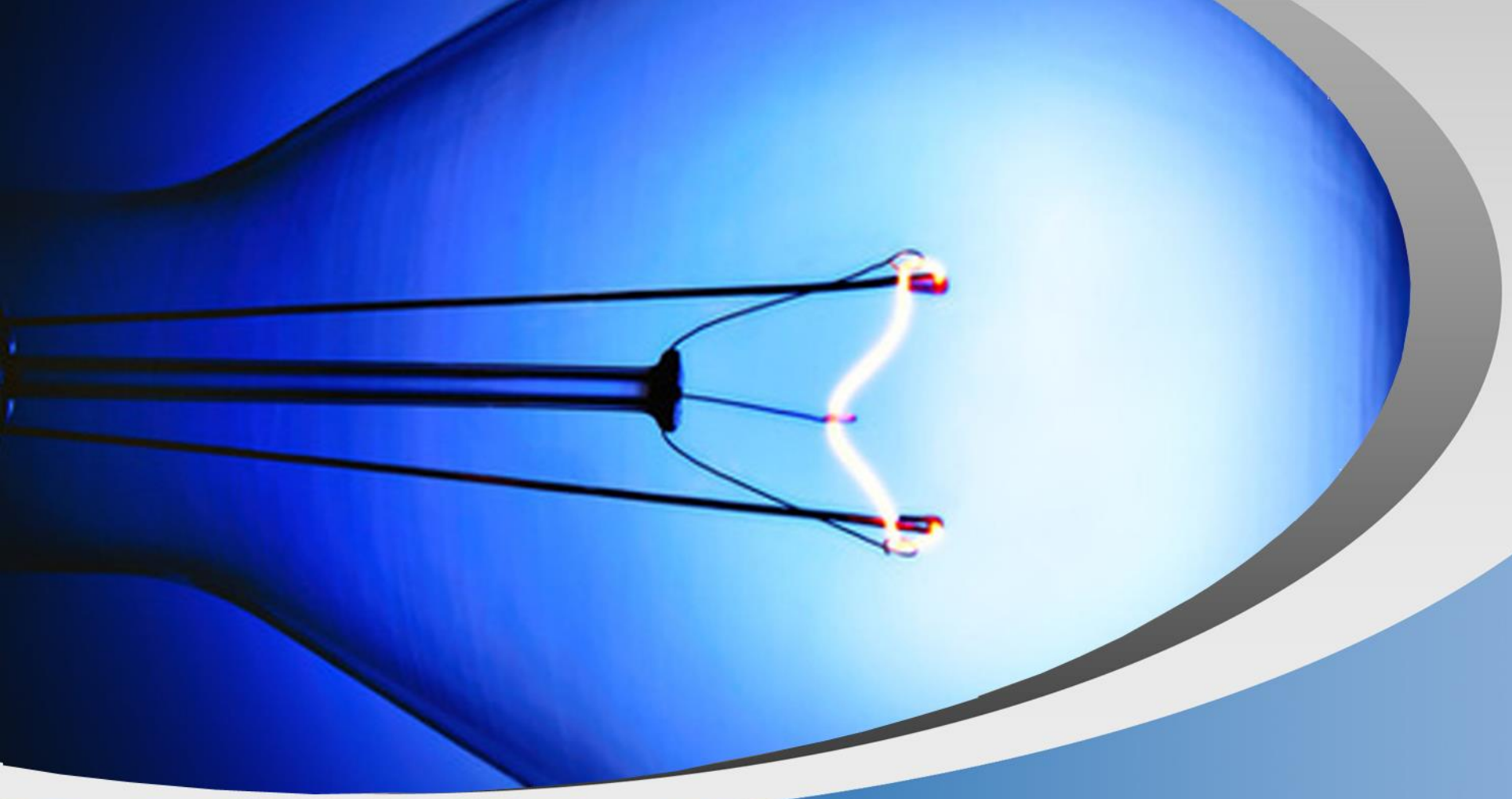


# SystemC Multi-language requirements

Accellera Multi-Language Verification Working Group

accellera SYSTEMS INITIATIVE

# Outline

- Introduction
- Multi-language Verification Working Group
- Requirements
- Implementation proposal
- Next steps

accellera SYSTEMS INITIATIVE

# Introduction

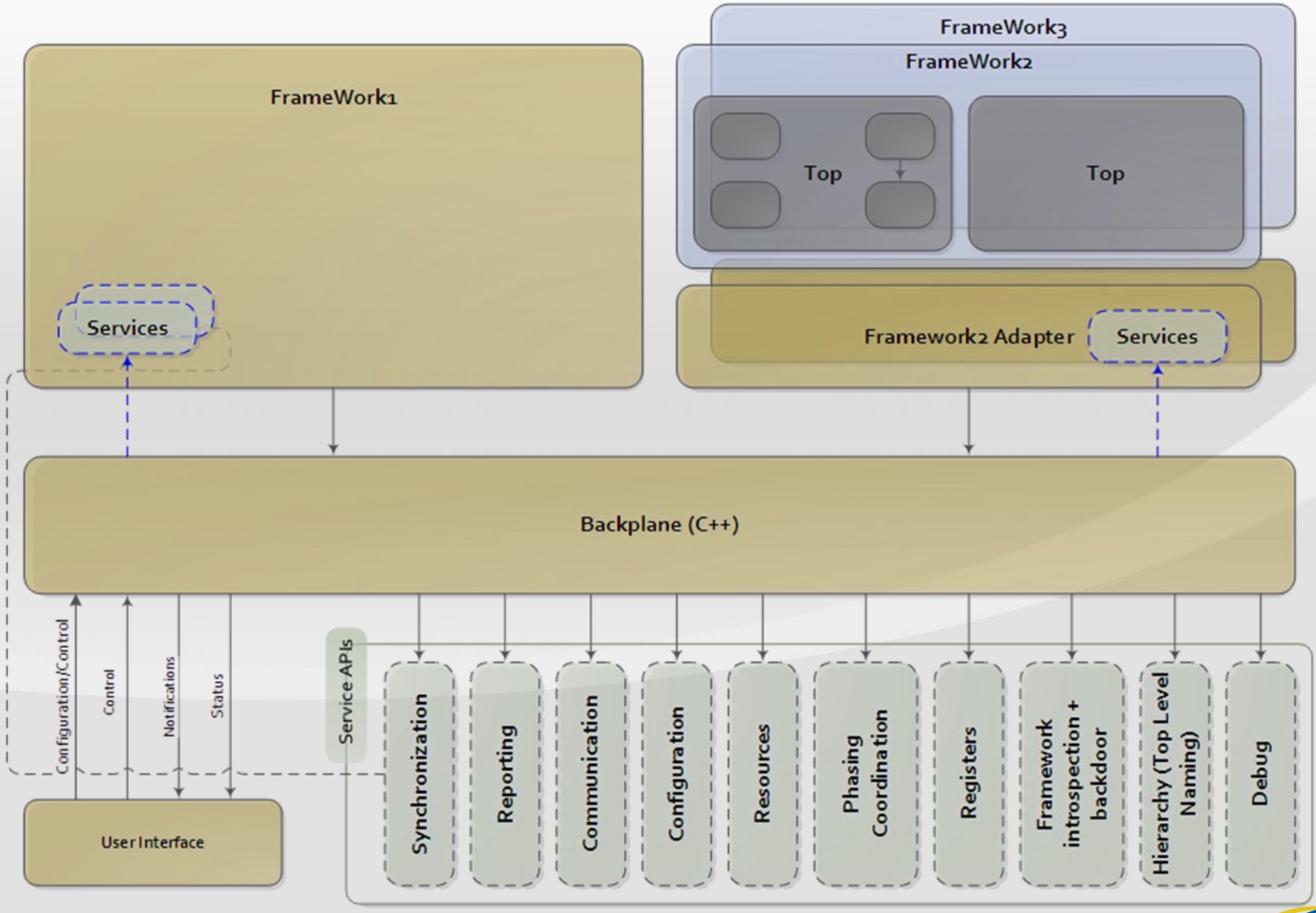
- SystemC is more and more used in advanced verification frameworks to act as test bench or Verification IP (VIP)
- In verification, test bench and VIP construction and elaboration happens separately and independently from the design module hierarchy
- In a multi-language environment, the creation and instantiation of SystemC verification components or subtrees is often initiated from other verification languages/frameworks (e.g. UVM)
- Enhancements to the SystemC standard and language are expected to support its application in multi-language verification frameworks

# Multi-language Verification WG

- **Chair: Warren Stapleton (AMD)**
- **Vice Chair: Faris Khundakjie (Intel)**
- **The mission of the MLVWG is to create a standard and functional reference for interoperability of multi-language verification environments and components.**
- **The MLVWG have reviewed requirements and are developing an open source proof-of-concept library for creating a standards-based approach for combining verification environments developed in different languages/frameworks.**
- **In addition, the group will look at ways to enable the introduction of UVM (Universal Verification Methodology) concepts in other environments and languages that come from legacy projects or developed with other frameworks for beneficial reasons**

accellera SYSTEMS INITIATIVE

# MLVWG – Architecture




accellera SYSTEMS INITIATIVE

# Requirements

## related to UVM-SV ↔ SystemC interoperability

- Allow instantiation of SystemC modules acting as testbench/VIP component after start of simulation, e.g. introducing dedicated “testbench elaboration” callback(s)
- These additional “testbench elaboration” callbacks should not be called by the SystemC module registry, but by the application (e.g. UVM environment)
- For proper time synchronization between SystemC and the external verification environment, sc_start(<time>) should stop at the first delta of the time in question, not before
- Introduce a dedicated function to start a test (similar to UVM run_test), including the invocation of the “testbench elaboration” callback(s), UVM phasing and ML time synchronization

accellera SYSTEMS INITIATIVE

# Implementation proposal

- **Introducing “quasi-static elaboration”**
- **Introducing multi-language-enabled time synchronization**

- ***NOTE: The syntax and naming applied for the implementation is just an initial proposal. Definitive syntax and naming is considered part of the alignment discussion and prototyping exercise***

# Quasi-static elaboration

- **Introducting “testbench elaboration” specific callback(s)**
  - quasi_static_end_of_construction
  - quasi_static_end_of_elaboration
  - quasi_static_complete_binding
  - quasi_static_start_of_simulation (which also closes the quasi-static stage)
- **All methods are public**
- **These callbacks are not called by the SystemC kernel, but by an application (e.g. UVM)**
- **When quasi_static_start_of_simulation callback is invoked (e.g., by the UVM phase engine), it merges the new registries into the old ones, places the old ones into their places and deletes the new ones.**

# ML-enabled time synchronization

- **Problem:  sc_start(<time>) causes the SystemC kernel to stop just before the first delta of the specified time, rather than entering it.**
- **Implementation options**
  1. Introduce new method sc_simcontext::co_simulate(<time>) which actually enters the first delta of the specified time.
  2. sc_simcontext::simulate(<time>, <flag>) if <flag> is enabled, the simulation steps after entering the first delta of the specified time, otherwise the default behavior is maintained.

# Next steps

- **Discussion in MLVWG and LWG on requirements and technical feasibility**
- **Explore standardization and implementation opportunities**
  - SystemC standard API update (incorporate in upcoming IEEE 1666 cycle?)
  - Update Accellera SystemC PoC
- **Discuss topic at Accellera's SystemC Evolution Day 2018**
  - Talk/abstract submission under development
- **Inform Technical Committee and TC Chair on this X-WG initiative**

# Acknowledgment


- Alex Chudnovsky, Vitaly Yankelevich - Cadence
- Bryan Sniderman, Warren Stapleton - AMD
- Faris Khundakjie – Intel


accellera SYSTEMS INITIATIVE

# Interesting links

- **MLVWG public page**
  **http://www.accellera.org/activities/working-groups/multi-language**
- **MLVWG Accellera working area**
  **http://workspace.accellera.org/apps/org/workgroup/mlwg/**
- **MLVWG at github**
  **https://github.com/OSCI-WG/Multi-Language**