Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 2026

N. Kale
Cisco Systems

January 2026

Privacy-Preserving Federated Learning Architecture for Multi-Tenant AI Agent Systems

draft-kale-agntcy-federated-privacy-00

Abstract

This document specifies a reference architecture for privacy-preserving federated learning in multi-tenant AI agent deployments. It addresses the challenge of enabling collaborative model training across organizational boundaries while maintaining formal privacy guarantees and tenant data isolation. The architecture combines federated averaging, differential privacy mechanisms, and secure aggregation to enable cross-tenant knowledge transfer without exposing sensitive behavioral data.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 7, 2026.

Table of Contents

1. Introduction

AI agent systems increasingly operate in multi-tenant enterprise environments where behavioral data from user interactions could improve system performance through machine learning. However, centralizing such data across organizational boundaries conflicts with privacy regulations (GDPR, HIPAA, CCPA), contractual data isolation requirements, and enterprise security policies.

This document specifies a federated learning architecture that enables collaborative model training while ensuring:

- Tenant data isolation: Raw behavioral data never leaves tenant boundaries
- Formal privacy guarantees: Differential privacy bounds information leakage

- Regulatory compliance: Architecture supports GDPR, HIPAA, and CCPA requirements
- Scalable deployment: Protocol supports large numbers of tenants with heterogeneous data distributions

The architecture applies broadly to any multi-tenant machine learning deployment requiring privacy preservation, with particular relevance to AI agent systems operating across organizational boundaries. This work builds on foundational research in federated learning [McMahan17] and differential privacy [Dwork14] [Abadi16], as well as recent advances addressing open problems in the field [Kairouz21] [Wang23].

1.1. Relationship to AI Agent Protocol Work

This document complements ongoing work on AI agent protocols at the IETF, including frameworks for agent communication [Rosenberg25] and secure messaging for agentic AI [SLIM25]. The privacy-preserving aggregation mechanisms specified here address federated learning requirements that arise when AI agents operate across

organizational boundaries and must learn from distributed behavioral data without centralizing sensitive information.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Aggregation Server: A central coordinator that receives model updates from tenants and computes aggregated models.

Differential Privacy (DP): A mathematical framework providing formal bounds on information leakage from statistical queries.

Federated Learning (FL): A machine learning approach where model training occurs on decentralized data without centralizing the data itself.

Local Model: A model trained on a single tenant’s data.

Global Model: An aggregated model computed from multiple local model updates.

Privacy Budget: Parameters (epsilon, delta) that quantify the privacy guarantee provided by differential privacy mechanisms.

Tenant: An organizational entity with isolated data and computing resources within a multi-tenant deployment.

3. Problem Statement

Consider an AI agent system deployed across N organizational tenants. Each tenant generates behavioral data from user interactions with the agent. The system operator wishes to:

1. Train personalization models that adapt to user behavior

2. Leverage patterns across tenants to improve model quality

3. Maintain strict tenant data isolation

4. Comply with privacy regulations

Traditional centralized machine learning requires aggregating data from all tenants, violating requirements 3 and 4. Purely local training (each tenant trains independently) satisfies isolation but sacrifices the benefits of cross-tenant learning (requirement 2).

This document specifies an architecture that satisfies all four requirements through federated learning with differential privacy.

4. Architecture Overview

4.1. System Components

The architecture comprises the following components, illustrated in Figure 1:

- Local Data stores residing within each tenant boundary
- Local Model training infrastructure at each tenant
- Differential Privacy (DP) noise injection modules
- A central Aggregation Server for computing global model updates

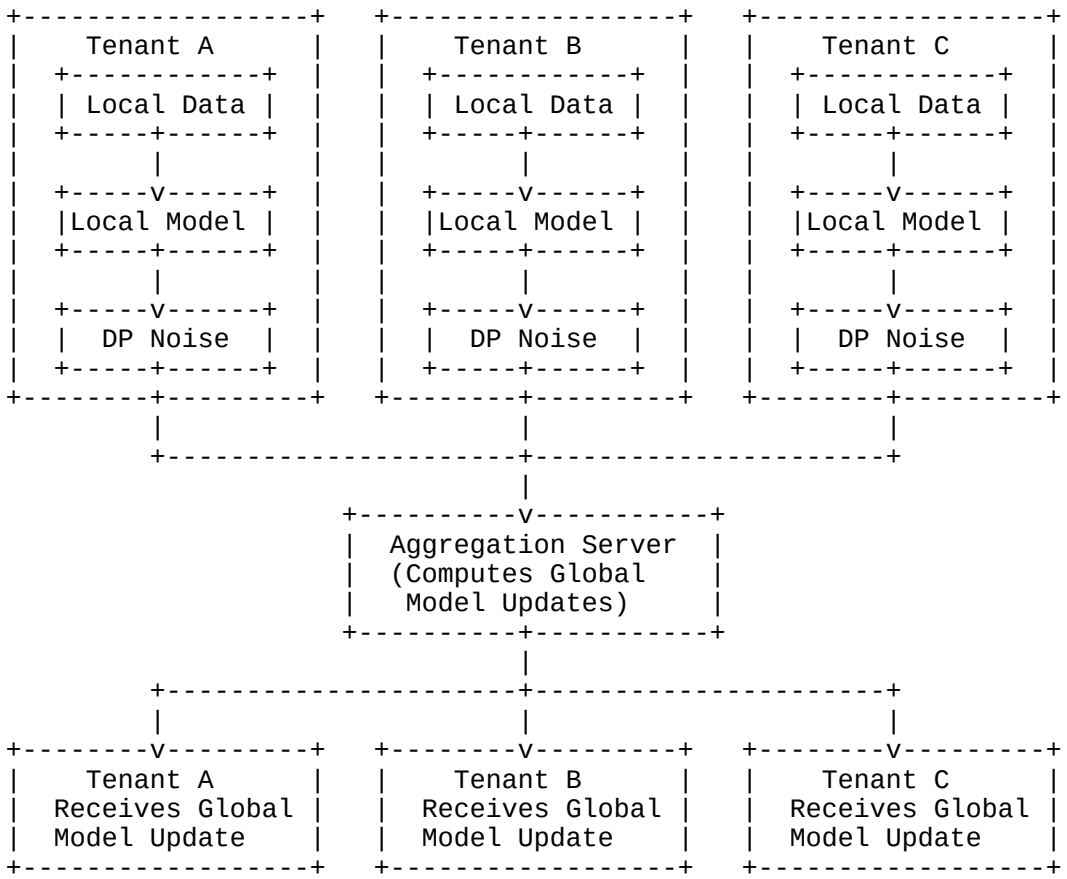

Figure 1: Federated Architecture

4.2. Data Flow

1. Each tenant trains a local model on tenant-specific data
2. Model updates (not raw data) are computed
3. Differential privacy noise is added to updates
4. Noisy updates are transmitted to aggregation server
5. Server computes weighted average of updates
6. Global model is distributed back to tenants
7. Process repeats for specified number of rounds

4.3. Trust Model

The architecture assumes:

- Aggregation server is honest-but-curious: It follows the protocol correctly but may attempt to infer information from received updates
- Tenants are honest: They train models correctly and do not attempt to poison the global model
- Communication channels are secure: TLS protects updates in transit

Section 7 discusses extensions for stronger threat models.

5. Federated Learning Protocol

This section specifies the federated learning protocol in detail. The protocol follows the FedAvg algorithm structure with modifications for differential privacy.

5.1. Initialization

The aggregation server MUST:

1. Generate initial global model parameters theta_0
2. Distribute theta_0 to all participating tenants
3. Specify privacy budget (epsilon, delta) for the training session
4. Specify number of training rounds T

5.2. Local Training Phase

For each round t, each tenant i MUST:

1. Receive current global model theta_t from aggregation server
2. Train local model on tenant data for E local epochs:
   theta_i = LocalTrain(theta_t, D_i, E)
3. Compute model update: delta_i = theta_i - theta_t
4. Clip update to bound sensitivity:
   delta_i_clipped = Clip(delta_i, C) where C is the clipping bound
5. Add Gaussian noise for differential privacy:
   delta_i_dp = delta_i_clipped + N(0, sigma^2 * I)
6. Transmit delta_i_dp to aggregation server

5.3. Aggregation Phase

The aggregation server MUST:

1. Receive noisy updates {delta_1_dp, ..., delta_n_dp} from tenants
2. Compute weighted average:
   delta_global = sum(w_i * delta_i_dp) where sum(w_i) = 1
3. Update global model: theta_{t+1} = theta_t + delta_global
4. Distribute theta_{t+1} to all tenants

5.4. Weighting Strategies

Tenant weights w_i MAY be computed based on:

- Population size: Larger tenants contribute proportionally more

- Data quality: Tenants with lower-variance updates receive higher weight
- Equal weighting: w_i = 1/n for all tenants

The specific weighting strategy SHOULD be documented in the deployment configuration.

6. Privacy Mechanisms

6.1. Differential Privacy Definition

A mechanism M satisfies (epsilon, delta)-differential privacy if for all datasets D and D' differing in one record, and all output sets S:

```
Pr[M(D) in S] <= e^epsilon * Pr[M(D') in S] + delta
```

6.2. Gaussian Mechanism

The Gaussian mechanism achieves (epsilon, delta)-DP by adding noise:

```
sigma >= C * sqrt(2 * ln(1.25/delta)) / epsilon
```

where C is the L2 sensitivity bound (clipping threshold).

6.3. Privacy Budget Allocation

For T training rounds with subsampling rate q, the total privacy budget follows composition theorems. Implementations SHOULD use advanced composition or Renyi differential privacy accounting for tighter bounds.

Recommended privacy parameters for enterprise deployments:

```
epsilon: 1.0 to 10.0 (depending on data sensitivity)
delta: 1/n where n is the minimum tenant population size
C (clipping bound): Determined empirically based on gradient
norms
```

6.4. Gradient Clipping

Before noise addition, model updates MUST be clipped:

```
delta_clipped = delta * min(1, C / ||delta||_2)
```

This bounds the sensitivity of individual data points, enabling precise privacy accounting.

7. Security Considerations

7.1. Threat Model

This architecture protects against:

- Honest-but-curious aggregation server attempting to infer tenant data from model updates
- External attackers observing aggregated models
- Membership inference attacks against the global model

7.2. Attacks Not Addressed

The basic architecture does NOT protect against:

- Malicious tenants submitting poisoned updates
- Collusion between aggregation server and tenants
- Model inversion attacks against the final trained model

7.3. Extensions for Stronger Security

7.3.1. Secure Aggregation

To protect against curious aggregation servers, implementations MAY use secure aggregation protocols where the server learns only the sum of updates, not individual tenant contributions. See [Bonawitz17] for protocol details.

7.3.2. Byzantine Fault Tolerance

To protect against malicious tenants, implementations MAY use Byzantine-resilient aggregation methods such as coordinate-wise median or trimmed mean.

7.4. Compliance Considerations

Implementations targeting GDPR compliance SHOULD:

- Document privacy budget selection rationale
- Maintain audit logs of aggregation operations
- Implement data subject access request procedures
- Specify data retention policies for model checkpoints

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9.2. Informative References

[Abadi16] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., and L. Zhang, "Deep Learning with Differential Privacy", Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016.

[Bonawitz17] Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., and K. Seth, "Practical Secure Aggregation for Privacy-Preserving Machine Learning", Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017.

[Dwork14] Dwork, C. and A. Roth, "The Algorithmic Foundations of Differential Privacy", Foundations and Trends in

Theoretical Computer Science, Vol. 9, No. 3-4, 2014.

[Kairouz21]
Kairouz, P., McMahan, H.B., Avent, B., et al., "Advances and Open Problems in Federated Learning", Foundations and Trends in Machine Learning, Vol. 14, No. 1-2, 2021.

[McMahan17]
McMahan, H.B., Moore, E., Ramage, D., Hampson, S., and B. Aguera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data", Proceedings of AISTATS, 2017.

[Rosenberg25]
Rosenberg, J. and C. Jennings, "Framework, Use Cases and Requirements for AI Agent Protocols", Work in Progress, Internet-Draft, draft-rosenberg-aiproto-framework-00, October 2025.

[SLIM25] Muscariello, L., Papalini, M., Sardara, S., and S. Betts, "Secure Low-Latency Interactive Messaging (SLIM)", Work in Progress, Internet-Draft, draft-mpsb-agntcy-slim-00, October 2025.

[Wang23] Wang, J., Charles, Z., Xu, Z., Joshi, G., McMahan, H.B., et al., "A Field Guide to Federated Optimization", arXiv:2107.06917, 2023.

Appendix A. Example Configuration

Example deployment configuration for enterprise AI agent system:

```
{
  "federated_learning": {
    "rounds": 100,
    "local_epochs": 5,
    "learning_rate": 0.01,
    "privacy": {
      "epsilon": 3.0,
      "delta": 1e-6,
      "clipping_bound": 1.0,
      "noise_multiplier": 1.1
    },
    "aggregation": {
      "method": "fedavg",
      "weighting": "population_proportional",
      "min_tenants_per_round": 10
    }
  }
}
```

Author’s Address

Nik Kale
Cisco Systems, Inc.
3700 Cisco Way
San Jose, CA 95134
United States of America

Email: nikkal@cisco.com