Network Working Group
Internet-Draft
Intended status: Informational
Expires: 23 January 2026

A. Akhavain
H. Moussa
Huawei Canada

22 July 2025

AI Network for Training, Inference, and Agentic Interactions
draft-akhavain-moussa-ai-network-00

Abstract

Artificial Intelligence (AI) is rapidly reshaping industries and daily life, driven by advances in large language models (LLMs) such as ChatGPT, Claude, Grok, and DeepSeek. These models have demonstrated the transformative potential of AI across diverse applications, from productivity tools to complex decision-making systems. However, the effectiveness and reliability of AI hinge on two foundational processes: training and inference. Each presents unique challenges related to data management, computation, connectivity, privacy, trust, security, and governance. This document introduces the Data Aware-Inference and Training Network (DA-ITN)—a unified, intelligent, multi-plane network architecture designed to address the full spectrum of AI system requirements. DA-ITN provides a scalable and adaptive infrastructure that connects AI clients, data providers, service facilitators, and computational resources to support end-to-end AI lifecycle operations. The architecture features dedicated control, data, and operations & management (OAM) planes to orchestrate training and inference while ensuring reliability, transparency, and accountability. By outlining the key requirements of AI systems and demonstrating how DA-ITN fulfills them, this document presents a vision for the future of AI-native networking—an "AI internet"—optimized for continuous learning, scalable deployment, and seamless agent-to-agent collaboration.

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 23 January 2026.

Table of Contents

1. Introduction

AI has become a major focus in recent years, with its influence rapidly expanding from everyday tasks like scheduling to complex areas such as healthcare. This growth is largely driven by advances in large language models (LLMs) like ChatGPT, Claude, Grok, and DeepSeek, which are now widely used for tasks such as brainstorming, editing, coding, and data analysis. These real-world applications highlight AI's transformative power to boost productivity and simplify life. It's clear that AI is not a passing trend but a lasting and evolving force.

However, it is crucial to recognize that the success of AI systems relies on two fundamental pillars: training and inference. Both of these pillars have a number of factors and moving parts that need to be carefully coordinated, designed, and managed to ensure accuracy, resilience, usability, continuous evolution, trustworthiness, and reliability. Moreover, once deployed, AI systems must be continuously monitored and governed to safeguard user safety and societal well-being.

As such, aspects such as data management, computational resources, connectivity, security, privacy, trust, billing, and rigorous testing are all crucial when handling AI systems. Thus, it is important to clearly understand the requirements of the AI systems from both the training and inference prospective as both of these pillars constitute an entangled framework and cannot be tackled in isolation.

In this document, we present a vision of an ecosystem, especially designed to satisfy the requirements of AI from training and inference points of view. We propose a unified, intelligent network architecture— the Data Aware-Inference and Training Network (DA-ITN). This ecosystem is envisioned as a comprehensive, multi-plane network with dedicated control, data, and operations & management (OAM) planes. It is designed to interconnect all relevant stakeholders, including clients, AI service providers, data providers, and third-party facilitators. Its core objective is to provide the infrastructure and coordination necessary to support an ecosystem for enabling AI of the future at scale.

This document aims to introduce the DA-ITN vision and establish a compelling case for its central role in enabling a new generation of AI-native networks, i.e., AI internet. These networks will be optimized not only for learning and inference but also for seamless collaboration, interaction, and communication among AI agents. To that end, we begin by outlining the specific requirements of AI from both the training and inference standpoints. We then introduce the core components of the DA-ITN and illustrate how they collectively meet these requirements. Finally, this network is positioned as an ecosystem for agent-to-agent collaborations, interactions, and communications.

2. Training Requirements

AI model training is the foundational process through which an artificial intelligence system learns to perform tasks by analyzing data and adjusting its internal parameters— typically the weights in neural networks— to minimize prediction errors. At its core, this process involves feeding input data into a model, and applying optimization algorithms to iteratively refine the model' s performance. Among the most influential outcomes of this process are foundation models, such as ChatGPT and its peers, which are capable of performing a wide range of tasks across domains. Training these models now occurs at an unprecedented scale, requiring massive compute infrastructure, enormous amounts of training data, high-speed interconnects, and parallelized training frameworks (e.g., data, model, and pipeline parallelism).

2.1. Centralized versus Decentralized Training

It is clear from the above that no matter how advanced the model architecture may be, the success of any training process ultimately hinges on two fundamental components: the model and the data. While the model itself is often developed and hosted in a centralized location— typically within the secure infrastructure of the model owner or designer— data is inherently distributed. It originates from sensors, devices, logs, events, documents, and other diverse sources spread across different geographies and domains. To be exact, whether due to geographic dispersion, organizational silos, privacy constraints, or edge-device generation, data rarely exists in a single, clean repository.

Today, model training can happen in one of two ways or a combination thereof: centralized or decentralized. In centralized training, thanks to the development of robust data collection techniques and high-throughput connectivity networks, it is now feasible to collect data and bring it to where the model training would occur. This traditional approach is often referred to as model-centralized

training. On the other hand, a more recent paradigm known as model-follow-data has emerged, advocating for the reverse: rather than transporting large volumes of potentially sensitive data to a central location, the model is dispatched to where the data resides—enabling distributed or federated training.

Accordingly, to facilitate the training process, rendezvous points scheduling between distributed data, compute and storage resources, and an AI model awaiting training needs to be arranged and managed, which is fundamental for successful model training. However, this scheduling process introduces a number of challenges spanning privacy, trust, utility, and computational and connectivity resources management. Moreover, as AI adoption accelerates, both centralized and decentralized approaches will drive increasing pressure on underlying connectivity infrastructure. Therefore, to ensure scalable, efficient, and cost-effective AI training, it is vital to implement intelligent mechanisms for managing data and model movement, selecting relevant subsets for training, and minimizing unnecessary transfers.

In the sections that follow, we explore the architectural and operational requirements needed to support this vision and lay the foundation for a high-performance, AI-native training ecosystem.

2.2. Requirements Breakdown

Consider a number of AI model training clients awaiting training service. An AI model training client is a user with a raw or a pre-trained model who wishes to train or continue training their AI model using data that can be found in the data corpus. The data corpus (the global dataset), as has been previously established, consists of a group of datasets that are distributed across various geographical locations. AI clients require access to this data either in a centralized or distributed manner.

2.2.1. Data Collection/Model Dispatching

As previously discussed, data is inherently distributed. In centralized training paradigms, this data must be transferred from its sources to centralized locations where model training occurs. Consider a scenario involving multiple clients, each awaiting centralized training of AI models using distinct data sets of interest. Aggregating large volumes of data from geographically dispersed sources to centralized servers introduces several significant challenges:

* Communication Overhead: The sheer volume of data to be transmitted can place substantial strain on the underlying transport networks, resulting in increased latency and bandwidth consumption.

* Redundant Knowledge Transfer: Despite originating from different sources, data sets may carry overlapping or identical knowledge content. Transmitting such redundant content leads to unnecessary duplication, wasting resources without providing additional training value.

* Timely Delivery: In certain applications, the freshness of data is critical. Delays in transmission can degrade the value of the information, as these applications are sensitive to the Age of Information (AoI)— the time elapsed since data was last updated at the destination.

* Multi-Modal Data Handling: Data often exists in various formats— such as text, images, audio, video, etc— each with distinct transmission requirements. Ensuring accurate and reliable delivery of these diverse data types necessitates differentiated Quality of Service (QoS) levels tailored to the characteristics and sensitivity of each modality.

* Heterogeneous Access Media: Data may reside across diverse communication infrastructures— for example, some data may be accessible only via 3GPP mobile networks, while other data may be confined to wireline networks. Coordinating data collection across these heterogeneous domains, while maintaining synchronization and consistency, presents a significant operational challenge.

Importantly, many of these challenges are alleviated in decentralized training frameworks, where data remains local to its source and is not transferred over the network. Instead, the model itself is distributed to the various data locations. However, this alternate paradigm introduces its own set of unique challenges.

As previously noted, modern AI models are growing increasingly large in size. In decentralized training, it is often necessary to replicate the model and transmit it to multiple, geographically dispersed data sites. This results in a different but equally significant set of logistical and technical hurdles:

* Communication Overhead: While data transfer is avoided, dispatching large model files across the network to multiple destinations can still impose substantial load on communication infrastructure, particularly in bandwidth-constrained environments.

* Redundant Knowledge Transfer: Data residing at different locations may share overlapping knowledge content. Sending models to multiple sites with redundant knowledge content leads to inefficient use of network resources. In some cases, even when knowledge content is only partially redundant, it may be more efficient—considering communication cost—to forego marginal training benefits in favor of reduced overhead.

* Timeliness and Data Freshness: In certain applications, the Age of Information (AoI) remains critical. Prioritizing model dispatch to data sources with soon-to-expire or time-sensitive information is essential to maximize the utility of training and to maintain up-to-date model performance.

2.2.2. Data and Resource Discovery

Given the distributed nature of data, there must be a mechanism through which data owners can advertise information about their datasets to AI model training clients. This requires the ability to describe the characteristics of the data—such as its knowledge content, quality, size, and Age of Information (AoI)—in a way that allows AI clients to discover and evaluate whether the data aligns with their training objectives. Training objectives can be one or more of: target performance, convergence time, training cost, etc.

Crucially, this discovery process may need to operate across multiple network domains and heterogeneous communication infrastructures. For example, an AI training client operating over a wireline connection may be interested in data residing on a 3GPP mobile network. This raises an important question: How can data owners effectively advertise their datasets in a way that is discoverable across diverse domains? To enable such cross-domain data visibility and discovery, the following key requirements must be considered:

* Data Descriptors: These are metadata objects used by data owners to reveal essential information about their datasets to AI clients. Effective data descriptors must be self-contained, privacy-preserving, and informative enough to support decision-making by training clients. They should allow data owners to selectively disclose details about their data— such as type, relevance, quality metrics, freshness, and perhaps cost of utility— while concealing sensitive or proprietary information (privacy preservation). Data descriptors also need to be easily modified as data can be dynamic, and the change in data needs to be effectively reflected into the data descriptions. To ensure interoperability, data descriptors can either follow a standardized format or adopt a flexible but well-defined structure that enables consistent interpretation across different systems and domains.

* Data Discovery Mechanisms: These refer to the processes by which AI training clients locate and identify datasets across potentially vast and heterogeneous environments. An effective discovery mechanism should support global-scale searchability and cross-domain operability, allowing clients to find relevant datasets regardless of where they reside or which communication infrastructure they are accessible through. Discovery protocols may be standardized within specific domains (e.g., mobile networks, IoT platforms) or designed to function interoperable across multiple domains, enabling seamless integration and visibility. It should also be highlighted that, discovery mechanisms should be considerably up-to-date with the changes that would occur as the underlying data changes dynamically.

* Data Relationship Maps: Training often requires identifying groups of datasets that collectively meet specific requirements. Evaluating each dataset in isolation may be insufficient. Instead, a mechanism is needed to establish relationships among datasets, enabling AI training clients to assemble the appropriate combination of data for their tasks. These relationships can be envisioned to look like maps or topologies. This is a crucial step as if an AI model client was not able to find the right dataset that satisfies its requirements, the client might choose not to submit the model for training at this time which may reduce resource wastage from the get go.

* Timely reporting: Given the dynamic nature of data availability, characteristics, and accessibility, it is essential to have advertisement mechanisms that can promptly reflect any changes. Real-time or near-real-time updates ensure that the AI training process remains aligned with the most current data conditions, thereby maximizing both effectiveness and accuracy. Timely

reporting helps prevent training on outdated or irrelevant data and supports optimal decision-making in model selection and training pipeline configuration.

Additionally, it should be highlighted that in AI training, discovering data alone is not enough. For instance, third-party resources like compute and storage are essential, and the providers of those resources must be able to advertise their capabilities so AI clients can locate and utilize them effectively. Just like with data, resource discovery requires descriptors, multi-domain accessibility, and timely updates to support seamless coordination between models, data, and infrastructure. It should be highlighted that data and resource discovery is essential in both centralized and decentralized training, as both can be done on third party infrastructure.

2.2.3. Mobility and Service Continuity Handling

In some decentralized training applications, AI models are designed to traverse a predefined route, training on multiple datasets in a sequential or federated manner. This introduces the need to manage model mobility. However, the underlying data landscape is often dynamic— new data is continuously generated, existing data may be deleted, or datasets may be relocated to different nodes or domains.

As a result, enabling reliable model mobility in such a fluid environment requires robust mobility management mechanisms. For instance, while a model is en-route to a specific data location for training, that dataset may be moved elsewhere. In such cases, the model must either be re-routed to the new location or redirected to an alternative dataset that satisfies similar training objectives.

Additionally, since training occurs on remote compute infrastructure and can be time-intensive, unexpected resource shutdowns or failures may interrupt the process. These interruptions can lead to service discontinuity, which must be addressed through mechanisms such as checkpointing, fallback resource selection, or dynamic rerouting of model or data to maintain training progress and system reliability.

Additionally, model mobility may involve training on datasets that are distributed across heterogeneous communication infrastructures. Some infrastructures, such as emerging 6G networks, offer built-in mobility support— for example, when data resides on mobile user equipment (UE), its location can be tracked using native features of the network. However, such mobility handling capabilities may not exist in other infrastructures, such as traditional wireline networks or legacy systems, making seamless model movement and data access more challenging in those environments.

2.2.4. Privacy, Trust, and Data Ownership and Utility

Privacy and trust are mutual responsibilities—both data owners and model owners must be protected. Granting clients access to data for training and knowledge building should be a regulated process, with mechanisms to track data ownership and future use. Initial discussions on this topic have taken place in forums such as the AI-Control Working Group.

Equally important is ensuring that model owners are protected from data poisoning. They must have confidence that the datasets they use are accurately described and not misrepresented. If data owners provide false metadata—intentionally or otherwise—model owners may unknowingly train on unsuitable or harmful datasets, leading to degraded model performance. To safeguard both parties, innovative verification and enforcement mechanisms are needed. Technologies like blockchain could offer potential solutions for establishing trust and accountability, but further research and exploration are necessary to develop practical frameworks.

2.2.5. Testing and Performance Management

Another critical aspect of training is testing and performance evaluation, typically carried out using a separate subset of the data known as the testing dataset. This dataset is not used to update the model's weights but to assess its performance on unseen samples. In centralized training, this process is straightforward because all data resides in a single, accessible location, making it easy to partition the dataset into training and testing subsets. However, in distributed training environments, where data is spread across multiple locations or devices, creating a representative and unbiased testing dataset without aggregating the data centrally becomes a major challenge. Developing effective, privacy-preserving methods for testing in such settings requires innovative solutions

2.2.6. QoS Guarantee

Beyond ensuring traditional Quality of Service (QoS) for data transmission, a new dimension of QoS must be considered—the QoS of training itself. In AI training workflows, it is crucial to guarantee that key performance indicators (KPIs) related to training, such as accuracy convergence, training time, and resource utilization, are met consistently. This raises several important questions: * How can these training KPIs be guaranteed in dynamic or distributed environments?

* What mechanisms can be used to monitor and track training performance in real time?

* Should AI training be treated like best-effort traffic, where no guarantees are made and resources are allocated as available?

* Or should training tasks receive prioritized or differentiated service levels, similar to high-priority traffic in traditional networks?

Addressing these questions is essential to ensure predictable and reliable AI model development, especially as training workloads grow in complexity and scale. It may require introducing new QoS frameworks tailored specifically to the needs of AI training systems.

2.2.7. Charging and Billing

The AI training process involves a diverse ecosystem of stakeholders, including data owners, model owners, and resource providers. Each of these parties plays one or more vital roles in enabling successful training workflows.

For example, communication providers contribute not only by transporting data and models across the network but also they themselves may also serve as data providers. This is particularly evident in the emerging design of 6G networks, which integrate sensing capabilities with communication infrastructure. As a result, 6G operators are uniquely positioned to offer both connectivity and data, making them central players in the training pipeline.

Despite their different roles, all parties contribute to enabling AI training as a service, a complex and resource-intensive process that is far from free. Therefore, it is essential to establish a robust charging and billing framework that ensures each participant is fairly compensated based on their contribution.

Several open questions arise in this context:

* Should training services follow a prepaid model, or adopt a pay-per-use structure?

* Will there be tiered service offerings, such as gold, silver, and platinum, each providing different levels of performance guarantees or priority access?

* How should these tiers be defined and enforced in terms of service quality, resource allocation, and response time?

Developing fair, transparent, and scalable billing mechanisms is critical to facilitating collaboration across stakeholders and sustaining the economic viability of distributed AI training

ecosystems. These challenges call for further research into incentive structures, dynamic pricing models, and smart contract-based enforcement, especially in scenarios involving cross-organizational or cross-network cooperation.

3. Inference

Inference is critical because it represents the phase where the model begins to deliver practical value. Unlike training, which is typically a one-time or periodic, resource-intensive process, inference often needs to operate continuously and efficiently, sometimes in real-time. Although inference is a less resource-intensive process, it has strict requirements that govern its success. In what follows, we explore these requirements that shall enable a successful AI inference ecosystem.

3.1. Requirement Breakdown

We envision an inference ecosystem composed of a large number of pre-trained AI models (or agents) distributed across the globe. These models are capable of performing a wide range of tasks, such as image classification, language translation, or speech recognition. Some models may specialize in the same task but vary in performance, accuracy, latency, or resource demands. This diverse pool of models is accessed by numerous inference clients (users or applications) who submit inputs, referred to as queries, and receive task-specific outputs.

These queries can vary greatly in complexity, structure, and modality, with some requiring the cooperation of multiple models to fulfill a single request. The overarching goal of the ecosystem is to efficiently match incoming queries with the most suitable models, ensuring accurate, timely, and resource-aware responses. Achieving this requires intelligent orchestration, load balancing, and potentially dynamic model selection based on factors such as performance, availability, cost, and user-specific requirements. In what follows, we discuss the various aspects of this ecosystem and discuss the different requirements needed for its success.

3.1.1. Model Deployment and Mobility

The first step toward building a successful AI inference ecosystem is the optimal deployment of trained models, or agents. In this context, optimality refers to both the physical or network location of the model and the manner in which it is deployed. AI models vary significantly in size and resource requirements— ranging from lightweight models that are only a few kilobytes to large-scale models with billions of parameters. This wide range makes deployment

decisions critical to achieving both efficient performance and effective resource utilization. Also, a unique factor to AI models/agents is the fact that they are software components that are not bounded to a certain hardware. They can be deleted, copied, moved, or split across multiple compute locations. All these unique aspects provide flexibility in design if the real-time status of the underlying network dynamics and resources is made accessible.

* Choosing the right facility to host a model: whether it's a lightweight edge device, a local server, or a high-performance cloud data center, deployment will depend on the model's size, computational requirements, and expected query volume. For example, smaller models might be best suited for deployment on edge devices closer to users, enabling low-latency responses. In contrast, larger models may require centralized or specialized infrastructure with high compute and memory capacity.

* Load balancing: Once models are deployed, inference traffic begins to flow, with users or applications sending queries to the appropriate agents. If not managed properly, this traffic can lead to congestion, creating bottlenecks that degrade inference performance through increased latency or dropped requests. To avoid such scenarios, models should be deployed strategically to distribute the load, ensuring smooth operation. Traditional load balancing techniques can be employed to redirect traffic away from overburdened nodes and towards underutilized ones. However, more sophisticated strategies may involve replicating models and placing these replicas closer to regions with high query demand, thereby minimizing latency and easing network traffic engineering challenges.

* Mobility-aware deployment: the dynamic nature of inference traffic necessitates mobility-aware deployment. For instance, consider a large data center acting as a centralized inference hub, hosting numerous models and handling a significant volume of queries. Over time, this hub may experience traffic overload. In such cases, migrating certain models to alternative locations can help alleviate pressure. However, model migration is not without its challenges— particularly if a model is actively serving queries at the time of migration. In such situations, mobility handling mechanisms must be in place to ensure seamless service continuity. These mechanisms could involve session handovers, temporary state preservation, or model version synchronization, all designed to maintain uninterrupted service during the migration process.

In summary, optimal model deployment requires careful consideration of model size, resource needs, query distribution, and real-time adaptability. Achieving this lays the foundation for a responsive, scalable, and resilient AI inference ecosystem.

3.1.2. Model Discovery and Description

Just as data descriptors and discovery mechanisms are essential during the training phase, AI model inference clients also require a robust discovery mechanism during the inference stage. In an ecosystem populated by a large and diverse pool of models—each with unique capabilities and specializations—clients are presented with significant flexibility and choice in selecting the most suitable models for their queries. However, to make informed decisions, clients must have access to information that enables them to distinguish between models based on criteria such as performance, specialization, availability, and resource requirements.

This discovery process becomes even more complex when it needs to function across multiple network domains and heterogeneous communication infrastructures. For instance, a client connected via a wireline network might need to interact with a model deployed on a mobile 3GPP network. Such scenarios raise a critical question: How can model owners advertise their models in a way that ensures discoverability and interoperability across diverse domains?

Addressing this challenge requires the development of standardized model advertisement and discovery protocols that can operate seamlessly across infrastructure boundaries. These protocols must accommodate differences in network technology, latency constraints, and security requirements while providing consistent and reliable access to model information. Ensuring cross-domain discoverability is crucial to unlocking the full potential of a globally distributed inference ecosystem.

To enable such cross-domain model visibility and discovery, the following key requirements must be considered:

* Model Descriptors: These are metadata objects used by model owners to reveal essential aspects about their datasets to AI inference clients. Effective data descriptors must be self-contained, privacy-preserving, and informative enough to support decision-making by inference clients. They should allow model owners to selectively disclose details about their model— such as skills, performance reviews, trust level, relevance, quality metrics, freshness, and perhaps cost of utility— while concealing sensitive or proprietary information. To ensure interoperability, model descriptors can either follow a standardized format or adopt a flexible but well-defined structure that enables consistent interpretation across different systems and domains.

* Model/agent Discovery Mechanisms: These refer to the processes by which AI inference clients locate and identify models/agents across potentially vast and heterogeneous environments. An effective discovery mechanism should support global-scale searchability and cross-domain operability, allowing clients to find relevant model/agents regardless of where they reside or which communication infrastructure they are accessible through. Discovery protocols may be standardized within specific domains (e.g., mobile networks, IoT platforms) or designed to function interoperable across multiple domains, enabling seamless integration and visibility.

* Model/agent relationship maps: As queries may requiring the collaboration between multiple models/agents, relationships between models/agents with respect to different task might present useful tools as to help clients choose the appropriate subset of models/agents that would handle their queries.

* Timely Reporting: Similar to data, the status of a model can change over time— for example, due to shifts in workload or resource availability. It is important that such changes are reported promptly and accurately, allowing clients to make informed decisions based on the model’ s current state. This is essential for ensuring efficient model selection and maintaining high-quality, reliable inference outcomes.

It is important to emphasize that model discovery differs fundamentally from data discovery. While data are passive objects that require external querying or manipulation, models are intelligent, autonomous agents capable of making decisions based on their own capabilities, status, and context. This distinction opens up new and more dynamic possibilities for how models are discovered and engaged in an inference ecosystem.

In traditional data discovery, clients search for and retrieve relevant datasets based on metadata or predefined criteria. However, in the case of model discovery, the process can be much more interactive and flexible. One approach involves the client actively discovering models by querying a directory or registry using model descriptors. Based on these descriptors, the client selects one or more models to handle a specific inference task. However, given that models can reason and act independently, model discovery does not have to be limited to client-driven selection. An alternative approach is to reverse the flow of interaction. Instead of clients seeking out models, they can publish their tasks to a shared task pool, accessible to all available models. These tasks include descriptors that define the type of work to be done, expected outputs, and quality-of-service requirements. Models can then autonomously scan this pool, evaluate whether they are well-suited for specific tasks, and choose to express interest in executing them. This self-selection process allows models to play an active role in task matching, improving system scalability and efficiency.

The final assignment of a task can be handled in different ways. Clients may retain full control and approve or reject interested models based on their preferences or priorities. Alternatively, the system may operate in a fully autonomous mode, where tasks are assigned automatically to the first or best-matching model, without requiring client intervention—depending on the client’s chosen policy.

This agent-driven paradigm reflects the shift toward more decentralized and intelligent AI ecosystems, where models are not merely passive computation endpoints but active participants in task negotiation and resource allocation. Such a system not only enhances scalability and flexibility but also allows for more efficient utilization of the available model pool, especially in heterogeneous and dynamic environments.

3.1.3. Query and Inference Result Routing

A significant challenge in AI inference networks lies in efficiently routing client queries to the appropriate inference models and ensuring the corresponding results are reliably delivered back to the client. This becomes particularly complex in scenarios involving mobility and multi-domain environments, where both the client and the model may exist across different types of network infrastructures. The key challenges and considerations include:

* Query Routing Across Heterogeneous Networks: When a client accesses the inference ecosystem through a mobile network such as 3GPP 6G, and the target model is hosted in a wireline or cloud-

based infrastructure, routing the query across these distinct domains is non-trivial. Differences in network architecture, protocols, and service guarantees complicate the end-to-end flow.

* Mobility Management During Inference Execution: While mobile networks like 6G are designed to handle user mobility, inference tasks may take time to process—particularly when using large models or performing complex computations. During this time, the client may change physical location, switch devices, or even go offline. Ensuring that inference results can still reach the client under these dynamic conditions poses a significant challenge.

* Handling Client State Changes: If a client becomes idle or disconnects entirely during inference, the system must decide what to do with the completed result. Should it be queued, buffered, forwarded to another linked device, or simply discarded? A robust mechanism is needed to track client state, maintain context, and guarantee result delivery or at least graceful degradation.

* Support for Live and Streaming Inference: Some use cases, such as real-time audio transcription, involve live streaming of data from the client to the model and vice versa. These sessions require sustained, low-latency connections and are particularly sensitive to interruptions caused by mobility or handoffs between networks. Ensuring session continuity and maintaining streaming quality across network boundaries is a complex but critical aspect of real-world inference deployments.

* Cross-Domain Connectivity and Session Management: The involvement of multiple network operators and domains introduces questions around interoperability, session tracking, and handover coordination. There is a need for intelligent infrastructure capable of end-to-end session management, including maintaining metadata, context, and service quality as the session traverses' different networks.

3.1.4. Inference Chaining/Collaborative Inference

Another critical aspect of an AI inference ecosystem is the need for model collaboration to fulfill complex or multi-faceted tasks. Not all inference requests can be handled by a single model; in many cases, collaboration between multiple models is necessary. Effectively managing this task-based collaboration is essential to ensure accurate, efficient, and scalable inference services. Model collaboration can take several distinct forms:

* Inference Chaining: In this model, the output of one model serves as the input to the next in a sequential pipeline. Each model performs a specific stage of the task, and the final result—produced by the last model in the chain—is returned to the client. This is common in multi-stage tasks such as image processing followed by object detection and then classification.

* Parallel Inference: Here, a complex task is decomposed into multiple subtasks, each of which is assigned to a specialized model. These models operate concurrently, and their outputs are aggregated to form a unified inference result. This approach is particularly useful when dealing with large data sets or when a task spans different domains of expertise.

* Hierarchical inference: A model is assigned as a task manager and is responsible for delegating tasks to service models

* Collaborative Inference: In this more dynamic and decentralized form, the task is assigned to a group of models that are capable of discovering one another, assessing their respective capabilities, and coordinating among themselves to devise a shared strategy for completing the task. This model requires more sophisticated communication, negotiation, and orchestration mechanisms.

Regardless of the collaboration format, the success of such multi-model interactions depends on the availability of a robust management infrastructure. This infrastructure must enable seamless coordination between models, even when:

* The models are hosted by different providers,

* They are deployed across heterogeneous communication networks,

* They use varying protocols, or

* They have differing performance characteristics.

Such a management system must abstract away the underlying complexities and provide standardized interfaces, discovery mechanisms, communication protocols, and coordination frameworks that allow models to interact effectively. Without this, collaborative inference would be brittle, inefficient, or impossible to scale. In essence, the ability to orchestrate model collaboration across diverse environments is a cornerstone of a flexible, intelligent, and robust AI inference ecosystem.

3.1.5. Compute and Resource Management

In many scenarios, the compute infrastructure used to host and run inference models is managed by third-party providers, not the model owners themselves. These compute providers are responsible for meeting the Quality of Service (QoS) levels agreed upon with the model owners—such as latency, uptime, throughput, and reliability.

* Ensuring these service levels are consistently met raises the question of accountability. If performance degrades due to compute resource issues—such as overloaded hardware or network outages—who is responsible for the failed inference tasks?

* There must be clear, enforceable service-level agreements (SLAs) that define roles, responsibilities, and penalties for non-compliance.

* Mechanisms for performance monitoring, auditing, and dispute resolution need to be integrated into the ecosystem to make such arrangements viable and trustworthy.

3.1.6. Privacy Preservation and Security

While models are the intellectual property of their owners, they may operate on infrastructure owned by others. This raises significant concerns around privacy and intellectual property protection.

* Sensitive model details such as architecture, weights, and optimization strategies must be protected from exposure or reverse engineering by untrusted compute hosts.

* Techniques such as secure computing, encrypted model execution, and remote attestation protocols may be necessary to ensure that models run securely without revealing proprietary details.

* Model owners must also be assured that inference inputs and outputs remain confidential, particularly in applications involving personal or sensitive data.

3.1.7. Utility Handling and QoS Requirements

Utility handling refers to the regulation, protection, and fair governance of how models are used, accessed, and monitored throughout the ecosystem. This encompasses several critical questions:

* How can we guarantee that a model deployed on remote infrastructure is not being tampered with, copied, or intentionally repurposed?

* How do we ensure that workload distribution is fair across available models, preventing monopolization by a few and giving equal visibility and opportunity to all participating models?

* What protections are in place to ensure that models are not being poisoned, exploited, or involved in illegal activities, either through malicious inputs or untrusted outputs?

* How do we ensure the integrity of inference results, so that outputs are delivered to clients without alteration, manipulation, or censorship? Addressing these concerns may require digital rights management (DRM) for AI models, usage monitoring tools, and potentially blockchain-based logging or audit trails to ensure transparency and traceability.

On the other hand, the definition of Quality of Service (QoS), when it comes to inference tasks, is very broad and can take many forms. For instance, QoS could be to guarantee a certain accuracy of a response, or time of the response, or expertise level needed. We believe that the topic of QoS guarantee requires extensive studying and analysis.

3.1.8. Model Upgrade Streamlining

AI models are not static; they undergo continuous upgrades, improvements, and fine-tuning to maintain accuracy, adapt to new data, or support evolving tasks.

* The ecosystem must support seamless model versioning, including adding, removing, or modifying model agents without disrupting ongoing services.

* Updated model profiles must be instantly reflected in the discovery layer, ensuring clients always have access to the most current and accurate model descriptions.

* For large models, upgrade procedures must be efficient and bandwidth-conscious, potentially using incremental update techniques to avoid full redeployment.

* Moreover, strategies must be in place to handle hot-swapping of models, where an old model is gracefully decommissioned and replaced by a new one—without causing inference failures or data loss during the transition.

3.1.9. Charging and Billing

The AI inference process involves a diverse ecosystem of stakeholders, including model owners, compute providers, and communication providers. Each of these parties plays one or more vital roles in enabling successful inference workflows. Therefore, it is essential to establish a robust charging and billing framework that ensures each participant is fairly compensated based on their contribution.

Several open questions arise in this context:

* Should inference services follow a prepaid model, or adopt a pay-per-use structure?

* Will there be tiered service offerings—such as gold, silver, and platinum—each providing different levels of performance guarantees or priority access?

* How should these tiers be defined and enforced in terms of service quality, resource allocation, and response time?

* What about discovery framework providers? Would they be offering a free service like google search or would it be more structured?

Developing fair, transparent, and scalable billing mechanisms is critical to fostering collaboration across stakeholders and sustaining the economic viability of distributed AI training ecosystems. These challenges call for further research into incentive structures, dynamic pricing models, and smart contract-based enforcement, especially in scenarios involving cross-organizational or cross-network cooperation.

4. Data Aware Inference and Training Network (DA-ITN): General Framework

The DA-ITN is envisioned as a multi-domain, multi-technology network operating at the AI layer, designed to address the various layers of complexity inherent in modern AI ecosystems. It aims to support a wide range of requirements across AI training, inference, and agent-to-agent interaction, as previously outlined. To manage these complexities and cater for the requirements, we propose structuring the DA-ITN around four core components: a Control Plane (CP), a Data Plane (DP), an Operations and Management (OAM) Plane, and an Intelligence Layer. It is important to note that the DA-ITN is agnostic to the underlying communication infrastructure, allowing it to operate seamlessly over heterogeneous networks, whether mobile, wireline, or satellite-based. he DA-ITN integrates with these

underlying infrastructures through any available means, embedding its control and intelligence capabilities to coordinate and manage AI-specific services in a flexible and scalable manner.

4.1. Control plane and Intelligence Layer

The Control Plane and Intelligence Layer work together to enable an efficient, reliable, and timely information collection infrastructure. They continuously gather up-to-date information on data availability, model status, agent conditions, resource utilization, and reachability across all participating entities. The collected information comes in the form of dynamic descriptors for data, models, and resources, essential components for enabling intelligent, context-aware decision-making within the AI ecosystem as has previously been highlighted. Also, with the help of data, resource, and reachability topology engine (DRRT) housed within the intelligence layer, the gathered information and descriptors can be used to construct meaningful relationships across the ecosystem. These are captured in the form of dynamic topologies or map-like structures, which help optimize decision-making processes across training, inference, and agent-to-agent collaboration tasks. This design provides a continuous awareness that is very essential for the success, reliability, accuracy, and responsiveness of the AI functionalities and services enabled by the DA-ITN within the AI ecosystem.

The DA-ITN control plane also lays a foundation for an advanced discovery infrastructure where the generated descriptors can be made easily accessible to all authorized participants to facilitate their required AI service For example, AI clients subscribed to training services can access up-to-date data descriptors and resource topologies, enabling them to select appropriate datasets and compute resources that align with their performance and accuracy goals. Similarly, inference clients or agents seeking collaboration can discover models based on capabilities, or submit task descriptors that enable models to respond intelligently and autonomously.

Aside from descriptor collection, topology creation, and discovery, the DA-ITN control plane also supports a secure and trusted environment where clients, data providers, model providers, and resource providers can engage in AI processes without compromising integrity or accountability. It also plays a key role in managing charging, billing, and rights enforcement, ensuring that all contributors to the AI service chain are fairly compensated and protected.

It is worth noting that the DA-ITN's Control Plane is not constrained by specific protocol stacks. Instead, it provides a flexible connectivity and coordination infrastructure upon which various AI-related protocols—such as Agent-to-Agent (A2A), Model Control Protocol (MCP), or AI Coordination Protocol (ACP)—can operate. Regardless of the protocol used, implementations must meet the core DA-ITN requirements, including timely information exchange, flexible descriptor encapsulation, support for multi-model and multi-domain environments, and robust security and privacy protections. The DA-ITN is also designed to support both centralized and decentralized modes of operation, offering high adaptability across different deployment contexts.

4.2. Data Plane

On the other hand, the Data Plane of the DA-ITN provides support for mobility management and intelligent scheduling, enabling the dynamic creation of rendezvous points where data, queries, models, and compute infrastructure can be brought together with minimal latency and overhead. Thanks to its infrastructure-agnostic nature, the DA-ITN leverages existing communication networks—such as those offered by 6G or edge service providers—as tools to enable model mobility, data mobility, and agent-to-agent coordination. This capability is essential for supporting scenarios where mobility or geographical dispersion of resources would otherwise lead to performance degradation or inefficiency.

4.3. Operation and Management Plane (OAM)

Finally, the Operations and Management (OAM) layer plays a critical role in supporting the day-to-day operational needs of the AI ecosystem. This layer is responsible for a wide range of essential functions, including monitoring, registration, configuration, fault management, and lifecycle maintenance of models, data, and services. It serves as the management backbone of the DA-ITN, ensuring transparency, accountability, and operational control throughout the system.

Consider the scenario of an AI model training client deploying a model into the ecosystem for training. Through the capabilities of the OAM layer, the client can continuously monitor the training performance of their model in real time—tracking key performance indicators such as convergence speed, loss metrics, resource usage, and network traversal. The model's location within the ecosystem can be dynamically tracked, allowing clients to know exactly where their model resides or which data centers or devices it is interacting with.

Moreover, the OAM layer enables interactive control. Clients can use it to adjust training parameters on the fly, such as learning rates, data sampling strategies, or the choice of collaborative partners. They can even pause, resume, or terminate the training process at will, giving them full agency over the lifecycle of their models. This flexibility is crucial in adaptive AI systems where responsiveness and real-time decision-making are valued.

In this way, the OAM layer effectively functions as the control dashboard or command-line terminal of the DA-ITN-enabled AI ecosystem. Whether through a graphical user interface (GUI), APIs, or automated orchestration scripts, the OAM provides the necessary tools for fine-grained management, status visualization, and policy enforcement.

Beyond individual model control, the OAM layer also facilitates system-wide coordination and policy administration— ensuring compliance with service-level agreements (SLAs), enforcing data governance policies, and managing access rights across domains. It plays a foundational role in building trustworthy, maintainable, and operationally efficient AI services across diverse infrastructure providers and stakeholders.

4.4. Summary of the DA-ITN General Framework

Accordingly, the DA-ITN is well positioned and designed to provide a range of intelligent services that can be leveraged by both AI clients and service providers. It forms the foundation for a scalable, decentralized AI internet, driving the emergence of a vibrant and cooperative agent-based ecosystem. By enabling the formation of adaptive and intelligence-driven topologies and being agnostic to the infrastructure, the DA-ITN facilitates more effective decisions in AI training, inference, and agent-to-agent interactions— ultimately supporting a more responsive, resilient, and capable AI infrastructure that can scale with future demands.

In the following sections, we provide more detailed insights into the specific DA-ITN components that support training and inference services.

5. DA-ITN for Training

The training architecture of the DA-ITN consists of five layers: i) the terminal layer; ii) the network layer; iii) the data, resource, and reachability topology layer (DRRT); iv) the DA-ITN intelligence layer; and v) the OAM layer. The layers interact together using control and data planes (CP and DP respectively) as is discussed in the following.

First, the network layer, which is at the heart of the DA-ITN training system, is responsible for providing connectivity services to the four other layers. It provides both control and data plane connectivity to enable various services. The network layer connects to the terminal and DRRT layers via CP and DP links, and connects to the intelligence layer via a CP link only. The network layer also enables the overarching OMA layer by enabling a multi-layer connectivity structure.

Second, the terminal layer, the lowest layer in the architecture, contains the terminal components of the system. These include nodes that host the training data, facilities that provide computing resources where the model can be trained, and newly proposed components that we refer to as the model performance verification units (MPVUs), where the model testing phase takes place. It should be noted that facilities providing computing resources come in various forms including private property such as personal devices, in a distributed form such as in the case of mobile edge computing in 6G networks, on the cloud such as on the AWS cloud, or anywhere that is accessible by both the data and the model and holds sufficient compute for training. As for the MPVU, this unit is important when conducting distributed training as it takes the role of a trusted proxy node that holds a globally constructed testing dataset - the dataset is constructed via collecting sample datasets from each participating node - and provides safe and secure access to it. Last, the terminal layer also hosts the AI training clients.

The terminal layer relies on the network layer to build an overarching knowledge-sharing network. To be exact, the network layer provides three main services to the terminal layer, namely: i) moving models and data between the identified rendezvous compute points where training can happen; ii) moving the models towards the MPVU units where performance evaluation can be conducted to keep track of the training progress; and iii) enabling AI training clients to submit their models, monitor the training progress, modify training requirements, and collect the trained models. Control and data traffic exist for each one of these services. For instance, moving a model toward a compute facility requires authorization for the utility of the resources; hence, authorization control data is required to be exchanged over the Terminal-NET CP links. The service also requires the physical transmission of the model to the computing facility which is handled over the Terminal-NET DP link. Similar situations can be extrapolated for the other provided services. It is worth noting that the network layer can be built on top of any access network technology including 3GPP cellular networks, WiFi, wireline, peer-to-peer, satellites, and non-terrestrial networks (NTN), or a combination of the above. These networks can be used to build dedicated CP and DP links strictly designed to enable the DA-ITN training system and its services.

Third, the DRRT layer holds all the information required to make accurate decisions and sits between the intelligence layer and the terminal layer. It consists of a DRRT-manager (DRRT-M) unit which is the brain of this layer and interfaces with the other layers over CP links. The DRRT layer provides the intelligence layer with visibility and accessibility services to specific information about the underlying terminal layer’s data, resource, and reachability status. To be exact, the DRRT layer holds information regarding the type, quality, amount, age, dynamics, and any other essential information about the data available for training. It also provides reachability information of the participating nodes to avoid unnecessary communication overhead and packet droppage. Lastly, the DRRT also contains information about computing resources and MPVUs such as resource availability, location, trustworthiness, and nature of the testing datasets hosted at the different MPVF units.

The DRRT relies on the network layer to collect the necessary information to build the Global-DRRT topology (G-DRRT). The G-DRRT is a none model specific topology, it is rather a large canvas that holds the high-level view of the data, resource, and reachability information. The DRRT-M unit in the DRRT layer communicates with the network layer over CP links to manage the collection process of the required information. For instance, the DRRT-M may instruct the 3GPP component of the network layer to convey connectivity information about the data nodes, or it might instruct it to wake up an ideal

data provider device. It might also instruct satellites to share GPS locations of mobile data nodes. The collected data by the network layer are then shipped toward the G-DRRT component of the DRRT layer over DP links. The G-DRRT hosts intelligence that allows it to convert the collected information into useful global topology ready to provide services to the AI training clients.

Fourth, The Intelligence Layer is responsible for hosting the decision-making logic required to fulfill the specific training requirements submitted by clients. It contains several key components that collaboratively determine how, where, and whether training should proceed. Among these is the Model Training Route Compute Engine (MTRCE), which identifies suitable rendezvous points between models and data. Another critical component is the Training Feasibility Assessment Module (T-FAM), which functions as an admission controller— evaluating whether a submitted model, given its requirements and constraints, can be effectively trained within the available ecosystem.

Additional intelligent modules include the Training Algorithm Generator (TAG) and the Hyperparameter Optimizer (HPO). These components are responsible for selecting the appropriate training paradigm— such as reinforcement learning (RL), federated learning (FL), or supervised learning (SL)— as well as determining other configuration details like the number of training epochs, batch size, and optimization strategy. The Intelligence Layer also interfaces with both the Network Layer and the DRRT Layer to acquire the context needed for effective decision-making. From the Network Layer, it receives control data over CP links— this includes model structure, target accuracy, convergence time, monitoring instructions, and client-specified training preferences. It also receives feedback data that allows the TAG and HPO modules to refine their recommendations dynamically.

Meanwhile, the Intelligence Layer connects to the DRRT Layer via both CP and DP links to access up-to-date visibility into training data, compute resources, and node reachability. This information is essential for components like MTRCE and T-FAM to make routing and admission decisions. To further enhance decision efficiency, the Intelligence Layer may also host a DRRT-Adaptability Unit (DRRT-A). This optional module works in coordination with MTRCE, T-FAM, and the DRRT Manager (DRRT-M) to generate model-specific DRR topologies— lightweight, targeted representations carved out from the global DRR topology. These customized topologies are optimized to reduce computational overhead and accelerate decision-making for individual training requests.

Last, the OAM layer, which spans all the layers, is mainly intended as a management layer to configure the training components, the connectivity of the network layer, and enable feedback functions essential for progress monitoring and model localization and tracking. It is also intended to provide feedback to the clients about their submitted models every step of the way.

6. DA-ITN for Inference

The Inference architecture of the DA-ITN provides automated AI inference services using a similar structure to the training architecture with a few differences.

First, unlike training, where the moving components are models and training data, and the rendezvous points are computing facilities, in inference, models/agents and queries/tasks are the moving components that require networking, and the rendezvous points are model hosting facilities.

Second, in inference, the clients are both the task/query owners as well as the model/agent owners. Query owners are the inference service users who send their queries into the system and collect the resulting inference. On the other hand, model owners are divided into two types. The first type consists of model hosts - the model used for inference does not have to be owned by them, but it is hosted on their computing facilities. The second type consists of model providers - they develop models and deploy them either at their own facilities or at model hosts. Model owners are represented in the terminal layer as model deployment facility providers (MDFP) which are distributed across the global network.

Third, the network layer provides the following services to the terminal layer using its control and data planes: i) model mobility from model generators to model hosts; ii) query routing towards models deployed on MDFPs; iii) model mobility from one location to the other in case of load balancing situations; iv) model mobility towards re-training and calibration facilities which may be hosted on MVPF units; v) query response and inference result routing towards the query owners or any indicated destination around the globe; and vi) feedback and monitoring information to model and query owners.

Fourth, the DRRT layer is replaced by a query, resource, and reachability topology (QRRT) layer. It provides the same type of services to the other layers; however, from the point of view of queries and models. That is, it provides information about both models and queries such as i) for models: model locations, model capabilities, current loading conditions, inference speed, inference accuracy, model reachability and accessibility (i.e., reachability

and accessibility of the MDFP), and ii) for query: query patterns and dynamics (could be associated with a geographical location), query types, and reachability status of query owners for response communication purposes. The information collected by the QRRT is used to make appropriate decisions about model deployment and distribution strategies, query-to-model routing decisions, and response routing decisions. The QRRT has a management function that coordinates with the Network layer to collect the required information from the terminal layer to build the Global-QRRT (G-QRRT). It also optionally communicates with the QRRT-adaptation (QRRT-A) function in the inference intelligence layer to build query- or model-specific QRRTs.

Last, the inference intelligence layer hosts different intelligent decision-making components including the Query Feasibility Assessment Module (Q-FAM), the Query Inference Route Compute Engine (QIRCE), and the Model Deployment Optimizer module (MDO). Just like with the training, these components make decisions based on the QRRT. For instance, the Q-FAM hosts intelligence that acts as an admission control unit that evaluates if a submitted query could be serviced given the current network inference capabilities. The QIRCE handles query routing towards the correct models while observing loading conditions. Furthermore, the MDO module acts as an admission controller for newly submitted models where it evaluates deployment feasibility based on the submitted model’s architecture, compute requirements, and storage requirements. It matches these requirements to the currently available resources indicated in the QRRT and makes an admittance decision. It also handles deployment location optimization, aiming to minimize query response time and cost for inference.

7. DA-ITN-Facilitation Agentic Networks

While agent-to-agent interaction is commonly associated with task-oriented collaboration— often relying on inference chaining as discussed in the inference section— we propose that this only reflects one side of the coin. We believe there is a transformative alternative: collaborative agent training, where agents not only work together to complete tasks, but also contribute to each other’s learning and evolution. This paradigm marks a significant shift from traditional models and positions the DA-ITN as an ideal enabler of a truly agentic future, where intelligent agents can grow, adapt, and improve continuously through structured cooperation.

It is important to distinguish clearly between collaborative training and task-based collaboration. In task-based collaboration, agents exchange data or partial inferences related to the execution of a specific, external objective— such as processing a query or generating

an output. Their internal models remain unchanged; they simply contribute to a shared computational goal. In contrast, collaborative training focuses on internal evolution: the goal is not to solve an external task, but to enhance the capabilities of the participating agents themselves.

In a collaborative training setup, agents may exchange model parameters, training datasets, or knowledge representations. They may engage in distributed training paradigms such as federated learning, where learning happens locally and updates are shared globally, or continual learning, where agents adapt over time based on new experiences. They may also employ knowledge distillation or transfer learning, where more advanced "teacher agents" guide "student agents" through structured training programs. One can even envision a highly dynamic and autonomous system where agents attend “agent schools” —virtual environments where they gather to learn, be tested, and graduate. In this imagined scenario, teacher agents would be responsible for training student agents, evaluating their performance, and possibly issuing certifications or verifiable credentials that guarantee the agent’s competencies and readiness for deployment. These credentials serve trust foundations in the broader agent ecosystem, ensuring that certified agents can be reliably selected and trusted by inference clients or other agents.

To support such a vision, a wide range of new functional and technical requirements must be addressed. These include secure model sharing, certification and validation infrastructure, identity management, trust negotiation, resource discovery for training, and scheduling of learning sessions. Fortunately, many of these requirements align naturally with the capabilities and components of the DA-ITN architecture—including its support for mobility, discovery, descriptor sharing, trust enforcement, dynamic rendezvous, and topology management.

8. Security Considerations

Security considerations are as outlined within the document under the privacy and security requirements

9. IANA Considerations

This document has no IANA actions.

10. Conclusions

As AI continues to evolve and integrate into every facet of modern life, it becomes increasingly clear that the supporting infrastructure must evolve with it. The training and inference processes—central to the success of AI—are no longer simple, isolated tasks; they are complex, distributed, and require intelligent coordination across data, compute, and communication domains.

The DA-ITN architecture offers a forward-looking response to this complexity by providing a cohesive, scalable, and intelligent network ecosystem. With its dedicated control, data, and operations & management planes, DA-ITN not only supports the technical requirements of training and inference but also addresses critical concerns such as mobility, privacy, trust, and agent collaboration.

Ultimately, DA-ITN lays the foundation for a new generation of AI-native networks—capable of enabling persistent learning, dynamic agent interaction, and decentralized intelligence at scale. As we move toward an AI-driven future, such architectures will be essential for building reliable, trustworthy, and efficient AI ecosystems.

Contributors

Arashmid Akhavain
Huawei Canada
Email: arashmid.akhavain@huawei.com

Hesham Moussa
Huawei Canada
Email: hesham.moussa@huawei.com

Tong Wen
Huawei
Email: tongwen@huawei.com

Authors' Addresses

Arashmid Akhavain
Huawei Canada
Email: arashmid.akhavain@huawei.com

Hesham Moussa
Huawei Canada

Email: hesham.moussa@huawei.com