Project Report

1.1. Project: OKL4 Microkernel

1.2. Duration: 2010.5 ~ 2011.5

1.3. Brief Description ok OKL4 Microkernel

OKL4 is a highly flexible, general purpose, high performance microkernel from Open Kernel Labs (acquired by General Dynamics in 2002). It provides a minimal layer of hardware abstraction on which various *operating system personalities* can be built using modules. It isolates each component in the system from effects of programming errors or malicious code contained in other components. OKL4 applies the principle of *minimality with respect to security*. As such, a service or *feature* is included in OKL4 only if it is impossible to provide that service outside OKL4 without sacrificing security, or if including that feature in OKL4 provides significant benefits without increasing the complexity of OKL4.

OKL4 allows for trust and security models to be implemented using the *libokl4 API*. *libokl4* takes a layered approach to providing functionality. At a low level of abstraction, *libokl4* provides a relatively lightweight interface of underlying kernel functionality comprising of the *execution context interface* and the *communication interface*. The execution context interface provides access to threading, MMU, cache, and capability based domain security interfaces. The communication interface provides access to the messaging, notification, synchronization, interrupt and event interfaces. At the highest level of abstraction, *libokl4* provides the *high level interface to* the user with access to concepts such as *memory sections*, *realms*, *protection domains* and *extensions*. Utility interfaces of the library provide basic data structures such as allocators and system call bindings and comprises of the *allocator interface*, *memory pools interface* and the *environment interface.* At this level, *libokl4* provides only a minimal level of abstraction, placing the responsibility of regulating the execution environment on the user. In each case, the user is encouraged to use the highest level available API.

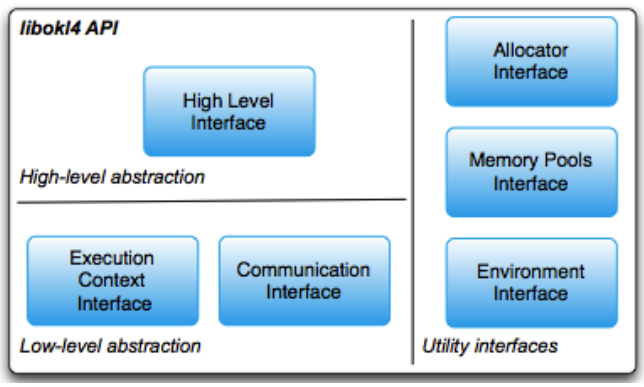

Figure 1: Levels of abstraction in libokl4 library.

OKL4 has been deployed mainly for Qualcomm platform.

1.4. Role

■ As a senior software engineer, I was involved in refactoring and reimplementation of the OKL4 for its performance and memory footprint. Tasks are:

- Threading: supported mutex attribute in the kernel and libokl4, refactored priority inheriting and syncpoint
- MMU/Cache/TLB: supported cache invalidate API, cache API by range, refactored cache API and TLB
- Refactoring memory pool: reduced foot print, supported unaligned memory pool
- Removed libgcc: replaced with non-GPL based library
- Refactoring for small footprint
- Refactoring communication API: removed heavy-weight IPC, reimplemented notification-based IPC.
- Supported global lock for OK:RTX product: supported multi-threaded environment for OK:RTX.
- Testing scripts for hardware simulator

Appendix

■ Definition & Acronyms

■ References

1. http://www.ok-labs.com