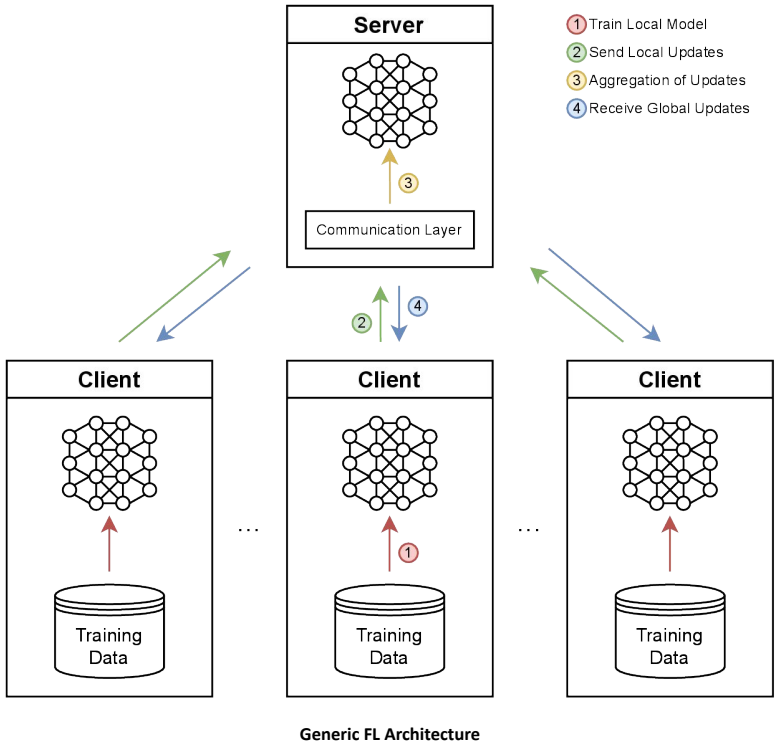
Impact Analysis of Different Consensus, Participant Selection and Scoring Algorithms in Blockchain-based Federated Learning Systems Using a Modular Framework

Henrique Afonso Coelho Dias (Nr. 1531484)

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Federated Learning

- Federated Learning (FL)[1] allows multiple distributed clients to collaborate on training the same Machine Learning (ML) model *without sharing raw data*.
- Currently, most FL networks have a central server that coordinates the federated training process, leading to a *single point of failure*[2].
- **Possible solution?** Combine Blockchain with Federated Learning!

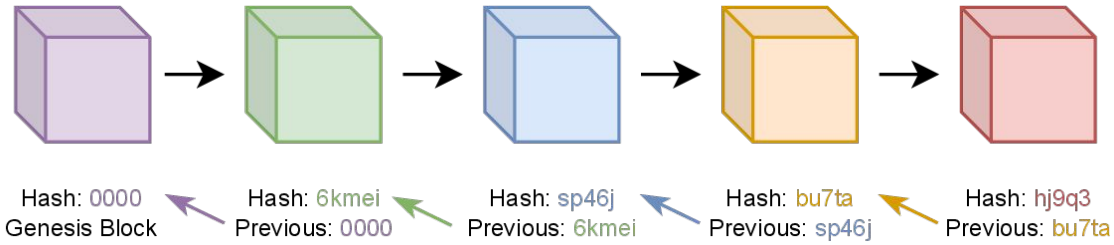


Generic FL Architecture

1. McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. Communication-efficient learning of deep networks from decentralized data.
2. Li, D., Han, D., Weng, T.-H., Zheng, Z., Li, H., Liu, H., Castiglione, A., and Li, K.-C. Blockchain for federated learning toward secure distributed machine learning systems: a systemic survey. Soft Computing (Nov. 2021).

Blockchain

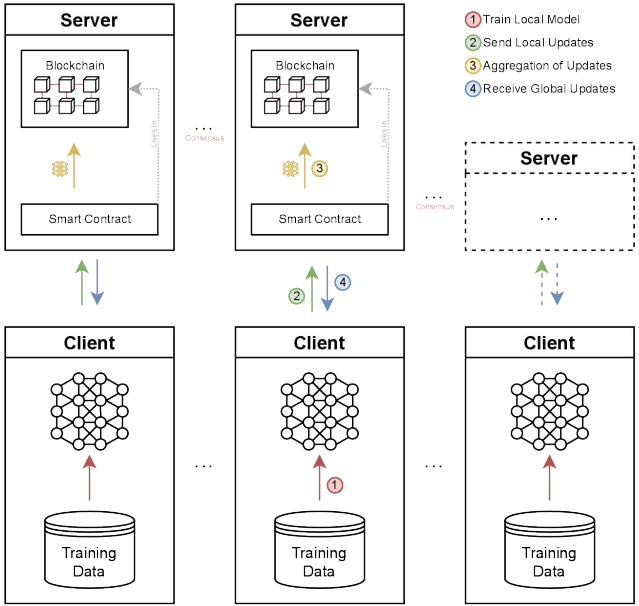
- Distributed ledger maintained by distributed computers.
- Guarantees decentralization[1].
- Facilitates traceability[2], auditability[2], data persistency[2], and authentication[1].
- Transparent reward distribution[3].
- Can support smart contracts[4].
- Has a consensus algorithm.



Blockchain Representation

1. Qu, Y., Uddin, M. P., Gan, C., Xiang, Y., Gao, L., and Yearwood, J. Blockchain-Enabled Federated Learning: A Survey. ACM Computing Surveys (Mar. 2022), 3524104.
2. Bonawitz, K. et al.. Towards federated learning at scale: System design. In Proceedings of Machine Learning and Systems (2019), A. Talwalkar, V. Smith, and M. Zaharia, Eds., vol. 1, pp. 374–388.
3. Li, D., Han, D., Weng, T.-H., Zheng, Z., Li, H., Liu, H., Castiglione, A., and Li, K.-C. Blockchain for federated learning toward secure distributed machine learning systems: a systemic survey. Soft Computing (Nov. 2021).
4. Wang, S., Yuan, Y., Wang, X., Li, J., Qin, R., and Wang, F.-Y. An overview of smart contract: Architecture, applications, and future trends. In 2018 IEEE Intelligent Vehicles Symposium (IV) (2018), pp. 108–113.

Blockchain-based Federated Learning (BFL)



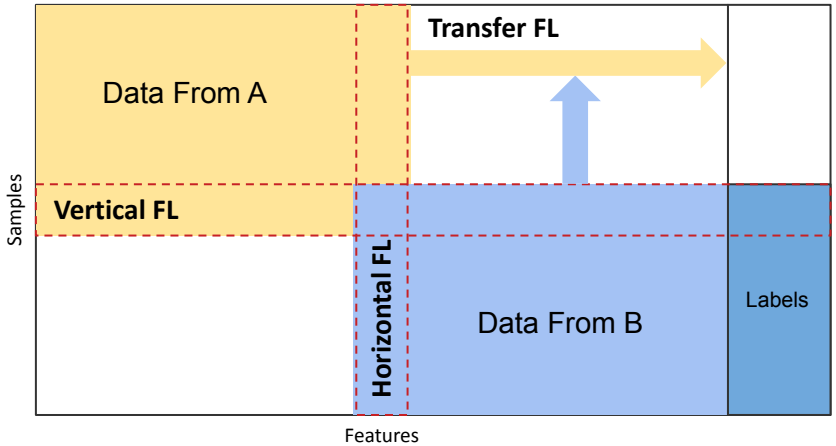
Generic BFL Architecture

- No more single point of failure!
- **Consensus Algorithm**: blockchain process consensus. E.g.: PoW, PoA, QBFT.
- **Participant Selection Algorithms**: how many and which clients participate in each round. E.g.: random, first-come first-served.
- **Scoring and Aggregation Algorithms**: classify client model updates, to prevent poisoning[1] and plagiarism attacks[2]. E.g.: BlockFlow, Marginal Gain, Multi-KRUM.
- **Privacy Mechanisms:** prevent inference attacks[3]. E.g.: Local Differential Privacy through a randomized algorithm A to apply noise to the original data[4]

1. Qu, Y., Pokhrel, S. R., Garg, S., Gao, L., and Xiang, Y. A blockchained federated learning framework for cognitive computing in industry 4.0 networks. IEEE Transactions on Industrial Informatics 17, 4 (2021), 2964–2973.
2. Ma, C., Li, J., Shi, L., Ding, M., Wang, T., Han, Z., and Poor, H. V. When federated learning meets blockchain: A new distributed learning paradigm. IEEE Computational Intelligence Magazine 17, 3 (2022), 26–33.
3. Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol. 10, 2 (jan 2019).
4. Wei, K., Li, J., Ma, C., Ding, M., Wei, S., Wu, F., Chen, G., and Ranbaduge, T. Vertical federated learning: Challenges, methodologies and experiments, 2022.

State of the Art

- Consensus Algorithms: PoW, PoA, PoS.
- Model Parameter Storage: mostly off-chain.
- Model Update Submission Validation: mostly through scoring algorithms.
- Privacy Mechanisms: Local Differential Privacy, Homomorphic Encryption.
- Data Partition: mostly horizontal.



Federated Learning Categories (Adapted From 1)

1. Adapted From Yang, Q., Liu, Y., Chen, T., and Tong, Y. Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol. 10, 2 (jan 2019).

Open Problems In The Literature

- Lack of analysis on how different components of a BFL system impact the accuracy, communication and computation costs of the system - specially important in IoT networks.
- No open-source or modular BFL framework that can be re-used by others.
- No implementation of Vertical Federated Learning in a BFL setting.

Main Research Question

What is the impact of different consensus, participant selection and scoring algorithms in a Blockchain-based Federated Learning system on execution time, convergence and accuracy, as well as communication and computation costs?

Framework Design & Implementation

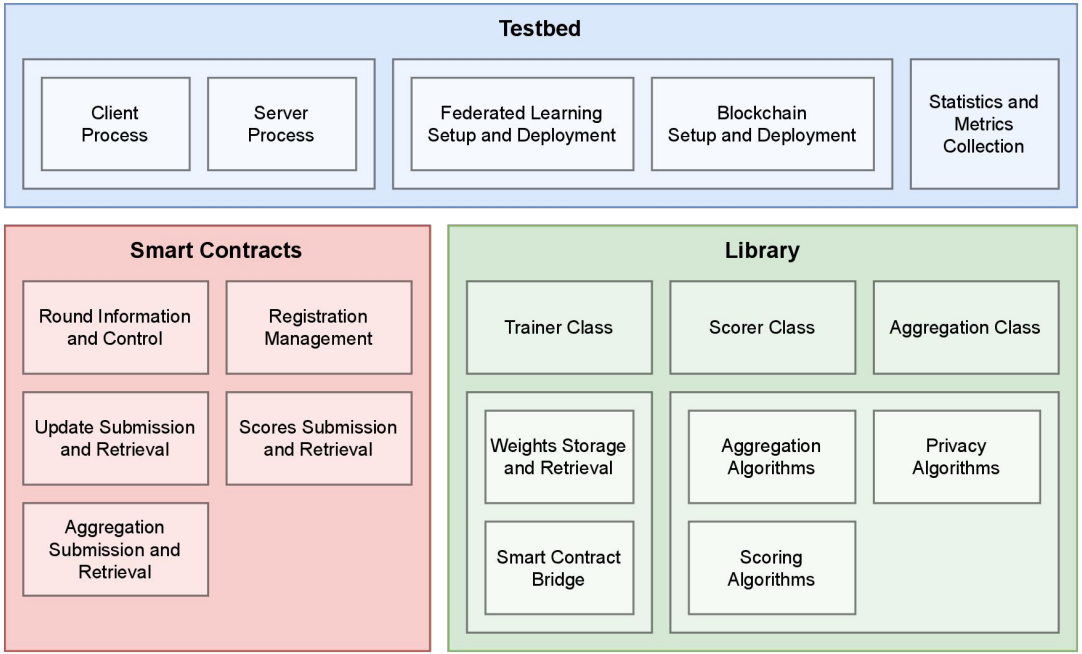
Framework Design

BlockLearning

- Devices are categorized as: trainers, aggregators, scorers.
- Model owner: who deploys the system and starts the training process.
- Modular execution flow.
- 50% threat model.

BlockLearning's Execution Flow

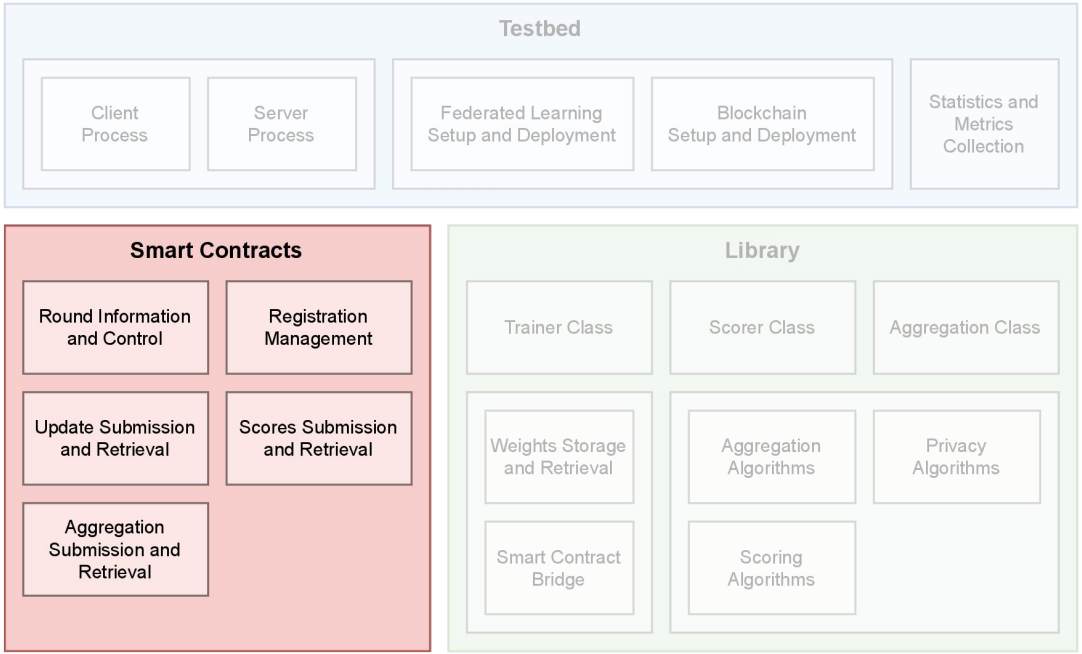
Components



BlockLearning's Structure and Modules

Framework Design

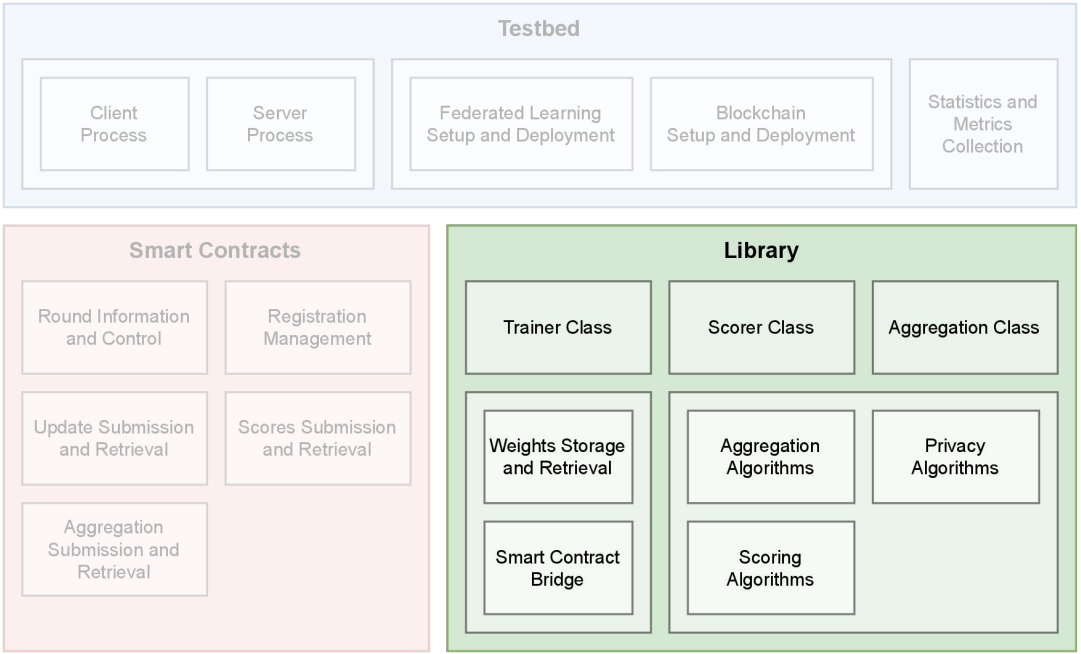
Components



BlockLearning's Structure and Modules

Framework Design

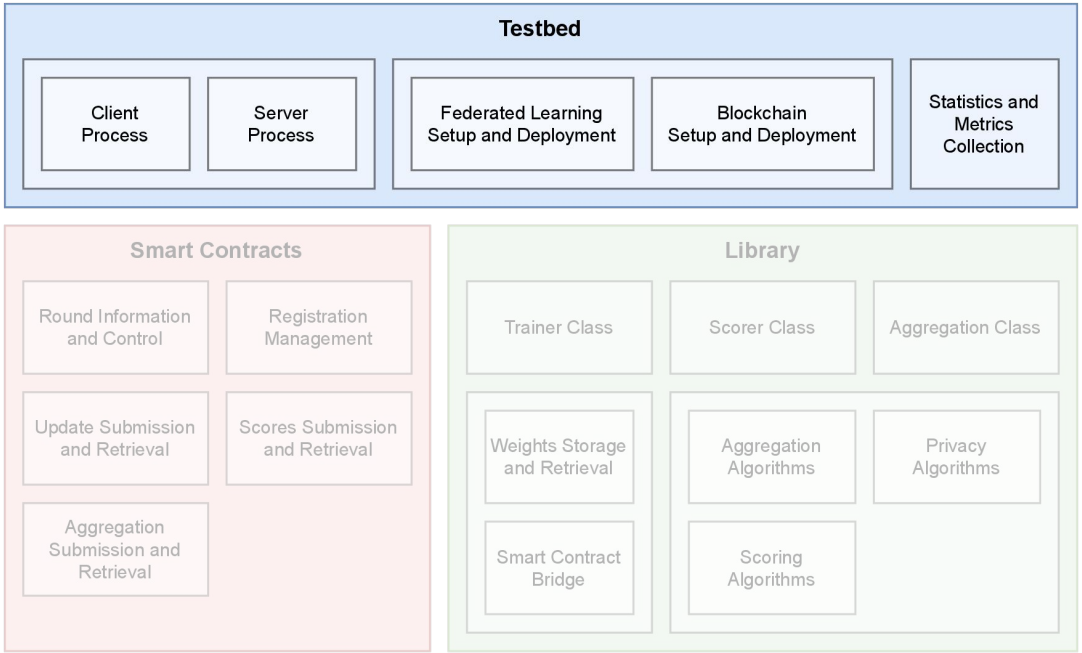
Components



BlockLearning's Structure and Modules

Framework Design

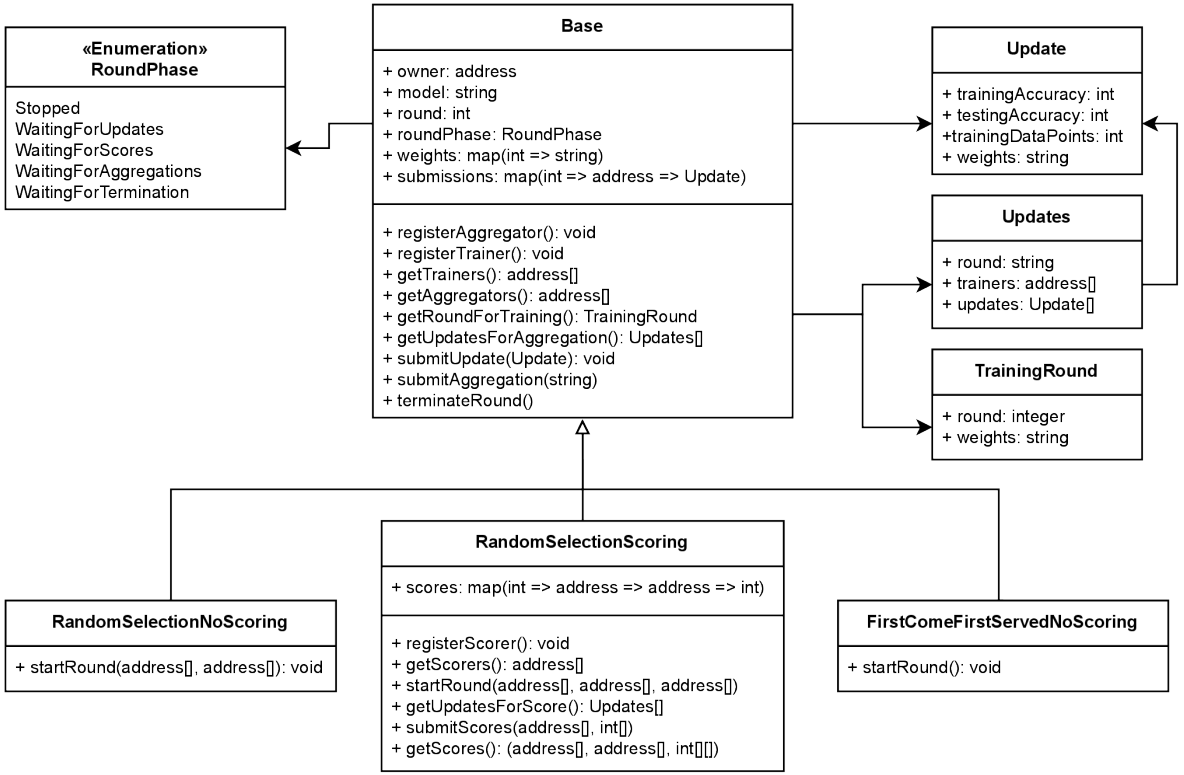
Components



BlockLearning's Structure and Modules

Smart Contracts

- Platform: Ethereum[1]
- Language: Solidity
- `Base` provides the common data structures and functionality.
- Other smart contracts provide specific functionalities.



Smart Contracts Class Diagram

1. Ethereum, https://ethereum.org

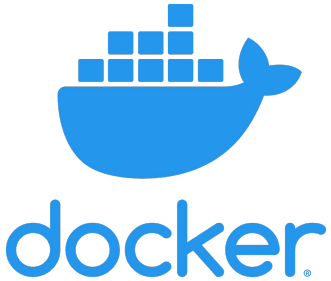
Library

- Language: Python[1]
- Aggregation, Scoring and Privacy Algorithms
  - Common interfaces, e.g.:
    - `score(round, trainers, updates) → trainers, scores`
  - Used IBM Differential Privacy Library[2]
- Weights Storage and Retrieval
  - Common interface
  - InterPlanetary File System[3]
- Smart Contract Bridge
  - Common interface
  - Web3.py[4]
- Trainer, Scorer and Aggregator Classes
  - Common interface
  - `train(), score(), aggregate()`

1. Python, https://www.python.org
2. Diffprivlib: The IBM Differential Privacy Library, https://diffprivlib.readthedocs.io
3. InterPlanetary File System (IPFS), https://ipfs.tech
4. Web3.py, https://web3py.readthedocs.io

Testbed

- Execution Environment: Docker[1]
- Client, Server and Owner Scripts
  - Implemented using the BlockLearning library.
  - Load data, initialize algorithms, run procedure.
- Blockchain Setup and Deployment
  - `go-ethereum`[2], and `quorum`[3]
  - Docker Compose
- Federated Learning Setup and Deployment
  - Docker Compose
- Statistics and Metrics Collection
  - Internal Logs from the Trainer, Scorer and Aggregator classes
  - `docker stats` for RAM, CPU and Network Traffic

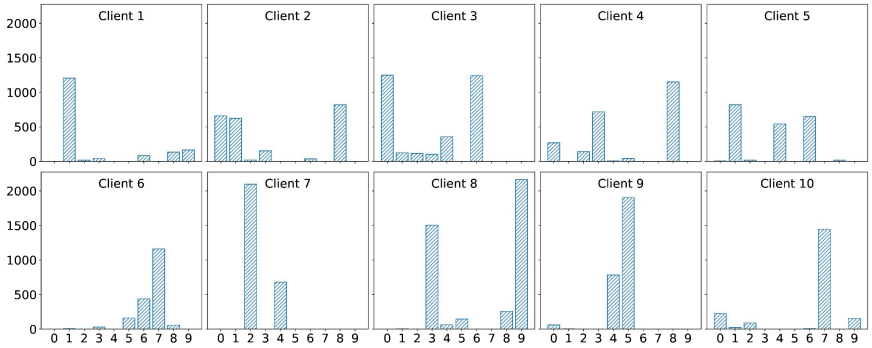


1. Docker, https://www.docker.com
2. Go Ethereum, https://geth.ethereum.org/
3. Quorum, https://consensys.net/quorum/

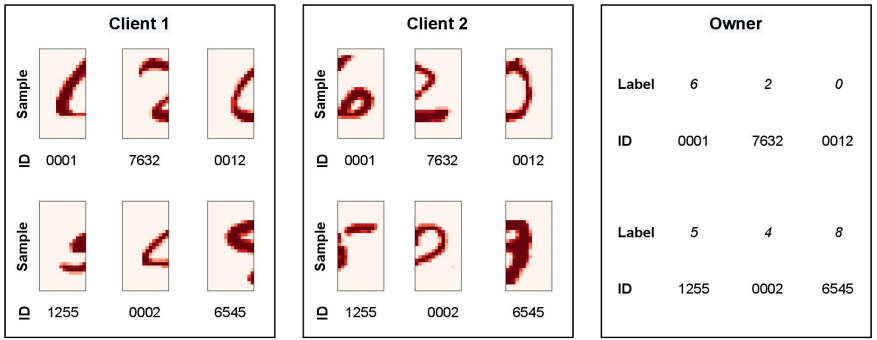
Experimental Setup & Evaluation

Dataset & Client Sampling

- **Dataset:** MNIST[1].
- **Horizontal Client Sampling:** Dirichlet distribution *Dir(α)* to simulate a *non-iid* distribution[2] with α = 0.1.
- **Vertical Client Sampling:** Different features = different parts of the image[3].



Horizontal Data Distribution For 10 Clients

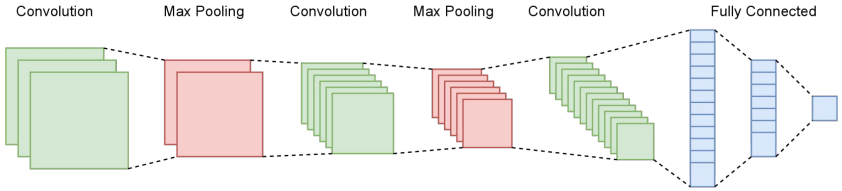


Vertical Data Distribution for 2 Clients

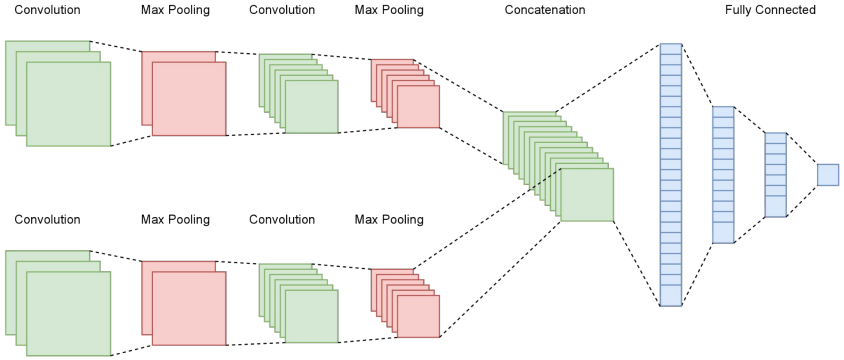
1. LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist 2 (2010).
2. Lin, T., Kong, L., Stich, S. U., and Jaggi, M. Ensemble distillation for robust model fusion in federated learning. In Advances in Neural Information Processing Systems (2020), H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., pp. 2351–2363.
3. Romanini, D., Hall, A. J., Papadopoulos, P., Titcombe, T., Ismail, A., Cebere, T., Sandmann, R., Roehm, R., and Hoeh, M. A. Pyvertical: A vertical federated learning framework for multi-headed splitnn, 2021.

Experimental Setup and Evaluation

ML Models



Horizontal CNN



Dual-headed Split-CNN

Metrics & Experimental Groups

Metrics

- Execution Time
  - End-to-End (E2E) Execution Time (*m*), Mean Round Execution Time (*s*)
- Blockchain Costs
  - Transaction Latency (*s*), Transaction Cost (*gas*)
- Model Performance
  - Accuracy (*%*)
- Communication Costs*
  - Network Traffic Per Round (*MB*)
- Computation Costs*
  - RAM Usage (*MB*), CPU Usage (*%*)

* On client, server and blockchain processes. Note that blockchain and server usually run on same machine.

Experiment Groups

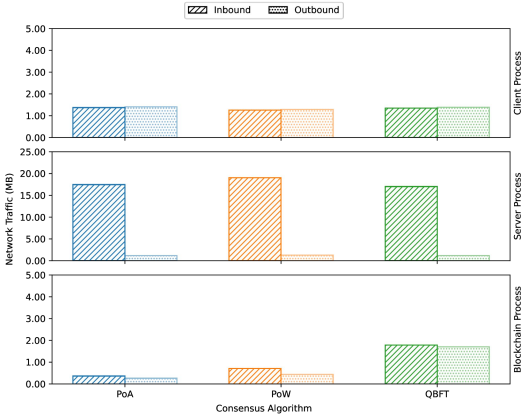
- Consensus Algorithms: PoA, PoW, QBFT.
- Horizontal FL
  - Participant Selection Mechanisms: random and first-come first-served.
  - Scoring Algorithms: BlockFlow, Multi-KRUM, Marginal Gain, none.
    - Number of Clients: 5, 10, 25, 50.
    - Privacy Degree: 1, 5, none.
- Vertical FL
  - Implementation of Vertical Blockchain-based Federated Learning
  - Extension of BlockLearning to support the Split-CNN

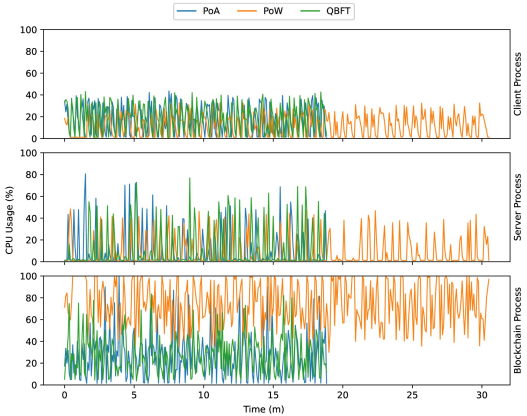
Results

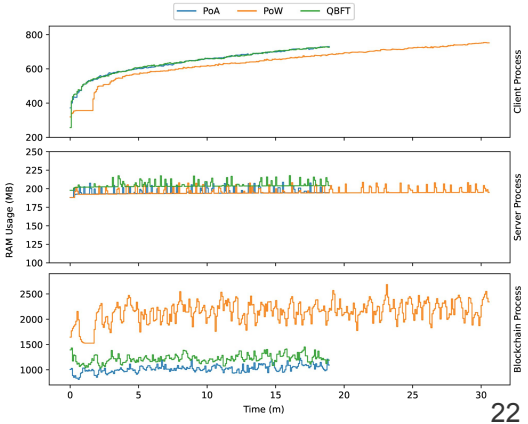
Impact Analysis of Consensus Algorithms

- PoW performed the worst, compared to PoA and QBFT.
  - 1.6x times slower.
  - 1.2x higher Tx Latency.
  - 1.2x higher Tx Costs.
  - Much higher RAM and CPU.
- Accuracy is unchanged (Fig 6.1, p. 45).
- QBFT incurs 2x more communication costs than PoW, 4x more than PoA.
- In general, PoA is the most cost-efficient.

	PoA	PoW	QBFT
E2E Time (m)	18.93	30.62	18.97
Mean Round Time (s)	22.70	36.72	22.74
Mean Transaction Latency (s)	1.549	1.821	1.558
Mean Transaction Cost (Gas)	183124	227052	18288

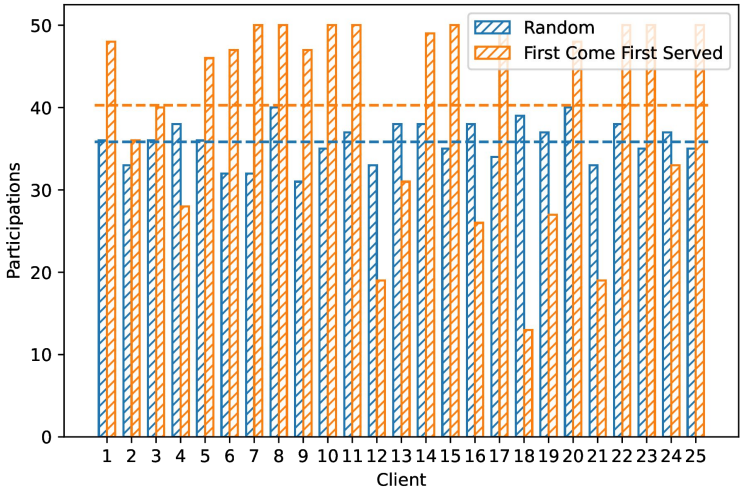


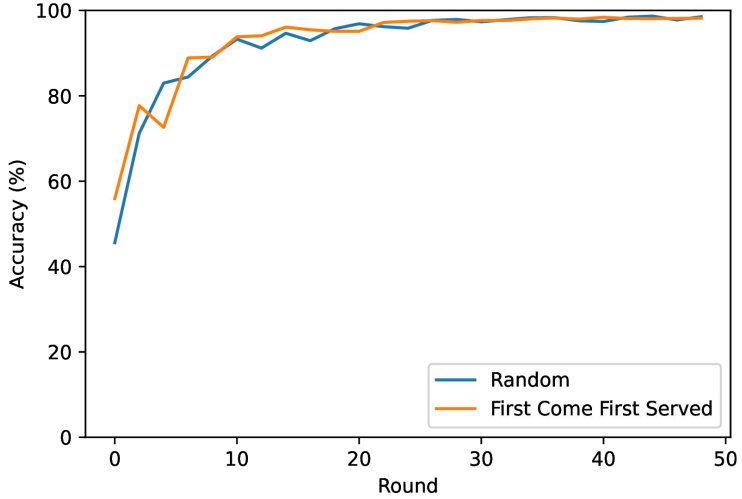




Impact Analysis of Participant Selection Algorithms

- When clients take initiative, some may participate more than others. May lead to skewed results.
- Random (uniform) selection gives all clients an equal chance of participating.





- Round Time, Tx Latency and Costs, Communication and Computation costs are similar between algorithms.

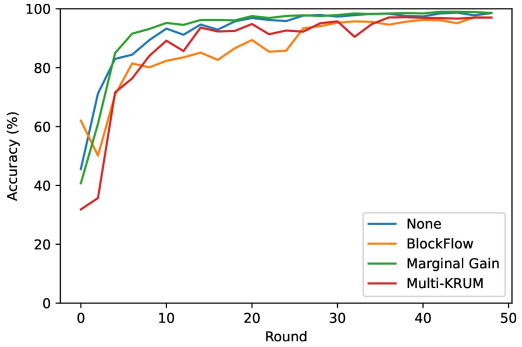
(Remaining plots: pp. 44-45)

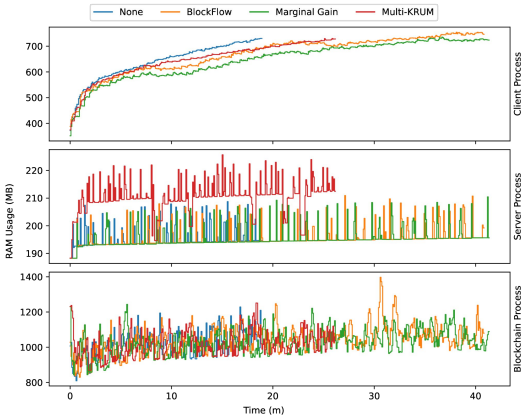
Impact Analysis of Scoring Algorithms

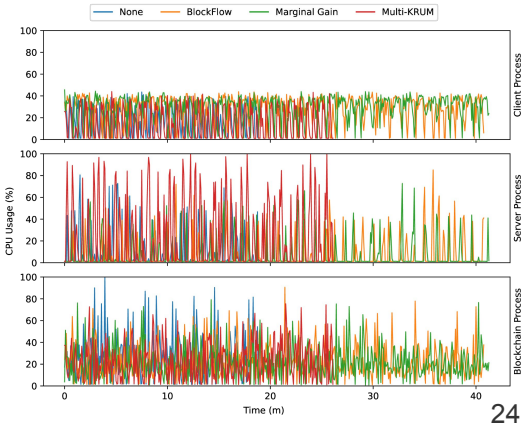
- Accuracies are high, but convergence and execution time differ:
  - Marginal Gain converges the fastest, higher accuracy than without scores.
  - BlockFlow converges the slowest.
  - Multi-KRUM converges in between and is the fastest to execute.
- Overall, scoring algorithms executed by clients have higher impact on the system.

(Remaining plots: pp. 45-50)

Metric	None	BlockFlow	Marginal Gain	Multi--KRUM
E2E Time (m)	18.93	40.95	41.38	26.25
Mean Round Time (s)	22.70	49.11	49.64	31.48
Mean Transaction Latency (s)	1.549	1.564	1.577	1.573
Mean Transaction Cost (Gas)	183124	339645	257686	280733





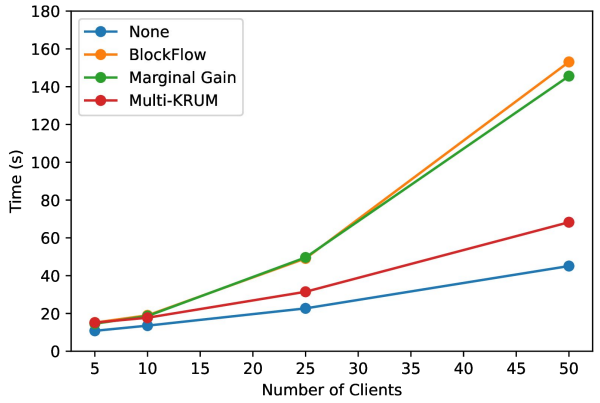


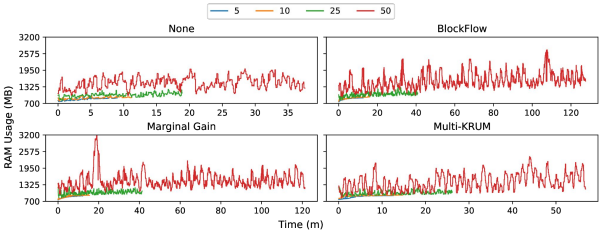
Results in Scoring Algorithms in Horizontal Federated Learning

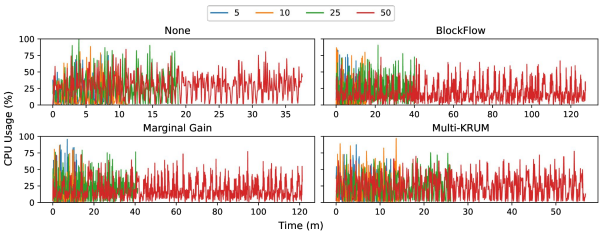
Impact Analysis of Number of Clients

- Execution time increase is higher for scoring algorithms executed by clients.
- In addition, #clients ≫ #servers.
- Trade-off between the number of clients with the blockchain resource usage.

(Remaining plots: pp. 50-54)







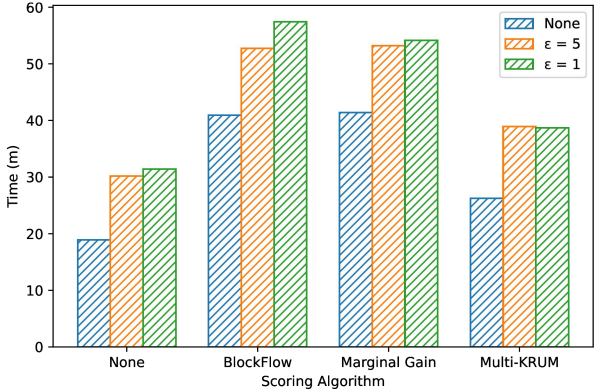
Blockchain Process

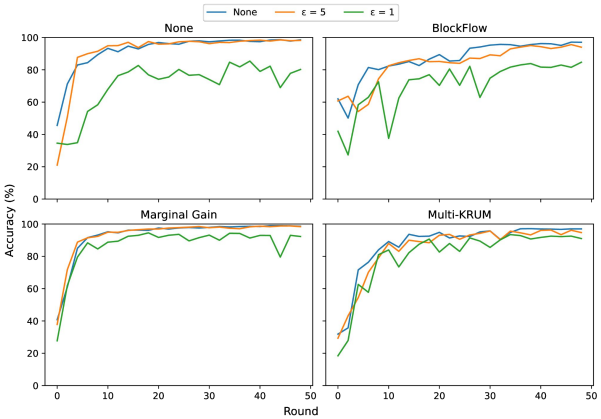
Results in Scoring Algorithms in Horizontal Federated Learning

Impact Analysis of Privacy Degrees

- Trade-off between execution time and privacy.
- Increasing the privacy degree does not have influence execution time or resource usage.
- Overall, higher privacy degrees lead to lower accuracy levels.
- Trade-off between traceability and auditability and the requirement for privacy mechanisms.

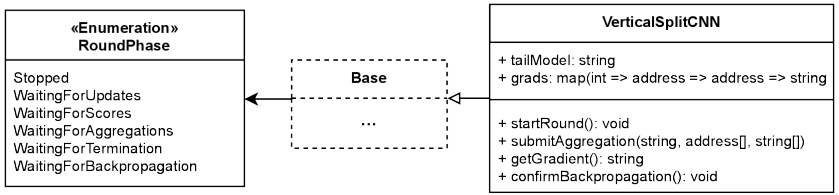
(Remaining plots: pp. 54-62)





Extending the Framework

- New requirements:
  - Support different models for the trainers and the aggregators
  - Support additional backpropagation confirmation phase
- Implementation:
  - New phase `WaitingForBackpropagation`
  - New contract `VerticalSplitCNN`
  - `TrainerSplitCNN`, `AggregatorSplitCNN`



Split-CNN Smart Contracts Extension Class Diagram

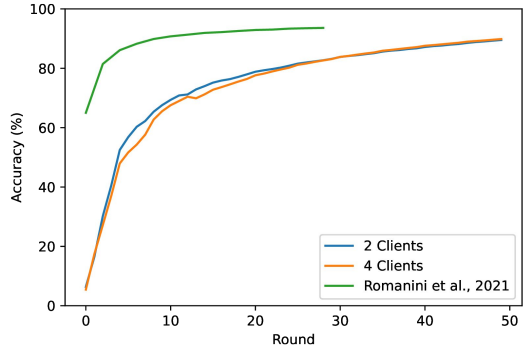
BlockLearning's Vertical Execution Flow

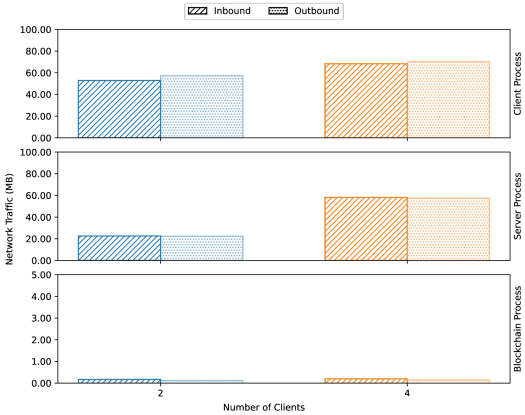
Experiment and Results

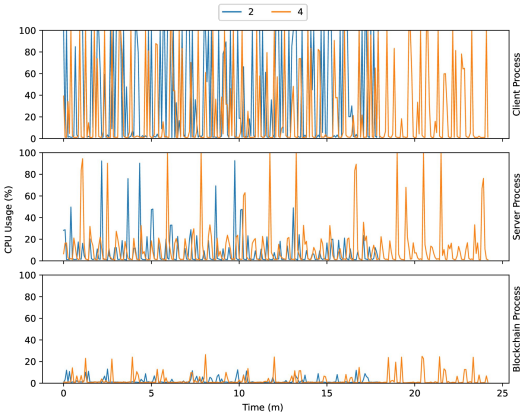
- Executed with 2 and 4 clients, the same way as the HBFL.
- Experimental results are within the expected values.
- Original accuracy is higher. May be related to:
  - Different implementation
  - Different ML library used
  - Other unknown variables in the system

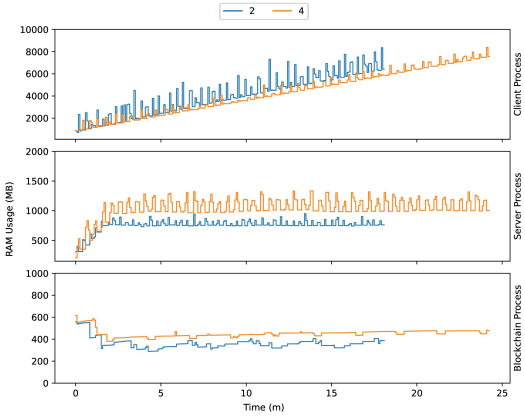
(Remaining plots: pp. 70-75)

Metric	2	4
E2E Time (m)	18.08	24.30
Mean Round Time (s)	21.68	29.15
Mean Transaction Latency (s)	1.482	1.418
Mean Transaction Cost (Gas)	138659	141013









Conclusions

Looking Back At The Main Research Question

What is the impact of different consensus, participant selection and scoring algorithms in a Blockchain-based Federated Learning system on execution time, convergence and accuracy, as well as communication and computation costs?

Looking Back At The Main Research Question

- **Blockchain**
  - Overall execution time and costs increase.
- **Consensus Algorithms**
  - PoW: highest computation costs, slowest.
  - QBFT: highest communication costs, fast.
  - PoA: lower overall costs, fastest.
  - PoA is the most cost-efficient consensus algorithm analyzed
- **Participant Selection Algorithms**
  - Both have similar costs.
  - Random selection is fairer.
- **Scoring Algorithms**
  - Increase the execution time up to 2x.
  - Marginal Gain: highest accuracy with high number of clients and privacy degree. But highest computation costs for the clients.
  - Multi-KRUM: lowest computation costs for clients, high accuracy (better for IoT).
  - BlockFlow: worst in all aspects.
- **Vertical Federated Learning**
  - Possible to implement in a BFL setting.
  - First known implementation of VBFL.
  - BlockLearning is flexible and modular.

Contributions

1. Designed and implemented the first open-source and modular framework for BFL. Available at https://github.com/hacdias/blocklearning.
2. Provided the first comparative study on how different consensus, participant selection and scoring algorithms impact a BFL system*.
3. Provided the first comparative study on how the number of clients and different degrees of privacy for different scoring algorithms impact a BFL system*.
4. Implemented the first Vertical Blockchain-based Federated Learning framework.

* in terms of execution time, transaction costs, transaction latency, model accuracy and convergence, communication costs, and computation costs.

Future Work

- **Consensus Algorithms:** investigate if it is feasible to extend the Ethereum blockchain with custom resource-efficient algorithms presented by others.
- **Scoring Algorithms**: investigate and develop new scoring algorithms that do not require model evaluation at the clients side, reducing resources usage.
- **Blockchain-based VFL**: make VFL BlockLearning extension more generic in order to support other vertical models and the Private Set Intersection phase.
- **BlockLearning GUI**: develop a graphical interface for BlockFlow to allow submission of training requests, visualization of the training process, and download the weights without needing a command-line.

Thank you!

TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY