Automotive Linux
LinuxCon Japan 2012

Rudi Streif

THE LINUX FOUNDATION

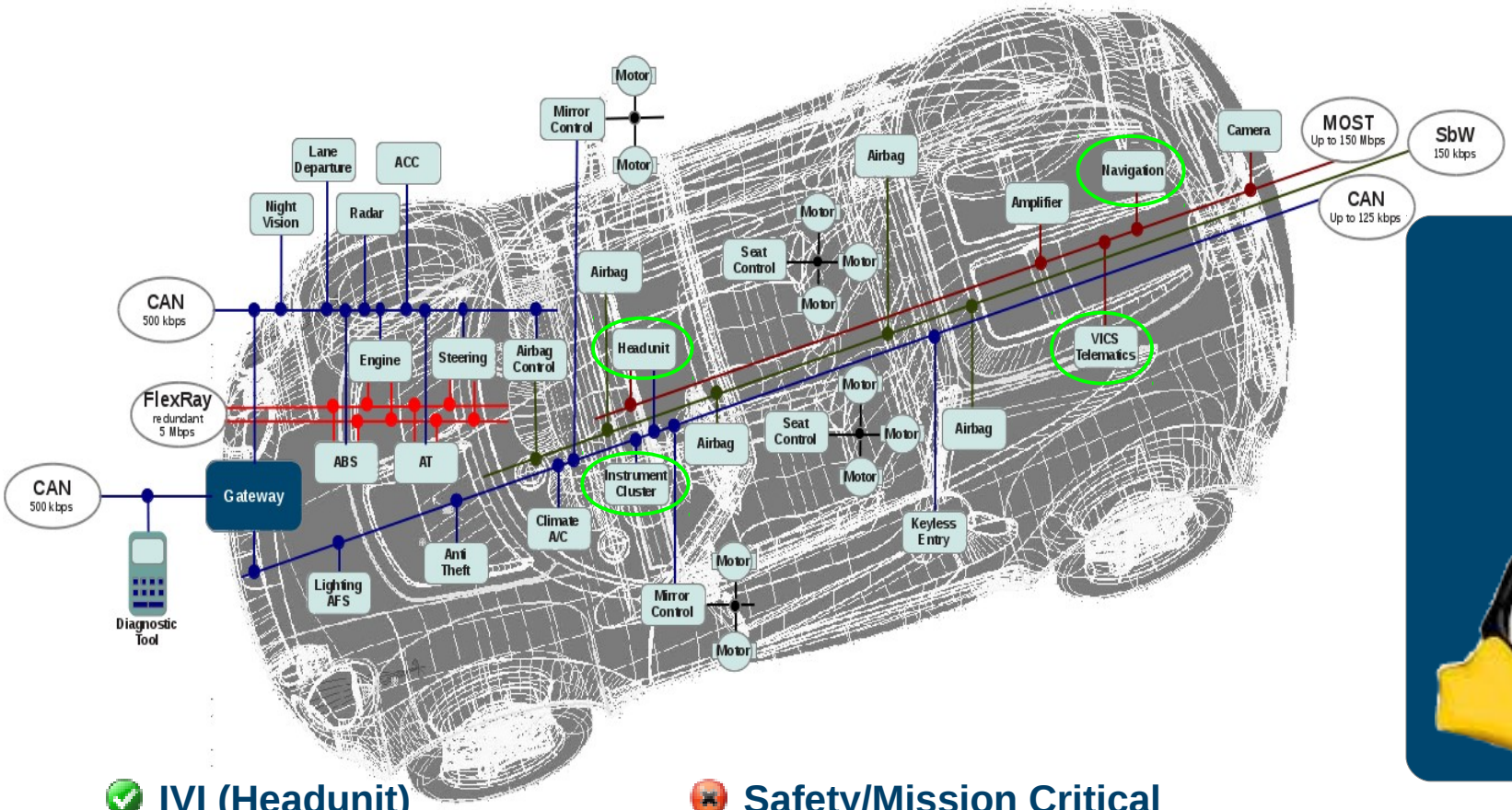
The Gordian Knot of Automotive Software Design

- **Consumer Electronics Industry is setting the pace...**
  - Rapid innovation and commodization of functionality are driving customer expectations.
  - Smart phones and other CE devices are increasingly being used to perform “traditional” tasks of automotive electronics: media playback, navigation, etc.
  - Car buyers demand integration and interoperability of their latest gadgets with in-vehicle systems.
  - Market demand for individualization and customization.
- **Conflicting with Automotive Industry reality...**
  - Product life cycles 10 years or longer.
  - Quality, reliability, durability and safety requirements.
  - Little to no software component reuse.

Linux looks like the perfect solution

- Proven technology used in millions of CE devices.
- Support for all major CPU architectures, a large variety of SoCs and a long list of peripherals.
- Contributions made by communications, consumer electronics and enterprise computing industries can be directly leveraged.
- Free of royalties. No cost for runtime licenses.
- No vendor lock-in.
- Reuse of software components.
- Large pool of developers and engineers.

Where does Linux fit in a car?



- IVI (Headunit)
- Instrument Cluster
- Navigation
- Telematics
- After-market

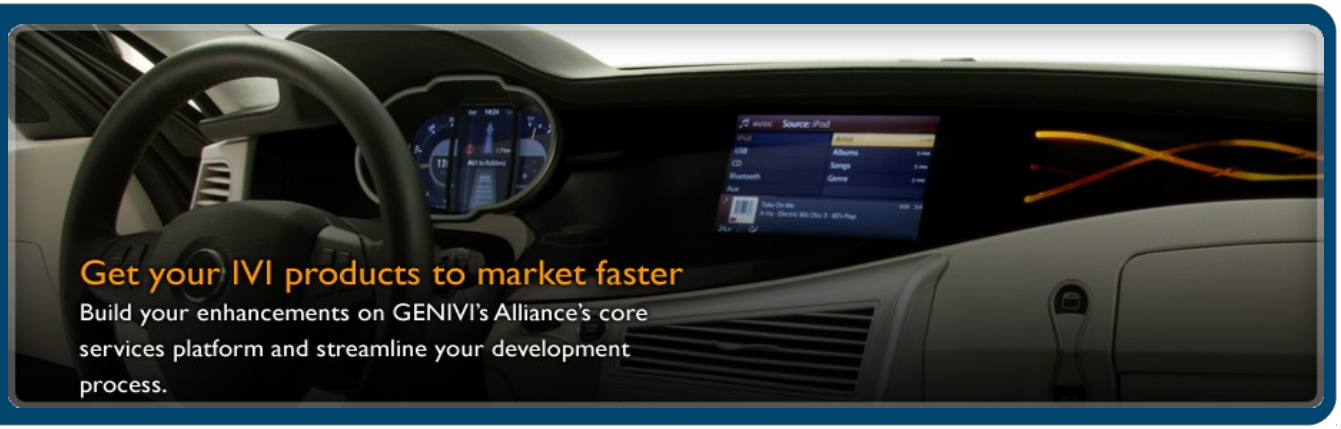
- Safety/Mission Critical
- Dedicated Control Systems

THE LINUX FOUNDATION

Early adopters are leading the way

GM's Linux-based Cadillac User Experience (CUE) will debut in 2012.

The GENIVI Alliance is standardizing a Linux-based software stack for in-vehicle infotainment.



THE LINUX FOUNDATION

Is Embedded Linux ready for automotive prime time?

Seven areas to bring Embedded Linux up to speed

- Embedded power management
- Startup, shutdown and loss of power
- File systems, storage and persistency
- Remote system updates and upgrades
- Diagnostic logging and tracing
- Embedded system security

Embedded Power Management

- **Relatively new discipline within Linux**
  - Android wakelocks have become a de-facto standard but they only address a small portion of the problem
- **Only a coherent set of functionality tied into the Linux kernel provides the necessary granularity for power management**
  - CPU Frequency Control
  - CPU Idle State Control
  - Clock Management
  - System-wide Suspend to RAM/Disk and Resume
  - Regulator Framework
  - Resource Management
  - Power Instrumentation and Profiling

Startup, Shutdown and Loss-of-Power

- **Initialize critical hardware components in less than 50 ms from cold start**
  - Requires tight control over when the Linux kernel initializes device drivers
- **Audio playback in less than 1 s from cold start**
  - Some driver assist systems are using audio feedback e.g. proximity sensors
- **Video display in less than 3 s from cold start**
  - Driver assist system using ive images from rear view (surround view) cameras
  - Possibly overlaid with computer-generated graphics visualizing information from other sensors
- **Wake-on-Network**
  - Partial or entire startup on activity on vehicle networks e.g. CAN, MOST
- **Loss-of-Power Tolerance**
  - Must never result in unrecoverable state

File Systems, Storage and Persistency

- **Temporary Storage**
  - Store temporary files using file systems on volatile memory (RAM disk).
  - Avoid wearing flash memory over the lifetime of the vehicle (potentially > 10 years).
- **Persistent Storage**
  - On wear-leveling flash file systems.
- **User Data Protection**
  - Unlike a mobile phone a car is a multi-user device.
  - User data must be identifiable by user and partitioned from each other.
- **User Data Quotas**
  - Enforce quotas for user data to prevent running out of capacity.

System Updates and Upgrades

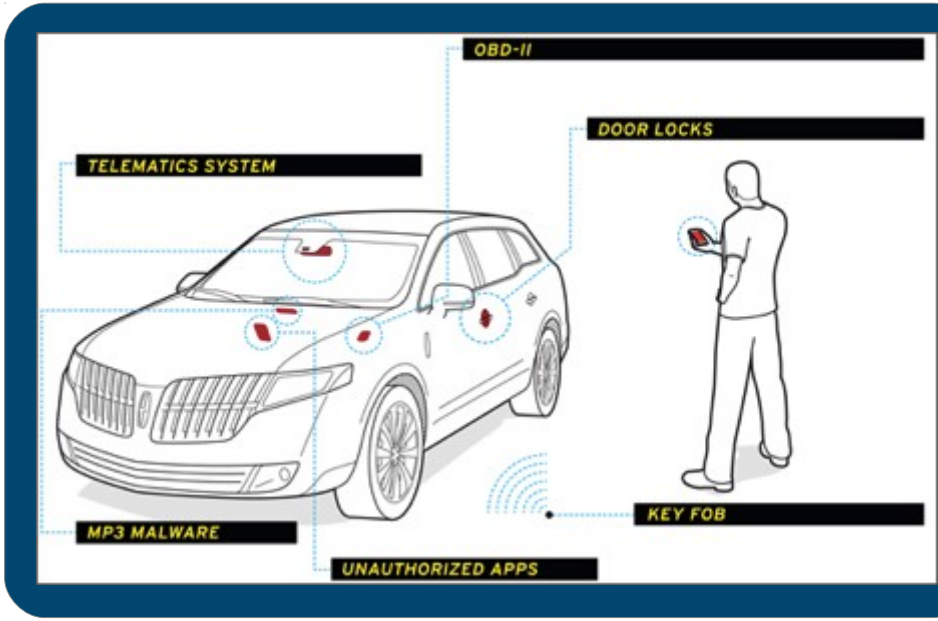
- **Embedded systems impose several constraints on system updates**
  - Limited access to interfaces
  - User interaction is either not possible or very limited and in many cases not desired
  - Update package size limited by the device's storage and processing capabilities
  - Constrained time windows for updates
- **Vehicles impose additional constraints**
  - Not permanently connected to data networks
  - QoS for delivery networks is not guaranteed
- **Update mechanism must meet several requirements**
  - Delivery must be resumable
  - Verification of integrity, point of origin and destination
  - Updates must be transactional
  - Updates must be incremental

Diagnostic Logging and Tracing

- **The wheel reinvented over and over again**
  - Syslog is the UNIX standard but it has its limitations
    - Only one of many logging facilities and not for kernel log, early boot and late shutdown messages
    - Mostly unstructured and unstandardized log data
    - Messages are not authenticated
    - Timestamps have no timezone information
    - No compression, limited disk space monitoring
  - Every embedded developer seems to write their own logging facility
  - “Journal” is seeking to address the issues
- **For embedded systems also required are:**
  - Remote retrieval of log data
  - Access control to ensure security and privacy (potentially encrypted log data)
  - Deterministic performance

Caution! Malware Ahead!

- **Attack Surfaces Exposed**
  - MP3 Malware via USB Memory Sticks
  - OBD-II
    - CarShark (UCSD/UW, www.autosec.org)
  - Key Fob Attacks
    - ETH Zuerich
  - Tire-pressure Monitor (RFID)
    - USC/Rutgers
- **And there is more down the road**
  - Unauthorized apps
  - Intentional and unintentional modification of system software



Automotive Network Security

- **Automotive networks are not secure!**
  - CAN, MOST, FlexRay, LIN are not designed to use authentication, encryption and other security mechanisms.
  - Most of them will probably never implement any security mechanisms for various reasons: cost, speed, processing overhead, etc.
  - They are wire-bound and therefore physical access is required – well yes, but...
- **Physical access is relatively simple!**
  - Vehicle service
  - Aftermarket devices
  - Consumer devices
- **Systems operating as gateways extend connectivity beyond vehicle boundaries.**
  - VICS connects smartphones via Bluetooth.
  - WiFi hot spots allow access to data and media and provide tethering.
  - Wireless data radio for telematics.

Securing an Embedded Linux Platform

- **Start with Secure Software Practices**
  - Hardened Platform
    - Trimmed codebase
    - Locked accounts and permissions
  - Design for Security with a Framework for Trust
    - Establish trust boundary (interfaces)
    - Define root of trust for each component/interface
    - Establish a digital trust assessment model
    - Integrate failsafe/recovery proceducre
  - Software Reviews and Assessment
    - Source code reviews with qualified security practitioners
    - Blackbox testing
  - Sustained Integrity Monitoring through Usable Life
- **Utilize proven Security Concepts**
  - Secure Hardware
    - TPM or other hardware-bound security
  - Remote Attestation
    - Hash-key summary of hardware/software configuration
  - Binding
    - Encrypt data destined for a target device
  - Sealed Storage
    - Protect user data and privacy

Is the Automotive Industry ready for Linux?

- **Four stages of “Open Source Maturity”**
  - Discovery
  - Adoption
  - Contribution
  - Initiation
- **The majority of the industry is at the Discovery and Adoption stages**
- **Eventually the industry will need to mature to become a “good citizen” of the open source community**
  - Contribute to the Linux kernel and other open source projects the industry is building on.
  - Initiate new open source projects for the broader benefit of the community.

Learn more at the Automotive Linux Summit 2012

https://events.linuxfoundation.org/events/automotive-linux-summit

THE LINUX FOUNDATION

Thank you for your time! Questions?