Trusted Federated Learning with Blockchain Technology

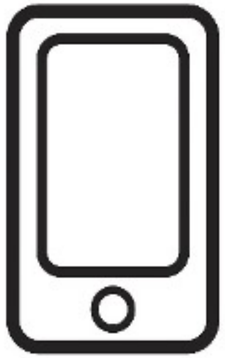
DS 2022 Project - Final Presentation

Giacomo Lombardo, André Gaillard

Introduction

Challenges in Centralized Machine Learning

Increasing amount of data



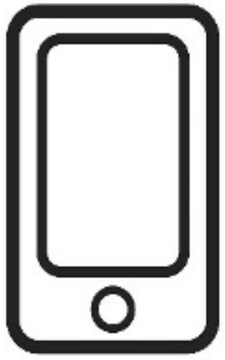
Availability of commodity devices

Privacy concerns

Legal concerns

Challenges in Centralized Machine Learning

Increasing amount of data



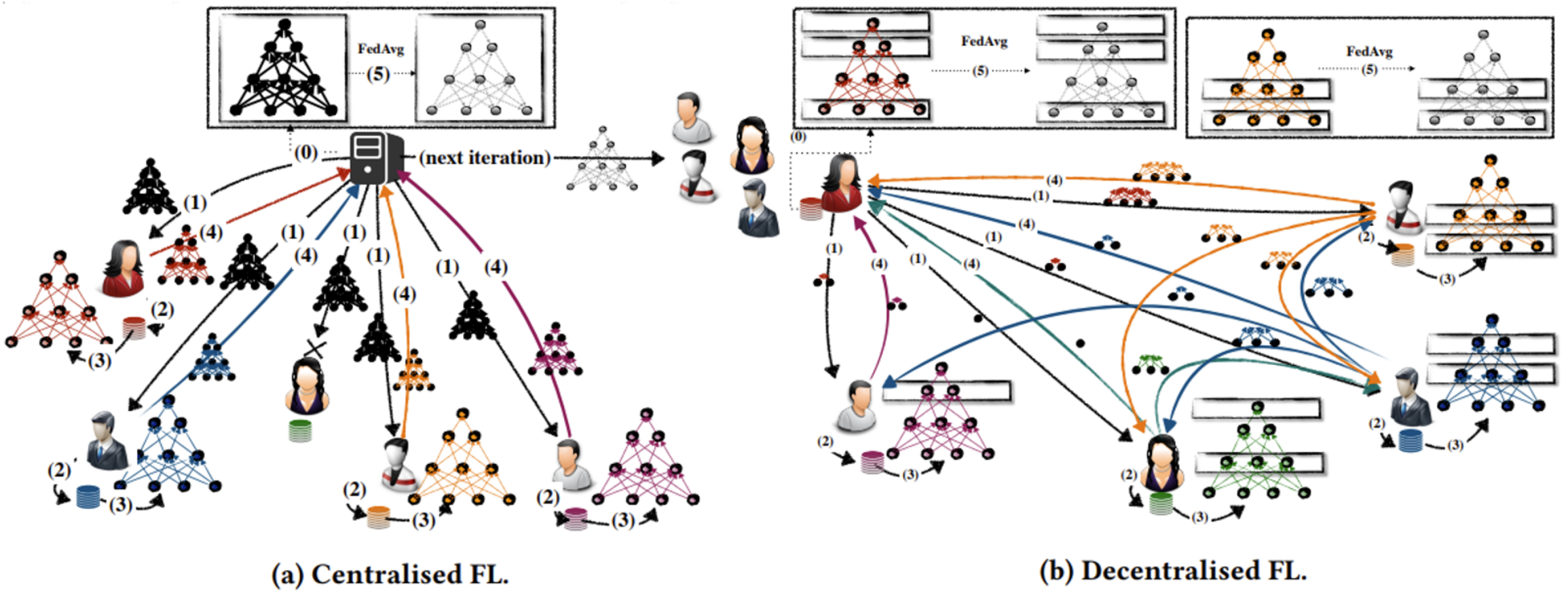
Availability of commodity devices

Privacy concerns

Legal concerns

Federated Learning emerges as a new paradigm

Federated Learning: Two Schools



Blockchain for Decentralised FL

- **Use Blockchain to coordinate the FL learning process.**
  - **Blockchain-based Federated Learning: A Comprehensive Survey**
    Zhiling Wang, Qin Hu - **arxiv.org/abs/2110.02182**
  - **BAFFLE : Blockchain Based Aggregator Free Federated Learning**
    Paritosh Ramanan, Kiyoshi Nakayama - **arxiv.org/abs/1909.07452**
  - **BlockFLow: An Accountable and Privacy-Preserving Solution for FL**
    Vaikkunth Mugunthan, Ravi Rahman, Lalana Kagal - **arxiv.org/abs/2007.03856**
  - **Mechanism Design for An Incentive-aware Blockchain-enabled Federated Learning Platform**
    Kentaroh Toyoda, Allan N. Zhang - **ieeexplore.ieee.org/document/9006344**

Decentralised FL: Challenges

- Ensure trust
  - What is trust?

Decentralised FL: Challenges

- Ensure trust
  - What is trust?
- Limitations of blockchain
  - Algorithmic and in storage size

Decentralised FL: Challenges

- Ensure trust
  - What is trust?
- Limitations of blockchain
  - Algorithmic and in storage size
- Incentivization system
  - Why should people participate (fairly!)

Decentralised FL: Challenges

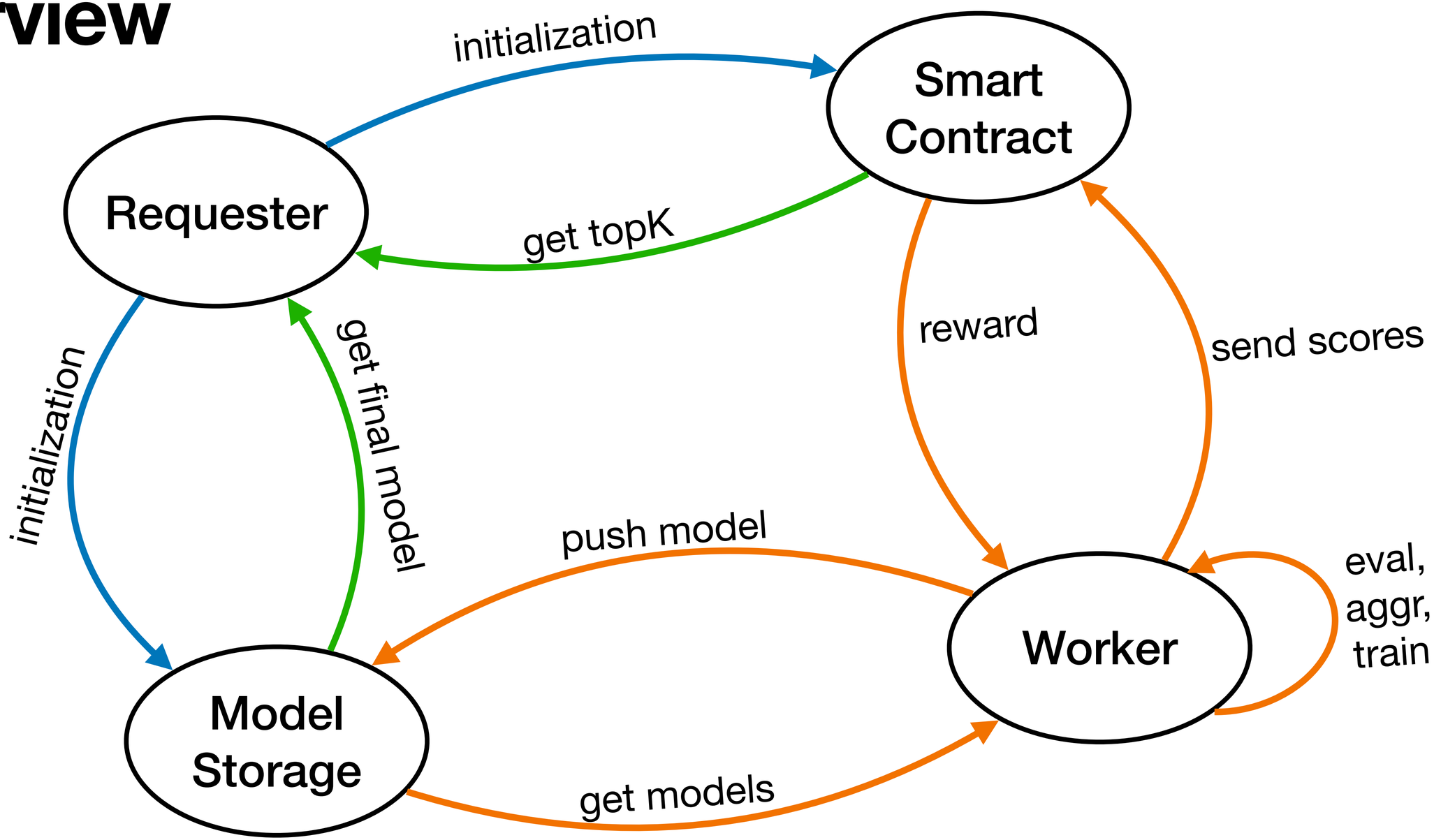
- Ensure trust
  - What is trust?
- Limitations of blockchain
  - Algorithmic and in storage size
- Incentivization system
  - Why should people participate (fairly!)
- Creation of a final global model
  - To be used by anybody

DiscoFL

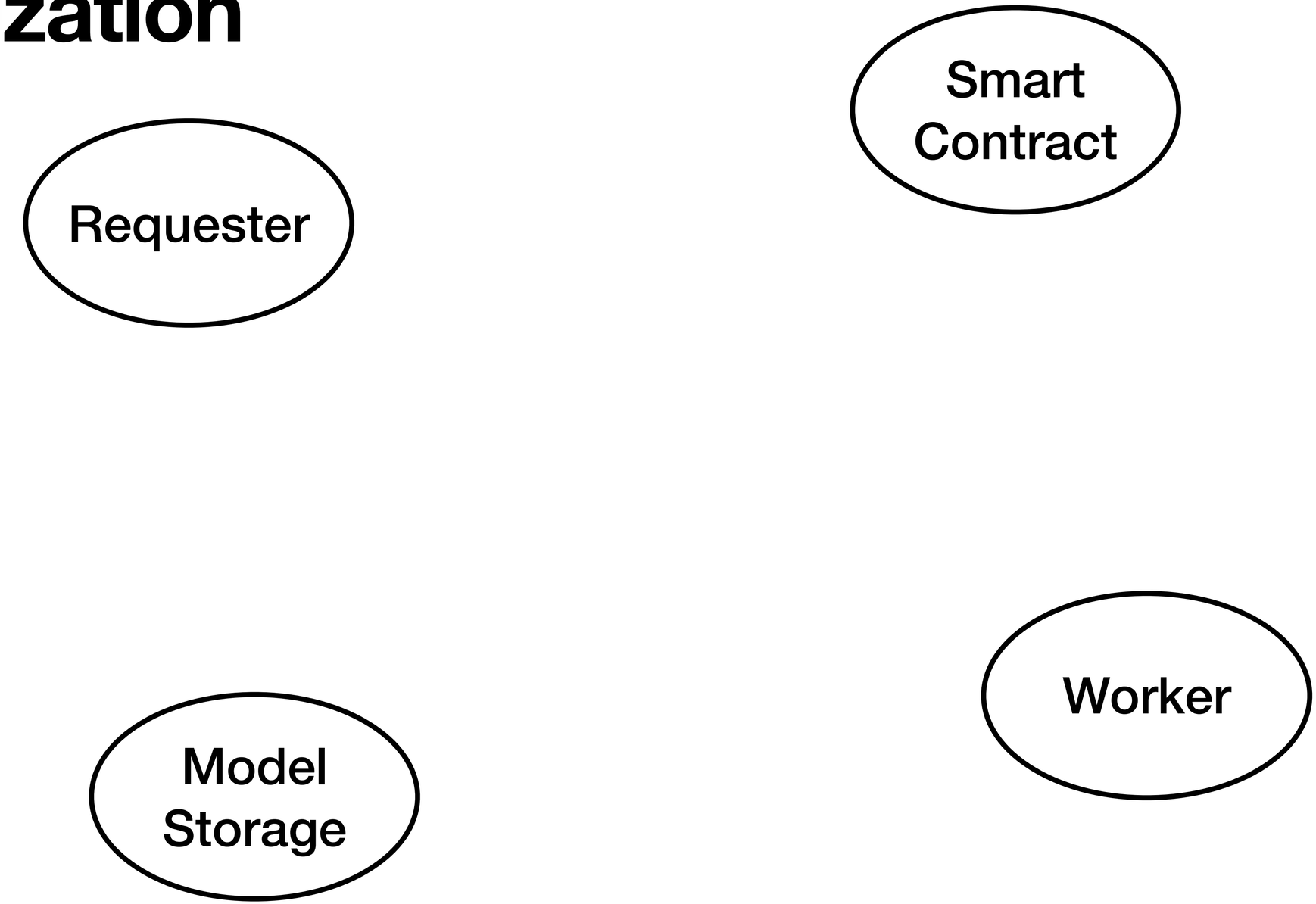
Distributed Incentive System for Cooperatively Orchestrated FL

Architecture

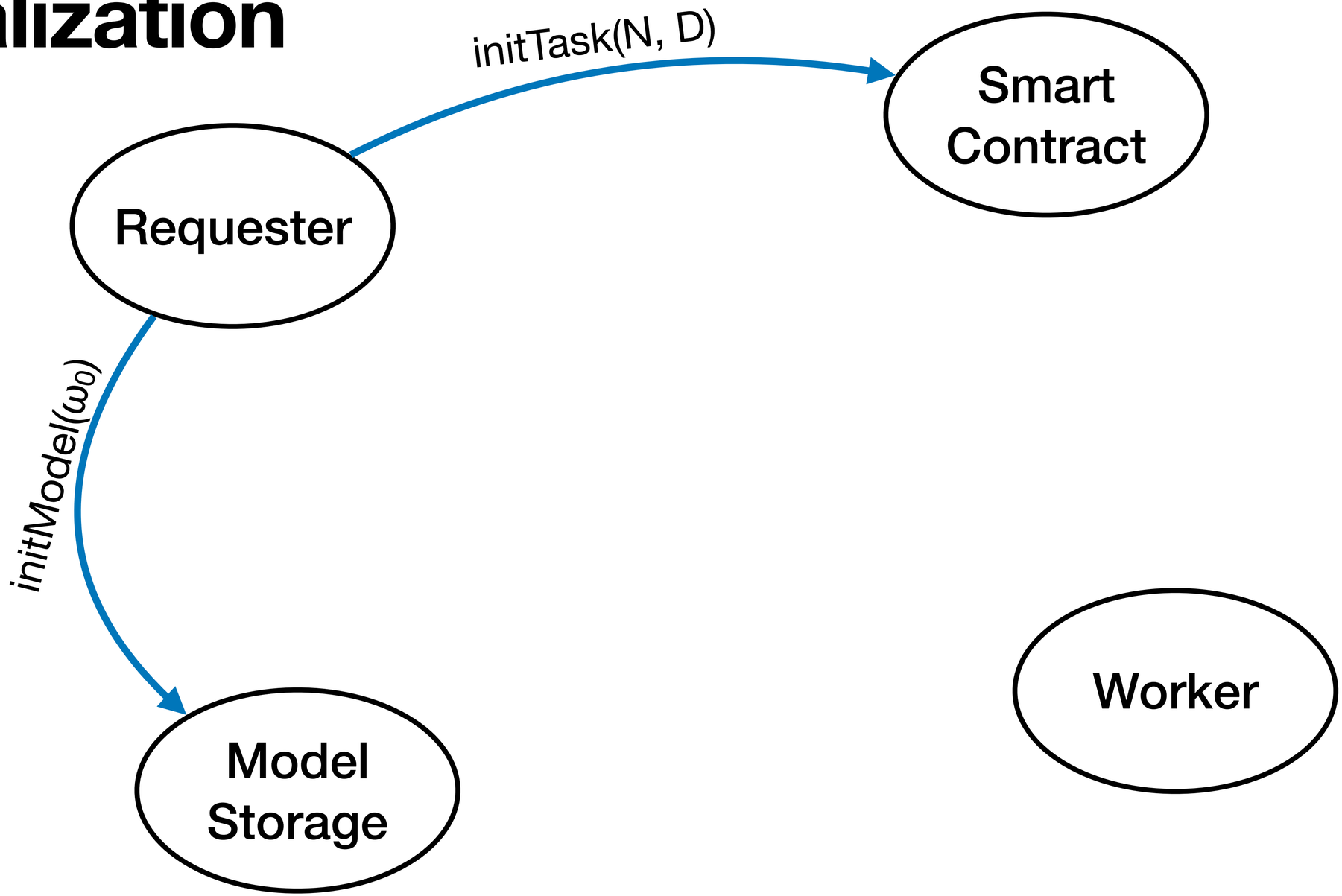
Overview


initialization
Smart Contract
Requester
get topK
reward
send scores
initialization
get final model
push model
eval, aggr, train
Worker
Model Storage
get models

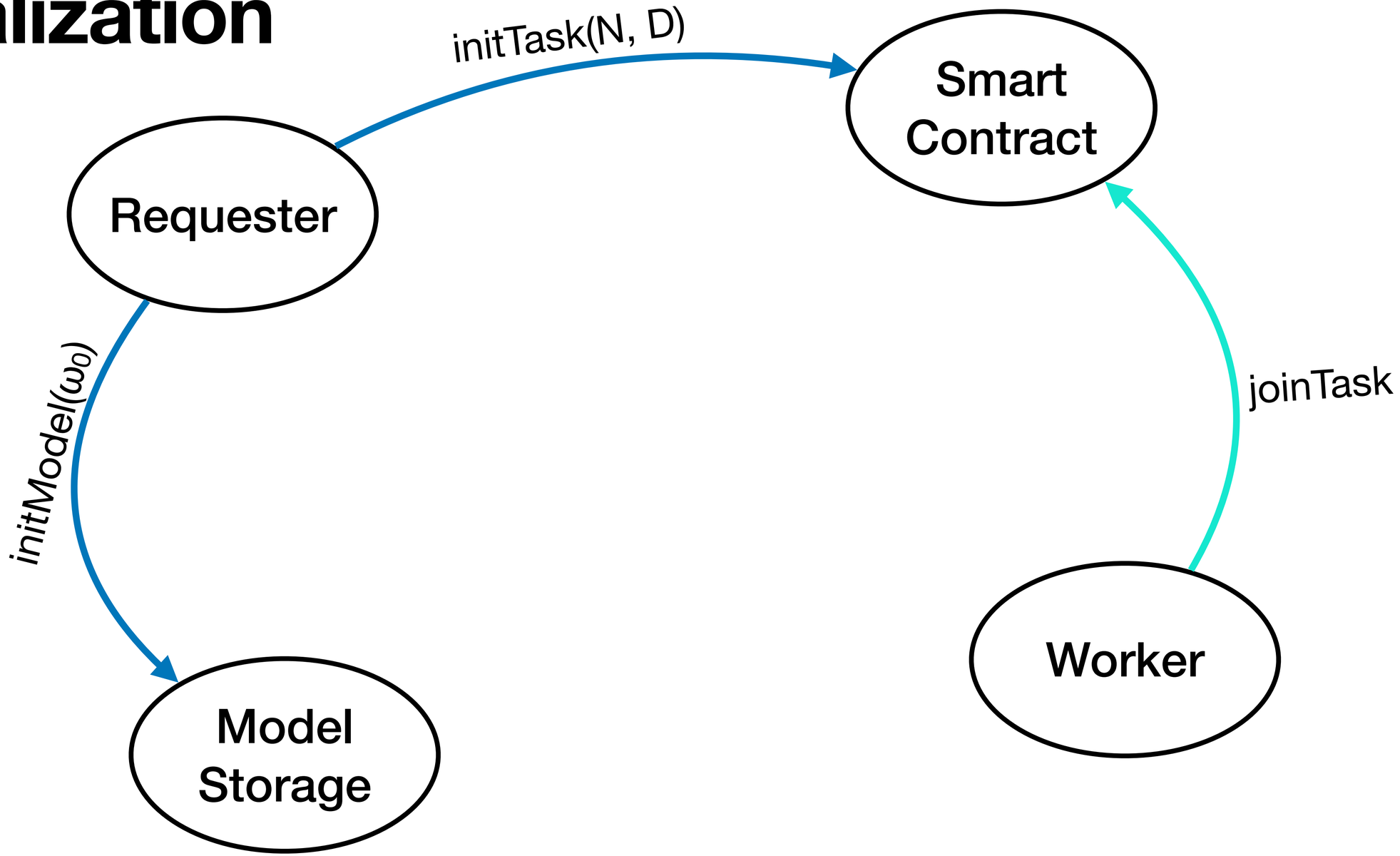
Initialization



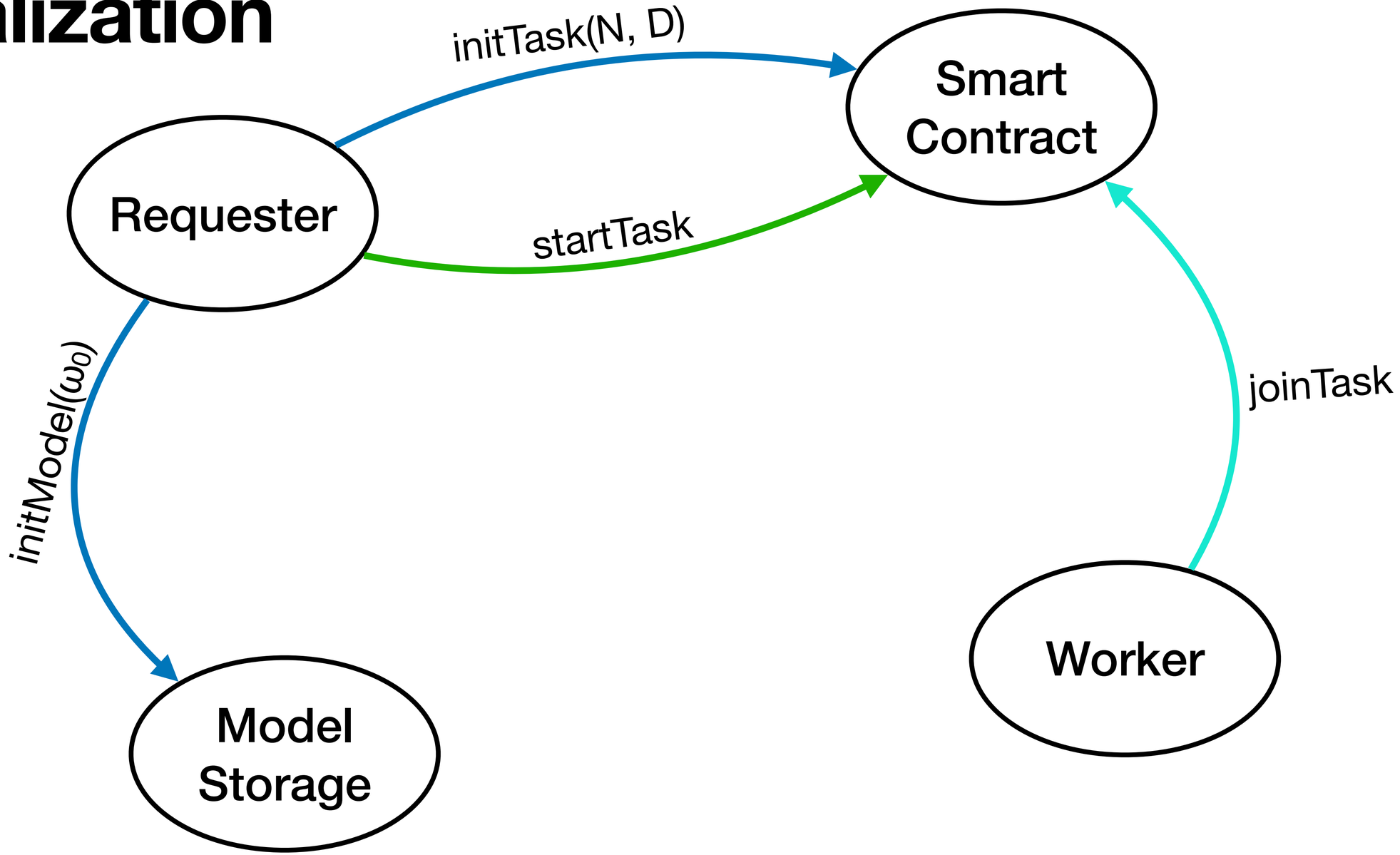
Initialization



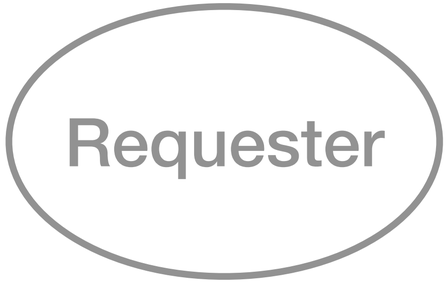
Initialization

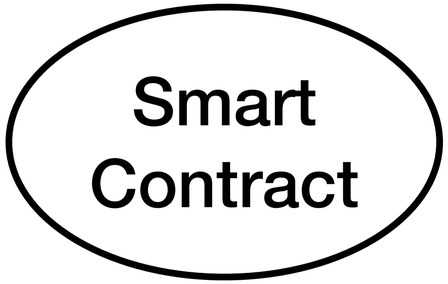


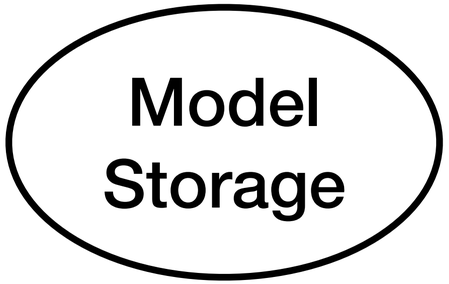
Initialization

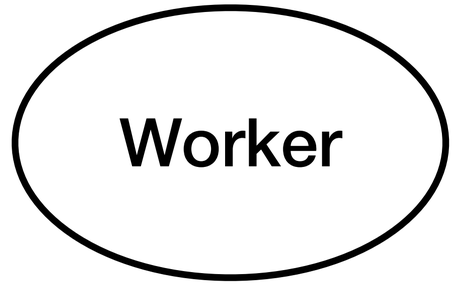


Training

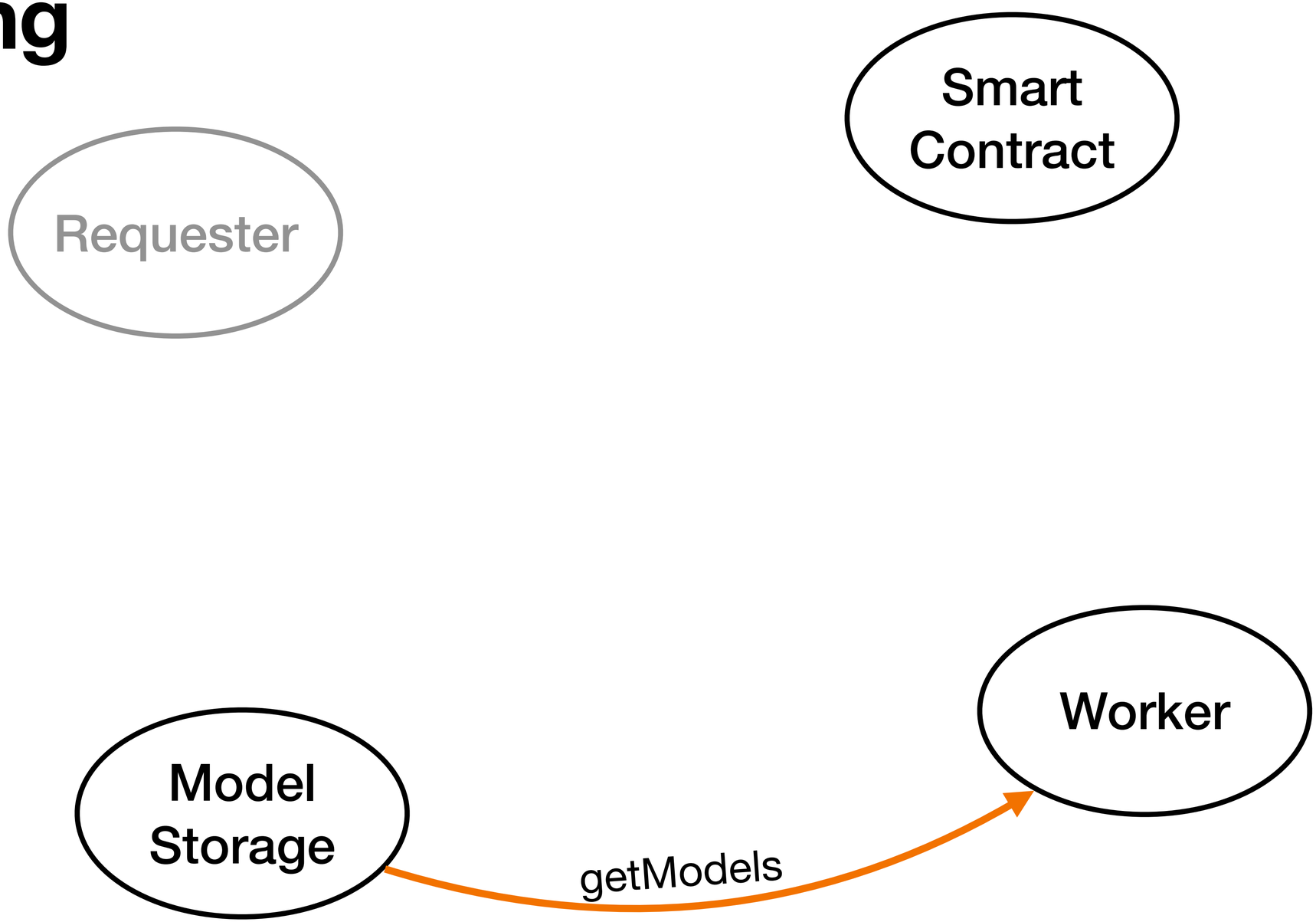




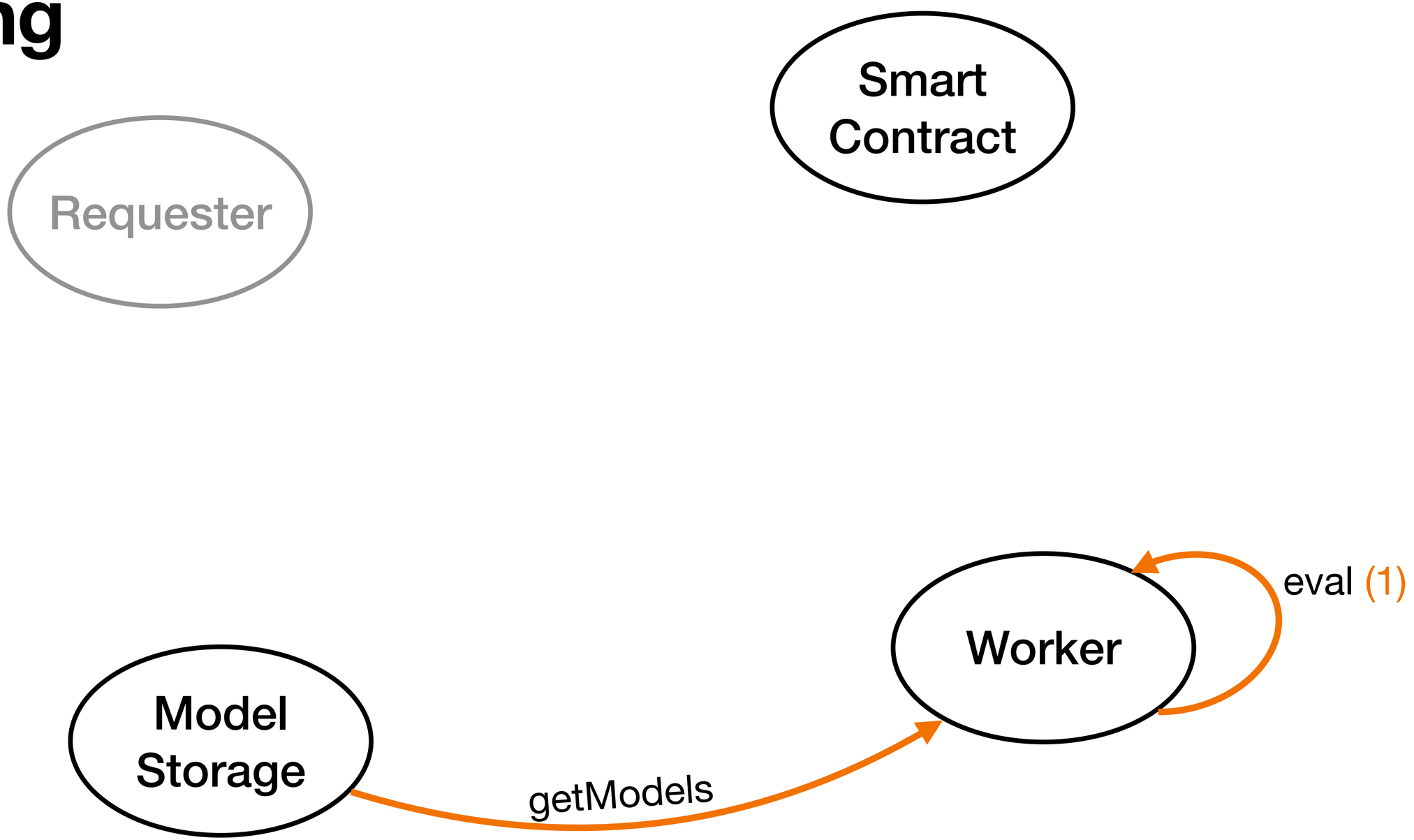




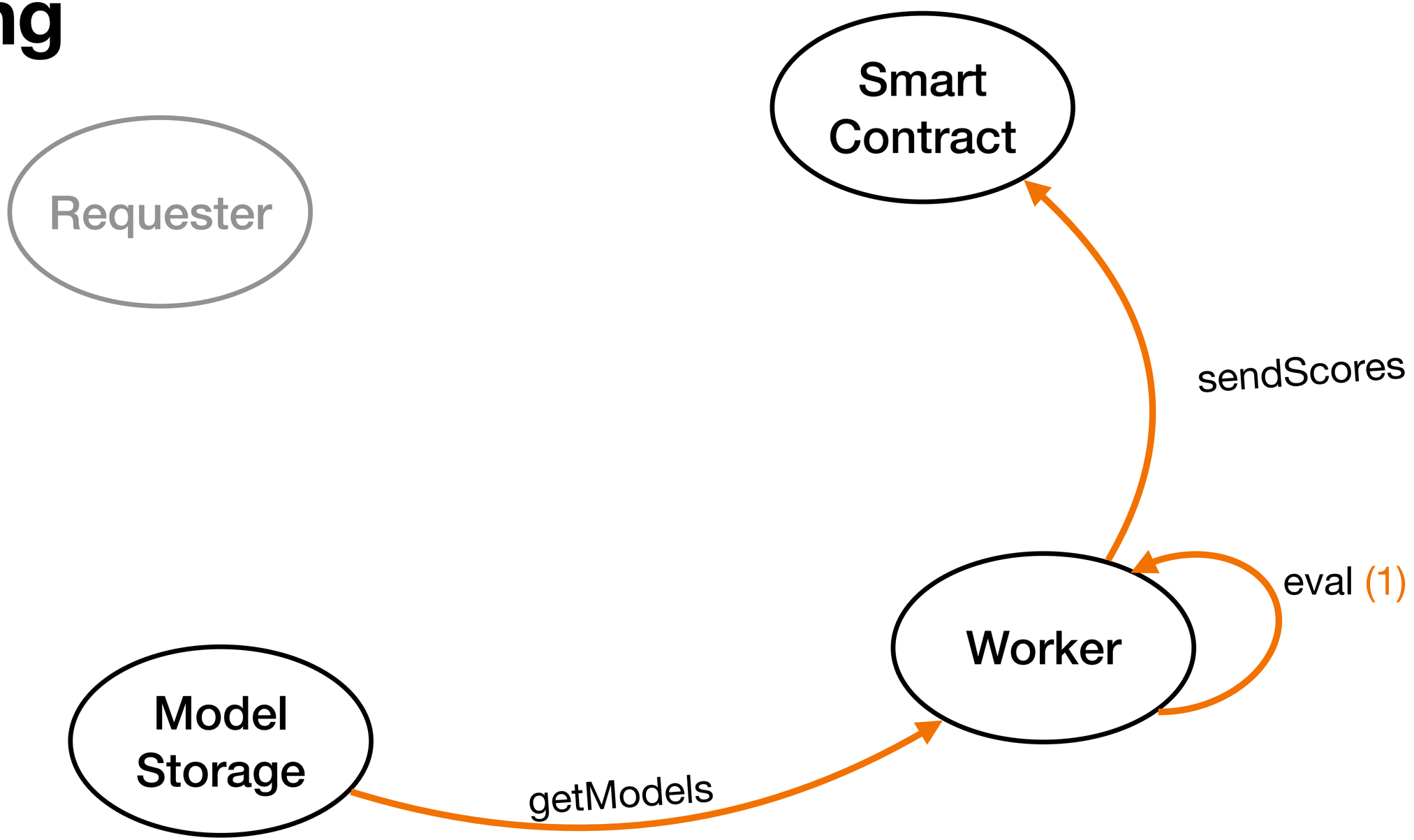
Training



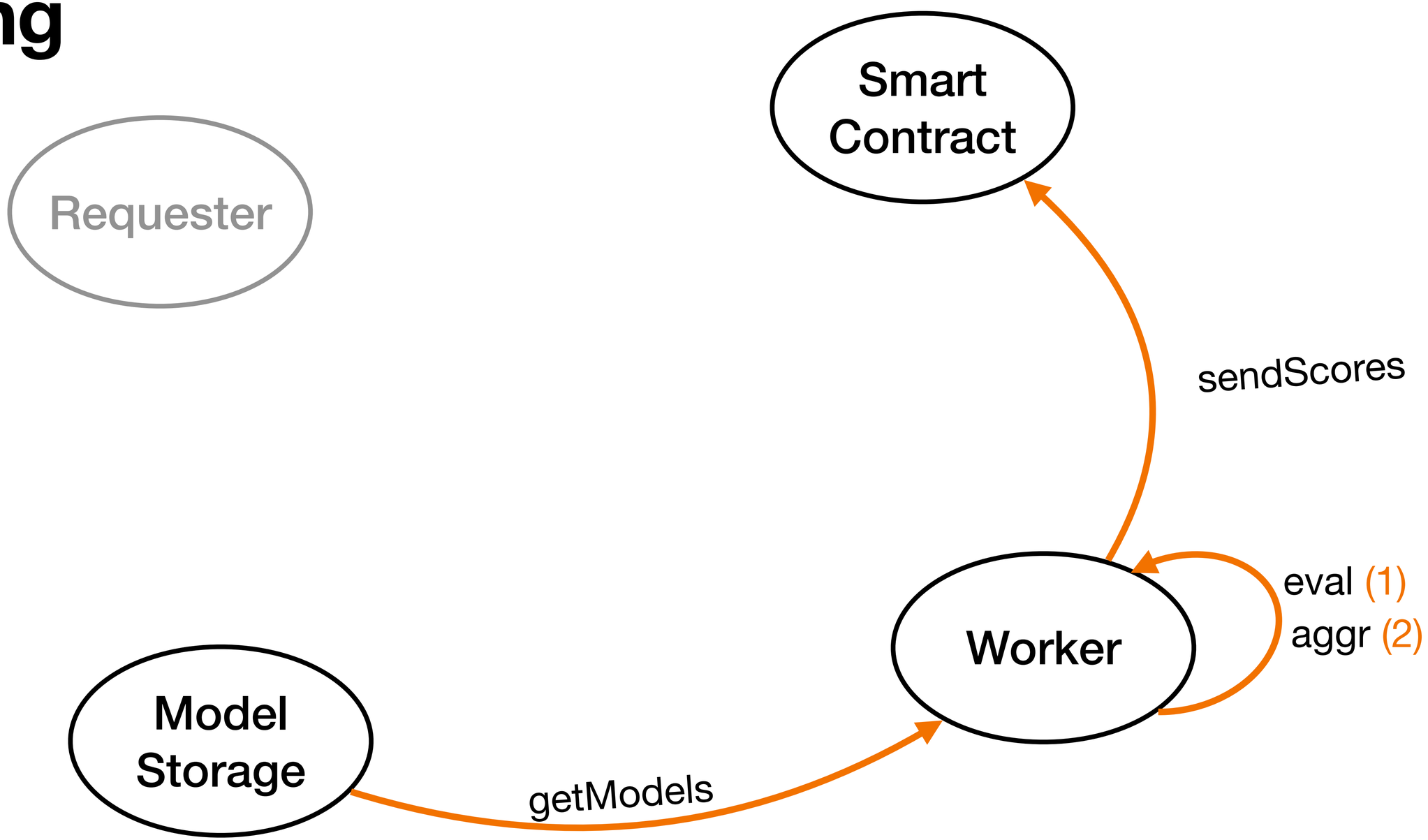
Training


Smart Contract
Requester
Worker
eval (1)
Model Storage
getModels

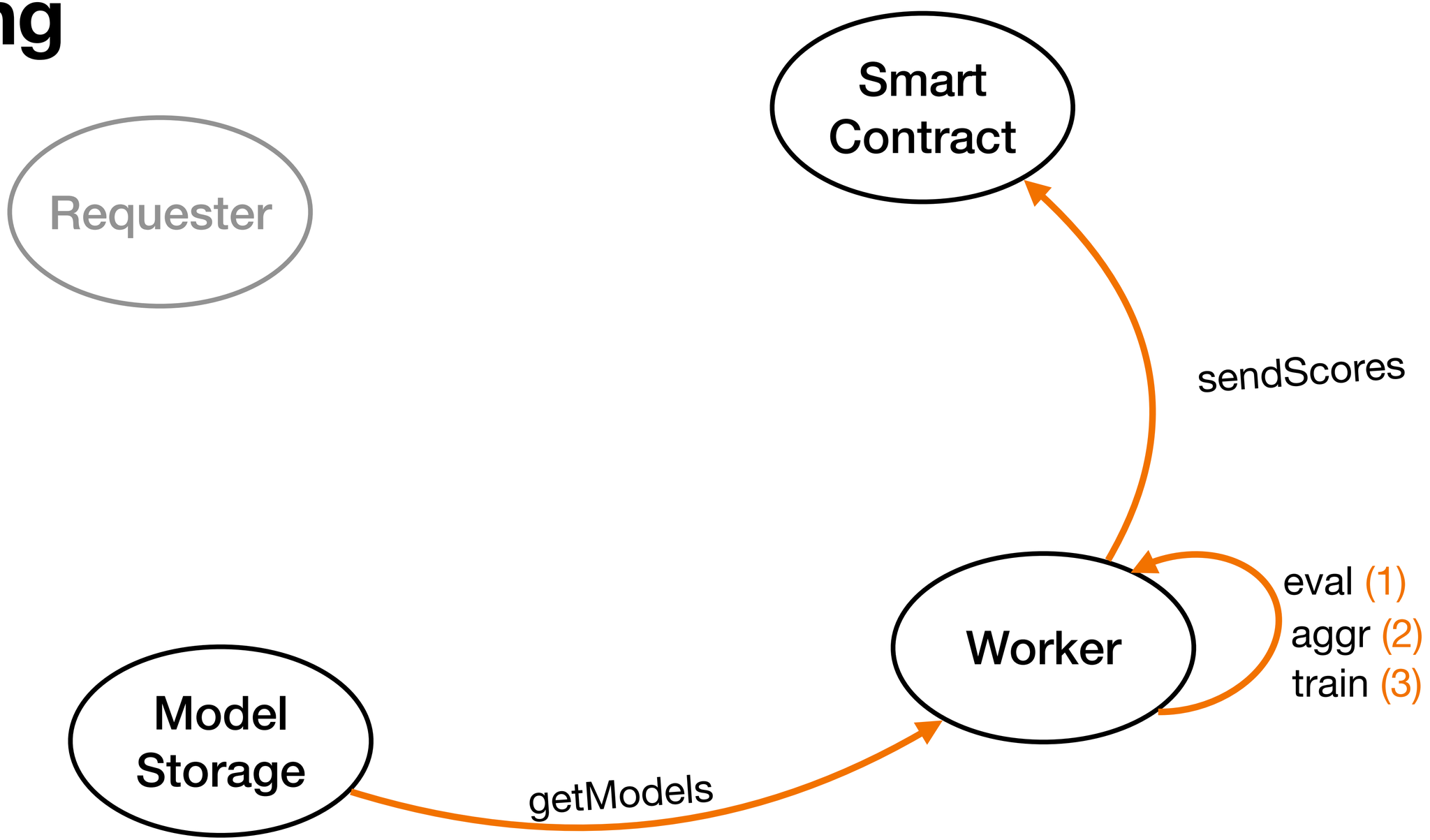
Training


Requester
Smart Contract
sendScores
eval (1)
Worker
Model Storage
getModels

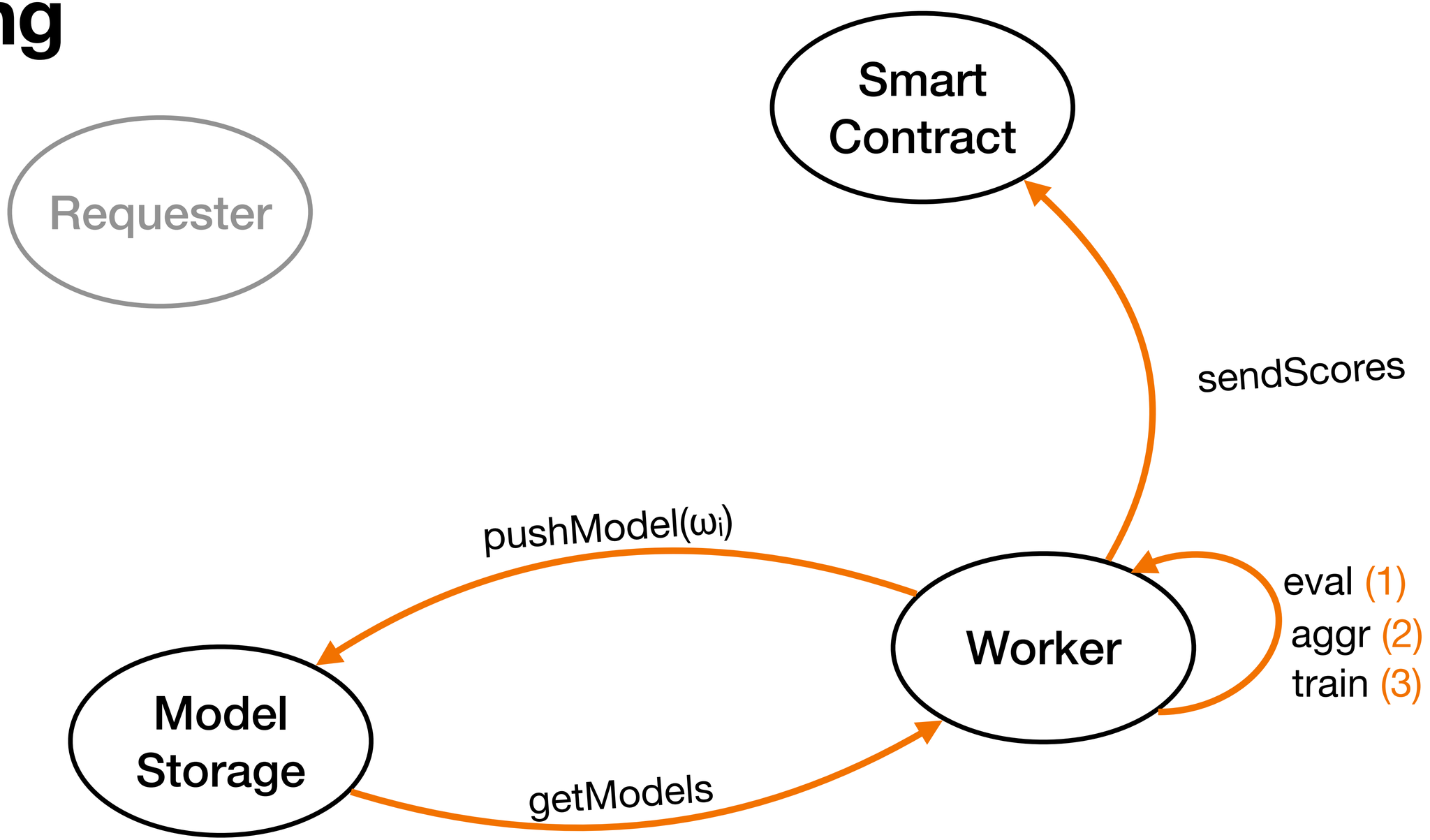
Training


Requester
Smart Contract
sendScores
eval (1)
aggr (2)
Worker
Model Storage
getModels

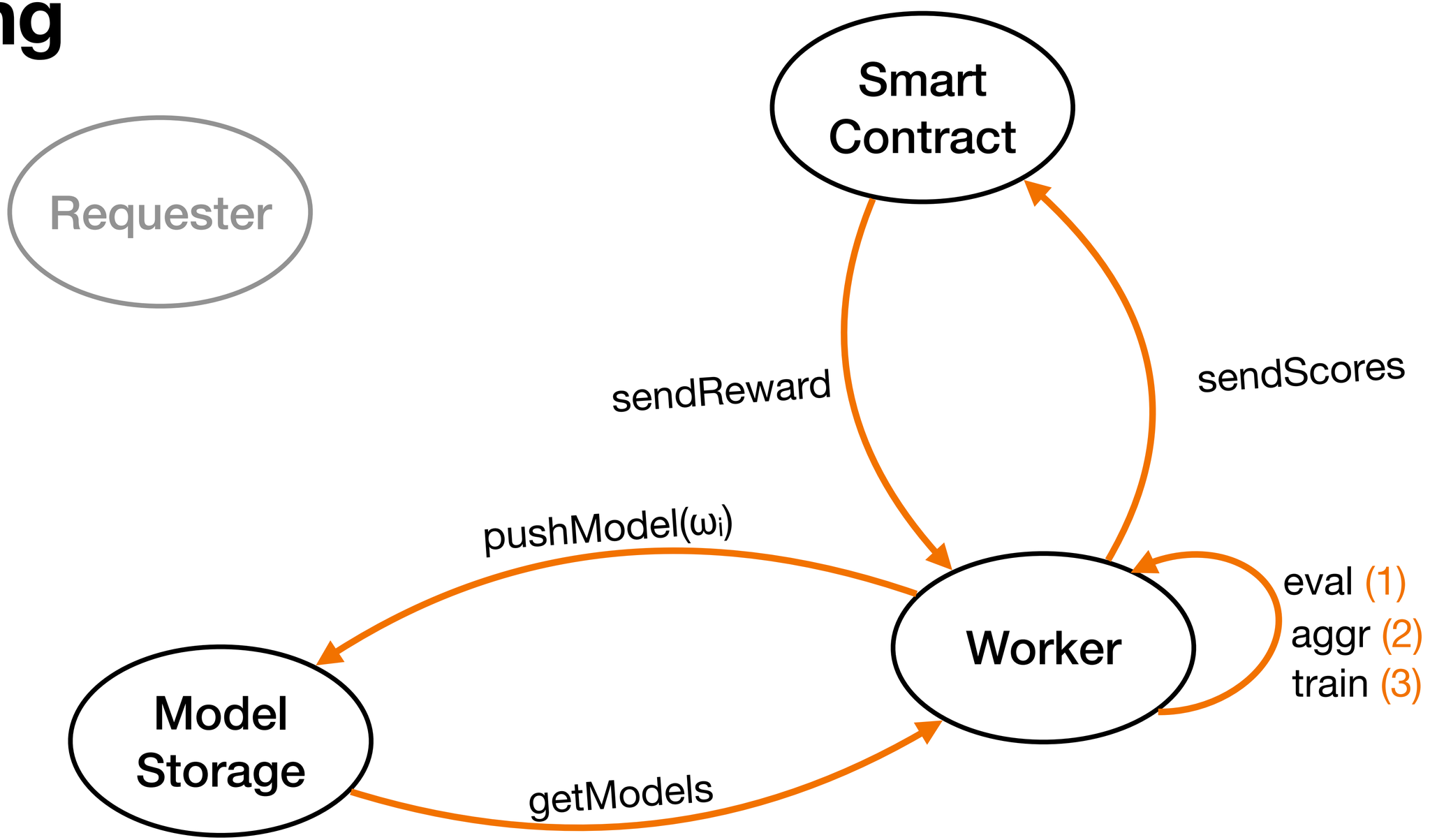
Training


Requester
Smart Contract
sendScores
eval (1)
aggr (2)
train (3)
Worker
Model Storage
getModels

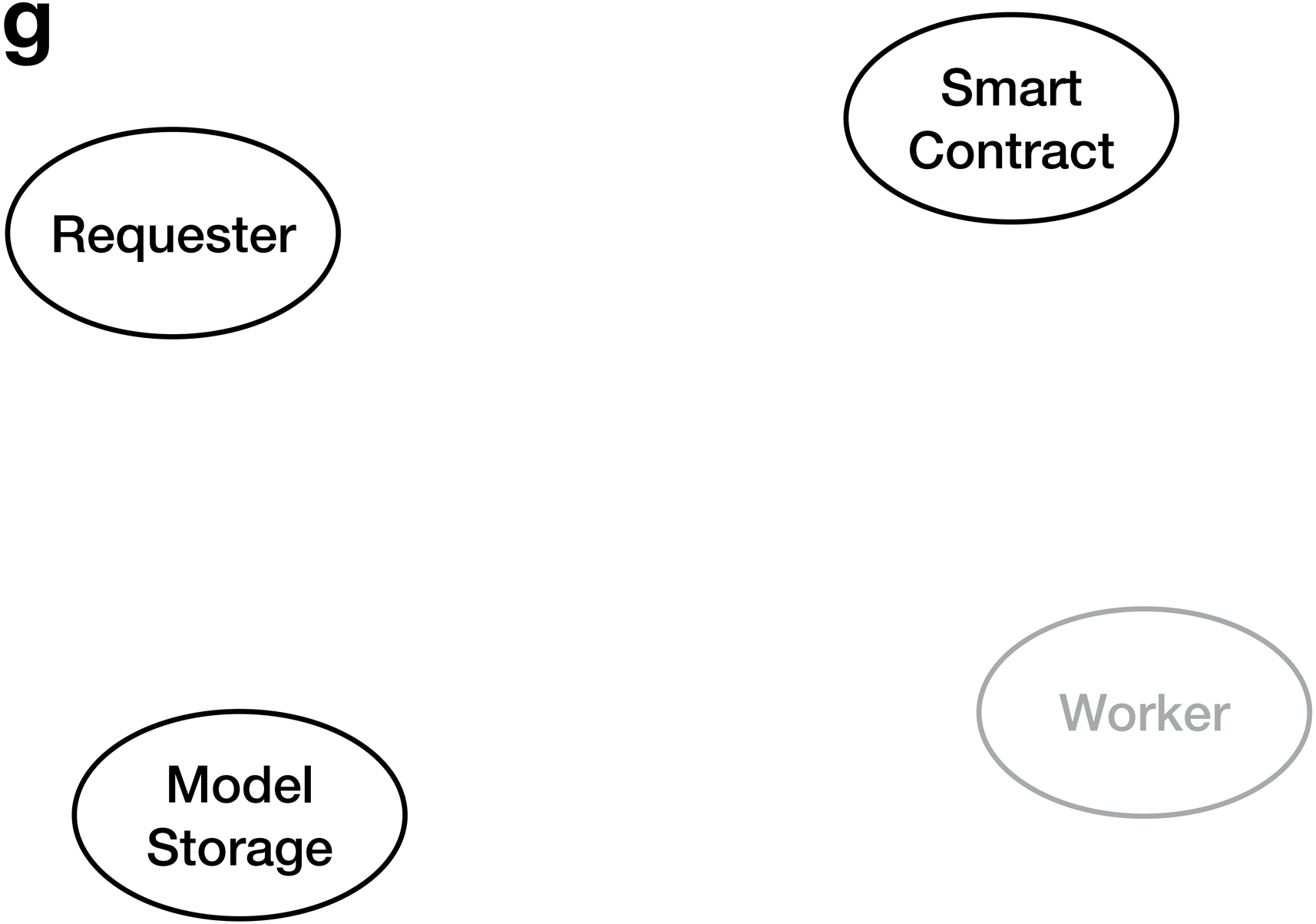
Training


Requester
Smart Contract
sendScores
pushModel(ωi)
Worker
eval (1)
aggr (2)
train (3)
Model Storage
getModels

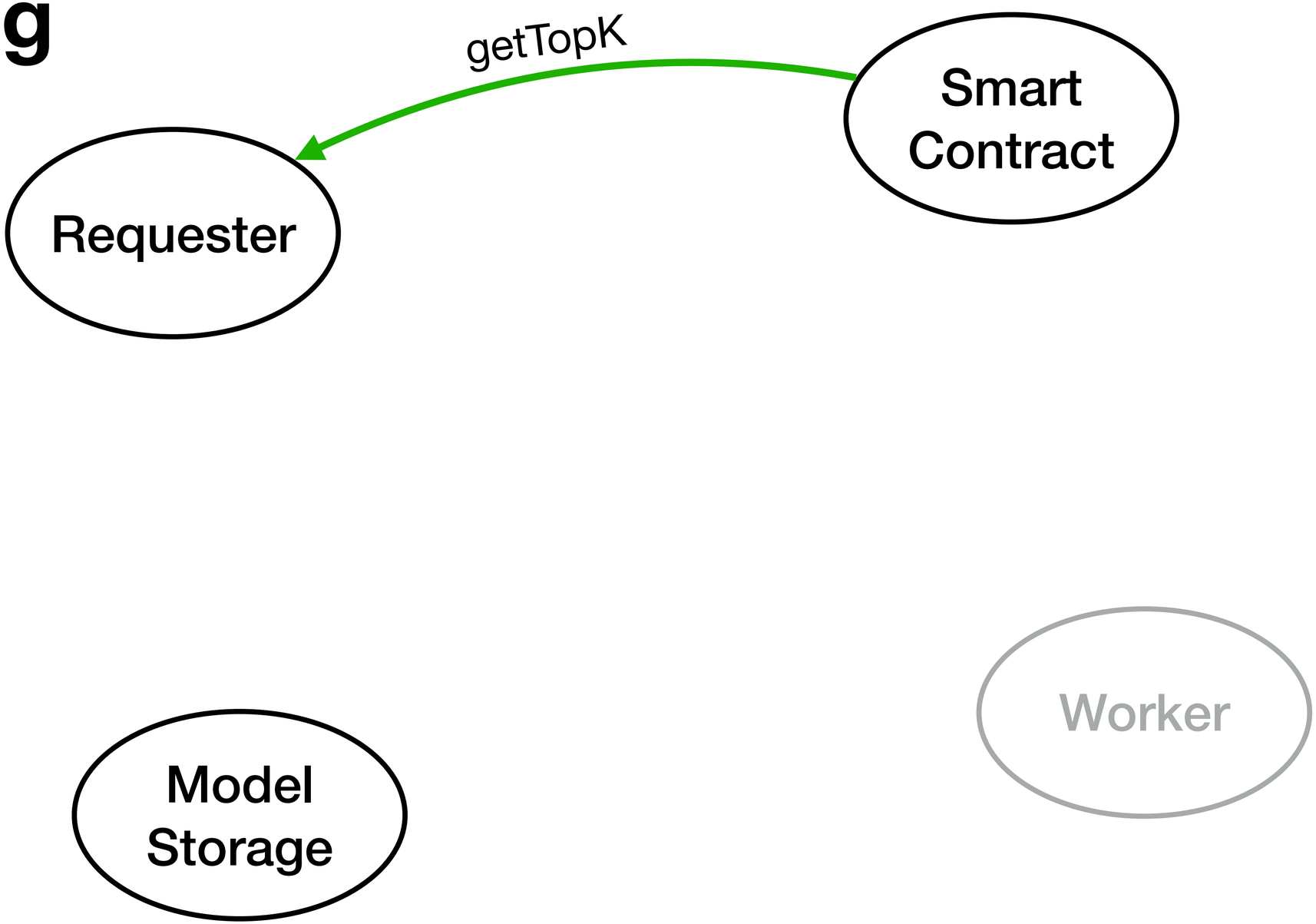
Training


Requester
Smart Contract
sendReward
sendScores
pushModel(ωi)
Worker
eval (1)
aggr (2)
train (3)
Model Storage
getModels

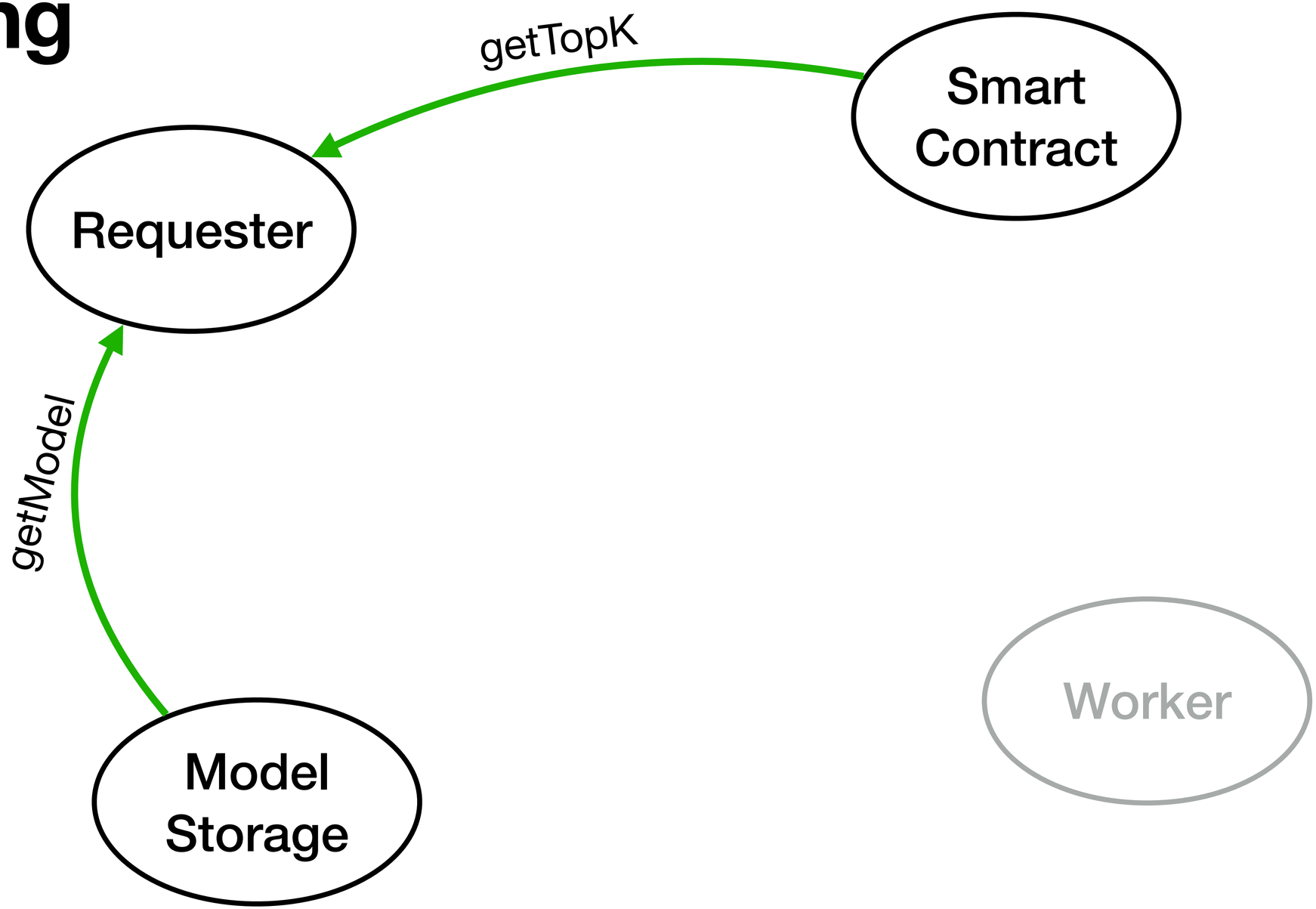
Closing



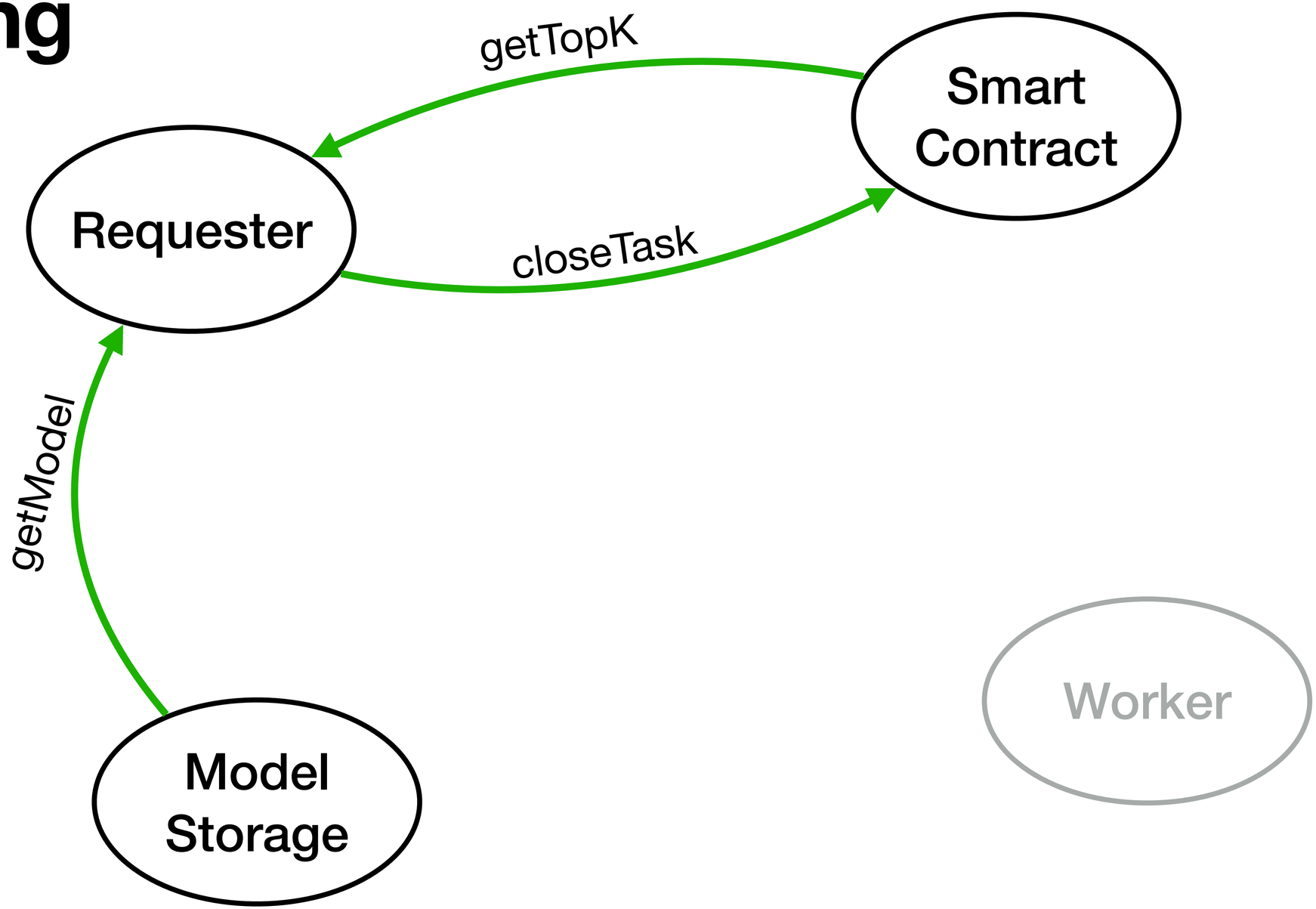
Closing



Closing


getTopK
Smart Contract
Requester
getModel
Model Storage
Worker

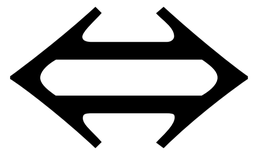
Closing


getTopK
Smart Contract
Requester
closeTask
getModel
Model Storage
Worker

Trust

Trust

Trustworthy system



Fair behaviour of workers and **rewards** based on **contribution**

Trust

Trustworthy system ⇔ **Fair behaviour** of workers and **rewards** based on **contribution**

- Poisoning the model is not leading to worthy rewards
- The workers are anonymized
  - Not possible to trigger rewards for “friend workers”

Rewarding system

- Based on currency **deposit** on Smart Contract and mutual **evaluation**
- Workers are “paid” by the requester
- The requester is rewarded with the training of the model

Evaluation system

- Each worker evaluates other workers' model on his validation set
  - Better model performance ⇒ Better reward
- BlockFLow-like contribution scoring procedure
  - Penalize low-quality models and inaccurate evaluations

Implementation

ML Implementation

- **Assumption:** workers data is disjoint
- 20% of worker's datasets are used for validation
- In principle, the architecture is model agnostic
  - Requester distributes the initial model
  - More sophisticated aggregation schemes could be implemented (*see future work*)

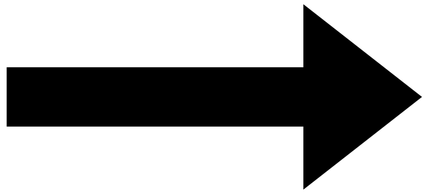
ML Implementation

- **Lack of infrastructure enforces simulation of distributedness**
  - Simplifies playing with different parameters
  - Simplifies timing and synchronization issues
  - Simplifies storage access

Smart Contract Implementation

- Incentive mechanism implemented through smart contract
- Deployed by the requester, interacts with workers and requester
- Main functionalities:
  - Round coordination
  - Score aggregation
  - Reward distribution

Smart Contract Implementation



Future work & Conclusions

Future work

- Decentralize File System
  - Use IPFS (c.f. BlockFLow)
  - Store model on Blockchain (c.f. BAFFLE)
- Extend prototype to work with distributed network
- Implement cryptography and anonymity
- Improve the aggregation algorithm

Conclusions

- We designed **DiscoFL**: an architecture for decentralized FL powered by a Blockchain-based incentive mechanism.
- We defined our notion of trust and how our system can be trustworthy.
- We implemented a simplified prototype providing a demo of the main functionalities of DiscoFL.

Thanks for the attention!

Any questions?