TTCN-3 User Conference 2009
3 – 5 June 2009
ETSI, Sophia Antipolis, France

ISP RAS

Log once - Debug anywhere

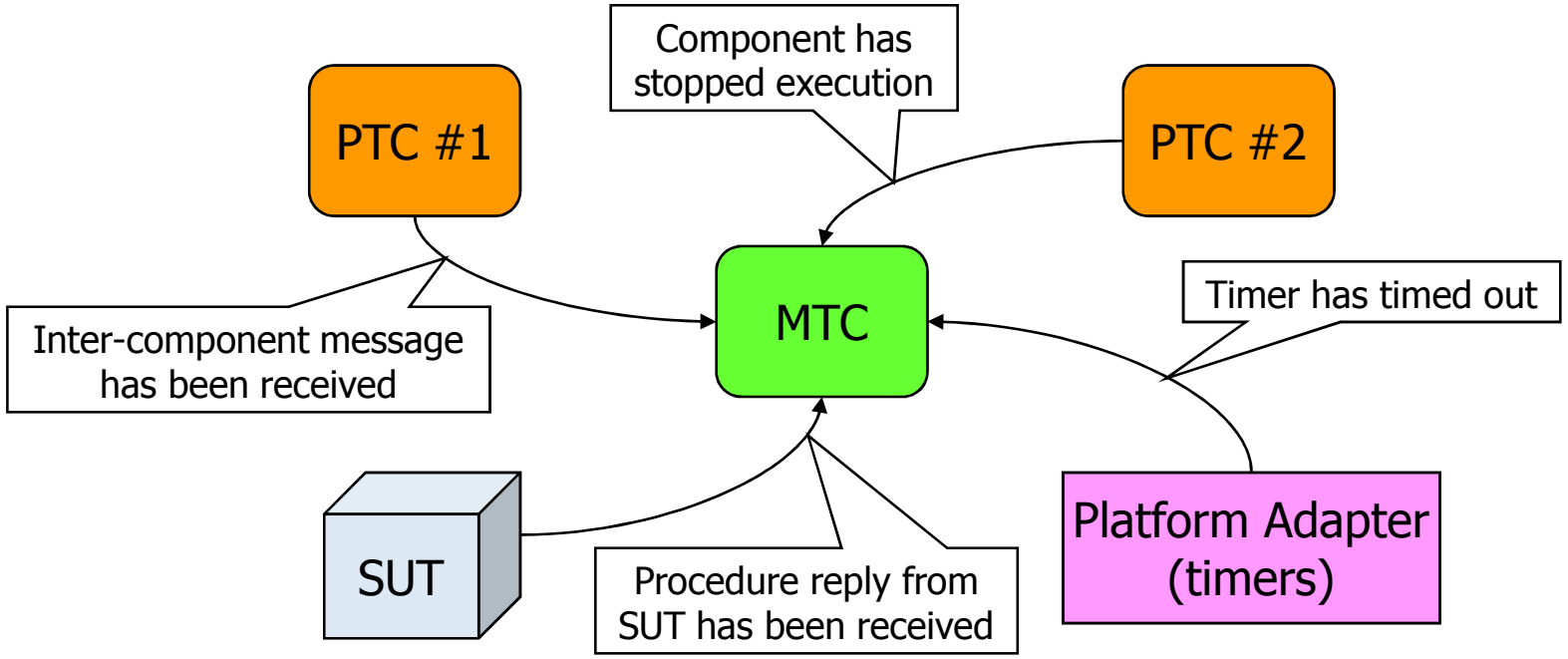
a portable deterministic approach to record-replay test case execution

Institute for System Programming
Russian Academy of Sciences

Pavel Iakovenko
yak@ispras.ru

TTCN-3 debugging obstacles

- Variations in the test case behavior
  - Sequences of events registered by a component during two subsequent test case executions are different
- Test suite execution involves interactions between the set of asynchronously behaving entities
  - Events that are "external" to a component may be delivered to it in the different order and at different time (related to the component start)

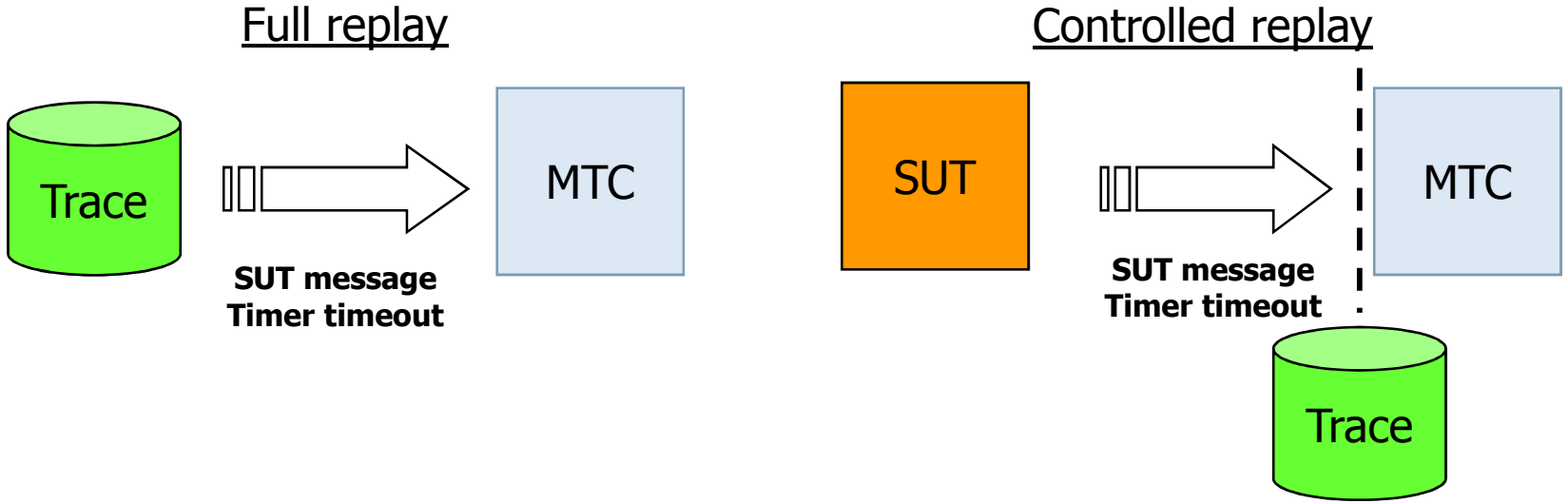


Other debugging obstacles

- Debugger watch point may require waiting for a long time until test case execution flow reaches it
  - Large execution time may be caused by the overall waiting for the SUT messages to arrive and timers to time out
- Watch point within protocol transaction may result in test case fail
  - SUT may fire internal timeout while user inspects test component state at the watch point hence causing test case to fail
- Necessity to have SUT available for every test case execution
  - Complicates off-site problem investigation if SUT is a non transferrable device

The Goal

- Eliminate non-determinism in the test case execution
- Record a trace of components' events during the reference execution and use it to drive (deterministically replay) subsequent test case executions
- The trace allows two replay modes
  - "**Full replay**" - all external events are taken from the trace
  - "**Controlled replay**" - the trace is used to reproduce the order and timing of the external events according to the reference execution
  - Mixed modes are possible



“Full replay” benefits

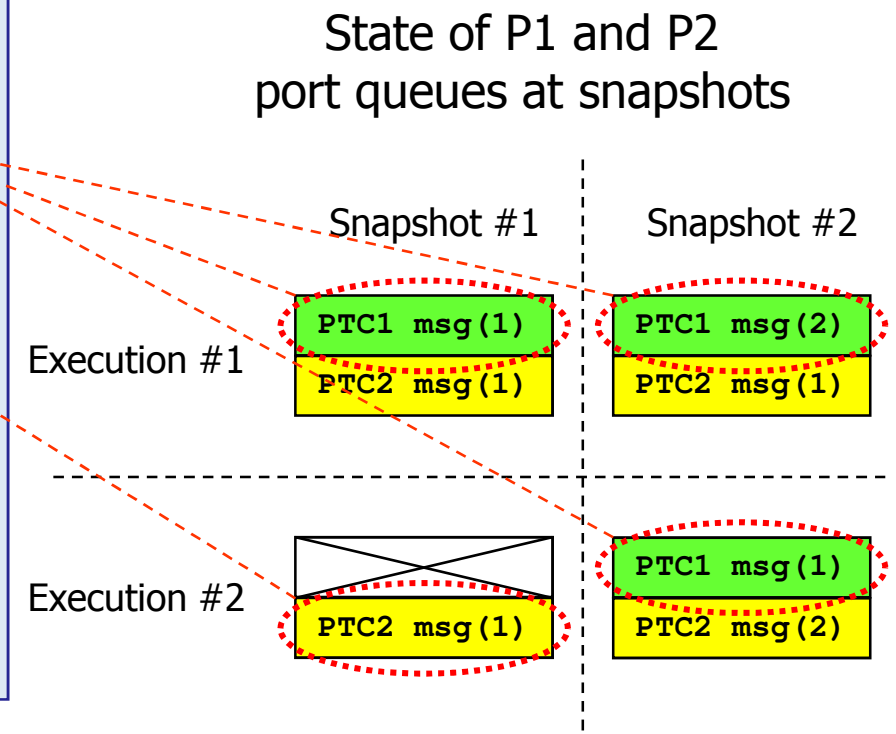
- SUT is not needed during the replayed executions
  - And original System Adapter too
- You may debug test suite on arbitrary PC
- No running timers
  - Timeouts are generated according to the trace
  - So you do not need original Platform Adapter too
- Timeouts are fired as soon as possible (without harming the order of “external” events)
  - You get simulated time for free
  - Test case execution time reduces
- Replay mode is non-intrusive
  - You may stay at the watch point in the debugger as long as you want without affecting test case or SUT execution

Variations in behavior

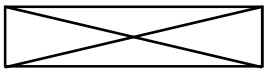
- Component behavior is well defined by the TTCN-3 code
- Behavior is driven by the input data
  - Module parameters
  - Test case parameters
  - SUT and component messages
  - Randomly generated data (rnd() predefined function)
  - External functions' out parameters and the return value
- Snapshots influence component behavior too!
  - Snapshot content depends on the order and timing of external events (e.g. SUT messages and timeouts)
  - For a particular 'alt' statement different snapshots mean different alternative branches are taken and different code is executed
- Order and timing of "external" events may change between test case executions
  - Hardware interrupt handling
  - CPU scheduling
  - Communication media

Snapshot example

```
PTC1.start(Behave1())
PTC2.start(Behave2())
alt
{ /* Snapshot is evaluated here */
  [] P1.receive from PTC1 -> value V1
     {
       Process1(V1);
       repeat; /* cycle */
     }
  [] P2.receive from PTC2 -> value V2
     {
       Process2(V2);
       repeat; /* cycle */
     }
  [] all component.done {}
}
```



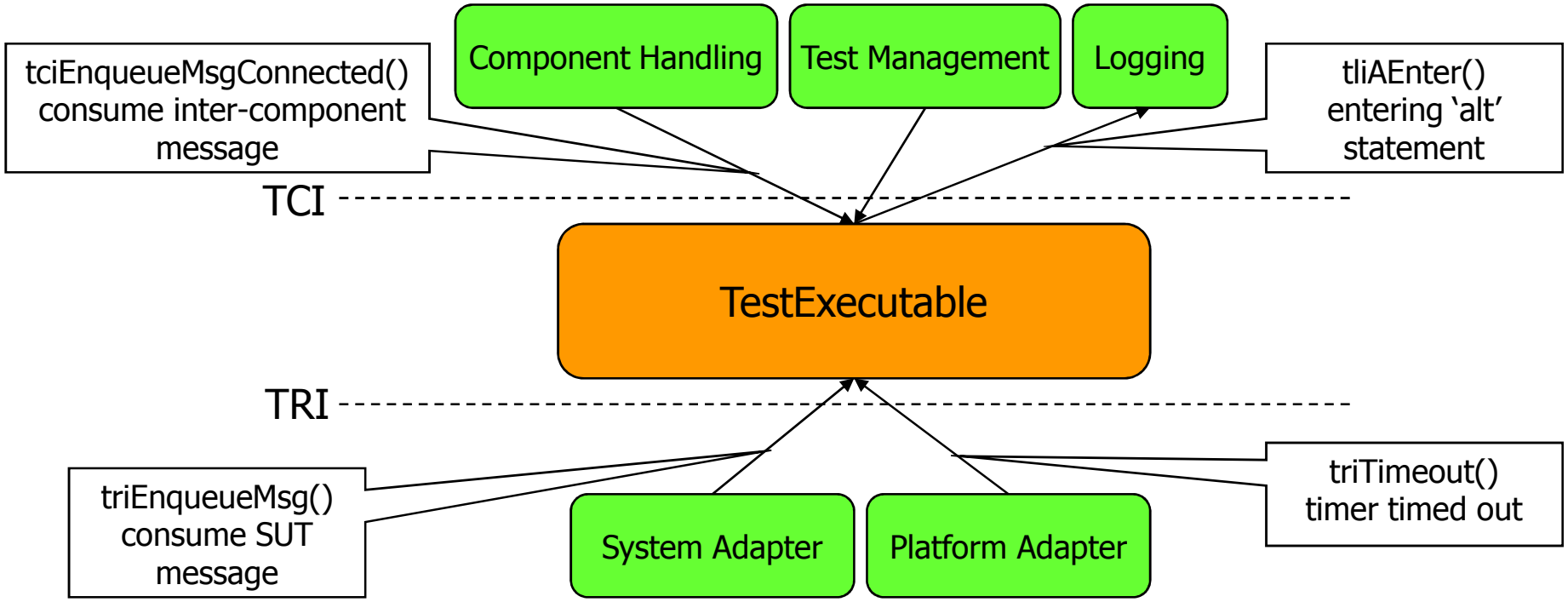
- This message will be received



- Port queue is empty

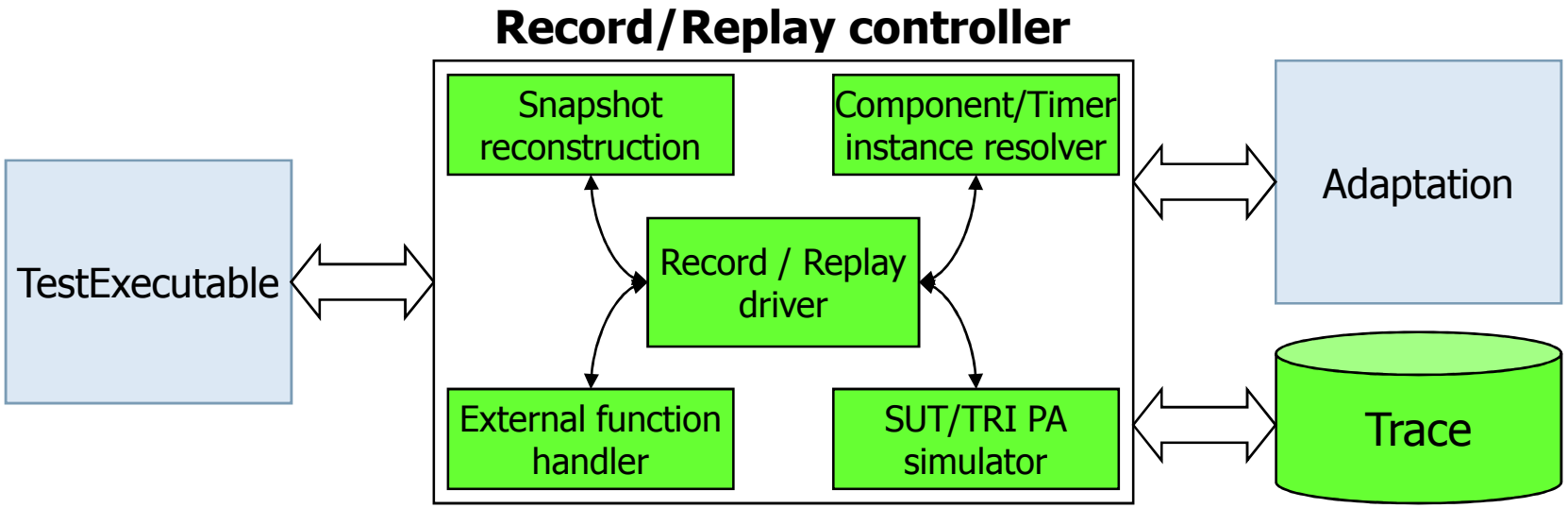
TTCN-3 runtime interfaces

- Test Executable interacts with the outer world via the set of runtime (TRI) and control (TCI) interfaces
- TRI and TCI provide complete control over the "external" events
- TCI-TL allows tracking "internal" events (e.g. entering 'alt' statement)

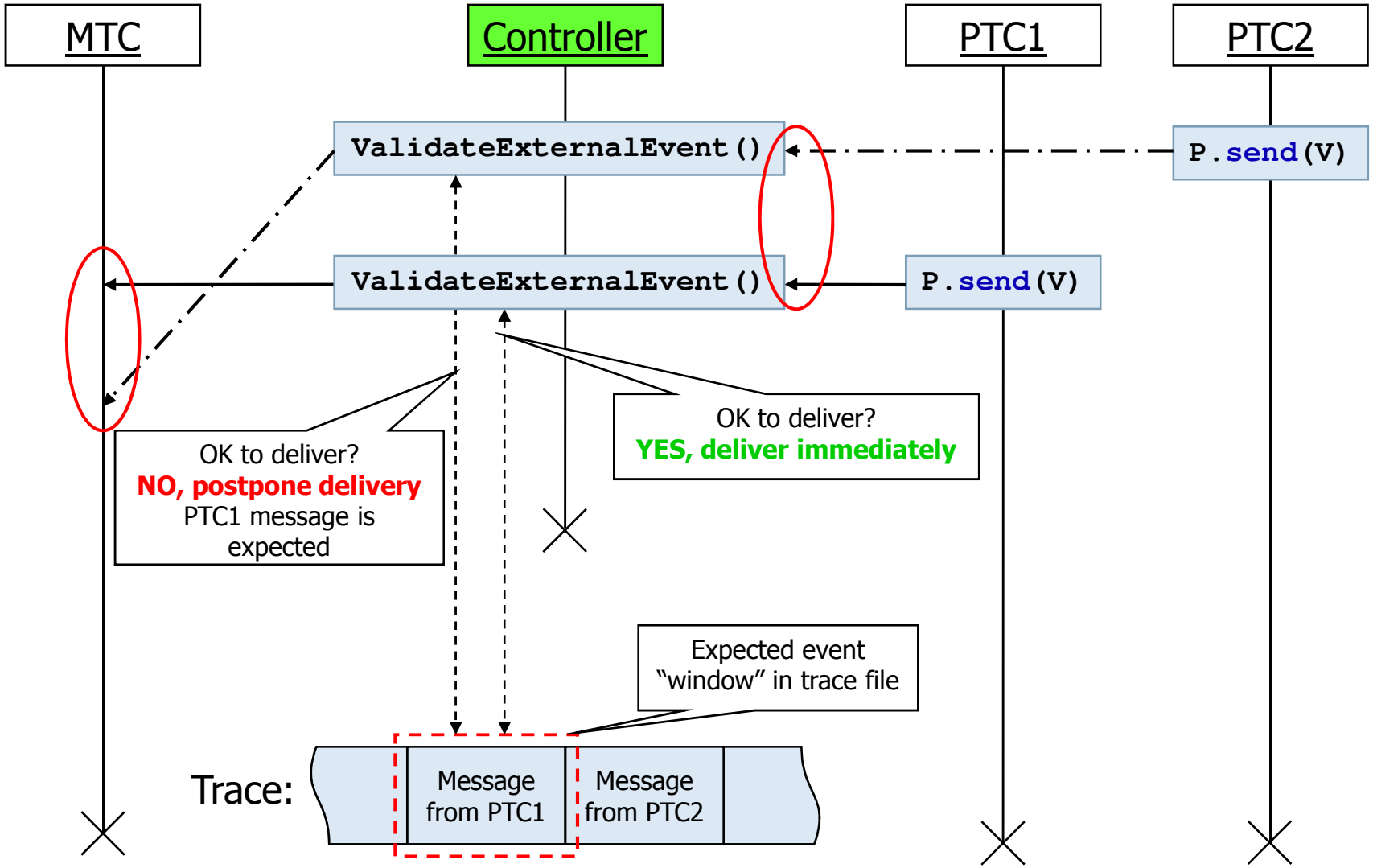


System architecture

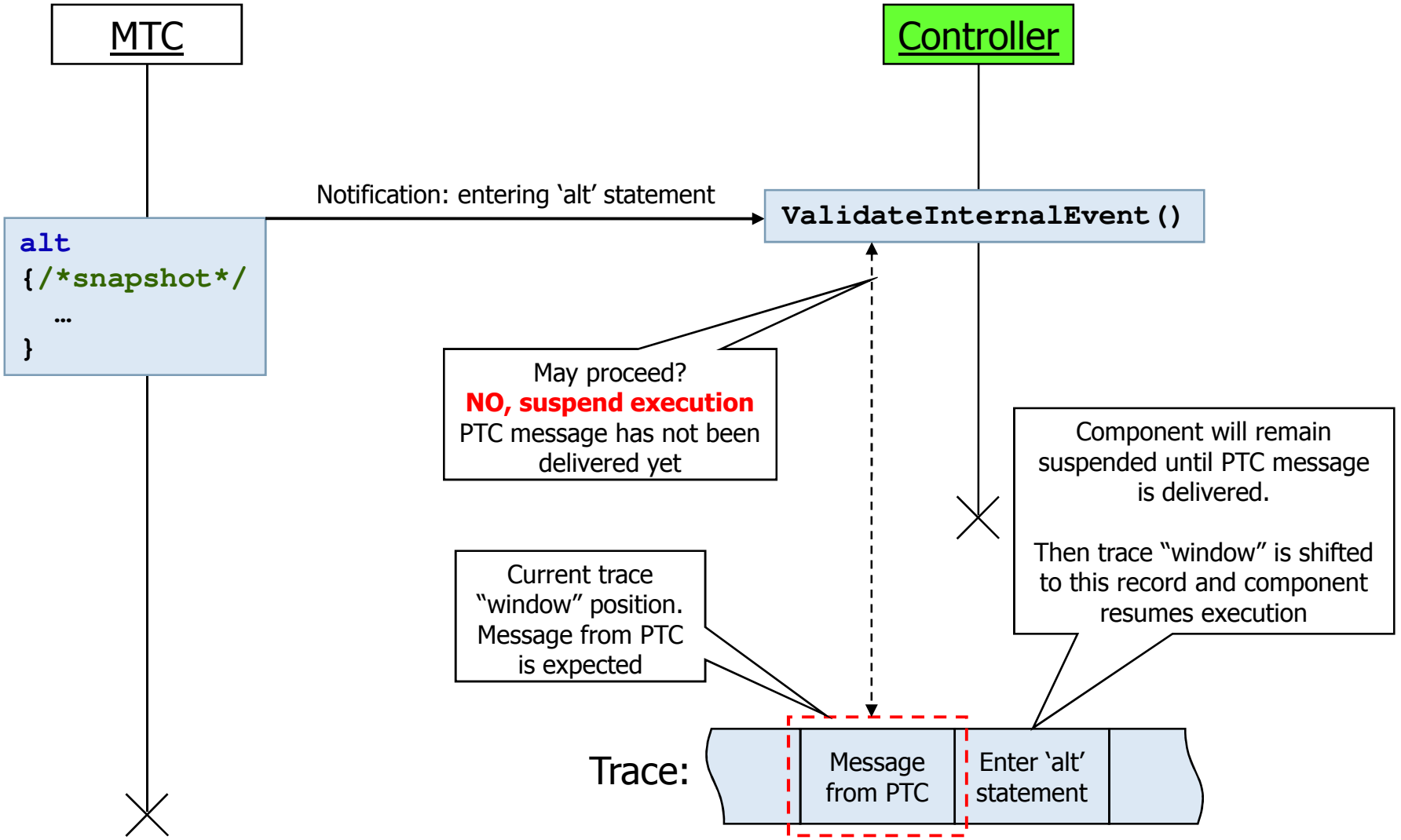
- Record/Replay controller interposes on runtime interactions between the TestExecutable and the Adaptation
- During the replay "external" events (injected by the Adaptation) are reordered and synchronized with the "internal" component events according to the trace
- The Adaptation component is not required in "Full Replay" mode



"External" event reordering

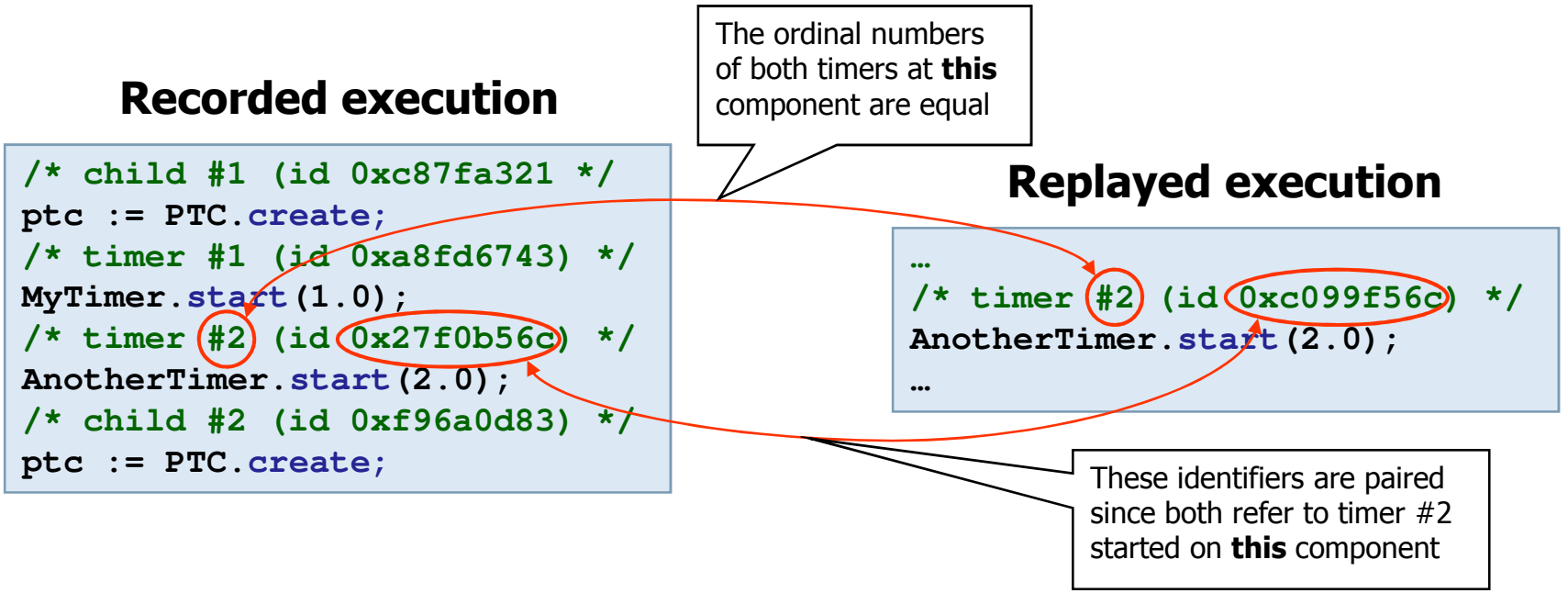


Snapshot reconstruction



Trace/live instance pairing

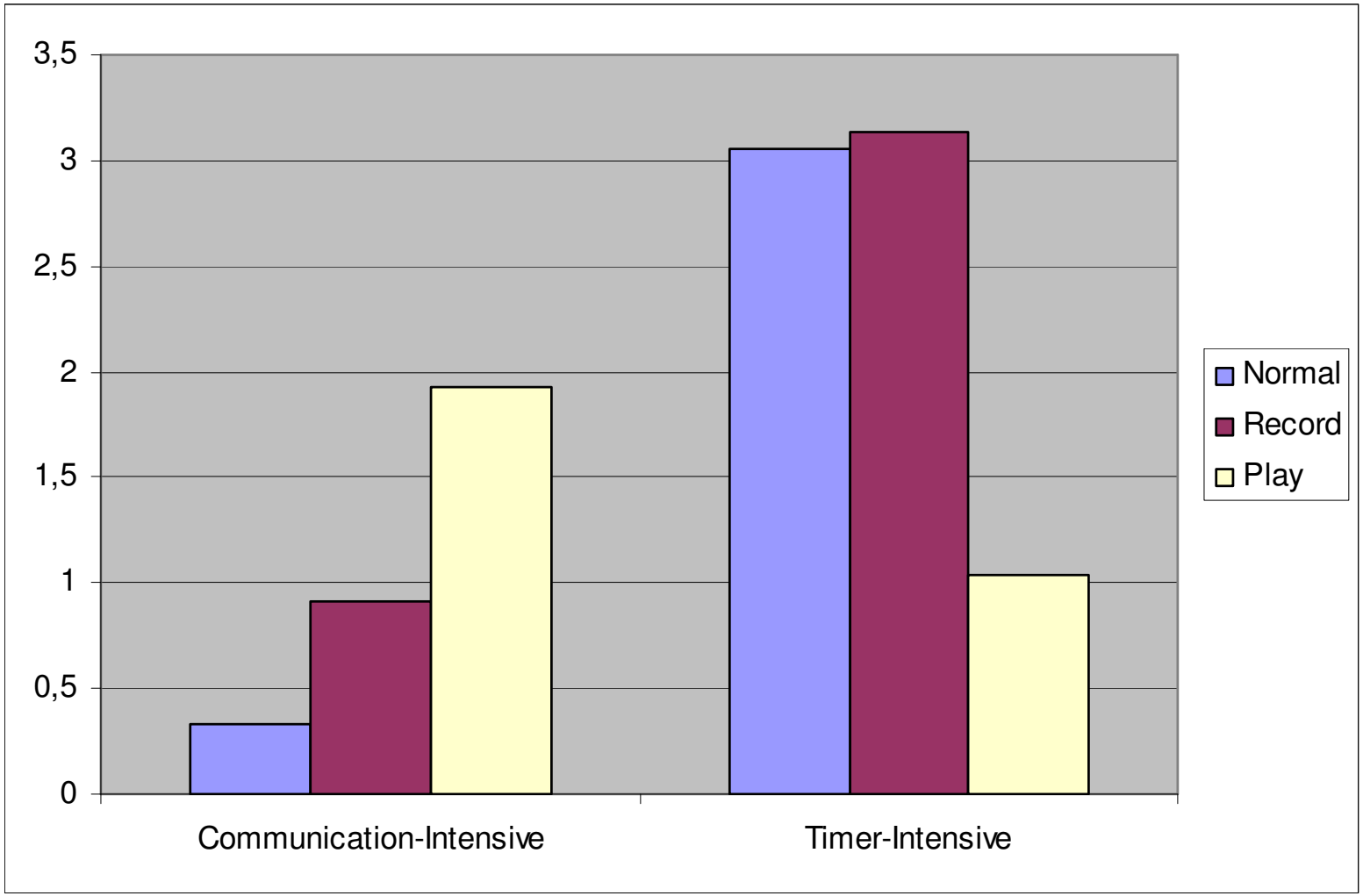
- Component and timer instance identifiers (binary strings) as visible on the Adaptation layer may differ between executions
  - They may not be directly used to validate trace events against the live events during the replayed execution
- Record/Replay controller tracks the order of creating child components and starting timers on each running component
  - A trace and a live instance identifier become paired if their parent components are paired and they have equal ordinal numbers of creation (since the component start)



External functions

- External functions may have arbitrary behavior
  - Generate random data
  - Read real time clock
  - Communicate through the network
- External functions may be the source of test case execution variations
- During the replayed execution the values for ‘out’ parameters and the return value are replaced with the data stored in the trace
- No actual external function invocation is required during the replay

Evaluation


3,5
3
2,5
2
1,5
1
0,5
0
Communication-Intensive
Timer-Intensive
Normal
Record
Play

ISP RAS

Demonstration

ISP RAS

Thank you for your attention!

Questions?