A SystemC TLM 2.0 Extension for the Model Exchange of Off-Chip Communication Protocols

G. Stazi, V. Di Valerio, S. Sinisi, A. Ulisse, S. Soffia

Raytheon Technologies Research Center Italy

accellera SYSTEMS INITIATIVE™

SYSTEMC™ EVOLUTION DAY
OCT 29, 2020 | VIRTUAL WORKSHOP

Copyright Permission

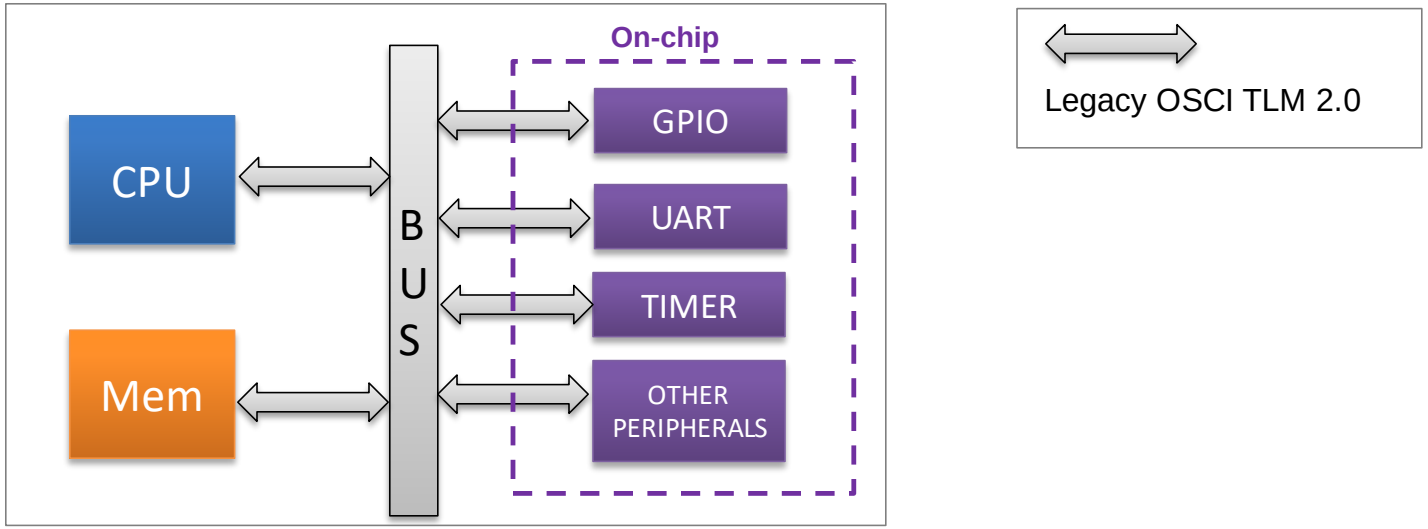
- A non-exclusive, irrevocable, royalty-free copyright permission is granted by **Raytheon Technologies Research Center Italy – ALES S.r.l** to use this material in developing all future revisions and editions of the resulting draft and approved Accellera Systems Initiative **SystemC** standard, and in derivative works based on the standard.

Outlines

- **Need for TLM2.0 extension**
- **ISO / OSI communication model**
- **Network protocols and TLM semantics**
- **Network protocols and TLM 2.0 Accellera execution Interface**
- **Conclusion**

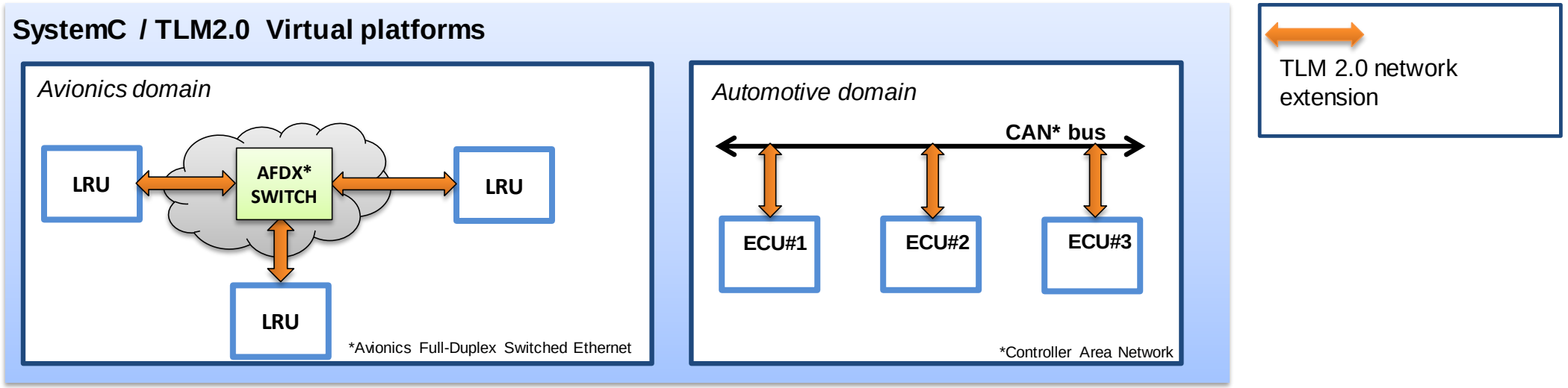
TLM2.0 – Current status

- **OSCI TLM2.0**:
  - Standard for modeling memory mapped buses
  - Interoperable register-based communication on chip



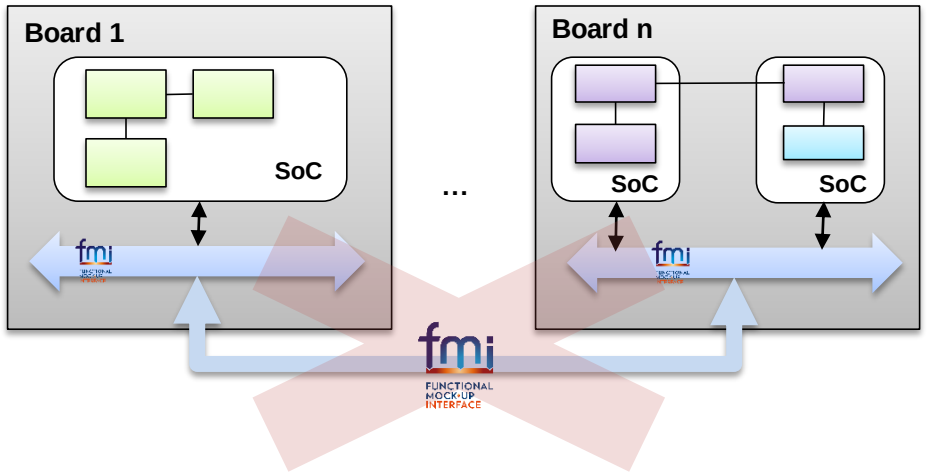
TLM2.0 – Limitations

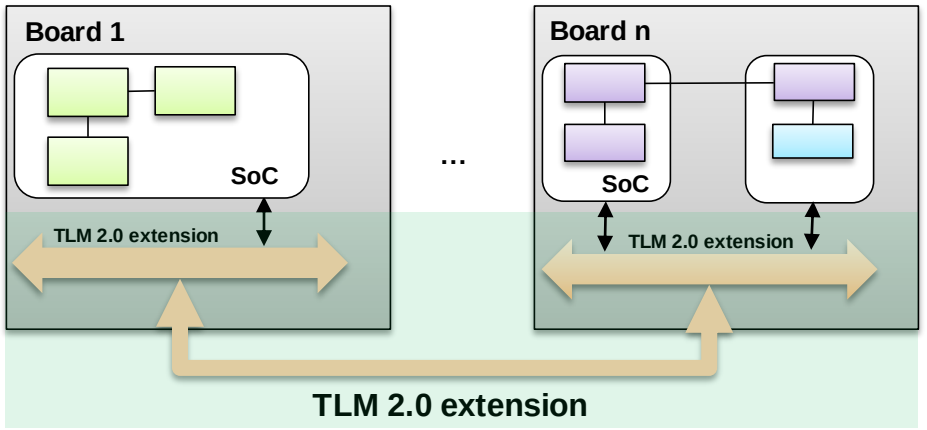
- **Features not supported by TLM2.0:**
  - Off chip network communication protocols:
    - Intra-board communication (e.g. SPI, I2C, …)
    - Inter-board communication (e.g. CAN, Ethernet, AFDX, etc.)



TLM2.0 – Need for extension

- **Motivations:**
  - Enable interoperable simulation of heterogeneous distributed platforms;
  - Enable the model exchange of network protocol models
    - **No existing standard supports this feature**

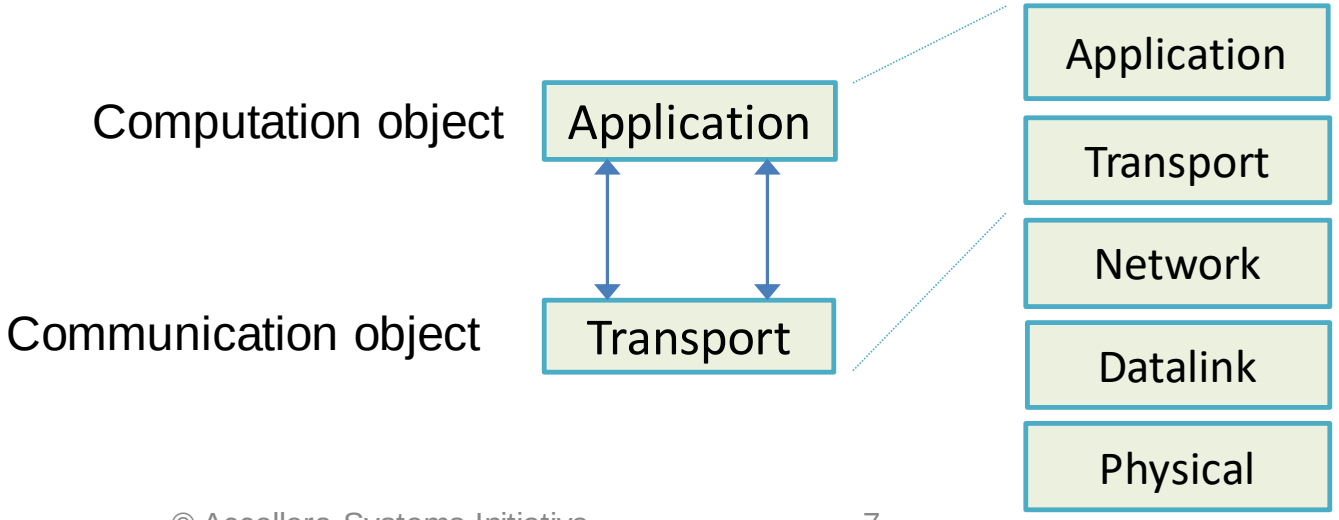




**Several on-going activities concerning TLM Standard for Serial Interfaces (SCED 2016, SCED 2018)*

Network protocols and TLM

- **Transaction-Level Modeling** = < {**objects**}, {**compositions**} >
  - **Object** = {computation object} | {communications object}
  - **Composition**: Computation objects send and receive data via communications objects
- **Network protocols:** inherently implement the concept of transaction modeling (*LT and AT timing accuracy*)
- **Example**: Application protocol sends/receives data via Transport protocol

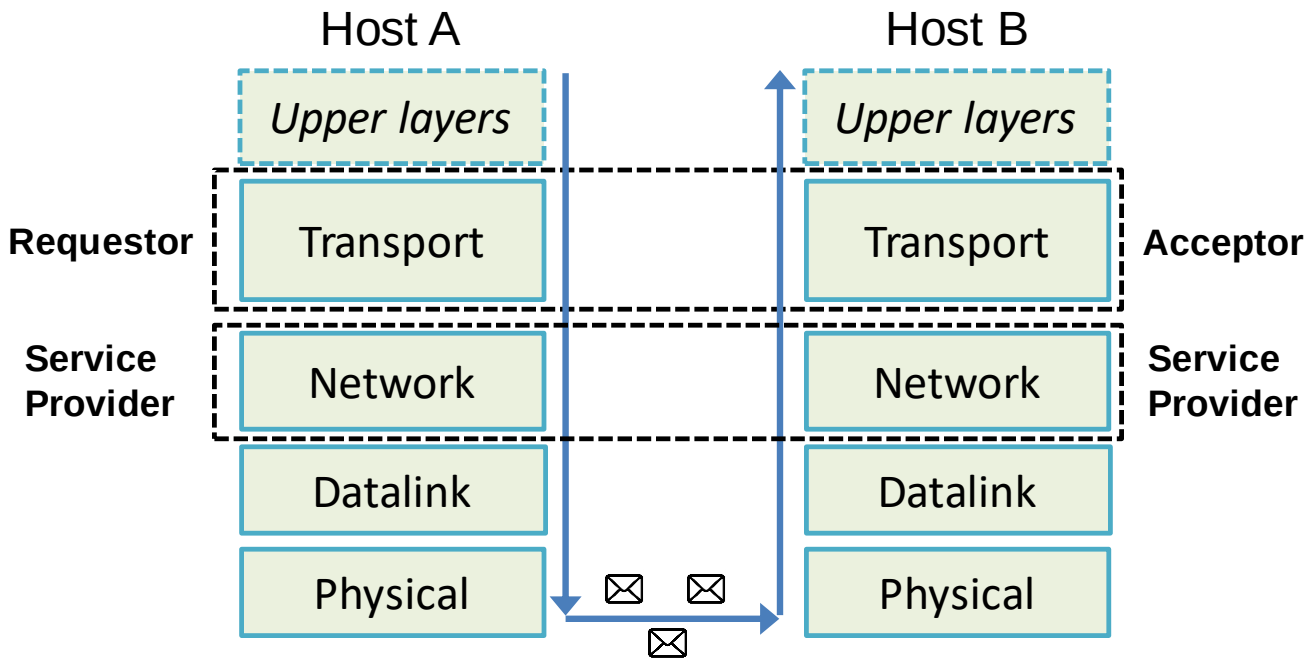


ISO-OSI standard - A Network Reference Model

- **OSI-service**: capability of an OSI-service-provider which is provided to OSI-service-users
- **OSI-service-user**: An entity that makes use of an OSI-service
- **OSI-service-provider**: An entity that provides an OSI-service
- Basic Primitives:
  - **Submit**: the primitive invoked by the service-user to exchange information with the service-provider
  - **Deliver**: the primitive invoked by the service-provider to exchange information with the service-user

ISO-OSI standard – Requestors and Acceptors

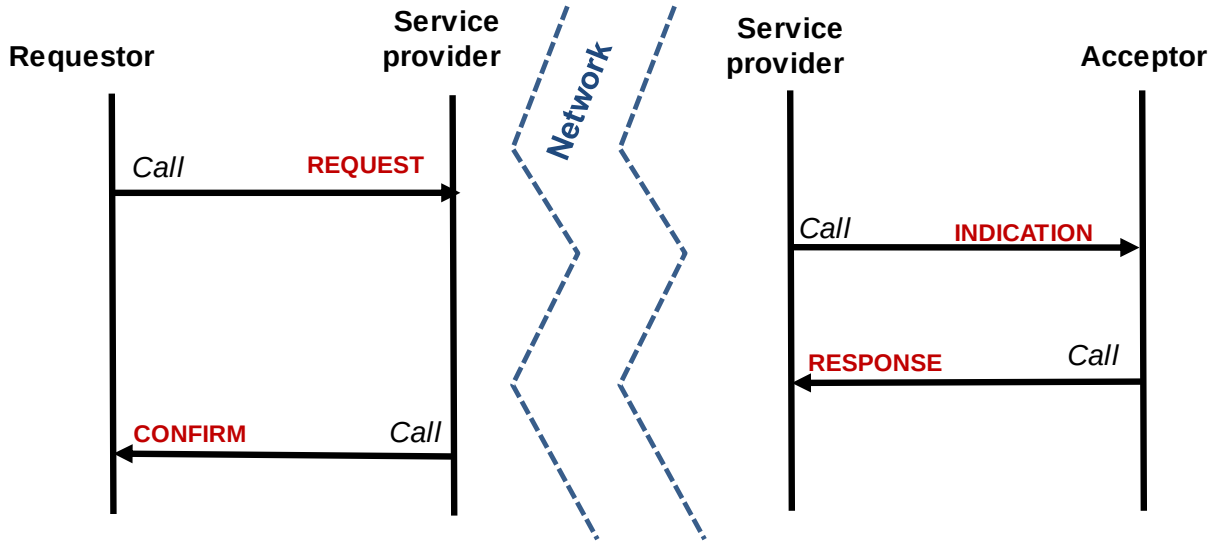
- **Requestor:** An OSI-service-user that issues a submit primitive
  - And as a result may receive one or more deliver primitives
- **Acceptor:** An OSI-service-user that receives a deliver primitive
  - And as a result may issue one or more submit primitives is, for that exchange



ISO-OSI standard – Connection mode service

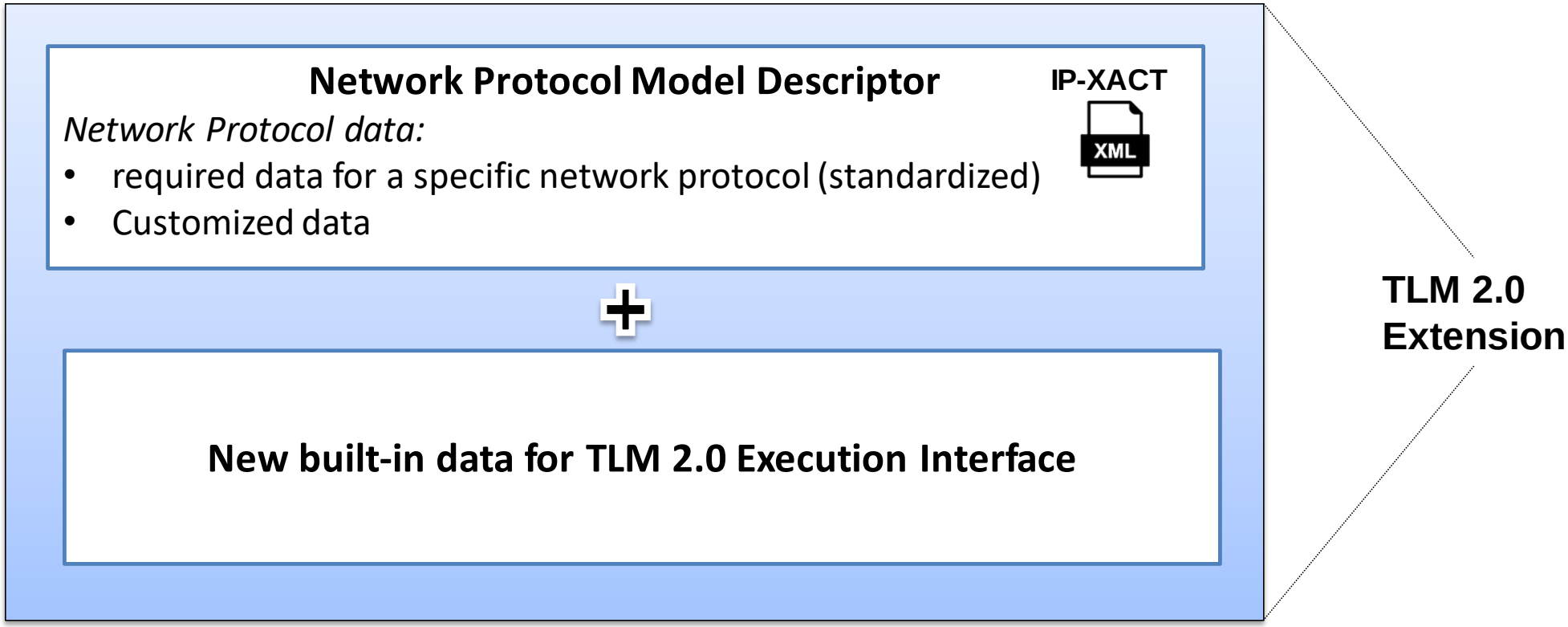
- A requestor.submit primitive is called a **request** primitive
- A requestor.deliver primitive is called a **confirm** primitive
- An acceptor.deliver primitive is called an **indication** primitive
- An acceptor.submit primitive is called a **response** primitive

The connectionless mode service uses a subset of the primitives

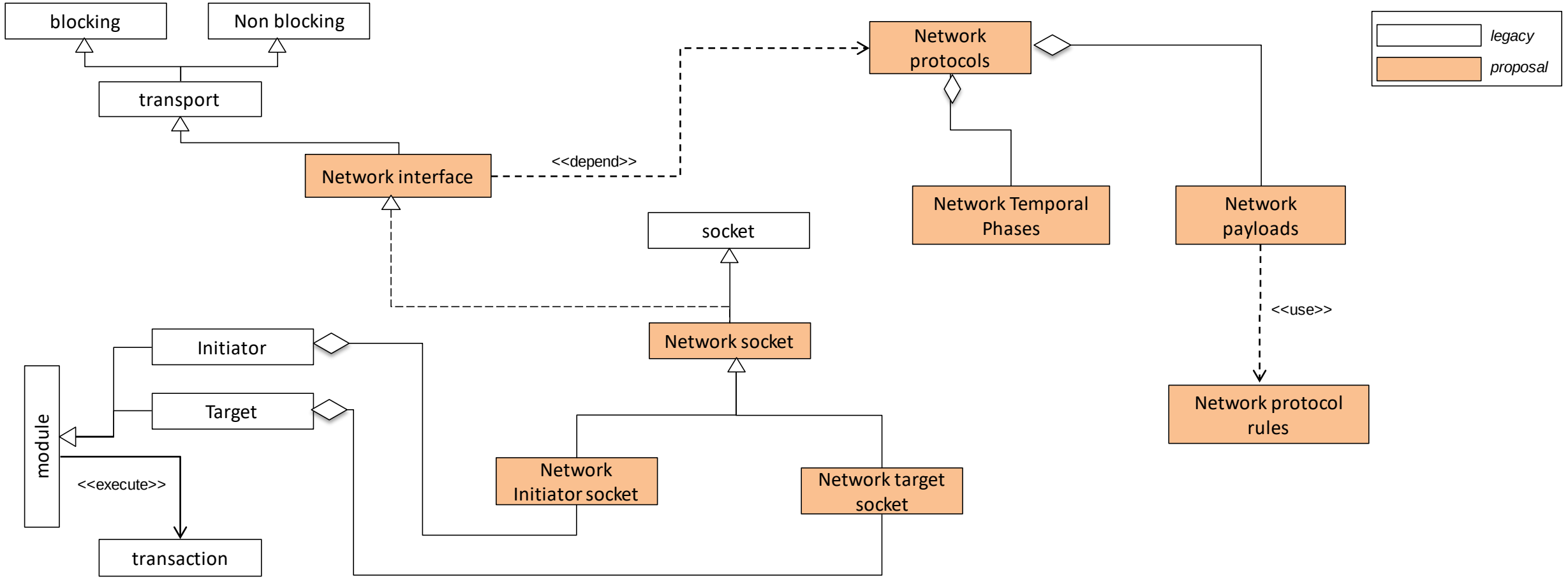


TLM Extension for Networking – Global picture

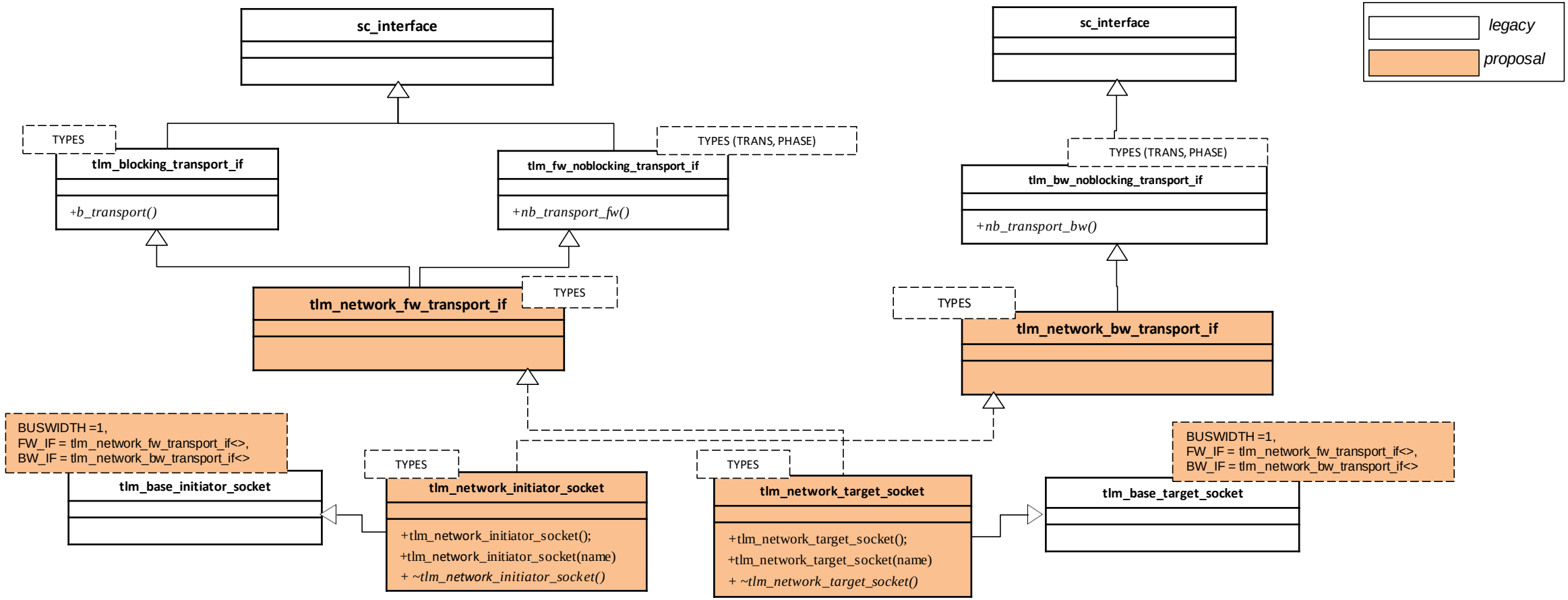
- TLM extension philosophy for network protocols includes:



TLM Extension for Networking – Global picture



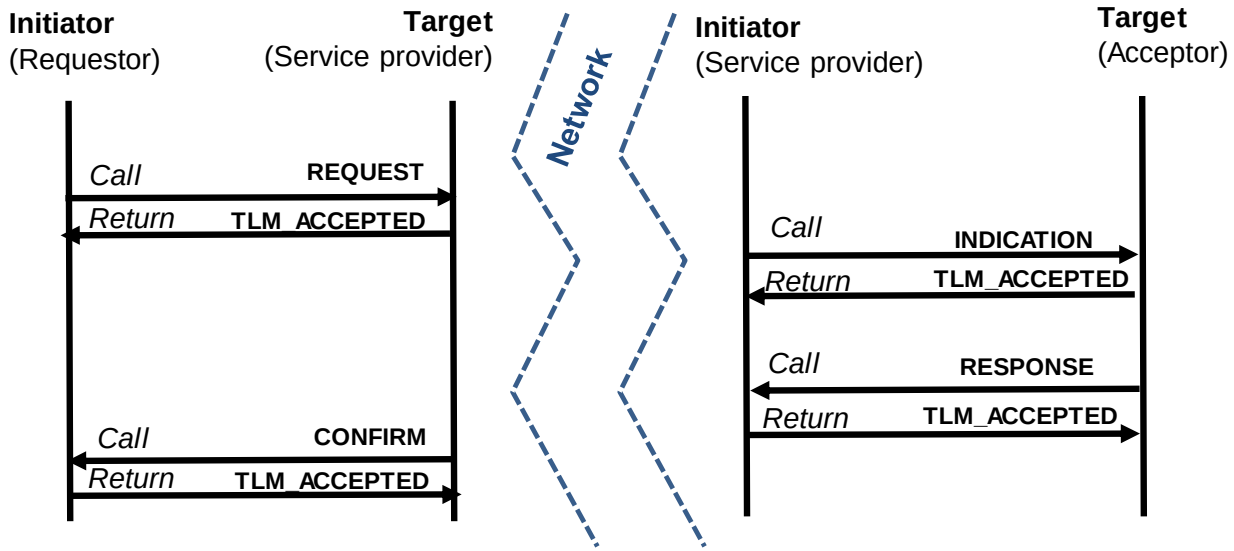
TLM Extension for Networking– Syntax



TLM2.0 Backward compatible

TLM Extension for Networking - Semantic

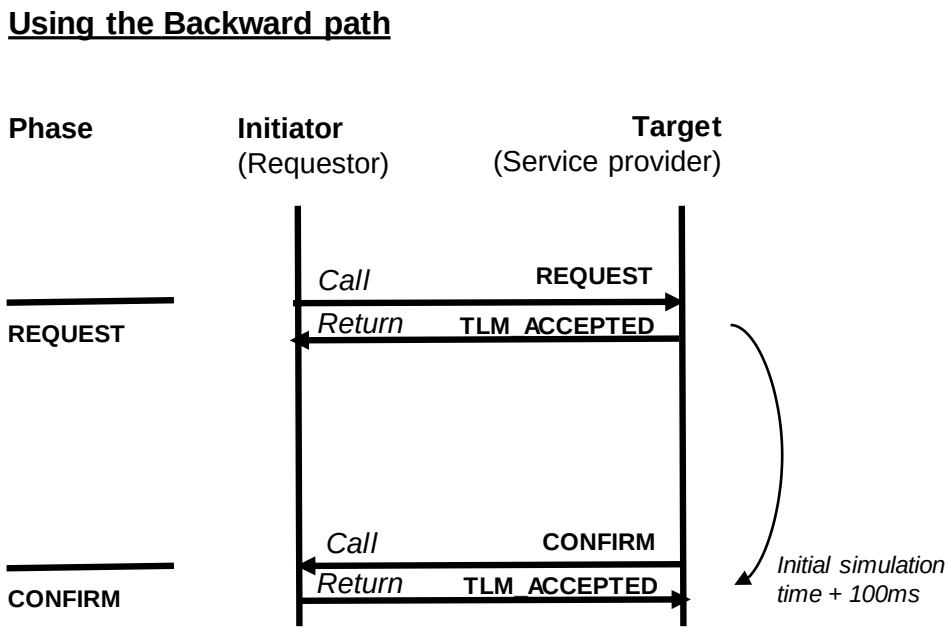
- **Based on AT coding style →** Network transactions are generally non blocking
- **4 new TLM phases defined →** 4 synchronization points (REQUEST, INDICATION, RESPONSE, CONFIRM) mapping ISO-OSI primitives

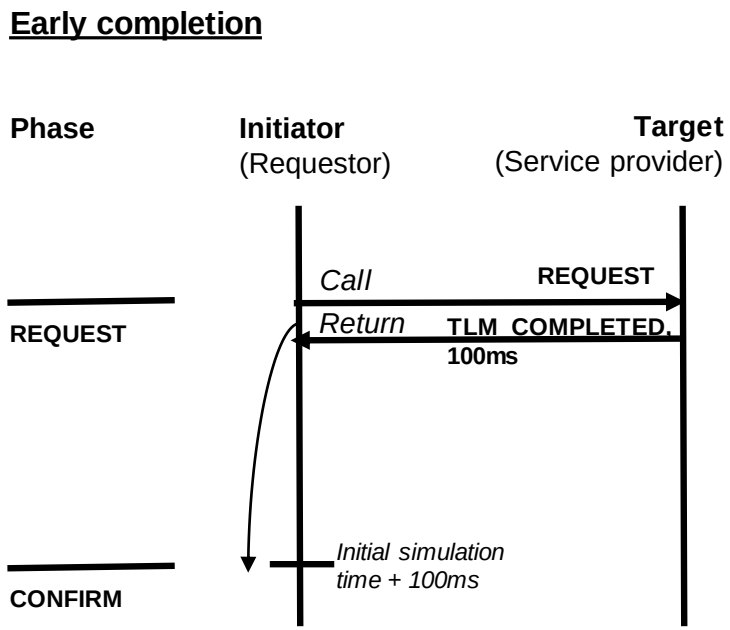


New TLM phases in action

TLM Extension for Networking - Semantic

- Sequence diagrams:
  - Only backward path and early completion used, return path is not so meaningful for networking

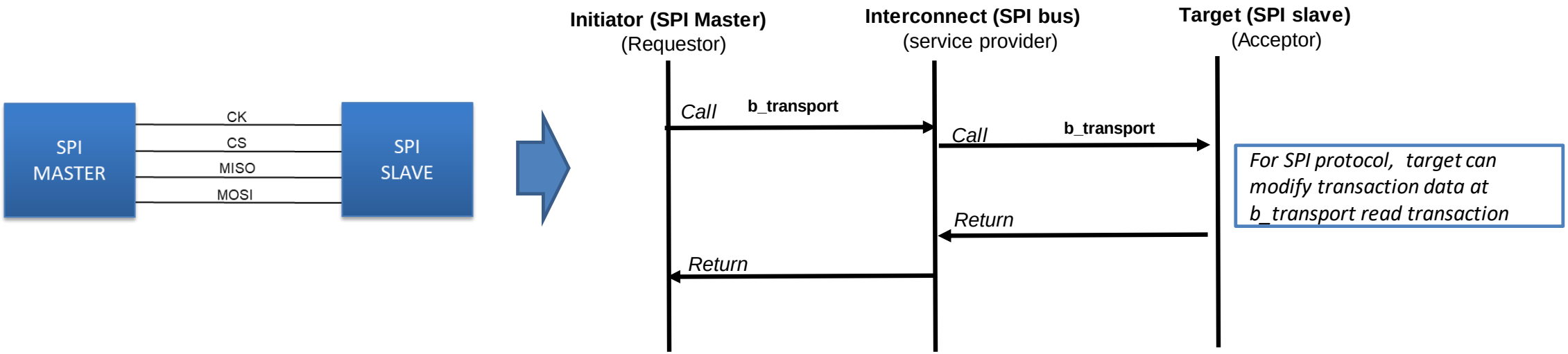




TLM Extension for Networking - Semantic

- **LT coding style**

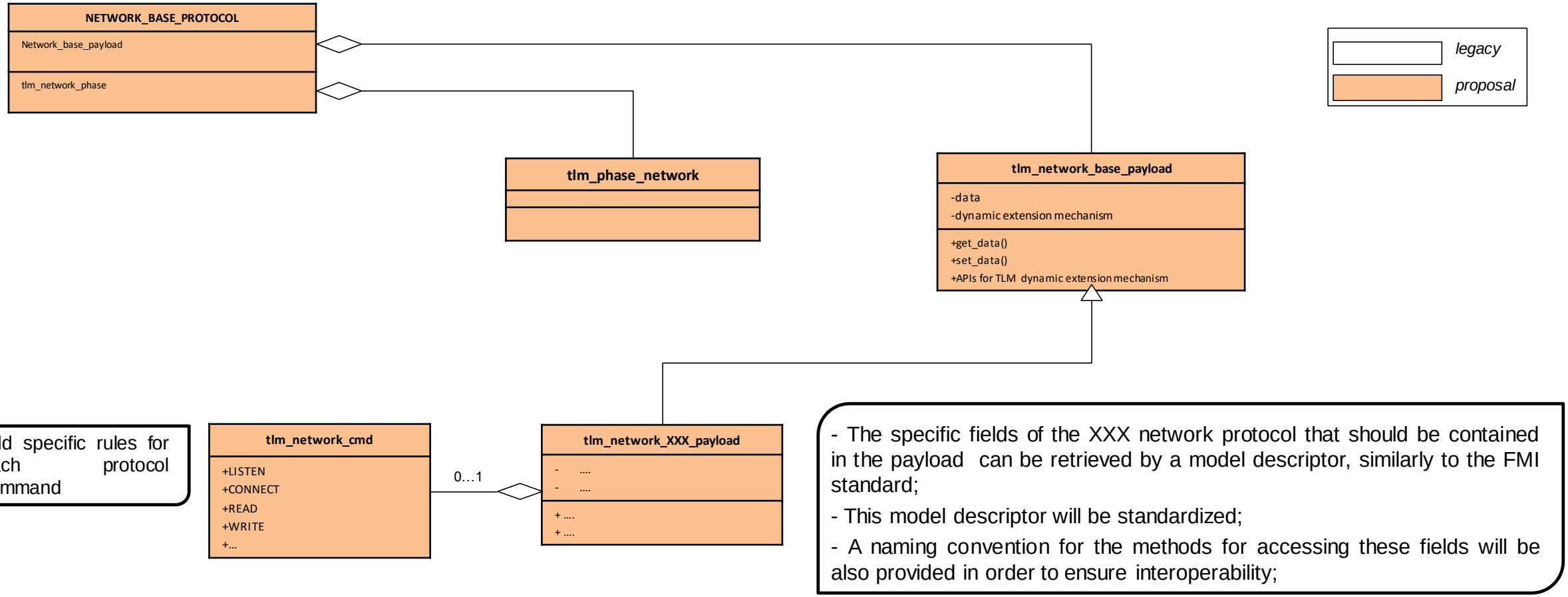
  For example SPI communication model



TLM Extension for Networking - Protocol

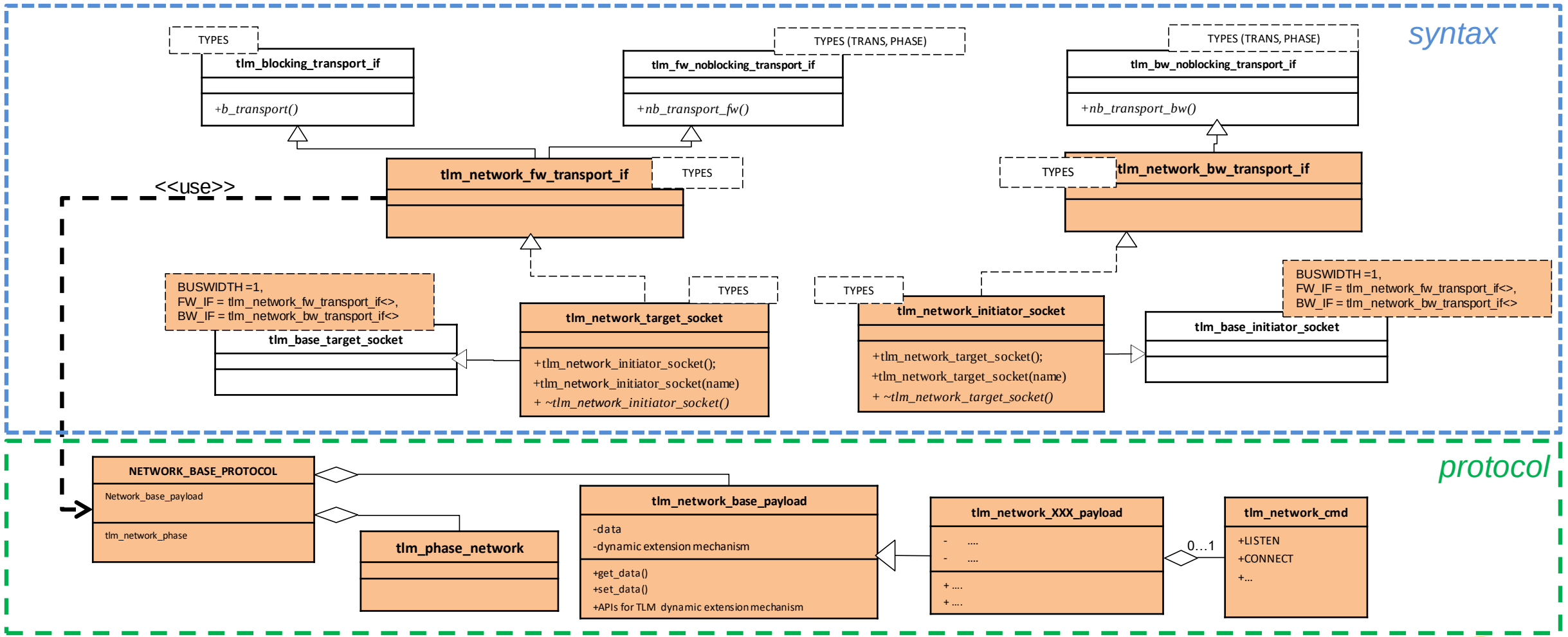
- **Any network protocol is defined by:**
  - Network temporal phases (mapping ISO-OSI semantic)
  - A network payload extending a base network payload
- **Network payload (XXX_tlm_network_payload):**
  - Standardized data
  - Standardized network commands
    - Specific behavioral rules can be defined for each command
  - Custom data described in a model descriptor
    - Extension of IP-XACT XML format
  - Standardized naming convention for access methods

TLM Extension for Networking - Protocol



- The specific fields of the XXX network protocol that should be contained in the payload can be retrieved by a model descriptor, similarly to the FMI standard;
- This model descriptor will be standardized;
- A naming convention for the methods for accessing these fields will be also provided in order to ensure interoperability;

TLM Extension for Networking – Global picture



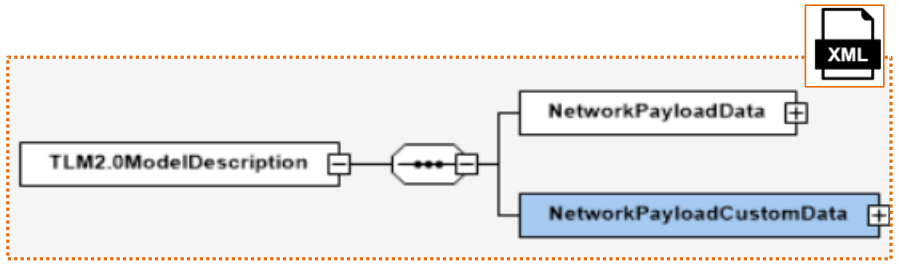
TLM Extension for Networking

- ***Example – ARINC 429***

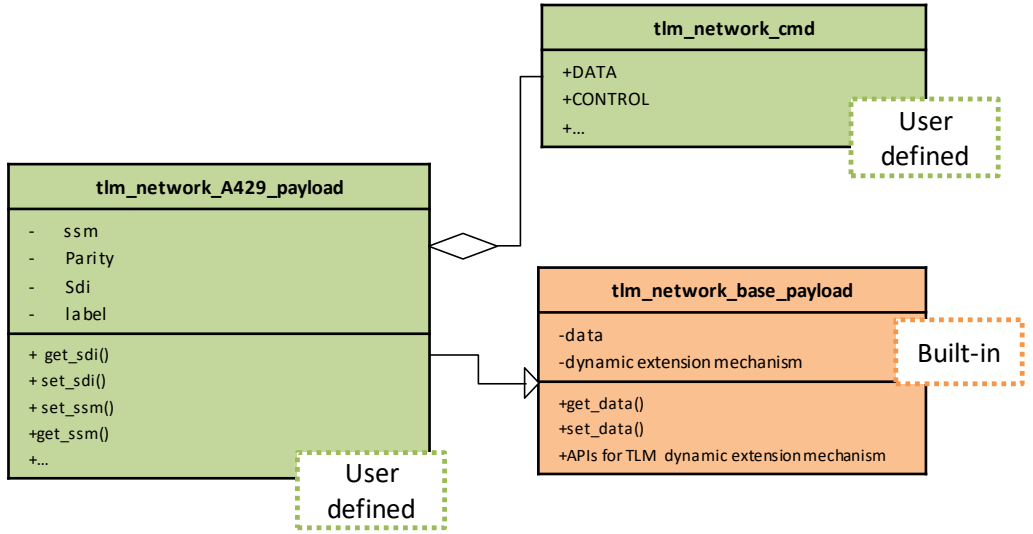
ARINC 429 Word Format																													
P	SSM		MSB																	LSB	SDI		Label						
32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

- *Parity_bit*
- *SSM*
- *SDI*
- *Label*
- *<custom data>*

A429 REQUIRED DATA



A429_MODEL_DESCRIPTOR



Conclusion

- OSCI TLM2.0 extension for off-chip network communication protocols
- Generic and re-usable approach applicable to any network protocol (avionics and automotive domain)
  - Model each layer (or even multiple layers together) of a protocol stack as separate TLM IP component
  - Enable the model exchange of components of the network protocol stack
- Ensure interoperability when modeling communication interfaces of modern SoCs

Q&A

Thank you for your time

Backup

TLM Extension for Networking

- IP-XACT extension

```
<TLM2.0ModelDescription
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  protocolName=""
  description="">
    <NetworkPayloadData>
        <Fields>
            <Field
        name = ""
        type = " "></Field>
        </Fields>
    </NetworkPayloadData>
    <NetworkPayloadCustomData>
        <CustomFields>
            <CustomField
        name = ""
        type = " "></CustomField>
        </CustomFields>
    </NetworkPayloadCustomData>
</TLM2.0ModelDescription>
```

High level information: protocol name & brief description

Standardized network payload data : for each payload field → field name and data type

Customizable network payload data

Information to be added