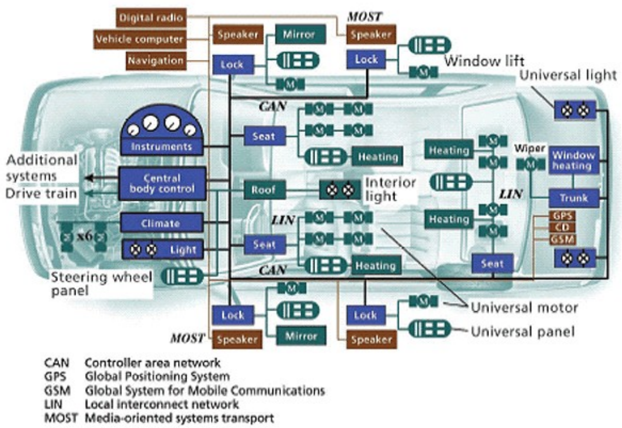
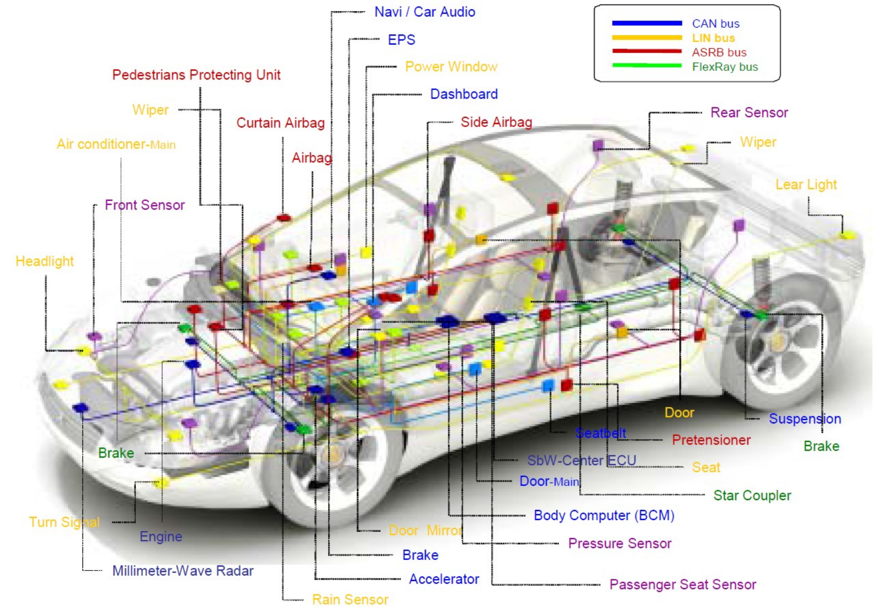
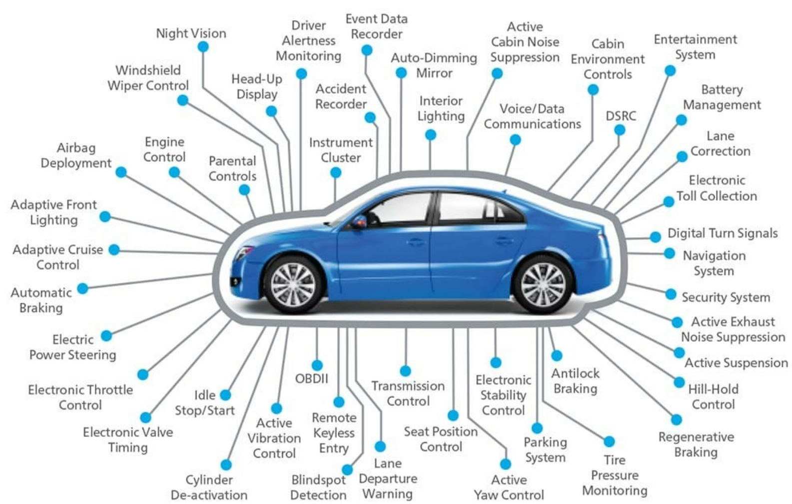
Towards an Integrated Vehicle Management System in DriveOS

Soham Sinha and Richard West
Department of Computer Science, Boston University and Drako Motors Inc., USA

Modern Automotive Systems

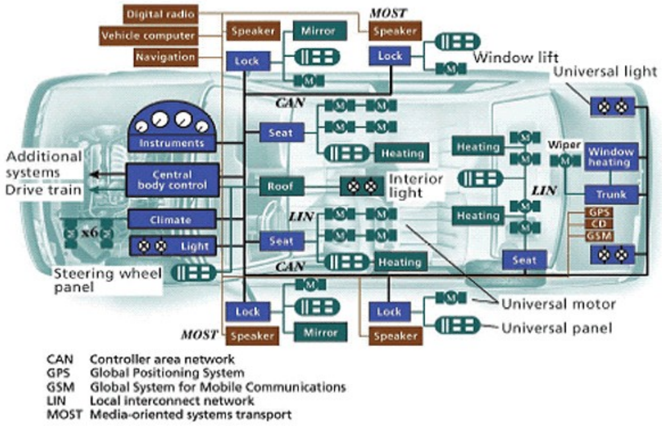






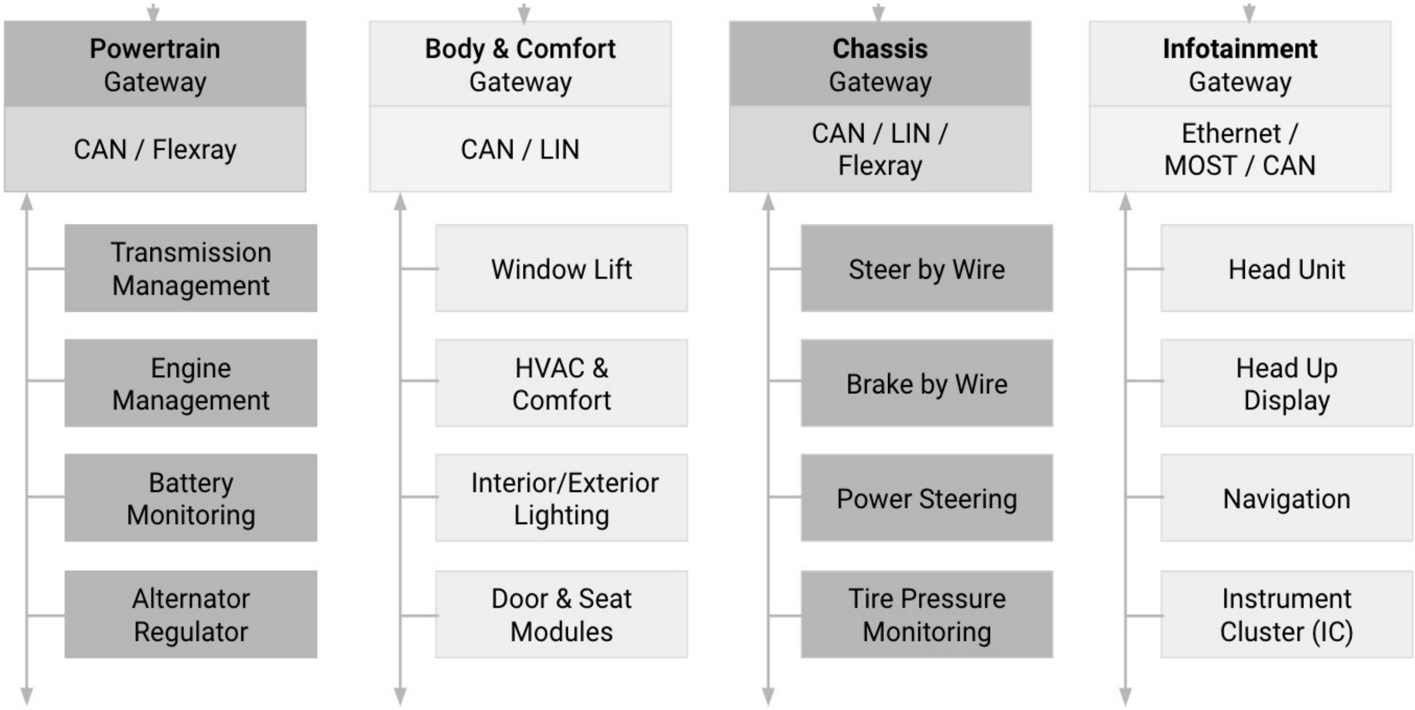
Modern Automotive Systems

- Today‘s vehicles have 50 – 150 Electronic Control Units (ECUs)
- 10s to 100s of millions lines of code
- Complex CAN bus network of ECUs



Functional Domains in a Vehicle

- Vehicle software services are divided into a number of functional domains.



New Functions → New Functional Domains

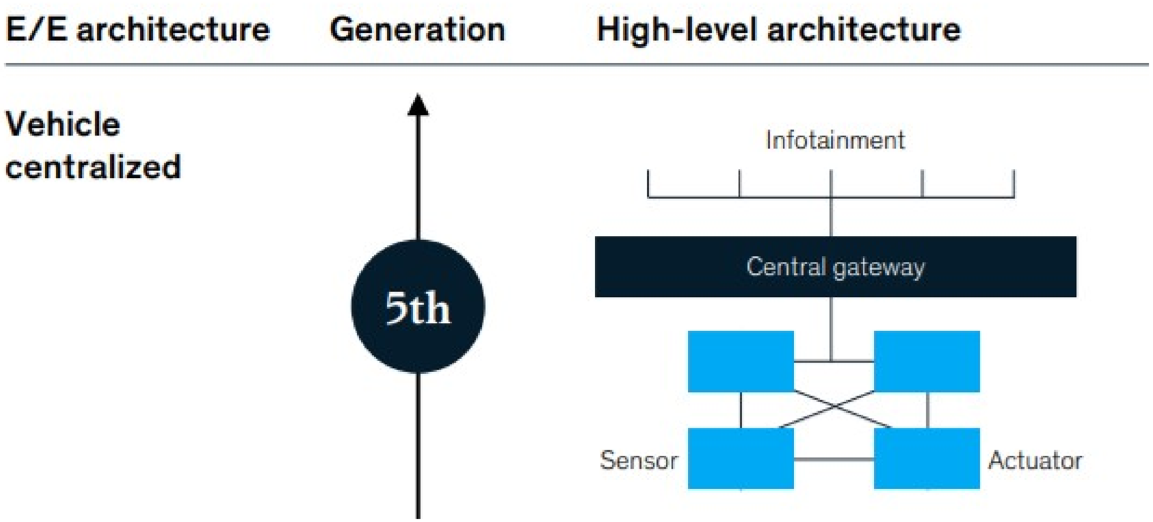
- New vehicle functions like ADAS, high-quality IVI add new functional domains.
- New domains mean more electronics and more software.

New Functions → New Functional Domains

- New vehicle functions like ADAS, high-quality IVI add new functional domains.
- New domains mean more electronics and more software.

Hardware, Wiring and Packaging, Upgradability Cost ↑

Functional Consolidation



Requirements for a Centralized Vehicle OS

- Timing-predictability
- Critical and non-critical tasks in the same platform
  - e.g., ADAS, IVI and IC
- Safety, Security and Reliability
- Preferably Low-cost Computing Hardware

DriveOS

DriveOS

Centralized Vehicle Management System

Timing-predictable

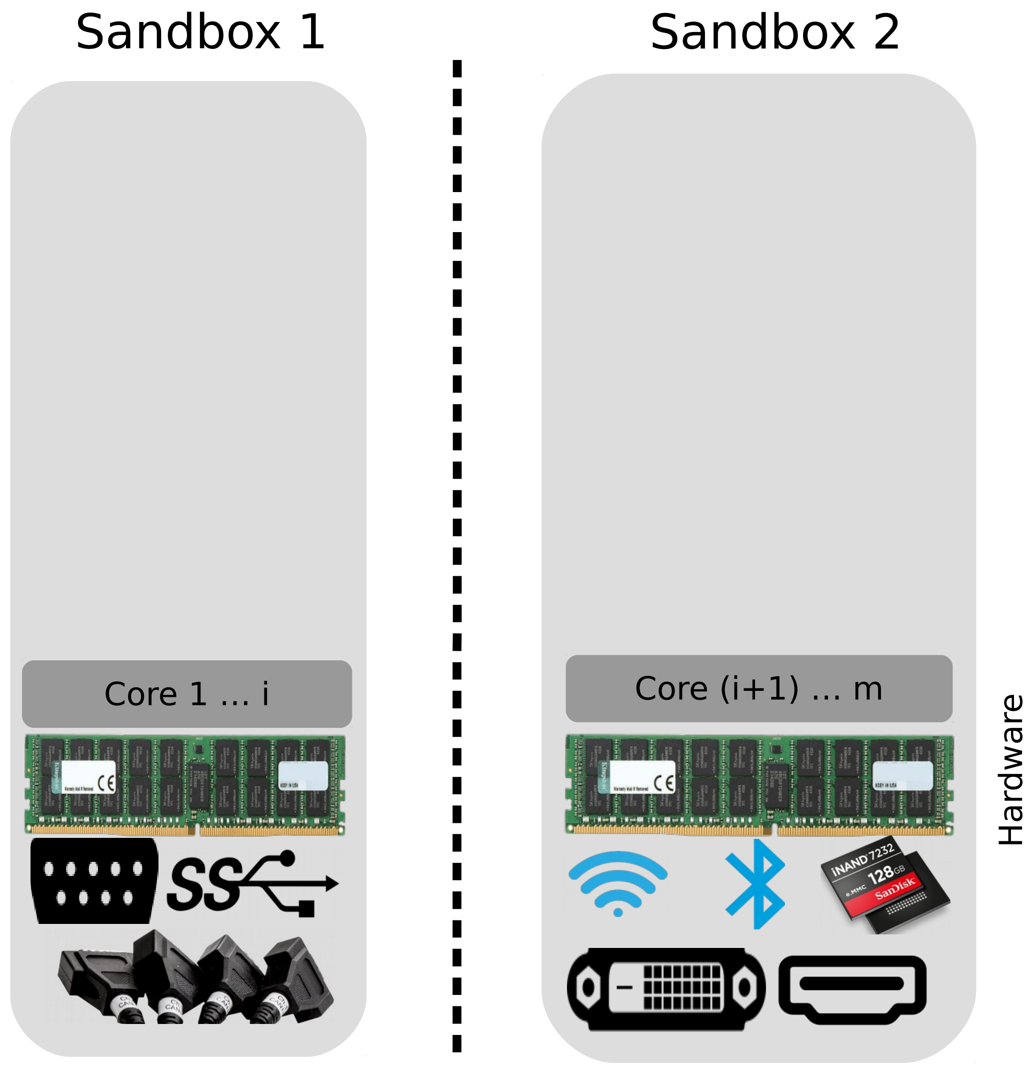
x86 Secure Hardware Virtualization

Runs on PC-class hardware

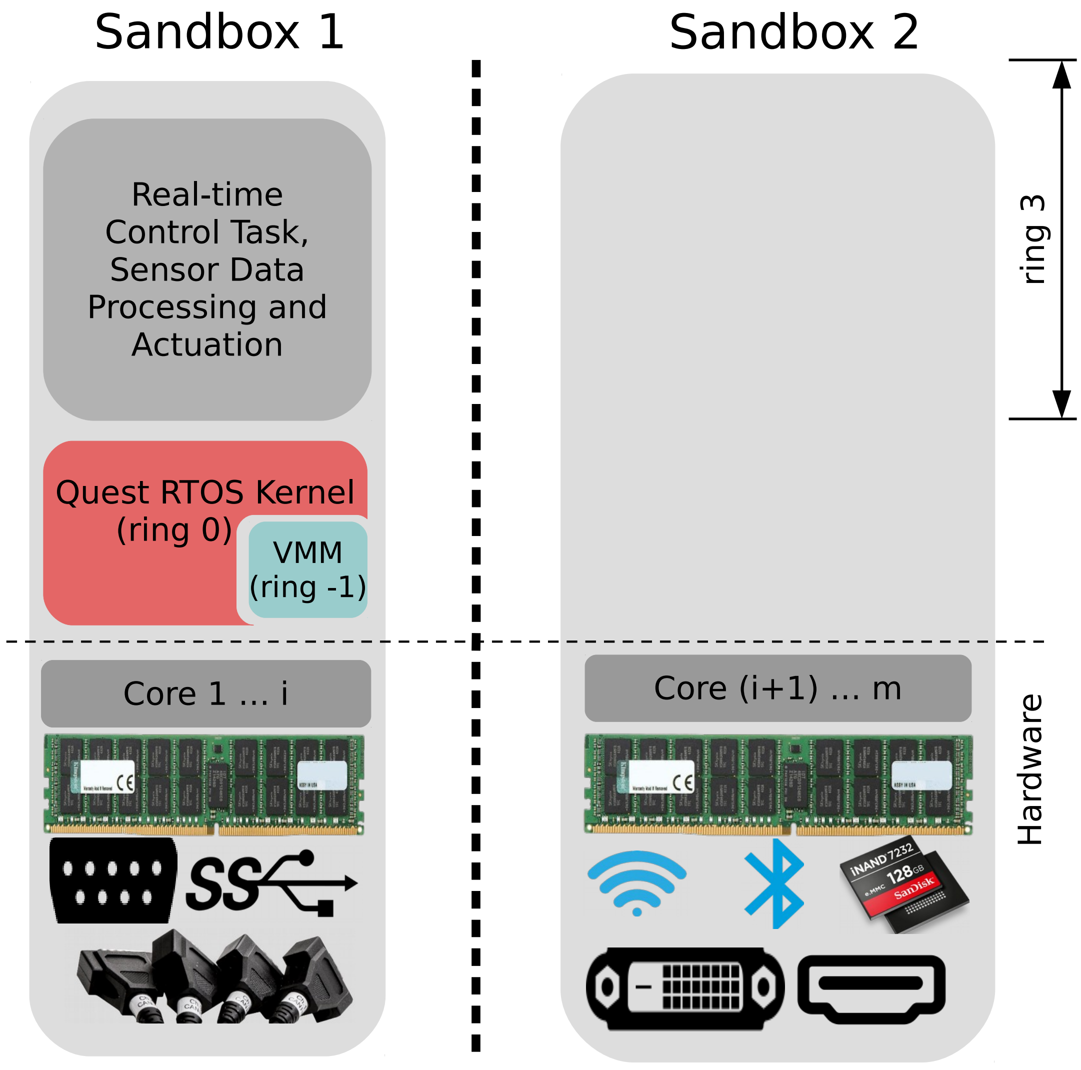
DriveOS

Target Platform : DX1100

DriveOS Design: Quest-V Separation Kernel

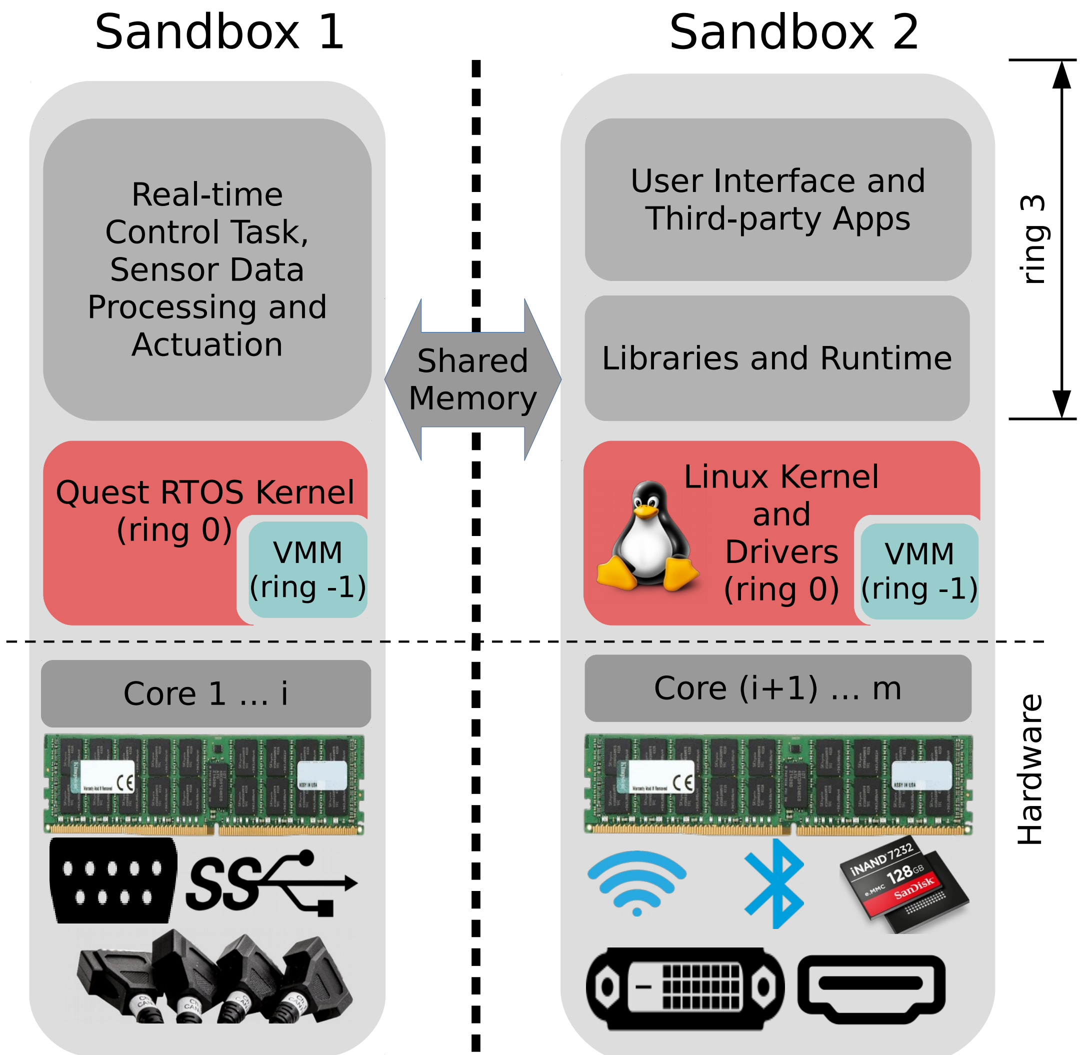


DriveOS Design: RTOS Sandbox

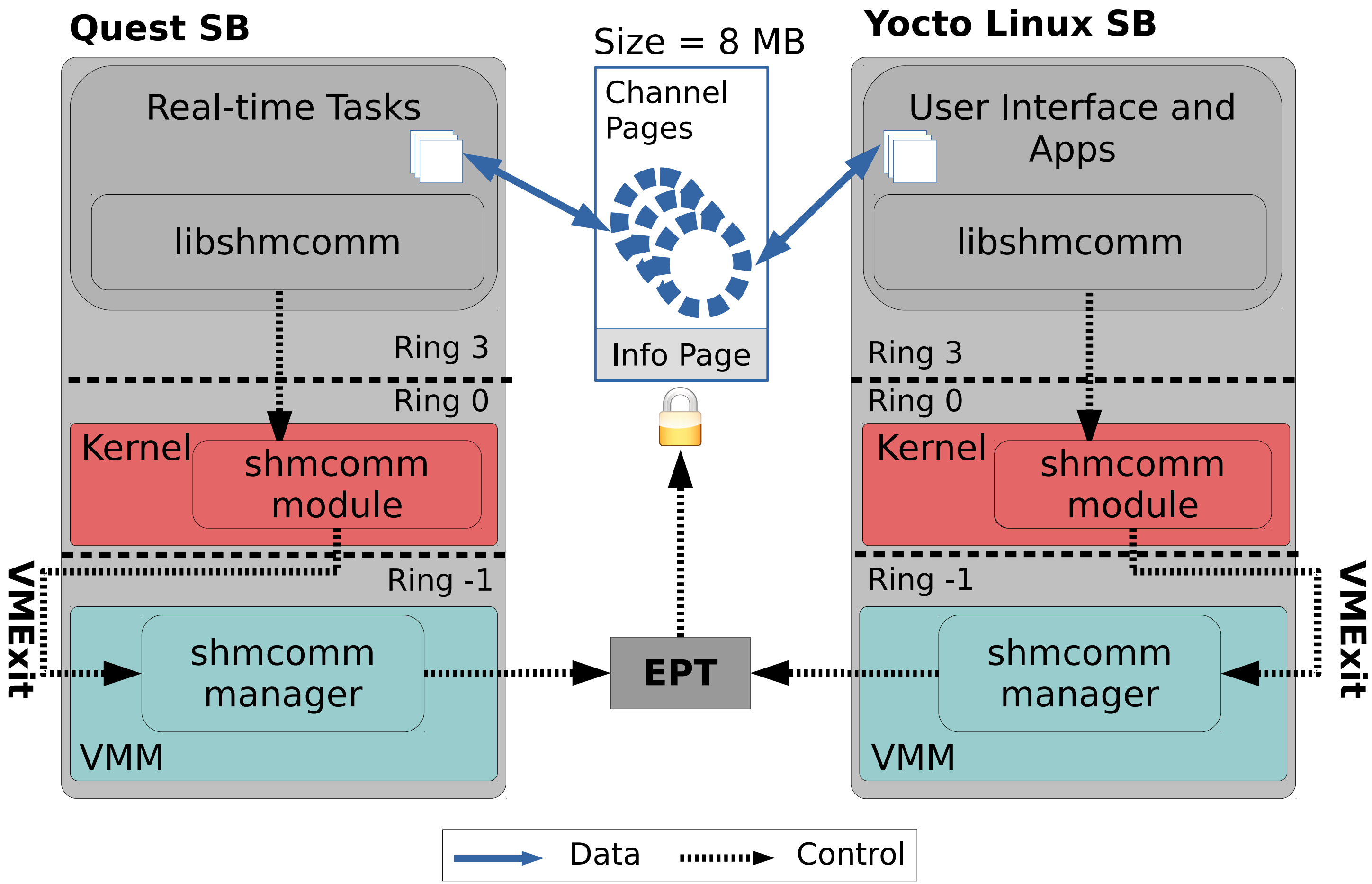


BU DRAKO

DriveOS Design: Linux Sandbox



DriveOS Inter-sandbox Communication: `shmcomm`



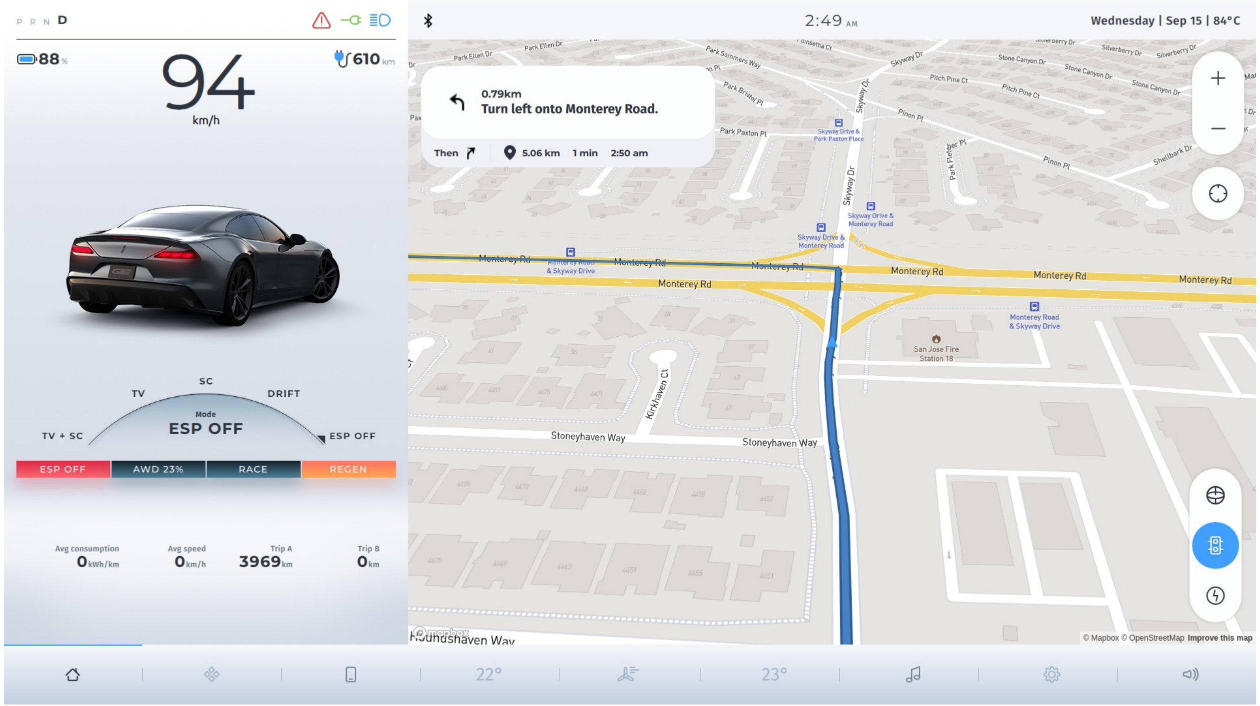
BU DRAKO

DriveOS Applications

- Instrument Cluster (IC)

DriveOS Applications

- Instrument Cluster (IC)
- In-vehicle Infotainment (IVI)



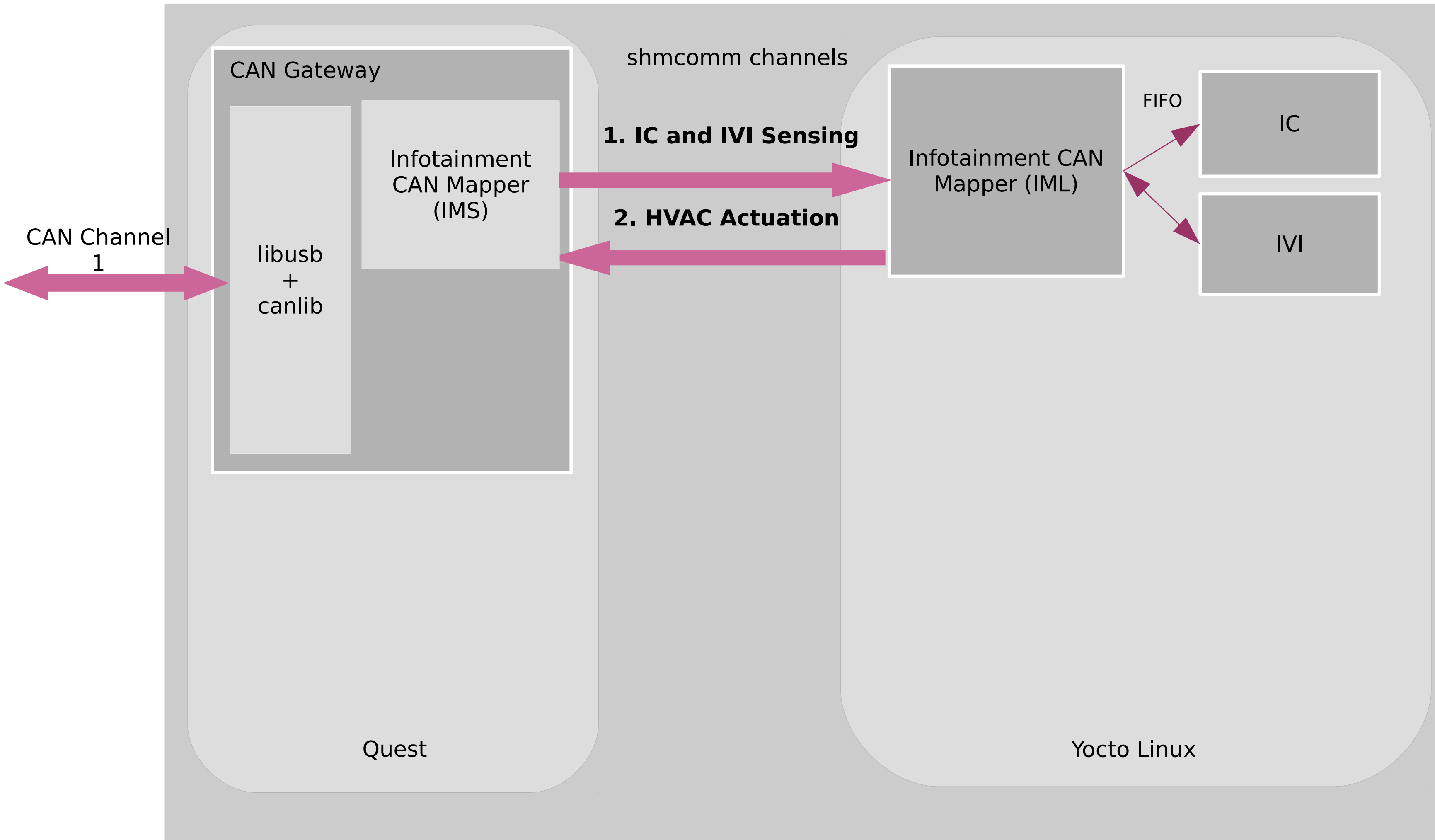
BU DRAKO

DriveOS Applications

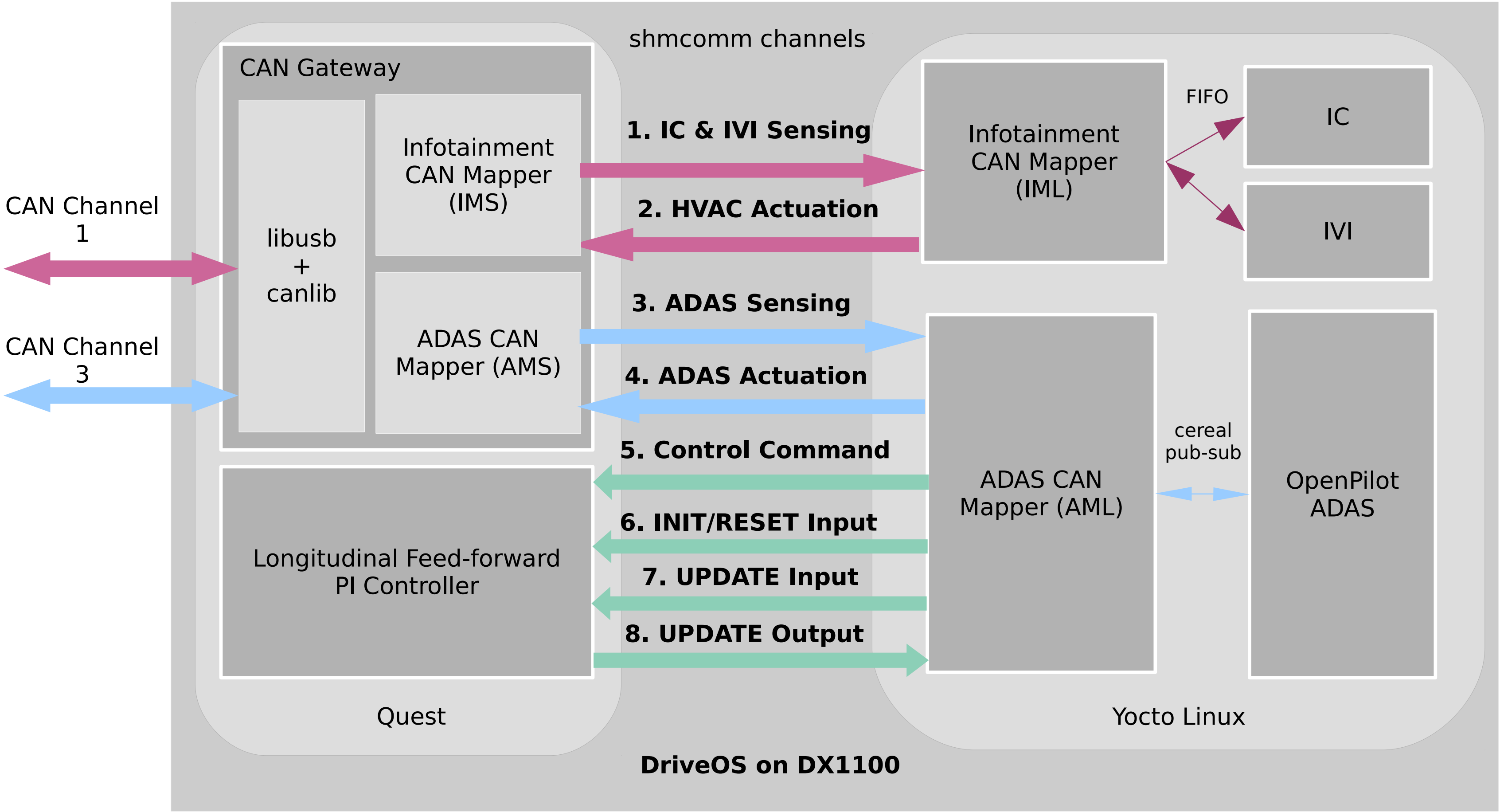
- Instrument Cluster (IC)
- In-vehicle Infotainment (IVI)
- OpenPilot Advanced Driver Assistance System (ADAS)

BU DRAKO

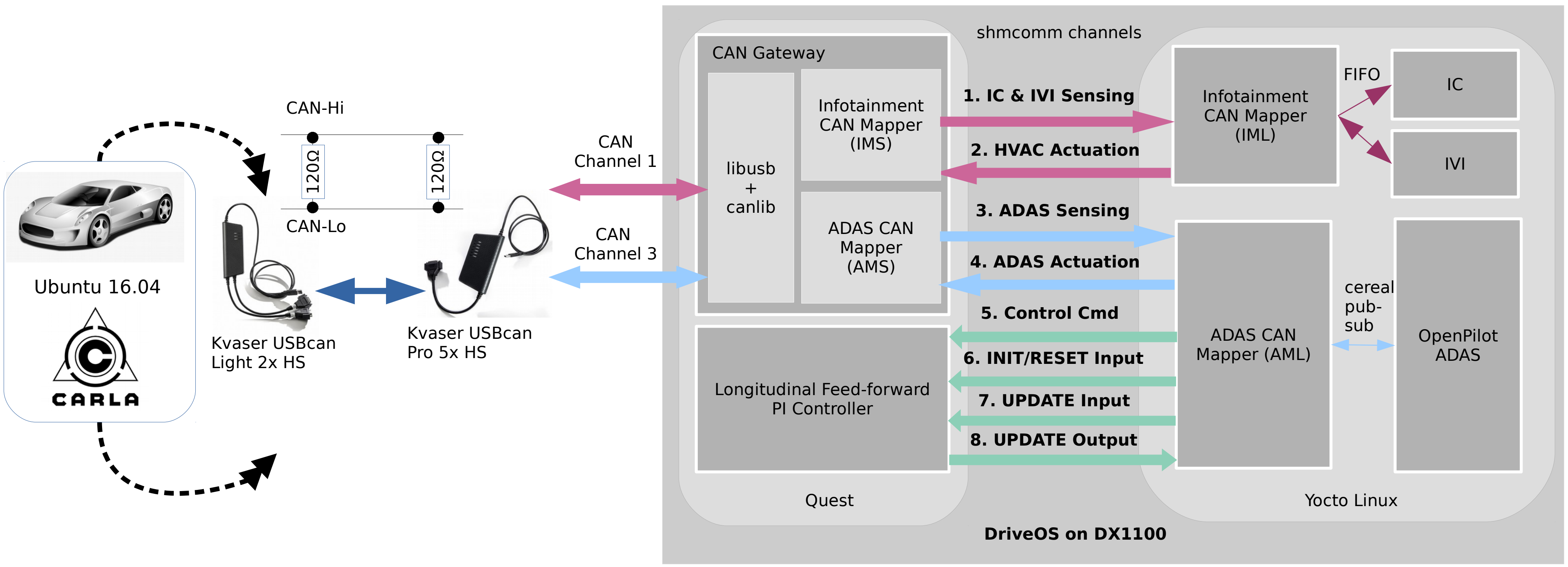
DriveOS Applications: IC and IVI



DriveOS Applications: OpenPilot ADAS



Hardware-in-the-loop Simulation

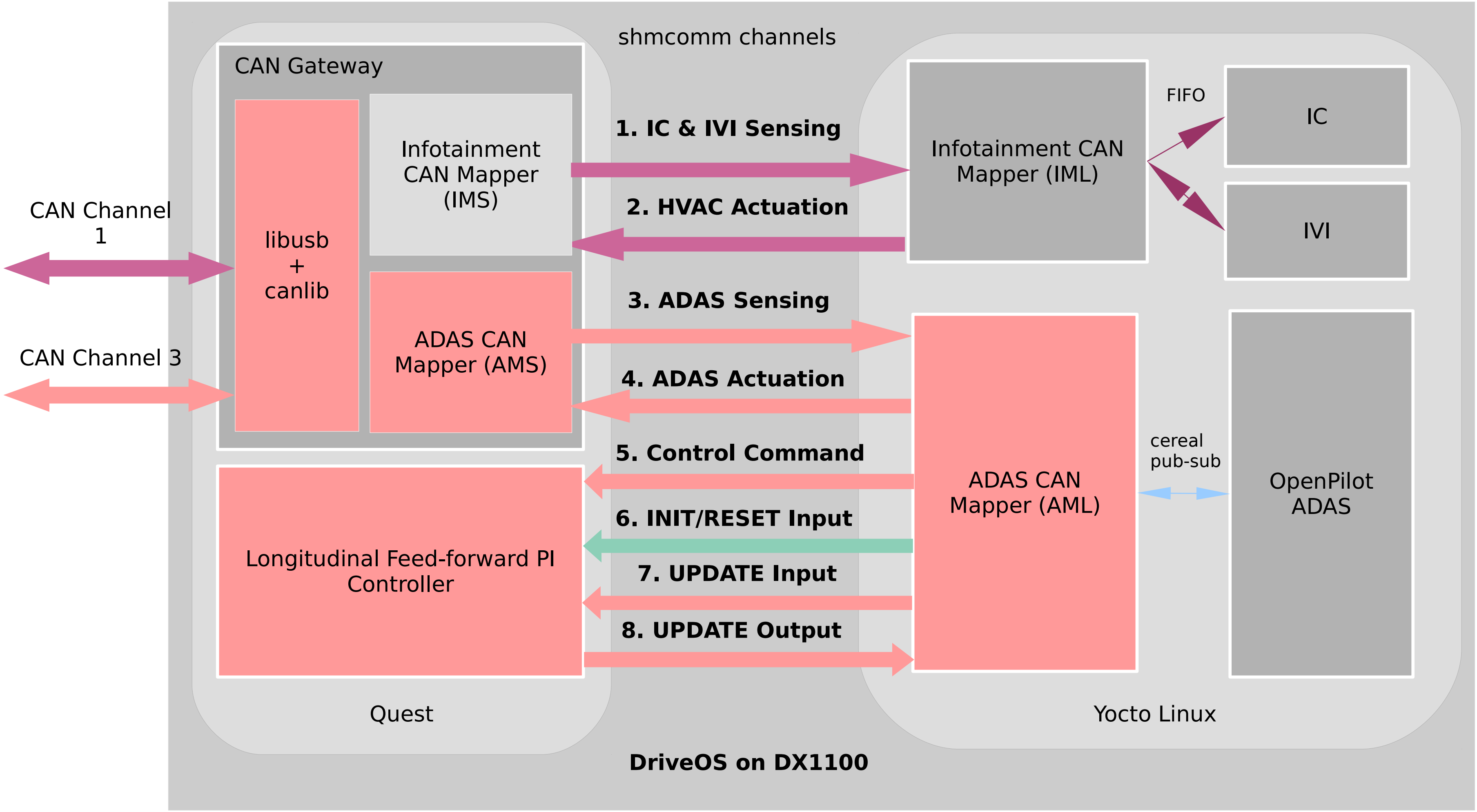


BU DRAKO

Evaluation Results

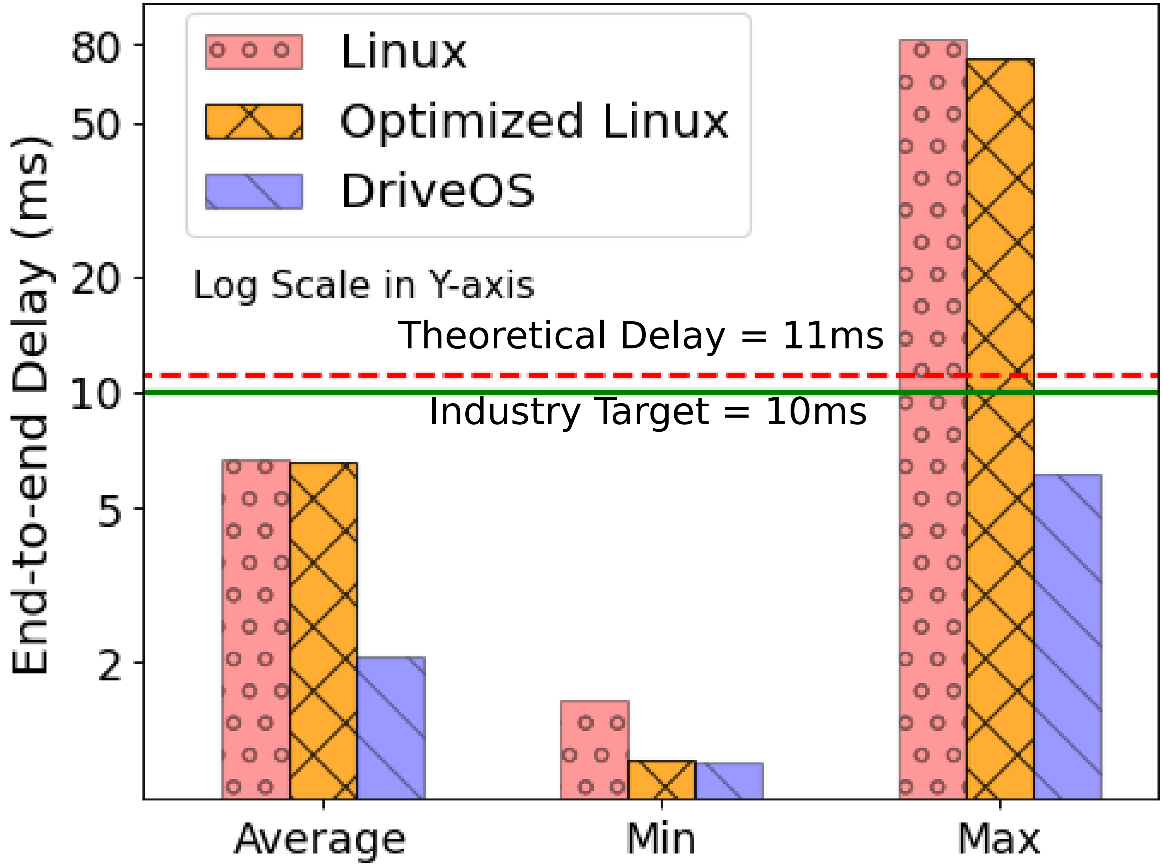
Evaluation: End-to-end Latency

- ADAS Control Loop path is highlighted in pink



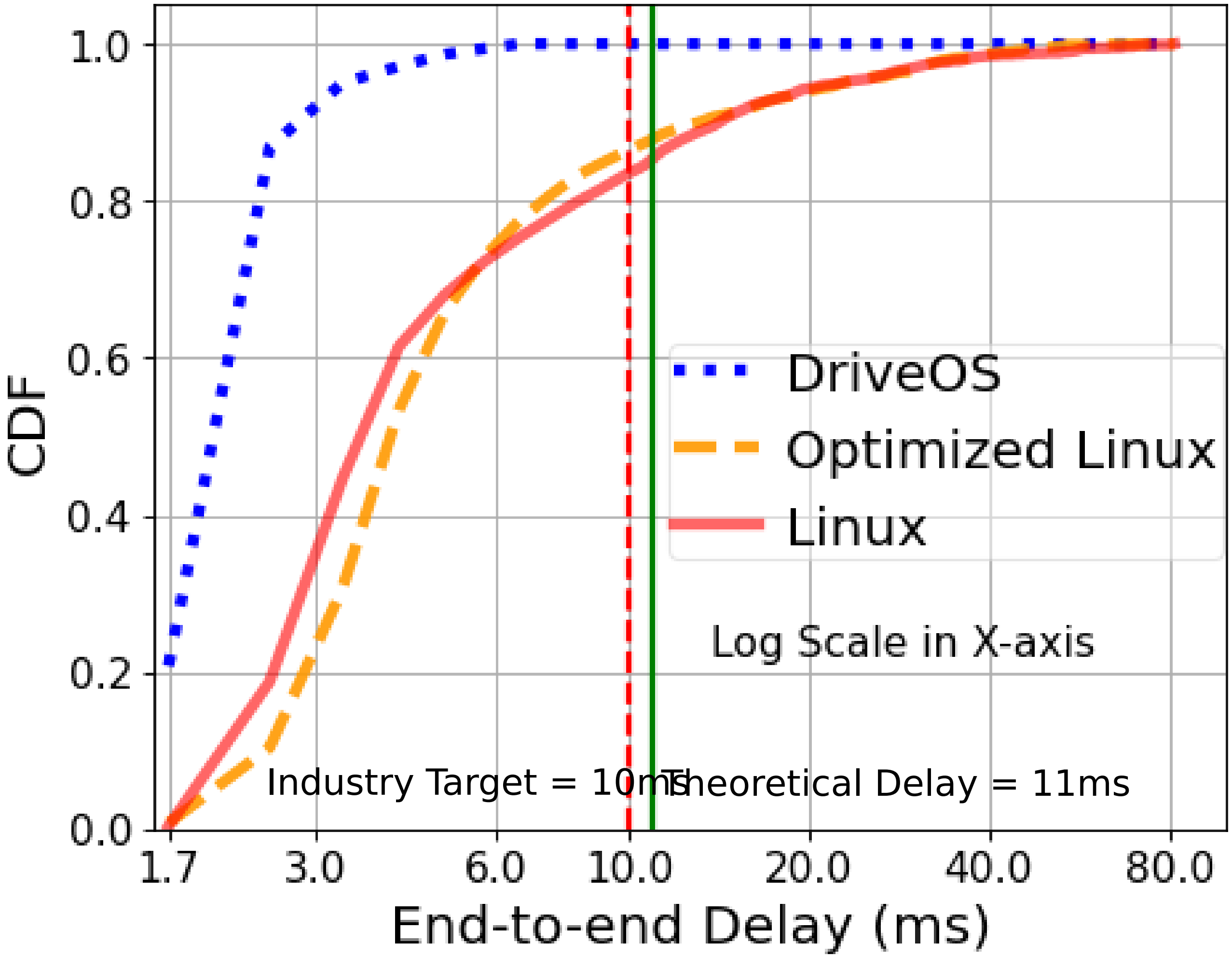
Evaluation: End-to-end Latency

- ADAS Control Loop End-to-end Latency in presence of background Linux tasks.



Evaluation: End-to-end Latency - CDF

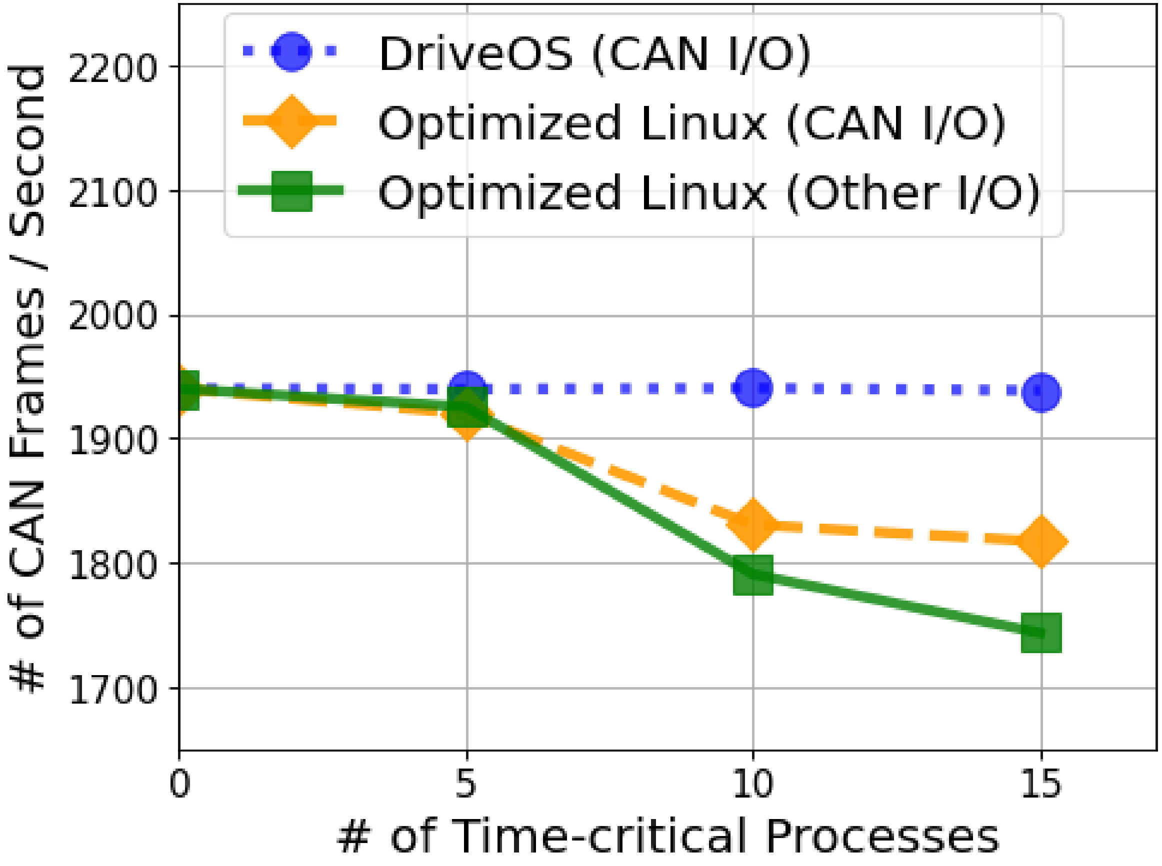
- ADAS Control Loop End-to-end Latency in presence of background Linux tasks.



BU DRAKO

Evaluation: Throughput

- CAN I/O: Other CAN Channels are being accessed
- Other I/O: Random disk and network I/Os are being issued



BU DRAKO

Ongoing Work

- More core vehicle functions (e.g., HVAC, Powertrain) in DriveOS
- Implement a MATLAB/Simulink interface for DriveOS
- Intelligent Power Management
  - Fast suspend/resume functionality

Conclusion

- DriveOS is a centralized vehicle-management system.
- It runs on low-cost PC-class machines.
- It is a timing-predictable, extensible and secure system.
- For this paper, DriveOS focuses on IC, IVI and ADAS.
- It meets throughput and latency requirements for the industry.

Acknowledgements

- Big thanks to the whole Celenum and Drako Motors team!
- Thanks to all the past and current Quest(-V) developers including but not limited to Matt, Ying, Tom, Chris, Eric and Sasan!

Thank you!

Questions?

Contact: <soham1, richwest>@bu.edu

References

- Images are from Google Images, Statista
- Few slides and images are taken from our previous presentations at HotMobile 2020, BEVA 2020 and other conferences.
- Our previous work:
  - A Paravirtualized Android for Next Generation Interactive Automotive Systems (HotMobile 2020)
  - Boomerang: Real-Time I/O Meets Legacy Systems (RTAS 2020)