



JSR xxx – A new telematics API ?

hendrik.hoefer@microdoc.com

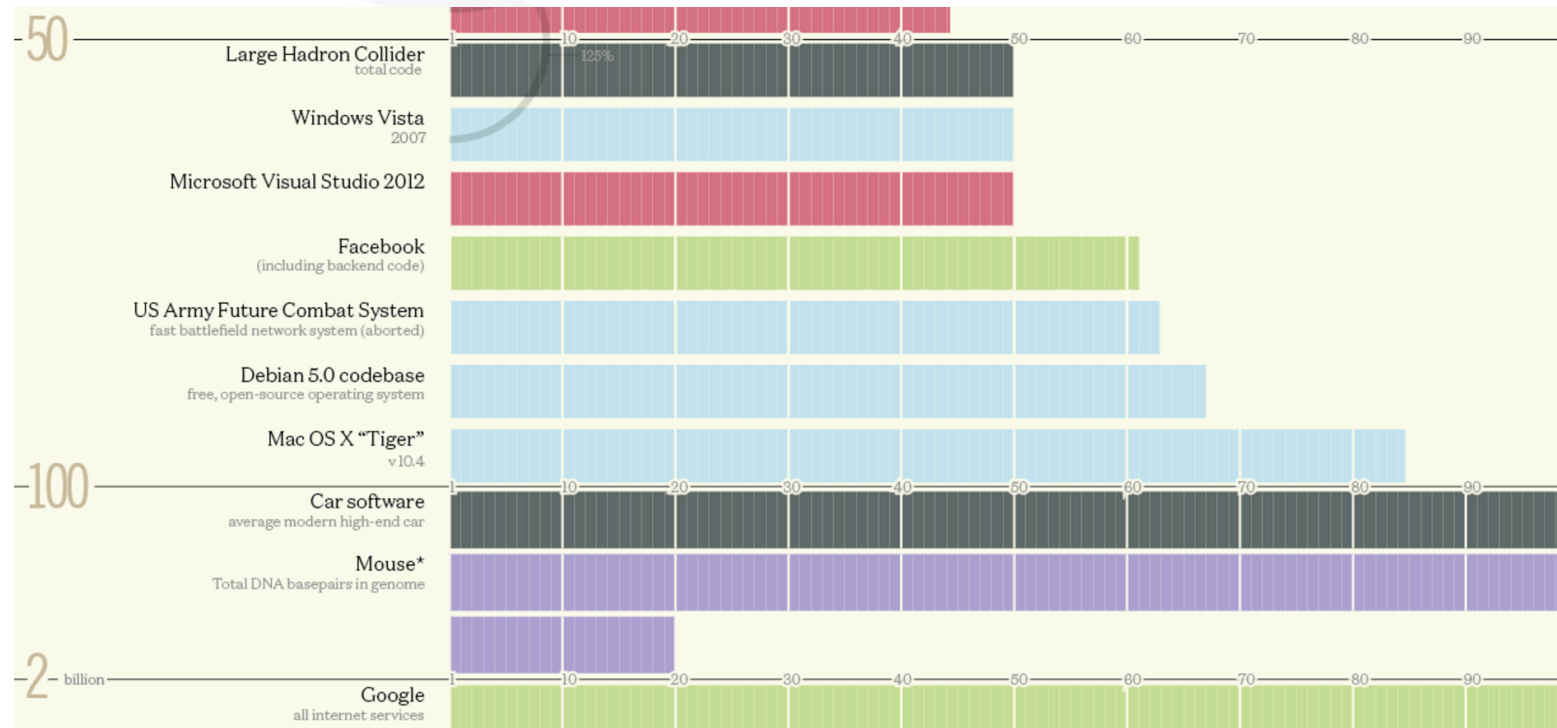
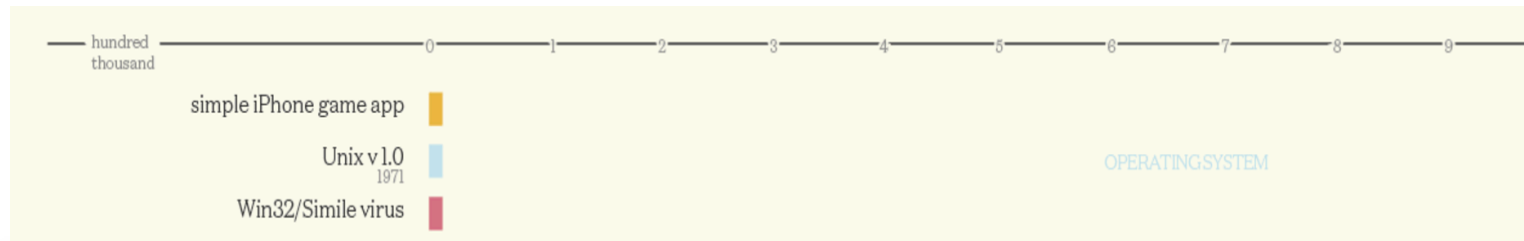


Connected Car: Some observations

- Telematics business models are quite successful in commercial environments (e.g. Logistics, Remote Maintenance), but have a slow adoption in the consumer space
- Most OEMs are building their own, mainly proprietary app – ecosystem to increase the value of their products
- OEMs consider Google and Apple to be their competitors
- OEMs are very concerned about safety/privacy/data ownership and want to/must control access to the safety critical components and private data in a vehicle



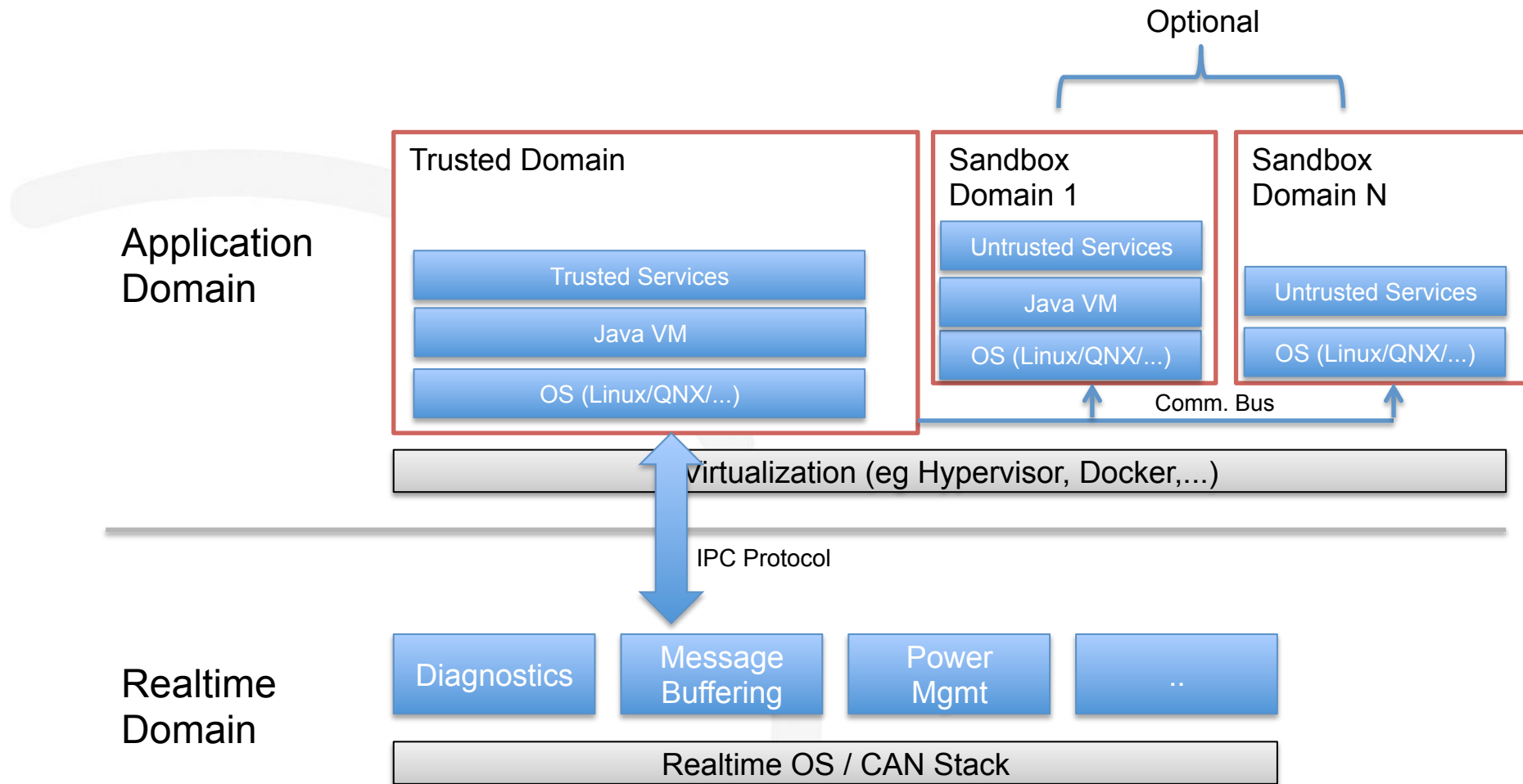
The amount of software in modern cars is astonishing



<http://www.informationisbeautiful.net/visualizations/million-lines-of-code/>



A typical Telematics Component Architecture





JSR 298: Telematics API for Java ME – 2008

- “This specification defines methods for controlling and obtaining diagnostic information and conditions on various components built in to the vehicle.”

Interface name	Component name	Possible Operations	States or Value [Unit]
AntilockBrakingSystem	AntilockBrakingSystem		On/Off
Airbag	DriverAirbag	- Set Mode	Enabled/Disabled
	PassengerAirbag		Fired off/Not Fired off
AirConditioning	AirConditioning	Turn on/off	On/Off
Antenna	Antenna	Extend/Retract	Extended/Retracted
Battery	Battery	-	Electricity Level [Voltage]
Brake	Brake	-	Engaged/Not Engaged
BrakeFluid	BrakeFluid	-	Fluid Level [Liter]
ChargingPlug	ChargingPlug	Plug Out/In	Plugged/Unplugged
DashboardIllumination	DashboardIllumination	Level set	Illumination Level [Level]
DifferentialOdometer	DifferentialOdometer	Reset	0 (Zero)
			Incremental Distance [Kilometer]
Door	DriverDoor	Lock/Unlock	Locked/Unlocked
	PassengerDoor	Open/Close	Opened/Closed
Engine	Engine	Start/Stop	Running/Stopped



- What might have hindered adoption also is
 - “For TCK we will charge a single one time fee of max \$50000 USD and an annual maintenance fee, max \$20000 USD/pa.”
- This JSR might be useful to build a telematics product within an organization, but as an API for 3rd party code it seems too “broad”



A very successful business model in the consumer space

- <http://www.octotelematics.com>
 - “Octo is the global brand leader in providing insurance telematics services as well as pioneering applications in motor rental and fleet management, car manufacturing, governmental sectors and a fast growing range of specialist applications.”
 - Octos solutions (hard- & software) run in several million vehicles
- Octo wants to help standardize (some) in-vehicle APIs

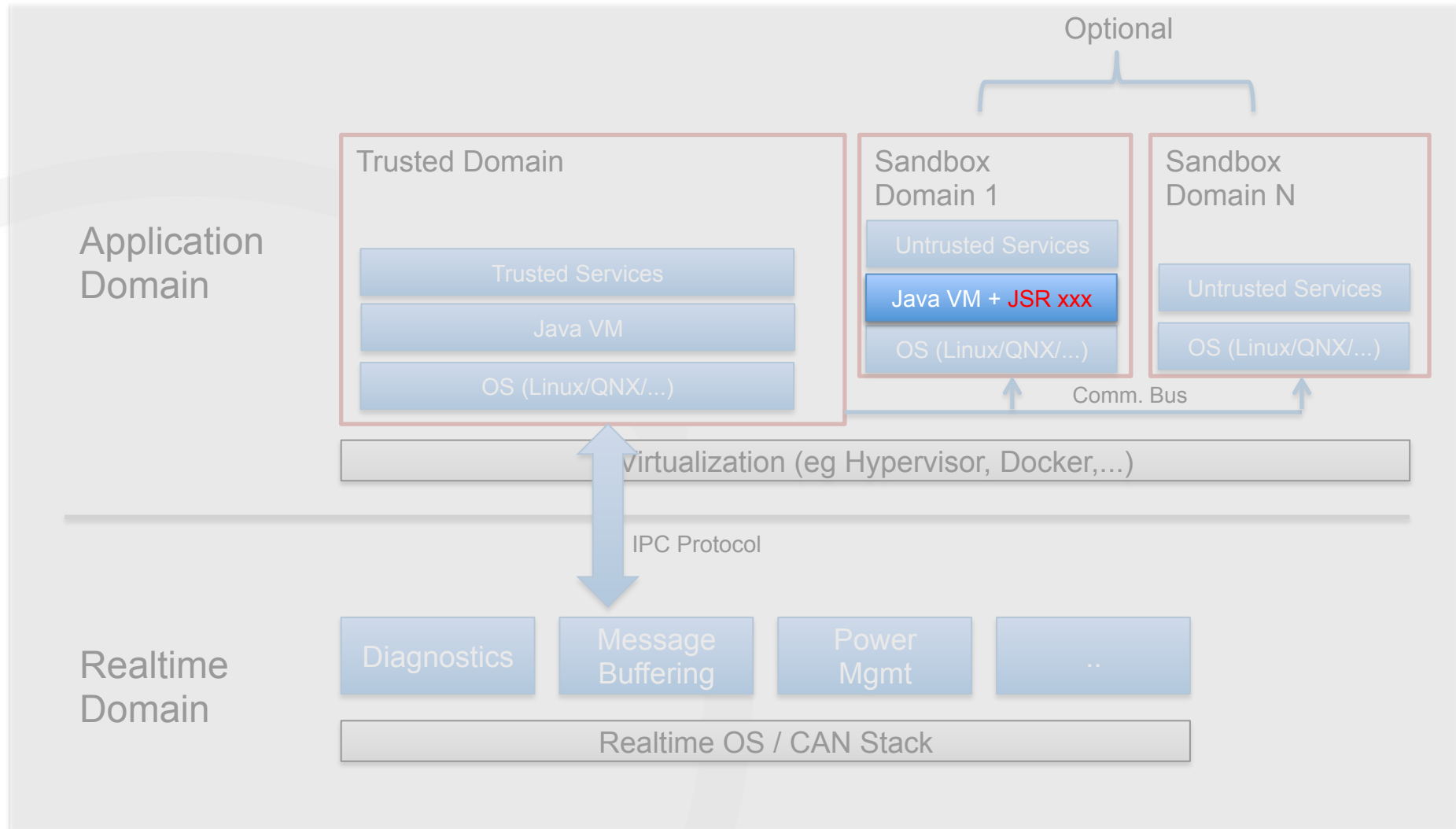


Technical scope for Version 1 of a new Telematics JSR

- APIs for consuming data
 - Derived from OBD, FMS standards
 - Speed, Accelerometer, GPS, Gyro, Ignition, etc.
- Other APIs
 - Lifecycle (startup/shutdown/suspend/reset)
 - Configuration
 - Persistence
- APIs for sending/receiving data to/from backend
 - Likely in version 1
- API HMI integration
 - Unlikely in version 1



A typical Telematics Component Architecture





What are we doing right now ?

- Working through the legal framework (JSPA)
- Define the technical scope for this JSR
- Work on the the JSR proposal
- Find the right people for the expert group