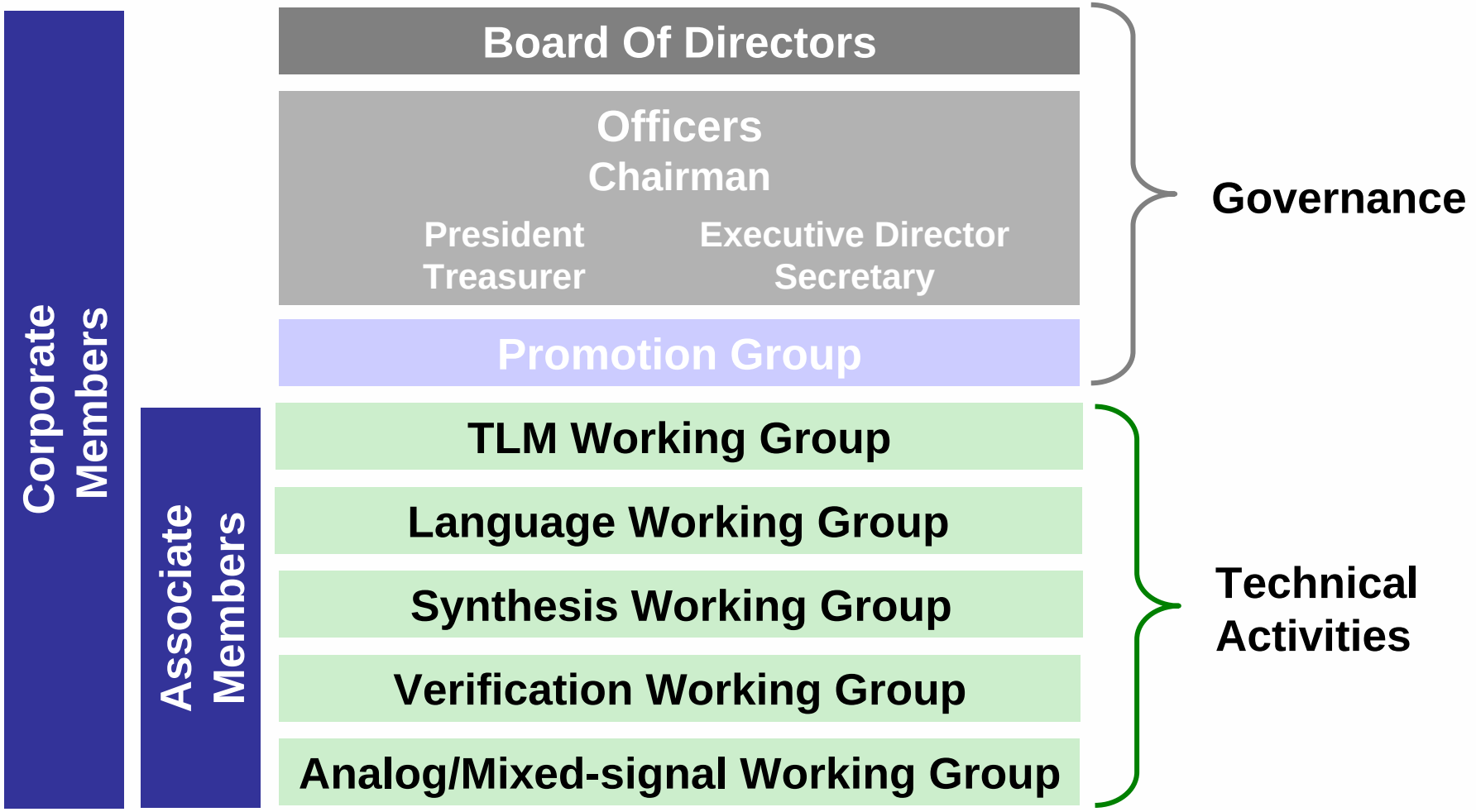
SystemC Community Update

September 18, 2006

OSCI Organization



OSCI Membership

- **Corporate Members**
  - ARM Ltd
  - Cadence Design Systems
  - CoWare
  - Forte Design Systems
  - Intel Corporation
  - Mentor Graphics
  - NXP
  - Synopsys
  - Summit Design Inc
  - ST Microelectronics

- **Associate Corporate Members**
  - Actis
  - Atrenta, Inc.
  - Bluespec
  - Calypto Design Systems
  - Carbon Design Systems
  - Celoxica Ltd
  - Chipvision Design Systems
  - Denali
  - Doulos
  - ESLX Inc.
  - Freescale
  - Fraunhofer Institute for Integrated Circuits
  - GreenSocs
  - Jeda Technologies
  - NEC Electronics
  - SpiraTech Ltd.
  - Springsoft
  - Synfora Inc
  - Tenison Technology EDA Ltd
  - Vast Technologies

30 member companies total
8 new since last year

OSCI Board of Directors and Officers

- OSCI Board Members
  - ARM – Nizar Romdhane
  - Cadence - Stuart Swan
  - CoWare - Pat Sheridan
  - Forte - Mike Meredith
  - Intel – Ken Tallo
  - Mentor - Mark Glasser
  - Philips - Ralph von Vignau
  - STMicroelectronics - Alain Clouard
  - Synopsys – Marcus Willems
- OSCI Officers
  - Chairman, Alain Clouard
    - alain.clouard@ST.com
  - President, Mike Meredith
    - mmeredith@ForteDS.com
  - Executive Director, Pat Sheridan
    - psheridan@CoWare.com
  - Secretary, Paul Tauber (Legal counsel)
    - PJT@cpdb.com
  - Treasurer, Stan Krolikoski
    - stank@chipvision.com

Significant OSCI Achievements This Year

- Approval of IEEE 1666™-2005 standard for SystemC
- Availability of IEEE 1666™-2005 LRM on IEEE web site without charge to users
- Release of SystemC 2.1v1 open source proof-of-concept library
- Release of SCV 1.0p2 verification library
- Public review of SystemC 2.2 library
- Public review of Synthesizable Subset document
- ~30% increase in number of member companies

SystemC Language is IEEE 1666™-2005

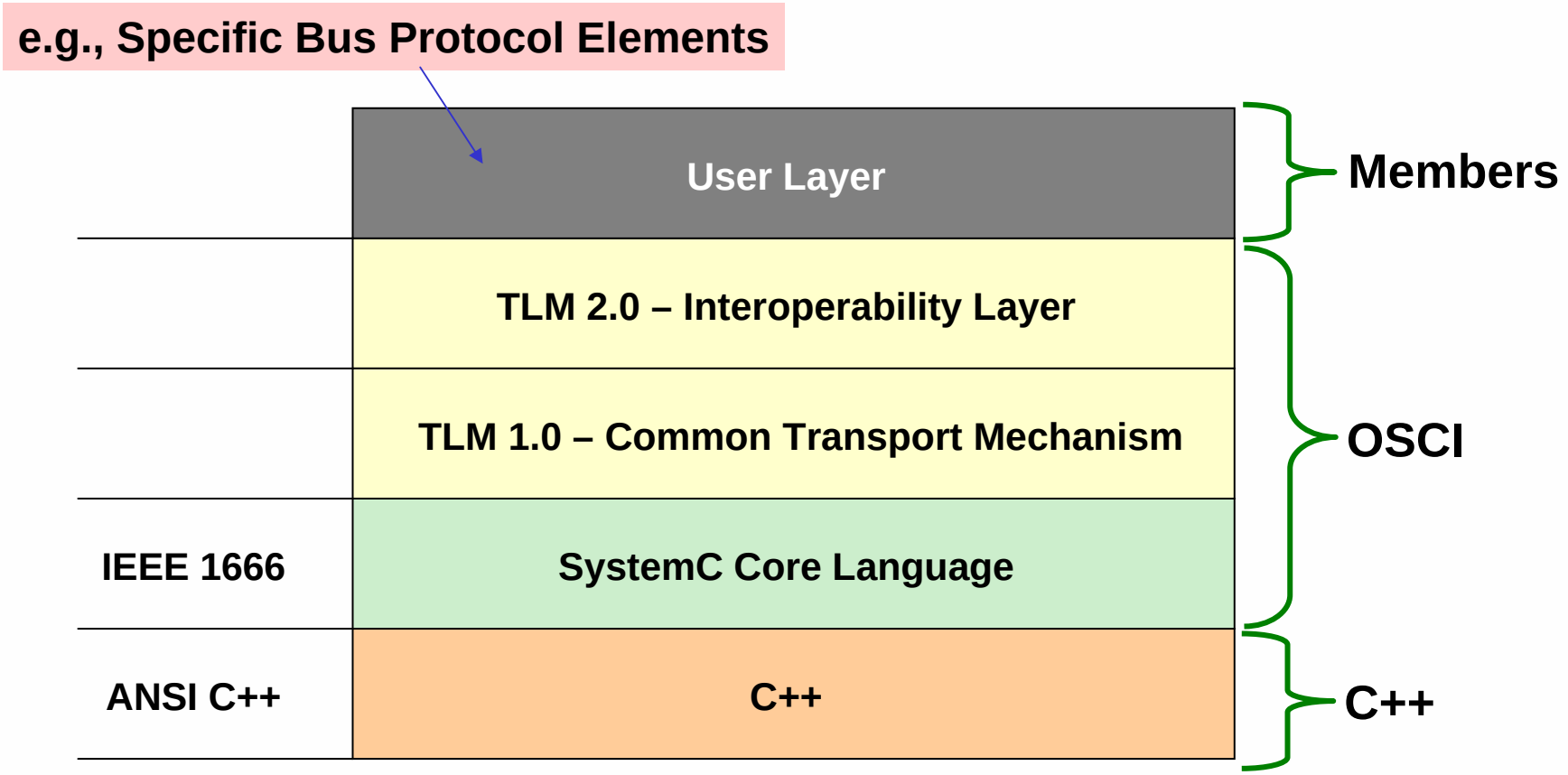
- Approved by IEEE on Dec. 6, 2005
- Partnership between OSCI and IEEE makes LRM available without charge to users
  - http://standards.ieee.org/getieee/1666/index.html
- Current OSCI open source proof-of-concept library 2.1v1 very close to IEEE 1666™ compatible
- OSCI library version 2.2 available in draft form for public review removes remaining known incompatibilities with IEEE 1666™-2005

Activity in Working Groups

Working Group Activity

- Language Working Group
  - SystemC 2.2 draft available for public review
  - Repairs all known incompatibilities with 1666
- Synthesis Working Group
  - Synthesizable subset document draft available for public review
- AMS Working Group
  - Newly formed
  - Defining Analog/Mixed-signal extensions for SystemC
  - Meeting at FDL
    - Wed, 13:00 Room E215
- Verification WG
  - SCV 1.0p2 released
    - Functionally the same as 1.0p1
    - Compatible with SystemC v2.1 v1
    - Compatible with SystemC 2.2 (draft)
  - Work underway for future SCV releases
    - Temporal Assertion Support
      - TLM assertions
      - Signal-level assertions
      - Gathering requirements
      - Reviewing proposals

TLM Layered Standards



Active OSCI TLM WG Members

Having an expert participating in most TLM WG conference calls :

- **ARM**
- **Bluespec**
- **Cadence**
- **ChipVision**
- **CoWare**
- **Doulos**
- **ESLX**
- **Forte**
- **GreenSocs**
- **Intel**
- **Mentor**
- **Philips**
- **SpringSoft**
- **ST**
- **Summit**
- **Axel Braun - Tuebingen**

Involvement of other organizations:
OCP
There is significant intersection of membership between the two organisations
Technical Chair of OCP SLDWG is a TLMWG member
SPIRIT
There is significant intersection of membership with SPIRIT

What are we working on ?

- **Minor release of existing TLM 1.0 kit (bug fixes)**
- **Standard Bus Modeling APIs**
  - **Generic PV**
  - **Generic PVT**
  - **Interrupt Modeling**
  - **Memory Map Services**
  - **Memory / Register Modeling**
- **Standard Configuration and Control APIs**
  - **Configuration Interface**
  - **Debug Interface**
  - **Analysis Interface**

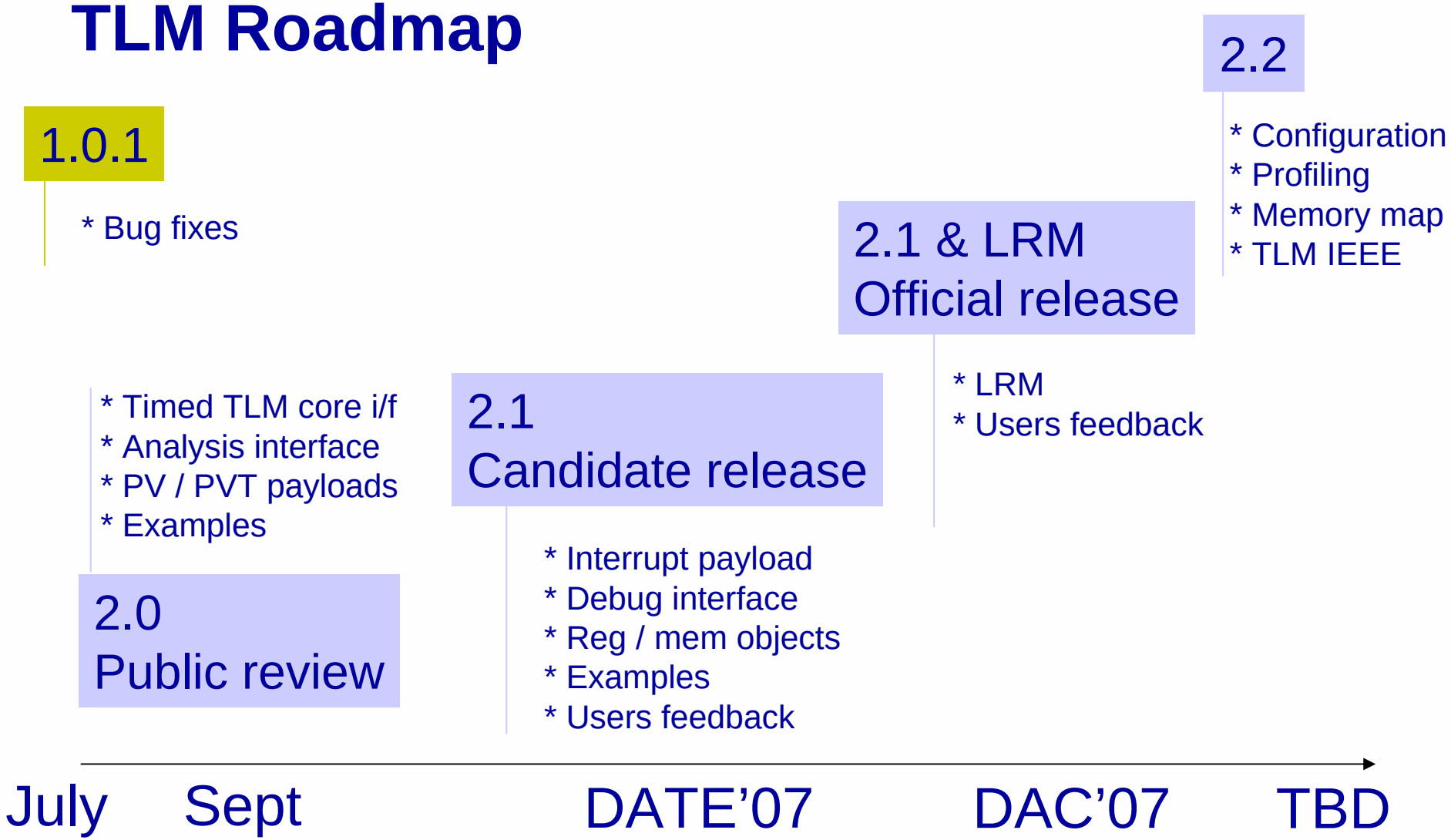
How do we move transactions about ?

What transactions do we move about ?

How do we ***control*** and ***analyse*** the transactions moving through the TLM ?

SYSTEMC™

TLM Roadmap


1.0.1
* Bug fixes
2.0
Public review
* Timed TLM core i/f
* Analysis interface
* PV / PVT payloads
* Examples
2.1
Candidate release
* Interrupt payload
* Debug interface
* Reg / mem objects
* Examples
* Users feedback
2.1 & LRM
Official release
* LRM
* Users feedback
2.2
* Configuration
* Profiling
* Memory map
* TLM IEEE
July
Sept
DATE'07
DAC'07
TBD

TLM 1.0 Principles

- **OSCI TLM standard intends to support efficient and safe exchange of transactions between SystemC modules**
- **TLM 1.0 defines core interfaces to transfer transactions between modules:**
  - ***Based on message-passing scheme***
  - ***transport()* for bi-directional information exchange with one single IMC**
  - ***put()*, *get()*, *peek()*, *poke()* variants for unidirectional information transfer**
  - **Copy-by-value mechanism:**
    - ***transport() and put(): "const &" for interface parameters***
    - ***transport() and get() : data sent back to callee as return value***

Improving on TLM 1.0

- Agreeing on the message format
- Efficiency for large payloads
- Annotating the timing

TLM 2.0 principles

- ***TLM 2.0 maintains message-passing "philosophy" with optimizations for PV and PVT use models***
  - ***PV: pointers for efficiency. Safety ensured with rules on usage of the "PV protocol"***
  - ***PVT: effective pass-by-value, with optimizations based on lazy-write type of algorithms (copy-on-write). Safety ensured by automatic copy of TLM transaction***
  - ***Both PV and PVT inherit from a base class for common attributes***

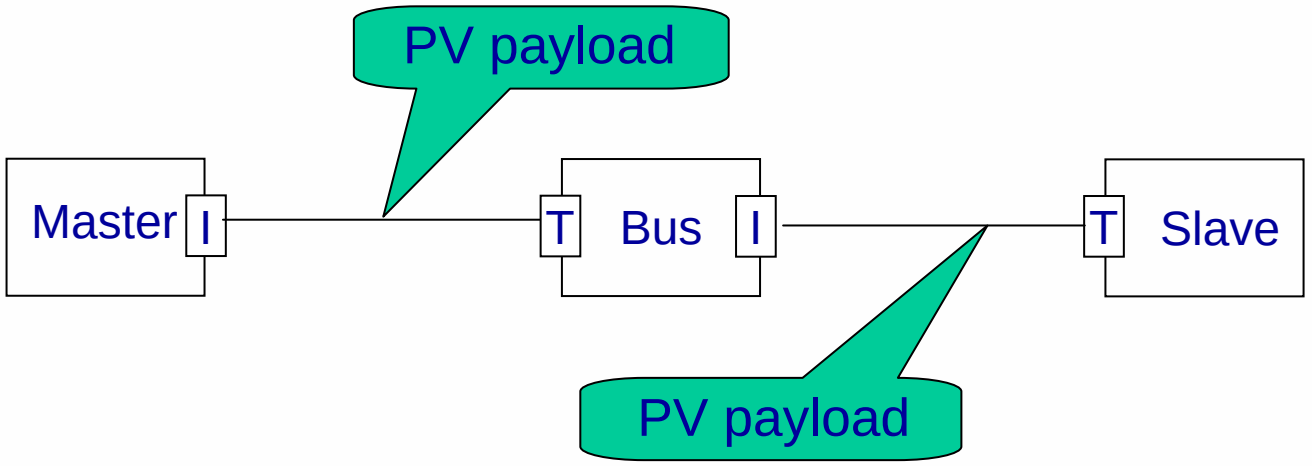
- ***TLM 2.0 adds capability to connect analysis code to TLM port for monitoring, scoreboarding, etc.***

September release: Overview

- **Untimed TLM modeling**
  - **Generic PV payload**
    - **Mostly for abstract modeling of transactions over on-chip bus**
- **Timed TLM modeling**
  - **Update of core interfaces for timed modeling**
    - **Mostly sc_time parameter to core interfaces**
  - **Generic PVT payload**
    - **Mostly for performance modeling of communication over on-chip bus, taking into account pipelining**
- **Analysis interface**
  - **To monitor TLM ports**

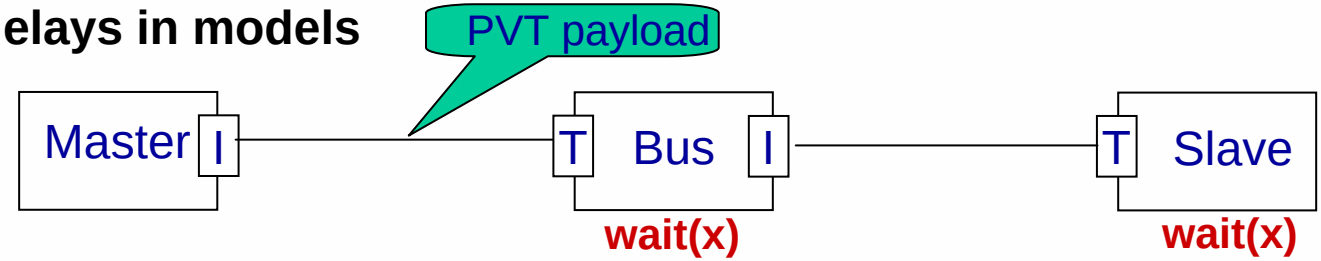
Untimed TLM modeling

- **Based on transport core TLM interface**
- **With PV payload**
- **All models using the PV payload and transport core interface can be connected and simulated together**

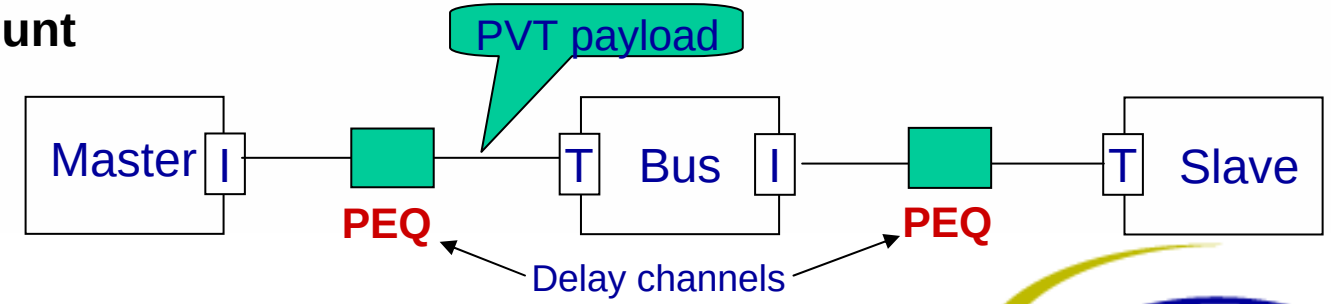


Timed TLM modeling

- **Enable timing annotations of event-based simulations**
- **With PVT payload, based on put/get core TLM interfaces**
- **All models using the PVT payload and put/get core interfaces can be connected and simulated together**
- **2 structural approaches supported:**
  - **1) Insert delays in models**

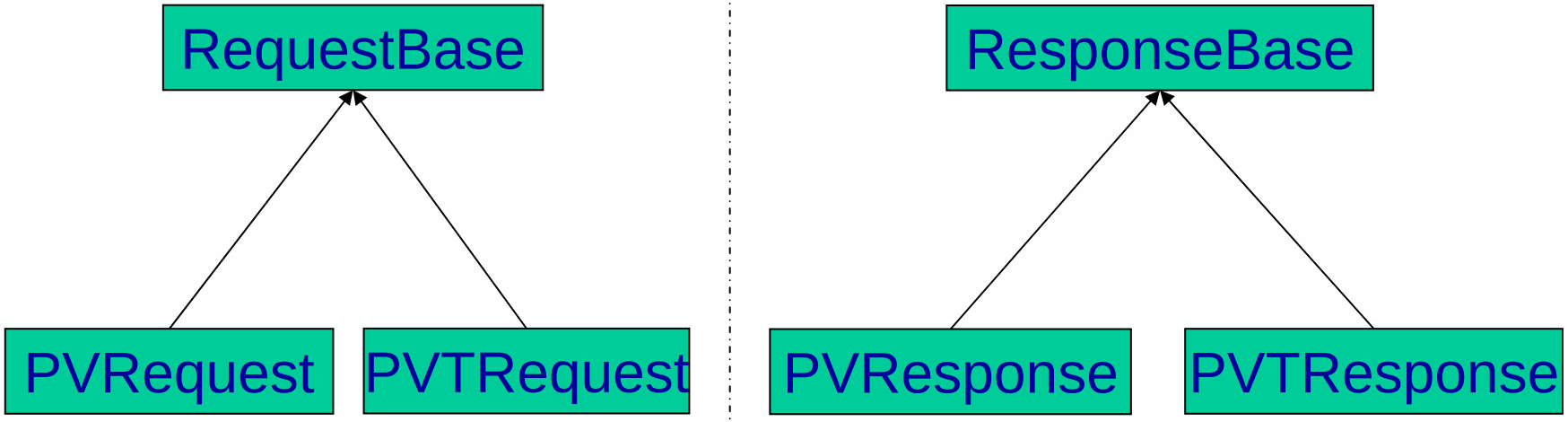


  - **2) Rely on a "delay channel" between components to take timing delays into account**



Standard payloads structure

- **Intend to provide generic support for bus-based communication (ie. memory-mapped systems)**
- **PV and PVT payloads share a significant subset of attributes:**



Payload Structures

RequestBase Structure

Command
Mode
Address
Metadata
Block_mode
Block address incr
Master Thread ID
Transaction Id
Custom Request Base *

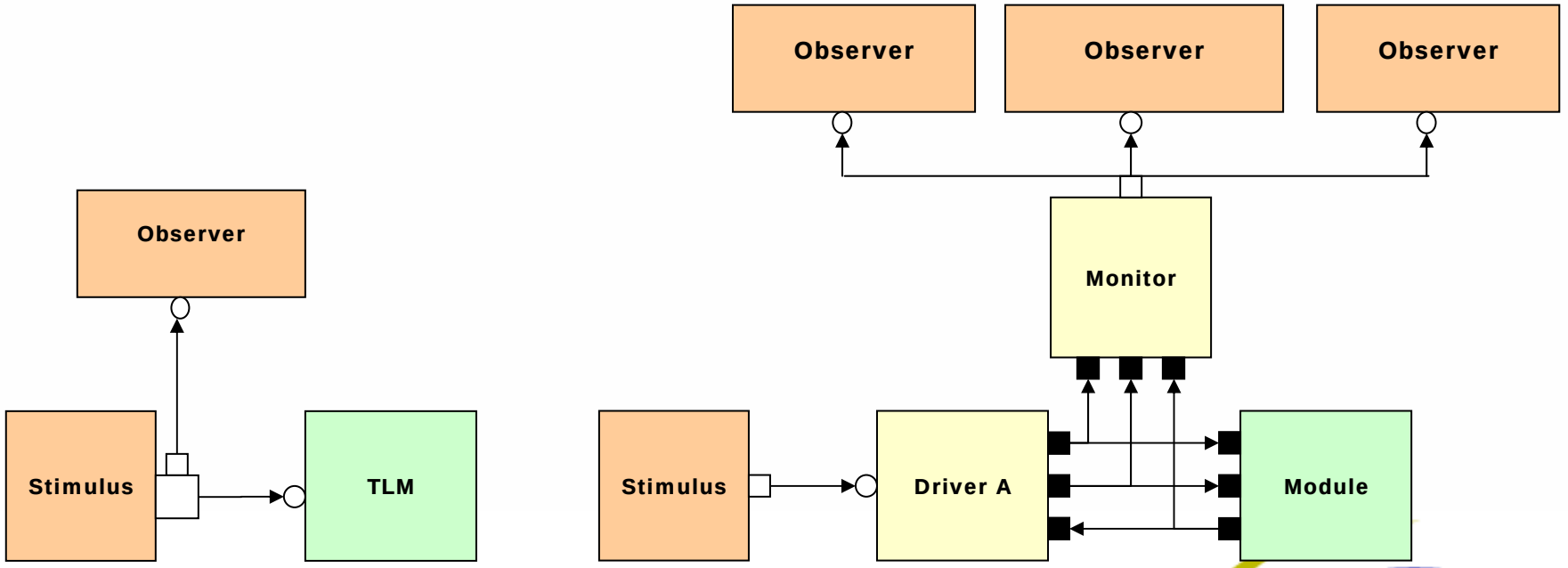
ResponseBase Structure

Status
Metadata
Error reason
Transaction Id
Custom response base*

PV and PVT specializations add Data and abstract info for bursts etc.

Analysis Ports

- Non-intrusive monitoring of transactions going through TLM ports
  - Essentially, this is a SystemC implementation of the observer pattern
- Main features
  - Possibility to connect zero, one or many observers to a single analysis port
  - Non blocking interface
  - Possibility to use the same port and interface for RTL monitors and TLM level communication



Conclusion

- TLM 1.0 provided foundation for TLM modeling
- TLM 2.0 specifies PV and PVT interfaces, enabling model interoperability for untimed and timed TLM models. Kit also provides mechanism for analysis of transactions
- TLM WG will address remaining required features defined in roadmap. Next high priorities: debug and cycle-accurate modeling style
- Join and contribute!
- Start working with first kit soon!


SYSTEMC™
THANK YOU