Android Automotive OS

Guru Nagarajan, March 25, 2021
Agenda

● Android Automotive OS Platform
● Virtualization
● LTS & Security
● Performance & Reliability
The **Android Operating System**, optimized and extended into a built-in platform for automotive infotainment systems

Rich set of developer tools and SDK to enable application development

Multi-layer Security to protect the user

---

**Android Automotive OS | OS for Automotive Infotainment**

<table>
<thead>
<tr>
<th>Android</th>
<th>Android Automotive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Applications</strong></td>
<td>Car dialer, Car media, Car messaging, Car notification, Car system bar, HVAC, Radio</td>
</tr>
<tr>
<td><strong>Android Framework</strong></td>
<td>Audio policy API, Car sensors, HVAC manager, Global voice trigger API, Multi-User, Multi-Display, Cluster</td>
</tr>
<tr>
<td><strong>Android System Services</strong></td>
<td>Audio focus, Bluetooth stack, Car service, Car UI mode, Vehicle network service, Watchdogs</td>
</tr>
<tr>
<td><strong>Connectivity</strong></td>
<td>Bluetooth - Profiles, Browsable Media Sources, Cover Art, MMS, Wifi - Dynamic Wifi Interface</td>
</tr>
<tr>
<td><strong>System Health &amp; Telemetry</strong></td>
<td>CAN interface, Deep sleep, Multi-profile USB host, Rear-view camera</td>
</tr>
<tr>
<td><strong>HAL</strong></td>
<td>Extended View System / Camera HAL, Vehicle HAL</td>
</tr>
<tr>
<td><strong>Linux Kernel</strong></td>
<td>CAN interface, Low Power, Rear-view camera, Android Common Kernel</td>
</tr>
</tbody>
</table>
Android Automotive OS | Overview of Features

**Android P**
- Basic rotary support
- Multi stream audio routing
- Multiple IP network support
- Bluetooth improvements
- EV API
- ADAS/Maps data integration
- Driving state & UX Restrictions
- Suspend to RAM
- Flash wear management

*And more...*

**Android 10 (Q)**
- Multi-display capability
- Multi-zone audio
- Multi-user support
- Themeable system apps
- Updated system UI
- Remote SIM (SMS via BT)
- Identity mgmt via trusted device
- Multiple UX restriction configs
- Garage mode integration
- Passenger Mode API
- Improvements to VHAL, user media mgmt
- Watchdogs & Reliability

*And more...*

**Android 11 (R)**
- Multi-zone Audio input, Per user Audio Zone, Multi-display
- Early Camera, Surround View
- CVML framework to support execution across discrete HW, Virt env
- Trusted Execution Support
- VirtIO based subsystems
- Metrics
- Vehicle Integration to abstract vehicle bus (ex: CAN)
- Cover Art, MMS (Bluetooth)

*And more...*

---

*Current best assessment of plans. It is possible that the final list of delivered features is different than what is listed here as planning and development continue to unfold.*
Development, Sustaining & Security

- Software in Cars have a lifetime of a decade or more
- To protect the user and ensure user experience, sustaining performance and tooling are critical.
  - Performance
  - Reliability
  - LTS
  - Security
  - Virtualization

Android Automotive OS in Cars

Linux Kernel, Android Phones
● Virtualization
● LTS & Security
● Performance & Reliability
Virtualization
Virtualization

Cockpit domain controllers: An emerging category of car infotainment hardware platform.

- Weight reduction
- Cost reduction
- Improved x-domain integration
- Improved power consumption
- Improved OTA/updates
Virtualization | Automotive Drivers

- Cockpit consolidation is a real trend: Lower BOM cost, lower vehicle weight
- Ever more powerful SoCs enable more use cases
- Some use cases are critical (for example, interrupt response guarantees, memory isolation)
- Shared hardware access imposes unique challenges (for example, which network does one Wi-Fi adapter connect to?)
Our Approach to Virtualization

● Standards based and Open Platform
  ○ VirtIO, Open standard for virtualized devices
    ■ Started in 2008, is maintained and improved by an open committee
    ■ Google is a member of the OASIS committee
  ○ Portability across implementations, Easier Updates
● Leverage virtio where possible
● Extend it as needed (for example, virtio-snd, virtio-scmi)
● HAL virtualization where it makes sense (Vehicle HAL)
● Passthrough for Android-only devices (for example, connectivity)
Virtualization Architecture

**Linux host (as example)**

- **Userspace**
- **Kernel**
  - **Device driver**
  - `*.ko`
- **Physical HW**

**Android Automotive OS Guest**

- **Userspace**
- **Kernel**
  - **HAL**
  - **VirtIO driver**
- **Hypervisor**
- **SOC**
Virtualization | VirtIO Devices

- VirtIO has origins in the cloud/desktop world
- Supports common devices such as disk (\texttt{virtio-blk}), network (\texttt{virtio-net}), and random number generators (\texttt{virtio-rng}) among others
- Multimedia device support is under active development:
  - \texttt{virtio-snd} (new standard in v1.2)
  - \texttt{virtio-gpu} (standard + extensions)
  - \texttt{virtio-video} (WiP in v1.2)
  - \texttt{virtio-scsi} (WiP in v1.2)
- Automotive use cases are a key driver for these new specifications
Virtualization | Vehicle HAL

- Android can work without direct access to vehicle bus
- Host-side runs a HAL server and communicates to Android through `vsock`
- Vehicle HAL really does two things:
  - Management of property subscriptions and overall state
  - Communication to and from vehicle
    - Discover properties configuration
    - Receiving updated property value
    - Set property value
- Only the latter needs to change for virtualization
Virtualization | GPU

- **virtio-gpu** is sufficient for many use cases
- Performance is key. SOCs can have optimized paths for guest VMs (for example, dedicated command queue). Focus on providing performance semantics, but providing a standardized protocol.
- Plan to enable vendor extensions; allow additional virtqueues to be negotiated for vendor-specific commands
Virtualization | Security

- Goal is to integrate Arm TrustZone
- Requires vendor / hypervisor support
- Planned for mid-2021
LTS & Security
LTS Background

- A Long Term Stable (LTS) kernel is a version of the upstream Linux kernel that is maintained for an extended period of time (versions selected for use with Android receive 6 years of support).
- Security and functional fixes are regularly checked in.
- Android and other major Linux distributions (e.g., Ubuntu, Debian, Red Hat) typically base their releases on a Linux LTS kernel in order to ensure updates and support for the lifetime of the product.
What happens today?

- Many security vulnerabilities that are fixed on upstream Linux are not fixed until much later on Android, putting users and the Android brand at risk.
- The Android Security team makes a best effort to identify fixes that address security vulnerabilities and to require them for Security Patch Level (SPL) compliance in the monthly security bulletins.
- However, we are limited to issues that are explicitly flagged as security vulnerabilities or that researchers bring to our attention as security vulnerabilities affecting Android.
- An analysis in 2019 showed that 92% of Linux kernel security vulnerabilities that are required for SPL compliance were already fixed in the LTS kernel at the time they were identified as security vulnerabilities.
SoC ecosystem & LTS updates

- Starting with Android 9, new device launches are required to ship with the most recent LTS release.
- SoCs update the kernel to the required LTS version to support new device launches.
- Android Common Kernels are updated regularly with the latest LTS kernel and tested/verified on all hardware and virtual platforms associated with them.
- SoC partners acknowledge the merge of the latest ACK and report any issues they encounter and are provided support by the Android kernel team.
- LTS update requirements are published in the partner security bulletin after we have confirmed with SoCs that the LTS version is merged in and tested.
Performance & Reliability
Performance

- Performance is key
- Challenges
  - User needs are evolving, new use cases are pushing the boundaries
  - Benchmarks are not always representative of real-world interactions
  - Creation of a representative use cases that can be used to evaluate our priorities are a start
- Need to prepare early for “future killer apps”, ensure headroom
Performance & Reliability | Building Blocks

- Throughput and Latency both are critical in Cars
- Power and Energy Consumption, as in mobile are critical
- Standardize on the counters, tools, and HALs

Data Sources
- Linux kernel ftrace
- Android event log
- Android Power /sys/power
- Linux /proc/vmstat
- Linux /proc/vmstat

Data Collection
- Android collection agents
- C++ Native process

Trace Output
- FTrace events
- VMS Stats
- CPU Stats
- Clock Domain Sync
- Power Stats

Android

Google
Performance & Reliability | Watchdog

- Android Automotive OS introduced a Watchdog (CarWatchdog) for ensuring reliability.
- CarWatchdog is a service that monitors system health and identifies/terminates badly behaving processes.
- Monitors I/O performance at boot time, at periodic intervals, or at a custom duration.
- CarWatchdog is different from activity lifecycle monitoring for detecting Android Application Not Responding (ANR) - native services and Android services are the clients.
- Facilities for managing restarts and process control are provided.
Thank you!