ARCHITECTURE & SAFETY-CRITICAL SOFTWARE FOR NEXT GENERATION VEHICLES

DR RICHARD WEST
CHIEF SOFTWARE ARCHITECT
DRAKO MOTORS

PROFESSOR
BOSTON UNIVERSITY
VEHICLE GROWTH IN ELECTRONICS

- Electric vehicles, ADAS, IVI, V2X driving up cost and complexity of electronics
  
- Modern luxury vehicles have 50-150 ECUs
  source: Strategy Analytics, IHS Markit

- Global ECU market $63.6 billion (2018)
  source: grandriewresearch.com

- Electronic share of total vehicle cost is rising exponentially

How do we reverse the trend?
AUTOMOTIVE SOFTWARE COMPLEXITY

- Growth in automotive electronics has given rise to growth in software complexity

Source: https://informationisbeautiful.net/visualizations/million-lines-of-code/
SOFTWARE EXPLOSION

- Software growth driven by increased vehicle functionality + increased ECU count
## HARDWARE & OS EVOLUTION

### AUTOMOTIVE DOMAIN
- 8→16→32 bit microcontrollers
- 1-3 cores, often single function
- Typically 10s-100s MHz
- Freescale PowerPC, Infineon TriCore ...
- Integrated CAN, GPIOs, ADCs

Simple RTOS
- OSEK, FreeRTOS, Tresos, ECOS ...

### PC DOMAIN
- 64-bit CPUs, integrated GPUs
- Multicore, multiple tasks
- GHz clock speed, hardware virtualization
- Intel & AMD x86, ARM
- USB, PCIe, Ethernet, WiFi

Complex General Purpose OS
- Windows, Mac OS, Linux
AUTOMOTIVE SYSTEM CHALLENGES

Reduce electronic costs
- Replace ECUs with multicore PC-class processors
- Consolidate ECU functions as software tasks

...but...

Need new vehicle OS
- To manage 100s of tasks on multiple cores
- Too complex to write new OS from scratch
- Combine real-time with legacy code
- Safety (ISO26262), security, predictability
- Mixed-criticality-aware (ASIL A-D)
- Fast critical reboot (current PC-based OSes too slow)
MOVING FORWARD: DriveOS
DRAKO DriveOS

DriveOS supports traditional hardware functions as software tasks running on a multicore virtualized platform.

- **Software easier to reconfigure, upgrade, and extend**
- **Machine virtualization provides safe isolation of cores**
- **Hardware & software redundancy for fault recovery**
DRAKO DriveOS I/O

USB-centric solution: works with legacy devices + supports higher bandwidth future needs

*Secure access to USB + CAN mediated by trusted I/O sandbox in DriveOS
REFERENCE DESIGN: DRAKO GTE DriveOS

- Thermal Control
- Powertrain Control
- Infotainment
- Battery Management
- Cloud Connectivity
- Drako DriveOS Intel PC
- ADAS
- Suspension Control
- Instrument Cluster
- Body Control
- Fast Charging
# DRAKO DriveOS REFERENCE STACK

<table>
<thead>
<tr>
<th>Cloud Services Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Secure V2X Communication Layer</td>
</tr>
<tr>
<td>Powertrain &amp; I/O Services</td>
</tr>
<tr>
<td>USB Bus Scheduler</td>
</tr>
<tr>
<td>Real Time Device I/O</td>
</tr>
</tbody>
</table>

**Real Time Secure Shared Memory Communication**

- Secure Separation Kernel (Quest-V)

**Hardware Layer**: Multi-Core PC (Intel x86)
EXAMPLE: Quest-V for DriveOS

- Separation kernel (a.k.a. partitioning hypervisor)
- Partitions CPU cores, RAM, I/O devices among guests
- Each sandbox runs its own OS
DRIVEOS: EXAMPLE OpenPilot ADAS + IC + IVI

- **CAN Gateway**
  - `libusb + canlib`

- **Infotainment CAN Mapper (IMS)**
  - ADAS CAN Mapper (AMS)

- **Longitudinal Feed-forward PI Controller**

- **Infotainment CAN Mapper (IML)**

- **ADAS CAN Mapper (AML)**

- **FIFO**
  - `cereal pub-sub`

- **shmcomm channels**

1. **IC and IVI Sensing**
2. **HVAC Actuation**
3. **ADAS Sensing**
4. **ADAS Actuation**
5. **Control Command**
6. **INIT, RESET Input**
7. **UPDATE Input**
8. **UPDATE Output**

- **Quest RTOS**
- **DriveOS on DX1100**
- **IC**
- **IVI**
- **OpenPilot ADAS**
- **Yocto Linux**
DriveOS: OpenPilot CONTROL LOOP LATENCY

- ADAS Control Loop End-to-end Latency in presence of background Linux tasks
  
  Target bound = 10ms
Jumpstart POWER MANAGEMENT

- PC hardware requires Firmware POST, bootloader, device & service initialization to boot OS
- DriveOS uses Jumpstart ACPI S3 suspend-to-RAM & resume-from-RAM for low latency restart of critical tasks (e.g., CAN gateway services)
Jumpstart POWER MANAGEMENT

- Jumpstart services span all guests
  - RTOS coordinates suspension but enables parallel reboot
- Potential for ACPI S4 suspend-to-disk using non-volatile memory (e.g., Intel Optane)
  - Eliminates system power usage during suspension

Jumpstart reduces DriveOS boot delay
Better than standalone Linux!
CONCLUSIONS

Now is the time to look to alternative hardware + OS automotive solutions

DriveOS uses hardware virtualization for real time temporal and spatial isolation of software functions

+ Multicore PC-class platform replaces ECUs with software tasks
+ Symbiosis between RTOS & legacy OS
+ Real-time I/O & task pipeline processing
+ Fast reboot of critical services on PC-class hardware