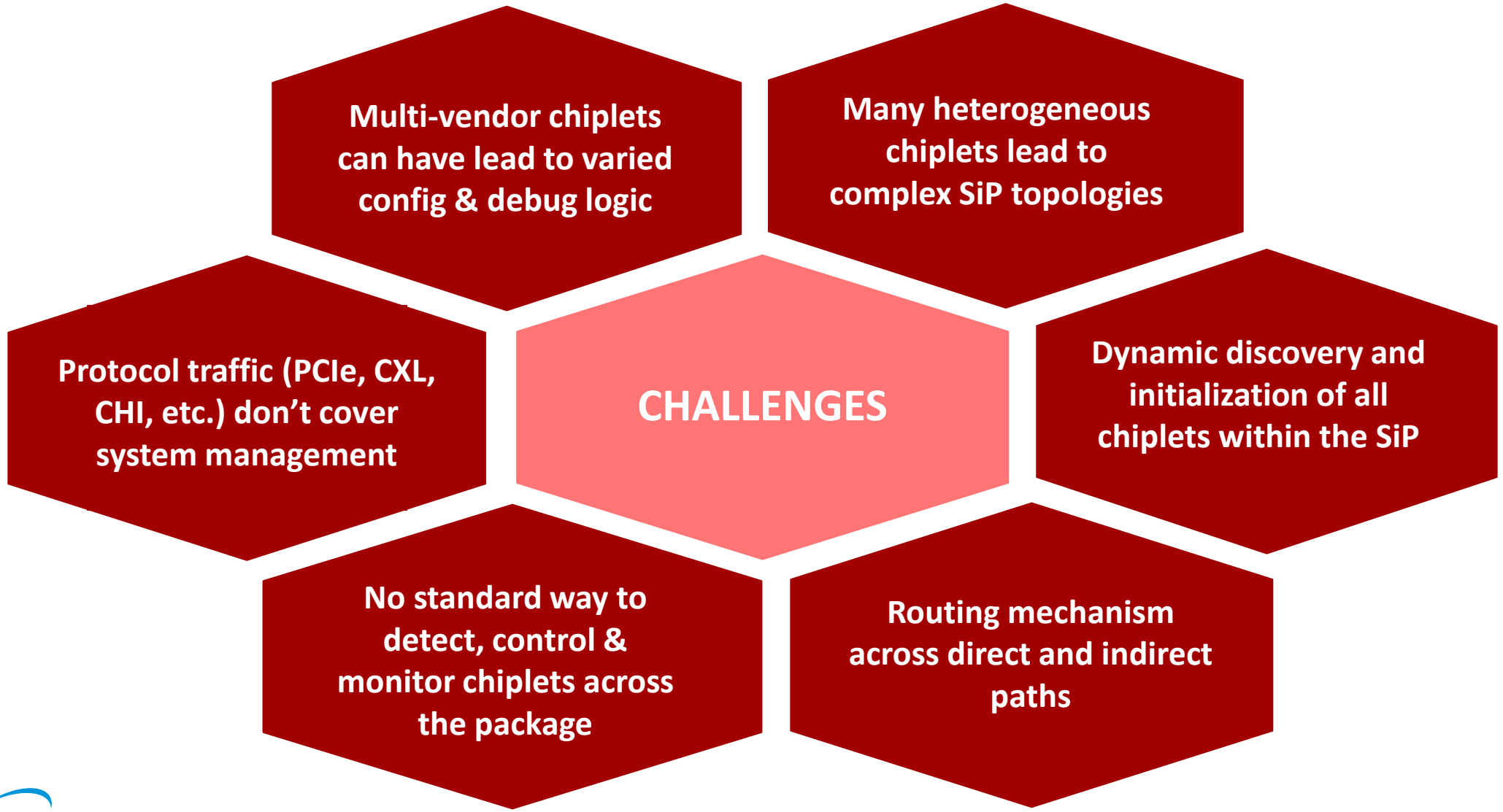
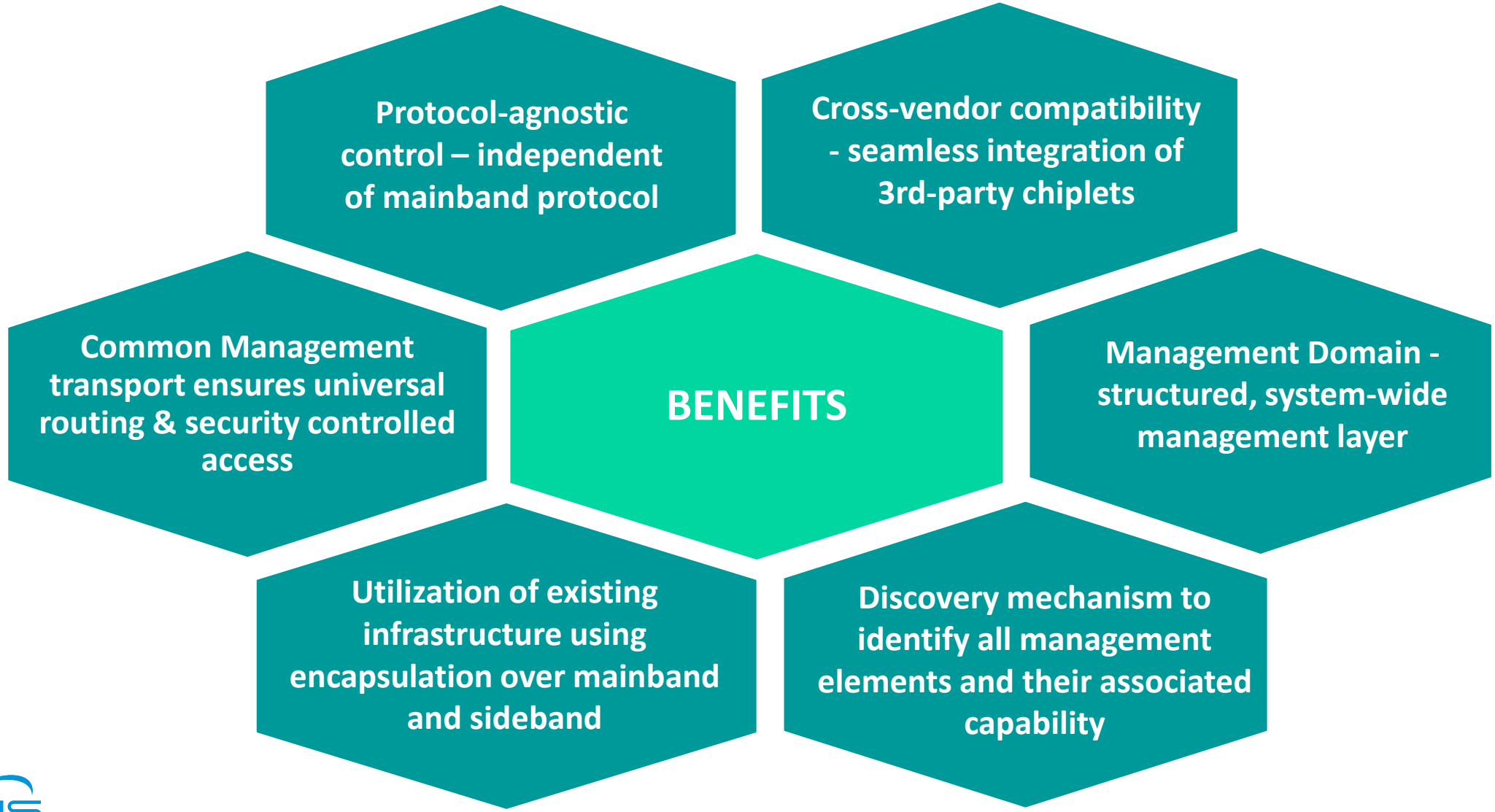
Mastering UCIe 2.0: Overcoming Fabric Management Hurdles for Chiplet Integration

Prashant Dixit & Ujjwal Negi
Siemens EDA

Why Manageability Matters in a Chiplet World



Why Manageability Matters in a Chiplet World

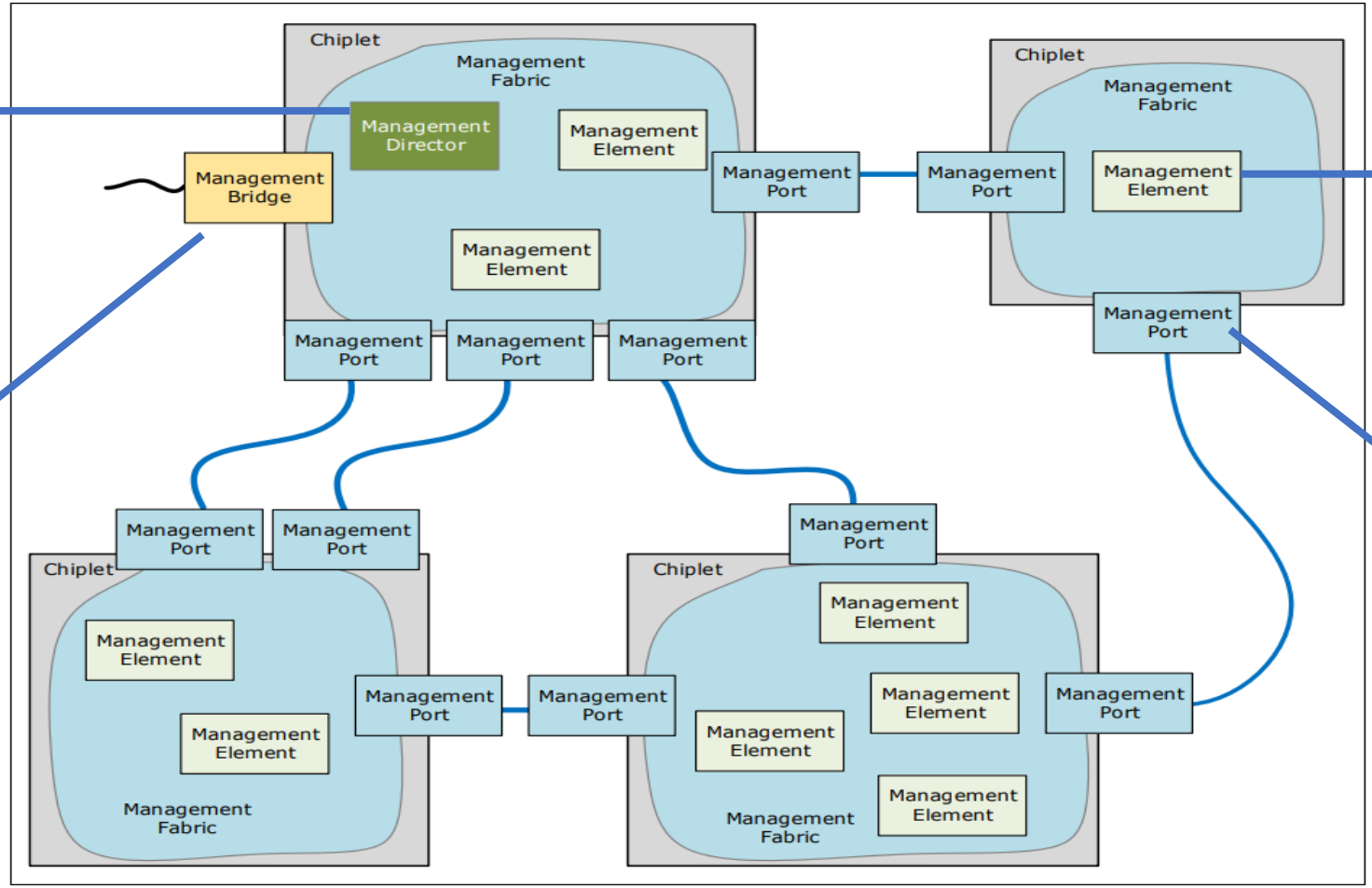


Inside UCIe 2.0: A Layered Management Architecture

- **Layered management fabric with Elements, Ports & Bridges**

Discovers and configures – Chiplets, Management Elements, Management Ports

Acts as a bridge between two management domains



functional block that handles local management tasks for a chiplet

point-to-point management link that interconnects two chiplets and handles routing

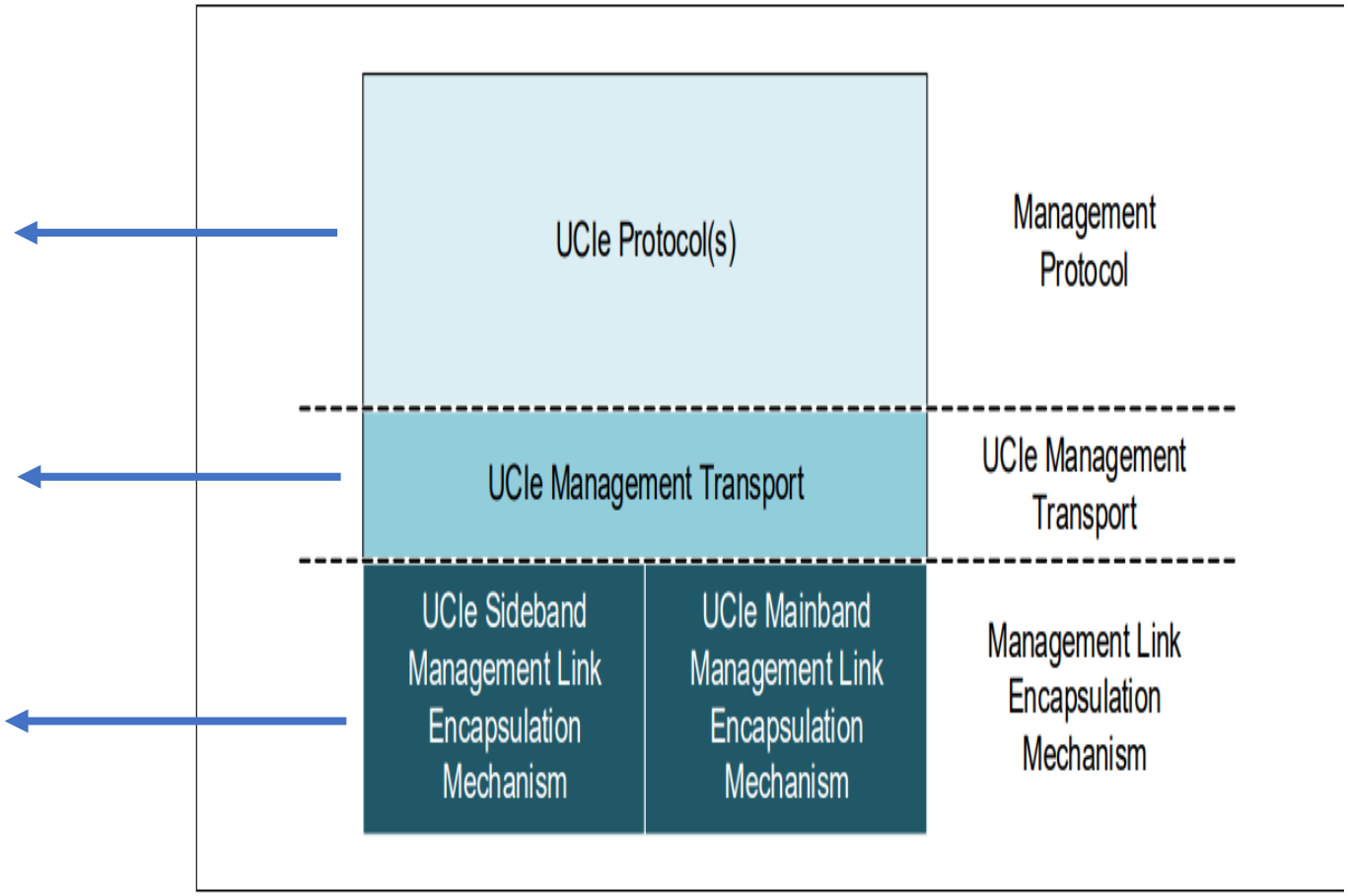
Inside UCIe 2.0: A Layered Management Architecture

❑ Layered management protocol hierarchy

Standard management protocols like **UMAP**, **UDA**, etc., or vendor defined management protocol define the content and format the management data

These protocols are carried by the Management Transport Packets **(MTP),** which handles routing and security enforced access to management entities

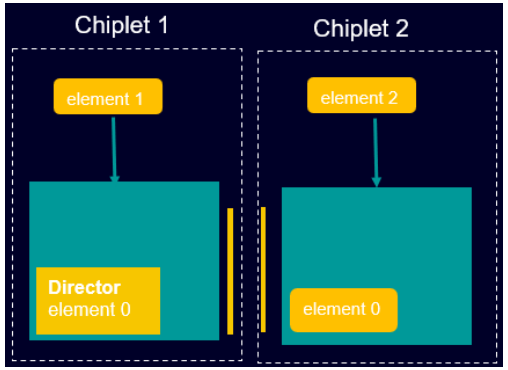
MTP packets are then encapsulated inside the Management Port Message **(MPM)** which handle segmentation and credit flow, that are packed into **management flits** over mainband



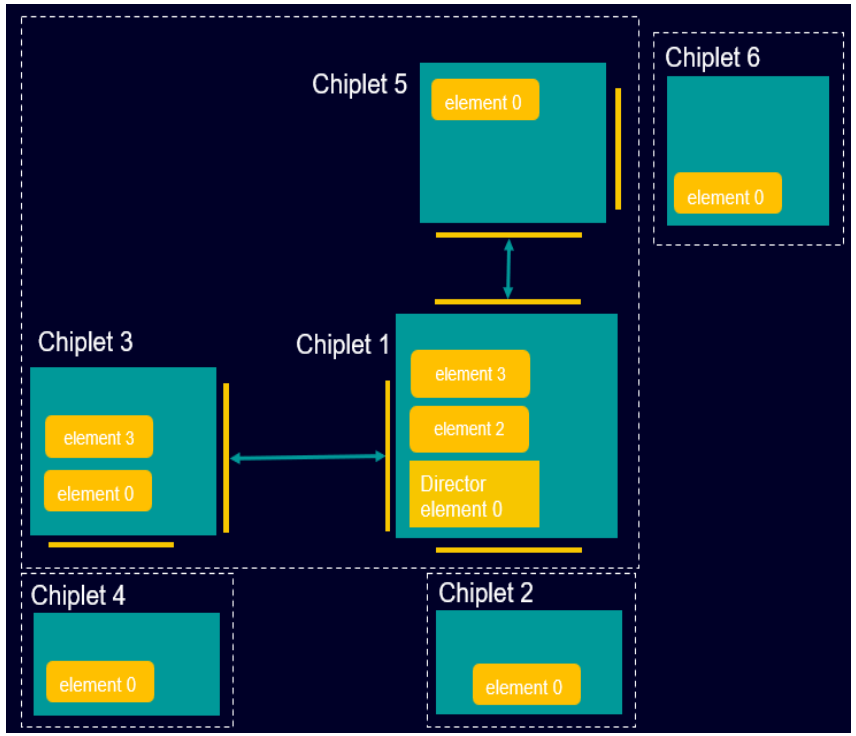
Verification Challenges

❑ Management Domain Configuration

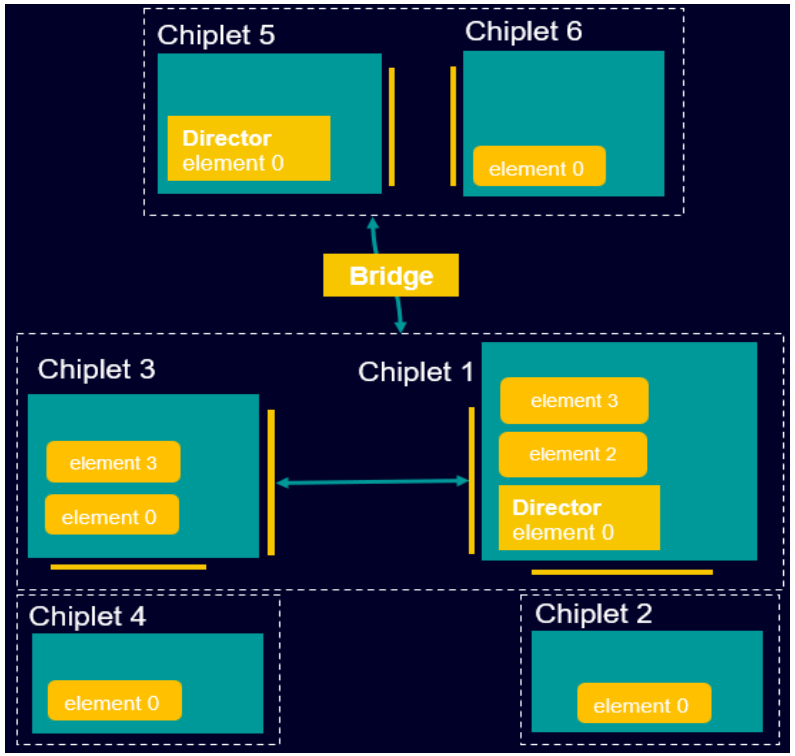
- SiPs vary: point-to-point, daisy chain, multi-chiplet, multi-domain topologies
- Need flexible verification to track multiple management elements & ports per chiplet



P2P



Multi-chiplet single domain



Multi-chiplet multi-domain

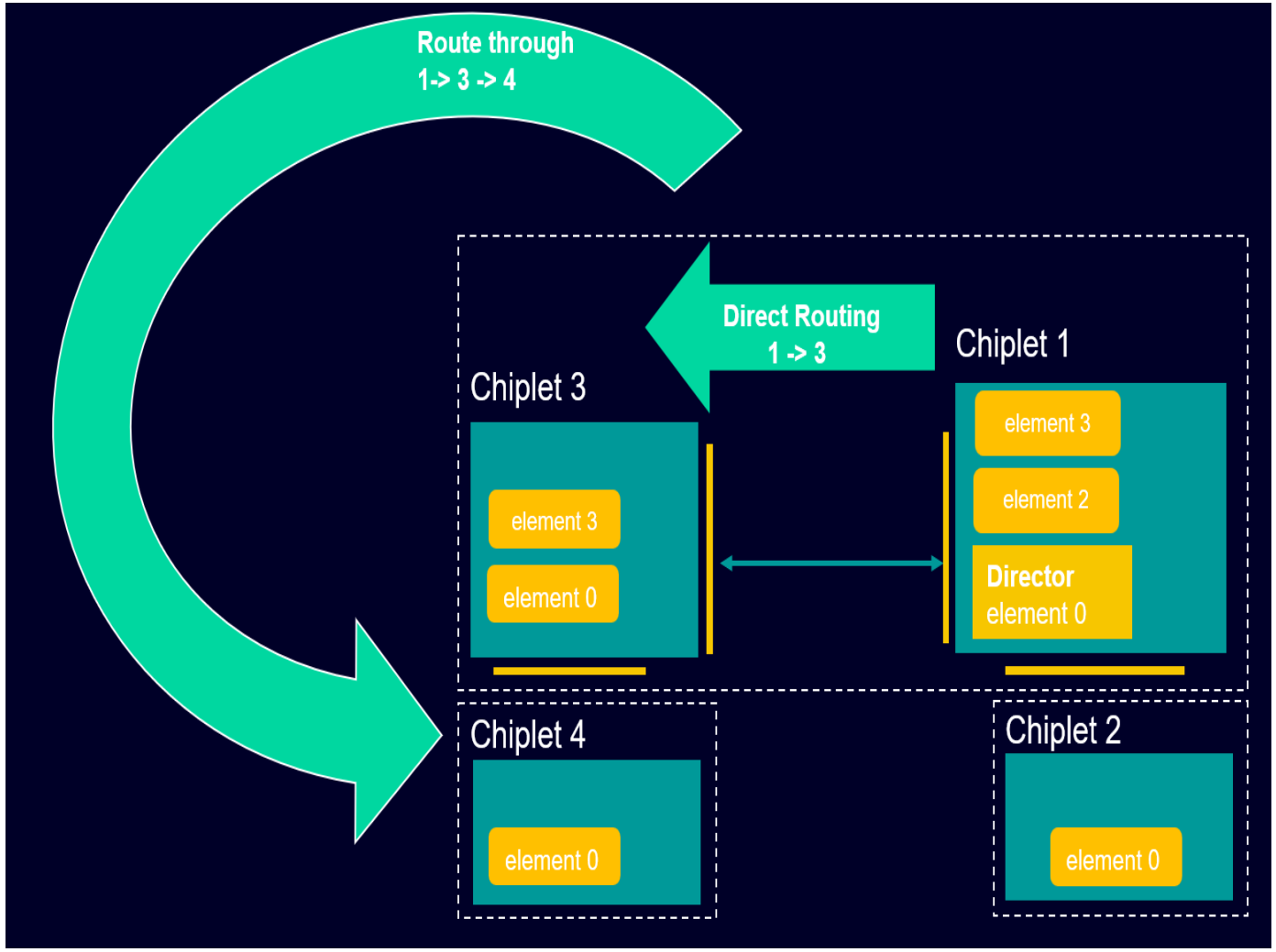
Verification Challenges

❑ Discovery and Initialization

- Identify all management elements & configure chiplet ID, CMPS
- Support both front door and backdoor initialization flows

❑ Routing and Network ID Mapping

- Validate routing tables, network IDs & route-through across hops
- Handle dynamic updates, fallback paths & cyclic scenarios



Verification Challenges

❑ Transmitter and Receiver Rules

- Ensure correct **encapsulation** of MTP within MPMs and then packing them inside management flits
- Must check for **segmentation, interleaving**, credit handling across rxqid and virtual channels
- Handle reassembly, idle detection & retraining under multi-port conditions

❑ Support for Protocol Diversity

- Payloads may use UMAP, UDA etc.; verification must decode & check them
- Ensure protocol compliance & interoperability within the management transport

Management Transport Packet (MTP)	
QWORD 0	MTP Header
QWORD 1	MTP Data 0
QWORD 2	MTP Data 1
QWORD 3	MTP Data 2
QWORD 4	MTP Data 3
QWORD 5	MTP Data 4
QWORD 6	MTP Data 5
QWORD 7	MTP Data 6
QWORD 8	MTP Data 7
QWORD 9	MTP Data 8
QWORD 10	MTP Data 9
QWORD 11	MTP Data 10
QWORD 12	MTP Data 11
QWORD 13	MTP Data 12
QWORD 14	MTP Data 13
QWORD 15	MTP Data 14

1st Segment[b] — This goes on RxQ-ID=x	
QWORD 0	MPM Header (s = 1, length = 6h)
QWORD 1	MTP Header
QWORD 2	MTP Data 0
QWORD 3	MTP Data 1
QWORD 4	MTP Data 2
QWORD 5	MTP Data 3
QWORD 6	MTP Data 4
QWORD 7	MTP Data 5
2nd Segment[b] — This goes on RxQ-ID=MOD((x+1)/N)	
QWORD 8	MPM Header (s = 1, length = 6h)
QWORD 9	MTP Data 6
QWORD 10	MTP Data 7
QWORD 11	MTP Data 8
QWORD 12	MTP Data 9
QWORD 13	MTP Data 10
QWORD 14	MTP Data 11
QWORD 15	MTP Data 12
3rd Segment[b] — This goes on RxQ-ID=MOD((x+2)/N)	
QWORD 0	MPM Header (s = 0, length = 1h)
QWORD 1	MTP Data 13
QWORD 2	MTP Data 14

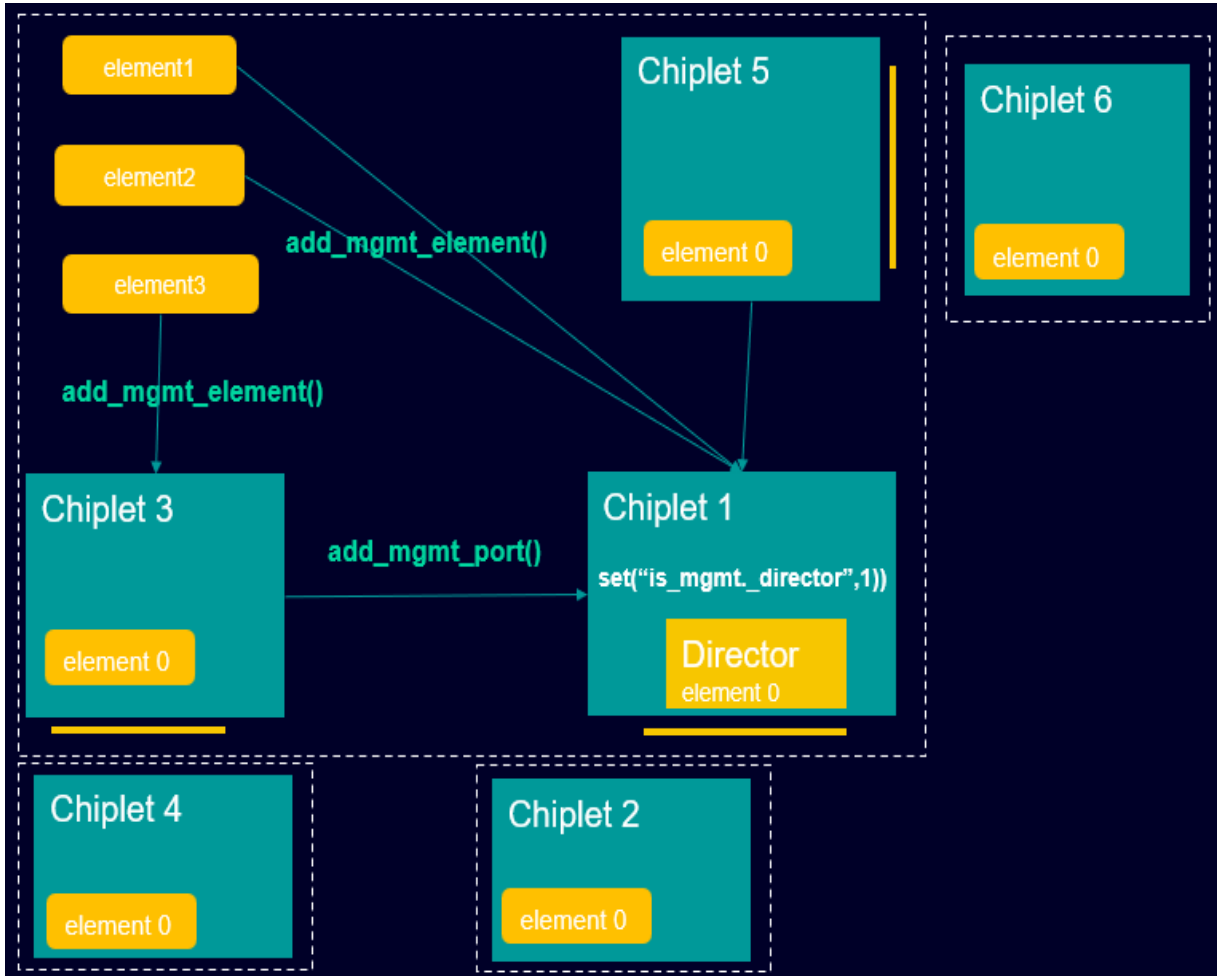
Proposed VIP Architecture

❑ Layered Modular Design

- Built on UVM architecture → scalable & reusable
- Modular classes for key management components: Management Element, MTP, MPG

❑ Flexible topology Modelling

- Configurable APIs to represent any chiplet topology - supports simple point-to-point to complex multi-chiplet domains
- Attach multiple management elements and ports to a particular chiplet
- Configure any chiplet within the domain as management director, security director



Proposed VIP Architecture

❑ Discovery Engine

- Frontdoor & backdoor routines to initialize management elements & ports
- Handles dynamic discovery & capability setup
- Allows bypassing of default discovery routine or customizing it

❑ Role-Agnostic Verification

- Allows verification of any component within the manageability domain as the DUT - whether it's a separate chiplet or a management element residing within the same chiplet as the VIP.

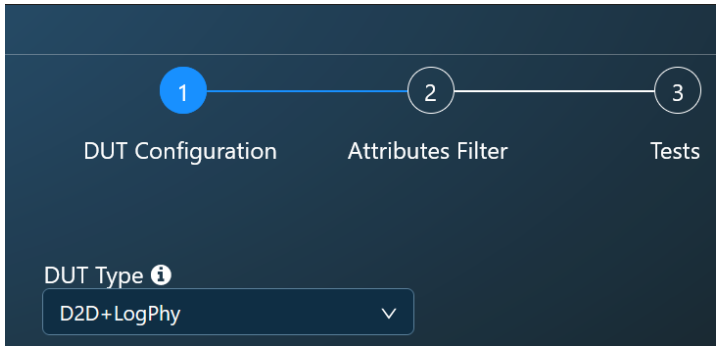
❑ Exhaustive Compliance Checks

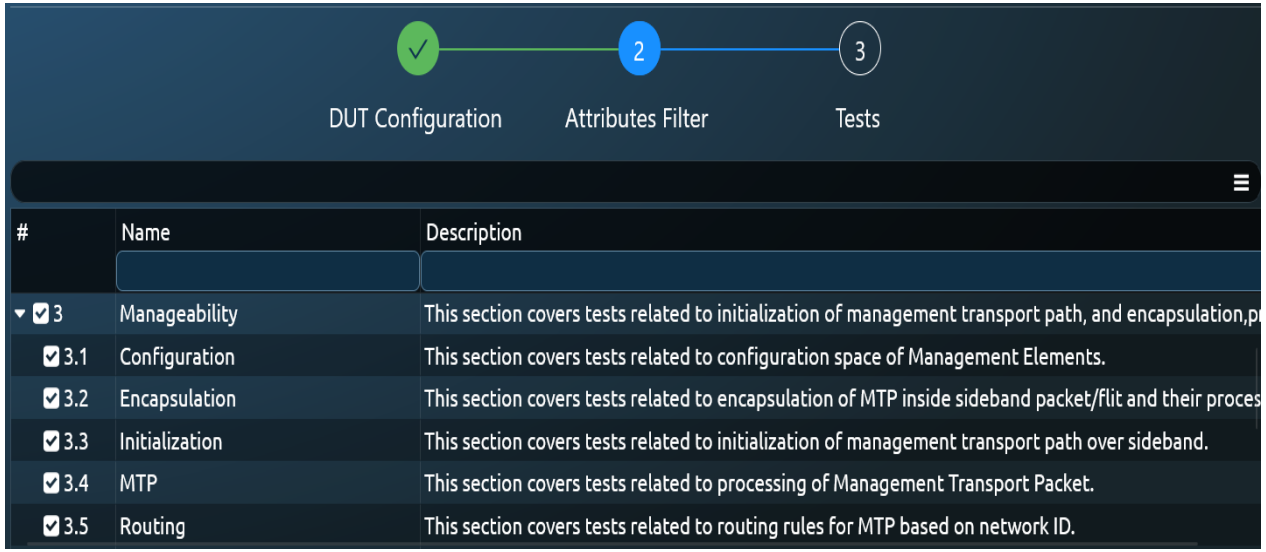
- Validates discovery, routing, segmentation & credits rules
- Protocol checkers ensure adherence to Management transport, UMAP, UDA rules

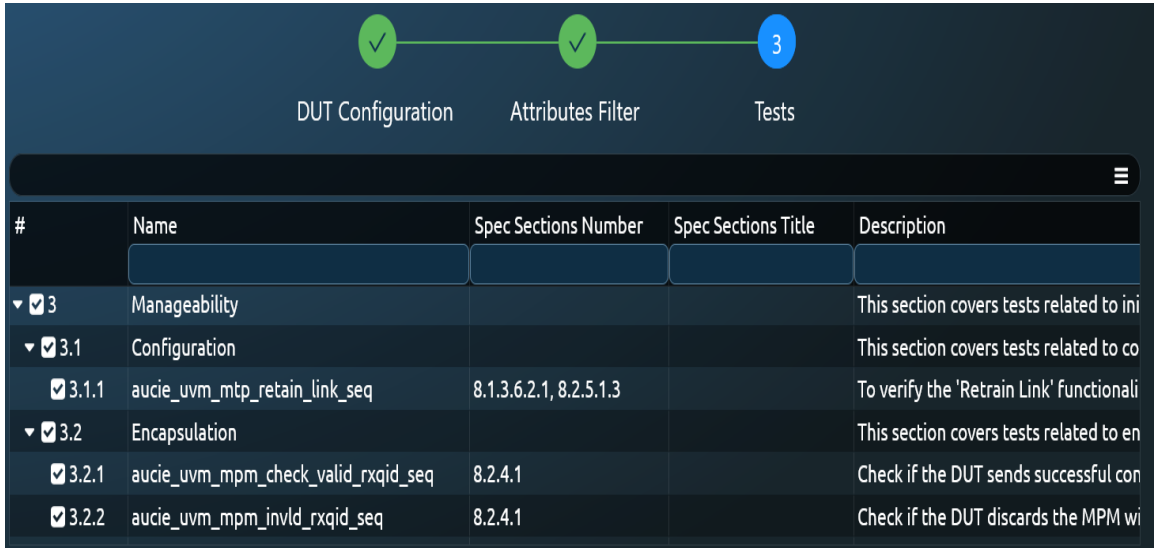
Proposed VIP Architecture

❑ Ready made Compliance Test Suite

- Dump ready to use compliance test plan based on DUT Type and specific sectionsand tests for targeted testing using smart tools like VIQ







Proposed VIP Architecture

❑ End-to-End Debug and Traceability

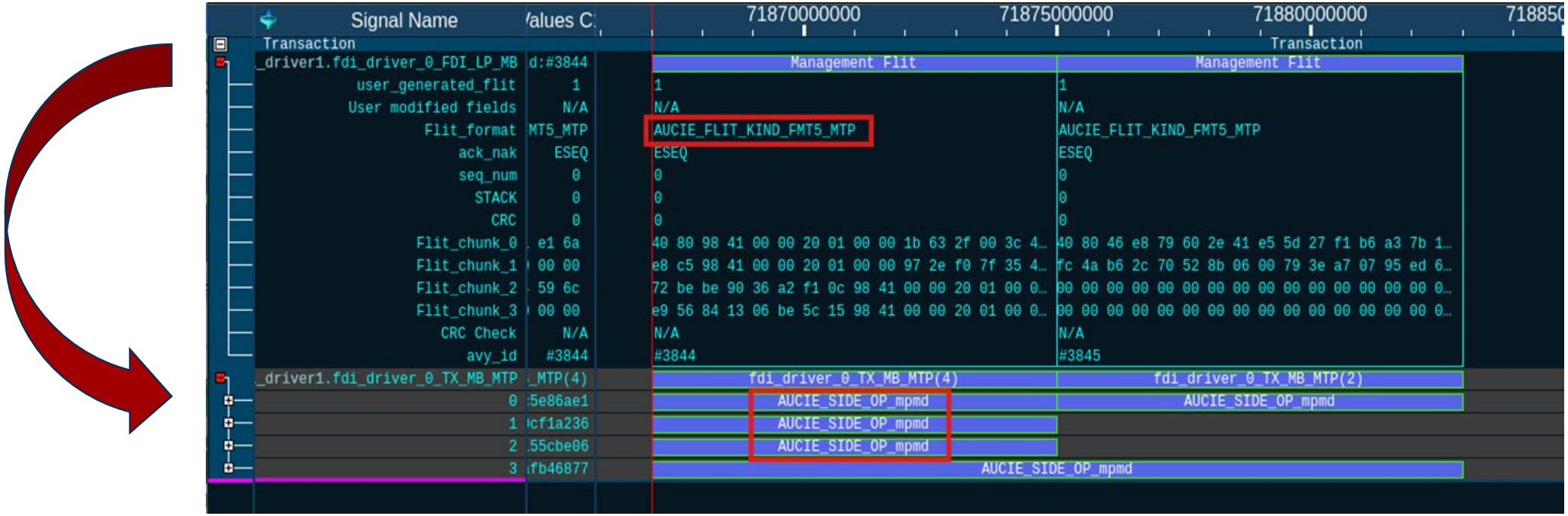
- Trackers associate : **MTP → MPM → Management Flit**
- Enables root-cause debug from transaction to low-level interface

```
1041
1042 ==> Time:999.000ns  MTP (AUCIE_MP_ID_MA) #5419
1043 +-------------------------------------------------------------------+
1044 | dstid:0 | mgmt_prot_id:1 | tc:3 | pipp:3   | resp:1  | rsvd2:0 | ver:0   |
1045 | srcid:0                  | scg:0          | length:4                 |
1046 +-------------------------------------------------------------------+
1047 | Rsvd:0 | Opcode:0 | Rsvd:0   | Status:0 | Tag:11                     |
1048 +--------------------------------- Data ---------------------------------+
1049 |  Data [  31:   0]: 00000a00                                           |
1050 +--------------------------- Packet Protection --------------------------+
1051 | PIP:4e2ed47                                                           |
1052 +-------------------------------------------------------------------+
test_results/auciet_uvm_mtp_mb/fdi_driver1_mtp_tracker.txt
1987
1988 ==> @999.000ns Post MPM (Complete) of MTP #5419 from MPG: MPMD #541b(AUCIE_ENCAP_MTP_MSG)
1989 +-------------------------------------------------------------------------------------------------------------------------------+
1990 | SrcID: 0 | Rsvd: 0 | Resp: 1 | VC: 0    | MsgCode: 01    | Length: 02   | Rsvd: 0     | OpCode: 18                          |
1991 | Rsvd: 0  | CP: 0   | Rsvd: 0 | DstID: 0 | cr_ret_resp: 0 | cr_ret_vc: 0 | cr_ret: 000 | Rsvd: 0    | S: 0 | P: 1 | Rsvd: 0 | rxq_id: 0 |
1992 +-------------------------------------------------------------------------------------------------------------------------------+
1993 +--------------------------- Data ---------------------------+
1994 | Data [  31:   0]: 802f0000                                |
1995 | Data [  63:  32]: 02000000                                |
1996 | Data [  95:  64]: 11000000                                |
1997 | Data [ 127:  96]: 00000a00                                |
1998 | Data [ 159: 128]: 04e2ed47                                |
test_results/auciet_uvm_mtp_mb/fdi_driver1_mpg_tracker.txt
2002
2003 ==> @999.000ns #541c Post Management flit for MTP #5419 from MPG
2004
2005 ==> @999.000ns AUCIE_FLIT_KIND_FMT3_MTP#541c (Seq cf)
2006 Prot_id: 1 | Stack_id: 0 | dlp: 0, | S_upper: c | flit_type: 2 | Ack_Nak: 0 | S_lower: f
2007 Chunk0: 98400002 10000000 00002f80 00000002 00000011 000a0000 47ede204 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000
2008 Chunk1: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000
2009 Chunk2: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000
2010 Chunk3: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2011 4B RSVD: 00000000 | 4B CRD: 00000000 | 10B RSVD: 00000000000000000000 | CRC0: 0000 | CRC1: 0000
test_results/auciet_uvm_mtp_mb/fdi_driver1_mpg_tracker.txt
```

Proposed VIP Architecture

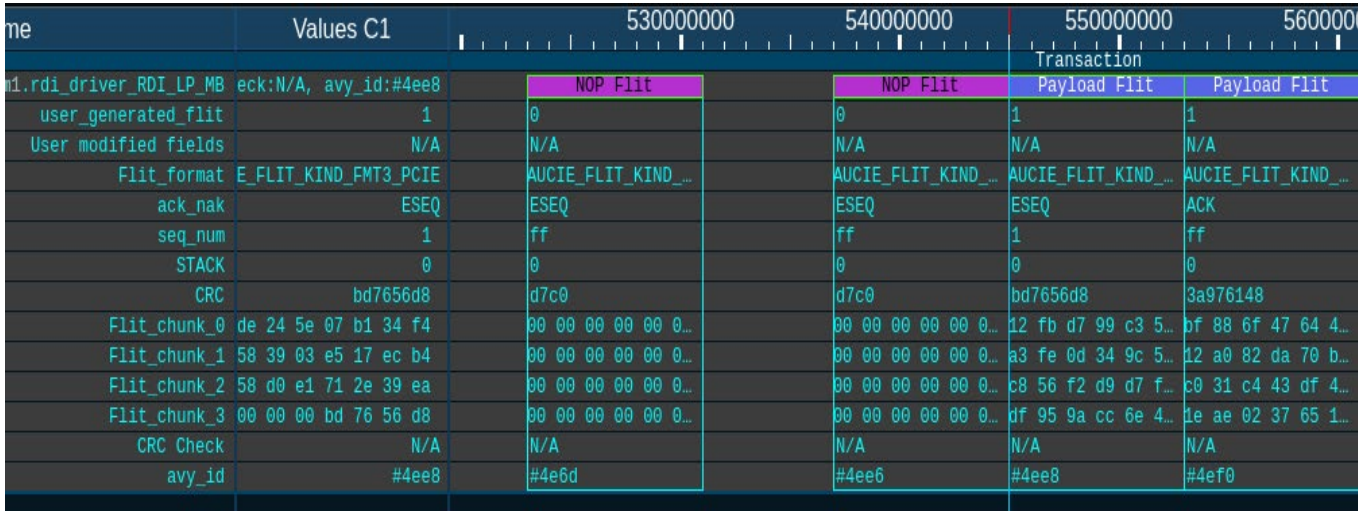
❑ Protocol Aware Debug

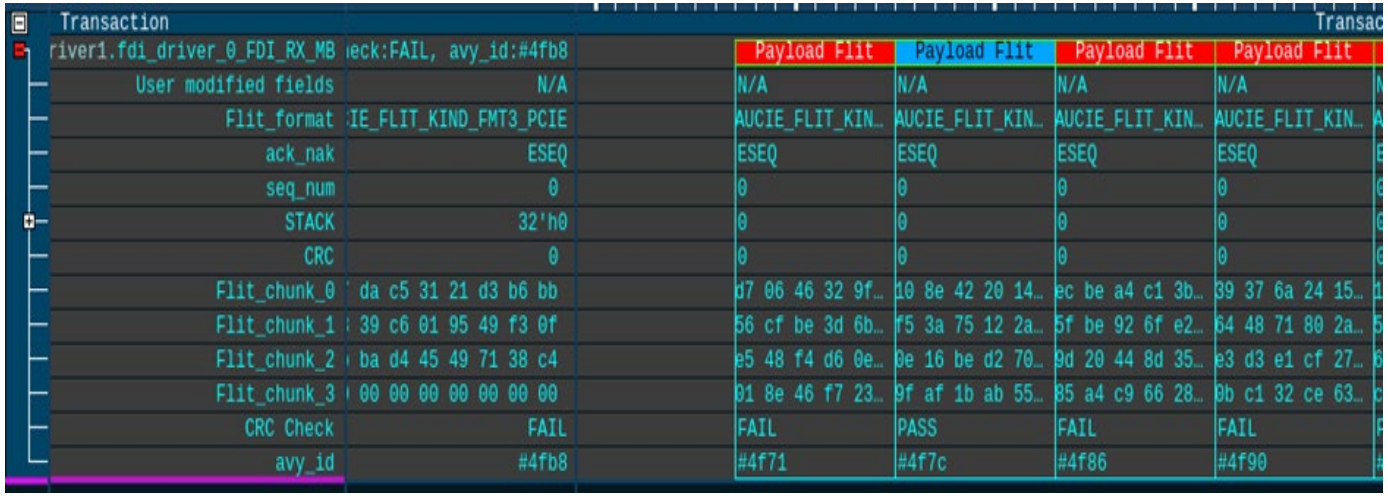
- Transactions are recorded on various interfaces at different layers
- Transaction Association – associate Management Flits with corresponding MPMS



Proposed VIP Architecture

- **Protocol Aware Debug (contd…)**
  - Transaction color differentiation for
    - transmitted and received traffic
    - BFM and user generated traffic
    - Flits with error

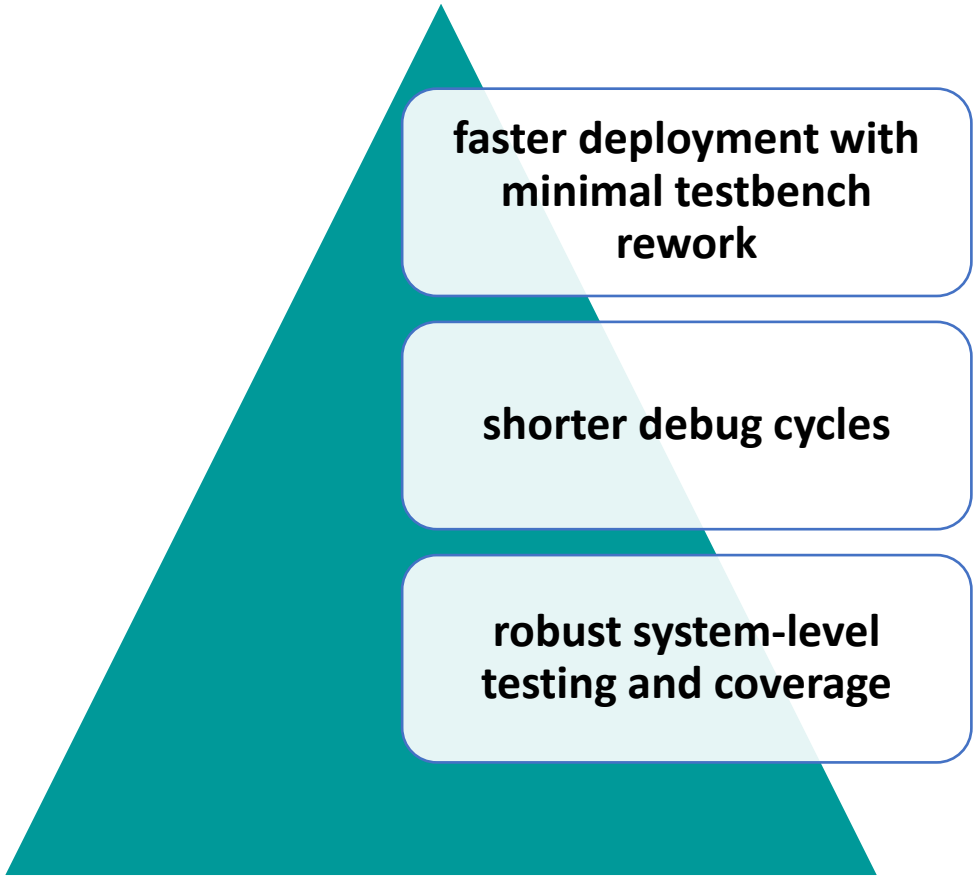




Putting it to work: Case Study

- **This solution was deployed in multiple customer SiP projects with varied requirements**
  - Simple/mesh topology
  - Direct routing/ route through
  - Diverse management protocols
  - Diverse roles – DUT as management director, VIP as management director

RESULTS



Thank You

Visit us at Siemens EDA booth!