

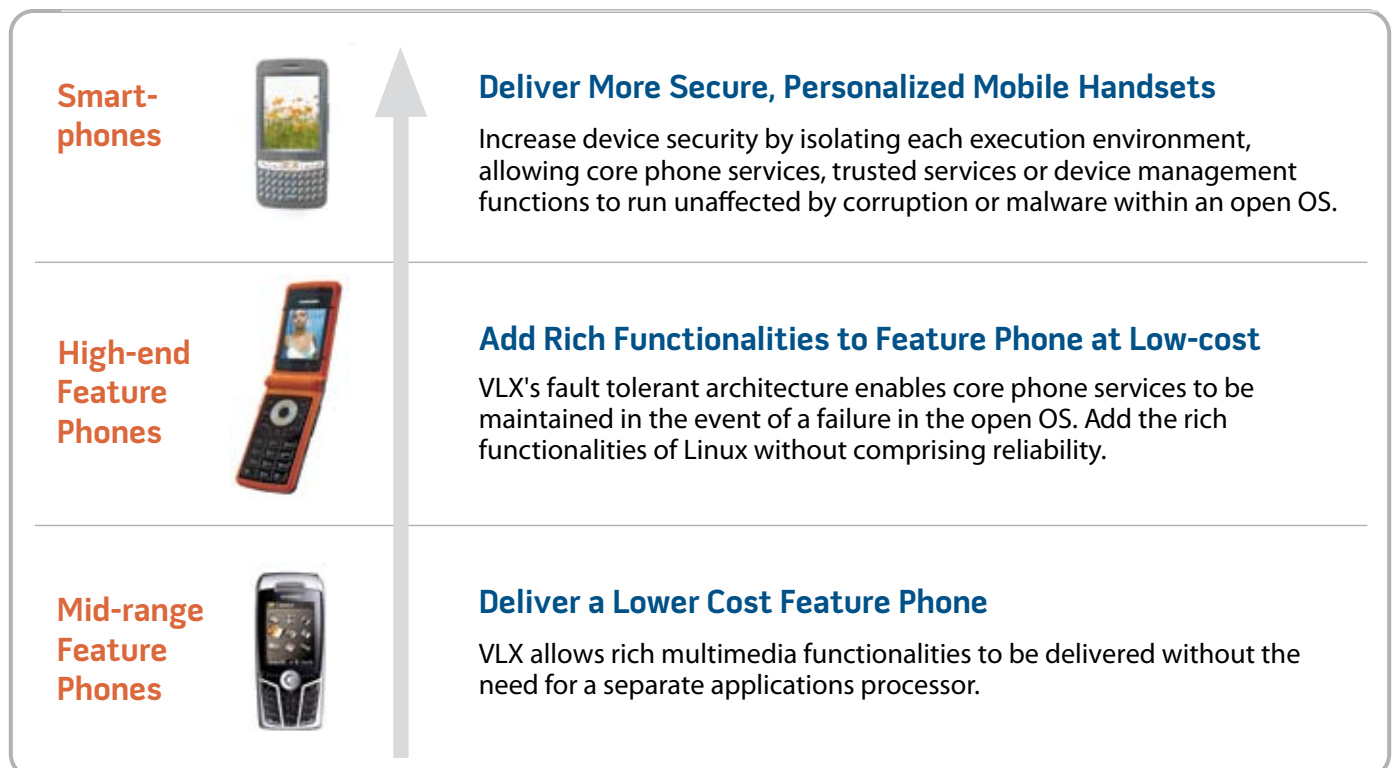
## VLX for Mobile Handsets

### Overview

Mobile handset developers struggle with the need to lower hardware and software costs while also providing a rich multimedia user experience, securing the device from malware, and getting their product to market as quickly as possible.

VirtualLogix enables all feature and smartphone designs to benefit from increased security and fault tolerance by using VLX to isolate security and device management functions from potential corruptions that may be present in the open operating system. VLX also enables low cost, single-CPU phone designs to provide full user features, normally reserved for more expensive multicore designs, by enabling open operating systems such as Linux to run together with the mobile phone's modem stack on a single processor core.

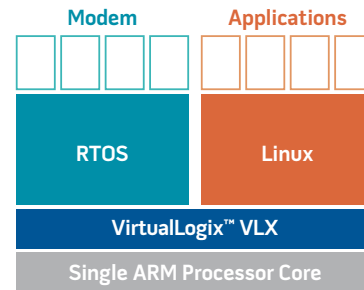
The real-time virtualization technology in VLX for Mobile Handsets allows manufacturers to reduce development and deployment costs by adopting a common software platform across disparate hardware designs from low cost single-core feature phones to multicore smart phones.



## Use-case Examples

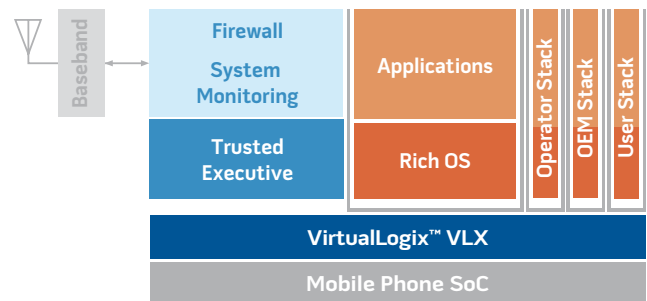
### Deliver a Lower Cost Feature Phone

- Deliver rich user features through Linux
- Eliminate the cost of an additional application processor
- Reuse existing proven modem stack and RTOS
- Isolate proprietary IP from open source software
- Typically applied to a feature phone architecture using the baseband processor for both the wireless stack and application stack



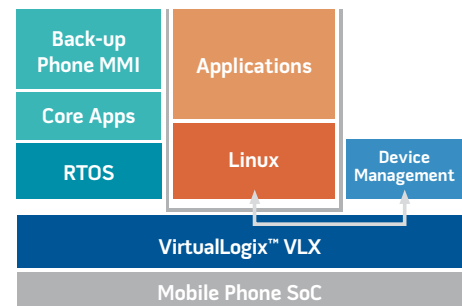
### Deliver More Secure, Personalized Mobile Handsets

- Isolate and protect core phone services from mobile malware through OS independent firewall
- Secure DRM protected content and financial transactions controlled by SIM card and more...
- Maintain continuity of service or voice in case of failure in the open OS
- Multiple application stacks (i.e., operator, OEM, user) enable easy device personalization
- Typically applied to a smartphone architecture with a dedicated application processor



### Add Rich Functionalities with Improved Fault Tolerance

- Management agents run in trusted container, isolated from application OS
- OS Monitoring and automatic restart allows the system to be repaired or restored independently of the rich OS
- Fault tolerant architecture provides revenue protection by maintaining core phone services in the event of a failure of the open OS
- Control phone behavior independently of native software and user download applications
- Minimize the risk of releasing products to the market without complete validation of complex multimedia OS and applications
- Typically applied to a feature phone architecture containing a single ARM core and hardware accelerators



### VirtualLogix Actively Participates in Industry Organizations



## Key Features

### Supported Processor Cores

VLX for Mobile Handsets has off-the-shelf support for the following CPU families:

- ARM7xx
- ARM9xx
- ARM11xx
- XScale™ PXA25x & PXA27x

VLX for Mobile Handsets is incrementally modifiable for other ARM core families, please contact us for details.

### Linux and RTOS Support

VLX for Mobile Handsets enables multiple operating systems to run concurrently on a single ARM processor core. Each operating system can be of the same type (e.g. all Linux), different types (e.g. Windows Mobile™ + RTOS), or even different releases of the same OS (e.g. Linux v2.4 + Linux v2.6). Unlike other virtualization solutions, the VLX architecture ensures that the RTOS continues to provide guaranteed response to the system regardless of the work-load of other operating systems running on the same core. This high level of performance ensures that dual core-based solutions where one core runs a RTOS and the other runs applications, can be safely and quickly migrated to a single core architecture.

### Guest OS: Performance/Isolation Configurable

VLX for Mobile Handsets does not enforce a single “virtualization” philosophy onto developers. Instead, a build time configuration parameter allows an OS to maintain direct access to the hardware, even whilst running in a virtualized environment. This direct-access approach provides an unmatched level of performance and design flexibility for the implementation of both trusted and untrusted software.

Alternatively any guest OS can be configured to operate within an isolated “sandbox” where it’s physical memory and I/O device access is restricted by VLX according to build-time parameters. This sandbox ensures that bugs or malware within the isolated OS are unable to affect the functionality of any other OS coexisting on the platform.

### Inter-OS Communications

VLX for Mobile Handsets provides high-performance and rich OS inter-communication, data sharing and task coordination capabilities which ease the creation of usable, real-world multi-OS solutions. VLX for Mobile Handsets supports both low-level and high level communications services. Low-level services consist of zero-copy shared memory blocks, FIFOs, and cross-interrupts for inter-OS data sharing and task coordination. High-level services provide an OS with turn-key access to virtual inter-OS hardware such as virtualized Ethernet and UART, or to a shared physical device such as Audio I/O, Keypad, Touch screen, and LCD Screen. To enable the creation of virtual devices customized to the customer’s specific product, VirtualLogix’s Device Driver Framework provides a rich application programming interface (API) and set of services for exactly this purpose.

### Schedulers

VLX for Mobile Handsets includes priority, fair-share, and fine-grained schedulers to prioritize each OS access to platform resources for multiple product scenarios.

### Cross-development Tools

The VLX architecture ensures that developers can continue to use those OS-specific development tools to which they have become accustomed. VirtualLogix also includes a set of Eclipse-based tools which facilitate the configuration, build, and monitoring of a VLX-based system.

### Performance

VLX for Mobile Handsets has been designed to be fast, small and efficient. Special care has been taken to avoid unnecessary latency in interrupt response or context switch times. VLX performance overhead presents a negligible load on system resources with a footprint that can be as small as 64KB.