point-of-failure. The ownership of the blockchain belongs to the learning coordinator that creates the new training tasks and maintains the blockchain. Furthermore, the record of models on a blockchain is immutable that increases the reliability of the system. It also increases the trust of the system as the record is transparent and accessible by all the client devices.

Consequences:

Benefits:

- *System reliability.* The removal of single-point-of-failure increases the system reliability by reducing the security risk of the central server from any adversarial attack or the failure of the entire training process due to the malfunction of the central server.
- *System accountability.* The adoption of blockchain promotes accountability as the records on a blockchain is immutable and transparent to all the stakeholders.

Drawbacks:

- *Latency.* Client device as a replacement of the central server for model aggregation is not ideal for direct communication with multiple devices (star-topology). This may cause latency in the model aggregation process due to blockchain consensus protocols.
- *Computation cost.* Client devices have limited computation power and resource to perform model training and aggregation parallel. Even if the training process and the aggregation is performed sequentially, the energy consumption to perform multiple rounds of aggregation is very high.
- *Storage cost.* High storage cost is required to store all the local and global models on storage-limited client devices or blockchain.
- *Data privacy.* Client devices can access the record of all the models under decentralised aggregation and blockchain settings. This might expose the privacy-sensitive information of the client devices to other parties.

Related patterns: *Model Co-versioning Registry, Incentive Registry*

Known uses:

- *BrainTorrent* [36] is a server-less, peer-to-peer approach to perform federated learning where clients communicate directly among themselves, specifically for federated learning in medical centers.
- *FedPGA* [20] is a decentralised aggregation algorithm developed from *FedAvg*. The devices in *FedPGA* exchange partial gradients rather than full model weights. The partial gradient exchange pulls and merges the different slice of the updates from different devices and rebuild a mixed update for aggregation.
- A fully decentralised framework [26] is an algorithm in which users update their beliefs by aggregate information from their one-hop neighbors to learn a model that best fits the observations over the entire network.
- A Segmented gossip approach [16] splits a model into segmentation that contains the same number of non-overlapping model parameters. Then, the gossip protocol is adopted where each client stochastically selects a few other clients to exchange the model segmentation for each training iteration without the orchestration of a central server.

3.4.3. Pattern 13: Hierarchical Aggregator

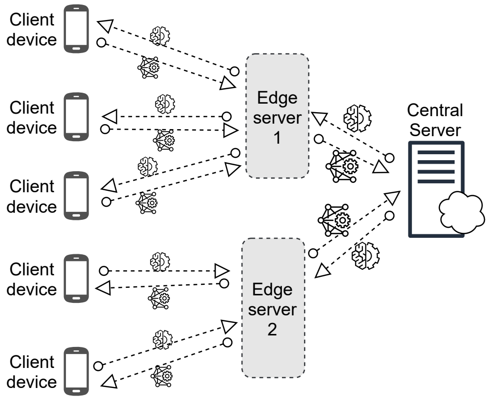

Figure 16: Hierarchical Aggregator.

Summary: To reduce non-IID effects on the global model and increase system efficiency, a hierarchical aggregator adds an intermediate layer (e.g., edge server) to perform partial aggregations using the local model parameters from closely-related client devices before the global aggregation. In Fig. 16, edge servers are added as an intermediate layer between the central server and client devices to serve the client devices that are closer to them.

Context: The communication between the central server and the client devices is slowed down or frequently disrupted due to being physically distant from each other and are wirelessly connected.

Problem: The central server can access and store more data but requires high communication overhead and suffers from latency due to being physically distant from the client devices. Moreover, client devices possess non-IID characteristics that affect global model performance.

Forces: The problem requires the following forces to be balanced:

- *System efficiency.* The system efficiency of the server-client setting to perform federated learning is low, as the central server is burdensome to accommodate the communication and the model aggregations of the widely-distributed client devices.
- *Data heterogeneity.* In the server-client setting of a federated learning system, the data heterogeneity character-