Federated Learning for Healthcare Using NVIDIA Clara

White Paper

SWE-CLARA-001-USCA | September 2021

Document History

SWE-CLARA-001-USCA

Version	Date	Authors	Description of Change
01	September 2021	This white paper is authored by the NVIDIA Federated Learning team. To contact: FederatedLearning@nvidia.com	Initial release.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Artificial Intelligence (AI) has been compared to the industrial revolution. It is infiltrating every aspect of modern life, including education, commerce, finance, manufacturing, social platforms and healthcare. The availability of data, along with new advanced algorithms and fast computing, has paved the way for the creation of new AI models. In healthcare, data is increasing at an exponential rate, with omnipresent sensors, wearables, mobile applications and the digitization of health care records, including medical imaging records such as radiology and pathology data. This is promising for the development of all kinds of AI models, providing the data is easily accessible in a central location to train these models. In certain domains, and certainly in medicine, the centralization of data is difficult if not impossible at times, due to constraints such as privacy, regulation, competition, and budget. Distributed learning is one way to mitigate the challenge of having sufficient data to train AI models, and Federated Learning (FL) may provide a solution for building robust AI models from diverse data across institutions, patient types and countries [1].

Chapter 2. Challenges

Federated Learning is not without its challenges. From the creation of a federation to work together in a study, through standardization of data and compatibility of IT systems, many vital aspects need to be considered when using Federated Learning. Here are a few of the challenges to be expected using this approach to AI model training.

2.1 Creating a Federation

For the above-described federated approach to work, you need to have several client-sites willing to participate in this Federated Learning training. The client-sites need to agree on the objective of the Federated Learning and the choice of administrator for the Federated Learning server. Intellectual property (IP) rights, as well as the usage criteria for the resultant Federated Learning model, need to be determined before commencing the Federated Learning training. Creation of such a federation is challenging, and it requires the cooperation of many teams, including ethics teams, Institutional Review Boards (IRBs), legal teams, contract teams, Information Technology teams and others. The more client-sites there are, and the more geographically distributed they are, with different local practices and regulatory requirements, the more difficult this becomes. Hence, we suggest budgeting ample time to figure all these pieces out, and to set agreements in place before the commencement of Federated Learning training.

2.2 Model Selection

Another challenge to Federated Learning training is the selection of an appropriate model. You might want to start with a pre-trained model from a specific institution, or to train a neural network from scratch. A pre-trained model allows a user to get started faster in the development of a robust model, especially if many of the client-sites have a limited number of cases. This allows them to take advantage of transfer learning and allows their local models to converge despite their small dataset. However, this creates more challenging IP conversations and legal arrangements. It is also important to decide up front on the model to be trained to ensure that client-sites have access to the kind of data needed to train this model.

2.3 Standardization of Data

Once a model is selected for training, and it is established that all sites have access to the kind of training data required, it is important to check the quality of this data. While imaging data might be easier to standardize using standards such as DICOM, you still need to make sure, for instance, that images are in the right orientation and scaled appropriately amongst other considerations. For other data, such as data from the Electronic Medical Record (EMR), you need to make sure that the data collected corresponds to the requisite data fields. You also need to make sure it is reported in the same units and formats. Scripts for preprocessing the data are often needed to ensure the quality of the data. It is useful to share pre-agreed upon summary statistics of the raw and preprocessed data between participating client-sites to detect outliers and preprocessing errors before commencing training. However, data privacy needs to be considered when sharing these statistics.

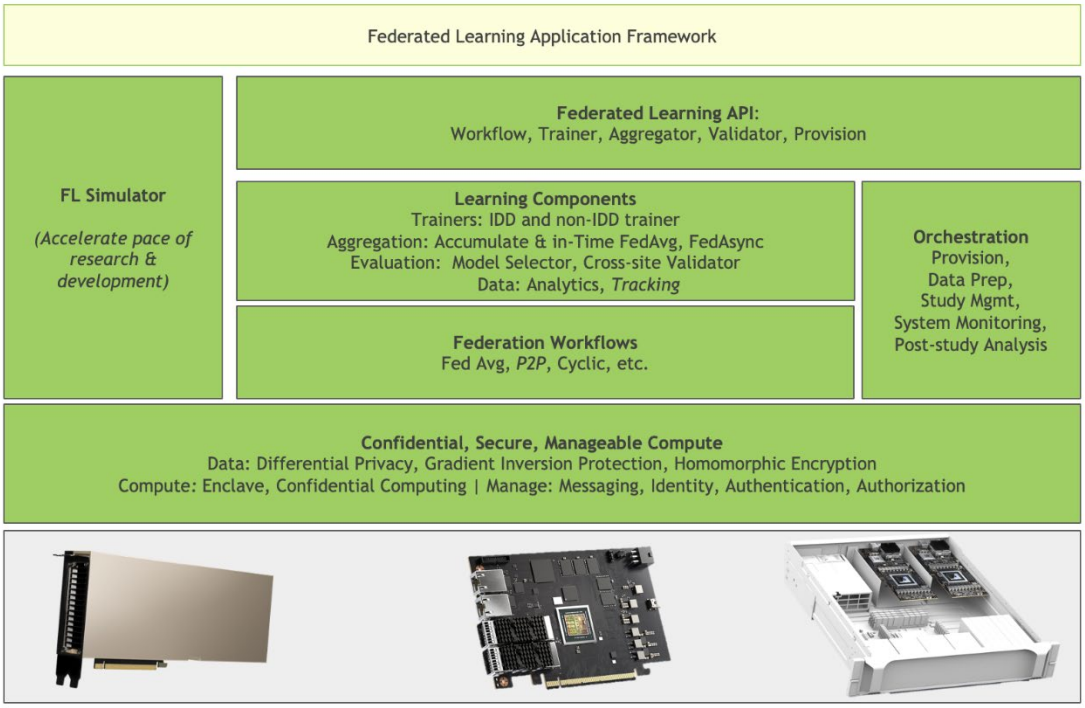
2.4 Compatibility with Information Technology Systems

The successful implementation of Federated Learning is highly dependent on different client-sites being able to communicate effectively with the server. This requires specific protocols and approvals to handle fire walls and communication needs, which are likely to be different at different sites. Security issues are bound to arise, and they are again client-site dependent. These issues need to be resolved in advance to allow for unhindered Federated Learning training.

Chapter 3. NVIDIA Federated Learning Overview

The space of Federated Learning and its applications for healthcare are quickly emerging. NVIDIA is building an enterprise-grade, secure, manageable Federated Learning platform for developers and researchers to build and extend upon.

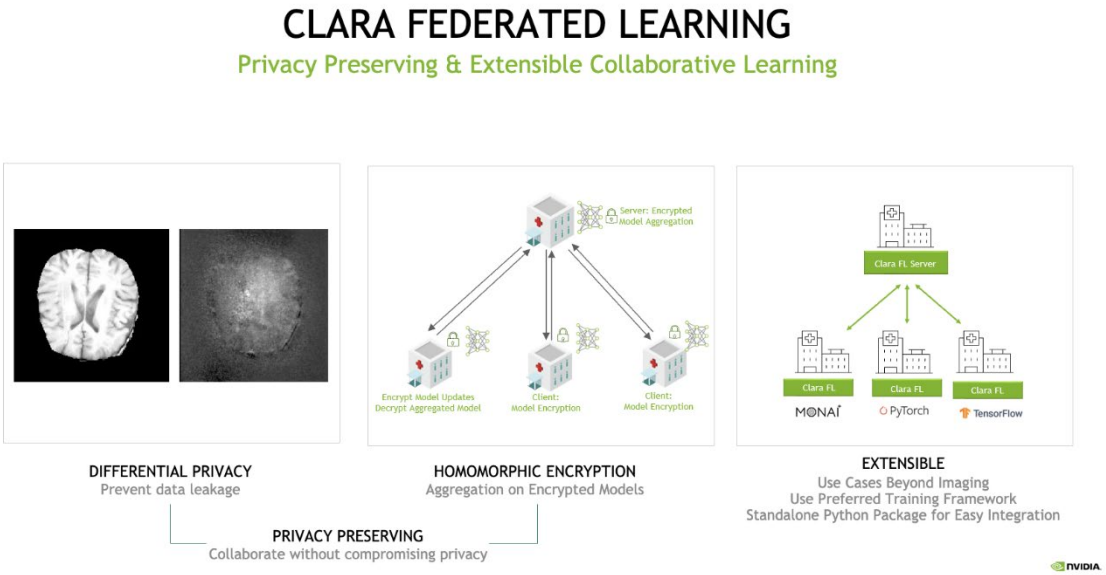
Figure 1. Federated Learning Application Framework



The core engine driving this platform is the NVIDIA Federated Learning Application Runtime Environment (NVFlare), an extensible Federated Learning framework (Figure 1). You can bring existing machine learning tasks into a federated setting easily, allowing researchers to experiment with different Federated Learning strategies. Learn More

Clara Train is the healthcare specific application framework that integrates NVFlare and provides domain specific implementations of Trainer and Validator components.

Figure 2. Clara™ Federated Learning



In this white paper, we focus on a specific way to do distributed training using the FL approach available through Clara™ Federated Learning (Figure 2). This Federated Learning approach utilizes a hub-and-spoke communication model consisting of a Federated Learning server as the hub and client-sites as spokes. The Federated Learning server hosts the initial model and then sends its weights to each of the client-sites. The client-sites train the model on their own data at their location, and at the conclusion of this local training, they send back the updated model weights to the Federated Learning Server. The Federated Learning server waits for a certain number of the models to finish local training, and to receive the updated weights from these client-sites, before it aggregates these weights. The newly aggregated weights are then sent back to each of the clients to be used in a new round of local training. These steps are repeated for a predetermined number of rounds. The resultant Federated Learning model is then automatically validated at each of the client-sites to assess it for generalizability.

Chapter 4. Federated Learning in Practice

4.1 Client-Site Approvals

In addition to the challenges cited above in creating a federation, client-site specific challenges also need to be addressed. The challenges are twofold: (1) obtaining permission to participate in Federated Learning, and (2) permission to do local training. This requires approval from the IRB at each institution that governs data access and usage. The fact that data does not leave the client-site makes such approvals easier to obtain, but they are nevertheless required at most institutions and might take a considerable amount of time to obtain.

4.2 Infrastructure Deployment

A reference implementation of Federated Learning using NVFlare is included in the Clara Train SDK, based on the NVIDIA Pytorch Docker container. This container leverages CUDA 11.2, which requires the NVIDIA R460 driver and a GPU with CUDA compute capability 6.0 or higher for GPU-accelerated deployments. This corresponds to GPUs in the Pascal, Volta, Turing, and NVIDIA Ampere GPU architecture families. The container-based deployment allows flexibility in the underlying hardware and software, which can be easily accommodated by on-premises or cloud-based compute resources, or a mix of thereof.

A Federated Learning study is comprised of a central server and any number of training clients, each of which may require a different set of compute resources based on the role. The central server does not necessarily require a GPU, and its compute requirements may be met with a modest set of resources that depend on the size of the federation. For example, a federated learning proof of concept study with only a few clients may be served with a minimal configuration, whereas a more realistic study on the order of ten clients would require a server with higher specifications (Figure 3).

Table 1. Compute Requirements

System Resources	Proof of concept - few clients	Production study – O(10) clients
CPU	8 core	32 core
System Memory	16GB	64GB
Storage	250GB	1TB

Ultimately, the server configuration is dependent on the specifics of the Federated Learning study and not solely on the number of participants. Considerations include the complexity of the model and size of model weights, which drive both storage and network utilization, the number of experiments in the study, which impact storage, and the use of security measures, such as differential privacy and homomorphic encryption (see Privacy and Security section), which increase compute and memory consumption.

Client system requirements are similarly determined by the specifics of the Federated Learning study, where the size of the local dataset and security measures will drive the number of GPUs and requirements for the CPU, memory, and disk space. The federated clients perform the bulk of the work in a Federated Learning study and should be equipped accordingly, for example with an NVIDIA Tesla GPU such as the A100 designed for high performance AI training workloads. The number of GPUs required depends on the size of the local dataset and the number of epochs per training round. For example, a client with a 2x larger dataset would require 2x the number of GPUs in order to complete training in parallel with other clients.

4.2.1 Deployment Options: On Premises or Cloud-Based

The NVIDIA EGX Enterprise Platform and NVIDIA Certified Systems are optimized for peak performance on accelerated applications such as AI training and offer streamlined deployment of the system and software required for these demanding workloads. This allows for simplified on-premises deployment of systems designed for compatibility with GPU-optimized software from the NVIDIA® NGC™ catalog such as federated learning with Clara Train.

A baseline client can be configured using an NVIDIA Certified server platform from any OEM. An example client configuration would include the following:

- Two 8-core server-class CPUs
- 192 GB DDR4 memory (6x 16GB RDIMMs per CPU)
- Two 1.92TB Enterprise NVMe read-intensive SSD
- NVIDIA Ampere A100 PCIe 250W 40GB GPU(s)

- Mellanox ConnectX-5 Dual Port 10/25GbE NIC
- Mellanox ConnectX-6 DX Dual Port 100GbE NIC
- Dual hot-plug power supply

All major cloud service providers (CSP) offer access to instances with NVIDIA Tesla GPUs and virtual machine images pre-configured for compatibility with NVIDIA NGC and GPU accelerated applications. The flexibility of CSP instance types allows easy prototyping as well as production Federated Learning studies. For a small test or proof of concept, the FL server could use a minimal configuration with 8 vCPU, 32GB memory, 100GB attached storage, and a Gbit network. For a production study with tens of FL clients, a more capable server instance would be required. In this case, an instance with 32 vCPU, 128GB memory, 500GB attached disk, and 10Gbps network bandwidth would be more appropriate.

The variety of single- and multi-GPU instance types available in the cloud also allows flexibility in specifying resources for federated clients, with GPU count and accompanying hardware configuration dependent on the client datasets and training requirements. An instance with a single V100 or A100 GPU may be appropriate for a small test client, whereas multi-GPU instances may be used for production clients with larger datasets.

4.2.2 Network Considerations

Clara Train Federated Learning implements a client-server communication model in which all participants use signed certificates to establish identity and secure SSL communication between clients and the server. All certificates are signed by the server's self-signed SSL Certificate Authority (CA) and embed the unique client or server identity in each certificate. When using the Federated Learning provisioning tool as described below, these certificates are generated and signed automatically for each participant and packaged in encrypted archives that are distributed to the participants.

All communication to the Federated Learning server occurs over unprivileged ports, by default 8002 (Federated Learning training clients) and 8003 (Federated Learning admin clients). In all cases communication is initiated by the client systems, simplifying the firewall configuration required for clients and the server. On the client side, the only requirement is that response traffic from the server is allowed through the firewall; it is not necessary to open any ports on the client. The server requires the Federated Learning (8002) and admin (8003) ports to be opened to inbound traffic. You can use a server firewall such as iptables or CSP security groups to restrict access to these ports to only the known IP addresses of the FL and admin clients.

When provisioning a server on a CSP, care must be taken in configuring the CSP network security. The SSL handshake required to establish secure communication between client and server requires that the TLS requests terminate at the client and server endpoints. The use of

an Application (L7) Load Balancer is not compatible with this handshake as the TLS requests are terminated at the ALB level. It is possible to use a Network (L4) Load Balancer if it is configured to forward all SSL traffic to the server endpoint. Note that load balancing between multiple Federated Learning servers is currently not supported, making such a configuration unnecessary.

To initialize communication in a Federated Learning study, the client submits a request to participate to the server. The server validates the client certificate and, if valid, authorizes the client and responds with a unique token used to identify the client contributions through the Federated Learning study. The certificates generated during provisioning and used to establish this secure communication embed the server's fully qualified domain name or hostname. This makes it necessary that the client resolve the server at this hostname. In the case that the server is not resolvable via DNS, it is necessary to map the server's IP to the hostname used in provisioning with an entry in the client's `/etc/hosts` file.

Once secure communication has been established between clients and server, the remaining network consideration is the bandwidth required to distribute the global model and aggregate client contributions during the rounds of Federated Learning training. This requirement is a function of the number of client participants as well as the complexity of the model and thus size of model weights. Generally, it is expected that clients have on the order of gigabit network bandwidth. The server bandwidth is driven by the number of clients. For a small test with few clients, gigabit server bandwidth may suffice. For a larger study with tens of clients, the server should be provisioned with 10Gbit or greater network bandwidth. When hosting the server on a CSP, choose an instance tier with guaranteed network bandwidth.

4.3 Operation and Execution

To better understand the Operation and Execution cycle of a Federated Learning workflow requires understanding some Federated Learning-specific terminology. We'll cover general high-level Federated Learning terminology and the roles that are typically required when setting up a Federated Learning study. We'll then provide an overview of Federated Learning Security, PSO (Provision, Start, and Operate), and Administration.

4.3.1 Clara Train Federated Learning Terminology

First, we'll walk through some of the more general terminologies that are used for various aspects of a Federated Learning setup. These terms help establish a base level of understanding required for all users involved in the study.

Study

A Federated Learning Study is a project with preset goals and participants who will be involved in the training. Defining a Study is one of the essential pieces of a Federated

Learning workflow, as it helps guide the overall end goal and helps set the collaboration effort between institutions.

E.g., Training the EXAM model

Organization

An Organization is a hospital, consortium, university, or other group involved in the Federated Learning study. Each organization will have its computing resources and data that will be used while participating in the study.

Site

A specific Site or location that will be participating in the study. A Site can vary depending on how the organization is structured but typically indicates the location where the compute and resource data are hosted.

Provisioning Tool

The Provisioning tool is provided by Clara Train Federated Learning and generates the setup configuration for the study. These configurations define study roles, organizations, sites, number of clients, and homomorphic encryption settings. These configuration files are distributed to each of the participants in the study.

Federated Learning Server

A server responsible for the client coordination, based on federation rules and model aggregation settings, established by the provisioning setup. This server will contain the global model that is trained throughout the study.

Federated Learning Client

A server running at a client site, performing model training with its local datasets, and collaborating with the Federated Learning Server for federated study.

Admin Client

An application running on a user's machine that allows the user to perform Federated Learning system operations with a command-line interface or API. This user is the only role with full control over experiments performed during the study and will be the one to start and stop training (see the following sections).

4.3.2 Roles in the Federal Learning Study

Next, we'll discuss the various roles that are involved in a Federated Learning Study. We're defining these roles as independent people in the study, but depending on the organization, a single person may fill more than one role depending on the organization and study.

Lead IT

The Lead IT person is responsible for running the provisioning tool and coordinating with IT personnel from all the study sites to ensure the infrastructure is correctly

provisioned. They are also responsible for setting up and managing the Federated Learning Server.

Site IT

The Site IT person is responsible for the management of the Site of their client organization. They will work with the Lead IT to gather the provisioned configuration files and make sure the site is ready to participate in the study (i.e. the local compute and data resources are available).

Lead Researcher

The Lead Researcher is the person who works with Site Researchers to ensure the success of the study. They are involved in determining specific criteria and requirements involving the data and model that may need to be coordinated before starting the study.

Site Researcher

The Site Researcher is the person who works with the Lead Researcher to make sure the client site data is properly prepared for the study.

4.3.3 Scope of Security in Federated Learning

The security of a Federated Learning study depends on two different infrastructures: the Clara Train Federated Learning Framework and the on-site IT security infrastructure. The total security of a study is the combination of the security measures implemented in this application and the security measures of the site IT infrastructure. Below we'll touch on a few aspects of security, but by no means is this a comprehensive list; ensure you're following all industry-standard security practices.

Since Federated Learning requires a framework for provisioning, coordination, and training, this means the framework itself should implement a specific set of security measures to make sure that the study is secure. The Clara Train Federated Learning system implements security measures in the following areas:

Identity Security

The authentication and authorization of the communicating entities, including role-based authorization and site-specific security configurations.

Communication Security

The confidentiality of data communication messages using secure and encrypted communication methods.

Model Protection

Model weight protection to counter against reverse-engineering training data characteristic from model weights, including homomorphic encryption.

Data Privacy

The Federated Learning system inherently gives data privacy as only the participating sites exchange "knowledge" in the form of model weight updates but not their underlying data that generated this knowledge. Of course, the appropriate model protection needs to be applied in order to protect against the above-mentioned risks.

The site IT security infrastructure must handle all other security concerns. These include, but are not limited to the following:

Physical security

Ensuring that physical access to buildings, systems, and server rooms is limited to only essential personnel.

Firewall policies

Defining and limiting access to network traffic, ports, and IP addresses based on specific applications and content types.

Data protection policies

Ensuring that all regional policies for data storage, retention, cleaning, distributions, and access are followed.

4.3.4 Provision, Start, Operate (PSO)

After defining a study, establishing roles, and ensuring the security of your infrastructure, you're now ready to get started with your Federated Learning study. The following is a high-level guide to deploying your first Federated Learning study following the Provision, Start, and Operate (PSO) steps.

Provision

In this step, the Lead IT person generates the packages for the server, client, and admin identities, which are protected with passwords. They will then coordinate with each Site IT person to ensure the package is transferred securely.

Start

Each Site IT person will set up their Site infrastructure by installing their packages, starting the relevant services, and mapping the data location. They will make sure to use the provisioning file given to them by the Lead IT person for the current Federated Learning study.

Operate

The Lead Researcher or Admin operates the Federated Learning Study. They will deploy the initial MMAR, verify that all the clients are available and ready to start training and start, abort, and shut down training.

Figure 3. High-Level Steps of a Federated Learning Study

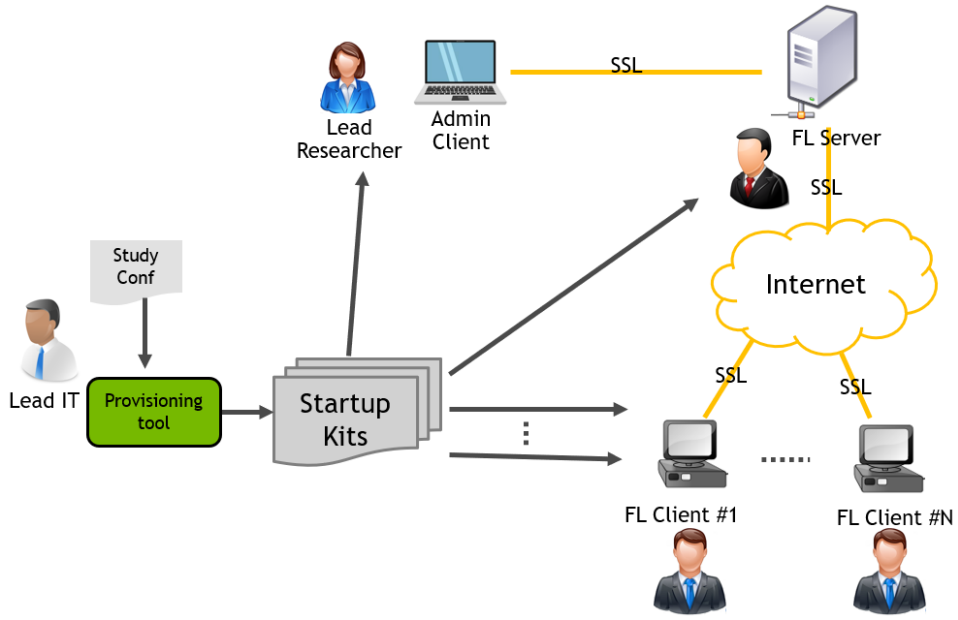


Figure 3 depicts the high-level steps of a Federated Learning Study.

1. Lead IT configures the `config.yaml` file and runs the provisioning tool, which will generate ZIP packages for each client. These packages contain everything a Federated Learning client needs to get started in the Study, including how to start their Docker container, SSL certificates, and other various configurations required to start and complete the Federated Learning experiment.
2. Each Site IT person starts the docker, and the Federated Learning client will use the provided Provisioning Startup Kit.
3. The Lead IT person starts the Federated Learning server using the instructions on how to start the Docker container and use the generated Provisioning Startup Kit.
4. Last, the Admin can either use the Docker container or pip install the admin tool, which will connect to the Federated Learning Server and allows them to start the Federated Learning experimentation. In some cases, there may be multiple Admin's that can control a Federated Learning experiment.

4.3.5 Role-Based Authorization Framework for Admin Clients

Clara Federated Learning implements a role-based authorization framework that determines what a user can or cannot do based on the user's assigned roles and organization rules. To better understand the Federated Learning Authorization policy, we'll define a few terms used when defining the authorization policy for the Federated Learning Study.

Rights

Rights are the permissions for a user to be able to perform specific actions.

For example, a user is given the right to `train_all`, which would allow them to initiate training for all organizations in a group.

Rules

A rule is a policy that gives the Federated Learning study organizer the ability to provide greater or fewer restrictions on the customizations allowed by an organization.

For example, allowing an organization to include custom code or data lists.

Roles

A user can usually be categorized into several types of roles that share the same authorization setting. By creating specific roles, you can easily assign a user to one or more of these roles.

For example, some common users are Lead Researcher, Site Researcher, Site IT, and Lead IT.

Groups

There may also be many organizations in a study, but, similar to user roles, you can easily share a specific set of authorization settings by assigning them to a group.

For example, General Access, Strict Access, or Relaxed Access.

By using a combination of the above authorization options, the Lead IT and Lead Researcher can implement either a flexible and open ruleset or restrictive and narrow ruleset depending on the security requirements for a study and the participating Organizations. The combination of these authorization options could be used, for example, to give the Lead Researcher the ability to operate and monitor all participants in a study, while granting Site Researchers and IT the ability to monitor only the clients within their organization or Group.

4.3.6 Admin Controls of a Federated Learning Study

An Administrator for a Federated Learning Study plays a crucial role as they're the only person who has complete control over the Federated Learning experiments. After the Lead and Site IT have established the server and client setup, the Lead Researcher will run the Federated Learning experiment using the Command-Line interface through the admin client. The following are commands available to the admin:

Check System Operating Status

Allows viewing user permissions, server and client status, environment information, hardware information, and more.

View System Logs

Ability to view system logs to understand what's happening on the clients and server. Useful for debugging any issues.

Verify Role Based Authorization Policy

Ability to verify the role and associated rights of admin clients.

Deploy MMARs

Once an experiment MMAR has been created and staged in the transfer folder, the admin can distribute the folder to the server and clients and set a run number.

Start and Stop Training

After the experiment has been set up, the admin can start and stop training as needed.

Shutdown or Restart the Server or Clients

If there are issues during the experiment, the admin may need to restart a client or server. They can also shut down the client and server once all the experiments are done.

The admin can start the Federated Learning experiment once the Federated Learning server is running and at least one client has joined.

Figure 4. Typical Workflow of a Lead Researcher When Running a Federated Learning Experiment

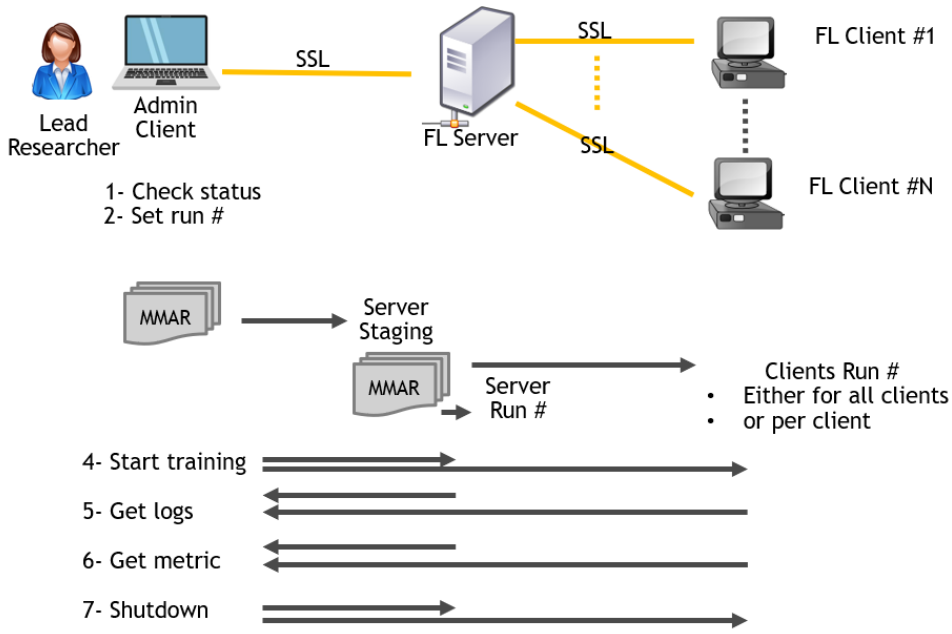


Figure 4 shows the typical workflow of a Lead Researcher when running a Federated Learning experiment:

1. Start the admin tool and log in
2. Check server and client status
3. Set the run number
4. Transfer MMAR to Server Staging
5. Upload MMAR to server and clients
6. Start training
7. Get Logs
8. Get Metrics (when using cross-site validation)
9. Shutdown the server and clients

Once training is complete, you have the option of running cross-site validation. Previously, you needed to move either the data or the selected model to each site and run validation manually. With the cross-site validation feature, it is done automatically for you.

The cross-validation feature is another area where Clara Train Federated Learning shines since the true power of Federated Learning is to develop more robust and generalizable models, which can be analyzed in the off-diagonal values of the cross-site validation result.

Each Site has the opportunity to enable or disable cross-validation through the configuration files. If a Site has cross-validation enabled, their model will generate validation metrics between all other participating Sites. The Lead Researcher can parse the cross-site validation results to show the models' performance on each participating Site as shown in Figure 5.

Figure 5. Example of a Cross-Site Validation Results [16]

Federated:	**Test**						
client	1	2	3	4	5	6	7
1	**0.62**	0.62	0.48	0.15	0.23	0.24	0.11
2	0.22	**0.65**	0.11	0.04	0.00	0.00	-0.01
3	0.41	0.17	**0.63**	0.07	-0.00	0.01	-0.01
4	0.06	0.48	-0.02	**0.69**	0.57	0.65	0.52
5	0.24	0.13	0.02	0.64	**0.62**	0.69	0.52
6	0.23	0.01	-0.00	0.53	0.68	**0.76**	0.31
7	0.10	0.21	0.13	0.55	0.44	0.52	**0.77**
Global	0.51	0.52	0.49	0.31	0.4852	0.31	0.0893
			diag. mean				**0.68**
			off-diag. mean				**0.26**

For more details on how to run your own Federated Learning project using Clara Train and NVFlare, please consult these Jupyter Notebooks and the NVFlare and Clara Train SDK documentation.

4.4 Privacy and Security

In addition to the operational and physical security aspect of Federated Learning, there are additional concerns with Data Privacy and Model Protection.

Model inversion attacks aim to recover or reconstruct the training data from the model parameters. Although DNNs have a much larger amount of parameters and are usually trained with a large amount of data, it has been shown that methods exist to reconstruct portions of the training data (if not all) with relatively high quality and reliability, purely from a trained model [2] or from model gradients [3], [4]. With more info on the model (e.g., the gradient, loss, etc), more training data could be reconstructed with possibly higher quality. This results in a high risk of privacy leakage during model sharing in both regular and FL algorithm development. Differential privacy is one of the widely used algorithms to reduce such risk. It is shown to have minimum impacts on model accuracy during FL training. Meanwhile, the research community has been actively working on both model inversion and defense algorithms.

4.4.1 Data Privacy and Model Protection

To mitigate the risk of recovering the training data from the trained model, which is also commonly known as reverse engineering or model inversion, we provide a configurable client-side privacy control based on the differential-privacy (DP) technique. During training, each client could have their own privacy policy and could be updated by the admin client during training.

The DP protection consists of two major components: selective parameter update and sparse vector technique (SVT):

- For selective parameter update, the client only sends a partial of the model weights/updates, instead of the whole, to limit the amount of information shared. This is achieved by (1) only uploading the fraction of the model weights/updates whose absolute values are greater than a predefined threshold or percentile of the absolute update values and (2) further replacing the model weights by clipping the value to a fixed range.
- The sparse vector technique operates on a random fraction of the weights/updates x by first adding random noise to its absolute value $abs(x)+Lap(s)$; then the clipped noisy values $clip(x+Lap(s), \gamma)$ are shared if the thresholding condition is satisfied. Here $abs(x)$ represents an absolute value, $Lap(x)$ denotes a random variable sampled from the

Laplace distribution, γ is a predefined threshold, and $clip(x, \gamma)$ denotes clipping of x to be in the range of $[-\gamma, \gamma]$.

For details, please refer to [5]. The experimental results show that there is a tradeoff between model performance and privacy-preservation.

4.4.2 Homomorphic Encryption

NVIDIA Clara Train 4.0 adds **homomorphic encryption (HE)** tools for Federated Learning (FL). HE enables you to compute data while the data is still encrypted.

In Clara Train 3.1, all clients used certified SSL channels to communicate their local model updates with the server. The SSL certificates are needed to establish trusted communication channels and are provided through a third party that runs the provisioning tool and securely distributes them to the hospitals. This secures communication to the server, but the server can still see the raw model (unencrypted) updates to do aggregation.

With Clara Train 4.0, the communication channels are still established using SSL certificates and the provisioning tool. However, each client also optionally receives additional keys to homomorphically encrypt their model updates before sending them to the server. The server doesn't own a key and only sees the encrypted model updates.

With HE, the server can aggregate these encrypted weights and then send the updated model back to the client. The clients can decrypt the model weights because they have the keys and can then continue with the next round of training

HE ensures that each client's changes to the global model stays hidden by preventing the server from reverse-engineering the submitted weights and discovering any training data. This added security comes at a computational cost on the server. However, it can play an important role in securing patient data at each hospital while still benefiting from Federated Learning with other institutions.

A benchmarking of HE in Clara Train and notebook of how to use it can be found at https://developer.nvidia.com/blog/federated-learning-with-homomorphic-encryption.

HE can reduce model inversion or data leakage risks if there is a malicious or compromised server. However, your final models might still contain or memorize privacy-relevant information. That's where differential privacy methods can be a useful addition to HE. Clara Train SDK implements the sparse vector technique (SVT) and partial model sharing that can help preserve privacy. For more information, see [5]. Keep in mind that there is a tradeoff between model performance and privacy protection.

Chapter 5. Future Directions

5.1 Addressing Heterogeneity of Data Distribution and Heterogeneity of Client Environment

Important challenges in Federated Learning remain [6], like how to efficiently train models in the non-I.I.D. setting that is bound to arise in real-world Federated Learning studies where acquisition settings and data populations vary among clients. The performance of the Federated Learning models should be comparable or very close to the performance of models training on large, centralized datasets but without breaching privacy of the individual participants in the Federated Learning study. Likely, all machine learning tasks could be adapted to a federated setting so that Federated Learning is not only restricted to the supervised learning scenario. These settings could include un-, self-, and semi-supervised learning, meta-learning, and even few- and zero-shot learning, in single and multi-task scenarios.

At the same time, communication protocols or advanced Federated Learning topologies and workflows should be explored to tackle issues with unstable internet connections or unreliable clients. Asynchronous updates might improve performance and reduce idle times of clients when waiting for the next global model update.

5.2 Federated Datasets: Regulatory Approval, Need for a Standard for Federated Learning

To scale up the application of Federated Learning in practice, standardization efforts will be needed [7]. Potentially, this will allow different parties to collaborate even if they use different software ecosystems to implement their particular Federated Learning servers or clients. One could even envision the use of Federated Learning technology for supporting distributed dataset queries and federated data analytics in general [8].

Software tools to create federated datasets, manage Federated Learning studies, and provide the traceability of individual contributions, will be necessary in the future to streamline the preparation, execution, and regulatory approval of AI model development, in particular in the healthcare sector [1].

5.3 Confidential Computing

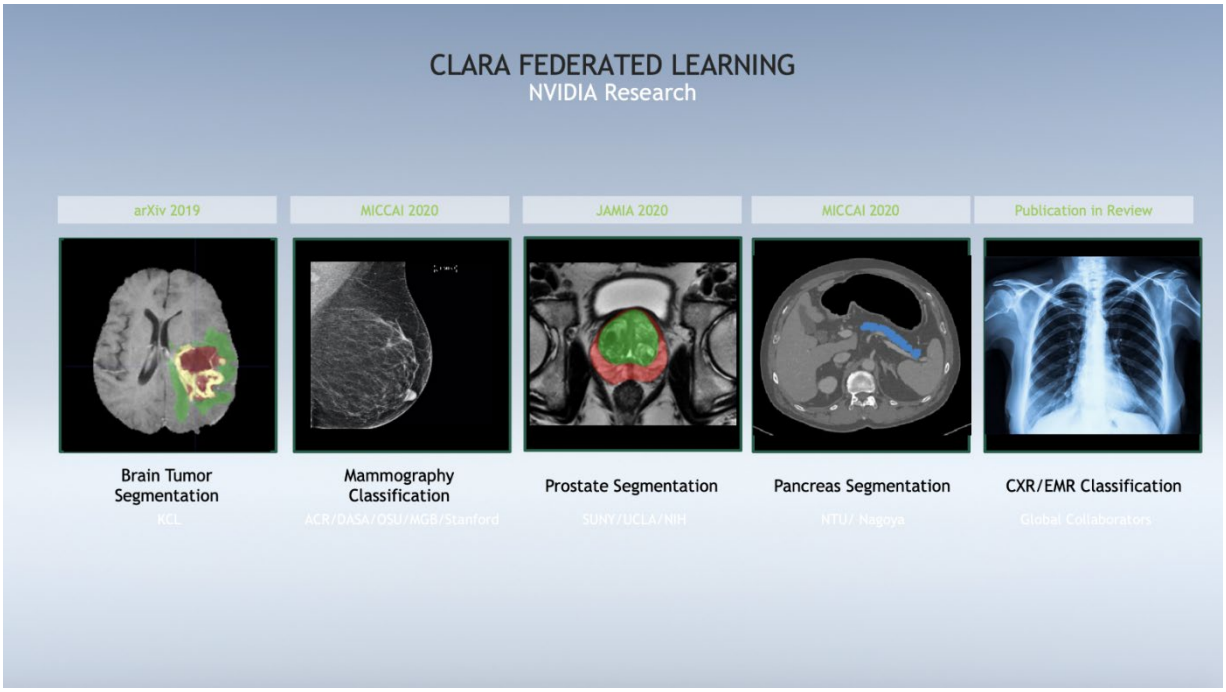
Privacy-preservation protocols like differential privacy [5] still need further study to ensure that they do not impact the model performance while preventing data leakage through gradient and model inversion techniques [3], [4], but also prevent adversarial or backdooring attacks [9].

Homomorphic encryption (HE) as described above can already counteract some of the issues described above by making the server-side aggregation more secure but comes with significant computational overheads. Secure computation schemes such as secure multi-party computation and secret sharing [10], [11] could be alternatives or be combined with HE approaches. Further traceability and security could be achieved with blockchain technology [12] and decentralized Federated Learning ("swarm learning") approaches where no single server might be the only place for attack [13]. The training on the client side itself could also be made more secure by performing all training operations in encrypted space [14] or using a trusted execution environment (TEE) [15]. This would allow protecting one's model IP from untrusted clients.

Despite increased security measures, the resulting models could still memorize training data and hence leak private information after they are trained using Federated Learning. Hence, cryptographic schemes most likely must be combined with differential privacy and other privacy-preserving methods guarantee most confidentiality.

Chapter 6. Case Studies

NVIDIA has worked with several institutions to test and validate the utility of federated learning. To date, we have had five real life implementations in healthcare, pushing the envelope for training robust, generalizable AI models. These initiatives, ADOPS (ACR DASA OSU Partners HealthCare Stanford), FAIRVIEW, SUN Initiative (SUNY UCLA NIH) Prostate Model, Nagoya University & National Taiwan University Pancreas Model, and EXAM (EMR CXR AI Model) Oxygen Requirement Model following COVID are described below.



6.1 ADOPS Breast Mammography AI Model

Early detection through mammography is critical when it comes to reducing breast cancer deaths, but breast density can make it harder to detect the disease. The American College of Radiology (ACR), Diagnosticos da America (DASA), Ohio State University (OSU), Partners HealthCare (PHS), and Stanford University collaborated to improve a breast density classification AI model using NVIDIA Clara Federated Learning.

The team used a 2D mammography classification model provided by PHS, which was trained using NVIDIA Clara Train on NVIDIA GPUs. The model was then retrained using Clara Federated Learning at PHS, as well as the client-sites, without any data being transferred. The result: each institution obtained a better performing model that had overall superior predictive power on their own local dataset. In doing so, Federated Learning enabled improved breast density classification from mammograms, which could lead to better breast cancer risk assessment [16].

6.2 University of Minnesota and Fairview X-Ray COVID AI Model

A Federated Learning study led by University of Minnesota and Fairview MHealth in collaboration with NVIDIA using NVIDIA Clara Train and NVFlare was used to improve real-world AI models for COVID-19 diagnosis based on chest X-rays. This study leverages a three-phase pipeline composed of U-Net lung segmentation, a conditional Generative Adversarial Network (cGAN) for outlier detection, and a DenseNet121 COVID-19 Classification model. The lung segmentation and outlier detection are used in preprocessing the chest X-ray datasets which then feed the COVID-19 classification model. This classification model was trained with a federation of Federated Learning server and Federated Learning clients at University of Minnesota and Fairview (Minnesota, USA), with additional participant clients at Indiana University (Indiana, USA) and Emory University (Georgia, USA) using a mix of cloud (AWS/Azure) and local servers. The aggregate multi-institutional dataset consists of approximately 80,000 labeled images with a 30/70% positive/negative COVID classification. Initial results show an improvement in performance of the global model of 5% AUROC and 8% AUPRC on the UMN local dataset as compared to the UMN local model.

6.3 SUN Initiative Prostate Cancer AI Model

Prostate cancer is a common cancer of the prostate gland in men. It has a high prevalence rate, and is the second-leading cause of cancer deaths for men in the U.S.

Accurate segmentation of the prostate gland is useful for developing AI models to help in detection of Prostate cancer. In this initiative, we tested the hypothesis that Federated Learning can be used to train a segmentation model comparable to one trained from a pooled data (PD) set. We proceeded to train two models for prostate segmentation using the ProstateX Challenge Dataset. One model was trained on a pooled data set, and the other one in a Federated Learning manner using Clara™ Federated Learning with the dataset divided amongst three client sites.

Our results showed equivalent performance from both the experimental Federated Learning and benchmark PD models, showing the feasibility of training an AI model in a Federated Learning approach [17].

6.4 CT Pancreas Segmentation AI Model

NVIDIA worked with National Taiwan University, Taiwan, and Nagoya University, Japan, to utilize federated learning to build models for the automated segmentation of the pancreas and pancreatic tumors in abdominal CT [18]. A 3D segmentation model based on neural architecture search developed by NVIDIA's Applied Research team [19] was collaboratively trained using Clara™ Federated Learning. The global Federated Learning model achieved a segmentation performance of 82.3% Dice score on healthy pancreatic patients on average. Read more in the paper:

https://arxiv.org/abs/2009.13148

6.5 EXAM AI Model for Predicting Oxygen Requirements in COVID Patients

As evidenced by the COVID-19 pandemic, efficient allocation of scarce medical resources is crucial.

These resources include staffing, hospitals beds and ventilators. Triaging patients to the right level of care can make the most use of these resources allowing, a hospital to treat a larger number of patients efficiently and effectively. As patients present to the Emergency Department (ED), it is helpful to know which patients will need a higher level of care in the near future, despite perhaps presenting with minimal symptoms.

Researchers at NVIDIA and Massachusetts General Brigham Hospital have used NVIDIA Clara™ Federated Learning to train a previously developed AI model that determines whether a person showing up in the emergency room with COVID-19 symptoms will need supplemental oxygen hours or even days after an initial exam.

Rather than needing to pool patient chest X-rays and other confidential information into a single location, each client-site used a secure, in-house server for its data. A separate server, hosted on AWS, held the global deep neural network, and each client-site got a copy of the model to train on its own dataset.

Training was completed in two weeks, resulting in a global model with .94 Area Under the Curve (AUC), resulting in excellent prediction for the level of oxygen required by incoming patients. The federated learning model is accessible as part of NVIDIA Clara ™ on NGC. You can read more about

this initiative in this blog: Triaging COVID-19 Patients: 20 Hospitals in 20 Days Build AI Model that Predicts Oxygen Needs.

A preprint of the journal paper describing the study and outcomes can be found at [20].

Chapter 7. References

[1] N. Rieke *et al.*, "The future of digital health with federated learning," *NPJ digital medicine*, vol. 3, no. 1, pp. 1–7, 2020.

[2] B. Wu *et al.*, "P3sgd: Patient privacy preserving sgd for regularizing deep cnns in pathological image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2099–2108.

[3] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting Gradients–How easy is it to break privacy in federated learning?," *NeurIPS 2020*, 2020.

[4] H. Yin, A. Mallya, A. Vahdat, J. M. Alvarez, J. Kautz, and P. Molchanov, "See through Gradients: Image Batch Recovery via GradInversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 16337–16346.

[5] W. Li *et al.*, "Privacy-preserving federated brain tumour segmentation," in *International workshop on machine learning in medical imaging*, 2019, pp. 133–141.

[6] P. Kairouz *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[7] F. Qiang, T. Lixin, L. Richard, and others, "White Paper-IEEE Federated Machine Learning," 2021.

[8] D. Ramage and S. Mazzocchi, "Federated analytics: Collaborative data science without data collection, May 2020," *URL https://ai. googleblog. com/2020/05/federated-analytics-collaborative-data. html. Google AI Blog*.

[9] H. Wang *et al.*, "Attack of the tails: Yes, you really can backdoor federated learning," *arXiv preprint arXiv:2007.05084*, 2020.

[10] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, no. 6, pp. 305–311, 2020.

[11] G. Kaissis *et al.*, "End-to-end privacy preserving deep learning on multi-institutional medical imaging," *Nature Machine Intelligence*, vol. 3, no. 6, pp. 473–484, 2021.

[12] M. N. Galtier and C. Marini, "Substra: a framework for privacy-preserving, traceable and collaborative machine learning," *arXiv preprint arXiv:1910.11567*, 2019.

[13] S. Warnat-Herresthal *et al.*, "Swarm Learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

[14] B. Knott, S. Venkataraman, A. Hannun, S. Sengupta, M. Ibrahim, and L. van der Maaten, "CrypTen: Secure multi-party computation meets machine learning," 2020.

[15] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: what it is, and what it is not," in *2015 IEEE Trustcom/BigDataSE/ISPA*, 2015, vol. 1, pp. 57–64

[16] H. R. Roth *et al.*, "Federated learning for breast density classification: A real-world implementation," in *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*, Springer, 2020, pp. 181–191.

[17] K. v Sarma *et al.*, "Federated learning improves site performance in multicenter deep learning without data sharing," *Journal of the American Medical Informatics Association*, vol. 28, no. 6, pp. 1259–1264, 2021.

[18] P. Wang *et al.*, "Automated pancreas segmentation using multi-institutional collaborative deep learning," in *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*, Springer, 2020, pp. 192–200.

[19] Q. Yu *et al.*, "C2fnas: Coarse-to-fine neural architecture search for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4126–4135.

[20] M. Flores *et al.*, "Federated Learning used for predicting outcomes in SARS-COV-2 patients," *Research Square*.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation ("NVIDIA") makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer ("Terms of Sale"). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer's sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer's product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Trademarks

NVIDIA, the NVIDIA logo, NVIDIA Clara, Clara Train SDK, Clara Federated Learning, and NVFlare are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com