a parameter of the data permit: the researcher specifies the desired budget in the permit application, and the HDAB evaluates whether the proposed budget provides sufficient privacy protection for the requested data categories and population size.

Why RDP accounting: Naive DP composition (adding ε per round) yields loose bounds: 20 rounds at ε=0.5 each would consume ε=10 total. RDP provides 5–6× tighter bounds [26], [27], meaning the same 20 rounds can achieve the same privacy guarantee with significantly less noise—and therefore better model utility.

Hard budget enforcement: When the cumulative expenditure approaches ε_{total}, the accountant raises a `BudgetExhaustedError` that terminates training. This prevents "privacy bankruptcy"—a situation where continued training would violate the privacy guarantee approved in the data permit. The per-round allocation strategy distributes remaining budget uniformly across remaining rounds, adapting dynamically if training converges faster than expected.

Algorithm S8: RDP Privacy Budget Accountant

Input: Total budget $(\varepsilon_{total}, \delta_{total})$, rounds T
Output: Per-round budget allocation

$\lambda \leftarrow [0] \times$ MAX_ORDER *// Rényi moments*
rounds_completed $\leftarrow 0$
function AllocateRound():
  $\varepsilon_{spent} \leftarrow$ ComputeEpsilon(λ, δ_{total})
  $\varepsilon_{remaining} \leftarrow \varepsilon_{total} - \varepsilon_{spent}$
  if $\varepsilon_{remaining} < \varepsilon_{min}$ **then**
    raise BudgetExhaustedError
  $\varepsilon_t \leftarrow \varepsilon_{remaining}/(T-$ rounds_completed)
  return ε_t
function RecordRound(σ, q):
  for order $= 1$ to MAX_ORDER **do**
    λ[order] $+=$ ComputeMoment(order, σ, q)
  rounds_completed $+= 1$

III. Supplementary Experimental Figures

This section presents detailed experimental results from the FL-EHDS benchmark suite. All figures are generated from real experimental runs available in the repository.

Note on experimental configurations: Figures 2–9 were generated from an extended 50-round, 5-client training run using the framework's synthetic EHDS scenario (simulated European hospitals: Rome, Amsterdam, Berlin, Madrid, Paris). These complement the main paper's 20-round experiments on Heart Disease UCI (4 real hospitals) and Diabetes (5 Dirichlet-partitioned clients). The 50-round configuration illustrates longer-horizon convergence properties, client participation dynamics, and gradient evolution patterns that are not visible in the shorter 20-round evaluation.

A. Hospital Data Distribution

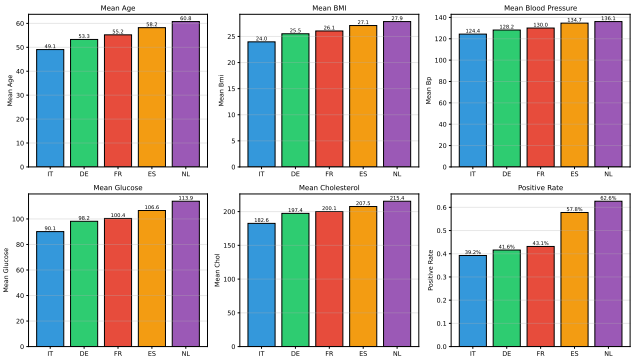

Fig. 2. Data distribution across hospitals. Notable heterogeneity: Amsterdam shows older population (60.8 years mean age) with higher positive rate (62.6%) compared to Rome (49.1 years, 39.2%). This reflects realistic cross-border EHDS variability.

B. Per-Client Training Time

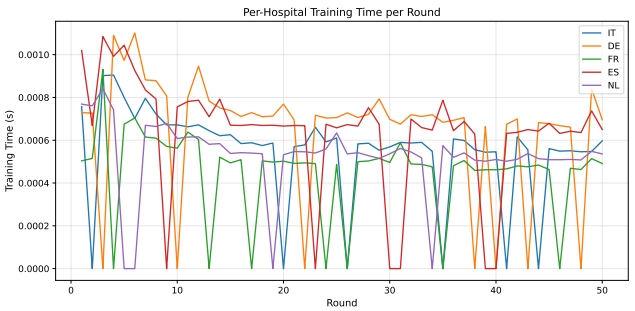

Fig. 3. Per-client training time per round. Larger hospitals (Berlin: 500 samples) exhibit slightly longer training times. The adaptive training engine compensates by adjusting batch sizes for stragglers.

C. Client Participation Matrix

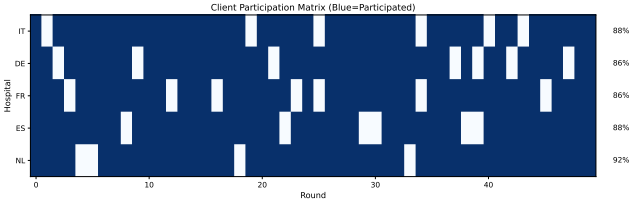

Fig. 4. Client participation matrix (50 rounds × 5 clients). Participation rates: IT 88%, DE 86%, FR 86%, ES 88%, NL 92%. The framework tolerates 10–15% dropout per round while maintaining convergence.

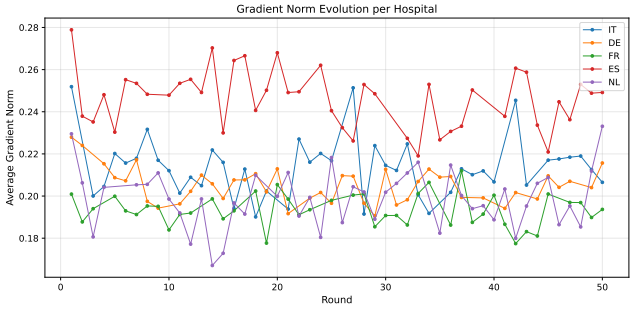

Fig. 5. Gradient norm evolution per client over 50 rounds. All clients show decreasing trends indicating stable convergence. Clipping threshold C=1.0 bounds extreme values for DP compatibility.

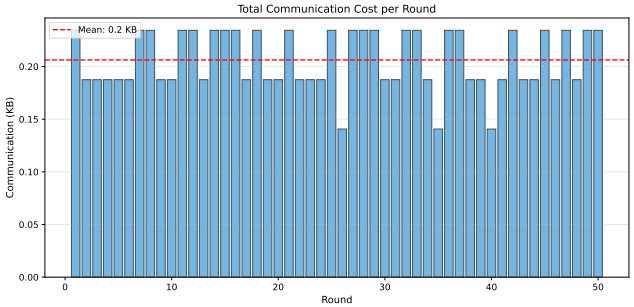

Fig. 6. Cumulative communication cost per round. Linear scaling with participating clients (3.5 KB/client/round). Total 50-round overhead: 875 KB for 5 clients—feasible even for bandwidth-constrained environments.

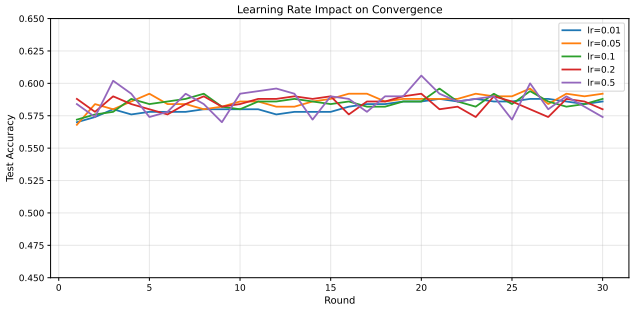

Fig. 7. Learning rate sensitivity analysis. η=0.01: slow convergence (53.8% at round 50). η=0.1: optimal (58.6%). η=0.5: instability with oscillations.

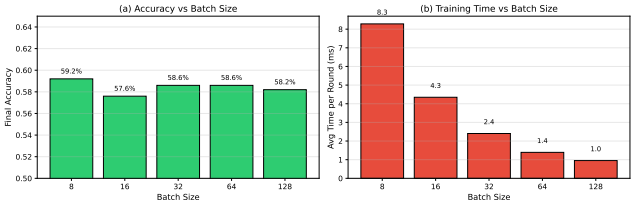

Fig. 8. Batch size impact on convergence. Smaller batches (8–16) provide noisier gradients but faster initial progress. Batch size 32 balances gradient quality and computational efficiency.

D. Gradient Norm Evolution

E. Communication Cost Analysis

F. Learning Rate Sensitivity

G. Batch Size Impact

H. Per-Client Accuracy Trajectories

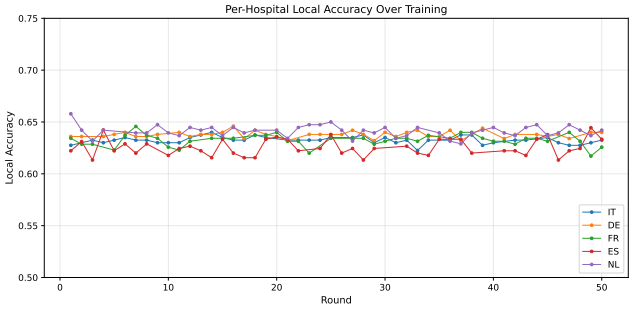

Fig. 9. Per-client accuracy over training rounds. Variance reflects non-IID data: NL (older, higher-risk population) reaches 64% accuracy while FR (mid-range demographics) stabilizes at 55%.

IV. Extended Algorithm Comparison

A. Algorithms Evaluated

We compare foundational FL algorithms plus 2022–2025 advances:

Foundational: FedAvg [13], FedProx [14], SCAFFOLD [31], FedAdam/FedYogi/FedAdagrad [33].

Recent (2022–2025): FedLC [37] (logit calibration for label skew), FedSAM [36] (flat minima), FedDecorr [38] (decorrelation against dimensional collapse), FedSpeed [39] (fewer rounds), FedExP [40] (server-side acceleration), FedLESAM [41] (globally-guided SAM, ICML 2024 Spotlight), HPFL [42] (personalized classifiers, ICLR 2025).

B. Non-IID Configuration

Data heterogeneity is controlled via Dirichlet distribution with α:

- $\alpha = 0.1$: **Extreme non-IID**—highly skewed label distributions
- $\alpha = 0.5$: **High non-IID**—significant heterogeneity
- $\alpha = 1.0$: **Moderate non-IID**—balanced heterogeneity
- $\alpha = 10.0$: **Near-IID**—approximately uniform

C. Convergence at Different Heterogeneity Levels

Findings: (1) At α=0.1, SCAFFOLD achieves most stable convergence via variance reduction. (2) FedProx provides marginal improvement over FedAvg at α=0.5–1.0. (3) Adaptive methods (FedAdam, FedYogi) excel in near-IID but may oscillate under extreme heterogeneity. (4) FedAvg remains competitive in near-IID, suitable for homogeneous federations.

D. Final Accuracy vs. Heterogeneity

E. Convergence Speed

F. Algorithm Selection Guidelines

Table I maps EHDS deployment scenarios to recommended algorithms.

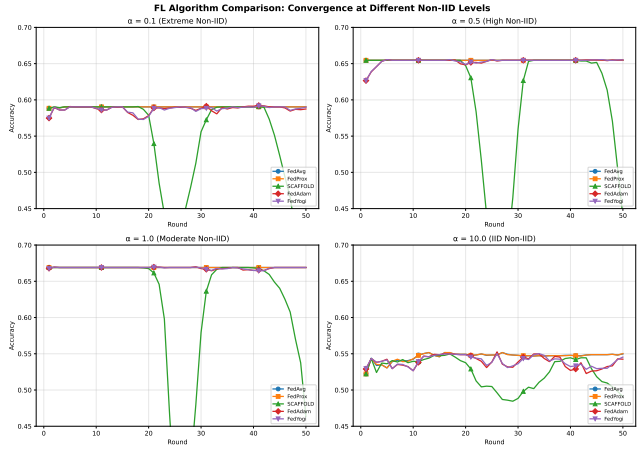

Fig. 10. Algorithm convergence across non-IID levels ($\alpha \in \{0.1, 0.5, 1.0, 10.0\}$). SCAFFOLD and adaptive methods show superior stability under extreme heterogeneity.

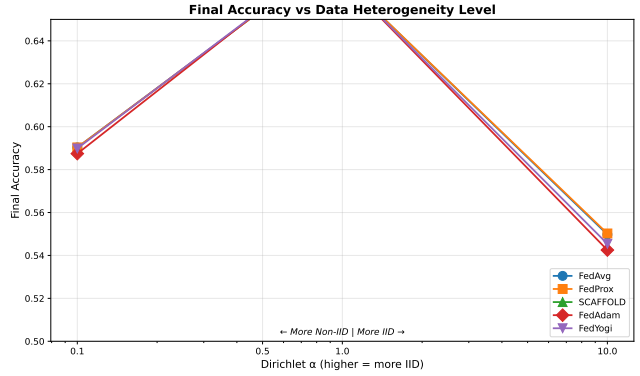

Fig. 11. Final accuracy vs. Dirichlet α. All algorithms degrade under extreme non-IID. SCAFFOLD shows smallest gap between α=0.1 and α=10.

V. Advanced FL Paradigms

The core FL-EHDS pipeline (Section II) addresses the standard "horizontal" FL scenario where all hospitals share the same feature schema. However, real EHDS deployments will encounter more complex configurations: institutions with complementary features for the same patients (vertical FL), adversarial participants (Byzantine resilience), evolving data distributions over the 2025–2031 timeline (continual FL), heterogeneous clinical objectives (multi-task FL), and the hierarchical governance structure of the EU itself (hierarchical FL). This section presents the advanced paradigms implemented in the reference framework, each motivated by a specific EHDS deployment challenge. Algorithms S9–S13 formalize the core mechanisms.

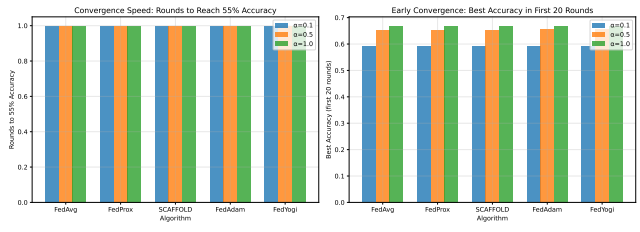

Fig. 12. Convergence speed comparison. Left: rounds to 55% accuracy. Right: best accuracy in first 20 rounds. Adaptive methods converge faster but may plateau.

TABLE I
Algorithm Selection for EHDS Deployments

EHDS Scenario	Algorithm	Rationale
Homogeneous MS	FedAvg	Simplicity, proven
Heterogeneous MS	SCAFFOLD	Variance reduction
Resource-limited	FedAdam	Fast convergence
Privacy-critical	FedAvg + DP	Well-studied bounds
Sparse participation	FedProx	Dropout resilience
Label-imbalanced	FedLC	Class-freq. calib.
Deep models, non-IID	FedDecorr	Dim. collapse prev.
Comm.-constrained	FedSpeed	Fewer rounds
No client changes	FedExP	Server-side only
SAM + global drift	FedLESAM	Global flatness
Per-hosp. classif.	HPFL	Local boundaries

MS = Member States. Scenarios may combine: heterogeneous + privacy-critical → SCAFFOLD + DP.

A. Vertical Federated Learning

Vertical FL addresses scenarios where institutions hold *different features* for the *same patients*—a common situation in EHDS cross-border analytics. For example, a hospital may hold demographics and diagnoses, a laboratory holds test results, and a pharmacy holds prescription histories. Under EHDS Article 33, these correspond to different data categories (Patient Summary, Laboratory Results, E-Prescription) that may be held by different data holders within the same or different Member States.

Private Set Intersection (PSI): Before training, the participating institutions must identify their common patients without revealing their full patient lists. RSA-based PSI achieves this with $O(n \log n)$ complexity using pseudonymized identifiers, ensuring EHDS compliance: no institution learns which patients the other holds beyond the intersection.

Split Learning: Algorithm S9 implements the forward pass in split learning, where each party computes activations on its local features up to a "cut layer," then the server concatenates activations to produce the final prediction. Only intermediate representations (not raw data) cross institutional boundaries.

Algorithm S9: Split Learning Forward Pass

Input: Features X_A, X_B at parties A, B; cut layer k
Output: Prediction $\hat{y}$

$h_A \leftarrow f^A_{1:k}(X_A)$ // *Party A: features → activations*
$h_B \leftarrow f^B_{1:k}(X_B)$ // *Party B: features → activations*
$h \leftarrow \text{Concat}(h_A, h_B)$
$\hat{y} \leftarrow f_{k+1:L}(h)$ // *Server: cut layer → output*
return $\hat{y}$

B. Byzantine-Resilient Aggregation

In a cross-border EHDS federation spanning 27 Member States, the aggregation server cannot blindly trust every participant. A compromised institution—whether through malicious intent, software bugs, or data corruption—could submit adversarial gradient updates that poison the global model, potentially affecting clinical decisions across the entire federation. Byzantine-resilient aggregation protects model integrity by detecting and excluding anomalous updates.

Algorithm S10 implements Krum, which selects the gradient closest to $n-f-2$ nearest neighbors, effectively filtering outliers. Six defense methods protect against up to $f < n/3$ adversarial clients:

Algorithm S10: Krum Byzantine Defense
Input: Gradients $\{g_1, \ldots, g_n\}$, Byzantine bound f
Output: Selected gradient g^*
for each gradient g_i **do**
  $D_i \leftarrow \{\|g_i - g_j\|^2 : j \neq i\}$
  $s_i \leftarrow \sum_{d \in \text{smallest}_{n-f-2}(D_i)} d$
$g^* \leftarrow g_{\arg\min_i s_i}$
return g^*

Other methods: **Trimmed Mean** (removes β-fraction extreme values per coordinate), **Coordinate-wise Median** (robust estimator), **Bulyan** (two-stage Krum + trimmed mean), **FLTrust** (server-guided trust weighting), **FLAME** (clustering-based). Attack simulation: label flipping, gradient scaling, additive noise, sign flipping, model replacement.

C. Continual Federated Learning

The EHDS is designed for long-term operation (2025–2031 and beyond), during which healthcare data distributions will evolve: new diseases emerge (as demonstrated by COVID-19), clinical protocols change, and demographic compositions shift. A model trained in 2027 may perform poorly on 2029 data if it has "forgotten" how to handle earlier patterns. Continual Federated Learning addresses this *catastrophic forgetting* problem by preserving knowledge from previous training tasks while adapting to new data.

The Elastic Weight Consolidation (EWC) loss function adds a quadratic penalty:

$$\mathcal{L}_{EWC}(\theta) = \mathcal{L}(\theta) + \frac{\lambda}{2} \sum_i F_i(\theta_i - \theta_i^*)^2$$

where F_i is the Fisher Information for parameter i and θ^* are optimal parameters for previous tasks.

Additional strategies: Learning without Forgetting (LwF), Experience Replay, drift detection (ADWIN, Page-Hinkley) triggering adaptation.

D. Multi-Task Federated Learning

In EHDS cross-border studies, different hospitals may pursue related but distinct clinical objectives from the same data. A cardiology network might simultaneously predict heart failure risk (Hospital A), readmission probability (Hospital B), and medication response (Hospital C). Multi-Task FL enables these institutions to collaborate on shared feature representations while maintaining task-specific prediction heads.

Architectures: **Hard Parameter Sharing** (common feature extractor, task-specific heads), **Soft Parameter Sharing** (separate networks with similarity regularization), **FedMTL** (dynamic task relationship learning).

E. Hierarchical Federated Learning

The EHDS governance structure is inherently hierarchical: individual hospitals report to regional health authorities, which coordinate under national HDABs, which in turn connect to the EU-level HealthData@EU infrastructure. Hierarchical FL mirrors this governance topology, aggregating gradients at intermediate levels before reaching the central server. This reduces communication costs (hospitals communicate with regional aggregators, not directly with the EU server) and aligns FL operations with the jurisdictional boundaries of HDABs.

Four-tier hierarchy reflecting EU governance:

1) **Client Tier**: Individual hospitals/data holders
2) **Regional Tier**: Regional aggregators (e.g., Lombardy, Bavaria)
3) **National Tier**: National HDABs coordinate Member State aggregation
4) **EU Tier**: HealthData@EU central aggregator

Benefits: reduced communication costs (hospitals $\rightarrow$ regional, not directly EU), alignment with EHDS governance where HDABs have national jurisdiction.

F. Personalized Federated Learning

A single global model may underperform at individual hospitals because clinical populations differ substantially across Member States. Personalized FL maintains both a global model (encoding shared medical knowledge) and hospital-specific local models (adapted to local demographics and clinical practices). Algorithm S11 shows pFedMe [35] as a representative personalized method, using Moreau envelopes to balance personalization with global knowledge: the regularization term $\lambda(\theta_k - \theta)$ pulls the local model toward the global consensus, while local gradient descent adapts to hospital-specific data patterns. Ditto [34] follows a similar dual-model principle but with a simpler formulation: it trains a personalized model regularized toward the global model via $\frac{\lambda}{2}\|\theta_k - \theta\|^2$. In our experiments, Ditto—the best-performing personalized method—achieves 75.1% accuracy on Heart Disease, a 12.6pp improvement over FedAvg, precisely because it learns hospital-specific decision boundaries.

Other approaches: **FedPer** (shared base, local personalization layers), **Per-FedAvg** (MAML-based meta-learning), **APFL** (adaptive mixing α between global and local), **Ditto** (personalization regularization).

EHDS Relevance: Member States have different healthcare systems, disease prevalence, and clinical practices. Personalized FL enables institution-specific adaptation while benefiting from collaborative training.

Algorithm S11: pFedMe Local Update

Input: Data $\mathcal{D}_k$, global θ, personal θ_k, λ, η
Output: Updated personal model θ'_k

for $i = 1$ to R **do**
  $\theta_k \leftarrow \theta_k - \eta \nabla \mathcal{L}(\theta_k; \mathcal{D}_k)$
// Moreau envelope: balance with global
$\theta'_k \leftarrow \theta_k - \lambda(\theta_k - \theta)$
$g_k \leftarrow \lambda(\theta - \theta'_k)$
return θ'_k, g_k

G. Asynchronous Federated Learning

Standard synchronous FL requires all participating hospitals to complete local training before the server can aggregate. In an EHDS federation spanning 27 Member States with heterogeneous computational resources, this creates a "straggler" problem: a resource-constrained rural hospital delays the entire federation. Asynchronous FL eliminates this bottleneck by allowing the server to aggregate updates as they arrive, weighting stale updates (from slow clients) less heavily. Algorithm S12 implements polynomial staleness weighting: an update computed τ rounds ago receives weight $(1 + \tau)^{-a}$, ensuring that fresher updates contribute more while still incorporating information from slower participants.

Algorithm S12: FedAsync with Staleness Weighting

Input: Client update Δ_k, client round t_k, server round t
Output: Updated global model θ

$\tau \leftarrow t - t_k$ *// Staleness*
$\alpha \leftarrow (1 + \tau)^{-a}$ *// Polynomial decay, $a > 0$*
$\theta \leftarrow \theta + \alpha \cdot \eta \cdot \Delta_k$
return θ

Staleness functions: Constant (α=1), Polynomial ($(1+\tau)^{-a}$), Exponential ($e^{-a\tau}$), Hinge (1 if $\tau \leq \tau_{max}$, else 0). Additional: FedBuff (buffered async), semi-async (wait for α-fraction of clients).

H. Fairness-Aware Federated Learning

The EHDS serves 450 million citizens across Member States with different population sizes, disease prevalence, and healthcare quality. Standard FL optimizes average performance, which can disproportionately favor large hospitals with more data while neglecting smaller institutions or underrepresented patient populations. This creates a "digital health equity" concern: a model that achieves 85% accuracy for a large German hospital but only 55% for a small Romanian clinic is not equitable. Algorithm S13 implements q-FedAvg, which reweights client contributions by their loss: hospitals where the model performs poorly receive higher aggregation weights, pulling the global model toward equitable performance across all participants.

Fairness metrics: Performance Variance ($\text{Var}(\{L_k\})$), Worst-case Loss ($\max_k L_k$), Demographic Parity Gap, Equalized Odds Gap. Additional methods: AFL, FedMGDA+, TERM, FairFed.

Algorithm S13: q-FedAvg Fair Aggregation

Input: Losses $\{L_1, \ldots, L_K\}$, updates $\{\Delta_1, \ldots, \Delta_K\}$, q
Output: Fair aggregated update Δ

for each client k **do**
  $w_k \leftarrow L_k^q$ *// Higher loss → higher weight*
$W \leftarrow \sum_k w_k$; $w_k \leftarrow w_k / W$
$\Delta \leftarrow \sum_k w_k \cdot \Delta_k$
return Δ

VI. Infrastructure Components

Deploying FL across 27 EU Member States requires production-grade infrastructure: reliable communication channels between hospitals and the SPE aggregator, efficient serialization of gradient tensors, distributed coordination for concurrent studies, and comprehensive monitoring with EHDS-specific alerting. This section describes the infrastructure components implemented in the reference framework, each designed to operate within the constraints of cross-border healthcare networks (firewalls, bandwidth limitations, regulatory requirements).

A. Communication Layer

The communication layer must bridge heterogeneous network environments: high-bandwidth data center connections between national HDABs, moderate hospital-to-aggregator links, and potentially bandwidth-constrained rural clinics. The framework supports two transport protocols selectable per deployment, with configurable compression and retry policies.

Communication Manager Configuration

```
transport: gRPC | WebSocket
compression: gzip | lz4 | zstd | none
chunk_size: 1MB
retry_policy:
  max_retries: 3
  backoff: exponential
  base_delay: 1s
connection_pool:
  max_connections: 100
  idle_timeout: 300s
```

gRPC: Bidirectional streaming, Protocol Buffers (30% bandwidth reduction vs. JSON), HTTP/2 multiplexing. Ideal for data center deployments.

WebSocket: Browser-compatible, firewall-friendly (standard HTTP upgrade), event-driven. Ideal for edge deployments and browser-based participation.

Selection criteria: gRPC is recommended for production EHDS deployments where both endpoints support HTTP/2 (typical for hospital-to-national aggregator links). WebSocket is preferred when traffic must traverse web application firewalls or when browser-based dashboards participate directly in federation monitoring.

B. Serialization

Binary Format: Tensor metadata + raw binary, 30% smaller than JSON, 15% smaller than pickle, cross-platform (Python, C++, Java).

Delta Serialization: Transmits only changed parameters, sparse encoding, up to 90% bandwidth reduction for fine-tuning.

EHDS-Compliant: Embeds permit ID, timestamp, provenance; cryptographic signatures; GDPR Article 30 audit fields.

C. Caching Layer

In production EHDS deployments, multiple FL studies may run concurrently on overlapping data holders. A distributed locking mechanism prevents race conditions during gradient aggregation—ensuring that two concurrent studies do not interfere with each other's model updates. Algorithm S14 implements Redis-based distributed locking with TTL-based automatic release, preventing deadlocks if a server node fails mid-aggregation.

Algorithm S14: Distributed Lock for Aggregation
Input: Lock name, TTL, client ID
Output: Lock acquired (boolean)

```
acquired ← Redis.SET(lock_name, client_id, NX, EX=TTL)
if acquired then
    PerformAggregation()
    if Redis.GET(lock_name) == client_id then
        Redis.DEL(lock_name)
return acquired
```

Redis-based caching: model checkpoints, client states, real-time metrics. Features: LRU/LFU/TTL eviction, distributed locking, automatic serialization, cache warming.

D. Orchestration

Kubernetes: Deploys FL clients/aggregators as pods, HPA for elastic scaling, ConfigMaps for hyperparameters, Secrets for HDAB API keys.

Ray: Actor-based FL, automatic fault tolerance, Ray Tune for federated HPO, Object Store for gradient sharing.

Auto-Scaling: Reactive (queue depth/latency), Predictive (ML-based forecasting), Scheduled (time-based patterns).

E. Monitoring

Prometheus Metrics: Counters (rounds_total, permits_validated), Gauges (active_clients, privacy_budget_remaining), Histograms (round_duration, communication_latency), Summaries (gradient_norm_quantiles).

Grafana Dashboards: FL training progress, client health, latency heatmaps, privacy budget consumption, EHDS compliance status.

Alerting: Privacy budget exhaustion, client dropout threshold, model divergence, permit expiration.

F. Model Watermarking

IP protection for FL models trained on EHDS data: **Spread Spectrum** (frequency domain, robust to fine-tuning), **LSB** (low-order weight bits), **Backdoor-based** (input-output ownership proof), **Passport Layers** (dedicated ownership encoding).

G. Cross-Silo Enhancements

EHDS federations are inherently cross-silo: each participant is an institution (hospital, registry, research center) with significant computational resources, distinct data distributions, and long-term participation commitments. This differs from cross-device FL (e.g., mobile phones) and enables advanced optimization strategies.

Multi-Model Federation: Weighted voting, stacking, mixture of experts with diversity enforcement.

Automatic Algorithm Selection: The 17 FL algorithms in the framework have different strengths depending on the federation characteristics (heterogeneity level, number of participants, communication budget). Algorithm S15 implements adaptive aggregation selection via multi-armed bandit (UCB/Thompson Sampling), automatically switching algorithms mid-training if performance metrics indicate a better alternative. A cooldown period prevents oscillation between strategies.

Algorithm S15: Adaptive Aggregation
Input: Client updates, metrics history, cooldown
Output: Aggregated model, selected algorithm

```
score ← WeightedScore(loss, accuracy, variance, conv.)
if RoundsSinceSwitch > Cooldown then
    for each candidate ∈ Algorithms do
        alt ← EstimatePerformance(candidate)
        if alt > score + Threshold then
            SwitchTo(candidate)
aggregated ← CurrentAlgo.Aggregate(updates)
return aggregated
```

VII. Extended EHDS Interoperability

A. OMOP Common Data Model

OMOP CDM v5.4 provides standardized analytical format used by European research networks (EHDEN, OHDSI).

ETL Pipelines: Transform source EHR to OMOP. **Vocabulary Mapping**: SNOMED, ICD10, LOINC, RxNorm. **Cohort Definitions**: ATLAS-compatible SQL generation. **Feature Extraction**: FeatureExtraction package for ML-ready datasets.

FL Integration: (1) Each hospital transforms local EHR to OMOP; (2) Feature extraction produces identical schema; (3) FL training on homogeneous feature spaces.

B. IHE Integration Profiles

ATNA: TLS mutual authentication, syslog audit messages (RFC 5424), maps to GDPR Article 30.

BPPC: Maps Article 71 opt-out to consent documents, XDS.b integration, consent enforcement at FL initiation.

XCA: Cross-border document query/retrieve, Initiating/Responding Gateways, patient identity correlation.

PIX/PDQ: Patient matching across boundaries, pseudonymization-aware identity management, national eHealth integration.

XUA: SAML 2.0 federated authentication, role-based access control, HDAB authorization token propagation.

C. Cross-Border Data Exchange

Message Formats: EHDS Data Permit Exchange Format (JSON-LD), Federated Query Protocol (SPARQL Federation), Model Update Message Format (Protocol Buffers).

Security: eIDAS-compliant electronic signatures, TLS 1.3, certificate-based authentication (EU trust framework).

Metadata: DCAT-AP Health extension, W3C PROV-O provenance, EMA data quality indicators.

D. Interoperability Architecture

Figure 13 presents the complete interoperability architecture, showing how heterogeneous data sources across EU Member States are harmonized through multiple standards layers before reaching the FL training engine. The architecture reflects a key EHDS challenge: real-world healthcare institutions use diverse formats, terminologies, and exchange protocols that must be reconciled to produce a consistent feature space for federated model training.

VIII. Clinical Imaging: Extended Details

A. Datasets

Three clinical imaging datasets cover representative EHDS scenarios:

- **Chest X-ray** [48]: 5,860 pediatric radiographs (NORMAL/PNEUMONIA, 2.7:1 imbalance)
- **Brain Tumor MRI**: 3,064 T1-weighted CE MRI slices (3-class: glioma, meningioma, pituitary)
- **Skin Cancer**: 3,297 dermoscopy images (binary benign/malignant)

B. Model Architectures

HealthcareResNet: ResNet-18 [47] pretrained on ImageNet, GroupNorm replacing BatchNorm for FL stability. FedBN [46] skips normalization during aggregation. Partial backbone freeze (level 1). $\sim$11.2M parameters.

HealthcareCNN: 5-block CNN with GroupNorm, progressive channels (32$\rightarrow$512), graduated Dropout (0.15$\rightarrow$0.3). Classifier: Flatten$\rightarrow$FC(512)$\rightarrow$FC(128)$\rightarrow$FC(K). $\sim$12M parameters.

Data augmentation: random horizontal flip, rotation ($\pm$15°), brightness jitter ($\pm$10%). ImageNet normalization.

C. V2 Experimental Configuration

- 5 hospitals, 25 rounds, 3 local epochs, batch size 32
- Adam optimizer (lr=0.001), early stopping (patience=6)
- Non-IID via Dirichlet α=0.5
- FedBN enabled, partial backbone freeze (level 1)
- 7 algorithms: FedAvg, FedProx, Ditto, FedLC, FedExP, FedLESAM, HPFL
- 3 seeds per configuration (42, 123, 456)
- Total: 7 $\times$ 5 datasets $\times$ 3 seeds = 105 experiments

D. Reproducibility

All experiments are fully reproducible:

```
cd fl-ehds-framework
# Full experiments (7 algo x 5 datasets x 3 seeds)
python -m benchmarks.run_full_experiments
# Quick validation (~1-2h)
python -m benchmarks.run_full_experiments --quick
# Resume after interruption
python -m benchmarks.run_full_experiments --resume
```

Results, checkpoints, and logs are auto-saved to `benchmarks/paper_results/`.

Repository: https://github.com/FabioLiberti/FL-EHDS-FLICS2026

References

[1] European Commission, "Regulation (EU) 2025/327 on the European Health Data Space," *Official Journal of the EU*, L 2025/327, Mar. 2025.

[2] C. Staunton *et al.*, "Ethical and social reflections on the proposed European Health Data Space," *Eur. J. Human Genetics*, vol. 32, no. 5, pp. 498–505, 2024.

[3] P. Quinn, E. Ellyne, and C. Yao, "Will the GDPR restrain health data access bodies under the EHDS?" *Computer Law & Security Review*, vol. 54, art. 105993, 2024.

[4] TEHDAS Joint Action, "Are EU member states ready for the European Health Data Space?" *Eur. J. Public Health*, vol. 34, no. 6, pp. 1102–1108, 2024.

[5] H. Fröhlich *et al.*, "Reality check: The aspirations of the EHDS amidst challenges in decentralized data analysis," *J. Med. Internet Res.*, vol. 27, art. e76491, 2025.

[6] S. van Drumpt *et al.*, "Secondary use under the European Health Data Space," *Frontiers in Digital Health*, vol. 7, art. 1602101, 2025.

[7] R. Hussein *et al.*, "Interoperability framework of the EHDS for secondary use," *J. Med. Internet Res.*, vol. 27, art. e69813, 2025.

[8] R. Forster *et al.*, "User journeys in cross-European secondary use of health data," *Eur. J. Public Health*, vol. 35, Suppl. 3, pp. iii18–iii24, 2025.

[9] L. Svingel *et al.*, "Shaping the future EHDS: Recommendations for HDABs," *Eur. J. Public Health*, vol. 35, Suppl. 3, pp. iii32–iii38, 2025.

[10] C. Christiansen *et al.*, "Piloting an infrastructure for secondary use of health data," *Eur. J. Public Health*, vol. 35, Suppl. 3, pp. iii3–iii4, 2025.

[11] M. Shabani and P. Borry, "The European Health Data Space: Challenges and opportunities," *Eur. J. Human Genetics*, vol. 32, no. 8, pp. 891–897, 2024.

[12] A. Ganna, E. Ingelsson, and D. Posthuma, "The EHDS can be a boost for research beyond borders," *Nature Medicine*, vol. 30, pp. 3053–3056, 2024.

[13] B. McMahan *et al.*, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, pp. 1273–1282, 2017.

[14] T. Li *et al.*, "Federated optimization in heterogeneous networks," in *Proc. MLSys*, vol. 2, pp. 429–450, 2020.

[15] P. Kairouz *et al.*, "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, vol. 14, no. 1–2, pp. 1–210, 2021.

[16] N. Rieke *et al.*, "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, art. 119, 2020.

[17] K. Bonawitz *et al.*, "Towards federated learning at scale," in *Proc. MLSys*, pp. 374–388, 2019.

[18] M. Chavero-Diez *et al.*, "Federated learning frameworks: Quality and interoperability for biomedical research," *NAR Genomics Bioinformatics*, vol. 8, no. 1, art. lqag010, 2026.

[19] Z. L. Teo *et al.*, "Federated machine learning in healthcare: A systematic review," *Cell Reports Medicine*, vol. 5, no. 2, art. 101419, 2024.

[20] L. Peng *et al.*, "Federated machine learning in healthcare: A systematic review," *Comput. Methods Programs Biomed.*, vol. 247, art. 108066, 2024.

[21] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Proc. NeurIPS*, vol. 32, pp. 14774–14784, 2019.

[22] R. Shokri *et al.*, "Membership inference attacks against machine learning models," in *Proc. IEEE S&P*, pp. 3–18, 2017.

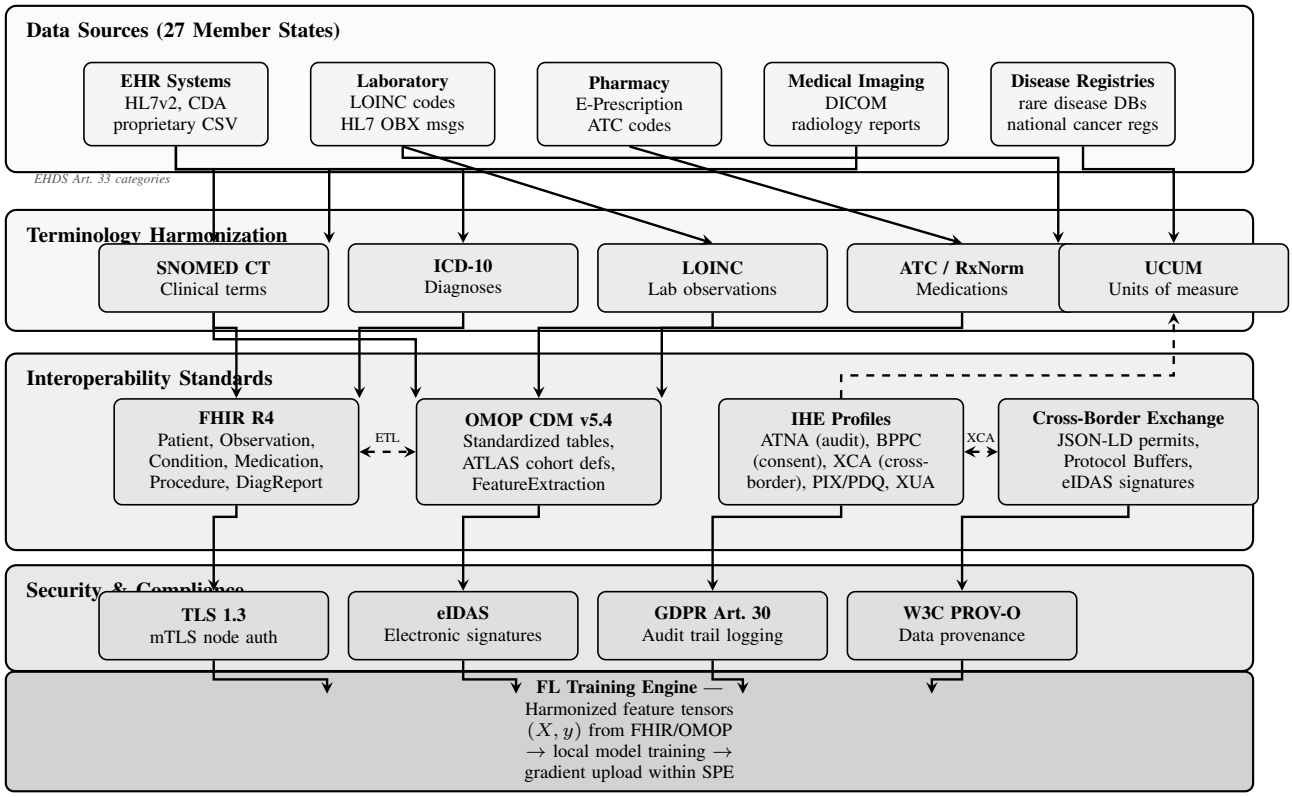

Fig. 13. EHDS interoperability architecture for FL-based secondary use. Data from heterogeneous sources across 27 Member States (top) flows through terminology harmonization (SNOMED CT, ICD-10, LOINC, ATC, UCUM), then through interoperability standards (FHIR R4 for structured data exchange, OMOP CDM for observational research, IHE profiles for cross-institutional workflows, cross-border exchange protocols with eIDAS signatures). A security and compliance layer enforces TLS 1.3 mutual authentication, eIDAS electronic signatures for permits, GDPR Article 30 audit logging, and W3C PROV-O data provenance before harmonized feature tensors reach the FL training engine. Bidirectional ETL between FHIR and OMOP enables institutions to use either standard based on their existing infrastructure.

[23] N. Carlini *et al.*, "Membership inference attacks from first principles," in *Proc. IEEE S&P*, pp. 1897–1914, 2022.

[24] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3–4, pp. 211–407, 2014.

[25] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM CCS*, pp. 308–318, 2016.

[26] I. Mironov, "Rényi differential privacy," in *Proc. IEEE CSF*, pp. 263–275, 2017.

[27] K. Wei *et al.*, "Federated learning with differential privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 3454–3469, 2020.

[28] J. Jordon *et al.*, "Synthetic data—A privacy mirage?" *J. Mach. Learn. Res.*, vol. 23, no. 1, art. 298, 2022.

[29] I. Dayan *et al.*, "Federated learning for predicting clinical outcomes in patients with COVID-19," *Nature Medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.

[30] M. J. Sheller *et al.*, "Federated learning in medicine," *Scientific Reports*, vol. 10, art. 12598, 2020.

[31] S. P. Karimireddy *et al.*, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proc. ICML*, pp. 5132–5143, 2020.

[32] J. Wang *et al.*, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Proc. NeurIPS*, vol. 33, pp. 7611–7623, 2020.

[33] S. Reddi *et al.*, "Adaptive federated optimization," in *Proc. ICLR*, 2021.

[34] T. Li *et al.*, "Ditto: Fair and robust federated learning through personalization," in *Proc. ICML*, PMLR 139, pp. 6357–6368, 2021.

[35] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with Moreau envelopes," in *Proc. NeurIPS*, vol. 33, pp. 21394–21405, 2020.

[36] Z. Qu *et al.*, "Generalized federated learning via sharpness aware minimization," in *Proc. ICML*, PMLR 162, pp. 18250–18280, 2022.

[37] J. Zhang *et al.*, "Federated learning with label distribution skew via logits calibration," in *Proc. ICML*, PMLR 162, pp. 26311–26329, 2022.

[38] Y. Shi *et al.*, "Towards understanding and mitigating dimensional collapse in heterogeneous federated learning," in *Proc. ICLR*, 2023.

[39] Y. Sun *et al.*, "FedSpeed: Larger local interval, less communication round, and higher generalization accuracy," in *Proc. ICLR*, 2023.

[40] D. Jhunjhunwala, S. Wang, and G. Joshi, "FedExP: Speeding up federated averaging via extrapolation," in *Proc. ICLR*, 2023.

[41] Z. Qu *et al.*, "FedLESAM: Federated learning with locally estimated sharpness-aware minimization," in *Proc. ICML*, PMLR 235, 2024.

[42] Y. Chen, X. Cao, and L. Sun, "HPFL: Hot-pluggable federated learning with shared backbone and personalized classifiers," in *Proc. ICLR*, 2025.

[43] D. J. Beutel *et al.*, "Flower: A friendly federated learning research framework," *arXiv:2007.14390*, 2023.

[44] NVIDIA, "NVIDIA FLARE: An open-source federated learning platform," *GitHub Repository*, 2023.

[45] Google, "TensorFlow Federated: Machine learning on decentralized data," 2019.

[46] X. Li *et al.*, "FedBN: Federated learning on non-IID features via local batch normalization," in *Proc. ICLR*, 2021.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, pp. 770–778, 2016.

[48] D. S. Kermany *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.