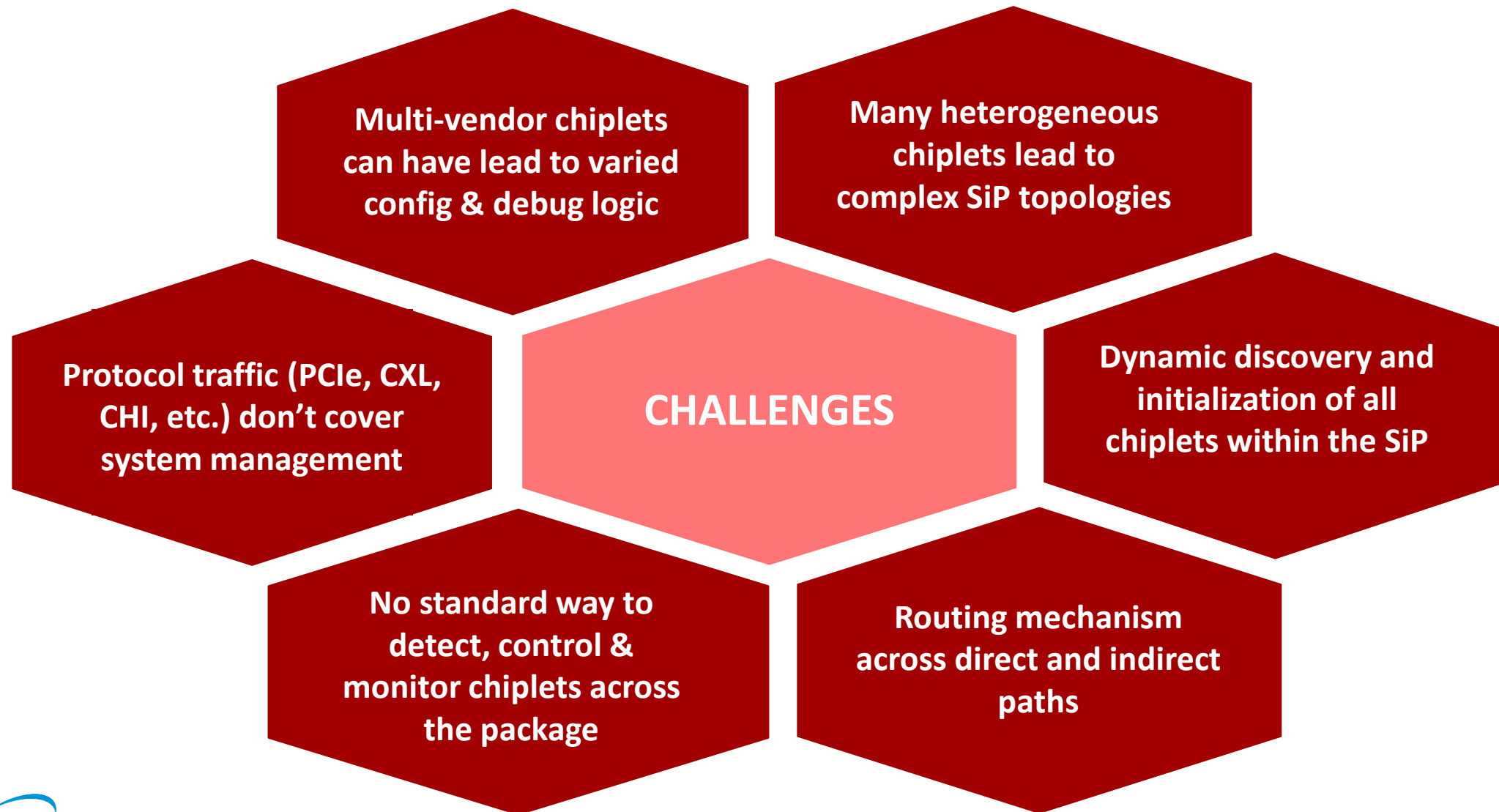


# Mastering UCle 2.0: Overcoming Fabric Management Hurdles for Chiplet Integration

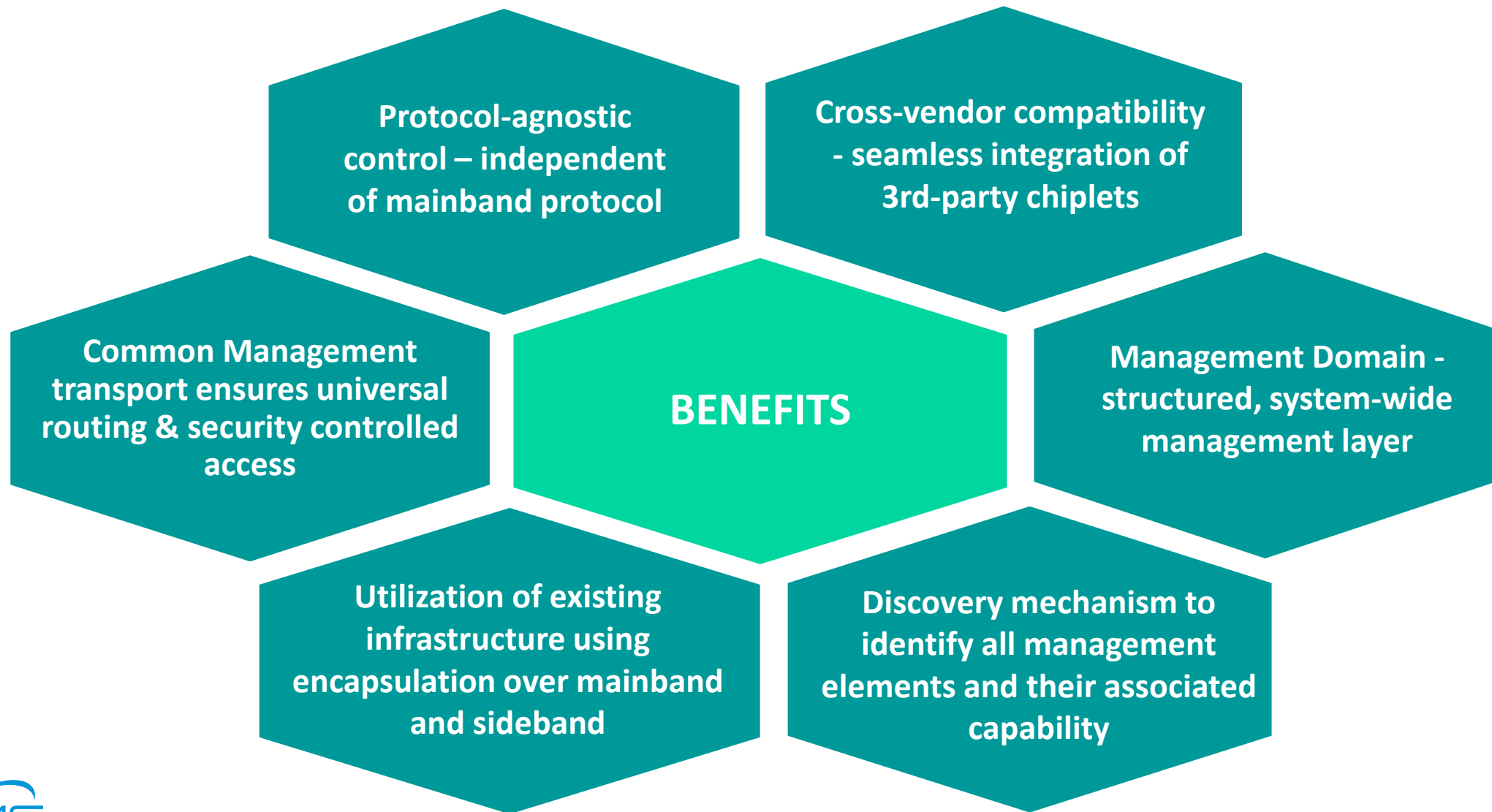
Prashant Dixit & Ujjwal Negi  
Siemens EDA

# Why Manageability Matters in a Chiplet World



# Why Manageability Matters in a Chiplet World

---

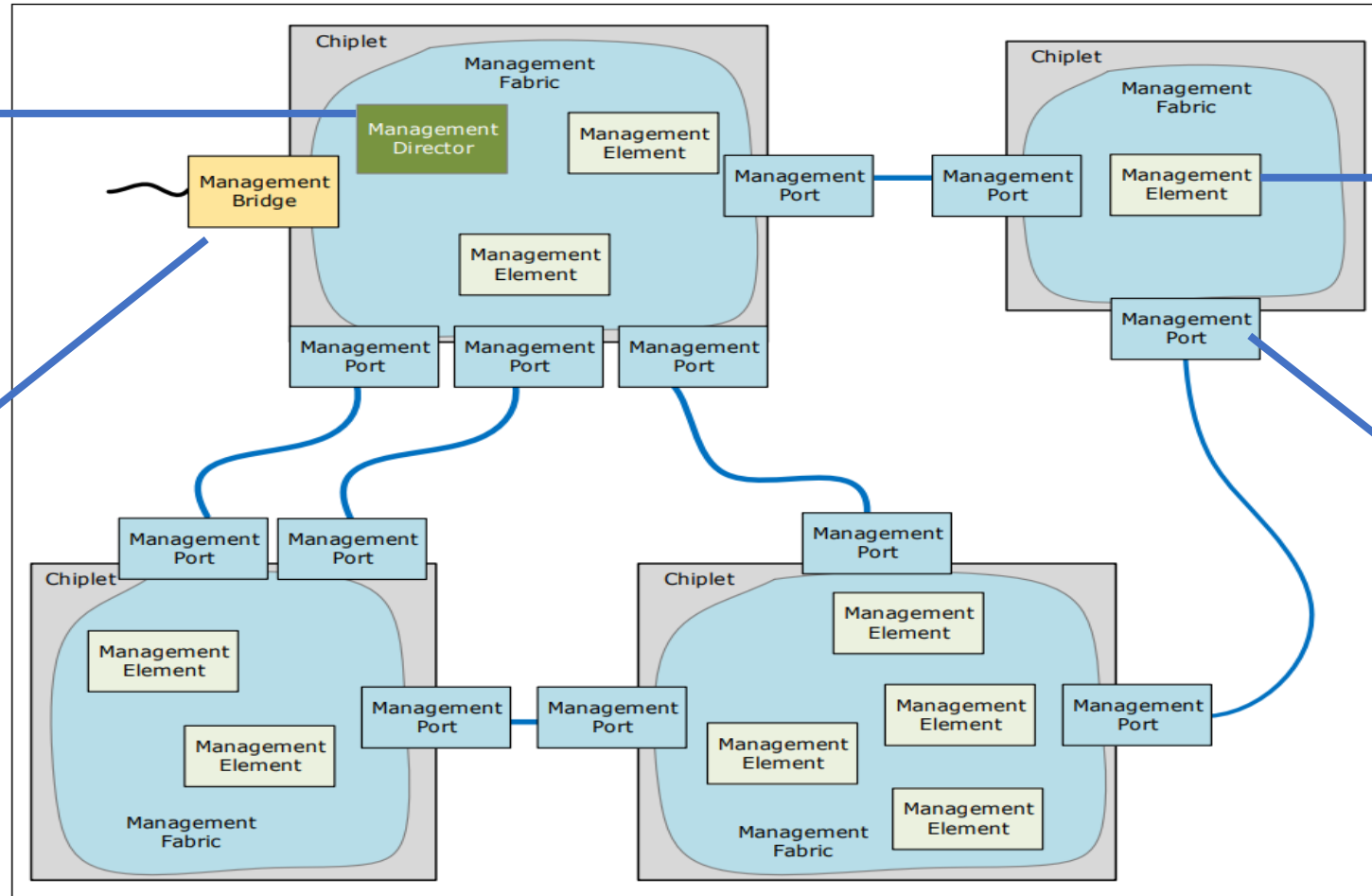


# Inside UCle 2.0: A Layered Management Architecture

## ❑ 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

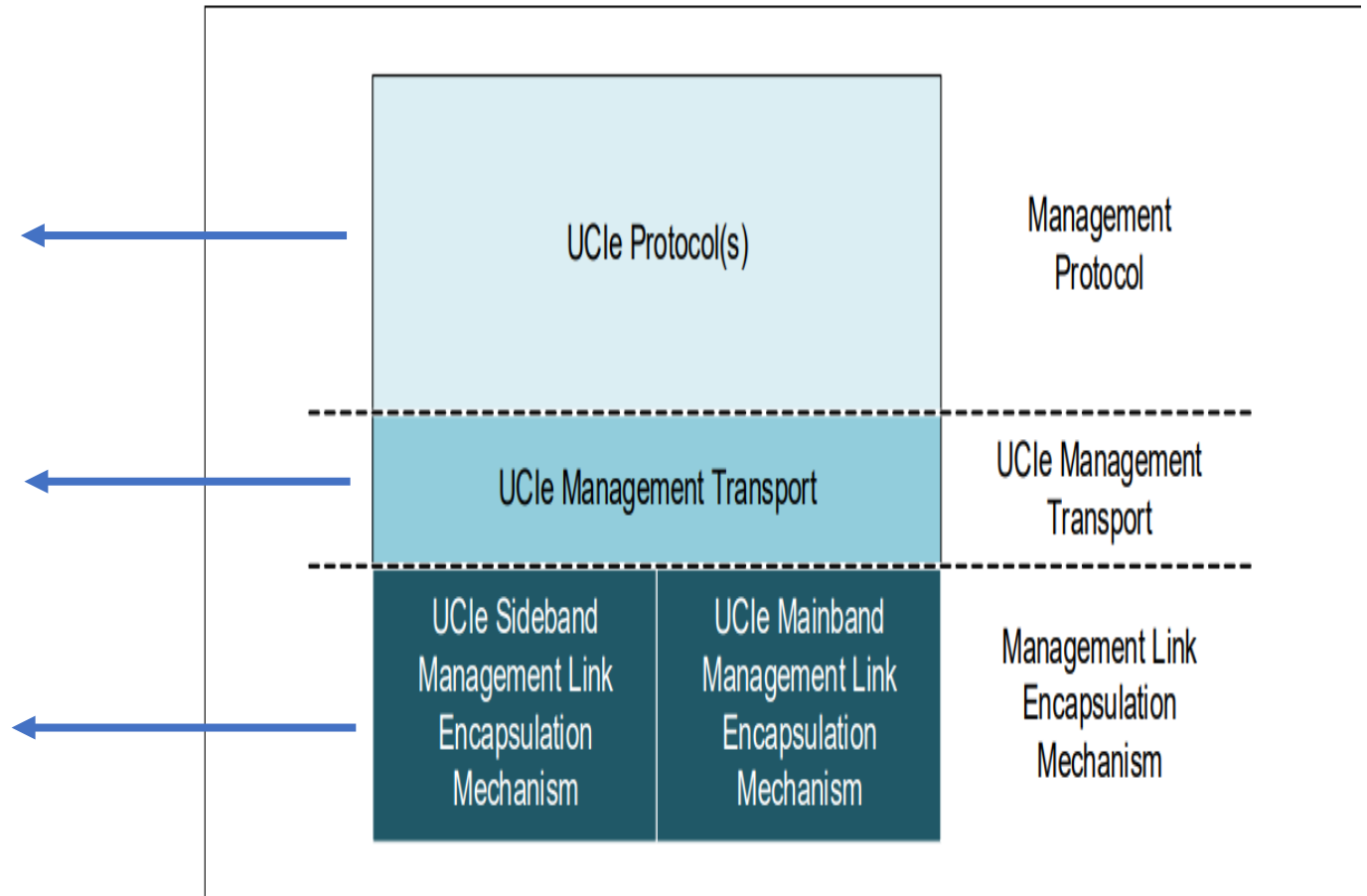
# Inside UCle 2.0: A Layered Management Architecture

## ❑ 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

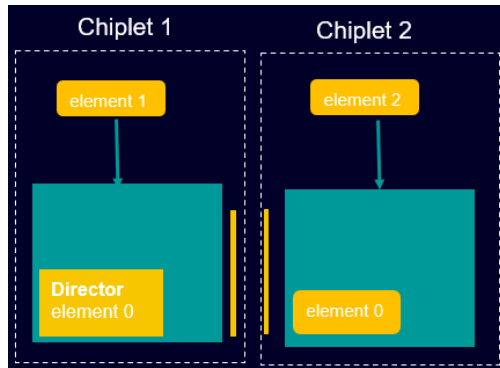
MTP packets are then encapsulated inside the Management Port Message (**MPM**) which handle segmentation and credit flow, that are packed into **management flits** over mainband



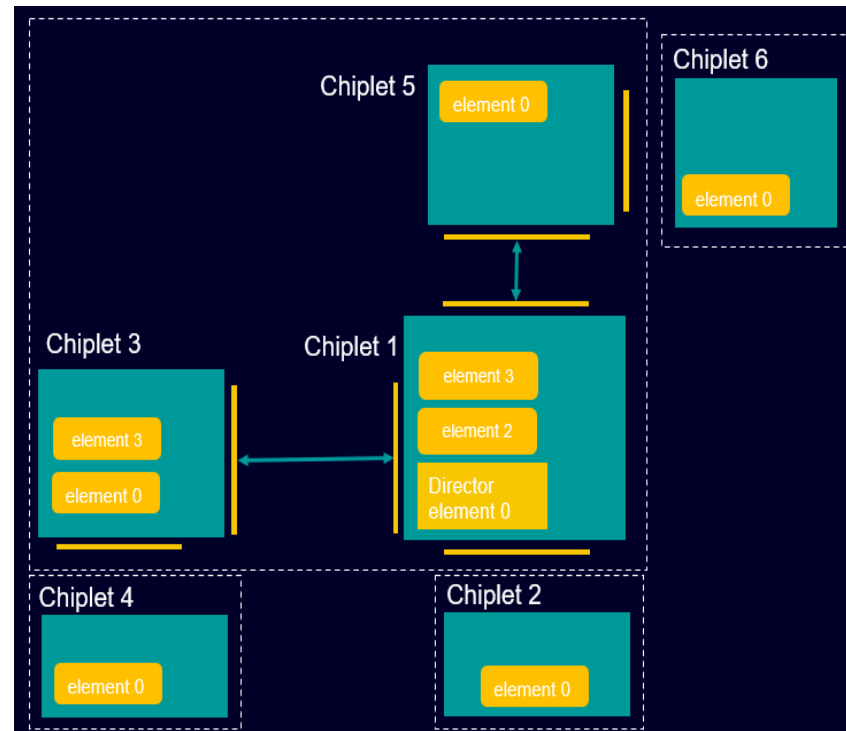
# Verification Challenges

## ❑ Management Domain Configuration

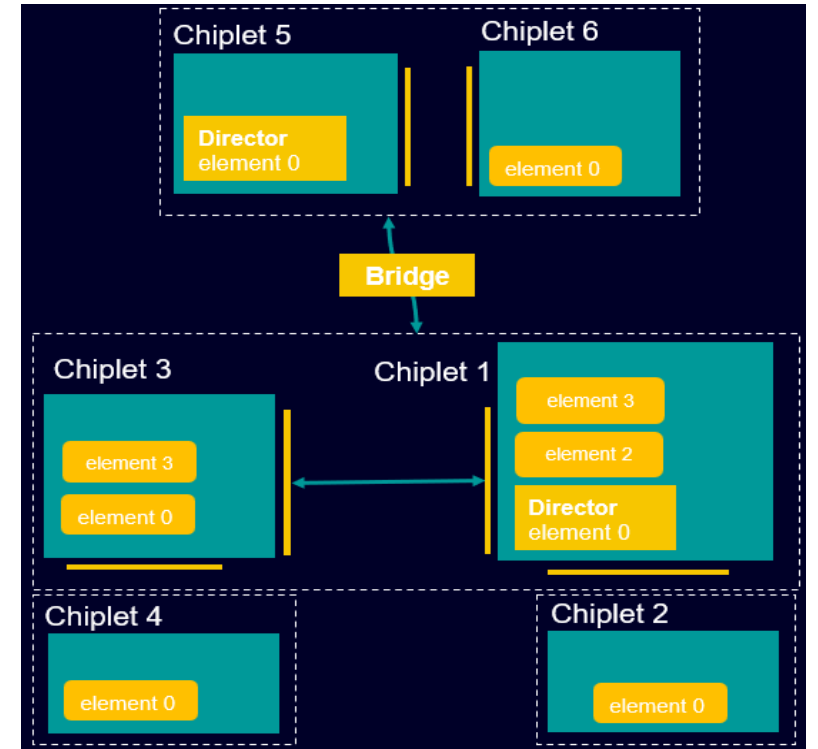
- SiPs vary: point-to-point, daisy chain, multi-chiplet, multi-domain topologies
- Need flexible verification to track multiple management elements & ports per chiplet



**P2P**



**Multi-chiplet single domain**



**Multi-chiplet multi-domain**

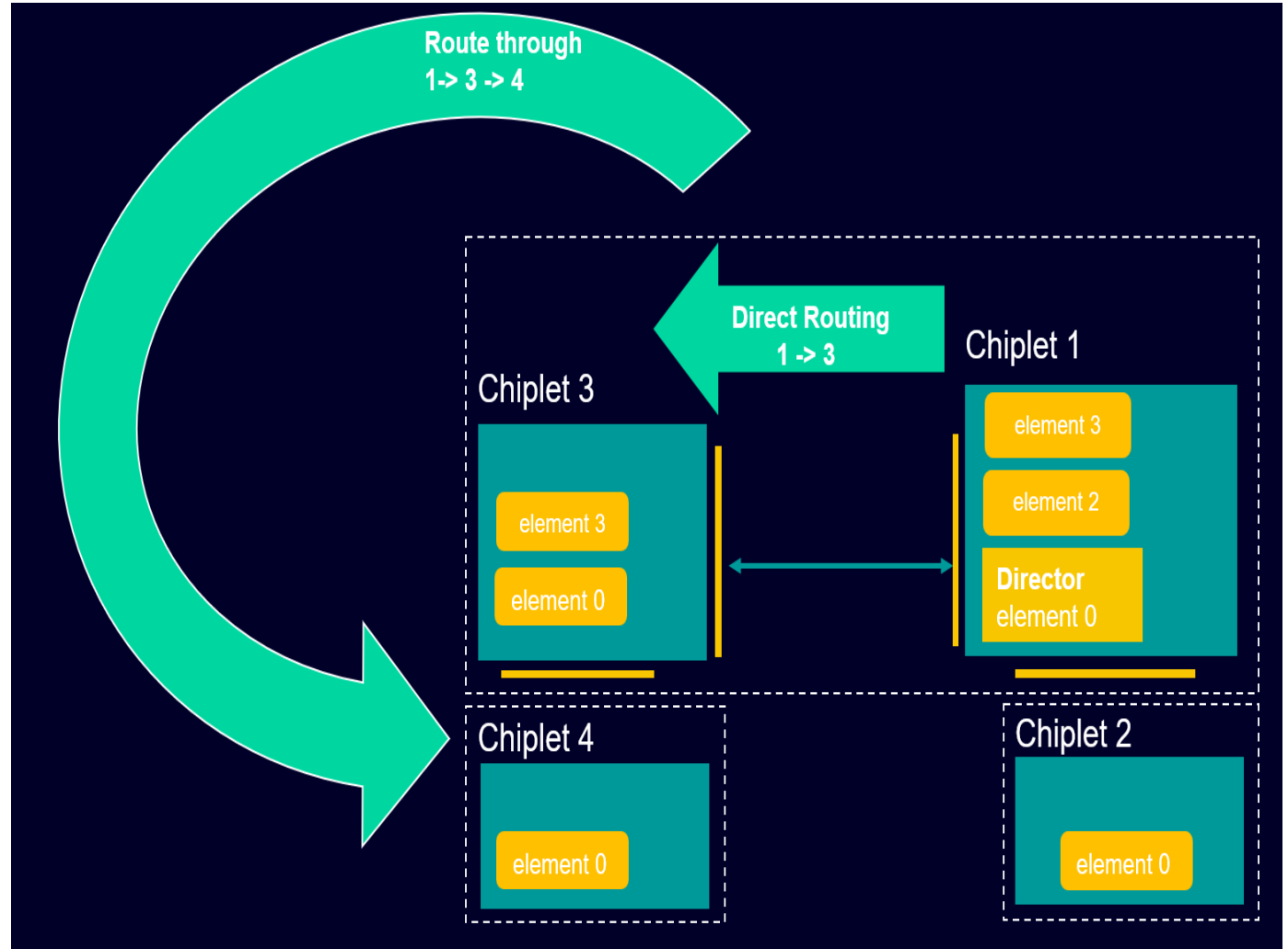
# Verification Challenges

## ❑ 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

1 <sup>st</sup> Segment <sup>b</sup> — 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
2 <sup>nd</sup> Segment <sup>b</sup> — 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
3 <sup>rd</sup> Segment <sup>b</sup> — 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



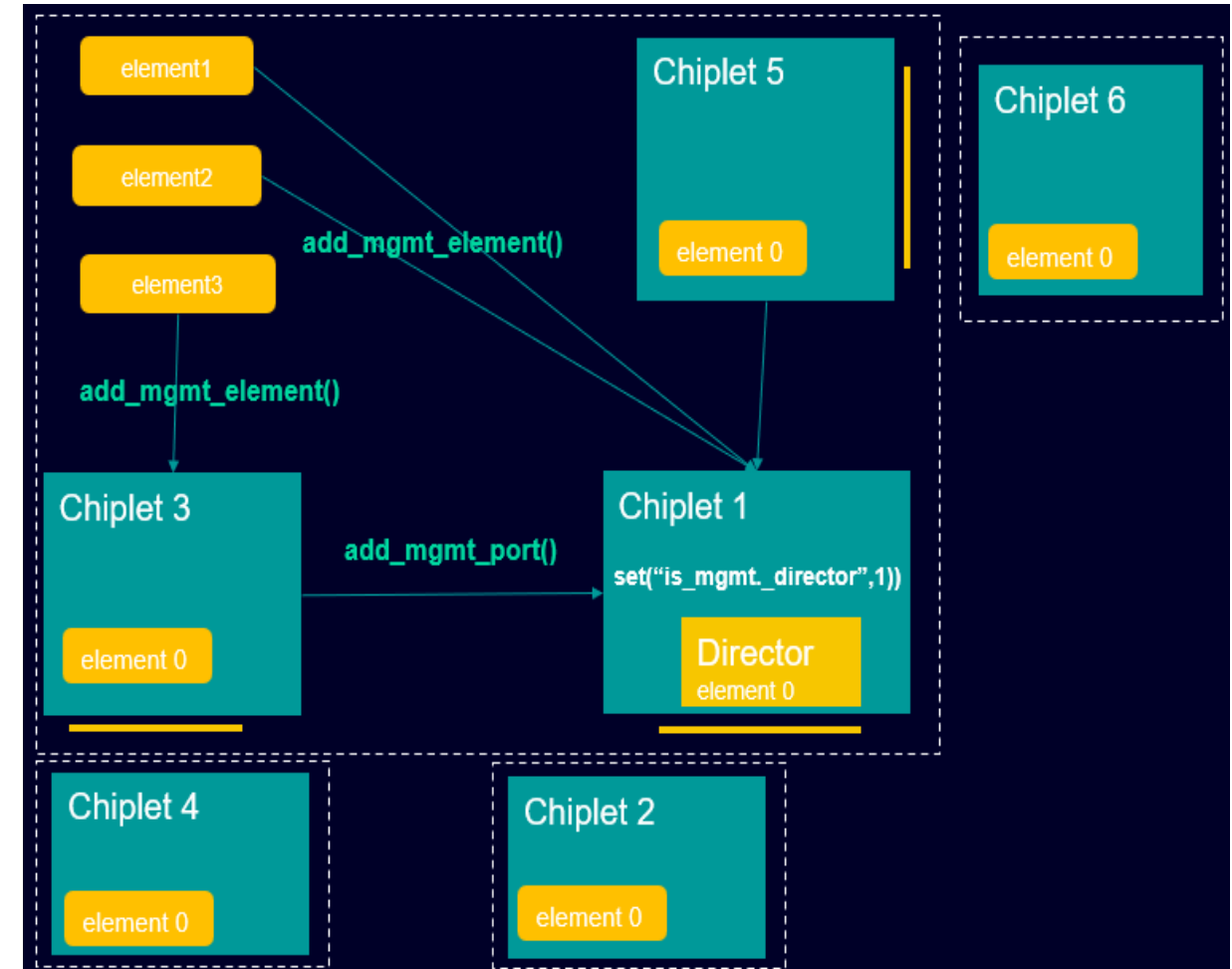
# Proposed VIP Architecture

## ❑ 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.

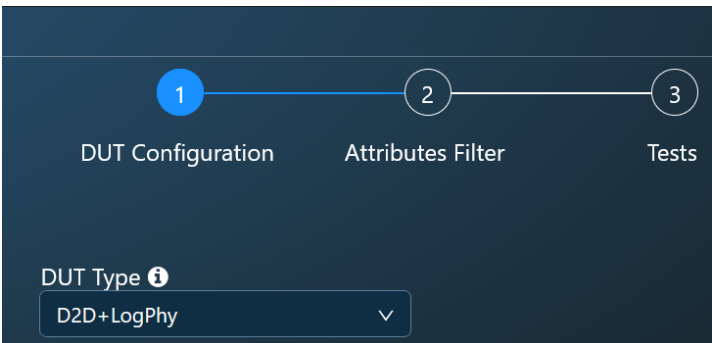
## ❑ Exhaustive Compliance Checks

- 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 sections and tests for targeted testing using smart tools like VIQ



✓ — 2 — 3

DUT Configuration   Attributes Filter   Tests

#	Name	Description
▼ 3	Manageability	This section covers tests related to initialization of management transport path, and encapsulation, p
✓ 3.1	Configuration	This section covers tests related to configuration space of Management Elements.
✓ 3.2	Encapsulation	This section covers tests related to encapsulation of MTP inside sideband packet/flit and their proces
✓ 3.3	Initialization	This section covers tests related to initialization of management transport path over sideband.
✓ 3.4	MTP	This section covers tests related to processing of Management Transport Packet.
✓ 3.5	Routing	This section covers tests related to routing rules for MTP based on network ID.

✓ — ✓ — 3

DUT Configuration   Attributes Filter   Tests

#	Name	Spec Sections Number	Spec Sections Title	Description
▼ 3	Manageability			This section covers tests related to ini
▼ 3.1	Configuration			This section covers tests related to co
✓ 3.1.1	aucie_uvm_mtp_retain_link_seq	8.1.3.6.2.1, 8.2.5.1.3		To verify the 'Retrain Link' functionali
▼ 3.2	Encapsulation			This section covers tests related to en
✓ 3.2.1	aucie_uvm_mpm_check_valid_rxqid_seq	8.2.4.1		Check if the DUT sends successful con
✓ 3.2.2	aucie_uvm_mpm_invlid_rxqid_seq	8.2.4.1		Check if the DUT discards the MPM wi

# Proposed VIP Architecture

## ❑ End-to-End Debug and Traceability

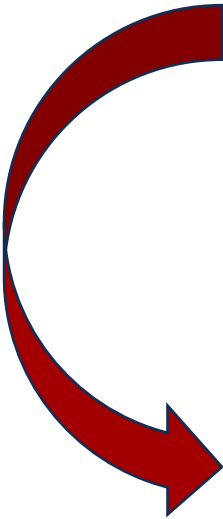
- Trackers associate : MTP → MPM → Management Flit
- Enables root-cause debug from transaction to low-level interface

```
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/auviet 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/auviet 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 00000000
2008 Chunk1: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2009 Chunk2: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2010 Chunk3: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
2011 4B RSVD: 00000000 | 4B CRD: 00000000 | 10B RSVD: 000000000000000000000000 | CRC0: 0000 | CRC1: 0000
test results/auviet 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



Signal Name	Values C	71870000000	71875000000	71880000000	71885000000
Transaction		Management Flit		Transaction	
_driver1.fdi_driver_0_FDI_LP_MB	d:#3844	Management Flit		Management Flit	
user_generated_flit	1	1		1	
User modified fields	N/A	N/A		N/A	
Flit_format	MT5_MTP	AUCIE_FLIT_KIND_FMT5_MTP		AUCIE_FLIT_KIND_FMT5_MTP	
ack_nak	ESEQ	ESEQ		ESEQ	
seq_num	0	0		0	
STACK	0	0		0	
CRC	0	0		0	
Flit_chunk_0	e1 6a	40 80 98 41 00 00 20 01 00 00 1b 63 2f 00 3c 4...		40 80 46 e8 79 60 2e 41 e5 5d 27 f1 b6 a3 7b 1...	
Flit_chunk_1	00 00	e8 c5 98 41 00 00 20 01 00 00 97 2e f0 7f 35 4...		fc 4a b6 2c 70 52 8b 06 00 79 3e a7 07 95 ed 6...	
Flit_chunk_2	59 6c	72 be be 90 36 a2 f1 0c 98 41 00 00 20 01 00 0...		00 00 00 00 00 00 00 00 00 00 00 00 00 00 0...	
Flit_chunk_3	00 00	e9 56 84 13 06 be 5c 15 98 41 00 00 20 01 00 0...		00 00 00 00 00 00 00 00 00 00 00 00 00 00 0...	
CRC Check	N/A	N/A		N/A	
avy_id	#3844	#3844		#3845	
_driver1.fdi_driver_0_TX_MB_MTP	_MTP(4)	fdi_driver_0_TX_MB_MTP(4)		fdi_driver_0_TX_MB_MTP(2)	
0	5e86ae1	AUCIE_SIDE_OP_mpm		AUCIE_SIDE_OP_mpm	
1	cf1a236	AUCIE_SIDE_OP_mpm			
2	55cbe06	AUCIE_SIDE_OP_mpm			
3	fb46877			AUCIE_SIDE_OP_mpm	



# Proposed VIP Architecture

## ❑ Protocol Aware Debug (contd...)

Transaction color differentiation for

- transmitted and received traffic
- BFM and user generated traffic
- Flits with error

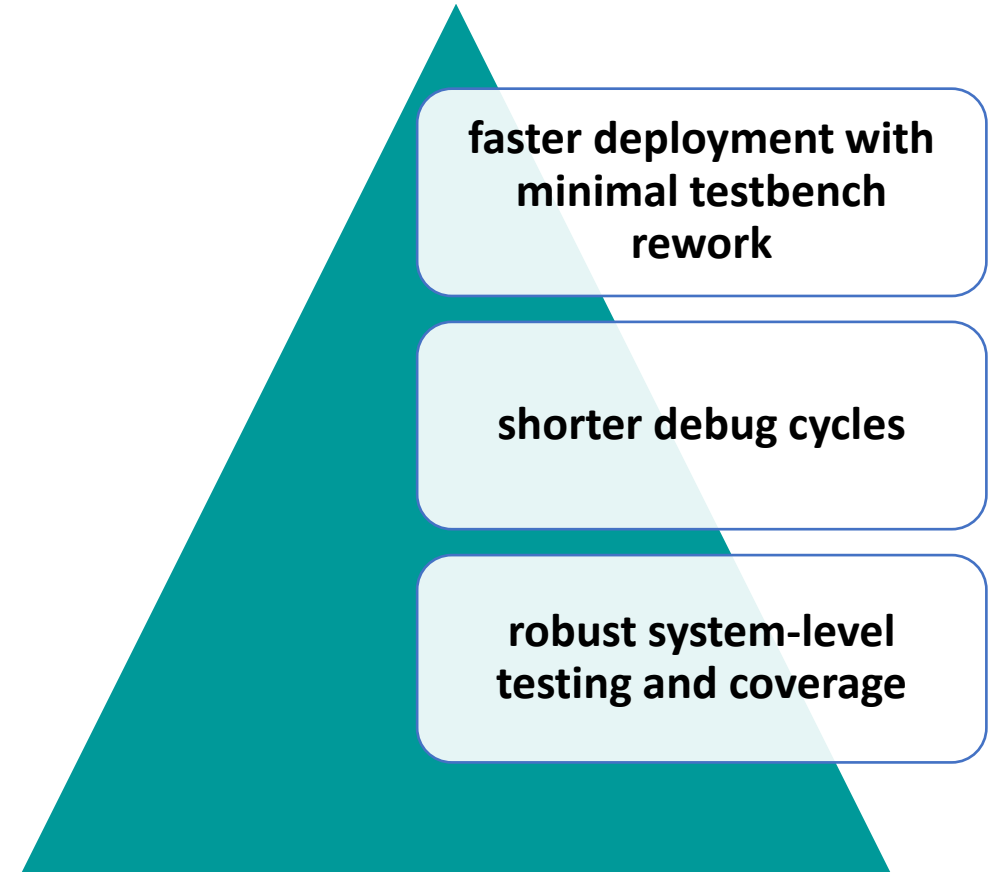
me	Values C1	530000000	540000000	550000000	560000000
		Transaction			
1.rdi_driver_RDI_LP_MB	eck:N/A, avy_id:#4ee8	NOP Flit	NOP Flit	Payload Flit	Payload Flit
user_generated_flit	1	0	0	1	1
User modified fields	N/A	N/A	N/A	N/A	N/A
Flit_format	E_FLIT_KIND_FMT3_PCIE	AUCIE_FLIT_KIND_...	AUCIE_FLIT_KIND_...	AUCIE_FLIT_KIND_...	AUCIE_FLIT_KIND_...
ack_nak	ESEQ	ESEQ	ESEQ	ESEQ	ACK
seq_num	1	ff	ff	1	ff
STACK	0	0	0	0	0
CRC	bd7656d8	d7c0	d7c0	bd7656d8	3a976148
Flit_chunk_0	de 24 5e 07 b1 34 f4	00 00 00 00 00 0...	00 00 00 00 00 0...	12 fb d7 99 c3 5...	bf 88 6f 47 64 4...
Flit_chunk_1	58 39 03 e5 17 ec b4	00 00 00 00 00 0...	00 00 00 00 00 0...	a3 fe 0d 34 9c 5...	12 a0 82 da 70 b...
Flit_chunk_2	58 d0 e1 71 2e 39 ea	00 00 00 00 00 0...	00 00 00 00 00 0...	c8 56 f2 d9 d7 f...	c0 31 c4 43 df 4...
Flit_chunk_3	00 00 00 bd 76 56 d8	00 00 00 00 00 0...	00 00 00 00 00 0...	df 95 9a cc 6e 4...	1e ae 02 37 65 1...
CRC Check	N/A	N/A	N/A	N/A	N/A
avy_id	#4ee8	#4e6d	#4ee6	#4ee8	#4ef0

Transaction		Transaction			
river1.fdi_driver_0_FDI_RX_MB	eck:FAIL, avy_id:#4fb8	Payload Flit	Payload Flit	Payload Flit	Payload Flit
User modified fields	N/A	N/A	N/A	N/A	N/A
Flit_format	IE_FLIT_KIND_FMT3_PCIE	AUCIE_FLIT_KIN...	AUCIE_FLIT_KIN...	AUCIE_FLIT_KIN...	AUCIE_FLIT_KIN...
ack_nak	ESEQ	ESEQ	ESEQ	ESEQ	ESEQ
seq_num	0	0	0	0	0
STACK	32'h0	0	0	0	0
CRC	0	0	0	0	0
Flit_chunk_0	da c5 31 21 d3 b6 bb	d7 06 46 32 9f...	10 8e 42 20 14...	ec be a4 c1 3b...	39 37 6a 24 15...
Flit_chunk_1	39 c6 01 95 49 f3 0f	56 cf be 3d 6b...	f5 3a 75 12 2a...	5f be 92 6f e2...	64 48 71 80 2a...
Flit_chunk_2	ba d4 45 49 71 38 c4	e5 48 f4 d6 0e...	0e 16 be d2 70...	9d 20 44 8d 35...	e3 d3 e1 cf 27...
Flit_chunk_3	00 00 00 00 00 00	01 8e 46 f7 23...	9f af 1b ab 55...	85 a4 c9 66 28...	0b c1 32 ce 63...
CRC Check	FAIL	FAIL	PASS	FAIL	FAIL
avy_id	#4fb8	#4f71	#4f7c	#4f86	#4f90

# 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



# Thank You

Visit us at Siemens EDA booth!