From Quest to DriveOS: A Journey in Space and Time

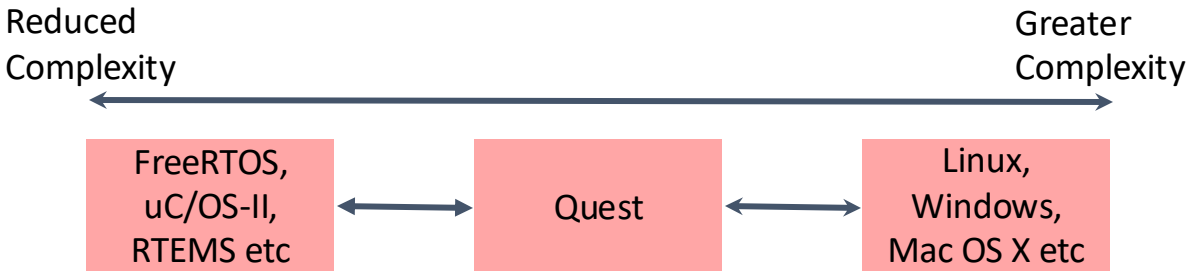
Dr. Richard West
Professor, Boston University
Chief Software Architect, Drako Tech
richwest@bu.edu

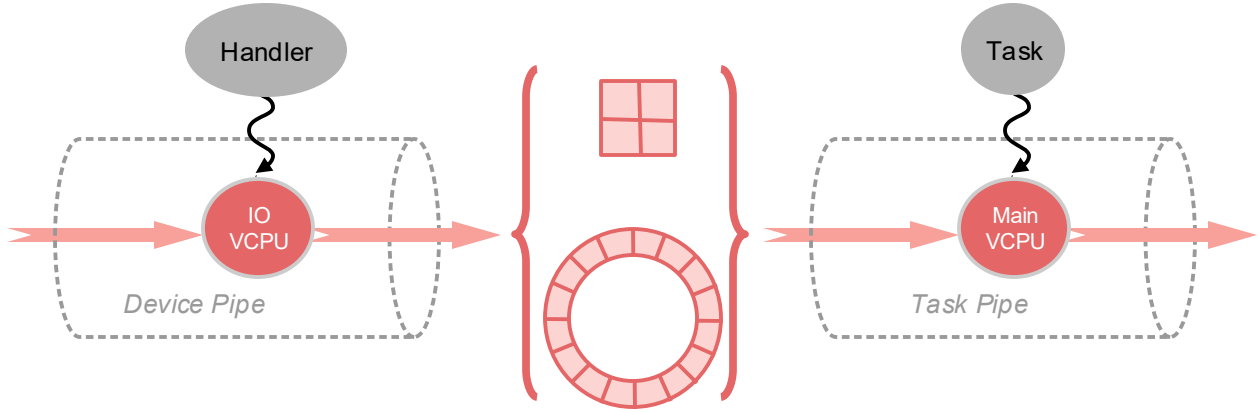


NSF CPS 20th Anniversary Workshop Oct 21-22, 2026

In the Beginning...The Quest RTOS [2005-Present]

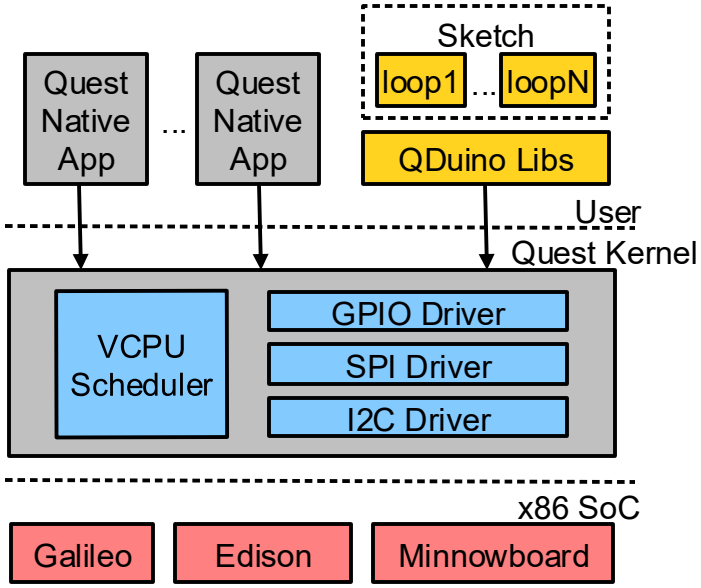
- Real-time Virtual CPU (VCPU) task and interrupt scheduling (RTAS'11)
- Real-time USB 2 (EHCI) and 3.x (xHCI) stack
  - Intent to address deficiencies in how systems predictably process I/O data (e.g., from sensors to actuators)
- Tuned pipes (RTSS'18, RTAS'20)
  - Predictable task pipeline construction from I/O devices through to host tasks
  - Like POSIX pipes but guarantee throughput and delay on communication

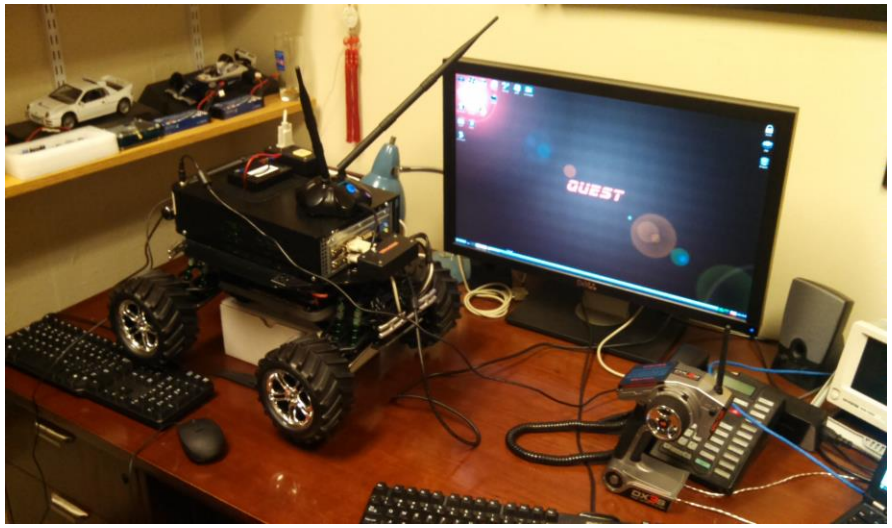




Qduino – Real-Time Arduino API for Quest

- **Arduino** – supports single control loops with manually inserted time delays
- **Qduino** – supports **multiple time-budgeted control loops** and **real-time I/O** on **multicore systems** without having to manual insert time delays into control loop execution (RTSS'15, RTAS'17 Best Student Paper)
  - Applied to web-connected 3D printer and autonomous ground vehicles







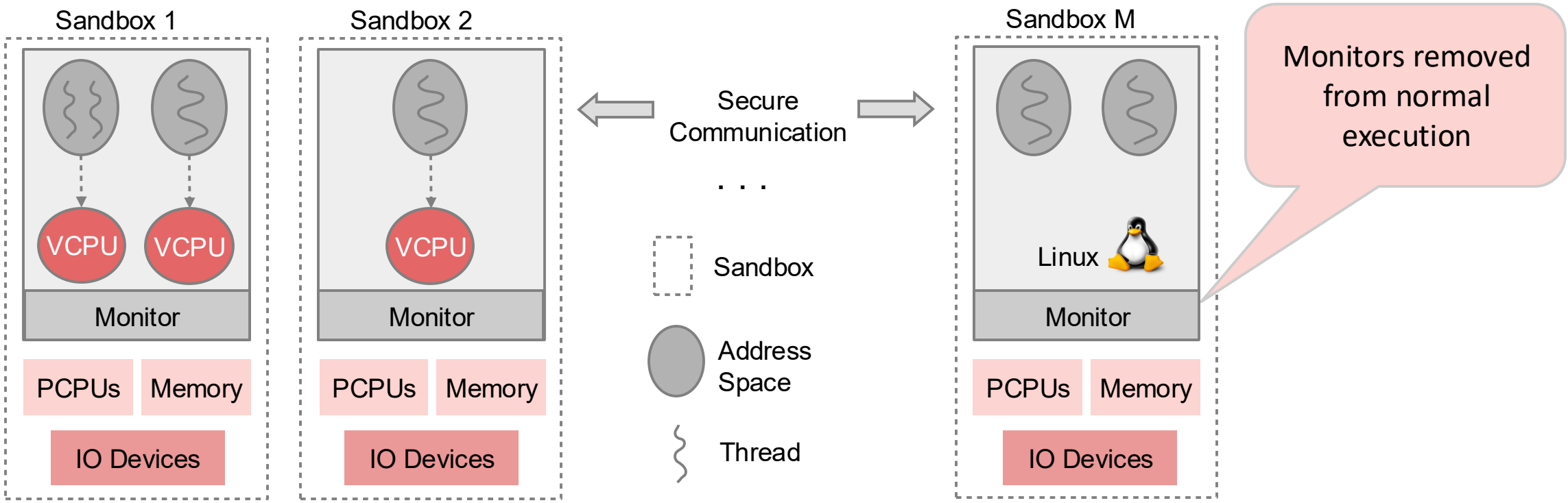
From Quest to Quest-V

- The need for **systems of systems** for embedded/CPS applications!
- **Symbiotic union** of services of legacy systems with complementary services of custom e.g., real-time system

- Distributed system on a chip
- Uses Intel VT-x capabilities found on PCs and SBCs/SoCs
- Separate sandbox kernels for system components
- Memory isolation using hardware-assisted memory virtualization
- Also CPU, I/O, cache partitioning (PACT'14, RTSS'16)

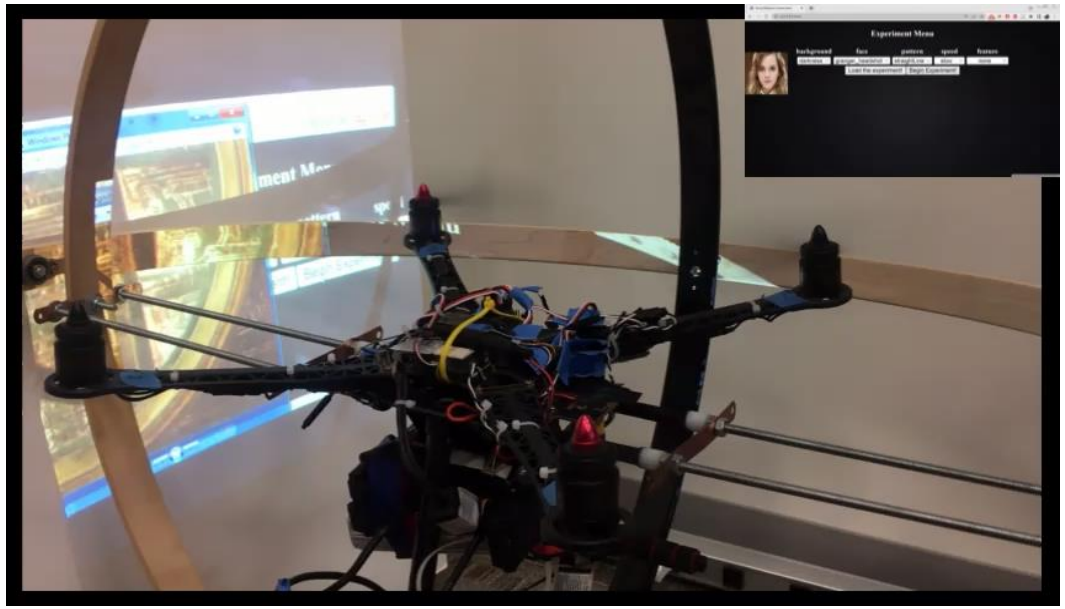
Quest-V Separation Kernel (VEE'14, ACM TOCS'16)

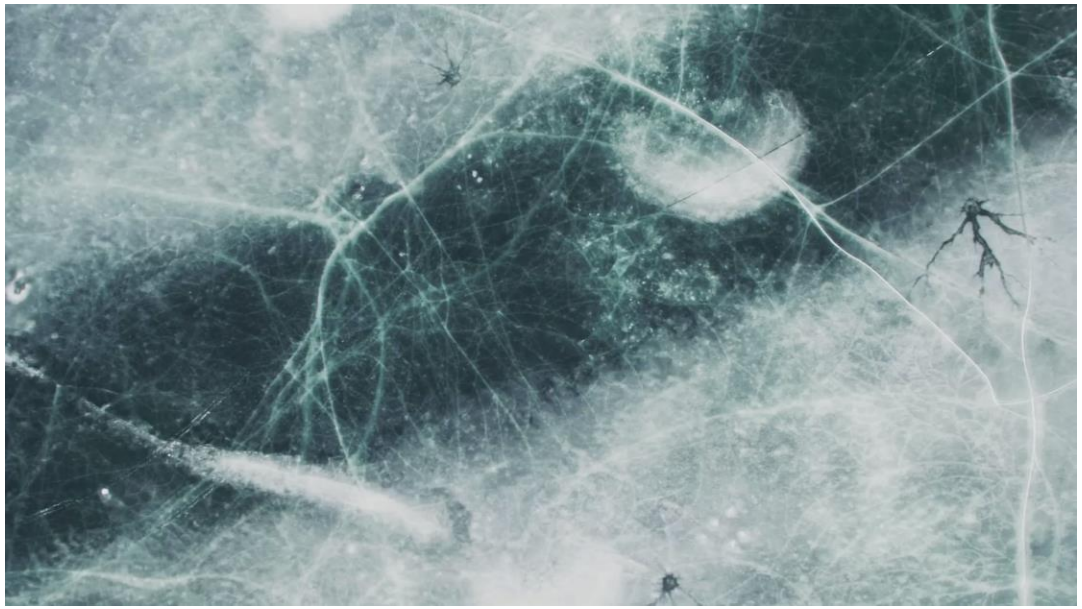
- **Partitioning hypervisor** – statically partitions resources (here, using a per-sandbox monitor)
- **Separation kernel** – distributed collection of sandboxed components, indistinguishable from separate private machines for each component



Quest(-V) Applications

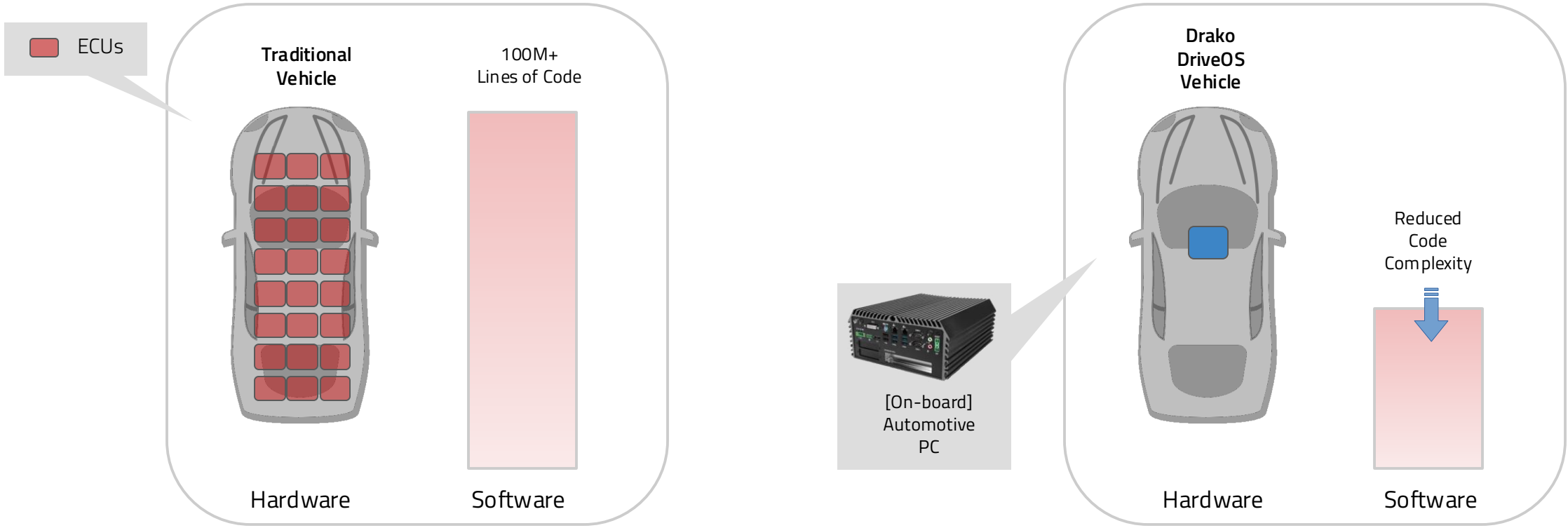
- **SmartFlight** (ECRTS'20 Best Paper) -- Environmentally-aware adaptive scheduling of drone flight controller
- **FlyOS** (RTAS'22 Outstanding/Best Student Paper) – Integrated Modular Avionics consolidated mixed-criticality mission and flight control
- **DriveOS** (EMSOFT'21 Best Paper nomination) -- Centralized software-defined vehicle system





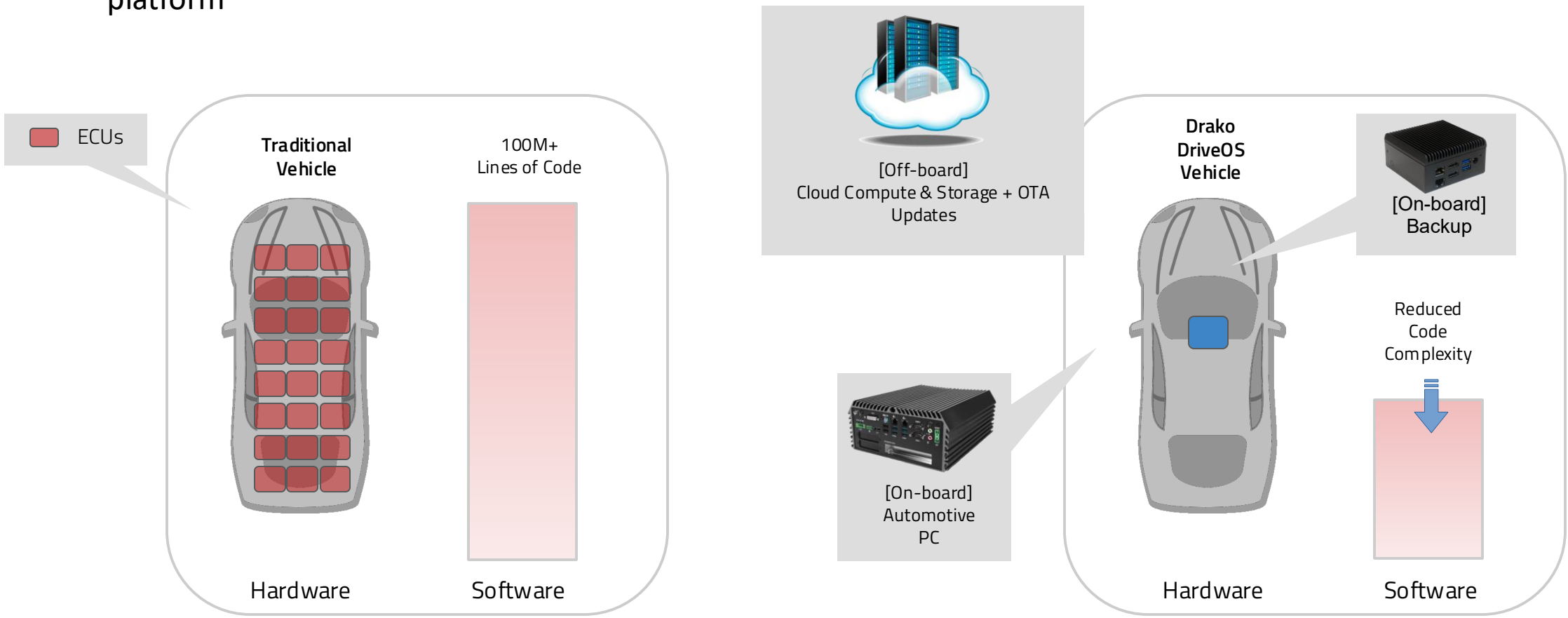
DRAKO DriveOS

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

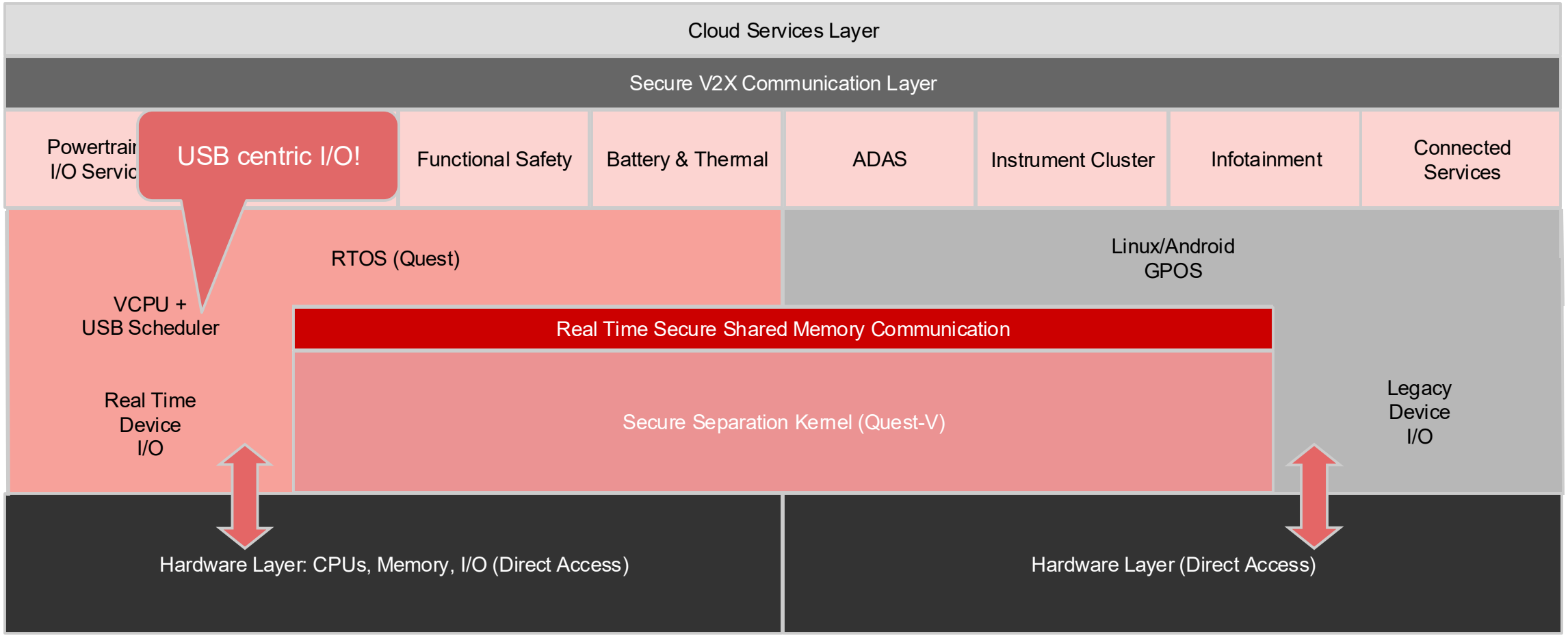


DRAKO DriveOS

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



DRAKO DriveOS Reference Stack



Where are We Going?

- 1980s – Personal Computing
- 1990s – World-wide web
- 2000s – Sensor networks (CPS Part 1) and mobile computing
- 2010s – Cloud computing & big data
- 2020s – AI/ML
- 2030s and beyond – Physical AI (CPS Part 2)

Grand Challenges

- Safety, security, predictability
  - e.g., automotive ISO26262 for software-defined vehicles
  - e.g., automotive ISO21434 for cyber-security (Note: Miller & Valasek Jeep Cherokee compromise)
- Need to move beyond memory safety and formal verification to embrace timing safety
- Compartmentalization (e.g., using new hardware features, MPKs, virtualization) to securely compartmentalize system components rather than entire OSes
- Security even more important with AI e.g., OpenAI sandbox breakout on Hugging Face hosting platform, Project Glasswing
- Will we have a Terminator (movie) scenario?

Acknowledgements

- Thanks to various funding sources (including but not limited to):
  - NSF #0615153, #0720464, #1117025, #1527050, #2007707 among others
  - Drako Tech Next-Generation Vehicle Fund gift
  - Intel gifts for Quest and Quest-V development
  - DARPA / Charles River Analytics for security-related research using Quest-V