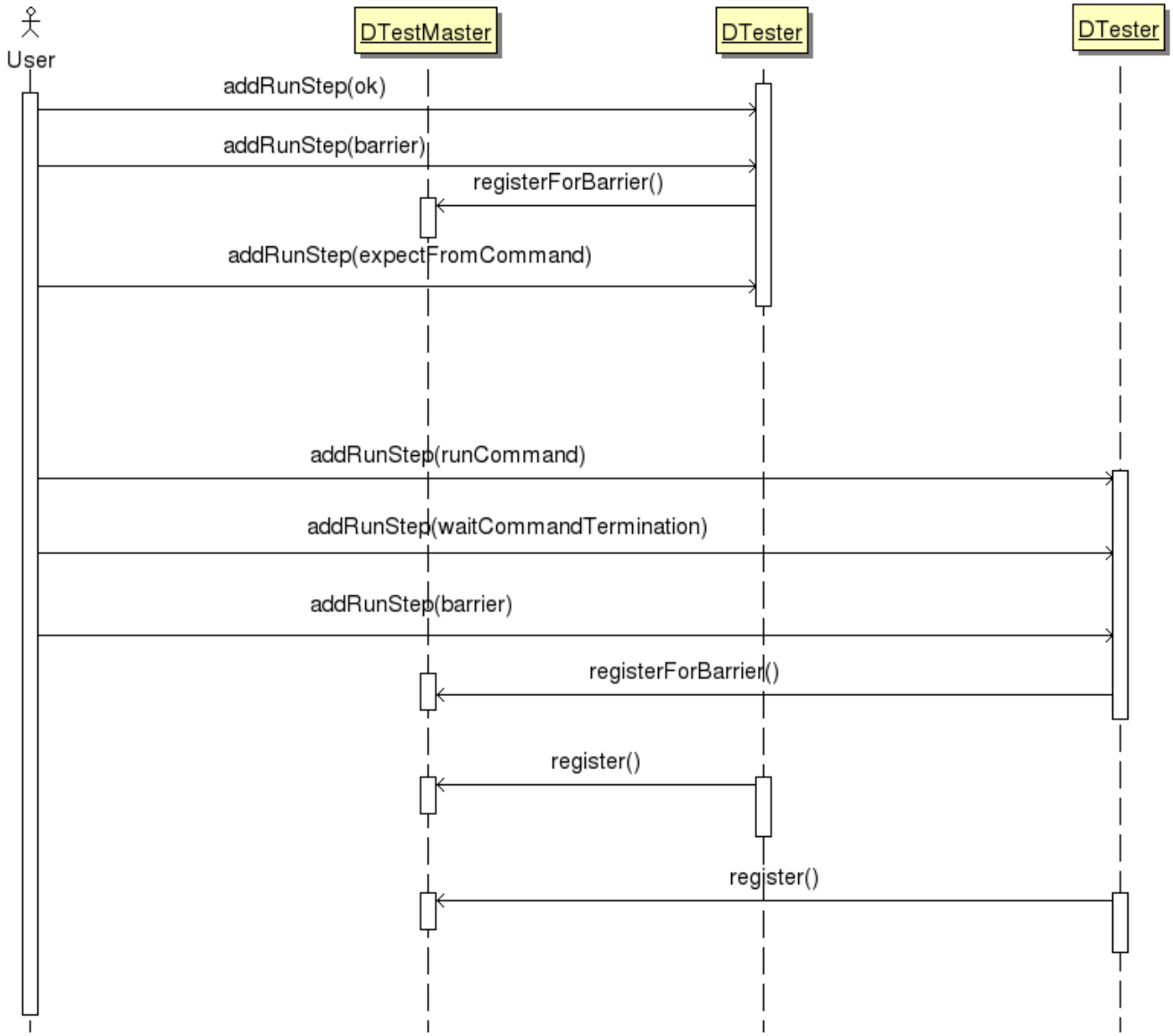
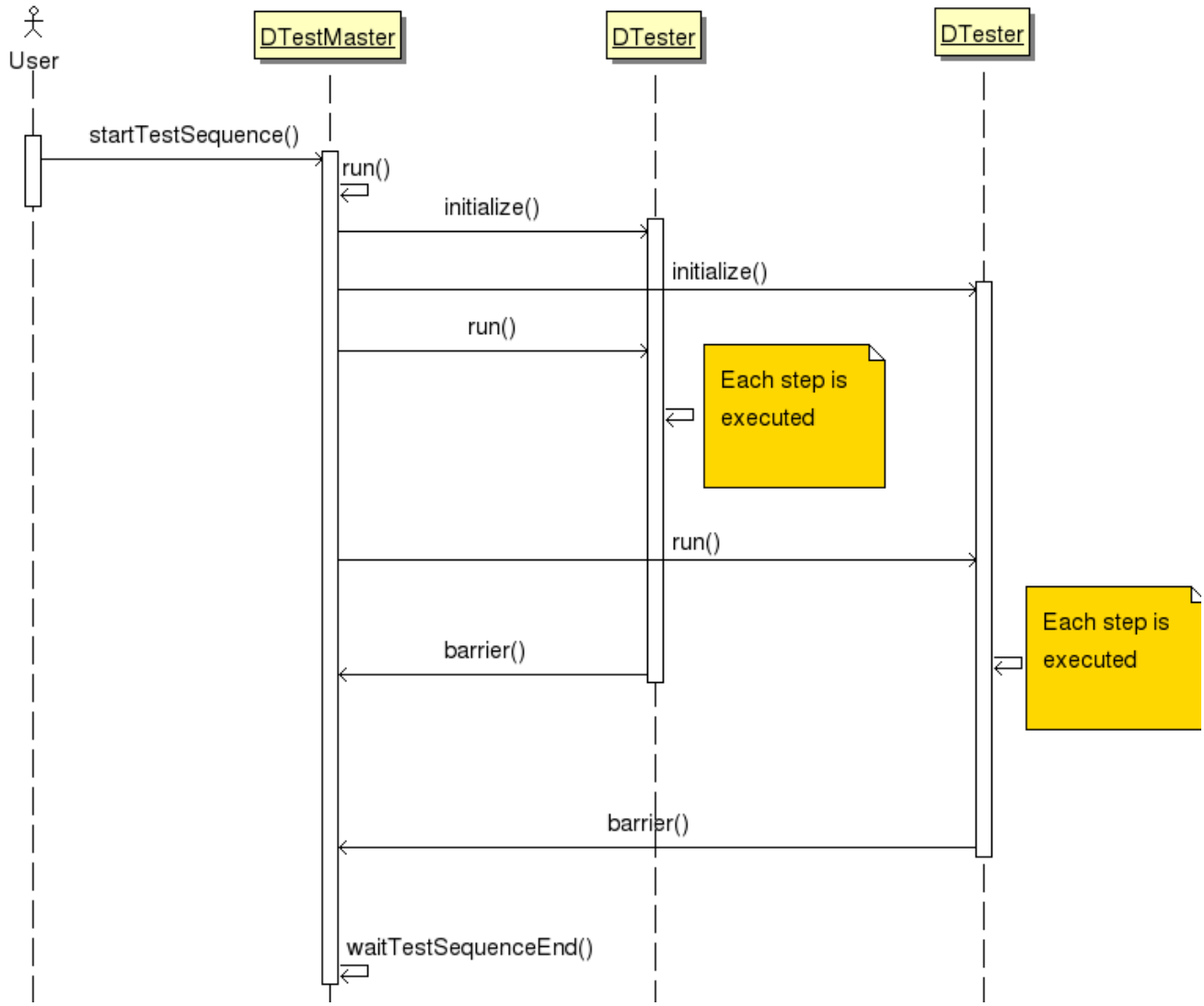
dtest
Documentation

Phase 1 : definition of execution steps of each DTester
The user adds execution steps to Dtester with addRunStep().
Barrier type steps automatically register to DTestMaster with registerForBarrier().
Then DTesters must register to DTestMaster with register().

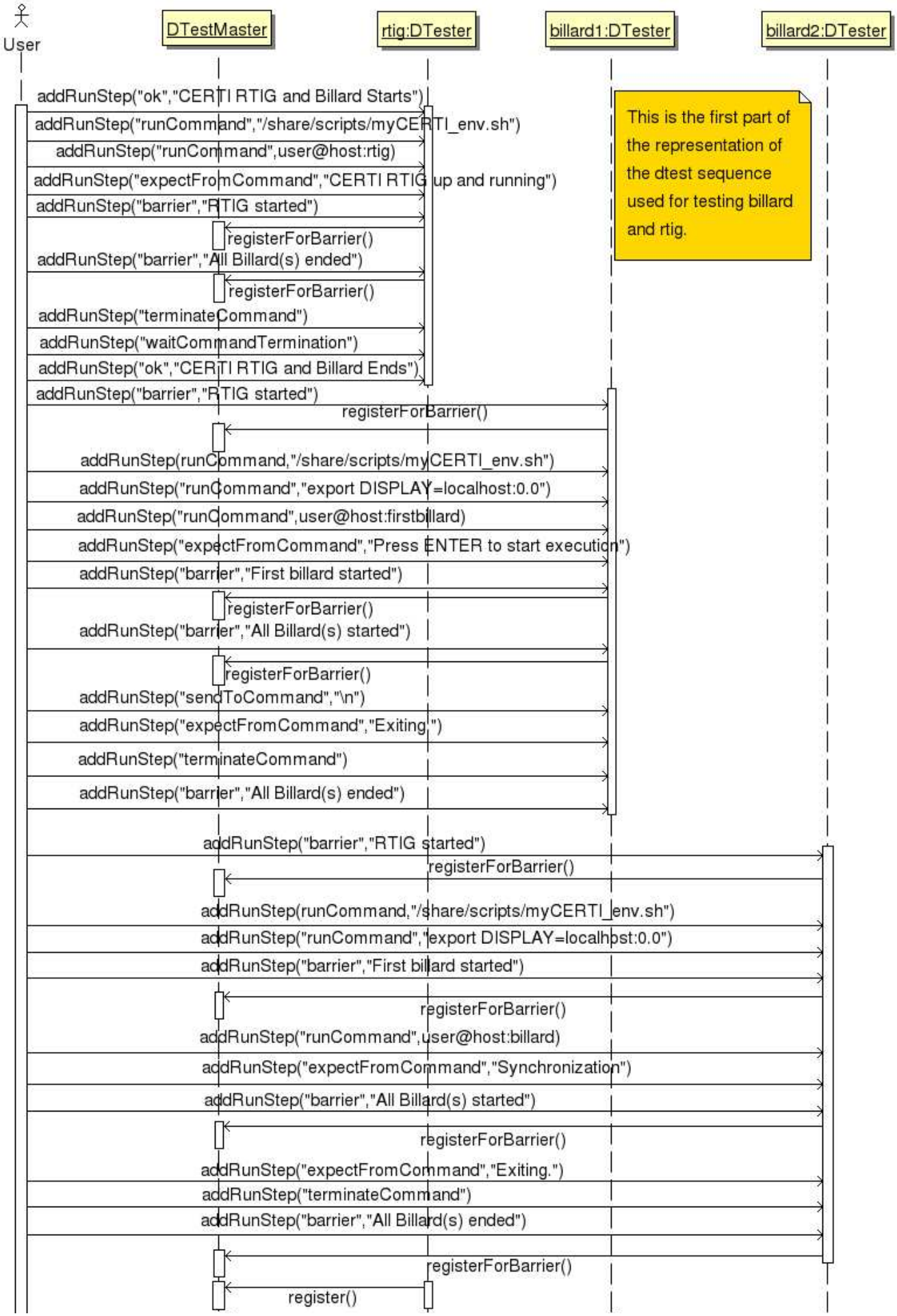


diagram_sequence_dtest

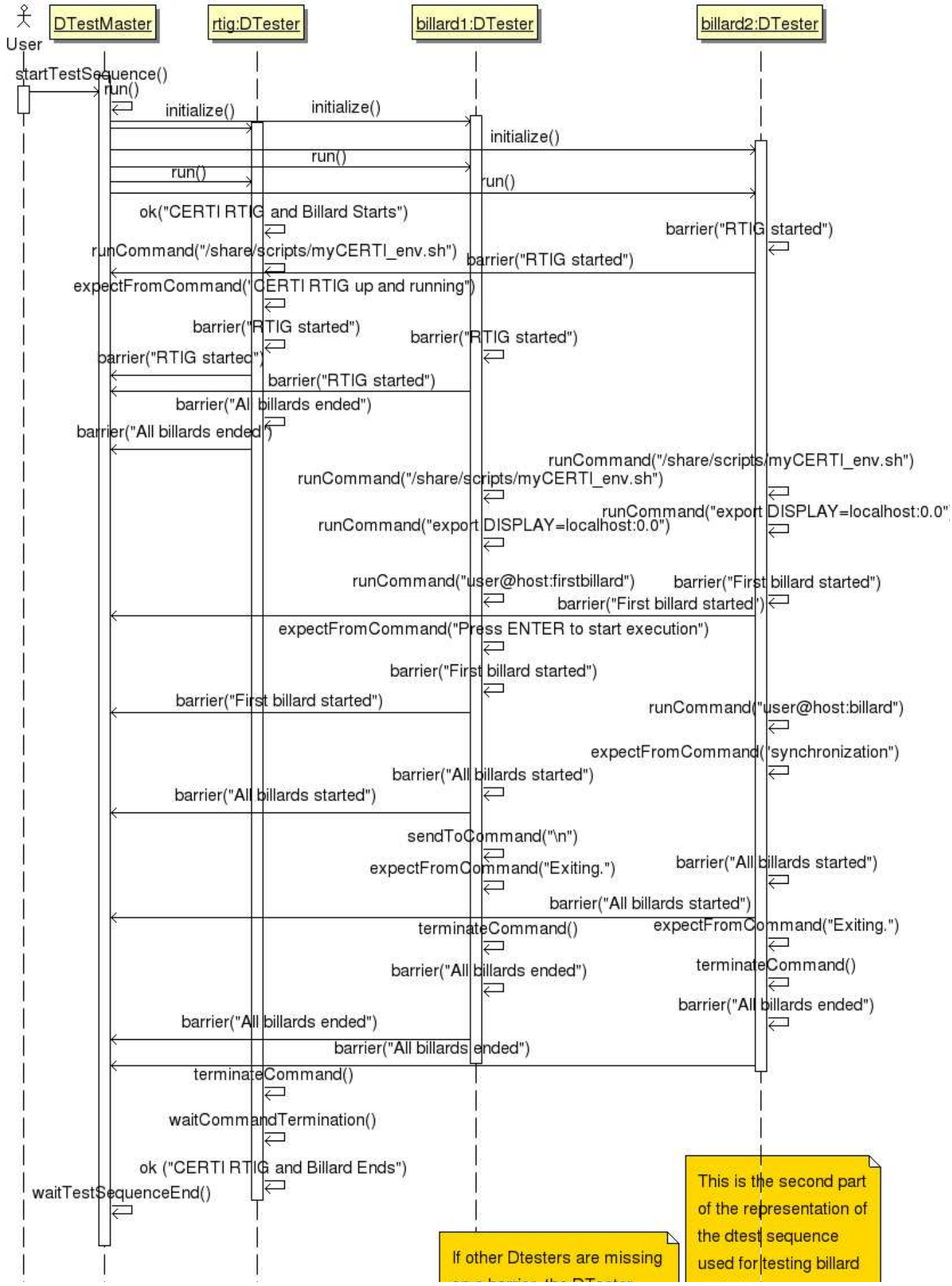
Phase 2 : the user starts test sequence with startTestSequence().
First, Dtester threads are initialized and launched.
Then, each step previously defined is executed.
When a DTester reaches a barrier, it informs DTestMaster with barrier().
If other Dtesters are missing on this barrier, the DTester waits them.
Finally, the DTestMaster wait the end of each DTester with waitTestSequenceEnd().



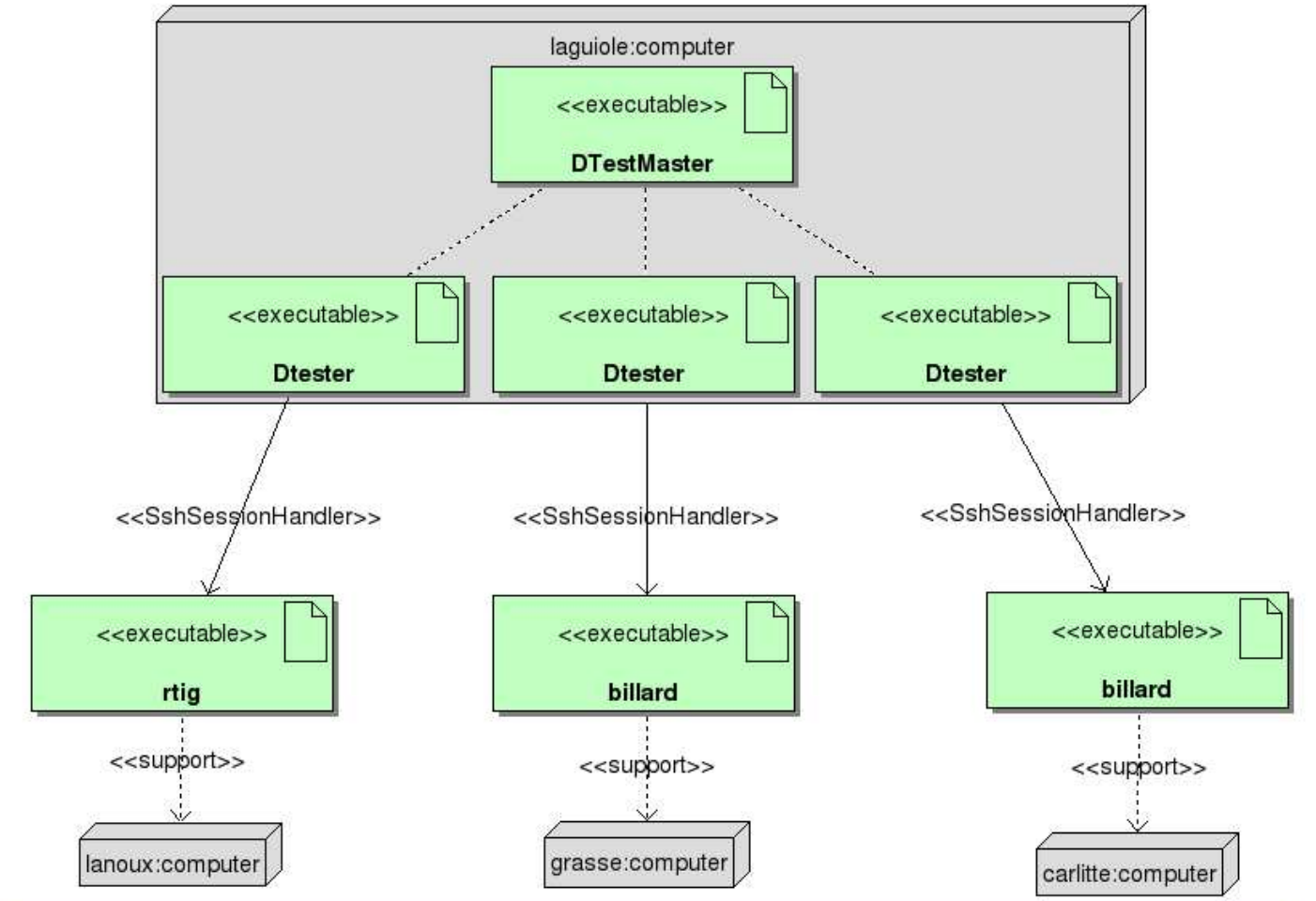
diagram_sequence_dtest2


User
DTestMaster
rtig:DTester
billard1:DTester
billard2:DTester
This is the first part of the representation of the dtest sequence used for testing billard and rtig.
addRunStep("ok","CERTI RTIG and Billard Starts")
addRunStep("runCommand","/share/scripts/myCERTI_env.sh")
addRunStep("runCommand",user@host:rtig)
addRunStep("expectFromCommand","CERTI RTIG up and running")
addRunStep("barrier","RTIG started")
registerForBarrier()
addRunStep("barrier","All Billard(s) ended")
registerForBarrier()
addRunStep("terminateCommand")
addRunStep("waitCommandTermination")
addRunStep("ok","CERTI RTIG and Billard Ends")
addRunStep("barrier","RTIG started")
registerForBarrier()
addRunStep(runCommand,"/share/scripts/myCERTI_env.sh")
addRunStep("runCommand","export DISPLAY=localhost:0.0")
addRunStep("runCommand",user@host:firstbillard)
addRunStep("expectFromCommand","Press ENTER to start execution")
addRunStep("barrier","First billard started")
registerForBarrier()
addRunStep("barrier","All Billard(s) started")
registerForBarrier()
addRunStep("sendToCommand","\n")
addRunStep("expectFromCommand","Exiting.")
addRunStep("terminateCommand")
addRunStep("barrier","All Billard(s) ended")
addRunStep("barrier","RTIG started")
registerForBarrier()
addRunStep(runCommand,"/share/scripts/myCERTI_env.sh")
addRunStep("runCommand","export DISPLAY=localhost:0.0")
addRunStep("barrier","First billard started")
registerForBarrier()
addRunStep("runCommand",user@host:billard)
addRunStep("expectFromCommand","Synchronization")
addRunStep("barrier","All Billard(s) started")
registerForBarrier()
addRunStep("expectFromCommand","Exiting.")
addRunStep("terminateCommand")
addRunStep("barrier","All Billard(s) ended")
registerForBarrier()
register()

diagram_sequence_dtest_example


User
DTestMaster
rtig:DTester
billard1:DTester
billard2:DTester
startTestSequence()
run()
initialize()
initialize()
initialize()
run()
run()
run()
ok("CERTI RTIG and Billard Starts")
barrier("RTIG started")
runCommand("/share/scripts/myCERTI_env.sh")
barrier("RTIG started")
expectFromCommand("CERTI RTIG up and running")
barrier("RTIG started")
barrier("RTIG started")
barrier("RTIG started")
barrier("RTIG started")
barrier("All billards ended")
barrier("All billards ended")
runCommand("/share/scripts/myCERTI_env.sh")
runCommand("/share/scripts/myCERTI_env.sh")
runCommand("export DISPLAY=localhost:0.0"
runCommand("export DISPLAY=localhost:0.0")
runCommand("user@host:firstbillard")
barrier("First billard started")
barrier("First billard started")
expectFromCommand("Press ENTER to start execution")
barrier("First billard started")
barrier("First billard started")
runCommand("user@host:billard")
expectFromCommand("synchronization")
barrier("All billards started")
barrier("All billards started")
sendToCommand("\n")
expectFromCommand("Exiting.")
barrier("All billards started")
barrier("All billards started")
terminateCommand()
expectFromCommand("Exiting.")
barrier("All billards ended")
terminateCommand()
barrier("All billards ended")
barrier("All billards ended")
barrier("All billards ended")
terminateCommand()
waitCommandTermination()
ok ("CERTI RTIG and Billard Ends")
waitTestSequenceEnd()
If other Dtesters are missing
This is the second part of the representation of the dtest sequence used for testing billard

diagram_sequence_dtest_example2



Here is represented the interaction between rtig, billards and their DTesters located on different computers.
Each Dtester is registred in a DTestMaster which coordinates the synchronization of all DTesters mainly with the barrier synchronization method.
A DTester is associated to one executable (here rtig or billard) , they can communicate through a remote ssh connection SshSessionHandler started by the DTester.
Once this connection is established, a DTester can send asynchronous commands to its executable and wait for output data to control the behavior of its executable.

diagram_deployment_dtest