Fed-BioMed: Open, Transparent and Trusted Federated Learning for Real-world Healthcare Applications

Francesco Cremonesi, Marc Vesin, Sergen Cansiz, Yannick Bouillard, Irene Balelli, Lucia Innocenti, Santiago Silva, Samy-Safwan Ayed, Riccardo Taiello, Laetita Kameni, et al.

Fed-BioMed: Open, Transparent and Trusted Federated Learning for Real-world Healthcare Applications

Francesco Cremonesi[1], Marc Vesin[2], Sergen Cansiz[2], Yannick Bouillard[2], Irene Balelli[1], Lucia Innocenti[1], Santiago Silva[1], Samy-Safwan Ayed[1], Riccardo Taiello[1], Laetita Kameni[7], Richard Vidal[7], Fanny Orlhac[4], Christophe Nioche[4], Nathan Lapel[5], Bastien Houis[5], Romain Modzelewski[5], Olivier Humbert[3], Melek Önen[6], and Marco Lorenzi[1]

[1]Université Côte d'Azur, Inria Sophia Antipolis, Epione Research Group, France
[2]Université Côte d'Azur, Inria Sophia Antipolis, SED Group, France
[3]UCA, Université Côte d'Azur, France
[4]Laboratoire d'Imagerie Translationnelle en Oncologie (LITO), U1288 Inserm, Université Paris-Saclay, Centre de Recherche de l'Institut Curie, Orsay, France
[5]Nuclear Medicine Department, Henri Becquerel Center, Rouen, France
[6]EURECOM, France
[7]Accenture Labs, France

Abstract

The real-world implementation of federated learning is complex and requires research and development actions at the crossroad between different domains ranging from data science, to software programming, networking, and security. While today several FL libraries are proposed to data scientists and users, most of these frameworks are not designed to find seamless application in medical use-cases, due to the specific challenges and requirements of working with medical data and hospital infrastructures. Moreover, governance, design principles, and security assumptions of these frameworks are generally not clearly illustrated, thus preventing the adoption in sensitive applications. Motivated by the current technological landscape of FL in healthcare, in this document we present

Fed-BioMed: a research and development initiative aiming at translating federated learning (FL) into real-world medical research applications. We describe our design space, targeted users, domain constraints, and how these factors affect our current and future software architecture.

1 Introduction

The need for large amounts of data to develop Artificial Intelligence (AI) in healthcare has motivated a number of national and international initiatives aimed at creating medical data lakes accessible to researchers, such as the French Health Data Hub [10], the UK BioBank [59], the US ADNI [26] and TCGA [60], among the many [58, 40, 7]. In spite of these initiatives, there are still major bottlenecks preventing the widespread availability of large centralized repositories of healthcare information [63].

To overcome these limitations, Federated Learning (FL) has been proposed as a working paradigm to enable the training of ML models on large datasets from diverse sources while guaranteeing the respect of data privacy and governance. The basic paradigm of FL consists of iterating the following steps: i) model training is performed locally in the hospitals starting from a common initialization, ii) the resulting model parameters are subsequently shared (instead of the data) and aggregated, to define a global model iii) transmitted back to the hospitals to initiate a new local training step. Under certain conditions [39], this procedure is guaranteed to converge to a final global model representing an optimal consensus among the hospitals participating in the experiment. FL is particularly suited for applications in sensitive domains, such as healthcare and biomedical research [48, 9, 13]. The current societal and economical interest in FL for healthcare is paramount [56, 52], as demonstrated by the several large-scale medical research projects based on FL at the national and international level, focusing for example on rare hematological diseases[1], drug development[2], blood cancer[3], among the many [15, 53, 19].

In spite of the current popularity, the real-world implementation of FL is complex and requires research and development actions at the crossroad between different domains spanning data science, software programming, networking, and security. Today, several FL software frameworks are currently being proposed to data scientists and users, based on different design spaces, goals, and with varying degrees of software maturity. Nevertheless, most of these frameworks are not designed to find seamless application in medical use-cases, due to the specific challenges and requirements of working with medical data and hospital infrastructures. Furthermore, several widely-used FL libraries oftentimes devote little attention to describe the design space and guiding principles, while the spotlight is often placed on describing implementation details without justifying particular design choices.

[1] `https://genomed4all.eu/`

[2] `https://www.melloddy.eu/`

[3] `https://www.harmony-alliance.eu/`

In this document we present Fed-BioMed: a research and development initiative aiming at translating FL into real-world medical research applications, motivated by the current technological landscape of FL for healthcare. We describe our design space, targeted users, domain constraints, and how these factors affect our current and future software architecture. While implementation details may change with the evolution of new technologies and contributions from a growing pool of developers, we believe that a description of Fed-BioMed's philosophy and guiding principles, along with the relevant architectural details, provides a faithful and transparent representation of our goal and ambitions.

1.1 Contribution

While several FL frameworks are currently available, none of them have documented a clear set of design principles and guiding concepts inspired by the application domain, nor a demonstration of how their architecture and implementation satisfy the specific requirements of FL in clinical applications. In what follows, we identify the requirements arising from the needs of medical data owners and biomedical data scientists, and show how Fed-BioMed addresses the specific challenges of this domain.

To meet the strict security requirements typical of medical environments, Fed-BioMed focuses on empowering the user with a tight control of data management and model training process. This is based on a deployment workflow enabling easy setup of its main components, on prototypes of a data and model governance system with a graphical interface for non-technical users such as clinical data managers and physicians, on a Jupyter notebook interface for researchers and data scientists, and on extensive documentation and tutorials targeting biomedical data scientists as well as health data providers[4]. Overall, the straightforward design of Fed-BioMed aims at simplifying the development and deployment of federated learning analysis in real-world healthcare research.

2 Federated Learning for biomedical research applications: design considerations

The application of ML methods, and in particular of FL, in the context of medical data analysis presents unique challenges requiring a targeted approach. First and foremost, is the conflict between the need for large datasets to train ML models and data sharing regulations, such as the European General Data Protection Regulation (GDPR)[5] and the US Health Insurance Portability and Accountability Act (HIPAA)[6]. Additionally, ethical, economical, and technical barriers hinder the sharing of medical information [63, 61]. The entities that provide medical records for FL experiments are therefore compelled to enforce

[4] `https://fedbiomed.gitlabpages.inria.fr/`
[5] `https://gdpr-info.eu/`
[6] `https://www.cdc.gov/phlp/publications/topic/hipaa.html`

strict data governance policies on their data, requiring strong security and privacy guarantees as well as retaining the ability to exert fine-grained control over data processing and flow.

Medical data are generally not directly amenable to FL analysis: they are often stored in unstructured, potentially proprietary formats inside independent silos, leading to large degrees of heterogeneity making data potentially biased and difficult to compare [24, 61]. For example, the lack of standardized coded definitions may lead to noisy labels, with a detrimental effect on training performance, especially for medical data where missingness is a common problem [21]. Big data in medicine is characterized by relatively low volumes with high information density [24], and often requires integration of multiple data acquisition methods, bringing all the challenges associated with the analysis of multimodal data [30]. Furthermore, hospital IT infrastructure has been designed to support clinical and billing operations, but is not optimized for data analytics [24]. The procedures for installing and operating FL software may vary wildly across different hospitals, leading to difficulties during the deployment process and inconsistencies in the execution.

In a biomedical research setting, the prototypical FL experiment consists of a dynamic and highly interactive series of training rounds interspersed with sessions of model and hyperparameter tuning, interpretation of partial results with domain experts, and general debugging. This process requires a large degree of interactivity that can be in contrast with the design of FL systems for other applications that are more focused on batch execution of training rounds, model stability, and high availability of the infrastructure.

Any software being operated in the context of biomedical research must satisfy strict security rules arising not only from data privacy concerns but also from intellectual property and compliance to guidelines [12]. The FL process itself exposes multiples vulnerabilities such as model inversion, membership inference, and model poisoning attacks [6, 57], which must be mitigated through a combination of classical cybersecurity approaches — such as e.g. encryption, firewalls, malware protection, and network segmentation— as well as FL-specific techniques such as secure aggregation [36].

2.1 Primary Requirements

From the challenges described above we derive a set of requirements that FedBioMed aims at satisfying, categorized in four primary (i.e. must-have), four secondary (i.e. nice-to-have) and three minor (i.e. could-have) requirements. We summarize the challenges and requirements in Table 1.

Data and model governance

We define these as the granting the following rights to clinical data nodes: i) the ability to review, add or revoke at any time the availability of any given dataset for federated training; ii) the ability to approve, audit and monitor the execution of specific FL workflows; iii) the ability to review, audit and customize

the deployment of the software infrastructure. Furthermore, as the time of data managers and clinical experts is often a scarce resource, the ability to exercise such fine-grained control on data governance and processing should be provided in a simple user interface requiring only minimal learning efforts.

Integration with biomedical data sources

An FL framework targeting interoperability should at the very least be aware of existing interoperability efforts in medical data management and analysis, and ideally should offer direct and seamless integration for the management of data in such formats.

Researcher interactivity

FL frameworks suited for research purposes should provide the ability to launch, stop and generally manipulate the training process, modify model and training parameters on the fly, resume training from checkpoints, and monitor convergence, while respecting the FL paradigm, the data providers' privacy, and node's governance requirements.

Security

An FL framework for the biomedical research domain should provide a secure environment for the FL technical infrastructure, minimizing the surface of attacks on the data providers' systems through the framework itself, implementing network segmentation, securing network communication, and insulating the execution from external attackers. Furthermore, the framework should support and allow for easy activation of gradient protection, against model-targeted attacks such as model poisoning, membership inference, and model inversion.

2.2 Secondary and Minor Requirements

Federated pre/post-processing, i.e. the framework's ability to support data or model pre/post-processing in a federated approach.

Supported ML libraries/frameworks, meaning the framework's capability to seamlessly integrate with a variety of state-of-the-art ML libraries and providing state-of-the-art FL algorithms.

Portability, intended as reproducibility of the development environment as obtained e.g. by the use of containers and virtual environments.

Drop-out tolerance, as in the framework's resilience to node drop-outs, other unexpected failure events, or the framework's ability to provide fine-grained node selection during training. Note that in a controlled cross-silo setting unexpected drop-outs are usually considered less relevant than in a cross-device FL deployment.

Finally, we note that some other requirements are often cited in the broader context of FL applications but are of relatively minor importance to the domain

Challenges	**Primary requirements**	**Secondary and minor**
• Data privacy and protection • Siloed data • Heterogeneous, unstructured data • Integration of multiple data modalities • Dynamic model development cycle • Cyberattacks • Model and gradient attacks • Suboptimal IT infrastructure and computing capabilities • IP protection	**Data and model governance** Powerful and easy-to-use governance tools for nodes. **Integration with biomedical data sources** Support biomedical and interoperability standards; support diverse computing infrastructures. **Researcher interactivity** Support a highly interactive, dynamic model development cycle steered by the researcher. **Security** Secure software and related IT infrastructure against hacking; support model and gradient protection strategies.	federated pre/post processing support for several ML libraries portability drop-out tolerance high availability scalability lightweight

Table 1: Summary of challenges and requirements addressed by fed-BioMed in the application of FL to biomedical research.

of biomedical research, such as e.g. high availability of servers, scalability in terms of number of clients, and lightweight software implementation.

3 Background: FL Frameworks Landscape

The number of FL frameworks has dramatically surged in recent years, witnessing the growing interest in the applications of this technology [31, 56, 44]. We are aware of well-established products whose focus is not related to biomedical applications, but whose implementation does not exclude future deployments in this domain, such as e.g. Tensorflow Federated (TFF) [5], FedML [23], IBM-FL [34], FATE [33], PaddleFL [35], and PySyft [51]. Henceforth, we restrict our analysis to SubstraFL [46], OpenFL [17], Flare [50], and Flower [4], which are frameworks that have already been applied in medical use-cases, in accordance with the focus of this paper. Table 2 summarizes our frameworks review in light of the requirements identified in section 2.1.

LabeliaLab's SubstraFL framework is a Python library based on the Substra software developed by the company Owkin [46, 20]. It is currently being used in healthcare applications for drug discovery in the context of the Mellody project [8], as well as oncology, anatomopathology and fertility in the context of the Healthchain project [45, 43]. SubstraFL's architecture is based on a fully-decentralized distributed ledger, and is designed upon the three core principles of collaboration, privacy, and traceability. Data governance is implemented through operations on the distributed ledger, which also guarantees traceability of all ML operations within the consortium. Operational roles are well-defined, and a permissions regime inspired by the Role-Based Access Control paradigm (RBAC) can be used to enable additional governance measures for handling remote *assets* (algorithms and data). Furthermore, a Graphical User Interface (GUI) is provided to simplify the management of assets on nodes. While SubstraFL has been deployed in healthcare settings, to our knowledge no tools specifically dedicated to the management of healthcare or biomedical assets are provided with the library, nor are there any tutorials or deployment examples focused on this domain. SubstraFL's target use-case is the execution of FL experiments at scale, and therefore the library is intended to be mainly used in production environments. For this reason, it may not be an ideal choice for exploratory workflows requiring a high degree of interactivity during model development and training. The distributed ledger at the core of SubstraFL's architecture also acts as a security feature by providing a strong guarantee against the alteration of training and inference metadata, and potentially against model poisoning attacks. No explicit description of encrypted secured communications, such as TLS, is currently provided in the framework, while differential privacy and secure aggregation approaches are not integrated in the library. Finally, governance and roadmap of SubstraFL are centralised, preventing the development of an open-source community around the project.

OpenFL is another Python-based FL library, originally designed for a healthcare application but later expanded to be use case-agnostic [17]. It has been

	SubstraFL	**OpenFL**	**Flare**	**Flower**
Design focus	collaboration, privacy, traceability	cybersecurity	scalability, flexibility, lightweight	scalability, flexibility, agnostic to clients, communication and privacy
Data governance	Distributed ledger, RBAC, GUI	HTTP API, Python SDK	configuration files, Python SDK	Python SDK
Model governance	Distributed ledger, RBAC, GUI	None built-in	None built-in	None built-in
Integration with medical domain	Only demonstrators	Only demonstrators	Only demonstrators	None built-in
Researcher interactivity	Explicit focus on production	Limited	Limited	High interactivity through Python SDK
Cybersecurity	encrypted communication	TEE, encrypted communication, PKI	encrypted communication	encrypted communication
Model and update attacks	Future development	None built-in	Only demonstrators	Secure aggregation

Table 2: Comparison of FL frameworks with respect to the requirements identified in Section 2.1. The label "Only demonstrators" signifies that a feature is not built in the framework, but that a demonstrator has been provided in the form of a reference implementation, a tutorial, or documentation.

used in the context of medical applications for a global FL deployment aimed at detecting glioblastoma sub-compartment boundaries [47]. OpenFL has been designed to support multi-institutional collaborations with a strong focus on cybersecurity. The system architecture is based on the star topology paradigm with the aggregator as the central node and collaborators as edge nodes, authenticated through PKI certificates and communicating via encrypted TLS connections. Governance is implemented mainly through API operations or code snippets written by data owners. For example, the code implementing a *ShardDescriptor* needs to be present on each node. In some setups, data scientists may query centralized datasets and experiment registries holding metadata describing the whole federation. We could not find any core functionalities in the library that are specific to the integration with biomedical data sources. However, tutorials with a medical focus are provided in the examples section and can serve as a rough template for simple biomedical applications. From the point of view of interactivity, OpenFL's documentation and experiment API seems to implicitly emphasize batch execution of training experiments rather than model exploration, even though some features, such as e.g. the simple porting of model descriptions from the serial to the federated approach, do enable some degree of interactivity. OpenFL has a strong declared focus on cybersecurity. The usage of hardware-level features such as Trusted Execution Environments (TEE), as well as more conventional network-level measures such as TLS-encrypted communication and PKI certificates guarantee high degree of protection against attacks. Nevertheless, fully exploiting the capabilities of OpenFL is bound to the adoption of proprietary hardware. This condition is not necessarily compatible with practical deployment of FL in hospitals, and may critically prevent the adoption of the software in typical real-world scenarios.

Flare is yet another Python FL library, developed by Nvidia [50], which has been recently used in a world-wide federation of clinical sites to develop a new model for triaging patients affected by COVID-19 [14], as well as other healthcare-related demonstrations for e.g. classification and segmentation tasks on medical images [49, 54]. Flare has been designed on the principles of scalability, flexibility and lightweight, and is targeted towards cross-silo FL, not limited to healthcare, supporting both production settings as well as simulated FL for researchers. Flare's architecture is built on the paradigm of one *Controller* distributing *tasks* to several *workers*, thus leading to a more generic framework with respect to the other libraries analysed here. Governance is handled through configuration files and a Python code managed manually by node administrators. Specifications for a specific training experiment are provided partially by the data scientist (e.g. the model) and partially by the clients (e.g. the learners). To our knowledge, no special tools are provided to node administrators for the management of data nor for fine-grained control over algorithm execution. Moreover, while Flare's utility has been showcased in healthcare settings, it still aims to be a generic framework, and as such we could not identify features of the library aimed specifically at the integration with biomedical data sources. Similarly as with OpenFL, the capabilities of Flare are oriented towards the adoption of specialized hardware, as demonstrated by the focus of the project

on GPU usage, scalability, and high availability, which in our experience does not correspond to a prototypical hospital deployment scenario. Flare's reliance on configuration files leads to a somewhat complex ecosystem that could potentially lead to a more rigid structure, lacking some of the flexibility that medical researchers may wish for during an exploratory model development phase. From a cybersecurity point of view, Flare offers protection against man in the middle and impersonation attacks, through the use of encrypted communication, SSL authentication, and a robust provisioning workflow. Furthermore, the concept of local *filters* enables the implementation of secure aggregation, homomorphic encryption, and differential privacy, which are however not available as part of the standard features of the framework.

Recently, the widely-used Flower framework [4] has announced a collaboration with the Swedish decentralized AI project in order to improve the national healthcare ecosystem[7]. The design principles at the basis of Flower's architecture are scalability, flexibility, and generality w.r.t ML, communication and privacy frameworks. Given the general focus, Flower does not come pre-equipped with any data governance tools, nor does it provide any specific tools for the integration with biomedical data formats. On the other hand, Flower's flexible structure makes it highly amenable to highly interactive workflows, which are also further customizable thanks to the extensible `Strategy` class. Flower offers native support for encrypted communication through its use of gRPC [18]. Their Secure Aggregation implementation, named `Salvia`, remains to our knowledge a proof of concept that is not yet built into the library.

4 Methods

4.1 Design space and goals of Fed-BioMed

The main purpose of Fed-BioMed is to enable seamless collaboration between medical investigators, data providers, and data scientists in a high trust, highly interactive research environment. First and foremost, we aim at providing secure tools for the governance of personal biomedical data and the federated training of models on such data. Additionally, we strive to make all our interfaces, especially those facing the medical researchers and data providers, easy to use even for non-technical users. Our second goal is to make the interface for the data scientist as interactive and flexible as possible to allow fast turnaround during the development of new ML models and strategies, while respecting the privacy needs of the data providers.

In the context of the life cycle of a FL experiment, we make a clear-cut distinction between research applications and model deployment in production. Both begin with a data generation and preparation step, which is typically within the scope of a specific clinical investigation. In the case of secondary use of data, this step includes extracting data from an already existing hospital database, cleaning and wrangling the data, and applying the necessary

[7] `https://flower.dev/conf/flower-summit-2022/`

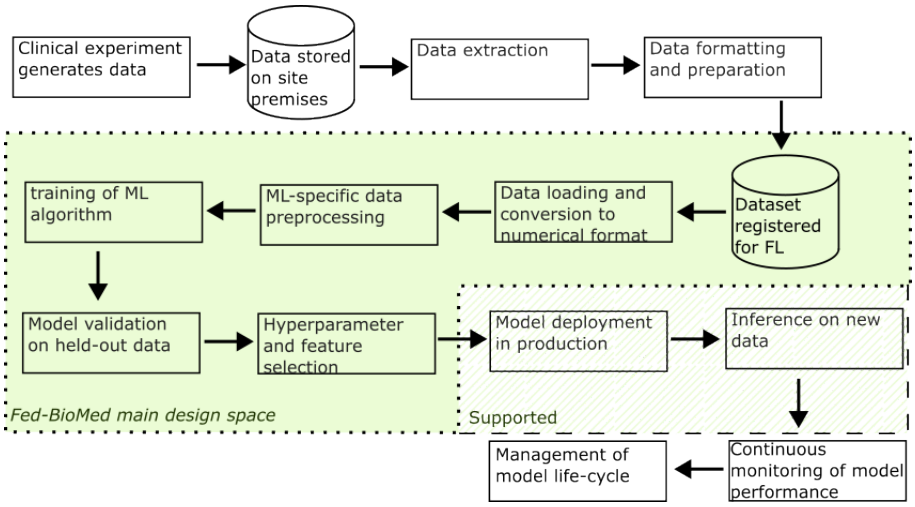

Figure 1: Workflow of an FL experiment from the point of view of a clinical data provider. The shaded area represents Fed-BioMed's design space in terms of functionalities and targeted usage.

anonymization or pseudonymization procedures. This step is followed by a model development phase, comprising federated data preprocessing, training, validation, and hyperparameter tuning. Sometimes the model development phase is split in two sub-phases: an initial exploration where a subset of the data is centralized, followed by the actual federated training phase. In the development phase, emphasis is put on the scientific process of hypothesis formulation, experiment and model design, architecture and hyperparameter search, and training. Finally, once a stable version has been reached, the model can be deployed in production mode, where emphasis is placed more on inference, reliability, robustness and high availability. The main scope of Fed-BioMed is to support the deployment and translation of AI to biomedical research and healthcare during the model development phase, but we also support deployments in production mode. To this end, we are actively working and collaborating with experts in medical data analysis, optimization, security, databases, and visualization. Figure 1 shows the typical workflow of a FL experiment, with the main design space of Fed-BioMed highlighted in the shaded area and the supported deployment modes highlighted in the dashed area.

Our current design goals include using Fed-BioMed in a high-trust environment, where all parties are honest and have an open, secure channel of communication outside of it. We target mainly research consortia composed of a relatively small number of data providers (e.g. university hospitals or medical research centers) and data consumers (e.g. data scientists and researchers with an expertise on biomedical data). In the future, we aim to extend our paradigm to account for less-trusted environments, such as e.g. in the presence

of malicious clients.

Federated learning requires the availability of computing infrastructure at the edge nodes: rather than focusing on high performance computing, we target the more common scenario of clinical sites with varying computing capabilities. We do not make the assumption that the edge nodes' computing infrastructure is able to sustain a high availability cycle; instead we consider that the research work will be organized in periods of active testing and model training, alternating with periods of independent work where the computing infrastructure at the edge nodes may be disconnected.

During periods of research activity and model training, we assume that all actors involved will be available and able to communicate with low response latency. Our researcher and medical data provider interfaces are designed with this underlying assumption, thus enabling us to provide what we deem to be a good balance between automatizing some governance processes while keeping a human-in-the-loop approach to maximize security.

The data scientists using Fed-BioMed, called *researchers* in our notation, are assumed to be knowledgeable users of at least one of the supported ML frameworks, and to be able to read API documentation and code basic Python functions complying with it. We also assume that they are able to discuss with biomedical researchers and data providers about technical aspects involving both the preparation of data (e.g. developing a common data model and format) and the development or interpretation of ML models for biomedical data. Coherently with the FL paradigm, we make the assumption that the edge nodes wish to protect their data from arbitrary access of the researchers while preventing data leakages, from either a malicious source or a manipulation mistake.

4.2 Design and functional architecture

The functional architecture of Fed-BioMed, as shown in Figure 2, is based on the design space described in Section 4.1 and the requirements identified in Section 2. The Fed-BioMed ecosystem comprises three architectural components: the network, the researcher, and one or more nodes. The network is responsible for brokering the communication between all Fed-BioMed components, the researcher is responsible for configuring and driving the federated learning experiment, as well as aggregating the trained models, while the nodes are responsible for local data governance and actually performing the training.

A standard training experiment is described in Figure 3, where nodes are first required to make their data available for training by inserting an appropriate metadata entry in a locally-stored database, and assigning unique identifying tags to those data. This process is simplified by the web-GUI built-in to our framework, but a CLI is also available for programmatic approaches. Optionally, node-specific customizations to the data loading process may be specified here through a plugin system called `DataLoadingPlan`, with the intention of reducing the data formatting burden by providing a logical layer between the researcher and the actual data format as stored locally. Then researchers define

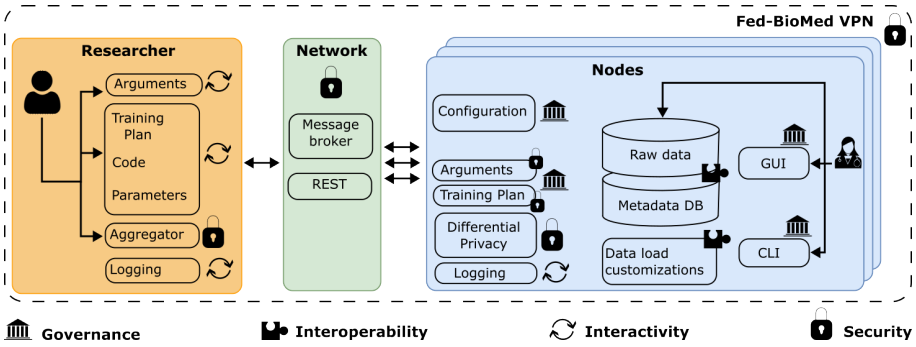

Figure 2: High-level architecture and design pillars of Fed-BioMed: node-side governance and control, interoperability with medical data standards and infrastructure, researcher interactivity, and data privacy and security. Fed-BioMed is composed of three components: the researcher, a data scientist responsible for designing and steering the training of the ML model; the nodes, i.e. the clinical data providers; and the network, responsible for brokering all communication between the researcher and nodes. Each component has been designed following the Fed-BioMed requirements (Section 2), and the figure highlights which requirement affects the architectural subcomponents.

a `TrainingPlan`, i.e. an object containing the description of the ML model and aggregation parameters, the code for the training and validation routines, a data loading and preprocessing routine, and other training-specific information such as e.g. the description of the optimizer. The `TrainingPlan` is packaged in an `Experiment` object along with the aggregator, training arguments, and data identifier tags.

The researcher then issues a train command through the network's message broker, which is broadcasted to all the nodes. Nodes that identify the requested data tags within their metadata database may begin training: first the `TrainingPlan` object is recreated on the node, then data are loaded, preprocessed using both node-specific customizations as well as researcher-specified functions, and finally the training routine of the training plan is executed. For convenience, Fed-BioMed comes with pre-packaged `TrainingPlans` for widely-used frameworks, thus requiring minimal configuration from researchers.

Node-side governance

Fed-BioMed empowers nodes by maximising the amount of control they have over the execution of a FL experiment. In Fed-BioMed, there is no notion of a trusted third party to which nodes must delegate full trust. For training, we make the assumption that datasets have already been treated to comply to existing data privacy regulations (e.g. pseudonymisation), and that they have been at least partially harmonised, both semantically and syntactically, to a pre-defined format common to all the nodes participating in the experiment.

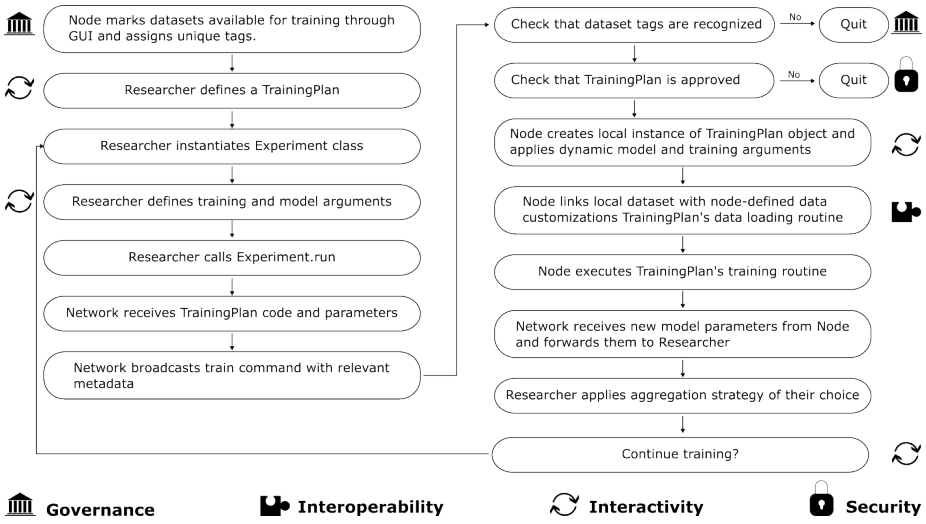

Figure 3: Prototypical FL training workflow in Fed-BioMed. Icons indicate the steps where the design was influenced by a particular requirement. First the nodes mark their dataset as available for federated training, while the researcher defines and obtains approval for a TrainingPlan. Then the researcher may launch multiple experiments, and within each experiment interactively launch multiple rounds of training. In each round, data are loaded locally on the nodes and the researcher-defined training routine is executed. Communication of model parameters and metadata always happens through the Network component.

When an experiment is launched, the datasets corresponding to the requested tags are loaded in the node's worker instance and prepared for training. A researcher-defined `training_data` function is then used to load the data and apply local pre-processing. Although this is not yet strictly enforced, this function is expected to use one of the dataset classes defined by Fed-BioMed providing controlled access to filesystem resources and best compatibility with the rest of the Fed-BioMed workflow. It is also possible to emulate a federated data pre-processing pipeline by assembling multiple experiments, thanks to Fed-BioMed's interactive approach.

Fed-BioMed's answer to the obvious tradeoff between allowing researchers to execute custom code snippets on the worker nodes and the associated security issues, is to provide a mechanism called *training plan approval*. When enabled, this feature prevents the execution of code contained in `TrainingPlans` that has not yet been inspected and approved by the node. A hash of the code is computed and checked at every training execution to prevent substitution attacks, and while this feature is not a definitive fail-safe measure against malicious users, it provides an additional layer of protection, as it empowers nodes and fosters researcher-node collaboration through shared responsibility. Furthermore, nodes are granted the right to override certain training parameters, regardless of the researcher's original request, for matters concerning security and resource usage.

Integration with biomedical data sources

Like most FL frameworks, Fed-BioMed does not offer a direct connection with raw data sources such as hospital EHR, PACS, and other clinical IT systems. Instead, clinical data managers are requested to perform a one-time data preparation task to extract and wrangle the data. Contrary to other frameworks, Fed-BioMed tries to minimize this effort by implementing dedicated mechanisms and offering built-in integration with widely-used data standards, at the cost of losing some generality by restricting its focus on biomedical data.

Fed-BioMed introduces the notion of a `DataLoadingPlan`, meaning a set of customizations configured by the node which allow changing the way data is presented to the researcher. Built-in `DataLoadingPlans` are already integrated in Fed-BioMed's GUI, thus providing a mechanism for data harmonization that does not require extensive effort on the node side.

Fed-BioMed also provides a built-in suite of dataset classes that provide an interface to common standards. For example, our `MedicalFolderDataset` class is inspired by the BIDS standard[8] as well as PyTorch's `ImageFolder`. Furthermore, our generic `TabularDataset` class supports any standard that may be reduced to csv format. As part of our continuous development effort, we plan to significantly expand this suite to include several other healthcare interoperability standards, depending on the needs of our collaborators.

Hospitals participating in the same FL experiment may have different com-

[8] `https://bids.neuroimaging.io/`

puting infrastructures. Fed-BioMed supports execution on containers, virtual machines, bare-metal CPUs and GPUs in the same experiment, allowing for the reconciliation of an heterogeneous computing infrastructure across nodes.

Researcher interactivity

The researcher-facing side of Fed-BioMed is a Python SDK for configuring and managing FL experiments, preferably via jupyter notebooks. To create an experiment, researchers first instantiate a class called `TrainingPlan` containing the definition of the federated data and the training routine. The code written by data scientists in the `TrainingPlan` is designed to be as similar as possible to the serial local version of the training loop. This interface is intentionally generic to support a wide variety of use cases, from workflows allowing to compute federated summary statistics to general distributed optimization based for example on Stochastic Gradient Descent (SGD), Expectation Maximization (EM), and Variational Inference (VI). For convenience, we offer specialized `TrainingPlans` for specific optimization strategies such as SGD, and for the ML framework being used.

Our approach promotes researcher interactivity via the `Experiment` class, which allows to easily set the participating nodes, the FL strategy, and includes the `TrainingPlan` itself. The `Experiment` is provided with a logging functionality and integration with tensorboard [38]. The modular design of the Fed-BioMed training loop allows researchers to dynamically adjust hyperparameters on the fly. Furthermore, a checkpointing system allows saving and loading the state of an experiment in persistent memory.

In a highly secure environment where training plans must be approved by each node, minor changes in a training plan may lead to insufferable delays for obtaining multiple approvals. Therefore, we make a distinction between the `TrainingPlan` source code and a set of training and model arguments. The former must be approved by the node, but it may leverage the latter as a way to dynamically configure details about the experiment that lie within the node's acceptable ranges. For example, a `TrainingPlan` may include a dropout layer but the dropout rate would be specified as a model argument, thus providing security guarantees to the node while allowing flexibility for the researcher.

Cybersecurity

Fed-BioMed 's current threat model assumes that nodes and server are honest-but-curious, nodes are independent actors and do not collude, and researchers may be malicious. This set of assumptions corresponds to our current application scenarios in hospital networks, where clients are trusted and strong protection is required with respect to researcher's manipulations. Thus, the first measure that we put in place is deploying the whole Fed-BioMed ecosystem inside a VPN, to effectively isolate the execution and protect it against external attacks. In the honest-but-curious model, an entity that gains access to the model weights may still attempt to re-identify individual samples through

model inversion attacks. Fed-BioMed implements both differential privacy as well as secure aggregation based on additively homomorphic encryption [27, 36], with currently some limitations concerning the supported ML frameworks and aggregation methods.

Fed-BioMed's architecture offers one final layer of protection, by insulating the researcher from the nodes through the existence of a middle component, the network, that brokers all communication between them. Furthermore, our `TrainingPlan` approval mechanism can also be viewed as a security feature, by allowing nodes to review any code intended to be executed within their systems.

In our short-term roadmap we plan to strengthen our security features by adopting a tighter threat model. The first measure that we intend to introduce is encrypted communication inside the VPN, preventing potentially malicious actors who have gained internal access from being able to listen-in on the communication exchanges. Secondly, we intend to introduce trusted digital certificates to improve our protection against impersonation attacks.

5 Status report

5.1 Current state of the implementation and future plans

Fed-BioMed is a constantly evolving project, with an active group of core maintainers striving to issue monthly releases. The current maturity of our library has been acknowledged to allow a first real-world deployment and validation within a federation of some members from the UniCancer consortium, as described in Section 5.2, thus characterizing its Technology Readiness Level as TLR 5[9]. However, we strive to constantly improve several aspects of the implementation in future releases, following a development roadmap inspired by the design pillars described in this paper and actively driven by the needs of our growing community of users. Building on our functional description in Section 4.2, we provide in the Supplementary Information 8.2 a fine-grained description of the core library, interface and cybersecurity functionalities currently implemented in Fed-BioMed.

5.2 Demonstration on real-world federated hospitals networks

To illustrate the capabilities of Fed-BioMed for real world applications of federated learning in hospitals networks, in this section we report the results obtained from the end-to-end deployment and training of a federated prostate segmentation model, based on the data hosted by three different medical institutes from the Unicancer consortium [10].

[9] `https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf`

[10] `https://www.unicancer.fr/en/`

The French hospitals involved are the *Centre Henri Becquerel*, Rouen, the *Institut Curie*, Orsay, and the *Centre Antoine Lacassagne*, Nice. Each hospital loaded into the Fed-BioMed client the respective dataset composed by prostate magnetic resonance images (MRIs), and associated prostate segmentation masks. The dataset considered for this test were the following:

- **Medical Segmentation Decathlon - Prostate** [1], composed by 32 prostate MRIs for training, with respective masks of peripheral and transition zone, merged into a single mask for the whole prostate. This dataset was hosted at *Institut Curie* (CURIE).
- **Promise12** [32], consisting of 50 training cases obtained with different scanners. Of those, 27 cases were acquired by using an endorectal coil. This dataset was hosted at *Centre Henri Becquerel* (CHB).
- **ProstateX** [2], containing prostate MRIs acquired by using two different scanners, providing prostate segmentation masks for 189 cases [11]. This dataset was hosted at *Centre Antoine Lacassagne* (CAL).

Each of the three hospitals was assigned a single dataset from the ones listed above, which was subsequently loaded into the Fed-BioMed node client. At each site, the data was further divided in a training subset (90% of samples) and a holdout subset (10% of samples). The resulting federated learning setup reflects a realistic scenario presenting the typical data heterogeneity of multicentric medical imaging applications (see Figure 4a and Appendix 8.1 for more details). The use of publicly available data is in the scope of this experiment, and is aimed at demonstrating the FL task on a reproducible benchmark, allowing the comparison of the results with respect to the centralized scenario. Moreover, this setup facilitated the ethical approval of this experiment by each clinical center, avoiding the cumbersome process related to the use of hospital data.

5.2.1 Experimental details

Within this setup, training hyperparameters were previously identified via a simulated FL scenario on the same data. The chosen aggregation method used was FedAvg [39], and the `TrainingPlanApproval` security feature was enabled. On the hospitals node side, the Fed-BioMed nodes were running on GPU-enabled machines, even though the existence of specific hardware is not a requirement for running Fed-BioMed. The central aggregator and the network component were hosted on a separate server provided by Inria. The training experiment consisted of 40 rounds with 25 local gradient updates each. Additional details are provided in the Supplementary Material 8.1.

5.2.2 FL does not affect final model's performance

After training, the model obtained a Dice score of 0.868. To assess the generalizability of this result, we performed a 5-fold cross-validation with similar

90-10 splits in both a centralized and a simulated federated setting. The cross-validation Dice score for the simulated FL model was 0.854 ± 0.028, while for the centralized (CL) model it was 0.850 ± 0.035. The difference in cross-validation scores between the simulated FL and CL models is not statistically significant ($p = 0.63$). Furthermore, the original model trained in the real-world setting has a realistic performance that falls within 0.5σ of the average of the simulated FL cross-validation scores. The distribution of Dice scores for all cross-validation folds combined, shown in Figure 4c, are qualitatively similar between the CL and FL models. The final segmentations are also visually indistinguishable, as shown in the bottom row of Figure 4d.

5.2.3 FL runtime overhead

Each round required on average just over 60 seconds of wallclock time for completion. Federated Learning was found to induce a significant overhead on training time, ranging from 39% to 56% of the overall experiment wallclock time, as shown in Figure 4b. This is in contrast with other studies that report negligible overhead [62, 53], and can be explained by the smaller dataset sizes in our experiment, associated with a smaller number of samples used for training within each round. We believe that our experiment represents a realistic common scenario for most real-world hospital deployments where data may be scarce and high performance computing not applicable. Some Fed-BioMed implementation details, such as an hard-coded delay at round initialization, may also impact this measure and we are investigating whether useful runtime gains could be made through software improvements.

6 Lessons learned

Answering to the specific needs of biomedical research applications The field of biomedical research is characterized by specific needs that do not necessarily generalize to other fields of application of FL, such as tighter regulations for data protection, very large degrees of heterogeneity, and others identified in Section 2.1. FL frameworks, on the other hand, are often developed with a generic approach in mind, striving for generality in terms of application domains. While this approach ensures that such frameworks can be more widely used, this may lead to a sub-optimal experience for users who are interested only in a specific application domain. We believe that some communities, and in particular the biomedical research field, may greatly benefit from personalized framework development approaches able to cater to specific requirements such as providing the necessary governance tools, ensuring their usability by members of the application domain's community, and providing a software architecture tailored to the privacy or performance needs.

Model development and debugging in a federated setting The development of a new ML model almost always requires an initial exploratory

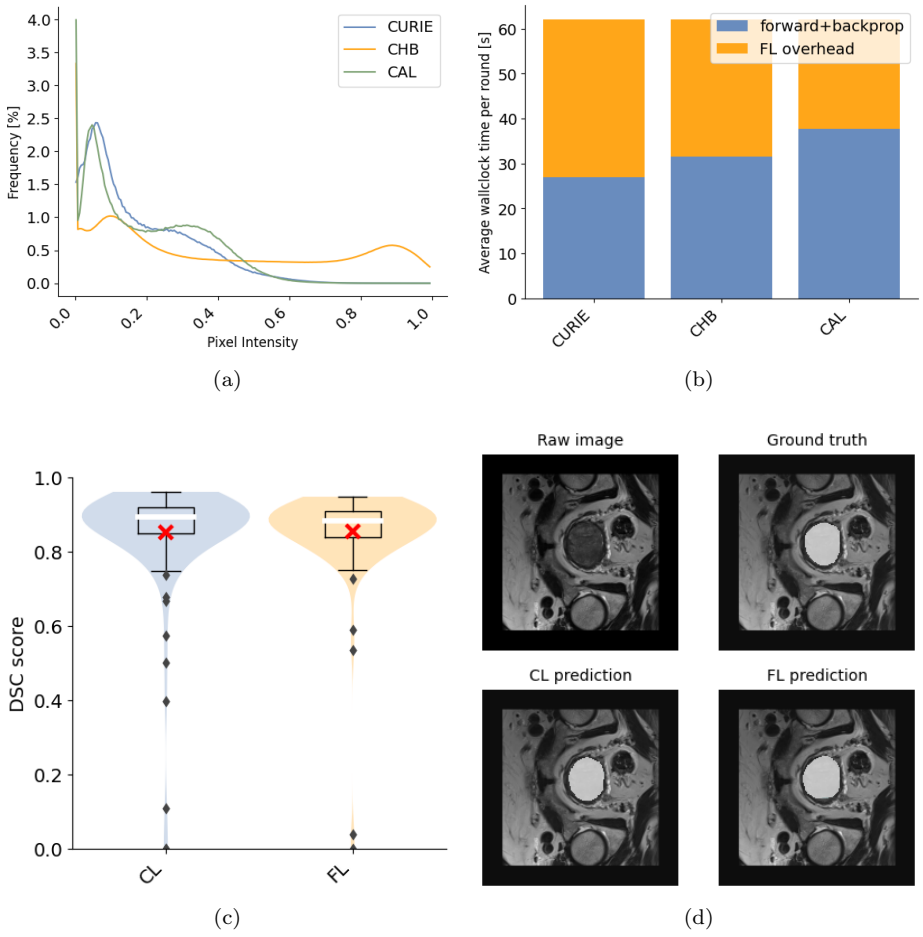

Figure 4: **a)** Pixel intensity distribution grouped by clinical site. Prostate images exhibit significant differences between sites, especially in the case of Site 2. **b)** Breakdown of FL experiment wallclock runtime. FL introduces a significant overhead in our case, likely attributable to the relatively small number of samples seen by the model in each FL round. **c)** Distribution of Dice scores for centralized (CL) and federated (FL) models, combining all cross-validation folds. The performance of the two models has a statistically significant difference, but with a small effect size. The figure shows boxplots inside the violin plots. The top bottom of each boxplot depict the 3rd and 1st quartile of each measure. The white line and the red x indicate the median and mean values, respectively. The whiskers depict the extremal observations still within 1.5 times the interquartile range. **d)** Clockwise from top left: an example raw image, the ground truth segmentation, the FL model prediction and the CL model prediction.

phase where data are examined, data preprocessing pipelines are designed, the model architecture is refined, and finally hyperparameters are tuned. Most of the strategies for such operations usually involve some data manipulation and observation, for example it may be insightful to examine the data samples with the highest losses for a given model. In a federated setting, all of this is not possible because data are not allowed to travel outside of the source institution. This leads to a mismatch between who has the right to examine the data — i.e. the clinical data sources —, and who has the knowledge to interpret it — i.e. the data scientists —. Fed-BioMed is designed to operate in high-trust environments where the communication between these two entities is encouraged, therefore we may imagine a scenario where the data scientist instructs an operator on the clinical side to look at specific images and try to identify abnormal patterns. Furthermore, our modular `TrainingPlan` design can support advanced explainable techniques, while still guaranteeing node's data privacy through our `TrainingPlan` approval process. Despite these mitigation efforts, the issue of remotely debugging ML models in a privacy-preserving way remains a difficult problem to solve.

Data preparation prior to federated training Data in clinical databases are typically stored in proprietary formats that are not generally readily usable for ML analysis. However, all FL frameworks, including Fed-BioMed, make the assumption that the input data is provided in a format ready for ingestion by an ML model, or maybe a preprocessing pipeline. This gap has been, in our experience, a major source of frustration and delays, with the burden of data export and conversion falling usually on clinical data managers lacking the budget and training for such operations. Fed-BioMed tries to mitigate this by providing `Dataset` classes that can be populated with data that has been exported but not highly transformed, for example our `MedicalFolderDataset` class that can handle data in a simplified BIDS structure. In the future, we also plan to extend list of supported formats with some that are closer to the raw data exported by hospital IT systems, such as DICOM for imaging, OMOP or FHIR for clinical data, VCF files for genomics, and others. An ideal scenario would include a direct integration with the hospital PACS or EHR systems, however the proprietary nature of most of this software and the strict security rules unfortunately make this an unlikely scenario. Regardless of the feasibility, we believe that FL frameworks targeted at biomedical research applications must simplify the process of data preparation and be able to interoperate with standard data formats in their raw form, to ease the burden on clinical data managers and improve data reusability.

Data privacy and regulation for real-world deployment Contrary to controlled academic environments, real-world deployment scenarios are characterized by strict privacy and security measures. Some notable examples include: avoiding training if a client's dataset has too few samples, requiring secure aggregation and differential privacy measures, restricting the data flow in the net-

work, and ensuring compliance with hospitals firewall policies. The integration in Fed-BioMed of such security measures to meet the requirements asked by Data Protection Officers of hospitals has been a long, iterative process consisting of multiple rounds of discussion needed to achieve a common understanding and a shared vocabulary. Ultimately, this dialectic process is what allowed us to carry out our first deployment in a collaborative clinical project.

7 Conclusion

We have presented version v4.1 of Fed-BioMed, a Python-based FL framework aimed at supporting real-world deployments for biomedical and healthcare research applications. This document focuses on Fed-BioMed's philosophy, guiding principles, and relevant details of the current architecture. Fed-BioMed is an open-source project available in our public repository[11] under an Apache license, and we welcome contributions from the community. We also provide extensive user documentation and tutorials, in addition to an API reference for developers[12]. The development of Fed-BioMed is an ongoing effort currently brought forward by a group of vetted collaborators coordinated by a small number of core developers. In the future, we plan to grow our community of users and developers, and integrate open-source contributions through a consortium. Our short-term implementation roadmap includes finalizing our implementation of Secure Aggregation, securing client-server communications, and improving even further the usability for clinical data providers. The domain of healthcare and clinical research is characterized by specific challenges and requirements that are difficult or impossible to satisfy with the currently available frameworks. We hope that Fed-BioMed will lower the threshold to apply FL in this domain, enabling clinical data scientists to perform experiments and train ML models within a federated consortium of data providers in a secure, practical, and privacy-preserving way.

8 Supplementary Information

8.1 Real-world use case scenario: prostate segmentation

Data were gathered following the process described in [25] from three public datasets [1, 32, 2]. Each clinical site received data corresponding to a single source, to reflect the more realistic scenario of high heterogeneity of data among clinical sites. Details of the data distribution are given in Table 3, while details of the experiment setup are given in Table 4. We used Fed-BioMed v4.1.2 relied on MONAI's (v1.0.1) implementation of UNet [29] and data transformations, and on Pytorch's SGD implementation, for definition of the model and optimizer in each site's local training.

[11] `https://gitlab.inria.fr/fedbiomed/fedbiomed`

[12] `https://fedbiomed.gitlabpages.inria.fr/v4.1/getting-started/what-is-fedbiomed/`

Center	Dataset	Train samples	holdout samples
CAL	ProstateX [2]	147	37
CHB	Promise [32]	21	6
CURIE	Decathlon [1]	25	7

Table 3: Distribution of data among clinical sites during the FL experiment, including the number of samples in each cross-validation fold.

Data	Transformation	Center cropping and padding
	Common shape	320, 320, 16
	Transformation	Intensity normalization
UNet	channels	16, 32, 64, 128, 256
	strides	2, 2, 2, 2
	residual units	3
	normalization	Batch normalization
	dropout	0.3
Local	Optimizer	SGD
	Learning rate	0.1
	Momentum	0.9
	Batch size	8
	Local updates	25
	Loss	Dice loss
Federated	Aggregation	FedAVG
	Rounds	40

Table 4: FL experimental setup details.

8.2 Technical implementation details

Building on our functional description in Section 4.2, we provide here a more detailed explanation of Fed-BioMed's implementation details, as shown in Figure 2, trying to make a clear distinction between those that, as of today, have already reached a satisfactory level of maturity and those that are expected to improve in the near future, or may change at some point due to the dynamic nature of our project.

8.2.1 Core library functionalities

On the node side we have implemented: a database of dataset metadata based on the TinyDB[13] package; a task manager based on persist-queue[14]; a suite of `Dataset` classes to represent tabular, medical imaging (BIDS [22]), and other data formats; and a `Round` class handling all the logic for the execution of one federated training round. In the future, we plan to vastly extend our library of supported data formats to include other imaging formats such as e.g. DICOM, add support for NLP formats, and integrate interoperability standards and data models such as i2b2 and OMOP[42, 41, 55]. Furthermore, we may consider moving to more complex database solutions if dataset management becomes too cumbersome, and similarly we may consider more advanced task management systems should the need arise among clinical data providers for better resource handling and improved performance. On the researcher side, an `Experiment` class represents the entry-point for the configuration and steering of a FL experiment, while the `TrainingPlan` class contains all the logic and code to be shipped to nodes for remote execution. The highly interactive design of the researcher-facing classes represents one of the highlights of Fed-BioMed, and in the future we plan to improve this feature by moving towards an ever more modular design with clearer separation of responsibilities, a hierarchical approach, and improved user documentation.

Communication between the researcher and the nodes happens through an intermediate component called the `Network`. This component is based on two technologies: MQTT for the brokering of short messages (e.g. train, search datasets, and others) [3], and an HTTP API based on Django REST[15] for exchanging larger files such as model parameters. MQTT was chosen because of its natural mapping to a star topology and good support of broadcast operations, which we use for discovering datasets and clients. However, in the future we plan to evaluate other approaches, more commonly used in the literature, such as relying on gRPC or other remote execution protocols. In the current implementation, nodes are pure slaves that execute commands issued by the researcher. In the future, we plan to evaluate endowing nodes with a more active role, for example inverting the direction of communication such that the nodes would request tasks instead of passively receiving them, which would entail se-

[13] `https://pypi.org/project/tinydb/`
[14] `https://github.com/peter-wangxu/persist-queue`
[15] `https://www.django-rest-framework.org/`

curity benefits (reduced attack surface on the nodes), while shifting some of the complexity from the node implementation to the network component.

8.2.2 User interfaces

In terms of user interfaces, on the node side we have implemented both a Command Line Interface (CLI) and a Graphical User Interface (GUI). Our GUI implementation is composed of a web client based on React, and a Flask backend server that manages the interactions with the Fed-BioMed library, in particular with the dataset database. This simplifies the process of managing datasets (CRUD operations [37]), training plan approval by data science experts on the node side, and has the general goal of facilitating governance operations on data and models. Currently, the GUI is limited to executing on the same machine as the node process, but in the future we plan to decouple the two through a micro-services approach, as well as improve the RBAC with better security and account management. Another feature that we plan to implement in the future is adding experiment monitoring functionalities for the node, as well as allowing basic experiment steering functionalities such as stopping an experiment through the GUI. The researcher's interface is a Python SDK, mainly designed to be executed by interpreter such as Jython and within notebooks. The researcher may monitor the training via Tensorboard, which plots common metrics such as training loss as well as custom ones defined via a plugin system. Defining the best interface for the researcher in terms of tradeoff between interactivity, security, and simplicity is one of our main goals, and we constantly strive to improve the API for the `Experiment` and `TrainingPlan` classes.

8.2.3 Cybersecurity and model security

Cybersecurity is one of the main design pillars of Fed-BioMed. As described above, we have implemented a VPN deployment mode based on the Wireguard framework to isolate the execution of Fed-BioMed from external parties [16]. The VPN deployment mode is shipped with the software, and designed to be easily configurable and deployable. One of our short-term goals is to implement TLS-encrypted communication within the VPN, by exploring suitable MQTT and Django-REST extensions. To protect nodes from the execution of arbitrary code, a training plan approval mode can be enabled. This process is handled by the GUI on the Node side, making it easy and immediate for clinical data providers to conduct their reviews. In this configuration, when a researcher sends a train request to the node, the hash of the training plan is compared to the hashes of a set of training plans that have been previously reviewed and approved by the node, and is executed only in the case of a match. Differential Privacy can also be enabled in Fed-BioMed, when using PyTorch as the training backend, by leveraging the Opacus library and Tensorflow's accountant[16]. Secure aggregation is being implemented based on additively homomor-

[16] `https://raw.githubusercontent.com/tensorflow/privacy/7eea74a6a1cf15e2d2bd890722400edd0e470db8/research/hyperparameters_2022/rdp_`

phic encryption [27, 36], with the computation of keys based on a multi-party computation (MPC) approach inspired by the MP-SPDZ benchmark suite [28].

References

[1] Michela Antonelli et al. "The medical segmentation decathlon". In: *Nature communications* 13.1 (2022), p. 4128.

[2] Samuel G Armato III et al. "PROSTATEx Challenges for computerized classification of prostate lesions from multiparametric magnetic resonance images". In: *Journal of Medical Imaging* 5.4 (2018), pp. 044501–044501.

[3] Andrew Banks et al. *MQTT Version 5.0*. Tech. rep. 2019.

[4] Daniel J Beutel et al. "Flower: A friendly federated learning research framework". In: *arXiv preprint arXiv:2007.14390* (2020).

[5] Keith Bonawitz et al. "Towards federated learning at scale: System design". In: *Proceedings of Machine Learning and Systems* 1 (2019), pp. 374–388.

[6] Nader Bouacida and Prasant Mohapatra. "Vulnerabilities in federated learning". In: *IEEE Access* 9 (2021), pp. 63229–63249.

[7] Kenneth Clark et al. "The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository". In: *Journal of digital imaging* 26.6 (2013), pp. 1045–1057.

[8] Mellody consortium. *Mellody*. Tech. rep. Accessed on 28th November 2022. 2019. URL: `https://www.melloddy.eu/`.

[9] Matthew G Crowson et al. "A systematic review of federated learning applications for biomedical data". In: *PLOS Digital Health* 1.5 (2022), e0000033.

[10] Marc Cuggia and Stéphanie Combes. "The French Health Data Hub and the German Medical Informatics Initiatives: two national projects to promote data sharing in healthcare". In: *Yearbook of medical informatics* 28.01 (2019), pp. 195–202.

[11] Renato Cuocolo et al. "Quality control and whole-gland, zonal and lesion annotations for the PROSTATEx challenge public dataset". In: *European Journal of Radiology* 138 (2021), p. 109647.

[12] European Union Agency for Cybersecurity (ENISA). *ICT Security Certification Opportunities in the Healthcare Sector*. 2018.

[13] Erfan Darzidehkalani, Mohammad Ghasemi-Rad, and Pma van Ooijen. "Federated Learning in Medical Imaging: Part I: Toward Multicentral Health Care Ecosystems". In: *Journal of the American College of Radiology* (2022).

`accountant.py`

[14] Ittai Dayan et al. “Federated learning for predicting clinical outcomes in patients with COVID-19”. In: *Nature medicine* 27.10 (2021), pp. 1735–1743.

[15] Timo M. Deist et al. “Distributed learning on 20 000+ lung cancer patients – The Personal Health Train”. en. In: *Radiotherapy and Oncology* 144 (2020), pp. 189–200. DOI: 10.1016/j.radonc.2019.11.019. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167814019334899.

[16] Jason A Donenfeld. “Wireguard: next generation kernel network tunnel”. In: *NDSS*. 2017, pp. 1–12.

[17] Patrick Foley et al. “OpenFL: the open federated learning library”. In: *Physics in Medicine & Biology* (2022). DOI: 10.1088/1361-6560/ac97d9. URL: http://iopscience.iop.org/article/10.1088/1361-6560/ac97d9.

[18] Cloud Native Computing Foundation. *high performance, open-source universal RPC framework*. Accessed on 16 December 2022. 2018. URL: %5Curl%7Bhttps://grpc.io/%7D.

[19] David Froelicher et al. “Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption”. en. In: *Nature Communications* 12.1 (Oct. 2021), p. 5910. DOI: 10.1038/s41467-021-25972-y. URL: https://www.nature.com/articles/s41467-021-25972-y.

[20] Mathieu N Galtier and Camille Marini. “Substra: a framework for privacy-preserving, traceable and collaborative machine learning”. In: *arXiv preprint arXiv:1910.11567* (2019).

[21] Marzyeh Ghassemi et al. “A Review of Challenges and Opportunities in Machine Learning for Health”. In: *AMIA Summits on Translational Science Proceedings* 2020 (May 2020), pp. 191–200. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7233077/.

[22] Krzysztof J Gorgolewski et al. “The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments”. In: *Scientific data* 3.1 (2016), pp. 1–9.

[23] Chaoyang He et al. “Fedml: A research library and benchmark for federated machine learning”. In: *arXiv preprint arXiv:2007.13518* (2020).

[24] Tim Hulsen et al. “From big data to precision medicine”. In: *Frontiers in medicine* (2019), p. 34.

[25] Lucia et al. Innocenti. “Consensus-Based Methods as a Cost-Effective Alternative to Federated Learning: Benchmark on Prostate MRI Segmentation”. In: (2023). under review.

[26] Clifford R Jack Jr et al. “The Alzheimer’s disease neuroimaging initiative (ADNI): MRI methods”. In: *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine* 27.4 (2008), pp. 685–691.

[27] Marc Joye and Benoît Libert. "A scalable scheme for privacy-preserving aggregation of time-series data". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2013, pp. 111–125.

[28] Marcel Keller. "MP-SPDZ: A Versatile Framework for Multi-Party Computation". In: *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 2020. DOI: 10.1145/3372297.3417872. URL: https://doi.org/10.1145/3372297.3417872.

[29] Eric Kerfoot et al. "Left-ventricle quantification using residual U-Net". In: *Statistical Atlases and Computational Models of the Heart. Atrial Segmentation and LV Quantification Challenges: 9th International Workshop, STACOM 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers 9*. Springer. 2019, pp. 371–380.

[30] Adrienne Kline et al. "Multimodal machine learning in precision health: A scoping review". In: *npj Digital Medicine* 5.1 (2022), pp. 1–14.

[31] Qinbin Li et al. "A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection". In: *IEEE Transactions on Knowledge and Data Engineering* (2021), pp. 1–1. DOI: 10.1109/TKDE.2021.3124599.

[32] Geert Litjens et al. "Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge". In: *Medical image analysis* 18.2 (2014), pp. 359–373.

[33] Yang Liu et al. "FATE: An Industrial Grade Platform for Collaborative Learning With Data Protection." In: *J. Mach. Learn. Res.* 22.226 (2021), pp. 1–6.

[34] Heiko Ludwig et al. "Ibm federated learning: an enterprise framework white paper v0. 1". In: *arXiv preprint arXiv:2007.10987* (2020).

[35] Yanjun Ma et al. "PaddlePaddle: An open-source deep learning platform from industrial practice". In: *Frontiers of Data and Domputing* 1.1 (2019), pp. 105–115.

[36] Mohamad Mansouri, Melek Önen, and Wafa Ben Jaballah. "Learning from failures: Secure and fault-tolerant aggregation for federated learning". In: *ACSAC 2022, Annual Computer Security Applications Conference, 5-9 December 2022, Austin, Texas, USA*. Ed. by ACM. © ACM, 2022. This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in ACSAC 2022, Annual Computer Security Applications Conference, 5-9 December 2022, Austin, Texas, USA https://doi.org/10.1145/3564625.3568135. Austin, 2022.

[37] James Martin. *Managing the data base environment*. Prentice Hall PTR, 1983.

[38] Martín Abadi et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems". In: (2015). Software available from tensorflow.org. URL: https://www.tensorflow.org/.

[39] H McMahan et al. "Communication-efficient learning of deep networks from decentralized data. arXiv e-prints". In: *Artificial intelligence and statistics. PMLR.* (2017).

[40] Bjoern H Menze et al. "The multimodal brain tumor image segmentation benchmark (BRATS)". In: *IEEE transactions on medical imaging* 34.10 (2014), pp. 1993–2024.

[41] Shawn N Murphy et al. "Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2)". In: *Journal of the American Medical Informatics Association* 17.2 (2010), pp. 124–130.

[42] Mario Mustra, Kresimir Delac, and Mislav Grgic. "Overview of the DICOM standard". In: *2008 50th International Symposium ELMAR*. Vol. 1. IEEE. 2008, pp. 39–44.

[43] Jean Ogier du Terrail et al. "Federated learning for predicting histological response to neoadjuvant chemotherapy in triple-negative breast cancer". In: *Nature Medicine* (2023), pp. 1–12.

[44] Roseline Oluwaseun Ogundokun et al. "A Review on Federated Learning and Machine Learning Approaches: Categorization, Application Areas, and Blockchain Technology". en. In: *Information* 13.5 (May 2022), p. 263. DOI: 10.3390/info13050263. URL: https://www.mdpi.com/2078-2489/13/5/263.

[45] Owkin. *Healthchain*. Tech. rep. Accessed on 28th November 2022. 2019. URL: https://www.labelia.org/fr/healthchain-project.

[46] Owkin. *Substrafl overview*. Tech. rep. Accessed on: 28th November 2022, Revision b152eecf. 2022. URL: https://docs.substra.org/en/stable/substrafl_doc/substrafl_overview.html#.

[47] Sarthak Pati et al. "Federated Learning Enables Big Data for Rare Cancer Boundary Detection". In: *Nature Communication* (2022).

[48] Nicola Rieke et al. "The future of digital health with federated learning". In: *NPJ digital medicine* 3.1 (2020), pp. 1–7.

[49] Holger 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.

[50] Holger R Roth et al. "FLARE: Federated Learning from Simulation to Real-World". In: *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*.

[51] Theo Ryffel et al. "A generic framework for privacy preserving deep learning". In: *arXiv preprint arXiv:1811.04017* (2018).

[52] Adam Sadilek et al. “Privacy-first health research with federated learning”. In: *NPJ digital medicine* 4.1 (2021), pp. 1–8.

[53] Oliver Lester Saldanha et al. “Swarm learning for decentralized artificial intelligence in cancer histopathology”. en. In: *Nature Medicine* 28.6 (June 2022), pp. 1232–1239. DOI: 10.1038/s41591-022-01768-5. URL: https://www.nature.com/articles/s41591-022-01768-5.

[54] Karthik V Sarma et al. “Federated learning improves site performance in multicenter deep learning without data sharing”. In: *Journal of the American Medical Informatics Association* 28.6 (2021), pp. 1259–1264.

[55] Observational Health Data Sciences and Informatics. *The Book of OHDSI*. Independently published, 2019. DOI: \url{https://doi.org/10.5281/zenodo.4265255}.

[56] Momina Shaheen et al. “Applications of federated learning; Taxonomy, challenges, and research trends”. In: *Electronics* 11.4 (2022), p. 670.

[57] Chi-Ren Shyu et al. “A systematic review of federated learning in the healthcare area: From the perspective of data properties and applications”. In: *Applied Sciences* 11.23 (2021), p. 11191.

[58] Olaf Sporns, Giulio Tononi, and Rolf Kötter. “The human connectome: a structural description of the human brain”. In: *PLoS computational biology* 1.4 (2005), e42.

[59] Cathie Sudlow et al. “UK biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age”. In: *PLoS medicine* 12.3 (2015), e1001779.

[60] K. Tomczak, P. Czerwińska, and M. Wiznerowicz. “The cancer genome atlas (tcga): an immeasurable source of knowledge.” In: *Contemporary Oncology* 19 (2015).

[61] Eric J Topol. “High-performance medicine: the convergence of human and artificial intelligence”. In: *Nature medicine* 25.1 (2019), pp. 44–56.

[62] Daniel Truhn et al. “Encrypted federated learning for secure decentralized collaboration in cancer image analysis”. en. In: (July 2022). DOI: 10.1101/2022.07.28.22277288. URL: https://www.medrxiv.org/content/10.1101/2022.07.28.22277288v1.

[63] Willem G Van Panhuis et al. “A systematic review of barriers to data sharing in public health”. In: *BMC public health* 14.1 (2014), pp. 1–9.