Hypervisor based Smartphone Solution

Spreadtrum Smart Phone Solution

SPREADTRUM
Innovation changes the future

- Single Core
  - Cortex A5 1GHz
  - OpenGL ES 2.0 (Mali 300 – Mali 400 UP version)
  - Neon Multimedia Processor
  - VFP
- Hypervisor Based
  - Red Bend (VirtualLogix) Hardware Partition Hypervisor

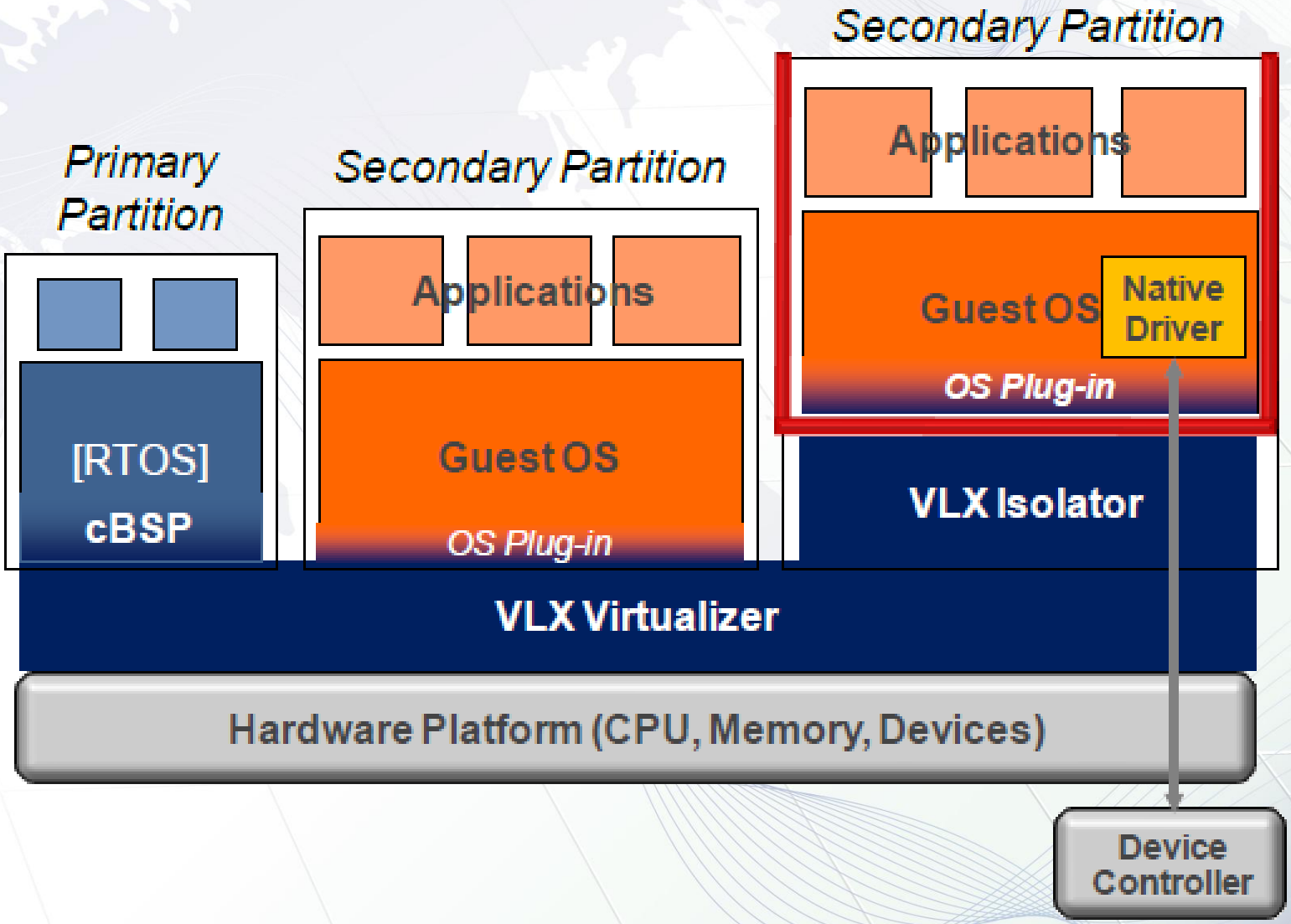
Hypervisor

- **Hypervisor**, also called **virtual machine manager** (**VMM**), is one of many hardware virtualization techniques that allow multiple operating systems, termed *guests*, to run concurrently on a host computer.

Leading Embedded Hypervisors

- Red Bend VirtualLogix
  - Hardware Partition between guests
- OK-Labs Microvisor
  - Microkernel based embedded hypervisor
- Xen ARM Project
  - Open Source
  - Led by Mr. Sang-Bum Suh from Samsung

VirtualLogix Architecture



VirtualLogix Main Components

SPREADTRUM
Innovation changes the future

- VLX Virtualizer
  - Create and manage physical *memory partitions, allocate HW platform resources and schedule the processor(s)*
  - Guarantee real-time deterministic performance to the real-time operating system (RTOS), if any, running in its *"primary" partition*
- VLX Isolator
  - Isolation of the associated guest operating system for security purposes

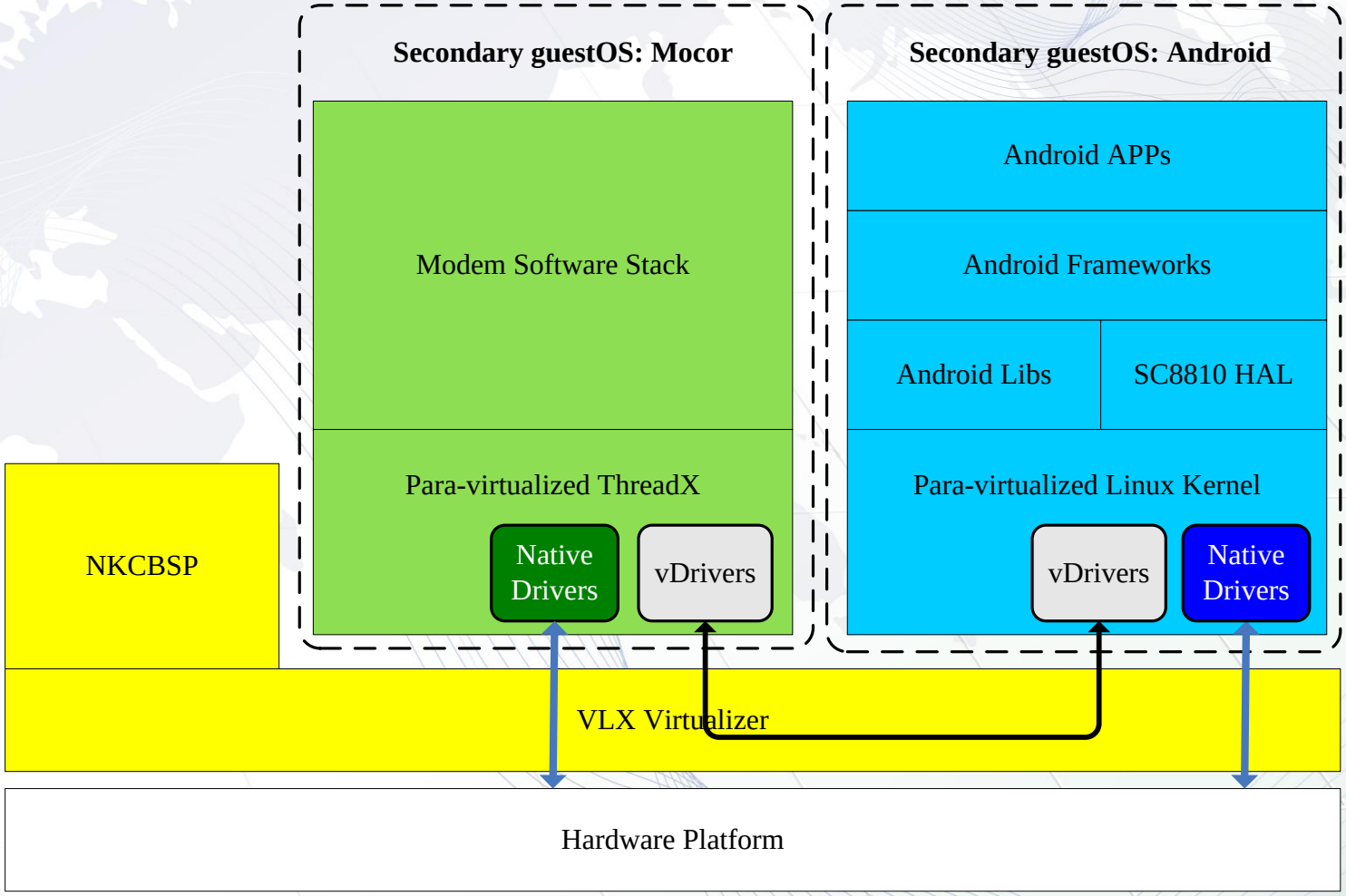
VirtualLogix Main Components

- VLX Virtual Device Framework
  - Distributed implementation of the drivers for each peripheral device found on the hardware platform
  - “Back-end” (BE) device driver for each peripheral runs in the context of the operating system owning the peripheral device controller with its corresponding native device driver (virtual device server)
  - “Front-end” (FE) device driver(s) run in the context of operating system(s) supporting application environment requiring access to such device (virtual device client)
- VLX Virtualized Linux/Android
  - Para-virtualized version running on the hypervisor

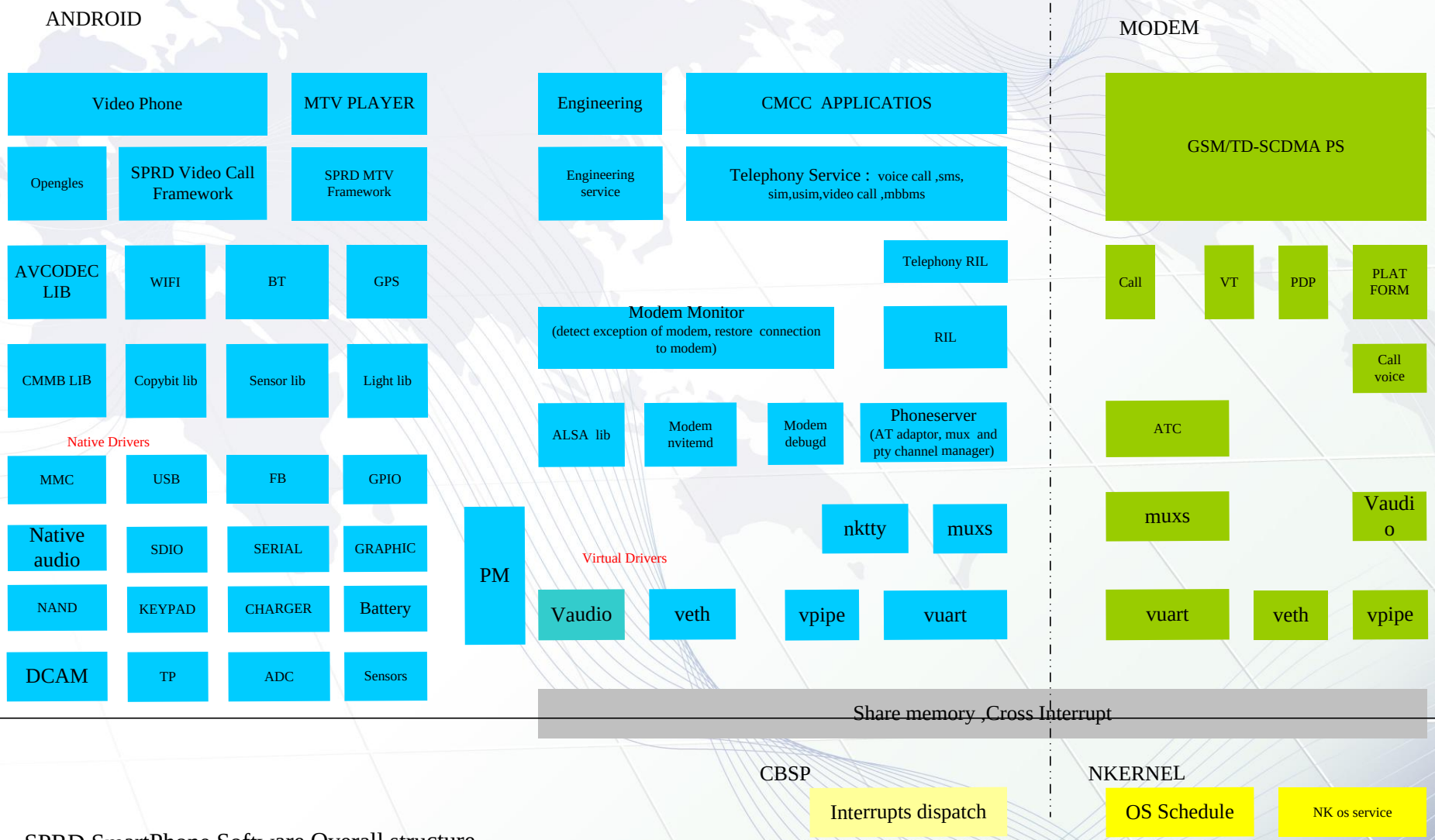
Spreadtrum Realization

- Chip Architecture Improvement
  - Contained MIPS consumption when doing high speed downloading, leaving more resources for Smart Phone OS
  - 2 Independent Timers for OS Ticking
- Strict Hardware Partition
  - Smart Phone OS controls all peripherals, not bridged through RTOS
  - Guest OS side device drivers are identical to their native ones
  - Communications between guest OS are through virtual Ethernet or virtual pipe in memory

The Way Spreadtrum Virtualizing the System

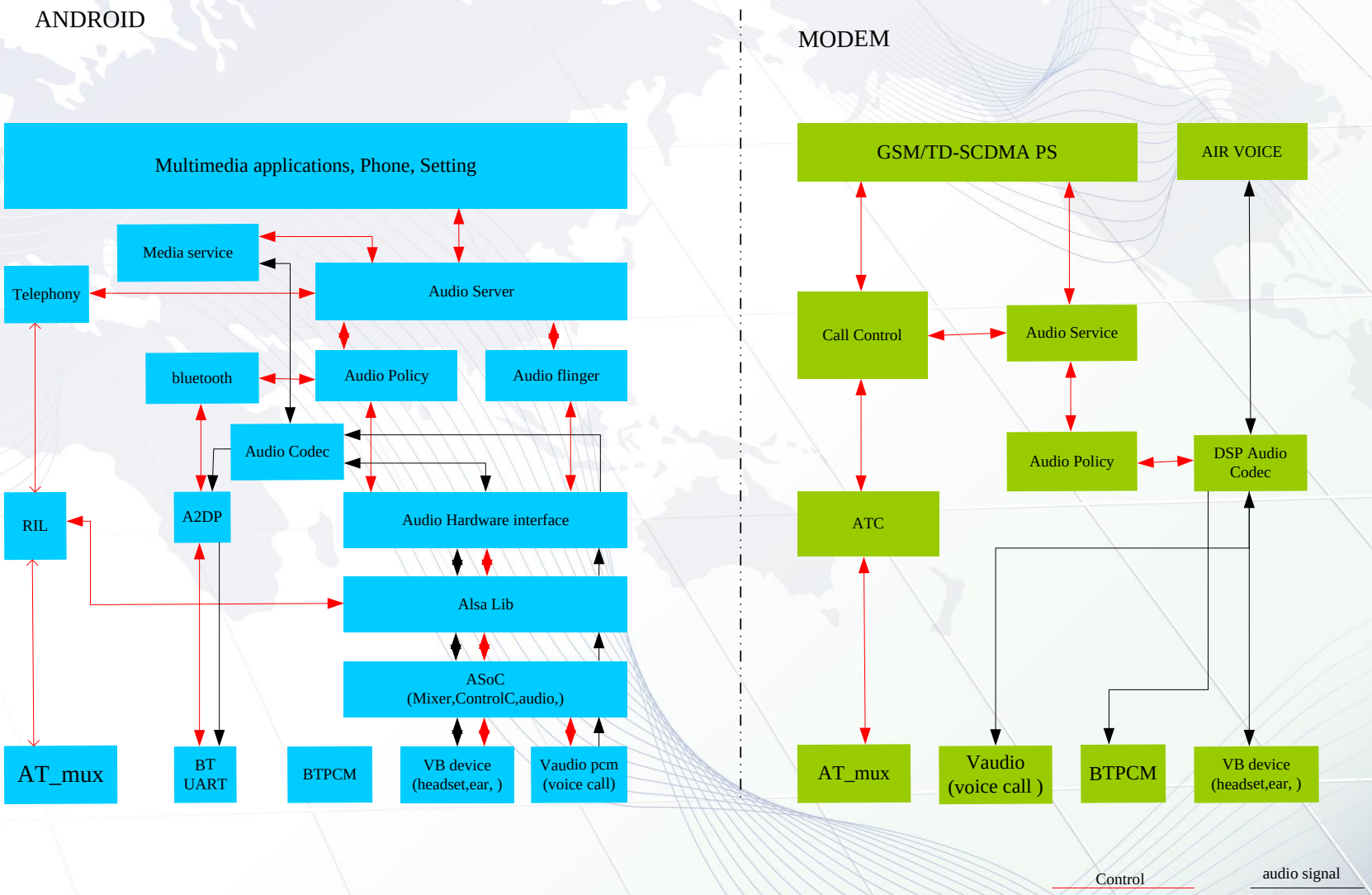


SPRD SmartPhone Software Overall Structure

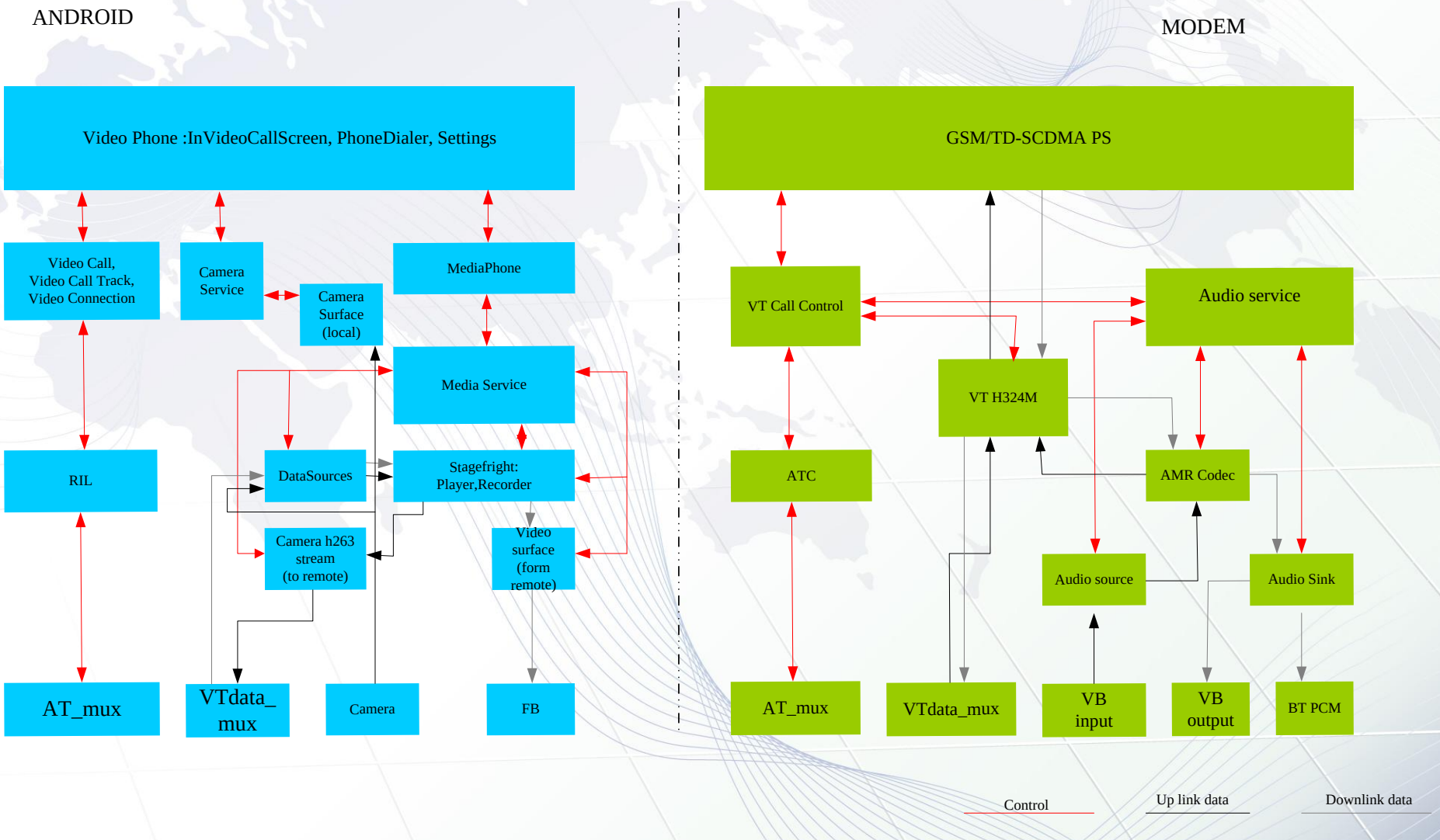


SPRD SmartPhone Software Overall structure

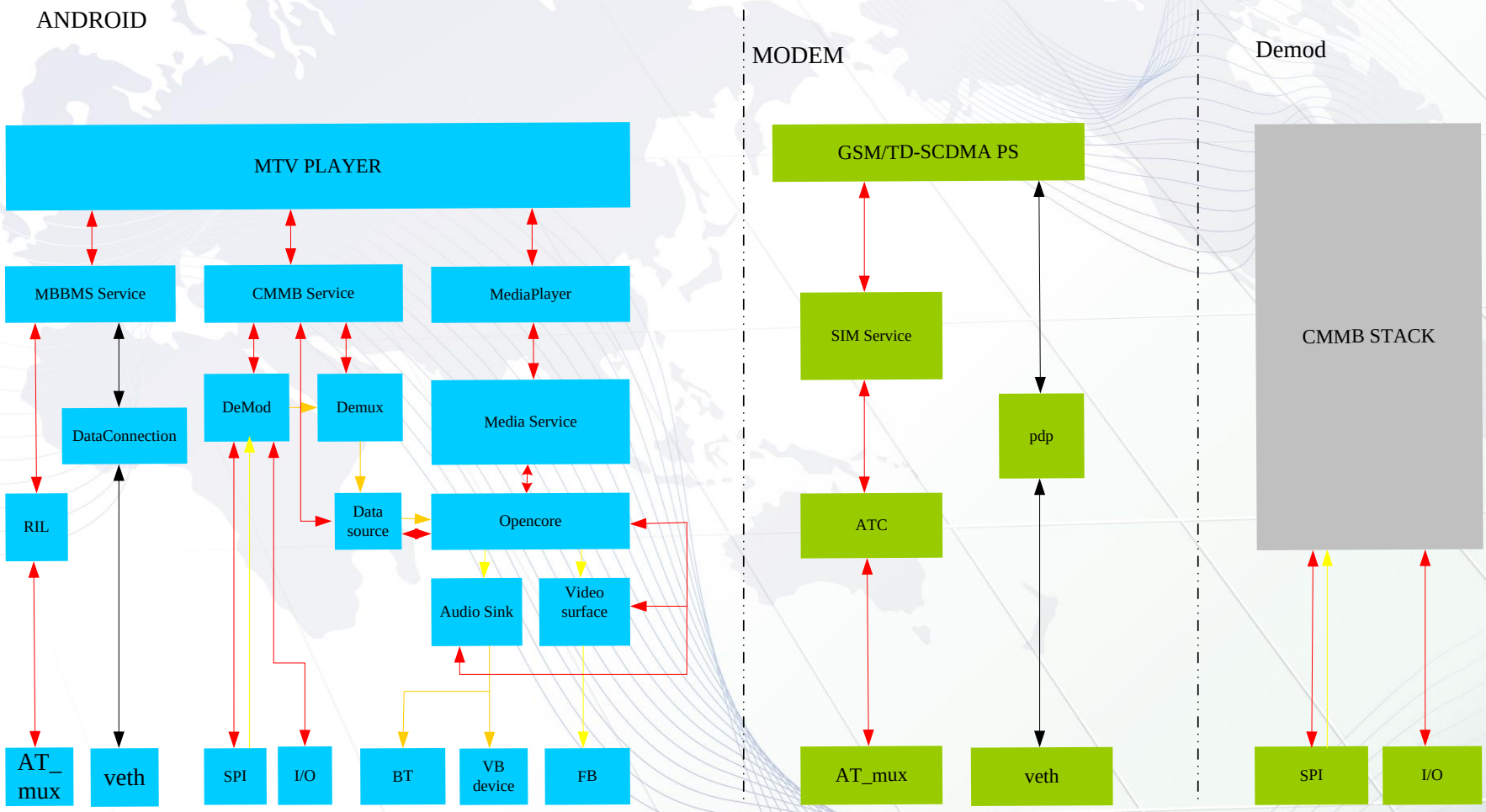
Audio Sharing



Audio Sharing Software Overall Structure



Video Call Software Overall Structure



Mobile TV Software Overall Structure

Power On Process

- Rom code
- Bootloader (u-boot)
- VLX/nkcbsp
- Modem
- Linux/Android

Communication between Modem & Android

SPREADTRUM
Innovation changes the future

- Control: AT/mux/vuart
- Data: veth
- Others (e.g. modem arm log): vbpipe

Development - debugging

SPREADTRUM
Innovation changes the future

- Android/Linux (same to the native solution)
  - kernel log, ftrace, etc.
  - logcat, apanic, anr trace, tomestones, etc.
  - gdb, ddms, etc.
  - application IDE…
- Modem (almost same to the native solution)
  - arm log, assert tool
  - DSP log

Development - compiling

- Android/Linux
  - kernel/u-boot: same to native
  - Android: use Spreadtrum scripts to patch/unpatch 3rd party stuff and trigger Android build (can include kernel/u-boot build in a single run)
- Modem: same to native
- VLX: one single command

Development – summary

- Android
  - native android + Spreadtrum HAL
- Linux kernel
  - VLX paravirtualized kernel + native drivers (for real h/w) + virtual drivers (for IPC)
- u-boot (including 1st bldr, 2nd bldr, 2nd fdl)
  - standard u-boot + drivers + Android support (e.g. boot.img parsing, recovery mode, etc.)
- Modem
  - VLX paravirtualized ThreadX + minor changes to some modules (e.g. vEth based data transferring)

Thanks