

is affected by the slowest model update that arrives at the central server.

Solution: The global model aggregation is conducted whenever a model update is received, without being delayed by other clients. Then the server starts the next iteration and distributes the new central model to the clients that are ready for training. The delayed model updates that are not included in the current aggregation round will be added in the next round with some reduction in the weightage, proportioned to their respective delayed time.

Consequences:

Benefits:

- *Low aggregation latency.* Faster aggregation time per round is achievable as there is no need to wait for the model updates from other clients for the aggregation round. The bandwidth usage per iteration is reduced as fewer local model updates are transferred and receive simultaneously every round.

Drawbacks:

- *Communication cost.* The number of iteration to collect all local model updates increases for the asynchronous approach. More iterations are required for the entire training process to train the model till convergence compares to synchronous global model aggregation.
- *Model bias.* The global model of each round does not contain all the features and information of every local model update. Hence the global model might have a certain level of bias in prediction.

Related patterns: *Client Registry, Client Selector, Model Co-versioning Registry, Client Update Scheduler*

Known uses:

- Asynchronous Online Federated Learning (*ASO-fed*) [11] is a framework for federated learning that adopted asynchronous aggregation. The central server update the global model whenever it receives a local update from one client device (or several client devices if the local updates are received simultaneously). On the client device side, online-learning is performed as data continue to arrive during the global iterations.
- Asynchronous federated SGD-Vertical Partitioned (*AFSGD-VP*) [14] algorithm uses a tree-structured communication scheme to perform asynchronous aggregation. The algorithm does not need to align the iteration number of the model aggregation from different client devices to compute the global model.
- Asynchronous Federated Optimization (*FedAsync*) [43] is an approach that leverages asynchronous updating technique and avoids server-side timeouts and abandoned rounds while requires no synchronous model broadcast to all the selected client devices.

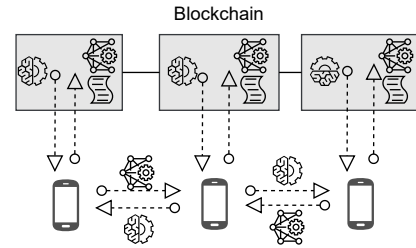


Figure 15: Decentralised Aggregator.

3.4.2. Pattern 12: Decentralised Aggregator

Summary: A decentralised aggregator improves system reliability and accountability by removing the central server that is a possible single-point-of-failure. Fig. 15 illustrates the decentralised federated learning system built using blockchain and smart contract, while the model updates are performed through the exchange between neighbour devices.

Context: The model training and aggregation are coordinated by a central server and both the central server and the owner may not be trusted by all the client devices that join the training process.

Problem: In *FedAvg*, all the chosen devices have to submit the model parameters to one central server every round. This is extremely burdensome to the central server and network congestion may occur. Furthermore, centralised federated learning possesses a single-point-of-failure. Data privacy threats may also occur if the central server is compromised by any unauthorised entity. The mutual trust between the client devices and the central server may not be specifically established.

Forces: The problem requires to balance the following forces:

- *Decentralised model management.* The federated learning systems face challenges to collect, store, examine, and aggregate the local models due to the removal of the central server.
- *System ownership.* Currently, the central server is own by the learning coordinator that creates the federated learning jobs. The removal of the central server requires the re-definition of system ownership. It includes the authority and accessibility of learning coordinator in the federated learning systems.

Solution: A decentralised aggregator replaces the central server's role in a federated learning system. The aggregation and update of the models can be performed through peer-to-peer exchanges between client devices. First, a random client from the system can be an aggregator by requesting the model updates from the other clients that are close to it. Simultaneously, the client devices conduct local model training in parallel and send the trained local models to the aggregator. The aggregator then produces a new global model and sends it to the client network. Blockchain is the alternative to the central server for model storage that prevents single-