



Unit Testing Framework for Operating System Kernels

Walter, Maxwell; Karlsson, Sven

Publication date:
2014

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Walter, M., & Karlsson, S. (2014). *Unit Testing Framework for Operating System Kernels*. Poster session presented at 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI '14), Broomfield, United States.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Problem

- We need a way to test a new OS kernel as it is being developed
 - Before hardware drivers are fully implemented
 - With multiple different hardware setups
- Current methods rely on manual information retrieval

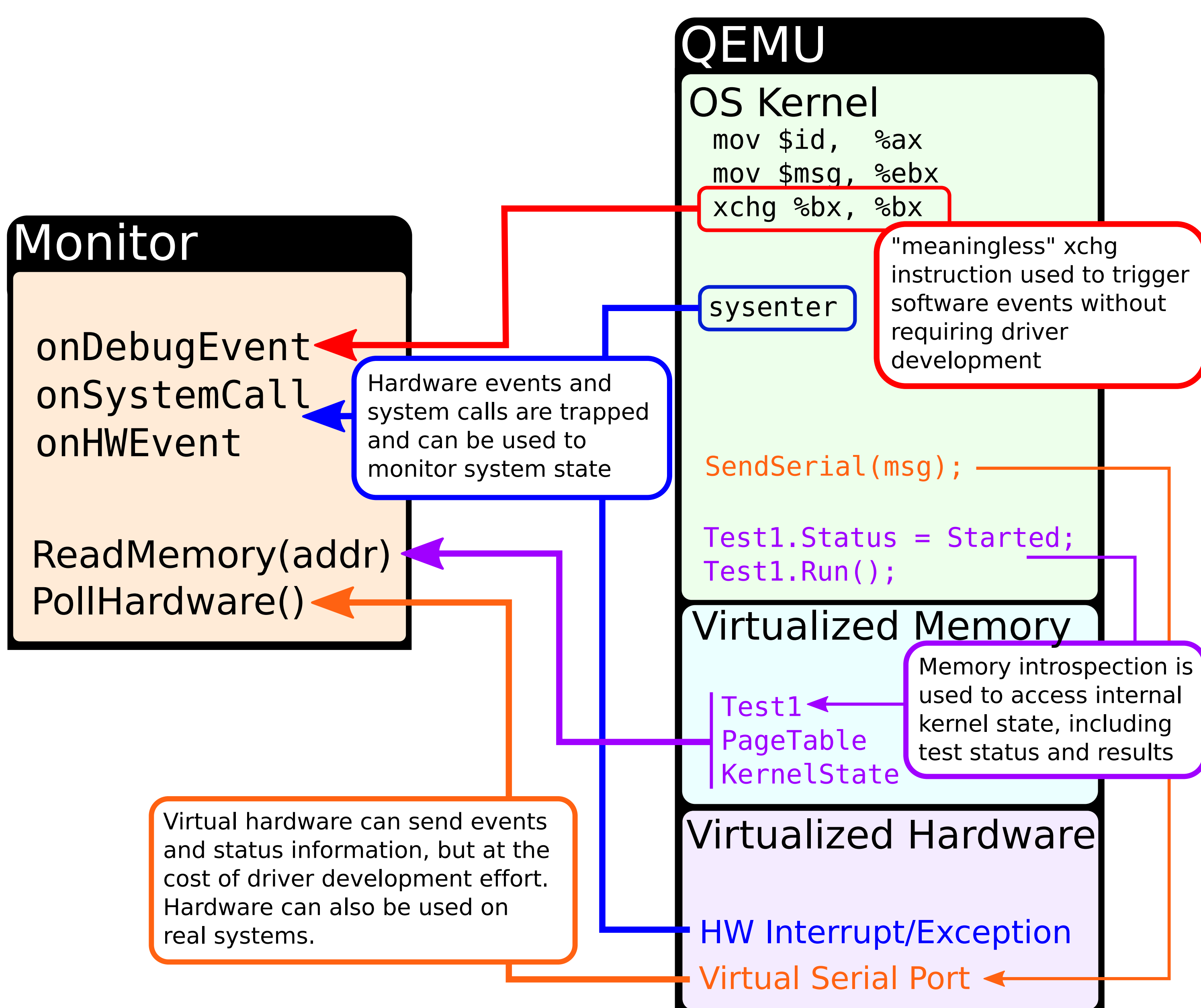
Approach

- Run the OS kernel in a **virtual machine** (VM)
 - Use **QEMU** as our virtualization platform
 - Create virtual hardware configurations as needed
 - Use virtual machine introspection (VMI) to access internal state
- Create an in-kernel **Testing API** that
 - Provides a way to create, start, and report on tests
 - Provides software test points for VMI

Contributions

- A **testing framework** for operating system kernels consisting of
 - A management interface for testing HW configurations
 - A test monitor interface for accessing kernel and VM state
 - An in-kernel testing API

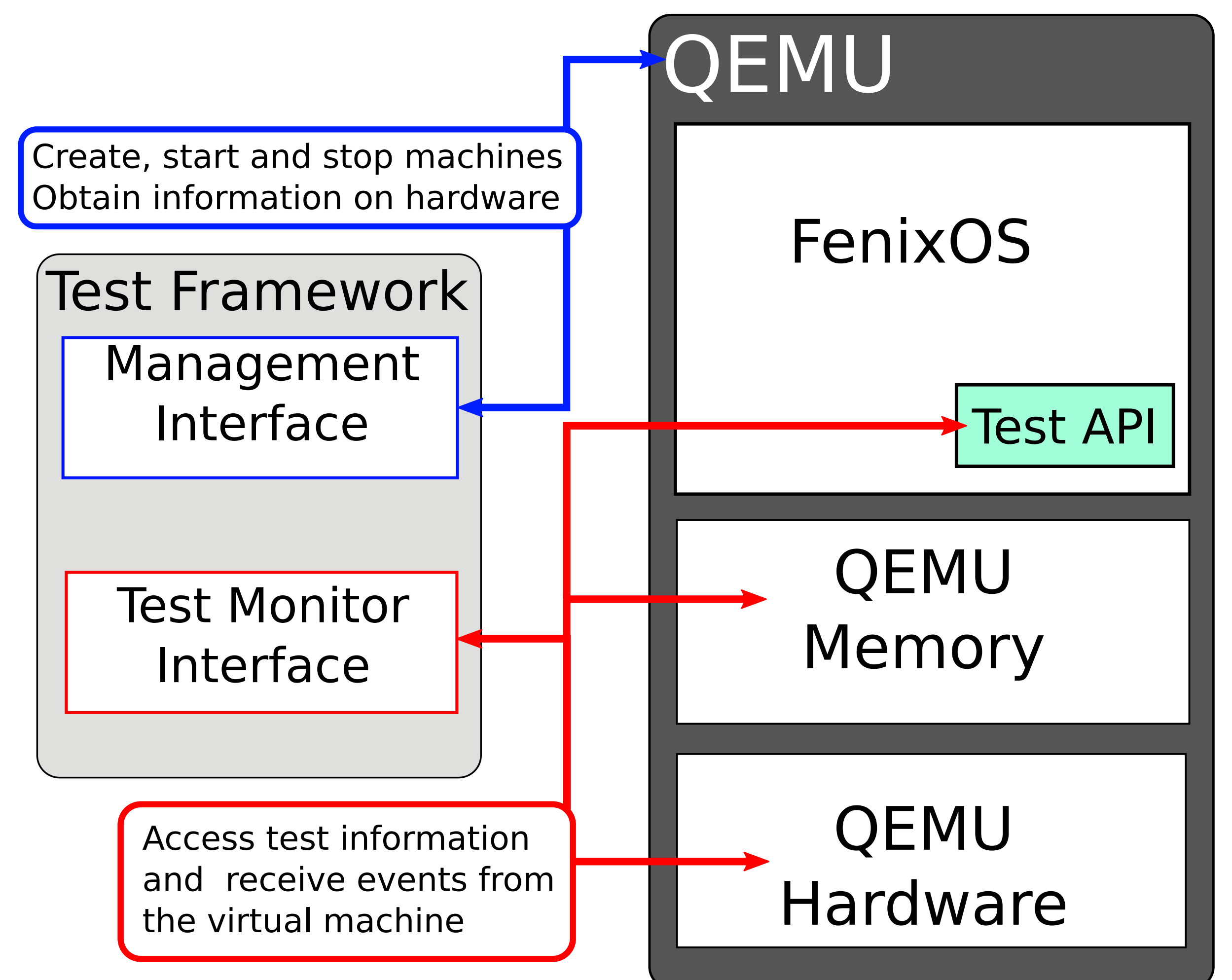
Framework and VM Communication



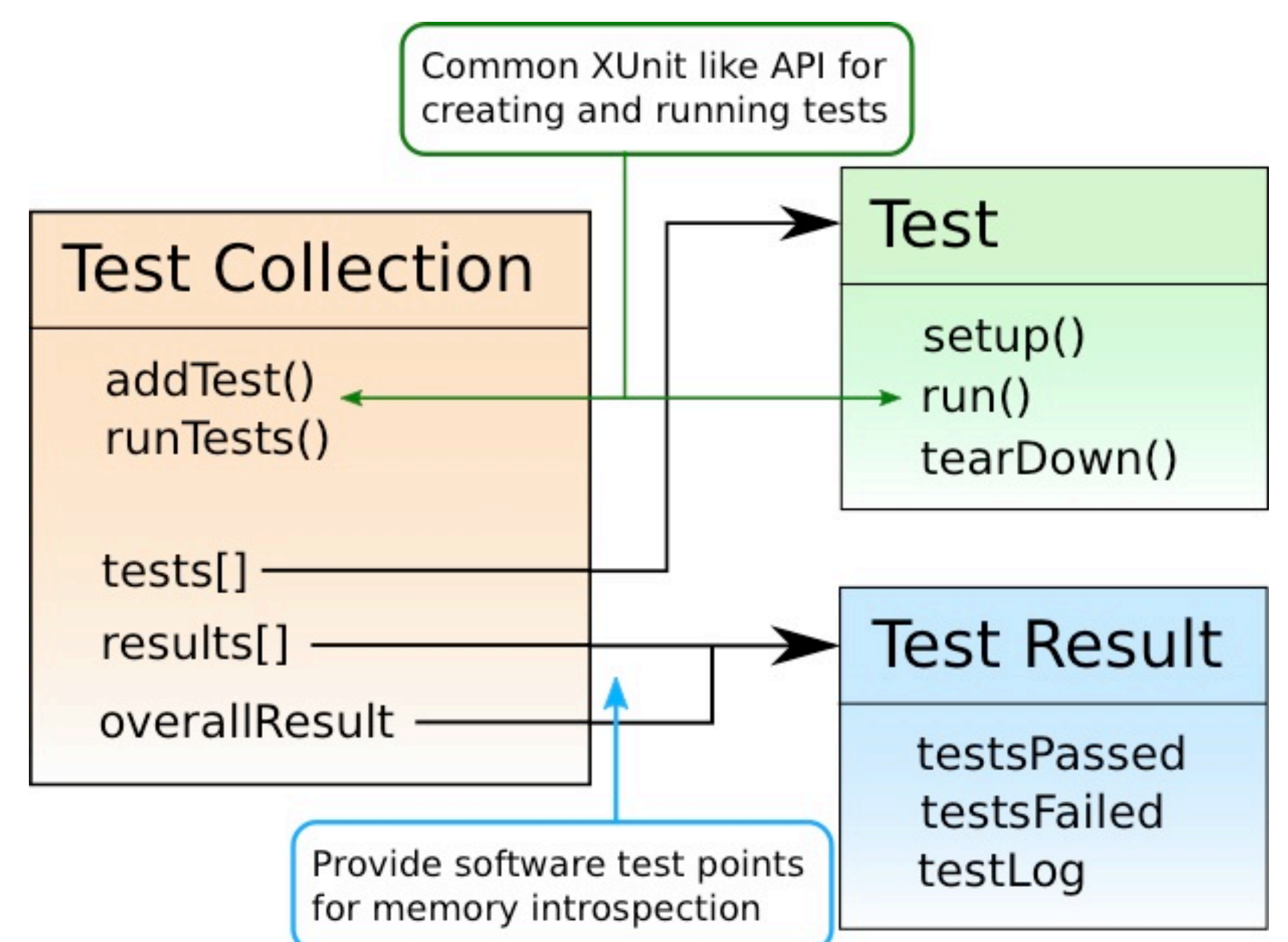
Framework Limitations

- Virtualized environments present idealized, not real, hardware
 - Introduces potential inaccuracies into testing
 - Can be partially mitigated with hypervisor and HW pass-through
- Dynamic data may not be available for introspection
 - Locations may be unknown
 - Data may change before or while it is being observed

Architecture



In-kernel Test API



Conclusions

- The testing framework is currently being used for development
 - Provides status information before HW and memory is initialized
 - Ensures future development does not break working code
- A sample GUI using the framework is shown below
 - Uses memory introspection to retrieve information

