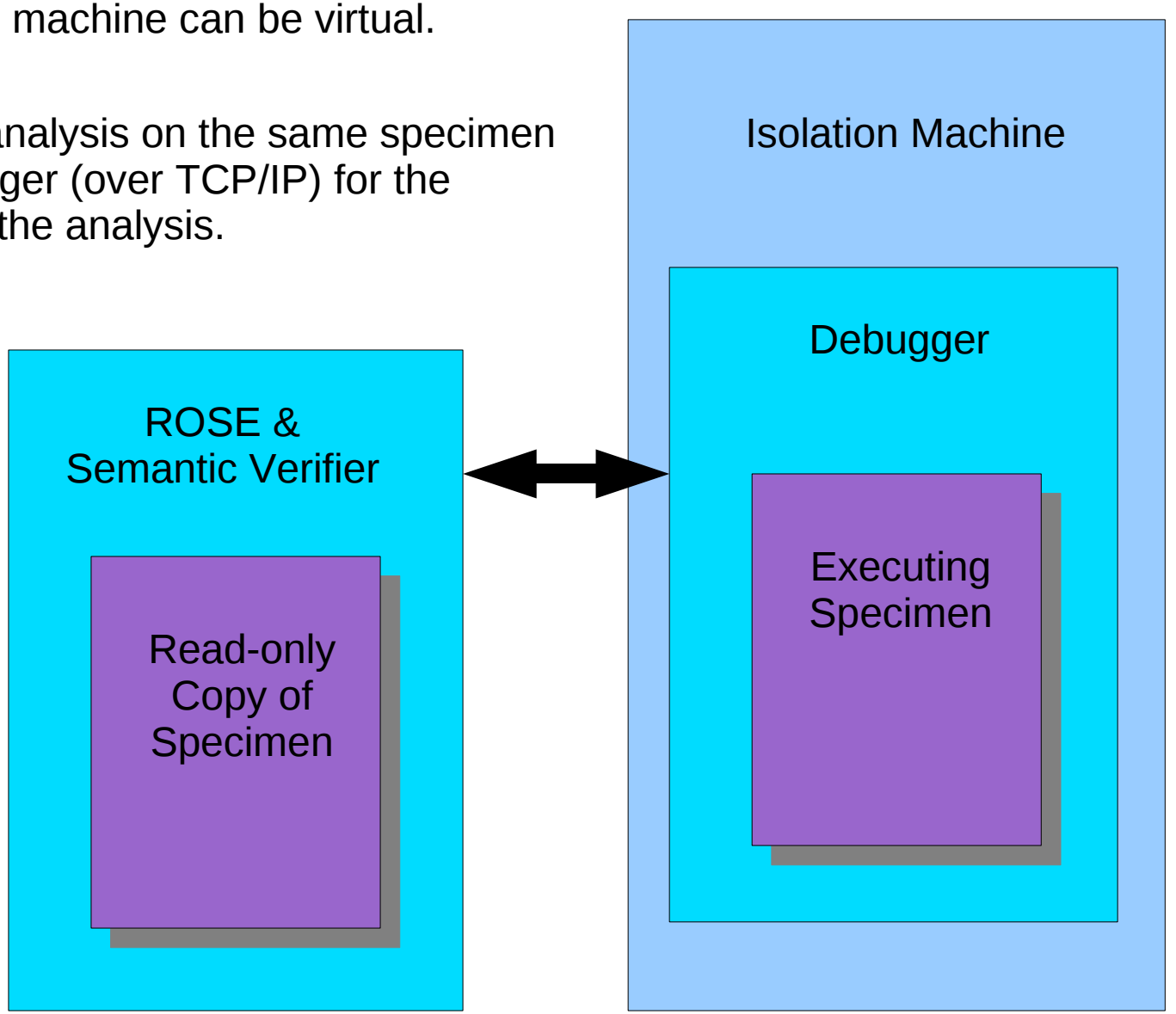
Verification of Instruction Semantics

- Correct analysis of binary executables depends upon the correct understanding and implementation of instruction semantics.
- Chip manufacturers provide detailed descriptions of each instruction which can be used to build an instruction semantics layer in software.
- ***Can we verify that our software implementation matches the hardware implementation?***

Semantic Verification Architecture

The specimen can be isolated on its own machine, perhaps a different architecture than ROSE. It runs under a very simple debugger. The isolation machine can be virtual.

ROSE performs static analysis on the same specimen and contacts the debugger (over TCP/IP) for the dynamic component of the analysis.



Method

- ROSE and the debugger execute instructions in tandem—ROSE in software, the debugger on hardware.
- ROSE simulates execution using its Instruction Semantics Layer, obtaining data (registers and memory) from the debugger.
- After each instruction executes, ROSE compares its simulated state with actual state reported by the debugger.
- Differences are reported as bugs and a trace of each offending instruction is displayed.