Checking for TLM-2.0 Compliance - Why Bother?

Dr. Andrea Kroll

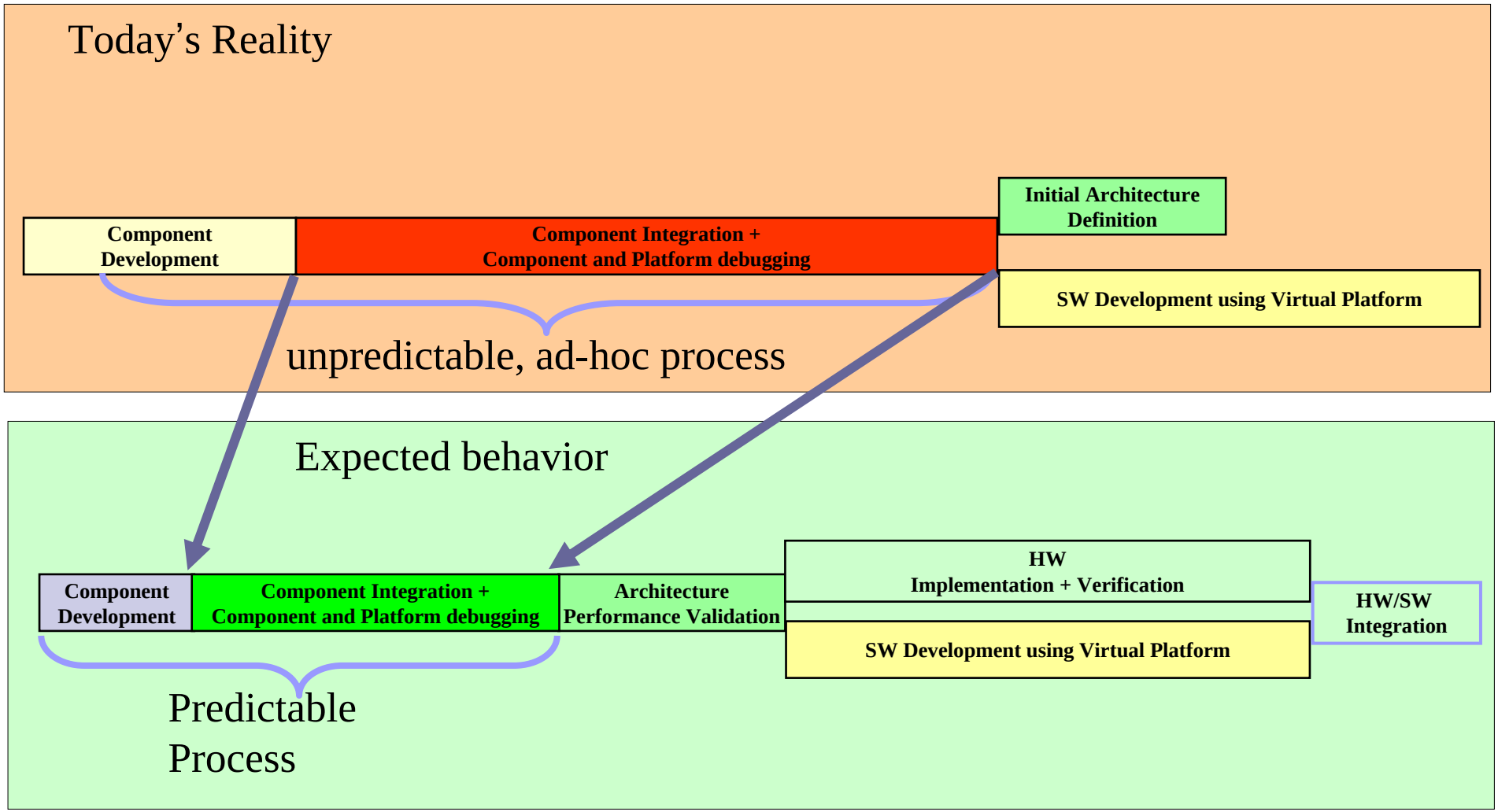
VP Marketing and Business Development

JEDA Technologies, Inc.

Agenda

- Model Integration Challenges
- TLM2.0 Rules – Beyond Transport Calls
- OSCI TLM2.0 Rule Violation Examples
- How to debug interoperability issues?
- Automated Compliance Checking and reports
- Summary

Model Integration Challenges



Modeling Case Study

Model Developer Impact

- **3 hour** average debugging time to narrow down a protocol related bug or to identify i
- inside a platfo
- JEDA checker)
- **1-3 days** deb
- inconsistencie
- (1 minute wit
- **30-40%** effo
- writing testbe

Timing impact assume 6-9 month project (24-36 weeks)

Model End User Impact

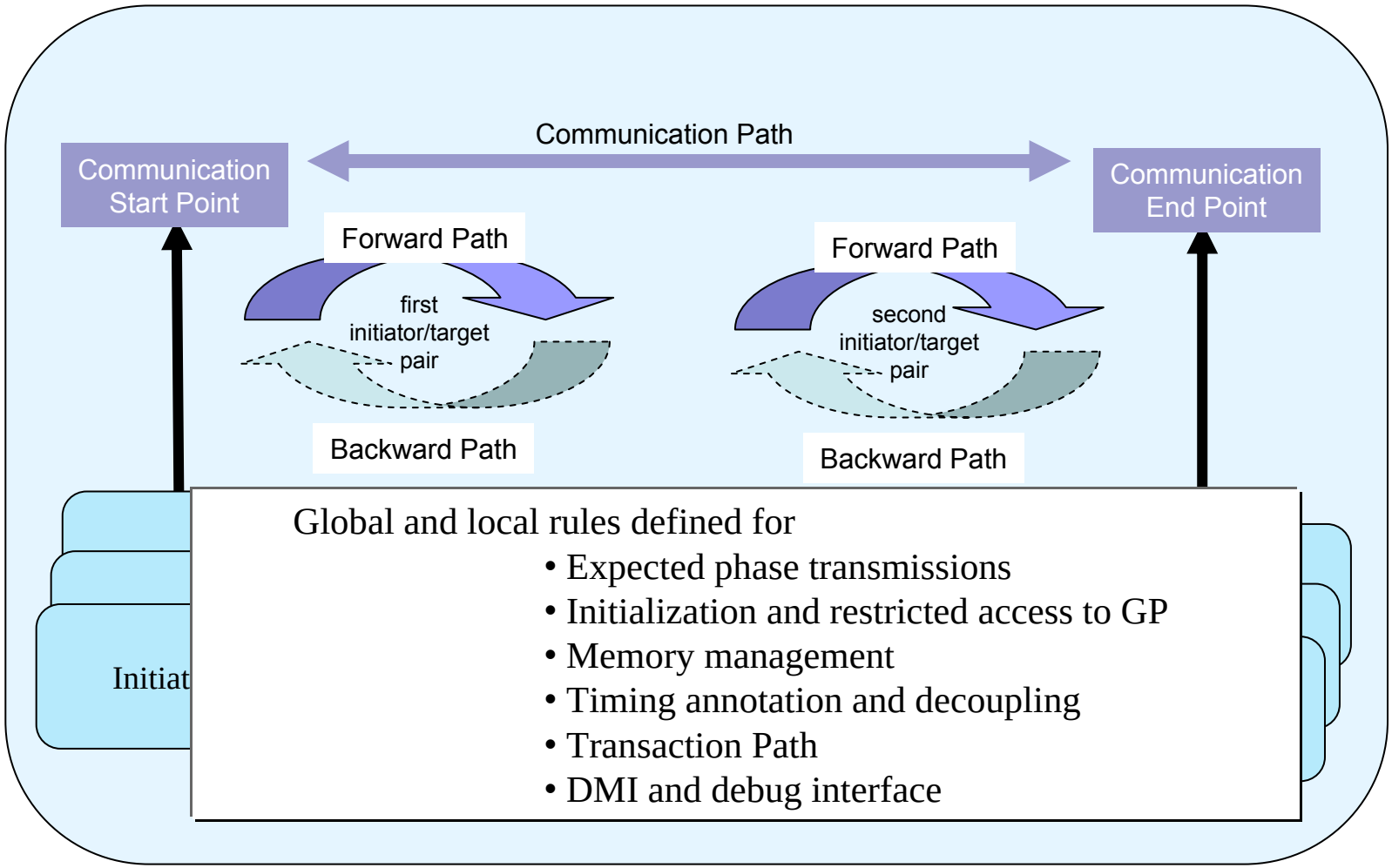
	Timing impact assume 6-9 month project (24-36 weeks)
1-2 weeks for installation and model bring-up (5-6 new release per year)	5-6 weeks
2-5 days average project delay per model bugs in models shipped to the user	6-7 weeks
1-2 weeks debugging time to identify interoperability/ synchronization issue	3-6 weeks
	14-19 weeks

> 50%

TLM2.0 Rules: The Goals

- Ensure interoperability between components from different sources
- Create safe data communication
- Base Protocol Rule
  - Ensure maximal interoperability between transaction level models of components that interface to memory-mapped buses.
  - Maximize coverage of features in today's interconnect architectures
  - Minimize need for adapters

TLM2.0 System Overview

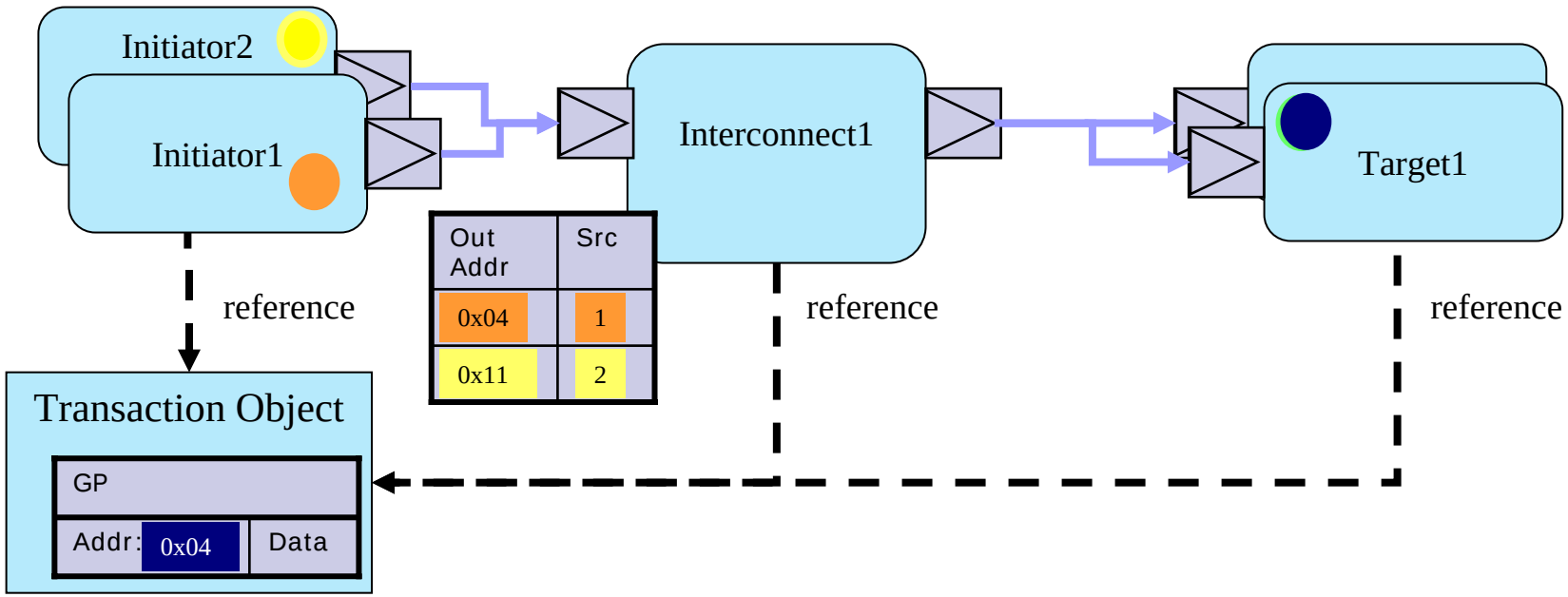


Generic Payload Rule Violation Example

Rule 6.7: default value and modifiability

Attribute	Default value	Modifiable by interconnect?	Modifiable by target?
Command	TLM_IGNORE_COMMAND	No	No
Address	0	Yes	No

(TLM2.0 user manual page 78)

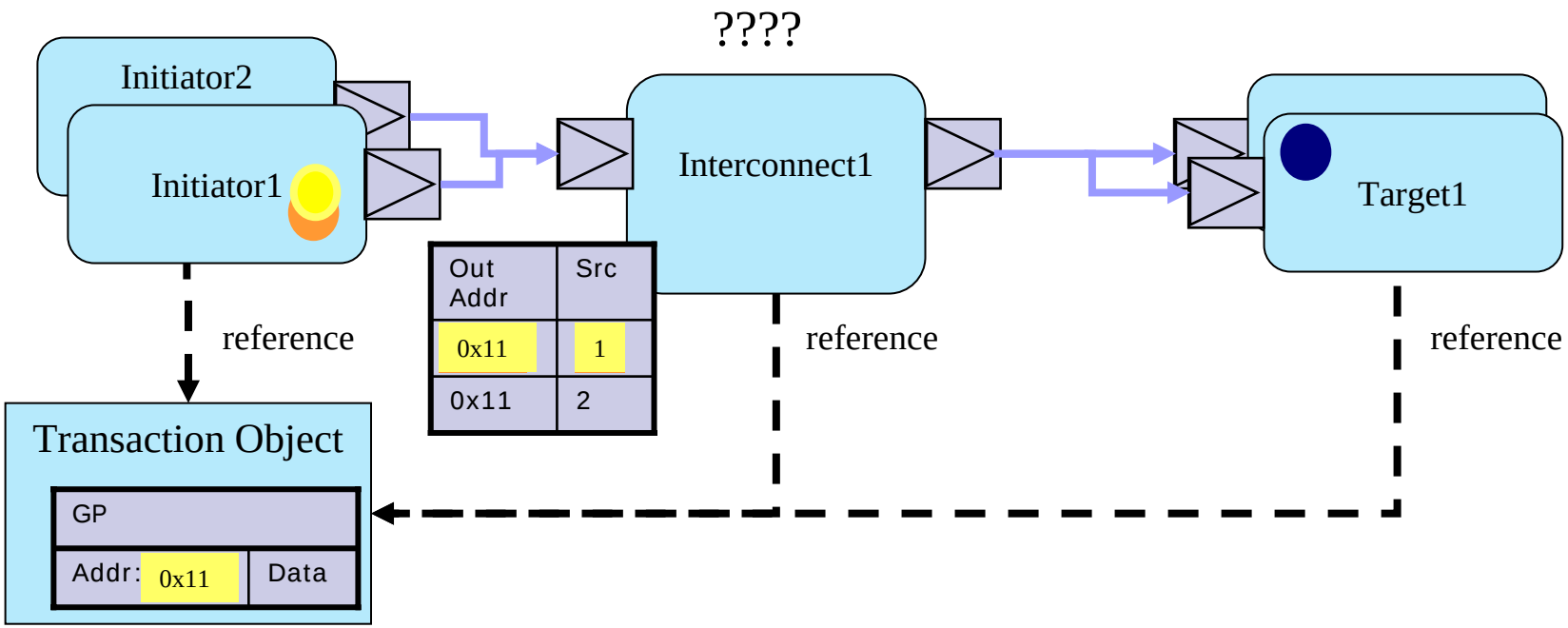


Base Protocol Rule Violation Example

- Rule 7.2.4b:

"For the base protocol, an initiator shall not start a new transaction through a given socket with phase BEGIN_REQ until it has received END_REQ or BEGIN_RESP from the target for the immediately preceding transaction"

(TLM2.0 User manual page 112)



Finding Interoperability Problems

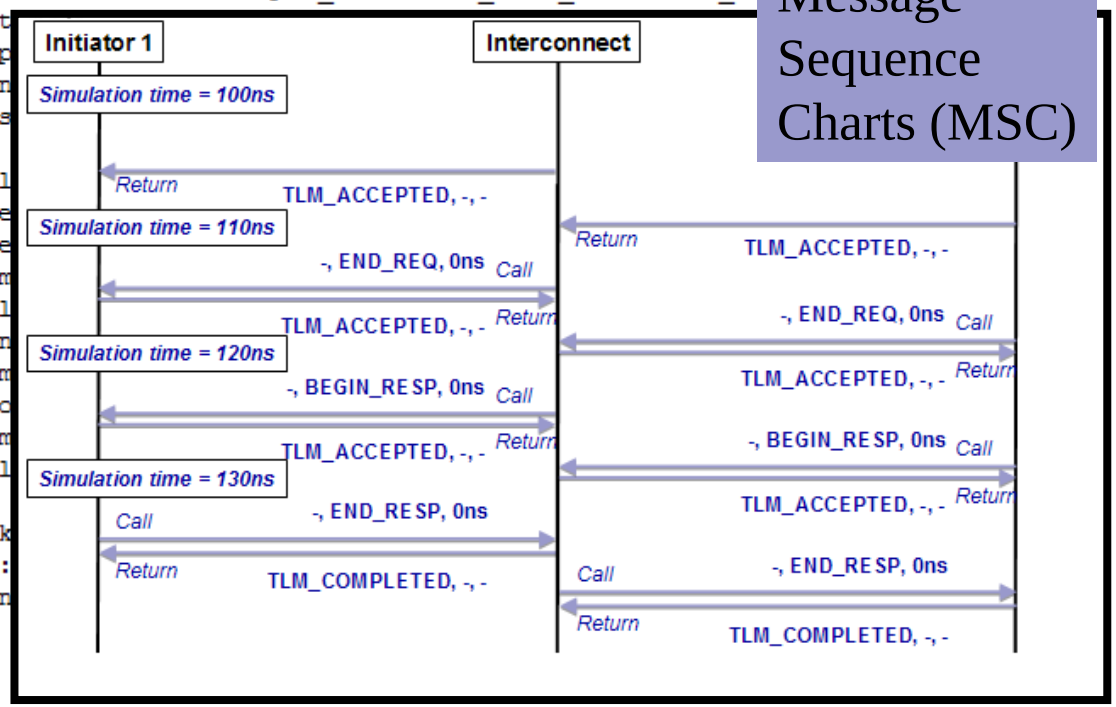
Transaction Trace

```
Info: traffic_generator.cpp: 0 s - traffic_generator_thread
    Initiator: 101 Starting Traffic

Info: traffic_generator.cpp: 0 s - traffic_generator_thread
    Initiator: 102 Starting Traffic

Info: select_initiator.cpp: 0 s - initiator_thread
    Initiator: 101 starting new transaction
    Initiator: 101 nb_transport_fw (GP, BEGIN_REQ, 0 s)

Info: select_initiator.cpp: 0 s - initiator_thread
    Initiator: 101 ACCEPTED (GP, BEGIN_REQ, 0 s)
    Initiator: 101 transaction waiting end-request on backward-path

Info: select_initiator.cpp: 0 s - initiator_thread
    Initiator: 102 starting new transaction
    Initiator: 102 nb_transport_fw (GP, BEGIN_REQ, 0 s)

Info: select_initiator.cpp: 0 s - initiator_thread
    Initiator: 102 ACCEPTED (GP, BEGIN_REQ, 0 s)
    Initiator: 102 transaction waiting end-request on backward-path

Info: at_target_4_phase.cpp: 0 s - nb_transport_fw
    Target: 201 nb_transport_fw (GP, BEGIN_REQ, 0 s)
    Target: 201 ACCEPTED (GP, BEGIN_REQ, 0 s)

Info: at_target_4_phase.cpp: 10 ns - end_request_method
    Target: 201 starting end-request method
    Target: 201 transaction moved to send-response PEQ
    Target: 201 nb_transport_bw (GP, END_REQ, 0 s)

Info: at_target_4_phase.cpp: 10 ns - end_request_method
    Target: 201 ACCEPTED (GP, END_REQ, 0 s)

Info: select_initiator.cpp: 10 ns - nb_transport_bw
    Initiator: 101 nb_transport_bw (GP, END_REQ, 0 s)
    Initiator: 101 transaction waiting begin-response on backward path
    Initiator: 101 ACCEPTED (GP, END_REQ, 0 s)

Info: at_target_4_phase.cpp: 10 ns - nb_transport_fw
    Target: 201 nb_transport_fw (GP, BEGIN_REQ, 0 s)
    Target: 201 ACCEPTED (GP, BEGIN_REQ, 0 s)
```

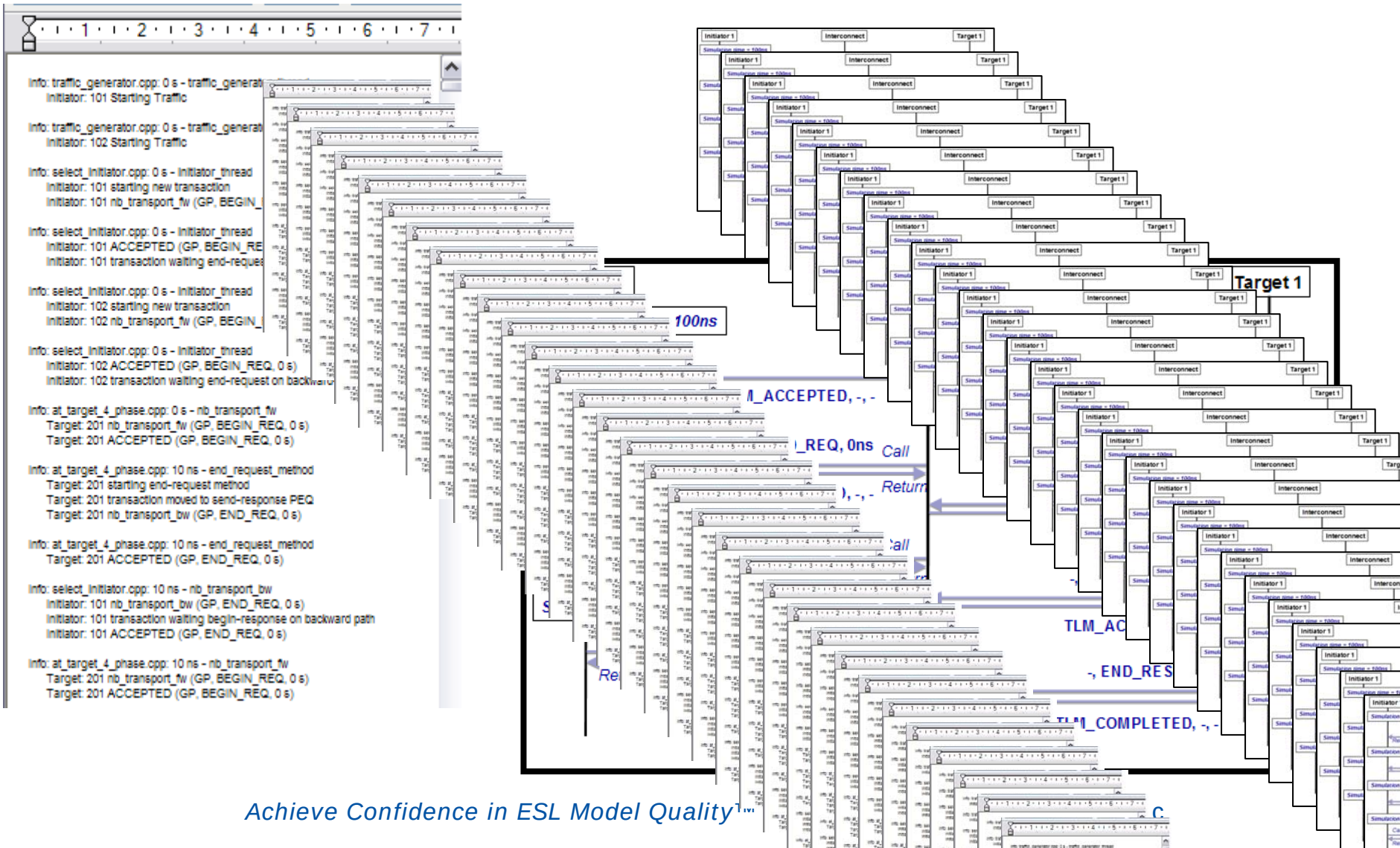
Verbose Transaction Trace

```
transaction details:
  Checker name:        top.initiator_checker1
  Initiator name:      top.m_initiator_1.at_initiator_s
  Target
  GP's p
  Comman
  Addres
  Data:
  Data l
  Byte e
  Byte e
  Stream
  DMI al
  Respon
  Has mm
  Ref co
  SC tim
  SC del
  Delay
  Proc k
  Phase:
  Action
```

Message Sequence Charts (MSC)

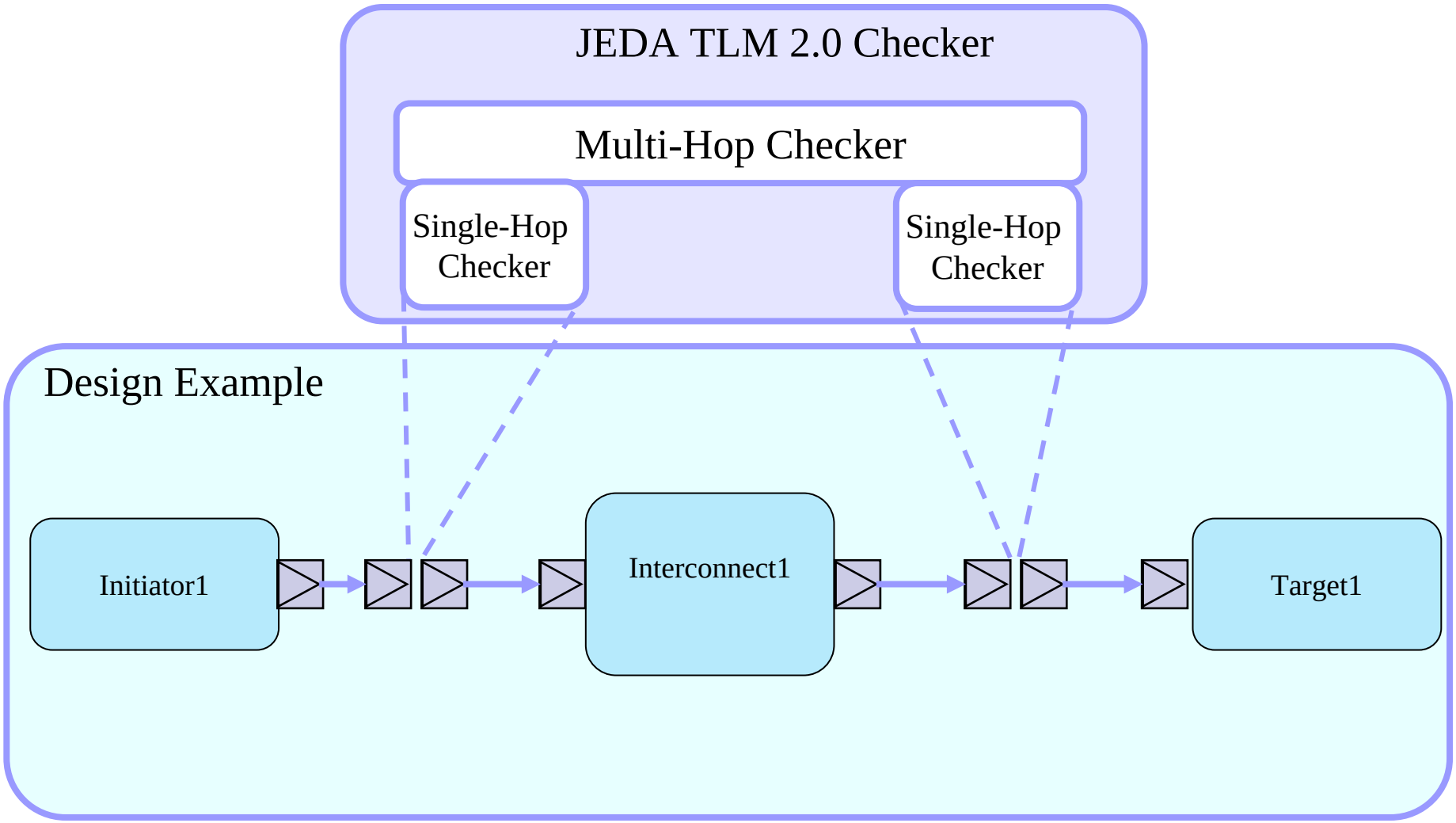


Finding Interoperability Problems



Achieve Confidence in ESL Model Quality™

Automated Compliance Checking



Checker Reports

JEDA TLM2checker error @ 755 ns, detected in checker top.target checker1

Error message:
Through top.m_bus.simple_initiator_socket_tagged_0, a new transaction with an incompleted previous one is sent by nb_transport_fw.

Expected behavior:
Each BEGIN_REQ shall be completed with an END_REQ, BEGIN_RESP, or the transaction is completed, before the next BEGIN_REQ is sent.

Refer to TLM-2.0 User Manual, version June 2008: 7_2_4_b.

Previous transaction details:

Checker name:	top.target_checker1
Initiator name:	top.m_bus.simple_initiator_socket
Target name:	top.m_at_target_4_phase_1.memory_
GP's pointer:	0x39a228
Command:	READ
Address:	0x100
Data:	0x399ba0

.....

Current transaction details:

Checker name:	top.target_checker1
Initiator name:	top.m_bus.simple_initiator_socket
Target name:	top.m_at_target_4_phase_1.memory

Detailed Error report

```
////////////////////////////////////////////////////////////////////////////////
// JEDA TLM2.0 Compliance CHECKER summary:
//
// Note:
// Rules Violated      : TLM2 rule violations are detected in simulation
// Rules Passed        : Rules are triggered in simulation and passed successfully
// Rules Not-triggered : Either because of not applicable or weak traffic generator
// Rule Disabled       : Rules are disabled by user in configuration.
// Rule levels         : 1 = continue with error
//                       2 = stop only on max errors
//                       3 = stop on error
//
////////////////////////////////////////////////////////////////////////////////

 * ------------------------------------------------
 * Rules Violated:
       Rule Name        Rule Level        Failed count
       4_2_5_a             1                 1
       4_2_5_f             1                 1
       4_2_5_s             1                 1
       4_2_5_aa            1                 2

 * ------------------------------------------------
 * Rules Passed:
       Rule Name        Rule Level
       4_1_1_4_b           1
       4_1_1_4_e           1
       4_1_1_4_h           1
```

Summary Report

Summary

- TLM2.0 Compliance is critical for interoperability and safe communication
- Quickly identify interoperability problems with compliance checking
- Reduce risk for model bring-up and debugging time with automatic interoperability error detection

TLM2.0 is real and ready for commercial adoption