



SpaceWire Codec VIP: an innovative architecture of UVM-based Verification Environment

Simone Vagaggini^{1,2}, Marco Trafeli², Daniele Davalle², Roberto Ciardi^{1,2}, Lucana Santos³, Pietro Nannipieri¹, Luca Fanucci¹

¹Dept. Of Information Engineering, University of Pisa, Via Caruso 16, Pisa, Italy

²IngeniArs S.r.l., Via Ponte a Piglieri 8, Pisa, Italy

³European Space Research and Technology Centre, European Space Agency, Keplerlaan 1, Noordwijk, The Netherlands

- Introduction

- Introduction
- SpaceWire Systems Functional Verification

- Introduction
- SpaceWire Systems Functional Verification
- UVM-based SpaceWire Codec Twin Model

- Introduction
- SpaceWire Systems Functional Verification
- UVM-based SpaceWire Codec Twin Model
- UVM-based SpaceWire Data-Strobe Error Injection Model

- Introduction
- SpaceWire Systems Functional Verification
- UVM-based SpaceWire Codec Twin Model
- UVM-based SpaceWire Data-Strobe Error Injection Model
- UVM-based Verification Environment Architecture

- Introduction
- SpaceWire Systems Functional Verification
- UVM-based SpaceWire Codec Twin Model
- UVM-based SpaceWire Data-Strobe Error Injection Model
- UVM-based Verification Environment Architecture
- Verification Campaign

- Introduction
- SpaceWire Systems Functional Verification
- UVM-based SpaceWire Codec Twin Model
- UVM-based SpaceWire Data-Strobe Error Injection Model
- UVM-based Verification Environment Architecture
- Verification Campaign
- UVM-based Approach Advantages

- Introduction
- SpaceWire Systems Functional Verification
- UVM-based SpaceWire Codec Twin Model
- UVM-based SpaceWire Data-Strobe Error Injection Model
- UVM-based Verification Environment Architecture
- Verification Campaign
- UVM-based Approach Advantages
- Results and Conclusions

- Introduction
- SpaceWire Systems Functional Verification
- UVM-based SpaceWire Codec Twin Model
- UVM-based SpaceWire Data-Strobe Error Injection Model
- UVM-based Verification Environment Architecture
- Verification Campaign
- UVM-based Approach Advantages
- Results and Conclusions
- Q&A

Universal Verification Methodology (UVM) Verification Intellectual Property (VIP) for Functional Verification of any SpaceWire Codec

Universal Verification Methodology (UVM) Verification Intellectual Property (VIP) for Functional Verification of any SpaceWire Codec

- Developed by IngeniArs S.r.l.



Universal Verification Methodology (UVM) Verification Intellectual Property (VIP) for Functional Verification of any SpaceWire Codec

- Developed by IngeniArs S.r.l.
- European Space Agency project



Universal Verification Methodology (UVM) Verification Intellectual Property (VIP) for Functional Verification of any SpaceWire Codec

- Developed by IngeniArs S.r.l.
- European Space Agency project
- Activity goals:
 - Check the compliance of IP Cores with the SpW standard
 - Evaluation of UVM advantages
 - Applicability of UVM to space systems



Universal Verification Methodology (UVM) Verification Intellectual Property (VIP) for Functional Verification of any SpaceWire Codec

- Developed by IngeniArs S.r.l.
- European Space Agency project
- Activity goals:
 - Check the compliance of IP Cores with the SpW standard
 - Evaluation of UVM advantages
 - Applicability of UVM to space systems



Promotion of UVM-based Verification Approach

Functional Verification of a SpaceWire Codec

Functional Verification of a SpaceWire Codec

- Critical and complex phase

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling
 - NULL codes transmission

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling
 - NULL codes transmission
 - Errors recognition

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling
 - NULL codes transmission
 - Errors recognition
- Common approach: direct test-bench

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling
 - NULL codes transmission
 - Errors recognition
- Common approach: direct test-bench
 - Manual calibration of inputs required

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling
 - NULL codes transmission
 - Errors recognition
- Common approach: direct test-bench
 - Manual calibration of inputs required
 - Manual check of outputs required

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling
 - NULL codes transmission
 - Errors recognition
- Common approach: direct test-bench
 - Manual calibration of inputs required
 - Manual check of outputs required
 - Inefficient and time-consuming

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling
 - NULL codes transmission
 - Errors recognition
- Common approach: direct test-bench
 - Manual calibration of inputs required
 - Manual check of outputs required
 - Inefficient and time-consuming
- Proposed approach: advanced and automated Verification Environment

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling
 - NULL codes transmission
 - Errors recognition
- Common approach: direct test-bench
 - Manual calibration of inputs required
 - Manual check of outputs required
 - Inefficient and time-consuming
- Proposed approach: advanced and automated Verification Environment
 - Based on SpW Codec Twin Model

Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling
 - NULL codes transmission
 - Errors recognition
- Common approach: direct test-bench
 - Manual calibration of inputs required
 - Manual check of outputs required
 - Inefficient and time-consuming
- Proposed approach: advanced and automated Verification Environment
 - Based on SpW Codec Twin Model
 - Fully compliant with Universal Verification Methodology (UVM)

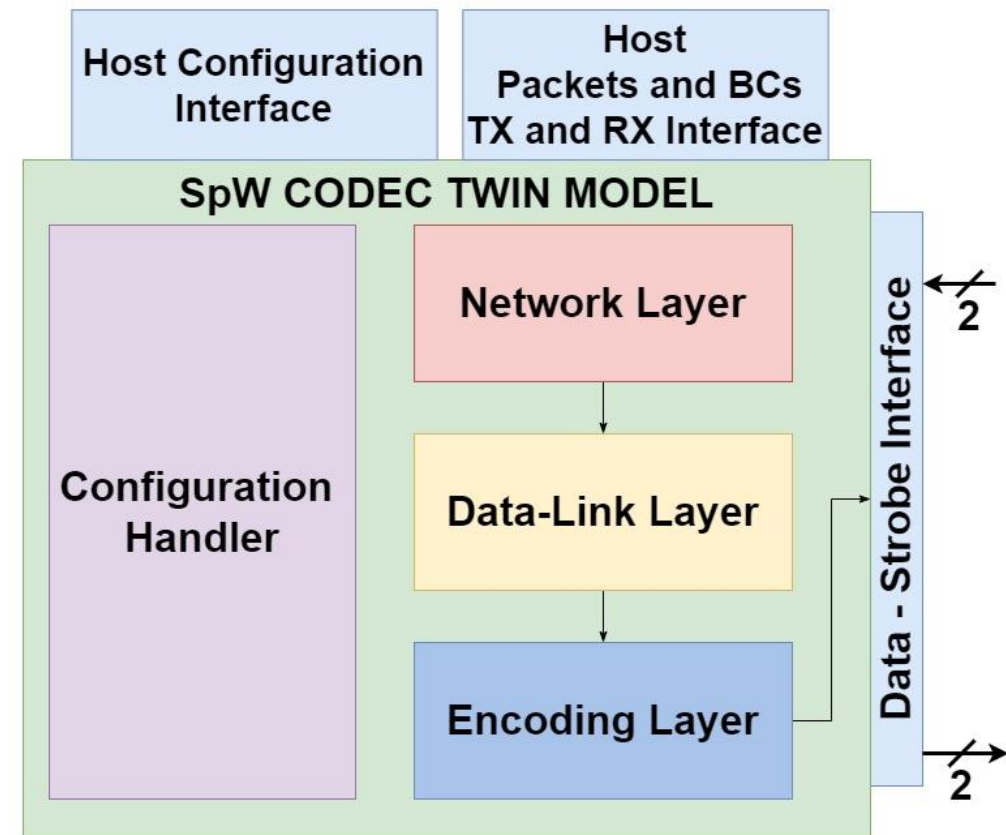


Functional Verification of a SpaceWire Codec

- Critical and complex phase
 - Handshake for link initialization
 - TX and RX credit handling
 - NULL codes transmission
 - Errors recognition
- Common approach: direct test-bench
 - Manual calibration of inputs required
 - Manual check of outputs required
 - Inefficient and time-consuming
- Proposed approach: advanced and automated Verification Environment
 - Based on SpW Codec Twin Model
 - Fully compliant with Universal Verification Methodology (UVM)
 - Capable of verifying any SpW Codec

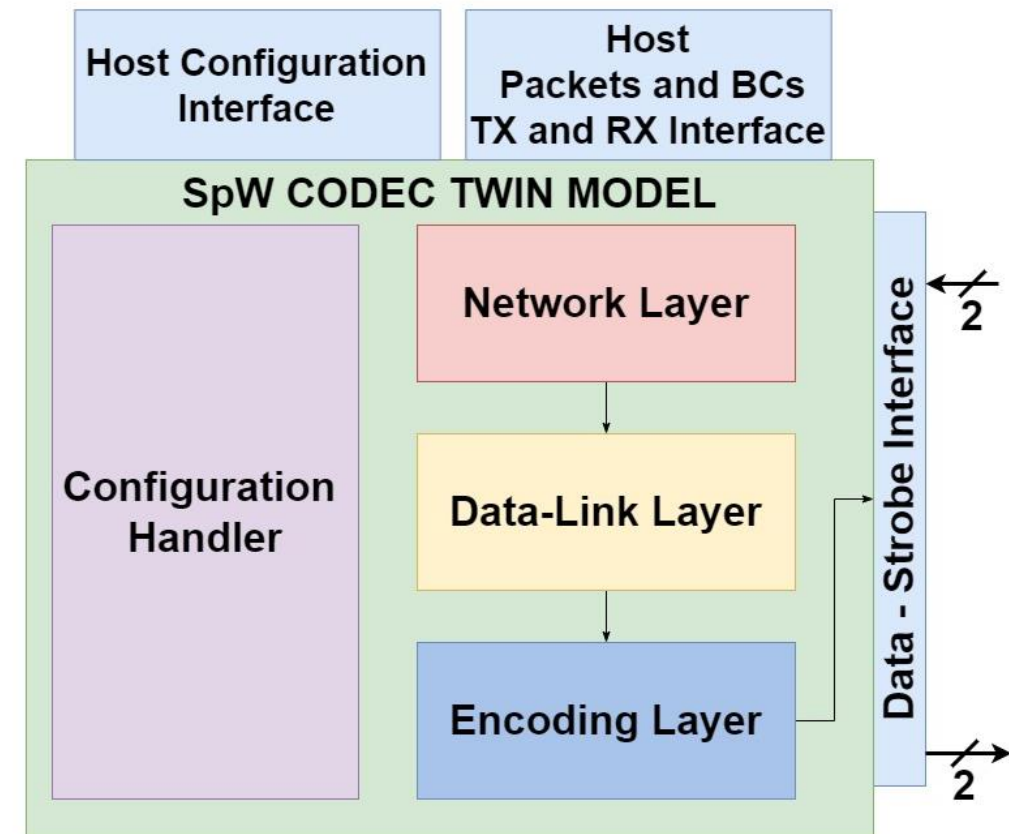


UVM-BASED SPACEWIRE CODEC TWIN MODEL



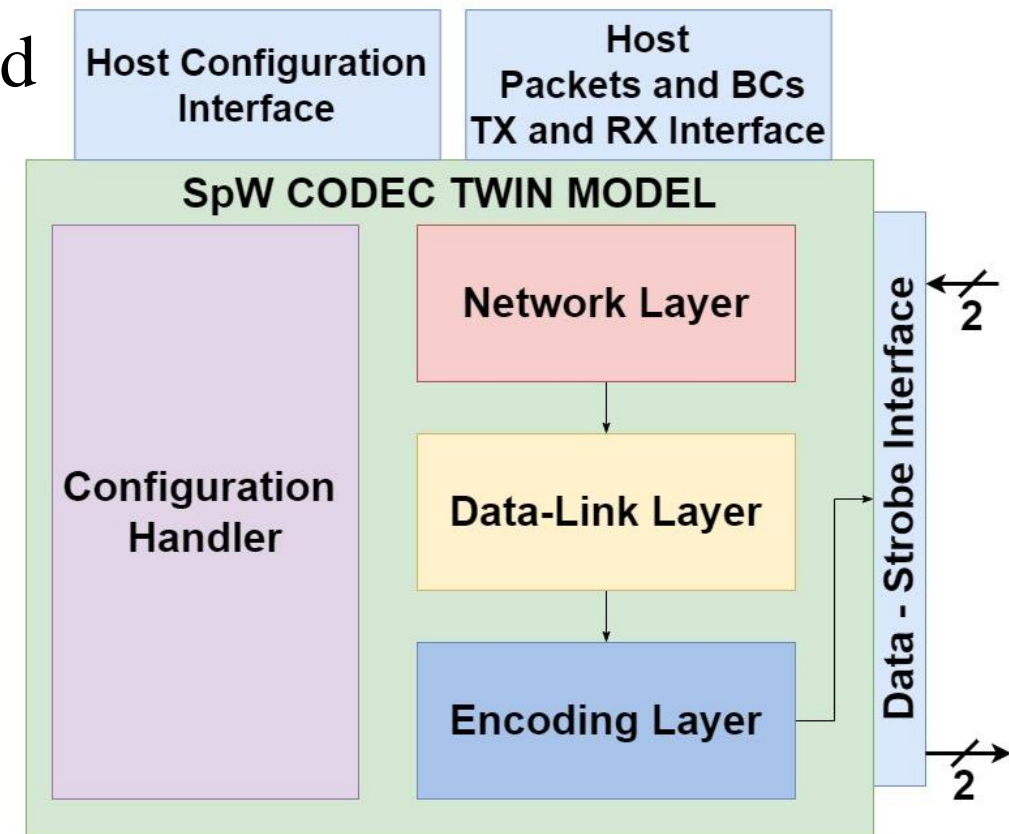
UVM-BASED SPACEWIRE CODEC TWIN MODEL

- Emulates the ideal behavior of a SPW Codec



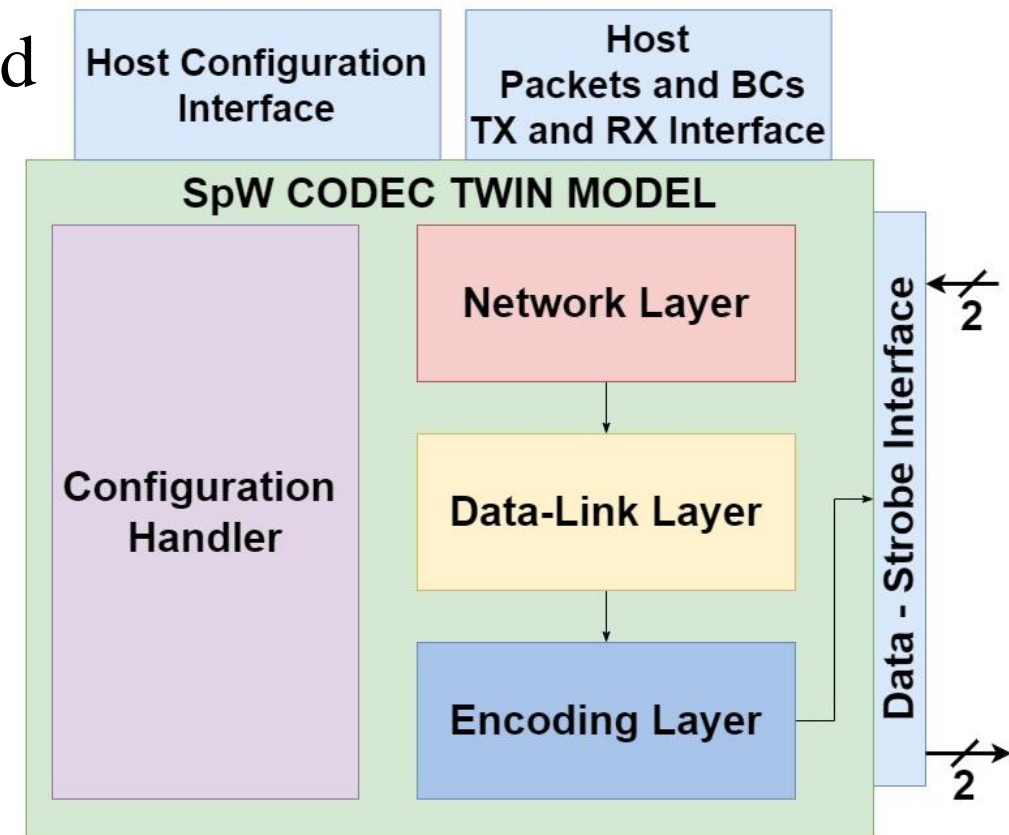
UVM-BASED SPACEWIRE CODEC TWIN MODEL

- Emulates the ideal behavior of a SPW Codec
- Fully compliant with Rev.1 of SpW Standard



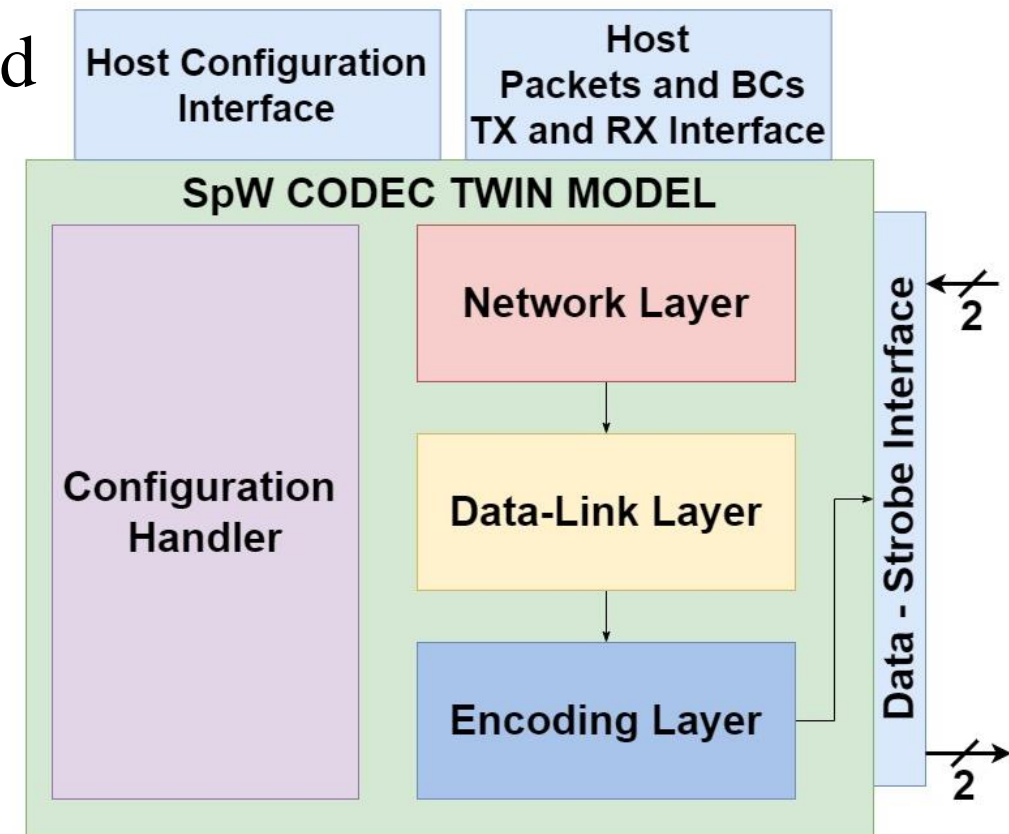
UVM-BASED SPACEWIRE CODEC TWIN MODEL

- Emulates the ideal behavior of a SPW Codec
- Fully compliant with Rev.1 of SpW Standard
- Host interface
 - Packets and BC transmission and reception
 - Configuration changes and readings



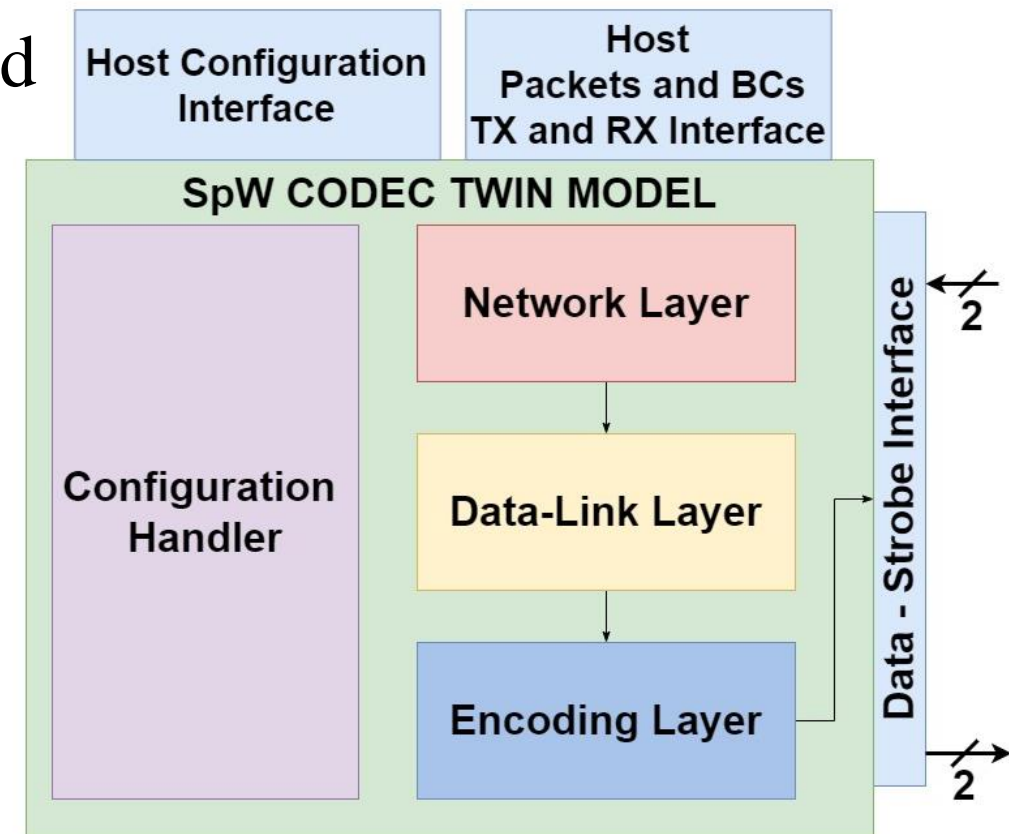
UVM-BASED SPACEWIRE CODEC TWIN MODEL

- Emulates the ideal behavior of a SPW Codec
- Fully compliant with Rev.1 of SpW Standard
- Host interface
 - Packets and BC transmission and reception
 - Configuration changes and readings
- Data-Strobe interface
 - Communication with DUT or any system with SpW interface



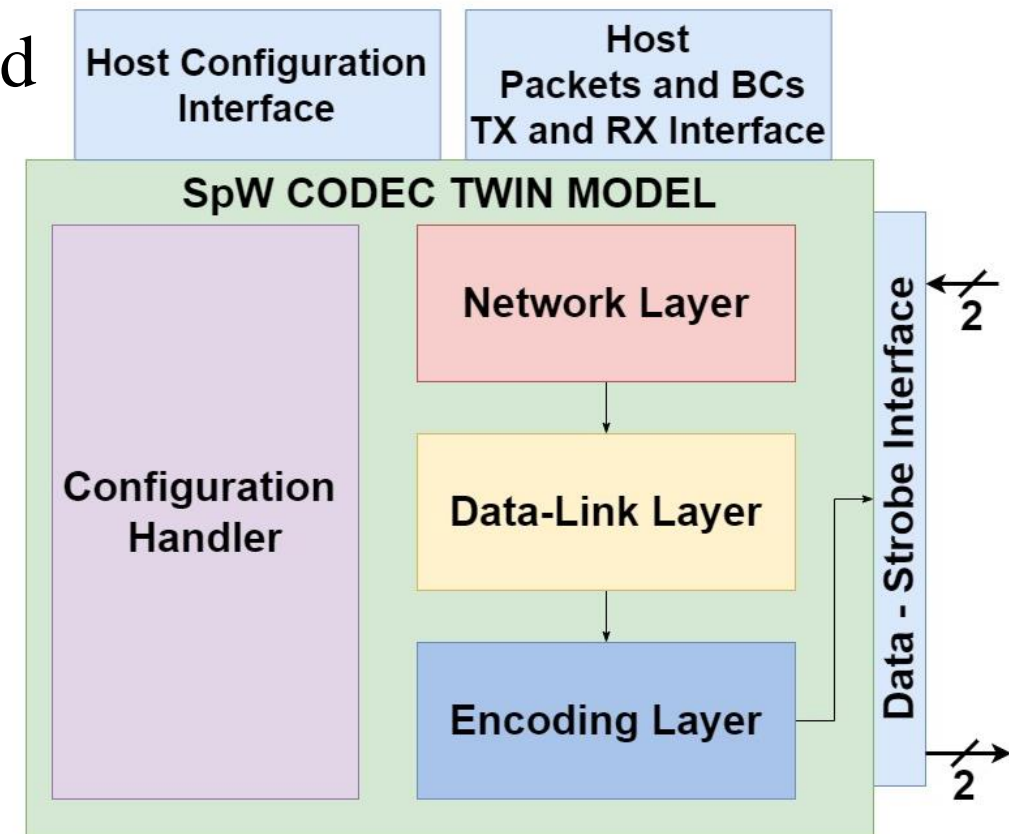
UVM-BASED SPACEWIRE CODEC TWIN MODEL

- Emulates the ideal behavior of a SPW Codec
- Fully compliant with Rev.1 of SpW Standard
- Host interface
 - Packets and BC transmission and reception
 - Configuration changes and readings
- Data-Strobe interface
 - Communication with DUT or any system with SpW interface
- Developed in SystemVerilog HVL

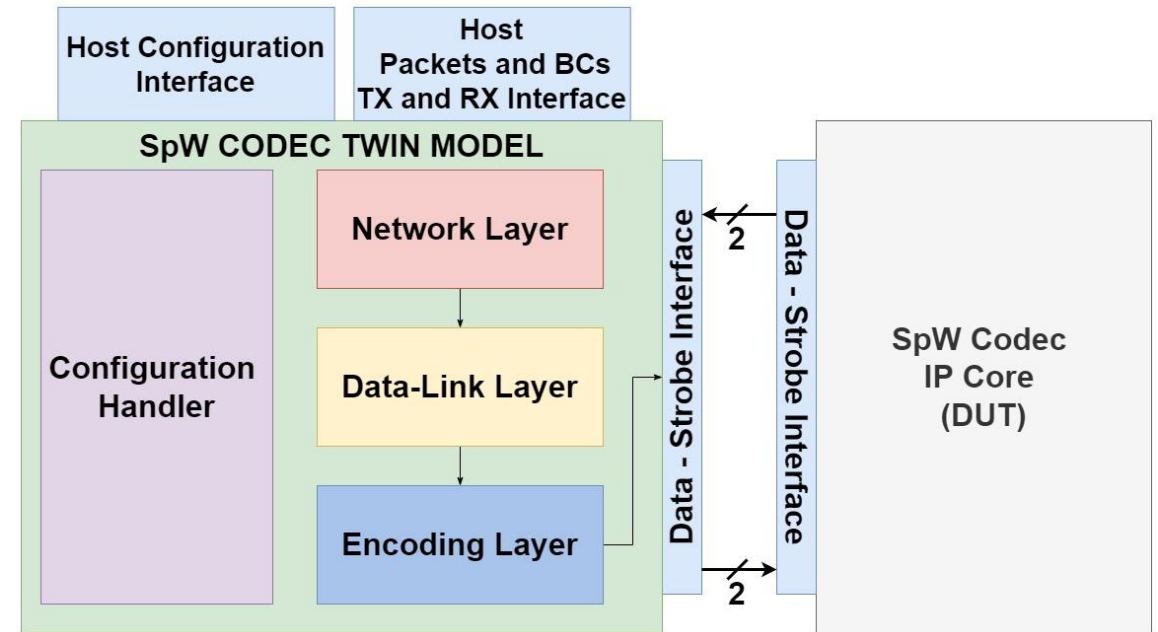


UVM-BASED SPACEWIRE CODEC TWIN MODEL

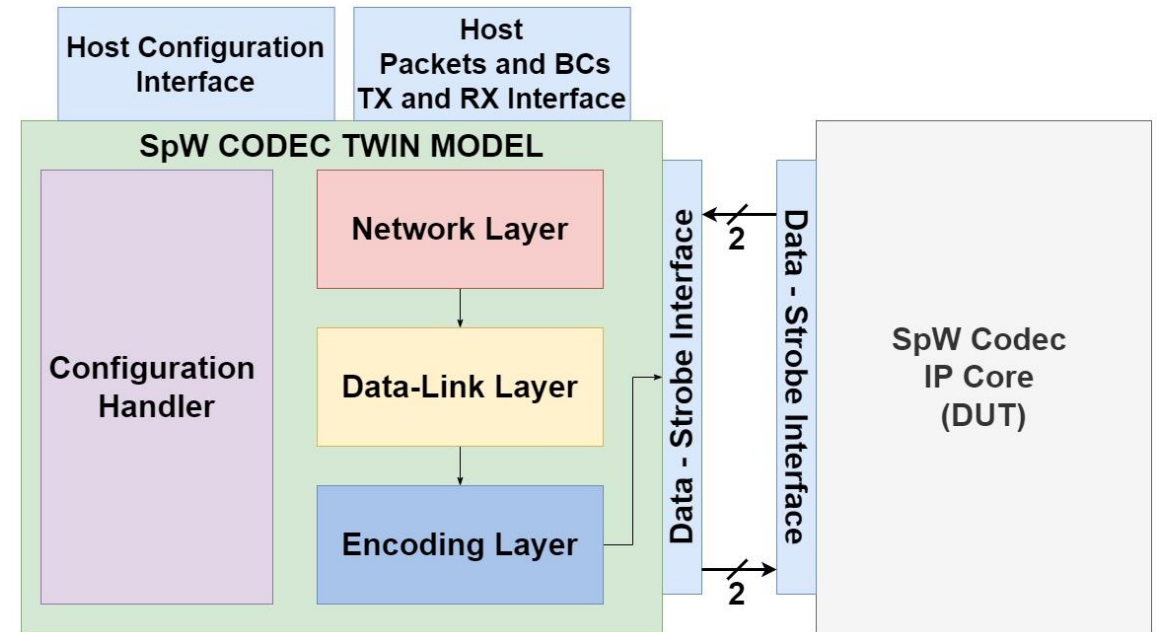
- Emulates the ideal behavior of a SPW Codec
- Fully compliant with Rev.1 of SpW Standard
- Host interface
 - Packets and BC transmission and reception
 - Configuration changes and readings
- Data-Strobe interface
 - Communication with DUT or any system with SpW interface
- Developed in SystemVerilog HVL
- Fully compliant with UVM
 - Easily reusable and maintainable



- Direct Communication with DUT

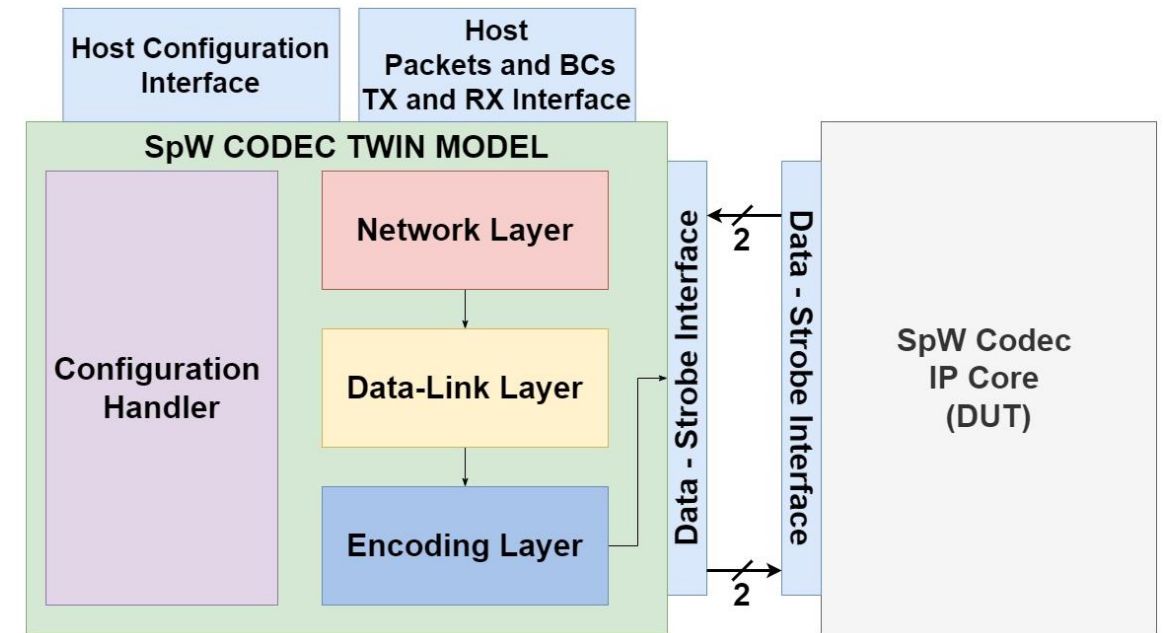


- Direct Communication with DUT
 - Automatic link initialization

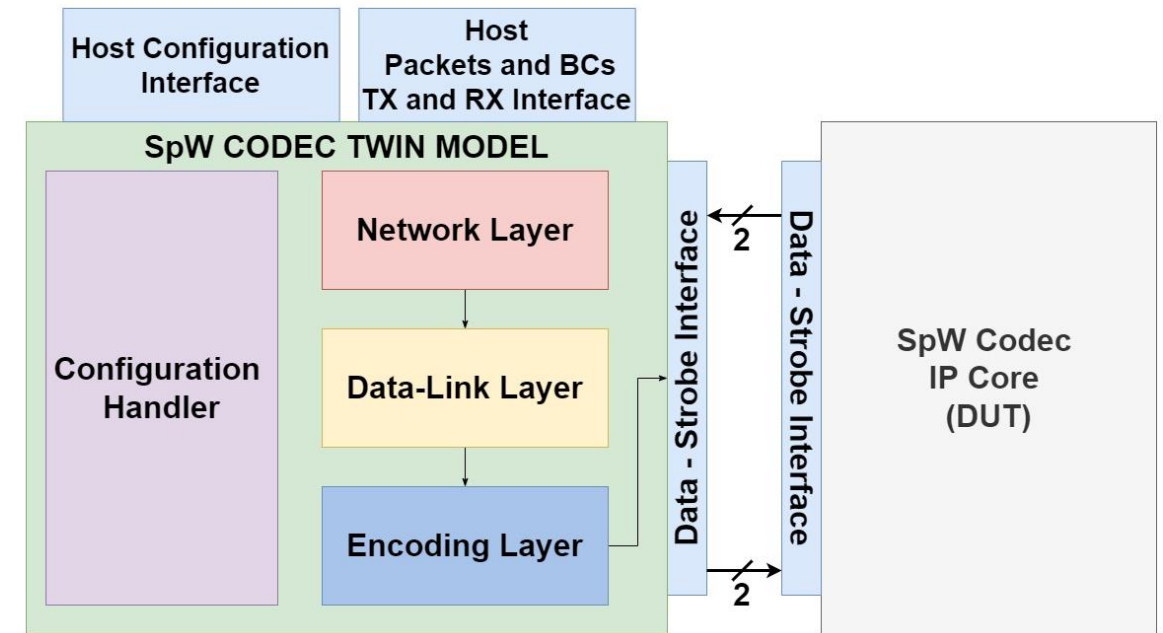


UVM-BASED SPACEWIRE CODEC TWIN MODEL

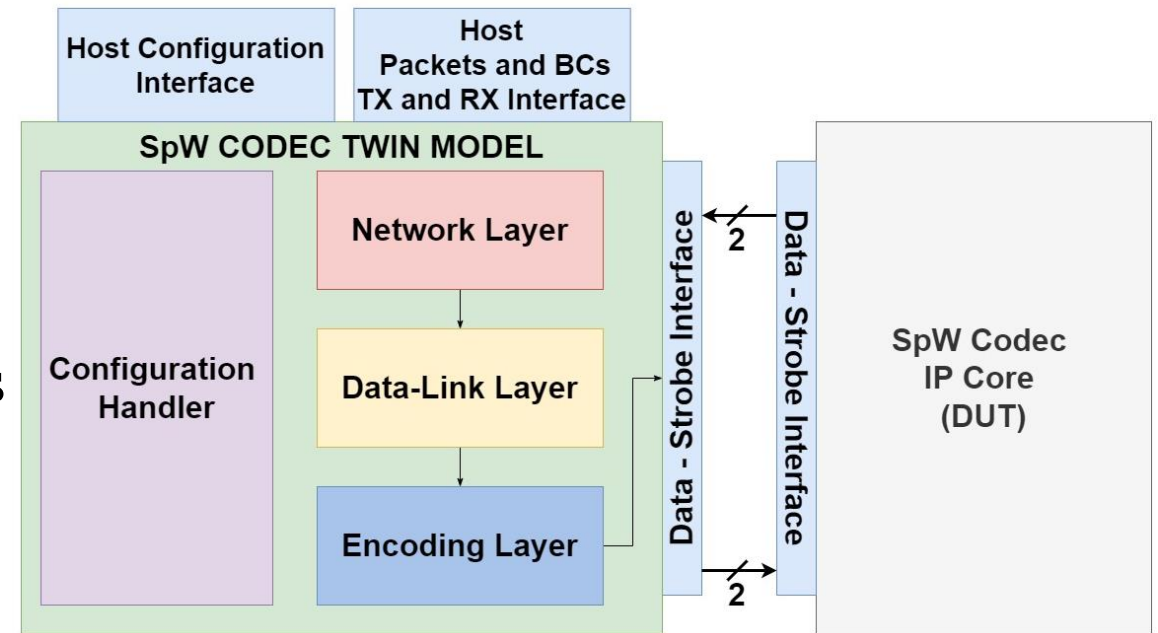
- Direct Communication with DUT
 - Automatic link initialization
 - Automatic TX and RX credit handling



- Direct Communication with DUT
 - Automatic link initialization
 - Automatic TX and RX credit handling
 - Automatic TX of NULLs when needed

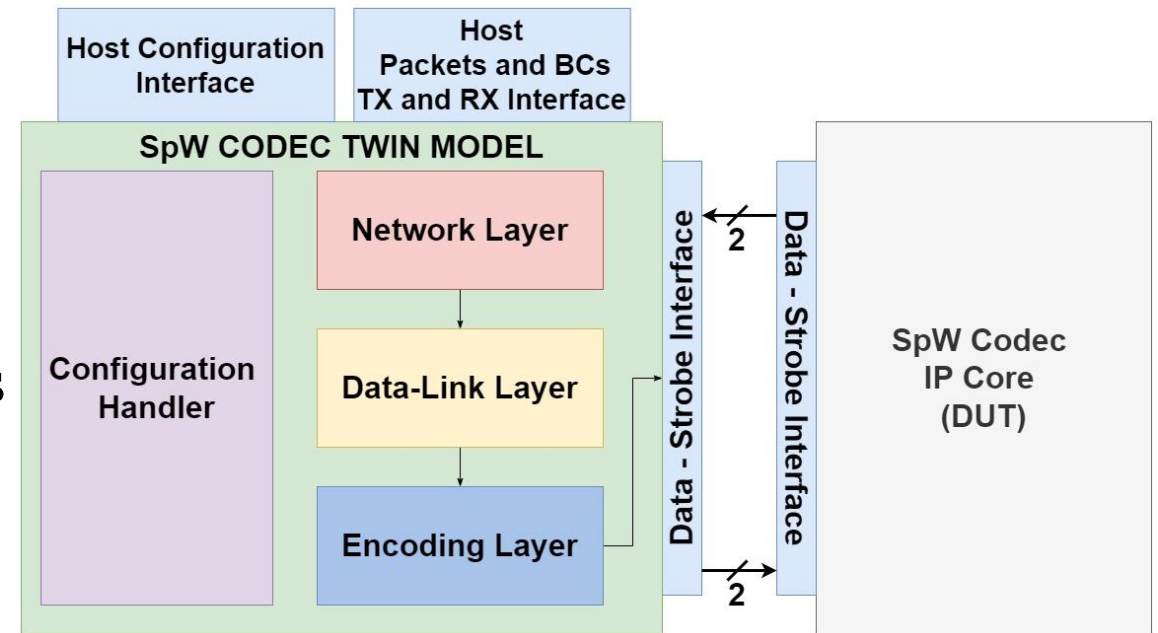


- Direct Communication with DUT
 - Automatic link initialization
 - Automatic TX and RX credit handling
 - Automatic TX of NULLs when needed
 - Automatic recognition of all types of errors



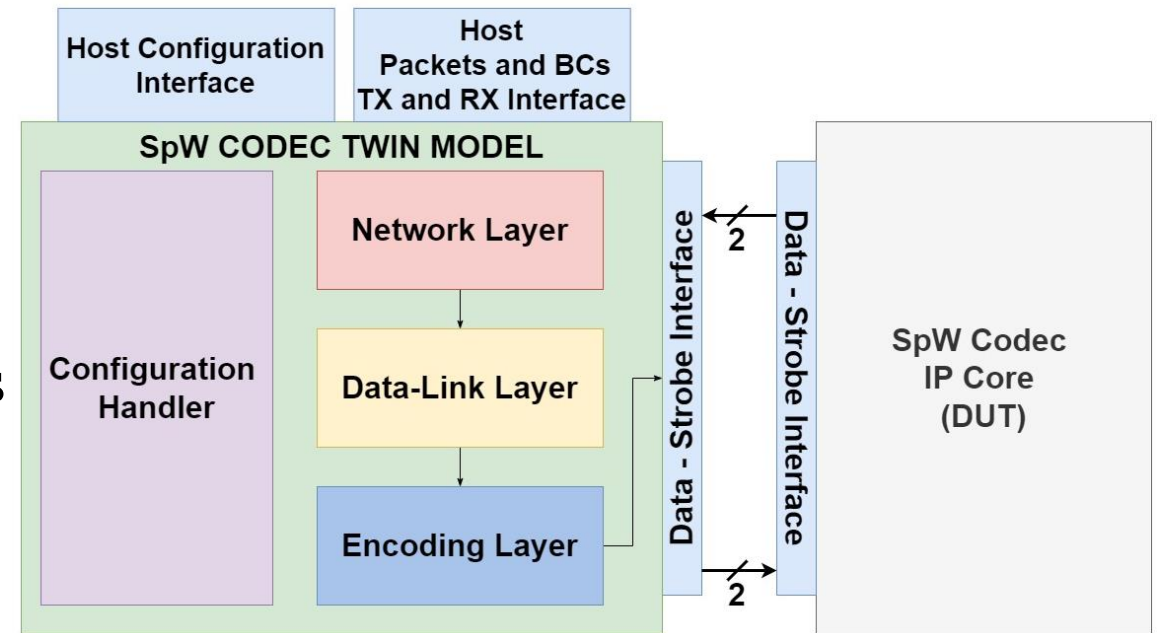
UVM-BASED SPACEWIRE CODEC TWIN MODEL

- Direct Communication with DUT
 - Automatic link initialization
 - Automatic TX and RX credit handling
 - Automatic TX of NULLs when needed
 - Automatic recognition of all types of errors
- Higher level of abstraction
 - Only packets and BCs to be defined



UVM-BASED SPACEWIRE CODEC TWIN MODEL

- Direct Communication with DUT
 - Automatic link initialization
 - Automatic TX and RX credit handling
 - Automatic TX of NULLs when needed
 - Automatic recognition of all types of errors
- Higher level of abstraction
 - Only packets and BCs to be defined

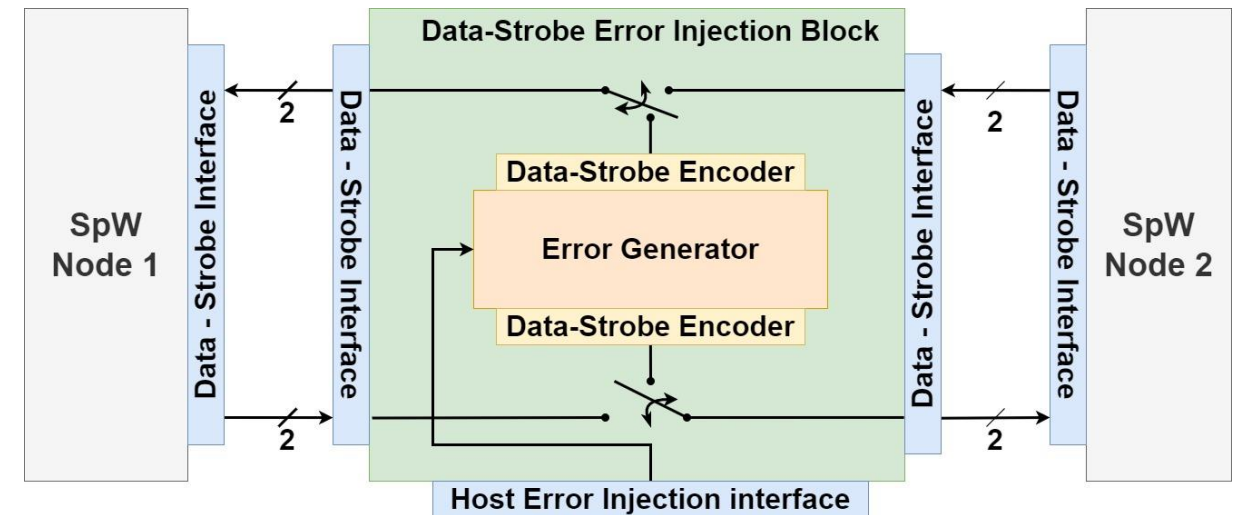


Significant Simplification of the Verification Process
Significant Reduction in Verification Time and Cost

- SpW Standard defines:
 - Parity Error
 - Disconnect Error
 - ESC Error
 - Credit Error
 - GotFCT/GotNChar/GotBC Errors

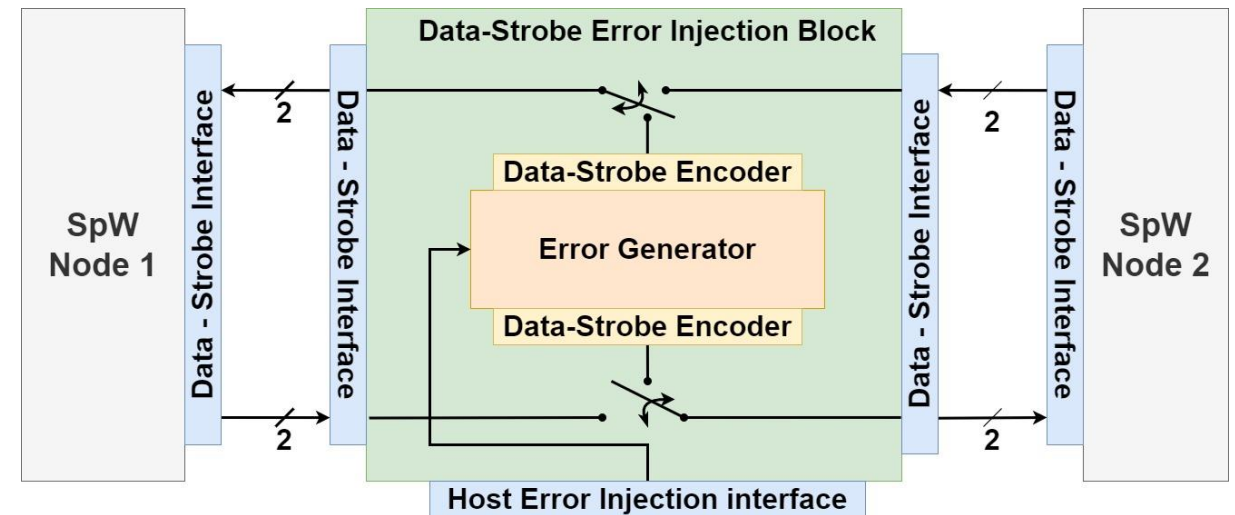
UVM-BASED SPACEWIRE DATA-STROBE ERROR INJECTION MODEL

- SpW Standard defines:
 - Parity Error
 - Disconnect Error
 - ESC Error
 - Credit Error
 - GotFCT/GotNChar/GotBC Errors
- Data-Strobe Error Injection Model
 - Two Data-Strobe Interface
 - Host Interface



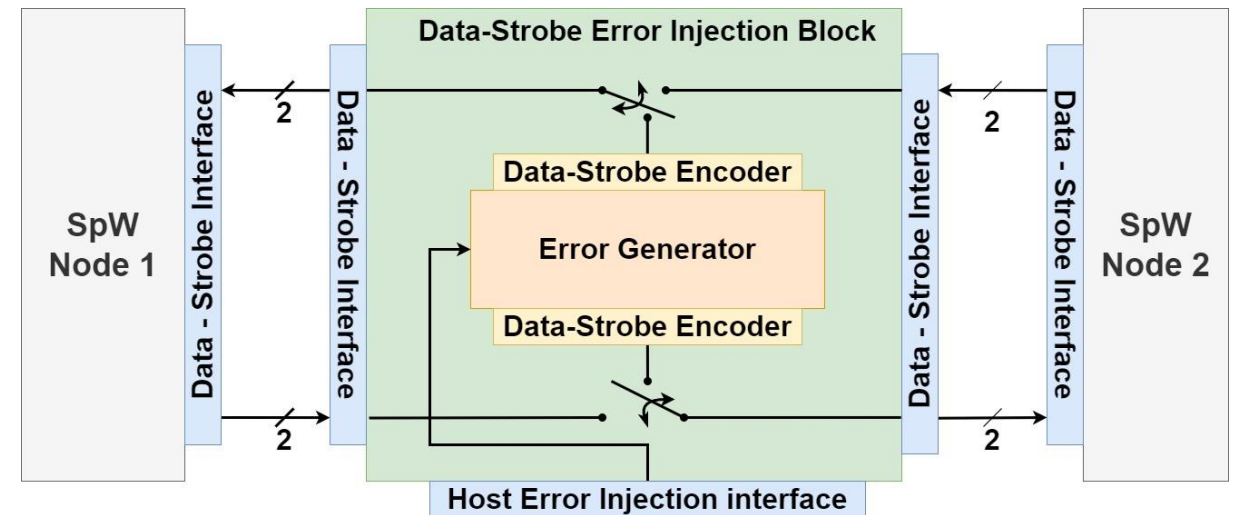
UVM-BASED SPACEWIRE DATA-STROBE ERROR INJECTION MODEL

- SpW Standard defines:
 - Parity Error
 - Disconnect Error
 - ESC Error
 - Credit Error
 - GotFCT/GotNChar/GotBC Errors
- Data-Strobe Error Injection Model
 - Two Data-Strobe Interface
 - Host Interface
- Bi-directional Error Injection



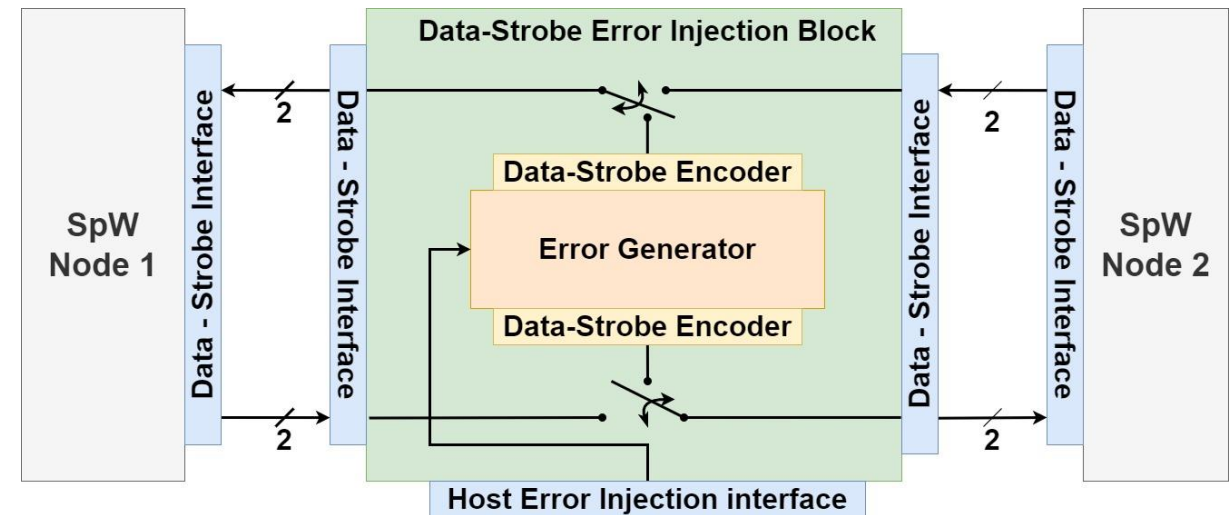
UVM-BASED SPACEWIRE DATA-STROBE ERROR INJECTION MODEL

- SpW Standard defines:
 - Parity Error
 - Disconnect Error
 - ESC Error
 - Credit Error
 - GotFCT/GotNChar/GotBC Errors
- Data-Strobe Error Injection Model
 - Two Data-Strobe Interface
 - Host Interface
- Bi-directional Error Injection
- All types of Errors Supported

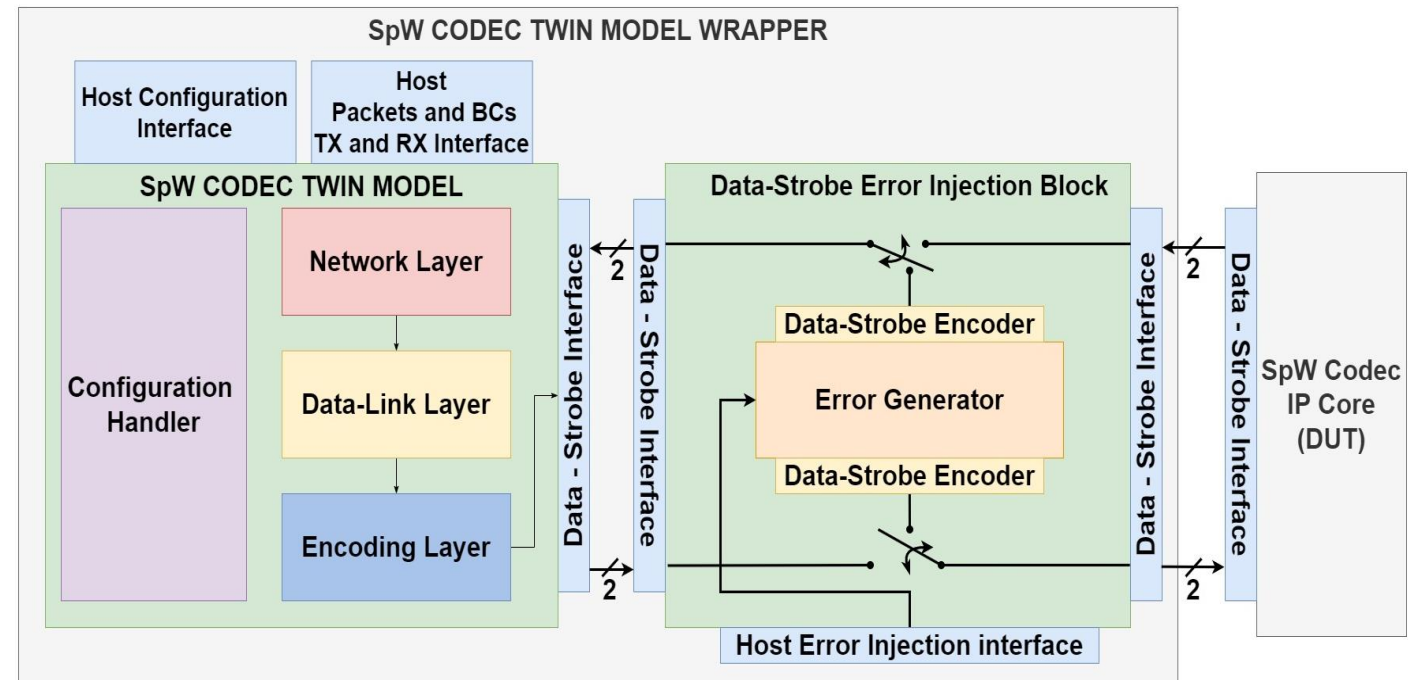


UVM-BASED SPACEWIRE DATA-STROBE ERROR INJECTION MODEL

- SpW Standard defines:
 - Parity Error
 - Disconnect Error
 - ESC Error
 - Credit Error
 - GotFCT/GotNChar/GotBC Errors
- Data-Strobe Error Injection Model
 - Two Data-Strobe Interface
 - Host Interface
- Bi-directional Error Injection
- All types of Errors Supported
- Compliant with UVM
 - Reusable in any project with SpW Interfaces

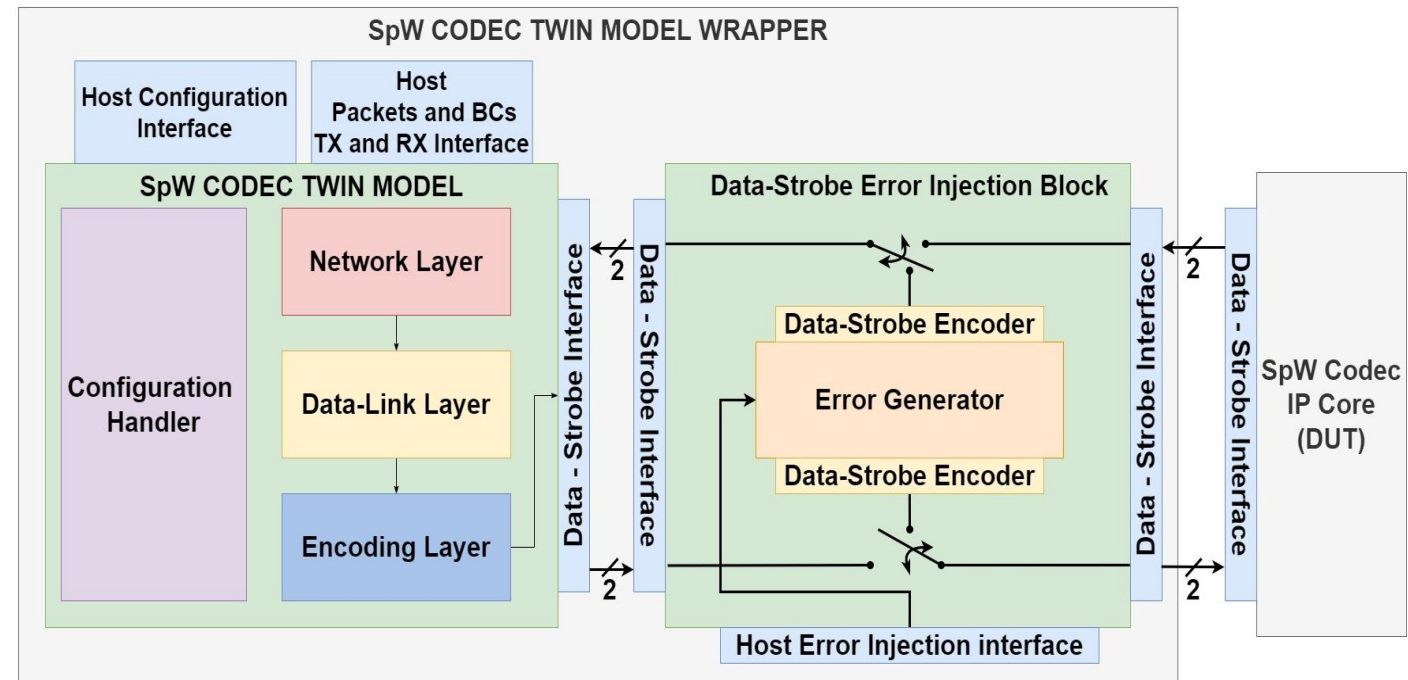


- Integration within SpW Codec Twin Model



UVM-BASED SPACEWIRE DATA-STROBE ERROR INJECTION MODEL

- Integration within SpW Codec Twin Model
- Leveraging UVM power
 - Extension operation
 - Inter-block communication

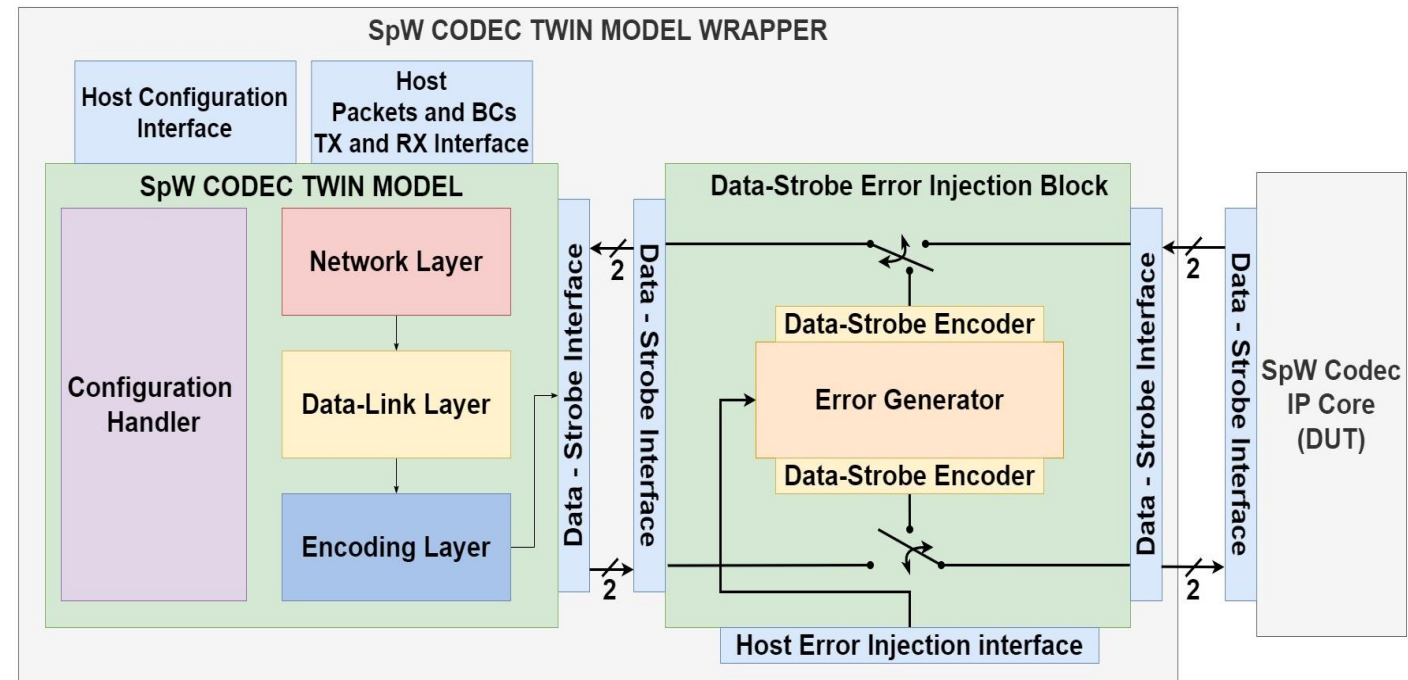


UVM-BASED SPACEWIRE DATA-STROBE ERROR INJECTION MODEL

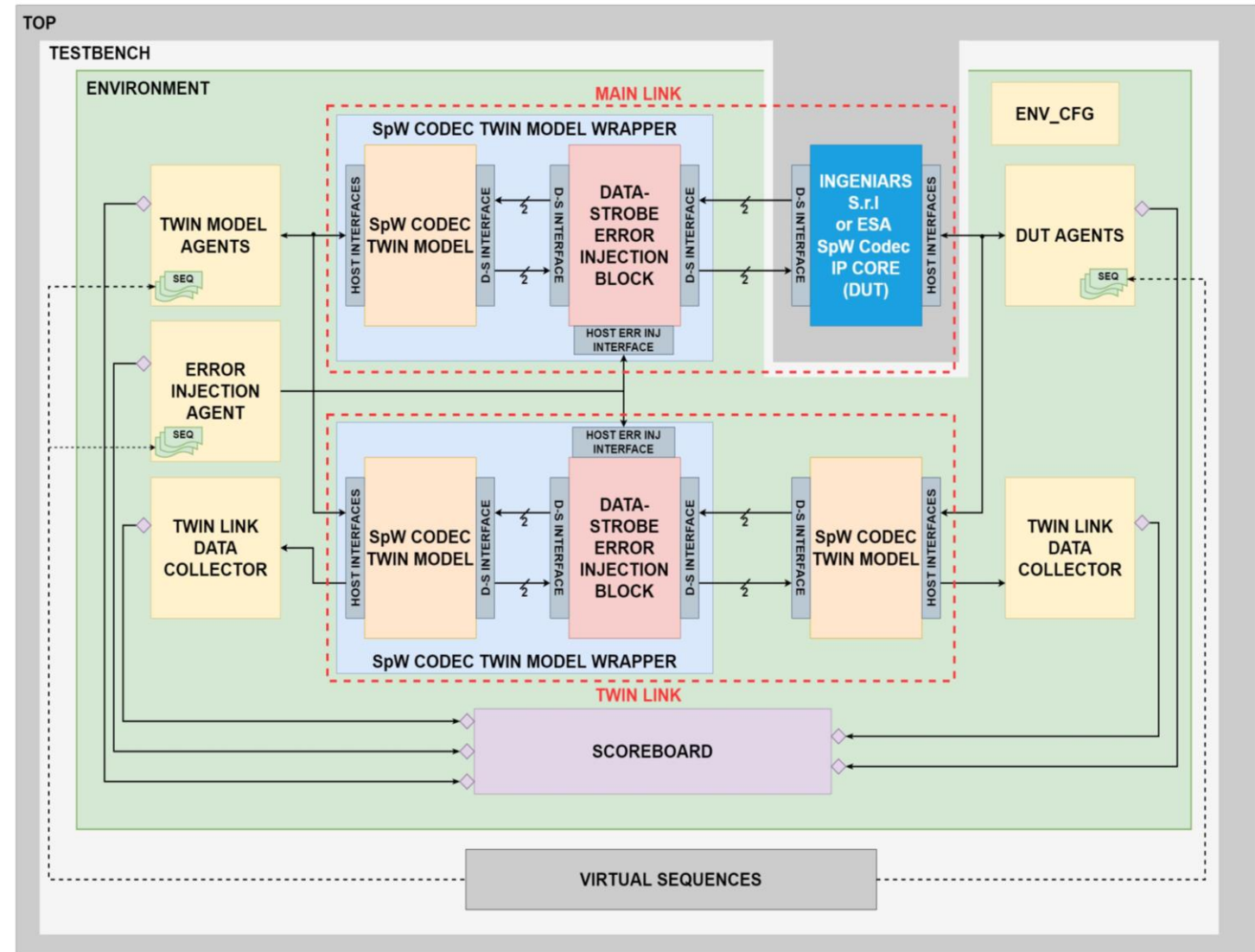
- Integration within SpW Codec Twin Model
- Leveraging UVM power
 - Extension operation
 - Inter-block communication

Extended Twin Model:

- Features of Twin Model
- Possibility to Inject Errors
- Direct communication with DUT

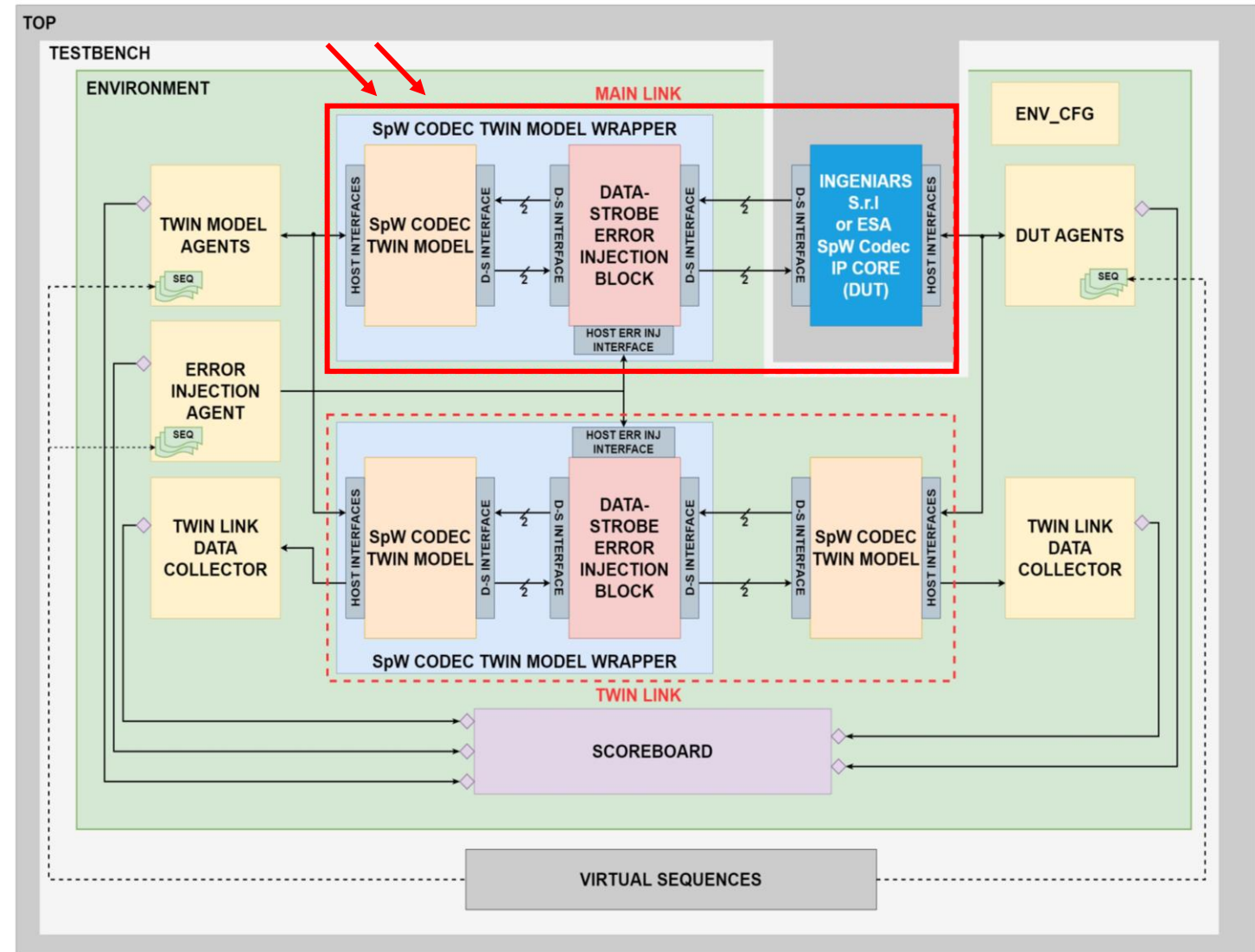


- Innovative architecture



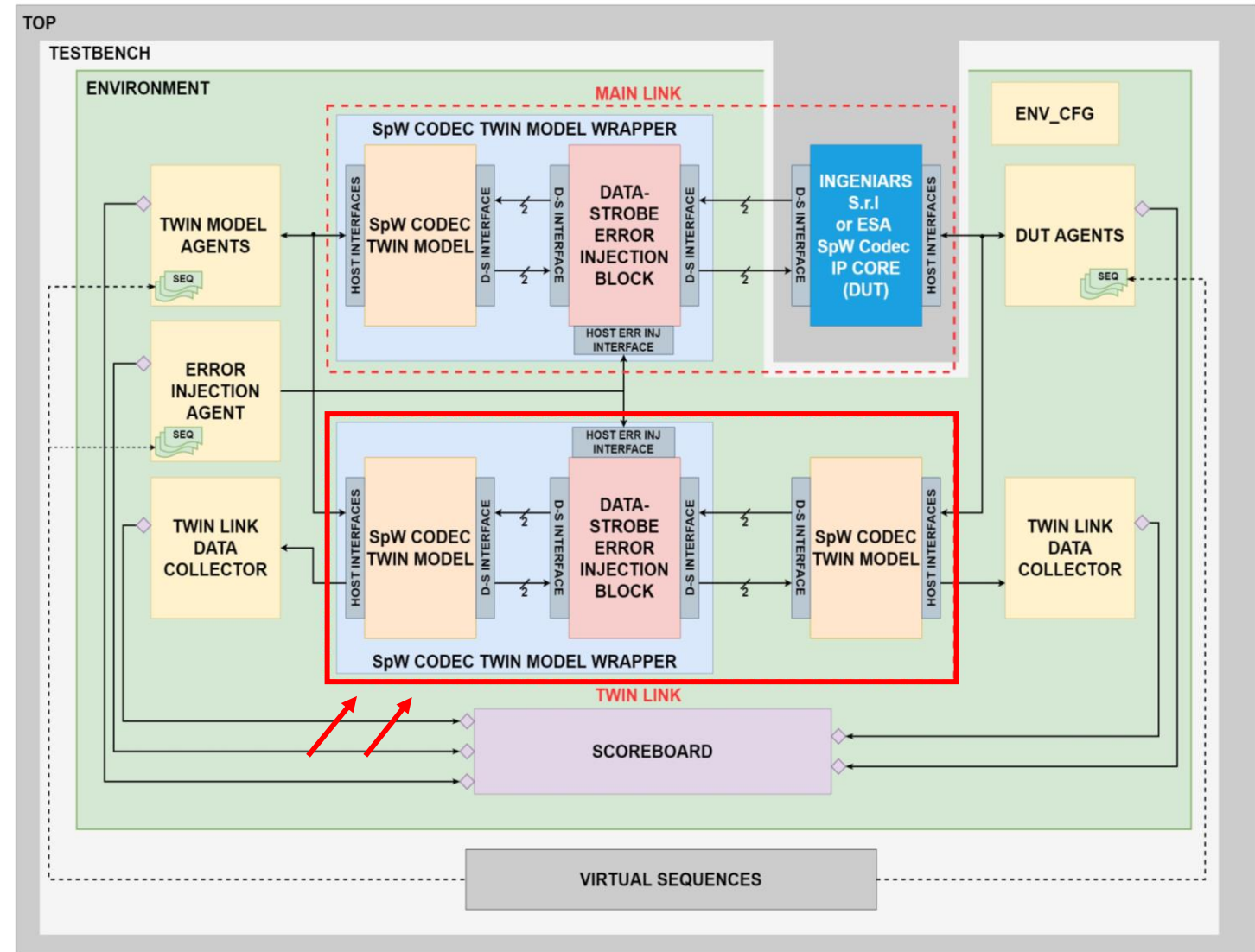
UVM-BASED VERIFICATION ENVIRONMENT ARCHITECTURE

- **Innovative architecture**
- **Main Link**
 - Communication between DUT and Twin Model



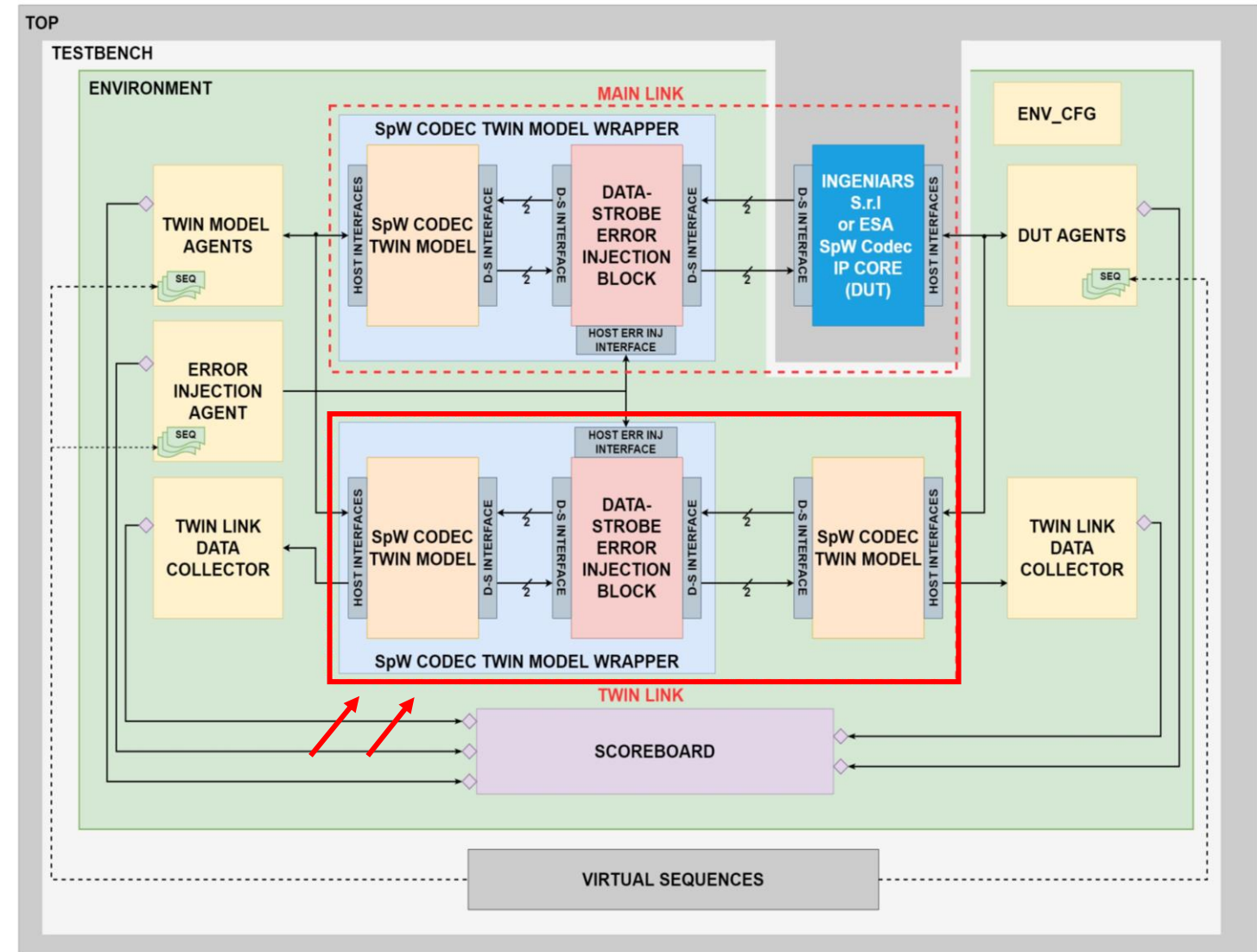
UVM-BASED VERIFICATION ENVIRONMENT ARCHITECTURE

- **Innovative architecture**
- **Main Link**
 - Communication between DUT and Twin Model
- **Twin Link**
 - Communication between two Twin Models



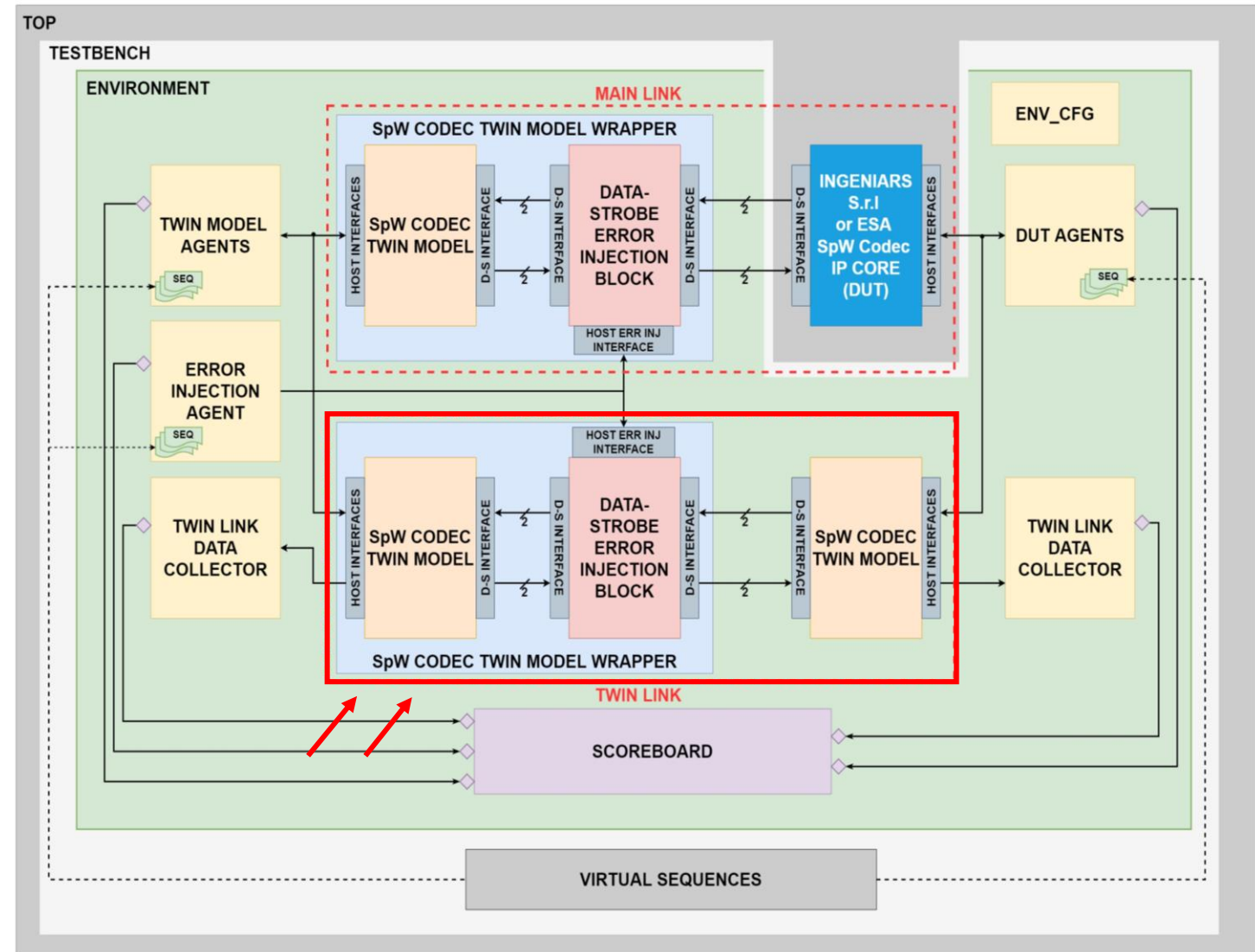
UVM-BASED VERIFICATION ENVIRONMENT ARCHITECTURE

- **Innovative architecture**
- **Main Link**
 - Communication between DUT and Twin Model
- **Twin Link**
 - Communication between two Twin Models
 - Same stimuli of the Main Link



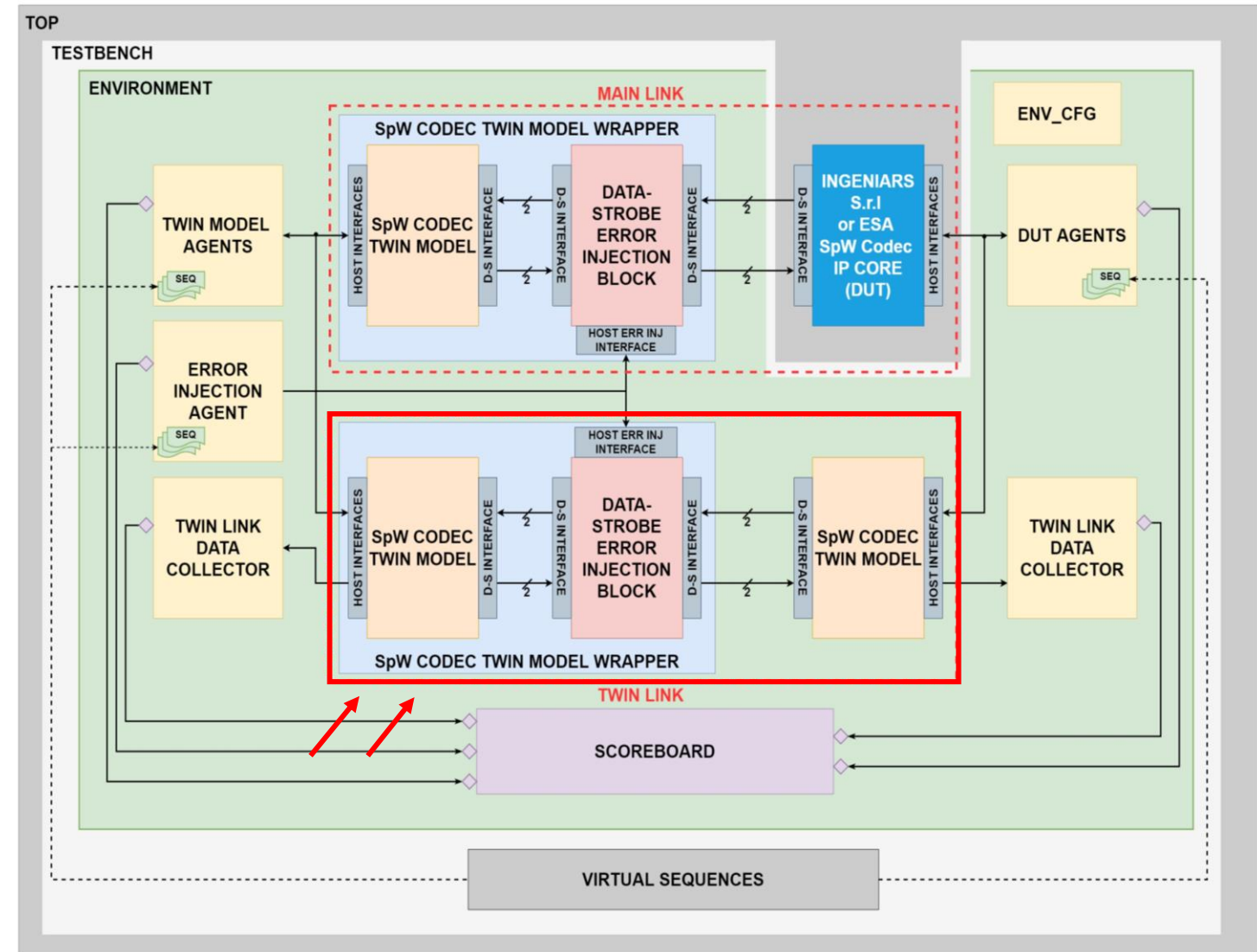
UVM-BASED VERIFICATION ENVIRONMENT ARCHITECTURE

- **Innovative architecture**
- **Main Link**
 - Communication between DUT and Twin Model
- **Twin Link**
 - Communication between two Twin Models
 - Same stimuli of the Main Link
 - Emulates the ideal behavior of the SpW Link



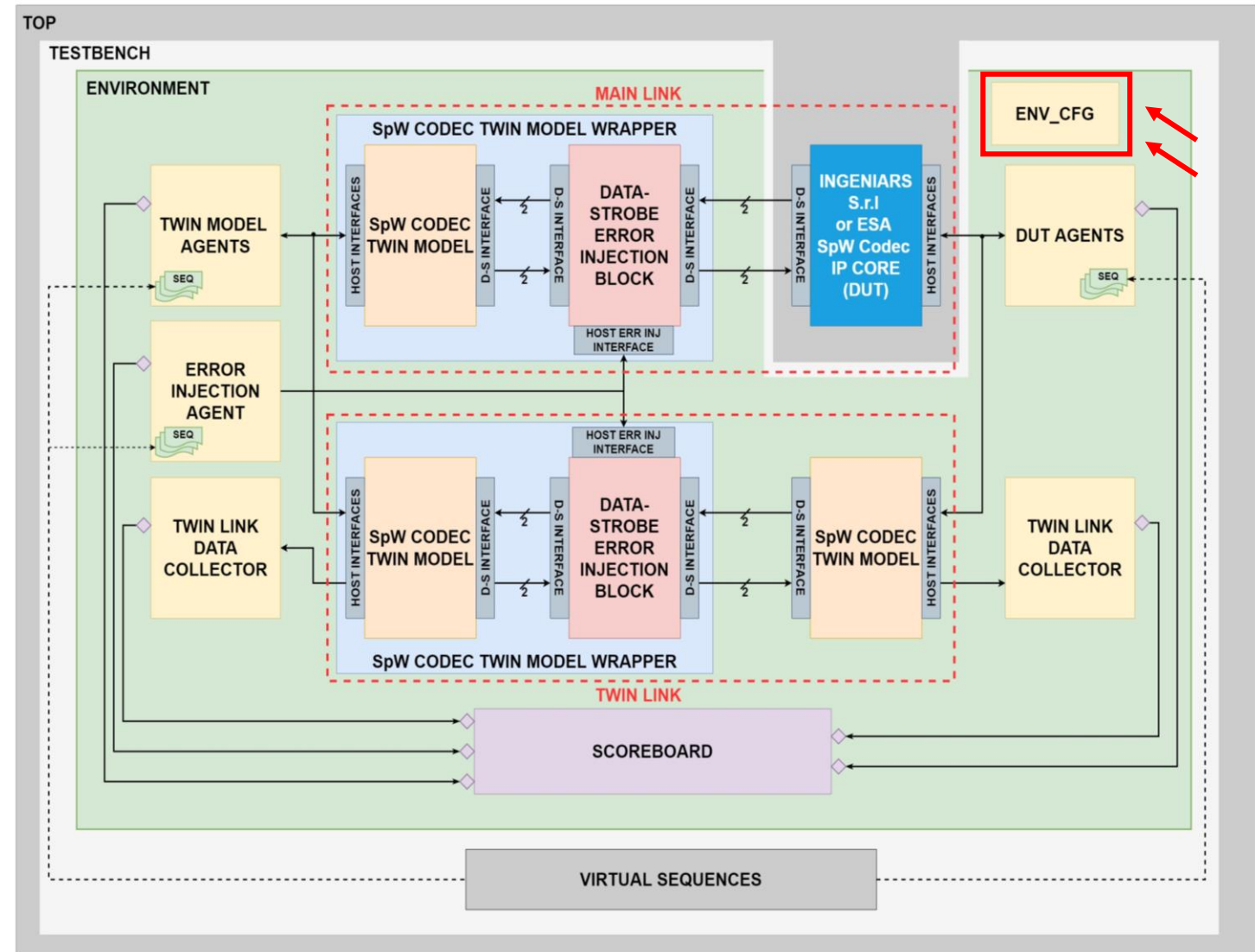
UVM-BASED VERIFICATION ENVIRONMENT ARCHITECTURE

- **Innovative architecture**
- **Main Link**
 - Communication between DUT and Twin Model
- **Twin Link**
 - Communication between two Twin Models
 - Same stimuli of the Main Link
 - Emulates the ideal behavior of the SpW Link
 - Possibility to be disabled



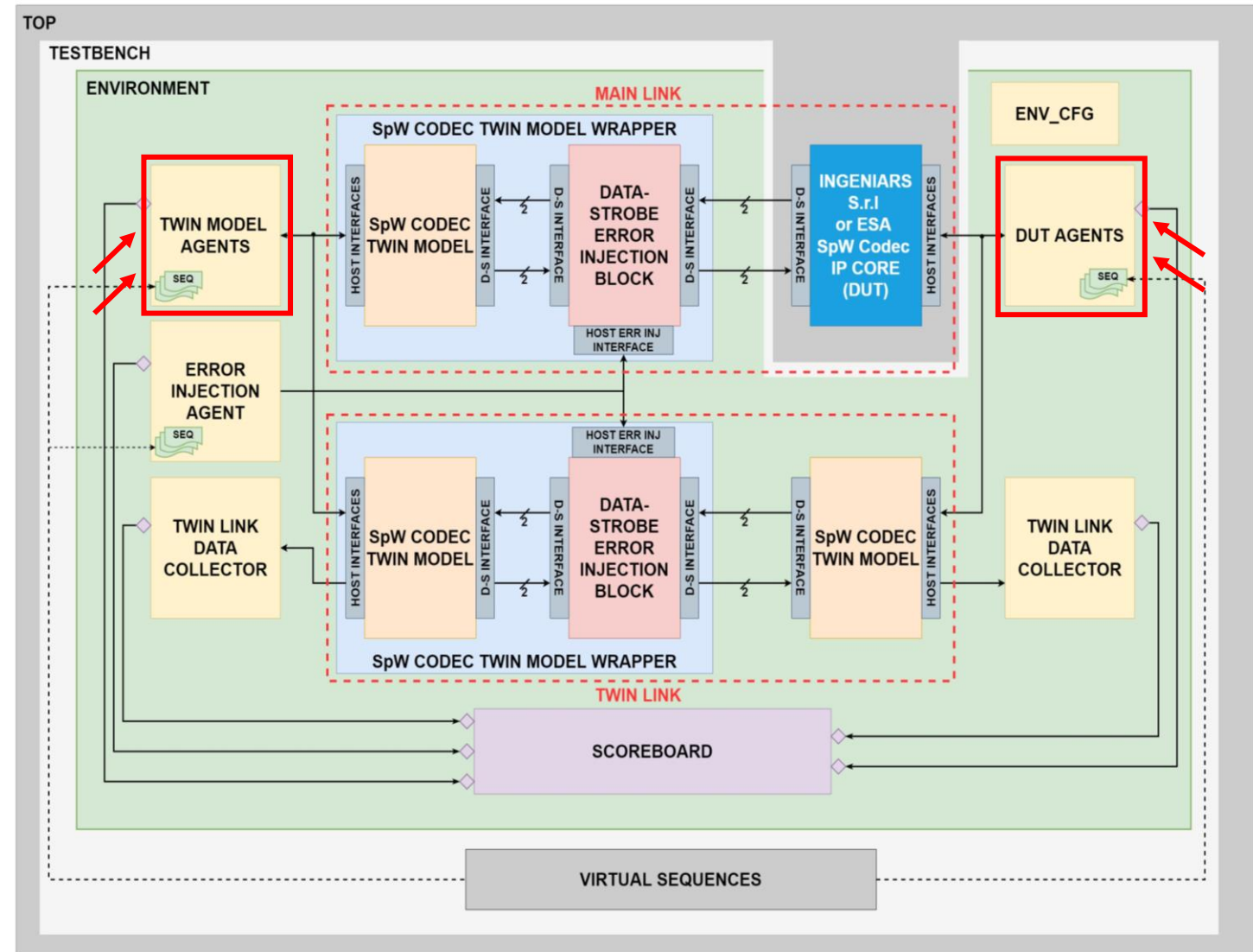
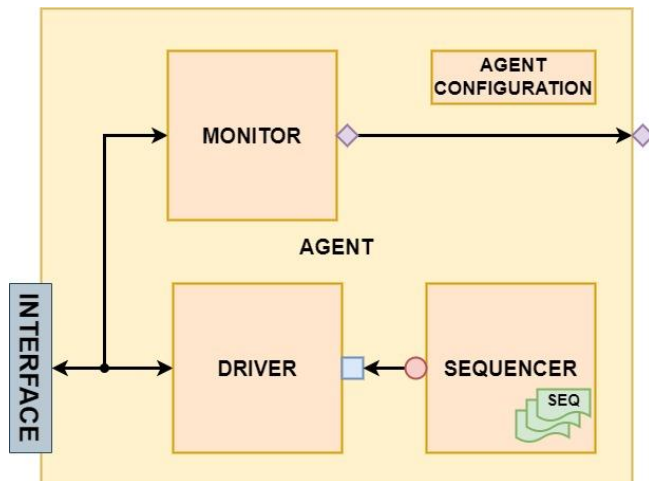
UVM-BASED VERIFICATION ENVIRONMENT ARCHITECTURE

- Environment Configuration



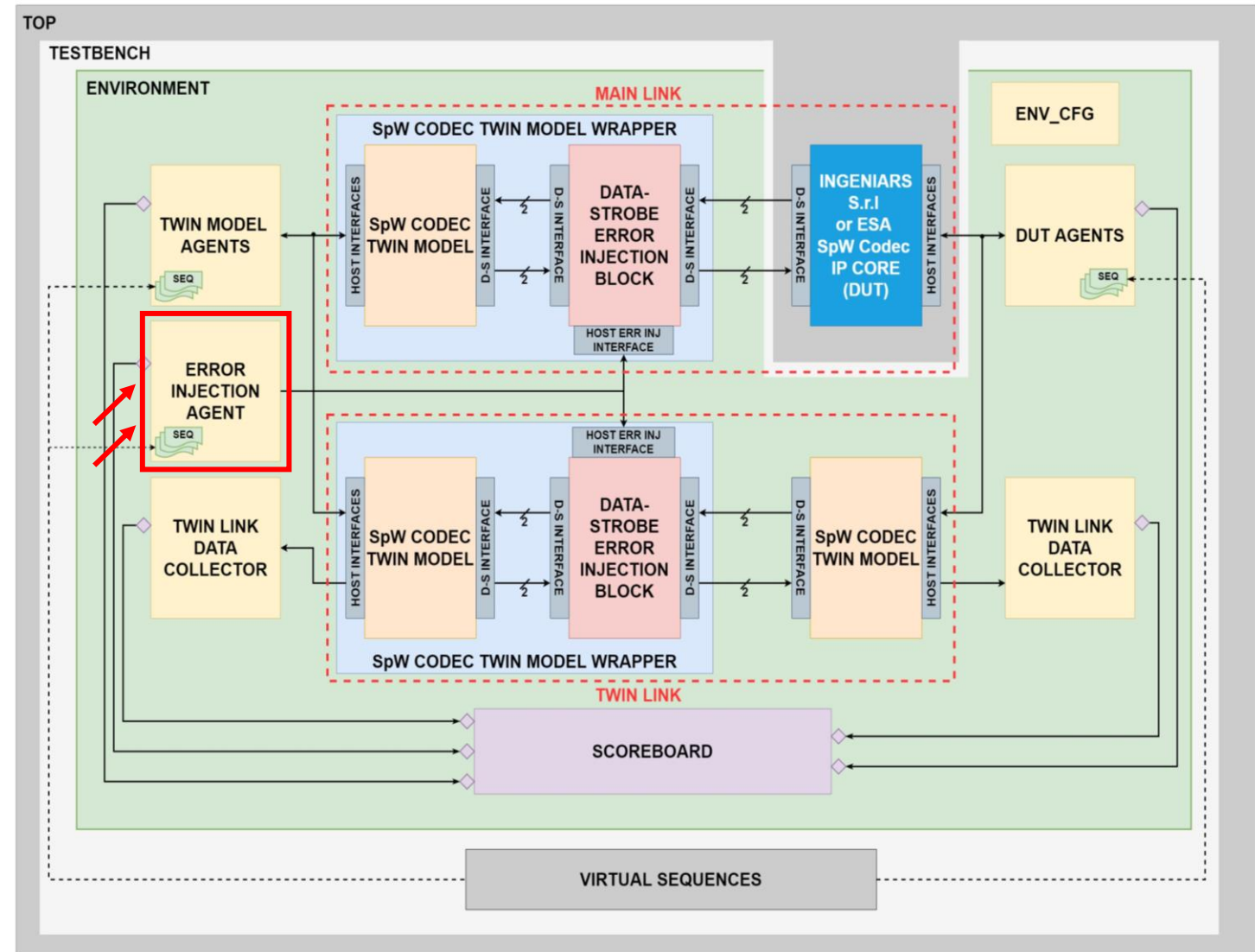
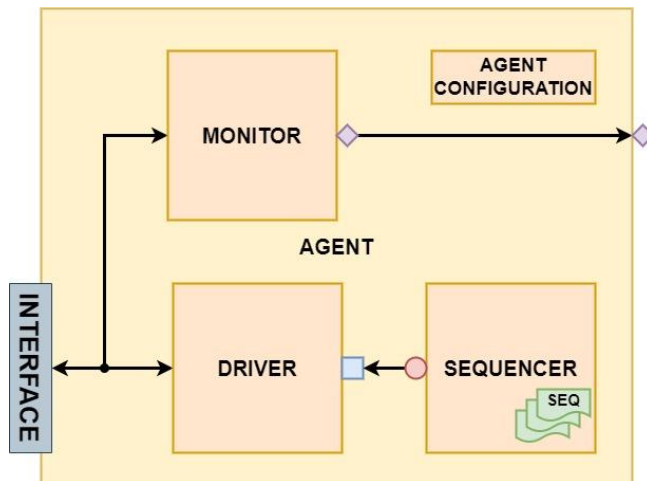
UVM-BASED VERIFICATION ENVIRONMENT ARCHITECTURE

- Environment Configuration
- DUT Agents
- Twin Model Agents



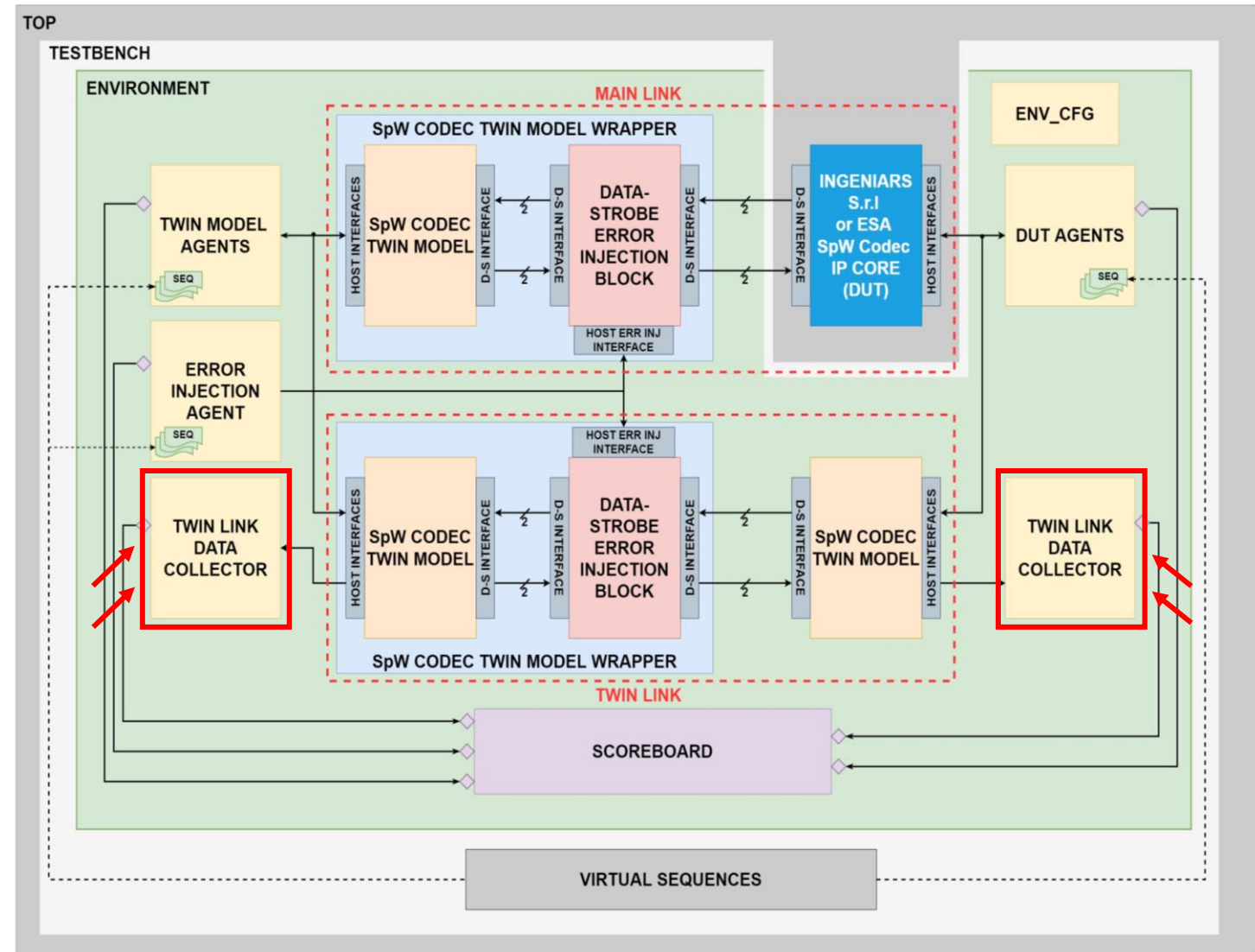
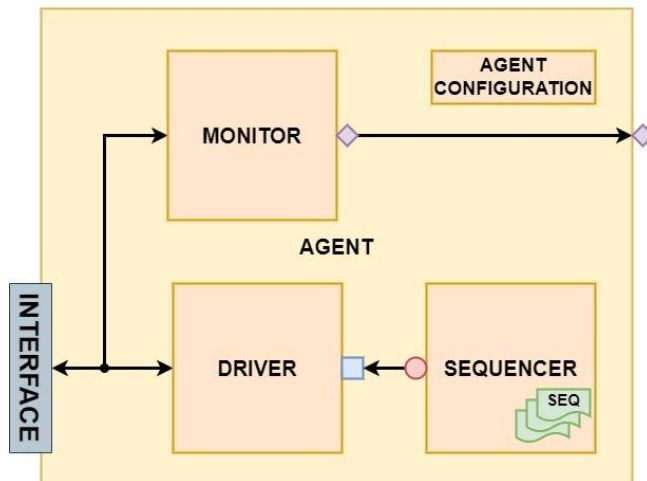
UVM-BASED VERIFICATION ENVIRONMENT ARCHITECTURE

- Environment Configuration
- DUT Agents
- Twin Model Agents
- Error Injection Agent



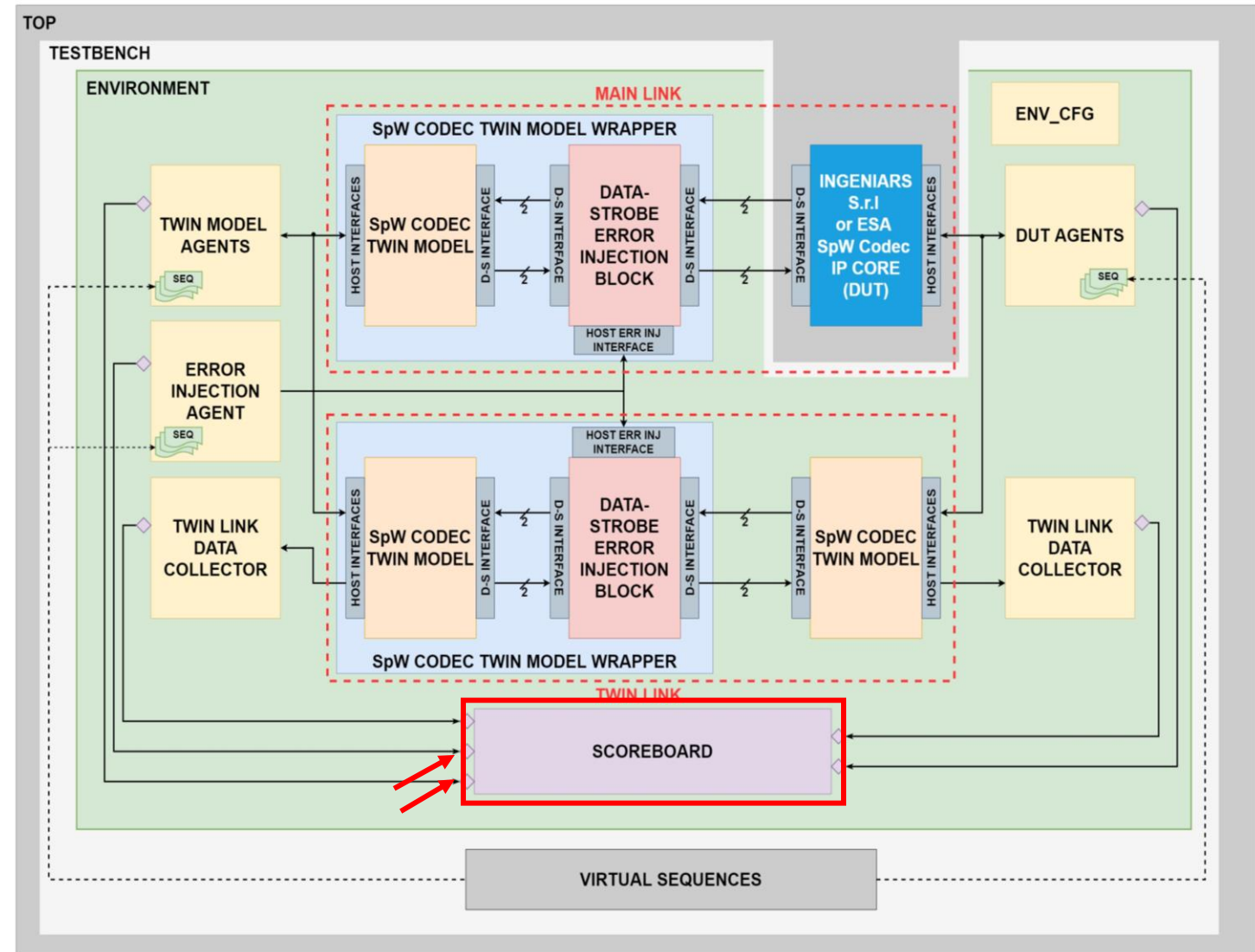
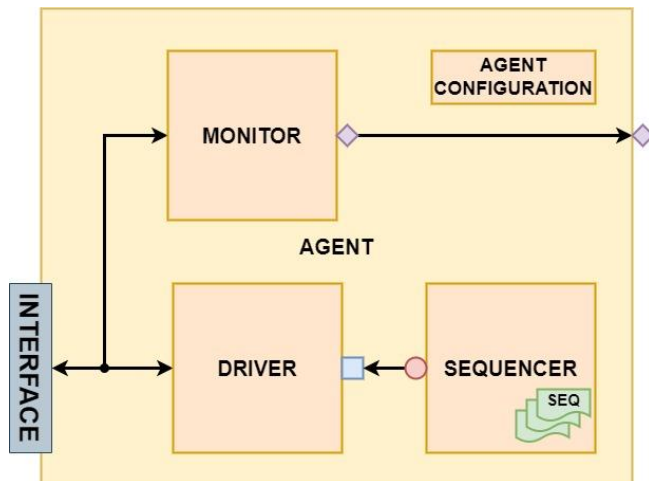
UVM-BASED VERIFICATION ENVIRONMENT ARCHITECTURE

- Environment Configuration
- DUT Agents
- Twin Model Agents
- Error Injection Agent
- Twin Link Data Collectors



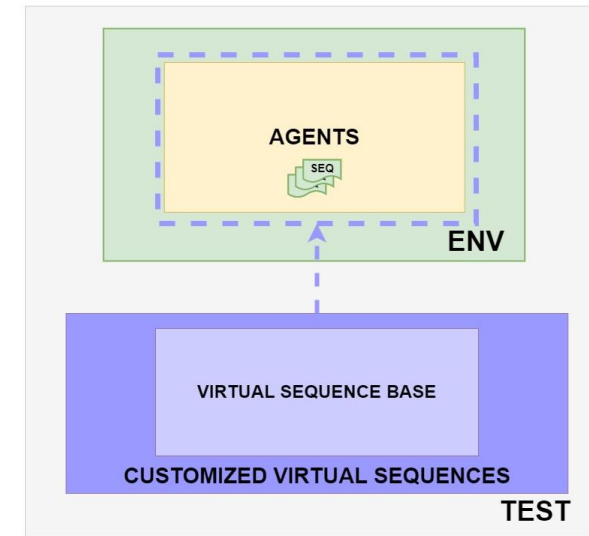
UVM-BASED VERIFICATION ENVIRONMENT ARCHITECTURE

- Environment Configuration
- DUT Agents
- Twin Model Agents
- Error Injection Agent
- Twin Link Data Collectors
- Scoreboard

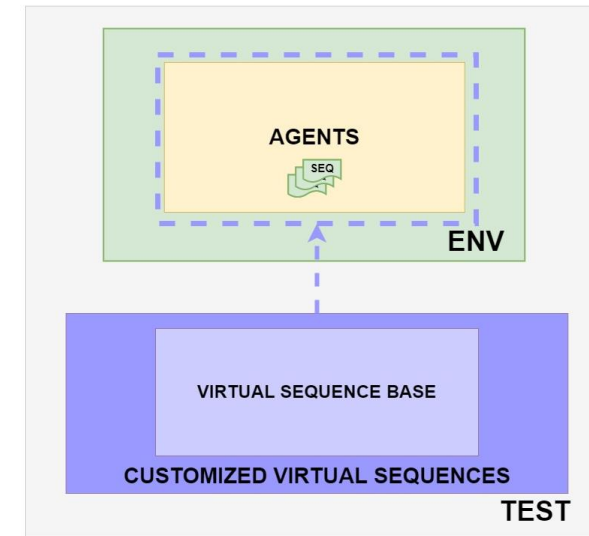


- Simulation scenario configuration

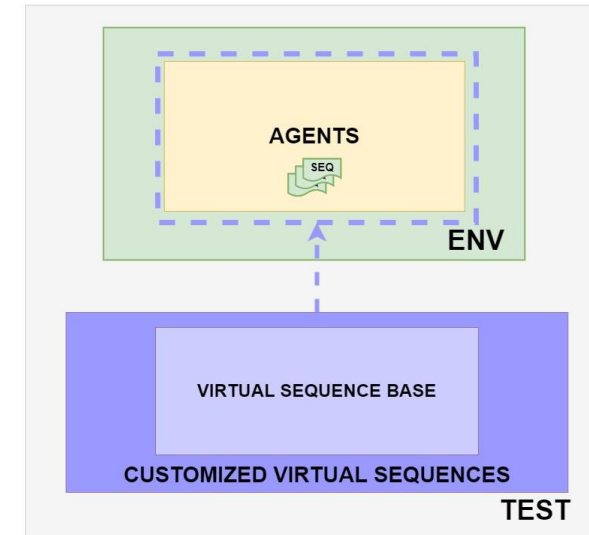
- Simulation scenario configuration
 - UVM Virtual sequences method



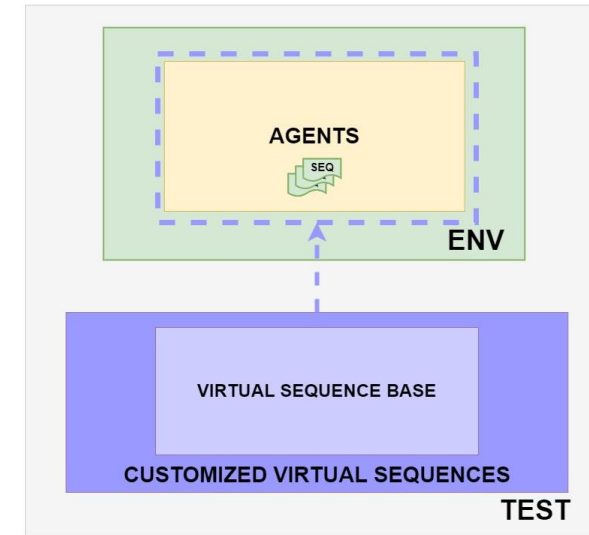
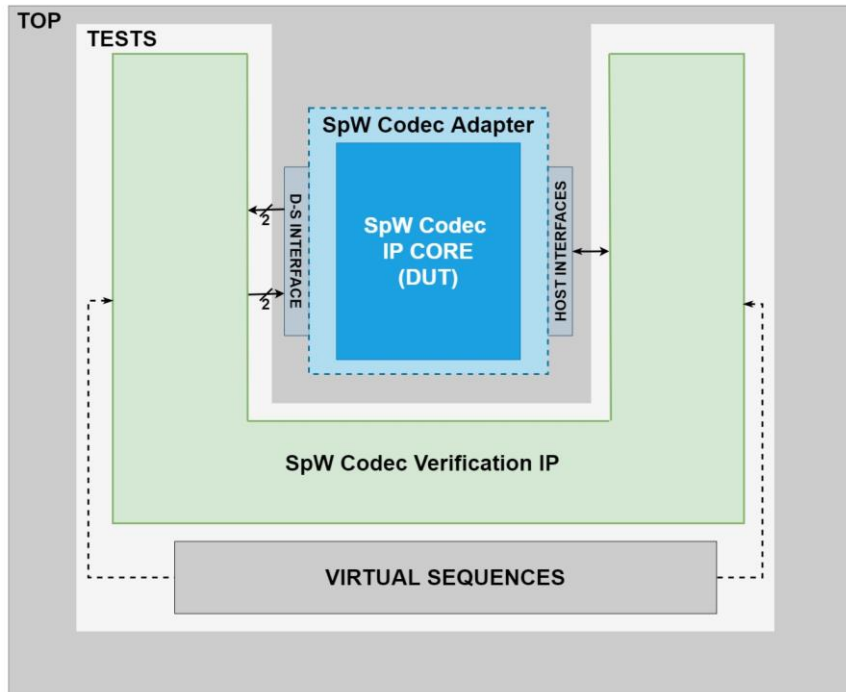
- Simulation scenario configuration
 - UVM Virtual sequences method
 - By setting variables



- Simulation scenario configuration
 - UVM Virtual sequences method
 - By setting variables
 - Not knowing the internal VE architecture



- Simulation scenario configuration
 - UVM Virtual sequences method
 - By setting variables
 - Not knowing the internal VE architecture

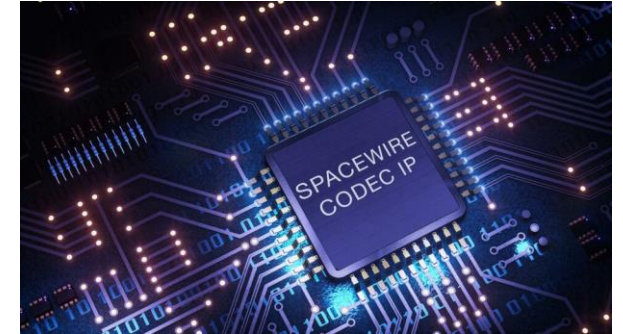


Verification Intellectual Property (VIP)

- Complete Functional Verification of any SpW Codec (a basic adapter could be required)

- Developed all test-cases to achieve 100% functional coverage

- Developed all test-cases to achieve 100% functional coverage
- Performed complete Verification Campaign on:
 - IngeniArs S.r.l. SpW Codec IP Core
 - European Space Agency SpW Codec IP Core



VERIFICATION CAMPAIGN

- Developed all test-cases to achieve 100% functional coverage
- Performed complete Verification Campaign on:
 - IngeniArs S.r.l. SpW Codec IP Core
 - European Space Agency SpW Codec IP Core

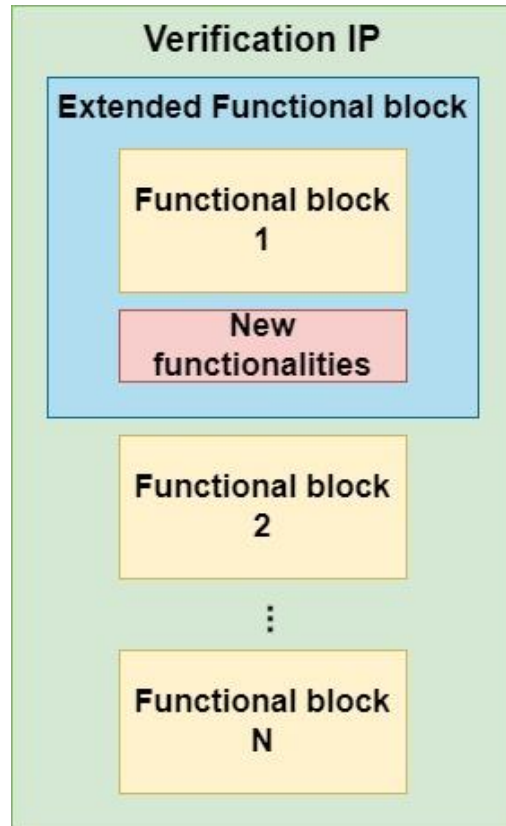


DUT	Compliance with SpW Standard	Compliance with SpW Standard Rev.1
IngeniArs S.r.l. SpW Codec IP Core	✓	✓
European Space Agency SpW Codec IP Core	✓	



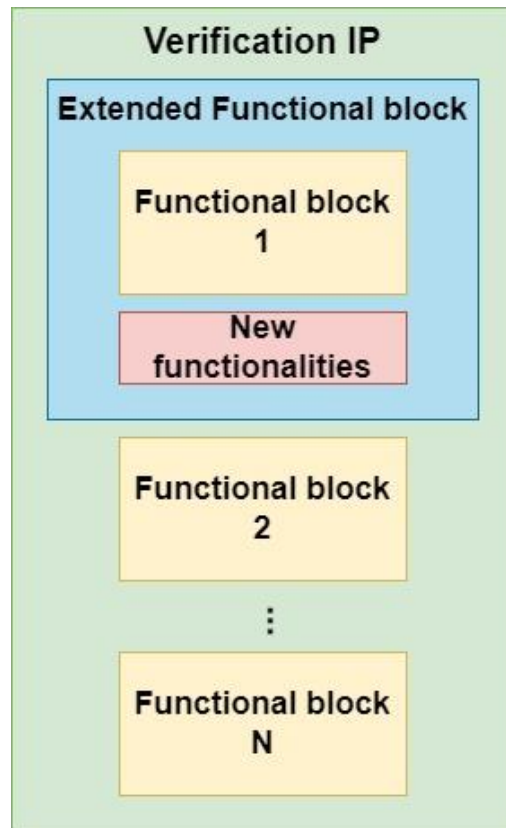
- Maintainability

- Maintainability
 - Add functionalities

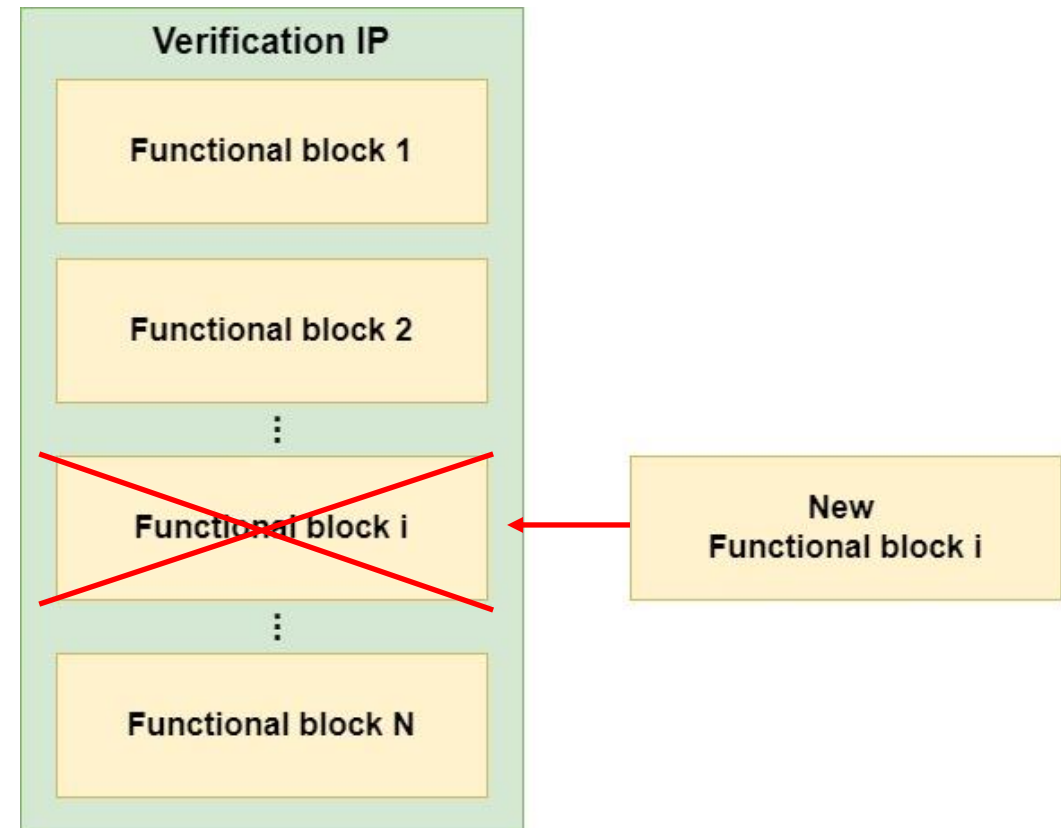


- Maintainability

- Add functionalities

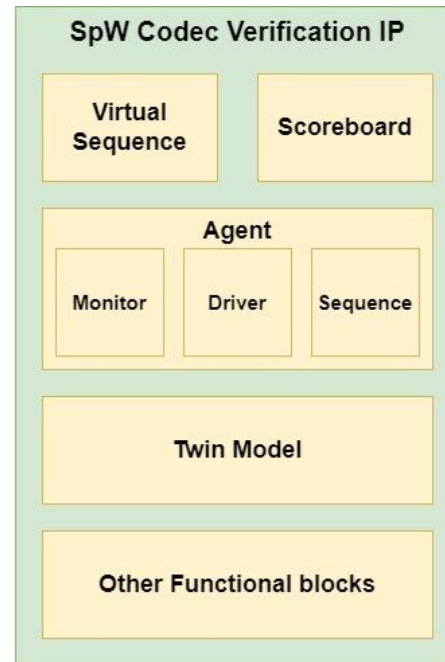
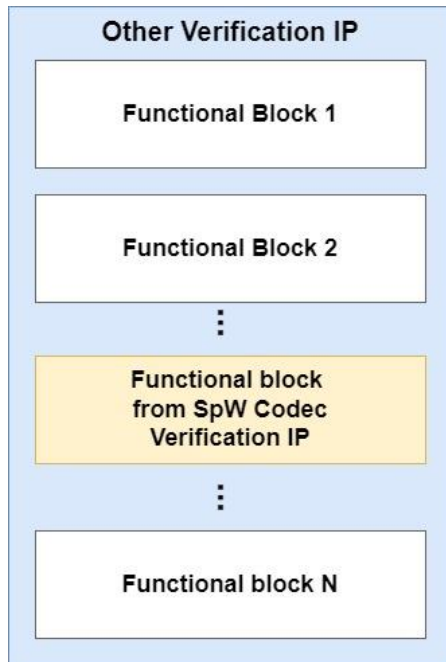


- Update functionalities

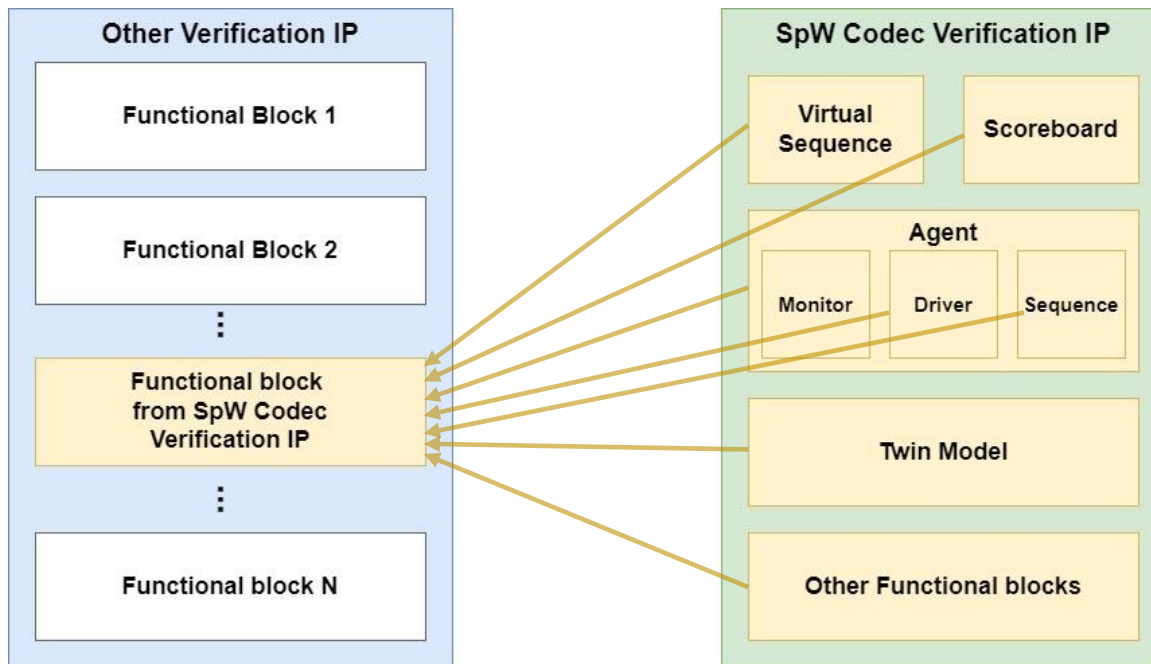


- Reusability

- Reusability
 - From a single functional block



- Reusability
 - From a single functional block

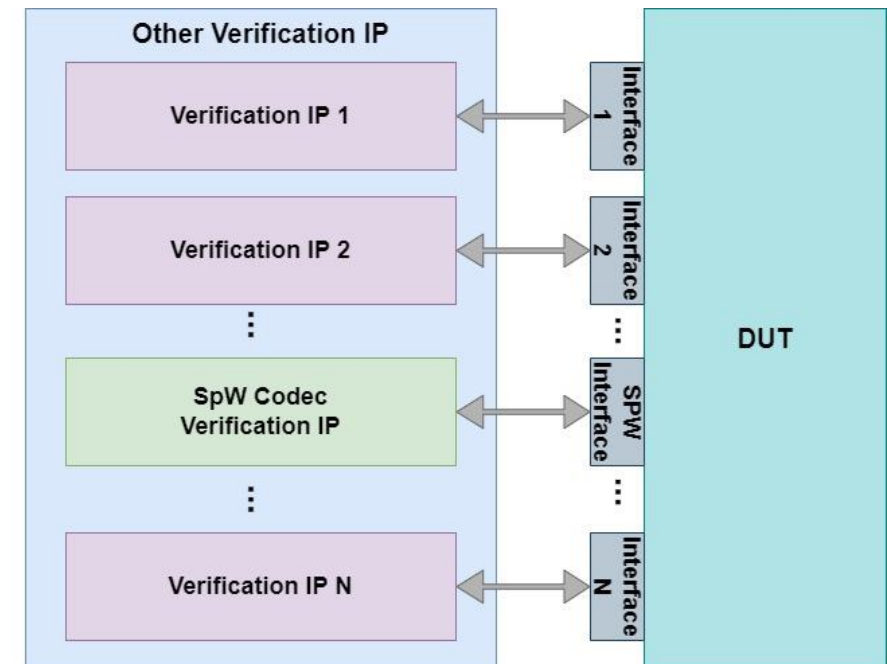
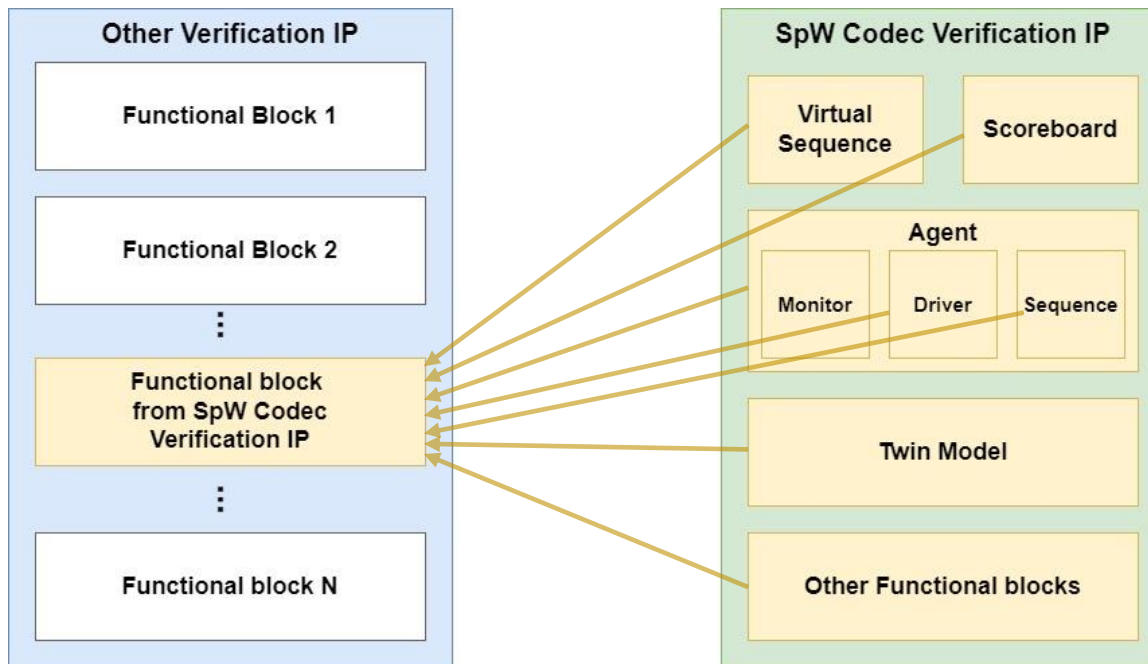


UVM-BASED APPROACH ADVANTAGES

- Reusability

- From a single functional block

- To all the VIP

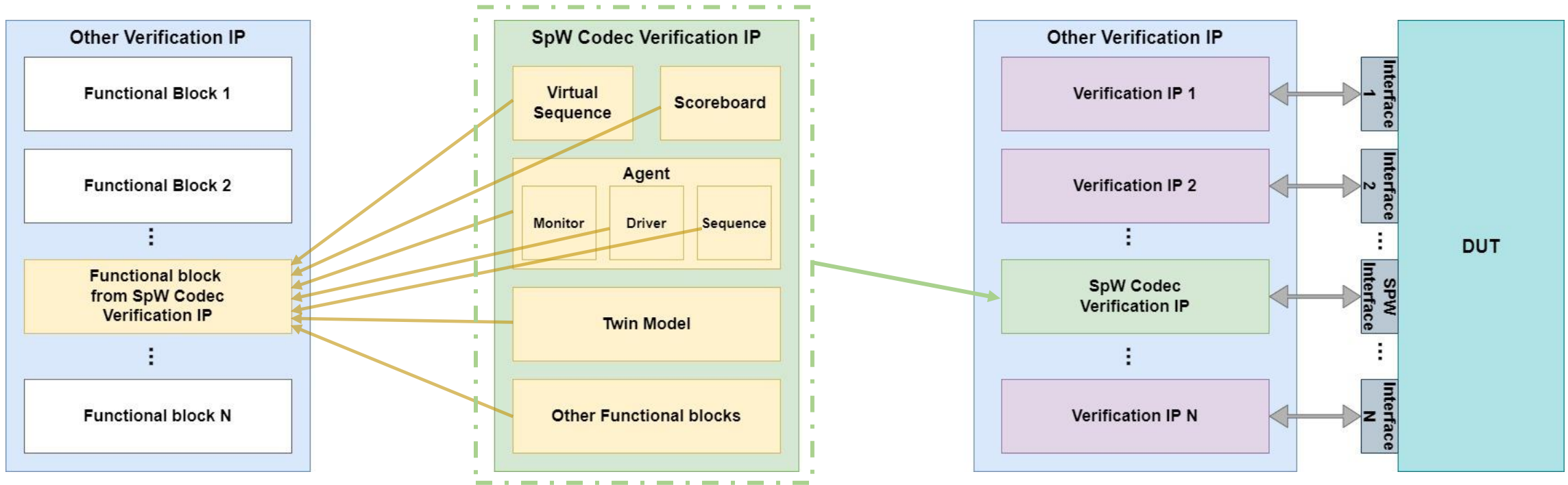


UVM-BASED APPROACH ADVANTAGES

- Reusability

- From a single functional block

- To all the VIP



Verification IP supporting a full test Campaign of any SpW Codec

- Based on a SpW Codec Twin Model

Verification IP supporting a full test Campaign of any SpW Codec

- Based on a SpW Codec Twin Model
- Double Link Architecture

Verification IP supporting a full test Campaign of any SpW Codec

- Based on a SpW Codec Twin Model
- Double Link Architecture
- Complete support of all simulation scenarios involving errors

Verification IP supporting a full test Campaign of any SpW Codec

- Based on a SpW Codec Twin Model
- Double Link Architecture
- Complete support of all simulation scenarios involving errors
- Support all testcases for 100% functional coverage of SpW Standard, Rev.1.

Verification IP supporting a full test Campaign of any SpW Codec

- Based on a SpW Codec Twin Model
- Double Link Architecture
- Complete support of all simulation scenarios involving errors
- Support all testcases for 100% functional coverage of SpW Standard, Rev.1.
- Used for functional verification of:
 - IngeniArs S.r.l. SpW Codec IP Core
 - European Space Agency SpW Codec IP Core

Verification IP supporting a full test Campaign of any SpW Codec

- Based on a SpW Codec Twin Model
- Double Link Architecture
- Complete support of all simulation scenarios involving errors
- Support all testcases for 100% functional coverage of SpW Standard, Rev.1.
- Used for functional verification of:
 - IngeniArs S.r.l. SpW Codec IP Core
 - European Space Agency SpW Codec IP Core
- Fully compliant with UVM with advantages in terms of:
 - Maintainability
 - Reusability in other projects

Verification IP supporting a full test Campaign of any SpW Codec

- Based on a SpW Codec Twin Model
- Double Link Architecture
- Complete support of all simulation scenarios involving errors
- Support all testcases for 100% functional coverage of SpW Standard, Rev.1.
- Used for functional verification of:
 - IngeniArs S.r.l. SpW Codec IP Core
 - European Space Agency SpW Codec IP Core
- Fully compliant with UVM with advantages in terms of:
 - Maintainability
 - Reusability in other projects

Significant reduction of verification time and costs
Significant increase of SpW Codec reliability

Thanks for your attention!

For more information about SpW Codec Verification IP:

Email:

simone.vagaggini@ingeniars.com
marco.trafeli@ingeniars.com
daniele.davalle@ingeniars.com
roberto.ciardi@ingeniars.com

Company Website:

www.ingeniars.com

