7 Appendix

Here we show an example of smart contract code generated for the Paper Review collaboration.

Code 1.2. Part of contract code for Paper Review collaboration.

```
1   function AssignPaper() external {  // a SMST
2       if (isActiveAssignPaper) { // check the state
3           emit mesag("", "", "AssignPaper"); // sending msg to blockchain
4           isEnabledReceivePaper++; // enable the receive task
5           CountAssignPaper ++; // increase the instance count
6           if (CountAssignPaper == 5){ // enough instances
7               isActiveAssignPaper = false;  // no more insatnce for this task
8               isActiveReceiveReviews = true;// next task can start
9               // time window for next task
10              DDLReceiveReviews = block.timestamp + DurationReceiveReviews;
11          }
12      }
13  }
14  function ReceivePaper() external{  // a task in a PMP
15      if ((isActiveReceivePaper) && (isEnabledReceivePaper > 0)) { // check the state
16          emit mesag("", "", "ReceivePaper"); // sending msg to blockchain
17          CountReceivePaper ++; // increase the instance count
18          isEnabledReceivePaper --;
19          isActivatedSubmitReview ++; // next task can have a instance
20          if (CountAssignPaper == 5){ // enough instances
21              isActiveReceivePaper = false;   // no more insatnce for this task
22          }
23      }
24  }
25  function ReceiveReviews() external{ // a TPMRT
26      // check the state and time
27      if (isActiveReceiveReviews && (block.timestamp < DDLReceiveReviews)) {
28          emit mesag("", "", "ReceiveReviews"); // sending msg to blockchain
29          CountReceiveReviews ++; // increase the instance count
30          if (CountReceiveReviews == 3){ // enough instances
31              isActiveReceiveReviews = false;  // no more insatnce for this task
32              isActiveSendResult = true;   // next tasks can start
33              isActiveteSendFeedback = true;
34          }
35      }
36  }
```