AUTOMOTIVE LINUX SUMMIT

OpenGL Graphics Drivers in Safety Critical Environments: Fact, Fiction and Future

Rick Tewell
July 2016

Joshua Brown

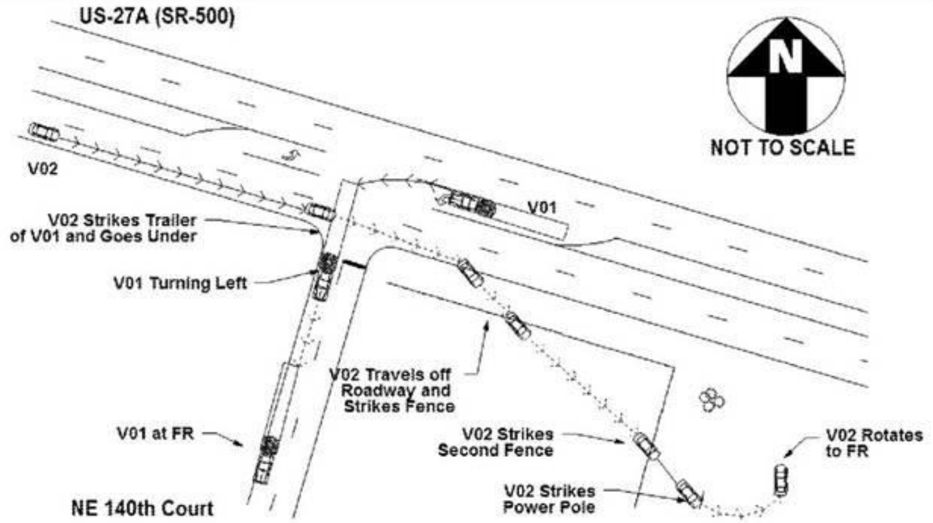
40 years old
Navy SEAL for 11 years
Owner of a wireless networking tech company
Lived in Canton Ohio

Joshua Brown

40 years old
Navy SEAL for 11 years
Owner of a wireless networking tech company
Lived in Canton Ohio
Avid lover of his Tesla Model S

Tesla-S Autopilot Failure

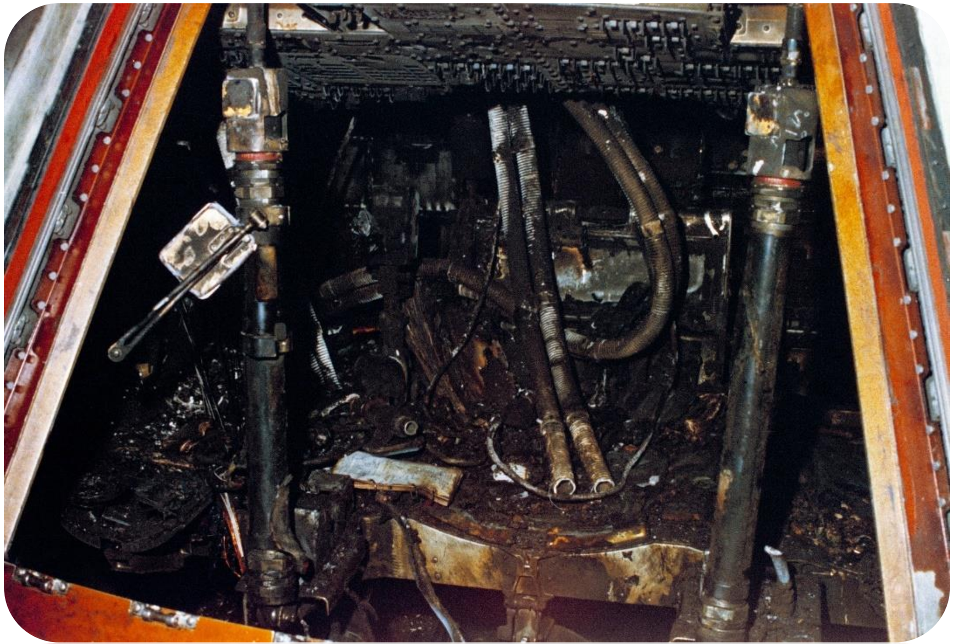
Date of Crash	Date of Report	Invest. Agency Report Number	HSMV Crash Report Number
07/May/2016 04:40 PM	07/May/2016 04:40 PM	FHPB16OFF012208	85234095



Died on May 7, 2016 in northern Florida when his Tesla-S "autopilot" failed to stop when a tractor-trailer made a legal turn in front of him His Tesla-S struck the trailer at 65 mph (105 kph).

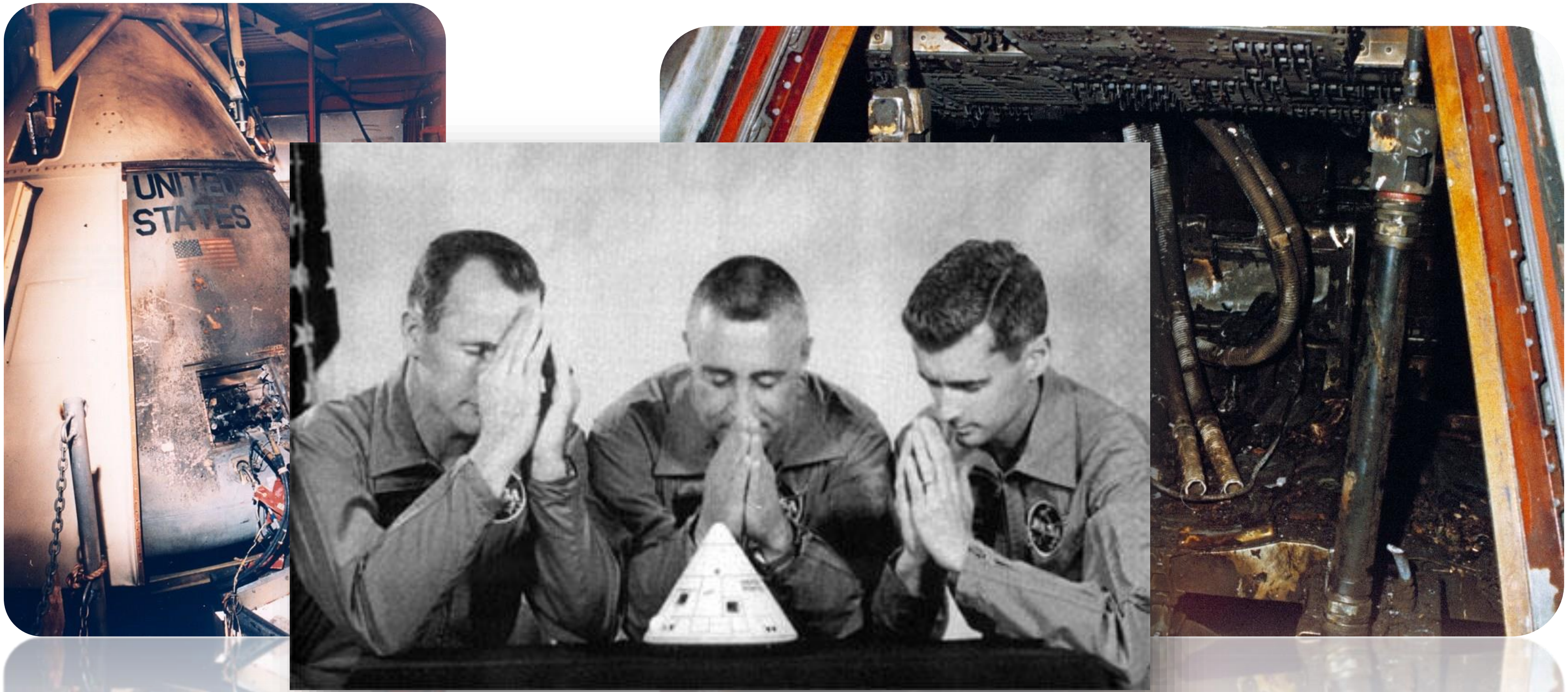
His Tesla Model S crashed after failing to activate its brakes because the auto pilot function didn't realize that the white side of a tractor-trailer in front of the vehicle was not the sky.

Apollo 1 Fire – January 27, 1967



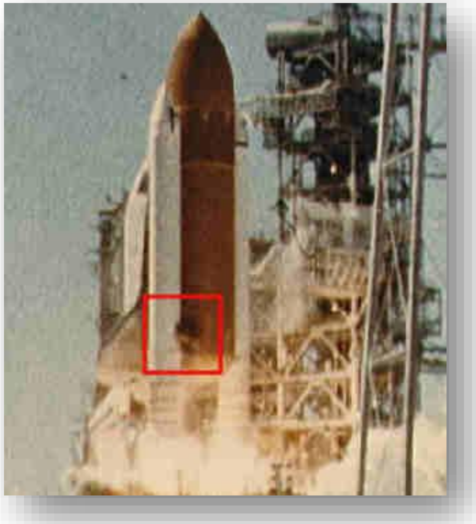
- ✓ Pure oxygen environment
- ✓ Capsuled at high pressure - 16.7 psi - 14.7 psi (sea level)
- ✓ 34 square feet of super flammable Velcro - almost like carpeting
- ✓ Highly flammable nylon space suits
- ✓ Hatch design - couldn't be opened if pressure above sea level
- ✓ It was generally known that Apollo Block I had potential safety issues

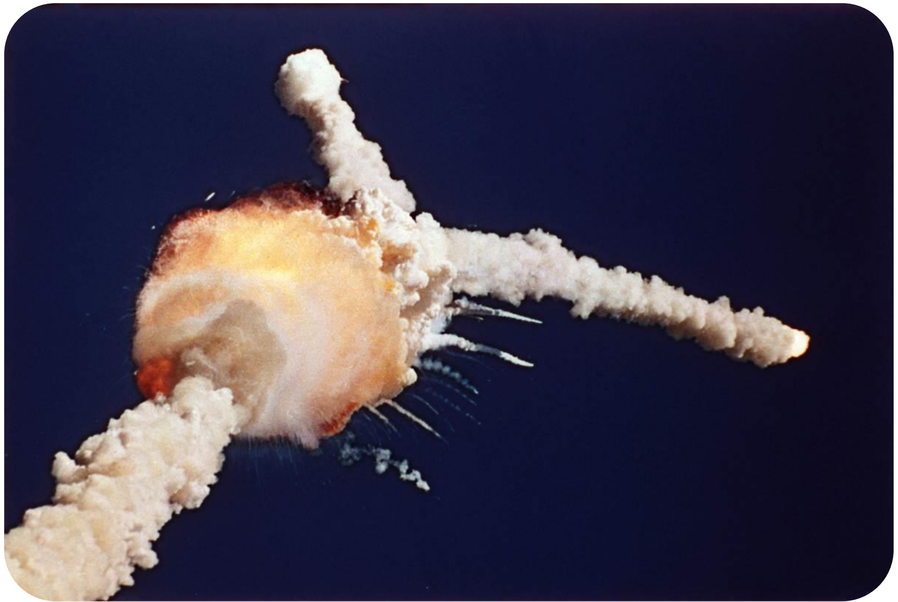
Apollo 1 Fire – January 27, 1967

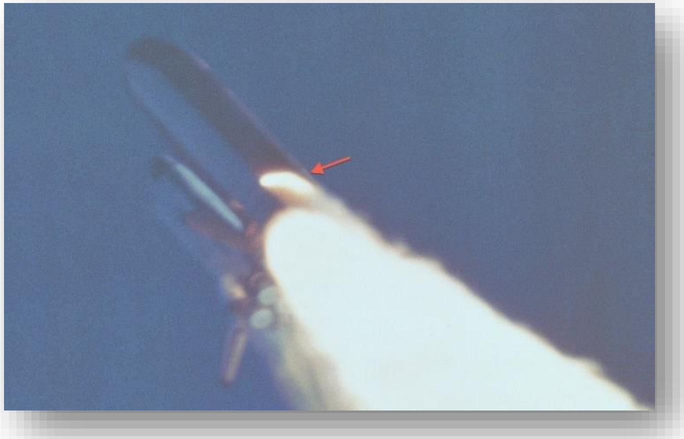


- ✓ Pure oxygen environment
- ✓ Capsuled at high pressure - 16.7 psi - 14.7 psi (sea level)
- ✓ 34 square feet of super flammable Velcro - almost like carpeting
- ✓ Highly flammable nylon space suits
- ✓ Hatch design - couldn't be opened if pressure above sea level
- ✓ It was generally known that Apollo Block I had potential safety issues

Space Shuttle Challenger – January 28, 1986







- ✓ Solid rocket boosters O-rings become brittle at low temperatures (known at time of launch)
- ✓ Shuttle sat at extremely low temperatures for hours overnight prior to launch
- ✓ At launch temperature was "within range" but O-rings were still frozen solid and brittle

Rare Occurrences?

- Shutdown of Atlanta Airport due to software not reporting that a security screening test was underway. Thought by security to be a "real" incident…

- Total loss of communication between Air Traffic Control and Aircraft at LA Airport for three hours - Microsoft Windows server 50-day "reboot" to prevent data overload…

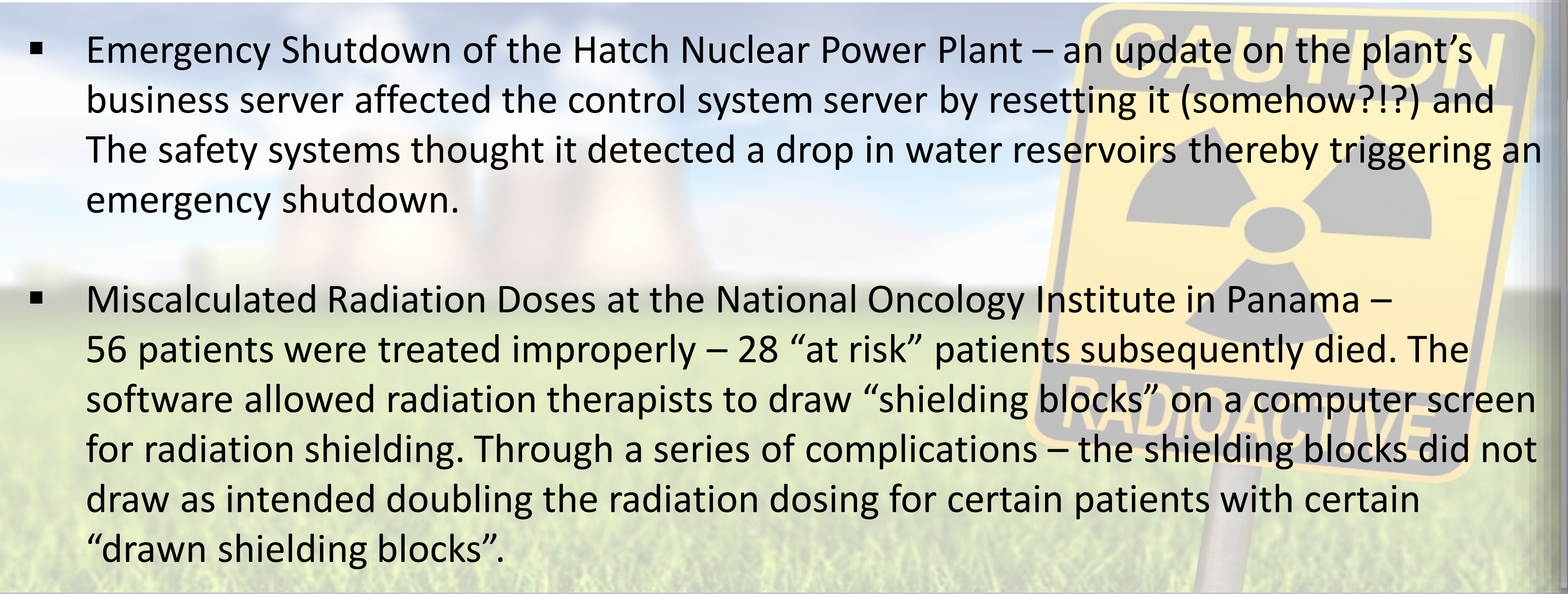
- Crash of Air France Flight 447 – Airbus A330-200 - unreliable cockpit reporting of airspeed and other critical flight information –all 228 people on board perished…

- Crash of Korean Air Flight 801 – Boeing 747-300 – ATC disabled minimum safe altitude warning function in the radar system because it generated alerts that were considered annoying – flight crashed into a hill on approach to Guam airport - 228 died and 26 survived with major injuries.

- Crash of American Airlines Flight 965 – autopilot flew a Boeing 757 into a mountain near Bogota, Colombia – flight system was set for waypoint Rozo instead of Romeo because the co-pilot entered an "R" <enter> into the system and the FMS selected the wrong waypoint by default.

Rare Occurrences?

- Loss of Mars Polar Lander – premature engine shutdown due to spurious signals that touchdown had occurred – total loss of spacecraft.
- Loss of Mars Climate Orbiter – imperial units programmed into flight system instead of metric units – total loss of spacecraft.
- Misplacement of Satellite by Launch Vehicle – RCS system ran out of fuel due to unexpectedly large number of initial launch stabilization corrections due to improper constants being compiled into the flight software causing the vehicle to roll during uphill flight – total loss of satellite.



- Emergency Shutdown of the Hatch Nuclear Power Plant – an update on the plant's business server affected the control system server by resetting it (somehow?!?) and The safety systems thought it detected a drop in water reservoirs thereby triggering an emergency shutdown.
- Miscalculated Radiation Doses at the National Oncology Institute in Panama – 56 patients were treated improperly – 28 "at risk" patients subsequently died. The software allowed radiation therapists to draw "shielding blocks" on a computer screen for radiation shielding. Through a series of complications – the shielding blocks did not draw as intended doubling the radiation dosing for certain patients with certain "drawn shielding blocks".

Rare Occurrences?

- Patriot Missile – Software Bug Led to System Failure at Dhahran, Saudi Arabia – the radar ranging incoming detection system would "drift" over time – requiring a periodic restart to keep the range detection system accurate. This particular Patriot system had been running for well over 100 hours without a restart and therefore was wildly inaccurate (restarts were recommended every eight hours) and looking in the wrong place for incoming missiles. An incoming missile went undetected and 28 US military personnel were killed and 98 more injured.

Reasons for Catastrophic Failures?

- Failure of Imagination
- Irrational Exuberance - "Go Fever“
- Incorrect Assumptions

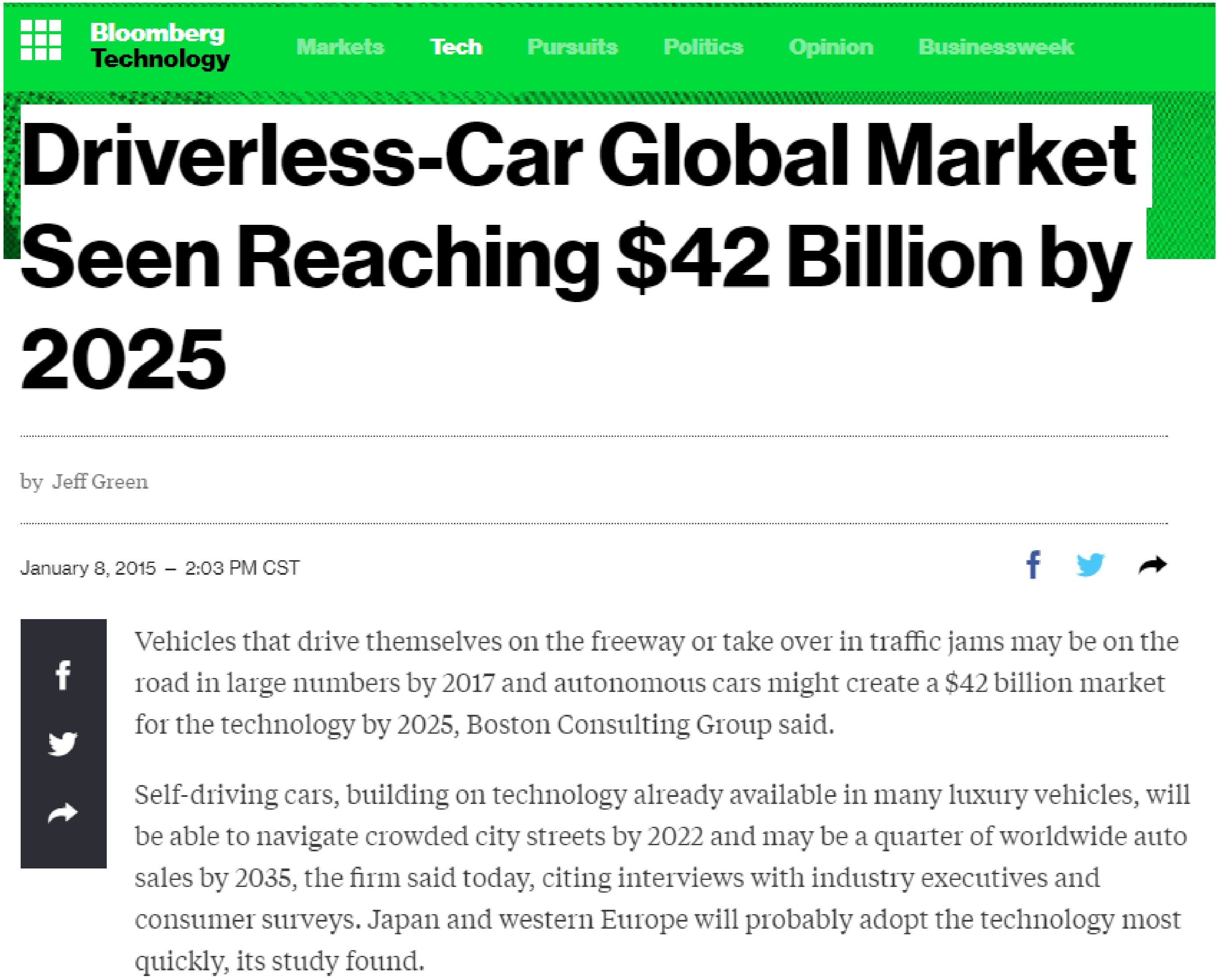
Consequences

- 38,000 Automobile Crash Deaths in 2015 in the USA

  This is the equivalent of a fully loaded Boeing 747 –and- a fully loaded Airbus A330 crashing ***<u>every</u>*** week killing everyone on board

- Will autonomous vehicles on the road improve the situation or make it worse?

Follow the Money



Driverless-Car Global Market Seen Reaching $42 Billion by 2025

by Jeff Green

January 8, 2015 – 2:03 PM CST

Vehicles that drive themselves on the freeway or take over in traffic jams may be on the road in large numbers by 2017 and autonomous cars might create a $42 billion market for the technology by 2025, Boston Consulting Group said.

Self-driving cars, building on technology already available in many luxury vehicles, will be able to navigate crowded city streets by 2022 and may be a quarter of worldwide auto sales by 2035, the firm said today, citing interviews with industry executives and consumer surveys. Japan and western Europe will probably adopt the technology most quickly, its study found.

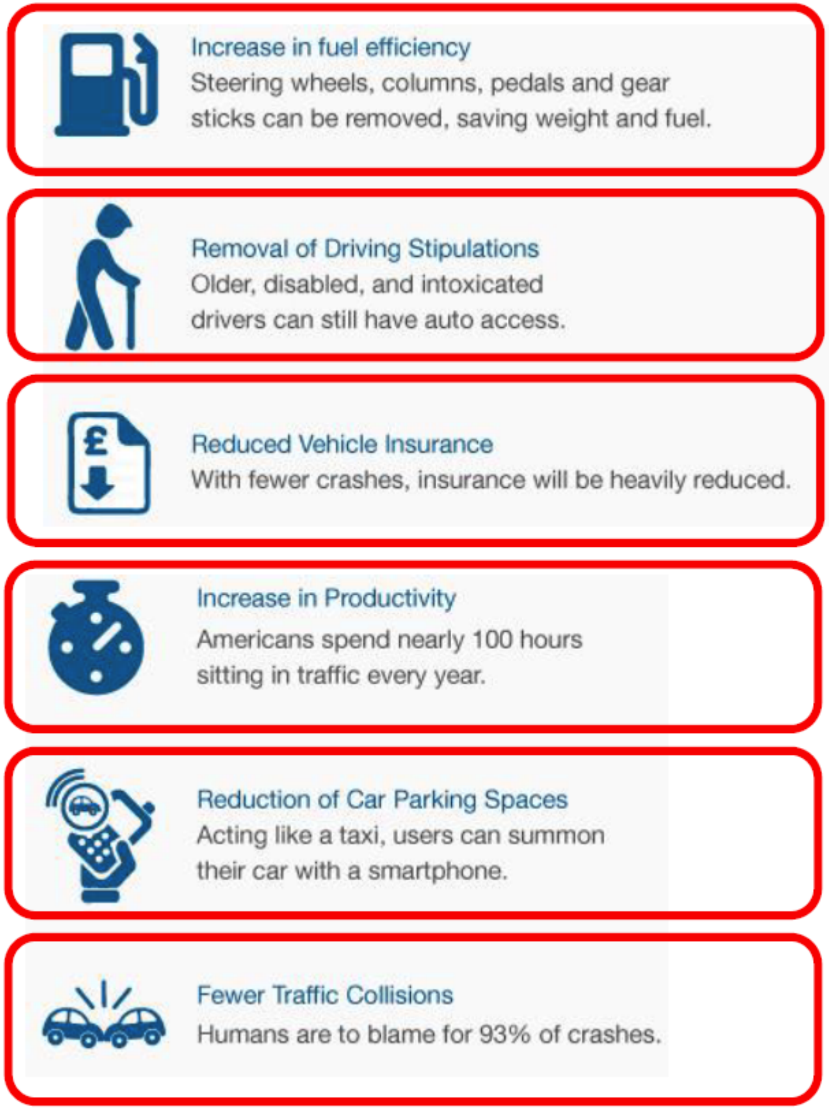
The Self Driving Car!

Are we ready?

Is the technology ready?

How can we help?

Benefits of Autonomous Car?



Increase in fuel efficiency
Steering wheels, columns, pedals and gear sticks can be removed, saving weight and fuel.

Removal of Driving Stipulations
Older, disabled, and intoxicated drivers can still have auto access.

Reduced Vehicle Insurance
With fewer crashes, insurance will be heavily reduced.

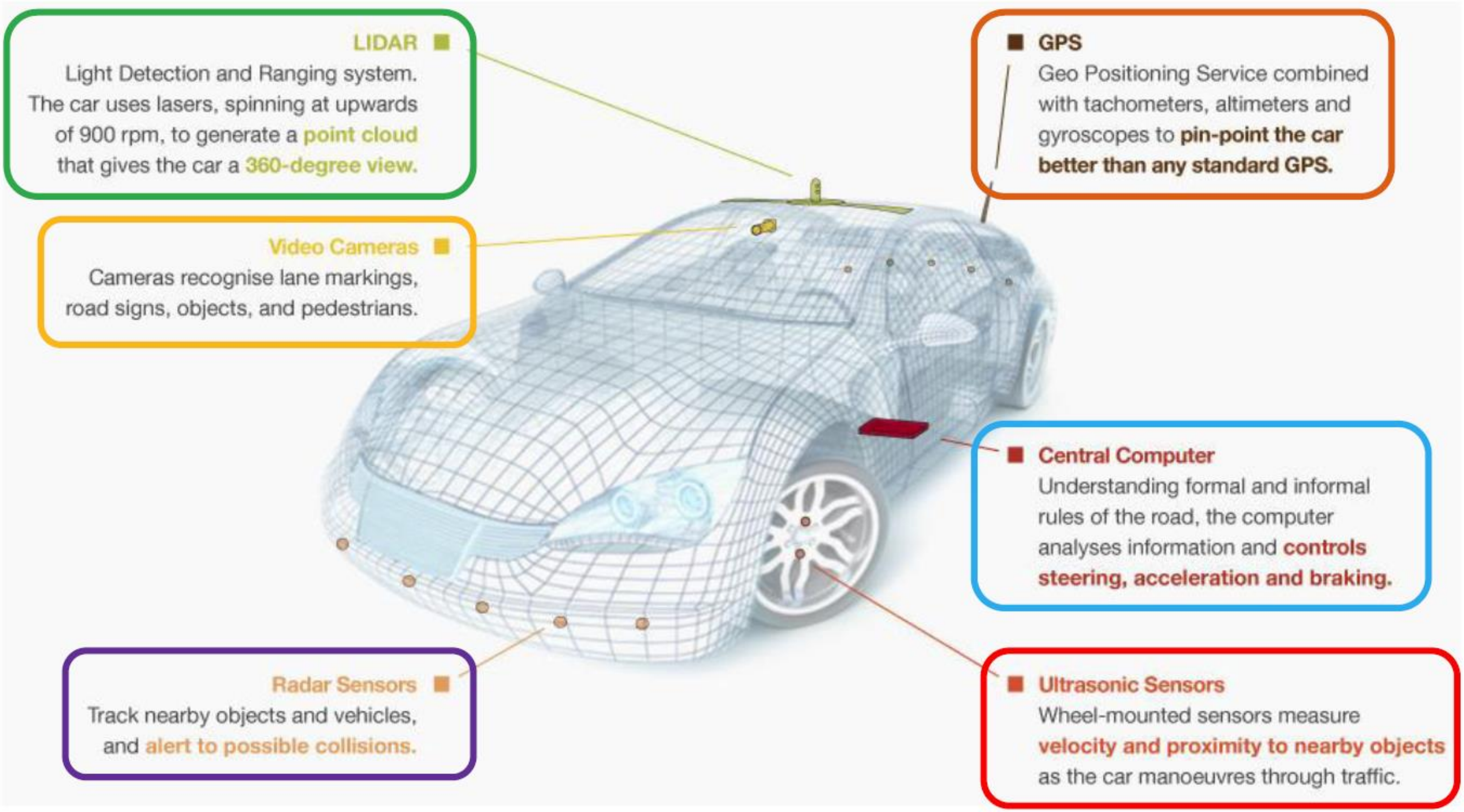
Increase in Productivity
Americans spend nearly 100 hours sitting in traffic every year.

Reduction of Car Parking Spaces
Acting like a taxi, users can summon their car with a smartphone.

Fewer Traffic Collisions
Humans are to blame for 93% of crashes.

Google's Aspiration	Potential Annual Benefits (US only)
90% reduction in ACCIDENTS	4.95 million fewer accidents 30,000 fewer deaths 2 million fewer injuries $400 billion saving in cost
90% reduction WASTED COMMUTING	4.8 billion fewer commuting hours 1.9 billion gallons in fuel savings $101 billion saved in lost productivity and fuel cost
90% reduction in CARS	Reduce cost per trip-mile by 80%+ Car utilization from 5-10% to 75%+ Better land use.

Self Driving Car Technologies



LIDAR

Light Detection and Ranging system. The car uses lasers, spinning at upwards of 900 rpm, to generate a **point cloud** that gives the car a **360-degree view.**

Video Cameras

Cameras recognise lane markings, road signs, objects, and pedestrians.

Radar Sensors

Track nearby objects and vehicles, and **alert to possible collisions.**

GPS

Geo Positioning Service combined with tachometers, altimeters and gyroscopes to **pin-point the car better than any standard GPS.**

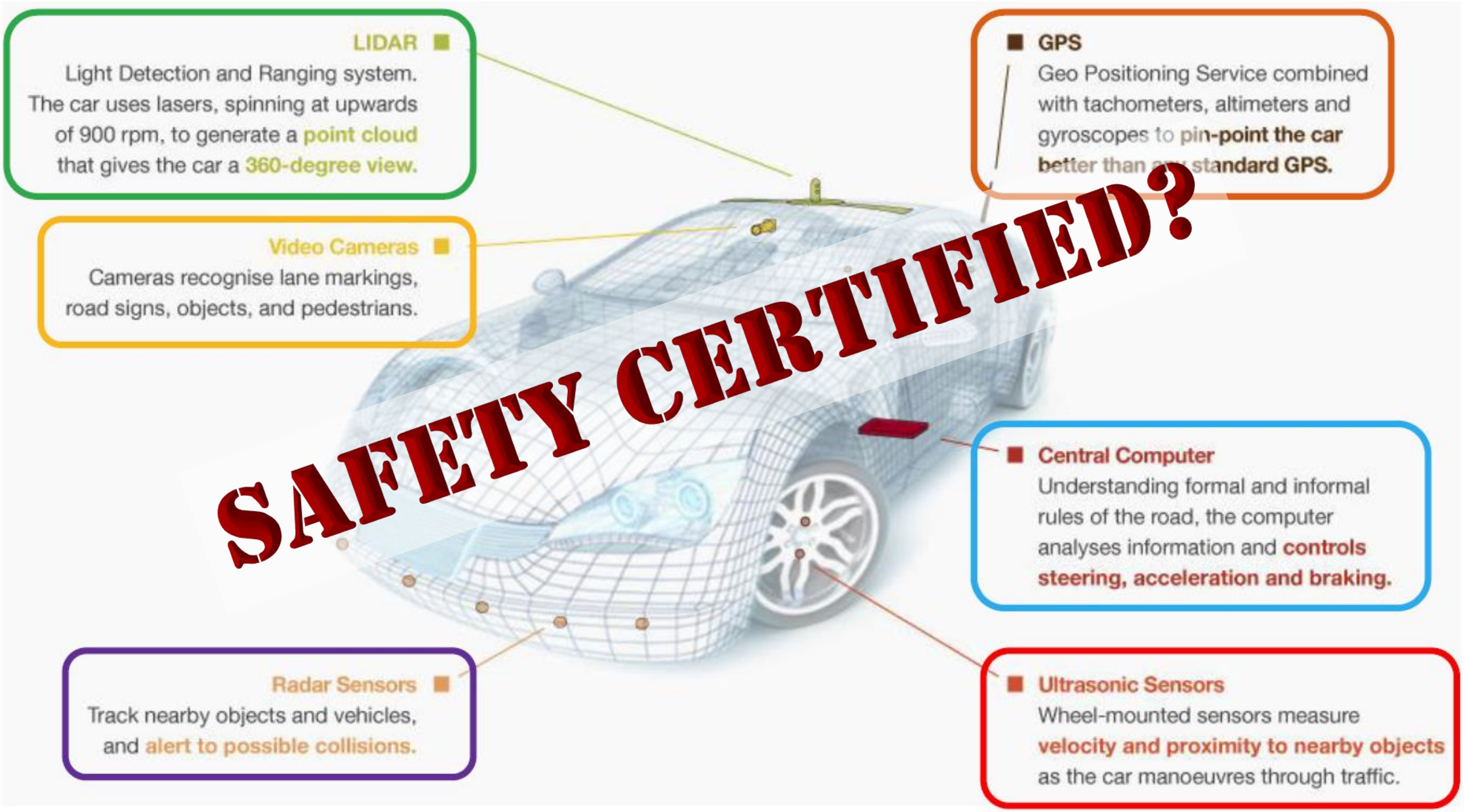
Central Computer

Understanding formal and informal rules of the road, the computer analyses information and **controls steering, acceleration and braking.**

Ultrasonic Sensors

Wheel-mounted sensors measure **velocity and proximity to nearby objects** as the car manoeuvres through traffic.

Self Driving Car Technologies



LIDAR

Light Detection and Ranging system. The car uses lasers, spinning at upwards of 900 rpm, to generate a **point cloud** that gives the car a **360-degree view.**

Video Cameras

Cameras recognise lane markings, road signs, objects, and pedestrians.

Radar Sensors

Track nearby objects and vehicles, and **alert to possible collisions.**

GPS

Geo Positioning Service combined with tachometers, altimeters and gyroscopes to **pin-point the car better than any standard GPS.**

Central Computer

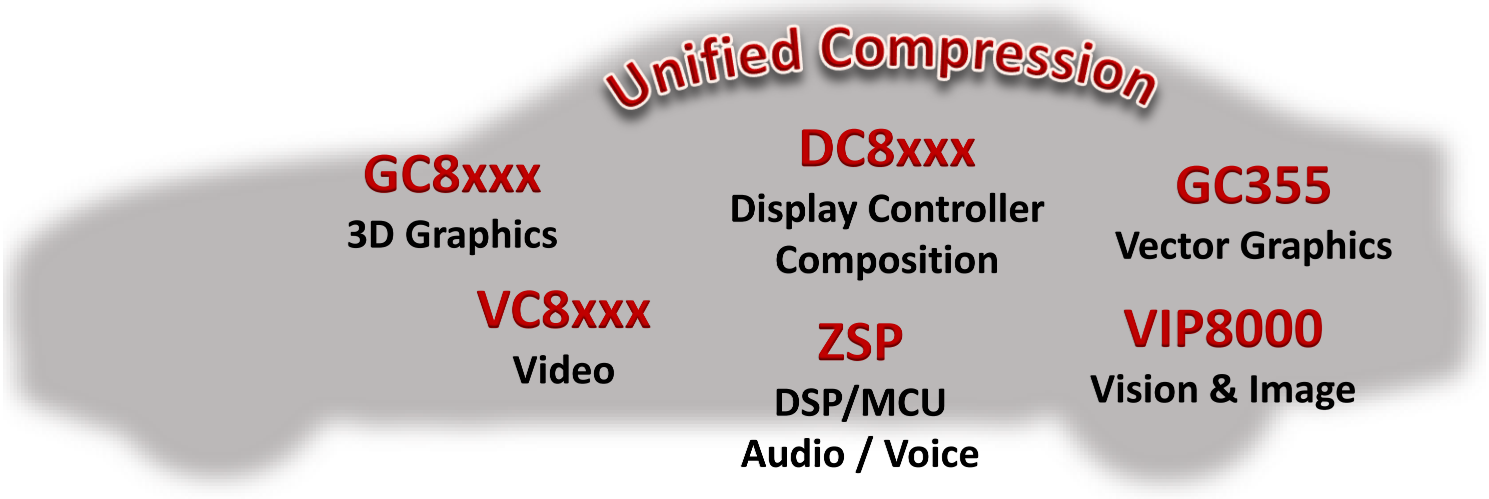
Understanding formal and informal rules of the road, the computer analyses information and **controls steering, acceleration and braking.**

Ultrasonic Sensors

Wheel-mounted sensors measure **velocity and proximity to nearby objects** as the car manoeuvres through traffic.

SAFETY CERTIFIED?

VeriSilicon Automotive Technologies



Unified Compression

GC8xxx
3D Graphics

VC8xxx
Video

DC8xxx
Display Controller
Composition

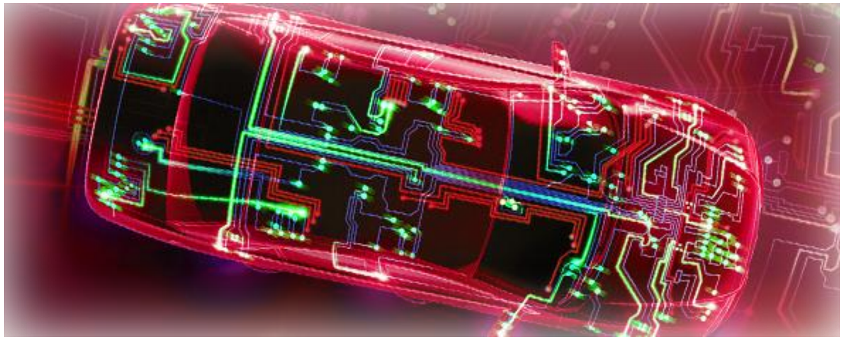
ZSP
DSP/MCU
Audio / Voice

GC355
Vector Graphics

VIP8000
Vision & Image

VIVANTE

VeriSilicon Automotive Technology Leadership



#1 Graphics IP supplier for Automotive LCD Clusters

#2 Graphics IP supplier for In-Vehicle Infotainment Systems

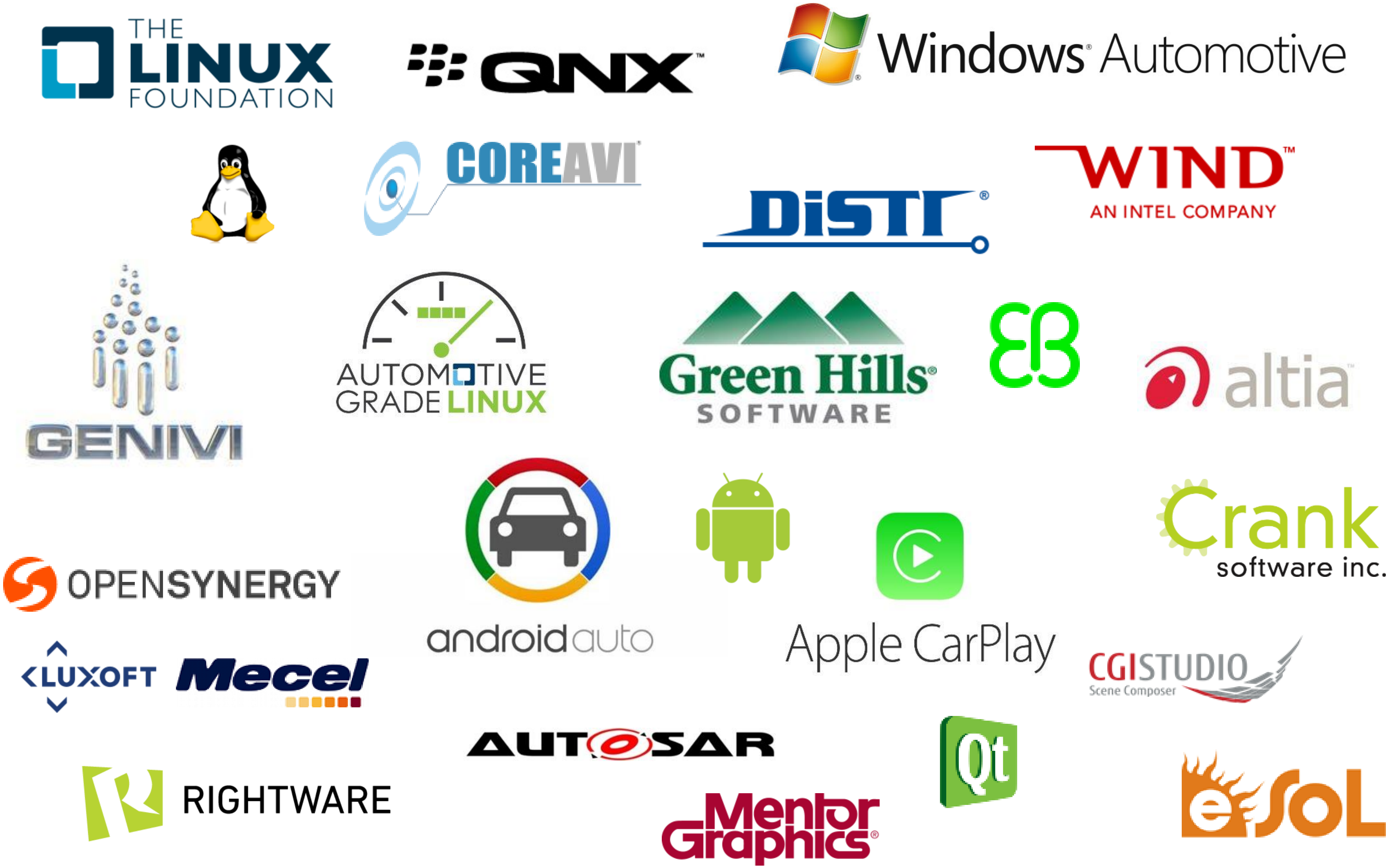
#3 Graphics IP supplier for Rear Seat Entertainment Systems

Vivante Graphics IP is used by 7 of the top 10 automotive OEMs for IVI systems

…and 6 of the top 10 luxury brands for reconfigurable instrument cluster

** Over 20 million cars on the road use Vivante GPUs **

VeriSilicon Automotive Deep Partnerships



VeriSilicon Automotive Deep Customer Experience

DELPHI

DENSO

Visteon®

BOSCH

MAGNA

MAGNETI MARELLI

Pioneer

TRW Automotive

LG

HYUNDAI MOBIS

Continental

Autoliv

TechniSat®

Valeo

ALPINE®

Panasonic

HARMAN

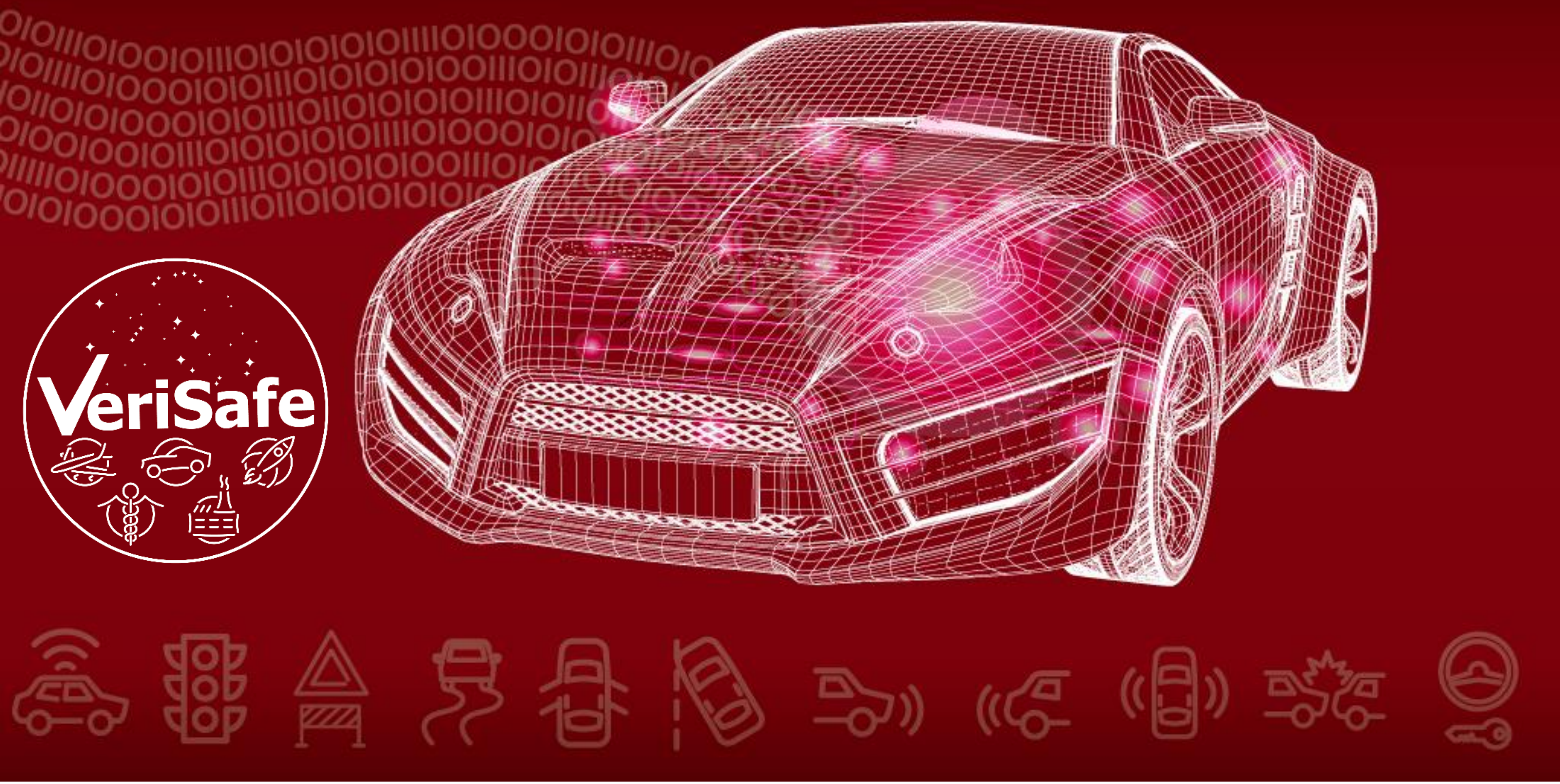
LEAR® CORPORATION

YAZAKI

NIPPON SEIKI CO.,LTD.

VeriSafe Technology

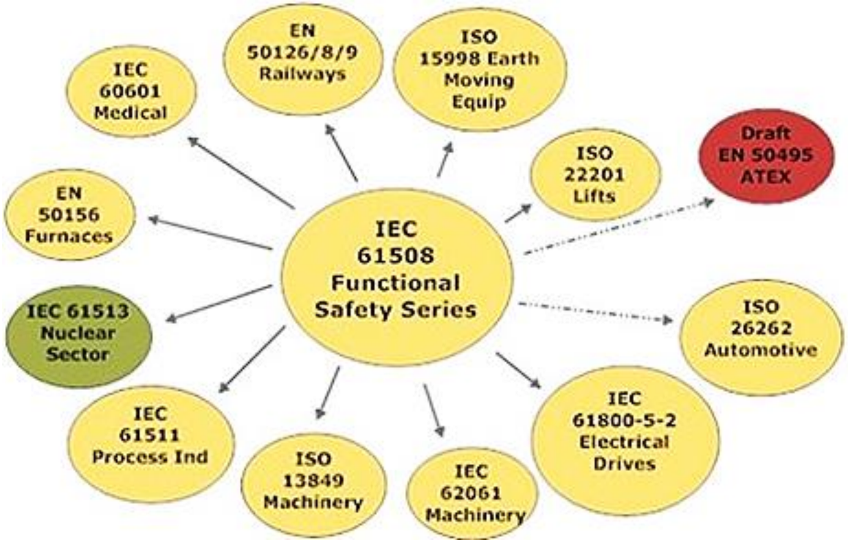
A *combination* of software and hardware technologies / features to bring TRUE safety critical GPU solutions to safety critical markets…



Safety Critical Software?

IEEE

“software whose use in a system can result in unacceptable risk. Safety-critical software includes software whose operation or Failure to operate can lead to a hazardous state, software intended to recover from hazardous states, and software intended to mitigate the severity of an accident”



IEC 61508 Functional Safety Series

IEC 60601 Medical

EN 50126/8/9 Railways

ISO 15998 Earth Moving Equip

ISO 22201 Lifts

Draft EN 50495 ATEX

EN 50156 Furnaces

IEC 61513 Nuclear Sector

ISO 26262 Automotive

IEC 61511 Process Ind

ISO 13849 Machinery

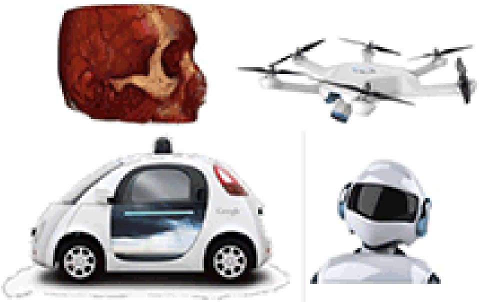
IEC 62061 Machinery

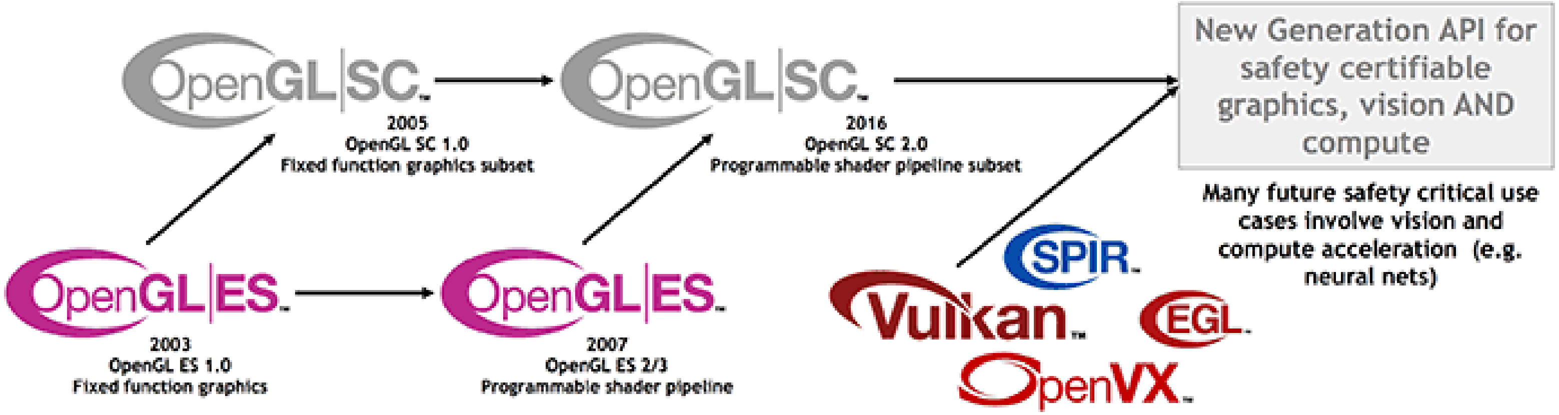
IEC 61800-5-2 Electrical Drives

Software Safety Standards

Avionics	→	DO-178C / ARP 4754A
Medical	→	IEC 60601 Edition 3
Nuclear Power	→	IEC 60880-2
Automotive	→	ISO26262
Industrial	→	IEC 61508 Edition 2

Khronos Safety Critical Working Group





OpenGL SC

2005
OpenGL SC 1.0
Fixed function graphics subset

OpenGL SC

2016
OpenGL SC 2.0
Programmable shader pipeline subset

New Generation API for safety certifiable graphics, vision AND compute

Many future safety critical use cases involve vision and compute acceleration (e.g. neural nets)

OpenGL ES

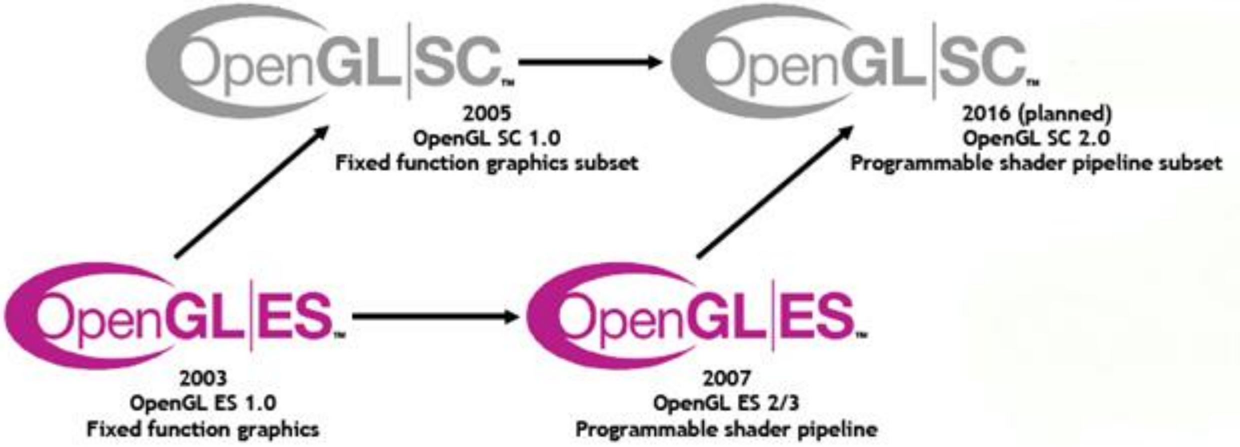
2003
OpenGL ES 1.0
Fixed function graphics

OpenGL ES

2007
OpenGL ES 2/3
Programmable shader pipeline

KHRONOS GROUP

Target Applications for OpenGL SC



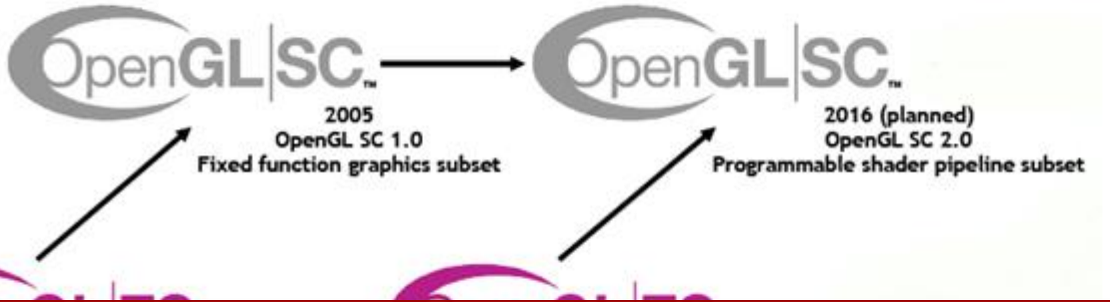
2005
OpenGL SC 1.0
Fixed function graphics subset

2016 (planned)
OpenGL SC 2.0
Programmable shader pipeline subset

2003
OpenGL ES 1.0
Fixed function graphics

2007
OpenGL ES 2/3
Programmable shader pipeline

Goals of OpenGL SC



2005
OpenGL SC 1.0
Fixed function graphics subset

2016 (planned)
OpenGL SC 2.0
Programmable shader pipeline subset

OpenGL SC is specifically designed to be able to be used in safety critical systems. The two primary requirements for any safety critical system are that the system is *deterministic* and *fully testable*.

It will always produce the *same output from a given initial state*, and it is fully testable in accordance with industry safety critical certifications. OpenGL SC is designed to meet FAA Mandated DO-178C Level A and EASA ED-12C Level A for avionics and ISO 262626 for automotive systems.

Safety Critical Systems Require

- Independent certification authority
- Constant Monitoring and Failure Detection
- True Determinism
- Risk Assessments and Mitigation
- Reliability (proven service hours)
- Process and Traceability
- Documentation (planning, development and verification phases)
- Firewalling from non-safety centric processes

Ref: http://vector.com/portal/medien/vector_consulting/publications/Webinar_Safety.pdf

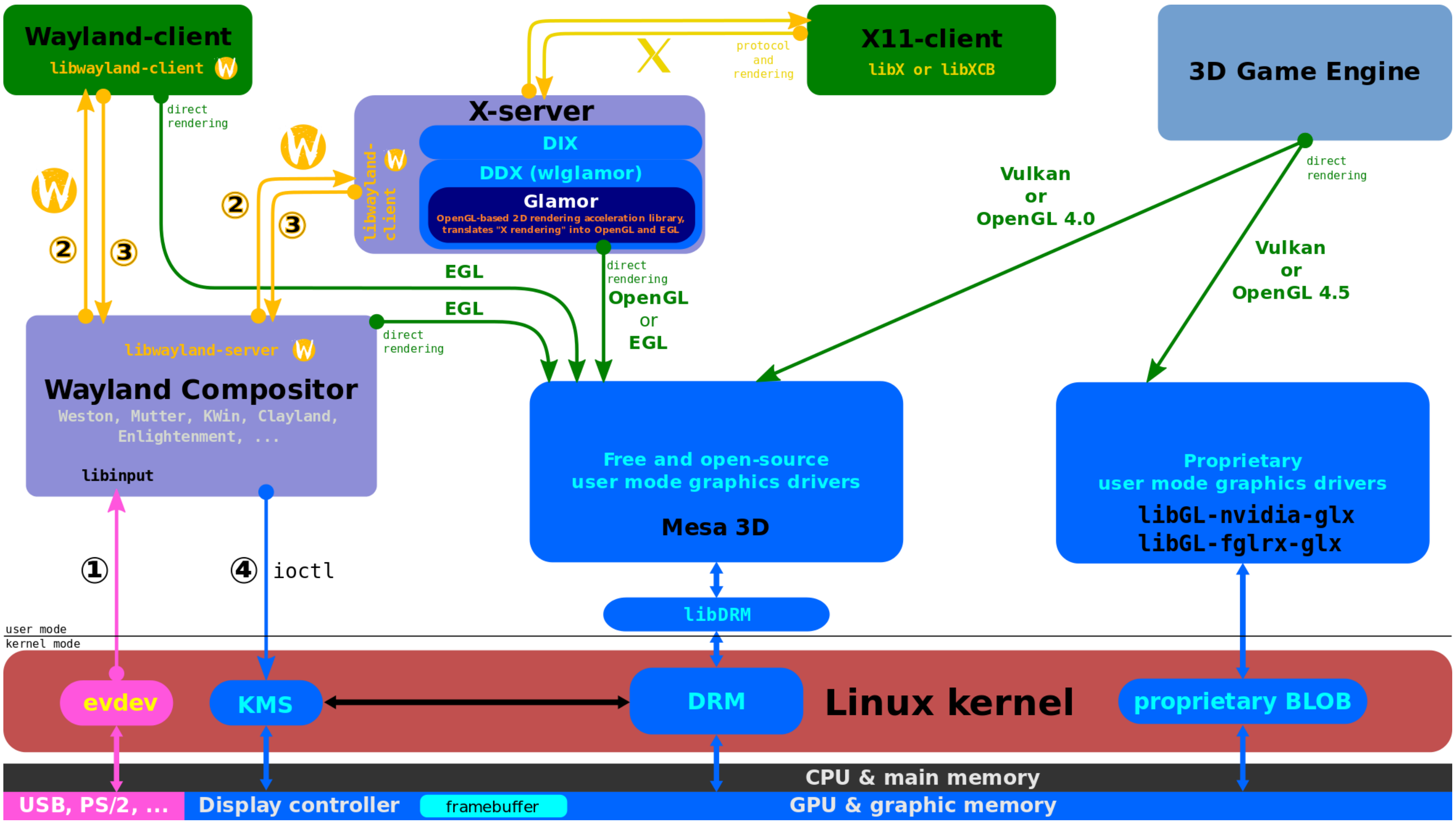
Safety Critical Systems Require

- Independent certification authority
- Constant Monitoring and Failure Detection
- True Determinism
- Risk Assessments and Mitigation
- Reliability (proven service hours)
- Process and Traceability
- Documentation (planning, development and verification phases)
- Firewalling from non-safety centric processes

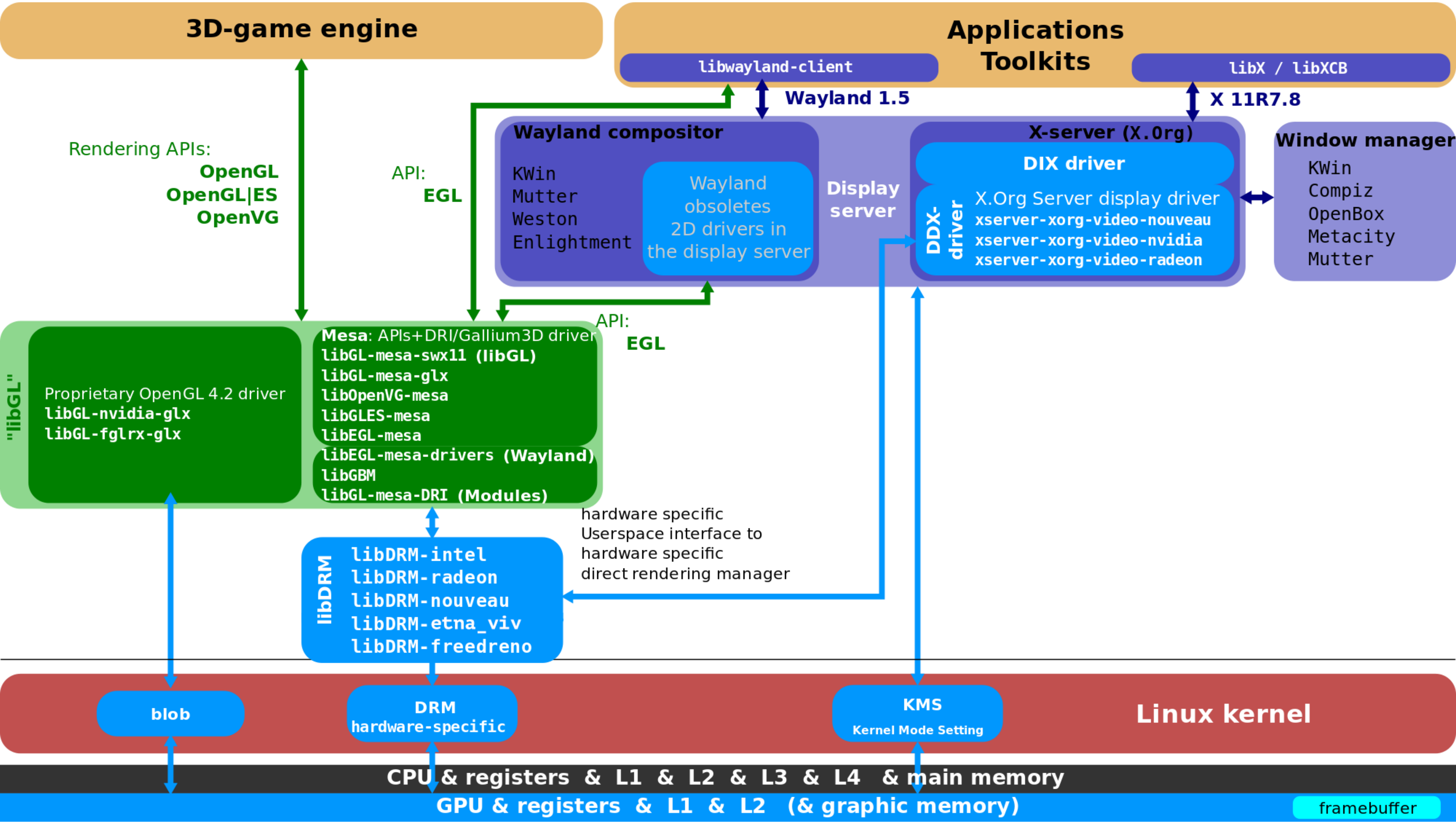
FAR MORE THAN JUST SOURCE CODE!

Ref: http://vector.com/portal/medien/vector_consulting/publications/Webinar_Safety.pdf

Linux OpenGL Ecosystem



Linux OpenGL Ecosystem



3D-game engine

Applications
Toolkits

libwayland-client

libX / libXCB

Wayland 1.5

X 11R7.8

Rendering APIs:
OpenGL
OpenGL|ES
OpenVG

API:
EGL

Wayland compositor

KWin
Mutter
Weston
Enlightment

Wayland obsoletes 2D drivers in the display server

Display server

X-server (X.Org)

DIX driver

DDX-driver

X.Org Server display driver
xserver-xorg-video-nouveau
xserver-xorg-video-nvidia
xserver-xorg-video-radeon

Window manager

KWin
Compiz
OpenBox
Metacity
Mutter

"libGL"

Proprietary OpenGL 4.2 driver
libGL-nvidia-glx
libGL-fglrx-glx

Mesa: APIs+DRI/Gallium3D driver
libGL-mesa-swx11 (libGL)
libGL-mesa-glx
libOpenVG-mesa
libGLES-mesa
libEGL-mesa
libEGL-mesa-drivers (Wayland)
libGBM
libGL-mesa-DRI (Modules)

API:
EGL

libDRM

libDRM-intel
libDRM-radeon
libDRM-nouveau
libDRM-etna_viv
libDRM-freedreno

hardware specific
Userspace interface to
hardware specific
direct rendering manager

blob

DRM
hardware-specific

KMS
Kernel Mode Setting

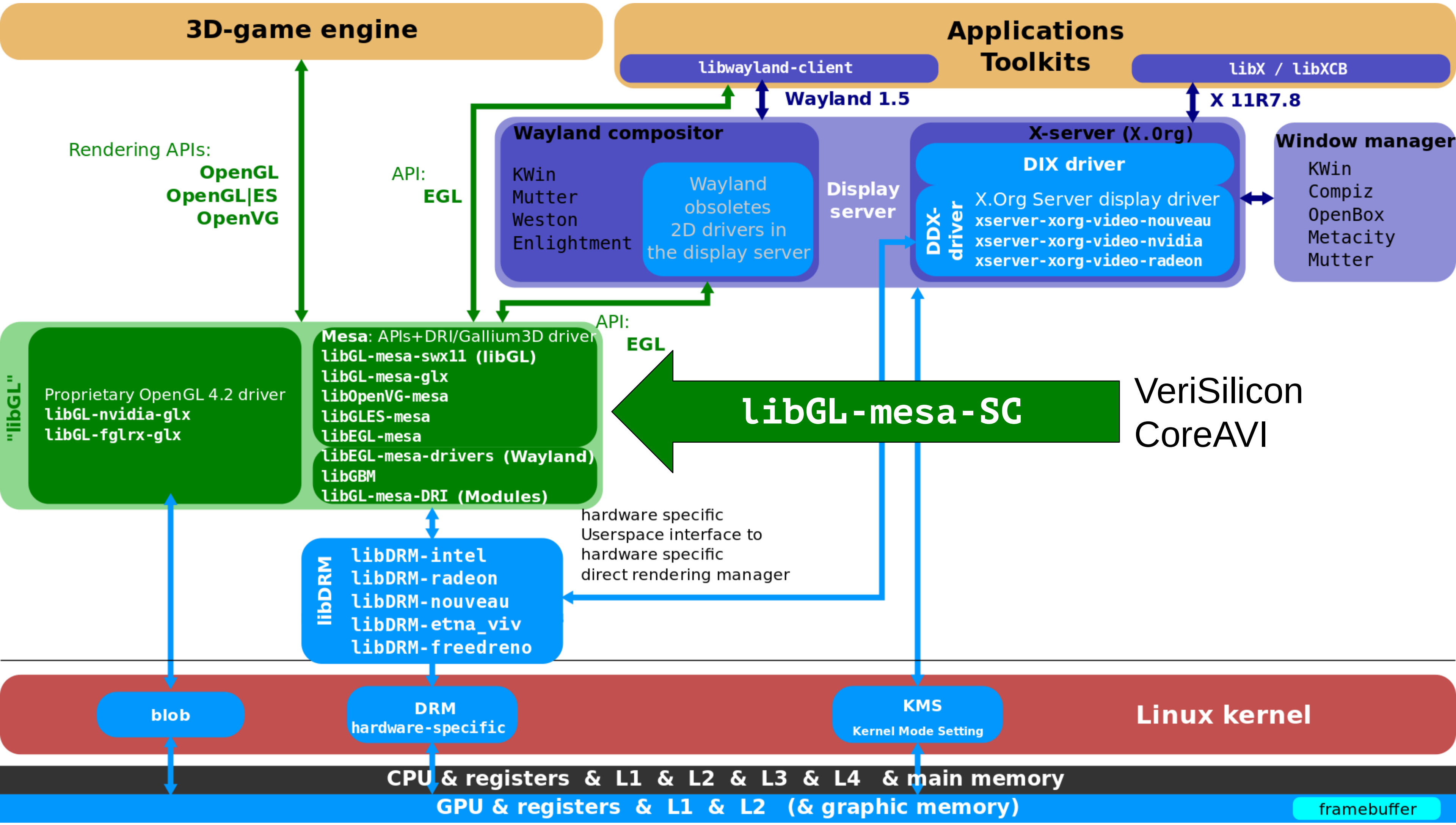
Linux kernel

CPU & registers & L1 & L2 & L3 & L4 & main memory

GPU & registers & L1 & L2 (& graphic memory)

framebuffer

Linux OpenGL Ecosystem



3D-game engine

Applications
Toolkits

libwayland-client

libX / libXCB

Wayland 1.5

X 11R7.8

Rendering APIs:
OpenGL
OpenGL|ES
OpenVG

API:
EGL

Wayland compositor
KWin
Mutter
Weston
Enlightment

Wayland obsoletes 2D drivers in the display server

Display server

X-server (X.Org)
DIX driver
DDX-driver
X.Org Server display driver
xserver-xorg-video-nouveau
xserver-xorg-video-nvidia
xserver-xorg-video-radeon

Window manager
KWin
Compiz
OpenBox
Metacity
Mutter

"libGL"

Proprietary OpenGL 4.2 driver
libGL-nvidia-glx
libGL-fglrx-glx

Mesa: APIs+DRI/Gallium3D driver
libGL-mesa-swx11 (libGL)
libGL-mesa-glx
libOpenVG-mesa
libGLES-mesa
libEGL-mesa
libEGL-mesa-drivers (Wayland)
libGBM
libGL-mesa-DRI (Modules)

API:
EGL

libGL-mesa-SC

VeriSilicon
CoreAVI

hardware specific
Userspace interface to
hardware specific
direct rendering manager

libDRM
libDRM-intel
libDRM-radeon
libDRM-nouveau
libDRM-etna_viv
libDRM-freedreno

blob

DRM
hardware-specific

KMS
Kernel Mode Setting

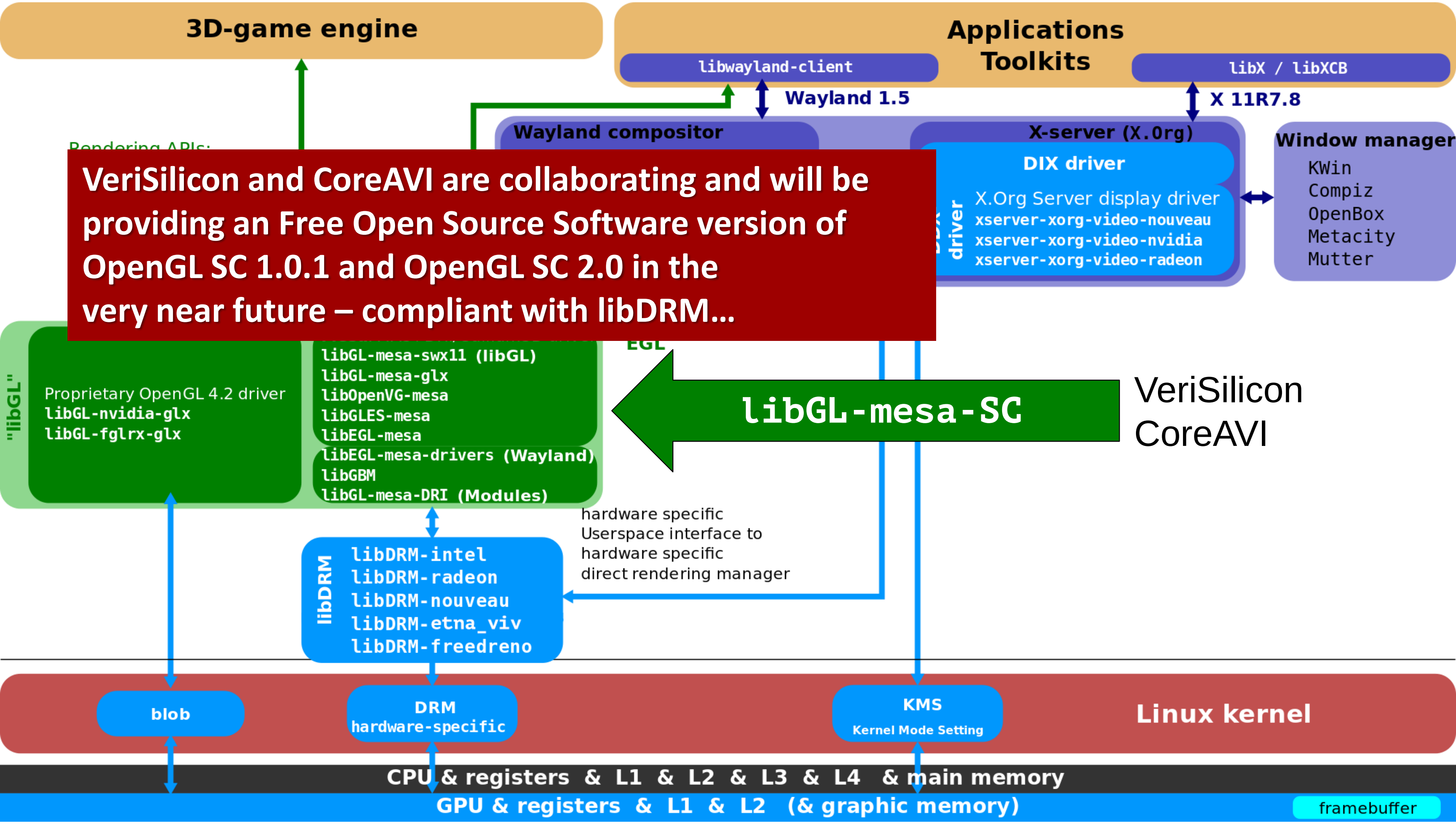
Linux kernel

CPU & registers & L1 & L2 & L3 & L4 & main memory

GPU & registers & L1 & L2 (& graphic memory)

framebuffer

Linux OpenGL Ecosystem + OpenGL SC



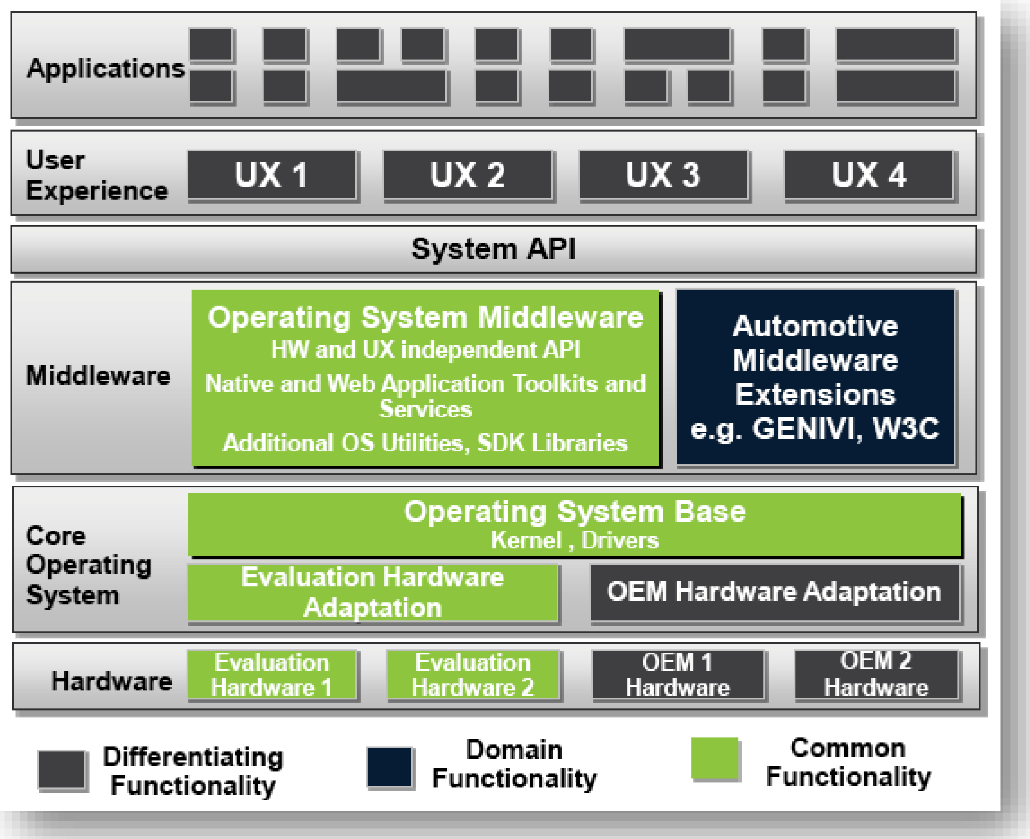
VeriSilicon and CoreAVI are collaborating and will be providing an Free Open Source Software version of OpenGL SC 1.0.1 and OpenGL SC 2.0 in the very near future – compliant with libDRM…

libGL-mesa-SC

VeriSilicon
CoreAVI

AGL Built on Linux

AUTOMOTIVE GRADE LINUX



Applications

User Experience: UX 1, UX 2, UX 3, UX 4

System API

Middleware

Operating System Middleware
HW and UX independent API
Native and Web Application Toolkits and Services
Additional OS Utilities, SDK Libraries

Automotive Middleware Extensions
e.g. GENIVI, W3C

Core Operating System

Operating System Base
Kernel , Drivers

Evaluation Hardware Adaptation

OEM Hardware Adaptation

Hardware: Evaluation Hardware 1, Evaluation Hardware 2, OEM 1 Hardware, OEM 2 Hardware

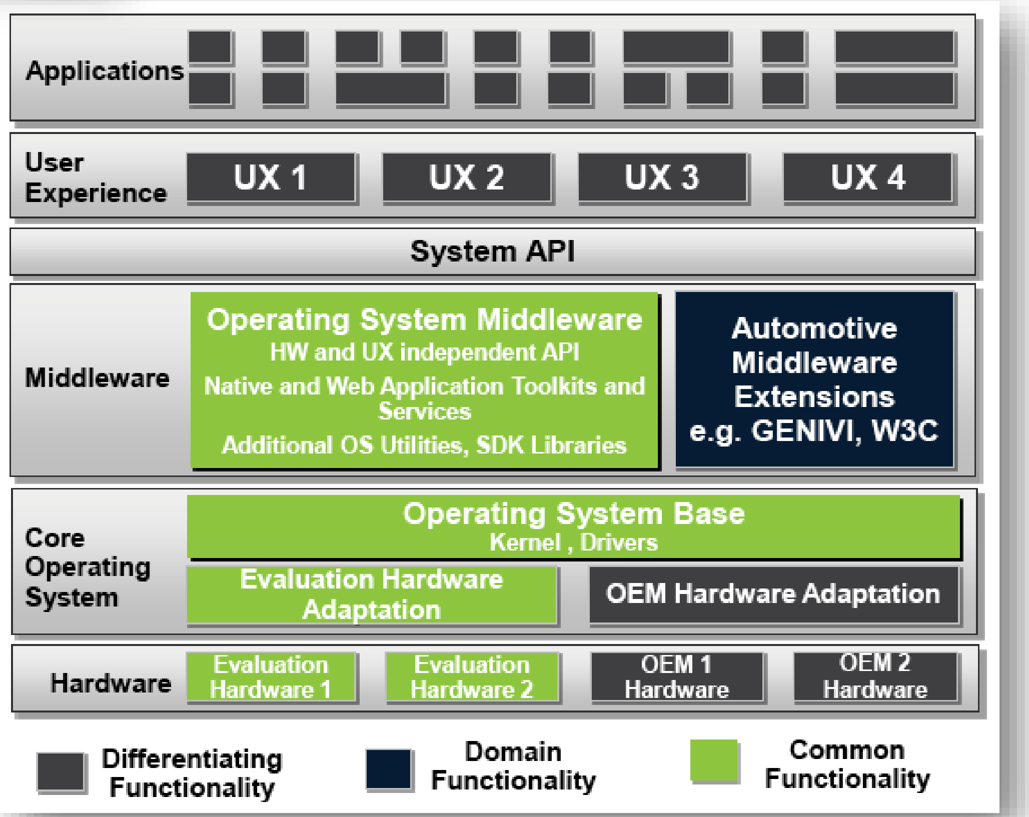
■ Differentiating Functionality

■ Domain Functionality

🟩 Common Functionality

Wide Automotive Industry Support

AUTOMOTIVE GRADE LINUX



Applications

User Experience

UX 1

UX 2

UX 3

UX 4

System API

Middleware

Operating System Middleware
HW and UX independent API
Native and Web Application Toolkits and Services
Additional OS Utilities, SDK Libraries

Automotive Middleware Extensions
e.g. GENIVI, W3C

Core Operating System

Operating System Base
Kernel , Drivers

Evaluation Hardware Adaptation

OEM Hardware Adaptation

Hardware

Evaluation Hardware 1

Evaluation Hardware 2

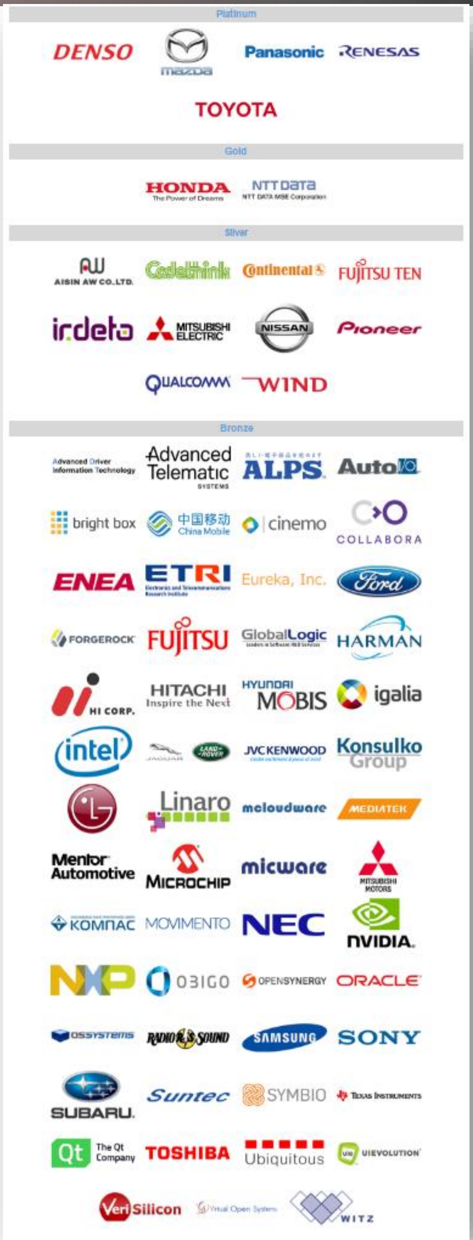
OEM 1 Hardware

OEM 2 Hardware

⬛ Differentiating Functionality

⬛ Domain Functionality

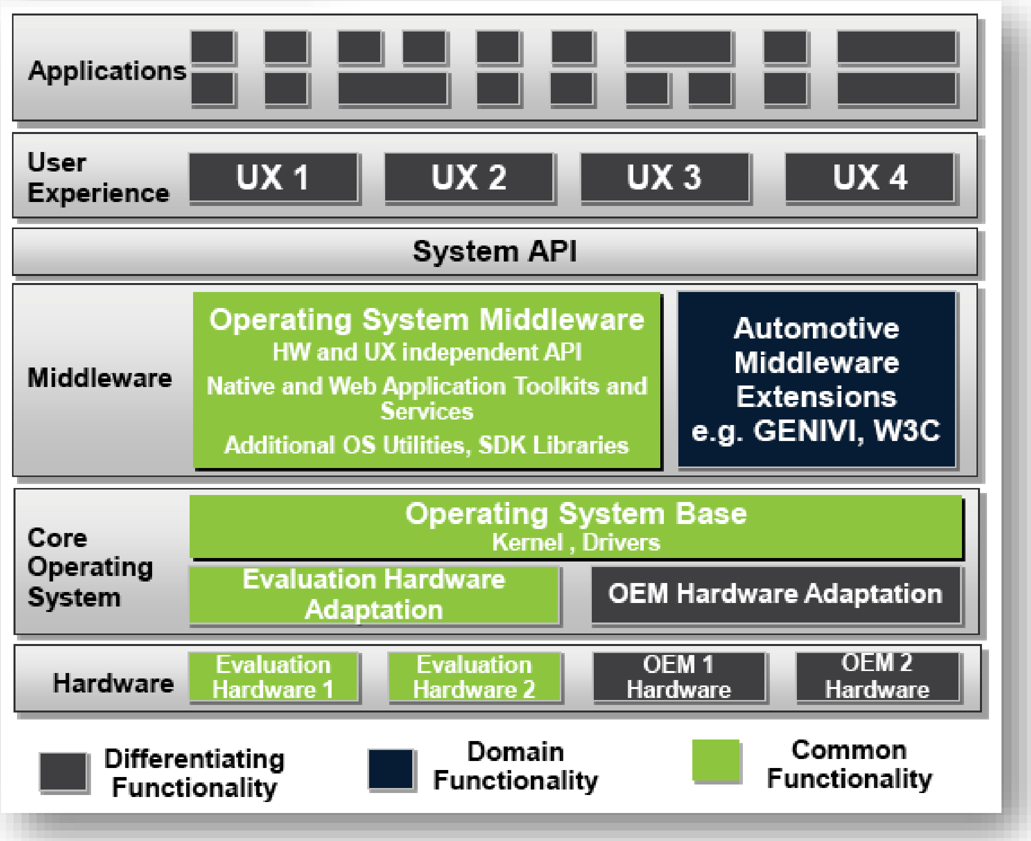
🟩 Common Functionality



AGL + OpenGL SC

The OpenGL ES software stack is commonly the largest and most complicated software in a cluster / IVI system…and the source of most software failures.

OpenGL SC will help…



Applications

User Experience: UX 1, UX 2, UX 3, UX 4

System API

Middleware

Operating System Middleware
HW and UX independent API
Native and Web Application Toolkits and Services
Additional OS Utilities, SDK Libraries

Automotive Middleware Extensions
e.g. GENIVI, W3C

Core Operating System

Operating System Base
Kernel , Drivers

Evaluation Hardware Adaptation

OEM Hardware Adaptation

Hardware: Evaluation Hardware 1, Evaluation Hardware 2, OEM 1 Hardware, OEM 2 Hardware

⬛ Differentiating Functionality

🟦 Domain Functionality

🟩 Common Functionality

OpenGL SC Implementations

ArgusCore SC1™

CoreAVI's ArgusCore SC1 drivers are a superset of Khronos' OpenGL SC 1.0 API specification (OpenGL for safety critical applications). The OpenGL SC 1.0 graphics libraries are implemented to support a fixed function graphics rendering pipeline. Today, CoreAVI's ArgusCore SC1 libraries are used extensively in certified avionics display systems utilizing fixed function safety critical graphics applications.

ArgusCore SC2™

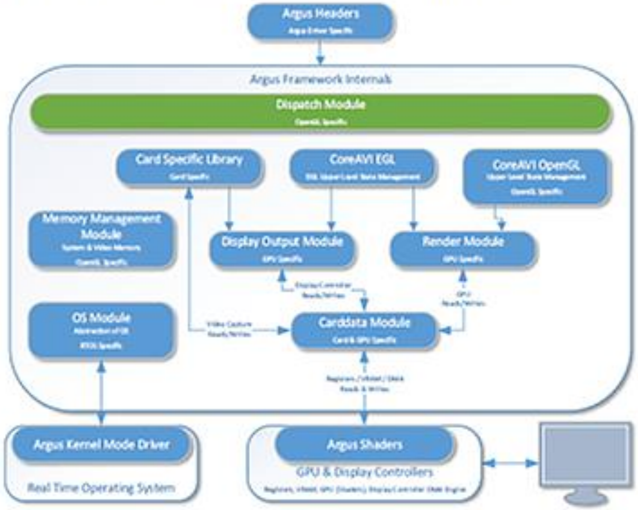
CoreAVI completed and deployed the industry's very first OpenGL SC 2.0 graphics driver. CoreAVI's ArgusCore SC2 drivers are a superset of Khronos' OpenGL SC 2.0 API specification. The OpenGL SC 2.0 graphics libraries support a programmable graphics rendering pipeline. The drivers allows safety critical applications to take greater advantage of the performance gains by utilizing modern graphics processor shader engines while still maintaining the ability to achieve the highest levels of safety certification. **ArgusCore SC2** enables users to deploy modern GPU shader programs in safety certifiable environments.

CertCore178™

Available today, CoreAVI's complete FAA DO-178C and EASA ED-12C Level A certification data packages support the use of ArgusCore SC graphics drivers in any FAA DO-178C / EASA ED-12C avionics safety certification.

Modular and Adaptable Architecture

Based on a highly modular architecture, CoreAVI can optimize their customer's specific applications and quickly adapt the OpenGL libraries to new hardware platforms, operating systems and even add customer specific features. Video capture enhancements, display controller settings, and deterministic memory management modules can be quickly modified to address unique device specific requirements.

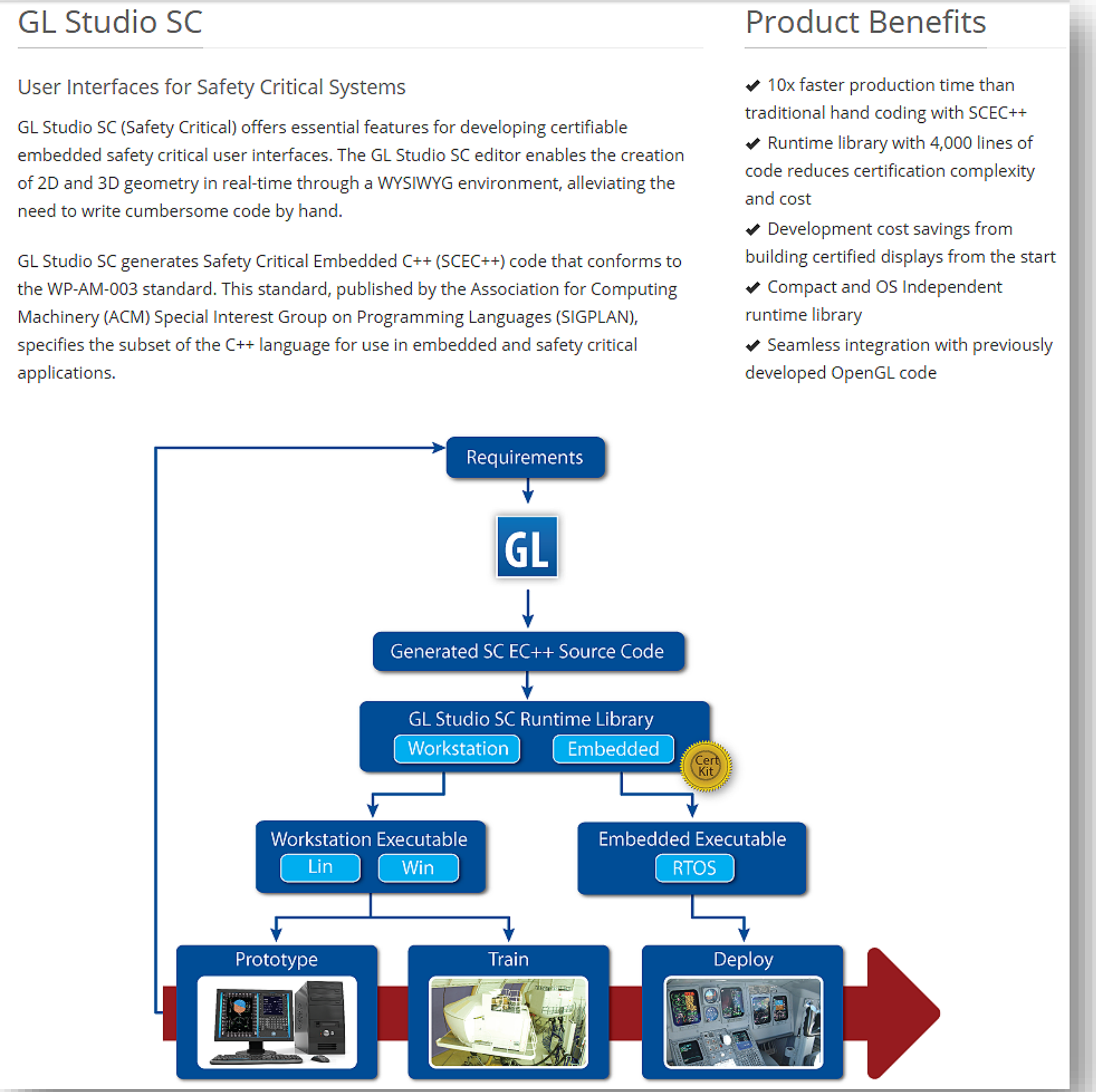


ArgusCore SC Modular Architecture

OpenGL SC Implementations

DiSTI®

When the Interface Matters



GL Studio SC

User Interfaces for Safety Critical Systems

GL Studio SC (Safety Critical) offers essential features for developing certifiable embedded safety critical user interfaces. The GL Studio SC editor enables the creation of 2D and 3D geometry in real-time through a WYSIWYG environment, alleviating the need to write cumbersome code by hand.

GL Studio SC generates Safety Critical Embedded C++ (SCEC++) code that conforms to the WP-AM-003 standard. This standard, published by the Association for Computing Machinery (ACM) Special Interest Group on Programming Languages (SIGPLAN), specifies the subset of the C++ language for use in embedded and safety critical applications.

Product Benefits

- ✓ 10x faster production time than traditional hand coding with SCEC++
- ✓ Runtime library with 4,000 lines of code reduces certification complexity and cost
- ✓ Development cost savings from building certified displays from the start
- ✓ Compact and OS Independent runtime library
- ✓ Seamless integration with previously developed OpenGL code

Autonomous Vehicle Safety Critical Engineering

The University of Texas at Austin
Electrical and Computer Engineering

VeriSilicon

UT AutoSafe Lab

Launching this Fall to tackle the issues surrounding safety critical engineering and autonomous vehicles…

Initial Goals

The University of Texas at Austin
Electrical and Computer Engineering

VeriSilicon

UT AutoSafe Lab

- Techniques and practices to make open source safe for use in Safety Critical systems – i.e. Automotive Grade Linux, Zephyr (RTOS for embedded systems – Linux Foundation Initiative), OpenGL SC, etc.
- Analysis of existing government regulations for safety critical solutions – FAA DO-178C, DO-330, etc. and its applicability to other safety critical applications such as automotive – this includes specific recommendations of changes that should be made to make the standards broader based instead of a specific application like aviation.
- Analysis / auditing of current autonomous vehicle solutions for potential points of failure and unexpected behaviors – The result of such audits would be concrete recommendations to improve such solutions from a safety critical standpoint.
- Recommendations and engineering solutions relative to COTS hardware and software systems to improve their role in safety critical systems – with a special focus on graphics and vision systems.

Contacts

The University of Texas at Austin
Electrical and Computer Engineering

VeriSilicon

UT AutoSafe Lab

Dr. Ahmed Tewfik - tewfik@austin.utexas.edu
Rick Tewell – rick.tewell@verisilicon.com

Launching this Fall to tackle the issues surrounding safety critical engineering and autonomous vehicles…

Questions