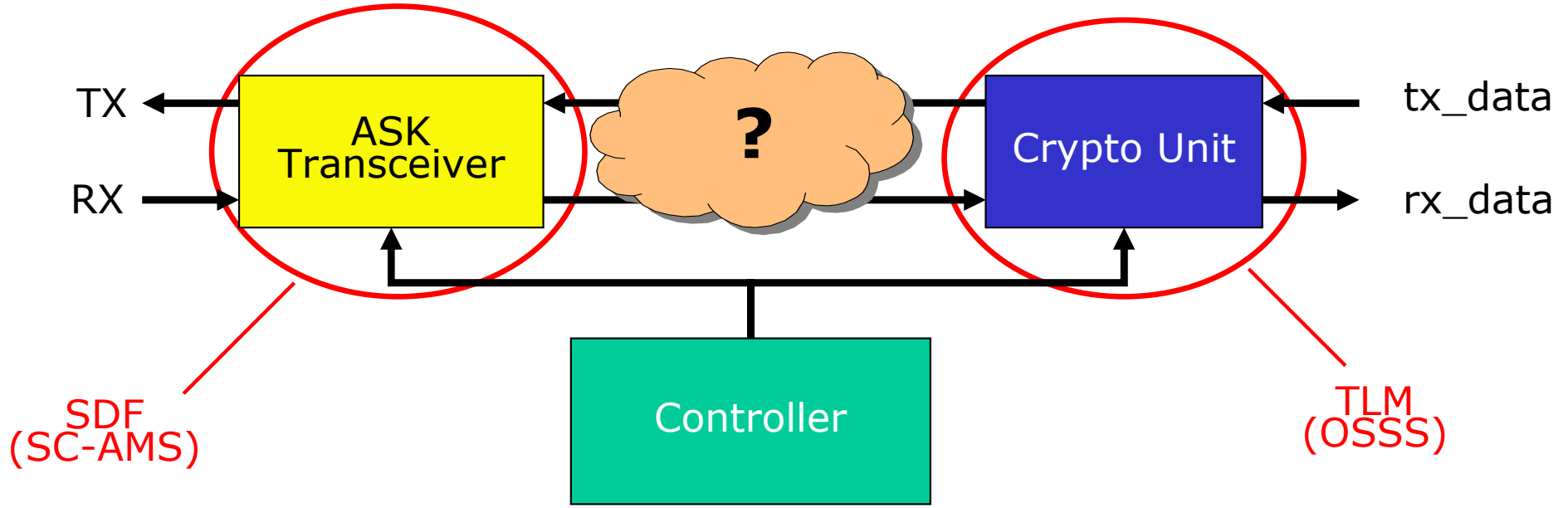
Interfacing cycle-accurate TLM Models and AMS SDF Clusters in SystemC
A Case Study

Andreas Herrholz
OFFIS
Oldenburg, Germany

Motivation: ASK Transceiver

- Case study used in ANDRES project to demonstrate issues and problems in heterogeneous system design
- Has been modeled using SystemC, SystemC-AMS and OSSS
- Combines SDF-models with cycle-accurate TLM-Models



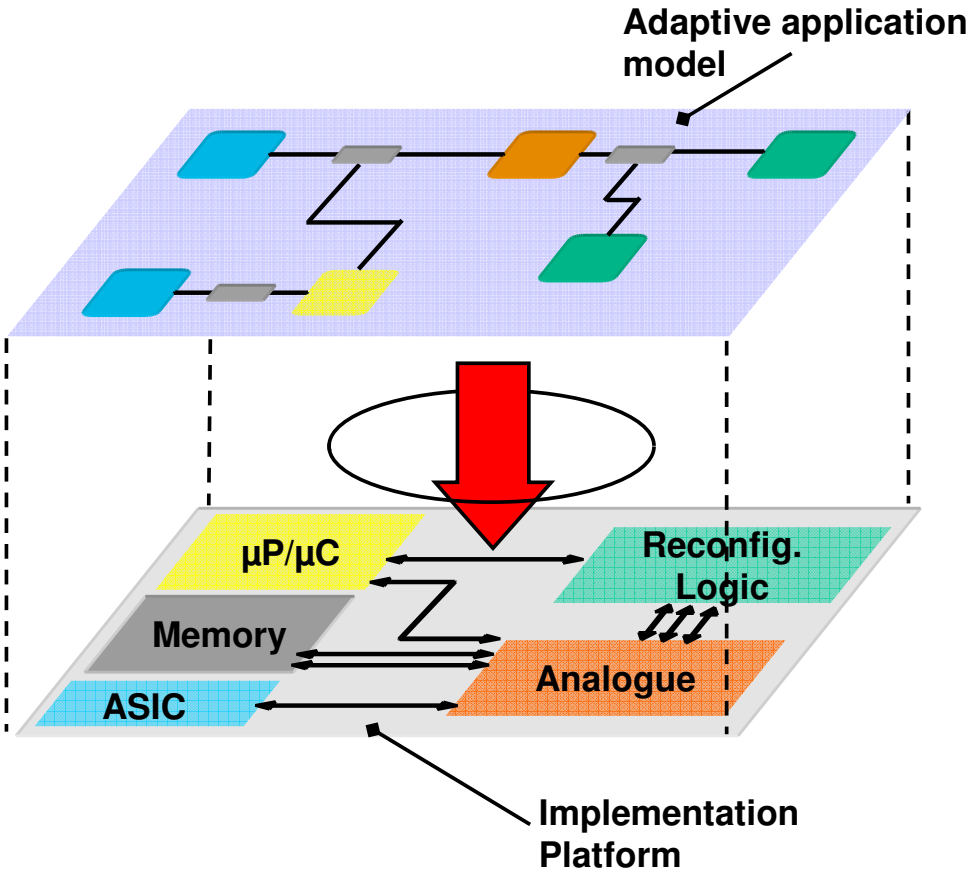
Outline

ANDRES

- Motivation
- The ANDRES Project and OSSS
- Case study scenarios
- Issues and Problems
- Outlook
- Summary

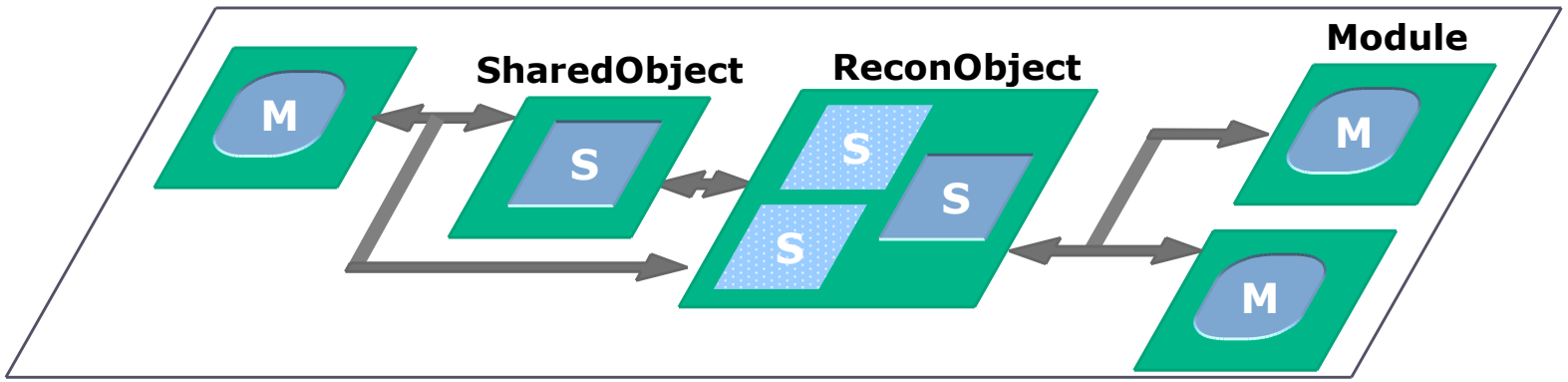
The ANDRES Project

- FP6 STREP (http://andres.offis.de)
- Goals
  - SystemC based framework for modelling, analysis and synthesis of adaptive heterogeneous embedded systems
  - Modelling of software, analog hardware and reconfigurable digital hardware components
  - Developing tools for automatic synthesis of reconfigurable digital hardware
- Integrates three SystemC based modelling libraries: OSSS(+R), SystemC-AMS and HetSC



OSSS (Oldenburg System Synthesis Subset)

- Extends synthesisable subset of SystemC 2.2.0 (IEEE 1666)
- Object-oriented modelling of synthesisable HW/SW-Systems
  - Cycle-accurate models using implicit FSMs (SystemC CThreads)
  - Concurrent access to shared resources is modelled via specialized slave modules (SharedObject, ReconObject)
  - Communication between modules is modelled using method calls
  - Logical connections between modules can be refined to busses and point-to-point-connections



Where is the TLM in OSSS?

ANDRES

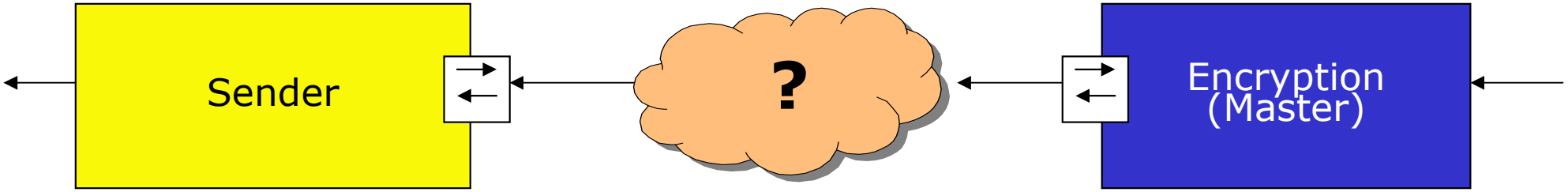
- OSSS is TLM
  - Communication is modelled via blocking bidirectional method-calls
  - Method-interfaces can be user-defined
  - Modules are synthesisable cycle-accurate TLM-models
  - Method-calls can be mapped to lower-level bus transactions via synthesisable transactors

  → OSSS enables synthesisable (cycle-accurate) TLM

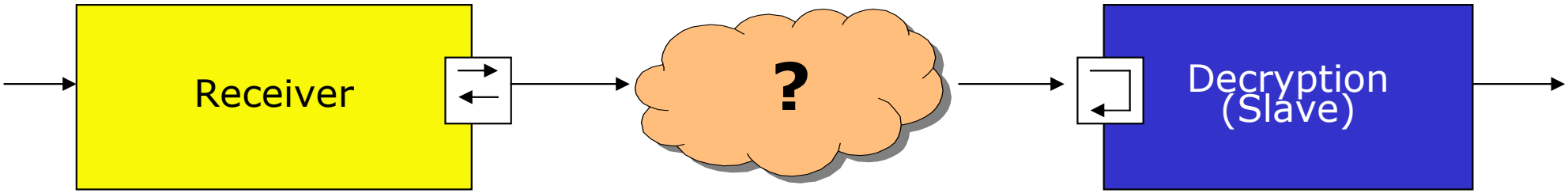
- OSSS is not OSCI TLM, but
  - Method-Interface can be restricted to OSCI TLM compatible methods (blocking put(), get() and transport())
  - Every user-defined method can be mapped to OSCI TLM (using a remote method invocation (RMI) protocol)

Two Use Cases

- Original use case can be divided into two separate cases
  - Interfacing a TLM master (encryption module) with SDF-cluster (Sender)

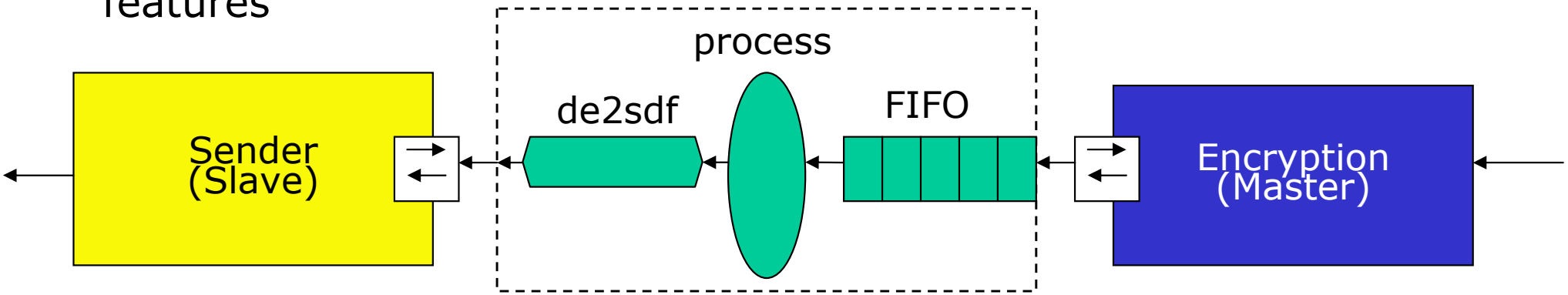


  - Interfacing a TLM slave (decryption module) with SDF-cluster (Receiver)



1st Case : TLM Master & SDF cluster

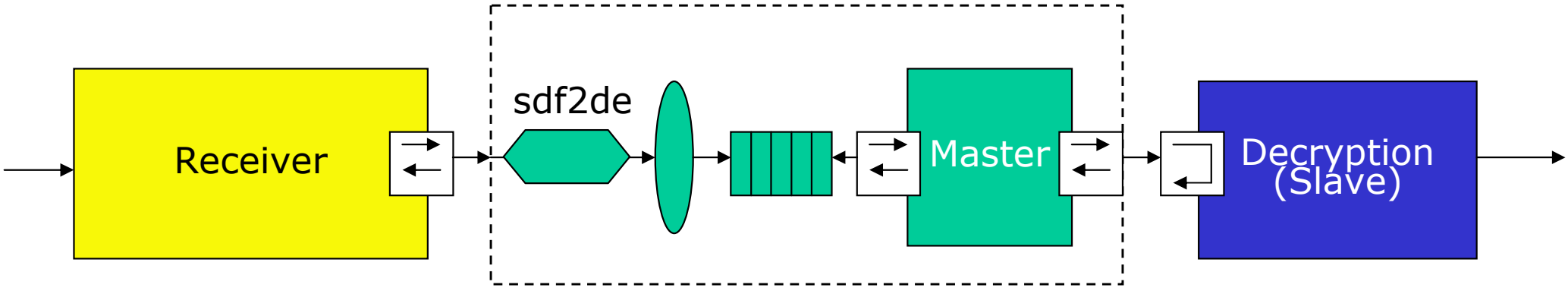
- SDF cluster becomes TLM slave
- Every frame encrypted by the *Encryptor* should be transmitted by the *Sender*
  - → every frame needs to be converted to an input token
- TLM model of the encryptor does not have a constant output rate!
  - → Models need to be decoupled → use a FIFO
  - → if output rate of encryptor is slower than input of sender (FIFO runs empty), define a default value or throw an exception
- All of this can be done using standard SystemC and SystemC-AMS features



2nd case: TLM Slave and SDF cluster

ANDRES

- More complicated than first case, as SDF-cluster cannot become TLM master
  - → Need to insert TLM master, converting tokens to transactions
- Receiver's output frames have to be sampled by separate process and written into FIFO
  - →Sample period needs to be the same as the SDF output period!
  - →Alternative: make process sensitive to sdf2de (need sc_buffer semantics)
- TLM master gets data from FIFO and calls interface method of slave



Issues and Problems

ANDRES

- Approach requires much knowledge of the system and its internal functions and timing behaviour
  - e.g. types and sizes of FIFOs (bounded or unbounded) , sampling rates, periods, define how to handle exceptions
- Manually coded approach
  - → can generate a lot of coding overhead
  - → may tend to be very error-prone
- Creates additional modules, process and context switches
  - → may slow down simulation speed
- Does only work for cycle-accurate TLM

What about timing relations?

ANDRES

- Timing relation between SDF and cycle-accurate TLM is not a big issue
- Both have defined constant period
  - Cycle-accurate TLM has a fixed duration of a cycle (clock period)
  - SDF has fixed period T for duration between two firings

→ Timing relation between both is also fixed

→ However, this does not solve the issue of non-constant output rates of TLM

- There are of course bigger issues for approximately and loosely timed TLM (see OSCI TLM 2.0)
- Not yet in the scope of ANDRES, but will become topic if OSSS is extended to non strictly timed/clocked models

Outlook

ANDRES

- ANDRES project develops *Converter Channels* to ease connection of different Models of Computation
- Converter Channels could be extended to support interfacing of SDF and TLM
  - Needs to be flexible to cope with different TLM timing styles
  - Must be configurable to adapt to different application cases (types of FIFOs, exception handling, sampling rates, …)
  - Should support at least OSCI TLM 1.0 Core interface
  - Should be easy to use (reduce coding errors)
- Alternative: Provide general patterns for connecting TLM and AMS models (more flexible, but usually much harder to use)

Summary

- ANDRES project develops solutions to cope with problems and issues in heterogeneous system design (including AMS and TLM)
- TLM models and SDF-cluster can be connected using standard SystemC and SystemC-AMS features, but it is not very easy to use, may be error-prone and requires much application knowledge
- Timing relations can become greater issue for non-strictly timed TLM models
- Need for facilities to ease interfacing of AMS and TLM → *Converter Channels*