



DOULOS

WELCOME

Using the SystemC™ Verification Library

John Aynsley

Agenda

- Introduction to the SystemC Verification Library
- Example Design and Test Bench
- Benefits of SCV
- Issues Using SCV

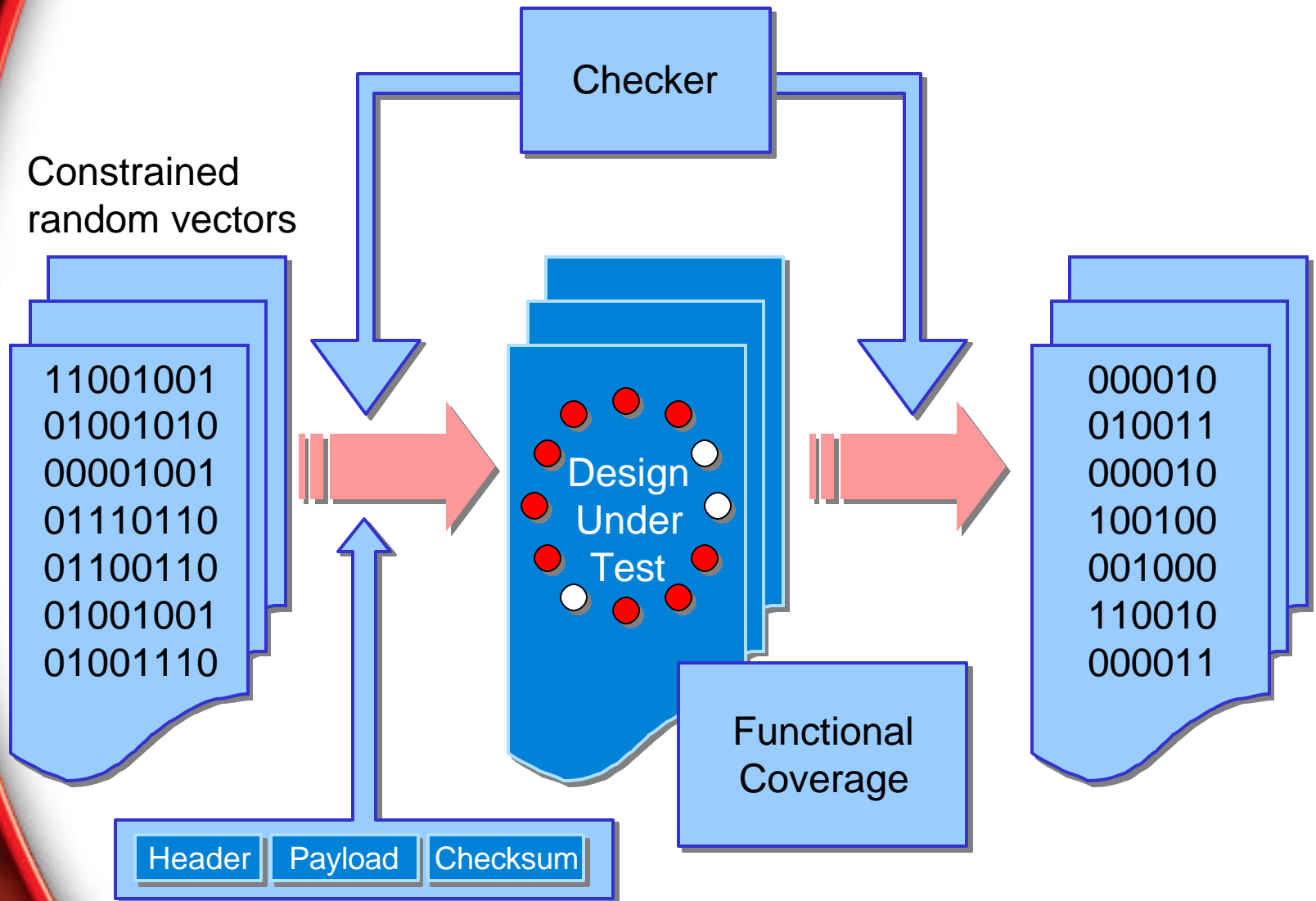


SystemC Verification Library (SCV)

- C++ class library on top of SystemC
- Introduced December 2002 (Beta 1)
- Currently at Beta 4
- Available on Unix, GNU/Linux, and Windows
 - Needs recent compiler versions



Verification Methodology

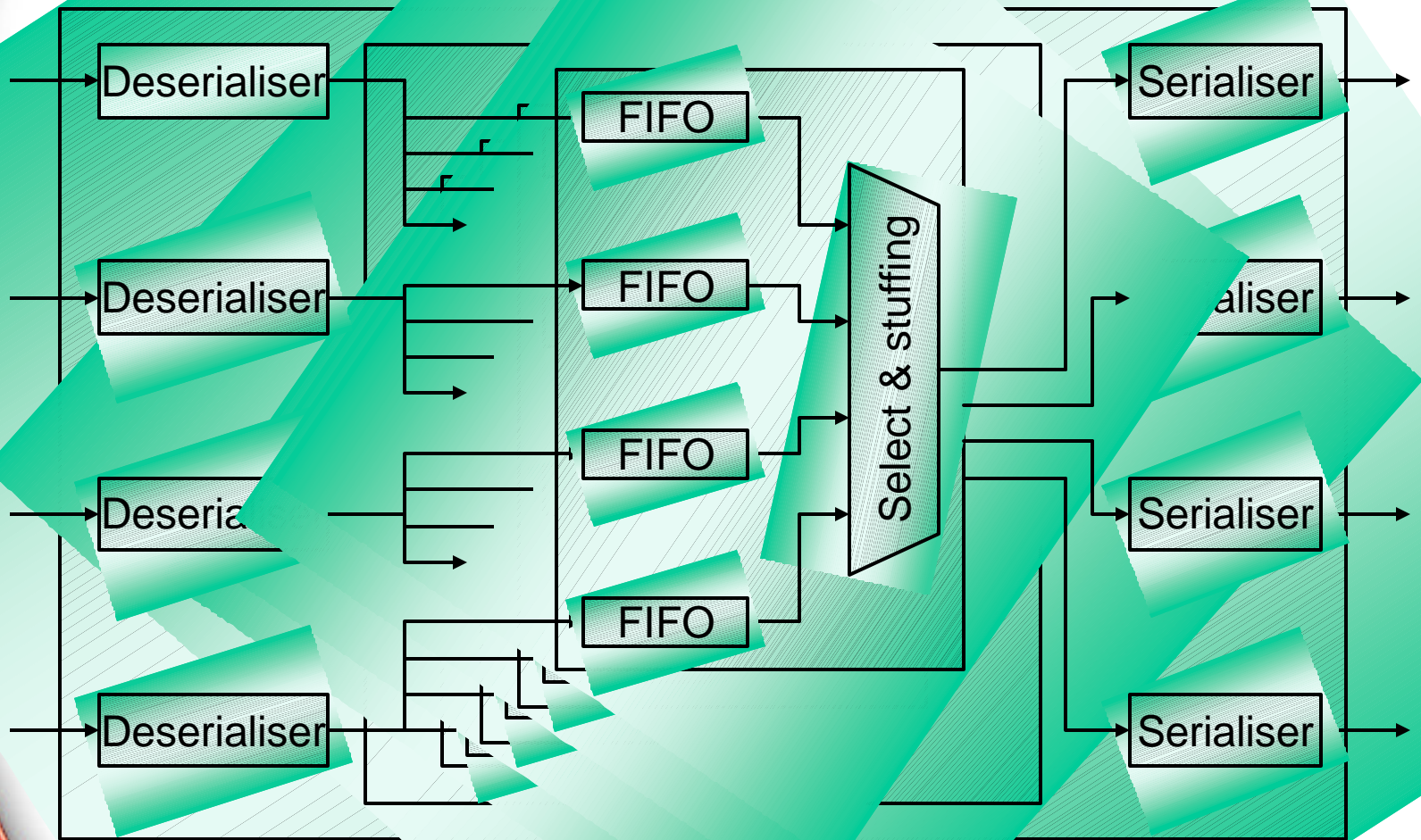


Technical Features of SCV

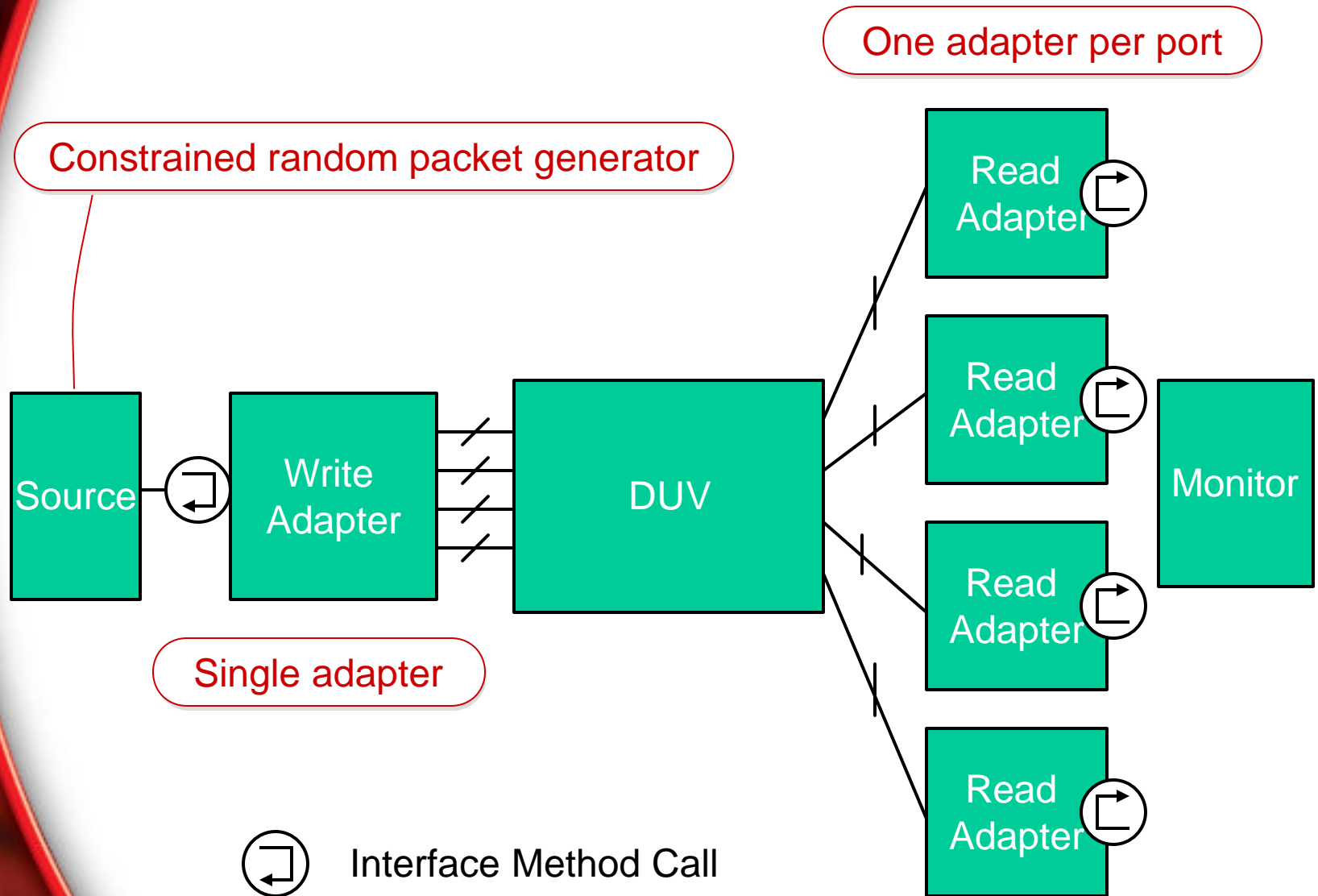
- Data introspection for SystemC, C++ and user-defined types
 - Reference counted smart pointers
 - Callback-on-assignment
- Randomization
 - Weighted probability distributions
 - Seed management for determinism
 - Hierarchical constraints
- Transaction recording
 - Multiple transaction streams and databases
 - Transaction attributes and linked transactions
 - Built-in text database (only)



Ex



Testbench Structure



Major Benefits of SCV

- Adds constrained random stimulus generation to C++ and SystemC
 - High quality pseudo-random sequence generator
 - Built-in constraint solver engine
- Transaction recording API
 - Allows use of proprietary transaction databases
 - Allows tool support for functional coverage analysis (post-processing step)



Issues with SCV - Temporals

- SCV doesn't support temporal constraints or temporal assertions
- SystemC issue -> one adapter per read port
 - Cannot identify which event caused a process to resume following a *wait()*
- Cadence Verification Extensions (CVE)
 - Synchronization Expressions address these issues



Issues with SCV - Testbench Structure

- Structuring an SCV testbench requires experience and planning
- Transaction database API
 - Proprietary implementation of API imposed limitations on overlapping threads
 - Linked transactions are tedious to code using the API



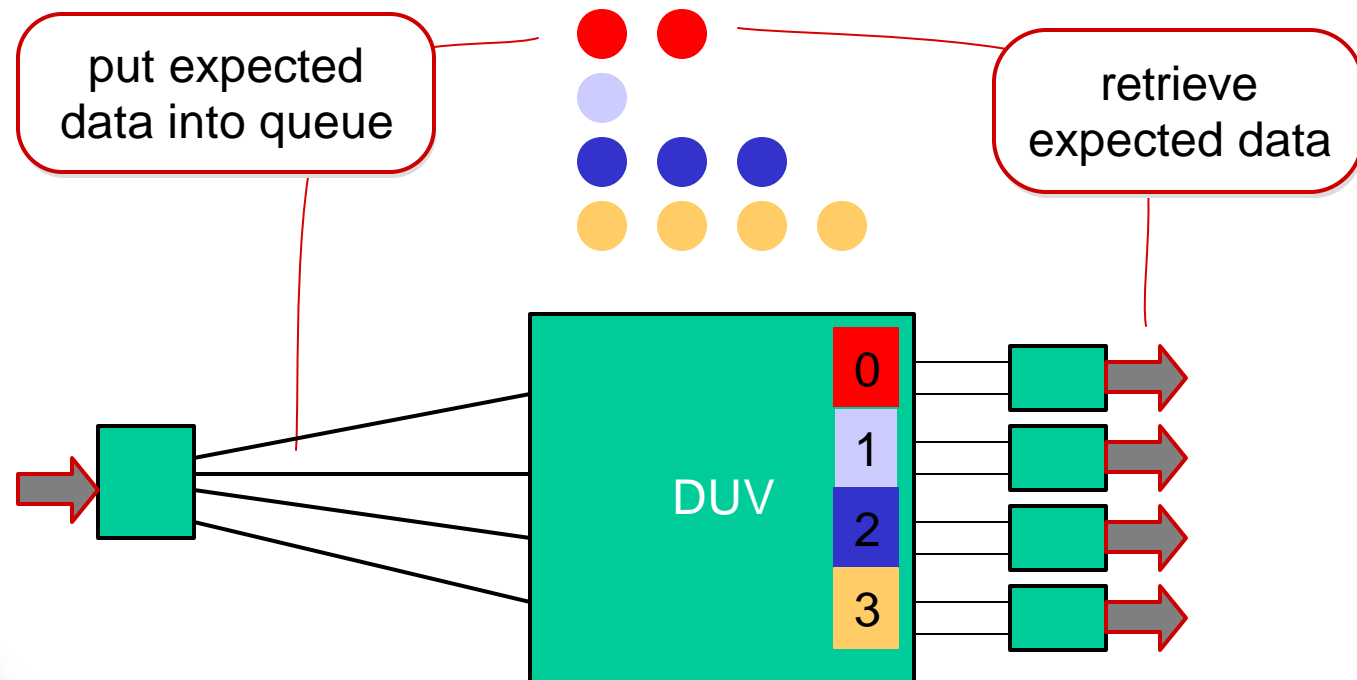
Need a Better Grasp of C++

- SCV uses smart pointers - confusing for the C++ novice
- Constrained randomization requires the use of derived classes
- STL
 - Many SystemC users can avoid the Standard Template Library
 - SCV uses STL features (e.g. `pair<>`, `vector<>`, `list<>`)



Dynamic Threads

- Dynamic threads in CVE proved useful
 - but challenging for the C++ novice
- Promised in SystemC 2.1



Conclusion

- SCV adds high-quality pseudo-random generation and transaction recording to SystemC
- Benefits from expertise in C++
- Will benefit from future enhancements to SystemC

