

# An Open Database for the Open Verification Methodology



Alberto Allara  
High-Speed IP, R/W channel group  
STMicroelectronics  
Via Tolomeo 1, Cornaredo (MI), Italy  
alberto.allara@st.com

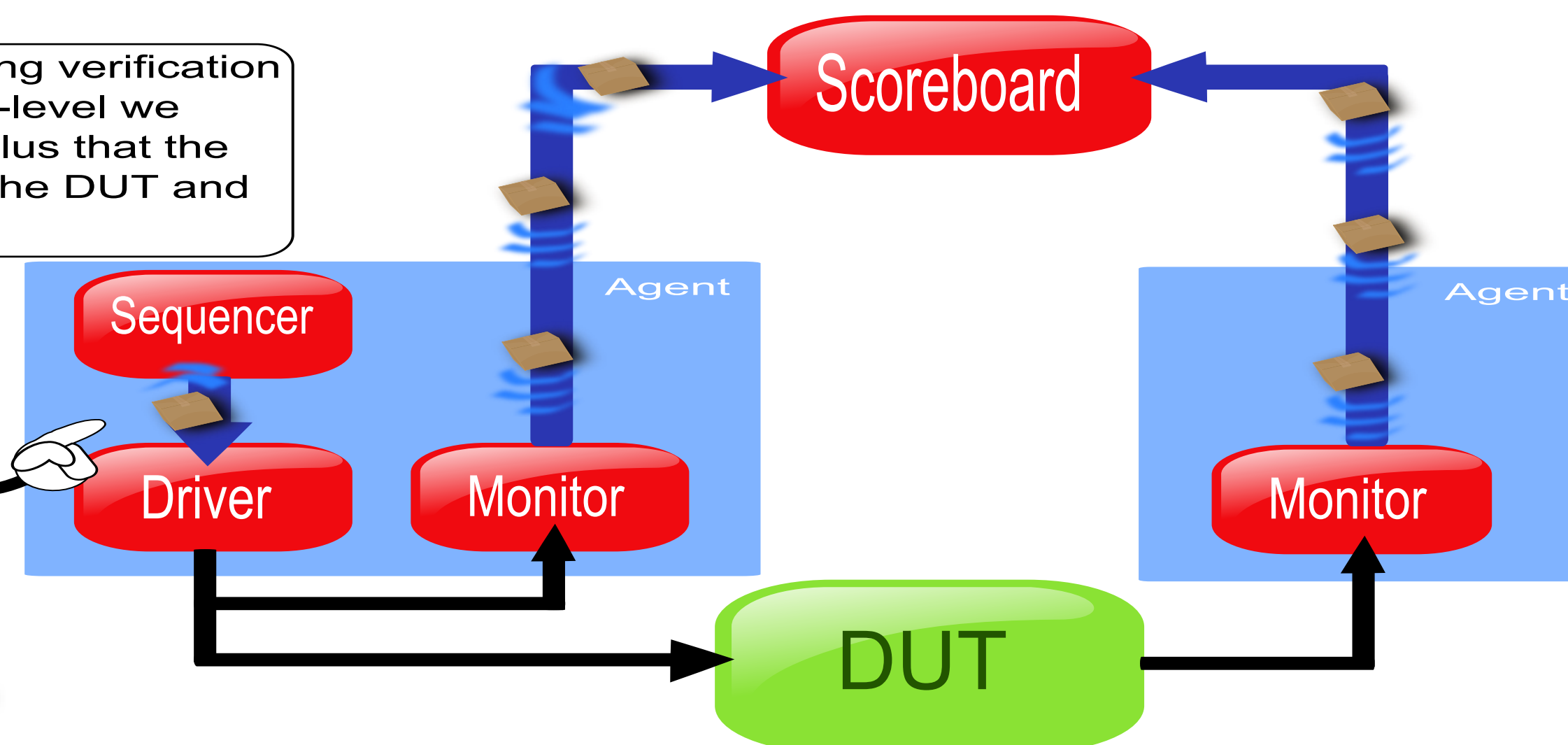
Sara Comai, Alessandro Alice  
Dipartimento di Elettronica e Informazione  
Politecnico di Milano  
Milan, Italy

sara.comai@polimi.it, alessandro.alice@mail.polimi.it



Hi!  
I am a Transaction, a verification data packet.  
There are plenty of us in your verification environment

We are exchanged among verification components and at high-level we represent both the stimulus that the environment provide to the DUT and also its responses



The support of the verification data packet by the current EDA tools presents several issues

1. Each EDA tool uses proprietary data formats for storing the data. This limits the exchange of data across different simulators
2. Current EDA tool are not optimized to be used in multi-site or in multi-user environments
3. The information saved by the EDA tool cannot be easily extended or combined with other data sources

Aim of our work is to efficiently integrate a Relational Database within a Verification environment

The verification data will become open, i.e. accessible and combinable with other sources

With the advent of Standard Verification Methodology like OVM the integration of a verification environment with a DB can exploit some reusability features of the library



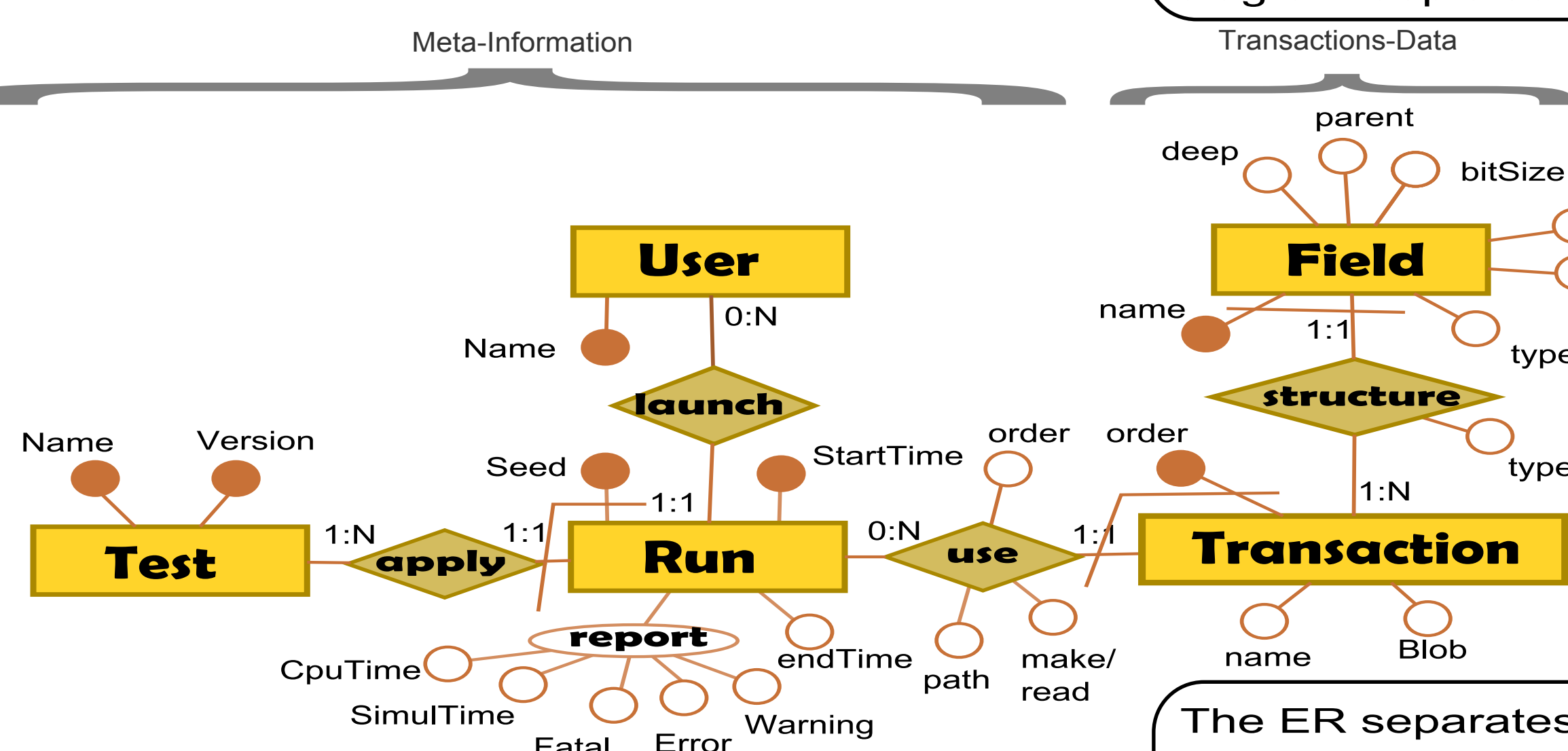
1) serialization/deserialization through pack()/unpack()  
2) data packet structure through 'ovm\_object\_utils' macros

OVM allows to serialize/deserialize any packet specialized from ovm\_object into a string of bits as a binary object (BLOB). In addition, it allows to specify the class structure of the fields that constitute the transactions

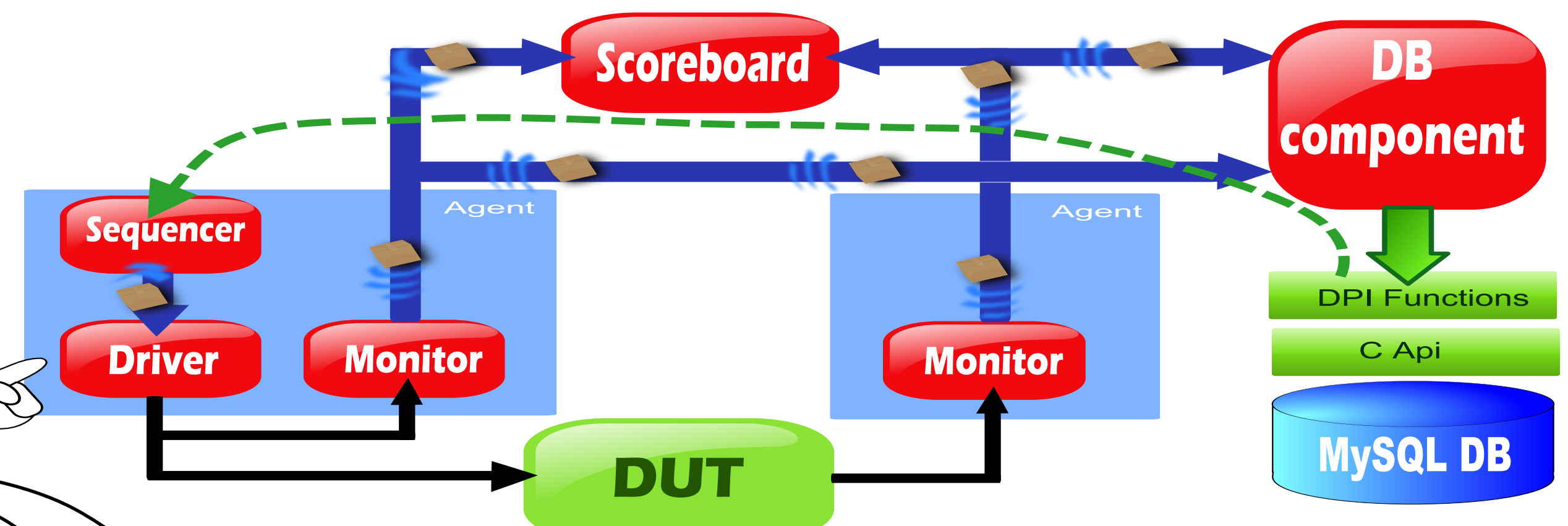
Thanks to those 2 features we can automatically store in a DB any transactions, read them back when needed and query/modify any field

Ooooho

The information that can be stored in the DataBase is exemplified in the Entity-Relationship diagram depicted on the left

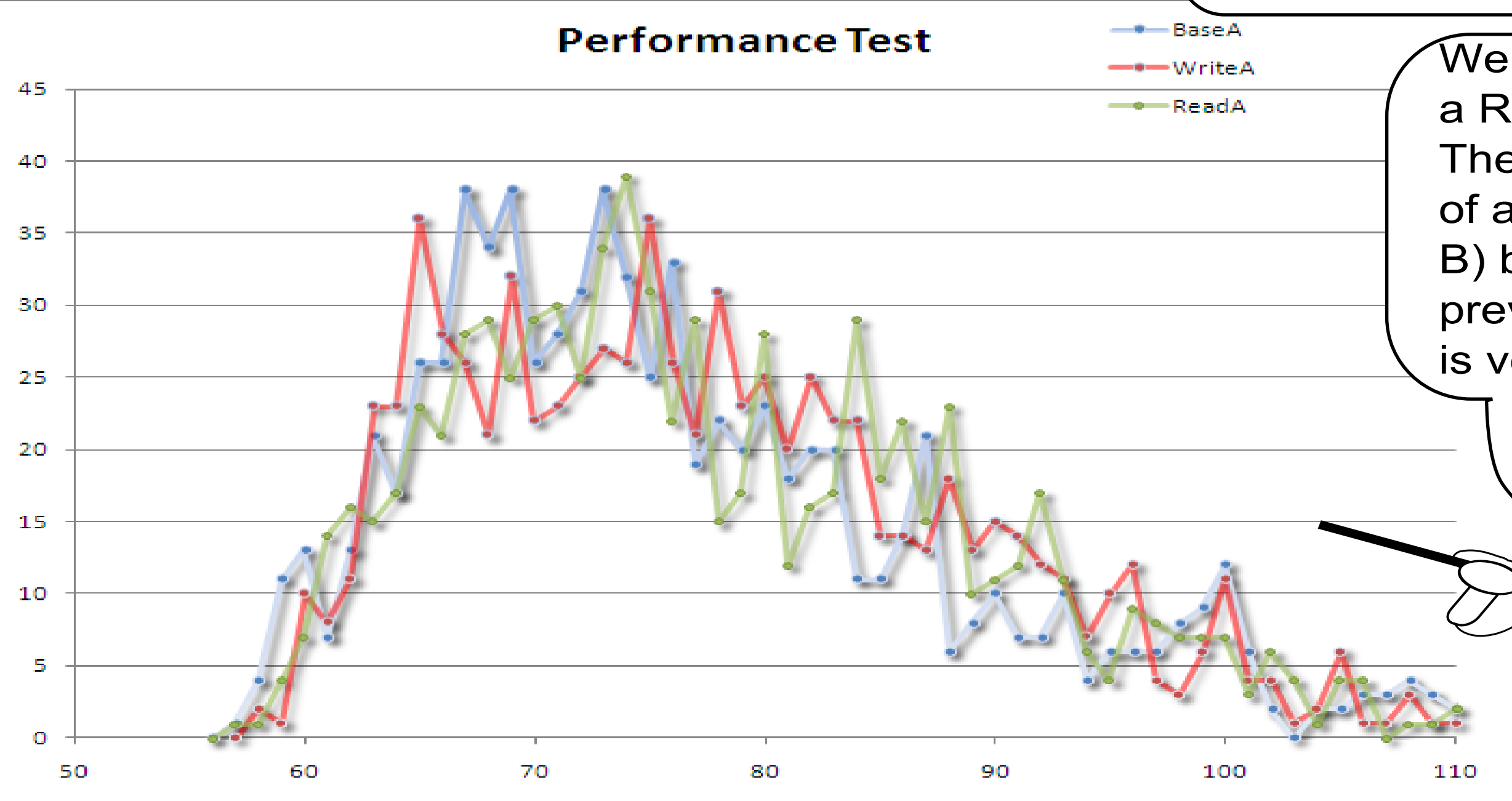


The ER separates the information related to the transaction contents from meta-information like, e.g., the user that launches the test, the start and end time of the test, and so on



A singleton OVM component (DB component) collects the interesting transactions of any verification environment and orchestrates the recording of such information into the DB by calling DPI functions  
The DPI functions are the SystemVerilog version of the set of procedural interfaces in C provided by the database

Performance Test



We tested our approach on a verification environment of a R/W channel a DSP intensive application in the data storage domain. The histogram chart on the left shows the distribution of the execution time of a Hard-Disk read operation of the channel, A) without operations on the DB, B) by storing the transactions on the DB, and C) reading back the transactions previously stored, respectively. The overhead of the use of a relational DB is very limited

We believe that the approach is promising and we demonstrated its feasibility on a real life verification environment. As a future work we plan to employ data analysis techniques to support functional coverage directly at the DB level

Happy Verification with Open DBs Cheers!!