Payment Dev Spec

Summary:

In MainNet 1.0, Smart Contract will

- Monitor all jobs status
- Monitor all nodes status
- Trigger job
- Punishment and reward

Related Data Structure

JobStatus {
    Type = Training | PSI | Aggregation - 8bit
    Epoch = 0 .. n -> 0xffffffff... means whole training - 32bit //it can also be used by PSI to record the levels. i.e. level 0: A-B => A1, C-D=>C1; level 1: A1-B1=>A2
    Batch = 0 .. n, 0xffffffff means whole epoch - 32bit
    Status = Started | InProgress | Completed | Failed - 8bit
}

RewardSchema {
    ModelComplexityFactor: default 1
    //Horizontal: slots = ModelComplexityFactor * Number of DOs
    Fees: [Price of DO_i * ModelComplexityFactor, MPC_i=Price, GDS=extra 10%]
    //Vertical: slots = ModelComplexityFactor * Number of DOs, Number of DOs * Factor
    Fees [DO_i * ModelComplexityFactor, MPC=Price * Number Of DOs * Factor, GDS=extra 10%)
}

Job {
    StakeIn
    Type = Vertical | Horizontal
    DOs
    MPCs //Can be updated multiple times by GDS, but number of MPCs cannot be changed
    RewardSchema
    ID //UUID - 16bytes
    ….

}

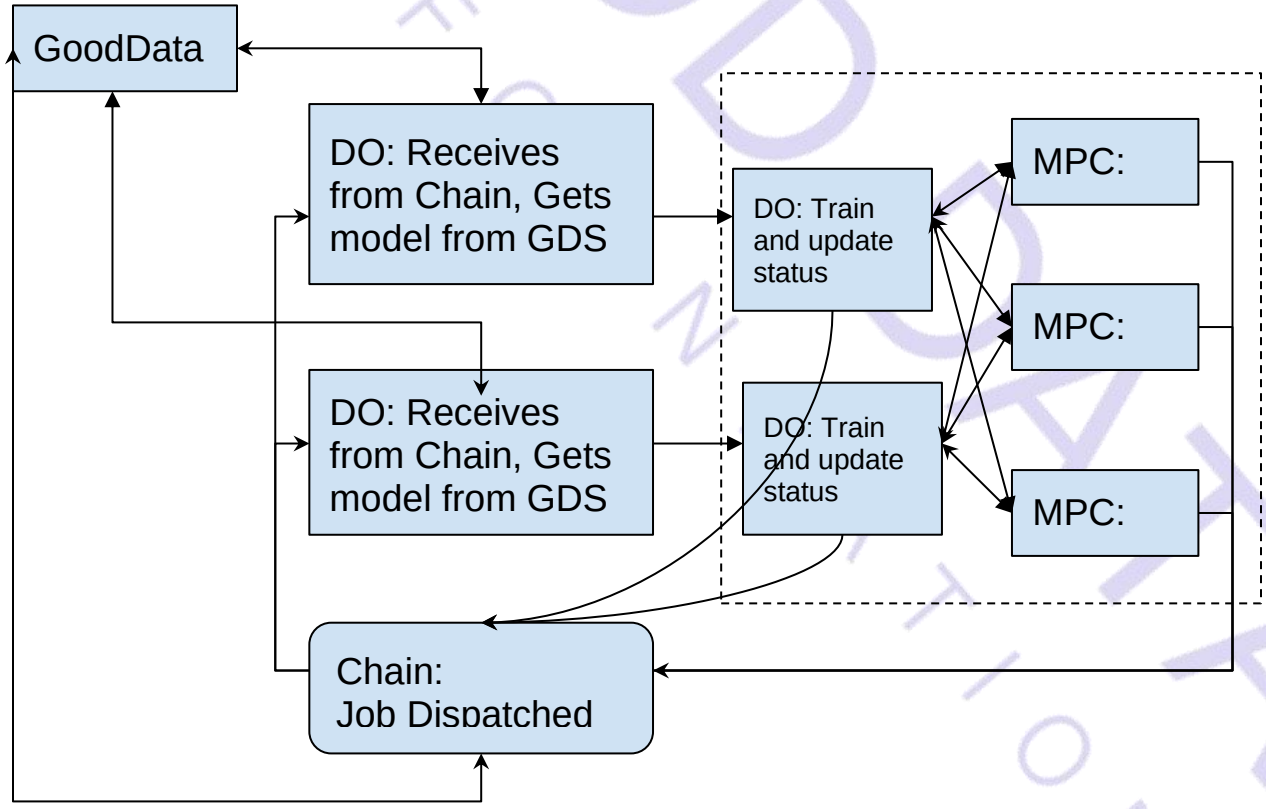
Job Trigger

GDS Places Order with DOs, Quote(Might not be exactly amount since MPCs is selected after quotation) and ID to Chain
QC Submits Query to Chain(Frontend generates the TX)
    If MPCs and RewardSchema are ready, SC emits JobDispatched(Type, DOs, MPCs)
GDS chooses MPCs, update MPCs and RewardSchema to Chain
    If QC has submitted query(StakeIn), SC emits JobDispatched(Type, DOs, MPCs)

Anti-Fraud

Horizontal Training

Workflow:



Payment and Punishment:

If Message comes from DO/MPC, validate the Epoch/Batch/Type, emit JobStatus(ID, Status)

1. Happy Cases: All Dos complete the training, based on RewardSchema, distribute reward to all roles
2. DO Failed:
    - Fine Tokens = Price DOi * Number of DOs, and distribute the tokens to other DOs and MPCs. e.g.
    - DO1, DO2, DO3, MPC1, MPC2, MPC3 are working on a job, DO1 failed.
    - During allocating DOs, require DO1 has tokens = 3 * Price * ModelComplexityFactor slots. Now 80%(Why? Do reports actively) tokens are fined, and distribute to other DOs, MPCs, might 10% for GDS, 30% for MPCs, and 40% for other DOs
    - Refund StakeIn to QC
3. MPC Failed:
   How DOs communicate with MPCs? It is better to ask GDS to replace MPC, the failure MPC cannot get any reward.
    - Once DO gets one of MPC is not usable, it should ASK GDS to replace MPC
    - DO receives MPC replace message, re-do aggregation batch(might multiple DOs)
    - System can limit the replacement times.

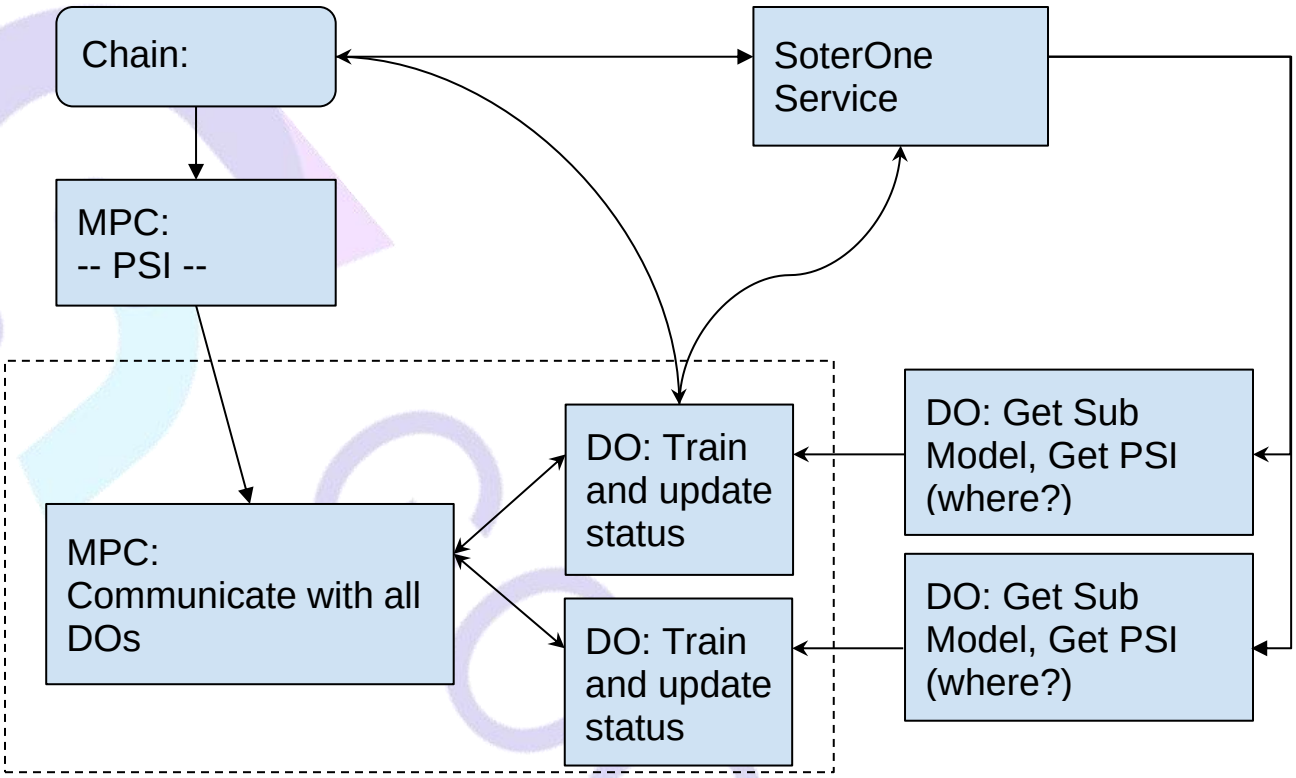
Otherwise the Message comes from GDS

4. GDS checked failure
   Same as above, but all tokens are fined.
5. QC cancels the job (submit request to GDS)
   60% StakeIn are fined, and distribute to other roles

Vertical Training

Workflow:

- PSI Stage, no payment is required now, but every node still requires to update its job status.
  UpdateStatus(UUID, PSI, epoch, batch, status)
- Training Stage:



Reward and Punishment:

If Message comes from DO/MPC, validate the Epoch/Batch/Type, emit JobStatus(ID, Status)

1. Happy Cases: MPC completes the training, based on RewardSchema, distribute reward to all roles
2. DO Failed:
    - Fine Tokens = Price DOi * Number of DOs, and distribute the tokens to other DOs and MPC. e.g.
    - DO1, DO2, DO3, MPC are working on a job, DO1 failed.
    - During allocating DOs, require DO1 has tokens = 3 * Price * ModelComplexityFactor slots. Now 80%(Why? Do reports actively) tokens are fined, and distribute to other DOs, MPC, might be 10% for GDS, and 90% for other DOs and MPC
    - Refund StakeIn to QC
3. MPC Failed:
   Fine Tokens = Price * Number Of DOs * Factor * 80%, and distribute to other members

Otherwise the Message comes from GDS

4. GDS checked failure(DOi or MPC)
   Same as above, but all tokens are fined.
5. QC cancels the job (submit request to GDS)
   60% StakeIn are fined, and distribute to other roles

Smart Contract

- No fallback function, no one can transfer token to SC directly
- Payment related function
  - UpdateJobStatus(ID, JobStatus)
- Payment related events
  - JobStatusUpdated(ID, JobStatus)
  - RewardDistributed(ID, address=tokens, …)
  - Refund(ID, address=tokens, …)
  - JobDispatched(ID, DOs, MPCs)

Corner Scenarios:
Assume 10 DOs, each DO is planned to be rewarded 1000 Tokens.
Requires each DO to deposit 10*1000 = 10k token
Make sure the smart contact is upgradable
Need to require each DO to have minimum token
Each DO will have sufficient token for the penalty

- Testing: speed up block generation in go-ethereum
  - Unit tests: Ethereum Development with Go: Using a Simulated Client. Call `client.Commit()` repeatedly to mine blocks
  - Integration and End-to-End tests: make the "number of blocks till timeout" configurable.
- Data and time mismatch between: MySQL <-> on-chain
  - One design: data in MySQL should get time from chain. Chain is the source of truth.

Scenarios	Penalty	Error logic needed
1. GDS detects DOx did not update heart beat after 3600 blocks	• GDS will return 10K back to QC • GDS will deduct total 10K from all the offended DOs that failed (i.e., deduct 10k for one DOs, and deduct 5k from each DO for 2 DOs, etc.) • Distribute originally promised 1000 token to good behavior DOs	
2. DO claims to fail during the MPC	• Assume one or more coreDO(s) claim to fail at x percentage of epoch, and notify	Failed DO will periodically update this status on the chain. Upon detecting the failed DO

calculation	GDS • GDS will send the abort messages to all other DOs, and request the latest epoch of each DO • Based on the actual computation cost of each good performed DOs, deduct from the offended DOs' deposit • Distribute to each DOs based on their completed epoch percentage out of total epoch	from the chain event, GDS will start to collect the epoch info from the chain of each good behaved DOs.
3. GDS detects MPC did not finish the job exceeding 3600 blocks	• GDS will return 10K back to QC • GDS will deduct total 10K from MPC • Distribute originally promised 1000 token	
4. MPC claims to fail during the MPC calculation	• Assume the MPC node claim to fail at x percentage of epoch, and notify GDS • GDS will send the abort messages to all other DOs, and request the latest epoch of each DO • Based on the actual computation cost of each good performed DOs, deduct from MPC node's deposit • Distribute to each DOs based on their completed epoch percentage out of total epoch	
5. GDS detect DO (horizontal) 's returned results do not comply with modeling requirements	• GDS will return 10K back to QC • GDS will deduct total 10K from all the offended DOs that failed (i.e., deduct 10k for one DOs, and deduct 5k from each DO for 2 DOs, etc.) • Distribute originally promised 1000 token	
6. GDS detect MPC (vertical) returned results do not comply with modeling requirements	• GDS will return 10K back to QC • GDS will deduct total 10K from all the offended DOs that failed (i.e., deduct 10k for one DOs, and deduct 5k from each DO for 2 DOs, etc.) • Distribute originally promised 1000 token	